LOW COST INERTIAL NAVIGATION: LEARNING TO INTEGRATE NOISE AND FIND YOUR WAY

By

KEVIN J. WALCHKO

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2002

ACKNOWLEDGMENTS

The research conducted in this thesis could not have been accomplished without the help of the Machine Intelligence Lab and the Autonomous Submarine Team at the University of Florida who donated the use of their IMU. Their support was of great help.

Special thanks go to Dr. David Novick and Dr. Paul Mason. Both of these individuals always provided me with a sounding board to bounce my ideas and concerns off of. They also provided me with an indispensable wealth of information, knowledge, understanding, and friendship. For that alone I will always be grateful.

Last, but not least, I would like to thank my loving wife, Nina Walchko, and our pack of wild cats. They suffered with me through the hard times and long hours, but were always there to help cheer me up when I needed it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure page

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

LOW COST INERTIAL NAVIGATION:
LEARNING TO INTEGRATE NOISE
AND FIND YOUR WAY

By

Kevin J Walchko

August 2002

Chair: Dr. Mike Nechyba
Major Department: Electrical and Computer Engineering

Navigation is becoming more common in all areas of industry and commercial sectors. The main tool being utilized is GPS. However there are situations in which higher levels of accuracy are required which can not be achieved by GPS alone. This thesis will discuss the design and implementation of an inertial navigation system (INS) using an inertial measurement unit (IMU) and GPS. The INS is capable of providing continuous estimates of a vehicle’s position and orientation. Typically IMU’s are very expensive sensors; however this INS will use a “low cost” version costing around $5,000. Unfortunately with low cost also comes low performance and is the main reason for the inclusion of GPS into the system. Thus the IMU will use accelerometers and gyros to interpolate between the 1Hz GPS positions. All important equations regarding navigation are presented along with discussion. Results are presented to show the merit of the work and highlight various aspects of the INS.

CHAPTER 1
INTRODUCTION

Navigation has been present for thousands of years in some form or another. The birds, the bees, and almost everything else in nature must be able to navigate from one point in space to another. For people, navigation had originally included using the sun and stars. Over the years we have been able to develop better and more accurate sensors to compensate for our limited range of senses. This thesis will discuss work using one of these advanced sensors, an inertial measurement unit (IMU). This sensor, coupled with the proper mathematical background, is capable of detecting accelerations and angular velocities and then transforming those into the current position and orientation of the system.

Inertial Navigation

Unfortunately Inertial Navigation Systems (INS) have been relegated to the realm of military applications due to the extreme cost of the systems. High precision components must be incorporated into the IMU to produce good results. The most critical are the gyros, which are used to determine the orientations of the system. The orientation is critical to properly account for gravity. If gravity is accounted for in the accelerometer readings, then the system thinks it is moving when in fact it is not. Also high quality accelerometers are important. Ideally their should be no bias (or very low at least) and behave in a simple linear fashion. Unfortunately in real life, noise, disturbances, drifts, misalignments, and manufacturing complications enter into the equation and make developing an INS difficult.

If INS is so expensive, how do you create a low cost solution? The saving grace of low cost INS is the Global Positioning System (GPS). This system provides accuracies of 10 m 95% of time at an update rate of 1 Hz. The receiver only costs $100-$200 depending on size, features, etc. This system now provides a way to bound the errors in an INS and the use of lower cost components in the IMU. Thus cheap but accurate INS systems can be developed for all sorts of applications that previously were cost prohibitive (i.e. consumer cars, remotely controlled vehicles, autonomous vehicles, military munitions, etc).

Previous Work

INS's have been developed for a wide range of vehicles. Sukkarieh [1] developed a GPS/INS system for straddle carriers that load and unload cargo ships in harbors. When the carriers would move from ship to ship, they would periodically pass under obstructions that would obscure the GPS signal. Also, as the carriers got closer to the quay cranes, it became more difficult to get accurate positions due to the GPS signal being reflected about the crane's metal structure. This increases the time of flight of the GPS signal and results in jumps in the position. During these times the INS would then take over, and guide the slow moving carrier until a reliable GPS signal could be acquired.

Mandapat [2] developed a low cost INS for Dr. Carl Crane here at University of Florida to replace their current system which is highly accurate, but very expensive. The current system is the Honeywell Modular Azimuth Positioning System (MAPS) and Ashtech Z-12 Differential GPS. This system has an accuracy of less than 10 cm at a data rate of 10 Hz. The new system uses an IMU also developed from Honeywell which has ring laser gyros and costs about three times the IMU used in this work. Mandapat's GPS/INS produced positional errors within 1 to 2 meters of the MAPS system at a fraction of the cost.

Bennamoun et al. [3] developed a GPS/INS/SONAR system for an autonomous submarine. The SONAR added another measurement to help with accuracy, and provided a positional reference when the GPS antenna got submerged and could not receive a signal.

Ohlmeyer et al. [4] developed a GPS/INS system for a new smart munitions, the EX-171. Due to the high speed of the missile, update rates of 1 second from a GPS only solution were too slow, and could not provide the accuracy needed. The GPS/INS smart munitions are cheaper than their more accurate cousins, Laser Guided Munitions (LGM). They are also immune to conditions which can greatly deteriorate the accuracy of LGM, such as bad weather, heavy dust/snow storms, and fog. This integration of GPS/INS is a growing trend for military munitions (i.e. bombs, missiles, artillery shells, remotely operated vehicles). Boeing [5] has developed a GPS/INS kit that converts old gravity bombs into precision-guided smart bombs. A control unit is attached to the end of the warhead which contains the GPS/INS system and battery powered motors to control the flight of the bomb. Actual use by American aircraft in Afghanistan during the 2002 War on Terrorism proved these bombs can strike within 13 meters of their intended target.

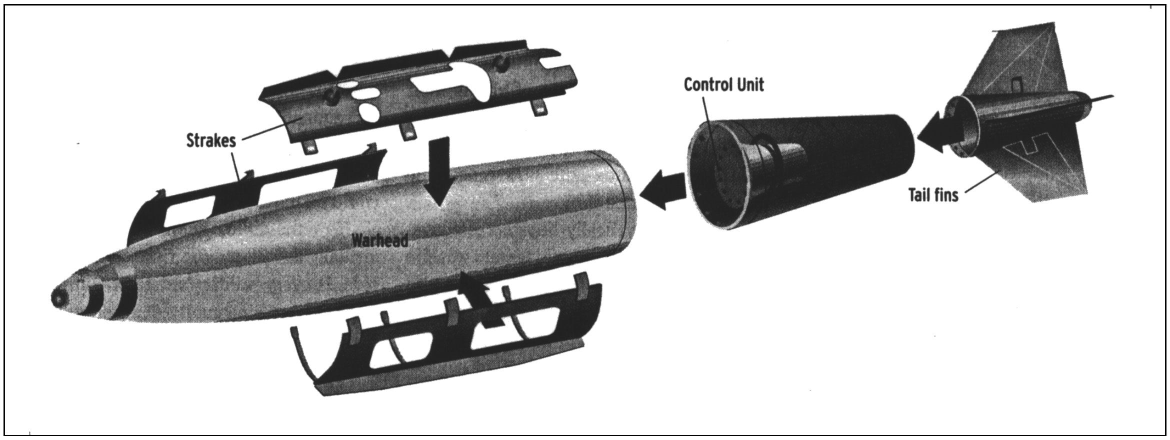

Figure 1-1. Boeing conversion kit which transforms a standard gravity bomb into a smart bomb.

Thesis Outline

The remainder of this thesis will flow as follows. Chapter 2 will discuss GPS and familiarize the reader with how it works. Chapter 3 will introduce all of the key concepts for inertial navigation and derive all important equations. Chapter 4 will introduce problems inherent in inertial navigation and provide the necessary background for the extended kalman filter used which will aid in correcting these errors. Chapter 5 will provide background on hardware used in inertial navigation and the IMU and GPS that were used in this work. Chapter 6 will cover the experimental setup and present results using the developed INS. Various aspects of the INS's performance will be presented to show its merit. Finally chapter 7 will present the conclusions drawn from this work.

CHAPTER 2
WHERE IN THE WORLD IS WALDO

This chapter will give an overview of the global positioning system (GPS). The purpose is to provide the reader with the basic fundamental understanding of how GPS works and some of the technologies involved. Most of the information in this chapter comes from three sources: Dr. Peter H. Dana [6], Garmin Users Manual [7], and Federal Aviation Administration (FAA) [8].

GPS Network Overview

The GPS network of satellites contains a minimum of 21 satellites that orbit the Earth at an altitude of about 11,000 nautical miles. They orbit once every 11 hours and 58 minutes, so that they drift in the sky 4 minutes a day (they drift 2 minutes each orbit). There are 6 different orbits that the satellites can be in, with multiple satellites in each orbit. Each possible orbit is inclined 55 degrees with the equator with no orbits going directly over the poles.

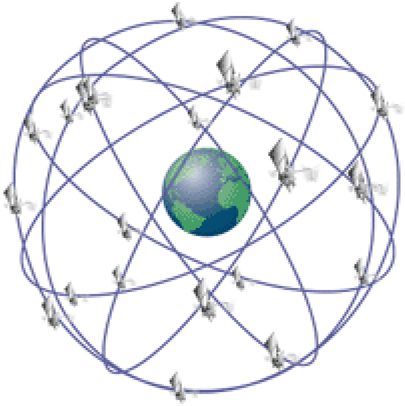

Figure 2-1. Orbits of GPS network.

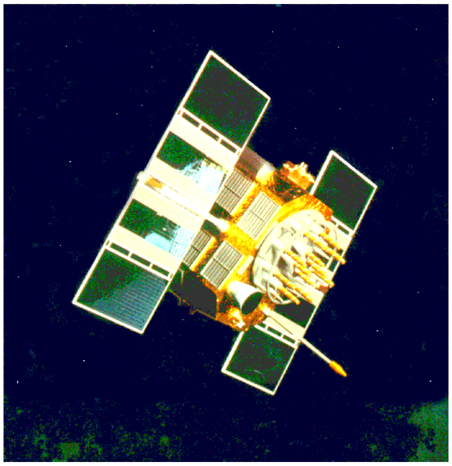

Figure 2-2. Standard GPS satellite in orbit.

How GPS Works

Finding your location using GPS is accomplished by triangulating your position from the known positions of GPS satellites. Distance measured from your position to the satellite is measured by time of flight of a signal sent from the GPS satellite. The signal sent from the satellite tells its location and the time the signal was sent. Since the signal travels at the speed of light, the distance from the receiver to the satellite can be calculated. With a GPS receiver and one satellite, you can determine your possible location on a sphere with the satellite at the center. Unfortunately this is not very useful. With two satellites we get two spheres that are centered at different locations. Now our position lies at the intersection of these two spheres, which is a circle. Unfortunately this is still not useful. Now expanding the our network of satellites to three and a third sphere we get two possible locations the receiver can be in 3D space. Immediately one of the two possible solutions can be discarded, leaving us with our position. Thus with three satellites, we can easily determine our position in 3D space. But this now raises the question, how accurately can we calculate this position? This question will be answered in the next section which covers accuracy.

NMEA Messages

The signals or messages that are sent from the satellites are defined by the National Marine Electronics Association (NMEA). This group has defined standards for just about every possible device used for navigation and instrumentation. In fact the standards defined for use with GPS actually define not one, but many different messages designed to provide every possible piece of useful information. NMEA also allows hardware vendors to define their own proprietary messages. An example sentence might look like:

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*42

Listing 2-1. Meaning of example NMEA message fields

$	Start character of message
GGA	Global Positioning System Fix Data
123519	Fix taken at 12:35:19 UTC
4807.038,N	Latitude 48 deg 07.038’ N
01131.000,E	Longitude 11 deg 31.000’ E
1	Fix quality: 0 = invalid
	1 = GPS fix
	2 = DGPS fix
08	Number of satellites being tracked
0.9	Horizontal dilution of position
545.4,M	Altitude, Meters, above mean sea level
46.9,M	Height of geoid (mean sea level) above WGS84 ellipsoid
(empty field)	time in seconds since last DGPS update
(empty field)	DGPS station ID number
*42	The checksum data, always begins with *

The current NMEA standard is 0183 version 2.0 which transmits data at 4800 baud.

WGS-84

There are hundreds of datum that have been defined all over the world and through out the years. Some of these datums are regional (such as the US datums NAD27 CONUS used by the USGS and NAD83 or the European 1979) and some are global (such as the WGS72). The World Geodetic System (WGS-84) is a world wide datum with its origin located at the center of the Earth and defines a reference ellipse. The reference ellipse is itself defined by the major and minor axes of the Earth, which were determined by examining all known data in 1984. The data came from every possible source including satellies. This is the datum which GPS uses to determine altitude above the reference ellipse.

Grids

Now that we had defined a datum for out coordinate system, we need to be able to divide the world up into nice pieces. Most people are familiar with latitude and longitude which accomplish this division of the world. However the problem with this system is that

Figure 2-1. An airplane receiving a 3D position from 4 GPS satellites.

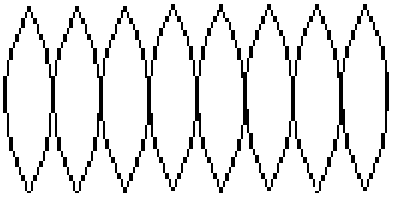

Figure 2-2. How the UTM breaks up the surface of the world.

the world is round and not an nice even sphere. Navigation has defined one nautical mile as 1 minute of latitude along the equator. Since there are 60 minutes to each degree, there are 60 nautical miles per degree of latitude. The problem occurs as we increase our longitude, the distance between lines of latitude decrease. Thus at 45 degrees longitude there are only 42.426 nautical miles. GPS uses a more precise system called the Universal Tansverse Mercator (UTM). This divides the world up into 60 slices with minimal distortion compared to the traditional latitude/longitude system (an example of this is shown in Figure 2-2).

Accuracy

Time is a critical value in the equations that need to be solved to determine the receiver's position in 3D space (to well under a micro-second). Each satellite in the GPS network that orbits the Earth is equipped with an atomic clock so that they know exactly when they send a signal (which is encoded in the signal itself using a pseudo-random signal of 1023 bits). Unfortunately these clocks are very large and expensive. Thus it would not be feasible to embed these into every receiver on the market. So a compromise was made to include more inexpensive clocks into the receivers, but instead of treating time as a known in the GPS equations, it becomes a variable. Thus now a fourth satellite is required to reach a solution for the receiver's 3D position. The assumption is also made that any error in the system comes from our clock being in error (which is an incorrect assumption). Thus the time variable can be changed to reduce the error in the solutions from the four satellites (Figure 2-1).

Once the data is collected from the satellites, a set of 7 simultaneous equations and unknowns are solved. The unknowns are positions (x, y, z), doppler (dx, dy, dz), and time. Proprietary methods are employed when more that 4 satellites are seen by the GPS to solve the now overdetermined solution.

The equation from Pythagoras for solving a receiver's position is

$$Prs + T + Es = \sqrt{(X - Xs)^2 + (Y - Ys)^2 + (Z - Zs)^2} \quad [2\text{-}1]$$

where X, Y, and Z are the position of the receiver that we are trying to find, T is the time error at the receiver, Prs is the estimated range from the receiver to the satellite, and Xs, Ys, and Zs are the are the positions for each of the satellites. The Es term is the sum of all errors in the system, such as: clock errors, troposphere errors, ionosphere errors, etc. From this equation, only the receiver's X, Y, Z position and T are unknown. But with four satellites we get can get 4 equations. However in reality (as already mentioned above) the equations used have 7 unknowns which include the 4 mentioned here plus the doppler data dx, dy, and dz.

Generally (since SA[1] was turned off) most companies will specify the accuracy of their GPS equipment as follows:

Table 2-1: Accuracy of GPS 95% of the time.

Method	Accuracy
GPS	10 m
DGPS (Coast Guard corrections)	1-5 m
DGPS (WAAS)	< 3

Differential GPS

There are several kinds of DGPS available. The most common is the kind that receives corrections from a radio beacon. The beacon listens to transmissions which the U.S. Coast Guard sends containing corrections which provide better accuracy than stand alone GPS. This section will concentrate on a new type called Wide Area Augmentation System (WAAS) which is a product of the FAA to increase the accuracy of GPS for aviation purposes.

WAAS [9] utilizes ground stations which detect and send GPS error information to a Master Control site. The Master Control site uses this information to compute in order of importance or effect:

1. Integrity information
2. Ionospheric and Tropospheric delays
3. Short-term and long term satellite clock errors
4. Short-term satellite position error (Ephemeris)
5. Long term satellite position error (Almanac)

This information is relayed to two WAAS geosynchronous Inmarsat satellites (AOR-W and POR) from the Master Control Stations and is re-broadcast to user receivers as a

[1] Selective Availability (SA) is a method used by the government to degrade the GPS signal globally so foreign powers and terrorists do not have precision positioning capabilities to attack the U.S or its allies. However on May 1, 2000, President Clinton ended the use of SA in an effort to increase its acceptance and use as a positioning system worldwide.

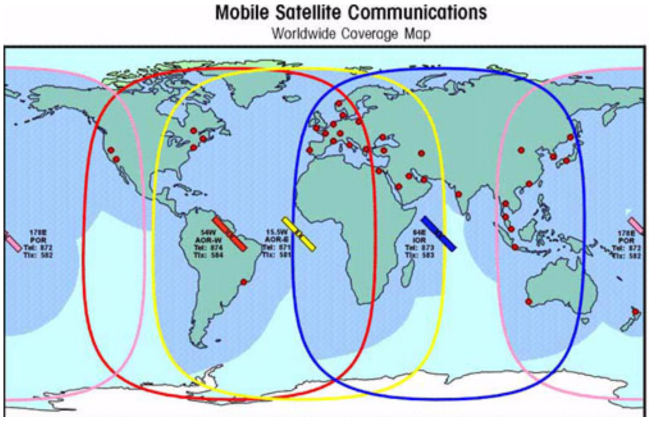

Figure 2-1. WAAS coverage areas. Currently only the Pacific (pink) and Atlantic (yellow) satellites are up and running.

grid of corrections. From this grid, a GPS receiver interpolates the proper Ionospheric correction based on its position in the grid. The “extrapolation” of this information outside the WAAS coverage is less and less precise -to the point of INDUCING errors. Other errors are not location dependant.

The WAAS correction information is different than RTCM corrections (transmitted by the Coast Guard for uses in DGPS) because WAAS decomposes the errors into their primary elements (Iono, clock, & ephemeris). RTCM, on the other hand, broadcasts pseudo range corrections which are the sum of all error sources as observed by the RTCM reference station. This information is only valid relatively close to the reference station. This is why spatial decorrelation is such a large factor for RTCM, but not for WAAS (thus the reason it is “wide area” augmentation).

CHAPTER 3
INERTIAL NAVIGATION

This chapter will introduce strap-down inertial navigation. Derivation for the navigational equations will be presented and discussed.

Overview of Inertial Navigation Systems

A basic flow chart of how inertial navigation works is shown in Figure 3-1. However, this is not all that needs to be done to have an INS that works. There are many problems with noise and unbounded error that must be handled to get any meaningful result out of the INS.

Gimballed INS

The first type of INS developed was a gimballed system. The accelerometers are mounted on a motorized gimballed platform which is always kept aligned with the navigation frame. Pickups are located on the outer and inner gimbals which keep track of the attitude of the stabilized platform relative to the vehicle on which the INS is mounted. This setup has several detractors which make it undesirable.

- Bearings are not frictionless.
- Motors are not perfect (i.e. dead zones, etc.).
- Consumes power to keep the platform aligned with the navigational frame which is not always good on an embedded system.
- Cost is high due to the need for high quality motors, slip rings, bearings and other mechanical parts. Thus the typical customers for such systems were military uses on planes, ships, and intercontinental ballistic missiles.

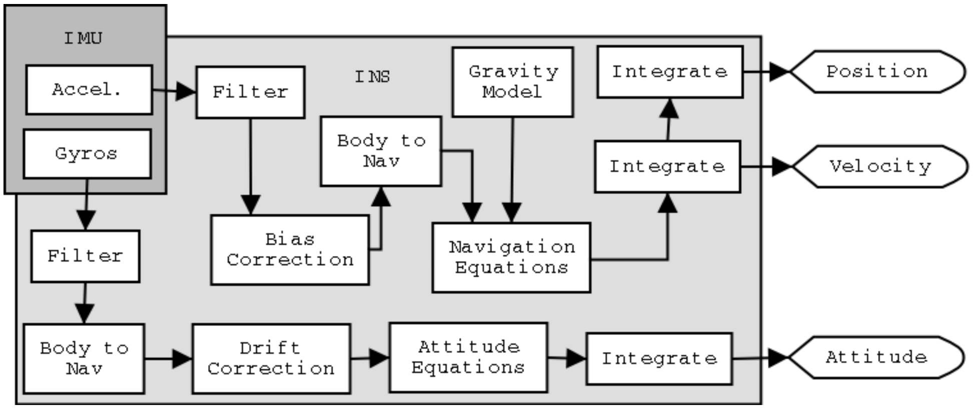

Figure 3-1. A flow chart of a strap-down INS which takes accelerations and rotation rates from the IMU and produces position, velocity, and attitude of the system.

- Recalibration is difficult, and requires regular maintenance by certified personnel which could be difficult on an autonomous vehicle. Plus any maintenance that must be performed on the system (i.e. replace bearings, motors, etc.) must be done in a clean room and then the system must go through a lengthy recertification process.

Strap-down INS

A strap-down system is a major hardware simplification of the old gimballed systems. The accelerometers and gyros are mounted in body coordinates and are not mechanically moved. Instead, a software solution is used to keep track of the orientation of the IMU (and vehicle) and rotate the measurements from the body frame to the navigational frame. This method overcomes the problems encountered with the gimballed system, and most importantly reduces the size, cost, power consumption, and complexity of the system.

Reference Frames and Rotations

There are many different coordinate systems to choose from when doing navigation. Each has its own advantages and disadvantages so their use is really application specific.

- Earth Centered Inertial (ECI), also referred to as inertial for short. This system is fixed in inertial space with its origin located at the center of the Earth. The Earth rotates around the z-axis and the x axis is aligned with the fixed stars.[1]

[1] In reality the stars move, but their change in position relative to Earth is small enough to be neglected.

- Earth Centered Earth Fixed (ECEF) is similar to the inertial system except the axes rotate with the Earth. This system is also used by the U.S. Defense Mapping Agency, which is good since the military maps the world in a grid defined in meters not latitude and longitude. Since we will not be using latitude and longitude, this will be a better system to work with.

- Local Geodetic Vertical (LGV), this system is more convenient when used with latitude and longitude. This system proved to be more complex, harder to understand, and more a computational system to work in though.

- Wander Azimuth Frame (WA) is used in high latitude regions where singularities hamper the previous systems. The previous systems always have an axis that faces North (referred to as North slaved), however the WA adds an additional variable alpha which allows the north facing axis to rotate. This is because magnetic north originates around Greenland, but the navigation equations refer to true north which is located at the rotational axis of the Earth. Thus as aircraft or ballistic missiles pass over the North Pole, their compasses will point to Greenland while navigational north will point somewhere else. The angular difference between the two is alpha.

- Body reference frame is the coordinate system associated with the vehicle. Typically, but not always, the x-axis points out the front of the vehicle, the y-axis is pointed out the right side, and the z-axis is pointed downward. This is the coordinate system in which all measurements will be taken.

Table 3-1: Coordinate system subscripts and superscript definitions.

Symbol	Frame
n	navigation
b	body
c	ECI
e	ECEF

The reference frames which were used are shown in Figure 3-2 and Figure 3-3. Several rotation matrices are needed to transition between the various reference frames. A more detailed look at rotations is presented in Appendix A. The first rotation takes measurements in the body frame and puts them into the navigation frame,

$$R_b^n = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \qquad [3\text{-}1]$$

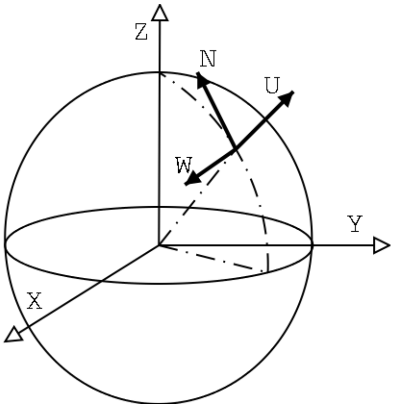

Figure 3-2. The XYZ frame is the inertial frame ECEF and the NWU frame is the local navigational frame, where the axes are north (N), west (W), and up (U).

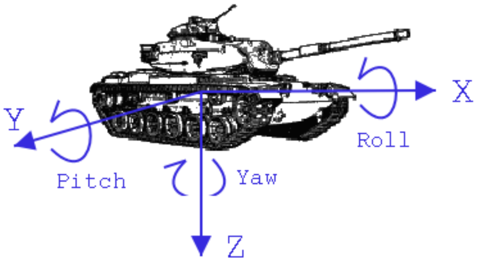

Figure 3-3. Body frame which is aligned with the axes of the IMU. The center of this frame is located at the origin of the navigational frame.

where ϕ is roll, θ is pitch, and ψ is yaw. This rotation is the sequence 1-2-3, which is typically used in aerospace applications. This is a type 1 sequence which has singularities when the pitch is +/- 90 degrees since at this angle both the roll and yaw have similar effects. Thus for fighter aircraft which typically encounter this range, other methods must be included to account for this problem.

The next rotation will transform points from the ECEF frame to the navigation frame,

$$R_e^n = \begin{bmatrix} -s\phi c\lambda & -s\phi s\lambda & c\phi \\ -s\lambda & c\lambda & 0 \\ -c\phi c\lambda & -c\phi s\lambda & -s\phi \end{bmatrix} \qquad [3\text{-}2]$$

where ϕ is latitude and λ is longitude. Now with these two rotations we can get another rotation, the one we really need.

$$R_b^e = R_n^e R_b^n \qquad [3\text{-}3]$$

Modelling the Earth

Periodically all satellite and geodetic data is used to determine the best fitting Earth model. The latest is the World Geodetic System 1984 (WGS 84) and results of some of the important constants are shown below.

Table 3-2: Earth model constants (WGS 84)

Const	Value
R	6,378,137.0 m
ω_{ie}	7.292115E-5 rads/s
g_{WGS0}	9.7803267715 m/s^2
g_{WGS1}	0.0193185138639

Position on the Earth's Surface

Using the model, an extensive derivation (not shown here) can be done to determine a vehicle's position on the Earth given its latitude, longitude, and altitude. The resulting equations are shown below.

$$R_{meridian} = \frac{R_e(1-\varepsilon^2)}{\sqrt[3/2]{1-\varepsilon^2\sin^2\phi}} \qquad [3\text{-}4]$$

$$R_{normal} = \frac{R_e}{\sqrt{1-\varepsilon^2\sin^2\phi}} \qquad [3\text{-}5]$$

$$x_{ECEF} = (R_{normal}+H)\cos\phi\cos\lambda \qquad [3\text{-}6]$$

$$y_{ECEF} = (R_{normal}+H)\cos\phi\sin\lambda \qquad [3\text{-}7]$$

$$z_{ECEF} = (R_{meridiannp}+H)\sin\phi \qquad [3\text{-}8]$$

Gravity Model

The Earth's gravity field is not constant over its surface. One gravity model used to define the expected gravity seen at any point on the Earth is shown below.

$$g = g_{WGS0} \cdot \frac{(1 + g_{WGS1}\sin^2\phi)}{(1 - \varepsilon^2\sin^2\phi)^{1/2}} \qquad [3\text{-}9]$$

$$g^{c}_{SHC} = \begin{bmatrix} \xi g \\ -\eta g \\ g \end{bmatrix} \qquad [3\text{-}10]$$

Navigation Equations

Looking at Newton's second law of motion, a change in motion occurs as a force is applied to a body. Now, dividing both sides of the equation by the mass of the object results in the specific force.

$$\frac{f}{m} = a = S \qquad [3\text{-}11]$$

In inertial navigation, accelerometers detect accelerations due to forces exerted on the body. These forces are typically referred to as specific forces (S). Thus readings from the IMU will be referred to as specific forces, which are independent of the mass.

Position and Velocity

Using the Coriolis theorem, the equations will be derived in the ECI (inertial) reference frame. Remember the ECI frame has its z-axis aligned with the axis of rotation of the Earth and is fixed in inertial space so it does not move.

The Coriolis theorem states that the total velocity of a vehicle (v_i) in a non-rotating reference frame is equal to its ground speed (v_s) plus the added speed due to the rotation of the Earth. Here the subscript i refers to the non-rotating inertial frame, ECI.

$$v_i = v_s + \Omega_{ie} \times r_i \qquad [3\text{-}12]$$

where the rotation of the Earth is $\Omega_{ie} = \begin{bmatrix} 0 & 0 & \omega_{ie} \end{bmatrix}^T$ and r is the vehicles position. Now differentiating this with respect to the inertial reference frame.

$$\dot{v}_i = \dot{v}_s + \dot{\Omega}_{ie} \times r_i + \Omega_{ie} \times v_i \qquad [3\text{-}13]$$

The rotation of the earth is constant ($\Omega_{ie} = const$), therefore its derivative is zero ($\dot{\Omega}_{ie} = 0$). Now substituting for the velocity from the original equation we get.

$$\dot{v}_i = \dot{v}_s + \Omega_{ie} \times v_s + \Omega_{ie} \times [\Omega_{ie} \times r_i] \qquad [3\text{-}14]$$

The acceleration term on the left hand side of the equation can be replaced with all of the accelerations in the system.

$$S^i + g^i = \dot{v}_s + \dot{\omega}_{ie} \times v_s + \Omega_{ie} \times [\Omega_{ie} \times r_i] \qquad [3\text{-}15]$$

Where S^i is the specific force (i.e. all of the accelerations seen by the IMU) and g^i is the gravity model of the system. Finally solving for the acceleration of the vehicle results in the navigation equations which when integrated will turn IMU accelerations into velocities.

$$\dot{v}_s = S^i - \Omega_{ie} \times v_s + g^i - \Omega_{ie} \times [\Omega_{ie} \times r_i] \qquad [3\text{-}16]$$

where the term subtracted from the specific force is the coriolis acceleration and the term subtracted from the gravity is the centripetal acceleration. Although these terms are small, as we will see, small constant acceleration offsets produce a velocity ramp when integrated. This in turn produces a quadratic position change when the velocity ramp is integrated.

Now these equations, which where derived in the inertial frame, must be transformed into the ECEF (Earth) reference frame. Luckily this is an easy task.

$$\dot{v}_e = \dot{v}_s - \Omega_{ie} \times v_s \qquad [3\text{-}17]$$

Thus the inertial equation becomes:

$$\dot{v}_s = S_e - 2\Omega_{ie} \times v_e + g_e - \Omega_{ie} \times [\Omega_{ie} \times r_e] \qquad [3\text{-}18]$$

These equations can also be represented in a state space form.

$$\begin{bmatrix} \dot{V}^e \\ \dot{P}^e \end{bmatrix} = \begin{bmatrix} -2\underline{\Omega}^e_{ie} & -\underline{\Omega}^e_{ie}\underline{\Omega}^e_{ie} \\ I & 0 \end{bmatrix} \begin{bmatrix} V \\ P \end{bmatrix} + \begin{bmatrix} R^e_c & R^e_b \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g^c_{SHC} \\ S^b \end{bmatrix} \qquad [3\text{-}19]$$

where P is the position and V is the velocity vector in the ECEF coordinate system. Also the cross products in the previous equations are replaced with a skew matrix.

$$\underline{\Omega}^e_{ie} = \begin{bmatrix} 0 & -\omega_{ie} & 0 \\ \omega_{ie} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad [3\text{-}20]$$

Attitude

The attitude of the IMU is need so that the measurement taken (which are in the body frame) can be rotated in to the Earth frame. To accomplish this, quaternions will be used to represent the attitude of the IMU (and hence the vehicle itself since the IMU is bolted on to the vehicle). But why use quaternions and what the hell is a quaternion[1] any ways?

Well lets answer the first part of that question. Quaternions have the following good properties (in no particular order) for use in navigational systems.

[1] See Appendix A for a more detailed explanation of quaternions.

- They are computationally efficient. Subsequent rotations (for example a rotation about the x-axis followed by the rotation about the y-axis) is accomplished by multiplying a 4x4 by a 4x1. While in Euler space, two subsequent rotations would be handled by multiplying a 3x3 by 3x3 matrix.

- Quaternions require less memory to represent them. Euler angles require nine elements to represent a rotation (3x3 matrix) while a quaternion only requires four elements. However this is a weak point, but some embedded applications do have very constrained memory requirements.

- The propagation of a quaternion in time always results in an orthogonal rotation matrix. An IMU will supply angular rates from its gyros which will contain noise. Ideally if the IMU is rotated 90 degrees in a plane, the integration of the data from the gyros will properly reflect this 90 degree rotation. However, since there is noise, the IMU will report rotations on all axes. The real problem is that the resulting 3x3 rotation matrix is not guaranteed to be orthogonal any more which means that the inverse of the rotation matrix is not equal to the transpose. Now extra steps must be taken at each time step to make the rotation matrix orthogonal again.

To answer the second part of the question, a quaternion is a way to represent the orientation of a rigid body, but also it is a way to represent a rotation about an arbitrary axis in space. A quaternion can be thought of as a complex number which contains a real and imaginary part. However, the imaginary part of a quaternion contains three values while the real part only has one. The attitude of an object is assumed to be represented by a quaternion in which the imaginary part is the elements 1-3 and the real part is the fourth element.

$$q = \left[q_{1-3}\ q_4\right]^T \qquad [3\text{-}21]$$

$$q_{1-3} = \left[q_1\ q_2\ q_3\right]^T = \hat{n} \cdot \sin\left(\frac{\theta}{2}\right) \qquad q_4 = \cos\left(\frac{\theta}{2}\right) \qquad [3\text{-}22]$$

where $\hat{n}$ is a unit vector corresponding to the axis of rotation and θ is the angle of rotation.

For navigation we will need to be able to convert from roll, pitch, and yaw into quaternion space and back again. This is easily done by the following set of equations.

$$q = \begin{bmatrix} c\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} - s\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} \\ c\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} + s\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} \\ c\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} - s\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \\ s\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} + c\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \end{bmatrix} \qquad \begin{bmatrix} \tan\phi \\ \sin\theta \\ \tan\psi \end{bmatrix} = \begin{bmatrix} \dfrac{2q_1q_2 + 2q_4q_3}{2q_4^2 + 2q_1^2 - 1} \\ 2q_4q_2 - 2q_1q_3 \\ \dfrac{2q_2q_3 + 2q_4q_1}{2q_4^2 + 2q_3^2 - 1} \end{bmatrix} \qquad [3\text{-}23]$$

Alternatively a rotation matrix can be calculated directly from the elements of a quaternion. This is useful since there are no sine or cosine functions called during this process. Trigonometric functions tend to be very computationally expensive and in systems that require high performance are replaced with lookup tables or other faster (less accurate) implementations.

Once the attitude is put into quaternions, the kinematic differential equations for attitude rotation are given as follows.

$$\dot{q} = \frac{1}{2}\underline{\Omega}q \qquad [3\text{-}24]$$

$$\underline{\Omega} = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \qquad [3\text{-}25]$$

This equation allows readings from the gyros to be used to update the current quaternion, which represents the attitude of the system.

Summary of Navigational Equations

The navigation equations for the Earth Centered Earth Fixed (ECEF) system are combined below in state space form.

$$\begin{bmatrix} \dot{V}^e \\ \dot{P}^e \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} -2\underline{\Omega}_{ie}^{e} & -\underline{\Omega}_{ie}^{e}\underline{\Omega}_{ie}^{e} & 0 \\ I & 0 & 0 \\ 0 & 0 & Q \end{bmatrix} \begin{bmatrix} V \\ P \\ \Phi \end{bmatrix} + \begin{bmatrix} R_b^e & R_b^e \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g_{SHC}^{c} \\ S^b \end{bmatrix} \qquad [3\text{-}26]$$

$$\underline{\Omega}_{ie}^{e} = \begin{bmatrix} 0 & -\omega_{ie} & 0 \\ \omega_{ie} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad Q = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \qquad [3\text{-}27]$$

where ω_{ie} is the rotation rate of the earth, R is a rotation matrix between different coordinate systems, P is the position and V is the velocity vector in the ECEF coordinate system as denoted by the superscript e.

CHAPTER 4
CORRECTING INERTIAL NAVIGATION

Low cost inertial navigation unfortunately is not a simple task. There are numerous sources of errors, noise, and disturbances that must be overcome. This chapter will discuss some of these problems and outline a solution. The solution will come in the form of the extended kalman filter. The necessary equations will be derived and discussed.

Sources of Error

This section will provide a quick overview of some difficulties present in inertial navigation, and provide a better understanding for the difficulties encountered with the IMU.

Bias and Drift

These are the most devastating effectors on accuracy to an IMU. Drift rate for the gyros and accelerometer bias are small offsets which the IMU incorrectly reads, that must be properly accounted for. The bias has a quadratic effect on the position derived from the IMU.

$$error = \frac{1}{2} bias \cdot t^2 \qquad [4\text{-}1]$$

Table 4-1: Positional error that results from biases after a time of 100 seconds and 30 mins.

Bias (m/s^2)	Error (m) t=100 sec.	Error (m) t = 30 mins
.1	500	162000
.01	5	16200
.001	.5	1620
.0001	.05	162

Looking at the Table 4-1 it becomes apparent that determining the bias is of critical importance if any accurate measurement is expected.

The drift rate has a similar, and an equally massive impact on the position of a system. If a drift is not properly accounted for, and the IMU thinks it is rotating, then the navigation equations will not properly account for gravity and the system will think it is moving due to a maximum acceleration of 9.8 m/s^2 depending on how far the system has drifted.

Temperature

The IMU's accelerometers and gyros are sensitive to temperature as shown by Nebot and Durran-Whyte [10]. Thus as the temperature of the IMU changes, the associated bias and drift will change until the temperature reaches steady state or remains the same. This is not critical in our application, we just wait for the IMU to reach steady state before trusting the readings. However if this system was mounted in an aircraft which changed altitude and temperatures, this would be a problem.

Hysteresis

The drift rates and accelerometer biases tend to change each time the unit is switched on. This is due to the fact that measurements are noisy. Typically there is a low pass filter used to remove some of this noise before the measurements are used in the navigation equations (also realistically, there tends to be low pass filtering somewhere in the system due to hardware limitation because not everything has infinite bandwidth). When random noise is filtered, this produces what is called a random walk. The integration of this random walk will result in velocity and positions moving at different rates during different runs even though the IMU (and vehicle) are in the same orientation and experiencing the same accelerations during each run.

To give an idea of the performance of a strap-down system, the following quote is taken from an article [11] written by A. D. King, Chief Engineer of Navigation and Electro-optic Systems Division of Marconi Electronic Systems. Marconi produces INS for virtually all of the RAF's combat aircraft as well as many other systems.

> *"Many of these instrument errors vary each time you switch the system on - INS have good days and bad days. To characterize the performance of an INS, you have to resort to statistics, and take the r.m.s. total error from an ensemble of many representative missions. A typical standard expected from a 'good' INS produces an error that increases with time (not an entirely linear fashion), and reaches .6 miles after one hour (referred to as .6 nautical miles/hour system)."*

Vibrations

Vibration in a strap-down system can cause many problems with the INS. Generally great care must be taken to isolate the IMU from any resonance frequencies. In high precision systems, various tests must be done to try to identify what these frequencies are then design elaborate mounts to hold the IMU.

Extended Kalman Filter

An extended kalman filter was developed to estimate the biases and drifts of the system and then update the navigational solution. The full kalman filter equations are presented in the Appendix B, but an overview of the process is shown in Figure 4-1 and further information can be found in Brown and Hwang [12].

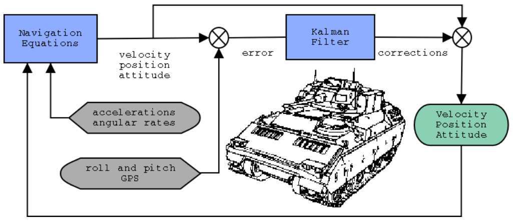
Navigation Equations
velocity position attitude
error
Kalman Filter
corrections
accelerations angular rates
roll and pitch GPS
Velocity Position Attitude

Figure 4-1. Overview of the extended kalman filter's integration with the INS.

Kalman filtering relies on a dynamic model of the system. The following error models for both position and attitude were developed based on derivations by Chatfield [13] and Rogers [14].

Position Error Model

The position of an object in the inertial frame rotated in to the earth frame is given below.

$$P^e = R_i^e P^i \tag{4-2}$$

Taking the derivative of this equation and replacing the derivative of a rotation matrix with its alternative form results in:

$$\dot{P}^i = \underline{\Omega}^i_{ie} R^i_e P^e + R^i_e \dot{P}^e \tag{4-3}$$

Again, taking the derivative and some more algebra.

$$\ddot{P}^e = R^i_e(\ddot{P}^e + 2\underline{\Omega}^e_{ie}\dot{P}^e + \underline{\Omega}^e_{ie}\underline{\Omega}^e_{ie}P^e) \tag{4-4}$$

Finally the equations of motion for navigation are found. This derivation resulted in the same answer as previously given. This form is assumed to represent both the computed and real dynamics of navigation in the earth frame.

$$\ddot{P}^e + 2\underline{\Omega}^e_{ie}\dot{P}^e + \underline{\Omega}^e_{ie}\underline{\Omega}^e_{ie}P^e = g^e + S^e \tag{4-5}$$

Now defining the actual value ($\bar{P}$) as the computed value (P) minus some error (δP) and substituting this into the above equation ($\bar{P} = P – \delta P$) where we are assuming the equation represents the actual values.

$$(\ddot{P}^e + 2\underline{\Omega}^e_{ie}\dot{P}^e + \underline{\Omega}^e_{ie}\underline{\Omega}^e_{ie}P^e) - (\delta\ddot{P}^e + 2\underline{\Omega}^e_{ie}\delta\dot{P}^e + \underline{\Omega}^e_{ie}\underline{\Omega}^e_{ie}\delta P^e) = S^e + \delta S^e + g + \delta g \tag{4-6}$$

Now canceling the computed values on each side and assuming the $\underline{\Omega}_{ie}^{e}\underline{\Omega}_{ie}^{e} \cong 0$ and the error in gravity is also negligible, we have a way to estimate error in the system.

$$\delta\ddot{P}^{e} + 2\underline{\Omega}_{ie}^{e}\delta\dot{P}^{e} + \underline{\Omega}_{ie}^{e}\underline{\Omega}_{ie}^{e}\delta P^{e} = \delta S^{e} \qquad [4\text{-}7]$$

The last remaining term on the right hand side of the equation still has to be defined. Using the idea that the error in the measured accelerations is due to misalignment of the system, we can rewrite the right hand side (where the terms with an underline represent skew matrices).

$$\delta S^{e} = \delta R_{b}^{e} S^{b} = -\delta\underline{\phi}^{e} R_{b}^{e} S^{b} \qquad [4\text{-}8]$$

$$= -\delta\underline{\phi}^{e} S^{e} = \underline{S}^{e}\delta\phi^{e} \qquad [4\text{-}9]$$

However this derivation made use of the following definition by Chatfield [13]:

$$\delta R_{b}^{e} = R_{b}^{e} - \underline{R_{b}^{e}} = -\delta\underline{\phi}^{e} R_{b}^{e} \qquad [4\text{-}10]$$

Attitude Error Model

Similar to how the position error was derived, the attitude error model starts off by looking at a rotation from the body frame to the earth frame. Then the definition for computed values is substituted into the equation and simplified.

$$\bar{R}_{b}^{e} = \bar{R}_{n}^{e}\bar{R}_{b}^{n} \qquad [4\text{-}11]$$

$$R_{b}^{e} - \delta R_{b}^{e} = (R_{n}^{e} - \delta R_{b}^{n})(R_{b}^{n} - \delta R_{b}^{n}) \qquad [4\text{-}12]$$

$$-\delta R_{b}^{e} = -R_{n}^{e}\delta R_{b}^{n} - \delta R_{n}^{e} R_{b}^{n} \qquad [4\text{-}13]$$

where the product of error is zero and the term $R_{b}^{e} = R_{n}^{e}R_{b}^{n}$ cancels itself out. Now post-multiplying both sides with R_{e}^{b} and renaming the terms.

$$\delta\underline{E}^{e} = \delta\underline{N}^{e} + \delta\underline{\Phi}^{e} \qquad [4\text{-}14]$$

where

$$\delta E^e = -\delta R^e_b R^b_e \qquad [4\text{-}15]$$

$$\delta \underline{\Phi}^e = -R^e_n \delta R^n_b R^b_e = -R^e_n \delta R^n_b R^b_n R^n_e = -R^e_n \delta \underline{\Phi}^n R^n_e \qquad [4\text{-}16]$$

$$\delta \underline{N}^e = -\delta R^e_n R^n_e \qquad [4\text{-}17]$$

The term δE^e is a skew-symmetric form of the attitude error, $\delta\Phi^e$ is the skew-symmetric form of the error in the orientation of the instrument cluster, and the δN^e is the skew-symmetric error of the angular equivalent of the position error.

Now that we have the nifty definitions of attitude error, we need to find the error in the earth frame from our calculations of euler angles. Using our definition of the computed rotation matrix we have

$$\bar{R}^e_b = R^e_b - \delta R^e_b \qquad [4\text{-}18]$$

Now pulling out true rotation from the right hand side, substituting in our definition of error, and simplifying.

$$\bar{R}^e_b = (I - \delta R^e_b R^b_e) R^e_b = (I + \delta \underline{E}^e) R^e_b \qquad [4\text{-}19]$$

Next, take the derivative and solve for error.

$$\dot{\bar{R}}^e_b = \delta \dot{\underline{E}}^e R^e_b + (I + \delta \underline{E}^e) \dot{R}^e_b \qquad [4\text{-}20]$$

$$-\underline{\Omega}^e \underline{\bar{R}}^e_b = \delta \dot{\underline{E}}^e R^e_b - (I + \delta \underline{E}^e) \underline{\Omega}^e R^e_b \qquad [4\text{-}21]$$

$$\delta \dot{E}^e = -\underline{\Omega}^e \bar{R}^e_b R^b_e + (I + \delta \underline{E}^e) \underline{\Omega}^e \qquad [4\text{-}22]$$

Substituting in equation [4-19] and some more algebra.

$$\delta \dot{\underline{E}}^e = -\underline{\Omega}^e (I + \delta \underline{E}^e) + (I + \delta \underline{E}^e) \underline{\Omega}^e \qquad [4\text{-}23]$$

$$\delta\dot{\underline{E}}^{e} = -\underline{\Omega}^{e}\delta\underline{E}^{e} + \delta\underline{E}^{e}\underline{\Omega}^{e} + \delta\underline{\Omega}^{e} \qquad [4\text{-}24]$$

Now comes the magic, Chatfield then claims the above equation can be reduced to the following where the skew error matrices are transformed in to error vectors.

$$\delta\dot{e}^{e} \approx \underline{\Omega}^{e}\delta e^{e} - \delta\omega_{b}^{e} \qquad [4\text{-}25]$$

The instrument attitude error vector deferential equation is derived in a similar way.

$$\delta\dot{\phi}^{e} \approx \underline{\Omega}^{e}\delta\phi^{e} - \delta\omega_{b}^{e} \qquad [4\text{-}26]$$

Summary of Error Model Equations

The complete model is presented below. This filter model is small compared to other models which have anywhere between 20 and 50 different states, depending on how their navigational models were defined. Note that there is also the inclusion of two sets of terms which now makes this an extended kalman filter model. The terms are the errors in bias on the accelerometers, and drift of the gyros. Each is modelled as a random walk (or could have modelled them as a Markov process), where the terms with the subscript N on the far right of the equation are zero mean, random white noise with the appropriate standard deviation. The purpose of this is to estimate these new parameters, since they are difficult to determine, and (as in the case of the bias) change greatly depending on temperature, time, and orientation.

$$\begin{bmatrix} \delta\dot{V} \\ \delta\dot{P} \\ \delta\dot{\phi} \\ \delta\dot{S}^{b} \\ \delta\dot{\omega}^{b} \end{bmatrix} = A\begin{bmatrix} \delta V \\ \delta P \\ \delta\phi \\ \delta S^{b} \\ \delta\omega^{b} \end{bmatrix} + B\begin{bmatrix} \delta S_{N}^{b} \\ \delta\omega_{N}^{b} \end{bmatrix} \qquad [4\text{-}27]$$

$$A = \left[\begin{array}{ccccc} -2\underline{\Omega}_{ie}^{e} & \underline{\Omega}_{ie}\underline{\Omega}_{ie} & \underline{S}^{e} & R_{b}^{e} & 0_{3x3} \\ I_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & \underline{\Omega}_{b}^{e} & 0_{3x3} & -R_{b}^{e} \\ \multicolumn{5}{c}{} \\ \multicolumn{5}{c}{0_{6x15}} \\ \multicolumn{5}{c}{} \end{array}\right] \quad B = \begin{bmatrix} 0_{6x6} \\ I_{6x6} \end{bmatrix} \quad \underline{S}^{e} = \begin{bmatrix} 0 & -S_{z}^{e} & S_{y}^{e} \\ S_{z}^{e} & 0 & -S_{x}^{e} \\ -S_{y}^{e} & S_{x}^{e} & 0 \end{bmatrix} \qquad [4\text{-}28]$$

CHAPTER 5
HARDWARE AND EXPERIMENTAL SETUP

This chapter will provide an overview of hardware typically used in inertial navigation. Also the two primary sensors used in this work, the IMU and GPS, will be discussed.

Standard IMU Hardware

The two sensors that make up an IMU are the gyros and accelerometers. The gyros are the most important since they will be used to determine the orientation of the system. This is crucial since this allows the INS to properly account for gravity and other known biases, in which even small errors can quickly accumulate in the system.

Gyroscopes

Gyroscopes, or more commonly gyros, are sensors that measure the rotation rate of a system. This is important so the measurements taken by the IMU can be properly orientated from body coordinates to some navigational reference frame. There are many different types of gyros present on the market.

Ring Laser Gyro (RLG)

The RLG shown in Figure 5-1 is very expensive, but accurate single degree of freedom sensor for rate determination designed to replace mechanical gyros. It consists of a laser, a closed pathway (two way) in a triangular shape, mirrors at each corner, and an interferometer/photodetector. The frequency with which laser pulses travel around the pathway is constant when the gyro is sitting still. However, as the gyro is rotated the frequency increases or decreases depending on if the photodetector has rotated closer or far-

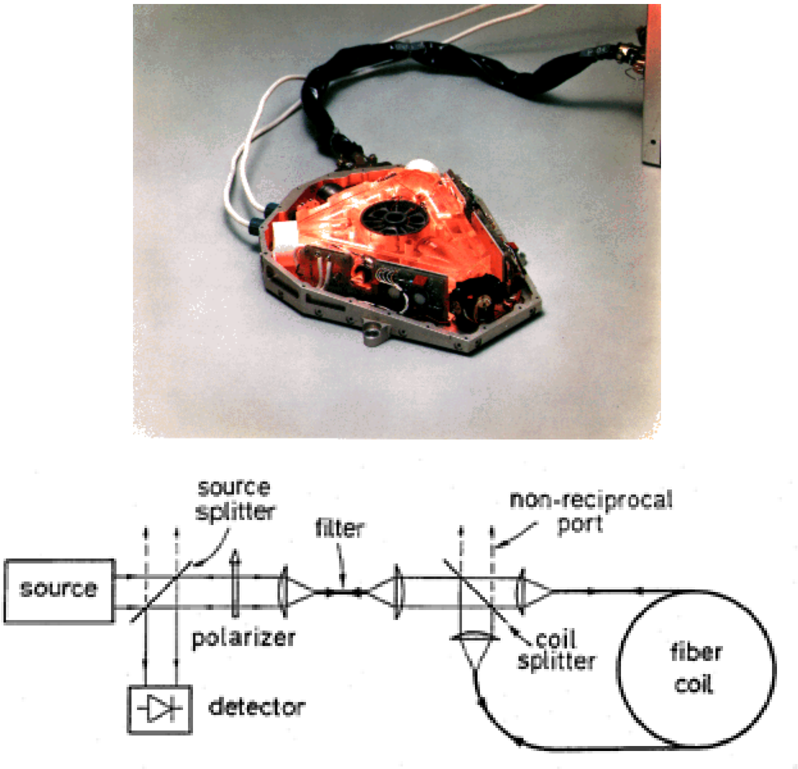

Figure 5-1. Ring laser gyro shown at top (note the triangular shape) and fiber-optic gyro diagram shown below.

ther from the emitted laser pulse. Thus rotation rate can be determined from the frequency change of the laser. A laser beam is typically sent in both directions of the configuration.

Fiber-Optic Gyros (FOG)

The FOG shown in Figure 5-1 is a lower cost alternative to RLG, where the path is a coil of fiber-optics and there is a beam splitter located at the beginning and end of the coil. There is also a laser source and a detector so that the frequency of the laser light can be determined.

MEMS

Micro-Electro-Mechanical Systems (MEMS) is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through the utilization of microfabrication technology. While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micromechanical components are fabricated using compatible “micromachining” processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices. MEMS promises to revolutionize nearly every product category by bringing together silicon-based microelectronics with micromachining technology, thereby, making possible the realization of complete systems-on-a-chip. MEMS is truly an enabling technology allowing the development of smart products by augmenting the computational ability of microelectronics with the perception and control capabilities of microsensors and microactuators. MEMS is also an extremely diverse and fertile technology, both in the applications, as well as in how the devices are designed and manufactured. MEMS technology makes possible the integration of microelectronics with active perception and control functions, thereby, greatly expanding the design and application space.

Some pictures of MEMS devices are shown in Figure 5-2 to give an idea of the size. Here these microscopic creatures appear to be giants compared to the miniature mechanical gears they walk over.

Another device is shown in Figure 5-3. This one is a 6 DOF IMU made by Integrated Micro Instruments (which was bought out by Analog Devices Inc.). The device measures only 5 mm by 9 mm and is complete with three gyros and three accelerometers.

MEMS is not limited to just sensors but also propulsion. A MEMS thruster made by TRW Space and Electronics is shown in Figure 5-4 which could be used in space applica-

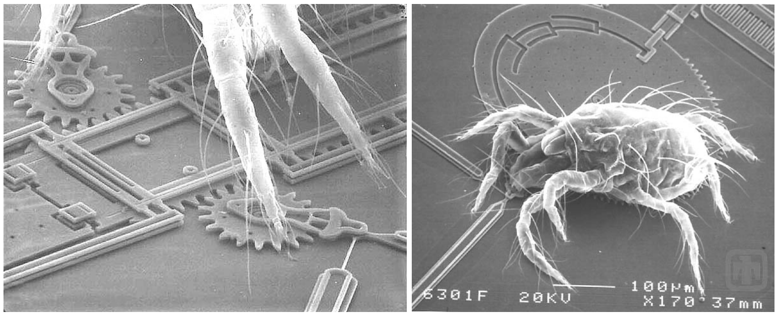

Figure 5-2. Spider mites walking on some MEMS parts.

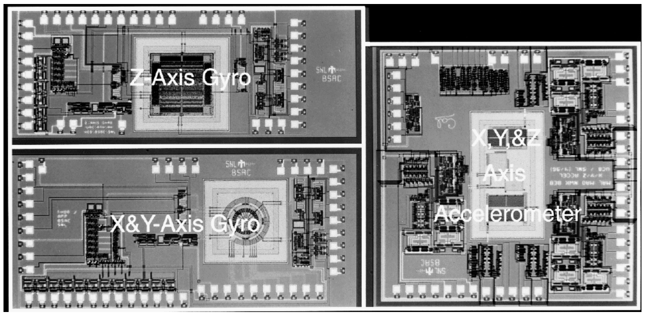

Figure 5-3. MEMS IMU.

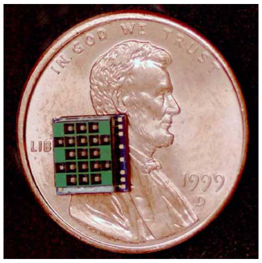

Figure 5-4. MEMS thrusters.

tions. This device is capable of delivering 10E-4 Nsec of impulse from the poppy seed sized cells which contain the lead styphnate fuel propellant.

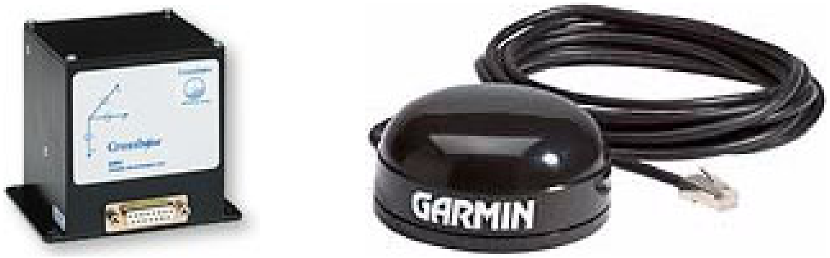

Figure 5-5. Sensors used in the INS. (left) The Crossbow DMU-HDX which is a solid state vertical gyro capable of measuring angular rates and accelerations on all three axes. It also has the capability of measuring the roll and pitch of the device too. (right) Garmin 16LVS OEM GPS which is both a receiver and antenna.

Hardware Used

This section will cover the two main pieces of hardware used in this thesis, the IMU and the GPS shown in Figure 5-5. An attempt will be made to evaluate their performance and show their shortcomings when used individually.

Crossbow IMU

The IMU is a solid state vertical gyro (DMU_HDX) from Crossbow Technologies intended for airborne applications such as UAV control, Avionics, and Platform Stabilization. This high reliability, strap-down inertial subsystem provides attitude measurement with static and dynamic accuracy comparable to traditional spinning mass vertical gyros. Data will be transmitted by the DMU digitally via a serial connection (RS-232) and converted to the proper format using the conversions in Table 5-1. The gyros on the Crossbow IMU are low cost, low performance MEMS (Mechanical Electrical Micro-Systems) gyros. These gyros are much less expensive to produce, but performed at least an order of magnitude worse than another low cost IMU system being developed by Rommel Mandapat [2]. That system uses an IMU developed from Honeywell which has ring laser gyros. Unfortunately, the gyro performance is a critical element in accounting for gravity in the system.

Table 5-1: Data conversions for the Crossbow IMU.

Data	Equation
X, Y, and Z Acceleration	$Accel(G) = data \cdot GR \cdot \frac{1.5}{2^{15}}$
Roll and Pitch	$Angle(°) = data \cdot \frac{180}{2^{15}}$

Table 5-1: Data conversions for the Crossbow IMU.

Data	Equation
Angular Rates	$Rate(°/s) = data \cdot RR \cdot \frac{1.5}{2^{15}}$
Internal Temperature	$Temperature(°C) = \left(data \cdot \frac{5}{4.096} - 1.375\right) \cdot 44.44$

IMU Performance

Lets look at the performance of the IMU a little closer. As discussed earlier, temperature plays a role in the accuracy of the IMU. The first test, shown in Figure 5-6, shows the change in the accelerometer readings over a period of 2.5 hours while the IMU was sitting still on a table. Since the IMU is capable of recording its' temperature, we also have a temperature history during this test. The IMU's accelerations do change slightly (remember that a offset of only .001 over a period of 30 minutes will result in a position error of 1620 m) but the exact amount of change is difficult to see due to the excessive amount of noise in the IMU readings. Thus the data will have to filtered prior to using it.

All of the little offsets on the accelerometers, noise, temperature effects, etc will amount to positional error, but how much? Again, taking four sets of data while the IMU was sitting still on a table and subtracting off the mean values of each set, which are assumed to be the biases. Each data set covered approximately 35 minutes and data was taken at 10 Hz from the x axis accelerometer. Then integrating the resulting data should result in zero positional change since it is not moving and biases were taken care of by subtracting of the mean values (assuming that the noise seen in the system is random white noise with a zero or constant mean). However as shown in Figure 5-7 this is shown to not be true. The IMU does think it has moved because the noise, temperature effects,

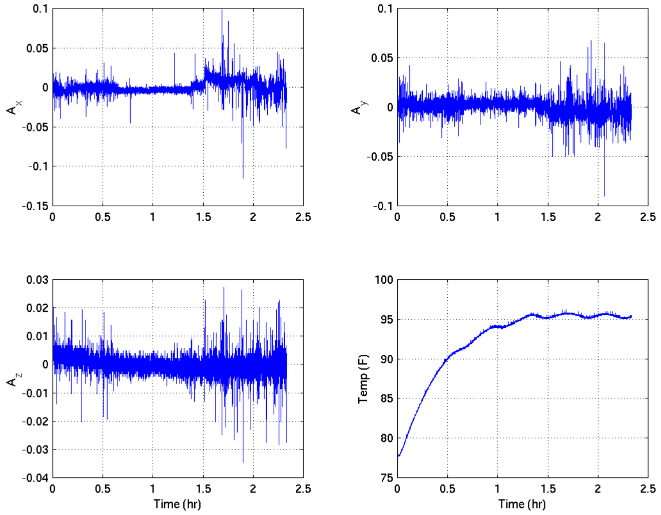

Figure 5-6. Change in accelerometer reading over time due to temperature change.

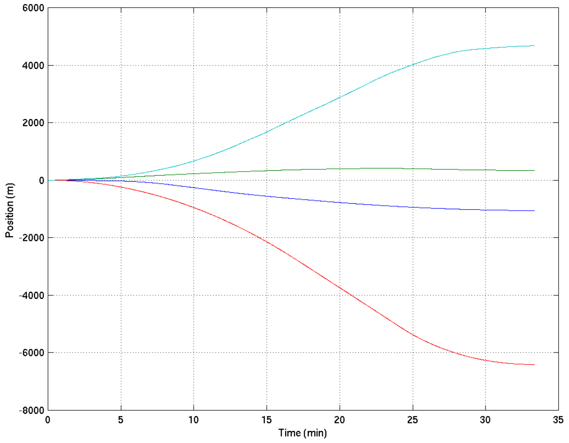

Figure 5-7. Random movement of system due to hysteresis.

biases, etc (all or some) must be changing with time to produce these offsets. The system thinks that it has move -6410 m, -1054 m, 343 m, or 4673 m depending on which set of

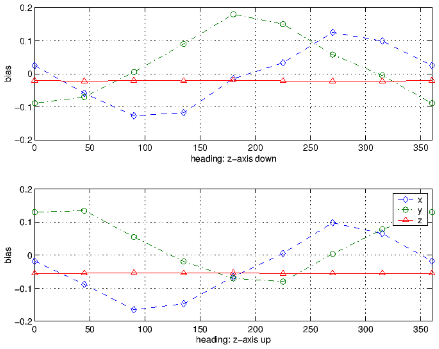

Figure 5-8. This is a plot of the biases as the IMU was rotated around the z-axis (yaw). Rotations around the other axes would also effect the biases, thus this mapping is not useful since the values are changing nonlinearly.

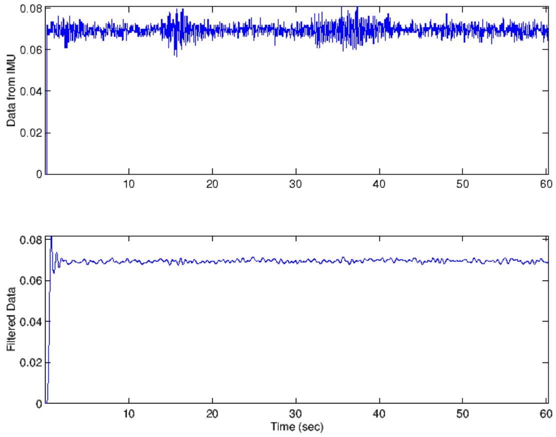

Figure 5-9. Comparison of the unfiltered data (top) produced by the IMU and the filtered data (bottom) using the Chebyshev II filter.

data you look at. Thus not only do the biases change with orientation and temperature, but also it depends on when you take the data.

Prefiltering IMU Data

The data produced by the IMU is extremely noisy, thus a filter was designed. Matlab's signal toolbox was used to accomplish this task. The toolbox is capable of designing all of the classic FIR and IIR filters. A IIR filter was decided on since it produces the same results as an FIR filter but with a much lower order. This lower order results in a less computational process. The following specifications for the filter were decided on:

Table 5-2: IMU prefilter specifications.

Specification	Value
Pass Band	2 Hz
Stop Band	3 Hz
Stop Band Attenuation	-50 dB

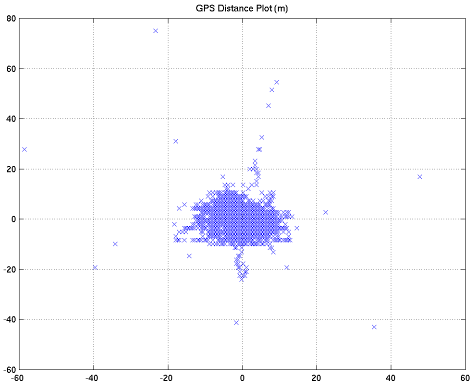

Figure 5-10. This is a test of the GPS accuracy. The GPS was set in a stationary location for 4 hours. The center of the plot was taken to be the average latitude and longitude reported by the sensor. Then the corresponding distances from the average were calculated. This GPS receiver is capable of providing the standard 10 meter accuracy 95% of the time.

Additionally, the desired filter should not have any ripple in the pass band range, thus the Equiripple, Elliptic, and Chebyshev I filters were eliminated as possible designs. The remaining Butterworth and Cheyshev II filters were looked at. After much testing with various options, the Chebyshev II filter was settled one as the best one for the job and its performance can be seen in Figure 5-9.

Garmin GPS

The GPS system used in this work is the Garmin 16LVS. Garmin is a common name in commercial civilian GPS systems, and this OEM device has performance that is on par with all other GPS systems available currently (i.e. accuracy of about 10 m 95% of the time) as shown in Figure 5-10. However this GPS was specifically bought because it included a WAAS (Wide Area Augmentation System) filter which should increase the accuracy to less than 3 m 95% of the time.

Experimental Setup

The two primary pieces of hardware used for this these have been covered, and now the rest of the setup will be explained. The IMU and GPS were connected to a 200 MHz laptop running Linux[1] by serial[2] connections. The IMU and GPS were powered from a battery pack. The INS was mounted in a car (with the GPS magnetically mounted to the outside roof) and driven on local streets.

The navigational software was a multi-threaded program with five main threads. Threads are a more modern way to do multi-processing using a standard called pthreads or POSIX Threads. The main thread displayed various information to the screen during the experiment using menu system written in ncurses. The second thread talked to the IMU over the serial connection. It also did the prefiltering of the data and then inserted the information into shared memory. A third thread talked to the GPS and inserted that information into shared memory. Another thread performed the navigation using the information that was being inserted into shared memory. All of the navigational equations and the kalman filter were implemented in C. Several libraries had to be written to perform matrix and vector operations along with numerical integration of differential equations and kalman filtering. The final thread was a data logger which wrote selected information from shared memory to a data file for later analysis and plotting in Matlab.

[1] Linux is a free unix clone operating system written by Linux Torvalds.

[2] Laptops only come equipped with a single serial port, thus the second serial port needed was a USB to serial converter.

CHAPTER 6
RESULTS

The experiment is broken up into two parts. The first part is the navigation solution which does not utilize the kalman filter or the GPS positional corrections. The second part will include these so the limitation of the IMU and benefits of the kalman filter and GPS can be seen.

Navigational Solution Only

The first set of results was without the use of the extended kalman filter, to see if it was really necessary. The INS was driven along the route shown in Figure 6-1. During the test (although not used in this phase of the test) the GPS was able to see plenty of satellites to generate good 3D positions. The results of estimating the roll, pitch, and yaw without any corrections is shown in Figure 6-3. The estimated angles appear to track the true angles to

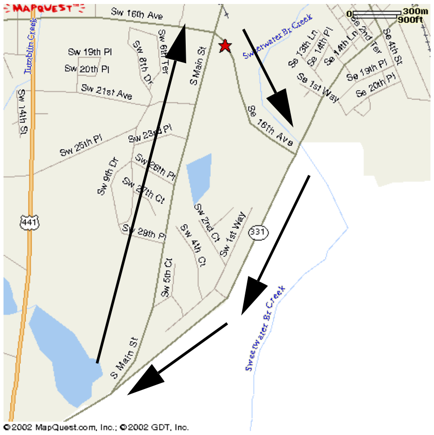

Figure 6-1. Map of the entire route taken from www.mapquest.com

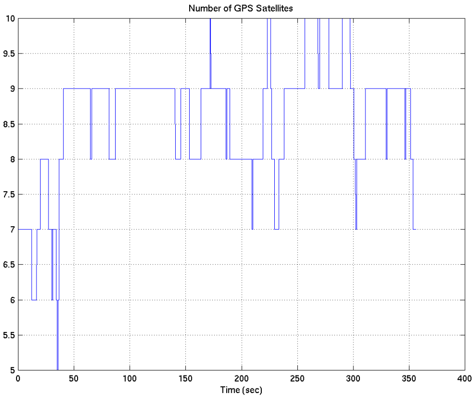

Figure 6-2. Number of satellites seen by the GPS receiver during the test.

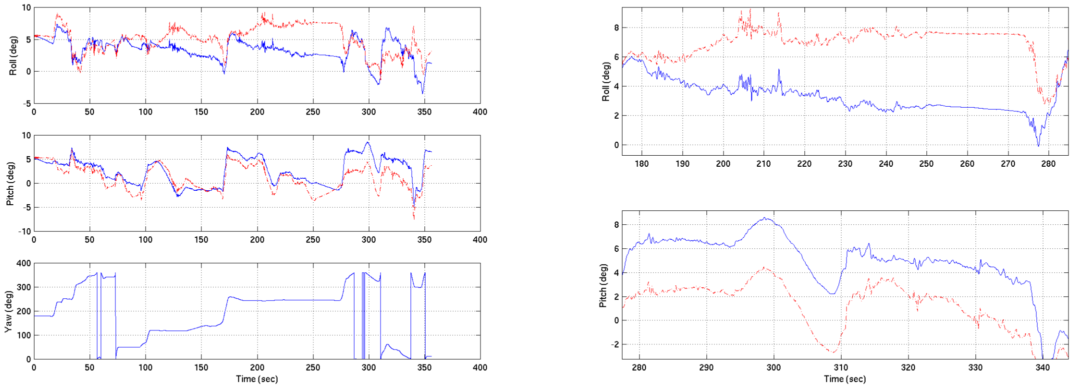

Figure 6-3. INS attitude solution with out extended kalman filter. The estimated roll, pitch, and yaw are shown by the solid line, while the true roll and pitch reported by the IMU is the dashed line.

an acceptable degree. The IMU is capable of reporting it's true roll and pitch, but not yaw. Assuming the performance between estimating the yaw, pitch, and roll angles are the same, it should not be necessary to require a compass to update the true yaw angle. The couple degrees of error should not effect INS results much since the car is traveling on flat roads.

The performance changes when we look at Figure 6-4. The GPS and INS (i.e. using the navigation equations and IMU data only) differ greatly. Thus the GPS with the kalman filter must be included into the INS to give any good results.

GPS/INS

After the inclusion of the GPS and kalman filter, the plot shown in Figure 6-5 is much better. The GPS and INS lie right on top of each other. Taking a closer look at this plot, Figure 6-6 and Figure 6-7 show that the two do not really lie exactly on top, but rather the INS transitions smoothly through the GPS points.

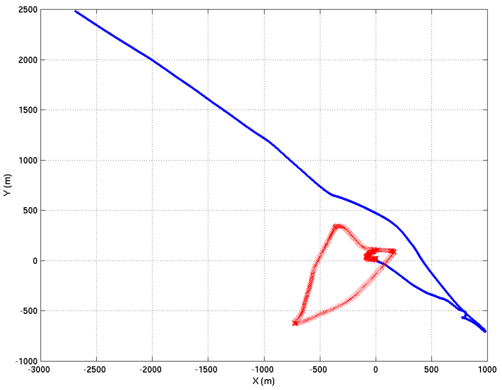

Figure 6-4. INS results without GPS and kalman filter integrated into the system.

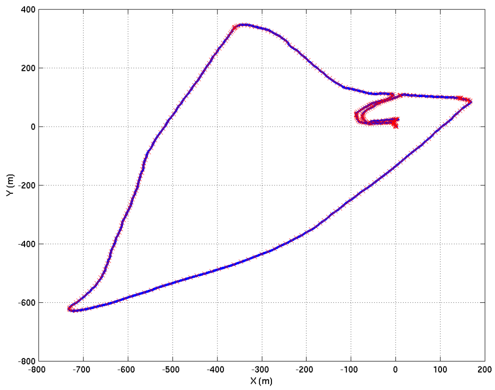

Figure 6-5. INS results with GPS and kalman filter integrated into the system.

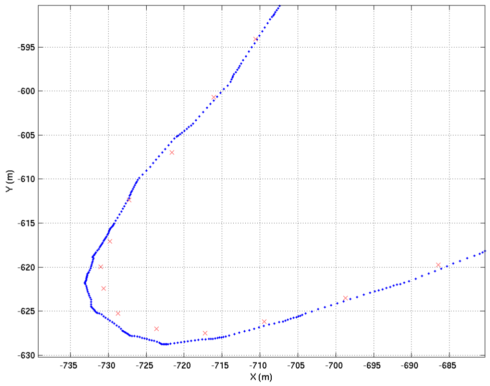

Figure 6-6. This plot shows the interpolating capabilities of the INS system in X and Y.

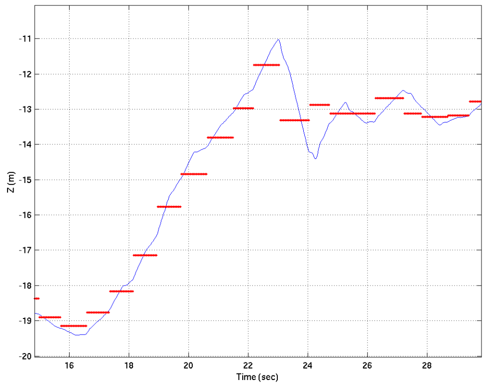

Figure 6-7. This plot shows the interpolating capabilities of the INS system in Z.

Looking specifically at Figure 6-6, it can be seen that the IMU is picking up some of the accelerations in the turn and shifted the position left of the GPS points. But going into the turn and once the turn is completed, the INS and GPS positions merge back together.

Figure 6-7 is a better example showing how the INS is able to take the discrete GPS position and the accelerations from the IMU and fit a curve through the two. This level of continuous positioning can not be offered by GPS alone.

Finally the distances traveled during the experiment were calculated and the results were close as shown in Table 6-1. The car's odometer was felt to be the most accurate and the GPS and INS distances are on either side of the value.

Table 6-1: Distances traveled as reported by the different systems.

	GPS	INS	Car
km	4.89	5.16	
miles	3.04	3.21	3.1

The extended kalman filter attempts to estimate the biases and drifts present in the system to increase the accuracy of the system. However there appeared to be no difference between using the estimated biases and drifts from the filter or using constant ones. This is attributed to the excessive amount of noise from the low cost IMU. Chatfield's [13] work was the prime motivator for including these terms in the extended kalman filter, but he assumed measurements that were much better (i.e. less noisy) than the ones being produced by the Crossbow IMU. Thus this part of the kalman filter could be eliminated to reduce computational expense with no loss of performance.

Another test was conducted. The route is shown in Figure 6-8 and the result from the INS are shown in Figure 6-9. There is a good correspondence between the INS solution and the route taken.

Again the influence of the accelerometers can be seen in the data for the intersection of Archer Road and 34th Street, shown in Figure 6-10. Another interesting result seen in this plot, is the path just prior to the second stoplight. After the turn was made onto 34th Street, the vehicle switched from the right lane, to the center lane, and back to the right lane again. These motions were picked up nicely by the INS.

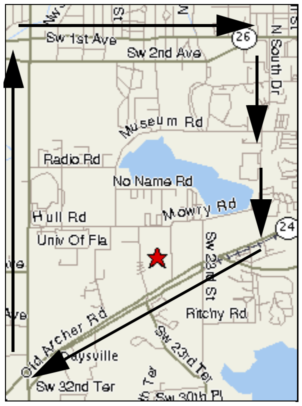

Figure 6-8. Route taken for second test: starting at the commuter parking lot take North-South Drive, Archer Road, 34th Street, University, and back.

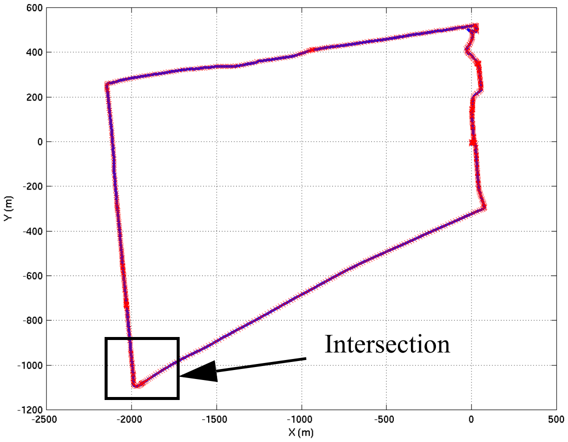

Figure 6-9. Results from the INS which match good with the map of the route.

Several stops were made during the trip due to stoplights on the route. The results are shown in Figure 6-11 for one of the stops. Ideally the INS solution would not move since the vehicle is stopped and there would be no accelerations. The GPS positions would

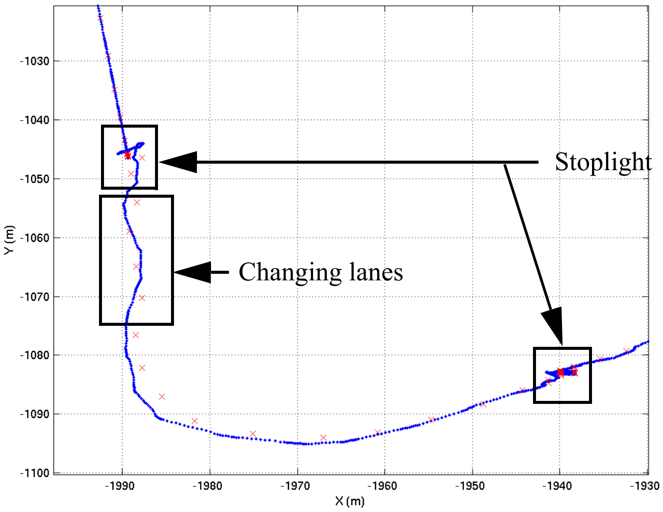

Figure 6-10. Corner of Archer Road and 34th Street.

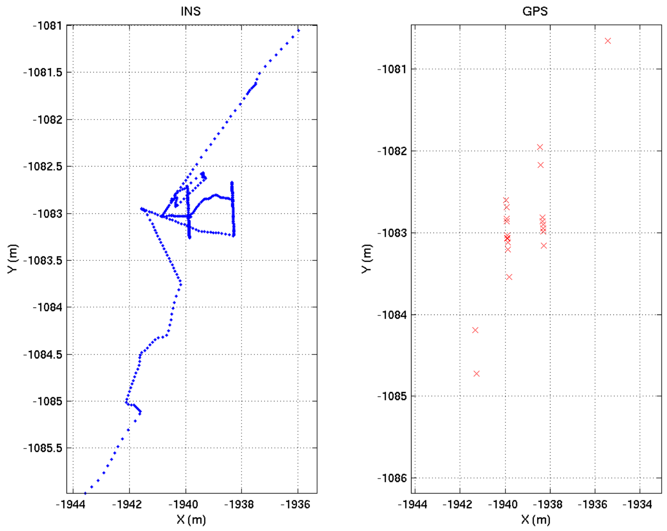

Figure 6-11. The GPS data and INS solution for one of the stoplights on Archer Road.

move due to the amount of error associated with GPS. The more sensors that can be incorporated into the system, the better the results would be. If the velocity of the vehicle, from the speedometer could be incorporated into the system, this would help the INS to reject the moving GPS data and the noisy IMU reading and should show that the vehicle is standing still.

The distances reported by the three systems are shown below. In this case, the GPS distance were closer to the distance measured by the odometer. This is accounted for by the fact that the road surface was rougher (i.e. more bumps in the road) which effected the IMU's accelerometers.

Table 6-2: Distances traveled as reported by the different systems.

	GPS	INS	Car
km	9.23	9.46	
miles	5.73	5.87	5.61

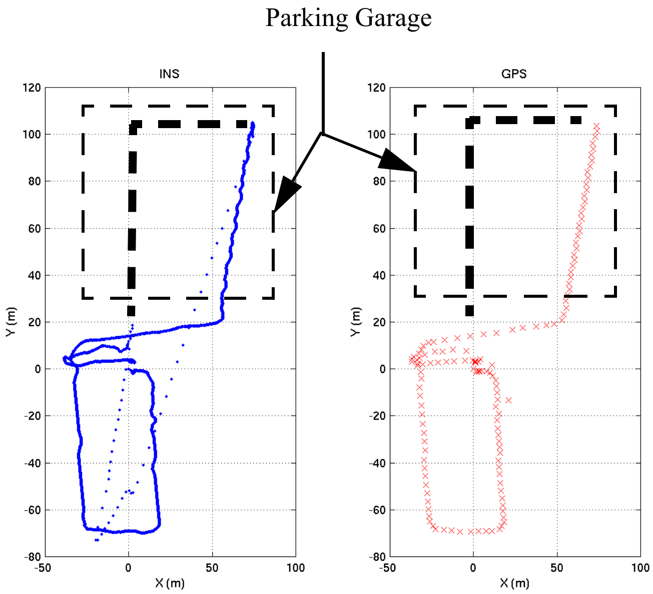

Figure 6-12. The results from driving through a parking garage which blocks the GPS signal. The INS solution is shown on the left while the GPS readings are shown on the right. The location of the parking garage and the correct path are drawn on the plots.

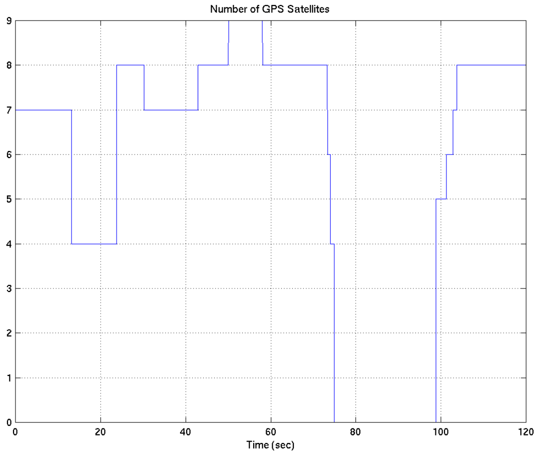

Figure 6-13. Number of satellites seen during the experiment.

The final test looked at the effects of loosing the GPS signal for a short period of time. The INS was first driven in a parking lot and then into a parking garage. Inside the garage, the GPS signal was eventually lost. The results are shown in Figure 6-12. After entering the parking garage, the GPS eventually looses the satellite signal (shown in Figure 6-13) and is unable to determine its position. Again, the inclusion of other sensors to aid the INS (in addition to a better IMU) would have produced better results.

CHAPTER 7
CONCLUSIONS

This thesis has shown the effective combination of two different sensors: GPS and IMU. Each sensor alone has its own strengths and weaknesses. The "low cost" IMU used in this work is not capable of running by itself and providing any reasonable positioning information. GPS alone provides good results, but is only capable of determining position every second with 10-15 meters of error 95% of the time. The two sensors combined have the capability of producing good estimates of position in between the one second updates and overcoming their individual weaknesses. Together they are capable of showing the small scale movements (i.e. vehicle changing lanes) that are not apparent with GPS alone due to accuracy and the 1 Hz position rate.

APPENDIX A
ATTITUDE REPRESENTATIONS AND ROTATION MATRICES

When dealing with a subject such as inertial navigation, controlling spacecraft, or computer vision, it is common to have to deal with multiple reference frames and have to relate the position of one point in one frame to its position in another reference frame. Thus it is important to know how to rotate points from one frame to another efficiently and understand what limitations may be present in various methods. This Appendix will attempt to familiarize the reader with quaternions, rotations, attitude in space, and euler angles. Comparisons will be drawn between euler angles and quaternions to show their strengths and weaknesses.

Fixed Angle Rotations

Typically when trying to relate a common point between different reference frames, a rotation matrix will be constructed. This matrix will allow easy transformation between frames.

$$P^b = R_a^b P^a \qquad \text{[A-1]}$$

Here a point (P) is rotated from where it is originally defined, in reference frame a, to another reference frame b. Not that the rotation matrix R does not include any translation, but only rotation. Note also that R has the following properties:

Table A-1: Properties of a rotation matrix.

Rotation Matrix R
$R^{-1} = R^T$
$R_a^b = \begin{bmatrix} x_a & y_a & z_a \end{bmatrix} = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}$
$\lVert R \rVert = 1$
$\lVert column_i(R) \rVert = \lVert row_i(R) \rVert = 1$
$R_a^c = R_b^c R_a^b$

When a rigid body is rotated about a fixed inertial reference frame, it is referred to as a fixed angle rotation. These types of rotations are common in robotics where one is trying to determine the joint angles of a serial robot relative to some point in space. What is important to understand, is rotations of this type are made about the inertial axis which is inconvenient. A better method would be rotations about the body axes. These type of rotations are known as Euler Angles.

Euler Angles

Euler angles are typically though of in terms of roll, pitch, and yaw angle about body axes. These terms are shown graphically below for a rigid body in space.

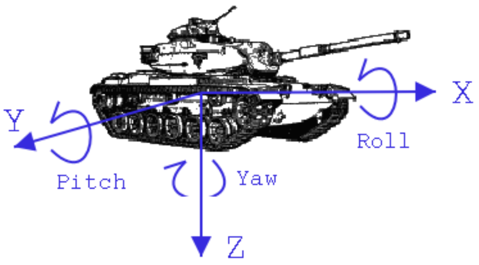

Figure A-1. Body reference frame attached to a rigid body. The x-axis points out the front of the vehicle.

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix} \qquad [A-2]$$

There are two different groups of euler rotations out of the possible twelve. The two groups are distinguished by their singularity locations. Typically aerospace and navigational applications use the 1-2-3 rotation where this singularity only effects fighter aircraft who dive or climb at such steep angles. Some space applications, such as orbits around the Earth, use the 3-1-3 rotation sequence.

Table A-2: Comparison of the two major types of rotations, sequential and first and third axes rotation

	Singularities in Pitch	Possible Rotation Sequence
Type I	+/-90	1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, 3-2-1
Type II	0/180	1-2-1, 1-3-1, 2-1-2, 2-3-2, 3-1-3, 3-2-3

In order to perform these operations on the rigid body, a rotation matrix (R) is used to find the new orientation of the spacecraft given the old orientation. Given below are the attitude matrices for rotations about the x, y, and z-axes.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \qquad [A-3]$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \qquad [A-4]$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [A-5]$$

Now, subsequent rotations about these primary axes can be accomplished by multiplying the matrices together. Thus, successive rotations about the z-axis, x-axis, and y-axis are given by:

$$R_{zyx}(\psi, \theta, \phi) = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \qquad \text{[A-6]}$$

Quaternions

Quaternions where invented by William Rowan Hamilton in 1843. Prior to his discovery, it was believed impossible that any algebra could violate the laws of commutativity for multiplication. His work introduced the idea of hyper-complex numbers. Here real numbers can be thought of as hyper-complex numbers with a rank of 1, ordinary complex numbers with a rank of 2, and quaternions with a rank of 4. Hamilton's crucial rule that made this possible:

$$i^2 = j^2 = k^2 = ijk = -1 \qquad \text{[A-7]}$$

Hamilton supposedly developed this rule while on his way to a party. When he realized what the solution was, he took out his pocket knife and carved the answer into a wooden bridge. This rule would forever change mathematics as was known at the time. Now mathematicians could look at algebra where communitivity did not work. This is where Gibbs and others developed algebra of vector spaces, and quickly eclipsed Hamilton's work until recently.

Quaternions, also known as Euler symmetric parameters, are more mathematically efficient ways to compute rotations of rigid and non-rigid body systems than traditional methods involving standard rotational matrices or Euler angles. Quaternions have the advantage of few trigonometric functions needed to compute attitude. Also, there exists a product rule for successive rotations that greatly simplifies the math, thus reducing processor computation time. Quaternions also hold the advantage of being able to interpolate between two quaternions (through a technique called spherical linear interpolation or slerp) without the danger of singularities, maintaining a constant velocity, and minimum distance travelled between points.

The major disadvantage of quaternions is the lack of intuitive physical meaning. Most people would understand where a point was if they were given [1 2 3]. However, few would comprehend where a point was if given the quaternion [1 2 3 4].

This section does not attempt to provide the extensive understanding needed to employ quaternions but rather a simple introduction.

Quaternion Algebra

The quaternion is composed of a scalar and a vector part. The scalar is a redundant element that prevents singularities from occurring since the four elements are all dependent upon each other.

$$q = \begin{bmatrix} q_x & q_y & q_z & q_r \end{bmatrix}^T = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} v, r \end{bmatrix} = ix + jy + kz + r \qquad \text{[A-8]}$$[1]

where the v stands for vector and the r is a scalar real. Given below is a table that will summarize the important mathematical operations of quaternions.

Table A-3: Quaternion Algebra Summary

Operation	Formula
Add / Subtract	$q \pm q' = \begin{bmatrix} q_x \pm q_x' & q_y \pm q_y' & q_z \pm q_z' & q_r \pm q_r' \end{bmatrix}^T$
Scalar Multiplication	$\alpha q = \begin{bmatrix} \alpha q_x & \alpha q_y & \alpha q_z & \alpha q_r \end{bmatrix}^T$

[1] The order of the quaternion elements is not standardized. I have chosen the element order that NASA uses which is the imaginary part first then the real. Some authors put the real first then the imaginary which is similar to complex numbers. Thus before you use any equations that utilize quaternions, make sure to understand how the author is arranging the elements.

Table A-3: Quaternion Algebra Summary

Operation	Formula
Norm	$\|q\| = \sqrt{q_x^2+q_y^2+q_z^2+q_r^2}$
Quaternion Multiplication	$q' \cdot q = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \cdot q'$ [a]
Conjugate	$q^* = \begin{bmatrix} -q_x & -q_y & -q_z & q_r \end{bmatrix}^T$
Inverse	$q^{-1} = \frac{q^*}{\|q\|}$

[a] Notice that the order of the quaternions in the matrix multiplication have been reversed.

Multiplication is an important operation for quaternions, thus we will elaborate on it a little.

$$pq = r = r_4 + ir_1 + jr_2 + kr_3 \qquad [A-9]$$

where

$$r_4 = p_4q_4 - p_1q_1 - p_2q_2 - p_3q_3 \qquad [A-10]$$

$$r_1 = p_4q_1 + p_1q_4 + p_2q_3 - p_3q_2 \qquad [A-11]$$

$$r_2 = p_4q_2 - p_1q_3 + p_2q_4 + q_3q_1 \qquad [A-12]$$

$$r_3 = p_4q_3 + p_1q_2 - p_2q_1 + p_3q_4 \qquad [A-13]$$

or rewritten in matrix form.

$$pq = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} p_4 & -p_3 & p_2 & p_1 \\ p_3 & p_4 & -p_1 & p_2 \\ -p_2 & p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \qquad [A-14]$$

The last part is the same representation present in the table of quaternion operations above. Notice how only the signs change when we reverse the order of the multiplication in the matrix-vector form of the quaternion multiplication. This is because of the cross product relationship with the vector part: $q \times p = -p \times q$.

Rotations of Rigid Bodies in Space.

Quaternions are able to represent a unique rotation in space. To perform a rotation (ϕ) of a rigid body about an arbitrary moving/fixed axis (e) in space, the quaternion representation of this operation is:

$$q_{1-3} = e \cdot \sin\left(\frac{\phi}{2}\right) \qquad q_4 = \cos\left(\frac{\phi}{2}\right) \tag{A-15}$$

$$e = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T \tag{A-16}$$

Notice that only one sine and one cosine function call is needed to calculate a quaternion, where euler would require three sine and three cosine function calls, one each for roll, pitch, and yaw. Since trigonometric function calls are computationally expensive, this is a great savings.

Rotation of a point in space with a quaternion is done as follows:

$$P^b = qP^aq* \tag{A-17}$$

$$P^a = \begin{bmatrix} x & y & z & 0 \end{bmatrix}^T \tag{A-18}$$

Since the quaternion has four elements, the point P must have a zero appended on to it in place of the real part of the quaternion. Thus the point P values constitute the imaginary (vector) part of the quaternion. This rotation can also be done similar to euler and fixed angle rotations by creating a rotation matrix from the quaternions. The attitude matrix (A) or rotation matrix (R) for this is

$$A(q) = R = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \qquad \text{[A-19]}$$

Successive rotations can be accomplished by multiplying the attitude matrices together.

Hopefully it can be seen that quaternions are better than other Euler operations to determine attitude. Quaternions lack singularities due to one redundant element. They also lack the computationally intensive trigonometric functions, and contain a simplified way to determine successive rotations about an arbitrary axis.

Attitude Errors in Quaternions

Since quaternions can represent attitudes and not just rotations, it is important to be able to calculate the difference between where you want to point and where you are pointing. For example, say you had to control where a satellite was to point and you need an error term to feed in to your control system.

$$q_E = q_S^{-1} q_T = \begin{bmatrix} q_{T4} & q_{T3} & -q_{T2} & q_{T1} \\ -q_{T3} & q_{T4} & q_{T1} & q_{T2} \\ q_{T2} & -q_{T1} & q_{T4} & q_{T3} \\ -q_{T1} & -q_{T2} & -q_{T3} & q_{T4} \end{bmatrix} \begin{bmatrix} -q_{S1} \\ -q_{S2} \\ -q_{S3} \\ p_{S4} \end{bmatrix} \qquad \text{[A-20]}$$

where q_E is the quaternion error, q_S is the attitude of the satellite, and q_T is the target attitude. Note that the quaternions are in reversed order in the matrix and the spacecraft quaternion has the imaginary part negated due to the inversion.

Summary of Quaternions

- Quaternions have singularities at angles of rotation of 180 degrees, but this is generally not a problem. Typically if an application encounters a rotation greater than 180 degrees, it is shorter to go in the other direction.

- The quaternion is a compact method of representing a unique rotation with only four elements with out the redundant information present in euler rotation's nine elements.

- Quaternions are difficult to visualize without extracting out the axis of rotation and rotation angle. Although the quaternion can be thought of as a sphere in 4D space, this does the majority of people little good, since they can not perceive a 4D surface.

- The use of quaternions allows you to compute the shortest distance between two different attitudes. If lerp is used, the resulting velocity between the two attitudes will not be constant but more bell shaped. If slerp is used, the resulting velocity will be constant. This process is difficult using euler angles, which contain many singularities that must be avoided and the solution is not guaranteed to be the shortest distance. Also the solutions that result from euler angles are not uniformly distributed along the surface of a sphere. Thus it again is not always possible to transition from one orientation to another using the shortest distance, because these empty areas must be skirted around to arrive at the desired location.

APPENDIX B
KALMAN FILTERING AND ESTIMATION

This Appendix will introduce the reader to kalman filtering. First some background material will be given, then a simple explanation of the fundamental concepts. Further explanation about kalman filtering can be obtained in Brown and Hwang [12].

Introduction

The kalman filter is an alternative way to calculate the minimum mean-squared error (MMSE) using state space. R. E. Kalman, a graduate research professor in the electrical engineering department of University of Florida, first developed the filter in 1960. Some of the advantages that the kalman filter has over other estimators are: computational efficient by recursively processing noisy data, real-time estimator, can be adapted to non-stationary signals, handle complicated time-variable multiple-input/output systems, vector model random processes under consideration.

The kalman filter estimates a process by using a form of feedback control. The filter estimates the process state at some point in time and then obtains feedback in the form of noisy measurements. The filter equations fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations.

Kalman Filter Theory

This section will give a brief derivation of the kalman filter which follows the derivation in Brown and Hwang [12]. The kalman filter assumes the random process to be estimated is of the form[1]:

$$\dot{x} = Fx + Bu + Gw \qquad \text{[B-1]}$$

where x is a state value, u is a control effort, w is white noise with known covariance. When measurements are taken of the process at discrete moments in time, they occur according to the following relationship:

$$z = Hx + Du + v \qquad \text{[B-2]}$$

where z is a noisy sample, D is the direct transmission of the input to the output, C is the ideal (noiseless) connection between the measurement and the state, and v is measurement error.

This process can be modelled discretely in the following form, assuming there are not control inputs u to the system.

$$x_{k+1} = \phi_k x_k + w_k \qquad \text{[B-3]}$$

$$z_k = H_k x_k + v_k \qquad \text{[B-4]}$$

The system error is defined as:

$$e_k^- = x_k - \hat{x}_k^- \qquad \text{[B-5]}$$

[1] Typically the state space system is represented as $\dot{x} = Ax + Bu$, and this form will be used when discussing the controller. However, during the discussion of the kalman filter the above notation ($\dot{x} = Fx + Bu$) will be used since this form is more common in kalman filter literature, and $\dot{x} = Ax + Bu$ will be used otherwise. There is no difference between the representations other than notation.

where $\hat{x}_k^-$ is the best estimate prior to receiving a measurement at time t_k. The error covariance matrix at this time is:

$$P_k^- = E[e_k^- e_k^{-T}] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \qquad \text{[B-6]}$$

where E[*] is the expected value. Now a linear blending of both the estimate and a measured value is taken.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \qquad \text{[B-7]}$$

where $\hat{x}_k$ is the new updated estimate, z is the measured value, and K is a weighting value that determines the amount of error between the measured value and the best estimate. This gain is referred to as the kalman gain which is capable of changing value over time. Now looking at the error covariance of this new updated estimate, we get the following.

$$P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \qquad \text{[B-8]}$$

$$P_k = E[[(x_k - \hat{x}_k^-) - K_k(Hx_k + v_k - H_k\hat{x}_k^-)][(x_k - \hat{x}_k)(-K_k(Hx_k + v_k - H_k\hat{x}_k^-))]^T] \text{[B-9]}$$

Now, after some algebra and realizing that $(x_k - \hat{x}_k^-)$ is the estimation error which is uncorrelated with the measurement error v, the following expression results for the error covariance.

$$P_k = (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k \qquad \text{[B-10]}$$

This is a general expression for updating the error covariance matrix, and it applies for any value of K. What is needed is a K that will minimize the individual terms along the main diagonal of P since these represent the estimation error variances for each state that is being estimated.

The gain that minimizes the mean-square estimation error is found by taking the derivative of the general expression above with respect to K, setting it equal to zero, and solving for K. The resulting gain is referred to as the kalman gain.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \qquad \text{[B-11]}$$

Implementing The Kalman Filter

The covariance matrices for both the system noise (w) and the measurement noise (v) are given by:

$$E[w_k w_k^T] = Q_k = G_k Q_k G_k^T \qquad \text{[B-12]}$$

$$E[v_k v_k^T] = R_k \qquad \text{[B-13]}$$

The kalman filter is executed in a loop which involves calculating propagation error (P) and updating the estimate from a measured value (z). The first step is to project ahead and calculate the state estimates and errors.

$$\hat{x}_{k+1}^- = \Phi \hat{x}_k + B_k u + w \qquad \text{[B-14]}$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^{\ T} + Q_k \qquad \text{[B-15]}$$

where u is the control effort and w is the process noise. Next the kalman gain is calculated using the predicted error (P_k) in the system and the covariance of the measurement noise (R_k). The predicted error is divided by itself plus the measurement error, which has the effect of reducing the kalman gain as the predicted error reduces.

$$K_k = P_k^- H_k^T (H_k P_k H_k^T + R_k)^{-1} \qquad \text{[B-16]}$$

The current estimate is then updated by the weighted (kalman gain) difference between the measured data and the noiseless model. Notice that if there is no noise or disturbances in the system and all dynamics are properly modeled, then the kalman gain is multiplied by zero.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^- - D_k u) \qquad [B-17]$$

From here on out, we will assume that there is no direct transmission of the control effort to the output of the system, thus $D_k = 0$.

Table B-1: Description of kalman filter variables.

Variable	Description
x_k	Vector containing the current state and parameters at time k.
$\hat{x}_k^-$	Vector containing the current state and parameter estimates at time k before updating the error between x_k and z_k.
$\hat{x}_{k+1}^-$	Vector containing the state and parameter estimates at the next time sample.
P_k	Matrix containing the current error covariance for the states and parameters.
P_k^-	Matrix containing the current error covariance for the estimated states and parameters.
z_k	Vector containing the current measured states which contain noise.
H H_k	Continuous and discrete output matrix which contains the noiseless connections between the states and the output of the system at time k.
R R_k	Continuous and discrete matrix containing the error covariance of the measurement noise for the system.
Q Q_k	Continuous and discrete matrix containing the error covariance of the process noise for the system.
F Φ_k	Continuous and discrete state transition matrix.
B B_k	Continuous and discrete gain matrix.
K_k	Kalman gain matrix which weight the amount of influence on the error between system model ($\hat{x}_k^-$) and the measured data (z_k).
w	Process noise.
v	Measurement noise.

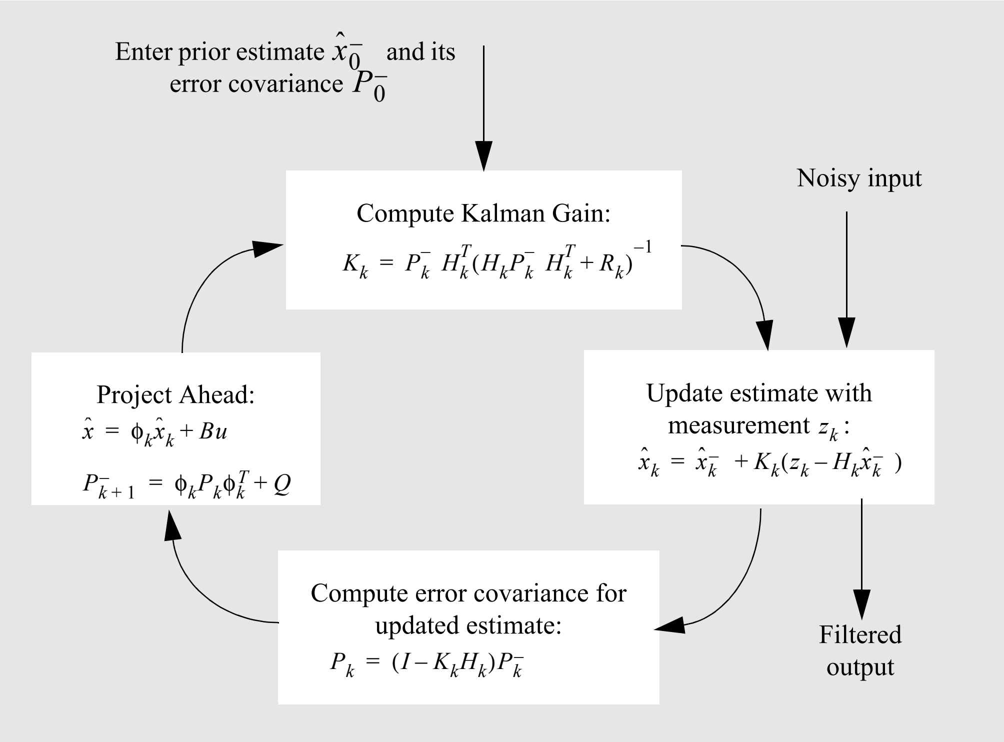
Enter prior estimate $\hat{x}_0^-$ and its error covariance P_0^-
Compute Kalman Gain:
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
Noisy input
Project Ahead:
$\hat{x} = \phi_k \hat{x}_k + Bu$
$P_{k+1}^- = \phi_k P_k \phi_k^T + Q$
Update estimate with measurement z_k:
$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-)$
Compute error covariance for updated estimate:
$P_k = (I - K_k H_k) P_k^-$
Filtered output

Figure B-1. Kalman filtering process at work, taking noisy input measurements and producing filtered output measurements. Note that the initial values of the state estimate ($\hat{x}_0$) and error covariance (P_0) are zero.

Extended Kalman Filter Design

The extended kalman filter is based on the standard kalman filter, but with a first-order Taylor approximation of both the state transition matrix and the observations equations about the current estimated state trajectory. This allows nonlinear equations to be used in the kalman filter, which are of the form:

$$\dot{x} = f(x, u, t) + Gw(t) \qquad [B\text{-}18]$$

$$y = h(x, t) + v(t) \qquad [B\text{-}19]$$

The non-linearity may enter into the equations either in the dynamics of the system or in the observation of the states. Thus the state transition matrix and observation matrix become:

$$A = \frac{\partial}{\partial x} f(x, u, t) \qquad [B\text{-}20]$$

$$H = \frac{\partial}{\partial x} h(u, t) \qquad [B\text{-}21]$$

where both of these terms are now the Jacobian matrix with respect to x.

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \qquad \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \qquad [B\text{-}22]$$

Augmented Kalman Filter

The kalman filter can be used for estimating unknown parameters. This can be accomplished by augmenting the state vector (i.e. adding the parameters to be estimated in to the state vector) and modifying both the state transition matrix and the observation matrix. The new state transition matrix will take the form:

$$\begin{bmatrix} A_{system} & A_{coupling} \\ 0 & A_{augment} \end{bmatrix} \qquad [B\text{-}23]$$

where A_{system} is the state transition matrix associated with the dynamics of the system to be controlled. $A_{coupling}$ is the coupling between the system and the parameters to be estimated. Finally, $A_{augment}$ is the model of the estimated parameters. There are several models that can be used for the augmented parameters.

Augmenting a kalman filter increases the numerical complexity of the filter, thus it is desirable to limit the parameters to be estimated in order achieve good run-time performance. The augmented kalman filter is limited by its observability to the number of parameters it can estimate. Thus it is important to check the observability of the kalman filter prior to attempting to estimate parameters otherwise the estimated parameters could diverge greatly from the real values and lead to instability.

Estimation Models

The models used for estimation are application specific, thus there is no general model that is used in every situation. Shown below are several different ways to model parameters that are being estimated.

Table B-2: Various models used in the kalman filter.

Type	Model	Description
Random Constant	$A = 0$	A parameter that is modelled as a random constant has a fixed, but random amplitude with the initial conditions as a Gaussian with zero mean and a constant variance.
Random Walk	$A = randn(var)$	A random walk estimation model varies from one integration step to another. Here the parameters are modelled as white noise with zero mean and the appropriate mean-squared values.
Markov Process	$A = e^{at}$	The Markov process model for parameter estimation allows estimates from one integration step to the next to be exponentially correlated.

Controllability and Observability of the Kalman Filter

The contollability and observability of a system is important since you cannot control what is not observable nor what is not controllable. If a system is determined to not be controllable, then no matter what type of controller you attempt to implement, you will not be able to control the system. For this section we will use the standard definition of a state space system, shown below.

$$\dot{x} = Ax + Bu \qquad \text{[B-24]}$$

where x is the state vector and u is the control effort.

Controllability

The controllability matrix for a linear system of size n is defined below. In order for a system to be fully controllable, the controllability matrix must be full rank.

$$M_c = \left[B\ AB\ A^2B\ \dots\ A^{n-1}B\right] \tag{B-25}$$

Observability

The observability matrix for a linear system defines the states that can be observed by a control system. Thus for a system to be fully observable, it must have full rank of the observability matrix.

$$M_o = \left[C^T\ A^TC^T\ A^{2T}C^T\ \dots\ A^{(n-1)T}C^T\right] \tag{B-26}$$

REFERENCES

[1] Sukkarieh, S., “Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles,” Ph.D. Thesis, University of Sydney, March 2000.

[2] Mandapat, R., “Development and Evaluation of Positioning Systems for Autonomous Vehicle Navigation,” MS Thesis, University of Florida, December 2001.

[3] Bennamoun, M., Boashash, B., Faruqi, F. and Dunbar, M., “The Development of an Intergrated GPS/INS/Sonar Navigation System for Autonomous Underwater Vehicle Navigation,” 1990 IEEE Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, pp. 256-261, June 5-6, 1990.

[4] Ohlmeyer, E., Pepitone, T., and Miller, B., “Assessment of Integrated GPS/INS for the EX-171 Extended Range Guided Munition,” July 2000, http://www.aerospacetechnology.com/download/paper3.pdf, 1/1/2002.

[5] Kariya, S. and Kaufman, P., “New Technology Transforms Tactics in Afghanistan,” IEEE Spectrum, Vol. 39, No. 4, April, 2002, pp. 30-32.

[6] Dana, P. H.,”GPS Overview,” May 2000, http://www.colorado.edu/geography/gcraft/notes/gps/gps.html, 1/1/2002.

[7] Mehaffey, C., “Garmin Users Manual,” April 2000, http://celia.mehaffey.com/dale/wgarmin.htm, 1/1/2002.

[8] Federal Aviation Administration, “Satellite Navigation Product Teams,” July 2001, http://gps.faa.gov, 1/1/2002.

[9] Jack Yeazel, “GPS WAAS Operation: Your Questions Answered,” July 2002, http://www.gpsinformation.net/waasgps.htm, 1/1/2002.

[10] Nebot, E., and Durrant-Whyte, H., “Initial Calibration and Alignment of Low-Cost Inertial Navigation for Land Vehicle Applications,” Journal of Robotic Systems, Vol. 16, No. 2, February, 1999, pp. 81-92.

[11] King, A. D., “Inertial Navigation -- Forty Years of Evolution,” GEC Review, Vol. 13, No. 3, 1998, pp. 140-149.

[12] Brown, Robert and Hwang, Patrick, *Introduction to Random Signals and Applied Kalman Filtering,* Third Ed., John Wiley and Sons, New York, 1997.

[13] Chatfield, A., *Fundamentals of High Accuracy Inertial Navigation*, AIAA, Inc., New York, 1997.

[14] Rogers, R. M., *Applied Mathematics in Integrated Navigation Systems*, Reston, VA: American Institute of Aeronautics and Astronautics, New York, 2000.

BIOGRAPHICAL SKETCH

Kevin J. Walchko was born on 1 Nov. 1972 in Sarasota, FL, and attended the University of Florida starting in the fall of 1991. After several years, a B.S. degree in mechanical engineering was earned in 1997. His college was paid for through the Montgomery G.I. Bill during his 7 years of service in the Florida Army National Guard as a Light Cavalry Scout. Kevin next received a master's degree in spring of 1999 for his work with NASA and satellite attitudes control. Most of the research Kevin conducted during this time was in controls and dynamics. Specifically, the research involved intelligent controls such as fuzzy logic and neural networks. Continuing his work with NASA, Kevin stayed at the University of Florida for a Ph.D in mechanical engineering which is expected in the spring of 2003.

During his time in graduate school, Kevin also fell in love and finally married Nina C. Massie on 20 October 2001. They are currently expecting the arrival of their first child sometime in the beginning of 2003.

Kevin has also conducted much research in robotics and autonomous vehicles. He has been a main member of the autonomous submarine team, Subjugator, for several years. Working closely with several members of the Machine Intelligence Laboratory, in the Electrical and Computer Engineering Department at the University of Florida, he concurrently completed a master's degree in electrical engineering during the summer of 2002 which involved inertial navigation.