②

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A273 370

DTIC
S ELECTE D
DEC 06 1993
E

THESIS

Integration of Differential GPS and Inertial Navigation using a Complementary Kalman Filter

by

Carl William Marquis, III
September, 1993

Thesis Advisor: Isaac I. Kaminer

Approved for public release; distribution is unlimited.

93-29534

93 12 3 008

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School

6b. OFFICE SYMBOL *(if applicable)* 55

7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Integration of Differential GPS and Inertial Navigation using a Complementary Kalman Filter

12. PERSONAL AUTHOR(S)
Marquis, Carl William III

13a. TYPE OF REPORT Master's Thesis

13b. TIME COVERED FROM ______ TO ______.

14. DATE OF REPORT *(Year, Month, Day)* September 1993

15. PAGE COUNT 114

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Kalman filtering, Differential Global Position System, Inertial Navigation System, Modeling

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Precise navigation with high update rates is essential for automatic landing of an unmanned aircraft. Individual sensors currently available — INS, AHRS, GPS, LORAN, *etc.* — cannot meet both requirements. The most accurate navigation sensor available today is the Global Positioning System or GPS. However, GPS updates only come once per second. INS, being an on-board sensor, is available as often as necessary. Unfortunately, it is subject to the Schuler cycle, biases, noise floor, and cross-axis sensitivity. In order to design and verify a precise, high update rate navigation system, a working model of Differential GPS has been developed including all of the major GPS error sources — clock differences, atmospherics, Selective Availability and receiver noise. A standard INS system was also modeled, complete with the inaccuracies mentioned. The outputs of these two sensors — inertial acceleration and pseudoranges — can be optimally blended with a complementary Kalman filter for positioning. Eventually, in the discrete case, the high update rate and high precision required for autoland can be achieved.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL Isaac I. Kaminer

22b. TELEPHONE *(Include Area Code)* (408) 656-2972

22c. OFFICE SYMBOL AA/KA

Approved for public release; distribution is unlimited

Integration of Differential GPS and Inertial Navigation using a Complementary Kalman Filter

by

Carl W. Marquis, III
Lieutenant, United States Naval Reserve
B.S., Rensselaer Polytechnic Institute, 1985

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

September, 1993

Author: Carl W. Marquis III
Carl W. Marquis, III

Approved by: I. Kaminer
Isaac I. Kaminer, Thesis Advisor

M.K. Shields
LCDR Michael K. Shields, Second Reader

Daniel J. Collins
Daniel J. Collins, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

Precise navigation with high update rates is essential for automatic landing of an unmanned aircraft. Individual sensors currently available — INS, AHRS, GPS, LORAN, *etc.* — cannot meet both requirements. The most accurate navigation sensor available today is the Global Positioning System or GPS. However, GPS updates only come once per second. INS, being an on-board sensor, is available as often as necessary. Unfortunately, it is subject to the Schuler cycle, biases, noise floor, and cross-axis sensitivity. In order to design and verify a precise, high update rate navigation system, a working model of Differential GPS has been developed including all of the major GPS error sources — clock differences, atmospherics, Selective Availability and receiver noise. A standard INS system was also modeled, complete with the inaccuracies mentioned. The outputs of these two sensors — inertial acceleration and pseudoranges — can be optimally blended with a complementary Kalman filter for positioning. Eventually, in the discrete case, the high update rate and high precision required for autoland can be achieved.

Accesion For
NTIS CRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification ....
By ....
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

DTIC QUALITY INSPECTED 3

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 5

  A. COORDINATE SYSTEMS 5

    1. Earth-Centered, Earth-Fixed Cartesian Coordinate System . 6

    2. Geodetic Coordinate System 7

    3. Tangent Plane (Local Geodetic) Cartesian Coordinate System 8

  B. COORDINATE TRANSFORMATIONS 8

    1. Geodetic to ECEF Coordinate Transformation 9

    2. ECEF to Geodetic Coordinate Transformation 10

    3. ECEF to Tangent Plane (Local Geodetic) Coordinate Transformation 11

    4. Tangent Plane (Local Geodetic) to ECEF Coordinate Transformation 12

  C. THE NAVSTAR GPS 12

    1. Space Segment 13

    2. User Segment 17

    3. Control Segment 18

    4. Differential GPS 18

  D. GPS ERROR SOURCES 19

    1. Atmospheric Delays 19

      a. *Ionospheric Delays* 19

      b. *Tropospheric Delays* 22

    2. Selective Availability 24

3. Clock Differences 24
4. Ephemeris Error 28
5. Multipath 29
6. Receiver Noise 30
7. Dilution of Precision 31
E. INERTIAL NAVIGATION 32
1. Gimbaled IMU 33
2. Strapdown IMU 34
F. INS COMPUTATIONS 35
G. INS ERROR SOURCES 36
1. Biases 37
2. Cross-Axis Sensitivity 37
3. Noise Floor 37
III. SIMULINK MODEL 39
A. DGPS SIMULINK MODEL 39
1. Ephemeris Model 41
2. Pseudorange Computation 42
3. Clock Model 45
4. Selective Availability Model 45
5. Pseudorange Correction 46
B. DGPS MODEL VERIFICATION 46
1. Verification with Clock Error 48
a. *Atmospheric Correction* 48
b. *Differential Corrections* 49
c. *Differentially Corrected Pseudoranges* 49
2. Verification without Clock Error 50

a. *Atmospheric Correction* 51

b. *Differential Corrections* 51

c. *Differentially Corrected Pseudoranges* 52

C. INS SIMULINK MODEL 54

1. Strapdown IMU 54

2. INS Computations 55

D. INS MODEL VERIFICATION 56

1. Euler Angles 58

2. Linear Acceleration 60

E. NAVIGATION (LOOKING AHEAD) 62

IV. DGPS/INS INTEGRATION 64

A. GENERAL LINEAR KALMAN FILTER DESIGN 64

B. DESIGN OF THE COMPLEMENTARY, LINEAR, KALMAN FILTER FOR DGPS/INS INTEGRATION 67

C. FILTER VERIFICATION 75

1. Verification without Accelerometers 77

2. Verification with Accelerometers 77

V. CONCLUSIONS 82

A. DISCRETIZE 82

B. EXTENDED KALMAN FILTER 83

C. ACCOUNT FOR CHANGING SATELLITE GEOMETRIES . . . 83

D. CARRIER PHASE GPS 84

APPENDIX A: LINEARIZED SYNTHESIS MODEL 85

APPENDIX B: MATLAB FILES 87

APPENDIX C: USER'S MANUAL 96

REFERENCES 98

INITIAL DISTRIBUTION LIST 100

LIST OF TABLES

2.1 ALLAN VARIANCE PARAMETERS FOR THREE COMMON TIMING STANDARDS 27
3.1 MEAN AND STANDARD DEVIATION OF DIFFERENTIALLY CORRECTED PSEUDORANGE ERRORS (WITHOUT CLOCK ERROR) 53
4.1 SENSOR BANDWIDTHS FOR DGPS/INS KALMAN FILTER . . . 73
4.2 MEAN AND STANDARD DEVIATION OF X, Y, AND Z ESTIMATED POSITION ERRORS WITH AND WITHOUT ACCELEROMETERS 78

LIST OF FIGURES

2.1 Earth–centered, Earth–fixed Coordinate System 6

2.2 Geodetic Coordinate System 7

2.3 Tangent Plane Coordinate System 8

2.4 A NAVSTAR GPS Satellite 13

2.5 Diurnal Ionospheric Delay 21

2.6 Receiver Clock Model 25

2.7 Ideal Allan variance 26

2.8 Real Allan variance 26

2.9 Ephemeris error 29

2.10 Dilution of Precision 32

2.11 Gimbaled IMU 34

3.1 DGPS Model 40

3.2 Ephemeris Model 42

3.3 Pseudorange Computation Model 43

3.4 Selective Availability Model 46

3.5 Pseudorange Correction 47

3.6 Atmospheric correction for pseudorange #1 (with clock error) 48

3.7 Differential Corrections for pseudorange #1 (with clock error) 49

3.8 Corrected and differentially adjusted pseudorange errors to satellite #1 (with clock error) 50

3.9 Atmospheric correction for pseudorange #1 (without clock error) . . 51

3.10 Differential Corrections for pseudorange #1 (without clock error) . . 52

3.11 Corrected and differentially adjusted pseudorange errors to satellite #1 (without clock error) 53
3.12 INS Model 54
3.13 Strapdown IMU Model 55
3.14 INS Computations Model 56
3.15 Euler angles estimator 57
3.16 Actual and estimated roll angles 58
3.17 Actual and estimated pitch angles 59
3.18 Actual and calculated X acceleration 60
3.19 Actual and calculated Y acceleration 61
3.20 Actual and calculated Z acceleration 62
3.21 Z acceleration error 63
4.1 DGPS/INS Integration Synthesis Model 69
4.2 Frequency response from ρ_1 to $\hat{\rho}_1$ and ρ_2 to $\hat{\rho}_2$ 72
4.3 Frequency response from ρ_3 to $\hat{\rho}_3$ and ρ_4 to $\hat{\rho}_4$ 72
4.4 Frequency response from ρ_5 to $\hat{\rho}_5$ and ρ_6 to $\hat{\rho}_6$ 73
4.5 Frequency response from ${}^{U}a_x$ to $\hat{\rho}_3$ 74
4.6 Frequency response from ${}^{U}a_z$ to $\hat{\rho}_1$ 74
4.7 Frequency response from ${}^{U}a_z$ to $\hat{\rho}_5$ 75
4.8 Integrated Navigation System 76
4.9 X displacement error without accelerometers 78
4.10 Y displacement error without accelerometers 79
4.11 Z displacement error without accelerometers 79
4.12 X displacement error with accelerometers 80
4.13 Y displacement error with accelerometers 80
4.14 Z displacement error with accelerometers 81

ACKNOWLEDGMENT

Special thanks to Professor Kaminer for his patient tutelage without which I could not have completed this work. Thanks also to Dave and Shiva for their frequent and able assistance with LaTeX and Unix.

I. INTRODUCTION

Precise navigation has been one of mankind's endeavors for thousands of years. Even prehistoric man had to answer the question, "Where am I?" frequently. Celestial navigation, landmark navigation, and other forms of "natural" navigation suffice for some purposes even today. But the human race's non–stop technological progress has both enabled and created a need for more precise navigation. The advent of aviation and space travel have placed greatly increased demands on navigation. Recently developed "smart" weapons, as well, require exact knowledge of their position in order to find their targets. Furthermore, unmanned recoverable aircraft could benefit greatly from improving navigation accuracy.

Today, work in remotely piloted vehicles is at the forefront of technology. The goal of the current research and development on the "Archytas" (taking place here at The Naval Postgraduate School) is totally autonomous flight in all regimes -- automatic take–off, transition from vertical take–off to horizontal flight, automatic waypoint tracking, and finally automatic landing. The task of automatically landing an aircraft, on land or on a pitching ship at sea, requires nearly flawless guidance, navigation, and control. Positioning errors which would be considered minuscule under normal circumstances could easily be disastrous during an automatic landing. "Autoland" requires not only exceptional positioning accuracy in three dimensions, but also rapid updates. Thus, the goal of this undertaking is to provide the most accurate, most rapidly updated, three–dimensional navigation system available.

Many existing navigation systems provide either a high update rate *or* accuracy. Probably the most popular long–range navigation system in use is the Inertial Navigation System (INS). This system is entirely self–contained on the platform which carries it. It senses aircraft acceleration which it converts to velocity and position. Inertial navigation is subject to numerous errors, among them the notorious Schuler cycle, which make the position accuracy degrade over time. This drift is typically around one nautical mile per hour. While the INS position can be updated many times each second, its accuracy is insufficient as a stand-alone navigation sensor for autoland. There are several radio navigation aids currently available — GPS, LORAN, OMEGA, TACAN, VOR/DME. They offer various degrees of precision due to their technical sophistication and overall age.

LORAN, short for Long Range Navigation, is a low frequency system which allows receivers to estimate their position by time-difference-of-arrival. First, a receiver calculates its position along a hyperbolic, earth-bound curve by using a pair of LORAN transmitters. A second pair of transmitters allows the receiver to generate another hyperbola representing the locus of possible positions. It is now possible to resolve its two-dimensional position as the intersection of the two hyperbolas. LORAN wavelengths are approximately 1.6 miles, making truly precise positioning impossible. Also, fixes are available only ten to 20 times per minute. This system is unsuitable for autoland because of its imprecision, low update rate and the fact its fixes are only two-dimensional.[Ref. 1]

OMEGA navigation is based on the phase-difference-of-arrival technique. Rather than using time of arrival, OMEGA uses phase differences at the receiver to generate two hyperbolas, similar to LORAN. However, this system uses a far lower frequency giving it excellent range. In fact, OMEGA covers the entire earth with only eight transmitters. This low frequency yields wavelengths of approximately 16 miles. Position accuracy can be expected to be two to four nautical miles. LORAN and OMEGA position accuracy can be greatly improved by using the differential mode. This mode allows a receiver's position to be calculated relative to the known position of another receiver. Sub-mile accuracy can be achieved with this augmented mode. Even in the differential mode, the OMEGA system is unacceptable for autoland as it cannot provide the accuracy or the three dimensional fixing required.[Ref. 1]

VOR/DME (VHF Omnidirectional Range) and TACAN (Tactical Air Navigation) systems both provide bearing and range information to aircraft within their line-of-sight. The bearing portion of these systems is composed of two signals — a rotating "lighthouse" signal and an omnidirectional signal. The "lighthouse" portion of the signal rotates continuously at 30 revolutions per minute. Each time it reaches magnetic north, the omnidirectional signal transmits one pulse. A receiver's magnetic bearing to the transmitter is a linear function of the difference in arrival time between the omnidirectional pulse and the "lighthouse pulse". This method is accurate to within approximately three degrees [Ref. 2, p. 36]. The ranging portion of these systems uses two way spherical ranging. Equipment on board the aircraft broadcasts a pulse which the station immediately rebroadcasts on another frequency. Once again, the difference in time between transmission and reception of this pulse multiplied by half of the speed of light yields slant range to the station. Overall range accuracy of DME can be expected to be ±0.5 nautical miles or three parts in 100, whichever is greater [Ref. 1]. TACAN range accuracy is approximately ±0.1 nautical miles. VOR uses VHF frequencies while DME and TACAN use UHF. Both systems are limited by line-of-sight. This limitation is troublesome for an aircraft needing precise data at low altitude. While position updates from these systems are available virtually continuously, their accuracies are insufficient for the autoland problem.

The remaining sensor is the Global Positioning System (GPS). This is the newest and most accurate radio navigation aid available. GPS provides positioning accuracy on the order of 100 meters to everyone via the Standard Positioning Service (SPS), and 16 meters in three dimensions via the Precise Positioning Service (PPS) to those authorized by the United States Department of Defense. [Ref. 3, p. 3–1]. GPS accuracy can be further improved to just a few meters by using differential corrections (see Chapter II.). While its title as most precise navigation aid available is unchallenged, its update rate is currently once per second. Thus, it too is insufficient as a stand–alone navigation sensor in the autoland phase of flight.

It seems that no single sensor can provide highly precise, rapidly updated positioning information. The high update rate of INS coupled with the superb precision of differential GPS can solve the autoland navigation problem. By using the techniques of optimal estimation, these two sensors can be blended to produce optimal estimates of position and velocity. The estimates will be both precise and rapidly updated, thus meeting the established criteria.

This thesis is composed of 4 basic parts. The first portion discusses background material related to DGPS/INS navigation including basic coordinate systems. The following section describes, in detail, the SIMULINK computer models of DGPS and INS. The heart of this work, the Kalman filter which integrates the outputs of the two sensors, is described and verified in the third section. The final section offers suggestions for further refinements of the concepts advanced in this thesis.

II. BACKGROUND

Before embarking on a study of the intricacies of integrated Global Positioning System (GPS) and Inertial Navigation System (INS) navigation, it is necessary to be familiar with some underlying concepts. For example, a GPS/INS navigation system uses a minimum of three coordinate systems. In this chapter, these reference frames are explicitly defined, as well as the transformations between them. Secondly, an overview of GPS is presented to highlight the principles which are used and the errors which effect the resulting system. The chapter concludes with an overview of inertial navigation.

A. COORDINATE SYSTEMS

GPS-aided inertial navigation involves the use of three distinct coordinate systems. They are:

- earth-centered, earth-fixed Cartesian
- latitude, longitude, altitude (geodetic)
- tangent-plane Cartesian (local geodetic)

The earth-centered, earth-fixed system is independent of the mathematical model of the earth's surface. However, both the geodetic and the local geodetic systems depend on the specification of the earth model. The current standard for modeling the surface of the earth is the WGS-84 ellipsoid. This ellipsoid is generated by rotating an ellipse, whose semi-major axis is 6378137.0 meters and whose semi-minor axis is 6356752.3 meters, about its minor axis. The resulting closed surface is the model of the earth's surface. The true north pole (conventional terrestrial pole) and true south pole are the endpoints of the minor axis of the ellipsoid.

The remainder of this section will define the three coordinate systems in detail.

1. Earth–Centered, Earth–Fixed Cartesian Coordinate System

GPS uses the earth–centered, earth–fixed Cartesian system for fixing satellite positions. Likewise, GPS receivers calculate their navigation solutions (*i.e.*, position) in the ECEF system before transforming to latitude, longitude, and altitude (geodetic). The origin of this system, as the name implies, is at the center of the earth. The X–axis goes through the intersection of the equator (0° of latitude) and the prime meridian (0° longitude). The Y–axis departs the origin and passes through the intersection of the equator and 90° E longitude. To complete the right–handed triad, the Z–axis leaves the origin and passes through the true north pole. Many surveying oriented texts refer to this system as the Conventional Terrestrial System (CTS). It is depicted in Figure 2.1.

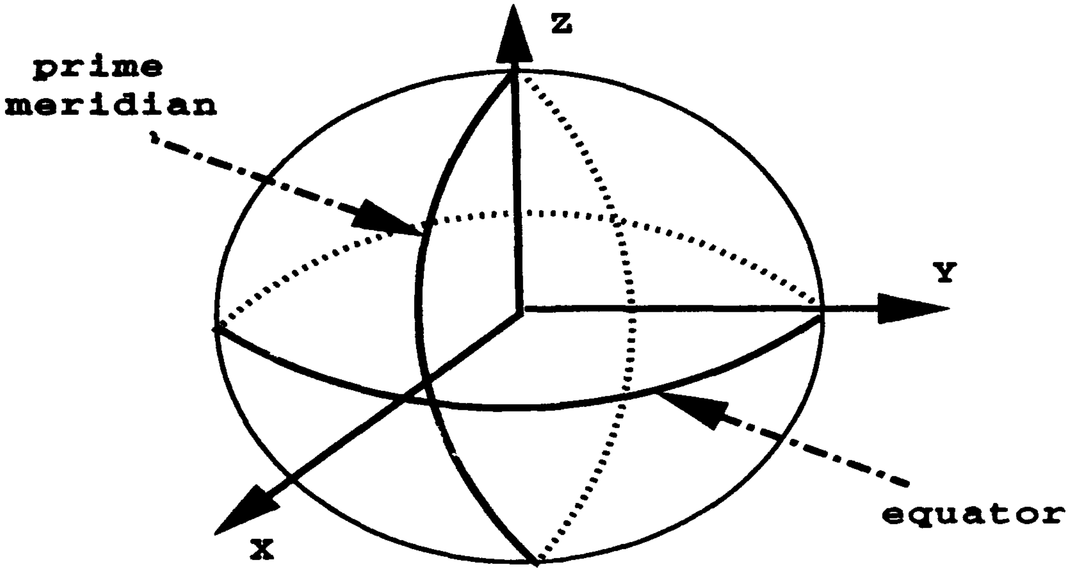

Figure 2.1: Earth–centered, Earth–fixed Coordinate System

2. Geodetic Coordinate System

The output of navigation systems used on aircraft today is generally latitude, longitude, and altitude — *i.e.* resolved in the geodetic coordinate system. This is the system used for describing positions of most earth bound objects. Charts developed for long range land and sea navigation invariably use geodetic coordinates.

The geodetic coordinate system is somewhat analogous to spherical coordinates. The primary difference is that the elevation angle or latitude, ϕ, is the angle between the ***ellipsoidal normal*** and the equatorial plane. This means that the ray that defines this angle does not intersect the equatorial plane at the exact center of the earth. Instead, it intersects the equatorial plane at a small radius outside of the center as shown in Figure 2.2. The longitude, λ, is identical to the spherical concept of that angle. It is the angle in the equatorial plane from 0° latitude and 0° longitude to any given point. Finally, h, the geodetic height or altitude, is the distance along the ellipsoidal normal away from the surface of the earth.

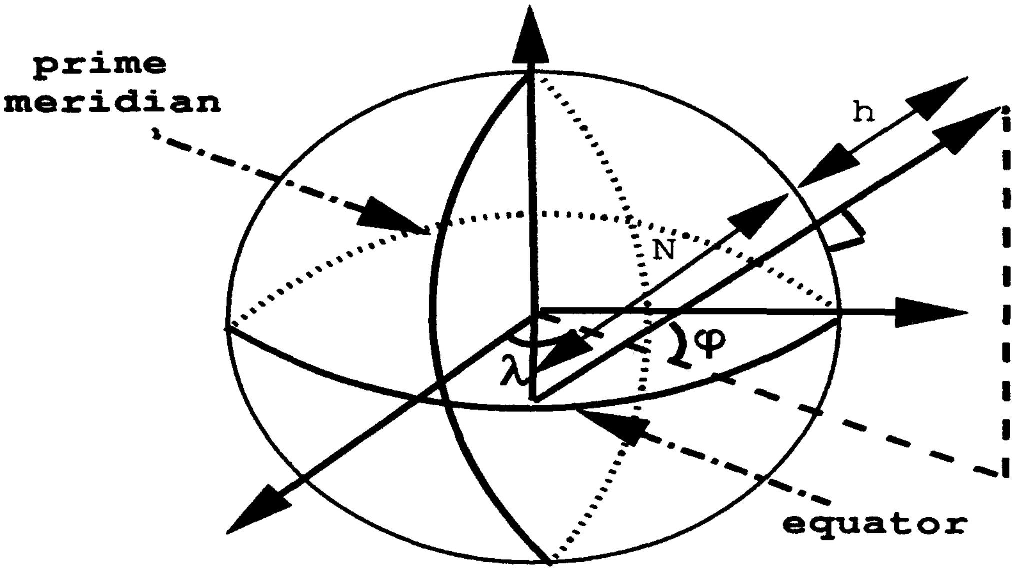

Figure 2.2: Geodetic Coordinate System

3. Tangent Plane (Local Geodetic) Cartesian Coordinate System

Typically, pure inertial systems navigate in a so-called tangent plane coordinate system, before outputting position in geodetic coordinates. The tangent plane system is defined by passing a plane through any point on the earth's surface (see Figure 2.3). The intersection of the plane with the surface of the earth becomes the origin of the system. The X-axis points toward true east. The Y-axis points toward true north. Lastly, the Z-axis is perpendicular to the defining plane of the system, away from the center of the earth. It is the Z coordinate of the triad which defines a point's altitude in this system.

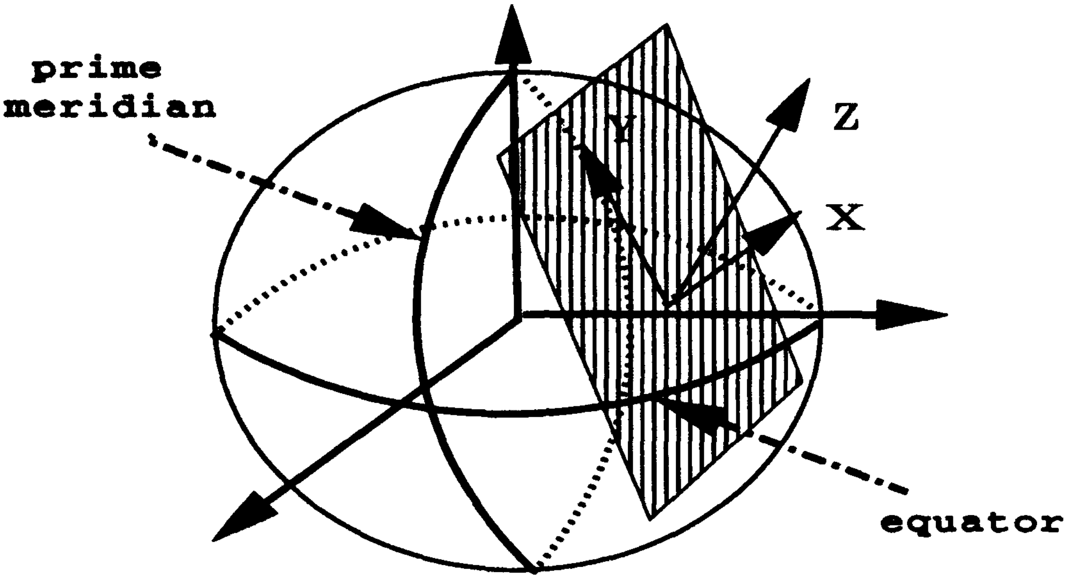

Figure 2.3: Tangent Plane Coordinate System

B. COORDINATE TRANSFORMATIONS

In order to use these three coordinate systems, one must be able to transform between them freely. For example, satellite coordinates enter the DGPS model (detailed in a later chapter) in ECEF coordinates. However, the ranges to the satellites are calculated in the tangent plane coordinate system. Therefore, a transformation between ECEF and tangent plane systems is required. For other similar circumstances, transformations are necessary between all systems. With a total of three systems, six conversions are required — one from each system to the two others, multiplied by three systems, yields six transformations. However, two of these algorithms can be assembled as combinations of the others. By being able to convert from geodetic to ECEF, and ECEF to local geodetic, one can convert from geodetic to local geodetic by chaining the two conversions together. Thus, the only four transformations discussed are

- geodetic to ECEF
- ECEF to geodetic
- ECEF to tangent plane
- tangent plane to ECEF

Since the vectors to be transformed are all position (not free) vectors, the transformations are defined for non–free vectors.

1. Geodetic to ECEF Coordinate Transformation

In order to compute the transformation from geodetic to ECEF coordinates, three auxiliary quantities — f, e, and N — must first be defined. The flattening factor, f, represents the relative flatness of the ellipsoid. A zero flattening factor would mean an unflattened ellipse (a sphere), while a unity value would mean a totally flattened ellipsoid (a circle in the plane perpendicular to the minor axis). The mathematical definition of f is

$$f = \frac{a - b}{a}, \tag{2.1}$$

where a and b are the semi–major and semi–minor axes of the ellipsoid, respectively.

Directly related to the flattening factor is the eccentricity, e. It is defined by

$$e^2 = 2f - f^2. \tag{2.2}$$

The eccentricity is a variable similar to the flattening factor. It represents how close the ellipsoid is to a sphere. It, too, is one for a sphere and zero for a completely flat figure. The eccentricity, rather than the flattening factor, is typically used in coordinate transformations.

Lastly, N is the length of the ellipsoidal normal from the ellipsoidal surface to its intersection with the ECEF Z - axis. Mathematically, N is

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}, \tag{2.3}$$

where ϕ is the geodetic latitude.

Using these quantities, one may define the transformation

$$\begin{aligned} x &= (N + h)\cos\phi \, \cos\lambda \\ y &= (N + h)\cos\phi \, \sin\lambda \\ z &= [N(1 - e^2) + h]\sin\phi. \end{aligned} \tag{2.4}$$

2. ECEF to Geodetic Coordinate Transformation

The transformation from ECEF to geodetic coordinates is clearly the inverse of the process presented in the previous section. First, λ, the longitude, can be found by dividing first two expressions of Equations 2.4 yielding

$$\tan\lambda = \frac{y}{x}. \tag{2.5}$$

By examining the geometry in Figure 2.2, one can determine the following relationship

$$\tan \phi = \frac{(N + h)\sin\phi}{\sqrt{x^2 + y^2}}, \tag{2.6}$$

which is a non-linear equation in ϕ. Solving the third of Equations 2.4 for $(N+h)\sin\phi$ and substituting into Equation 2.6 yields

$$\tan\phi = \frac{z}{\sqrt{x^2+y^2}}\left(1+\frac{e^2 N\sin\phi}{z}\right), \tag{2.7}$$

which is still an analytically unsolvable equation in geodetic latitude. To solve this equation, one initially assumes that h is zero, an excellent assumption for intra-atmospheric flight. Now, the Z equation of Equations 2.4 can be simplified to

$$Z_{h=0} = N(1-e^2)\sin\phi. \tag{2.8}$$

This equation can be substituted into Equation 2.7 to give the initial solution for ϕ

$$\tan\phi_1 = \frac{1}{1-e^2}\frac{w}{\sqrt{x^2+y^2}}. \tag{2.9}$$

This initial solution for ϕ can be substituted back into the second term of Equation 2.7 to yield an updated ϕ. Iteration of this process commencing with Equation 2.7 continue until the geodetic latitude stops changing. Finally, solving Equation 2.6 for h

$$h = \frac{\sqrt{x^2+y^2}}{\cos\phi} - N, \tag{2.10}$$

which completes the conversion.

3. ECEF to Tangent Plane (Local Geodetic) Coordinate Transformation

Anytime one uses a local geodetic or tangent plane coordinate system, one must first specify the geodetic coordinates — latitude and longitude - of the origin. Once specified, the origin must be expressed in ECEF coordinates. Then, a vector from this origin to the point being transformed can be formed, resolved in the ECEF system

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix}_{ECEF} = \begin{bmatrix}x_2-x_1\\ y_2-y_1\\ z_2-z_1\end{bmatrix}, \tag{2.11}$$

where the '2' subscript denotes the point being transformed. The product of two rotation matrices operates on the difference vector defined in Equation 2.11 to yield the local geodetic coordinates of P_2

$$P_2 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{tp} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}_{ECEF}, \qquad (2.12)$$

where λ is the geodetic longitude and ϕ is the geodetic latitude.

4. Tangent Plane (Local Geodetic) to ECEF Coordinate Transformation

Unlike converting between geodetic and ECEF, transforming between local geodetic and ECEF is invertible. The transformation from tangent plane to ECEF can be derived by merely reversing the process developed in the previous section.

First, the origin of the local geodetic system must be converted from geodetic to ECEF coordinates. Next, the inverse of the rotations performed in Equation 2.12 must be executed yielding the Δ vector from the origin of the tangent plane system to the point being transformed. This vector, expressed in ECEF coordinates is

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}_{ECEF} = \begin{bmatrix} -\sin\lambda & -\sin\phi\cos\lambda & \cos\phi\cos\lambda \\ \cos\lambda & -\sin\phi\sin\lambda & \cos\phi\sin\lambda \\ 0 & \cos\phi & \sin\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{tp}. \qquad (2.13)$$

Clearly, adding the Δ vector to the position of the origin of the tangent plane system (both now in ECEF coordinates) completes the transformation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} = \begin{bmatrix} x_{orig} \\ y_{orig} \\ z_{orig} \end{bmatrix}_{ECEF} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}_{ECEF}. \qquad (2.14)$$

5. THE NAVSTAR GPS

The NAVigation Satellite Timing And Ranging Global Positioning System is a satellite–based radio navigation system with the capability to provide locating data to an unlimited number of users. The first satellite was deployed in 1978 although the first receiver did not become commercially available until 1982 [Ref. 4, section 1.2]. This system is the product of experience gained from several previous space–based navigation systems like TRANSIT AND USAF System 621B. It is comprised of three segments:

- Space segment
- User segment
- Control segment

Each segment is discussed in the following sections.

1. Space Segment

A total of 24 satellites now constitute a fully operational space segment. Twenty–one of theses space vehicles (shown in Figure 2.4 operate continuously, while the remaining three act as orbiting spares. Today, only three Block I satellites remain in orbit. These satellites were the first GPS satellites in space. They were launched

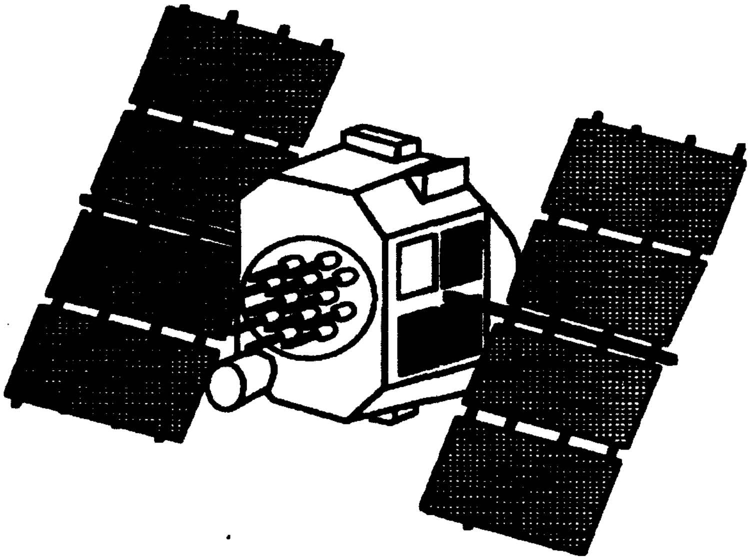

Figure 2.4: A NAVSTAR GPS Satellite from [Ref. 5, p. 4.01]

from 1978 through 1985 [Ref. 4, section 1.2]. Block I space vehicles weigh 960 pounds and generate 420 watts of electrical power [Ref. 2, p. 27]. The remaining satellites in orbit were creatively named the Block II vehicles. These vehicles are less vulnerable to radiation and have more memory. With this increase in capability has come an increase in weight and power requirements. These newer satellites weigh 2000 pounds and generate 700 watts of electrical power.

The space vehicles are inserted into orbits defined by the six Keplerian constants:

- semi-major axis, a
- orbital eccentricity, e
- orbital inclination, i
- ascending node, Ω
- argument of perigee, ω
- time of perigee passage, T

The semi-major axis of a GPS satellite orbit is nominally 26,560 kilometers. It is half of the length of the ellipse which defines the space vehicle's path. Second. the eccentricity (different from the eccentricity defined with respect to the shape of the WGS-84 ellipsoid) or oblateness of the satellite's orbit is defined as follows:

$$e = \frac{r_a - r_p}{r_a + r_p}, \tag{2.15}$$

where r_a is the apogee radius and r_p is the perigee radius. For GPS satellites, the eccentricity cannot exceed two percent. The third of the Keplerian elements is orbital inclination. It is the angle between the plane defined by the orbit and the equator.

For example, satellites in a polar orbit have a 90° orbital inclination; those in an equatorial orbit have 0° orbital inclination. Block I NAVSTAR GPS vehicle orbits are inclined at 63° while Block II satellites are inclined at 55°. The ascending node is the satellite's geodetic longitude as it passes through 0° of latitude toward the northern hemisphere. GPS satellites orbit in six different planes. Thus, there are exactly six different ascending nodes. The last two Keplerian elements are the argument of the perigee and the time of perigee passage. The argument of the perigee is the angle in the orbital plane between the ascending node and the closest point of approach of the satellite to earth. The time when the vehicle reaches this point is the time of perigee passage. The ranges of these last two parameters span all possible values for GPS satellites. That is, satellites reach their perigee at all different times of day and different locations.

Ideally, the six Keplerian elements would be sufficient to define any satellite's three-dimensional position and velocity vectors for all time. However, the orbits become perturbed by lunar and solar gravity and the earth's equatorial bulge, as well as several other less significant effects. Thus, the number of quantities required to fully specify the position and velocity of a satellite in a real orbit is increased to 16. The extra elements consist of time rates of change of the ascending node and inclination, *i. e.* $\dot{\Omega}$ and $\dot{i}$, as well as six other coefficients which account for variation in the earth's gravitational field. These 16 coefficients must become part of the navigation message.

The navigation message is the framework in which the GPS satellites broadcast their data. One "frame", the complete message, consists of five subframes of 300 bits each. Subframe one contains coefficients used to correct the satellite's clock to exact GPS time. The 16 pieces of ephemeris data are broadcast in the second and third subframes. Subframe four contains special messages, ionospheric correction coefficients, and coefficients for conversion of GPS time to Universal Coordinated Time. Ephemeris and health data for the entire GPS constellation is transmitted in subframe five. Because of the volume of data in subframes four and five, both must be subdivided into 25 pages. Therefore, it takes 25 full frames, broadcast at the rate of 30 seconds per frame, or 750 seconds (12.5 minutes) to receive the entire navigation message. Critical navigation data - ephemerides, and clock correction coefficients — are updated every frame. Secondary data transmitted in subframes four and five is provided primarily to assist the receiver in acquiring other satellites. These data are not intended to be precise so the lower update rate of once every 12.5 minutes is satisfactory.

Satellites use a pseudorandom binary code superimposed on the two carrier frequencies to communicate. The two frequencies are L1 (1575.42 MHz) and L2 (1227.6 MHz). The L1 frequency carries both the C/A- (coarse acquisition) and P-(precise) codes, while L2 carries only the P-code. The less precise C/A-code is broadcast at a rate of 1.023 million bits per second (Mbps) and is 1023 bits long. Therefore, this code repeats itself every millisecond. The C/A-code is unique to each satellite and does not change, allowing GPS receivers to quickly distinguish between space vehicles, even without access to the P-code. The P-code, being more precise, is transmitted at 10.23 Mbps with a code length of approximately six trillion bits. This code takes 37 weeks to repeat. Since the codes are reset every week at midnight between Saturday and Sunday, there are sufficient "code weeks" available in the P-code such that one can be assigned to each space vehicle each week. Therefore, GPS receivers can easily distinguish satellites from each other with their individual, weekly-assigned P-codes. The remaining code weeks are available for uplink from the control segment to the satellites.

2. User Segment

The many thousands of GPS receivers constitute the user segment. The receiver's functions are to receive and interpret the navigation message and to calculate and output position. GPS receivers must determine the time the navigation message takes to travel the distance from the satellite to the receiver. This is achieved by autocorrelating the pseudorandom binary pulse train received from the satellite with the one in memory. A typical civilian GPS receiver must have the C/A-codes for all 24 satellites in its memory (requiring only three kilobytes). As the pseudorandom code is received, the receiver slews its code until the result of the autocorrelation function jumps to one. The receiver is now "locked-on" to that code. Multiplying the length of time that the receiver must slew its code to achieve a unity correlation by the speed of light yields the "pseudorange". This is not the actual range because the offset of the receiver clock is uncertain. When the receiver can lock-on to four satellites and thus measure pseudoranges to each, is three-dimensional position and clock error can be solved for by inverting this set of four equations

$$\begin{aligned}
\rho_1 &= \sqrt{(x_{sat_1} - x_{rcvr})^2 + (y_{sat_1} - y_{rcvr})^2 + (z_{sat_1} - z_{rcvr})^2} + c\Delta t \\
\rho_2 &= \sqrt{(x_{sat_2} - x_{rcvr})^2 + (y_{sat_2} - y_{rcvr})^2 + (z_{sat_2} - z_{rcvr})^2} + c\Delta t \\
\rho_3 &= \sqrt{(x_{sat_3} - x_{rcvr})^2 + (y_{sat_3} - y_{rcvr})^2 + (z_{sat_3} - z_{rcvr})^2} + c\Delta t \\
\rho_4 &= \sqrt{(x_{sat_4} - x_{rcvr})^2 + (y_{sat_4} - y_{rcvr})^2 + (z_{sat_4} - z_{rcvr})^2} + c\Delta t, \qquad (2.16)
\end{aligned}$$

where ρ_i are pseudoranges, $[x_{sat_i}, y_{sat_i}, z_{sat_i}]$ are the ECEF coordinates of a satellite, $[x_{rcvr}, y_{rcvr}, z_{rcvr}]$ are the ECEF coordinates of the GPS receiver, c is the speed of light, and Δt is the receiver's clock error. Having solved this set of equations, the receiver now need only transform the solution to the geodetic system and display the results.

3. Control Segment

The GPS Control segment is responsible for generating and uplinking clock correction coefficients and ephemeris corrections for all satellites in the constellation. Five control stations — Hawaii, Ascension Island, Diego Garcia, Kwajalein, and Colorado Springs, Colorado - comprise this segment. These control stations are essentially GPS receivers capable of constantly tracking all satellites in view. Additional capabilities include highly accurate Cesium clocks and recording facilities. The first four of these stations track all satellites in view and record pseudorange information continuously. This data is sent to the Master Control Station at Colorado Springs where it is processed. When all four monitor stations have locked-on to a single space vehicle simultaneously, the exact inverse of the standard navigation problem mentioned in the previous section exists. Since the exact locations of the monitor stations are known and their clocks are extremely accurate, the four unknowns become the three coordinates of the space vehicle position and its clock offset. The Master Control Station calculates these quantities and from them, derives the necessary ephemeris and clock corrections. This information is uplinked to each space vehicle at least daily.

4. Differential GPS

Although GPS alone provides highly accurate positioning, it can be made still more accurate by augmenting it with a differential station. A differential station is merely another GPS receiver whose exact location is known. When this second receiver is near the first receiver, both are subject to nearly the same errors, *i. e.* the same local atmospheric properties, nearly identical elevation angles and propagation paths to any given GPS satellite, the same clock errors and the same ephemeris errors for each satellite. By employing the Pythagorean theorem on its position and the satellite position broadcast in the navigation message, the differential station can calculate the exact range to that satellite. Meanwhile, it can also calculate pseudorange in the standard way (see Section 2.). By comparing these two values for each satellite in view, the differential station can evaluate the pseudorange error to each satellite. These values can be broadcast periodically to be used by receivers in the local area to improve accuracy. By using the differential corrections, GPS accuracy, even for single frequency C/A-code users, can be improved to one to seven meters rms [Ref. 2, p. 64].

D. GPS ERROR SOURCES

Despite its exceptional accuracy, GPS is subject to numerous error sources. Clearly, the major error sources must be included in the DGPS model. Error sources are:

- atmospheric delays
- Selective Availability
- clock differences
- ephemeris error
- multipath
- receiver noise
- Dilution of Precision (DOP)

Each of these error sources is discussed in detail in the following sections.

1. Atmospheric Delays

a. *Ionospheric Delays*

The ionosphere is a layer of charged particles between 100 and 1000 kilometers above the earth's surface. These particles interact with the transmitted

GPS signal and slow it, increasing pseudoranges. The equation describing this delay is

$$\Delta t = \frac{40.3}{cf^2} TEC, \tag{2.17}$$

where Δt is the delay in seconds, c is the speed of light (3×10^8 m/s), f is the system frequency (1575.42 MHz for L1), and TEC is the Total Electron Content (electrons/m^2) along the signal's path. The TEC is strongly effected by the solar cycle, season, time of day, and latitude. TEC maxima occur:

- daily: 1400 local
- seasonally: spring equinox
- solar cycle: every 11 years (next 2001-2002)
- geographic: 20° magnetic latitude

It varies ± 25% rms at all latitudes during daylight [Ref. 4, section 2.5, p. 24]. The pseudorange error due to ionospheric effects can be as great as 40 meters [Ref. 4, section 2.5, p. 13].

The algorithm described in [Ref. 6] which removes 55–60% of the ionospheric delay is based on Figure 2.5, which shows the typical diurnal variation of the ionospheric delay. The "ACTUAL DATA" curve shown in Figure 2.5 is modeled with the "COSINE MODEL" curve, a half-cosine. The equation of the model curve is

$$\Delta t = DC + A \cos \frac{2\pi(t - T_p)}{P}, \tag{2.18}$$

where DC, A, T_p, and P (constant offset, amplitude, phase, and period, respectively) describe the diurnal variation of the ionospheric delay. DC and T_p are assumed constant at five nanoseconds and 1400 local time, respectively. Amplitude and period

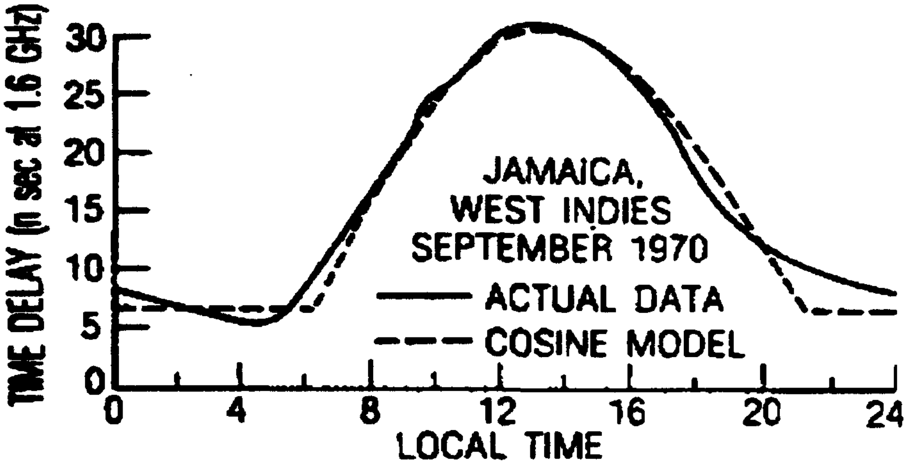

Figure 2.5: Diurnal Ionospheric Delay from [Ref. 6, p.208]

are each modeled as four term power series as follows:

$$\begin{aligned} A &= \sum_{n=0}^{3} \alpha_n \phi_m^n \\ P &= \sum_{n=0}^{3} \beta_n \phi_m^n, \end{aligned} \tag{2.19}$$

where the α_n's and β_n's are constants which are broadcast in the GPS navigation message, chosen based on the day of the year and average solar flux over the past five days, and ϕ_m is the geomagnetic latitude of the ionospheric subpoint. The ionospheric subpoint is the intersection of the line between the space vehicle and the receiver with the surface at the mean height of the ionosphere.

Next, one must first find the subtended earth angle between the user and the satellite, EA (degrees)

$$EA = \frac{445}{el + 20} - 4, \tag{2.20}$$

where el is the elevation angle of the satellite with respect to the user in degrees. Knowing EA, the geodetic location of the ionospheric subpoint can be approximated by

$$\phi_I \;=\; \phi_{rcvr} + EA\cos az$$

$$\lambda_I = \lambda_{rcvr} + \frac{EA\sin az}{\cos\phi_I}, \tag{2.21}$$

where ϕ and λ denote geodetic latitudes and longitudes, respectively and az is the azimuth of the satellite with respect to the receiver. Now the geodetic latitude can be converted to geomagnetic latitude (the required quantity) with the following approximation:

$$\phi_m = \phi_I + 11.6\cos(\lambda_I - 291), \tag{2.22}$$

where all angles are in degrees.

The dimensionless scale factor (SF) which scales the entire delay is

$$SF = 1 + 2(\frac{96 - el}{90})^3. \tag{2.23}$$

The final expression for Δt is a three term Taylor series expansion of Equation 2.18 is

$$\Delta t = \begin{cases} SF \cdot [DC + A(1 - \frac{x^2}{2!} + \frac{x^4}{4!})] & \text{for } \mid x \mid < \frac{\pi}{2} \\ SF \cdot [DC] & \text{for } \mid x \mid \geq \frac{\pi}{2} \end{cases}$$

where

$$x = \frac{2\pi(t - T_p)}{P}. \tag{2.24}$$

Due to the 25% rms variation, the error is modeled with a 25% standard deviation. Therefore, the random part of the error remains within ±25% of nominal 68% of the time.

b. *Tropospheric Delays*

The lower section of the atmosphere also causes signal propagation delays. Typical tropospheric delay is approximately two meters for 90° satellite elevation (directly overhead) up to 28 meters at a five degree elevation angle [Ref. 7, p. 218]. In this application, the atmosphere can be modeled as being composed of "wet air" and "dry air". Dry air is responsible for 90% of the total tropospheric delay, whereas, wet air is responsible for only ten percent. While the moisture content in the troposphere is virtually impossible to model accurately, this inaccuracy has minimal impact. Numerous models which calculate the tropospheric delay have been developed. Black developed the following model in [Ref. 8]. Let

$$\begin{aligned}
\Delta s &= \Delta s_d + \Delta s_w, \\
\text{where} & \\
\Delta s_d &= 2.343P \cdot [\frac{(T-4.12)}{T}] \cdot I(h = h_d, E), \\
\Delta s_w &= k_w \cdot I(h = h_w, E), \\
I(h,E) &= \{1 - [\frac{\cos E}{(1+(1-l_c)\cdot \frac{h}{r_s}}]^2\}^{(-1/2)}, \\
h_d &= 148.98(T-4.12)\,\mathrm{m}, \\
h_w &= 13,000\,\mathrm{m}, \\
k_w &= 0.2, \\
r_s &= 6378137\,\mathrm{m}, \\
P &= 1\,\mathrm{atm}, \\
T &= 15\,^{\circ}\,\mathrm{C}, \qquad (2.25)
\end{aligned}$$

and Δs is the wet or dry delay in meters, r_s is the distance from the center of the earth to the station, P is the surface pressure in atmospheres, E is the satellite elevation angle, and T is the surface temperature in degrees Celsius. It should be noted that l_c is an empirical constant. The value of 0.85 is only valid for elevation angles above five degree (GPS receivers typically ignore satellites at lesser elevation angles). Similarly, k_w is an empirical constant which varies based on latitude and season. The value 0.20 corresponds to the value for spring or fall in mid-latitudes.

This model has been shown to be virtually exact at elevation angles greater than 40°, with its worst error of about 0.045 m occurring between five and ten degrees of elevation.

This tropospheric model is assumed to vary 15% from the nominal value. Therefore, it is modeled with a 7.5% nominal standard deviation. This maintains the random part of the error within 15% of the model value 95% of the time.

2. Selective Availability

Selective Availability is a method that the Department of Defense can use to intentionally degrade the accuracy of pseudorange measurements. Typically, this is accomplished by dithering the space vehicle clock signal. Dithering the clock involves encoding the binary time signal the space vehicle broadcasts. The decryption process is classified and available only to DOD authorized users.

The use of SA essentially results in the satellites' "lying" to the receiver about their position. Clearly, this adversely impacts precision. Currently, SA is in operation on all Block II space vehicles which comprise the majority of the constellation. The DOD's stated goal for the positioning accuracy under SA is 100 meters (twice rms) for a two-dimensional fix [Ref. 9]. According to [Ref. 10, p. 420], the selective availability error can be modeled as a zero-mean, five meter standard deviation low frequency noise. The suggested cutoff frequency for SA noise is 1/180 Hz.

3. Clock Differences

The clock model used in this treatment of DGPS is a two state model shown in Figure 2.6. It is reasonable to expect the clock to have both a bias and drift. From daily exposure to clocks, the average person realizes that most clocks are slightly offset from correct time (bias), and that their accuracy tends to degrade with time (drift). From an engineering standpoint, these two phenomena can be best modeled with zero mean, white, Gaussian noise. Rewriting the model in state space

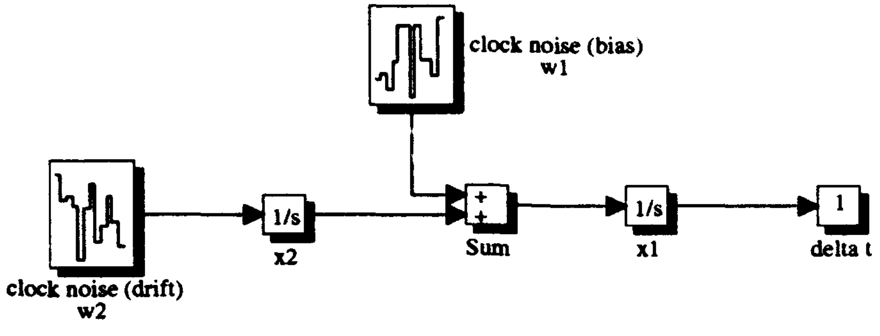

Figure 2.6: Receiver Clock Model

form for further analysis yields

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w \\ &\text{where} \\ E(ww^T) &= \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix}. \end{aligned} \tag{2.26}$$

The covariance of the clock error state (x_1) can be found by solving the Lyapunov equation which can be found numerous control textbooks, one of which is [Ref. 11, p. 104]. This equation must be solved over a finite time interval (Δt) because the two state clock model is unstable. This interval would normally be the sampling time, if the model were discrete. The result is

$$E(x_1^2) = S_1 \Delta t + \frac{S_2 \Delta t^3}{3}. \tag{2.27}$$

By taking the square root of the variance and dividing by Δt, one finds the more standard clock parameter the Allan variance, V_A

$$V_A = \sqrt{\frac{S_1}{\Delta t} + \frac{S_2 \Delta t}{3}}. \tag{2.28}$$

A representative plot of the two state model Allan variance as a function of averaging time (*i. e.* Δt) is shown in Figure 2.7.

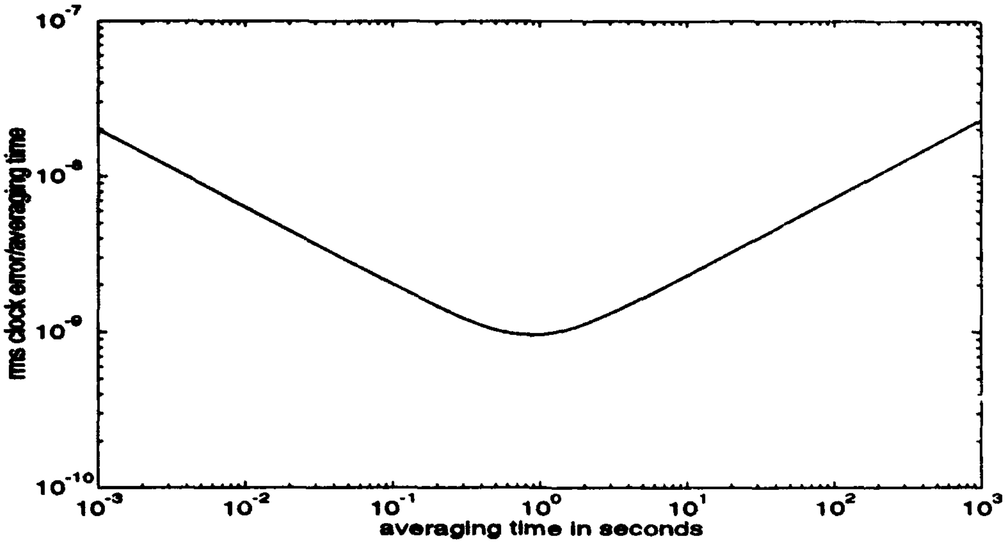

Figure 2.7: Ideal Allan variance

Real clocks behave somewhat differently from this simple model. A typical Allan variance plot for a real clock is shown in Figure 2.8.

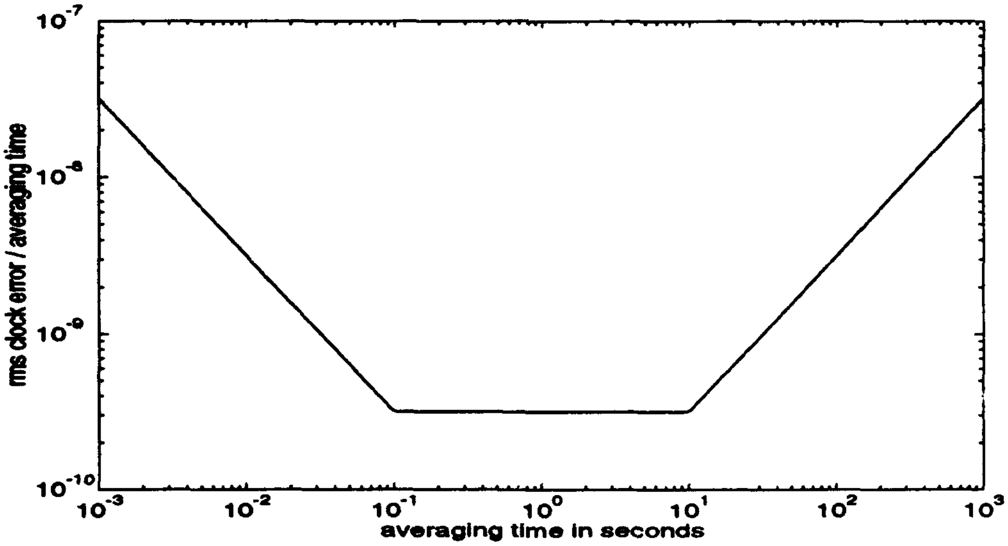

Figure 2.8: Real Allan variance

The flat portion of the curve is called the flicker floor. It is the result of a non-linear effect which cannot be modeled by the two state model. This causes a significant discrepancy between this simple model and the real world.

In order for the model to better represent reality, it must be carefully crafted to fit the actual plot as much as possible. By carefully choosing the values of S_1 and S_2, it is possible to make the actual and model curves fairly close. The key Allan variance parameters between 0.1 and ten seconds of averaging time are h_0, h_{-1}, and h_{-2} [Ref. 12]. Values of these three parameters for three common GPS timing standards are shown in the Table 2.1 from [Ref. 10, p. 428].

TABLE 2.1: ALLAN VARIANCE PARAMETERS FOR THREE COMMON TIMING STANDARDS

timing standard	h_0	h_{-1}	h_{-2}
crystal	2×10^{-19}	7×10^{-21}	2×10^{-20}
ovenized crystal	8×10^{-20}	2×10^{-21}	4×10^{-23}
Rubidium	2×10^{-20}	7×10^{-24}	4×10^{-29}

Brown [Ref. 10] finds that one must choose where the 2-state model is accurate. Since normal averaging times are in the 0.1 to ten second interval already mentioned, this is the region where the model is made accurate. Maximizing the accuracy of the model in that region dictates the following values for the noise spectral densities:

$$\begin{aligned} S_1 &\sim \frac{h_0}{2} \\ S_2 &\sim 2\pi^2 h_{-2}\ \mathrm{s}^{-2}. \end{aligned} \tag{2.29}$$

In order to remain conservative in this generic model of DGPS, the least accurate clock — the crystal clock — is used. The values for S_1 and S_2 for this clock are

$$\begin{aligned} S_1 &= 4 \times 10^{-19} \\ S_2 &= 1.58 \times 10^{-18}\ \mathrm{s}^{-2}. \end{aligned} \qquad (2.30)$$

These values are be used in the DGPS error model.

4. Ephemeris Error

In converting the pseudoranges of at least four satellites (six in this model) to a three–dimensional position and clock error, one must solve a series of non-linear, coupled algebraic equations. In these equations, the positions of the satellites are critical. The only way a GPS receiver or navigation filter knows the space vehicle positions is the navigation message. If broadcast satellite positions are incorrect, the accuracy of the resulting receiver position suffers. The control segment of the GPS system maintains positions on the entire constellation quite accurately.However, it would be folly to expect the satellites to broadcast inerrantly accurate positions. Typical ephemeris inaccuracies according to [Ref. 13] are shown in Figure 2.9.

These errors are resolved in a coordinate system local to each space vehicle. The three mutually perpendicular directions are radial, along track, and cross track. Because the DGPS model developed in this thesis does not account for satellite motion, along track and cross track directions cannot be resolved. To remain conservative, these two errors are combined into a circular error in the plane they define. This error is modeled as zero mean, white, Gaussian noise with a standard deviation of 3.5 meters. Likewise, the radial error is modeled as zero mean, white Gaussian noise with a 0.5 meter standard deviation. Both of these errors make the stated accuracy the two rms point. In other words, the value stays within the stated accuracy 95% of the time.

Since the space vehicle location enters the model in earth centered, earth

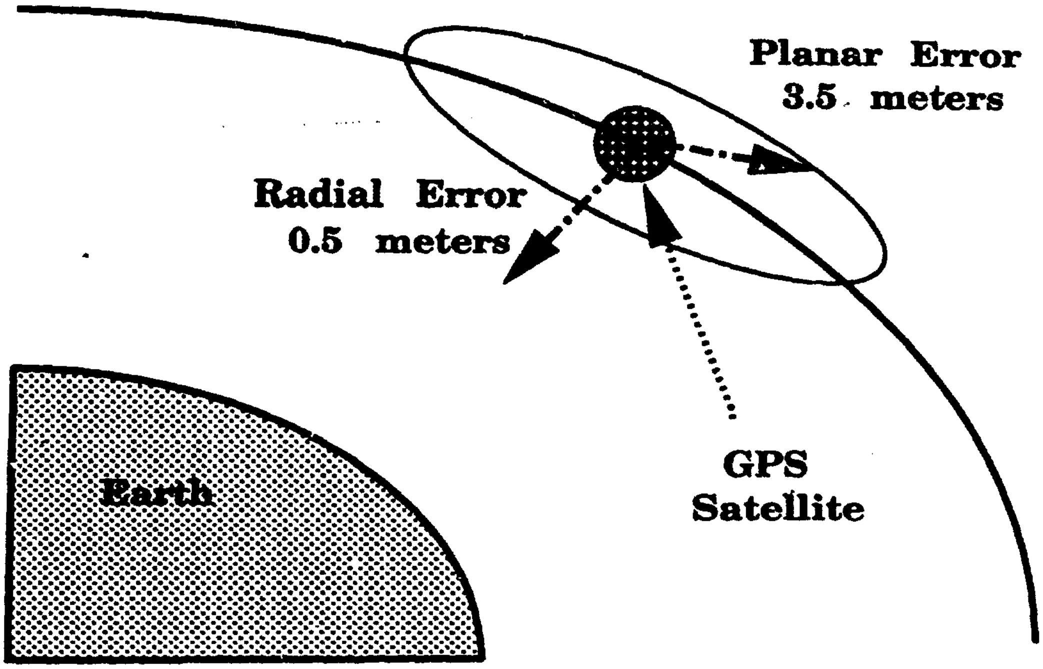

Figure 2.9: Ephemeris error

fixed Cartesian coordinates while the error is added in geodetic coordinates, a transformation is between the two coordinate systems must be performed (see Subsection 2. of Section B., Chapter II.). Having converted the satellite positions to geodetic coordinates, the random errors can be added. However, since the geodetic coordinates contain angles, the random errors in position must be converted to equivalent angles in latitude and longitude by dividing by the radius of the space vehicles' orbits, ~26,560,000 meters. The position and the error are now in compatible coordinates and can thus be summed.

5. Multipath

The signal radiated by a satellite is not required to take a direct path to a receiver. If the signal encounters an electromagnetically reflective object, it may bounce off that object and still find its way to the receiver. This signal has now traveled a greater distance than the straight line joining the space vehicle and the receiver. Because the receiver assumes the direct path is used, this *multipath* phenomenon can introduce pseudorange errors. Due to the satellite/receiver geometry, multipath is far more likely at low elevation angles. The GPS system has several attributes that minimize multipath effects [Ref. 4]:

- The L band frequency (1227.6 MHz) tends to undergo diffuse rather than specular reflection.
- The receiver antennas tend to reject multipath signals.
- The navigation message is broadcast with circular polarization. Circularly polarized signals undergo reversal upon reflection.
- GPS receivers generally use mask angles (the elevation angle below which the satellite is ignored) of at least five degrees.

All of these factors tend to attenuate the strength of any reflected signal making the multipath effect insignificant. Therefore, it is not modeled.

6. Receiver Noise

All receivers corrupt the signals they receive. GPS receivers are no exception. Inaccuracies resulting from quantization error, loop tracking errors and other hardware inadequacies corrupt the pseudorange accuracy. According to [Ref. 4], representative GPS receiver noise has a standard deviation of 7.5 m. Receiver noise is thus modeled as a zero mean, white Gaussian noise with 7.5 m standard deviation. As the differential station and the aircraft receivers are assumed identical, both aircraft pseudoranges and differential station pseudoranges are contaminated with this noise. The noise in the two receivers is assumed to be independent.

7. Dilution of Precision

All of the errors discussed to this point directly effect pseudorange accuracy. The various delays and noise sources can cause the receiver's evaluation of range to the space vechicle's to be inaccurate. However, the pseudoranges themselves are irrelevant. The science of navigation is concerned with *positioning.* Dilution of Precision is the effect that links pseudorange accuracy to position accuracy. DOP can be further classified into several different types:

- VDOP — Vertical DOP (z)
- HDOP — Horizontal DOP (x,y)
- PDOP — Position DOP (x,y,z)
- TDOP — Time DOP (t)
- GDOP — Geometric DOP (x,y,z,t)

The equation which relates DOP and pseudorange error is

$$\sigma_p = \text{DOP} \cdot \sigma_0, \tag{2.31}$$

where σ_p is the standard deviation of the position error and σ_0 is the standard deviation of the pseudorange error, often called the User Equivalent Range Error (UERE).

Dilution of Precision is a function of satellite to receiver geometry. As Figure 2.10 shows for a four satellite constellation, GDOP is minimized with the space vehicles spread out as much as possible. In fact, the volume of the tetrahedron formed by the unit vectors from the receiver to each satellite is an empirical measure of DOP. PDOP is inversely proportional to the volume of the tetrahedron. If, for example, all of the space vehicles a receiver was using for positioning were in the same plane, PDOP would approach infinity (the volume of the tetrahedron would be

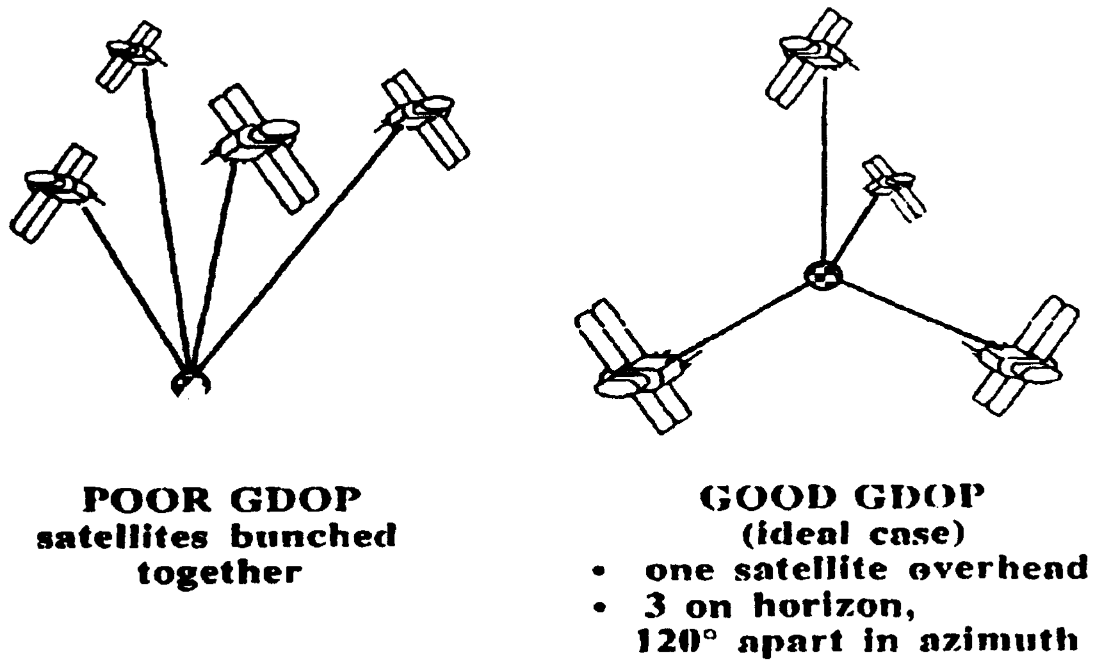
POOR GDOP
satellites bunched together
GOOD GDOP
(ideal case)
• one satellite overhead
• 3 on horizon, 120° apart in azimuth

Figure 2.10: Dilution of Precision from [Ref. 5, p. 4.22]

zero). Likewise, minimum PDOP is achieved with the geometry shown at the right of Figure 2.10.

The goal for the design of the entire GPS constellation of satellites is a PDOP no greater than six everywhere on the earth. With the entire set of 24 space vehicles now in orbit, users can expect PDOP values under three [Ref. 14].

The entire concept of GPS navigation has now been thoroughly discussed. All of the information put forth in this discussion will be used in developing the DGPS computer model in the next chapter. To complete the sensor discussion, a description of INS follows.

E. INERTIAL NAVIGATION

Inertial navigation has long been the standard for self-contained, long-range aircraft navigation. An Inertial Navigation System (INS) senses aircraft thrust acceleration, angular rates and spatial orientation resolved in an orthogonal system and computes the inertial acceleration. This acceleration can now be integrated — once to find velocity, and twice to find position.

The primary component in any inertial navigation system is the Inertial Measuring Unit (IMU). The IMU is composed of three accelerometers, three rate gyros, and two inclinometers. The accelerometers measure thrust acceleration. Thrust acceleration is composed of linear, centripetal, and gravitational effects. Einstein's Principle states that it is impossible for a sensor to distinguish between the effects of gravity and acceleration. Thus, the thrust acceleration that it provides is

$$^{B}a = ^{B}\dot{v} + ^{B}\omega \times ^{B}v + ^{B}g, \tag{2.32}$$

where ^{B}a is the thrust acceleration, $^{B}\dot{v}$ is the linear acceleration, $^{B}\omega$ is the angular velocity, ^{B}v is the velocity, and ^{B}g is gravity. All quantities are in the body frame. This principle necessitates that the computation portion of the INS compute and remove local gravity from the measured accelerations.

Rate gyros sense angular velocities. These angular velocities can be resolved in either the body or inertial frame, depending on the type of IMU implementation. The two inclinometers sense aircraft inertial orientation, *i.e.*, Euler angles.

There are two conceptual methods for implementing inertial navigation:

- Gimbaled IMU
- Strapdown IMU

Brief discussions of each method follow.

1. Gimbaled IMU

A gimbaled IMU rotates about its four gimbals during operation (see Figure 2.11. Aircraft IMU's must have four gimbals to prevent gimbal lock, while earthbound IMU's require only three gimbals [Ref. 15, p. 192]. A controller maintains

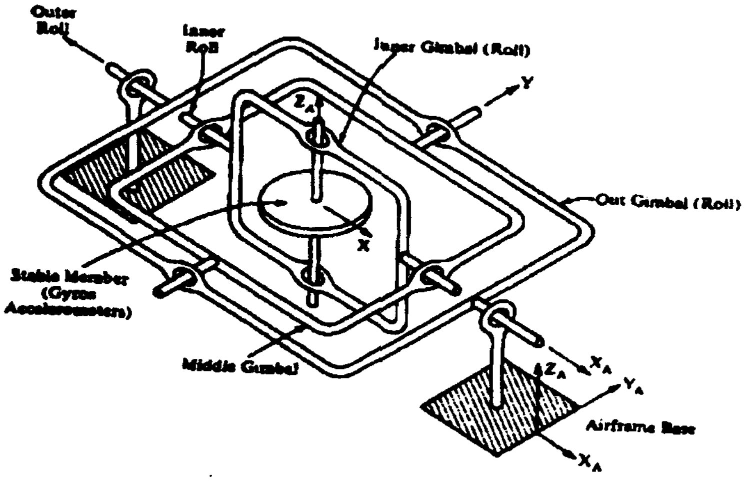

Figure 2.11: Gimbaled IMU from [Ref. 15, p. 193]

the IMU in a constant inertial orientation toward true North and lying in the locally horizontal plane. By maintaining this orientation, the gimbaled IMU measures inertial quantities directly. This data can be integrated without transformation to yield inertial velocity, position and Euler angles. From a navigation standpoint, this configuration seems ideal. However, gimbaled systems are large and heavy, making them impractical for small aircraft. Also, kinematic quantities resolved in the aircraft–fixed coordinate system, necessary for stability augmentation and control, are not directly available. Instead, they must be computed through a series of Euler rotations. The computations required take time and thus introduce delays into an often time critical control problem. This fact makes the gimbaled system less than desirable for control.

2. Strapdown IMU

Strapdown IMU is conceptually the reverse of the gimbaled system mentioned above. Rather than maintaining a constant inertial orientation, a strapdown system is "strapped down" to the aircraft, thus maintaining a constant orientation in the aircraft–fixed coordinate system. Therefore, the output of the IMU is resolved in the local coordinate system. Kinematic quantities are immediately available for the control system. The extra computational burden now rests on the navigation computer which must transform the accelerations and angular velocities sensed in the local coordinate system to the inertial system. Currently, most inertial systems are of the strapdown variety. With the advent of high speed, low cost, lightweight computing power, the required calculations in transforming from the aircraft-fixed to the inertial coordinate system are no obstacle to navigation. This is the system which is modeled in this thesis.

F. INS COMPUTATIONS

In the strapdown configuration, the IMU measures angular velocities and thrust acceleration in the body frame, as well as Euler angles. However, the INS must provide position and orientation, both in the inertial frame. In order to compute position and orientation in the inertial, tangent plane coordinate system, inertial accelerations and Euler rates must be calculated.

Before converting the inertial acceleration from the body frame to the inertial frame, it is necessary to compute the inertial orientation. Computing the Euler rates is a tricky endeavor. The Euler rates are simply related to the body angular rates by Poisson's equation

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2.33}$$

where p, q, and r are the components of ${}^{B}\omega$, the body's angular velocity. Obviously, this formula requires the exact Euler angles. However, one can only measure these angles directly at very low frequency. At frequencies greater than a fraction of a Hertz, one must integrate the Euler rates found from Equation 2.33 to find the angles. This presents the seeming paradox of needing to know the Euler angles in order to find the Euler rates which must be integrated to find the Euler angles. This process must be implemented recursively. That is, the results of the integration to find the angles must be fed back into Equation 2.33. Furthermore, a complementary Kalman filter is necessary to provide an optimal estimate of the Euler angles, trusting the inclinometers at the low frequencies and the rate gyros at high frequencies. This process will be discussed in detail in Chapter IV.

Now having inertial acceleration in expressed in the body frame and the orientation of the body (Euler angles), the transformation from body to inertial can be executed. This coordinate transformation is defined by

$$ {}^{U}\ddot{P} = {}^{U}_{B}R\, {}^{B}a + {}^{U}g, \tag{2.34} $$

where Φ, Θ, and Ψ are the roll, pitch and yaw Euler angles, respectively, ${}^{U}\ddot{P}$ is the aircraft's acceleration in inertial coordinates, ${}^{B}a$ is thrust acceleration in the aircraft-fixed coordinate system, ${}^{U}g$ is gravity in inertial coordinates and ${}^{U}_{B}R$ is the transformation matrix from the body-fixed coordinates to inertial tangent plane coordinates as follows

$$ \begin{bmatrix} \cos\Psi\cos\Theta & \cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi \\ -\sin\Psi\cos\Theta & -\sin\Theta\sin\Phi\sin\Psi + \cos\Psi\cos\Phi & -\sin\Theta\cos\Phi\sin\Psi - \cos\Psi\sin\Phi \\ \sin\Theta & -\cos\Theta\sin\Phi & -\cos\Theta\cos\Phi \end{bmatrix}. $$

It is ${}^{U}\ddot{P}$ which can be integrated to provide velocity and position in the inertial, tangent plane frame.

G. INS ERROR SOURCES

The Inertial Measuring Unit is subject to a few main error sources. These are:

- bias
- cross–axis sensitivity
- noise floor

All of these errors apply to both accelerometers and rate gyros. Brief descriptions of these problems follow.

1. Biases

A bias in an accelerometer or a rate gyro is defined as a constant offset of the output of the device from the true value. In other words, an accelerometer might constantly read 0.01 m/s^2 while the device is not accelerating. This 0.01 m/s^2 would be referred to as a bias. This error is modeled with a series of small step functions, one for each accelerometer and rate gyro.

2. Cross–Axis Sensitivity

Cross–axis errors are caused by misalignment of the IMU with the aircraft coordinate axes. Ideally, the components of the IMU — the accelerometers and the rate gyros — would each be perfectly aligned with the three axes of the aircraft–fixed coordinate system. Unfortunately, even the highest fidelity inertial sensors are never precisely coincident with the appropriate axes. Misalignment of both types of devices causes errors. The cross–axis errors are modeled with the following equation:

$$a_{ca} = \begin{bmatrix} 0 & e_Y & e_Z \\ e_X & 0 & e_Z \\ e_X & e_Y & 0 \end{bmatrix} a, \tag{2.35}$$

where a_{ca} is the cross–axis error, e_X, e_Y, and e_Z are the cross–axis error terms and a is the actual acceleration resolved in the aircraft coordinate system. The e's from the previous equation are determined by the amount of angular offset of each sensor from the correct position.

3. Noise Floor

All sensor devices corrupt the quantities they measure. Various factors including thermal noise can cause a constant output of white noise, regardless of the actual acceleration (or angular velocity). This noise floor makes accelerations below it not measurable. That is, the output of the sensor still includes the actual quantities, but it is "invisible" because the noise floor obscures it. This process is modeled by introducing a threshold to the actual acceleration in the aircraft coordinate system and adding white noise. Taking these steps result in the accelerometer always reporting noisy, zero–mean acceleration unless the actual value is above the threshold. When the actual value exceeds the threshold, that value is added to the noise floor value (and the bias) to create the output of the accelerometer. This process also applies to the rate gyros.

Now both of the sensors — GPS and INS — have been completely investigated. In the following chapter, computer models of both sensors are constructed using the basic principles of operation of each sensors and all of the error sources.

III. SIMULINK MODEL

A. DGPS SIMULINK MODEL

The discussion of the basic principles of GPS navigation and the various phenomena effecting its performance can now be used to model the system. The system shown in Figure 3.1 is a complete working model of Differential GPS constructed with SIMULINK.

SIMULINK is a non-linear simulation package based on MATLAB©. This language uses block diagrams and embedded MATLAB routines to model arbitrarily complex systems. SIMULINK allows simulations to be carried out in continuous or discrete time. It also has several choices of integration routine — Euler, Adams, Adams–Gear, Runge–Kutta, *etc.* Simulation results can be viewed as they happen or sent to the MATLAB workspace for future analysis or plotting.

Figure 3.1 depicts the inputs to the DGPS simulation — "Aircraft position", "Satellite positions", and "Differential Station position" — which are used to compute pseudoranges. The receiver position and the satellite positions are *multiplexed* together before they enter the "Aircraft Pseudorange" block (multiplexing assembles an arbitrary number of separate vectors into one vector for use by a program block). The "Aircraft Pseudorange" block produces pseudoranges complete with ephemeris error, ionospheric delay, and tropospheric delay. The receiver noise, clock error, and Selective Availability error, are subsequently added to the pseudoranges. The remaining portion of the simulation removes some of the errors. The "correct Pseudoranges" block takes receiver position, satellite positions, and corrupted pseudoranges as a multiplexed input vector, and removes the non-random tropospheric and atmospheric

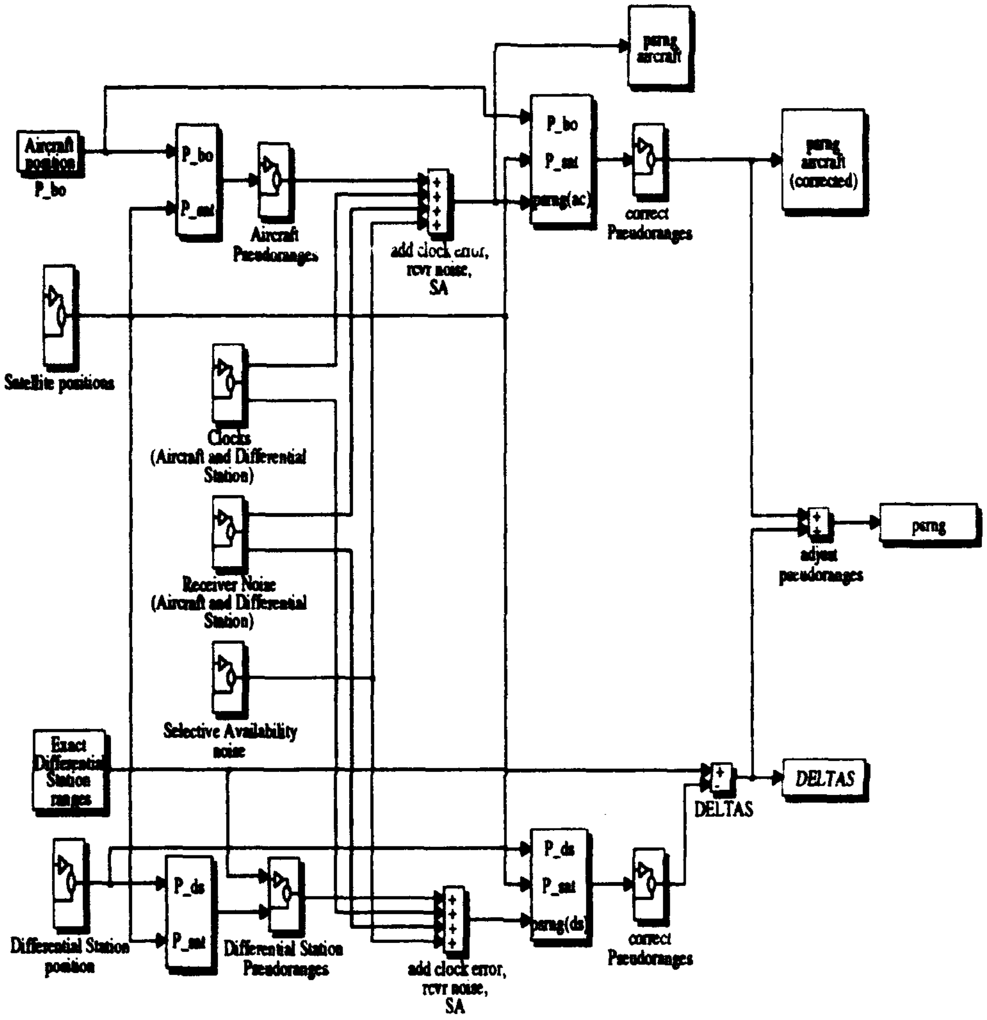
parng aircraft
Aircraft position
P_bo
P_bo
P_sat
Aircraft Pseudoranges
add clock error, rcvr noise, SA
P_bo
P_sat
parng(ac)
correct Pseudoranges
parng aircraft (corrected)
Satellite positions
Clocks (Aircraft and Differential Station)
Receiver Noise (Aircraft and Differential Station)
Selective Availability noise
adjust pseudoranges
parng
Exact Differential Station ranges
DELTAS
DELTAS
Differential Station position
P_ds
P_sat
Differential Station Pseudoranges
add clock error, rcvr noise, SA
P_ds
P_sat
parng(ds)
correct Pseudoranges

Figure 3.1: DGPS Model

delays. Lastly, the atmospherically corrected pseudoranges are further corrected using the "DELTAS" from the differential station. Six differential pseudoranges are outputs of the model and become inputs to a Kalman filter.

The calculation of the "DELTAS" is accomplished as follows. As in the Aircraft pseudorange calculations above, the "Differential Station Pseudoranges" accept the Differential Station position and the Satellite positions as a multiplexed vector input. The pseudorange is calculated with a routine identical to that used in calculating the Aircraft Pseudoranges, but with independent random variations. Corrupted differential station pseudoranges are outputs of the "Differential Station Pseudoranges" block. Clock error, receiver noise, and selective availability noise are added. The "correct Pseudoranges" block is identical to the one used to adjust the aircraft pseudoranges. Having used the Pythagorean theorem to pre–compute the exact ranges from the differential station to the satellites, the simulation now calculates the "DELTAS", the error in the differential station pseudorange. These values are used to adjust the aircraft pseudoranges.

Next, each of the major sections of the model will be discussed in more detail.

1. Ephemeris Model

The "Satellite Position" block computes the noisy space vehicle positions in tangent plane (Local Geodetic) coordinates for use by the DGPS model (see Figure 3.2). The satellite positions enter from the left in earth-centered, earth-fixed Cartesian coordinates (Conventional Terrestrial System). The block "Convert to latitude, longitude, elevation", a MATLAB m–file *ecef2ll.m* which encodes the transformation defined in Subsection 2., Section B. in Chapter II. , must first be used to transform the satellite positions to the geodetic system. Next, the ephemeris errors, resolved in the geodetic system, are added to the space vehicle positions. Finally, the corrupted positions are transformed to the tangent plane system by the block "Con-

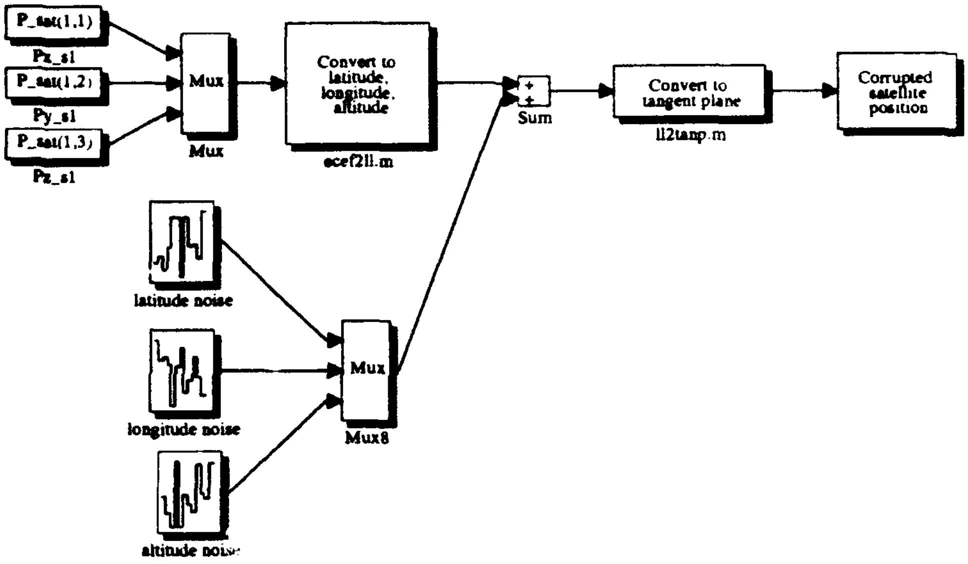
P_sat(1,1)
Px_s1
P_sat(1,2)
Py_s1
P_sat(1,3)
Pz_s1
Mux
Mux
Convert to latitude, longitude, altitude
ecef2ll.m
+
+
Sum
Convert to tangent plane
ll2tanp.m
Corrupted satellite position
latitude noise
longitude noise
altitude noise
Mux
Mux8

Figure 3.2: Ephemeris Model

vert to tangent plane", a MATLAB m-file *ll2tanp.m* which resolves the corrupted satellite location in the local geodetic or tangent plane coordinate system.

2. Pseudorange Computation

The pseudorange computation blocks "Aircraft Pseudoranges" and "Differential Station Pseudoranges" take space vehicle positions, receiver (either aircraft of differential station) position as applicable, take-off time, atmospheric pressure and temperature and computes the pseudoranges as shown in Figure 3.3. The block "convert seconds to minutes" merely converts the output of the clock in seconds to minutes and seconds. This value is added to the "take-off time" in order too maintain accurate time for use in the ionospheric delay calculation. The "Ionospheric delay" block uses this time and computes one delay to be applied to all six pseudoranges. Unlike the model previously discussed, a generic set of coefficients (α_n and β_n) has been assumed for this model yielding: $A=$ 20 nanoseconds and P=12 hours. The scale

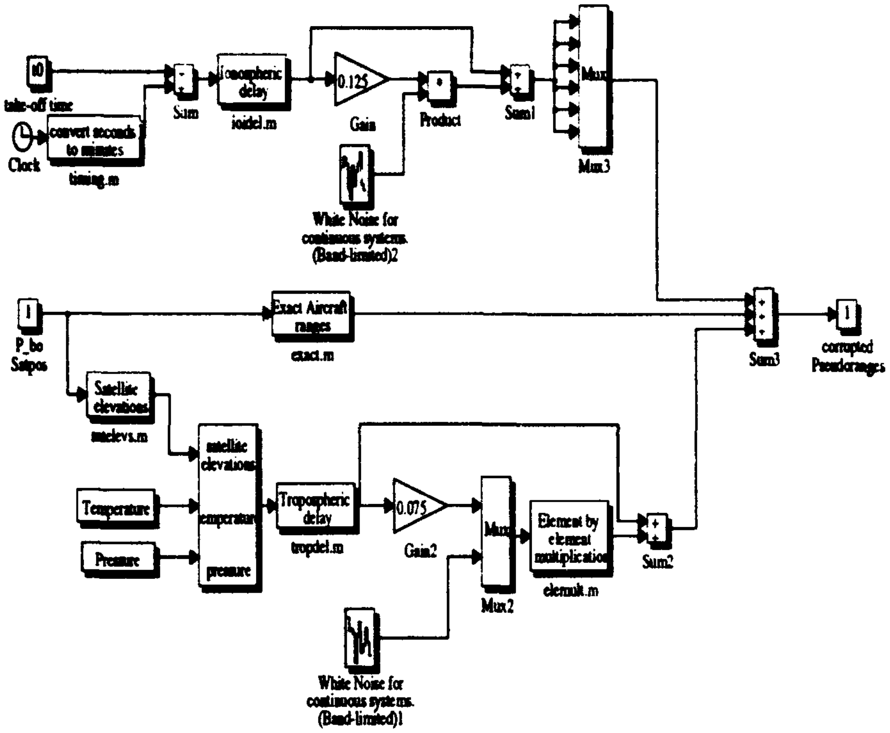
t0
take-off time
Clock
convert seconds to minutes
timing.m
Sum
Ionospheric delay
ioidel.m
0.125
Gain
Product
White Noise for continuous systems. (Band-limited)2
Sum1
Mux
Mux3
1
P_bo Satpos
Exact Aircraft ranges
exact.m
Sum3
1
corrupted Pseudoranges
Satellite elevations
satelevs.m
satellite elevations
temperature
pressure
Temperature
Pressure
Tropospheric delay
tropdel.m
0.075
Gain2
Mux
Mux2
White Noise for continuous systems. (Band-limited)1
Element by element multiplication
elemult.m
Sum2

Figure 3.3: Pseudorange Computation Model

factor *SF*, is assumed to be one. In order to achieve the 25% random error discussed in Section 1., the output of the "Ionospheric delay" block is multiplied by a gain of 0.125. The output of this gain block is multiplied by white, Gaussian noise with unity variance. by adding this value to the nominal delay, the desired nominal ± 25% delay is achieved 95% of the time. Since this value applies to all six pseuodranges, the one value is converted to a vector of six with the multiplexer.

The block "Exact ranges" uses the MATLAB subroutine *exact.m* to compute the exact geometric range from the receiver to each satellite. It accepts the receiver position and the positions of the six satellites as a multiplexed input vector. Using the Pythagorean theorem in three dimensions, this block outputs the exact ranges to all six space vehicles.

The remainder of the simulation diagram calculates the tropospheric delay. The "Satellite elevations" block accepts the receiver position and the satellite positions and computes the elevation angle to each satellite. Since all quantities are in the tangent plane Cartesian coordinate system already, this is a very simple operation. First, the vector from the receiver to the satellite is found by subtracting the receiver position vector from the satellite position vector

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} x_{sat} \\ y_{sat} \\ z_{sat} \end{bmatrix} - \begin{bmatrix} x_{rcvr} \\ y_{rcvr} \\ z_{rcvr} \end{bmatrix}. \tag{3.1}$$

Next, the vector is normalized to unit length

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \frac{1}{\sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}. \tag{3.2}$$

Finally, the elevation angle is given by

$$\theta = \arcsin \hat{z}. \tag{3.3}$$

Note that normalizing the vector and taking the arcsin is an equivalent process.

The "Tropospheric delay" block takes the output of the "Satellite elevations" block, the surface temperature, and the surface pressure as inputs to compute the tropospheric delay to each individual satellite. This block executes the MATLAB function *tropdel.m* which is the tropospheric delay algorithm discussed earlier implemented in MATLAB. The 7.5% random error is added by multiplying the output of the "Tropospheric delay" block, the six element vector of delays to the satellites, by

0.075 to form a "vector gain". This output is multiplied, element by element, by white noise with unity variance to produce a scaled noise vector. The result of this operation is the output of the block "Element by element multiplication". Adding this error vector to the nominal tropospheric delay vector yields the desired randomly varying delay vector.

The outputs of each of these processes — exact range, tropospheric delay, and ionospheric delay — are summed to form the noisy pseudoranges. An identical process occurs in finding the differential station pseudoranges. However, the noise added in the differential station calculation is uncorrelated with that added in the aircraft calculation.

3. Clock Model

The clocks of the differential station and the aircraft receiver are both modeled as previously shown in Figure 2.6. The variances of the white noise for the aircraft and differential station clocks are identical, but the *random seeds* are different. This makes the white noise in the two models uncorrelated. The outputs of the clock models are multiplexed into six element vectors and multiplied by the speed of light to convert them to range errors.

4. Selective Availability Model

Selective availability is modeled identically for each satellite. As shown in Figure 3.4, the error is white noise put through a low pass filter. As discussed in Section 2., the low pass filter has a time constant of three minutes (180 seconds). All of the white noise sources driving the errors for each satellite are independent because they have different random seeds.

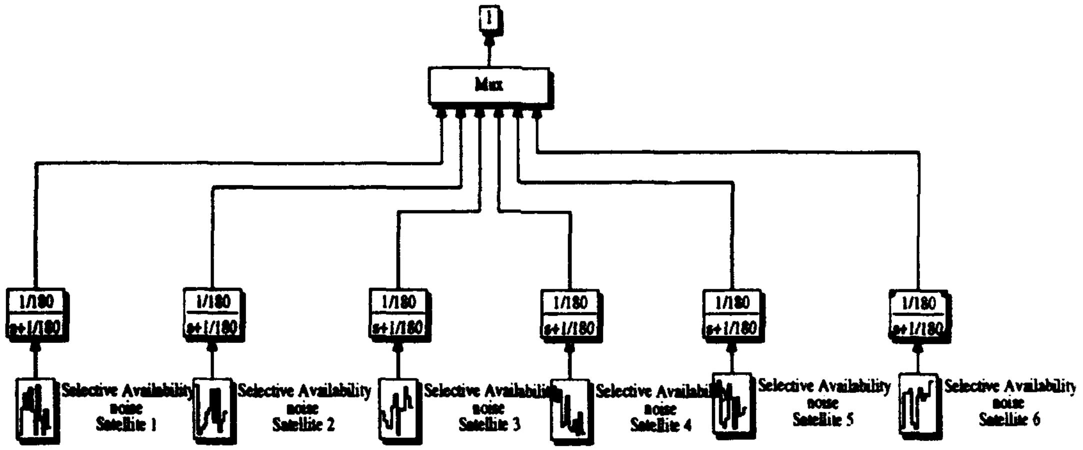
1
Max
1/180
s+1/180
1/180
s+1/180
1/180
s+1/180
1/180
s+1/180
1/180
s+1/180
1/180
s+1/180
Selective Availability noise Satellite 1
Selective Availability noise Satellite 2
Selective Availability noise Satellite 3
Selective Availability noise Satellite 4
Selective Availability noise Satellite 5
Selective Availability noise Satellite 6

Figure 3.4: Selective Availability Model

5. Pseudorange Correction

In a manner similar to the way the pseudoranges are calculated, the corrupted pseudoranges are corrected with known tropospheric and ionospheric models as shown in Figure 3.5. The only difference between this part of the model and the initial pseudorange calculation is that this portion subtracts the nominal delays in an attempt to remove the atmospheric errors. Obviously, this correction cannot compensate for the random part of the errors.

B. DGPS MODEL VERIFICATION

This model of Differential GPS was verified by simulating it under the following circumstances:

- receiver is stationary at the origin of the tangent plane system
- differential station is stationary at the origin of the tangent plane system

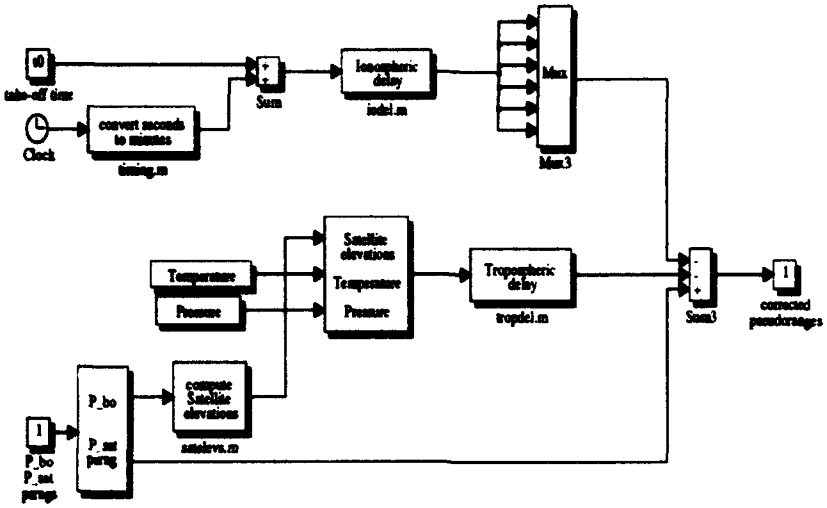
t0
take-off time
Clock
convert seconds to minutes
timing.m
Sum
Ionospheric delay
iodel.m
Mux
Mux3
Temperature
Pressure
Satellite elevations
Temperature
Pressure
Tropospheric delay
tropdel.m
Sum3
1
corrected pseudoranges
1
P_bo
P_sat
psrng
P_bo
P_sat
psrng
compute Satellite elevations
satelevs.m

Figure 3.5: Pseudorange Correction

- six satellites are uniformly distributed at elevation angles greater than ten degrees
- the origin of the tangent plane system is at 45° N latitude and 45° E longitude (on the earth's surface)

The values of the differential station exact ranges are computed directly (with the Pythagorean theorem) in the model. Since the aircraft receiver is stationary at the same location as the differential station, any differences between the aircraft pseudoranges and the exact ranges are errors. Comparison of the two outputs labeled "psrng aircraft", "psrng aircraft (corrected)", and "psrng" (for only satellite #1) confirms the validity of the model. That is, the magnitude of the mean error (the difference between the aforementioned outputs and the exact range) at each of these stages should decrease at each subsequent output. Examination of the "DELTAS" output reveals the benefit of the Differential correction.

1. Verification with Clock Error

a. *Atmospheric Correction*

As Figure 3.6 shows, the atmospheric correction subroutine decreased the pseudoranges. This figure depicts the difference between the pseudorange before it enters the correction block and after it exits. Since the ionospheric and tropospheric delays are assumed to always be positive, the correction routine should always reduce the pseudoranges, resulting in a positive correction on the plot. The increase in the atmospheric correction with time is due to the ionospheric error increasing because of the time of day. The simulation starts at 1300 local, one hour before the maximum ionospheric delay. Thus, the correction monotonically increases and would continue to increase until 1400 local.

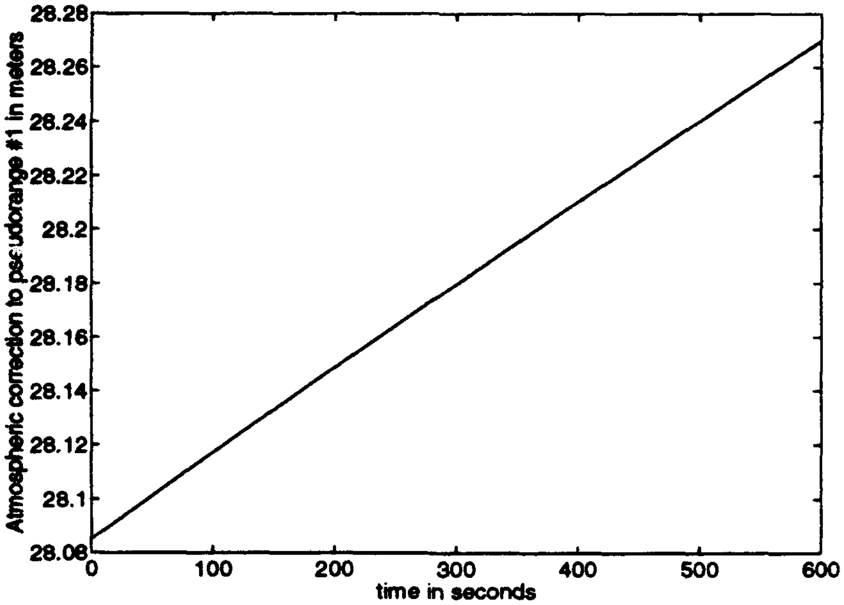

Figure 3.6: Atmospheric correction for pseudorange #1 (with clock error)

b. *Differential Corrections*

All differential corrections increased during the simulation time (ten minutes), reaching about 4000 meters for some of the satellites. Figure 3.7 shows the differential correction for pseudorange #1 monotonically increasing during the simulation. The reason for the constant increase in the differential correction is its clock instability. The differential correction algorithm can only interpret differences between the pseudorange and exact range to a given satellite as error. Unfortunately, this means that any error in the differential station clock results in an error in the differential correction. In fact, this situation causes differential correction #1 (as well as the other undepicted corrections) to increase so rapidly.

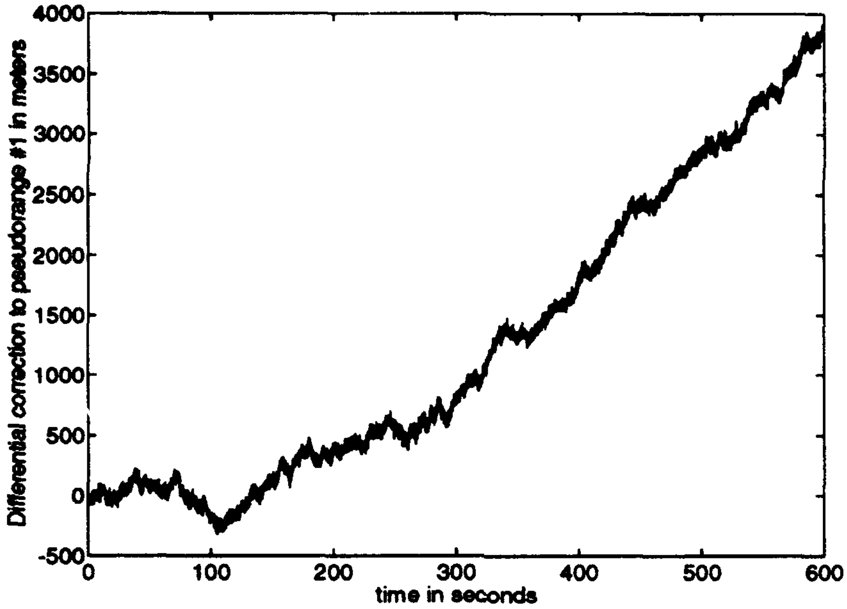

Figure 3.7: Differential Corrections for pseudorange #1 (with clock error)

c. *Differentially Corrected Pseudoranges*

This exacerbating effect of the differential clock error on the pseudorange error is further illustrated in Figure 3.8. While the pseudorange error is slowly

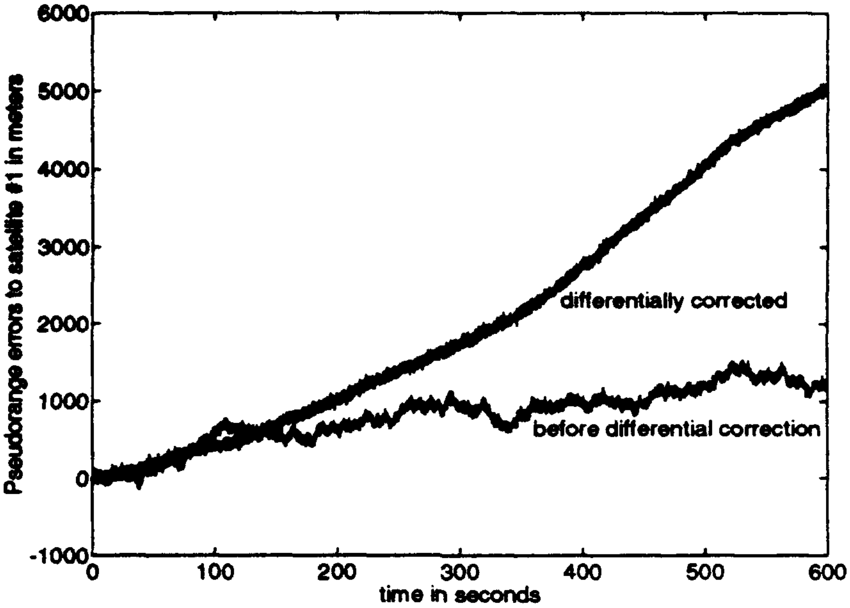

Figure 3.8: Corrected and differentially adjusted pseudorange errors to satellite #1 (with clock error)

diverging before it is differentially corrected, it is obviously worse after the correction. The reduced accuracy of the differentially corrected range is due to the clock divergence discussed in the previous section.

2. Verification without Clock Error

The clock error of the Differential station causes the differential correction to diverge, however, the clock error is removed by the Kalman filter. Therefore, to better understand the effect of the differential correction, the simulation should be repeated without clock errors. This should highlight the effects of selective availability, receiver noise, and the differential correction and can be achieved by simply disconnecting the clock error segment from the rest of the model.

In this "clock error free" model, one expects much better accuracy in the differentially adjusted pseudoranges. Selective availability errors should be largely canceled by the differential adjustment. The primary remaining errors are receiver noise, both on the aircraft and at the differential station. The mean pseudorange errors should be near zero; the standard deviation of the error should be one to seven meters [Ref. 2, p. 64].

a. *Atmospheric Correction*

As expected, the atmospheric correction remained the same (see Figure 3.9). It turns out that atmospheric effects are totally independent of the clock errors. Thus, they are expected to remain unchanged in this simulation.

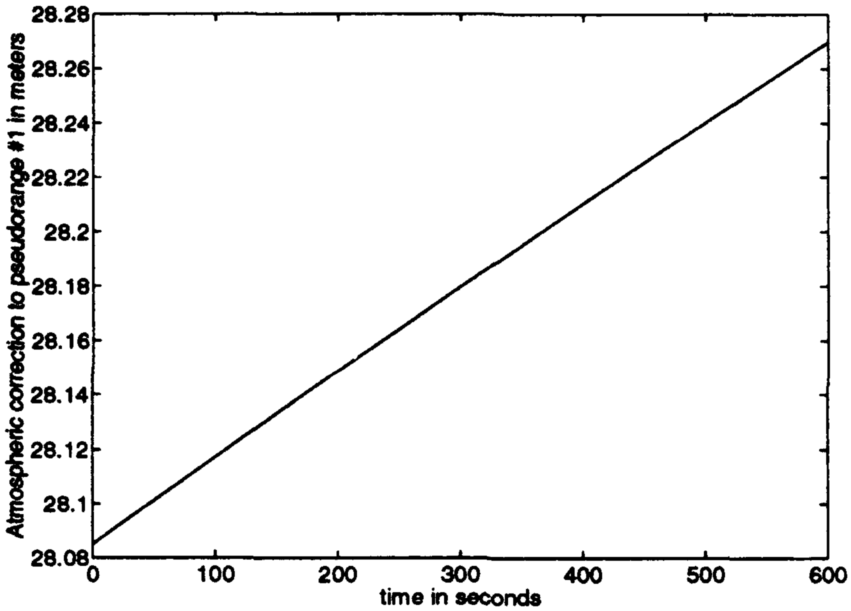

Figure 3.9: Atmospheric correction for pseudorange #1 (without clock error)

b. *Differential Corrections*

The benefit of the differential correction can be seen in Figure 3.10. The high frequency variations in the differential correction are due to receiver noise and ephemeris error. The low frequency variation is clearly due to selective availabil-

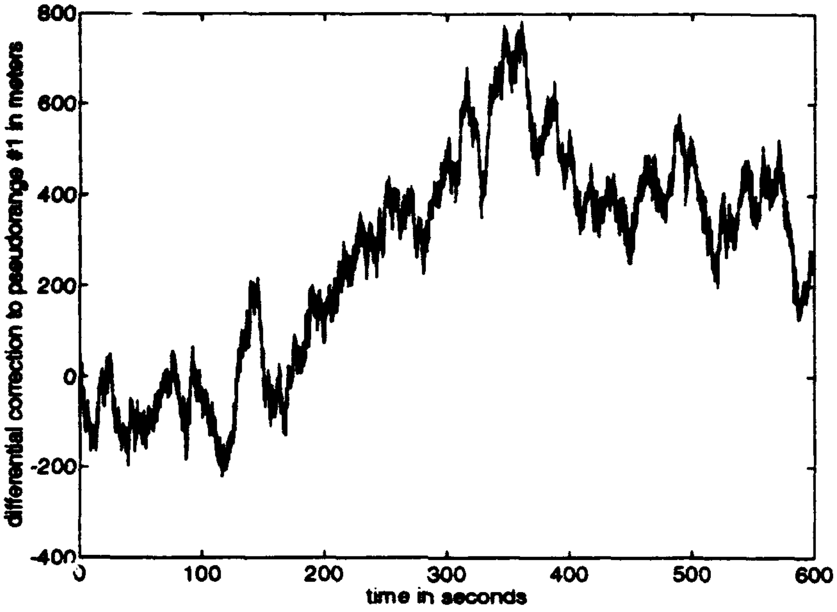

Figure 3.10: Differential Corrections for pseudorange #1 (without clock error)

ity. Selective Availability, being the only low frequency error in this model, must be causing the low frequency variation. Since SA is common to both the differential station and the aircraft, the differential correction accounts for it. Therefore, SA errors are eliminated from the aircraft pseudorange error by the differential correction.

c. *Differentially Corrected Pseudoranges*

Figure 3.11 clearly shows the benefits of Differential GPS. The differentially corrected errors are now virtually zero mean. That is, the differential adjustment has removed the biases from the pseudorange errors. The standard deviations of the differentially adjusted pseudorange errors are quite reasonable. Table 3.1 summarizes the means (μ) and standard deviations (σ) of the differentially corrected pseudorange errors for each satellite in meters. These figures show that the mean pseudorange errors to each satellite are practically zero. Clearly, the differential cor-

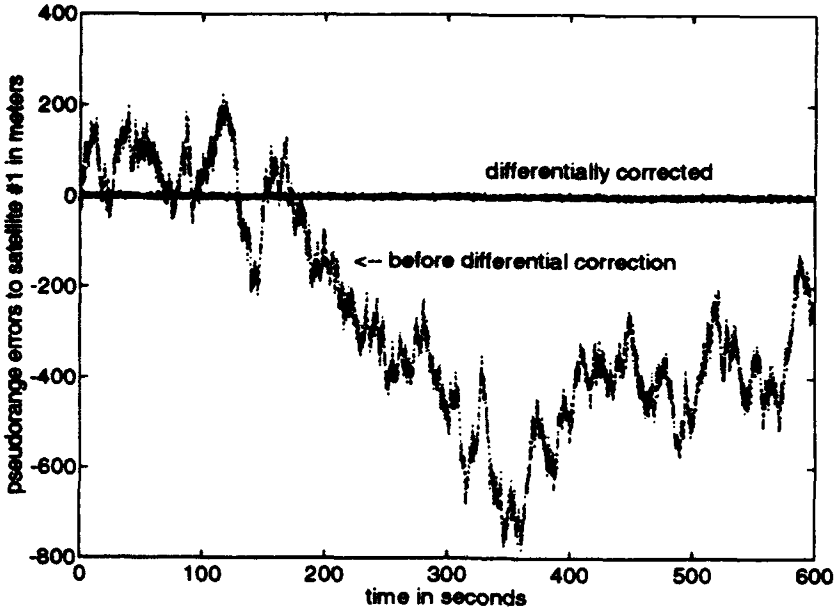

Figure 3.11: Corrected and differentially adjusted pseudorange errors to satellite #1 (without clock error)

TABLE 3.1: MEAN AND STANDARD DEVIATION OF DIFFERENTIALLY CORRECTED PSEUDORANGE ERRORS (WITHOUT CLOCK ERROR)

	Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6
μ	-0.0060	-0.0133	-0.0051	-0.0498	0.0024	0.0127
σ	2.765	1.831	1.602	1.653	2.456	2.319

rection has resulted in the error becoming zero mean.

The standard deviations are all on the order of two meters. Using a representative PDOP (Position Dilution of Precision) of three, the DGPS fixes from this system should be accurate to within four meters (one rms). These values are consistent with the one to seven meter values published for DGPS accuracy.

C. INS SIMULINK MODEL

The INS SIMULINK Model is composed of the "strapdown IMU" and the "INS Computations" blocks as shown in Figure 3.12. At this top level, the INS accepts

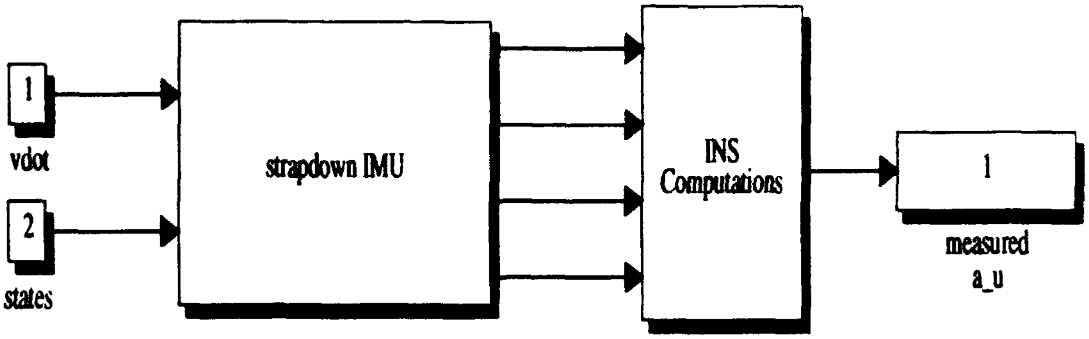

Figure 3.12: INS Model

"vdot" (${}^{B}\dot{v}$ and "states" ${}^{B}v$, ${}^{B}\omega$, Φ, Θ, and Ψ) from which it ultimately computes "measured a_u" (${}^{U}a$). The "strapdown IMU" block computes exact thrust acceleration and subsequently subjects it, as well as the angular rates and Euler angles, to various sensor errors and produces the measured outputs. The "INS Computations" block subsequently uses the sensor outputs to compute inertial acceleration.

These two blocks will be explained in more detail in the remainder of this section.

1. Strapdown IMU

The "strapdown IMU" depicted in Figure 3.13 is comprised of four subgroups — "Compute thrust acceleration", "Rate Gyros", "Accelerometers", and "Inclinometers". Modeling of the rate gyros and accelerometers include all of the error sources — bias, noise floor, and cross-axis sensitivity — defined in Section E. of Chapter II. Inclinometer outputs are corrupted by coupling with linear acceleration. Since Kuechenmeister developed the IMU model in [Ref. 16], this portion of the model will not be further discussed or verified.

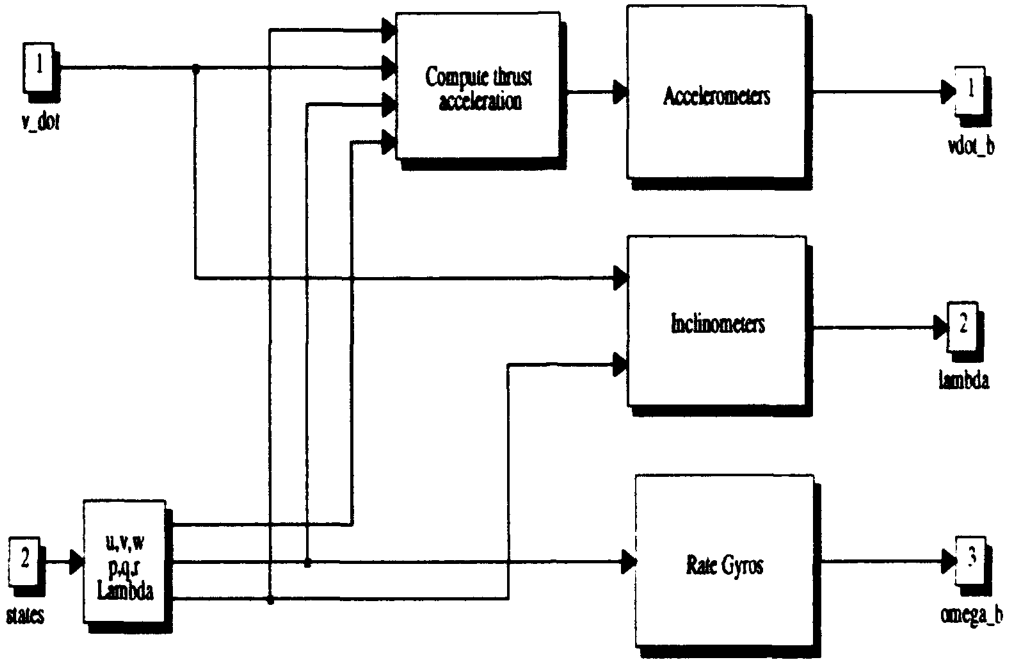
1
v_dot
Compute thrust acceleration
Accelerometers
1
vdot_b
Inclinometers
2
lambda
Rate Gyros
3
omega_b
2
states
u,v,w
p,q,r
Lambda

Figure 3.13: Strapdown IMU Model

2. INS Computations

The "INS Computations" block converts the sensor outputs from the IMU into inertial acceleration. The structure of this block is represented in Figure 3.14. Inertial body acceleration in the body frame is computed by the "compute a_b" block and provided to the "convert a_b to a_u" block for transformation to the inertial frame. The "Estimate lambda" block provides an optimal estimate of the Euler angles based on its inputs, "lambda" (from the inclinometers) and "omega_b" from the rate gyros. These estimated Euler angles are used by the "convert a_b to a_u" block to compute the rotation matrices for the transformation.

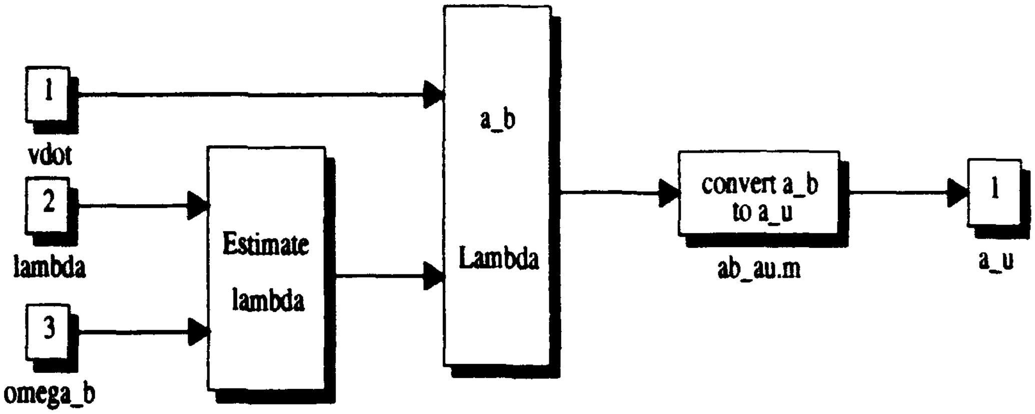

Figure 3.14: INS Computations Model

Before transforming the acceleration from the body to the inertial frame, the model must compute the Euler angles. This task is performed by the "Estimate lambda" block shown in Figure 3.15. The top half of this figure provides measured Euler angles and calculated Euler rates for the bottom half of the figure, the complementary Kalman filters, to estimate lambda. The block labeled "S(lambda) * omega" computes Euler rates from Poisson's equation (Equation 2.33) as previously discussed. The "rearrange into rate/angle pairs" block pairs each Euler angle with its derivative. Each pair then enters its own filter — "Roll filter", "Pitch filter", and "Yaw filter" — which provides an estimate of each respective angle. The result is fed back to be used in the calculation of the Euler rates as shown.

Finally, with "a_b" and "Lambda" computed, the "convert a_b to a_u" block in Figure 3.14 can calculate inertial acceleration, "a_u", through the transformation defined in Equation 2.34.

D. INS MODEL VERIFICATION

As previously stated, the IMU Model will not be verified. However, the remainder of the INS, the INS Computations portion, must be verified. Since this portion of the INS is purely computational, no particular flight regime need be tested. The circumstances of the verification simulation are:

- Bluebird aircraft flying at sea level
- elevator fixed at 0°
- lateral controls fixed at 0°
- trim thrust applied

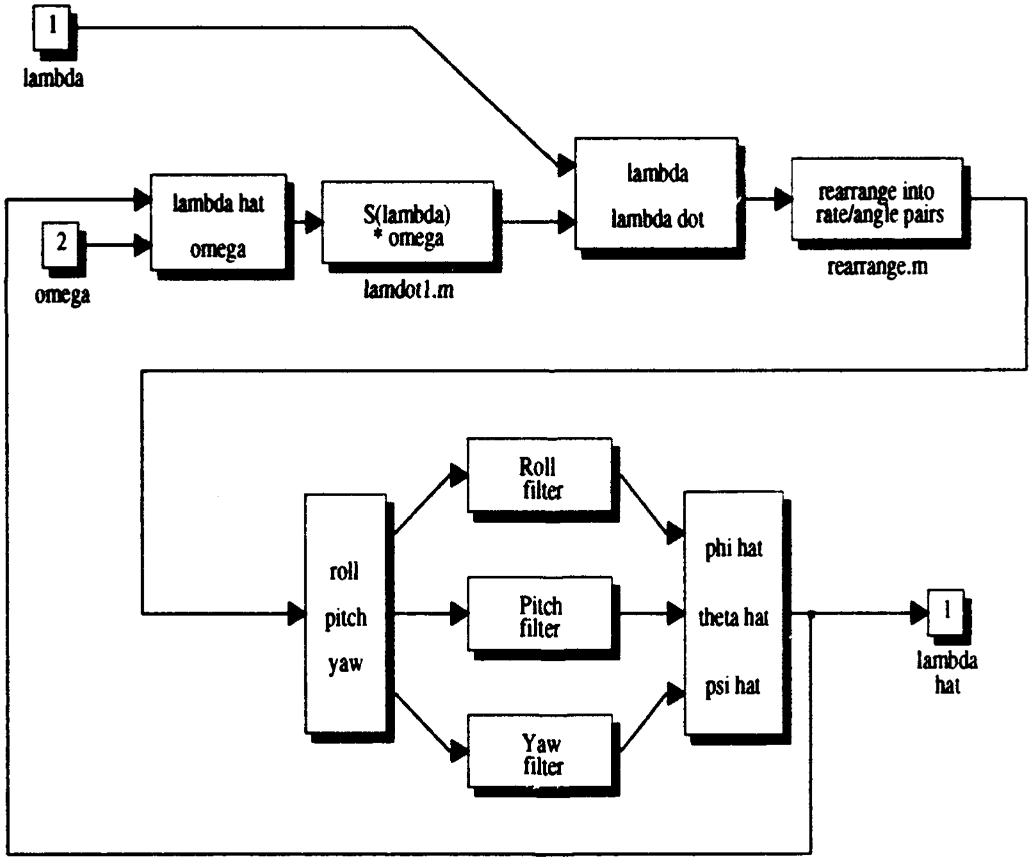
1
lambda
2
omega
lambda hat
omega
S(lambda) * omega
lamdot1.m
lambda
lambda dot
rearrange into rate/angle pairs
rearrange.m
roll
pitch
yaw
Roll filter
Pitch filter
Yaw filter
phi hat
theta hat
psi hat
1
lambda hat

Figure 3.15: Euler angles estimator

There are two sets of data to be compared in this verification. The "INS Computations" block computes both inertial acceleration and Euler angles. By comparing these quantities with the correct values calculated from the uncorrupted kinematic quantities, one can evaluate the accuracy of the model. Thus, the remainder of this section will consist of comparing exact and calculated values of inertial acceleration and Euler angles.

1. Euler Angles

Figure 3.16 depicts the difference between the exact roll angle (which remained zero) and the roll angle calculated by the INS. At first glance, the difference appears to be substantial. However, when one realizes that the maximum difference between the calculated and estimated roll angle is a mere 0.42°, one recognizes that there is really very little error. This small error is due primarily to cross-axis sensitivity in the rate gyros. This phenomenon is causing a small difference between the

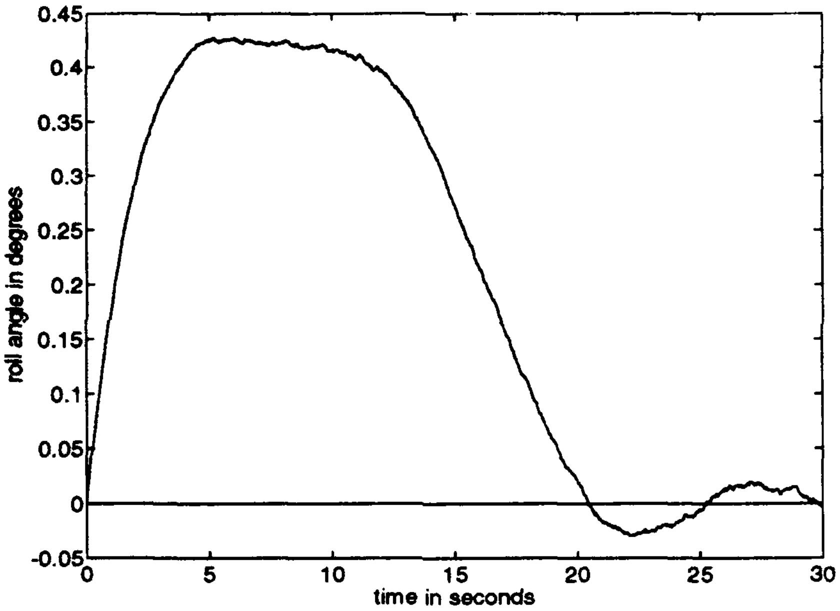

Figure 3.16: Actual and estimated roll angles

measured and actual roll rate. Furthermore, as the phugoid motion damps out, all angular rates will tend to zero. Therefore, the cross-axis error will tend to zero. This is certainly an effective estimator in the roll or X-axis. The estimates of the yaw angle behaved similarly and were also quite accurate, with the maximum error only 0.6°. The error is again a result of cross-axis sensitivity.

Pitch angles estimated by the complementary filter were also quite accurate (see Figure 3.17). The maximum error between the estimated pitch angle and the actual pitch angle is only 1°. Furthermore, as can be seen from the figure, the error is decreasing toward zero. The minimum error is driven by the noise in the IMU. Since the angular estimator relies on the virtually noise-free inclinometer at steady state, the error in the estimate at steady state would certainly be nearly zero.

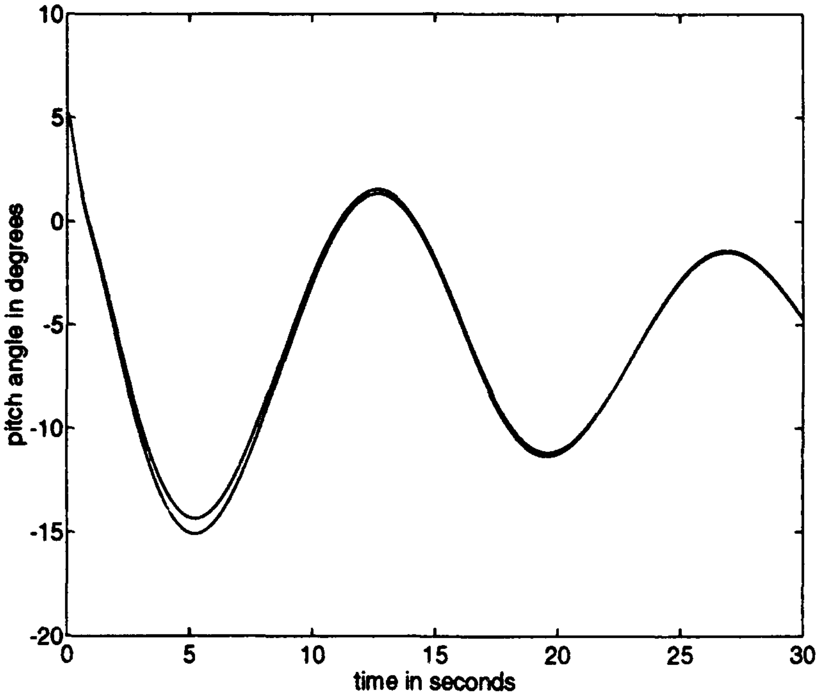

Figure 3.17: Actual and estimated pitch angles

2. Linear Acceleration

The inertial acceleration calculation is quite accurate, just as the Euler angle estimation scheme is. Since the accelerometers and rate gyros are subject to nearly identical types of errors, it is not surprising that the acceleration errors would be similar in nature to the angular errors. For example, Figure 3.18 shows actual and calculated inertial X acceleration. Once again, the values are quite close, with the maximum deviation being around 0.5 ft/s^2. Moreover, the error appears to be decreasing as the phugoid motion damps out. The error, however, will *not* go to zero. Because the Z accelerometer will continue to sense thrust acceleration in steady, level flight, the X measured acceleration will have a small steady state error due to cross-axis coupling. Furthermore, the "fuzziness", caused by noise in the accelerometer measurement of thrust acceleration, clearly identifies the calculated

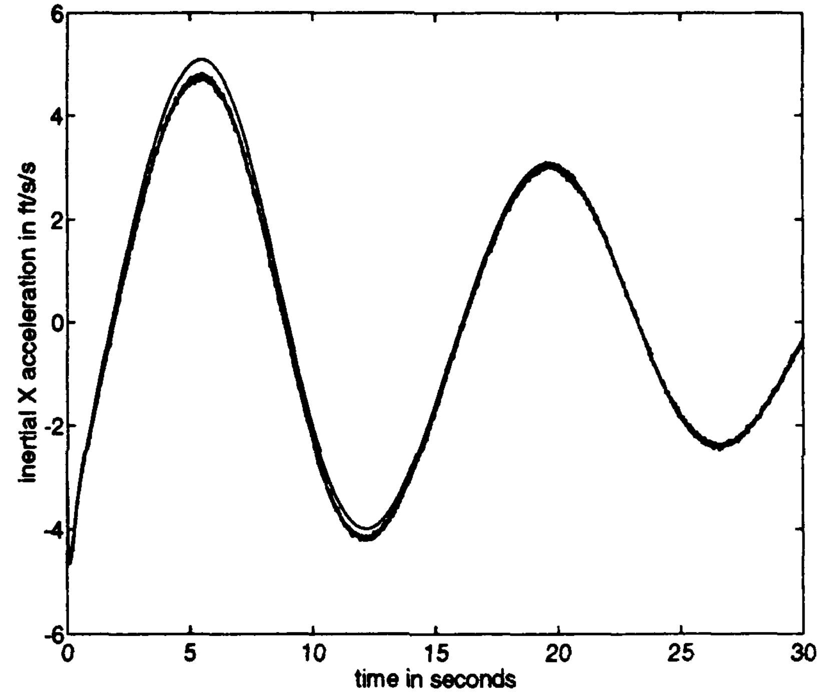

Figure 3.18: Actual and calculated X acceleration

curve. As expected, this high frequency component remains constant over the entire curve.

Figure 3.19 more clearly exposes the effects of cross-coupling and noise in the accelerometers. Although the aircraft never actually accelerates in the Y direction, cross-coupling with both the X and Z accelerometers causes the general downward swing of the calculated curve. While this curve also appears to be quite noisy, the standard deviation of the noise is actually only a fraction of a ft/s^2. Like the X measured acceleration, the error will tend to a small, non-zero value because of cross-coupling with the Z accelerometer. Even so, the calculation of the inertial acceleration in the Y axis is satisfactory.

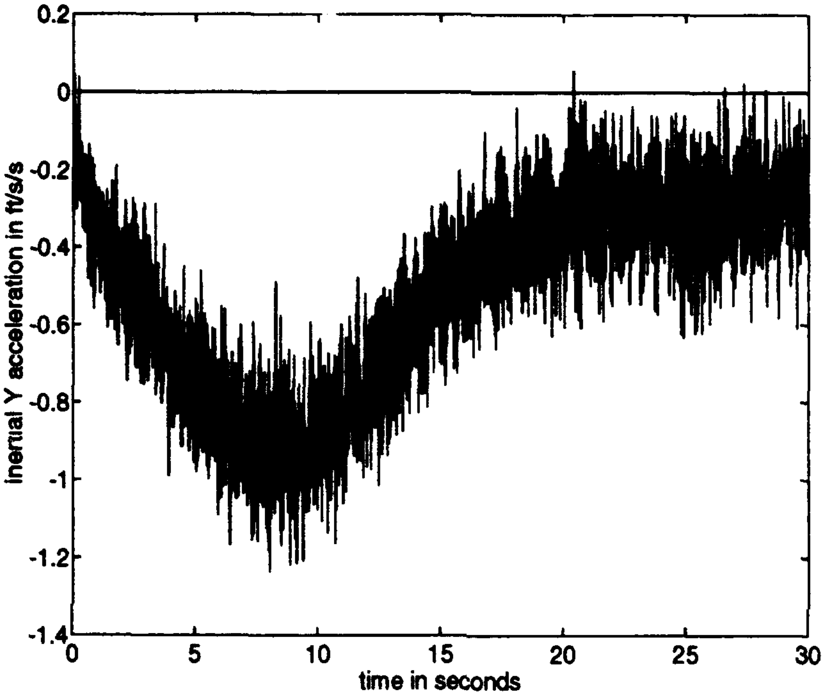

Figure 3.19: Actual and calculated Y acceleration

Finally, Figure 3.20 shows the accuracy of the inertial acceleration calculation in the vertical or Z direction. This acceleration error does not appear to be

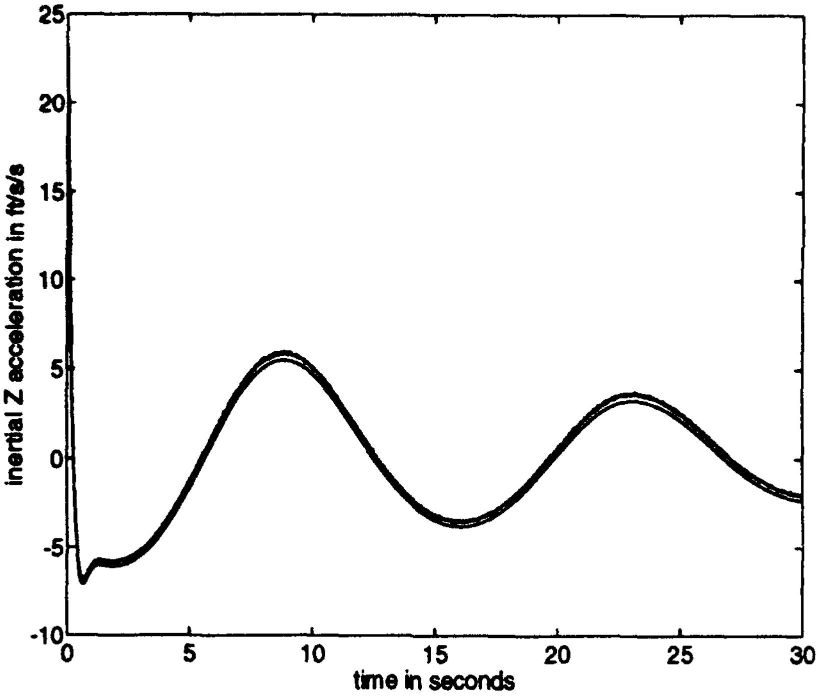

Figure 3.20: Actual and calculated Z acceleration

converging to zero as do the accelerations in the other axes. This lack of convergence is made more clear by Figure 3.21. This effect is due to the fact that, unlike the other accelerometers, the Z accelerometer will not measure at or even near zero acceleration in steady, level flight. In fact, the one 'G' of acceleration that the Z accelerometer senses in steady, level flight causes small errors in the other two axes due to cross-axis sensitivity.

E. NAVIGATION (LOOKING AHEAD)

The simulations conducted in the previous sections clearly indicate which errors dominate Differential GPS and INS. Certainly the most troublesome DGPS errors are caused by clock differences. In fact, after only ten minutes of simulation time, the clocks *alone* had driven the pseudorange errors for some of the satellites to two miles.

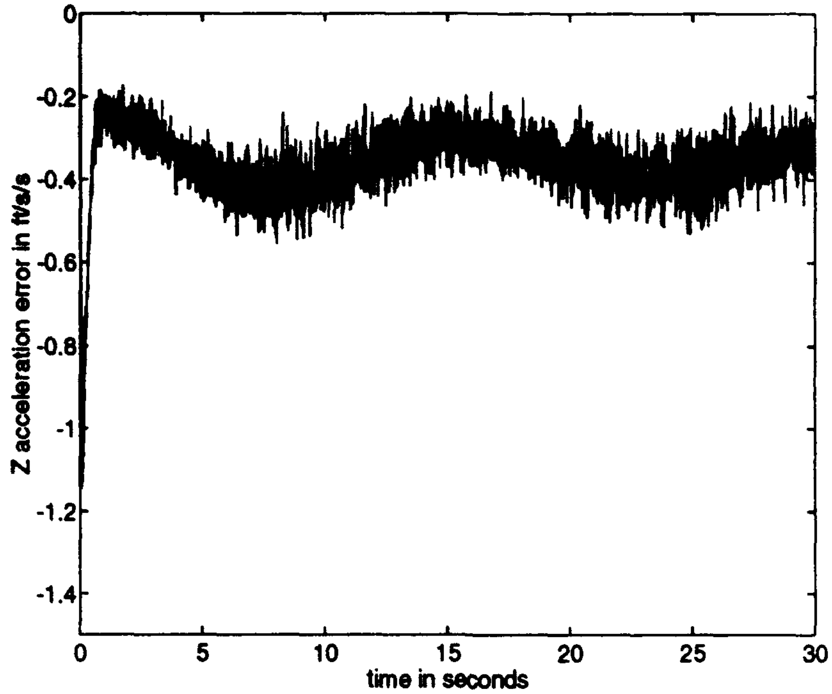

Figure 3.21: Z acceleration error

Simulating the system without the clock errors showed that the system's accuracy would be vastly improved if the clock errors could be eliminated. Fortunately, the INS proved to be quite accurate. Thus, no INS errors need to considered in the Kalman filter design. Therefore, the Kalman filter design for navigation (see next chapter) must include a clock model of the difference between the aircraft receiver clock and the differential station clock.

IV. DGPS/INS INTEGRATION

Having completely defined, encoded, and tested models of Differential GPS and INS, the issue of integration can finally be addressed. As mentioned early in this work, the engineering device which accomplishes the task of fusing the outputs of the INS (inertial acceleration) and the DGPS (six pseudoranges) to produce inertial position is the complementary Kalman filter. A description of the general design process for a complementary Kalman filter follows.

A. GENERAL LINEAR KALMAN FILTER DESIGN

Before detailing the Kalman filter design process, it is critical to define the framework in which it operates. The mechanism which defines this process is called the *synthesis model.* The *synthesis model* models the process whose states the Kalman filter is designed to estimate. In general, this model is non-linear. Since this is a linear design process, one must first linearize the synthesis model to produce a standard linear system which can be represented by a state space

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= Cx + Du + v, \end{aligned} \tag{4.1}$$

where x is the state vector, z is the measured output vector, w is the process noise vector. u is the input vector, v is the sensor noise vector, and A, B_1, B_2, and C are the standard matrices defining a linear, dynamic system. The feed-forward matrix, D, is assumed to be zero. Complete explanations of all linear systems concepts can be found in any basic control text, such as [Ref. 17, Ch. 2]. Typically, w and v are specified as zero-mean, white, Gaussian noise vectors with covariances R_w and R_v, respectively.

Most often, a Kalman estimator is used in estimating states which cannot be directly measured, although they must be observable. For example, suppose the measured output of the system defined in Equations 4.1, z, is a scalar which is the sum of all of the system's states (plus measurement noise). A Kalman estimator could take u, the system input vector, and z, the measured system output, and produce an optimal estimate of each state. The estimator is constructed as follows

$$\dot{\hat{x}} = A\hat{x} + B_2u + H(z - C\hat{x}), \tag{4.2}$$

where $\hat{x}$ is the estimator state vector, H is the Kalman gain, and the remaining variables represent the same quantities as in Equation 4.1. Rewriting Equation 4.2 by collecting $\hat{x}$'s yields

$$\dot{\hat{x}} = (A - HC)\hat{x} + B_2u + Hz. \tag{4.3}$$

This form shows that the stability of the estimator is determined by the poles of its state matrix, $A - HC$. If this matrix has all stable poles, the estimates are stable. But, will they approach the actual states, x, that they are supposed to estimate? By defining the error, $\tilde{x}$, as the difference between the estimator states and the actual states and subtracting Equation 4.2 from the first of Equations 4.1, one finds

$$\dot{\tilde{x}} = (A - HC)\tilde{x} + B_1w + Hv. \tag{4.4}$$

This relationship demonstrates that estimator stability alone results in the error converging to zero, if there is no process or sensor noise. The filter cannot be designed to remove the error caused by these disturbances. Thus, the accuracy of the Kalman filter is limited by the two noise vectors. To complete the design, the Kalman gain must be calculated.

The Kalman gain matrix is computed by solving two equations in sequence — the algebraic Riccatti equation and the gain equation. The algebraic Riccatti equation solution, P_∞, is the steady state covariance matrix of the error state, $\tilde{x}$, provided the H is computed using Equation 4.6. It is (see, for example [Ref. 15, p. 132])

$$AP_\infty + P_\infty A^T - P_\infty C^T R_v^{-1} C P_\infty + B_1 R_w B_1{}^T = 0. \tag{4.5}$$

Having solved this equation for P_∞, one can use this result to compute the Kalman gain with the following equation

$$H = -P_\infty C^T R_v^{-1}. \tag{4.6}$$

By inserting this gain into Equation 4.3, the Kalman filter is constructed. The remaining issue to be resolved is the specification of the covariances, R_w and R_v. It is the choice of these two matrices that determines the characteristics of the filter. The effects of varying these covariances are discussed in the following paragraph.

Increasing the sensor noise covariance, R_v, results in decreased gain, as one can see from Equation 4.6. Since the gain is a measure of the emphasis given to the measurement, z, this makes sense. That is, a noisier measurement should be deemphasized by the filter. Conversely, decreasing the measurement noise covariance will increase the gain. In the limiting case, as the sensor noise covariance approaches infinity, the filter will be identical to the dynamic system whose states it is designed to estimate. Since the Kalman gain would be zero, the measurements, z, would be ignored entirely. Likewise, if the sensor noise covariance becomes very small, the gain would be very large and the system would essentially invert the measured output to find the states.

The concept of "emphasis" introduced in the previous paragraph is more commonly referred to as bandwidth. Increasing the "emphasis" of the sensor measurement by decreasing the sensor noise covariance is equivalent to increasing the bandwidth of the estimator with respect to that input. This concept forms the basis for the design of a complementary Kalman filter. Strictly speaking, the process noise and sensor noise covariances must be measured characteristics of the noise inputs to the system. Realistically, this measurement procedure is impractical and unnecessary. By using the bandwidth concept, one can tailor the estimator by selecting the covariance matrices to produce the the desired frequency characteristics.

The procedure for designing a complementary, linear Kalman filter is:

- construct the synthesis model
- linearize the synthesis model (if necessary)
- set both R_w and R_v to the identity matrix of the appropriate dimension
- compute the Kalman gain
- construct the complementary filter using Equation 4.3
- determine the sensor bandwidths using Bode analysis
- adjust R_v to increase or decrease the bandwidths as required and iterate

This procedure was used to develop the complementary linear Kalman estimator, which integrates inertial acceleration and six DGPS pseudoranges to yield inertial position.

B. DESIGN OF THE COMPLEMENTARY, LINEAR, KALMAN FILTER FOR DGPS/INS INTEGRATION

Before describing the exact design process, the goal of the process must be established. The final filter should rely on the DGPS pseudorange errors at low frequencies, below 0.5 Hz or about three radians per second. At these low frequencies, the filter should essentially invert the six pseudorange errors to find position. At higher frequencies, the filter should rely on the inertial accelerations for positioning; by integrating these accelerations twice, the filter will calculate position. Based on these considerations, the Bode plot of the transfer function from pseudorange error as input to corresponding pseudorange error estimate as output should look like a low pass filter. Conversely, the frequency response from the accelerometer inputs to pseudorange error estimates should be similar to a high pass filter.

As delineated in the previous section, the first step in the design process is the construction of the synthesis model. It is shown in Figure 4.1. The synthesis model defines the process whose states the Kalman filter will estimate. In this case, the desired output of the filter is inertial position — P_x, P_y, and P_z in the synthesis model. Since each of the scalar elements of the position vector are states of the synthesis model, they can be estimated by the filter.

The basic construction of the synthesis model is quite simple. Each of the scalar components of the inertial acceleration vector, a_x, a_y, and a_z, are inputs to the model. These accelerations are integrated twice to yield the inertial positions, P_x, P_y, and P_z. Three other inputs to the upper part of this diagram are w1, w2, and w3. These are three scalar, independent, white, zero–mean, Gaussian noise inputs. Integrating them before adding them to the accelerations as sensor noise forces the filter to compensate for an accelerometer bias.

The bottom of the model defines the unstable clock dynamics. Note that this is identical to the previously defined clock model in Figure 2.6. Once again, the purpose of this portion of the synthesis model is to alert the design process to the dynamics of the clock errors such that it can more effectively account for them. As before, the clock difference must be multiplied by the speed of light to convert it from a time error to a distance error.

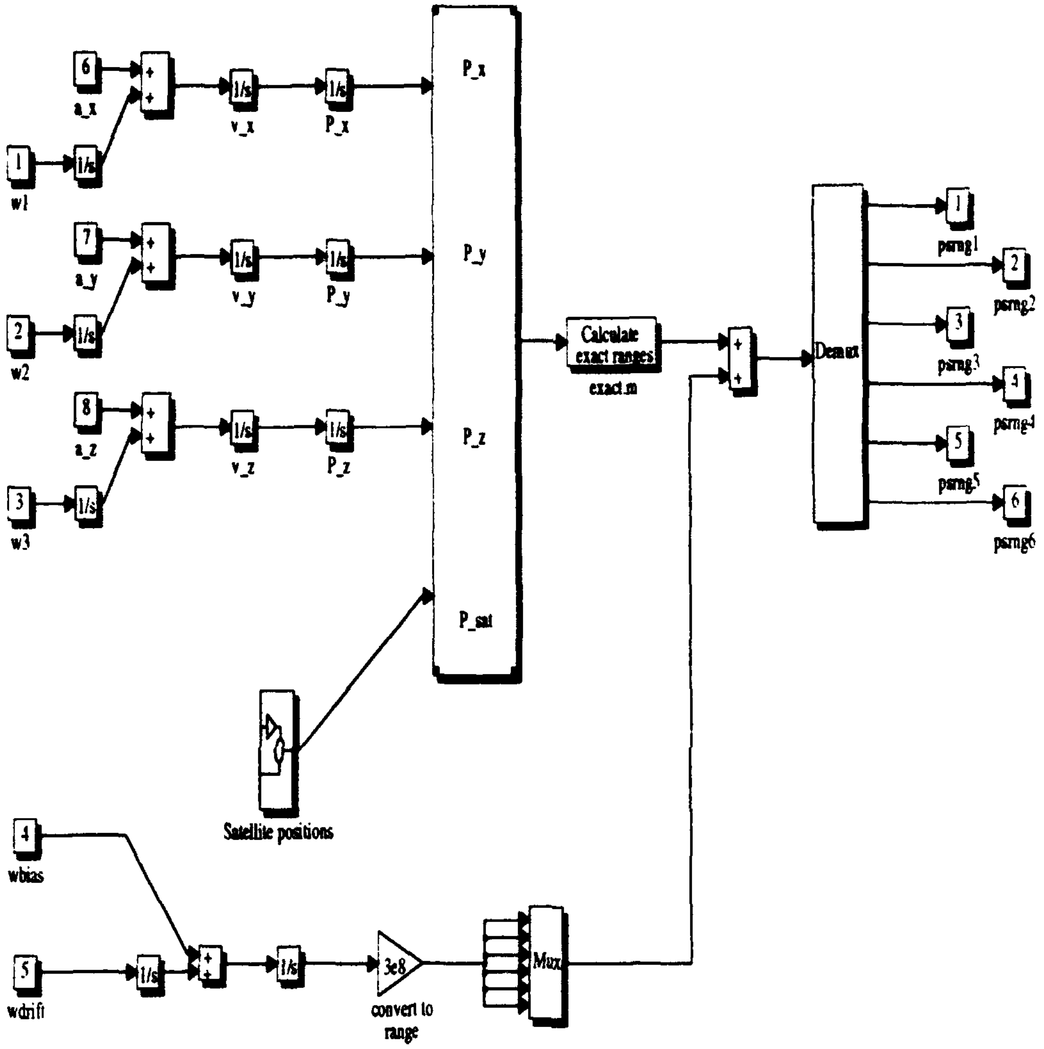
6
a_x
1
w1
1/s
+
+
1/s
v_x
1/s
P_x
7
a_y
2
w2
1/s
v_y
1/s
P_y
8
a_z
3
w3
1/s
v_z
1/s
P_z
P_x
P_y
P_z
P_sat
Satellite positions
Calculate exact ranges
exact.m
Demux
1
psrng1
2
psrng2
3
psrng3
4
psrng4
5
psrng5
6
psrng6
4
wbias
5
wdrift
1/s
1/s
3e8
convert to range
Mux

Figure 4.1: DGPS/INS Integration Synthesis Model

The synthesis model does not contain two separate clock models whose difference is applied to the exact ranges to create pseudoranges. The reason for this is quite simple — only six independent integrators can be observable with six measured outputs. Including two complete clock models as well as three independent process noise inputs would result in seven independent noise inputs. Therefore, the model had to be simplified to include only five independent noise inputs in order for all of them to be observable.

The "Compute exact ranges" block uses the Pythagorean theorem to compute the exact ranges to the six GPS satellites. The clock difference error, now converted to distance, is added to each exact range to create six pseudoranges, the measured outputs, z, of the synthesis model.

Since the use of the Pythagorean theorem causes the problem to be non-linear, the synthesis model must be linearized at a given space vehicle constellation and aircraft position. The linearization process yields a state space with 11 states, five process noise inputs, three control inputs, and six measured outputs. Due to the complexity of this state space model, the complete model is relegated to Appendix A.

The designer can now determine the Kalman gain. Unlike the general case, the process noise covariance matrix is not identity in this case. Since the clock noise variances are so small and measurable, they must be specified. If the designer uses unity variance instead of these miniscule values, the resulting filter will not perform satisfactorily. The process noise covariance matrix used in this design is

$$R_w = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 \times 10^{-19} & 0 \\ 0 & 0 & 0 & 0 & 1.58 \times 10^{-18} \end{bmatrix} . \qquad . \tag{4.7}$$

For the first iteration, the sensor noise covariance, R_v, is set to a 6 by 6 identity matrix ($I \in R^6$). The Kalman gain is now calculated by first solving the algebraic Riccatti equation (Equation 4.5) and then computing the gain (Equation 4.6).

The state space of the filter is now calculated according to Equation 4.3 with computed Kalman gain, H. The frequency response of the filter from measured pseudorange error, $\Delta\rho$, to the corresponding estimated pseudoranges error, $\Delta\hat{\rho}$, can now be evaluated. In evaluating the bandwidth of each of the pseudorange inputs, one must consider the relative bandwidth. Since none of the zero frequency gains are unity, the bandwidth is three decibels less than the steady state gain. The state space of the system whose frequency characteristics must be investigated is

$$\begin{aligned} \dot{\hat{x}} &= (A - HC)\hat{x} + H\Delta\rho \\ \Delta\hat{\rho} &= C\hat{x}. \end{aligned} \tag{4.8}$$

After several iterations, the design was finalized with the following sensor noise covariance matrix

$$R_v = \begin{bmatrix} 0.11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.28 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.11 \end{bmatrix}. \tag{4.9}$$

Bode plots from each sensor input to its corresponding estimate are shown in Figures 4.2 through 4.4. In all of these plots, the "dash-dot" line represents the 3 dB less than the DC gain. Thus, the intersection of the two lines occurs at the bandwidth.

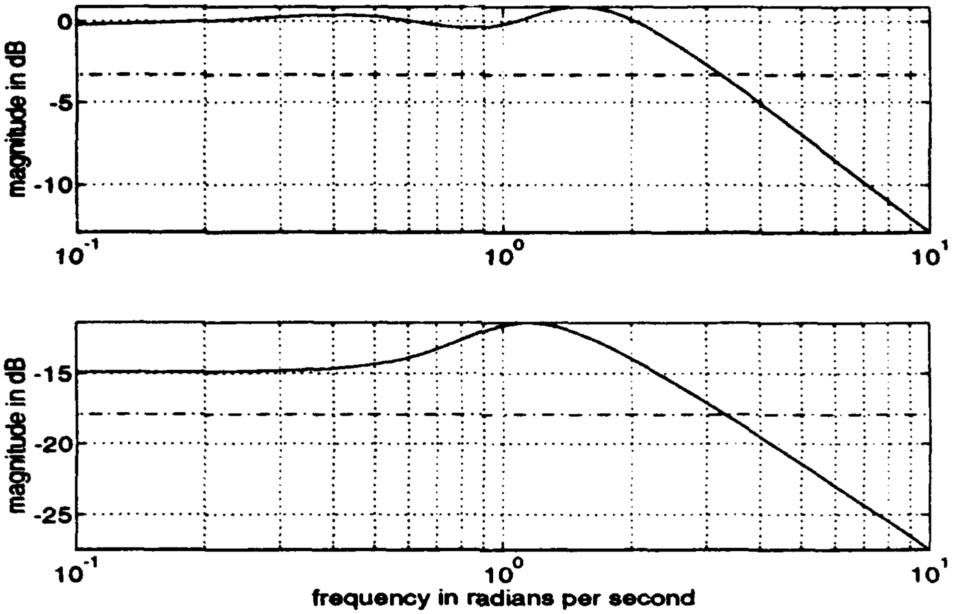

Figure 4.2: Frequency response from ρ_1 to $\hat{\rho}_1$ and ρ_2 to $\hat{\rho}_2$

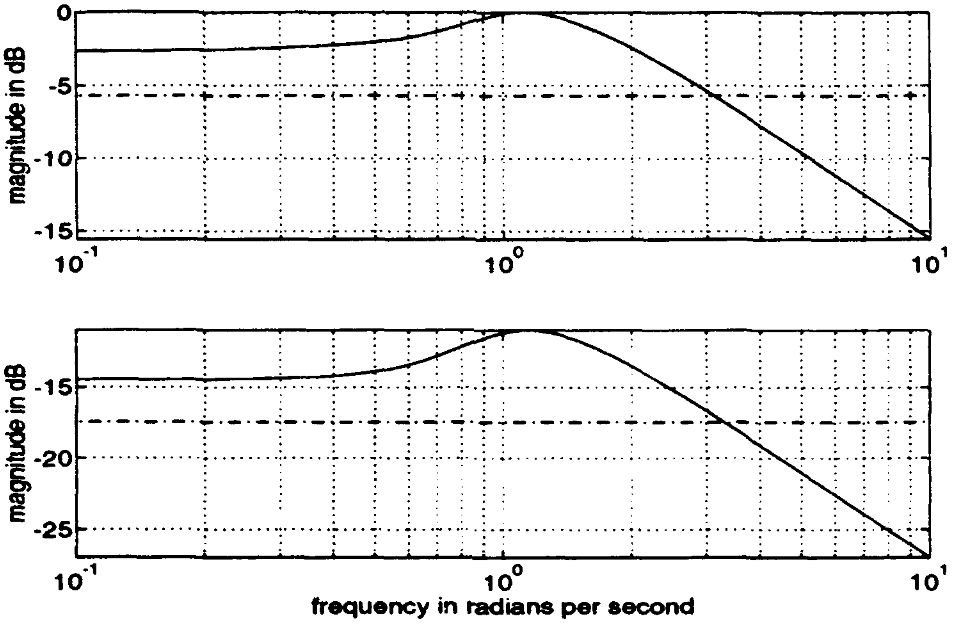

Figure 4.3: Frequency response from ρ_3 to $\hat{\rho}_3$ and ρ_4 to $\hat{\rho}_4$

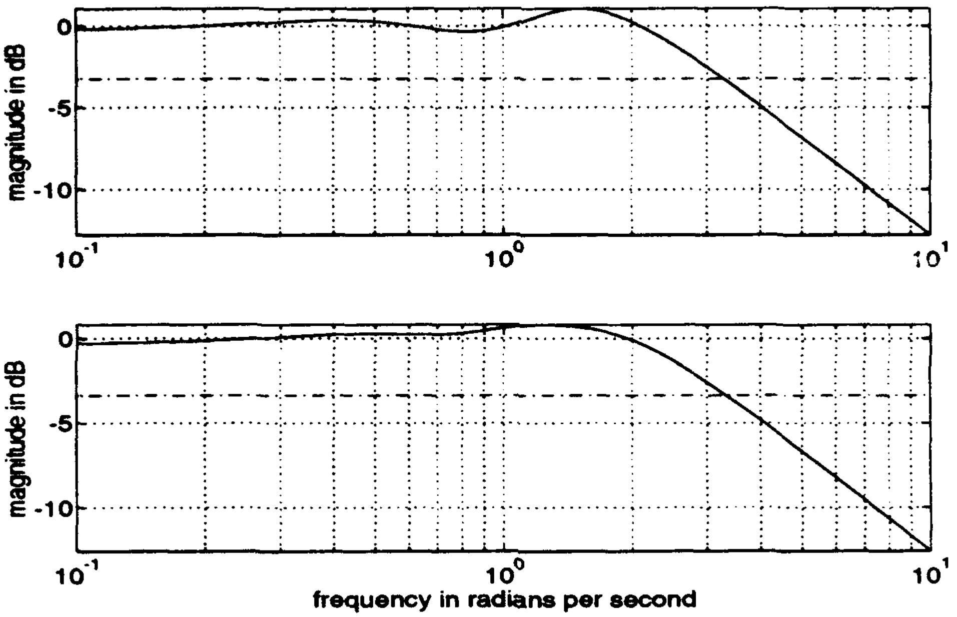

Figure 4.4: Frequency response from ρ_5 to $\hat{\rho}_5$ and ρ_6 to $\hat{\rho}_6$

From the Bode plots, one can see that all sensor bandwidths are near three radians per second, as shown in Table 4.1.

TABLE 4.1: SENSOR BANDWIDTHS FOR DGPS/INS KALMAN FILTER

	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6
3 dB bandwidth	3.10	3.15	3.00	3.15	3.10	3.15

It is also worthwhile to note that, as previously mentioned, the frequency response of the transfer function from the accelerometers to the pseudorange estimates has negligible gain at low frequencies. A few of these are shown in Figures 4.5 through 4.7.

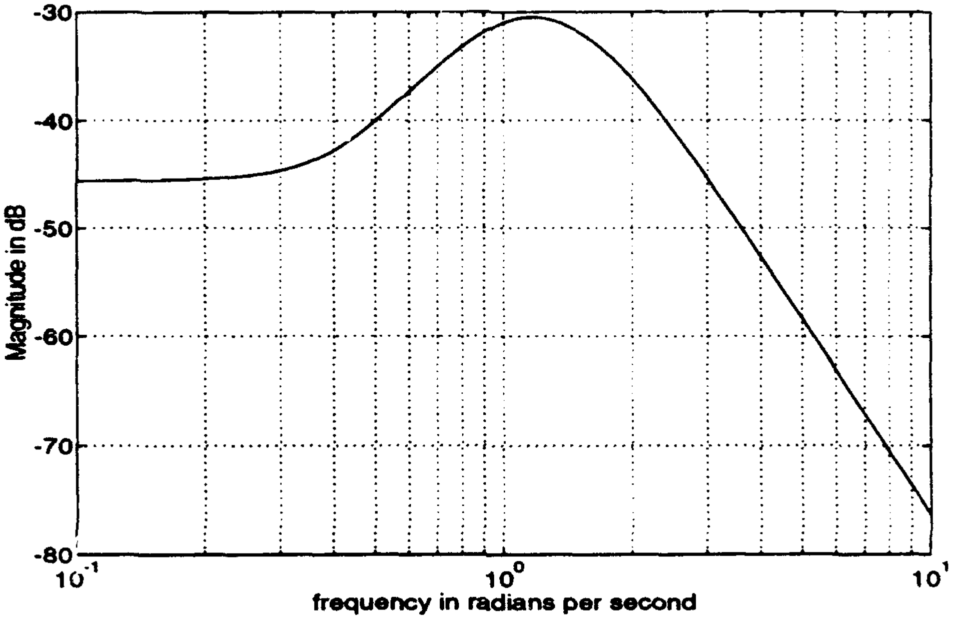

Figure 4.5: Frequency response from ${}^{U}a_x$ to $\hat{\rho}_3$

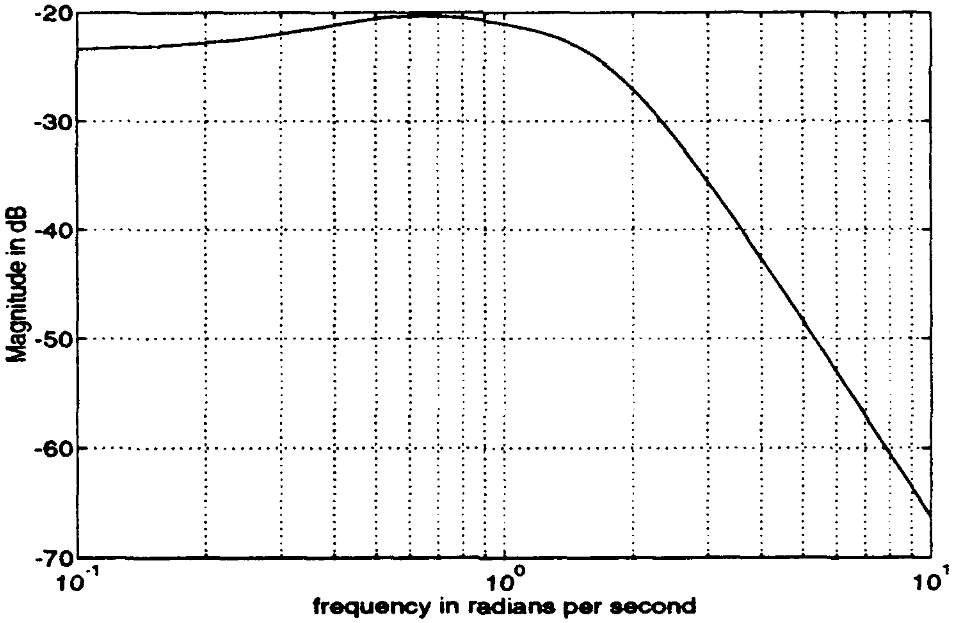

Figure 4.6: Frequency response from ${}^{U}a_z$ to $\hat{\rho}_1$

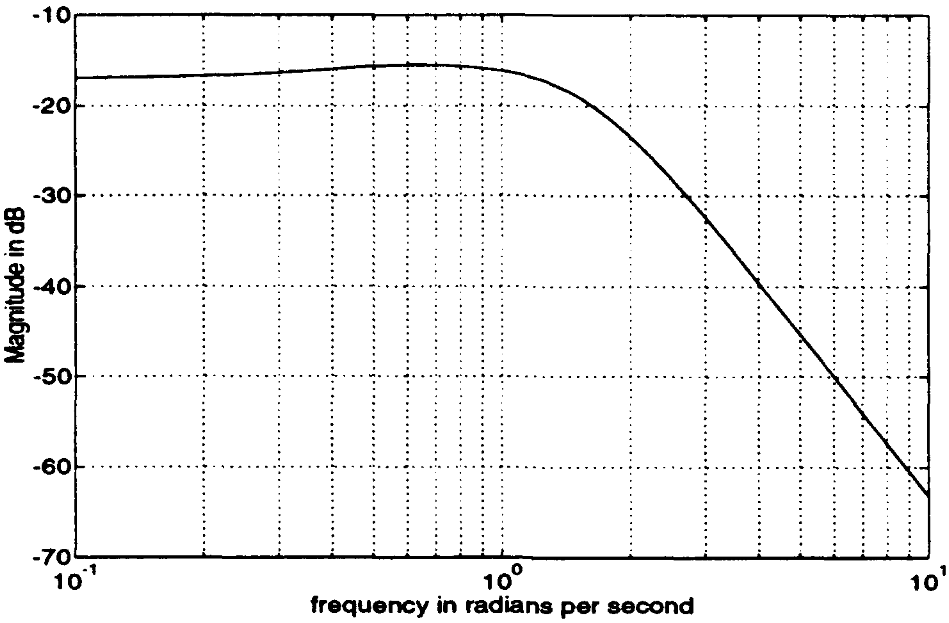

Figure 4.7: Frequency response from $^{U}a_z$ to $\hat{\rho}_5$

Once the the frequency characteristics sought at the outset of the process are achieved, the filter can be tested in simulation.

C. FILTER VERIFICATION

The SIMULINK representation of the simulation used to verify the filter is depicted in Figure 4.8. The "Linear filter" block accepts nine inputs — three components of inertial acceleration, "a_u", and six pseudorange errors, "psrng" — and provides estimates of three dimensional position. It is important to note that this filter, since it is linear, must be driven with pseudorange errors, rather than pseudoranges, themselves.

In order to show that this navigation system is functioning, it was simulated both with and without the accelerometer inputs. Since in normal, steady-state, trimmed flight, the accelerometer inputs are ignored (*i.e.*, "washed-out"), it is necessary to include some high frequency dynamics. By using a five degree elevator

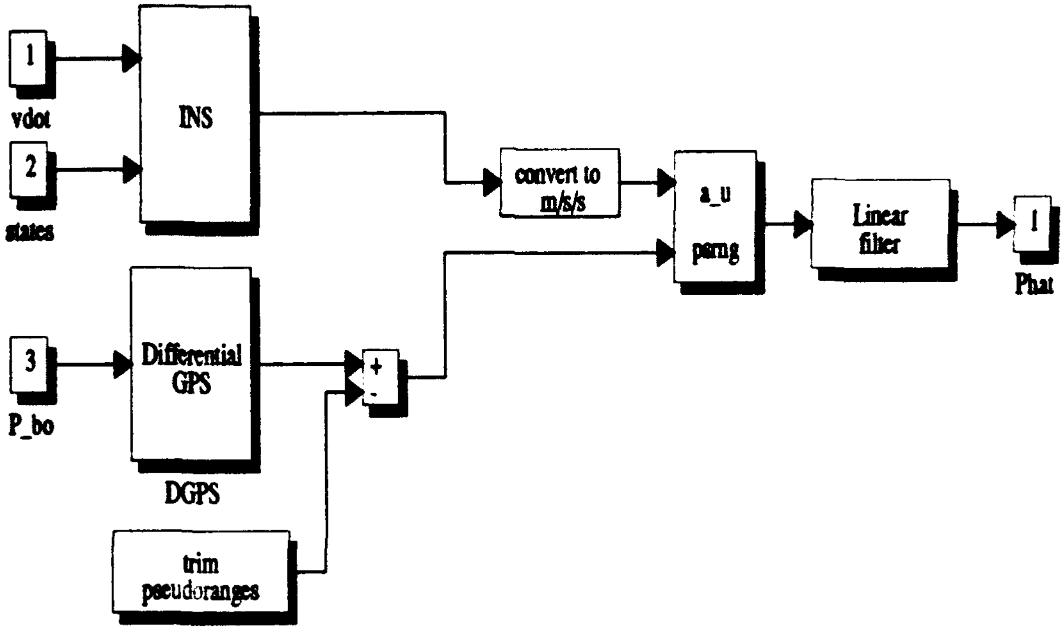

Figure 4.8: Integrated Navigation System

half-doublet (zero degree elevator from zero to ten seconds, five degree nose-down elevator from ten to 20 seconds, and zero degree elevator from 20 to 30 seconds), sufficiently high frequency dynamics are excited enabling testing of the complementary Kalman filter.

Due to computer limitations, the navigation model was not be tested over a long period of time. The computers being used to perform the simulation did not have sufficient speed or memory to run the simulation for ten minutes of real time, as was done in verifying the DGPS Model. This shortcoming prevented a detailed study of the individual effects of the INS and DGPS on the position output. The goal of this verification was to show that both sensors effect the output. This was shown by demonstrating a difference between the position errors with and without the accelerometer inputs. One would expect the position to be less accurate with the accelerometer inputs since they are so much more corrupted by the various errors.

Both simulations were performed with the following conditions:

- Bluebird aircraft flying at sea level
- lateral controls fixed at 0°
- trim thrust applied
- 5° elevator half-doublet applied

1. Verification without Accelerometers

The position estimates from the simulation without accelerometers are fairly accurate. This system navigates entirely by the differential GPS. The following three plots, Figures 4.9 through 4.11, show the difference between the estimated and actual position in each direction of the tangent plane system. The estimated position does, in fact, track the actual position quite accurately in all three directions. This proves that the Kalman filter is capable of effectively inverting the six pseudoranges to provide inertial tangent plane position without the assistance of the INS.

2. Verification with Accelerometers

Having shown the performance of the system with DGPS only, one can compare the filter's performance without accelerometers to its performance with accelerometers. Figures 4.12 through 4.14 show the displacement errors in all three directions of the inertial tangent plane system.

Once again, it is clear that the position estimate tracks the actual position. In fact, it is quite difficult to see a great deal of difference in the plots representing navigation with and without accelerometer inputs. However, reducing the data to means and standard deviations for each simulation reveals differences in accuracies. Table 4.2 summarizes the difference between the two simulations where μ is the mean and σ is the standard deviation. As forecast, the accelerometer augmented

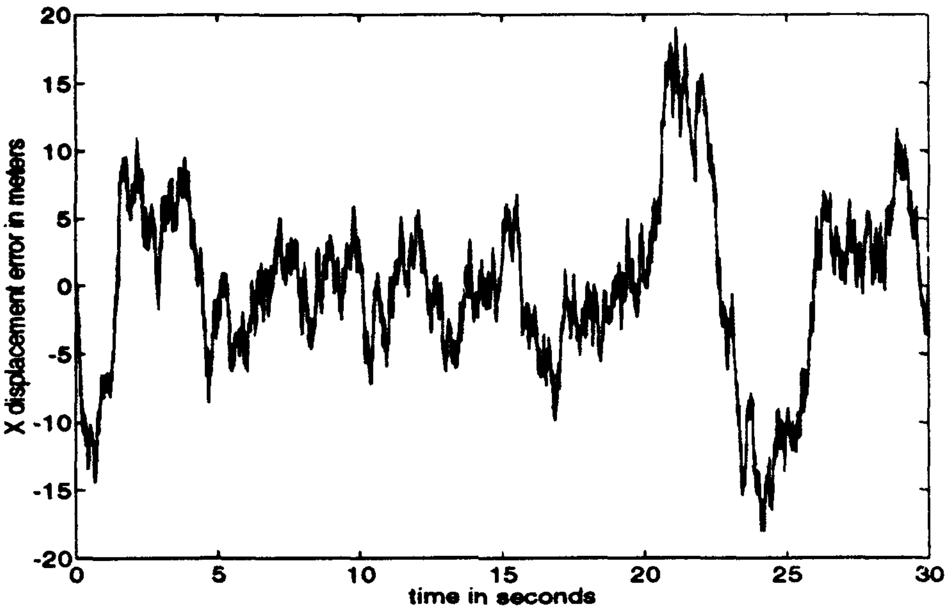

Figure 4.9: X displacement error without accelerometers

TABLE 4.2: MEAN AND STANDARD DEVIATION OF X, Y, AND Z ESTIMATED POSITION ERRORS WITH AND WITHOUT ACCELEROMETERS

meters	X error	Y error	Z error
μ w/o accel	0.0200	-0.4481	0.0607
σ w/o accel	6.323	5.194	14.84
μ with accel	1.679	0.4093	2.0084
σ with accel	6.594	5.174	14.86

positioning data demonstrated a slightly greater standard deviation than the pure DGPS positioning. Furthermore, it has a significant, non-zero mean error, while the system which did not use the accelerometers did not.

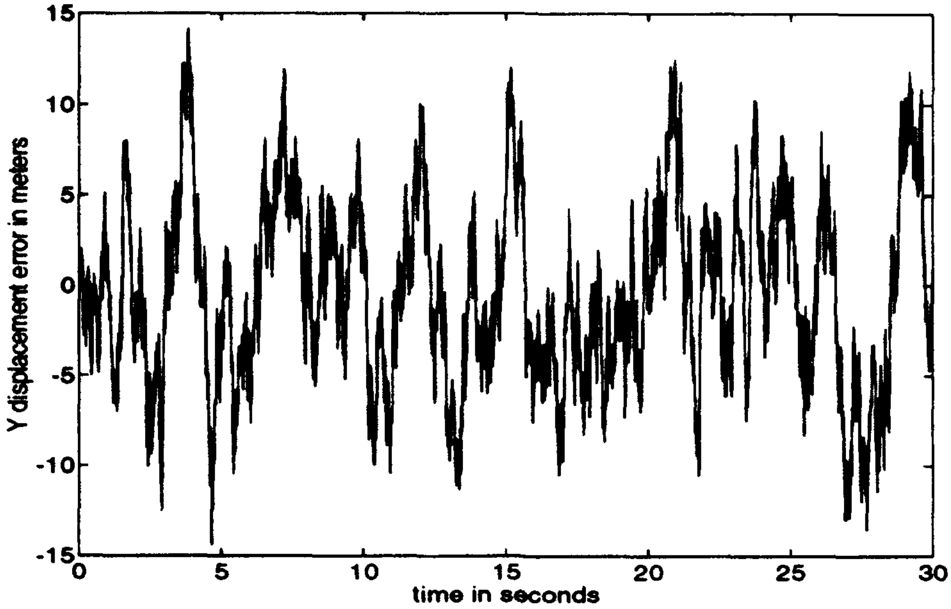

Figure 4.10: Y displacement error without accelerometers

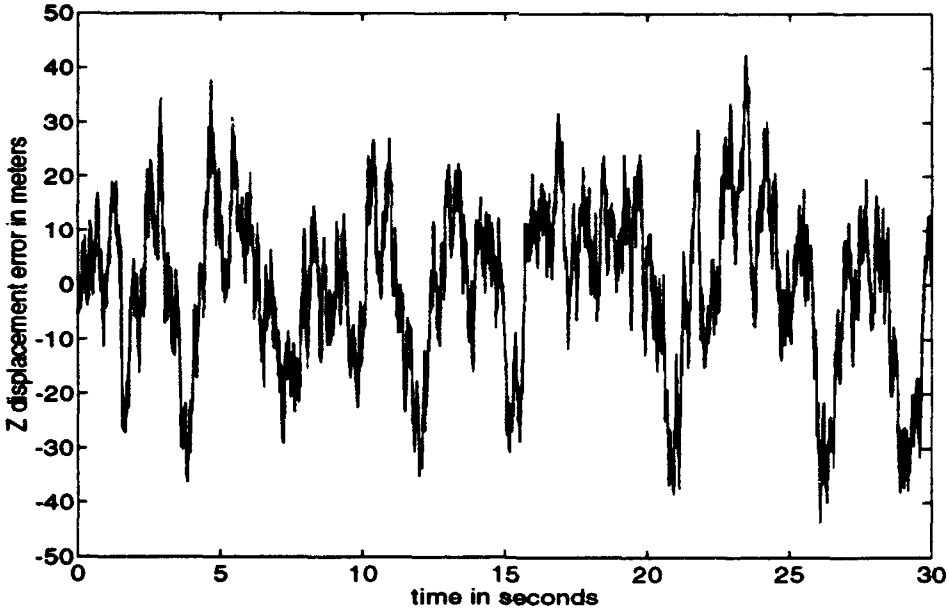

Figure 4.11: Z displacement error without accelerometers

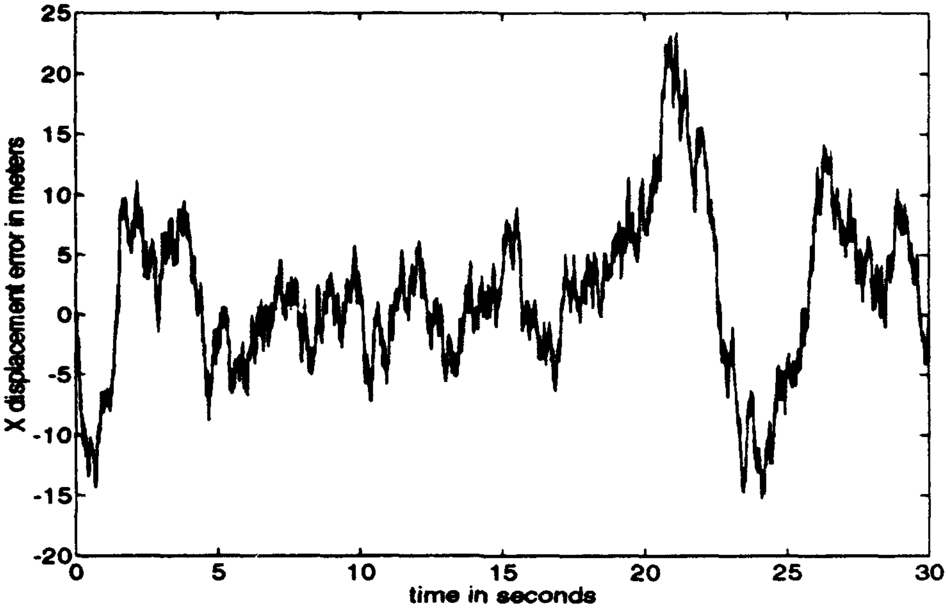

Figure 4.12: X displacement error with accelerometers

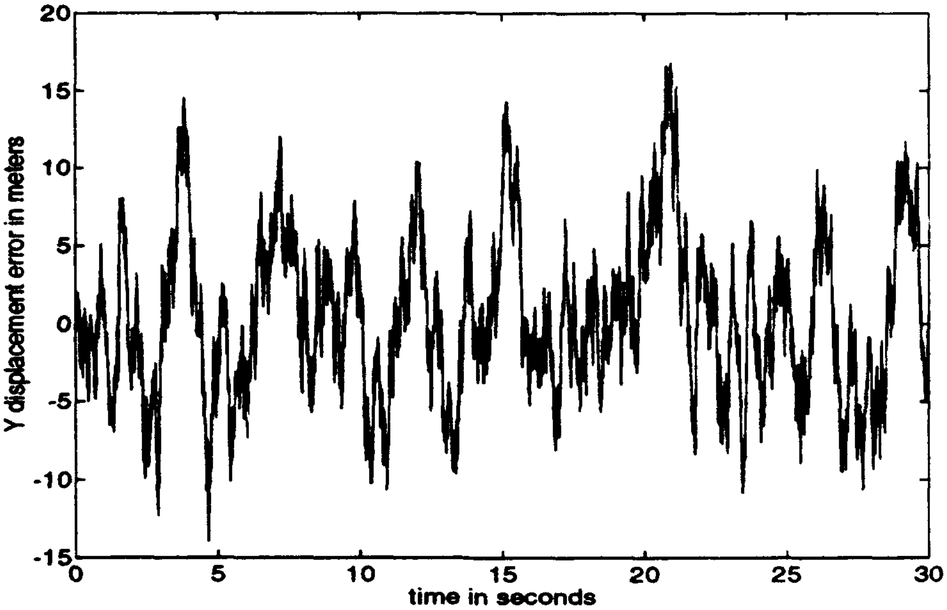

Figure 4.13: Y displacement error with accelerometers

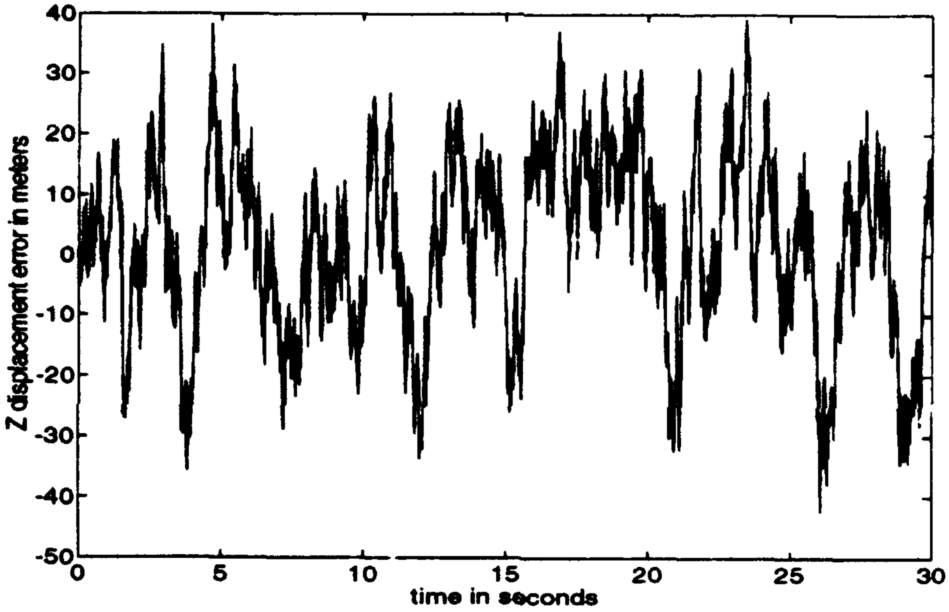

Figure 4.14: Z displacement error with accelerometers

V. CONCLUSIONS

It has been shown that raw Differential GPS outputs (pseudoranges) and raw INS outputs (inertial accelerations) can be effectively blended to yield a highly accurate position using a complementary Kalman filter. Furthermore, complete models of INS and DGPS have been developed and rigorously verified. These high fidelity models can be "transplanted" to any other work in this area.

Despite the successes just mentioned, there yet remains a great deal of work to do before this concept can be actually implemented on an aircraft. The additional refinements required are

- discretize the entire system
- use an extended Kalman filter
- account for changing satellite geometries
- use carrier phase DGPS

Each of these elements will be discussed briefly.

A. DISCRETIZE

All of the modeling done in this thesis has been done in continuous time, since it is really a proof of concept. Since computers cannot operate with zero sampling time, this is not realistic. The entire system must be implemented in discrete time. For the DGPS model, this fact necessitates no change. Nor does it require any change in the INS model. Because there is no dynamics in either the DGPS or INS model, the continuous models is identical to the discrete models.

The dynamic model of the aircraft must also be discretized. This could be easily accomplished by first linearizing the model around a cruise condition. This linear model could then be easily discretized by converting the plant matrix, A, and the input matrix, B_2, to their discrete time counterparts. The output matrices, C and D do not change.

Obviously, changing to discrete time will require the complementary filter to be implemented discretely as well. This will require the solution of the discrete, rather than the continuous, Riccatti equation. This equation is far less computationally intensive and will cause still greater rewards in the next section.

B. EXTENDED KALMAN FILTER

The navigation system designed in this work proved quite accurate, but it suffers from some unrealistic assumptions. One of these assumptions is stationary GPS satellites. This assumption allowed the complementary Kalman filter to be designed for a single, hypothetical case of space vehicle constellation to aircraft geometry. Since the satellites are in 12 hour orbits, it will be critical to account for their motion in the actual system. Therefore, an extended, complementary, Kalman filter will be required. The extended filter differs from the regular filter in that it is redesigned each time step. Rather than having a fixed linear system that the filter operates on, the system is re–linearized every time step at the estimated state. The new state space is used in the discrete Riccatti equation to find new Kalman gains. The new state space and Kalman gain matrix is used for only one iteration. Clearly, with the flight times of more than 30 minutes, this extended filter will prove necessary.

C. ACCOUNT FOR CHANGING SATELLITE GEOMETRIES

A separate but significant problem resulting from the use of stationary satellites is that the effects of altering satellite to aircraft geometry cannot be established.

Dilution of Precision is a very troublesome aspect of GPS and must be considered in a real world navigation system. There are several ways one could investigate the effects of changing geometry. One way is to actually model the satellites' orbits as a function of time. To do this accurately could prove quite cumbersome. The fact that the space vehicles broadcast 16 different coefficients which describe their trajectories indicates the complexity of the orbits.

Rather than develop a model of the entire constellations positions as a function of time, several geometries with stationary satellites can be investigated. Since the exact paths of the satellites is of no concern, there is no benefit to determining it exactly. One can just as easily verify that the Kalman filter is effectively accounting for satellite geometry by merely considering several different cases of satellite orientation. This method should provide an equally sound test of the entire scheme while greatly simplifying the process itself.

D. CARRIER PHASE GPS

Even Differential GPS based on pseudoranges turned out to be insufficiently accurate for autoland. If this is the best sensor available, how can this be improved? The answer lies with Carrier Phase GPS. This system uses a phase difference of arrival technique, like OMEGA, to evaluate position. Unlike OMEGA with its long wavelengths, GPS signals have wavelengths less than a meter. In fact, static Carrier Phase GPS positioning accuracies on the order of one centimeter have been achieved [Ref. 2, p. 64]. While the computational demands of this may still be too great for most computers light enough to fly, this method may be feasible in the near future.

APPENDIX A: LINEARIZED SYNTHESIS MODEL

This appendix includes the state space of the model derived by linearizing the non–linear synthesis model depicted in Figure 4.1. The standard state space matrices — A, B, C, and D are defined below.

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

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

$$C = \begin{bmatrix} 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & 0.86203 & 0.14603 & -0.48541 \\ 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & 0.53793 & -0.29281 & -0.79051 \\ 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & 0.15274 & -0.54389 & -0.98683 \\ 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & -0.34384 & -0.12182 & -0.93095 \\ 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & -0.56140 & 0.44964 & -0.69477 \\ 0 & 0 & 0 & 3 \times 10^8 & 0 & 0 & 0 & 0 & 0.25146 & 0.57295 & -0.78008 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.1}$$

Note that the fourth state is clearly the clock difference while the last three states are obviously the position states.

APPENDIX B: MATLAB FILES

This appendix is composed of all of the MATLAB m-files which are used by the SIMULINK diagrams. They are in alphabetical order, since no other order suggests itself.

AB_AU.M

```
function au=ab_au(x)
% this function converts the noisy "sensed" body acceleration
% to inertial coordinates and adds gravity back in
% assign local variables
ab=x(1:3);Eulers=x(4:6);
% transform to inertial coordinates
auhat=[1 0 0;0 -1 0;0 0 -1]*ru2b(Eulers)'*ab;
% add inertial gravity
au=(auhat-[0 0 32.174]');
```

ECEF2LL.M

```
function ll=ecef2ll(w)
% converts from earth-centered, earth-fixed Cartesian coordinates
% to latitude, longitude, altitude (geodetic)
x=w(1);y=w(2);z=w(3);
% define semi-major and semi-minor ellipsoid axes
a=6378137;b=6356000;
% define auxiliary parameters e and f
```

```
f=(a-b)/a;e=f*(2-f);
% convert to geodetic latitude, longitude, height
lambda=atan(y/x);
phi=atan(inv(1-e^2)*z/sqrt(x^2+y^2));
% calculate N
N=a/sqrt(1-(e*sin(phi))^2);
% find geodetic height h
h=sqrt(x^2+y^2)/cos(phi)-N;
ll=[phi;lambda;h];
```

ECF2TAN.M

```
function tanp=ecf2tan(w,phi,lambda)
% converts from earth-centered, earth fixed Cartesian coordinates
% to tangent plane
% given the latitude and longitude from the workspace
% convert tangent plane origin to radians
phi=phi;lambda=lambda;
% transformation matrix
T=[-sin(lambda)  cos(lambda) 0
   -sin(phi)*cos(lambda) -sin(phi)*sin(lambda) cos(phi)
    cos(phi)*cos(lambda)  cos(phi)*sin(lambda) sin(phi)];
% convert to tangent plane
tanp=T*(w-ll2ecef([phi,lambda,0]));
```

ELEMULT.M

```
function product=elemult(x)

% multiplies 15% of tropospheric delay noise by univariant white noise

% for randomness simulation

for i=1:6

new(i)=x(i)*x(7);

end

product=new;
```

EXACT.M

```
function r = exact(x)

% this function computes the exact range to six satellites

% given receiver position and the satellite position

satpos=zeros(6,3);

% define receiver position

recpos=x(1:3);

% define satellite position matrix

for j=1:6

satpos(j,:)=x(3*j+1:3*(j+1))';

end

% compute ranges to each satellite

for j=1:6

      ra(j)=sqrt((recpos(1)-satpos(j,1))^2+(recpos(2)-...

    satpos(j,2))^2+(recpos(3)-satpos(j,3))^2);

end

r=ra;
```

IODEL.M

```
function io_del = iodel(time)

% computes propagation delay length for signal transmitted

% from GPS satellite due to troposphere based on time of day

if time < 900

delay=5;

    elseif time > 1900

delay=5;

    else

delay=5+20*sin((time-900)*pi/1000);

end

%

io_del=delay;
```

LAMDOT1.M

```
function lam_dot=lamdot1(x)

% computes Euler derivatives given Euler angles and omega

phi=x(1);theta=x(2);psi=x(3);p=x(4);q=x(5);r=x(6);

lam_dot=[p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(theta)

 q*cos(phi)-r*sin(phi)

 (q*sin(phi)+r*cos(phi))/cos(theta)];
```

LL2ECEF.M

```
function ecef=ll2ecef(x)

% converts from geodetic latitude, longitude, elevation to

% earth-centered earth-fixed Cartesian coordinates
```

```
phi=x(1);lambda=x(2);h=x(3);

% define semi-major and semi-minor earth axes

a=6378137;b=6356000;

% define auxiliary quantities f,e, and N

f=(a-b)/a;e=f*(2-f);N=a/sqrt(1-(e*sin(phi))^2);

% convert to Cartesian

ecef=[(N+h)*cos(phi)*cos(lambda)

      (N+h)*cos(phi)*sin(lambda)

      (N*(1-e^2)+h)*sin(phi)];
```

LL2TANP.M

```
function tanp=ll2tanp(x,lat,long)

% convert from latitude, longitude, altitude to

% tangent plane coordinates given tangent plane origin


% convert to ecef

ecef=ll2ecef(x);

% convert to tangent plane

tanp=ecf2tan(ecef,lat,long);
```

REARRANGE.M

```
function new=rearrange(x)

% rearranges lambda and omega into corresponding pairs

new=[x(4) x(1) x(5) x(2) x(6) x(3)]';
```

RU2B.M

```
function Rub = ru2b(e)
% Euler angle transformation from {U} to {B}
Tpsi=[cos(e(3)) sin(e(3)) 0; -sin(e(3)) cos(e(3)) 0; 0 0 1];
Ttheta=[cos(e(2)) 0 -sin(e(2)); 0 1 0;sin(e(2)) 0 cos(e(2))];
Tphi=[1 0 0;0 cos(e(1)) sin(e(1));0 -sin(e(1)) cos(e(1))];
Rub=Tphi*Ttheta*Tphi;
```

SATELEVS.M

```
function elevs=satelevs(x)
% this function computes the elevation angles in radians of
% all six satellites
recpos=x(1:3)';satpos(1,:)=x(4:6)';satpos(2,:)=x(7:9)';
satpos(3,:)=x(10:12)';satpos(4,:)=x(13:15)';
satpos(5,:)=x(16:18)';satpos(6,:)=x(19:21)';
for i=1:6
diff=satpos(i,:)-recpos;
%  normalize difference vector to a unit vector
diff(3)=diff(3)/sqrt(diff(1)^2+diff(2)^2+diff(3)^2);
%  the angle is computed
satelev(i)=asin(diff(3));
end
elevs=satelev;
```

TAN2ECF.M

```
function ecef=tan2ecf(x,phi,lambda)
% converts from tangent plane coordinates
% to earth-centered, earth fixed Cartesian
% given the latitude and longitude of the
% origin of the tangent plane system
% convert tangent plane origin latitude and longitude to radians
phi=phi*pi/180;lambda=lambda*pi/180;
% define transformation matrix
T=[-sin(lambda)    -sin(phi)*cos(lambda)   cos(phi)*cos(lambda)
    cos(lambda)    -sin(phi)*sin(lambda)   cos(phi)*sin(lambda)
    0     cos(phi)             sin(phi)];
% convert back to degrees for ll2ecef
phi=phi*180/pi;lambda=lambda*180/pi;
% convert
ecef=T*x+ll2ecef([phi,lambda,0]);
```

TANP2LL.M

```
function ll=tanp2ll(x,lat,long)
% converts from tangent plane to latitude, longitude,
% and altitude given tangent plane origin
% convert to ecef
ecef=tan2ecf(x,lat,long);
% convert to ll
ll=ecef2ll(ecef);
```

TIMING.M

```
function minute=timing(x)
% computes number of hours and minutes elapsed since
% clock start to be added to take-off time for use in
% computing ionospheric delay
min=x/60;
% check to see if an hour has passed
if min > 60
hour=round(min/60);
else
hour=0;
end
minute=hour*100+min;
```

TROPDEL.M

```
function tropdel=tropdel(x)
% used a complex model to compute the tropospheric delay for
% all six satellites in meters
% first calculate "dry" portion, 90%
elev=x(1:6);temp=x(7);press=x(8);
ae=6378137;
hdry=148.98*(temp-4.12);
for i=1:6
idry(i)=(1-(cos(elev(i))/(1+(1-0.85)*hdry/ae))^2)^(-0.5);
sdry(i)=2.343*press*((temp-4.12)/temp)*idry(i);
end
```

```
% now the wet contribution

hwet=13000;

for i=1:6

iwet(i)=(1-(cos(elev(i))/(1+(0.15)*hwet/ae)^2))^(-0.5);

swet(i)=0.2*iwet(i);

end

% total delay in meters is the sum of the two

tropdel=sdry+swet;
```

APPENDIX C: USER'S MANUAL

This appendix describes in detail the steps required to open and run the software which generated the final results in Chapter IV. A basic familiarity with MATLAB must be assumed. Additionally, it is assumed that the user is already logged on to a SIMULINK capable Unix work station.

Before entering the MATLAB environment, one must change the working directory to the one which contains all of the code, "thesis", in this case. The command is

```
cd thesis
```

If the user is remotely logged on to a work station, he must set the DISPLAY environment variable appropriately in order to display graphics. The command which sets this variable to intrepid, a Sparc 2 work station in the Avionics Lab is

```
setenv DISPLAY intrepid.aa.nps.navy.mil:0
```

Now it is time to begin the MATLAB session by typing

```
matlab4
```

Still two more tasks must be accomplished within MATLAB before the simulation can be started. First, the numerous parameters which initialize the model must be loaded from a data file with the command

```
load bluinit
```

To bring up the SIMULINK diagram, type the command

```
bird3
```

It will take approximately 15 seconds for the simulation diagram to appear on the screen. Once it does, double click on the icon which looks like a clock at the top of the diagram. This will allow you to see the elapsed time as it slowly counts up in hundredths of a second. Finally, a single click on the "Simulation" pull down menu will reveal several choices. Choose the first of them

```
Start  ^t
```

This particular simulation takes several hours to run. It outputs "P", the actual aircraft position, "v_u", the actual aircraft velocity, "time", simulation time in seconds, and "Phat", the estimated aircraft position to the workspace. Once the simulation is complete, these data can be accessed like any other MATLAB data.

REFERENCES

1. Department of Defense and Department of Transportation, DOD-4650.4/DOT-TSC-RSPA-88-4, *1990 Federal Radionavigation Plan*, Appendix A, 1990.
2. Logsdon, T., *The NAVSTAR Global Positioning System*, Van Nostrand Reinhold, 1992.
3. NATO Team, NAVSTAR GPS Joint Program Office, *Introduction to NAVSTAR GPS User Equipment*, February, 1991.
4. "GPS Seminar", presented to the Naval Security Group at the Applied Research Laboratories of the University of Texas (Austin), January 26-28, 1993, vol. I and II.
5. Wells, D. et al., *Guide to GPS Positioning*, Canadian GPS Associates, 1987.
6. Feess, W., and Stephens, S., "Evaluation of GPS Ionospheric Time Delay Algorithm for Single Frequency Users", *Proceedings of the IEEE Position, Location and Navigation Symposium (PLANS '86)*, Las Vegas, NV, pp. 206-213, November 4-7, 1986.
7. Leick, A., *GPS Satellite Surveying*, John Wiley and Sons, 1990.
8. Black, H., "An Easily Implemented Algorithm for the Tropospheric Range Correction", Journal of Geophysical Research, v. 83, no. B4, April 1978.
9. Clynch, J., "A Global Positioning System (GPS) Users Seminar", Department of Oceanography, Naval Postgraduate School, Monterey, CA, January 16, 1992.
10. Brown, R. and Hwang, P., *Introduction to Random and Applied Kalman Filtering*, Wiley and Sons, 1992.
11. Kwakernaak, H., and Sivan, R., *Linear Optimal Control Systems*, John Wiley and Sons, 1972.
12. Van Dierendonck, A., McGraw, J., and Brown, R., "Relationship Between Allan Variances and Kalman Filter Parameters", *Proceedings of the Sixteenth Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting*, pp. 273–293, NASA Goddard Space Flight Center, November 27-29, 1984.
13. Brown, A., "Extended DGPS", *Navigation*, v. 36, no. 3, pp. 265–285, Fall 1989.
14. Langley, R., "The Mathematics of GPS", *GPS World*, August 1991.
15. Lin, C., *Modern Navigation, Guidance, and Control Processing*, Prentice Hall, 1991.
16. Kuechenmeister, D., *A High-Fidelity Non-Linear Simulation of an Autonomous Unmanned Air Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1993.

17. Gelb, A. and others, *Applied Optimal Estimation*, The MIT Press, 1974.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Dr. Isaac I. Kaminer Department of Aeronautics and Astronautics, Code AA/KA Naval Postgraduate School Monterey, CA 93943-5000	5
4.	LCDR Michael K. Shields Department of Electrical and Computer Engineering, Code EC/SL Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Chairman Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943-5000	2
6.	Capt. David R. Kuechenmeister 1995 Skidmore Circle Lawrenceville, GA 30244	1
7.	Carl W. Marquis, III 1276 Spruance Road Monterey, CA 93940	1

		No. of Copies
8.	Carl W. Marquis 128 B Delphia Drive Brevard, NC 28712	1
9.	LT Joe Fordham P. O. Box 6778 Salinas, CA 93912	1