Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering

M.Tech. Dissertation

Submitted in fulfillment of the requirements

for the Dual Degree Program in Aerospace Engineering

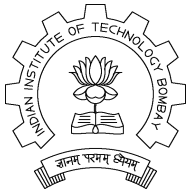
by

Vikas Kumar N.

99D01010

Under the guidance of

Prof. K. Sudhakar



DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

MUMBAI

July 2004

Acceptance Certificate

This is to certify that the dissertation titled, **Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering** by **Vikas Kumar N., Roll No. 99D01010** has been approved towards the fulfillment of the requirements for the Dual Degree Program.

Guide

_____

External Examiner

_____

Internal Examiner

_____

Chairman

_____

Date: _____

ACKNOWLEDGEMENT

I take this opportunity to express my deep appreciation and gratitude to my guide, **Prof. K. Sudhakar** for his valuable insights, guidance and patience which got me through the project. I would also like to thank **Prof. Arya** for his help during my project. I would also like to express my gratitude to **Mr. S. Bhaktavatsala** for making the hardware and helping me run my program on it.

Vikas Kumar N.

99D01010

July 5, 2004

ABSTRACT

Inertial navigation blended with other navigation aids like GPS, has gained significance due to enhanced navigation and inertial reference performance. The INS, individually can calculate the position of the aircraft without any help from the outside world. However, a large number of errors are introduced by sensors leading to an unacceptable drift in the output. Hence a GPS is used to aid the INS, using a Kalman filter which helps in estimating the errors in the INS and thus updating position to improved accuracy.

The simulation of the integration of the INS and GPS using Kalman filtering has been completed using MATLAB and C. This has been tested on the simulator for the target hardware. The details have been explained in the report.

Contents

List of Figures

List of Tables

Nomenclature

α	Angle of attack
β	Sideslip angle
γ	Gravity vector
ϵ	Attitude error
λ	Latitude
μ	Longitude
ϕ	Bank angle
Φ_k	Discrete state transition matrix
θ	Pitch angle
ψ	Yaw or heading angle
σ	Standard deviation
$\boldsymbol{\omega}$'	Additional angular velocity due to rotation of navigational frame
$\boldsymbol{\Omega}$	Angular velocity of the earth (15°/hour)
a_x, a_y, a_z	Accelerations along the 3 body axes
A	$n \times n$ matrix
Acc_{output}	Voltage given out by the accelerometer
B	$n \times 1$ matrix
c	Uncertainty in the scale factor of the accelerometer
$\mathbf{C}_b^n$	Direction cosine matrix from navigation frame to body frame
C, C_c	Scale factor of the accelerometer

d	Uncertainty in the bias of the accelerometer
D, D_c	Bias of the accelerometer
DCM	Direction cosine matrix
e_0, e_1, e_2, e_3	Euler parameters
$\hat{\mathbf{e}}_k^-$	Estimation error at time instant k
$\mathbf{E}^n$	Skew symmetric 3×3 matrix of attitude errors in navigation frame
$\mathbf{f}$	Aircraft body acceleration vector
F	9×9 state transition matrix
F_{lat}, F_{lon}	Distance corresponding to a degree change in latitude or longitude
g	Acceleration due to gravity
G	9×6 design matrix
h	Altitude
H_k	$m \times n$ matrix at time instant k, 3×9 matrix at time instant k
$\mathbf{I}$	Identity matrix
$\mathbf{K}_k$	Kalman gain at time instant k
p	Roll rate
$\mathbf{P}_k$	Error covariance matrix at time instant k
q	Pitch rate
$\mathbf{Q}$	Spectral density matrix
$\mathbf{Q}_k$	Process noise covariance at time instant k
r	Yaw rate
$\mathbf{r}$	Position vector
$\mathbf{R}_k$	Measurement noise covariance at time instant k
R_e	Radius of the Earth
t	Time
U	Velocity along the body X axis
$\mathbf{u}_k$	Input or driving function at time instant k
V	Velocity along the body Y axis

$\mathbf{v}$ Velocity vector

$\mathbf{v}_k$ Measurement noise at time instant k

V_D Velocity along the Down axis in the navigation frame

V_E Velocity along the East axis in the navigation frame

V_N Velocity along the North axis in the navigation frame

V_T Magnitude of the velocity of the aircraft

$\mathbf{w}_k$ Process noise at time instant k

W Velocity along the body Z axis

X Position along the North axis in the navigation frame

$\mathbf{x}_k$ State vector at time instant k

$\hat{\mathbf{x}}_k^-$ *a priori* estimate of state vector at time instant k

$\hat{\mathbf{x}}_k$ Updated estimate

Y Position along the East axis in the navigation frame

Z Position along the Down axis in the navigation frame

$\mathbf{z}_k$ Measurement vector at time instant k

Chapter 1

Introduction

For automatic machines, be it robots, aircraft or other autonomous vehicles, navigation is of utmost importance. Various systems are used in navigation of aircraft, viz. inertial navigation systems (INS), global positioning systems (GPS), air-data dead reckoning systems, radio navigation systems, Doppler heading reference systems, to name a few. Our interest lies in integrating both the INS and the GPS to provide the best possible estimate of the aircraft position in terms of the latitude, longitude and height above the surface of the earth.

The INS gives us the position, velocity and attitude of the aircraft but it is inundated with errors due to the fact that any small bias error can grow the error with time. Hence, an update or position fix is taken from the GPS and using a Kalman filter we can estimate the errors in both the INS and the GPS thus giving the user a better position information.

Applications are not limited to aircraft alone. Although these integrated systems find extensive usage in airborne vehicles, they have also been used in the navigation of cars, ships and satellites.

There are considerable advantages in developing this kind of a navigation system as compared to the ones used earlier in terms of compactness and speed. Micro-

gyroscopes and GPS chips can be integrated on a small board and can effectively give the position of the vehicle concerned. With the advent of MEMS technology, all this can be done at extremely high levels of accuracy and at lower costs.

Our aim is to develop the GPS-INS integrated system so that it can be implemented on realtime hardware like a microcontroller or a digital signal processor. Even though high accuracy sensors like gyroscopes and accelerometers are available, their costs are on the higher side. Usage of low cost and low accuracy sensors may find application where high accuracy is not required. Initially the simulation of the whole navigation would be done on a computer, where given the initial state of the aircraft and regular updates from the sensors and the GPS, the program would return the estimated position of the aircraft. Eventually this simulated model would be implemented on realtime hardware.

The next chapter describes in brief some of the INS/GPS systems which have been integrated and implemented on hardware. Chapter 3 gives an overview of the working of each of the subsystems, i.e. the INS and the GPS followed by a theoretical explanation of Kalman filtering in general. Chapter 4 describes how the simulation has been done, viz. the INS programming, sensor modelling, GPS modelling and the Kalman filter modelling. In this chapter we also see why the Kalman filtered output is better than the output of the individual subsystems. The report concludes with the chapter discussing the hardware used to run the program as well as the issues associated with the working of the whole system.

Chapter 2

Literature Survey

Several GPS-INS integration techniques have been implemented. Some of them are described here briefly.

Schmidt [1] describes in detail the computations for a gimballed INS and the strapdown INS. The 9 state Kalman filter which is discussed by him uses a barometric altimeter to correct for the height. Bar-Itzhack et al [2] describes a control theoretical approach to INS/GPS integration using Kalman filtering in his paper. The psi-angle error model explained has been used extensively in many models of the INS/GPS Kalman filter and also has been used in this project. Grewal et al [3, 4] have discussed in their books the working of the INS, GPS and Kalman filtering in detail and have given a complicated model of a possible Kalman filter with 54 states.

Wolf et al [5] use Systron Donner's MotionPak inertial measurement unit (IMU) and a Trimble Advanced Navigation Sensor (TANS) Vector receiver system (as the GPS component), which is a multi-antenna, attitude determination and position location system. They have developed a real-time navigation software to calculate position, velocity and attitude from the outputs of the MotionPak gyroscopes and accelerometers. Besides giving position and velocity updates, TANS also gives attitude measurement data. A Kalman filter with 27 states has been implemented by them. A Packard Bell 486 computer was used to carry out the computations.

Grejner-Brzezinska et al [6] have tested the feasibility of attitude estimation improvement by using high accuracy deflection of vertical (DOV) information in the integrated GPS/INS navigation system. The estimability of attitude components improves by adding partially compensated gravity information. A fully digital Airborne Integrated Mapping System (AIMS) has been designed and the integrated INS/GPS forms an integral part of this AIMS. A dual frequency differential GPS (DGPS) and the Litton LN100 IMU are tightly integrated. IMU data was updated at 256Hz. A centralized 21 state Kalman filter was used to process the GPS L1/L2 signals as well as the errors from the INS. Accuracy of order of 10cm was achieved.

Srikumar and Deori [7] have used an airdata based dead reckoning system to calculate position of their MAV. Navigation accuracy has been improved by using updates from a GPS receiver as well as a ground-based tracking system. A Pentium 90MHz personal computer has been used to control navigation and many other features of the aircraft.

Randle and Horton [8] describe in their works the integration of GPS/INS using a low cost IMU consisting of micro-machined sensors and on-board calibration. Simulations have been done for both flight and automotive navigation. With complete loss of GPS signals, position accuracy is shown to be less than 10m after 30s. Kalman filter has 23 states and a DGPS was simulated to give measurement updates.

Navigation for reusable launch vehicles has been studied by Gaylor et al [9]. A number of navigation sensors have been studied and the GPS/INS integrated system was settled for. Error models of INS and GPS operating in the vicinity of the International Space Station have been developed. Effects of signal blockage and multipath errors of the GPS have been modelled. An extended Kalman filter with 18 states was developed.

Brown and Sullivan [10] have described test results for a system that uses an improved kinematic alignment algorithm suite providing high quality navigation solution using direct carrier-phase and pseudo-range GPS measurements, tightly coupled

with measurements from a low cost IMU system. A 32 state Kalman filter has been implemented using InterNav software made by Navsys Corporation.

Moon et al [11] first process the GPS values before sending it to their 9 state Kalman filter. Honeywell's HG1700 IMU has been used along with a Motorola UT Oncore GPS, which can track 8 GPS satellites simultaneously. Salychev et al [12] use MotionPak IMU and integrate it with GPS (GPS - GLONASS) and DGPS information, to provide navigation capability to bridge GPS outages for tens of seconds. Airborne and ground tests have been conducted and robustness of the system has been studied. IMU data was recorded at 46Hz.

A 21 state filter using tightly coupled integration scheme was implemented by Wang et al [13] and two digital signal processors were used to carry out the computations. A sensitivity performance has been analysed in their research paper.

Kwon [14] has studied airborne gravimetry and compared it to ground measurement of gravity. The combination of GPS/INS is known to show very good performances for recovering the gravity signal. A new algorithm using acceleration updates instead of conventional position or velocity updates has been developed. It is computationally proven to be less expensive since navigation equations need not be integrated. Real flight data has been tested upon the system desgined by Kwon and the simulations have been discussed in his works in detail. A comparative study of the Kalman filter using traditional approach and the new approach has been done.

Ronnback [15] has tested his INS/GPS navigation filter written in C++ on an air vehicle. A redundant 4 axes IMU called Tetrad has been used. A 9 state Kalman filter was implemented with measurements of position and velocity from the GPS. Gautier [16] has designed GPS INS generalized evaluation tool (GIGET) which aids in the selection of sensor combinations for any general application or set of requirements. It includes a unique five antenna, forty channel GPS receiver providing attitude, position and timing. Honeywell's HG1700 tactical grade IMU is integrated with this GPS using a 21 state extended Kalman filter and tested on their homemade Dragonfly

unmanned air vehicle.

In his work, Mayhew [17] proposes several methods for improving the position estimation capabilities of a system by incorporating other sensor and data technologies, including Kalman filtered intertial navigational systems, rule-based and fuzzy-based sensor fusion techniques, and a unique map-matching algorithm. Ground testing of the system has been done. Dead reckoning sensors are used to collect odometry data. A Pentium 133MHz computer was used to carry out all the computations and control handling of the aircraft. The 9 state Kalman filter was run every time a measurement update from the INS took place.

Moore and Qi [18] have implemented a direct Kalman filtering technique to integrate their GPS and INS systems, where they use two stage GPS filtering to preprocess the GPS data before the Kalman filter can use it. Their eight state direct Kalman filter uses the position and velocity as its state vector. Shang et al [19] use two GPS receivers to not only estimate the position but also the azimuth alignment. They use a PC/104 microcomputer to carry out their computations. They have carried out a tightly coupled implementation of the Kalman filter.

Cao et al [20] have implemented a 15 state Kalman filter successfully using MEMS based sensors for intelligent transport systems. They have used a strapdown INS system. Panzieri et al [21] have implemented a 5 state extended Kalman filter to manouever a robot's movements. They use a GPS because the usage of the robot is outdoors.

Dorobantu and Zebhauser [22] have implemented an extended Kalman filter of five states for a 2-D case as they are using it on a land vehicle. They use a DGPS to get measurement updates every 1s and have discussed results when there are regular GPS outages.

Chapter 3

INS,GPS, Kalman filtering

Today's trend in navigation sees the rise of integrated navigation systems, where the components (sensors) that are usually integrated are the Inertial Navigation Systems (INS) and the Global Positioning System (GPS). The integration of two subsystems provides more accuracy than that of individual subsystems.

3.1 Inertial Navigation Systems

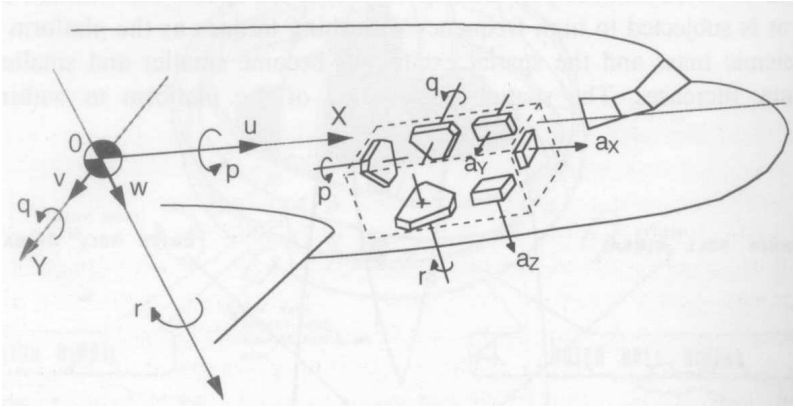


Figure 3.1: Orientation of axes

The INS consists of 3-axis gyroscopes which give the system computer the roll,

pitch and yaw rates about the body axes as shown in figure 3.1 [23]. It also has 3-axis accelerometers which give the accelerations along the three body axes. There are two basic inertial mechanisms which are used to derive the Euler angles from the rate gyros, viz. stable platform and strap-down INS. We would be concerned with the strap-down INS where the gyros and accelerometers are 'strapped-down' to the aircraft body frame. The acceleration values from the accelerometers are then corrected for rotation of the earth and gravity to give the velocity and position of the aircraft.

3.1.1 Equations of Motion

The orientation of an aircraft with respect to a fixed inertial frame of axes is defined by three Euler angles. The aircraft is imagined to be oriented parallel to the fixed reference frame of axes. A series of rotations bring it to the orientation about axes OX, OY and OZ, as shown in figure 3.2 [23]:

1. clockwise rotation about the yaw axis, through the yaw (or heading) angle ψ , followed by
2. a clockwise rotation about the pitch axis, through the pitch angle θ, followed by
3. a clockwise rotation about the roll axis, through the bank angle ϕ.

The relationship between the angular rates of roll, pitch and yaw, p, q, r (measured by the body mounted gyros), the Euler angles, ψ, θ, ϕ and their rates, is given below.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.1}$$

By integration of the above equations we can derive the Euler angles using initial conditions of a known attitude at a given time. But, for pitch angles around $\pm 90°$,

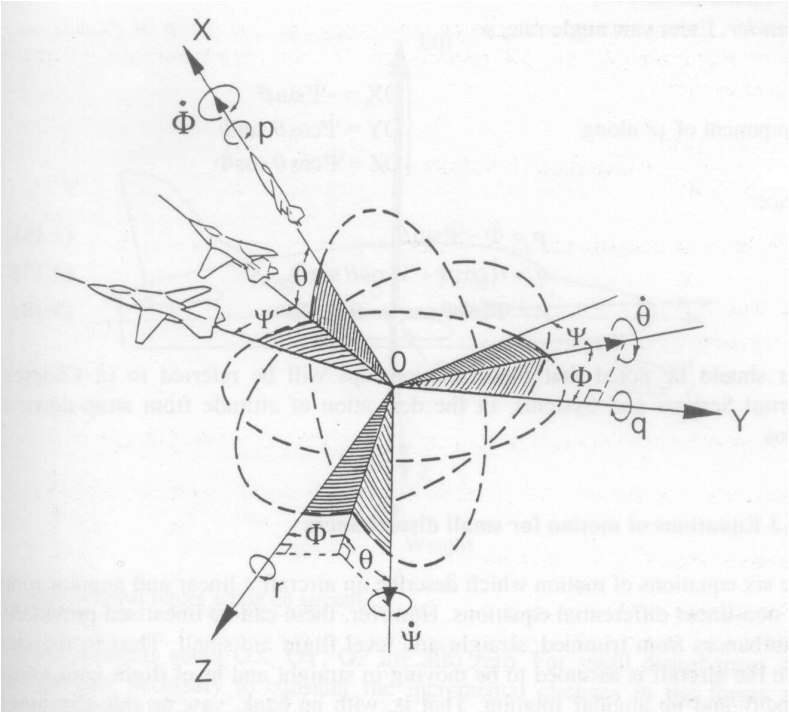


Figure 3.2: Euler Angles

the error becomes unbounded as $\tan\theta$ tends to infinity. Quaternion algebra comes to the rescue here. We use four parameters, called the Euler parameters, that are related to the Euler angles as follows [24].

If e_0, e_1, e_2, e_3 were the four parameters then in terms of angular rates, we have

$$\dot{e_0} = -\frac{1}{2}(e_1 p + e_2 q + e_3 r) \tag{3.2}$$

$$\dot{e_1} = \frac{1}{2}(e_0 p + e_2 r - e_3 q) \tag{3.3}$$

$$\dot{e_2} = \frac{1}{2}(e_0 q + e_3 p - e_1 r) \tag{3.4}$$

$$\dot{e_3} = \frac{1}{2}(e_0 r + e_1 q - e_2 p) \tag{3.5}$$

with the parameters satisfying the following equation at all points of time.

$$e_0{}^2 + e_1{}^2 + e_2{}^2 + e_3{}^2 = 1 \tag{3.6}$$

The above equations can be used to generate the time history of the four parameters e_0, e_1, e_2, and e_3. The initial values of the Euler angles are given which are used to

calculate the initial values of the four parameters using the following equations.

$$e_0 = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} \tag{3.7}$$

$$e_1 = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} \tag{3.8}$$

$$e_2 = \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} \tag{3.9}$$

$$e_3 = -\cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} \tag{3.10}$$

Once we have calculated the time history of the four parameters, we can calculate the Euler angles using the following equations.

$$\theta = \sin^{-1}[-2(e_1e_3 - e_0e_2)] \tag{3.11}$$

$$\phi = \cos^{-1}\left[\frac{{e_0}^2 - {e_1}^2 - {e_2}^2 + {e_3}^2}{\sqrt{1-4(e_1e_3 - e_0e_2)^2}}\right]\text{sign}[2(e_2e_3 + e_0e_1)] \tag{3.12}$$

$$\psi = \cos^{-1}\left[\frac{{e_0}^2 + {e_1}^2 - {e_2}^2 - {e_3}^2}{\sqrt{1-4(e_1e_3 - e_0e_2)^2}}\right]\text{sign}[2(e_1e_2 + e_0e_3)] \tag{3.13}$$

We now have with us the attitude of the aircraft. To calculate the position we use the accelerations given by the accelerometers.

The accelerations (a_x, a_y and a_z) of the aircraft along the three body axes, as read by the accelerometers, are given by the equations 3.14 - 3.16. U, V, W and p, q, r are all available as states. If the acceleration due to gravity (g) model is supplied as a function of location around the earth, then $\dot{U}$, $\dot{V}$ and $\dot{W}$ can be calculated.

$$\dot{U} = a_X + Vr - Wq + g\sin\theta \tag{3.14}$$

$$\dot{V} = a_Y - Ur + Wp - g\cos\theta\sin\phi \tag{3.15}$$

$$\dot{W} = a_Z + Uq - Vp - g\cos\theta\cos\phi \tag{3.16}$$

The earth is rotating in space at a rate $\mathbf{\Omega}$ (15° per hour) around an axis South to North as shown in figure 3.3.

$$\mathbf{\Omega} = \begin{bmatrix} \Omega\cos\lambda \\ 0 \\ -\Omega\sin\lambda \end{bmatrix} \tag{3.17}$$

The motion of the vehicle at a constant height above the ground will induce an additional rotation given by

$$\boldsymbol{\omega'} = \begin{bmatrix} \dot{\mu}\cos\lambda \\ -\dot{\lambda} \\ -\dot{\mu}\sin\lambda \end{bmatrix} \tag{3.18}$$

The measured angular rates include $\boldsymbol{\Omega}$ and $\boldsymbol{\omega'}$, we have the actual angular rates given

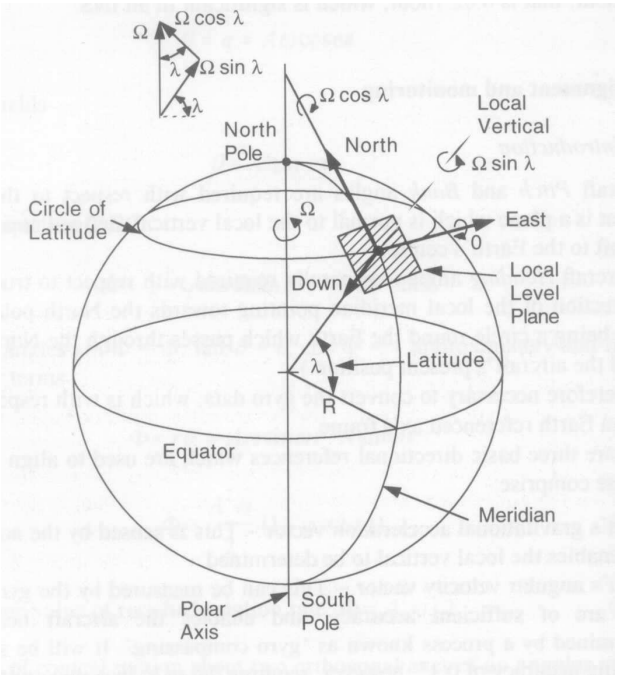


Figure 3.3: Local earth frame or Navigation frame

by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_m - \mathbf{DCM} \begin{bmatrix} \boldsymbol{\Omega} + \boldsymbol{\omega'} \end{bmatrix} \tag{3.19}$$

where **DCM** is the the direction cosine matrix or the transformation matrix, from the local earth or navigation frame to the body frame, given by equation 3.20, $\dot{\mu}$ is

the rate of change of longitude and $\dot{\lambda}$ is the rate of change of latitude. .

$$\mathbf{DCM} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\phi\cos\theta \\ \sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi & \sin\phi\sin\theta\cos\phi - \cos\psi\sin\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3.20}$$

$\dot{U}$, $\dot{V}$ and $\dot{W}$ are integrated to calculate the velocity components (U, V and W), which are then transformed using the direction cosine matrix (equation 3.20) to give velocity along North (V_N), velocity along East (V_E) and downward velocity (V_D) in the navigation frame or local earth frame, as shown in figure 3.3.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} = \mathbf{DCM}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.21}$$

V_N, V_E and V_D are then integrated to give distances moved along the navigation axes (X, Y, Z) on the surface of the earth. Let λ, μ and H denote the latitude, longitude and height of the aircraft at any instant, then rate of change of latitude [23, 25] is given by

$$\dot{\lambda} = \frac{V_N}{R_e} \tag{3.22}$$

and rate of change of longitude is given by

$$\dot{\mu} = \frac{V_E}{R_e \cos\lambda} \tag{3.23}$$

where R_e is the radius of the earth. The rate of change of altitude of the aircraft is given by

$$\dot{H} = -V_D \tag{3.24}$$

The position of the aircraft in terms of latitude, longitude and altitude can be thus calculated using equations 3.22, 3.23 and 3.24.

3.1.2 Errors in the INS

Most INS errors are attributed to the inertial sensors (instrument errors). These are the residual errors exhibited by the installed gyros and accelerometers following calibration of the INS. The dominant error sources are shown in table 3.1[6, 26].

Table 3.1: Sensor generated errors in the INS

Alignment errors	roll, pitch and heading errors
Accelerometer bias or offset	a constant offset in the accelerometer output that changes randomly after each turn-on.
Accelerometer scale factor error	results in an acceleration error proportional to sensed acceleration.
Nonorthogonality of gyros and accelerometers	the axes of accelerometer and gyro uncertainty and misalignment.
Gyro drift or bias (due to temperature changes)	a constant gyro output without angular rate presence.
Gyro scale factor error	results in an angular rate error proportional to the sensed angular rate
Random noise	random noise in measurement

Errors in the accelerations and angular rates lead to steadily growing errors in position and velocity components of the aircraft, due to integration. These are called navigation errors and there are nine of them – three position errors, three velocity errors, two attitude errors and one heading error. If an unaided INS is used, these errors grow with time. It is for this reason that the INS is usually aided with either GPS, Doppler heading sensor or air-data dead reckoning systems. Gravity model can also cause some errors. The acceleration due to gravity varies from place to place along the earth and also with height. These errors have to be modelled accordingly.

Inertial sensors for strapdown systems experience much higher rotation as compared to their gimballed counterparts. Rotation introduces error mechanisms that

require attitude rate-dependent error compensation.

3.2 Global Positioning System

3.2.1 Introduction

GPS uses a one-way ranging technique from the GPS satellites that are also broadcasting their estimated positions. Signals from four satellites are used with the user generated replica signal and the relative phase is measured. Using triangulation the location of the receiver is fixed. Four unknowns can be determined using the four satellites and appropriate geometry : latitude, longitude, altitude and a correction to the user's clock. The GPS receiver coupled with the receiver computer returns elevation angle between the user and satellite, azimuth angle between the user and satellite, measured clockwise positive from the true north, geodetic latitude and longitude of the user.

The GPS ranging signal is broadcast at two frequencies : a primary signal at 1575.42 MHz (L_1) and a secondary broadcast at 1227.6 MHz (L_2). Civilians use L_1 frequency which has two modulations, viz. C/A or Clear Acquisition (or Coarse Acquistion) Code and P or Precise or Protected Code. C/A is unencrypted signal broadcast at a higher bandwidth and is available only on L_1. P code is more precise because it is broadcast at a higher bandwidth and is restricted for military use. The military operators can degrade the accuracy of the C/A code intentionally and this is known as Selective Availability. Ranging errors of the order of 100m can exist with Selective Availability. There are six major causes of ranging errors : satellite ephemeris, satellite clock, ionospheric group delay, tropospheric group delay, multipath and receiver measurement errors, including software.

The primary role of GPS is to provide highly accurate position and velocity worldwide, based on range and range-rate measurements. GPS can be implemented in navigation as a fixing aid by being a part of an integrated navigation system, for

example INS/GPS.

3.2.2 Errors in GPS

Ephemeris errors occur when the GPS message does not transmit the correct satellite location and this affects the ranging accuracy. These tend to grow with time from the last update from the control station. Satellite clock errors affect both C/A and P code users and leads to an error of 1-2m over 12hr updates [27]. Measurement noise affects the position accuracy of GPS pseudorange absolute positioning by a few meters. The propagation of these errors into the position solution can be characterized by a quantity called Dilution of Precision (DOP) which expresses the geometry between the satellite and the receiver and is typically between 1 and 100. If the DOP is greater than 6, then the satellite geometry is not good. Ionospheric and tropospheric delays are introduced due to the atmosphere and this leads to a phase lag in calculation of the pseudorange.These can be corrected with a dual-frequency P-code receivers. Multipath errors are caused by reflected signals entering the front end of the receiver and masking the correlation peak. These effects tend to be more prominent due to the presence of reflective surfaces, where 15m or more in ranging error can be found in some cases.

3.3 Kalman Filtering

The Kalman Filter (KF) is a very effective stochastic estimator for a large number of problems, be it in computer graphics or in navigation. It is an optimal combination, in terms of minimization of variance, between the prediction of parameters from a previous time instant and external observations at a present time instant.

3.3.1 Discrete Kalman Filter

The KF addresses the general problem of trying to estimate the state $\mathbf{x} \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation [3, 4, 28]

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_k + \mathbf{w}_{k-1} \tag{3.25}$$

with a measurement $\mathbf{z} \in \Re^m$ that is

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k. \tag{3.26}$$

The variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise respectively. They are assumed to be independent of each other, white, and with normal probability distributions

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}), \tag{3.27}$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}). \tag{3.28}$$

$\mathbf{Q}$ is the process noise covariance and $\mathbf{R}$ is the measurement noise covariance. Equation 3.25 is similar to the standard state differential equation

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{3.29}$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the input or driving function, the only difference being that equation 3.25 is a system whose state vector is sampled for discrete time state, whereas equation 3.29 is sampled for continuous time state.

The $n \times n$ matrix A in the difference equation 3.25, relates the state at the previous time step $k-1$ to the state at the current time step k, in the absence of a driving function or a process noise. The $n \times 1$ matrix B relates the optional control input $\mathbf{u} \in \Re^n$ to the state $\mathbf{x}$. The $m \times n$ matrix H in equation 3.26 relates the state $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$.

An initial estimate of the process at some point t_k is assumed, and this estimate is based on our knowledge of the process prior to t_k. Let this *a priori* estimate be denoted by $\hat{\mathbf{x}}_k^-$, where the "hat" denotes estimate, and the "super minus" reminds us

that this is the best estimate we have prior to assimilating the measurement at t_k. Assuming that the error covariance matrix associated with $\hat{\mathbf{x}}_k^-$ is also known, then the estimation error is defined as

$$\hat{\mathbf{e}}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \tag{3.30}$$

and the associated covariance matrix as

$$\mathbf{P}_k{}^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}^-)(\mathbf{x}_k - \hat{\mathbf{x}}^-)^T] \tag{3.31}$$

Since we have assumed a prior estimate $\hat{\mathbf{x}}_k^-$, we use $\mathbf{z}_k$ to improve the prior estimate, by the following equation.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{3.32}$$

where $\hat{\mathbf{x}}_k$ is the updates estimate and $\mathbf{K}_k$ is the blending factor or Kalman gain that minimizes the *a posteriori* error covariance equation 3.33.

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}})(\mathbf{x}_k - \hat{\mathbf{x}})^T] \tag{3.33}$$

Substituting equation 3.26 into 3.32 and then substituting the resulting expression into 3.33 we get

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{3.34}$$

where the Kalman gain which minimizes the mean-square estimation error is given by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \tag{3.35}$$

We then estimate the next step measurement $\hat{\mathbf{x}}_{k+1}^-$, the error covariance $\mathbf{P}_{k+1}^-$ and repeat the process.

$$\hat{\mathbf{x}}_{k+1}^- = A_k\hat{\mathbf{x}}_k + B_k\mathbf{u}_k \tag{3.36}$$

$$\mathbf{P}_{k+1}^- = A_k\mathbf{P}_k A_k^T + \mathbf{Q}_k \tag{3.37}$$

We ignore the contribution of $\mathbf{w}_k$ because it is a zero mean function and not correlated with the earlier $\mathbf{w}$'s.

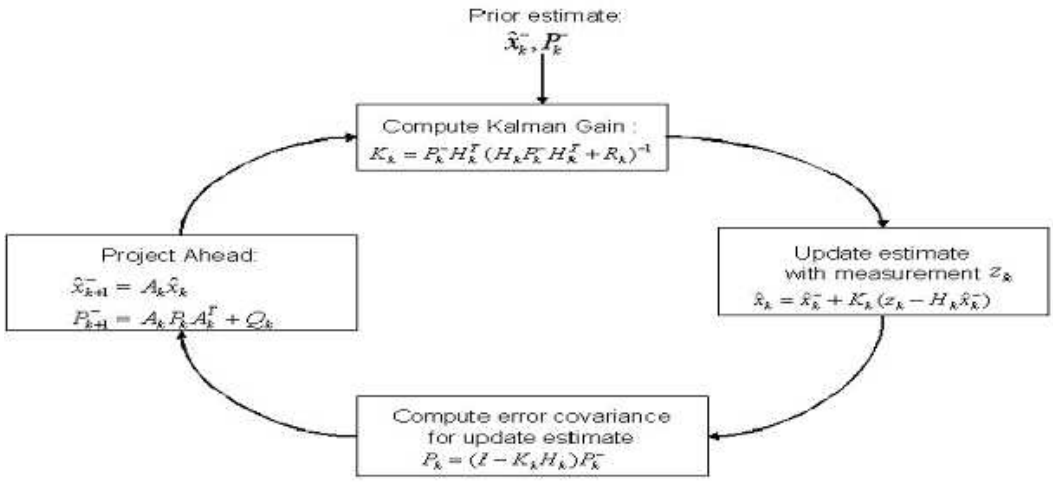


Figure 3.4: The Kalman filter loop

3.3.2 Kalman filter and navigation

KF is an extremely effective and versatile procedure for combining noisy sensor outputs to estimate the state of a system with uncertain dynamics. Noisy sensor outputs include outputs from the GPS and INS; state of the system may include position, velocity, attitude and attitude rate of a vehicle or an aircraft; and uncertain dynamics includes unpredictable disturbances in the sensor parameters or disturbances caused by a human operator or a medium (like wind).

The KF is used to estimate the errors introduced into the unaided INS system due to the gyros and accelerometers as discussed in table 3.1. These errors form the state vector $\hat{\mathbf{x}}_k$ and the measured values of the state vector from the GPS forms the measurement vector $\mathbf{z}$. Once the errors are modelled, the KF loop, as shown in figure 3.4, is implemented after giving the initial estimates of the state vector and its covariance matrix at time $t = 0$. This is the GPS-aided INS system configuration, and the errors are either compensated by the feedforward or the feedback mechanism as shown in figures 3.5 and 3.6.

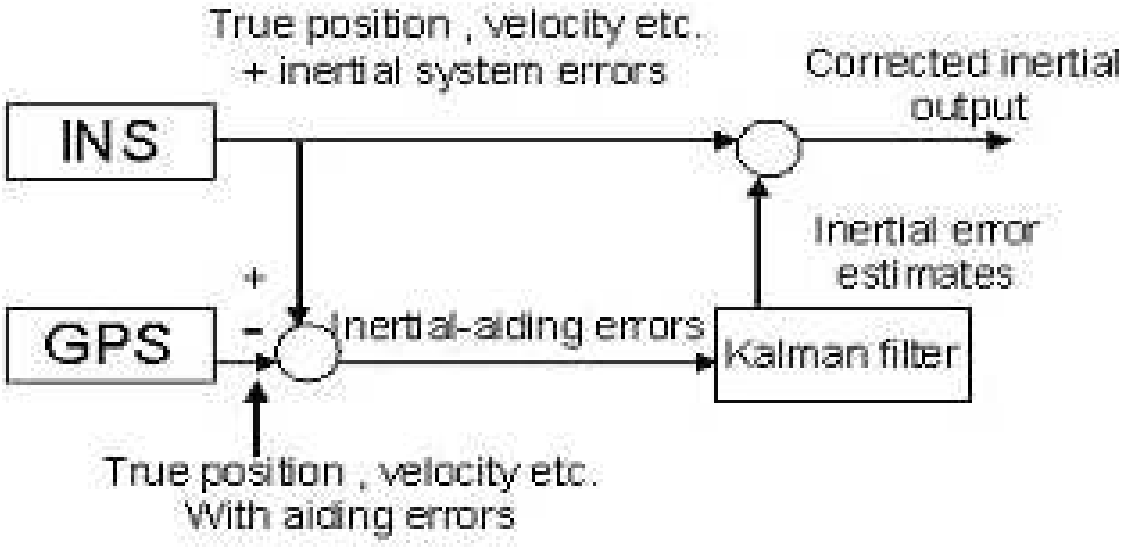


Figure 3.5: Feedforward aided INS

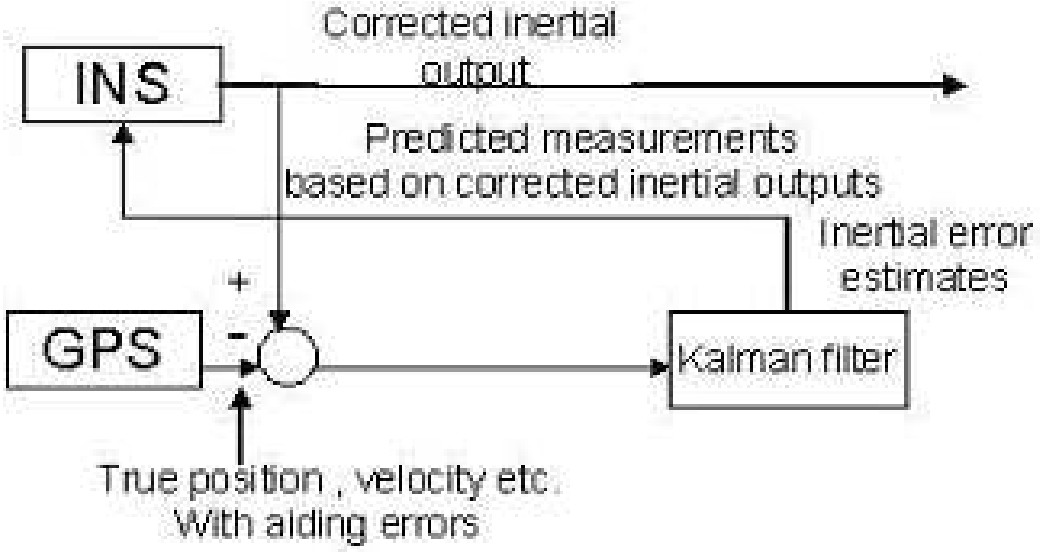


Figure 3.6: Feedback aided INS

Chapter 4

Simulation

The simulation of the integration of INS and GPS using a Kalman filter has been done. Separate programs have been written for modelling the INS, creating the errors in the sensors and modelling the output of the GPS using both MATLAB and C.

4.1 Implementation

A program called Flight Dynamics and Controls (FDC) toolbox, when given the initial conditions of the aircraft thrust and aerodynamics, gave as its output the time history of the aircraft in the form of a state vector $\mathbf{X}$, where

$$\mathbf{X} = \begin{bmatrix} \phi & \theta & \psi & p & q & r & a_x & a_y & a_z & X & Y & Z & V_T & \alpha & \beta \end{bmatrix}^T \tag{4.1}$$

- ϕ, θ, ψ are the Euler angles in radians,
- p, q, r are the roll, pitch and yaw rates from the gyroscopes in radians per second,
- a_x, a_y, a_z are the accelerations from the accelerometers in m/s^2,
- X, Y, Z are the distances along the three axes in the navigation frame in meters,
- V_T, α, β are the velocity of the aircraft in m/s, the angle of attack in radians and the sideslip angle in radians, respectively.

The FDC program can generate these values at any time step as required. Typical time steps or update rates range from 10ms - 100ms.

4.1.1 INS module

The INS program now takes 6 states from this time history, viz. p, q, r, a_x, a_y, a_z. These act as if the program is reading directly from the gyros and accelerometers. Then the program integrates and calculates the four Euler parameters using equations 3.2 - 3.6. From these Euler parameters, the Euler angles are calculated using equations 3.11 - 3.13. Now the accelerations from the accelerometers are used to calculate $\dot{U}$, $\dot{V}$, $\dot{W}$ given by equations 3.14 - 3.16, which are then integrated to get the values of U, V, W.

We now have the velocity components of the aircraft in the body frame. To convert it to the navigation frame or local earth frame, we use the **DCM** matrix, as in equation 3.20, and calculate $\mathbf{V_T}$ using equation 3.21. These velocity components are then integrated to get the position X, Y, Z along the three axes in the local earth frame. The latitude, longitude and height can be calculated using equations 3.22 - 3.24. All the integrations are carried out using fourth order Runge-Kutta methods.

The initial conditions for time $t = 0$ have to be given and they are calculated using the values generated by the FDC at $t = 0$. Equations 3.7 - 3.10 are used to calculate the values of the four Euler parameters at $t = 0$ taking the values of ϕ, θ, ψ from the FDC state vector $\mathbf{X}$. Similarly, the initial values of the velocities are also calculated using the following equations [23],

$$U = V_T \cos\alpha \cos\beta \tag{4.2}$$

$$V = V_T \sin\beta \tag{4.3}$$

$$W = V_T \cos\beta \sin\alpha, \tag{4.4}$$

where the values of V_T, α, β are taken from the FDC state vector $\mathbf{X}$ at time $t = 0$. Finally, the initial values of X, Y, Z are also taken from their corresponding FDC

counterparts.

4.1.2 GPS module

The GPS gives the latitude, longitude and altitude of the current location of the receiver. What our program does is that it converts the X, Y, Z given out by the FDC into latitude, longitude and altitude as would be given out by the GPS receiver. The update rate is 1 second. The GPS program uses WGS-84 approximation in which the earth is considered as an ellipse with a semi-major axis (equatorial radius) of $a = 6,378,137m$, and a semi-minor axis (polar radius) of $b = 6,356,752.3142m$ [29].

It is necessary to define the distance corresponding to a 1° change in longitude (F_{lon}) and latitude (F_{lat}) for a specified location (latitude and height or altitude). The following equations define F_{lon} and F_{lat} for a specified latitude λ and height h.

$$F_{lon} = \frac{\pi}{180^\circ}\left(\frac{a^2}{\sqrt{a^2\cos^2\lambda + b^2\sin^2\lambda}} + h\right)\cos\lambda \tag{4.5}$$

$$F_{lat} = \frac{\pi}{180^\circ}\left(\frac{a^2b^2}{\left(a^2\cos^2\lambda + b^2\sin^2\lambda\right)^{\frac{3}{2}}} + h\right) \tag{4.6}$$

Hence, the latitude and longitude at the current location (λ_2, μ_2) can be calculated from the latitude and longitude from the previous location (λ_1, μ_1) in the following manner :

$$\lambda_2 = \frac{\delta X}{F_{lat}} + \lambda_1, \tag{4.7}$$

$$\mu_2 = \frac{\delta Y}{F_{lon}} + \mu_1. \tag{4.8}$$

where δX and δY are the changes in position along North direction and East direction on the earth, respectively. If we consider the earth as a sphere, F_{lon} and F_{lat} can be replaced by just the radius of the earth and the latitude and longitude can be calculated. However, to make the GPS modeling more authentic, we have considered the earth as an ellipse.

4.1.3 Sensor Modelling

The accelerometer senses the acceleration in terms of g and sends it to the INS in Volts by conversion using a scale factor. A certain offset at zero g called the bias exists by default. The scale factor and the bias details are available from the specification sheets of the accelerometers. Errors arise in the acceleration sensed because the scale factor and the bias are not fixed. They vary stochastically and they lie within a certain range which is specified in the data sheets of the accelerometers. Let C mV/g be the nominal scale factor, c mV/g be the possible deviation from the nominal scale factor, D be the voltage at 0 g offset and d be the deviation from that offset. If a_0 is the acceleration sensed by the accelerometer then, the output from the accelerometer in voltage units is given by

$$Acc_{output} = (C \pm c)a_0 + (D \pm d) \tag{4.9}$$

c and d are normally distributed random numbers with zero mean, and $c/3$ and $d/3$ as the standard deviations, respectively. d changes randomly after each turn-on.

This voltage signal from the accelerometer passes through the ADC. Hence, we have to include the errors for the ADC as well in Acc_{output}. If we use a 16-bit ADC with a range of $0 - 5000$mV, we divide Acc_{output} with $\frac{5000}{65536}$, round off the result and then multiply it by $\frac{5000}{65536}$, thus incorporating the errors due to the ADC into the acceleration measured by the accelerometer. Once this is done, the Acc_{output} is the digitized voltage output which has to be converted back to acceleration in terms of g which is done using the following equation,

$$a = \frac{Acc_{output} - D_c}{C_c} \tag{4.10}$$

where a is the acceleration value sent to the INS program, C_c and D_c are the calibrated values of the scale factor and bias, respectively. The gyroscope error modelling is also done in a similar way accounting for the corresponding scale factors and offset biases. These errors together lead to a drift, which grows with time, in the output (location) given by the INS and it could be up to hundreds of metres. Table 4.1 gives a set

of values given by the specification sheets which were used in the simulation. The errors due to temperature effects and due to the misalignment of accelerometers and gyroscopes have been ignored.

Table 4.1: Sensor specifications used in the simulation

Quantity	**Value**	**Standard Deviation**
Scale factor of the accelerometer	250 mV/g	$\pm \frac{25}{3}$ mV/g
Zero g Offset of the accelerometer	2500 mV	$\pm \frac{625}{3}$ mV
Scale factor of the gyroscope	1.11 mV/°/s	$\pm \frac{10}{3}\%$
Typical turn-on drift of the gyroscope	0.12 °/s	–
Random noise incorporated in the GPS	–	±20 m

4.1.4 Kalman Filter Module

The error dynamics model given in the works of Schmidt [1], Bar-Itzhack et al [2], and Grewal [4] has been used for the simulation. The error dynamics equations are obtained when the nominal equations are perturbed in the local level north-pointing coordinate system that corresponds to the geographic location indicated by the INS. The differential equations that describe the error behavior of the INS are divided into equations describing the propagation of the translatory errors and equations describing the propagation of attitude errors. Translatory errors are the velocity and position errors. The translatory errors and the attitude errors are not coupled to each other. The nine state INS/GPS integration Kalman filter will then be built using the error dynamics equations. The perturbation of the position, velocity, attitude DCM, and gravity can be expressed as

$$\hat{\mathbf{r}}^n = \mathbf{r}^n + \delta\mathbf{r}^n \tag{4.11}$$

$$\hat{\mathbf{v}}^n = \mathbf{v}^n + \delta\mathbf{v}^n \tag{4.12}$$

$$\hat{\mathbf{C}}_b^n = (\mathbf{I} - \mathbf{E}^n)\mathbf{C}_b^n \tag{4.13}$$

$$\boldsymbol{\gamma}^n = \mathbf{g}^n + \delta\mathbf{g}^n \tag{4.14}$$

where $\mathbf{r}^n$, $\mathbf{v}^n$ and $\boldsymbol{\gamma}^n$ denote the position, velocity and gravity vectors in the navigational frame, respectively; $\mathbf{C}_b^n$ denotes the attitude direction cosine matrix from the navigational frame to the body frame and $\mathbf{E}^n$ is the skew symetric form of the attitude errors ($\boldsymbol{\epsilon}^n$)

$$\mathbf{E}^n = (\boldsymbol{\epsilon}^n\times) = \begin{bmatrix} 0 & -\epsilon_D & \epsilon_E \\ \epsilon_D & 0 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{bmatrix} \tag{4.15}$$

and $\hat{}$ and δ denote computed values and errors, respectively.

The linear position error dynamics can be obtained by perturbing equations 3.22 - 3.24, which are the dynamics equations for the geodetic positions. Since the position dynamics equations are functions of position and velocity, the position error dynamics equations are obtained using partial derivatives [30]:

$$\delta\hat{\mathbf{r}}^n = F_{rr}\delta\mathbf{r}^n + F_{rv}\delta\mathbf{v}^n \tag{4.16}$$

where

$$F_{rr} = \begin{pmatrix} \frac{\partial\dot{\lambda}}{\partial\lambda} & \frac{\partial\dot{\lambda}}{\partial\mu} & \frac{\partial\dot{\lambda}}{\partial h} \\ \frac{\partial\dot{\mu}}{\partial\lambda} & \frac{\partial\dot{\mu}}{\partial\mu} & \frac{\partial\dot{\mu}}{\partial h} \\ \frac{\partial\dot{h}}{\partial\lambda} & \frac{\partial\dot{h}}{\partial\mu} & \frac{\partial\dot{h}}{\partial h} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{-V_N}{(R_e+h)^2} \\ \frac{V_E\sin\lambda}{(R_e+h)\cos^2\lambda} & 0 & \frac{-V_E}{(R_e+h)^2\cos\lambda} \\ 0 & 0 & 0 \end{pmatrix},$$

$$F_{rv} = \begin{pmatrix} \frac{\partial\dot{\lambda}}{\partial V_N} & \frac{\partial\dot{\lambda}}{\partial V_E} & \frac{\partial\dot{\lambda}}{\partial V_D} \\ \frac{\partial\dot{\mu}}{\partial V_N} & \frac{\partial\dot{\mu}}{\partial V_E} & \frac{\partial\dot{\mu}}{\partial V_D} \\ \frac{\partial\dot{h}}{\partial V_N} & \frac{\partial\dot{h}}{\partial V_E} & \frac{\partial\dot{h}}{\partial V_D} \end{pmatrix} = \begin{pmatrix} \frac{1}{R_e+h} & 0 & 0 \\ 0 & \frac{1}{(R_e+h)\cos\lambda} & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

and R_e is the radius of the earth and is considered a constant.

The velocity dynamics equation is expressed as

$$\hat{\dot{\mathbf{v}}}^n = \hat{\mathbf{C}}_b^n\mathbf{f}^b - (2\boldsymbol{\Omega} + \boldsymbol{\omega'})\times\hat{\mathbf{v}}^n + \boldsymbol{\gamma}^n \tag{4.17}$$

where $\mathbf{f}^b$ is the acceleration of the aircraft in the body frame, $\boldsymbol{\Omega}$ and $\boldsymbol{\omega'}$ are given in equations 3.17 and 3.18. The gravitation vector in the navigation frame, $\mathbf{g}^n$, can be

appromiated by the normal gravity vector $\begin{pmatrix} 0 & 0 & \gamma \end{pmatrix}^T$, and γ varies with altitude. Assuming a spherical earth model, we can write

$$\gamma = \gamma_0 \left(\frac{R_e}{R_e + h} \right)^2, \tag{4.18}$$

where γ_0 is the normal gravity at $h = 0$. On perturbing equation 4.17 and using equations 3.17, 3.18 and 4.18, we can obtain the velocity error dynamics equation [1, 2, 30] as follows :

$$\delta \dot{\mathbf{v}}^n = F_{vr} \delta \mathbf{r}^n + F_{vv} \delta \mathbf{v}^n + (\mathbf{f}^n \times) \epsilon^n + \mathbf{C}_b^n \delta \mathbf{f}^b \tag{4.19}$$

where

$$F_{vr} = \left(\begin{array}{c|c|c} -2V_E\Omega\cos\lambda - \frac{V_E^2}{(R_e+h)\cos^2\lambda} & 0 & \frac{-V_N V_D}{(R_e+h)^2} + \frac{V_E^2 \tan\lambda}{(R_e+h)^2} \\ \hline 2\Omega(V_N\cos\lambda - V_D\sin\lambda) + \frac{V_E V_N}{(R_e+h)\cos^2\lambda} & 0 & \frac{V_E V_D}{(R_e+h)^2} - \frac{V_N V_E \tan\lambda}{(R_e+h)^2} \\ \hline 2V_E\Omega\sin\lambda & 0 & \frac{V_E^2+V_N^2}{(R_e+h)^2} - \frac{2\gamma}{(R_e+h)} \end{array} \right),$$

$$F_{vv} = \left(\begin{array}{c|c|c} \frac{V_D}{R_e+h} & -2\Omega\sin\lambda - 2\frac{V_E\tan\lambda}{R_e+h} & \frac{V_N}{R_e+h} \\ \hline 2\Omega\sin\lambda + \frac{V_E\tan\lambda}{R_e+h} & \frac{V_D + V_N\tan\lambda}{R_e+h} & 2\Omega\cos\lambda + \frac{V_E}{R_e+h} \\ \hline -2\frac{V_N}{R_e+h} & -2\Omega\cos\lambda - 2\frac{V_E}{R_e+h} & 0 \end{array} \right)$$

and $\delta \mathbf{f}^b$ is the perturbation in the acceleration vector in the body frame.

The attitude error dynamics equation [30] can be written as

$$\dot{\boldsymbol{\epsilon}}^n = F_{er} \delta \mathbf{r}^n + F_{ev} \delta \mathbf{v}^n - ((\boldsymbol{\Omega} + \boldsymbol{\omega'}) \times) \boldsymbol{\epsilon}^n - \mathbf{C}_b^n \delta \boldsymbol{\omega}_{ib}^b \tag{4.20}$$

where

$$F_{er} = \begin{pmatrix} -\Omega\sin\lambda & 0 & \frac{-V_E}{(R_e+h)^2} \\ 0 & 0 & \frac{V_N}{(R_e+h)^2} \\ -\Omega\cos\lambda - \frac{V_E}{(R_e+h)\cos^2\lambda} & 0 & \frac{V_E\tan\lambda}{(R_e+h)^2} \end{pmatrix},$$

$$F_{ev} = \begin{pmatrix} 0 & \frac{1}{R_e+h} & 0 \\ \frac{-1}{R_e+h} & 0 & 0 \\ 0 & \frac{-\tan\lambda}{R_e+h} & 0 \end{pmatrix}$$

and $\delta\boldsymbol{\omega}_{ib}^{b}$ is the perturbation in the angular rate vector between the inertial frame and the body frame.

A state space model (equation 3.29) can be constructed by augmenting the equations 4.16, 4.19 and 4.20 as follows :

$$\dot{\mathbf{x}} = F\mathbf{x} + G\mathbf{u} \tag{4.21}$$

where F is the dynamics matrix, $\mathbf{x}$ is the state vector, G is a design matrix, $\mathbf{u}$ is the forcing vector function [1, 2, 30]:

$$F = \begin{pmatrix} F_{rr} & F_{rv} & 0 \\ F_{vr} & F_{vv} & (\mathbf{f}^{b}\times) \\ F_{er} & F_{ev} & -((\boldsymbol{\Omega} + \boldsymbol{\omega'})\times) \end{pmatrix} \quad \mathbf{x} = \begin{bmatrix} \delta\mathbf{r}^{n} \\ \delta\mathbf{v}^{n} \\ \boldsymbol{\epsilon}^{n} \end{bmatrix}$$

$$G = \begin{pmatrix} 0 & 0 \\ \mathbf{C}_{b}^{n} & 0 \\ 0 & -\mathbf{C}_{b}^{n} \end{pmatrix} \quad \mathbf{u} = \begin{bmatrix} \delta\mathbf{f}^{b} \\ \delta\boldsymbol{\omega}_{ib}^{b} \end{bmatrix}$$

The elements of $\mathbf{u}$ are white noise whose covariance matrix is given by

$$E[\mathbf{u}(t)\mathbf{u}(t)^{T}] = Q(t)\delta(t-\tau) \tag{4.22}$$

where the operator δ denotes the Dirac delta function whose unit is 1/time [30]. Q is called the spectral density matrix and has the form

$$Q = diag\begin{pmatrix} \sigma_{ax}^{2} & \sigma_{ay}^{2} & \sigma_{az}^{2} & \sigma_{\omega x}^{2} & \sigma_{\omega x}^{2} & \sigma_{\omega z}^{2} \end{pmatrix} \tag{4.23}$$

where σ_a and σ_ω are the standard deviations of the accelerometers and gyroscopes, respectively.

We now transform equation 4.21 to its discrete time form :

$$\mathbf{x}_{k+1} = \Phi_k\mathbf{x}_k + \mathbf{w}_k \tag{4.24}$$

where Φ_k is the state transition matrix, and $\mathbf{w}_k$ is the driven response at t_{k+1} due to the presence of input white noise during time interval (t_k, t_{k+1}) [28]. For the

implementation of the INS, because the time interval $\Delta t = t_{k+1} - t_k$ is very small, we can numerically approximate the state transition matrix as

$$\Phi_k = \exp(F\Delta t) \approx I + F\Delta t \tag{4.25}$$

The covariance matrix associated with $\mathbf{w}_k$ is

$$Q_k = E[\mathbf{w}_k \mathbf{w}_k^T] \approx \Phi_k G Q G^T \Phi_k^T \Delta t \tag{4.26}$$

If the norm of Q_k is larger than the real one, the Kalman filter trusts the measurements more than the system, thus making the estimates noisy due to free passage of measurement noise [30]. However, there is no time lag. If the norm of Q_k is less than one, the time lag exists. When the norm of Q_k is much smaller than the real one, the filter diverges, which may result in numerical instabilities. Hence, for low cost inertial systems, Q_k must be selected pessimistically so that the trajectory follows that of the GPS. The elements corresponding to δf_z should be large enough so that they can account for the uncertainties in gravity as well as sensor imperfection.

The observation equation 3.26 expresses the vector measurement, $\mathbf{z}_k$, at time t_k as a linear combination of the state vector $\mathbf{x}_k$ and a random measurement error, $\mathbf{v}_k$. The process noise, $\mathbf{w}_k$ and the measurement noise, $\mathbf{v}_k$ are uncorrelated, hence their covariance is 0. The covariance matrix for $\mathbf{v}_k$ is given by

$$E[\mathbf{v}_k \mathbf{v}_k^T] = R_k \tag{4.27}$$

The Kalman filter is then implemented using equations 3.32 - 3.37. The position from GPS is considered as measurements. The formulation of the measurement equation can be written as

$$\mathbf{z}_k = \mathbf{r}_{INS}^n - \mathbf{r}_{GPS}^n = \begin{pmatrix} \lambda_{INS} - \lambda_{GPS} \\ \mu_{INS} - \mu_{GPS} \\ h_{INS} - h_{GPS} \end{pmatrix} \quad H_k = \left(\begin{array}{c|c|c} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{array} \right) \tag{4.28}$$

Since λ and μ are in radians and hence very small, they cause numerical unstabilities in calculating the Kalman gain $\mathbf{K}_k$. Hence, the first two rows are multiplied by

$(R_e + h)$ and $(R_e + h)\cos\lambda$, respectively [30]. The measurement equation now takes the form :

$$\mathbf{z}_k = \begin{pmatrix} (R_e + h)(\lambda_{INS} - \lambda_{GPS}) \\ (R_e + h)\cos\lambda(\mu_{INS} - \mu_{GPS}) \\ h_{INS} - h_{GPS} \end{pmatrix}$$

$$H_k = \left(\begin{array}{ccc|c|c} (R_e + h) & 0 & 0 & & \\ 0 & (R_e + h)\cos\lambda & 0 & 0_{3\times3} & 0_{3\times3} \\ 0 & 0 & 1 & & \end{array}\right) \tag{4.29}$$

and the following measurement noise matrix has been used

$$R_k = diag\begin{pmatrix} \sigma_\lambda^2 & \sigma_\mu^2 & \sigma_h^2 \end{pmatrix} \tag{4.30}$$

which can be obtained from GPS processing. In our simulation, we have taken the error sphere of the GPS to have a radius of 20m. Hence $\sigma_\lambda = \sigma_\mu = \sigma_h = 20$m.

The initial estimation uncertainty standard deviations must be given to start a Kalman filter. If an inertial measurement unit is initialised in stationary mode, the position uncertainty will be that of the GPS solution, the velocity uncertainty zero and the attitude uncertainty will depend wholly on the accelerometer and gyroscope biases [30]. If the biases can be estimated, the attitude uncertainty can be reduced.

The estimated errors are fed back to the mechanization (INS module) (see figure 3.6) or fed forward to the output (figure 3.5). In the feedforward method, the inertial system operates as if there was no aiding : it is unaware of the existence of the filter or the external data. The disadvantage of this method is that the mechanization can experience unbounded error growth, which makes unbounded error observations delivered to the Kalman filter. This causes problem to the linear filter since only small errors are allowed due to the linearization process. Therefore, the feedback method is optimal for low cost INSs. The estimated state vector $\mathbf{x}_k$ is used to correct for the position, velocity and attitude calculated by the INS, using equations 4.11 - 4.13 every time a measurement, $\mathbf{z}_k$, is taken, i.e. every 1 second.

$$\mathbf{r}^n = \hat{\mathbf{r}}^n - \delta\mathbf{r}^n \tag{4.31}$$

$$\mathbf{v}^n = \hat{\mathbf{v}}^n - \delta\mathbf{v}^n \,. \tag{4.32}$$

The following characteristic holds for the first order attitude errors :

$$(\mathbf{I} - \mathbf{E}^n)^{-1} = (\mathbf{I} + \mathbf{E}^n) \tag{4.33}$$

Hence, equation 4.13 can be manipulated to yield the DCM attitude feedback as follows :

$$\mathbf{C}_b^n = (\mathbf{I} + \mathbf{E}^n)\hat{\mathbf{C}}_b^n \tag{4.34}$$

After feedback is done, the error state vector should be set to zero, because the state vector is zero until the next measurements are made for a feedback nine-state INS/GPS integration Kalman filter. If the feedback is made everytime measurements take place, the state prediction does not need to be implemented [30].

4.2 Results

In this section we discuss the results obtained from the simulation of individual subsystems, i.e. the INS and GPS and the integrated system.

4.2.1 Individual subsystems

Due to mechanical errors existing in the accelerometers and gyroscopes, the INS, individually, does not accurately give the position of the aircraft. As seen in figures 4.1 - 4.3, the unaided INS (blue line) deviates from the actual trajectory (black line) by a very large extent. This simulation has been done by modelling the sensors as explained in section 3.1.3. The updates from the gyroscopes and accelerometers are taken every 10ms. The above mentioned figures show us the typical output given

by the GPS (red circles), with an update taken every second. A standard deviation of 20m has been assumed in modelling the GPS output. The GPS has a long term accuracy and the INS has a short term accuracy, hence the individual systems by themselves are not enough to give us a good and accurate measure of the location. If Selective Availability is introduced, the GPS output would have a standard deviation of around 40-50m. Hence, we go for an integrated system of the INS and GPS using a Kalman filter.

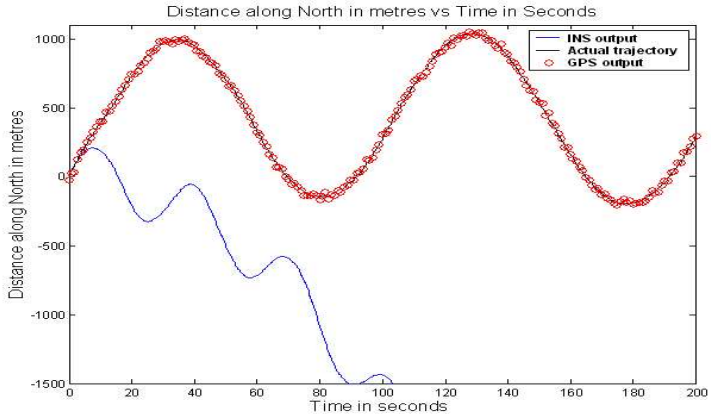


Figure 4.1: Distance along North calculated by the unaided INS and GPS

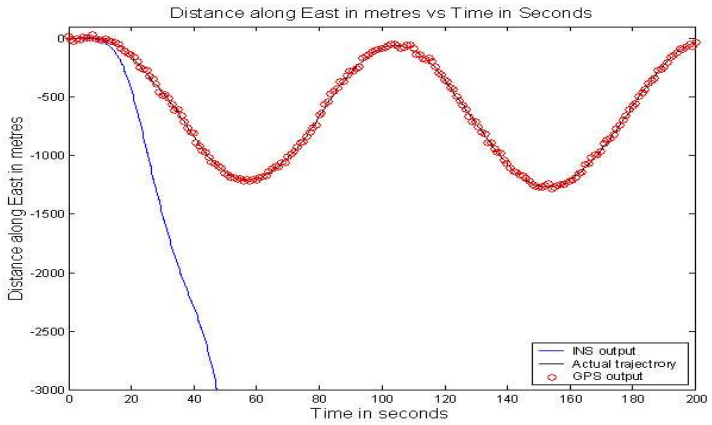


Figure 4.2: Distance along East calculated by the unaided INS and GPS

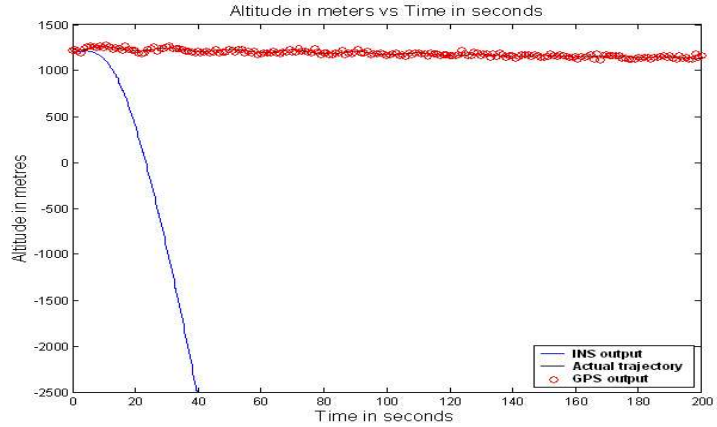


Figure 4.3: Altitude calculated by the unaided INS and GPS

4.2.2 Integrated system

A nine-state model Kalman filter was implemented as described in section 3.1.4. Figures 4.4 - 4.2.2, show the output of the simulation as well as the GPS output simulated for a period of 200s. The standard deviation chosen for the accelerometers here was 10mGal. The standard deviations of the accelerometers was varied and we have got two sets of outputs. As given in the works by Ronnback [15] and Shin [30], the standard deviations of the accelerometers were increased to give an output with a much better accuracy as seen in figures 4.7 - 4.9. The standard deviations of the accelerometers chosen was 30mGal.

The update from the accelerometers and gyroscopes was taken every 0.01s, the GPS update was taken every 1s and the Kalman filter was run every 0.5s [17] to achieve better accuracy. Every alternate 0.5s instant, when the GPS update is not available, equation 3.32 is used to predict the error state $\hat{\mathbf{x}}_k$, using the most recent GPS update as the measurement, i.e. the GPS update is taken constant for that whole one second. This also comes in use when there are GPS outages. Whenever the GPS update is taken, $\hat{\mathbf{x}}_k^-$ is made zero, and whenever the GPS update is not taken (every alternate 0.5s or when there are GPS outages), $\hat{\mathbf{x}}_k^-$ is left as it is and updated using the equations 3.32 - 3.36.

The graphs for attitude computed and corrected by the Kalman filter are given in figures 4.10 - 4.12. We cannot expect the Kalman filter to correct the attitude given by the INS perfectly as attitude is not a part of the measurement vector. We can only correct the attitude given by the INS using the attitude errors predicted by the state matrix. This corrected attitude forms a part of the integration loop in the whole system. The graphs 4.10 - 4.12 correspond to the graphs 4.7 - 4.9.

Figures 4.13 - 4.15 show us the results from running the program when we assume a GPS outage of 8s, from the period $t = 25$s to $t = 32$s. For this time period of GPS outage, the GPS values used by the program remain the same as the last measured values, i.e. the values measured at $t = 24$s. At $t = 33$s, the GPS starts reading again,

and new values are read by the program. During this time the Kalman filter relies totally on the INS and state predictions, and the accuracy is affected as we can see from the graphs. But once the new GPS values are read by the program, the Kalman filter takes very less time, of the order of a few seconds, to settle down towards the actual trajectory.

If Selective Availability (SA) is introduced or exists in the GPS, the standard deviation of the GPS position measured is around 40-50m. Figures 4.16 - 4.18 show the output of the program, when SA was introduced with a standard deviation of 40m. The Kalman filter does not know that SA has been introduced, and it still uses the standard deviation of 20m, as given by the specification sheets of the GPS receiver, in the measurement noise covariance matrix $\mathbf{R}_k$. SA has been introduced in the GPS values for the period of 200s. We can see that due to introduction of SA, the accuracy of the output has decreased but it still better than the GPS but not as good as what it would be without SA. If while running the program for a longer time, suddenly we introduce SA, even then the program will not waver from its regular output, although the accuracy will decrease for the time period in which SA has been introduced.

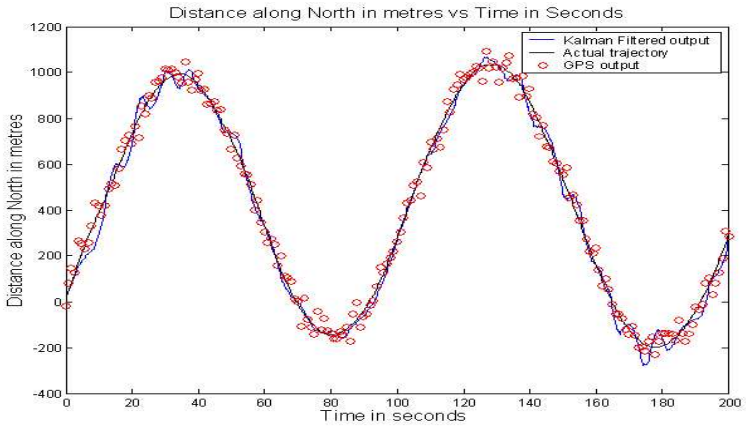


Figure 4.4: Kalman filtered output of distance along North

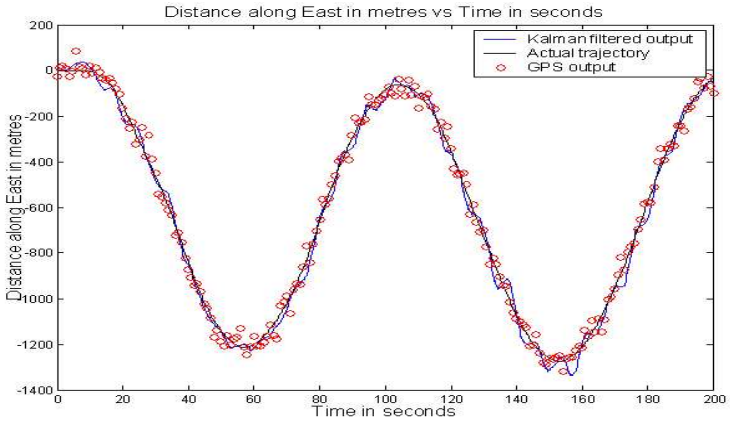


Figure 4.5: Kalman filtered output of distance along East

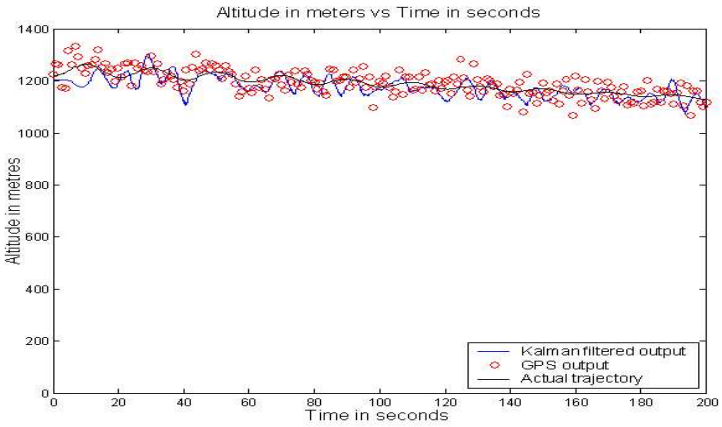


Figure 4.6: Kalman filtered output of Altitude

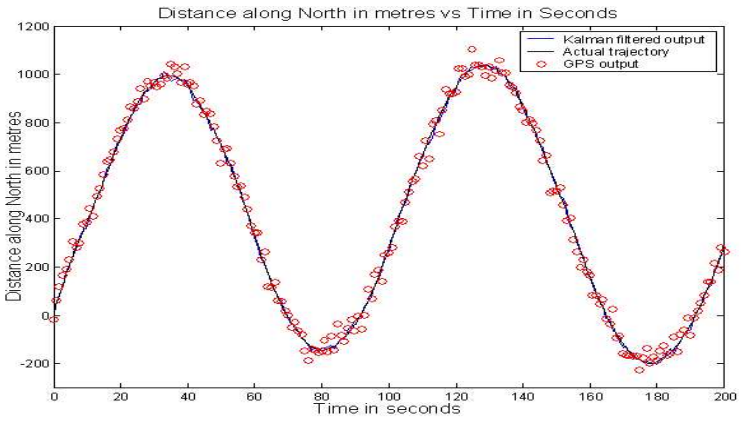


Figure 4.7: Distance along North calculated with higher variance of accelerometers

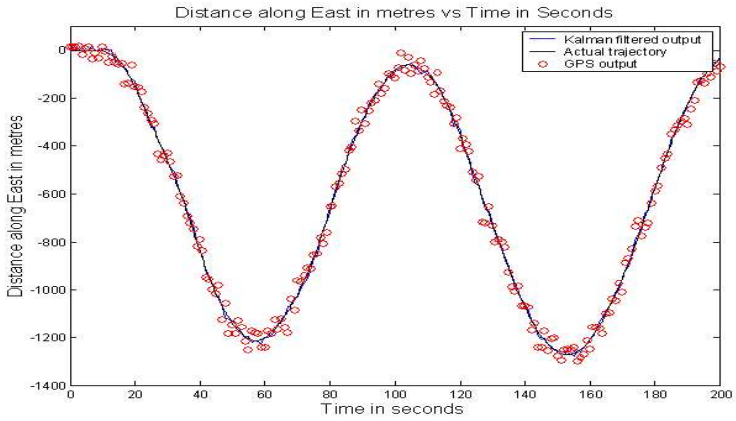


Figure 4.8: Distance along East calculated with higher variance of accelerometers

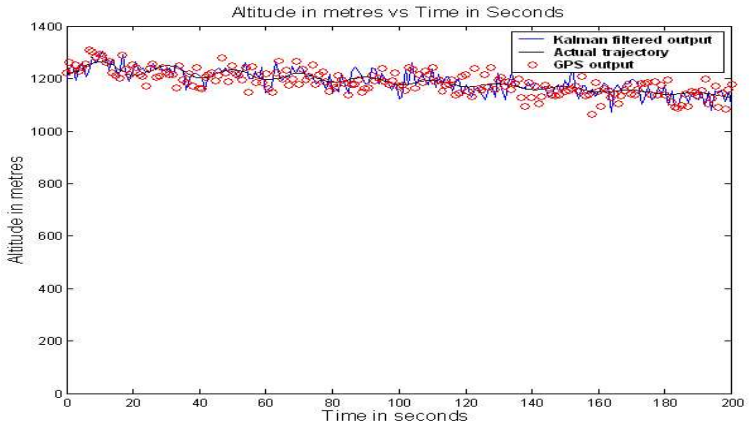


Figure 4.9: Altitude calculated with higher variance of accelerometers

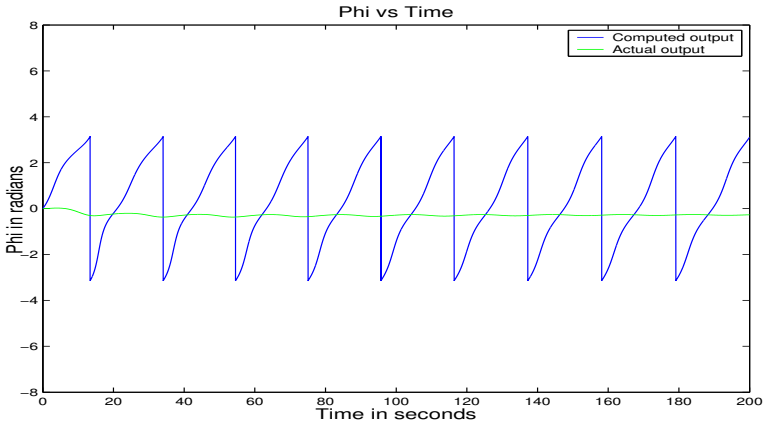


Figure 4.10: Euler angle Phi vs Time

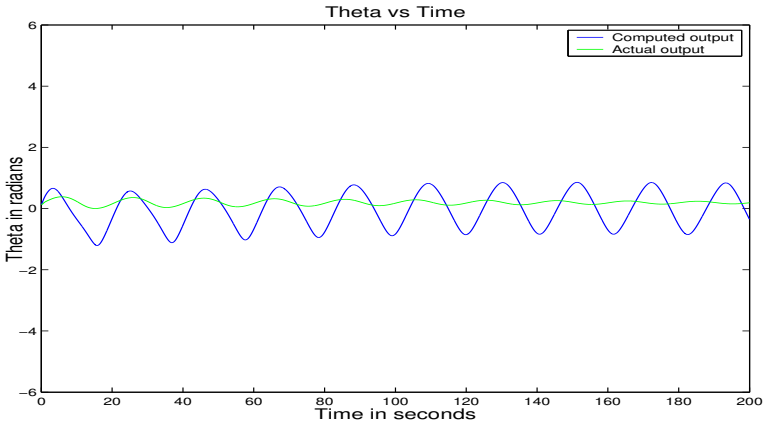


Figure 4.11: Euler angle Theta vs Time

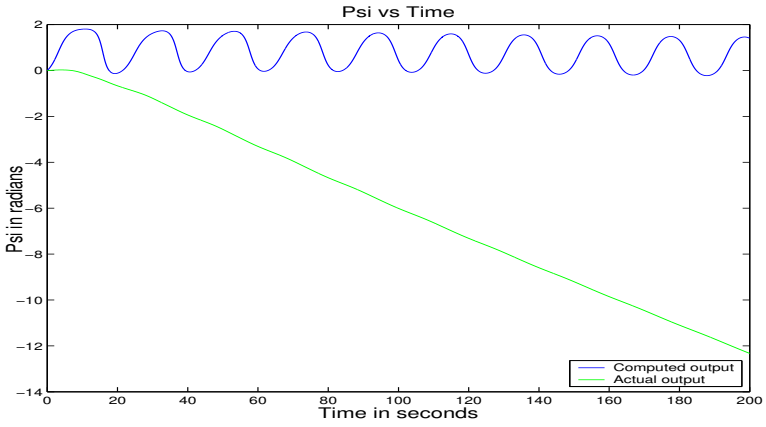


Figure 4.12: Euler angle Psi vs Time

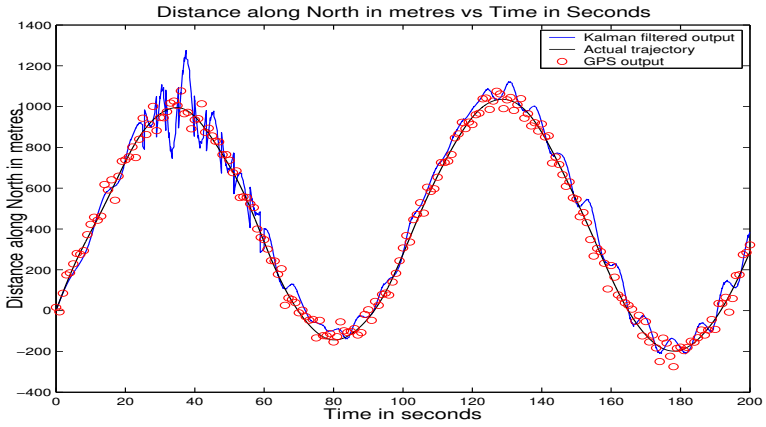


Figure 4.13: Distance along North calculated with GPS outage between 25s and 33s

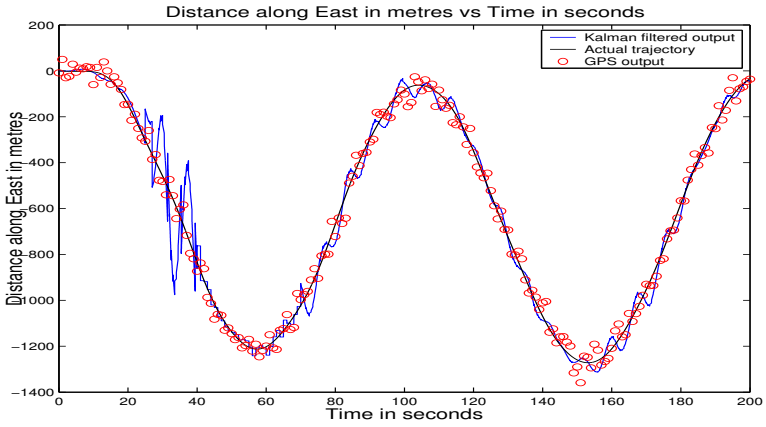


Figure 4.14: Distance along East calculated with GPS outage between 25s and 33s

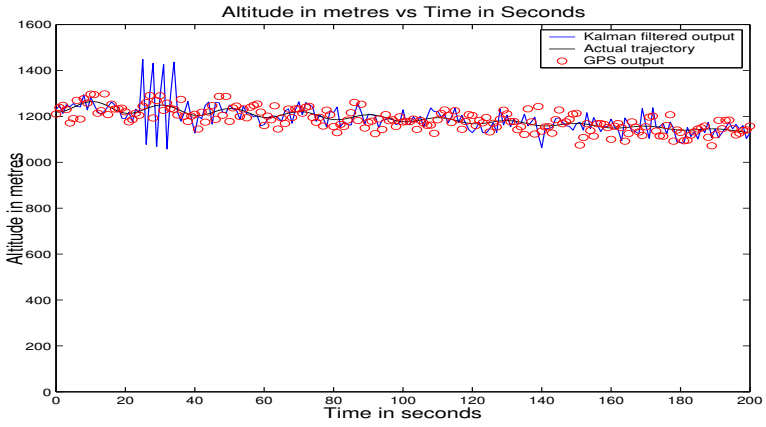


Figure 4.15: Altitude calculated with GPS outage between 25s and 33s

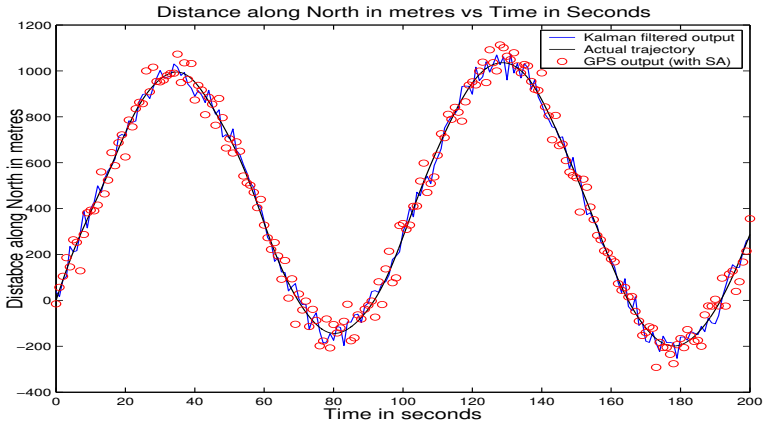


Figure 4.16: Distance along North calculated with Selective Availability introduced

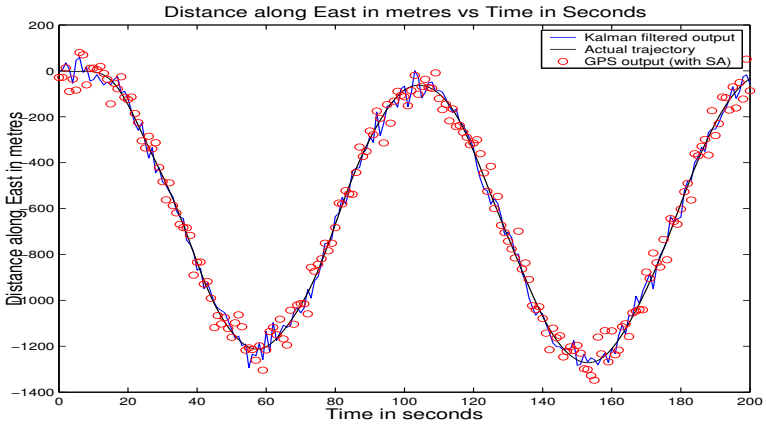


Figure 4.17: Distance along East calculated with Selective Availability introduced

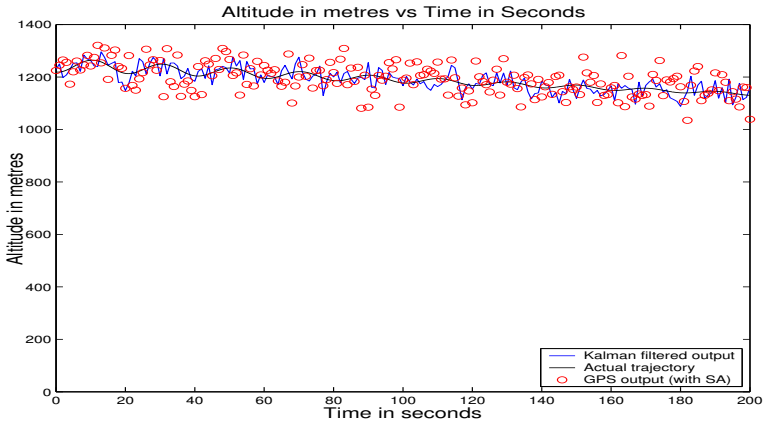


Figure 4.18: Altitude calculated with Selective Availability introduced

Chapter 5

Hardware Implementation

The present study is undertaken concurrently with a Masters project to design and develop a hardware for implementing the INS/GPS integrated system [35]. The objective of this project included generating requirements for the other project and to test the INS/GPS integrated system program on the hardware. The hardware has been specifically designed [35] for a mini aerial vehicle (MAV). For the computations of the INS and the Kalman Filter a digital signal processor (DSP) has been used in the hardware.

5.1 Hardware Description[35]

The system to be used is compact,light and single supply operated. The schematic of the whole system is shown in figure 5.1.

The system can be divided into two blocks :

- GPS and INS Data Acquisition (GIDAC) card
- Navigation Processor Card (NPC)

The analog signals from the accelerometers and gyroscopes are signal conditioned

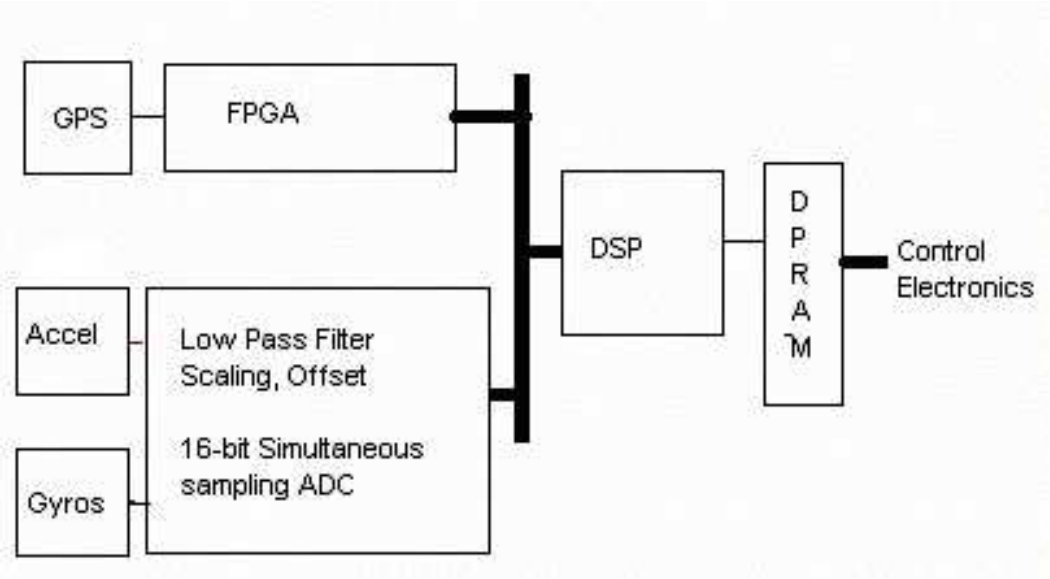


Figure 5.1: Schematic of the hardware system

(filtered for noise, and scaled to the range of 0 – 5V) to the data acquisition input range using a second order Butterworth low pass filter. These signals are sampled simultaneously using a 16-bit parallel output, analog - to - digital converter *ADS8364* made by Texas Instruments Inc.

General 16-bit ADCs have multiplexers at the input, which do individual sampling causing delays if all inputs have to be taken at the same instant, hence the *ADS8364* has been chosen as it is a six channel, simultaneous sampling, 16-bit parallel ADC. The device incorporates an internal buffer that can be powered from the same 3.3V supply as the DSP. All the six sensor voltages can be read simultaneously using this ADC and they can be processed by the NPC to give us the required position. Simultaneous sampling of input signals are performed to eliminate any phase lag which might exist otherwise. All digitized signals from the ADC are interfaced to the NPC.

To relieve main processors from computational processing overhead during slow speed serial I/O, a Field Programmable Gate Array (FPGA) based dedicated serial port interface is used. The total number of chips is reduced to one and it is much faster than the existing microcontroller schemes.

In the system architecture, the FPGA chip is programmed to receive the GPS data from GPS receiver, and generates a busy signal when accessing the internal Dual Port RAM (DPRAM) of FPGA. This low-going busy signal interrupts the DSP processor, and the processor fetches the data from internal DPRAM of the FPGA chip. The DPRAM stores the position updates given by the GPS, in the proprietary (SIRF sentences for the GPS receiver we are using) sentences format, every second. Asynchronous communication is maintained between the GPS module and the NPC card using DPRAM, thus saving the processor time during the transfer of data.

The INS computations and its integration with the GPS is carried out on the Navigation Processor Card (NPC). The NPC comprises of a *TMS320VC33* DSP manufactured by Texas Instruments Inc. and its supporting hardware. This DSP board gives us the freedom to download the software directly from the computer using a printer port interface and communicate with other computer programs as well. The *TMS320VC33* is a floating point DSP with an instruction cycle time of 13ns or 150MHz and provides upto 75 MIPS, 150M FLOPS. The DSP has a standard 50 pin connector interface with external circuitry. It has 34K words (1 word = 32 bits) dual access SRAM, bootloader and onchip peripherals. It is inexpensive and easily available. An inexpensive tool for the processor is available. The processor has 32 bit data bus and 24 bit address bus. The control signals for selecting peripheral chips on the DSP board are generated using programmable array logic (PAL). The DSP board has a DPRAM *7130* for parallel data transfer through standard 20-pin connector. A standard 6-pin interface for serial I/O exists on the board and can be configured as a general purpose I/O pins. It also has a standard-14 pin emulator connector.

5.2 System Flow

The INS/GPS integration program has several subprograms and header files, along with assembly coding for initialisation of DSP and its peripherals.

An interrupt vector table is created with /INT0 and /INT1 defined as interrupts from the GPS and INS acquisitions. Two timers, Timer0 and Timer1, are then configured using the *Timer Global Control* registers and *Timer Period* registers in the initialization routine. The interrupt /INT0 is received from the FPGA chip, when the GPS update every second takes place, and is given the highest priority. The six channels of the ADC (A_0, A_1, B_0, B_1, C_0 and C_1) are paired up two at a time. Hence, the end of conversion (EOC) signal comes in 3 pulses from the ADC. Therefore, a variable count is assigned a value 2 during initialization for counting the EOC signal from the ADC. Variables to begin the INS computations, Start_INS, and to check if the GPS readings are available, GPS_available, are set to low or 0. Timer0 generates a clock of 5MHz for the ADC sampling and Timer1 generates interrupts at a frequency of 100Hz. These timers are enabled by configuring the *Timer Control* registers.

On initialization, a software reset is given to the ADC chip by the program, which is then configured to operate in *CYCLE MODE*. In this mode, the six channels of the ADC are read in a fixed order every time an INS acquisition takes place. Timer0 is run, thus providing a clock to the ADC continuously. Timer1 generated interrupts at a rate of 100Hz for each INS time step. On every Timer1 overflow flag, the /HOLDx signals (the 'x' in /HOLDx stands for each of the six channels of the ADC, for eg. /HOLDA_0) are made low and at this instant the ADC samples all the six channels simultaneously. During this time the main program is waiting in an infinite loop or IDLE mode. The respective Interrupt Service Routines (ISR) are enabled based on the interrupts received.

- When the EOC of ADC (/INT1) occurs, the data of all six channels is stored. Start_INS is set to high or 1. The program then returns from the /INT1 ISR.
- When the reading from GPS by the FPGA (/INT0) occurs, the data is read from the internal DPRAM within the FPGA, and GPS_available is set to high or 1. The program then returns from the /INT0 ISR.

When the Start_INS and GPS_available variables are set high, the INS and

Kalman filter computations are performed in the main program loop, respectively. Figures 5.2 - 5.4 [35] show flowcharts describing the flow of instructions in the initialization, ISR and computation part of the program.

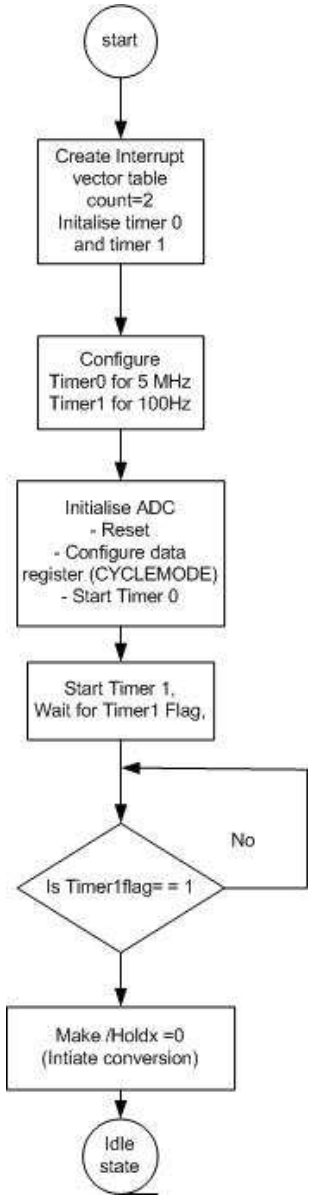


Figure 5.2: Flow of instructions : initialization and reading data

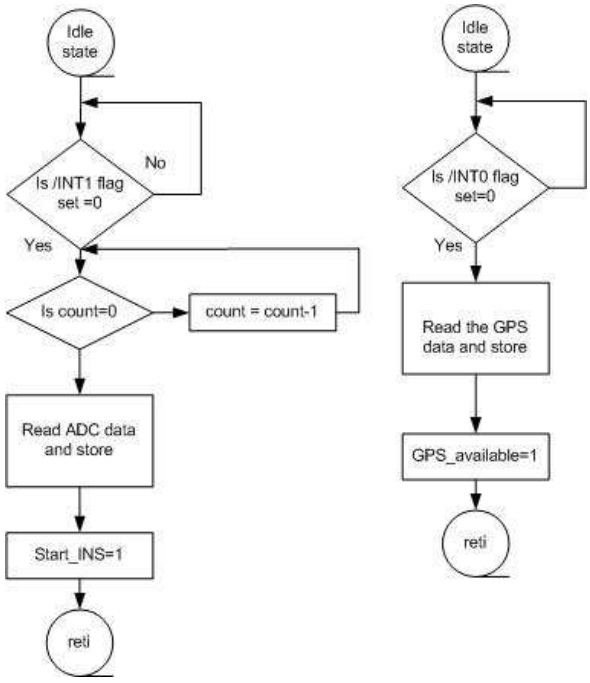


Figure 5.3: Interrupt Service Routines (ISR)

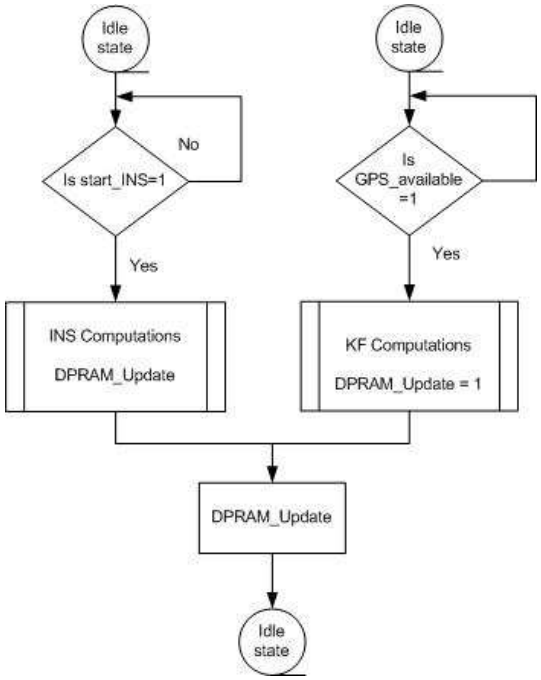


Figure 5.4: Flow of instructions : computation and output

5.3 DSP Simulator

The C program for INS/GPS integration using Kalman filtering, has been tried and tested on the simulator for the *TMS320VC33* DSP series in a software simulator made by Texas Instruments Inc. known as *Code Composer Studio c3x4x*. This software simulates the actual DSP on the computer. With this simulator we can debug the programs without the target hardware. Time critical code, as well as individual portions of the program can be tested. The simulator uses the standard C or assembly source debugger interface, allowing the user to debug the programs in C or in assembly language or both.

Earlier, the integration code was written for a standard C compiler which would take its inputs from files every 10ms for the INS and every second for the GPS. These inputs were the accelerations and angular rates in standard decimal format, on which sensor modelling was done. For the integration code to run on Code Composer Studio, the voltages obtained after modelling the sensors, were converted to hexadecimal format. These were then manipulated upon, as explained in section 3.1.3 on sensor modelling, so as to account for the ADC modelling. The modified values (digitized signals from the ADC) were converted back to accelerations and angular rates, using the calibration values of the biases and scale factors, as required by the INS/GPS integration programs. The values of accelerations and angular rates were generated using the FDC program instead of using actual sensors on flying vehicle. For the GPS data, the positions along the North, East and Down axes (in metres) were converted to hexadecimal format, so as to match the data as would be given by a GPS in SIRF sentences. These hexadecimal values were converted to latitude and longitude using the formulae given in section 3.1.2. The data for the initial state of the system at time $t = 0$, was hardcoded into the program.

The Code Composer Studio handles file inputs and outputs using a tool called *probe points*, wherein each variable, to be read from or written to a file, has to be assigned a probe point and its own input or output file, whichever applicable. These

probe points are nothing but pointers which act as sensor readers. Here they read from a file, on the actual hardware they will be directly getting voltages from the sensors or reading stored values from the DPRAM.

Code Composer Studio converts the whole C program into highly efficient instructions in assembly language (a file in COFF format) which are then input to a *TMS320c3x* assembler or linker. Cycle counting displays the number of clock cycles in a single-step operation or in the run mode. To count the number of instruction cycles the program takes, there is a tool called *profile point*. If we assign a profile point to the beginning and end of the program and run it for a fixed number of readings (i.e. a fixed number of times the inputs from the sensors are given), the profile points help us clock the number of cycles taken by the program. Each instruction in assembly language has a certain number of cycles assigned to it. Once the total number of cycles is known, we can calculate the time taken by the program to run each INS computation or each Kalman filtering computation.

The number of instruction cycles were calculated using profile points for both the INS program and KF individually. The INS program was run for 50s or 5001 steps (each reading taken in steps of 0.01s) and the total number of instruction cycles were calculated to be 224744224, which when divided by 5001 gave 44940 cycles per INS computation. Since each cycle corresponds to 26ns, one INS computation on an average took 1.17ms to run completely. One INS computation excludes the time taken to read the inputs from the sensors and comprises of only the integration of the inputs to give us the position. The maximum number of cycles taken by an INS computation in the 5001 steps was 51475 and the minimum was 44375 which correspond to a time of 1.34ms and 1.15ms, respectively. Considering the worst case possible, we can safely say that an INS computation takes approximately 1.5ms. The program was run for one Kalman filter computation and the number of instruction cycles used were 2443314 which corresponded to a time of 6.33ms. One Kalman filter computation includes the time taken to read the GPS values,the Kalman filtering and the INS computation for that instant to finally give us the updated position. To these

Table 5.1: Instruction cycles for the program

Computation	**No. of Cycles**	**Time taken (26ns/cycle)**
INS	44940	1.17ms
Kalman Filter	2443314	6.33ms
ADC reading	–	1ms
DPRAM writing	–	1ms

computation timings, we also need to add the time taken to read the sensor inputs which is 1ms and to write the outputs to the DPRAM which is another 1ms.

If the DSP is run at its maximum speed of 150MHz, the cycle time per instruction is 13ns. We plan to run the DSP at 75MHz which is why the cycle time per instruction is 26ns. The memory consumed by the program was only 17K words. The full memory of the DSP was not used. Hence no external RAM is required unlike if we use the DSP *TMS320VC31* which has only onboard memory of 1K word, and an external memory chip needs to be attached to the board.

The INS program was run on Code Composer Studio for 50 seconds and the output was matched with the output given by the INS program written using MATLAB. The output from the Code Composer Studio matched with the MATLAB output upto 3-4 decimal places. The output from the MATLAB code has a better match with the actual trajectory because MATLAB is a highly accurate professional software, and its Runge Kutta function is much more accurate than the fourth order Runge Kutta used in the C program.

The Kalman filter program was also run for 50 seconds with the GPS update being given every second. The output was similar to the output of the Kalman program written in C.

Although the Code Composer Studio is a very slow working software, it is the number of cycles taken by the program which gives us a clear indication of the speed

at which the program would run on the DSP.

5.4 Hardware Issues and Future Work

Once we have debugged and run the program on the Code Composer Studio simulator, we need to run it on the target hardware using an emulator. An emulator is a powerful, high speed software or kernel used for system-level integration and debugging on the DSP. Each DSP series has its own set of emulators. Emulators are user-friendly and support hardware development on the target processor. Access is provided to every memory location and register of the target processor through a JTAG cable connector. Emulators can be DOS based or Windows based. The program in COFF format is yet to be run on the emulator.

The speed of the INS and Kalman filter computations have to be checked again on the target hardware. Although the Code Composer Studio gives us an idealistic count, it is always safer to test on the target hardware. The results of the program need to be checked after running on the hardware for a long time so as to find out what level of accuracy is achieved on the target hardware.

As of now the sensor board is not designed, and hence we are giving inputs directly from the computer. Once the sensors are finalised, the sensor modelling in the program has to be adjusted according to the sensors being used. Calibration needs to be done to calculate the biases and scale factors of the accelerometers and gyroscopes. Hewitson et al [36] and Shin [30] give us methods to calibrate the sensors effectively.

The initial state data at time $t = 0$ is currently hardcoded into the program. We have to design a way to input the initial state directly from the hardware.

The output position is written onto the DPRAM. This needs to be fed into a computer or any other display interface.

Chapter 6

Conclusions

The INS system was modelled as given by the specification sheets. A nine state Kalman filter was designed and implemented using the perturbation theory model for postion, velocity and attitude. The accuracy of the results obtained was better than the accuracy given by the GPS and INS as individual systems. The accuracy can be further improved if we increase the states of the filter and model for the scale factors,biases and nonorthogonality of the sensors. The INS updates were taken every 10ms, GPS updates every 1s and the Kalman filter was executed every 0.5s. Results were generated for possible GPS outages of 8s during which the Kalman filter relies on the INS and the state prediction vector. Although the accuracy decreases during this period, the system settles down once the GPS updates are available. Results were also generated for possible introduction of Selective Availability in the GPS data. Although the accuracy of the output was not as good as that acheived by regular GPS data (without SA), but nevertheless it was better than the GPS values.

The program of integration of the INS and GPS using Kalman filtering was run on the DSP simulator software and the computation time was well within the requirements of 10ms.

References

[1] Schmidt, G.T., "Strapdown Inertial Systems - Theory and Applications," *AGARD Lecture Series*, No. 95, 1978.

[2] Bar-Itzhack, I.Y., and Berman, N., "Control Theoretic Approach to Inertial Navigation Systems," *Journal of Guidance*, Vol. 11, No. 3, 1988, pp. 237-245.

[3] Grewal, M.S., and Andrews, A.P., *Kalman Filtering: Theory and Practice using MATLAB*, John Wiley and Sons, New York, 2001.

[4] Grewal, M.S., Weill, L.R., and Andrews, A.P., *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley and Sons, New York, 2001.

[5] Wolf, R., Eissfeller, B., Hein, G.W., " A Kalman Filter for the Integration of a Low Cost INS and an attitude GPS," *Institute of Geodesy and Navigation*, Munich, Germany.

[6] Grejner-Brzezinska, D.A., and Wang, J., "Gravity Modelling for High-Accuracy GPS/INS Integration," *Navigation*, Vol. 45, No. 3, 1998, pp. 209-220.

[7] Srikumar, P., Deori, C.D., " Simulation of Mission and navigation Functions of the UAV - Nishant," *Aeronautical Development Establishment*, Bangalore.

[8] Randle, S.J., Horton, M.A., " Low Cost Navigation Using Micro - Machined Technology," *IEEE Intelligent Transportation Systems Conference*, 1997.

[9] Gaylor, D., Lightsey, E.G., " GPS/INS Kalman Filter desing for Spacecraft operating in the proximity of te International Space Station," *University of Texas - Austin*, Austin.

[10] Brown, A., Sullivan, D., " Precision Kinematic Alignment Using a low cost GPS/INS System," *Proceedings of ION GPS 2002*, Navsys Corporation, Oregon, 2002.

[11] Moon, S.W., Kim, J.H., Hwang, D.H., Ra, S.W., Lee, S.J., " Implementation of a Loosely Coupled GPS/INS integrated system, " *Chungnam National University*, Korea.

[12] Salychev, O.S., Voronov, V.V., Cannon, M.E., nayak, R., Lachapelle, G., " Low Cost INS/GPS Integration : Concepts and Testing," *Institute of Navigation National Technical Meeting*, California, 2000.

[13] Wang, J., Lee, H.K., Rizos, C., " GPS/INS Integration : A Performance Sensitivity Analysis," *University of New South Wales*, Sydney.

[14] Kwon, J.H., "Airborne Vector Gravimetry Using GPS/INS," *Ohio State University*, Ohio, April 2000.

[15] Ronnback, S., "Development of a INS/GPS Navigation Loop for an UAV," *University of Sydney*, Sydney, February 2000.

[16] Gautier, J.D., " GPS/INS Generalized Evaluation Tool (GIGET) for the design and testing of integrated Navigation Systems," *Stanford University*, California, 2003.

[17] Mayhew, D.M., "Multi-rate Sensor Fusion for GPS Navigation Using Kalman Filtering," *Virginia Polytechnic Institute and State University*, Virginia, May 1999.

[18] Moore, J.B., Qi, H., "Direct Kalman Filtering Approach for GPS/INS Integration", *IEEE Transactions on Aerospace and Electronic Systems* , Vol 38, No.2, April 2002.

[19] Shang, J., Mao, G., and Gu, Q., "Design and Implementation of MIMU/GPS Integrated Navigation Systems," *Tsinghua University*, China, 2002.

[20] Cao, F.X., Yang, D.K., Xu, A.G., " Low Cost SINS/GPS Integration for Land Vehicle Navigation, " *IEEE International Conference on Intelligent Transport Systems* , September 2002, Singapore.

[21] Panzieri, S., Pascucci, F., Ulivi, G., "An Outdoor navigation system using GPS and Inertial Platform," *IEEE ASME Transactions on Mechatronics*, Vol. 7, No. 2, June 2002.

[22] Dorobantu, R., Zebhauser, b., "Field Evaluation of a Low-Cost Strapdown IMU by means of a GPS," *Technical University of Munchen*, Germany.

[23] Collinson, R.P.G., *Introduction to Avionics*, Chapman and Hall, London, 1996.

[24] Pamadi, B.N., *Performance, Stability, Dynamics and Control of Airplanes*, AIAA Education Series,Virginia, 1998.

[25] Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley and Sons, Toronto, 1972.

[26] Omerbashich, M., "Integrated INS/GPS Navigation from a Popular Perspective," *Journal of Air Transportation*, Vol. 7, No. 1, 2002, pp. 103-119.

[27] Parkinson, B.W., and Spilker, J.J.Jr., *Global Positioning System: Theory and Applications*, Volume 1, AIAA, Washington DC, 1996.

[28] Brown, R.G., and Hwang, P.Y.C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley and Sons, New York, 1992.

[29] Agricultural and Biological Engineering, Purdue University, "*http://abe.www.ecn.purdue.edu/∼abegps/web_ssm/web_GPS_eq.html*".

[30] Shin, E.H., "Accuracy Improvement of Low Cost INS/GPS for Land Application," *University of Calgary*, December 2001.

[31] Bona, B.E., and Smay, R.J., "Optimum reset of Ship's Inertial Navigation System," *IEEE Trans. on Aerospace and Electronic Systems,* AES-2 : No. 4, 1966, pp. 409-414.

[32] Farrell, J.A., and Barth, M., *The Global Positioning System & Inertial Navigation*, McGraw-Hill, 1998.

[33] *ADXRS300*, Analog Devices Inc., *http://www.analog.com/Analog_Root/productPage/productHome/0.2121.ADXRS300.00.html*.

[34] *ADXL105EM-3* , Analog Devices Inc., *http://www.analog.com/UploadedFiles/Evaluation_Boards/Tools/222772852ADXL105_190EM_c.pdf*.

[35] Bhaktavatsala, S., "Design and Development of DSP Based GPS-INS Integrated System," M.Tech. Dissertation, *Indian Institute of Technology*, Bombay, June 2004.

[36] Hewitson, S.A., Wang, J. and Kearsley, A.H.W., "Performance Evaluation of Inertial Navigation Systems for Surveying," *The 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services*, Australia, 2003.

Appendix A

The equations that have to be integrated for the INS to give the position of the aircraft in term of latitude, longitude and height are reprinted below.

$$\dot{e_0} = -\frac{1}{2}(e_1 p + e_2 q + e_3 r) \tag{A.1}$$

$$\dot{e_1} = \frac{1}{2}(e_0 p + e_2 r - e_3 q) \tag{A.2}$$

$$\dot{e_2} = \frac{1}{2}(e_0 q + e_3 p - e_1 r) \tag{A.3}$$

$$\dot{e_3} = \frac{1}{2}(e_0 r + e_1 q - e_2 p) \tag{A.4}$$

with the four Euler parameters satisfying the following equation at all points of time.

$$e_0{}^2 + e_1{}^2 + e_2{}^2 + e_3{}^2 = 1 \tag{A.5}$$

Euler angles can hence be calculated as shown below :

$$\theta = \sin^{-1}[-2(e_1 e_3 - e_0 e_2)] \tag{A.6}$$

$$\phi = \cos^{-1}\left[\frac{e_0{}^2 - e_1{}^2 - e_2{}^2 + e_3{}^2}{\sqrt{1 - 4(e_1 e_3 - e_0 e_2)^2}}\right] \text{sign}[2(e_2 e_3 + e_0 e_1)] \tag{A.7}$$

$$\psi = \cos^{-1}\left[\frac{e_0{}^2 + e_1{}^2 - e_2{}^2 - e_3{}^2}{\sqrt{1 - 4(e_1 e_3 - e_0 e_2)^2}}\right] \text{sign}[2(e_1 e_2 + e_0 e_3)] \tag{A.8}$$

The angular rate of rotation of the earth in the North-East-Down frame of reference is given below :

$$\mathbf{\Omega} = \begin{bmatrix} \Omega \cos\lambda \\ 0 \\ -\Omega \sin\lambda \end{bmatrix} \tag{A.9}$$

$$\boldsymbol{\omega'} = \begin{bmatrix} \dot{\mu}\cos\lambda \\ -\dot{\lambda} \\ -\dot{\mu}\sin\lambda \end{bmatrix} \tag{A.10}$$

The measured angular rates (denoted with subscript m) from the gyroscopes then help calculate the actual angular rates in the body frame using the following equation.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_m - \mathbf{DCM} \begin{bmatrix} \boldsymbol{\Omega} + \boldsymbol{\omega'} \end{bmatrix} \tag{A.11}$$

$$\mathbf{DCM} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\phi\cos\theta \\ \sin\theta\cos\phi\cos\psi + \sin\psi\sin\phi & \sin\phi\sin\theta\cos\phi - \cos\psi\sin\theta & \cos\phi\cos\theta \end{bmatrix} \tag{A.12}$$

From the measured accelerations (a_X, a_Y, a_Z) and a given gravity model for g depending on position coordinates we calculate the velocities.

$$\dot{U} = a_X + Vr - Wq + g\sin\theta \tag{A.13}$$

$$\dot{V} = a_Y - Ur + Wp - g\cos\theta\sin\phi \tag{A.14}$$

$$\dot{W} = a_Z + Uq - Vp - g\cos\theta\cos\phi \tag{A.15}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} = \mathbf{DCM}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{A.16}$$

$$\dot{X} = V_N \tag{A.17}$$

$$\dot{Y} = V_E \tag{A.18}$$

$$\dot{Z} = V_D \tag{A.19}$$

We can then calculate the latitude (λ), longitude (μ) and height (H) using the following equations.

$$\dot{\lambda} = \frac{V_N}{R_e} \tag{A.20}$$

$$\dot{\mu} = \frac{V_E}{R_e\cos\lambda} \tag{A.21}$$

$$\dot{H} = -V_D \tag{A.22}$$