12-1995

Design and Analysis of a Navigation System Using the Federated Filter

Stephen J. Delory

DESIGN AND ANALYSIS OF A NAVIGATION SYSTEM USING THE FEDERATED FILTER

THESIS

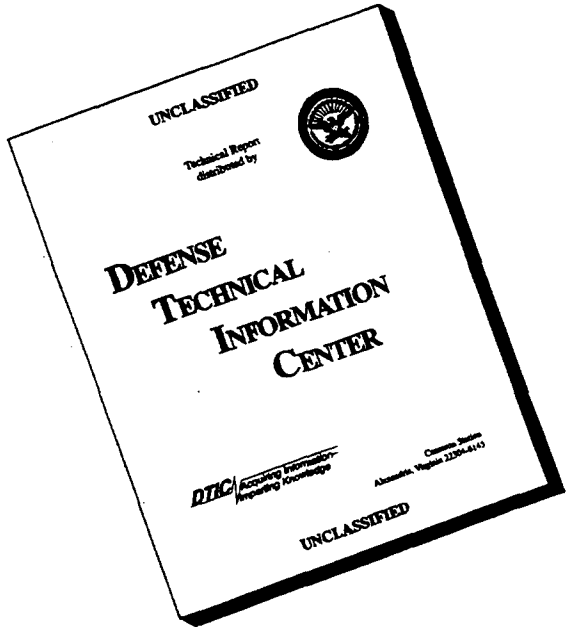
STEPHEN J. DELORY
MAJOR, CANADIAN FORCES

AFIT/GSO/ENG/95D-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GSO/ENG/95D-02

DESIGN AND ANALYSIS OF A NAVIGATION SYSTEM USING THE FEDERATED FILTER

THESIS

STEPHEN J. DELORY
MAJOR, CANADIAN FORCES

AFIT/GSO/ENG/95D-02

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	Master's Thesis

4. TITLE AND SUBTITLE
Design and Analysis of a Navigation System
Using the Federated Filter

5. FUNDING NUMBERS

6. AUTHOR(S)
Stephen J. DeLory
Major, Canadian Forces

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENG/95D-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Capt Paul Lawrence
WL/AAAI-3
Wright-Patterson AFB, OH 45433

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this paper was to design and analyse a federated filter design, to be used for retrofit of an Embedded GPS/INS (EGI) navigation unit into an existing Kalman filter-based air navigation system. A design was selected and simulations were conducted in the Distributed Kalman Filter Simulation software (DKFSIM). As well, a centralized Kalman filter design was simulated under identical conditions for comparison purposes. The federated filter was shown to be a feasible design, with accuracy in position and velocity very close to centralized Kalman filter values. The federated filter design also showed some atractive fault detection and isolation features, superior to the centralized Kalman filter, due to the independent operation of the component Kalman filters. The federated filter was shown to be well worthy of continued study for implementation in air navigation systems, especially where distributed filters are required.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Federated filters, distributed filters, inertial navigation systems	111
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

AFIT/GSO/ENG/95D-02

Design and Analysis of a Navigation System using the Federated Filter

Thesis

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Space Operations

Stephen J. DeLory, B. Eng. (Electrical)

Major, Canadian Forces

December, 1995

Preface

As an Air Navigator, I feel indebted to the AFIT/ENS and ENG faculty for allowing me to pursue the Guidance, Navigation and Control sequence, and allowing me the opportunity to conduct research in a critical-technology area. I want to thank my thesis advisors, Lt Col Bob Riggins and Capt Ron DeLap, for providing me with the necessary support and knowledge to get the work done. Administrative support from my sponsors at Wright Labs, Capt Paul Lawrence and Mr Mike Berarducci, allowed me to 'get on with the job' without having to worry about the paperwork.

Understanding and implementing the federated filter model would have been impossible without the excellent Kalman filter preparation and expertise provided by Dr Peter Maybeck, and the tremendous support, both software and moral, provided by Dr Neal Carlson. Gentlemen, my thanks.

Most of all, I want to thank my boys Benjamin, Simon, and Timothy, and especially my wife Pauline, and, for standing by me throughout my time at AFIT. Your sacrifices were my gain, and I hope I may be allowed to repay you someday.

This work is dedicated to the men and women of Canada's operational sea-going squadrons, HS 423 and HS 443. May you one day be given the tools, such as those researched in this thesis, to do your job safely.

Table of Contents

List of Figures

List of Tables

Design and Analysis of a Navigation System using the Federated Filter

1. Introduction

The United States Air Force Wright Laboratories (Avionics Directorate) has a continued interest in navigation systems, to help maintain a leading position in air vehicle technology, and to sponsor continuing research in these systems. An interest in new methods of data processing for navigation systems, especially distributed filters [1], has been motivated by factors such as distributed sensors in an aircraft, security of classified information, and fault tolerance.

The advent of Global Positioning System (GPS) operational capability makes available unparalleled accuracy in navigation [2], yet for military aircraft is not suitable for a stand-alone navigation system. Military aircraft have unique missions and flight profiles in which:

- GPS signals may be obscured by terrain, jammed, or not viewed by antennae, rendering the stand-alone GPS receiver unusable; or
- high dynamics of the aircraft may make position and velocity estimates by the GPS receiver less accurate than required.

For existing aircraft, a retrofit with GPS receiver equipment is highly desirable, but poses many design problems which must be studied carefully before implementation. One such retrofit possibility is the insertion of an Inertial Navigation System (INS) containing a GPS receiver into an existing navigation system. Wright Labs is interested in research to determine the feasibility and method for such a retrofit, and has sponsored the author to address this particular design problem.

1.1 Background

1.1.1 Air Navigation Systems

The need for the best navigation system possible for military aircraft is clear. In tasks such as routine navigation, tactical navigation, airdrop, or weapons delivery, to name but a few, the success of the military mission is directly tied to the military aircraft's navigation system.

Initially, INS-based navigation systems consisted of a free-running INS which was not filtered. Comparison with other navigation systems to make a single navigation solution was done

by hand. With the advent of powerful onboard digital computers and development of real-time data processing algorithms such as the Kalman filter, different sources of data could be combined mathematically in flight to produce an integrated navigation solution.

Modern military aircraft have a multitude of sensor inputs. One aircraft may have measurements on position, velocity, attitude, and time from a wide variety of sensors such as Doppler ground radar, synthetic aperture radar, barometric and radar altimeter, a GPS receiver, and star tracker. These sensors produce many types of data, at different rates, with widely varying noise types and strengths. The problem facing a data processing algorithm for a navigation system is to use this data effectively to determine the desired parameters.

1.1.2 The Kalman filter

The Kalman filter, a data processing algorithm which has the capability of combining different measurements to provide optimal estimates of parameters of interest, is probably the most widely used data integrator in air navigation, although there are others [3]. It has gained acceptance due to its optimal performance under certain assumptions, good stability and robustness in the regime of operation, and reasonable processing throughput.

Typically, the Kalman filter in an air navigation system operates in what is known as the indirect method. Instead of estimating position, velocity, and attitude of the air vehicle, the Kalman filter estimates position, velocity, and attitude *errors* of the INS, as well as compensatory states for INS components and external sensors [3,4,5]. Since the dynamics of the INS errors are much slower than the dynamics of the aircraft, update rates can proceed at a lower rate, which is less burdensome to the on-board processor. The Kalman filter state estimates are summed with the INS output to yield best estimates of position, velocity, and attitude.

The Kalman filter is an optimal filter; that is, it extracts the maximum possible information from measurements when forming the conditional mean estimates as a minimum mean square error (MMSE) solution. The estimates of the Kalman filter are jointly Gaussian [5], and so by most measures (mean, mode, or median) the estimate is optimal. Additionally, due to the Gaussian nature of the data, the state estimate (first order statistics) and covariance matrix (second order statistics) completely define the Gaussian density for states, given the measurements observed up to a given time.

Some applications use a single Kalman filter for estimations, where others use a network of Kalman filters to derive their estimates. Three types of Kalman filter applications are considered here: the centralized Kalman filter, the cascaded Kalman filter, and the federated filter. For the purposes of practical implementation, versions of the centralized filter and the federated filter could be implemented as a retrofit without significant hardware change, and so are two practical solutions. The cascaded Kalman filter is not considered as an alternative design for analysis in this thesis, but is discussed for completeness.

1.1.2.1 Centralized Kalman Filter Application

In this thesis, the centralized Kalman filter is defined to be the following. It is essentially the Kalman filter described in Section 1.1.2, implemented in real-time in a dedicated processor in an airborne computer. This centralized Kalman filter provides a navigation solution by combining measurement information from sensors, usually unprocessed. The centralized Kalman filter is able to combine these measurements to produce, typically, estimates of errors in INS indications of position, velocity, and attitude [6].

The centralized Kalman filter has many advantages in the navigation application. It requires only a single processor, will adapt to different measurement rates and accuracies, and is relatively simple to implement. The estimates of the Kalman filter may be used to correct errors in the INS, in a feedback configuration. A block diagram showing a centralized Kalman filter in feedforward configuration (no corrective resets from the Kalman filter to the INS) implemented in an aircraft navigation system is shown in Figure 1 - 1.

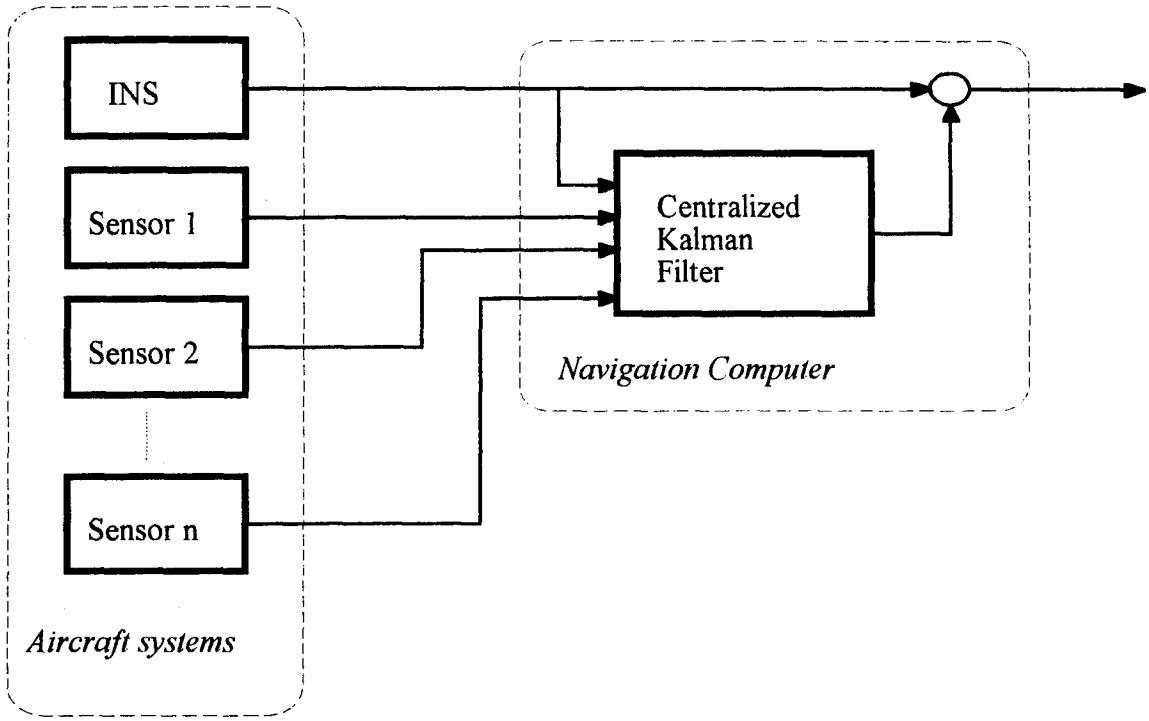


Figure 1 - 1 Centralized Kalman Filter Air Navigation System

1.1.2.2 Cascaded Filter

The cascaded Kalman filter, hereafter called the cascaded filter, is a subclass of distributed Kalman filters, defined as a filter network where estimates are made by at least two filters and combined using some algorithm to form a single solution. In this thesis, the cascaded Kalman filter is defined to be the following. Some or all sensors provide measurements to one or more Kalman filters, which also incorporates measurements from an INS. The output of this (these) local filter(s) is directed to another Kalman filter termed the master filter, which may or may not have measurements from other sensors. The master filter treats each of the state estimates of the local filters as measurements, and uses these measurements to update state estimates of the master filter. Estimates of the master filter, combined with the INS outputs, are used as system outputs.

The cascaded filter is used in aircraft where a distributed architecture is required [3,7]. For example, a GPS receiver may not allow the direct transmission of pseudorange (PR) and pseudorange rate (PRR) information due to security reasons. The GPS receiver can contain a local filter, and send filtered information as measurements to a master filter in the navigation computer, thus implementing the cascaded filter.

Factors motivating the cascaded filter are :

- Redundancy. Reliability of the navigation system may be improved by incorporating multiple filters, each of which may provide a navigation solution in the event of component failures;
- GPS Receiver Design. Some GPS receivers were designed to have GPS data processed within the receiver by a Kalman filter and not output raw GPS measurements. This motivated a cascaded filter implementation in aircraft using a Kalman filter for a navigation system; and
- Parallel Processing. All processing in the centralized filter is carried out typically in one processor. In distributed filters, processing for each local filter and the master filter is carried out in parallel on separate processors. The amount of processing time for a Kalman filter is dependent on the number of states that the filter maintains. The overall throughput can be better than the processor running a centralized filter, since each of the filters in the decentralized approach have fewer states than the centralized filter.

A block diagram representation of a possible implementation for a cascaded filter is shown in Figure 1 - 2.

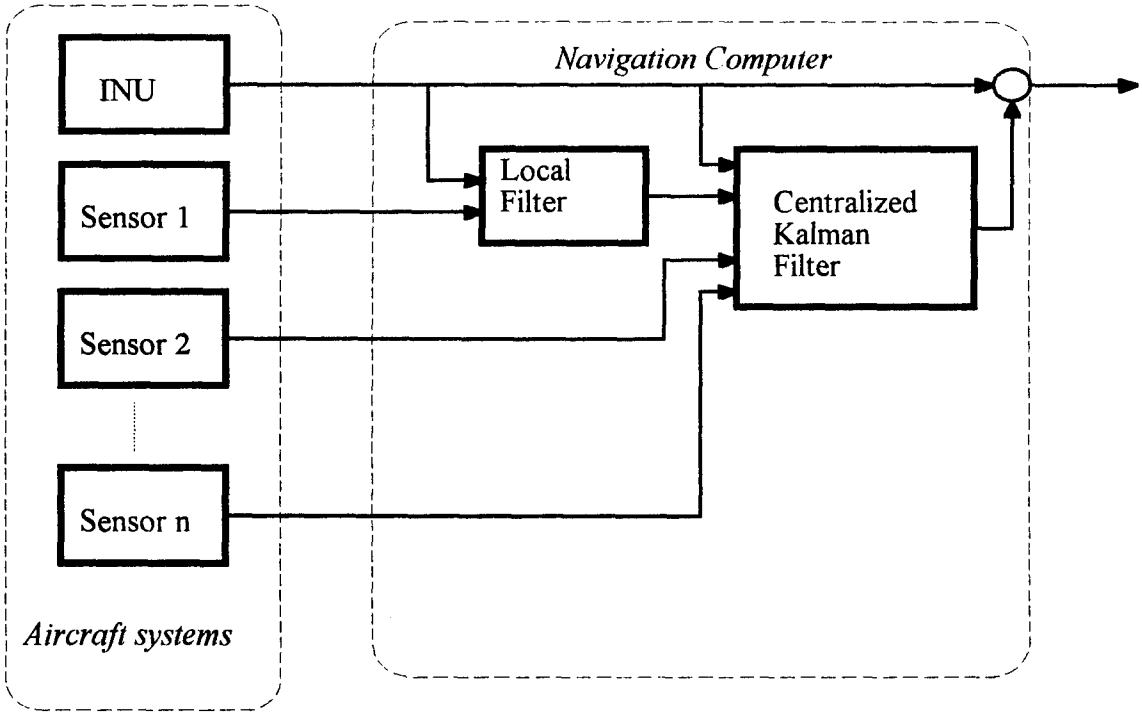


Figure 1 - 2 Cascaded Filter

1.1.3 Federated Filter

The federated filter is considered as a possible solution to the retrofit problem. The development of the federated filter, a brief description of the federated filter operation, and some salient characteristics are presented in this section.

1.1.3.1 Description

Recently a new technique in data processing with navigation systems application, called the federated filter, was developed [4]. This filter is a subclass of the distributed Kalman filter, and is best generally described as a method of sharing the information available to the navigation system among the local and master filters. Sensor inputs are sent as measurements to local filters (typically one measurement per local filter), which operate autonomously. Each of these local filters, by virtue of the INS and sensor data it receives, holds part of the total system information. This system information from the local filters is then fused with the master filter information to form a full solution using all of the system information.

The federated filter shows promise for a multi-input navigation system for a number of reasons [8]:

- The distributed nature of the federated filter allows parallel processing in the local filters, possibly allowing faster throughput than the centralized filter does.
- The federated filter allows the ability to detect difficult sensor faults, in particular slowly-deteriorating sensor data or 'soft faults', which may be difficult to do in the cascaded filter or centralized filter.
- The master filter of the federated filter is not susceptible to problems caused by correlated outputs from local filters, and may be more robust than the cascaded filter [4]. This problem is further defined in Section 2.4.5.

A block diagram description of a federated filter navigation system is shown in Figure 1 - 3. Note that each sensor sends measurements to a local Kalman filter which can operate independently as shown, or with information 'feedback' (not shown for simplicity). Although similar in information flow and distribution of filters, the master filter of the federated filter works much differently than that of the cascaded filter. The federated filter is explained in more detail in Chapter 2.

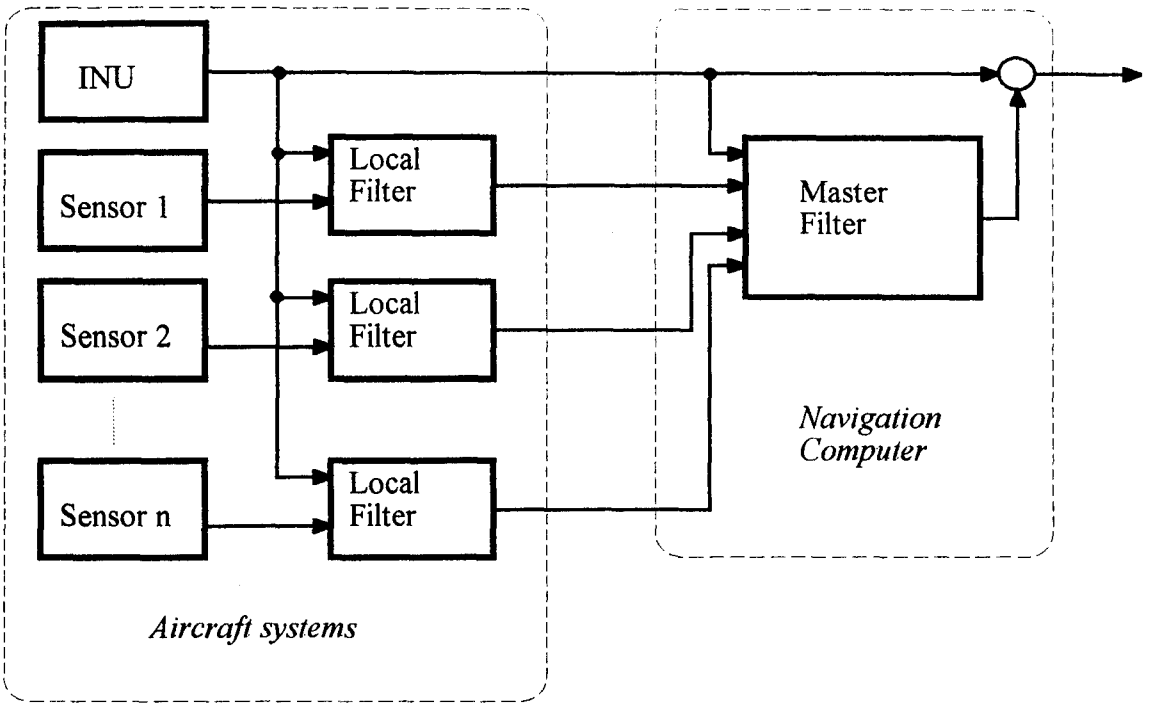


Figure 1 - 3 Federated Filter

1.1.3.2 Literature Review

Dr. Neal Carlson presented the theory of information sharing in the federated filter in a number of forums [4,8,9]. He demonstrated that certain advantages may be gained by using the federated filter over the centralized Kalman filter.

Following the introduction of the federated filter, follow-on research was conducted. Wright Labs sponsored research into uses of the federated filter as a navigation system, and independent researchers looked into the characteristics of the federated filter, as well as possible uses for the federated filter.

The Common Kalman Filter development program, whose objective has been to establish a basic set of estimation and Fault Detection, Isolation and Reconfiguration (FDIR) system design techniques, examined different distributed filter designs [10]. In addition, Wright Labs sponsored Integrity Systems, Inc. to build the Distributed Kalman Filter Simulator (DKFSIM) software (described in Chapter 3) [11]. Wright Labs also sponsored the construction of a real-time Integrated Test Bed (ITB), where combined hardware/software research and development of the federated filter could be conducted. The ITB is not yet operational [12].

Some research in the federated filter was conducted by Drs. Gao, Krakiwsky, and McLellan of the University of Calgary Department of Surveying Engineering, to examine the

suitability of the filter for kinematic GPS positioning [13]. They concluded that a better fault tolerance performance than a cascaded Kalman filter could be achieved with the federated filter, and indicated directions for follow-on research.

Captain Paul Lawrence conducted a Master's thesis researching the federated filter at the Air Force Institute of Technology at Wright-Patterson AFB [14]. He compared the federated filter with a centralized Kalman filter in order to examine accuracy and overall performance, operation under sensor failure conditions, and potential for failure detection and isolation. The federated filter operation was characterized, with promising results, and great potential for follow-on study was noted. Capt Lawrence suggested that additional testing under simulated sensor and INS failure conditions would extend the set of comparisons for the federated and centralized Kalman filters.

1.1.4 Introduction of EGI

GPS information, used in the formation of a navigation solution, is highly complementary to INS information [6,3,15]. GPS data is highly accurate in low frequency, but subject to high-frequency noise and to short-term outages. INS systems, on the other hand, are subject to long-term drift, but will provide a solution almost continuously, even if all external sensor information, such as GPS, is unavailable. The GPS/INS combination provides a very good, consistent, reliable navigation solution.

Early research in the GPS system [6] showed that an integration of a GPS receiver and an INS, especially at a low level (raw data may be passed from one system to the other) would result in a synergistic relationship between the two, with information from each unit aiding the operation of the other.

Technology development has allowed the miniaturization of the GPS receiver down to a single circuit board. It has become possible to embed the GPS receiver into the INS electronics unit. Now, the low-level integration is implemented within the confines of a single electronics unit enclosure. This embedded GPS/INS (EGI) has a number of attractive features that allow it to operate as a complete navigation system;

- Reliability. The INS is able to provide accurate position data when the GPS receiver has insufficient data to provide a full navigation solution. Thus a high-accuracy solution is available when a stand-alone GPS system would be unable to provide a solution.

- Use of Partial GPS data. Partial GPS data (for example, reception of data from only 1,2 or 3 satellites) may be incorporated into the Kalman filter, and used to update INS error states, even if there is insufficient information for a full GPS solution. Thus, all available GPS information is used to maintain a high-quality navigation solution. The stand-alone GPS receiver must have four-satellite reception (or three with altitude data) in order to form a solution.

- Ease of data exchange. With both INS and GPS electronics in a single housing, low-level information may be passed through digital or analog data lines or databusses, shared memory, or other specialized signal transfer techniques such as optical links. This can be implemented with relative ease as compared to data transfer between a stand-alone GPS receiver and INS.

- Small size and weight. Incorporating both INS and GPS in one housing eliminates the need for two sets of power and signal wiring, shock mounting, and housing that would be required with separate GPS receiver and INS.

- Security of GPS information. Since GPS pseudorange and pseudorange rate information for military users is classified, data transfer of this information from the stand-alone GPS receiver to another device would require a classified bus. With the EGI, pseudorange and pseudorange rate are passed within the confines of the EGI, and security of the information is retained.

Firms which develop avionics, such as Litton Systems and Honeywell, have been working to develop the EGI. One such EGI is the Honeywell H-764G, which combines a high-accuracy INS and P-code GPS receiver in one self-contained unit measuring 18 x 18 x 25 cm, and weighing 8.4 kg [16]. The Honeywell system provides a filtered GPS/INS solution of 16 meters Spherical Error Probable (SEP), meaning 50% of the solutions are within a sphere of this radius centered at the true position, and velocity with maximum velocity error of 0.01 m/sec rms. It can also provide a navigation solution from GPS-only or INS-only information.

1.1.5 The Problem: Incorporating an EGI into a Navigation System

A number of air vehicles currently have navigation systems which incorporate INS information as well as data from a number of other sensors. These systems use a centralized Kalman filter or cascaded Kalman filter to provide a navigation solution. Since many of these

systems were designed and built prior to GPS operational capability, GPS receivers were not incorporated in the initial design. Adding a GPS receiver to an existing navigation system, to increase system accuracy, is desirable.

With the advent of the EGI, a retrofit could incorporate not just a GPS receiver, but for the same mechanical effort, a fully integrated GPS/INS navigation system. This system would retain all of the previously installed sensors, as well as the installed INS, and so would be equipped with two INS after EGI retrofit. The EGI may be able to provide a number of desirable features such as high-accuracy navigation data, higher reliability, and fault detection into a larger overall navigation system, *if the navigation system is properly designed.*

One design approach is the application of Kalman filter theory to create a single solution, incorporating information from all sources available in such a manner that best satisfies the criteria for the navigation system. These criteria (stated broadly for now) are navigation solution accuracy and system reliability.

1.1.6 Possible Designs for Incorporating an EGI into a Navigation System

There are three groupings of designs to be considered using the filtering methods listed in Section 1.1.4. These are the centralized Kalman filter, the cascaded Kalman filter and the federated filter. The cascaded Kalman filter is not considered further in this thesis, due to modeling constraints and implementation problems.

1.2 Problem Statement

This effort will research, model, and evaluate the federated filter design against the centralized filter design for a multi-sensor navigation system incorporating an embedded GPS/INS system.

1.3 Research Objectives

1.3.1 Assess Performance of Both Designs

This research will assess the performance of the centralized Kalman filter design and the federated filter design, in a high-dynamics environment of a military aircraft. The criteria for comparison will be accuracy of position, velocity and attitude information from the system, with all sensor information and with some sensor information absent. The flight profile of the vehicle will

contain both low-dynamic and high-dynamic portions to challenge the navigation systems in different regimes of flight.

It is worth noting from the outset that the centralized filter design receives information from one INS and the federated filter design receives information from two INSs. Also, the federated filter design does not strictly follow the theory of the federated filter as described in Section 2.5. The comparison in this thesis, then, is between two possible system designs that use the centralized filter and federated filter, not between the centralized Kalman filter and federated filter methodologies.

1.3.2 System Reliability

System reliability is an important consideration for a military navigation system, as successful completion of the mission may depend on a functioning navigation system. Both the cascaded Kalman filter and the federated filter will be assessed for fault tolerance and continued operation in the face of sensor outages and failed navigation system subcomponents.

1.4 Research Approach

1.4.1 Assess Performance of Both Filter Implementations

A set of computer models, incorporating an EGI in a larger navigation system, will be developed in the simulation software DKF Simulator (DKFSIM) [11,22]. Models of the centralized filter and the federated filter will be simulated in Monte Carlo runs, using identical input data, to produce side-by-side comparisons. Input data of flight profiles will be used to simulate a variety of military tasks such as routine flight, low-level tactical flight and weapons delivery. These comparisons will provide insight into the performance characteristics of the compared models.

1.4.2 System Reliability

System reliability will be examined using the simulation software DKF Simulator (DKFSIM). Reliability aspects to be examined are:

- Degraded/failed sensor input
- System internal failures; and

- Fault detection and isolation.

1.5 Resources Needed

In order to conduct the research, the following resources have been required:

- A 486-based PC with 487 co-processor, necessary for use of the FORTRAN installation;
- The simulation software DKFSIM, including manuals, and permission to modify source code from the author; and
- Lahey FORTRAN 77 installation. The Lahey FORTRAN is used due to Lahey-specific routines in DKFSIM.

1.6 Assumptions

A number of assumptions are required to proceed with the research. These are listed below.

- The DKFSIM filter models are correct and adequate for the simulation of the EGI as well as the remainder of the navigation system. This is a fair assumption based on comparison of the DKFSIM model proposed for use and the Kalman filter used in the Honeywell H-764G.
- The EGI can output state estimates and covariances. This is reasonable based on a study of the Honeywell H-764G [16,17], indicating the H-764G or similar EGI may be configured to make information such as the state estimates and covariances available on a 1553B bus.
- The EGI can have state estimates and covariances reset by the navigation system. Again, a study of the Honeywell H-764G indicates this information may be passed to the EGI via 1553B bus. This access to the EGI Kalman filter would be required for federated filter operation in some modes.

1.7 Overview of Thesis

Chapter 1 gives a chronological development of navigation systems up to the present time and provides motivation for the research. Chapter 2 explains basic Kalman filter theory and federated filter theory applied to navigation systems. Chapter 3 presents the filter models and computer simulations used in the analysis. Chapter 4 presents the findings and results of the simulations, and Chapter 5 presents conclusions based on the research and gives recommendations based on these conclusions.

2. Navigation Systems and Filter Theory

2.1 Introduction

This chapter is comprised of three parts. First, navigation system components, sensors, and related information are presented. Next, Kalman filter theory and federated filter theory is developed to show how these navigation system component outputs are used. Last, two alternative designs are described for the EGI retrofit problem presented in Chapter 1.

2.2 Frames of Reference

Before proceeding with further implementation, some discussion of the frame of reference for the data is necessary. There are a number of different frames of reference that are relevant to this discussion. They are:

- the Earth Centered, Earth Fixed (ECEF) frame;
- the Geographic frame; and
- the Body frame

A short description of each follows.

The ECEF frame of reference is an orthogonal, right-hand coordinate system, with origin at the earth's center of mass. The z_e axis is aligned with the Greenwich meridian, the y_e axis projects through the North Pole, and the x_e axis projects through the equator at 90^0 E latitude. Note that the entire frame rotates with and is fixed with respect to the earth. Figure 2 - 1 shows the ECEF frame.

The geographic frame is also an orthogonal, right-hand coordinate system, but with origin at the INS location. The east-north plane of the frame is parallel to a plane tangent to the earth's surface directly beneath the aircraft. The north axis lies in the intersection of this plane and a plane in which the local meridian lies. Figure 2 - 2 illustrates the geographic frame.

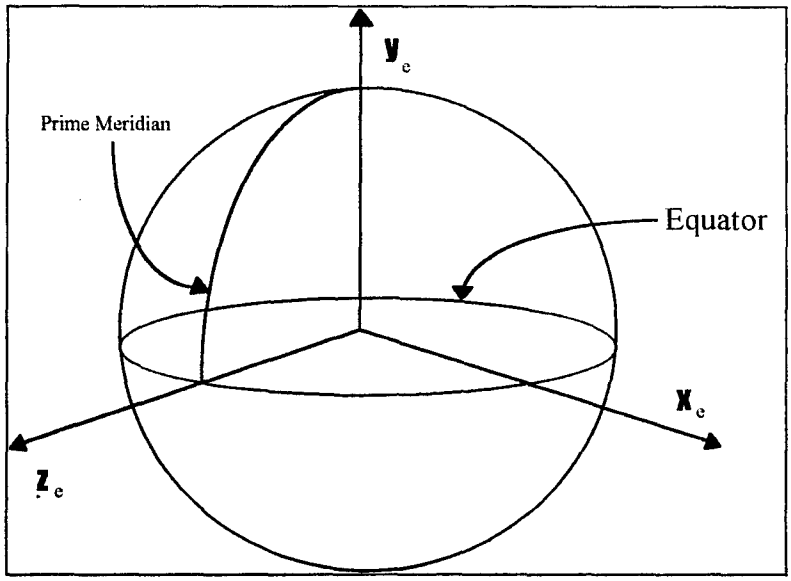


Figure 2 - 1 ECEF Frame

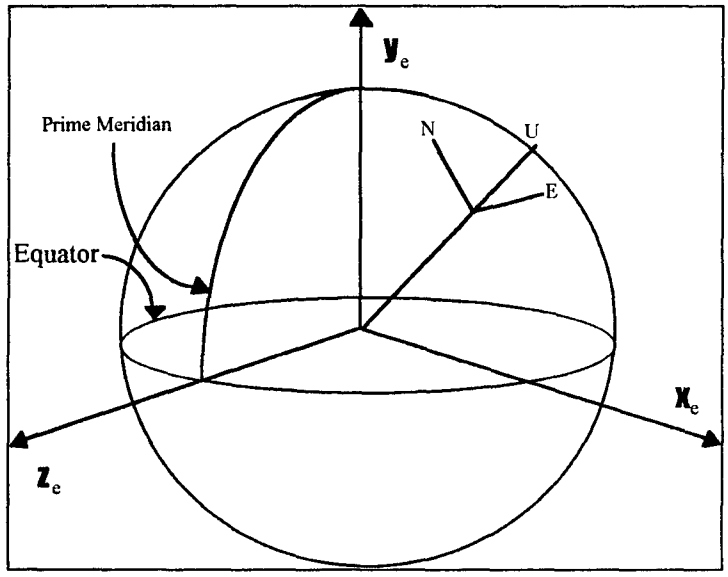


Figure 2 - 2 Geographic Navigation Frame

The body frame is an orthogonal right-hand system, with origin at the aircraft center of mass. The body frame axes are the same as the pitch, roll and yaw axes of the aircraft. Figure 2 - 3 describes the body frame (view of aircraft is from below).

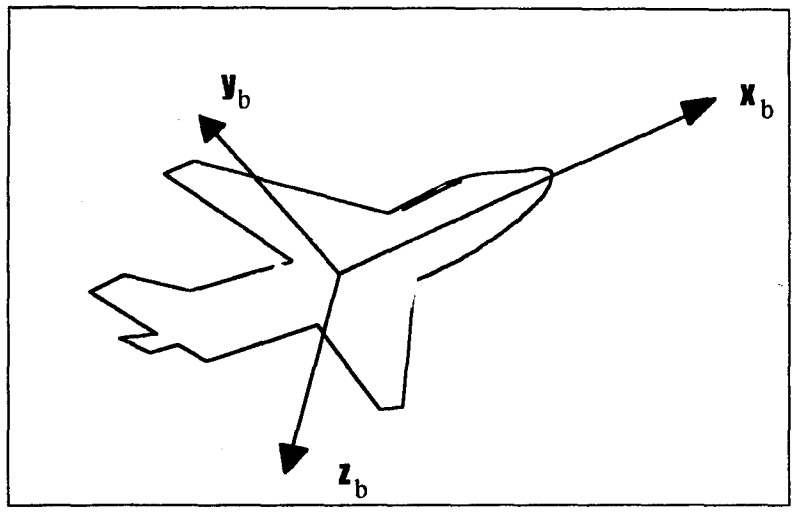


Figure 2 - 3 Body Frame

2.3 Navigation System Components

Navigation system components of a typical high-performance military aircraft are described in this section. The heart of the navigation system, the strapdown Ring Laser Gyro (RLG) INS is described first, followed by the sensors that provide measurements to the navigation system - GPS, Synthetic Aperture Radar High Resolution Mapping (SAR-HRM) and Synthetic Aperture Radar Precision Velocity Updating (SAR-PVU), as well as a Terrain-Aided Navigation (TAN) system.

2.3.1 Ring Laser Gyro Strapdown INS

The strapdown RLG INS consists of sensing devices, an INS computer, and associated supporting hardware. It is able to provide rapidly updated real-time solutions for position, velocity, and attitude. The sensing devices consist of at least three accelerometers and three RLGs. The accelerometers provide a measure of acceleration, whereas the RLGs provide a measure of angular motion. Since these devices are fixed with respect to the body axes of the airframe (or strapped down), the INS is referred to as a strapdown INS. The measurements of acceleration are integrated twice to determine aircraft position. The RLG information is used to determine the orientation of the aircraft. Together, these devices provide all the information required for an inertial solution of position and velocity. The INS solution is determined without external measurements, just its own measurements of aircraft motion and a model of gravity.

However, due to errors in the INS (measurement errors in the sensing devices, computing errors, etc.), the INS solution tends to drift from the actual values of position and velocity. Typical drift rates for an aircraft range from a few nautical miles per hour to a small fraction of a nautical mile per hour, depending on the quality of the INS.

2.3.2 GPS

GPS is a space-based positioning, velocity and time system that has three segments: Space Segment, Control Segment, and User Segment [18].

The Space Segment consists of 24 satellites in six orbital planes. The satellites operate in nearly circular orbits at an altitude of 20,200 km. The satellites circle the earth at an inclination angle of 55 degrees with approximately a 12 hour period. The spacing of satellites is carefully designed to provide adequate coverage over the entire surface of the earth to allow accurate position fixing.

The Control segment consists of the support infrastructure required to keep the satellites flying and keep them updated with the latest information they require for transmission of navigation signals. The Master Control Station is located at Falcon AFB, Colorado.

The User Segment refers to the GPS receiver units. These units may be fixed in location, in vehicles such as ships and aircraft, or hand-held.

Two types of GPS service are available: Standard Positioning Service and Precise Positioning Service (PPS). The PPS is designed for military use, and provides to the user a position solution with accuracy on the order of 10 meters SEP. The SPS is designed for civilian use, and provides a solution on the order of 30 meters SEP.

2.3.3 Synthetic Aperture Radar

Synthetic Aperture Radar is a radar system in which radar return signals are digitized and mapped to geographic space by summing the position of the vehicle with the position information of the returned radar signal [19]. Precisely measured Doppler shift in return signals can also be used to provide highly accurate relative velocities between the aircraft and the ground.

In addition to other tactical uses of SAR, navigation information can be derived from landmark ranging using the 'map' constructed from radar returns, as well as velocity

measurements of the aircraft using relative velocity information. A more detailed explanation is contained in Section 3.2.7.

2.3.3.1 High Resolution Mapping

In synthetic aperture radar high-resolution mapping, (SAR-HRM), the SAR generates a high-resolution map of the terrain features generated by the radar from a two-dimensional array of range and range-rate returns [11]. The operator (i.e. pilot or navigator) designates a known target on the map by moving a cursor to the associated location on the map. This process provides a range and range rate measurement to that target. If the aircraft velocity is known, the range-rate measurement provides a measure of the angle between the velocity vector and the line-of-sight to the target. A three-dimensional position fix may be obtained by using two such fixes, or by using an independent source of altitude.

2.3.3.2 Precision Velocity Updating

In the SAR-PVU mode, the SAR makes a series of range-rate measurements in different directions according to a pre-set geometric pattern. If taken simultaneously, each set of four range-rate measurement would be sufficient to provide a velocity fix for the aircraft in all three dimensions. In reality, a finite period of time is needed to slew the antenna across the different pointing directions. The navigation system can process these measurements one at a time, and use the INS outputs to extrapolate the velocity measurements from one time to the next.

2.3.4 Terrain-aided Navigation

The terrain aided navigation (TAN) system uses a radar altimeter to make measurements of height above ground. This data is subtracted from the altitude from the INS to give terrain height. The time-referenced sequence of terrain height data results in a profile of terrain elevations and slopes along the aircraft track. This data is compared to a stored database of Digital Terrain Elevation Data (DTED). The DTED data is stored in two-dimensional array form as a function of latitude and longitude. The comparison results in measurements of the aircraft altitude and aircraft horizontal position.

2.4 Kalman Filters

This section addresses Kalman filters, the data processing algorithm on which all the considered navigation systems are based. First, operation of the Kalman filter is discussed. Next, the linearization of the Kalman filter is developed. Applications of the Kalman filter in a centralized filter and federated filter are then presented.

2.4.1 Fundamentals of Kalman Filter Theory

The Kalman filter is an optimal recursive data processing algorithm. The algorithm estimates the value of a condition or number of conditions of a system, called system state(s). The state space formulation of the dynamics model takes the following form:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \qquad (2\text{-}1)$$

where:

x is the state vector of dimension n

$\dot{\mathbf{x}}$ is the first-order time derivatives of the state vector

F is the homogeneous state dynamics matrix (n x n)

B is the control input matrix (n x r)

u is the control input vector of dimension r

G is the driving noise input matrix

w is the noise inputs to the system

Since there are no control inputs related to our system of interest, the **B**(t)**u**(t) term is dropped. The expected value of the white Gaussian driving noise vector, **w**(t) is:

$$E[\mathbf{w}(t)] = \mathbf{0} \qquad (2\text{-}2)$$

and:

$$E\{\mathbf{w}(t)\mathbf{w}^T(t+\tau)\} = \mathbf{Q}(t)\delta(\tau) \tag{2-3}$$

where:

$\mathbf{Q}(t)$ is the process noise strength

$\delta(t)$ is the Dirac delta function

For an air navigation system, the actual output parameters of the navigation system such as pitch, roll, position, etc. have high dynamics, i.e. rapidly changing positions, velocities, and attitudes. The preferred method of operation is to estimate the error states of the INS (plus other required states). These error states have low dynamics, by comparison, e.g. north position INS error changes much more slowly than north position. Because they change relatively slowly, error states are more easily estimated and propagated from one time to the next in the filter. Restated in the error state variables, Equation 2-1 is rewritten as:

$$\dot{\delta\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2-4}$$

The Kalman filter incorporates sampled-data measurements from external sensors. The equation used to describe linear measurements is:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\delta\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2-5}$$

where:

$\mathbf{H}$ is the observation matrix

$\mathbf{v}$ is the discrete-time measurement noise vector

The noise vector $\mathbf{v}$ has zero mean and is white Gaussian noise, with covariance

$$E\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\} = \begin{cases} \mathbf{R}(t_i) \text{ for } t_i = t_j \\ \mathbf{0} \text{ for } t_i \neq t_j \end{cases} \tag{2-6}$$

The vector of state estimates and covariance matrix of those estimates are propagated from one time to a later time, without outside measurements. The discrete-time propagation equations for the Kalman filter are:

$$\delta\hat{\mathbf{x}}(t_{i+1}^-) = \Phi(t_{i+1}, t_i)\delta\hat{\mathbf{x}}(t_i^+) + \mathbf{B}_d(t_i)\delta\mathbf{u}(t_i) \tag{2-7}$$

$$\mathbf{P}(t_{i+1}^-) = \Phi(t_{i+1}, t_i)\mathbf{P}(t_i^+)\Phi^T(t_{i+1}, t_i) + \mathbf{G}_d(t_i)\mathbf{Q}_d(t_i)\mathbf{G}_d{}^T(t_i) \tag{2-8}$$

where:

P is the covariance matrix

Φ is the state transition matrix

$\mathbf{G}_d$ is the discrete-time noise distribution matrix

$\mathbf{Q}_d$ is the discrete-time process noise covariance matrix

The state transition matrix Φ is used to propagate the state vector and covariance matrix forward in time. The minus sign superscript indicates prior to measurement update, the plus sign superscript indicates after measurement update, and the subscript d indicates discrete-time formulation.

When measurements are available, the system state and covariance estimates are updated with the new information. The Kalman filter measurement update equations are:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}(t_i)\{\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)\}^{-1} \tag{2-9}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\,[\mathbf{z}(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \tag{2-10}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{2-11}$$

where:

K is the Kalman filter gain

The new updated estimates may then be propagated to a later time by the Kalman filter.

The Kalman filter theory assumes a number of conditions [5]. These are:

- The system can be described as a linear system.
- Process noises are have a flat spectral density, i.e. are 'white', and measurement noises are discrete-time white noises, i.e., independent in time.
- The probability density function describing measurement and process noise amplitude is Gaussian.

Estimation of systems which do not meet these criteria must use a technique of implementing the Kalman filter (or use another means of estimation).

2.4.2 Linearized Kalman Filter

One such technique is the linearized Kalman filter. It is used where the system is nonlinear, such as in the model for INS error states. The system is then described by [20]:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2-12}$$

The state dynamics vector **f[x(t),u(t),t]** is a nonlinear function of the state variable **x**(t), time t, and the control input (assumed zero in this case). For clarity in the derivation, the state vector **x**(t) in this section represents the error state vector $\delta\mathbf{x}(t)$ used in the previous section. In addition, the measurement equation may be a nonlinear function:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \tag{2-13}$$

Since the Kalman filter depends on having a linear model, the nonlinear system shown must be linearized for Kalman filter operation.

Assume that a nominal state trajectory $\mathbf{x}_n(t)$ may be generated which satisfies:

$$\mathbf{x}_n(t_0) = \mathbf{x}_{n_0} \tag{2-14}$$

and

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}(t), t] \tag{2-15}$$

The nominal measurements which accompany the nominal trajectory are

$$\mathbf{z}_n(t_i) = \mathbf{h}[\mathbf{x}_n(t_i), t_i] \tag{2-16}$$

The perturbation of the state derivative is obtained by subtracting the nominal trajectory from the original nonlinear equation:

$$[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)] = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] - \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2-17}$$

The equation above may be approximated to first order by a Taylor series expansion:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}[t; \mathbf{x}_n(t)]\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2-18}$$

where $\delta\mathbf{x}(t)$ represents a first-order approximation of the process $[\mathbf{x}(t)-\mathbf{x}_n(t)]$, and $\mathbf{F}[t;\mathbf{x}_n(t)]$ is a matrix of partial derivatives of $\mathbf{f}$ with respect to its states, evaluated along the nominal trajectory

$$\mathbf{F}[t; \mathbf{x}_n(t)] = \left.\frac{\partial \mathbf{f}[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t)} \tag{2-19}$$

The perturbation measurement equation is similarly derived and is given as:

$$\delta \mathbf{z}(t_i) = \mathbf{H}[t; \mathbf{x}_n(t_i)]\delta \mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2-20}$$

where:

$$\mathbf{H}[t; \mathbf{x}_n(t)] = \left.\frac{\partial \mathbf{h}[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t)} \tag{2-21}$$

The linearized filter is driven by $[\mathbf{z}(t_i) - \mathbf{z}(t_i)]$ and produces best estimates of $\delta \mathbf{x}(t)$, denoted as $\delta \hat{\mathbf{x}}(t)$. An estimate of the whole-valued quantities of interest is given by the equation:

$$\hat{\mathbf{x}}(t) = \mathbf{x}_n(t) + \delta \hat{\mathbf{x}}(t) \tag{2-22}$$

The expression for the linearized Kalman filter is useful, provided that the linearization assumption is not violated. However, if the nominal and actual trajectories differ by too much, unacceptable errors may result. Thus, care must be used not to exceed the acceptable range of the linearized Kalman filter.

2.4.3 Centralized Kalman Filter Navigation System

The centralized Kalman filter navigation system implements the equations shown in the previous section. The hardware implementation for the filter is usually carried out on a single processor using memory available to that processor. A typical centralized Kalman filter for an air navigation system has nine error states for the INS (three positions, three velocities, and three tilts), plus other INS and sensor states. Processing time is a challenge, and speed increases are usually sought through reducing the number of states to the minimum required to fulfill operational system accuracy requirements.

2.4.4 Cascaded Kalman Filter Navigation System

In the cascaded Kalman filter navigation system, one or more local filters operate by using sensor and INS data for measurement updates. State estimates from the local filters are then sent to a master Kalman filter (master filter), as measurements for the master filter. Estimates provided

by the master filter are then the system output. This is sometimes called "filter driving filter" or "loose integration".

Processing of the raw information by the local Kalman filter causes the output data to contain time-correlated noise. The master filter uses this processed data as measurements for updating. This violates the assumptions of white Gaussian noise in the operation of the Kalman filter, and has the effect of creating falsely low covariances in the master filter. A number of problems arise from this data.

- The solution provided by this system is sub-optimal, where the level of suboptimality may be small or may constitute serious degradations.
- The time-correlated error in the measurements may cause stability problems in the master filter.

This problem of correlated noise is effectively avoided by spacing measurements sufficiently far apart in time to the master filter. For example, with a GPS/INS local filter, measurements are typically spaced 5-15 seconds to the master filter in a cascaded system. This allows the measurement errors to decorrelate in time and permit stable filter operation.

For optimal performance, the local filter would pass the state estimates and covariance matrix values to the master filter. However, some implementations are constrained by the type of data available to the master filter from the local filter(s), or by high computational loading. For example, in a common implementation the covariance matrix **P** is *not* passed on to the master filter. The covariance matrix contains information on the covariance of each state, as well as cross-covariances between states. The master filter attempts to account for **P** through the measurement noise covariance matrix **R** (Equation 2-6, 2-9). Usually, the terms of **R** in the master filter are given a fixed value, adjusted to account for **P** over the dynamic range of the aircraft, through an iterative 'tuning' process. Figure 2 - 4 describes the information flow for the cascaded filter.

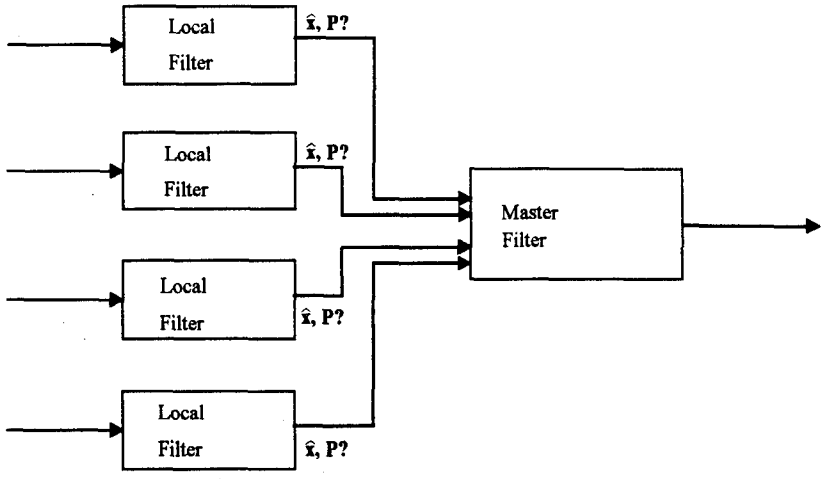


Figure 2 - 4 Cascaded Filter Navigation System

An important consideration in distributed filters is internal fault tolerance. Consider a cascaded system consisting of four local filters providing input to a master filter. Examining the reliability of the system based on component reliability gives insight into performance with filter internal failures. If each filter has a reliability probability p_i (probability of not failing) over a given time period, the reliability of the system is denoted **r(p)**, where **p** is the vector of component probabilities made up of p_1, p_2,..,p_n, local filter reliabilities, and the reliability of the master filter, p_m [21]. The probability of failure of the system will depend on the network structure. This could be grouped into 3 cases of distributed Kalman filter systems:

Case 1: *All filters are required for continued operation.* Mathematically, this is the same as the reliability of a system composed of a series of components, the failure of any which will make the system fail. In this case:

$$\mathbf{r(p)} = [\prod_{i=1}^{n} p_i] * p_m \qquad (2\text{-}23)$$

For example, if $p_{i,m} = .9500$ and $n = 4$, then $\mathbf{r(p)} = 0.7738$.

Case 2: *One local filter plus the master filter are required for continued operation.* The reliability function is structured as a parallel system for the local filters, in series with the master filter, and is computed as follows:

$$r(\mathbf{p}) = \left[1 - \prod_{i=1}^{n}(1 - p_i)\right] * p_m \tag{2-24}$$

For example, if $p_{i,m} = .9500$ and $n = 4$, then $r(\mathbf{p}) = 0.9499$. Due to the high reliability of parallel local filters, this turns out to be approximately the reliability of the master filter.

Case 3: Any one filter is required for continued operation. The reliability function in this case is computed as the reliability of a system of parallel components, defined as:

$$r(\mathbf{p}) = 1 - \left[\prod_{i=1}^{n}(1 - p_i) * (1 - p_m)\right] \tag{2-25}$$

For example, if $p_{i,m} = .9500$ and $n = 4$, then $r(\mathbf{p}) = 0.9999$.

Thus, internal fault tolerance may be enhanced or reduced, depending on the requirements of the distributed filter design. Note that this applies to cascaded filters as well as federated filters.

2.5 Federated Filters

2.5.1 Federated Filter Theory

The foundation of the federated filter is an information sharing method used to build a total solution from subcomponent filters. Each sensor in the navigation system has a dedicated local filter to which it provides information at a rate which may be different than the other sensors. As well, INS information is directed to each local filter and the master filter. The information-sharing methodology is [8]:

- Divide the total system information among several component filters.
- Perform local time propagation and measurement processing, updating with local sensor information when available.
- Recombine the updated local information into a new total sum.

Suppose there is a federated filter consisting of n local filters and a master filter, totaling k = n+1 filters. Let the full, centralized filter solution be represented by the covariance matrix $\mathbf{P}_f$ and state vector $\hat{\mathbf{x}}_f$, and the kth local filter solution by $\mathbf{P}_k$ and $\hat{\mathbf{x}}_k$, and the master filter solution by $\mathbf{P}_m$

and $\hat{\mathbf{x}}_m$. If the measurement errors from each of the sensors are statistically independent, they can be optimally combined as follows [9]:

$$\mathbf{P}_f^{-1} = \mathbf{P}_m^{-1} + \mathbf{P}_1^{-1} + \mathbf{P}_2^{-1} + \ldots + \mathbf{P}_n^{-1} \tag{2-26}$$

$$\mathbf{P}_f^{-1}\hat{\mathbf{x}}_f = \mathbf{P}_m^{-1}\hat{\mathbf{x}}_m + \mathbf{P}_1^{-1}\hat{\mathbf{x}}_1 + \mathbf{P}_2^{-1}\hat{\mathbf{x}}_2 + \ldots + \mathbf{P}_n^{-1}\hat{\mathbf{x}}_n \tag{2-27}$$

where $\mathbf{P}_k^{-1}$ is the information matrix for the k^{th} filter. Now, if we start with the full solution matrix $\mathbf{P}_f, \hat{\mathbf{x}}_f$, this solution can be divided so that the local filters and the master filter each receive fractions β_k of the total information:

$$\mathbf{P}_f^{-1} = \mathbf{P}_m^{-1} + \mathbf{P}_1^{-1} + \mathbf{P}_2^{-1} + \ldots + \mathbf{P}_n^{-1} = \mathbf{P}_f^{-1}\beta_m + \mathbf{P}_f^{-1}\beta_1 + \mathbf{P}_f^{-1}\beta_2 + \ldots + \mathbf{P}_f^{-1}\beta_n \tag{2-28}$$

To maintain constant total information across the sum in Equation 2-33, the share-fraction values must sum to unity:

$$\sum_{k=1}^{m,n} \beta_k = \beta_m + \sum_{k=1}^{n} \beta_k = 1 \tag{2-29}$$

So the LF and MF fractions can be recombined to yield the total correct solution $\mathbf{P}_f$, $\hat{\mathbf{x}}_f$.

Propagation of the covariance matrix for each filter is performed by each component filter independently, in parallel, with the covariance propagation equation (from Equation 2-8):

$$\mathbf{P}_k(t_{i+1}^-) = \Phi_k(t_{i+1}, t_i)\mathbf{P}_k(t_i^+)\Phi_k^T(t_{i+1}, t_i) + \mathbf{G}_k(t_i)\mathbf{Q}_k(t_i)\mathbf{G}_k^T(t_i) \tag{2-30}$$

Assume that the local filters and master filter are all of the same size. The state transition matrices Φ_k are equal to Φ_f, and the noise distribution matrices $\mathbf{G}_k$ equal $\mathbf{G}_f$. The process noise covariance matrices $\mathbf{Q}_k$ are governed by the information-sharing rule:

$$\mathbf{Q}_f^{-1} = \mathbf{Q}_m^{-1} + \mathbf{Q}_1^{-1} + \mathbf{Q}_2^{-1} + \ldots + \mathbf{Q}_n^{-1} \tag{2-31}$$

$$\mathbf{Q}_k^{-1} = \mathbf{Q}_f^{-1}\beta_k \text{ or } \mathbf{Q}_k = \mathbf{Q}_f\beta_k^{-1} \tag{2-32}$$

Now, if we have all the components $\mathbf{P}_k$ and $\mathbf{Q}_k$, we can propagate the solution component-wise and form the solution $\mathbf{P}_f$:

$$\mathbf{P}_f^{-1}(t_{i+1}) = \sum_{k=1}^{m,n} \mathbf{P}_k^{-1}(t_{i+1}) = \sum_k \left[\Phi_f \mathbf{P}_f(t_i)\beta_k^{-1}\Phi_f^T + \mathbf{G}_f\mathbf{Q}_f\beta_k^{-1}\mathbf{G}_f^T\right]^{-1} \tag{2-33}$$

Regrouping terms, we can derive the following equation:

$$\mathbf{P}_f^{-1}(t_{i+1}) = \left[\sum_{k=1}^{n,m}\beta_k\right]\left[\Phi_f \mathbf{P}_f(t_i)\Phi_f^T + \mathbf{G}_f\mathbf{Q}_f\mathbf{G}_f^T\right]^{-1} \tag{2-34}$$

where $\sum_{k=1}^{n,m}\beta_k = 1$.

For measurement updates, each local filter incorporates discrete measurements $\tilde{\mathbf{z}}_k$ from the kth local filter. Measurement information is added to local filter k using the Kalman filter Equations 2-9 to 2-11:

$$\mathbf{P}_{ki}^{-1(+)} = \mathbf{P}_k^{-1} + \mathbf{H}_k\mathbf{R}_k^{-1}\mathbf{H}_k^T \tag{2-35}$$

$$\mathbf{P}_k^{-1(+)}\hat{\mathbf{x}}_k^+ = \mathbf{P}_k^{-1}\hat{\mathbf{x}}_k + \mathbf{H}_k\mathbf{R}_k^{-1}\tilde{\mathbf{z}}_k \tag{2-36}$$

This employs Equations 2-9 to 2-11, where the superscript + refers to post-measurement values. The fusion algorithm (Equations 2-31 and 2-32) can then be used to find the total solution. A representative federated filter navigation system is shown in Figure 2 - 5.

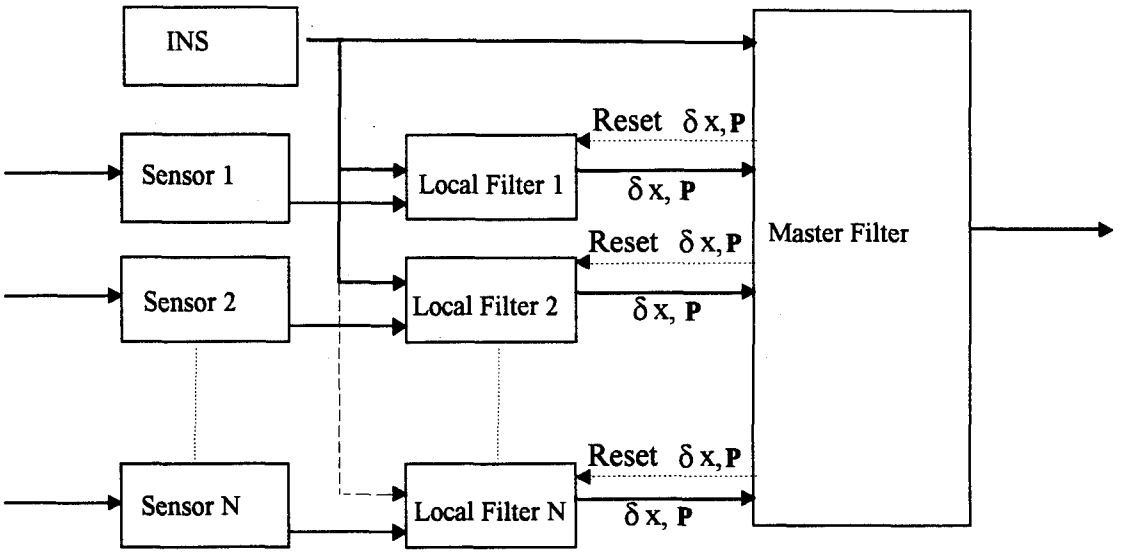


Figure 2 - 5 General Federated Filter Architecture

Thus, the federated filter solution will provide the same estimates as that of a single, centralized Kalman filter, and is globally optimal, when certain assumptions are satisfied:

- Each filter employs a single β_k value for all of the full-system states and process noises.
- Equations 2 - 28 and 2 - 32 are valid.
- The information fusion and reset (dividing) operations are performed after every measurement cycle (to be explained more fully in Section 2.5.3).
- All filters have the same state-space formulation for the INS states.

Some deviations from these assumptions can be made, with small loss of optimality [9]. First, the federated filter can be implemented such that the local filters are of minimum size, each local filter (filter k) containing only the common INS states and the states unique to the kth sensor. Also, the matrices $\mathbf{P}_k$, $\mathbf{\Phi}_k$, $\mathbf{Q}_k$, and $\mathbf{G}_k$ contain only the common INS state and k^{th} sensor bias state partitions of the full matrices. The β_k fraction values are presumed to apply only to the common INS states, since only those states are shared among the local filters and the master filter.

2.5.2 Advantages of the Federated Filter

The federated filter is designed to provide a weighted least-squares solution to the estimation problem [22]. It is intended to overcome the following problems of the centralized filter:

- Heavy computation loads due to a large number of states.
- Poor fault tolerance in terms of detecting gradual sensor faults.
- Slow regeneration a solution after a failure.

The federated filter also does not violate the white Gaussian noise assumption for measurement noise, as does the simplest form of the cascaded filter. In addition, the federated filter retains covariance information, unlike the cascaded filter.

With a centralized filter, the number of states can grow large, since all INS plus sensor-specific states are required. An approximation for computation time for the Kalman filter [5] is given by:

$$\text{Load} = \lambda\left[n^3 + \left(\sum m\right)n^2\right] \tag{2-37}$$

where n is the total number of states, $\sum m$ is the total number of measurements, and λ is a proportionality factor. For example, if there were a navigation system with 9 INS states, and three sensor measurements, each sensor adding two bias states (and letting $\lambda = 1$), the total computational load for a centralized filter would be:

$$\text{Load} = 15^3 + (3)15^2 = 4050 \tag{2-38}$$

Compare this to the computational load of any of the local filters (9+2 states):

$$\text{Load} = 11^3 + (1)11^2 = 1452 \tag{2-39}$$

Thus, the additional states required in the centralized filter causes about a threefold increase in computational load. Generally, the computations required by the centralized filter are decreased by

a factor equal to the number of local filters in the design. Note, however, that the federated filter will require additional processors, one for each filter.

The centralized filter may have difficulty detecting slow-onset or "soft" failures. In addition, if faulty sensor information is incorporated into the filter, the full solution (means and covariances) may be corrupted. The federated filter's distributed nature makes it able to compare solutions from each local filter, and sensor failures are more easily detected. Also, the errors will not corrupt the master filter solution, if detected [14].

The federated master filter is not susceptible to stability problems caused by filter driving filters as in the cascaded Kalman filter. The master filter does not treat the inputs from the local filters as measurements, but as fractional components of a whole solution, so the federated filter does not violate the assumptions of the Kalman filter.

2.5.3 Federated Filter Configurations

There are currently four primary federated filter implementations, each using a different information sharing method. In each case the local filters receive sensor measurement and reference system information from the INS. The master filter provides the INS corrections and the reset information to the local filters, while combining the information provided by the filters into a globally optimal navigation solution.

For each of the reset modes, the time propagation and measurement update steps are essentially the same. During the propagation cycles, each of the local filters multiplies its common process noise variances by the information sharing fractions in order to split up the whole process noise information between them. During measurement update cycles, the local filters perform normal processing of the data from their independent sensors. Figure 2 - 6 shows a generic federated filter, describing information flow among filters. The federated filter modes of operation are: zero reset mode, partial reset mode, no reset mode, and full reset mode.

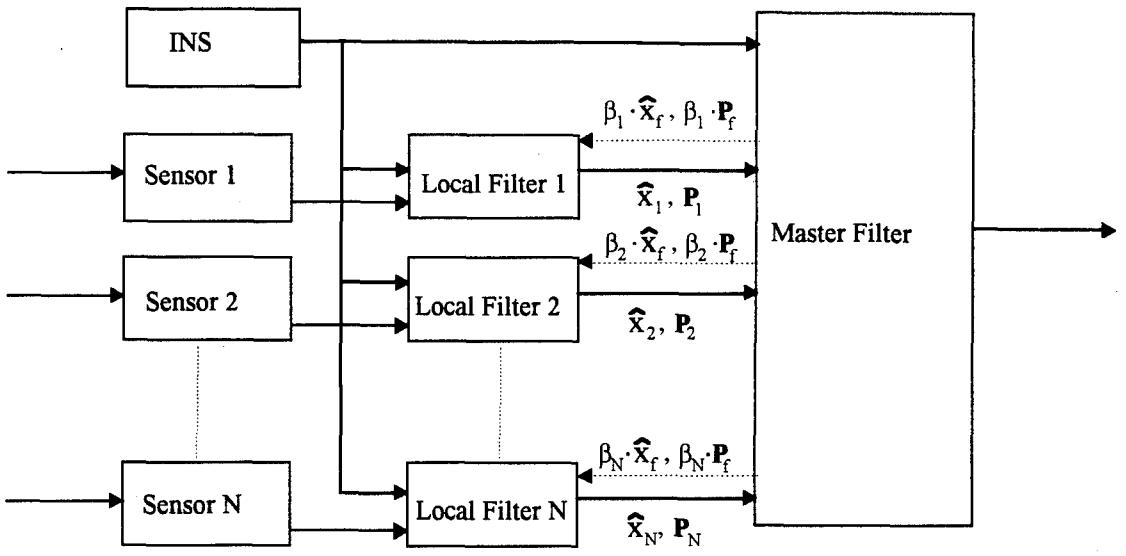


Figure 2 - 6 Federated Filter Resets

2.5.3.1 Zero reset mode

In this mode, the master filter retains all of the system long-term memory and the local filters act as data compression filters with short-term memory only. There is no feedback of the fused solutions to the local filters. Instead, the local filter is given the command to reset to zero information after each fusion update, resulting in an infinite covariance matrix, or a zero information matrix for each local filter.

The local filters can be based upon relatively low order INS and sensor models. Consequently, data bus loads are reduced.

2.5.3.2 Partial reset mode

The master filter and the local filters share the system long-term memory. This design involves feedback of only a fractional component ($\beta_k \mathbf{P}_f$) of the full-fused solution to the local filter. The master filter would benefit from having higher order system models than the local filters, thereby allowing for improved fault detection (as compared to zero reset mode) because the sensor data is treated independently.

2.5.3.3 No Reset mode

The local filters collectively retain all of the local information, which is passed to the master filter at fusion. The master filter retains the fused solution to propagate forward in time, to provide estimates when required, but this fused solution is discarded at the next fusion. This method is similar to the partial-reset mode, except that the master filter solution may be propagated but does not participate in the next fusion update. This no-reset design is highly fault tolerant, since poor measurement information from one sensor would not affect any of the other local filters. The no reset mode also provides the best overall performance for FDI because the local filters operate independently of each other, and so individual solutions may be compared, giving an opportunity for fault detection and an indication of which sensor is faulty. Theoretically, however, it is sub-optimal, since the fused solution information is not fed back to the local filters, and so correlation information between filters is lost.

2.5.3.4 Full-reset mode

In full-reset mode, feedback of the fused solutions to the local filters is accomplished. The long-term memory resides wholly in the local filters. The master filter solution is propagated in the master filter from time of fusion until the next fusion time, when it is discarded, and the new fused solution is formed and fed back to the local filters.

2.6 EGI Retrofit Design

2.6.1 General

A design to retrofit the EGI into the navigation system would have the following characteristics:

- High reliability and accuracy.
- Ease of implementation.

Design alternatives to satisfy these characteristics are presented in this section, along with supporting theory.

2.6.2 Alternative Designs

A number of designs were formulated. From these designs, two were selected that satisfied the requirements. These two designs were constructed in simulation and compared in terms of performance and fault tolerance. Both designs are indirect feedforward configurations, but also could have been configured as feedback configurations. A problem with the feedforward configuration is that the INS is free to drift. When this happens, assumptions of small errors in the INS, used in dynamics calculations, are violated. This problem is circumvented by monitoring the magnitude of the INS errors and resetting them to zero, based on the Kalman filter state estimates, when the estimates exceed a designated threshold.

Design A: EGI-based Navigation System

In this design, the existing navigation computer and reference INS is replaced by the EGI. Sensor data is routed to the EGI, and the EGI Kalman filter is augmented with the required states to operate as a centralized filter. Figure 2 - 7 shows this design.

Design B: EGI Transformation Navigation System

In this implementation, the theory of the federated filter is invoked. The output of the EGI (both state estimates and covariance matrix) is considered the output of a local filter, as defined in federated filter parlance. Other sensor information is directed to individual local filters, one per sensor, and the master filter is implemented in the original navigation computer. In this way, the

reference INS and the navigation computer need not be removed to add in the EGI. Figure 2 - 8 shows this design.

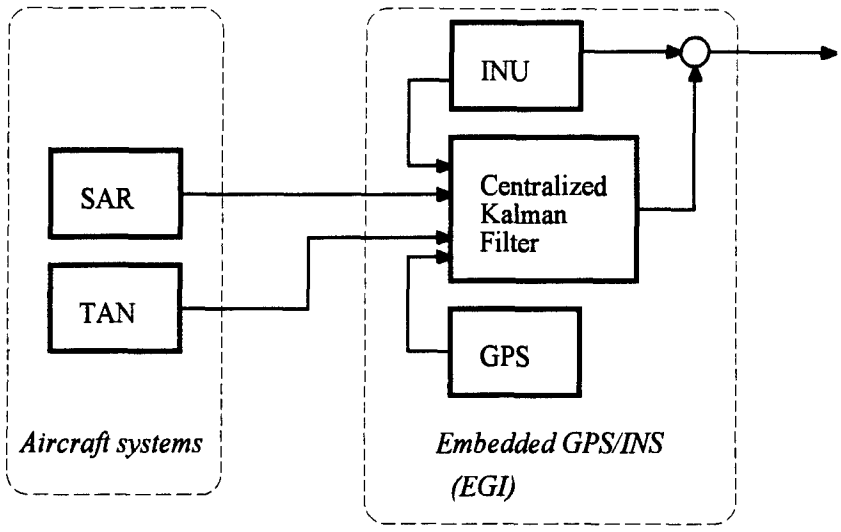


Figure 2 - 7 Design A (EGI-Based Design)

Again, note that Design A uses one INS, while Design B uses two INS, and so both designs are fed different information. Also, Design B's use of the federated filter technique is a departure from the federated filter theory stated in Section 2.5. For these reasons, the design comparison should not be considered one of centralized filter versus federated filter, but simply a comparison between possible navigation system designs. For a better, 'apples versus apples' comparison, Design B would have to be compared to a reduced-order centralized Kalman filter design, containing error states for both INS plus the additional states for sensor biases.

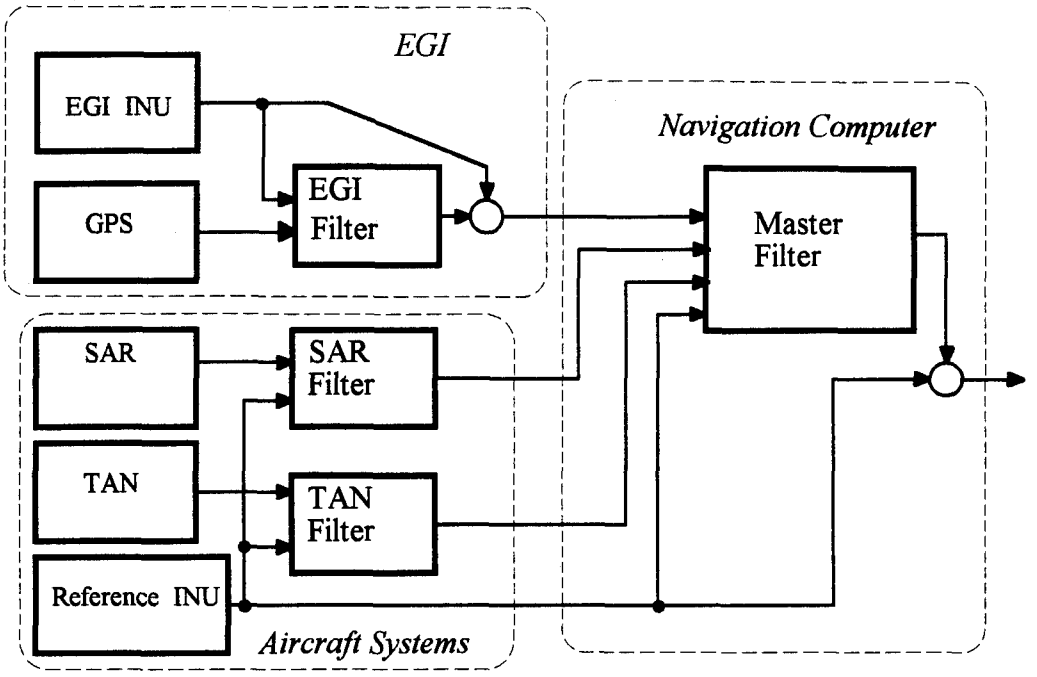


Figure 2 - 8 Design B (Federated Filter Design)

2.6.3 Information Sharing with Two INSs

A problem with information transfer between the EGI local filter and the master filter becomes obvious. INS error state estimates created by the EGI Kalman filter are error states for the EGI INS, not the reference INS. Second, the covariance matrix in the EGI Kalman filter contains the second-order statistics for the EGI INS error-state covariances, the GPS covariances, and the cross-covariances. In the cascaded filter approach, estimates of position and velocity (and possibly others) from the EGI filter could be passed to the navigation computer as measurements, but much of the information contained in the EGI Kalman filter is not passed. How can the full information of states and covariances in the EGI be used in a larger navigation system?

One technique incorporating the federated filter is to transform the EGI filter information contained in $\hat{\mathbf{x}}, \mathbf{P}$ from using the EGI INS as a reference to using the existing navigation system INS as a reference.

Start with the basic equation:

$$\mathbf{x}(t) = \mathbf{x}_{ref}(t) + \delta\mathbf{x}_{ref}(t) \tag{2-40}$$

where:

$\mathbf{x}(t)$ is the position, velocity and tilt (PVT) of the aircraft as a function of time

$\mathbf{x}_{ref}(t)$ is the reference INS-determined values of PVT

$\delta\mathbf{x}_{ref}(t)$ is the error in the reference INS PVT

Equation 2-45 would be valid for the EGI INS in the same aircraft:

$$\mathbf{x}(t) = \mathbf{x}_{egi}(t) + \delta\mathbf{x}_{egi}(t) \tag{2-41}$$

where:

$\mathbf{x}_{egi}(t)$ is the EGI INS-determined values of PVT

$\delta\mathbf{x}_{egi}(t)$ is the error in the EGI INS PVT

To convert from one position error to another, the RHS of Equations 2-45 and 2-46 are equated, to yield:

$$\delta\mathbf{x}_{ref}(t) = \delta\mathbf{x}_{egi}(t) + [\mathbf{x}_{egi}(t) - \mathbf{x}_{ref}(t)] = \delta\mathbf{x}_{egi}(t) + \mathbf{y}(t) \tag{2-42}$$

where:

$\mathbf{y}(t)$ is the vector difference between the two INS outputs as a function of time

During the operation of the filter, the values of $\mathbf{x}_{ref}(t)$ (current reference INS position) and $\mathbf{x}_{egi}(t)$ (current EGI INS position) are available, but the actual reference errors are not available. Since the realizations of $\mathbf{y}(t)$ are available for all required calculations, $\mathbf{y}(t)$ *can be considered to be a constant for any time t.*

The Kalman filter estimates are considered to be distributed as jointly Gaussian random variables. Since a linear relationship exists between $\delta\mathbf{x}_{egi}(t)$ and $\delta\mathbf{x}_{ref}(t)$, the state estimate (first moment) is shifted by $\mathbf{y}(t)$, but the covariance matrix (second moment) remains the same:

$$\delta\hat{\mathbf{x}}_{ref}(t) = \delta\hat{\mathbf{x}}_{egi}(t) + \mathbf{y}(t) \tag{2-43}$$

$$\mathbf{P}_{ref}(t) = \mathbf{P}_{egi}(t) \tag{2-44}$$

The result is a transformation for position, velocity and tilt information from the EGI INS to the reference INS, which may then be used as local filter information in the federated filter design. Note that this is not strictly in accordance with the federated filter theory. However, it presents a workable method of using the EGI information in a larger navigation system.

2.6.4 Federated Filter Design

The mechanics of this transformation as given in Equation 2-43 is shown in Figure 2 - 9. Note that part of the differencing operation [use of $\mathbf{y}(t)$] is already conducted within the EGI.

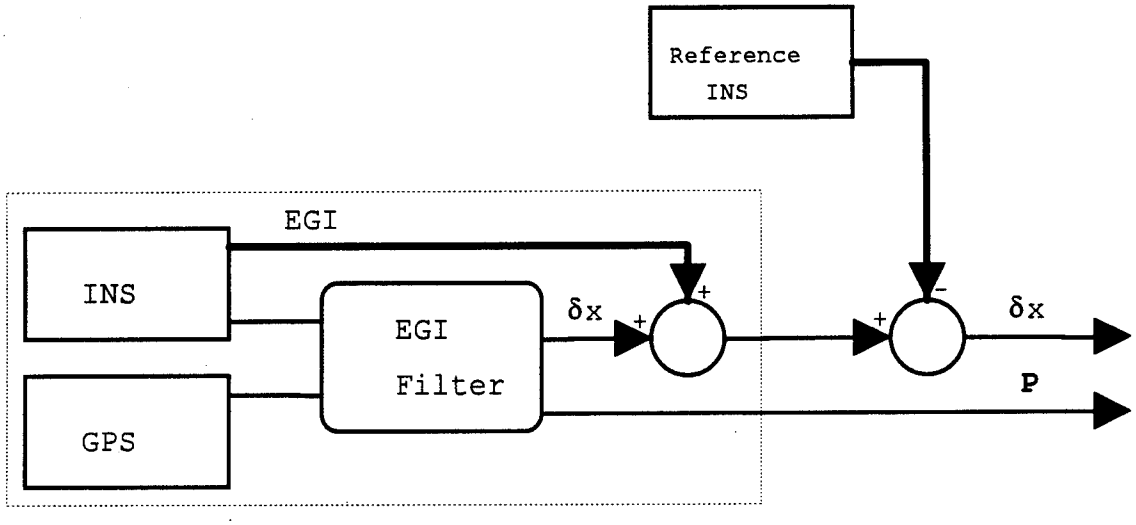


Figure 2 - 9 EGI INS to Reference INS Transformation Mechanization

The full implementation is shown in Figure 2 - 10.

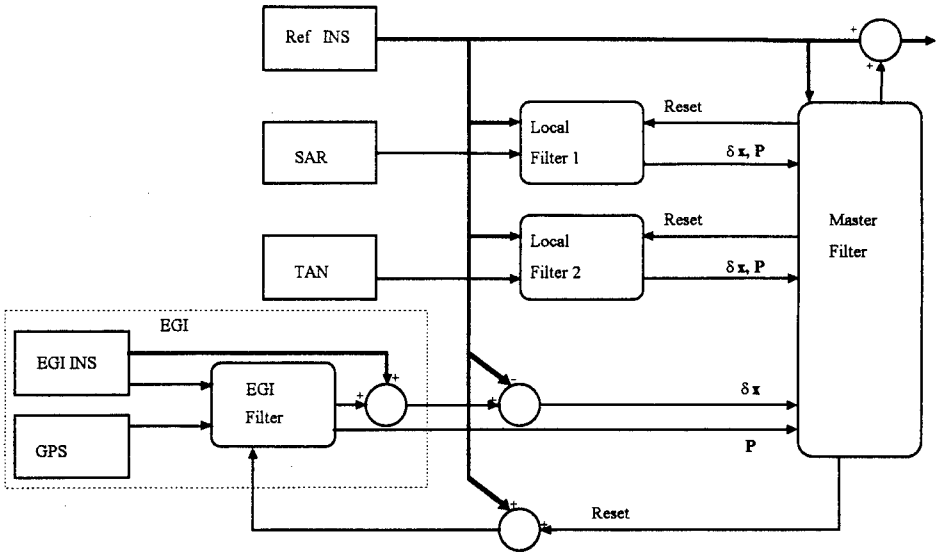


Figure 2 - 10 Federated Filter with EGI

Although this provides a method of using EGI information in the federated filter, it does not form a type of federated filter that is optimal, since some information is lost. This lost information can be shown by comparison between the proposed federated filter implementation and a full-order (states for both INS and all sensors) centralized Kalman filter. Although the full-order centralized filter is not used as a comparison design for this problem, it serves as a benchmark for optimal estimation and gives insight into the proposed design.

The centralized Kalman filter for the total suite of sensors would include separate states for two INSs, plus sensor-specific bias states. The covariance matrix for the centralized Kalman filter would contain covariances of all these states plus cross-covariances. Comparing this to the federated filter design, there is a major structure difference in that the cross-covariance information between the two INS states is not generated in the federated filter design. If the two INS have uncorrelated errors, then these cross-covariances would be zero, and there would be no loss of information by the transformation to a single set of INS states as calculated in the federated filter. On the other hand, if the two INS measurements are somehow correlated, then there would be a loss of information.

The question is: are the INS output errors uncorrelated? One way to show uncorrelatedness would be to show independence. It is not clear whether or not the errors in two INSs in the same aircraft would be independent or not. Similar sensors, in proximity to one another, subject to the same vibrations, etc. may very well show error correlation. It would be

difficult to tell whether or not meaningful cross-correlations between INS error states existed, and how much information would be lost if these cross-correlations were neglected. Due to the independent truth models in DKFSIM, this could only be modelled by building a dual-INS full-order truth model, and is beyond the scope of this thesis.

2.7 Summary

In this chapter, the necessary theory for design of the federated filter navigation system was developed. This included navigation subsystems explanations, presentation of the existing Kalman filter and federated filter theory. Following this, a new transformation for dual-INS integration in a navigation system, and a new navigation system design, were developed.

3. Filter Models

3.1 Introduction

This chapter deals with the modeling process used to compare the two configurations: the centralized filter design and the federated filter design. The first section describes the modeling package used, the Distributed Kalman Filter Simulator (DKFSIM 3.3). The next section highlights the modifications made to DKFSIM in the version used (DKFSIM 3.P1). The last section explains the centralized filter and federated filter models used in the simulations.

3.2 DKF Simulator (DKFSIM) 3.3

3.2.1 Description

The Distributed Kalman Filter Simulator (DKFSIM) Version 3.3 [11,22] is a computer simulation package written by Dr. Neal Carlson of Integrity Systems Inc., under contract for Wright Laboratories. It was developed "to support performance evaluation of Distributed Kalman Filter (DKF) techniques for multisensor navigation systems in a simulated flight hardware/software environment". All computer programs, data, and manuals for the installation and operation of DKFSIM are available from the Avionics Directorate (WL/AAAI), Wright Laboratories, 2185 Avionics Circle, Wright-Patterson AFB OH 45433, for appropriate users.

DKFSIM will allow the user to model and run in Monte Carlo simulation a navigation system comprised of a number of separate and independent Kalman filters. The current implementation allows three local filters and one master filter for a maximum of four filters in the navigation system. These filters may be 'networked' in a number of ways, and can be used to model:

- A centralized Kalman filter, with a selectable number of INS and sensor-specific states. This may be done by using only one of the local filter models and directing all INS and sensor information to that model; or
- A federated filter in any one of the modes outlined previously in Chapter 2. From two to four filter models may be used to build the desired federated filter.

DKFSIM will provide outputs of all filter states and covariances, plus other variables, in a digital file format. DKFSIM is written to provide a ready-made, easily reconfigurable package to

evaluate many types of systems. In addition, since the source code is provided, with detailed comments, additional configurations are available through modification of the existing source code.

DKFSIM will allow the user to incorporate flight profiles generated by PROFGEN, a trajectory generation system developed by Mr. Stanton Musick at Wright Laboratories [23], with user control of the operation of truth and filter models through the use of input data files. It incorporates the ability to simulate the sensor measurements of a strapdown INS, a GPS receiver which can provide data from up to four satellites, a synthetic aperture radar providing landmark ranging and precision velocity information, radar altimeter information, barometric altimeter information, and terrain-map matching information.

DKFSIM contains the software to accomplish the two major operations for simulation: generate the simulated sensor and INS, or 'truth' data, and operate the navigation filter or filter network.

The operation of the filters portion of DKFSIM is purposely designed to resemble the operation of a modern navigation computer. Sensor data are sent as messages on a data bus to the filter or system of filters. This bus structure, resembling a MIL-STD 1553B databus, allows the user to control data exchange between the simulated sensors and the navigation system. Overall filter operation is controlled by a navigation computer. Data from the filter model is sent, if required, on a data bus to the sensors. Figure 3 - 1 shows a general description of information flow in DKFSIM.

3.2.2 Architecture

DKFSIM is implemented in FORTRAN 77, including some code unique to Lahey FORTRAN. The FORTRAN code is divided functionally into programs and subroutines; for example, each truth model is implemented as a separate program. To run DKFSIM, all programs and subroutines must be compiled separately to produce object code files for each source code file. All object code files, as well as the required libraries, must then be linked to form an executable program. The executable program may then be run, with the appropriate input data and parameters.

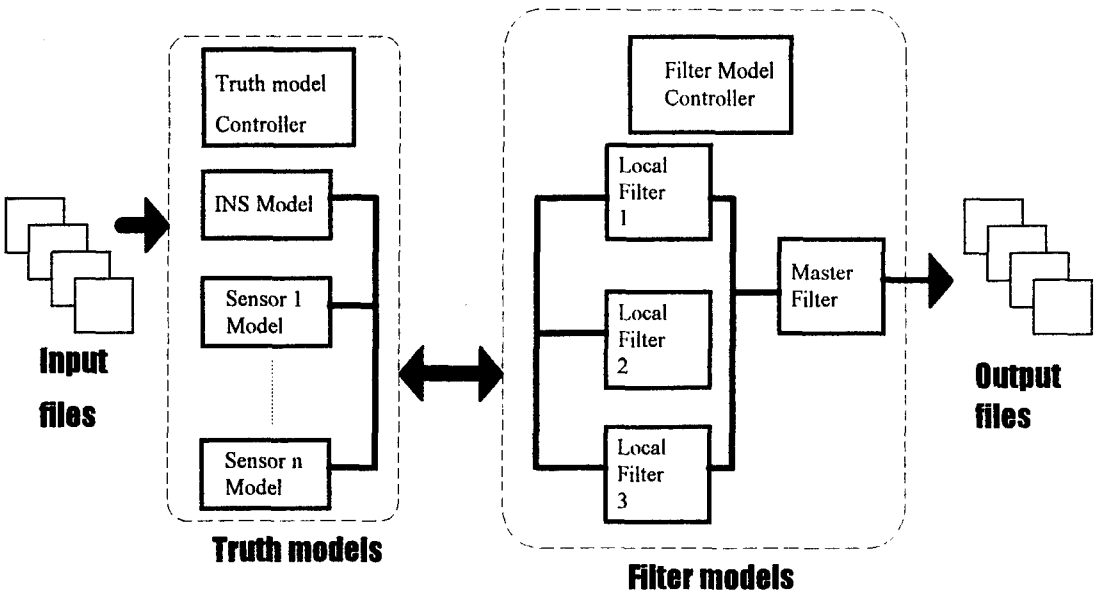


Figure 3 - 1 DKFSIM General Description

To modify DKFSIM beyond the range of built-in configurations and constraints, the source code file may be changed. Only that source code file need be recompiled to produce a modified object code file. The modified object code file may then be linked with the existing object code files to produce an executable program. Thus, the entire DKFSIM source code need not be recompiled at every source code change.

DKFSIM emulates real-time system operation while running in non-real time. This is accomplished by maintaining an appropriate time variable and stepping through all the relevant sequences in the program corresponding to actions occurring in a real-time system.

DKFSIM contains two major functional modules, the DKF Models module and the DKF Filter module. The DKF Models module operates truth models for the simulations, which in turn provide sensor data for the DKF Filter module. DKFSIM can be operated either in an integrated operation or in a separated operation. In the integrated operation, data created by the DKF Models module is passed as time-tagged bus messages to the DKF Filter module during the operation of the program. In the separated operation, the two modules are operated independently. The DKF Models module produces a data file of the sensor data created by the truth models, and the DKF Filters module may then be run, using as input the data file created by DKF Models module. The only limitation in operation is that reset of the INS, involving data return from the DKF Filters module to the DKF Models module, is not possible during separated mode operation. Figure 3 - 2

shows the DKFSIM structure. Each block represents a major FORTRAN program. The descriptive name for each program is given, as well as the computer filename. If running in separated mode, the program DKFMOD controls the operation of the DKF Models module, and the program DKFFIL controls the operation of the DKF Filters module. In the combined mode of operation, the program DKFSIM controls the operation of both the DKF Models and DKF Filters modules, and both DKFMOD and DKFFIL are inactive.

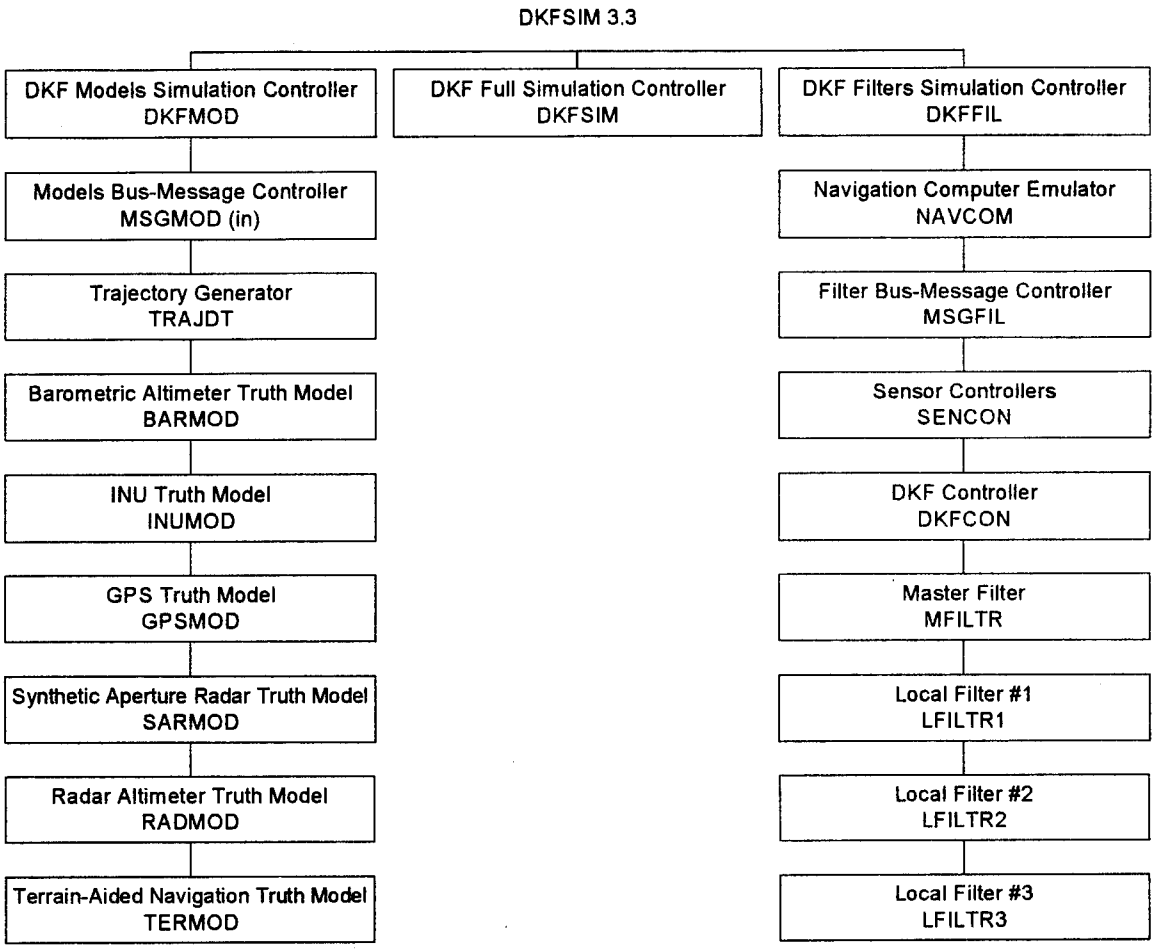


Figure 3 - 2 DKF Models Module and DKF Filters Module

3.2.3 Trajectory Generator

In order for the DKF Models module to produce outputs simulating aircraft flight, data regarding the flight trajectory of an aircraft is required [11]. Trajectory generation in DKFSIM is accomplished as a two-part process: first, the flight profile is generated using a profile generator

program; then the program TRAJDT in DKFSIM uses this flight profile to generate the required data. These steps are explained in further detail.

The Flight Profile Generator (PROFGEN) program developed produces a six degree-of-freedom flight profile with user-defined dynamic characteristics that can be tuned to represent an advanced tactical fighter, bomber, missile, or other air vehicle. Inputs to PROFGEN consist of an input file containing information on the desired profile as a sequence of segments, where each segment corresponds to a flight maneuver.

The primary output from PROFGEN is a data file containing sequential, time-tagged records of user-specified dynamic variables, and is called a flight file. The frequency of data output is also user-specified, except during highly dynamic maneuvers, when additional points are output to ensure adequate information is available for modeling.

The flight file produced by PROFGEN is then read into DKFSIM using the program TRAJDT. The program TRAJDT is called when dynamic variables from a particular time are needed for modeling. TRAJDT reads in the appropriate segment of data, using records with times before and after the time of interest, and interpolates the records to provide the variables required at the time of interest. These variables are then passed to the DKFSIM truth models.

3.2.4 Truth models

The DKF Models module incorporates six truth models: the INS, GPS, SAR, TAN, Barometric Altimeter (BARALT) and Radar Altimeter (RADALT) truth models. Each of these six modules produces different data, but share a common purpose - to provide the DKF Filters module with high-quality, realistic data, in order that high fidelity simulations of real-world missions may be accomplished.

3.2.5 INS truth model

The INS truth model is designed to simulate the outputs of a medium accuracy strapdown RLG INS. It incorporates the dominant instrument and environmental error sources of such a system. The parameter values of the model can be selected to represent strapdown INSs of different accuracy levels.

3.2.5.1 Description

The INS truth model implements an error-state formulation of the strapdown inertial navigation equations, mechanized with the ECEF frame as the computational reference frame. The ECEF frame was selected for internal computations rather than the more commonly-used wander-navigation frame for the following reasons:

- The linearized error-state propagation equations are far simpler in the ECEF frame.
- There are no singularities in the ECEF frame.

The basic computation procedure involves propagation of the true whole-valued solution **x** (true trajectory), parallel propagation of the INS solution error $\delta\mathbf{x}$, and generation of the INS -indicated whole-value solution as the sum $\mathbf{x}+\delta\mathbf{x}$ as indicated in Tables 3 - 1 to 3 - 3:

$\mathbf{p}(t)$	true position vector
$\mathbf{v}(t)$	true velocity vector
$\mathbf{C}(t)$	true attitude direction cosine matrix
$\mathbf{f}(t)$	true specific force vector
$\mathbf{w}(t)$	true angular velocity vector

Table 3 - 1 True Trajectory Variables

$\delta\mathbf{p}(t)$	true position error vector
$\delta\mathbf{v}(t)$	true velocity error vector
$\delta\boldsymbol{\theta}(t)$	true attitude error vector
$\delta\mathbf{f}(t)$	true specific force error vector
$\delta\mathbf{w}(t)$	true angular velocity error vector

Table 3 - 2 INS Solution Errors

$\mathbf{p}_{ins}(t) = \mathbf{p}(t) + \delta\mathbf{p}(t)$	INS indicated position vector
$\mathbf{v}_{ins}(t) = \mathbf{v}(t) + \delta\mathbf{v}(t)$	INS indicated velocity vector
$\mathbf{C}_{ins}(t) = [\mathbf{I} + \delta\boldsymbol{\theta}(t)\mathbf{x}]\mathbf{C}(t)$	INS indicated attitude direction cosine matrix
$\mathbf{f}_{ins}(t) = \mathbf{f}(t) + \delta\mathbf{f}(t)$	INS indicated measured force vector
$\mathbf{w}_{ins}(t) = \mathbf{w}(t) + \delta\mathbf{w}(t)$	INS indicated angular velocity vector

Table 3 - 3 INS-Indicated Solution

The operation of the model is summarized by the following steps:

1. The true solution is computed as a function of time by the trajectory program TRAJDT.

2. The INS truth model generates the INS errors $\delta\mathbf{x}(t)$, by integration of the corresponding error differential equations.

3. The INS model then outputs the INS-indicated solution $\mathbf{x}_{ins}(t) = \mathbf{x}(t) + \delta\mathbf{x}(t)$.

A 48-state truth model to propagate error dynamics was developed for DKFSIM [11]. This model has been used in a number of programs, and refined over time to provide a high-fidelity reproduction of an actual strapdown RLG INS. Error states are represented by the mathematical processes of random constants, first-order Markov processes, or second-order Markov processes, as required (refer to [5] for a description of Markov processes).

Table 3 - 4 describes the 48 states for the INS truth model. Along with the states are listed the mathematical representation of each state.

INS Truth Model States	INS Truth Model Representation	Model Parameters
3 Position drifts	Linearized propagation driven by velocity drifts	Started from initial errors
3 Velocity drifts	Linearized propagation driven by acceleration drifts	Started from initial errors
3 Attitude drifts	Linearized propagation driven by angular rate errors	Started from initial errors
3 Body (case) misalignment errors	Random constants (independent)	Body frame
3 Gravity perturbations	First-order Markov (independent, spatially correlated)	Body frame
3 Accelerometer Biases	First-order Markov (independent, time-correlated)	Body frame
3 Accelerometer Scale Factor Errors	Random constants (independent)	Body frame
3 Accelerometer misalignments	Random constants (independent)	Body frame
3 Accelerometer wideband noises	Random walk (independent)	Body frame
3 Gyro bias drift rates	First-order Markov (independent, time-correlated)	Body frame
3 Gyro scale factor errors	Random constants (independent)	Body frame
6 Gyro input-axis misalignments	Random constants (independent)	Body frame
6 Gyro acceleration-sensitive drift coefficients	Random constants (independent)	Body frame
3 Gyro wideband noises	Random walk (independent)	Body frame
48 States		

Table 3 - 4 INS Truth Model Error States

3.2.5.2 Failure Models

In addition to simulating the normal operation of an INU, the INU model incorporates a failure model into the INU model software, allowing the user to use predetermined failures in

simulation and study the results. These failures are controlled by simple changes to the model input file in DKFSIM. The strapdown RLG INS failure model augments the INS truth model by adding abnormal errors to the calculated error state $\delta\mathbf{x}(t)$. The types of INS failures that can be modeled are listed in Table 3 - 5.

Failure Type	INS Truth Model Representation	Coordinate Frame
Accelerometer Failure Types		
Accelerometer bias failure (1 of 3 axes)	Bias shift plus ramp	Body frame
Accelerometer noise failure	Random noise sigma shift plus ramp	Body Frame
Rate Gyro Failure Types		
Gyro rate bias failure	Bias shift plus ramp	Body frame
Gyro rate noise failure	Random noise sigma shift plus ramp	Body frame

Table 3 - 5 INS Failure Models

An example of a failure is given for illustration. Suppose a gradual failure of one of the accelerometers occurs, characterized by a slowly increasing bias. To simulate the failure, the user supplies a scalar value to set the 'slope' of the ramp, i.e. how quickly the ramp will increase or decrease in time. The accelerometer bias failure model calculates, based on user-supplied parameters, a bias as a function of time. Note that a bias shift is also available for use, but since a shift is not desired, the input value is set to zero. The failure model-calculated accelerometer bias (see Table 3 - 5) is added to the truth model-calculated accelerometer bias to produce the failure value for accelerometer bias.

3.2.6 GPS truth model

3.2.6.1 Introduction

The model for the GPS satellites consists of a set of 24 satellites in circular orbits with the nominal parameters of the actual satellite orbits [11]. There are three orbital planes, with ascending nodes separated by 120 degree intervals (ascending node is where the orbit passes upward through the earth's orbital plane). Each orbital plane has an inclination of 63 degrees, and contains eight satellites in essentially the same orbit, spaced at 45 degree intervals. All of the satellite orbits are nominally circular, with an orbital radius of 4.2 earth radii, and a 12 hour period.

3.2.6.2 Description

The GPS truth model implements an error-state formulation of the receiver pseudorange and pseudorange-rate, for four satellites only. The measured pseudorange is computed as the sum of the true geometric range, receiver clock phase error, and other range measurement errors. Table 3 - 6 shows the error states formulated in the GPS truth model.

GPS Truth Model States	Representation	Referred to
1 User clock phase bias error	Integral of user clock frequency bias error	All channels
4 Satellite range errors due to satellite position and clock phase drifts	State 1 of second-order Markov errors (independent)	Each channel
4 Ionospheric delay errors after differential frequency correction	State 1 of second-order Markov errors (independent)	Each channel
4 Tropospheric delay errors after compensation	State 1 of second-order Markov with $1/\sin(e_k)$ scale factor	Each channel
4 Receiver phase noise	Random sequence	Each channel
1 User clock frequency bias error	First-order Markov	All channels
1 User clock acceleration-sensitive frequency error	Random constant coefficients times specific force	All channels
4 Satellite range-rate errors due to satellite velocity and clock frequency drifts	State 2 of second-order Markov (independent)	Each channel
4 Ionospheric delay-rate errors	State 2 of second-order Markov (independent)	Each channel
4 Tropospheric delay-rate error	State 2 of second-order Markov (independent)	Each channel
4 Receiver frequency noise	Random sequence	Each channel
35 Total States		

Table 3 - 6 GPS Truth Model Error States

3.2.6.3 Failure Models

The following failure modes can be simulated with the DKFSIM GPS truth model:

1. Satellite clock failures
    - Frequency bias shift (constant)
    - Frequency bias ramp (growing with constant rate)
    - Frequency random noise shift (constant noise σ)
    - Frequency random noise ramp (growing σ)
2. Satellite ephemeris faults

- Radial position and velocity errors (initial values of sinusoidal oscillation)

3. Receiver clock failures

   - Frequency bias shift (constant)
   - Frequency bias ramp (growing with constant rate)
   - Frequency random noise shift (constant σ)
   - Frequency random noise ramp (growing σ)

This range of failure modes covers many GPS failures, such as bad ephemeris uploads, receiver clock frequency shifts, and satellite clock frequency shifts. In addition, the user has the option of turning off any number of satellites to provide a reduced set (<4) of satellite data, simulating terrain obstruction, or eliminating satellite reception altogether to simulate jamming.

3.2.7 SAR Truth Model

3.2.7.1 Introduction

The total SAR system uses two subsystems, the SAR Position Velocity Updating (PVU) and the SAR High Resolution Mapping (HRM) truth models. Each state is listed, along with the mathematical representation and the application. The SAR PVU and SAR HRM subsystems may operate independently in the DKF filter implementation of the SAR dedicated local filter. These two systems complement each other such that PVU measurements taken just prior to HRM measurements improve the update accuracy. The SAR HRM capability is useful because it bounds the growth of the INS position errors when GPS is not available, while the SAR PVU capability provides measurements for aircraft velocity.

3.2.7.2 Description

The SAR-PVU model operates in the following manner:

- Eight range-rate measurements are calculated simultaneously. In reality there would be some time lag between measurements, but this is a simplifying assumption and does not significantly impact operation of the truth model [11].

- The SAR-PVU error model generates measurement errors, based on user input parameters.
- The measurement errors are summed with the true values to create measurements, which are then passed to the DKF Filters module.

To generate measurement data for use by the DKF Filters module, the SAR-HRM truth model uses the following procedure:

- The user specifies in an input file how often the SAR-HRM will generate landmark positions, where these position are relative to the aircraft, and how long each landmark is in view.
- A math routine in the truth model, using this landmark, generates true values of range, range-rate, azimuth, and elevation.
- The true values are summed with the truth model error-state output, to create SAR-HRM measurements.
- The measurements are then passed to the DKF Filters module.

The error model states for both the SAR-PVU and SAR-HRM parts of the SAR truth model are listed in Table 3 - 7.

3.2.7.3 Failure Models

There is no separate failure model for the SAR truth model. However, the user can direct the model to limit the number of landmark positions available, or corrupt the measurements with very high noise values.

3.2.8 TAN Truth Model

3.2.8.1 Introduction

The TAN truth model generates altitude and horizontal position measurements through the use of two error models: a radar altimeter truth model and a terrain/map generator truth model. These are described in the following section, along with a description of error states in each model.

SAR Truth Model States	Representation	Application
SAR PVU		
3 Velocity measurement bias errors	First-order markovs (independent)	All components
3 Velocity scale factor errors	First-order markovs (independent)	All components
6 Mounting misalignment errors	Random constants (independent)	All components
3 Velocity measurement noises	Random sequences	All components
SAR HRM		
3 Landmark position bias components	First-order markovs, spatially correlated	Each landmark
1 Target designation error down-range component	Random sequence	Each landmark
1 Target designation error cross-range component	Random sequence	Each landmark
1 Range bias error	First-order Markov	All range measurements
1 Range scale factor error	Random constant	All range measurements
1 Range measurement noise	Random sequence	All range measurements
1 Range-rate bias error	First-order Markov	All range-rate measurements
1 Range-rate measurement noise	Random sequence	All range-rate measurements
1 Elevation bias error	First-order Markov	All elevation measurements
1 Elevation measurement noise	Random sequence	All elevation measurements
1 Azimuth bias error	First-order Markov	All azimuth measurements
1 Azimuth measurement noise	Random sequence	All azimuth measurements
29 Total States		

Table 3 - 7 SAR Truth Model Error States

3.2.8.2 Description

The terrain/map generator truth model consists of a terrain profile generator and a terrain elevation-map generator. Measurement data is generated as follows:

- The terrain profile generator produces a profile of terrain elevations and slopes, as well as points in an n x n planar grid centered below the aircraft.
- The terrain elevation-map generator adds errors to the true elevations in the n x n grid to obtain the map elevation values.
- The elevation values are used by the RADALT truth model to produce height above ground measurements. In addition, horizontal position can be measured by comparison of RADALT profile with the terrain-map profile.

The RADALT and terrain/map error model states are listed in Table 3 - 8. The states are listed with their mathematical representations and their parameters, for a 5-by-5 planar grid (the 5-by-5 grid provides a good compromise between accuracy and model size).

TAN Truth model states	**Representation**
Radar Altimeter	
1 Radar altimeter bias error	First-order Markov
1 Radar altimeter scale factor error	Random constant
1 Radar height measurement noise	Random sequence
Terrain-Map	
3 Map sector location errors	First-order Markov, spatially correlated
25 (1 Local elevation error at each grid point i,j)	Random sequence
31 Total States	

Table 3 - 8 TAN Truth Model Error States

3.2.9 Barometric Altimeter (BARALT) Truth Model

3.2.9.1 Introduction

Atmospheric pressure is a well-defined function of altitude. The Barometric Altimeter (BARALT) obtains altitude measurements by measuring the output of a pressure transducer. This measurement is subject to errors due to variations in pressure versus altitude. The BARALT truth model generates data which reproduces these noisy measurements.

3.2.9.2 Description

The BARALT truth model works in the following manner:

- True altitude data is obtained from the trajectory generator.
- Errors from the BARALT error model are added to the true altitude data, to create a BARALT measurement.
- The measurement is then sent to the DKF filters module.

If the INS truth model has a baro-damped vertical channel, measurements from the BARALT truth model are also sent to the INS truth model for vertical channel damping. Error states are listed in Table 3 - 9.

BARALT Truth Model States	**Representation**
1 Pressure altitude bias error	First-order Markov
1 BARALT scale factor error due to non-standard temperature	First-order Markov
1 Static pressure coefficient error	Random constant
1 BARALT time delay	Random constant
4 Total States	

Table 3 - 9 BARALT Truth Model Error States

3.2.10 Filter models

3.2.10.1 Introduction

The filter models used in DKFSIM for the centralized and federated filter designs are reduced-order implementations of the truth models for each system. The reduced order model still provides reasonable accuracy, and the order reduction keeps the computational requirement low as compared to a full-order model.

DKFSIM models the parallel operation of the federated filter by controlling operation of each filter individually by use of a program to simulate a navigation computer (NAVCOM). Filter operations which would be done in parallel on a number of processors are carried out in a sequential fashion, controlled by NAVCOM. For example, if propagation of the federated filter to the next measurement is required, NAVCOM ensures that all filters are propagated forward in time to the next measurement time. Careful time-tagging and task management allows this to be done in an efficient and realistic manner.

3.2.10.2 Description

Each filter model in the DKF Filters module has common features. For each sensor, except the INS, there is a measurement model describing the characteristics of each of the discrete measurements output by the sensor. For each sensor including the INS, there is a propagation model describing how the sensor states propagate from one time to the next. In addition, for some of the sensors, there are also measurement source models, describing how the position and velocity of the measurement source are determined.

There are a number of different sizes for INS states in each Kalman filter. In DKFSIM 3.3, the largest available is the so-called ABIAS model, which consists of 9 INS basic error states (east, north, and up states of position, velocity, and attitude), and three accelerometer bias states, plus other sensor bias states. Since the INS modelled is a RLG INS, the absence of gyro drift states does not significantly affect the accuracy of the reduced order model. The state vector $\mathbf{x}_f$ then, is [11]:

$$\mathbf{x}_f = \begin{bmatrix} \mathbf{x}_i & \text{INS error state} \\ \mathbf{x}_a & \text{Sensor A measurement - bias states} \\ \vdots & \\ \mathbf{x}_s & \text{Sensor S measurement - bias states} \end{bmatrix} \tag{3-1}$$

Sensor information may be directed to any of the local filters and/or the master filter, and those filters are increased in size to accommodate the sensor biases. For example, if GPS data is directed to Local Filter 1, then DKFSIM configures local filter 1 for 15 states - 9 PVT error states, 1 baro error state, 3 accelerometer bias error states, and two GPS error states.

The filter model states, representations and parameters are listed in Table 3 - 10. Note that if a filter were configured for all available INS error states, plus all measurements, it would be a 26-state filter.

FILTER MODEL STATE	REPRESENTATION	PARAMETERS
INS States - ABIAS Model		
3 Position drifts	Linearized propagation driven by velocity drifts	ECEF
3 Velocity drifts	Linearized propagation driven by acceleration errors	ECEF
3 Attitude drifts	Linearized propagation driven by	ECEF

	angular rate errors	
1 Barometric altimeter bias error	First-order Markov (independent)	
3 Accelerometer biases	First-order markovs (independent)	Body frame
13 Total INS Filter States		
GPS States		
1 User clock phase bias plus average satellite range error	random process	All channels
1 User clock frequency bias drift	random process	All channels
2 Total GPS Filter States		
SAR-PVU States		
1 X-axis tilt toward y	Random constant (independent)	
1 Y-velocity scale	First-order Markov (independent)	
1 Z-axis tilt toward y	Random constant (independent)	
3 Total SAR-PVU Filter States		
SAR-HRM States		
1 Range measurement bias error	First-order Markov	All measurements
1 Range-rate measurement bias error	First-order Markov	All measurements
1 Elevation measurement bias error	First-order Markov	All measurements
1 Azimuth measurement bias error	First-order Markov	All measurements
4 Total SAR-HRM Filter States		
7 Total SAR Filter States		
TAN States		
1 Radar altimeter bias error	First-order Markov	
1 Total TAN Filter State		

Table 3 - 10 DKF Filter Model Error States

3.3 DKFSIM 3.P1

3.3.1 Introduction

During the course of this thesis work, the author worked to develop a dual-INS implementation of DKFSIM to analyze the problem presented in this thesis. This proved to be quite difficult and time-consuming. The DKFSIM software is quite complicated and a good deal of information sharing and nested operations (one routine calling another, which in turn will call another, etc.) is carried out. The author did in fact implement a dual-INS DKF Models module and was working on modifying the DKF Filters module, when in October 1995, DKFSIM 3.P1 upgrades were released for use. The author ceased development work on his DKFSIM development model in favor of the DKFSIM 3.P1 upgrade.

3.3.2 Description

The DKFSIM 3.P1 upgrade contained the following modifications:

DKF Models module

- A second INS truth model was added for both separated mode and integrated mode of operation; and
- Modifications to random number generators, data blocks, and input/output messages were incorporated to accommodate data from a second INS.

DKF Filters module

- Navigation computer software (NAVCOM) was modified to use dual-INS data;
- Message handling routines were modified to accommodate data from two INS; and
- Local filter and master filter input parameters were modified to accept, individually, either reference INS or EGI INS data.

This modification allowed the author to carry on with modeling of the dual-INS EGI retrofit problem. Following issue, the author modified the source code of DKFSIM 3.P1 to accommodate the desired dual-INS integration of Design B of Section 2.6.2, on a PC 486. The changes were as follows:

- The software was installed on the PC 486.
- All compilation batch files were modified to allow compilation with the Lahey FORTRAN F77L-EM/32 compiler.
- A program error caused by compiler differences was tracked down and corrected by modifying a GPS control routine in the DKF Filters module.
- The master filter program MFILTR was modified to achieve the estimator conversion via Equations 2-63 and 2-64.

A subroutine was added to provide change of variable names for the EGI INS values, to allow use of dual-INS data in the same program block. To carry out the state estimate conversion process of

Equations 2-63 and 2-64, data from both INSs was read into the master filter program MFILTR just prior to fusion (which is the only time the conversion is required). The equations were:

$$\delta\hat{\mathbf{p}}_{ref}(t_i) = \delta\hat{\mathbf{p}}_{egi}(t_i) + \mathbf{y}_p(t_i) \tag{3-2}$$

$$\delta\hat{\mathbf{v}}_{ref}(t_i) = \delta\hat{\mathbf{v}}_{egi}(t_i) + \mathbf{y}_v(t_i) \tag{3-3}$$

$$\hat{\theta}_{ref}(t_i) = \hat{\theta}_{egi}(t_i) + \mathbf{y}_\theta(t_i) \tag{3-4}$$

where:

$\delta\hat{\mathbf{p}}_{ref}(t_i)$ are state estimates of the reference INS position error at time t_i

$\delta\hat{\mathbf{p}}_{egi}(t_i)$ are state estimates of the EGI INS position error at time t_i

$\mathbf{y}_p(t_i)$ are realizations of the EGI INS/reference INS difference in position error at time t_i

and likewise for velocity and tilt state estimates.

Equations 3-2 to 3-4 were implemented as follows:

- Common data blocks of both the reference INS and the EGI were included in the MFILTR program. These data blocks contained the INS realizations in double precision real format, to provide the necessary accuracy for position, velocity and tilt information; the error estimate values were in single precision real format (only single precision required due to small magnitude of the error estimates).
- Keeping in mind that the realizations of $\mathbf{y}(t_i)$ were whole-state values from the INSs, compared with error state values from the estimators (e.g. millions of feet vs. 15-20 feet values), an effort was made to minimize problems due to scale. Realizations of

$\mathbf{y}(t_i)$ were computed in double precision, then converted to single precision prior to being added to $\hat{\mathbf{p}}_{egi}(t_i)$, $\hat{\mathbf{v}}_{egi}(t_i)$ and $\hat{\theta}_{egi}(t_i)$.

- Values for $\mathbf{y}_p(t_i)$ and $\mathbf{y}_v(t_i)$ were computed in state-to-state subtraction of INS values, and values for $\mathbf{y}_\theta(t_i)$ had to be computed from body to ECEF direction cosine matrices (DCMs) from both INSs.

Values for $\mathbf{y}_\theta(t_i)$ were computed based on the following derivation [25]:

$$\mathbf{C}^b_{ei} = (\mathbf{I} - \mathbf{E}^n{}_i)\mathbf{C}^b_e \tag{3-5}$$

where:

$\mathbf{C}^b_{ei}$ is the INS-indicated angular differences between the body frame and ECEF frame in DCM form from INS #i

$\mathbf{E}^n{}_i$ is the error in the INS-indicated angular differences

$\mathbf{C}^b_e$ is the true angular differences between the body frame and ECEF frame in DCM form

$\mathbf{E}^n{}_i$ is the skew-symmetric form of the error vector θ:

$$\mathbf{E}^n = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \tag{3-6}$$

where:

θ_x, θ_y, θ_z are the errors in the INS-indicated angles, expressed in ECEF coordinates

For both INSs, we obtain the following pairs (for easy notation in this derivation, INS1 is the reference INS and INS2 is the EGI INS):

$$\mathbf{C}_{e1}^{b} = (\mathbf{I} - \mathbf{E}^{n}{}_{1})\mathbf{C}_{e}^{b} \tag{3-7}$$

$$\mathbf{C}_{e2}^{b} = (\mathbf{I} - \mathbf{E}^{n}{}_{2})\mathbf{C}_{e}^{b} \tag{3-8}$$

The difference between the INS1 and INS2 errors can be extracted:

$$\mathbf{C}_{e1}^{b}(\mathbf{C}_{e2}^{b})^{T} = \mathbf{C}_{e1}^{e2} = (\mathbf{I} - \mathbf{E}^{n}{}_{1})\mathbf{C}_{e}^{b}(\mathbf{C}_{e}^{b})^{T}(\mathbf{I} - \mathbf{E}^{n}{}_{2})^{T} \tag{3-9}$$

$$\mathbf{C}_{e1}^{e2} = (\mathbf{I} - \mathbf{E}^{n}{}_{1})(\mathbf{I} - \mathbf{E}^{n}{}_{2})^{T} \tag{3-10}$$

Multiplying out the two matrices results in the following direction cosine matrix:

$$\mathbf{C}_{e1}^{e2} = \begin{bmatrix} 1 & \delta\theta_{12z} & -\delta\theta_{12y} \\ -\delta\theta_{12z} & 1 & \delta\theta_{12x} \\ \delta\theta_{12y} & -\delta\theta_{12x} & 1 \end{bmatrix} + \text{ Higher Order Terms} \tag{3-11}$$

where:

$\delta\theta_{12x}$, $\delta\theta_{12y}$, $\delta\theta_{12z}$ are the differences in the errors of the INS-indicated angles ($\delta\theta_{ref}$ - $\delta\theta_{egi}$)

Higher Order terms are the products of the error terms. Since the angular errors are typically quite small, the higher order terms are negligible and are dropped. From $\mathbf{C}_{e1}^{e2}$, the redundant elements are extracted and averaged to obtain the angle differences:

$$\delta\theta_{12x} = \frac{\mathbf{C}_{e1}^{e2}(2,3) - \mathbf{C}_{e1}^{e2}(3,2)}{2} \tag{3-12}$$

$$\delta\theta_{12y} = \frac{C_{el}^{e2}(3,1) - C_{el}^{e2}(1,3)}{2} \tag{3-13}$$

$$\delta\theta_{12z} = \frac{C_{el}^{e2}(1,2) - C_{el}^{e2}(2,1)}{2} \tag{3-14}$$

These error angle differences comprise $y_\theta(t)$:

$$y_\theta(t_i) = \begin{bmatrix} \delta\theta_{12x}(t_i) \\ \delta\theta_{12y}(t_i) \\ \delta\theta_{12z}(t_i) \end{bmatrix} \tag{3-15}$$

This modification allows for the operation of the federated filter in the No Reset and Zero Reset modes. With additional coding to perform estimate conversions for resets (not done in this research), Partial Reset and Full Reset modes could be implemented.

3.4 Comparison Models

3.4.1 Introduction

Both the federated filter and centralized filter designs were modeled in the modified version of DKFSIM 3.P1, to ensure that identical DKF Models modules were used. Identical input data sets, with the exception of the filter configuration input files, ensured that the same noise values, data output times, etc. were used in the simulations.

From the four possible federated filter modes, the No Reset mode was chosen for the design, for the following reasons:

- The No Reset mode federated filter has a high degree of fault tolerance. Since there is no feedback to local filters from the master filter, the local filters cannot be corrupted by bad measurement data from a faulty sensor (other than the one local filter using the sensor measurements).

- A developmental approach was taken. Due to the EGI-to-reference INS state conversion requirement, the No Reset mode was the logical first step for implementing this new theory. If it worked, further development could add in reference-to-EGI conversion to implement reset to the EGI filter.
- Ease of implementation was essential. The latest DKFSIM versions were only acquired in October, and the DKFSIM code was quite complex. The author had the best chance of succeeding with an implementation of the federated filter in the time available by using the simplest mode to code, the No Reset mode.

3.4.2 Description

Design A

Figure 3 - 3 shows the Design A implementation. Local Filter 1 represents an EGI, with its Kalman filter augmented with states for the additional sensors, SAR and TAN. Thus, it operates as a centralized filter.

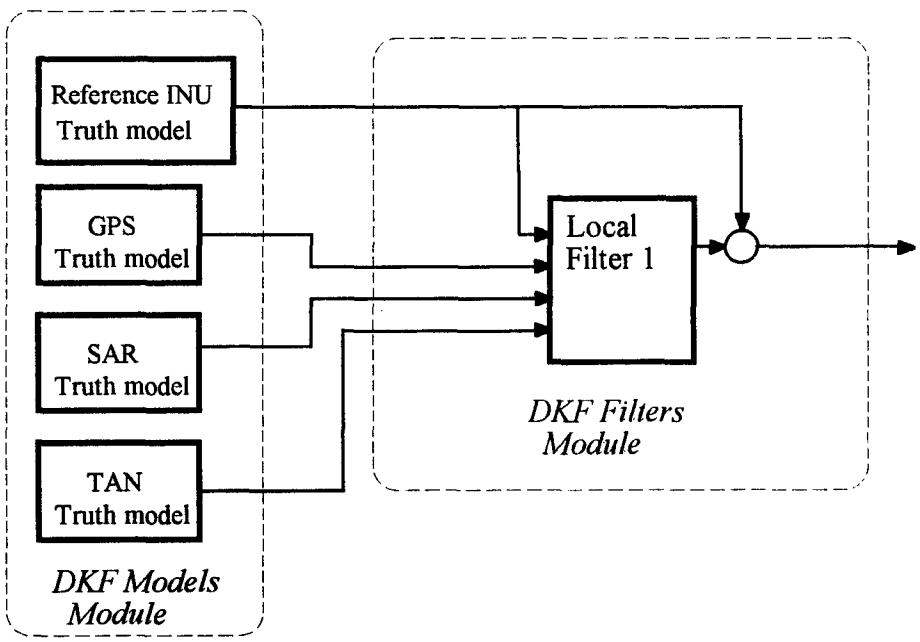


Figure 3 - 3 Design A in DKFSIM

Design B

This model emulates the federated filter design by using Local Filter 1 in the DKF Filters module as the EGI filter. As the actual EGI filter has an INS and GPS receiver for sensor inputs,

the Local Filter 1 inputs are from the GPS truth model from the DKF MODELS module, plus a second INS truth model, separate and independent from the navigation system reference INS (truth model). Local Filter 2 has as inputs measurements from the reference INS (truth model) and measurements from the SAR (truth model). Local Filter 3 has as inputs measurements from the reference INS (truth model) and TAN system (truth model), respectively.

For master filter fusion, the state estimates from Local Filter 1 are converted from EGI INS errors to system INS errors in a time-synchronized fashion, in the master filter. Then, fusion of the covariance matrices and state estimates may proceed. Outputs from the master filter are used as system output. Figure 3 - 4 shows the federated filter in block-diagram form.

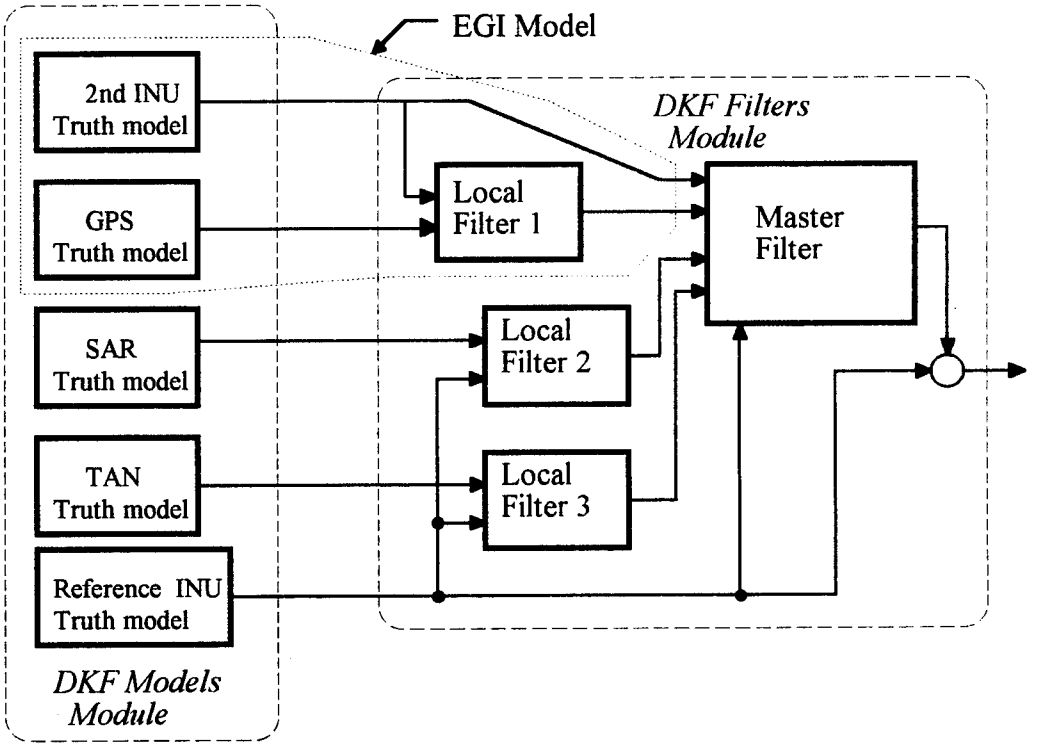


Figure 3 - 4 Design B in DKFSIM

3.5 Summary

Chapter 3 explained the models used for simulation studies. First, the simulation software DKFSIM was described, including truth and filter models. Next, modifications to DKFSIM were explained. Finally, the DKFSIM configurations for the centralized and federated filter designs were given.

4. Results

4.1 Introduction

This chapter shows the results of computer simulations for both the centralized filter and the federated filter. The model setup (input information and initial conditions) is first described in Section 4.2. Following this, data and analysis of the computer simulations are presented in Section 4.3.

4.2 Model Setup

4.2.1 General

Model setup for all simulations is presented in this section. Strict control of input data files and model versions were used to ensure that all input parameters, except for commanded changes, were identical. All input files were initially set up for simulation of the federated filter and centralized filter with all sensors operating correctly. When configuration changes were required for simulations with failures, copies of these files were created and modified with the required changes, but the original versions of the input files were kept.

4.2.1.1 Flight Profile

A flight trajectory file produced by PROFGEN was used for the flight trajectory input data file. The trajectory was called the Advanced Tactical Fighter (ATF) profile, and was written by Mr. Stanton Musick. It was intended to provide representative dynamics of a multi-mission fighter/attack aircraft.

An excellent description of the ATF profile is contained in [11]. A detailed description of the input parameters to the profile and the output maneuvers in a segment-by-segment basis is contained in an output file produced by PROFGEN called PROF.OUT. From [11], the first four thousand seconds (about one hour and six minutes) of this profile are summarized in Table 4 - 1.

Start (sec)	End (sec)	Maneuver
0	230	Take-off and climb to cruise altitude
230	1790	Outbound cruise at 40,000 ft and 500 knots, including surface-to-air missile evasion at about 1700 sec, using a 6 G "roll and pull" maneuver
1790	2100	Descent to 200 ft. , 500 knots, same heading
2100	2300	Low altitude (200 ft) penetration run including 90 sec period of 6 G jinking (rapid-succession S-turns) representing surface-to-air missile site evasions
2300	3250	Heading change of 90 degrees for final run toward target (same altitude and airspeed)
3250	3300	"Pop-up" maneuver involving 4 G pull-up, roll to inverted flight, pull-down (inverted), roll-back to normal attitude, and simulated weapons release
3300	3600	Climbing turn (6 G) to escape route heading, including 50 sec period of high-dynamic air-to-air combat
3600	4000	Straight and level cruise outbound flight at 27,000 ft

Table 4 - 1 ATF Trajectory Profile Summary

This profile was selected for the following reasons:

- The profile provide the required high- and low-dynamics situations required to assess the navigation system performance. This is provided by the run in portion (0 - 2050 sec, low dynamics) and attack portion (2050 - 3600 sec, high dynamics). The final portion (3600 - 4000 sec, low dynamics) was used to assess the two design's ability to resume what should be the more accurate low-dynamics navigation after the simulated combat situation.
- It was available for use in DKFSIM.
- It was familiar to the author and had been used at AFIT for navigation systems simulation [25].

4.2.1.2 Sensor and Noise Parameters

All sensors were established with typical noise values for a navigation system. Both the reference INS and EGI INS truth models were established with values allowing them to operate as 0.8 NM/hr drift strapdown RLG INSs. Both INS models were started in simulation as if they had been aligned prior to flight; fine-align covariance values for filter INS errors were the initial conditions for the simulation. Parameters for all of the truth models were contained in the DKFSIM input files; these files are included as Appendix B.

The GPS truth model was configured for four satellite coverage, with sensor measurements of PR and PRR passed to the DKF Filters module at 4 second intervals. There is no wait time for commencement of measurements; GPS measurements are available from the beginning of flight. In addition, the GPS truth model determines whether or not satellites are visible based on satellite position and aircraft attitude. If a satellite line-of-sight is not part of the upper hemisphere of body-frame coordinates, simulating upper-aircraft antenna siting, the satellite measurement is not available.

The SAR truth model is configured to provide a landmark measurement every 100 seconds, commencing 300 seconds after take-off. A SAR precision velocity update occurs just before the landmark measurement.

The radar altimeter truth model provides altimeter measurements every 2 seconds, commencing 300 seconds after take-off. If the aircraft attitude is more than 30 degrees in pitch, roll, or some combination of pitch and roll from the horizontal, radar altimeter measurements are disabled. This simulates the directional beam pattern effects of the radar altimeter.

4.2.1.3 Filter Tuning

Filter models in DKFSIM were provided with pre-tuned values [22,26], for initial conditions and noise values. To ensure the filters involved were in fact tuned properly, outputs from the centralized filter and outputs from all local filters and master filter were examined. State estimates and filter-predicted covariances were examined for stability and expected accuracy, and covariances calculated from the Monte Carlo simulations were compared to all filter-predicted covariances. Twenty Monte Carlo runs were conducted for each simulation run, where twenty provided a good compromise between statistical confidence in the results and manageable size of output files. These comparisons for the centralized filter and federated filter (master filter only) are shown in Appendix A.

After tuning for best performance, all filter states performed conservatively, with the exception of the East Position and North Position states in Local Filter 1 (GPS-fed filter), and in the Master Filter when Local Filter 1 information was fused in the solution. Although these states are non-conservatively tuned, best overall performance of the filter was attained with this tuning. This is due to the noise characteristics of the GPS pseudorange measurements [26]. The filter

could have been more conservatively tuned for these states by increasing process noise strength values, but this would not have been appropriate to tuning for the GPS measurements.

4.2.1.4 Performance metrics

The mean and standard deviation of state estimates for 20 Monte Carlo runs for both designs are contained in Appendix A. Another set of performance metrics, other than state-by-state comparison, are RSS errors of position and velocity, called system error and velocity error. System error is defined in this thesis as the total position error, calculated in the following manner:

$$E_{sys}(t_i, mc_j) = \sqrt{[E_e(t_i, mc_j)]^2 + [E_n(t_i, mc_j)]^2 + [E_u(t_i, mc_j)]^2} \quad \textbf{(4-1)}$$

$$E_{sys}(t_i) = \frac{\sum_{j=1}^{n} E_{sys}(t_i, mc_j)}{n} \quad \textbf{(4-2)}$$

where:

$E_{sys}(t_i, mc_j)$ is the RSS error calculated for time t_i and Monte Carlo run mc_j

$E_{e,n,u}(t_i, mc_j)$ are the East, North, and Up state estimate errors in filter-predicted values for time t_i and Monte Carlo run mc_j

$E_{sys}(t_i)$ is the system error calculated for time t_i, averaged over n Monte Carlo runs

The system error and velocity error plots show error in three dimensions, with only one plot. These plots give a good indication of system performance, and do not inundate the reader with multiple graphs.

Figure 4 - 1 describes the system error vector (note that only the magnitude of the vector is calculated).

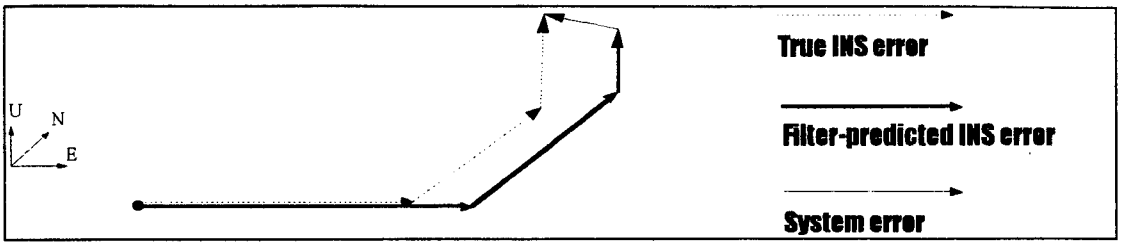


Figure 4 - 1 System Error

Velocity error $E_v(t_i)$ is a scalar value yielding the magnitude of the velocity error vector. It is calculated in the same manner as system error.

Thus, there are two diagrams (system error and velocity error) which are used to summarize the performance of the navigation systems for each simulation, along with explanation of the initial conditions, changes during the simulation, failures, etc. Where applicable, other data and diagrams are introduced for explanation and emphasis

The simulations carried out are summarized in Table 4 - 2. First, Runs 1A and 1B provide the necessary benchmarks for performance under normal operating conditions, defined here as all sensors operating correctly, with the aircraft undergoing both high and low dynamic conditions of flight. Following this, Runs 2A and 2B allow the analysis of both designs' performance with GPS outages. Runs 3A and 3B simulate a receiver clock failure early in the flight to examine the effects on system performance. Finally, Runs 4A, 4B and 4C simulate accelerometer failure of steadily increasing noise (noise ramp), and the resulting operation.

Run #	Filter Design Used	Sensors			Failures		
		Sensor	Start (t=)	End (t=)	Device	Start (t=)	End (t=)
1A	centralized filter	GPS SAR TAN	0 300 300	4000 4000 4000	No failures		
1B	federated filter	GPS SAR TAN	0 300 300	4000 4000 4000	No failures		
2A	centralized filter	GPS SAR TAN	0 3600 300 300	3100 4000 4000 4000	GPS outage simulates jamming		
2B	federated filter	GPS SAR TAN	0 3600 300 300	3000 4000 4000 4000	GPS outage simulates jamming		
3A	centralized filter	GPS SAR TAN	0 300 300	4000 4000 4000	GPS Rcvr clock frequency	300	4000
3B	federated filter	GPS SAR TAN	0 300 300	4000 4000 4000	GPS Rcvr clock frequency	300	4000
4A	centralized filter	GPS SAR TAN	0 300 300	4000 4000 4000	Long. accel noise ramp	500	4000
4B	federated filter	GPS SAR TAN	0 300 300	4000 4000 4000	Long. accel noise ramp in Ref INS	500	4000
4C	federated filter	GPS SAR TAN	0 300 300	4000 4000 4000	Long. accel noise ramp in EGI INS	500	4000

Table 4 - 2 Simulation Summary Table

4.2.2 Runs 1A and 1B - Performance Comparison and Benchmark

4.2.2.1 Conditions

For performance comparison, all simulations were conducted with all sensor measurements available for the entire flight profile (except where conditions of flight prevented sensor measurements). All input information is contained in the input files contained in Appendix C.

4.2.2.2 Analysis

The purpose of Runs 1A and 1B was to provide direct comparison between the designs when both were working perfectly, and to provide benchmarks for further simulations involving sensor outages and failures. Graphs showing filter-predicted state estimates and covariances for all nine INS error states for both designs, as well as one-sigma values calculated from the data, are presented in Appendix A.

Figures 4 - 2 and 4 - 3 show the system error and Figures 4 - 4 and 4-5 show the veloctiy error for the centralized filter and federated filter, using inputs of GPS, SAR, and TAN. By studying the output of each of the local filters, it is apparent that the GPS measurements dominate the statistics of the filter, and system error can be seen to be approximately 25 -35 feet, after reaching steady-state operation. Some loss of accuracy can be seen approximately centered at times 1700 sec, 2300 sec and 3400 sec in the velocity error plots. These times correspond to highly dynamic maneuvers as detailed in Section 4.2.1.1, where some measurement data is lost, and possibly due to the dynamics effects in the covariance matrix **P**.

From the diagrams shown, the position error for both designs, after reaching steady state, varies between an average of 30 to 35 feet. The federated filter shows more variability in this error, where the centralized filter tends to be somewhat more consistent. This is probably due to the master filter solution being calculated only once every ten seconds, and then propagated for ten seconds, as compared to a propagate/measurement update cycle of two seconds in the centralized filter. The velocity error is better for the centralized filter, averaging 0.05 - 0.06 ft/sec error, compared with the federated filter, averaging about 0.1 ft/sec error.

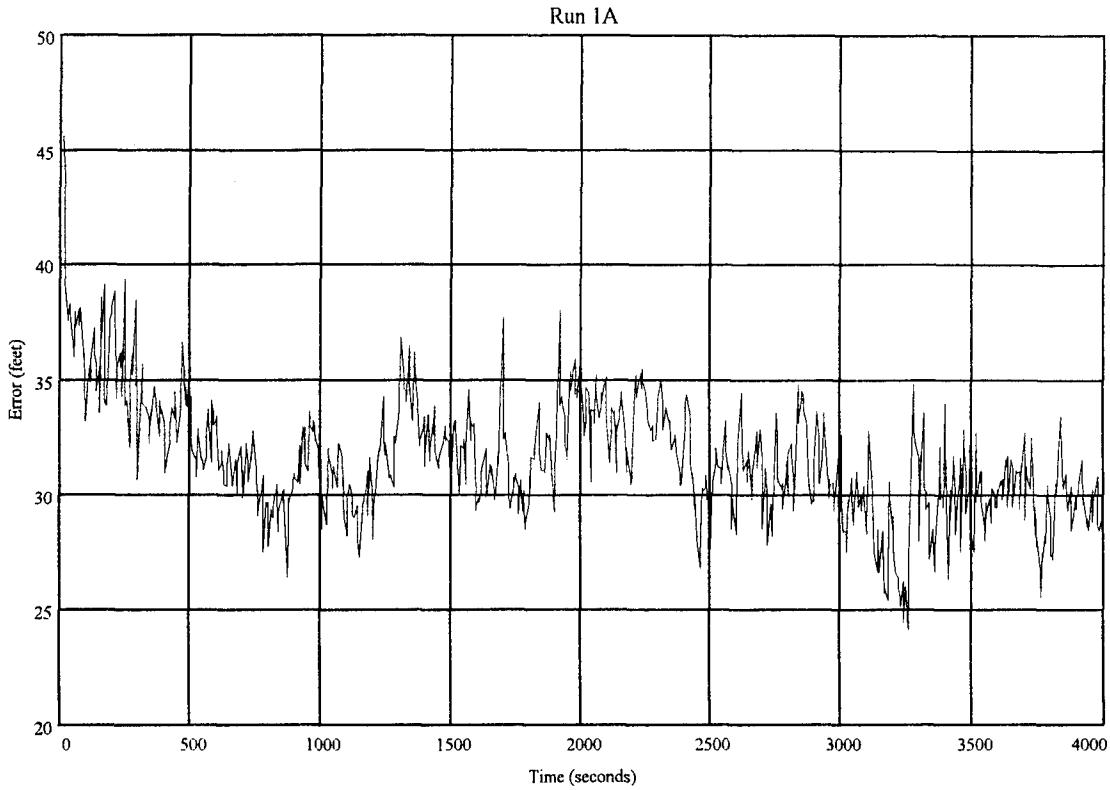


Figure 4 - 2 Run 1A, All Sensors, Centralized Filter System Error

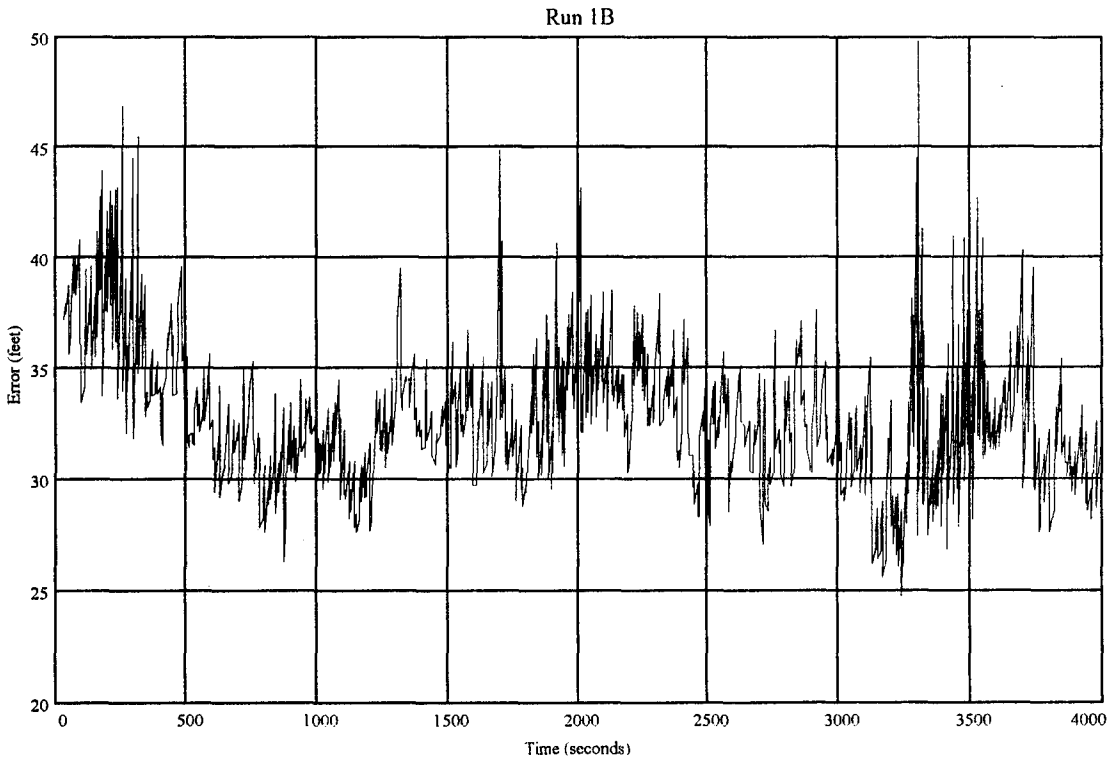


Figure 4 - 3 Run 1B, All Sensors, Federated Filter System Error

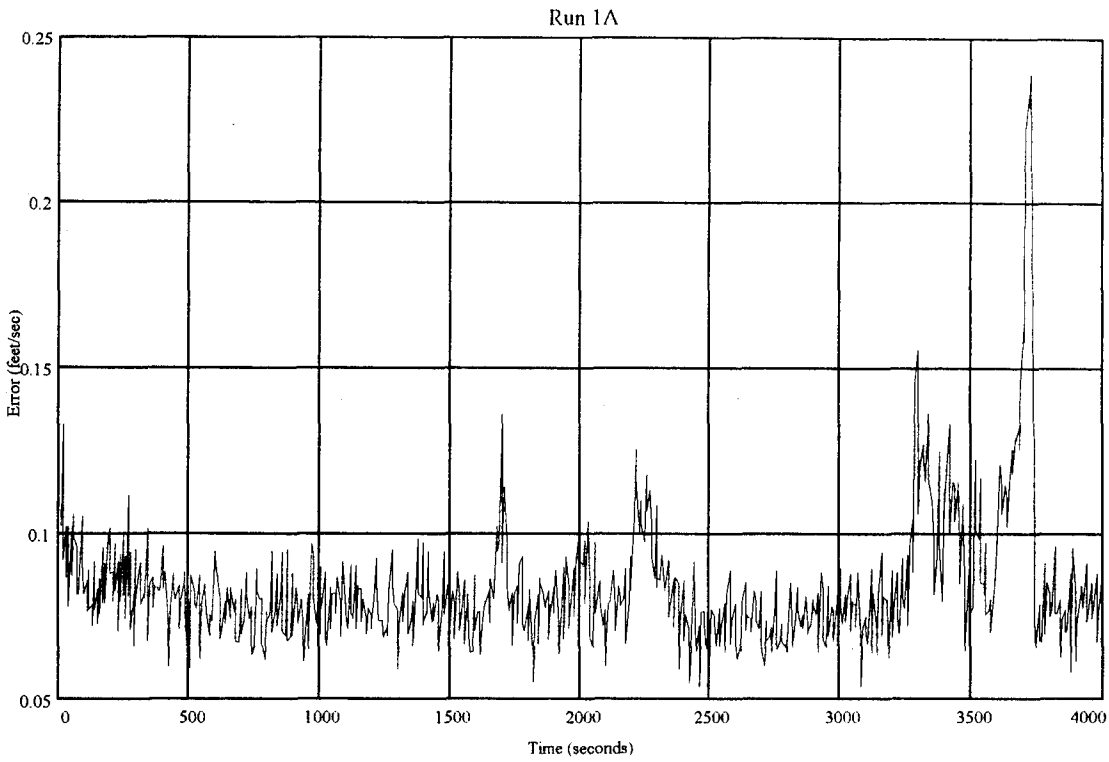


Figure 4 - 4 Run 1A, All Sensors, Centralized Filter Velocity Error

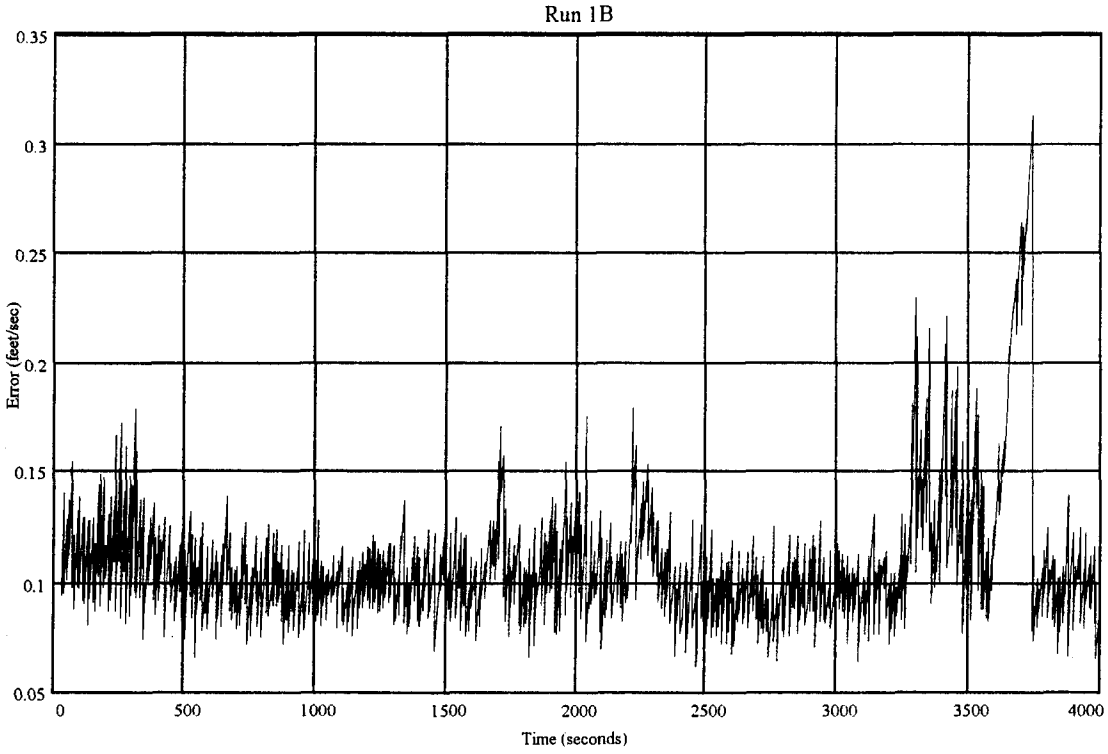


Figure 4 - 5 Run 1B, All Sensors, Federated Filter Velocity Error

4.2.3 Runs 2A and 2B - GPS Outages

4.2.3.1 Conditions

This run simulates GPS jamming conditions at the weapons release site. GPS data is not available from 3000 - 3600 sec (400 sec prior to weapons delivery at 3400 through 200 sec after), simulating a jammer located close to the weapons release site. All other sensors are available as in Run 1A and 1B.

4.2.3.2 Analysis

Figures 4 - 6 and 4 - 7 show the centralized and federated filter system error, and Figures 4 - 8 and 4 - 9 display the centralized and federated filter velocity error. Comparison of system error for both designs at the time of interest shows that the solution quickly degrades after t = 3000 due to lack of GPS information. The SAR measurements then become the dominant (most accurate) sensor data. Note the filter updates every 100 seconds using SAR measurements, and these filter updates are quite noticeable. When GPS jamming ceases (at t = 3600 sec), the aircraft is maneuvering, and it is not until about t = 3700 that the aircraft is sufficiently stable to receive GPS measurements.

Again, filter accuracy in position is almost the same in the centralized and federated filter, and velocity accuracy is somewhat better in the centralized filter as compared to the federated filter.

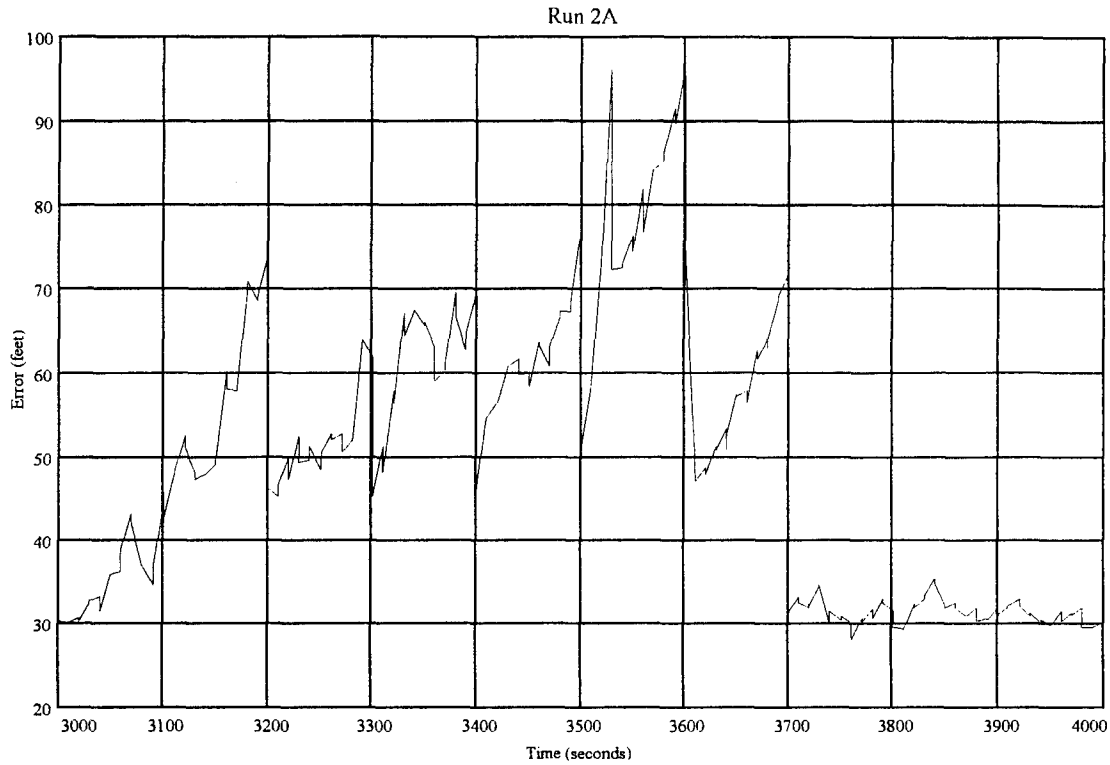


Figure 4 - 6 Run 2A, Centralized Filter (GPS jammed) System Error

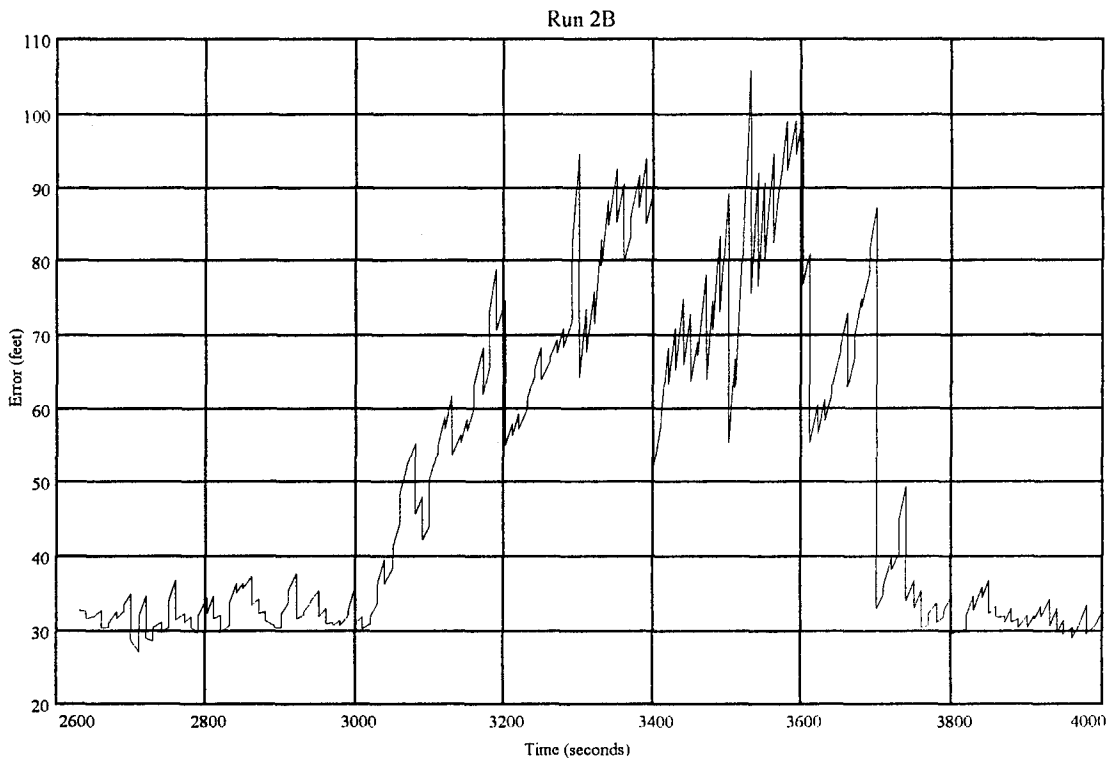


Figure 4 - 7 Run 2B, Federated Filter (GPS jammed) System Error

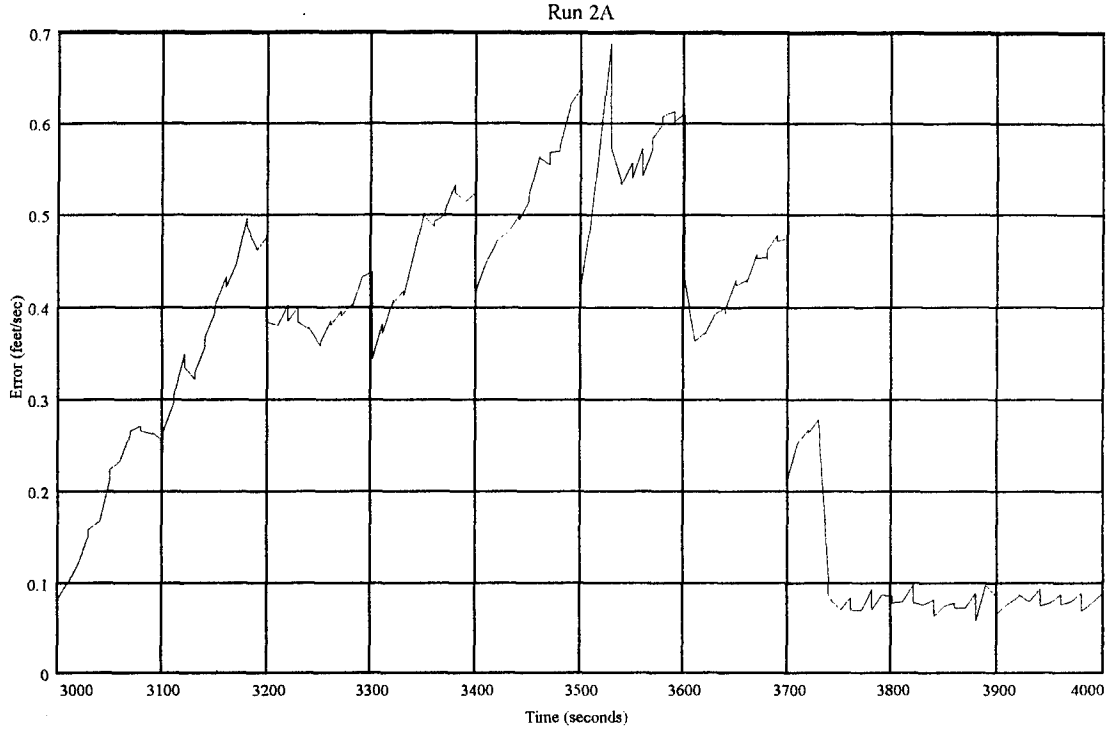


Figure 4 - 8 Run 2A, Centralized Filter (GPS jammed) Velocity Error

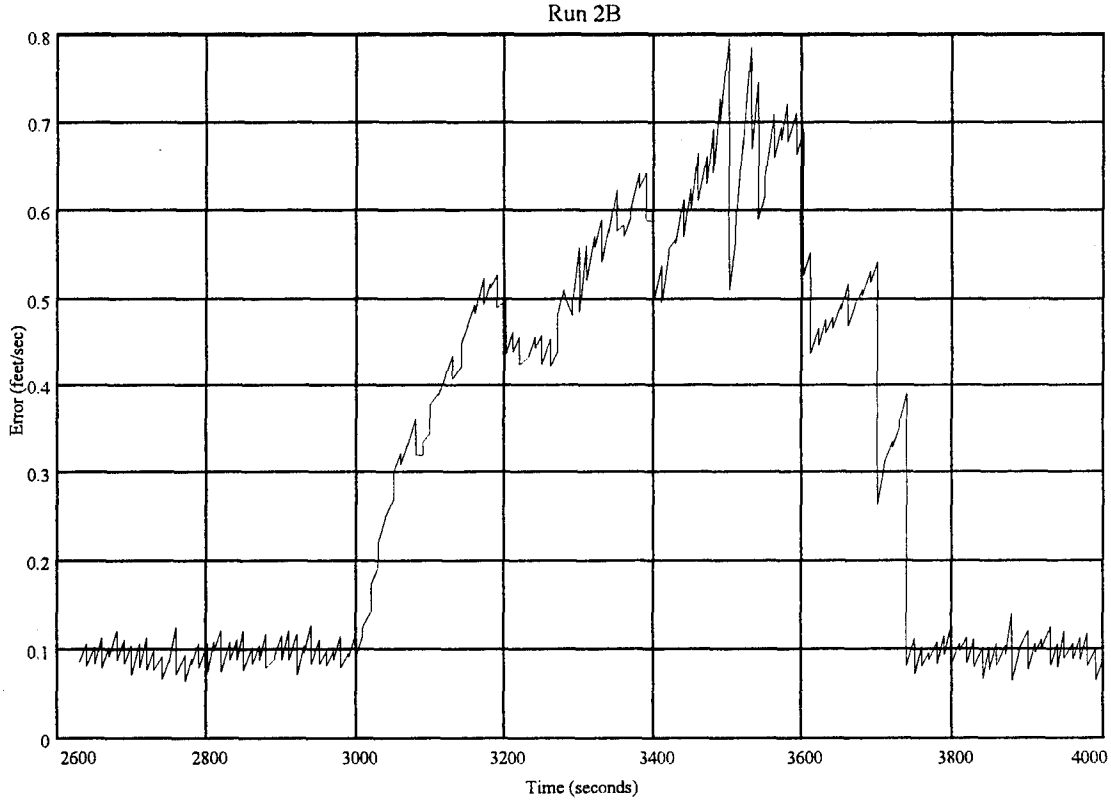


Figure 4 - 9 Run 2B, Federated Filter (GPS jamming) Velocity Error

4.3.1 Runs 3A and 3B - GPS Receiver Clock Failure

4.3.1.1 Conditions

These runs simulated a failure of the GPS receiver clock. The failure commenced at t = 300 sec, with the clock frequency error growing linearly with time at a rate of 0.004 ft/sec^2. Because the receiver clock is used to make all PR measurements, an error was seen in all PR and PRR measurements.

4.3.1.2 Analysis

System error for the centralized filter is shown in Figure 4 - 10, and the corresponding velocity error in Figure 4 - 12. The centralized filter was configured to reject residuals greater than 5 times the value of a weighted moving average filter incorporating 10 measurements. From the log file, measurement rejections did not occur to any great extent until t = 1500. Since the failure is a gradual one, the state estimates and covariances were corrupted. Errors accumulated rapidly, since the poor measurements were not rejected by the moving average filter.

System error for the federated filter is shown in Figure 4 - 11, and velocity error in Figure 4 - 13. The federated filter was configured for measurement residual rejection in each of the local filters in the same manner as the centralized filter.

Although the federated filter design could distinguish the difference between the local filter solutions, it could not discern which local filter was in error. Fusion starts by bringing in the Local Filter 1 solution, then adding the other local filter solutions to it. During this fusion process, a residual is formed for the Local Filter 2 and 3 solutions . If a certain threshold is passed, the local filter solution is rejected.

For Run 3B, the federated filter was reconfigured so that the master filter would begin fusion with Local Filter 2, which has SAR measurements, then Local Filter 3, with TAN measurements, and lastly, Local Filter 1. According to the log file created during this run, rejection of local filter solutions through fusion residual monitoring started almost immediately after Local Filter 1 started to receive bad measurements. MF fusion was from then on accomplished with Local Filter 2 and 3 only, which provided consistent solutions for the remainder of the flight.

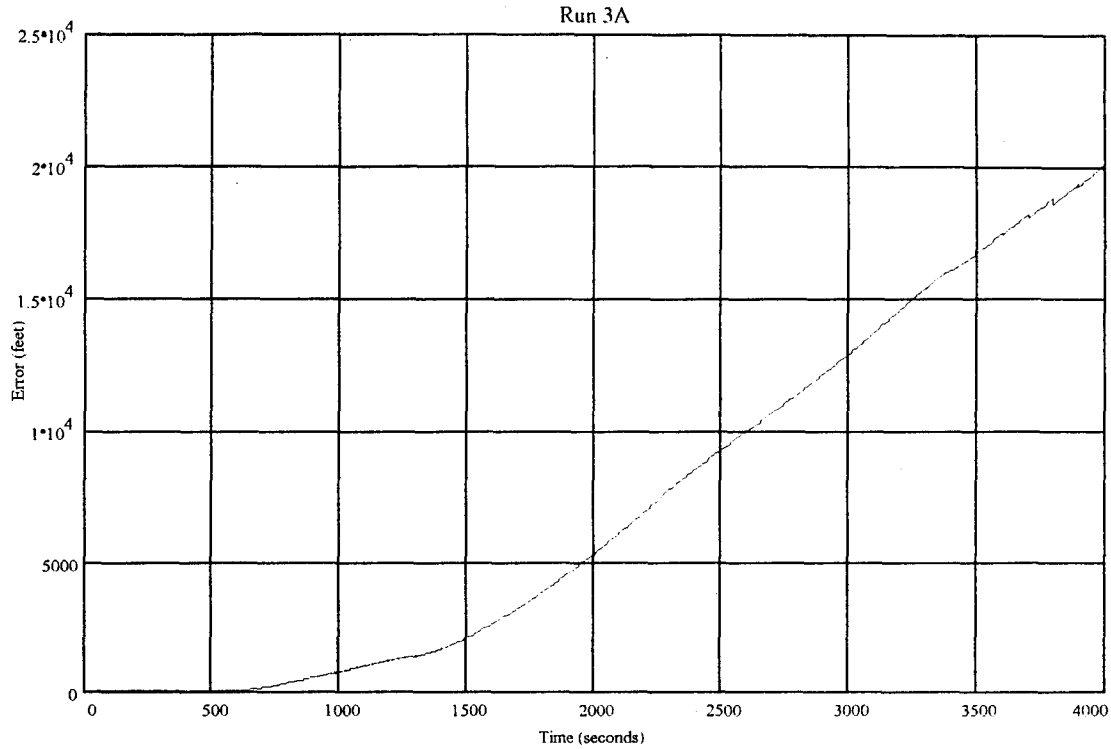


Figure 4 - 10 Run 3A, Centralized Filter, System Error

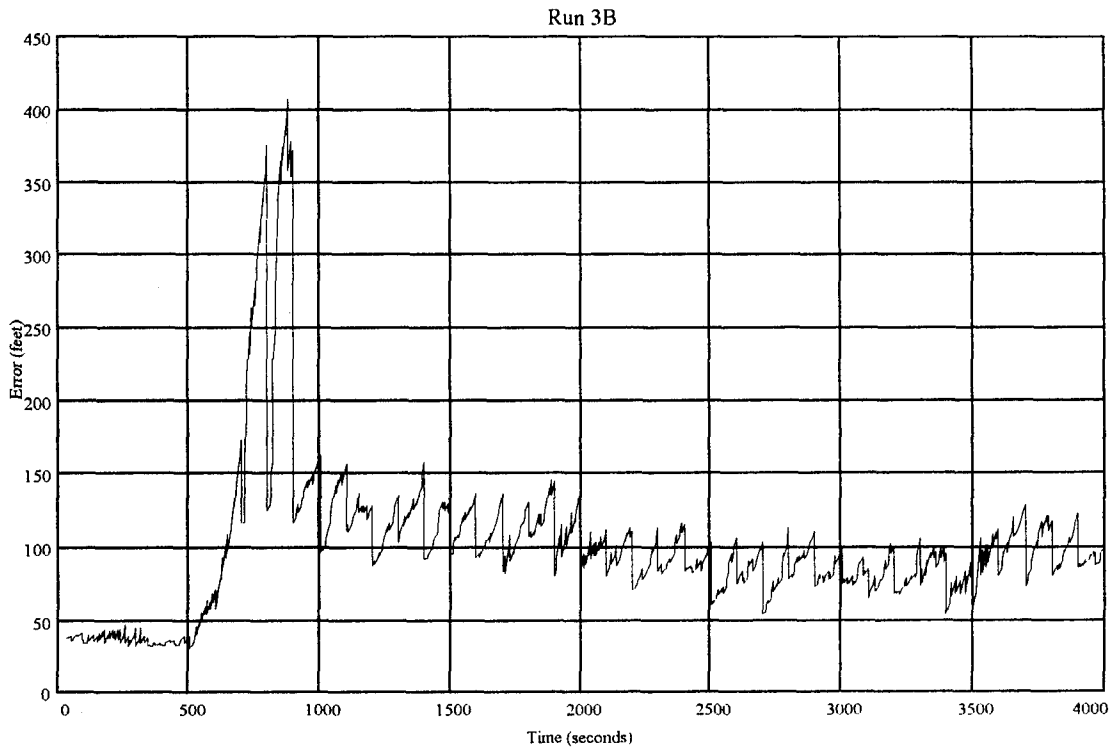


Figure 4 - 11 Run 3B, Federated Filter, System Error

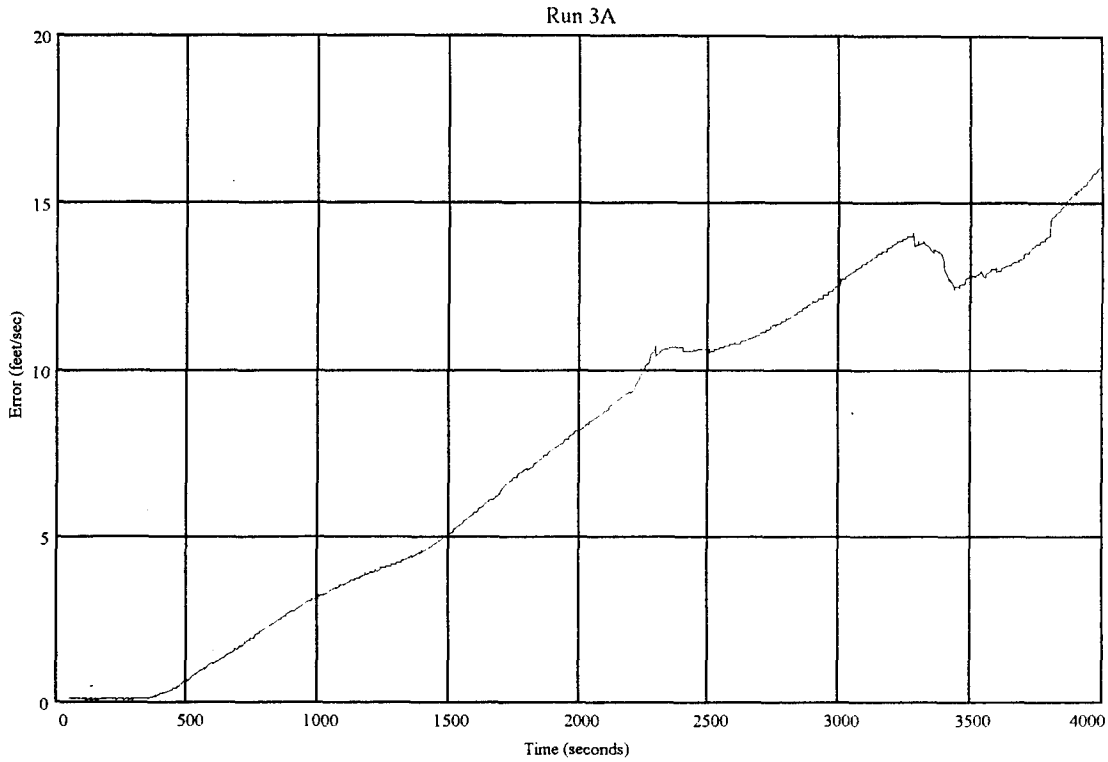


Figure 4 - 12 Run 3A, Centralized Filter, Velocity Error

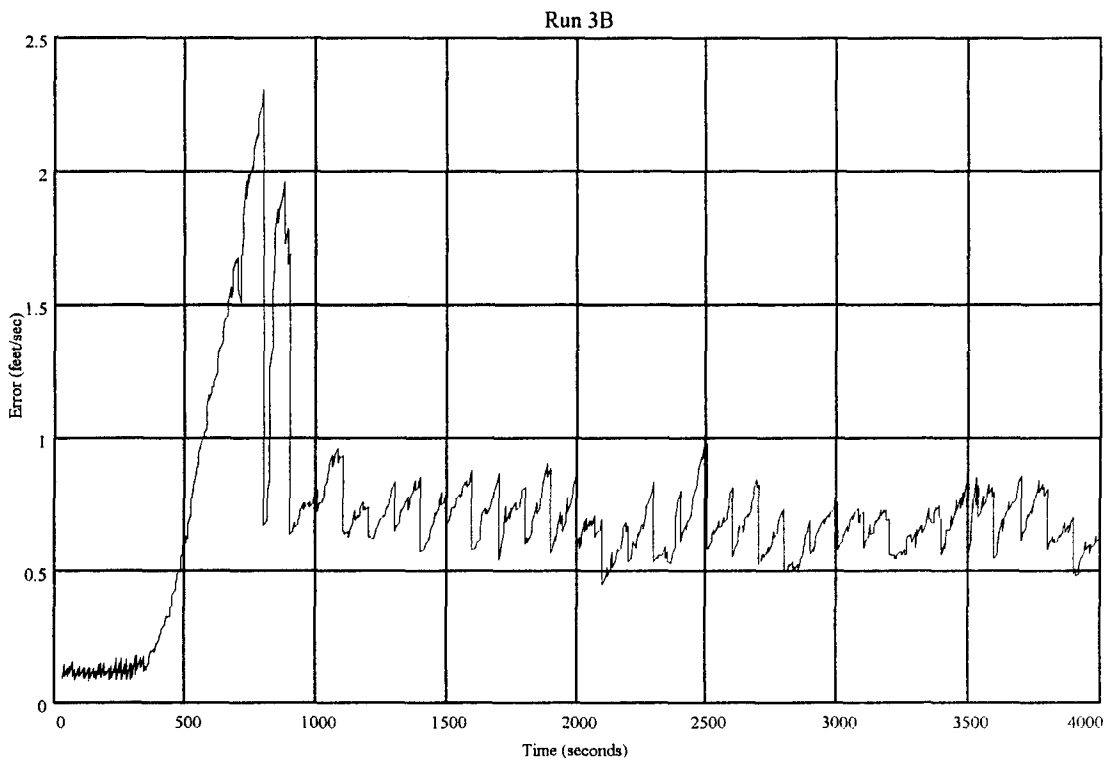


Figure 4 - 13 Run 3B, Federated Filter, Velocity Error

4.3.2 Runs 4A, 4B and 4C - Accelerometer Failure

4.3.2.1 Conditions

Three sets of conditions were set for this run; all involve failure of the longitudinal accelerometer in either the reference INS or the EGI INS. The longitudinal accelerometer was chosen to provide a significant effect, although any accelerometer failure could have been chosen. The accelerometer, commencing at t = 500 sec and continuing to the end of flight, has noise strength increasing linearly. The noise strength one-sigma value increases at a rate of 0.005 ft/sec^2 every second. In Runs 4A and 4B, the failure occurs in the reference INS. In Run 4C, the failure occurs in the EGI INS.

4.3.2.2 Analysis

Figures 4 - 14 and 4 - 17 showed that the centralized filter had rapidly increasing error due to the accelerometer failure. The filter was partially aided on regaining GPS measurements at about t = 3400 sec, but the accelerometer noise quickly dominated the filter operation and the filter performed poorly in the latter part of flight. There was no error detection scheme or algorithm for coping with an accelerometer failure built into the centralized filter design.

Figures 4 - 15 and 4 - 18, the federated filter with a reference INS accelerometer failure, showed much better performance. There is still quite a degradation in solution, which is due to the nature of the Kalman filter estimate conversion. Since it uses information from the reference INS to do the conversion, and this information is corrupt, the fused solution is corrupted. A more accurate solution at this point is Local Filter 1, the EGI filter. However, there is no way currently implemented to determine the problem with the INS, and the conversion/fusion process does not take advantage of this high-accuracy solution.

Figures 4 - 16 and 4 - 19, the federated filter with an EGI INS accelerometer failure, show a straightforward operation. When solutions from the EGI started to show large errors, this error was detected in the fusion residuals. The Local Filter 1 solution was rejected in the fusion process, and the master filter contained solutions fused from Local Filters 2 and 3. Thus, the high-accuracy GPS information was not used in the solution, but the solution was not corrupted by any erroneous data.

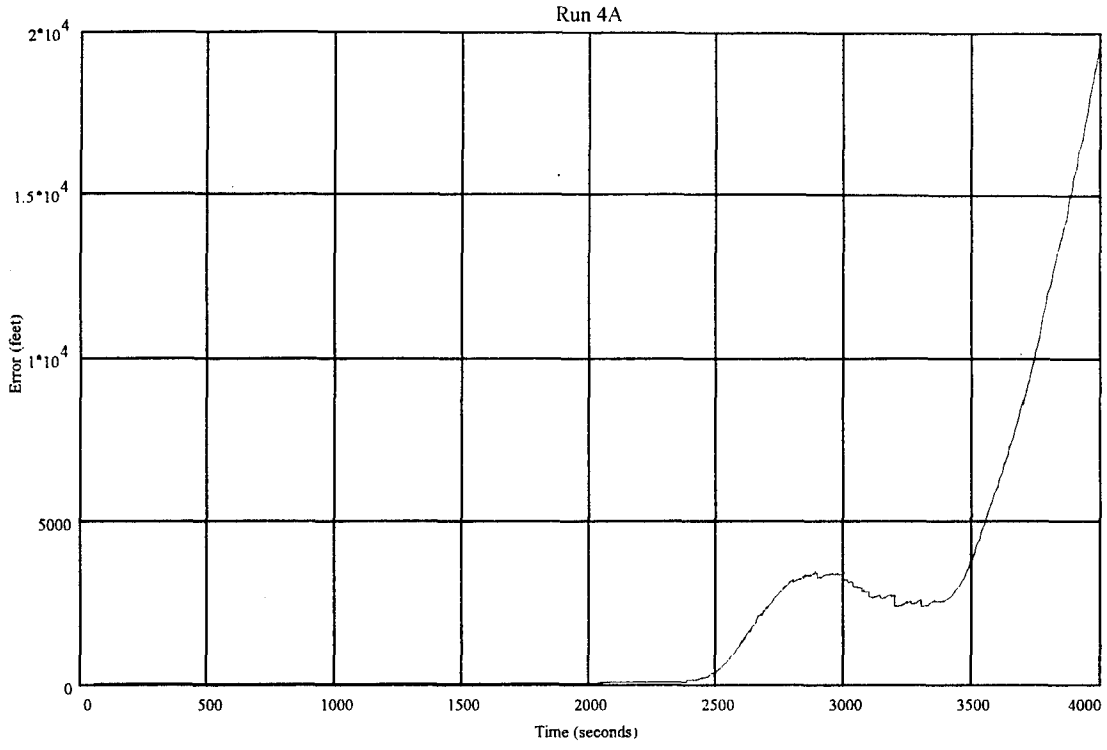


Figure 4 - 14 Run 4A, Centralized Filter, Position Error

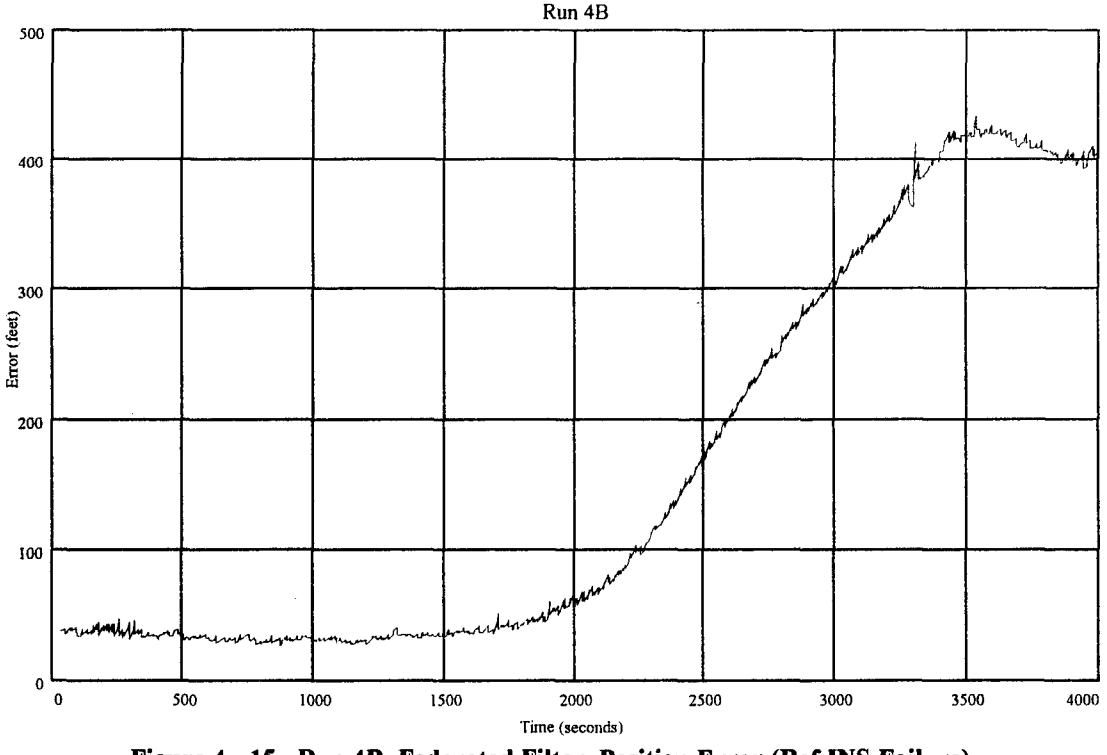


Figure 4 - 15 Run 4B, Federated Filter, Position Error (Ref INS Failure)

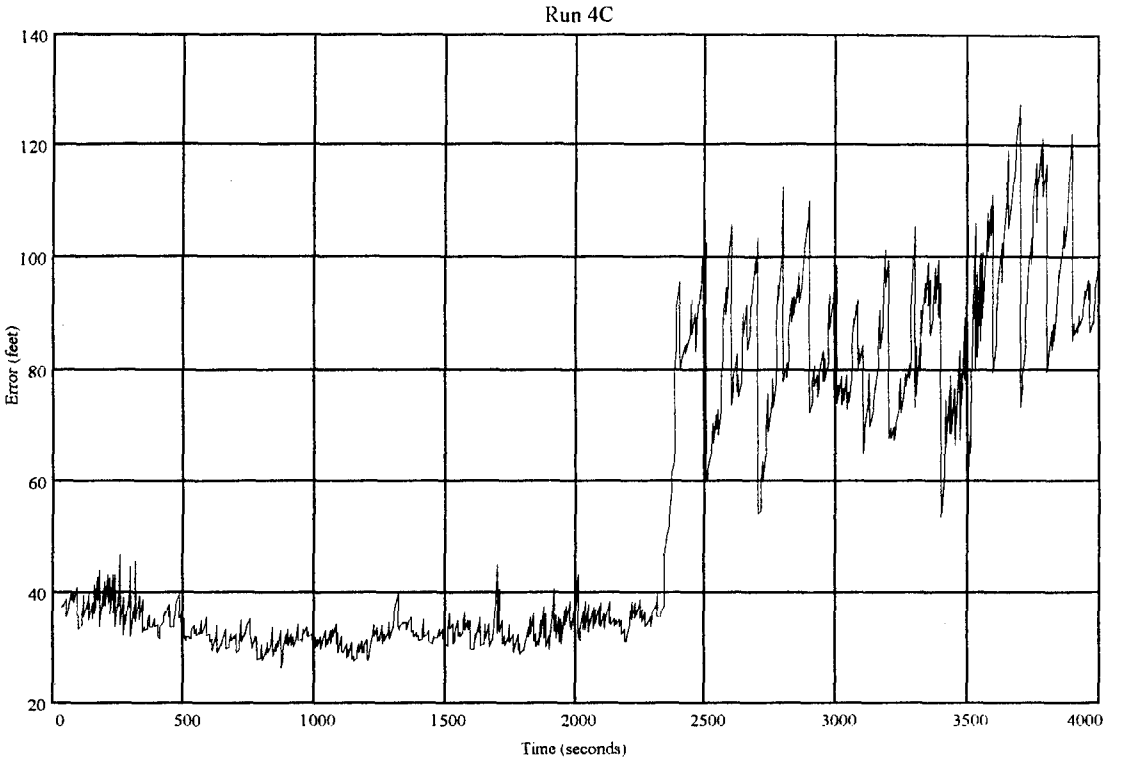


Figure 4 - 16 Run 4C, Federated Filter, Position Error (EGI INS Failure)

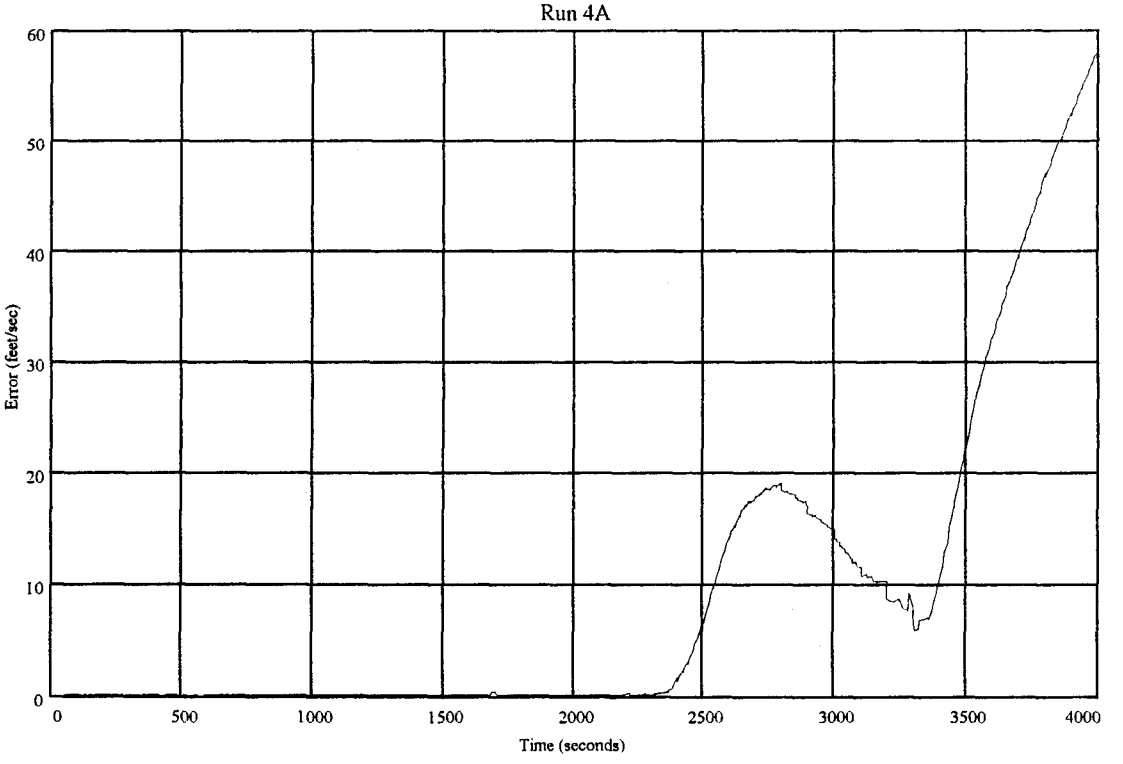


Figure 4 - 17 Run 4A, Centralized Filter, Velocity Error

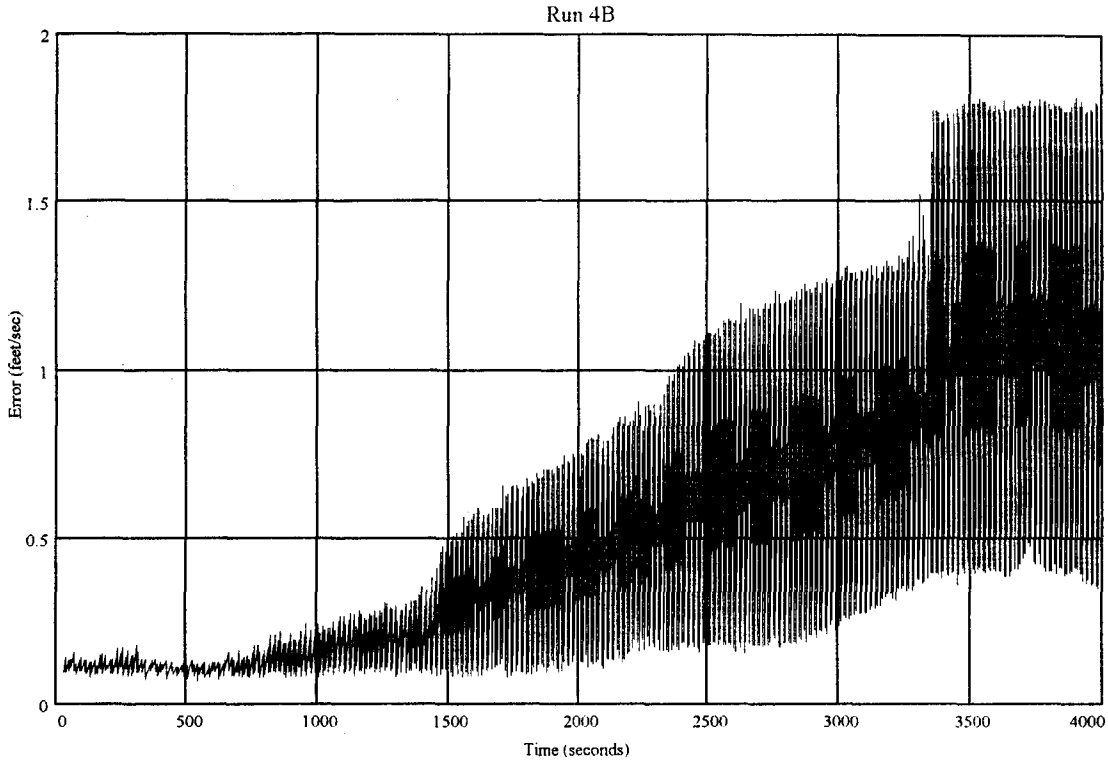


Figure 4 - 18 Run 4B, Federated Filter, Velocity Error (Ref INS Failure)

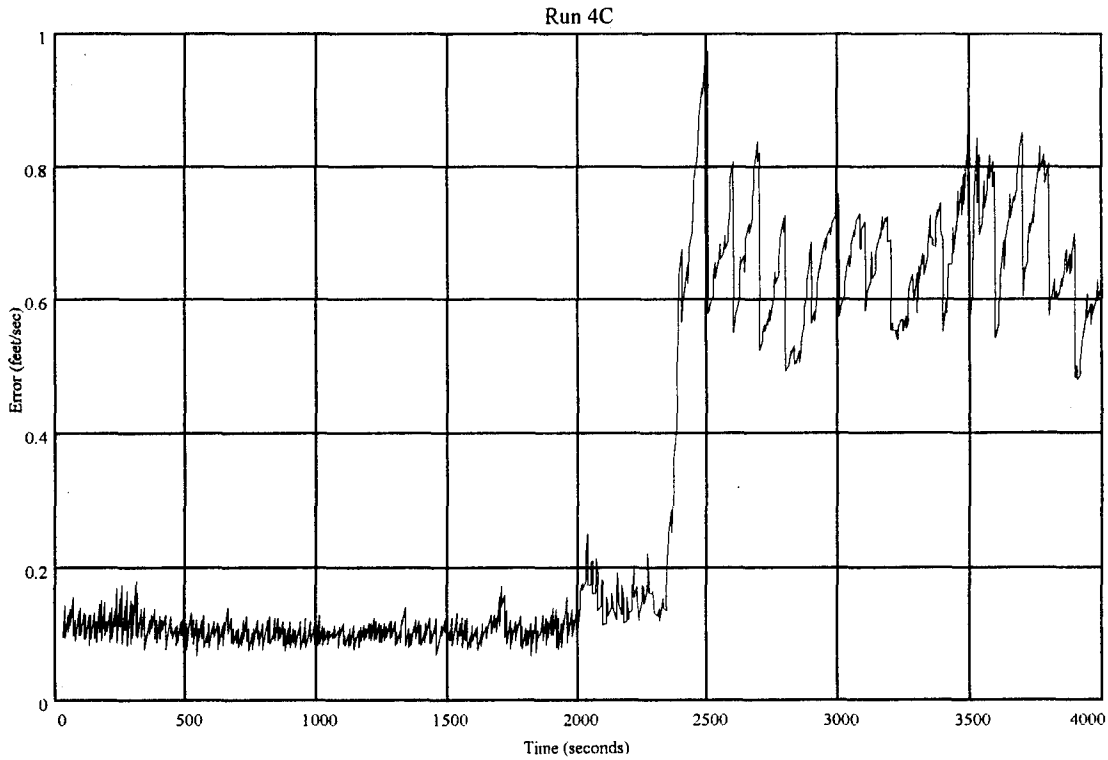


Figure 4 - 19 Run 4C, Federated Filter, Velocity Error (EGI INS Failure)

4.4 Summary

This chapter set out conditions for the computer simulations of the centralized and federated filter in performance comparisons for all-sensor conditions, GPS measurement outages, and GPS and INS failure conditions. Data from the computer simulations were presented, and these data were analysed to characterize the federated filter design.

5. Conclusions and Recommendations

5.1 Introduction

This chapter concludes the research into the EGI retrofit problem by presenting conclusions based on the results of simulations, as well as other aspects of the research such as the use of DKFSIM. Recommendations for future research and development are then presented.

5.2 Conclusions

5.2.1 Simulations

The federated filter model worked. The dual-INS federated filter model operated as predicted in theory. The state estimates and covariances from Local Filter 1 were converted from EGI INS to reference INS and fused with the master filter without any fusion residual rejection or any other processing problem. The conversion was not difficult, computationally speaking, and did not appreciably add to the computation time.

The federated filter model performed as well as the centralized filter model under normal operating conditions. The federated filter performed as well as the centralized filter model under low and high dynamic conditions when no faults were introduced. Figures 4 - 2 and 4 - 3 show that RSS accuracy of the position states was identical for both filters at 35-40 feet. Figures 4 - 4 and 4 - 5 show that the RSS velocity accuracy was somewhat better in the centralized filter, but both were quite accurate.

The federated filter model was comparable to the centralized filter model without GPS inputs. Figures 4- 6 through to 4 - 9 show that the federated filter adapted as well as the centralized filter to the loss of measurement information. The solution provided while GPS measurements were lost was the same for both filters.

The federated filter design provided some failure protection. When configured for the best monitoring for fusion residual rejection (EGI filter fused in last), GPS receiver clock excessive drift and excessive accelerometer noise was detected and dealt with by rejecting EGI filter information. Also, Figures 4 - 10 through 4 - 13 demonstrate that the federated filter solution degraded gracefully to a lower accuracy solution, unlike the centralized filter.

The federated filter design fusion residual rejection scheme could not detect faulty solutions without assistance. Some fault detection was accomplished using the fusion residual rejection algorithm in the master filter, but this only worked when the faulty solution was fused in last. The fusion process works by starting with a local filter solution and adding the other local filter solutions, one at a time. When fusion was done starting with the EGI filter, the good solutions from the other two local filters were rejected. Clearly, this is an inadequate error-detection and isolation scheme.

When the reference INS had an accelerometer failure, there was no way to provide a non-drifting master filter solution. Although the EGI local filter alone operated with a high-accuracy solution, the federated filter was unable to detect the failure and provide the EGI solution.

Measurement residual monitoring, and exploration of residual monitoring techniques applied to this filter type, might increase its error-detection capability. This is an area well worthy of further research.

The federated filter model was an effective method for dual-INS integration. The federated filter design allowed information from two INSs to be integrated into a single solution. This aspect alone of the federated filter design is interesting, and worth further examination and development.

5.2.2 DKFSIM

DKFSIM is a mature, comprehensive and flexible simulation tool. DKFSIM was found to be a well-developed, extensive simulation tool. It has a great deal of capability in simulating both centralized and distributed navigation systems, incorporating extensive flexibility in INS and sensor configurations, as well as failure simulation.

The executable DKFSIM program was easy to operate. The user's manual was straightforward to use, and the installation instructions and software were easy to carry out. All the necessary information and software programs to conduct simulations, from input file descriptions and choices to output data formatting, was provided.

DKFSIM source code was difficult to modify. Although DKFSIM is built in a modular fashion, there is by necessity a great deal of data sharing in a number of ways. This heavy

interdependency made even simple code changes difficult to implement, due to a code change's impact on all other code modules. Code changes were, therefore, complicated and time-consuming.

DKFSIM documentation does not explain how DKFSIM operates, nor what models it uses. The User's Manual is good for operation of the given run-time version. For modification, the code is well-formatted and extensively commented. Unfortunately, there is no manual for users that expands on the models used, algorithms developed, etc. The Distributed Kalman Filter Architectures Phase II report [10] contains a lot of the required information for further DKFSIM development, but is not generally available to users.

DKFSIM was, and will continue to be, well supported. DKFSIM has improved since initial implementation, and newer iterations have expanded the scope of systems modeled. Wright Laboratories continues to be interested in distributed filters, and will likely continue to encourage development in DKFSIM through contract work and sponsorship of AFIT research such as this thesis.

5.3 Recommendations

Additional simulations using the federated filter design of this thesis should be conducted to further characterize the design. Only some representative failures were carried out in the author's research, due to time constraints. Additional work with the run-time version, simulating different conditions of operation and failures, would help to further determine the design characteristics.

Fault detection algorithms added to the federated filter should be explored, for failure detection and isolation. For the fusion residual monitoring, it was suggested [22] that a more robust residual fault detection system could be designed, where the order of fusion was varied in order to isolate faults. This type of algorithm would be relatively simple to design and implement, and may greatly improve fault detection capability of the federated filter.

In addition, measurement residual monitoring algorithms could be implemented to increase the failure detection capability of the federated filter design.

DKFSIM should continue to be used for distributed filter research. DKFSIM is an excellent tool, and provides a flexible enough environment to accommodate future growth.

5.4 Summary

This thesis presented the problem of retrofitting an EGI into an existing navigation system. A federated filter was designed to work in the problem, and using simulations the filter characteristics were explored. The federated filter design worked in nearly optimal fashion, and was shown to have some attractive fault detection and isolation features.

The creation of a workable filter for the EGI retrofit problem indicates the potential of the federated filter in air navigation systems. Hopefully, research will continue into federated filter design, eventually leading to testing in an aircraft. Leadership in research of this nature will ensure a strong future defense capability for the USAF.

Bibliography

1. Lawrence, P.J. Memorandum to ASC/ASR, Wright-Patterson AFB, 24 May 1995.

2. Glazer, B.G. *GPS Receiver Operation*, Global Positioning System, The Institute of Navigation, 1980.

3. Lewantowicz, Z.H., Colonel, USAF. *Deep Integration of GPS, INS, SAR, and Other Sensor Information*, Draft Paper.

4. Carlson, N.A. *Federated Filter for Fault-Tolerant Integrated Navigation Systems*, IEEE Position, Location, and Navigation Symposium, 1988.

5. Maybeck, P.S. Stochastic Models, Estimation, and Control, Volume I, Navtech Book and Software Store, 1994.

6. Cox, D.B. *Integration of GPS with Inertial Navigation Systems*, GPS Papers Vol.1, The Institute of Navigation, 1980.

7. Solomon, J.K. *Analysis of the F-16 Aircraft's Integrated Navigation Implementation*. Final Technical Memorandum, Avionics Directorate, Wright Laboratories, WPAFB, December 1994.

8. Carlson, N.A. *Information-Sharing Approach to Federated Kalman Filtering*. Proceedings of the 1988 National Aerospace and Engineering Conference (NAECON 88), May 1988.

9. Carlson, N.A. *Federated Kalman Filter Simulation Results*, Proceedings of 49th Annual Meeting of the Institute of Navigation, Boston, MA, June 1993.

10. TAU Corporation. *Common Kalman Filter Development*. Contract No. F33615-86-C-1047, Document No. WRDC-TR-89-1152. Los Gatos, CA, Feb 1990.

11. Carlson, N.A. *Distributed Kalman Filter Architectures Phase II, Part B*, Wright Laboratories Report WL-TR-95-zzzz (Draft), April 1995.

12. Berarducci, Michael. Program Manager, Avionics Directorate, Wright Laboratories. Briefing and Tour, October 1995.

13. Gao, Y., Krakiwsky, E.J., and McLellan, J.F. *Experience with the Application of Federated Filter Design to Kinematic GPS Positioning*, IEEE Position, Location and Navigation Symposium, 1992.

14. Lawrence, P.J., Captain, USAF. *Comprarison of a Distributed Kalman Filter Versus a Centralized Kalman Filter with Fault Detection Considerations*, AFIT MS Thesis, Wright-Patterson AFB, 30 July 1993.

15. Tazartes, D.A. and Mark, J.G. *Integration of GPS Receivers into Existing Inertial Navigation Systems*, Navigation, Vol. 35,No. 1, Spring 1988.

16. Honeywell Corp. H-764 System Description, Honeywell Military Avionics, St Petersburg, Florida, 1995.

17. Morgan, Ken. Avionics Engineer, Honeywell Military Systems. Telephone Interview, July 1995.

18. Wells, D. and others. Guide to GPS Positioning. Canadian GPS Associates, University of New Brunswick Graphic Services, May 1987.

19. Stimson, G.W. Introduction to Airborne Radar, Hughes Aircraft Company, 1983.

20. Maybeck, P.S. Stochastic Models, Estimation, and Control, Volume 2, Navtech Book and Software Store, 1994.

21. Ross, S.M. Introduction to Probability Models, Academic Press, Inc., 1993.

22. Carlson, N.A. User's Manual for the Distributed Kalman Filter Simulator (DKFSIM), Integrity Systems Inc., Belmont, MA, July 1994.

23. Musick, S.H. *PROFGEN - A Computer Program for Generating Flight Profiles*, Technical Report AFAL-TR-76-247, Air Force Wright Aeronautical Laboratories, 1977

24. Carlson, N.A. President, Integrity Systems, Inc. Facsimile Correspondence, 6 October 1995.

25. DeLory, S.J. and others. AFIT Course EENG 735 Final Report, May 1995.

26. Carlson, N.A. President, Integrity Systems, Inc. Telephone Interview, October 1995.

Appendix A: Performance Runs 1A and 1B

This appendix shows data outputs from DKFSIM for Runs 1A and 1B (no faults) performance run. The first nine plots are the three position, three velocity and three tilt states from simulation of the centralized filter design. The second nine plots are the corresponding states from simulation of the federated filter design.

Each plot has the same format. The solid line centered roughly around zero is the mean of 20 Monte Carlo simulations. The outer solid black lines are the one-sigma values of the 20 Monte Carlo simulations, derived from the output state estimates. The dashed black lines are the plus and minus values of the means of the filter-predicted one-sigma value.

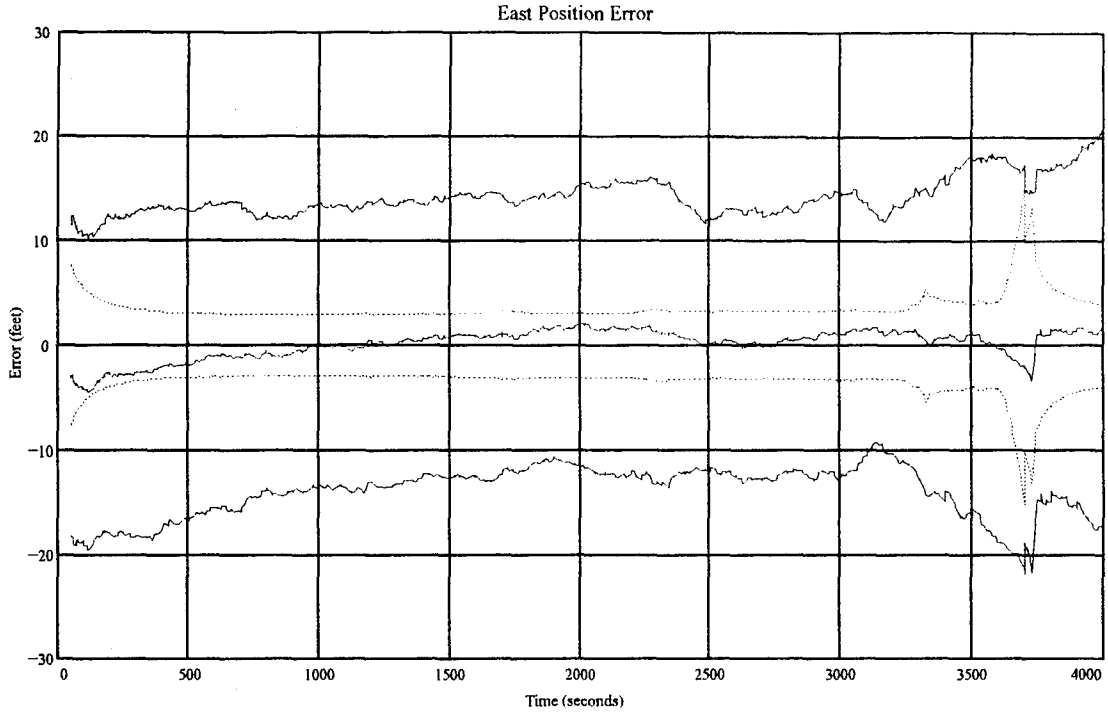


Figure A - 1 Run 1A, Centralized Filter, East Position State

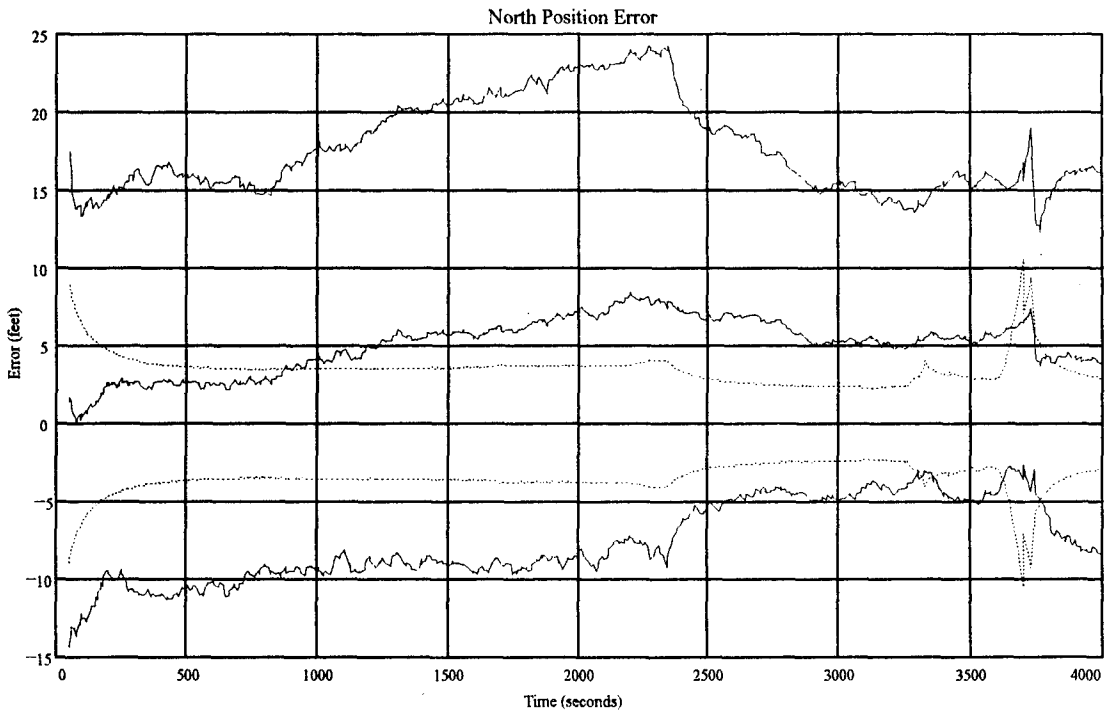


Figure A - 2 Run 1A, Centralized Filter, North Position State

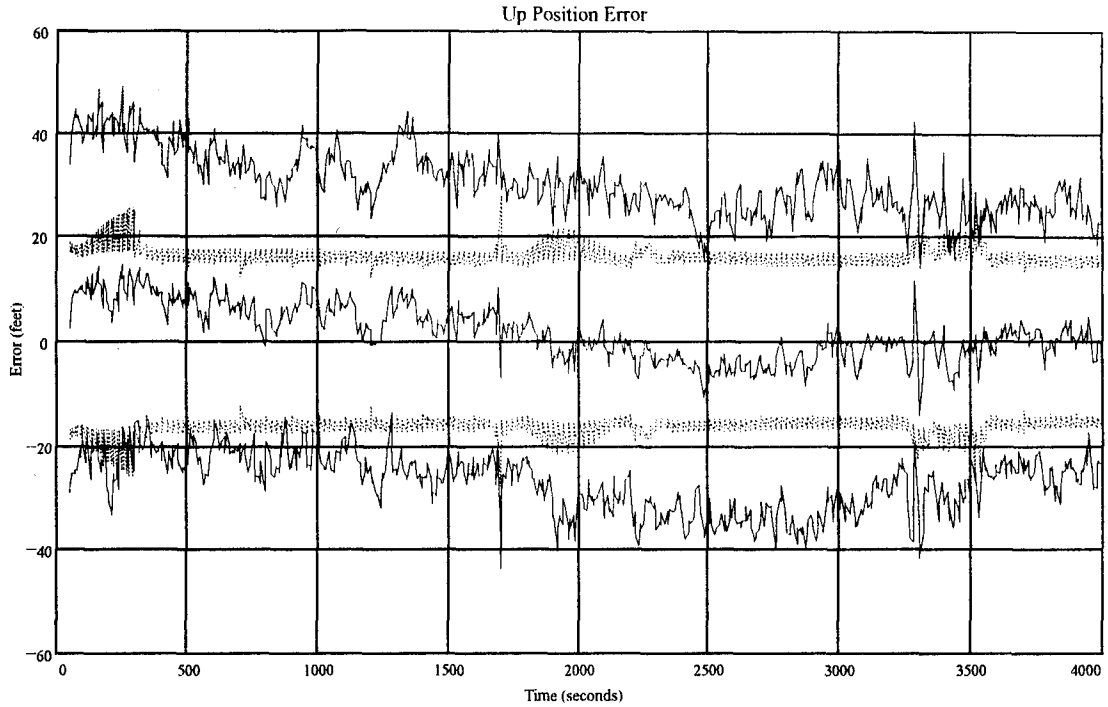


Figure A - 3 Run 1A, Centralized Filter, Up Position State

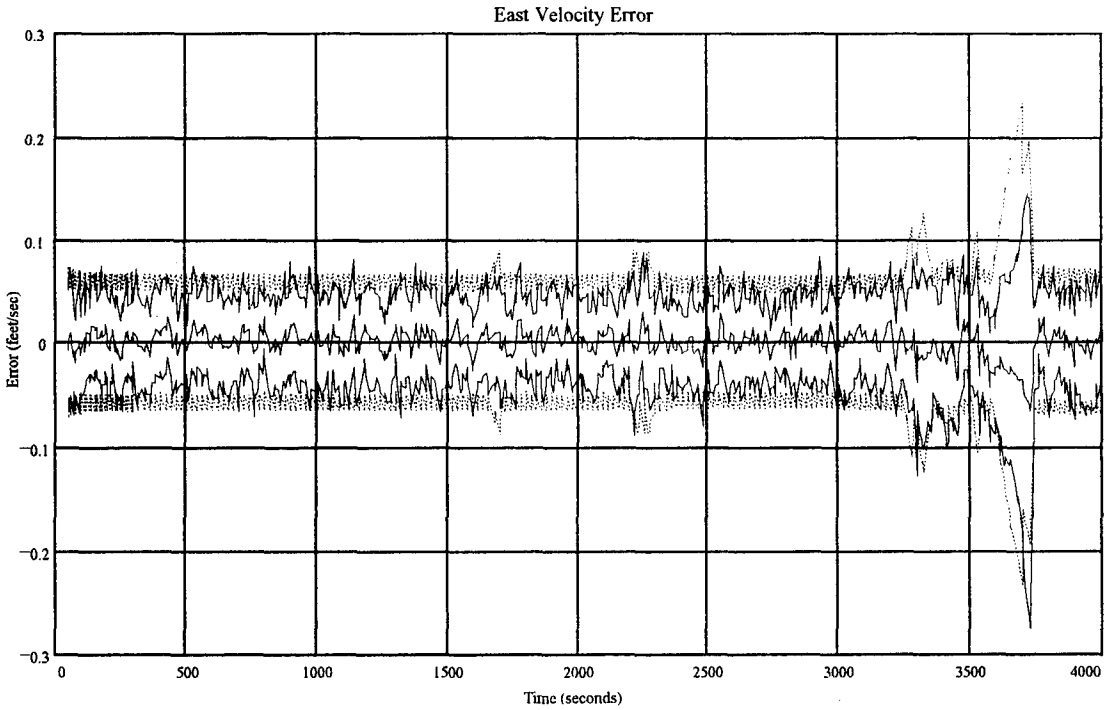


Figure A - 4 Run 1A, Centralized Filter, East Velocity State

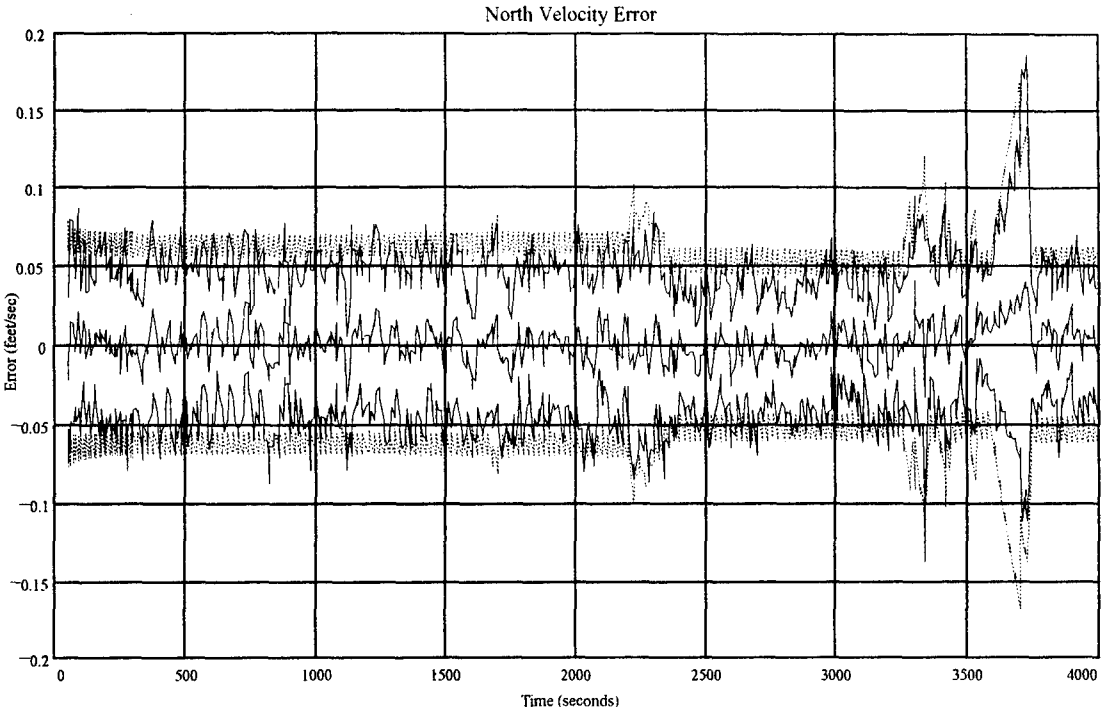


Figure A - 5 Run 1A, Centralized Filter, North Velocity State

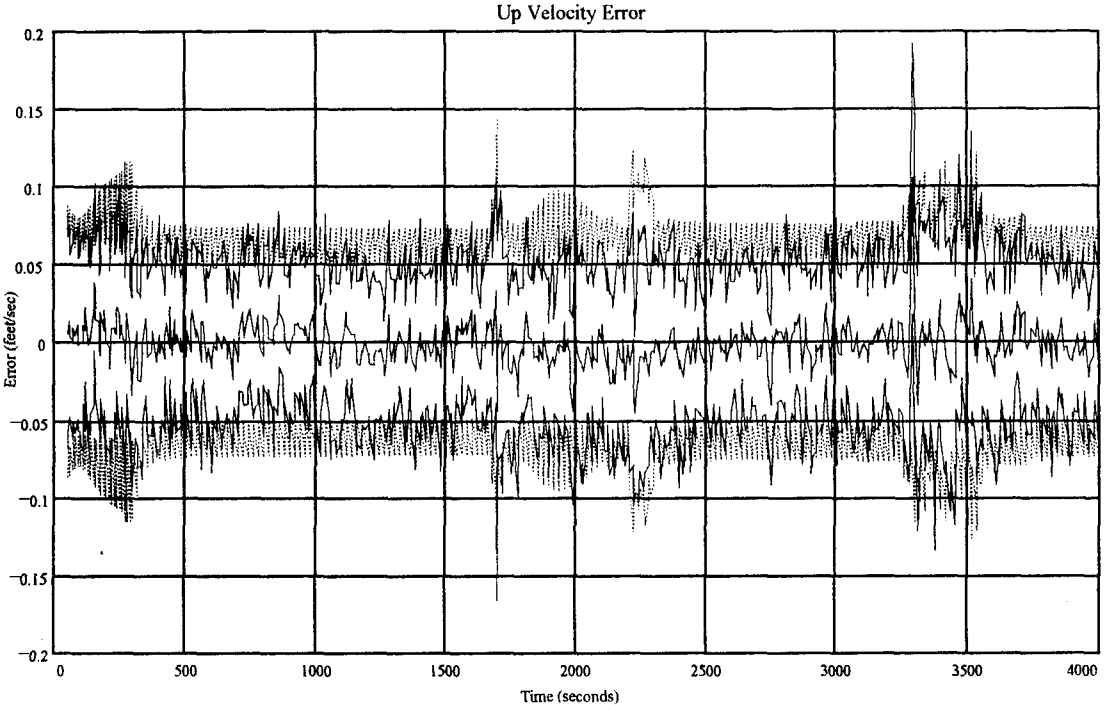


Figure A - 6 Run 1A, Centralized Filter, Up Velocity State

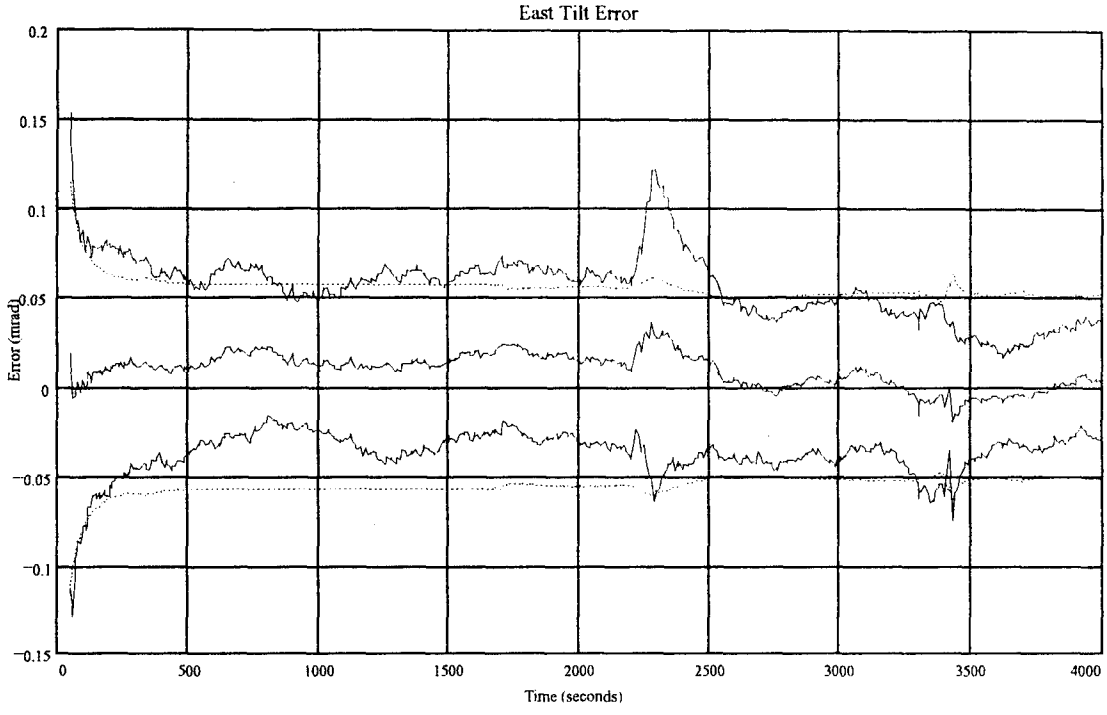


Figure A - 7 Run 1A, Centralized Filter, East Tilt State

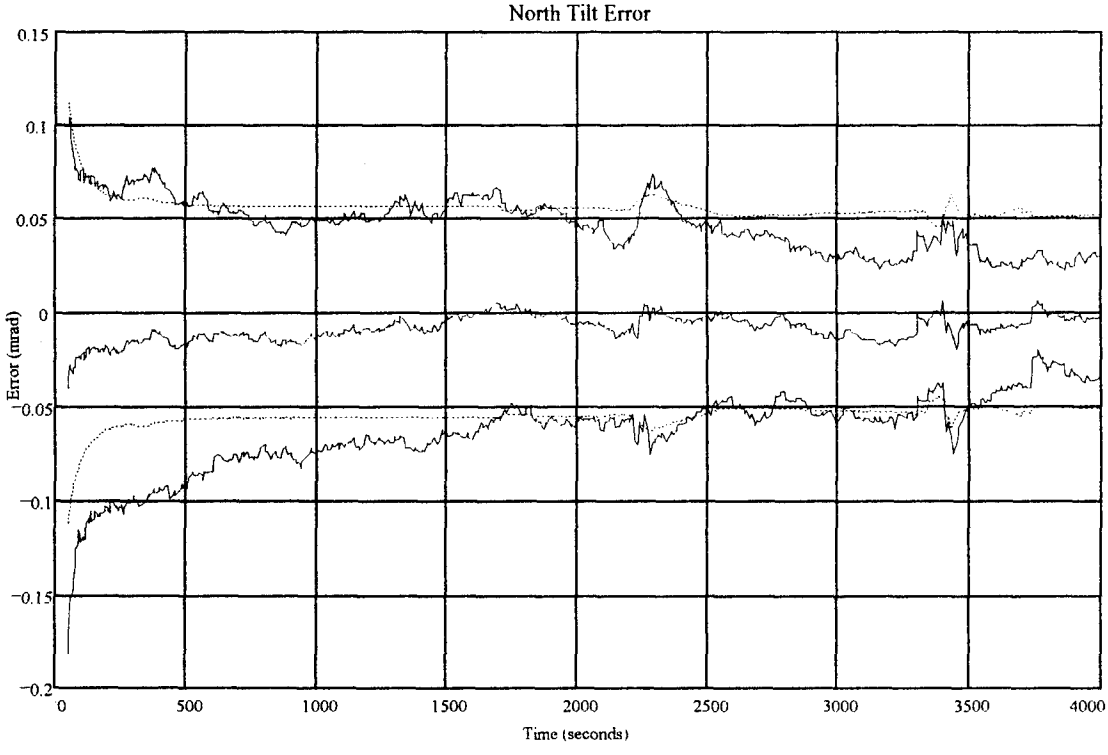


Figure A - 8 Run 1A, Centralized Filter, North Tilt State

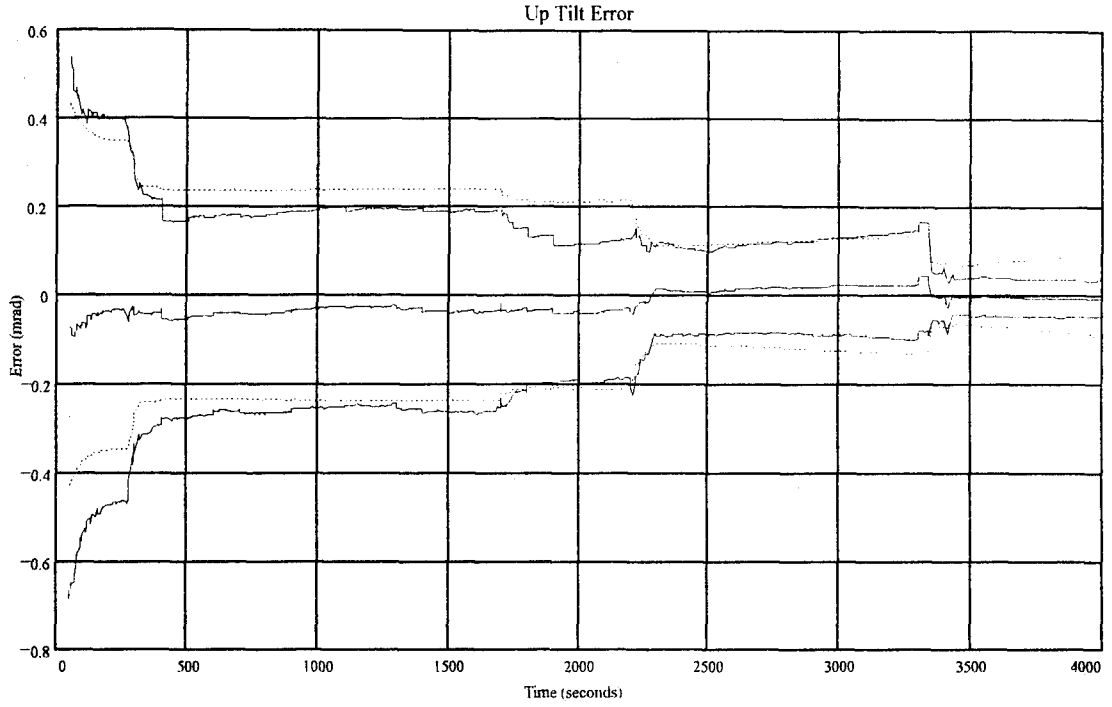


Figure A - 9 Run 1A, Centralized Filter, Up Tilt State

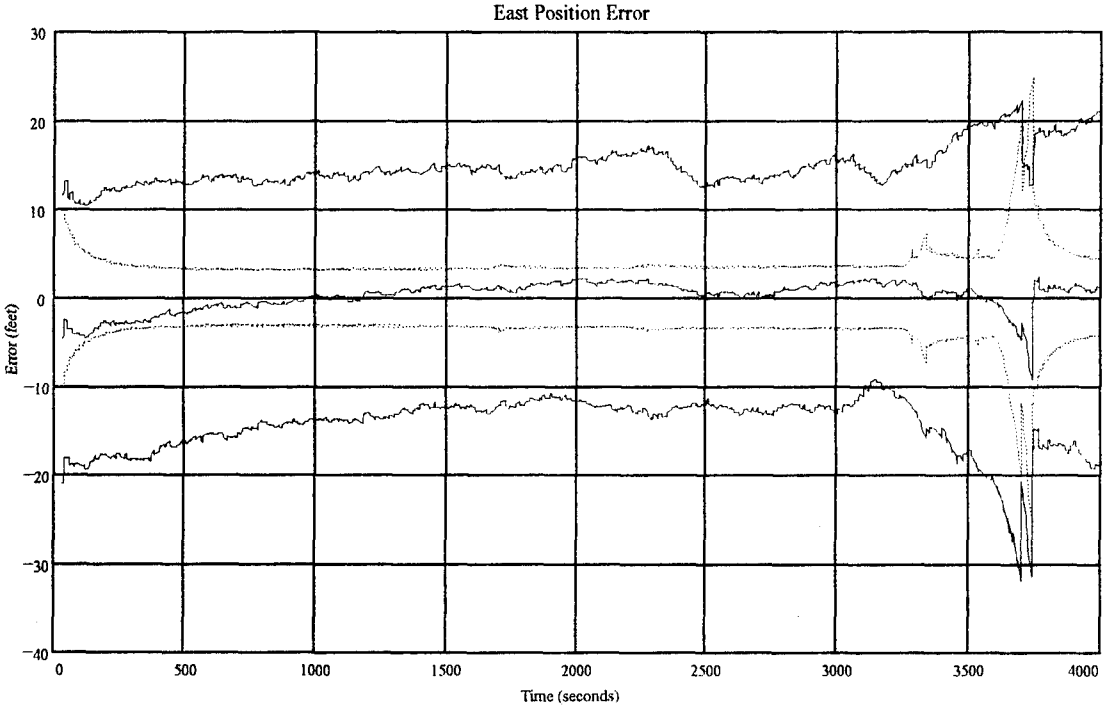


Figure A - 10 Run 1B, Federated Filter, East Position State

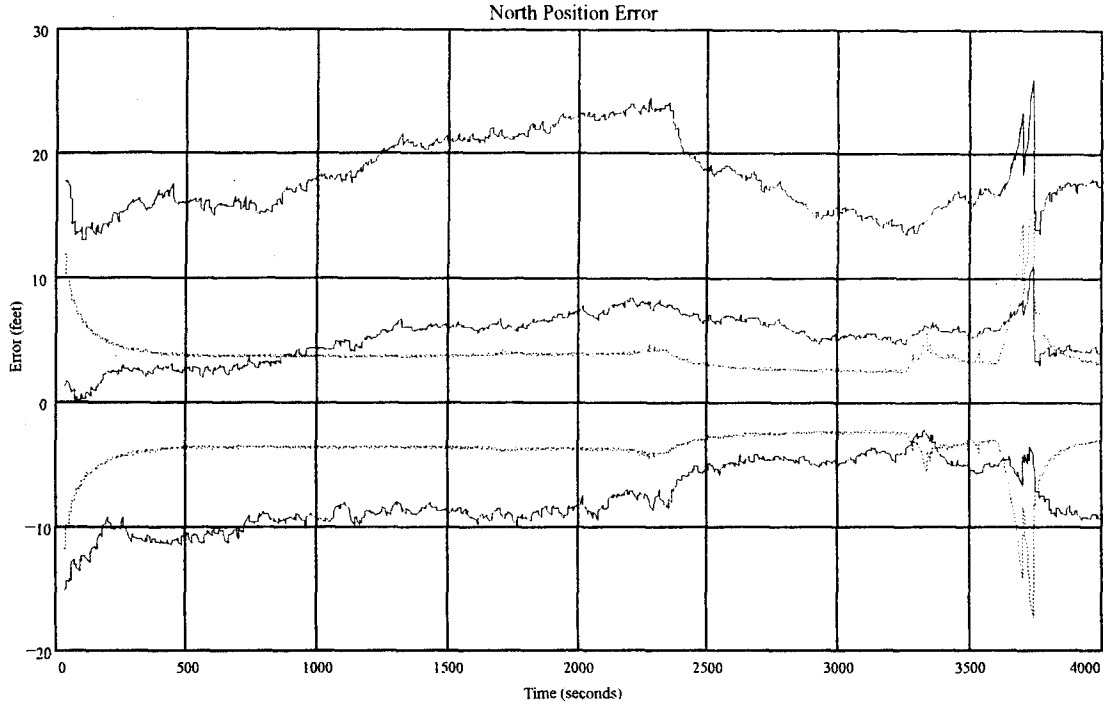


Figure A - 11 Run 1B, Federated Filter, North Position State

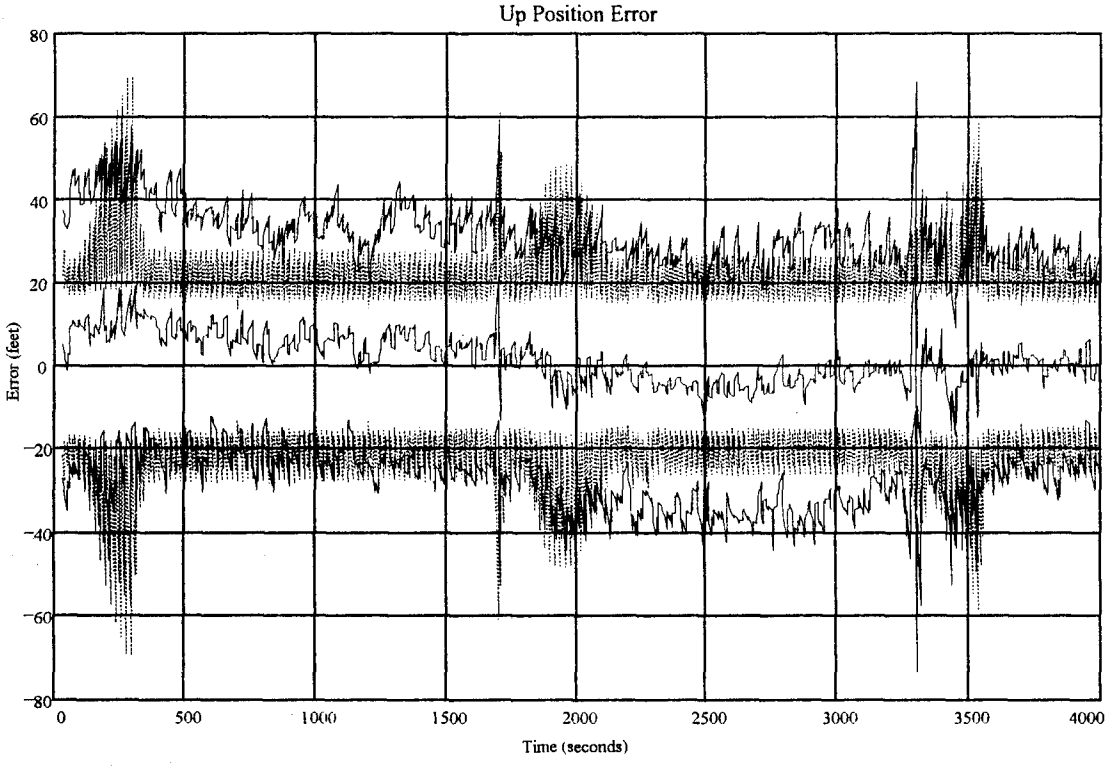


Figure A - 12 Run 1B, Federated Filter, Up Position State

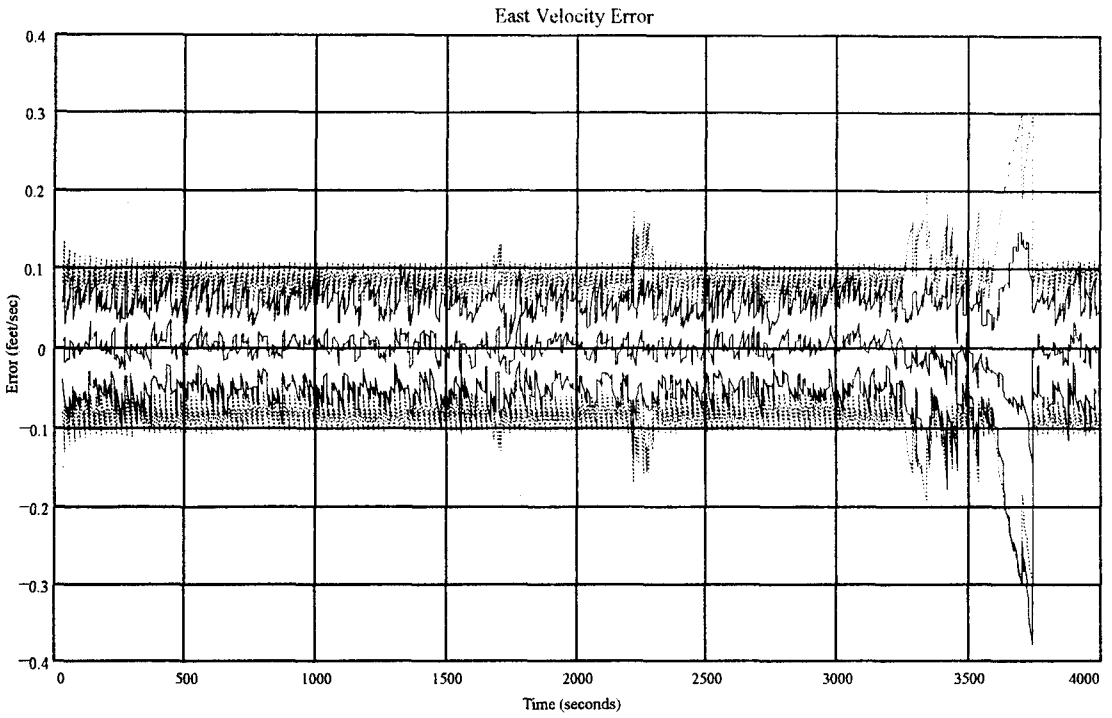


Figure A - 13 Run 1B, Federated Filter, East Velocity State

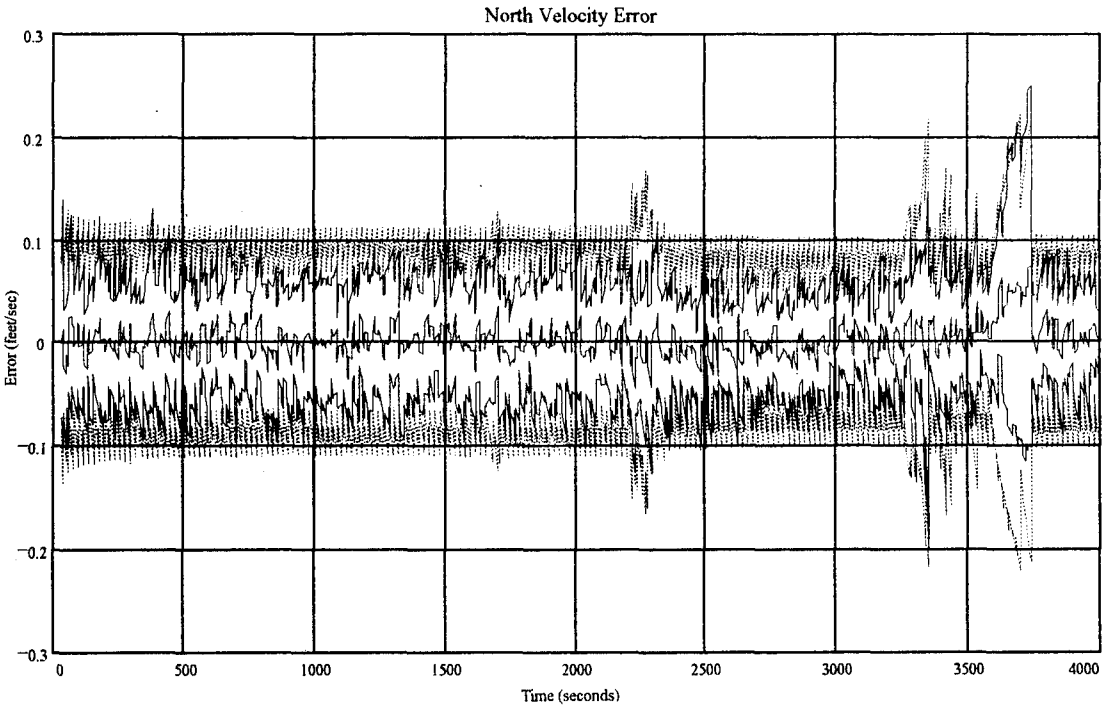


Figure A - 14 Run 1B, Federated Filter, North Velocity State

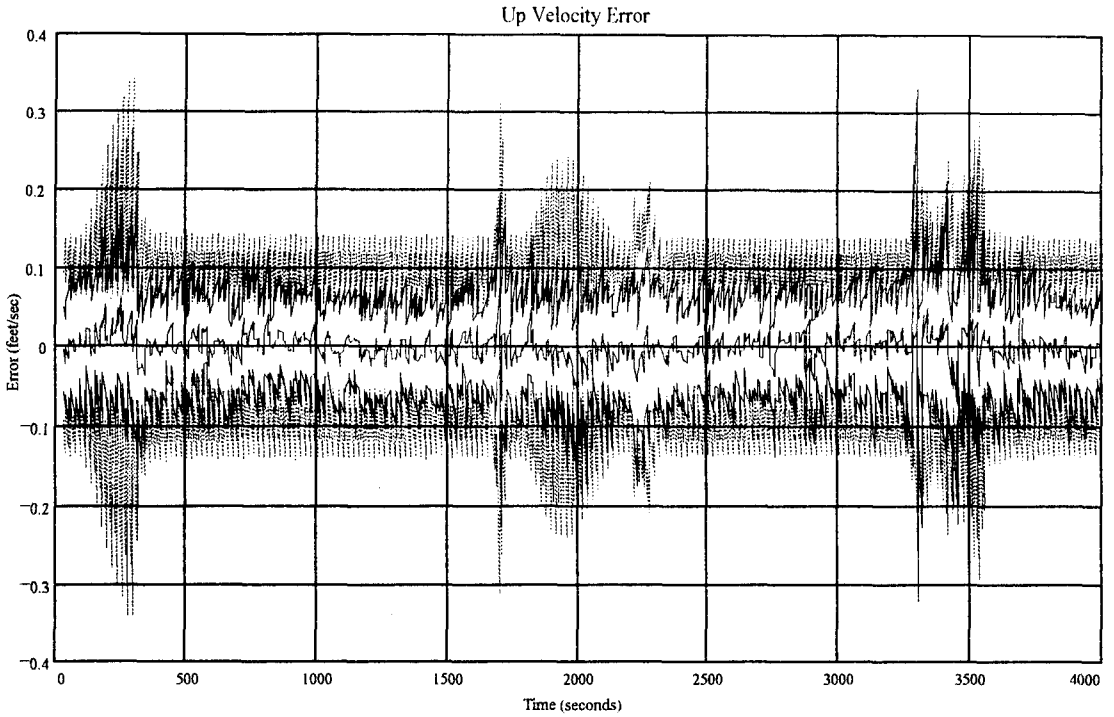


Figure A - 15 Run 1B, Federated Filter, Up Velocity State

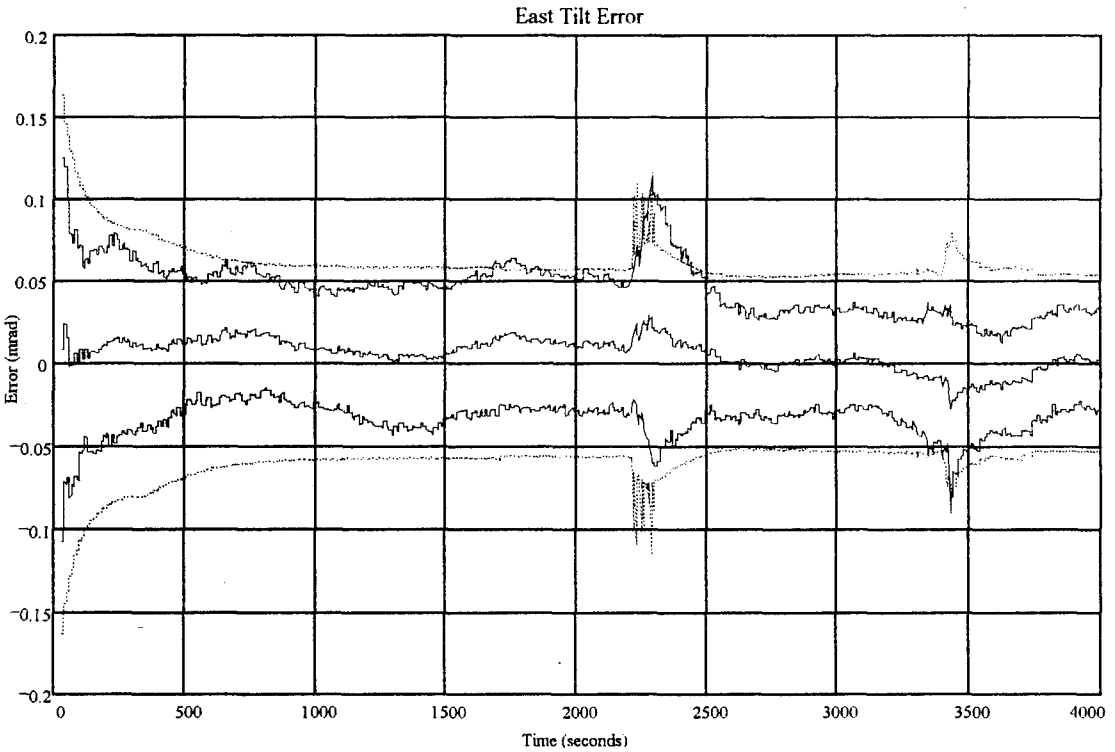


Figure A - 16 Run 1B, Federated Filter, East Tilt State

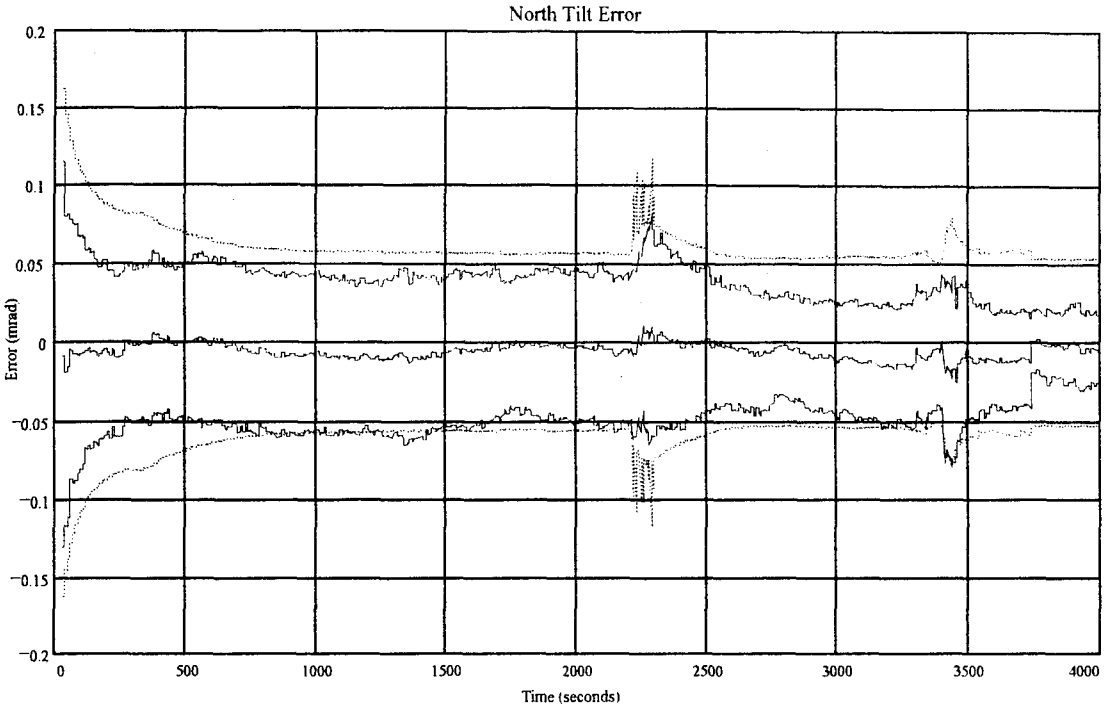


Figure A - 17 Run 1B, Federated Filter, North Tilt State

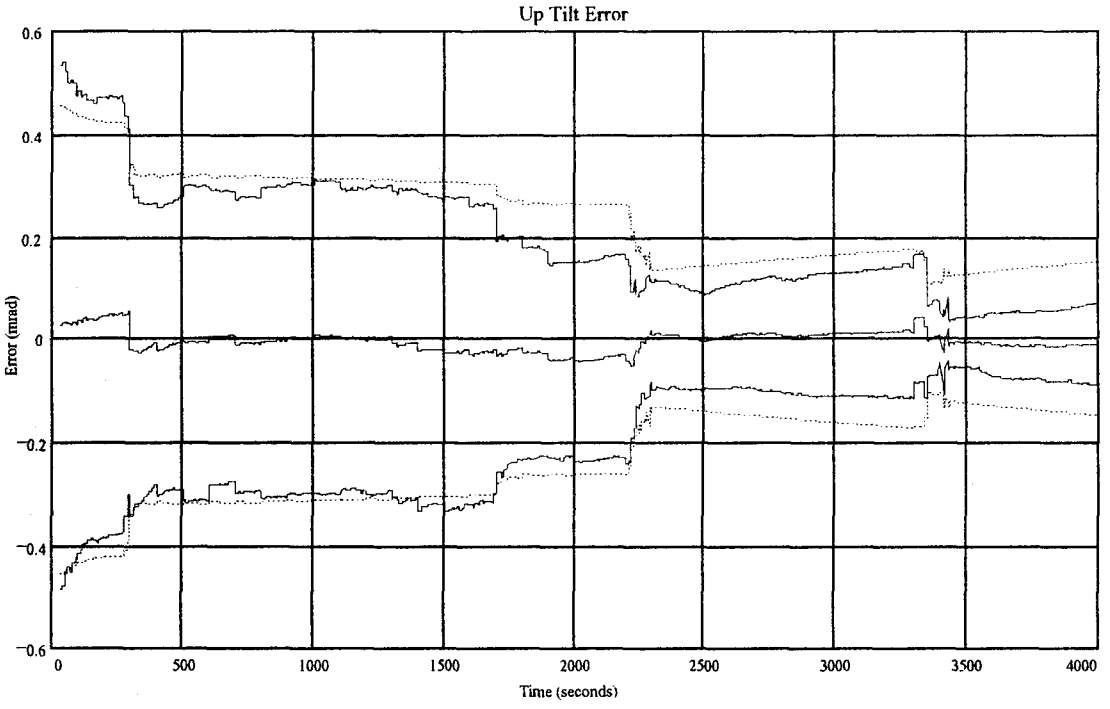


Figure A - 18 Run 1B, Federated Filter, Up Tilt State

Appendix B: DKFSIM Run 1A and 1B Input Parameters

The FORTRAN input files used to define operation of the truth models in DKFSIM are contained in this appendix. In each file, a title, date, and author is given, followed by values for input variables. The title is IN*XXX*, where *XXX* corresponds to the first three characters of the truth model name, eg. ININU, INGPS, etc. Finally, the input variables are defined for each file. The file ININU was used to control the reference INU truth model, and the file INJNU was used to control the EGI INU truth model.

```
ININU: INU TRUTH MODEL INPUT PARAMETERS                    06/22/94 NAC
 .ST2  * STANDARD ERRORS, 2ND-ORD *

INU CONTROL AND ERROR PARAMETERS:
&INUPR
ISDINU =   4,
OINUE  =   T,    OINUV  =   F,  DTINUO =  5.,
DTCALN =  0.0,   DTFALN =  0.0,  DTINUI =  0.0,
BARDMR =   1,    LWNAV  =   F,

SGPN0(1) =  500.,  500.,  500.,
SGVN0(1) =   2.0,   2.0,   2.0,
SGTHNC(1) = 200E-6, 200E-6, 500E-6,
SGTHNF(1) = 200E-6, 200E-6, 500E-6,

SGABR =  1.283E-3, TCABR =  3600.,
SGASR =   60.E-6, SGAMR =  50.E-6,
SGGBR =  3.889E-8, TCGBR =  3600.,
SGGSR =    2.E-6, SGGMR =  15.E-6, SGGADR = 8.6E-11,
NPHIR = 3.4028E-13, NVELR =  .0001,

SGGRNR(1) = .802E-3, .802E-3, 1.123E-3,
DCGRNR(1) = 5.804E4, 5.804E4, 3.696E5,

KBARPR = 02, KBARVR = 1.03E-4, KBARAR = 0.0E-6,
&

INU FAILURE PARAMETERS:
&INUFPR
TFINU  = 2000.,
IFACCL =   0,
DAMF0  =  .032,  DAMFR =  .0032,
IFGYRO =   0,
DGMF0  = 4.E-6, DGMFR =  4.E-8,
INUFTP =   0,
&
-------------------------------------------------------------
C  * INUPR DEFINITIONS:
C    ALNMOD  CH          INU ALIGNMENT MODE (GROUND/IN-AIR)
C    BARDMR  IN          BARO INU-DAMPING INDEX (0=NONE, 1=DAMP)
C    DCGRNR  RL  3       CORR DISTANCES INU GRAV PERTNS NAV, RW
C    DTCALN  RL          SIMULATED INU COARSE ALIGN INTERVAL
C    DTFALN  RL          SIMULATED INU FINE ALIGN INTERVAL
C    DTINUI  RL          DEFAULT TIME INTERVAL TO INU TURN-ON
C    DTINUO  RL          TIME INTERVAL FOR INU DATAFILE OUTPUTS
C    ISDINU  RL          INPUT SEED FOR INU RANDOM NO. GENR
C    KBARAR  RL          BARO DAMPING GAIN #1, FOR ACCELERN, RW
C    KBARPR  RL          BARO DAMPING GAIN #1, FOR POSITION, RW
C    KBARVR  RL          BARO DAMPING GAIN #2, FOR VELOCITY, RW
C    LWNAV   LG          INU WANDER-NAV FRAME OUTPUT FLAG
C    NPHIR   RL          INU TILT NOISE POWER DENISTY, RW
```

```
C   NVELR  RL          INU VELOCITY NOISE POWER DENISTY, RW
C   OINUE  LG          OUTPUT SWITCH FOR INU ERRORS
C   OINUV  LG          OUTPUT SWITCH FOR INU VARIABLES
C   SGABR  RL          SIGMA INU ACCELR BIAS ERRORS, RW
C   SGAMR  RL          SIGMA INU ACCELR MISALIGNMENT ERRS, RW
C   SGASR  RL          SIGMA INU ACCELR SCALE FACTOR ERRS, RW
C   SGGADR RL          SIGMA INU GYRO ACCL-SENS DRIFT COEFS, RW
C   SGGBR  RL          SIGMA INU GYRO BIAS DRIFT RATES, RW
C   SGGRNR RL 3        SIGMA INU GRAV PERTURBNS NAV, RW
C   SGGMR  RL          SIGMA INU GYRO INPUT-AXIS MISALMTS, RW
C   SGGSR  RL          SIGMA INU GYRO SCALE FACTOR ERRS, RW
C   SGPN0  RL 3        SIGMA INU INIT POSITION ERRORS NAV, RW
C   SGTHNC RL 3        SIGMA INU CRS ALN TILT ERRORS NAV, RW
C   SGTHNF RL 3        SIGMA INU FINE ALN TILT ERRORS NAV, RW
C   SGVN0  RL 3        SIGMA INU INIT VELOCITY ERRORS NAV, RW
C   TCABR  RL          CORR TIME INU ACCELR BIASES, RW
C   TCGBR  RL          CORR TIME INU GYRO RATE BIASES, RW
C
C  * INUFPR DEFINITIONS:
C   DAMF0  RL          ACCELR MSMT FAILURE INITL VALUE
C   DAMFR  RL          ACCELR MSMT FAILURE GROWTH RATE
C   DGMF0  RL          GYRO RATE MSMT FAILURE INITL VALUE
C   DGMFR  RL          GYRO RATE MSMT FAILURE GROWTH RATE
C   IFACCL IN          INDEX OF FAILED ACCELR AXIS
C   IFGYRO IN          INDEX OF FAILED GYRO AXIS
C   INUFTP IN          INU FAILURE TYPE (0 BIAS, 1 RANDOM)
C   TFINU  DP          TIME OF INU FAILURE
```

```
INJNU: JNU TRUTH MODEL INPUT PARAMETERS                    06/22/94 NAC
 .ST2  * STANDARD ERRORS, 2ND-ORD *

 INU CONTROL AND ERROR PARAMETERS:
 &JNUPR
 ISDINU =  4,
 OINUE  =   T,    OINUV  =   F,  DTINUO =  5.,
 DTCALN =  0.0,   DTFALN =  0.0,  DTINUI =  0.0,
 BARDMR =   1,    LWNAV  =   F,

 SGPN0(1) =   500.,   500.,   500.,
 SGVN0(1) =   2.0,    2.0,    2.0,
 SGTHNC(1) = 200E-6,  200E-6,  500E-6,
 SGTHNF(1) = 200E-6,  200E-6,  500E-6,

 SGABR =   1.283E-3,  TCABR =   3600.,
 SGASR =    60.E-6,   SGAMR =   50.E-6,
 SGGBR =   3.889E-8,  TCGBR =   3600.,
 SGGSR =    2.E-6,    SGGMR =   15.E-6,  SGGADR = 8.6E-11,
 NPHIR = 3.4028E-13,  NVELR =   .0001,

 SGGRNR(1) =  .802E-3,  .802E-3,  1.123E-3,
 DCGRNR(1) =  5.804E4,  5.804E4,  3.696E5,

 KBARPR = .02, KBARVR = 1.03E-4, KBARAR = 0.0E-6,
```

```
&

INU FAILURE PARAMETERS:
&JNUFPR
TFINU  = 2000.,
IFACCL =    0,
DAMF0  =  .032,  DAMFR =  .0032,
IFGYRO =    0,
DGMF0  = 4.E-6,  DGMFR =  4.E-8,
INUFTP =    0,
&
-----------------------------------------------------------------
C  * INUPR DEFINITIONS:
C    ALNMOD  CH          INU ALIGNMENT MODE (GROUND/IN-AIR)
C    BARDMR  IN          BARO INU-DAMPING INDEX (0=NONE, 1=DAMP)
C    DCGRNR  RL  3       CORR DISTANCES INU GRAV PERTNS NAV, RW
C    DTCALN  RL          SIMULATED INU COARSE ALIGN INTERVAL
C    DTFALN  RL          SIMULATED INU FINE ALIGN INTERVAL
C    DTINUI  RL          DEFAULT TIME INTERVAL TO INU TURN-ON
C    DTINUO  RL          TIME INTERVAL FOR INU DATAFILE OUTPUTS
C    ISDINU  RL          INPUT SEED FOR INU RANDOM NO. GENR
C    KBARAR  RL          BARO DAMPING GAIN #1, FOR ACCELERN, RW
C    KBARPR  RL          BARO DAMPING GAIN #1, FOR POSITION, RW
C    KBARVR  RL          BARO DAMPING GAIN #2, FOR VELOCITY, RW
C    LWNAV   LG          INU WANDER-NAV FRAME OUTPUT FLAG
C    NPHIR   RL          INU TILT NOISE POWER DENISTY, RW
C    NVELR   RL          INU VELOCITY NOISE POWER DENISTY, RW
C    OINUE   LG          OUTPUT SWITCH FOR INU ERRORS
C    OINUV   LG          OUTPUT SWITCH FOR INU VARIABLES
C    SGABR   RL          SIGMA INU ACCELR BIAS ERRORS, RW
C    SGAMR   RL          SIGMA INU ACCELR MISALIGNMENT ERRS, RW
C    SGASR   RL          SIGMA INU ACCELR SCALE FACTOR ERRS, RW
C    SGGADR  RL          SIGMA INU GYRO ACCL-SENS DRIFT COEFS, RW
C    SGGBR   RL          SIGMA INU GYRO BIAS DRIFT RATES, RW
C    SGGRNR  RL  3       SIGMA INU GRAV PERTURBNS NAV, RW
C    SGGMR   RL          SIGMA INU GYRO INPUT-AXIS MISALMTS, RW
C    SGGSR   RL          SIGMA INU GYRO SCALE FACTOR ERRS, RW
C    SGPN0   RL  3       SIGMA INU INIT POSITION ERRORS NAV, RW
C    SGTHNC  RL  3       SIGMA INU CRS ALN TILT ERRORS NAV, RW
C    SGTHNF  RL  3       SIGMA INU FINE ALN TILT ERRORS NAV, RW
C    SGVN0   RL  3       SIGMA INU INIT VELOCITY ERRORS NAV, RW
C    TCABR   RL          CORR TIME INU ACCELR BIASES, RW
C    TCGBR   RL          CORR TIME INU GYRO RATE BIASES, RW
C
C  * INUFPR DEFINITIONS:
C    DAMF0   RL          ACCELR MSMT FAILURE INITL VALUE
C    DAMFR   RL          ACCELR MSMT FAILURE GROWTH RATE
C    DGMF0   RL          GYRO RATE MSMT FAILURE INITL VALUE
C    DGMFR   RL          GYRO RATE MSMT FAILURE GROWTH RATE
C    IFACCL  IN          INDEX OF FAILED ACCELR AXIS
C    IFGYRO  IN          INDEX OF FAILED GYRO AXIS
C    INUFTP  IN          INU FAILURE TYPE (0 BIAS, 1 RANDOM)
C    TFINU   DP          TIME OF INU FAILURE
```

```
INBAR: BARO ALTIMETER TRUTH MODEL INPUT PARAMETERS          04/18/94 NAC
 .STD  * STANDARD ERRORS *

 &BARPR
 ISDBAR =  4,
 OBAREV =  F,   DTBARO =  10.,
 DTBARI =  0.,

 SGBBR  =  500.,
 SGBSFR =  .03,
 DCBAR  = 1.25E6,

 SGBTDR =  .25,
 SGBSPR = 1.5E-4,
 SGBNR  =  10.,
 &

-------------------------------------------------------------
C  * DEFINITIONS:
C    DCBAR  RL          CORR DIST BARO ATMOS VARIATIONS, RW
C    DTBARI RL          DEFAULT TIME INTERVAL TO BARO TURN-ON
C    DTBARO RL           TIME INTERVAL FOR BARO DATAFILE OUTPUTS
C    ISDBAR IN          INPUT SEED FOR BARO RANDOM NO. GENR
C    OBAREV LG           OUTPUT SWITCH FOR BARO ERRORS & VARS
C    SGBBR  RL          SIGMA BARO PRESSURE ALT BIAS, RW
C    SGBNR  RL          SIGMA BARO RANDOM MSMT NOISE, RW
C    SGBSPR RL          SIGMA BARO STATIC PRESS COEF, RW
C    SGBSFR RL          SIGMA BARO ALT SCALE FACTOR ERR, RW
C    SGBTDR RL           SIGMA BARO ALT TIME DELAY, RW
```

```
INGPS: GPS TRUTH MODEL INPUT PARAMETER FILE                 06/22/94 NAC
 .SMF  * STANDARD ERRORS, MED GDOP, FULL COVERAGE *

*GPS MODEL CONTROL AND MSMT ERROR TRUTH PARAMETERS:
 &GPSPR
 ISDGPS =  4,
 USEPR  =  T,   USEPRR =  T,
 OGPSE  =  F,   OGPSV  =  F,   OGSAT =  F,
 DTGPSO =  10.,  DTGPSI =  0.,

 SGCP0T =  1000.,
 SGCFT  =  0.05,  TGCFT  =  1800.,
 SGCFAT =  .0015,
 SGIRT  =  10.,  TGIRT  =  3600.,
 SGSRT  =  10.,  TGSRT  =  3600.,
 SGTRT  =  5.,  TGTRT  =  3600.,
 SGRPNT =  20.,  SGRFNT =  0.075,
 HORLIM =  2.D0,

 NGPSM  =  5,
 TGPSMF(1) =  1600.,  2300.,  2400.,  3600.,  9999.,
 ONGPSM(1) =  T,    T,    T,    T,    T,
```

```
&

*GPS RECEIVER PARAMETERS:
 &GPSRCV
 DTACQ  =  0.,  DTSEL  =  60.,
 &

*GPS FAILURE PARAMETERS:
 &GPSFPR
 TFGPS  =  1700.,  GPSFTP  =  0,
 IFCLK  =  0,
 FCLKF0  =  .10,  FCLKFR  =  .0010,
 IFSAT  =  0,
 FSATF0  =  .10,  FSATFR  =  .0010,
 &

*GPS SATELLITE ORBIT PARAMETERS:
 &GPSORB
 NPGPS  =  6,
 NSPP  =  3,  3,  3,  3,  3,  3,
 EPHASE  =  0.D0,
 TPLANE  =  0.D0,  0.D0,  0.D0,  0.D0,  0.D0,  0.D0,
 DELNOM  =  0.D0,  0.D0,  0.D0,  0.D0,  0.D0,  0.D0,

 USESAT  =  T,  T,  T,  F,  F,  F,
           T,  T,  T,  F,  F,  F,
           T,  T,  T,  F,  F,  F,
           T,  T,  T,  F,  F,  F,
           T,  T,  T,  F,  F,  F,
           T,  T,  T,  F,  F,  F,

 SAGPS  = 87138451.44D0,
 INCGPS  = 55.0D0,
 LGPSD  =  317.D0,  17.D0,  77.D0,  137.D0,  197.D0,  257.D0,

 ANGPSD  = 280.7D0,  59.7D0, 176.9D0, 314.2D0,  0.0D0,  0.0D0,
        350.7D0,  88.7D0, 219.7D0,  0.0D0,  0.0D0,  0.0D0,
         19.7D0, 158.7D0, 262.4D0, 125.2D0,  0.0D0,  0.0D0,
         57.9D0, 176.4D0, 306.7D0,  0.0D0,  0.0D0,  0.0D0,
        101.9D0, 198.7D0, 337.4D0, 240.7D0,  0.0D0,  0.0D0,
        132.7D0, 262.9D0,  21.4D0,  0.0D0,  0.0D0,  0.0D0,
 &
 NOTE: The Aerospace-supplied nominal GPS satellite orbit radius value
 SAGPS = 87138451.44D0 ft (26559.8 km) leads to an orbital period of
 11.967 hours when utilized with the value of MU in PHYSCN.PRM.

-----------------------------------------------------------------
*** GPSPR:
C  * PARAMETER DEFINITIONS:
C    MXGMS  IN            MAX NO. OF GPS MSMT ON/OFF SEGMENTS
C  * VARIABLE DEFINITIONS:  * NOTE: SUFFIX "T" MEANS TRUE (REALWORLD)
C    DTGPSI  RL           TIME INTERVAL FOR GPS MODEL ACTIVATION
C    DTGPSO  RL           TIME INTERVAL FOR GPS DATAFILE OUTPUTS
```

```
C   HORLIM  RL            GPS SATELLITE MIN ELEVN ABOVE HORIZON
C   ISDGPS  IN           INPUT SEED FOR GPS RANDOM NO. GENERATOR
C   NGPSM   IN            NO. OF GPS MSMT ON/OFF SEGMENTS
C   OGPSE   LG            OUTPUT SWITCH FOR GPS MSMT ERRORS
C   OGPSV   LG            OUTPUT SWITCH FOR GPS MSMT VARIABLES
C   OGSAT   LG            OUTPUT SWITCH FOR GPS SATELLITE DATA
C   ONGPSM  LG  MXGMS       GPS MSMT ON/OFF FLAG FOR SEGMENT #K
C   SGCFAT  RL            SIGMA GPS CLK FREQ ACCEL-SENS COEF, RW
C   SGCFT   RL           SIGMA GPS CLK FREQ DRIFT, RW
C   SGCP0T  RL            SIGMA GPS CLK PHASE DRIFT INITL, RW
C   SGIRT   RL           SIGMA GPS IONOSPHERIC RANGE ERROR
C   SGRFNT  RL            SIGMA GPS RECEIVER FREQ NOISE, RW
C   SGRPNT  RL            SIGMA GPS RECEIVER PHASE NOISE, RW
C   SGSRT   RL           SIGMA GPS SAT EQUIV RANGE ERROR, RW
C   SGTRT   RL           SIGMA GPS TROPO RANGE ERROR, RW
C   TGCFT   RL           CORR TIME GPS CLOCK FREQ DRIFT, RW
C   TGPSMF  RL  MXGMS       FINAL TIME FOR GPS MSMT ON/OFF SEGMT #K
C   TGIRT   RL           CORR TIME GPS IONOSPHERIC RNG ERROR, RW
C   TGSRT   RL           CORR TIME GPS SAT EQUIV RNG ERROR, RW
C   TGTRT   RL           CORR TIME GPS TROPO RANGE ERROR, RW
C   USEPR   LG           USE GPS PR MSMTS FLAG
C   USEPRR  LG            USE GPS PRR MSMTS FLAG

*** GPSRCV:
C  * VARIABLE DEFINITIONS:
C   DTACQ   RL            SIMULATED GPS ACQUISITION DELAY (SEC)
C   DTSEL   RL           DEFAULT INTERVAL BETWEEN SATELLITE SEL

*** GPSFPR:
C  * VARIABLE DEFINITIONS:
C   FCLKF0  RL           CLOCK FREQUENCY FAILURE INITL VALUE
C   FCLKFR  RL            CLOCK FREQUENCY FAILURE GROWTH RATE
C   FSATF0  RL           SAT FREQ/RRATE FAILURE INITL VALUE
C   FSATFR  RL            SAT FREQ/RRATE FAILURE GROWTH RATE
C   GPSFTP  IN           GPS FAILURE TYPE (0 BIAS, 1 RANDOM)
C   IFCLK   IN           INDEX OF FAILED RECVR CLOCK (0 OR 1)
C   IFSAT   IN           INDEX OF FAILED GPS SATELLITE (0 OR K)
C   TFGPS   DP            TIME OF FAILURE IN GPS

*** GPSORB:
C  * PARAMETER DEFINITIONS:
C   MXORB   IN            MAX NO. OF GPS SATELLITE ORBIT PLANES
C   MXSPO   IN           MAX NO. OF GPS SATS PER ORBIT PLANE
C  * VARIABLE DEFINITIONS:
C   ANGPSD  DP  MXSPO,      GPS SATELLITE INITIAL TRUE ANOMALY
C           MXORB
C   DELNOM  DP  MXORB       TRUE ANOMALY OFFSET (NOMINALLY ZERO)
C   EPHASE  DP           ASCENDING NODE OFFSET (NOMINALLY ZERO)
C   INCGPS  DP           GPS SATELLITE ORBIT PLANE INCLINATION
C   LGPSD   DP  MXORB       GPS SATELLITE ORBIT PLANE ASCENDING NODE
C   NPGPS   I4           NUMBER OF GPS SATELLITE ORBIT PLANES
C   NSPP    I4  MXORB       NUMBER OF GPS SATELLITES PER ORBIT PLANE
C   SAGPS   DP           GPS SATELLITE ORBIT SEMI-AXIS MAJOR
```

```
C   TPLANE  DP  MXORB       SATELLITE PLANE REF TIME (NOMINALLY 0)
C   USESAT  LG  MXSPO.      SATELLITE ENABLE COMMAND (NOMINALLY "T")
C           MXORB
```

```
INRAD: RADAR ALTIMETER TRUTH MODEL INPUT PARAMETERS          04/18/94 NAC
 .STL  * STANDARD ERRORS W. ATTITUDE LIMIT *

 &RADPR
 ISDRAD =   4,
 ORADEV =   F,    DTRADO =   2.,
 DTRADI =   0.,

 SGRBR  =  20.,   TCRBR  =  600.,
 SGRSFR =  .001,
 SGRNR  =  50.,

 ANGMAX =  30.,
 HRMAX  = 50000.,
 HRMIN  =  50.,
 &
C---------------------------------------------------------
C  * DEFINITIONS:
C    ANGMAX  RL          MAX RADAR BORESIGHT ANGLE FROM VERT
C    DTRADI  RL          DEFAULT TIME INTERVAL TO RADAR TURN-ON
C    DTRADO  RL          TIME INTERVAL FOR RAD DATAFILE OUTPUTS
C    HRMAX   RL          RADAR MAXIMUM OPERATING HEIGHT
C    HRMIN   RL          RADAR MIN HEIGHT FOR LOW-ALT WARNING
C    ISDRAD  IN          INPUT SEED FOR RADAR RANDOM NO. GENR
C    ORADEV  LG          OUTPUT SWITCH FOR RADAR ERRORS & VARS
C    SGRBR   RL          SIGMA RADAR ALT BIAS ERROR
C    SGRNR   RL          SIGMA RADAR RANDOM MSMT NOISE, RW
C    SGRSFR  RL          SIGMA RADAR ALT SCALE FACTOR ERROR
C    TCRBR   RL          CORR TIME OF RADAR BIAS ERROR
```

```
INSAR: SAR TRUTH-MODEL INPUT PARAMETERS                  06/22/94 NAC
 .SLF  * STD ERRORS EX. ZERO LM ERRORS, FULL COVERAGE *

 SAR TRUTH MODEL CONTROL PARAMETERS:
 &SARPR
 ISDSAR =  4,
 OSARE  =  F,   OSARV  =  F,   DTSARO =  1.,
 DTSARI =  0.,

 NSARLM =    4,
 DTSLM(1) =   100.,    60.,    80.,    120.,
 PDRLM(1) = 100000.,  60000.,  80000.,  120000.,
 PCTLM(1) =  30000., -50000.,  45000., -25000.,
 ALTLM(1) =   1000.,   500.,     0.,    250.,

 NSARM  =  3,
 TSARMF(1) = 3000., 3600., 9999.,
 ONSARM(1) =  T.    T.    T.
 &
```

```
SAR/EO MEASUREMENT PARAMETERS:
&SEOPR
USEAZM =   T,  USEELV =   T,  USERNG =   T,  USERRT =   T,
SSLMR  =  0.0,  DSLMR  = 50000.,
SSABR  =  .001,  SSANR  =  .0005,  TSABR  =  600.,
SSEBR  =  .001,  SSENR  =  .0005,  TSEBR  =  600.,
SSRBR  =  100.,  SSRNR  =  50.,  SSRSFR =  .0004,  TSRBR  =  600.,
SSRRBR =  .1,  SSRRNR =  0.1,  TSRRBR =  600.,
SRNGMX = 200.E3,  SAZMMX =  2.0,  SELVMX =  1.0,  WSMAX  =  1.0,
&

SAR/PVU MEASUREMENT PARAMETERS:
&SPVPR
USEVEL(1) = T,    T,    T,
SSVBR  =  1.,  SSVNR  =  .5,  TSVBR  =  1000.,
SSVMR  =  .001,  SSVSR  =  .001,  TSVSR  =  1000.,
SALTMX = 50.E3,  VSMIN  =  100.,
&

-------------------------------------------------------------
C  * DEFINITIONS:
C    ALTLM  R8  MXSLM      SAREO LANDMARK ALTITUDE
C    DSLMR  RL             CORR DIST SAREO LANDMARK POSN ERRORS
C    DTSARI RL             DEFAULT TIME INTERVAL TO SAR TURN-ON
C    DTSARO RL             TIME INTERVAL FOR SAR DATAFILE OUTPUTS
C    DTSARS R8  MXSARS     SAR MSMT TIME-STEPS PER SEGMENT
C    DTSLM  R8  MXSLM      SAREO LANDMARK ACQUISITION INTERVAL
C    ISDSAR IN             INPUT SEED FOR SAR RANDOM NO. GENR
C    NSARSG IN             NO. OF SAR MSMT SEGMENTS
C    NSARLM IN             NO. OF SAREO TARGET LANDMARKS
C    OSARE  LG             OUTPUT SWITCH FOR SAR MSMT ERRORS
C    OSARV  LG             OUTPUT SWITCH FOR SAR MSMT VARIABLES
C    PCTLM  R8  MXSLM      SAREO LANDMARK CROSS-TRACK POSITION
C    PDRLM  R8  MXSLM      SAREO LANDMARK DOWN-RANGE POSITION
C    SALTMX RL             SARPV MAX ALTITUDE CONSTRAINT
C    SAZMMX RL             SAREO MAX LANDMARK AZIMUTH LOOK ANGLE
C    SELVMX RL             SAREO MAX LANDMARK ELEVATION LOOK ANGLE
C    SRNGMX RL             SAREO MAX LANDMARK RANGE
C    SSABR  RL             SIGMA SAREO AZIMUTH MSMT BIAS, RW
C    SSANR  RL             SIGMA SAREO AZIM MSMT RANDOM NOISE, RW
C    SSEBR  RL             SIGMA SAREO ELEVATION MSMT BIAS, RW
C    SSENR  RL             SIGMA SAREO ELEV MSMT RANDOM NOISE, RW
C    SSLMR  RL             SIGMA SAREO LANDMARK POSITION ERROR, RW
C    SSRBR  RL             SIGMA SAREO RANGE MSMT BIAS, RW
C    SSRNR  RL             SIGMA SAREO RANGE MSMT RANDOM NOISE, RW
C    SSRRBR RL             SIGMA SAREO RANGE-RATE MSMT BIAS, RW
C    SSRRNR RL             SIGMA SAREO RRATE MSMT RANDOM NOISE, RW
C    SSRSFR RL             SIGMA SAREO RANGE SCALE FACTOR ERROR, RW
C    SSVBR  RL             SIGMA SARPV VELOCITY MSMT BIAS, RW
C    SSVMR  RL             SIGMA SARPV MOUNTING MISALIGNMENTS, RW
C    SSVNR  RL             SIGMA SARPV VELOC MSMT RANDOM NOISE, RW
C    SSVSR  RL             SIGMA SARPV FWD VEL SCALE FACTR ERR, RW
```

```
C   TFSARS R8 MXSARS      SAR MSMT SEGMENT FINAL TIME
C   TSABR  RL             CORR TIME SAREO AZIMUTH BIAS, RW
C   TSEBR  RL             CORR TIME SAREO ELEVATION BIAS, RW
C   TSRBR  RL             CORR TIME SAREO RANGE BIAS, RW
C   TSRRBR RL             CORR TIME SAREO RRATE BIAS, RW
C   TSVBR  RL             CORR TIME SARPV VELOCITY MSMT BIAS, RW
C   TSVSR  RL             CORR TIME SARPV FWD VEL SCL FCT ERR, RW
C   USEAZM LG             USE SAREO AZIMUTH MSMT FLAG
C   USEELV LG             USE SAREO ELEVATION MSMT FLAG
C   USERNG LG             USE SAREO RANGE MSMT FLAG
C   USERRT LG             USE SAREO RRATE MSMT FLAG
C   USEVEL LG 3           USE SARPV VELOCITY MSMT FLAG
C   VSMIN  RL             SARPV MINIMUM VELOCITY LIMIT
C   WSMAX  RL             SAREO MAX ANGULAR RATE LIMIT
```

```
INTERR: TERRAIN GENERATOR/MAP PARAMETERS                04/18/94 NAC
 .SS5  * STANDARD ERRORS, STD TERRAIN, 5X5 GRID *

TRUE TERRAIN CONTROLS AND PARAMETERS:
&TERPR
ISDTER = 4,   TERVAR = F,
OTERV  = F,   DTTERO = 1.,

NGRIDI = 5,
DLATI  = 14.54441E-6,
DLONI  = 14.54441E-6,

ELVBAS  = -2000.,
NTERSG  =    3,
TTERSF(1) = 2200.,  2500.,  9999.,
STSAR(1)  =  .05,    .15,    .10,
DTEAR(1)  = 1800.,  1800.,  1800.,
STSCR(1)  =  .05,    .15,    .10,
DTECR(1)  = 1800.,  1800.,  1800.,
&

TERRAIN MAP CONTROLS AND PARAMETERS:
&MAPPR
ISDMAP = 4,
OMAPE  = F,   OMAPV = F,
DTMAPO = 1.,

STMRVR =  45.,  DTMRVR =  52000.,
STMRHR = 150.,  DTMRHR =  52000.,
STMENR =  15.,
&

C-------------------------------------------------------------
C  * DEFINITIONS - TRUE TERRAIN PARAMETERS
C   DLATI  RL             TRN GRID SECTOR LATITUDE SPACING INPUT
C   DLONI  RL             TRN GRID SECTOR LONGITUDE SPACING INPUT
C   DTEAR  RL MXTERS      CORR DIST TERRAIN ELEV ALONG-TRACK, RW
C   DTECR  RL MXTERS      CORR DIST TERRAIN ELEV CROSS-TRACK, RW
```

```
C   DTTERO RL            TIMESTEP FOR TRUE TERRAIN DATA OUTPUT
C   ELVBAS RL            TERRAIN BASE ELEVATION
C   ISDTER IN            INITIAL RANDOM NO. SEED FOR TERRAIN
C   OTERV  LG            OUTPUT SWITCH FOR TERRAIN VARIABLES
C   NGRIDI IN            NO. OF TRN GRID PTS PER AXIS, INPUT
C   NTERSG IN            NO. OF TERRAIN SEGMENTS
C   STSAR  RL MXTERS     SIGMA TERRAIN SLOPE ALONG-TRACK, RW
C   STSCR  RL MXTERS     SIGMA TERRAIN SLOPE CROSS-TRACK, RW
C   TERVAR LG            FLAG, TERRAIN VARIES RUN-TO-RUN
C   TTERSF R8 MXTERS     FINAL TIMES OF TERRAIN SEGMENTS
C
C  * DEFINITIONS - TERRAIN-MAP PARAMETERS
C   DTMAPO RL            TIMESTEP FOR TRN MAP DATAFILE OUTPUT
C   DTMRHR RL            CORR DIST TRN MAP REGL HORZ ERRORS, RW
C   DTMRVR RL            CORR DIST TRN MAP REGL VERT ERRORS, RW
C   ISDMAP IN            INITIAL RANDOM NO. SEED FOR MAP ERRORS
C   OMAPE  LG            OUTPUT SWITCH FOR TRN MAP ERRORS
C   OMAPV  LG            OUTPUT SWITCH FOR TRN MAP VARIABLES
C   STMRHR RL            SIGMA TRN MAP REGL HORZ ERRORS, RW
C   STMRVR RL            SIGMA TRN MAP REGL VERT ERRORS, RW
C   STMENR RL            SIGMA TRN MAP ELEVN NOISE ERRORS, RW
```

```
INTRAJ: TRAJECTORY DATA-FILE MODEL INPUT PARAMETERS        07/12/94 NAC
 ALL  * SPL TRAJ ALL SEGMENTS, M100 - 7200 *


&TRJPAR
OTRAJ  =  F,
DTTRJO =  1.0,
FLTFIL =  '..\..\DKFTRAJ\DATA\FLM1-72.SPL'
&

C-------------------------------------------------------------
C  * DEFINITIONS:
C   DTTRJO RL            TIME INTERVAL FOR TRAJ DATA OUTPUTS
C   FLTFIL CH            PATHNAME OF FLIGHT DATA FILE
C   OTRAJ  LG            OUTPUT SWITCH FOR TRAJECTORY DATA
```

Vita

Major Stephen J. DeLory [redacted] He joined the Canadian Army Reserves (Militia) in 1975, with the Prince Edward Island Regiment (Armoured Reconnaissance). He served with the regiment as Crew Commander, Troop Sergeant and Troop Leader. Major DeLory participated in basic and advanced armoured reconnaissance training, including a four-month series of NATO exercises in West Germany.

Major DeLory transferred to the Canadian Forces in August 1978. After Basic Officer Training School and Basic Air Navigation School, he received his commission and Air Navigator wings in 1979. He served with seagoing helicopter detachments from No. 423 Helicopter Antisubmarine Squadron on a number of Her Majesty's Canadian ships. While at sea, he participated in a number of national, international and NATO exercises, as well as a number of sea rescue operations. He then served as Flight Instructor at No. 406 Helicopter Training Squadron, with duties of flight and acoustics instruction and controlling the Operational Flight and Tactics Trainer.

Major DeLory was voluntarily released from the Canadian Forces in 1984, and studied at Dalhousie University and the Technical University of Nova Scotia to obtain his Bachelor of Engineering (Electrical), majoring in electronics. He subsequently returned to the Canadian Forces as an Air Navigator and served as a Project Officer with the Helicopter Operational Test and Evaluation Facility. He was responsible for operational evaluation of a number of avionics, most notably the Sea King installation of the AN/AAR-47 Missile Warning Set during the Persian Gulf conflict. In addition, he developed specialized airborne data acquisition equipment and provided technical support of all projects at the Facility.

Major DeLory returned to No. 423 Squadron in 1991, where he served as Helicopter Detachment Commander and Squadron Standards Officer. He was selected for the Space Operations program at the Air Force Institute of Technology commencing in 1994, and specialized in Guidance, Navigation and Control. He was promoted to his present rank in June 1995.