54

Evaluation of Deterministic and Statistical Failure Detection Algorithms for Aircraft Applications

by

Nusawardhana

Sarjana Teknik Elektro (B.S.),
Bandung Institute of Technology, Indonesia (1993)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1997

Author .....
Department of Aeronautics and Astronautics
August 25, 1997

Certified by .....
John J. Deyst, Jr.
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Professor Jaime Peraire
Chairman, Department Committee on Graduate Students

Evaluation of Deterministic and Statistical Failure Detection Algorithms for Aircraft Applications

by

Nusawardhana

Submitted to the Department of Aeronautics and Astronautics
on August 25, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The new failure detection algorithms ; the Robust Failure Detection and Isolation and the Marginalized Likelihood Ratio Test; were applied to the problem of attitude determination based on inertial instrument measurements and the Global Positioning System measurements.

Direct application of the algorithms will cause significant false alarms, since the Global Positioning System solution degrades severely under the following conditions: insufficient number of observable satellites, narrowly scattered satellite constellation and obstructed Global Positioning System receiver antenna.

Providing the filters a different set of gains in both the predetection stage and in the failure detection stage will reduce the number of these false alarms. In the preprocessing stage, the different set of gains reduces the number of excessive spikes in the Global Positioning System solutions. In the failure detection stage, using the different set of gains allows the failure detection filters distinguish an unreliable Global Positioning System solution from the instrument failure. The failure detection filters will recognize that any difference between the two measurements are due to the low quality of the Global Positioning System solution rather than the bias or ramp failure in the instruments.

Thesis Supervisor: John J. Deyst, Jr.
Title: Professor of Aeronautics and Astronautics

Acknowledgments

Your word is a lamp to my feet and a light to my path. I have sworn an oath and confirmed it, to observe your righteous ordinance - Psalm 119.105,106.

I would like to express my thanks to Professor John J. Deyst, my thesis advisor, who provided guidance and support for my research and academic matters. Besides suggesting the topic for experiment, he carefully read the thesis, proposed improvement, and evaluated the revision. I am very grateful for his patience and help, especially when it came to language problem.

I would like also to thank Professor R. John Hansman, my first academic advisor, for his guidance at the beginning of my life at MIT. Both Professors provided support to me when things did not go as it had been planned.

I would like to express my gratitude to Professor J.K. Kuchar for our discussions of various problems in flight dynamics and detection theory. His patience in answering my curiosity should be greatly appreciated.

I would like to thank P.T. Industri Pesawat Terbang Nusantara Indonesia that provided the financial support for my graduate study at MIT. Special gratitude should be addressed to Dr. Said D. Jenie and Dr. Sulaiman Kamil for their encouragement and their effort so that I could pursue a higher education and degree.

I would like to thank Richard Kornfeld for preparing the instruments for the experiment and the discussion of the result. I would also like to thank for his letting me do experiments with his flight test data. I am also grateful for Eric Duran's generosity to provide the vehicle for the experimental purpose.

I would like to thank Sean George for proof reading and correction in English writing. It would have been difficult for me to submit this thesis if I had not received his help.

My fiancee Quinike and I were separated for two years. The patience of hers should be appreciated. I hope we can learn a lot from this period for the future. I shall never forget her love and prayer as I shall always keep my commitment to her.

My parents are those who will stand by me whatever my situation is. I am very blessed to have them in this life. I am very grateful for their support, especially when I was in misery. It will never be forgotten.

There are friends in Boston, Washington D.C, and New York, who made my life very

vibrant. I would like to express my gratitude to them. I gratefully acknowledge the sincere support from Vera Aurelia and Mrs. Josephine Handojo. I will always remember the way they encouraged me when things seemed difficult to handle. It will always be a relief to remember the precious moments. As the river flows from the small river to the great sea, where will it carry the float of the future in this world ?

Thanks to Arian Kurniawan, I could concentrate on finishing the Thesis without having to worry about the shelter in the final week.

I would like to express my gratitude to my host family in Boston, Ernest and Barbara Beevers. They had given precious help and attention. They helped me in preparing the thesis, especially in correcting my English, introduced me with American culture, especially the Thanksgiving. Thanks to them, I know a nice couple : James and Michelle Harper as well as their daughter Sarah. I always appreciate these nice and warm friendships.

This thesis was prepared at the Instrumentation, Control and Estimation Division at Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, in the time period of 1996-1997. All labmates had given their contribution in their own unique ways to make the life in the laboratory enjoyable. I would always appreciate the friendships.

Cambridge, Summer 1997

Nusawardhana

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In April 9, 1994, the Boeing company rolled-out its newest commercial aircraft, the Boeing 777. The Boeing 777 was designed as a brand new airplane and it embodies many new features enabled by new technologies. It was designed to meet the economical requirements of the future airline industry, and to be the commercial jet for the twenty-first century, [43], [41], [40].

One interesting feature of the Boeing 777 is its fly-by-wire flight-control system. In the fly-by-wire system, the pilot's control inputs from cockpit are turned into electrical signals and those signals are transmitted through wires to computers and eventually to actuators of movable surfaces on the wings and the tail. The states of the control surfaces are sensed by various sensors, and these measurements are fed back along with inputs from inertial and air data sensors to the computers. The computers in the fly-by-wire system serve several purposes : monitoring the system performance, processing signals from one system to the other, performing active control over the flight and others.

Modern fly-by-wire systems incorporate large numbers of sensors and actuators. For the sake of reliability, there are redundant sensors and actuators to perform critical functions. Redundancy also enables the aircraft tolerate failures and continue to fly without loss of functionality. Design for redundancy was an integrated part of the development of the Boeing 777.

It is extremely important that the redundant system know the condition of its redundant elements. Once there is a failure, the system must recognize the failure and reconfigure itself

by isolating the failed element. Detecting a failure by comparing the outputs of replicated elements is a well understood and fairly straightforward procedure. However, it is not always the case that redundancy is achieved by replicating elements. In some systems, information from diverse elements is combined through dynamic models, to monitor the health of the system and identify failures.

There are several approaches to identifying the presence of failures in dynamics system using dynamic models. The approach typically employs probability theory, detection theory and linear system theory. Information from various sensors is combined in the mathematical dynamic model, and the failure detection system compares the outputs of these sensors with the output of the mathematical model. When there is significant discrepancy between the output from a sensor and the output predicted by the mathematical model, the failed elements can be diagnosed as failed and isolated, with appropriate reconfiguration of the remaining unfailed system elements.

The model based failure detection process can be used to reduce the number of sensors onboard. The output of the mathematical model acts as if it is real instrument and thus represents redundancy. The drawback of this approach lies in the level of the actual inaccuracy of the mathematical model. In certain extreme situations, the model may not follow the actual dynamics of the real system due to unmodeled dynamics nonlinearities, or parameter uncertainties. Thus, care must be taken to assure that the system functions correctly over the entire flight envelope.

There is an increasing trend toward fly-by-wire systems. Redundancy management and failure detection are key technologies which are enabling this trend. Some important questions that must be addressed in designing fault-tolerant fly-by-wire systems are:

1. how mathematical models serve to augment redundant elements,

2. the relative performance of different failure detection algorithms,

3. the validity of failure detection systems, and

4. the range of operating conditions over which the failure detection system will work.

This Thesis will attempt to develop systematic methods for addressing these questions and others.

1.2 Thesis Coverage

In December 1995 and January 1996, there were two new failure detection algorithms that appeared in the engineering society journals. *The Marginalized Likelihood Ratio* appeared in IEEE Transaction of Automatic Control [22], and a *Robust Failure Detection and Isolation* [35], was published in the Proceeding of American Control Conference. These two algorithms are the main focus of this thesis. These algorithms were derived using different approaches. The derivations of the two were analyzed and discussed in the sequel in order to provide insight into the algorithms. The performance of these algorithms is analyzed and applied to a real problem of failure diagnosis. The algorithms are compared to each other and to the well-known Generalized Likelihood Ratio test algorithm. Comparisons were based on detection capability and the computational complexity. For detection capability, the time of detection and the sensitivity of detection were used as performance measures. In addition, the performance of the algorithms is evaluated using measures of missed alarm and false alarm probability.

Finally, improvements in the performance of the failure detection algorithms using actual instrument data are proposed and tested. The data obtained from actual instrument outputs was corrupted by noise and other spurious signals. The failure detection algorithms in the real experiment faced difficulties in distinguishing failures from random disturbances, when they were applied without adaptive gain compensation. Improvements by providing adaptive gains for the failure detection filters under certain circumstances is proposed and was tested.

1.3 Thesis Outline

Chapter 2 provides background of the failure detection algorithm. The underlying principle of the failure detection algorithm is the main theme of this chapter. The relationship between reliability, redundancy and failure detection algorithm issue initiates the chapter. Hypothesis testing that underlies the detection principle is presented for both types of redundancy : hardware redundancy and analytical redundancy. This chapter ends after the measure of the failure detection performance is given.

Chapter 3 presents the failure detection algorithms for linear discrete-time dynamic systems based on statistics. Before proceeding to the main discussion, the structure of failure

detection process is discussed. The well-known Generalized Likelihood Ratio test for failure detection in linear dynamic system follows the discussion. The Marginalized Likelihood Ratio test for failure detection that appeared recently in 1996 is another statistical based failure detection algorithm presented in this chapter. It has the properties that enhance the performance of the GLR test. Numerical examples that demonstrate the GLR and the MLR tests for failure detection in a linear dynamic system close the discussion of statistical failure detection algorithm.

Chapter 4 presents another approach of failure detection. In this Chapter, the algorithm developed deterministically becomes the focus of discussion. Instead of characterizing the failure, the system dynamics, and the noise, with statistical attributes, the deterministic algorithm characterizes them with a set descriptions. The algorithm is based on the worst possible set of condition. The robust residual, data for decision making, is important for detection purpose. Numerical examples is given how to generate this type of residual from a robust estimator before proceeding to the discussion of the robust failure detection algorithm. The development of the algorithm is given for both case : unknown dynamic model and unknown dynamic and disturbance models. A numerical example of how to use this algorithm is also presented here.

Chapter 5 presents the experiment of failure detection on the attitude determination problem. Data from the real measurements are manipulated, and failures are injected at certain time locations. The performance of both statistic failure detection algorithms and deterministic one are then analyzed. The way to improve the performance of each algorithm is presented in this chapter.

Chapter 6 presents the summary, conclusions and suggestions for future works.

In Appendixes, derivations of equations in the prior chapters are given.

Chapter 2

Reliability, Redundancy, and Failure Detection-Isolation

The purpose of this chapter is to indicate the importance of redundancy for reliability. To achieve an acceptable degree of reliability, redundancy is often the most effective method, especially for flight critical systems. Redundancy in practice does not always involve replicating similar components but sometimes physical components can be replaced by a behavior model run in a computing system. The latter redundancy principle, together with replicating redundancy, can maintain an acceptable level of reliability if the behavioral model is sufficiently accurate. Section 2.1 will discuss this concept.

When a component fails, it is important for the system to recognize the presence of failure and isolate the failed component. This is the purpose of failure detection and isolation algorithms. In Section 2.2, a typical failure detection algorithm is described. Even though the isolation problem is not pursued in this thesis, its principle is briefly described.

There are many kinds of failure detection algorithms. The problem of choosing which algorithm to apply in a particular situation is resolved if the required criteria are known. Measures of failure detection algorithm performance are presented in Section 2.3.

2.1 Reliability and Redundancy

The performance of a dynamic system over a specified duration of time can often be predicted by applying reliability analysis. Reliability analysis makes use of probability theory to determine the failure probability of a system. It is shown in [44] that systems with cer-

tain configurations have better reliability than the others. The following example is taken from [44] to illustrate how different configurations of components can improve the reliability of a system's performance.

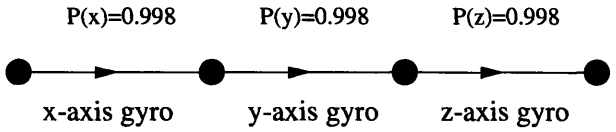

Figure 2-1: Reliability model of 3 one-degree-of-freedom gyros

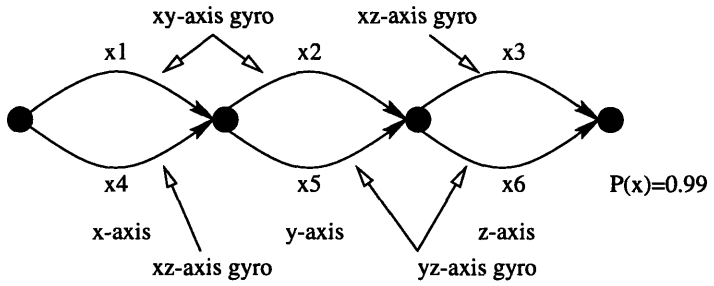

Figure 2-2: Reliability model of 3 two-degree-of-freedom gyros

The first configuration, illustrated in Figure 2-1, consists three high reliability one-degree-of-freedom gyros that measure angular rate in three orthogonal directions (x, y, z). The second configuration, illustrated in Figure 2-2, uses three two-degree-of-freedom gyros to do the same function as the first configuration. The reliability of the individual gyros used in the second configuration is 5 times lower than for the gyros of the first configuration. The probability of failure-free operation of the first configuration is

$$P(x_{good})P(y_{good})P(z_{good}) = (0.998)^3 = (1 - 0.002)^3 \approx 1 - 3 \times 0.002 = 0.994.$$

The failure-free probability of the x-axis of the second configuration is

$$P(x_{good}) = P(x_1 + x_4) = 1 - P(\bar{x}_1\bar{x}_4) = 1 - 0.01^2 = 0.9999$$

and for failure-free operation of the entire system is

$$P(x_{good})P(y_{good})P(z_{good}) = (1 - 0.0001)^3 \approx 1 - 0.0003 = 0.9997$$

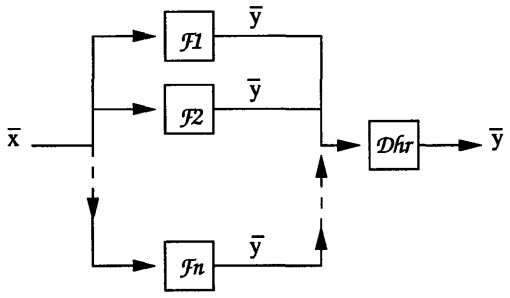

Figure 2-3: Redundancy with parallel structure

which is twenty times more reliable than the first system.

In the redundant system of Figure 2-3, several signal paths perform the same function. Failure of one or more paths still allows the remaining paths to perform the function properly. The reliability is eventually maintained. There are two approaches that can be used to provide redundancy, i.e.

1. hardware or direct redundancy

2. analytical redundancy

Each will be discussed in the following paragraphs. *Hardware redundancy* applies replicated hardware elements to provide a desired function. The hardware elements are sometimes distributed spatially around the system to provide protection against localized damage. Such schemes typically operate in triple or quadruple redundant configurations and redundant outputs are compared for consistency. When a failure occurs, the output from the failed sensor is isolated and the output of an unfailed sensor is used to perform system functions. The hardware redundancy principle can be described mathematically as

$$\mathcal{F}_i \quad : \quad \bar{x} \rightarrow \bar{y} \qquad (2.1)$$

$$i = 1, \ldots, n$$

$$\mathcal{F}_i \quad = \quad \mathcal{F}_j \qquad (2.2)$$

$$i \neq j$$

$$\bar{y} \quad = \quad \mathcal{D}_{hr}(\mathcal{F}_1, \ldots, \mathcal{F}_n) \qquad (2.3)$$

Figure 2-3 is an illustration of the Equations (2.1) - (2.3). The $\mathcal{F}_i$ represents the function of each of the hardware components. There are n components and each component receives

inputs $\bar{x}$ and provides outputs $\bar{y}$. Each component performs the same function and the decision function $\mathcal{D}_{hr}$ determines which outputs are to be used.

Analytic redundancy applies dissimilar inputs to provide some desired function. This information is related to the input variables by a certain functional relationship. This redundancy principle can be described by the following mathematical expressions.

$$\begin{aligned} \mathcal{F}_i &: \bar{x}_i \rightarrow \bar{y} \qquad (2.4) \\ & \quad i = 1, \ldots, n \\ \mathcal{F}_i &\neq \mathcal{F}_j \qquad (2.5) \\ & \quad i \neq j \\ \bar{y} &= \mathcal{D}_{ar}(\mathcal{F}_1, \ldots, \mathcal{F}_n) \qquad (2.6) \end{aligned}$$

Different functions $\mathcal{F}_i$ receive different inputs $\bar{x}_i$ to produce the same output $\bar{y}$. The output $\bar{y}$ used for system operation is determined by the decision function $\mathcal{D}_{ar}$.

Both hardware redundancy and analytic redundancy can improve the reliability of dynamic systems. The major disadvantage encountered with hardware redundancy is the cost and the additional space required to accommodate the redundant equipment. The major disadvantage with analytic redundancy is the additional complexity and software required.

To maintain the reliability of a redundant system, the system itself should have the capability to recognize the presence of failures of its components. It can be catasthropic if the system uses the output of failed components. Thus, the ability to recognize the failure of its components and to reconfigure its structure is an important attribute of the system. Hence care must be taken to assure that the possibility of common mode failure is remote.

Hardware redundancy uses direct voting to identify failures. The voting process assumes that each sensor is independent of the others. This assumption raises the question of whether the identical sensors operating in the same condition can be considered independent.

Various approaches to *Failure Detection and Isolation (FDI)* using analytical redundancy have been reported in technical papers, for example [15], [12], [29], [13], and [42]. Several survey papers [49], [17], examine several FDI algorithms and provide comment on the characteristics, advantages, disadvantages, and trade-offs involved in the various FDI techniques.

2.2 Failure Detection and Isolation

To determine whether or not a component has failed, detection algorithms are often used. These algorithms are often based on statistical hypothesis testing because ordinary noise and other errors in the outputs of sensors make the failure detection task uncertain. In this approach, each of the possible scenarios corresponding to a hypothesis is defined. For FDI purposes, the following hypotheses are usually used

- H_0 : the hypothesis of no-failure mode
- H_1 : the hypothesis of failure mode

The object is to determine whether H_0 or H_1 holds at each point in time that a test is made, based on insufficient information to determine with certainty which hypothesis holds. This approach is known as a binary hypotheses testing, [46], [50], in which either H_0 or H_1 is true. At each time the hypothesis test is conducted and a decision is made, one of the following possibilities will occur:

1. H_0 true, H_0 chosen
2. H_0 true, H_1 chosen
3. H_1 true, H_1 chosen
4. H_1 true, H_0 chosen

The result of a hypothesis test is specified in terms of a decision rule. Mathematically, a deterministic decision rule is a function $\hat{H}(.)$ that uniquely maps every possible $N-$dimension of observation y_t to one of the two hypotheses, H_0 and H_1, i.e. :

$$\hat{H} : y_t \in \mathcal{R}^N \longrightarrow \{H_0, H_1\}.$$

From this perspective, choosing the function $\hat{H}(.)$ is equivalent to partitioning the observation space $\mathcal{Y} = \{y_1, \ldots, y_N\}$ into two disjoint decision regions, corresponding to the values of y for which each of the two possible decisions are made, [50].

From all the possibilities given, it is desirable that the decision rule will minimize probability of false alarm and probability of miss alarm. The probability of *false alarm* corresponds to the situation that H_1 is chosen while H_0 is true, and the probability of *miss-alarm* corresponds to the situation that H_0 is chosen while H_1 is true.

Binary hypothesis testing has a geometric interpretation. As the observations are obtained, there is a certain region in decision space that belongs to H_0 and there is also another region that belongs to H_1. The decision space, the H_0 region, and the H_1 region will be described in the sequel.

2.2.1 Hypothesis Testing in Hardware Redundant Systems

Consider the following measurement model of H_0:

$$Y_t = H_t X_t + V_t \tag{2.7}$$

where

- t : time at which an observation is made
- Y_t : observation vector, $Y_t \in \mathcal{R}^n$
- X_t : state vector, $X_t \in \mathcal{R}^r$
- H_t : measurement matrix of rank r, $H_t \in \mathcal{R}^{n \times r}$, and $r < n$
- V_t : measurement noise with covariance matrix R

Similarly, the measurement model for the failure hypothesis H_1 is :

$$Y_t = H_t X_t + V_t + \Upsilon(t, t_0) \tag{2.8}$$

and $\Upsilon(t, t_0)$ is vector of change.

Hypothesis testing, in the following discussion, will be developed in parity space [4], which is a linear transformation of redundant measurements to facilitate failure detection. The parity vector that spans the parity space is a measure of the relative consistencies between the elements of the redundant observation Y_t. The mapping from the observation vector Y_t into a parity vector ζ_t can be described by the following equation :

$$\zeta_t = C_t Y_t \tag{2.9}$$

The matrix C is a special matrix such that $C_t \in \mathcal{R}^{(r-n) \times n}$. For linear time-invariant system, the subscript of t can be dropped. The rows of the of C are an orthonormal basis of the

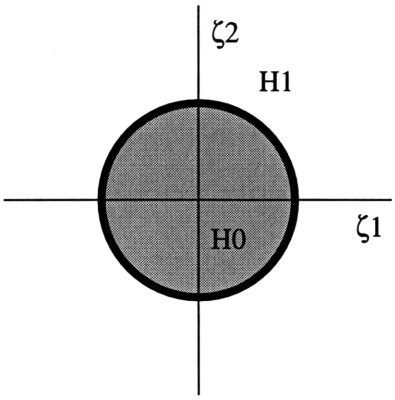

Figure 2-4: Typical 2 dimension decision region

left null space of H, i.e.

$$\begin{aligned} CH &= 0 \\ CC^T &= I_{r-n} \end{aligned} \tag{2.10}$$

A method for constructing the C matrix can be found in [14], [38] and Appendix A. When there is no additive failure, the parity equation is

$$\zeta_t^0 = C(H_t X_t + V_t) = CV_t \tag{2.11}$$

and hence ζ_t^0 is a linear transformation of unfailed sensor errors. In the presence of additive failure, the parity equation becomes

$$\zeta_t^1 = C(H_t X_t + V_t + \Upsilon(t, t_0)) = CV_t + C\Upsilon(t, t_0) \tag{2.12}$$

Note that the system state is eliminated in both (2.11) and (2.12) and only sensor errors and failure effects remain. When the magnitude of additive failure is known *a priori*, the decision space can be illustrated in Figure 2-4. This figure describes the parity space spanned by $\zeta \in \mathcal{R}^2$.

Decision function is eventually developed based on the following criteria :

1. magnitude of parity vector

2. direction of parity vector

There are several ways to infer the presence or absence of a failure from the parity vector and compare it to a certain threshold. One method applies the likelihood ratio for hypothesis testing. The following approach is used in [24], and the derivation can be obtained in [11].

In the absence of failures, the parity vector is Gaussian random variable with the following properties :

$$E[\zeta_t] = \mathbf{0} \tag{2.13}$$

$$C_\zeta = cov[\zeta_t] = C_t R_t C_t^T \tag{2.14}$$

The detection function is defined by

$$\mathcal{D} = \zeta^T C_\zeta^{-1} \zeta \tag{2.15}$$

If $\mathcal{D}$ exceeds a certain threshold value, it is likely that a failure has occurred inside the system. This situation indicates that the the magnitude of $\zeta = \|\zeta\|$ is large compared to what would be expected if there were no failure.

2.2.2 Hypothesis Testing in Analytically Redundant Systems

The analytic redundancy approach can be formulated using state-space methods. The state-space equations characterize the dynamic behavior of the systems and the properties of the measurement outputs. The state-space equations, thus, provide complete description of the systems of interest, [10].

The hypotheses developed for FDI purposes are usually described by the following discrete time state-space equations :

$$H_0: \quad X_{t+1} = A_t X_t + B_t U_t + W_t \tag{2.16}$$

$$Y_t = H_t X_t + J_t U_t + V_t \tag{2.17}$$

$$H_1: \quad X_{t+1} = A_t X_t + B_t U_t + W_t + \Gamma \Upsilon_x(t, t_0) \tag{2.18}$$

$$Y_t = H_t X_t + J_t U_t + V_t + \Xi \Upsilon_y(t, t_0) \tag{2.19}$$

where

- $t \in \mathcal{R}^1$: time of observation
- $X_t \in \mathcal{R}^n$: state vector
- $Y_t \in \mathcal{R}^r$: observation vector
- $U_t \in \mathcal{R}^m$: input vector
- $A_t \in \mathcal{R}^{n \times n}$: dynamic matrix
- $B_t \in \mathcal{R}^{n \times m}$: control input matrix
- $H_t \in \mathcal{R}^{r \times n}$: state measurement matrix
- $J_t \in \mathcal{R}^{r \times m}$: input measurement matrix
- W_t : process noise with covariance $Q_t \in \mathcal{R}^{n \times n}$
- V_t : measurement noise with covariance $R_t \in \mathcal{R}^{n \times n}$
- $\Upsilon_{x,y}(t, t_0)$: dynamic vector of additive failure
- Γ, Ξ : failure matrices

Hypothesis testing based on the parity space approach can be extended to analytically redundant systems. In a hardware redundant system, one observation is adequate to construct the parity space. This is not the case for analytically redundant systems.

Instead of considering the left null space of the observation matrix H_t, the left null space of the observability matrix is used in an analytically redundant system. The observability matrix is defined by

$$\mathcal{O}(H, A) = \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n-1} \end{bmatrix} \tag{2.20}$$

Consider the array of measurements from time l, $(1 \leq l \leq n)$ until time t. The measurement equation from equation (2.17) will give the following description :

$$\mathcal{Y}_{t-l+1}^{t} = \mathcal{O}_l \mathcal{X}_{t-l+1} + \mathcal{J}_l(B, J)\mathcal{U}_{t-l+1}^{t} + \mathcal{J}_l(I_n, 0)\mathcal{W}_{t-l+1}^{t} + \mathcal{V}_{t-l+1}^{t} \tag{2.21}$$

where

$$\mathcal{O}_l(H, A) = \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{l-1} \end{bmatrix}$$

is the subset of the observability matrix and

$$\mathcal{Y}_{t-l+1}^{t} = \begin{bmatrix} Y_{t-l+1} \\ Y_{t-l+2} \\ \vdots \\ Y_t \end{bmatrix}, \quad \mathcal{X}_{t-l+1}^{t} = \begin{bmatrix} X_{t-l+1} \\ X_{t-l+2} \\ \vdots \\ X_t \end{bmatrix} \quad \textit{and} \quad \mathcal{U}_{t-l+1}^{t} = \begin{bmatrix} U_{t-l+1} \\ U_{t-l+2} \\ \vdots \\ U_t \end{bmatrix}.$$

The system dynamic noise and the measurement noise arrays are

$$\mathcal{W}_{t-l+1}^{t} = \begin{bmatrix} W_{t-l+1} \\ W_{t-l+2} \\ \vdots \\ W_t \end{bmatrix} \quad \text{and} \quad \mathcal{V}_{t-l+1}^{t} = \begin{bmatrix} V_{t-l+1} \\ V_{t-l+2} \\ \vdots \\ V_t \end{bmatrix}.$$

The covariance matrix of $\mathcal{W}_{t-l+1}^{t}$ will be a block diagonal matrix with l Q's on the diagonal, and the covariance matrix of $\mathcal{V}_{t-l+1}^{t}$ will be block diagonal matrix with l R's on the diagonal. For further reference, the definition above will be described in the following operational form:

$$I_l \otimes Q \quad \text{and} \quad I_l \otimes R$$

Two matrices from the Equation (2.21) are defined as follows

$$\mathcal{J}_l(B,J) = \begin{bmatrix} J & \dots & \dots & \dots \\ HB & J & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ HA^{l-2}B & \dots & HB & J \end{bmatrix}$$

$$\mathcal{J}_l(I_n,0) = \begin{bmatrix} 0 & \dots & \dots & \dots \\ H & 0 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ HA^{l-2} & \dots & H & 0 \end{bmatrix}$$

If we define the following quantities

$$\text{observation}: \quad \tilde{\mathcal{Y}}_{t-l+1}^t = \mathcal{Y}_{t-l+1}^t - \mathcal{J}_l(G,J)\mathcal{U}_{t-l+1}^t \tag{2.22}$$

$$\text{noise}: \quad \tilde{\mathcal{V}}_{t-l+1}^t = \mathcal{J}_l(I_n,0)\mathcal{W}_{t-l+1}^t + \mathcal{V}_{t-l+1}^t \tag{2.23}$$

$$\tag{2.24}$$

with covariance matrix

$$\mathcal{R}_l = \mathcal{J}_l(I_n,0)(I_l \otimes Q)\mathcal{J}_l^T(I_n,0) + I_l \otimes R, \tag{2.25}$$

a regression model that resembles Equation (2.7) will be obtained.

$$\tilde{\mathcal{Y}}_{t-l+1}^t = \mathcal{O}_l \mathcal{X}_{t-l+1} + \tilde{\mathcal{V}}_{t-l+1}^t \tag{2.26}$$

The parity space of order $l, (1 \leq l \leq n)$ is the left null space of the observability matrix, namely the set

$$\mathcal{S}_l = span\{v | v^T \mathcal{O}_l(H,A) = 0\} \tag{2.27}$$

The parity vector ζ_t is defined by

$$\zeta_t = v^T \tilde{\mathcal{Y}}_{t-l+1}^t. \tag{2.28}$$

When there is no additive failure, the parity vector will be

$$\begin{aligned} \zeta_t &= v^T(\mathcal{O}_l X_{t-l+1} + \tilde{\mathcal{V}}_{t-l+1}^t) \\ &= v^T \tilde{\mathcal{V}}_{t-l+1}^t \end{aligned} \tag{2.29}$$

The geometrical interpretation of the parity space is the same as that of the hardware redundancy case.

The hypothesis testing presented in this chapter with parity space is not the only way to identify the failure. There are other ways as we will see in the subsequent chapters. However, these algorithms still work on the same framework of hypothesis testing.

2.2.3 Failure Isolation

It is important to provide isolation to the failed components after the failure is detected so the system can disregard the output of the failed components. There are basically two approaches for fault isolation [39] :

1. *Structured residuals*

   which are designed to generate residual vectors so that in response to a fault only a fault-specific subset of the residual elements are non-zero

2. *Fixed direction residuals*

   where the residuals are designed to lie in fixed and fault-specific direction in the residual space in response to each different fault. This is the basic idea of the *failure detection filter* [5], [26].

Residuals are processed observations that carry the failure information.

For the systems with hardware redundancy, as described in Section 2.2.1, once a failure is detected, the failure is isolated to the sensor which maximizes the isolation decision function. The isolation decision function for the $j-$th component is defined as [24]:

$$\mathcal{D}_{hr_j} = \frac{\left(\zeta^T C_\zeta^{-1} \underline{c_j}\right)^2}{\underline{c_j}^T C_\zeta^{-1} \underline{c_j}} \tag{2.30}$$

where $\underline{c_j}$ is the $j-$th column of matrix C defined in the parity derivation.

In this thesis, the problem of isolation and reconfiguration is not pursued. The discussion of the subsequent chapters will be for detection problems only.

2.3 Measures of Failure Detection Algorithm Performance

Some indices of performance, that are often used for evaluation of detection algorithms are the following [4], [38].

1. Probability of false alarm or false alarm rate.

   The probability of false alarm is defined as

   $$P_{FA} = P_{\theta_0}[H_1 \quad chosen]$$

2. Mean delay for detection.

   The mean delay for detection is defined by

   $$t_{MD} = E_{\theta_1}(t_a)$$

   where t_a is the alarm time of the change detector

3. Probability of missed detection.

   $$P_{MD} = P_{\theta_1}[H_0 \quad chosen]$$

4. Accuracy of estimates of change time, direction and magnitude of failure.

5. Computational complexity of the algorithm

It was discussed in Section 2.2 that two hypotheses were used for failure detection purposes. Suppose that we have a one dimensional observation with the probability densities of H_0 and H_1 as shown in Figure 2-5. Before the failure occurs, the observation has θ_0 mean and after the failure occurs, the observation has θ_1 mean. The threshold θ_{th} is chosen to determine whether the observation is failed or not. Figure 2-5 also illustrates the one-dimensional decision region. In this particular region, the false-alarm probability and the

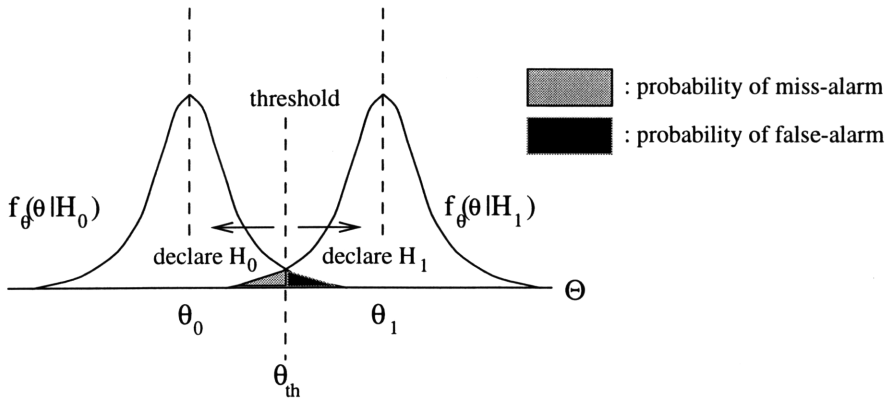

Figure 2-5: Decision region and its probability densities

miss-alarm probability are defined in [46], [4], [14] as

$$p_{FA} = \int_{\theta_{th}}^{\infty} f_\theta(\theta|H_0)d\theta$$
$$p_{MA} = \int_{-\infty}^{\theta_{th}} f_\theta(\theta|H_1)d\theta$$

It is not an easy task to set the threshold θ_{th} to satisfy a certain performance. The trade-off between p_{FA} and p_{MA} will determine the performance of a failure detection system.

Chapter 3

Statistical Failure Detection Algorithms

Failure detection algorithms based on statistical hypothesis testing are the main focus of this chapter. Hypothesis testing is applied to the statistics of the observation. Before exploring failure detection algorithms, Section 3.1 introduces the structure of failure detection processes. Each stage of the process in this detection structured will be discussed.

The discussion of several statistical failure detection algorithms will be given in Sections 3.2, 3.3, and 3.4. In Section 3.2, the discussion is focused on the well-known GLR, Generalized Likelihood Ratio, test for failure detection in linear dynamic systems. Section 3.3 will present the MLR, Marginalized Likelihood Ratio algorithm from Gustafsson, which can provide better performance than the GLR test. The efficient implementation of the MLR test algorithm is discussed in Section 3.4.

Numerical examples that illustrates how to apply the GLR test and the MLR test algorithms will be given in Section 3.5.

3.1 The Structure of The Failure Detection Process

The failure detection process essentially works in a certain structure, [7] (Figure 3-1) :

- residual generation
- decision making

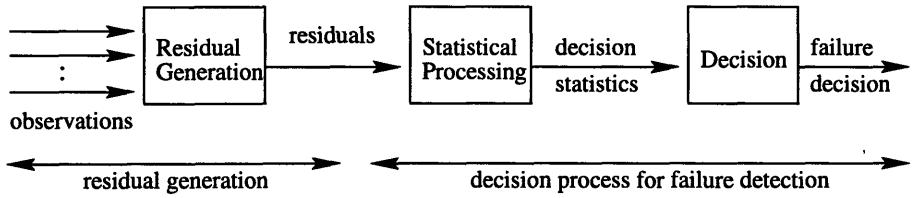

Figure 3-1: The structure of the failure detection process

The formal statement of the residual generation process is given as follows, [17]. *For a given system, e.g.,*

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= h(x(t), u(t), t) \end{aligned}$$

where

- $x(t)$ *is the state vector,*
- $u(t)$ *is the input vector, and*
- $y(t)$ *is the observation vector,*

and under the assumption that a residual vector $r(t)$ *carrying information about a particular fault exists, find an algorithm that generates* $r(t)$ *when the fault has occurred, under the conditions that there is time constraint for residual generation and there are uncertainties in*

1. *time of failure;*
2. *nominal system model;*
3. *process noise and measurement noise models.*

Residuals carry information that reflects possible changes of interest in the observed signal or system. In residual generation, the observations are processed to enhance the effect of possible failures. The residuals represent the difference between various functions of the observations and the expected values of these functions in the failure-free mode. When a failure occurs there is a difference between the values of the residuals in the presence of failure and the values that can be expected in the normal mode.

To generate the residuals, it is essential that the following conditions exist.

1. Knowledge of normal behavior of the system.

2. Definitiveness of the "faulty" behavior.

3. Existence of analytical redundancy relations.

4. Availability of observations that reflect the failure.

5. Satisfactory accuracy of the redundant information.

The second element of a failure detection algorithm is decision making. The residual will be processed statistically to provide information as to whether or not a failure occurs. The decision function can be formulated using a hypothesis testing framework. The level of complexity in decision making varies. The simplest method is to compare the instantaneous residuals with a certain threshold. More complex approaches, e.g., sequential probability ratio test of Wald, [48], can also be used to increase performance of the system. The choice of which statistical method to use for decision making depends on the need based on the criteria given in Chapter 2.

The following subsections provide descriptions of each stage of the failure detection structure.

3.1.1 Residual Generation

For a dynamic system, there are a number of different techniques for generating residuals. Those techniques can be classified into the following categories :

1. Parity space approach.

2. Observer based approach :

   - diagnostic observer;
   - innovation-based filter.

3. Parameter identification approach.

The Parity space method was described in the Chapter 2.

The Diagnostic observer approach actually involves a number of different but closely connected methods based on the idea of predicting the system's outputs from the measurements using a state observer. The failure detection filter and the Luenberger based observers belong to this class.

The Innovation-based filter or Kalman filter generates zero mean Gaussian innovations with known covariance when no failures are present.

The Parameter identification approach makes use of the fact that the failures in a dynamic system are reflected in the physical parameters, e.g., friction, mass, viscocity, etc.. The idea of the parameter identification approach is to detect the faults via estimation of the parameters of the mathematical model of the system [17]. This approach is not pursued further in this thesis.

A model will now be developed for generating residuals. For a linear discrete-time system for which there are no measurement errors, the following model is appropriate.

$$x_{t+1} = A_t x_t + B_t u_t \tag{3.1}$$

$$y_t = H_t x_t, \tag{3.2}$$

where the observer of Equation (3.1), is described by

$$\hat{x}_{t+1} = A_t \hat{x}_t + K_t[y_t - H_t \hat{x}_t] + B_t u_t \tag{3.3}$$

$$r_t = y_t - H_t \hat{x}_t. \tag{3.4}$$

The purpose of the observer method is to find K_t such that the error of state-estimation is small. This is not the case for failure detection purposes. The gain matrix K_t, for failure detection purposes, is chosen such that the effects of certain failures are accentuated in the filter residual.

To illustrate how the failure detection filter works, we consider the following system in the presence of failure

$$x_{t+1} = A_x x_t + B_t u_t + b_i \nu \delta_{t-\tau^*}, \tag{3.5}$$

where b_i is the $i-$th column of the matrix B and characterizes the effect of a failure in the actuator-i, ν is the magnitude of the failure, and τ^* is the time of failure. The residual

dynamics are described by the following difference equation :

$$r_{t+1} = [A_t - K_t H_t] r_t + b_i \nu \delta_{t-\tau^*} \tag{3.6}$$

This shows that r_t qualifies as a residual since it is an output from a system solely driven by failures. Consider the problem of selecting K_t for the purpose of enhancing failure detection. Let

$$K_t = \sigma I^{n \times n} + A_t \tag{3.7}$$

be the gain for the failure detection filter. The dynamic of the residual described by equation (3.6) becomes

$$r_{t+1} = -\sigma I^{n \times n} r_t + b_i \nu \delta_{t-\tau^*} \tag{3.8}$$

$$r_t = r_{\tau^*} \sigma^t + \frac{b_i \left[1 - (-\sigma)^t\right] \nu}{\sigma} \tag{3.9}$$

If we choose σ carefully, as the effect of the initial condition vanishes, r_t maintains in a fixed direction given by b_i with magnitude proportional to the failure size ν.

An observer that works well in the presence of noise is the Kalman filter [8], [6], [19], [28]. This filter is developed to provide the best estimate of the state vector to minimize variance. The Kalman filter generates a special residual vector, i.e., innovation sequence. It is special since the sequence is uncorrelated in time. Why it is important to use uncorrelated variables will be clear in the subsequent section.

The Kalman filter for discrete-time linear system :

$$x_{t+1} = A_t x_t + B_t u_t + w_t + \delta_{t-\tau^*} \nu \tag{3.10}$$

$$y_t = H_t x_t + e_t \tag{3.11}$$

assuming Gaussian statistics for the noise inputs :

$$w_t \in N(0, Q_t)$$

$$e_t \in N(0, R_t)$$

$$x_0 \in N(0, \Pi_0)$$

is given by the following algorithm :

$$\begin{aligned}
\hat{x}_{1|0}(\tau^*) &= x_0 \\
P_{1|0}(\tau^*) &= P_0 \\
\varepsilon_t(\tau^*) &= y_t - H_t\hat{x}_{t|t-1}(\tau^*) \\
S_t(\tau^*) &= H_t P_{t|t-1}(\tau^*)H_t^T + R_t \\
K_t(\tau^*) &= P_{t|t-1}(\tau^*)H_t^T S_t^{-1}(\tau^*) \\
\hat{x}_{t+1|t}(\tau^*) &= A_t\hat{x}_{t|t-1}(\tau^*) + A_t K_t(\tau^*)\varepsilon_t(\tau^*) \\
P_{t+1|t}(\tau^*) &= A_t\left(P_{t|t-1}(\tau^*) - K_t(\tau^*)H_t P_{t|t-1}(\tau^*)\right)A_t^T + Q_t + \delta(t-\tau^*)P_\nu
\end{aligned} \tag{3.12}$$

The observations are denoted by y_t, the input is u_t, and x_t is the state. The innovation sequence of the Kalman filter is denoted by ε_t. The state jump ν with covariance P_ν occurs at the unknown time instant τ^*, and δ_j is the impulse function that is one if $j = 0$ and zero otherwise.

3.1.2 Statistical Decision Function

Testing between two hypotheses H_0 and H_1 on the residual is the method used to determine the presence of a failure. In [46], it is shown how hypothesis testing can be expressed as a *likelihood ratio test.* The logarithm of the likelihood ratio, defined by

$$s_y = \ln\frac{p_{\theta_1}(y)}{p_{\theta_0}(y)}, \tag{3.13}$$

becomes the main tool for failure detection. The variable s_y in Equation (3.13) is called the *log-likelihood ratio or LLR.* The following example shows how LLR can be applied to identify the presence of a failure.

Example of a LLR test for detecting a change in the mean of the observation. Let us suppose that the output of a typical sensor has a Gaussian distribution with mean value of μ_0 and variance σ^2. Assume that failures can cause shifts in the mean and that we have *a priori* knowledge of the mean shift μ_1 that constitutes a failure. To identify the

presence of failure, we set the following hypotheses :

$$\begin{aligned} H_0 &: \mu = \mu_0 \\ H_1 &: \mu = \mu_1 \end{aligned}$$

The probability density corresponding to each hypothesis is

$$\begin{aligned} p_{\theta_0}(y) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(y-\mu_0)^2}{2\sigma^2}\right) \\ p_{\theta_1}(y) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(y-\mu_1)^2}{2\sigma^2}\right) \end{aligned}$$

For each observation y_i we define the LLR

$$\begin{aligned} s_i &= \ln \frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)} \\ &= \frac{\mu_1 - \mu_0}{\sigma^2}\left(y_i - \frac{\mu_1 + \mu_0}{2}\right) \end{aligned}$$

The decision function is

$$\begin{aligned} S_1^N &= \sum_{i=1}^{N} s_i \\ &= \frac{\mu_1 - \mu_0}{\sigma^2} \sum_{i=1}^{N}\left(y_i - \frac{\mu_1 + \mu_0}{2}\right) \end{aligned}$$

The stopping rule for the failure detection algorithm is

$$t_a = N \cdot min\{K : d_K = 1\}$$

where

$$\begin{aligned} d &= \begin{cases} 0 & \text{if } S_1^N(K) < h \\ 1 & \text{if } S_1^N(K) \geq h \end{cases} \\ S_1^N &= S_{N(K-1)+1}^{NK} \end{aligned}$$

It is described in [4] that this change detection algorithm is one of the oldest and most well-known algorithm for continuous inspection, and is called *Shewhart control chart.*

There are several new variables that appear in the previous example. These will be referred to in the subsequent sections and chapters. These quantities are defined as :

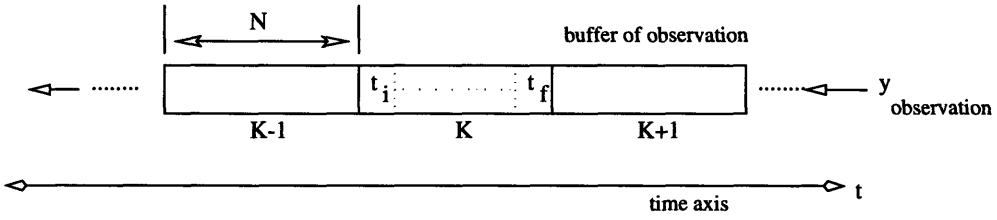

Figure 3-2: Determining the decision function from observation

- Cumulative sum :

$$S_i^k = \sum_{i=j}^{k} s_i \tag{3.14}$$

- Decision function : g_k.

  For problem with Gaussian distribution

$$g_k = S_1^N(K) \tag{3.15}$$

  where

$$S_1^N = S_{N(K-1)+1}^{NK} \tag{3.16}$$

- Decision rule :

$$d = \begin{cases} 0 & \text{if } g_k < h;\ H_0 \text{ is chosen} \\ 1 & \text{if } g_k \geq h;\ H_1 \text{ is chosen} \end{cases} \tag{3.17}$$

  For problem with Gaussian distribution

$$d = \begin{cases} 0 & \text{if } S_1^N(K) < h;\ H_0 \text{ is chosen} \\ 1 & \text{if } S_1^N(K) \geq h;\ H_1 \text{ is chosen} \end{cases} \tag{3.18}$$

- Stopping rule :

$$t_a = N \cdot min\{K : d_K = 1\} \tag{3.19}$$

Figure 3-2 illustrates how the observation is conducted and the indices that correspond to each definition. The observation is obtained sequentially in time, and is put inside the buffer of observation. The decision function is determined from the processing of data with fixed-length N. The processing for each step K is initiated from time t_i and ended at time t_f. The observation is stopped after the fist sample of size N for which the decision is in favor of H_1. The choice of threshold h depends on the *miss-alarm* and *false-alarm* probability selection.

In many applications, it is common that μ_1 is not known *a priori.* In [4], it is explained that there are two possible solutions in this case. The likelihood ratio for known θ_1 is given by

$$\Lambda_n = \frac{p_{\theta_1}(y_1, \ldots, y_n)}{p_{\theta_0}(y_1, \ldots, y_n)}. \tag{3.20}$$

For unknown θ_1, Λ_n is replaced by another statistic. The first one is based upon the weighted likelihood ratio :

$$\tilde{\Lambda}_n = \int_{-\infty}^{+\infty} \frac{p_{\theta_1}(y_1, \ldots, y_n)}{p_{\theta_0}(y_1, \ldots, y_n)} dF(\theta_1). \tag{3.21}$$

In this approach, the likelihood ratio is weighted by a weighting function $dF(\theta_1)$ with respect to all possible values of the parameter θ_1. $F(\theta_1)$ may be interpreted as the cumulative distribution function of a probability measure. The failure detection algorithm resulting from this approach is known as the χ^2-*cumulative sum.*

The χ^2-*cumulative sum failure detection algorithm* is given as follows

$$\tilde{\Lambda}_j^k = \cosh\left[b\,(k-j+1)\,\chi_j^k\right] \exp\left(-\frac{b^2}{2}(k-j+1)\right) \tag{3.22}$$

$$\chi_j^k = \frac{1}{k-j+1}|\tilde{S}_j^k| \tag{3.23}$$

$$\tilde{S}_j^k = \frac{1}{\sigma}\sum_{i=j}^{k}(y_i - \mu_0) \tag{3.24}$$

$$b = \frac{\nu}{\sigma} \tag{3.25}$$

This algorithm is developed based on a problem of detecting a change in mean of Gaussian

sequence with known variance σ^2. The derivation here is special, the distribution $F(\theta) = F(\mu)$ is concentrated on two points, i.e. $\mu_0 - \nu$ and $\mu_0 + \nu$. For real application, this situation is reasonable, the output of sensor can be shifted by either positive or negative bias. The weighted likelihood ratio of Equation (3.21), then can be expressed as

$$\tilde{\Lambda}_j^k = \int_{-\infty}^{+\infty} \exp\left[b\tilde{S}_j^k - \frac{b^2}{2}(k-j+1)\right] dF(\nu) \tag{3.26}$$

The stopping time in the χ^2-cumulative sum failure detection algorithm is thus

$$t_a = \min k : g_k \geq h \tag{3.27}$$

$$g_k = \max_{1\leq j\leq k}\left[\ln\cosh(b\tilde{S}_j^k) - \frac{b^2}{2}(k-j+1)\right] \tag{3.28}$$

The second approach to determine the likelihood ratio when θ_1 is unknown uses the *generalized likelihood ratio - GLR*

$$\tilde{\Lambda}_n = \frac{\sup_{\theta_1} p_{\theta_1}(y_1,\ldots,y_n)}{p_{\theta_0}(y_1,\ldots,y_n)}. \tag{3.29}$$

In this approach, the unknown parameter θ_1 is replaced by its maximum likelihood estimate. The decision function in GLR is given by

$$g_k = \max_{1\leq j\leq k} \ln\tilde{\Lambda}_j^k = \max_{1\leq j\leq k} \sup_{\theta_1} S_j^k(\theta_1) \tag{3.30}$$

The following example shows how GLR is used to provide the decision function g_k for detecting changes in the mean from independent Gaussian sequence. From the previous example of detecting change from sensor output, we have

$$S_j^k = \frac{\mu_1 - \mu_0}{\sigma^2}\sum_{i=j}^{k}\left(y_i - \frac{\mu_1+\mu_0}{2}\right) \tag{3.31}$$

Define

$$\nu = \mu_1 - \mu_0 \tag{3.32}$$

then

$$g_k = \max_{1\leq j\leq k}\sum_{i=j}^{k}\left[\frac{\hat{\nu}(y_i-\mu_0)}{\sigma^2} - \frac{\hat{\nu}^2}{2\sigma^2}\right] \tag{3.33}$$

where

$$|\hat{\nu}_j| = \left(\frac{1}{k-j+1} \sum_{i=j}^{k} |y_i - \mu_0| - \nu_{\min} \right) + \nu_{\min} \tag{3.34}$$

The next sections describe the development of these decision functions for failure detection in linear dynamic systems. It is interesting to note in the next section how the output of linear system is fit into these statistical decision functions.

3.2 The GLR Test for Failure Detection in Linear Dynamic System

The GLR test used for failure detection in linear dynamic system was proposed by Jones and Willsky in [51]. The hypotheses used for the purpose of failure detection in linear dynamic systems are given by Equations (2.16) - (2.19). For observations over a finite interval $t_i \leq \tau^* \leq t_f$, the likelihood ratio of the hypothesis test is given by :

$$\Lambda_{t_f}(\tau^*, \nu) = \frac{p(y_{t_i}, \ldots, y_{t_f} | H_1, \tau^*, \nu)}{p(y_{t_i}, \ldots, y_{t_f} | H_0)} \tag{3.35}$$

If y_t is the observation sequence from a dynamical system it is difficult to compute Equation (3.35), because the observation sequence is not independent [1]. It will reduce the complexity of the computation if the innovation sequence is used to determine the likelihood ratio instead of the observation sequence. The innovation sequence is a white process, i.e. it is an uncorrelated sequence. This property enables the computation of the joint probability density of the white sequence as

$$p(\varepsilon_{t_i}, \ldots, \varepsilon_{t_f}) = p(\varepsilon_{t_i}) \ldots p(\varepsilon_{t_f}) = \prod_{t_i}^{t_f} p(\varepsilon_t) \tag{3.36}$$

Under each hypotheses of Equations (2.16) - (2.19), the new hypotheses can be defined as

$$H_0 \quad : \quad \varepsilon_t \varepsilon_t^0 \tag{3.37}$$

$$H_1 \quad : \quad \varepsilon_t(\tau^*)\varepsilon_t^0 + \varphi_t^T(\tau^*)\nu \tag{3.38}$$

where $\varphi_t^T(\tau^*)$ is the failure signature, which can be recursively computed, ε_t^0 is the innovation sequence in the absence of failure. Recall that the innovation sequence is obtained

from the Kalman filter.

The likelihood ratio for hypothesis test now becomes

$$\begin{aligned}\Lambda_{t_f}(\tau^*,\nu) &= \frac{p(\varepsilon_{t_i},\ldots,\varepsilon_{t_f}|H_1,\tau^*,\nu)}{p(\varepsilon_{t_i},\ldots,\varepsilon_{t_f}|H_0)} \\ &= \prod_{k=t_i}^{k=t_f}\frac{p(\varepsilon_k|H_1,\tau^*,\nu)}{p(\varepsilon_k|H_0)} \end{aligned} \tag{3.39}$$

The last equality follows from the independence of the innovation process. The LLR from Equation (3.35), with the Gaussian assumption, is given by

$$\begin{aligned} l_t(\tau^*,\nu) &= \ln\Lambda_{t_f}(\tau^*,\nu) \\ &= \nu f_{t_i}^{t_f}(\tau^*) - \frac{1}{2}\nu^2\tilde{R}_{t_i}^{t_f}(\tau^*)^2 \end{aligned} \tag{3.40}$$

where

$$f_{t_i}^{t_f}(\tau^*) = \sum_{t=t_i}^{t=t_f}\varphi_t(\tau^*)S_t^{-1}\varepsilon_t(\tau^*) \tag{3.41}$$

$$\tilde{R}_{t_i}^{t_f}(\tau^*) = \sum_{t=t_i}^{t=t_f}\varphi_t(\tau^*)S_t^{-1}\varphi_t^T(\tau^*) \tag{3.42}$$

How Equation (3.35) is derived to result Equations (3.41) - (3.42) is given in the Appendix B.4. The quantity S_t is obtained from the Kalman filter. Note that l_t is used to describe the decision function instead of g_t. The intention here is to distinguish the decision function for linear dynamic system l_k from the general decision function g_k.

The GLR is given by

$$l_t = \max_{\tau^*,\nu} l_t(\tau^*,\nu) \tag{3.43}$$

As described in the previous section, the magnitude of failure the ν is replaced by its maximum likelihood estimate. This maximum likelihood estimate of ν is obtained by taking the derivative of Equation (3.42), and set that derivation equal to zero. Hence,

$$\hat{\nu}(\tau^*) = \left(\tilde{R}_{t_i}^{t_f}(\tau^*)\right)^{-1} f_{t_i}^{t_f}(\tau^*) \tag{3.44}$$

Substituting the Equation (3.44) into Equation (3.42) gives

$$l_t = \left(f_{t_i}^{t_f}(\tau^*)\right)^T \left(\tilde{R}_{t_i}^{t_f}(\tau^*)\right)^{-1} f_{t_i}^{t_f}(\tau^*) \tag{3.45}$$

Another interpretation of Equation (3.45) is given by the following equation

$$l_t = \hat{\nu}(\tau^*)^T \tilde{R}_{t_i}^{t_f} \hat{\nu}(\tau^*) \tag{3.46}$$

The last equation shows that essentially $\tilde{R}^{-1}$ is the error variance of the estimate $\hat{\nu}_t$. The corresponding hypotheses with respect to Equation (3.46) are

$$H_0 \quad : \quad \nu = 0 \tag{3.47}$$

$$H_1 \quad : \quad \nu \neq 0 \tag{3.48}$$

A failure is declared if l_t is larger than a threshold h. The test is computed for each possible τ^* and the estimated τ^* is the one giving the maximum likelihood of l_t :

$$\hat{\tau}^* = \arg\max_t l_t\left(\tau^*, \hat{\nu}_t(\tau^*)\right) \tag{3.49}$$

To implement the GLR test for failure detection in linear dynamic system, the following algorithm can be used [22], [21]:

1. Calculate the innovation sequence ε_t under H_0 hypothesis.

2. Determine the quantities $\varphi_t(\tau^*)$, $f_{t_i}^{t_f}(\tau^*)$, and $\tilde{R}_{t_i}^{t_f}(\tau^*)$ for each $(\tau^*) : t_i \leq (\tau^*) \leq t_f$.

3. Compute the estimate of $\nu_t(\tau^*)$, i.e., $\hat{\nu}_t(\tau^*)$ recursively.

4. Compute the estimate of LLR

$$l_t = \left(f_{t_i}^{t_f}(\tau^*)\right)^T \left(\tilde{R}_{t_i}^{t_f}(\tau^*)\right)^{-1} f_{t_i}^{t_f}(\tau^*)$$

this quantity is compared against the threshold h.

5. The possible jump time is given by

$$\hat{\tau}^* = \arg \max_{\tau^*} l_t(\tau^*, \hat{\nu}_t(\tau^*))$$

Real-time or on-line application of the GLR requires the recursive form of Equation (3.44). In Appendix B.3 and B.4, the recursive expression form for Equation (3.44) is given.

3.3 The MLR Test for Failure Detection

The MLR - *Marginalized Likelihood Ratio* test treats the parameter ν in a different way. Instead of taking the maximum-likelihood estimate for ν, the MLR test treats ν as a *random variable.*

From probability theory, we know that for random variables A and B, the following equality is true :

$$p(A) = \int_{-\infty}^{+\infty} p(A|B)p(B) \quad dB \tag{3.50}$$

The technique given in Equation (3.50) is called marginalization. Integrating the conditional probability $p(A|B)$ multiplied by $p(B)$ with respect to random variable B will give the distribution function $p(A)$ which is independent of the random variable B. This is the technique used by MLR to eliminate the nuisance parameter ν.

Marginalized Likelihood Ratio (MLR) test is somewhat related to the GLR test. The MLR test for failure detection is claimed in [22] to have several advantages compared to GLR. It is stated in [22] that the MLR test has the following features

- There is no sensitive threshold to choose in MLR.
- The MLR test is robust to modeling error of noise variances.
- The MLR requires lower level of computational complexity for off-line application.

The hypotheses used for failure detection in the MLR test are the same as those of the GLR test. Even the LLR of both algorithm expressions are the same. The LLR of both the MLR test and the GLR test is given as

$$l_1^N(\tau^*, \nu) = l_N = 2 \ln \frac{p(y^N|H_1(\tau^*, \nu))}{p(y^N|H_0)} \tag{3.51}$$

for the observation $y^N = \{y_1, y_2, \ldots, x_N\}$. In the GLR test, ν is replaced by the maximum likelihood estimate of ν, $\nu(\tau^*)$, while in the MLR test, ν is considered as a random variable. Marginalization of the probability density function from the numerator given in (3.51) over ν makes the probability density independent of ν as given in the following equation :

$$p(\varepsilon_1^N) = \int_{-\infty}^{+\infty} p(\varepsilon_1^N|\nu)p(\nu)d\nu. \tag{3.52}$$

The LLR of the MLR test in Equation (3.51) shows that the numerator is the conditional probability of the observations. After marginalizing the conditional probability of the numerator, the following LLR will be obtained :

$$l_1^N(\tau^*, \nu) = 2\ln\frac{p(y^N|H_1(\tau^*))}{p(y^N|H_0)} \tag{3.53}$$

There are two ways for assigning the prior probability $p(\nu)$ [22], [23]. First, $p(\nu)$ is assumed to have a Gaussian distribution. Second, $p(\nu)$ is assumed to have infinite invariance, in which the $p(\nu)$ is given a constant value. The second approach is considered as a non-informative problem.

Developing the Equation (3.53) will give us the following results

$$\begin{aligned} l_N(\tau^*) &= \arg\max_{\nu} 2\ln\frac{p(y^N|\tau^*, \nu)}{p(y^N)} - \ln|\tilde{R}_N(\tau^*)| + C_{prior}(\tau^*) \\ &= l_N(\tau^*, \hat{\nu}(\tau^*)) - \ln|\tilde{R}_N(\tau^*)| + C_{prior}(\tau^*) \end{aligned} \tag{3.54}$$

The quantities from the GLR test are used in Equation (3.54). Those quantities are given in Equations (3.45) , (3.46).

When we assume that the prior probability density of ν is a constant, we will have $C_{prior}(\tau^*)$ given by

$$C_{prior}(\tau^*) = 0. \tag{3.55}$$

If we choose the Gaussian distribution as the density of ν such that

$$\nu \quad \in \quad N(\nu_0, P_\nu)$$

then the value of $C_{prior}(\tau^*)$ will be given by

$$C_{prior}(\tau^*) = -\log |P_\nu| + (\hat{\nu}_N(\tau^*) - \nu_0)^T P_\nu^{-1} (\hat{\nu}_N(\tau^*) - \nu_0) \tag{3.56}$$

Appendix B.5 will provide the derivation of the MLR test. The process of marginalization from conditional probability for each prior probability density for ν is shown therein.

3.4 Two-Filter Implementation of The MLR Test

The GLR test requires a lot of computing resource for exploring the maximation process. The MLR test, however, can be arranged such that it is computationally efficient. Especially for off-line application, the MLR test only requires two filters, [20], [22], [21]. One filter executes forward processing in time, while the other executes backward processing in time. Under certain circumstances, by allowing a certain amount of delay, the two-filter MLR test can be applied on-line.

The following derivation of the MLR test will show how this algorithm can be implemented by two filters only. The derivation in the sequel is different from what we saw in the previous section.

From Bayes' rule we have

$$\begin{aligned} p(A,B) &= p(A|B)p(B) \\ &= p(B|A)p(A) \end{aligned}$$

Based on that rule, the sequence of $y_m^n = \{y_{m+1}, y_{m+2}, \ldots, y_{n-1}, y_n\}$ will have the probability expression as either

$$\begin{aligned} p(y_m^n) &= p(y_n|y_m, y_{m+1}, \ldots, y_{n-2}, y_{n-1})p(y_m, y_{m+1}, \ldots, y_{n-2}, y_{n-1}) \\ &= p(y_n|y_m^{n-1})p(y_m^{n-1}) \\ &= \prod_{t=m}^{n} p(y_t|y_m^{t-1}) \end{aligned} \tag{3.57}$$

or

$$p(y_m^n) = p(y_m|y_{m+1}, y_{m+2}, \ldots, y_{n-1}, y_n)p(y_{m+1}, y_{m+2}, \ldots, y_{n-1}, y_n)$$

$$
\begin{aligned}
&= p(y_m|y_{m+1}^n)p(y_{m+1}^n) \\
&= \prod_{t=m}^{n} p(y_t|y_{t+1}^n)
\end{aligned} \tag{3.58}
$$

Reversing the order of the sequence does not change the probability $p(y_m^n)$. The probability of measurements under the hypotheses that jump occurs at time τ^* is

$$
\begin{aligned}
p(y^N|\tau^*) &= p(y_1^k|\tau^*).p(y_{\tau^*+1}^N|\tau^*, y^{\tau^*}) \\
&= p(y_1^{\tau^*})p(y_{\tau^*+1}^N)
\end{aligned} \tag{3.59}
$$

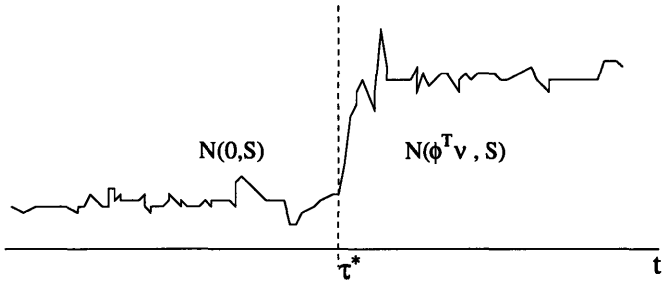

Figure 3-3: The density functions of observation before and after the jump

The conditional probability factors in Equation (3.59) become the marginal probability factors. Figure 3.4 illustrates the probabilities involved for the event of failure. Before the jump occur at time τ^*, the measurements are not affected by the presence of the jump. Hence, $p(y_1^{\tau^*}|\tau^*) = p(y_1^{\tau^*})$. This factor is computed by

$$
p(y_1^{\tau^*}) = \prod_{t=1}^{\tau^*} \gamma\left(y_t - H_t^F \hat{x}_{t|t-1}^F, H_t^{FT} P_{t|t-1}^F H_t^F + R^F + t\right) \tag{3.60}
$$

as it is shown in Appendix B.1. Note that $\gamma(\mu, R)$ is a Gaussian distribution function with mean value of μ and covariance R. The computation is done by applying Kalman filter on the forwards Markovian model.

The factor $p(y_{\tau^*+1}^N|\tau^*, y^{\tau^*}) = p(y_{\tau^*+1}^N)$ is computed from the reversed-time point of view. This term is expanded backwards in time in the same way as the forward computation case. Note that from the backward in time point of view, the probability distribution of measurement before jump is $\gamma(\varphi_t^T(\tau^*), S)$, with $\varphi_t^T(\tau^*)$ mean value and S covariance. The probability density $\gamma(\varphi_t^T(\tau^*), S)$ is computed by

$$p(y_{\tau^*+1}^N) = \prod_{t=\tau^*+1}^{N} \gamma\left(y_t - H_t^B \hat{x}_{t|t-1}^B, H_t^{BT} P_{t|t+1}^B H_t^B + R^B + t\right) \tag{3.61}$$

The Kalman filter applied in backward model will give $\hat{x}_{t|t+1}^B$ and $P_{t|t+1}^B$.

The Kalman filter running backward applies the same algorithm given in the Equation (3.12). Modification, however, should be applied for the dynamic matrix and the covariance matrix, [47]. Appendix B.3 provides a description of how to modify the dynamic matrix of the state space and the covariance matrix from the given state-space dynamic system, running forward in time.

3.5 Numerical Examples

We consider now a numerical example to illustrate how to apply the failure detection algorithms : GLR - MLR - 2-filter MLR to a discrete-time linear dynamic system. The simplified model of the inertial system heading gyro error,

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.62}$$

$$Y = y \tag{3.63}$$

$$X_{t+1} = \begin{bmatrix} 1 & \varsigma \\ 0 & 1 - \frac{\varsigma}{\tau_g} \end{bmatrix} X_t + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} w_t \tag{3.64}$$

$$Y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} X_t + e_t \tag{3.65}$$

will be used with the following numerical values $\varsigma = 1$, $\tau_g \to \infty$, $b_1 = 0.5$ and $b_2 = 1$. This model is taken from [36]. For information, ς is the sampling period, and τ_g is the gyro error time constant. The initial state X_0, the disturbance noise w_t and the measurement noise e_t have the following covariance matrices :

$$cov[X_0, X_0] = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{3.66}$$

$$cov[w_t, w_t] = \begin{bmatrix} 1 \end{bmatrix} \tag{3.67}$$

$$cov[e_t, e_t] = \begin{bmatrix} 1 \end{bmatrix} \tag{3.68}$$

In the first simulation, the failure of impulse type is injected at time $t = 25$. The system dynamic simulation is run from $t = 0$ - $t = 50$. The residual generated from the Kalman

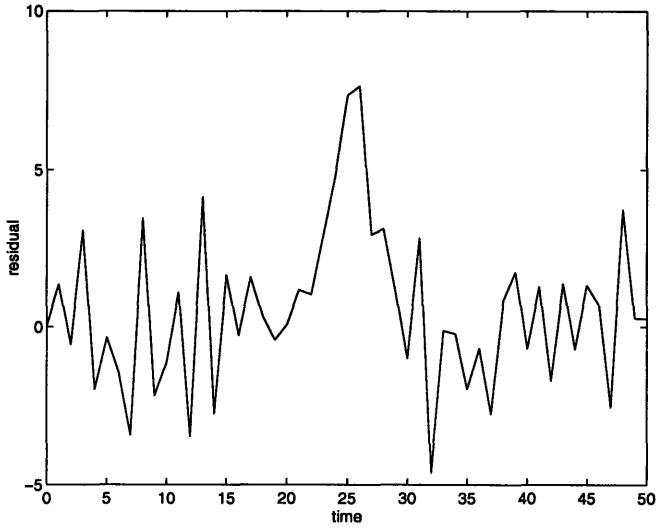

Figure 3-4: The residual from the Kalman filter

filter is presented in Figure 3-4. We see in Figure 3-4 that approximately at the time when failure occurs, the residual has the highest magnitude. The residual is eventually processed to give the decision function presented in Figure 3-5. At time $t = 25$, the time when failure

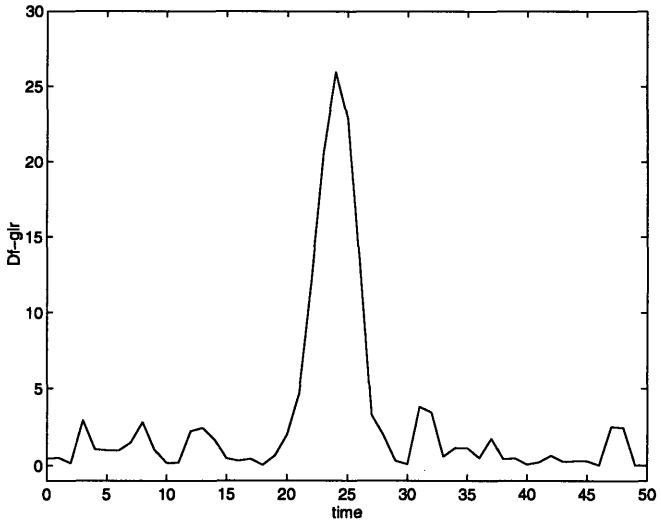

Figure 3-5: The decision function from the GLR test

occurs, the decision function from the GLR test gives the maximum value.

The MLR test relies on the result from the GLR. The MLR test's decision function is a shifted version of the GLR test's result in magnitude. As we can see in Figure 3-6, for the Gaussian prior, it has the same form as that of Figure 3-5, however, the magnitude of the decision function envelope is amplified drastically at the time of impulse.

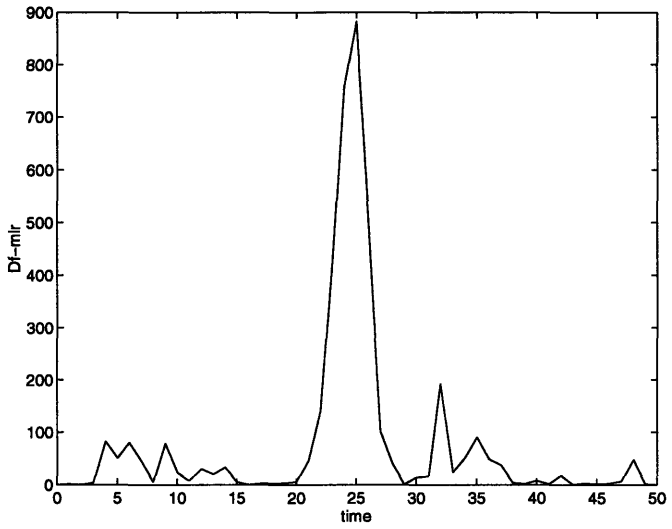

Figure 3-6: The decision function from the MLR test - Gaussian prior

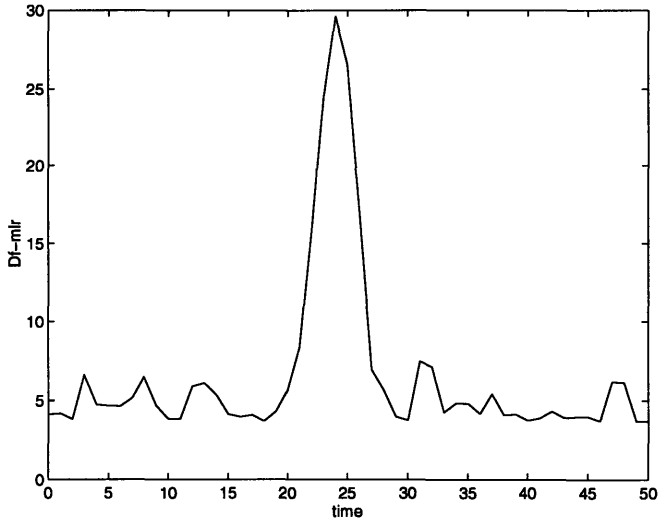

Figure 3-7: The decision function from the MLR test - constant prior

If the probability density of the failure is assumed constant, the decision function of the MLR test will look like that of Figure 3-7. Still, amplification of the envelope from the

original GLR test underlies the MLR test.

The decision function given by the MLR test with two-filter implementation is slightly different from the previous tests. The magnitude of the decision function goes below zero and at the time when the failure occurs, it gives the impulse-type decision as is given in Figure 3-8

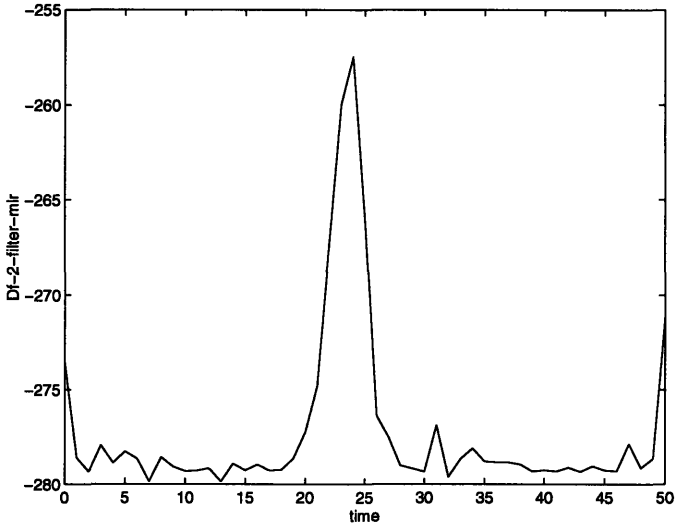

Figure 3-8: The decision function from the 2-filter MLR test

The behavior of failure detection algorithms with respect of the location of the failure inside the buffer of observation is also presented here. In this simulation, failure time is varied from the beginning of the buffer or window of observation, i.e. $t = 0$, to the end of the buffer, i.e. $t = 50$. A Monte-Carlo method was used for this simulation. Only two algorithms are compared here, excluding the MLR test. The GLR test and the 2-filter MLR test provide a significant different from each other, while the MLR test is basically the GLR with amplification. The results of this study are presented in Figures 3-9 and 3-10.

Figure 3-9 shows that the GLR test provide good performance in tracking the failure time. Even when the time failure is at the beginning of the buffer or at the end of the buffer, the GLR test still can identify the failures. The decision function space of the GLR test excluding the region of failure stays approximately at the same value as we can see in Figure 3-9.

The 2-filter MLR test, however, requires different threshold for each different failure time. Additional filtering is required for identification of failure time in the 2-filter MLR

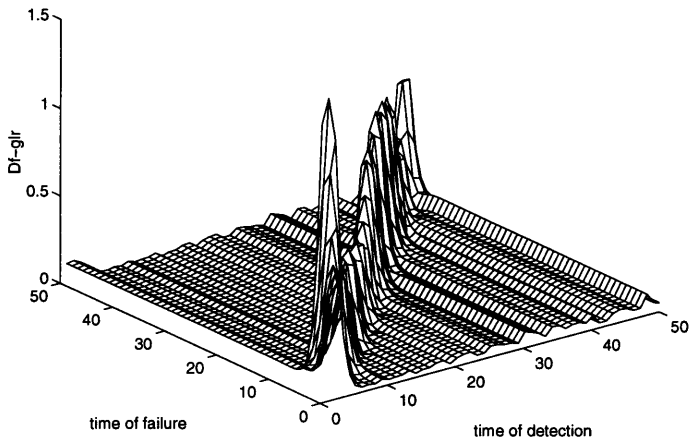

Figure 3-9: The decision function from the GLR test-various failure time

test since as the time of failure varies, the mean of the detection function as well as the magnitude of the peak of the detection function within the observation interval change. As we can see in Figure 3-11, the contour of the decision function is very noisy due to the variation of the mean of detection space as the time of failure changes. For each failure time k, as we normalize the decision function through the following equation :

$$\bar{l}_t(k,\hat{\nu}(k)) = l_t(k,\hat{\nu}(k)) + \inf_{t_i \le t \le t_f} l_t(k,\hat{\nu}(k)), \tag{3.69}$$

we will obtain the decision function space given in Figure 3-13 with its contour given in Figure 3-12. The normalized detection space is now smoother. This indicates that the decision function does not require adaptive threshold. We can see in Figure 3-12 that the common threshold facilitates the detection function to identify the location of failure whatever the location of failure within the window of observation.

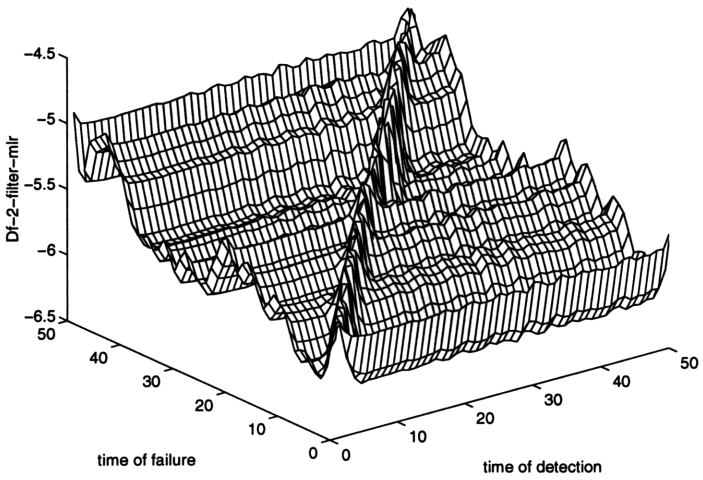

Figure 3-10: The decision function from the 2-filter MLR-various failure time

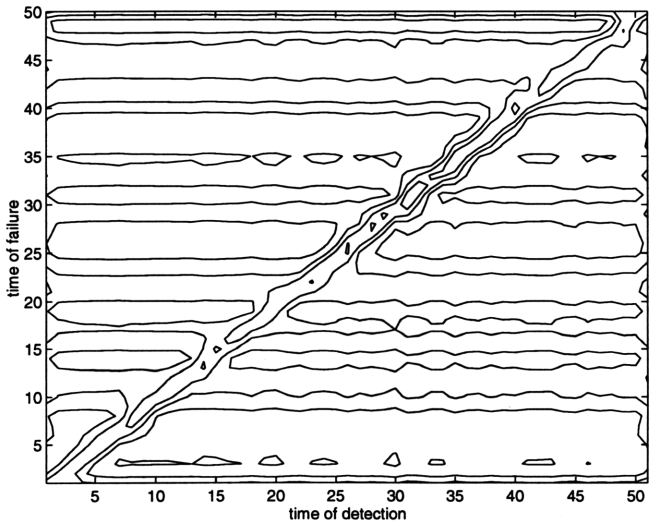

Figure 3-11: The contour decision function from the 2-filter MLR test

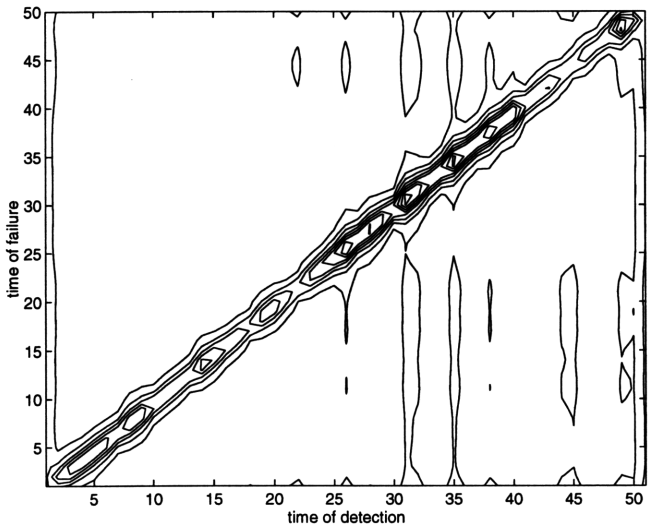

Figure 3-12: The contour of function from the normalized 2-filter MLR test

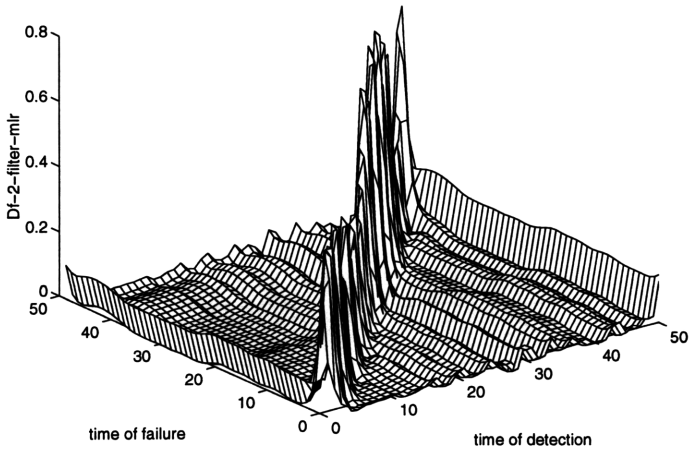

Figure 3-13: The normalized decision function from the 2-filter-MLR test for various failure time

Chapter 4

Deterministic Failure Detection Algorithm

The failure detection algorithms described in this chapter are developed based on a different approach from what was given in Chapter 3. These algorithms are derived deterministically, meaning that no probability model is assigned to characterize the unknown or undermined quantities. The deterministic algorithm for failure detection discussed in this chapter is developed in such a way that it still functions under the worst possible combination of unknowns in both the system model and the process noise model, on the assumption that the unknowns are within strict bounds. This algorithm is a result of the research work by Mangoubi [31].

In Section 4.1, the problem of generating robust residuals is presented. A number of estimators that have been developed recently provide an estimate of the system dynamic states under the worst possible condition of unknown or uncertain quantities. One of those filters resulting from the research of Appleby [2], is discussed in section 4.2. This section explores how this filter works to generate residuals through a numerical example. The main discussion of this chapter is the deterministic failure detection algorithm, which is discussed thoroughly in section 4.3. At the end of this chapter, numerical examples demonstrate the application of this deterministic algorithm.

4.1 Robust Residuals

The role of residuals in failure detection was explored in Chapter 3. Using the measurement residual ε_t for hypothesis testing rather than the measurement y_t has the following advantages :

- ε_t contains the same statistical information as y_t, relative to failure detection,
- ε_t is not correlated in time, while y_t is,
- using ε_t reduces the complexity of LLR computation,

The residual ε_t as defined in Chapter 3 is generated by a Kalman filter. In the research work of Appleby, [2], [3], it is shown that the Kalman filter performance degrades in generating residuals when there is uncertainty in system modeling and process noise modeling.

Several methods have been developed to provide estimators that are robust in the presence of model uncertainties, as described in [2], [3] and the references therein. Performance comparison of these estimators to the Kalman filter also can be found in the research works of Appleby [2], [3], and Mangoubi [32]. For the system :

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t \\ y_t &= Cx_t + Du_t + e_t \end{aligned}$$

the robust estimator will generate the estimated x_t, $\hat{x}_t$ which will have the following dynamics:

$$\begin{aligned} \hat{x}_{t+1} &= A\hat{x}_t + K\left(y_t - C\hat{x}_t\right) + Bu_t \\ r_t &= y_t - C\hat{x}_t. \end{aligned}$$

The quantity r_t is denoted as the robust residual, since it is generated by the robust estimator.

The uncertainties in the system model and the process noise model will introduce errors in estimation. These errors will be reflected in residuals. The robust estimators are designed so as to minimize the error due to the uncertainties in both the system model and the process noise model. For failure detection purpose, the residuals carry the following information :

- errors in estimation,
- failure information.

It is expected that the use of the robust estimator will give residuals that carry failure information only.

An initial study of applying robust residuals for LLR computation is given in the work of Mangoubi [32]. It is shown therein that the performance of robust estimators is better than that of the residuals from the Kalman filter when there are unknown parameters. The next section will describe the idea of using a robust estimator to generate the residuals for failure detection. This section precedes the main discussion of the deterministic failure detection algorithm, RFDI.

4.2 Robust H_∞ Filter for Residual Generation

To design an estimator for a linear dynamic system given by the following state-space equations :

$$\begin{aligned} x_{t+1} &= (A + \Delta A)x_t + (B + \Delta B)u_t + w_t \\ y_t &= (C + \Delta C)x_t + (D + \Delta D)u_t + e_t \end{aligned}$$

we often use the matrices A, B, C, D to estimate the state x_t with the following estimator :

$$\begin{aligned} \hat{x}_{t+1} &= A\hat{x}_t + K\left(y_t - C\hat{x}_t\right) + Bu_t \\ r_t &= y_t - C\hat{x}_t. \end{aligned}$$

The reliability of state estimate $\hat{x}_t$ given by the estimator working on the nominal model, therefore, becomes a question.

The H_∞ filter was developed to provide estimates of plant states under the worst case noise and plant modeling errors. These filters or estimators provide robust guaranteed estimation with respect to those modeling errors.

The residuals can be described by the following equation :

$$r_t = r_t^e + r_t^f$$

where r_t^e corresponds to error in estimation and r_t^f corresponds to the failure information. For failure detection, it is expected that

$$r_t^e \to 0$$
$$r_t \approx r_t^f$$

Robust residual is expected to have the aforementioned properties.

In this section, we explore the use of an H_∞ filter for failure detection. The H_∞ filter used is the one developed by Appleby [2], [3]. The output from the filter is then processed to generate a residual vector. The residual vector is then used in the GLR algorithm given in Chapter 3 for failure detection problems. This approach follows, in part, the framework given in section 3.1.

The availabity of the MATLAB computational software package makes it straightforward to develop an H_∞ filter. We need to formulate the problem in a standard form, and MATLAB will then produce the gains for the robust filter.

For a given dynamic system with the following state-space equations :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \begin{bmatrix} B_1 & 0 \end{bmatrix} \begin{bmatrix} w(t) \\ e(t) \end{bmatrix} \\ z(t) &= Mx(t) \\ y(t) &= Cx(t) + \begin{bmatrix} 0 & B_2 \end{bmatrix} \begin{bmatrix} w(t) \\ e(t) \end{bmatrix} \end{aligned}$$

where

- $x(t)$: system states,
- $w(t)$: process noise,
- $e(t)$: measurement noise,
- $z(t)$: performance vector,
- A : dynamic matrix,
- $B1$: process noise matrix,

- $B2$: measurement noise matrix,
- M : performance matrix,

the estimator will be described by the following equations:

$$\begin{aligned}
\dot{\hat{x}}(t) &= A\hat{x}(t) + K_\infty \left(y(t) - \hat{y}(t)\right) \\
\hat{z}(t) &= M\hat{x}(t) \\
\hat{y}(t) &= C\hat{x}(t)
\end{aligned}$$

To solve the problem with MATLAB, we need to construct the following matrix :

$$P = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right]$$

where

$$\begin{aligned}
\tilde{A} &= A \\
\tilde{B} &= \left[\begin{array}{cc} \left[\begin{array}{cc} B_1 & 0 \end{array}\right] & \underline{0} \end{array}\right] \\
\tilde{C} &= \left[\begin{array}{c} M \\ C \end{array}\right] \\
\tilde{D} &= \left[\begin{array}{cc} & \underline{0} \\ \left[\begin{array}{cc} \underline{0} & B_2 \end{array}\right] & \underline{0} \end{array}\right].
\end{aligned}$$

The MATLAB command **hinfsyn** from the μ-Synthesis Toolbox, will provide the lower bound of γ which is associated with the level of uncertainties, [18]. This γ is immediately used to construct the following Hamiltonian matrix

$$H_{Y_\infty} = \left[\begin{array}{cc} A^T & -C^T(B_2B_2^T)^{-1}C + \frac{1}{\gamma^2}M^TM \\ -B_1B_1^T & -A \end{array}\right].$$

For the Hamiltonian matrix H_{Y_∞} there is an associated Riccati equation of the following form :

$$AY_\infty + Y_\infty A^T + Y_\infty \left[-C^T(B_2B_2^T)^{-1}C + \frac{1}{\gamma^2}M^TM\right] Y_\infty - (-B_1B_1^T) = 0$$

The MATLAB command **ric_schr** can be used to generate the solution of the Riccati equation by inputting the corresponding Hamiltonian matrix structure. The solution of the Riccati equation, Y_∞, determines the gain matrix K_∞ :

$$K_\infty = Y_\infty \cdot C^T \cdot \left(B_2 B_2^T\right)^{-1}$$

The following numerical example shows the performance of Appleby's H_∞ filter in state estimation. The residual vector from the Appleby's H_∞ filter is used for failure detection.

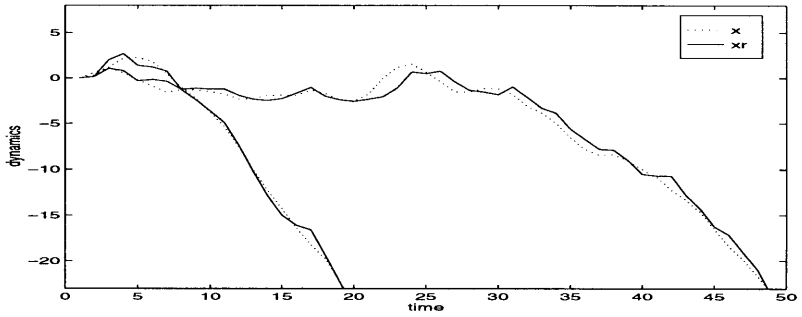

Figure 4-1: System dynamics and its estimates from H_∞ filter

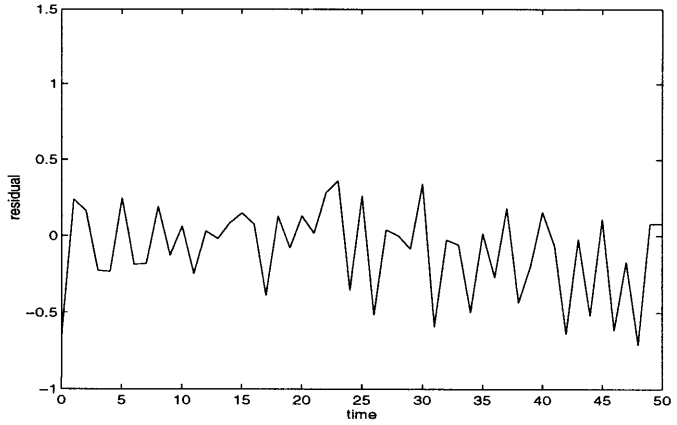

Figure 4-2: The residual from H_∞ filter

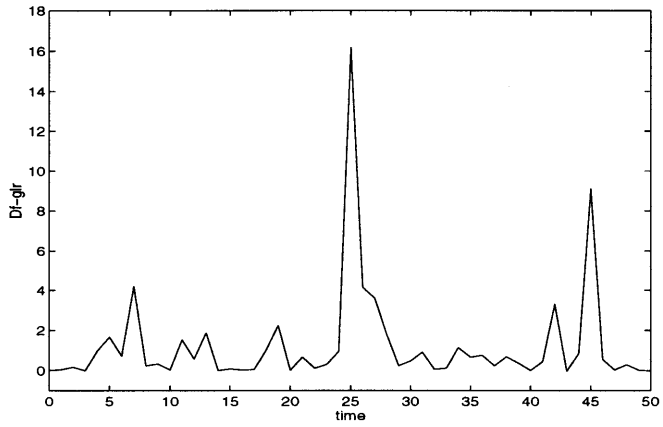

Figure 4-3: The decision function of GLR using robust residual

Figure 4-1 shows a simulated output of the system described in Section 3.5. We use the same system model as the one given in numerical example of Section 3.5. The system in Section 3.5 is discrete-time while the Appleby's H_∞ filter works for continuous time system. Transformation from the discrete-time system to the continuous-time system is performed by *forward-Euler rectangular* method so that the dynamic matrix and the input matrix become

$$A_{cont} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$B_{cont} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Appleby's H_∞ filter is developed based on these nominal matrices A_{cont}, B_{cont}, C and D. What we see in Figure 4-1 is the state estimate dynamics, given by the solid lines, and the perturbed system dynamics, given by the dotted lines. The perturbed system has a different dynamic matrix A :

$$A_{perturbed} = \begin{bmatrix} 0.010 & 1.001 \\ 0.001 & 0.001 \end{bmatrix}$$

Figure 4-2 shows the residual generated by the H_∞ filter. From this residual, the decision

function is illustrated by Figure 4-3. As in the previous numerical examples given in Section 3.5, the impulse jump is injected at time $t = 25$. We can see that the failure detection algorithm, given the residual from the H_∞ filter, can still provide correct signal of failure in the presence of modeling errors.

4.3 Robust Failure Detection Filter

The *Robust Failure Detection Filter - RFDI* - is developed based on a deterministic approach. The uncertainties that accompany the failure detection process are considered and treated in the worst case sense. In [31], it is claimed that the RFDI algorithm can account for several categories of modeling uncertainties, i.e. :

- noise and plant model uncertainties,
- failure model uncertainties,
- statistical outliers or contaminated noise.

The discussion of RFDI, based on the compilation of [33], [34], [35], [31], will be carried out step by step. First, the robust estimator for a system with uncertainty in process noise model will be presented in Section 4.3.1. Second, the robust estimator for both system model and process noise model will be given in Section 4.3.2. Finally, in Section 4.3.3, the RFDI algorithm that considers uncertainties of system model, process noise model and failure model will be discussed. The RFDI algorithm applies the robust estimator given in Section 4.3.1 and 4.3.2.

4.3.1 Robust Estimator for Systems with Uncertain Process Noise Model

In the estimation problem, estimator F tries to mimic the dynamic behavior of plant P as illustrated in Figure 4-4. In the presence of input disturbance r and uncertainty of initial condition x_0, the estimator F, given the measurement y_t and an estimate of initial condition $\hat{x}_0$, reconstructs the state $\hat{x}$ of the system P.

For a *deterministic* estimator, the initial state x_0 and the disturbance r are characterized by a certain set description, rather than their stochastic attributes. The description of characterizing noise by a certain set is given in Paganini's Ph.D thesis [37].

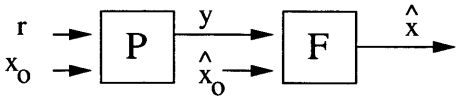

Figure 4-4: The estimation problem of plant dynamic

Before proceeding to the formulation of the robust estimation problem, we introduce the following terms :

$$\begin{aligned} r &= [r_0, \ldots, r_{N-1}] \\ y &= [y_0, \ldots, y_{N-1}] \\ e &= [e_1, \ldots, e_N] \end{aligned}$$

Note that each vector has finite time dimension N which constitutes a finite-horizon problem. The following norm descriptions are used

$$\begin{aligned} \|r\| &= (\sum_{k=0}^{N-1} r_k^T r_k)^{\frac{1}{2}} \\ \|y\| &= (\sum_{k=1}^{N} y_k^T y_k)^{\frac{1}{2}} \\ \|e\| &= (\sum_{k=0}^{N-1} e_k^T e_k)^{\frac{1}{2}} \end{aligned}$$

A robust estimator is developed in such a way that the state estimate $\hat{x}_t$ will have minimum error in the presence of a worst-case disturbance r and a worst-case initial state estimation error $x_0 - \hat{x}_0$. The problem can be illustrated by Figure 4-5

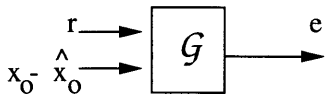

Figure 4-5: The robust estimator problem

Let $\mathcal{G}$ be a mapping from r_t and $x_0 - \hat{x}_0$ to e_t, or

$$\begin{aligned} \mathcal{G} \quad &: \quad r_t, x_0 - \hat{x}_0 \rightarrow e_t \\ e_t \quad &= \quad \mathcal{G}(r_t, x_0 - \hat{x}_0) \end{aligned} \tag{4.1}$$

The problem of robust estimation is to construct an estimator such that the function norm $\mathcal{G}$ is minimized in the worst-case combination of r_t and $x_0 - \hat{x}_0$. Minimizing the function norm is equivalent to minimizing the 2-norm of $\mathcal{G}$. Mathematically, this problem can be stated as

$$\|\mathcal{G}\|_{i2}^2 = \sup_{(r, x_0 - \hat{x}_0) \neq 0} \frac{\|e\|^2}{\|r\|^2 + \|x_0 - \hat{x}_0\|_{\check{P}_0^{-1}}^2} \tag{4.2}$$

$$= \sup_{\{\|r\|^2 + \|x_0 - \hat{x}_0\|_{\check{P}_0^{-1}}^2 \leq 1\}} \|e\|^2 \tag{4.3}$$

$$\|\mathcal{G}\|_{i2}^2 \leq \kappa^2 \tag{4.4}$$

The estimator is developed for P that has the following state-space description :

$$x_{t+1} = A_t x_t + B_t r_t \tag{4.5}$$

$$e_t = M_t(x_t - \hat{x}_t) \tag{4.6}$$

$$y_t = C_t x_t + D_t r_t \tag{4.7}$$

To obtain the optimal solution of the criterion given in equation (4.4), with system dynamics described by equations (4.5), (4.6), (4.7) are treated as constraints, a game theoritic approach is used for optimization. Appendix C.1 will provide a thorough derivation for a robust estimator with the uncertainty in noise model.

Now, we are ready to formulate the robust estimator for uncertain noise model.
For the linear time-varying system described by the following state-space equations

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t r_t \\ e_t &= M_t(x_t - \hat{x}_t) \\ y_t &= C_t x_t + D_t r_t \end{aligned}$$

with initial condition x_0 and initial error $e_0 = x_0 - \hat{x}_0$, the robust estimator for known

system dynamics and unknown noise model will be

$$\hat{x}_{t+1} = (A_t - K_t C_t)\hat{x}_t + K_t y_t \tag{4.8}$$

$$K_t = [B_t D_t^T + A_t H_t^{-1} C_t^T][D_t D_t^T + C_t H_t^{-1} C_t^T]^{-1} \tag{4.9}$$

$$H_t = P_t^{-1} - \gamma^{-2} M_t^T M_t \tag{4.10}$$

$$\tilde{A}_t = A_t - K_t C_t \tag{4.11}$$

$$\tilde{B}_t = B_t - K_t D_t \tag{4.12}$$

$$P_{t+1} = \tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T \tag{4.13}$$

$$P_0 = \breve{P}_0 > 0 \tag{4.14}$$

The equation (4.8) is the standard form of an estimator equation, with gain K_t given by equation (4.9), and noise covariance propagation given in equation (4.13). When we take the limit of γ, such that :

$$\lim_{\gamma \to \infty} H_t = P_t^{-1}$$

then we obtain the Kalman filter algorithm. In this case, γ reflects the uncertainty in noise model.

4.3.2 Robust Estimator for System with Uncertain System Model and Process Noise Model

The problem of estimation in the presence of uncertainties in both the system model and the process noise model is the generalization of the problem given in Section 3.1. This problem can be illustrated by Figure (4-6) There are two additional variables in Figure (4-6),

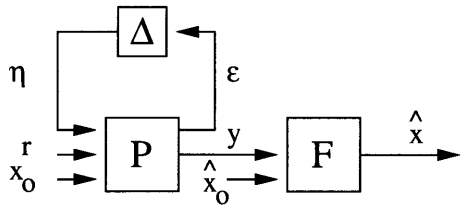

Figure 4-6: The estimation problem with uncertainties in system and noise models

$$\eta = [\eta_0, \ldots, \eta_{N-1}]$$
$$\epsilon = [\epsilon_0, \ldots, \epsilon_{N-1}],$$

with their norms are defined by

$$\|\eta\| = (\sum_{k=0}^{N-1} \eta_k^T \eta_k)^{\frac{1}{2}}$$
$$\|\epsilon\| = (\sum_{k=0}^{N-1} \epsilon_k^T \epsilon_k)^{\frac{1}{2}}.$$

These variables will be included in the optimization process. As for the previous discussion in Section 3.1, all variables are defined over a finite time interval.

For the general case, the robust estimator will minimize the induced operator $\mathcal{G}$ that maps the input disturbance r_t and initial estimation error $x_0 - \hat{x}_0$ to the estimation error e_t, for all possible model perturbations Δ of bounded induced 2-norm. The performance index for this problem is given by the following equations :

$$\|\mathcal{G}\|_{i2}^2 = \sup_{\{\|r\|^2 + \|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \|x_0\|^2_{X_0=1}\}} \|e\|^2 \tag{4.15}$$
$$\|\mathcal{G}\|_{i2}^2 < \kappa^2 \tag{4.16}$$
$$\forall \Delta \ni \|\Delta\|_{i2}^2 = \sup_{\|\epsilon\| \neq 0} \frac{\|\eta\|^2}{\|\epsilon\|^2} < \beta^2 \tag{4.17}$$

The system dynamics given by the following state-space equations

$$x_{t+1} = A_t x_t + B_t d_t \tag{4.18}$$
$$\epsilon_t = S_t x_t + T_k d_t \tag{4.19}$$
$$e_t = M_t(x_t - \hat{x}_t) \tag{4.20}$$
$$y_t = C_t x_t + D_t d_t \tag{4.21}$$
$$d_t = [r_t^T \quad \eta_t^T]^T \tag{4.22}$$

will act as constraints in the optimization problem of equation (4.16).

In the state-space equations (4.18) - (4.22) that describe the system dynamics and its uncertainty, we see that there are two new matrices S and T. These matrices describe how the uncertainties enter into the system. In general, the system with uncertainties can be

described by the following state-space equations :

$$x_{t+1} = \left(A_t + \sum_{j=1}^{k} \Delta A_{j_t} \delta_j\right) x_t + \left(G_t + \sum_{j=1}^{k} \Delta G_{j_t} \delta_j\right) r_t \tag{4.23}$$

$$y_t = \left(H_t + \sum_{j=1}^{k} \Delta H_{j_t} \delta_j\right) x_t + \left(E_t + \sum_{j=1}^{k} \Delta E_{j_t} \delta_j\right) r_t \tag{4.24}$$

where k indicates the order of uncertainties. Note that the state-space equations in (4.23) and (4.24) are time-varying. For each uncertainty j, we can construct the concatenated matrix $N_{j_t} \in \mathcal{R}^{(dim(x)+dim(y))\times(dim(x)+dim(r))}$ in the following way :

$$N_{j_t} = \begin{bmatrix} \Delta A_{j_t} & \Delta G_{j_t} \\ \Delta H_{j_t} & \Delta E_{j_t} \end{bmatrix}. \tag{4.25}$$

It is possible to decompose matrix N_{j_t} into

$$N_{j_t} = \begin{bmatrix} Q_{j_t} \\ R_{j_t} \end{bmatrix} \times \begin{bmatrix} S_{j_t} & L_{j_t} \end{bmatrix} \tag{4.26}$$

where

- $Q_{j_t} \in \mathcal{R}^{dim(x)\times rank(N_{j_t})}$
- $R_{j_t} \in \mathcal{R}^{dim(y)\times rank(N_{j_t})}$
- $S_{j_t} \in \mathcal{R}^{rank(N_{j_t})\times dim(x)}$
- $L_{j_t} \in \mathcal{R}^{rank(N_{j_t})\times dim(r)}$

Recall that $d_t = [\eta_t \quad r_t]$. The equations (4.23) and (4.24) are now rewritten with $Q_{j_t}, R_{j_t}, S_{j_t}$ and L_{j_t} matrices included :

$$\begin{aligned} x_{t+1} = & \left(A_t + \sum_{j=1}^{k} Q_{j_t} \delta_j I^{rank(N_{j_t})\times rank(N_{j_t})} S_{j_t}\right) x_t \\ & + \left(G_t + \sum_{j=1}^{k} Q_{j_t} \delta_j I^{rank(N_{j_t})\times rank(N_{j_t})} L_{j_t}\right) r_t \end{aligned} \tag{4.27}$$

$$
\begin{aligned}
&= A_t x_t + \begin{bmatrix} Q_{1_t} & \dots & Q_{k_t} \end{bmatrix} \begin{bmatrix} \eta_{1_t} \\ \vdots \\ \eta_{k_t} \end{bmatrix} + G_t r_t \\
&= A_t x_t + Q_t \eta_t + G_t r_t \\
&= A_t x_t + B_t d_t \\
y_t &= \left(H_t + \sum_{j=1}^{k} R_{j_t} \delta_j I^{rank(N_{j_t}) \times rank(N_{j_t})} S_{j_t} \right) x_t \\
&\quad + \left(E_t + \sum_{j=1}^{k} R_{j_t} \delta_j I^{rank(N_{j_t}) \times rank(N_{j_t})} L_{j_t} \right) r_t \\
&= H_t x_t + \begin{bmatrix} R_{1_t} & \dots & R_{k_t} \end{bmatrix} \begin{bmatrix} \eta_{1_t} \\ \vdots \\ \eta_{k_t} \end{bmatrix} + E_t r_t \\
&= H_t x_t + R_t \eta_t + E_t r_t \\
&= H_t x_t + D_t d_t \\
\epsilon_t &= \begin{bmatrix} \epsilon_{1_t} \\ \vdots \\ \epsilon_{k_t} \end{bmatrix} \\
&= \begin{bmatrix} S_{1_t} \\ \vdots \\ S_{k_t} \end{bmatrix} x_t + 0\eta_k + \begin{bmatrix} L_{1_t} \\ \vdots \\ L_{k_t} \end{bmatrix} r_t \\
&= S_t x_t + T_t d_t
\end{aligned}
\tag{4.28}
\tag{4.29}
$$

where

$$
T_t = \begin{bmatrix} 0 & L_{1_t} \\ \dots & \dots \\ 0 & L_{k_t} \end{bmatrix} \tag{4.30}
$$

The relationship between η_t and ϵ_t is then described by

$$
\begin{aligned}
\eta_t &= \begin{bmatrix} \delta_1 & & \\ & \ddots & \\ & & \delta_k \end{bmatrix} \begin{bmatrix} \epsilon_{1_t} \\ \vdots \\ \epsilon_{k_t} \end{bmatrix} && (4.31) \\
&= \Delta \epsilon_t && (4.32)
\end{aligned}
$$

The following 2-dimension continuous system is given to illustrate how to construct the uncertainty matrices. In this example, the system is time-invriant, so we can drop the subscript "$_t$" off the equations. This example is taken from [2].
Suppose we have the following state-space equations :

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_0}{m_0} & -\frac{b_0}{m_0} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_0} & 0 \end{bmatrix} \begin{bmatrix} w(t) \\ e(t) \end{bmatrix}$$

$$\dot{x}(t) = A_0 x(t) + B_0 d(t)$$

$$y(t) = [1 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + [0 \quad 1] \begin{bmatrix} w(t) \\ e(t) \end{bmatrix}$$

$$y = H_0 x(t) + D_0 d(t)$$

where the subscript "$_0$" indicates the nominal values. Suppose we have the knowledge that the parameter values are within certain ranges, i.e.,

$$\begin{aligned} \frac{1}{m} &= \frac{1}{m_0} + \omega_1 \delta_1 \\ b &= b_0 + \omega_2 \delta_2 \\ k &= k_0 + \omega_3 \delta_3 \end{aligned}$$

where

$$-1 < \delta_j < 1$$

and ω_j describes the distance from the nominal value to its maximum and minimum values. The uncertainties appear in $\frac{b}{m}$, $\frac{k}{m}$ and mathematically, these can be described as

$$\begin{aligned} \frac{b}{m} &= [b_0 + \omega_2 \delta_2] \times \left[\frac{1}{m_0} + \omega_1 \delta_1\right] \\ &= \frac{b_0}{m_0} + b_0 \omega_1 \delta_1 + \frac{\omega_2}{m_0} \delta_2 + O(\delta^2) \end{aligned}$$

$$\frac{k}{m} = [k_0 + \omega_3 \delta_3] \times \left[\frac{1}{m_0} + \omega_1 \delta_1\right]$$

$$= \frac{k_0}{m_0} + k_0\omega_1\delta_1 + \frac{\omega_3}{m_0}\delta_3 + O(\delta^2).$$

Using the description of these uncertainties, ignoring the higher order terms, we rewrite the matrices A, B, H, and D as

$$\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_0}{m_0} & -\frac{b_0}{m_0} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -k_0\omega_1 & -b_0\omega_1 \end{bmatrix}\delta_1 + \begin{bmatrix} 0 & 0 \\ 0 & -\frac{\omega_2}{m_0} \end{bmatrix}\delta_2 + \begin{bmatrix} 0 & 0 \\ -\frac{\omega_3}{m_0} & 0 \end{bmatrix}\delta_3$$

$$A = A_0 + \Delta A_1 + \Delta A_2 + \Delta A_3$$

$$\begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{1}{m_0} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \omega_1 & 0 \end{bmatrix}\delta_1 + \underline{\mathbf{0}}\delta_2 + \underline{\mathbf{0}}\delta_3$$

$$B = B_0 + \Delta B_1 + \Delta B_2 + \Delta B_3$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} + \underline{\mathbf{0}}\delta_1 + \underline{\mathbf{0}}\delta_2 + \underline{\mathbf{0}}\delta_3$$

$$H = H_0 + \Delta H_1 + \Delta H_2 + \Delta H_3$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} + \underline{\mathbf{0}}\delta_1 + \underline{\mathbf{0}}\delta_2 + \underline{\mathbf{0}}\delta_3$$

$$D = D_0 + \Delta D_1 + \Delta D_2 + \Delta D_3$$

The matrix N_{jt} for each j uncertainty is described by

$$N_1 = \begin{bmatrix} \Delta A_1 & \Delta B_1 \\ \Delta H_1 & \Delta D_1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ -k_0\omega_1 & -b_0\omega_1 & \omega_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$N_2 = \begin{bmatrix} \Delta A_2 & \Delta B_2 \\ \Delta H_2 & \Delta D_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\omega_2}{m_0} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$N_3 = \begin{bmatrix} \Delta A_3 & \Delta B_3 \\ \Delta H_3 & \Delta D_3 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{\omega_3}{m_0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrices N_1, N_2, N_3 are then decomposed into their corresponding matrices Q_j, R_j, S_j and L_j for each uncertainty mode j :

$$Q_1 = \begin{bmatrix} 0 \\ \omega_1 \end{bmatrix}, \qquad S_1 = \begin{bmatrix} -k_0 & -b_0 \end{bmatrix}, \qquad L_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 0 \\ \omega_2 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 0 & -\frac{1}{m_0} \end{bmatrix}, \qquad L_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0 \end{bmatrix}$$

$$Q_3 = \begin{bmatrix} 0 \\ \omega_3 \end{bmatrix}, \qquad S_3 = \begin{bmatrix} -\frac{1}{m_0} & 0 \end{bmatrix}, \qquad L_3 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 0 \end{bmatrix}$$

The concatenated matrices of Q, R, S and T are

$$Q = \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \omega_1 & \omega_2 & \omega_3 \end{bmatrix}$$

$$R = \begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} -k_0 & -b_0 \\ 0 & -\frac{1}{m_0} \\ -\frac{1}{m_0} & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & L_1 \\ 0 & L_2 \\ 0 & L_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The approach used in [31] to solve the optimization problem is by treating η as input and ϵ as output variables. The new problem now can be illustrated by Figure (4.3.2) The

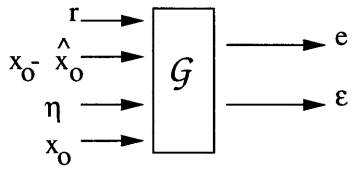

Figure 4-7: The general optimization in estimation problem

new formulation of the optimization problem is to minimize the following performance index

$$\bar{J}_1 = \sup_{(r,\eta,x_0,x_0-\hat{x}_0)\neq 0} \frac{\|e\|^2 + \|\epsilon\|^2}{\|r\|^2 + \|\eta\|^2 + \|x_0\|^2_{\breve{X}_0} + \|x_0 - \hat{x}_0\|^2_{\breve{P}_0^{-1}}} \leq \kappa^2 \qquad (4.33)$$

Note the following facts :

- $\bar{J}_1 < \kappa^2$ implies

$$\|e\|^2 + \|\epsilon\|^2 < \kappa^2(\|r\|^2 + \|\eta\|^2 + \|x_0\|^2_{\hat{X}_0} + \|x_0 - \hat{x}_0\|^2_{P_0^{-1}})$$

- the bound on perturbation Δ and the condition $\beta\kappa < 1$ give

$$\begin{aligned} \|\eta\|^2 &< \beta^2\|\epsilon\|^2 \\ \|\eta\|^2 &< \kappa^{-2}\|\epsilon\|^2. \end{aligned}$$

- These facts will lead to the following inequalities

$$\begin{aligned} \|e\|^2 + \|\epsilon\|^2 &< \kappa^2(\|r\|^2 + \|\eta\|^2 + \|x_0\|^2_{\hat{X}_0} + \|x_0 - \hat{x}_0\|^2_{P_0^{-1}}) \\ \|e\|^2 + \|\epsilon\|^2 &< \kappa^2(\|r\|^2 + \kappa^{-2}\|\epsilon\|^2 + \|x_0\|^2_{\hat{X}_0} + \|x_0 - \hat{x}_0\|^2_{P_0^{-1}}) \\ \mathcal{G}_{i2}^2 &< \kappa^2 \end{aligned}$$

This means that working on the $\bar{J}_1$ performance index for the optimization problem will solve the optimization with respect to minimizing $\|\mathcal{G}\|$. In appendix C.2 , we will see how a game theoritic approach is used to solve the optimization problem of equation (4.33) subject to the constraints given by equations (4.18)- (4.22). We will present the solution here.
In the general estimation problem, where the uncertainties exist in the system model and noise model, the robust estimator of the linear time-varying system :

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t d_t \\ \epsilon_t &= S_t x_t + T_k d_t \\ e_t &= M_t(x_t - \hat{x}_t) \\ y_t &= C_t x_t + D_t d_t \\ d_t &= [r_t^T \quad \eta_t^T]^T \end{aligned} \tag{4.34}$$

will be given by the the following equations :

$$\hat{x}_{t+1} = (\bar{A}_t - \bar{K}_t\bar{C}_t)\hat{x}_t + \bar{K}_t y_t \tag{4.35}$$

$$\bar{K}_t = [\bar{B}_t\bar{D}_t^T + \bar{A}_t\bar{H}_t^{-1}\bar{C}_t^T][\bar{D}_t\bar{D}_t^T + \bar{C}_t\bar{H}_t^{-1}\bar{C}_t^T]^{-1} \tag{4.36}$$

$$\bar{H}_t = \bar{P}_t^{-1} - \gamma^{-2}M_t^T M_t \tag{4.37}$$

$$\begin{aligned} \bar{P}_{t+1} &= (\bar{A}_t - \bar{K}_t\bar{C}_t)\bar{H}_t^{-1}(\bar{A}_t - \bar{K}_t\bar{C}_t)^T \\ &\quad +(\bar{B}_t - \bar{K}_t\bar{D}_t)(\bar{B}_t - \bar{K}_t\bar{D}_t)^T \end{aligned} \tag{4.38}$$

$$\bar{P}_0 = \breve{P}_0 \tag{4.39}$$

where

$$\bar{A}_t = A_t + \gamma^{-2} B_t Z_t^{-1} F_t^T \tag{4.40}$$

$$\bar{B}_t = B_t Z_t^{-\frac{1}{2}} \tag{4.41}$$

$$\bar{C}_t = C_t + \gamma^{-2} D_t Z_t^{-1} F_t^T \tag{4.42}$$

$$\bar{D}_t = D_t Z_t^{-\frac{1}{2}}. \tag{4.43}$$

The quantities F_t and Z_t are defined as follows

$$F_t = S_t^T T_t + A_t^T X_{t+1} B_t \tag{4.44}$$

$$Z_t = I - \gamma^{-2}(T_t^T T_t + B_t^T X_{t+1} B_t) \tag{4.45}$$

where X_t is the solution of the following Riccati equation

$$X_t = A_t^T X_{t+1} A_t + S_t^T S_t + \gamma^{-2} F_t Z_t^{-1} F_t^T \tag{4.46}$$

$$X_N = 0 \tag{4.47}$$

The Riccati equation given in equation (4.46) runs backward in time. This situation is very similar to the LQR problem whose Riccati equation should be solved in reversed time. It is still possible, however, to apply this algorithm *on-line*. The intermediate sequence X_t can be computed off-line knowing only the plant matrices and the disturbance matrices. This method of solving backward Riccati equation for real-time application is given in [30]. The method for real-time robust estimation application is illustrated by Figure 4-8.

The robust estimator that has been derived is related to the Kalman filter. It is obvious that taking the limit of $\gamma \to \infty$ will give the following quanitities :

$$\lim_{\gamma\to\infty} \bar{A}_t = A_t$$

$$\lim_{\gamma\to\infty} \bar{B}_t = B_t$$

$$\lim_{\gamma\to\infty} \bar{C}_t = C_t$$

$$\lim_{\gamma\to\infty} \bar{D}_t = D_t$$

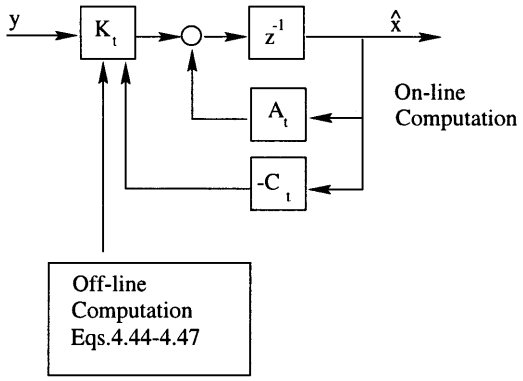

Figure 4-8: On-line application of robust estimator

$$\lim_{\gamma \to \infty} \bar{H}_t = P_t^{-1}$$

The robust estimator again returns to the Kalman filter. It is clear now that the Kalman filter will provide the best estimate of x_t under the absence of the system model and noise model uncertainties.

4.3.3 The General RFDI

Robust Failure Detection and Isolation - RFDI was developed to be sensitive to the presence of a failure while remaining robust to failure mode, noise and plant model uncertainties. In [35], it is shown that the RFDI has wider robustness coverage compared to the existing failure detection algorithms.

This algorithm applies the robust estimator, developed in Section 3.2 in its failure detection scheme. The failure in the RFDI scheme is treated as an output vector of a dynamic system. The dynamics model of failure is assumed to be a shaping filter driven by white noise. Hence, the failure shaping filter has the following Gauss-Markov model :

$$\phi_{t+1} = A_{f_t}\phi_t + B_{f_t}\xi_t \tag{4.48}$$

$$\nu_t = C_{f_t}\phi_t \tag{4.49}$$

where ξ_t is the white noise driving the failure shaping filter and ν_t is the failure that enters the plant. The RFDI can be illustrated by Figure 4-9, with P_f is the shaping filter driven

by the white noise ξ_t producing the output ν_t.

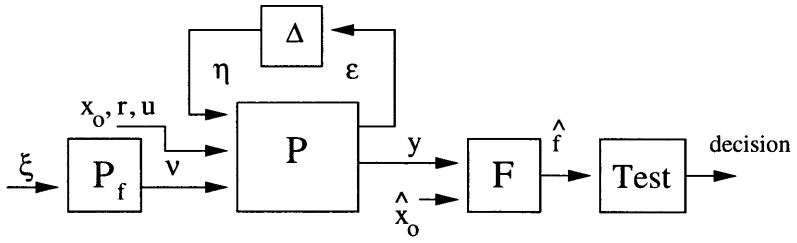

Figure 4-9: Representation of RFDI algorithm

The failure dynamic system in equations (4.48) - (4.49), from a deterministic point of view, is driven by bounded l_2-norm of ξ_t and has an l_2-bounded ϕ_0. The same system, from a stochastic point of view, is driven by a unitary white-noise ξ_t with unit variance, and ϕ_0 has zero mean and covariance Q_ϕ.

The failure, in the form of a jump, is a special form of the system described in equations (4.48) - (4.49). In this particular case, the model will be

$$\phi_{t+1} = \phi_t \tag{4.50}$$

$$\nu_t = \phi_t \tag{4.51}$$

where $Q_\phi = \infty$.

Representing the failure in this fashion is not the only way. In Hall's thesis [25], the algorithm for failure detection called *Orthogonal Series GLR - OSGLR*, all failure modes are represented by orthogonal series of Laguerre or Legendre functions. The idea is developed based on the principle that the orthonormal bases span the failure space. All failure modes can be described by combination of each orthonormal basis. This approach is not explored here since the OSGLR algorithm requires more states to represent each failure [31].

The problem of hypothesis testing with the RFDI can now be formulated as follows. For the hypotheses:

$$\begin{aligned} H_0: \quad x_{t+1} &= A_t x_t + Q_t \eta_t + B_t + U_t u_t \\ \epsilon_t &= S_t x_t + T_t \eta_t \end{aligned}$$

$$
\begin{aligned}
& & y_t &= H_t + R_t\eta_t + D_t r_t + W_t u_t, \\
H_1: & & x_{t+1} &= A_t x_t + Q_t\eta_t + B_t + U_t u_t + G_t\nu_t \\
& & \epsilon_t &= S_t x_t + T_t\eta_t \\
& & y_t &= H_t + R_t\eta_t + D_t r_t + W_t u_t + L_t\nu_t,
\end{aligned}
$$

the RFDI algorithm is applied by treating the system of Figure 4-9 as an augmented system. The plant state-space equations and the shaping filter state-space are augmented to give the following state-space equations :

$$
\begin{aligned}
x_{t+1} &= A_t x_t + G_t C_{f_t}\phi_t + Q_t\eta_t + B_t r_t + U_t u_t \\
\phi_{t+1} &= A_{f_t}\phi_t + B_{f_t}\xi_t
\end{aligned}
$$

or, in a more compact form :

$$
\begin{bmatrix} x_{t+1} \\ \phi_{t+1} \end{bmatrix} = \begin{bmatrix} A_t & G_t C_{f_t} \\ 0 & A_{f_t} \end{bmatrix} \begin{bmatrix} x_t \\ \phi_t \end{bmatrix} + \begin{bmatrix} B_t & Q_t & U_t & 0 \\ 0 & 0 & 0 & B_{f_t} \end{bmatrix} \begin{bmatrix} r_t \\ \eta_t \\ u_t \\ \xi_t \end{bmatrix} \tag{4.52}
$$

with initial estimate

$$
\begin{bmatrix} x_0 \\ \phi_0 \end{bmatrix} = \begin{bmatrix} \hat{x}_0 \\ 0 \end{bmatrix} \tag{4.53}
$$

The matrices G_t and L_t describe the way the failure enters the system states and measurements. When the failure enters the states, then it is considered as actuator failures, while if the failure enters the measurements, then it is considered as sensor failures.

For a stochastic approach, the initial error is a zero mean and has a covariance given by

$$
P_0 = \begin{bmatrix} \breve{P}_0 & 0 \\ 0 & Q_\phi \end{bmatrix}. \tag{4.54}
$$

While for a deterministic point of view, the initial estimation error has a weighted Euclidean norm, with weight given by P_0^{-1}.

The associated observaton equation is given by :

$$\begin{bmatrix} y_t \\ \nu_t \end{bmatrix} = \begin{bmatrix} H_t & L_t C_{f_t} \\ 0 & C_{f_t} \end{bmatrix} \begin{bmatrix} x_t \\ \phi_t \end{bmatrix} + \begin{bmatrix} R_t & D_t & W_t & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_t \\ \eta_t \\ u_t \\ \xi_t \end{bmatrix} \tag{4.55}$$

To put the augmented system into the RFDI framework, we define the following matrices first :

$$\check{A} = \begin{bmatrix} A_t & G_t C_{f_t} \\ 0 & A_{f_t} \end{bmatrix} \tag{4.56}$$

$$\check{B} = \begin{bmatrix} B_t & Q_t & U_t & 0 \\ 0 & 0 & 0 & B_{f_t} \end{bmatrix} \tag{4.57}$$

$$\check{C} = \begin{bmatrix} H_t & L_t C_{f_t} \end{bmatrix} \tag{4.58}$$

$$\check{D} = \begin{bmatrix} R_t & D_t & W_t & 0 \end{bmatrix}. \tag{4.59}$$

Hence, following the form of the robust estimator given in Section 4.3.2,

$$\begin{bmatrix} \hat{x}_{t+1} \\ \hat{\phi}_{t+1} \end{bmatrix} = (\bar{A}_t - \bar{K}_t \bar{C}_t) \begin{bmatrix} \hat{x}_t \\ \hat{\phi}_t \end{bmatrix} + \bar{K}_t y_t \tag{4.60}$$

$$\bar{K}_t = [\bar{B}_t \bar{D}_t^T + \bar{A}_t \bar{H}_t^{-1} \bar{C}_t^T][\bar{D}_t \bar{D}_t^T + \bar{C}_t \bar{H}_t^{-1} \bar{C}_t^T]^{-1} \tag{4.61}$$

$$\bar{A}_t = \check{A}_t + \gamma^{-2} \check{B}_t Z_t^{-1} F_t^T \tag{4.62}$$

$$\bar{B}_t = \check{B}_t Z_t^{-\frac{1}{2}} \tag{4.63}$$

$$\bar{C}_t = \check{C}_t + \gamma^{-2} \check{D}_t Z_t^{-1} F_t^T \tag{4.64}$$

$$\bar{D}_t = \check{D}_t Z_t^{-\frac{1}{2}} \tag{4.65}$$

$$F_t = S_t^T T_t + \check{A}_t^T X_{t+1} \check{B}_t \tag{4.66}$$

$$Z_t = I - \gamma^{-2} (T_t^T T_t + \check{B}_t^T X_{t+1} \check{B}_t) \tag{4.67}$$

$$\bar{H}_t = \bar{P}_t^{-1} - \gamma^{-2} M_t^T M_t \tag{4.68}$$

The following two Riccati equations will give the positive definite solutions for X_t and P_t.

$$X_t = \check{A}_t^T X_{t+1} \check{A}_t + S_t^T S_t + \gamma^{-2} F_t Z_t^{-1} F_t^T \tag{4.69}$$

$$X_{t_f=N} = 0 \tag{4.70}$$

$$\begin{aligned} \bar{P}_{t+1} &= (\bar{A}_t - \bar{K}_t \bar{C}_t) \bar{H}_t^{-1} (\bar{A}_t - \bar{K}_t \bar{C}_t)^T \\ &\quad + (\bar{B}_t - \bar{K}_t \bar{D}_t)(\bar{B}_t - \bar{K}_t \bar{D}_t)^T \end{aligned} \tag{4.71}$$

$$\bar{P}_0 = \check{P}_0 \tag{4.72}$$

We now define several quantities before proceeding to describe the decision function. Defining the following vectors :

$$Y = \left[y_{t_0}^T, \ldots, y_{t_f}^T \right]^T \tag{4.73}$$

$$\underline{\nu} = \left[\nu_{t_0}^T, \ldots, \nu_{t_f}^T \right]^T \tag{4.74}$$

$$\mathcal{G} = \begin{bmatrix} L & & & \\ CG & L & & \\ CAG & CG & L & \\ \ldots & \ldots & \ldots & \\ CA^{t_f - t_i - 1} G & CA^{t_f - t_i - 2} G & CG & L \end{bmatrix} \tag{4.75}$$

$$\Sigma_0 = cov\left[Y_0, Y_0 \right] \tag{4.76}$$

The decision function will be given by

$$\begin{aligned} \mathcal{D} &= \log \frac{\exp\left[-Y^T (\Sigma_0 + \mathcal{G} \Sigma_{\underline{\nu}} \mathcal{G}^T)^{-1} \right]}{\exp\left[-Y^T \Sigma_0 Y \right]} + C_{const} \\ &= -Y^T \left[(\Sigma_0 + \mathcal{G} \Sigma_{\underline{\nu}} \mathcal{G}^T)^{-1} - \Sigma_0^{-1} \right] Y + C_{const} \end{aligned}$$

If we apply the matrix inversion lemma, such that

$$\begin{aligned} (\Sigma_0 + \mathcal{G} \Sigma_{\underline{\nu}} \mathcal{G}^T)^{-1} &= \Sigma_0^{-1} - \Sigma_0^{-1} \mathcal{G} (\mathcal{G}^T \Sigma_0^{-1} \mathcal{G} + \Sigma_{\underline{\nu}}^{-1})^{-1} \mathcal{G}^T \Sigma_0^{-1} \\ &= \Sigma_0^{-1} - \Sigma_0^{-1} \mathcal{G} \Sigma_{\underline{\nu}|Y} \mathcal{G}^T \Sigma_0^{-1}, \end{aligned}$$

then we will have, apart from the constant term,

$$\mathcal{D} = Y^T \Sigma_0^{-1} \mathcal{G} \Sigma_{\underline{\nu}|Y} \mathcal{G}^T \Sigma_0^{-1} Y \tag{4.77}$$

If what we have is the estimate of ν, we use the following equation

$$\mathcal{D} = \sum_{k=t_i}^{t_f} \hat{\nu}_k^T \mathcal{S}_k \hat{\nu}_k \tag{4.78}$$

For accurate models, $\mathcal{S}_k$ is the S_k^{-1}, the error covariance matrix from the Kalman filter. For deterministic model with uncertainties, it is claimed in [35] that it is a matrix of free parameters, whose selection is an open issue.

4.4 Numerical Examples

In this section, we use the same 2-dimension system as in Section 3.5. We will explore the algorithm for two cases. For the first case, we assume that we know the model accurately, but we do not have good knowledge of the noise model. For the second case, we assume that both system model and noise model are not known precisely.

Case 1: Unknown noise model. The failure model is assumed to have the following model :

$$\begin{aligned} \phi_{t+1} &= \phi_t \\ \nu_t &= \phi_t. \end{aligned}$$

The augmented system dynamic equations are then described by :

$$\left[\begin{array}{c} x_{t+1} \\ \hline \phi_{t+1} \end{array}\right] = \left[\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & 1 & 10 \\ \hline 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c} x_t \\ \hline \phi_t \end{array}\right] + \left[\begin{array}{cc} 0.5 & 0 \\ 1 & 0 \\ \hline 0 & 0 \end{array}\right] r_t$$

$$y_t = \left[\begin{array}{cc|c} 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} x_t \\ \hline \phi_t \end{array}\right] + \left[\begin{array}{cc} 0 & 1 \end{array}\right] r_t$$

where $r_t = [w_t \quad e_t]^T$. For simulation, the following quantities are used

$$\underline{x}_{t_0} = \left[\frac{x_{t_0}}{\phi_{t_0}}\right] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$P_{\underline{x}_{t_0}} = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 10000 \end{bmatrix}$$

$$\gamma = 5$$

To simulate this system, we follow the algorithm from equation (4.8) - (4.13). The outputs of the estimator with no failure are shown in Figure 4-10. With the absence of failure, we

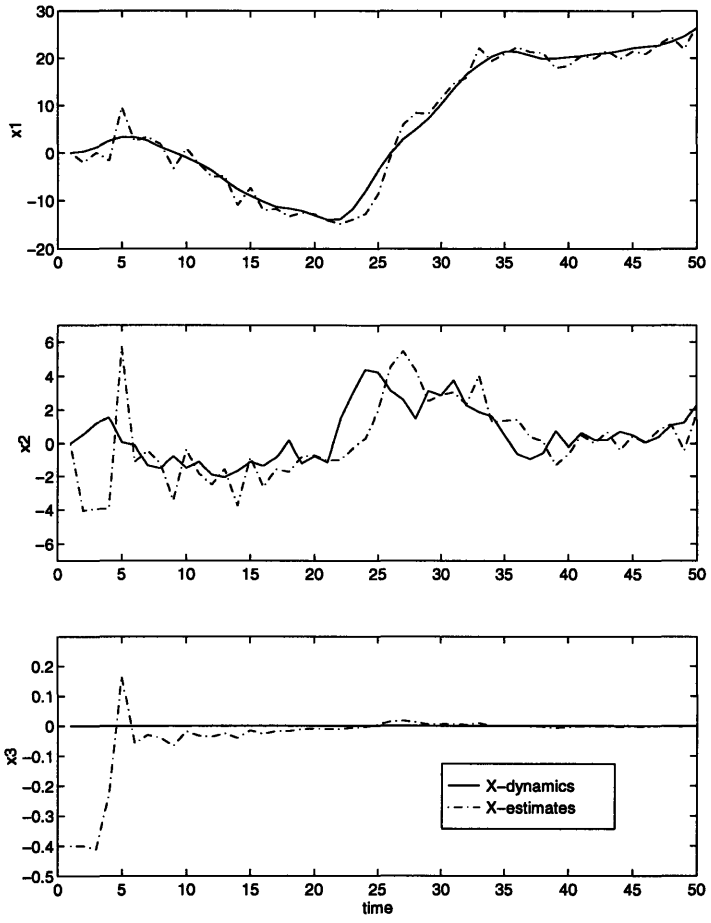

Figure 4-10: Robust estimation in the absence of failure

can see that the third state, x_3, is zero after the transient period. The second state, x_2 is not estimated very well due to the fact that we only measure the first state x_1. The estimator tracks the first state, x_1, quite well, even though a small γ is used. Using higher γ will give a better estimation.

In the presence of failure, the detection function is given by the following equation :

$$\mathcal{D}_f = \sum_{t=t_0}^{t=t_f} \nu_t P_{33_t}^{-1} \nu_t$$

where P_{33_t} is the $(3,3) - th$ entry of matrix P from equation (4.13). The decision function for jump failure is given in Figure 4-11 and the ramp failure is presented in Figure 4-12.

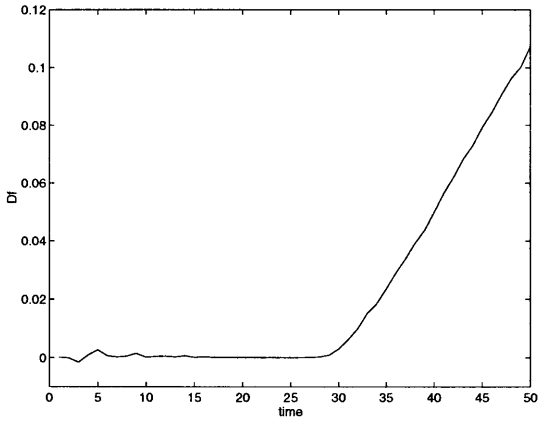

Figure 4-11: Decision function of jump failure

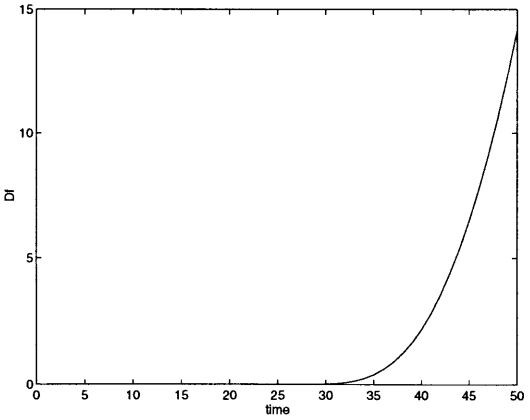

Figure 4-12: Decision function of ramp failure

The use of inverted P_{33_t} suppresses the transient response which is significantly large. The decision functions of both ramp failure and jump failure do not grow instantly. If the delay does not affect the whole performance of the system, this algorithm would be an

alternative for real-time applications.

Case 2: Unknown dynamics and noise model . We still use the same model of failure dynamics. Suppose that we have uncertainties in time constant τ_g and sampling rate ς from the system model. The corresponding matrix N_1 and N_2 will be

$$N_1 = \begin{bmatrix} 0 & \frac{1}{100} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$N_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{10000} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After decomposition, we obtain the following matrices

$$Q_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & \frac{1}{100} & 0 \end{bmatrix}, \quad L_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0 \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0 & \frac{1}{10000} & 0 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0 \end{bmatrix}$$

The matrices Q_i and the matrices R_i, $i = 1, 2$, are concatenated with matrices B and D. The new system dynamics will be :

$$\begin{bmatrix} \frac{x_{t+1}}{\phi_{t+1}} \end{bmatrix} = \left[\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & 1 & 10 \\ \hline 0 & 0 & 1 \end{array}\right] \begin{bmatrix} \frac{x_t}{\phi_t} \end{bmatrix} + \begin{bmatrix} 1/2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_t \\ \eta_{1_t} \\ \eta_{2_t} \end{bmatrix}$$

$$y_t = \left[\begin{array}{cc|c} 1 & 0 & 0 \end{array}\right] \begin{bmatrix} \frac{x_t}{\phi_t} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_t \\ \eta_{1_t} \\ \eta_{2_t} \end{bmatrix}$$

There is an additional Riccati equation that has to be solved for this second case. The matrices S and T, given by

$$S = \begin{bmatrix} 0 & \frac{1}{100} & 0 \\ 0 & \frac{1}{10000} & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

are used to solve the second Riccati equation. The Riccati equation

$$\begin{aligned} X_{t_f} &= 0 \\ X_t &= \check{A}^T X_{t+1} \check{A} + S^T S + \gamma^{-2} F_t Z_t^{-1} F_t^T \end{aligned}$$

together with

$$\begin{aligned} F_t &= S^T T + \check{A} X_{t+1} \check{B} \\ Z_t &= I - \gamma^{-2} \left(T^T T + \check{B}^T X_{t+1} \check{B} \right) \end{aligned}$$

have to be solved first. These equations can be computed off-line before the whole process begin. The computed values of X_t, F_t and Z_t where t refers to the index inside the buffer of computation, are eventually used to determine the estimate of the augmented system.

Figure 4-13 shows the performance of the estimator in the absence of failure. Even

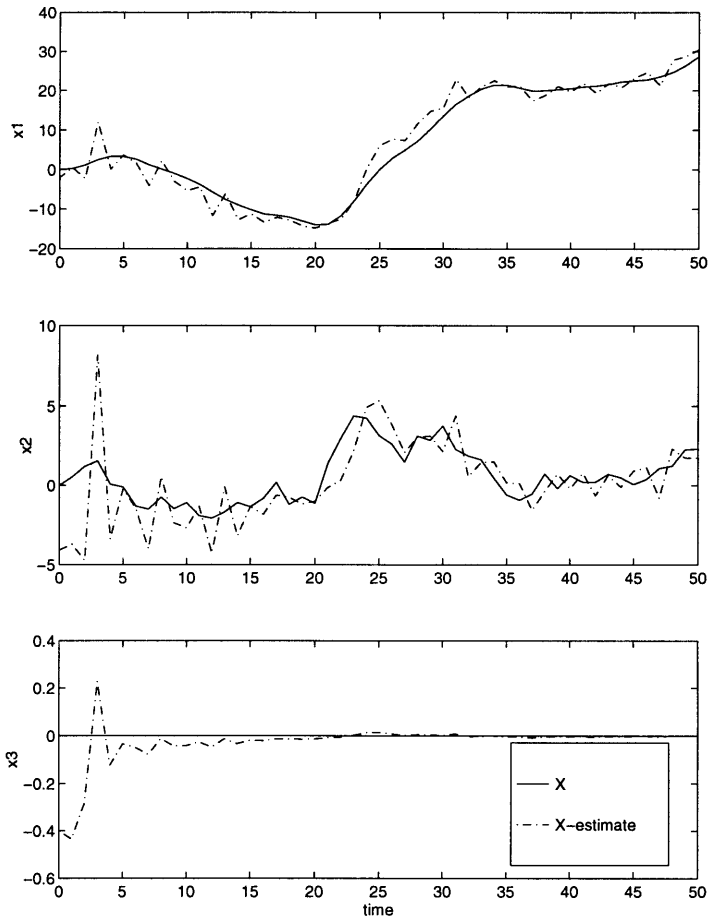

Figure 4-13: The estimate of system dynamics : Case 2

though the estimation for x_1 and the x_2 is not very accurate, the estimates $\hat{x}_1, \hat{x}_2$, still follow the dynamics of the states x_1, x_2. We also see that the estimator provides good estimate of failure at post-transient period.

Figure 4-14 presents the decision function of detection when jump failure is present, and Figure 4-15 presents the decision function when ramp failure occurs. The decision functions for both failure types are a function of the failure magnitude estimate, or x_2 from the state-space dynamic equation.

There is an interesting property that we see from both cases, the presence of detection delay. In both cases, failures were injected at time $t = 25$, and the decision functions for jump failure started increasing at $t = 30$, while the decision function for ramp failure started increasing at approximately $t = 35$. This delay may cause problems, unless the failure detection filter is set to give a decision after the whole computation of the decision

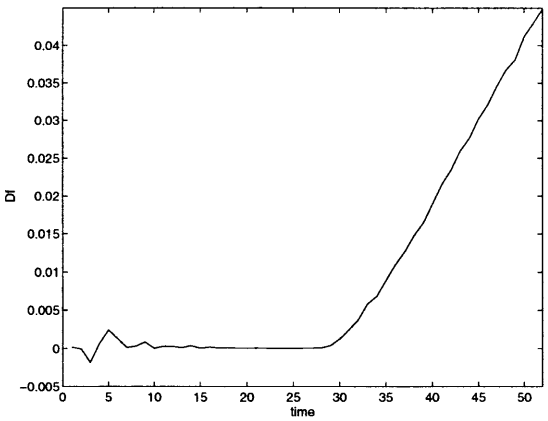

Figure 4-14: The decision function for jump failure : Case 2

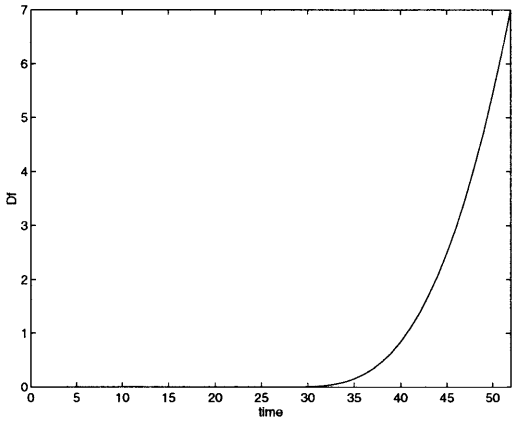

Figure 4-15: The decision function for ramp failure : Case 2

function within the buffer of observation is completed.

Chapter 5

Application of Failure Detection Algorithms

The algorithms for failure detection discussed in Chapter 3 and Chapter 4 were applied to the problem of monitoring the aircraft cockpit instrument crucial for aircraft attitude determination. The instrument configuration for aircraft attitude determination in this experiment was minimum, in the sense that only two dissimilar sensors were used to provide the attitude information and the goal is to detect any disagreement between them. This configuration is an effort to minimize the cost for reliable instruments. In the beginning of this chapter, the background of experiment is introduced. The description of the experiment, the result and the discussion are presented subsequently.

5.1 Background of Experiment : The Problem of Aircraft Attitude Determination

Attitude determination - the way to determine vehicle's orientation with respect to a specific coordinate system - is crucial for safe aircraft flight, especially when the horizon is obscured by clouds or fog. Without a reference for aircraft attitude in this situation there is virtual certainty that the pilot will lose control of the aircraft.

The straightforward solution to determine the aircraft attitude is to use a set of inertial instruments. Typically, redundant instruments are used to achieve reliability. Multiple gyroscopes are used to measure aircraft orientation angles. The word "multiple" is em-

phasized here to imply that the redundancy is applied for reliability. However, redundant instruments are expensive and it is desirable to minimize the level of redundancy, especially in general aviation aircraft.

The existence of GPS - *Global Positioning Systems* - provides another solution for attitude determination. Cohen in [9] and [16] proposed multi-antenna GPS receiver for attitude determination. The configuration is illustrated in Figure 5-1. During the research period of 1996-1997, the price of the multi-antenna GPS was still significantly high. This approach was not explored during the research.

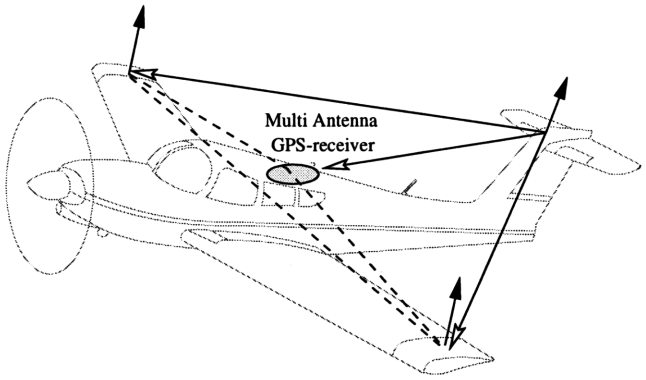

Figure 5-1: Multi-Antenna GPS receiver for attitude determination

A different approach in utilizing the GPS for attitude determination is still under development at MIT. Research work by Kornfeld [27] is exploring the minimum configuration of single-antenna GPS and IMU - *Inertial Measurement Unit* - to determine the aircraft attitude. The minimum configuration approach is illustrated in Figure 5-2. It is important to monitor the integrity of the GPS-IMU measurements. When two measurements disagree, the computer should be aware that there is a possibility that one of the instruments fails. A failure detection system conducts the continuous integrity monitoring of the two instruments. The failure detection system should inform the computer as correctly as possible that there is a disagreement in the measurements and it is solely due to the instrument failure, not other causes.

The failure detection algorithms: the GLR/MLR tests, the 2-filter MLR test and the RFDI algorithms, were used to perform the integrity monitoring task for the single antenna GPS and IMU configuration. In this experiment, these algorithms were used to evaluate the measurements from the instruments used by Kornfeld. Although the instruments used

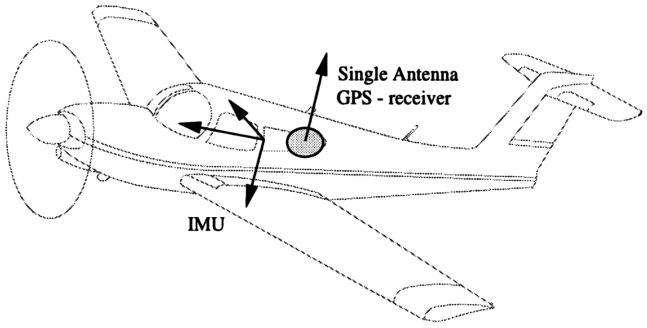

Figure 5-2: Single-Antenna GPS receiver and IMU for attitude determination

for the experiment are the same as those used in Figure 5-2, the experiment was conducted using an automobile rather than an aircraft. Figure 5-3 illustrates the configuration of the instruments for the failure detection algorithm experiment. The result of the experiment

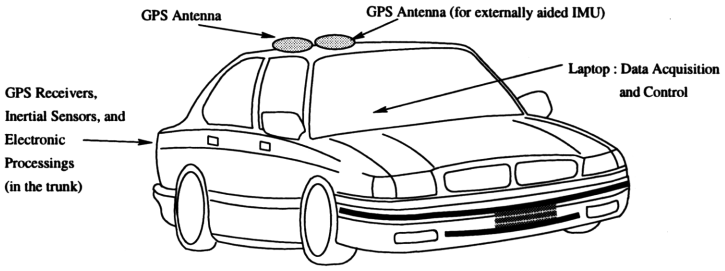

Figure 5-3: Single-Antenna GPS receiver and IMU on automobile for failure detection algorithm experiment

with the ground vehicle while not completely representative of the aircraft environment, is expected to be applicable to the aircraft problem. On the ground vehicle, the GPS antennas will suffer obstacles such as trees, roads, and tunnels. These problems will not be faced in aircraft application. The similar situation, however, might occur when an aircraft flies across mountains or hills, or when it turns so that the wings hinder the GPS receivers. The point here is that in both ground vehicle and aircraft, there will be a period when the GPS solution is not reliable due to some reasons. Hence, the failure detection performance on the ground vehicle experiment result should be useful for the aircraft application.

5.2 Experiment

5.2.1 Hardware Installation and Data Acquisition Process

The two GPS antennas were installed at the top of the automobile. One antenna was used by the NOVATEL GPS receiver, while the other was used by the MIGITS IMU. The IMU is externally aided, that it uses the GPS information to enhance the performance of the inertial measurements. The receivers, the inertial sensors, the electronic units were put in the automobile trunk. Each instrument was controlled by a portable laptop computer. The laptop also executed the data acquisition process. The software for data acquisition directed the measurements from both the GPS and the IMU and gave the following quantities :

The GPS measurements :

1. acquisition time : in ms
2. the GPS time
3. the GPS latency
4. velocities in North - East - Down : in $m.s^{-1}$
5. the GPS solution status
6. the GPS velocity status
7. geographic location : longitude, latitude, altitude : in ^{0}
8. HDOP
9. VDOP
10. number of satellites observable by the GPS receiver

The IMU measurements :

1. acquisition time : in ms
2. velocities in North - East - Down : in $m.s^{-1}$
3. attitude angles :θ, ϕ, ψ : in degree

4. geographic location : longitude, latitude, altitude : in degree

5. accelerations in body axes : a_x, a_y, a_z : in $m.s^{-2}$

6. angular rates : $\dot{\theta}$, $\dot{\phi}$, $\dot{\psi}$: in ${}^0.s^{-1}$

7. instrument status

The result of the measurements were stored in the laptop hard-disk. The data were then used for the failure detection experiment.

The IMU provided the measurements of accelerations from three orthogonal axes resolved in body axes, while the GPS measurements gave the velocities in three orthogonal axes resolved in North-East-Down (NED) axes. The transformation of a vector expressed in body axes to the one expressed in NED axes requires the direction cosine matrix whose entries are given by the IMU measurements. The relationship between measurements in this experiment can be stated in the following equation :

$$\underline{a}_t^b = C_{NED}^b \hat{\underline{a}}_t^{NED} \tag{5.1}$$

where

- $\underline{a}_t^b$: the acceleration measurements in body axes from IMU
- $\hat{\underline{a}}_t^{NED}$: the acceleration estimates from velocity measurements from the GPS receiver
- C_{NED}^b : the direction cosine matrix which transforms a vector in NED axes to a vector in body axes

The direction cosine matrix is given by :

$$\begin{aligned} C_{NED}^b &= \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \end{aligned} \tag{5.2}$$

where

- ψ : yaw angle
- θ : pitch angle
- ϕ : roll angle

and ψ, θ, and ϕ were obtained as outputs from the IMU. The C^b_{NED} matrix is an orthogonal matrix, so the inverse of C^b_{NED} is its transpose. The C^{NED}_b, given by

$$C^{NED}_b = \left[C^b_{NED}\right]^{-1} = \left[C^b_{NED}\right]^T \tag{5.3}$$

was also used in the experiment to compare the performance of the failure detection algorithms on either body axes or NED axes.

5.2.2 Failure Detection Algorithm Comparison

Using the data gathered from the IMU and the GPS measurements, a comparison of failure detection algorithms was performed as shown in Figure 5-4. Three algorithms were used to monitor the status of the instruments using the observation data. The observation data stream was segmented and put into the buffer of observation. Each algorithm performed a finite-horizon detection [31] which means that the failure detection algorithm searched for failure time τ^* over a finite interval $t_i \leq \tau^* \leq t_f$, where t_i corresponds to the first data within the buffer and t_f is the last data within the buffer of observation.

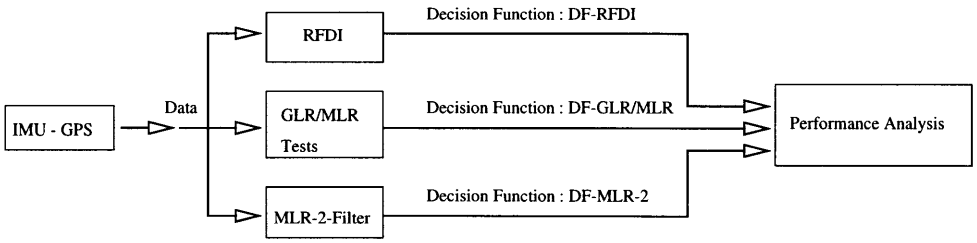

Figure 5-4: The structure of the experiment of failure detection algorithms with real measurments

There were two types of failure; *jump* failure and *ramp* failure, that were injected to the measurements of each axis. The injected failures had adjustable variables to observe the

sensitivity of the algorithms. For jump type failure, the sensitivity of each algorithm was tested against the magnitude of failure, while for ramp type failure, the slope of the failure was varied to test the sensitivity of the algorithms. Each algorithm performance in detecting these failures from the measurements in each axis was compared. Time of detection is used as a measure of detection capability of each algorithm. Undetected failure corresponds to the time of detection $t_D = \infty$. Figure 5-5 shows the scenario of the experiment.

In addition to the detection capability, computational complexity of each algorithm was also compared. The computational complexity refers to number of *flops - floating point operations* - used by each algorithm in performing finite-horizon failure detection.

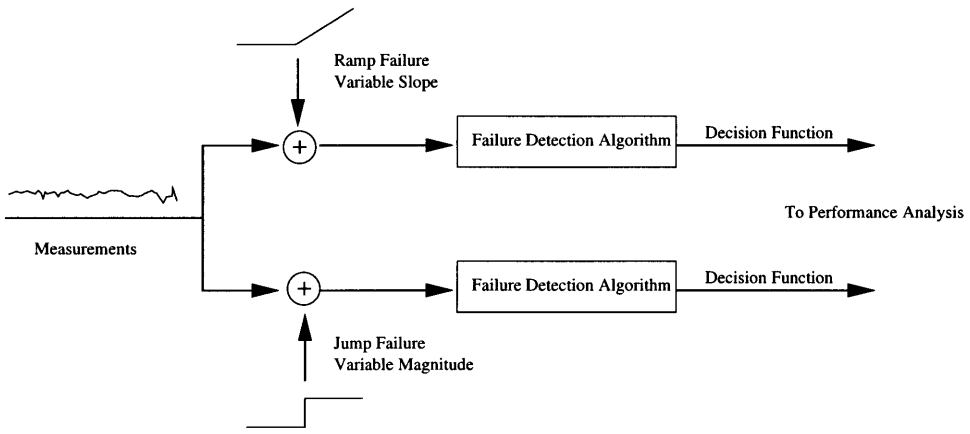

Figure 5-5: The scenario of experiment : failure injected measurements were analyzed by each failure detection algorithm

5.2.3 Measurement Data and Preprocessing

The ground track of the vehicle during the experiment is given in Figure 5-6 and the altitude variation of the vehicle is given in Figure 5-7. The experimental data was corrupted by noise and spikes due to the quality of the measurements and a hardware problem in data storage. The quality of measurements from the GPS receiver depends on the number of satellites observable by the receiver and the constellation of the satellites. If the number and geometry of satellites are not adequate, the receiver provides unreliable information as well as *warning* signals indicating reception problems. When the satellite constellation is such that satellites are not significantly separated in space, the receiver also gives a *warning* of *Geometric Dilution of Precision (GDOP)* status [16]. In addition the data recording

system also experienced problems which introduced occasional errors in the recorded IMU and GPS data. These problems could not be resolved during the time period available for data acquisition so these errors were removed from the data before the data was introduced into the failure detection algorithms. Figure 5-8 presents some typical raw measurement

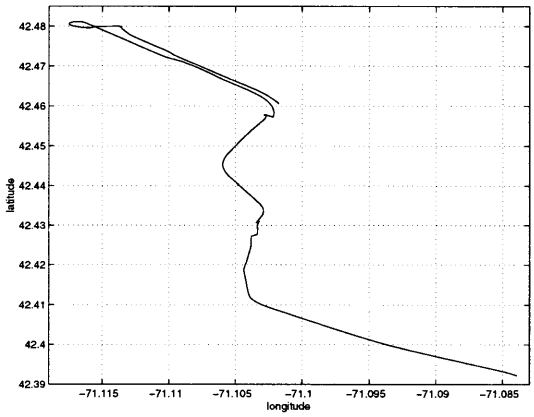

Figure 5-6: The gound track of the excursion used for the experiment

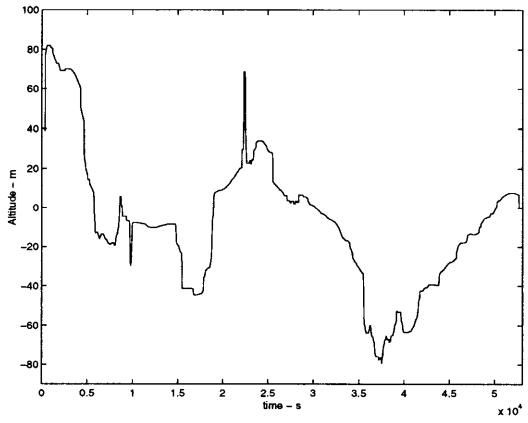

Figure 5-7: The altitude variation of the excursion for the experiment

data and the corresponding GPS warning indicators. As can be seen the velocity data contains both noise and spikes. At several points, the spikes occur at the same time as when the number of observable satellites decreased as well as when the HDOP status increases. HDOP, *Horizontal Dilution of Precision* indicates the variance of measurement in horizontal plane. Other spikes were due to the data storage problem mentioned earlier.

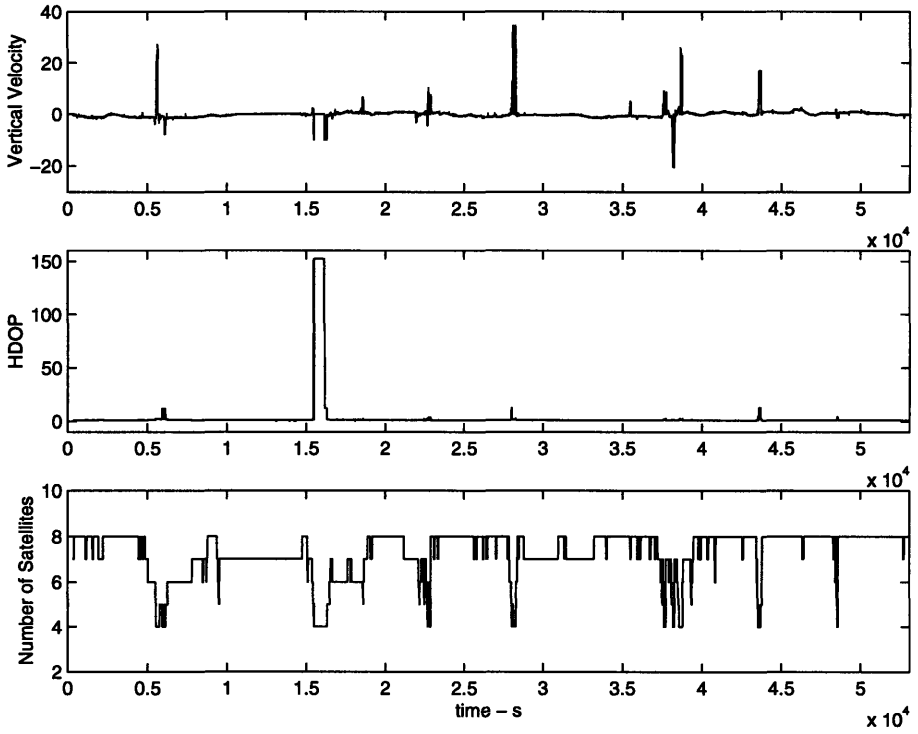

Figure 5-8: The typical raw measurement was corrupted by spikes in addition to noise

To remove the spikes in this preprocessing stage, the following algorithm was used :

$$\begin{aligned} d_i &= \begin{cases} d_i = d_i & \text{if } d_i - d_{i-1} < h_{threshold} \\ d_i = d_{i-1} & \text{otherwise} \end{cases} \\ i &= t_i, \ldots, t_f \end{aligned} \tag{5.4}$$

where d is the difference between the GPS and the IMU measurements, $h_{threshold}$ is the threshold used to remove the spikes. The idea of this algorithm is that two sets of measurements from the GPS and the IMU should not disagree significantly. When they do disagree significantly, then the spikes must have corrupted the measurements. It is not possible for the measurement difference to rise or to drop beyond the given $h_{threshold}$. If it does then the current measurement is replaced by the previous one. The change from d_{t-1} to d_t beyond the $h_{threshold}$ is associated with the presence of the spike. The $h_{threshold}$'s were chosen carefully so as to remove spikes but also so as not to reduce sensor noise.

While this is a significant problem for this experiment, there are two mitigating factors for aircraft applications. First, because aircrafts fly above trees and highway overpasses,

the errors due to GPS signal loss will be far fewer in an aircraft than was the case for this automobile experiment. Second, as the GPS constellation grows the HDOP and signal loss problems are likely to be eliminated.

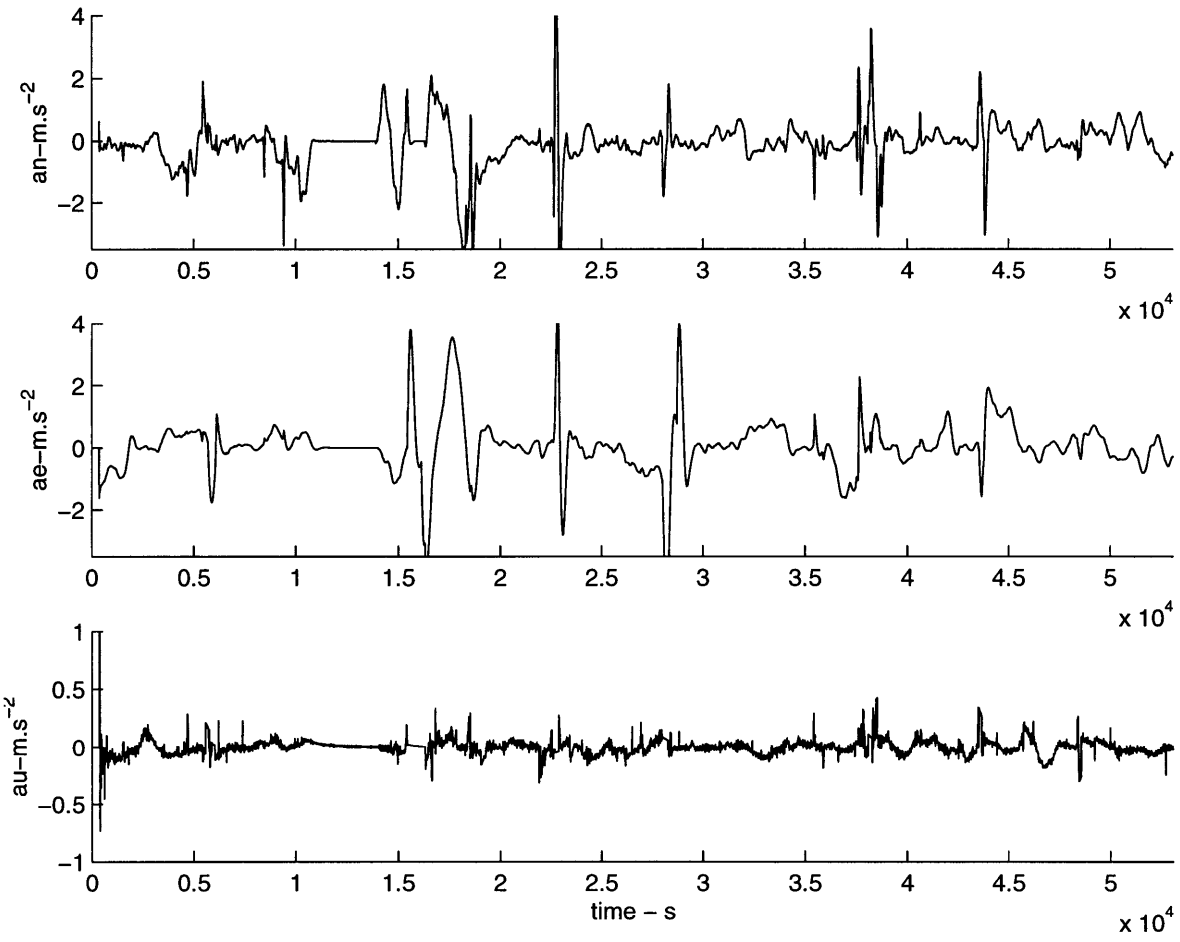

Figure 5-9: The estimated accelerations from the GPS velocity measurements

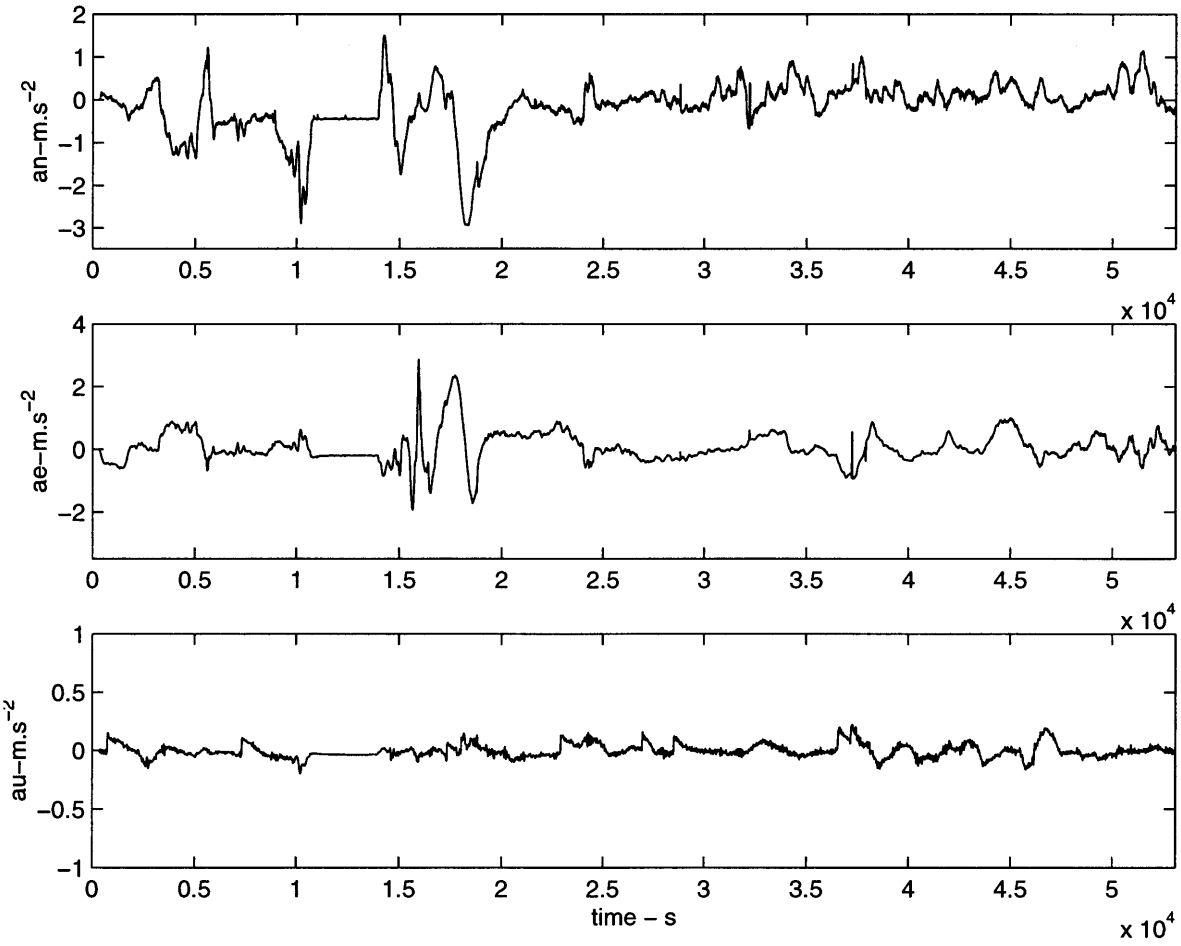

Figure 5-10: The accelerations from the IMU

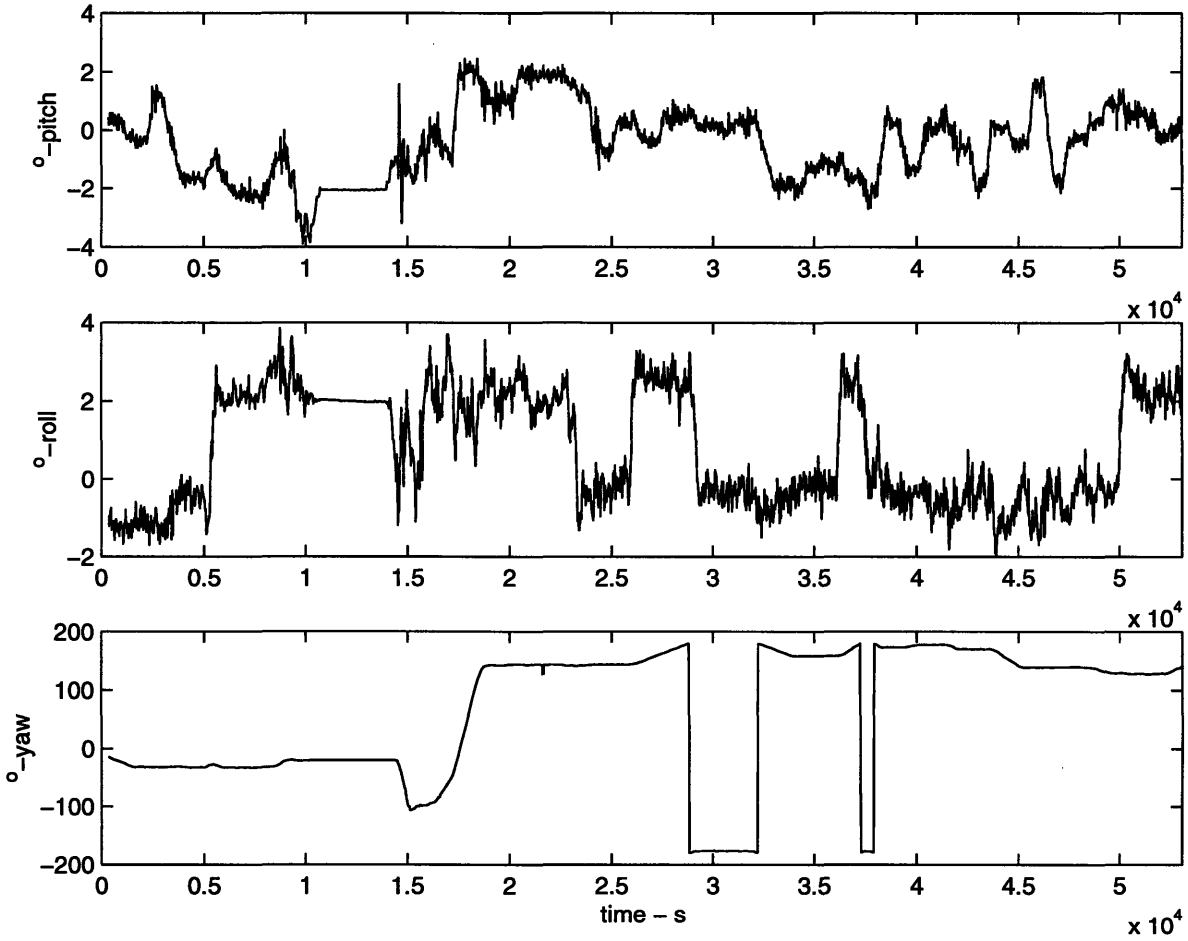

Figure 5-11: The attitude angles from the IMU

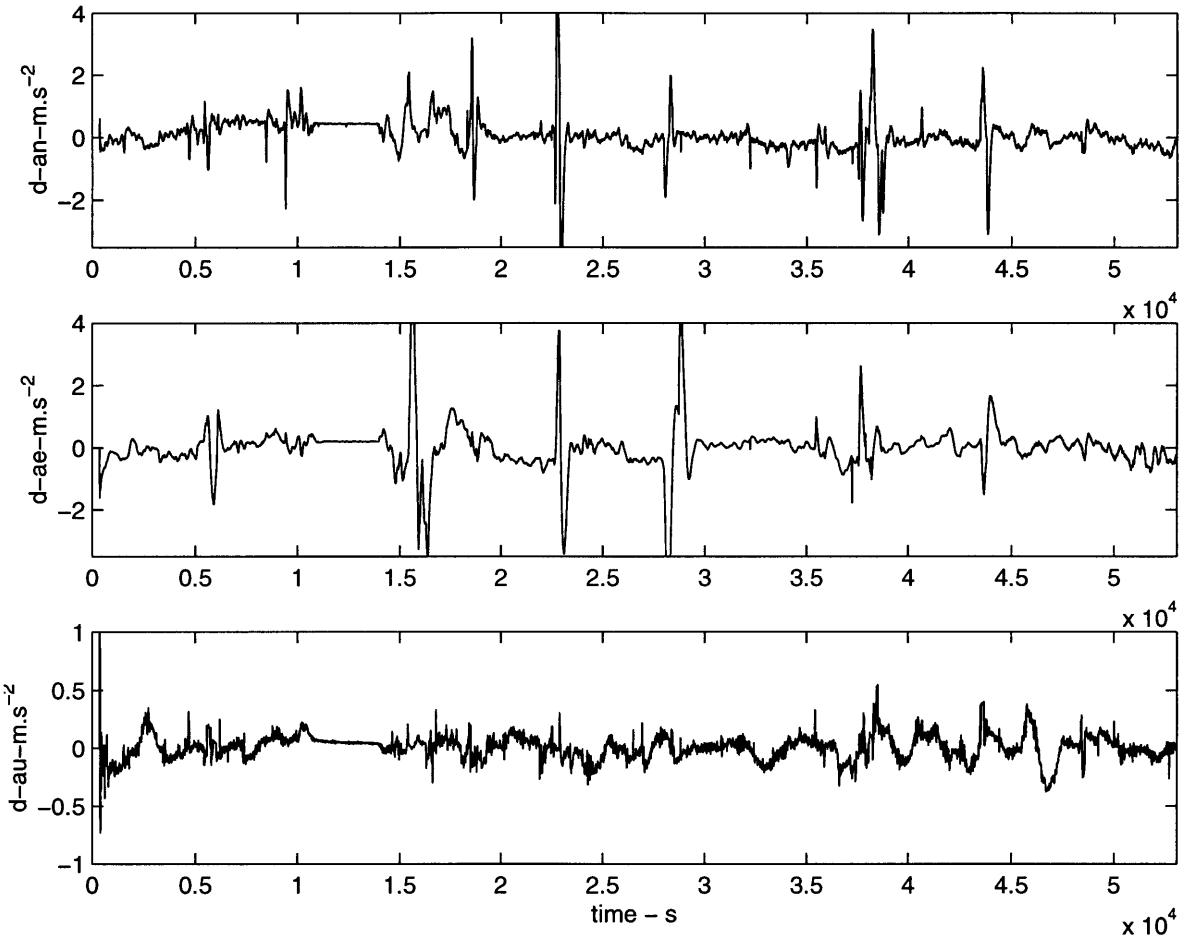

Figure 5-12: The difference of accelerations from the IMU and accelerations from GPS resolved in NED reference frame

The Kalman filters were developed to estimate accelerations from GPS velocities. Since the GPS measurement output is velocity in NED axes, three Kalman filters were used three components of vehicle accelerations. The following system model was used to derive the acceleration in each of the three axes :

$$x_{t+1} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_t + w_t \tag{5.5}$$

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} x_t + e_t \tag{5.6}$$

$$cov_N[w_t, w_t^T] = \begin{bmatrix} 3.33 \times 10^{-6} & 5.00 \times 10^{-5} \\ 5.00 \times 10^{-5} & 1.00 \times 10^{-3} \end{bmatrix} \tag{5.7}$$

$$cov_E[w_t, w_t^T] = \begin{bmatrix} 3.33 \times 10^{-7} & 5.00 \times 10^{-6} \\ 5.00 \times 10^{-6} & 1.00 \times 10^{-4} \end{bmatrix} \tag{5.8}$$

$$cov_D[w_t, w_t^T] = \begin{bmatrix} 3.33 \times 10^{-8} & 5.00 \times 10^{-7} \\ 5.00 \times 10^{-7} & 1.00 \times 10^{-5} \end{bmatrix} \tag{5.9}$$

$$cov_N[e_t, e_t^T] = 10^{-2} \tag{5.10}$$

$$cov_E[e_t, e_t^T] = 10^{-3} \tag{5.11}$$

$$cov_D[e_t, e_t^T] = 10^{-4} \tag{5.12}$$

$$cov_N[x_0, x_0^T] = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{5.13}$$

$$cov_E[x_0, x_0^T] = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{5.14}$$

$$cov_D[x_0, x_0^T] = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{5.15}$$

$$x_{N_0} = [10 \quad 10]^T \tag{5.16}$$

$$x_{E_0} = [10 \quad 10]^T \tag{5.17}$$

$$x_{D_0} = [10 \quad 10]^T \tag{5.18}$$

where

- $dt = 100$ ms
- $x_t = [v_t \quad a_t]^T$
- $cov_i[w_t, w_t^T], i = N, E, D$: covariance matrix of process noise in i axis
- $cov_i[e_t, e_t^T], i = N, E, D$: covariance matrix of measurement noise in i axis
- $cov_i[x_0, x_0^T], i = N, E, D$: covariance matrix of initial states in i axis.
- $x_{i_0}, i = N, E, D$: initial states of each axis i.

The accelerations given by the Kalman filters are shown in Figure 5-9. Also, acceleration outputs in body coordinates from the IMU were transformed to NED axes using the C_b^{NED} matrix obtained from roll, pitch and yaw indicator from the IMU. The acceleration outputs from the IMU resolved in NED axes is shown in Figure 5-10. The attitude angles obtained from the IMU are presented at Figure 5-11. The difference of the two sets of measurements of vehicle acceleration is presented in Figure 5-12.

The estimated acceleration from GPS velocity measurements contained spikes at several points in time. These points correspond to the times when the quality of the GPS measurements was poor. The receiver itself generated signals indicating that the measurements were of low quality. In Figure 5-13, 5-14, and 5-15 we see that there were significant differences between the two sets of measurements when the GPS soulutions were poor. These signals are :

1. Solution status.

   The solution status is zero when the GPS provides a good solution. When it is one, the number of satellites observable is not adequate. Higher number of solution status indicates worse quality of the GPS solution, for example, solution status of 4 means that the trace of the error covariance matrix exceeds 1000 meters.

2. HDOP.

   HDOP, *horizontal dilution of precision*, indicates the error variance in the horizontal plane.

3. VDOP

   VDOP, *vertical dilution of precision*, indicates the error variance in the vertical axis.

4. Number of satellites.

   GPS requires at least four satellites to provide an accurate solution.

These signals will be essential in providing logic for effective decision functions. The method will be discussed in the sequel.

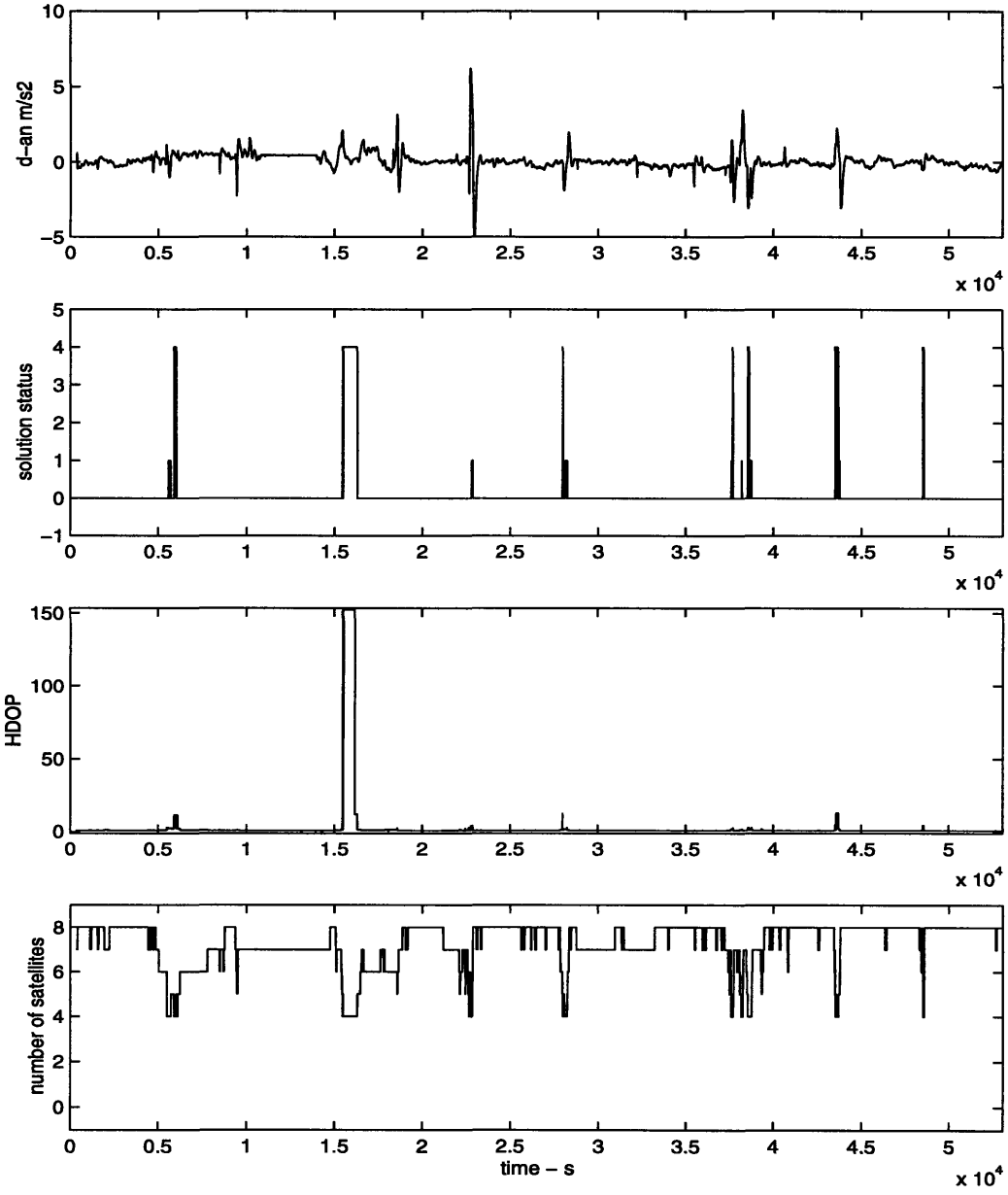

Figure 5-13: Measurement differences and low quality GPS signal indicators for north acceleration

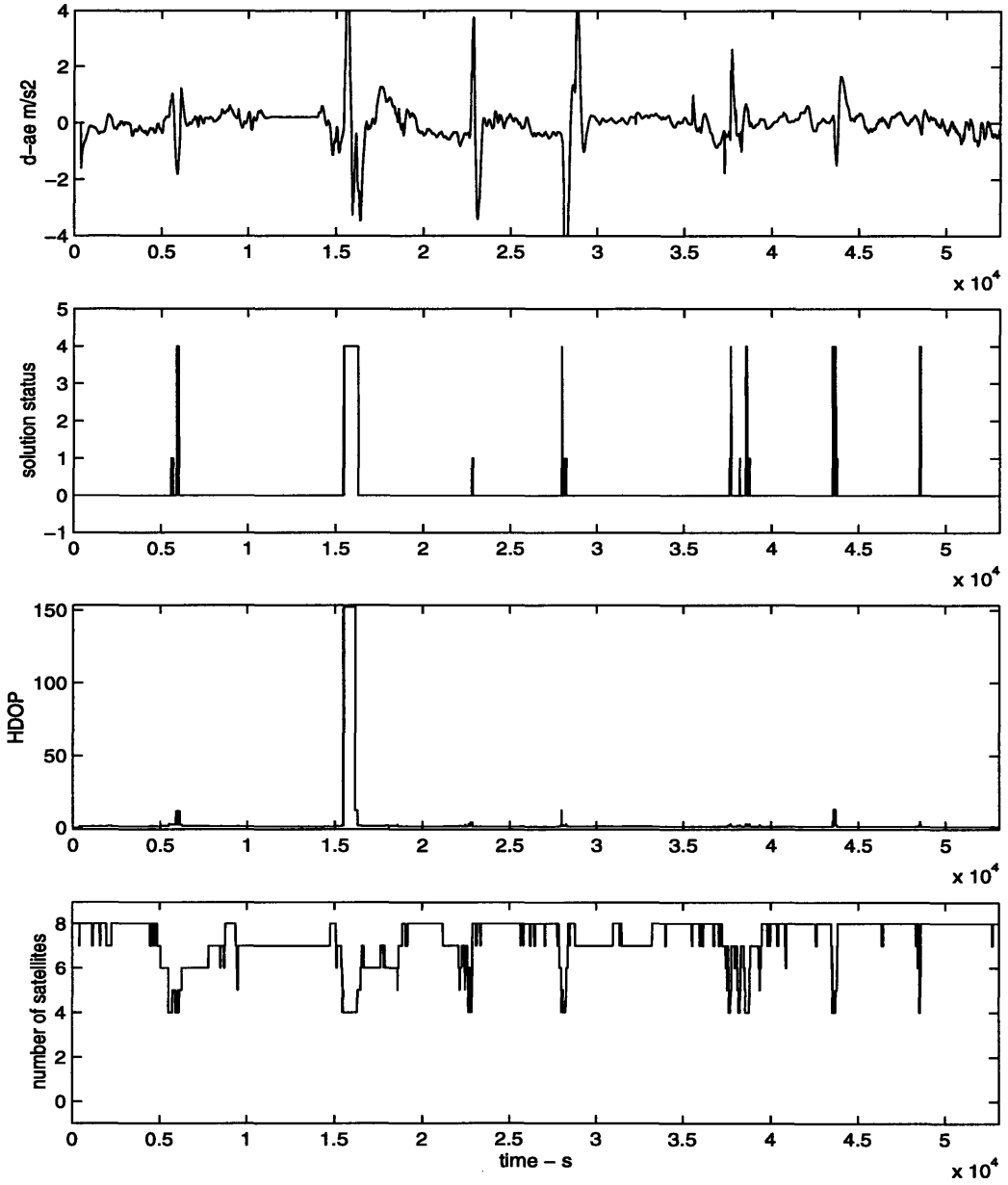

Figure 5-14: Measurement differences and low quality GPS signal indicators for east acceleration

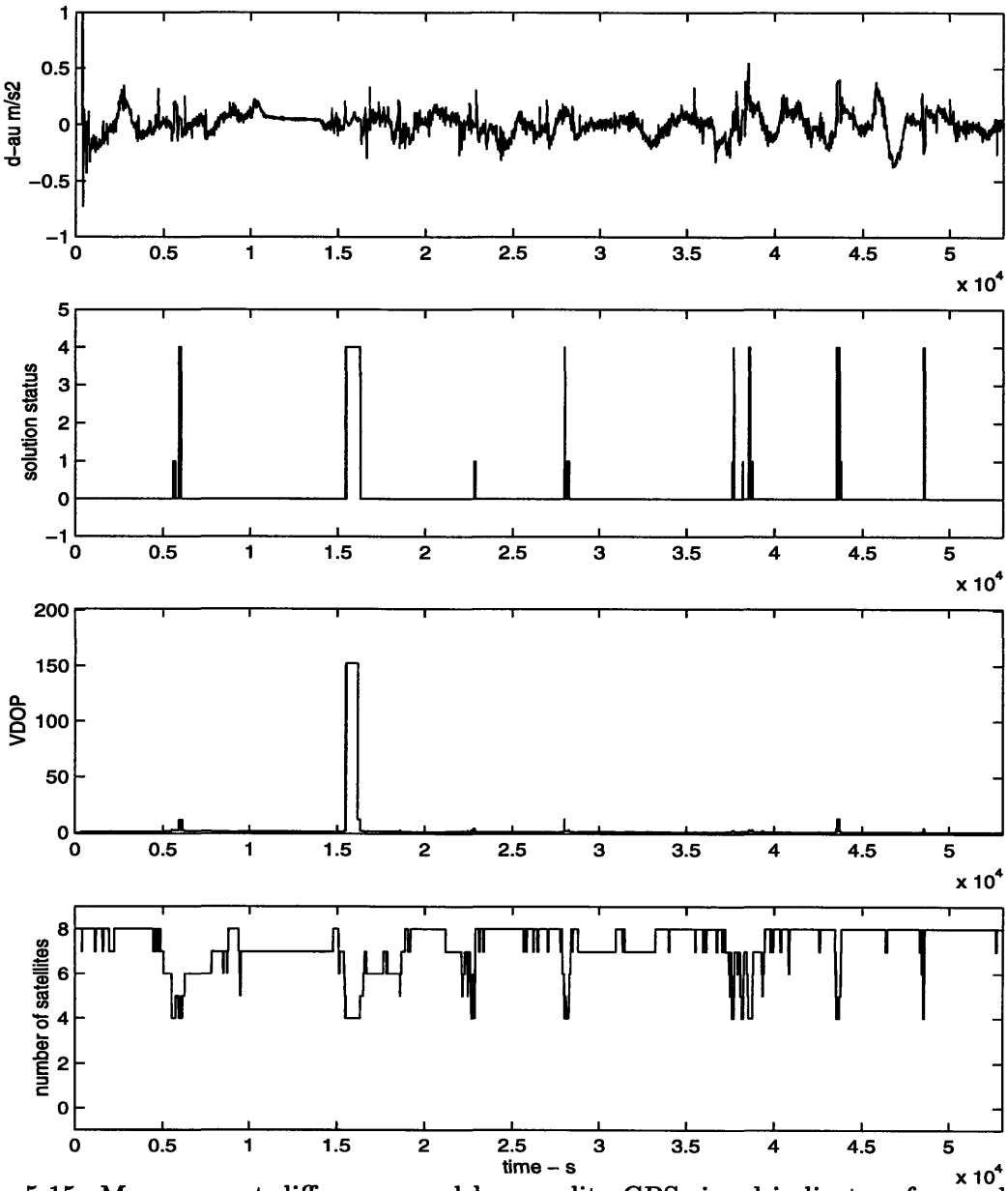

Figure 5-15: Measurement differences and low quality GPS signal indicators for up-down acceleration

5.2.4 Failure Detection Algorithm Development

Failure detection problems in navigation given in the research works of Labarrere [29], Nikiforov [36], for instance, used either error model of instruments or dynamic model of vehicles. These models, expressed in state-space equations, are eventually fit into the failure detection frameworks presented in the previous chapters. In this experiment, error models of both the GPS and the IMU were not used.

The failure detection algorithms used in this experiment required a state-space model

of the observations. The state variables used were *the difference between the acceleration indications* obtained from the GPS and the IMU measurement resolved in NED reference frame as shown in Figure 5-12. The state-space model for the unfailed condition is given as

$$x_{t+1} = x_t + w_t \tag{5.19}$$

$$y_t = x_t + e_t \tag{5.20}$$

$$x_t = \underline{a}_t^{IMU} - \hat{\underline{a}}_t^{GPS} \tag{5.21}$$

$$cov[x_0, x_0^T] = 1 \tag{5.22}$$

$$cov[w_t, w_t^T] = 1 \times 10^{-4} \tag{5.23}$$

$$cov[e_t, e_t^T] = 0.1050 \tag{5.24}$$

The Development of GLR/MLR Test. The GLR test in this experiment used the system model of equations 5.19- 5.24. A recursive approach was used in computing the failure signature with :

$$\varphi_{t+1}^T(k) = 1 - \mu_t(k) \tag{5.25}$$

$$\mu_{t+1}(k) = \mu_t(k) + K_{t+1}\varphi_t^T(k) \tag{5.26}$$

$$\varphi_k(k) = 0 \tag{5.27}$$

$$\mu_k(k) = 0 \tag{5.28}$$

were used to compute

$$f_N(k) = \sum_{t=t_i}^{N} \varphi_t(k) S_t^{-1} \varepsilon_t. \tag{5.29}$$

The decision function was given by

$$l_N(k, \hat{\nu}(k)) = f_N^T(k)\hat{\nu}(k) \tag{5.30}$$

where $\hat{\nu}(k)$ was calculated recursively by the following equations :

$$\hat{\nu}_t = \hat{\nu}_{t-1} + L_t \left[\varepsilon_t - \varphi_t^T \hat{\nu}_{t-1}\right] \tag{5.31}$$

$$L_t = P_{t-1}^{\nu}\varphi_t(k) \left[\varphi_t^T(k) P_{t-1}^{\nu} \varphi_t(k) + 1\right]^{-1} \tag{5.32}$$

$$P_t^{\nu} = P_{t-1}^{\nu} - L_t \varphi_t^T(k) P_{t-1}^{\nu} \tag{5.33}$$

$$P_{t_i}^{\nu} = 1. \tag{5.34}$$

Note that S_t is the innovation variance and K_t is the Kalman gain matrix. The quantity ν is the magnitude of a failure assumed under the H_1 hypothesis ($\nu = 0$ under the H_0 hypothesis). Thus

$$\begin{aligned} \hat{x}_{t|t}(k) &= \hat{x}_{t|t} + \mu_t(k)\nu \\ \varepsilon_{t|t}(k) &= \varepsilon_{t|t} + \varphi_t^T(k)\nu \end{aligned}$$

where $\hat{x}_{t|t}$ and $\varepsilon_{t|t}$ are the quantities obtained from the Kalman filter under the hypothesis of no failure. The GLR test then seeks the possible time of failure τ^* by searching over $k, t_i < k < t_i + N$ where N is the size of the window of observation, t_i is the initial time, and k is the index that refers to the specific data within the window.

The MLR test for on-line application, as we saw in Section 3.3 and 3.5, relies on the result of the GLR test. The detection function of the on-line MLR test will only amplify the decision function of the GLR test under the assumption of Gaussian distributed failure. Hence, if the GLR test performs well, the MLR will give good performance as well.

The Development of 2-filter MLR Test. The MLR test with two filters required two Kalman filters : one running forward and the other running backward in time. The following filters were used :

Forward Kalman filter : $t = t_i, \ldots, t_i + N$

$$\varepsilon_t^F = y_t - \hat{x}_t^F \tag{5.35}$$

$$S_t^F = P_t^F + 0.3742 \tag{5.36}$$

$$\hat{x}_{t+1}^F = \hat{x}_t^F + P_t^F \left[S_t^F\right]^{-1} \varepsilon_t^F \tag{5.37}$$

$$P_{t+1}^F = \left[I - P_t^F S_t^{-F}\right] P_t^F + 1.00 \times 10^{-4} \tag{5.38}$$

$$\mathcal{D}_{\mathcal{F}_{t+1}^F} = \mathcal{D}_{\mathcal{F}_t^F} + \ln|S_t^{-F}| - \ln(2\pi) - \varepsilon_t^{F^T} S_t^{-F} \varepsilon_t^F \tag{5.39}$$

with the following innitial conditions

- $P_0^F = 1$

- $x_0^F = 0$

Backward Kalman filter : $t = t_i + N, \ldots, t_i$

$$\varepsilon_t^B = y_t - \hat{x}_t^B \tag{5.40}$$

$$S_t^B = P_t^B + 0.3742 \tag{5.41}$$

$$\hat{x}_{t-1}x^B = \hat{x}_t^B + P_t^B \left[S_t^B\right]^{-1} \varepsilon_t^B \tag{5.42}$$

$$P_{t-1}^B = \left[I - P_t^B S_t^{-B}\right] P_t^B + 1.00 \times 10^{-4} \tag{5.43}$$

$$\mathcal{D}_{\mathcal{F}_{t-1}^B} = \mathcal{D}_{\mathcal{F}_t^B} + \ln|S_t^{-B}| - \ln(2\pi) - \varepsilon_t^{B^T} S_t^{-B} \varepsilon_t^B \tag{5.44}$$

with the following initial conditions :

- $P_0^B = 1$
- $x_0^B = 1$

The decision function was given by the following equation :

$$l_t(k, \hat{\nu}(k)) = \mathcal{D}_{\mathcal{F}_t^F} + \mathcal{D}_{\mathcal{F}_t^B} \tag{5.45}$$

As it was illustrated through numerical example in Section 3.5, the decision function l_t should be normalized so that a common threshold for failure detection can be applied regardless the time of failure. The normalization process is described by the following equation :

$$\bar{l}_t(k, \hat{\nu}(k)) = l_t(k, \hat{\nu}(k)) + \inf_{t_i \leq t \leq t_f} l_t(k, \hat{\nu}(k)) \tag{5.46}$$

where t_i is the initial data within the buffer and t_f is the last data within the buffer of observation.

The measurements are divided into P segments, each of length N. When the 2-filter MLR algorithm is applied on-line, there will be a delay of detection of N. After segment-k, the backward filter is started at time $k \times N$, while the forward filter starts processing segment-$(k+1)$. When the forward filter reaches the end of segment-$(k+1)$ at time $(k+1) \times N$, the backward filter reaches the beginning of segment-k at time $(k-1) \times N + 1$ and the decision function of segment-k is available. Thus, this algorithm exhibits a delay of one window of observation due to requirement of running the backwards filter.

In the following sections, this algorithm will be referred as the MLR-2 test.

The Development of RFDI. It was described in Section 4.3.3 that to develop the RFDI, we need to develop an estimator for an augmented system. A failure state is augmented to system states. The RFDI algorithm used in the experiment assumed that the system model was known. Hence, as it was described in Section 4.3.1 and illustrated through numerical example in Section 4.4 Case 1, there was only one Riccati equation to solve. The system, described by the state-space model of Equations 5.19 and 5.24, was assumed to be driven by the noise with unknown model yet bounded in l_2 setting. The augmented system is described by the following state-space description :

$$x_{t+1} = Ax_t + Br_t \tag{5.47}$$

$$y_t = Cx_t + Dr_t \tag{5.48}$$

$$z_t = Mx_t \tag{5.49}$$

where

- $x_t = [d_{a_t} \quad \nu_t]^T$ is the augmented states, d_{a_t} is the differences between the acceleration indications obtained from the GPS and the IMU measurements resolved in NED axes, and ν_t is the failure state.

- $r_t = [w_t \quad e_t]^T$ is the noise vector, w_t is the process noise, and e_t is the measurement noise.

- The matrices used in the Equations (5.47) - (5.49) are :

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$M_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For estimator design purpose, the following quantities were used :

$$\begin{aligned} \gamma &= 4 \\ x_0 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \\ P_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

For the dynamic system described by the Equations (5.47) - (5.49), the RFDI is given by :

$$H_t = P_t^{-1} - \gamma^{-2} M_t^T M_t \tag{5.50}$$

$$K_t = \left[BD^T + AH_t^{-1}C^T\right] \cdot \left[DD^T + CH_t^{-1}C^T\right]^{-1} \tag{5.51}$$

$$\breve{A} = A - K_t C \tag{5.52}$$

$$\breve{B} = B - K_t D \tag{5.53}$$

$$P_{t+1} = \breve{A} H_t^{-1} \breve{A}^T + \breve{B}\breve{B}^T \tag{5.54}$$

$$\hat{x}_{t+1} = \breve{A}\hat{x}_t + K_t y_t \tag{5.55}$$

The decision function for the RFDI is

$$l_t = sign(\hat{\nu}_t) \cdot \hat{\nu}_t \cdot P_{22_t}^{-1} \cdot \hat{\nu}_t \tag{5.56}$$

where

$$sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x \leq 0 \end{cases}$$

The thesis of Mangoubi [31] does not include the $sign(\hat{\nu}_t)$ factor in the decision function of the RFDI filter. This term is proposed in this thesis for failure detection of the GPS application. The test results showed that this factor suppressed the transient response significantly leading to a reduction in false alarms.

5.3 Performance Analysis

We saw in the Section 5.2.2, in the preprocessing stage, that the *measurement-difference-states* changed significantly when the quality of the GPS solution was poor. Even though

the Kalman filters in the preprocessing stage are successful in creating white processes when the GPS signals have high quality, they do not suppress the large errors that result from low quality GPS solutions. The measurement quality of the GPS affected the performance of the failure detection algorithms. The variance of measurement statistics when the GPS solution was poor was large enough to be considered as failure by each failure detection filter. From the experiment with the IMU and the GPS measurements, there are two cases that should be considered here.

Case 1 : Low quality GPS solutions. As we saw in Figures 5-13, 5-14, and 5-15, the difference between the two sets of measurements deviate significantly from zero when the GPS receiver has a low quality solution.

We will show the decision function of each algorithm for each axis when the GPS solution is poor. During a period when the GPS signals have poor quality, the failure detection algorithms do not respond very well to the injected failures. The decision function of each algorithm increases rapidly when the GPS signal quality degrades.

Consider the time interval that includes $t = 3.73 \times 10^4$ s. At $t = 3.73 \times 10^4$ s, the solution status is 4, which means that the trace of the error covariance matrix exceeds 1000 meters. The differences between the measurements obtained from the GPS and the IMU measurements especially in N and E axes differed significantly from zero within this interval. Suppose that both jump failure and ramp failure were injected at $t = 3.60 \times 10^4$ s. The ramp injected failure had the slope of 0.001 $m.s^{-3}$, which means that for every 0.1 s the failure increases 0.01 $m.s^{-2}$, the magnitude of jump failure was 0.3 $m.s^{-2}$. For the observation buffer with a length of 10 s, the decision functions in NED axes are shown in Figures 5-13, 5-14, and 5-15.

The decision function of each algorithm for North axis is shown in Figure 5-13. Each algorithm did not respond to the injected failures at $t = 3.60 \times 10^4$ s. The decision function of each algorithm, however, increased rapidly after $t = 3.73 \times 10^4$ s. The decision function of each algorithm for both types of failures was the same. The measurement differences during this interval hindered the real failures. The injected failures were undetected.

The failure detection process on the East axis also failed. Each algorithm showed that the failures occured at $t = 3.73 \times 10^4$ s. The GLR/MLR test and the MLR-2 test decision functions increased at $t = 3.60 \times 10^4$ s, the time when failures were injected, but they are

less dominant compared to the decision functions of the rapid dynamics due to the poor GPS solutions.

On the vertical axis, the GLR/MLR test and the MLR-2 test showed through their decision functions that there were failures at $t = 3.60 \times 10^4$ s. As in the case of the East axis, the decision functions increased significantly at $t = 3.73 \times 10^4$ s. Thus, rapid changes were interpreted as failures by the failure detection algorithms.

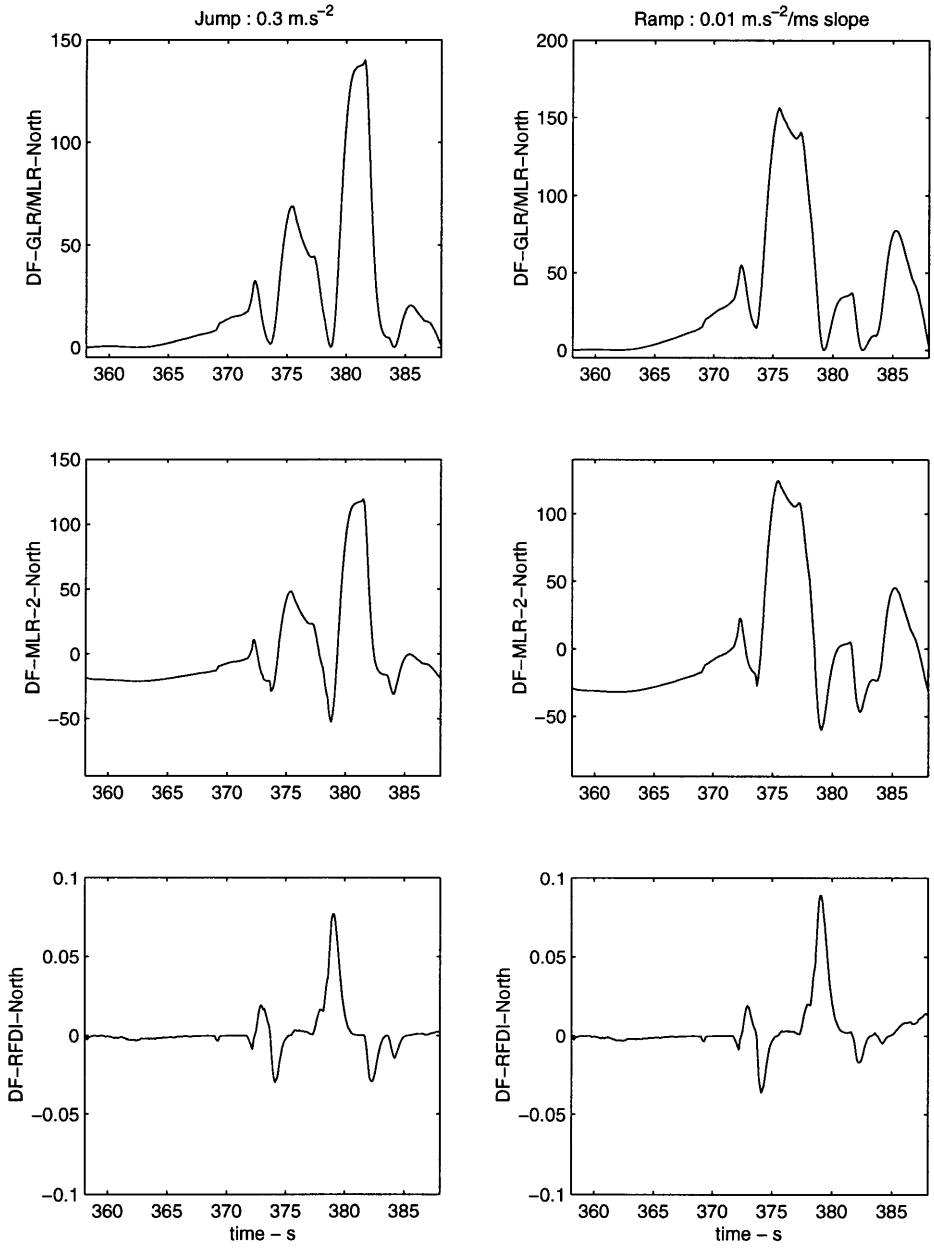

Figure 5-16: The false decision functions on North axis

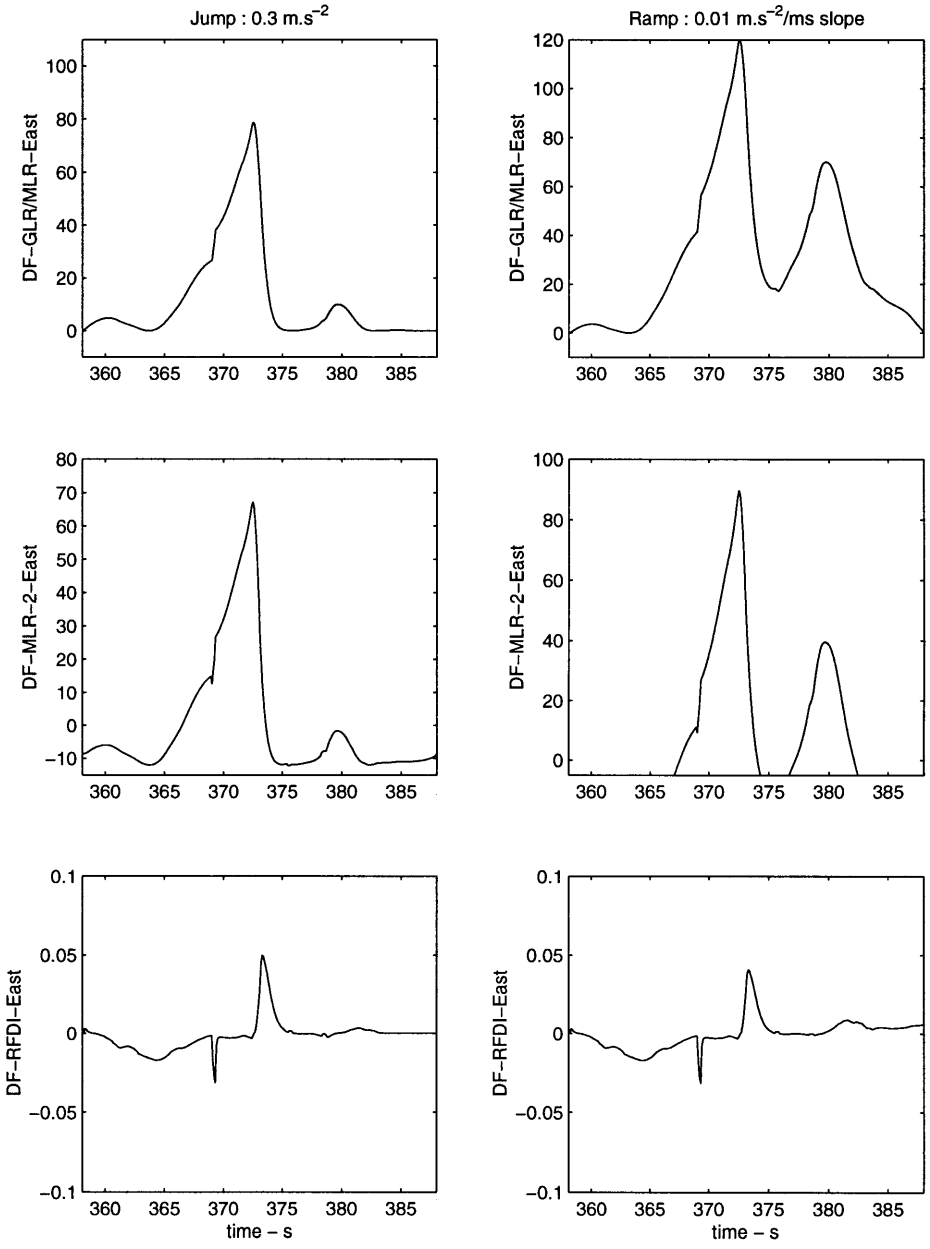

Figure 5-17: The false decision functions on East axis

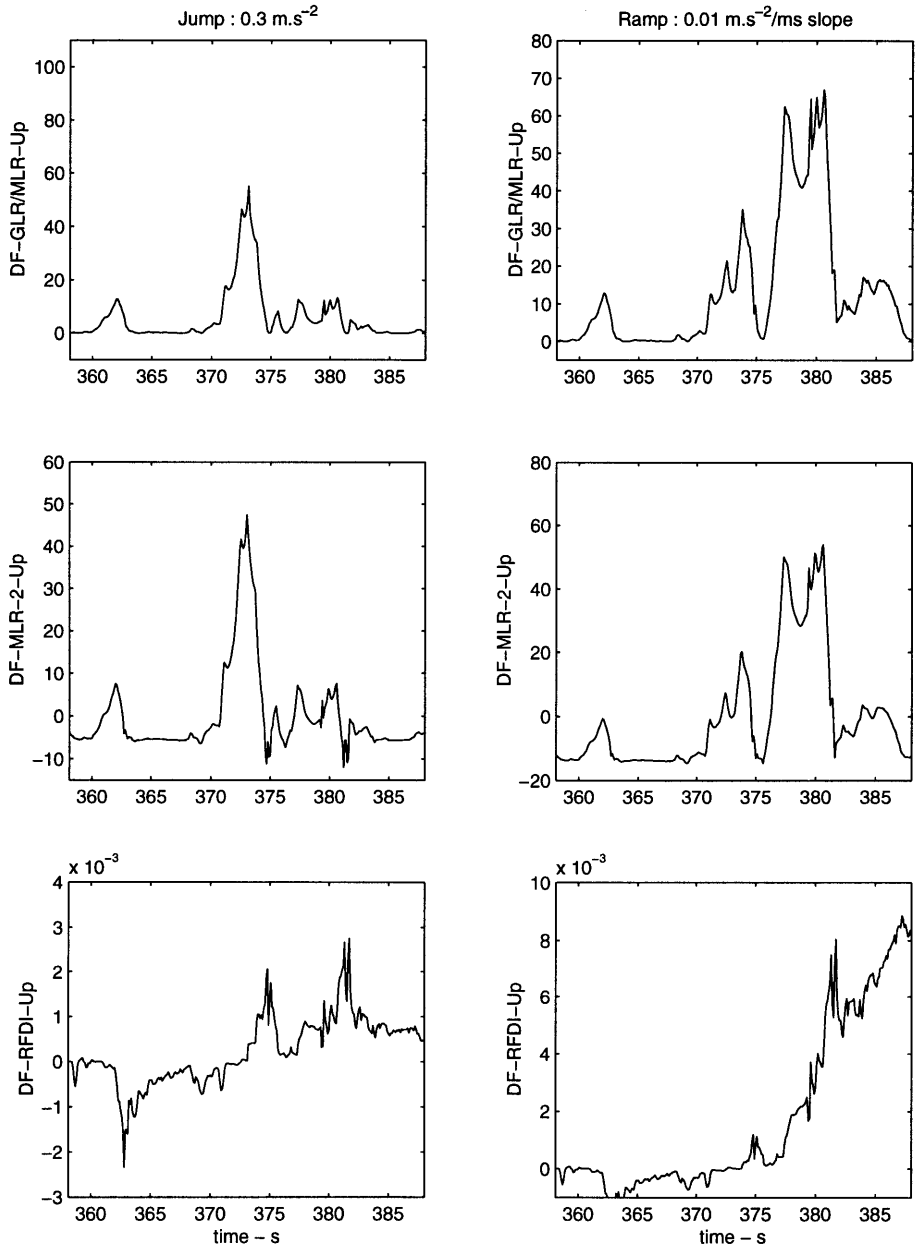

Figure 5-18: The false decision functions on Down/vertical axis

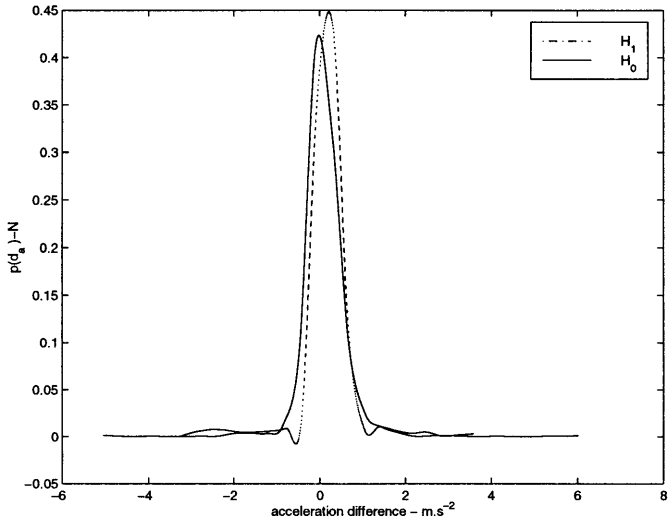

Figure 5-19: The statistics of the observations on North axis

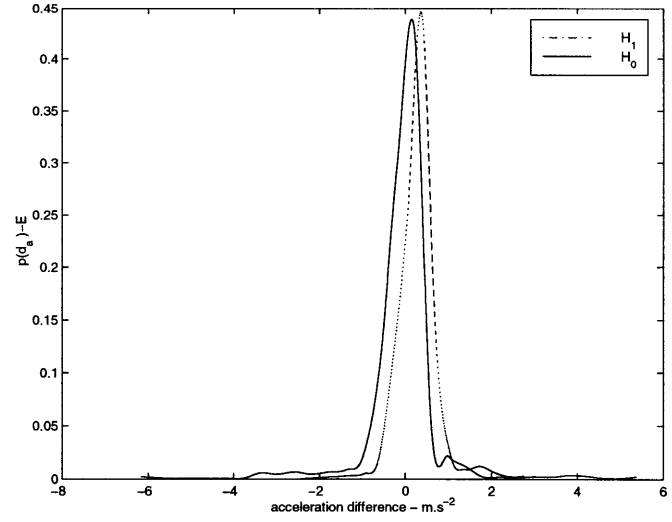

Figure 5-20: The statistics of the observations on East axis

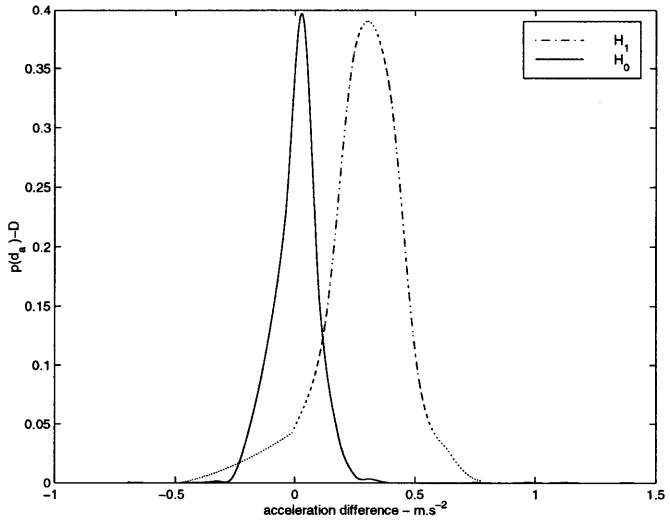

Figure 5-21: The statistics of the observations on Down/vertical axis

The statistics of the measurements before and after the jump also describes why the failures were undetected. Histograms of the measurements on each axis are given in Figures 5-19, 5-20, and 5-21. The solid lines indicate the sample density of the measurement before failure, while the dash-dotted lines refer to the sample density after the ump failure. Except for measurements on the vertical axis, the sample distribution densities of normal(H_0) and failed(H_1) hypotheses were not well separated. These quasi overlapped distributions correspond to the equality of probability of detection and the probability of false alarm.

Performance Analysis Tool : Receiver Operating Characteristic. In signal detection theory, there is an important tool called *ROC - receiver operating characteristic* that illustrates the trade-off between the probability of detection and the probability of false alarm for a receiver. This tool will be used to analyze the performance of the failure detection algorithms.

In Figures 5-19, 5-20, and 5-21, there are two sample distribution densities : $p(y|H_0)$ and $p(y|H_1)$. For each axis, the probability of of detection, p_D, and the probability of false alarm, p_{FA}, are calculated based on the following equations :

$$p_D = \int_{\eta}^{\infty} p(d_a|H_1)d(d_a) \tag{5.57}$$

$$p_{FA} = \int_{\eta}^{\infty} p(d_a|H_0)d(d_a) \tag{5.58}$$

where η is the threshold that divides the decision function into two mutually exclusive decisions : H_0 and H_1. For a particular η, there are associated p_D and p_{FA}. The ROC is constructed by plotting the p_D and p_{FA} as we vary the threshold η.

The ROC has the following properties [46] :

- all continuous LLR have ROC's that are concave downward
- all continuous LLR have ROC's that are above the $p_D = p_{FA}$ line
- the slope of a curve in a ROC at a particular point is equal to the value of the threshold η required to achieve the p_D and p_{FA} of that point.

There are three significant points in the ROC :

1. $(p_D = 0, p_{FA} = 0)$, that corresponds to $\eta = \infty$,
2. $(p_D = 1, p_{FA} = 0)$,
3. $(p_D = 1, p_{FA} = 1)$, that corresponds to $\eta = 0$.

The ideal location in the ROC is the $(p_D = 1, p_{FA} = 0)$ point which means that the selected threshold will give no false alarm but failure detection only. If it is not possible to operate the system in this ideal point, we need to choose the threshold so that the operating point in the ROC is as close as possible to the ideal point. Thus, we will have the system with maximum probability of detection and minimum probability of false alarm.

We are now ready to analyze the statistics of the measurement with ROC. The ROC's of the measurements on NED axes, shown in Figure 5-22, have the following properties :

- as we increase jump magnitude from 0.3 $m.s^{-2}$ to 0.7 $m.s^{-2}$, the points of each ROC approach the $(p_D = 1, p_{FA} = 0)$.
- When the jump magnitude is 0.3 $m.s^{-2}$, the ROC's of the measurements on N and E axes almost lie on the line $p_D = p_{FA}$, which means that we will always have the probability of detection equal to the probability of false alarm regardlesss of the choice of η. Thus, it is difficult to detect the failure with the magnitude 0.3 $m.s^{-2}$.

- The ROC on the D axis, in the contrary, shows that it is possible to detect the jump failure with the magnitude 0.3 $m.s^{-2}$. On this axis, it is possible to detect the failure with $p_D > 0.9$, $p_{FA} < 0.05$.

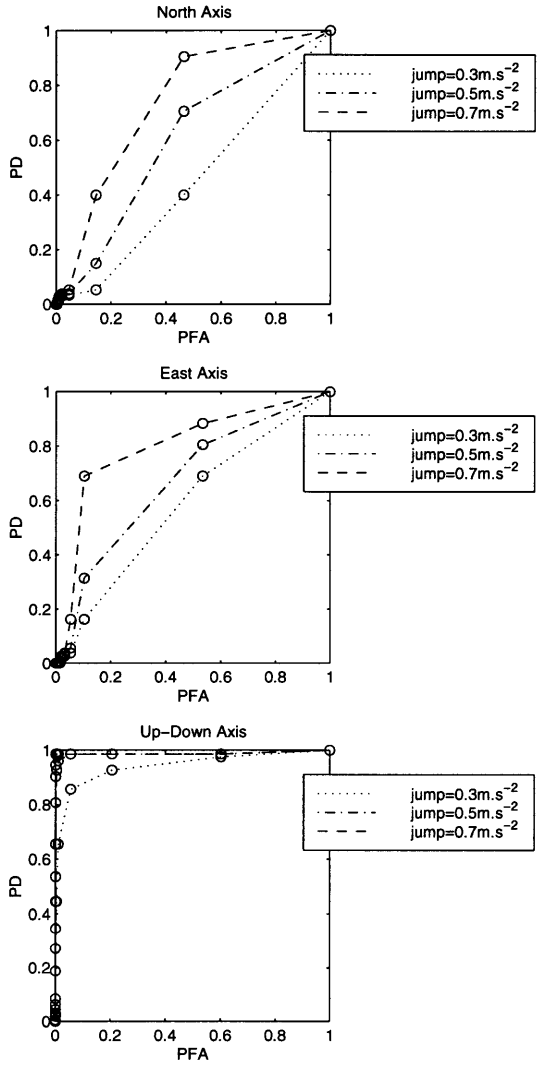

Figure 5-22: The ROC's of the observations in NED axes

Case 2: High quality of the GPS solutions. In this section, we will present the performance of the failure algorithms when the quality of the GPS signal is good. It will be shown that the failure detection algorithms can identify the presence of failures even if the jump failure magnitude is small and the ramp failure slope has a small gradient.

Consider the measurement interval that includes the point $t = 3.00 \times 10^4$ s. Suppose that we run the failure algorithms from time $t = 2.85 \times 10^4$ s to $t = 3.15 \times 10^4$ with buffer length of 100. Hence we have 300 data points and divide those measurements into 3 segments, each having a length of 100. Two types of failures, jump failure and ramp failure were injected at $t = 3.00 \times 10^4$ s. The jump failure has a magnitude 0.3 $m.s^{-2}$, while the ramp failure has a slope 0.001 $m.s^{-3}$. The failure detection algorithm performances are shown in Figures 5-23, 5-24, and 5-25.

We see in each of Figures 5-23, 5-24, and 5-25 that the algorithms detect the presence of failures. Each algorithm even gives significant alarm when the ramp failure occurs.

In Figure 5-23, all the algorithms detect the jump failure at about the same time, at $t = 3.00 \times 10^4$ s. The decision function of the MLR-2 increases in the beginning and in the end of the detection process, but it provides a smooth transition from one window of observation to the next (The transitions occur at $t = 2.95 \times 10^4$ s and at $t = 3.05 \times 10^4$ s). The effect of transient response in the RFDI decision function is forced to go below zero due to the $\mathbf{sign}(\hat{\nu})$ factor and its magnitude is diminished due to the P_{22}^{-1} factor. The decision function of RFDI in detecting the ramp failure, however, does not respond as fast as those of the GLR/MLR test and the MLR-2 test.

In Figure 5-24, each algorithm detects the jump failure on East axis at $t = 3.00 \times 10^4$ s. The ramp failure can also be detected by the algorithms. The RFDI decision function, however, exhibits delay in detecting ramp failure.

In Figure 5-25, we see the same detection time of each algorithm. The RFDI decision function stays at a certain constant value after the detection, while the other algorihtms; the GLR/MLR and the MLR-2 decision functions, return to zero after detection. The Kalman filters in these two statistical detection algorithms adjust themselves so that the estimated outputs follow the measurements.

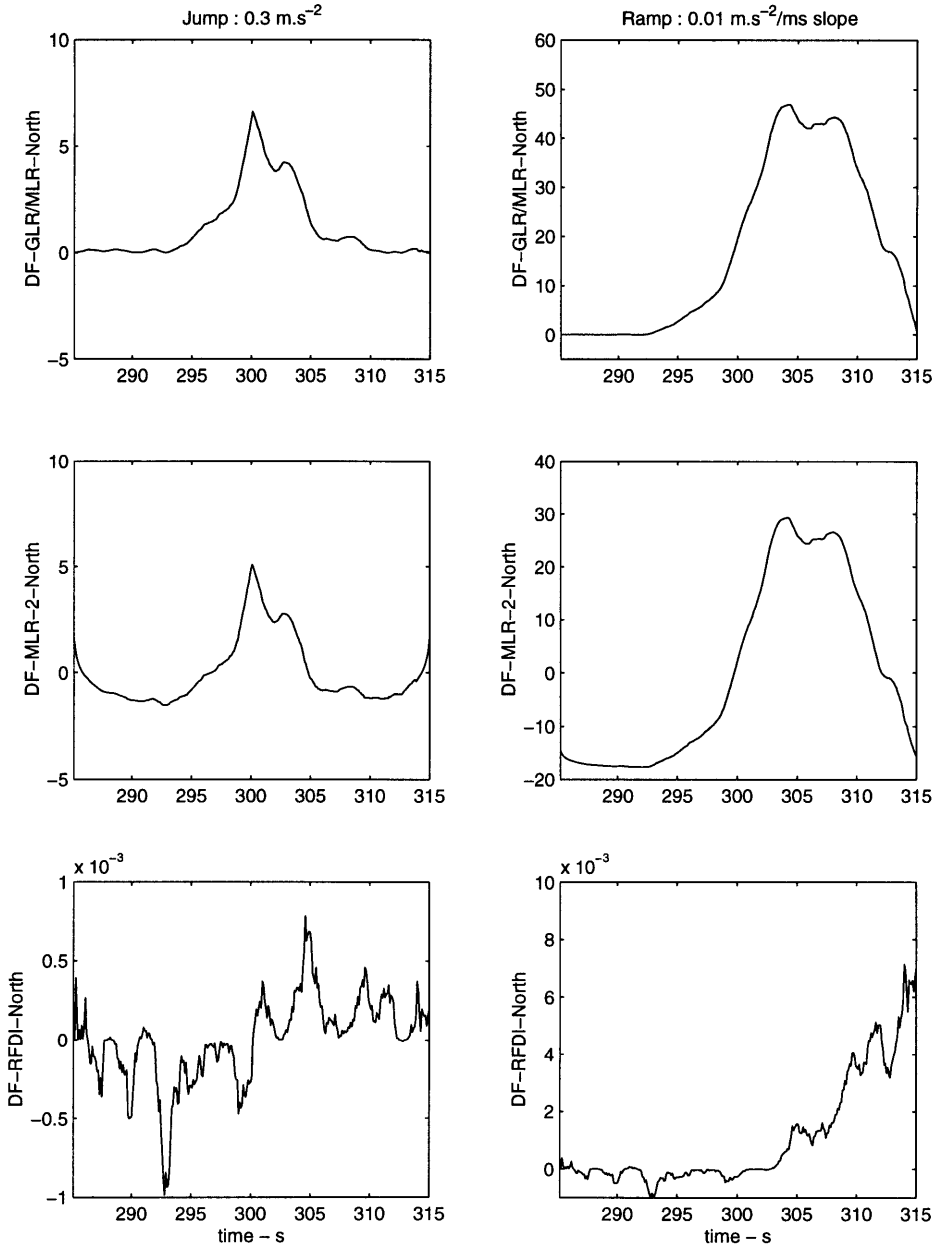

Figure 5-23: The correct decision functions 0n North axis

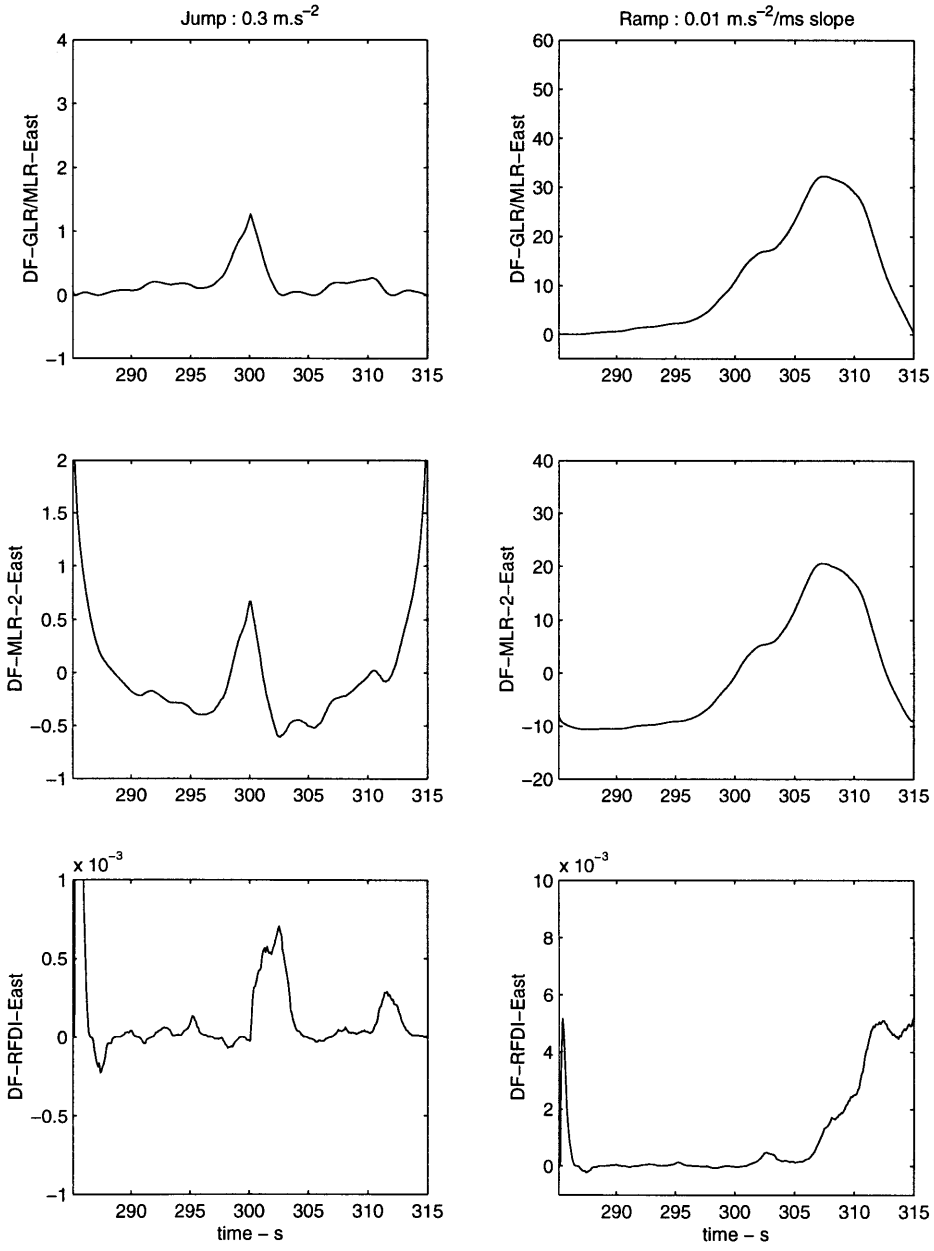

Figure 5-24: The correct decision functions on East axis

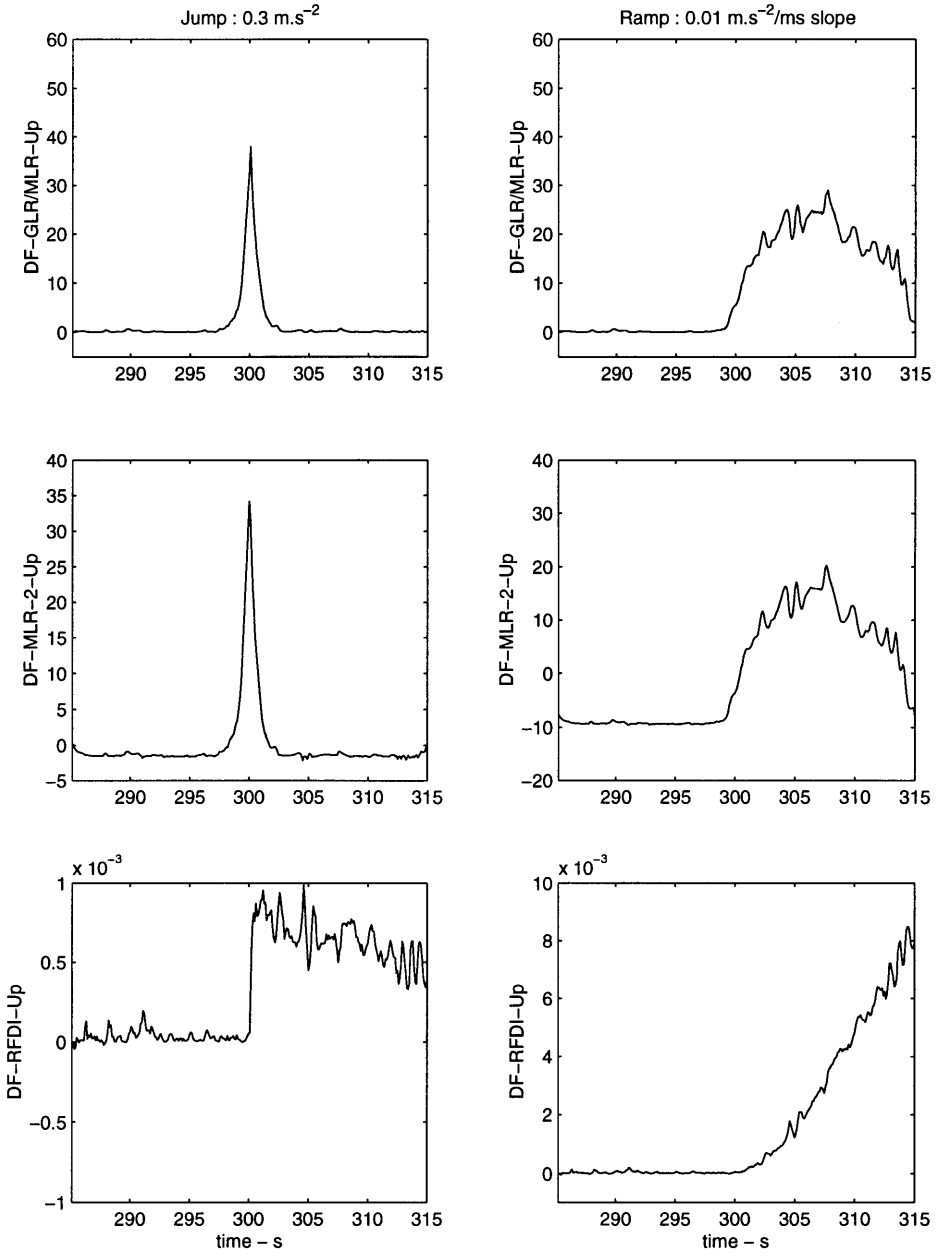

Figure 5-25: The correct decision functions on Down/vertical axis

5.4 Performance Improvement of Failure Detection Algorithms

From the previous discussions in Section 5.3, we can see that the failure detection algorithms perform good detection during the time when the GPS receiver give high quality solution. We also saw that the failure detection algorithms fail to distinguish between the failure and the measurement dynamics when the GPS solution quality is poor. If we remove parts of

the measurement differences that correpond to the low quality GPS solution, we will obtain a new set of measurements from the high quality GPS solution. The ROC's of the new set of measurements on the NED axes are shown in Figure 5-26. We see in this figure that the ROC's on the N and E axes do not lie on the line $p_D = p_{FA}$ any more. The ROC in D axis even almost collides with the $p_D = 1, p_{FA} = 0$ point.

These facts suggest that we need to incorporate the solution status signals into the failure detection process. Hence, it is expected that the failure detection process only works on the part of the measurements in which the GPS gives a high quality solution.

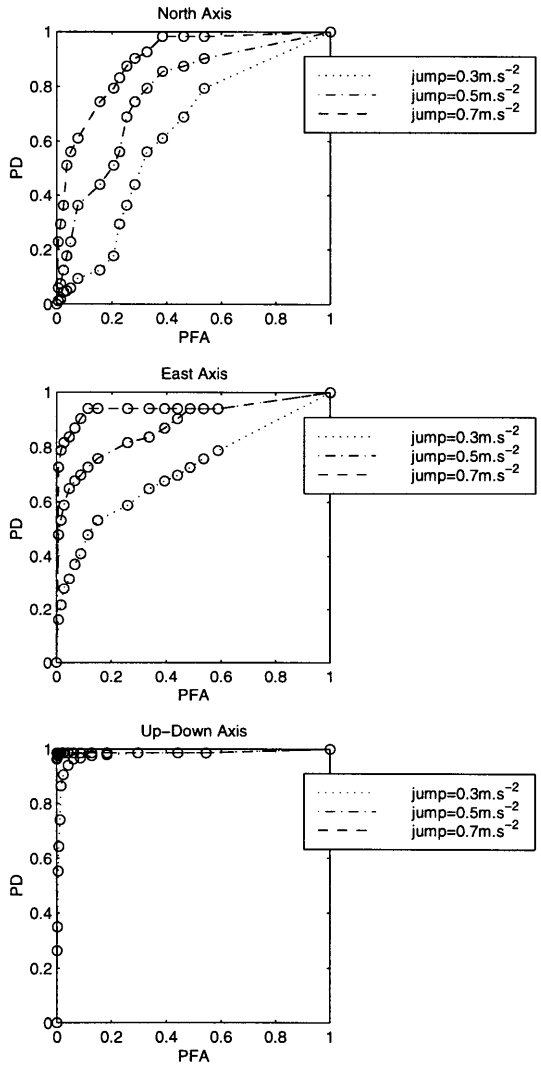

Figure 5-26: The ROC of the observations on NED axes after the fragments of the low quality measurements were removed

To reach this objective, we propose the following improvement methods.

Adaptive Kalman filter in preprocessing stage We saw in Section 5.3 that the failure detection algorithms failed to distinguish between failures and measurement dynamics when the GPS solutions were poor. Each of the failure detection algorithms estimate the presence of failures by analyzing the measurement differences from the IMU and the GPS. We actually compared the accelerations from the IMU and the estimated accelerations from the GPS measurement. The acceleration estimates were generated by the Kalman filter.

The Kalman filters in the preprocessing stage tried to follow the dynamics of measurement outputs even when the solution was not reliable. It is necessary then to inform the Kalman filter not to track the output of the measurement when the solution is poor. This information of GPS status will then be used to adjust the Kalman filter gain so that it will provide estimates based on the previous states only, not based on the measurement dynamics. It is expected that this approach will reduce the number of spikes.

The two sets of gain are given to the Kalman filter for this compensation purpose. The first gain is the standard gain, obtained from the Equation (3.12), while the second gain is

$$K_t = \mathbf{0} \tag{5.59}$$

which means that the Kalman filter will estimate the next states based on the previous state. The second gain is used whenever the solution-status signal exceeds zero. The effect of this compensation is given in Figures 5-27- 5-28 that show the norm of the measurement-difference state when the Kalman filters were not compensated and when the Kalman filter with two sets of gains were used. Using the compensated version of the Kalman filters, we could expect to obtain the difference-states with smaller variances. The peak corresponds to the H_∞ norm and the energy correspond of the H_2 norm of the signals.

Logic operator Another way to reduce a false alarm is by ignoring the alarm that corresponds to the poor GPS solution. Here, we use the logical operator **AND** - $\wedge$. In this improvement method, we simply use the output of the decision function without any compensation and compare it to the output of the solution-status signal :

$$\mathcal{D}_{\mathcal{F}}(k) = l_N(k) \wedge \bar{y}_{solution-status}(k) \tag{5.60}$$

where $\mathcal{D}_{\mathcal{F}}(k)$ is the decision function of window k, $\bar{y}_{solution-status}(k)$ is the GPS-solution-status signal. If the observation at window k contains the poor solution, then the output is simply zero, indicating the difference is due to the poor GPS solution.

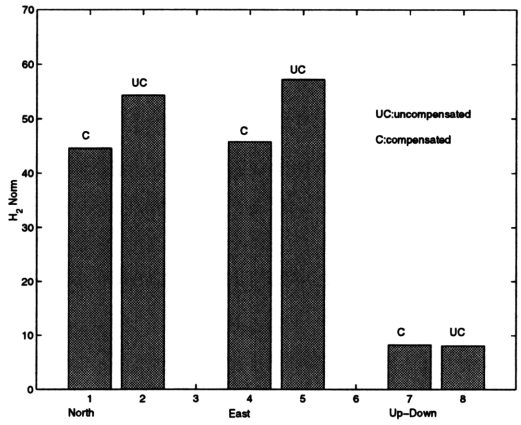

Figure 5-27: The comparison of the H_2 norm of the difference of measurement on NED axes

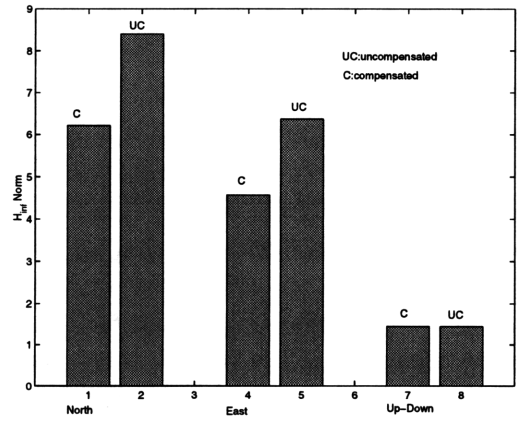

Figure 5-28: The comparison of the H_∞ norm of the difference of measurement on NED axes

Adaptive Gain in failure detection algorithms Before proceeding to the main discussion regarding failure detection algorithm compensation, we present the following theorems from the linear system theory.

Definition : The Controllable Subspace. *The controllable subspace of the linear time-invariant system*

$$x_{t+1} = Ax_t + Bu_t \tag{5.61}$$

is the linear subspace of the states that can be reached from the zero state within a finite time.

Theorem: *The controllable subspace of the* n*-dimensional linear time-invariant system is the linear subspace spanned by the columns of the controllability matrix:*

$$P = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \tag{5.62}$$

In the GLR/MLR test, the Kalman filter has dimension one. Both the forward Kalman filter and the backward Kalman filter in the MLR-2 test also have dimension one. One dimensional Kalman filter has the following equation:

$$\hat{x}_{t+1} = (A - K \cdot C)\hat{x}_t + Ky_t \tag{5.63}$$

If we use $K = 1$, the $\hat{x}_{t+1}$ is forced to be y_t, the next state is forced to follow the current measurement.

In the RFDI, the augmented system has dimension two. The robust estimator in RFDI has the following equation:

$$\begin{bmatrix} d_{a_{t+1}} \\ \nu_{t+1} \end{bmatrix} = (A - K \cdot C) \begin{bmatrix} d_{a_t} \\ \nu_t \end{bmatrix} + Ky_t \tag{5.64}$$

If we want the state estimate not to grow in the failure state direction, we need to assign certain numbers to K so that the controllability matrix P has the eigenvectors:

$$P = \begin{bmatrix} K & (A - K \cdot C) \cdot K \end{bmatrix} \tag{5.65}$$

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad e_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5.66}$$

The choice of $K = [1 \quad 0]^T$ satisfies Equation (5.65) - (5.66)

A poor solution of GPS makes the difference significantly greater than zero. To alert the failure detection filters the presence of the poor solution, we could use the following approach:*when the GPS solution status increases, use different set of gains for the calculation within the window of observation. The following gain can be used for this particular problem:*

- *GLR/MLR failure detection filter :*

$$K_t = 1 \tag{5.67}$$

- *RFDI failure detection filter :*

$$K_t = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.68}$$

- *MLR-2 failure detection filter :*

$$K_t^F = 1 \tag{5.69}$$

$$K_t^B = 1, \tag{5.70}$$

where the superscripts F and B refer to forward filter and backwards filter.

Those gains will force the filter to follow the measurement and give zero residuals even with the existence of poor solution signals. For the RFDI filter, the gain is selected so that the estimation lies in the direction of

$$\begin{bmatrix} d_a \\ \nu \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

which is the direction of the measurement-difference state. This direction will prevent growth in the failure estimation. This principle is adopted from the controllability property in linear system theory.

The performance of each filter after this compensation is shown in Figures 5-29, 5-30, and 5-31. These figures should be compared to the Figures 5-16, 5-17, and 5-18 since both sets show the detection function at the same time interval.

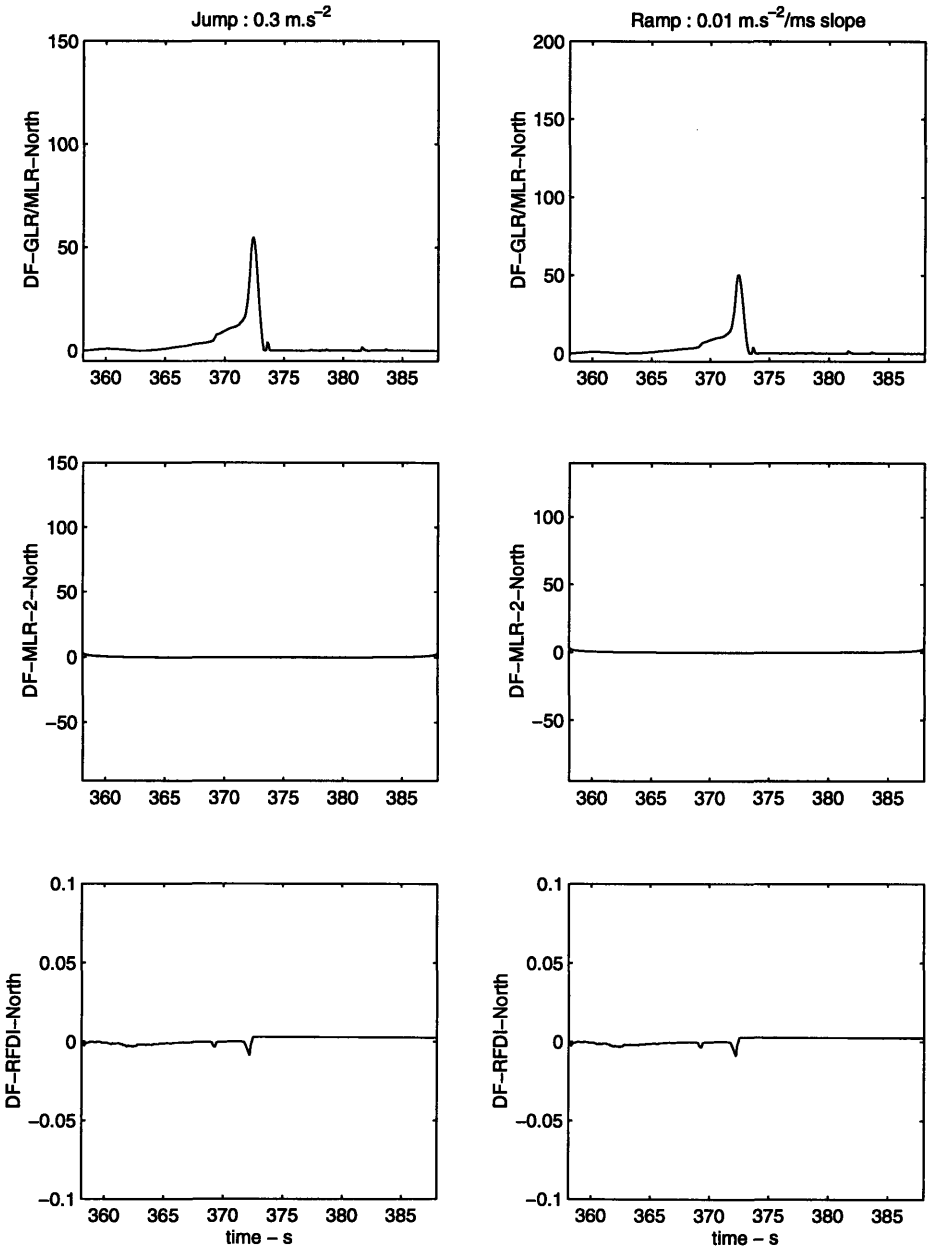

Figure 5-29: The compensated decision functions on North axis

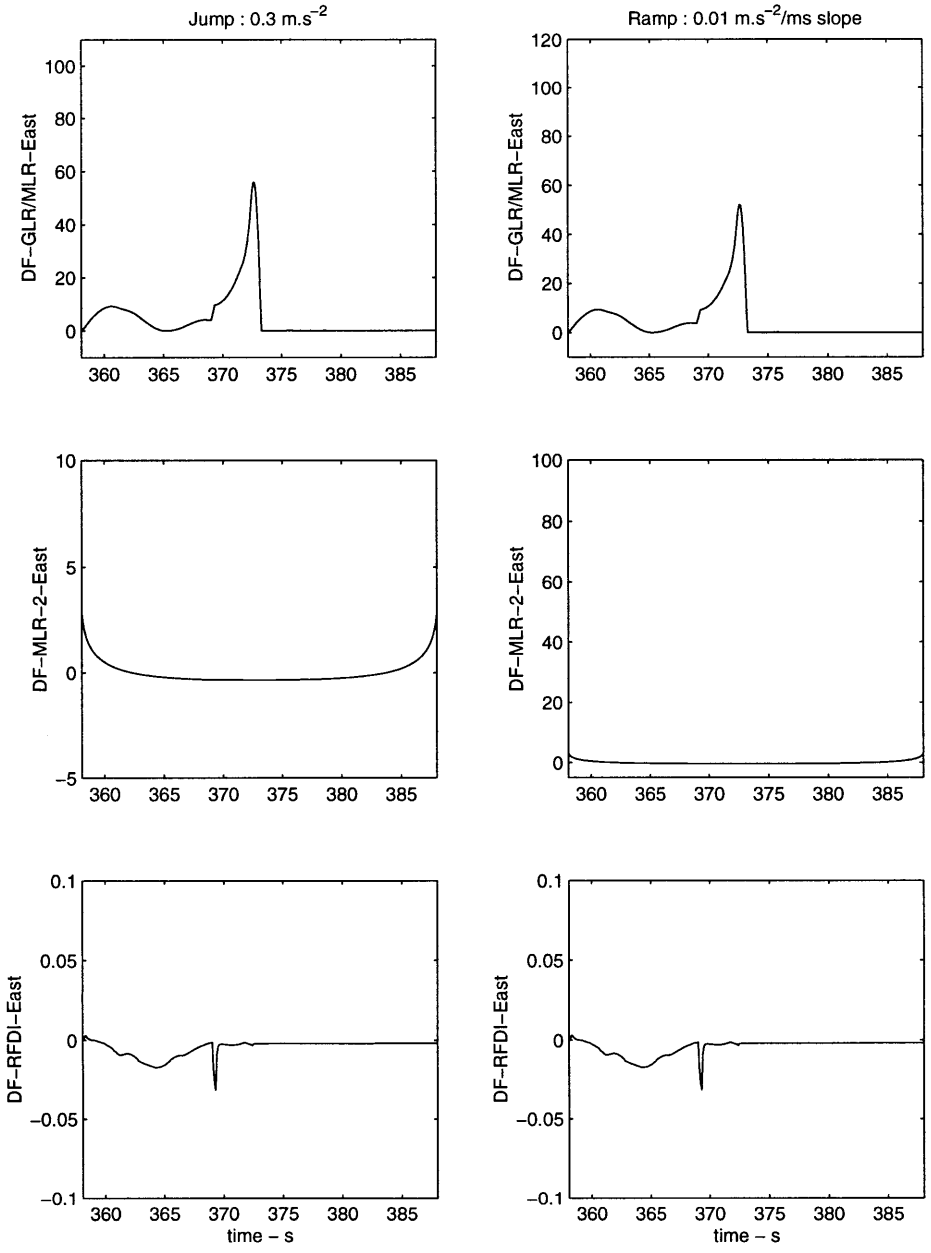

Figure 5-30: The compensated decision functions on East axis

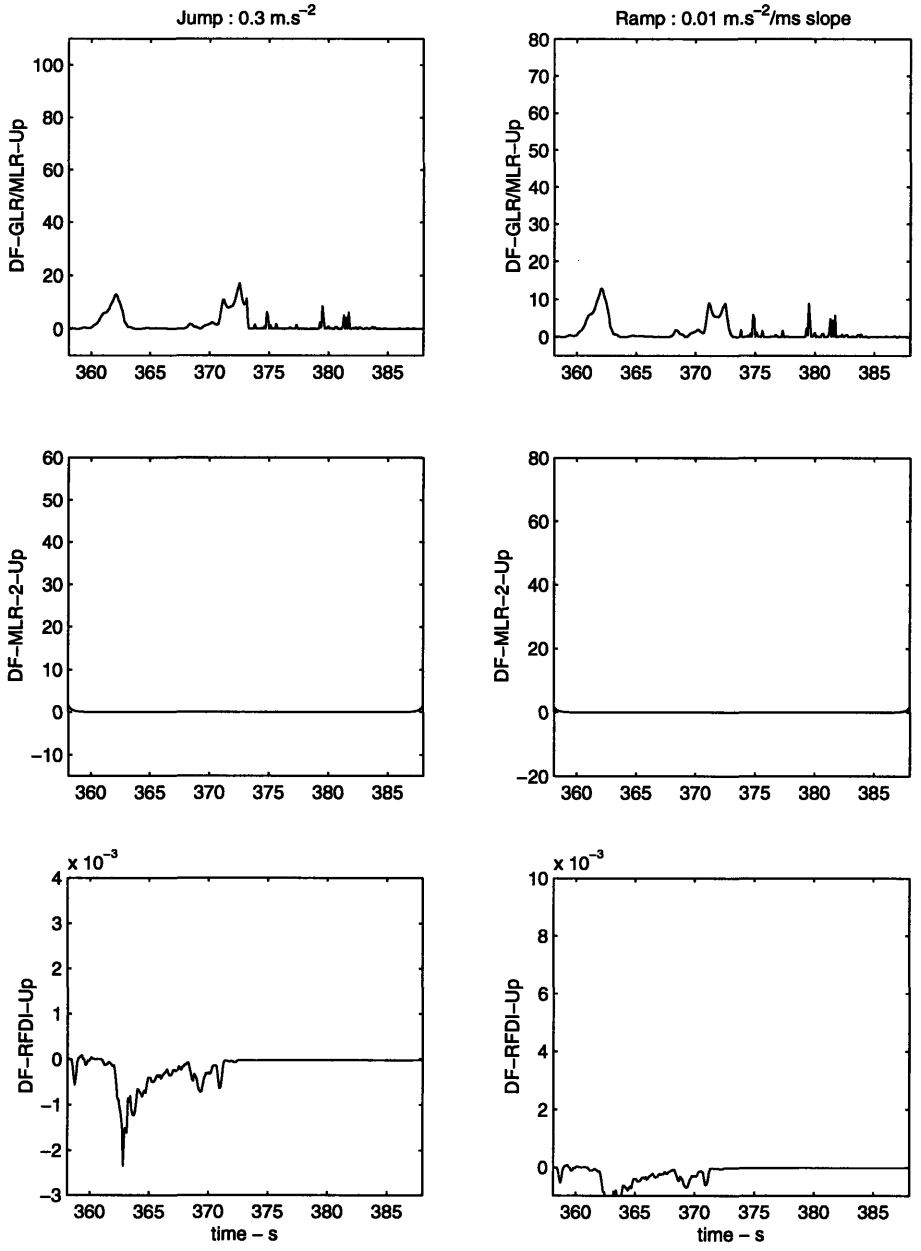

Figure 5-31: The compensated decision functions on Down/vertical axis

The detection functions died off after the new set of gains was used. The decision functions of the RFDI filter immediately went to zero. The same thing also occured with the MLR-2 filter. The increases at both ends are due to the inital and final processings. This would not be a problem if the filter was run continuously. The decision of the GLR/MLR tests however still increased before the poor solution status appeared. This shows that the algorithm is still sensitive to the high dynamic of the measurement.

The GLR/MLR test did not work very well due to the following reason:

$$\begin{aligned} r_t &= y_t - C\hat{x}_t \\ &= y_t - Cy_{t-1} \\ &= y_t - y_{t-1} \end{aligned} \tag{5.71}$$

The Equation (5.71) shows the difference equation of the measurement variable y_t. As differentiation process in continuous function, taking the difference in the discrete-time function will amplify the noise. Hence, the residual is sensitive to the measurement dynamics. The residual may exceed a certain bound due to rapid dynamics so that the GLR/MLR test will interpret this as a failure.

5.5 Computational Complexity

The complexity of each algorithm is shown in Figure 5-32. The GLR/MLR test indeed requires computation on the order of $O(n^2)$, where n indicates the length of the buffer of observation. The MLR-2 and the RFDI require a much smaller number of operations, with the MLR-2 algorithm requiring the least computation.

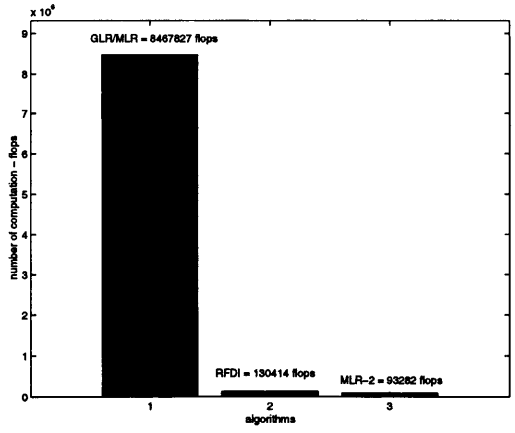

Figure 5-32: The computational complexity of failure detection algorithms

5.6 Closing Remarks

We saw in the beginning of Section 5.4 that without the measurements of the the poor GPS solution, the ROC's of measurements on NED axes were concave downward. It means that if we select the correct η such that the corresponding point on ROC is close to the point $(p_D = 1, p_{FA} = 0)$, the failure detection will give good performance. The algorithm decision functions will have high probability of detection and low probability of false alarm.

The proposed compensation makes the algorithms only work on the measurements when the GPS receiver provides high quality solution. Thus, the nice property of ROC's shown in Figure 5-26 is expected to be achievable.

The RFDI or the MLR-2 have the least computational complexity, but the GLR/MLR test and the RFDI give the fastest detection. The MLR-2 decision function of at time $k \cdot N$ is given at time $(k+1) \cdot N$. The time of detection of the GLR/MLR test within the buffer, however, is faster than that of the RFDI. This fact will not be a problem as long as the final decision is given at the end of observation, after N observations are made.

The compensation with two sets of gains reduces the number of false alarms. We need to be careful, however, since prior to the poor solution signal, the GLR/MLR test's decision function can already rise to a significant value. Other signals might need to be used instead of the *solution status* signal, such as the *velocity status* signal for compensation.

Chapter 6

Conclusion

Before proceeding to the conclusion, we will summarize the discussion of each chapter. The conclusion of this thesis is presented after the summary section and the suggestions will close this chapter.

6.1 Thesis Summary

Chapter 2 should provide an adequate background of the hypothesis testing for failure detection. The hypothesis testing described in Chapter 2 was developed for the system with either hardware redundancy or analytical redundancy. The performance measures used in Chapter 6, i.e., the probability of false alarm, the probability of detection, and computational complexity were given in Chapter 2.

Chapter 3 gave the description of failure detection algorithms from the statistical point of view. The framework of failure detection is developed based on probability theory. The GLR test, for instance, develops the decision function by maximizing the LLR of the observation by varying both possible failure location and possible failure magnitude. The MLR name implies the marginalization of the nuisance parameter, so that the maximalization is developed only on a single parameter. The MLR test tries to find the possible time of failure that will maximize the LLR of the observation. Marginalization of the related observation densities makes it possible to derive the failure detection algorithm implemented by only two filters. The MLR test implemented by only two filters was described. Numerical examples were given to illustrate how the algorithm worked to detect the failure.

Chapter 4 gave a description of the failure detection algorithm based on the deterministic

framework. The general idea of this deterministic approach is to build a *robust observer* that will *estimate the failure state.* The robust estimator is developed on the game-theoretic optimization where the noise and plant perturbation are one player, and the state estimate is their opponent. There are two kinds of estimators derived, the first one corresponds to the estimator for systems with unknown noise model, the second one correspond to the estimator for systems whose dynamic model and noise model are not known accurately. The RFDI is basically the estimator whose state vector is the augmentation of state dynamics and failure dynamics. The RFDI eventually acts as the observer that will estimate the failure state. Numerical examples were given to illustrate how to use each type of estimator for failure detection purposes.

Chapter 5 illustrated the application of the algorithms given in the previous chapters in a real situation. The problem faced by each of the algorithms encouraged the improvements proposed in Section 5.3. The proposed improvements are expected to reduce the number of false alarms due to the low quality of observation.

6.2 Conclusion

The deterministic failure detection algorithm : the RFDI and the statistical failure detection algorithms : the GLR/MLR test and the MLR-2 test were studied and applied on the real data observation. The results lead to the following conclusion :

1. The deterministic and the statistical failure detection algorithms function well under the condition that the observation behaves as it is predicted from the model. When the observation dynamic is beyond the operating region of the model, the false alarm probability will increase significantly.

2. The compensation by adaptive filter gain allows the reduction of false alarm rate. The deterministic filter as well as the MLR-2 filter respond very well with the adaptive gain approach.

3. The detection time of the deterministic algorithm is not as fast as that of the statistical algorithm : the GLR/MLR test. The estimation process of the failure magnitude in the deterministic setting is not instantaneous since the deterministic filter should distinguish whether it is really a failure state or it is just a transient response.

4. Introducing the additional factor in the decision function of the deterministic algorithm, $\mathbf{sign}(\nu)$ turns out to be useful. It will suppress the squared transient response below zero. The actual failure magnitude is greater than zero and the squared failure magnitude is distinguishable from the squared transient response. The use of inverse variance of failure states will heavily penalize the high transient response of the failure estimate.

5. The deterministic algorithm and the MLR-2 are computationally efficient. The GLR/MLR test requires parallel Kalman filters for on-line application, hence will require more computing power and memory space. The deterministic algorithm and the MLR-2 test require less computing power and memory space.

6.3 Recommendation for Future Work

Refinement should be made before the algorithms are applied to real applications.

1. Modeling

   A better model should be used rather than just one single dimension of measurement difference. The capability of this model is very limited, since the deviation of the nominal value will easily be judged as failure by the failure detection algorithms. Knowing the error model of each instrument will give a better performance, since the filters will have more knowledge of the real system. The filter then will know whether the real failure occurs or it is just the dynamic of the observation.

2. Real-time In-flight Experiment

   Flying the instruments of experiment off-the-ground will reduce the number of obstacle between the GPS satellites and the GPS receiver antenna, hence it is expected that the frequency of bad solution status will decrease. It will be interesting to see the performance of the compensated failure detection algorithms in the real flight and in real-time.

Appendix A

Constructing The Parity Matrix

Consider the hardware redundancy system with the following measurement equation

$$Y_t = H_t X_t + V_t + \Upsilon(t, t_0) \tag{A.1}$$

Y_t represents the observation vector, X_t represents the quantities measured, V_t represents the noise of measurement and $\Upsilon(t, t_0)$ represents the additive failure vector. The parity vector is given by

$$\zeta_t = C_t V_t + C_t \Upsilon(t, t_0) \tag{A.2}$$

The expected value and the covariance of the parity vector will be

$$E[\zeta_t] = C_t \Upsilon t, t_0 \tag{A.3}$$

$$cov[\zeta_t] = C_t R_t C_t^T \tag{A.4}$$

Consider the least squares estimate of the state vector X_t, given the measurement Y_t, with noise covariance R_t

$$\begin{aligned} \hat{X}_t &= \left(H_t^T R_t^{-1} H_t\right)^{-1} H^T R_t^{-1} Y_t \\ &= X_t + \left(H_t^T R_t^{-1} H_t\right)^{-1} H^T R_t^{-1} \left(V_t + \Upsilon(t, t_0)\right) \end{aligned} \tag{A.5}$$

The residual vector resulting from this estimate of X_t is

$$\varepsilon_t = Y_t - H_t \hat{X}_t$$

$$
\begin{aligned}
&= \left[I^{n\times n} - H_t(H_t^T R_t^{-1} H_t)^{-1} H^T R_t^{-1}\right] Y_t \\
&= \left[I^{n\times n} - H_t(H_t^T R_t^{-1} H_t)^{-1} H^T D_t^{-T} D_t^{-1}\right] D_t D_t^{-1} Y_t \\
&= D_t \left[I^{n\times n} - D_t^{-1} H_t(H_t^T R_t^{-1} H_t)^{-1} H^T D_t^{-T}\right] D_t^{-1} Y_t \\
&= D_t C_t^T C_t D_t^{-1} Y_t.
\end{aligned} \tag{A.6}
$$

The last relationship provides the equality for C matrix.

$$C_t^T C_t = I^{n\times n} - D_t^{-1} H_t(H_t^T R_t^{-1} H_t)^{-1} H^T D_t^{-T} \tag{A.7}$$

$$R_t = D_t D_t^T \tag{A.8}$$

In the case where $R_t = I$, the following algorithm can be used to determine C_t, or C for time-invariant case.

Let

$$W = C^T C = I - H(H^T H)^{-1} H^T. \tag{A.9}$$

The elements of the C matrix are

$$c_{11} = (w_{11})^{\frac{1}{2}} \tag{A.10}$$

$$c_{1j} = \frac{w_{1j}}{v_{11}}, \quad j = 2, \ldots, n \tag{A.11}$$

$$c_{ij} = 0, \quad i = 2, \ldots, n-r; \quad j = 1, \ldots, i-1 \tag{A.12}$$

$$c_{ii} = \left(w_{ii} - \sum_{l=1}^{i-1} c_{li}^2\right)^{\frac{1}{2}}, \quad i = 2, \ldots, n-r \tag{A.13}$$

$$c_{ij} = \frac{\left(w_{ij} - \sum_{l=1}^{i-1} v_{li} v_{lj}\right)}{v_{ii}} \tag{A.14}$$

$$i = 2, \ldots, n - r$$

$$j = i + 1, \ldots, n$$

where

$$H \in \mathcal{R}^{n \times r}$$

$$I \in \mathcal{R}^{n \times n}$$

Appendix B

Statistical Failure Detection Algorithm Notes

B.1 The Density of Measurement

Suppose we have observation $y^N = \{y_1, y_2, \ldots, x_N\}$. From Bayes' rule we have

$$p(A, B) = p(A|B)p(B) \tag{B.1}$$

Following the Bayes' rule, the density function of the measurements is expressed as

$$\begin{aligned} p(y^N) &= p\left(y_1, y_2, \ldots, y_N\right) \\ &= p\left(y_N|y_1, y_2, \ldots, y_{N-1}\right) p\left(y_1, y_2, \ldots, y_{N-1}\right) \\ &= p\left(y_N|y^{N-1}\right) p\left(y^{N-1}\right) \\ &= \prod_{t=1}^{N} p\left(y_t|y^{t-1}\right) \\ &= \prod_{t=1}^{N} \gamma\left(y_t - H_t\hat{x}_{t|t-1}, H_t P_{t|t-1} H_t^T + R_t\right) \end{aligned} \tag{B.2}$$

B.2 The Backwards Markovian Model

Given a wide sense forwards Markovian model, namely a state-space system driven by a white-noise process. The white-noise process is uncorrelated with the random inital state of the system. For this forward model, we can obtain a corresponding backwards Markovian

model,i.e. a system driven backwards in time from the *terminal* state by a white-noise process that is uncorrelated with this terminal state.

For the forwards Markovian model on the interval $[0,T]$,

$$\begin{aligned} x_{t+1} &= F_t x_t + G_t u_t \\ y_t &= H_t x_t + J_t u_t \\ E[u_t] &= 0 \\ E[u_t u_\tau] &= I\delta_{t\tau} \\ E[x_0] &= 0 \\ E[x_t x_\tau] &= \Pi_0 \\ E[x_0 u_t] &= 0, \qquad t \geq 0 \end{aligned} \tag{B.3}$$

the backwards Markovian model of the same interval $[0,T]$ will be

$$\begin{aligned} x^B_{t-1} &= F^B_t x^B_t + G^B_t u^B_t \\ y^B_{t-1} &= H^B_t x^B_t + J^B_t u^B_t \\ E[x^B_T u^B_t] &= 0, \qquad t \leq T \end{aligned} \tag{B.4}$$

In the case of invertible F_t, the backward matrices will be

$$\begin{aligned} F^B_{t+1} &= F_t^{-1} - F_t^{-1} G_t G_t^T \Pi_{t+1}^{-1} \\ G^B_{t+1} &= -F_t^{-1} G_t \\ H^B_{t+1} &= H_t F^B_{t+1} + J_t G_t^T \Pi_{t+1}^{-1} \\ J^B_{t+1} &= H_t G^B_{t+1} + J_t \end{aligned} \tag{B.5}$$

More information related to backwards Markovian model can be found in [47].

B.3 Derivation of Linear Regression Through State-Space Approach

Consider the linear regression model :

$$y_t^T = \nu_t^T \varphi_t + e_t^T \tag{B.6}$$

and assuming that the true value of parameter vector θ varies according to

$$\nu_{t+1} = \nu_t + w_t \tag{B.7}$$

We then have the following state-space description of the system :

$$\begin{aligned} \nu_{t+1} &= \nu_t + w_t \\ y_t &= \varphi_t^T \nu_t + e_t \end{aligned} \tag{B.8}$$

Given the following description of noise :

$$E[w_t w_\tau^T] = R_t^w \delta_{t-\tau} \tag{B.9}$$

$$E[e_t e_\tau^T] = R_t^e \delta_{t-\tau} \tag{B.10}$$

$$E[w_t \nu_\tau^T] = 0$$

The Kalman filter equations to estimate θ are :

$$\begin{aligned} \hat{\nu}_{t+1|t+1} &= \hat{\nu}_{t|t} + G_{t+1}[y_{t+1} - \varphi_{t+1}^T \hat{\nu}_{t|t}] \\ G_t &= P_{t,t-1}\varphi_t[\varphi_t^T P_{t,t-1}\varphi_t + R^e]^{-1} \\ P_{t,t-1} &= P_{t-1,t-1} \\ P_{t,t} &= (I - G_t\varphi_t)P_{t-1,t-1} \\ &= P_{t-1,t-1} - P_{t-1,t-1}\varphi_t[\varphi_t^T P_{t-1,t-1}\varphi_t + R^e]^{-1}\varphi_t^T P_{t-1,t-1} \end{aligned} \tag{B.11}$$

with :

- G_t : Kalman filter gain

- P_t : covariance matrix of θ

- $R_t^w = 0$

B.4 On-Line Expressions for The GLR Test

From the Kalman filter equations under the hypotheses of no jump, we have the following variables :

$\hat{x}_{t|t}$, $K_t = P_{t|t-1}H_tS_t^{-1}$, $\varepsilon_t = y_t - H_t\hat{x}_{t|t}$, $S_t = R_t + H_tP_{t|t=1}H_t^T$

Suppose there was jump ν at time k, we could postulate the following model,

$$\begin{aligned} \hat{x}_{t|t}(k) &= \hat{x}_{t|t} + \mu_t(k)\nu \\ \varepsilon_{t|t}(k) &= \varepsilon_{t|t} + \varphi_t^T(k)\nu \end{aligned} \tag{B.12}$$

Here $\hat{x}_{t|t}$ and $\varepsilon_{t|t}$ are quantities obtained from the Kalman filter under the hypotheses of no jump. The relationships hold due to the linearity properties of the model.

The regressor $\mu_t(k)$ and $\varphi_t^T(k)$ are computed recursively by

$$\begin{aligned} \varphi_{t+1}^T(k) &= H_{t+1}(\prod_{i=k}^{t} F_i - F_t\mu_t(k)) \\ \mu_{t+1}(k) &= F_t\mu_t(k) + K_{t+1}\varphi_t^T(k) \end{aligned} \tag{B.13}$$

with the initial conditions :

$$\begin{aligned} \mu_k(k) &= 0 \\ \varphi_k^T(k) &= 0 \end{aligned} \tag{B.14}$$

The recursive expressions of μ_t and φ_t are derived as follows.

$$\begin{aligned} \varphi_t^T(k)\nu &= \varepsilon_{t+1}(k) - \varepsilon_{t+1} \\ &= H_{t+1}[x_{t+1}(k) - \hat{x}_{t+1}(k)] - H_{t+1}[x_{t+1} - \hat{x}_{t+1}] \\ &= H_{t+1}[x_{t+1}(k) - x_{t+1}] - H_{t+1}[\hat{x}_{t+1}(k) - \hat{x}_{t+1}] \\ &= H_{t+1}(\prod_{i=k}^{t} F_i)\nu - H_{t+1}F_t[\hat{x}_{t|t}(k) - \hat{x}_{t|t}] \end{aligned}$$

$$= H_{t+1}[\prod_{i=k}^{t} F_i - F_t\mu_t(k)]\nu \tag{B.15}$$

and

$$\begin{aligned}
\mu_{t+1}(k)\nu &= \hat{x}_{t+1|t+1}(k) - \hat{x}_{t+1|t+1} \\
&= \hat{x}_{t+1|t}(k) + K_{t+1}\varepsilon_{t+1}(k) - \hat{x}_{t+1|t} - K_{t+1}\varepsilon_{t+1} \\
&= F_t\hat{x}_{t|t}(k) + K_{t+1}\varepsilon_{t+1} + K_{t+1}\varphi_{t+1}^T(k)\nu - F\hat{x}_{t|t} - K_{t+1}\varepsilon_{t+1} \\
&= F_t[\hat{x}_{t|t}(k) - \hat{x}_{t|t}] + K_{t+1}\varphi_{t+1}^T(k)\nu \\
&= F_t\mu_t(k)\nu + K_{t+1}\varphi_{t+1}^T(k)\nu
\end{aligned} \tag{B.16}$$

The likelihood ratio test is expressed as :

$$\begin{aligned}
l_N(k,\hat{\nu}(k)) &= 2\log\frac{p(\varepsilon_1^N|k,\hat{\nu}(k))}{p(\varepsilon_1^N|k=N)} \\
&= \sum_{t=k+1}^{N} \varepsilon_t^T S_t^{-1}\varepsilon_t - (\varepsilon_t - \varphi_t^T(k)\hat{\nu}(k))^T S_t^{-1}(\varepsilon_t - \varphi_t^T(k)\hat{\nu}(k)) \\
&= f_N^T(k)R_N^{-1}f_N(k)
\end{aligned} \tag{B.17}$$

The quantities $f_N(k)$ and $R_N(k)$ are the quantities of the LS estimator. They are computed recursively in the following manner.

$$\begin{aligned}
f_N(k) &= \sum_{t=1}^{N} \varphi_t(k)S_t^{-1}\varepsilon_t \\
R_N(k) &= \sum_{t=1}^{N} \varphi_t(k)S_t^{-1}\varphi_t^T(k)
\end{aligned} \tag{B.18}$$

The maximum likelihood estimate of ν, given the jump instant at k is

$$\hat{\nu}(k) = R_N^{-1}f_N(k) \tag{B.19}$$

For on-line application, we need to modify equation for $\hat{\nu}$ to avoid matrix inversion computation of R. Instead, $\hat{\nu}$ is computed through recursive equations :

$$\begin{aligned}
\hat{\nu}_t &= \hat{\nu}_{t-1} + L_t[\varepsilon_t - \varphi_t^T\hat{\nu}_{t-1}] \\
L_t &= P_{t-1}^{\nu}\varphi_t[\varphi_t^T P_{t-1}^{\nu}\varphi_t + \Lambda_t]^{-1}
\end{aligned} \tag{B.20}$$

$$P_t^{\nu} = P_{t-1}^{\nu} - L_t \varphi_t^T P_{t-1}^{\nu}$$

as it was given in Appendix B.3.

B.5 The MLR Test Derivation

Consider the following hypotheses :

$$H_0 : \varepsilon_t \in N(0, S_t) \tag{B.21}$$

$$H_1 : \varepsilon_t \in N(\varphi_t^T \nu, S_t). \tag{B.22}$$

Let

- $y_t = \varepsilon_t$;
- $Y = \{y_1, y_2, \ldots, y_N\}$;
- $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$.

where p is the system dimension, N is the observation length, and d is the dimension of ν. In the MLR test, ν is considered as a random variable. This is the *key* for marginalization. Marginalizing the conditional probability of $p(y^N|\nu)$ will give

$$\begin{aligned} p(y^N) &= \int_{-\infty}^{+\infty} p(y^N|\nu)p(\nu)d\nu \\ &= (2\pi)^{-(Np+d)/2} \times |\Lambda|^{-N/2} \times |R_0|^{-d/2} \times \\ &\quad \int_{-\infty}^{+\infty} \exp\left\{-\frac{1}{2}\left[(Y-\Phi^T\nu)^T\Lambda^{-1}(Y-\Phi^T\nu) + (\nu-\nu_0)^T R_0(\nu-\nu_0)\right]\right\} d\nu \end{aligned} \tag{B.23}$$

The expression inside the exponent can be rewritten as

$$\left(Y-\Phi^T\nu\right)^T \Lambda^{-1}\left(Y-\Phi^T\nu\right) + (\nu-\nu_0)^T R_0 (\nu-\nu_0)$$

$$= Y^T\Lambda^{-1}Y - 2\,\nu^T\Phi\,\Lambda^{-1}Y + \nu^T\Phi\,\Lambda^{-1}\Phi^T\nu + \nu^T R_0\nu - \nu_0^T R_0\nu_0$$

$$= \nu^T (R_0 + R_N)\,\nu - 2\,\nu^T (f_0 + f_N) + Y^T\Lambda^{-1}Y + \nu_0^T R_0\nu_0$$

$$
\begin{aligned}
= \quad & \left(\nu - (R_0 + R_N)^{-1}(f_0 + f_N)\right)^T (R_0 + R_N) \left(\nu - (R_0 + R_N)^{-1}(f_0 + f_N)\right) - \\
& (f_0 + f_N)^T (R_0 + R_N)^{-1}(f_0 + f_N) + Y^T \Lambda^{-1} Y + \nu^T R_0 \nu_0 \\
= \quad & (\nu - \hat{\nu})^T (R_0 + R_N)(\nu - \hat{\nu}) - \hat{\nu}^T (f_0 + f_N) + Y^T \Lambda^{-1} Y + \nu_0^T R_0 \nu_0 \\
= \quad & (\nu - \hat{\nu})^T (R_0 + R_N)(\nu - \hat{\nu}) + \left(Y - \Phi^T \hat{\nu}\right)^T \Lambda^{-1} \left(Y - \Phi^T \hat{\nu}\right) + \\
& \hat{\nu}^T f_N - \hat{\nu}^T R_N \hat{\nu} - \hat{\nu}^T f_0 + \nu_0^T R_0 \nu_0 \\
= \quad & (\nu - \hat{\nu})^T (R_0 + R_N)(\nu - \hat{\nu}) + \left(Y - \Phi^T \hat{\nu}\right)^T \Lambda^{-1} \left(Y - \Phi^T \hat{\nu}\right) + \\
& \hat{\nu}^T (f_0 + f_N) - 2\hat{\nu} f_0 - \hat{\nu}^T (R_0 + R_N)\hat{\nu} + \hat{\nu}^T R_0 \hat{\nu} + \nu_0^T R_0 \nu_0 \\
= \quad & (\nu - \hat{\nu})^T (R_0 + R_N)(\nu - \hat{\nu}) + \left(Y - \Phi^T \hat{\nu}\right)^T \Lambda^{-1} \left(Y - \Phi^T \hat{\nu}\right) \\
& + (\hat{\nu} - \nu_0)^T R_0 (\hat{\nu} - \nu_0)
\end{aligned}
$$

The following relationships are used for expanding the exponential.

$$
\begin{aligned}
R_N &= \Phi \Lambda^{-1} \Phi^T \\
f_N &= \Phi \Lambda^{-1} Y \\
f_0 &= R_0 \nu_0 \\
\hat{\nu} &= (R_0 + R_N)^{-1}(f_0 + f_N)
\end{aligned}
$$

For $\hat{\nu} \in N(\nu, R_0 + R_N)$ the probability $p(y^N)$ can be written as

$$
\begin{aligned}
p(y^N) &= (2\pi)^{-Np/2} \times |\Lambda|^{-N/2} \times \left(\frac{|R_0|}{|R_0 + R_N|}\right)^{1/2} \\
& \quad \times \exp\{-\frac{1}{2}[(Y - \Phi^T \hat{\nu}_N)^T \Lambda^{-1} (Y - \Phi^T \hat{\nu}_N) + (\hat{\nu}_N^T - \nu_0)^T R_0 (\hat{\nu}_N^T - \nu_0)]\} \\
& \quad \times \int_{-\infty}^{+\infty} (2\pi)^{-d/2} \times |R_0 + R_N|^{1/2} \\
& \quad \times \exp[-\frac{1}{2}(\nu - \hat{\nu}_N)^T (R_0 + R_N)(\nu - \hat{\nu}_N)] \quad d\nu \\
&= p(y^N | \hat{\nu}) \times \left(\frac{|R_0|}{|R_0 + R_N|}\right)^{1/2} \\
& \quad \times \exp[-\frac{1}{2}(\nu - \hat{\nu}_N)^T (R_0 + R_N)(\nu - \hat{\nu}_N)] \\
&= p(y^N | \hat{\nu}_N) \times p_\nu(\hat{\nu}) \times |P_N|^{1/2} \times (2\pi)^{d/2} \qquad \text{(B.24)}
\end{aligned}
$$

For $p_\nu(\nu) = 1$, the case of non-informative prior, R_0 is set to zero since this prior has infinite variance. By doing so, the probabilty $p(y^N)$ can be written as

$$\begin{aligned} p(y^N) &= p(y^N|\hat{\nu}_N) \times (2\pi)^{-d/2} \times \\ &\quad \int_{-\infty}^{+\infty} \exp[-\frac{1}{2}(\nu - \hat{\nu}_N)^T (R_0 + R_N)(\nu - \hat{\nu}_N)] \quad d\nu \\ &= p(y^N|\hat{\nu}_N) \times |P_N|^{1/2} \end{aligned} \tag{B.25}$$

Now, the LLR for Gaussian prior is given by

$$\begin{aligned} 2\ln\frac{p(y^N|\tau^*)}{p(y^N)} &= 2\ln\frac{p(y^{\tau^*})p(y_{\tau^*+1}^N|y^k,\tau^*)}{p(y^{\tau^*})p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= 2\ln\frac{p(y_{\tau^*+1}^N|y^k,\tau^*)}{p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= 2\ln\frac{p(y^N|\hat{\nu}_N) \times p_\nu(\hat{\nu}) \times |P_N|^{1/2} \times (2\pi)^{d/2}}{p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= \sum_{t=\tau^*+1}^{N} \varepsilon_t^T S_t^{-1} \varepsilon_t - (\varepsilon_t - \varphi_t^T(\tau^*)\hat{\nu}_N(\tau^*))^T S_t^{-1} \times \\ &\quad (\varepsilon_t - \varphi_t^T(\tau^*)\hat{\nu}_N(\tau^*)) + \log|P_N(\tau^*)| - \log|P_\nu(\tau^*)| + \\ &\quad (\hat{\nu}(\tau^*) - \nu_0)^T P_\nu^{-1}(\hat{\nu}(\tau^*) - \nu_0) \\ &= l_N(\tau^*, \hat{\nu}(\tau^*)) - \log|R_N(\tau^*)| + C_{prior} \end{aligned} \tag{B.26}$$

where d is the dimension of ν. The LLR for non-informative prior is derived in the same manner as that of equation (B.26).

$$\begin{aligned} 2\ln\frac{p(y^N|\tau^*)}{p(y^N)} &= 2\ln\frac{p(y^{\tau^*})p(y_{\tau^*+1}^N|y^k,\tau^*)}{p(y^{\tau^*})p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= 2\ln\frac{p(y_{\tau^*+1}^N|y^k,\tau^*)}{p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= 2\ln\frac{p(y^N|\hat{\nu}_N) \times |P_N|^{1/2}}{p(y_{\tau^*+1}^N|y^{\tau^*})} \\ &= \sum_{t=\tau^*+1}^{N} \varepsilon_t^T S_t^{-1} \varepsilon_t - (\varepsilon_t - \varphi_t^T(\tau^*)\hat{\nu}_N(\tau^*))^T S_t^{-1} \times \\ &\quad (\varepsilon_t - \varphi_t^T(\tau^*)\hat{\nu}_N(\tau^*)) + \log|P_N(\tau^*) + 0 \\ &= l_N(\tau^*, \hat{\nu}(\tau^*)) - \ln|R_N(\tau^*)| + C_{prior}. \end{aligned} \tag{B.27}$$

where p is the dimension of ε_t.

Appendix C

Derivations of Deterministic Failure Detection Algorithm

C.1 Derivation of Robust Estimator for The Uncertain Noise Model

The derivation of the robust estimator in the case of unknown noise model will be given in this section. The state-space equations from equations (4.5) - (4.7) are repeated here.

$$x_{t+1} = A_t x_t + B_t r_t \tag{C.1}$$

$$e_t = M_t(x_t - \hat{x}_t) \tag{C.2}$$

$$y_t = C_t x_t + D_t r_t \tag{C.3}$$

For the performace index

$$\begin{aligned} J_1 &= \frac{1}{2}\|e\|^2 \\ &= \frac{1}{2}\sum_{t=1}^{N}(x_t - \hat{x}_t)^t M_t^T M_t (x_t - \hat{x}_t) \end{aligned} \tag{C.4}$$

the game theoritic approach is used to derive the estimator. The optimization of the performace index that leads to the derivation of the estimator is expressed as

$$\min_{\hat{x}} \max_{r, x_0} J_1 \tag{C.5}$$

with the constraints of equations (C.1) - (C.3) and

$$\|r\|^2 + \|x_0 - \hat{x}_0\|^2_{\breve{P}_0^{-1}} \leq 1 \tag{C.6}$$

The estimator will be in the same form as the Kalman filter :

$$\begin{aligned} \hat{x}_{t+1} &= (A_t - K_t C_t)\hat{x}_t + K_t y_t \qquad \text{(C.7)} \\ \tilde{x}_{t+1} &= (A_t - K_t C_t)\tilde{x}_t + (B_t - K_t D_t) r_t \\ &= \tilde{A}_t \tilde{x}_t + \tilde{B}_t r_t \\ \tilde{x}_t &= x_t - \hat{x}_t \qquad \text{(C.8)} \\ e_t &= M_t \tilde{x}_t \end{aligned}$$

The game theoritic optimization now can be described as :

$$\begin{aligned} \min_K \max_{r,x_0} \quad & J_1 \qquad \text{(C.9)} \\ \text{subject to} \quad & \tilde{x}_{t+1} = \tilde{A}_t \tilde{x}_t + \tilde{B}_t r_t \\ \text{and} \quad & \|r\|^2 + \|x_0 - \hat{x}_0\|^2_{\breve{P}_0^{-1}} \leq 1 \\ K &= [K_0, \ldots, K_{N-1}] \end{aligned}$$

To incorporate constraints into the performance index, Langrange multipliers are introduced to the new performance index J_2, [45]:

$$\begin{aligned} J_2 &= J_1 + \sum_{t=0}^{N-1} [\lambda_{t+1}^T (\tilde{x}_{t+1} - \tilde{A}_t \tilde{x}_t - \tilde{B}_t r_t)] - \frac{\gamma^2}{2}(\|r\|^2 + \|x_0 - \hat{x}_0\|^2_{\breve{P}_0^{-1}}) \\ &= -\frac{\gamma^2}{2}(r_0^T r_0 + \tilde{x}_0^T \breve{P}_0^{-1} \tilde{x}_0) + \lambda_1^T(-\tilde{A}_0 \tilde{x}_0 - \tilde{B}_0 r_0) \\ &\quad + \sum_{t=1}^{N-1} [\lambda_t^T \tilde{x}_t + \frac{1}{2}\tilde{x}_t^T M_t^T M_t \tilde{x}_t - \frac{1}{2}\gamma^2 r_t^T r_t + \lambda_{t+1}^T(-\tilde{A}_t \tilde{x}_t - \tilde{B}_t r_t)] \\ &\quad + \frac{1}{2}\tilde{x}_N^T M_N^T M_N \tilde{x}_N + \lambda_N^T \tilde{x}_N \qquad \text{(C.10)} \end{aligned}$$

where $\lambda_i; \quad i = 0, \ldots, N-1$ are the Lagrange multipliers associated with the dynamic constraints and $\frac{\gamma^2}{2}$ is the Lagrange multiplier of the disturbance constraint.

Taking the variation of J_2 and setting the variation $\delta J_2 = 0$ will give the extreme values of the performance index J_2.

$$\begin{aligned}
\delta J_2 &= \frac{\partial J_2}{\partial \tilde{x}_0}\delta\tilde{x}_0 + \frac{\partial J_2}{\partial \tilde{x}_t}\delta\tilde{x}_t + \frac{\partial J_2}{\partial r_t}\delta r_t + \frac{\partial J_2}{\partial \lambda_t}\delta\lambda_t \\
\frac{\partial J_2}{\partial \tilde{x}_0} &= -\gamma^2 \tilde{x}_0^T \breve{P}_0^{-1} - \lambda_1^T \tilde{A}_0 \\
\frac{\partial J_2}{\partial \tilde{x}_t} &= \sum_{t=1}^{N-1} (\tilde{x}_t^T M_t^T M_t - \lambda_{t+1}^T \tilde{A}_t + \lambda_t^T) \\
\frac{\partial J_2}{\partial r_t} &= \sum_{t=0}^{N-1} (-\gamma^2 r_t^T - \lambda_{t+1}^T \tilde{B}_t) \\
\frac{\partial J_2}{\partial \lambda_t} &= \sum_{t=0}^{N-1} (\tilde{x}_{t+1} - \tilde{A}_t \tilde{x}_t - \tilde{B}_t r_t)^T
\end{aligned}$$

Setting $\delta J_2 = 0$, we will get the following equations :

$$r_t^* = -\gamma^{-2} \tilde{B}_t^T \lambda_{t+1} \tag{C.11}$$

$$\tilde{x}_{t+1} = \tilde{A}_t \tilde{x}_t + \tilde{B}_t r_t \tag{C.12}$$

$$\tilde{x}_0^* = -\gamma^{-2} \breve{P}_0 \tilde{A}_0^T \lambda_1 \tag{C.13}$$

$$-\lambda_t = M_t^T M_t \tilde{x}_t - \tilde{A}_t^T \lambda_{t+1} \tag{C.14}$$

where r_t^* and $\tilde{x}_0^*$ are quantities that maximize J_2. From equations (C.12)- (C.14), we can compose the Hamiltonian system based on the states $\tilde{x}_t$ and λ_t :

$$\begin{bmatrix} \tilde{x}_{t+1} \\ -\lambda_t \end{bmatrix} = \begin{bmatrix} \tilde{A}_t & -\gamma^{-2}\tilde{B}_t\tilde{B}_t^T \\ M_t^T M_t & -\tilde{A}_t^T \end{bmatrix} \begin{bmatrix} \tilde{x}_t \\ \lambda_{t+1} \end{bmatrix} \tag{C.15}$$

The solution of the linear state-space system incorporates the state transition matrix.

$$\begin{bmatrix} \tilde{x}_t \\ \lambda_t \end{bmatrix} = \begin{bmatrix} \Phi_{11}(t,0) & \Phi_{12}(t,0) \\ \Phi_{21}(t,0) & \Phi_{22}(t,0) \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \lambda_0 \end{bmatrix} \tag{C.16}$$

$$\begin{bmatrix} \tilde{x}_0 \\ \lambda_0 \end{bmatrix} = \begin{bmatrix} \Phi_{11}(0,t) & \Phi_{12}(0,t) \\ \Phi_{21}(0,t) & \Phi_{22}(0,t) \end{bmatrix} \begin{bmatrix} \tilde{x}_t \\ \lambda_t \end{bmatrix} \tag{C.17}$$

With $M_0 = 0$, we can rearrange equation (C.14) and combine the result from equation (C.17)

to obtain the following equations.

$$\begin{aligned} 0 &= \tilde{x}_0 + \gamma^{-2}\breve{P}_0\lambda_0 \\ &= \Phi_{11}(0,t)\tilde{x}_t + +\Phi_{12}(0,t)\lambda_t + \gamma^{-2}\breve{P}_0[\Phi_{21}(0,t)\tilde{x}_t + \Phi_{22}(0,t)\lambda_t] \\ &= [\Phi_{11}(0,t) + \gamma^{-2}\breve{P}_0\Phi_{21}(0,t)]\tilde{x}_t + [\Phi_{12}(0,t) + \gamma^{-2}\breve{P}_0\Phi_{22}(0,t)]\lambda_t \end{aligned}$$

Define

$$P_t = [\gamma^2\Phi_{11}(0,t) + \breve{P}_0\Phi_{21}(0,t)]^{-1}[\Phi_{12}(0,t) + \gamma^{-2}\breve{P}_0\Phi_{22}(0,t)] \tag{C.18}$$

then

$$\tilde{x}_t = -\gamma^{-2}P_t\lambda_t \tag{C.19}$$

P_t is an important variable for determining the gain of the estimator. In what follows, we can see that P_t can be expressed recursively. Recursive equation is required for real-time application. To derive the recursive expression of P_t, we combine the equations (C.14), (C.14), (C.13) and (C.12).

$$\begin{aligned} \tilde{x}_t &= -\gamma^{-2}P_t\lambda_t \\ \tilde{x}_t &= \gamma^{-2}P_t(M_t^T M_t\tilde{x}_t - \tilde{A}_{t+1}^T\lambda_{t+1}) \\ \tilde{x}_t &= -\gamma^{-2}(I - \gamma^{-2}P_tM_t^TM_t)^{-1}P_t\tilde{A}_t^T\lambda_{t+1} \\ \tilde{x}_{t+1} &= \tilde{A}_t(-\gamma^{-2}(I - \gamma^{-2}P_tM_t^TM_t)^{-1}P_t\tilde{A}_t^T\lambda_{t+1}) + \tilde{B}_t(-\gamma^{-2}\tilde{B}_t^T\lambda_{t+1}) \\ \tilde{x}_{t+1} &= -\gamma^{-2}\{\tilde{A}_t(I - \gamma^{-2}P_tM_t^TM_t)^{-1}P_t\tilde{A}_t^T + \tilde{B}_t\tilde{B}_t^T\}\lambda_{t+1} \end{aligned}$$

Substituting $\tilde{x}_{t+1}$ with that of equation (C.19) expressed at time $t+1$ gives

$$\begin{aligned} -\gamma^{-2}P_{t+1}\lambda_{t+1} &= -\gamma^{-2}\{\tilde{A}_t(I - \gamma^{-2}P_tM_t^TM_t)^{-1}P_t\tilde{A}_t^T + \tilde{B}_t\tilde{B}_t^T\}\lambda_{t+1} \\ &= -\gamma^{-2}\{\tilde{A}_t(P_t^{-1} - \gamma^{-2}M_t^TM_t)^{-1}\tilde{A}_t^T + \tilde{B}_t\tilde{B}_t^T\}\lambda_{t+1} \end{aligned}$$

The Riccati equation (in recursive form) of P_t will be

$$P_{t+1} = \tilde{A}_tH_t^{-1}\tilde{A}_t^T + \tilde{B}_t\tilde{B}_t^T \tag{C.20}$$

$$H_t = P_t^{-1} - \gamma^{-2}M_t^TM_t \tag{C.21}$$

$$P_0 = \breve{P}_0 \tag{C.22}$$

$$\begin{aligned} \lambda_N &: \textit{arbitrary} \\ t &= 0, \ldots, N-1 \end{aligned} \tag{C.23}$$

Both r_t^* and $\tilde{x}_0^*$ are maximizing values, however, the proof of this statement is not included here. The proof can be obtained from [31]. Define the new performance index

$$J_3 = \max_{r, x_0} J_1 \tag{C.24}$$

We need to derive the values of K that will minimize J_3 to satisfy equation (C.5) . Taking the derivative of J_3 with respect to K and setting the derivation equal to zero will give K that maximizes the performance index J_3.

$$\begin{aligned} J_3 &= \max_{r, x_0} \frac{1}{2} \sum_{t=0}^{N-1} \tilde{x}_{t+1}^T M_{t+1}^T M_{t+1} \tilde{x}_{t+1} \\ &= \max_{r, x_0} \frac{1}{2} \sum_{t=0}^{N-1} (\tilde{A}_t \tilde{x}_t + \tilde{B}_t r_t)^T M_{t+1}^T M_{t+1} (\tilde{A}_t \tilde{x}_t + \tilde{B}_t r_t) \\ &= \frac{\gamma^{-4}}{2} \sum_{t=0}^{N-1} \lambda_{t+1}^T (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T)^T M_{t+1}^T M_{t+1} \\ &\quad (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T) \lambda_{t+1} \\ &= \frac{\gamma^{-4}}{2} \sum_{t=0}^{N-1} tr[(\lambda_{t+1}^T \lambda_{t+1}) (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T)^T M_{t+1} M_{t+1}^T \\ &\quad (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T)] \\ \frac{\partial J_3}{\partial K_t} &= \gamma^{-4} \sum_{t=0}^{N-1} tr[(\lambda_{t+1}^T \lambda_{t+1}) (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T)^T M_{t+1} M_{t+1}^T \\ &\quad \frac{\partial}{\partial K_t} (\tilde{A}_t H_t^{-1} \tilde{A}_t^T + \tilde{B}_t \tilde{B}_t^T)] \end{aligned}$$

The minimizing K can be obtained by setting

$$\frac{\partial J_3}{\partial K_t} = 0. \tag{C.25}$$

The derivative in equation (C.25) vanishes for

$$\frac{\partial}{\partial K_t} [(A_t - K_t C_t) H_t^{-1} (A_t - K_t C_t)^T + (B_t - K_t D_t)(B_t - K_t D_t)^T] = 0. \tag{C.26}$$

Solving the equation (C.26) for K_t gives

$$K_t^* = [B_t D_t^T + A_t H_t^{-1} C_t^T][D_t D_t^T + C_t H_t^{-1} C_t^T]^{-1} \tag{C.27}$$

Equation (C.27) provides the minimizing K_t since the Hessian of J_3 with respect to K_t is positive definite under the condition that $D_t D_t^T + C_t H_t^{-1} C_t^T$ is positive definite.

C.2 Derivation of Robust Estimator for Uncertain System and Noise Models

The state-space representation of the nominal model and the estimator from the Chapter 4 is repeated here.

$$x_{t+1} = A_t x_t + B_t d_t \tag{C.28}$$

$$\epsilon_t = S_t x_t + T_t d_t \tag{C.29}$$

$$e_t = M_t(x_t - \hat{x}_t) \tag{C.30}$$

$$y_t = C_t x_t + D_t d_t \tag{C.31}$$

where

$$\tilde{x}_0 = x_0 - \hat{x}_0$$

$$d_t = [r_t^T \quad \eta_t^T]^T$$

In Chapter 4, the performance criterion $\bar{J}_1$ is given in equation (4.33). The constrained optimization of $\bar{J}_1$ is solved through game theoritic optimization. This optimization requires modification of the performance index $\bar{J}_1$ to the new performance index $\bar{J}_2$. The game theoritic optimization problem formulation used to derive the estimator is stated in the following mathematical expression

$$\min_{\hat{x}} \max_{d, x_0} \bar{J}_2 \tag{C.32}$$

subject to equations (C.28)- (C.31) as constraints, with

$$\bar{J}_2 = \frac{1}{2}\|e\|^2 + \frac{1}{2}\|\epsilon\|^2 - \frac{\gamma^2}{2}(\|d\|^2 + \|x_0 - \hat{x}_0\|^2_{\breve{P}_0^{-1}}) \tag{C.33}$$

The weighted 2-norm of x with time-varying weight can be described as follows :

$$\begin{aligned} \|x\|_{X_t}^2 &= \sum_{t=0}^{N-1} x_t^T X_t x_t \\ &= \sum_{t=0}^{N-1} x_{t+1}^T X_{t+1} x_{t+1} + x_0^T X_0 x_0 \end{aligned}$$

under the consdition that the weight at N is

$$X_N = 0.$$

Zero term can then be expressed as follows

$$0 = \frac{1}{2} \sum_{t=0}^{N-1} \left(x_{t+1}^T X_{t+1} x_{t+1} - x_t^T X_t x_t \right) + \frac{1}{2} x_0^T X_0 x_0 \tag{C.34}$$

We are interested in the second and the third terms of the performance index $\bar{J}_2$. Expanding those terms and adding with zero term obtained from equation (C.34) will give the expression for the worst case disturbance d_t^*.

$$\begin{aligned} \frac{1}{2}\|\epsilon\|^2 - \frac{\gamma^2}{2}\|d\|^2 &= \frac{1}{2} \sum_{t=0}^{N-1} \{(S_t x_t + T_t d_t)^T (S_t x_t + T_t d_t) - \gamma^2 d_t^T d_t\} + \\ &\quad \frac{1}{2} \sum_{t=0}^{N-1} (x_{t+1}^T X_{t+1} x_{t+1} - x_t^T X_t x_t) + \frac{1}{2} x_0^T X_0 x_0 \end{aligned} \tag{C.35}$$

The difference equation describing the state dynamics is used to substitute the x_{t+1} factor. We proceed the derivation by expanding the right hand side of the equation C.35 :

$$\begin{aligned} \frac{1}{2}\|\epsilon\|^2 - \frac{\gamma^2}{2}\|d\|^2 &= \frac{1}{2} \sum_{t=0}^{N-1} \{x_t^T [S_t^T S_t + A_t^T X_{t+1} A_t - X_t] x_t + \\ &\quad d_t^T [T_t^T T_t + B_t^T X_{t+1} B_t - \gamma^2] x_t + \\ &\quad 2 x_t^T [S_t^T T_t + A_t^T X_{t+1} B_t] d_t\} + \frac{1}{2} x_0^T X_0 x_0 \\ &= -\frac{\gamma^2}{2} \sum_{t=0}^{N-1} \{\gamma^{-2} x_t^T [X_t - S_t^T S_t - A_t^T X_{t+1} A_t] x_t + \\ &\quad d_t^T [I - \gamma^{-2} (T_t^T T_t + B_t^T X_{t+1} B_t)] d_t - \\ &\quad 2\gamma^{-2} x_t^T [S_t^T T_t + A_t^T X_{t+1} B_t] d_t\} + \frac{1}{2} x_0^T X_0 x_0 \end{aligned} \tag{C.36}$$

Define

$$Z_t = I - \gamma^{-2}[T_t^T T_t + B_t^T X_{t+1} B_t] \tag{C.37}$$

$$F_t = S_t^T + A_t^T X_{t+1} B_t \tag{C.38}$$

Note that Z_t is symmetric. Rewriting the equation (C.36) with factors introduced in equations (C.37) and (C.38) will give

$$\begin{aligned}
\frac{1}{2}\|\epsilon\|^2 - \frac{\gamma^2}{2}\|d\|^2 &= -\frac{\gamma^2}{2}\sum_{t=0}^{N-1}\{d_t^T Z_t d_t - 2\gamma^{-2} x_t^T F_t d_t + \\
&\quad \gamma^{-4} x_t^T F_t Z_t^{-1} F_t^T x_t\} + \frac{1}{2} x_0 X_0 x_0 \\
&= -\frac{\gamma^2}{2}\sum_{t=0}^{N-1}\{d_t^T Z_t^{T/2} Z_t^{1/2} d_t - 2\gamma^{-2} x_t^T F_t z_t^{-1/2} Z_t^{1/2} d_t + \\
&\quad \gamma^{-4} x_t^T F_t Z_t^{-1/2} Z_t^{-1/2} F_t^T x_t\} + \frac{1}{2} x_0 X_0 x_0 \\
&= -\frac{\gamma^2}{2}\sum_{t=0}^{N-1}\|Z_t^{1/2} d_t - \gamma^{-2} Z_t^{-1/2} F_t^T x_t\|^2 + \frac{1}{2} x_0 X_0 x_0
\end{aligned} \tag{C.39}$$

There are several things that should be noted here. First, to complete the square inside the summmation, we introduce the following equality :

$$F_t Z_t^{-1} F_t^T = X_t - A_t^T X_{t+1} A_t - S_t^T S_t. \tag{C.40}$$

If we rearrange the terms in equation (C.40), we will have the following Riccati equation :

$$X_t = A_t^T X_{t+1} A_t + S_t^T S_t + F_t Z_t^{-1} F_t^T \tag{C.41}$$

with the final condition

$$X_N = 0 \tag{C.42}$$

Second, the terms inside the summation on the right hand side of equation (C.39) will be defined as

$$\|\bar{d}\|^2 = \sum_{t=0}^{N-1}\|Z_t^{1/2} d_t - \gamma^{-2} Z_t^{-1/2} F_t^T x_t\|^2 \tag{C.43}$$

The last equality will lead us to the following result

$$\bar{d}_t = Z_t^{1/2} d_t - d_t^* \tag{C.44}$$

$$d_t^* = \gamma^{-2} Z_t^{-1/2} F_t^T x_t \tag{C.45}$$

where d_t^* is the worst possible disturbance.

The estimator for the time-varying system of equation (C.28) will be derived based on the worst possible disturbance d_t^*. If we substitute d_t in the state-space equation with d_t^* from equation (C.45), the new state-space description will be as follows.

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t (Z_t^{-1/2} d_t^* + Z_t^{-1/2} \bar{d}_t) \\ &= (A_t + B_t Z_t^{-1/2} \gamma^{-2} Z_t^{-1/2} F_t^T) x_t + B_t Z_t^{-1/2} \bar{d}_t \\ y_t &= C_t x_t + D_t (Z_t^{-1/2} d_t^* + Z_t^{-1/2} \bar{d}_t) \\ &= (C_t + D_t Z_t^{-1/2} \gamma^{-2} Z_t^{-1/2} F_t^T) x_t + D_t Z_t^{-1/2} \bar{d}_t \end{aligned}$$

The state-space equations in a new coordinate system are :

$$x_{t+1} = \bar{A}_t x_t + \bar{B}_t \bar{d}_t \tag{C.46}$$

$$y_t = \bar{C}_t x_t + \bar{D}_t \bar{d}_t \tag{C.47}$$

where

$$\bar{A}_t = A_t + B_t Z_t^{-1/2} \gamma^{-2} Z_t^{-1/2} F_t^T \tag{C.48}$$

$$\bar{B}_t = B_t Z_t^{-1/2} \tag{C.49}$$

$$\bar{C}_t = C_t + D_t Z_t^{-1/2} \gamma^{-2} Z_t^{-1/2} F_t^T \tag{C.50}$$

$$\bar{D}_t = D_t Z_t^{-1/2} \tag{C.51}$$

To derive the estimator based on the new state-space equations (C.48) - (C.51), a new performance index $\bar{J}_3$ is defined as follows to replace the performance index $\bar{J}_2$.

$$\bar{J}_3 = \|e\|^2 - \gamma^2 (\|\bar{d}\|^2 + \|x_0 - \hat{x}_0\|^2_{\check{P}_0^{-1}}) \tag{C.52}$$

The optimization problem then can be expressed as

$$\min_{\hat{x}} \max_{\bar{d},x_0} \bar{J}_3 \tag{C.53}$$

with the constraints of equations (C.48) - (C.51). This problem is equivalent with that expressed in equation (C.5), (C.1) - (C.3), (C.7). The solution for this optimization is given by the following equations :

$$\begin{aligned}
\hat{x}_{t+1} &= (\bar{A}_t - \bar{K}_t\bar{C}_t)\hat{x}_t + \bar{K}_t y_t && \text{(C.54)}\\
\bar{K}_t &= [\bar{B}_t\bar{D}_t^T + \bar{A}_t\bar{H}_t^{-1}\bar{C}_t^T][\bar{D}_t\bar{D}_t^T + \bar{C}_t\bar{H}_t^{-1}\bar{C}_t^T]^{-1} && \text{(C.55)}\\
\bar{H}_t &= \bar{P}_t^{-1} - \gamma^{-2}M_t^T M_t && \text{(C.56)}\\
\bar{P}_{t+1} &= (\bar{A}_t - \bar{K}_t\bar{C}_t)\bar{H}_t^{-1}(\bar{A}_t - \bar{K}_t\bar{C}_t)^T + \\
&\quad (\bar{B}_t - \bar{K}_t\bar{D}_t)(\bar{B}_t - \bar{K}_t\bar{D}_t)^T && \text{(C.57)}\\
\bar{P}_0 &= \breve{P}_0 && \text{(C.58)}\\
t &= 0,\ldots,N-1
\end{aligned}$$

References

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[2] Brent D. Appleby. *Robust Estimator Design Using the H_∞ and μ Synthesis.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1990.

[3] Brent D. Appleby and John R. Dowdle. "Robust Estimator Design using μ Synthesis". In *Proceedings of The 30th Conference on Decision and Control*, Brighton, England, December 1991. IEEE.

[4] Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes, Theory And Applications.* Prentice-Hall, 1993.

[5] Richard V. Beard. *Failure Accommodation in Linear Systems Through Self-Reorganization.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1971.

[6] R.G. Brown and P.Y.C. Hwang. *Introduction To Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc, 1992.

[7] Edward Y. Chow and Alan S. Willsky. "Analytical Redundancy and the Design of Robust Failure Detection Systems". *IEEE Transaction on Automatic Control*, 29(7):603–614, 1984.

[8] C.K. Chui and G. Chen. *Kalman Filterring With Real-Time Applications.* Springer-Verlag, 1991.

[9] Clark E. Cohen. *Attitude Determination Using GPS: Development of An All Solid-State Guidance, Navigation, and Control Sensor for Air and Space Vehicles Based On The Global Positioning Systems.* PhD thesis, Stanford University, Stanford, CA, 1992.

[10] Munther A. Dahleh and G.C. Verghese. "Dynamic Systems". 6.241 Lecture Notes, Fall 1996.

[11] Kevin C. Daly, Eliezer Gai, and James V. Harrison. "Generalized Likelihood Test for FDI in Redundant Sensor Configuration'. *Journal of Guidance and Control*, 2(1):9–17, January-February 1979.

[12] James C. Deckert, Mukund N. Desai, John J. Deyst, and Alan S. Willsky. "F-8 DFBW Sensor Failure Identification Using Analytic Redundancy". *IEEE Transaction on Automatic Control*, 1977.

[13] Mukund N. Desai, James C. Deckert, and John J. Deyst. "Dual-Sensor Failure Identification Using Analytical Redundancy". *Journal of Guidance and Control*, pages 213–220, May–June 1979.

[14] J.J. Deyst. "Real-Time Systems for Aerospace Vehicles". 16.840 Lecture Notes, Spring 1996.

[15] John J. Deyst, James V. Harrison, Eliezer Gai, and Kevin C. Daly. "Fault Detection, Identification, and Reconfiguration for Spacecraft Systems". *Journal of the Astronautical Sciences*, 1981.

[16] Bradford W. Parkinson et.al., editor. *Global Positioning System: Theory and Applications.* AIAA, INC, 1996.

[17] Paul M. Frank. "Fault Diagnosis in Dynamic Systems Using Analytical and Knowldege-based Redundancy - A Survey and Some New Results". *Automatica*, 1990.

[18] et.al. Garry J. Ballas. *μ-Analysis and Synthesis Toolbox User's Guide.* MathWorks, INC, 1995.

[19] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[20] Frederik Gustafsson. "A Two-Filter Off-Line Solution To Optimal Detection". Technical report, Linköping University, Sweden, 1991.

[21] Frederik Gustafsson. *Estimation of Discrete Parameters in Linear Systems.* PhD thesis, Linköping University, Sweden, 1992.

[22] Frederik Gustafsson. "The Marginalized Likelihood Ratio Test for Detecting Abrupt Changes". *IEEE Transactions On Automatic Control*, 41(1), January 1996.

[23] Frederik Gustafsson and Hakan Hjalmarsson. "Twenty-one ML Estimators for Model Selection". *Automatica*, 31(10):1377–1392, 1995.

[24] Steven R. Hall. Parity vector compensation for FDI. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1981.

[25] Steven R. Hall. *A Failure Detection Algorithm for Linear Dynamic System.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1985.

[26] Harold L. Jones. *Failure detection in linear systems.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1973.

[27] Richard Kornfeld. *Research in progress : Attitude determination with GPS-IMU.* PhD thesis, *-in progress*, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 1997.

[28] Huibert Kwakernaak and R. Sivan. *Linear Optimal Control Systems.* John Wiley, 1972.

[29] M. Labarrere. "Aircraft Sensor Failure : Detection by Analytic Redundancy". In M.G. Singh, editor, *Systems and Control Encyclopedia*, volume 1, pages 246–251. Pergamon Press, Oxford, 1987.

[30] Frank L. Lewis and Vassilis L. Syrmos. *Optimal Control.* John Wiley & Sons, INC, 1995.

[31] Rami S. Mangoubi. *Robust Estimation and Failure Detection for Linear Systems.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[32] Rami S. Mangoubi, Brent D. Appleby, and Jay Farrell. "Robust Estimation in Fault Detection". In *Proceedings of The 31st Conference on Decision and Control*, pages 2317–2322, Tucson, Arizona, December 1992. IEEE.

[33] Rami S. Mangoubi, Brent D. Appleby, and G.C. Verghese. "Robust Estimation for Discrete-Time Linear Systems". In *Proceedings of The American Control Conference*, pages 656–661, Baltimore, Maryland, June 1994. ACC.

[34] Rami S. Mangoubi, Brent D. Appleby, and G.C. Verghese. "Stochastic Interpretaion of H_∞ and Robust Estimation". In *Proceedings of The 33rd Conference on Decision and Control*, pages 3943–3948, Lake Buena Vista, FL, December 1994. IEEE.

[35] Rami S. Mangoubi, Brent D. Appleby, G.C. Verghese, and Wallace E. Vander Velde. "A Robust Failure Detection and Isolation Algorithm". In *Proceedings of the 34th Conference on Decision and Control*, pages 2377–2382, New Orleans, LA, December 1995. IEEE.

[36] I. Nikiforov, V. Varavva, and V. Kireichikov. "Application of Statistical Fault Detection Algorithms to Navigation Systems Monitoring". *Automatica*, 29(5):1275–1290, 1993.

[37] Fernando Paganini. *Sets and Constraints in the Analysis of Uncertain Systems.* PhD thesis, California Institute of Technology, Pasadena, CA, 1995.

[38] Jan Palmqvist. *On Integrity Monitoring of Integrated Navigation System.* Licentiae Thesis, Linköping University, Sweden, 1997.

[39] R.J. Patton and J. Chen. *Robust Fault Detection and Isolation Systems*, volume 74 of *Control and Dynamic Systems*, chapter 3. Academic Press, 1996.

[40] PBS. 21st century jet. Video Recording, PBS Home-Video, 1993.

[41] PBS. 777 first flight. Video Recording, PBS Home-Video, 1993.

[42] Asok Ray, Robert Geiger, Mukund Desai, and John J. Deyst. "Analytic Redundancy for On-Line Fault Diagnosis in a Nuclear-Reactor". *Journal Energy*, 7(4):367–373, July–August 1983.

[43] Karl Sabbagh. *21st Century Jet : The Making and Marketing of Boeing 777.* Scribner, 1996.

[44] Martin L. Shooman. *Probabilistic Reliability : An Engineering Approach.* McGraw-Hill, 1968.

[45] Gilbert Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Cambridge, Massachusetts, 1986.

[46] Harry L. Van-Trees. *Detection, Estimation and Modulation Theory, Part I.* John Wiley & Sons, Inc, 1968.

[47] George Verghese and Thomas Kailath. "A Further Note On Backward Markovian Models". *IEEE Transactions On Information Theory*, 25:121–124, 1979.

[48] A. Wald. *Sequential Analysis.* John Wiley & Sons, Inc, 1947.

[49] Alan S. Willsky. "A Survey of Design Methods for Failure Detection in Dynamic Systems". *Automatica*, 1976.

[50] Alan S. Willsky, G.W. Wornell, and J.H. Shapiro. "Stochastic Processes, Detection and Estimation". 6.432 Course Notes, Spring 1997.

[51] A.S. Willsky and H.L. Jones. "A Generalized Likelihood Ratio Approach To The Detection And Estimation Of Jumps In Linear Systems". *IEEE Transactions On Automatic Control*, pages 108–112, February 1976.