CRANFIELD UNIVERSITY

Huamin Jia

Data Fusion Methodologies for Multisensor Aircraft Navigation Systems

College of Aeronautics

PhD Thesis

Cranfield College of Aeronautics

Cranfield UNIVERSITY

PhD Thesis

Academic Year 2003-2004

Huamin Jia

Data Fusion Methodologies For Multisensor Aircraft Navigation Systems

Supervisor: Professor David J. Allerton

April 2004

This thesis is submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy

ABSTRACT

The thesis covers data fusion for aircraft navigation systems in distributed sensor systems. Data fusion methodologies are developed for the design, development, analysis and simulation of multisensor aircraft navigation systems. The problems of sensor failure detection and isolation (FDI), distributed data fusion algorithms and inertial state integrity monitoring in inertial network systems are studied.

Various existing integrated navigation systems and Kalman filter architectures are reviewed and a new generalised multisensor data fusion model is presented for the design and development of multisensor navigation systems. Normalised navigation algorithms are described for data fusion filter design of inertial network systems.

A normalised measurement model of skewed redundant inertial measurement units (SRIMU) is presented and performance criteria are developed to evaluate optimal configurations of SRIMUs in terms of the measurement accuracy and FDI capability. Novel sensor error compensation filters are designed for the correction of SRIMU measurement errors. Generalised likelihood ratio test (GLRT) methods are improved to detect various failure modes, including short time and sequential moving-window GLRT algorithms.

State-identical and state-associated fusion algorithms are developed for two forms of distributed sensor network systems. In particular, innovative inertial network sensing models and inertial network fusion algorithms are developed to provide estimates of inertial vector states and similar node states. Fusion filter-based integrity monitoring algorithms are also presented to detect network sensor failures and to examine the consistency of node state estimates in the inertial network system.

The FDI and data fusion algorithms developed in this thesis are tested and their performance is evaluated using a multisensor software simulation system developed during this study programme. The moving-window GLRT algorithms for optimal SRIMU configurations are shown to perform well and are also able to detect jump and drift failures in an inertial network system. It is concluded that the inertial network fusion algorithms could be used in a low-cost inertial network system and are capable of correctly estimating the inertial vector states and the node states.

ACKNOWLEDGEMENT

I would like to express my honest appreciation to Professor David J. Allerton for offering me an assistant research post to pursue my PhD study in data fusion methodologies for multisensor aircraft navigation systems at Cranfield University and to write up at The University of Sheffield, and for his greatest help, enthusiasm and encouragement during the past four years. I have benefited greatly from his supervision in many aspects, including the method and attitude of scientific research. All I learn from Dave has been and will be priceless in my PhD study and my future scientific career.

I would like to deeply thank all members of my PhD academic administrative panel, Professor Rade Vignjevic, Drs Stephen Hobbs and Peter Roberts for their valuable suggestions and advices during the undertaking of this study.

I also would like to thank all of the people in Cranfield University, who helped, encouraged and supported me during my PhD study. Special thanks are given to Professor John Fielding and Dr Xiang Zhang for their encouragement and help.

This work was supported by the European Commission under the Framework V programme, through the SHINE (Smart Hybrid Integrated Navigation Equipment) project (Contract No. G4RD-CT-2000-00227) and cooperation with Thales Avionics France. I am also grateful to Thales Avionics, France for their support and special thanks are given to Mr Marc Revol and Mr. Pierre-Jerome Clemenceau for helpful discussions and suggestions at each project progress meeting.

Private thanks are given to Mr Tony Clare and Dr Gramham in The University of Sheffield; Professors Housheng Hu in University of Essex and Demin Xiu in Northwestern Polytechnical University, China; Dr Sam Bose of Technalytics, USA Dr. A. Politopoulos of American GNC Corporation, USA, and Dr Christoph Eck in Swiss Federal Institute of Technology at Zurich.

Finally, I would like to thank my wife, daughter, my parents, my sisters and brother for their support, understanding and encouragement and for being a loving family.

To remember my grandmother-in-law

RESEARCH DELIVERIES

Work Done During the Course of My PhD Study

1. The SHINE system safety analysis
2. Development of sensor failure detection and isolation algorithms for skewed redundant inertial measurement units (SRIMU) and the SHINE integrity monitoring algorithms
3. Development of lab static and dynamic calibration methods for SRIMU
4. Design and development the hybridised multi-mode redundant AHRS/GNSS navigation filter
5. Development of software simulation system in Matlab for test and evaluation of the SHINE SRIMU configurations, SHINE FDI algorithms and multi-model integrated navigation filter

Papers

1. *New Distributed Data Fusion Algorithms for Inertial Network Systems*, submitted to IEEE Trans. On Automatic Control.
1. *Application of A Multi-Mode Kalman Filter to Redundant Inertial/ Multifunctional GNSS Navigation System*, submitted to IEEE Trans. On Aerospace and Electronics Systems.
2. *Multisensor Fusion Methodologies for Aircraft Navigation Systems: Architectures and Algorithms*, submitted to The Royal Institute of Navigation.
3. *Enhancement of GLRT Functions using SRISS Compensation Filters*, to be submitted to some technical journals.
4. *An Error Compensation Method for Redundant Inertial Configuration*, ION 58th Annual Meeting and the CIGTF 21st Guidance Test Symposium, June 24-26, 2002, Albuquerque, USA.

RESEARCH DELIVERIES

Research Reports

1. *Literature Survey on Redundant IMU Architectures and FDI Techniques*, Research Report submitted to Thales Avionics, France, Project Ref. NAV/01/011864-00, July, 2000.
2. *Wander-Azimuth Strapdown Inertial Navigation Algorithm and Software Design in Modula-2*, Sept., 2000.
3. *Literature Survey on Integrated Kalman Filter Design, GNSS Attitude Determination and Ambiguity Resolution Techniques*, Research Report submitted to Thales Avionics, France, Project Ref: NAV/01/011685-00, Feb., 2001.
4. *GPS Error Modelling and Software Simulation in Modula-2*. May 2001
5. *Functional Hazard Analysis for Future SHINE Product*, Research Report submitted to Thales Avionics, France, Project Ref. NAV/01/010790-00, Nov., 2001.
6. *Development of SRIMU Failure Detection and Isolation System for Future SHINE Product*, Research Report submitted to Thales Avionics, France, Project Ref. NAV/02/010952-00, Aug., 2002.
7. *Redundant AHRS/GNSS Hybridised Navigation Filter Design*, Research Report submitted to Thales Avionics, France, May, 2003.
8. *SHINE Software Simulation System in Matlab*, Simulation Software Package submitted to Thales Avionics, France, May 2003.
9. *SHINE Software Simulation Environment*: *User Manual*, Research Report submitted to Thales Avionics, France, May 2003.

Presentations

1. *SHINE System Safety Analysis*, Presentation at Thales Avionics, France, June 2001
2. *FDI and SRIMU Calibration Methods*, Presentation at Thales Avionics, France, Oct., 2002.
3. Redundant AHRS/GNSS Hybridised Navigation Filter Design, Presentation at Thales Avionics, France, Oct., 2002.

GLOSSARY

ADS	Air Data System
AHRS	Attitude and Heading Reference System
AAIM	Aircraft Autonomous Integrity Monitoring
BIH	the Bureau International de l'Heure
CTP	Conventional Terrestrial Pole
DOF	Degree-Of-Freedom
DF	Detection Function
DR	Dead-Reckoning
FDI	Failure Detection and Isolation
GLRT	Generalised Likelihood Ratio Test
GLONASS	Global Orbital Navigation Satellite System
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IMA	Integrated Modular Avionics
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IRS	Inertial Reference Systems
LOP	Lines Of Position
LOS	Lines Of Sight
LRM	Line Replaceable Modules
LRU	Line Replaceable Unit
MSDF	Multi-Sensor Data Fusion
NSIM	Navigation Solution Integrity Monitoring
NQI	Normalised Quadratic Innovation
NQR	Normalised Quadratic Residual
MW-GLRT	Moving-Window GLRT
PVAT	Position, Velocity, Attitude, and Time
PR	Pseudorange
PRR	Pseudorange Rate

GLOSSARY

SPRT	Sequential Probability Ratio Test
SRIMU	Skewed Redundant Inertial Measurement Unit
RAIM	Receiver Autonomous Integrity Monitoring
RNP	Required Navigation Performance
WGS-84	World Geodetic System of 1984

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Aircraft Navigation Sensors/Systems

The purpose of aircraft navigation is to determine significant position, velocity, attitude, and time (PVAT) information of an aircraft with respect to reference frames. These PVAT parameters are referred as to the navigation states in this thesis.

Aircraft navigational sensor systems, which measure the dynamic motion of an aircraft with reference to specific frames, provide continuous inertial data and other measurement information that is required by onboard avionics systems for the implementation of various functions, including aircraft flight control and guidance, navigation computation and attitude determination, flight management and display, local motion compensation and inertial system correction and alignment, as well as air traffic management. A navigational sensor measures quantities related to one or more elements of the navigation states. A set of navigational sensors, which is able to determine all the navigation states by using appropriate navigation algorithms, makes up a navigation system.

An aircraft navigation system combines all the measurement information from the navigational sensor systems of an aircraft to determine the following parameters and information:

- Kinematic parameters (accelerations and angular rates)
- Navigation states
- Trajectory and track parameters

- System health status information.

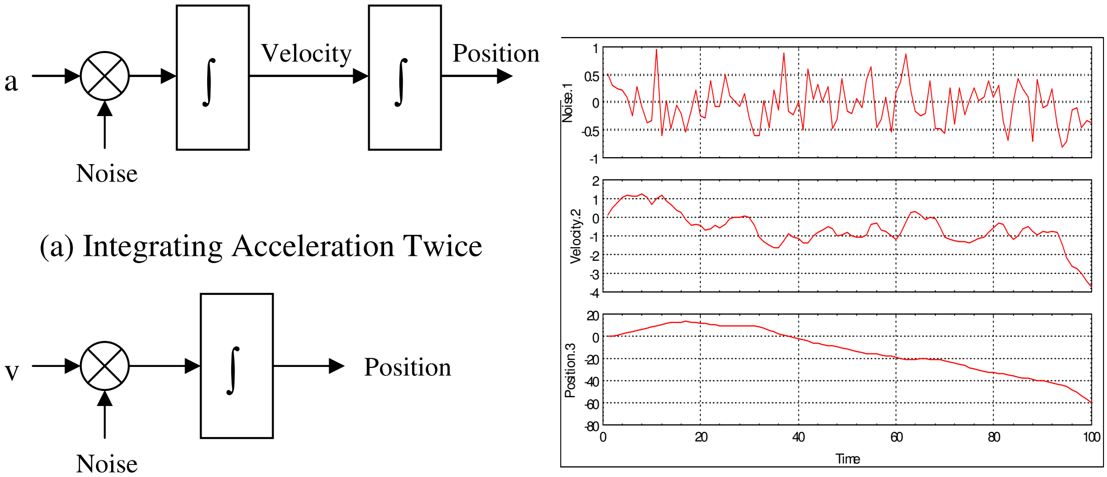
The main navigation sensors/systems used by aircraft are summarised in Table 1-1 from the literature survey. These aircraft navigation sensors/systems can be categorised as two types: self-contained navigation systems and external aiding navigation systems. The self-contained navigation systems perform the navigation functions independent of external signals. In contrast, the external aiding navigation systems implement the navigation functions through reception of signals from and/or transmission of signals to external systems. These two types of navigation systems are examined in the following subsections.

Table 1-1 Aircraft Navigation Sensors/Systems

Navigation Systems	System		Sensors		Coordinates
	Subsystems	Derived States	Sensors	Raw Data	Raw Measurements
Self-Contained Navigation Systems	INS, AHRS	Position Velocity Acceleration Attitude	Inertial sensors	Accelerations and angular rates	Inertial instrument frame
	Air Data System	Mach Airspeed Pressure altitude	Air data sensor, Baro-altimeter, Air speed sensor	Static and dynamic pressures, air speed	Air mass/wind reference frame
	Heading Indicator	Heading	Magnetic heading sensor	Earth magnetic field components	
	Radar Altimeter	Height above ground	Radar altimeter	Range	Radar antenna frame
	Doppler Radar	Ground velocity	Doppler radar	Relative LOS range rate	Radar antenna frame
External Aiding Navigation Systems	Space-Based Navigation Systems	Position Velocity Time, Attitude	GNSS receiver	LOS range and range rate	WGS84 reference frame
	Ground-Based Navigation System	Location Height Angles	VOR, LORAN, VOR/DME, ILS	Relative range and angle	Relative reference frame
	Relative Navigation System	Position Velocity	MIDS (JTIDS) PLRS	Relative range and range rate	WGS-84 and Relative grid frame

1.1.1 Self-Contained Navigation Systems

A self-contained navigation system is a system that computes aircraft position, velocity and attitude relative to a reference frame by means of dead-reckoning (DR) techniques without reception of externally generated signals. Using DR techniques, aircraft velocities are determined by integrating the measured aircraft accelerations from known initial velocities. Aircraft position is obtained by integrating the aircraft velocity from a known initial position. Typical DR procedure for a single axis case is illustrated in Figure 1.1 where all initial values are zero. Position and velocity errors caused by white noise sensor errors are shown in Figure 1.1(c). This DR procedure continuously accumulates sensor errors so that the navigation state errors grow over time and are unbounded unless they are constrained by aiding navigation systems. This characteristic is a vital limitation of all self-contained navigation systems.



(a) Integrating Acceleration Twice

(b) Integrating Velocity

(c) DR Errors caused by white noise

Figure 1.1 Dead Reckoning Procedure

The sensor systems applied for self-contained navigation systems are typically inertial sensor systems, air data sensor systems and Doppler radar. An air data system provides altitude with respect to mean sea level and true air speed. Doppler radar can measure aircraft velocity relative to the ground by transmitting a radar beam to and receiving the echo beam from the ground. But, Doppler radar signals are susceptible to interference from external signals or the environment. Doppler radar and air data system cannot provide all the navigation states, whereas an inertial system alone can determine all the navigation states.

Two basic inertial mechanisations are used to implement an inertial navigation system (INS). The first method is known as a stable platform system where a set of mutually orthogonal accelerometers is mounted on a gimballed gyro platform. The gyros sense the angular rate of the platform and control the gimbal servos so that the platform maintains a stable platform orientation with respect to a known reference frame irrespective of the aircraft rotation. The gimbal angles provide a direct readout of aircraft attitude angles. The accelerometer triad on the platform provides aircraft accelerations relative to the known reference frame. Integration of the accelerations can derive the velocity and position of an aircraft. The second method is referred to as strapdown inertial system where gyros and accelerometers are mounted on a rigid frame that is strapped down to an aircraft. The inertial sensors measure accelerations and angular rates of the aircraft relative to inertial space. The aircraft attitude angles are then derived by performing a so-called analytical platform algorithm, commonly known as the strapdown attitude determination algorithm. The accelerometer outputs are transformed to this analytical platform frame and are then integrated to obtain the velocity and position in a navigation reference frame.

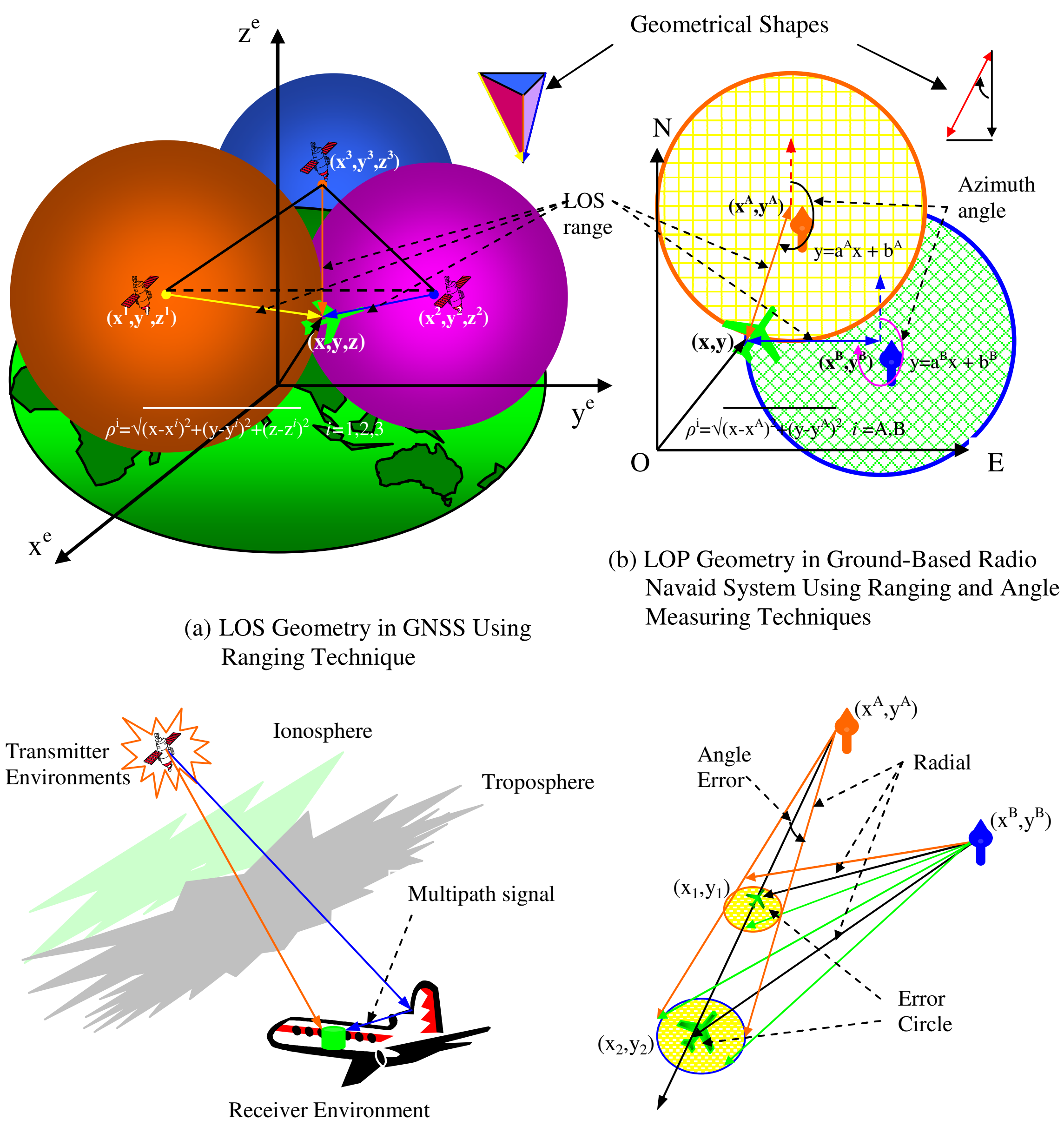
Although inertial systems exhibit some disadvantages of the dead reckoning method, their high dynamic characteristics and short-term measurement accuracy are ideal for aircraft attitude determination and flight control systems. In addition, other airborne avionics systems require inertial information to stabilise and compensate for local motion.

1.1.2 External Aiding Navigation Systems

An external aiding navigation system is a radio navigation system and consists of two parts: airborne subsystems and external signal source systems. An airborne subsystem is a signal-processing unit, which receives and processes the coded signals transmitted by external signal sources to facilitate position fixing. An external signal source system is typically a network of transmitters that transmit coded signals and can be further classified as ground-based radio navaid systems (e.g., VOR/DME, ILS and LORAN) and space-based navigation systems, also known as Global Navigation Satellite Systems (GNSS). Two communication modes are used in external aiding navigation systems: one-way and two-way modes. In the one-way communication mode, an airborne subsystem passively receives signals and data from an external signal source system whereas in the two-way mode, an airborne subsystem actively transmits signals and receives replies from external signal sources. External aiding navigation systems are usually based on an algebraic geometry principle to determine the aircraft navigation states. The geometry is shaped by lines of sight (LOS) or lines of position (LOP) from external signal sources to an airborne receiver, as depicted in Figure 1.2(a) and (b). The coordinates of the points, which are the positions of the aircraft and external transmitters, are represented by a set of nonlinear or linear algebraic equations. The forms of and the constraints on the algebraic equations depend on the navigation mechanisms of external aiding navigation systems.

Navigation mechanisms applied to external aiding navigation systems are primarily based on the timing/ranging techniques, angle measurement and Doppler techniques. The angle measuring technique measures the azimuth angle of an aircraft with respect to an external reference transmitter and is usually used in ground-based radio navaid systems. In other words, this method computes the direction of a radial line from the transmitter to the aircraft; that is, the coefficient of a linear algebraic equation, as illustrated in Figure 1.2 (b). Therefore, the position of an aircraft is the solution of a set of linear algebraic equations. Two transmitters provide a unique fix in angle measuring systems. As a result, the uncertainty of aircraft location caused by the measurement errors increase with distance from the aircraft to the transmitters, as shown in Figure 1.2(d). VOR/DME is a typical angle/range measurement navigation system.

The Doppler positioning technique, which measures the rates of changes of the relative ranges along the signal LOS between an aircraft and external signal sources, was used in the first generation of GNSS, known as the Transit system. The Doppler technique can provide an accurate velocity measurement. However, the uncertainties of position solutions, caused by integrating the Doppler measurement errors, increase over time. For example, the positioning accuracy of the Transit system degraded with time.



(a) LOS Geometry in GNSS Using Ranging Technique

(b) LOP Geometry in Ground-Based Radio Navaid System Using Ranging and Angle Measuring Techniques

(c) Signal Propagation and Error Sources in GNSS

(d) Location Uncertainty in Angle Measuring System

Figure 1.2 Principles of External Aiding Navigation Systems

The timing/ranging techniques use the principle of elapsed time measurement as the basis for the LOS range measurements. The elapsed time is the time difference between the time at which the ranging signal is transmitted by an external transmitter and the time at which it is detected by an airborne receiver. Several timing/ranging techniques have been applied to ground-based radio navigation systems and space-based navigation systems, including LORAN and GPS. The LOP geometry of GPS is the surface of a sphere whereas the LORAN system is a location hyperbola. Hence, the position of an aircraft is computed in terms of the solution of nonlinear algebraic equations.

In comparison with dead reckoning techniques, a significant advantage of the timing/ranging techniques is that the accuracy of the navigation systems based on the timing/ranging techniques does not degrade over time or distance because the navigation states are derived from a set of nonlinear algebraic equations rather than a set of integral equations. Moreover, the uncertainty of a position solution is restricted to a circle or hyperbola of location or the surface of a position sphere instead of the radial line in the angle measuring systems. The Doppler positioning technique can be also combined with the timing/ranging techniques used in GNSS navigation systems. Consequently, GNSS affords long-term stability of accuracy for the position and velocity solution. A GNSS receiver is inexpensive, small size and low power. It is these advantages that make GNSS an ideal external navigation system to aid all self-contained navigation systems, particularly inertial systems.

The accuracy of external aiding navigation systems is affected by the geometry of the positions of aircraft and external transmitters[1]. In space-based navigation systems, the radio ranging signals transmitted by satellites propagate through the atmosphere to airborne receivers, the signal dispersion and refraction caused by the ionosphere and troposphere introduce signal propagation path delays in the range measurements, as shown in Figure 1.2 (c). In addition, the uncertainty of satellite orbits, and satellite and receiver clock errors also introduce range measurement errors. As a result, the measured time difference is not perfect and the resultant range is known as the pseudorange.

External aiding navigation systems and other self-contained navigation systems (such as Doppler radar) are generally used to aid inertial navigation systems. Such systems are referred to as navaid systems in this thesis.

1.1.3 Required Navigation Performance

The concept of Required Navigation Performance (RNP) was established by the International Civil Aviation Organization (ICAO) to develop aircraft navigation standards for all phases of aircraft operations. In ICAO Document 9650, RNP is defined as *a statement of the navigation performance accuracy, integrity, continuity and availability necessary for operations within a defined airspace.* RNP can include both performance and functional requirements, which is indicated by the RNP type. The RNP types specify the minimum navigation performance accuracy required in an airspace. These standards are intended for system designers, manufacturers, and installers of avionics equipment, as well as service providers and users of the systems for global operations. Four primary parameters are used to define RNP requirements: accuracy, integrity, continuity and availability, and their definitions in this thesis are based on published descriptions[2][3][4][5].

RNP accuracy is defined in terms of the total system error (TSE) with respect to the reference flight trajectory required for each phase of flight. The TSE comprises two error components: flight technical errors and navigation system errors. The accuracy requirement is for the TSE to remain within a normal performance region, under fault-free conditions, at least 95% of the time.

RNP integrity is defined as a measure of the trust which can be placed on the correctness of the information supplied by a navigation system. Integrity includes the ability of a navigation system to provide timely and valid alerts to flight crew when the navigation system must not be used for its intended purpose. Integrity risk is the probability that an undetected failure results in the TSE exceeding the containment region.

RNP continuity is the ability of a navigation system to perform navigation functions without interruption during a certain period of time. Continuity risk is the probability that a navigation system will be interrupted and will be unable to provide navigation information over the intended period of operation. More specially, continuity is the probability that the navigation system will be available for the duration of operation.

RNP availability is an indication of the ability of a navigation system to supply usable service within a specified coverage area, and is defined as the portion of time that reliable navigation information is presented to the flight crew. Availability is specified in terms of the probability of the navigation function being available at the beginning of the intended operation.

RNP accuracy and integrity are achieved by developing innovative data fusion methods while RNP continuity and availability are satisfied by fault-tolerant design. In this thesis, fault tolerance is the ability of an aircraft navigation system to continue satisfactory operation in the presence of one or more hardware or software failures.

The aim of this thesis is to investigate data fusion methodologies for the design and development of aircraft multisensor navigation systems in order to fulfil the RNP requirements.

1.2 Multisensor Data Fusion

1.2.1 The Concept of Multisensor Data Fusion

Data fusion refers to the combination of data from a variety of sensors that are able to act in cooperation such that the total effect is greater than the sum of effects taken independently. The concept of multisensor data fusion (MSDF) was initially developed for military applications[6-8] and afterwards applied to civil industries[9-12], including battlefield surveillance, automatic multi-target tracking and recognition, guidance and control of autonomous vehicles and robotic systems. Traditionally, multisensor data fusion is considered as a data/information processing technology, covering a wide range of disciplines, for example, estimation and identification theory, control engineering, statistics and decision theory, signal processing and pattern recognition, artificial intelligence and knowledge engineering. Owing to the multidisciplinary nature of multisensor data fusion and a wide range of applications, researchers have described the concept of multisensor data fusion from diverse perspectives, focusing on either the description of functions to be completed or data processing methods used by multisensor data fusion systems. In order to improve communications among researchers and system developers, the US Joint Directors of Laboratories (JDL) Data Fusion Working Group has developed a functional model of multisensor data fusion and defined multisensor data fusion as *a multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources*[7][9]. The JDL model and definition have been accepted by many data fusion researchers and primarily served for military applications, for example, command, control, communication, computer and intelligence (C^4I) systems. Other application-oriented MSDF models[10][11], established by the US National Institute of Standard and Technology (NIST), are mostly used for industrial control systems and intelligent systems, such as robotic systems. The JDL and NIST models, as well as other models are summarised by Kokar and Kim[12], who have identified three major sources of misunderstanding about multisensor data fusion, including lack of precise methods and standards to represent multisensor data fusion architectures, low-level design solutions against the common practice of software engineering and definitions of multisensor data fusion.

However, these proposed models are not directly applicable to the design of multisensor aircraft navigation systems because they ignore the consideration of selections of sensor systems and architectures that are the basis of fault tolerance of aircraft navigation systems. Moreover, the functional descriptions are not concerned with methods to fulfil the RNP requirements.

1.2.2 Data Fusion for Aircraft Navigation Systems

Traditionally, the terms *integrated (integration), combined (combination)* and *hybridised (hybridisation)* are used to describe multisensor-based aircraft navigation systems. Integration (or combination) of multiple independent navigation systems for aircraft navigation is referred to as fault-tolerant design[13] and the resultant system is known as a fault-tolerant navigation system. The integration of multiple cooperative sensors to form a navigation system is known as an integrated navigation system. These two forms of aircraft navigation systems have been developed since the 1970s.

More interests have also been given to both the development of fault-tolerant system architectures[14-23] and the improvement of integrated filter architectures and filtering algorithms[42-64]. However, the traditional fault-tolerant navigation systems are a static fault tolerance design where multiple navigation systems are structured in a federated architecture and the system fault detection is completed by use of simple weighted mean methods or majority voting methods. This traditional approach to design cannot effectively exploit the advantages of redundant sensor systems and a fault-tolerant system cannot be dynamically reconfigured by using redundant sensor systems.

A recent approach is the use of distributed modular avionics architecture where multiple inertial sensor systems are located in several positions in an aircraft in order to increase survivability and provide the localised compensation for other airborne avionic systems[20][33]. This distributed architecture affords an enhanced level of fault tolerance by reconfiguration and sharing spare computing resources, which can be dynamically allocated to functioning sensor systems.

The term *multisensor data fusion* used in aircraft navigation applications has appeared in recent years with the advent of

- Low-cost, small-size and low-mass navigation sensors (e.g. optical gyros, MEMS inertial sensors and GNSS sensors),
- High-speed, large memories and embedded microprocessors, and
- Distributed and integrated modular avionics architectures.

Significant advancements in the inertial sensor technologies and predictable improvements in the performance, low cost, small size and low mass of the new generation of inertial sensors will enable widespread use of inertial sensor networks integrated with navaid systems (especially GNSS) in many commercial and military aircraft systems. The use of an inertial network architecture not only improves the accuracy and fault tolerance of aircraft navigation systems, but also increases the survivability of the navigation system and provides local motion compensation and stabilisation for other avionic systems. The novel integration of emerging navigation sensor technologies and distributed modular avionics architectures based on high-speed data buses and embedded microprocessors will change the traditional methods used in the design and development of aircraft navigation systems.

However, the literature survey undertaken in this thesis has not identified any rational definitions or comprehensive MSDF models, which can be used to guide researchers and engineers to develop aircraft multisensor navigation systems. For example, Kayton and Friend, Huddle and Brown[1], in the book Avionics Navigation System, describes multisensor navigation as *a process of estimating the navigation variables of position, velocity, and attitude from a sequence of measurement from more than one navigation sensor*. Obviously this definition implies development of various novel state estimate algorithms, but it does not clearly indicate how to design and develop an aircraft multisensor navigation system.

This thesis treats multisensor data fusion as *a system engineering methodology that can guide system developers, by using appropriate sensor allocations, failure detection and isolation techniques and data fusion algorithms, to design, develop and implement a highly reliable multisensor-based navigation system in order to obtain required navigation system performance in terms of accuracy, integrity, continuity and availability*. This definition covers the whole system design process from system requirements to system architecture design.

1.3 Aims

As the existing multisensor data fusion models are either application-oriented intelligent systems or military C^4I systems, it is proposed to develop a generalised MSDF model for aircraft navigation systems. This model will provide a framework for system engineers and researchers to design and develop multisensor navigation systems.

A further motivation for this study is the emerging concepts and technologies in aviation, including seamless navigation/positioning and free flight concepts, and the applications of MEMS inertial technologies and integrated modular avionics architectures. A major development, which underpins the recent developments in navigation systems, is global navigation satellite systems (GNSS). These concepts and technologies will be used in future Air Traffic Management/Communication

Navigation Surveillance (ATM/CNS) systems. The core of these innovative concepts is the techniques for high precision positioning. It is expected that modern navigation systems, based on the fusion of various multiple redundant navigation sensors, can provide 10 degree-of-freedom (DOF) parameters in a 4-dimensional space, including time, position and velocity, and attitude values. In addition, techniques developed for the MSDF-based navigation systems can also be applied to the spacecraft industries and intelligent transportation systems.

Recently, a great deal of interest has arisen in manufacturing processes that allow the monolithic integration of MEMS with driving, controlling, and signal processing electronics. With the development of MEMS inertial sensors and high-speed and large memory microprocessor, complex data fusion algorithms and multi-state sensor error dynamic models will be able to be implemented in a single microprocessor in a distributed integrated modular avionics architecture.

This study also originated from an EU Framework 5 project, the SHINE (smart hybridised integrated navigation equipment) programme, which was to develop a low-cost redundant inertial/GNSS-based attitude integrated navigation system for aircraft. In this project, the author was responsible for performing the SHINE system safety analysis, evaluation of the different SRIMU configurations, and development and simulation of multi-model Kalman filtering algorithms and FDI algorithms for SHINE system. During my PhD study, these researches were further extended into the development of inertial network data fusion algorithms for wider applications of airborne distributed inertial systems. Most research results obtained from this PhD programme were delivered into the SHINE project.

1.4 Research Objectives

This thesis examines the problem of multisensor data fusion for aircraft navigation in distributed sensor network systems and investigates data fusion methodologies for the design, analysis, development and simulation of multisensor aircraft navigation systems. It is expected that such multisensor navigation systems can improve the system reliability and the navigation performance in terms of accuracy, integrity, availability and continuity and enhance the fault tolerance of aircraft navigation systems by using FDI and integrity monitoring techniques. The specific objectives of the study programme are:

- To gain a theoretical understanding of the problems of multisensor data fusion for aircraft navigation and to develop a generalised multisensor data fusion model for aircraft navigation systems.
- To investigate methods for evaluation and analysis of various architectures of redundant sensor configurations and to develop error dynamic models for skewed redundant inertial measurement units (SRIMU).
- To establish the normalised navigation and attitude determination equations of inertial reference systems and other navaid systems, and to analyse their error dynamics.
- To develop methods for the detection and isolation of various sensor failures and for monitoring of the integrity of the navigation states and inertial vector state in inertial network systems in order to ensure the safety of multisensor aircraft navigation systems.
- To develop innovative distributed data fusion algorithms in order to enhance the accuracy of the distributed inertial states and navigation states estimates.
- To develop a simulation system for the evaluation of the performance of inertial sensors of varying quality in an inertial network system, and FDI and distributed data fusion filter algorithms developed in this thesis.
- To undertake a series of case studies and simulations of sensor configurations.

This thesis will contribute new understanding to the design methodologies used in the integration of distributed low cost sensors for aircraft navigation. The research programme will cover the development of software tools for multisensor data fusion and performance analysis, and provide insight into the effectiveness of these systems in the form of simulation models of sensor systems and navigation systems.

1.5 Outline of the Thesis

Chapter 2 reviews traditional data fusion methods and architectures applied to integrated navigation systems, including various fault-tolerant navigation system architectures, data fusion filter architectures and filtering algorithms, sensor failure detection and isolation techniques and integrity monitoring methods. The advantages and disadvantages of these traditional methods and architectures are summarised and compared. Based on the literature survey, a generalised MSDF model is presented as a frame for development of multisensor aircraft navigation system in this thesis.

Chapter 3 introduces the mathematical formulations of statistical estimation theory and hypothesis testing theory, which are required in this thesis to understand the development of multisensor data fusion algorithms for multisensor aircraft navigation systems. Estimation theory is a powerful mathematical tool that has been used in various engineering fields to accurately estimate the states of complex dynamic systems and to implement the most effective control of the systems. This chapter first introduces conventional Kalman filter algorithms, including linear and extended Kalman filters. The information form of the Kalman filter is then given in order to deduce various distributed data fusion filter algorithms.

Statistical testing theory is an auxiliary tool that is used to further confirm the validity of sensor data and the estimated system states. This chapter also introduces Bayesian detection and Newman-Pearson detection problems and the statistics of the Kalman estimate errors and residuals (innovations).

Chapter 4 first introduces various coordinate systems used in this thesis and evolution of the inertial technology, and examines the performance of different-grade inertial sensors. The major efforts of this chapter are to establish the normalised navigation equations of major navigation systems and to analyse their error dynamic models, including inertial systems and global navigation satellite systems (GNSS). These normalised navigation, attitude determination and error dynamic equations constitute the mathematical foundations to design, develop and simulate multisensor data fusion filters.

Chapter 5 analyses and evaluates redundant sensor system configurations and develops sensor-level data fusion methods. The main purpose of sensor-level data fusion is to provide highly reliable and accurate sensor data for subsequent system–level data fusion modules and an ability to detect sensor failures and reconfigure SRIMU systems and inertial state vectors in sensor network systems in the event of sensors failures. Two detection methods are developed to improve the generalised likelihood ratio test (GLRT) method for monitoring sensor failures of different modes. This chapter also presents SRIMU error compensation filters to enhance the performance of the GLRT-based methods. This chapter provides the basis for the design of fault-tolerant navigation systems with highly reliable integrity.

Chapter 6 addresses the problem of distributed sensor network systems and develops data fusion methods for distributed sensor network systems, including inertial measurement (data) algorithms, state fusion algorithms and inertial network integrity monitoring algorithms. For the first time, this chapter presents inertial network sensing models and develops dynamic relationships among the inertial network nodes. Two kinds of inertial sensor network architectures are identified in this chapter, each with two different communication modes. In the first kind of the distributed systems, all of the sensor systems directly or indirectly measure identical system states. In the second kind of distributed systems, different sensor subsystems observe their local states. However, all of the local system states are dynamically related though dynamic relationships. For these different distributed systems, four distributed data fusion filters are presented in this chapter.

Chapter 7 develops a simulation system environment to test and evaluate the FDI and integrity monitoring algorithms, and the data fusion algorithms developed during this study programme. For this purpose, this chapter describes the overall architecture of this software simulation system and the sub-architectures of the inertial simulation system and the GPS simulation system. The results of simulation studies are presented in this chapter.

Chapter 8 summarises the work of this thesis and provides final conclusions. Finally, areas of further work are recommended.

Chapter 2

MULTISENSOR DATA FUSION FOR AIRCRAFT NAVIGATION: OVERVIEW & METHODOLOGY

2.1 Introduction

This chapter reviews existing fault-tolerant navigation system architectures and data fusion methods for the development of multiple sensor navigation systems. In Section 2.2, conventional fault-tolerant architectures used for the design of aircraft navigation systems are outlined and briefly compared. The progression of various Kalman filter architectures and filtering algorithms employed in many integrated aircraft navigation systems are assessed and their advantages and disadvantages are summarised in Section 2.3. Section 2.4 examines the evolution of sensor failure detection and isolation (FDI) and integrity monitoring techniques, which are used in GNSS and inertial sensor systems. On the basis of the literature survey, Section 2.5 presents a generalised multisensor data fusion model (MSDF), which will be used for the development of future aircraft multisensor navigation systems.

2.2 Overview of Fault-Tolerant Navigation Systems

Fault-tolerant navigation systems have been in use for over 30 years. The design methods incorporate fault-tolerant strategies and data fusion techniques to enhance the reliability and safety and also to improve the performance of aircraft navigation systems. During this development, three forms of redundancy have been proposed: hardware redundancy, software redundancy and analytical redundancy. Figure 2.1 outlines the fault-tolerant design methods used in aircraft navigation. Hardware redundancy takes advantage of multiple navigation sensors/systems to achieve fault tolerance and to improve the performance of a navigation system. This approach is based on the fact that measurements from various sensor systems may be independent, redundant, complementary or cooperative. These different types of measurements can be fused by means of sensor data fusion algorithms so that the overall system performance is better than that each system can obtain independently. Hardware redundancy techniques have been widely applied to many avionics systems[21-23].

Software redundancy makes use of different software versions to increase the safety and reliability of navigation solutions by avoiding possible errors caused by software design and computing failures. However, software redundancy cannot increase the accuracy of navigation solutions.

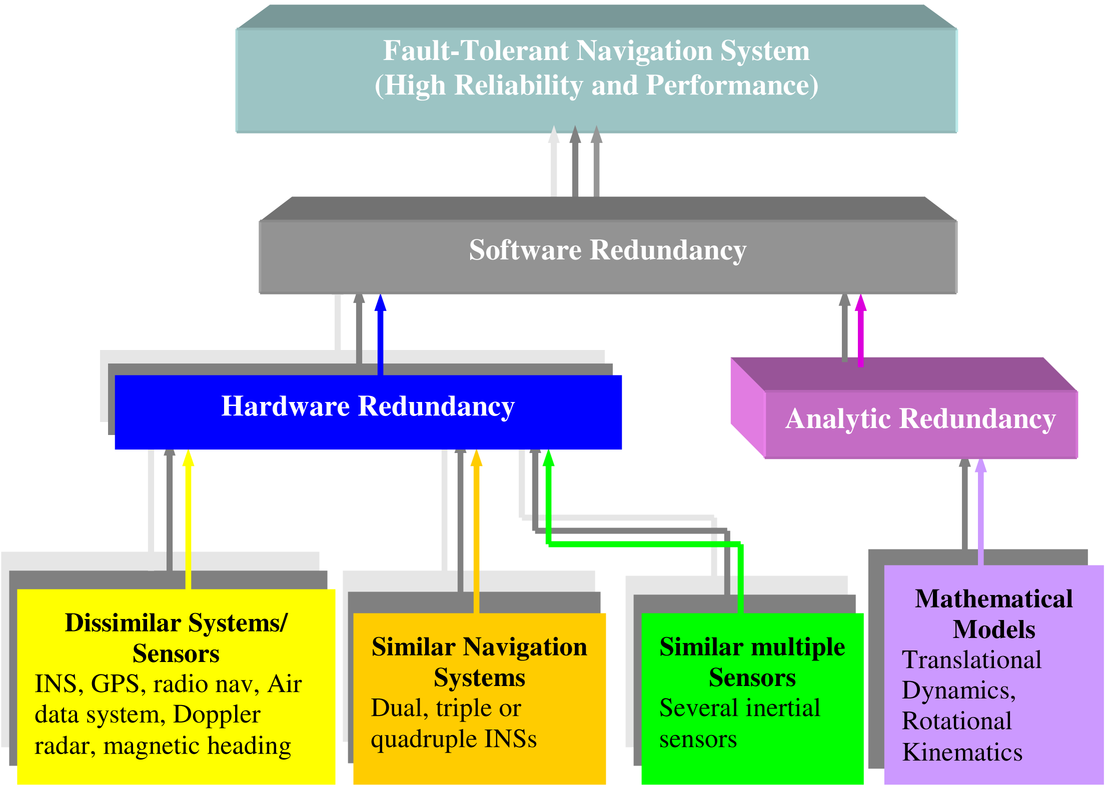

Fault-Tolerant Navigation System
(High Reliability and Performance)
Software Redundancy
Hardware Redundancy
Analytic Redundancy
Dissimilar Systems/ Sensors
INS, GPS, radio nav, Air data system, Doppler radar, magnetic heading
Similar Navigation Systems
Dual, triple or quadruple INSs
Similar multiple Sensors
Several inertial sensors
Mathematical Models
Translational Dynamics, Rotational Kinematics

Figure 2.1 Hierarchical Structure of Fault-Tolerant Design Methods

Analytical redundancy is based on the knowledge of rotational kinematics and translational dynamics of an aircraft to enhance hardware redundancy[24], and is usually used to generate additional redundant information for the diagnosis of sensor/ system failures rather than the improvement of accuracy of navigation systems[25]. Therefore, the analytical redundancy is considered as a failure detection method in many practical systems.

Hardware redundancy plays an essential role in the design of fault-tolerant navigation systems and the level of fault tolerance depends on both the architectures of hardware redundant systems and the data fusion methods implemented. Two types of hardware redundancy have been developed for the design of fault-tolerant aircraft navigation systems, system-level redundancy and sensor-level redundancy, which are described in the following subsections.

2.2.1 System-Level Redundancy

A system-level redundancy architecture is illustrated in Figure 2.2 where each INS in a triplex or quadruplex system must operate independently. It is also known as an independent system architecture because there is no data communication between these INSs. Each inertial system can also be integrated with other navaid systems to improve the navigation accuracy and to control the accumulation of inertial sensor errors with time. Fault-tolerant management checks the consistency of the outputs of all INSs to diagnose a failed inertial system, typically by using a majority-voting method or a weighted-mean method. In order to provide fail-operational/fail-safe operation, the fault-tolerant navigation system must have at least three INSs. In other words, nine pairs of inertial sensors (accelerometers and gyros) are needed where each INS is a conventional orthogonal configuration.

The main advantages of this architecture are that the design and integration are simple and that it does not need complex fault-tolerant techniques for diagnosis of system failures. However, if any one sensor in one INS fails, then this INS has to be removed from the fault-tolerant architecture. As a result, this architecture cannot exploit the benefits of redundant inertial sensors to dynamically reconfigure an aircraft navigation system in the event of one INS failure.

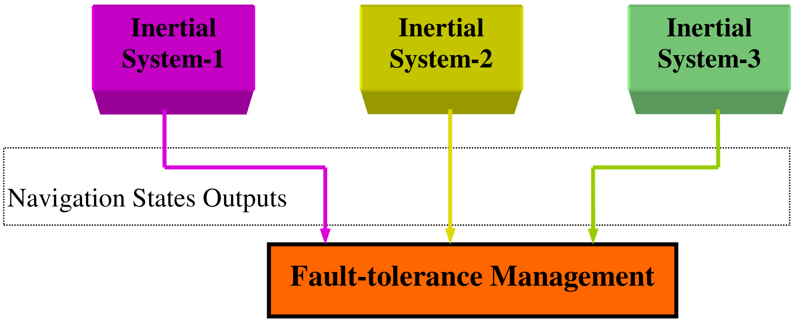


Figure 2.2 System-Level Redundancy Architecture

This traditional redundant architecture is still used in many current military and commercial avionic systems[26]. But, it is expensive and the duplication of INS modules results in a significant increase in mass.

2.2.2 Sensor-Level Redundancy

Sensor-level redundant architectures were developed with the advent of high-speed, large memory embedded microprocessors and low-cost, small-size and low-mass inertial measurement units (IMU). Several redundant schemes have been proposed[27~31], including IMU-level redundancy and multisensor redundancy.

An IMU-level redundant architecture is depicted in Figure 2.3 where duplex or triplex conventional IMUs are configured in a federated architecture to provide fault tolerance. Each IMU can be skewed with respect to the aircraft body axes when it is mounted in the aircraft to reduce the number of IMUs[27][29][31].

In principle, a fault-tolerant navigation system consisting of two IMUs affords the fail-operational/fail-operational/fail-safe operation if one of the IMUs is skewed relative to the aircraft body axes, or a non-orthogonal configuration. Then, six pairs of inertial sensors can achieve a higher level of fault tolerance in comparison with three independent INSs. Each navigation processor can combine the outputs of all IMUs with data from navaid systems to estimate the aircraft motion states, and to perform sensor failure detection and isolation, as well as navigation system reconfiguration. This IMU-level architecture significantly increases the level of fault tolerance and effectively makes use of existing IMU equipment. But, the resultant fault-tolerant system is still expensive. Considerable efforts are being made to reduce volume, weight and cost.

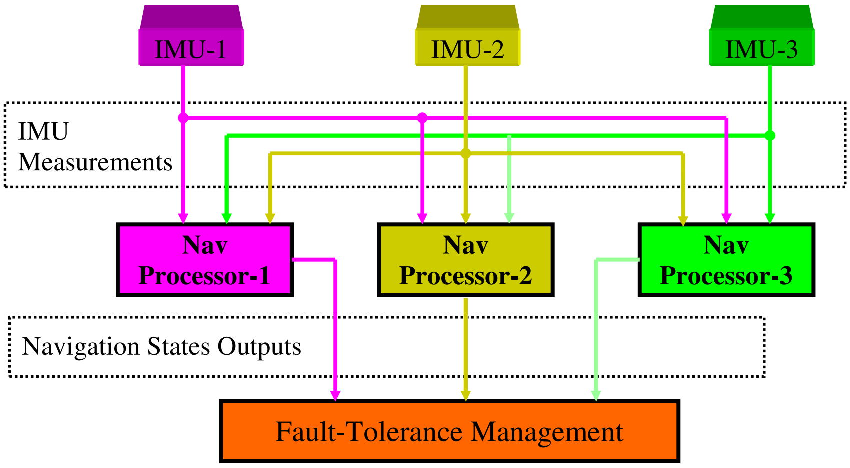


Figure 2.3 IMU-Level Redundancy Architecture

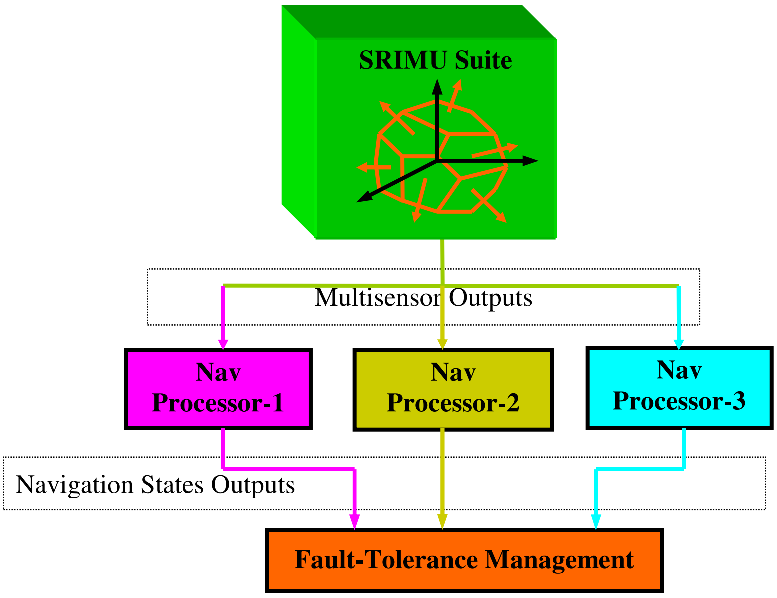


Figure 2.4 Multisensor Redundancy Architecture

A recent development is to integrate multiple inertial sensors in a single suite in the form of non-orthogonal configurations[28][30], known as skewed redundant IMU (SRIMU) configurations. One multisensor suite can thus replace multiple IMUs to reduce the volume, weight and power required for an aircraft navigation system. A representative architecture of a multisensor fault-tolerant system is depicted in Figure

2.4 where the multisensor suite consists of a dodecahedron configuration. Six inertial sensors are installed perpendicular to the parallel faces of a regular dodecahedron. Outputs from the multisensor suite are sent to several redundant processors, which individually perform navigation and attitude computations, sensor FDI functions and navigation system reconfiguration.

Multisensor redundancy is a cost-effective method that exploits the benefits of emerging inertial sensor technologies and high-speed embedded microprocessor systems. Multisensor technology provides the basis for the future generations of navigation systems.

2.2.3 Distributed Redundant Architecture

The distributed redundant architecture is a new fault-tolerant concept, which was developed with the introduction of distributed and integrated modular avionics architectures. For example, a current combat platform may have a total of twelve traditional IMUs of various quality providing the inertial state vector information required by avionics systems and weapon systems[33]. In this architecture, inertial sensor systems are mounted at several locations in an aircraft not only to meet the fault tolerance requirements of navigation systems but also to provide highly accurate local inertial data for other systems, for example, weapon system controls, radar stabilization and motion compensation. The concept of an inertial network used for aircraft avionics was initially proposed by Kelley, Carlson and Berning[32] in 1994 and then further developed by Berning, Howe and Jenkins[33] in 1996 and by Kaiser, Beck and Berning[34] in 1998.

However, the research published to date does not provide a systematic study of this inertial network architecture, specially in terms of data fusion methods, dynamic alignment and correction of distributed inertial sensor systems, and distributed sensor failure detection and isolation techniques. Therefore, there is a need for systematic investigation of data fusion methodologies in the design, development and simulation of fault-tolerant aircraft navigation system based on distributed inertial network architectures.

2.3 Data Fusion Filter Architectures

Kalman filtering techniques have been developed for applications in aircraft navigation, control and guidance since the 1970s. During this period, various Kalman filter architectures and filtering algorithms have been proposed as prime data fusion methods for fusing multiple navigation sensors/systems in order to achieve the required navigation performance. The data fusion filter architectures currently used in aircraft integrated navigation systems can be categorised as four types: centralised, cascaded, federated and distributed data fusion architectures.

2.3.1 Centralised Filter Architecture

The centralised filter architecture is shown in Figure 2.5. Measurements or data from all navigation sensors/systems are processed in a central data fusion filter to obtain the accurate estimates of the navigation states. It is the most common filter design implemented in current integrated navigation systems, for example, INS/GPS/ Doppler integrated systems[36], Doppler/GPS integrated systems[37] and almost all tightly-coupled GPS/inertial systems[35][38][39] where raw GPS measurements and INS outputs are combined in a centralised filter to estimate the navigation state errors and sensor errors, including the GPS receiver clock errors, inertial sensor errors and baro-altimeter errors.

Numerous covariance analysis methods and numerical computations of the standard and extended Kalman filters have been reported[40][41]. Theoretically, the centralised filter can obtain optimal estimates of the aircraft motion states. However, with the increasing numbers of sensor systems in aircraft, the filtering algorithms can be quite complex and the centralised filter computation can be time-consuming as a result of the large state dimension in the dynamic models of the filter. Accordingly, the centralised filter may not necessarily be a proper approach to the development of fault-tolerant multisensor navigation systems[49][62][67]. To overcome the weaknesses of the centralised filter, other filter architectures have been proposed in the recent years.

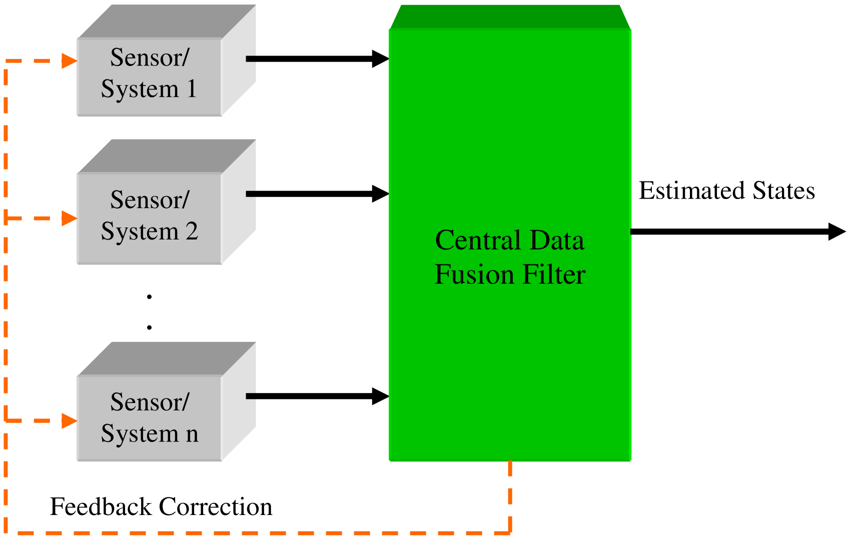


Figure 2.5 Centralised Data Fusion Architecture

2.3.2 Cascaded Filter Architecture

The cascaded filter architecture is depicted in Figure 2.6 where the outputs of one filter are used as inputs to a subsequent filter. The filter outputs include the estimates of the system states and their error covariances. This filter architecture has been especially proposed for integration of existing navigation systems that contain their own Kalman filters.

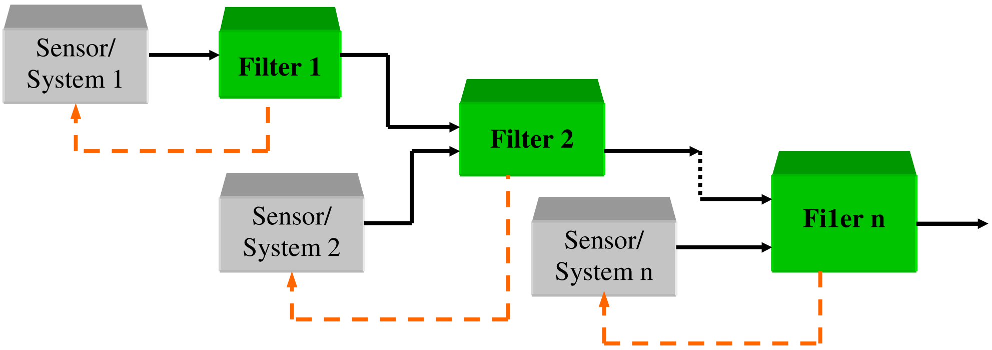


Figure 2.6 Cascaded Data Fusion Architecture

The cascaded filter can improve the accuracy of integrated navigation systems and perform in-flight calibration or transfer alignment between an INS/GNSS integrated system and an INS or attitude heading reference system (AHRS). This architecture has been used in several GPS/INS/terrain aided navigation systems[42-45] and loosely coupled GPS/INS integrated navigation systems where the GPS-based navigation solutions derived by an GPS internal filter and INS data are combined in a separate cascaded filter external to the GPS receiver to estimate the navigation state errors and the inertial sensor errors. The GPS filter estimates the GPS receiver clock errors. However, the GPS filter is usually based on a simplified model and may not output computed error covariances. Consequently, the cascaded filter may not have access to covariance information.

Schlee et al[42] develop a cascaded filtering algorithm to improve the accuracy of an existing GPS/inertial system, known as a master INS, which utilised an internal GPS filter to estimate the master INS navigation solutions and the GPS clock errors. This cascaded algorithm also provides transfer alignment between the master INS and a second inertial system. This study has shown that improvement in the accuracy of the master INS and the obtainable accuracy of the transfer alignment largely depend on the update rate of the cascaded filter. However, correlations of the state errors caused by the internal GPS filter are ignored in the measurement noise matrix of the cascaded filter. From Kalman filter theory, the non-diagonal elements of the state error covariance matrix of the GPS filter (which represent the correlations) can only be ignored if the filter contains highly accurate estimates of the navigation states and the values of non-diagonal elements are far less than the main diagonal elements. Otherwise, the performance of the cascaded filter may be degraded as a result of ignoring the correlation.

Wade and Grewal[43] analyse the effect of this correlation on the accuracy of cascaded GPS/INS systems and their results show that the accuracy of the cascaded systems depends on the correlation matrix in many cases. When the state errors estimated by the internal GPS filter are closely correlated, the cascaded filter may incorrectly estimate the navigation state errors and the inertial sensor errors. Wade and Grewal further suggested adjusting the measurement noise matrix by using adaptive process noise in the cascaded filter. However, development of this adaptive process and identification of the measurement noise matrix are not reported.

In order to improve the robustness of the cascaded filter to input conditions and adverse environments, Karatsinides[44] proposes two methods for dealing with the

GPS position biases and identifying the statistical values of the measurement noise for the cascaded filter. The GPS positioning solution contains bias resulting from satellite clock errors, ephemeris errors, ranging signal propagation delay and the geometry of satellites. Although this GPS position bias is unobservable and cannot be estimated in the GPS filter, it can influence the accuracy of cascaded GPS/INS systems through the error covariance matrix. The first method models the GPS position biases as a first-order Gauss-Markov process and then uses these biases as the consider-states of a Schmidt-Kalman filter. But, the part of the Schmidt-Kalman gain matrix related to the consider-states is set to zero in order to ignore the estimated consider-states. The second method computes the variances and covariances of the errors of the navigation states derived by the GPS filter using conventional computation equations of variance and covariances provided that the update rate of the cascaded filter is less than the GPS filter.

The cascaded filter architecture is readily implemented by means of existing navigation systems and needs minimisation of required modifications for customised applications. In practice, most existing navigation systems do not output covariance data of the navigation state errors. Consequently, the cascaded filter is extremely dependent upon the methods that are used to estimate covariances of the primary filter and the performance of the primary filter. Moreover, tuning of the primary filter is of critical importance to the performance of the cascaded filter[43].

2.3.3 Federated Filter Architecture

The federated filter architecture was initially recommended by Carlson[46] for integrating multiple navigation sensor systems in order to provide a high level of fault tolerance and accuracy. This is actually a two-stage filtering architecture, as shown in Figure 2.7 where all parallel local filters combine their own sensor systems with a common reference system, usually an inertial navigation system, to obtain the local estimates of the system states. These local estimates are subsequently fused in a master filter to achieve the global estimations. By using a common reference system, all parallel filters have a common state vector. The federated filter is generally designed on the basis of two different strategies[46][47]. In the first method, the local filters are designed independent of the global performance of the federated filter and estimate n sets of local state vectors and their associated covariances by using their own local measurements. These n sets of the local state estimates are then weighted by their error covariances to obtain the global state estimates. The second method is based on the global optimality of the federated filter and the local filters are derived from the global model of the federated filter and estimate n versions of the global states from local sensor measurements. These n versions of estimates are weighted by their error covariances to obtain the global optimality. The master filter is a weighted least-squares estimator. Furthermore, Carlson[48] developed a square-root form of the federated filtering algorithm to increase the computational precision and the numerical stability of the federated filter.

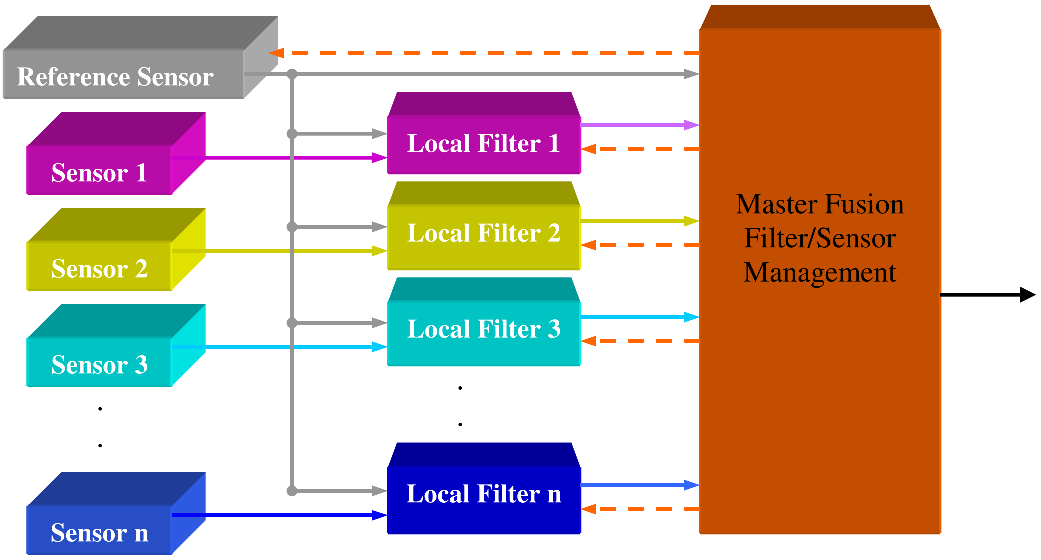


Figure 2.7 Federated Data Fusion Architecture

A significant feature of the federated filtering process is that a reference INS must be used to create the common system states in the local and master filters, which are the aircraft navigation states. Therefore, each local filter can obtain the suboptimal global navigation states. A comparison of the federated and centralised filters[49] has shown that the federated architecture offers improvements in failure detection, isolation and recovery (FDIR) and the fault tolerance over the centralised filter.

Levy[50] uses dual state suboptimal analysis to model the true world state vector and develops covariance analysis algorithms for assessing the suboptimality of both the cascaded and the federated filters. The dual state contains the states of the first and second filters in the case of the cascaded filter (or the states of all parallel filters and the master filter in the case of the federated filter). Levy's results have shown that the cascaded and federated filters are seldom optimal in comparison with the centralised Kalman filter. As the master filter updates become sparser, the actual performance of the federated filter degrades in comparison with the centralised filter. The federated filter is only optimal (or equivalent to the centralised filter) when the full global state is modelled in each local filter and the master filter is run at the update rate of the local filters.

Tupysev[51] develops a federated filtering algorithm based on the principles of state vector augmentation and the rejection of partial information. Unlike Carlson's filter, the global state model that is used to derive the parallel local filters contains a common state vector plus individual local bias state vectors instead of all the states of the local filters. The local state is a subset of the global states.

However, the use of a reference navigation system as a common information source of all local filters in the federated filter architecture means that common mode failures in the reference system can corrupt the performance of these filters. This influence can further degrade the level of fault tolerance and FDIR functions. This problem seems to have been ignored in current designs of federated integrated navigation systems. In addition, this federated filter architecture and corresponding filtering algorithms are not applicable to integration of distributed inertial sensor systems with navaid systems because there are no common system states in the distributed inertial sensor systems.

The federated filter has been applied to several multisensor navigation systems, for example, GPS/INS/SAR/terrain aided navigation and tracking systems[52-54] and is sometimes referred to as the decentralised filter[55][56]. To avoid confusion, the term *decentralised* is used as a synonym of *distributed* in this thesis.

2.3.4 Distributed Filter Architecture

The distributed filter architecture was originally developed for target tracking and identification where distributed sensor systems (possibly in different platforms) are combined in order to estimate and identify various moving targets in military applications[57]. Liggins et al[58] give a comprehensive survey of the distributed fusion architectures for target tracking. Distributed filtering techniques used for the design and development of fault-tolerant navigation systems have appeared since the 1990s[55]. Different from the filter architectures described above, distributed filter architectures have no standard model. From the perspective of use of information, there are two main data fusion approaches to the design of distributed filters, known as measurement fusion and state fusion. In state fusion, the local states estimated by the local filters are fused in a central filter to obtain global estimations. By contrast, in measurement fusion, various subsets of all the sensor measurements are fused by means of a bank of Kalman filters to obtain multiple state estimation versions of the global system states, which are compared or weighted to obtain the more accurate global state estimation and to detect sensor or system failures. However, there may be no central data fusion in a fully distributed multisensor data fusion system. In fact, the distributed filter architecture offers the most flexible scheme in the design of multisensor navigation systems.

Kerr[55] proposes a decentralised filtering structure which uses a voter/monitor method to check outputs of all local filters for failure detection, but the distributed filter algorithms developed for this structure are not explained in detail. In terms of the filter architecture, Kerr’s version is similar to the federated filter architecture given by Carlson[46]. The differences between them are the individual methods used for detection and isolation of subsystem failures. For example, Carlson’s filter uses filter residuals to detect sensor and subsystem failures whereas Kerr’s filter uses the voter/monitoring methods based on Gaussian confidence regions of the estimated states. However, some filtering algorithms, for example, Speyer’s parallel filtering algorithms[62] or others, may be used for this decentralised structure. Strictly, Kerr's structure is not a distributed filter architecture and it lacks systematic study on the corresponding filtering algorithms.

Brumback and Srinath[59] describe a distributed filtering mechanism. This is a hierarchical filtering architecture where the local filters fuse different subsets of all measurements for local state estimates and failure detection and isolation. A master filter combines the outputs of failure-free local filters to yield the global estimation. The local filters in the distributed filter architecture can have system models, which are different from the global model. The cascaded and federated filter algorithms are special cases of the distributed filter architectures.

Several distributed filtering algorithms have been developed since the 1980s for the design of various distributed control systems, target tracking systems and navigation systems[60-69]. Speyer[60] designed a distributed filtering algorithm in which each of K local filters has its own local sensor measurements and the same state model. Each local filter computes the global estimate of the system state vector. The information shared between these local filters consists of the local estimates, the local error covariances, and an additional (locally computed) data-dependent term, which is a dynamic compensation to account for the correlation between the local estimates. Speyer's filter is a fully distributed filtering architecture and has a high level of fault tolerance. However, by using the same state model, this filtering algorithm cannot be used in a distributed inertial sensor system where the local state vector is needed for specialised purposes, for example, local motion compensation.

Willsky et al[61] consider a problem where two local filters have state models which are different from the global model. Each local filter processes its local measurements and a fusion algorithm (based on the global model) computes a dynamic correlation correction term, combining the local estimates to obtain the global estimate. A necessary and sufficient condition for recovering the global state from the local states is that a relationship must exist between the observation matrix of the global state model and that of the local state models. This relationship is explained as a static matrix transformation. In other words, the local state vector is a subset of components of the global state vector. This algorithm has been extended to the design of a multisensor navigation system[59]. However, these algorithms imply that the local and the global states are represented in the same coordinate system and this is not necessarily true for distributed inertial sensor systems.

Hashemipour et al[62] introduce decentralised Kalman filtering algorithms for three types of multisensor networks: sensors collected, time sequential measurements and a hybridisation of these two types. In Hashemipour's filter, each local filter has the same state model as the central filter and the observation matrix of each local model corresponds to one sub-matrix of the observation matrix of the global model. Each local filter computes the global estimation and the local error covariance that are subsequently fused in a central filter to obtain the global optimal estimation. Accordingly, this filtering algorithm is similar to Speyer's filter. But it uses the information form of the Kalman filter and does not need feedback from the central filter to the local filters. Although this algorithm can be used to solve target-tracking problems, it cannot be used for distributed multisensor navigation systems because the local subsystem states are different in a distributed inertial sensor system.

Hong[63][64] introduces a distributed multisensor integration algorithm in which the local measurements, together with the previous global estimate obtained via the communication network, are locally processed to obtain the local state estimate and the local error covariance. These local estimates (state and covariance) are fused in a central filter to obtain the global estimate. Because the local state and covariance predictions are derived from the previous global estimates, the local filters have no the state models. However, the rotation matrixes and the translation transformations are introduced to establish the relationships between the local states and the global (central) state. Moreover, this algorithm was designed to minimise the uncertainties of these transformations. It should be noted that the same relationships are also used for measurement transformations from the local nodes to the central node. This is not necessarily true in distributed inertial sensor systems, especially when a nonlinear relationship exists between the measurements and the states. In comparison with Speyer's filtering algorithm, this method simplifies the complexity of the distributed filtering algorithms. However, the local states greatly depend on the global states because this method lacks local dynamic models.

Roy et al[65] proposes a square root filtering structure where parallel local filters have a smaller dimension than the global filter. Paik et al[66] develops a gain fusion algorithm for decentralised parallel Kalman filters to obtain computation-efficient suboptimal estimation results. Raol et al[67] describe a decentralised square-root information filtering scheme where all information fusion is processed locally at each node and there is no central fusion. These algorithms improve the computational precision and numerical stability of the existing distributed filtering algorithms.

A fully distributed filtering architecture and information fusion algorithm are developed, where no central data fusion centre is needed[68][69]. Each local filter has its own local system model and processes the local measurements and information assimilated from other filters to obtain a global estimate of the system state. However, there is still a key problem to be considered; the dynamic relation between the local states must be determined, especially if the local state models are different. Berg et al[70] describe the static relation between the local states and the global state by an approach similar to Speyer's method[60].

Multisensor data fusion for aircraft navigation aims at the improvements of the performance in terms of the three aspects:

- Aircraft navigation system RNP parameters;
- Fault tolerance of navigation system; and
- Estimation of local motion states.

The majority of previous developments have generally focused on the first two aspects. In other words, existing distributed filtering algorithms have preserved the global optimality of the navigation states, which is a desirable feature and serves as a benchmark for other avionic systems. However, these methods rarely consider the dynamics of the local subsystems and the dynamic relationships between the local subsystems. Some algorithms still require extensive computations of local and global inverse covariances. Very few studies have addressed estimation of the local states. In fact, distributed inertial sensor systems consisting of several IMUs mounted in an aircraft affords both redundant inertial measurement information and distributed inertial state vectors, which can be used both for aircraft navigation, guidance and control, and also for the implementation of local motion compensation functions. These IMUs measure local motion with reference to specific coordinate frames defined by their installation positions, and have individual error dynamics. Therefore, the local states must be accurately estimated to determine the local dynamic motion. The development of distributed filtering algorithms can also be used to investigate methods for dynamic alignment and calibration of distributed IMUs. Problems related to these considerations have not been addressed in the open literature and this thesis addresses the solutions of these problems by developing innovative distributed data fusion filters algorithms.

2.4 Multisensor Navigation System Integrity

Multisensor aircraft navigation systems can be subject to unforeseen changes resulting from sensor failures, the uncertainty of system models and variations in the operating conditions, which can lead to the degradation of the overall navigation performance. Such changes are known as failures even if they may not represent actual failures of physical sensors or components. In order to ensure the reliability of an aircraft navigation system, the data fusion mechanism has to detect and isolate sensor or system failures from the navigation system and also monitor the integrity of the navigation states derived by the fusion filter. These two important procedures are usually known as sensor/system failure detection and isolation (FDI) and navigation solution integrity monitoring (NSIM). Both functions must check the consistency and availability of data. The FDI procedure assesses data from sensor systems and issues a confidence range of the sensor data. The NSIM procedure confirms the integrity of the navigation solutions and provides alarms and system status information to flight crew.

A typical FDI or NISM algorithm has in general two objectives:

- To detect the failures,
- To isolate the failed sensors or components.

In some cases, an additional objective may be included to estimate the failure signals. FDI and NISM procedures rely on redundant data provided by hardware and software and analytical redundancy to fulfil the above objectives. A representative FDI or NSIM procedure usually consists of three steps, as shown in Figure 2.8. The first step, the *Residual Generator*, processes redundant data to generate a decision function (referred to as test statistic), which is a function of the data residual and a measure of the inconsistency of redundant data. Ideally, the decision function is independent of the true navigation states or measured states. To decrease the influence of noise on the decision function, a pre-processing filter may be used to enlarge the signal-to-noise ratio of the failure signals so that failure signals can be more easily detected and identified. The second step, the *Statistical Test*, establishes a decision threshold on the basis of certain criteria that are a measure of both the performance of the FDI/NISM algorithms and the accuracy of sensor measurements or the navigation states. The third step is a *decision-making* procedure that compares the test statistic with the decision threshold to verify if a sensor or component failure has occurred or if there are abnormalities in the navigation states or sensor data. Depending on the form of the decision functions, the statistical testing procedure can be performed by using Gaussian, Rayleigh, χ^2 - or t-distribution statistical tests.

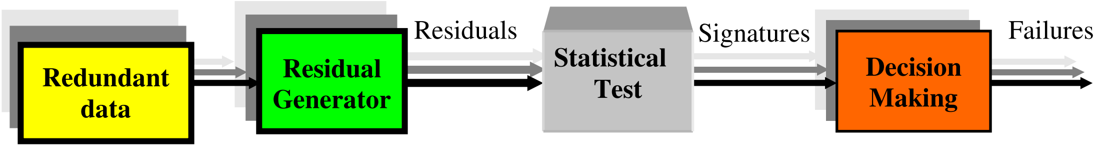


Figure 2.8 A Typical FDI/NSIM Procedure

There are numerous approaches to the generation of residuals. But, the three commonly used methods are parity matrix transformation, least-squares residuals and data fusion filter residuals (or innovation).

The FDI and NSIM performance is generally characterised by the probabilities of two decision errors: false alarms and missed alarms, which are the functions of the decision thresholds and are also related to the requirement of the navigation system accuracy.

All FDI and NSIM techniques can be categorised as either snapshot techniques or sequential techniques. The snapshot techniques use a single sample to detect and/ or isolate instantaneous sensor failures, typically for relatively large magnitudes of failures. The sequential techniques employ the cumulative information provided by the complete data history to detect drift failures and other soft failures, normally for smaller magnitude failure signals. The snapshot technique has the advantage that it does not rely on any assumptions on how a data fusion filter attains its current state whereas the sequential technique can improve the FDI reliability because it uses history data of the system.

2.5 Snapshot FDI Techniques

Many snapshot FDI algorithms have been developed over the past decades and the majority of these methods are based on the parity vector/space techniques. Evans and Wilcox[71] introduce a test matrix method and Gilmore and Mckern[72] present a parity equations comparison method. Both methods are used to detect and isolate failures of redundant inertial sensor systems in a dodecahedron configuration. The parity equations comparison tests a set of 15 parity equations each containing output values of four sensors while the test matrix method uses a 15x6-dimensional matrix with each row consisting of coefficients of one parity equation and performing a test on a different subset of four sensors. By minimising the maximum measurement residual magnitude, Potter and Deckert[73] develop a so-called minimax FDI algorithm for non-orthogonal redundant inertial systems in which any set of four sensors are tested and compared. Generally, these methods all need to compute and compare each parity equation and employ a least-squares estimator to estimate the measured states. Obviously, it becomes a time-expensive procedure for a large number of redundant sensors.

Wilcox[74] gives comparisons of eight earlier FDI algorithms for strapdown redundant inertial systems in a dodecahedron configuration. The differences between these algorithms are in the approaches used to generate test signals. One method uses a Kalman-Bucy filter for failure correction while all the other algorithms compute the measured states by means of weighted least-squares estimators. However, these algorithms did not link the integrity of the navigation system to the performance of the FDI algorithms.

To overcome the shortcomings of earlier FDI methods, generalised likelihood ratio test (GLRT) methods[75][76] have been introduced for the detection and isolation of failures in redundant inertial sensor systems. The GLRT methods are based on the maximum likelihood estimate of residual magnitudes where the resultant test statistic is a function of the parity vector. One of the advantages of GLRT algorithms is that the performance of the FDI algorithms is related to the integrity performance of the navigation system and the failed signals can theoretically be estimated. However, the GLRT algorithms are not able to detect two simultaneous sensor failures.

Hall et al[77] designed a 12-state Kalman filter to compensate the parity vector by eliminating the effects of normal sensor errors on test signals. This compensation technique enhances the effect of true failure signals on the resultant parity vector and improves the GLRT performance. However, after the parity space transformation, the filter state represents a combination of the sensor errors rather than physical sensor errors. In other words, the state estimates no longer correspond to physical sensor errors and consequently, the practical sensor errors cannot be dynamically corrected.

Sturza[78][79] describes the parity vector approach to the detection of jump failures of skewed redundant inertial systems and GPS signals, as well as statistical methods for the determination of detection thresholds of RAIM and FDI algorithms. Brown and Sturza[80] further analysed the effect of geometry of the GPS satellites on the parity vector-based RAIM.

Sturza and Brown[81] give two RAIM algorithms CFAR (constant false alarm rate) and CPOD (constant probability of detection) for GPS integrity monitoring. In the CFAR algorithm, the detection threshold is based on a constant false alarm rate. In the CPOD algorithm, the detection threshold varies in order to provide a constant missed alarm rate. Clearly, these methods are not suited to SRIMU configurations because the measured states and the navigation states are the same for GPS RAIM. For SRIMU FDI, the navigation states are derived from the measured states by solving a set of differential equations.

The mathematical background of the RAIM methods is given by van Diggelen and Brown[82]. A number of GPS signal failure detection algorithms, known as receiver autonomous integrity monitoring (RAIM) methods, are based on the parity vector technique[83].

These proposed parity vector-based FDI or RAIM algorithms can effectively detect jump failures of navigation sensor systems or GPS signals. However, they are not able to detect any soft failures arising from drifts or two or more simultaneous sensor failures.

2.5.1 Sequential FDI Techniques

Several sequential FDI algorithms have also been developed to detect both jump and time-drift failures in dynamic systems. Sequential FDI techniques can be classified as two types. One type directly uses the history data of the sensor outputs. Wald originally introduced a sequential probability ratio test (SPRT) algorithm to make a binary decision of one mode (degradation) against an alternative mode (normal). Chien and Adams[84] present an improved SPRT algorithm whose design criterion is based on the minimization of the mean detection time to detect system failures subject to constraints on the false alarm and missed alarm probabilities. The time minimization is implemented using positive feedback control of the likelihood function. It is also used for the detection of jump mode failures.

The other type implicitly employs the history data of system outputs. This type of sequential FDI method is usually based on analytical redundancy techniques that require the development of dynamic models of detected systems and this FDI method is known as a model-based FDI method. Willsky and Jones[85] describe an improved GLRT method for detecting abrupt changes in linear dynamic systems by using the sequential system outputs. Willsky[86], Gertler[24] and Patton[25] summarise various FDI methods used in dynamic systems. The majority of these model-based FDI methods apply various GLRTs to test the Kalman residuals for the presence of sensor failures or abrupt change of system states. They may also detect system degradation but cannot detect time-drift sensor failures. From a survey of the literature, many existing sequential or model-based FDI algorithms are often used to detect jump failures.

Kerr[87] proposes a method known as the two-confidence region comparison approach to failure detection. One confidence region is determined by the Kalman predictor and is centred about the expected unfailed state and reflects the uncertainty of the system noise. The other confidence region is created by the Kalman estimator and is centred about the Kalman estimate. This region reflects the uncertainties of the system states and the measurement noise. A failure is detected by comparing these two confidence regions. As Kerr states in his paper, *as long as the two confidence intervals overlap, the true state may be in both confidence intervals; however, when both confidence intervals are disjoint, the true state cannot be in both intervals simultaneously and a failure is declared.* However, the increasing number of sensors makes the computation of failure thresholds very complex. In addition, because an identical system dynamic model is used in the state predictor and the state estimator, uncertainties of the filter dynamic model may cause false alarms. In the design of an INS/GNSS integrated filter, the filter dynamic model can be derived by disturbing the inertial navigation equation about the nominal navigation states. In this case, the sensor drift failures can contribute to errors in the nominal navigation states. This disturbance error further affects both the predictor and estimator. This effect may lead to missed alarms. Accordingly, this method does not apply to error dynamic models where the errors of the navigation states are used as the filter state rather than the navigation states.

NSIM methods are based on sequential FDI techniques, which analyse the covariance matrix and residuals (or innovations) of the data fusion filter. The most direct method is to compare different versions of the navigation states estimated by a bank of Kalman filters. For example, Brenner[88] proposes a solution separation method for GPS/INS integrated system. In this method, a bank of Kalman filters is used to obtain both full-set solutions and sub-set solutions. The test statistic and the decision thresholds are determined on the basis of the horizontal separations between the full-set solution and sub-set solutions, and the Kalman filter covariance matrices.

Diesel et al[89] give an autonomous integrity monitored extrapolation (AIME) algorithm used in the Litton GPS/IRS integrated system. AIME is an open control system using range differential measurements as the filter observables, which are the differences between the observed GPS pseudoranges and the computed ranges based on the predicted navigation states and the satellite positions. The difference between the time-updated (predicted) state and the measurement-updated (estimated) state of a Kalman filter are used as the test statistic to monitor the navigation states. Hanlon and Maybeck[90] analyse the effects of mismodelled input matrix, output matrix and state transition matrix on the residuals resulting from a bank of Kalman filters, each using a different filter model to describe the same dynamic system. They also develop a hypothesis testing algorithm using these residuals to detect failure status of a flight control system and to estimate the true system model. Although this method has advantages in the design of a reliable flight control system, it cannot be used in distributed inertial sensor systems. However, this method may be suitable for the federated filter architecture or traditional multisensor-based navigation systems where the main purpose is to estimate the centralised navigation states.

All of the published sensor FDI algorithms that have been located are capable of detecting hard sensor failures in clustered inertial sensor system architectures. Although some of these FDI algorithms can enhance the performance of sensor FDI, they cannot improve the accuracy of an SRIMU system. Earlier NSIM methods were developed for special GPS/INS integrated navigation systems with a centralised filtering architecture. However, they are not amenable to expansion and cannot be used in distributed inertial network systems. In this thesis, several improved FDI and NSIM methods are presented to detect the drift sensor failures and the navigation state abnormalities in distributed inertial network systems. The compensation filters are developed for the correction of SRIMU measurements.

2.6 Multisensor Fusion Model for Navigation Systems

Multisensor data fusion covers fault-tolerant design and data fusion methods. As identified in Chapter 1, the JPL MSDF model and other models do not apply to the development of distributed multisensor navigation systems. From the definition of multisensor data fusion given in Chapter 1, a multisensor data fusion model for aircraft navigation systems is a conceptualised framework in which sensor network topology architecture, data communication mechanism, system functions and related operational modes are defined. The data fusion methodologies are then developed to implement the required system functions and operational modes. This thesis presents a generalised MSDF model for the design, analysis, development and simulation of multisensor aircraft navigation systems, as illustrated in Figure 2.9.

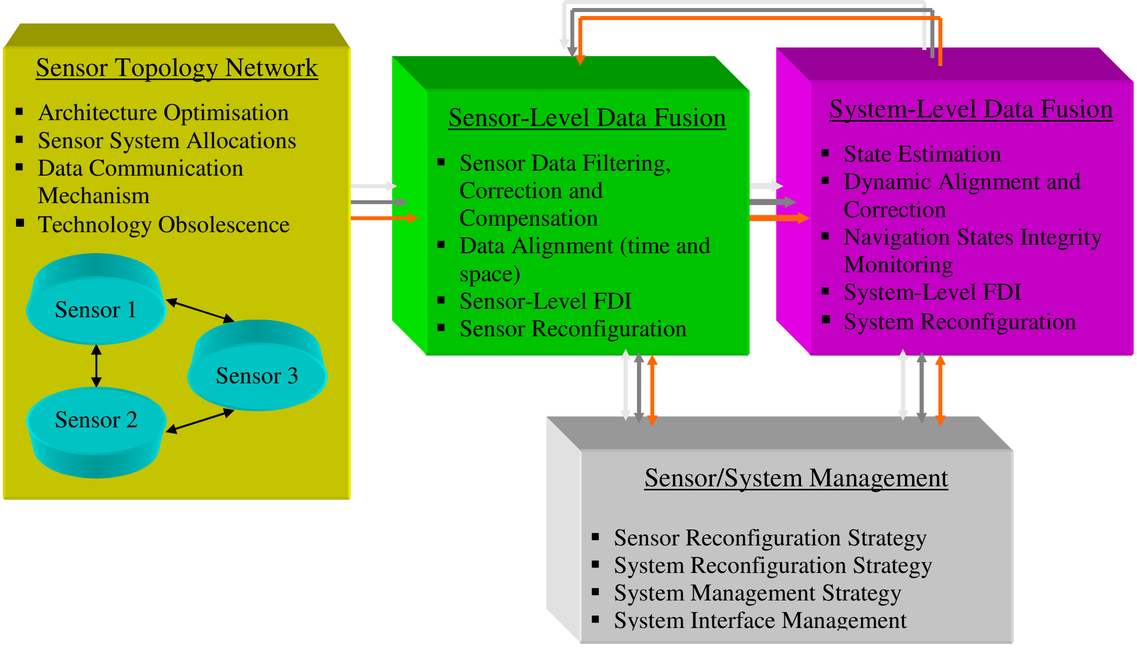

Sensor Topology Network
Architecture Optimisation
Sensor System Allocations
Data Communication Mechanism
Technology Obsolescence
Sensor 1
Sensor 2
Sensor 3
Sensor-Level Data Fusion
Sensor Data Filtering, Correction and Compensation
Data Alignment (time and space)
Sensor-Level FDI
Sensor Reconfiguration
System-Level Data Fusion
State Estimation
Dynamic Alignment and Correction
Navigation States Integrity Monitoring
System-Level FDI
System Reconfiguration
Sensor/System Management
Sensor Reconfiguration Strategy
System Reconfiguration Strategy
System Management Strategy
System Interface Management

Figure 2.9 Generalised MSDF Model for Aircraft Navigation Systems

2.6.1 Sensor Topology Network

The sensor topology network provides a hardware foundation for the design and development of multisensor navigation systems and describes distributions and allocations of various sensor systems in the network. The architecture of a sensor topology network is specified according to the system design requirements. A sensor network topology can be a serial, parallel or hybridised architecture; or a completely packaged, distributed network or combination of both. Parallel and distributed sensor network architectures are the most commonly used sensor topologies in modern aircraft.

Optimisation of the topological architectures of a sensor network determines the optimal sensor system configurations and allocations in an aircraft navigation system. The allocations of sensor systems depend on the requirements of both the aircraft navigation system (e.g. survivability and fault tolerance) and other avionics systems for the inertial and navigation states. For example, many avionics systems require highly reliable, continuous inertial data to implement individual functions, as stated in Section 1.1. Some inertial systems must be located close to specific avionics systems to provide the precise local motion states for stabilisation of specific avionics systems, such as weapon pointing systems and imaging radars.

The data communication specifies the architecture of a communication network and the requirements for data buses in order to exchange data among individual sensor systems and to transmit data to other avionics systems. The data protocol and transfer speed must be selected so that the data communication network can meet the requirements that data fusion algorithms require from sensor data.

The evaluation of technology obsolescence is a key to the mitigation of ageing technologies and to the application of emerging technologies to meet the long-term operational lifetime requirements for aircraft navigation systems.

Data fusion methodologies can then be developed so that the resultant data fusion algorithms, in combination with a data communication network, can fuse various sensor data to achieve the required performance for aircraft navigation and other airborne applications.

2.6.2 Sensor-Level Data Fusion

Sensor-level data fusion is preliminary data fusion. It analyses and qualifies all sensor measurements to provide highly reliable sensor data for subsequent system-level data fusion. It can also transmit health status information of all sensor systems to the sensor management. At this level, the following functions are performed:

- Sensor corrections and compensations to obtain accurate sensor data;
- Data alignment in time and space to ensure that associated measurements of all sensor systems are time-synchronised and common-coordinated;
- Detection of sensor failures and isolation of failed sensors if necessary;
- Reconfiguration of sensor systems based on certain sensor reconfiguration strategies.

Sensor failure detection and isolation (FDI) is the core of this functional module.

2.6.3 System-Level Data Fusion

System-level data fusion is the kernel of a multisensor data fusion system. It fuses data from sensors and subsystems in terms of optimised data fusion algorithms to estimate the required system states and to monitor the integrity of the estimated states by performing specific error covariance analysis and statistical tests. At this level, the following functions are undertaken:

- State estimation. This function covers the design and development of both data fusion filter architectures on the basis of the topological architecture of sensor network and optimal data fusion algorithms suitable for the filter architectures;
- Navigation solution integrity monitoring and system FDI. These functions are required in order to obtain the integrity of the navigation system. They are concerned with analysis and evaluation of the state error covariance and residual information of the data fusion filter;
- Alignment and correction of inertial systems in distributed sensor network. This function is concerned with development of data fusion algorithms to dynamically align and correct distributed inertial systems.
- Reconfiguration of system models. This function implements fault-tolerant design in a multisensor navigation system. It is provided to fulfil system reconfiguration strategies and operational modes.

2.6.4 Sensor/System Management

Sensor/system management performs three types of management functions: sensor network system management, data communication management and human-machine interface management. According to the health status information from the sensor-level data fusion and system-level data fusion modules, and command inputs from the pilot, the sensor network system management determines the operational modes and reconfiguration strategies of the navigation system, and transmits the associated commands to the two data fusion modules. The sensor-level data fusion module and the system-level data fusion module then separately reconfigure the sensor systems and navigation system to meet the required navigation performance and fault tolerance of the navigation system. Sensor/system management strategies are specific to the architecture of sensor network and fault tolerance requirements.

Data communication management manages the data exchange among the nodes of the sensor network system according to the sensor/system reconfiguration strategies and external commands. Communication management strategies allow sensor systems to be added or failed sensor systems to be removed from the sensor network architecture without affecting the data communication architecture and operation of the complete system. Human-machine interface management provides a user-friendly interface for flight crew.

In operation, the sensor/system management dynamically allocates tasks to the functionary sensor systems and software components to execute the required system functions.

Investigations to be performed in this thesis will follow this generalised MSDF model.

2.7 Summary

This chapter has reviewed developments of fault-tolerant aircraft navigation systems and data fusion methods based on a wide range of literature survey. The main issues covered in this chapter are as follows:

1. Review of three fault-tolerant navigation system architectures, which have been employed in aircraft navigation systems.
2. Analysis and comparison of four forms of data fusion filter architectures, which are currently used in integrated aircraft navigation systems.
3. Description of FDI and NSIM techniques applied to inertial sensor systems and GNSS.
4. Development of a generalised multisensor data fusion model, which will be used to design and develop future aircraft multisensor navigation systems.
5. Identification of several main problems existing in the design of current

multisensor fusion navigation systems, including detection of time-drift sensor/system failures, SRIMU error compensation, and multisensor data fusion methods and distributed state vector integrity monitoring strategies for distributed dynamic systems, especially inertial network systems. This PhD study will address the solutions of these problems.

Chapter 3

STATISTICAL ESTIMATION AND TESTING THEORIES

3.1 Introduction

This chapter introduces mathematical formulations on statistical estimation theory and hypothesis testing theory, which are required in this thesis to understand the development of multisensor data fusion algorithms. Estimation theory is a powerful mathematical tool, which has been used in various engineering fields to accurately estimate the states of complex dynamic systems and to implement the effective control of the systems. Statistical testing theory is an auxiliary tool that is used to further confirm the validity of sensor data and the estimated system states. In aerospace engineering, these theories have successfully applied to the development of aircraft guidance, navigation and control systems. Section 3.2 introduces the conventional Kalman filter algorithms and analyses the statistical characteristics of the Kalman filter. Section 3.3 gives the information filter and Section 3.4 describes statistical hypothesis testing methods. Finally, a summary is given in Section 3.5.

3.2 The Kalman Filter

3.2.1 Stochastic Process Model

Since the Kalman filter was originally presented by R. E. Kalman, it has become a standard estimation method that is widely used in the development of navigation systems. In order to develop various forms of Kalman filter algorithms, the dynamic model of a stochastic process (system) must be constructed in the form of state space representations. In this thesis, the Kalman filter is used as a state estimator and does not perform any control functions. Therefore, the control input to the system is not considered in the system model.

The stochastic process model develops dynamic relations between the states of a stochastic process. It consists of a set of the first-order differential equations driven by random input noise, which describe the evolution of the stochastic system state in time. It can be represented in a generalised formulation in the continuous-time form as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t),t] + \mathbf{G}[\mathbf{x}(t),t]\mathbf{n}(t); \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{3.1}$$

where $\mathbf{f}$ is an $n-$known function vector, t denotes time, $\mathbf{x}(t)$ is an $n-$system state vector at time t with the initial value of $\mathbf{x}_0$, $\mathbf{n}(t)$ is an $q-$ additive process noise vector and $\mathbf{G}[\mathbf{x}(t_k),t_k]$ is an $n \times q$ function matrix. The process noise $\mathbf{n}(t)$ takes into account the perturbations to the system.

The discrete-time form of this continuous-time system model is needed for the computer implementation of the Kalman filter and can be formulated as follows:

$$\mathbf{x}(t_{k+1}) = \mathbf{f}[\mathbf{x}(t_k),t_{k+1},t_k] + \mathbf{G}[\mathbf{x}(t_k),t_k]\mathbf{n}(t_k); \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{3.2}$$

where t_k is the sampling time.

The discrete-time process model can be deduced by integrating the continuous-time process model between successive sampling times. The associated process noise and control input vectors must also be redefined to reflect the integration[41][91]. Hereafter, only the discrete-time process model will be considered in this thesis.

For the stochastic process model, the Kalman filter assumes that the sequence of the process noise $\mathbf{n}(t_k)$ is a white Gaussian process with zero mean and known covariance, and is independent of the system state $\mathbf{x}(t_k)$. The sequence of the system states $\mathbf{x}(t_k)$ is a Gauss-Markov process. The initial system state $\mathbf{x}(t_0)$ has known mean and covariance. Therefore, the following assumptions[91] are given:

$$\mathbf{x}(t_k) \sim \mathcal{N}(\hat{\mathbf{x}}(t_k), \mathbf{P}(t_k))$$

$$\mathrm{E}\{[\mathbf{x}(t_k)-\hat{\mathbf{x}}(t_k)][\mathbf{x}(t_k)-\hat{\mathbf{x}}(t_k)]^{\mathrm{T}}\}=\mathbf{P}(t_k)$$

$$\mathbf{n}(t_k) \sim \mathcal{N}(0,\mathbf{Q}(t_k))$$

$$\mathrm{E}[\mathbf{n}(t_k)\mathbf{n}^{\mathrm{T}}(t_i)]=\mathbf{Q}(t_k)\delta_{ki}$$

$$\mathrm{E}[\mathbf{n}(t_k)\mathbf{x}^{\mathrm{T}}(t_i)]=0$$

where $\mathcal{N}$ represents the Gaussian (normal) distribution, $\hat{\mathbf{x}}(t_k)$ is the estimate or mean value of $\mathbf{x}(t_k)$, $\mathbf{P}(t_k)$ is the symmetric, positive semidefinite covariance matrix of the state errors, $\mathrm{E}[\cdot]$ denotes the expectation operator, $\mathbf{Q}(t_k)$ is a covariance matrix of the process noise, which is positive semidefinite and δ_{ki} is the Kronecker delta function.

3.2.2 Stochastic Measurement Model

The stochastic measurement model develops the relations between the system states and physical quantities measured by the sensor systems. It can be represented in a generalised formulation in the discrete-time form as follows:

$$\mathbf{z}(t_k)=\mathbf{h}[\mathbf{x}(t_k),t_k]+\mathbf{w}(t_k) \tag{3.3}$$

where $\mathbf{h}$ is an $m-$known function vector, and the vectors $\mathbf{z}(t_k)$ and $\mathbf{w}(t_k)$ are an $m-$measurement vector of a sensor system and an $m-$additive measurement noise vector, respectively. The measurement noise accounts for effects of the measurement system errors on the measured physical quantities.

For the stochastic measurement model, the Kalman filter assumes that $\mathbf{z}(t_k)$ is a Gaussian distributed random variable at each sampling time and the sequence of the noise $\mathbf{w}(t_k)$ is a white Gaussian process with zero mean and known covariance and is independent of $\mathbf{n}(t_k)$ and $\mathbf{x}(t_k)$, separately. The following assumptions[91] are then given:

$$\mathbf{w}(t_k) \sim \mathcal{N}(0,\mathbf{R}(t_k))$$

$$\mathrm{E}[\mathbf{w}(t_k)\mathbf{w}^{\mathrm{T}}(t_i)]=\mathbf{R}(t_k)\delta_{ki}$$

$$\mathrm{E}[\mathbf{w}(t_k)\mathbf{x}^{\mathrm{T}}(t_i)]=0$$

$$\mathrm{E}[\mathbf{w}(t_k)\mathbf{n}^{\mathrm{T}}(t_i)]=0$$

where $\mathbf{R}(t_k)$ is a positive definite covariance matrix of the measurement noise.

However, in some stochastic processes and measurement systems, it may be not appropriate to represent both the process noise and the measurement noise by using a white Gaussian process. Hence, state augmentation techniques are frequently used to adjust the system model and the measurement model to fit the requirements of the Kalman filter[41][91].

3.2.3 Stochastic Estimation Model

The Kalman filter estimates the state of a stochastic process using the process and measurement models with the assumptions given in Sections 3.2.1 and 3.2.2. Two estimates of the state are distinguished: the estimate of the state $\mathbf{x}(t_k)$, $\hat{\mathbf{x}}(t_k^+)$, is a conditional estimate conditioned on the measurement history up to the current time t_k, whereas the estimate of the state $\mathbf{x}(t_k)$, $\hat{\mathbf{x}}(t_k^-)$, is an estimate conditioned on the measurement history up through the previous sample time t_{k-1}. $\hat{\mathbf{x}}(t_k^-)$ is known as the predicted state derived from the process model with the time update. The associated conditional mean and covariances are defined as follows:

$$\hat{\mathbf{x}}(t_k^+) = \mathrm{E}[\mathbf{x}(t_k) | \{\mathbf{z}(t_k)\}]$$

$$\hat{\mathbf{x}}(t_k^-) = \mathrm{E}[\mathbf{x}(t_k) | \{\mathbf{z}(t_{k-1})\}]$$

Defining the errors corresponding to these two estimates as

$$\tilde{\mathbf{x}}(t_k^+) = \mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)$$

$$\tilde{\mathbf{x}}(t_k^-) = \mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)$$

then the covariances of these errors can be defined as follows:

$$\mathbf{P}(t_k^+) = \mathrm{E}[\tilde{\mathbf{x}}(t_k^+)\tilde{\mathbf{x}}^{\mathrm{T}}(t_k^+)] = \mathrm{E}\{[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)][\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)]^{\mathrm{T}} | \{\mathbf{z}(t_k)\}\}$$

$$\mathbf{P}(t_k^-) = \mathrm{E}[\tilde{\mathbf{x}}(t_k^-)\tilde{\mathbf{x}}^{\mathrm{T}}(t_k^-)] = \mathrm{E}\{[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)][\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)]^{\mathrm{T}} | \{\mathbf{z}(t_{k-1})\}\}$$

The estimation process of the Kalman filter is illustrated in Figure 3.1 where the predicated state and the current measurement are combined by the Kalman filter to obtain the current state estimate.

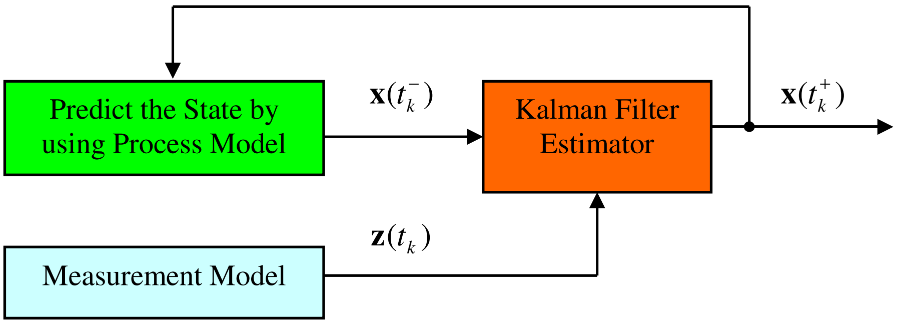


Figure 3.1 The Estimation Process

3.2.4 Linear Kalman Filter

The linear Kalman filter is a standard Kalman filter applicable to linear stochastic processes. The Kalman filter models are formulated as follows:

$$\mathbf{x}(t_k) = \Phi(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{n}(t_{k-1}) \tag{3.4}$$

$$\mathbf{z}(t_k) = \mathbf{H}(t_k)\mathbf{x}(t_k) + \mathbf{w}(t_k) \tag{3.5}$$

where Φ is an $n \times n$ state transition matrix and $\mathbf{H}$ is an $m \times n$ measurement matrix.

The predicted state and measurement are computed as follows[41]:

$$\hat{\mathbf{x}}(t_k^-) = \Phi(t_k, t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+) \tag{3.6}$$

$$\hat{\mathbf{z}}(t_k^-) = \mathbf{H}(t_k)\hat{\mathbf{x}}(t_k^-) \tag{3.7}$$

then the covariance of the predicted state error is:

$$\begin{aligned}\mathbf{P}(t_k^-) &= \mathrm{E}[\tilde{\mathbf{x}}(t_k^-)\tilde{\mathbf{x}}^{\mathrm{T}}(t_k^-)] \\ &= \mathrm{E}\{[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)][\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)]^{\mathrm{T}} \big| \{\mathbf{z}(t_{k-1})\}\} \\ &= \mathrm{E}\{[\Phi(t_k, t_{k-1})\tilde{\mathbf{x}}(t_{k-1}^+) + \mathbf{G}(t_{k-1})\mathbf{n}(t_{k-1})][\Phi(t_k, t_{k-1})\tilde{\mathbf{x}}(t_{k-1}^+) + \mathbf{G}(t_{k-1})\mathbf{n}(t_{k-1})]^{\mathrm{T}}\}\end{aligned}$$

According to the assumptions given in Sections 3.2.1 and 3.2.2, the above equation can be simplified as follows:

$$\mathbf{P}(t_k^-) = \Phi(t_k, t_{k-1})\mathbf{P}(t_{k-1}^+)\Phi^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1}) \tag{3.8}$$

Defining the innovation $\mathbf{r}(t_k)$ as

$$\mathbf{r}(t_k) = \mathbf{z}(t_k) - \hat{\mathbf{z}}(t_k^-) = \mathbf{H}(t_k)\tilde{\mathbf{x}}(t_k^-) + \mathbf{w}(t_k) \tag{3.9}$$

then the covariance of the innovation is computed as:

$$\begin{aligned}\mathbf{S}(t_k) &= \mathrm{E}[\mathbf{r}(t_k)\mathbf{r}^{\mathrm{T}}(t_k)] \\ &= \mathbf{H}(t_k)\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k) + \mathbf{R}(t_k)\end{aligned} \tag{3.10}$$

And the covariances of the innovation and predicted state error are

$$\mathbf{S}_{\mathbf{r}\tilde{\mathbf{x}}}(t_k) = \mathrm{E}[\mathbf{r}(t_k)\tilde{\mathbf{x}}^{\mathrm{T}}(t_k^-)] = \mathbf{H}(t_k)\mathbf{P}(t_k^-) \tag{3.11}$$

$$\mathbf{S}_{\tilde{\mathbf{x}}\mathbf{r}}(t_k) = \mathrm{E}[\tilde{\mathbf{x}}(t_k^-)\mathbf{r}^{\mathrm{T}}(t_k)] = \mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k) \tag{3.12}$$

Therefore, $\mathbf{S}_{\tilde{\mathbf{x}}\mathbf{r}}(t_k) = \mathbf{S}^{\mathrm{T}}_{\mathbf{r}\tilde{\mathbf{x}}}(t_k)$

With the assumption that the prior state and measurement $\hat{\mathbf{x}}(t_k^-)$ and $\hat{\mathbf{z}}(t_k^-)$ are known, the current measurement $\mathbf{z}(t_k)$ can be used to update the prior state estimate in accordance with the following equation

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)\mathbf{r}(t_k) \tag{3.13}$$

or

$$\begin{aligned}\hat{\mathbf{x}}(t_k^+) &= \hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)[\mathbf{z}(t_k) - \hat{\mathbf{z}}(t_k)] \\ &= [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k)]\hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)\mathbf{z}(t_k)\end{aligned} \tag{3.14}$$

where $\mathbf{K}(t_k)$ is a blending factor to be determined. Then, the error covariance of the updated state estimate $\hat{\mathbf{x}}(t_k^+)$, $\mathbf{P}(t_k^+)$, can be computed as follows:

$$\begin{aligned}\mathbf{P}(t_k^+) &= \mathrm{E}[\tilde{\mathbf{x}}(t_k^+)\tilde{\mathbf{x}}^{\mathrm{T}}(t_k^+)] \\ &= \mathrm{E}\{[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)][\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)]^{\mathrm{T}}\} \\ &= \mathrm{E}\{[(\mathbf{I} - \mathbf{KH})\tilde{\mathbf{x}}(t_k^-) - \mathbf{Kw}(t_k)][(\mathbf{I} - \mathbf{KH})\tilde{\mathbf{x}}(t_k^-) - \mathbf{K}(t_k)\mathbf{w}(t_k)]^{\mathrm{T}}\}\end{aligned}$$

Note that the a priori estimation error $\tilde{\mathbf{x}}(t_k^-)$ is independent of the measurement noise $\mathbf{w}(t_k)$, therefore,

$$\mathbf{P}(t_k^+) = [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k)]\mathbf{P}(t_k^-)[\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t)]^{\mathrm{T}} + \mathbf{K}(t_k)\mathbf{R}(t_k)\mathbf{K}^{\mathrm{T}}(t_k) \tag{3.15}$$

An optimal blending factor $\mathbf{K}(t_k)$, which minimise the mean-square estimation error and is known as the Kalman gain, can be obtained by the optimisation of Eq. (3.15) as follows[41]

$$\mathbf{K}(t_k) = \mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)\mathbf{S}^{-1}(t_k) \tag{3.16}$$

Expanding Eq. (3.15) and substituting Eq. (3.16) into the resultant equation leads to

$$\mathbf{P}(t_k^+) = \mathbf{P}(t_k^-) - \mathbf{K}(t_k)\mathbf{H}(t_k)\mathbf{P}(t_k^-) \tag{3.17}$$

or

$$\mathbf{P}(t_k^+) = [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k)]\mathbf{P}(t_k^-) \tag{3.18}$$

or $$\mathbf{P}(t_k^+) = \mathbf{P}(t_k^-) - \mathbf{K}(t_k)\mathbf{S}(t_k)\mathbf{K}^{\mathrm{T}}(t_k) \tag{3.19}$$

These three expressions for $\mathbf{P}(t_k^+)$ fail to assure both the positive definiteness and symmetry of the error covariance update, and are valid only for the Kalman optimal gain. In contrast, Eq. (3.15), known as the Joseph form, can maintain the symmetry and the positive definiteness of the error covariance propagation and is valid for any value of gain.

The mechanism of the standard Kalman filter algorithm is summarised in Table 3-1.

Table 3-1 Standard Kalman Filter Algorithms

Step 1: Initialisation

$$\mathbf{P}(t_0) = \mathbf{P}_0\,;\ \hat{\mathbf{x}}(t_0) = \mathbf{x}_0$$

Step 2: Time update (effect of dynamics)

$$\hat{\mathbf{x}}(t_k^-) = \Phi(t_k, t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+)$$

$$\mathbf{P}(t_k^-) = \Phi(t_k, t_{k-1})\mathbf{P}(t_{k-1}^+)\Phi^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1})$$

Step 3: Measurement update (effect of measurement)

$$\mathbf{r}(t_k) = \mathbf{z}(t_k) - \mathbf{H}(t_k)\hat{\mathbf{x}}(t_k^-)\,,\ \mathbf{S}(t_k) = \mathbf{H}(t_k)\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k) + \mathbf{R}(t_k)$$

$$\mathbf{K}(t_k) = \mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)\mathbf{S}^{-1}(t_k)$$

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)\mathbf{r}(t_k)\,,\ \mathbf{P}(t_k^+) = \mathbf{P}(t_k^-) - \mathbf{K}(t_k)\mathbf{H}(t_k)\mathbf{P}(t_k^-)$$

3.2.5 Extended Kalman Filter

The extended Kalman filter is used to handle the estimation problems which occur in non-linear stochastic processes. A typical non-linear process model can be written in the form:

$$\mathbf{x}(t_k) = \mathbf{f}[\mathbf{x}(t_{k-1}), t_k, t_{k-1}] + \mathbf{G}[\mathbf{x}(t_{k-1}), t_{k-1}]\mathbf{n}(t_{k-1}) \tag{3.20}$$

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k), t_k] + \mathbf{w}(t_k) \tag{3.21}$$

where $\mathbf{f}$ and $\mathbf{h}$ are known non-linear functions.

To apply the Kalman filter to estimation problems in a nonlinear process, these nonlinear models must be linearised. Two methods can be used to linearise nonlinear models. One method linearises the nonlinear models about some nominal trajectory in the state space, which is independent of the measurements. The nominal trajectory is pre-computable. The resultant filter is referred to as a linearised Kalman filter. The other method linearises the non-linear models about an estimated trajectory that is continuously updated by the state estimates resulting from the measurements. The filter is known as an extended Kalman filter.

The predicted state and measurement are computed as

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{f}[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}] \tag{3.22}$$

$$\hat{\mathbf{z}}(t_k^-) = \mathbf{h}[\hat{\mathbf{x}}(t_k^-), t_k] \tag{3.23}$$

Consider the perturbation of the current state from the predicted state $\hat{\mathbf{x}}(t_{k-1}^+)$. The system model in Eq. (3.20) can be approximated by expanding about $\hat{\mathbf{x}}(t_{k-1}^+)$ as follows:

$$\begin{aligned}\mathbf{x}(t_k) \approx\ & \mathbf{f}[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}] + \Phi[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}]\tilde{\mathbf{x}}(t_{k-1}^+) \\ & + \mathbf{G}[\mathbf{x}(t_{k-1}^+), t_{k-1}]\mathbf{n}(t_{k-1})\end{aligned} \tag{3.24}$$

where the higher terms of the Taylor series expansion have been ignored and

$$\Phi(\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}) = \left.\frac{\partial \mathbf{f}[\mathbf{x}(t_{k-1}), t_k, t_{k-1}]}{\partial \mathbf{x}}\right|_{\mathbf{x}(t_{k-1}) = \hat{\mathbf{x}}(t_{k-1}^+)} = \begin{bmatrix} \frac{\partial \mathrm{f}_1}{\partial x_1} & \cdots & \frac{\partial \mathrm{f}_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{f}_\mathrm{n}}{\partial x_1} & \cdots & \frac{\partial \mathrm{f}_\mathrm{n}}{\partial x_n} \end{bmatrix}$$

$$\tilde{\mathbf{x}}(t_{k-1}^+) = \mathbf{x}(t_{k-1}) - \hat{\mathbf{x}}(t_{k-1}^+)$$

Subtracting Eq. (3.22) from Eq. (3.24), the predicted state error is derived as follows:

$$\tilde{\mathbf{x}}(t_k^-) = \Phi[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}]\tilde{\mathbf{x}}(t_{k-1}^+) + \mathbf{G}[\mathbf{x}(t_{k-1}^+), t_{k-1}]\mathbf{n}(t_{k-1}) \tag{3.25}$$

where

$$\tilde{\mathbf{x}}(t_k^-) = \mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^-)$$

From Eq.(3.25), the associated error covariance is computed as:

$$\begin{aligned}\mathbf{P}(t_k^-) =\ & \Phi[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}]\mathbf{P}(t_{k-1}^+)\Phi^{\mathrm{T}}[\hat{\mathbf{x}}(t_{k-1}^+), t_k, t_{k-1}] \\ & + \mathbf{G}[\mathbf{x}(t_{k-1}^+), t_{k-1}]\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}[\mathbf{x}(t_{k-1}^+), t_{k-1}]\end{aligned}$$

Now consider the perturbation of the measurement model from the predicted state $\hat{\mathbf{x}}(t_k^-)$. The measurement model in Eq. (3.21) can be approximated as follows:

$$\mathbf{z}(t_k) \approx \mathbf{h}[\hat{\mathbf{x}}(t_k^-), t_k] + \mathbf{H}[\hat{\mathbf{x}}(t_k^-), t_k]\tilde{\mathbf{x}}(t_k^-) + \mathbf{w}(t_k) \tag{3.26}$$

where the higher terms of the Taylor series expansion have been ignored and

$$\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]=\left.\frac{\partial\mathbf{h}[\mathbf{x}(t_k),t_k]}{\partial\mathbf{x}}\right|_{\mathbf{x}(t_k)=\hat{\mathbf{x}}(t_k^-)}=\begin{bmatrix}\frac{\partial \mathrm{h}_1}{\partial x_1} & \cdots & \frac{\partial \mathrm{h}_1}{\partial x_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial \mathrm{h}_m}{\partial x_1} & \cdots & \frac{\partial \mathrm{h}_m}{\partial x_n}\end{bmatrix}$$

Subtracting Eq. (3.23) from Eq. (3.26), the measurement innovation is given by

$$\mathbf{r}(t_k)=\mathbf{z}(t_k)-\hat{\mathbf{z}}(t_k^-)=\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]\tilde{\mathbf{x}}(t_k^-)+\mathbf{w}(t_k) \qquad (3.27)$$

then the covariance of the measurement innovation is computed as:

$$\begin{aligned}\mathbf{S}(t_k)&=\mathrm{E}[\mathbf{r}(t_k)\mathbf{r}^{\mathrm{T}}(t_k)]\\&=\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}[\hat{\mathbf{x}}(t_k^-),t_k]+\mathbf{R}(t_k)\end{aligned} \qquad (3.28)$$

Applying the standard Kalman filter to the linearised models given in Eqs. (3.25) and (3.27), the measurement updates are computed by

$$\hat{\mathbf{x}}(t_k^+)=\hat{\mathbf{x}}(t_k^-)+\mathbf{K}(t_k)\mathbf{r}(t_k) \qquad (3.29)$$

$$\mathbf{P}(t_k^+)=\mathbf{P}(t_k^-)-\mathbf{K}(t_k)\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{P}(t_k^-) \qquad (3.30)$$

where $\mathbf{K}(t_k)$ is the Kalman filter gain matrix.

$$\mathbf{K}(t_k)=\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{S}^{-1}(t_k) \qquad (3.31)$$

The extended Kalman filter algorithm is summarised in Table 3-2.

In this thesis, the linear and extended Kalman filter algorithms are referred to as conventional Kalman filter algorithms. An iterative modular algorithm structure for the conventional Kalman filter algorithm is shown in Figure 3.2 where the three modules are initialisation, predictor and estimator.

Table 3-2 The Extended Kalman Filter Algorithm

Step 1: Initialisation

$$\mathbf{P}(t_0)=\mathbf{P}_0\,;\ \hat{\mathbf{x}}(t_0)=\mathbf{x}_0$$

Step 2: Time update (effect of dynamics)

$$\hat{\mathbf{x}}(t_k^-)=\mathbf{f}[\hat{\mathbf{x}}(t_{k-1}^+),t_k,t_{k-1}]$$

$$\begin{aligned}\mathbf{P}(t_k^-)&=\Phi[\hat{\mathbf{x}}(t_{k-1}^+),t_k,t_{k-1}]\mathbf{P}(t_{k-1}^+)\Phi^{\mathrm{T}}[\hat{\mathbf{x}}(t_{k-1}^+),t_k,t_{k-1}]\\&\quad+\mathbf{G}[\mathbf{x}(t_{k-1}^+),t_{k-1}]\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}[\mathbf{x}(t_{k-1}^+),t_{k-1}]\end{aligned}$$

$$\Phi(\hat{\mathbf{x}}(t_{k-1}^+),t_k,t_{k-1})=\left.\begin{bmatrix}\dfrac{\partial \mathrm{f}_1}{\partial x_1} & \cdots & \dfrac{\partial \mathrm{f}_1}{\partial x_n}\\ \vdots & \ddots & \vdots\\ \dfrac{\partial \mathrm{f}_\mathrm{n}}{\partial x_1} & \cdots & \dfrac{\partial \mathrm{f}_\mathrm{n}}{\partial x_n}\end{bmatrix}\right|_{\mathbf{x}(t_{k-1})=\hat{\mathbf{x}}(t_{k-1}^+)}$$

Step 3: Measurement update (effect of measurement)

$$\mathbf{r}(t_k)=\mathbf{z}(t_k)-\mathbf{h}[\hat{\mathbf{x}}(t_k^-),t_k]$$

$$\mathbf{S}(t_k)=\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}[\hat{\mathbf{x}}(t_k^-),t_k]+\mathbf{R}(t_k)$$

$$\mathbf{K}(t_k)=\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{S}^{-1}(t_k)$$

$$\hat{\mathbf{x}}(t_k^+)=\hat{\mathbf{x}}(t_k^-)+\mathbf{K}(t_k)\mathbf{r}(t_k)$$

$$\mathbf{P}(t_k^+)=\mathbf{P}(t_k^-)-\mathbf{K}(t_k)\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]\mathbf{P}(t_k^-)$$

$$\mathbf{H}[\hat{\mathbf{x}}(t_k^-),t_k]=\left.\begin{bmatrix}\dfrac{\partial \mathrm{h}_1}{\partial x_1} & \cdots & \dfrac{\partial \mathrm{h}_1}{\partial x_n}\\ \vdots & \ddots & \vdots\\ \dfrac{\partial \mathrm{h}_m}{\partial x_1} & \cdots & \dfrac{\partial \mathrm{h}_m}{\partial x_n}\end{bmatrix}\right|_{\mathbf{x}(t_k)=\hat{\mathbf{x}}(t_k^-)}$$

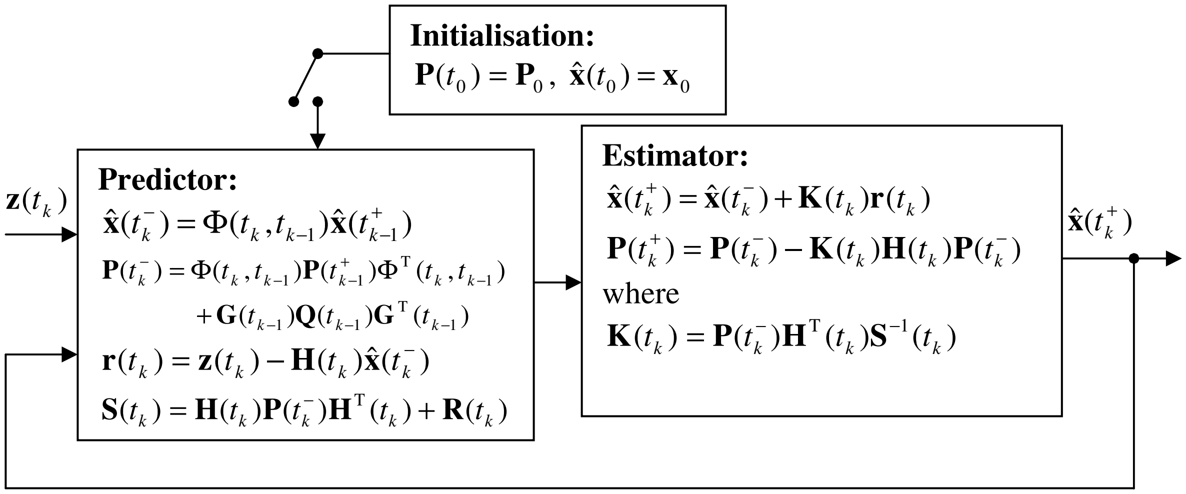

Initialisation:
$\mathbf{P}(t_0)=\mathbf{P}_0,\ \hat{\mathbf{x}}(t_0)=\mathbf{x}_0$
$\mathbf{z}(t_k)$
Predictor:
$\hat{\mathbf{x}}(t_k^-)=\Phi(t_k,t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+)$
$\mathbf{P}(t_k^-)=\Phi(t_k,t_{k-1})\mathbf{P}(t_{k-1}^+)\Phi^{\mathrm{T}}(t_k,t_{k-1})$
$+\mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1})$
$\mathbf{r}(t_k)=\mathbf{z}(t_k)-\mathbf{H}(t_k)\hat{\mathbf{x}}(t_k^-)$
$\mathbf{S}(t_k)=\mathbf{H}(t_k)\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)+\mathbf{R}(t_k)$
Estimator:
$\hat{\mathbf{x}}(t_k^+)=\hat{\mathbf{x}}(t_k^-)+\mathbf{K}(t_k)\mathbf{r}(t_k)$
$\mathbf{P}(t_k^+)=\mathbf{P}(t_k^-)-\mathbf{K}(t_k)\mathbf{H}(t_k)\mathbf{P}(t_k^-)$
where
$\mathbf{K}(t_k)=\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)\mathbf{S}^{-1}(t_k)$
$\hat{\mathbf{x}}(t_k^+)$

Figure 3.2 Conventional Kalman Filter Algorithm Structure

3.2.6 Statistics of Kalman Estimation Errors and Residuals

The conventional Kalman filter algorithm has been used as a standard means of optimal or near-optimal estimation of the states of a stochastic system. By examining the predictor and the estimator given in Figure 3.2, the filter algorithm also provides very useful statistical information that can be used to monitor both the convergence and the consistency of the filter estimation procedure. As illustrated in Figure 3.3, these statistics can be obtained by generating information at different stages of the filter algorithm.

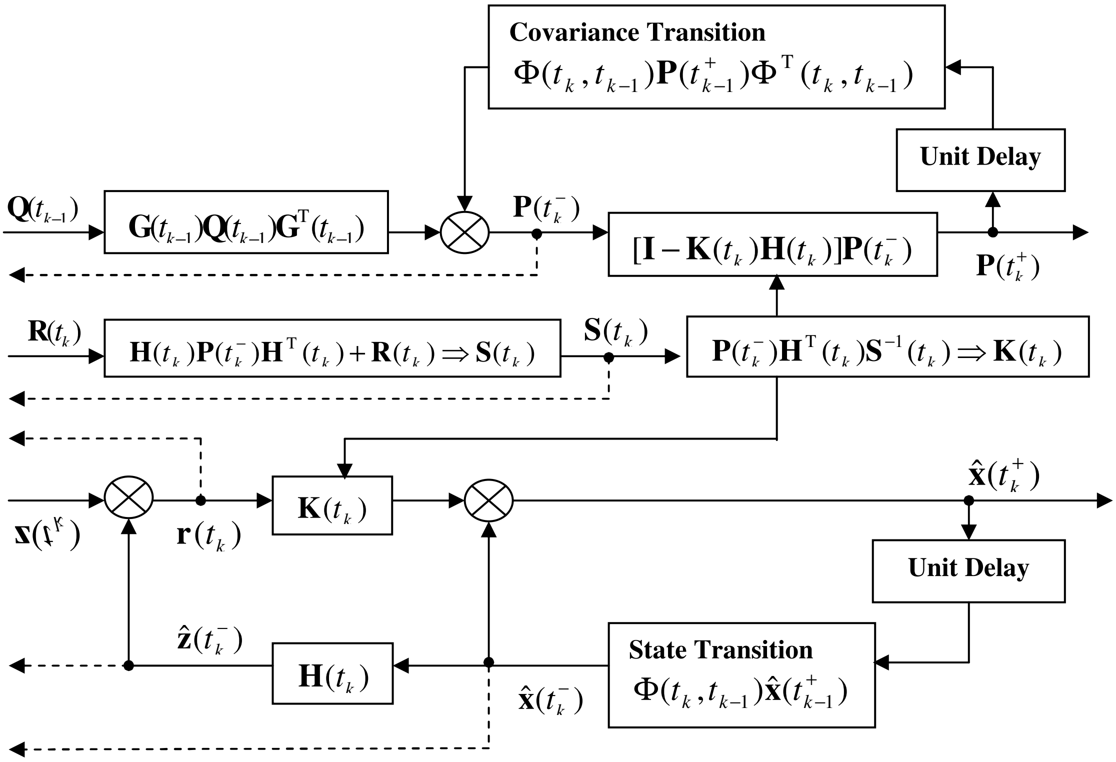

Covariance Transition
$\Phi(t_k,t_{k-1})\mathbf{P}(t_{k-1}^{+})\Phi^{\mathrm{T}}(t_k,t_{k-1})$
Unit Delay
$\mathbf{Q}(t_{k-1})$
$\mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1})$
$\mathbf{P}(t_k^-)$
$[\mathbf{I}-\mathbf{K}(t_k)\mathbf{H}(t_k)]\mathbf{P}(t_k^-)$
$\mathbf{P}(t_k^+)$
$\mathbf{R}(t_k)$
$\mathbf{H}(t_k)\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)+\mathbf{R}(t_k)\Rightarrow\mathbf{S}(t_k)$
$\mathbf{S}(t_k)$
$\mathbf{P}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k)\mathbf{S}^{-1}(t_k)\Rightarrow\mathbf{K}(t_k)$
$\mathbf{z}(t_k)$
$\mathbf{r}(t_k)$
$\mathbf{K}(t_k)$
$\hat{\mathbf{x}}(t_k^+)$
Unit Delay
$\hat{\mathbf{z}}(t_k^-)$
$\mathbf{H}(t_k)$
$\hat{\mathbf{x}}(t_k^-)$
State Transition
$\Phi(t_k,t_{k-1})\hat{\mathbf{x}}(t_{k-1}^{+})$

Figure 3.3 Kalman Filter Outputs

The outputs of the filter predictor $\mathbf{r}(t_k)$ and $\mathbf{S}(t_k)$ are the filter innovation and the filter innovation covariance, respectively. The outputs of the filter estimator $\mathbf{r}(t_k^+)$ and $\mathbf{S}(t_k^+)$ are the filter residual and the filter residual covariance as follows:

$$\mathbf{r}(t_k^+)=\mathbf{z}(t_k)-\mathbf{H}(t_k)\mathbf{x}(t_k^+)$$

$$\mathbf{S}(t_k^+)=\mathbf{H}(t_k)\mathbf{P}(t_k^+)\mathbf{H}^{\mathrm{T}}(t_k)+\mathbf{R}(t_k)$$

Theoretically, it has been shown that the filter innovation and residual processes[41] are a zero-mean white Gaussian random process of the given covariance. This feature can be exploited in the analysis of the Kalman filter integrity for some practical purposes, for example, consistency checking of the measurement data, sensor failure detection and analysis of the filter divergence.

Defining a normalised quadratic innovation function by the following terms

$$NQI(t_k) = \mathbf{r}^{\mathrm{T}}(t_k)\mathbf{S}^{-1}(t_k)\mathbf{r}(t_k) \tag{3.32}$$

then, the $NQI(t_k)$ is a measure of the inconsistency of the measurement data or the filter innovation (residual) and is a χ^2 -distributed random variable with m degrees of freedom where m is the number of statistically independent measurements. Testing $NQI(t_k)$ for consistency of the filter innovation can be used to detect sensor failure.

The output of the filter estimator $\mathbf{P}(t_k^+)$ is a measure of the uncertainty of the filter state estimate $\hat{\mathbf{x}}(t_k^+)$. The uncertainties along the different state space directions can be represented geometrically as follows:

$$[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)]^{\mathrm{T}}\mathbf{P}^{-1}(t_k^+)[\mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k^+)] = 1 \tag{3.33}$$

For a 2-dimensional state vector, it is given by an ellipse shown in Figure 3.4.

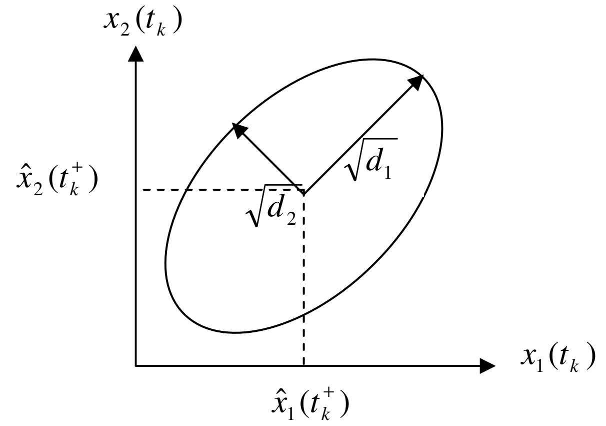


Figure 3.4 The Ellipsoid of Estimate Uncertainty

The axes of the ellipsoid are oriented along the singular directions of $\mathbf{P}(t_k^+)$. Applying the singular value decomposition to $\mathbf{P}(t_k^+)$, Eq.(3.33) can be rewritten in the following form.

$$[\mathbf{U}^{-1}\tilde{\mathbf{x}}(t_k^+)]^{\mathrm{T}}\mathbf{D}^{-1}(t_k^+)[\mathbf{U}^{-1}\tilde{\mathbf{x}}(t_k^+)]=1$$

where $\mathbf{D}$ is a diagonal matrix containing the singular values of $\mathbf{P}(t_k^+)$, which are the lengths of the ellipsoid axes. The columns of $\mathbf{U}$ are singular vectors that indicate the singular directions and form the orthogonal basis in the state space. Small singular values correspond to directions in the state space with small uncertainties while large singular values indicate directions with large uncertainties. Convergence of the state estimate means that the ellipsoid shrinks in all directions. Therefore, analysis of the error covariance matrix $\mathbf{P}(t_k^+)$ is used to determine whether the estimation procedure is convergent or divergent.

Defining a normalised error quadratic function as follows:

$$NEQ(t_k)=[\hat{\mathbf{x}}(t_k^+)-\bar{\mathbf{x}}(t_k)]^{\mathrm{T}}\mathbf{P}^{-1}(t_k^+)[\hat{\mathbf{x}}(t_k^+)-\bar{\mathbf{x}}(t_k)] \tag{3.34}$$

where $\bar{\mathbf{x}}(t_k)$ is the true value of the system state $\mathbf{x}(t_k)$.

$NEQ(t_k)$ is a measure of the uncertainty of the filter state estimate and is a χ^2-distributed random variable with n degrees of freedom. Testing $NEQ(t_k)$ for consistency of the state estimation enables abnormalities of the estimated system states to be detected. However, this test is only applicable to system simulation because $\bar{\mathbf{x}}(t_k)$ has to be known.

Defining $\Delta\mathbf{x}(t_k)=\hat{\mathbf{x}}(t_k^-)-\hat{\mathbf{x}}(t_k^+)$, then $\Delta\mathbf{x}(t_k)$ is a zero-mean Gaussian random variable with the known covariance of $\mathbf{P}_{\Delta\mathbf{x}}(t_k)=\mathbf{K}(t_k)\mathbf{H}(t_k)\mathbf{P}(t_k^-)$. When the system model is highly accurate, an improved NEQ function is designed to check abnormal changes of the estimated system states as follows:

$$INEQ(t_k)=\Delta\mathbf{x}^{\mathrm{T}}(t_k)\mathbf{P}_{\Delta\mathbf{x}}^{-1}(t_k^+)\Delta\mathbf{x}(t_k) \tag{3.35}$$

Analysis of the state error covariance matrix is used to determine if the Kalman filter converges, but is unable to verify if the Kalman filter converges to the correct value.

From Figure 3.4, the accuracy of the error covariance matrix depends solely on the system model ($\mathbf{\Phi},\mathbf{Q},\mathbf{P}_0$) and the measurement model ($\mathbf{H},\mathbf{R}$). In other words, the covariance recursion is independent of the actual measurements taken, and thus it can be computed without knowledge of the realised measurement values. For this reason, $\mathbf{P}(t_k^-)$ and $\mathbf{P}(t_k^+)$ can be pre-computed before actual measurements are available. This pre-computability allows early design tradeoffs of expected estimation accuracy versus sensor system accuracy to be undertaken. Proper estimation accuracy can be evaluated by off line simulation of different sensor systems and system models until the error covariance analysis meets the required accuracy.

3.2.7 Drawbacks of Kalman Filter

The conventional Kalman filter algorithms, when implemented on a computer, may undergo computational inaccuracies owing to computer roundoff errors and the occurrence of very small values in the error covariance matrix. In particular, computational inaccuracies in the covariance matrix update procedure can cause the computed covariance matrix to become numerically inaccurate, resulting in a loss of symmetry and positive semi-definitiveness of the covariance matrix. These situations may lead to divergence and instability of the conventional Kalman filter.

In order to overcome the computational inaccuracies in the conventional Kalman filter algorithms, many numerically accurate forms of the Kalman filter have been introduced, for example, various squared-root filters or the Potter filter[40][91]. These improved Kalman filter algorithms seek to propagate and update some factorisation of the error covariance matrix rather than the error covariance matrix itself. However, it should be noted that the Potter filter increases the computational accuracy of the error covariance matrix but loses the simplicity of the standard Kalman filter algorithm.

3.3 The Information Filter

Both the conventional Kalman and the Potter filters need an accurate estimate of the initial error covariance $\mathbf{P}_0$. However, it may be impractical to obtain an accurate estimate of $\mathbf{P}_0$ in some applications because there may be no a priori knowledge of the initial system states or the initial states available are inconsistent.

In addition, with the increase of the number of measurements, for example, the data fusion filters in distributed multiple sensor systems must execute matrix inverse operation on the innovation covariance, which may be time consuming particularly. In order to resolve such problems, an information form of the Kalman filter has been introduced. This information filter is an algebraically equivalent Kalman filter, but processes and propagates the inverse matrix or the inverse matrix square root of the error covariance. This form possesses several unique characteristics and is suited to problems where the measurement dimension is large or where there is no a priori knowledge of the initial system state (particularly allowing a startup procedure in the case of singular $\mathbf{P}_0$).

3.3.1 The Linear Information Filter

In the linear Kalman filter algorithm, the state update equation is given by

$$\hat{\mathbf{x}}(t_k^+) = [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k)]\hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)\mathbf{z}(t_k) \tag{3.36}$$

Assuming $\mathbf{P}(t_k^-)$ is non-singular, Eq. (3.18) can be represented as

$$[\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k)] = \mathbf{P}(t_k^+)\mathbf{P}^{-1}(t_k^-) \tag{3.37}$$

Substituting Eq. (3.37) into Eq. (3.36) and pre-multiplying the resultant expression by $\mathbf{P}^{-1}(t_k^+)$ leads to

$$\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \mathbf{P}^{-1}(t_k^+)\mathbf{K}(t_k)\mathbf{z}(t_k) \tag{3.38}$$

From Eqs. (3.10), (3.16) and (3.18),

$$\mathbf{K}(t_k) = \mathbf{P}(t_k^+)\mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k) \tag{3.39}$$

Substituting Eq. (3.39) into Eq. (3.38) leads to

$$\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{z}(t_k) \tag{3.40}$$

This expression is known as the inverse covariance state update equation.

From Eqs. (3.37) and (3.39),

$$\mathbf{P}^{-1}(t_k^+) = \mathbf{P}^{-1}(t_k^-) + \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{H}(t_k) \tag{3.41}$$

This expression is known as the inverse covariance update equation.

From Eq. (3.8), the inverse of the a priori error covariance is given by

$$\mathbf{P}^{-1}(t_k^-) = [\mathbf{\Phi}(t_k, t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1})]^{-1} \tag{3.42}$$

Applying the matrix inversion lemma

$$(\mathbf{Z} + \mathbf{X}^{\mathrm{T}}\mathbf{Y})^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{X}^{\mathrm{T}}(\mathbf{I} + \mathbf{Y}\mathbf{Z}^{-1}\mathbf{X}^{\mathrm{T}})\mathbf{Y}\mathbf{Z}^{-1}$$

to the right side of Eq. (3.42) by identifying

$$\mathbf{Z} = \mathbf{\Phi}(t_k, t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_k, t_{k-1});\ \mathbf{X}^{\mathrm{T}} = \mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1});\mathbf{Y} = \mathbf{G}^{\mathrm{T}}(t_{k-1}),$$

it has been shown that[40]

$$\mathbf{P}^{-1}(t_k^-) = [\mathbf{I} - \mathbf{\Pi}(t_k)\mathbf{G}^{\mathrm{T}}(t_{k-1})]\mathbf{M}(t_k) \tag{3.43}$$

where

$$\mathbf{M}(t_k) = \mathbf{\Phi}^{\mathrm{T}}(t_{k-1}, t_k)\mathbf{P}^{-1}(t_{k-1}^+)\mathbf{\Phi}(t_{k-1}, t_k) \tag{3.44}$$

$$\mathbf{\Pi}(t_k) = \mathbf{M}(t_k)\mathbf{G}(t_{k-1})[\mathbf{G}^{\mathrm{T}}(t_{k-1})\mathbf{M}(t_k)\mathbf{G}(t_{k-1}) + \mathbf{Q}^{-1}(t_{k-1})]^{-1} \tag{3.45}$$

In the information filter, the a priori information state estimate $\hat{\mathbf{y}}(t_k^-)$, the a priori information matrix $\mathbf{Y}(t_k^-)$, the information state $\hat{\mathbf{y}}(t_k^+)$, the information matrix $\mathbf{Y}(t_k^+)$, the new information vector $\mathbf{u}(t_k)$ and the new information matrix $\mathbf{U}(t_k)$ are defined as follows:

$$\hat{\mathbf{y}}(t_k^-) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) \tag{3.46a}$$

$$\mathbf{Y}(t_k^-) = \mathbf{P}^{-1}(t_k^-) \tag{3.46b}$$

$$\hat{\mathbf{y}}(t_k^+) = \mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) \tag{3.46c}$$

$$\mathbf{Y}(t_k^+) = \mathbf{P}^{-1}(t_k^+) \tag{3.46d}$$

$$\mathbf{u}(t_k) = \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{z}(t_k) \tag{3.46e}$$

$$\mathbf{U}(t_k) = \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{H}(t_k) \tag{3.46f}$$

From Eqs. (3.43), (3.46a) and (3.46c)

$$\hat{\mathbf{y}}(t_k^-) = [\mathbf{I} - \mathbf{\Pi}(t_k)\mathbf{G}^{\mathrm{T}}(t_{k-1})]\mathbf{\Phi}^{\mathrm{T}}(t_{k-1}, t_k)\hat{\mathbf{y}}(t_{k-1}^+) \tag{3.47a}$$

Substituting Eqs. (3.46a), (3.46c) and (3.46e) into Eq. (3.40) leads to

$$\hat{\mathbf{y}}(t_k^+) = \hat{\mathbf{y}}(t_k^-) + \mathbf{u}(t_k) \tag{3.47b}$$

Substituting Eqs. (3.46b), (3.46d) and (3.46f) into Eq. (3.41) yields

$$\mathbf{Y}(t_k^+) = \mathbf{Y}(t_k^-) + \mathbf{U}(t_k) \tag{3.47c}$$

From Eq. (3.44), the inverse of the state transition matrix instead of the state transition matrix $\mathbf{\Phi}^{-1}(t_k, t_{k-1}) = \mathbf{\Phi}(t_{k-1}, t_k)$ has been introduced in the information filter. The information filter algorithm is summarised in Table 3-4.

Table 3-3 The Information Filter Algorithm

Step 1: Initialisation

$$\mathbf{Y}(t_0) = \mathbf{P}_0^{-1}; \ \hat{\mathbf{y}}(t_0) = \mathbf{Y}(t_0)\mathbf{x}_0$$

Step 2: Time Update

$$\hat{\mathbf{y}}(t_k^-) = [\mathbf{I} - \mathbf{\Pi}(t_k)\mathbf{G}^{\mathrm{T}}(t_{k-1})]\mathbf{\Phi}^{\mathrm{T}}(t_{k-1}, t_k)\hat{\mathbf{y}}(t_{k-1}^+)$$

$$\mathbf{Y}(t_k^-) = [\mathbf{I} - \mathbf{\Pi}(t_k)\mathbf{G}^{\mathrm{T}}(t_{k-1})]\mathbf{M}(t_k)$$

Step 3: Information Update

$$\mathbf{u}(t_k) = \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{z}(t_k)$$

$$\hat{\mathbf{y}}(t_k^+) = \hat{\mathbf{y}}(t_k^-) + \mathbf{u}(t_k)$$

$$\mathbf{U}(t_k) = \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{H}(t_k)$$

$$\mathbf{Y}(t_k^+) = \mathbf{Y}(t_k^-) + \mathbf{U}(t_k)$$

Step 4: State Recovery

$$\hat{\mathbf{x}}(t_k^+) = \mathbf{Y}^{-1}(t_k^+)\hat{\mathbf{y}}(t_k^+)$$

In summary, the information filter simplifies the information update procedure but increases the computation complexity of the time update procedure.

3.4 Statistical Hypothesis Test

A failure occurring in a multisensor navigation system can be seen as a sudden change in one or more of the system parameters, system outputs, or sensor outputs. Such changes can be classified as: additive failures or non-additive failures. Additive failures produce a change in the mean of the sensor measurements or the navigation state outputs. Non-additive failures result in changes in covariance of either the state estimate errors or the sensor measurement noise, or in the system parameters caused by the uncertainties of the system models. These failures may cause the performance of an aircraft navigation system to be degraded and can even lead to catastrophic events. In order to guarantee the safety of aircraft system, statistical hypothesis tests are used to check for anomalies in the mean of the sensor measurements and the navigation states or abnormalities of the covariances of the state estimate errors or measurement noise.

3.4.1 Hypothesis Test

A hypothesis is tested for possible rejection on the assumption that it is true. The concept was originally introduced by R. A. Fisher. In failure detection problems considered in this thesis, the determination of valid navigation states reduces to two competitive claims/hypotheses; the null hypothesis H_0 and the alternative hypothesis H_1. The null hypothesis H_0 represents a statement that no failures have happened in the aircraft multisensor navigation system while the alternative hypothesis H_1 is a statement that failures have occurred in the system. Based on these two hypotheses, a test statistic is calculated from sensor or system data and its value is used to decide which of these two hypotheses should be rejected in the hypothesis test. The choice of a test statistic depends on the assumed probability model and the hypotheses under question.

In a hypothesis test, two types of erroneous decisions may be made, known as type I error and type II error in statistical theory. A type I error occurs when the null hypothesis is rejected if it is in fact true; that is, H_1 is wrongly declared when H_0 is present. A type II error occurs when the null hypothesis is not rejected if it is in fact false, that is, H_0 is wrongly declared when H_1 is present.

In this thesis and for navigation systems in general, the type I error is known as a false alarm whereas the type II error is known as a missed alarm. The performance of a hypothesis test procedure is usually measured in terms of several probability values related to these two errors as follows.

The probability of a false alarm is denoted by P_{FA} and defined as follows:

$$P_{FA} = P(\text{Declare}\, H_1 \,|\, H_0 \text{ present}) = \int_{\Xi_1} p(\gamma \,|\, H_0) d\gamma \tag{3.48}$$

where γ represents a test statistic, $p(\gamma)$ is the probability density function of γ and Ξ_1 is called a rejection or critical region containing γ under the hypothesis H_1.

The probability of a missed alarm is denoted by P_{MA} and defined as:

$$P_{MA} = P(\text{Declare}\, H_0 \,|\, H_1 \text{ present}) = \int_{\Xi_0} p(\gamma \,|\, H_1) d\gamma \tag{3.49}$$

where Ξ_0 is an acceptance region containing γ under the hypothesis H_0. Moreover, $\Xi_0 \bigcup \Xi_1 = \Xi$ is the full set of observations of γ and $\Xi_0 \bigcap \Xi_1 = \text{O}$ is the null set.

The probability of detection is denoted by P_D and defined by

$$P_D = P(\text{Declare}\, H_1 \,|\, H_1 \text{ present}) = \int_{\Xi_1} p(\gamma \,|\, H_1) d\gamma \ = 1 - P_{MA} \tag{3.50}$$

Assume that a priori probabilities for each hypothesis H_0 and H_1 are $P_0(H_0)$ and $P_1(H_1)$, then the probability of the hypothesis test error is given by

$$P_E = P_0 P_{FA} + P_1 P_{MA} \tag{3.51}$$

A good detection algorithm is usually designed to minimise average probability of the hypothesis test error given by Eq. (3.51). Because P_0 is far larger than P_1, a false alarm is often considered to be more serious in the design of multisensor aircraft navigation systems, and therefore more important to take evasive action, than a missed alarm. Consequently, a hypothesis test procedure should be adjusted to obtain a guaranteed 'low' probability of false alarm. A confidence interval, that is, the probability that a test statistic will fall within a given critical region, is normally used to indicate a range of the uncertainty of test results and is expressed as a certain percentage. The concept of confidence interval is more informative than the simple results of a hypothesis test.

To find a range associated with a given confidence interval, a critical value is specified for a hypothesis test and is referred to as a threshold η so that the probability of a false alarm can be computed as

$$P_{FA}(\eta) = \alpha \tag{3.52}$$

where α is called a significance level and is expressed as a percentage. The confidence level is then given by $1-\alpha$. The threshold for a hypothesis test depends on the significance level at which the test is carried out.

The probability of a missed alarm error is usually unknown, but is symbolised by β and written as

$$P_{MA} = \beta \tag{3.53}$$

3.4.2 Bayesian Detection

Bayesian detection methods are based on the minimisation of error probability in a hypothesis test, as given in Eq. (3.51). Let C_{ij} $(i = 0,1; j = 0,1)$ represent the risks of declaring H_i true when H_j is present. The Bayesian decision rule minimises the following Bayesian risk function (*BRF*).

$$\begin{aligned} BRF &= \sum_{i=0}^{1}\sum_{j=0}^{1} C_{ij} P(H_j) P(\text{Declare } H_i \middle| H_j \text{ present}) \\ &= \sum_{i=0}^{1}\sum_{j=0}^{1} C_{ij} P(H_j) \int_{\Xi_i} p(\gamma \mid H_j) d\gamma \end{aligned} \tag{3.54}$$

Considering $\int_{\Xi} p(\gamma \mid H_i) d\gamma = 1,\ i = 0,1$, Eq. (3.54) can be simplified as

$$BRF = P_0 C_{10} + P_1 C_{11} + \int_{\Xi_0} \left([P_1 (C_{01} - C_{11}) p(\gamma \mid H_1)] - [P_0 (C_{10} - C_{00}) p(\gamma \mid H_0)] \right) d\gamma \tag{3.55}$$

Minimising BRF generates a likelihood ratio test (LRT) as follows[92]

$$\lambda(\gamma) = \frac{p(\gamma \mid H_1)}{p(\gamma \mid H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0 (C_{10} - C_{00})}{P_1 (C_{01} - C_{11})} = \eta \tag{3.56}$$

where $\lambda(\gamma)$ is called the LRT function and η is the threshold. Eq. (3.56) shows that the decision H_1 is made if $\lambda(\gamma)$ is larger than η, otherwise the decision H_0 is made. The log form of LRT function, as given in Eq. (3.57), is often used to simplify the computation of LRT.

$$\log[\lambda(\gamma)] \underset{H_0}{\overset{H_1}{\gtrless}} \log(\eta) \tag{3.57}$$

Eq. (3.56) or (3.57) is known as the Bayesian detection rule and is widely used in many failure detection problems where the a priori probabilities $P_0(H_0)$ and $P_1(H_1)$ are known.

3.4.3 Neyman-Pearson Detection

In some practical multisensor navigation systems, the a priori probabilities of two hypotheses may be unknown and the risk assignments are difficulty to estimate. These situations constrain the applications of Bayesian detection methods in sensor/ system failure detection problems. However, the constraints occurring in Bayesian detection do not take place in Neyman-Pearson detection methods. The objective of Neyman-Pearson detection is to maximise the probability of detection P_D for a given probability of false alarm P_{FA}. In other words, Newman-Pearson detection can obtain the minimum probability of a missed alarm P_{MA} under the condition that the acceptable value of P_{FA} is less α.

In failure detection problems, it is desirable to make both P_{FA} and P_{MA} as small as possible. Unfortunately, these are conflicting objectives. To obtain a tradeoff, a cost function F is constructed by using Eqs. (3.48), (3.49 and (3.50) as follows:

$$\begin{aligned} F &= P_{MA} + \eta(P_{FA} - \alpha) \\ &= \eta(1-\alpha) + \int_{\Xi_0} [p(\gamma \mid H_1) - \eta p(\gamma \mid H_0)] d\gamma \end{aligned} \tag{3.58}$$

where $\eta \geq 0$ is the Lagrange multiplier.

Minimising the cost function and employing the LRT leads to the following hypothesis test[92]

$$\lambda(\gamma)=\frac{p(\gamma|H_1)}{p(\gamma|H_0)}\underset{H_0}{\overset{H_1}{\gtrless}}\eta \tag{3.59}$$

Therefore, the threshold of this test is the Lagrange multiplier η, which is chosen to satisfy the given significance level α.

$$P_{FA}=\int_{\Xi_1}p(\gamma|H_0)d\gamma=\int_{\eta}^{\infty}p(\lambda|H_0)d\lambda=\alpha \tag{3.60}$$

Eq. (3.59) is referred to as the Newman-Pearson detection rule and is applied to many failure detection problems where the a priori probabilities $P_0(H_0)$ and $P_1(H_1)$ are difficulty to determine.

From the above analysis, it is important to determine the forms of probability density function $p(\gamma)$ under the two hypotheses, which are associated with the measure of the performance of a hypothesis test procedure. The probability density function is formulated on the basis of the statistical analysis of sensor measurement noise and residuals or the errors of the system state estimates.

3.5 Summary

This chapter has introduced mathematical fundamentals of the statistical estimation and hypothesis testing theories. The main activities covered include:

1. Introduction of three forms of Kalman filtering techniques and algorithms, including the conventional Kalman filter and the information filter.
2. Analysis of the statistical characteristics of Kalman filter estimation errors and residuals.
3. Introduction of statistical hypothesis test methods, including Bayesian detection and Newman-Pearson detection methods.

Chapter 4

NAVIGATION EQUATIONS AND ERROR DYNAMICS

4.1 Introduction

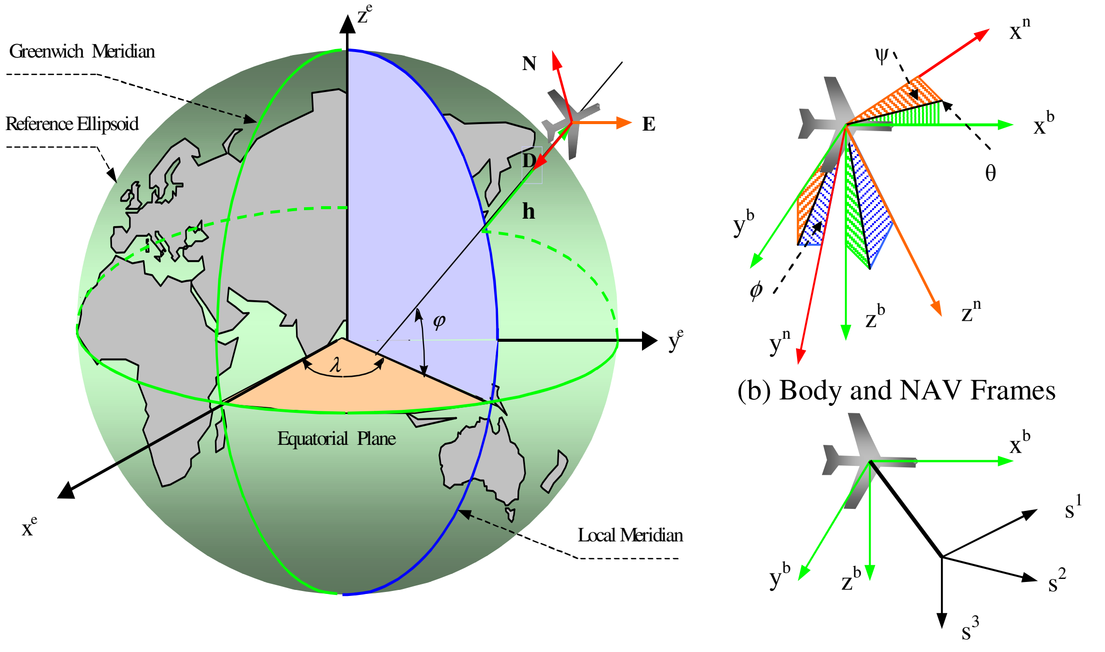
This chapter develops navigation equations and system error dynamic models of several aircraft navigation systems, including inertial and main navigation aiding systems. These equations and models constitute the mathematical foundations to design, develop and simulate fault-tolerant multisensor aircraft navigation systems.

Section 4.2 introduces various coordinate systems used in this thesis. Evolution of inertial sensor technologies and performance of different grade inertial sensors are highlighted in Section 4.3. In Section 4.4 inertial navigation equations are developed. Section 4.5 analyses the error dynamic models of inertial system. In Section 4.6 normalised navigation equation equations of major navaid systems are developed. A summary of this chapter is given in Section 4.7.

4.2 Coordinate Systems

Coordinate systems are established to develop the navigation equations and to describe the dynamic motion of an aircraft. Aircraft navigation systems resolve the navigation equations to determine position, velocity, attitude and time (PVAT) information with respect to specific frames. Several reference coordinate systems are used in the development and design of multisensor fusion navigation systems to represent the navigation system states, aircraft kinematic parameters and navigation sensor measurements. This section reviews the definitions of some commonly used reference frames and their relationships.

Earth Centred Inertial (ECI) Frame (x^I, y^I, z^I) This frame has its origin at the centre of the Earth. Its axes are non-rotating relative to the inertial space. For aircraft navigation, the ECI frame is an approximation of the Newtonian inertial frame. The x^I axis is in the Earth's equatorial plane and points toward the vernal equinox. The z^I axis is aligned with the Earth rotation axis. The y^I axis completes the right-hand system.



(b) Body and NAV Frames

(a) ECEF and NAV Frames

(c) Body and Instrument Frames

Figure 4.1 Coordinate Systems

Earth Centred Earth Fixed (ECEF) Frame (x^e, y^e, z^e) This frame is fixed to and rotating with the Earth. The ECEF frame has its origin at the Earth's centre of mass, as shown in Figure 4.1(a). This is a right-hand Cartesian coordinate system. The z^e axis is the Earth's rotation axis and points towards the direction of the Conventional Terrestrial Pole (CTP) for polar motion, as defined by BIH on the basis of the coordinates adopted for the BIH stations[93]. The x^e axis lies in the Earth's equatorial plane and points the intersection of the CTP's equator and the reference meridian being the zero meridian defined by the BIH on the basis of the coordinates adopted for the BIH stations. The y^{e} axis realises the right-hand system. The World Geodetic System of 1984 (WGS-84) is a commonly used ECEF frame that defines the Earth model and the reference ellipsoid. Geodetic parameters of the WGS-84 ellipsoid are summarised in Table A-1 of Appendix A. The position of aircraft above the surface of the Earth is defined in the ECEF frame by the geodetic coordinates.

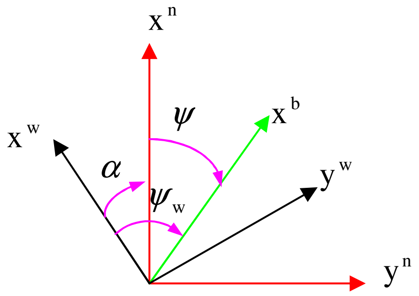
The WGS-84 coordinate system is used by global satellite navigation systems, for example, the Global Positioning System (GPS), to describe the satellite orbits. The coordinates of a point position in the WGS-84 frame can be expressed in the geodetic or Cartesian coordinates, and the transformation from geodetic coordinates to Cartesian coordinates is given by the following equations:

$$\begin{aligned}
\mathrm{x}^{\mathrm{e}} &= (\mathrm{N}+\mathrm{h}) * \cos(\varphi) * \cos(\lambda) \\
\mathrm{y}^{\mathrm{e}} &= (\mathrm{N}+\mathrm{h}) * \cos(\varphi)\sin(\lambda) \\
\mathrm{z}^{\mathrm{e}} &= [(1-e^2) * \mathrm{N}+\mathrm{h}] * \sin(\varphi)
\end{aligned} \tag{4.1}$$

where $(\mathrm{x}^{\mathrm{e}}, \mathrm{y}^{\mathrm{e}}, \mathrm{z}^{\mathrm{e}})$ are the Cartesian coordinates of a point position, (φ, λ, h) are the geodetic coordinates of the point position (latitude, longitude and height above the reference ellipsoid of the Earth), and $N = \dfrac{R_{\mathrm{a}}}{\sqrt{1-e^2 * \sin^2(\varphi)}}$ is the radius of curvature in the prime vertical.

Navigation Frame $(\mathrm{x}^{\mathrm{n}}, \mathrm{y}^{\mathrm{n}}, \mathrm{z}^{\mathrm{n}})$ This frame is attached to the aircraft and has its origin at the aircraft centre of gravity, as shown in Figure 4.1(a)(b). The z^{n} axis points down perpendicular to the reference ellipsoid. The x^{n} and y^{n} axes lie in a plane tangent to the reference ellipsoid. Therefore, the navigation frame is a local level frame and a Cartesian coordinate system.

This navigation frame is a north-slaved frame if the x^{e} axis points north and the y^{e} axis points east, and is generally referred to as a North-East-Down (NED) frame. It is known as the wander azimuth frame $(\mathrm{x}^{\mathrm{w}}, \mathrm{y}^{\mathrm{w}}, \mathrm{z}^{\mathrm{w}})$ if the x^{n}-y^{n} plane is allowed to rotate freely about the z^{n} axis. The wander azimuth mechanisation permits the operation of an inertial system at latitudes close to the Polar Regions, avoiding the singularity associated with the north-slaved mechanisation. In this thesis, inertial navigation equations are developed in the wander azimuth frame. The relationship between the wander azimuth frame and the NED frame is illustrated in Figure 4.2 and is formulated in Eq.(4.2), where α is the wander azimuth angle, ψ is the heading angle and ψ_{w} is the heading angle of the wander frame.



$$\mathbf{C}_{\mathrm{w}}^{\mathrm{n}} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

Figure 4.2 Wander Azimuth Frame

The location of the origin of the navigation frame is specified relative to the ECEF frame by the geodetic coordinates (λ, φ, h). Aircraft velocity and attitude angles are defined with respect to the navigation frame. The transformation matrix between the NED and ECEF frames is known as the position direction cosine matrix (DCM) and can be obtained through a series of rotation transformations as follows:

$$\mathbf{C}_{\mathrm{e}}^{\mathrm{n}} = \begin{bmatrix} -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\varphi)\cos(\lambda) & -\cos(\varphi)\sin(\lambda) & -\sin(\varphi) \end{bmatrix} \tag{4.3}$$

Let $\mathbf{C}_{\mathrm{e}}^{\mathrm{w}}$ be the transformation matrix from the ECEF frame to the wander-azimuth frame, then

$$\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} = \mathbf{C}_{\mathrm{n}}^{\mathrm{w}}\mathbf{C}_{\mathrm{e}}^{\mathrm{n}} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\varphi)\cos(\lambda) & -\cos(\varphi)\sin(\lambda) & -\sin(\varphi) \end{bmatrix}$$

$$\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} = \begin{bmatrix} \begin{array}{l} -\cos(\alpha)\sin(\varphi)\cos(\lambda) \\ +\sin(\alpha)\sin(\lambda) \end{array} & \begin{array}{l} -\cos(\alpha)\sin(\varphi)\sin(\lambda) \\ -\sin(\alpha)\cos(\lambda) \end{array} & \cos(\alpha)\cos(\varphi) \\ \begin{array}{l} -\sin(\alpha)\sin(\varphi)\cos(\lambda) \\ -\cos(\alpha)\sin(\lambda) \end{array} & \begin{array}{l} -\sin(\alpha)\sin(\varphi)\sin(\lambda) \\ +\cos(\alpha)\cos(\lambda) \end{array} & \sin(\alpha)\cos(\varphi) \\ -\cos(\varphi)\cos(\lambda) & -\cos(\varphi)\sin(\lambda) & -\sin(\varphi) \end{bmatrix} \tag{4.4}$$

Body Frame (x^b, y^b, z^b) This frame is fixed to the aircraft and has its origin at the aircraft centre of gravity or mass, as shown in Figure 4.1(b). The x^b and z^b axes are in the plane of symmetry of the aircraft, where x^b points toward the nose of the aircraft and z^b axis points downward. The y^b axis points down the starboard wing. The body frame is a Cartesian coordinate system.

The orientation of the body frame relative to the navigation frame is specified by the Euler angles (roll ϕ, pitch θ and yaw ψ), as shown in Figure 4.1(b). The transformation matrix between the body frame and the navigation frame is referred to as the attitude direction cosine matrix. The transformation from the body frame to the wander frame is formulated by a series of rotation transformations as follows:

$$\mathbf{C}_b^w = \begin{bmatrix} \cos(\theta)\cos(\psi_w) & \begin{matrix}\sin(\phi)\sin(\theta)\cos(\psi_w) \\ -\cos(\phi)\sin(\psi_w)\end{matrix} & \begin{matrix}\cos(\phi)\sin(\theta)\cos(\psi_w) \\ +\sin(\phi)\sin(\psi_w)\end{matrix} \\ \cos(\theta)\sin(\psi_w) & \begin{matrix}\sin(\phi)\sin(\theta)\sin(\psi_w) \\ +\cos(\phi)\cos(\psi_w)\end{matrix} & \begin{matrix}\cos(\phi)\sin(\theta)\sin(\psi_w) \\ -\sin(\phi)\cos(\psi_w)\end{matrix} \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{4.5}$$

The body frame is used to develop the aircraft equations of motion and for the attitude control and determination. Aircraft kinematic parameters are defined with reference to this frame. For example, the acceleration and angular rate information required by a flight control system or autopilot are normally represented in terms of the body frame.

Instrument Frame $(s^1, s^2, s^3, ..., s^n)$ This frame specifies the orientations of the "sensing axes" of a sensor system with respect to a reference frame (for example, the body or the navigation frame) and has its origin at the installation location of sensor system, as illustrated in Figure 4.1(c). For example, in inertial sensor systems, the sensing axes are the installation axes of the inertial sensors. If an inertial instrument frame is a Cartesian coordinate system, this inertial sensor system is an orthogonal configuration. In Doppler radar systems, the instrument frame is the radar antenna frame that defines the orientations of radar beams relative to a reference frame. In a radio navigation system, the 'sensing axes' are usually the directions of lines of sight (LOS) from receiver antenna to transmitters.

The transformation matrix from an instrument frame to a reference frame (the body or navigation frame) is usually known as the design or measurement matrix of the navigational sensor system. Sensor measurements are represented in terms of the instrument coordinates.

4.3 Inertial Sensor Technology

4.3.1 Inertial Sensor System

Inertial sensors are classified as gyroscopes and accelerometers. Gyroscopes are angular rate sensors while accelerometers are specific force sensors. The specific force is a combination of the gravitational forces or their projections and total inertial force acting on aircraft. Gyroscopes and accelerometers can be integrated into a case to form an inertial reference system (IRS), also known as an inertial measurement unit (IMU). An inertial reference system can measure all the kinematic parameters of an aircraft. Integration of an IMU system and a computer solving inertial navigation algorithms makes up an inertial navigation system (INS). In an INS, gyroscopes are used to maintain a stable reference platform or establish it by analytical means. The measurements from the accelerometers can be referenced to this reference frame for computation of the navigation states. Misalignments of the reference frame caused by gyro drifts couple the measured accelerations along each axis so that the distance error is the time-cubical dependence[94]. Consequently, gyroscope performance plays a critical role in the improvement of the accuracy of inertial navigation system and development of inertial sensor technology has focused on gyroscope technology.

Inertial sensor systems, depending on the numbers of sensors and installations, can be classified as orthogonal or non-orthogonal configurations. Non-orthogonal configurations will be discussed in Chapter 5. For development of inertial navigation algorithms, it is assumed here that three accelerometers and three gyros are mounted in orthogonal triads and their input axes are aligned with the axes of the body frame. In this case, the IMU outputs are coordinated in the body frame. Strapdown inertial navigation algorithms will be developed in Section 4.4.

Inertial sensor systems, according to the accuracy requirements for different applications, can categorised as from low quality though tactical to navigation grade (high quality) shown in Table A-2 of Appendix A. The navigation-grade IMU can be alone used to implement an inertial navigation system with level of allowable errors for one-hour flight. The control grade IMU is generally used to provide inertial measurements for control systems. The tactical grade IMU can be applied for the attitude display and flight control and short-time navigation and guidance. When inertial sensor systems are combined with aiding navigation systems, for example, GNSS, the accuracy requirements for inertial sensors can be further relaxed from the navigational to low grades. Therefore, many emerging inertial sensor technologies can be used for the development of aircraft multisensor navigation systems[95][96].

4.3.2 Inertial Sensor Performance

For navigation applications and the development of the inertial system error model in this thesis, the performance of inertial sensors are dominantly characterised by the following parameters.

An inertial sensor provides an output signal in response to its input, either rotation or acceleration of aircraft. The scale factor is a transform factor that defines the ratio of the output signal to the input signal. Ideally, the scale factor is a constant and there exists a linear relationship between the sensor output and input. Owing to imperfections of the manufacturing and signal processing process, the linearity of scale factor may change over different input ranges. As a result, the scale factor may give different values for different input ranges or may have second or higher-order terms relating the output signal to input. This leads to the nonlinearity or instability of scale factor. Furthermore, a sensor may have a different scale factor for positive and negative inputs, known as scale factor asymmetry.

Sensor bias is an offset of an inertial sensor when an output is detected for no input or input signal change. This bias may be different for positive and negative inputs and may be turn-on dependent. The uncertainty or instability of sensor bias is an important parameter in assessing the accuracy of sensor measurements. For a gyroscope, the sensor output error caused by nonlinearity, instability or asymmetry of scale factor and sensor bias is known as the gyro drift.

When inertial sensors are integrated in a case to form an inertial system, the input axis of each sensor should be aligned to its associated case reference axis. The angle between the input axis and its associated axis is defined as the input-axis misalignment.

These performance parameters are common for all kinds of inertial sensors. But, because of diverse design principles and manufacturing procedures, individual sensors may have specific error sources. The main performance parameters of inertial sensors for aircraft navigation applications are summarised in Table A-2.

4.3.3 Gyroscope Technology

The evolution of gyroscope technologies covers three generations. The first generation is the traditional rigid rotor gyroscopes, which are distinguished as two classes: attitude gyros and rate gyros. Attitude gyros were based on the principle of conservation of angular momentum and were specially adapted to the stable platform systems. Rate gyros are principally based on the Newton's second law and were used for strapdown navigation systems. Traditional rotor gyros have the highest accuracy but expensive cost and large volume. Physical implementations of various rotor gyroscopes can be mainly featured by the methods in which the maintenance of reference angular momentum of the rotor is achieved. This ranges from simple inexpensive flywheel design to highly accurate and complex design, for example, the floated integrating gyro and the electro-statically suspended gyro. Rotor gyros are the most mature in the development of gyroscope technology. The traditional rotor gyros are continually used in marine navigation applications, but have been replaced by optical gyroscopes for aircraft navigation applications.

The second generation is the optical gyroscopes, which are based on the Sagnac effect[94][97]. The Sagnac effect is an optical phenomenon of the relativistic effect. In an optical gyroscope, two laser beams from the same laser source propagate around a closed path in opposite directions. If this closed path is rotating around its rotation axis, these two beams will arrive at a detector at slightly different times because the optical path travelled by one beam along the direction of rotation becomes longer than the path travelled by the other beam. The angular rate of this closed path rotation can be measured by detecting the time difference between the two paths. Optical gyroscopes can be classified as two kinds of classes: the ring laser gyroscope (RLG) and the fibre optic gyroscope (FOG). Optical gyroscopes were developed for strapdown systems and have replaced the mechanical gyros in many applications. Optical gyroscopes have smaller volume, lower cost and wider dynamic rate range in comparison with mechanical gyros.

In the RLG, two counter-propagating laser beams travel through a laser cavity with reflecting mirrors (resonant cavity). When the resonant cavity is rotating around its sensitive axis, these two waves resonate in the rotating cavity to generate the frequency shift, which is proportional to the angular rate of laser cavity rotation. A photodiode detector can detect this frequency shift in the form of interfering fringes to derive the angular rate. Current RLG sensors have reached to the performance of traditional rotor gyroscopes[97]. However, in order to attain such high measurement accuracy, a RLG needs a large volume to increase the length of the optical cavity. In addition, the RLG sensors are expensive.

In the FOG, two counter-propagating lasers travel along a closed-loop optical fibre. When FOG is rotating around its sensitive axis, these two counter-propagating waves interfere with each other to induce the Sagnac phase (or frequency) shift that can be measured by a photo-detector to obtain the angular rate of FOG rotation. The FOG sensor has some desirable features, such as reduced weight. It is smaller than the RLG and significantly cheaper. However, FOG has lower sensitivity and current FOG technology cannot reach the performance of a RLG sensor. Sensors based on an interferometric FOG and a resonant FOG sensors have been developed for tactical, AHRS and aided navigation applications. Recent developments in optical gyroscope are concerned with integrated optic gyro (IOG), which is insensitive to environment effects and is relatively inexpensive.

The third generation of gyroscope involves MicroElectroMechanical Systems (MEMS) inertial sensors. All MEMS-based gyroscopes make use of the Coriolis principle[98]. MEMS inertial sensors have the smallest volume and cheapest cost but at present, their accuracy is less than optical gyros. MEMS gyros have been applied in the automobile industries and for guided munitions. With further development and maturation of MEMS sensor technologies, it is expected that MEMS gyros will achieve a performance of better than 1 deg/hr and will replace some optical gyros in many aerospace applications. For example, the current test results from the Charles Stark Draper Laboratory[99] have shown that the bias stability of MEMS gyro over small temperature ranges of 0.5°C has surpassed 10 deg/hr in tests lasting six hours, while the companion accelerometer demonstrates submilli-g performance.

4.3.4 Accelerometer Technology

The development of accelerometer technology can be classified by two types. The first type uses the principal of the force rebalance. For both translational proof-mass and pendulous proof-mass accelerometers, the displacement of the proof-mass resulting from external force or acceleration is measured by a detector and the position of the mass is restored by closed-loop control. This displacement is a direct measure of the acceleration. This type of accelerometer has been used in most inertial navigation systems.

The second type is based on the vibratory accelerometer, which senses acceleration by detecting transverse resonant frequency of a pendulous proof-mass. There are several different versions, including the vibrating string accelerometer, the vibrating beam accelerometer, the quartz resonator accelerometer and the integrated silicon accelerometers. This type of accelerometer has been used in aided inertial navigation systems but currently, cannot reach to the performance of the first type for inertial navigation systems. However, they offer the advantage of direct digital output, they consume relatively little power and they are more rugged.

MEMS accelerometers have been based on both the force rebalance and the quartz resonator principle and several MEMS accelerometers are currently used in aided inertial navigation systems. The performance and trends of MEMS inertial sensors are briefly summarised in Table A-3 of Appendix A.

4.4 Inertial Navigation Equations

This section develops the set of differential equations defining the navigation states, which are expressed in terms of the sensed accelerations and angular rates available from an IMU. The principal of inertial navigation is based on Newton's second law of motion, which is valid in an inertial frame. Aircraft navigation however occurs in a terrestrial navigation frame. Therefore, the navigation states have to be referenced to the local geodetic coordinates and the navigation frames. In this thesis, the wander azimuth frame is used to represent the navigation frame for the reasons outlined in Section 4.1.

4.4.1 Velocity Equations

The velocity differential equations are derived on the basis of the Coriolis theorem. The physical interpretation of the Coriolis theorem is that the rate of change of a vector takes a different quantity when observed in two relative moving reference frames. In vector operator notation, the Coriolis theorem is written as[100]

$$\mathbf{u}^{\mathrm{A}} = \mathbf{C}_{\mathrm{B}}^{\mathrm{A}}[\mathbf{u}^{\mathrm{B}} + (\boldsymbol{\omega}_{\mathrm{A/B}}\times)\mathbf{u}] \tag{4.6}$$

where $\mathbf{u}$ is an arbitrary vector, A and B are two relative moving reference frames and $\boldsymbol{\omega}_{\mathrm{A/B}}$ represents angular rate vector of rotation of B relative to A.

By applying the above relative motion equation to Newton's second law, the velocity equation in the wander frame can be obtained.

When an aircraft flies around the Earth, rotating again around the ECI frame, the aircraft velocity in the wander frame $\mathbf{v}^{\mathrm{w}}$ is defined in terms of the aircraft position $\mathbf{r}^{\mathrm{e}}$ in the rotating ECEF frame as follows:

$$\mathbf{v}^{\mathrm{w}} = \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}\dot{\mathbf{r}}^{\mathrm{e}} \tag{4.7}$$

Furthermore, the aircraft position $\mathbf{r}^{\mathrm{e}}$ in the ECEF frame is represented in terms of its corresponding position $\mathbf{r}^{\mathrm{I}}$ in the ECI frame as

$$\mathbf{r}^{\mathrm{e}} = \mathbf{C}_{\mathrm{I}}^{\mathrm{e}}\mathbf{r}^{\mathrm{I}} \tag{4.8}$$

where $\mathbf{C}_{\mathrm{I}}^{\mathrm{e}}$ is the rotation transformation matrix from the ECI to ECEF frames.

The time derivatives of Eqs.(4.7) and (4.8) lead to

$$\dot{\mathbf{v}}^{w} = \dot{\mathbf{C}}_{e}^{w}\dot{\mathbf{r}}^{e} + \mathbf{C}_{e}^{w}\ddot{\mathbf{r}}^{e} = -(\boldsymbol{\omega}_{e/w}^{w}\times)\mathbf{C}_{e}^{w}\dot{\mathbf{r}}^{e} + \mathbf{C}_{e}^{w}\ddot{\mathbf{r}}^{e} = -(\boldsymbol{\omega}_{e/w}^{w}\times)\mathbf{v}^{w} + \mathbf{C}_{e}^{w}\ddot{\mathbf{r}}^{e} \tag{4.9}$$

$$\dot{\mathbf{r}}^{e} = \dot{\mathbf{C}}_{I}^{e}\mathbf{r}^{I} + \mathbf{C}_{I}^{e}\dot{\mathbf{r}}^{I} = -\mathbf{C}_{I}^{e}(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I} + \mathbf{C}_{I}^{e}\dot{\mathbf{r}}^{I} \tag{4.10}$$

where $\boldsymbol{\omega}_{e/w}^{w}$ is the angular rate vector of rotation of the wander frame relative to the ECEF frame in terms of the wander coordinates and $\boldsymbol{\omega}_{I/e}^{I}$ is the Earth's rotation rate vector in the ECI coordinates. For aircraft navigation, $\boldsymbol{\omega}_{I/e}^{I}$ is assumed to be a constant.

Again the time derivative of Eq. (4.10) is

$$\ddot{\mathbf{r}}^{e} = \mathbf{C}_{I}^{e}(\boldsymbol{\omega}_{I/e}^{I}\times)(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I} - 2(\boldsymbol{\omega}_{I/e}^{e}\times)\mathbf{C}_{I}^{e}\dot{\mathbf{r}}^{I} + \mathbf{C}_{I}^{e}\ddot{\mathbf{r}}^{I} \tag{4.11}$$

Substituting Eqs. (4.7) and (4.10) into Eq. (4.11) results in

$$\ddot{\mathbf{r}}^{e} = -\mathbf{C}_{I}^{e}(\boldsymbol{\omega}_{I/e}^{I}\times)(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I} - 2\boldsymbol{\omega}_{I/e}^{e}\times\mathbf{C}_{w}^{e}\mathbf{v}^{w} + \mathbf{C}_{I}^{e}\ddot{\mathbf{r}}^{I} \tag{4.12}$$

Substituting Eq. (4.12) into Eq. (4.9) leads to

$$\dot{\mathbf{v}}^{w} = -[(\boldsymbol{\omega}_{e/w}^{w}\times) + 2(\boldsymbol{\omega}_{I/e}^{w}\times)]\mathbf{v}^{w} + \mathbf{C}_{I}^{w}[\ddot{\mathbf{r}}^{I} - (\boldsymbol{\omega}_{I/e}^{I}\times)(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I}] \tag{4.13}$$

where $(\boldsymbol{\omega}_{I/e}^{I}\times)(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I}$ is the centripetal acceleration caused by the Earth's rotation and $\ddot{\mathbf{r}}^{I}$ is the inertial acceleration of aircraft.

However, accelerometer does not directly measure the acceleration $\ddot{\mathbf{r}}^{I}$ rather than the specific force $\mathbf{f}^{b}$ coordinated in the body frame in a strapdown system. This specific force is a combination of both the inertial and gravitational accelerations:

$$\mathbf{C}_{b}^{I}\mathbf{f}^{b} = \ddot{\mathbf{r}}^{I} - \mathbf{G}^{I} \tag{4.14}$$

The total gravitational acceleration includes the local gravity component $\mathbf{g}^{I}$ and the centripetal acceleration:

$$\mathbf{G}^{I} = \mathbf{g}^{I} + (\boldsymbol{\omega}_{I/e}^{I}\times)(\boldsymbol{\omega}_{I/e}^{I}\times)\mathbf{r}^{I} \tag{4.15}$$

Substituting Eqs. (4.14) and (4.15) into Eq.(4.13) produces

$$\dot{\mathbf{v}}^{w} = -[(\boldsymbol{\omega}_{e/w}^{w}\times) + 2(\boldsymbol{\omega}_{I/e}^{w}\times)]\mathbf{v}^{w} + \mathbf{C}_{I}^{w}[\mathbf{C}_{b}^{I}\mathbf{f}^{b} + \mathbf{g}^{I}]$$

Therefore, the velocity equation in the wander frame is given as follows:

$$\dot{\mathbf{v}}^{w} = \mathbf{C}_{b}^{w}\mathbf{f}^{b} - [(\boldsymbol{\omega}_{e/w}^{w}\times) + 2(\boldsymbol{\omega}_{i/e}^{w}\times)]\mathbf{v}^{w} + \mathbf{g}^{w} \tag{4.16}$$

where $\mathbf{C}_{b}^{w}\mathbf{f}^{b} \equiv \mathbf{f}^{w}$ is the measured specific force vector coordinated in the wander frame, $\boldsymbol{\omega}\times$ is the skew-symmetric matrix of the vector $\boldsymbol{\omega}$, $\mathbf{v}^{w}$ is the velocity vector of aircraft in terms of the wander coordinates, $\boldsymbol{\omega}_{e/w}^{w}$ is the transport rate vector of aircraft, $\boldsymbol{\omega}_{I/e}^{w}$ is the earth rate vector represented in the wander coordinates and $\mathbf{g}^{w}$ is the gravity vector. The expressions of $\mathbf{g}^{w}$ and $\boldsymbol{\omega}_{e/w}^{w}$ are given in Appendix B.

4.4.2 Attitude Equations

In order to derive the velocity in Eq. (4.16), the attitude DCM $\mathbf{C}_{b}^{w}$ must first be determined so that the sensed specific force vector from an IMU can be referenced to the wander frame. From the transport equation given in Eq. (4.6), the differential equation of the attitude DCM $\mathbf{C}_{b}^{w}$ can be derived as follows[101]

$$\dot{\mathbf{C}}_{b}^{w} = -(\boldsymbol{\omega}_{b/w}^{w}\times)\mathbf{C}_{b}^{w} \tag{4.17}$$

where $\boldsymbol{\omega}_{b/w}^{w}$ is the angular rate vector of rotation of the wander frame relative to the body frame, coordinated in the wander azimuth frame. From the addition of angular velocities, $\boldsymbol{\omega}_{b/w}^{w}$ can be decomposed as follows:

$$(\boldsymbol{\omega}_{b/w}^{w}\times) = (\boldsymbol{\omega}_{I/w}^{w}\times) - \mathbf{C}_{b}^{w}(\boldsymbol{\omega}_{I/b}^{b}\times)\mathbf{C}_{w}^{b} = (\boldsymbol{\omega}_{I/e}^{w}\times) + (\boldsymbol{\omega}_{e/w}^{w}\times) - \mathbf{C}_{b}^{w}(\boldsymbol{\omega}_{I/b}^{b}\times)\mathbf{C}_{w}^{b} \tag{4.18}$$

Substituting Eq. (4.18) into Eq. (4.17) leads to

$$\dot{\mathbf{C}}_{b}^{w} = \mathbf{C}_{b}^{w}(\boldsymbol{\omega}_{I/b}^{b}\times) - [(\boldsymbol{\omega}_{I/e}^{w}\times) + (\boldsymbol{\omega}_{e/w}^{w}\times)]\mathbf{C}_{b}^{w} \tag{4.19}$$

where $\boldsymbol{\omega}_{I/b}^{b}$ is the measured body angular rate vector from a strapdown IMU.

Eq. (4.19) is known as the attitude matrix differential equation. The attitude DCM establishes an analytical platform. From Eq. (4.16), the measurements from the accelerometers must be resolved in this analytical platform in order to derive the navigation states. Because $\mathbf{C}_{b}^{w}$ is a symmetric orthogonal matrix, at least six first-order differential equations in Eq. (4.19) must be resolved to obtain the attitude DCM. To simplify the computation of the attitude DCM differential equation, a quaternion form of the attitude matrix differential equation is commonly used. The quaternion differential equation and the relationship between the quaternion and the attitude DCM elements are given in Appendix C.

4.4.3 Position Equations

The differential equation of the position DCM can also be derived from the transport theorem as follows:

$$\dot{\mathbf{C}}_{e}^{w} = -(\boldsymbol{\omega}_{e/w}^{w}\times)\mathbf{C}_{e}^{w} \tag{4.20}$$

$\boldsymbol{\omega}_{e/w}^{w}$ is the transport rate of an aircraft and is given by Eq.(4.21). From an initial position DCM or position, Eq.(4.20) can be integrated to give the current position DCM. From Eq. (4.4), the geographic location of aircraft and the wander angle can be computed as follows:

$$\varphi = \sin^{-1}[-\mathbf{C}_{e}^{w}(3,3)],\ \lambda = \tan^{-1}\left[\frac{-\mathbf{C}_{e}^{w}(3,2)}{-\mathbf{C}_{e}^{w}(3,1)}\right],\ \alpha = \tan^{-1}\left[\frac{\mathbf{C}_{e}^{w}(1,3)}{\mathbf{C}_{e}^{w}(2,3)}\right] \tag{4.21}$$

The altitude of aircraft above the surface of the WGS-84 ellipsoid is obtained by integrating the vertical component of aircraft velocity v_z as follows:

$$\mathrm{h} = -\dot{\mathrm{v}}_{z} \tag{4.22}$$

Therefore, Eqs. (4.16), (2.19) or (C.3) and (4.20) constitute the navigation algorithms of strapdown inertial navigation systems.

The architecture of the strapdown inertial navigation algorithms is illustrated in Figure 4.3.

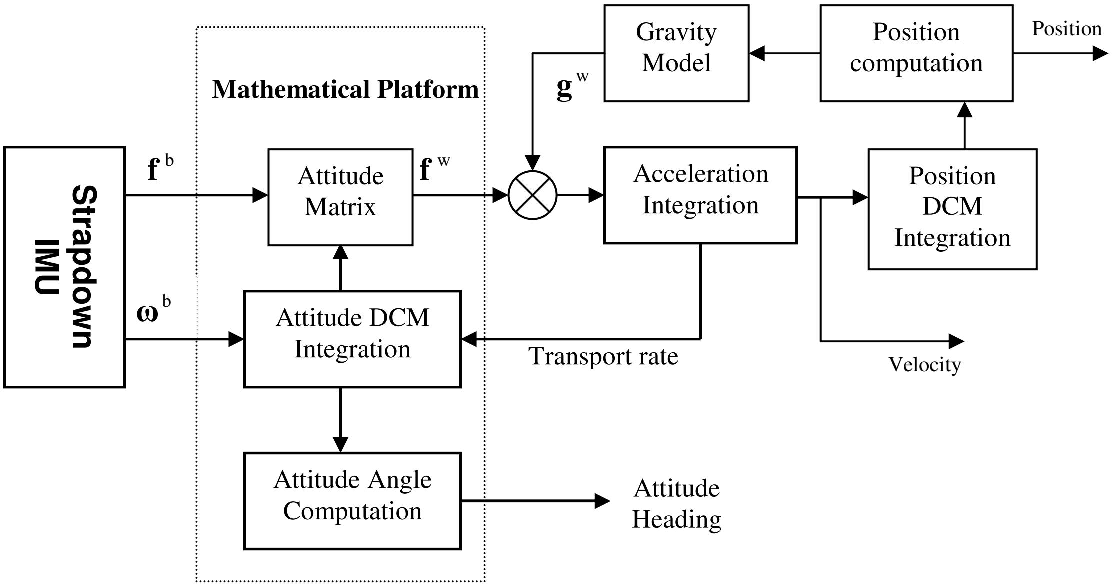


Figure 4.3 Modular Navigation Algorithm Architecture

4.4.4 Disadvantages of Inertial Navigation Systems

From the previous subsections, in order to integrate the velocity, the attitude DCM and the position DCM differential equations, initial values of the navigation states must be known accurately. Even though a high quality IMU can determine the initial attitude DCM by performing a so-called initial alignment procedure, initial velocity and position values have to be provided by other navigation aiding means. In addition, the integrating procedure will accumulate the navigation state errors caused by various sensor error sources. Accordingly, inertial navigation systems are generally aided by other aiding navigation systems.

Initial alignment is a static ground alignment procedure and comprises two steps: coarse alignment and fine alignment. The coarse alignment makes use of the known properties of the Earth's gravity and rotation at specific geographic locations to estimate the initial attitudes.

Assume that an aircraft is at a known location, the Earth's gravity and rotation at this known location point can be accurately computed in the wander frame as follows:

$$\mathbf{g}^{w} = \mathbf{C}_{e}^{w}\mathbf{g}^{e} \tag{4.23}$$

$$\boldsymbol{\omega}_{I/e}^{w} = \mathbf{C}_{e}^{w}\boldsymbol{\omega}_{I/e}^{e} \tag{4.24}$$

In this case, outputs from a strapdown IMU can be formulated as follows:

$$\tilde{\mathbf{f}}^{b} = \mathbf{f}^{b} + \nabla^{b} = \mathbf{C}_{w}^{b}\mathbf{g}^{w} \tag{4.25}$$

$$\tilde{\boldsymbol{\omega}}_{I/e}^{b} = \boldsymbol{\omega}_{I/e}^{b} + \Delta^{b} = \mathbf{C}_{w}^{b}\boldsymbol{\omega}_{I/e}^{w} \tag{4.26}$$

where ∇^{b} is the total accelerometer measurement error vector and Δ^{b} is the total gyro measurement error vector.

Combining Eqs (4.25) and (4.26) gives

$$\left[\tilde{\mathbf{f}}^{b} \quad \tilde{\boldsymbol{\omega}}_{I/e}^{b} \quad \tilde{\mathbf{f}}^{b} \times \tilde{\boldsymbol{\omega}}_{I/e}^{b}\right] = \mathbf{C}_{w}^{b}\left[\mathbf{g}^{w} \quad \boldsymbol{\omega}_{I/e}^{w} \quad \mathbf{g}^{w} \times \boldsymbol{\omega}_{I/e}^{w}\right] \tag{4.27a}$$

or $$\left[\tilde{\mathbf{f}}^{b} \quad \tilde{\boldsymbol{\omega}}_{I/e}^{b} \quad \tilde{\mathbf{f}}^{b} \times \tilde{\boldsymbol{\omega}}_{I/e}^{b}\right]^{T} = \left[\mathbf{g}^{w} \quad \boldsymbol{\omega}_{I/e}^{w} \quad \mathbf{g}^{w} \times \boldsymbol{\omega}_{I/e}^{w}\right]^{T}\mathbf{C}_{b}^{w} \tag{4.27b}$$

Eqs. (4.27a) and (4.27b) are known as the coarse alignment equation from which initial attitude DCM can be determined. The uncertainty of the coarse alignment is approximated as

$$\delta\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \approx \left[\mathbf{g}^{\mathrm{w}} \quad \boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{w}} \quad \mathbf{g}^{\mathrm{w}} \times \boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{w}}\right]^{-\mathrm{T}} \left[\nabla^{\mathrm{b}} \quad \Delta^{\mathrm{b}} \quad \tilde{\mathbf{f}}^{\mathrm{b}} \times \Delta^{\mathrm{b}} + \nabla^{\mathrm{b}} \times \boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{b}}\right]^{\mathrm{T}} \qquad (4.28)$$

From Eq.(4.28), the accuracy of initial alignment totally depends on the performance of inertial sensors. A low quality IMU cannot obtain expected alignment accuracy.

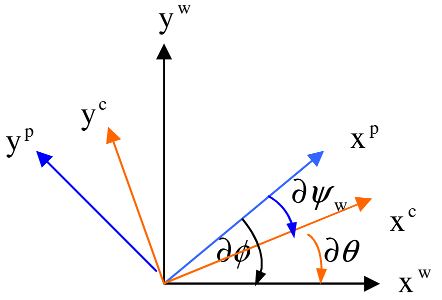
The fine alignment is based the fact that aircraft's velocity with respect to the ground is zero at rest. The fine alignment usually uses a Kalman filter to refine the attitude DCM estimated by the coarse alignment. The observable for a fine alignment filter includes the velocity and the Earth rotation.

However, the initial alignment procedure cannot be used for the alignment or correction of an in-flight INS. In order to correct the INS-driven navigation states and calibrate IMU sensor errors in-flight, inertial systems are usually combined with other aiding navigation systems using data fusion techniques. In the following section, dynamic error models of strapdown inertial navigation system will be established. The error models are used to analyse the initial alignment accuracy, and to design data fusion filters and fine alignment filters.

4.5 Error Analysis of Inertial Navigation System

Development of strapdown inertial navigation algorithms is based on detailed error analysis, which is a critical aspect in the design and development of various multisensor data fusion navigation systems. Error analysis is not only used to assess the accuracy of aircraft navigation systems and verify the performance of required inertial sensors, but is also used to determine the design requirements for integrated navigation filters and measurement requirements for aiding navigation systems in a multisensor navigation system. Error analysis is based on the derivation of error dynamic models of the navigation states. Dynamic models of the navigation state errors provide the mathematical foundation for navigation system failure detection and isolation, the implementation of an integrated navigation filter in a multisensor navigation system and the initial alignment and dynamic calibration of inertial systems.

Two basic methods have been suggested in the literature to derive error models of the navigation states for inertial navigation system: the Phi-angle error model (also referred to as the tilt errors) and the Psi-angle error model (also known as the attitude errors). Both models make use of a perturbation approach but the perturbations are performed with reference to different reference frames. The Phi-angle error model is derived from a linear perturbation of the navigation equations with respect to the true navigation frame while the Psi-angle error model is from a linear perturbation of the navigation equations in the **computer frame**. The computer frame is the navigation frame retained by the navigation system and has its origin at the computed position. The geometric relationships between the true navigation frame, the platform frame and the computer frame are illustrated in Figure 4.4, where the **platform frame** is an imaginary mathematical platform in a strapdown system and is determined by the computed DCM from the body frame to the estimate of the wander azimuth frame.



$$\partial\phi = \partial\boldsymbol{\psi}_w + \partial\boldsymbol{\theta} \tag{4.29}$$

Figure 4.4 Relation Between Three Frames

Benson[102] proves the equivalence of the two error models using analytical methods and simulation. Goshen-Meskin and Bar-Itzhack[103] extend these methods and present a unified approach to the development of inertial navigation system error models. Scherzinger and Reid[104] further introduce modified error models, which is based on the computed velocity instead of the measured specific forces. However, this replacement may introduce large uncertainty into the error models' parameters because the computed velocity contains accumulated sensor errors. In this thesis, the error dynamic models are developed on the basis of perturbation with respect to the true wander navigation frame. However, the velocity error is selected to simplify the formulation of the error model. The representation of the Phi-angle errors has certain advantages for control of the navigation state errors because the estimated errors of the navigation states are directly related to the true navigation frame.

4.5.1 Velocity Error Equations

In order to deduce the velocity error differential equation, assume that the angular position error $\partial\theta$, caused by uncertainty of the computed position DCM $\tilde{\mathbf{C}}_{e}^{w}$, and the tilt error $\partial\phi$, caused by uncertainty of the computed attitude DCM $\tilde{\mathbf{C}}_{b}^{w}$, are known. The velocity obtained by integrating Eq. (4.16) can be represented in terms of the true velocity plus the velocity error term as follows:

$$\tilde{\mathbf{v}}^{w} = [\mathbf{I} - (\partial\theta\times)]\mathbf{v}^{w} + \delta\mathbf{v}^{1} \qquad (4.30a)$$

Let the measured specific force vector $\tilde{\mathbf{f}}^{b}$ include the true specific force vector plus the total accelerometer measurement error ∇^{b}, then the computed specific force vector in the wander frame can be expressed as

$$\tilde{\mathbf{f}}^{w} = \tilde{\mathbf{C}}_{b}^{w}\tilde{\mathbf{f}}^{b} = [\mathbf{I} - (\partial\phi\times)]\mathbf{C}_{b}^{w}(\mathbf{f}^{b} + \nabla^{b})$$

Therefore,

$$\tilde{\mathbf{f}}^{w} \approx [\mathbf{I} - (\partial\phi\times)]\mathbf{f}^{w} + \nabla^{w} \qquad (4.30b)$$

where ∇^{w} is the total accelerometer error in the wander frame.
The gravity vector is approximated as

$$\tilde{\mathbf{g}}^{w} = \mathbf{g}^{w} + \delta\mathbf{g}^{w} \qquad (4.30c)$$

where $\delta\mathbf{g}^{w}$ is the variation of the gravity vector in terms of the wander frame.

Substituting Eqs. (4.30a, b, c) into Eq. (4.16), the velocity error equation can be derived as follows:

$$\delta\dot{\mathbf{v}}^{1} = -[(\boldsymbol{\omega}_{e/w}^{w}\times) + 2(\boldsymbol{\omega}_{I/e}^{w}\times)]\delta\mathbf{v}^{1} + \mathbf{f}^{w}\times\partial\phi + \delta\mathbf{g}^{w} - (\mathbf{f}^{w} + \mathbf{g}^{w})\times\partial\theta + \nabla^{w} \qquad (4.31)$$

This represents a simplified velocity error differential equation. The approximate expression of $\delta\mathbf{g}^{w}$ is given in Appendix D.

4.5.2 Position Error Equations

The angular position error $\partial\theta$ is defined in terms of the computed position DCM $\tilde{\mathbf{C}}_{e}^{w}$ and the true position DCM $\mathbf{C}_{e}^{w}$ as follows:

$$\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}} = [\mathbf{I} - (\partial\theta\times)]\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} \tag{4.32}$$

It can be rewritten as

$$\delta\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} = \tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}} - \mathbf{C}_{\mathrm{e}}^{\mathrm{w}} \tag{4.33a}$$

$$\delta\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} = -(\partial\theta\times)\mathbf{C}_{\mathrm{e}}^{\mathrm{w}} \tag{4.33b}$$

Let the latitude, longitude and wander angles be expressed in terms of their true values plus error terms as follows:

$$\tilde{\lambda} = \lambda + \delta\lambda, \qquad \tilde{\varphi} = \varphi + \delta\varphi, \qquad \alpha = \alpha + \delta\alpha \tag{4.34}$$

From Appendix D, the linear position error differential equation can be deduced as follows:

$$\begin{bmatrix} \delta\dot{\varphi}_{\mathrm{R}} \\ \delta\dot{\lambda}_{\mathrm{R}} \\ \delta\dot{\mathrm{h}} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta\mathrm{v}_{\mathrm{x}}^{1} \\ \delta\mathrm{v}_{\mathrm{y}}^{1} \\ \delta\mathrm{v}_{\mathrm{z}}^{1} \end{bmatrix} +$$

$$\begin{bmatrix} \dfrac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} & 0 & -\dfrac{\mathrm{v}_{\mathrm{x}}\cos(\alpha) + \mathrm{v}_{\mathrm{y}}\sin(\alpha)}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} \\ \dfrac{-\mathrm{v}_{\mathrm{x}}\sin(\alpha) + \mathrm{v}_{\mathrm{y}}\cos(\alpha)}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}}\tan(\varphi) & \dfrac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} & -\dfrac{-\mathrm{v}_{\mathrm{x}}\sin(\alpha) + \mathrm{v}_{\mathrm{y}}\cos(\alpha)}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} \\ \dfrac{\mathrm{v}_{\mathrm{x}}\cos(\alpha) + \mathrm{v}_{\mathrm{y}}\sin(\alpha)}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} & \dfrac{-\mathrm{v}_{\mathrm{x}}\sin(\alpha) + \mathrm{v}_{\mathrm{y}}\cos(\alpha)}{\mathrm{R}_{\mathrm{a}} + \mathrm{h}} & 0 \end{bmatrix} \begin{bmatrix} \delta\varphi_{\mathrm{R}} \\ \delta\lambda_{\mathrm{R}} \\ \delta\mathrm{h} \end{bmatrix} \tag{4.35}$$

where the linear position errors are defined as

$$\delta\varphi_{\mathrm{R}} = (\mathrm{R}_{\mathrm{n}} + \mathrm{h})\delta\varphi \approx (\mathrm{R}_{\mathrm{a}} + \mathrm{h})\delta\varphi$$

$$\delta\lambda_{\mathrm{R}} = (\mathrm{R}_{\mathrm{e}} + \mathrm{h})\cos(\varphi)\delta\lambda \approx (\mathrm{R}_{\mathrm{a}} + \mathrm{h})\cos(\varphi)\delta\lambda$$

4.5.3 Attitude Error Equations

Due to inertial sensor measurement and computation errors, the attitude DCM obtained by integrating Eq. (4.19) contains errors. This computed attitude DCM $\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}}$ can be represented in terms of the true attitude DCM $\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ as follows:

$$\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = [\mathbf{I} - (\partial\phi\times)]\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{4.36}$$

where $\partial\phi$ is known as the tilt error vector.

Let $\delta\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} = \tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} - \mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ (4.37a) then $\delta\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} = -(\partial\phi\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ (4.37b)

Differentiating Eqs. (4.37a) and (4.37b) yields

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -(\partial\dot{\phi}\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} - (\partial\phi\times)\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} \tag{4.38}$$

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = \dot{\tilde{\mathbf{C}}}_{\mathrm{b}}^{\mathrm{w}} - \dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -(\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{\mathrm{w}}\times)\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} + (\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{4.39}$$

From Appendix E, the tilt error differential equation can be derived as follows:

$$\partial\dot{\phi} = -[(\boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{w}}\times) + (\boldsymbol{\omega}_{\mathrm{e/w}}^{\mathrm{w}}\times)]\partial\phi + \delta\boldsymbol{\omega}_{\mathrm{e/w}}^{\mathrm{w}} + (\boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{w}}\times)\partial\theta - \Delta^{\mathrm{w}} \tag{4.40}$$

Let the computed Euler angles be expressed in terms of their true values plus error terms as follows:

$$\tilde{\phi} = \phi + \delta\phi,\ \tilde{\theta} = \theta + \delta\theta,\ \tilde{\psi}_{\mathrm{w}} = \psi_{\mathrm{w}} + \delta\psi_{\mathrm{w}} \tag{4.41}$$

From Appendix E, the attitude errors can be written in vector form as

$$\begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi_{\mathrm{w}} \end{bmatrix} = \mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}} \begin{bmatrix} \partial\phi_{\mathrm{x}} \\ \partial\phi_{\mathrm{y}} \\ \partial\phi_{\mathrm{z}} \end{bmatrix} \tag{4.42}$$

where $\mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}}$ is the transformation matrix from the tilt errors to the Euler errors.

4.5.4 Inertial Sensor Error Models

An inertial sensor system measures the kinematic parameters (6DOF angular velocities and linear accelerations) of aircraft motion, which are used in navigation algorithms, as described in Section 4.3, to obtain the navigation states. Ideally, the output of an inertial sensor should provide an exact measurement of its input, the measured kinematic parameter. In practice, the output of an inertial sensor will contain errors, including nonlinearity of scale factor, misalignment between sensor sensing axis and input axis, coupling effect between angular and linear motions, uncertainty of sensor system design itself (including uncertainty of sensor dynamic model), imperfect sensor signal detection and processing and measurement noise. In this thesis, separate model equations are defined for gyroscopes and accelerometers. These models define the mathematical relationship between the outputs of inertial sensors and the inputs, including applied acceleration, angular velocity and angular acceleration along the sensor reference axes.

The generalised model equation of a gyro is defined as follows:

$$\omega_{\text{out}} = \omega_{\text{in}} + \Delta = \omega_{\text{in}} + \Delta_{\text{B}} + \Delta_{\text{R}} + \Delta_{\text{SF}}\omega_{\text{in}} + \Delta_{\text{Mis}} + \Delta_{\text{A}} + \Delta_{\text{T}}(T) + \varepsilon_{\Delta} \tag{4.43}$$

where

ω_{out} and ω_{in} are the gyro output and input, respectively,

Δ_{B} is the gyro bias, or zero offset,

Δ_{R} is the gyro random drift rate, which may be caused by environmental and other external influences, such as disturbed torque in a mechanical gyro.

Δ_{SF} is the gyro scale factor error, caused by nonlinearity or instability resulting from the gyro scale factor.

Δ_{Mis} is a gyro misalignment-dependent error, caused by misalignment between the gyro input axis and its associated reference axis.

Δ_{A} is the acceleration-sensitive drift rate, which may include acceleration and acceleration-squared sensitivities.

Δ_{T} is a temperature-dependent gyro drift rate, and

ε_{Δ} is gyro measurement noise.

For different gyro sensors, some of the terms in Eq. (4.43) may be omitted. For example, laser gyros usually exhibit random walk in the gyro drift but it is not necessary to specify acceleration-sensitive drift. However, for mechanical gyros, gyro drift caused by acceleration sensitivity has to be considered.

Accelerometer errors may arise from the angular motion and the acceleration motion of the aircraft, random bias, scale factor, dead zone, cross-axis sensitivity, temperature and other factors. A generalised model equation of an accelerometer is defined as

$$\text{f}_{\text{out}} + \nabla = \text{f}_{\text{in}} + \nabla_{\text{B}} + \nabla_{\text{R}} + \Delta_{\text{SF}}\text{f}_{\text{in}} + \nabla_{\text{Mis}} + \nabla_{\omega} + \nabla_{\text{A}} + \nabla_{\text{T}} + \varepsilon_{\nabla} \tag{4.44}$$

where

f_{out} and f_{in} are the output and input of an accelerometer, separately.

∇_{B} is the accelerometer bias,

∇_{R} is an accelerometer time-dependent random bias. The random bias is a critical aspect of accelerometer performance. This bias must be estimated and corrected with a stochastic process in the integrated Kalman filter.

Δ_{SF} is the accelerometer scale factor error,

∇_{Mis} is the accelerometer misalignment bias, caused by misalignment angle between the accelerometer input axis and its associated reference axis.

∇_{ω} is the accelerometer output bias, caused by angular motion of aircraft,

∇_{A} is the acceleration-sensitive accelerometer bias, including cross-coupling effect and higher order acceleration-sensitive terms.

∇_{T} is a temperature-dependent accelerometer bias, and

ε_{∇} is the accelerometer measurement noise.

Generally, the first four error terms of inertial sensors in Eqs. (4.43) and (4.44) are critical to inertial navigation system. These error terms need to be estimated and corrected in flight to improve the performance of aircraft navigation systems.

4.6 Navaid Systems

Although an inertial reference system can provide all the necessary information for computation of all the navigation states, it suffers from time-accumulated drift errors, as described in Section 4.4. Navaid systems generally supply only partial information on the navigation states. However, they exhibit a long-term stability and high positioning accuracy dependent on navaid systems. Therefore, the navigation states given by navaid systems can be used as constraints on some of the navigation states derived by INS.

The measurement equations and navigation models of several navaid systems, for example, global navigation satellite system (GNSS) and Doppler radar systems, are developed in this section. An air data sensor system is used in aircraft systems for navigation and flight control. For example, the pressure altitude is widely used to aid the vertical channel of inertial navigation systems and to maintain vertical height. However, air data sensor systems will not be discussed in this thesis.

GNSS is an all-weather, space-based radio navigation system providing global coverage. There are at present three similar versions of GNSS: the US Global Positioning System (GPS), the Russian Global Orbital Navigation Satellite System (GLONASS) and the European Galileo satellite navigation system. These three satellite navigation systems are mainly distinguished by the satellite orbital planes, the number of operating satellites, representation of satellite orbit parameters, data modulation methods, frequency bands and signal structures. For example, GPS uses the code division multiple access (CDMA) technique whereas GLONASS uses the frequency division multiple access (FDMA). GPS has 24 operational satellites in six orbit planes around 20200 km above the Earth's surface whereas the Galileo system will have 30 satellites in three orbit planes around 24000 km. GLONASS uses PZ-90 coordinate frame whereas GPS uses the WGS-84 system. The GPS C/A-code rate is 1.023 Mbit/s while GLONASS has a value of 0.511 Mbit/s.

GNSS timing signals are very precisely defined pseudo random noise (PRN) codes, which are modulated on the satellite carrier signals together with the navigation message containing the satellite orbit parameters. GNSS satellites may broadcast different PRN codes on several carrier frequencies for different services. For example, GPS satellites broadcast the C/A-code on the L1 carrier (1575.42 MHz) for civilian standard positioning services and the P(Y)-code on both the L1 and L2 (1227.60 MHz) carriers for military precise positioning services[83]. By offering dual frequencies as standard, Galileo will deliver higher real-time position accuracy than the current GPS or GLONASS positioning services. However, modernised GPS will offer a new L5 frequency and L2 civil signal to enhance civil and aviation services.

These systems share the same positioning principle, that is, they all determine the position of a receiver by measuring time differences of timing signals travelling from GNSS satellites to the receiver. Therefore, the positioning and navigation equations developed in this section apply to all three GNSS systems. A GNSS receiver is designed to track and capture the satellite timing codes and to demodulate the navigation message in order to compute position. This method is known as the code-phase measurement. A GNSS receiver can also track and measure the phases of carrier signals transmitted by GNSS satellites, which are referred to as carrier phase measurement.

4.6.1 GNSS Observation Equations

From the user's perspective, GNSS satellites generate and broadcast a series of timing codes. These code signals propagate through the atmosphere in space to a user receiver. The receiver tracks and measures the time delays of these codes to estimate the time difference of the signal propagation. Let the time at which a GNSS satellite transmits its timing code signal be t_{S} and the time at which a GNSS receiver receives this timing code signal be t_{R}, then the time difference of this signal propagation is

$$\Delta t = t_{\mathrm{R}} - t_{\mathrm{S}}$$

Because the satellite and receiver clocks are not perfect, the receiver time t_{R} and the satellite time t_{S} will contain errors dt_{R} and dt_{S}, respectively. Therefore, the above equation can be rewritten as:

$$\Delta t = t_{\mathrm{R}} + dt_{\mathrm{R}} - (t_{\mathrm{S}} + dt_{\mathrm{S}}) = (t_{\mathrm{R}} - t_{\mathrm{S}}) + dt_{\mathrm{R}} - dt_{\mathrm{S}} \tag{4.45}$$

Assume that the timing signals transmitted by GNSS satellites travel at the speed of light c, then the range between the satellite and receiver can be represented as

$$\mathrm{r} \equiv c\Delta t = c(t_{\mathrm{R}} - t_{\mathrm{S}}) + cdt_{\mathrm{R}} - cdt_{\mathrm{S}}$$

or $$\mathrm{r} = \rho + cdt_{\mathrm{R}} - cdt_{\mathrm{S}} \tag{4.46}$$

where $c(t_{\mathrm{R}} - t_{\mathrm{S}}) = \rho$ is the true distance between the GNSS satellite and receiver, cdt_{R} is the range error caused by uncertainty of the receiver clock, cdt_{S} is the range error caused by the satellite clock error and r is usually known as the pseudorange measurement.

Consider various signal propagation path delays, including the clock errors, satellite orbit errors and measurement noise. The model equation of the pseudorange measurement can be expressed as[83]

$$\mathrm{r} = \rho + d\rho + d_{\mathrm{iono}} + d_{\mathrm{trop}} + d_{\mathrm{mp}} + cdt_{\mathrm{R}} - cdt_{\mathrm{S}} + \varepsilon_{\mathrm{r}} \tag{4.47}$$

where

$d\rho$ is the range error caused by satellite orbit errors,

d_{iono} is the range error caused by the ionospheric path delay,

d_{trop} is the range error caused by the tropospheric path delay,

d_{mp} is the range error caused by multipath effects, reflecting surfaces around the GNSS receiver antenna.

ε_{r} is the measurement noise.

If a GNSS receiver is able to track both the timing signals and the carrier signals, then carrier phase measurements can also be obtained. When the carrier signal transmitted by a satellite reaches a GNSS receiver, the relative motion between the satellite and the receiver causes a Doppler shift of the arrival carrier signal at the receiver side. If the receiver's carrier phase tracking loop can lock onto the carrier signal, the receiver can continuously measure or count the Doppler shift, known as the Doppler count. Because the initial locking time is unknown, the initial Doppler count is unknown and is referred to as carrier phase integer ambiguity. The sum of the accumulated Doppler count and fractional phase measurement is the total carrier phase, which is an equivalent range measurement.

The model equation of the carrier phase measurement can be represented as[83]

$$\lambda\varphi = \rho + d\rho + \lambda N - d_{\mathrm{iono}} + d_{\mathrm{trop}} + d_{\mathrm{mp}} + cdt_{\mathrm{R}} - cdt_{\mathrm{S}} + \varepsilon_{\varphi} \tag{4.48}$$

where

$\lambda\varphi$ is the equivalent pseudorange from a satellite to a receiver,

φ is the totally measured phase,

λ is the wavelength of measured carrier frequency,

N is the carrier phase integer ambiguity, which is a constant once the carrier signal is locked and tracked.

ε_{φ} is the carrier phase measurement noise.

Because the ionosphere causes the group speed of radio signals to be delayed and the phase speed of the radio signals to be advanced, the ionospheric delay d_{iono} is negative in Eq. (4.48) and positive in Eq. (4.47).

Although the carrier phase measurement is potentially more accurate than the code phase measurement, the carrier phase integer ambiguity occurring in Eq. (4.48) is an inherent drawback in carrier tracking measurements. In order to benefit from the high accuracy property of the carrier phase measurement, the integer ambiguity must be correctly resolved.

The Doppler shift is a measure of the rate of change of the relative range between a GNSS satellite and a receiver along the line of sight. If the GNSS satellite velocities are known, then the instantaneous Doppler measurement can be used to determine the receiver velocity. The model equation of the Doppler measurement can be expressed as

$$\dot{\mathrm{r}} = \dot{\rho} + d\dot{\rho} + \dot{d}_{\mathrm{iono}} + \dot{d}_{\mathrm{trop}} + \dot{d}_{\mathrm{mp}} + cd\dot{t}_{\mathrm{R}} - cd\dot{t}_{\mathrm{S}} + \varepsilon_{\dot{\mathrm{r}}} \tag{4.49}$$

where

$\dot{\mathrm{r}} = \lambda\dot{\varphi}$ is the pseudorange rate, $\dot{\varphi} = \Delta f$ is the Doppler frequency shift.

$\dot{\rho}$ is the true range rate along the line of sight between satellite and receiver,

$d\dot{\rho}$ is the range rate error caused by satellite velocity errors,

$\dot{d}_{\mathrm{iono}}$ is the range rate error caused by the ionosphere,

$\dot{d}_{\mathrm{trop}}$ is the range rate error caused by the troposphere,

$\dot{d}_{\mathrm{mp}}$ is the range rate error caused by multipath effects,

$cd\dot{t}_{\mathrm{R}}$ is the range rate error caused by the receiver clock frequency drift,

$cd\dot{t}_{\mathrm{S}}$ is the range rate error caused by the satellite clock frequency drift, and

$\varepsilon_{\dot{\mathrm{r}}}$ is the Doppler measurement noise.

Eqs. (4.47), (4.48 and (4.49) constitute the GNSS measurement model. Since GNSS satellite orbit parameters are precisely estimated by GNSS ground data processing centres and satellite system time is held by highly precise atomic clocks, these error terms can be neglected for navigation users.

Two methods are used to reduce the error terms in the GNSS measurement equations. One method is to use an ionospheric and tropospheric delay models[83]. For example GPS generally broadcasts ionospheric correction parameters as a part of the navigation message. However, it is difficult to eliminate all the range errors caused by atmospheric path delays simply by use of these models[83].

Another approach is to develop augmented GNSS systems. There are two kinds of augmentation systems: local and global augmentations. Local area augmentation systems use differential GNSS[83] and pseudolite[105] techniques to correct the error terms in the GNSS measurements and to improve the integrity and availability of GNSS satellites in a local area. In a local area augmentation system, a reference station at a known location receives and processes its local GNSS observables to obtain the range and range rate corrections to each visible GNSS satellite, and then broadcasts these differential corrections and GNSS signal integrity information to near users. These users can utilise this correction information to correct their GNSS measurements. This differential GNSS technique can only cancel those common-view error sources to both the reference station and the users, such as, satellite orbit and clock errors and atmospheric path delay errors. A pseudolite is a ground-based beacon at known location and transmits timing signals similar to GNSS satellites[105]. The pseudolite techniques can improve the availability of GNSS signals and the LOS geometry of user receivers within a specific region. Therefore, the local positioning accuracy and signal integrity are improved.

There are three compatible versions of global augmentation system: the US wide area augmentation system (WAAS)[106], the European geostationary navigation overlay service (EGNOS))[107] and the Japanese multifunctional transport satellite space-based augmentation system (MSAS))[108]. All these systems are space-ground combined systems and broadcast the real-time clock, ephemeris and atmospheric correction parameters, augmented timing/ranging signals and integrity information of GNSS satellites signals. These correction parameters allow users to obtain accuracies approaching those of local-area differential GPS systems[108]. These spaced-based systems not only improve the accuracy of GNSS but also enhance the integrity, time availability and continuity of GNSS service[109][110].

It should be noted that the local area augmentation systems are to mainly reduce the effects of the common-view error sources on the GNSS measurements. Therefore, the correction is local. However, the space-based augmentation systems directly reduce or remove the error sources. Their corrections are global because all GNSS users can use these corrections.

In addition, the carrier phase measurements are generally used to smooth the pseudoranges to improve the accuracy of the pseudorange measurements.

4.6.2 GNSS Navigation Equations

In order to develop GNSS navigation equations, assuming that the main error terms in Eqs. (4.47) and (4.49) have been properly corrected by use of the techniques mentioned as above. Therefore, the model equations of these GNSS measurements can be simplified in terms of range and range rate measurements, as follows:

Range equivalent measurement: $\mathrm{r}^i(t_k) = \rho^i(t_k) + c * dt + v_{\mathrm{r}}^i$ (4.50)

Range rate equivalent measurement: $\dot{\mathrm{r}}^i(t_k) = \dot{\rho}^i(t_k) + c * d\dot{t} + v_{\dot{\mathrm{r}}}^i$ (4.51)

where v_{r} is the range equivalent measurement noise, covering the measurement noise and all residual errors after the propagation path delay corrections; $v_{\dot{\mathrm{r}}}$ is the range rate equivalent measurement noise, including the receiver measurement noise and all residual rate errors after the propagation path delay corrections, and the superscript i represents an observed GNSS satellite. Hereafter, the subscript R in the receiver clock error terms is omitted for simplifying representation.

The true range ρ^i in Eq. (4.50) is a nonlinear algebraic equation containing the unknown position of the receiver and the known position of the satellite i as follows:

$$\rho^i(t_k) = \sqrt{[\mathrm{x}^i(t_{\mathrm{S}k}) - \mathrm{x}(t_k)]^2 + [\mathrm{y}^i(t_{\mathrm{S}k}) - \mathrm{y}(t_k)]^2 + [\mathrm{z}^i(t_{\mathrm{S}k}) - \mathrm{z}(t_k)]^2} \qquad (4.52)$$

where $[\mathrm{x}^i, \mathrm{y}^i, \mathrm{z}^i]^{\mathrm{T}}$ is the position vector of the satellite i in the ECEF frame at the signal transmission time $t_{\mathrm{S}k}$ and $[\mathrm{x}, \mathrm{y}, \mathrm{z}]^{\mathrm{T}}$ is the position vector of GNSS receiver in ECEF coordinates at the signal reception time t_k.

The true range rate $\dot{\rho}^i$ in Eq. (4.51) is a projection of the relative velocity between the satellite and the receiver onto the line of sight along the GPS satellite i to the receiver. The range rate can be represented as:

$$\dot{\rho}^i(t_k) = \frac{(\mathrm{x}^i - \mathrm{x})}{\rho^i}(\dot{\mathrm{x}}^i - \dot{\mathrm{x}}) + \frac{(\mathrm{y}^i - \mathrm{y})}{\rho^i}(\dot{\mathrm{y}}^i - \dot{\mathrm{y}}) + \frac{(\mathrm{z}^i - \mathrm{z})}{\rho^i}(\dot{\mathrm{z}}^i - \dot{\mathrm{z}}) \qquad (4.53)$$

where $[\dot{\mathrm{x}}^i, \dot{\mathrm{y}}^i, \dot{\mathrm{z}}^i]^{\mathrm{T}}$ is the velocity vector of the satellite i at time $t_{\mathrm{S}k}$ in terms of the ECEF coordinates and $[\dot{\mathrm{x}}, \dot{\mathrm{y}}, \dot{\mathrm{z}}]^{\mathrm{T}}$ is the true velocity vector of receiver at time t_k in terms of the ECEF coordinates. Hereafter, the time symbols in round brackets will be omitted to simplify the forms of the GNSS measurement equations.

Two forms of GNSS navigation algorithms are commonly used. One is based on the least-squares method whereas the other uses Kalman filtering techniques. Both needs to linearise the GNSS measurement models about nominal points, i.e. the approximate position and velocity of a GNSS receiver.

From Appendix F, the GNSS navigation equations can be represented in vector form as follows:

$$\delta\mathbf{r} = \mathbf{H}_{\mathrm{GNSS}}\delta\mathbf{p}_{\mathrm{p}} + \boldsymbol{\upsilon}_{\mathrm{r}} \tag{4.54}$$

$$\delta\dot{\mathbf{r}} = \mathbf{H}_{\mathrm{GNSS}}\delta\dot{\mathbf{p}}_{\mathrm{p}} + \mathbf{H}_{\mathrm{LOSV}}\delta\mathbf{p} + \boldsymbol{\upsilon}_{\dot{\mathrm{r}}} \tag{4.55}$$

where

$$\delta\mathbf{p}_{\mathrm{p}} = \begin{bmatrix} \delta\mathbf{p} \\ cdt \end{bmatrix}, \delta\mathbf{p} = \begin{bmatrix} \delta\mathrm{x} \\ \delta\mathrm{y} \\ \delta\mathrm{z} \end{bmatrix},$$

Eqs. (4.54) and (4.55) constitute the fundamental GNSS navigation algorithms. In the extended least-squares method, Eqs. (4.54) and (4.55) are directly solved to obtain the position and velocity states. In the Kalman filtering method, Eqs. (4.54) and (4.55) are used in the filter measurements and the aircraft motion or the inertial system error dynamics must be modelled. This thesis will investigate multisensor data fusion navigation systems using the Kalman filter approach and the dynamic models of inertial system errors developed in Section 4.4.

From the least-squares estimation, the uncertainty of estimate of the GNSS-based navigation state can be characterised by the inverse matrix $\left[(\mathbf{H}_{\mathrm{GNSS}})^{\mathrm{T}}\mathbf{H}_{\mathrm{GNSS}}\right]^{-1}$, which represents the geometry of visible GNSS satellites in space. The squared-root of the trace of this inverse matrix is usually known as the geometric dilution of precision (GDOP) factor. Apparently, GDOP changes with the number of observed GNSS satellites and their configurations. Minimising GDOP is generally used as a criterion to select optimal visible satellites. The GDOP is formulated in the ECEF frame as

$$\mathrm{GDOP} = \sqrt{tr(\left[(\mathbf{H}_{\mathrm{GNSS}})^{\mathrm{T}}\mathbf{H}_{\mathrm{GNSS}}\right]^{-1})} \tag{4.56}$$

Transforming this inverse matrix from the ECEF frame to the NEU frame, the GDOP factor in the NEU frame can be formulated as

$$\mathrm{GDOP}^{\mathrm{n}} = \sqrt{tr(\mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\left[(\mathbf{H}_{\mathrm{GNSS}})^{\mathrm{T}}\mathbf{H}_{\mathrm{GNSS}}\right]^{-1}\mathbf{C}_{\mathrm{n}}^{\mathrm{e}})} = \sqrt{tr(\left[\mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\mathbf{H}_{\mathrm{GNSS}}^{\mathrm{T}}(\mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\mathbf{H}_{\mathrm{GNSS}}^{\mathrm{T}})^{\mathrm{T}}\right]^{-1})} \quad (4.57)$$

where $\mathbf{C}_{\mathrm{e}}^{\mathrm{n}}$ is the 4x4 transformation matrix from the ECEF frame to the NEU frame. Several other alternative DOP values used in evaluating satellite constellations are position dilution of precision (PDOP), horizontal dilution of precision (HDOP) and vertical dilution of precision (VDOP). The product of a DOP value and ranging error determines the corresponding position fix error[83].

Further investigations in GNSS applications[83][111-114] have shown that the use of differential carrier phase measurement techniques, based on three or four GNSS antennas, can obtain aircraft attitude information. This technique is usually known as GNSS-based attitude determination. The relative position vector between two GNSS antennas is referred to as the baseline. In the aircraft body frame, the orientation of this baseline is known very precisely. The phase difference between two antennas is an estimate of the projection of this baseline onto the line of sight to the observed satellite. This principle is illustrated in Figure 4.5 where the parallel carrier signals of the satellite i arrive at the different antennas. Because the magnitude of this baseline is constant in any coordinate frames, taking the NEU navigation frame as a reference, the model for GNSS-based attitude determination can be represented as follows:

$$d\lambda\varphi_j^i = (\mathbf{b}_j^{\mathrm{b}})^{\mathrm{T}}\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}\mathbf{s}^{\mathrm{n}/i} + \lambda N_j^i + v_{d\varphi-j}^i \quad (4.58)$$

where

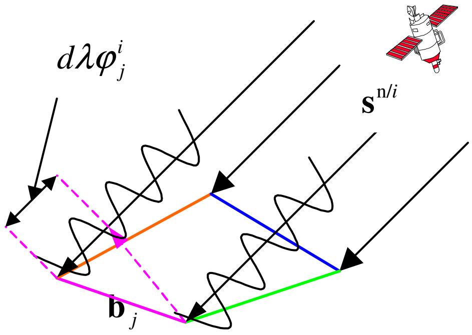


Figure 4.5 Baseline Measurements

$d\lambda\varphi_j^i$ is the differential phase observation to the satellite i from the baseline j,

$\mathbf{b}_j^{\mathrm{b}}$ is the known baseline j, represented in the body coordinates,

N_j^i is the relative phase ambiguity of carrier frequency of the observed satellite i with respective to the baseline j,

$\mathbf{s}^{\mathrm{n}/i} = \mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\mathbf{s}^{\mathrm{e}/i}$ is the known LOS vector of the satellite i coordinated in the NEU frame,

$\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}$ is the unknown attitude DCM and $(\mathbf{b}_j^{\mathrm{n}})^{\mathrm{T}} = (\mathbf{b}_j^{\mathrm{b}})^{\mathrm{T}}\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}$, and

$v_{d\varphi-j}^{i}$ is the differential carrier phase measurement noise relative to the satellite i along the baseline j.

For three baselines (J, K, L), where more than three GNSS satellites are visible, the model equation of the GNSS-based attitude determination can be rearranged in matrix form as follows:

$$d\mathbf{\Phi} = \mathbf{B}_{\mathrm{bl}}\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}\mathbf{S}^{\mathrm{n}} + \lambda\mathbf{N} + \boldsymbol{\upsilon}_{d\Phi} \tag{4.59}$$

where

$d\mathbf{\Phi} = \begin{bmatrix} d\lambda\varphi_J^1 & d\lambda\varphi_J^2 & d\lambda\varphi_J^3 & \cdots & d\lambda\varphi_J^m \\ d\lambda\varphi_K^1 & d\lambda\varphi_K^2 & d\lambda\varphi_K^3 & \cdots & d\lambda\varphi_K^m \\ d\lambda\varphi_L^1 & d\lambda\varphi_L^2 & d\lambda\varphi_L^3 & \cdots & d\lambda\varphi_L^m \end{bmatrix}$ is the differential phase matrix,

$\mathbf{N} = \begin{bmatrix} N_J^1 & N_J^2 & N_J^3 & \cdots & N_J^m \\ N_K^1 & N_K^2 & N_K^3 & \cdots & N_K^m \\ N_L^1 & N_L^2 & N_L^3 & \cdots & N_L^m \end{bmatrix}$ is the relative ambiguity matrix,

$\mathbf{B}_{\mathrm{bl}} = \begin{bmatrix}\mathbf{b}_J^{\mathrm{b}} & \mathbf{b}_K^{\mathrm{b}} & \mathbf{b}_L^{\mathrm{b}}\end{bmatrix}^{\mathrm{T}}$ is the baseline matrix in terms of the body coordinates,

$\mathbf{S}^{\mathbf{n}} = \begin{bmatrix}\mathbf{s}^{\mathrm{n}/1} & \mathbf{s}^{\mathrm{n}/2} & \mathbf{s}^{\mathrm{n}/3} & \cdots & \mathbf{s}^{\mathrm{n}/m}\end{bmatrix} = \mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\mathbf{H}_{\mathrm{LOS}}^{\mathrm{T}}$ is the known LOS DCM coordinated in the NEU frame, and

$\boldsymbol{\upsilon}_{d\Phi}$ is the differential phase measurement noise matrix.

Given that the relative ambiguity matrix $\mathbf{N}$ has been resolved, two methods can be used to solve Eq. (4.59), depending on the configuration of the baseline vectors in the aircraft body frame. If these three baseline vectors are non-coplanar in the body frame, then an inverse of the baseline matrix $\mathbf{B}_{\mathrm{bl}}$ exists. By using least-squares techniques, the attitude matrix can be computed as follows:

$$\tilde{\mathbf{C}}_{\mathrm{n}}^{\mathrm{b}} = \mathbf{B}_{\mathrm{bl}}{}^{-1}d\mathbf{\Phi}(\mathbf{S}^{\mathrm{n}})^{\mathrm{T}}\left[\mathbf{S}^{\mathrm{n}}(\mathbf{S}^{\mathrm{n}})^{\mathrm{T}}\right]^{-1} \tag{4.60}$$

where $d\mathbf{\Phi}$ includes the resolved relative ambiguity matrix $\mathbf{N}$. The inverse matrix $\left[\mathbf{S}^{\mathrm{n}}(\mathbf{S}^{\mathrm{n}})^{\mathrm{T}}\right]^{-1}$ can be achieved by selecting appropriate GNSS satellites. GNSS attitude algorithms based on this method are generally known as the direct attitude matrix determination algorithm.

If the baseline vectors are coplanar, Eq. (4.59) is reduced to vector form as follows:

$$d\mathbf{\Phi}_j = (\mathbf{b}_j^{\mathrm{b}})^{\mathrm{T}}\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}\mathbf{S}^{\mathrm{n}} + \lambda\mathbf{N}_j + \boldsymbol{\upsilon}_{d\Phi-j} \tag{4.61}$$

where $d\mathbf{\Phi}_j (j = J, K, L)$ are the row vectors of the differential phase matrix $d\mathbf{\Phi}$, $\mathbf{N}_j$ is the row vector of $\mathbf{N}$ and $\boldsymbol{\upsilon}_{d\Phi-j}$ is the row vector of $\boldsymbol{\upsilon}_{d\Phi}$.

By independently resolving Eq. (4.61) for each row vector of $d\mathbf{\Phi}$, the baseline vectors can be obtained in terms of the navigation coordinates, as follows:

$$\tilde{\mathbf{b}}_j^{\mathrm{n}} = \left[\mathbf{S}^{\mathrm{n}}(\mathbf{S}^{\mathrm{n}})^{\mathrm{T}}\right]^{-1}\mathbf{S}^{\mathrm{n}}(d\mathbf{\Phi}_j)^{\mathrm{T}} \equiv \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\mathbf{b}_j^{\mathrm{b}} \tag{4.62}$$

Combining any two baseline vectors given by Eq. (4.62) results in

$$\left[\tilde{\mathbf{b}}_J^{\mathrm{n}} \quad \tilde{\mathbf{b}}_K^{\mathrm{n}} \quad \tilde{\mathbf{b}}_J^{\mathrm{n}} \times \tilde{\mathbf{b}}_K^{\mathrm{n}}\right] = \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\left[\mathbf{b}_J^{\mathrm{b}} \quad \mathbf{b}_K^{\mathrm{b}} \quad \mathbf{b}_J^{\mathrm{b}} \times \mathbf{b}_K^{\mathrm{b}}\right] \tag{6.63}$$

The attitude matrix is then given by

$$\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} = \left[\tilde{\mathbf{b}}_J^{\mathrm{n}} \quad \tilde{\mathbf{b}}_K^{\mathrm{n}} \quad \tilde{\mathbf{b}}_J^{\mathrm{n}} \times \tilde{\mathbf{b}}_K^{\mathrm{n}}\right]\left[\mathbf{b}_J^{\mathrm{b}} \quad \mathbf{b}_K^{\mathrm{b}} \quad \mathbf{b}_J^{\mathrm{b}} \times \mathbf{b}_K^{\mathrm{b}}\right]^{-1} \tag{4.64}$$

Obviously, this method requires only two baselines. These two baseline vectors (resolved in the navigation coordinates) must first be determined, and then the attitude angles or the attitude matrix can be computed by using the estimates of these baselines. GNSS attitude algorithms based on this method are referred to as indirect attitude matrix determination algorithm or relative positioning attitude determination algorithm. Using this relative technique, a single baseline can be used to determine the heading and pitch angles of an aircraft if this single baseline is orientated along the aircraft body x-axis.

Existing many GNSS attitude determination algorithms are generally based on one of these two fundamental methods. These algorithms may be distinguished by the computing methods used to resolve Eq. (4.59) and Eq. (4.61).

Different from kinematic positioning where the baseline length is usually long and unknown and the integer ambiguity is searched in a relatively large search space, the baseline length in aircraft attitude determination problem is precisely known and very short (typically 1.0-2.0 meters). Consequently, the integer ambiguity search in

GNSS-based aircraft attitude determination is based on the known baseline length or antenna geometry and the ambiguity search space is smaller. Several least squares-based integer ambiguity resolution techniques have been suggested for the attitude determination[83][113][115-118]. For any integer ambiguity resolution algorithm, a very important factor to be considered is the resolution time of each ambiguity algorithm, which is used to characterise how fast an ambiguity algorithm can obtain the correct value of integer ambiguity. Although the angular accuracy of a GNSS-based attitude determination algorithm is inversely proportional to the baseline length[83], the angular accuracy of the attitude solutions better than 0.5^0 (root-mean-square) has been achieved[83][119].

4.6.3 Normalised Measurement Models

The INS navigation state error models developed in Section 4.4 are represented in the navigation frame whereas the GNSS-based navigation state error models are coordinated in the ECEF frame. In order to develop data fusion filter, it is necessary to represent the states in these two kinds of models in a unified coordinate system. For aircraft navigation, the navigation frame is preferred as the reference frame.

A. Normalised Range Difference Equation

Rewriting Eq. (4.50) as

$$\mathrm{r}^i_{\mathrm{GNSS}}(t_k) = \rho^i(t_k) + c \cdot dt + v^i_{\mathrm{r}} \tag{4.65}$$

From the INS-derived aircraft position, the computed range between the satellite i and aircraft $\tilde{\mathrm{r}}^i_{\mathrm{INS}}$, corresponding to $\mathrm{r}^i_{\mathrm{GNSS}}$, can be expressed as follows:

$$\begin{aligned}\tilde{\mathrm{r}}^i_{\mathrm{INS}}(t_k) &= \rho^i(t_k) + d\mathrm{r}^i_{\mathrm{INS}}(t_k) \\ &= \sqrt{[\mathrm{x}^i(t_{\mathrm{T}k}) - \tilde{\mathrm{x}}(t_k)]^2 + [\mathrm{y}^i(t_{\mathrm{T}k}) - \tilde{\mathrm{y}}(t_k)]^2 + [\mathrm{z}^i(t_{\mathrm{T}k}) - \tilde{\mathrm{z}}(t_k)]^2}\end{aligned} \tag{4.66}$$

where $[\tilde{\mathrm{x}}(t_k), \tilde{\mathrm{y}}(t_k), \tilde{\mathrm{z}}(t_k)]^{\mathrm{T}}$ is the INS-derived aircraft position in terms of the ECEF coordinates and $d\mathrm{r}^i_{\mathrm{INS}}(t_k)$ is the range error caused by uncertainty of the INS-derived position.

Let the INS-derived position be expressed as follows:

$$\tilde{\mathrm{x}}(t_k) = \mathrm{x} + \delta\mathrm{x},\ \tilde{\mathrm{y}}(t_k) = \mathrm{y} + \delta\mathrm{y},\ \tilde{\mathrm{z}}(t_k) = \mathrm{z} + \delta\mathrm{z}$$

Linearising the squared-root term in Eq.(4.66) around this nominal point, $d\mathrm{r}_{\mathrm{INS}}^{i}$ can be derived as follows:

$$d\mathrm{r}_{\mathrm{INS}}^{i}(t_k) \approx \frac{-(\mathrm{x}^i - \tilde{\mathrm{x}})}{\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}}\delta\mathrm{x} + \frac{-(\mathrm{y}^i - \tilde{\mathrm{y}})}{\mathrm{r}_{\mathrm{INS}}^{i}}\delta\mathrm{y} + \frac{-(\mathrm{z}^i - \tilde{\mathrm{z}})}{\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}}\delta\mathrm{z} \qquad (4.67)$$

Differencing Eqs.(4.66) and (4.65) and comparing Eqs. (4.67) and (4.54) where ρ_0^i and $[\mathrm{x}_0, \mathrm{y}_0, \mathrm{z}_0]^{\mathrm{T}}$ are approximated by $\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}$ and $[\tilde{\mathrm{x}}, \tilde{\mathrm{y}}, \tilde{\mathrm{z}}]^{\mathrm{T}}$, respectively, the difference between the INS-derived and GNSS-measured ranges can be represented in vector form as follows:

$$\mathbf{r}_{\mathrm{GNSS}} - \tilde{\mathbf{r}}_{\mathrm{INS}} = -\mathbf{H}_{\mathrm{LOS}}\delta\mathbf{p} + \mathbf{1}cdt + \boldsymbol{\upsilon}_{\mathrm{r}}$$

In the NEU frame, this range difference equation is normalised as follows:

$$\mathbf{r}_{\mathrm{GNSS}} - \tilde{\mathbf{r}}_{\mathrm{INS}} = -\mathbf{H}_{\mathrm{LOS}}\mathbf{C}_{\mathrm{n}}^{\mathrm{e}}\delta\mathbf{p}^{\mathrm{n}} + \mathbf{1}cdt + \boldsymbol{\upsilon}_{\mathrm{r}} \qquad (4.68)$$

where $\delta\mathbf{p}^{\mathrm{n}} = [\delta\varphi_{\mathrm{R}} \quad \delta\lambda_{\mathrm{R}} \quad \delta\mathrm{h}]^{\mathrm{T}}$ is the linear position error coordinated in the NEU frame.

B. Normalised Range Rate Difference Equation

The GNSS range rate equation can be rewritten as

$$\dot{\mathrm{r}}_{\mathrm{GNSS}}^{i}(t_k) = \dot{\rho}^{i}(t_k) + cd\dot{t} + v_{\mathrm{r}}^{i} \qquad (4.69)$$

The INS-derived range rate can be expressed as

$$\begin{aligned}\dot{\tilde{\mathrm{r}}}_{\mathrm{INS}}^{i}(t_k) &= \dot{\rho}^{i}(t_k) + d\dot{\mathrm{r}}_{\mathrm{INS}}^{i} \\ &= \frac{(\mathrm{x}^i - \tilde{\mathrm{x}})}{\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}}(\dot{\mathrm{x}}^i - \dot{\tilde{\mathrm{x}}}) + \frac{(\mathrm{y}^i - \tilde{\mathrm{y}})}{\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}}(\dot{\mathrm{y}}^i - \dot{\tilde{\mathrm{y}}}) + \frac{(\mathrm{z}^i - \tilde{\mathrm{z}})}{\tilde{\mathrm{r}}_{\mathrm{INS}}^{i}}(\dot{\mathrm{z}}^i - \dot{\tilde{\mathrm{z}}})\end{aligned} \qquad (4.70)$$

where $d\dot{\mathrm{r}}_{\mathrm{INS}}^{i}$ is the range rate error caused by uncertainty of the INS velocity solution and $[\dot{\tilde{\mathrm{x}}}, \dot{\tilde{\mathrm{y}}}, \dot{\tilde{\mathrm{z}}}]^{\mathrm{T}}$ is the aircraft velocity derived by the INS in the ECEF frame.

Let the INS derived aircraft velocity $[\dot{\tilde{\mathrm{x}}}, \dot{\tilde{\mathrm{y}}}, \dot{\tilde{\mathrm{z}}}]$ be expressed as follows:

$$\dot{\tilde{\mathrm{x}}}(t_k) = \dot{\mathrm{x}} + \delta\dot{\mathrm{x}},\ \dot{\tilde{\mathrm{y}}}(t_k) = \dot{\mathrm{y}} + \delta\dot{\mathrm{y}},\ \dot{\tilde{\mathrm{z}}}(t_k) = \dot{\mathrm{z}} + \delta\dot{\mathrm{z}}$$

Linearising Eq. (4.70) around this nominal point, $d\dot{\mathrm{r}}_{\mathrm{INS}}^{i}$ is approximated as

$$
\begin{aligned}
d\dot{\mathrm{r}}^{i}_{\mathrm{INS}}(t_k) \approx & \frac{-(\mathrm{x}^{i}-\tilde{\mathrm{x}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\ddot{\mathrm{x}} + \frac{-(\mathrm{y}^{i}-\tilde{\mathrm{y}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\ddot{\mathrm{y}} + \frac{-(\mathrm{z}^{i}-\tilde{\mathrm{z}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\ddot{\mathrm{z}} + \\
& \frac{-(\dot{\mathrm{x}}^{i}-\dot{\tilde{\mathrm{x}}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\mathrm{x} + \frac{-(\dot{\mathrm{y}}^{i}-\dot{\tilde{\mathrm{y}}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\mathrm{y} + \frac{-(\dot{\mathrm{z}}^{i}-\dot{\tilde{\mathrm{z}}})}{\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}}\delta\mathrm{z}
\end{aligned}
\qquad (4.71)
$$

Differencing Eqs.(4.70) and (4.69) and comparing Eqs. (4.71) and (4.55) where ρ^{i}_{0}, $[\mathrm{x}_0,\mathrm{y}_0,\mathrm{z}_0]^{\mathrm{T}}$ and $[\dot{\mathrm{x}}_0,\dot{\mathrm{y}}_0,\dot{\mathrm{z}}_0]^{\mathrm{T}}$ are replaced by $\tilde{\mathrm{r}}^{i}_{\mathrm{INS}}$, $[\tilde{\mathrm{x}},\tilde{\mathrm{y}},\tilde{\mathrm{z}}]^{\mathrm{T}}$ and $[\dot{\tilde{\mathrm{x}}},\dot{\tilde{\mathrm{y}}},\dot{\tilde{\mathrm{z}}}]^{\mathrm{T}}$, respectively, the range rate difference measurement equation is given in vector form as follows:

$$
\dot{\mathbf{r}}_{\mathrm{GNSS}} - \dot{\tilde{\mathbf{r}}}_{\mathrm{INS}} = -\mathbf{H}_{\mathrm{LOSV}}\mathbf{C}^{\mathrm{e}}_{\mathrm{n}}\delta\mathbf{p}^{\mathrm{n}} - \mathbf{H}_{\mathrm{LOS}}\mathbf{C}^{\mathrm{e}}_{\mathrm{w}}\delta\mathbf{v}^{\mathrm{w}} + \mathbf{1}cd\dot{t} + \boldsymbol{\upsilon}_{\dot{\mathrm{r}}} \qquad (4.72)
$$

where $\mathbf{C}^{\mathrm{e}}_{\mathrm{w}}$ is the transformation from the wander frame to the ECEF frame and $\delta\mathbf{v}^{\mathrm{w}}$ is the velocity error state in the wander frame. From Eq. (4.30a), $\delta\mathbf{v}^{\mathrm{w}}$ is represented as follows:

$$
\delta\mathbf{v}^{\mathrm{w}} = -\delta\boldsymbol{\theta}\times\mathbf{v}^{\mathrm{w}} + \delta\mathbf{v}^{1} = \mathbf{T}^{\mathrm{v}}_{\mathrm{p}}\delta\mathbf{p}^{\mathrm{n}} + \delta\mathbf{v}^{1}
$$

where

$$
\mathbf{T}^{\mathrm{v}}_{\mathrm{p}} = \begin{bmatrix}
\frac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha) & -\frac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha) & 0 \\
\frac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha) & \frac{\mathrm{v}_{\mathrm{z}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha) & 0 \\
-\left(\frac{\mathrm{v}_{\mathrm{x}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha) + \frac{\mathrm{v}_{\mathrm{y}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha)\right) & \left(\frac{\mathrm{v}_{\mathrm{x}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha) - \frac{\mathrm{v}_{\mathrm{y}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha)\right) & 0
\end{bmatrix}
$$

Therefore, the range rate difference equation can be normalised as follows:

$$
\dot{\mathbf{r}}_{\mathrm{GNSS}} - \dot{\tilde{\mathbf{r}}}_{\mathrm{INS}} = -(\mathbf{H}_{\mathrm{LOSV}}\mathbf{C}^{\mathrm{e}}_{\mathrm{n}} + \mathbf{H}_{\mathrm{LOSS}}\mathbf{C}^{\mathrm{e}}_{\mathrm{w}}\mathbf{T}^{\mathrm{v}}_{\mathrm{p}})\delta\mathbf{p}^{\mathrm{n}} - \mathbf{H}_{\mathrm{LOS}}\mathbf{C}^{\mathrm{e}}_{\mathrm{w}}\delta\mathbf{v}^{1} + \mathbf{1}cd\dot{t} + \boldsymbol{\upsilon}_{\dot{\mathrm{r}}} \qquad (4.73)
$$

C. Normalised Relative Phase Difference Equation

From (4.36), given the nominal matrix $\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}$, Eq. (4.58) can be rewritten as follows:

$$
d\lambda\varphi^{i}_{j\text{-GNSS}} = (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}[\mathbf{I}+\partial\phi\times]\mathbf{s}^{\mathrm{n}/i} + \lambda N^{i}_{j} + v^{i}_{d\varphi-j}
$$

or $\quad d\lambda\varphi^{i}_{j\text{-GNSS}} - (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}\mathbf{s}^{\mathrm{n}/i} = (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}(\partial\phi\times)\mathbf{s}^{\mathrm{n}/i} + \lambda N^{i}_{j} + v^{i}_{d\varphi-j}$

Defining $d\lambda\varphi^{i}_{j\text{-INS}} = (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}\mathbf{s}^{\mathrm{n}/i}$, then $d\lambda\varphi^{i}_{j\text{-INS}}$ is the INS-derived magnitude of the projection of the baseline $\mathbf{b}^{\mathrm{b}}_{j}$ onto the LOS $\mathbf{s}^{\mathrm{n}/i}$.

$$d\lambda\varphi^{i}_{j-\mathrm{GNSS}} - d\lambda\varphi^{i}_{j-\mathrm{INS}} = (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}(\partial\phi\times)\mathbf{s}^{\mathrm{n}/i} + \lambda N^{i}_{j} + v^{i}_{d\varphi-j} \tag{4.74}$$

It is obvious that Eq. (4.74) can be used to determine the initial relative phase integer ambiguity N^{i}_{j}. Assume that the relative phase ambiguity is known, then the relative phase difference equation can be normalised as follows:

$$d\lambda\varphi^{i}_{j-\mathrm{GNSS}} - d\lambda\varphi^{i}_{j-\mathrm{INS}} = -(\mathbf{b}^{\mathrm{n}}_{j})^{\mathrm{T}}(\mathbf{s}^{\mathrm{n}/i}\times)\partial\phi + v^{i}_{d\varphi-j} \tag{4.75}$$

or $$d\lambda\varphi^{i}_{j-\mathrm{GNSS}} - d\lambda\varphi^{i}_{j-\mathrm{INS}} = (\mathbf{s}^{\mathrm{n}/i})^{\mathrm{T}}(\mathbf{b}^{\mathrm{n}}_{j}\times)\partial\phi + v^{i}_{d\varphi-j}$$

where $(\mathbf{b}^{\mathrm{n}}_{j})^{\mathrm{T}} = (\mathbf{b}^{\mathrm{b}}_{j})^{\mathrm{T}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n0}}$ is the baseline vector in the navigation frame.

When multiple satellites are observed with respect to one baseline, from Eq. (4.61), the relative phase difference equation for GNSS attitude determination can be written in vector form as follows:

$$\left[d\mathbf{\Phi}_{j-\mathrm{GNSS}} - d\mathbf{\Phi}_{j-\mathrm{INS}}\right]^{\mathrm{T}} = (\mathbf{S}^{\mathrm{n}})^{\mathrm{T}}(\mathbf{b}^{\mathrm{n}}_{j}\times)\partial\phi + \boldsymbol{\upsilon}^{\mathrm{T}}_{d\Phi-j} \tag{4.76}$$

where $d\mathbf{\Phi}_{j-\mathrm{GNSS}}$ and $d\mathbf{\Phi}_{j-\mathrm{INS}}$ are the GNSS measured and the INS computed row vectors of $d\mathbf{\Phi}_{j}$ in Eq. (4.61), respectively.

D. Normalised Position and Velocity Difference Equations

If the GNSS-based navigation states are available, the normalised measurement equations for data fusion filter can be obtained as follows:
The position difference equations are normalised as follows:

$$\begin{aligned}
&(\tilde{\varphi}_{\mathrm{INS}} - \tilde{\varphi}_{\mathrm{GNSS}})(\mathrm{R}_{\mathrm{a}} + \mathrm{h}) = \delta\varphi_{\mathrm{R}} + v_{\varphi\text{-GNSS}}\\
&(\tilde{\lambda}_{\mathrm{INS}} - \tilde{\lambda}_{\mathrm{GNSS}})(\mathrm{R}_{\mathrm{a}} + \mathrm{h})\cos(\varphi) = \delta\lambda_{\mathrm{R}} + v_{\lambda\text{-GNSS}}\\
&(\tilde{\mathrm{h}}_{\mathrm{INS}} - \tilde{\mathrm{h}}_{\mathrm{GPS}}) = \delta\mathrm{h} + v_{\mathrm{h\text{-}GNSS}}
\end{aligned} \tag{4.77}$$

and the normalised velocity difference equation is

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{INS}} - \mathbf{C}^{\mathrm{w}}_{\mathrm{n}}\tilde{\mathbf{v}}^{\mathrm{n}}_{\mathrm{GNSS}} = \mathbf{T}^{\mathrm{v}}_{\mathrm{p}}\delta\mathbf{p}^{\mathrm{n}} + \delta\mathbf{v}^{1} + \boldsymbol{\upsilon}_{\mathrm{v\text{-}GNSS}} \tag{4.78}$$

In addition, normalised Doppler radar navigation equations are given in Appendix G.

E. Normalised Attitude Difference Equation

When attitude information is available from the GNSS attitude determination or other aircraft sensor systems, the attitude difference equation can be normalised as follows:

$$\mathbf{Euler}_{\mathrm{INS}} - \mathbf{Euler}_{\mathrm{Aided}} = \mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}} \partial\phi + \boldsymbol{\upsilon}_{\mathrm{Att}} \tag{4.79}$$

where $\mathbf{Euler}_{\mathrm{Aided}}$ are the Euler angles provided by GNSS-based attitude determination or other attitude measurement systems, $\mathbf{Euler}_{\mathrm{INS}}$ are the Euler angles derived by the INS, $\mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}}$ is given by Eq. (4.42) and $\boldsymbol{\upsilon}_{\mathrm{Att}}$ is the attitude measurement noise.

When a magnetic heading sensor is available, the corresponding measurement is as follows:

$$\delta\psi_{\mathrm{w}} = -\tan(\theta)[\cos(\psi_{\mathrm{w}})\partial\phi_{\mathrm{x}} + \sin(\psi_{\mathrm{w}})\partial\phi_{\mathrm{y}}] - \partial\phi_{\mathrm{z}} - b_{\mathrm{M}} + \upsilon_{\mathrm{M}} \tag{4.80}$$

where b_{M} is the magnetic heading deviation and υ_{M} is the measurement noise. b_{M} can be modelled as the combination of a random constant and the first-order Markov process.

4.7 Summary

The aim of this Chapter has been to develop the navigation equations and error dynamic models of inertial systems and normalised measurement models of navaid systems, which are required to develop and simulate fault-tolerant, multisensor-based aircraft navigation systems. The following activities have been described in this Chapter:

1. Mechanisation of the wander-azimuth strapdown inertial navigation equations, which allow aircraft to fly in the high latitude regions.
2. Development of the error dynamic models for strapdown inertial system. These models are used not only for analysis and evaluation of the error behaviour of inertial systems, but more importantly, in the design and development of the data fusion filter developed in Chapter 6. These models also provide the basis of dynamic calibration and in-flight correction of inertial sensor systems.
3. Presentation of the normalised measurement and navigation equations of GNSS. These normalised measurement models are used in the design of

multisensor data fusion filters. It should be noted that the normalisation of the sensor measurement equations can simplify the design of multisensor navigation systems and development of sensor/system failure detection and isolation algorithms. In particular, these different forms of normalised GNSS equations further explain how GNSS measurements are used in multisensor data fusion for GNSS/inertial hybridised navigation systems.

4. Derivation of the error correction and control equations for the inertial sensor systems and the navigation states.

These achievements provide the necessary background and fundamental theory for the design and development of multisensor-based aircraft navigation systems and also for the simulation and evaluation of different-grade inertial sensor systems in this thesis.

Chapter 5

SENSOR NETWORK TOPOLOGY AND FAILURE DETECTION METHODS

5.1 Introduction

This chapter introduces sensor system network topologies and develops sensor-level data fusion methods. The main purposes of the sensor-level data fusion are to provide highly reliable and accurate sensor data for subsequent data fusion modules and also reconfigure sensor network systems if some sensors failed. These create the fundamentals for the design of fault-tolerant navigation systems and the achievement of reliability and integrity of aircraft navigation systems.

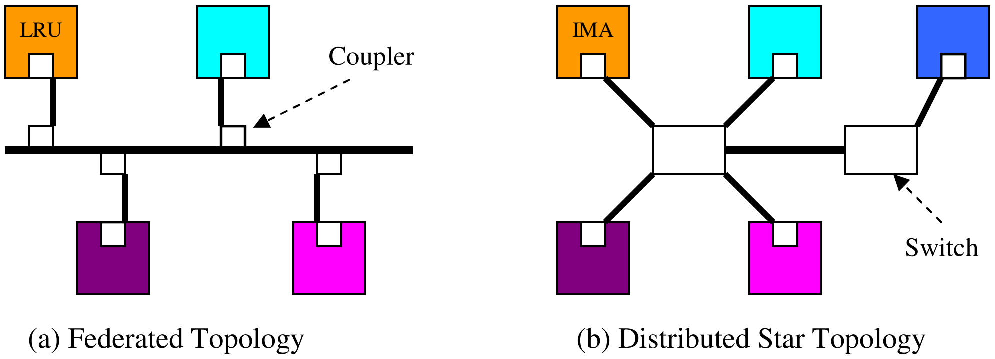
Section 5.2 presents a distributed inertial network architecture and develops optimal redundant inertial system configurations in terms of measurement accuracy, reliability and failure detection capability. The error models and calibrations of skewed redundant IMU (SRIMU) systems are considered in Section 5.3. Section 5.4 introduces the basic generalised likelihood ratio test (GLRT) method for the detection of sensor/system failures. SRIMU error compensation filters are developed to improve the performance of the basic GLRT methods in Section 5.5. Moving-window detection methods are presented to enhance the capability of the GLRT methods for the detection of drift sensor failures in Section 5.6. A summary is given in Section 5.7.

5.2 Sensor System Network Topology

Measurement information provided by various navigation sensor systems can be independent, redundant, complementary or cooperative. For example, gyroscope set and accelerometer set, each individually providing independent measurements, are integrated in an IMU to provide complementary and cooperative information that are used to derive the navigation states. Multiple IMUs then offer redundant inertial measurements. AHRS and Doppler radar together present cooperative information while GPS and IMU are complementary navigational sensor systems. By structuring different types (also known as dissimilar) of and redundant (also known as similar) navigation sensor systems in a rational sensor network topology, these various types of measurement information can be combined to achieve the required navigation performance and to provide the inertial vector state information required by other avionics systems.

Sensor system network topology is a collection of various sensor systems and explains logical relationships and physical interconnections between these sensor systems. There are two typical avionics system architectures today widely used in civil and military aircraft of all types, known as the federated and integrated modular avionics (IMA) architectures[120-122]. The federated avionics systems have a topology architecture, as shown in Figure 5.1a where subsystems are encapsulated in various special-purpose hardware units, known as line replaceable units (LRU), to implement their individual avionic functions, such as navigation and flight control. These LRUs share the use of common data buses for data transmission between themselves. For example, ARINC 429 (single-transmitter multiple-receiver) and 629 (multiple access data bus) topology buses are usually used for the federated architecture.

Although the federated architecture has its inherent fault tolerance, it does not efficiently make use of today’s powerful computer processing modules and needs to develop costly special-purpose hardware systems. With technology advancements in avionics integration and modularity designs of hardware and software systems, the concept of integrated modular avionics (IMA) has been presented for the purpose of developing more reliable and cost-effective, modular and highly integrated avionics systems. Various IMA implementation architectures are described in ARINC 651, ‘Design Guidance for Integrated Modular Avionics’. Instead of single-function black boxes (LRUs), an IMA cabinet contains several line replaceable modules (LRM) and avionics functions are implemented with common, programmable modules which are software-reconfigured to process many different LRU functions. Therefore, LRMs are shared resources for different avionics functions. Several IMA cabinets can be interconnected to sensor systems by high speed data buses to form a distributed system for performing all avionics functions on the aircraft. A typical IMA star topology is shown in Figure 5.1(b) where data transmission between IMAs is through switch units. Although ARINC 629 topology bus is used in some current IMA-based avionics systems, its main limitations are lower data rate and expensive components. Future IMA architecture will be based on faster commercial networking data buses, such as full duplex Ethernet (FDX).



(a) Federated Topology

(b) Distributed Star Topology

Figure 5.1 Avionics topology Architectures

With the introduction of high speed avionics data buses and integrated modular avionics systems and the advent of low-cost, small-size, low-mass navigation sensors, high-speed and embedded microprocessors, it is feasible to install redundant inertial sensors in a single IMU box using a non-orthogonal configuration in order to improve the system reliability and to reduce the cost, size and mass of aircraft navigation systems. In this thesis, two forms of sensor network topologies used in the design of aircraft multisensor navigation systems are discussed as follows:

- A distributed sensor system architecture
- A clustered sensor system architecture

In the distributed sensor system architecture topology, multiple sensor systems are spatially distributed at different locations in an aircraft for the implementation of different functions. This architecture topology is compatible with the new generation of avionics IMA and can enhance the fault tolerance and the survivability of aircraft navigation systems.

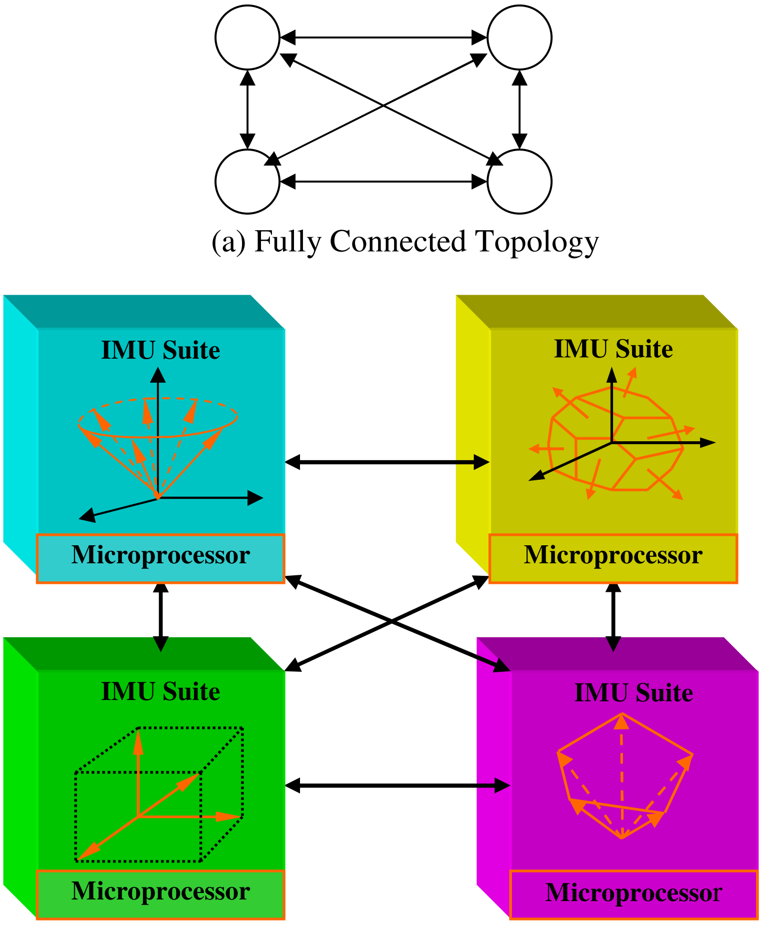
In the clustered architecture topology, multiple inertial sensors are assembled into a single box to provide redundant inertial information. This architecture grants fault tolerance and is usually used to create redundant inertial measurement units. Multiple clustered sensor systems located at different locations in an aircraft form a distributed sensor network system.

5.2.1 Distributed Sensor System Architecture

Distributed sensor system architectures may have different topological forms dependent on the data flow control and communication between the nodes of sensor system networks. Individual data fusion filtering algorithms have to be developed to adapt to those diverse architectures, as identified in Sections 2.2 and 2.3.

In this thesis, a distributed inertial network architecture is proposed, as shown in Figure 5.2 where each node represents an individual sensing place and consists of an IMU suite and an embedded microprocessor module. This architecture is a fully connected topology and allows multi-source sensor data to be fused at each network node. Each IMU suite can be integrated with other navaid systems. Each node is assumed to be in communication with others so that information from each node can be shared in the network architecture. The node located at the aircraft centre of gravity (cg) is a master node, also referred to as cg node, and others are local nodes, known as slave modes. The data fusion filter located at the cg (known as the cg filter) provides the navigation states and the cg inertial state vector while the data fusion filters located at slave nodes (known as the slave filters) provide the local inertial state vector information.

Motivation to investigate this distributed inertial network architecture is based on two critical necessities; the inertial sensor system is an essential aircraft sensor system that provides vital inertial information for all safety-critical avionic systems, including navigation system and flight control system. Additionally, fault tolerance of aircraft navigation system is primarily obtained from redundant inertial systems.



(a) Fully Connected Topology

(b) Physical Interconnections

Figure 5.2 Distributed Inertial Network Architecture

This distributed inertial system network affords the following advantages:

- Fault tolerance and robustness to sensor/system failures. Data fusion algorithms are designed so that the failure of any node or element of the node will not lead to the degradation of the performance of aircraft navigation system. Moreover, the degradation of the performance of the slave filter located at the failed node will be gradual.
- Flexibility. It is easy to add and/or remove one or more sensor systems in and from the distributed system network.

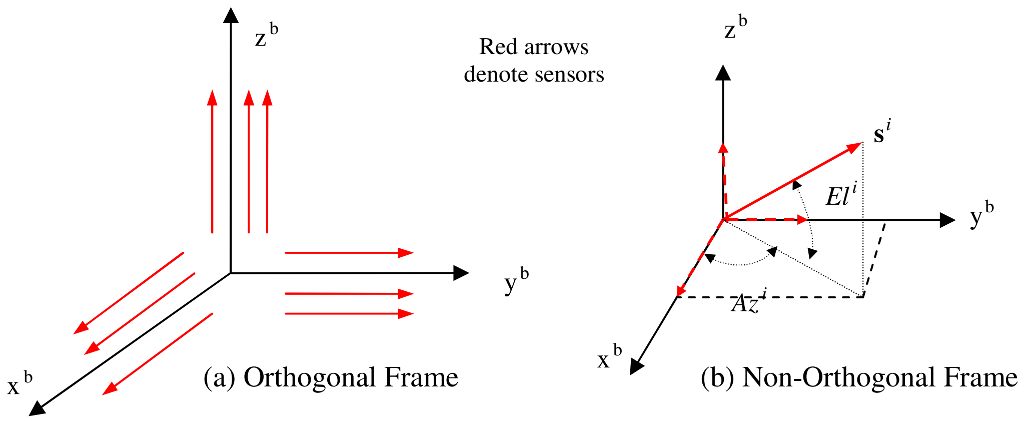
- Highly reliable *cg* state estimation. The *cg* data fusion filter combines all local estimates and its own estimate to obtain the aircraft *cg* motion states, which are used to support aircraft navigation, flight control and guidance, and other functions that require the *cg* referenced data
- Accurate local state estimation. Local data fusion filter located at each mode fuses all measurements from all healthy sensor systems to afford optimal estimates of the local states that are used to support the stabilisation of various avionics system platforms and local motion compensation.
- Automatic alignment. Because information is shared at all nodes, the distributed data fusion filters can autonomously use the local estimates at a node of high quality IMU to dynamically correct and align low quality IMUs at other nodes. Therefore, traditional inertial system alignment algorithms, for example, fine alignments and transfer alignments, are no longer necessary in distributed inertial network systems. In traditional alignment methods, aircraft is usually requested to perform some specified manoeuvres, which can increase risk especially to military aircraft and pilots. The elimination of the traditional alignment procedures allows aircraft to perform free flight and manoeuvres.

Distributed data fusion algorithms and their significant advantages will be discussed in Chapter 6.

5.2.2 Clustered Sensor Topology

The clustered sensor topology has different configurations. Two approaches to the configuration of a redundant IMU system have been suggested in the past[71][72]. One is an orthogonal configuration shown in Figure 5.3(a) where the sensing axes of redundant inertial sensors are orthogonal or parallel with respect to the body axes. The other uses a non-orthogonal configuration relative to the body axes shown in Figure 5.3(b), referred to as skewed redundant IMU (SRIMU) configurations. In the orthogonal configuration, the inertial measurement sensed by one sensor mounted on one axis is independent of other measurements sensed by other sensors mounted on other axes. Therefore, the orthogonal IMU measurements are decoupled along the orthogonal axes.

In a non-orthogonal configuration, the measurement sensed by one sensor can be decomposed into three components along the orthogonal axes, red dash arrows shown in Figure 5.3(b). Therefore, the measured states are coupled with each other in the SRIMU measurements. This nature allows fewer sensors to be used in an SRIMU configuration in order to achieve system performance equivalent to the orthogonal IMU system. Although the orthogonal IMU system is a conventional configuration, it is not the most efficient way to exploit the benefits of redundant sensor systems in a fault-tolerant navigation system. The orthogonal configuration has been used in traditional fault-tolerant navigation systems and also appears in multisensor fusion navigation systems with distributed sensor network to simplify the system design.



(a) Orthogonal Frame

(b) Non-Orthogonal Frame

Figure 5.3 Sensor Installation Orientation

SRIMU systems can most effectively make use of redundant measurements provided by multiple sensors and have various configuration geometries dependent on the number of sensors. The typical configuration geometries are based on regular polyhedrons in order to simplify the engineering implementation. Several geometries commonly used in redundant sensor configurations are summarised in Table 5-1.

Table 5-1 Polyhedrons in Redundant Sensor Configurations

Polyhedron	Number of Faces	Min Number of Sensors for Redundancy
Cube	6	≥4
Cone (Pyramid)	≥4	≥4
Dodecahedron	12	6

5.2.3 Criteria for Optimal SRIMU Configurations

In an SRIMU configuration, the orientation of each instrument axis is defined by its azimuth and elevation angles with respect to an orthogonal reference frame, such as the body frame. Let each axis of the instrument frame be presented by a unit vector $\mathbf{s}^i$ along the sensing direction of sensor i, the unit vector can be defined in the orthogonal reference frame by

$$\mathbf{s}^i = \cos(El^i)\cos(Az^i)\mathbf{i} + \cos(El^i)\sin(Az^i)\mathbf{j} + \sin(El^i)\mathbf{k} \tag{5.1}$$

where the bold symbols $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are three unit vectors along the corresponding axes of the reference frame $(\mathrm{x}^\mathrm{b}, \mathrm{y}^\mathrm{b}, \mathrm{z}^\mathrm{b})$, the superscript i denotes a sensor and its sensing axis, El^i and Az^i are the elevation and azimuth angles of the instrument axis i with respect to the reference frame, as shown in Figure 5.3(b).

Provided that an SRIMU system encloses n sensors, identified by $1, 2, 3, \cdots, n$, the failure-free measurement equations of the SRIMU system can be formulated as follows:

$$\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} = \begin{bmatrix} \mathbf{i}\circ\mathbf{s}^1 & \mathbf{j}\circ\mathbf{s}^1 & \mathbf{k}\circ\mathbf{s}^1 \\ \mathbf{i}\circ\mathbf{s}^2 & \mathbf{j}\circ\mathbf{s}^2 & \mathbf{k}\circ\mathbf{s}^2 \\ \vdots & \vdots & \vdots \\ \mathbf{i}\circ\mathbf{s}^n & \mathbf{j}\circ\mathbf{s}^n & \mathbf{k}\circ\mathbf{s}^n \end{bmatrix} \begin{bmatrix} \omega_\mathrm{x} \\ \omega_\mathrm{y} \\ \omega_\mathrm{z} \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{5.2a}$$

or in vector form $\mathbf{m} = \mathbf{H}\boldsymbol{\omega} + \boldsymbol{\upsilon}$ (5.2b)

where $\omega_\mathrm{x}, \omega_\mathrm{y}$ and ω_z are three measured quantities, such as accelerations or angular rates in the body frame, m_i is the measurement of sensor i and v_i is a Gaussian white noise with a zero-mean value and standard deviation σ_i. The symbol $\circ$ presents the operation of dot product of two vectors. The matrix $\mathbf{H}$ is known as the measurement or design matrix and describes the configuration of an SRIMU system.

Applying a weighted least-squares estimator to Eq. (5.2b), the estimate of the measured state vector $\hat{\boldsymbol{\omega}}$ is given by

$$\hat{\boldsymbol{\omega}} = (\mathbf{H}^\mathrm{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathrm{T}\mathbf{W}\mathbf{m} = \mathbf{C}^\mathrm{b}_\mathrm{instru}\mathbf{m} \tag{5.3}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{C}^\mathrm{b}_\mathrm{instru}$ is referred to as the transformation matrix from the inertial instrument frame to the body frame.

Defining the estimate error vector $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$, then

$$\begin{aligned}\tilde{\boldsymbol{\omega}} &= \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} = \boldsymbol{\omega} - (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{m} \\ &= \boldsymbol{\omega} - (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}(\mathbf{H}\boldsymbol{\omega} + \boldsymbol{\upsilon}) \\ &= \boldsymbol{\omega} - (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}\boldsymbol{\omega} - (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\boldsymbol{\upsilon} \\ &= -(\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\boldsymbol{\upsilon}\end{aligned} \tag{5.4}$$

Therefore, the estimate error is the normal distribution and the covariance matrix of the estimate errors according to the covariance transfer law is given by

$$\mathrm{Var}(\tilde{\boldsymbol{\omega}}) = \mathrm{E}[(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})^{\mathrm{T}}] = (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{R}\mathbf{W}\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1} \tag{5.5}$$

where $\mathbf{R} = \mathrm{E}(\mathbf{v}\mathbf{v}^{\mathrm{T}})$ is the noise covariance matrix.

To simplify the analysis of performance of an SRIMU configuration, assume that all of sensor noises are independent and that the standard deviation of the noise for each sensor measurement is identical σ_v, and if the weight matrix $\mathbf{W}$ is taken as the inverse of $\mathbf{R}$, then the covariance matrix of the estimate error becomes

$$\mathrm{Var}(\tilde{\boldsymbol{\omega}}) = (\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1} = \sigma_v^2(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1} \tag{5.6a}$$

or is represented by the following normalised form

$$\Sigma = \frac{\mathrm{Var}(\tilde{\boldsymbol{\omega}})}{\sigma_v^2} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1} \tag{5.6b}$$

The probability density function of the estimate error can be given by

$$f_{\tilde{\omega}}(\mathbf{x}) = (2\pi)^{-3/2}|\Sigma|^{-1/2}\exp(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\Sigma^{-1}\mathbf{x})$$

Then, the locus of the point **x** is determined by

$$\mathbf{x}^{\mathrm{T}}\Sigma^{-1}\mathbf{x} = K$$

This represents an error ellipsoid with a surface of constant likelihood. For any *K*, the volume of this ellipsoid is given by[123]

$$V = \frac{4}{3}K^{3/2}\pi\sqrt{|\Sigma|}$$

From the analysis above, the smaller the volume of this ellipsoid, the smaller the estimate errors, and the performance of navigation systems with various SRIMU configurations can be determined by $\sqrt{|\Sigma|}$.

Defining a performance index (*PI*) as

$$PI = \sqrt{|\Sigma|} = \sqrt{Det[(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}]} \tag{5.7}$$

This equation can be used to determine the azimuth and elevation angles of each sensor to construct an optimal SRIMU configuration. If the square root of the trace of the normalised covariance matrix is selected as a criterion to optimise an SRIMU configuration, known as the geometric dilution of precision (*GDOP*), then

$$GDOP = \sqrt{tr[(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}]} \tag{5.8}$$

On the basis of the criterion of minimum *GDOP*, Sturza[78] analyses the optimal installation angles for several cone configurations. However, this criterion cannot be applied to non-cone SRIMU configurations. To evaluate the optimal performance of non-cone SRIMU configurations, the estimate error variances of the measured states in the body frame from Eq. (5.4) can be formulated as follows.

$$\sigma_{\omega}^{2}(i) = \sum_{j=1}^{n} \mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}(i,j)^{2}\sigma_{j}^{2}, \quad i = \mathrm{x, y, z} \tag{5.9}$$

Based on the assumption that all measurement noises have an identical variance $\sigma_{v}^{2} = \sigma_{j}^{2}$, a normalised error variance is given by

$$\sigma_{N}^{2}(i) = \frac{\sigma_{\omega}^{2}}{\sigma_{v}^{2}} = \sum_{j=1}^{n} \mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}(i,j)^{2}, i = \mathrm{x, y, z} \tag{5.10}$$

where $\mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}(i,j)$ is the corresponding element of $\mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}$.

Accordingly, the criterion for determining the optimal SRIMU installation angles is based on the allocation of the uncertainty of SRIMU measurement to three orthogonal reference axes, usually the body axes. For example, to precisely sense aircraft motion along a specific body-axis direction, the criterion for minimising the corresponding $\sigma_{N}(i)$ can be used to determine the SRIMU installation angles. To allocate the uncertainty of SRIMU measurement equally to three body axes, then the following criteria

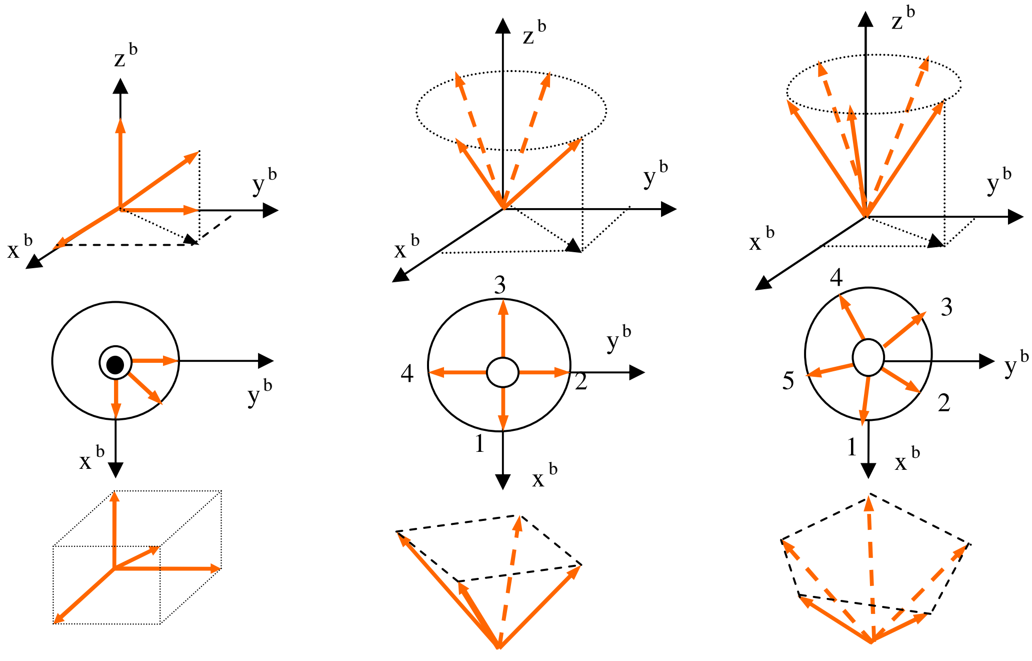
$$\sigma_{N}(\mathrm{x}) = \sigma_{N}(\mathrm{y}) = \sigma_{N}(\mathrm{z}) \tag{5.11}$$

can be selected to determine the SRIMU installation angles.

Based on these optimal criteria given in Eqs. (5.7) to (5.11), several SRIMU configurations shown in Figure 5.4 are evaluated and the results are summarised in

Tables 5-2 and 5-3. If one sensor in the 5-sensor cone configuration in Figure 5.4(c) is aligned with an orthogonal axis, Figure 5.4(c) will be degraded into the similar configuration shown in Figure 5.4(b).

However, the above criteria cannot guarantee that sensor failure detection and isolation methods based on these optimal SRIMU configurations also have optimal performance. It will be revealed in the development of sensor failure detection and isolation methods that the initial installation azimuth angle of the first sensor in a symmetrical SRIMU configuration should not be zero. Accordingly, this requirement has to be considered as one restriction to construct a skewed redundant IMU system.



(a) 4-Sensor Cube (b) 4-Sensor Cone (c) 5-Sensor Cone

⦿ One sensor aligned with the

○ No sensor aligned with the reference

Figure 5.4 Redundant Sensor Configurations

Comparing the fourth and fifth columns of Tables 5-2 and 5-3, separately, if sensor failures occurred, optimal configurations many not obtain better measurement accuracy in comparison with a non-optimal configuration. Therefore, the selection of an SRIMU configuration is a tradeoff between failure detection performance and measurement accuracy under conditions of no sensor failures and sensor failures.

Table 5-2 Comparisons of Two 4-Sensor SRIMU Configurations

Configuration	Accuracy of State Estimates	Accuracy Degradation	Az = El = 45^0
Cube $El = 35.2644^0$ $Az = 45^0$	$\sigma_N(i) = 0.9129$ $i = \text{x, y, z}$	When any one on three orthogonal axes has failed, $\sigma_N(i) = 2.2361,\ i = \text{x, y, z}$ $\sigma_N(j) = 1.0,\ j \neq i$	$\sigma_N(\text{x}) = 0.9354$ $\sigma_N(\text{y}) = 0.9354$ $\sigma_N(\text{z}) = 0.8660$
Cone $El = 35.2644^0$ Az = any	$\sigma_N(i) = 0.8660$ $i = \text{x, y, z}$	When any one has failed $\sigma_N(\text{z}) = 1.2247$ Dependent on which sensor, $\sigma_N(\text{x}) = 1.5 \text{ or } 0.8660$ $\sigma_N(\text{y}) = 0.8660 \text{ or } 1.5$ In the case of $Az = 45^0$ $\sigma_N(i) = 1.2247,\ i = \text{x, y, z}$	$\sigma_N(\text{x}) = 1.0$ $\sigma_N(\text{y}) = 1.0$ $\sigma_N(\text{z}) = 0.7071$

Table 5-3 Comparisons of Several 5-Sensor SRIMU Configurations

Configuration	Accuracy of State Estimates[1]	Accuracy Degradation when any one has failed	Azimuth =45° When any sensor on cone failed
Cone $El = 35.2644^0$ Az = any	$\sigma_N(i) = 0.7746$ $i = \text{x, y, z}$ $DGOP = 1.3416$	Max $\sigma_N = 1.0954$ Min $\sigma_N = 0.7746$ $DGOP = 1.6432$	
Cone + Spin[2] $El = 24.0929^0$ Az = any	$\sigma_N(i) = 0.7746$ $i = \text{x, y, z}$ $DGOP_\text{z} = 1.3416$	Max $\sigma_N = 1.2248$ Min $\sigma_N = 0.7746$ $DGOP = 1.6432$	$\sigma_N(\text{x}) = 0.9874$ $\sigma_N(\text{y}) = 0.9874$ $\sigma_N(\text{z}) = 0.8660$ $DGOP = 1.6432$
Cone + x-Axis[3] $El = 38.1876^0$ Az = any	$\sigma_N(\text{x}) = 0.6688$ $\sigma_N(\text{y}) = 0.8996$ $\sigma_N(\text{z}) = 0.8087$ $DGOP_\text{x} = 1.3823$	Dependent on the failed sensor Max $\sigma_N = 1.5582$ Min $\sigma_N = 0.6688$ Min $DGOP_\text{x} = 1.5076$ Max $DGOP_\text{x} = 2.0454$	$\sigma_N(\text{x}) = 0.7862$ $\sigma_N(\text{y}) = 1.1699$ $\sigma_N(\text{z}) = 1.0517$ $DGOP_\text{x} = 1.7586$
Cone + y-Axis[4] $El = 38.1876^0$ Az = any	$\sigma_N(\text{x}) = 0.8996$ $\sigma_N(\text{y}) = 0.6688$ $\sigma_N(\text{z}) = 0.8087$ $DGOP_\text{y} = 1.3823$	The same as above	$\sigma_N(\text{x}) = 1.1699$ $\sigma_N(\text{y}) = 0.7862$ $\sigma_N(\text{z}) = 1.0517$ $DGOP_\text{y} = 1.7586$

[1] GDOP is also used to describe the geometry of redundant inertial sensor configurations as in satellite constellations.

[2] One of sensors is aligned with the spin axis of a cone configuration, or the z-axis of the body frame.

[3] One of sensors is aligned with the x-axis of the body frame.

[4] One of sensors is aligned with the y-axis of the body frame .

5.2.4 Reliability Analysis of SRIMU Configurations

As stated in the above section, the use of SRIMU configurations can obtain the fault tolerance and reliability of aircraft navigation systems. To compare reliabilities of various configuration SRIMU systems, assume that all sensors are single degree-of-freedom sensors and the failure rate λ of each sensor is constant and identical for each type of inertial sensor. Then the reliability function of inertial sensor is given by

$$R(t) = e^{-\lambda t} \tag{5.12}$$

and the MTBF(mean time between failures) is defined as

$$MTBF = \int_0^\infty R(t)dt = \frac{1}{\lambda} \tag{5.13}$$

The reliability of the redundant sensor system is given by the following equation

$$R_{sensor}(t) = [R\ (t)]^n + C_n^{n-1}[R(t)]^{n-1}[1 - R(t)] + ... + C_n^{n-m}[R(t)]^{n-m}[1 - R(t)]^m \tag{5.14}$$

where

$$C_n^m = \frac{n!}{(n-m)!m!}$$

n is the number of sensors in the redundant configuration and m is the number of allowable failure sensors in the redundant system.

Therefore, the reliability of an SRIMU system is given by

$$R_{SRIMU}(t) = R_{Gyro}(t) \cdot R_{Accel}(t) \tag{5.15}$$

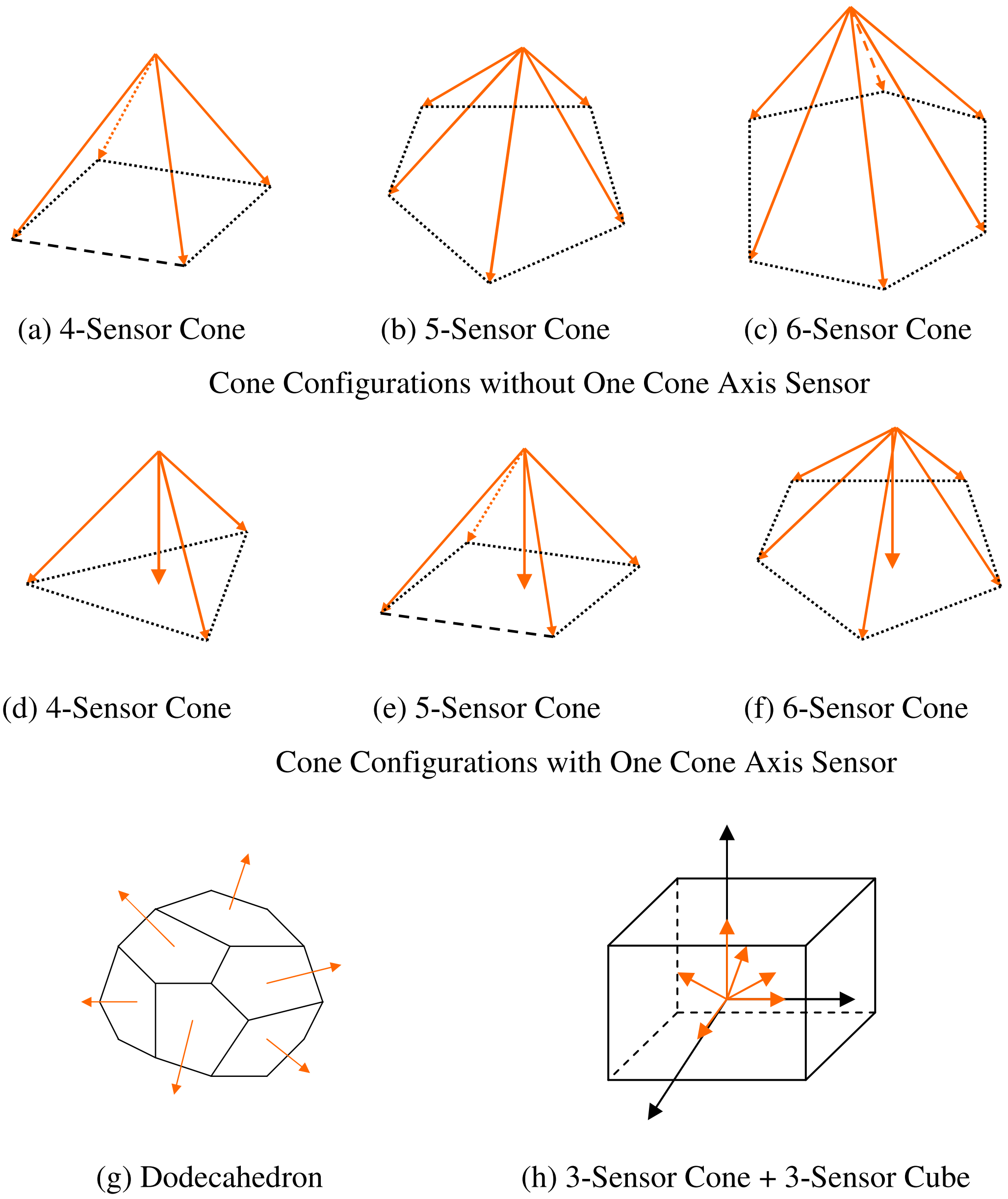
For the orthogonal configuration in a conventional IMU, the reliability and MTBF are given by

$$R_{3-Gyro}(t) = e^{-3\lambda t}$$

$$MTBF_{3-Gyro} = \frac{1}{3\lambda}$$

For the configurations shown in Figure 5.5, the reliability figures and MTBF values are computed and normalised with respect to the $MTBF_{3-Gyro}$ value. The results are summarised in Table 5-4 where the reliability increases with the ratio. From inspection of Table 5-4, the reliability of an SRIMU configuration depends on the number of redundant sensors and the failure rate of sensor. The accuracy of SRIMU measurements relies on the sensor installation configurations.

In the dodecahedron configuration shown in Figure 5.5(g), each of six sensors is separately mounted along the axis of each pair of parallel faces. The configuration in Figure 5.5(h) is a combination of 3-sensor cone and 3-sensor cube configurations.



(a) 4-Sensor Cone (b) 5-Sensor Cone (c) 6-Sensor Cone

Cone Configurations without One Cone Axis Sensor

(d) 4-Sensor Cone (e) 5-Sensor Cone (f) 6-Sensor Cone

Cone Configurations with One Cone Axis Sensor

(g) Dodecahedron (h) 3-Sensor Cone + 3-Sensor Cube

Figure 5.5 SRIMU Configurations

Table 5-4 Reliabilities of Several SRIMU Configurations

Sensor Configuration	Elevation (deg)	Azimuth (deg)	MTBF	Ratio	*PI*	Fault Tolerance[5]
3-sensor orthogonal			$\frac{1}{3\lambda}$	1		
4-Sensor Cube as in Figure 5.4(a)	35.264	45	$\frac{7}{12\lambda}$	1.75	0.7071	Fail Safe
4-Sensor Cone as in Figure 5.5(a)	35.264	45 90	$\frac{7}{12\lambda}$	1.75	0.6495 0.6495	Fail Safe
4-Sensor Cone as in Figure 5.5(d)	19.472	120	$\frac{7}{12\lambda}$	1.75	0.6495	Fail Safe
5-Sensor Cone as in Figure 5.5(b)	35.264	72	$\frac{47}{60\lambda}$	2.35	0.4648	Fail Op/Fail Safe
5-Sensor Cone as in Figure 5.5(e)	24.092	90	$\frac{47}{60\lambda}$	2.35	0.4648	Fail Op/Fail Safe
6-Sensor Cone as in Figures 5.5(c), Anyone sensor failed, 2 adjacent sensors failed, 2 skipping sensors failed	35.264	60	$\frac{57}{60\lambda}$	2.86	0.3536 0.5000 0.9487 0.7071	Fail Op/Fail Op/Fail Safe Fail Op/Fail Safe Fail Safe Fail Safe
6-Sensor Cone as in Figures 5.5(f), Anyone sensor failed, Any two sensors failed,	26.564	72	$\frac{57}{60\lambda}$	2.86	0.3536 0.5000 0.7906	Fail Op/Fail Op/Fail Safe Fail Op/Fail Safe Fail Safe
Dodecahedron as in Figures 5.5(g), One sensor failed, Any two sensors failed	31.717	90	$\frac{57}{60\lambda}$	2.86	0.3536 0.5000 0.7906	Fail Op/Fail Op/Fail Safe Fail Op/Fail Safe Fail Safe
6-Sensor Cube as in Figure 5.5(h), Anyone sensor failed, Any two sensors (in the same set) failed, Two sensors (in different sets) failed	35.264	120	$\frac{57}{60\lambda}$	2.86	0.3536 0.5000- 0.7071 0.7071- 1.2247	Fail Op/Fail Op/Fail Safe Fail Op/Fail Safe Fail Safe Fail Safe

5.3 SRIMU Calibration

Consider the main sensor errors, including drifts, sensor misalignments and scalar factor errors, the compensated SRIMU measurement model corresponding to Eq. (5.2b) becomes

$$\mathbf{m} = \mathbf{H}\boldsymbol{\omega} + \boldsymbol{\delta}_{\mathrm{D}} + \mathbf{S}_{\mathrm{SF}}\mathbf{m} + \mathbf{G}\mathbf{m} + \boldsymbol{\upsilon} \tag{5.16}$$

[5] Fault tolerance in this table is characterised by Fail safe and Fail Operational. Fail safe means that the sensor system can issue alarm information and interrupts its work if one sensor has failed. Fail operational means that the sensor system continues its work even if one sensor has failed.

where $\boldsymbol{\delta}_{\mathrm{D}}$ is an n-dimensional sensor drift vector, $\mathbf{S}_{\mathrm{SF}}$ is an $n \times n$-dimensional diagonal matrix, defining scale factor errors and $\mathbf{G}$ is an $n \times n$-dimensional matrix, containing sensor misalignments. These errors can be estimated and corrected by using external aiding measurements in a multisensor navigation system to improve the accuracy of the navigation system and also the performance of sensor failure detection and isolation functions. The misalignment matrix $\mathbf{G}$, which has a well-known formulation for the orthogonal sensor configuration, has to be redefined for an SRIMU configuration.

The misalignments between the designed installation axis and the actual sensor sensing axis can be represented by two small disturbances of azimuth and elevation angles δ_{Az^i} and δ_{El^i}, as shown in Figure 5.6 where $\mathbf{s}^i$ and $\tilde{\mathbf{s}}^i$ are unit vectors along the designed and actual instrument axes. The practical installation angles are defined as follows:

$$El^i = El_0^i + \delta_{El^i} \tag{5.17}$$

$$Az^i = Az_0^i - \delta_{Az^i} \tag{5.18}$$

where Az_0^i and El_0^i are the designed installation azimuth and elevation angles of the instrument axis i, respectively, as shown in Figure 5.3(b).

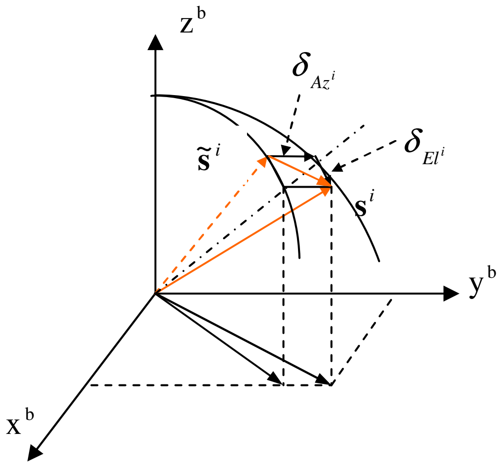


Figure 5.6 Definitions of Sensor Misalignments

Furthermore, the perturbation form of Eq. (5.2b), caused by sensor misalignments, can be expressed as

$$\mathbf{m} + \Delta\mathbf{m} = (\mathbf{H} + \Delta\mathbf{H})\boldsymbol{\omega} + \boldsymbol{\upsilon} \tag{5.19}$$

Therefore,

$$\Delta\mathbf{m} = \Delta\mathbf{H}\boldsymbol{\omega} \tag{5.20}$$

From Appendix H, the total SRIMU measurement error in the instrument frame is

$$\Delta\boldsymbol{\omega}^{\mathrm{instru}} = \boldsymbol{\delta}_{\mathrm{D}} + \mathrm{Diag}(\mathbf{m})\boldsymbol{\delta}_{\mathrm{SF}} + \mathrm{Diag}(\Pi)\boldsymbol{\delta}_{\mathrm{Az}} + \mathrm{Diag}(\Sigma)\boldsymbol{\delta}_{\mathrm{El}} \tag{5.21}$$

$$\Pi = \boldsymbol{\Theta}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\mathbf{m}\,,\ \Sigma = \boldsymbol{\Gamma}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\mathbf{m}$$

where $\boldsymbol{\delta}_x (x = Az,\ El)$ are the misalignment angle vectors, $\boldsymbol{\delta}_{\mathrm{SF}}$ is an n-dimensional scale factor error vector, $\mathbf{Diag}(\mathbf{m})$ is a diagonal matrix consisting of the SRIMU measurement vector $\mathbf{m}$, $\mathbf{Diag}(\Pi)$ is a diagonal matrix consisting of the vector Π , $\mathbf{Diag}()$ is a diagonal matrix consisting of its element, the elements of the matrix $\boldsymbol{\Gamma}$ correspond to the coefficients of the elevation misalignments in $\Delta\mathbf{H}$ and the elements of $\boldsymbol{\Theta}$ are the coefficients of the azimuth misalignments in $\Delta\mathbf{H}$.

These SRIMU error terms are normally estimated by means of appropriate data fusion filters in multisensor navigation systems. As analysed in Section 4.3, however, the sensor error states are formulated in the navigation frame to simplify the system error models. For example, ∇^{w} appearing in the velocity error model of Eq. (4.31) and Δ^{w} in the tilt error model of Eq. (4.40) are described in the navigation frame. Accordingly, $\Delta\boldsymbol{\omega}^{\mathrm{instru}}$ has to be transferred into $\Delta\boldsymbol{\omega}^{\mathrm{n}}$ as follows:

$$\begin{aligned}\Delta\boldsymbol{\omega}^{\mathrm{n}} &= \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\boldsymbol{\delta}_{\mathrm{D}} + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\mathrm{Diag}(\mathbf{m})\boldsymbol{\delta}_{\mathrm{SF}} \\ &\quad + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\mathbf{Diag}(\Pi)\boldsymbol{\delta}_{Az} + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{instru}}\mathbf{diag}(\Sigma)\boldsymbol{\delta}_{El}\end{aligned} \tag{5.22}$$

This equation can be used to determine the sub-matrixes of the system state transition matrix in data fusion filter, which are related to the SRIMU sensor error terms. Once the SRIMU sensor errors are estimated, Eq. (5.16) can also be used as the calibration equation to correct the SRIMU measurements.

The main advantage of SRIMU configurations is that the minimum redundant sensors are needed in order to form a fault-tolerant navigation system, decreasing the size and weight of the SRIMU system. Fault tolerance can be achieved by the design of reliable failure detection isolation algorithms. FDI problems will be discussed in the following sections

5.4 Basic GLRT Method

The generalised likelihood ratio test (GLRT) approach to the detection of jump change in linear systems is proposed by Willsky and Jones[85]. Since that, various improved versions of GLRT algorithms have been developed to detect measurement failures in GPS and SRIMU systems[75-80]. Basic GLRT method for SRIMU FDI is introduced as follows.

From the SRIMU measurement model given by Eq. (5.2b) and the GNSS measurement model given by Eq. (4.54), these measurement equations, in normal operating conditions, can be unified into a normalised form as follows:

$$\mathbf{m} = \mathbf{H}\boldsymbol{\omega} + \boldsymbol{\upsilon} \tag{5.23}$$

where $\mathbf{m}$ is an $n-$dimensional measurement vector, $\boldsymbol{\omega}$ is a measured state vector and its dimension depends on the GNSS and SRIMU systems. For example, $\boldsymbol{\omega}$ is a 3-demensional vector for an SRIMU system and is a 4-dimensional vector for the GNSS, $\mathbf{H}$ is an measurement matrix of proper dimensions, $\boldsymbol{\upsilon}$ is an $n-$dimensional measurement noise with zero mean and covariance $\mathbf{R}_{\boldsymbol{\upsilon}}$. The variances of all sensor measurement noises are hereafter assumed to be identical, that $\mathbf{R}_{\boldsymbol{\upsilon}} = \sigma_{\upsilon}^2 \mathbf{I}_{n\times n}$.

Because the number of measurements in an n-sensor SRIMU configuration is larger than the dimension of the measured state vector, these n measurements are linearly dependent. Without consideration of the measurement errors, there exists a set of scalars, at least one of which is non-zero, such that

$$\sum_{i}^{n} \mathrm{P}_i m_i = 0 \tag{5.24}$$

Eq. (5.24) is generally known as a parity equation. There are $\frac{n!}{3!(n-3)!}$ different parity equations although not all the parity equations are independent. The number of independent parity equations is equal to the number of redundant measurements. The matrix, consisting of the coefficients of n-3 linearly independent parity equations, is known as a parity matrix $\mathbf{P}$. Therefore, Eq. (5.24) can be rewritten in matrix form as follows:

$$\mathbf{Pm} = \mathbf{0} \tag{5.25}$$

The left-hand side of Eq. (5.25) is referred to as a parity vector and can be used to examine the consistency of the SRIMU measurements.

From Eqs. (5.23) and (5.25), a parity matrix $\mathbf{P}$ can be constructed to satisfy the following constraints:

- $\mathbf{PH} = \mathbf{0}$
- $\mathbf{P}$ has $n-3$ linearly independent row vector. Therefore, the parity space is an orthogonal space.
- $\mathbf{PP}^{\mathrm{T}} = \mathbf{I}$ to simplify the detection and isolation functions. This condition normalises each row of the parity matrix.
- $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I} - \mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$. The relationship is proved in Appendix I.

Considering the normal measurement noise, the parity vector is

$$\mathbf{p}_0 = \mathbf{Pm} = \mathbf{PHx} + \mathbf{P}\boldsymbol{\upsilon} = \mathbf{P}\boldsymbol{\upsilon} \tag{5.26}$$

This failure-free parity vector is a Gaussian white noise of zero mean and covariance as follows:

$$\begin{aligned} &E[\mathbf{p}_0] = \boldsymbol{\mu} = 0 \\ &E[\mathbf{p}_0\mathbf{p}_0{}^{\mathrm{T}}] = \mathbf{R}_{\mathrm{p}} = \mathbf{PR}_{\boldsymbol{\upsilon}}\mathbf{P}^{\mathrm{T}} = \sigma_{\upsilon}^2\mathbf{PP}^{\mathrm{T}} \end{aligned} \tag{5.27}$$

5.4.1 Detection Procedure

Assume that sensor failure mode is a jump change with unknown sign and amplitude; the faulty SRIMU measurement equation can be modelled as follows:

$$\mathbf{m} = \mathbf{H}\boldsymbol{\omega} + \mathbf{b} + \boldsymbol{\upsilon} \tag{5.28}$$

where $\mathbf{b}$ is an n-dimensional failure vector and b_i is a nonzero element if the i^{th} sensor has failed, otherwise $\mathrm{b}_i = 0$.

Therefore, the parity vector under failure conditions becomes

$$\mathbf{p}_f = \mathbf{Pb} + \mathbf{P}\boldsymbol{\upsilon} = \mathbf{Pb} + \mathbf{p}_0 \tag{5.29}$$

This failure parity vector is a Gaussian white noise of nonzero mean and the same covariance as the failure-free parity vector. Therefore,

$$\begin{aligned} E[\mathbf{p}_f] &= \boldsymbol{\mu} = \mathbf{Pb} \\ E[\mathbf{p}_f \mathbf{p}_f{}^{\mathrm{T}}] &= \mathbf{R}_{\mathrm{p}} = \mathbf{P}\mathbf{R}_{\upsilon}\mathbf{P}^{\mathrm{T}} = \sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}} \end{aligned} \tag{5.30}$$

From Eqs. (5.27) and (5.30), the statistics of the parity vector **p** is summarised as follows

$$E[\mathbf{p}] = \begin{cases} 0, & H_0 : \text{no failure} \\ \boldsymbol{\mu}, & H_1 : \text{ failure} \end{cases}$$

The probability density functions of the Gaussian distributed parity vector under these two hypotheses are given by

$$f(\mathbf{p}|H_0) = (2\pi)^{-\frac{n-3}{2}} \left|\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}}\right|^{-\frac{1}{2}} \exp[-\frac{1}{2}\mathbf{p}^{\mathrm{T}} (\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}})^{-1} \mathbf{p}]$$

$$f(\mathbf{p}|H_1) = (2\pi)^{-\frac{n-3}{2}} \left|\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}}\right|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\mathbf{p}-\boldsymbol{\mu})^{\mathrm{T}} (\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}})^{-1} (\mathbf{p}-\boldsymbol{\mu})]$$

The log likelihood ratio for the two hypotheses is given by

$$\lambda(\mathbf{p}) = \ln \frac{f(\mathbf{p}|H_1)}{f(\mathbf{p}|H_0)} = \frac{1}{2}[\mathbf{p}^{\mathrm{T}} (\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}})^{-1} \mathbf{p} - (\mathbf{p}-\boldsymbol{\mu})^{\mathrm{T}} (\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}})^{-1} (\mathbf{p}-\boldsymbol{\mu})] \tag{5.31}$$

The maximum likelihood estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ is the value, which maximises $\lambda(\mathbf{p})$. Because two terms on the right-hand side of Eq. (5.31) are positive, the maximum value of $\lambda(\mathbf{p})$ occurs if and only if $\hat{\boldsymbol{\mu}} = \mathbf{p}$.

$$\lambda_{\max}(\mathbf{p}) = \frac{1}{2}\mathbf{p}^{\mathrm{T}} (\sigma_v^2 \mathbf{P}\mathbf{P}^{\mathrm{T}})^{-1} \mathbf{p} = \frac{\mathbf{p}^{\mathrm{T}}\mathbf{p}}{2\sigma_v^2} \tag{5.32}$$

Therefore, the decision function for detection is defined as

$$DFD = \mathbf{p}^{\mathrm{T}}\mathbf{p} \tag{5.33}$$

Given a pre-specified detection threshold η, the detection decision can be stated as follows:

- If $DFD > \eta$, then sensor failures have occurred.
- If $DFD \le \eta$, then no sensor failures have occurred.

However, when two or more sensor failures occur simultaneously, the failure parity vector can be represented as

$$\mathbf{p} = \mathrm{P}_{,i}\mathrm{b}_i + \mathrm{P}_{,j}\mathrm{b}_j$$

where $\mathrm{P}_{,i}$ and $\mathrm{P}_{,j}$ are the i^{th} and j^{th} columns of the parity matrix $\mathbf{P}$, respectively. The detection function takes the following form.

$$\begin{aligned} DFD &= \mathbf{p}^{\mathrm{T}}\mathbf{p} \\ &= (\mathrm{P}_{,i}\mathrm{b}_i + \mathrm{P}_{,\mathrm{j}}\mathrm{b}_j)^{\mathrm{T}}(\mathrm{P}_{,i}\mathrm{b}_i + \mathrm{P}_{,\mathrm{j}}\mathrm{b}_j) \\ &= \mathrm{P}_{,i}{}^{\mathrm{T}}\mathrm{P}_{,i}\mathrm{b}_i{}^2 + \mathrm{P}_{,j}{}^{\mathrm{T}}\mathrm{P}_{,j}\mathrm{b}_j{}^2 + 2\mathrm{P}_{,i}{}^{\mathrm{T}}\mathrm{P}_{,j}\mathrm{b}_i\mathrm{b}_j \end{aligned}$$

The first two items on the right-hand side of the above equation are positive. But the sign of the last item is uncertain because the signs and amplitudes of the failures are arbitrary. If the last item is positive, the test statistic generated through this detection function will increase and the detection decision may give a false alarm because of the accumulation of small biases in individual sensors. On the other hand, if the last item is negative, the inverse situation may lead to a missed detection. Therefore, this detection function cannot guarantee a reliable detection decision on sensor failures when two sensor failures happened simultaneously. In addition, if the noise level is close to the parity residual level, a sensor failure may also become undetectable from Eq. (5.32). These shortcomings have to be overcome in order to improve the performance of the GLRT algorithms.

5.4.2 Isolation Procedure

Failure isolation is to identify those failed sensors after the detection procedure has declared that sensor failures have occurred. As assumed in the above section, the failure parity vector is a nonzero mean Gaussian random variable and a unique nonzero element b_i is contained in the failure vector $\mathbf{b}$ in Eq. (5.29). The associated likelihood function for the failure hypothesis is given by

$$\lambda(\mathrm{b}_i) = \ln f(\mathbf{p}|H_1) = K - \frac{1}{2}(\mathbf{p} - \mathrm{P}_{,i}\mathrm{b}_i)^{\mathrm{T}}(\sigma_v^2\mathbf{PP}^{\mathrm{T}})^{-1}(\mathbf{p} - \mathrm{P}_{,i}\mathrm{b}_i)] \tag{5.34}$$

Because the matrix $\mathbf{PP}^{\mathrm{T}}$ is symmetric, the maximum likelihood estimate $\hat{\mathrm{b}}_i$ of the failure magnitude b_i is

$$\hat{\mathrm{b}}_i = \frac{(\mathrm{P}_{,i}{}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathbf{p})}{\mathrm{P}_{,i}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathrm{P}_{,i}}, \quad i = 1,2,\cdots,n \tag{5.35}$$

Substituting $\hat{\mathrm{b}}_i$ into Eq. (5.34), the maximum likelihood value is given by

$$\lambda_{\max}(b_i) = K - \frac{1}{2\sigma_\upsilon^2}\mathbf{p}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathbf{p} + \frac{1}{2\sigma_\upsilon^2}\frac{(\mathrm{P}_{,i}{}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathbf{p})^2}{\mathrm{P}_{,i}{}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathrm{P}_{,i}} \tag{5.36}$$

The first two terms in the left-hand side of Eq.(5.36) are constant for all sensors but the third term depends on the orientation of sensors. Therefore, the decision function for isolation is defined by

$$DFI_i = \frac{(\mathrm{P}_{,i}{}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathbf{p})^2}{\mathrm{P}_{,i}^{\mathrm{T}}(\mathbf{PP}^{\mathrm{T}})^{-1}\mathrm{P}_{,i}} = \frac{(\mathrm{P}_{,i}{}^{\mathrm{T}}\mathbf{p})^2}{\mathrm{P}_{,i}^{\mathrm{T}}\mathrm{P}_{,i}}, \quad i = 1,2,\cdots,n \tag{5.37}$$

The isolation decision is made as follows:

- If the i^{th} sensor has the maximum value of DFI_i, then it is declared failed.

From Eq. (5.35), this isolation procedure can also be used to estimate the failed sensor signals. The basic GLRT algorithm is illustrated in Figure 5.7.

In summary, the GLRT detection function given by Eq. (5.33) is only used to detect a single sensor failure of a jump change and it cannot simultaneously detect two or more sensor failures. This detection function cannot unambitiously detect sensor failures when the measurement noise level is close to the parity residual level. In addition, the basic GLRT method cannot detect time-varying failures.

From Eq. (5.16), when aircraft is experiencing a high dynamic or manoeuvring motion, the measurement errors caused by scale factor and sensor misalignments will contribute the sensor failures. Consequently, the product of the parity matrix and the measurement matrix is not zero but depends on the measured states. If these measurement errors are not compensated or corrected, false decisions may be made by the decision functions.

Therefore, to improve the sensor FDI performance in terms of the probabilities of false alarm and missed alarm, it is necessary to develop innovative methods to compensate for normal sensor measurement errors and to obtain a sufficiently large failure signal-to-noise ratio before the detection procedure is performed.

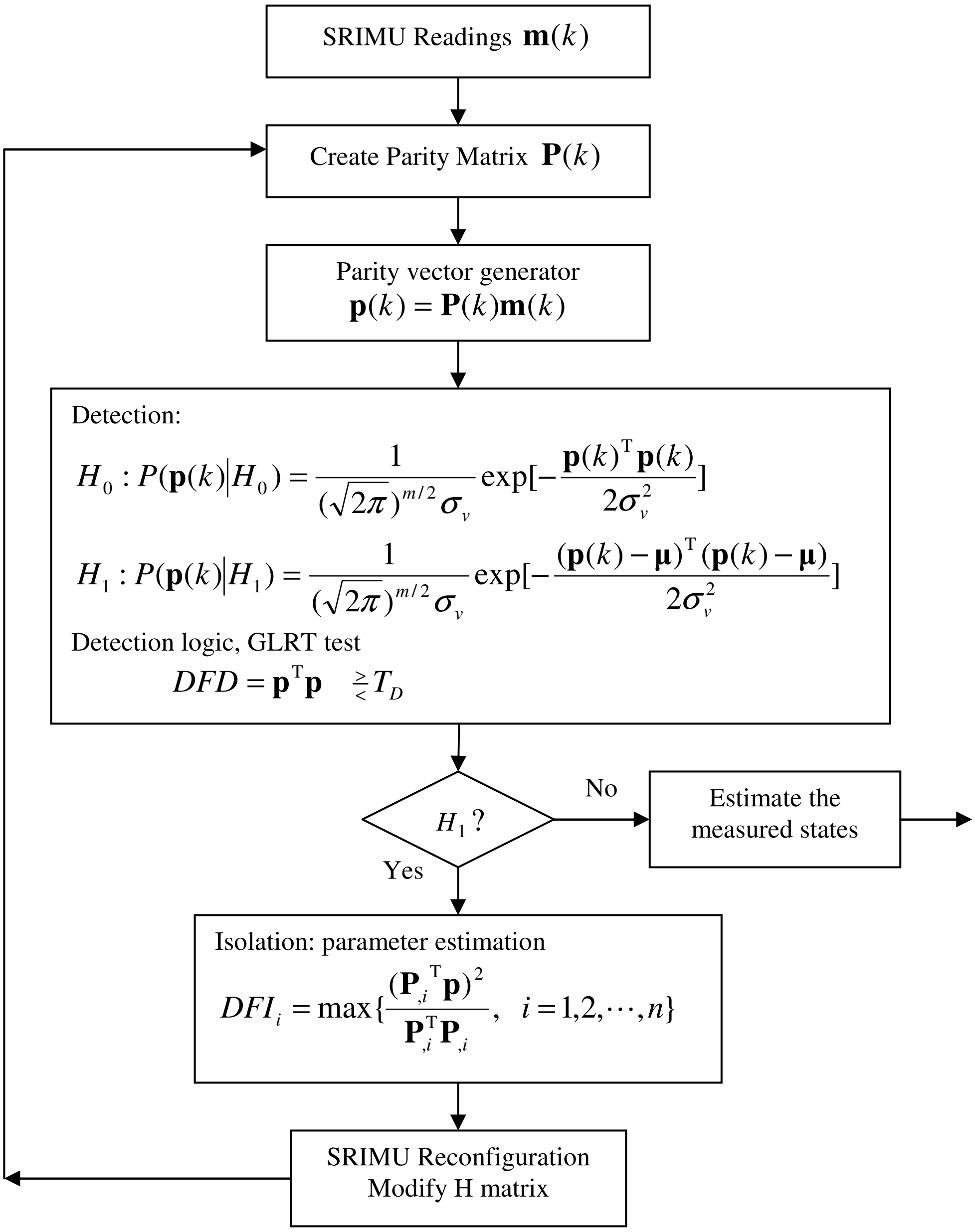

SRIMU Readings $\mathbf{m}(k)$
Create Parity Matrix $\mathbf{P}(k)$
Parity vector generator
$\mathbf{p}(k)=\mathbf{P}(k)\mathbf{m}(k)$
Detection:
$H_0 : P(\mathbf{p}(k)|H_0)=\frac{1}{(\sqrt{2\pi})^{m/2}\sigma_v}\exp[-\frac{\mathbf{p}(k)^{\mathrm{T}}\mathbf{p}(k)}{2\sigma_v^2}]$
$H_1 : P(\mathbf{p}(k)|H_1)=\frac{1}{(\sqrt{2\pi})^{m/2}\sigma_v}\exp[-\frac{(\mathbf{p}(k)-\boldsymbol{\mu})^{\mathrm{T}}(\mathbf{p}(k)-\boldsymbol{\mu})}{2\sigma_v^2}]$
Detection logic, GLRT test
$DFD=\mathbf{p}^{\mathrm{T}}\mathbf{p} \quad \gtreqless T_D$
H_1?
No
Yes
Estimate the measured states
Isolation: parameter estimation
$DFI_i=\max\{\frac{(\mathbf{P}_{,i}{}^{\mathrm{T}}\mathbf{p})^2}{\mathbf{P}_{,i}^{\mathrm{T}}\mathbf{P}_{,i}},\ \ i=1,2,\cdots,n\}$
SRIMU Reconfiguration
Modify H matrix

Figure 5.7 Basic GLRT Algorithm Structure

5.5 SRIMU Error Compensation Filter

This section develops and designs SRIMU error compensation filters, which can be used to compensate for the measurement errors caused by normal sensor error sources. As a result, the performance of the basic GLRT algorithms and the accuracy of the SRIMU measurements can be improved.

5.5.1 SRIMU Error Dynamics

The generalised SRIMU measurement equation is rewritten as follows:

$$\mathbf{m} = \mathbf{H}\boldsymbol{\omega} + \boldsymbol{\delta}_{\mathrm{D}} + \mathbf{S}_{\mathrm{SF}}\mathbf{m} + \mathbf{G}\mathbf{m} + \boldsymbol{\upsilon} \tag{5.38}$$

In theory, the dynamics of these error sources given in the above equation can be modelled by a combination of random constant, random walk and exponentially correlated random processes[124]. The random constant process is used to model the SRIMU measurement errors caused by sensor long-term bias and misalignments. The first-order discrete Gauss-Markov process models the measurement errors caused by combination of the scale factor error and time-dependent sensor drifts. The random walk process models those short-term time-dependent errors.

Therefore, for each sensor, the error terms given in Eq. (5.38) are modelled as follows:

$$\delta_{\mathrm{D}}^{i} = \delta_{\mathrm{B}}^{i} + \delta_{\mathrm{T}}^{i} + \delta_{\mathrm{W}}^{i}$$

$$\dot{\delta}_{\mathrm{B}}^{i} = 0$$

$$\dot{\delta}_{\mathrm{W}}^{i} = \upsilon_{\mathrm{W}}^{i}$$

$$\dot{\delta}_{\mathrm{T}}^{i} = -\beta_{\mathrm{T}}^{i}\delta_{\mathrm{T}}^{i} + \upsilon_{\mathrm{T}}^{i}$$

$$\dot{\delta}_{\mathrm{SF}}^{i} = -\beta_{\mathrm{SF}}^{i}\delta_{\mathrm{SF}}^{i} + \upsilon_{\mathrm{SF}}^{i}$$

$$\dot{\delta}_{\mathrm{Az}^{i}} = 0$$

$$\dot{\delta}_{\mathrm{El}^{i}} = 0$$

where the sensor drift δ_{D} is decomposed into a constant bias error δ_{B}, a time-dependent drift error δ_{T} and a random walk process δ_{W}.

From Eq. (5.22), three additional states are needed to relate the above sensor error states to the resultant navigation state errors. The three additional states are modeled as follows:

$$\dot{\mathbf{x}} = \Delta\boldsymbol{\omega}^{\mathrm{n}} \tag{5.39}$$

An SRIMU consists of two types of sensor set, accelerometer set and gyroscope set. Therefore, two SRIMU compensation filters are needed to separately compensate for the accelerometer and gyro sets. Combining the above sensor error dynamic models and considering (5.22), the dynamic model of each SRIMU compensation filter can be formulated as follows:

$$\mathbf{x}(k+1) = \Phi(k+1)\mathbf{x}(k) + \mathbf{w}(k) \qquad (5.40)$$

where $\mathbf{x}$ is a $6n+3$-dimensional SRIMU error state vector, as explained above, $\Phi(k)$ is a state-transition matrix and it elements are determined by the coefficients of the above sensor error dynamic models, and $\mathbf{w}(k)$ is a white noise sequence of zero mean and variance $\mathbf{Q}(k)$.

Through combination of the SRIMU measurements, three forms of the SRIMU measurement residual equations can be developed as the observables of the SRIMU compensation filters, separately known as the least-squares measurement, the state-free measurement and the parity vector residual equation. In addition, the aircraft velocity and attitude information can also be obtained from navaid systems, for example a multifunctional GNSS receiver or other IMUs located at other nodes in the distributed inertial network system. Therefore, two additional observable equations can be generated, known as the velocity and attitude residual equations, respectively.

5.5.2 Least-Squares Residual Equation

This method is based on the estimate of the measured state vector. Using a least-squares (LS) estimate given by Eq. (5.3), the LS residual vector is given by

$$\Delta\mathbf{m}_{LS} = \mathbf{m} - \hat{\mathbf{m}} = \mathbf{m} - \mathbf{H}\hat{\boldsymbol{\omega}} = \boldsymbol{\delta}_{\mathrm{D}} + \mathbf{S}_{\mathrm{SF}}\mathbf{m} + \mathbf{G}\mathbf{m} + \mathbf{W}\boldsymbol{\upsilon} \qquad (5.41)$$

where $\mathbf{W} = \mathbf{I} - \mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$ is a weighted matrix for the measurement noises and is introduced by the least-squares estimator.

From Eqs. (5.21) and (5.41), the LS-based measurement residual equation is formulated as follows:

$$\Delta\mathbf{m}_{LS} = \begin{bmatrix}\mathrm{I} & \mathbf{Diag(m)} & \mathbf{Diag}(\Pi) & \mathbf{Diag}(\Sigma)\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}_{\mathrm{B}} \\ \boldsymbol{\delta}_{\mathrm{SF}} \\ \boldsymbol{\delta}_{\mathrm{Az}} \\ \boldsymbol{\delta}_{\mathrm{El}}\end{bmatrix} + \mathbf{W}\boldsymbol{\upsilon} \qquad (5.42)$$

From Eq. (5.41), it should be noted that if the state estimates are based on failed sensors, the estimate errors will contribute to the measurement residuals and may decrease the sensitivity of the detection function to actual sensor failures.

5.5.3 State-Free Measurement Equation

This method is based on the linear transformation to the SRIMU measurement equation given by Eq. (5.38).

Defining $\mathbf{U} = \mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I} - \mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$

then $\mathbf{U}$ is an $n \times n$-dimensional symmetric, positive semi-definite matrix because the rank of $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ is the same as the rank of $\mathbf{P}$. Premultiplying the two sides of Eq. (5.38) by $\mathbf{U}$, the state-free measurement equation is

$$\Delta\mathbf{m}_{SF} = \mathbf{Um} = \mathbf{U}\boldsymbol{\delta}_{\mathrm{D}} + \mathbf{US}_{\mathrm{SF}}\mathbf{m} + \mathbf{UGm} + \mathbf{U}\boldsymbol{\upsilon} \tag{5.43}$$

From Eq. (5.21), the above equation can be formulated as follows:

$$\Delta\mathbf{m}_{SF} = \mathbf{U}\left[\mathrm{I} \quad \mathbf{Diag(m)} \quad \mathbf{Diag}(\Pi) \quad \mathbf{Diag}(\Sigma)\right]\begin{bmatrix} \boldsymbol{\delta}_{\mathrm{B}} \\ \boldsymbol{\delta}_{\mathrm{SF}} \\ \boldsymbol{\delta}_{\mathrm{Az}} \\ \boldsymbol{\delta}_{\mathrm{El}} \end{bmatrix} + \mathbf{U}\boldsymbol{\upsilon} \tag{5.44}$$

This formulation of the measurement equation eliminates errors caused by estimating the measured state vector.

5.5.4 Parity Residual Equation

The parity residual equation is derived by directly using the parity vector as the filter measurement. Performing a linear transformation to the SRIMU measurement vector from the measurement space to the parity space, then

$$\mathbf{p} = \mathbf{Pm} = \mathbf{P}\boldsymbol{\delta}_{\mathrm{D}} + \mathbf{PS}_{\mathrm{SF}}\mathbf{m} + \mathbf{PGm} + \mathbf{P}\boldsymbol{\upsilon} \tag{5.45}$$

From Eq. (5.21), the above equation can be normalised as follows:

$$\mathbf{p} = \mathbf{P}\left[\mathrm{I} \quad \mathbf{Diag(m)} \quad \mathbf{Diag}(\Pi) \quad \mathbf{Diag}(\Sigma)\right]\begin{bmatrix} \boldsymbol{\delta}_{\mathrm{B}} \\ \boldsymbol{\delta}_{\mathrm{SF}} \\ \boldsymbol{\delta}_{\mathrm{Az}} \\ \boldsymbol{\delta}_{\mathrm{El}} \end{bmatrix} + \mathbf{P}\boldsymbol{\upsilon} \tag{5.46}$$

5.5.5 Velocity Residual Equation

For the accelerometer error compensation filter, the velocity residual equation is driven on the basis of the difference between the SRIMU-driven velocity $\mathbf{v}_{\text{SRIMU}}$ and the navaid-driven velocity or other SRIMU-driven velocity $\mathbf{v}_{\text{NAVAID}}$.

$$\delta\mathbf{v} \equiv \mathbf{T}_{\text{v}}\mathbf{v}_{\text{NAVAID}} - \mathbf{v}_{\text{SRIMU}} \tag{5.47}$$

where $\mathbf{T}_{\text{v}}$ is a velocity transformation matrix from other SRIMU node frame to the detected SRIMU frame.

From Eqs. (5.22) and (5.39), the velocity residual equation is given by

$$\delta\mathbf{v} = \mathbf{x}_{\text{v}} + \boldsymbol{\upsilon}_{\text{NAVAID-v-noise}} \tag{5.48}$$

where $\boldsymbol{\upsilon}_{\text{NAVAID-v-noise}}$ is the navaid-driven velocity solution noise.

5.5.6 Attitude Residual Equation

For the gyroscope error compensation filter, the attitude residual equation is generated by differencing the SRIMU-based attitude solution $\boldsymbol{\theta}_{\text{SRIMU}}$ and the navaid-based attitude solution or other IMU-driven attitude solution $\boldsymbol{\theta}_{\text{NAVAID}}$. For example, a multifunctional GNSS receiver can output all of the navigation states.

$$\delta\boldsymbol{\theta} \equiv \mathbf{T}_{\boldsymbol{\theta}}\boldsymbol{\theta}_{\text{NAVAID}} - \boldsymbol{\theta}_{\text{SRIMU}} \tag{5.49}$$

where $\mathbf{T}_{\boldsymbol{\theta}}$ is an attitude transformation matrix from other SRIMU node frame to the detected SRIMU frame.

From Eqs. (5.22) and (5.39), the attitude residual equation is given by

$$\delta\boldsymbol{\theta} = \mathbf{x}_{\theta} + \boldsymbol{\upsilon}_{\text{NAVAID-}\theta\text{-noise}} \tag{5.50}$$

where $\boldsymbol{\upsilon}_{\text{GNSS-}\theta\text{-noise}}$ is the navaid-driven attitude solution noise.

It should be noted that relationships created in Eqs. (5.47) and (5.49) enable the inertial and navigation state information in a distributed sensor network system be shared at all network nodes. This information sharing technique enhances both the performance of local sensor FDI functions and the fault tolerance of the distributed multisensor navigation system. The transformations in Eq (5.47) and (5.49) can be a unit matrix when the SRIMU error compensation filters are located at the cg node and the velocity and attitude residual equations are based on the GNSS navigation solutions.

5.5.7 SRIMU Error Compensated FDI Algorithm Structure

In the case of normal SRIMU operation, the compensation filters estimate the SRIMU errors, including gyro and accelerometer errors, and these error estimates are used to correct raw SRIMU measurements. The corrected SRIMU measurements are then fed into a sequential moving-window GLRT to detect sensor failures. A modular architecture for the improved FDI algorithms is shown in Figure 5.8.

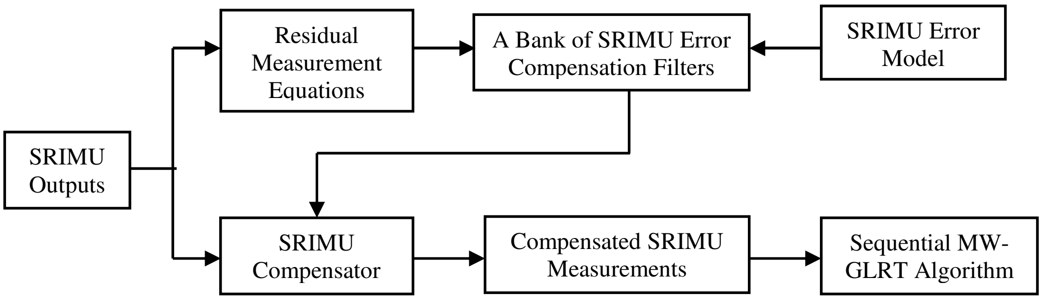

Residual Measurement Equations
A Bank of SRIMU Error Compensation Filters
SRIMU Error Model
SRIMU Outputs
SRIMU Compensator
Compensated SRIMU Measurements
Sequential MW-GLRT Algorithm

Figure 5.8 Modular Architecture of Improved FDI Algorithm

5.6 Moving-Window GLRT Methods

Traditionally, sequential FDI methods are based on the sequential probability ratio test (SPRT) developed by Wald, which use all the residual samples from the initial time to the current time. This detection method decreases the sensitivity to the detection of actual sensor failures as time progresses. A sequential moving-window GLRT (MW-GLRT) method is presented for detecting both jump and drift failures, which may degrade the performance of an SRIMU system. The structure of this moving-window GLRT is depicted in Figure 5.9. The parity residual vector or the measurement residual vector sequentially passes a first-in-first-out buffer of a length L, which generates a sequential test statistic. Failure detection is then performed by comparing this test statistic with a pre-specified threshold.

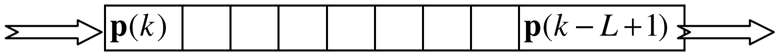


Figure 5.9 MW-GLRT Structure

5.6.1 Sequential MW-GLRT Detection Procedure

Consider that the normal SRIMU sensor errors have been largely corrected by means of the error compensation filters, the failed SRIMU measurement is modelled as follows:

$$\mathbf{m}=\mathbf{H}\boldsymbol{\omega}+\mathbf{b}+\mathbf{d}(t_k)+\boldsymbol{\upsilon} \tag{5.51}$$

where $\mathbf{b}$ is a jump bias failure and $\mathbf{d}(t_k)$ is a random drift failure with unknown statistical characteristics.

In this case, the failure detection is to check the maximum allowable jump bias failure and the maximum allowable drift rate failure. Considering the constraints on the parity matrix, and the statistical character of the parity vector, the parity vectors are a sequence of independent Gaussian random variables. A moving-window joint likelihood ratio function is given as follows:

$$\begin{aligned}\lambda_L(k)&=\ln\frac{f[\mathbf{p}(k),\mathbf{p}(k-1),\cdots\mathbf{p}(k-L+1)|H_1]}{f[\mathbf{p}(k),\mathbf{p}(k-1),\cdots\mathbf{p}(k-L+1)|H_0]}\\&=\frac{1}{2\sigma_\upsilon^2}\sum_{i=k-L+1}^{k}\{\mathbf{p}^{\mathrm{T}}(i)\mathbf{p}(i)-[\mathbf{p}(i)-\boldsymbol{\mu}(i)]^{\mathrm{T}}[\mathbf{p}(i)-\boldsymbol{\mu}(i))]\}\\&=\frac{1}{\sigma_\upsilon^2}\sum_{i=k-L+1}^{k}\{[\mathbf{p}(i)-\frac{1}{2}\boldsymbol{\mu}(i)]^{\mathrm{T}}\boldsymbol{\mu}(i)\}\end{aligned} \tag{5.52}$$

Furthermore,

$$\begin{aligned}\lambda_L(k)=\lambda_L(k-1)+\frac{1}{\sigma_\upsilon^2}\{[\mathbf{p}(k)-\frac{1}{2}\boldsymbol{\mu}(k)]^{\mathrm{T}}\boldsymbol{\mu}(k)\\-[\mathbf{p}(k-L)-\frac{1}{2}\boldsymbol{\mu}(k-L)]^{\mathrm{T}}\boldsymbol{\mu}(k-L)\}\end{aligned} \tag{5.53}$$

Two methods are used to define the sequential decision function for detection. One assumes a constant design value $\hat{\boldsymbol{\mu}}_{\mathrm{b}}$ of the parity vector average $\boldsymbol{\mu}_i$ within the L-length window, which depends on the accuracy requirement to an SRIMU navigation system. The sequential detection function is then defined by

$$\begin{aligned} DFD_L(k) &= \lambda_L(k) \\ &= \lambda_L(k-1) + \frac{1}{\sigma_v^2}[(\mathbf{p}(k) - \mathbf{p}(k-L)]^{\mathrm{T}} \hat{\boldsymbol{\mu}}_{\mathrm{b}} \end{aligned} \tag{5.54}$$

For each type of inertial sensors, the constant design value $\hat{\boldsymbol{\mu}}_{\mathrm{b}}$ is determined on the basis of the sensor statistics and allowable error, and has the sign of the moving window average value and an amplitude value as follows:

$$sign(\hat{\boldsymbol{\mu}}_{\mathrm{b}}) = sign[\sum_{i=k-L+1}^{k} \mathbf{p}(i)]$$

$$|\hat{\boldsymbol{\mu}}_{\mathrm{b}}| = \frac{\text{maximum allowable angular error}}{\text{Time Interval} = L \times \Delta t}$$

where Δt is the sample interval of SRIMU outputs.

Therefore, the change trend of $\lambda_k(L)$ is illustrated in Figure 5.10.

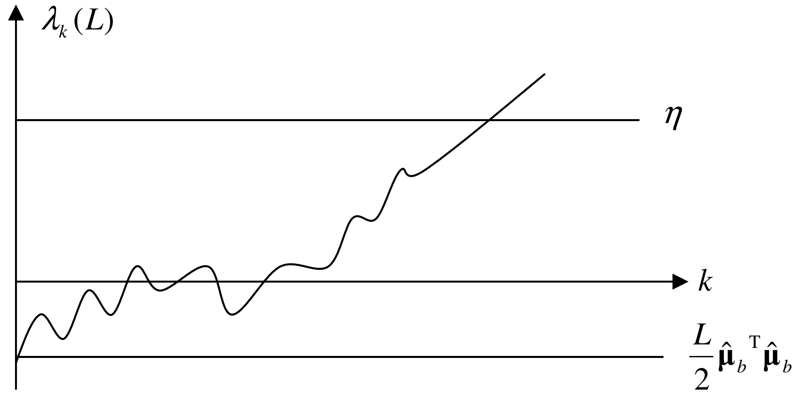

$\lambda_k(L)$
η
k
$\frac{L}{2}\hat{\boldsymbol{\mu}}_b{}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_b$

Figure 5.10 Detection Function for A Constant Design Value of $\boldsymbol{\mu}_i$

The detection rules are stated as follows:

- If $DFD_L(k) \geq \eta$, then sensor failures have occurred.
- If $DFD_L(k) < \eta$, then no sensor failure has occurred.

The other method defines the detection function on the basis of the maximum likelihood estimates of average of the sequential parity vectors. Assume the average value is an unknown constant $\boldsymbol{\mu}$ within the L-length window; from Eq. (5.52), the estimate of $\boldsymbol{\mu}$, $\hat{\boldsymbol{\mu}}$, is given by

$$\hat{\boldsymbol{\mu}} = \frac{1}{L}\sum_{i=k-L+1}^{k}\mathbf{p}(i) \tag{5.55}$$

The maximum likelihood ratio is

$$\lambda_{\max,L}(k) = \frac{1}{2\sigma_v^2 L}[\sum_{i=k-L+1}^{k}\mathbf{p}(i)]^{\mathrm{T}}[\sum_{i=k-L+1}^{k}\mathbf{p}(i)] \tag{5.56}$$

and the detection function is defined as

$$DFD_L(k) = [\sum_{i=k-L+1}^{k}\mathbf{p}(i)]^{\mathrm{T}}[\sum_{i=k-L+1}^{k}\mathbf{p}(i)] \tag{5.57}$$

Therefore, the detection rules are stated as follows:

- If $DFD_L(k) \geq \eta$, then failures occurred;
- If $DFD_L(k) < \eta$, then no failures occurred.

The change trend of this detection function is illustrated in Figure 5.11.

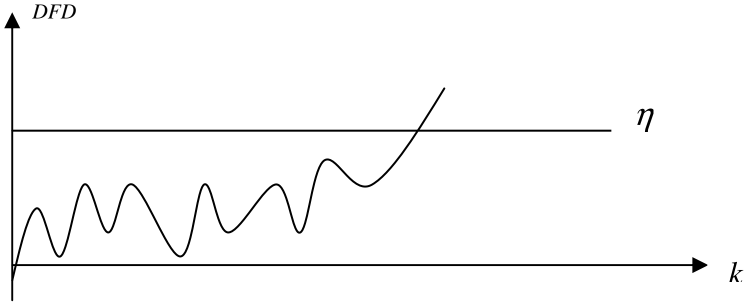


Figure 5.11 Detection Function based on Estimate of Window Average

From Eqs. (5.54) and (5.56), this sequential detection is easily affected by the measurement noise and the detection performance may be degraded with the increase of the noise variance, especially in less accurate, low-cost SRIMU system. Improved sequential MW GLRT methods will be introduced to overcome the above problem in the following sections.

5.6.2 Sequential-Averaged Method

In order to reduce the effects of measurement noise on the sequential residual signals, it is necessary to pre-process the original parity vector sequences to generate a new parity vector. This new parity vector or residual signal is then used to detect failures. A sequential averaged detection method is introduced in this section. This method sequentially averages the parity vector or residual signal sequences within a moving window. The sequential average value can be computed as follows:

$$\begin{aligned}\overline{\mathbf{p}}(k) &= \frac{\mathbf{p}(k)+\mathbf{p}(k-1)+\cdots+\mathbf{p}(k-L+1)}{L} \\ &= \overline{\mathbf{p}}(k-1)+\frac{1}{L}[\mathbf{p}(k)-\mathbf{p}(k-L)]\end{aligned} \tag{5.58}$$

$$\overline{\mathbf{p}}(1)=\mathbf{p}(1)$$

This sequential average is also a Gaussian distributed random variable. Its mean is the same as the original parity vector, that is,

$$E[\overline{\mathbf{p}}(k)]=\frac{1}{L}\sum_{i=(k-1)L+1}^{kL}E[\mathbf{p}(i)]=\frac{1}{L}\sum_{i=(k-1)L+1}^{kL}\boldsymbol{\mu}_{\mathrm{p}}=\boldsymbol{\mu}_{\mathrm{p}}$$

But, the covariance $\mathbf{R}_{\bar{\mathrm{p}}}$ is

$$\begin{aligned}\mathbf{R}_{\bar{\mathrm{p}}} &= E[\overline{\mathbf{p}}\overline{\mathbf{p}}^{\mathrm{T}}]=E[\frac{1}{L^2}\sum_{i=(k-1)L+1}^{kL}\mathbf{p}(i)\sum_{i=(k-1)L+1}^{kL}\mathbf{p}(i)] \\ &= \frac{1}{L^2}E\{[\mathbf{p}(kL)+\cdots+\mathbf{p}((k-1)L+1)][\mathbf{p}(kL)+\cdots+\mathbf{p}((k-1)L+1)]^{\mathrm{T}}\}\end{aligned}$$

Assuming the parity vector sequences are independent, $E[\mathbf{p}(i)\mathbf{p}^{\mathrm{T}}(j)]=0\ \ (i\neq j)$, so that the above equation can be simplified as follows:

$$\begin{aligned}\mathbf{R}_{\bar{\mathrm{p}}} &= \frac{1}{L^2}\{E[\mathbf{p}(kL)\mathbf{p}^{\mathrm{T}}(kL)]+\cdots+E[\mathbf{p}((k-1)L+1)\mathbf{p}^{\mathrm{T}}((k-1)L+1)]\} \\ &= \frac{1}{L}\mathbf{R}_{\mathrm{p}}\end{aligned} \tag{5.59}$$

Therefore, this sequential-averaged method reduces the variance of the measurement noise and can remove high-frequency noise and wild-values. It can also identify the feature of the drift failures. The size of window should be chosen so that the influence of the measurement noise on the failure detection procedure can be largely decreased. As a result, it enhances the sensitivity of the detection algorithm to true drift failures.

The detection of allowable maximum rate of the drift failure is usually needed in order to afford highly reliable angular rate data for flight control systems and other avionics systems. To detect the drift rate of the drift failure $\mathbf{d}(t_k)$ in Eq. (5.51), the averaged parity vector is sequentially differenced and the results are averaged within the moving-window of a length L as a decision function for the detection of drift rate failures as follows:

$$\mathbf{s}_L(k) = \frac{2}{L * \Delta t} \sum_{i=(k-1)L/2+1}^{kL/2} \Delta \overline{\mathbf{p}}(i) = \frac{2}{L * \Delta t} \sum_{i=(k-1)L/2+1}^{kL/2} [\overline{\mathbf{p}}(2i) - \overline{\mathbf{p}}(2i-1)] \tag{5.60}$$

where $\mathbf{s}_L(k)$ represents an average rate of the drift failure within the window.

The norm of $\mathbf{s}_L(k)$ is defined as the decision function for detection of drift rate failure $DFD_L(k) = \|\mathbf{s}_L(k)\|$. Given a drift rate threshold η_D, then the detection rules are described as follows:

1. If $DFD_L(k) \geq \eta_D$, then drift rate failures have occurred,
2. Otherwise, no drift rate failure happened.

5.6.3 Sequential-Averaged MW-GLRT Methods

This improved method is a combination of the sequential MW-GLRT method and the sequential-averaged detection. To simplify the mathematical equations, the sequential average value is computed by the following equation,

$$\overline{\mathbf{p}}(k) = \frac{1}{L} \sum_{i=(k-1)L+1}^{kL} \mathbf{p}(i) \tag{5.61}$$

Combining Eqs.(5.52) and (5.61), the normalised sequential likelihood ratio of the sequential moving-window average can be formulated by

$$\begin{aligned} \lambda(k) &= \sum_{i=1}^{L} \sum_{j=1}^{N} [\overline{\mathbf{p}}^{\mathrm{T}}((i-1)N+j)\boldsymbol{\mu}_{\mathrm{b}} - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathrm{b}}] \\ &= \sum_{i=1}^{L} \sum_{j=1}^{N} [\overline{\mathbf{p}}^{\mathrm{T}}((i-1)N+j) - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{b}}^{\mathrm{T}}]\boldsymbol{\mu}_{\mathrm{b}} \end{aligned} \tag{5.62}$$

where $\boldsymbol{\mu}_{\mathrm{b}}$ is a constant design value dependent of the accuracy requirement for the SRIMU system.

$$\boldsymbol{\mu}_{\mathrm{b}} = \mathbf{P}\mathbf{b}_{\mathrm{d}}$$

$$sign(\boldsymbol{\mu}_b(k)) = sign(\overline{\mathbf{p}}(k))$$

where $\mathbf{b}_{\mathrm{d}}$ is a minimum SRIMU drift error, determined according to the accuracy requirement of an SRIMU system, and the sign of the elements of $\mathbf{b}_{\mathrm{d}}$ depends on the sign of the estimated measurement residuals.

$$sign[\mathbf{b}_{\mathrm{d}}(k)] = sign[\Delta\mathbf{m}(k)]$$

Eq.(5.62) can be rewritten as follows

$$\lambda(k) = \lambda(k-1) + \pi(k) \tag{5.63}$$

where

$$\pi(k) = \sum_{j=1}^{N} [\bar{\mathbf{p}}^{\mathrm{T}}((k-1)N + j)\boldsymbol{\mu}_{\mathrm{b}} - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathrm{b}}]$$

Because $\boldsymbol{\mu}_{\mathrm{b}}$ has the sign of $\bar{\mathbf{p}}(k)$, the above equation can also be expressed as

$$\pi(k) = \sum_{j=1}^{N} [\left|\bar{\mathbf{p}}^{\mathrm{T}}((k-1)N + j)\boldsymbol{\mu}_{\mathrm{b}}\right| - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathrm{b}}] \tag{5.64}$$

When $\sum_{j=1}^{N} [\left|\bar{\mathbf{p}}^{\mathrm{T}}((k-1)N + j)\boldsymbol{\mu}_{\mathrm{b}}\right| - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathrm{b}}] < 0$, specifying $\pi(k) \equiv 0$. Therefore, $\lambda(k)$ is monotonously incremental function. This makes the sensor drift be rapidly detected.

Accordingly, the sequential decision rule is

- If $\lambda(n) \geq \eta_{\bar{\mathrm{p}}}$, then sensor failures have occurred at the time $t = L * N * \Delta t$.
- Otherwise, no sensor failure happened.

where $\eta_{\bar{\mathrm{p}}}$ is a sequential detection threshold which is based on the probabilities of missed detection and false alarm and $\eta_{\bar{\mathrm{p}}} = \ln\frac{1-P_{MD}}{P_{FA}}\sigma_{\bar{\mathrm{p}}}^2$.

5.7 Summary

This chapter introduced the topology architectures of sensor network systems, and developed methodologies for evaluation of various configurations of the skewed redundant inertial measurement units (SRIMUs) and for detection and isolation of sensor failures appearing in the SRIMUs. The main deliveries cover:

1. Description of two forms of sensor system architectures: the distributed sensor system architecture and the clustered sensor system architecture.

2. Development of three criteria to evaluate optimal configurations of skewed redundant inertial systems in the clustered sensor system architecture, including minimum GDOP factor, identical variance errors along the three orthogonal body axes and optimal SRIMU FDI performance. Based on these criteria, coplanar sensor installations in SRIMUs should be avoided in order to obtain the maximum SRIMU FDI capability.
3. Comparison of the performance of several SRIMU configurations, including their measurement accuracy and reliability.
4. Development of the SRIMU error calibration algorithms for design of local Kalman filter and dynamic SRIMU error controls.
5. Design of the SRIMU error compensation Kalman filters to improve the performance of the FDI algorithms and the accuracy of SRIMU systems.
6. Development and improvement of the moving-window GLRT methods to detect three kinds of inertial sensor failure modes, including jump, time-drift and drift rate failures.

Chapter 6

DISTRIBUTED DATA FUSION ALGORITHMS

6.1 Introduction

This chapter develops data fusion methodologies for distributed sensor network systems, including data fusion filter algorithms and integrity monitoring algorithms. Section 6.2 introduces general distributed fusion algorithms for several distributed sensor systems. Section 6.3 develops inertial network measurement models and also establishes dynamic relationships among inertial network nodes. Inertial network data fusion algorithms are developed in Section 6.4. Inertial network integrity monitoring algorithms are presented in Section 6.5. Finally, a summary is given in Section 6.6.

6.2 Distributed Sensor Systems and Fusion Algorithms

This section develops several forms of distributed fusion filter algorithms for differently distributed sensor systems. The dynamics of a distributed sensor network system can be described by one global dynamic model and N local dynamic models where N is the number of the nodes or the local sensor systems in a distributed sensor network system. Let the global system model be formulated as follows:

$$\mathbf{x}(t_k) = \mathbf{\Phi}(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{w}(t_{k-1}) \tag{6.1}$$

$$\mathbf{z}(t_k) = \mathbf{H}(t_k)\mathbf{x}(t_k) + \boldsymbol{\upsilon}(t_k) \tag{6.2}$$

and the local system models be represented by

$$\mathbf{x}_i(t_k)=\boldsymbol{\Phi}_i(t_k,t_{k-1})\mathbf{x}_i(t_{k-1})+\mathbf{B}_i(t_{k-1})\mathbf{w}_i(t_{k-1}) \qquad i=1,\cdots,N \tag{6.3}$$

$$\mathbf{z}_i(t_k)=\mathbf{C}_i(t_k)\mathbf{x}_i(t_k)+\boldsymbol{\upsilon}_i(t_k) \qquad i=1,\cdots,N \tag{6.4}$$

Because all the sensor systems are independent and their measurement noise is also independent of the measured quantities, the global measurement model given in Eq. (6.2) can be partitioned into the following block matrices or vectors.

$$\mathbf{z}(t_k)=\left[\mathbf{z}_1^{\mathrm{T}}(t_k) \quad \mathbf{z}_2^{\mathrm{T}}(t_k) \quad \cdots \quad \mathbf{z}_N^{\mathrm{T}}(t_k)\right]^{\mathrm{T}} \tag{6.5}$$

$$\boldsymbol{\upsilon}(t_k)=\left[\boldsymbol{\upsilon}_1^{\mathrm{T}}(t_k) \quad \boldsymbol{\upsilon}_2^{\mathrm{T}}(t_k) \quad \cdots \quad \boldsymbol{\upsilon}_N^{\mathrm{T}}(t_k)\right]^{\mathrm{T}} \tag{6.6}$$

$$\mathbf{H}(t_k)=\left[\mathbf{H}_1^{\mathrm{T}}(t_k) \quad \mathbf{H}_2^{\mathrm{T}}(t_k) \quad \cdots \quad \mathbf{H}_N^{\mathrm{T}}(t_k)\right]^{\mathrm{T}} \tag{6.7}$$

$$\mathbf{R}(t_k)=blockdiag\left[\mathbf{R}_1(t_k) \quad \mathbf{R}_2(t_k) \quad \cdots \quad \mathbf{R}_N(t_k)\right] \tag{6.8}$$

where $\mathbf{R}_i$ is the covariance matrix of the measurement noise $\boldsymbol{\upsilon}_i$.

The local system model is a practical dynamic model of a node or a local system in a distributed network system. The global system model can be either a true model of a practical distributed system or a virtual model, which is established in order to develop distributed data fusion algorithms. For example, in many target-tracking systems, the global system models normally describe the dynamic motion of the tracked targets. In distributed control systems, a global system model may not exist although global optimisation is usually required. In an integrated aircraft navigation system with distributed sensor systems, the aircraft centre of gravity (*cg*) is a special location with respect to which many parameters or states used in aircraft navigation and flight control systems are defined. Therefore, the global system model of aircraft navigation system usually describes the dynamic motion of the aircraft centre of gravity and is approximated by the error dynamic model of an inertial system located at the *cg*. However, this is not necessarily true in a distributed inertial network system where each node has its own local dynamic model and a global model is not needed.

In this thesis, the development of distributed data fusion filter algorithms is based on two principles. One is known as the global-to-local optimisation method and the other is referred to as the local-to-global optimisation method. In the global-to-local optimisation method, the distributed local filters are designed on the basis of optimisation of the global system model whereas in the local-to-global optimisation method, global optimisation is obtained by the optimisation of the local systems.

Depending on the network communication modes and the characteristics of the node dynamic models, two classes of distributed fusion algorithms, referred to as state-identical distributed fusion algorithms and state-associated distributed fusion algorithms, are discussed in the following subsections. For each class of distributed fusion algorithm, several distributed fusion filters are developed for different sensor network systems.

6.2.1 State-Identical Distributed Fusion Algorithms

State-identical distributed fusion algorithms are developed for distributed sensor network systems where all the sensor systems are distributed but the observed object is identical. Depending on data communication modes among the network nodes, two types of distributed filter algorithms are analysed. The Type I algorithm is used for distributed systems using one-way communication, as shown in Figure 6.1 where the arrows indicate the directions of data flow. The Type II algorithm applies to distributed systems using two-way communication, as shown in Figure 6.2. In both these distributed systems, all the sensor systems observe the same dynamic system. Therefore, all the local system models and the global system model are identical.

The distributed network system model given by from Eqs. (6.1) to (6.4) can be simplified where the following assumptions apply:

$$\mathbf{x}_i(t_k) = \mathbf{x}_j(t_k) \equiv \mathbf{x}(t_k)$$

$$\mathbf{\Phi}_i = \mathbf{\Phi}_j \equiv \mathbf{\Phi}$$

$$\mathbf{B}_i(t_k) = \mathbf{B}_j(t_k) \equiv \mathbf{G}(t_k)$$

$$\mathbf{w}_i(t_k) = \mathbf{w}_j(t_k) \equiv \mathbf{w}(t_k)$$

$$\mathbf{C}_i(t_k) = \mathbf{H}_i(t_k)$$

By following Hashemipour’s work[62], the distributed filter algorithms are derived as follows.

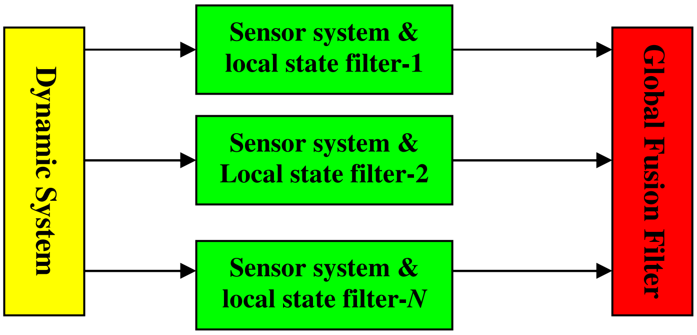

Dynamic System
Sensor system & local state filter-1
Sensor system & Local state filter-2
Sensor system & local state filter-N
Global Fusion Filter

Figure 6.1 State-Identical One-Way Model

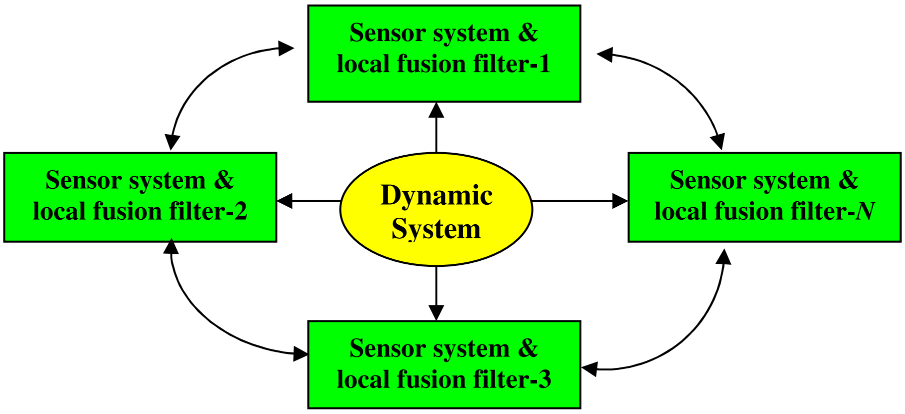

Sensor system & local fusion filter-1
Sensor system & local fusion filter-2
Dynamic System
Sensor system & local fusion filter-N
Sensor system & local fusion filter-3

Figure 6.2 State-Identical Two-Way Model

From the information form of the Kalman filter algorithm given in Chapter 3, the local estimates can be obtained and represented by the following forms:

$$\hat{\mathbf{x}}_i(t_k^-) = \mathbf{\Phi}_i(t_k, t_{k-1})\hat{\mathbf{x}}_i(t_{k-1}^+) \tag{6.9}$$

$$\mathbf{P}_i(t_k^-) = \mathbf{\Phi}_i(t_k, t_{k-1})\mathbf{P}_i(t_{k-1}^+)\mathbf{\Phi}_i^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{B}_i(t_{k-1})\mathbf{Q}_i(t_{k-1})\mathbf{B}_i^{\mathrm{T}}(t_{k-1}) \tag{6.10}$$

$$\mathbf{H}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{H}_i(t_k) = \mathbf{P}_i^{-1}(t_k^+) - \mathbf{P}_i^{-1}(t_k^-) \tag{6.11}$$

$$\mathbf{H}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{z}_i(t_k) = \mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-) \tag{6.12}$$

and the global time-update and measurement-update equations are as follows:

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{\Phi}(t_k, t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+) \tag{6.13}$$

$$\mathbf{P}(t_k^-) = \mathbf{\Phi}(t_k, t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{G}(t_{k-1})\mathbf{Q}(t_{k-1})\mathbf{G}^{\mathrm{T}}(t_{k-1}) \tag{6.14}$$

$$\begin{aligned}\mathbf{P}^{-1}(t_k^+) &= \mathbf{P}^{-1}(t_k^-) + \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{H}(t_k) \\ &= \mathbf{P}^{-1}(t_k^-) + \sum_{i=1}^{N}\mathbf{H}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{H}_i(t_k)\end{aligned} \tag{6.15}$$

$$\begin{aligned}\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) &= \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \mathbf{H}^{\mathrm{T}}(t_k)\mathbf{R}^{-1}(t_k)\mathbf{z}(t_k) \\ &= \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \sum_{i=1}^{N}\mathbf{H}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{z}_i(t_k)\end{aligned} \tag{6.16}$$

Substituting Eq. (6.11) into Eq. (6.15) and Eq. (6.12) into Eq. (6.16) generates the following global update equations

$$\mathbf{P}^{-1}(t_k^+) = \mathbf{P}^{-1}(t_k^-) + \sum_{i=1}^{N}[\mathbf{P}_i^{-1}(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)] \tag{6.17}$$

$$\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \sum_{i=1}^{N}[\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-)] \tag{6.18}$$

Eqs. (6.9) to (6.14), and (6.17) and (6.18) describe the Type I algorithm. In this algorithm, each parallel local filter only processes its own measurements in order to generate its local estimates. The global fusion filter assimilates all the local estimates to update the global estimate. Obviously, this algorithm has a simple structure. The main disadvantage of this algorithm is that the degradation of the local filter performance may critically affect the performance of the global fusion filter because the global estimate cannot be used to refresh the local estimates.

For the two-way model shown in Figure 6.2, the time-update equations are given by Eqs. (6.13) and (6.14). Placing the global update equations (6.17) and (6.18) at each node yields

$$\mathbf{P}_j^{-1}(t_k^+) = \mathbf{P}_j^{-1}(t_k^-) + \sum_{i=1}^{N}[\mathbf{P}_i^{-1}(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)] \tag{6.19}$$

$$\mathbf{P}_j^{-1}(t_k^+)\hat{\mathbf{x}}_j(t_k^+) = \mathbf{P}_j^{-1}(t_k^-)\hat{\mathbf{x}}_j(t_k^-) + \sum_{i=1}^{N}[\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-)] \tag{6.20}$$

Eqs. (6.9) and (6.12), and (6.19) to (6.20) constitute the Type II algorithm. In this algorithm, each local fusion filter updates its global estimate by assimilating the local estimates from the other local fusion filters. Accordingly, all the local estimates can be dynamically corrected by their global estimate updates. The Type II algorithm overcomes the disadvantage of the Type I algorithm and provides the redundant global state estimates. Therefore, this algorithm is a fault-tolerant fusion algorithm.

In many practical examples of state-identical distributed sensor network systems, not all the sensor systems can observe the complete states of the same dynamic system. Therefore, the local state is a subset of the global system state and the local filters are generally designed to be reduced-order. In such cases, the Type I and Type II algorithms given above must be modified.

Let the local system states be abstracted from the global state as follows:

$$\mathbf{x}_i(t_k) = \mathbf{D}_i\mathbf{x}(t_k) \tag{6.21}$$

where $\mathbf{D}_i$ is a one-way state abstraction matrix consisting of ones or zeros and each row of $\mathbf{D}_i$ has at most one non-zero element.

From Eq. (6.4), a sub-matrix of the global measurement matrix is associated with a local measurement matrix by

$$\mathbf{H}_i(t_k) = \mathbf{C}_i(t_k)\mathbf{D}_i \tag{6.22}$$

From Eqs. (6.15) and (6.16), the global update equations are given by

$$\mathbf{P}^{-1}(t_k^+) = \mathbf{P}^{-1}(t_k^-) + \sum_{i=1}^{N}\mathbf{D}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)]\mathbf{D}_i \tag{6.23}$$

$$\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \sum_{i=1}^{N}\mathbf{D}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-)] \tag{6.24}$$

and the local estimates can be obtained as follows:

$$\mathbf{P}_i(t_k^-) = \mathbf{\Phi}_i(t_k, t_{k-1})\mathbf{P}_i(t_k^+)\mathbf{\Phi}_i^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{B}_i(t_k)\mathbf{Q}_i(t_k)\mathbf{B}_i^{\mathrm{T}}(t_k) \tag{6.25}$$

$$\mathbf{P}_i^{-1}(t_k^+) = \mathbf{P}_i^{-1}(t_k^-) - \mathbf{C}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{C}_i(t_k) \tag{6.26}$$

$$\hat{\mathbf{x}}_i(t_k^-) = \mathbf{\Phi}_i(t_k, t_{k-1})\hat{\mathbf{x}}_i(t_{k-1}^+) \tag{6.27}$$

$$\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) = \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-) - \mathbf{C}_i^{\mathrm{T}}(t_k)\mathbf{R}_i^{-1}(t_k)\mathbf{z}_i(t_k) \tag{6.28}$$

Eqs. (6.23) to (6.28) constitute the modified Type I algorithm.

To deduce the modified Type II algorithm, from Eq. (6.21), the relationship between the covariances of the local states and the covariance of the global state can be obtained as follows:

$$\mathbf{P}_i(t_k) = \mathbf{D}_i\mathbf{P}(t_k)\mathbf{D}_i^{\mathrm{T}} \tag{6.29}$$

$$\mathbf{P}_i^{-1}(t_k) = [\mathbf{D}_i^{\mathrm{T}}]^*\mathbf{P}^{-1}(t_k)[\mathbf{D}_i]^* \tag{6.30}$$

$$\mathbf{P}^{-1}(t_k) = \mathbf{D}_i^{\mathrm{T}}\mathbf{P}_i^{-1}(t_k)\mathbf{D}_i \tag{6.31}$$

where $\mathbf{D}_i^*$ is a Moore-Penrose inverse of $\mathbf{D}_i$ and $[\mathbf{D}_i^*]^* = \mathbf{D}_i$. As the row dimension of $\mathbf{D}_i$ is less than its column, $\mathbf{D}_i^* = \mathbf{D}_i^{\mathrm{T}}(\mathbf{D}_i\mathbf{D}_i^{\mathrm{T}})^{-1}$.

It should be noted that $\mathbf{D}_i$ is a one-way abstraction matrix and Eq. (6.31) cannot be used to restore the global covariance from a local covariance. However, if the purpose of the local fusion filter at each node is to update the local estimates, Eq. (6.31) will be valid.

From Eqs. (6.21) and (6.31), the global system state at each network node can be represented by the following equation

$$\mathbf{P}^{-1}(t_k)\hat{\mathbf{x}}(t_k) = \mathbf{D}_i^{\mathrm{T}}\mathbf{P}_i^{-1}(t_k)\mathbf{D}_i\hat{\mathbf{x}}(t_k) = \mathbf{D}_i^{\mathrm{T}}\mathbf{P}_i^{-1}(t_k)\hat{\mathbf{x}}_i(t_k) \tag{6.32}$$

Eq. (6.32) implies restoring the global version of the local estimate rather than the global estimate from the local estimate.

Substituting Eq. (6.31) into Eq. (6.23) and Eq. (6.32) into Eq. (6.24) yields

$$\mathbf{P}_j^{-1}(t_k^+) = \mathbf{P}_j^{-1}(t_k^-) - [\mathbf{D}_j^{\mathrm{T}}]^*\{\sum_{i=1}^{N}\mathbf{D}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_i^+) - \mathbf{P}_i^{-1}(t_k^-)]\mathbf{D}_i\}[\mathbf{D}_j]^* \tag{6.33}$$

Substituting Eq. (6.32) into Eq. (6.24) yields

$$\begin{aligned}\mathbf{P}_j^{-1}(t_k^+)\hat{\mathbf{x}}_j(t_k^+) &= \mathbf{P}_j^{-1}(t_k^-)\hat{\mathbf{x}}_j(t_k^-) \\ &\quad - [\mathbf{D}_j^{\mathrm{T}}]^*\sum_{i=1}^{N}\mathbf{D}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-)]\end{aligned} \tag{6.34}$$

Eqs. (6.33) and (6.34) create the modified Type II algorithm. It must be noted that the modified Type II algorithm has no global fusion model and all the local fusion filters have their own fusion models. This is different from the Type II algorithm where a global fusion model can be used in all the local fusion filters. A constraint on the choice of the local fusion model is that the dimension of the states in the local fusion model must be equal to that of the states in the corresponding local state filter model. The modified Type II algorithm is suitable for applications where the local state estimate is more important than the global state estimate.

The Type I and II algorithms and their modifications are widely used in target tracking and identification applications and can also be used to design and develop conventional aircraft integrated navigation systems where the main requirement is to determine the motion states of the aircraft centre of gravity. The traditional cascaded and federated filters are special examples of these distributed algorithms. However, a common constraint applied to both algorithms is that all the local filters must have the same dynamic model. This limits the application of the Type I and II algorithms.

6.2.2 State-Associated Distributed Fusion Algorithms

State-associated distributed fusion algorithms are presented for distributed sensor network systems, as illustrated in Figures 6.3 and 6.4 where all the sensor systems are distributed and observe their local dynamic motion states. Therefore, all the local dynamic models may be different from each other and the global dynamic model. Similar to the analysis in Section 6.2.1, this subsection describes two kinds of sensor fusion algorithms. The Type IA algorithm is used for the distributed system shown in Figure 6.3 while the Type IIA algorithm applies to the distributed system in Figure 6.4. In Figure 6.3, all the local estimates are transferred to the global fusion filter whereas in Figure 6.4, each local fusion filter assimilates all the local filter outputs.

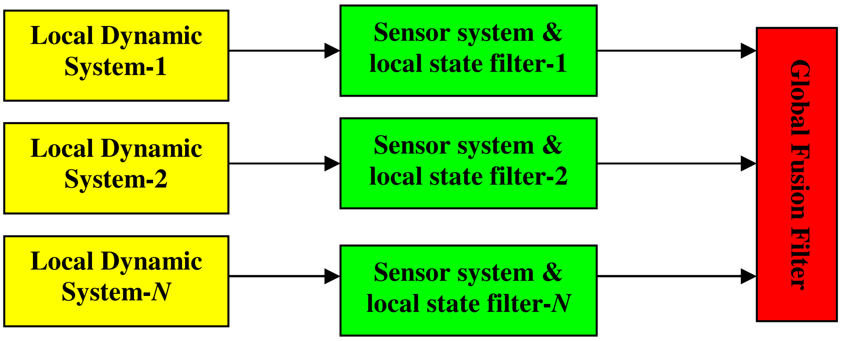

Local Dynamic System-1
Sensor system & local state filter-1
Local Dynamic System-2
Sensor system & local state filter-2
Local Dynamic System-*N*
Sensor system & local state filter-*N*
Global Fusion Filter

Figure 6.3 State-Associated One-Way Model

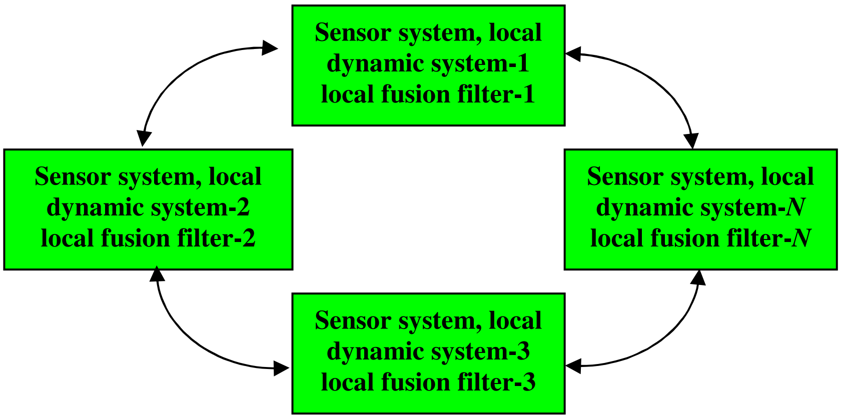

Sensor system, local dynamic system-1 local fusion filter-1
Sensor system, local dynamic system-2 local fusion filter-2
Sensor system, local dynamic system-*N* local fusion filter-*N*
Sensor system, local dynamic system-3 local fusion filter-3

Figure 6.4 State-Associated Two-Way Model

In these distributed systems, the following assumptions apply.

$\mathbf{x}_i(t_k) \neq \mathbf{x}_j(t_k) \neq \mathbf{x}(t_k)$

$\mathbf{w}_i(t_k) \neq \mathbf{w}_j(t_k) \neq \mathbf{w}(t_k)$

$\mathbf{\Phi}_i \neq \mathbf{\Phi}_j \neq \mathbf{\Phi}$

$\mathbf{B}_i(t_k) \neq \mathbf{B}_j(t_k) \neq \mathbf{G}(t_k)$

$\mathbf{C}_i(t_k) \neq \mathbf{C}_j(t_k) \neq \mathbf{H}_i(t_k)$

However, some relationships must exist between the local states and the global state. Assuming the local states are restored from the combination of the elements of the global state, then Eq. (6.21) can be rewritten as follows

$$\mathbf{x}_i(t_k) = \mathbf{T}_i\mathbf{D}_i\mathbf{x}(t_k) \tag{6.35}$$

where $\mathbf{T}_i$ is an invertible square matrix having the dimension consistent with the dimension of the local state $\mathbf{x}_i$.

Therefore, from Eq. (6.4), the local measurement matrix is given by

$$\mathbf{H}_i(t_k) = \mathbf{C}_i(t_k)\mathbf{T}_i\mathbf{D}_i \tag{6.36}$$

From Eqs. (6.23), (6.24), (6.35) and (6.36), the global update equations of the global fusion filter are as follows:

$$\mathbf{P}^{-1}(t_k^+) = \mathbf{P}^{-1}(t_k^-) + \sum_{i=1}^{N}\{\mathbf{D}_i^{\mathrm{T}}\mathbf{T}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_k^+) - \mathbf{P}_i^{-1}(t_k^-)]\mathbf{T}_i\mathbf{D}_i\} \tag{6.37}$$

$$\mathbf{P}^{-1}(t_k^+)\hat{\mathbf{x}}(t_k^+) = \mathbf{P}^{-1}(t_k^-)\hat{\mathbf{x}}(t_k^-) + \sum_{i=1}^{N}\mathbf{D}_i^{\mathrm{T}}\mathbf{T}_i^{\mathrm{T}}[\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^-) - \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-)] \tag{6.38}$$

The global time-update equations and the local time-update equations are based on the standard information filter. The fusion algorithm based on Eqs. (6.37) and (6.38) is known as the Type IA algorithm.

For the distributed system shown in Figure 6.4, from Eqs. (6.33), (6.34), (6.35), and (3.36), the Type IIA algorithm can be implemented as follows:

$$\begin{aligned}\mathbf{P}_i^{-1}(t_k^+) &= \mathbf{P}_i^{-1}(t_k^-) \\ &\quad - [\mathbf{D}_i^{\mathrm{T}}]^*[\mathbf{T}_i^{\mathrm{T}}]^{-1}\sum_{j=1}^{N}\{\mathbf{D}_j^{\mathrm{T}}\mathbf{T}_j^{\mathrm{T}}[\mathbf{P}_j^{-1}(t_k^+) - \mathbf{P}_j^{-1}(t_k^-)]\mathbf{T}_j\mathbf{D}_j\}[\mathbf{D}_i]^*[\mathbf{T}_i^{\mathrm{T}}]^{-1}\end{aligned} \tag{6.39}$$

$$
\begin{aligned}
\mathbf{P}_i^{-1}(t_k^+)\hat{\mathbf{x}}_i(t_k^+) &= \mathbf{P}_i^{-1}(t_k^-)\hat{\mathbf{x}}_i(t_k^-) \\
&\quad -[\mathbf{D}_i^{\mathrm{T}}]^*[\mathbf{T}_i^{\mathrm{T}}]^{-1}\sum_{j=1}^{N}\mathbf{D}_j^{\mathrm{T}}\mathbf{T}_j^{\mathrm{T}}[\mathbf{P}_j^{-1}(t_k^+)\hat{\mathbf{x}}_j(t_k^-)-\mathbf{P}_j^{-1}(t_k^-)\hat{\mathbf{x}}_j(t_k^-)]
\end{aligned}
\qquad (6.40)
$$

At each network node, the local filter estimate equations are given by the standard information filter algorithms. If the number of the local states is the same as the global state, then $\mathbf{D}_i = \mathbf{I}_i$ and $\mathbf{T}_i$ represents a transformation between the local states and the global state.

In comparison with the Type I and Type II algorithms, the Type IA and Type IIA algorithms are applicable to a wider range of distributed sensor systems. But they need to establish the transformation relationships between the local states and the global state. In many practical distributed sensor systems, this transformation $\mathbf{T}_i$ may be time varying $\mathbf{T}_i(t_k)$. Therefore, it is very important to develop this dynamic transformation in applications of the Type IA and Type IIA algorithms.

Comparing the Type I and Type IA algorithms with the Type II and Type IIA algorithms, the former needs to establish the global system model of the distributed sensor network system whereas the latter does not. From the viewpoint of fusion filter distributions, the Type I and Type IA algorithms can be also referred to as the centralised fusion filter algorithm and the Type II and Type IIA algorithms are known as the fully distributed fusion filter algorithm.

The Type IA and Type IIA algorithms can be used to resolve the problems of distributed controls and estimations where the local state information is particularly needed for local system controls and stabilisations and also for estimation of the local motion states.

In the following sections, a distributed inertial network system is presented as an example to explain the development of fully distributed fusion algorithms.

6.3 Distributed Inertial Sensing Models

A simplified version of the distributed inertial network system architecture shown in Figure 5.3 is illustrated in Figure 6.5 where three IMUs are located at different positions in an aircraft. These IMUs independently measure individual local qualities, but the measured or estimated states are not completely independent due to the rigid structure of the aircraft and are dynamically associated with each other. The development of this dynamic relationship between the local states and the measured qualities can drastically utilise the inertial information provided by the inertial network to detect and isolate sensor/system failures, and to particularly implement dynamic calibration and transfer alignments between the various inertial systems. Consequently, this dynamic relationship can be used to improve the required navigation performance in terms of the RNP parameters and to greatly increase the fault tolerance of an aircraft navigation system. This dynamic relationship can be established by the development of the rotational and translational transformations between the node frames.

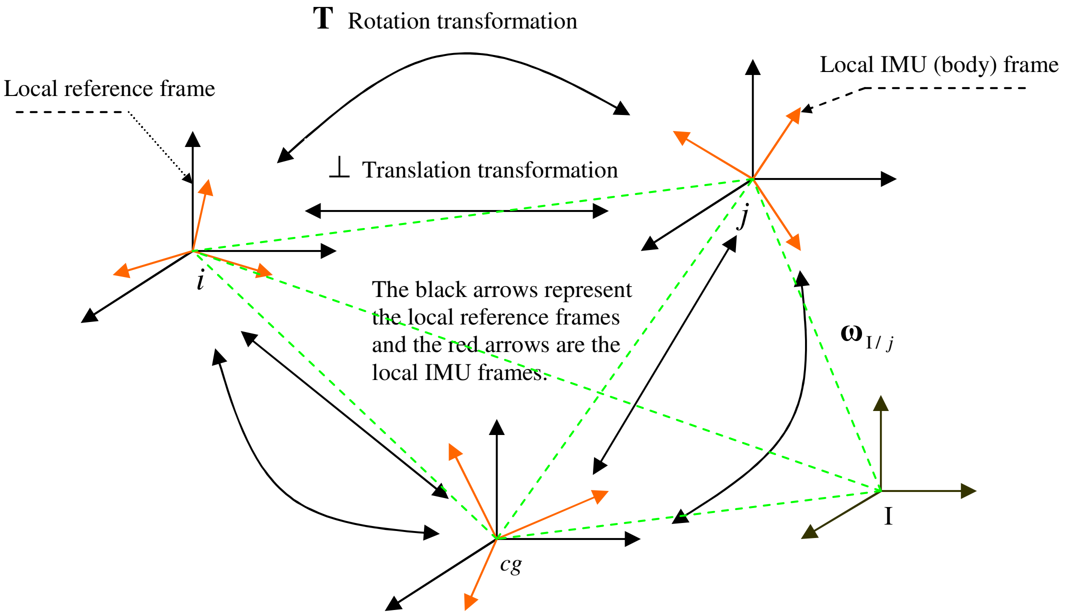

T Rotation transformation
Local reference frame
Local IMU (body) frame
⊥ Translation transformation
i
j
The black arrows represent the local reference frames and the red arrows are the local IMU frames.
$\omega_{I/j}$
I
cg

Figure 6.5 Relationships among IMU Nodes

In Figure 6.5, the IMU frames and the corresponding local reference frames are indicated. Let I be the inertial reference frame *cg* is the master IMU node located at the aircraft centre of gravity and its local body frame; *i* and *j* represent the slave IMU nodes and their individual local body frames(in this thesis, the IMU frames are assumed to be aligned with the local body frames, otherwise, fixed transformations are needed to align these two frames). ⊥ denotes a translational transformation, for example, $\perp_i^j$ is the translation vector from nodes i to j $\mathbf{T}$ represents a rotational transformation, for example, the transformation from i to j is denoted by $\mathbf{T}_i^j$. Exchanging the superscript and the subscript of a transformation represents the inverse of the transformation, for example $(\mathbf{T}_i^j)^{-1} = \mathbf{T}_j^i$. Let the local reference frames at the nodes of i, j and cg be denoted by L^i, L^j and L^{cg}, then the transformations from the local reference frames to the local body frames are given by $\mathbf{T}_{L^i}^i$, $\mathbf{T}_{L^j}^j$ and $\mathbf{T}_{L^{cg}}^{cg}$. If the local level frames are used as the local reference frames, these rotation matrices indicate the orientations of the local body axes relative to the local reference frames. If one local reference frame is not the local-level frame, then its orientation should be known relative to the local-level frame. Because the local-level frames are defined by the geographic locations of IMU nodes, their orientation differences caused by the translation vectors between these nodes can be ignored. Consequently, in this thesis it is assumed that the local-level frames located at all inertial network nodes are identical, that is $L^j = L^i = L^{cg}$.

Let the relative rotation of one IMU frame i with respect to another frame j be $\boldsymbol{\omega}_{j/i}$ and its inverse rotation be $\boldsymbol{\omega}_{i/j} = -\boldsymbol{\omega}_{j/i}$. From the theory of multi-body rotation[100], the absolute angular velocity of each IMU frame in the inertial network system is the sum of the absolute angular velocity of the other IMU frame and the relative angular velocity between these two frames, and is generally written in the following form:

$$\boldsymbol{\omega}_{\mathrm{I}/i} = \boldsymbol{\omega}_{\mathrm{I}/cg} + \boldsymbol{\omega}_{cg/i} \tag{6.41}$$

$$\boldsymbol{\omega}_{\mathrm{I}/j} = \boldsymbol{\omega}_{\mathrm{I}/cg} + \boldsymbol{\omega}_{cg/j} \tag{6.42}$$

$$\boldsymbol{\omega}_{\mathrm{I}/cg} = \boldsymbol{\omega}_{\mathrm{I}/j} + \boldsymbol{\omega}_{j/cg} \tag{6.43}$$

The terms on the left side of the above equations are measured by the corresponding local IMUs. The rotational transformations among the local IMU (local body) frames depend on the relative angular velocities between these frames.

Determination of the stationary and dynamic relationships among the network nodes is discussed in the following subsections.

6.3.1 Stationary Inertial Sensing Model

In this method, the aircraft is assumed to be a completely rigid body to simplify the analysis. Based on this assumption, there is no relative rotation motion between the local node body frames. Therefore, the dynamic relationships between different IMU frames can be described by stationary rotation and translation transformations from one node frame to the other node frame. These transformations can be measured precisely and determined after the IMUs have been installed in an aircraft.

When the local state $\mathbf{x}$ is a rate vector, such as acceleration, velocity or angular velocity, a rotation matrix is used to complete the rotation transformation from one node frame to the other node frame as follows:

$$\mathbf{x}_i = \mathbf{T}_j^i \mathbf{x}_j \tag{6.44}$$

If the local state is a displacement vector, a rotation matrix can be combined with a translation vector together to complete the transformation from one frame to the other frame as follows:

$$\mathbf{x}_i = \mathbf{T}_j^i \mathbf{x}_j + \perp_j^i \tag{6.45}$$

When the local states are the Euler angles, the attitude matrix transformation from one frame to the other frame has the following form:

$$\mathbf{T}_{L^i}^i = \mathbf{T}_j^i \mathbf{T}_{L^j}^j \tag{6.46}$$

where the states $\mathbf{x}_i$ and $\mathbf{x}_j$ are expressed in their individual local frames and $\mathbf{T}_j^i$ and $\perp_j^i$ are known as the rotation matrix and the translation vector, respectively.

At an IMU node, the measured inertial states, accelerations and angular rates, are expressed in terms of the local body frames and the IMU outputs are represented in the inertial instrument frames. The transformation between the inertial instrument frame and the local body frame, for example, at the cg node, is given by $\mathbf{H}_{imu^{cg}}^{cg}$ where the subscript imu^{cg} denotes the IMU instrument frame at the cg node. The matrix $\mathbf{H}_{imu^{cg}}^{cg}$ depends on the IMU configuration and can be dynamically reconfigured if the IMU is an SRIMU. From Eq. (5.2b), the measurement of the cg IMU can be rewritten as

$$\mathbf{m}_{imu^{cg}} = \mathbf{H}_{cg}^{imu^{cg}} \mathbf{x}_{cg} \tag{6.47}$$

By the rotational transformations, the measurements provided by IMUs i and j can be represented in terms of the local body frame at the node cg as follows:

$$\mathbf{m}_{imu^{i}} = \mathbf{H}_{i}^{imu^{i}} \mathbf{x}_{i} = \mathbf{H}_{i}^{imu^{i}} \mathbf{T}_{cg}^{i} \mathbf{x}_{cg} \tag{6.48}$$

$$\mathbf{m}_{imu^{j}} = \mathbf{H}_{j}^{imu^{j}} \mathbf{x}_{j} = \mathbf{H}_{j}^{imu^{j}} \mathbf{T}_{cg}^{j} \mathbf{x}_{cg} \tag{6.49}$$

Eqs. (6.48) and (6.49) mean that the node cg assimilates the inertial measurement information from the slave nodes i and j. Therefore, the inertial measurement at the node cg can be represented as

$$\mathbf{m}_{cg} \equiv \begin{bmatrix} \mathbf{m}_{imu^{cg}} \\ \mathbf{m}_{imu^{i}} \\ \mathbf{m}_{imu^{j}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{cg}^{imu^{cg}} \\ \mathbf{H}_{i}^{imu^{i}} \mathbf{T}_{cg}^{i} \\ \mathbf{H}_{j}^{imu^{j}} \mathbf{T}_{cg}^{j} \end{bmatrix} \mathbf{x}_{cg} = \mathbf{H}_{cg} \mathbf{x}_{cg} \tag{6.50}$$

Similarly, the inertial measurements at the nodes i and j are as follows:

$$\mathbf{m}_{i} \equiv \begin{bmatrix} \mathbf{m}_{imu^{cg}} \\ \mathbf{m}_{imu^{i}} \\ \mathbf{m}_{imu^{j}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{cg}^{imu^{cg}} \mathbf{T}_{i}^{cg} \\ \mathbf{H}_{i}^{imu^{i}} \\ \mathbf{H}_{j}^{imu^{j}} \mathbf{T}_{i}^{j} \end{bmatrix} \mathbf{x}_{i} = \mathbf{H}_{i} \mathbf{x}_{i} \tag{6.51}$$

$$\mathbf{m}_{j} \equiv \begin{bmatrix} \mathbf{m}_{imu^{cg}} \\ \mathbf{m}_{imu^{i}} \\ \mathbf{m}_{imu^{j}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{cg}^{imu^{cg}} \mathbf{T}_{j}^{cg} \\ \mathbf{H}_{i}^{imu^{i}} \mathbf{T}_{j}^{i} \\ \mathbf{H}_{j}^{imu^{j}} \end{bmatrix} \mathbf{x}_{j} = \mathbf{H}_{j} \mathbf{x}_{j} \tag{6.52}$$

Eqs. (6.50), (6.51) and (6.52) imply that each node shares the same redundant inertial measurements even if all the IMUs are traditionally orthogonal systems. Therefore, the SRIMU FDI algorithms developed in Chapter 5 can be used directly to detect and isolate inertial sensor failures in an inertial network system. Various weighted least-squares estimators can be used to estimate the inertial state. This procedure of data assimilation and least-squares estimation is referred to as inertial data fusion. The inertial data fusion procedure increases the measurement accuracy of each IMU and consequently improves the performance of the navigation system and the accuracy of the local state estimation.

One node estimates its local attitude matrix from the assimilated inertial measurements. From Eq. (6.46), it can also assimilate the attitude information from the other nodes using the following attitude matrices.

At node i,

$$\mathbf{T}_{L^i}^{i} \tag{6.53}$$

$$\mathbf{T}_{L^j}^{i} = \mathbf{T}_{j}^{i}\mathbf{T}_{L^j}^{j} \tag{6.54}$$

$$\mathbf{T}_{L^{cg}}^{i} = \mathbf{T}_{cg}^{i}\mathbf{T}_{L^{cg}}^{cg} \tag{6.55}$$

At node j,

$$\mathbf{T}_{L^j}^{j} \tag{6.56}$$

$$\mathbf{T}_{L^i}^{j} = \mathbf{T}_{i}^{j}\mathbf{T}_{L^i}^{i} \tag{6.57}$$

$$\mathbf{T}_{L^{cg}}^{j} = \mathbf{T}_{cg}^{j}\mathbf{T}_{L^{cg}}^{cg} \tag{6.58}$$

At node cg,

$$\mathbf{T}_{L^{cg}}^{cg} \tag{6.59}$$

$$\mathbf{T}_{L^i}^{cg} = \mathbf{T}_{i}^{cg}\mathbf{T}_{L^i}^{i} \tag{6.60}$$

$$\mathbf{T}_{L^j}^{cg} = \mathbf{T}_{j}^{cg}\mathbf{T}_{L^j}^{j} \tag{6.61}$$

Therefore, the redundant attitude information at each node can be fused to increase the accuracy of the local attitude estimates.

6.3.2 Dynamic Transformation Model

Although the assumption of a rigid body aircraft can apply to a wide range of applications in aircraft navigation and control systems, this assumption may be invalid in many military aircraft navigation and control systems because high-speed flight and high dynamic manoeuvres can cause the aircraft body to flex with flight conditions. The rotational transformations given in the above section are no longer stationary but are time-varying dynamic matrices. If the flexible structure character of an aircraft is ignored, the above assimilation equations will introduce errors in the rotation transformation $\Delta\mathbf{T}$, leading to larger errors in the estimates of the local states. Therefore, it is necessary to develop the dynamic relationships between the network nodes and to estimate these dynamic transformation matrices in flight. Two methods are presented in this thesis to determine these rotational transformations. One method establishes analytical models of the rotation matrices while the other method is an iterative processing method. In both methods, it is assumed that all initial transformation matrices are known. This assumption is reasonable because an initial transformation matrix can be approximated by a stationary transformation matrix, as given in Section 6.3.1.

The **iterative processing method** for the determination of the transformation matrices is based on Eq. (6.46) where the local attitude matrices at the IMU nodes are obtained by invoking the inertial attitude determination algorithms. The dynamic transformation matrices are then estimated from the computed local attitude matrices. The architecture of this iterative algorithm is illustrated in Figure 6.6.

Because the dynamic change of a rotation matrix relative to its initial matrix occurs over a small dynamic range, the estimated transformation matrix $\hat{\mathbf{T}}_{i\ k}^{j}$ can be expressed by the combination of the previous estimated rotation matrix and a small angle displacement vector $\boldsymbol{\psi}_{i\ k}^{j}$. Therefore, the estimated rotation matrix at the current time can be equivalently rewritten as

$$\hat{\mathbf{T}}_{i\ k}^{j} = \hat{\mathbf{T}}_{L^j}^{j}\hat{\mathbf{T}}_{i}^{L^i} = \hat{\mathbf{T}}_{i\ k-1}^{j}(\mathbf{I} + \boldsymbol{\psi}_{i\ k}^{j}\times) \tag{6.62}$$

This process is repeated until the norm of $\boldsymbol{\psi}_{i\ k}^{j}$ is less than a specified value; the current transformation matrix can then be determined.

From Figure 6.6, this iterative process is a time-consuming computation because each iteration must perform the inertial attitude determination algorithms for all the IMU nodes. The main advantage of this method is that the errors of the transformation matrix estimates are independent of the dynamic models of the transformation matrices. In addition, the inertial attitude determination algorithms play the role of a noise filter, which can reduce the effect of the IMU measurement noise on the rotation matrix estimates. However, uncertainties in the local attitude matrix estimates will contribute to errors in the rotation matrix estimates.

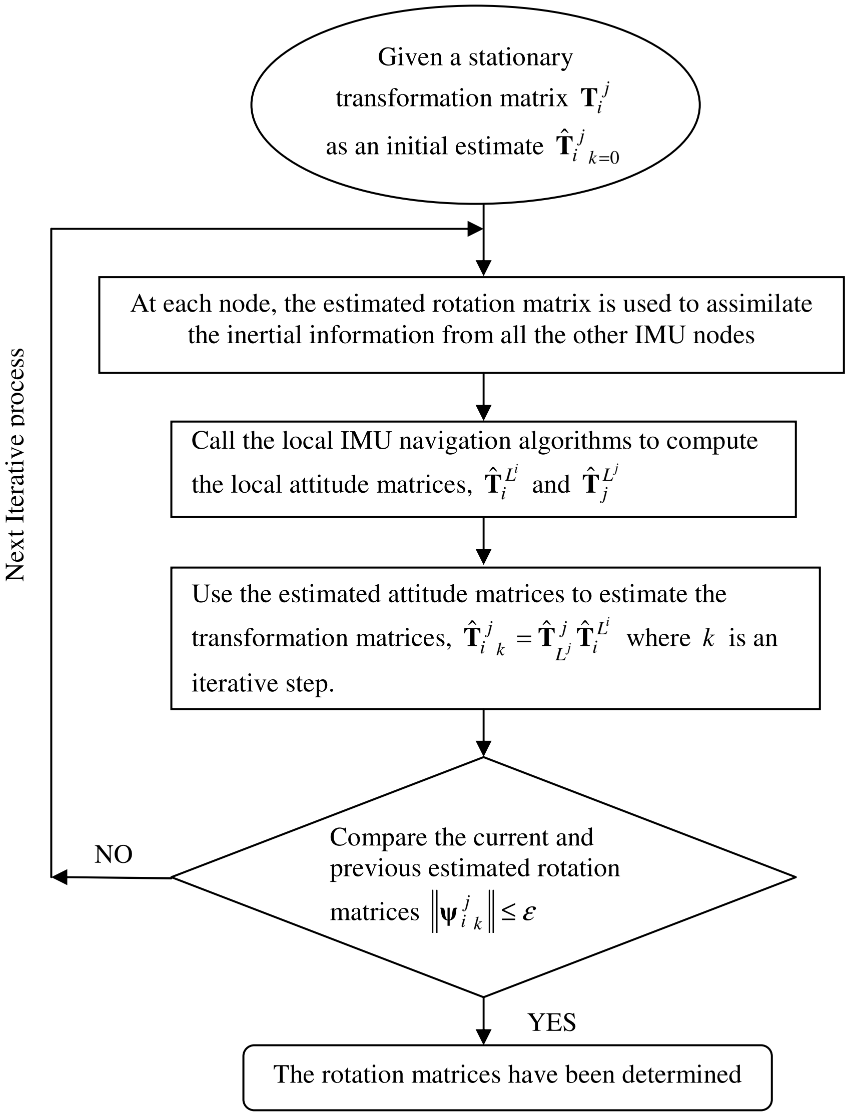

Given a stationary transformation matrix $\mathbf{T}_i^j$ as an initial estimate $\hat{\mathbf{T}}_{i\ k=0}^{j}$
At each node, the estimated rotation matrix is used to assimilate the inertial information from all the other IMU nodes
Call the local IMU navigation algorithms to compute the local attitude matrices, $\hat{\mathbf{T}}_i^{L^i}$ and $\hat{\mathbf{T}}_j^{L^j}$
Use the estimated attitude matrices to estimate the transformation matrices, $\hat{\mathbf{T}}_{i\ k}^{j} = \hat{\mathbf{T}}_{L^j}^{j}\hat{\mathbf{T}}_i^{L^i}$ where k is an iterative step.
Compare the current and previous estimated rotation matrices $\left\|\boldsymbol{\psi}_{i\ k}^{j}\right\| \leq \varepsilon$
NO
YES
The rotation matrices have been determined
Next Iterative process

Figure 6.6 Iterative Computation of Rotation Matrices

To deduce the ***analytical dynamic models*** of the transformation matrices, the *cg* body frame is used as a reference frame for the relative rotation motion of all the other local frames and the measured angular velocities. Therefore, Eqs. (6.41) to (6.43) can be rewritten in the matrix form of the angular velocity vectors as follows:

$$\boldsymbol{\Omega}_{\mathrm{I}/i}^{cg} = \boldsymbol{\Omega}_{\mathrm{I}/cg}^{cg} + \boldsymbol{\Omega}_{cg/i}^{cg} \tag{6.63}$$

$$\boldsymbol{\Omega}_{\mathrm{I}/j}^{cg} = \boldsymbol{\Omega}_{\mathrm{I}/cg}^{cg} + \boldsymbol{\Omega}_{cg/j}^{cg} \tag{6.64}$$

$$\boldsymbol{\Omega}_{\mathrm{I}/i}^{cg} = \boldsymbol{\Omega}_{\mathbf{I}/j}^{cg} + \boldsymbol{\Omega}_{j/i}^{cg} \tag{6.65}$$

where $\mathbf{\Omega}$ is the skew-symmetric matrix of the corresponding angular velocity vector $\boldsymbol{\omega}$. The superscript cg denotes that the angular velocity vectors are expressed in terms of the local cg body coordinates.

From the attitude matrix differential equation given in Eq. (4.22), Eq. (6.63) can be written as

$$\mathbf{\Omega}^{cg}_{\mathrm{I}/i} = \mathbf{\Omega}^{\mathrm{cg}}_{\mathrm{I}/cg} - \dot{\mathbf{T}}^{i}_{cg}\mathbf{T}^{cg}_{i} \tag{6.66}$$

Therefore

$$\dot{\mathbf{T}}^{i}_{cg} = (\mathbf{\Omega}^{cg}_{\mathrm{I}/cg} - \mathbf{\Omega}^{\mathrm{cg}}_{\mathrm{I}/i})\mathbf{T}^{i}_{cg} \tag{6.67}$$

$$\mathbf{\Omega}^{cg}_{\mathrm{I}/i} = \mathbf{T}^{cg}_{i}\mathbf{\Omega}^{i}_{\mathrm{I}/i}\mathbf{T}^{i}_{cg} \tag{6.68}$$

Similarly, the differential equation of the rotation matrix $\mathbf{T}^{j}_{cg}$ is as follows:

$$\dot{\mathbf{T}}^{j}_{cg} = (\mathbf{\Omega}^{cg}_{\mathrm{I}/cg} - \mathbf{\Omega}^{\mathrm{cg}}_{\mathrm{I}/j})\mathbf{T}^{j}_{cg} \tag{6.69}$$

$$\mathbf{\Omega}^{cg}_{\mathrm{I}/j} = \mathbf{T}^{cg}_{j}\mathbf{\Omega}^{j}_{\mathrm{I}/j}\mathbf{T}^{j}_{cg} \tag{6.70}$$

where $\mathbf{\Omega}^{cg}_{\mathrm{I}/cg}$, $\mathbf{\Omega}^{i}_{\mathrm{I}/i}$ and $\mathbf{\Omega}^{j}_{\mathrm{I}/j}$ are estimated from the IMU measurements at the nodes cg, i and j, as given in Eq. (5.3).

The rotation transformation matrix between j and i is then computed by the following equation.

$$\mathbf{T}^{i}_{j} = \mathbf{T}^{i}_{cg}\mathbf{T}^{cg}_{j} \tag{6.71}$$

Obviously, the dynamic models of the rotation transformations are non-linear matrix differential equations. The initial values of the matrix differential equations are given by the stationary transformations. These differential equations have to be iteratively resolved at the measurement time until the solutions become stable.

In comparison with the iterative processing method, the main advantage of the analytical method is that the time-consuming iterative computation of complex inertial attitude determination algorithms at all the IMU nodes is avoided. However, because the IMU outputs are directly used to drive the rotation matrix differential equations, the IMU measurement errors and noise may affect the accuracy of the solution of the rotation matrices. As a result, data pre-processing filters are needed to eliminate abnormal IMU measurement noise.

6.4 Distributed Inertial Network Fusion Algorithms

From Section 6.3, each node in the inertial network system can assimilate both the local measurements and the local estimates. Therefore, a two-stage data fusion strategy is presented to develop distributed data fusion filter algorithms. The first-stage fusion processes the inertial network measurement at all the nodes to obtain the more accurate local inertial state vectors, known as distributed inertial measurement fusion. The second-stage fusion is to increase the accuracy of the local system state estimates at all the network nodes and to enhance the fault tolerance of the inertial network system, referred to as distributed state fusion.

6.4.1 Distributed Inertial Data Fusion Algorithm

Assume that all the local IMUs are independent of each other and their measurements have a Gaussian probability distribution. Then the errors of the local inertial state estimates are also a Gaussian distributed random vector from Eq. (5.4). Therefore, the probability density function of the local inertial state is

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^3 \det \mathbf{P}_\text{x}}} \exp[-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^\text{T} \mathbf{P}_\text{x}^{-1}(\mathbf{x}-\hat{\mathbf{x}})] \qquad (6.72)$$

where $\mathbf{x}$ is an 3-dimensional local inertial state vector, for example, the acceleration or angular rate vector, and $\mathbf{P}_\text{x}$ is the covariance matrix of the error of the local inertial state estimate. From Eq. (5.5)

$$\mathbf{P}_\text{x} = (\mathbf{H}^\text{T}\mathbf{H})^{-1}\mathbf{H}^\text{T}\mathbf{R}\mathbf{H}(\mathbf{H}^\text{T}\mathbf{H})^{-1} = \mathbf{H}^*\mathbf{R}[\mathbf{H}^*]^\text{T} \qquad (6.73)$$

The objective of the inertial measurement fusion is to generate optimal estimates of all the local inertial states. Defining an optimisation criterion that maximises the following conditional probability

$$P(\mathbf{x}|\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j, \hat{\mathbf{x}}_{cg})$$

From the assumption that all the IMU measurements are independent, the conditional probability density function of the true local inertial state at each IMU node can be represented as follows:

$$p(\mathbf{x}|\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j,\hat{\mathbf{x}}_{cg}) = p(\mathbf{x}) = p(\mathbf{x}|\hat{\mathbf{x}}_i)p(\mathbf{x}|\hat{\mathbf{x}}_j)p(\mathbf{x}|\hat{\mathbf{x}}_{cg}) \tag{6.74}$$

Applying the maximum likelihood estimator to Eq. (6.74) and considering Eqs. (6.50) to (6.52), the inertial measurement fusion equations at each IMU node can be derived as follows:

$$\mathbf{x}_J = \left[\sum_{l=i,j,cg} \mathbf{T}_l^J \mathbf{P}_{\mathrm{x},l}^{-1} \mathbf{T}_J^l\right]^{-1} \sum_{l=i,j,cg} \mathbf{T}_l^J \mathbf{P}_{\mathrm{x},l}^{-1} \mathbf{H}_l^* \mathbf{m}_l \qquad J = i, j, cg \tag{6.75}$$

$$\mathbf{x}_J = \left[\sum_{l=i,j,cg} \mathbf{T}_l^J \mathbf{P}_{\mathrm{x},l}^{-1} \mathbf{T}_J^l\right]^{-1} \sum_{l=i,j,cg} \mathbf{T}_l^J \mathbf{H}_l^{\mathrm{T}} \mathbf{R}_l^{-1} \mathbf{m}_l \tag{6.76}$$

$$\mathbf{P}_{\mathrm{x},J}^{-1} = \sum_{l=i,j,cg} \mathbf{T}_l^J \mathbf{P}_{\mathrm{x},l}^{-1} \mathbf{T}_J^l \qquad J = i, j, cg \tag{6.77}$$

Eqs. (6.75) or (6.76) and (6.77) are the inertial measurement fusion algorithm at each IMU node. Any other methods, for example various weighted least-squares methods, which are used to resolve Eqs. (6.50) to (6.52), can be classified as the inertial measurement fusion. It should be noted that the inertial measurement fusion is mainly used to provide highly reliable local inertial state estimates. However, its outputs can also aid FDI systems to detect and isolate inertial sensor failures. The inertial measurement fusion can be considered as a pre-processing procedure for the second-stage fusion.

6.4.2 Distributed State Fusion Filter Algorithm

The architecture of the state fusion filter algorithm at each node is illustrated in Figure 6.7 where the local Kalman filter uses the assimilated sensor measurements to estimate the local states. The local fusion filter combines the local estimate and the assimilated estimates from the nodes to update the local estimates.

At each node of the inertial network system, the local Kalman filter model can be described as follows:

$$\mathbf{x}_J(t_k) = \mathbf{\Phi}_J(t_k,t_{k-1})\mathbf{x}_J(t_{k-1}) + \mathbf{G}_J(t_{k-1})\mathbf{w}_J(t_{k-1}) \tag{6.78}$$

$$\mathbf{z}_J(t_k) = \mathbf{H}_J(t_k)\mathbf{x}_J(t_k) + \boldsymbol{\upsilon}_J(t_k) \tag{6.79}$$

where $J = i, j, cg$ denote the IMU nodes.

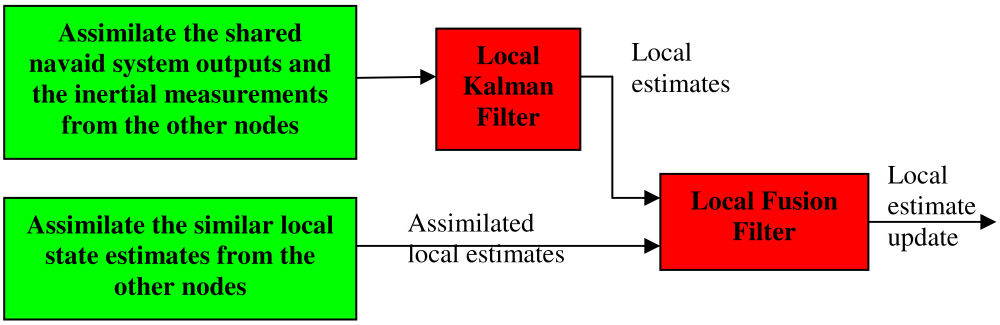

Assimilate the shared navaid system outputs and the inertial measurements from the other nodes
Local Kalman Filter
Local estimates
Local Fusion Filter
Local estimate update
Assimilate the similar local state estimates from the other nodes
Assimilated local estimates

Figure 6.7 State Fusion Algorithm Architecture

The navigation algorithms and the SRIMU error dynamic models developed in Chapter 4 can be applied to all the IMU nodes once the corresponding coordinate frames are specified. But, each local dynamic model describes its local state, which is different from the local states described by the other dynamic models. The local state $\mathbf{x}_J$ can be decomposed into the local system state $\mathbf{x}_{o,J}$ and the local sensor error state $\mathbf{x}_{s,J}$, that is

$$\mathbf{x}_J = \begin{bmatrix} \mathbf{x}_{o,J}^{\mathrm{T}} & \mathbf{x}_{s,J}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{6.80}$$

All the local system states at the network nodes are referred to as the similar states. The dynamic matrices established in Section 6.3 provide the transformations among the similar states.

$\mathbf{z}_J$ can be decomposed into three sub-vectors as follows:

$$\mathbf{z}_J = \begin{bmatrix} \mathbf{z}_{L,J}^{\mathrm{T}} & \mathbf{z}_{S,J}^{\mathrm{T}} & \mathbf{z}_{A,J}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{6.81}$$

where $\mathbf{z}_{L,J}$ is the measurement vector provided by local navaid sensor systems, $\mathbf{z}_{S,J}$ is the measurement vector given by the commonly-shared navaid systems and $\mathbf{z}_{A,J}$ is the combination of all the inertial measurements assimilated from the IMU nodes. The normalised measurement models of several navaid systems have been developed in Section 4.6 and the normalised measurement models of redundant inertial sensor systems have been given in Section 5.5.

Because these three forms of measurements are independent of each other, the terms in Eq. (6.79) can be decomposed into the following forms:

$$\mathbf{H}_J = \begin{bmatrix}\mathbf{H}_{L,J}^{\mathrm{T}} & \mathbf{H}_{S,J}^{\mathrm{T}} & \mathbf{H}_{A,J}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{6.82}$$

$$\boldsymbol{\upsilon}_J = \begin{bmatrix}\boldsymbol{\upsilon}_{L,J}^{\mathrm{T}} & \boldsymbol{\upsilon}_{S,J}^{\mathrm{T}} & \boldsymbol{\upsilon}_{A,J}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{6.83}$$

$$\mathbf{R}_J = blockdiag(\mathbf{R}_{L,J}, \mathbf{R}_{S,J}, \mathbf{R}_{A,J}) \tag{6.84}$$

All the local Kalman filters process these three forms of the measurements to obtain the local estimates, including the local states and covariances, as follows:

$$\hat{\mathbf{x}}_J(t_k^-) = \boldsymbol{\Phi}_J(t_k, t_{k-1})\bar{\mathbf{x}}_J(t_{k-1}) \tag{6.85}$$

$$\mathbf{P}_J(t_k^-) = \boldsymbol{\Phi}_J(t_k, t_{k-1})\overline{\mathbf{P}}_J(t_{k-1}^+)\boldsymbol{\Phi}_J^{\mathrm{T}}(t_k, t_{k-1}) + \mathbf{G}_J(t_{k-1})\mathbf{Q}_J(t_{k-1})\mathbf{G}_J^{\mathrm{T}}(t_{k-1}) \tag{6.86}$$

$$\begin{aligned}\mathbf{P}_J^{-1}(t_k^+) &= \mathbf{P}_J^{-1}(t_k^-) + \mathbf{H}_J^{\mathrm{T}}(t_k)\mathbf{R}_J^{-1}(t_k)\mathbf{H}_J(t_k) \\ &= \mathbf{P}_J^{-1}(t_k^-) + \sum_{l=L,S,A}\mathbf{H}_{l,J}^{\mathrm{T}}(t_k)\mathbf{R}_{l,J}^{-1}(t_k)\mathbf{H}_{l,J}(t_k)\end{aligned} \tag{6.87}$$

$$\begin{aligned}\mathbf{P}_J^{-1}(t_k^+)\hat{\mathbf{x}}_J(t_k^+) &= \mathbf{P}_J^{-1}(t_k^-)\hat{\mathbf{x}}_J(t_k^-) + \mathbf{H}_J^{\mathrm{T}}(t_k)\mathbf{R}_J^{-1}(t_k)\mathbf{z}_J(t_k) \\ &= \mathbf{P}_J^{-1}(t_k^-)\hat{\mathbf{x}}_J(t_k^-) + \sum_{l=L,S,A}\mathbf{H}_{l,J}^{\mathrm{T}}(t_k)\mathbf{R}_{l,J}^{-1}(t_k)\mathbf{z}_{l,J}(t_k)\end{aligned} \tag{6.88}$$

where $\hat{\mathbf{x}}_J$ and $\mathbf{P}_J$ are the local estimates given by the local Kalman filter and $\bar{\mathbf{x}}_J$ and $\overline{\mathbf{P}}_J$ are the estimate updates given by the local fusion filter.

To update the locally estimated system states at each node using the similar state estimates assimilated from the other nodes, a state fusion filter is needed in each node. Defining a quadratic cost function at the node i as follows:

$$J_i = \sum_{J=i,j,cg}(\mathbf{T}_J^i\hat{\mathbf{x}}_{o,J}^+ - \mathbf{x}_{o,i})^{\mathrm{T}}\mathbf{T}_J^i\mathbf{P}_{o,J}^{-1}\mathbf{T}_i^J(\mathbf{T}_J^i\hat{\mathbf{x}}_{o,J}^+ - \mathbf{x}_{o,i}) \tag{6.89}$$

where $\hat{\mathbf{x}}_{o,J}^+$ is the local system state estimate given by the local Kalman filter, $\mathbf{x}_{o,i}$ is the true local system state at i, $\mathbf{P}_{o,J}$ is a sub-matrix of $\mathbf{P}_J$ and is associated with the local system state and J_i is a measure of the displacement of the local state from its true value.

The state fusion filter is designed to minimise J_i. This is referred to as the minimum weighted mean square error criterion.

Differentiating J_i and setting the result to zero yields

$$\bar{\mathbf{x}}_{o,i} = \overline{\mathbf{P}}_{o,i}^{-1}(\mathbf{P}_{o,i}^{-1}\hat{\mathbf{x}}_{o,i}^+ + \mathbf{T}_j^i\mathbf{P}_{o,j}^{-1}\hat{\mathbf{x}}_{o,j}^+ + \mathbf{T}_{cg}^i\mathbf{P}_{o,cg}^{-1}\hat{\mathbf{x}}_{o,cg}^+) \tag{6.90}$$

$$\overline{\mathbf{P}}_{o,i}^{-1} = (\mathbf{P}_{o,i}^{-1} + \mathbf{T}_j^i \mathbf{P}_{o,j}^{-1} \mathbf{T}_i^j + \mathbf{T}_{cg}^i \mathbf{P}_{o,cg}^{-1} \mathbf{T}_i^{cg}) \tag{6.91}$$

In a similar fashion, the similar state update equations at the nodes j and cg can are given as follows:

$$\overline{\mathbf{x}}_{o,j} = \overline{\mathbf{P}}_{o,j}^{-1}(\mathbf{P}_{o,j}^{-1}\hat{\mathbf{x}}_{o,j}^{+} + \mathbf{T}_i^j \mathbf{P}_{o,i}^{-1}\hat{\mathbf{x}}_{o,i}^{+} + \mathbf{T}_{cg}^j \mathbf{P}_{o,cg}^{-1}\hat{\mathbf{x}}_{o,cg}^{+}) \tag{6.92}$$

$$\overline{\mathbf{P}}_{o,j}^{-1} = (\mathbf{P}_{o,j}^{-1} + \mathbf{T}_i^j \mathbf{P}_{o,i}^{-1}\mathbf{T}_j^i + \mathbf{T}_{cg}^j \mathbf{P}_{o,cg}^{-1}\mathbf{T}_j^{cg}) \tag{6.93}$$

$$\overline{\mathbf{x}}_{o,cg} = \overline{\mathbf{P}}_{o,cg}^{-1}(\mathbf{P}_{o,cg}^{-1}\hat{\mathbf{x}}_{o,cg}^{+} + \mathbf{T}_j^{cg} \mathbf{P}_{o,j}^{-1}\hat{\mathbf{x}}_{o,j}^{+} + \mathbf{T}_i^{cg} \mathbf{P}_{o,i}^{-1}\hat{\mathbf{x}}_{o,i}^{+}) \tag{6.94}$$

$$\overline{\mathbf{P}}_{o,cg}^{-1} = (\mathbf{P}_{o,cg}^{-1} + \mathbf{T}_j^{cg} \mathbf{P}_{o,j}^{-1}\mathbf{T}_{cg}^j + \mathbf{T}_i^{cg} \mathbf{P}_{o,i}^{-1}\mathbf{T}_{cg}^i) \tag{6.95}$$

Eqs. (6.90) to (6.95) form the distributed state fusion filter algorithm. By examining the above analysis, this distributed state fusion algorithm actually consists of two fusion procedures, the measurement fusion given by Eqs. (6.87) and (6.88) and the state fusion given by Eqs. (6.90) to (6.95). The state fusion provides the redundancy of the similar system states. Therefore, this fusion method can greatly improve the fault tolerance of an inertial network system.

From Eqs. (6.85) and (6.86), the outputs of each fusion filter are fed back to its corresponding local Kalman filter. This feedback operation implements the dynamic transfer alignments between the node IMUs and also allows the local Kalman filter to accurately estimate and calibrate its sensor error state. Consequently, this distributed state fusion algorithm provides the capability to perform the dynamic alignment and calibration of the inertial systems in an inertial network system. Therefore, traditional inertial system alignment algorithms, for example, fine and transfer alignments, are no longer necessary in inertial network systems. The traditional in-flight alignments generally require an aircraft to perform some specified manoeuvres, which may lead to enormous risk especially to military aircraft and pilots. The elimination of the traditional alignment procedures allows aircraft to execute free flight and arbitrary manoeuvres.

6.5 Inertial Network Integrity Monitoring

From the inertial network dynamic models given by Eqs (6.78) and (6.79), the failures in the inertial network system can be classified as sensor system failures and local system state failures. From Eq. (6.81), the sensor system failures may result from the inertial sensor systems, the local navaid sensors and the commonly-shared navaid systems. Therefore, the purpose of inertial network integrity monitoring is to ensure all the local filters are able to provide the very reliable local system state estimates and to detect and isolate various sensor failures and the abnormal local system states from the inertial network system in a short time.

In this thesis, two integrity monitoring strategies have been suggested to guarantee the integrity of the inertial network system, including the sensor-level FDI and the system-level integrity monitoring, as shown in Figure 2.9 of the generalised MSDF model. The sensor-level FDI methods have been developed in Chapter 5 and are initially used to detect and isolate the inertial sensor failures in skewed redundant configurations or the GNSS signal failures in the redundant GNSS measurements. These methods can be directly applied to the inertial assimilation equations given by Eqs. (5.50) to (5.52) to detect and isolate the distributed inertial sensor failures in the inertial network system. In addition, many other methods have been suggested in the recent years for detecting and isolating the GNSS signal failures, for example, receiver autonomous integrity monitoring (RAIM) methods and aircraft autonomous integrity monitoring (AAIM) methods[83]. However, these methods cannot guarantee that the system state estimates computed by the local Kalman filters are reliable.

To verify the integrity of Kalman filters, several Kalman filter-based detection methods have been developed in the past. For example, a method, called multiple model adaptive filters, has been suggested where a bank of Kalman filters is used, each with a different model. The innovations of these filters are monitored and the conditional probability that each system model is correct is computed. This technique has the advantage of being able to provide reliable filter outputs and to isolate failed sensors and improper filter models. However, with the number of the filter states, the computations required by all the node processors may be time-consuming.

Several failure detection filter methods[125-128] have also been suggested for the detection of actuator, plant and sensor failures in control systems. A detection filter is a full-order linear state-space observer. In detection filter design, the gain matrix must be chosen so that the residual vectors generated by certain actuator or sensor failures can be projected to lines and planes in the measurement space and each potential failure has a different fixed direction. Accordingly, the detection filter gain is adjusted in order to identify the feature of failure signals, but not to minimise the mean square error of estimation, as is done in a Kalman estimator. In addition, many detection filters assume that dynamic systems are noise free. Application of detection filters to inertial network integrity monitoring noticeably increases the complexity of data fusion algorithms in the network system. Other model-following approaches require that one Kalman filter performs the usual tracking and estimation, and the other filters are used to detect the presence of specified (or previously characterised) failure modes. Based on the above analysis, all of these three methods of Kalman filter-based failure detection are inappropriate for inertial network systems because they significantly increase the computation load at each node.

Therefore, in this section, easily implemented and real-time detection methods are developed not only to monitor the integrity of inertial network systems but also to reduce the computation load at the inertial network nodes.

6.5.1 Inertial Network Failure Model

As shown in the previous section, the local node system models in the inertial network system are described by the local IMU error dynamics. From the analysis in Chapter 4, if the inertial sensor failures occur, the local inertial navigation algorithms may produce abnormal similar system states, which further cause uncertainties of the state transition matrices of the local system models. If the navaid system failures occur, the local and commonly shared measurements may contain errors. Therefore, it can be assumed that the effect of the inertial sensor failures are considered as additional system state failures in the dynamic models whereas the navaid system failures are modelled as additional measurement failures in the measurement models of the local Kalman filters. Accordingly, a failure model of the Kalman filter at each node is established as follows:

$$\mathbf{x}_{o,J}(t_k) = \mathbf{\Phi}_{o,J}(t_k, t_{k-1})\mathbf{x}_{o,J}(t_{k-1}) + \boldsymbol{\delta}(t_k, \tau_{\mathbf{x},J})\mathbf{f}_{\mathbf{x}_o,J} + \mathbf{G}_{o,J}(t_{k-1})\mathbf{w}_{o,J}(t_{k-1}) \quad (6.96)$$

$$\mathbf{z}_{Aid,J}(t_k) = \mathbf{H}_{Aid,J}(t_k)\mathbf{x}_{o,J}(t_k) + \boldsymbol{\delta}(t_k, \tau_{Aid,J})\mathbf{f}_{Aid,J} + \boldsymbol{\upsilon}_{Aid,J}(t_k) \quad (6.97)$$

where $\mathbf{x}_{o,J}$ and $\mathbf{z}_{Aid,J}$ are the local system state and the navaid system measurement vectors at the node J, $\boldsymbol{\delta}$ is a diagonal matrix of Kronecker functions, $\mathbf{f}_{\mathbf{x}_o,J}$ is a time-variable local system state failure vector and $\mathbf{f}_{Aid,J}$ is a time-variable navaid system failure vector. This is a reduced-order filter model compared with the corresponding local filter.

Because the effects of the inertial sensor failures have been considered in the system dynamic model, the inertial data assimilation equations are not contained in the measurement model. Furthermore, the coefficient matrices of this failure model are the sub-matrices of the coefficient matrices of the corresponding local Kalman filter. Therefore, the outputs of the local Kalman filters are directly used to generate failure detection functions without additional computations.

Two efficient system-level methods are presented in this section to monitor the integrity of the inertial network system. One method is based on directly examining the consistency of the distributed state estimates whereas the other is to monitor the distributed filter innovations for detecting and even isolating the local navaid sensor failures and the commonly shared navaid systems failures.

6.5.2 Distributed State Consistency Monitoring

Assuming no sensor failures occur at any network node, then all the local similar system states must be consistent through the transformations among the local similar states. In the presence of any sensor system failures at one network node, this assumption will no longer be valid. Accordingly, the distributed state consistency monitoring method can employ combinations of the similar system state estimates computed by all the local filters to check the consistency of all the similar system states. Figure 6.8 shows the architecture of this integrity monitoring method.

At each node, the redundant local similar state estimate $\hat{\mathbf{x}}_{o,J}$ ($J = i, j, k$) can be represented as follows:

$$\hat{\mathbf{x}}_{o,i} = \mathbf{T}_J^i \hat{\mathbf{x}}_{o,J} + \boldsymbol{\varepsilon}_{\mathbf{x}_{o,i}}$$

$$\hat{\mathbf{x}}_{o,j} = \mathbf{T}_J^j \hat{\mathbf{x}}_{o,J} + \boldsymbol{\varepsilon}_{\mathbf{x}_{o,j}}$$

$$\hat{\mathbf{x}}_{o,cg} = \mathbf{T}_J^{cg} \hat{\mathbf{x}}_{o,J} + \boldsymbol{\varepsilon}_{\mathbf{x}_{o,cg}}$$

or

$$\begin{bmatrix} \hat{\mathbf{x}}_{o,i} \\ \hat{\mathbf{x}}_{o,j} \\ \hat{\mathbf{x}}_{o,cg} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_J^i \\ \mathbf{T}_J^i \\ \mathbf{T}_J^{cg} \end{bmatrix} \hat{\mathbf{x}}_{o,J} + \boldsymbol{\varepsilon}_J \tag{6.98}$$

where $\boldsymbol{\varepsilon}_{\mathbf{x}}$ is the error of the local similar state estimate with the known covariance $\mathbf{P}_{o,J}$, as computed by the local Kalman filter.

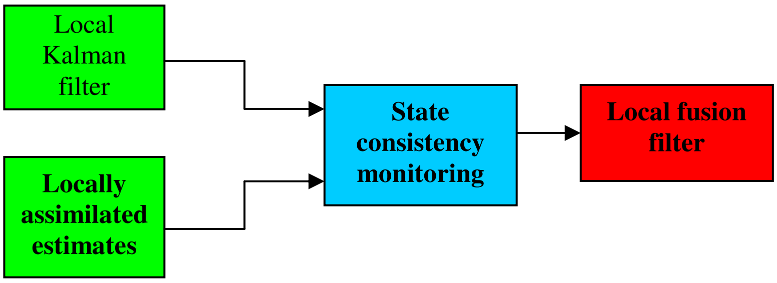


Figure 6.8 Architecture of State Consistency Monitoring

To check the consistency of the distributed attitude states, the quaternion form $\mathbf{q}$ of the local attitude matrix is used. Therefore, the redundant quaternion equations are derived as follows:

$$\mathbf{q}_i \overline{\mathbf{T}}_J^i = \overline{\mathbf{T}}_J^i \mathbf{q}_J$$

$$\mathbf{q}_j \overline{\mathbf{T}}_J^j = \overline{\mathbf{T}}_J^j \mathbf{q}_J$$

$$\mathbf{q}_{cg} \overline{\mathbf{T}}_J^{cg} = \overline{\mathbf{T}}_J^{cg} \mathbf{q}_J$$

or

$$\begin{bmatrix} \mathbf{q}_i \overline{\mathbf{T}}_J^i \\ \mathbf{q}_j \overline{\mathbf{T}}_J^j \\ \mathbf{q}_{cg} \overline{\mathbf{T}}_J^{cg} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{T}}_J^i \\ \overline{\mathbf{T}}_J^i \\ \overline{\mathbf{T}}_J^{cg} \end{bmatrix} \mathbf{q}_J \tag{6.99}$$

where $\overline{\mathbf{T}}_J^i$ is the quaternion transformation matrix corresponding to $\mathbf{T}_J^j$.

By structuring the redundant equations given in Eqs. (6.98) and (6.99), the problem of the distributed state consistency checking is similar to the sensor FDI problem described in Chapter 5. Therefore, applying the FDI methods developed in Chapter 5 to the redundant state equations, the state failures of the local Kalman filters can be detected and isolated. This is a simple and efficient method to monitor the integrity of all the local Kalman filters in real time without considering the sources of failures. From Figure 6.8, this state consistency checking can be combined with the local state fusion algorithm. If one local Kalman filter is diagnosed to be invalid, its outputs will be isolated from its related local fusion filter and the local fusion filter will use assimilated estimates to reconfigure the local similar system states according to Eqs. (6.90) to (6.95). This method is mainly used to monitor the integrity of the similar system states in the inertial network system and detect local Kalman filter failures. It should be noted that this method of integrity monitoring is based on the assumption that all the transformation matrices among the similar states are correct.

6.5.3 Distributed Filter Innovation Monitoring

The navaid system failure $\mathbf{f}_{Aid,J}$ appearing in the failure measurement model of Eq. (6.97) contains the local navaid sensor failures and the commonly-shared navaid system failures. Any failures resulting from the commonly-shared navaid systems are common-mode to all the local Kalman filters because such failures influence all the local systems. A common-mode failure causes the estimates of all the local Kalman filters to diverge from their anticipated values. The distributed state consistency monitoring method above cannot faithfully detect the local system state failures caused by common-mode failures. However, from the analysis given in Section 3.2.6, the Kalman filter innovation is independent of the actual measurements of the navaid systems. By monitoring the innovation generated by all the local Kalman filters, any failure caused by the local navaid sensors or the commonly shared navaid systems can be correctly detected.

From Eqs. (6.96) and (6.97), the failure filter innovation at J can be derived as follows:

$$\mathbf{r}_{f,J}(t_k) = \mathbf{r}_J(t_k) + \mathbf{H}_{o,J}(t_k)\boldsymbol{\delta}(t_k, \tau_{\mathbf{x},J})\mathbf{f}_{\mathbf{x}_o,J} + \boldsymbol{\delta}(t_k, \tau_{Aid,J})\mathbf{f}_{Aid,J} \qquad (6.100)$$

where the normal Kalman filter innovation $\mathbf{r}(t_k)$ is

$$\mathbf{r}_J(t_k) = \mathbf{H}_{o,J}\tilde{\mathbf{x}}_{o,J}(t_k) + \boldsymbol{\upsilon}_{o,J}(t_k) \tag{6.101}$$

$$\delta(t_k, \tau_{\mathbf{x},J})\mathbf{f}_{\mathbf{x}_o,J} = \begin{bmatrix} \delta_1(t_k, \tau_{\mathbf{x},J-1})\mathrm{f}_{\mathrm{x}_o,J\text{-}1} \\ \delta_2(t_k, \tau_{\mathbf{x},J-2})\mathrm{f}_{\mathrm{x}_o,J\text{-}2} \\ \\ \delta_n(t_k, \tau_{\mathbf{x},J-n})\mathrm{f}_{\mathrm{x}_o,J\text{-}n} \end{bmatrix} \tag{6.102}$$

$$\delta(t_k, \tau_{Aid,J})\mathbf{f}_{Aid,J} = \begin{bmatrix} \delta_1(t_k, \tau_{Aid,J-1})\mathrm{f}_{Aid,J\text{-}1} \\ \delta_2(t_k, \tau_{Aid,J-2})\mathrm{f}_{Aid,J\text{-}2} \\ \\ \delta_m(t_k, .\tau_{Aid,J-m})\mathrm{f}_{Aid,J\text{-}m} \end{bmatrix} \tag{6.103}$$

Exploiting Eq. (6.100) for detection and isolation of the navaid system failures and monitoring the integrity of the local Kalman filters, two assumptions are made:

- $\tau_{\mathbf{x},J} \neq \tau_{Aid,J}$. That is, the time $\tau_{\mathbf{x},J}$ of the occurrence of system state failure is different from the time $\tau_{Aid,J}$ of the occurrence of the navaid system failure, or
- Inertial sensor failures have been detected at the sensor-level FDI stage or by the inertial measurement fusion procedure.

Based on the above assumptions, a normalised quadratic innovation function of $\mathbf{r}_f$ can be used as a test statistic to test the filter integrity, as given in Eq. (3.32).

$$NQI_J(t_k) = \mathbf{r}_{f,J}{}^{\mathrm{T}}(t_k)\mathbf{S}_J^{-1}(t_k)\mathbf{r}_{f,J}(t_k) \tag{6.104}$$

If $NQI(t_k) \geq T_{NQI,J}$, then failures have occurred.

If $NQI(t_k) < T_{NQI}$, then no failure has taken place.

An advantage of this NQI testing is that its degree of freedom is equal to the number of the filter measurements. This means that the NQI testing can verify each measurement of the navaid systems one by one. Therefore, it can isolate each failure occurring in the navaid system measurements, for example, GNSS signal failures or Doppler radar signal failures. The sequential moving-window methods introduced in Chapter 5 can be used to pre-process the filter innovation in order to increase the reliability of NQI testing.

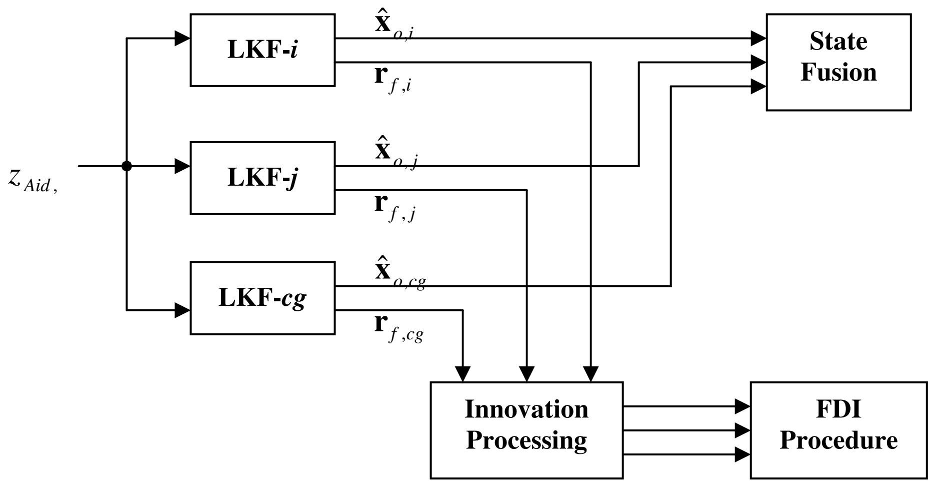


Figure 6.9 Architecture of Distributed Filter Innovation Monitoring

Without considering the above assumptions, from Eq. (6.100), the NQI testing examines the combined effect of the navaid system failures and the system state failure on the filter performance. When both the system state failures and the navaid system failures simultaneously occur, NQI testing only provides alarm information but cannot isolate the failed states or measurements. Therefore, the filter integrity monitoring should cooperate with sensor-level FDI procedures to ensure the integrity of an inertial network system.

6.6 Summary

This chapter has developed several distributed fusion algorithms for distributed multiple sensor navigation systems. The main activities covered include:

1. Introduction of four forms of the distributed sensor systems for navigation applications and the corresponding fusion filter algorithms, including the state-identical and state-associated distributed sensor systems, both with one-way and two-way data communication modes. Four fusion algorithms cover Type I, Type II, Type A and Type IIA, respectively corresponding to the above four distributed sensor systems.
2. Development of inertial network sensing models, including the stationary

inertial sensing model and the dynamic transformation model.

3. Study of two distributed inertial network fusion algorithms, including the distributed inertial data fusion algorithm and the distributed state fusion filter algorithm.
4. Introduction of two inertial network integrity monitoring algorithms, including the distributed state consistency monitoring algorithm and the distributed filter innovation monitoring algorithm.

Chapter 7

SIMULATION SYSTEM AND RESULTS

7.1 Introduction

This chapter develops a simulation system environment to test and evaluate the failure detection and isolation and integrity monitoring algorithms and the distributed data fusion algorithms developed in this thesis. Section 7.2 introduces the overall architecture of this software simulation system, the architectures of the SRIMU simulation system and the GPS simulation system, as well software development and evaluation strategies. The test results of MW-GLRT algorithms are given in Section 7.3. Several case studies of distributed data fusion algorithms are presented in Section 7.4. The results of the simulation studies are summarised in Section 7.5.

7.2 Simulation System Architecture

The software simulation system is of a modular system architecture consisting of the sub-modules, as shown in Figure 7.1 where the shadowed modules represent the functions to be implemented at different nodes of the inertial network. The functions of several main sub-modules are summarised below:

Trajectory Generator

The Trajectory Generator module generates the true 6 DOF parameters (three cg accelerations and three cg angular rates expressed in aircraft body frame), known as the inertial state, and the true aircraft cg-referenced navigation state, including aircraft position, velocity and attitude. These true parameters are used as reference values to evaluate the performance of the FDI and integrity monitoring algorithms, and the distributed data fusion algorithms. The aircraft dynamics is modelled by a set of ideal mathematical equations of the translational and angular motions without the consideration of the effects of any external disturbances on the aircraft motion. This assumption is rational because the outputs of this module, in this simulation system, are used only to drive other sensor system simulation modules rather than to design the aircraft control systems. In addition, because the specific force triad and the angular rate triad measured by the inertial sensors represent the driving forces of the translational and angular momentum equations of the aircraft, various disturbed aircraft motions can be equivalent to the inertial sensor system errors that are simulated in the inertial simulation system.

SRIMU Evaluator

This module evaluates the performance of individual redundant inertial system configurations in terms of the number of inertial sensors and specific criteria. Several criteria have been introduced in Chapter 5. This evaluator can evaluate up to 12 SRIMU configurations consisting of 3 to 6 inertial sensors, respectively. The outputs of this evaluator are specified configurations, represented by design matrices, which are then used in the SRIMU simulator to simulate realistic SRIMU measurements together with an inertial sensor error simulation module.

SRIMU Simulator

This module simulates the realistic SRIMU measurements. The dimension of the simulated SRIMU measurement vector depends on the design matrices set up in the SRIMU Evaluator. The inertial sensor errors mainly result from five error sources: bias, time-dependent drift, misalignment, scale factor errors and noise. These error sources are modelled by the random constant process, the first-order Markov process, the random walk process and Gaussian white noise. Inertial sensor failure modes are also simulated in this module. Sensor failures are classified as hard failures and time-drift soft failures for the evaluation of the performance of FDI algorithms.

To simulate an inertial network system, other local inertial states are derived from the inertial state at the aircraft cg node but take into account the dynamic transformations between the local node frames and the cg node frame. The dynamic transformations are modelled by the sinusoidal functions of different cycles plus the transformation noise.

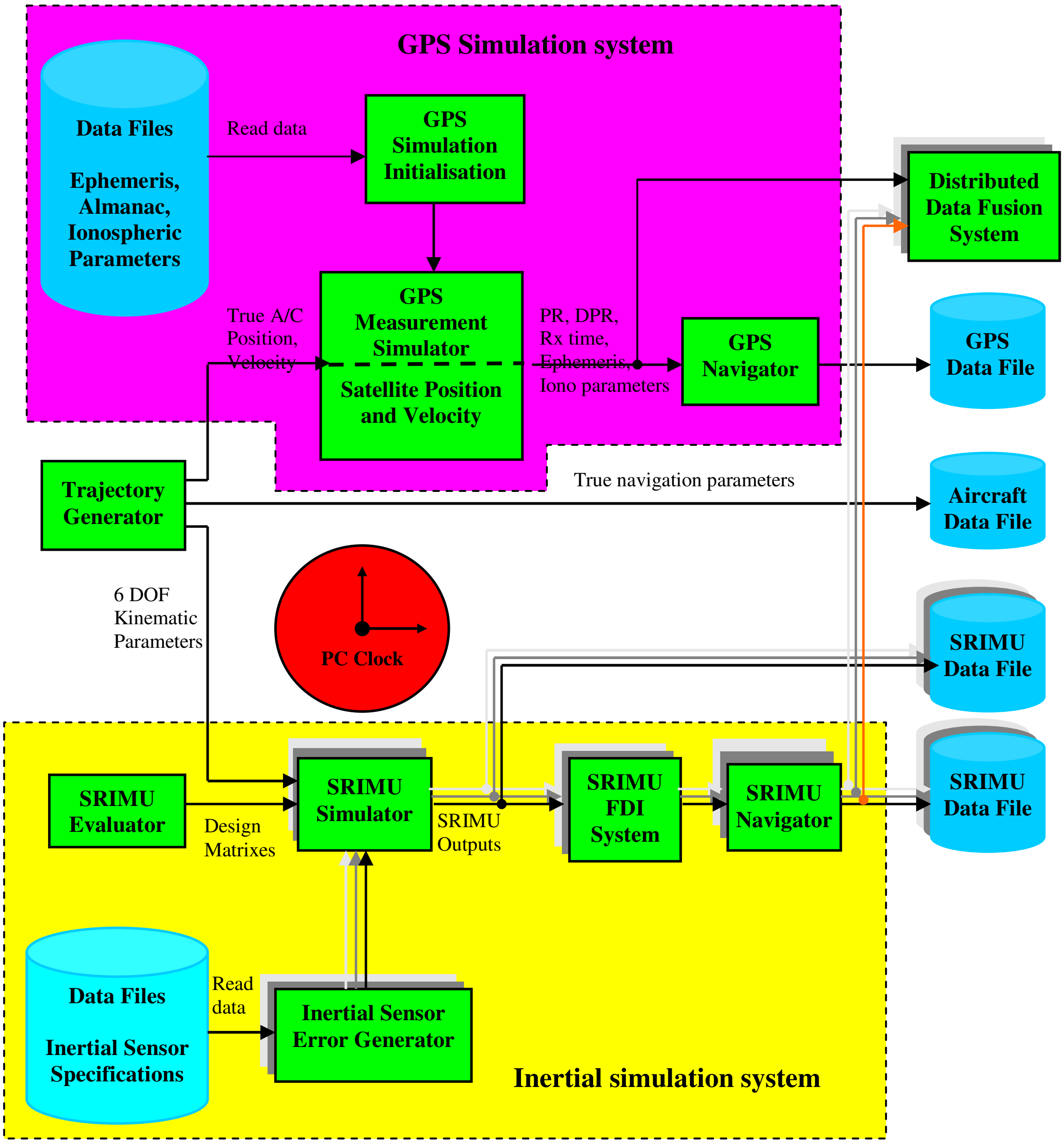

GPS Simulation system
Data Files
Ephemeris, Almanac, Ionospheric Parameters
Read data
GPS Simulation Initialisation
True A/C Position, Velocity
GPS Measurement Simulator
Satellite Position and Velocity
PR, DPR, Rx time, Ephemeris, Iono parameters
GPS Navigator
Distributed Data Fusion System
GPS Data File
Trajectory Generator
True navigation parameters
Aircraft Data File
6 DOF Kinematic Parameters
PC Clock
SRIMU Data File
SRIMU Data File
SRIMU Evaluator
Design Matrixes
SRIMU Simulator
SRIMU Outputs
SRIMU FDI System
SRIMU Navigator
Data Files
Inertial Sensor Specifications
Read data
Inertial Sensor Error Generator
Inertial simulation system

Figure 7.1 Simulation System Architecture

SRIMU Navigator

This module implements the inertial navigation computation and the dynamic SRIMU calibrations. At each inertial network node, this module completes similar computations but provides the local navigation states. The inertial navigation and

SRIMU calibration algorithms have been described in Chapter 4. The SRIMU FDI algorithms have been introduced in Chapter 5.

GPS Simulation System

The GPS simulation system consists of three sub-modules: GPS Measurement Simulator, GPS Navigator and GPS-based attitude simulator. The GPS Measurement Simulator generates the realistic GPS pseudorange and Doppler measurements. The dimension of the simulated GPS measurement vector depends on the true position of aircraft given by the Trajectory Generator. The GPS measurement errors come from several errors sources, including ionospheric and tropospheric delay errors, GPS satellite and receiver clock errors, selective availability (SA) errors (the SA has been turned off), satellite ephemeris errors and receiver measurement noise. The GPS navigator implements the GPS navigation computations.

If a multifunctional GPS receiving system is available in the inertial network system, it can provide the GPS-based attitude information. The GPS-based attitude simulator (not shown in Figure 7.1) simulates the aircraft cg-referenced attitude solutions derived by the GPS-based attitude determination algorithm. Although the GPS-based attitude determination algorithms have been developed by the author, this simulator does not simulate the GPS carrier phase measurements or perform the GPS-based attitude computations.

The multifunctional GPS receiver is a commonly-shared navaid sensor in the inertial network system.

Other navaid sensor systems simulated in this thesis include an air data system, a magnetic heading sensor and a Doppler radar. These sensor systems are considered as the local sensors located at the cg node and are omitted from in Figure 7.1.

7.2.1 Inertial Simulation System Architecture

The inertial simulation system performs two main functions: generation of SRIMU measurements and computation of the navigation states. These two functions are performed by the SRIMU simulator and the SRIMU navigator.

The architecture of the SRIMU simulator is illustrated in Figure 7.2 where the modules framed by the dash-lines on the right and left sides are the inputs and the outputs to other modules. The SRIMU simulator also produces the true navigation state in addition to the generation of the realistic SRIMU measurements. The true state is used to test and evaluate the performance of the distributed data fusion algorithms. The test results can further determine what grades of inertial sensors should be used in the inertial network system to achieve the required navigation performance. The inertial sensor error sources and simulation parameters are given in Table J-1 of Appendix J.

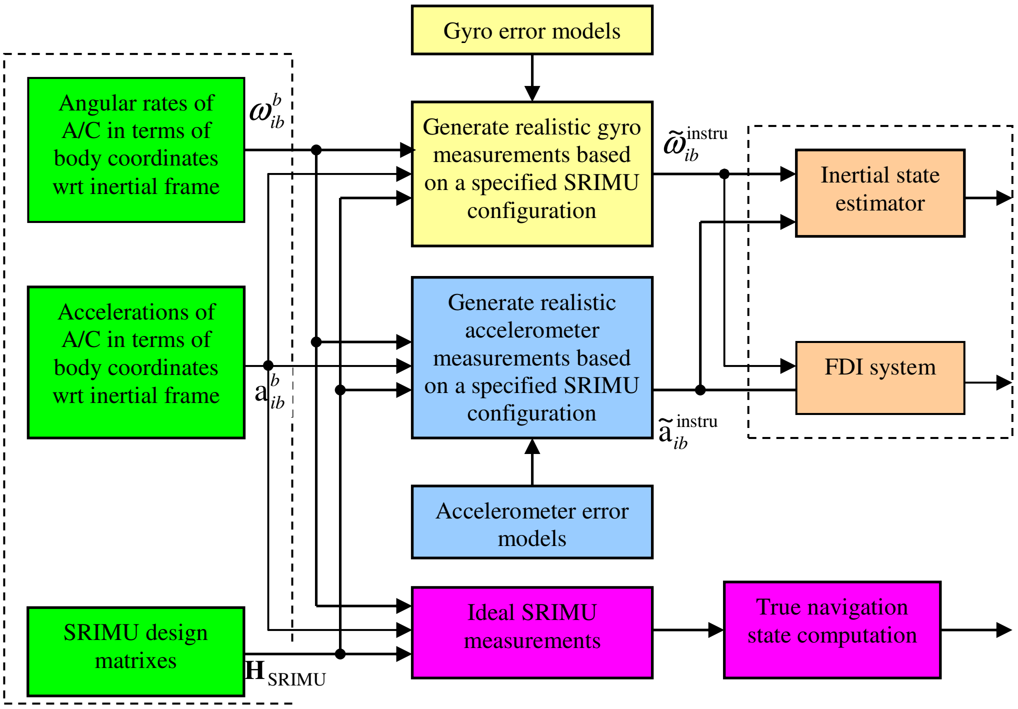

Gyro error models
Angular rates of A/C in terms of body coordinates wrt inertial frame
ω_{ib}^{b}
Generate realistic gyro measurements based on a specified SRIMU configuration
$\tilde{\omega}_{ib}^{instru}$
Inertial state estimator
Accelerations of A/C in terms of body coordinates wrt inertial frame
a_{ib}^{b}
Generate realistic accelerometer measurements based on a specified SRIMU configuration
$\tilde{a}_{ib}^{instru}$
FDI system
Accelerometer error models
SRIMU design matrixes
H_{SRIMU}
Ideal SRIMU measurements
True navigation state computation

Figure 7.2 SRIMU Simulator Architecture

The architecture of the SRIMU navigator located at the cg node is illustrated in Figure 7.3 where the modules framed by the dash-lines are the conventional coarse and fine alignment functions, and the vertical channel compensation of the inertial navigator, as described in Chapter 4. However, the other SRIMU navigators located at the other inertial network nodes do not have these three modules. The dynamic and transfer alignments of the SRIMUs in the inertial network system are completed by the distributed Kalman and fusion filters, as described in Chapter 6.

From Figure 7.3, each SRIMU navigator consists of an inertial state estimator and an inertial navigator. The inertial state estimator computes the measured local inertial state by using local SRIMU measurements and assimilating the other SRIMU measurements. The redundancy management reconfigures the design matrix based on the inertial sensor failure report from the FDI system. The accelerometer and gyro compensation modules dynamically correct the local SRIMU measurements by using the inertial sensor error estimates obtained from the local Kalman filter, as described in Chapter 6.

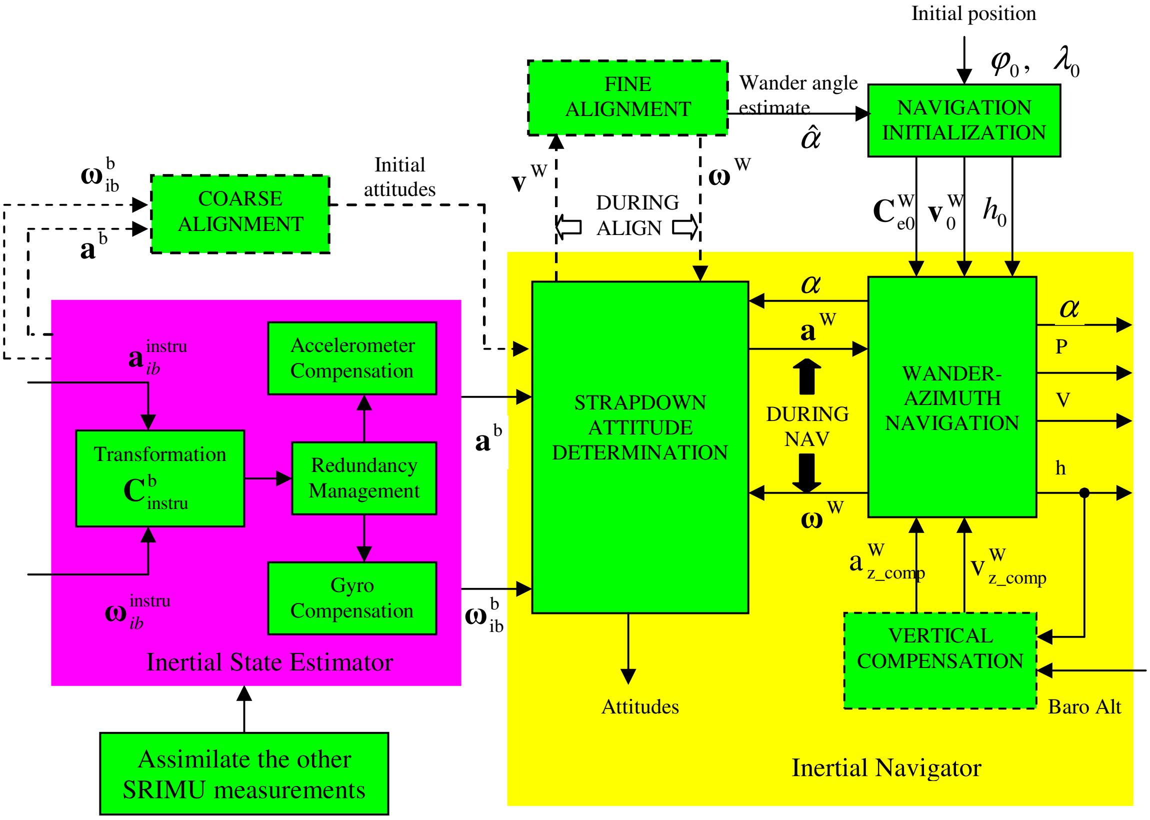


Figure 7.3 SRIMU Navigator Architecture

The inertial navigator computes the local navigation state. The local state is also used to establish the local Kalman filter model. The inertial navigation algorithm has been introduced in Chapter 4, the SRIMU compensation algorithms have been developed in Chapter 5 and the inertial state estimator algorithm has been described in Chapters 5 and 6.

7.2.2 GPS Simulation System Architecture

The architecture of the GPS simulation system is shown in Figure 7.4 where the GPS navigator outputs the GPS-derived aircraft position and velocity. The GPS navigator outputs are used to test and evaluate the distributed Kalman and fusion filters and to compare different GPS/INS integration mechanisms.

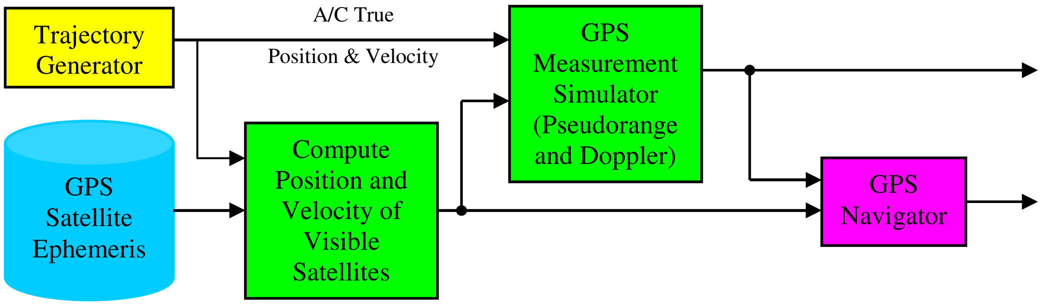

Trajectory Generator
A/C True
Position & Velocity
GPS Measurement Simulator (Pseudorange and Doppler)
GPS Satellite Ephemeris
Compute Position and Velocity of Visible Satellites
GPS Navigator

Figure 7.4 GPS Simulator Architecture

The GPS Measurement Simulator provides realistic GPS measurements, including pseudorange and Doppler (range rate) measurements, rather than deriving the actual GPS signals provided by an engineering GPS simulator. The simulated pseudoranges are obtained by a combination of the true geometric distances from the aircraft to visible satellites and the error terms of various error sources. The true geometric distances are computed from the true position of the aircraft and the positions of the visible GPS satellites. The visibility of GPS satellites is determined by the true position of the aircraft and the minimum elevation of the GPS satellites relative to the aircraft local level frame. In this simulation study, the minimum elevation for signal acquisition is defined as five degrees.

The simulated Doppler measurements are the aircraft to GPS satellites LOS range rates, which are the LOS projects of the velocity differences between the true velocity of the aircraft and the velocity of the visible satellites plus the range rate errors caused by various error sources.

The architecture of the GPS measurement simulation algorithm is illustrated in Figure J.1 of Appendix J where the GPS measurement error models were taken from The Johns Hopkins University Applied Physics Laboratory[3]. Main error sources and simulation parameters are summarised in Table J-2 of Appendix J[3][83]. The ADS and the magnetic heading sensor are simulated according to the parameters shown in Table J-3 of Appendix J.

7.2.3 Multisensor Fusion Simulation at Node cg

The cg node local system is a vital subsystem in an inertial network system and provides the aircraft systems with the navigation states. The purpose of this simulation study is to evaluate the FDI and data fusion algorithms developed in this thesis. The Kalman filter located at the cg node has multiple operating modes. The architecture of the multi-mode filter algorithm is illustrated in Figure 7.6 where the inputs to the Inertial Navigation module include the SRIMU measurements, pressure altitude and the positions of visible GPS satellites. This module outputs the coarse estimates of the navigation states and the estimated GPS measurements.

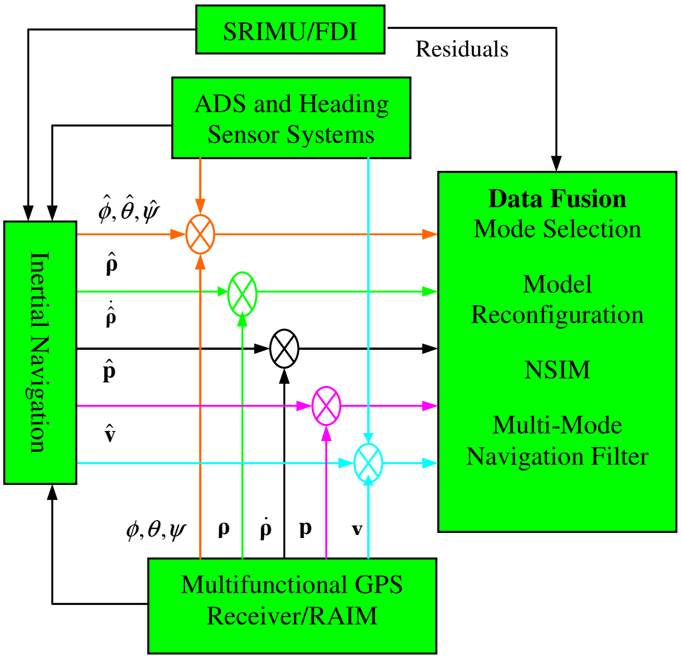


Figure 7.5 Multi-Mode Kalman Filter Architecture at the cg Node

The Multifunctional GPS module outputs the GPS-based attitude information and navigation states, and the GPS measurements. The SRIMU/FDI module provides the FDI test results to reconfigure the inertial system and the Kalman filter; it also outputs the raw SRIMU measurements for the inertial navigation computation and the SRIMU measurement residuals as the measurements of the multi-mode Kalman filter. The ADS and Heading Sensor System module provides pressure altitude compensation needed for the stabilisation of the vertical channel of the inertial system, and the heading and air speed measurements of the multi-mode Kalman filter. The Data Fusion module completes the following tasks:

- Processes the raw sensor measurements to generate the normalised data for the multi-mode navigation filter,
- Determines the operating mode according to the sensor health status reports from the SRIMU/FDI and GPS/RAIM modules,
- Reconfigures the dynamic model and measurement model of the cg node local system,
- Performs the local Kalman filter and the fusion filter algorithms to update the coarse estimates of the aircraft navigation states,
- Corrects the cg node SRIMU errors in flight,
- Monitors the abnormality of estimates of the local navigation states.

The filter state vector is subdivided into two sub-vectors, known as the basic state error vector and the sensor error vector as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\text{basic state errors}} \\ \mathbf{x}_{\text{sensor errors}} \end{bmatrix}$$

$$\mathbf{x}_{\text{basic state errors}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \delta\varphi_R \\ \delta\lambda_R \\ \delta h \\ \delta v_x^1 \\ \delta v_y^1 \\ \delta v_z^1 \\ \partial\phi_x \\ \partial\phi_y \\ \partial\phi_z \end{bmatrix} \quad \mathbf{x}_{\text{sensor errors}} = \begin{bmatrix} \text{Rx Clk Error States} \\ \text{SRIMU Accel Error States} \\ \text{SRIMU Gyro Error States} \\ \text{Magnetic Heading Error State} \\ \text{Barometer Bias State} \\ \text{True AirSpeed Bias State} \end{bmatrix} = \begin{bmatrix} x_{\text{clk-pha}} \\ x_{\text{clk-rate}} \\ \mathbf{x}_{\text{accel-1}} \\ \vdots \\ \mathbf{x}_{\text{accel-}n} \\ \mathbf{x}_{\text{gyro-1}} \\ \vdots \\ \mathbf{x}_{\text{gyro-}n} \\ \mathbf{x}_{\text{Mag}} \\ \mathbf{x}_{\text{Bar}} \\ \mathbf{x}_{\text{ADS}} \end{bmatrix}$$

The basic state error vector has a fixed dimension but the size of the sensor error vector changes with the numbers of SRIMU sensors and the number of aiding sensors available.

The filter measurement vector is formulated as follows and its size depends on available sensor systems.

$$\mathbf{z}=\begin{bmatrix}\text{GPS pseudorange vector}\\ \text{GPS pseudorange rate vector}\\ \text{GPS - based attitude vector}\\ \text{Magnetic heading output}\\ \text{Ture airspeed}\\ \text{SRIMU residual vector}\end{bmatrix}=\begin{bmatrix}\Delta\text{PR}\\ \Delta\text{PRR}\\ \Delta\text{Att}\\ \Delta\Psi_{\text{M}}\\ \Delta\text{V}_{\text{TAS}}\\ \text{Residuals}\end{bmatrix}$$

From the error analysis of GNSS and inertial systems given in Chapter 4, the system noise covariance $\mathbf{Q}$ and measurement noise covariance $\mathbf{R}$ may change with each time or measurement update, however here they are assumed to be constant in order to avoid the real-time computation of these two matrices. The determination of $\mathbf{R}$ is based on a prior statistical characteristics of the sensor measurement noises, which can be obtained by practically testing sensor systems. Rational selection of $\mathbf{R}$ should guarantee the robustness of the Kalman filter to the change of measurement noise. The standard deviations of the measurement noises of several navaid systems are summarised in Tables J-2 and J-3 in Appendix J, which are used to determine $\mathbf{R}$ in this simulation study.

The determination of the system noise covariance $\mathbf{Q}$ is generally more difficult as it cannot be determined by practically testing or directly observing the dynamic system. However, the uncertainties of the initial filter states, including the basic state error and sensor error, can be used as reference values for the selection of $\mathbf{Q}$. Tables J-1 and J-4 in Appendix J give the initial uncertainties of the basic state errors and sensor errors.

The dynamic reconfiguration of the cg node filter includes the filter state vector reconfiguration and the measurement vector reconfiguration. The cg node system compensation consists of the navigation state compensation and the SRIMU error compensation, as shown in Figure J.2 of Appendix.

The operating modes of the multi-mode Kalman filter and the sensor systems used the cg node are listed in Table 7-1. The initial values of the navigation states are given in Table J-4 of Appendix J.

Table 7-1 Kalman Filter Operating Modes

IMU	ADS	Heading Sensor	GNSS Range	GNSS Attitude	KF Mode
X	X	X	X	X	All measurement available
X	X		X	X	Lost heading sensor
X		X	X	X	Lost ADS
X			X	X	Lost ADS & heading sensor
X	X	X	X		Lost GPS attitude function
X	X	X			No GPS signals

7.2.4 Software Development and Evaluation

This software simulation system is implemented by use of Matlab. Software development is based on the top-down design and bottom-up realisation strategies. In the top-down software design, all the software modules described in the previous sections are decomposed into several sub-modules according to the functions to be completed. In each sub-module, the main function is further decomposed into many sub-functions, each realising one or more relatively independent tasks. Accordingly, the top-down design generates software function trees (see in Figure 7.6). In contrast, the programming is based on the bottom-up strategy. The lowest-level sub-functions are first programmed, tested and integrated into the higher-level sub-functions. Higher-level sub-functions are then integrated and tested to form a sub-module. Data transfer between modules and functions uses actual parameters. Furthermore, all the navigation states and kinematic parameters are defined as the global variables which can be directly invoked in all modules and functions.

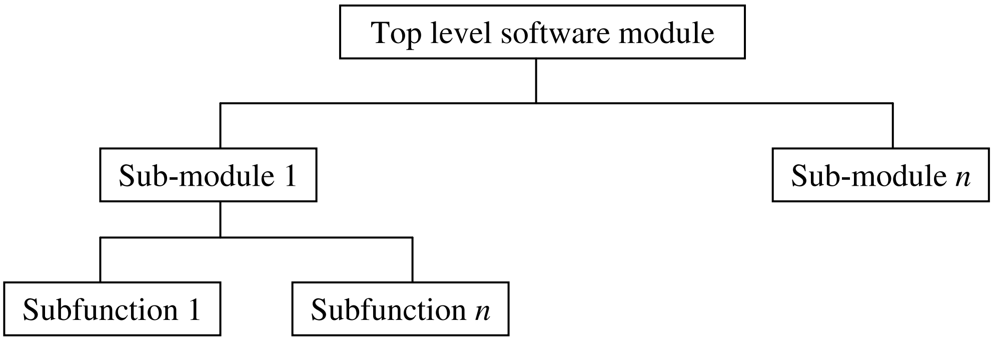


Figure 7.6 Top-Down Software System Design

The evaluation of the software system consists of the static and dynamic testing procedures. The static evaluation is also based on the bottom-up method. The lowest-level subroutines are first tested and then sub-modules are tested. The bottom-up testing procedures verify if the subroutines or sub-modules can produce the expected outputs using pre-known static data as their inputs. For example, in order to test the SRIMU attitude determination module, an IMU is assumed to be fixed on a static platform with known orientation. The SRIMU then measures the components of the Earth’s rotation rate and gravity vectors. The use of these components as the input data of the attitude module can examine the correctness of the attitude computation. All the sub-modules and functions were tested using similar methods before they were integrated into the software simulation system.

The dynamic evaluation is a system test procedure and is generally based on the predictability of the behaviour of the software simulation systems. For example, an inertial system shows the Schuler period of 84.4890 minutes. This feature can be used to test the correctness of the inertial simulation subsystem. If the Schuler period of the navigation states given by the inertial simulation software system differs from the theoretical Schuler value, then the sub-module and functions in the inertial software system must be re-evaluated using the static testing procedures. The GNSS positioning solutions normally have the maximum error thresholds if no GNSS signal failures occur. In order to evaluate the GNSS software simulation subsystem, the maximum error thresholds for the GNSS-derived position and velocity are assumed to be 200 m and 1 m/s. If the outputs of the GNSS software simulation subsystem exceed the specified thresholds, then all the sub-modules and functions in the GNSS software system will be re-evaluated by using the static testing procedures.

The evaluation of the distributed data fusion filter software systems is a more complex procedure. Although the static testing procedures can assure the correctness of subroutines, the dynamic testing procedure cannot completely check the suitability of the fusion filter software systems because there are many uncertainties of sensor measurements. Therefore, the dynamic evaluation is mainly to examine the abnormal behaviours of the filter software systems from the perspective of the filter states. In normal operation, the filter states will change smoothly over time if the sensor signal failures have not been injected into its measurements. The designed true trajectory is used as a reference to test the correctness of the software simulation system.

7.3 MW-GLRT Algorithm Test Results

This simulation study considers two cases for the evaluation of the MW-GLRT algorithms developed in Chapter 5, that is, SRIMU configurations consisting of four sensors and five sensors. The test procedure is illustrated in Figure 7.7.

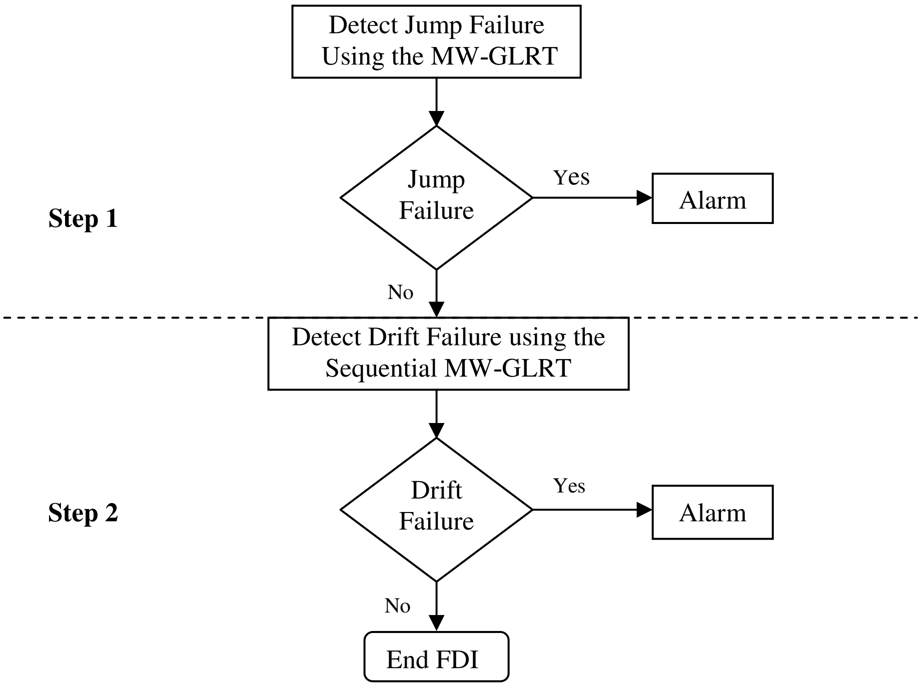

Detect Jump Failure Using the MW-GLRT
Step 1
Jump Failure
Yes
Alarm
No
Detect Drift Failure using the Sequential MW-GLRT
Step 2
Drift Failure
Yes
Alarm
No
End FDI

Figure 7.7 Two-Step FDI Test Procedure

In the 4-sensor configurations, the MW-GLRT algorithms are tested by detecting mid-value and drift failures. In the 5-sensor configurations, the MW-GLRT algorithms are evaluated by detecting, identifying and estimating large jump failures. The parameters for the design of the MW-GLRT algorithms are listed in Table J-6 of Appendix J.

The simulation test results of the SRIMU error compensation filters are shown in Figures 7.8 and 7.9 where the SRIMU is based on a 4-sensor cube configuration. These simulation results show that the use of the SRIMU error compensation filters can slow down the degradation of the accuracy of the free SRIMU navigation system. As a result, this error compensation filter can be used to compensate for the absence of the navaid systems, for example, for interruptions and unavailability of GPS signals.

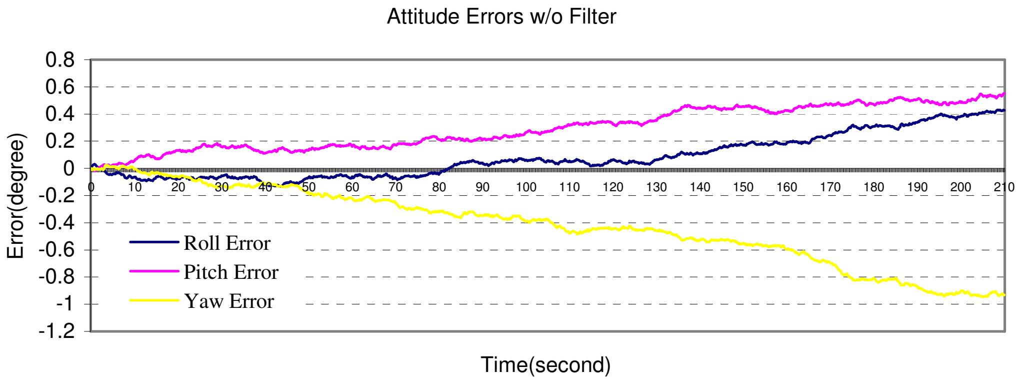


Figure 7.8 Attitude Errors without the SRIMU Filter

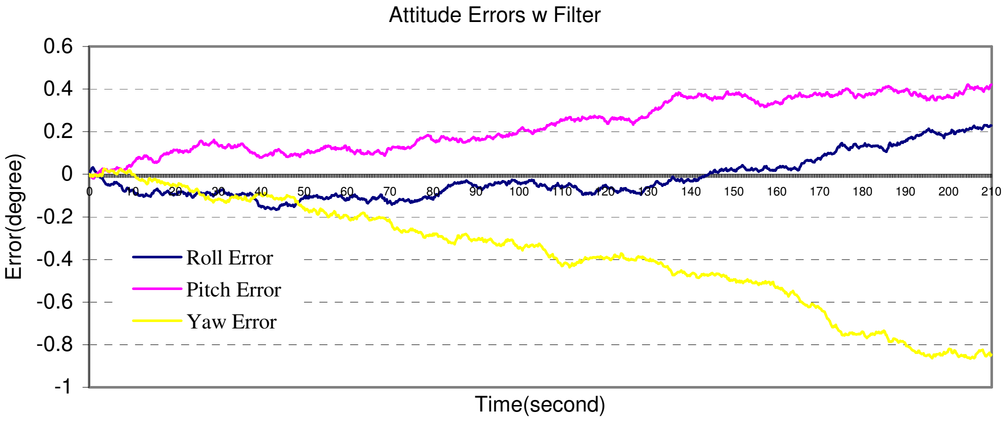


Figure 7.9 Attitude Errors with the SRIMU Filter

In the following simulations, it will further be shown that combination of the local node fusion filters and the SRIMU error compensation filters in the inertial network system can greatly increase the integrity and accuracy of the navigation state estimates.

The simulation test results of the MW-GLRT algorithms are shown in Figures 7.10 to 7.14 for the 4-sensor-cube configuration where sensor drift failures and a mid-value failure occur. The results given in Figure 7.10 show the attitude errors caused by the effects of a mid-value gyro failure on the inertial attitude determination function. Although the gyro mid-value failure may not be detected by the short time MW-GLRT method, as shown in Figure 7.11, it can be detected by the sequential MW-GLRT method, as illustrated in Figure 7.12.

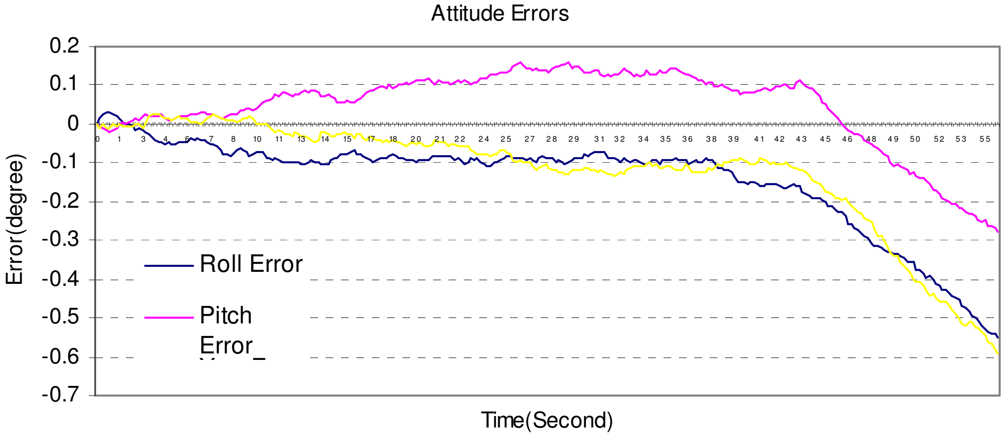


Figure 7.10 Attitude Errors in Present of a Mid-Value Gyro Failure

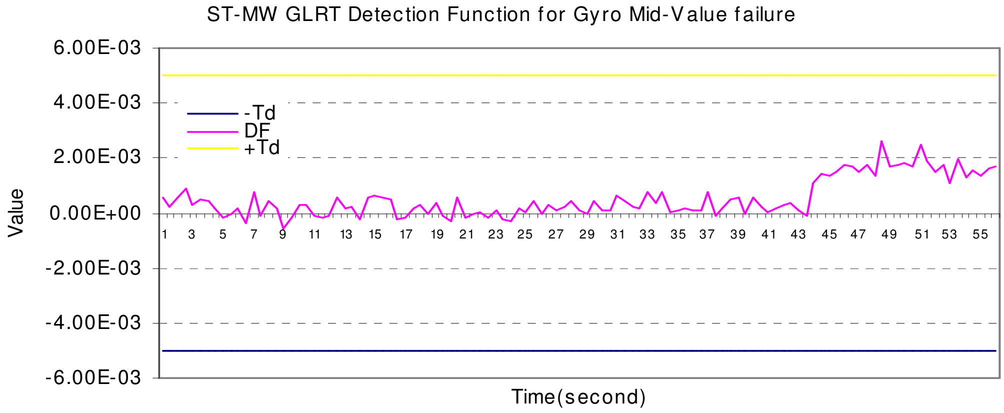


Figure 7.11 Gyro Detection Function Using Short Time MW-GLRT in 4-Sensor Cube[6]

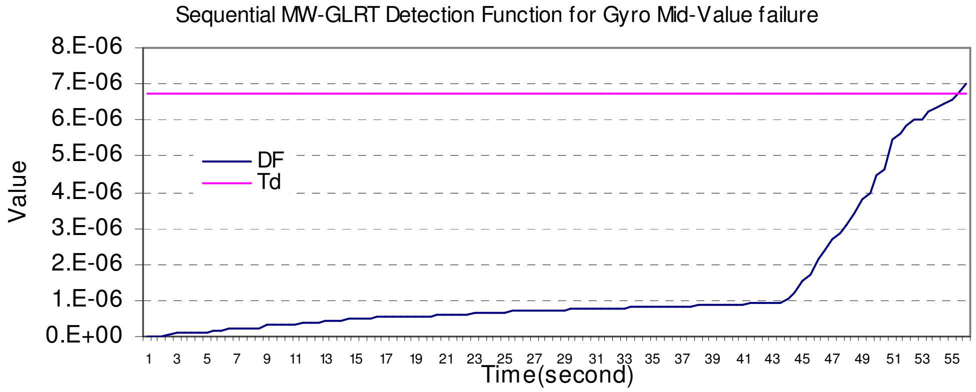


Figure 7.12 Gyro Detection Function Using Sequential MW-GLRT in 4-Sensor Cube

[6] In all figures, DF represents Detection Function and Td means detection Threshold.

The simulation test results of SRIMU drift failures are shown in Figures 7.13 and 7.14 where the drift failure is undetectable using the MW-GLRT algorithm, but can be detected by the sequential MW-GLRT algorithm.

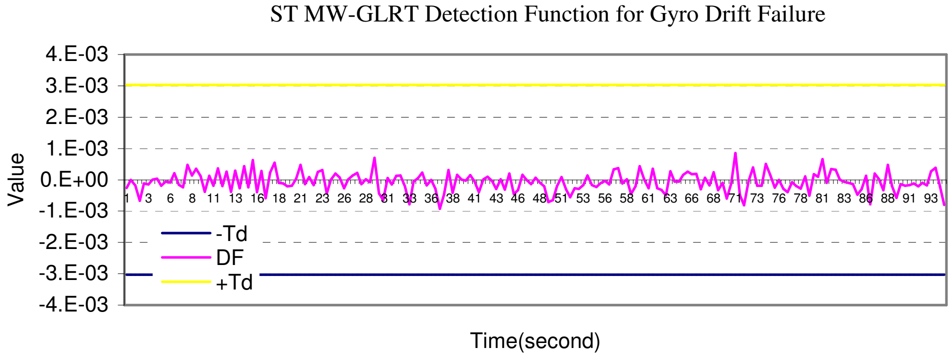


Figure 7.13 Gyro Drift Detection Function Using Short-Time MW-GLRT in 4-Sensor Cube

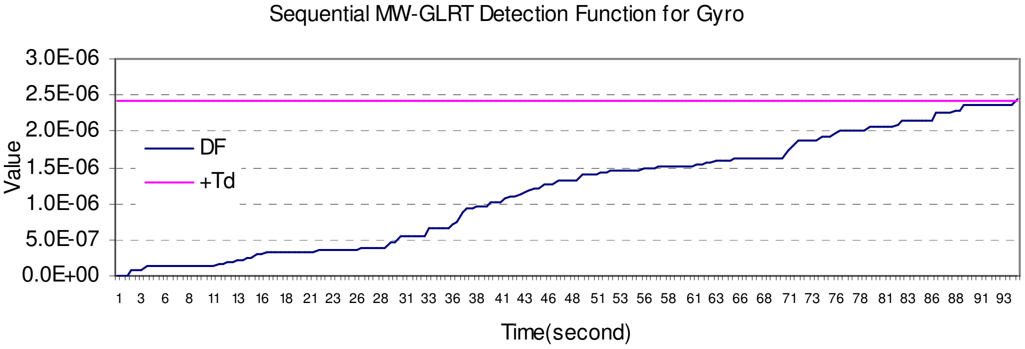


Figure 7.14 Gyro Drift Detection Function Using Sequential MW-GLRT in 4-Sensor Cube

For the 5-sensor cone configuration, the simulation test results are shown in Figures 7.15 to 7.23 where the SRIMU system may experience jump failures, mid-value failures and drift failures.

Figures 7.15 to 7.17 show the case where one gyro has a mid-value failure and one accelerometer has a jump-step failure. The accelerometer jump failure is detected and identified by the MW-GLRT algorithm. Furthermore, the accelerometer failure signal is estimated and compensated in the SRIMU measurements, as shown in Figure 7.16, where the failure signal of the fault accelerometer is 0.02g and its estimate is 0.1928 m/s^2.

The mid-value gyro failure is detected by the sequential MW-GLRT, as shown in Figure 7.17 in comparison with Figure 7.15.

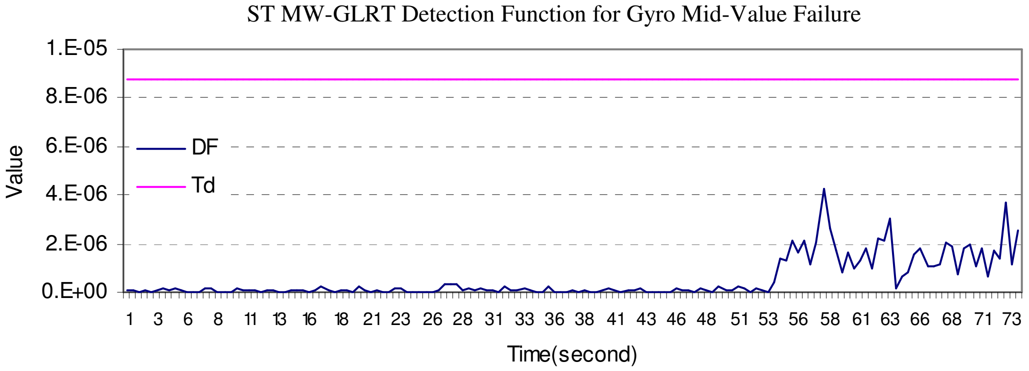


Figure 7.15 Gyro Detection Function Using Short-Time MW-GLRT in 5-Sensor Cone

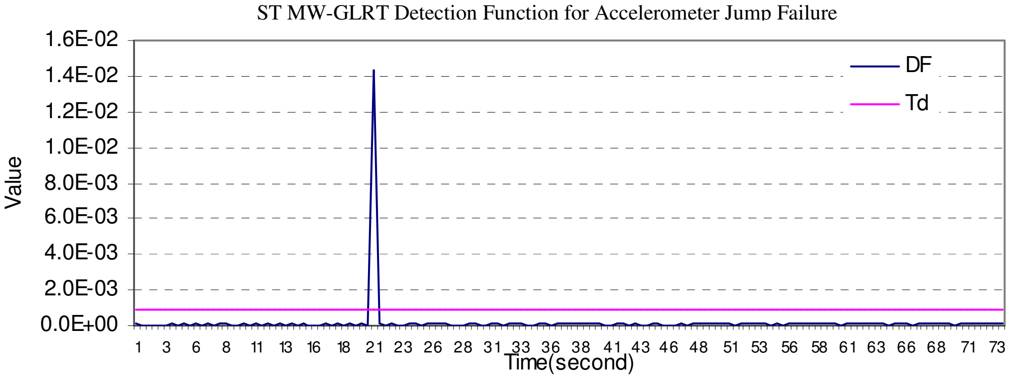


Figure 7.16 Accelerometer Detection Function Using Short-Time MW-GLRT in 5-Sensor Cone

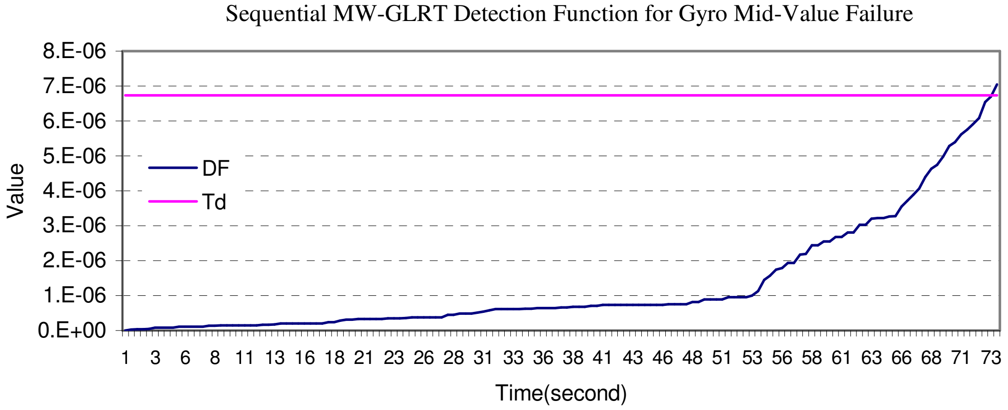


Figure 7.17 Gyro Detection Function Using Sequential MW-GLRT in 5-Sensor Cone

Figures 7.18 to 7.20 show the case where two accelerometers (sensors 1 and 4 shown in Figure 5.4(c)) have jump failures sequentially and one gyro has a mid-value failure.

From Figure 7.18, the mid-value gyro failure cannot be detected by the short-time MW-GLRT algorithm, but can be detected by the sequential MW-GLRT algorithm, as shown in Figure 7.19.

In Figure 7.20, the failure signal of accelerometer 1 is 0.006g and its estimate is 0.0711 m/s^2. The failure signal of accelerometer 4 is 0.02g and the corresponding estimate is 0.1928 m/s^2.

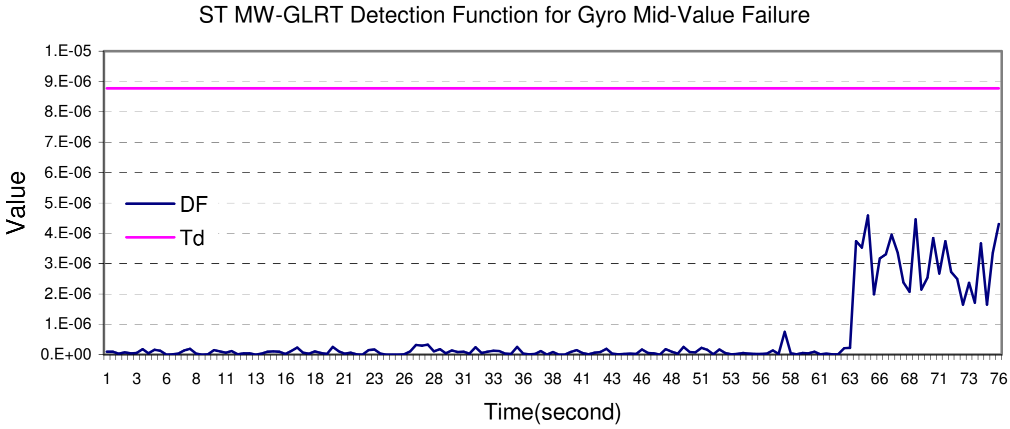


Figure 7.18 Gyro Detection Function Using Short-Time MW-GLRT in 5-Sensor Cone

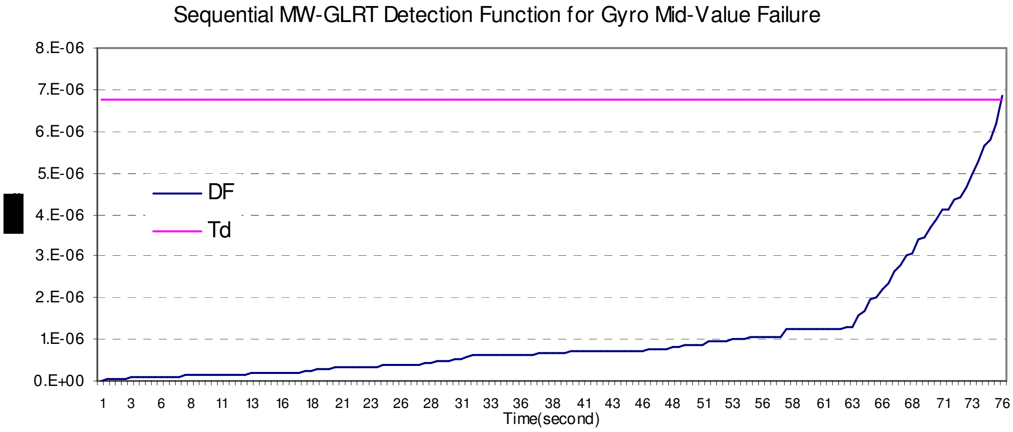


Figure 7.19 Gyro Detection Function Using Sequential MW-GLRT in 5-Sensor Cone

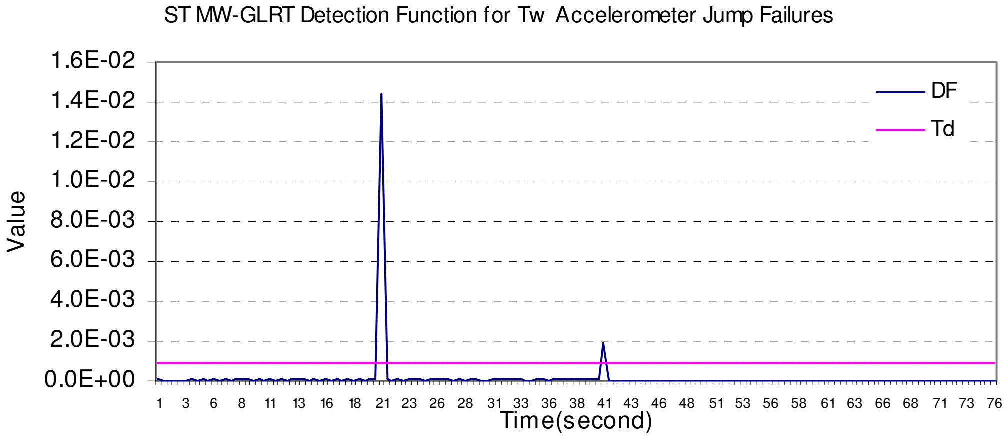


Figure 7.20 Accelerometer Detection Function Using Short-Time MW-GLRT

Figures 7.21 to 7.23 show the test results where both two accelerometers and two gyroscopes have jump failures. In Figure 7.21, gyro 2 has a failure signal of 0.04 rad/s and the corresponding estimate is 0.040 rad/s.

Gyro 4 has a failure signal of 0.002 rad/s and the estimate is 0.006 rad/s. Clearly, the estimate of the failure signal for gyro 4 is far from the true value. However, the error caused by the estimation can be detected by the sequential MW-GLRT, as shown in Figure 7.22.

In Figure 7.23, the failure signal of accelerometer 1 is 0.006g and its estimate is 0.0711 m/s^2. Accelerometer 4 has a failure signal of 0.02g and its estimate is 0.1928 m/s^2.

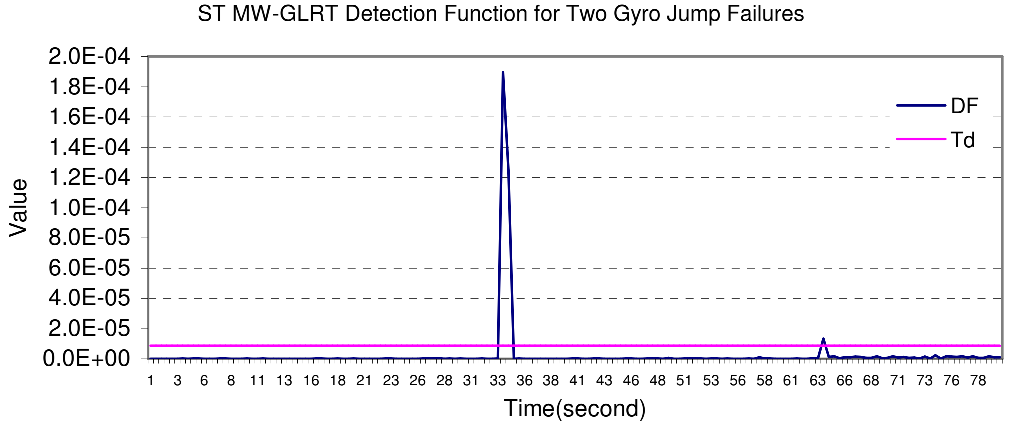


Figure 7.21 Gyro Detection Function Using Short-Time MW-GLRT in 5-Sensor Cone

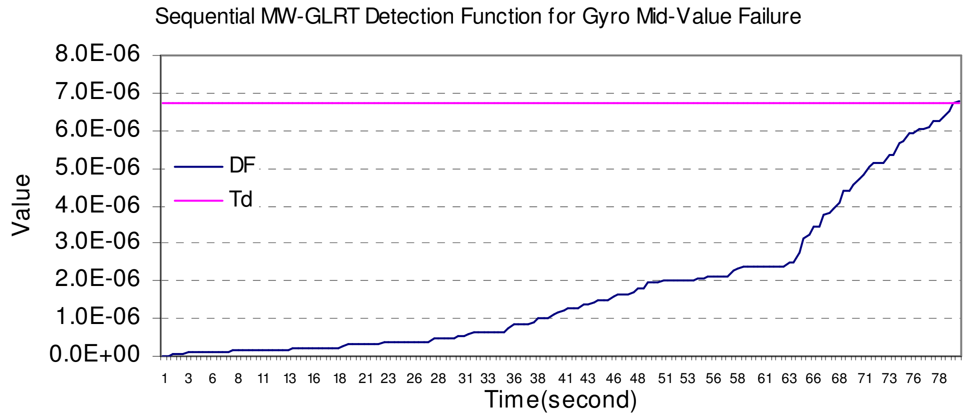


Figure 7.22 Gyro Detection Function Using Sequential MW-GLRT in 5-Sensor Cone

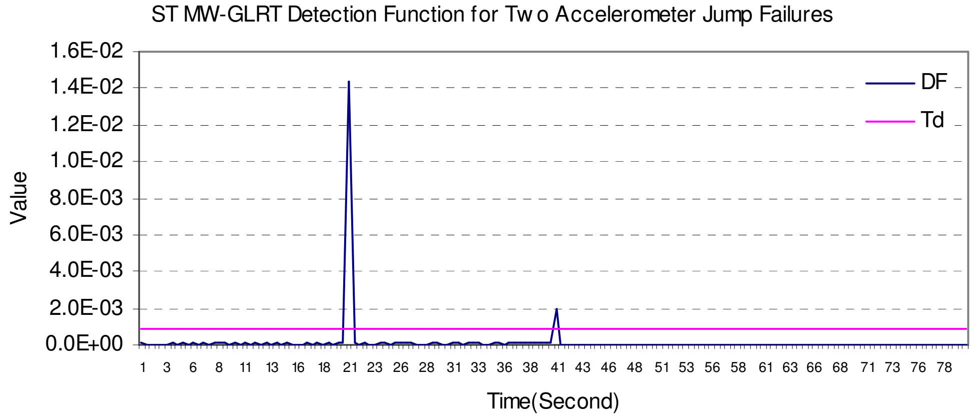


Figure 7.23 Accelerometer Detection Function Using Short-Time MW-GLRT in 5-Sensor Cone

Further simulations were performed for different SRIMU configurations consisting of 4 and 5 sensors. The simulation research results are summarised below:

- Optimal SRIMU configurations are determined on the basis of a trade-off of the minimum GDOP factor, allocation of normalised variances of SRIMU measurement errors in orthogonal body axes and SRIMU FDI capability. Coplanar sensor installation should be avoided in order to obtain maximum FDI capability in terms of sensor configuration.
- Two 4-sensor and four 5-sensor SRIMU configurations were also simulated in the occurrence of one sensor failure and their estimate accuracy and degradation performance were compared. The simulation results show that cone configurations in 4-sensor or 5-sensor SRIMUs can provide a better

estimate of the measured state and minimum degradation of performance for various SRIMU configurations for one sensor failure.

- The simulation results have shown that SRIMU error compensation filters can improve the accuracy of an SRIMU system and the performance of FDI algorithms developed in this thesis. The degradation of an SRIMU system is reduced so that the navaid systems (for example, GNSS) have sufficient time and access to satellite signals to correct the SRIMU systems.
- The short-time MW-GLRT algorithms can remove abnormal measurement noise and detect (and estimate) sensor hard failures. In the simulation, the detection threshold values were determined on the basis of the probability of a false alarm of 10^{-6} and the probability of a missed alarm of 10^{-5}.
- The sequential MW-GLRT algorithms can efficiently detect mid-value and drift failures, which degrade the accuracy of an SRIMU system without alarm.

7.4 Distributed Data Fusion Filter Test Results

The aim of this simulation study is to test and evaluate the node filters and the node state integrity monitoring algorithms that have been developed in Chapters 4 and 6. In this simulation, the GNSS simulation module is assumed to provide the raw measurements and navigation state information at the rate of 1Hz, the inertial navigation module outputs attitude information at the rate of 50 Hz and the position and velocity at the rate of 1Hz and other navaid systems output at the rate of 1Hz. Several cases in Table 7-2 were simulated and the results are given in the following subsections.

Table 7-2 Simulation Cases

Simulation Case		SRIMU (Gyro Drift Rate)	GNSS Information (PR, PRR & Attitudes)
Case 1	1A	1 ^{0}/h	No interruption
	1B	1 ^{0}/h	GNSS attitude available for maximum 6 minutes
Case 2	2A	10 ^{0}/h	No interruption
	2B	10 ^{0}/h	GNSS attitude interruption for maximum 6 minutes
Case 3		40 ^{0}/h	Slave node simulation
Case 4		gyro failures	GPS signal failures

The aircraft can perform arbitrary manoeuvres with a maximum acceleration of 3.0g. In order to compare the simulation results, however, an identical true trajectory is designed for all the simulation cases. A typical true flight trajectory is shown from Figures 7.24 to 7.26. The true flight path is depicted in Figure 7.24 where the arrow represents the flight direction. Figures 7.25 is the true horizontal manoeuvres and Figure 7.26 is the true vertical manoeuvres. To evaluate the performance of the data fusion algorithms developed in this PhD study, the true trajectory is used as reference values to compare with the estimated states of the aircraft motion. Therefore, all the state errors are the differences between the true and estimated aircraft motion states.

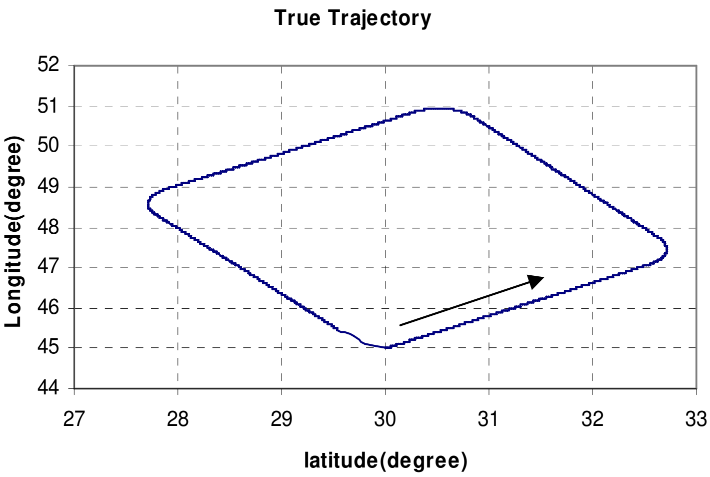


Figure 7.24 True Flight Trajectory

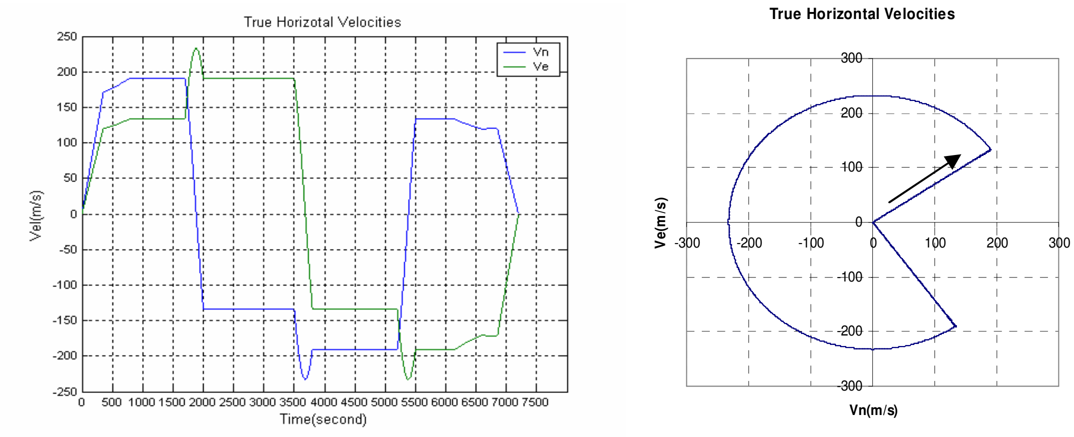


Figure 7.25 True Aircraft Horizontal Manoeuvres

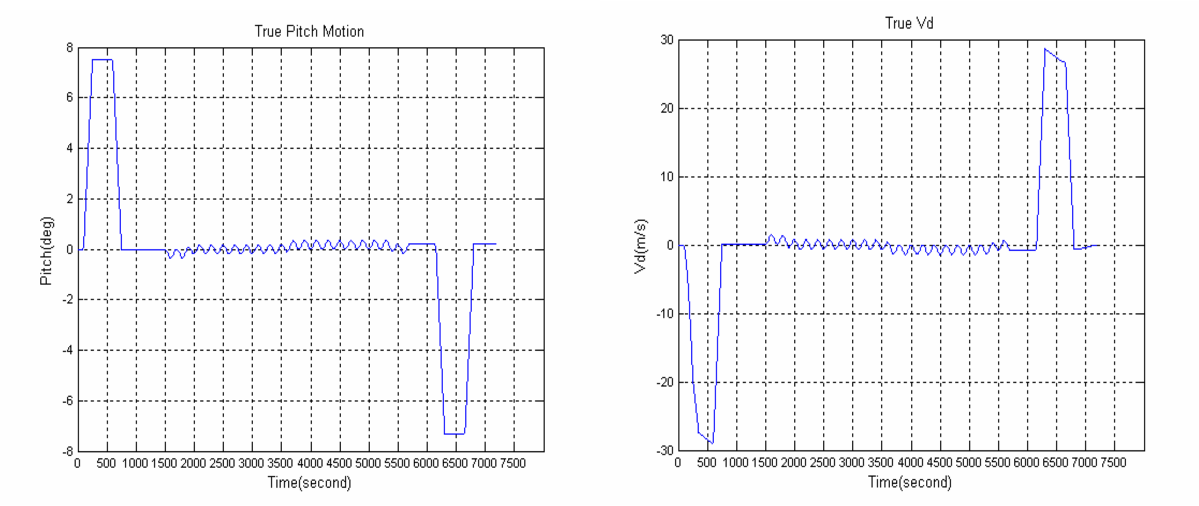


Figure 7.26 True Aircraft Vertical Manoeuvres

7.4.1 Simulation Results at cg Node

In this simulation, the GNSS information available is summarised as follows:

- GNSS pseudorange (PR) and pseudorange rate (PRR) measurements,
- GNSS-based attitude information.

For a gyro bias of 1^0/h, the simulation test results at the cg node are shown in Figures 7.27 to 7.35 where there is no GNSS signal interruption. In the following figures, the absolute errors, which are the differences between the true and estimated states, are used to describe the realistic state estimate errors whereas the standard deviations of the state estimate errors represent the accuracy of the state estimates.

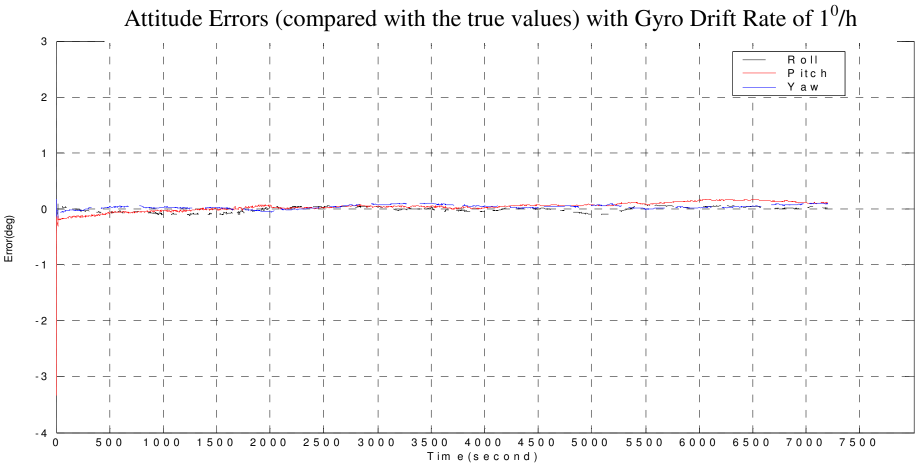


Figure 7.27 The cg Node Attitude Errors in Case 1A

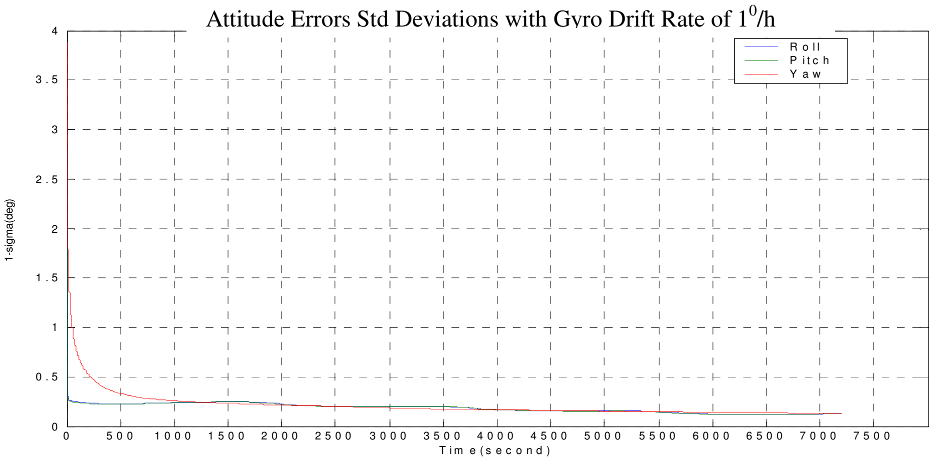


Figure 7.28 The cg Node Attitude Error Standard Deviations in Case 1A

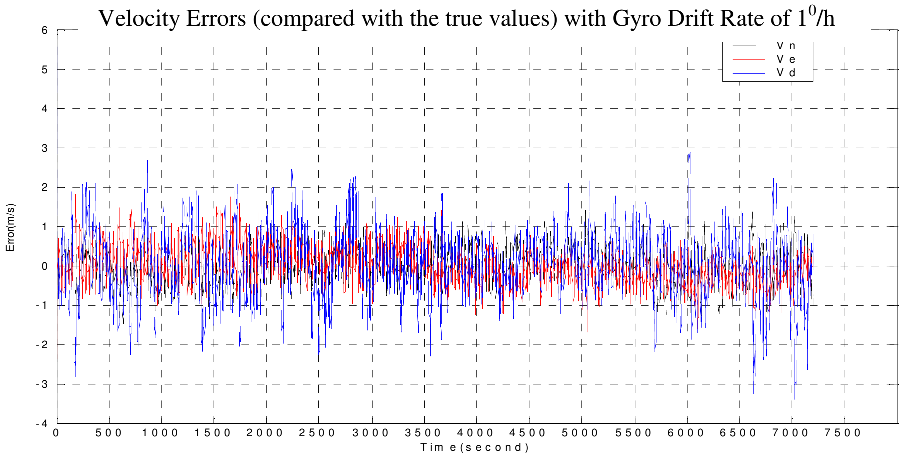


Figure 7.29 the cg Node Velocity Errors in Case 1A

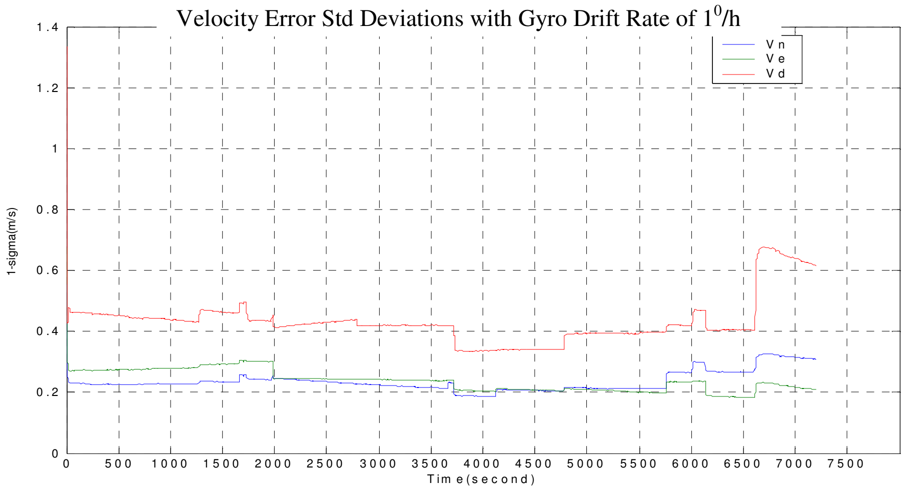


Figure 7.30 The cg Node Velocity Error Standard Deviations in Case 1A

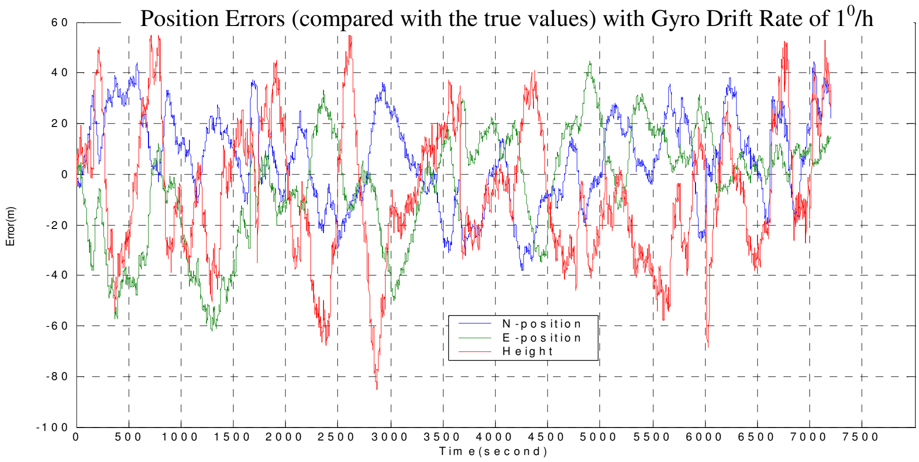


Figure 7.31 The cg Node Position Errors in Case 1A

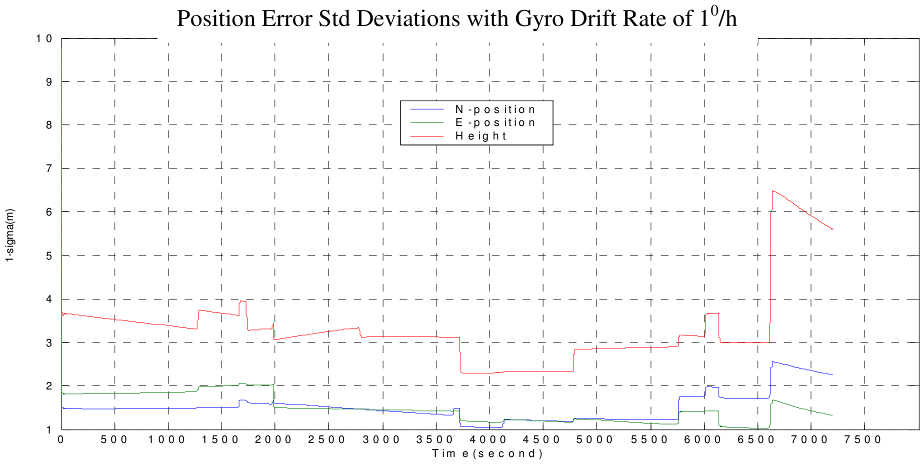


Figure 7.32 The cg Node Position Error Standard Deviations in Case 1A

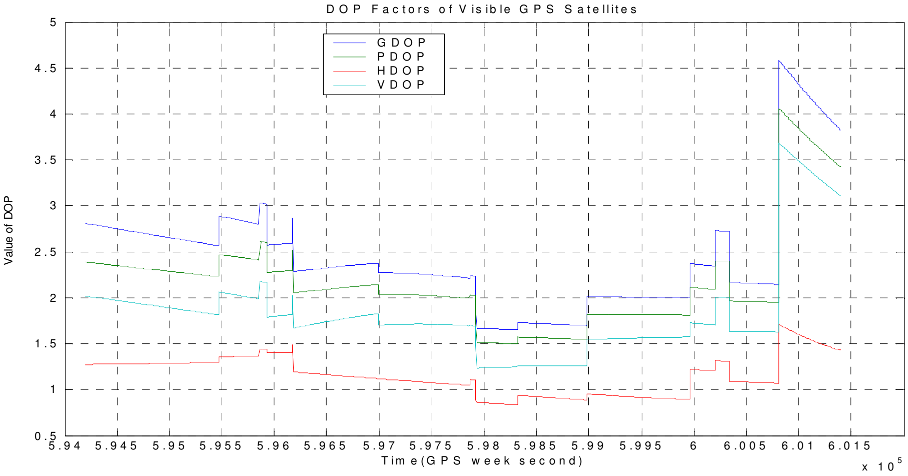


Figure 7.33 DOP Factors in Case 1A

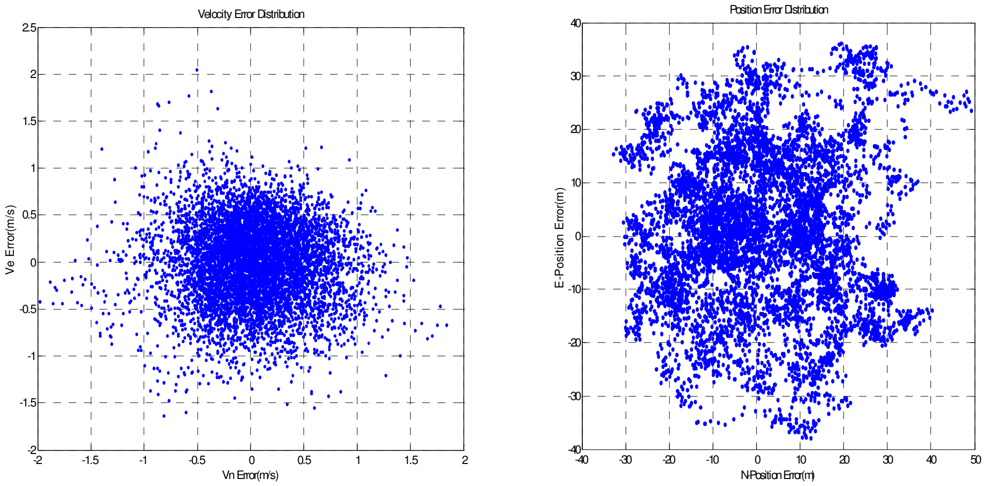


Figure 7.34 Velocity Error Distribution, Figure 7.35 Position Error Distribution

It should be noted that the changes of DOP factors are referenced to the GPS week time. However, the time scale lines in Figure 7.33 are completely corresponded to the time scale lines given in other figures.

From Figures 7.29 to 7.33, and the parallel simulation results in Case 1B, it is observed that during the different simulation time intervals, the changes of the DOP factors significantly affect the error standard deviations of the navigation state estimates but have less effect on the realistic state estimates. The following suggestion may explain why abnormal changes of the visible satellite geometries have different effects on the state estimates than on their error covariances. These simulations show that the fusion filter acts as a low-pass filter that can remove the effects of poor GNSS geometries on the navigation state estimates. However, because the fusion filter does not select the optimal geometry of visible satellites but instead uses all the available GNSS measurements in order to monitor the fusion filter integrity, the poor GNSS geometry may raise the uncertainty of the error covariance estimates through the measurement matrix **H**. As a result, the covariance matrix could not be used as a sole means of monitoring the integrity of the fusion filter. It is necessary to further study the effects of the GNSS geometry on both the estimate accuracy and integrity of the fusion filter in the future research.

The simulation results are shown in Figure 7.36 to 7.44 for the case where the GNSS attitude information is only available for a short time, for example from 100 to

350 seconds, 3500 to 3650 second and 5500 to 5650 seconds.

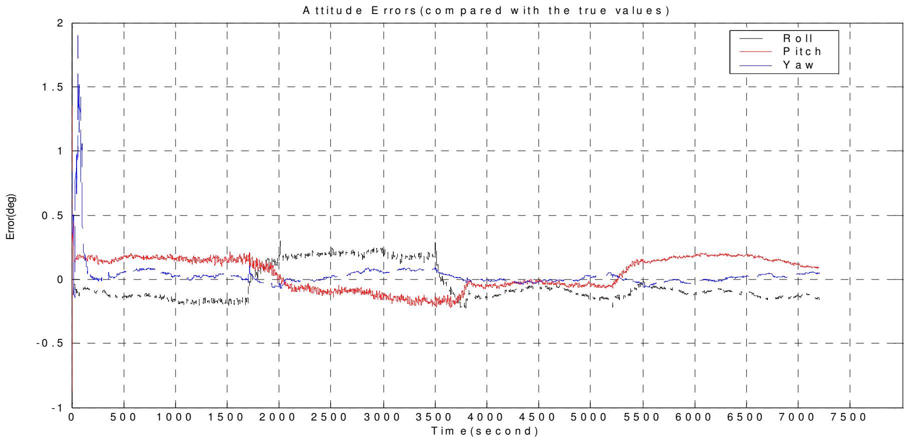


Figure 7.36 The cg Node Attitude Errors in Case 1B

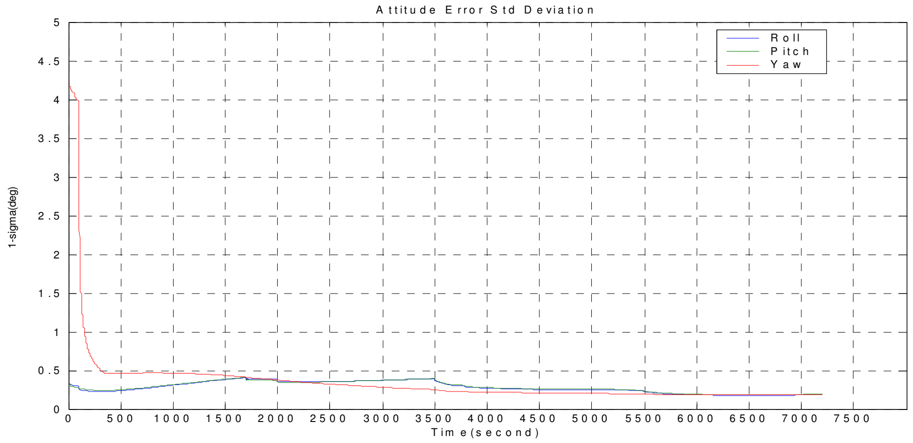


Figure 7.37 The cg Node Attitude Error Standard Deviations in Case 1B

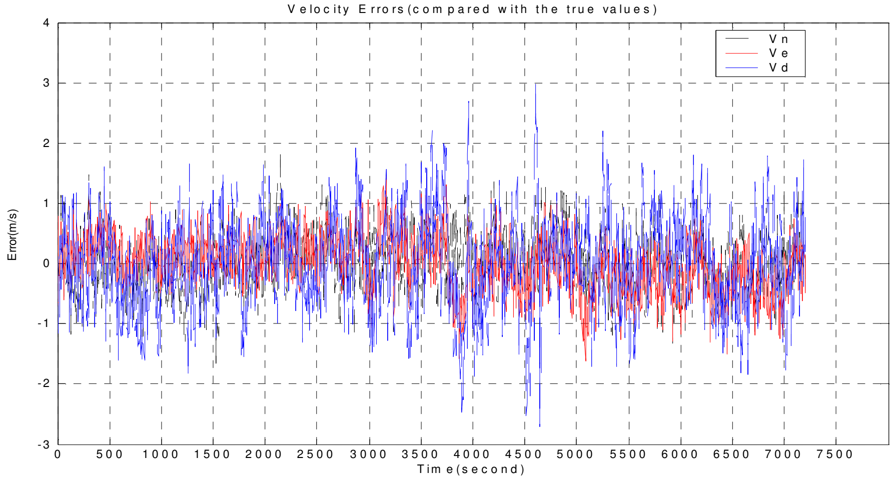


Figure 7.38 The cg Node Velocity Errors in Case 1B

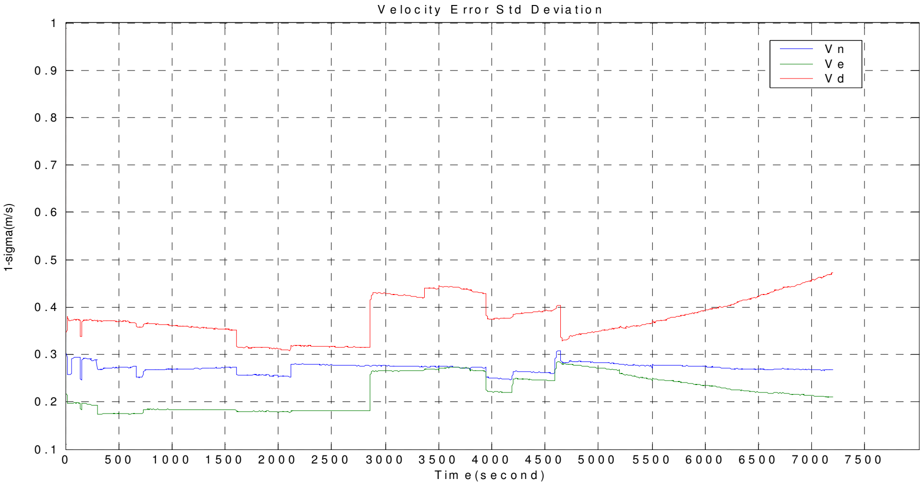


Figure 7.39 The cg Node Velocity Error Standard Deviations in Case 1B

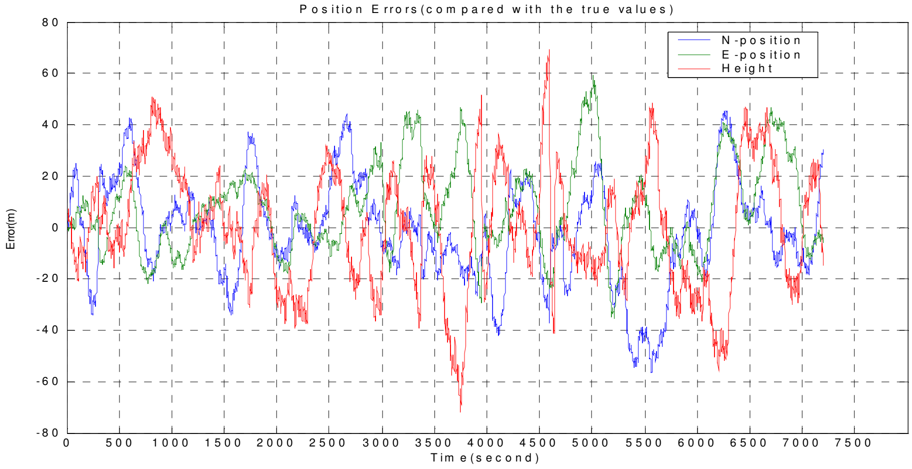


Figure 7.40 The cg Node Position Errors in Case 1B

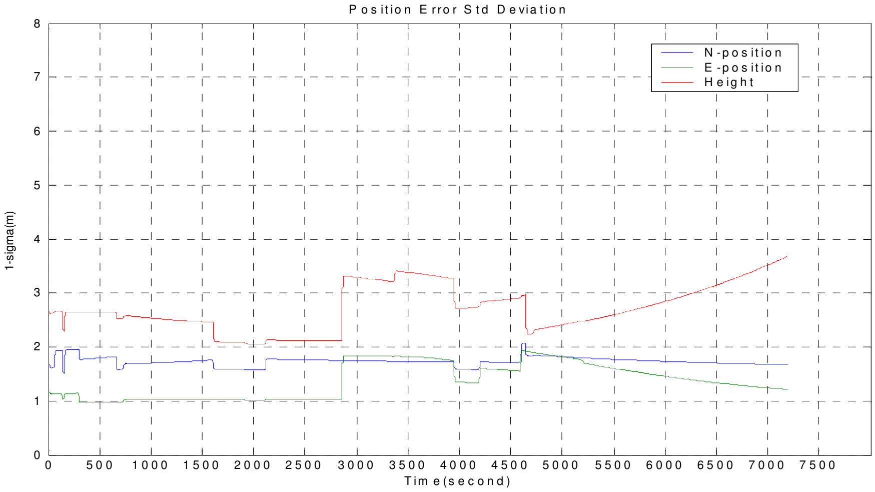


Figure 7.41 The cg Node Position Error Standard Deviations in Case 1B

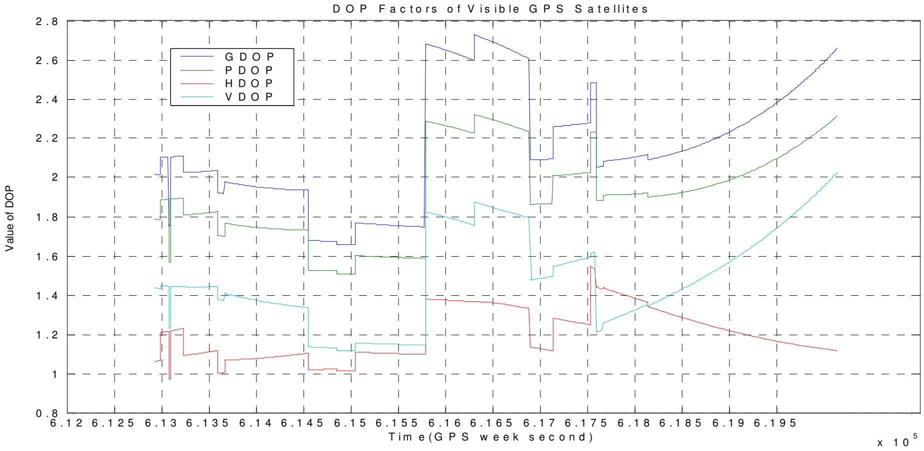


Figure 7.42 DOP Factors in Case 1B

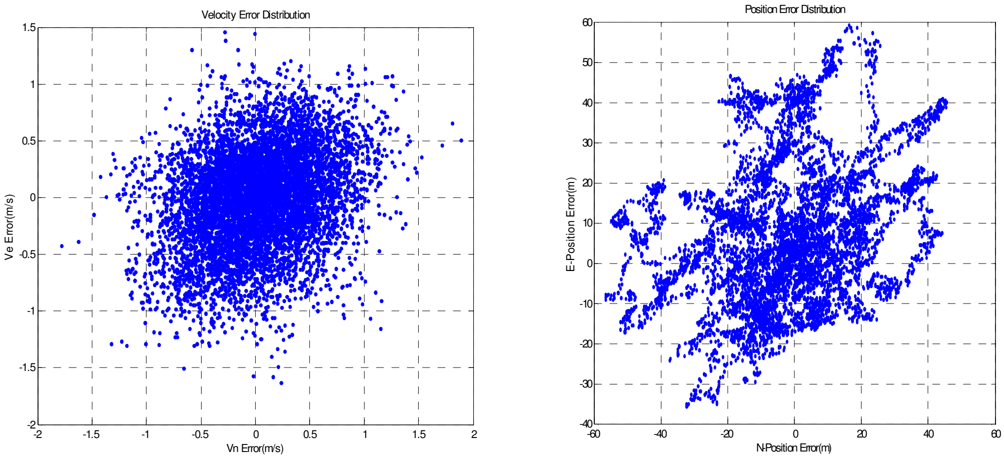


Figure 7.43 Velocity Error Distribution, Figure 7.44 Position Error Distribution

For a gyro bias of 10^0/h, the simulation results are given in Figures 7.45 to 7.53 where the GNSS information is not interrupted.

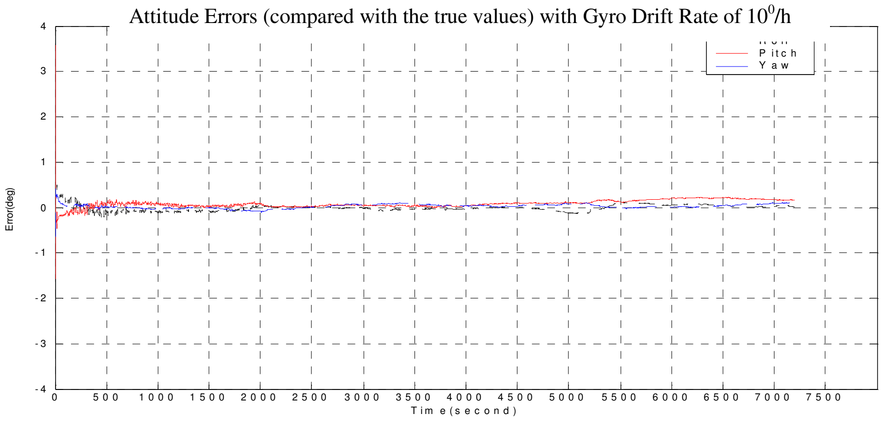


Figure 7.45 The cg Node Attitude Errors in Case 2A

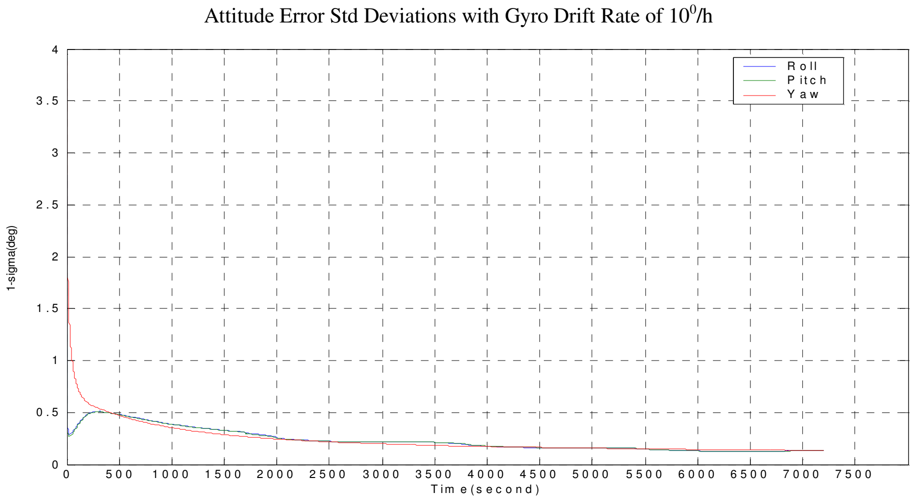


Figure 7.46 The cg Node Attitude Error Standard Deviations in Case 2A

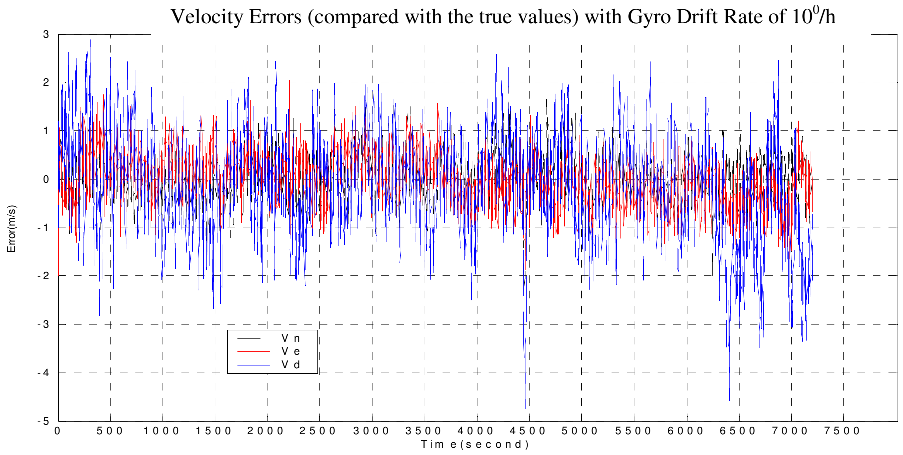


Figure 7.47 The cg Node Velocity Errors in Case 2A

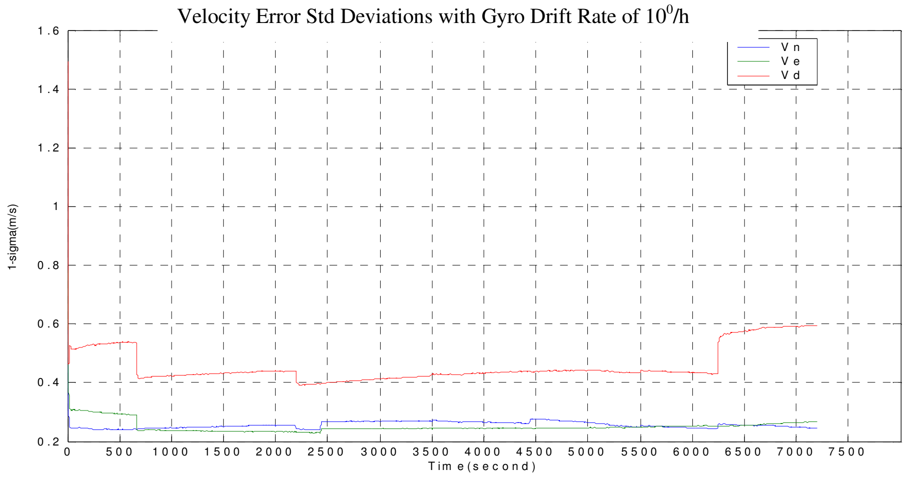


Figure 7.48 The cg Node Velocity Error Standard Deviations in Case 2A

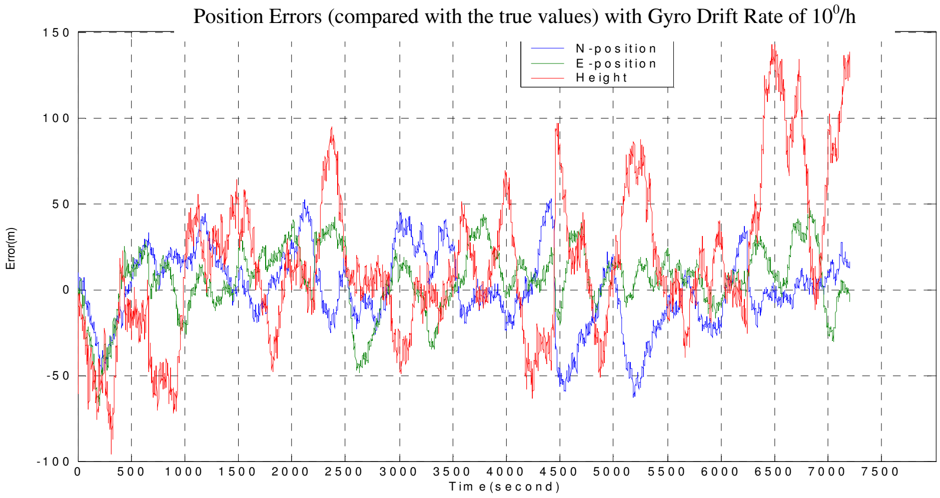


Figure 7.49 The cg Node Position Errors in Case 2A

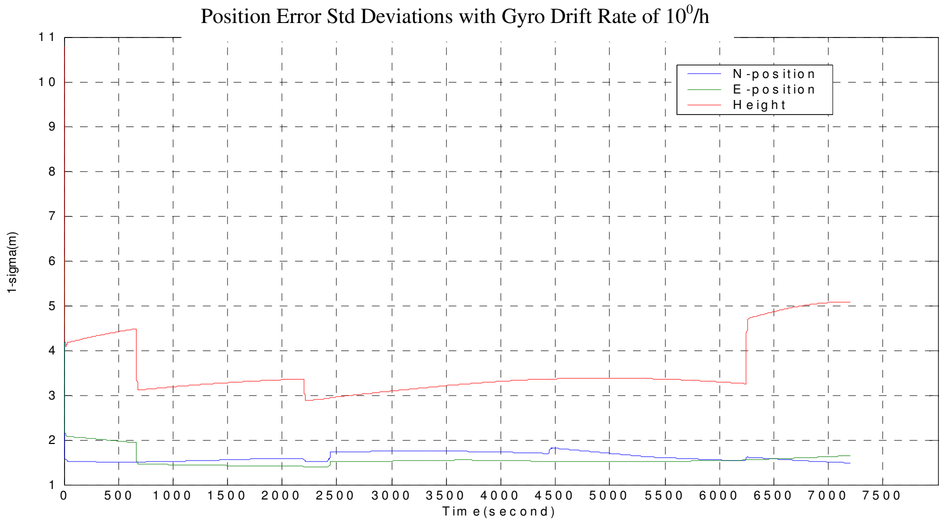


Figure 7.50 The cg Node Position Error Standard Deviations in Case 2A

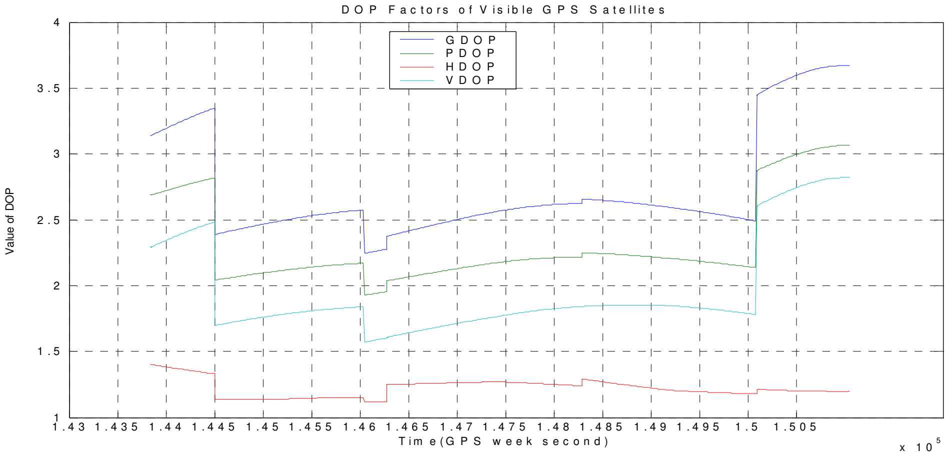
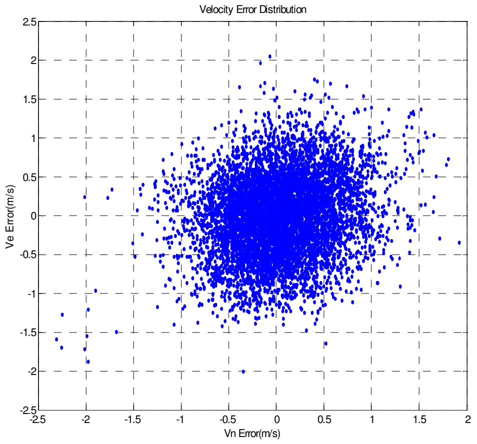


Figure 7.51 DOP Factors in Case 2A



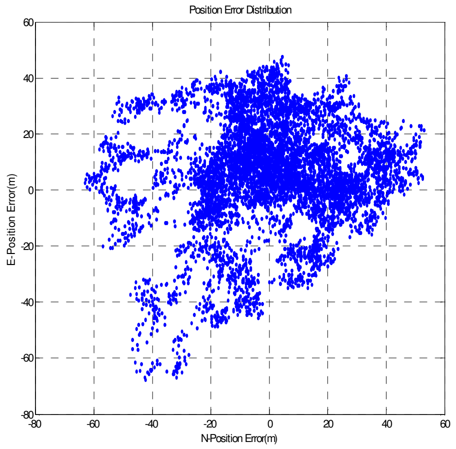


Figure 7.52 Velocity Error Distribution, Figure 7.53 Position Error Distribution

The simulation results are given in Figures 7.54 to 7.62 for the case where GNSS attitude information is interrupted from 600 to 950 second, 3500 to 3650 seconds and 5500 to 5650 seconds,

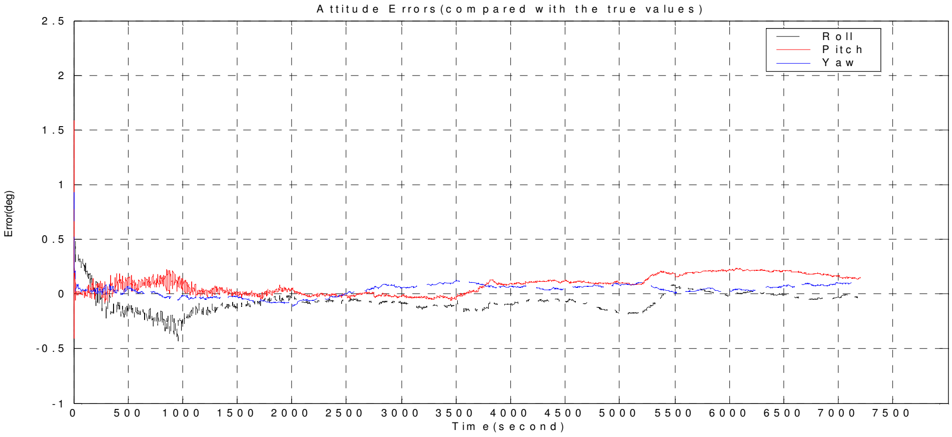


Figure 7.54 The cg Node Attitude Errors in Case 2B

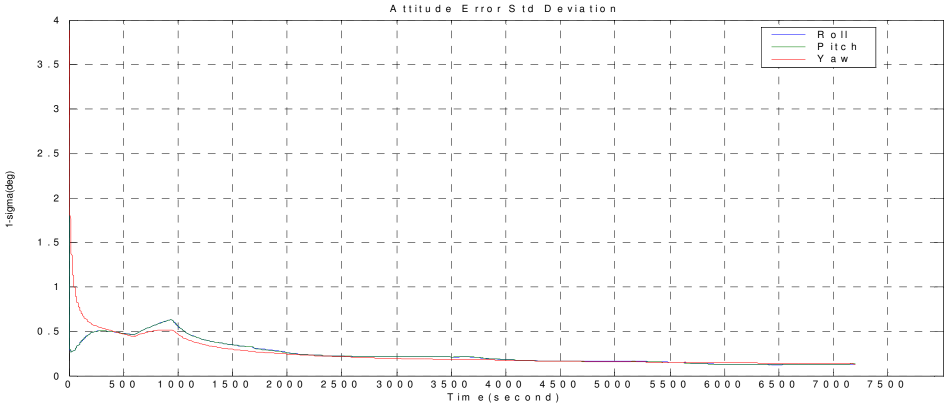


Figure 7.55 The cg Node Attitude Error Standard Deviations in Case 2B

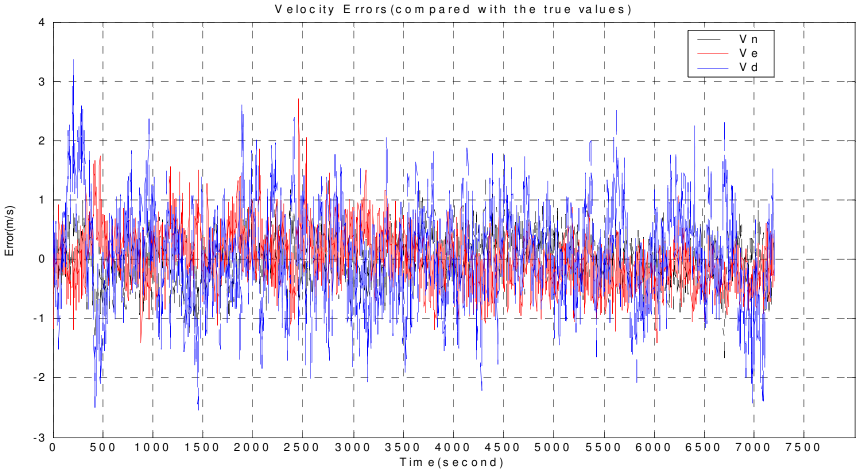


Figure 7.56 The cg Node Velocity Errors in Case 2B

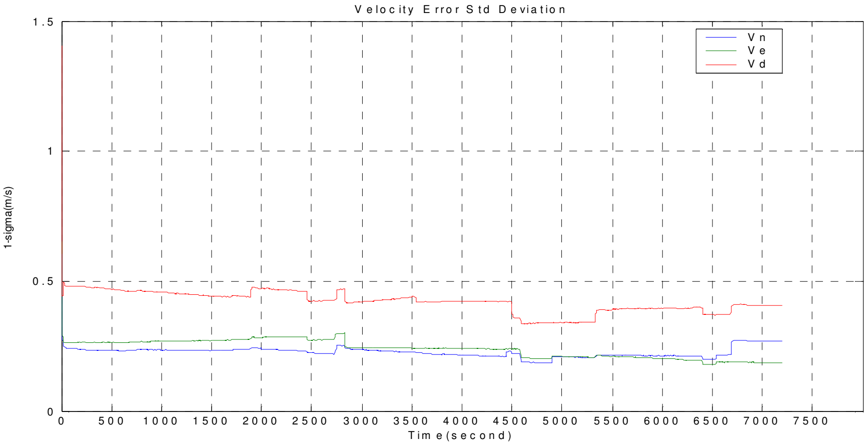


Figure 7.57 The cg Node Velocity Error Standard Deviations in Case 2B

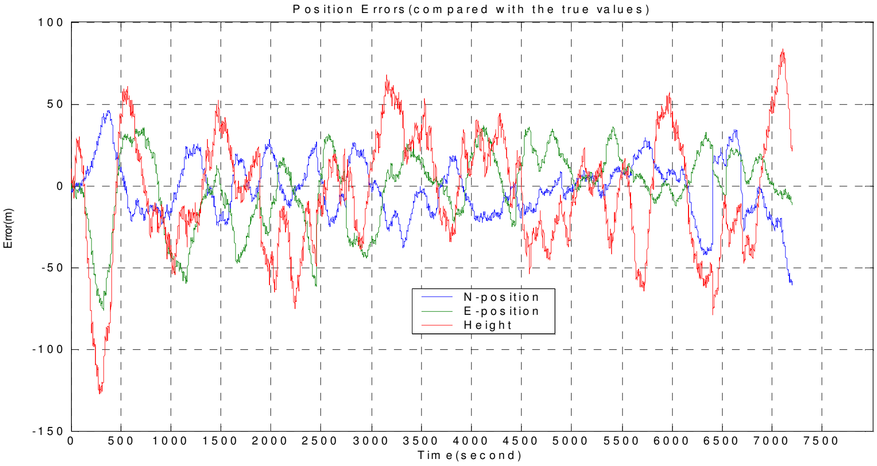


Figure 7.58 The cg Node Position Errors in Case 2B

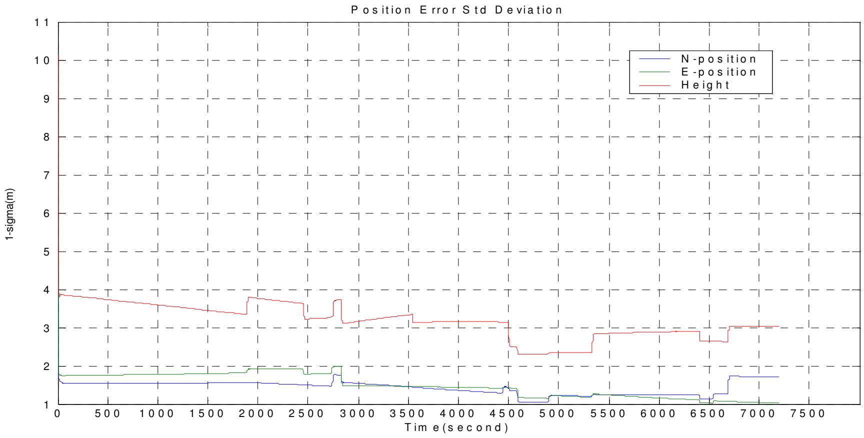


Figure 7.59 The cg Node Position Error Standard Deviations in Case 2B

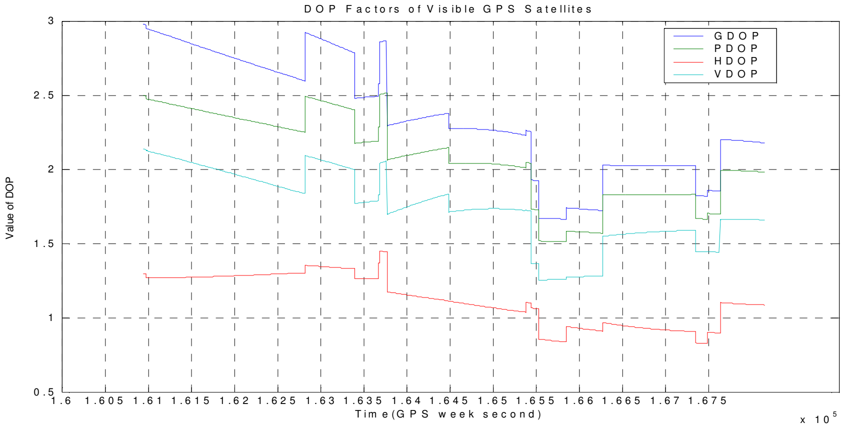
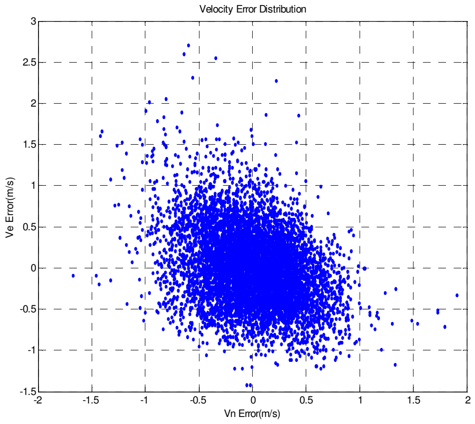


Figure 7.60 DOP Factors in Case 2B



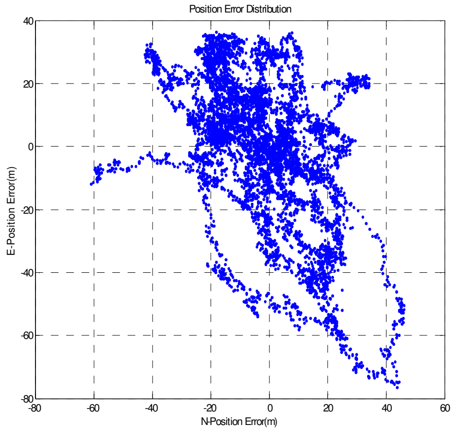


Figure 7.61 Velocity Error Distribution, Figure 7.62 Position Error Distribution

Further simulation studies were performed for other SRIMU configurations, low quality inertial sensors and GNSS-based position and velocity information. The simulation results are summarised in Table 7-2. These simulation results have shown that by fusion of low quality SRIMUs with raw GNSS measurements and GNSS-based attitude information, the distributed data fusion filters algorithms developed in this thesis can determine satisfactory navigation states although GNSS attitude information is only available for a short time, typically 1-2 minutes. It has also shown that the requirement for the drift or bias performance of gyro sensors of up to 40^0/h can be accommodated in a distributed inertial network systems. However, it should be noted that GNSS-based attitude information can significantly improve the accuracy of estimates of the attitude and velocity states. It is clear from these simulation results that the DOP factor has a significant effect on the accuracy of the estimates of aircraft position and velocity errors.

Although use of GNSS-based position and velocity information may produce better estimates of the navigation states, these fusion methods have a significant disadvantage. If the number of visible satellites is less than four, then GNSS-based position and velocity is unavailable. GNSS attitude information alone cannot reduce the degradation of the navigation velocity state.

Table 7-3 Summary of Simulation Results (gyro bias is 40^0/h)

Error State	4-Cone (1σ) PR+PRR	4-Cone (1σ) PR+PRR GNSS Att. Interruption	4-Cone (1σ) PR	4-Cone (1σ) PR,GNSS Att. Interruption	4-Cube (1σ) PR+PRR	5-Cone (1σ) PR+PRR
ϕ, θ ψ	$<0.2^0$ $<0.2^0$	$0.2^{0\,-}0.4^0$ 0.3^0	$<0.3^0$ $<0.3^0$	0.25^0 - 0.7^0 0.3^0	$<0.2^0$ $<0.2^0$	$<0.2^0$ $<0.2^0$
Vn, Ve Vd	0.25 – 0.4 m/s 0.5 m/s	0.25 - 0.4 m/s 0.5 m/s	0.6 m/s 0.8 – 1.0 m/s	0.6 – 0.7 m/s 0.8 – 1.0 m/s	0.25 - 0.4 m/s 0.5 m/s	0.2 – 0.4 m/s <0.5 m/s
Horizontal Vertical	1.5 - 2.5 m 2 – 5 m	1.5 – 2.5 m 2 - 5 m	2.5 - 3.0 m 5 m	2.5 - 3.0 m 5 –8 m	1.5 - 2.5 m 2 – 5 m	1.5 - 2.5 m 2 – 5 m

7.4.2 Simulation Results at Slave Nodes

For the simulation of the data fusion filters at all slave modes, position velocity and attitude information is available from the cg node or multifunctional GNSS sensor. Figures 7.63 to 7.67 show the simulation results without aiding attitude information interruptions and with a gyro bias of up to 40^0/h.

Although the accuracy of the velocity and attitude estimates at the slave node is less than the estimates at the cg node, the accuracy of the navigation state estimates still satisfies the requirements of the navigation states and local motion compensation and other airborne avionics systems.

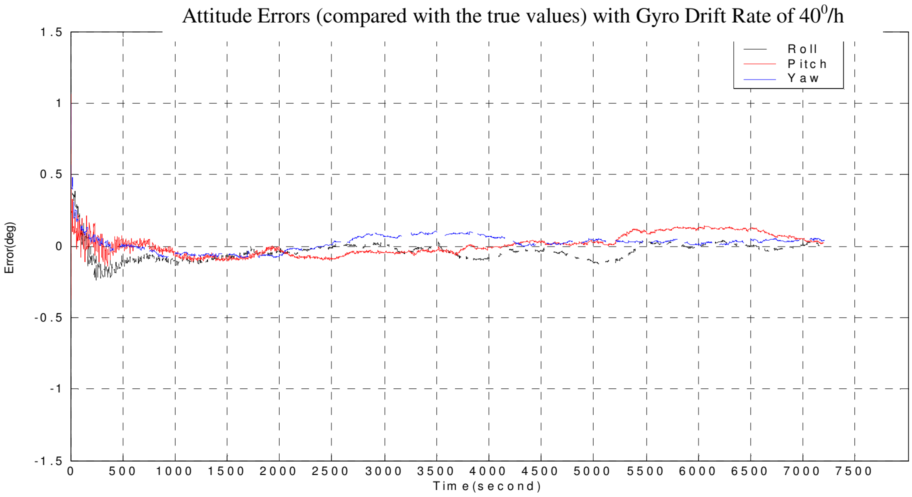


Figure 7.63 The Slave Node Attitude Errors in Case 3

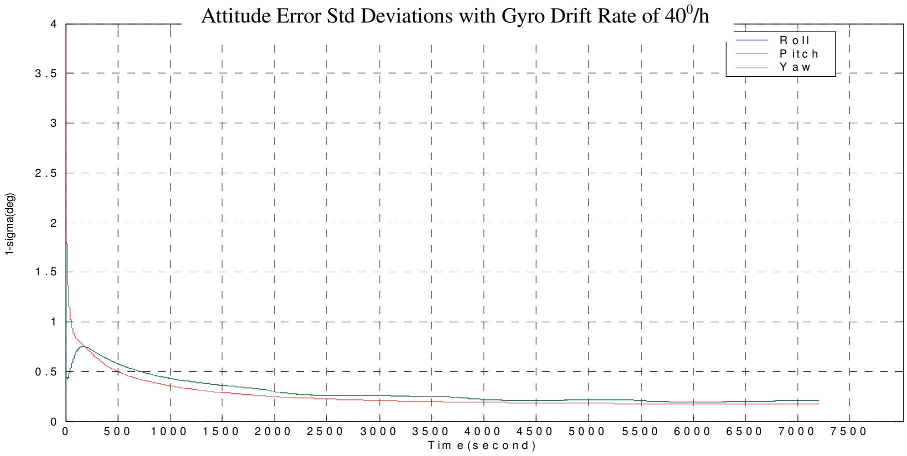


Figure 7.64 The Slave Node Attitude Error Standard Deviations in Case 3

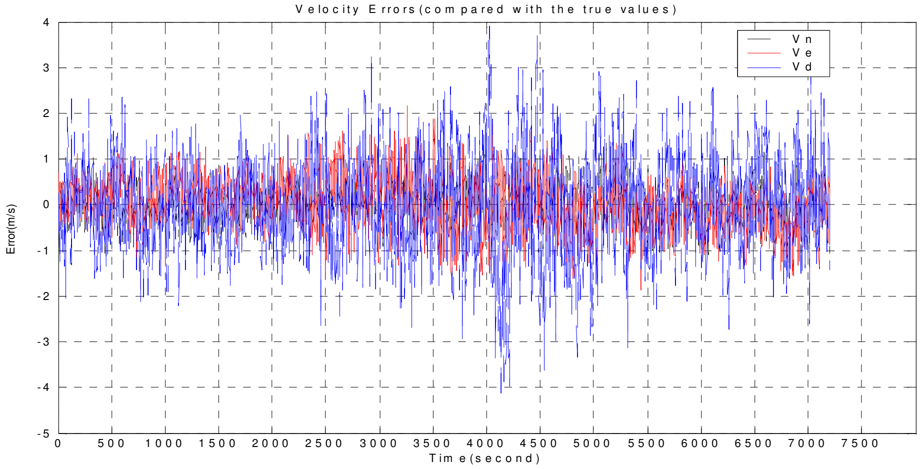


Figure 7.65 The Slave Node Velocity Errors in Case 3

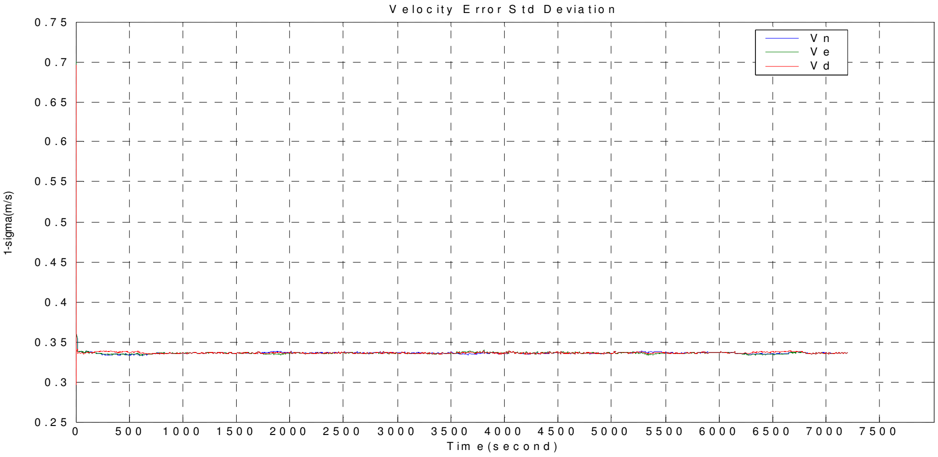


Figure 7.66 The Slave Node Velocity Error Standard Deviations in Case 3

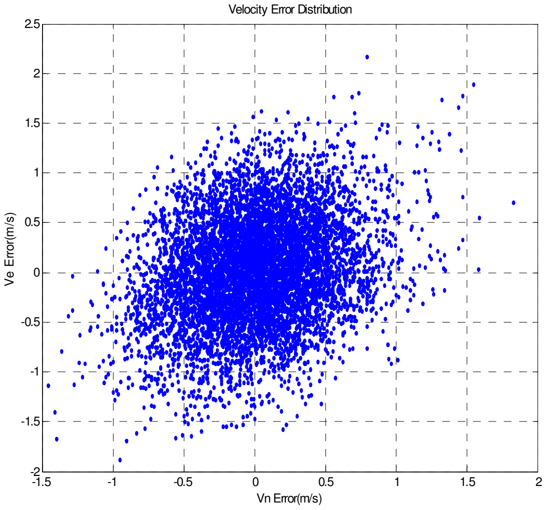


Figure 7.67 Velocity Error Distribution at the Slave Node in Case 3

7.4.3 Data Fusion Filter Integrity Testing Results

This simulation study tests the integrity of the distributed data fusion filters. Sensor failures have been inserted in the SRIMU and GNSS measurements. The statistical characteristics of two testing methods described in Sections 3.2 and 6.5 are examined and the simulation results are shown in Figures 7.68 to 7.76.

Figures 7.68 to 7.70 shown the test results for the case where one gyro has a jump failure signal after 6000 seconds for 100 seconds and one accelerometer has a jump failure signal after 2000 seconds in a 4-gyro cone configuration.

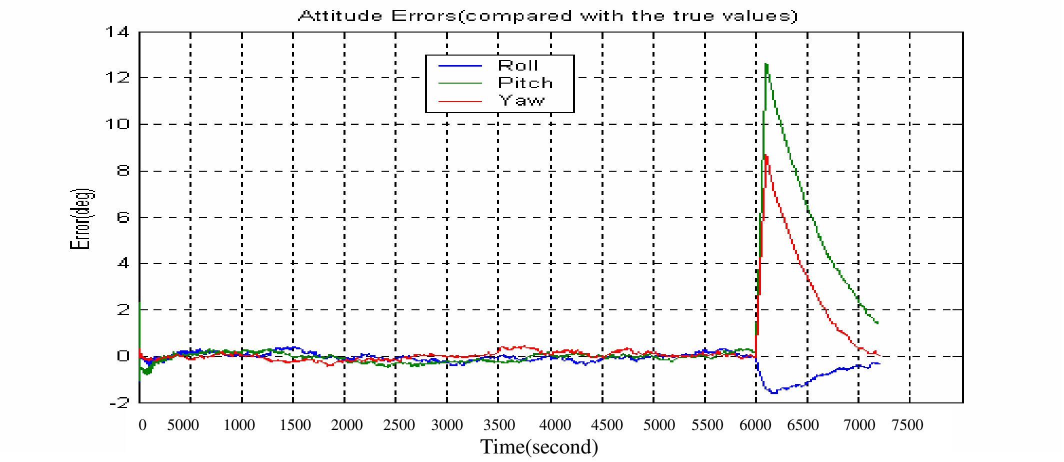


Figure 7.68 Attitude Errors at the cg Node in Case 4

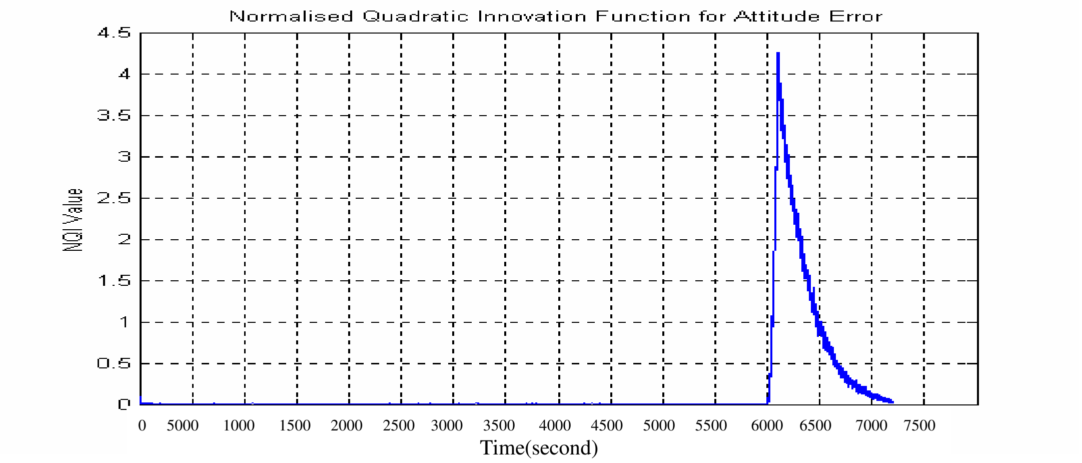


Figure 7.69 NQI for Attitude Innovation at the cg Node in Case 4

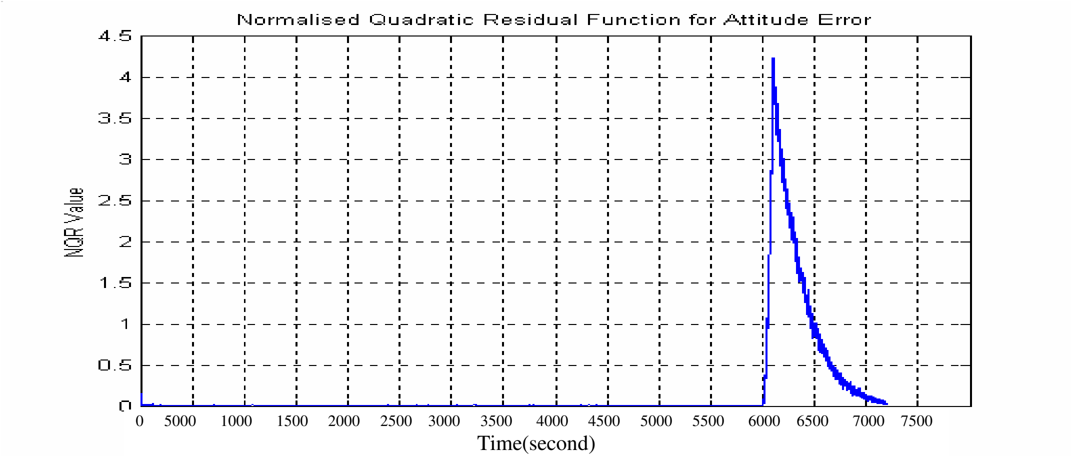


Figure 7.70 NQR for Attitude Residual at the cg Node in Case 4

Figures 7.71 to 7.76 show the test results for the case where one of visible GPS satellites has a jump failure signal after 4000 seconds for 100 seconds.

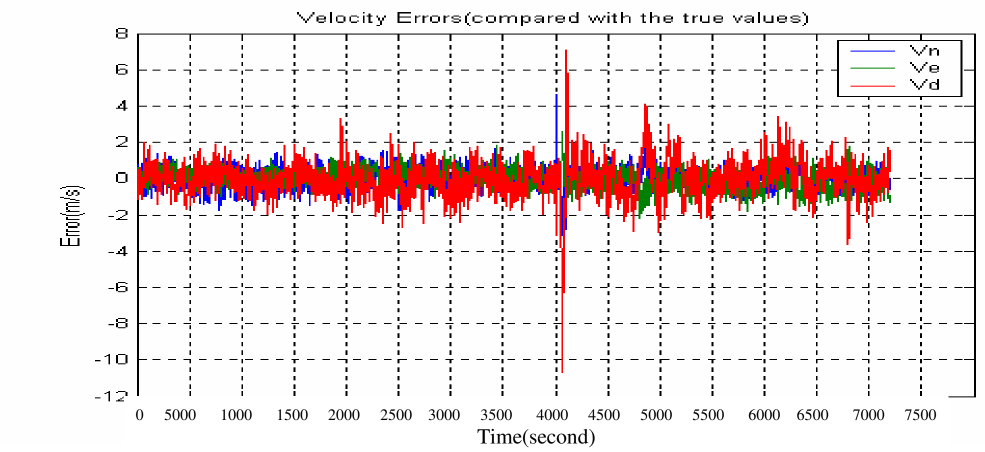


Figure 7.71 Velocity Errors at the cg Node in Case 4

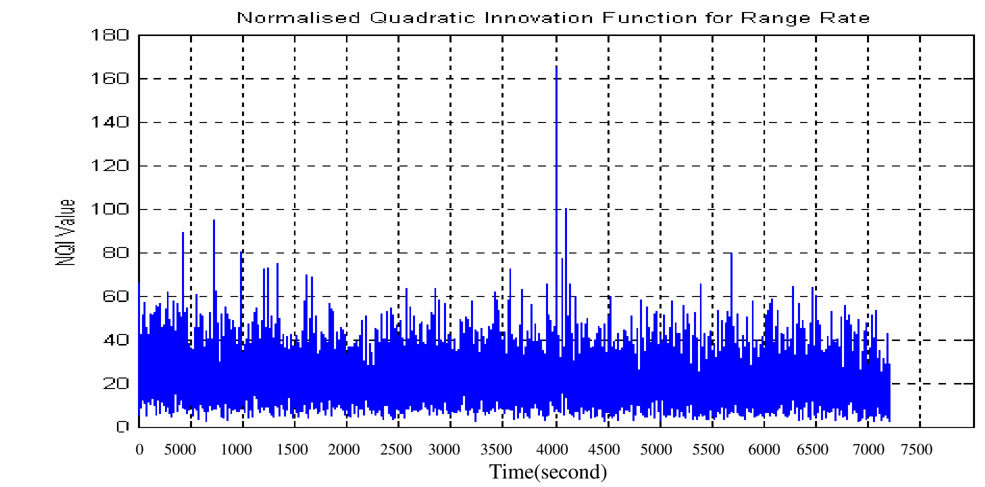


Figure 7.72 NQI for Range Rate Innovation at the cg Node in Case 4

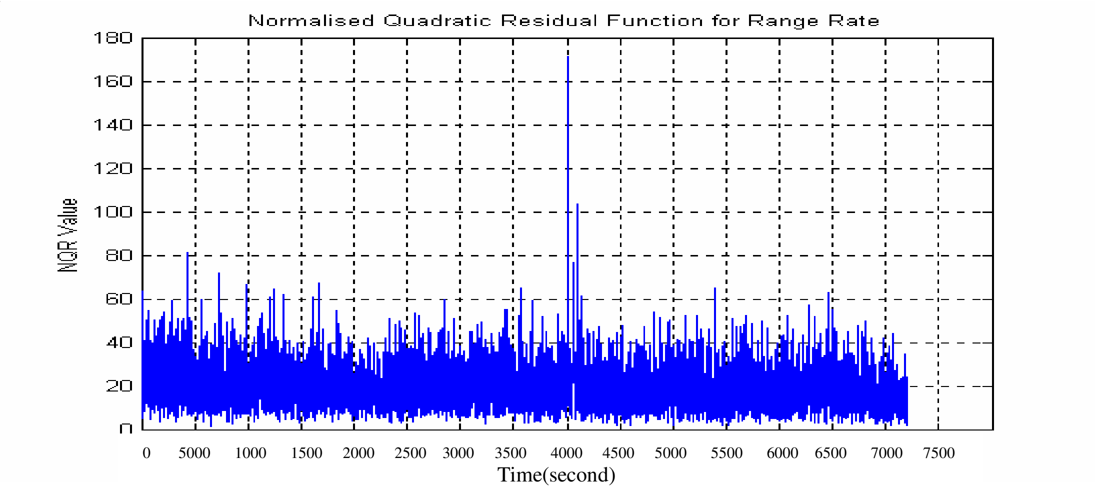


Figure 7.73 NQR for Range Rate Residual at the cg Node in Case 4

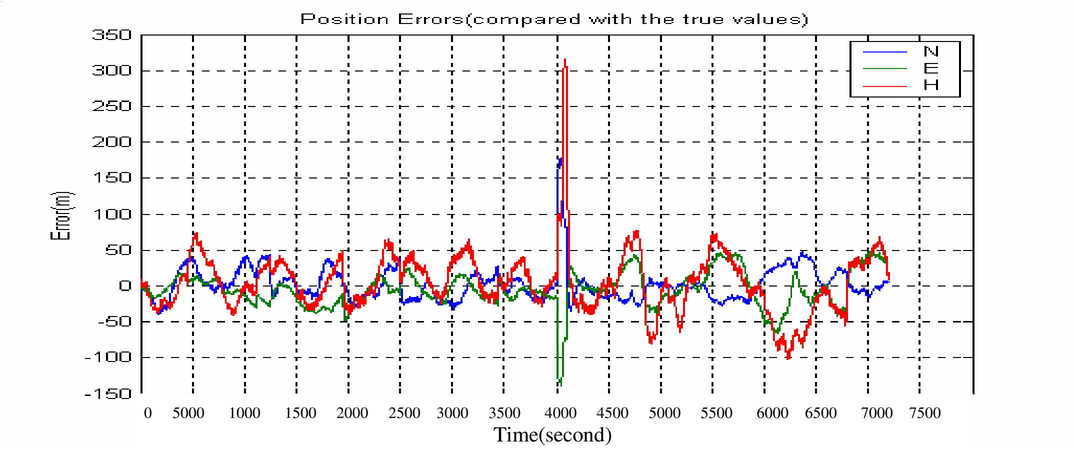


Figure 7.74 Position Errors at the cg Node in Case 4

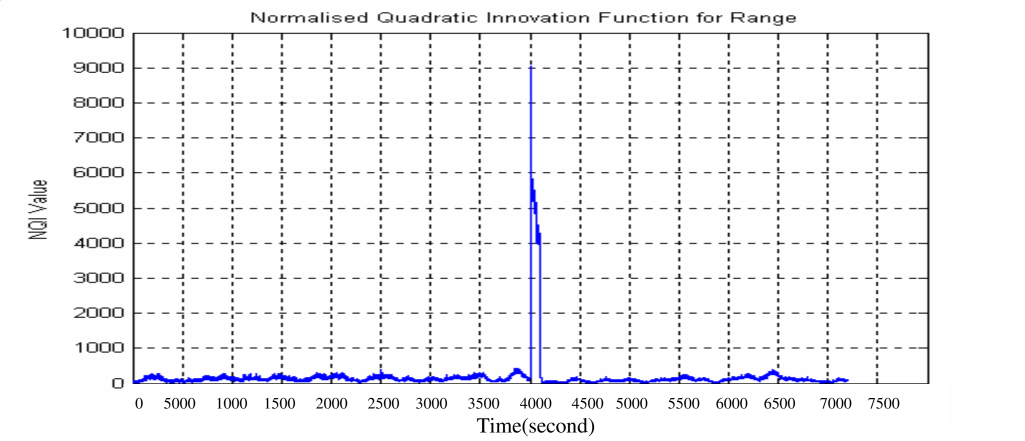


Figure 7.75 NQI for Range Innovation at the cg Node in Case 4

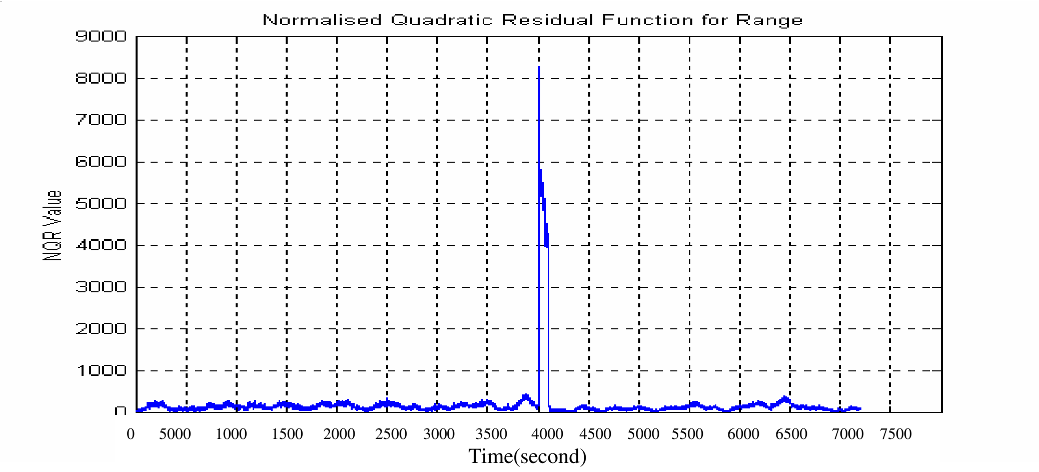


Figure 7.76 NQR for Range Residual at the cg Node in Case 4

These simulation results show that the normalised quadratic innovation (NQI) function can successfully detect the measurement bias signals caused by GPS signal failures and the normalised error quadratic (NEQ) function can monitor the abnormal shifts of the navigation states, caused by large sensor drifts or uncertainties of the model parameters.

The results obtained from these simulation researches are summarised below:

- The DOP factor affects the accuracy of estimates of navigation state errors. Large DOP factors result in large error variances of the navigation states. When the GNSS pseudorange measurements are only used as the observables of the data fusion filter, the jump of GDOP will cause large estimate errors of the position and velocity, even abnormal velocity errors if the GDOP has a step jump exceeding one unit. These simulations have shown that the effect of GDOP factors on the estimate accuracy can be significantly reduced if GNSS pseudorange and Doppler measurements are used as the filter measurements.
- Use of the GNSS PR or PR and PRR measurements with GNSS-based attitude information as the observables of the distributed data fusion filters can obtain similar attitude estimate accuracy but use of the GNSS PR and PRR information has the significant advantage in terms of control of velocity errors. Therefore, GNSS PRR information should be used to control system velocity errors.

- Up to 30-minute GNSS attitude interruption can be tolerated after the distributed data fusion filter has run for 10 minutes, depending on the performance of the gyro sensors. Maximum attitude error generally occurs in roll angle whereas errors in the pitch and yaw angles can be controlled by the velocity information.
- The navigation performance of a 4-sensor cube configuration is similar to a 4-sensor cone configuration. The 5-sensor cone has better attitude accuracy than 4-sensor configurations. The accuracy of attitude estimates in inertial network systems depend on the accuracy of the GNSS-based attitudes. The accuracy of the navigation velocity state largely depends on the accuracy of GNSS Doppler measurements.
- The NQI and NQR methods can be used to monitor the distributed data fusion filter integrity. It is recommended that the distributed inertial network system should combine three failure detection functions: the short time MW GLRT, the sequential MW-GLT FDI and the distributed data fusion filter integrity monitoring to achieve adequate safety requirements.

7.5 Summary

This chapter described the software simulation system and a number of simulation studies. The main activities included:

1. Development of a modularised multisensor software simulation system and description of its architecture and associated functions.
2. Introduction of inertial simulation and GPS simulation systems and the architecture of multi-mode data fusion filters at network nodes.
3. Performance analysis of numerous simulation studies to test and evaluate the methods and algorithms developed in this thesis.

Chapter 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Introduction

This chapter summarises the research undertaken in this thesis. Section 8.2 highlights the major contributions of the thesis. The main conclusions obtained from the simulation studies are summarised in Section 8.3. Further research studies are discussed in Section 8.4.

8.2 Contributions

This thesis covers the development of multisensor data fusion methodologies for the design, development, analysis, and simulation of reliable fault-tolerant aircraft navigation systems. The use of the methods developed in this thesis and their applications to low-cost (low-quality) inertial network systems integrated with multifunctional GNSS sensors can afford benefits in the cost, size, weight, accuracy, reliability and integrity of aircraft navigation systems. These methods can also be used for the design of navigation and attitude determination systems for marine vessels and space vehicles. The major contributions made during this PhD study are summarised in the following sections.

8.2.1 Multisensor Data Fusion Model

The application of the recent RNP concept to the design of multisensor navigation systems leads to a problem: how can multiple sensor systems and their measurements be combined to achieve the RNP requirements for aircraft navigation systems? The solution to this problem is based on multisensor data fusion technologies.

1. An original generalised multisensor data fusion model was presented for the design of aircraft navigation systems. This model consists of four modules, each addressing specific aspects of the RNP requirements and implementing different functions to ensure that the RNP requirements of aircraft navigation systems can be satisfied. This model can be used to assist and guide navigation system engineers to develop reliable and accurate aircraft navigation systems and to reduce the development cycle and costs.
2. The navigation and attitude determination equations of inertial systems and GNSS were normalised. It is shown in the development of the simulation software system that the normalised equations provide a convenient method for the design of data fusion filters for distributed inertial network systems. This normalisation can be used to provide a standardised development procedure for inertial/GNSS integrated systems.

8.2.2 Sensor FDI and Network Integrity Methods

Two important modifications to the traditional GLRT FDI methods are the MW-GLRT methods and sensor error compensation filters. A combination of sensor-level and system-level FDI procedures can achieve high levels of reliability and integrity for distributed sensor network systems.

1. In comparison with traditional GLRT methods, the sequential MW-GLRT methods exceed the performance of the previous methods in the detection of various sensor failures. The simulation studies show that the sequential methods provide improved detection performance and efficiency, for example, detection of drift sensor failures in a short time and compensation for normal SRIMU measurement errors.

2. These improved algorithms are further extended by using the outputs of the distributed fusion filters to monitor the integrity of an inertial network system, including the inertial vector states and the similar node states in the inertial network system. The simulation studies also show that any jump and drift failures in distributed sensor systems can be successfully detected by the use of the inertial network integrity monitoring algorithms.
3. In comparison with current integrity monitoring methods, which generally detect failure at the system level, the simulation studies have shown that the combination of sensor-level FDI and system-level integrity monitoring procedures greatly improves the integrity and fault tolerance of distributed sensor network systems.

8.2.3 Distributed Data Fusion Algorithms

Two forms of distributed sensor systems have been examined, focusing on distributed inertial network fusion algorithms.

1. Two data fusion algorithms were developed for state-identical distributed sensor systems, known as the Type I and Type IA algorithms. This form of distributed systems covers the majority of current designs of integrated navigation systems. It is significant that these two algorithms can be used for the design of integrated navigation systems for space, air and land vehicles and also marine vessels. This method was used in the SHINE programme to develop a multi-mode hybridised navigation filter.
2. Two data fusion algorithms were developed for state-associated distributed sensor network systems, known as the Type II and Type IIA algorithms. It is expected that this method can be used in the design of the next generations of aircraft navigation systems, particularly inertial network navigation systems for military aircraft.
3. An inertial network sensing model was developed and two algorithms were described to determine the dynamic transformation matrices.
4. Innovative distributed inertial network fusion algorithms were presented;

including distributed inertial data fusion algorithms and distributed state fusion algorithms. The simulation studies show that the inertial network fusion algorithms can significantly improve the accuracy of the estimations of the inertial vector states and similar node states in an inertial network system. Moreover, these algorithms are capable of performing dynamic alignment and calibration of inertial sensor systems in an inertial network system.

8.2.4 Multisensor Simulation Environment

A multisensor simulation system environment was developed to simulate GPS, skewed redundant inertial systems and other sensor systems. This simulation system has been used to test and evaluate the range of data fusion algorithms developed in this thesis for distributed sensor network systems. It was also used to evaluate the FDI algorithms and the multi-mode hybridised Kalman filter developed for the SHINE programme. This simulation system provided an extremely convenient tool for the design and development of multisensor navigation systems.

8.3 Conclusions

A wide range of simulation studies was performed during the course of this research study. The main simulation results are summarised below.

8.3.1 SRIMU Configurations

Several SRIMU configurations were evaluated and the following results were obtained:

1. Optimal SRIMU configurations were determined on the basis of trade-off of the minimum GDOP factor, the allocation of normalised variances of measurement errors along orthogonal body axes and the FDI capability. Coplanar sensor installation should be avoided in order to obtain the maximum FDI capability.

2. Accuracies of measured state estimates and degradation of performance of two 4-sensor SRIMU configurations and four 5-sensor configurations were compared in the presence of several sensor failures. The full cone SRIMU configurations in 4-sensor or 5-sensor SRIMU systems are recommended because degradation of SRIMU performance for the cone configuration is minimised for the various configurations simulated in this thesis.

8.3.2 FDI and Integrity Monitoring Algorithms

The MW-GLRT FDI algorithms and the inertial network integrity monitoring algorithms were tested in the simulation environment and the main test results are summarised below.

1. The short-time MW-GLRT method can eliminate abnormal measurement noise and detect sensor hard failures. Furthermore, the sequential MW-GLRT algorithms can efficiently detect mid-value and drift failures (which may degrade the accuracy of the SRIMU systems) without generating an alarm.
2. SRIMU error compensation filters can improve the accuracy of SRIMU systems and the performance of the MW-GLRT algorithms.
3. A combination of inertial data assimilation algorithms with the MW-GLRT algorithms can detect various sensor failures in an inertial network system. The filter-based integrity monitoring algorithms can effectively monitor the integrity of the distributed data fusion filters.

8.3.3 Distributed Inertial Network Fusion Algorithms

The distributed inertial network fusion algorithms were tested during this study and the significant conclusions are summarised below.

1. By applying distributed inertial network fusion algorithms to a low-cost inertial network system (low-cost inertial sensors are characterised by a gyro bias ranging from 10^0/h to 40^0/h in this thesis), the attitude states at the cg node were estimated at 0.2 degrees (one sigma) even though GNSS-

based attitude information was unavailable for thirty minutes. The attitude states at slave nodes were estimated at 0.26 degrees (one sigma) even though gyros at the node have a bias of 40^0/h. These simulation studies imply that low-cost inertial network systems would be used in future avionics systems to replace high-quality inertial sensors and that SRIMU configurations would replace orthogonal configurations.

2. Distributed inertial network fusion filters can dynamically correct and align SRIMU systems in an inertial network system. Therefore, the traditional inertial system alignment algorithms are no longer necessary in distributed inertial network systems, for example, the transfer and fine alignments that have been used to correct inertial system errors at initialisation and to in flight align low-accurate slave inertial systems to a high-accurate master inertial system. Traditional transfer alignments need an aircraft to perform specified manoeuvres, typically covering horizontal straight flight followed by an 'S' flight for up to ten minutes. The elimination of traditional alignment procedures allows an aircraft to perform manoeuvres without the consideration of the above constraints. This is particularly important for military aircraft where these constraints can increase the risk to aircraft and pilots.

8.4 Future Work

Although the main solutions to the problem of data fusion methodologies in the development of aircraft multisensor navigation systems have been addressed in this thesis, the author feels that further research is necessary in several areas.

8.4.1 SRIMU Calibration and Error Dynamic Models

In SRIMU configurations, inertial sensors are installed along skewed axes with respect to the orthogonal instrument frame axes. Such configurations results in gyros and accelerometers which are more sensitive to translational and rotational motion of an aircraft in comparison with an orthogonal IMU configuration. These coupling relationships should be modelled in SRIMU error models. This thesis has mainly identified the misalignment angles in SRIMU configurations as sources of error. For that reason, ***one area of the future research should be directed to develop more comprehensive SRIMU error dynamic models***. The author believes that this research will improve the performance of distributed inertial network fusion filters in terms of estimate accuracy and network integrity.

From review of the literature, current research is rarely concerned with SRIMU calibration although the calibration of orthogonal IMUs has been standardised. It is recommended that ***another area of the future research should be development of SRIMU calibration algorithms.*** It is particularly important to SRIMU manufacturers and companies developing SRIMU-based integrated navigation systems.

8.4.2 Distributed Data Fusion Problems

Normalised GNSS attitude determination equations were developed during this study programme. If the author had had sufficient time, the normalised GNSS attitude determination equation would have been integrated into the measurement models of distributed fusion filters to evaluate the performance of real-time (on-the-fly) kinematic GNSS/inertial network systems. In comparison with direct resolution of the GNSS attitude algorithm, the combination of the normalised GNSS attitude determination equation with inertial network dynamic models can provide benefits in terms of real-time carrier phase ambiguity resolution and GNSS attitude estimate accuracy. However, many current kinematic GPS/inertial integrated systems use a two-step estimation procedure. First, the GPS attitude determination equation is resolved to obtain the GPS-based attitude solution. Then, the GPS-based attitude solution is used as the observable of the integrated navigation filter. Therefore, ***further research should be directed to real-time kinematic GNSS/ inertial network systems***. It is not necessary to seek for an integer solution of carrier phase ambiguity in such studies. Nevertheless certain criteria have to be developed to minimise a cost function of attitude errors.

Recent research shows that the fusion of imaging sensors and navigational sensors can detect and locate obstacles in specific phases of flight. Fusion of real-time images, navigation states and GIS database information can provide a vision capability that allows an aircraft to operate in harsh weather environments and may reduce controlled flight into terrain and approach-and-landing accidents. Therefore, ***further research in multisensor data fusion methodologies should be directed to develop comprehensive sensor fusion algorithms for flight safety, enhancement of synthetic vision systems, terrain obstacle avoidance and guidance and proximity ground warning systems, as well as aircraft navigation systems***.

8.4.3 Inertial Network Failure Detection

Detection filter techniques have been used in many control systems for failure detection. It is likely that these methods could be used in distributed inertial network systems. ***Further research should be pointed to evaluate the usability and detection performance of failure detection filters in inertial network systems.***

REFERENCES

[1] Kayton, M. and Walter, W.R., *Avionics Navigation Systems*, John Wiley and Sons, Inc. 2nd, 1997.

[2] RTCA, *Minimum Aviation System Performance Standards (MASPS): Required Navigation Performance for Area Navigation*, DO-236, Jan. 1997.

[3] The Johns Hopkins University Applied Physics Laboratory, *GPS Risk Assessment Study*, VS-99-07, Jan. 1999.

[4] http://www.ecacnav.com/rnav/RNP-RNAV.htm

[5] ICAO GNSSP IP11, *The Evolution from Area Navigation (RNAV), Required Navigation Performance (RNP), to RNP RNAV*, Oct. 22-Nov 1, 2001, http://gps.faa.gov/Library/Data/RNAVPaper.DOC

[6] Blackman, S.S., *Multiple Targets Tracking with Radar Applications*. Artech House Inc. 1986

[7] Hall, D. L., *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Inc. 1992.

[8] Hall, D. L. and Llinas, J., *An Introduction to Multisensor Data Fusion*, Proceedings of the IEEE Vol. 85, No. 1, Jan. 1997, 6 -23

[9] http://www.data-fusion.org/article.php?sid=70 (accessed on 12th July 2002)

[10] Luo, R. C. and Kay, M. G., *Multisensor Integration and Fusion in Intelligent System*, IEEE Trans. On System, Man, and Cybernetics, Vol. 19, No. 5, Sept/Oct 1989, 901-931.

[11] Albus, J. S., *The NIST Real-Time Control (RCS): An Application Survey*. At http://www.isd.mel.nist.gov/documents/publist.htm.

[12] Kokar, M. and Kim, K., *Review of Multisensor Data Fusion: Architecture and Techniques*, Proceedings of The International Symposium on Intelligent Control, Chicago, Illinois, USA, Aug. 1993, 261-266

[13] Johnson, B. W., *Design and Analysis of Fault-Tolerant Digital Systems*. Addison-Wesley Publishing Company, Inc., 1989.

[14] Brock L.D. and Schor A.L., *Modular Avionics System Architecture (MASA)– The Impact of Fault Tolerance*, Proceedings of IEEE/AIAA/NASA 9th Digital Avionics System Conference, 1990, 310-315.

[15] Miller, H. and Hilts, D.A., *Fault Tolerant Integrated Inertial Navigation/ Global Positioning Systems for Next Generation Spacecraft*, Proceedings of IEEE/AIAA 10th Digital Avionics Systems Conference, 1991, 207-212.

[16] Mclntyre, M.D.W. and Gossett, C. A., *The Boeing 777 Fault Tolerant Air Data Inertial Reference System, A New Venture in Working Together*, IEEE/AIAA 14th Digital Avionics Systems Conference, 1995, 178-183.

[17] Audsley, N.C and Burke, M., *Distributed Fault-Tolerant Avionic Systems – A Real-Time Perspective*, IEEE Aerospace Conference, Vol. 4, 1998, 43-60.

[18] Swanson, D.L., *Evolving Avionics Systems from Federated to Distributed Architectures*, Proceedings of AIAA/IEEE/ SAE 17th Digital Avionics Systems Conference, Vol. 1, 1998, D26/1-D26/8.

[19] Hammett, R., *Design by Extrapolation: An Evaluation of Fault Tolerant*

Avionics, IEEE AES System Magazine, Vol. 17, No. 4, April 2002, 17-25.

[20] Kwan, W; DeVires; Lupash, L. and McLeen J., *AC-130U Aircraft GPS-INS Integration*, Proceedings of the first Regional Conference on Aerospace Control Systems, 1993, 782-786.

[21] Prasad, V.B, *Fault Tolerant Digital Systems*, IEEE Potentials, Vol. 8, No. 1, Feb. 1989, 17-21.

[22] Tuttle, F.L.; Kisslinger, R.L and Ritzema, D.F., *F-15 S/MTD IFPC Fault Tolerant Design*, Proceedings of NAECON, 1990, 501-506.

[23] Subbarao, E.V., *Fault Tolerant Digital System Design*, Proceedings of Southeastcon, Vol. 1, 1991, 124-128.

[24] Gertler, J.J., *Survey of Model-based Failure Detection and Isolation in Complex Plants*, IEEE Control System Magazine, Vol. 8, No. 6, Dec. 1988, 3-11.

[25] Patton, R.J., *Fault Detection and Diagnosis in Aerospace Systems using Analytical Redundancy*, Computing & Control Engineering Journal, Vol. 2, No. 3, May 1991, 127-136.

[26] Hegg, J. *Enhanced Space Integrated GPS/INS (SIGI),* IEEE Aerospace and Electronics Systems Magazine, Vol. 17, No. 4, April 2002, 26-33.

[27] Krogmann, U.K. *Optimal Integration of Inertial Sensor Functions for Flight-Control and Avionics*, Proceedings of AIAA/IEEE 8th Digital Avionics System Conference, Oct. 17-20, 1988, 148-155.

[28] Sebring, D.L. and McIntype, M.D., *An Air Data Inertial Reference System for Future Commercial Airplanes*, Proceedings of AIAA/IEEE 8th Digital Acionics Systems Conference, 1988, 308-313.

[29] Sudano, J.J.; Preisig, J.R. and Pokotylo, J., *Improved Fault Detection Using a Selected Grouping of Parity Equations for Advanced Flight Control Systems*, Proceedings of 1988 IEEE NAECON, 1988, 1565-1569.

[30] Sheffels, M.L., *A Fault-Tolerant Air Data/Inertial Reference Unit*, IEEE AES System Magazine, March 1993,48-52.

[31] Ebner, R.E. and Klein, A.D., *Integrated Navigation/Flight Control for Future High Performance Aircraft*, ADARD-AG-314, Analysis, Design and Synthesis Methods for Guidance and Control Systems. 1995, V1-8.

[32] Kelley, R.T.; Carlson, N.A. and Berning, S., *Integrated Inertial Network*, Proceedings of IEEE PLANS, 1994, 439-446.

[33] Berning, S.; Howe, P. and Jenkins, T., *Theater-Widde Reference Information Management*, Proceedings of IEEE NAECON, 1996, 122-128.

[34] Kaiser, J.; Beck, G. and Berning, S., *Vital Advanced Inertial Network*, Proceedings of IEEE PLANS, 1998, 61-68.

[35] Diesel, J. W., *Integration of GPS/INS for Maximum Velocity Accuracy*, Journal of The Institute of Navigation, Vol. 34, No. 3, Fall 1987.

[36] Rounds, S. and Casey, J., *A Fully Integrated GPS/Doppler/Inertial Navigation System*, Proceedings. of the Institute of Navigation Technical Meeting, Sept. 1987.

[37] Schneider, A.M. and Maida, J.L., *A Kalman Filter for an Integrated Doppler/GPS Navigation System,* Proceedings of IEEE PLANS '88, 1988, 408-415.

[38] Diesel, J.W. *GPS/INS Integration for Civil Aviation*, Proceedings of National Telesystems Conference, 1991, 223 -228

[39] Hyslop, G.; Gerth, D.; Kraemer, J. *GPS/INS integration on the Standoff Land Attack Missile (SLAM)***,** IEEE Aerospace and Electronics Systems Magazine, Vol. 5, No. 7, July 1990, 29 –34.

[40] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, Mathematics in Science and Engineering, Academic Press, 1977.

[41] Maybeck, P. S., *Stochastic Models, Estimation, and Control*. Vol. 1. Academic Press, 1979.

[42] Schlee, F.H.; Toda, N.F.; Islam, M.A. and Standish, C.J., *Use of an external cascade Kalman filter to improve the performance of a Global Positioning System (GPS) inertial navigator,* Proceedings of the IEEE NAECON 1988, 345-350.

[43] Wade, M. and Grewal, M. S., *Analysis of A Cascaded INS Calibration Filter*, Proceedings of the IEEE 1988, 366-373.

[44] Karatsinides, S. E., *Enhancing Filter Robustness in Cascaded GPS-INS Integrations*, IEEE Trans. On Aerospace and Electronic Systems, Vol. 30, No. 4, Oct., 1994, 1001-1008.

[45] Bell, W.B; Gorre, R.G; and Cockrell, L.D, *Cascading Filtered DTS Data into a Loosely Coupled GPS/INS System*, Proceedings of IEEE PLANS'98, 1998, 586-593.

[46] Carlson, N.A., *Federated Filter for Fault-Tolerant Integrated Systems*, Proceedings of 1988 IEEE PLANS, 1988, 110-119.

[47] Felter, S.C., *An overview of decentralized Kalman filter techniques*, Proceedings of IEEE Southern Tier Technical Conference, 1990, 79-87.

[48] Carlson, N.A., *Federated Square Root Filter for Decentralized Parallel Processors*, IEEE Trans. On Aerospace and Electronic Systems, Vol. 26 No. 3, May 1990, 517-525.

[49] Lawrence, P.J., Jr. and Berarducci, M.P., *Comparison of Federated and Centralized Kalman Filter with Fault Detection Considerations*, Proceedings of IEEE PLANS, 1994, 703-710.

[50] Levy, L.J., *Suboptimality of Cascaded and Federated Kalman Filters*, Proceedings of the ION 52nd Annual Meeting, Navigational Technology for the 3rd Millennium, June 1996, 399-407.

[51] Tupysev, V.A., *Federated Kalman Filter Via Formation of Relation Equations in Augmented State Space*, Journal of Guidance, Control, and Dynamics, Vol. 23, No. 3, May-June 2000, 391-398.

[52] Carlson, N.A., *Federated Filter for Computer-Efficient, Near-optimal GPS Integration,* Proceedings of IEEE PLANS, 1996, 306-314.

[53] Lawrence, P.J., Jr. and Berarducci, M.P., *Navigation Sensor, Filter, and Failure Mode Simulation Results Using the Distributed Kalman Filter Simulator (DKFSIM)*, Proceedings of IEEE PLANS, 1996,697-710.

[54] Carlson, N.A., *Federated Filter for Distributed Navigation and Tracking Applications,* Proceedings of the ION 58th Annual Meeting and the CIGTF 21st Guidance Test Symposium, June 2002, 340-353.

[55] Kerr, T. H., *Decentralized Filtering and Redundancy Management for*

Multisensor Navigation, IEEE Trans. on Aerospace and Electronics Systems, Vol. AES-23, No. 1, Jan., 1987, 83-119.

[56] Wei, M. and Schwarz, K.P., *Testing a Decentralized Filter for GPS/INS Integration*, Proceedings of IEEE PLANS, 1990, 429-435.

[57] Bar-Shalom, Y. and Fortmann, T. E., *Tracking and Data Association*, Academic Press, New York, 1988.

[58] Liggins, M. E., II; Chee-Yee Chong; Kadar, I.; Alford, M. G.; Vannicola, V. and Thomopoulos, S., *Distributed Fusion Architectures and Algorithms for Target Tracking*, Proceedings of the IEEE, Vol. 85, No. 1, Jan. 1997, 95-107.

[59] Brumback, B. D. and Srinath, M. D., *A Fault-Tolerant Multisensor Navigation System Design*, IEEE Trans. On Aerospace and Electronic Systems, Vol. 23, No. 6, Nov., 1987, 738-755.

[60] Speyer, J.L., *Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem*, IEEE Trans. On Automatic Control, Vol. 24, No. 2, April 1979, 266-269

[61] Willsky, A.S.; Bello, M.G.; Castanon, D.A.; Levy, B.C; and Verghese, G.C., *Combing and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks*, IEEE Trans. On Automatic Control, Vol.27, No.4, Aug., 1982, 799-813.

[62] Hashemipour, H.R.; Roy, S. and Laub, A.J.; *Decentralized Structure for Parallel Kalman Filtering*, IEEE Trans. on Automatic Control, Vol. 33, No. 1, Jan., 1988, 88-94.

[63] Hong, L., *Centralized and distributed multisensor integration with uncertainties in communication networks*, IEEE Trans. on Aerospace and Electronic Systems, Vol. 27, No. 2, Mar., 1991, 370-379.

[64] Hong, L., *Adaptive Distributed Filtering in Multicoordinated System*, IEEE Trans. on Aerospace and Electronic Systems, Vol. 27, No. 4, July, 1991, 715-724.

[65] Roy, S.; Hashemi, R.H. and Laub, A.J**.,** *Square root parallel Kalman filtering using reduced-order local filter*, IEEE Trans. on Aerospace and Electronic Systems, Vol. 21, No. 2, Mar. 1991, 276-289.

[66] Paik, B.S. and Oh, J.H., *Gain fusion algorithm for decentralised parallel Kalman filters*, IEE Proceedings of Control Theory and Applications, Vol. 147, No. 1, Jan. 2000, 97-103.

[67] Raol, J. R. and Girija, G., *Sensor data fusion algorithms using square-root information filtering*, IEE Proceedings of Radar, Sonar and Navigation, Vol. 149, No. 2, Apr 2002, 89-96.

[68] Durrant-Whyte, H.F.; Rao, B.Y.S. and Hu, H**.,** *Toward a fully decentralized architecturefor multi-sensor data fusion*, Proceedings of 1990 IEEE International Conference on Robotics and Automation, Vol.2 1331 –1336.

[69] Rao, B.S. and Durrant-Whyte, H.F. *Fully decentralised algorithm for multisensor Kalman filtering*, IEE Proceedings-Control Theory and Applications, Vol. 138, No.5, Sept., 1991 413 –420.

[70] Berg, T.M. and Durrant-Whyte, H.F., *Distributed and Decentralized Estimation*, Proceedings of international Conference on Intelligent Control and Instrumentation, SICICI'92, Vol. 2. Singapore 1992, 1118 –1123.

[71] Evans, F.A. and Wilcox, J. C., *Experimental Strapdown Redundant Sensor Inertial Navigation System*, Journal of Spacecraft and Rockets, Vol. 7, No. 9, Sept., 1970, 1070-1074.

[72] Gilmore, J. P. and Mckern, R. A., *A Redundant Strapdown Inertial Reference Unit (SIRU)*, Journal of Spacecraft and Rockets, Vol. 9, No. 1, Jan., 1972, 39-47.

[73] Potter, J.E. and Deckert, J.C., *Minimax Failure Detection and Identification in Redundant Gyro and Accelerometer System*, Journal of Spacecraft, Vol. 10, No. 4, April, 1973, 236-243.

[74] Wilcox, J.C., *Competitive Evaluation of Failure Detection Algorithms for Strapdown Redundant Inertial Instruments*, Journal of Spacecraft, Vol. 11, No. 7, July 1974, 525-530.

[75] Daly, K.C.; Gai, E. and Harrison, J.V., *Generalized Likelihood Ratio Test for FDI in Redundant Sensor Configurations*, Journal of Guidance Control, Vol. 2, No. 1, Jan., 1979, 9-17.

[76] Gai, E.; Harrison, J.V. and Daly, K.C., *FDI Performance of Two Redundant Sensor Configurations, IEEE Trans on Aerospace and Electronics Systems*, Vol. AES-15, No. 3, May 1979

[77] Hall, S.R.; Motyka, P.; Gai, E. and Deyst, J. J. JR., *In-Flight Parity Vector Compensation for FDI,* IEEE Trans. on Aerospace and Electronics Systems, Vol. AES-19, No. 5, Sept., 1983, 668-675.

[78] Sturza, M.A., *Skewed Axis Inertial Sensor Geometry for Optimal Performance,* Proceedings of AIAA/IEEE 8th Digital Avionics System Conference, 1988, 128-135.

[79] Sturza, M.A., *Navigation System Integrity Monitoring Using Redundant Measurements*, NAVIGATION: Journal of The Institute of Navigation, Vol. 35, No. 4, 1988-89, 69-87.

[80] Brown, A. and Sturza, M.A., *The Effect of Geometry on Integrity Monitoring Performance*, Proceedings of the ION 46th Annual Meeting June 26-28, 1990.

[81] Sturza, M.A. and Brown, A.K., *Comparison of Fixed and Variable Threshold RAIM Algorithms*, Proceedings of the ION 3rd International Technical Meeting, Sept., 1990.

[82] Diggelen, F. van and Brown, A., *Mathematical Aspects of GPS RAIM*, Proceedings of 1994 IEEE PLANS, 1994, 733-738.

[83] Parkison, B. W.; et al, *Global Positioning System: Theory and Applications*, Vol. I, II, AIAA Press, 1996.

[84] Chien, T. T. and Adams, M. B., *A Sequential Failure Detection Technique and Its Application*, IEEE Trans. On Automatic Control, Oct. 1976, 750-757

[85] Willsky A.S. and Jones, H.L., *A Generalized Likelihood Ratio Approach to Detection and Estimation of Jumps in Linear Systems*, IEEE Trans on Automatic Control, Vol. AC-21, No. 1, Feb., 1976.

[86] Willsky, A.S., *A Survey of Design Methods for Failure Detection in Dynamic Systems*, AGARD AG-224, 1977, 5-1 to 5-14.

[87] Kerr, T.H. *Statistical Analysis of a Two-Ellipsoid Overlap Test for Real-Time Failure Detection*, IEEE Trans. on Automatic Control, Vol. AC-25, No. 4, Aug. 1980.

[88] Brenner, M., *Integrated GPS/Inertial Fault Detection Availability*, Proceedings of ION GPS-95.

[89] Diesel, J. and Dunn, G., *GPS/IRS AIME Technology: Certification for Sole Means and Solution to RF Interference*, Proceedings of ION-GPS-96.

[90] Hanlon, P.D. and Maybeck, P.S., *Characterization of Kalman Filter Residuals in Presence of Mismodeling*, IEEE Trans. On Aerospace and Electronic Systems, Vol. 36, No. 1, Jan., 2000, 114-131.

[91] Minkler, G. and Minkler, J., *Theory and Application of Kalman Filtering*, Magellan Book, Co., 1993.

[92] Schervish, M.J. *Theory of Statistics*, *Springer Series in Statistics*. Springer-Verlag New York, Inc. 1995.

[93] EUROCONTROL and Institute of Geodesy and Navigation, *WGS 84 Implementation Manual*, Version 2.4, 1998..

[94] Lawrence, A., *Modern Inertial Technology, Navigation, Guidance, and Control*, 2nd, Springer-Verlag New York, Inc. 1998.

[95] Schmidt, G.T., *INS/GPS Technology Trends for Military Systems*, Draper Technology Digest 1998, 2-13

[96] Barbour, N. and Schmidt, G.T., *Inertial Sensor Technology Trends*, IEEE Sensors Journal, Vol. 1, No. 4, Dec., 2001, 332-3339

[97] The Royal Aeronautical Society, *Laser Gyros and Fibre Optic Gyros*, On-day Symposium Proceedings, 25 Feb., 1987,

[98] Yazdi, N., Ayazi, F., and Najiafi, K., *Micromachined Inertial Sensors,* Proceedings of IEEE, Vol 86, No. 8, Aug. 1998, 1640-1659.

[99] Kourepenis, A., Borenstein, J., Connelly, J., Elliott, R., Ward, P. and Weinberg, M., *Performance of MEMS Inertial Sensors*, Proceedings of IEEE PLANS, 1998, 1-8.

[100] Baruh, H., *Analytical Dynamics,* WCB/McGraw-Hill, 1999.

[101] Rogers, R. M., *Applied Mathematics in Integrated Navigation Systems*. AIAA, Inc., 2000.

[102] Benson, D.O., JR., *A comparison of Two Approaches to Pure-Inertial and Doppler-Inertial Error Analysis*, IEEE Trans. on Aerospace and Electronic Systems, Vol-AES-11, No. 4, July 1975, 447-455.

[103] Goshen-Meskin, D. and Bar-Itzhack, I.Y., *Unified Approach to Inertial Navigation System Error Modelling*, Journal of Guidance, Control, and Dynamics, Vol. 15, No. 3, May-June 1992.

[104] Scherzinger, B.M. and Reid, D.B., *Modified Strapdown Inertial Navigator Error Models*, Proceedings of 1994 IEEE PLANS, 1996, 426-430.

[105] Stewart, C. *GPS Pseudolites: Theory, Design, and Applications*, PhD Thesis, Stanford University, 1997.

[106] http://www.waasperformance.raytheon.com/

[107] http://www.esrin.esa.it/export/esaSA/GGGQI950NDC_navigation_0.html

[108] http://www.mitre.org/work/best_papers/best_papers_00/fernow_interop/fernow_interop.pdf

[109] Enge, P., Walter, T., Fullen, S., Kee, C., Chao, Y. and Tsai, Y., *Wide Area Augmentation of the Global Positioning System*, Proceedings of The IEEE, Vol. 84, No. 8, Aug., 1996, 1063-1088

[110] http://gps.faa.gov/Library/Data/waas/irbfl.pdf

[111] Grass, F. van, et al, *GPS Interferometric Attitude and Heading Determination: Initial Test Results*, NAVIGATION, Journal of The Institute of Navigation, Vol. 38, No. 4, Winter 1991-1992, 297-316.

[112] Cohen, C.E., Attitude Determination Using GPS, PhD Dissertation, Standford University, Dec. 1992.

[113] Brown, R.A**.,** *Instantaneous GPS Attitude Determination*, IEEE Aerospace and Electronic Systems Magazine, , Vol. 7 , No. 6 , June 1992, 3 - 8

[114] Crassidis, J.L. and Markley, F.L, *A New Algorithm for Attitude Determination Using Global Positioning System Signals*, www.acsu.buffalo.edu/gpsatt.pdf

[115] Teunissen, P.J.G., *The Least-Squares Ambiguity Decorrelation Adjustment: A Method for Fast GPS Integer Ambiguity Estimation*, Journal of Geodesy 1995, 65-82.

[116] Hill, C.D. and Euler, H.J., *An Optimal Ambiguity Resolution Technique for Attitude Determination*, IEEE 1996 Position Location and Navigation Symposium, April 1996, 262 – 269

[117] Pratt, M., et al, *Single-Epoch Integer Ambiguity Resolution with GPS L1-L2 Carrier Phase Measurements*, Proceedings of ION GPS-97, http://satnav.atc.ll.mit.edu/papers/ singleepotchGPS/sept97ion.html.

[118] Sutton, E., *Integer Cycle Ambiguity Resolution under Conditions of Low Satellite Visibility***,** 2002 IEEE Position Location and Navigation Symposium, April 2002, 91 - 98

[119] Thales Avionics Technical Report, *GNSS Attitude Design Report*, Ref. No. NAV/01/004763-00, July, 2001,

[120] Harris, R.L., *Modular Avionics: Its Impacts on Communication, Navigation, and Identification (CNI)*, in the Proceedings of IEEE NAECON, 1988, 1164-1169.

[121] Swanson. D.L., *Evolving Avionics Systems from Federated to Distributed Architectures*, in the Proceedings of the IEEE/AIAA/NASA 17th Digital Avionics Systems Conference, 1998, D26/1-D26/8

[122] Spitzer, C.R, *The Avionics Handbook*, CRC Press LLC, 2001

[123] Harrison, J.V., and Gai, E.G., *Evaluation of Sensor Orientations for Navigation Performance and Failure Detection*, IEEE Trans. on Aerospace and Electronics Systems, Vol. AES-13, No. 6, Nov. 1977.

[124] Hass, R.A. JR.; Kasper, J.F., JR; Crawford, B.S. and Levine, S.A., *Application of Optimal Smoothing to the Testing and Evaluation of Inertial Navigation Systems and Components*, IEEE Trans. On Automatic Control, Vol. AC-16, No. 6, Dec. 1971, 806-816.

[125] White, J.E. and Speyer, J.L., *Detection Filter Design: Spectral Theory and Algorithms*, IEEE Trans. On Automatic Control, Vol.AC-32, No. 7, July 1987, 593-603

[126] Piercy, N.P., Sensor Failure Estimators for Detection Filters, IEEE Trans. On Automatic Control, Vol. 37, No. 10, Oct., 1992, 1553-1558

[127] Douglas R.K. and Speyer, J.L., *Robust Fault Detection Filter Design*, Proceedings of the American Control Conference, Seattle, Washington, USA,

June 1995, 91-96

[128] Chen, R.H. and Speyer, J.L., *Optimal Stochastic Fault Detection Filter*, Proceedings of the American Control Conference, San Diego, CA, USA, June 1999, 91-96

[129] Mallory, G.J.W. and Miller, D.W., *Decentralized State Estimation for Flexible Space Structure*, Journal of Guidance, Control, and Dynamics, Vol. 23, No. 4, Jul-Aug 2000, 665-672.

APPENDIX A

Table A-1 WGS-84 Parameters

Parameters	Notation	Value
Semi-major axis	$\mathrm{R_a}$	$6378137\,m$
Normalized	C20	-484.16685×10^{-6}
Flattening (ellipticity)	f	$1./298.257223563$
Semi-minor axis	R_b	$R_b = R_a(1-f)$ $= 6356752.3142\,m$
Eccentricity squared	e^2	$e^2 = f(2-f)$ $= 6.69437999013\times10^{-3}$
Angular velocity of the Earth	Ω	7.292115×10^{-5} rad/s
The Earth's gravitational constant	GM	$3986004.418\times10^{8}\ \mathrm{m^3/s^2}$
Gravity at the equator	g_0	$9.780373\,\mathrm{m/s^2}$
Gravity formula constant	g_1	0.00193185138639
Velocity of light	C	$299792458\,m/s$

Table A-2 Inertial Sensor Performance Parameters

	Performance Parameters	Unit	Performance Requirements		
			Aided IRS (Control)	AHRS (Tactical Grade)	INS (Navigation Grade)
Gyroscope	Bias uncertainty	$^{0}/h$	10-40	1-10	0.005-0.01
	Scale factor stability	ppm	100-500	100-500	5-50
	Misalignment	arcsec	200	200	10
	Random noise	$^{0}/h/\sqrt{Hz}$	1-5	0.2-0.5	0.002-0.005
Accelerometer	Bias uncertainty	μg	2000	200-500	10-50
	Scale factor stability	ppm	500-1000	500-1000	200
	Misalignment	arcsec	200	200	10
	Random noise	$\mu g/\sqrt{Hz}$	200-400	200-400	5-10

Table A-3 Performance and Trends for MEMS-based Inertial Sensors

Sensor/Performance		Current State	Trends
Gyros	Bias ($^{0}/h$)	100-200	1-10
	Scale factor (ppm)	500	100-200
	Noise floor ($^{0}/h/\sqrt{Hz}$)	10-60	1-10
Accelerometers	Bias (μg)	500-1000	100-300
	Scale factor (ppm)	50-100	10-30
	Random noise ($\mu g/\sqrt{Hz}$)	100-200	10-100

APPENDIX B

The magnitude of the gravity at the surface of the WGS-84 ellipsoid can be approximated by the following equation[105]:

$$g_{\mathrm{WGS\text{-}84}} = g_0 \left[\frac{1 + g_1 \sin(\varphi)}{\sqrt{1 - e^2 \sin(\varphi)}} \right] \tag{B.1}$$

and the variation of the gravity with aircraft altitude can be approximated by

$$g = g_{\mathrm{WGS\text{-}84}} \left(\frac{R_a}{R_a + h} \right)^2 \tag{B.2}$$

In the wander frame, the direction of the gravity points downward along the axis z^w. Therefore, the gravity vector can be represented as

$$\mathbf{g}^w = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{B.3}$$

Given an initial velocity, Eq. (4.16) can be integrated to obtain the current aircraft velocity in terms of the wander coordinates, which can be transformed into the NED frame by.

$$\mathbf{v}^n \equiv \begin{bmatrix} v_n \\ v_e \\ v_d \end{bmatrix} = \mathbf{C}_w^n \mathbf{v}^w = \mathbf{C}_w^n \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{B.4}$$

In the wander angle mechanization, the vertical component of the transport rate $\omega_{e/w,z}^w$ is defined as zero $\omega_{e/w,z}^w \equiv 0$, and the horizontal components are computed as follows:

$$\begin{bmatrix} \omega^{w}_{e/w,x} \\ \omega^{w}_{e/w,y} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \rho_n \\ \rho_e \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{R_e + h} \\ -\frac{1}{R_n + h} & 0 \end{bmatrix} \begin{bmatrix} v_n \\ v_e \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{R_e + h} \\ -\frac{1}{R_n + h} & 0 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\begin{bmatrix} \omega^{w}_{e/w,x} \\ \omega^{w}_{e/w,y} \end{bmatrix} = \begin{bmatrix} (\frac{1}{R_n + h} - \frac{1}{R_e + h}) \sin(\alpha)\cos(\alpha) & \frac{\cos^2(\alpha)}{R_e + h} + \frac{\sin^2(\alpha)}{R_n + h} \\ -(\frac{\sin^2(\alpha)}{R_e + h} + \frac{\cos^2(\alpha)}{R_n + h}) & -(\frac{1}{R_n + h} - \frac{1}{R_e + h}) \sin(\alpha)\cos(\alpha) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{B.5}$$

where R_n and R_e are the radius of curvature along the lines of constant longitude and latitude, respectively.

$$R_n = R_a * (1 - e^2) / (1 - e^2 \sin^2(\varphi))^{\frac{3}{2}}, \quad R_e = R_a / (1 - e^2 \sin^2(\varphi))^{\frac{1}{2}}$$

APPENDIX C

Quaternion Differential Equation

Given an initial attitude DCM, the integration of Eq. (4.19) gives the current attitude DCM $\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$. From Eq. (4.5), the Euler angles can be computed as follows:

$$\theta = \sin^{-1}[-\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}(3,1)],\ \ \phi = \tan^{\text{-}1}\left[\frac{\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}(3,2)}{\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}(3,3)}\right],\ \psi_{\mathrm{w}} = \tan^{-1}\left[\frac{\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}(2,1)}{\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}(1,1)}\right] \qquad \text{(C.1)}$$

From Figure 4.2, the true heading angle is computed as

$$\psi = \psi_{\mathrm{w}} - \alpha \qquad \text{(C.2)}$$

To simplify the computation of the attitude DCM differential equation, the quaternion form of the attitude matrix differential equation is generally used and given as follows:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_{\mathrm{w/b,x}}^{\mathrm{b}} & -\omega_{\mathrm{w//b,y}}^{\mathrm{b}} & -\omega_{\mathrm{w/b,z}}^{\mathrm{b}} \\ \omega_{\mathrm{w/b,,x}}^{\mathrm{b}} & 0 & \omega_{\mathrm{w/b,z}}^{\mathrm{b}} & -\omega_{\mathrm{w/b,y}}^{\mathrm{b}} \\ \omega_{\mathrm{w/b,y}}^{\mathrm{b}} & -\omega_{\mathrm{w/b,z}}^{\mathrm{b}} & 0 & \omega_{\mathrm{w/b,x}}^{\mathrm{b}} \\ \omega_{\mathrm{w/b,z}}^{\mathrm{b}} & \omega_{\mathrm{w/b,y}}^{\mathrm{b}} & -\omega_{\mathrm{w/b,x}}^{b} & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \qquad \text{(C.3)}$$

The relationship between the quaternion and the attitude matrix is represented as:

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

The attitude angles can be computed from the following identities:

$$\begin{aligned} \sin(\theta) &= 2(q_1 q_3 - q_0 q_2) \\ \tan(\psi_{\mathrm{w}}) &= \frac{2(q_1 q_2 + q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \\ \tan(\phi) &= \frac{2(q_2 q_3 + q_0 q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \end{aligned} \qquad \text{(C.4)}$$

APPENDIX D

Linear Position Error Equations

The angular position error $\partial\theta$ is defined in terms of the computed position DCM $\tilde{\mathbf{C}}_{e}^{w}$ and the true position DCM $\mathbf{C}_{e}^{w}$ as follows:

$$\tilde{\mathbf{C}}_{e}^{w} = [\mathbf{I} - (\partial\theta\times)]\mathbf{C}_{e}^{w} \tag{D.1}$$

It can be rewritten as

$$\delta\mathbf{C}_{e}^{w} = \tilde{\mathbf{C}}_{e}^{w} - \mathbf{C}_{e}^{w} \tag{D.2}$$

$$\delta\mathbf{C}_{e}^{w} = -(\partial\theta\times)\mathbf{C}_{e}^{w} \tag{D.3}$$

Differentiating Eqs. (D.2) and (D.3) yields

$$\delta\dot{\mathbf{C}}_{e}^{w} = -(\partial\dot{\theta}\times)\mathbf{C}_{e}^{w} - (\partial\theta\times)\dot{\mathbf{C}}_{e}^{w} \tag{D.4}$$

$$\delta\dot{\mathbf{C}}_{e}^{w} = \dot{\tilde{\mathbf{C}}}_{e}^{w} - \dot{\mathbf{C}}_{e}^{w} = -(\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times)\tilde{\mathbf{C}}_{e}^{w} + (\boldsymbol{\omega}_{e/w}^{w}\times)\mathbf{C}_{e}^{w} \tag{D.5}$$

Substituting Eq. (4.29) into Eq. (D.4), and Eq. (D.1) into Eq. (D.5) yields

$$\delta\dot{\mathbf{C}}_{e}^{w} = -[(\partial\dot{\theta}\times) - (\partial\theta\times)(\boldsymbol{\omega}_{e/w}^{w}\times)]\mathbf{C}_{e}^{w} \tag{D.6}$$

$$\delta\dot{\mathbf{C}}_{e}^{w} = -[(\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times) - (\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times)(\partial\theta\times) - (\boldsymbol{\omega}_{e/w}^{w}\times)]\mathbf{C}_{e}^{w} \tag{D.7}$$

Equating Eqs. (D.6) and (D.7)

$$(\partial\dot{\theta}\times) = (\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times) - (\boldsymbol{\omega}_{e/w}^{w}\times) + (\partial\theta\times)(\boldsymbol{\omega}_{e/w}^{w}\times) - (\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times)(\partial\theta\times)$$

$$(\partial\dot{\theta}\times) = (\delta\boldsymbol{\omega}_{e/w}^{w}\times) + (\partial\theta\times)(\boldsymbol{\omega}_{e/w}^{w}\times) - (\tilde{\boldsymbol{\omega}}_{e/w}^{w}\times)(\partial\theta\times) \tag{D.8}$$

Assuming $\boldsymbol{\omega}_{e/w}^{w} \approx \tilde{\boldsymbol{\omega}}_{e/w}^{w}$, the vector form equivalent of Eq. (D.8) can be written as

$$\partial\dot{\theta} = \delta\boldsymbol{\omega}_{e/w}^{w} - (\boldsymbol{\omega}_{e/w}^{w}\times)\partial\theta \tag{D.9}$$

Eq.(D.9) is known as the angular position error equation.

In order to correct the errors of the geodetic location (latitude and longitude), it is necessary to determine the relationship between the angular position error $\partial\theta$ and the location errors. Let the latitude, longitude and wander angles be expressed in terms of their true values plus error terms as follows:

$$\tilde{\lambda} = \lambda + \delta\lambda, \qquad \tilde{\varphi} = \varphi + \delta\varphi, \qquad \alpha = \alpha + \delta\alpha \tag{D.10}$$

Substituting Eq.(D.10) into Eq. (4.4) and expanding elements of the resultant position DCM matrix $\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}$ produces

$$\begin{aligned}\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(3,3) &= -\sin(\tilde{\varphi}) = -\sin(\varphi+\delta\varphi)\\ &\approx -\sin(\varphi) - \cos(\varphi)\delta\varphi \\ &= \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(3,3) - \cos(\varphi)\delta\varphi\end{aligned} \tag{D.11a}$$

$$\begin{aligned}\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(3,2) &= -\cos(\tilde{\varphi})\sin(\tilde{\lambda}) = -\cos(\varphi+\delta\varphi)\sin(\lambda+\delta\lambda)\\ &\approx -\cos(\varphi)\sin(\lambda) + \sin(\varphi)\sin(\lambda)\delta\varphi - \cos(\varphi)\cos(\lambda)\delta\lambda \\ &= \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(3,2) + \sin(\varphi)\sin(\lambda)\delta\varphi - \cos(\varphi)\cos(\lambda)\delta\lambda\end{aligned} \tag{D.11b}$$

$$\begin{aligned}\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(1,3) &= \cos(\tilde{\alpha})\cos(\tilde{\varphi}) = \cos(\alpha+\delta\alpha)\cos(\varphi+\delta\varphi)\\ &\approx \cos(\alpha)\cos(\varphi) - \cos(\alpha)\sin(\varphi)\delta\varphi - \sin(\alpha)\cos(\varphi)\delta\alpha \\ &= \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(1,3) - \cos(\alpha)\sin(\varphi)\delta\varphi - \sin(\alpha)\cos(\varphi)\delta\alpha\end{aligned} \tag{D.11c}$$

Expanding the right side of Eq. (D.1), the elements corresponding to the DCM can be obtained as follows:

$$\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(3,3) = \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(3,3) + \cos(\alpha)\cos(\varphi)\partial\theta_{\mathrm{y}} - \sin(\alpha)\cos(\varphi)\partial\theta_{\mathrm{x}} \tag{D.12a}$$

$$\begin{aligned}\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(3,2) = \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(3,2) &- [\sin(\alpha)\cos(\lambda) + \cos(\alpha)\sin(\varphi)\sin(\lambda)]\partial\theta_{\mathrm{y}}\\ &- [\cos(\alpha)\cos(\lambda) - \sin(\alpha)\sin(\varphi)\sin(\lambda)]\partial\theta_{\mathrm{x}}\end{aligned} \tag{D.12b}$$

$$\tilde{\mathbf{C}}_{\mathrm{e}}^{\mathrm{w}}(1,3) = \mathbf{C}_{\mathrm{e}}^{\mathrm{w}}(1,3) + \sin(\alpha)\cos(\varphi)\partial\theta_{\mathrm{z}} + \sin(\varphi)\partial\theta_{\mathrm{y}} \tag{D.12c}$$

Equating Eqs. (D.11a) and (D.12a), Eqs. (D.11b) and (D.12b), and Eqs.(D.11c) and (D.12c), respectively, the following equations relate the angular position errors to the latitude, longitude and wander angle errors.

$$\delta\alpha = -\sin(\varphi)\delta\lambda - \partial\theta_{\mathrm{z}} \tag{D.13a}$$

$$\delta\varphi = \sin(\alpha)\partial\theta_{\mathrm{x}} - \cos(\alpha)\partial\theta_{\mathrm{y}} \tag{D.13b}$$

$$\delta\lambda = \frac{1}{\cos(\varphi)}[\cos(\alpha)\partial\theta_{\mathrm{x}} + \sin(\alpha)\partial\theta_{\mathrm{y}}] \tag{D.13c}$$

In vector form, these equations can be written as

$$\begin{bmatrix}\delta\varphi\\ \delta\lambda\\ \delta\alpha\end{bmatrix} = \mathbf{T}_{\mathrm{ang\_err}}^{\mathrm{geo\_err}}\begin{bmatrix}\partial\theta_{\mathrm{x}}\\ \partial\theta_{\mathrm{y}}\\ \partial\theta_{\mathrm{z}}\end{bmatrix} \tag{D.14}$$

where $\mathbf{T}_{\mathrm{ang\_err}}^{\mathrm{geo\_err}}$ is the transformation matrix from the angular position errors to the geodetic location errors.

$$\mathbf{T}_{\mathrm{ang\_err}}^{\mathrm{geo\_err}} \equiv \begin{bmatrix} \sin(\alpha) & -\cos(\alpha) & 0 \\ \dfrac{\cos(\alpha)}{\cos(\varphi)} & \dfrac{\sin(\alpha)}{\cos(\varphi)} & 0 \\ -\tan(\varphi)\cos(\alpha) & -\tan(\varphi)\sin(\alpha) & -1 \end{bmatrix} \tag{D.15}$$

In the wander mechanisation, the vertical component of angular position error $\partial\theta_{\mathrm{z}}$ can be assumed to be zero.

However, many aiding navigation systems use the linear position errors as the error states to develop the measurement equations. It is necessary to derive the linear position error equations of inertial system.

The linear position errors are defined as

$$\delta\varphi_{\mathrm{R}} = (\mathrm{R_n} + \mathrm{h})\delta\varphi \approx (\mathrm{R_a} + \mathrm{h})\delta\varphi \tag{D.16}$$

$$\delta\lambda_{\mathrm{R}} = (\mathrm{R_e} + \mathrm{h})\cos(\varphi)\delta\lambda \approx (\mathrm{R_a} + \mathrm{h})\cos(\varphi)\delta\lambda \tag{D.17}$$

and the relationship between the angular position errors and the linear position errors is given by

$$\delta\alpha = -\frac{\tan(\varphi)}{\mathrm{R_e} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \approx -\frac{\tan(\varphi)}{\mathrm{R_a} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \tag{D.18a}$$

$$\partial\theta_{\mathrm{x}} = \frac{\sin(\alpha)}{\mathrm{R_n} + \mathrm{h}}\delta\varphi_{\mathrm{R}} + \frac{\cos(\alpha)}{\mathrm{R_e} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \approx \frac{\sin(\alpha)}{\mathrm{R_a} + \mathrm{h}}\delta\varphi_{\mathrm{R}} + \frac{\cos(\alpha)}{\mathrm{R_a} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \tag{D.18b}$$

$$\partial\theta_{\mathrm{y}} = -\frac{\cos(\alpha)}{\mathrm{R_n} + \mathrm{h}}\delta\varphi_{\mathrm{R}} + \frac{\sin(\alpha)}{\mathrm{R_e} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \approx -\frac{\cos(\alpha)}{\mathrm{R_a} + \mathrm{h}}\delta\varphi_{\mathrm{R}} + \frac{\sin(\alpha)}{\mathrm{R_a} + \mathrm{h}}\delta\lambda_{\mathrm{R}} \tag{D.18c}$$

Expanding both sides of Eq. (4.29) and assuming $\omega_{\mathrm{e/w,z}}^{\mathrm{w}} = 0$, the geodetic location differential equations are derived as follows:

$$\dot{\varphi} = \sin(\alpha)\omega_{\mathrm{e/w}}^{\mathrm{w}} - \cos(\alpha)\omega_{\mathrm{e/w}}^{\mathrm{w}} \tag{D.19}$$

$$\dot{\lambda} = \frac{1}{\cos(\varphi)}[\cos(\alpha)\omega_{\mathrm{e/w}}^{\mathrm{w}} + \sin(\alpha)\omega_{\mathrm{e/w}}^{\mathrm{w}}] \tag{D.20}$$

Replacing $(\mathrm{R_n} + \mathrm{h})$ and $(\mathrm{R_e} + \mathrm{h})$ in Eq. (4.21) by $(\mathrm{R_a} + \mathrm{h})$ and then substituting the resultant Eq. (4.21) into Eqs. (D.19) and (D.20), the differential equations of the linear position errors can be derived as follows:

$$\begin{aligned}\delta\dot{\varphi}_R &= \frac{v_z}{R_a + h}\delta\varphi_R \\ &\quad - \frac{1}{R_a + h}[\cos(\alpha)v_x + \sin(\alpha)v_y]\delta h + \cos(\alpha)\delta v_x^1 + \sin(\alpha)\delta v_y^1\end{aligned} \tag{D.21}$$

$$\begin{aligned}\delta\dot{\lambda}_R &= \frac{v_z}{R_a + h}\delta\lambda_R + \frac{\tan(\varphi)}{R_a + h}[-\sin(\alpha)v_x + \cos(\alpha)v_y]\delta\varphi_R \\ &\quad - \frac{1}{R_a + h}[-\sin(\alpha)v_x + \cos(\alpha)v_y]\delta h - \sin(\alpha)\delta v_x^1 + \cos(\alpha)\delta v_y^1\end{aligned} \tag{D.22}$$

From Eq. (4.39a), the rate of change of height is defined as

$$\delta\dot{h} \equiv -(\tilde{v}_z - v_z) = -\partial\theta_y v_x + \partial\theta_x v_y - \delta v_z^1$$

Substituting Eqs. (D.18b, c) into the above equation yields

$$\begin{aligned}\delta\dot{h} &= (\cos(\alpha)v_x + \sin(\alpha)v_y)\frac{1}{R_n + h}\delta\varphi_R \\ &\quad + (-\sin(\alpha)v_x + \cos(\alpha)v_y)\frac{1}{R_e + h}\delta\lambda_R - \delta v_z^1\end{aligned} \tag{D.23}$$

In vector form, the linear position error differential equation can be written as

$$\begin{bmatrix}\delta\dot{\varphi}_R \\ \delta\dot{\lambda}_R \\ \delta\dot{h}\end{bmatrix} = \begin{bmatrix}\cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}\delta v_x^1 \\ \delta v_y^1 \\ \delta v_z^1\end{bmatrix} +$$

$$\begin{bmatrix}\dfrac{v_z}{R_a + h} & 0 & -\dfrac{v_x\cos(\alpha) + v_y\sin(\alpha)}{R_a + h} \\ \dfrac{-v_x\sin(\alpha) + v_y\cos(\alpha)}{R_a + h}\tan(\varphi) & \dfrac{v_z}{R_a + h} & -\dfrac{-v_x\sin(\alpha) + v_y\cos(\alpha)}{R_a + h} \\ \dfrac{v_x\cos(\alpha) + v_y\sin(\alpha)}{R_a + h} & \dfrac{-v_x\sin(\alpha) + v_y\cos(\alpha)}{R_a + h} & 0\end{bmatrix}\begin{bmatrix}\delta\varphi_R \\ \delta\lambda_R \\ \delta h\end{bmatrix}$$

From Eqs. (4.30c), (B.2) and (B.3)

$$\delta g^w \approx \begin{bmatrix}0 \\ 0 \\ -\dfrac{2g_0\delta h}{R_a + h}\end{bmatrix}$$

and from Eqs.(D.18b, c)

$$-(\mathbf{f}^{\mathrm{w}}+\mathbf{g}^{\mathrm{w}})\times\delta\theta=\begin{bmatrix} -\dfrac{\mathrm{f}_{\mathrm{z}}^{\mathrm{w}}+g_0}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha) & \dfrac{\mathrm{f}_{\mathrm{z}}^{\mathrm{w}}+g_0}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha) & 0 \\ -\dfrac{\mathrm{f}_{\mathrm{z}}^{\mathrm{w}}+g_0}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\sin(\alpha) & -\dfrac{\mathrm{f}_{\mathrm{z}}^{\mathrm{w}}+g_0}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}\cos(\alpha) & 0 \\ \dfrac{\sin(\alpha)\mathrm{f}_{\mathrm{y}}^{\mathrm{w}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}+\dfrac{\cos(\alpha)\mathrm{f}_{\mathrm{x}}^{\mathrm{w}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}} & \dfrac{\cos(\alpha)\mathrm{f}_{\mathrm{y}}^{\mathrm{w}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}}-\dfrac{\sin(\alpha)\mathrm{f}_{\mathrm{x}}^{\mathrm{w}}}{\mathrm{R}_{\mathrm{a}}+\mathrm{h}} & 0 \end{bmatrix}\begin{bmatrix}\delta\varphi_{\mathrm{R}}\\ \delta\lambda_{\mathrm{R}}\\ \delta\mathrm{h}\end{bmatrix}$$

APPENDIX E

Tilt Error Differential Equations

This computed attitude DCM $\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}}$ can be represented in terms of the true attitude DCM $\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ as follows:

$$\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = [\mathbf{I} - (\partial\phi\times)]\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{E.1}$$

where $\partial\phi$ is known as the tilt error vector.

Let $\delta\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} = \tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} - \mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ (E.2a)

then $\delta\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} = -(\partial\phi\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}$ (E.2b)

Differentiating Eqs. (E.2a) and (E.2b) yields

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -(\partial\dot{\phi}\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} - (\partial\phi\times)\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} \tag{E.3a}$$

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = \dot{\tilde{\mathbf{C}}}_{\mathrm{b}}^{\mathrm{w}} - \dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -(\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{\mathrm{w}}\times)\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} + (\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{E.3b}$$

Substituting Eq. (4.17) into Eq. (E.3a) results in

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -[(\partial\dot{\phi}\times) - (\partial\phi\times)(\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times)]\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{E.4a}$$

and substituting Eq. (E.1) into Eq. (E.3b) yields

$$\delta\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} = -(\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{\mathrm{w}}\times)[\mathbf{I} - (\partial\phi\times)]\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} + (\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}} \tag{E.4b}$$

From Eqs. (E.4a) and (E.4b)

$$(\partial\dot{\phi}\times) = (\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{\mathrm{w}}\times) - (\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times) + (\partial\phi\times)(\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}\times) - (\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{\mathrm{w}}\times)(\partial\phi\times) \tag{E.5}$$

In vector form, the above matrix equation can be equivalently represented as

$$\partial\dot{\phi} = (\partial\phi\times)\boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}} + (\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{w} - \boldsymbol{\omega}_{\mathrm{b/w}}^{\mathrm{w}}) \tag{E.6}$$

Expanding the second term of the right side of Eq. (E.6)

$$\begin{aligned}
\tilde{\boldsymbol{\omega}}_{\mathrm{b/w}}^{w} - \boldsymbol{\omega}_{\mathrm{b/w}}^{w} &= \tilde{\boldsymbol{\omega}}_{\mathrm{I/w}}^{w} - \tilde{\boldsymbol{\omega}}_{\mathrm{I/b}}^{\mathrm{b}} - \boldsymbol{\omega}_{\mathrm{I/w}}^{\mathrm{w}} + \mathbf{C}_{\mathrm{b}}^{\mathrm{w}}\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{b}} \\
&= \tilde{\boldsymbol{\omega}}_{\mathrm{I/w}}^{w} - \tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}}\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{b}} - \tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}}\Delta^{\mathrm{b}} - \boldsymbol{\omega}_{\mathrm{I/w}}^{\mathrm{w}} + \mathbf{C}_{\mathrm{b}}^{\mathrm{w}}\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{b}} \\
&= \tilde{\boldsymbol{\omega}}_{\mathrm{I/w}}^{\mathrm{w}} - \boldsymbol{\omega}_{\mathrm{I/w}}^{\mathrm{w}} - (\tilde{\mathbf{C}}_{\mathrm{b}}^{\mathrm{w}} - \mathbf{C}_{\mathrm{b}}^{\mathrm{w}})\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{b}} - \Delta^{\mathrm{w}} \qquad \text{(E.7)} \\
&= \tilde{\boldsymbol{\omega}}_{\mathrm{e/w}}^{\mathrm{w}} - \boldsymbol{\omega}_{\mathrm{e/w}}^{\mathrm{w}} + \tilde{\boldsymbol{\omega}}_{\mathrm{I/e}}^{\mathrm{w}} - \boldsymbol{\omega}_{\mathrm{I/e}}^{\mathrm{w}} + (\partial\phi\times)\mathbf{C}_{\mathrm{b}}^{\mathrm{w}}\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{b}} - \Delta^{\mathrm{w}} \\
&= \delta\boldsymbol{\omega}_{\mathrm{e/w}}^{\mathrm{w}} + \delta\boldsymbol{\omega}_{\mathrm{I/e}}^{w} + (\partial\phi\times)\boldsymbol{\omega}_{\mathrm{I/b}}^{\mathrm{w}} - \Delta^{\mathrm{w}}
\end{aligned}$$

where

$$\begin{aligned}\delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} &= \tilde{\boldsymbol{\omega}}^{\mathrm{w}}_{\mathrm{I/e}} - \boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} = \tilde{\mathbf{C}}^{\mathrm{w}}_{\mathrm{e}}\boldsymbol{\omega}^{\mathrm{e}}_{\mathrm{I/e}} - \boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} = [\mathbf{I} - (\partial\theta\times)]\mathbf{C}^{\mathrm{w}}_{\mathrm{e}}\boldsymbol{\omega}^{\mathrm{e}}_{\mathrm{I/e}} - \boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} \\ &= -(\partial\theta\times)\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} = (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times)\partial\theta\end{aligned} \qquad \text{(E.8)}$$

Note that $\tilde{\mathbf{C}}^{\mathrm{w}}_{\mathrm{e}} = [\mathbf{I} - \partial\theta\times]\mathbf{C}^{\mathrm{w}}_{\mathrm{e}}$ in the above equation.

Substituting Eq.(E.8) into Eq.(E.7) and the resultant Eq. (E.7) into Eq.(E.6) leads to

$$\begin{aligned}\partial\dot{\phi} &= \delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}} + (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{i/e}}\times)\partial\theta + (\partial\phi\times)\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{b/w}} + (\partial\phi\times)\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/b}} - \Delta^{\mathrm{w}} \\ &= \delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}} + (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times)\partial\theta + (\partial\phi\times)\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/w}} - \Delta^{\mathrm{w}} \\ &= \delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}} + (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times)\partial\theta + (\partial\phi\times)(\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}} + \boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}}) - \Delta^{\mathrm{w}}\end{aligned}$$

Therefore,

$$\partial\dot{\phi} = -[(\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times) + (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}}\times)]\partial\phi + \delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}} + (\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times)\partial\theta - \Delta^{\mathrm{w}} \qquad \text{(E.9)}$$

where

$$\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{I/e}}\times\partial\theta = \begin{bmatrix} \dfrac{\omega^{\mathrm{w}}_{\mathrm{I/e,z}}\cos(\alpha)}{\mathrm{R_a + h}} & -\dfrac{\omega^{\mathrm{w}}_{\mathrm{I/e,z}}\sin(\alpha)}{\mathrm{R_a + h}} & 0 \\ \dfrac{\omega^{\mathrm{w}}_{\mathrm{I/e,z}}\sin(\alpha)}{\mathrm{R_a + h}} & \dfrac{\omega^{\mathrm{w}}_{\mathrm{I/e,z}}\cos(\alpha)}{\mathrm{R_a + h}} & 0 \\ -\dfrac{\sin(\alpha)\omega^{\mathrm{w}}_{\mathrm{I/e,y}}}{\mathrm{R_a + h}} - \dfrac{\cos(\alpha)\omega^{\mathrm{w}}_{\mathrm{I/e,x}}}{\mathrm{R_a + h}} & -\dfrac{\cos(\alpha)\omega^{\mathrm{w}}_{\mathrm{I/e,y}}}{\mathrm{R_a + h}} + \dfrac{\sin(\alpha)\omega^{\mathrm{w}}_{\mathrm{I/e,x}}}{\mathrm{R_a + h}} & 0 \end{bmatrix} \begin{bmatrix} \delta\varphi_{\mathrm{R}} \\ \delta\lambda_{\mathrm{R}} \\ \delta\mathrm{h} \end{bmatrix}$$

$$\delta\boldsymbol{\omega}^{\mathrm{w}}_{\mathrm{e/w}} = \begin{bmatrix} 0 & \dfrac{1}{\mathrm{R_a + h}} & 0 \\ -\dfrac{1}{\mathrm{R_a + h}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\mathrm{v}^{1}_{\mathrm{x}} \\ \delta\mathrm{v}^{1}_{\mathrm{y}} \\ \delta\mathrm{v}^{1}_{\mathrm{z}} \end{bmatrix} + \begin{bmatrix} \dfrac{\sin(\alpha)\mathrm{v_z}}{(\mathrm{R_a + h})^2} & \dfrac{\cos(\alpha)\mathrm{v_z}}{(\mathrm{R_a + h})^2} & -\dfrac{\mathrm{v_y}}{(\mathrm{R_a + h})^2} \\ -\dfrac{\cos(\alpha)\mathrm{v_z}}{(\mathrm{R_a + h})^2} & \dfrac{\sin(\alpha)\mathrm{v_z}}{(\mathrm{R_a + h})^2} & \dfrac{\mathrm{v_x}}{(\mathrm{R_a + h})^2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\varphi_{\mathrm{R}} \\ \delta\lambda_{\mathrm{R}} \\ \delta h \end{bmatrix}$$

In order to correct the attitude angles, the relationship between the tilt errors and the Euler angle errors has to be determined. Let the computed Euler angles be expressed in terms of their true values plus error terms as follows:

$$\tilde{\phi} = \phi + \delta\phi,\ \tilde{\theta} = \theta + \delta\theta,\ \tilde{\psi}_{\mathrm{w}} = \psi_{\mathrm{w}} + \delta\psi_{\mathrm{w}} \qquad \text{(E.10)}$$

Substituting Eq. (E.10) into the left side of Eq. (E.1) and expanding both sides of the resultant equation, the following equations relate the tilt errors to the Euler errors.

$$\delta\phi = -\frac{\cos(\psi_{\mathrm{w}})\partial\phi_{\mathrm{x}} + \sin(\psi_{\mathrm{w}})\partial\phi_{\mathrm{y}}}{\cos(\theta)}$$

$$\delta\theta = \sin(\psi_{\mathrm{w}})\partial\phi_{\mathrm{x}} - \cos(\psi_{\mathrm{w}})\partial\phi_{\mathrm{y}}$$

$$\delta\psi_{\mathrm{w}} = -\tan(\theta)[\cos(\psi_{\mathrm{w}})\partial\phi_{\mathrm{x}} + \sin(\psi_{\mathrm{w}})\partial\phi_{\mathrm{y}}] - \partial\phi_{\mathrm{z}}$$

In vector form, they can be rewritten as

$$\begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi_{\mathrm{w}} \end{bmatrix} = \mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}} \begin{bmatrix} \partial\phi_{\mathrm{x}} \\ \partial\phi_{\mathrm{y}} \\ \partial\phi_{\mathrm{z}} \end{bmatrix} \tag{E.11}$$

where $\mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}}$ is the transformation matrix from the tilt errors to the Euler errors, given by

$$\mathbf{T}_{\mathrm{tilt}}^{\mathrm{Euler\_err}} = \begin{bmatrix} -\dfrac{\cos(\psi_{\mathrm{w}})}{\cos(\theta)} & -\dfrac{\sin(\psi_{\mathrm{w}})}{\cos(\theta)} & 0 \\ \sin(\psi_{\mathrm{w}}) & -\cos(\psi_{\mathrm{w}}) & 0 \\ -\tan(\theta)\cos(\psi_{\mathrm{w}}) & -\tan(\theta)\sin(\psi_{\mathrm{w}}) & -1 \end{bmatrix} \tag{E.12}$$

APPENDIX F

GNSS Navigation Equations

Given these nominal points at the time t_k, $[\mathrm{x}_0, \mathrm{y}_0, \mathrm{z}_0]^\mathrm{T}$ and $[\dot{\mathrm{x}}_0, \dot{\mathrm{y}}_0, \dot{\mathrm{z}}_0]^\mathrm{T}$, Eqs. (4.50) and (4.51) can be linearised as follows:

$$\delta\mathrm{r}^i(t_k) \equiv \mathrm{r}^i - \rho_0^i = \frac{-(\mathrm{x}^i - \mathrm{x}_0)}{\rho_0^i}\delta\mathrm{x} + \frac{-(\mathrm{y}^i - \mathrm{y}_0)}{\rho_0^i}\delta y + \frac{-(\mathrm{z}^i - \mathrm{z}_0)}{\rho_0^i}\delta z + cdt + v_\mathrm{r}^\mathrm{i} \tag{F.1}$$

$$\begin{aligned}\delta\dot{\mathrm{r}}^i(t_k) \equiv \dot{\mathrm{r}} - \dot{\rho}_0^1 = {} & \frac{-(\mathrm{x}^i - \mathrm{x}_0)}{\rho_0^i}\delta\dot{\mathrm{x}} + \frac{-(\mathrm{y}^i - \mathrm{y}_0)}{\rho_0^i}\delta\dot{\mathrm{y}} + \frac{-(\mathrm{z}^i - \mathrm{z}_0)}{\rho_0^i}\delta\dot{\mathrm{z}} + cd\dot{t} + \\ & \frac{-(\dot{\mathrm{x}}^i - \dot{\mathrm{x}}_0)}{\rho_0^i}\delta\mathrm{x} + \frac{-(\dot{\mathrm{y}}^i - \dot{\mathrm{y}}_0)}{\rho_0^i}\delta\mathrm{y} + \frac{-(\dot{\mathrm{z}}^i - \dot{\mathrm{z}}_0)}{\rho_0^i}\delta\mathrm{z} + v_{\dot{\mathrm{r}}}^i\end{aligned} \tag{F.2}$$

where v_r and $v_{\dot{\mathrm{r}}}$ include the additional errors resulting from the linearisation, dt and $d\dot{t}$ are the unknown receiver clock phase and frequency errors, and $[\delta\mathrm{x}, \delta\mathrm{y}, \delta\mathrm{z}]^\mathrm{T}$ and $[\delta\dot{\mathrm{x}}, \delta\dot{\mathrm{y}}, \delta\dot{\mathrm{z}}]^\mathrm{T}$ are the unknown receiver position and velocity error vectors,

$$\rho_0^i(t_k) = \sqrt{(\mathrm{x}^i - \mathrm{x}_0)^2 + (\mathrm{y}^i - \mathrm{y}_0)^2 + (\mathrm{z}^i - \mathrm{z}_0)^2} \tag{F.3}$$

$$\dot{\rho}_0^i(t_k) = \frac{(\mathrm{x}^i - \mathrm{x}_0)}{\rho_0^i}(\dot{\mathrm{x}}^i - \dot{\mathrm{x}}_0) + \frac{(\mathrm{y}^i - \mathrm{y}_0)}{\rho_0^i}(\dot{\mathrm{y}}^i - \dot{\mathrm{y}}_0) + \frac{(\mathrm{z}^i - \mathrm{z}_0)}{\rho_0^i}(\dot{\mathrm{z}}^i - \dot{\mathrm{z}}_0) \tag{F.4}$$

From Eqs. (F.1) and (F.2), at least four GNSS satellites must be observed concurrently in order to resolve three unknown position states, three velocity states and two receiver clock error states. Therefore, when more than four GNSS satellites are visible, the GNSS navigation equations can be rewritten in vector form as follows:

$$\delta\mathbf{r} = \mathbf{H}_\mathrm{GNSS}\delta\mathbf{p}_\mathrm{p} + \boldsymbol{\upsilon}_\mathrm{r} \tag{F.5}$$

$$\delta\dot{\mathbf{r}} = \mathbf{H}_\mathrm{GNSS}\delta\dot{\mathbf{p}}_\mathrm{p} + \mathbf{H}_\mathrm{LOSV}\delta\mathbf{p} + \boldsymbol{\upsilon}_{\dot{\mathrm{r}}} \tag{F.6}$$

where

$$\boldsymbol{\delta p}_{\mathrm{p}}=\begin{bmatrix}\boldsymbol{\delta p}\\ cdt\end{bmatrix},\boldsymbol{\delta p}=\begin{bmatrix}\delta \mathrm{x}\\ \delta \mathrm{y}\\ \delta \mathrm{z}\end{bmatrix},\boldsymbol{\delta r}=\begin{bmatrix}\delta \mathrm{r}^{1}\\ \delta \mathrm{r}^{2}\\ \delta \mathrm{r}^{3}\\ \vdots\\ \delta \mathrm{r}^{m}\end{bmatrix},\ \boldsymbol{\delta \dot{r}}=\begin{bmatrix}\delta \dot{\mathrm{r}}^{1}\\ \delta \dot{\mathrm{r}}^{2}\\ \delta \dot{\mathrm{r}}^{3}\\ \vdots\\ \delta \dot{\mathrm{r}}^{m}\end{bmatrix}\boldsymbol{\upsilon}_{\mathrm{r}}=\begin{bmatrix}v_{\mathrm{r}}^{1}\\ v_{\mathrm{r}}^{2}\\ v_{\mathrm{r}}^{3}\\ \vdots\\ v_{\mathrm{r}}^{m}\end{bmatrix},\ \boldsymbol{\upsilon}_{\dot{\mathrm{r}}}=\begin{bmatrix}v_{\dot{\mathrm{r}}}^{1}\\ v_{\dot{\mathrm{r}}}^{2}\\ v_{\dot{\mathrm{r}}}^{3}\\ \vdots\\ v_{\dot{\mathrm{r}}}^{m}\end{bmatrix}$$

$$\mathbf{H}_{\mathrm{GNSS}}=\begin{bmatrix}\frac{-(\mathrm{x}^{1}-\mathrm{x}_{0})}{\rho_{0}^{1}} & \frac{-(\mathrm{y}^{1}-\mathrm{y}_{0})}{\rho_{0}^{1}} & \frac{-(\mathrm{z}^{1}-\mathrm{z}_{0})}{\rho_{0}^{1}} & 1\\ \frac{-(\mathrm{x}^{2}-\mathrm{x}_{0})}{\rho_{0}^{2}} & \frac{-(\mathrm{y}^{2}-\mathrm{y}_{0})}{\rho_{0}^{2}} & \frac{-(\mathrm{z}^{2}-\mathrm{z}_{0})}{\rho_{0}^{2}} & 1\\ \frac{-(\mathrm{x}^{3}-\mathrm{x}_{0})}{\rho_{0}^{3}} & \frac{-(\mathrm{y}^{3}-\mathrm{y}_{0})}{\rho_{0}^{3}} & \frac{-(\mathrm{z}^{3}-\mathrm{z}_{0})}{\rho_{0}^{3}} & 1\\ \vdots & \vdots & \vdots & \vdots\\ \frac{-(\mathrm{x}^{m}-\mathrm{x}_{0})}{\rho_{0}^{m}} & \frac{-(\mathrm{y}^{m}-\mathrm{y}_{0})}{\rho_{0}^{m}} & \frac{-(\mathrm{z}^{m}-\mathrm{z}_{0})}{\rho_{0}^{m}} & 1\end{bmatrix}=\left[\mathbf{H}_{\mathrm{LOS}}\quad \mathbf{1}\right]$$

$\left[\frac{-(\mathrm{x}^{i}-\mathrm{x}_{0})}{\rho_{0}^{i}},\frac{-(\mathrm{y}^{i}-\mathrm{y}_{0})}{\rho_{0}^{i}},\frac{-(\mathrm{z}^{i}-\mathrm{z}_{0})}{\rho_{0}^{i}}\right]^{\mathrm{T}}=\mathbf{s}^{\mathrm{e}/i}$ is known as the direction cosine or the line of sight (LOS) vector of the satellite i in terms of the ECEF coordinates and $\mathbf{H}_{\mathrm{LOS}}$ is the visible satellite LOS DCM.

$$\mathbf{H}_{\mathrm{LOSV}}=\begin{bmatrix}\frac{-(\dot{\mathrm{x}}^{1}-\dot{\mathrm{x}}_{0})}{\rho_{0}^{1}} & \frac{-(\dot{\mathrm{y}}^{1}-\dot{\mathrm{y}}_{0})}{\rho_{0}^{1}} & \frac{-(\dot{\mathrm{z}}^{1}-\dot{\mathrm{z}}_{0})}{\rho_{0}^{1}}\\ \frac{-(\dot{\mathrm{x}}^{2}-\dot{\mathrm{x}}_{0})}{\rho_{0}^{2}} & \frac{-(\dot{\mathrm{y}}^{2}-\dot{\mathrm{y}}_{0})}{\rho_{0}^{2}} & \frac{-(\dot{\mathrm{z}}^{2}-\dot{\mathrm{z}}_{0})}{\rho_{0}^{2}}\\ \frac{-(\dot{\mathrm{x}}^{3}-\dot{\mathrm{x}}_{0})}{\rho_{0}^{3}} & \frac{-(\dot{\mathrm{y}}^{3}-\dot{\mathrm{y}}_{0})}{\rho_{0}^{3}} & \frac{-(\dot{\mathrm{z}}^{3}-\dot{\mathrm{z}}_{0})}{\rho_{0}^{3}}\\ \vdots & \vdots & \vdots\\ \frac{-(\dot{\mathrm{x}}^{m}-\dot{\mathrm{x}}_{0})}{\rho_{0}^{m}} & \frac{-(\dot{\mathrm{y}}^{m}-\dot{\mathrm{y}}_{0})}{\rho_{0}^{m}} & \frac{-(\dot{\mathrm{z}}^{m}-\dot{\mathrm{z}}_{0})}{\rho_{0}^{m}}\end{bmatrix}$$

APPENDIX G

Doppler Radar Navigation System

Doppler navigation system is an aircraft self-contained dead-reckoning system. A Doppler radar mounted underneath an aircraft can obtain the velocity vector of the aircraft relative to the ground by measuring the Doppler shifts of radar beam signals returned from the ground. A three-beam Janus Doppler radar system is illustrated in Figure G.1 with two forward-looking beams pointing to the right and left sides of airframe, respectively, and one beam looking backward. This radar system can measure three orthogonal components of aircraft velocity[1]. Two types of the Doppler radar mechanisations are used; one fixes the radar antenna array to the aircraft body frame, and one continuously stabilises the radar antenna array to the local horizontal by means of an attitude reference system. The airframe-fixed radar system resolves aircraft velocity in the body coordinates while the attitude stabilised radar system obtains aircraft velocity coordinated in the local horizontal frame.

These two measurements can be formulated in the wander frame as follows

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{DR}} = \tilde{\mathbf{C}}^{\mathrm{w}}_{\mathrm{b}}\tilde{\mathbf{v}}^{\mathrm{b}}_{\mathrm{DR}} \tag{G.1}$$

where $\tilde{\mathbf{v}}^{\mathrm{b}}_{\mathrm{DR}}$ is the Doppler radar output.

Therefore,

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{DR}} = (\mathbf{I} - \partial\phi\times)\mathbf{C}^{\mathrm{w}}_{\mathrm{b}}(\mathbf{v}^{\mathrm{b}}_{\mathrm{DR}} + \boldsymbol{\upsilon}_{\mathrm{DR}})$$
$$\approx \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\mathbf{v}^{\mathrm{b}}_{\mathrm{DR}} - (\partial\phi\times)\mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\mathbf{v}^{\mathrm{b}}_{\mathrm{DR}} + \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\boldsymbol{\upsilon}_{\mathrm{DR}}$$

or

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{DR}} = \mathbf{v}^{\mathrm{w}} + (\mathbf{v}^{\mathrm{w}}\times)\partial\phi + \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\boldsymbol{\upsilon}_{\mathrm{DR}} \tag{G.2}$$

where $\mathbf{v}^{\mathrm{w}} = \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\mathbf{v}^{\mathrm{b}}_{\mathrm{DR}}$ is the true velocity of aircraft and $\boldsymbol{\upsilon}_{\mathrm{DR}}$ is the Doppler radar error vector.

From Eqs. (4.30a) and (G.2), the velocity difference between the INS-derived and Doppler radar-derived velocities can be normalised as follows:

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{INS}} - \tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{DR}} = [\mathbf{I} - (\partial\theta\times)]\mathbf{v}^{\mathrm{w}} + \delta\mathbf{v}^{1} - \mathbf{v}^{\mathrm{w}} - (\mathbf{v}^{\mathrm{w}}\times)\partial\phi - \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\boldsymbol{\upsilon}_{\mathrm{DR}}$$

Therefore,

$$\tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{INS}} - \tilde{\mathbf{v}}^{\mathrm{w}}_{\mathrm{DR}} = \mathbf{T}^{\mathrm{v}}_{\mathrm{p}}\delta\mathbf{p}^{\mathrm{n}} + \delta\mathbf{v}^{1} - (\mathbf{v}^{\mathrm{w}}\times)\partial\phi - \mathbf{C}^{\mathrm{w}}_{\mathrm{b}}\boldsymbol{\upsilon}_{\mathrm{DR}} \tag{G.3}$$

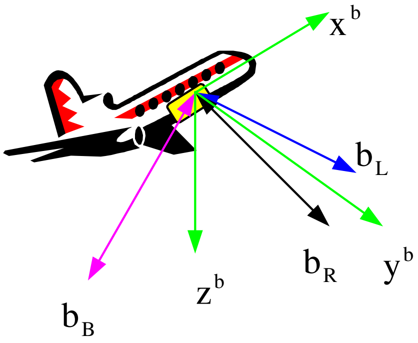


Figure G.1 Three-Beam Janus Doppler Radar Configuration

APPENDIX H

SRIMU Error Equations

Substituting Eqs. (5.17) and (5.18) into Eq. (5.2a) and using the first-order Taylor series expansion for each element of the design matrix $\mathbf{H}$ result in

$$
\begin{aligned}
\mathbf{i} \circ \mathbf{s}^i &= \cos El^i \cos Az^i \\
&= \cos(El_0^i + \delta_{El^i}) \cos(Az_0^i - \delta_{Az^i}) \\
&= \cos El_0^i \cos Az_0^i \\
&\quad - \delta_{El^i} \sin El_0^i \cos Az_0^i + \delta_{Az^i} \cos El_0^i \sin Az_0^i
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{j} \circ \mathbf{s}^i &= \cos El^i \sin Az^i \\
&= \cos(El_0^i + \delta_{El^i}) \sin(Az_0^i - \delta_{Az^i}) \\
&= \cos El_0^i \sin Az_0^i \\
&\quad - \delta_{El^i} \sin El_0^i \sin Az_0^i - \delta_{Az^i} \cos El_0^i \cos Az_0^i
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{k} \circ \mathbf{s}^i &= \sin El^i \\
&= \sin(El_0^i + \delta_{El^i}) \\
&= \sin El_0^i + \delta_{El^i} \cos El_o^i
\end{aligned}
$$

Therefore, $\Delta\mathbf{H}$ can be driven as follows

$$
\Delta\mathbf{H} = \begin{bmatrix}
\begin{array}{l} -\delta_{El^1} \sin El_0^1 \cos Az_0^1 \\ +\delta_{Az^1} \cos El_0^1 \sin Az_0^1 \end{array} & \begin{array}{l} -\delta_{El^1} \sin El_0^1 \sin Az_0^1 \\ -\delta_{Az^1} \cos El_0^1 \cos Az_0^1 \end{array} & \delta_{El^1} \cos El_0^1 \\
\begin{array}{l} -\delta_{El^2} \sin El_0^2 \cos Az_0^2 \\ +\delta_{Az2} \cos El_0^2 \sin Az_0^2 \end{array} & \begin{array}{l} -\delta_{El^2} \sin El_0^2 \sin Az_0^2 \\ -\delta_{Az^2} \cos El_0^2 \cos Az_0^2 \end{array} & \delta_{El^2} \cos El_0^2 \\
\vdots & \vdots & \vdots \\
\begin{array}{l} -\delta_{El^n} \sin El_0^n \cos Az_0^n \\ +\delta_{Az^n} \cos El_0^n \sin Az_0^n \end{array} & \begin{array}{l} -\delta_{El^n} \sin El_0^n \sin Az_0^n \\ -\delta_{Az^n} \cos El_0^n \cos Az_0^n \end{array} & \delta_{El^n} \cos El_0^n
\end{bmatrix}
$$

Observing the form of the above equation, $\Delta\mathbf{H}$ can be further represented as follows:

$$
\Delta\mathbf{H} = \mathbf{Diag}(\boldsymbol{\delta}_{\mathrm{El}})\boldsymbol{\Gamma} + \mathbf{Diag}(\boldsymbol{\delta}_{\mathrm{Az}})\boldsymbol{\Theta} \qquad \text{(H.1)}
$$

where $\mathbf{Diag}()$ is a diagonal matrix consisting of the elements of the elevation misalignment vector $\boldsymbol{\delta}_{\mathrm{El}}$ or the azimuth misalignment vector $\boldsymbol{\delta}_{\mathrm{Az}}$, the elements of the matrix $\boldsymbol{\Gamma}$ correspond to the coefficients of the elevation misalignments in the matrix

$\Delta\mathbf{H}$ and the elements of $\mathbf{\Theta}$ are the coefficients of the azimuth misalignments in $\Delta\mathbf{H}$.

Approximating the measured state vector $\boldsymbol{\omega}$ in Eq. (5.20) by its estimate $\hat{\boldsymbol{\omega}}$ given in Eq. (5.3), a generalised formulation of the measurement errors caused by the SRIMU misalignments can be expressed by

$$\Delta\mathbf{m} = \Delta\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{m} = \mathbf{G}\mathbf{m}$$

Therefore,

$$\mathbf{G} = \Delta\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} = \Delta\mathbf{H}\mathbf{C}_{\mathrm{instru}}^{\mathrm{b}} \tag{H.2}$$

The SRIMU error terms given in Eq. (5.16) are represented in the instrument frames. From Eqs. (5.16), (H.1) and (H.2), the total SRIMU measurement error is

$$\Delta\boldsymbol{\omega}^{\mathrm{instru}} = \boldsymbol{\delta}_{\mathrm{D}} + \mathrm{Diag}(\mathbf{m})\boldsymbol{\delta}_{\mathrm{SF}} + \mathrm{Diag}(\Pi)\boldsymbol{\delta}_{\mathrm{Az}} + \mathrm{Diag}(\Sigma)\boldsymbol{\delta}_{\mathrm{El}} \tag{H.3}$$

$$\Pi = \mathbf{\Theta}\mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}\mathbf{m},\ \Sigma = \mathbf{\Gamma}\mathbf{C}_{\mathrm{instru}}^{\mathrm{b}}\mathbf{m}$$

where $\boldsymbol{\delta}_x\,(x = Az,\ El)$ are the misalignment angle vectors, $\boldsymbol{\delta}_{\mathrm{SF}}$ is an n-dimensional scale factor error vector, $\mathbf{Diag}(\mathbf{m})$ is a diagonal matrix consisting of the SRIMU measurement vector $\mathbf{m}$, $\mathbf{Diag}(\Pi)$ is a diagonal matrix consisting of the vector Π and $\mathbf{Diag}(\Sigma)$ is a diagonal matrix consisting of the vector Σ.

APPENDIX I

From Eqs.(5.2b) and (5.26), a transformation from the measurement space to the measured state space and the parity space can be described by the following block matrix,

$$\{\text{The SRIMU Measurement Space}\}\Rightarrow\begin{bmatrix}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\\ \mathbf{P}\end{bmatrix}\Rightarrow\begin{Bmatrix}\text{The Measured State Space}\\ \text{The Parity Vector Space}\end{Bmatrix}$$

or

$$\begin{bmatrix}\boldsymbol{\omega}\\ \mathbf{p}\end{bmatrix}=\begin{bmatrix}(\mathbf{HH})^{-1}\mathbf{H}^{\mathrm{T}}\\ \mathbf{P}\end{bmatrix}\mathbf{m}$$

The inverse of this matrix transforms the measured state space and the parity space to the measurement space. From the SRIMU measurement equation, $\mathbf{H}$ is a sub-matrix of this inverse matrix, which determines the transformation from the measured state space to the measurement space. Let $\mathbf{M}$ be a sub-matrix of this inverse matrix, which specifies the transformation from the parity space to the measurement space, then the following relationships are created.

$$\begin{bmatrix}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\\ \mathbf{P}\end{bmatrix}\begin{bmatrix}\mathbf{H} & \mathbf{M}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{M}\\ \mathbf{0} & \mathbf{PM}\end{bmatrix}\equiv\mathbf{I}$$

and

$$\begin{bmatrix}\mathbf{H} & \mathbf{M}\end{bmatrix}\begin{bmatrix}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\\ \mathbf{P}\end{bmatrix}=\begin{bmatrix}\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}+\mathbf{MP}\end{bmatrix}\equiv\mathbf{I}$$

From the above two equations, the following matrix equations can be derived.

$$(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{M}=0 \tag{I.1}$$

$$\mathbf{PM}=\mathbf{I}_{n-3} \tag{I.2}$$

$$\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}+\mathbf{MP}=\mathbf{I}_{n} \tag{I.3}$$

Because the matrix $(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}$ is non-singular, then from Eq.(I.1),

$$\mathbf{H}^{\mathrm{T}}\mathbf{M}=\mathbf{0} \tag{I.4}$$

The problem is to derive a matrix $\mathbf{M}$ that satisfies the conditions given in Eqs. (I.2) and (I.4). If and only if $\mathbf{M}=\mathbf{P}^{\mathrm{T}}$, applying the following constraint conditions on $\mathbf{P}$: $\mathbf{PH}=\mathbf{0}$ and $\mathbf{PP}^{\mathrm{T}}=\mathbf{I}$, then

$$\mathbf{PM}=\mathbf{PP}^{\mathrm{T}}\equiv\mathbf{I}_{n-3}$$

and

$$\mathbf{H}^{\mathrm{T}}\mathbf{M}=\mathbf{H}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}=(\mathbf{PH})^{\mathrm{T}}\equiv 0$$

From Eq.(H.3)

$$\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}-\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$$

Let

$$\mathbf{U}=\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}-\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} \tag{I.5}$$

then $\mathbf{U}$ is an $n\times n$-dimensional symmetric, positive semi-definite matrix because the rank of $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ is the same as the rank of $\mathbf{P}$. The upper triangular parity matrix $\mathbf{P}$ with positive diagonal elements can be computed by the following algorithms[*].

$$P_{11}^2=U_{11}$$

$$P_{ij}=0 \text{ for } j<i$$

$$P_{1j}=U_{1j}/P_{11} \text{ for } j=2,3,\ \cdots\ ,n$$

$$P_{ii}^2=U_{ii}-\sum_{k=1}^{i-1}P_{ki}^2 \text{ for } i=2,3,\cdots,n-3$$

$$P_{ij}=(U_{ij}-\sum_{k=1}^{i-1}P_{ki}P_{kj})/P_{ii} \text{ for } i=2,3,\cdots,n-3;\ j=i+1,\cdots,n$$

[*] Potter, J. E. and Suman, M.C., *Thresholdless Redundancy Management With Arrays of Skewed Instruments*, AGARD AG-224, 1977, pp. 15-1 to 15-25.

APPENDIX J

Table J-1 Inertial Sensor Simulation Parameters

Sensor Parameters	1	2	3	4	5	6
Gyro drift time const (sec)	480	470	465	475.5	590	480
Gyro drift err (deg/hr)	0.74	0.7	0.69	0.71	0.65	0.63
Gyro bias err (deg/hr)*	40|2.0	40|1.9	35|2.2	35|2.3	45|1.8	42|2.1
Gyro SF err time const (sec)	280	260	270	260	300	265
Gyro SF error (ppm)	50	50	50	50	50	50
Gyro Az misalign err (arcsec)	4.3E+1	3.7E+1	3.8E+1	3.7E+1	3.7E+1	4.0E+1
Gyro El misalign err (arcsec)	4.4E+1	3.8E+1	3.8E+1	3.8E+1	3.6E+1	4.5E+01
Gyro noise (deg/sqrt(hr))	0.71	0.7	0.78	0.72	0.68	0.71
Accel drift time const (sec)	360	360	365	360.5	366	360
Accel drift err (ug)	160	170	154	165	175	150
Accel bias err (ug)	310	320	330	333	312	320
Accel SF err time const (sec)	250	260	250	260	245	260
Accel SF err(ppm)	400	415	420	410	420	405
Accel Az misalign err (arcsec)	4.0E+1	4.8E+1	4.7E+1	4.38E+1	4.1E+1	4.0E+1
Accel El misalign err (arcsec)	4.6E+1	4.7E+1	4.5E+1	4.2E+1	4.0E+1	4.68E+1
Accel noise (ug/sqrt(hz)	100	100	105	100	102	110

* The first column is a typical value of slave node sensor biases and the second column is a typical value of the cg node sensor biases.

Table J-2 GPS Error Simulation Parameters

Error Sources	Standard Deviation	Time Constant (Second)	Note
Ephemeris error	3 (m)	1800	
Ionospheric error	3-7(5)(m)	1800	Rx location and SV elevation dependent
Tropospheric error	1-5(2)(m)	3600	Rx height and SV elevation dependent
Rx Clk Frq error Rx Clk Pha error	0.2m/s		Random walk, equivalent range rate error, random drift
Pseudorange noise	1-3		Rx dependent, white noise
Doppler meas noise	0.1-0.3 m/s		Rx Dependent, white noise
SA effect	33	180	Second-order Markov
Simulated GNSS Attitude errors	Roll and Pitch 0.25^0 Yaw 0.2^0		White noise, dependent on PDOP

Table J-3 ADS and Magnetic Heading Simulation Parameters

Sensor/System Output	Sensor Errors	Standard Deviation (1-sigma)	Note
Pressure Height	Scale factor error (%)	0.02	Random process
	Time delay error (s)	0.06	Random process
	Measurement noise (m)	2	White noise
True Airspeed (TAS)	Measurement noise (m/s)	0.5	Total TAS error
Magnetic Heading	Heading deviation (deg)	2^0	Random constant
	Heading variance (deg)	0.5^0	White noise

Table J-4 Initial Navigation State Error Parameters

Parameter	Initial Errors	Note
L-Position Errors H-Position Error	400m 150m	Met by GNSS positioning solution and pressure altitude
Vn, Ve Vd	1.0m/s 0.5m/s	Met by GNSS velocity solutions
Attitude Errors	1^0	Met by GNSS-based attitude solution or inertial initial alignment

Table J-5 Parameter for MW-GLRT Requirements

Accelerometer Bias (g)	Accelerometer Noise (mg)	Gyro Drift (^{0}/hr)	Gyro Noise (^{0}/s √Hz)	Velocity Error	Attitude Error
2E-4	0.25	40	0.012	Max 12 knots for 2 minutes	Max 2^0 for 2 minutes
Integrity Requirements: Probability of a false alarm is 10^{-6} Probability of a missed alarm is 10^{-5}					

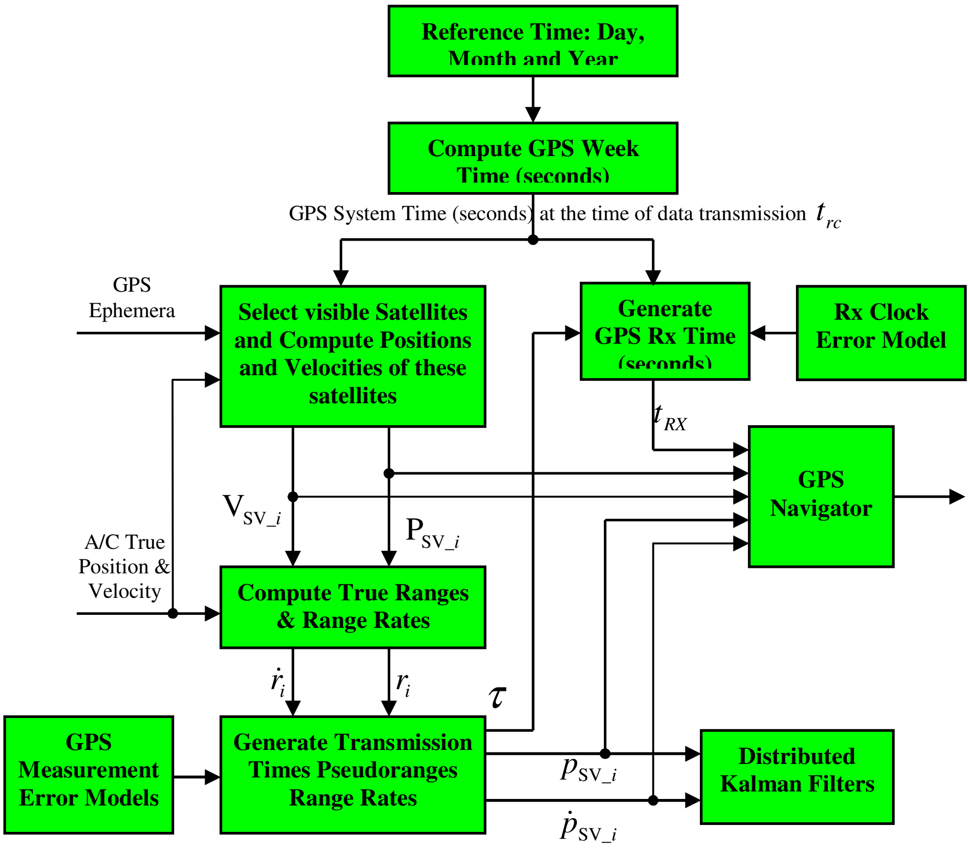

Reference Time: Day, Month and Year
Compute GPS Week Time (seconds)
GPS System Time (seconds) at the time of data transmission t_{rc}
GPS Ephemera
Select visible Satellites and Compute Positions and Velocities of these satellites
Generate GPS Rx Time (seconds)
Rx Clock Error Model
t_{RX}
GPS Navigator
V_{SV_i}
P_{SV_i}
A/C True Position & Velocity
Compute True Ranges & Range Rates
$\dot{r}_i$
r_i
τ
GPS Measurement Error Models
Generate Transmission Times Pseudoranges Range Rates
p_{SV_i}
$\dot{p}_{SV\_i}$
Distributed Kalman Filters

Figure J.1 GPS Measurement Simulation Algorithm Architecture

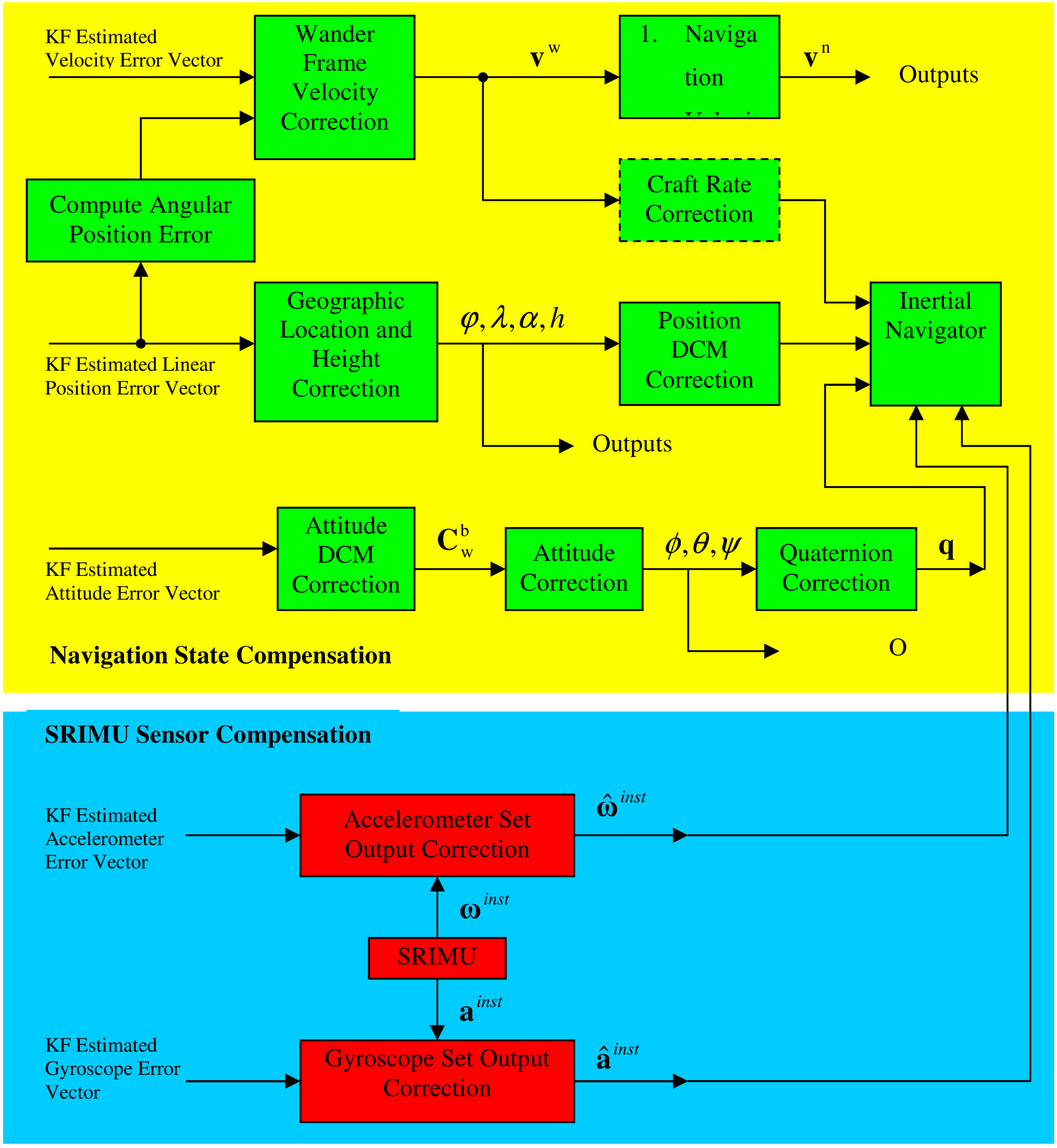


Figure J.2 System Compensation Architecture