2013

A Kalman Filter Based Attitude Heading Reference System Using a Low Cost Inertial Measurement Unit

Matthew Leccadito
Virginia Commonwealth University

An Attitude Heading Reference System using a Low Cost Inertial Measurement Unit

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University

by

Matthew T. Leccadito

Director: Dr. Robert H. Klenke

Associate Professor of Electrical and Computer Engineering

Virginia Commonwealth University

Richmond, Virginia

August 2013

Acknowlegements

I would like to thank my family for always supporting me with all of my struggles and keeping me focused on what is important in life. I would like to thank my advisor Dr. Klenke, who kept pushing me and believing in me when I was struggling and kept me motivated through the tough times. I would have never excelled the way that I have without his support, and I will never forget what he told me over and over "if the MIDG can do it, you can do it". I would also like to thank Tim Bakker who would sit down with me and struggle side by side through the many obstacles that I encountered, I have learned so much from him. I would like to thank Garrett Ward for the numerous programming questions he answered and many hours spent debugging, I would not have progressed as a programmer without the addition of his witty comments and vast knowledge of programming. I would like to thank Siva Patibandla, we have spent many nights together from beating our heads together working out a problem to seeking inspiration and trying to be the next Steve Jobs. Lastly I would like to thank Tom Carnes, enduring the rigors of VMI and VCU along side myself, as well as the enriching everyday experience of being my roommate. To all the other people in my life that have made it possible for me to accomplish what I have I would like to thank them, I will always cherish those moments spent in the extreme weather with my advisor and fellow graduate colleagues, from the freezing 6am mornings to the punishing sun beating down on us during UAV flights, these moments will stay with me forever.

Contents

List of Figures

List of Tables

Acronyms

UAV Unmanned Aerial Vehicle

MAV Micro Aerial Vehicle

IMU Inertial Measurement Unit

MARG Magnetic Angular Rate and Gravitational Sensor

AHRS Attitude and Heading Reference System

GPS Global Positioning System

INS Inertial Navigation System

DLPF Digital Low-pass Filter

Table of Symbols

$\mathbf{X}_k$ State Vector

$\mathbf{X}_{predicted}$ Predicted State Vector

$\mathbf{R}_k$ Noise Measurement Matrix

$\mathbf{Q}_k$ Error Covariance Matrix

$\mathbf{Q}_{predicted}$ Predicted Error Covariance Matrix

$\mathbf{z}_k$ Measurement Matrix

$\mathbf{y}_k$ Innovation Matrix

$\mathbf{W}_k$ Noise Model Covariance Matrix

$\mathbf{K}_k$ Kalman Gain

$\mathbf{T}_k$ Kalman State Transition Matrix

$\mathbf{A}_k$ Kalman Measurement Transition Matrix

$\mathbf{p}$ Rotational rate (gyro x-axis)

$\mathbf{q}$ Rotational rate (gyro y-axis)

$\mathbf{r}$ Rotational rate (gyro z-axis)

Abstract

This paper describes, the development of a sensor fusion algorithm-based Kalman filter architecture, in combination with a low cost Inertial Measurement Unit (IMU) for an Attitude Heading Reference System (AHRS). A low cost IMU takes advantage of the use of MEMS technology enabling cheap, compact, low grade sensors. The use of low cost IMUs is primarily targeted towards Unmanned Aerial Vehicle (UAV) applications due to the requirements for small package size, light weight, and low energy consumption. The high dynamics nature of smaller airframes, coupled with the typical vibration induced noise of UAVs require an efficient, reliable, and robust AHRS for vehicle control. To eliminate the singularities at $\pm 90°$ on the pitch and roll axes, and to keep the computational efficiency high, quaternions are used for state attitude representation.

Chapter 1

Introduction

The accurate measurement of orientation is an important role in many applications. With continuing research in the field of unmanned systems it is becoming a necessity to utilize smaller and less expensive components, which leads to the use of Micro-Electrical-Mechanical Systems (MEMS) based IMU providing less accurate readings. MEMS sensors in general can be effected by magnetic interference as well as having their own disadvantages which requires sensor fusion to combine the advantages of each sensor and compensate for the individual errors. Accelerometers are sensitive to vibrations and external forces including gravity; the gyroscope is prone to drift from integration over time, and the magnetometer is prone to corruption from ferrous materials in the environment.

The IMU is the platform of sensors which output measurements of the vehicles state, such as angular rates and accelerations. Typical low-cost IMUs consist of gyros that output angular rates about the three vehicle axes, accelerometers, which output acceleration (in terms of "g" units) along each of the three axes, and optionally magnetometers, which output the magnetic field flux density along each axis. An AHRS consists of an IMU in combination

with on-board computational ability to output vehicle attitude information in terms of yaw, pitch, and roll from the raw sensor data. The AHRS must provide an optimal sensor fusion algorithm to estimate the correct vehicle state (i.e orientation). In general the complexity of the state estimation algorithm increases as the quality of the sensors decrease. Therefore, the use of low-cost MEMS-based sensors requires more sophisticated state estimation algorithms in order to achieve the required AHRS performance.

There are many filters that can be used for the application of navigational systems however, the Kalman filter[13] was originally designed for this purpose by Rudolf E. Kalman in 1960. The Kalman filter is an algorithm which uses a series of measurements observed over time containing noise and other inaccuracies, to achieve an accurate output. The Kalman filter is capable of estimating the quaternion attitude orientation using a two-step process. The first step makes a prediction of the current vehicle state using the previous state information combined with the angular rates from the IMUs gyroscope. The next step is the measurement update, in which the Kalman gain is calculated using the attitude estimate calculated from the IMUs accelerometer readings. The two measurements are then combined to create sensor fusion, dynamically changing the Kalman gain depending on how accurate the measurement is.

The combination of accelerometers, gyroscopes, and magnetometers are used to accomplish the objective of creating an optimal sensor fusion algorithm to accurately compute orientation. Low-cost MEMS sensors cost approximately $10 - $15 for a tri-axis gyroscope, approximately $10 for a tri-axis accelerometer, and approximately $5-$10 for a tri-axis magnetometer. These sensors can be contained in individual packages, but are increasingly being integrated into a single small package, creating a 9 degree of freedom IMU. A few popular low-cost IMUs can be seen in table 1.

The term low-cost refers to a MEMS based IMU, low power, light weight, with no Global Positioning System (GPS), and costing less than $200. The design, implementation, and results of a low cost, robust, and efficient Kalman filter algorithm-based AHRS will be laid out in the following sections.

1.1 Problem Statement

Innovation in MEMS based sensors is revolutionizing inertial navigation in UAVs in size, weight, and energy consumption. With the increasing performance of technology and decreasing size, the capabilities of UAVs are growing at a fast rate. The miniature sensors and high performance micro controllers are increasing the potential of UAVs to decrease size and have the computational ability to estimate a better attitude solution for a lower cost. However, with the use of MEMS sensors comes the disadvantage of less accurate measurements.

In order to create a complete AHRS, it is necessary to incorporate an accurate, efficient, and robust sensor fusion algorithm to estimate the attitude of the UAV using the MEMS sensors. Sensor fusion is a technique used to combine the raw output data of the MEMS sensors to create a better estimate than the single sensors. In an ideal setting, a gyroscope can be used to determine the attitude solution of the platform, however with lower cost comes less precision and less accuracy. This creates a problem known as *drift*, where an error is accumulated on the integrated output. Therefore, the sensor needs a correction term obtained from another MEMS sensor known as the accelerometer. The accelerometer can be used to estimate the attitude directly from the sensors using measurements of linear accelerations. The accelerometer can be beneficial in an ideal setting, however it is common for a UAV to have a lot of vibrations and external forces which has an influence on the accelerometer. This

leads to a need for a sensor fusion algorithm to choose the optimal amount of each sensor to use in combination to estimate the attitude of the platform. The current commercial INS platforms available can range from hundreds to thousands of dollars to obtain an accurate attitude solution. These expensive platforms typically use more expensive MEMS based sensors or use Kalman filters algorithms that are very complex.

In current low-cost sensor fusion algorithms, GPS is a very important addition used for removing the centripetal/centrifugal forces. Centrifugal force is the fictitious force that is pulling outward when an object is in a coordinated turn. This external force corrupts the accelerometer measurement since the sensor is measuring linear accelerations with the addition of gravity. In a UAV application, the accelerometer is constantly being corrupted by these forces. In a typical scenario with no external forces present, the maximum force that any axis will experience on the accelerometer is 1g or 9.8m/s which is gravity. In a UAV application, the sensor will experience greater than 1g continuously throughout flight.

The Kalman filter can be enhanced by tightly coupling the AHRS with a GPS to create a complete INS solution. The GPS can be used to eliminate the centrifugal forces with the introduction of velocity measurements into the AHRS. The GPS can also be used to correct the heading calculated using the magnetometer. A flow diagram of the Kalman filter AHRS algorithm using GPS is shown in figure 1.1.
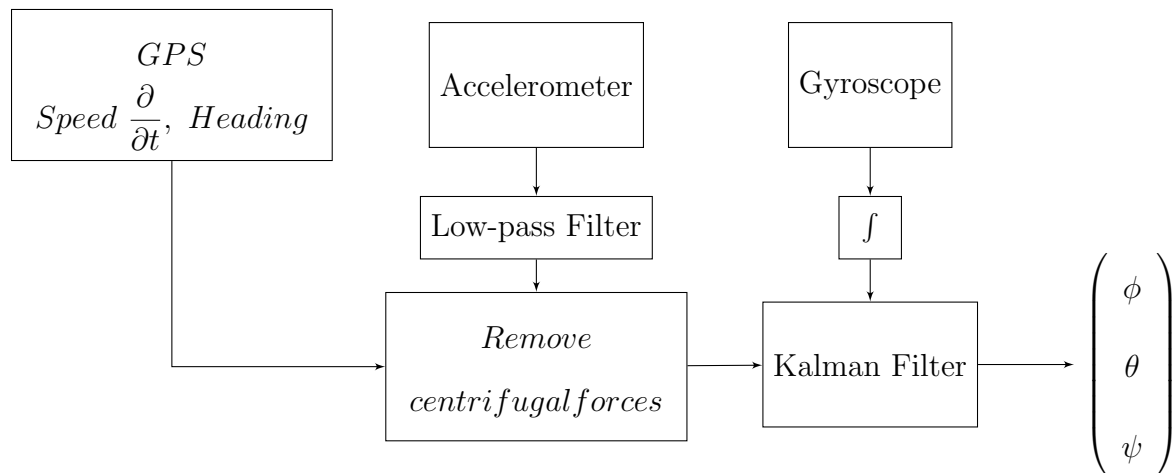
Figure 1.1: Kalman Filter coupled with GPS Flow Diagram

The extra measurement of velocity is needed to eliminate the effect of the external centrifu-
gal forces. The velocity rates can be used as a forward vector combined with the gyroscope
rotational rate to compensate for the centripetal forces pulling in the z-axis pointing down,
and the y-axis pointing out the right wing. Many other MEMS sensor based Kalman filter
estimation algorithms have been implemented to create a combined GPS/INS solution. Sys-
tems presented in [14, 15, 16, 17, 18] use this approach whereas the velocities and velocity
rates are required along the body axes as correction terms for the gravitational force com-
ponents and also used for magnetic declination. Without GPS, the AHRS must use a 3-axis
magnetometer to determine the heading angle. One method of doing so is by calculating
the geomagnetic vector in the z-axis using the magnetic field measurements. Using the ge-
omagnetic vector with the gravity vector which can be obtained by the accelerometers to
calculate attitude is known as a bi-vector method [19].

The current sensor fusion algorithms used in low cost AHRS require the use of a GPS.
The Kalman filter algorithm designed in this thesis was developed without the use of a GPS.
The sensors used in the calculation of the estimated attitude are the tri-axis accelerometer,

gyroscope, and magnetometer with no external corrections. The centrifugal forces present in the accelerometer measurements will be compensated for in the Kalman filter. The Kalman filter is typically very complex and requires high computational ability. The Kalman filter algorithm developed will use techniques that minimize complexity and be capable of running on a low-cost micro controller.

1.2 Thesis Overview

In the following chapters, the design, development, and testing of a low-cost Kalman filter based Attitude Heading Reference System will is presented. The first half of the thesis concentrates on the MEMS based sensors and commercially available low cost IMUs and low cost AHRS as well as the hardware platform used for real time flight testing. The second half concentrates on the development and testing of the Kalman filter algorithm.

Chapter 2 provides an introduction to the low-cost MEMS based sensors used to create a low-cost system. An explanation of how the accelerometer, gyroscope, and magnetometer operate. Also how the raw outputs from the sensors will be converted into useful data and used as inputs to the Kalman filter algorithm. Commercially available low-cost IMUs and AHRS systems will be explored. The chapter will end with a review of the literature.

Chapter 3 details the experimental setup of the hardware platform used for testing. An analysis of the MEMS based sensors is used to determine the optimal sampling rate and filtering of the raw sensor data to obtain useful information

Chapter 4 focuses on the design of the Kalman filter algorithm. The Kalman filter equations will be laid out and explained how the values of the covariance matrices are determined.

Techniques of how to improve the Kalman filter for the application will be proposed as well.

Chapter 5 provides the results from a simple table test to prove the accuracy of the Kalman filter in real time. A flight test is performed and the data is recorded in real time on-board the UAV. A reference high-cost Inertial Navigation System and a commercially available low-cost AHRS is used to compare against the proposed Kalman filter algorithm. Sensor data as well as attitude information is used to show the accuracy and robustness of the Kalman filter.

Chapter 6 concludes the work achieved as it relates to the research goal of developing a low-cost accurate, efficient, and robust AHRS. Future work to implement the Kalman filter on a low-cost flight control system in a UAV, and used to fly way points is proposed, and improvements of the accuracy, efficiency, and robustness is suggested.

Chapter 2

Background

This chapter provides background information pertaining to the design and development of a Kalman filter based AHRS using low-cost IMUs. This chapter begins with a description of the low-cost IMUs commercially available. It follows with a description of the commercially available Attitude Heading Reference Systems using low-cost IMUs. Euler angles and quaternions will be introduced to create a basis of understanding about the mathematics behind attitude representation. Low-cost sensors including the accelerometer, gyroscope, and magnetometer will be introduced with an explaination of how each sensor works and equations that convert raw sensor data into attitude information. The chapter concludes with a literature review of different AHRS techniques for use on low-cost IMUs used by open source and closed source companies.

2.1 Orientation

The attitude orientation of a UAV is a critical aspect in autonomous flight. In a low-cost AHRS, accuracy and low complexity are important in calculating the attitude of the UAV. There are various ways to represent attitude including: Euler angles, quaternions, direct cosine matrix, and rotational matrix.

2.1.1 Euler angles

Euler angles are the easiest to intuitively understand. Euler angles are comprised of three angles, roll, pitch and yaw. The roll angle ϕ is the axis pointed out the nose and rotates the plane, along the x-axis. The pitch angle θ is the axis pointing out the right wing, and represents the inclination and declination of the UAV. Lastly is the yaw angle ψ is the axis pointing out the bottom of the plane, it is used in the calculation of the heading of the UAV. This representation of Euler angles are shown in figure 2.1.
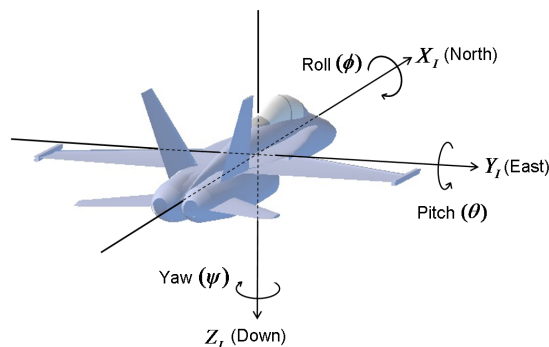
Figure 2.1: Euler Angles[1]

The order of which the angles are represented in a vector is not important, however the order of rotation is. The order of rotation used in the implemented Kalman filter algorithm

is: roll, pitch, and yaw. The reason for using this ordering is due to how the attitude is perceived. A UAV in autonomous flight typically does not incline or decline at a high rate, however it may roll at a high rate. To accurately understand the orientation of the UAV with respect to the Euler angles, it must be read from the roll first, then pitch, then yaw. The roll will have the range of $\pm 180°$, this allows the pitch to have the range of $\pm 90°$. The yaw is represented using the range of $\pm 180°$. The convention is used to keep the accurate orientation of the plane.

Euler angles have one disadvantage, which is known as Gimbal lock. Euler angles act as a gimballed system, where the three axes can be thought of as three separate gimbals connected together. Gimbal lock occurs when two axis are aligned, such as when the plane is pitched straight up, the pitch an the yaw axis are aligned, and as the roll gimbal is rotated, the pitch and the yaw angle are both effected the same, therefore losing orientation. The singularity problem can be fixed by adding another gimbal, or using an ad hoc approach to hard code values or use limits in the algorithm. The approach used in this thesis is to use a different representation adding another gimbal, known as quaternions. It is necessary to use an attitude representation with the least amount of elements because of the low-cost system requirements of low complexity. Whenever propagating attitude estimation through the Kalman filter, matrix multiplication increases in comlexity with more elements, especically in finding the inverse of a matrix.

2.1.2   Quaternions

As stated previously, to eliminate the singularities at $\pm 90°$ on the pitch and roll axes, and to keep the computational efficiency optimal, quaternions are used for state attitude rep-

resentation. To understand quaternions, one must first understand the complex number relationship. A complex number can represent a rotation in 2 dimensional coordinate frame with a real x-axis and imaginary axis y-axis or vise versa.

A quaternion is the same concept, except instead of one imaginary axis, there are three imaginary axes; in a sense its like combining three complex numbers into one. To go from a 2D interpretation to a 3D, four components are required; one real component qs and three imaginary qx qy qz which can also be represented as a vector $\tilde{\mathbf{v}}$. Equation (2.1) shows this relation with the complex number $\mathbf{x}$ having a real part $\mathbf{a}$ and imaginary part $\mathbf{b}$; and the quaternion $\mathbf{q}$ containing a real scalar part $\mathbf{s}$ and an imaginary vector component $\vec{\mathbf{v}}$.

$$
\begin{aligned}
Complex\ number: &\ \mathbf{x} = a + bi \\
Quaternion: &\ \mathbf{q} = s + \vec{\mathbf{v}}i \\
&\ \mathbf{q} = s + xi + yi + zi
\end{aligned}
\tag{2.1}
$$

A quaternion is represented as a single column matrix with four rows. The quaternion representation is shown in equation (2.2).

$$
\mathbf{q} = \begin{bmatrix} q_s & q_x & q_y & q_z \end{bmatrix}^T
\tag{2.2}
$$

To perform a quaternion operation, the real part will be used for the scaling of the vector, one imaginary component used for the axis of the rotation and the other two imaginary components used for the orientation of the platform. A pure rotation quaternion is of unit length. In order to keep the best accuracy due to a limited amount of precision, it is

necessary to normalize the quaternion to keep the error from accumulating. A quaternion is normalized the same way a vector is normalized. Equation (2.3) shows that the norm of a complex number and a quaternion are calculated similarly. The norm of the complex number $\mathbf{x}$ is calculated by taking the square root of the sum of the squared real and imaginary components. The norm of the quaternion can be found by taking the square root of the sum of the squared scalar part $\mathbf{s}$ and the imaginary vector $\vec{v}$.

$$
\begin{gathered}
Complex\ number: \\
\mathbf{x} = a + bi \\
\|x\| = \sqrt{a^2 + b^2} \\
Quaternion: \\
\mathbf{q} = s + \vec{v}i \\
\|q\| = \sqrt{s^2 + x^2 + y^2 + z^2}
\end{gathered}
\tag{2.3}
$$

Normalizing a quaternion is similar to normalizing a vector. The quaternion $\mathbf{q}$ is divided by the norm of the quaternion $\|\mathbf{q}\|$ shown in (2.4).

$$
\mathbf{q}_{norm} = \frac{q}{\|q\|}
\tag{2.4}
$$
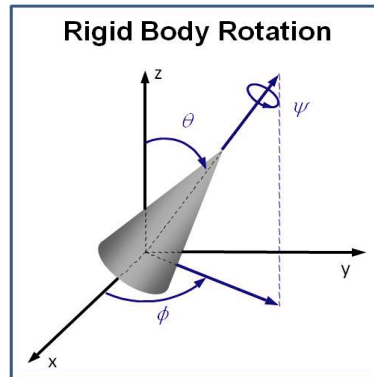
Figure 2.2: Quaternion Rotation[2]

To visualize a quaternion operation, think of Euler angles in the sense that there are three orthogonal angles each on a separate mutually exclusive axis, x,y and z as shown in figure 2.2. The rotation is not simultaneous, each rotation must happen sequentially, one axis at a time. In navigation terms, the plane must roll, then pitch, then yaw. In quaternion terms, the sequence can be thought of the same way as a rotating a vector laying on the x-axis, then rotating a vector on the y-axis, and finally rotating a vector on the z-axis.

2.1.3   Gyroscope

Gyroscopes have been used for many years in navigation. The typical function of a gyroscope involves a spinning object that is tilted perpendicular to the spin, where the angle of the reference surface can be measured. A more precise gyroscope involving a laser ring gyro contains two lasers being sent in a circular path and if a spin is observed, a phase shift can be detected since the speed of light is always constant. With the introduction of MEMS technology, gyroscopes can be produced in small, cheap packages. There are different types of gyroscope sensors including: piezoelectric which uses the Coriolis effect coupled with vibrations, tuning fork which measures the displacement of two objects, and wine glass

which measures the resonance of points on a hemisphere. Gyroscopes are not influenced by factors such as gravitation or magnetic fields. They can be used to orient technology to an absolute position in physical space, and can be used to detect acceleration, shock, and tilt.

Most MEMS gyroscopes are based on a tuning fork structure. The tuning fork uses the Coriolis effect to measure the angular rate. This is accomplished by two masses oscillating in opposite directions. When a rotation is applied, the masses are affected by the Coriolis force and the displacement is measured by a change in capacitance.
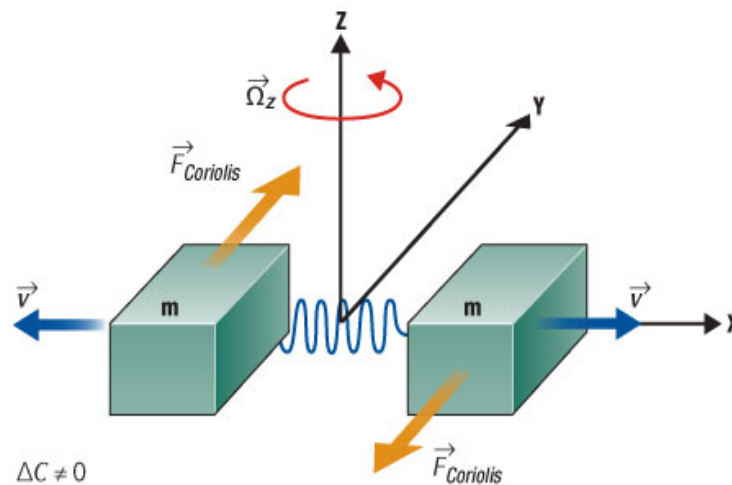
Figure 2.3: Tuning Fork Gyroscope Diagram (angular velocity applied)[3]

Three main specifications were evaluated when determining the best gyroscope: range, non-linearity, and noise-density. The range determines the maximum speed of which the sensor can measure, typically in (rad/sec) or (deg/sec). A UAV roll rate can change depending on the application, however for most applications involving a UAV such as surveillance, the UAV patrols way points and does not experience very high roll rate. The range of the sensor also effects the sensitivity of the sensor, as the range increases, the sensitivity decreases. The gyroscope outputs a voltage level proportional to the angular rate, and the non-linearity is

Manufacturer	Sensor	Price	Range $\pm$ (deg/sec)	Non-Linearity(%)	Noise Density (deg/sec/$\sqrt{Hz}$
Invensense	MPU-3000	$12.00	250 , 500, 1000, 2000	0.2	0.010
Invensense	MPU-3300	$12.00	225 , 450	0.2	0.005
Invensense	ITG-3200	$10.00	2000	0.2	0.030
StMicroelectronics	L3G4200D	$10.08	250 , 500 , 2000	0.2	0.030
StMicroelectronics	A3G4250D	$18.20	245	0.2	0.030
Invensense	MPU-6000 (gyro)	$15.00	250 , 500, 1000, 2000	0.2	0.005
Invensense	MPU-9150 (gyro)	$17.00	250 , 500, 1000, 2000	0.2	0.005

Table 2.1: Gyroscope (Triple-Axis)

the measure of how linear the voltage is to the actual angular rate. The last specification is the noise density, how much noise is present in the measured acceleration. The noise is typically Gaussian in nature. The actual noise in the sensor outputs can vary due to environmental noise which may include: temperature and voltage source of power. The gyroscope is usually not affected by mechanical accelerations (vibrations) that are acting on the sensor. A table of low-cost MEMS based gyroscopes are presented in table 2.1.

Angular velocity, as measured by an IMUs gyroscope, it the measurement of the rate of change of angular displacement. Most applications do not experience the same high dynamic nature of a UAV, and therefore can use the simple Euler angle representation. Since the gyroscope sensor outputs angular rates, the measurement can simply be integrated over a period of time (the sampling rate) and summed up to obtain the absolute angle θ. The integration of the angular rate is shown in equation (2.5). The angular rate is a measurement of the change in angle over time which is represented by $\omega(x)$. The rate is then integrated over a time period which in the Kalma filter will be determined by the frequency which the Kalman filter is operating at which is 50Hz, where Ts = 0.020.

$$\omega(x) = \frac{dx}{dt}$$

$$\theta = \int_0^t \omega(x)\,dx = \sum_0^t \omega(x) \cdot Ts$$

(2.5)

The angular rate is measured in radians per second which for Euler angles a simple integration may be performed. However, a quaternion cannot be integrated so easily. The measured angular rates need a way to be translated into a quaternion representation. The Omega matrix in equation (2.6) is used to converts the angular rates $\omega = \begin{bmatrix} p & q & r \end{bmatrix}$ obtained from the gyroscope, into quaternion rates so they can be transformed into a quaternion as the angular rate is integrated into Euler angles.

$$\Omega = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$

(2.6)

Once the Omega matrix is constructed, the quaternion rates can be used to calculate the attitude quaternion as shown in equation (2.7). In order to obtain a quaternion $\mathbf{q}_k$ rotated by the gyroscope, the previous attitude quaternion $\mathbf{q}_{k-1}$ is multiplied by the Omega matrix Ω, then half of the period Ts.

$$q_k = q_{k-1} * \Omega * \frac{1}{2} * Ts$$

(2.7)

2.1.4   Accelerometer

There are many different types of MEMS based accelerometers available, the typical low cost sensors are piezoresistive, capacitive sensing, and piezoelectric. More expensive MEMS sensors are laser and optical based.

To understand how an accelerometer works, the sensor can be thought of as a ball in a box[20]. If the accelerometer is still and there are no forces present, such as in space where no gravitational force exists, the sensor will measure 0g on all three axes; it is as if the ball is suspended in air, as shown in figure 2.4.
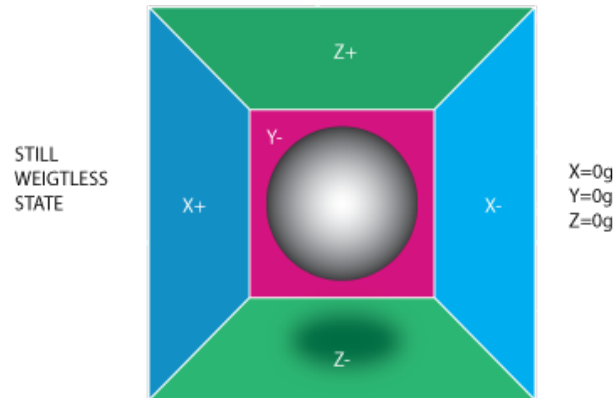
Figure 2.4: Accelerometer Diagram (no forces present)

If the sensor is suddenly moved to the left at an acceleration of 1g(9.81 m/s^2), the ball will hit the right wall, as shown in figure 2.5. Notice that the sensed acceleration is the opposite of the movement, this is known as the fictitious inertial force.
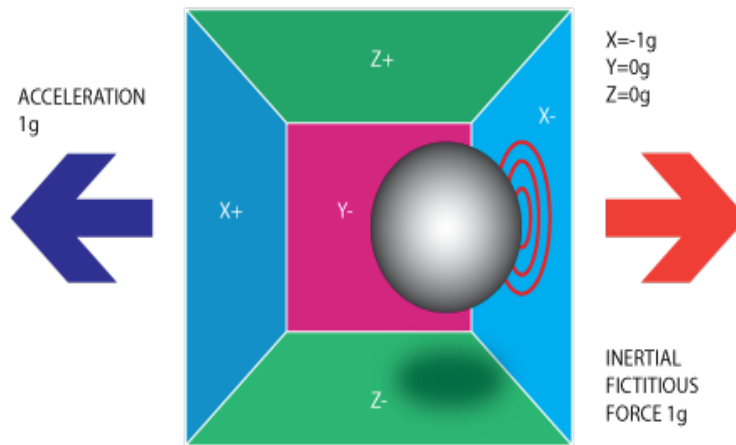
Figure 2.5: Accelerometer Diagram (-1g force in x-axis)

In a scenario where there is no external forces present the accelerometer would measure only the acceleration of the opposite direction of movement, however, on earth there is the external force of gravity pulling on the sensor. If the sensor is positioned on a flat surface with z-axis aligned as up and down, x-axis left and right, and y-axis forward and back, gravity will always be in the negative z direction, this is shown in figure 2.6. Gravity will always pull on the sensor providing a downward acceleration, unless the sensor is in a free fall or in a moving vehicle, such as a UAV in a coordinated turn with the external centrifugal forces being exerted. In a free fall scenario, an accelerometer experiences zero gravity, where the sensor is falling at $9.8m/s^2$, cancelling out the effect of gravity since the forces are the same in opposing directions.
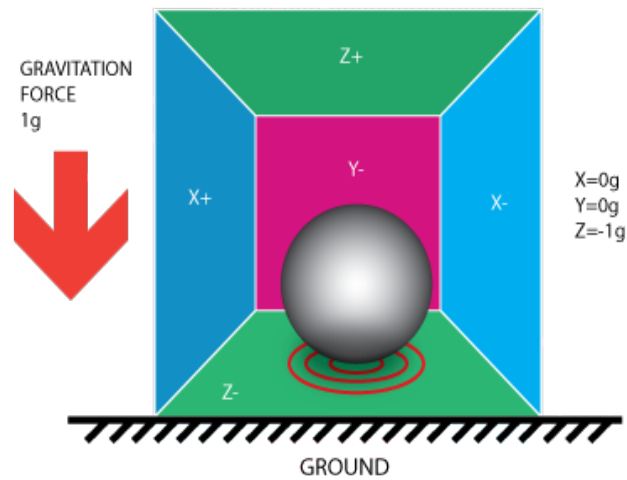
Figure 2.6: Accelerometer Diagram (-1g force due to gravity)

Many other applications involve rotations, where the measurement of the sensor will involve more than one axis of sensed acceleration. These rotations typically involve the sensing the acceleration on more than one axis, this can be visualized by the ball hitting two walls as the sensor is turned. The sensed acceleration is divided amongst the walls of which the ball is in contact with. The linear acceleration measurements are represented as vectors with a length and direction, g-force and axis respectively.
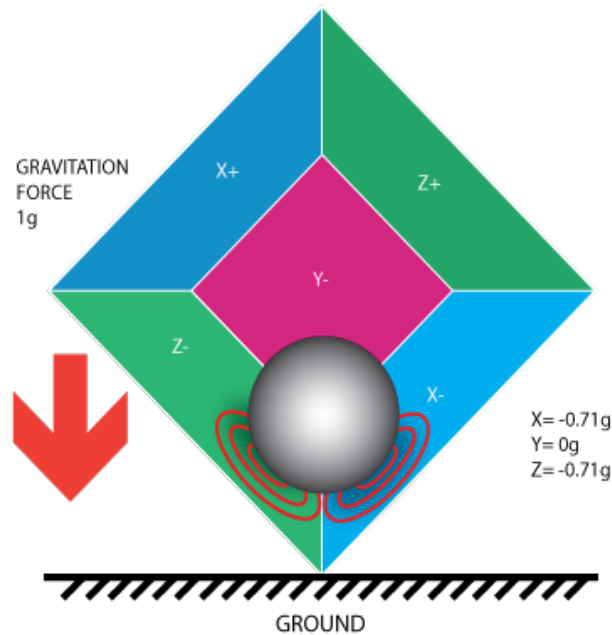
Figure 2.7: Accelerometer Diagram (-1g force due to gravity on a rotated sensor)

The accelerometer measures linear accelerations on all three axis. These accelerations are very sensitive to vibrations, however the Kalman filter needs as much information from the sensors as possible. To keep the most information about the accelerations from the sensors and for computational efficiency, a moving average filter was chosen to filter the data. The sampling rate of the sensor is 200Hz with an on-board low-pass filter set to 42Hz. The accelerometer measurements can be directly computed into position or Euler angles using trigonometry. The accelerometer will be used to estimate the roll ϕ and pitch θ in the measurement matrix z_k. The magnetometer will calculate the heading angle based on roll and pitch in the measurement matrix. In order to calculate the roll angle ϕ, the arctangent of the forward and down vectors are used. The pitch angle θ is calculated from the arcsin of the vector pointed to the right and the norm of the axes. These angles are used as the observation matrix z_k in the Kalman filter, shown in equation (2.8).

$$z_k = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan -a_y/a_z \\ \arcsin -a_x/\sqrt{a_x^2 + a_y^2 + a_z^2} \\ Magnetometer\, Heading \end{bmatrix} \tag{2.8}$$

MEMS based accelerometers have a variety of uses such as vibration sensing, motion sens-
ing, object tracking, gaming, etc. Due to the various types of applications an accelerometer
may be used, different specifications are needed. An accelerometer has different specifica-
tions that create a sensor more suitable for different applications. A table of low-cost MEMS
based accelerometers with their specifications are presented in table 2.2.

Manufacturer	Sensor	Price	Range $\pm$ (g)	Non-Linearity(%)	Noise Density ($\mu/\sqrt{Hz}$
Analog Devices	ADXL335	$4.46	3	0.3	150
StMicroelectronics	LSM303DLM	$5.27	2 , 4 , 8	0.4	218
Analog Devices	ADXL325	$3.73	5	0.2	250
StMicroelectronics	LIS344ALH	$5.89	2 , 6	0.5	50
Analog Devices	ADXL345	$4.56	2 , 4 , 8 , 16	0.5	290 x,y axes, 430 z-axis
Invensense	MPU-6000 (accel)	$15.00	2 , 4 , 8 , 16	0.5	400
Invensense	MPU-9150 (accel)	$17.00	2 , 4 , 8 , 16	0.5	400

Table 2.2: Accelerometer (Triple-Axis)

There were three main specifications that were evaluated when determining the best
accelerometer: range, non-linearity, and noise-density. The range determines how much
force may be present in the operation of the sensor. In the scenario of a UAV, the vehicle
will experience forces greater than 1g continuously during flight and depending on the type
of UAV can undergo high g-force turns. The range of the sensor also effects the sensitivity

of the sensor, as the range increases, the sensitivity decreases. Another specification is non-linearity, which can be described as the deviation of a constant sensitivity of the device; how perfectly constant the measurement is over a period of time. Normally the non-linearity of the accelerometer is low enough to be ignored. The last specification is the noise density, how much noise is present in the measured acceleration. The noise is typically Gaussian in nature. The actual noise in the sensor outputs can vary due to environmental noise which may include: temperature, voltage source of power, and mechanical accelerations (vibrations) that are acting on the sensor. A table of low-cost MEMS based accelerometers are presented in table 2.2.

2.1.5   Magnetometer

The magnetometer coupled with the accelerometer can effectively calculate a heading angle. The raw magnetometer measurements cannot be used to calculate the heading angle due to the decrease in sensitivity as elevation and bank angles increase, introducing error [21]. In order to obtain the correct heading a rotation must first be applied removing the bank angle, after which another rotation removes the pitch angle. Distortions of the earths magnetic fields has many influences on a magnetometer. These distortions can be classified into two categories: hard iron and soft iron.

Hard Iron

Hard iron distortion is a constant additive disturbance in the magnetic field of the magnetometer. This type of disturbance is created by ferrous materials around the sensor, for example if the body of the UAV has some metal that creates its own magnetic field and adds

to the sensors magnetic field and is in a constant position relative to the sensor, it is a hard iron distortion. Since hard iron is a constant distortion it is eliminated by a constant offset or bias. To eliminate the hard iron distortions the average of the maximum and minimum values must be obtained from the magnetometer. This requires that the sensor be rotated to the maximum rotation on each axis to determine the distance from the origin to the center of the circle. Hard iron distortion is shown in figure 2.8 where the ideal circle of the magnetic field is offset from the origin.
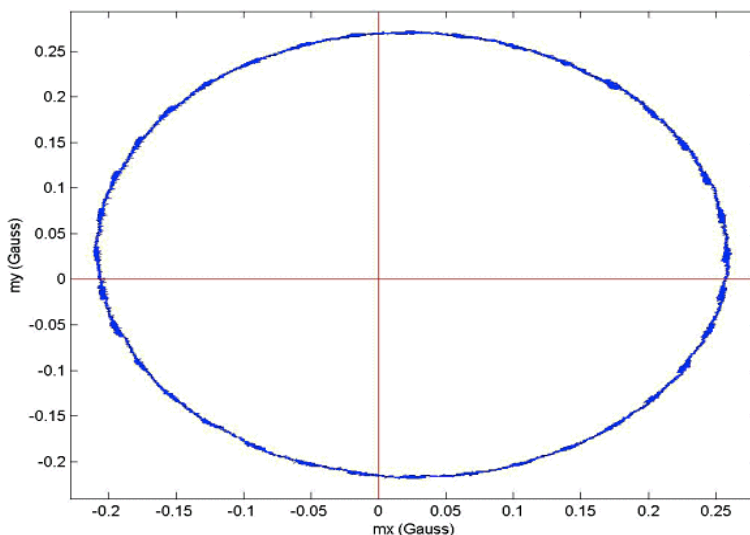
Figure 2.8: Hard Iron Distortion

These offsets can be measured once, and stored in memory. The measured values will be constant, such that the orientation of the sensor does not matter. The offsets for the x and y axis are calculated by taking the maximum value in each axis $\mathbf{x}_{max}$ and $\mathbf{y}_{max}$ and the minimum values in each axis, $\mathbf{x}_{min}$ and $\mathbf{y}_{min}$. Equation (2.9) shows the calculation of the hard iron compensation.

$$offset_x = \frac{(x_{max} - x_{min})}{2}$$
$$offset_y = \frac{(y_{max} - y_{min})}{2}$$

$$(2.9)$$

Soft Iron

The soft iron distortion is different than hard iron where the disturbance is the result of a material that distorts the magnetic field of the magnetometer, but does not necessarily generate its own magnetic field. The distortion of the soft iron effects on the sensor are determined by the orientation of the materials, and it is usually a perturbation of a circular magnetic field to an ellipse. The effects of hard iron distortion on the sensor can be shown in figure 2.9 where the circular magnetic field is distorted into an ellipse shape and the alignment is shifted from the x and y axis by the rotation of the sensor.
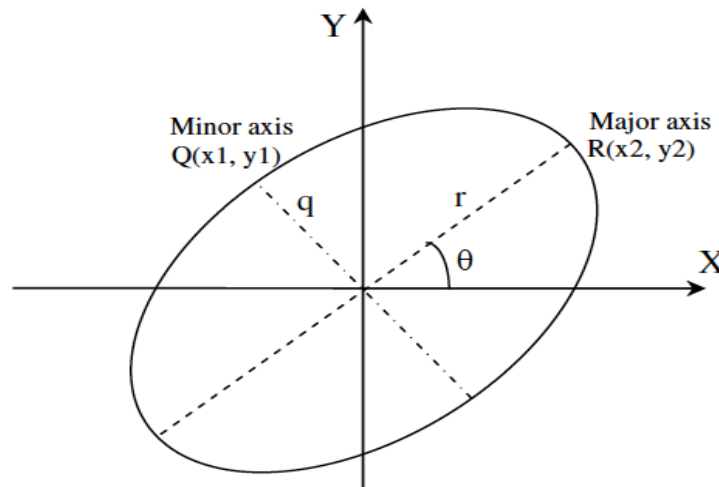
Figure 2.9: Soft Iron Distortion

Calculating the soft iron distortion is computationally more expensive than the hard iron elimination. There are two axes to take into account, the major and the minor axes. The

major axis is the axis that runs along the x axes. This will be used to find the angle θ by calculating the magnitude of the line segment. Essentially by computing r, it is the same as calculating the magnitude of each point on the ellipse and finding the maximum point.

$$r = \sqrt{x^2 + y^2} \tag{2.10}$$

Once θ is identified, the magnetometer x/y values can be rotated to the correct position so that the distorted x and y axes are aligned on the correct x and y axes. After alignment, it is possible to scale the ellipse to convert to a circular shape. The minor axes q is taken into account to calculate a scale factor that each magnetometer x value will be divided by.

$$scalefactor = \frac{q}{r} \tag{2.11}$$

2.2   Related Work

Innovation in MEMS sensor technology has advanced the research into robust AHRS algorithms. The Kalman filter algorithm has been used in navigation for many years for its optimal estimation properties. Proposed sensor fusion algorithms are investigated to help understand a low-cost approach.

2.2.1   Gauss Newton

A method used to solve nonlinear least squares problems is by finding the minimum of a function. Most techniques use the Taylor Series, however a Gauss Newton method can be

used to increase performance by substituting the update step with the gauss newton step in the standard Kalman filter. The Gauss Newton iteration algorithm can be utilized to find the optimal estimated attitude quaternion. It replaces the update step of the Kalman filter where the Kalman gain weights the predicted measurement or the actual measurement, and uses the estimated attitude quaternion from the Gauss Newton algorithm run in parallel with the Kalman filter. The approach reduces the computational load in the complementary filter [22] as well as the Kalman filter [23]. Two approaches may be taken, the first is to use the standard approach of using a seven state input vector which includes the angular rates and the attitude quaternion and have an output state vector coming directly from the three sensors, which includes highly nonlinear equations. The computational load due to this approach is very high. Therefore, a second approach is proposed using an external Gauss-Newton iteration algorithm to calculate an estimated attitude quaternion from the sensor measurements bypassing the update step of the Kalman filter. The Gauss-Newton Method was implemented in the works of [24], and compares the implementation using the performance of a complementary vs. Kalman filter. It is shown that the Kalman filter produces better results during high dynamics due to the dynamic gain which is inversely proportional to the variances and covariances of the observations, while the complementary filter relies on a static gain.

2.2.2   Complementary Filter

A less complex approach to sensor fusion is the complementary filter[25, 26]. The complementary filter is a type of estimation filter where the complement is the desired result. It has similar properties to a proportional integral derivative (PID) controller, with only the proportional and derivative gains. The gains act as a high and low pass filter on the input

data sources; the accelerometer and gyroscope. The accelerometer is very susceptible to vibrations and centripetal forces which needs to be filtered by a low pass filter to act as an enhanced moving average filter. The gyroscope is accurate in short term, but since the gyroscope outputs rates, due to precision in integration a drift is produced in the result. Therefore, a high pass filter is desired to allow the short-term gyroscope data through while eliminating the long-term drift induced gyroscope data. The complementary filter is based on the time constant to produce the desired gains on the accelerometer and gyroscope[27]. A high pass filter using the time constant formula can be found in equation (2.12).

$$\alpha = \frac{\tau}{\tau + dt} \tag{2.12}$$

The time constant τ is calculated from examining the gyroscope drift rate. The MPU6000 gyroscope has a drift rate of $2\frac{\deg}{sec}$. The complementary filter is a unity filter, meaning that the sum of the gains equals 1. Therefore, in order to calculate the gain α, the time constant should be chosen to be less than one second, to guarantee that the gyro will never drift more than a couple degrees per second. The sampling rate is very important in determining the gains of the complementary filter. For example, equation (2.13) shows if the algorithm is running at 50Hz, and the gyroscope time constant is chosen to be 0.80 seconds, α will be 0.975.

$$\alpha = \frac{\tau}{\tau + dt} = \frac{0.80}{0.80 + 0.020} = 0.975 \tag{2.13}$$

Once the gyroscope gain is calculated the accelerometer gain can be calculated by using the property of unity, where the sum of both gains is 1. Therefore, the accelerometer gain will be 1-0.975 = 0.0243. These gains can be tuned to create an optimal filter for the

application the IMU will be used for. The complementary filter can now be constructed using the two gains. Equation (2.14) shows the complementary filter angle calculation by summing the weighted portions of the gyroscope and accelerometer angles to create a more accurate combined attitude angle.

$$\theta_k = (0.975) * (\theta_{k-1} + gyro * dt) + (0.020) * (accel) \tag{2.14}$$

Low-cost implementations of sensor fusion algorithms incorporate low cost hardware which do not have a high amount of computational power. A reputable source of complementary filtering papers are the works referred to as the *Mahoney papers* incorporating the complementary filter. One of Mahoney's techniques is to eliminate centripetal forces on the accelerometer using GPS data to obtain the velocity vectors [28]. A group of rotations in mathematics can be represented by the special orthogonal matrix *S0(n)*; a special group of matrices closed under multiplication and the inverse operation[29]. This corresponds to a set of all transformations of the *n*-dimensional space. Use of the special orthogonal group to represent rotations is presented in [30]. Furthermore, a special orthogonal group may be used as a matrix to represent a rotation as well as a translation. However, if the rotation and translation matrices are mutually exclusive, this is known as the special euclidean group *SE(3)*. Another technique using the special euclidean group to represent the position and velocity of the body frame is presented in [31].

2.2.3 Direct Cosine Matrix

The direct cosine matrix (DCM) is similar to the rotational matrix, however it is a 3x3 matrix that contains the cosines of the 9 possible pairs of axes of two different Cartesian

coordinate systems. The DCM is typically used to translate from the body frame into the earth frame. A great introduction to the Direct Cosine Matrix theory is written by Premerlani, W. and Bizard, P. [32], providing information on how to use the DCM method integrated with IMUs. The DCM approach has gained much popularity for its advantage of linear measurement equations in Kalman filters[33, 34, 35, 36]. An analysis of various error sources using low-cost IMUs and optimizing fusion algorithms using the DCM method is presented in [37].

2.2.4 Other techniques

A gyro-free quaternion-based attitude determination algorithm is presented in [38] using the magnetic field vector provided by a tri-axis magnetometer, and the acceleration vector provided by a tri-axis accelerometer with error compensation using an acceleration vector provided by the GPS. Many faults were found in this algorithm; when the aircraft is in a coordinated turn or when the magnetic vector is aligned with the apparent gravity, the z-axis pointing down, the performance degrades.

During a high dynamics flight, orthogonality becomes an important aspect of a sensor fusion algorithm. The orthogonality refers to the alignment of axes in each sensor measurement, whereas if a sensor is measurement at a different time step, the two sensors will not be aligned on the same axes. If the measurements are used in a sensor fusion algorithm, it is assumed that the sensors are sampled during the same time step and obtain measurements with the same orientation of the platform at a particular instance in time. In most micro controllers, the sensors are sampled sequentially, this creates errors in orthogonality of the measurement axes since the different sensors are measured at different time steps and the orientation of the platform can be different when each individual sensor is sampled. To elim-

inate this problem, a technique of simultaneous sampling is used, where multiple analog to digital converters measure the different sensors in parallel. It is shown in[39] that the sensors are more capable of correcting errors during high dynamic maneuvers. A common problem with the use of Euler angles is the singularities at $\pm 90°$ which are because half angles are used in rotation to represent clockwise and counterclockwise rotation. The singularity problem can be introduced in the accelerometer measurement since the atan function is used. As the numerator and denominator approach zero, essentially random numbers are generated between $\pm 90°$ and the estimation of the angle becomes unstable. When rotating a quaternion the half angle is used as shown in (2.7), where the Omega matrix is created using the gyroscope measurements, and multiplied by the initial quaternion and delta in time, then divided by two. A technique known as the Modified Rodrigues Parameters (MRP) modifies this equation by replacing the division of 2 by 4, instead of half angles, quarter angles are calculated. This moves the singularities from $\pm 90°$, to $\pm 180°$. In [40] a stability analysis using the MRP vector in the kinematic equations is tested, resulting in the convergence of the errors in the system to zero are faster than not using the MRP method.

2.3   Low-Cost Inertial Measurement Units

2.3.1   Sparkfun Razor Stick 9DOF

The Sparkfun razor stick is a 9 degree of freedom low cost IMU board. The IMU is very small with dimensions of the board being 1.37"x0.42" with an input voltage of 3.3v. The tri-axis accelerometer included on the board is the Analog Devices ADXL345. The tri-axis gyroscope included on the board is the Invensense ITG-3200. Lastly, the tri-axis magnetometer included on the board is the Honeywell HMC5883L. The IMU communicates through the I2C interface and outputs the raw sensor data from the sensors. The IMU is available through Sparkfun Electronics and priced at $99.95.
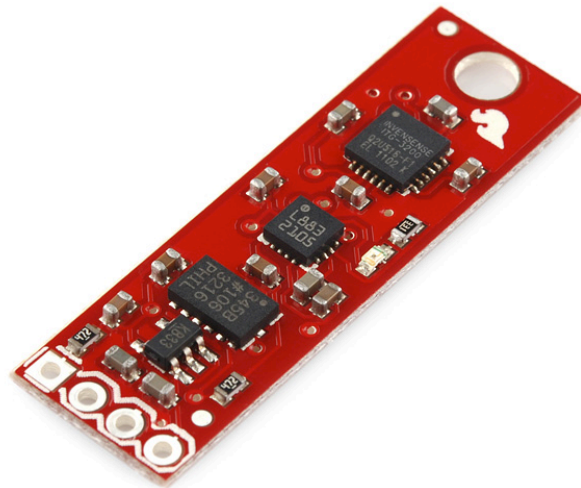
Figure 2.10: Sparkfun Razor Stick[4]

2.3.2   MPU-6000 Evaluation Board 9DOF

The MPU-6000 Evalution board was developed by Invensense with a double row header to breakout the outputs for for prototyping and testing. The evaluation board contains two sensors, the MPU-6000 which is a gyroscope and accelerometer, and the AK8975 magnetometer to create a 9 degree of freedom IMU. The sensors can be accessed through I2C at up to 400KHz or SPI at up to 20MHz. The MPU-6000 eliminates the gyroscope/accelerometer cross-axis misalignment by including both sensors in the same package. Included in the packaging with the two sensors is a Digital Motion Processor (DMP) which is capable of processing motion fusion algorithms. There is little documentation on the DMP included on the MPU-6000 creating difficulties in understanding how to initiate the use of it, and therefore was not explored further. The accelerometer and gyroscope sensors ouput at 200Hz, while the magnetometer outputs at 111Hz. The size specifications of the evaluation board are 38mm x 32mm with an input voltage of 5v. The MPU-6000 evaluation board is available for purchase at Invensense for $69.95

Figure 2.11: Invensense MPU-6000 Evaluation Board[5]

2.3.3   MPU-9150 Evaluation Board 9DOF

The MPU-9150 Evaluation board was developed by Invensense with a double row header to breakout the outputs for prototyping and testing. The evaluation board contains the MPU-9150 9 degree of freedom sensor package. The sensors communicate only through I2C at up to 400KHz. The MPU-9150 is similar to the MPU-6000 package which eliminates the cross-axis misalignment, however the benefit of the MPU-9150 is that the magnetometer is included in the same chip package, creating one package with all three sensors, saving space. Included in the packaging with the three sensors is a Digital Motion Processor (DMP) as well. The accelerometer and gyroscope sensors output at 200Hz, and the magnetometer outputs at 111Hz. The specifications of the evaluation board are 38mm x 32mm with an input voltage of 5v. The MPU-9150 evaluation board is available for purchase at Invensense for $79.95
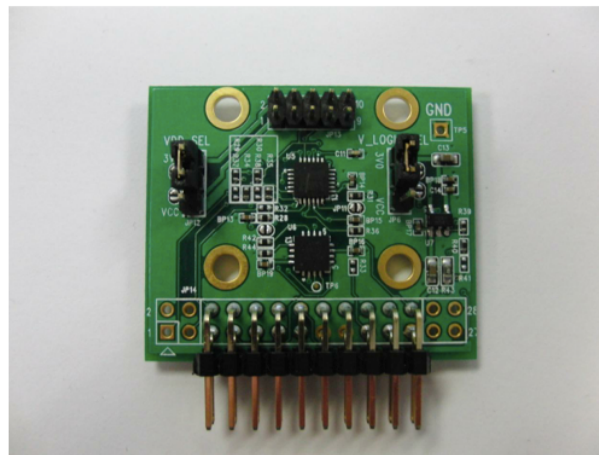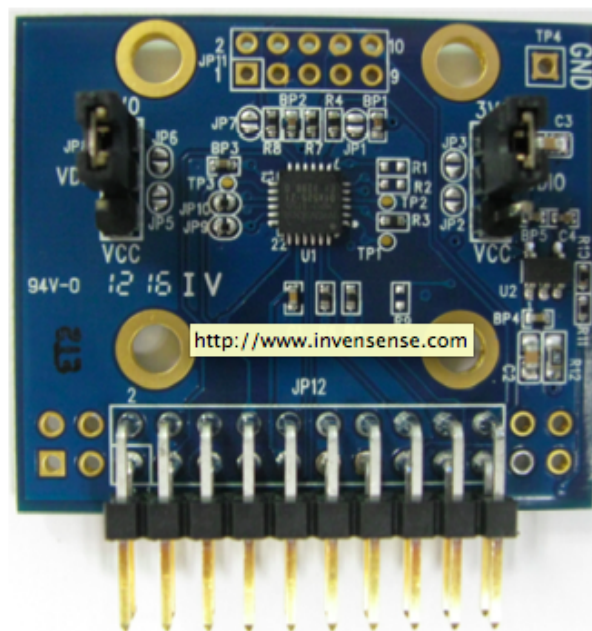

Figure 2.12: Invensense MPU-9150 Evaluation Board[6]

2.3.4   Pololu MiniIMU V2

The MiniIMU V2 is a low-cost IMU developed by Pololu Robotics and Electronics. Included in the IMU is the STMicroelectronics L3GD20, tri-axis gyroscope. Also, to create a 9 DOF IMU, the STMicroelectronics LSM303DLHC accelerometer and magnetometer are included in the same chip package. There is a voltage regulator and logic level shifting circuit included on the board creating a versatile sensor, with the ability to run on any voltage between 2.5V to 5.5V. The MiniIMU is compact with dimensons of 0.8"x0.5" and communicates only through I2C up to 400KHz, with a maximum supply current of 10mA. The MiniIMU V2 is available for purchase through Pololu Robotics and Electronics for $39.95.
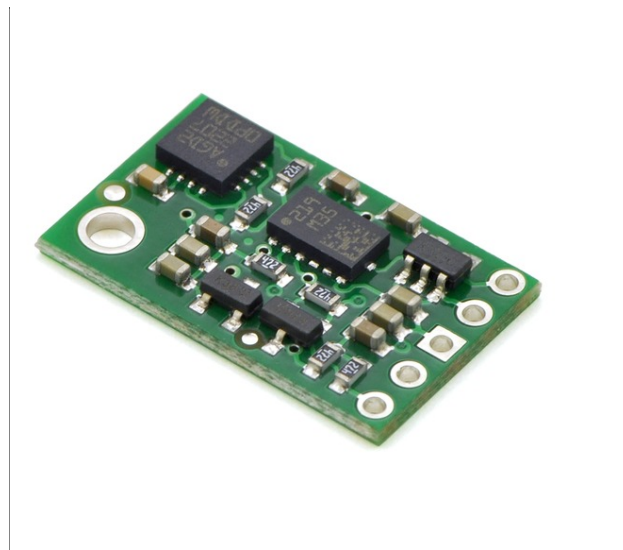
Figure 2.13: Pololu MiniIMU[7]

2.3.5   Aspirin IMU V2

The Aspirin IMU V2 is a low cost IMU developed by Transistion Robotics. Included on the Aspirin IMU is the Invensense MPU-6000 containing a tri-axis gyroscope and accelerometer, the Honeywell HMC5883 tri-axis magnetometer and a barometer capable of measuring altitude, creating a 10 degree of motion IMU. The dimensions are 0.53"x0.73", communicating through I2C up to 400KHz or SPI up to 20MHz, with an operating voltage of 3.3V. The Aspirin IMU is designed for and fits directly onto the Lisa and Booz autopilot platforms using the open source Paparazzi autopilot software. The Aspiring IMU is available for purchase through Transition Robotics for $125.00.
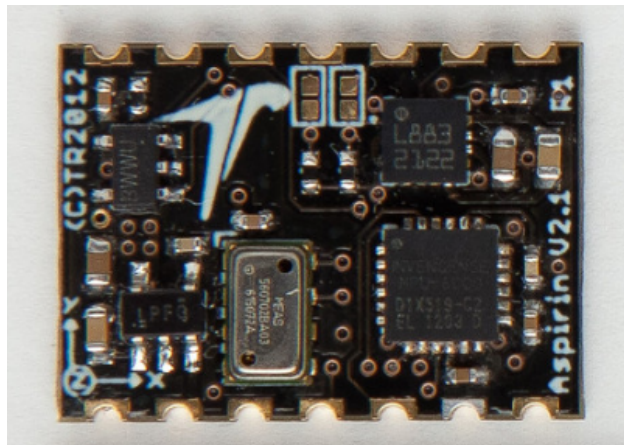
Figure 2.14: Transition Robotics, Inc. Aspirin IMU V2 10DOM[8]

2.4   Low Cost Attitude Heading Reference Systems

IMU boards sometimes have micro controllers on-board to create a more complete package that can be the basis of an AHRS. It may be necessary for the AHRS sensor fusion algorithm to have its own micro controller or processor to calculate the attitude to separate the computational load of the AHRS from the UAVs main control algorithms. The most popular low-cost AHRS packages integrated with MEMS sensors are examined in this section.

2.4.1   Sparkfun Razor IMU

The Sparkfun Razor IMU is one of the first low cost ARHS available. Through revisions, the sensors have been updated to the most current MEMS sensors including the Invensense ITG-3200 tri-axis gyroscope, Analog Devices ADXL345 tri-axis accelerometer, and the Honeywell HMC5883L magnetometer. The ATmega328 micro controller is included on-board with the sensors to create a complete AHRS capable of sensor fusion. The dimensions are 1.1"x1.6" communicating through Universal asynchronous receive and transmit (UART) consisting of a transmit and receive pin, and has an operating voltage of 3.3V.
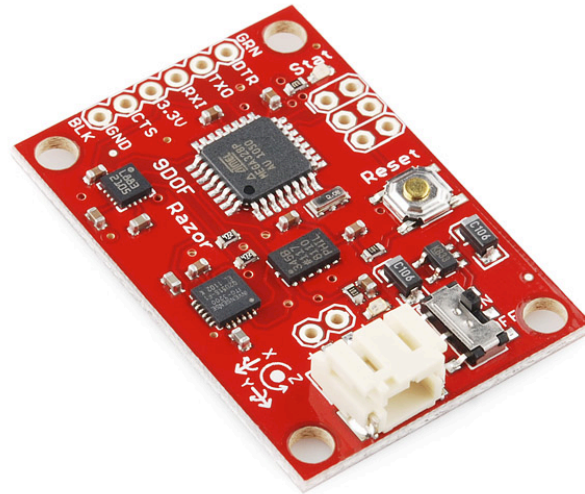
Figure 2.15: Sparkfun Razor IMU[3]

With the capability of on-board processing, the Sparkfun Razor IMU can output an attitude solution using a sensor fusion algorithm. However, the ATmega328 is operating at 8Mhz and only has 32KB of flash memory. Therefore, with little processing power, no hardware floating point unit, and little memory, the sensor fusion algorithm can not be very large, complex, or mathematically extensive. The sensor fusion algorithm used on-board the Sparkfun Razor IMU is a simple complementary filter algorithm using the direct cosine matrix representations discusssed in section 2.2.2. The Razor IMU is available through Sparkfun Electronics for $124.95.

2.4.2   Mongoose IMU

The Mongoose IMU is a low-cost AHRS developed by Ckdevice. The MEMS based sensors on-board include the Analog Devices ADXL345 tri-axis accelerometer, the Invensense ITG-3200 tri-axis gyroscope, the HMC5883L tri-axis magnetometer, and the BMP085 barometric pressure sensor for altitude sensing. The Mongoose also includes the ATmega328 micro controller for sensor fusion calculations on-board.
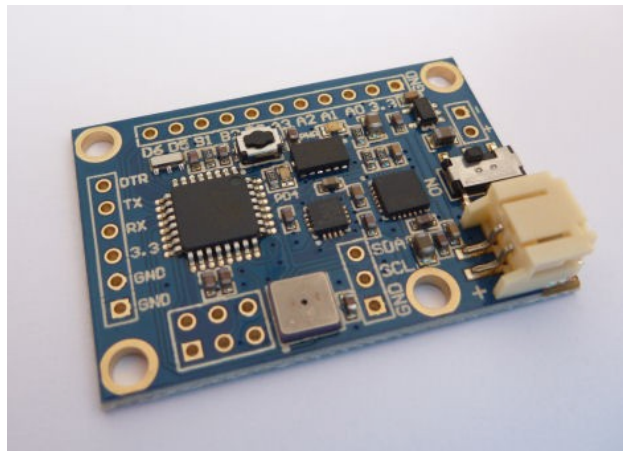
Figure 2.16: Mongoose IMU[9]

The Atmega328 is operating at 16Mhz, communicates through UART with the option of I2C for debugging purposes. The dimensions of the board is 1.6"x1.1" with an operating voltage of 3.3V. The sensor fusion algorithm uses a complementary filter technique, however it is not advised to use this IMU in an accelerating application, therefore cannot be used in a UAV. The Mongoose IMU is available for purchase from Ckdevices for $116.18.

2.4.3   ArduIMU V3

The ArduIMU is a low-cost AHRS developed by DIYDrones. The MEMs based sensors included are the Invensense MPU-6000 containing a tri-axis gyroscope and tri-axis accelerometer, and the Honeywell HMC-5883L magnetometer. The on-board processor used for sensor fusion calculation is the ATmega328 microprocessor. There is a GPS port for altitude and position estimation. The dimensions are 1.5"x1.0" and operates at 16Mhz. The operating voltage is 3.3v and communicates through UART. The sensor fusion uses the DCM based complementary filter and outputs the attitude at 8Hz.
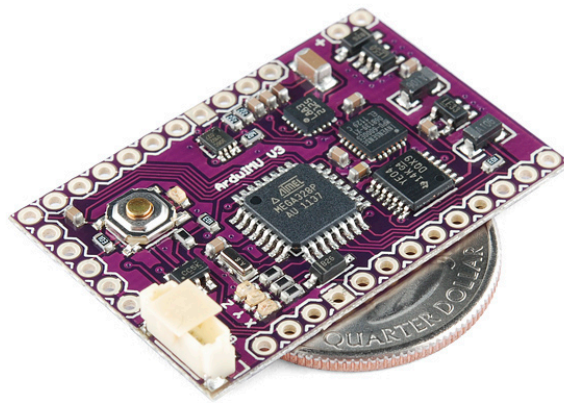
Figure 2.17: ArduIMU V3[10]

2.4.4 CHIMU

A closed-source low-cost AHRS system is the CHIMU developed by Ryan Mechatronics. The sensors on-board are not disclosed, however they are MEMS based. It is observed from the board that a tri-axis gyroscope and tri-axis accelerometer is used. It is noted in the user manual that GPS is needed for centripetal force compensation, and therefore will be needed for useful attitude estimation on a UAV. The dimensions of the CHIMU are 1"x0.83". The operational voltage is from 3.3V to 6.5V, with a supply current of 30mA. The communication protocol is SPI or UART operating at 200Hz. The CHIMU can be purchased from Ryan Mechatronics for $199.
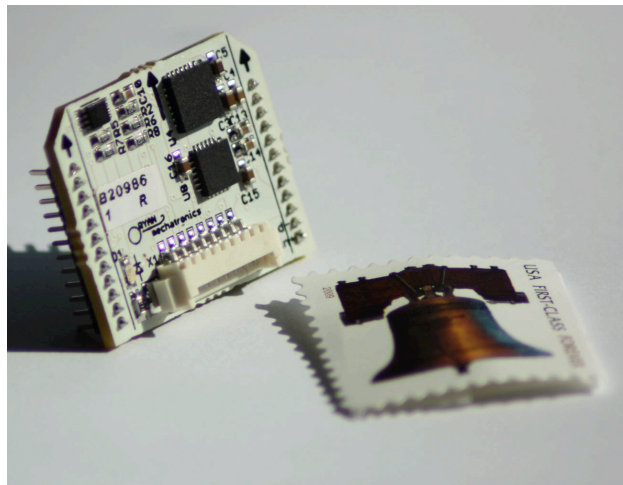
Figure 2.18: Ryan Mechatronics CHIMU[11]

Chapter 3

Experimental Setup

This section contains the hardware setup used to communicate with the low cost IMU sensors and the platform that the Kalman filter is implemented on. There were two Inertial Navigation Systems (INS) used to compare the results of the implemented Kalman filter to, the MIDGII and the UmarimLite. The MIDGII is an expensive INS known as the *golden standard* for AHRS and the UmarimLite is a commercially available low-cost AHRS that uses the open source Paparazzi autopilot software. The experimental setup was chosen such that the algorithm could be designed and tested without any of the low-cost considerations, this makes it possible to develop the algorithm with no constraints and then optimize the algorithm for low-cost systems.

3.1   Low-cost Considerations

The Kalman filter is a mathematically complex algorithm, most flight control systems demand an attitude estimation of 50Hz. Therefore, the low cost hardware platform must be

capable of running its sensor fusion algorithm at 50Hz. Low-cost systems have constraints on how fast the algorithm can run, and hardware and software limitations. A low-cost consideration is floating-point mathematics, such as using single precision opposed to double precision. Most low-cost micro controllers do not contain a floating point unit (FPU) and therefore it takes more clock cycles to execute floating-point math. Another consideration is the data type support, low-cost systems typically do not support hardware single or double precision floating-point math. Single precision mathematics refers to using data types of a *float*, which in the C/C++ programming language is a number with a maximum of 10^{-7} decimal places. A double precision is twice as much, however in a low-cost system it is more efficient to use single precision over double precision since the math is emulated in software and can consume a lot more clock cycles to perform a double precision operation vs. a single precision. Also, double precision variables take up twice as much space in memory vs. single precision where memory usage needs to be minimal. Low-cost micro controllers also typically do not contain external memory, therefore the code for the Kalman filter algorithm size needs to be small enough to fit on an inexpensive micro controller. Sampling rates must also be considered, in case the algorithm is used on the same micro controller that is used for other flight control algorithms to fly a UAV autonomously, sampling the sensors must not be too often, yet must be fast enough to accurately estimate the attitude.

3.2   Testing Platform

A testing platform must be capable of eliminating these constraints so that the algorithm can be designed and tested to prove accuracy first, then optimized for a low-cost system. The platform also must be able to simultaneously must have enough input/output ports for connectivity of the low-cost sensor, the MPU-6000 evaluation board, which uses the

communication protocol I2C. The reference INS and GPS unit use the UART communication protocol. There must also be external storage so that the data can be recorded on-board, and it would also be beneficial for connectivity of a wireless modem so that the data can be transmitted in real time to a ground station.

The Roboard-100[41] was chosen as the testing platform for the low-cost Kalman filter algorithm for its processing and I/O capabilities. It includes a 32 bit x86 CPU running at 1GHz, with 256MB of DRAM. On the CPU a version of Linux operation system is running on an external SD card creating an embedded Linux platform; this enables the use of multilevel threading capabilities as well as object oriented code written in C++. The Roboard has many input/output pins available including: two UART ports, a USB port, an Ethernet port, RS-232,I2C/SPI, and an 8GB micro SD card slot. With this capability, many different sensors could be connecting for testing. The Dimensions of the Roboard-100 are 96x56mm and has an operating voltage of 12V. The Roboard also has a hardware FPU.

Figure 3.1: Roboard-100

The Kalman filter will need to be tested on the table as well as in real time. In order to test the Kalman filter algorithm in real time a UAV platform is needed. To assure that the algorithm is robust, the gas powered MIG-27 UAV is chosen as the platform. As opposed to a glider, a gas powered engine produces vibrations, as well as making harder turns developing a higher g-force on the accelerometer. For example, in an electric powered glider, the typical forces exerted while maneuvering on the UAV do not exceed 1g, however on a gas powered UAV the forces exerted when maneuvering can exceed 3g's.

3.3   Data aquisition

The MPU-6000 Evaluation board was used as the sensor platform which contained the MPU-6000 with a 3-axis gyroscope and 3-axis accelerometer, as well as the AK8975 3-axis magnetometer. The MPU-6000 has an output rate of 1KHz with a digital low-pass filter(DLPF) implemented inside the chip. The DLPF causes a delay in ms from 0.98ms with a bandwidth of 256Hz to 18.6ms with a bandwidth of 5Hz. The more bandwidth used the smaller the delay, however the more external noise is included with the sample. To choose the optimal sampling frequency for the Kalman filter algorithm, an analysis of the accelerometer and gyroscope is performed. Two experiments were performed using the MPU-6000 sensor with the DLPF disabled, and sampled at 1000Hz. This allows for low and high frequency noise to be recorded in the signal, and will show the desired frequencies of the signal, and the frequencies that need to be filtered out. The first experiment is performed on the table with the sensor rotated 360 deg in each axis. This will be used as the reference of the bounds of the frequencies that the accelerometer and gyroscope data will lie in. The second experiment is performed on the MIG-27 gas powered UAV. The sensor was placed inside a foam box and secured inside the UAV under the same testing conditions that were

used during flight. The UAV was placed level on the ground and the throttle was slowly increased from no throttle to full throttle, providing similar noise conditions the sensors may undergo during flight. To perform an analysis on the data, the Fast Fourier Transform (FFT) is used to show the single-sided amplitude spectrum of the data recorded from the sensors. The results of the gyroscope data from this experiment can be seen in figure 3.2.

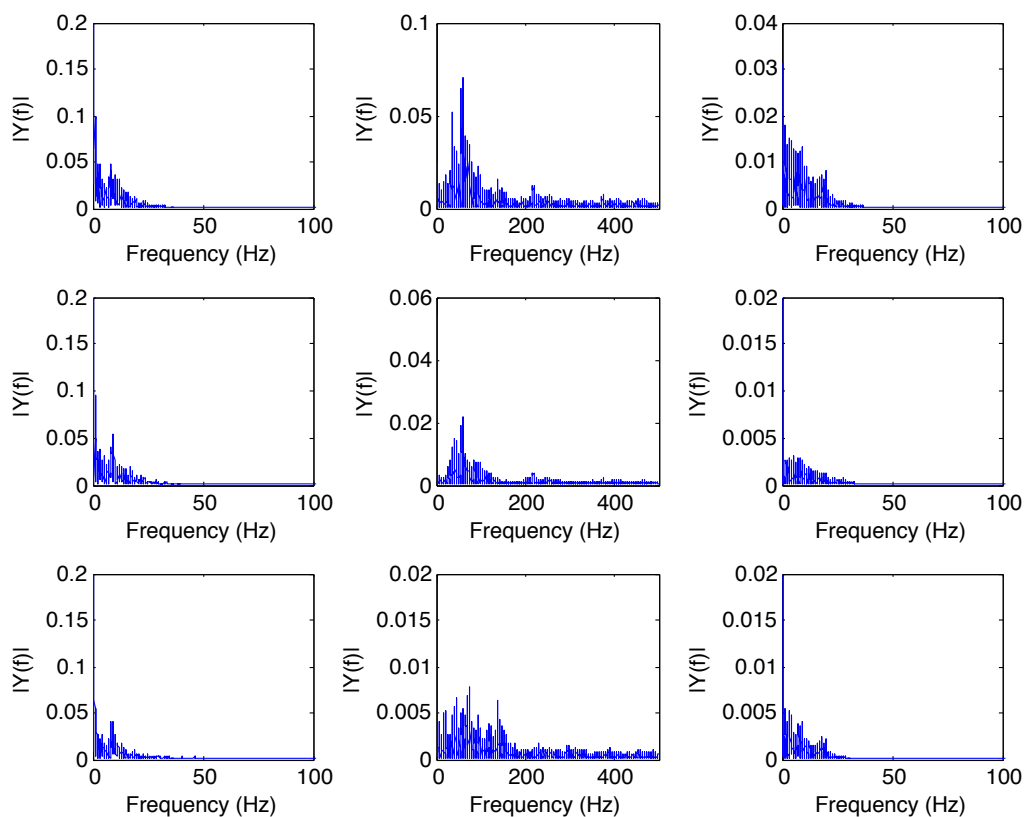**Figure 3.2: Sampling Data Experiment (left: $360\,\deg$ in each axis on table. center: Sensors in a foam box secured in UAV and the engine throttled up. right: 42Hz low-pass filter applied to the data from the UAV being throttled up.**

The results show the that the upper bound frequency of the data on the table is between 40Hz and 50Hz. The results from the data with the UAV being throttled up, shows a

large amount of the noise from the vibrations of the plane to be between 50Hz and 100Hz. Considering the analysis of the FFT, it can be seen that there is a large amount of noise included in the signal and the table data suggests that the important, desired data from the sampled signal is below 50Hz. Therefore, the MPU-6000 DLPF choices around the desired frequency cutoff are: 98Hz bandwidth with a 2.8ms delay, 42Hz bandwidth with a 4.8ms delay, or a 20Hz bandwidth with an 8.3ms delay. Choosing the 98Hz bandwidth would allow too much noise in the signal, and the 20Hz bandwidth would be eliminating important data from the signal. Therefore, the cutoff frequency of the DLPF will be set at 42Hz. In figure 3.2, the right column provides the data in the middle column, with a 42Hz cutoff low-pass filter applied using MATLAB. This verifies that the 200Hz sampling rate will be optimal for this application.

The sample rate will be 200Hz, which means that a moving average filter can be used to average the values together to create a smoother data sample. This is important because as the sample rate increases, the data we receive will get noisier, but the ability to average many samples will be used to try and minimize this. The sample rate of 200Hz is slow enough to reduce a lot of noise, but fast enough to be able to average around 4 values before being fed into the Kalman filter. The DLPF has a minimal delay included, but also gives the advantage of filtering the noise out of the sensor prior to sampling it, thus leaving more time for Kalman filter execution time. The magnetometer is accessed through the I2C bus on the MPU-6000 evaluation board, and has an output rate of 100Hz. The magnetometer will also use a moving average filter. Since the sensors are sampled faster than the Kalman filter is executing, the sensors can be sampled using interrupts on a low-cost sensor or they can be sampled using multi-threading on an embedded Linux platform. This gives the ability to sample while the Kalman filter is executing so that no data is missed as opposed to a sequential sequence where the sensors would be sampled, then the filter executes.

3.4   MidgII Inertial Navigation System

Microbotics, Inc. has developed an Inertial Navigation Solution known as the MIDGII. It is a tightly coupled GPS/INS solution using low-cost MEMS based sensors. The MIDII provides the "golden standard" reference for pitch, roll, and yaw. It is running an optimized Kalman filter using an internal microprocessor capable of calculating attitude, position, and velocity at 50Hz. The size specifications of the MIDGII are 1.5"x.810"x1.725", weighs 55 grams, and has a maximum power consumption of 1.2W.The MIDGII is priced at approximately $7000 [42]. The MIDGII will be used as the reference for the experiments that will be performed using the proposed Kalman filter, therefore the GPS antenna will not be connected creating an attitude calculation without the use of a GPS.

Figure 3.3: Microbotics MIDGII INS[12]

3.5 UmarimLite V2

The UmarimLite V2 is the hardware used for the Paparazzi open source Autopilot. The main processor running on the UmarimLite is the ARM based LPC2148, operating at 60Mhz. It consists of low cost sensors such as the 3-axis gyroscope from Invensense the ITG-3200, the 3-axis accelerometer from Analog Devices ADXL345. The size specifications of the Umarim-Lite are 2.1"x0.98" and weighs 8 grams. The combination of the UmarimLite and Paparazzi open source software, an IMU consisting of low cost sensors and a direct cosine matrix(DCM) complementary filter based AHRS form a low cost platform. The UmarimLite will be used as a low-cost comparison for the proposed Kalman filter algorithm. The PPZUAV UmarimLite hardware alone is priced at $299.95, however to use the paparazzi software the complete bundle priced at $699.95 must be bought which includes: UmarimLite main board, Paparazzi compatible GPS xbee 2.4gHz RF USB ground modem, xbee 2.4GHz RF airborne modem (wire antenna), and all of the cables to connect the modem and GPS to the Umarim.
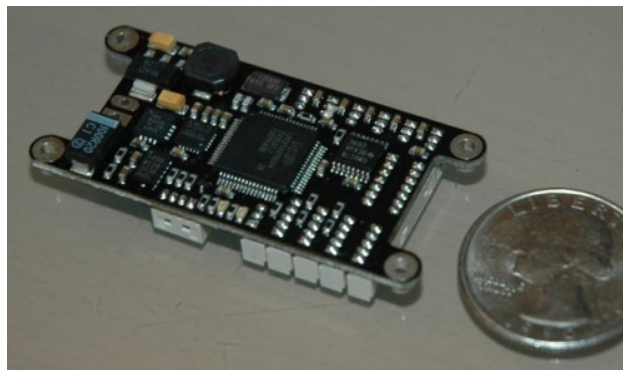
Figure 3.4: PPZUAV UmarimLite V2

Chapter 4

Modeling and Implementation

This chapter will explain how the Kalman filter is modeled and implemented

4.1 Kalman filter

The Kalman filter is an optimal estimator based on a prediction made from the previous input, and a current input. An example of the Kalman filter can be explained through a robot traveling through the desert [43]. The robot starts at a known location giving the robot a *priori* measurement. It begins to move forward, and estimates its position based on the amount of rotations of its wheels. The robot does not have very good wheels, and slips every so often. The next time the measurement is taken, its *predicted* position is not correct. However, a gps reading can be taken once in the time of three predictions are made. Therefore, a *measurement* can be taken and can be used to update the current estimation of its position. This process is repeated and the estimated output is improved by fusing the two sensors.

The previous example gives a basis of how the Kalman filter works. This approach can be used to create a sensor fusion algorithm to estimate the attitude using a 9 degree of freedom IMU consisting of a tri-axis gyroscope, accelerometer, and magnetometer. The Kalman filter and the complementary filter use the same principle of sensor fusion, however where the complementary filter has a static gain value, the Kalman filter has a dynamic gain value. The complementary filter is based of a time constant approach, to eliminate the gyro drift and correct it with a percentage of the accelerometer. The Kalman filter has a more complex approach which requires more information about the system. The Kalman filter uses information about the sensors to model the gyroscope drift and accelerometer noise dynamically changing the Kalman gain to converge to an optimal estimation. The Kalman filter is based on a two step process, predict and update. The prediction step is where the filter uses the gyroscope measurement and calculates an attitude estimation based on the gyroscope rates, and makes a prediction estimate of the error covariance matrix. The update step is where the Kalman gain is calculated based on the error covariance matrix, measurement transition matrix and noise covariance matrix. Once the Kalman gain is calculated, the accelerometer is incorporated to aid the gyroscope measurement which both values are multiplied by the Kalman gain to use a percentage of each measurement based on their noise characteristics. An overview of the Kalman filter is shown in figure 4.1.

Initialization

State Vector (X_0)
Error Covariance Matrix (Q_0)
Noise Measurement Matrix (R_k)
Noise Model Covariance Matrix (P_k)

1. Measurement Step

Conduct Observations (z_k)
(Accelerometer)
Obtain rates (p q r)
(Gyroscope)

2. Prediction Step

$$X_{predict} = T_k X_{k-1}$$

$$Q_{predict} = T_k Q_{k-1} T_k^T + P_k$$

3. Update Step

$$K_k = Q_{predict} A_k^T (A_k Q_{predict} A_k^T + R_k)^{-1}$$

$$X_k = X_{predict} + K_k(z_k - h_k)$$

$$Q_k = (Q_{predict} - K_k A_k Q_{predict})$$

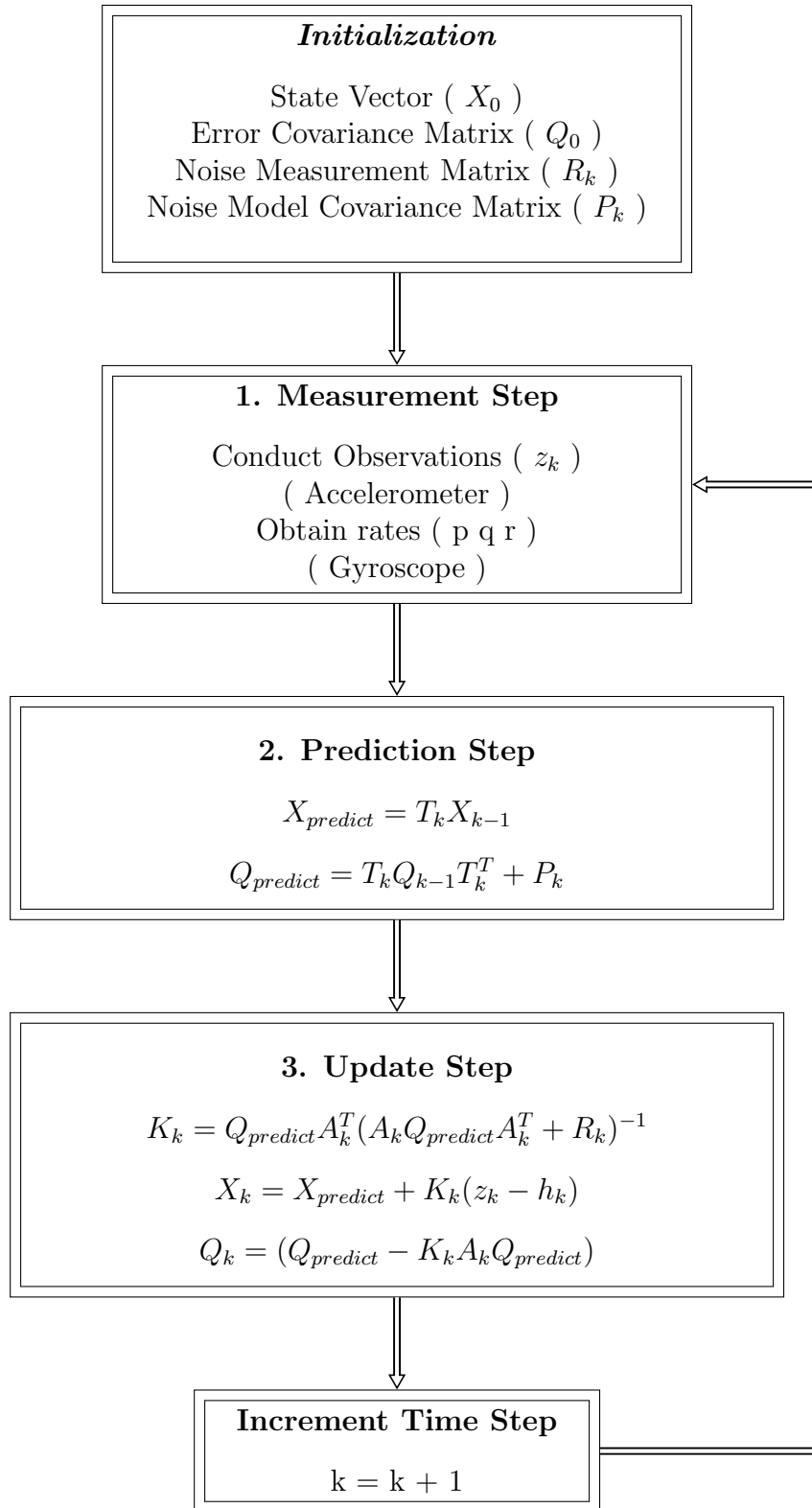**Increment Time Step**

k = k + 1

Figure 4.1: Kalman Filter Flow Diagram

4.1.1   State Vector

The state vector of the Kalman filter is the dynamic model of the system. In an AHRS, the dynamic model contains a representation of the orientation of the vehicle. The orientation can be represented in various ways as stated previously, however to eliminate the possibility of Gimbal lock and to keep complexity at a minimum, quaternions will be used. The state vector typically contains the variables that will be tracked and estimated. It contains seven elements, the state attitude quaternion and the gyroscope bias. Including the bias in the state vector gives the ability to track the bias and model the drift in the Kalman filter to minimize error and create a more optimal solution. The state vector is shown in equation (4.1).

$$X_k = \begin{bmatrix} q_s & q_x & q_y & q_z & b_x & b_y & b_z \end{bmatrix}^T \tag{4.1}$$

The state vector is initialized before the Kalman filter begins to operate. To initialize the state vector, the attitude must read a level estimate. A unit quaternion is used to initialize the attitude quaternion, since it is a level representation with no rotation of the IMU. The initialized state vector can be seen in equation (4.2).

$$\mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{4.2}$$

To initialize the gyroscope bias, an experiment was performed where the gyroscope data was recorded at 200Hz for ten trials of five seconds. The ten trials consists of the mean of each 1500 samples taken per trial. These values will be averaged and used to initialize the gyroscope bias in the state vector. These values will model the bias of the sensors

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10
Bias x	-0.01136	-0.01136	-0.01128	-0.01097	-0.01132	-0.01148	-0.01117	-0.01125	-0.01130	-0.01128
Bias y	0.03253	0.03220	0.03203	0.03179	0.03220	0.03243	0.03211	0.03223	0.03221	0.03216
Bias z	0.01083	0.01154	0.01157	0.01153	0.01162	0.01167	0.01150	0.01158	0.01158	0.01164

Table 4.1: Gyroscope Bias Measurements

since the readings are taken with no rotation involved. To initialize the gyroscope bias, and experiment of 10 trials were performed on a flat table with no rotations, the results are shown in table 4.1.

The gyroscope bias from the experiment were averaged together and used as the initial bias values in the state vector. The initialized state vector shown in equation (4.3) is composed of the unit quaternion and estimated gyroscope bias.

$$X_k = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.0113 & 0.0322 & 0.0115 \end{bmatrix}^T \qquad (4.3)$$

4.1.2 Covariance Matrices

Noise Measurement Covariance

The noise measurement covariance is the noise involved in the measurement matrix z_k. This matrix is the current measurement of the system, and is based on the accelerometer. The linear accelerations from the accelerometer are very sensitive to noise and vibrations, therefore it must be taken into account and modeled by this matrix. The initial values are chosen by calculating the estimated noise of accelerometers. The noise can be estimated in two steps. First by calculating the estimated mean value of a set of accelerometer data.

This is shown in equation (4.4), where N is the number of samples and $\vec{n}_k$ is a vector that contains the current data value for each axis x,y, and z resepectively.

$$E\{\vec{n}_k\} = \frac{1}{N}\sum_{i=1}^{N}\vec{n}_{k_i} \tag{4.4}$$

After the mean is calculated, it can be used to calculate the estimated covariance matrix. The estimated covariance matrix calculation is shown in equation (4.5)

$$cov\{\vec{n}_k\} = \frac{1}{N}\sum_{i=1}^{N}(n_{k_i} - E\{\vec{n}_k\})(n_{k_i} - E\{\vec{n}_k\})^T \tag{4.5}$$

Error Covariance Matrix

The 7x7 error covariance matrix contains the error in the measurements that make up the state vector. The important elements are along the diagonal which will contain the variance of the attitude quaternion and the gyroscope bias as shown in equation (4.6).

$$Q_k = \begin{bmatrix} \sigma_{q_s}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{q_x}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{q_y}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{q_z}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{b_x}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{b_y}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{b_z}^2 \end{bmatrix} \tag{4.6}$$

To initialize the error covariance matrix, it is necessary to have as accurate model of

the noise as possible. To accurately estimate the noise in the quaternion attitude, the accelerometer variance calculated from the sensor laying flat on a table with no rotations models the position of the sensor in an initialized state. The gyroscope variance will be calculated using equations (4.4) and (4.5) using data where the sensor is laying flat on a table with no rotations. Figure 4.2 shows the x, y, and z axis raw data from the accelerometer and gyroscope laying flat on a table with no rotations sampled at 200Hz for 50 seconds. The estimated variance values are calculated and shown in table 4.2
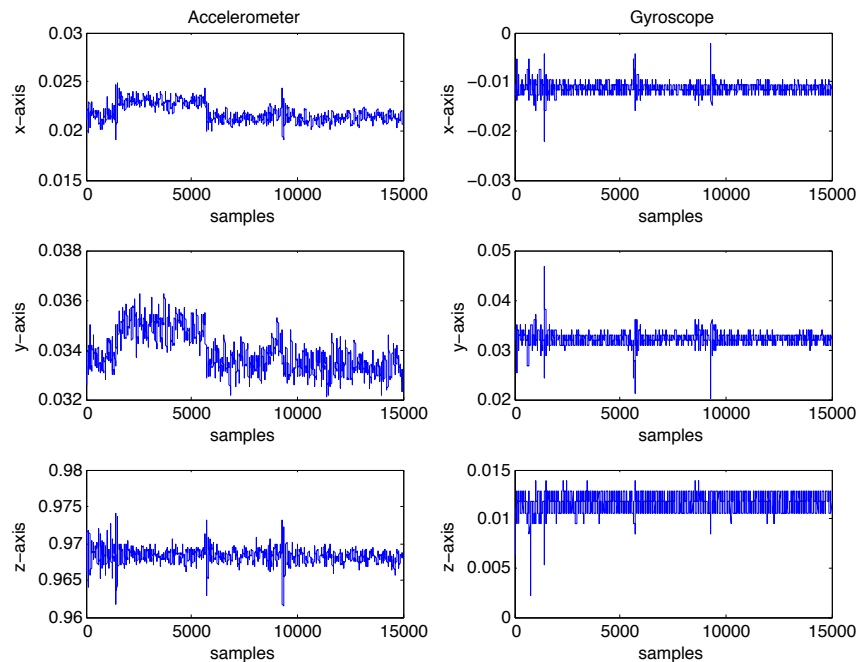

Figure 4.2: Accelerometer and Gyroscope on table with no rotations for 50 seconds

	X-Axis	Y-Axis	Z-Axis
Accelerometer	0.0006292	0.0006961	0.0006904
Gyroscope	0.0007760	0.0007458	0.0005423

Table 4.2: Accelerometer and Gyroscope bias measurement on the table with no rotations

The lower the values of the error covariance matrix, the more the filter will tend towards the gyroscope and less towards the accelerometer once the filter has stabilized. The error covariance matrix is updated each iteration and will be used to calculated the Kalman gain.

4.1.3   Noise Measurement Matrix

The noise measurement matrix can then be used to model the noise of the accelerometer data. The only important values are those along the diagnol of the matrix shown in equation (4.7).

$$
R_k = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \tag{4.7}
$$

The values of the noise measurement matrix are typically calculated from a previous data set. The data set used to calculate the covariance values comes from a real test flight manually piloted, putting the UAV through a series of maneuvers that the sensors may undergo during a flight. The data from the sensors was recorded at a rate of 200Hz, and the flight lasted 680 seconds (approximately 11 minutes). The data from the accelerometer and gyroscope are shown in figure 4.3.
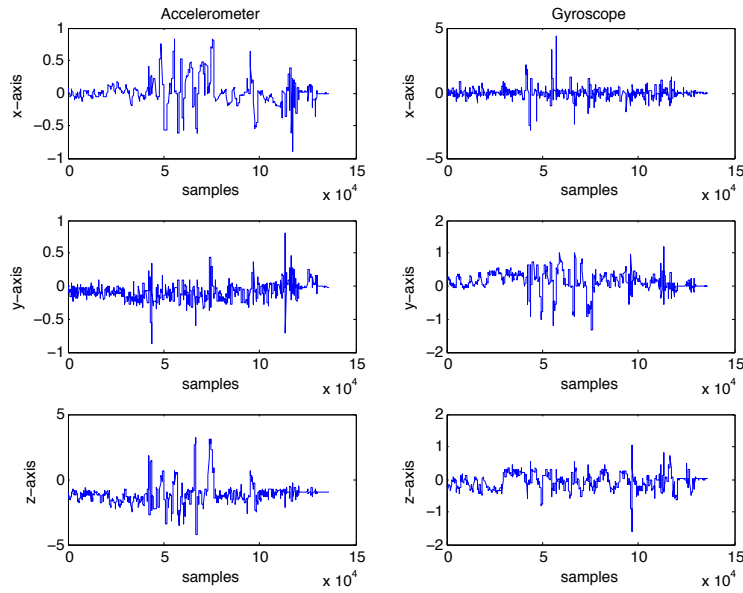
Figure 4.3: Accelerometer and Gyroscope data real time flight test for R matrix initialization

The z-axis is most influenced by the external forces, this can be seen in figure 4.3 where the z-axis is constantly greater than 1g and measured forces of over 4g's. Due to this high g force, the z-axis needs a high value so that the filter tends more towards the gyroscope so that the output is not corrupted by the centrifugal forces. Equations (4.4) and (4.5) were then used to calculate the covariance values. The values calculated from this data set represent the noise measurement covariance for the accelerometers and are shown in equation (4.8).

$$R_k = \begin{bmatrix} -0.005112 & 0 & 0 \\ 0 & 0.007291 & 0 \\ 0 & 0 & 0.16114 \end{bmatrix} \tag{4.8}$$

These values will change depending on the application. The noise measurement matrix depends how much the accelerometers will be weighted in the Kalman filter. The lower the values of the noise measurement matrix, the more the Kalman filter output will follow the accelerometer data, which will more likely record the vibration and external (centripetal/centrifugal) forces.

4.1.4   Noise Model Covariance Matrix

The noise model covariance matrix is used to model the noise in the system. Since the noise in the system is mostly introduced by the accelerometer, this noise is already modeled using the noise measurement matrix. However, At the low-cost level, single precision is an important factor. The quaternions used in the matrix calculations are of unit length and are between -1 and 1. This makes precision a high priority especially in a low-cost system which typically does not contain a floating point unit (FPU), which puts more stress on the main processor taking more processing time to execute floating point mathematics. This leads to a limiting factor in quaternion mathematics caused by precision due to truncation and rounding errors which are a result of minimizing the number of decimal places. The algorithm is coded using the C/C++ programming language, where the single precision limitation is 1^-7 decimal places. This value can also be used as an initialization value, since it represents the error in the attitude measurement as well as the calculations of the gyroscope integration. The initialized noise model covariance matrix is shown in equation (4.9).

$$
P_k =
\begin{bmatrix}
10^{-8} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10^{-8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10^{-8} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10^{-8} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10^{-8} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 10^{-8} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 10^{-8}
\end{bmatrix}
\tag{4.9}
$$

4.1.5 Prediction

The prediction step of the Kalman filter predicts the next state estimate using the current gyroscope rates combined with the previous state estimated by the Kalman filter in the previous time step. The gyroscope is used because it outputs a rate at which the platform is turning, and therefore requires an estimation using integration. This is known as a prediction since the output of the gyroscope is not an absolute attitude position.

The gyroscope has the disadvantage of accumulating a drift over time due to integration. The drift will be tracked in the state vector of the Kalman filter and continuously updated each iteration. This bias will be subtracted from the gyroscope angular rate measurements shown in equation (4.10) before they are put into the Kalman filter.

$$
\begin{pmatrix}
\omega_x = p - b_x \\
\omega_y = q - b_y \\
\omega_z = r - b_z
\end{pmatrix}
\tag{4.10}
$$

Using the rigid body rotation matrix Ω in equation (2.6), it is possible to create a transition matrix T_k that takes the gyroscope rates and transitions them to quaternion rates and integrating these rates to get the estimated quaternion. However, since equation (2.7) is nonlinear, it must be linearized by taking the first Taylor Series, also known as the Jacobian matrix, this is shown in equation (4.11). It is necessary to linearize the matrix to make it possible for propagation of the covariance matrix. The predicted state equation is expressed in terms of the previous state and gyroscope measurements which can be shown in equation (4.12).

$$X_k = X_{k-1} + f(X_n, W_n) = f'(X_n, W_n) \tag{4.11}$$

$$\left[\frac{\partial f}{\partial X}\right] = \begin{bmatrix} \frac{\partial f_0'}{\partial q_0} & \cdots & \frac{\partial f_0'}{\partial b_z} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_6'}{\partial q_0} & \cdots & \frac{\partial f_6'}{\partial b_z} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z & q_x & q_y & q_z \\ \omega_x & 0 & \omega_z & -\omega_y & -q_s & q_z & -q_y \\ \omega_y & -\omega_z & 0 & \omega_x & -q_z & -q_s & q_x \\ \omega_z & \omega_y & -\omega_x & 0 & q_y & -q_x & -q_s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.12}$$

The Jacobian enables the use of discrete equations and creates a 7x7 transition matrix for the predicted state. The transition matrix T_k can now be formulated using the quaternion theorem in equation (2.7). As shown in equation (4.13), the transition matrix is created by integrating, which is multiplying the Jacobian matrix $\left[\frac{\partial f}{X}\right]_k$ by half the delta time of the measurements and filling the diagonal with ones by adding an identity matrix I_{7x7}.

$$T_k = I_{7x7} + \frac{1}{2} \cdot \left[\frac{\partial f}{X} \right]_k \cdot \triangle t \qquad (4.13)$$

With the transition matrix constructed it is now possible to obtain the predicted state vector and the error covariance matrix using the Kalman filter equations. This is known as the *Time Update* because it is dependent on the time between measurements which is used to integrate the gyroscope measurement. The difference in measurements is referred to as $\triangle t = t_k - t_{k-1}$. The Kalman filter will be operating at 50Hz, therefore $\triangle t$ will be 0.020 seconds. The first equation in the Prediction Step is obtaining the estimated state vector using the transition matrix T_k and the previous state vector X_{k-1}, as shown in equation (4.14).

$$X_{predict} = T_k X_{k-1} \qquad (4.14)$$

The second equation is the estimated error covariance matrix using the previous error covariance matrix Q_{k-1}, the transition matrix T_k and the noise model covariance matrix P_k shown in equation (4.15).

$$Q_{predict} = T_k Q_{k-1} T_k^T + P_k \qquad (4.15)$$

4.1.6 Update

The Update step, also known as the measurement update is where the Kalman gain is calculated and the current state is estimated from the gyroscope and accelerometer measurements. This is the sensor fusion section and is optimized for low-cost with first order Taylor series approximation to reduce computational complexity. The first equation of the update step

is where the Kalman Gain is calculated. Since quaternions are being used for attitude, a translation matrix must be created so that the quaternions may be translated into Euler angles for the inverse operation. A quaternion contains four elements which can be very computationally expensive in an AHRS using a Kalman filter which has many matrix multiplications. Since this is a low-cost system, the process model or transition matrix is used to convert the four element quaternion to a three element Euler angle. The translation matrix also plays a role in the propagation of the error covariance matrix since the covariance matrix is expecting a four element quaternion. The translation matrix makes it possible to do 3x3 matrix operations using the Euler angles to compute the Kalman gain instead of the quaternions to decrease the complexity of a 4x4 inverse matrix operation. Even though Euler angles are used, Gimbal lock is not present, since the quaternions are used as the representation to rotate the angles each iteration, the Euler angles are only representing the quaternion attitude rotation for the calculation of the Kalman Gain The Euler angle to quaternion equations are shown in equation (4.16), where roll,pitch, and yaw is represented by ϕ, θ, ψ respectively, is calculated from the quaternion q = $[q_s \ q_x \ q_y \ q_z]$.

$$\phi_{nonlinear} = \arctan(2(q_y q_z + q_s q_x)/(1 - 2(q_x^2 + q_y^2)))$$
$$\theta_{nonlinear} = -\arcsin(2(q_x q_z - q_s q_y)) \tag{4.16}$$
$$\psi_{nonlinear} = \arctan(2(q_x q_y + q_s q_z)/(1 - 2(q_y^2 + q_z^2)))$$

However, these conversions are nonlinear in nature, therefore they must be linearized to be used in the Kalman filter. The first order Taylor Series will be used for this matrix to obtain the best possible estimate of the Euler angles from the quaternion attitude. The measurement transition matrix is formulated in equation (4.17).

$$A_k = \begin{bmatrix} \dfrac{\partial \phi}{\partial q_s} & \dfrac{\partial \phi}{\partial q_x} & \dfrac{\partial \phi}{\partial q_y} & \dfrac{\partial \phi}{\partial q_z} \\[2ex] \dfrac{\partial \theta}{\partial q_s} & \dfrac{\partial \theta}{\partial q_x} & \dfrac{\partial \theta}{\partial q_y} & \dfrac{\partial \theta}{\partial q_z} \\[2ex] \dfrac{\partial \psi}{\partial q_s} & \dfrac{\partial \psi}{\partial q_x} & \dfrac{\partial \psi}{\partial q_y} & \dfrac{\partial \psi}{\partial q_z} \end{bmatrix} \tag{4.17}$$

The error covariance matrix Q_k is a 7x7 matrix and the state vector X_k is a 7x1 matrix, therefore a 3x3 zero matrix must be added onto the end of the A_k matrix creating a 7x3 matrix, to allow for the gyro bias propagation to take place. The Kalman gain is computed in equation (4.18) using the error covariance matrix Q_k, the measurement transition matrix A_k, and the noise measurement matrix R_k.

$$K_k = Q_{predict} A_k^T (A_k Q_{predict} A_k^T + R_k)^{-1} \tag{4.18}$$

Once the Kalman gain is computed, then the new estimate can be updated. The updated estimate is based on the difference from the estimated gyroscope quaternion attitude and the actual measurement z_k taken from the accelerometer measurements. This matrix is known as the innovation matrix y_k, and shown in equation (4.19).

$$z_k = [\phi_{acc}, \theta_{acc}, \psi_{acc}]^T$$
$$y_k = z_k - A_k X_{predict} \tag{4.19}$$

At first, equation (4.19) was used to calculate the updated estimated state vector, however the measurement transition matrix A_k is calculated using the Taylor Series and is not a very accurate estimation. A technique to improve Kalman filter performance is to replace the

linearization of the state prediction with the nonlinear equations converting Euler angles to quaternions as shown in equation (4.20).

$$h_k = [\phi_{nonlinear}, \theta_{nonlinear}, \psi_{nonlinear}]^T$$
$$y_k = z_k - h_k$$

$$(4.20)$$

The state vector update is shown in equation (4.21) which adds the innovation matrix y_k multiplied by the Kalman gain K_k

$$X_k = X_{predict} + K_k \cdot y_k \qquad (4.21)$$

The error covariance is updated using the predicted error covariance matrix $Q_{predicted}$, the Kalman gain K_k, and the measurement transition matrix A_k.

$$Q_k = (Q_{predict} - K_k A_k Q_{predict}) \qquad (4.22)$$

Chapter 5

Testing and Results

The first experiment conducted was a simple test on the table. This is a basic test under ideal conditions to test the accuracy of the algorithm. There were no vibrations introduced or rigorous movements. The test involved a platform where both the low-cost IMU and the MIDGII were strapped down to a wooden platform and rotated by hand pitching approximately 45°, up and down, and then rolled approximately 45°. The sensor data from the table test can be seen in figure 5.1. It is important to note that the accelerometer is not induced by vibrations, and that the g-force exerted on the z-axis of the accelerometer does not exceed 1g. This replicates the ideal inputs from the sensors, which can be compared to the flight sensor data.

The results in figure 5.2, show good estimation of the pitch and the roll, however there is a constant offset of error, which can be reduced by calibration. The simple table test shows that the pitch, roll, and yaw are estimated appropriately in real time.
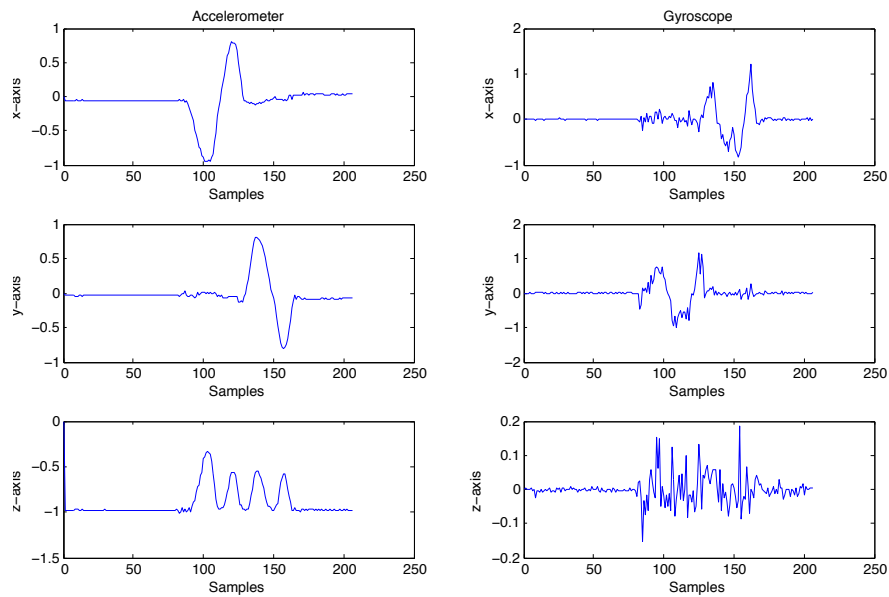
Figure 5.1: Simple test on the table, sensor data for all three axes recorded at 5Hz. left: Accelerometer, right: Gyroscope
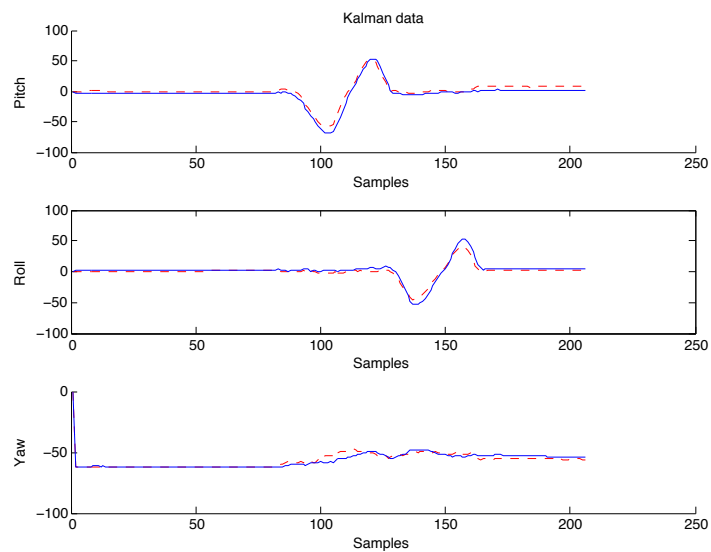

Figure 5.2: Simple test on the table, pitching up and down and rolling left and right; MIDGII 'solid line', Kalman filter 'dashed-line'

The second experiment conducted involved a small gas powered UAV. The low-cost IMU was placed inside of a housing to add some weight to the IMU and simulate similar conditions as the MIDGII. The MIDGII, low-cost IMU, and UmarimLite were placed inside the UAV, towards the middle and placed inside one inch thick soft foam boxes to reduce vibrations from the engine. The data is logged at 40Hz on the umarimLite. The MIDGII and Kalman filter algorithm is running at 50Hz and logged at 5Hz.

The data shown in the graph is real time data, the low-cost IMU is implemented in C++ on the Roboard-100 taking measurements from the MPU-6000 evaluation board. The Kalman filter did not behave as it did on the table and did not track the same roll and pitch as the MIDGII@. The pitch angle had some variance and the roll angle did not detect any of the turns on the UAV. Data from the UAV performing a constant bank to the left then to the right is shown in figure 5.4. This is a common problem in sensor fusion algorithms without GPS where the centrifugal forces acting upon the UAV are not accounted for and removed from the accelerometer. In figure 5.3 the accelerometer and gyroscope data from the constant bank is shown. Observing the z-axis in the accelerometer data, it can be seen that during the flight the external forces are acting on the z-axis with the force consistently exceeding 1g.
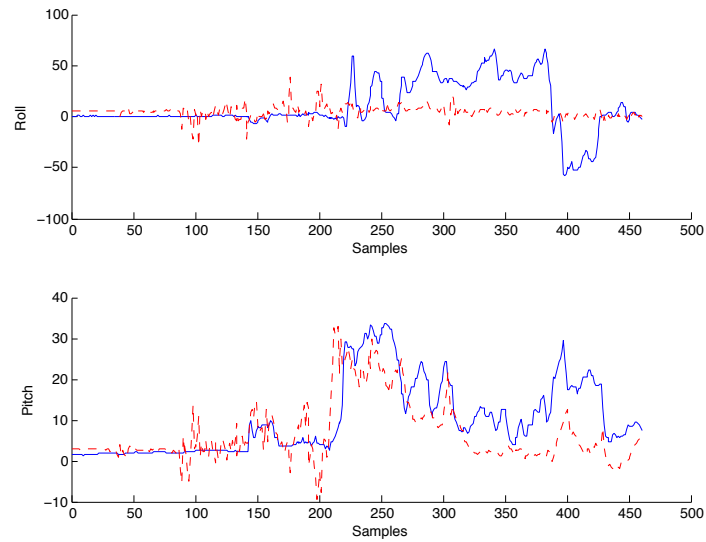
Figure 5.3: Constant Bank data from on-board real time flight data. Top: Pitch, Bottom: Roll
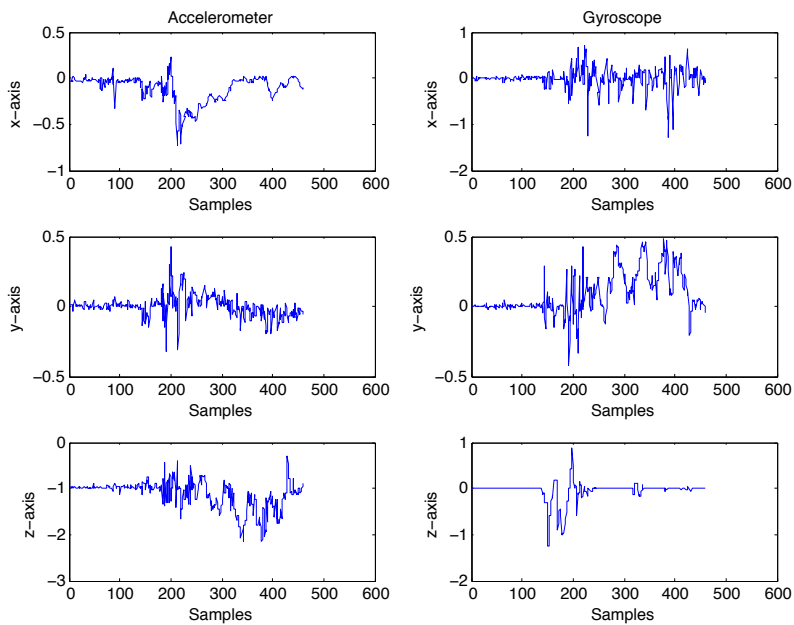

Figure 5.4: Constant bank sensor data from on-board real time flight data. left: Accelerometer, right: Gyroscope

The Kalman filter should be able to compensate for these forces, however they must be more accurately modeled in the noise measurement matrix. The reason for to much weighting of the accelerometer input in the measurement, is that the calculation of the noise measurement matrix is only an estimate and is not perfect. To improve the performance of the Kalman filter the noise measurement matrix is increased by a factor of 10 to decrease the amount of accelerometer influence of the estimated attitude.

$$
R_k = \begin{bmatrix} -0.05112 & 0 & 0 \\ 0 & 0.07291 & 0 \\ 0 & 0 & 1.16114 \end{bmatrix} \tag{5.1}
$$

The Third experiment contained the same setup as the first flight aboard the small gas powered UAV. The Kalman filter used the updated noise measurement matrix to test the influence of the accelerometer with the centrifugal forces on the estimated attitude. Without a GPS it is necessary to take account for these forces; the gyroscope is not effected by these forces, however they are present in the accelerometer measurement. This is why the only matrix needed to update is the noise measurement matrix which determines the likelihood of accelerometer influence. This flight path consisted of the UAV flying in a figure eight pattern maintaining a constant bank, then to a rolling vertical decent to a rolling vertical incline ending with a loiter circle before landing. It is important to observe the sensor data recorded from the flight in figure 5.5. The accelerometer is on the left measured in g's, the gyroscope is in the middle measured in radians/sec, and the magnetometer is on the right measured in gauss.
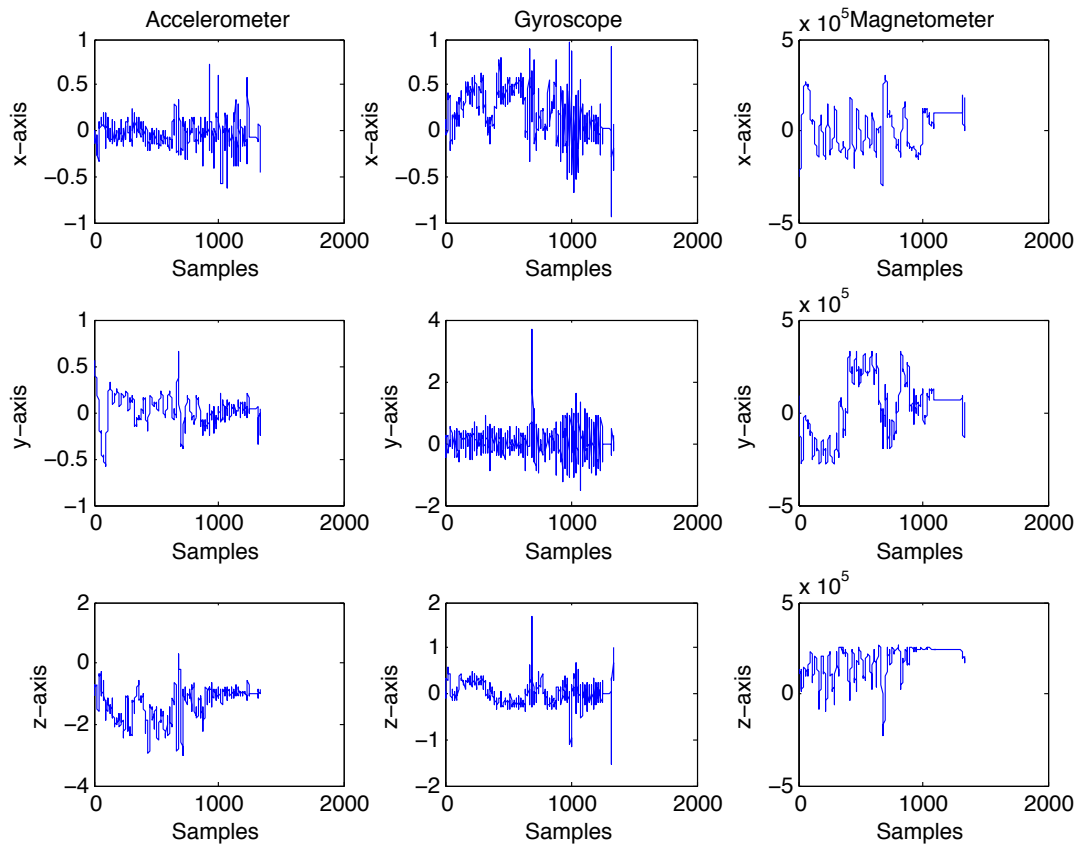
Figure 5.5: Sensor data for all three axes from on-board real-time flight data. left: Accelerometer, middle: Gyroscope, right: Magnetometer

The data is recorded at 5Hz since the noise is much greater when recorded at 50Hz, and it is difficult to make any observations. It can be seen that the accelerometer is influenced heavily by the centripetal/centrifugal forces by looking at the z-axis data. In the simple table data the mean of the z-axis data is -0.8936 g's, never exceeding 1g ($9.83m/s^2$) of gravity. In the real time flight data, the accelerometer z-axis senses turns greater than 3 g's, and has a mean of 1.3117 g's. Sensor fusion algorithms rely on the fact that the z-axis will never exceed 1g, and from the flight data it is shown that most of the flight has a force of greater than 1g on the z-axis rendering most sensor fusion algorithms useless on a gas powered UAV.

The flight data results in figure 5.6, 5.7, and 5.8, verify the accuracy of the Kalman filter algorithm in real time. It also can be seen that the Kalman filter performs better than the open source paparazzi DCM complementary filter algorithm. The algorithm is capable of maintaining accurate attitude estimation with the introduction of centripetal forces obstructing the accelerometer measurements. The algorithm was also able to keep attitude during the constant banking of the aircraft without the compensation of GPS data, as well as keeping the correct attitude orientation during the high dynamic conditions. The pitch angle was not as precise as the MIDGII during the constant bank turns, which may be corrected with tuning of the noise measurement covariance matrix. The yaw angle computed from the magnetometer with the soft and hard iron compensation verifies accurate calculation as well.
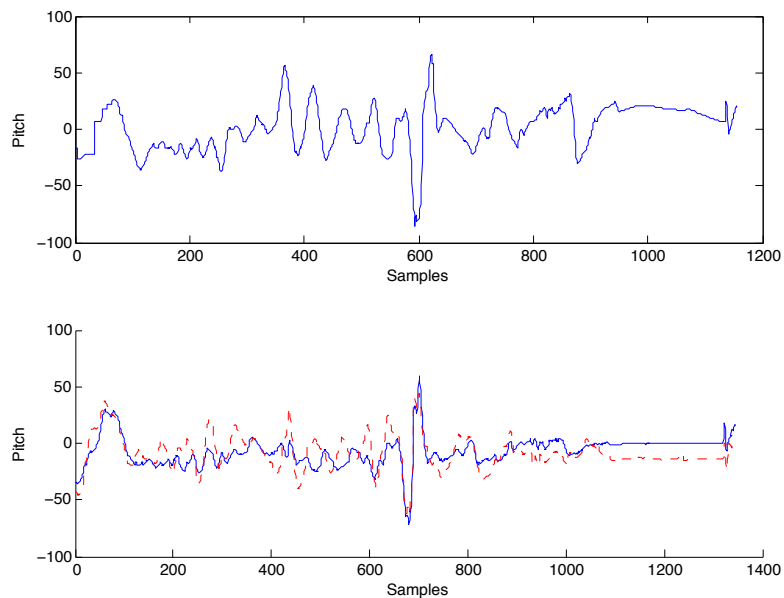

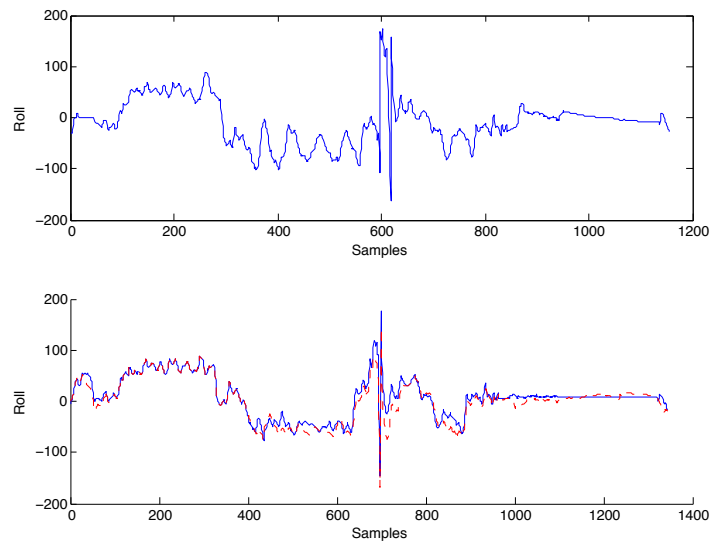**Figure 5.6: Pitch from on-board real-time flight data**

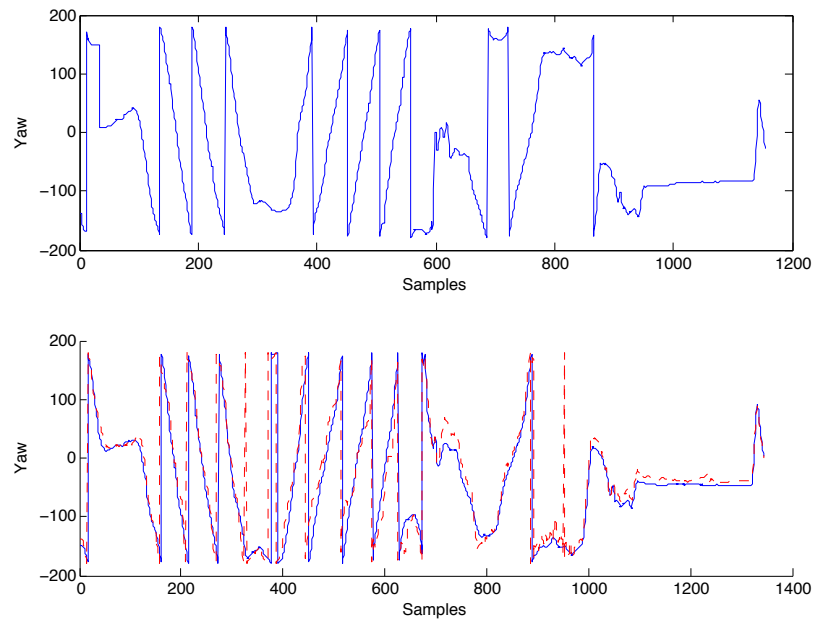**Figure 5.7: Roll from on-board real-time flight data**

Figure 5.8: Yaw from on-board real-time flight data

The last experiment was used to test the Kalman filter algorithms accuracy in flying way points. The setup is the same for the previous flight experiments with the addition of the VCU miniFCS autopilot. The yaw, pitch, and roll were calculated using the Kalman filter on the Roboard and transmitted through UART communication to the miniFCS autopilot. The miniFCS uses control algorithms onboard to calculate the target roll and pitch and a GPS to calculate the heading and position. The accuracy of the roll and the pitch are verified in the comparison of the MIDGII and UmarimLite data, however it is necessary for an AHRS to be capable of flying a plane autonomously which this experiment will verify.

In autonomous mode, the AHRS is the only indicator of the attitude of the UAV. If there is error in the attitude estimation from the Kalman filter, there can be disastrous results. The miniFCS autopilot uses PID loops that are continuously updated using the attitude information, if the pitch has error windup can occur in the PID loop and the plane may go into an unrecoverable steep decent. The same is possible in the roll angle, whereas if the centrifugal forces effect the roll angle, it is possible that the UAV may never receive attitude information that the plane is rolling, and be put into an unrecoverable roll and crash. This shows that an AHRS must not only be comparable to a reference, but also tested in a real time application to verify its use as and AHRS in an autopilot. When connected to a larger system such as an autopilot, there are other factors that may arise such as windup in the PID loops that can have disastrous effects.

The experiment included two flight paths, one with the MIDGII used for the attitude and another with the Kalman filter. The target way points are represented by triangles and the flightpath recorded in real time at 2Hz, is represented by 'o'. The target roll and pitch angles calculated from the autopilot control loops and the attitude from the MIDGII and Kalman filter is also recorded in real time at 2Hz.

In figure 5.9, the target waypoints and the clockwise flightpath of the UAV are shown. The UAV successfully completed flying to each waypoint. In figure 5.10 the target roll in red and MIDGII roll in blue are shown on the top and the target pitch in red and MIDGII pitch in blue are shown on the bottom. It can be seen that the MIGDII does not follow the target roll and pitch perfectly and has a little noise in the angles. This is acceptable since it is very difficult to match the exact target roll and pitch especially with the different environmental noises such as the wind or mechanical and software control parameters on the UAV. Even with the little error in the MIDGII attitude measurements, the UAV successfully flew the target way points.
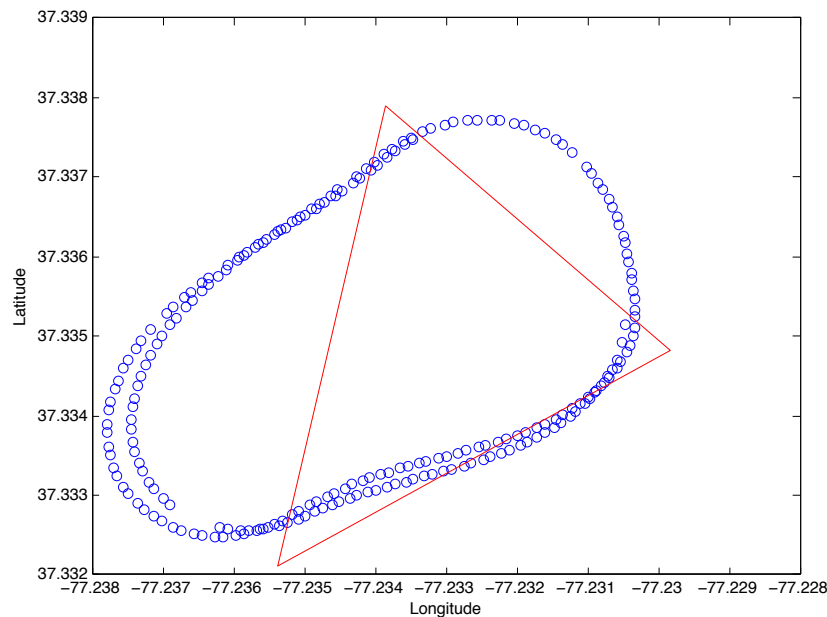

Figure 5.9: Way point flight test using the MIDGII attitude in real time: GPS target Way points (triangle), UAV flightpath 'o'
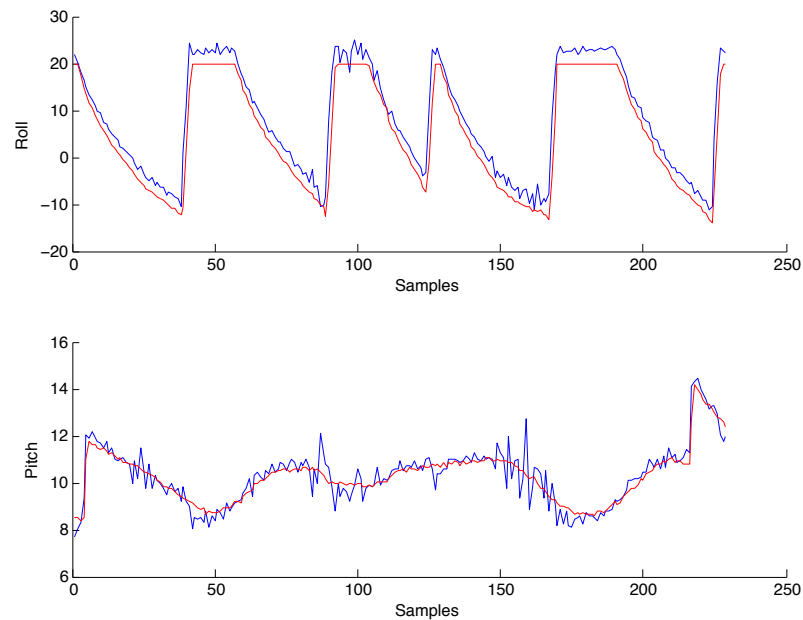
Figure 5.10: Way point flight test in real time: Roll on top and Pitch on bottom: Target 'red' , MIDGII 'blue'

In figure 5.11, the target way points and the counterclockwise flightpath of the UAV are shown. The UAV successfully completed flying to each way point. It is different from the previous flight with the MIDGII, however that is not caused by the attitude of the plane, but the algorithms on the miniFCS autopilot used to calculate a flightpath. In figure 5.12 the target roll in red and Kalman roll in blue are shown on the top and the target pitch in red and Kalman pitch in blue are shown on the bottom. It is important from the AHRS perspective, to match the target roll and pitch as closely as possible. There is some noise in the measurement which can be caused by the Kalman filter weighting too much accelerometer in the attitude estimation. It can also be caused by one of the PID tuning parameters in the miniFCS autopilot, which may be able to smooth it out since the UAV is tuned using the MIDGII. It may be that the MIDGII is detecting the turning of the vehicle faster or slower than the Kalman filter, causing the UAV to respond differently.

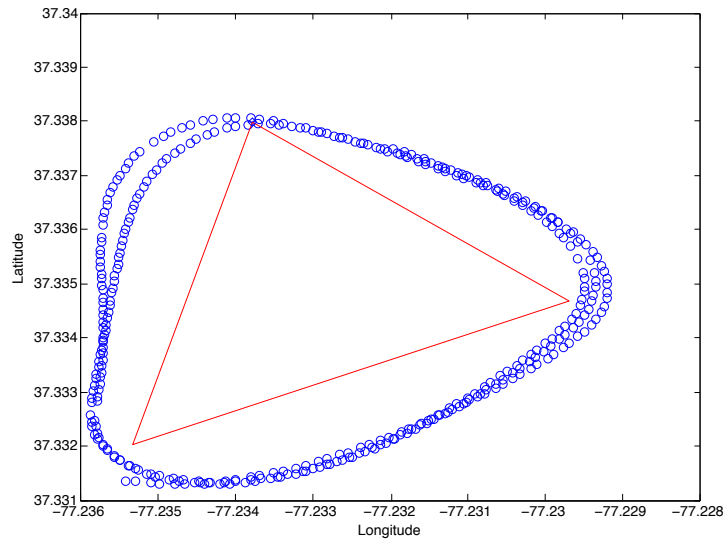**Figure 5.11: Way point flight test using the Kalman filter attitude in real time: GPS target Way points (triangle), UAV flightpath 'o'**
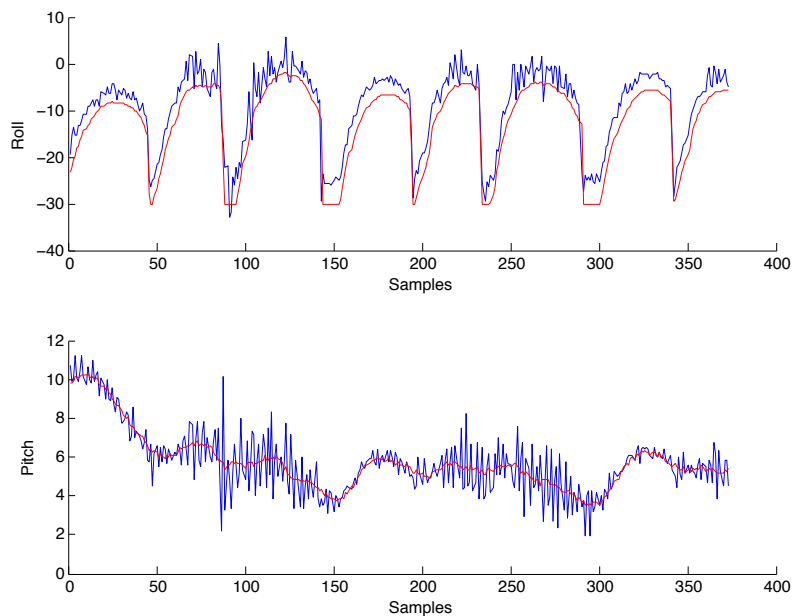

Figure 5.12: Way point flight test: Roll on top and Pitch on bottom in real time: Target 'red' , Kalman 'blue'

The results from the way point flight test show comparable results between the Kalman filter and the MIDGII attitude measurements. There is a little more noise in the Kalman filter measurements, however these can be fixed with a moving average filter on the output, or finer tuning of the covariance matrices of the Kalman filter. Overall, this verifies that the Kalman filter is capable of providing the attitude information with an accuracy capable of flying a UAV autonomously. In table 5.1 the mean error is shown for the target vs. measured values of the roll and pitch angles calculated by the MIDGII and Kalman filter. It can be seen that the error between the MIDGII and the Kalman for the Roll error is 0.5241° and the pitch angle is 0.2959°

Mean Error	Roll	Pitch
MIDGII	2.9205°	0.3294°
Kalman	3.4446°	0.6253°

Table 5.1: Mean Error of Target vs. Measured Roll and Pitch Angles

Chapter 6

Conclusions and Future Work

The implemented Kalman filter algorithm has verified that it is capable of providing an efficient, robust, and accurate solution comparable to the golden standard MIDGII as well as successfully providing attitude information to fly a UAV autonomously using a low-cost IMU. The Kalman filter shows excellent results in the ideal setting. This proves that the algorithm behaves appropriately under basic conditions. Under more rigorous conditions the algorithm has little error in the pitch and yaw angles. The algorithm has capitalized where other previously tested IMUs have failed with the centripetal/centrifugal forces problem where the IMU would output the attitude as a level orientation while in a constant bank. The problem has been observed with the initial setup of the Kalman filter algorithm. The noise measurement matrix was changed to accommodate for the absence of the GPS to correct for centrifugal forces and improved the modelling of the accelerometer measurement.

The error of the pitch angle may also be caused by the forces on the accelerometer, however the same type of error is shown in the UmarimLite low cost reference used for comparison. The error is not present in the MIDGII, which uses the Kalman filter; This may show how

the Kalman filter can progress where the complementary filter cannot. This leads to the assumption that the noise measurement matrix may have too little accelerometer influence in the pitch angle, and needs to be more accurately tuned. This assumption is made from observing that when the gyroscope data is used without the accelerometer, it produces an oscillating effect on the output due to the drift. In addition, this error might be caused by the low-cost tri-axis gyroscope's sensitivity to noise and distorted accelerometer measurement from vibrations and external forces, creating a less accurate solution. There are MEMS single-axis gyroscope sensors that may perform better, however, these single-axis sensors are more expensive and do not meet the requirements of a low-cost IMU. The magnetometer proves to perform accurately without GPS position or heading information.

Future work will include optimizing the algorithm beginning with porting the code from C++ to C for improved performance and portability. The Kalman filter algorithm will be implemented on the miniFCS autopilot system designed by VCU and tested to verify autonomous way point flight with on-board computation of the Kalman filter as well as the control algorithms on a low-cost system. When an AHRS is used in a flight control system Improvements of the Kalman filter accuracy and robustness will require more precise calibration as well as real-time tuning of the covariance matrices. To improve the efficiency of the Kalman filter, matrix multiplication can be optimized.

Bibliography

[1] CHRobotics, "Understanding euler angles," *http://www.chrobotics.com/wp-content/uploads/2012/11/Inertial-Frame.png*.

[2] M. C. F. Exchange, "Quaternion.m," *http://www.mathworks.com/matlabcentral/fileexchange/33341-quaternion-m*, 2013-16-1.

[3] S. Electronics, "Sparkfun razor imu," *https://www.sparkfun.com/products/10736*.

[4] S. Electronics, "Sparkfun razor stick 9dof," *https://www.sparkfun.com/products/10724*.

[5] Invensense, "Mpu-6000/mpu-6050 9-axis evaluation board user guide," Tech. Rep. AN-MPU-6000EVB-00, Invensense, Inc., 2011.

[6] Invensense, "Mpu-9150 9-axis evaluation board user guide," Tech. Rep. AN-MPU-9150EVB-00, Invensense, Inc., 2011.

[7] P. R. . Electronics, "Pololu miniimu-9 v2," *http://www.pololu.com/catalog/product/1268*.

[8] I. Transition Robotics, "Aspirin imu v2 10dom," *http://transition-robotics.com/products/aspirin-imu-v2-1*.

[9] ckdevices, "Mongoose 9dof imu," *http://store.ckdevices.com/products/Mongoose-9DoF-IMU-with-Barometric-Pressure-Sensor-.html*.

[10] S. Electronics, "Arduimu," *https://www.sparkfun.com/products/11055*.

[11] R. Mechatronics, "Chimu," *http://www.ryanmechatronics.com/ProductDetailCHIMU.htm*.

[12] I. Microbotics, "Midgii," *http://www.microboticsinc.com/Products.htm*.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[14] F. Sigernes, M. Marmion, and J. M. Holmes, "Airborne attitude estimation using a kalman filter," 2006.

[15] D. B. Kingston and A. W. Beard, "Real-time attitude and position estimation for small uavs using low-cost sensors," in *in AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, pp. 2004–6488, 2004.

[16] D.-M. Ma, J.-K. Shiau, I. C. Wang, and Y.-H. Lin, "Attitude determination using a mems-based flight information measurement unit," *Sensors*, vol. 12, no. 1, pp. 1–23, 2011. 22368455.

[17] V. Sazdovski, T. Kolemishevska-Gugulovska, and M. Stankovski, "Kalman filter implementation for unmanned aerial vehicles navigation developed within a graduate course," 2005.

[18] D. Jung and P. Tsiotras, "Inertial attitude and position reference system development for a small uav," *AIAA Infotech at aerospace*, pp. 7–10, 2007.

[19] P. Li, W. T. Miao, L. J. Hong, and W. Song, "An attitude estimate approach using mems sensors for small uavs," in *Industrial Informatics, 2006 IEEE International Conference on*, pp. 1113–1117, 2006. ID: 1.

[20] Starlino, "A guide to using imu (accelerometer and gyroscope) in embedded applications," *http://www.starlino.com/imu_guide.html*, 2009-12-09.

[21] C. Konvalin, "Technical document: compensating for tilt, hard iron and soft iron effects," Tech. Rep. MTD-0802, MEMSense, 2008.

[22] E. R. Bachmann, I. Duman, U. Y. Usta, R. B. Mcghee, X. P. Yun, and M. J. Zyda, "Orientation tracking for humans and robots using inertial sensors," in *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pp. 187–194, 1999. ID: 1.

[23] J. L. Marins, X. Yun, E. R. Bachmann, R. B. Mcghee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, pp. 2003–2011 vol.4, 2001. ID: 1.

[24] D. Comotti, "Orientation estimation based on gauss-newton method and implementation of a quaternion complementary filter," tech. rep., 2011.

[25] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira, "Low-cost attitude and heading reference system: Filter design and experimental evaluation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2624–2629, 2010. ID: 1.

[26] A. Tayebi, S. McGilvray, A. Roberts, and M. Moallem, "Attitude estimation and stabilization of a rigid body using low-cost sensors," in *Decision and Control, 2007 46th*

IEEE Conference on, pp. 6424–6429, 2007. ID: 1.

[27] S. Colton and F. Mentor, "The balance filter," *Presentation, Massachusetts Institute of Technology*, 2007.

[28] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing uav," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 340–345, 2008. ID: 1.

[29] W. MathWorld, "Special orthogonal matrix," *http://mathworld.wolfram.com/SpecialOrthogonalMatrix* 2013-08-05.

[30] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008. ID: 1.

[31] G. Baldwin, R. Mahony, J. Trumpf, T. Hamel, and T. Cheviron, "Complementary filter design on the special euclidean group se(3)," in *European Control Conference, 2007*, 2007.

[32] W. Premerlani and P. Bizard, "Direction cosine matrix imu: Theory," *DIY Drones.[Online][Cited: 1 7 2012.] http://diydrones.ning.com/profiles/blogs/dcm-imu-theory-first-draft*, 2009.

[33] E. Edwan, J. Zhang, J. Zhou, and O. Loffeld, "Reduced dcm based attitude estimation using low-cost imu and magnetometer triad," in *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on*, pp. 1–6, 2011. ID: 1.

[34] C. Liu, Z. Zhou, and X. Fu, "Attitude determination for mavs using a kalman filter," *Tsinghua Science and Technology*, vol. 13, no. 5, pp. 593–597, 2008. ID: 1.

[35] N. H. Q. Phuong, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *j-jucs*, vol. 15, pp. 859–876, feb 2009.

[36] D. Du, L. Liu, and X. Du, "A low-cost attitude estimation system for uav application," in *Control and Decision Conference (CCDC), 2010 Chinese*, pp. 4489–4492, 2010. ID: 1.

[37] L. Lou, X. Xu, J. Cao, Z. Chen, and Y. Xu, "Sensor fusion-based attitude estimation using low-cost mems-imu for mobile robot navigation," in *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*, vol. 2, pp. 465–468, 2011. ID: 1.

[38] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson, "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors," in *Position Location and Navigation Symposium, IEEE 2000*, pp. 185–192, 2000. ID: 1.

[39] C. Perez-D'Arpino, D. Vigouroux, W. Medina-Melendez, L. Fermin, R. R. Torrealba, J. C. Grieco, and G. Fernandez-Lopez, "Development of a low cost inertial measurement unit for uav applications with kalman filter based attitude determination," in *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pp. 178–183, 2011. ID: 1.

[40] M. R. Akella, J. T. Halbert, and G. R. Kotamraju, "Rigid body attitude control with inclinometer and low-cost gyro measurements," *Systems & Control Letters*, vol. 49, pp. 151–159, 6/14 2003.

[41] RoBoard, "Roboard-100," May 2012.

[42] D. Douxchamps, "A small list of imu / ins / inu," *http://damien.douxchamps.net/research/imu/*, 2013-03-10.

[43] S. Dave, "Kalman filter tutorial," *http://studentdavestutorials.weebly.com.*