INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

NOTE TO USERS

Page not included in the original manuscript is unavailable from the author or university. The manuscript was microfilmed as received.

144

This reproduction is the best copy available.

UMI

Comparison of Linearized and Extended Kalman Filter in GPS Aided Inertial Navigation System

by

Pande Hartana

B.Eng., Sepuluh Nopember Institute of Technology 1993

A thesis submitted to the faculty of
Graduate Studies and Research in partial
fulfillment of the requirements for the Degree of

Master of Engineering

**Department of
Mechanical and Aerospace Engineering**

**Ottawa-Carleton Institute
for Mechanical and Aerospace Engineering**

Carleton University

Ottawa, Ontario, Canada

July 7, 2000

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-57729-5

Canada

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

“COMPARISON OF LINEARIZED AND EXTENDED KALMAN FILTER
IN GPS AIDED INERTIAL NAVIGATION SYSTEM”

submitted by Pande Hartana, B.Eng.

in partial fulfillment of the requirements for
the Degree of Master of Engineering

Thesis Supervisor

Chair, Department of Mechanical and Aerospace Engineering

Carleton University

July 7, 2000

Abstract

One of the disadvantages of an inertial navigation system (INS) is its errors will grow unbounded. The global positioning system (GPS) can be used to improve position estimation resulting from INS measurement. The Kalman filter method is used to fuse measurements from the systems (INS and GPS). When the channels available in the GPS receiver are restricted, the receiver has to choose the combination of satellites, which will give the lowest noises in the GPS measurement signal. In this thesis, the method to find the optimal combination of the satellites has been examined. To effect of the change of satellite combination on error estimation has been investigated and presented. The comparison of the implementation of indirect half feedback and indirect feedforward Kalman filter configurations in an aircraft has been performed and analyzed. The GPS and INS measurement signals used in this experiment are real data, which was recorded on an aircraft during the flight. The experiment shows the relation between combination of the satellites and the estimated position and velocity of the aircraft. The improvements in the indirect half feedback configuration are also achieved.

Acknowledgements

I would like to express my deep gratitude to my supervisor, Prof. J. Z. Sasiadek for his encouragement and support. Without his patience and advices, this thesis would not have been completed.

I would also like to thanks the faculty members and staff of the Department of Mechanical and Aerospace Engineering of Carleton University for their help throughout my study in the Master of Engineering program at Carleton University.

Finally, I would like to acknowledge Mr. Stephen T. Ferguson's valuable help in data analyzing and his advice on technical matters.

Table of Contents

Title	i
Acceptance Form	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Figures	viii
List of Tables	x
Nomenclature	xi
1 Introduction	1
2 Reference Coordinate Systems	7
2.1 Types of Coordinate System	7
2.1.1 Earth-Centered Inertial (ECI) coordinate system	7
2.1.2 Earth-Centered, Earth-Fixed (ECEF) Cartesian coordinate system	8
2.1.3 Earth-Centered, Earth-Fixed (ECEF) Geodetic coordinates system	9
2.1.4 Local geodetic coordinate system	10
2.1.5 Local geodetic wander azimuth coordinate system	10
2.2 Relation between Coordinate Systems	11
2.2.1 Relation between ECEF Cartesian and ECEF geodetic coordinate system ..	12
2.2.2 Relation between local geodetic and ECEF Cartesian coordinate system	14
2.2.3 Relation between local geodetic wander azimuth and ECEF Cartesian coordinate system	16

3	Inertial Navigation Systems.....	20
3.1	Principle and Types of Inertial System	21
3.2	INS Mechanization Model	27
3.3	INS Error Considerations.....	30
4	Global Positioning Satellites.....	32
4.1	GPS Overview.....	33
4.2	Triangulation Using Time-of-arrival Measurements	35
4.3	Position Determination Using Pseudorandom Noise (PRN) Codes	38
4.4	Error Sources and Error Budget.....	44
4.5	Satellite Geometry and Dilution of Precision	47
5	Kalman Filter	53
5.1	The Concept of Kalman Filter	54
5.2	The Discrete Kalman Filter.....	58
5.3	Linearized and Extended Kalman Filters.....	65
6	Externally Aided Inertial Navigation System.....	67
6.1	Fusing Methods Using Kalman Filter.....	68
6.1.1	Total state space configuration	69
6.1.2	Error state space configuration	70
6.2	GPS Aided Inertial Navigation System.....	72
7	Simulation Experiment Design and Results	83
7.1	Kalman Filter Configurations	83
7.2	The INS and GPS Data	87
7.3	Satellite Selection Algorithm	96
7.4	Results of Simulation Experiment	98
8	Conclusions.....	118
	Reference	120
	Appendix A.....	126
A.1	GPS Measurements Data in Rinex2 Format	127
A.2	GPS Ephemeris Data in SP3 Format.....	129

Appendix B	132
B.1 Kalman.m	134
B.2 StateTransMatrix.m	137
B.3 ProcNoiseCov.m	138
B.4 MeasNoiseCov.m	139
B.5 ReadPosCar.m	139
B.6 CartToGeod.m	140
B.7 ChooseSatComb.m	140
B.8 ArrangeComb.m	141
B.9 SatVis.m	142
B.10 CreateNewSatComb.m	143
B.11 FindPdop.m	144
B.12 ChooseNewSatComb.m	145
B.13 MeasMatrix.m	146
B.14 HTranMat.m	147
B.15 PredMeas.m	148

List of Figures

Figure 1.1	Diagram block of automatic control system using one sensor.....	3
Figure 1.2	Diagram block of automatic control system using multi sensors.....	3
Figure 2.1	ECEF Cartesian coordinate system.....	9
Figure 2.2	Local geodetic coordinate system.....	10
Figure 2.3	Local geodetic wander azimuth coordinate system.....	11
Figure 2.4	Relation between two positions in local geodetic coordinate system.....	14
Figure 2.5	Relation between local, Earth, and inertial coordinate system.....	17
Figure 3.1	Typical mechanized-platform INS configuration.....	23
Figure 3.2	East axis error model.....	28
Figure 3.3	North axis error model.....	29
Figure 3.4	Altitude axis error model.....	29
Figure 4.1	Control segment operations.....	34
Figure 4.2	The schematic of GPS receiver components.....	35
Figure 4.3	User located on surface of sphere.....	36
Figure 4.4	User located on perimeter of shaded circle.....	37
Figure 4.5	Satellites' plane of intersection.....	37
Figure 4.6	User located at one of two points on shaded circle.....	38
Figure 4.7	User located at one of two points on circle perimeter.....	38
Figure 4.8	User position vector representation.....	40
Figure 4.9	Replica code is used to determine satellite code transmission time.....	41
Figure 4.10	Relationship of range measurement and time.....	42
Figure 4.11	Range measurement timing relationship.....	45
Figure 4.12	Geometry with low DOP.....	48
Figure 4.13	Geometry with high DOP.....	49
Figure 5.1	Guidance, navigation, and control block diagram [1].....	53
Figure 5.2	Typical Kalman filter application [31].....	56
Figure 5.3	System and measurement model.....	58
Figure 5.4	Kalman filter loop.....	64
Figure 5.5	Block diagram system, measurement model, and discrete-time Kalman filter.....	65

Figure 6.1	Total state space (direct) Kalman filter.	69
Figure 6.2	Indirect feedforward Kalman filter.	71
Figure 6.3	Indirect feedback Kalman filter.	72
Figure 6.4	GPS aided inertial navigation system configuration.	73
Figure 6.5	East position error model.	74
Figure 6.6	North position error model.	74
Figure 6.7	Vertical position error model.	75
Figure 6.8	Clock model.	76
Figure 7.1	Flow diagram for indirect feedforward Kalman filter.	85
Figure 7.2	Flow diagram indirect half feedback Kalman filter.	86
Figure 7.3	Path followed by the aircraft during the flight.	87
Figure 7.4	Longitude coordinate followed by the aircraft during the flight.	88
Figure 7.5	Latitude coordinate followed by the aircraft during the flight.	88
Figure 7.6	Aircraft velocity profile during the flight (East channel).	89
Figure 7.7	Aircraft velocity profile during the flight (North channel).	89
Figure 7.8	East channel accelerometer output.	90
Figure 7.9	North channel accelerometer output.	90
Figure 7.10	Vertical channel accelerometer output.	91
Figure 7.11	Distribution of INS measurement (East channel).	92
Figure 7.12	Distribution of INS measurement (North channel).	92
Figure 7.13	Distribution of INS measurement (Vertical channel).	93
Figure 7.14	Satellite visibility as function of time.	94
Figure 7.15	East position error between INS and GPS data.	95
Figure 7.16	North position error between INS and GPS data.	95
Figure 7.17	Flow diagram of satellite selection based on smallest PDOP.	97
Figure 7.18	Flow diagram of satellite selection algorithm.	97
Figure 7.19	East position error.	101
Figure 7.20	North position error.	102
Figure 7.21	Vertical position error.	103
Figure 7.22	East velocity error.	104
Figure 7.23	North velocity error.	105
Figure 7.24	Vertical velocity error.	106
Figure 7.25	East position error covariance.	107
Figure 7.26	North position error covariance.	108
Figure 7.27	Vertical position error covariance.	109
Figure 7.28	East velocity error covariance.	110
Figure 7.29	North velocity error covariance.	111
Figure 7.30	Vertical velocity error covariance.	112
Figure 7.31	Kalman gain.	113
Figure 7.32	Innovation process on first channel.	114
Figure 7.33	Innovation process on second channel.	115
Figure 7.34	Innovation process on third channel.	116
Figure 7.35	Innovation process on fourth channel.	117

List of Table

Table 4.1	GPS PPS pseudorange error budget.	47
Table 4.2	GPS SPS pseudorange error budget.	47
Table 7.1	Selection of the satellites during simulation.	99

Nomenclature

Chapter 2

x, y, z = Earth-Centered Inertial (ECI) coordinate system.

u, v, w = Earth-Centered, Earth-Fixed (ECEF) Cartesian coordinate system.

ρ, λ, h = Earth-Centered, Earth-Fixed (ECEF) geodetic coordinate system (latitude, longitude, height).

e, n, h = local geodetic coordinate system (east, north, height).

e', n', h' = local geodetic wander azimuth coordinate system (east, north, height).

α = wander azimuth angle.

a = the length of semimajor axis of the earth.

a = zenith angle.

f = flattening factor of the earth.

R_i = transformation matrices.

Chapter 3

$\varepsilon_x, \varepsilon_y, \varepsilon_z$ = gyroscope's measurements.

$\omega_x, \omega_y, \omega_z$ = INS angular velocity.

ϕ_x, ϕ_y, ϕ_z = INS angular misalignment.

a_x, a_y, a_z = accelerometer's measurements.

$\Delta\dot{x}, \Delta\dot{y}, \Delta\dot{z}$ = INS linear velocity.

$\Delta x, \Delta y, \Delta z$ = INS linear position error.

g = earth gravitational field.

R = radius of the earth.

Chapter 4

R = radius of the earth.

c = speed of the light.

ρ_i = pseudorange.

Chapter 5

$\mathbf{x}$ = vector of the state variables.

$\mathbf{P}$ = error covariance matrix.

$\mathbf{z}$ = vector of the measurement.

Φ = state transition matrix.

$\mathbf{H}$ = measurement sensitivity matrix.

$\mathbf{w}$ = system noise.

$\mathbf{v}$ = measurement noise.

$\mathbf{Q}$ = system noise covariance.

$\mathbf{R}$ = measurement noise covariance.

$\mathbf{K}$ = Kalman gain.

Chapter 1

Introduction

To follow the designed path, an autonomous vehicle has to be equipped with three systems: navigation, guidance, and control system [1], [2]. The navigation system is used to provide estimation of position and velocity of the vehicle. The guidance system is used to determine the optimal trajectory from a current position and velocity to a desired position and velocity. The control system is used to determine the commands for vehicle's actuators to drive the actual velocity and attitude of the vehicle to the value commanded by the guidance scheme.

For the navigation system, there are two basic position-estimation methods commonly applied, i.e. relative and absolute positioning [3], [4], [5], [6], [7]. Relative positioning, which is sometimes called dead reckoning, is usually based on inertial sensors or odometry sensors. In this method, the calculated distance from initial position determines current position estimation. In an absolute positioning system, the positioning sensors interact with a dynamic environment, which can be navigation beacons, active or passive landmark, map matching, or satellite-based navigation signal, to find the position estimation.

To solve the positioning problems, there are two types of sensors available: internal and external sensors [8]. Internal sensors measure physical variables on the

vehicle itself. This self-containing characteristic means the measurement results of these sensors are almost always available to solve positioning problems. The examples of these sensors are accelerometer, encoders, gyroscopes, and compasses. External sensors measure relationships between the robot and its environment, which can be natural or artificial objects. The examples of external sensors are satellite signal receiver, sonar sensor, radars, and laser range finders.

When the above sensors are implemented to solve positioning problems, both have advantages and disadvantages. For short periods, measurements using internal sensors are quite accurate. However, for long-term estimation, the measurements usually produce a drift. On the contrary, because it measures absolute quantity, external sensors do not produce the drift, however, the measurements from these sensors are usually not always available [9].

Figure 1.1 shows a diagram block for a common automatic control system, in this case the system uses only one sensor to provide feedback to the controller. When an error occurs on the sensor, as explained in the previous paragraph, the input to the controller will also contain error. In this case, the functioning of the system is faulty. This problem raises the idea of using multiple sensors in the system. When using multiple sensors, the problem now is which measurement will be used as the feedback to the controller.

The answer to that question is to use sensors fusion. The sensors fusion tries to fuse the measurements and find the optimal one to use as the feedback. The system now can be depicted as in Figure 1.2.

Figure 1.1 - Diagram block of automatic control system using one sensor.

Figure 1.2 - Diagram block of automatic control system using multi sensors.

Inertial navigation systems (INS) are widely used in aviation. It implements accelerometers and gyroscopes to provide velocity and angular rate information. By integrating this information, the position and orientation of the vehicle can be calculated. To improve the estimation resulting from the INS, its measurements are usually fused with the measurement from global positioning system (GPS), which are relatively accurate. The Kalman filter is the common algorithm used to fuse the measurements.

There is much research already conducted to improve position estimation when implementing INS and GPS, whether as a standalone estimation method, or as an integration system. Vanderwerf [10] tried to find the effect of gravity anomalies on the

inertial velocity errors. Harriman and Harrison [11] analyzed gravity-induced errors in airborne inertial navigation. Pozo-Ruz et al. [12] introduced satellite selection criterion to improve accuracy of the estimation when using GPS. Abusali et al. [13] used simulated cross-link pseudorange measurement to update the predicted trajectory of the satellite. This, in turn will improve the accuracy of navigation system using standalone GPS.

Integration of INS and radio signal was simulated by Li [14] in his work, “Simulated an Adaptive Kalman Filter for Tactical Missile Guidance Systems.” Lawrence [15] introduced the distributed Kalman filter simulator (DKFSIM), which is used to evaluate the performance of several different filter architectures and sensor model conditions for advanced, multi-sensor navigation systems. Sukkarieh [16] examined the integration of INS-GPS for autonomous land vehicles. The most recent work is done by Wang [17], who simulated the application of fuzzy logic and Kalman filtering to control and navigate autonomous vehicles and robots.

Currently, there are 27 global positioning satellites available to use for navigation. From any particular location, a receiver can view around 10 satellites in one time. Although the estimation is usually improved when using more satellites, there is common situation where the channels available to receive the measurement signals are restricted. Moreover, not all the visible satellites are good to be used as the sources of measurement [12]. Therefore, the receiver has to choose the best combination of satellites in view to get the optimal estimation.

With the above considerations, the following problems are solved in this thesis:

1. To find the best combination of satellites from a number of visible satellites.

2. To view the effects of the change of satellites in combination to the estimation.
3. To compare the implementation of linearized and extended Kalman filter in GPS aided inertial navigation system.

To solve those problems, this thesis is organized as follows:

□ Chapter 2

In this chapter, various coordinate systems commonly used in GPS and inertial navigation applications are explained. The transformation equations between coordinates are also derived.

□ Chapter 3

In Chapter 3, aspects of INS are explained. The error models used in the implementation of GPS aided inertial navigation system are derived here.

□ Chapter 4

Chapter 4 discusses GPS. The linearization of GPS measurements is explained in this chapter.

□ Chapter 5

The derivation of the Kalman filter based on the Bayesian approach is explained in Chapter 5. Linearization methods for nonlinear problems are also discussed.

□ Chapter 6

Various configurations of the integration of GPS and INS are discussed in Chapter 6. The derivation of equations used in the integration of GPS and INS measurement using half feedback Kalman filter configuration is derived here. These equations will be used in the simulation.

□ Chapter 7

In Chapter 7, the design of a simulation experiment and its results are discussed. The analysis of the results is also conducted here.

□ Chapter 8

Finally, in Chapter 8, the conclusions are drawn.

Chapter 2

Reference Coordinate Systems

In order to describe position, velocity, and altitude in navigation problems, it is important to choose a reference coordinate system in which those states can be represented. In this chapter, various coordinate systems commonly used in navigation are explained. The equations used to relate between coordinate systems are also represented. The chapter begins with the explanation of commonly used of coordinate systems, and continues to the relation between coordinate systems. The complete explanation of these coordinate systems can be found from [18], [19], [20], and [21].

2.1 Types of Coordinate System

In navigation problems, especially in aircraft navigation, various coordinate systems are commonly used. Carefully representing navigation problems in a particular coordinate system should reduce the complication of the analysis. In this section, those coordinate systems are represented.

2.1.1 Earth-Centered Inertial (ECI) coordinate system

This space-fixed rectangular coordinate system has its origin at the mass-center of the Earth. In typical ECI coordinate system, the xy -plane is taken to coincide with the Earth's equatorial plane, the $+x$ -axis is permanently fixed in a particular direction

relative to the celestial sphere, the $+z$ -axis is taken normal to the xy -plane in the direction of the north pole, and the $+y$ -axis is chosen so as to form a right-handed coordinate system. In ECI coordinate systems, Newton's laws are valid. Moreover, the angular coordinates of stars are conventionally tabulated in ECI. These make the ECI coordinate system suitable to be used in outer space navigation problem. In navigation problem, this type of coordinate is usually referred as inertial coordinate.

2.1.2 Earth-Centered, Earth-Fixed (ECEF) Cartesian coordinate system

This is the basic coordinate system for navigation near Earth. These rectangular coordinates have their origin at the mass center of the Earth. Its uv -plane (to make it different notation from ECI coordinate system) is coincident with the Earth equatorial plane, its w -axis is parallel to the direction of conventional terrestrial pole, and its u -axis is toward the intersection between Greenwich meridian and the plane of the terrestrial equator. The v -axis points to the direction of 90° E longitude, thereby completing the right-handed coordinate system.

This coordinate system rotates with the Earth. In these coordinates, accelerations are present, so the laws of general relativity must be taken into account. However, since the main relativistic effect is caused by the gravity field of the Earth itself, this geocentric coordinate system is better suited for the navigation problem close to the Earth. Satellite-based radio-navigation systems often use these ECEF coordinates to calculate satellite and aircraft positions.

2.1.3 Earth-Centered, Earth-Fixed (ECEF) Geodetic coordinates system

These coordinates are normal to the reference ellipsoid, the reference used to describe the ellipsoid of the Earth. It also moves with the Earth, so its relation to the ECEF Cartesian can be derived. The position of a point is described by latitude, longitude, and height. The angle between the ellipsoidal normal and the equatorial plane is the geodetic latitude ρ . The angle between two meridian planes counted in counter clockwise direction from the u -axis is the geodetic longitude λ . (Meridian is any intersection of the ellipsoid with a plane containing the w -axis.) The length of the ellipsoidal normal from the ellipsoid to the point is called the ellipsoidal or geodetic height h . Thus, the geodetic coordinates (ρ, λ, h) completely describe the position of a point in space.

Figure 2.1 shows the configuration of the ECEF coordinate system, and the relation between ECEF Cartesian and ECEF geodetic.

Figure 2.1 - ECEF Cartesian coordinate system.

2.1.4 Local geodetic coordinate system

When dealing with near-Earth navigation problems, sometimes it is beneficial to describe the position and velocity in local geodetic coordinate system. In this system, position and velocity of any particular point are referred from the other known point, so its position and velocity will be relative to this known point. The coordinates are described by (e, n, h) . The e -axis and n -axis lie on the local geodetic horizon. Local geodetic horizon is a plane, which is perpendicular to the ellipsoid normal through the known point. The e -axis point east, the n -axis point north, and the h -axis coincides with the ellipsoidal normal, with the positive end outward from the ellipsoid. Sometimes, this coordinate is referred as geographic coordinate. The depiction of this coordinate system is shown in Figure 2.2.

Figure 2.2 - Local geodetic coordinate system.

2.1.5 Local geodetic wander azimuth coordinate system

In inertial navigation problems, a local geodetic wander azimuth coordinate system is usually used. As in local geodetic coordinate system, these coordinates are

locally level to the reference ellipsoid. The position of any particular point is described by (e', n', h) . The e' -axis makes an angle α north of the true east, the n' -axis makes an angle α west of the true north, and the h -axis points outward of the ellipsoid. The angle α is called the wander azimuth angle. This coordinate is also referred to as the navigation coordinate system. Figure 2.3 shows the configuration of local geodetic wander azimuth coordinate system.

Figure 2.3 - Local geodetic wander azimuth coordinate system.

2.2 Relation between Coordinate Systems

When position of one particular object is related to the positions of the other objects, it is greatly simplified if all of those objects are represented in one coordinate system. In this section, the relations between coordinate systems and the equations involved in those relations are explained.

2.2.1 Relation between ECEF Cartesian and ECEF geodetic coordinate system

Based on Figure 2.1, the relation between ECEF Cartesian coordinates, which is expressed by (u, v, w) , and ECEF geodetic coordinate, expressed in (ρ, λ, h) is

$$u = (N + h) \cos \rho \cos \lambda \quad (2.1)$$

$$v = (N + h) \cos \rho \sin \lambda \quad (2.2)$$

$$w = [N(1 - e^2) + h] \sin \rho \quad (2.3)$$

where the quantity N and e are

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \rho}} \quad (2.4)$$

$$e^2 = 2f - f^2. \quad (2.5)$$

The quantity a in (2.4) is the length of semimajor axis of the Earth, which is the same as equatorial radius of the Earth. The value is 6378136.49 meter [22]. The quantity f in (2.5) is flattening factor of the Earth, and the value is 1/298.25642 [22].

The inverse solution of (2.1) to (2.3) can also be computed. The longitude λ follows immediately from (2.1) and (2.2):

$$\tan \lambda = \frac{v}{u}. \quad (2.6)$$

The latitude ρ can be approximated (see Figure 2.1) as:

$$\tan \rho = \frac{(N + h) \sin \rho}{\sqrt{u^2 + v^2}}. \quad (2.7)$$

This equation is nonlinear. To solve this equation, iteration processes can be used.

Rewriting (2.3) as

$$(N + h) \sin \rho = w + e^2 N \sin \rho \quad (2.8)$$

and substituting it into (2.7) gives

$$\tan \rho = \frac{w}{\sqrt{u^2 + v^2}} \left(1 + \frac{e^2 N \sin \rho}{w} \right). \quad (2.9)$$

In the iteration process, the second term in (2.9) can be modified using an approximation for the w coordinate. A zero ellipsoidal height $h = 0$ can be used as a first approximation. Using this, (2.8) can be written as:

$$w_{h=0} = N(1 - e^2) \sin \rho \quad (2.10)$$

Substituting (2.10) into (2.9) gives a first approximation for the geodetic latitude:

$$\tan \rho_1 = \frac{w}{(1 - e^2) \sqrt{u^2 + v^2}}. \quad (2.11)$$

This value can be substituted into the second term of (2.9), which results a new and improved value for ρ . The iteration stops after the error in the successive solutions of ρ is negligible. After the geodetic latitude is determined, the ellipsoidal height is calculated as

$$h = \frac{\sqrt{u^2 + v^2}}{\cos \rho} - N. \quad (2.12)$$

This completes the relation between ECEF Cartesian and ECEF geodetic coordinate system section.

2.2.2 Relation between local geodetic and ECEF Cartesian coordinate system

Figure 2.4 shows the relation between two points in a local geodetic coordinate system. The position of point P_2 relative to P_1 is described by azimuth angle α , zenith angle a , and the distance between two points s .

Figure 2.4 - Relation between two positions in local geodetic coordinate system.

It follows that

$$e = s \cos a \sin(-\alpha) \quad (2.13)$$

$$n = s \cos a \cos(-\alpha) \quad (2.14)$$

$$h = s \sin a. \quad (2.15)$$

The inverses of these equations are

$$-\alpha = \tan^{-1}\left(\frac{e}{n}\right) \quad (2.16)$$

$$a = \sin^{-1}\left(\frac{h}{s}\right) \quad (2.17)$$

$$s = \sqrt{n^2 + e^2 + h^2}. \quad (2.18)$$

The relationship between the local geodetic coordinate system and the ECEF Cartesian coordinate system then can be described by equation:

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = R_3(-90^\circ - \lambda) R_1(\rho - 90^\circ) \begin{bmatrix} e \\ n \\ h \end{bmatrix} \quad (2.19)$$

where

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (2.20)$$

$$R_3(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

are respectively the rotation matrices around the first axis and third axis, and

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \\ w_2 - w_1 \end{bmatrix}. \quad (2.22)$$

Combining the rotation matrices R_1 and R_3 results in

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = R(\rho, \lambda) \begin{bmatrix} e \\ n \\ h \end{bmatrix} \quad (2.23)$$

where

$$R(\rho, \lambda) = \begin{bmatrix} -\sin \lambda & -\cos \lambda \sin \rho & \cos \lambda \cos \rho \\ \cos \lambda & -\sin \lambda \sin \rho & \sin \lambda \cos \rho \\ 0 & \cos \rho & \sin \rho \end{bmatrix}. \quad (2.24)$$

The inverse of (2.23) can be written as:

$$\begin{bmatrix} e \\ n \\ h \end{bmatrix} = R'(\rho, \lambda) \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \quad (2.25)$$

where

$$R'(\rho, \lambda) = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\cos \lambda \sin \rho & -\sin \lambda \sin \rho & \cos \rho \\ \cos \lambda \cos \rho & \sin \lambda \cos \rho & \sin \rho \end{bmatrix} \quad (2.26)$$

This completes the relation between local geodetic and ECEF Cartesian coordinate systems.

2.2.3 Relation between local geodetic wander azimuth and ECEF Cartesian coordinate systems

In inertial navigation, the local geodetic wander azimuth (or navigation) coordinate system is usually used to describe the inertial measurement. In GPS problems, the positions of the satellites are described in the form of an ECEF coordinate system. It

is necessary to relate both of those coordinate systems so that the state in navigation problem can be clearly described in one particular coordinate system.

Figure 2.5 shows the relation between local, ECEF, and inertial coordinate systems. As described in the previous section, the ECEF coordinate system is centered at the mass center of the Earth. It rotates at $\Omega = 15.04107$ deg/hr relative to the ECI coordinate system, to an accuracy of 5×10^{-5} deg/hr.

Figure 2.5 - Relation between local, Earth, and inertial coordinate system

The local geodetic wander azimuth coordinates n' and e' represent a set of orthogonal accelerometers whose x and y axes are level and whose y -axis makes an angle α west of true north. The angle α is known as wander azimuth angle. The local

geodetic wander azimuth coordinate system (e', n', h) can be related to the ECEF Cartesian coordinate system by the following equation:

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = R_3(-90^\circ - \lambda) R_1(\rho - 90^\circ) R_3(-\alpha) \begin{bmatrix} e' \\ n' \\ h \end{bmatrix} \quad (2.27)$$

where R_1 and R_3 are the same as in (2.20) and (2.21), and the left term in (2.27) is the same as in (2.22).

Combining R_1 , R_2 , and R_3 , results in:

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = R(\rho, \lambda, \alpha) \begin{bmatrix} e' \\ n' \\ h \end{bmatrix} \quad (2.28)$$

where

$$R(\rho, \lambda, \alpha) = \begin{bmatrix} -S_\lambda C_\alpha - C_\lambda S_\rho S_\alpha & S_\lambda S_\alpha - C_\lambda S_\rho S_\alpha & C_\lambda C_\rho \\ C_\lambda C_\alpha - S_\lambda S_\rho S_\alpha & -C_\lambda S_\alpha - S_\lambda S_\rho C_\alpha & S_\lambda C_\rho \\ C_\rho S_\alpha & C_\rho C_\alpha & S_\rho \end{bmatrix} \quad (2.29)$$

and

$$C_i = \cos(i) \quad (2.30)$$

$$S_i = \sin(i). \quad (2.31)$$

The inverse of (2.28) can be written as:

$$\begin{bmatrix} e' \\ n' \\ h \end{bmatrix} = R'(\rho, \lambda, \alpha) \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \quad (2.32)$$

where

$$R'(\rho, \lambda, \alpha) = \begin{bmatrix} -S_\lambda C_\alpha - C_\lambda S_\rho S_\alpha & C_\lambda C_\alpha - S_\lambda S_\rho S_\alpha & C_\rho S_\alpha \\ S_\lambda S_\alpha - C_\lambda S_\rho C_\alpha & -C_\lambda S_\alpha - S_\lambda S_\rho C_\alpha & C_\rho C_\alpha \\ C_\lambda C_\rho & S_\lambda C_\rho & S_\rho \end{bmatrix}. \quad (2.33)$$

The quantities inside this matrix are the same as in (2.30) and (2.31).

Chapter 3

Inertial Navigation Systems

Inertial navigation systems (INS) are one of the navigation systems, which now become standard equipment on most plane, ships, and submarines [23]. It is also applied for missile guidance; in fact, it was first used during World War II in the German V1 and V2 weapons [24], [25].

The term inertial in this navigation system comes from the fact that it uses the inertial properties of matter for their operation. It consists of accelerometers and gyroscopes. Accelerometers measure acceleration. Integrating this acceleration gives speed, and integrating speed gives distance traveled. Gyroscopes measure rotation. They provide information on where the accelerations are directed. Therefore, the INS provides heading and distance, the essential factors in navigation.

Compared with other methods of navigation, the inertial navigation system has the following advantages [18]:

1. INS is able to indicate position and velocity instantaneously and continuously.
High data rates and bandwidths are easily achieved.
2. Since it is based on measurements of acceleration and angular rate made within the vehicle itself, the INS is non-radiating and non-jammable.

3. Navigation information (including azimuth) can be obtained at all latitudes (including the polar regions), in all weather, without the need for ground stations.
4. The INS provides outputs of position, ground speed, azimuth, and vertical. It is the most accurate means of measuring azimuth and vertical on a moving vehicle.

The disadvantages of INS are the following:

1. The navigation information recorded by INS degrades with time. This is true whether the vehicle is moving or stationary.
2. Initial alignment is necessary. Alignment is simple on a stationary vehicle at moderate latitudes; however, it degrades at latitudes greater than 75° and on moving vehicles.
3. The accuracy of navigation information is dependent on vehicle maneuvers.

In this chapter, the basic principle of INS is explained. The chapter begins with the explanation of INS principles and types. It continues to the derivation of a nine-state INS error model. The sources of error that have to be considered are also explained briefly.

3.1 Principle and Types of Inertial System

Inertial navigation is a technique to determine a vehicle's position and velocity from its acceleration. Its principle comes from the application of Newton's law of motion. Specifically, the Newton's first law of motion states that a body in motion tends to maintain its motion unless acted on by a force. This means that if a force-sensing

device is in motion, it maintains its motion until acted on by a force, which the device senses. INS, in particular, is a force-sensing device. Since it is designed with a known mass, Newton's second law can be applied to determine the acceleration as:

$$a = \frac{F}{m} . \quad (3.1)$$

Velocity of the vehicle can be derived from this acceleration. If the velocity of the vehicle at any particular time is v_0 , the velocity after time duration t is defined as:

$$v - v_0 = \int_0^t a \, dt . \quad (3.2)$$

In the same way, if the position of a vehicle at any particular time is s_0 , the position after time t can be defined as:

$$s - s_0 = \int_0^t v \, dt . \quad (3.3)$$

Using the above basic equations in every coordinate of the navigation frame, position and velocity of the vehicle can be solved. Because INS uses acceleration to determine navigation properties of the vehicle, when it is used in a gravitational field, it requires compensation for this effect [10], [11].

There are primarily two possible implementation approaches for inertial navigation: *mechanized-platform systems* and *strap-down systems*.

Figure 3.1 shows typical configuration of mechanized-platform system. In this system, the inertial sensors (accelerometers and gyroscopes) are mounted on an actuated

platform. The gyroscopes act as null-sensors. The angles sensed by gyroscopes drive the gimbal servos, which maintain the platform-frame alignment with a specified navigation coordinate system. When the platform is aligned, accelerometers measure the specific force, which can be transformed into acceleration, along the navigation coordinate system axes. Integrating this acceleration results in the desired navigation-frame position and velocity vectors. The attitude of the vehicle is determined by measuring the relative angles between the vehicle frame and the platform axes.

Figure 3.1 - Typical mechanized-platform INS configuration.

In strap-down systems, the inertial sensors are attached directly on the vehicle frame. Using this approach, the sensors have to be able to sense the full dynamic motion of the vehicle. Therefore, gyroscopes with higher bandwidth rate and higher dynamic

range are required. Because of the increased dynamic range, the importance of gyro scale-factor error and nonlinearity becomes significant. The relationship between vehicle, navigation, and inertial coordinate frames has to be maintained computationally. This increases the on-board computational load compared with the mechanized system.

The choice between these two approaches in application is complex and depends on the state of sensor and computing technology. Mechanized systems have smaller computational burdens and expose the inertial sensors to a more benign inertial environment, but are typically larger and more expensive. On the other hand, strap-down system has more advance requirement. Before continuing to the derivation of INS error model equations, in the following, the advantages and disadvantages of mechanized and strap-down systems are briefly discussed [24].

The advantages of mechanized system are:

1. It uses simpler gyros.

Because platform where the sensors are attached rotates only at the small rates to keep itself level, the gyros need only a small dynamic range.

2. It gives higher accuracy.

Because the platform is always level, so are the accelerometers. The accelerometer axes are always well defined. This makes the mechanized system can be very accurate. In North and East directions, accelerometers see no component of gravity and measure only the vehicle accelerations. In vertical direction, however, the accelerometer's measurement is affected by the presence of the earth's gravitational field, which makes it less accurate. In

aircraft application, this causes altitude errors, which can be compensated with a barometer signal.

3. Self-alignment by gyrocompassing.
4. Sensor calibration by platform rotations.

Sensor biases are obtained by orienting the platform so as the major axis of the sensor is in vertical direction.

The disadvantages of mechanized system are:

1. Complexity and cost.

To make the accelerometer axes remain oriented, even under vehicle vibrations, the gimbal structure and its bearings must be stiff. However, to reduce the power requirement, the bearings must also have as little friction as possible. As a result, the gimbal structure is an elaborate, precisely made, expensive, mechanism.

2. Gimbal magnetic.

The construction of mechanized system requires each gimbal must have a pickoff and torquer. The pickoffs measure the intergimbal angles with the resolution of a range 10^6 over a full revolution. When the system is first switched on, the torquers need to provide enough torque to accelerate the platform inertia so that it can gyrocompass and align itself to level and North in a reasonable time. This can produce high magnetic flux, which could affect the sensors as the gimbals move around them.

3. Reliability.

As a nature of mechanical devices, the bearings and slip rings in mechanized system tend to wear. This degrades the alignment and performance.

On the other hand, the advantages of strap-down system are:

1. Simple structure and low cost.

Strap-down systems are lighter, simpler, cheaper, and easily configured for odd-shape spaces. Because it is only necessary to mount the sensors so that their sensing axes point in known directions in the vehicle, they can be placed where the space is available.

2. Ruggedness.

Because the strap-down system has simpler and lighter structure, it has better resistance to shock and vibration.

3. Reliability.

The construction of strap-down system does not have gimbal magnetics, slip rings, and bearings. The electronics that replace them are inherently more reliable.

The disadvantages of strap-down system are:

1. Alignment.

Because the strap-down systems are directly mounted at the vehicle platform, they are difficult to align.

2. Sensor calibration.

As with alignment, the immobility of the sensors means they cannot be calibrated in the system. Therefore, they must be stable, adding complexity in sensor design.

3. Motion-induced errors.

The body motions induce unique sensors errors, which can be compensated to some degree.

4. Accelerometer errors.

In strap-down system, accelerometers may be subjected to components of gravitational acceleration as the vehicle rolls and pitches. This reduces the accuracy of the vehicle acceleration measurement and introduces cross-axis errors.

5. The strap-down computer.

The computer is not needed in mechanized system. This computer must be fast enough to do all the strap-down calculation in a few milliseconds.

3.2 INS Mechanization Model

If the inertial equations, (3.1) - (3.3) are implemented to find the position of the vehicle, it will result in nonlinear equations. It is known that nonlinear equations are more difficult to solve than the linear ones. The linear equation is preferred, beside of it is easier to manipulate, also because the theory of linear system is much more complete and practical than nonlinear. To solve the nonlinear problem, one typical approach is to linearize the equation about some nominal point, which results a perturbation or error model.

In this section, the mechanization model for the INS is derived. This mechanization model, in fact, is an error model. This model can be implemented when the INS measurements are used together with GPS measurement to solve the navigation problems. The error model derived here is only for the INS model, with nine states, three states for every axis. When this error model is implemented with GPS, the states have to be augmented, which is discussed in chapter 6.

Figure 3.2 until Figure 3.4 show the block diagram for the error model for each axis. The x -axis points east, the y -axis points north, and the z -axis denotes the altitude.

The variables used in these figures are:

ω_x = angular velocity of the gyroscope about x -axis.

ω_y = angular velocity of the gyroscope about y -axis.

R = radius of the earth.

g = gravitational field of the earth.

Figure 3.2 - East axis error model.

The error model equations for east axis are:

$$\frac{d^2 \Delta x}{dt^2} = a_x - g\phi_y \quad (3.4)$$

$$\frac{d\phi_y}{dt} = \frac{1}{R} \frac{d\Delta x}{dt} + \omega_x \phi_z + \varepsilon_y. \quad (3.5)$$

Figure 3.3 - North axis error model.

The error model equations for north axis are:

$$\frac{d^2 \Delta y}{dt^2} = a_y + g\phi_x \quad (3.6)$$

$$\frac{d\phi_x}{dt} = -\frac{1}{R} \frac{d\Delta y}{dt} - \omega_y \phi_z + \varepsilon_x. \quad (3.7)$$

Figure 3.4 - Altitude axis error model.

For the altitude axis, the equations for the error model are:

$$\frac{d^2 \Delta z}{dt^2} = a_z + 2 \frac{g}{R} \Delta z \quad (3.8)$$

$$\frac{d\phi_z}{dt} = \varepsilon_z. \quad (3.9)$$

3.3 INS Error Considerations

Although the physical laws describing the dynamics of an inertial system are theoretically exact, several error sources can limit its accuracy. These error sources have to be considered prior to analysis to establish the maximum permissible component tolerances, and also to simplify the mechanization equations. Simplification of mechanization equations can reduce the computing complexity. The following are some of the known major error sources:

1. Gyro drift errors caused by temperature variation, acceleration, magnetic fields, and vibration.

Ideally, the gyro can maintain the fixed direction relative to inertial space without any external torque. However, the factors described above can drift the gyro from its desired position. This drift sometimes is unrepeatable and unpredictable.

2. Gyro scale factor errors including nonlinearity and asymmetry.
3. Accelerometer bias errors caused by variations in temperature or by rectification of vibration inputs.
4. Accelerometer scale factor errors, including nonlinearity and asymmetry.

5. Sensor assembly error such as nonorthogonality of the gyro and accelerometer input axes, transmission of vibration inputs through the shock mounts, and thermal gradients that affect instrument performance and alignment.
6. Computational errors due to round off, truncation, readout accuracy, as well approximations inherent in the algorithms.
7. Initial condition errors.

Chapter 4

Global Positioning System

The history of Global Positioning System (GPS) began in the beginning of 1960s, when several U.S. government organizations, including the military, the National Aeronautics and Space Administration (NASA), and the Department of Transportation (DOT) were interested in developing satellite systems for position determination [20]. The satellites systems proposed should have the following characteristics: global coverage, continuous/all weather operation, ability to serve high-dynamic platform, and high accuracy. Beginning from this time, several systems have been developed, especially by the military, to meet those criteria.

In 1969, the Office of Secretary Defense (OSD) of the U.S government established the Defense Navigation Satellite System (DNSS) program. This program is intended to consolidate the independent development efforts of each military service to form a single joint-use system. From this effort, the system concept of NAVSTAR (Navigation Satellite Timing and Ranging) GPS was formed by the GPS Joint Program Office (JPO). Later, the system is generally referred to as simply “GPS”.

Presently, GPS is fully operational and meets the criteria established in the 1960s for an optimum positioning system. The system provides accurate, continuous,

worldwide, three-dimensional position and velocity information to users with the appropriate receiving equipment.

In this chapter, some aspects of GPS are described. The explanations are specifically for the application of multi-sensor fusion.

4.1 GPS Overview

GPS is a satellite navigation system, which is available worldwide. In general, it is comprised of three components: satellite constellation, ground control/monitoring network, and user receiving equipment. Those three components, in GPS Joint Program Office terms, are formally known as space, operational control, and user equipment segments, respectively.

The satellite constellation contains the satellites in orbit that provide the ranging signals and data messages to the user receiver equipment. It consists of 24 satellites arranged in six Earth-centered orbital planes with four satellites in each plane. The nominal period of a GPS satellite is one-half of a sidereal day, or 11 hr 58 min. The orbit is close to circular, with radius (i.e., nominal distance from the center of mass of the Earth to the satellite) of about 26,600 km. The orbits are inclined relative to the equator of nominally 55° , and equally spaced about the equator at a 60° separation. To optimize global satellite visibility, the satellites are not evenly spaced within the orbital plane. Using this satellite constellation, a 24-hr global user navigation and time determination capability can be achieved.

The operational control segment (OCS) is responsible for operating the GPS. It monitors satellite health and signal integrity and maintains the orbital configuration of the

satellites. Furthermore, the OCS updates the satellites clock corrections and ephemerides as well as numerous other parameters essential to determining user position, velocity, and time. To achieve these goals, the OCS is comprised of master control station and several monitoring stations distributed around the world. The master control station is located at Falcon Air Force base near Colorado Springs, Colorado. The monitoring stations are located at Colorado Springs, Hawaii, Florida, Ascension Island, Diego Garcia, and Kwajalein. Figure 4.1 shows the operation of the control segment. The monitoring stations continuously track all satellites in view. The data collected by monitoring stations are transmitted to the master control station, where extensive computations are performed to find the up-to-date navigation messages. These messages are then transmitted back to the monitoring stations. The updated messages are stored until a particular satellite is in view, when they are uploaded to the satellites using uplink antenna.

Figure 4.1 - Control segment operations.

The last component is user-receiving equipment. This component is typically referred to as a “GPS receiver”. It processes the signals transmitted from the satellites to determine user position, velocity, and time. Figure 4.2 shows the schematic of GPS receiver components. It consists of antenna, receiver, receiver processor, input/output devices, and power supply. Signals received via antenna are analyzed in receiver processor. The processes performed in the processor are primarily channel signal acquisition, signal tracking, and data collection. In addition, the processor may also perform the position, velocity, and time solution for the receiver measurements.

Figure 4.2 - The schematic of GPS receiver components.

4.2 Triangulation Using Time-of-arrival Measurements

To determine user position, GPS implements the triangulation concept using time-of-arrival (TOA) measurements. This concept is based on the principle that the distance between the source of the signal and the receiver can be calculated by multiplying the speed of the signal and the time-of-arrival. The time-of-arrival is the time needed by the signal to transmit from the source to the receiver. By measuring the time-of-arrival of signals broadcast from multiple sources at known locations, the receiver can determine its position.

Consider one satellite transmitting a ranging signal. The satellite contains an onboard clock, which controls the timing of the ranging signal broadcast. The user receiver also contains a clock to record the time when the ranging signal is achieved. If it is assumed that the clock within the satellite and the receiver is perfectly synchronized with the GPS system time, by noting the time when the signal was received, the satellite-to-user propagation time can be computed. Multiplying the satellite-to-user propagation time with the speed of light yields the satellite-to-user range, R . The user then will be located somewhere on the surface of a sphere centered about the satellite as shown in Figure 4.3.

Figure 4.3 - User located on surface of sphere.

If the measurements from two satellites are taken simultaneously, the user will be also on the surface of a second sphere that is concentric about the second satellite. The user position then will be somewhere on the surface of both spheres. From geometry, this position can be either on the perimeter of the shaded circle, which denotes the plane of intersection of both spheres as shown on Figure 4.4, or at a single point tangent to both spheres if the user is collinear with the satellites. The plane of intersection is always perpendicular to a line connecting both satellites, as shown in Figure 4.5.

Figure 4.4 - User located on perimeter of shaded circle.

Figure 4.5 - Satellites' plane of intersection.

If the measurement process is continued using the third satellite, the user will be collocated on the perimeter of the shaded circle and the surface of the third sphere, concentric about the third satellite. The position will be at one of two points, where the third sphere intersects the shaded circle perimeter as shown in Figure 4.6. The intersection is clearly viewed in Figure 4.7.

If a user is on the Earth's surface, the lower point is apparently the user position. For users that are above the Earth's surface, however, it may not be clear which point to select from those two. An additional step is needed, such as conducting a fourth satellite

measurement, or by calculating the speed of the user. The speed of the user is usually unreasonable if that point is not user position.

Figure 4.6 - User located at one of two points on shaded circle.

Figure 4.7 - User located at one of two points on circle perimeter.

4.3 Position Determination Using Pseudorandom Noise (PRN) Codes

GPS provides two services: the Standard Positioning Service (SPS) and the Precise Positioning Service (PPS). The SPS is designated for the civil community,

whereas the PPS is only available for U.S. authorized military and select government agency users.

As explained in previous section, to determine user position, GPS uses time-of-arrival of signals broadcast by satellites. The signals, which contain two types of ranging codes and navigation data, are broadcast on two carrier frequencies, called L1 (1575.42 MHz) and L2 (1227.6 MHz). The ranging codes, which are pseudorandom noise (PRN), consist of a “short” coarse/acquisition (C/A)-code (1.023 MHz) and a “long” precision (P)-code (10.23 MHz). The C/A-code, which is less accurate, has a 1-msec period and repeats constantly. It is used as initial acquisition and code-signal alignment, and available for all users. The SPS is based on this code. The P-code has 7-day period and repeats every midnight Saturday/Sunday. The P-code can be encrypted, which is denoted as Y-code. The Y-code is accessible only to PPS users through cryptography.

Figure 4.8 shows the vector representation of the relation between user position, satellite, and the Earth. We wish to determine vector $\mathbf{u}$, which represents the user position relative to the ECEF coordinate system origin. In three dimensions, this position is denoted by coordinates x_u, y_u, z_u . Vector $\mathbf{s}$ represents position of the satellite with respect to the coordinate origin. It can be computed using ephemeris data broadcast by the satellite. Vector $\mathbf{r}$ represents position of the satellite with respect to user position. This vector can be related to the other by equation:

$$\mathbf{r} = \mathbf{s} - \mathbf{u} . \quad (4.1)$$

If we take r represents the magnitude of $\mathbf{r}$, the value will be:

$$r = |\mathbf{s} - \mathbf{u}|. \quad (4.2)$$

The distance r can be computed by measuring propagation time required for a signal broadcast by satellite to reach the user receiver antenna. The signal can be C/A-code or P-code as explained before.

Figure 4.8 - User position vector representation.

Figure 4.9 shows the measurement process of determining propagation time. A specific code phase is transmitted from satellite at time t_1 . This code arrives at the receiver at time t_2 . The propagation time will be $\Delta t = t_2 - t_1$. To find the propagation time practically, within the receiver, an identical code ranging signal is generated. This signal is then shifted in time until it achieves correlation with the satellite-transmitted ranging code. If the satellite clock and the receiver clock are perfectly synchronized, as shown in Figure 4.9, the amount of time needed to shift the signal will be the propagation

time, Δt . Multiplying this propagation time by the speed of light, the geometric (vector $\mathbf{r}$ in Figure 4.8) satellite-to-user distance can be computed.

Figure 4.9 - Replica code is used to determine satellite code transmission time.

In real application, the satellite and receiver clock will generally have a bias error from GPS system time. Thus, the range determined by the correlation process is not truly the geometric range. This range is denoted by pseudorange ρ , as it contains (1) the geometric satellite-to-user range, (2) an offset caused by the difference between user clock and GPS system time, and (3) an offset caused by the difference between satellite clock and GPS system time. This timing relationship is shown in Figure 4.10. The notations use in this figure are:

T_s = GPS system time at which the satellite transmits the signal.

T_u = GPS system time at which the signal reach the user receiver.

t_s = The offset of satellite clock from GPS system time.

t_u = The offset of user receiver clock from GPS system time.

$T_s + t_s$ = Satellite clock reading when the satellite transmits the signal.

$T_u + t_u$ = User receiver clock reading when the signal reaches the user receiver.

Figure 4.10 - Relationship of range measurement and time.

From the figure, the geometric range is:

$$r = c(T_u - T_s) = c\Delta T . \quad (4.3)$$

where:

c = speed of light.

The pseudorange is calculated as:

$$\rho = c[(T_u + t_u) - (T_s + t_s)] , \quad (4.4)$$

which, after multiplication, (4.4) can be simplified as:

$$\rho = r + c(t_u - t_s) . \quad (4.5)$$

As explained before, pseudorange, as in (4.5), consists of geometric range, satellite clock offset, and user receiver clock offset. The corrections for satellite clock

offset are determined by the ground monitoring network. The corrections are then transmitted to the satellite for rebroadcast to the users in navigation messages. User receiver applies the corrections to synchronize the transmission of each signal to GPS system time. Therefore, in measurement, the satellite clock offset is already compensated. Equation (4.5) then can be written as:

$$\rho = r + ct_u, \quad (4.6)$$

where now, ρ is satellite clock offset compensated pseudorange.

Using notation in (4.2), equation (4.6) can be written as:

$$\rho = |\mathbf{s} - \mathbf{u}| + ct_u. \quad (4.7)$$

In three dimensions space, (4.7) consists of four unknowns, i.e. coordinates of user position, (x_u, y_u, z_u) , and user clock offset, t_u . This needs that the measurements are taken from four different satellites. Using these four measurements, (4.7) can be expanded into the following set of equations in the unknowns x_u, y_u, z_u , and t_u :

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \quad (4.8)$$

$$\rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \quad (4.9)$$

$$\rho_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u \quad (4.10)$$

$$\rho_4 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u \quad (4.11)$$

where x_i, y_i , and z_i denote the i th satellite position in three dimensions.

To solve these nonlinear equations, one can use either a closed form solution, iterative techniques based on linearization, or using Kalman filter.

4.4 Error Sources and Error Budget

In section 4.3, position determination using pseudorandom noise, denoted as pseudorange, was explained. It was introduced that inside the pseudorange, there are geometric range and timing error. So far, the timing error is only considered comes from satellite and user clock offset. There are other sources of error, which contribute to the timing error, such as error due to atmosphere characteristic; error that comes from receiver noise and resolution; error caused by multipath; error that comes from selective availability (SA); and error that comes from receiver hardware characteristic. These errors delay the signal, which comes from the satellite. Total time delay due to all of these errors can be put into one variable, δt , which is:

$$\delta t = \delta t_{\text{atm}} + \delta t_{\text{noise\&res}} + \delta t_{\text{mp}} + \delta t_{\text{SA}} + \delta t_{\text{hw}} \quad (4.12)$$

where:

δt_{atm} = error due to atmosphere characteristic

$\delta t_{\text{noise\&res}}$ = error that comes from receiver noise and resolution

δt_{mp} = error caused by multipath

δt_{SA} = error that comes from selective availability (SA)

δt_{hw} = error that comes from receiver hardware characteristic.

With these errors, the relation between geometric range and pseudorange will be different from the one shown in (4.5). Figure 4.11 shows the new relation between range

measurement and timing. From this figure, the pseudorange, ρ can be related to the others as:

$$\rho = c[(T_u + \delta t + t_u) - (T_s + t_s)] \quad (4.13)$$

which, after multiplication becomes:

$$\rho = r + c(t_u + \delta t - t_s) . \quad (4.14)$$

Figure 4.11 - Range measurement timing relationship.

To find the amount of each time delay in (4.12), one has to model the error. The explanation of the model can be found in [18], [19], [20], [21]. As guidelines for position error analysis, the error contributions are usually implemented in an error budget. Here, total user-equivalent range error (UERE) is comprised of components from each system segment. The error components are root-sum-squared (rss) to form the total system UERE, which has a Gaussian distribution. In Table 4.1, the error budget for GPS PPS user is presented. For SPS user, the error budget is shown in Table 4.2.

Table 4.1 - GPS PPS pseudorange error budget.

Segment Source	Error Source	Standard Deviation (m)
Space	Satellite clock stability	3.0
	Predictability of satellite perturbations	1.0
	Other (thermal radiation, etc.)	0.5
Control	Ephemeris prediction error	4.2
	Other (thruster performance, etc.)	0.9
User	Ionospheric delay	2.3
	Tropospheric delay	2.0
	Receiver noise and resolution	1.5
	Multipath	1.2
	Other (interchannel bias, etc.)	0.5
System UERE	Total (rss)	6.6

Taken from [20]

Table 4.2 - GPS SPS pseudorange error budget.

Segment Source	Error Source	Standard Deviation (m)
Space	Satellite clock stability	3.0
	Predictability of satellite perturbations	1.0
	Selective Availability	32.3
	Other (thermal radiation, etc.)	0.5
Control	Ephemeris prediction error	4.2
	Other (thruster performance, etc.)	0.9
User	Ionospheric delay	5.0
	Tropospheric delay	1.5
	Receiver noise and resolution	1.5
	Multipath	2.5
	Other (interchannel bias, etc.)	0.5
System UERE	Total (rss)	33.3

Taken from [20]

4.5 Satellite Geometry and Dilution of Precision

The quality of pseudorange measurement influences the accuracy of the GPS performance, as discussed in previous sections. The accuracy is also influenced by geometry of satellite distribution, which is known as dilution of precision (DOP).

Consider two satellites in space. A user locates his or her position from the pseudorange measurement taken from those satellites. For simplification, consider the relation between satellites and user position in two-dimensions, as shown in Figure 4.12. With the assumption that the pseudorange measurement is without range error, possible locations of the user from each satellite will be denoted by solid lines. The assumption is hardly achieved in real applications. The range error always occurs in measurement, so the possible locations no longer can be represented as the solid lines. With the error, the possible locations are now represented as the variation in range, shown as dotted lines in the figure. Using the measurement from two satellites, possible user positions will be somewhere in the shaded area in Figure 4.12.

Figure 4.12 - Geometry with low DOP.

Now, consider the two satellites are located close to each other as shown in Figure 4.13. The variation of the range error is the same as in the previous figure. By comparing the shaded area, it is clear that the geometry of the satellite position as in Figure 4.13 gives more error in the computed user position than in Figure 4.12.

Figure 4.13 - Geometry with high DOP.

This geometry factor is represented as DOP parameter in the GPS system. It is defined as geometry factors that relate parameters of the user position and time bias errors to the pseudorange errors. The derivation of the parameter begins with the linearization of the pseudorange equation given in (4.7).

In determination of the user position process, the more usable satellites that are available will result higher accuracy. In this case, the DOP parameter is calculated from all available satellites. However, the DOP criteria can also be used to find the four best satellites out of a geometric constellation consisting of more than four satellites. This gives advantages particularly if only a four-channel receiver is available.

For four satellites constellation, the DOP parameter can be derived from (4.8) - (4.11). If we denote the approximation of user position as $(\hat{x}_u, \hat{y}_u, \hat{z}_u)$, by expanding equations (4.8) - (4.11) in a Taylor series about approximate position, we can find the offset of user position, $(\Delta x_u, \Delta y_u, \Delta z_u)$, from the true position (x_u, y_u, z_u) .

If we write a single pseudorange as:

$$\begin{aligned}\rho_i &= \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_u \\ &= f(x_u, y_u, z_u, t_u),\end{aligned}\tag{4.15}$$

an approximate pseudorange can be calculated as:

$$\begin{aligned}\hat{\rho}_i &= \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2} + c\hat{t}_u \\ &= f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u).\end{aligned}\tag{4.16}$$

The true user position and receiver clock offset are defined as:

$$\begin{aligned}x_u &= \hat{x}_u + \Delta x_u \\ y_u &= \hat{y}_u + \Delta y_u \\ z_u &= \hat{z}_u + \Delta z_u \\ t_u &= \hat{t}_u + \Delta t_u.\end{aligned}\tag{4.17}$$

Therefore, we can write

$$f(x_u, y_u, z_u, t_u) = f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, \hat{t}_u + \Delta t_u).\tag{4.18}$$

The last equation can be expanded about the approximate user position and associated receiver clock offset $(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)$ using a Taylor series. For expansion until the first-order partial derivatives, the series can be written as:

$$\begin{aligned}
f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, \hat{t}_u + \Delta t_u) &= f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u) + \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{x}_u} \Delta x_u &+ \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{y}_u} \Delta y_u + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{z}_u} \Delta z_u + \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{t}_u} \Delta t_u &+ \dots.
\end{aligned} \tag{4.19}$$

The partial derivatives parts of (4.19) are evaluated as:

$$\begin{aligned}
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{x}_u} \Delta x_u &= -\frac{x_i - \hat{x}_u}{\hat{r}_i} \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{y}_u} \Delta y_u &= -\frac{y_i - \hat{y}_u}{\hat{r}_i} \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{z}_u} \Delta z_u &= -\frac{z_i - \hat{z}_u}{\hat{r}_i} \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{t}_u} \Delta t_u &= c,
\end{aligned} \tag{4.20}$$

where:

$$\hat{r}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2}. \tag{4.21}$$

Substituting (4.16) and (4.20) into (4.19), we get:

$$\rho_i - \hat{\rho}_i = -\frac{x_i - \hat{x}_u}{\hat{r}_i} \Delta x_u - \frac{y_i - \hat{y}_u}{\hat{r}_i} \Delta y_u - \frac{z_i - \hat{z}_u}{\hat{r}_i} \Delta z_u + c \Delta t_u. \tag{4.22}$$

Equation (4.22) can be written in matrix form as:

$$\begin{bmatrix} \Delta \rho_1 \\ \Delta \rho_2 \\ \Delta \rho_3 \\ \Delta \rho_4 \end{bmatrix} = \begin{bmatrix} a_{x1} & a_{y1} & a_{z1} & 1 \\ a_{x2} & a_{y2} & a_{z2} & 1 \\ a_{x3} & a_{y3} & a_{z3} & 1 \\ a_{x4} & a_{y4} & a_{z4} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ c \Delta t_u \end{bmatrix}, \tag{4.23}$$

where:

$$\begin{aligned} a_{xi} &= -\frac{x_i - \hat{x}_u}{\hat{r}_i} \\ a_{yi} &= -\frac{y_i - \hat{y}_u}{\hat{r}_i} \\ a_{zi} &= -\frac{z_i - \hat{z}_u}{\hat{r}_i} \end{aligned} \quad (4.24)$$

are often called direction cosines in navigation geometry.

Equation (4.23) can be written in compact form as:

$$\Delta \mathbf{p} = \mathbf{H} \Delta \mathbf{x}. \quad (4.25)$$

Geometric dilution of precision (GDOP) is defined as:

$$\text{GDOP} = \sqrt{\text{trace}(\mathbf{H}^T \mathbf{H})^{-1}}, \quad (4.26)$$

where $\text{trace}[\cdot]$ indicates the sum of diagonal elements of $[\cdot]$.

Beside GDOP, DOP factor can also be represented as vertical dilution of precision (VDOP), horizontal position of dilution (HDOP), position dilution of precision (PDOP), or time dilution of precision (TDOP). PDOP is computed by deleting the fourth diagonal in the trace of (4.26). TDOP is computed from square root of the fourth diagonal of (4.26). To compute VDOP and HDOP, equation (4.23), which is in ECEF coordinate system, must be transformed into local coordinate system using the transformation matrix derived in chapter 2. Once it is transformed, the VDOP is calculated as:

$$\text{VDOP} = \sigma_h, \quad (4.27)$$

and the HDOP as:

$$\text{HDOP} = \sqrt{\sigma_h^2 + \sigma_e^2}, \quad (4.28)$$

where σ_h , σ_n , and σ_e are the elements of the result of transformation of $\text{trace}(\mathbf{H}^T \mathbf{H})^{-1}$ from ECEF coordinate system into local coordinate system.

Chapter 5

Kalman Filter

One problem that occurs in guidance, navigation, and control system is how to estimate the current state of the vehicle, which can be position, velocity, or angles, so as the input for controller can be determined correctly by guidance algorithm. Figure 5.1 shows common block diagram for guidance, navigation, and control (GNC) system. To drive the vehicle to follow designed path, the guidance algorithm gives an input to the controller, which is then transformed into vehicle motion by actuator on the vehicle.

Figure 5.1 - Guidance, navigation, and control block diagram [1].

The movement of the vehicle indicates the change in the state variables of the system. To find the next correct input for the controller, the guidance algorithm needs to know the current states of the system. This can be found by using navigation system, where the states of the system are measured by using several types of sensor. However, in general cases, this method is not always possible. The state variables are usually “hidden variables,” in the sense that they cannot generally be measured directly. Therefore, the

state estimation process is needed. In the later method, the navigation system consists of states estimation algorithm, in which the output of this algorithm can be used as the input of guidance system.

Kalman filter is commonly used as states estimation algorithm. It estimates the state variables optimally by employing recursive algorithm of a linear dynamic system perturbed by Gaussian white noise [26].

When using multi sensors to measure the state variables, a special method has to be used to combine the results to get one optimal measurement. The commonly used method is fusing those results by using Kalman Filter [6], [27], [28], and [29].

This chapter will describe Kalman filter algorithm, especially when it is used in the GPS aided inertial navigation system.

5.1 The Concept of Kalman Filter

Kalman filter is a data processing algorithm, which uses recursive method to find the optimal estimation of current value of the variables of interest. It was first introduced in 1960 by R.E. Kalman in his paper “A New Approach to Linear Filtering and Prediction Problems [30].”

In its algorithm, Kalman filter processes all information that can be provided to it, regardless of their precision, to find the optimal estimation of the variables of interest. In addition to the combination of all available information, to estimate optimally, Kalman filter uses (1) knowledge of the system and measurement device dynamics, (2) statistical description of the system noises, measurement errors, uncertainty in the dynamic model, and (3) any available information about initial conditions of the variables of interest [31].

The algorithm in Kalman filter is called recursive because it uses the result of the previous step to aid in obtaining the desired result for current step. It does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken. This property is vital for practical implementation.

In GNC problems, Kalman filter is placed as a sub system in navigation algorithm. Figure 5.2 shows a typical situation in which a Kalman filter could be used in GNC problems. The diagram inside the dashed line indicates a dynamic system to be controlled, which can be a vehicle, a robot, an aircraft, etc. It is apparent that the state of the system is hidden inside the dynamic system. The system is driven by known control inputs, and measuring devices provide the value of certain pertinent quantities (observed measurements). The information available to solve the GNC problems is only the knowledge of these system inputs and outputs.

The nature of measuring devices makes the observed measurements will be corrupted to some degree by noises, biases, and device inaccuracies, and therefore, a means of extracting valuable information from a noisy signal is needed. There may also be a number of different measuring devices used in dynamic system. Each device has a particular dynamics and error characteristic, and each device can provide some information about a particular variable. It would be desirable to combine the outputs of these measuring devices in a systematic and optimal manner. The need for a filter now becomes apparent. A Kalman filter combines all available measurement data, plus prior knowledge about the system and measuring devices, to produce an estimate of the desired variables in such a manner that the error is minimized statistically.

Figure 5.2 - Typical Kalman filter application [31].

There are several characteristics that can be taken as criteria for optimal estimation. For Bayesian approach, where probabilities are used to show the existence of uncertainty, the possible choices would be the *mean*, the *mode*, and the *median*. The mean is the center of probability density function, the mode is the value of a scalar quantity x that has the highest probability, locating at the peak of the density, and the median is the value of a scalar quantity x such that half of the probability weight lies to the left and half to the right of it. If a Kalman filter is performed in a *linear* model, and in which system and measurement noises are *white* and *Gaussian*, the mean, mode, and median will be all coincide. In this case, therefore a unique “best” estimate of the value of a scalar quantity x is achieved.

As explained before, there are three basic assumptions for the application of Kalman filter, which are linear, white, and Gaussian. Although these assumptions seem

unrealistic, some methods and justification can be taken so that the Kalman filter is still can be applied.

Linear system is a system, which can be described by linear differential equations. A linear differential equation is an n th-order ordinary differential equation, in which each term of the equation contains at most only first power of the dependent variable or its derivatives [32]. A linear system model is preferred because it is easier to manipulate with engineering tools. The theory of linear system is also much more complete and practical than nonlinear. Although most of real system is nonlinear, a linear system model is justifiable for a number of reasons. When the nonlinearities do exist, the typical engineering approach is to linearize about some nominal point or trajectory, achieving a perturbation model or error model.

The white measurement noise means that the noise value is not correlated in time. This means that knowing the value of the noise in this time in no way can be used to predict the value at any other time. Gaussian property in measurement noise means that at any single point in time, the probability density function takes on the shape of a normal bell-shaped curve. These two assumptions make the mathematics of Kalman filter is tractable [31]. It is also easily simulated on a digital computer [33]. When the measurement noise is not really white and Gaussian, the adaptation scheme can be applied into Kalman filter, therefore the requirements of the Kalman filter are still achieved [6], [27], [29].

5.2 The Discrete Kalman Filter

In this section, the derivation of discrete Kalman filter is explained. The state-space representation is used as the basic model.

We begin by considering a dynamic system, which can be modeled by the following equations:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \quad (5.1)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (5.2)$$

These equations can be depicted as shown in Figure 5.3.

Figure 5.3 - System and measurement model.

The meaning of the notations used in (5.1) and (5.27) are:

$\mathbf{x}_k = (n \times 1)$ process state vector at time t_k .

$\Phi_k = (n \times n)$ matrix relating $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of forcing function (state transition matrix).

$\mathbf{w}_k = (n \times 1)$ system noise vector, assumed to be a white (uncorrelated) sequence with known covariance structure.

$\mathbf{z}_k = (m \times 1)$ measurement vector at time t_k .

$\mathbf{H}_k = (m \times n)$ matrix giving the ideal (noiseless) connection between the measurement and the state vector at time t_k (measurement sensitivity matrix).

$\mathbf{v}_k = (m \times 1)$ measurement noise vector, assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence.

The covariance matrices for the $\mathbf{w}_k$ and $\mathbf{v}_k$ vectors are given by

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (5.3)$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (5.4)$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \text{ for all } k \text{ and } i \quad (5.5)$$

It is assumed that the initial state estimate of the process at some point in time t_k is known. This estimate is based on all of the knowledge about the process prior to time t_k . This prior (or *a priori*) estimate will be denoted as $\hat{\mathbf{x}}_k^-$. It is also assumed that the error covariance matrix associated with $\hat{\mathbf{x}}_k^-$ is known. Clearly, if the estimation error is defined as

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \quad (5.6)$$

the associated error covariance matrix is

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]. \quad (5.7)$$

The objective is we try to improve the prior estimate $\hat{\mathbf{x}}_k^-$ by using the knowledge of the new measurement $\mathbf{z}_k$. This can be done by blending the new measurement and the prior estimate using a linear relation:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-). \quad (5.8)$$

The $\mathbf{K}_k$ in (5.27) is the gain for updated estimation process. The correction for prior estimation is achieved by correcting the prior estimation with the multiplication result between the gain and the residual, sometimes referred as innovation process, $(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$. The gain itself has to be calculated to get the optimal updated estimation. Minimum mean-square error can be used as performance criterion when calculating the optimal gain.

To find the optimal gain, we begin by expressing the error covariance matrix for the updated (*a posteriori*) estimate, which is

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]. \quad (5.9)$$

Substituting measurement equation (5.2) into (5.8) and then entering the result into (5.27) yield

$$\begin{aligned} \mathbf{P}_k = E\{ & [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)] \\ & [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)^T] \}. \end{aligned} \quad (5.10)$$

Solving the expectation and noting that $(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$ is the a priori estimation error, which is uncorrelated with the measurement error $\mathbf{v}_k$, we have

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T. \quad (5.11)$$

From the above equation, the optimal estimation is achieved if the individual terms along the major diagonal of $\mathbf{P}_k$ are minimized. This is because those terms represent the estimation error variances for the elements of the state vector being estimated. Let we regroup the terms in (5.27) to yield

$$\mathbf{P}_k = \mathbf{P}_k^- - \underbrace{\mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T}_{\text{Linear in } \mathbf{K}} + \underbrace{\mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T}_{\text{Quadratic in } \mathbf{K}}. \quad (5.12)$$

Now, it can be seen that $\mathbf{P}_k$ is quadratic in $\mathbf{K}_k$. If we make

$$\mathbf{S} \mathbf{S}^T = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k, \quad (5.13)$$

equation (5.12) can be written as

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{S} \mathbf{S}^T \mathbf{K}_k^T. \quad (5.14)$$

We can factorize the above quadratic form, and write $\mathbf{P}_k$ in the form

$$\mathbf{P}_k = \mathbf{P}_k^- + (\mathbf{K}_k \mathbf{S} - \mathbf{A})(\mathbf{K}_k \mathbf{S} - \mathbf{A})^T - \mathbf{A} \mathbf{A}^T, \quad (5.15)$$

where $\mathbf{A}$ does not involve $\mathbf{K}_k$. If equation (5.27) is expanded and compared term by term with (5.14), the following equality must be true:

$$\mathbf{K}_k \mathbf{S} \mathbf{A}^T + \mathbf{A} \mathbf{S}^T \mathbf{K}_k^T = \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T. \quad (5.16)$$

To get the equation (5.15) equivalent to (5.14), the $\mathbf{A}$ in (5.27) should be

$$\mathbf{A} = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{S}^T)^{-1}. \quad (5.17)$$

Return to equation (5.15), it can be seen that only the middle term of this equation involves $\mathbf{K}_k$, and it is the product of a matrix and its transpose, which ensures that all terms along the major diagonal will be nonnegative. To minimize $\mathbf{P}_k$, the best solution is to make the middle term of (5.15) zero, which implies

$$\mathbf{K}_k \mathbf{S} = \mathbf{A}. \quad (5.18)$$

Using (5.17) into (5.27), we get

$$\begin{aligned} \mathbf{K}_k &= \mathbf{A} \mathbf{S}^{-1} \\ &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{S}^T)^{-1} \mathbf{S}^{-1} \\ &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{S} \mathbf{S}^T)^{-1}. \end{aligned} \quad (5.19)$$

Replacing $\mathbf{S} \mathbf{S}^T$ from (5.13), the final expression for the optimum $\mathbf{K}_k$ is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \quad (5.20)$$

The $\mathbf{K}_k$ in (5.27), which minimize the mean square estimation error is called the *Kalman gain*. To calculate the covariance matrix associated with the optimal estimate, we write (5.11) as:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T. \quad (5.21)$$

Substitute $\mathbf{K}_k$ from (5.20) into (5.27) leads to

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \quad (5.22)$$

or

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T \quad (5.23)$$

or

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- . \quad (5.24)$$

The expression of $\mathbf{P}_k$ in (5.22) - (5.27) are valid only for Kalman (optimal) gain, whereas the expression in (5.11) and (5.21) are valid for any gain.

We have derived the optimal estimation for the state $\hat{\mathbf{x}}_k$ and its associated covariance $\mathbf{P}_k$. When we are using these equations, we need $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$. For the initial time, these values can be assumed. For the next step when we want to make optimal estimation of $\hat{\mathbf{x}}_{k+1}$, the values of $\hat{\mathbf{x}}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ have to be computed. The updated estimate of $\hat{\mathbf{x}}_k$ can be calculated using state transition matrix as follows:

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k . \quad (5.25)$$

The error covariance matrix associated with $\hat{\mathbf{x}}_{k+1}^-$ is obtained by first forming the expression for the a priori error

$$\begin{aligned} \mathbf{e}_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- \\ &= (\Phi_k \mathbf{x}_k + \mathbf{w}_k) - \Phi_k \hat{\mathbf{x}}_k \\ &= \Phi_k \mathbf{e}_k + \mathbf{w}_k . \end{aligned} \quad (5.26)$$

Because $\mathbf{w}_k$ and $\mathbf{e}_k$ are uncorrelated, we can write the expression for $\mathbf{P}_{k+1}^-$ as

$$\begin{aligned}\mathbf{P}_{k+1}^- &= E[\mathbf{e}_{k+1}^- \mathbf{e}_{k+1}^{-T}] = E[(\Phi_k \mathbf{e}_k + \mathbf{w}_k)(\Phi_k \mathbf{e}_k + \mathbf{w}_k)^T] \\ &= \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{Q}_k.\end{aligned}\tag{5.27}$$

Equations (5.8), (5.20), (5.24), (5.25), and (5.27) comprise the Kalman filter recursive equations. Figure 5.4 shows the computation steps when these equations are implemented into Kalman filter.

Figure 5.4 - Kalman filter loop.

As can be seen from the figure, Kalman filter is just a computer algorithm for processing discrete measurement data in an optimal fashion. The filtering process begins

with the initial estimate of state variable and its covariance at time t_k , $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{P}}_k^-$. From covariance and the knowledge of the system variables, we calculate the Kalman gain. When the measurement data becomes available, the initial state estimate and its variance are updated. Until this point, we have the optimal estimate of the state and its uncertainty, which is characterized by its variance. The Kalman filter now processes to the next time step. We replace the initial estimate using the time propagation equations. From here, the process is repeated to find the new Kalman gain and the new updated estimation.

Figure 5.5 shows the relation between system model and the Kalman filter. The Kalman filter to be implemented appears outside the dashed-line box. The block diagram inside the box is a mathematical model of the system and measurement process. As can be seen, the Kalman filter is based upon this model.

Figure 5.5 - Block diagram system, measurement model, and discrete-time Kalman filter.

5.3 Linearized and Extended Kalman Filters

As explained in section 5.1, in derivation of Kalman filter equation, the system is assumed to be linear. Many processes in practice are not absolutely linear, but they are not far from it. When the Kalman filter is implemented into nonlinear problems, the common method is to linearize the nonlinear equation about a particular value, therefore

the linear relation can be reached. This usually results a perturbation equation or error equation, which is linear.

There are two methods to linearize the equations. The first one is linearization about a nominal trajectory, which is usually known beforehand. In this method, linearization process does not depend on the measurement data, therefore matrices Φ , H , and K can be precomputed off-line. The resulting filter is referred as *linearized Kalman filter*. The second method is to linearize the equation about a trajectory that is continually updated with the state estimates resulting from the measurements. Using this method, the deviations between the estimated trajectory and the actual trajectory will remain sufficiently small [34]. The drawback of this method is the increase of real-time computation. This is because matrices Φ , H , and K have to be recomputed every time the new estimation is achieved. The filter resulted using this method is called an *extended Kalman filter*.

The derivation of the extended Kalman filter for the application of GPS-aided inertial navigation system is provided in chapter 6.

Chapter 6

Externally Aided Inertial Navigation System

Inertial navigation system (INS) is preferred in navigation because, as already explained in Chapter 3, it is fast, self-contained, non-jammable, and available in all weather. With those properties, it is not surprisingly that it has been used in many modern commercial and military applications.

Inertial navigation system, on the other hand, also exhibits position errors, which can increase unbounded with time. The degradation is due, in part, to errors in initialization of the inertial measurement unit and imperfections in the inertial sensor such as accelerometer bias and gyroscope drift. One way to avoid this problem is to update the inertial system periodically with external source. Specifically, the external source is used to reset the error in the inertial system to the same level of accuracy inherent in the fixing technique. For this reason, external sources of data are routinely used in navigation using inertial navigation system.

There are many of external sources that are usually used with INS. They can use radar, radio signal, or position fixes as the aid. Some examples for external sources are Navstar GPS, Global Orbiting Navigation Satellite System (GLONASS), Tactical Air Navigation (TACAN), Long Range Navigation (Loran), and Omega.

When more than one sources of data available (INS and external sources), one would want to combine the available information in an optimal manner if possible, efficiently providing an estimate of navigation parameters that is best with respect to some criterion. In some of the earlier navigation systems, it is done by resetting the INS data to agree with the external data. This method assumes that the external data is perfect and discards the information previously available in the INS. Newer approach is by implementing Kalman filter to fuse the available data [6], [27], [28], [29]. Using this method, the statistical characteristics of the error in both the external information and the inertial component are used to determine the optimal combination of information.

In this chapter, the implementation of the Kalman filter to fuse INS and external measurement sources is explained. The common methods used to combine the signals when using Kalman filter are first explained. In this explanation, total state space and error state space configuration are discussed. The advantages and drawbacks of each method are also discussed. The derivation of Kalman filter equations for 8-state GPS aided inertial navigation system is then explained.

6.1 Fusing Methods Using Kalman Filter

When implementing Kalman filter in inertial navigation problems (or any other applications), there are two aspects that have to be considered: *total state space formulation* and *error state space formulation*. These are also sometimes referred as *direct* and *indirect implementation*. For error state space formulation, there are also two possibilities of implementation, *feedforward* and *feedback* configuration. In this section, these aspects will be explained.

6.1.1 Total state space configuration

In total state space (direct) configuration, total states such as vehicle position and velocity are used as the state variables in the filter, and the measurements are INS accelerometer outputs and external source signals. Figure 6.1 shows the diagram of this configuration. The Kalman filter is in the loop of inertial system. The INS accelerometer signals and the external source data are both fed into the filter. This results not only the desired navigation information, but also the appropriate commands to the gimbal torquer motors to maintain the platform aligned with the chosen coordinate frame. The advantage of this configuration is the available information is weighted optimally rather than operated upon a fixed gain. This optimal time-varying gain can provide substantial improvement in performance over the classical approaches, where the gain is fixed, or updated in a particular time range.

Figure 6.1 - Total state space (direct) Kalman filter.

There is very serious drawback in this configuration. Because the Kalman filter is placed in the INS control loop, and it uses the total state space representation, the Kalman filter has to maintain the vehicle motion as well as attempt to suppress noisy and erroneous data. Because the INS is usually working in high dynamic environment, the filter needs a very fast sample rate and has to perform all computations within a short

sample period. The Kalman filter itself is usually allocated only a small portion of the capabilities of a central processor, and run at a lower priority than the INS algorithms, which is more critical in this case. Therefore, it is almost impossible to implement this configuration in high dynamic environment.

Using total state space implementation usually results in nonlinear equations. Although the linearization method is available, in most problems total state space equations cannot be assumed as linear, so this deviates the requirement of Kalman filter. Another drawback is if the filter should happen to fail, the entire navigation system fails. In this case, the inertial system cannot operate without the filter.

As a result of these considerations, total state space configuration is restricted in low dynamic application. The author implemented this configuration when fusing odometry signal and sonar measurement in two-wheel robot, which is one example of low dynamic system [35], [36].

6.1.2 Error state space configuration

In error state space (indirect) configuration, the difference between INS and external source data is used as the input for Kalman filter. In this case, the filter tries to estimate the errors in navigation and attitude information. The advantage of this configuration is the filter does not need to work at the same frequency as the INS. The frequency at which the filter works is the same as the frequency of the external source data, which is usually lower than the frequency of INS.

When calculate the input for the Kalman filter, the INS measurements are usually used as a reference. This is because the measurements from INS are always available to

use. There is also another advantage of using this method. Because the reference is the INS measurements, it can be combined with the measurement from a wide variety of external source. This is important, because the combination of aiding sensors may vary during a mission.

In general, there are two different implementations of error state space configuration: feedforward and feedback. These implementations are based on how the estimated errors resulted from the filter are combined with the inertial system.

Figure 6.2 depicts the indirect feedforward configuration. From this figure, it can be seen that the filter compares the two sets of data and uses the result to estimate the errors in the inertial system. The estimated errors are then subtracted from the inertial data, which results the optimal estimation of position, velocity, and attitude. The operation of the INS itself does not depend on the filter. If the filter fails, the unaltered INS information is still available. However, there is still a drawback in this configuration. The output of the INS measurement is still free to drift with unbounded errors. When this output is compared with the output of external source, the errors resulted will also grow unbounded. This will deviate the requirement of Kalman filter.

Figure 6.2 - Indirect feedforward Kalman filter.

The second configuration in indirect system is the indirect feedback. This configuration is shown in Figure 6.3. In this configuration, the estimated inertial errors resulted by the filter is used to correct the inertial system. Using this method, the inertial error are not allowed to grow unchecked, thus the adequacy of a linear model is enhanced. Because the filter operates in low sample rate, if it fails, it can be detected more easily and the corrections to the INS can be removed before much performance deterioration is caused.

Figure 6.3 - Indirect feedback Kalman filter.

6.2 GPS Aided Inertial Navigation System

One external measurement source that is most widely used in the integration with the INS is the measurement from GPS. With this integration, the characteristics of GPS measurement will bound the error introduced by INS. In this section, this integration is explained. The Kalman filter equations for 8-state integration are also derived.

Many configurations can be used to integrate the GPS and INS measurements. Beside the basic configurations explained in the previous section, the other configurations can be found from [20] and [37]. In this experiment, the configuration used to integrate the GPS and INS measurements is shown in Figure 6.4. For identification purpose, this configuration will be called as *indirect half feedback Kalman filter configuration*. In this

configuration, the INS measurements are used to provide the navigation variables for use in the Kalman filter transition and observation matrix equations. The filter itself, in this case, is the extended Kalman filter (EKF). This is because the optimal estimates of the state vector are used as reference values for the linearization process. In the following, the derivation of the Kalman filter equations for this model with eight state space variables is explained.

Figure 6.4 - GPS aided inertial navigation system configuration.

For 8-state Kalman filter, the model for the INS now can be depicted as shown in Figure 6.5 - Figure 6.7. For East direction, the equation for the INS model is:

$$\frac{d^2 \Delta x}{dt^2} = a_x \quad (6.1)$$

If we make:

x_1 = East position error

x_2 = East velocity error,

equations (6.1) can be written in state space form as follows:

$$\dot{x}_1 = x_2 \quad (6.2)$$

$$\dot{x}_2 = a_x. \quad (6.3)$$

Figure 6.5 - East position error model.

For North direction, the equation for the INS model is:

$$\frac{d^2 \Delta y}{dt^2} = a_y \quad (6.4)$$

If we make:

$$x_3 = \text{North position error}$$

$$x_4 = \text{North velocity error,}$$

the above equation can be written in state space form as:

$$\dot{x}_3 = x_4 \quad (6.5)$$

$$\dot{x}_4 = a_y. \quad (6.6)$$

Figure 6.6 - North position error model.

For vertical direction, the equation for the INS model is:

$$\frac{d^2 \Delta z}{dt^2} = a_z. \quad (6.7)$$

If we make:

x_5 = vertical position error

x_6 = vertical velocity error,

the equation (6.7) can be written in state space form as:

$$\dot{x}_5 = x_6 \quad (6.8)$$

$$\dot{x}_6 = a_z \quad (6.9)$$

Figure 6.7 - Vertical position error model.

The clock is modeled as two-state [38]. The model is shown in Figure 6.8. The equations for this model are:

$$\frac{dB}{dt} = D + u_B \quad (6.10)$$

$$\frac{dD}{dt} = u_D. \quad (6.11)$$

If we make:

x_7 = clock bias

x_8 = clock drift,

equation (6.10) and (6.11) can be written in state space form as:

$$\dot{x}_7 = x_8 + u_B \quad (6.12)$$

$$\dot{x}_8 = u_D \quad (6.13)$$

Figure 6.8 - Clock model.

Now, the state space equations can be written in matrix form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 \\ w_2 \\ 0 \\ w_4 \\ 0 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix} \quad (6.14)$$

where, we have introduced w 's to replace the notations for accelerations and clock noises in the previous equations.

The above equation can be simplified as:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t) . \quad (6.15)$$

This is the dynamic equation of the GPS aided inertial navigation system. The last term in the above equation is the noise introduced by INS measurements and receiver

clock. It is assumed that the noise is purely white noise, zero mean, and Gaussian with covariance Q .

Equation (6.15) is continues in time. The Kalman filter is mostly implemented in digital computer. Therefore, the above equation has to be transferred into discrete form. Discretization can be started by writing the solution of (6.15) as follow:

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)} \mathbf{w}(\tau) d\tau. \quad (6.16)$$

If we make $t_0 = kT$ and $t = (k+1)T$ for an integer k , and defining the sampled state as $x_k = x(kT)$, we can write (6.16) as

$$\mathbf{x}_{k+1} = e^{\mathbf{F}T} \mathbf{x}_k + \int_{kT}^{(k+1)T} e^{\mathbf{F}[(k+1)T-\tau]} \mathbf{w}(\tau) d\tau. \quad (6.17)$$

If we define

$$\Phi_k = e^{\mathbf{F}T} \quad (6.18)$$

and

$$\mathbf{w}_k = \int_{kT}^{(k+1)T} e^{\mathbf{F}[(k+1)T-\tau]} \mathbf{w}(\tau) d\tau, \quad (6.19)$$

we find the discrete form of the dynamic equation as

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k. \quad (6.20)$$

The covariance of $\mathbf{w}_k$ in discrete form ($\mathbf{Q}_d$) can be found by writing

$$\begin{aligned}
\mathbf{Q}_d &= \overline{\mathbf{w}_k \mathbf{w}_k^T} \\
&= \iint_T e^{F[(k+1)T-\tau]} \overline{w(\tau)w(\sigma)} e^{F^T[(k+1)T-\tau]} d\tau d\sigma \\
&= \iint_T e^{F[(k+1)T-\tau]} \mathbf{Q} e^{F^T[(k+1)T-\tau]} d\tau d\sigma
\end{aligned}$$

By changing variables twice ($\lambda = \tau - kT$ and then $\tau = T - \lambda$) we obtain

$$\mathbf{Q}_d = \int_0^T e^{F\tau} \mathbf{Q} e^{F^T\tau} d\tau \quad (6.21)$$

The observation for Kalman filter is the difference between pseudorange as measured by GPS and the predicted pseudorange. It can be derived from equation (4.23) in Chapter 4. This equation is

$$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \\ \Delta\rho_4 \end{bmatrix} = \begin{bmatrix} a_{x1} & a_{y1} & a_{z1} & 1 \\ a_{x2} & a_{y2} & a_{z2} & 1 \\ a_{x3} & a_{y3} & a_{z3} & 1 \\ a_{x4} & a_{y4} & a_{z4} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ c\Delta t_u \end{bmatrix}. \quad (6.22)$$

In this equation, it is assumed that the pseudorange is measured using four GPS receiver channels. The vector on the right side of (6.22) is the user position error from true position. It is defined in ECEF Cartesian coordinate. We define the state of the Kalman filter in this section in local coordinate frame. Therefore, this vector has to be transformed into local coordinate frame using the transformation matrix derived in Chapter 2. Clearly, we can use equation (2.26) to transform this vector from ECEF Cartesian coordinate into local coordinate frame. Once it is transformed, we can generate the observation equation for 8-state Kalman filter as

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} a_{x1} & 0 & a_{y1} & 0 & a_{z1} & 0 & 1 & 0 \\ a_{x2} & 0 & a_{y2} & 0 & a_{z2} & 0 & 1 & 0 \\ a_{x3} & 0 & a_{y3} & 0 & a_{z3} & 0 & 1 & 0 \\ a_{x4} & 0 & a_{y4} & 0 & a_{z4} & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (6.23)$$

The last term in (6.23) is the measurement noise. It is also assumed as zero mean Gaussian white noise with covariance $\mathbf{R}$.

Measurement equation (6.23) is already in discrete form, so it can be implemented directly into discrete Kalman filter. Writing this equation in matrix form result

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (6.24)$$

Therefore, we conclude the equations for the GPS aided inertial navigation system as follow:

The dynamic process:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k. \quad (6.25)$$

Measurement process:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (6.26)$$

The $\mathbf{\Phi}_k$ matrix in (6.25) is called state transition matrix. It represents the transition of the state from time t_k to the time t_{k+1} . It is calculated as an exponential as

shown in (6.18). If the matrix $\mathbf{F}$ is constant, and the period of the sampled data is T , the state transition matrix can be approximated by equation:

$$\Phi_k = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{(\mathbf{F}T)^2}{2!} + \dots \quad (6.27)$$

Implementing (6.27) only until first order, the state transition matrix in (6.25) can be written as

$$\Phi_k = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.28)$$

The process noise, equation (6.19), after integration will be:

$$\mathbf{w}_k = \begin{bmatrix} Tw_1 + \frac{1}{2}T^2w_2 \\ Tw_2 \\ Tw_3 + \frac{1}{2}T^2w_4 \\ Tw_4 \\ Tw_5 + \frac{1}{2}T^2w_6 \\ Tw_6 \\ Tw_7 + \frac{1}{2}T^2w_8 \\ Tw_8 \end{bmatrix}. \quad (6.29)$$

One important aspect when implementing the Kalman filter is the derivation of dynamic process noise covariance matrix. It shows the uncertainty of the model compared to the actual system. When many variables are ignored in the modeling

process, the value of process noise covariance should be greater than the accurate modeling. When the process varies with time, the difficulties in modeling the process noise covariance are also increase. Many works have shown that inaccurately designed process covariance matrix can result the instability of the Kalman filter [6], [27], [29], [17].

Equation (6.21) provides the way to calculate the process noise covariance. In this experiment, the process noise covariance matrix is calculated as explained in Brown [38], which has been derived based on (6.21). The covariance matrix is

$$\mathbf{Q}_d = \begin{bmatrix} \frac{S_p T^3}{3} & \frac{S_p T^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{S_p T^2}{2} & S_p T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{S_p T^3}{3} & \frac{S_p T^2}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{S_p T^2}{2} & S_p T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{S_v T^3}{3} & \frac{S_v T^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{S_v T^2}{2} & S_v T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_f T + \frac{S_g T^3}{3} & \frac{S_g T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{S_g T^2}{2} & S_g T \end{bmatrix} \quad (6.30)$$

where:

S_p = white noise spectral amplitude in horizontal direction

S_v = white noise spectral amplitude in vertical direction

S_f, S_g = white noise spectral amplitude for receiver clock model.

This derivation completes the chapter about integration of INS and external measurement. In the next chapter, the simulation experiment for GPS aided inertial navigation system is conducted.

Chapter 7

Simulation Experiment and Results

In this chapter, the simulation experiment of the GPS aided inertial navigation system is explained. The simulation is conducted based on the real data, which was measured from an aircraft equipped with GPS receiver and inertial system. Two Kalman filter configurations are used: indirect feedforward and indirect half feedback Kalman filter configuration. The simulation is performed on Matlab[®], the technical computing language from MathWorks.

The Kalman filter configurations used in this experiment are explained first. The data for the INS and GPS measurement are then discussed. Before the results of the simulation experiment are displayed, the algorithm used to choose four satellites from a number of visible satellites is explained.

7.1 Kalman Filter Configurations

The Kalman filter configurations used in this experiment are the indirect feedforward configuration as shown in Figure 6.2 and the indirect half feedback configuration as displayed in Figure 6.4. The full indirect feedback configuration as shown in Figure 6.3 cannot be simulated in this experiment because no measurement data are available for this particular configuration.

As explained in Chapter 6, for the GPS-INS measurement integration, the states for the Kalman filter are the error states. These states, in this case, show the deviation of the position and velocity of the aircraft from the planned path. To find the correct position and velocity of the aircraft, these error states are added into the inertial system measurements. How the error states are added into the inertial measurements defines the Kalman filter configuration. In the following, the implementations of those configurations in this experiment are explained.

Figure 7.1 shows the flow diagram for indirect feedforward configuration, and Figure 7.2 shows the flow diagram for indirect half feedback configuration. In both configurations, the Kalman filter begins by generation of state transition matrix Φ , process noise covariance matrix $\mathbf{Q}$, and measurement noise covariance matrix $\mathbf{R}$. The values of these matrices are constant during the simulation process. The formulation for state transition matrix is shown in equation (6.28). For process noise covariance, the equation is displayed in (6.30). Below are the values for the constants used in this equation.

$$S_p = 1 \text{ m}^2/\text{s}^3$$

$$S_v = 1 \text{ m}^2/\text{s}^3$$

$$S_f = 0.4 \cdot 10^{-18} \text{ m}^2/\text{s}$$

$$S_g = 1.58 \cdot 10^{-18} \text{ m}^2/\text{s}^3$$

Using the measurements taken from four satellites, the measurement noise covariance is assumed as:

$$\mathbf{R} = \begin{bmatrix} 225 & 0 & 0 & 0 \\ 0 & 225 & 0 & 0 \\ 0 & 0 & 225 & 0 \\ 0 & 0 & 0 & 225 \end{bmatrix}, \quad (7.1)$$

the unit in (7.1) is m^2 .

Figure 7.1 - Flow diagram for indirect feedforward Kalman filter.

An attention has to be placed in the simulation processes inside the dashed box. The processes inside this box are always varying with time. The satellites selection is the process to choose the combination of four satellites from a number of available satellites. This process will be explained later. The $\mathbf{H}$ matrix is measurement sensitivity matrix.

This matrix relates the observation and the states. Because of the linearization process, the values of this matrix, where its elements are the direction cosine, are always varying. For indirect feedforward configuration, this matrix is calculated based on predefined path, in this case the INS measurement. In the case of indirect half feedback, the values are calculated based on the updated estimation.

Figure 7.2 - Flow diagram indirect half feedback Kalman filter.

The difference between diagrams in Figure 7.1 and Figure 7.2 is in the position used to choose the satellite and to calculate the $\mathbf{H}$ matrix. In indirect half feedback configuration, the error state resulted from the Kalman filter is used to update the INS

measurements, and to calculate the $\mathbf{H}$ matrix. For indirect feedforward, the INS measurements are directly used in the calculation of $\mathbf{H}$ matrix.

7.2 The INS and GPS Data

The data used in this simulation was taken on August 9, 1999. It was recorded from the aircraft equipped with the GPS receiver and inertial system, which flew for almost four hours. Figure 7.3 shows the path followed by the aircraft during the flight, showed in geodetic coordinate system. Figure 7.4 and Figure 7.5 show the longitude and latitude coordinate as function of time.

Figure 7.3 - Path followed by the aircraft during the flight.

The aircraft starts to flight from position with coordinates:

North = 45.3356°

East = -75.6891° .

The velocity profiles are displayed in Figure 7.6 and Figure 7.7, and the accelerometer outputs are displayed in Figure 7.8 until Figure 7.10.

Figure 7.4 - Longitude coordinate followed by the aircraft during the flight.

Figure 7.5 - Latitude coordinate followed by the aircraft during the flight.

Figure 7.6 - Aircraft velocity profile during the flight (East channel).

Figure 7.7 - Aircraft velocity profile during the flight (North channel).

Figure 7.8 - East channel accelerometer output.

Figure 7.9 - North channel accelerometer output.

Figure 7.10 - Vertical channel accelerometer output.

The frequency of the accelerometer outputs is 32 Hz. The statistic properties of the outputs are

East channel acceleration mean value = 0.0013.

East channel acceleration standard deviation = 0.4101.

North channel acceleration mean value = $3.4634 \cdot 10^{-4}$.

North channel acceleration standard deviation = 0.4902.

Vertical channel acceleration mean value = -0.0053.

Vertical channel acceleration covariance = 0.6691.

The accelerometer outputs data are also plotted in histogram form, which are shown in Figure 7.11 until Figure 7.13. It can be seen that the properties of the acceleration noise data are near the properties of zero mean Gaussian noise. However, as displayed in Figure 7.8 - Figure 7.10, the accelerometer outputs are influenced by the movement of

the aircraft, therefore the acceleration noise data basically are not purely a zero mean Gaussian white noise.

Figure 7.11 - Distribution of INS measurement (East channel).

Figure 7.12 - Distribution of INS measurement (North channel).

Figure 7.13 - Distribution of INS measurement (Vertical channel).

For GPS measurements, the data consist of the outputs of GPS receiver and the GPS ephemeris data. The outputs of the GPS receiver are recorded in *Receiver Independent Exchange (Rinex)* format version 2. The example of this format data file is included in Appendix A. The following are the observation types included in the Rinex2 format:

L1, L2: phase measurement on L1 and L2 (full cycles).

C1: pseudorange measurement using C/A-code on L1 (meters).

P1: pseudorange measurement using P-code on L2 (Hz).

D1, D2: Doppler frequency on L1 and L2.

The receiver itself is able to receive more than four satellites in view in one particular time. In this experiment, only four satellites will be used for simulation. The algorithm to

choose the best satellite combination will be explained later. The frequency of the GPS measurement data is 10 Hz.

Satellite ephemeris data are recorded in *Standard Product #3 (SP3)* format. The example of this data format is also included in Appendix A. In this data, position of all available satellites is recorded together with satellite clock correction. The reference coordinate used in this data is ECEF Cartesian coordinate, and the time is recorded in GPS time. The unit used in position is km, and for GPS clock correction is microsecond.

Not all the available satellites are visible during the flight. Figure 7.14 shows the satellite visibility chart during the flight. The black block means that the satellite is visible during the time period displayed in the x -axis.

Figure 7.14 - Satellite visibility as function of time.

There is also the aircraft position data measured by standalone GPS. Comparing this measurement with the aircraft position measured by the INS, the position error is

resulted. Figure 7.15 and Figure 7.16 show the position error data in East and North direction, respectively.

Figure 7.15 - East position error between INS and GPS data.

Figure 7.16 - North position error between INS and GPS data.

7.3 Satellite Selection Algorithm

The GPS receiver in the aircraft is able to receive the GPS signal from more than four satellites in view. In this simulation, only four satellites are used as the source of the GPS measurement. When more satellites used as the measurement source, the results will be better, however, it is common that the amount of receiver channels available are restricted, and not all of the satellites are good to be used as the sources of measurements.

Using only four satellites, the problem will be how to choose the best combination of four satellites from a number of available satellites. Dilution of Precision (DOP) discussed in Chapter 4 provides a way to effectively choose the best satellites combination. This is because DOP only relates to the geometry of the satellite position. It does not relate to the measurement. In other word, no noises affect the satellites selection process.

In general, the criterion to choose the best satellites combination is to find the smallest DOP value. In this experiment, Position Dilution of Precision (PDOP) is used as criterion. Figure 7.17 shows the algorithm implemented to choose the smallest PDOP. The algorithm begins with the creation of the combination of four satellites from all satellites in view. The PDOP of each combination is then checked to find the smallest value.

When the simulation is running, the changes in satellite and aircraft position make a change in the value of PDOP. Therefore, the PDOP value needs to be checked in the loop. Figure 7.18 shows the satellite selection algorithm when implemented in the loop of Kalman filter. This is the algorithm inside the satellites selection box in Figure 7.1 and Figure 7.2. The satellites in the combination used in the previous step are checked to find

whether the satellites are still visible or not. If they are still visible, the combination's PDOP value is checked. If not visible, the new satellites combination is created. To reduce the change of satellites in combination, the threshold for PDOP value is implemented. The initial value of the threshold in this experiment is 2. Using the threshold, although the PDOP value of the combination at any particular time is not the smallest one, the combination is still accepted to use in the simulation. When after selects the new combination the PDOP value is greater than the threshold, and no other combinations are possible to lower the PDOP, the threshold is increased by factor of 1.

Figure 7.17 - Flow diagram of satellite selection based on smallest PDOP.

Figure 7.18 - Flow diagram of satellite selection algorithm.

7.4 Results of Simulation Experiment

The simulation experiment has been conducted for 2.5 hours long. Figure 7.19 until Figure 7.35 show the results of this simulation. In one figure, there are three sub figures. The large one is the comparison between the results of the implementation of feedforward and half feedback Kalman filter, and two smaller figures are the plot of the results displayed as separate figures. From the type of linearization method, feedforward configuration uses linearized Kalman filter, and half feedback configuration uses extended Kalman filter.

The results plotted in the figures consist of position and velocity error in East, North, and vertical directions, the covariance for position and velocity error, Kalman gain, and the innovation process. The position and velocity error show the accuracy of the position and velocity of the aircraft as estimated by the Kalman filter. Smaller error indicates higher accuracy. The covariance shows the uncertainty of the estimation. Larger covariance means higher uncertainty. The Kalman gain indicates the weighted factor for the observation. Smaller Kalman gain means that the observation is less used in the estimation, or in other word, the estimation is largely based on the inertial system. Innovation process is to show the characteristic of the signal before entering the Kalman filter.

The satellites combination chosen during the simulation is shown in Table 7.1. In this table, the PDOP value for a particular combination is also displayed. The reason why the change in combination occurs is displayed in the remarks. The change in combination is needed when one or more satellites are not visible anymore, or if the PDOP is greater than the threshold.

Table 7.1 - Selection of the satellites during simulation.

Time	Satellite Combination				PDOP Value	Remarks
1	4	8	14	16	1.7955	Beginning of the simulation
4987	4	30	14	16	1.9002	One or more satellites are not visible anymore
5257	4	30	10	16	1.9281	One or more satellites are not visible anymore

From the results, in general, the half feedback configuration, which uses extended Kalman filter, gives smaller position and velocity error. This configuration also results smaller uncertainty than the feedforward configuration. There are two reasons why the error in feedforward configuration is larger than in half feedback. First, in this configuration, the output of the Kalman filter is not used to correct the INS signals before entering the filter. Second, in feedforward configuration, linearized Kalman filter was used. In this linearization method, the error tends to grow up as time increased [26].

There are also two spikes shown in the figure. Those spikes are resulted because of the change of the satellites in combination. Kalman filter corrects this large error in very short time; therefore, it looks like the spike in the figure.

From the plots of position error, Figure 7.19 until Figure 7.21, the half feedback configuration gives smaller position error than the feedforward configuration. For horizontal directions, the not-purely-white-noise characteristic of the accelerometer measurements affects the position errors. It can be seen that the results, mainly in East direction, follow the oscillating of the accelerometer outputs. For velocity error, Figure 7.22 - Figure 7.24, the pattern is the same as in position error.

This simulation experiment also shows different pattern of the results between two configurations in North and vertical directions. This is because the point used in the

linearization process is different. In linearized Kalman filter, predefined trajectory is used and in extended Kalman filter, the point is the corrected trajectory.

For covariance, Figure 7.25 - Figure 7.30, in general the extended Kalman filter gives smaller uncertainty. It can also be seen that the uncertainty increases before the change of combination occurs. This is reasonable, because when the satellites become not visible, the measurements produced from GPS receiver also become worst. From the plot of covariance can be seen that the third combination gives smaller uncertainty than the first two. The third combination also gives good results until the end of simulation. This pattern is valid for position error covariance as well as velocity error covariance.

For Kalman gain, Figure 7.31 shows that in the first combination, half feedback configuration gives more weight to the INS measurement than the feedforward configuration. The opposite has resulted in the third combination, where the half feedback gives more weight to the GPS measurement.

For innovation process, shown in Figure 7.32 - Figure 7.35, there is not much difference between two configurations. In half feedback system, the change in the combination affects all the innovation processes. This is because in this configuration, the input to the filter is affected by its output. This is important aspect to consider when we want to implement the adaptation scheme to improve the estimation.

Although the indirect half feedback configuration results in better estimation, in general no significant improvement resulted from this configuration compared to the indirect feedforward. This is because the inertial system itself is not corrected by the Kalman filter outputs. The corrections in half feedback configuration are only implemented on the output of inertial system before entering the filter. The amount of

improvement resulted from these corrections, is not significant to reduce the INS measurement errors. In this case, the INS measurements are still free to drift.

Figure 7.19 - East position error.

Figure 7.20 - North position error.

Figure 7.21 - Vertical position error.

Figure 7.22 - East velocity error.

Figure 7.23 - North velocity error.

Figure 7.24 - Vertical velocity error.

Figure 7.25 - East position error covariance.

Figure 7.26 - North position error covariance.

Figure 7.27 - Vertical position error covariance.

Figure 7.28 - East velocity error covariance.

Figure 7.29 - North velocity error covariance.

Figure 7.30 - Vertical velocity error covariance.

Figure 7.31 - Kalman gain.

Figure 7.32 - Innovation process on first channel.

Figure 7.33 - Innovation process on second channel.

Figure 7.34 - Innovation process on third channel.

Figure 7.35 - Innovation process on fourth channel.

Chapter 8

Conclusions

From the discussions that have been conducted in the previous chapters, some conclusions can be drawn:

1. No actual satellite signals measured by receivers are needed to calculate the DOP value. This value could be calculated on the basis of ephemeris data only. Therefore, the DOP value can be used as criterion to choose the best satellite combination from a number of available satellites without being affected by noise.
2. When the change of satellites in combination occurs, it affects the innovation process, which in turn affects the estimation. In indirect half feedback configuration, the effects occur in all channels of the receiver. In indirect feedforward configuration, the effects only occur on channels in which the change of satellites occurs.
3. The change of satellites combination affects the innovation process. The resulted errors could be improved using adaptive scheme and system.

4. The implementation of indirect half feedback Kalman filter configuration in GPS-aided inertial navigation system gives better estimation compared to the indirect feedforward configuration.
5. The improvement in performance of indirect half feedback configuration occurs because the INS measurements are corrected before entering the filter and the linearization is conducted based on corrected INS measurement. In short, the indirect half feedback configuration implements extended Kalman filter, and the indirect feedforward configuration implements linearized Kalman filter.
6. Although some improvements resulted from the indirect half feedback configuration, the difference is not significant compared to the indirect feedforward configuration. The amount of correction applied in the half feedback configuration is not significant compared to the error introduced by the inertial sensor. The inertial system itself is not corrected, therefore the errors resulted from the inertial sensor still grow unbounded.
7. Larger improvement could be resulted using full feedback configuration, in which the estimated errors resulted from the Kalman filter are feedback to the inertial system to correct the inertial system internally. However, this method can be only implemented in real time application.

Reference

- [1] Gai, Eli, "Guidance, Navigation, and Control: From Instrumentation to Information Management." *Journal of Guidance, Control, and Dynamics* 19(1), 1996, pp. 10-14.
- [2] Kaminer, I., Pascoal, A., Hallberg, & E., Silvestre, C., "Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control." *Journal of Guidance, Control, and Dynamics* 21(1), 1998, pp. 29-38.
- [3] Borenstein, J., & Feng, L., "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." *IEEE Transactions on Robotics and Automation* 12(6), 1996, pp. 869-879.
- [4] Shoval, S., Mishan, A., & Dayan, J., "Odometry and Triangulation Data Fusion for Mobile-Robots Environment Recognition." *Control Engineering Practice* 6, 1998, pp. 1383-1388.
- [5] Jetto, L., Longhi, S., & Vitali, D., "Localization of a Wheeled Mobile Robot by Sensor Data Fusion Based on a Fuzzy Logic Adapted Kalman Filter." *Control Engineering Practice* 7, 1999, pp. 763-771.
- [6] Jetto, L., Longhi, S., & Venturini, G., "Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localization of Mobile Robots." *IEEE Transactions on Robotics and Automation* 15(2), 1999, pp. 219-229.

- [7] Roumeliotis, S. I., Sukhatme, G. S., & Bekey, G. A., "Circumventing Dynamic Modeling: Evaluation of the Error-State Kalman Filter applied to Mobile Robot Localization." *Proceeding of the 1999 IEEE International Conference on Robotic & Automation*, Detroit, Michigan, May 1999, pp. 1656-1663.
- [8] McKerrow, Phillip John, "Introduction to Robotics." Addison-Wesley Publishing Company, Sydney, 1991.
- [9] Santini, A., Nicosia, S., & Nanni, V., "Trajectory Estimation and Correction for a Wheeled Mobile Robot Using Heterogeneous Sensors and Kalman Filter." *Preprints of the Fifth IFAC Symposium on Robot Control*, September 3-5, 1997, Nantes, France, pp. 11-16.
- [10] Vanderwerf, Kevin, "Schuler Pumping of Inertial Velocity Errors Due to Gravity Anomalies along a Popular North Pacific Airway." *IEEE 1996 Position, Location and Navigation Symposium*, Atlanta, Georgia, April 22-26, 1996, pp. 642-648.
- [11] Harriman, David W. and Harrison, J. Chris, "Gravity-Induced Errors in Airborne Inertial Navigation." *Journal of Guidance* 9(4), 1986, pp. 419-426.
- [12] Pozo-Ruz, A., Martinez, J. L., & Garcia-Cerezo, A., "A New Satellite Selection Criterion for DGPS Using Two Low-Cost Receivers." *Proceeding of the 1998 IEEE International Conference on Robotics & Automation*, Leuven, Belgium, May, 1998, pp. 1883-1888.
- [13] Abusali, P. A. M., Tapley, B. D., & Schutz, B. E., "Autonomous Navigation of Global Positioning System Satellites Using Cross-Link Measurements." *Journal of Guidance, Control, and Dynamics* 21(2), 1998, pp. 321-327.

- [14] Li, Jingming, *An Adaptive Kalman Filter for Tactical Missile Guidance Systems*, Master Thesis, Department of Mechanical and Aerospace Engineering, Ottawa, 1992.
- [15] Lawrence, Paul J., "Navigation Sensor, Filter, and Failure Mode Simulation Results Using the Distributed Kalman Filter Simulator (DKFSIM)." *IEEE 1996 Position, Location and Navigation Symposium*, Atlanta, Georgia, April 22-26, 1996, pp. 697-710.
- [16] Sukkarieh, Salah, Nebot, Eduardo M., & Durrant-Whyte, Hugh F., "Achieving Integrity in an INS/GPS Navigation Loop for Autonomous Land Vehicle Applications." *Proceeding of the 1998 IEEE International Conference on Robotics & Automation*, Leuven, Belgium, May, 1998, pp. 3437-3442.
- [17] Wang, Qi, "Control and Navigation System for Autonomous Vehicles and Robots Using Fuzzy Logic and Kalman Filtering." M.Eng. Thesis, Carleton University, Ottawa, 1999
- [18] Kayton, Myron and Fried, Walter, R., "Avionics Navigation Systems." John Wiley & Sons, Inc., New York, 1997.
- [19] Leick, Alfred., "GPS Satellite Surveying." John Wiley & Sons, Inc., New York, 1995.
- [20] Kaplan, Elliot D., "Understanding GPS Principles and Application." Artech House, Boston, 1996.

- [21] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., "Global Positioning System – Theory and Practice." Fourth, revised edition. Springer-Verlag, Wien New York, 1997.
- [22] McCarthy, Dennis D., "IERS Technical Note 21." IERS Conventions 1996. July 1996.
- [23] Farrell, Jay and Barth, Matthew, "The Global Positioning System and Inertial Navigation." McGraw-Hill, New York, 1998.
- [24] Lawrence, Anthony, "Modern Inertial Technology. Navigation, Guidance, and Control." Second Edition, Springer, 1998.
- [25] MacKenzie, Donald, "Inventing Accuracy. A Historical Sociology of Nuclear Missile Guidance." The MIT Press, 1990.
- [26] Grewal, Mohinder S. and Andrews, Angus P., "Kalman Filtering, Theory and Practice." Prentice Hall, New Jersey, 1993.
- [27] Jetto, L., Longhi, S., & Venturini, G., "Development and Experimental Validation of an Adaptive Estimation Algorithm for the On-line Localization of Mobile Robots by Multisensor Fusion." *Preprints of the Fifth IFAC Symposium on Robot Control*, Nantes, France, September 3-5, 1997, pp. 165-172.
- [28] Tham, Y. K., Wang, H., and Teoh, E. K., "Multi-Sensor Fusion for Steerable Four-Wheeled Industrial Vehicles." *Control Engineering Practice* 7, 1999 pp. 1233-1248.
- [29] Sasiadek, J. Z., & Wang, Q., "Sensor Fusion Based on Fuzzy Kalman Filtering for Autonomous Robot Vehicle." *Proceeding of the 1999 IEEE International*

- Conference on Robotics & Automation*, Detroit, Michigan, May 1999, pp. 2970-2975.
- [30] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems." *Transaction of the ASME*, March 1960, pp. 35-45.
- [31] Maybeck, Peter S., "Stochastic Models, Estimation, and Control." Volume 1, Academic Press, Inc., San Diego, 1979.
- [32] Ogata, Katsuhiko, "State Space Analysis of Control Systems." Prentice-Hall, Inc., Englewood Cliffs, 1967.
- [33] Candy, James V., "Signal Processing. The Model-based Approach." McGraw-Hill Book Company, New York, 1986.
- [34] Athans, M., et al., "Special Issue on Linear-Quadratic-Gaussian Problem." *IEEE Transactions on Automatic Control*, Vol. AC-16, 1971.
- [35] Sasiadek, J. Z. and Hartana, Pande, "Sensor Data Fusion Using Kalman Filter." To be presented in 3rd International Fusion2000 Conference in Paris, July 10 – 13, 2000, Paris.
- [36] Sasiadek, J. Z. and Hartana, Pande, "Odometry and Sonar Data Fusion for Mobile Robot Navigation." To be presented in 6th International IFAC Symposium on Robot Control - SYROCO 2000, September 21 – 23, 2000, Vienna.
- [37] Siouris, George M., "Aerospace Avionics Systems. A Modern Synthesis." Academic Press, Inc., San Diego, 1993.
- [38] Brown, Robert G. and Hwang, P. Y. C., "Introduction of Random Signals and Applied Kalman Filter." John Wiley & Sons, Inc., 1992.

- [39] Britting, Kenneth R., *Inertial Navigation Systems Analysis*, Wiley-Interscience, 1971.

Appendix A

GPS Measurements and Ephemeris Data Format

In this appendix, the example of data format for GPS measurements and GPS ephemeris are provided. For GPS measurements, the data are in Receiver Independent Exchange (Rinex) format version 2. The example of these data is provided in appendix A.1. For GPS ephemeris, the data are in Standard Product #3 (SP3) format. The example of these data is provided in appendix 0. The explanations of these formats can be found from the web site. The addresses are:

- Rinex2 format:

<http://www.ngs.noaa.gov/CORS/Rinex2.html>.

- SP3 format:

http://www.ngs.noaa.gov/GPS/SP3_format.html.

A.1 GPS Measurements Data in Rinex2 Format

```

1          2          3          4          5          6          7          8
1234567890123456789012345678901234567890123456789012345678901234567890

1          2          Observation data      G      (GPS)      RINEX VERSION / TYPE
2      ALL2RNX      12-Nov-99 12:46      PGM / RUN BY /DATE
3      Converted from 221$0940.MON of 14-Sep-99 15:31      COMMENT
4                                          MARKER NAME
5                                          OBSERVER / AGENCY
6                                          REC # / TYPE / VERS
7                                          ANT # / TYPE
8      1110221.0090 -4351914.7812  4513676.6406      APPROX POSITION XYZ
9      0.0000      0.0000      0.0000      ANTENNA: DELTA H/E/N
10     1      1      6      G 2      G 4      G 5      G 6      G 7      G 8      WAVELENGTH FACT L1/2
11     1      1      6      G 9      G10      G13      G14      G16      G18      WAVELENGTH FACT L1/2
12     1      1      3      G24      G26      G30      WAVELENGTH FACT L1/2
13     6      C1      P2      L1      L2      D1      D2      # / TYPES OF OBSERV
14     0.10      INTERVAL
15     1999      8      9      11      14      58.500000      GPS      TIME OF FIRST OBS
16     1999      8      9      15      11      42.300000      GPS      TIME OF LAST OBS
17     12      LEAP SECONDS
18     15      # OF SATELLITES
19     G 2 33394 33394 33394 33394 33394 33394      PRN / # OF OBS
20     G 4142039142039142039142039142039142039      PRN / # OF OBS
21     G 5142039142039142039142039142039142039      PRN / # OF OBS
22     G 6 57953 57953 57953 57953 57953 57953      PRN / # OF OBS
23     G 7 98152 98152 98152 98152 98152 98152      PRN / # OF OBS
24     G 8 49859 49859 49859 49859 49859 49859      PRN / # OF OBS
25     G 9 35765 35765 35765 35765 35765 35765      PRN / # OF OBS
26     G10109537109537109537109537109537109537      PRN / # OF OBS
27     G13 80737 80737 80737 80737 80737 80737      PRN / # OF OBS
28     G14 52552 52552 52552 52552 52552 52552      PRN / # OF OBS
29     G16 97249 97249 97249 97249 97249 97249      PRN / # OF OBS
30     G18121244121244121244121244121244121244      PRN / # OF OBS
31     G24142039142039142039142039142039142039      PRN / # OF OBS
32     G26 12366 12366 12366 12366 12366 12366      PRN / # OF OBS
33     G30113451113451113451113451113451113451      PRN / # OF OBS
34                                          END OF HEADER
35     99  8  9 11 14 58.5000000  0  9G 2G 4G 5G 7G 8G 9G14G16G24  0  0  0
36     22491985.289      22491984.445      118196366.328      92100874.250      -3187.449
37     -2483.297
38     20727292.812      20727290.227      108923009.512      84874737.727      -1490.645
39     -1161.977
40     23930905.359      23930903.672      125757897.754      97992957.754      -3157.895
41     -2461.129
42     20448902.992      20448900.289      107460023.168      83734815.812      638.324
43     496.957
44     24208598.383      24208315.875      127217046.004      99129005.875      -730.395
45     -569.859
46     23916113.688      23916112.859      125680489.730      97932587.988      1621.387
47     1262.980
48     22538538.961      22538538.648      118441087.098      92291554.492      1433.387
49     1116.488
50     21394928.445      21394926.977      112431353.012      87608809.715      -712.957
51     -555.988
52     23244007.922      23244006.945      122148473.426      95180286.672      -2976.738
53     -2319.984
54     99  8  9 11 14 58.6000000  0  9G 2G 4G 5G 7G 8G 9G14G16G24  0  0  0
55     22492045.938      22492045.109      118196685.129      92101122.664      3187.363
56     2483.207
57     20727264.469      20727261.875      108922860.512      84874621.621      -1490.668
58     -1161.996
59     23930845.273      23930843.586      125757582.023      97992711.730      -3158.043
60     -2461.242
61     20448915.141      20448912.445      107460087.059      83734865.602      638.363
62     496.988

```


63	24208584.492	24208300.469	127216972.996	99128942.992	-731.012
64	-570.043				
65	23916144.578	23916143.727	125680651.930	97932714.375	1621.332
66	1262.938				
67	22538566.242	22538565.930	118441230.492	92291666.227	1433.426
68	1116.520				
69	21394914.891	21394913.422	112431281.762	87608754.195	-713.012
70	-556.031				
71	23243951.289	23243950.312	122148175.812	95180054.766	-2976.730
72	-2319.977				
73	99 8 9 11 14	58.7000000 0	9G 2G 4G 5G 7G 8G	9G14G16G24 0 0 0	
74	22492106.594	22492105.766	118197003.906	92101371.066	3187.309
75	2483.188				
76	20727236.086	20727233.523	108922711.504	84874505.512	-1490.754
77	-1162.062				
78	23930785.203	23930783.492	125757266.270	97992465.688	-3158.066
79	-2461.262				
80	20448927.312	20448924.594	107460150.938	83734915.375	638.184
81	496.855				
82	24208570.492	24208286.578	127216899.977	99128886.082	-731.066
83	-569.961				
84	23916175.438	23916174.586	125680814.113	97932840.754	1621.340
85	1262.945				
86	22538593.531	22538593.219	118441373.898	92291777.973	1433.434
87	1116.500				
88	21394901.336	21394899.867	112431210.520	87608698.680	-713.035
89	-556.047				
90	23243894.680	23243893.680	122147878.195	95179822.859	-2976.691
91	-2319.945				
92	99 8 9 11 14	58.8000000 0	9G 2G 4G 5G 7G 8G	9G14G16G24 0 0 0	
93	22492167.258	22492166.430	118197322.695	92101619.469	3187.141
94	2483.055				
95	20727207.742	20727205.172	108922562.484	84874389.395	-1490.922
96	-1162.191				
97	23930725.109	23930723.406	125756950.520	97992219.648	-3158.328
98	-2461.465				
99	20448939.469	20448936.750	107460214.828	83734965.160	638.016
100	496.723				
101	24208556.594	24208272.672	127216826.938	99128829.152	-730.953
102	-569.848				
103	23916206.289	23916205.453	125680976.297	97932967.133	1621.203
104	1262.836				
105	22538620.820	22538620.508	118441517.289	92291889.707	1433.203
106	1116.344				
107	21394887.773	21394886.312	112431139.273	87608643.164	-713.109
108	-556.105				
109	23243838.039	23243837.039	122147580.562	95179590.941	-2977.047
110	-2320.223				
111	99 8 9 11 14	58.9000000 0	9G 2G 4G 5G 7G 8G	9G14G16G24 0 0 0	
112	22492227.922	22492227.094	118197641.461	92101867.859	3187.023
113	2482.965				
114	20727179.383	20727176.820	108922413.445	84874273.262	-1490.914
115	-1162.184				

:

A.2 GPS Ephemeris Data in SP3 Format

```

1          1          2          3          4          5          6
123456789012345678901234567890123456789012345678901234567890

1 #aP1999 8 9 11 14 58.50000000 10802 D ITR96 FIT NOA
2 ## 1022 126898.50000000 1.00000000 51399 0.4687326388889
3 + 27 1 2 3 4 5 6 7 8 9 10 13 14 15 16 17 18 19
4 + 21 22 23 24 25 26 27 29 30 31 0 0 0 0 0 0 0
5 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 ++ 4 5 4 4 4 5 5 5 5 5 5 4 5 5 4 5 5
9 ++ 5 5 5 5 5 5 4 5 4 5 0 0 0 0 0 0 0
10 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 %c CC CC CCC CCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC
14 %c CC CC CCC CCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC
15 %f 0.0000000 0.000000000 0.000000000000 0.000000000000000
16 %f 0.0000000 0.000000000 0.00000000000 0.000000000000000
17 %i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18 %i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19 /* The NGS post-fit satellite positions have been combined
20 /* with Broadcast ephemeris clock values at the epochs shown
21 /* (Use these Broadcast clock values at your own risk.)
22 /* CCCC-----CCCC
23 * 1999 8 9 11 14 58.50000000
24 P 1 15022.457433 12231.665548 18346.540776 94.119298
25 P 2 10681.292798 -23955.897080 -841.500382 -95.540892
26 P 3 11024.293308 13523.121607 -20054.914430 30.468188
27 P 4 -2969.290943 -22270.135206 14113.411580 19.689000
28 P 5 -16080.289674 -1789.791689 21055.898727 211.424397
29 P 6 -25391.779043 5584.666391 -5933.145169 -0.021400
30 P 7 10749.602620 -14764.807931 19477.493259 577.124198
31 P 8 -16914.692513 -215.943184 20162.153489 56.615804
32 P 9 -19712.629195 -13766.835589 11561.368581 -5.393898
33 P 10 -4674.302625 -18708.589119 -18181.946468 39.375999
34 P 13 5493.871944 -15087.358809 -21095.379437 -80.215962
35 P 14 16725.706523 -1636.271173 20549.952338 29.140005
36 P 15 18336.191841 12964.726687 13947.773329 712.987394
37 P 16 18203.154104 -12982.667813 14144.297405 129.829797
38 P 17 -12050.014371 8625.183315 -22121.491346 -218.043399
39 P 18 23959.925641 -8478.393579 -8099.465148 25.781300
40 P 19 15750.996262 -5078.876682 -20737.870814 28.848579
41 P 21 -9488.961506 24226.643032 3337.402021 26.814605
42 P 22 665.383820 22890.040850 -13181.139412 424.906518
43 P 23 -16974.436469 17755.822810 -9354.882478 3.650700
44 P 24 -12468.950182 -23359.831307 2831.473310 589.369803
45 P 25 5049.933766 22802.140141 13064.956411 1.973000
46 P 26 -21191.514659 -8487.999482 -14016.325686 665.570691
47 P 27 5974.940052 -16375.795108 -19745.272973 28.466598
48 P 29 -11148.372656 14625.246351 19325.143629 522.334797
49 P 30 -17989.555231 11096.438989 15907.172009 -27.855898
50 P 31 23200.228367 6414.250075 -11761.999193 23.050799
51 * 1999 8 9 11 14 59.50000000
52 P 1 15020.169194 12231.916614 18348.257888 94.119299
53 P 2 10681.607166 -23955.580715 -844.650499 -95.540898
54 P 3 11021.770534 13523.479931 -20056.062916 30.468196
55 P 4 -2968.174110 -22268.737051 14115.887569 19.689067
56 P 5 -16079.271932 -1792.275000 21056.472321 211.424399
57 P 6 -25392.559439 5584.430590 -5930.125188 -0.021400
58 P 7 10752.028484 -14764.870180 19476.057822 577.124199
59 P 8 -16916.000124 -218.226943 20161.038673 56.615868
60 P 9 -19713.257426 -13768.199307 11558.744354 -5.393899
61 P 10 -4673.152687 -18710.602106 -18180.154887 39.376000
62 P 13 5496.561574 -15087.469099 -21094.594925 -80.215987

```

63	P 14	16724.520823	-1633.945441	20551.098989	29.140001
64	P 15	18336.957195	12966.373076	13945.280618	712.987398
65	P 16	18202.450081	-12980.989549	14146.753578	129.829799
66	P 17	-12051.356282	8622.749057	-22121.760032	-218.043400
67	P 18	23960.946834	-8478.217256	-8096.551159	25.781300
68	P 19	15752.569339	-5076.731957	-20737.225782	28.848594
69	P 21	-9489.410083	24226.869896	3334.198792	26.814601
70	P 22	664.425892	22891.446975	-13178.660142	424.906506
71	P 23	-16974.050416	17754.604725	-9357.791977	3.650700
72	P 24	-12468.618916	-23359.655782	2834.674985	589.369801
73	P 25	5048.750017	22800.996692	13067.422238	1.973000
74	P 26	-21189.768924	-8488.284630	-14018.736168	665.570697
75	P 27	5976.565403	-16373.812111	-19746.480865	28.466599
76	P 29	-11150.790132	14625.324986	19323.661111	522.334799
77	P 30	-17988.878712	11094.484266	15909.279189	-27.855899
78	P 31	23198.800332	6414.599831	-11764.642792	23.050800
79	*	1999 8 9 11 15	0.50000000		
80	P 1	15017.880756	12232.167822	18349.974615	94.119301
81	P 2	10681.921401	-23955.263996	-847.800597	-95.540902
82	P 3	11019.247636	13523.838407	-20057.210977	30.468204
83	P 4	-2967.057027	-22267.338701	14118.363256	19.689133
84	P 5	-16078.254295	-1794.758430	21057.045467	211.424401
85	P 6	-25393.339470	5584.194816	-5927.105081	-0.021400
86	P 7	10754.454172	-14764.932551	19474.621976	577.124201
87	P 8	-16917.307789	-220.510508	20159.923416	56.615932
88	P 9	-19713.885546	-13769.562717	11556.119885	-5.393901
89	P 10	-4672.002968	-18712.614960	-18178.362916	39.376000
90	P 13	5499.251100	-15087.579538	-21093.809962	-80.216013
91	P 14	16723.335195	-1631.619509	20552.245202	29.139999
92	P 15	18337.722491	12968.019143	13942.787607	712.987402
93	P 16	18201.746007	-12979.310973	14149.209446	129.829801
94	P 17	-12052.698359	8620.314858	-22122.028251	-218.043400
95	P 18	23961.967678	-8478.040948	-8093.636999	25.781300
96	P 19	15754.142476	-5074.587380	-20736.580307	28.848606
97	P 21	-9489.858469	24227.096419	3330.995490	26.814599
98	P 22	663.468157	22892.852867	-13176.180587	424.906494
99	P 23	-16973.664258	17753.386288	-9360.701271	3.650700
100	P 24	-12468.287428	-23359.479937	2837.876600	589.369799
101	P 25	5047.566022	22799.853063	13069.887794	1.973000
102	P 26	-21188.022905	-8488.569901	-14021.146359	665.570703
103	P 27	5978.190945	-16371.829081	-19747.688327	28.466601
104	P 29	-11153.207421	14625.403745	19322.178187	522.334801
105	P 30	-17988.202186	11092.529265	15911.386026	-27.855901
106	P 31	23197.371991	6414.949697	-11767.286147	23.050800
107	*	1999 8 9 11 15	1.50000000		
108	P 1	15015.592120	12232.419171	18351.690957	94.119302
109	P 2	10682.235503	-23954.946921	-850.950676	-95.540907
110	P 3	11016.724615	13524.197037	-20058.358612	30.468211
111	P 4	-2965.939691	-22265.940157	14120.838642	19.689200
112	P 5	-16077.236764	-1797.241980	21057.618164	211.424403
113	P 6	-25394.119137	5583.959068	-5924.084851	-0.021400
114	P 7	10756.879682	-14764.995046	19473.185724	577.124202
115	P 8	-16918.615506	-222.793879	20158.807719	56.615996
116	P 9	-19714.513557	-13770.925819	11553.495174	-5.393902
117	P 10	-4670.853466	-18714.627681	-18176.570556	39.376001
118	P 13	5501.940520	-15087.690128	-21093.024548	-80.216038
119	P 14	16722.149639	-1629.293379	20553.390977	29.139996
120	P 15	18338.487729	12969.664887	13940.294294	712.987406
121	P 16	18201.041883	-12977.632083	14151.665009	129.829803
122	P 17	-12054.040600	8617.880719	-22122.296003	-218.043401
123	P 18	23962.988172	-8477.864657	-8090.722669	25.781300
124	P 19	15755.715674	-5072.442952	-20735.934390	28.848619
125	P 21	-9490.306662	24227.322601	3327.792113	26.814596
126	P 22	662.510617	22894.258525	-13173.700747	424.906482
127	P 23	-16973.277996	17752.167499	-9363.610360	3.650700
128	P 24	-12467.955717	-23359.303772	2841.078156	589.369797
129	P 25	5046.381783	22798.709254	13072.353078	1.973000
130	P 26	-21186.276600	-8488.855296	-14023.556260	665.570709
131	P 27	5979.816677	-16369.846018	-19748.895357	28.466602
132	P 29	-11155.624524	14625.482626	19320.694858	522.334803
133	P 30	-17987.525654	11090.573985	15913.492520	-27.855902

134	P	31	23195.943343	6415.299672	-11769.929259	23.050801
135	*	1999 8 9 11 15	2.50000000			
136	P	1	15013.303285	12232.670663	18353.406914	94.119303
137	P	2	10682.549473	-23954.629491	-854.100737	-95.540912
138	P	3	11014.201472	13524.555819	-20059.505822	30.468219
139	P	4	-2964.822104	-22264.541420	14123.313726	19.689267
140	P	5	-16076.219338	-1799.725650	21058.190413	211.424405

⋮

Appendix B

Simulation Program

The simulation program was written in Matlab[®]. The program consists of two main programs. The first one is the Kalman filter simulation. The outputs of this program are written into the text file in ASCII format. The second program is for plotting purpose. The text file produced by the Kalman filter program is used as the input for this program. For Kalman filter program, there are 14 related functions. Those functions are:

1. StateTransMatrix.m

This function is used to generate the state transition matrix.

2. ProcNoiseCov.m

This function is used to generate process noise covariance matrix.

3. MeasNoiseCov.m

This function is used to generate measurement noise covariance matrix.

4. ReadPosCar.m

This function reads the INS measurement data.

5. CartToGeod.m

This function transforms Cartesian coordinate into geodetic coordinate system.

6. ChooseSatComb.m

This function chooses the satellites combination based on the lowest PDOP.

7. ArrangeComb.m

This function arranges the order of the satellites in combination so as it minimizes the tracking error.

8. SatVis.m

This function checks whether the satellites in combination are still visible.

9. CreateNewSatComb.m

This function creates new satellites combination when one or more satellites in current combination are not visible anymore.

10. FindPdop.m

This function calculates the PDOP value.

11. ChooseNewSatComb.m

This function chooses the new satellite combination when the PDOP is greater than threshold.

12. MeasMatrix.m

This function generates the measurement matrix.

13. HTranMat.m

This function transforms the measurement matrix from Cartesian into Geodetic coordinate system.

14. PredMeas.m

This function generates the observation which will be fed into Kalman filter.

In the following, the source codes for the main program and the related functions are provided.

B.1 Kalman.m

```
1 % Kalman is the Kalman filter simulation for GPS-INS
2 %
3 % File name: Kalman.m
4 % Kalman is the Kalman filter simulation program for
5 % GPS aided inertial navigation system.
6 %
7 % Creator: Pande Hartana
8 % Created: Saturday, April 15, 2000
9 % Last Updated: Tuesday, April 18, 2000
10 % =====
11 clear;
12 format long g;
13
14 % Print message to the screen
15 disp('INS-GPS using EKF processing.... ')
16
17 % Prepare file handle to solve the results
18 % Position error
19 fXPosErr = fopen('EPosErr.dat', 'w');
20 fYPosErr = fopen('NPosErr.dat', 'w');
21 fZPosErr = fopen('VPosErr.dat', 'w');
22
23 fXPosCov = fopen('EPosCov.dat', 'w');
24 fYPosCov = fopen('NPosCov.dat', 'w');
25 fZPosCov = fopen('VPosCov.dat', 'w');
26
27 % Velocity error
28 fXVelErr = fopen('EVelErr.dat', 'w');
29 fYVelErr = fopen('NVelErr.dat', 'w');
30 fZVelErr = fopen('VVelErr.dat', 'w');
31
32 fXVelCov = fopen('EVelCov.dat', 'w');
33 fYVelCov = fopen('NVelCov.dat', 'w');
34 fZVelCov = fopen('VVelCov.dat', 'w');
35
36 % Innovation process
37 fInnProc1 = fopen('InnProc1.dat', 'w');
38 fInnProc2 = fopen('InnProc2.dat', 'w');
39 fInnProc3 = fopen('InnProc3.dat', 'w');
40 fInnProc4 = fopen('InnProc4.dat', 'w');
41 % Kalman gain
42 fKalGain = fopen('KalGain.dat', 'w');
43 % Time
44 fTime = fopen('Time.dat', 'w');
45
46 % Result.txt
47 fResult = fopen('Result.txt', 'w');
48
49 % Sampling time
50 tSamp = 1;
51
52 % INS data frequency
```

```

53 insF = 2;
54
55 % Pseudorange data frequency
56 gpsPrF = 10;
57
58 % GPS position data frequency
59 gpsPF = 1;
60
61 % Process noise spectral amplitude
62 Sp = 1;
63 Sf = 0.4e-18;
64 Sg = 1.58e-18;
65
66 % Measurement noise spectral amplitude
67 Sv = 25;
68
69 % Initial error state
70 xEst = zeros(8, 1);
71
72 % Initial error covariance
73 pEst = 10*eye(8, 8);
74 pEst(1,1) = 400;
75 pEst(2,2) = 100;
76 pEst(3,3) = 400;
77 pEst(4,4) = 100;
78 pEst(5,5) = 400;
79 pEst(5,6) = 100;
80 pEst(7,7) = 10^10;
81 pEst(8,8) = 10^4;
82
83 % Build state transition matrix
84 stMat = StateTransMatrix(tSamp);
85
86 % Build process noise covariance
87 Q = ProcNoiseCov(Sp, Sf, Sg, tSamp);
88
89 % Build measurement noise covariance
90 R = MeasNoiseCov(Sv);
91
92 numIter = (1/tSamp)*60*180; % The last number is minute long
93 inIter = 1;
94
95 SatComb = [2, 4, 5, 30]; % Initialize any satellites combination
96 PreSatComb = SatComb;
97 notVis = 1;
98 PDOPTrHold = 2;
99
100 for k = inIter:numIter
101
102     tTime = k * tSamp;
103
104     % Find number of lines omitted in INS data
105     insOmit = (k-1) * insF;
106
107     % Find number of lines omitted in GPS data
108     gpsPrOmit = (k-1) * gpsPrF;
109
110     % Find number of lines omitted in GPS position data
111     gpsPOmit = (k-1) * gpsPF;
112
113     % Read position data measured by INS (in Cartesian)
114     [xPos, yPos, zPos] = ReadPosCar(insOmit);
115     iPos = [xPos; yPos; zPos];
116
117     % Correct INS position with EKF output
118     iPos = iPos - [xEst(1,1); xEst(3,1); xEst(5,1)];
119
120     % Transform corrected position into geodetic
121     [PosLati, PosLong, Height] = CartToGeod(iPos(1), iPos(2), iPos(3));
122
123     % Reset state estimation

```



```

124     xEst = zeros(8, 1);
125
126     % Find combination of satellites
127     % Check if the satellites are still visible
128     if (k == inIter)
129         % Find the correct satellites combination
130         [SatComb, PDOP] = ChooseSatComb(gpsPOmit, gpsPrOmit,...
131             iPos(1), iPos(2), iPos(3));
132         fprintf(fResult, '%s\n', ' ');
133         fprintf(fResult, '%s\n', 'Beginning of the process');
134         fprintf(fResult, '%s\n', ['Time = ', int2str(k)]);
135         fprintf(fResult, '%s\n', ['Satellites combination = ', int2str(SatComb)]);
136         fprintf(fResult, '%s\n', ['PDOP = ', num2str(PDOP)]);
137     else
138         % Check if the satellites in combination are still visible
139         [notVis, VisSatMat] = SatVis(SatComb, gpsPrOmit);
140         if (notVis == 1)
141             % If not visible anymore, find the new combination
142             [SatComb, PDOP] = CreateNewSatComb(gpsPOmit, gpsPrOmit,...
143                 iPos(1), iPos(2), iPos(3), VisSatMat, SatComb);
144             fprintf(fResult, '%s\n', ' ');
145             fprintf(fResult, '%s\n', ...
146                 'One or more satellites are not visible anymore');
147             fprintf(fResult, '%s\n', 'Need new satellites combination. ');
148             fprintf(fResult, '%s\n', ['Time = ', int2str(k)]);
149             fprintf(fResult, '%s\n', ['New Satellites combination = ', ...
150                 int2str(SatComb)]);
151             fprintf(fResult, '%s\n', ['New PDOP = ', num2str(PDOP)]);
152         end
153         % If still visible, check the value of PDOP
154         PDOPVal = FindPdop(SatComb, gpsPOmit, iPos(1), iPos(2), iPos(3));
155         if (PDOPVal > PDOPTrHold)
156             fprintf(fResult, '%s\n', ' ');
157             fprintf(fResult, '%s\n', 'PDOP is greater than threshold');
158             fprintf(fResult, '%s\n', 'Need new satellites combination. ');
159             fprintf(fResult, '%s\n', ['Time = ', int2str(k)]);
160             [SatComb, PDOP] = ChooseNewSatComb(gpsPOmit, gpsPrOmit,...
161                 iPos(1), iPos(2), iPos(3), SatComb);
162             fprintf(fResult, '%s\n', ['New Satellites combination = ', ...
163                 int2str(SatComb)]);
164             fprintf(fResult, '%s\n', ['New PDOP = ', num2str(PDOP)]);
165             if (PDOP > PDOPTrHold)
166                 fprintf(fResult, '%s\n', ' ');
167                 fprintf(fResult, '%s\n', ...
168                     'PDOP is still greater than threshold');
169                 fprintf(fResult, '%s\n', 'Threshold increased. ');
170                 fprintf(fResult, '%s\n', ['Time = ', int2str(k)]);
171                 PDOPTrHold = PDOPTrHold + 1;
172             end
173         end
174     end
175
176     % Build measurement matrix
177     HC = MeasMatrix(SatComb, gpsPOmit, iPos(1), iPos(2), iPos(3));
178
179     % Build transformation matrix from Cartesian to local frame
180     tranMat = HTranMat(PosLati, PosLong);
181
182     % Transform measurement matrix into local frame
183     H = HC * tranMat;
184
185     % Calculate a priori estimation
186     xMin = stMat * xEst;
187
188     % Calculate a priori state estimation covariance
189     pMin = stMat*pEst*stMat' + Q;
190
191     % Compute Kalman gain
192     KGain = pMin * H' * inv(H*pMin*H' + R);
193
194     % Read observable measurement error

```

```

195     Y = PredMeas(SatComb, gpsPOmit, gpsPrOmit, iPos(1), iPos(2), iPos(3), k);
196
197     % Build innovation process
198     Z = Y - H*xMin;
199
200     % Update estimation with correction
201     xEst = xMin + KGain * Z;
202
203     % Calculate updated estimation covariance
204     idMat = eye(8);
205     pEst = (idMat - KGain*H) * pMin * ...
206     (idMat - KGain*H)' + KGain * R * KGain';
207
208     % Save results to file
209     % Position error
210     fprintf(fXPosErr, '%s\n', num2str(xEst(1,1)));
211     fprintf(fYPosErr, '%s\n', num2str(xEst(3,1)));
212     fprintf(fZPosErr, '%s\n', num2str(xEst(5,1)));
213
214     fprintf(fXPosCov, '%s\n', num2str(pEst(1,1)));
215     fprintf(fYPosCov, '%s\n', num2str(pEst(3,3)));
216     fprintf(fZPosCov, '%s\n', num2str(pEst(5,5)));
217
218     % Velocity error
219     fprintf(fXVelErr, '%s\n', num2str(xEst(2,1)));
220     fprintf(fYVelErr, '%s\n', num2str(xEst(4,1)));
221     fprintf(fZVelErr, '%s\n', num2str(xEst(6,1)));
222
223     fprintf(fXVelCov, '%s\n', num2str(pEst(2,2)));
224     fprintf(fYVelCov, '%s\n', num2str(pEst(4,4)));
225     fprintf(fZVelCov, '%s\n', num2str(pEst(6,6)));
226
227     % Innovation process
228     fprintf(fInnProc1, '%s\n', num2str(Z(1,1)));
229     fprintf(fInnProc2, '%s\n', num2str(Z(2,1)));
230     fprintf(fInnProc3, '%s\n', num2str(Z(3,1)));
231     fprintf(fInnProc4, '%s\n', num2str(Z(4,1)));
232     % Kalman gain
233     fprintf(fKalGain, '%s\n', num2str(KGain(3,1)));
234     % Time
235     fprintf(fTime, '%s\n', num2str(tTime));
236
237     % Transform output into Cartesian
238     xEst = tranMat' * xEst;
239
240 end
241
242 disp(' ')
243 disp(['Time = ', int2str(k)])
244 disp('EKF process finished.')
245 fprintf(fResult, '%s\n', ' ');
246 fprintf(fResult, '%s\n', ['Time = ', int2str(k)]);
247 fprintf(fResult, '%s\n', 'EKF process finished.');
```

```

248
249 % Close all file handle
250 status = fclose('all');
```

B.2 StateTransMatrix.m

```

1 function stMat = StateTransMatrix(dT)
2 % StateTransMatrix state transition matrix
3 %
4 % File name: StateTransMatrix.m
5 % StateTransMatrix creates state transition matrix
6 %
7 % The input is the sampling time.
```

```

8 % The output is the state transition matrix.
9 %
10 % Creator: Pande Hartana
11 % Created: Saturday, April 15, 2000
12 % Last Updated: Tuesday, April 18, 2000
13 % =====
14
15 % Gravitational acceleration
16 g = 9.81;
17
18 % Radius of Earth
19 R = 6378136.49;
20
21 stMat = eye(8);
22 stMat(1,2) = dT;
23 stMat(3,4) = dT;
24 stMat(5,6) = dT;
25 stMat(6,5) = 2*g*dT/R;
26 stMat(7,8) = dT;
27
28 % End of function

```

B.3 ProcNoiseCov.m

```

1 function Q = ProcNoiseCov(Sp, Sf, Sg, dT)
2 % ProcNoiseCov process noise covariance matrix
3 %
4 % File name: ProcNoiseCov.m
5 % ProcNoiseCov builds process noise covariance matrix.
6 %
7 % The inputs are noise spectral amplitude (Sp, Sf, Sg), and
8 % sampling time (dT).
9 % The output is process noise covariance matrix
10 %
11 % Creator: Pande Hartana
12 % Created: Saturday, April 15, 2000
13 % Last Updated: Saturday, April 15, 2000
14 % =====
15
16 % Speed of light
17 % (from IERS Technical Note 21, IERS Conventions 1996)
18 c = 299792458; % in meter/second
19
20 % Build 2 by 2 zero matrix
21 Qs = zeros(2);
22
23 % Build two-three diagonal elements
24 Qd(1,1) = Sp*dT^3/3;
25 Qd(1,2) = Sp*dT^2/2;
26 Qd(2,1) = Qd(1,2);
27 Qd(2,2) = Sp*dT;
28
29 Ql(1,1) = ((Sf*dT) + (Sg*dT^3/3)) * c^2;
30 Ql(1,2) = c^2*Sg*dT^2/2;
31 Ql(2,1) = Ql(1,2);
32 Ql(2,2) = Sg*dT*c^2;
33
34 % Build Q matrix
35 Q = [Qd, Qs, Qs, Qs;
36      Qs, Qd, Qs, Qs;
37      Qs, Qs, Qd, Qs;
38      Qs, Qs, Qs, Ql];
39
40 % End of function

```

B.4 MeasNoiseCov.m

```
1 function R = MeasNoiseCov(Sv)
2 % MeasNoiseCov measurement noise covariance matrix
3 %
4 % File name: MeasNoiseCov.m
5 % MeasNoiseCov builds measurement noise covariance matrix
6 %
7 % The input is measurement noise one sigma (Sv)
8 % The output is Measurement noise covariance matrix
9 %
10 % Creator: Pande Hartana
11 % Created: Saturday, April 15, 2000
12 % Last Updated: Monday, May 22, 2000
13 % =====
14
15 % Diagonal element
16 Re = Sv^2;
17
18 Rd = [Re, Re, Re, Re];
19
20 R = diag(Rd);
21
22 % End of function
```

B.5 ReadPosCar.m

```
1 function [xPos, yPos, zPos] = ReadPosCar(nLine)
2 % ReadPosCar reading Cartesian position
3 %
4 % File name: ReadPosCar.m
5 % ReadPosCar reads position of the aircraft
6 % as measured by INS in Cartesian coordinate
7 %
8 % Creator: Pande Hartana
9 % Created: Tuesday, April 25, 2000
10 % Last Updated: Monday, May 22, 2000
11 % =====
12
13 % Data file name
14 fOpen = {'INSPosX.dat'; 'INSPosY.dat'; 'INSPosZ.dat'};
15
16 % Read data
17 for k = 1:3
18     fOpenID = fopen(fOpen(k, :), 'r');
19     fseek(fOpenID, 17*nLine, 'bof');
20     readData = fgetl(fOpenID);
21     dataRead(k) = sscanf(readData, '%g');
22     status = fclose(fOpenID);
23 end
24
25 % Function outputs
26 xPos = dataRead(1);
27 yPos = dataRead(2);
28 zPos = dataRead(3);
29
30 % End of function
```

B.6 CartToGeod.m

```
1 function [lat, long, height] = CartToGeod(x, y, z)
2 % CartToGeod conversion from Cartesian into Geoidetic
3 %
4 % File name: CartToGeod.m
5 % CartToGeod converts the Cartesian ECEF coordinate system
6 % into Geodetic ECEF coordinate system.
7 % The input is Cartesian coordinate system in
8 % the form x, y, z (meter)
9 % The output is Geodetic coordinate system in the
10 % form latitude(degree), longitude(degree), height(meter).
11 %
12 % Creator: Pande Hartana
13 % Created: Wednesday, April 12, 2000
14 % Last Updated: Wednesday, April 12, 2000
15 % =====
16
17 % Note:
18 % This m file was developed base on the equations from the book
19 % GPS Sarellite Surveying, Second Edition, Alfred Leick
20 % John Wiley & Sons, Inc.
21
22 % Equatorial radius of the Earth
23 % (from IERS Technical Note 21, IERS Conventions 1996)
24 a = 6378136.49; % in meter
25
26 % Flattening factor of the Earth
27 % (from IERS Technical Note 21, IERS Conventions 1996)
28 f = 1/298.25642;
29
30 e = sqrt(2*f - f^2);
31
32 % Conversion
33 long = atan(y/x);
34 again = 0;
35 while (again == 0)
36     lat1 = atan(z / ((1 - e^2)*sqrt(x^2 + y^2)));
37     N = a / sqrt(1 - e^2 * (sin(lat1))^2);
38     lat2 = atan(z / sqrt(x^2+y^2) * (1 + (e^2*N*sin(lat1)/z)));
39     if (abs(lat1 - lat2) < 1)
40         again = 1;
41     end
42 end
43 lat = lat2;
44 height = sqrt(x^2 + y^2)/cos(lat) - N;
45 lat = lat2*180/pi;
46 long = long*180/pi;
47
48 % End of function
```

B.7 ChooseSatComb.m

```
1 function [SatComb, PDOP] = ChooseSatComb(nLine, PrLine, xo, yo, zo)
2 % ChooseSatComb chooses a combination of satellites
3 %
4 % File name: ChooseSatComb.m
5 % ChooseSatComb chooses a combination of satellites, which
6 % gives the lowest PDOP.
7 % The input is the number of data that will be discarded
8 % and approximated vehicle position (xo, yo, zo).
9 % The output is the satellites combination and PDOP value.
10 %
```

```

11 % Creator: Pande Hartana
12 % Created: Saturday, April 15, 2000
13 % Last Updated: Monday, May 22, 2000
14 % =====
15
16 % Speed of light
17 % (from IERS Technical Note 21, IERS Conventions 1996)
18 c = 299792458; % in meter/second
19
20 % Open and read satellite position files.
21 visSatInData = [2, 4, 5, 6, 7, 8, 9, 10, 13, 14, 16, 18, 24, 26, 30];
22
23 % Read pseudorange data for particular line.
24 % If pseudorange is equal to zero,
25 % that means satellite is not visible.
26 visSat = [];
27 [n, numSat] = size(visSatInData);
28 for k = 1:numSat
29     fileName = strcat('DNG', int2str(visSatInData(k)), 'C1.dat');
30     fid = fopen(fileName, 'r');
31     % Discard not needed data
32     fseek(fid, PrLine*15, 'bof');
33     % Read data
34     meas = fgetl(fid);
35     meas = sscanf(meas, '%g');
36     if meas > 0.0
37         visSat = [visSat, visSatInData(k)];
38     end
39     status = fclose(fid);
40 end
41
42 % Make combination of visible satellite
43 satComb = nchoosek(visSat, 4);
44 [numComb, n] = size(satComb);
45
46 PDOP = 1000;
47 for k = 1:numComb
48     % Read satellite position data for each combination
49     visCombSat = [satComb(k, :)];
50     newPDOP = FindPdop(visCombSat, nLine, xo, yo, zo);
51     if newPDOP < PDOP
52         PDOP = newPDOP;
53         SatComb = visCombSat;
54     end
55 end
56
57 % End of function

```

B.8 ArrangeComb.m

```

1 function SatComb = ArrangeComb(PreSatComb, SatComb);
2 % ArrangeComb arranges combination
3 %
4 % File name: ArrangeComb.m
5 % ArrangeComb arranges the placement of satellites in combination
6 % so as it reduces the tracking error
7 % The inputs are combination satellites before and after changing
8 % is needed.
9 % The output is satellites combination.
10 %
11 % Creator: Pande Hartana
12 % Created: Saturday, April 15, 2000
13 % Last Updated: Monday, May 22, 2000
14 % =====
15
16 indexChange = [];

```

```

17 indexSame = zeros(1, 4);
18 dummSatComb = zeros(1, 4);
19 for k = 1:4
20     elSame = 0;
21     for l = 1:4
22         if ((PreSatComb(k)) == (SatComb(l)))
23             indexSame(l) = 1;
24             dummSatComb(k) = PreSatComb(k);
25             l = 4;
26             elSame = 1;
27         end
28         if ((l == 4) & (elSame == 0))
29             indexChange = [indexChange, k];
30         end
31     end
32 end
33 [n, numChange] = size(indexChange);
34 for k = 1:numChange
35     for l = 1:4
36         if (indexSame(l) == 0)
37             indexSame(l) = 1;
38             dummSatComb(indexChange(k)) = SatComb(l);
39             break;
40         end
41     end
42 end
43 SatComb = dummSatComb;
44
45 % End of function

```

B.9 SatVis.m

```

1 function NotVis = SatVis(visCombSat, PrLine)
2 % SatVis checks satellites visibility
3 %
4 % File name: SatVis.m
5 % SatVis checks if all the satellites in combination
6 % are still visible for any particular time
7 % The inputs are combination satellites used in measurement
8 % and current time.
9 % The output is boolean:
10 % 0 = True, Still visible
11 % 1 = False, Not Visible
12 %
13 % Creator: Pande Hartana
14 % Created: Saturday, April 15, 2000
15 % Last Updated: Monday, May 22, 2000
16 % =====
17
18 NotVis = 0;
19 for k = 1:4
20     fileName = strcat('DNG', int2str(visCombSat(k)), 'C1.dat');
21     fid = fopen(fileName, 'r');
22     % Discard not needed data
23     fseek(fid, PrLine*15, 'bof');
24     % Read data
25     meas = fgetl(fid);
26     pRange(k, 1) = sscanf(meas, '%g');
27     if pRange(k, 1) == 0
28         NotVis = 1;
29         break;
30     end
31     status = fclose(fid);
32 end
33
34 % End of function

```

B.10 CreateNewSatComb.m

```
1 function [SatComb, PDOP] = CreateNewSatComb(nLine, PrLine, xo, yo, zo,...
2 NotVisMat, CurSatComb)
3 % CreateNewSatComb create new satellites combination
4 %
5 % File name: CreateNewSatComb.m
6 % CreateNewSatComb create new satellites combination
7 % and minimize the change of satellites in combination.
8 % The input is the number of data that will be discarded,
9 % approximated vehicle position (xo, yo, zo), satellite
10 % visibility matrix, and current satellites combination.
11 % The output is the satellites combination.
12 %
13 % Creator: Pande Hartana
14 % Created: Saturday, April 15, 2000
15 % Last Updated: Friday, April 21, 2000
16 % =====
17
18 % Find current satellites which are not need to be replaced
19 NoReplace = [];
20 needChange = 4;
21 for k = 1:4
22     if (NotVisMat(k) == 0)
23         NoReplace = [NoReplace, CurSatComb(k)];
24         needChange = needChange - 1;
25     end
26 end
27
28 % Open and read satellite position files.
29 visSatInData = [2, 4, 5, 6, 7, 8, 9, 10, 13, 14, 16, 18, 24, 26, 30];
30
31 % Read pseudorange data for particular line.
32 % If pseudorange is equal to zero, that means satellite is not visible.
33 visSat = [];
34 [n, numSat] = size(visSatInData);
35 for k = 1:numSat
36     fileName = strcat('DNG', int2str(visSatInData(k)), 'C1.dat');
37     fid = fopen(fileName, 'r');
38     % Discard not needed data
39     fseek(fid, PrLine*15, 'bof');
40     % Read data
41     meas = fgetl(fid);
42     meas = sscanf(meas, '%g');
43     if meas > 0.0
44         visSat = [visSat, visSatInData(k)];
45     end
46     status = fclose(fid);
47 end
48
49 % Create new vector consists of visible satellites which are not
50 % in current combination
51 newSatVec = setdiff(visSat, NoReplace);
52
53 % Create combination from new satellite vector consists
54 % of needChange members.
55 newSatVecComb = nchoosek(newSatVec, needChange);
56 [numComb, n] = size(newSatVecComb);
57
58 % Calculate combination with smallest PDOP
59 PDOP = 1000;
60 for k = 1:numComb
61     visCombSat = [newSatVecComb(k, :)];
62     ind = 0;
63     for l = 1:4
64         if (NotVisMat(l) == 1)
65             ind = ind + 1;
```


NOTE TO USERS

Page not included in the original manuscript is unavailable from the author or university. The manuscript was microfilmed as received.

144

This reproduction is the best copy available.

UMI

B.12 ChooseNewSatComb.m

```
1 function [SatComb, PDOP] = ChooseNewSatComb(nLine, PrLine, ...
2 xo, yo, zo, PreSatComb)
3 % ChooseNewSatComb chooses new satellites combination.
4 %
5 % File name: ChooseNewSatComb.m
6 % ChooseNewSatComb chooses a combination of satellites, which
7 % gives the lowest PDOP and minimize the change of the satellites
8 % order in combination.
9 % The inputs are the number of data that will be discarded,
10 % approximated vehicle position (xo, yo, zo), and cuurent
11 % satellites combination.
12 % The output is the satellites combination.
13 %
14 % Creator: Pande Hartana
15 % Created: Saturday, April 15, 2000
16 % Last Updated: Friday, April 21, 2000
17 % =====
18
19 % Open and read satellite position files.
20 visSatInData = [2, 4, 5, 6, 7, 8, 9, 10, 13, 14, 16, 18, 24, 26, 30];
21
22 % Read pseudorange data for particular line.
23 % If pseudorange is equal to zero,
24 % that means satellite is not visible.
25 visSat = [];
26 [n, numSat] = size(visSatInData);
27 for k = 1:numSat
28     fileName = strcat('DNG', int2str(visSatInData(k)), 'C1.dat');
29     fid = fopen(fileName, 'r');
30     % Discard not needed data
31     fseek(fid, PrLine*15, 'bof');
32     % Read data
33     meas = fgetl(fid);
34     meas = sscanf(meas, '%g');
35     if meas > 0.0
36         visSat = [visSat, visSatInData(k)];
37     end
38     status = fclose(fid);
39 end
40
41 % Make combination of visible satellite
42 satComb = nchoosek(visSat, 4);
43 [numComb, n] = size(satComb);
44
45 PDOP = 1000;
46 for k = 1:numComb
47     % Read satellite position data for each combination
48     visCombSat = [satComb(k, :)];
49     visCombSat = ArrangeComb(PreSatComb, visCombSat);
50     newPDOP = FindPdop(visCombSat, nLine, xo, yo, zo);
51     if newPDOP < PDOP
52         PDOP = newPDOP;
53         SatComb = visCombSat;
54     end
55 end
56
57 % End of function
```

B.13 MeasMatrix.m

```
1  function H = MeasMatrix(visCombSat, nLine, xo, yo, zo)
2  % MeasMatrix measurement matrix
3  %
4  % File name: MeasMatrix.m
5  % MeasMatrix creates measurement matrix
6  % The inputs are visible satellites combination,
7  % current time, and current position.
8  % The output is the measurement matrix.
9  %
10 % Creator: Pande Hartana
11 % Created: Saturday, April 15, 2000
12 % Last Updated: Friday, April 21, 2000
13 % =====
14
15 % Speed of light
16 % (from IERS Technical Note 21, IERS Conventions 1996)
17 c = 299792458; % in meter/second
18
19 visSatNameArray = ['X', 'Y', 'Z', 'T'];
20
21 % Read satellite position data for each combination
22 for k = 1:4
23     for l = 1:4
24         fileName = strcat('Pos', int2str(visCombSat(k)), ...
25                             visSatNameArray(l), '.dat');
26         fid = fopen(fileName, 'r');
27         % Discard not needed data
28         fseek(fid, nLine*14, 'bof');
29         % Read data
30         meas = fgetl(fid);
31         data((k-1)*4 + 1) = 1000 * (sscanf(meas, '%g'));
32         status = fclose(fid);
33     end
34 end
35
36 % calculate satellite - aircraft geometric range
37 for k = 1:4
38     rangeToSat(k) = sqrt((data(4*k-3)-xo)^2 + ...
39                         (data(4*k-2)-yo)^2 + (data(4*k-1)-zo)^2);
40 end
41
42 % Linearization coefficients
43 for k = 1:4
44     a(k, 1) = -(data(4*k-3)-xo)/rangeToSat(k);
45     a(k, 2) = -(data(4*k-2)-yo)/rangeToSat(k);
46     a(k, 3) = -(data(4*k-1)-zo)/rangeToSat(k);
47     a(k, 4) = 1;
48 end
49
50 H = zeros(4, 8);
51 H(:, 1) = a(:, 1);
52 H(:, 3) = a(:, 2);
53 H(:, 5) = a(:, 3);
54 H(:, 7) = a(:, 4);
55
56 % End of function
```

B.14 HTranMat.m

```
1  function tranMat = HTranMat(LatiPos, LongPos)
2  % HTranMat transformation matrix
3  %
4  % File name: HTranMat.m
5  % HTranMat creates transformation matrix from Cartesian
6  % into local coordinate system.
7  % The inputs are latitude and longitude of vehicle position.
8  % The output is the transformation matrix.
9  %
10 % Creator: Pande Hartana
11 % Created: Saturday, April 15, 2000
12 % Last Updated: Sunday, May 14, 2000
13 % =====
14
15 % Build transformation matrix
16 sLati = sin(LatiPos);
17 cLati = cos(LatiPos);
18 sLong = sin(LongPos);
19 cLong = cos(LongPos);
20 R(1, 1) = -sLati;
21 R(1, 2) = -cLati*sLong;
22 R(1, 3) = cLati*cLong;
23 R(2, 1) = cLati;
24 R(2, 2) = -sLati*sLong;
25 R(2, 3) = sLati*cLong;
26 R(3, 1) = 0;
27 R(3, 2) = cLong;
28 R(3, 3) = sLong;
29
30 tranMat = zeros(8, 8);
31 tranMat(1, 1) = R(1, 1);
32 tranMat(1, 3) = R(1, 2);
33 tranMat(1, 5) = R(1, 3);
34 tranMat(2, 2) = R(1, 1);
35 tranMat(2, 4) = R(1, 2);
36 tranMat(2, 6) = R(1, 3);
37 tranMat(3, 1) = R(2, 1);
38 tranMat(3, 3) = R(2, 2);
39 tranMat(3, 5) = R(2, 3);
40 tranMat(4, 2) = R(2, 1);
41 tranMat(4, 4) = R(2, 2);
42 tranMat(4, 6) = R(2, 3);
43 tranMat(5, 1) = R(3, 1);
44 tranMat(5, 3) = R(3, 2);
45 tranMat(5, 5) = R(3, 3);
46 tranMat(6, 2) = R(3, 1);
47 tranMat(6, 4) = R(3, 2);
48 tranMat(6, 6) = R(3, 3);
49 tranMat(7, 7) = 1;
50 tranMat(8, 8) = 1;
51
52 % End of function
```

B.15 PredMeas.m

```
1  function Z = PredMeas(visCombSat, nLine, PrLine, xo, yo, zo)
2  % PredMeas predicted measurement
3  %
4  % File name: PredMeas.m
5  % PredMeas predicts the satellites range
6  % The inputs are combination satellites used in measurement,
7  % current time, and current vehicle position.
8  % The output is predicted measurement.
9  %
10 % Creator: Pande Hartana
11 % Created: Saturday, April 15, 2000
12 % Last Updated: Monday, May 22, 2000
13 % =====
14
15 visSatNameArray = ['X', 'Y', 'Z', 'T'];
16
17 % Read pseudorange
18 for k = 1:4
19     visSt = visCombSat(k);
20     fileName = strcat('DNG', int2str(visSt), 'C1.dat');
21     fid = fopen(fileName, 'r');
22     % Discard not needed data
23     fseek(fid, PrLine*15, 'bof');
24     % Read data
25     meas = fgetl(fid);
26     pRange(k, 1) = sscanf(meas, '%g');
27     status = fclose(fid);
28 end
29
30 % Read satellites position
31 for k = 1:4
32     for l = 1:4
33         fileName = strcat('Pos', int2str(visCombSat(k)), ...
34             visSatNameArray(l), '.dat');
35         fid = fopen(fileName, 'r');
36         % Discard not needed data
37         fseek(fid, nLine*14, 'bof');
38         % Read data
39         meas = fgetl(fid);
40         satPos((k-1)*4 + 1) = 1000 * sscanf(meas, '%g');
41         status = fclose(fid);
42     end
43 end
44
45 % Calculate the range between satellites and vehicle position
46 for k = 1:4
47     range(k, 1) = sqrt((satPos(4*k-3)-xo)^2 + ...
48         (satPos(4*k-2)-yo)^2 + (satPos(4*k-1)-zo)^2);
49 end
50
51 % Calculate the observable
52 Z = pRange - range;
53
54 % End of function
```