12-2006

Developing a Benchmark Suite for the Evaluation of Orientation Sensors

Kelly Waller
Clemson University, Ram73@juno.com

Developing a Benchmark Suite for the Evaluation of Orientation Sensors

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Kelly Waller
December 2006

Accepted by:
Adam Hoover, Committee Chair
Richard Brooks
Eric Muth

ABSTRACT

This paper examines the problem with the lack of standardization through which orientation sensors are evaluated. An orientation sensor is a device for measuring physical orientation, and has many uses including navigation, virtual reality, augmented reality, human-computer interface, and robotic systems. We are specifically concerned with 3-axis sensors implemented on the microelectromechanical level. These sensors are produced and sold with data sheets that outline their performance, but lack the conditions under which the testing takes place. In this research, a testing apparatus was developed, and testing routines were designed to evaluate the different characteristics of orientation sensors under different motion conditions. Three orientation sensors, each in a different price range, were evaluated with the benchmark suite. The developed testing apparatus is a turntable that can precisely spin an orientation sensor via a stepper motor, and can record its exact orientation along with the heading read from the orientation sensor. Sets of movements we call benchmark routines were designed and implemented to test different properties of the sensors.

The results of this research show that the turntable performs correctly, and is a viable way to test orientation sensors. The three orientation sensors tested show that as expected sensors with similar data sheet specifications perform very differently. Under smooth, low-speed conditions (.1 $\frac{rev}{s}$), the cheaper MOCOVE orientation sensor outperformed the more expensive InertiaCube3 orientation sensor. However, as the angular velocity increased, the error of the MOCOVE quickly increased. The error of the Honeywell HMR3300 increased as well. Only the InertiaCube3 could accurately measure quicker angular velocities (up to 5-6 $\frac{rev}{s}$) commonly found when tracking the movements of human head and arm movements. Overall, this research shows the need for such a benchmark, and will aid in the process of choosing an orientation sensor for a particular application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

List of Figures (Continued)

Chapter 1

Introduction

An orientation sensor is a device for measuring physical orientation. The obvious use for an orientation sensor is in navigation, however, there are many other uses for them in research such as studying the movements of humans, animals, and robots. The implementation of the orientation sensors we are interested in is done using a minimum of one of three technologies: accelerometers, gyroscopes, and magnetometers. Each technology has advantages and disadvantages, and companies combine the technologies in different ways to produce their sensors. Companies release data sheets that publish a sensor's accuracy and repeatability, but usually lack the conditions under which the testing was conducted. With no standard for evaluation, it is hard to compare the performance of one sensor to another. This thesis examines the problem with this lack of standardization, and evaluates three particular sensors.

An orientation sensor can be as simple as a compass. A compass is a 1-axis orientation sensor that measures orientation in the yaw axis. Figure 1.1 shows the common 3-axis coordinate system used to describe orientation. We are specifically interested in 3-axis sensors that return a heading electronically. They are made up of both mechanical and electrical parts, and work in an un-machined world. When markers are placed in the surrounding environment and are used by an onboard sensor to determine position or orientation, the

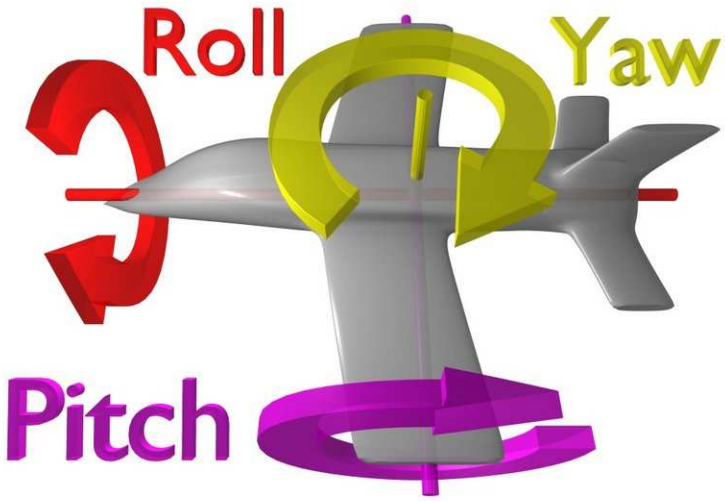

Figure 1.1: The yaw, pitch, and roll 3-axis coordinate system used in orientation sensors.

environment is said to be machined. Additionally, we are specifically looking at low-power sensors with a small package (less than 6”).

All of the sensors in this category operate using the three technologies mentioned. An accelerometer is a device that measures acceleration. Specifically in an orientation sensor, angular acceleration is measured instead of linear acceleration. Since acceleration is the 2nd derivative of position, orientation can be recovered through double integration of the measured angular acceleration. A gyroscope on the other hand is a device that measures angular velocity. Velocity is the 1st derivative of position, so similar to the accelerometer, the orientation can be recovered through a single integration of the measured angular velocity. Lastly, a magnetometer is a device that measures the strength and/or direction of magnetic fields. Since the earth has a magnetic field, this sensing technology can map an orientation to a field strength and direction directly without the need for integration.

Orientation sensors allow the tracking of rotational movements such as the head or limbs of a human. They have also been used in the tracking of robotic links, and to implement human-computer interface devices such as a mouse or joystick. Many modern cars

have an onboard digital compass, and while GPS can determine your position, many GPS receivers have a built in orientation device that is used to correctly orient the displayed map.

Not all orientation sensors are created equal. The accuracy and repeatability depend on how precise the sensors have been built, how many orientation sensing technologies are used, and how the sensing technologies are combined. There are a wide range of sensors available, from the low-cost, single-axis, single-sensing technology all the way up to the high-cost, 3-axis, three-sensing technologies.

Before explaining our current work, it is important to see what research has already been done on orientation sensors, along with how they are being used. The majority of the work with orientation sensors involves the tracking of head movements. Head mounted displays, or HMDs, are used to show a user a virtual or augmented world relative to the user head position. To measure the head position, some researchers have used an environment filled with precisely positioned LEDs paired with an HMD mounted with photo diode sensors to calculate head position [17]. It takes time to compute the position and orientation of the HMD, and this makes the virtual or augmented images lag. To improve the performance of this type of system, an orientation sensor has been added to the HMD, to predict both the head movement and orientation [5]. When the prediction is accurate, the images match reality. In many cases, a machined environment is not applicable, and another means of tracking head movements is needed. Research has been done to use small onboard sensors to calculate the head movement. One augmented reality project considered 3 different orientation sensors for the application, and tried each one and observed which performed the best in the project [4]. No specific measurements were made as to the amount of error, and under what conditions the error occurred. This type of testing is common as researchers are interested in their application working more than the overall performance of the sensor. Usually, the testing of a single sensor, which was chosen for the application via the published data sheet, is done to evaluate its performance in a particular

application. For example, professors at the University of Southern California measured the accuracy of a 3-axis gyroscope orientation sensor while working on an augmented reality project. The project combined the sensor with a camera to improve the tracking of objects in a scene. However, instead of having a ground truth to compare the heading of the orientation sensor to, the tracking of features using the camera was used in order to derive the position [21]. Another augmented reality project chose to use two gyroscopes in conjunction with a camera. It was found that the drift error was more than advertised due to changes in temperature and power fluctuations [1]. Even with drift correction from a closed loop system involving image processing techniques, large accelerations caused the absolute tracking performance to suffer greatly. Increasing beyond head tracking, research has been done to track the limbs of the human body in an effort to create an avatar, a real-time animation that captures human motion [22]. Orientation sensors were used in the application because they do not get in the way of the subjects movement as much as implementations that use markers that are worn by the subject and are seen by several cameras and/or other sensors mounted in the environment. Other applications such as a mouse that uses two accelerometers instead of a rolling ball or optical sensor have also been created [12]. Recent orientation sensors however are rarely built using only one sensing technology. Several research studies have been done to develop the MARG, or Magnetic Angular Rate Gravity sensor module. One study developed a MARG that could be used to measure rigid bodies such as robots or humans [6]. Research was also done on modeling how humans move, including the maximum acceleration and velocity of their joints [9]. This modeling helped determine the minimum performance characteristics needed by MARGs. In order to improve the accuracy of a MARG, research was done to combine the headings from each internal sensor via a Kalman filter [2]. Filtering can add latency, so one study looked at several different filters to see which filter would produce the best results, meaning the best balance between latency and accuracy, for the movement an HMD commonly undergoes [20]. Other research has looked at how orientation sensors are affected

by magnetic field variations [7], while another study focused on improving the accuracy of the sensors by improving the tuning methods used during their physical construction [11]. Companies such as InterSense develop, patent and sell their own MARGs which also use several Kalman filtering techniques [19].

Our research does not focus on the development of a new orientation sensor, or on evaluating a particular sensor that we are using in our application. It is also not focused on a particular orientation sensor technology. Rather, this thesis is on the development of a method for evaluating the performance of orientation sensors under different types of motion. Standard testing methods that evaluate the performance of a sensor under varying temperatures and power fluctuations are still needed, but are left as future work.

In this research a testing apparatus was developed, and testing routines were designed to evaluate the different characteristics of orientation sensors under different motion conditions. To our knowledge, we are the first research team to look at the effects of testing methods on the performance characteristics of orientation sensors. We expect that different angular velocities, accelerations, and jerks will produce varying accuracies. With a complete benchmark suite, more informative data sheets can be produced that will allow consumers to choose a sensor for their application more discriminately. Three sensors, the Honeywell HMR3300, the InterSense InertiaCube3, and the Artis MOCOVE, were used and evaluated for the pilot testing.

1.1 Orientation sensor technologies

In this section, we discuss the implementation of the three prominent orientation sensor technologies. We also look at the advantages and disadvantages of each technology. Lastly, we will look at the implementation of the three orientation sensors used while conducting this research.

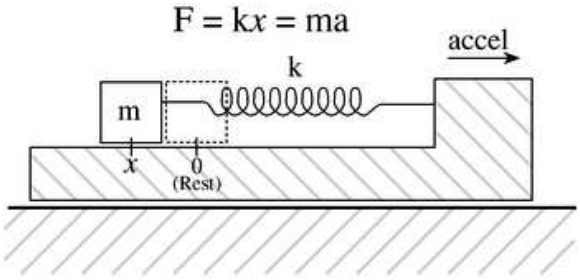

Figure 1.2: A mass-spring system used to measure acceleration. Adapted from [15]

The three main types of orientation sensor are accelerometers, gyroscopes, and magnetometers. All three technologies have been developed at the microelectromechanical systems (MEMS) level, which ranges in size from a micrometer to a millimeter. This small size makes the packaging of even 3-axis, multiple technology orientation sensors on the order of 3 centimeters or less. An accelerometer measures acceleration in g, and by integrating twice can produce a displacement. The basic physics of the sensor is based on the mass spring system shown in Figure 1.2. Hooke's law states,

$$F = kx \tag{1.1}$$

where F is the force, k is a proportionality constant, and x is the displacement. Newton's second law of motion states,

$$F = ma \tag{1.2}$$

where again F is force, m is mass, and a is acceleration. Combining these lets one measure acceleration via the displacement of a mass connected to a spring by

$$x = ma/k \tag{1.3}$$

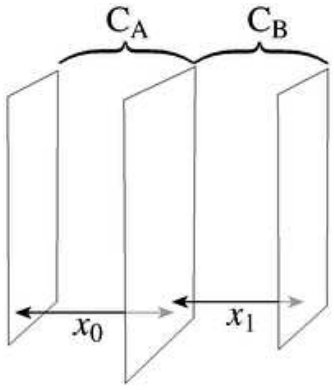

Figure 1.3: The dual capacitor setup used to measure distance. Adapted from [15]

To measure the displacement of the mass, the electrical properties of capacitance are used. In a parallel plate capacitor, the amount of capacitance is given by

$$C = k/x_0 \tag{1.4}$$

where k is a property of the material between the two plates, and x_0 is the space between the plates. Using this property, accelerometers make the moving mass a plate sandwiched between two fixed plates. In this configuration, the change in capacitance ($C_A - C_B$) is:

$$\Delta C \approx \frac{-2}{{x_0}^2} \tag{1.5}$$

for small values of displacement. In most sensors, an onboard application specific integrated circuit, or ASIC, conditions and filters the signal before outputting. Even so, double integrating discrete acceleration samples to get the position is inherently erroneous because of a limited sampling rate. This means some of the actual acceleration is lost (not sensed). Also, the sensor has limited precision, so error is introduced over time via round-off error. This error is seen as a drift over time. Most accelerometers have a low latency, and can be sampled quickly, and thus can keep up with quicker physical movements, up to 2-3 g. They

however lack an absolute position, and must be reset to a known position often to remove the accumulating drift.

A gyroscope (”gyro” for short) measures angular velocity, and is built around the Coriolis effect:

$$F = 2Mv\ X\ \Omega \tag{1.6}$$

where F is the force, M is the mass moving at velocity v, and Ω is the angular velocity of the reference frame. There are several MEMS gyroscope implementations including tuning-fork gyros, piezoelectric gyros, oscillating wheel gyros, Foucault pendulum gyros, and wine glass resonator gyros. All of them feature something that vibrates or spins, along with something that measures the displacement in the plane normal to the vibrating plane. Figure 1.4 shows a tuning fork that is being vibrated in one plane. The angular rate shown is the movement we would like to measure. When the tuning fork undergoes the angular rotation, the tuning fork vibrates in the plane normal to the driven vibration. Using equation 1.6, the mass of the tuning fork is known, as is the velocity at which we are vibrating it. The force is the sensed vibration, which leaves the only unknown being the angular velocity. This is essentially how the quartz tuning fork gyroscope works. It has 2 quartz tuning forks, arranged handle-to-handle that can be induced to vibrate. Once resonating, the displacement from the plane of oscillation can be measured to calculate the angular velocity. The displacement is measured using similar properties of capacitance that the accelerometers use. In the oscillating-wheel gyroscope shown in Figure 1.5, a wheel is made to spin/vibrate, and any rotation in the plane of the wheel causes the wheel to tilt. Using plates mounted under the wheel, the properties of capacitance can be use to measure the tilt, and the onboard ASIC can calculate the angular velocity. The piezoelectric gyroscope has a piezoelectric crystal that is vibrated, and measured, while the Foucault pendulum gyroscope uses a vibrating rod mounted out of the plane of a chip. Lastly, the wine glass resonators use a resonating fused silica ring. The single integration from veloc-

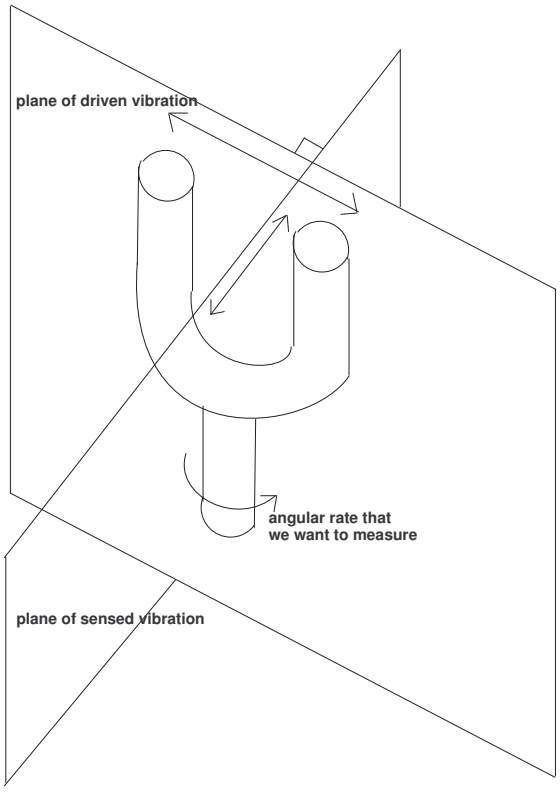

Figure 1.4: The general operation of a gyroscope.

ity to position will suffer from cumulative drift error for the same reasons accelerometers do.

A magnetometer measures the strength and direction of a magnetic field. In the case of an orientation sensor (digital compass), the magnetic field that is of interest is the magnetic field of the earth. The orientation sensors studied during this research relied on the magnetoresistive effect, which is the change in resistivity in thin film structures that are made up of alternating ferromagnetic and nonmagnetic metal layers [18]. In the case of the Honeywell HMR3300, the implementation is that of a resistive bridge device as shown in Figure 1.6. In this configuration, a magnetic field will cause a change in the bridge resistance which in turn causes a change in the voltage output. However, when the magnetic field is disturbed, the output voltage degrades. To fix this problem, and to clear the sensor

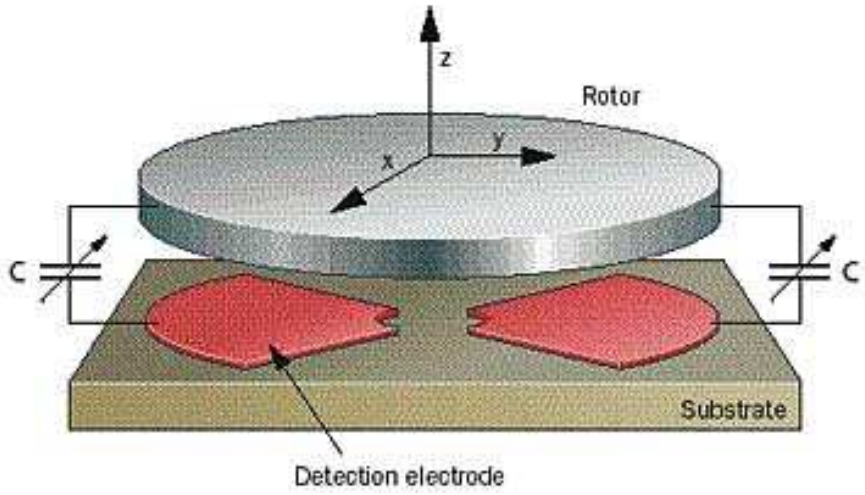

Figure 1.5: A diagram of a MEMS vibrating wheel gyroscope. Adapted from [8]

of past magnetic history, the sensor bridge is reset before every reading by pulsing a large current through the S/R (Set/Reset) strap. This large current generates a strong magnetic field which resets the bridge to an initial state. Once the current is turned off, the resistance of the bridge changes according to the surrounding magnetic fields, and the output voltage is measured. Since magnetometers measure absolute orientation, they do not suffer

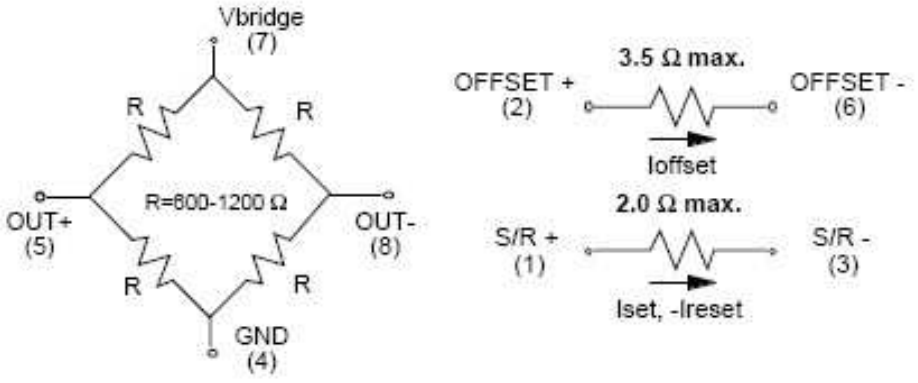

Figure 1.6: A circuit diagram of Honeywell HMR3300's resistive bridge. Adapted from [10]

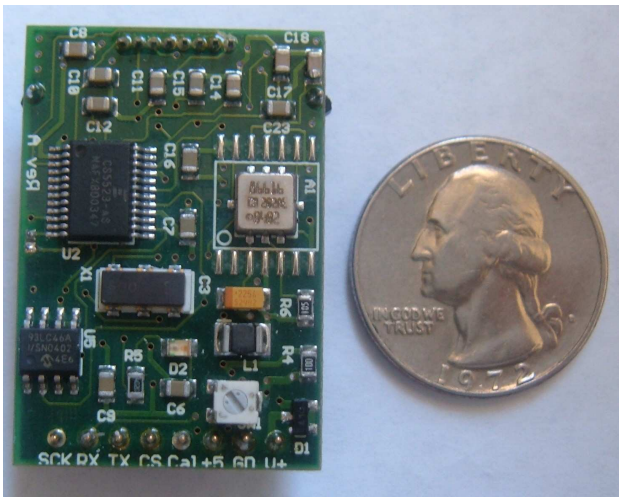

Figure 1.7: An external view of the Honeywell HMR3300 3-axis orientation sensor.

from drift error. However, ferrous metals and electromagnetic fields (EMF) can produce erroneous readings because a magnetometer cannot tell the difference between the earth's magnetic field and local magnetic fields. Also, since the circuit must be reset before each measurement, magnetometers of this type lack the high update rate of an accelerometer or gyroscope. Similar to a shaken hand held compass, it takes a little time for a magnetometer to converge on an accurate reading.

1.2 Commercial sensors

We studied three orientation sensors in this thesis: the Honeywell HMR3300, the InterSense InertiaCube3, and the Artis MOCOVE.

The HMR3300 shown in Figure 1.7 is a 3-axis sensor that contains a magnetometer on each of its axes. In addition, a 2g accelerometer is mounted to measure pitch and roll, and is only used to compute the heading when the sensor is tilted. The sensor is advertised to be accurate to 1 degree, with a resolution of .1 degrees, repeatability of .2 degrees, a long term drift of .2 degrees, and an update rate of 8Hz. It is priced at $400(U.S.).

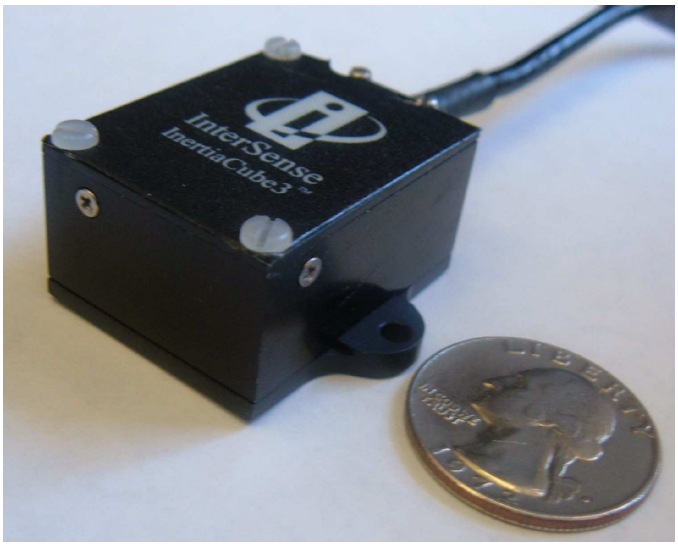

Figure 1.8: An external view of the InterSense InertiaCube3 3-axis orientation sensor.

The InertiaCube3 shown in Figure 1.8 is a 3-axis sensor that combines all three orientation tracking technologies in an attempt to have each technology overcome the weaknesses of the others. Each axis contains an accelerometer, a gyroscope, and an accelerometer in the configuration shown in Figure 1.9. The accelerometer and the gyroscope take care of the short-term, quick movements that the magnetometer cannot handle, while the magnetometer is used to combat the drift that the other two technologies suffer from.
Complimentary separate-bias Kalman filtering (an extended Kalman filter) is used to combine the measurements from all of the sensors. The sensor is advertised to be accurate to 1 degree, with a resolution of .03 degrees, and an update rate of 120 Hz. Repeatability and long term drift values are not given, however, more detailed conditions for the previous values are given. It is said to have a maximum angular rate of 1200 degrees, and a minimum of 0 degrees, which gives the user an indication of whether the sensor will fit a particular application. The accuracy is quoted at a temperature of 25 degrees Celsius, which is also helpful. Lastly, a minimum latency of 2ms is provided, which is due to the RS-232 communication speed. The sensor is priced at approximately $2000(U.S.).

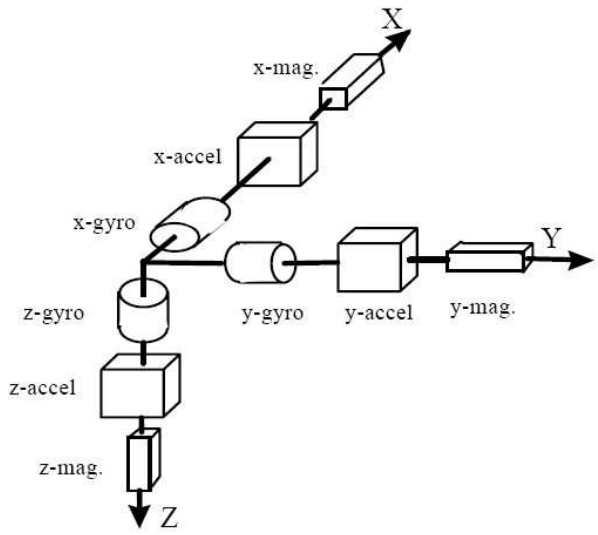

Figure 1.9: A function diagram of the InertiaCube3.

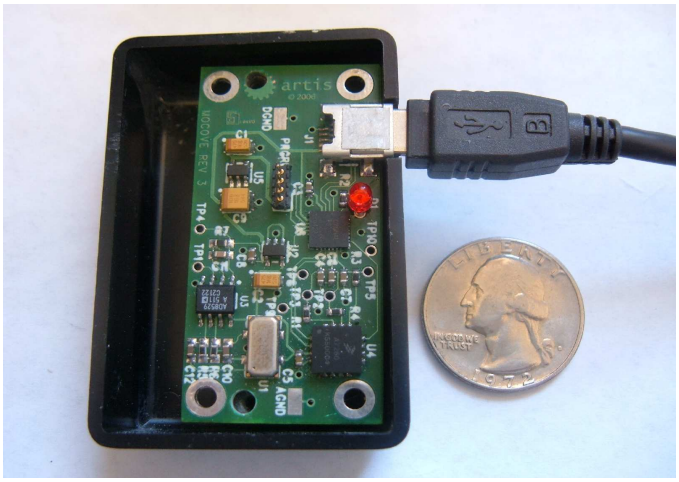

Figure 1.10: An external view of the Artis MOCOVE 3-axis orientation sensor.

The MOCOVE sensor shown in Figure 1.10 has 2g accelerometers on 2 axes (pitch and roll), and a gyroscope on the third axis (yaw). No official data sheet is available for the sensor, but according to [13], the sensor can be accurate to .2 degrees. The sensor has a resolution of .01 and costs approximately $250(U.S.).

1.3 MOUT background

This section provides background into the need for MOUT training, or Military Operations in Urban Terrain. It also covers some of the details about the training facility and equipment we built to record building clearing exercises.

This thesis was motivated by a larger research project into MOUT (Military Operations in Urban Terrain). Conflicts in areas such as Iraq, Somalia, and Afghanistan show that more and more fighting is occurring in urban environments. MOUT is becoming an important topic that infantry need to be trained in. While the military is doing research, current techniques for MOUT training are still young and most soldiers receive only a week of MOUT instruction before being deployed. This instruction comes from a training officer showing the proper positions and techniques, and then the trainees executing exercises in mock buildings or towns. Due to the large number of trainees and limited number of instructors, plus the size of the operations, a MOUT instructor can only give minimal feedback to the trainees based on what he can see, and the overall outcome of the operation. In an effort to improve on this, a few bases such as the U.S. Army base at Fort Polk, Louisiana have equipped their training facility with cameras and microphones that record an exercise to allow after-action review. While this is an improvement, the large platoon sized operations (40 men), limited tracking accuracy (1 m), and slow update rate (1 Hz), leave much to be desired.

We are currently doing MOUT research on the scale of a single building. A 4-5 man fireteam is trained, and then studied as they clear the house. Even though the number of

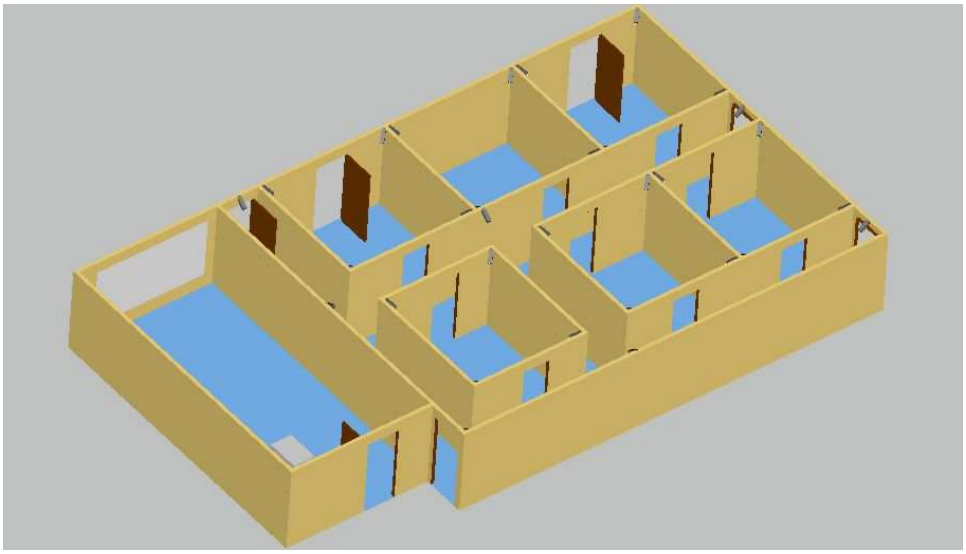

Figure 1.11: A CAD view of the shoothouse and instructor station.

participants trained per hour is lower, the information collected about each member's movements along with the team's overall movements is valuable. By learning what movements ensue success, training efforts can be maximized. Our facility is approximately 200 m^2 and is housed within the 263rd Army National Guard Air and Missile Defense Command site in Anderson, SC. This site was chosen due to the large amount of indoor space allotted, and its proximity to Clemson University. With 400 active personnel on-site, and an ROTC program at Clemson, this location also makes finding volunteers for our experiments easier. The building itself consists of two parts, a shoothouse, and an instructor station. The shoothouse shown in Figure 1.11 is where the building clearing exercises are performed and recorded, and consists of six rooms with interconnecting hallways. The walls are reconfigurable, allowing them to be different sizes, shapes, and have different entry points. The instructor station is used to view the after-action replay, to discuss operations both before and after the exercises, and to house the equipment. In order to record the position of the fireteam, a camera is placed in each corner of every room. Two additional cameras per room are installed to view each room's section of the adjacent hallway, which makes 6

Figure 1.12: The camera network at the shoothouse.

cameras per room, and 36 cameras total for the shoothouse. The images from the cameras are fed to a rack of computers in the instructor station, which use background subtraction along with several other techniques to track in real-time the floor-plane coordinates of each member of the fireteam. This system is accurate to approximately 10 cm and has an update rate of 20 frames per second.

Along with position, specific information about each team member such as weapon fires, weapon orientation, head orientation, hits taken, and heart rate can be viewed in real time and are recorded. To facilitate this, a laser tag system consisting of wireless laser tag rifles, and wireless helmets was developed. A wearable arousal meter (WAM) was used to measure heart rate. For more information about the WAM, see [14]. The design of the circuitry for the laser tag equipment can be seen in Figure 1.14. We were able to tie the wireless circuit and the control circuit together using the DPAC 802.11b Wireless LAN Module. The module gives us standard 802.11b communication capability along with

(a) rifle (b) helmet (c) WAM

Figure 1.13: The equipment used in building clearing exercises.

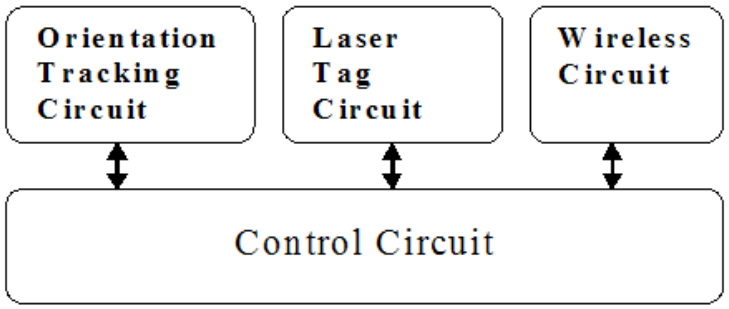

Figure 1.14: The circuit design for the laser tag equipment.

several I/O pins and an RS-232 serial port. Custom circuitry was developed for the laser emitter, and detector, and can be found in [16]. Figure 1.15 shows the DPAC module mounted on the custom circuit board, which implements a power supply, laser tag circuits, and the interface to the hardware on the rifle. For the orientation circuitry, the HMR3300 by Honeywell previously mentioned was interfaced with the DPAC's serial port. It was chosen because of its published size, power consumption, performance characteristics and price. The current research stems from the use of the Honeywell orientation sensor. While its data sheet specifications show it to be adequate for our application, our real world testing

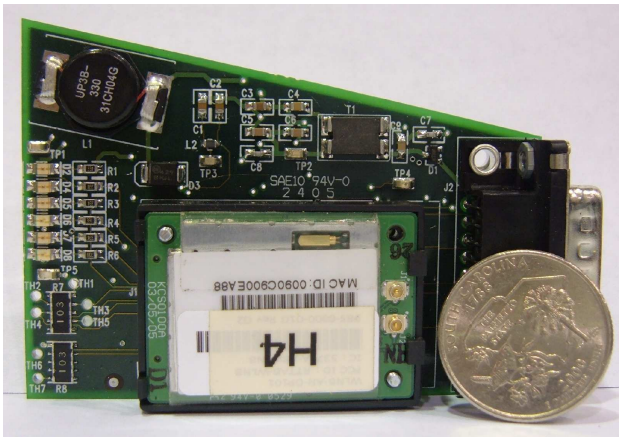

Figure 1.15: An external view of the DPAC 802.11b Wireless LAN Module mounted on our custom designed circuit board.

and implementation shows that it is not. The discrepancy between the two lies in the unpublished conditions that were used to evaluate the sensor for the reported performance characteristics.

Chapter 2

Methods

This section will cover the overall implementation of our benchmark suite. The first section will give the details of the physical hardware that were built and used, and the second section will give the details of the software written for controlling and recording. The last section will explain how the benchmark routines were designed and implemented.

To evaluate the performance characteristics of orientation sensors, we built a platform that we could attach an orientation sensor to, and then could precisely turn the platform via a stepper motor. This gave us a ground truth for the orientation to which we could compare the orientation heading produced by the attached sensor. The idea is that different jerks, accelerations and velocities will bring out the different qualities of each sensor. Software was written to implement sets of movements for the platform that tested the different characteristics of the sensors. The software also records the ground truth position of the platform, the measured orientation from the sensor, and then makes statistical calculations such as instantaneous error, average error, variance, and standard deviation. Section 2.1 will go into the details of the physical turntable, while section 2.2 will cover how the software control was implemented. The design of the platform movements, which we call the benchmark routines, will be covered in section 2.3.

Figure 2.1: The turntable apparatus built with a sensor attached.

2.1 Turntable hardware

Even though the orientation sensors in consideration all measure 2-3 axes, we assume the accuracy in each axis is comparable. To simplify the test bed, we focused on just the yaw axis. A spinning platform, or turntable, was built using a wooden table, a stepper motor and driver from Applied Motor Stepper Drives [3], a stainless steel shaft and coupler, and a wooden platform. This turntable can be seen in Figure 2.1 with the Honeywell orientation sensor attached to the platform.

A stepper motor was chosen due to its high resolution, high accuracy, high torque at low speed, high holding torque, and ease of programmability. The only drawback to a stepper motor compared to a servo motor is that it operates in an open-loop control system. This means that once the driver tells the motor to step, it has no way of knowing whether the motor actually completed the step or not. The drive must assume that the motor has stepped, which can lead to errors in positioning if for example the load on the motor was too great

Figure 2.2: An external view of the Applied Motion 2035i stepper motor driver and stepper motor.

for it to move. A servo on the other hand tells the motor to move, then checks to see where the motor actually is using an encoder, and can compensate accordingly. While this closed loop or feedback system is attractive, the price, ease of programmability, high resolution, and low-speed torque (without a gearhead) are all compromised. The stepper motor driver is the Si2035 and was chosen because of its 50,800 steps per revolution microstepping ability, its 2.0 amps/phase power, and because of its Si command language (SCL) mode which accepts real time commands and queries from a host PC. Figure 2.2 shows the driver along with the motor which is a NEMA 23 sized stepper motor that has a holding torque of 74.9 oz-in.

Several platform configurations were investigated before settling on the current setup. A gear drive system was employed in an attempt to limit the amount of weight the motor shaft has to bear. This was found to be impractical due to backlash issues that were going to limit the accuracy of the turntable. A coupled shaft setup was investigated with several different length shafts to see when the EMF from the motor was powerful enough to affect the orientation sensors. It was found that a 24” shaft, seemed to amplify the EMF when a sensor was placed right in the middle of the platform (on the shafts pole), and due to the

limits of machining and the rigidity of the material, the 24” shaft had a slight wobble to it, which would hurt the accuracy of the platform. In an attempt to support the shaft, pillow bearings were installed on an upright support, but the wobble of the shaft made the shaft bind in the bearings and increased the torque load on the motor.

The 6” shaft was more rigid and was found to be far enough away from the motor to not adversely affect the sensors. Even so, due to the findings with the longer shaft, all sensors were mounted on the edge of the platform instead of over the shaft’s pole. Two platform materials, Plexiglas and wood, and two platform sizes, 12” square and 4” square, were investigated as well. The Plexiglas platform, while lighter, was found to wobble under high jerk and/or velocity, while the more rigid, heavier wooden platform did not. While the 12” Plexiglas platform did not have weight issues, the 12” wooden platform’s larger mass, and moment of inertia limited the acceleration that the motor could generate without losing resolution. Thus, the 4” square wooden platform with the 6” shaft coupled to the stepper motor became our final turntable.

2.2 Turntable software

This section describes the software control for the turntable. All of the development was done in C++. The front-end GUI shown in Figure 2.3 was created to allow a user to start and stop each of the orientation sensors, monitor the heading of each sensor, monitor the position of the motor, choose a benchmark routine to run, start and stop the motor, and record the output of both the motor and the orientation sensor.

Communication with the stepper motor driver is accomplished via RS-232. The driver uses the proprietary Si Command Language (SCL), which consists of a set of ASCII commands and responses. There are two basic types of commands, buffered and immediate. Buffered commands are done sequentially one at a time, whereas immediate commands are executed right away, and are done in parallel with buffered commands. The immediate

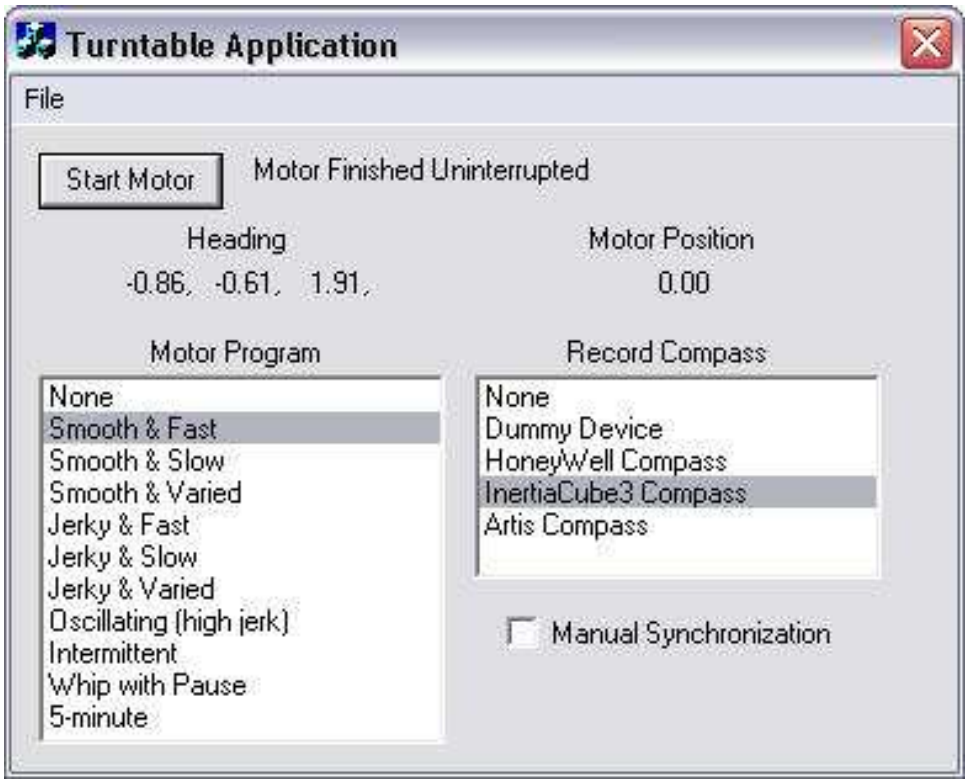

Figure 2.3: The GUI designed for the turntable application.

commands allow us to check the position of the motor while it is in motion running a move command. Example commands include the buffered command ”FL” which is a Feed to Length (move) command, and an immediate command, ”IP” which returns the Immediate Position in steps (hex) of the motor. We would like to know when a motor movement has finished so we can, for example, stop recording the orientation sensor output, or notify the user that a routine has completed successfully, or recognize when an error has occurred. To facilitate this, the properties of the buffered commands are harnessed though the Send String (SS string) command, which echoes whatever string is passed to it. Every time a move command is sent, it is immediately followed by an SS command. When the application receives the echo it knows that the move command has completed.

All of the platform and orientation sensor control was done through the use of threads. Figure 2.4 shows the general flow of the application. When the turntable application is

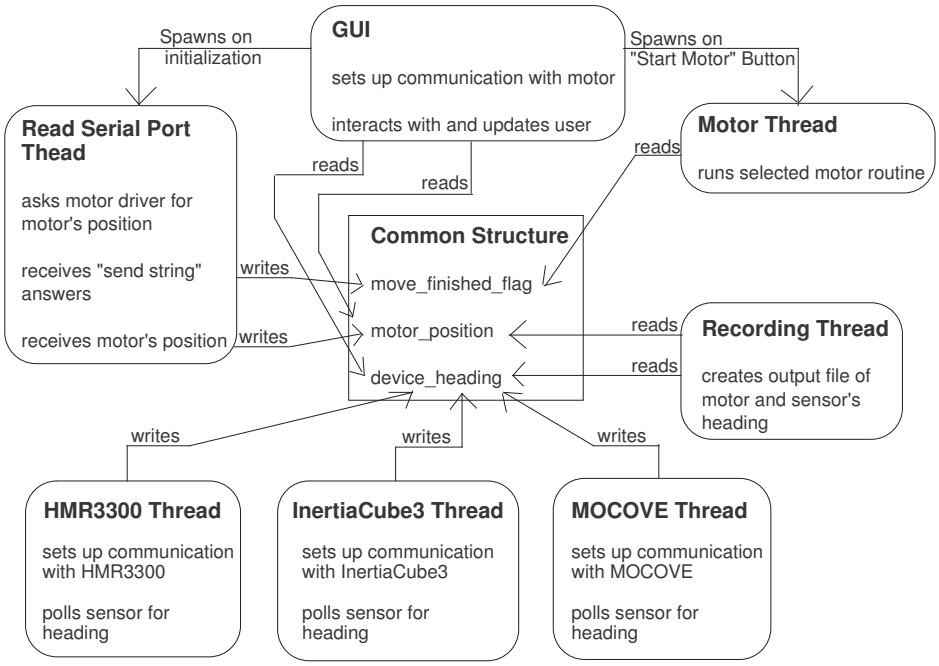

Figure 2.4: A simplified flow diagram describing the operation of the application.

first started, it attempts to set up communication with the motor. If successful, a thread is spawned that continually polls the motor driver, asking for the current position of the motor. Upon receiving an answer, the motor's position is stored in a structure that is protected by a mutex and is accessible to several other parts of the program. This thread is the only one that receives information from the stepper driver, and thus handles strings received by the SS command by setting a flag in the common structure.

When an orientation sensor is selected, a thread is created that sets up the needed communication for the sensor, then constantly polls the sensor for a new heading, which is stored in the common structure. Only one sensor is allowed to be active at a time, and thus the common heading storage location is never overwritten by more than one sensor. This configuration makes adding orientation sensors to the software easy. All that is needed is to add an option to the GUI, and then write the thread for the new sensor. The thread just needs to set up the communication with the sensor, then write to the common structure.

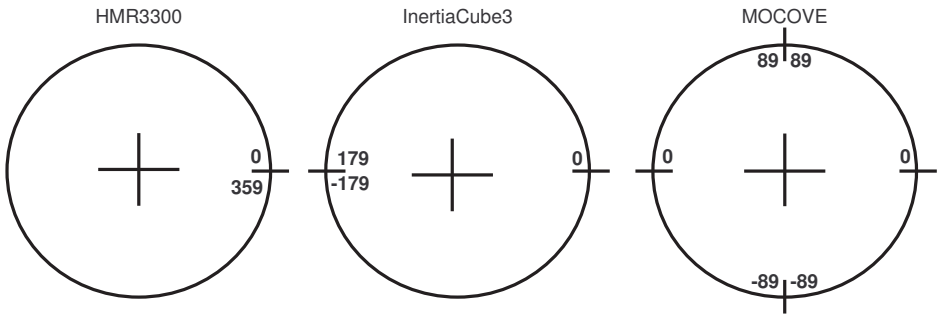

Figure 2.5: Circles showing the range of each orientation sensor.

The rest of the application remains untouched as it just reads from the common structure, and does not care what device set the heading.

When the ”Start Motor” button is pressed, it first spawns a recording thread if an orientation sensor is active, and reads the currently selected routine. The motor is set up with the proper acceleration and velocity, and the routine is run. When the routine completes, both the recording thread and the motor thread are terminated.

The recording thread creates a comma separated value (.csv) file, and then polls the common structure for the motor position and the headings from the orientation sensor. As mentioned earlier, these are being written by other parts of the program. The output file ends up containing the name of the sensor recorded, the routine chosen, the motor positions, the sensor headings, the errors, absolute errors, average error, variance, standard deviation, the sampling interval, and any settings that the particular orientation sensor may have. Careful attention has to be taken as the motor returns its position in steps, such as 108000 (108000 steps @ 36000 steps/rev = 3 rev = 1080 degrees), and each sensor has its own range which is depicted in Figure 2.5.

The motor position is converted to the correct range to match the sensor being recorded. Also, care is taken during the calculation of the statistics, such as the error and variance, to account for the range of each sensor. For example, if the motor position is 2 degrees, and

the Honeywell compass returns a heading of 359 degrees, the error calculated is 3 degrees instead of the straight subtraction of 357 degrees.

To make viewing and comparing the recorded data easier, a synchronize feature was written that set the motor position to zero when the orientation sensor measured zero. Due to the latency and inaccuracy of the sensors, this is done by taking a reading from the sensor, and moving the motor the negative of half this amount, then pausing for a moment. This is repeated until a measurement between 0 and .5 degrees is produced from the sensor. Once the sensor's heading is close to zero, the position counter on the motor is reset, the synchronization is complete, and the benchmark routine is started.

2.3 Benchmark routines

For this research, we wanted to investigate how the orientation sensors performed under different types of movements, as this would help in choosing the right sensor for a particular application. For example, an orientation sensor used as a digital compass for a driver in a car would likely encounter smooth, slow angular rotations, while an orientation sensor mounted on a person would likely see jerky, fast angular rotations. With this in mind, 10 motor routines were developed that each focused on a certain type of motion. Table 2.1 summarizes the routines in terms of acceleration and velocity. The acceleration listed is the initial acceleration used to reach the constant velocity listed. Only the smooth-varied routine and the jerky-varied routines have a constant acceleration throughout the routine, which is why the velocity is listed as increasing instead of constant.

The smooth-slow routine consists of three non-stop revolutions in one direction. This routine was designed to be the easiest, and thus was expected to produce the best possible accuracy. The magnetometer based sensors were expected to excel while the other technologies would lag behind the actual orientation due to the high sensitivity needed to measure the low inertial changes. The smooth-fast routine also consisted of three non-stop

Routine	Acceleration ($\frac{rev}{s^2}$)	Velocity ($\frac{rev}{s}$)	Modeled Motion
Smooth-slow	1	0.1	Vehicle digital compass
Smooth-fast	20	3	Spinning wheel / winch
Smooth-varied	1	increasing (40 max)	Accelerating wheel
Jerky-slow	5	0.1	Body movements
Jerky-fast	20	3	Arm/head movements
Jerky-varied	2	increasing (20 max)	Arm/head movements
Oscillating	10	3	Pendulum
Intermittent	20	20	Conveyor belt
Whip with pause	20	5	mobile robot wheels
12-minute	5	0.1	Long-term measurement

Table 2.1: The accelerations and velocities used for the benchmark routines.

revolutions in one direction, but did so at a higher velocity to see which sensor could handle a higher change in angular rotation. The smooth-varied routine again was three non-stop revolutions in one direction, but at an increasing angular velocity as opposed to the constant velocity of the previous two routines. Beyond just observing how a sensor would perform under constant acceleration, this routine allowed us to observe the degradation in accuracy as the velocity increased. This gave us an idea of the max angular velocity each sensor could handle. Due to our previous work with the Honeywell device, we expected it to lag behind the actual orientation in the smooth-fast routine, but catch up to the correct position once the platform stopped. The other two sensors were expected to keep up with the actual orientation better, but drift to an incorrect position by the time the platform stopped.

Technically, jerk is the derivative of acceleration, and to produce high jerk in this research, we use abrupt changes in direction, paired with high accelerations. All three of the jerky routines were set up with the same platform path, but one routine was slow, one fast, and one with an increasing angular velocity. Since accelerometers and gyroscopes have a lower latency and faster update rate, these routines, with their sharp changes in directions, should bring out the strength of the inertial sensors, and hinder the magnetometers. The path of the jerky routines can be seen in Figure 2.6.

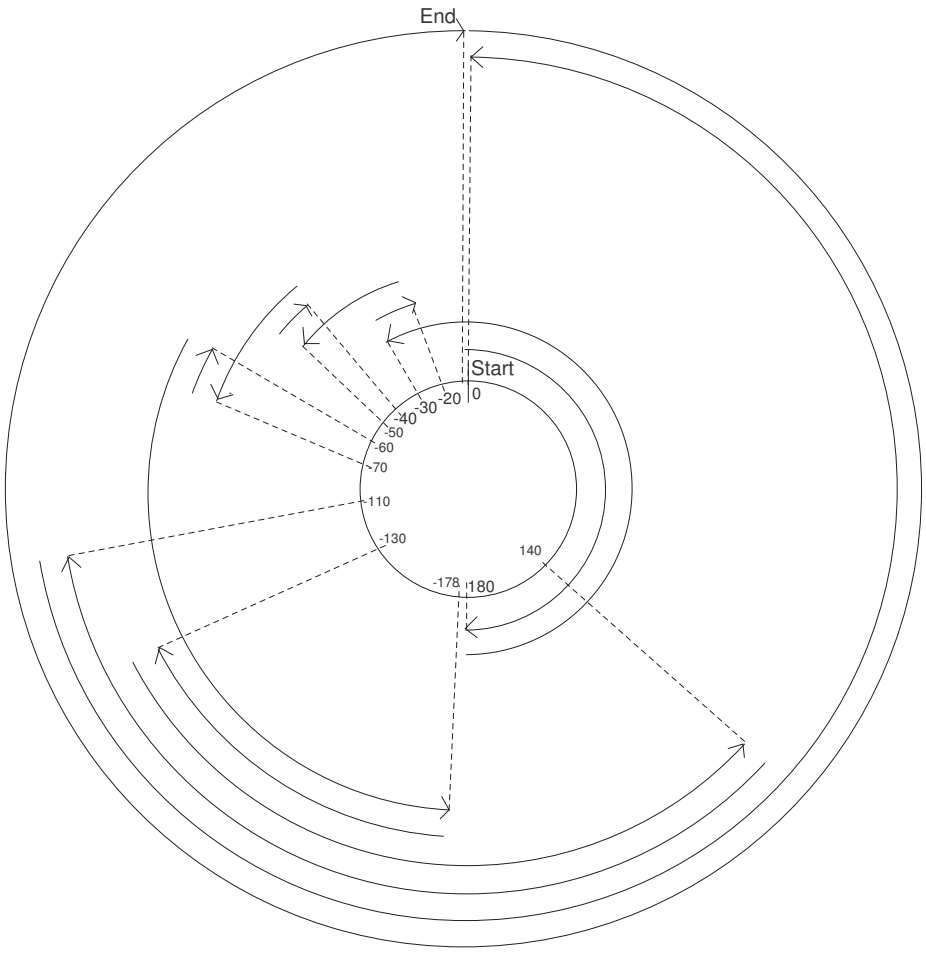

Figure 2.6: The path of the platform during a jerky routine.

The oscillating routine follows an under-damped oscillation. The platform swings back and forth increasing the distance of the swing by a degree every half period. It was expected that the large number of abrupt direction changes, along with the increasing time between changes, would cause a significant accumulation of drift in sensors implemented using just accelerometers and/or gyroscopes. Magnetometers on the other hand were expected to not be able to keep up with the abrupt direction changes, but would not drift over time.

The intermittent routine was created to model the tracking of a conveyor belt wheel that pulls a part along, stops for a moment to allow for assembly, and then pulls another part

up. We were unsure what this routine would bring out in our sensors, but included it in our testing since it models a commonly found motion.

The whip with pause routine spins the platform around as fast as possible 1080 degrees (3 revolutions), pauses for 3 seconds, then backwards 1080 degrees, and pauses for 3 seconds. It then spins the platform around as fast as possible for 720 degrees (2 revolutions, pauses for 3 seconds, then backwards for 720 degrees, and pauses for 3 seconds. This pattern continues for spins of size: 360 degrees, 180 degrees, 90 degrees, 45 degrees, and 20 degrees. The goal here was to confuse the sensor, then observe whether it would correct itself during the pause, and if so, by how much. It was also designed to show how big of a spin was needed to confuse the sensor. It was expected that the Honeywell sensor would instantly be at a correct orientation when paused, but the InertiaCube3, due to its filtering design, would be off when paused, but would correct itself during the pause. The MOCOVE device was expected to just drift, and not correct itself.

Lastly, the 12 minute routine is just a collection of jerky-slow movements that take roughly 12 minutes to complete. This routine was designed to have a slow enough angular rotation so that all of the sensors would be able to get good headings. It was used to see how much error each sensor would accumulate over a longer running time. The MOCOVE was expected to drift to the point where it would not return valid data. We expected the InertiaCube3 to either stay accurate due to the slow movements, or drift to non-valid data due to the lack of pauses in the routine where the filtering algorithm for the sensor could correct the position.

Along with the benchmark routines, the smooth routine was run at increasing velocities to find the max velocity that each sensor could measure accurately. This made the smooth-fast routine a bit redundant, but was added after viewing the poor performance of a sensor running the smooth-fast routine. Since the device failed during the routine, we wondered at what speed the device was reliable, and thus this test was built.

Chapter 3

Results

The results for determining the final implementation of the physical platform are covered in the first section. Next, an example of the data recorded from a benchmark routine will be walked through to highlight the information that is collected, and to explain how the overall results are found. Next, the results of the benchmark routines run with the orientation sensors are shown. Lastly, the results from the max velocity experiment are reviewed.

3.1 Platform test

After observing patterns in the error of the HMR3300 and the InertiaCube3 during initial testing of the platform, a test of the accuracy and repeatability of the stepper motor was done to eliminate its movement as a possible source of noise. The simple physical test involved cutting a small hole in the center of a sheet of paper, and then mounting it at the base of the platform with the motor's shaft through the hole. A pen with a retractable tip was attached to a 90 degree mounting bracket which was then attached to the platform itself. This allowed a mark to be made on the paper simply by clicking the pen. Software was written that allowed a user to tell the motor to turn exactly 20 degrees. The motor was turned a total of 360 degrees, and at each 20 degree interval, the pen was clicked to mark the position of the platform. A ruler and protractor were then used to calculate the

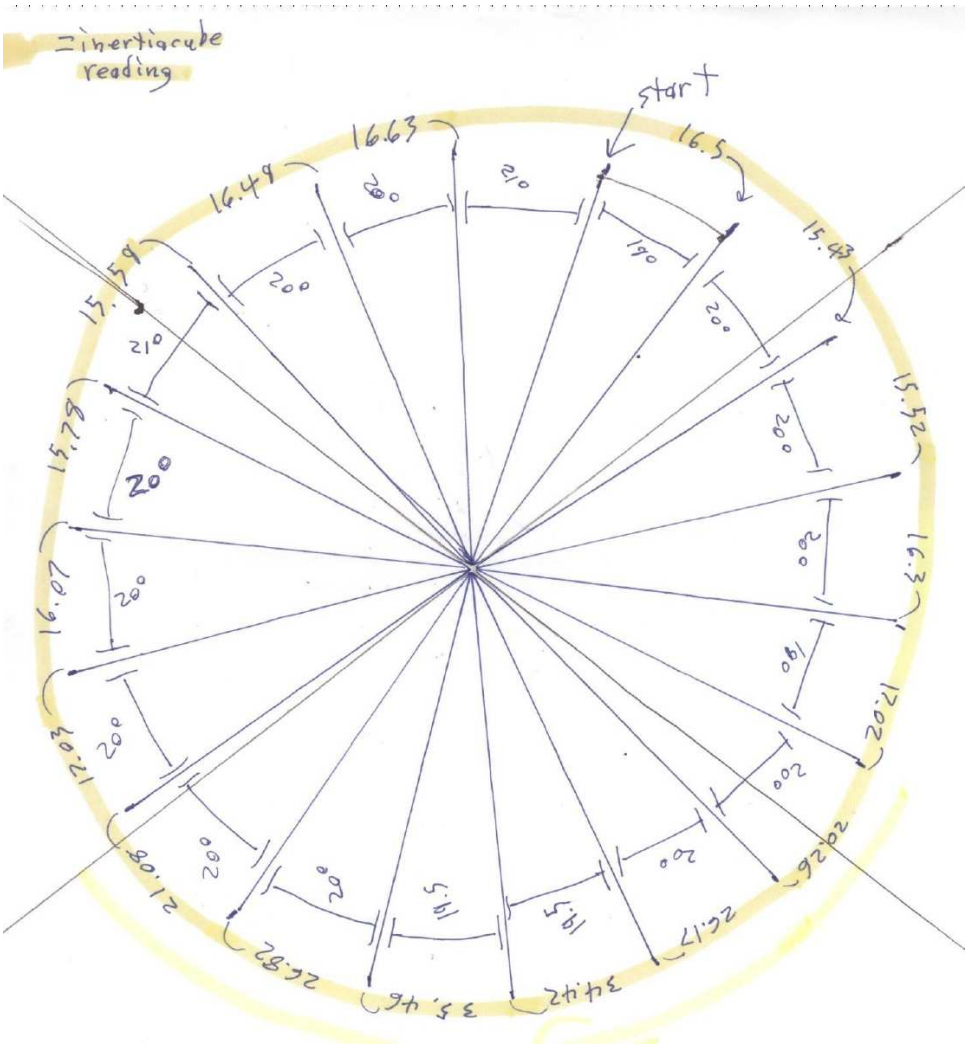

Figure 3.1: The results of the physical stepper motor test.

number of degrees moved in each iteration, and can be seen in Figure 3.1. As can be seen, the motor shows a consistent movement of 20 degrees. The few 19.5 degree and 21 degree measurements can be attributed to the inaccuracy of the measurement method. The test was looking for errors on the scale of 15+ degrees, and thus the test was not designed to be accurate at the sub-degree level. Figure 3.1 also shows the measurements of the InertiaCube3 at each 20 degree step, which sparked the physical test. For a third of the revolution, the InertiaCube3 consistently measured high, and for the rest of the revolution,

it measured low. This pattern was repeatable, and was found to be due to a combination of the platform configuration, the level of the platform, and nearby EMF's.

As mentioned in section 2.1 several different platform configurations were investigated before deciding on the final configuration. Figure 3.2 shows the average error of the InertiaCube3 spun 3 revolutions using different platform configurations. Three different height cardboard boxes were used to observe whether moving the orientation sensors a distance away from the motor and shaft would have an effect on the measurements. It was originally thought either the metal of the shaft, or the EMF from the motor could be enough to disturb the magnetometers. The results of the tests show that the 6" shaft coupled directly to the stepper motor, along with the 4" wooden platform performed better than any other configuration.

3.2 Routine results

The benchmark routines mentioned in section 2.3 were run on each of the 3 sensors mentioned, and the results were recorded. This section divides the results up by the type of motion used in each routine. Each subsection outlines one or more of the trials run with the type of motion routine, and then shows the overall results from the testing.

3.2.1 Smooth routines

Figure 3.3 shows a graph of the path of the motor during the smooth-slow routine compared to the measured heading from the Honeywell sensor. As can be seen, the device is accurate around the 0 degree position, but is off by approximately 20 degrees at the 180 degree position. The error plot seen in Figure 3.4 further shows this cyclic error. The error plot also shows the drift that occurs when the platform changes direction.

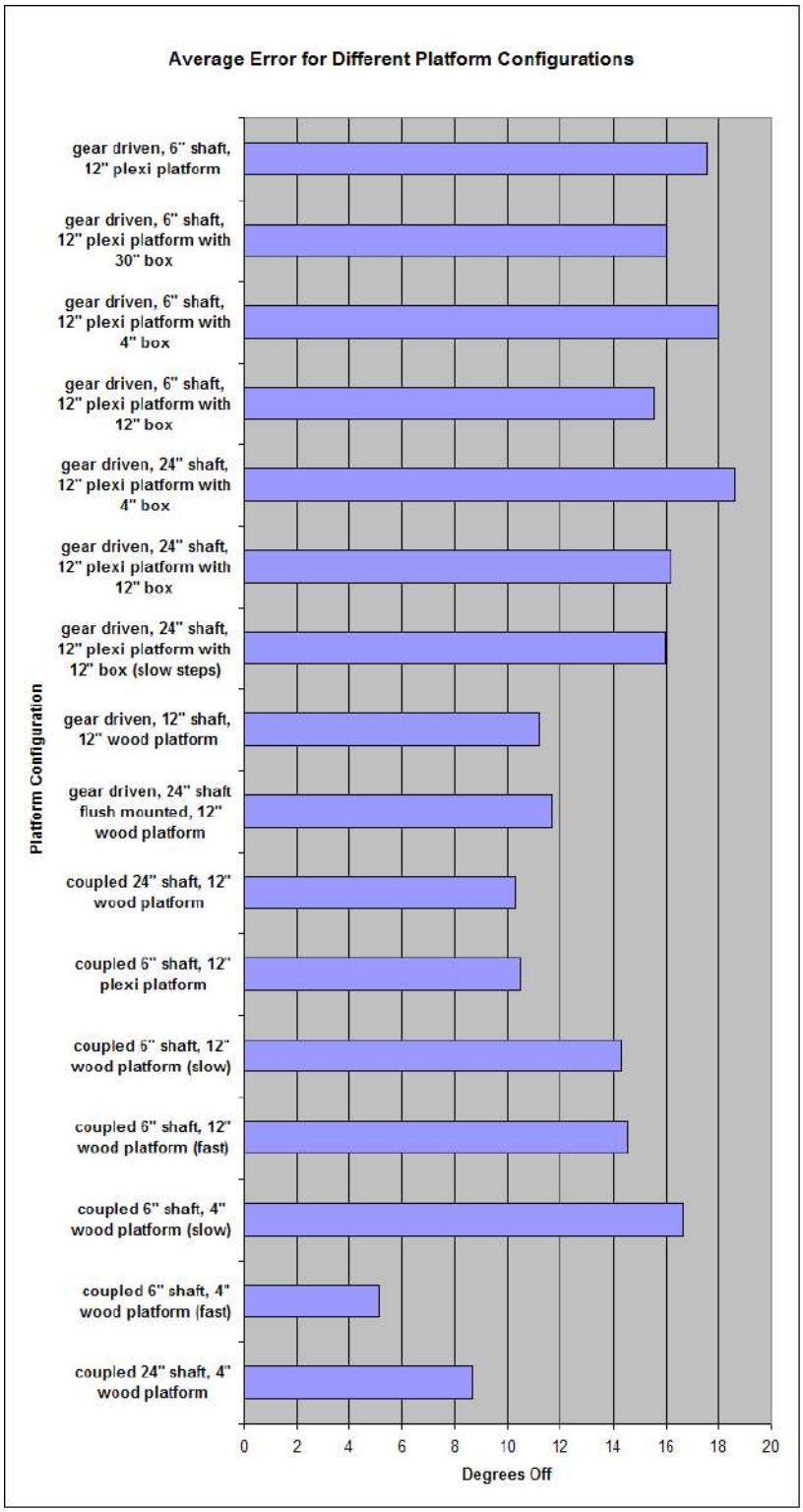

Figure 3.2: The average error of the InertiaCube3 measured on different platforms.

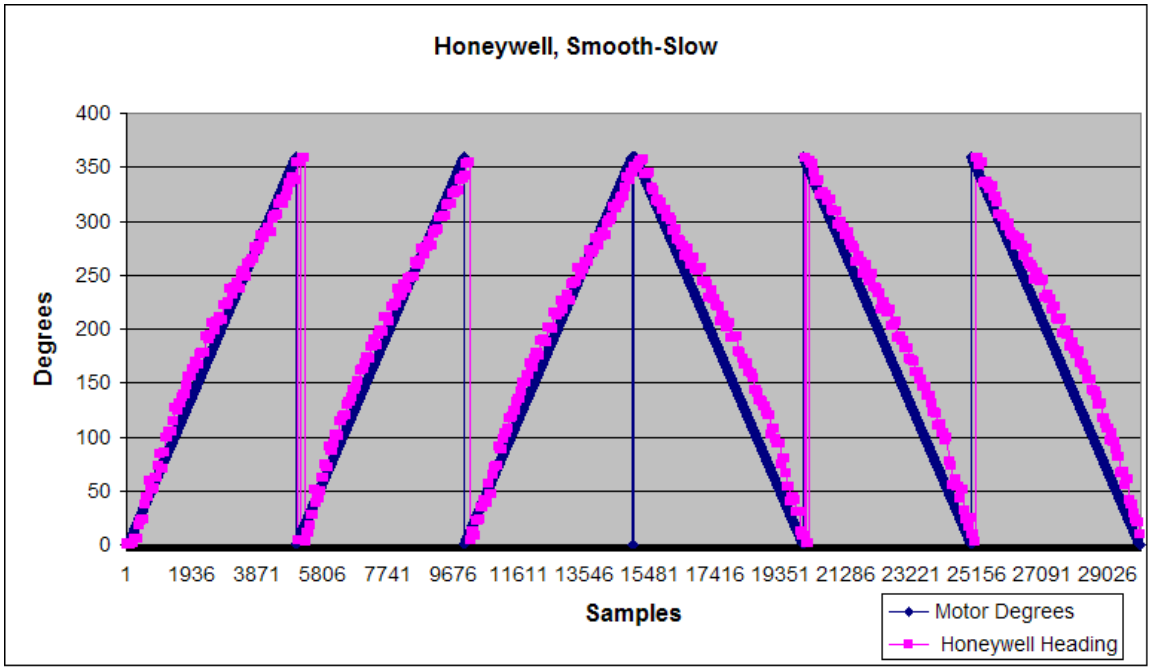

Figure 3.3: The headings plot for the Honeywell sensor during the smooth-slow routine.

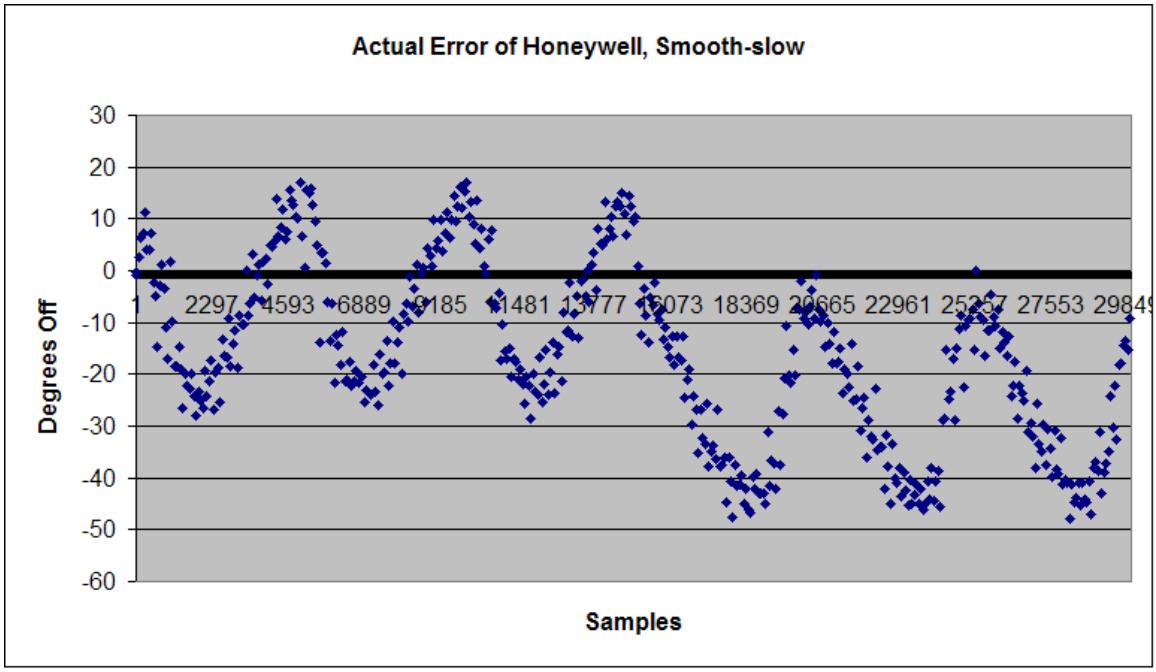

Figure 3.4: The error plot for the Honeywell sensor during the smooth-slow routine.

Figure 3.5 shows how badly both the MOCOVE and the Honeywell sensors perform compared to the InertiaCube3. It is obvious that only the InertiaCube3 can accurately track the larger angular velocities in the smooth-fast routine.

Figure 3.6 shows the average error for each sensor running the smooth routines. Care must be taken when viewing these graphs because when the error is large, it is also a function of the range of the sensor. For example, as mentioned, the MOCOVE has a range of -90 degrees to 90 degrees. When the absolute error is calculated, a max error of 90 degrees is possible. Now say for example, that the heading returned is random, with a Gaussian distribution. This would create a Gaussian random error between 0 and 90 with a mean of 45 degrees. That means the MOCOVE sensor with an average error of 45 degrees might as well be a random number generator as opposed to an orientation sensor. Using the same idea, the InertiaCube3 has a range between -180 and 180, which means a max error of 180 is possible. The mean of the Gaussian error distribution would be 90 degrees, and would be the average error if the sensor just generated random numbers. Lastly, the Honeywell has a range of 0 to 360, which following the previous logic would again give us 90 degrees for the average error given a random number generator. Therefore, when looking at the graphs, keep in mind that an average error of 45 degrees for the MOCOVE, and 90 degrees for the other two sensors, means that the sensor is totally lost. Also keep in mind that it is possible to do worse than random. As can be seen, the MOCOVE actually performs better than the other two sensors at slow speed, but then fails when the speed is increased. Also, the Honeywell sensor is only able to muster an average error of 19 degrees, which is a far cry from the published 1 degree accuracy.

3.2.2 Jerky routines

Figure 3.7 shows 3 graphs of the orientation sensors recorded while running the jerky-varied routine. It is obvious that the InertiaCube3 is again the only sensor to be able to handle the higher angular velocities. It can be seen that both the MOCOVE and Honeywell

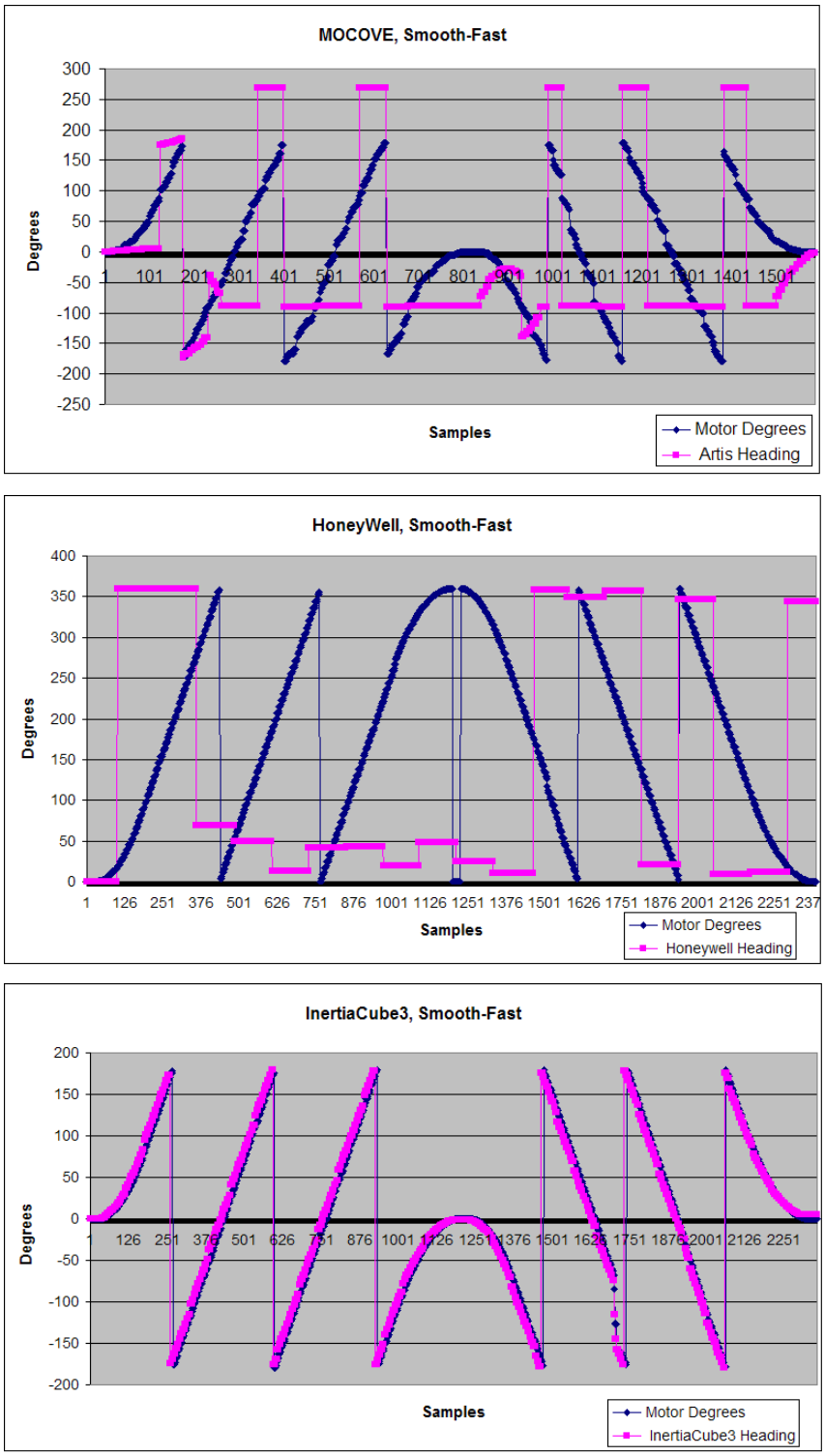

Figure 3.5: The path and recorded headings for the (a) MOCOVE (b) Honeywell (c) InertiaCube3 running the smooth-fast routine.

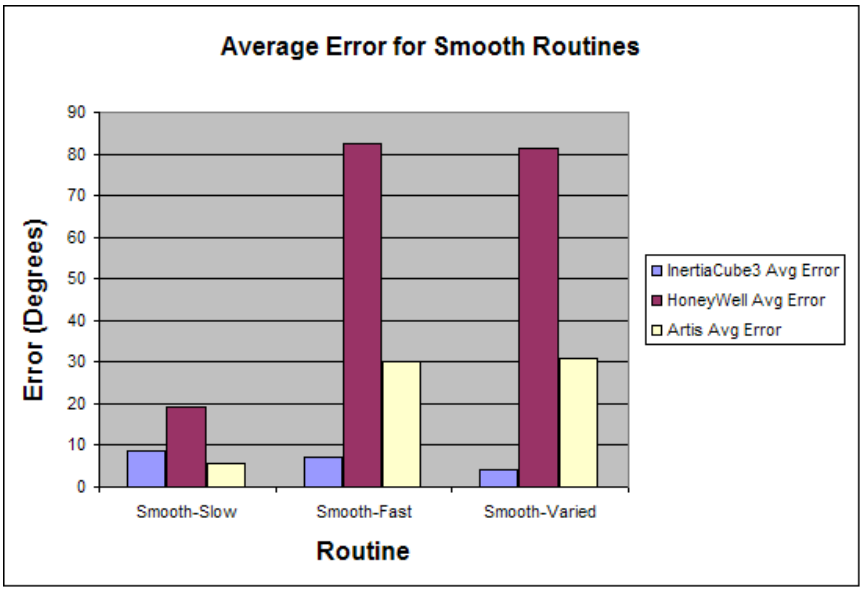

Figure 3.6: The average error calculated for each sensor running the smooth routines.

develop error as the speed of the movement in a particular direction increases. The shorter movements are tracked better since the platform does not have time to develop a lot of speed.

Figure 3.8 shows the results of the three sensors when the jerky routines are run. Again, the MOCOVE sensor does really well at slow speeds, but fails once things speed up. The InertiaCube3 on the other hand does better as the speed increases. It is also more accurate during these jerky routines than it was during the smooth routines. The likely cause is that the smooth routines only have two accelerations, the one that gets the platform spinning, and the one that slows it back down, while the jerky routines are filled with accelerations. As mentioned, the accelerometers are a major part of the InertiaCube3, and by being able to incorporate their measurements into the orientation calculation, a more accurate measurement can be made.

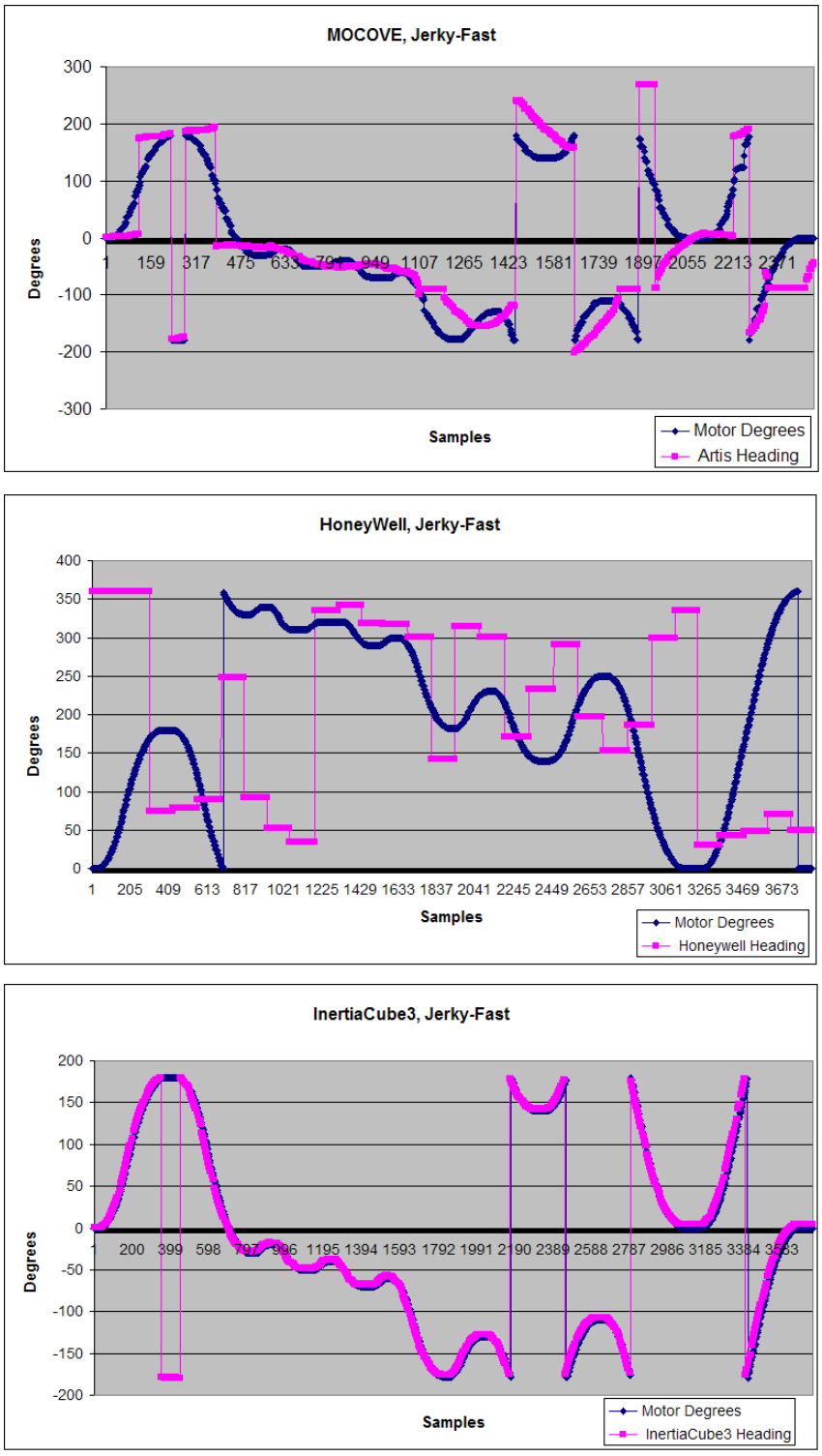

Figure 3.7: The path and recorded headings for the (a) MOCOVE (b) Honeywell (c) InertiaCube3 running the jerky-fast routine.

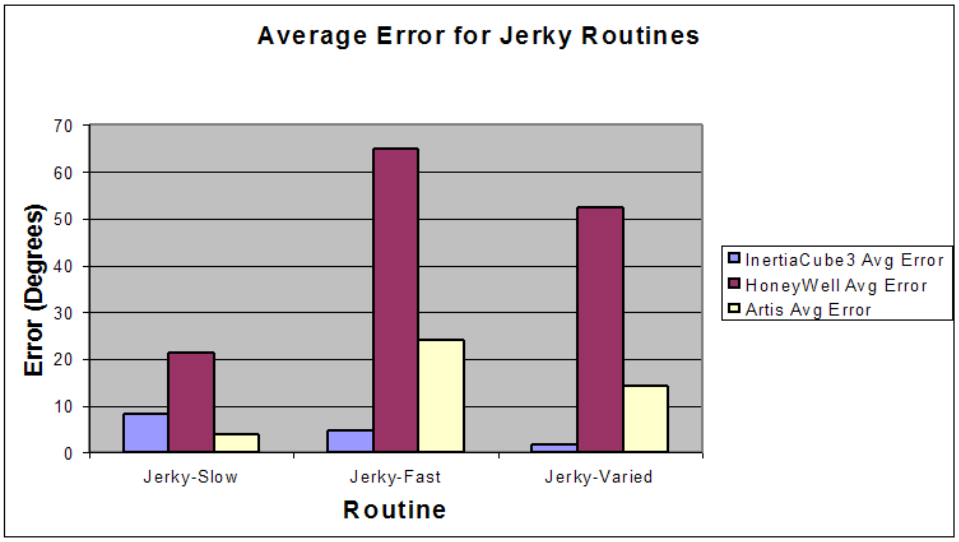

Figure 3.8: The average error calculated for each sensor running the jerky routines.

3.2.3 Other routines

While both the MOCOVE and the Honeywell sensors did not drift over the oscillating routine, they did have a difficult time staying accurate through the direction changes. The InertiaCube3 on the other hand, handled the direction changes, but suffered from drift as expected. Figure 3.9 shows this drift over time.

The intermittent routine did not produce any new results as again, the MOCOVE out performed the other sensors due to the low angular velocities experienced in the routine.

The whip with a pause routine produced both expected and unexpected results. Figure 3.10 highlights the expected operation of the filtering algorithm used in the InertiaCube3. The filter only corrects the orientation when the sensor is at rest. Unexpectedly, the MOCOVE sensor as seen in Figure 3.11 not only does not drift over time, but also took very little time to return to the correct heading after being whipped. Also unexpected was the drift that can be seen in Figure 3.12 by the Honeywell sensor. Since the sensor is based on a magnetometer, the sensor should not drift. No explanation is given for this result.

Figure 3.13 shows the average error of the sensors run on the remaining routines.

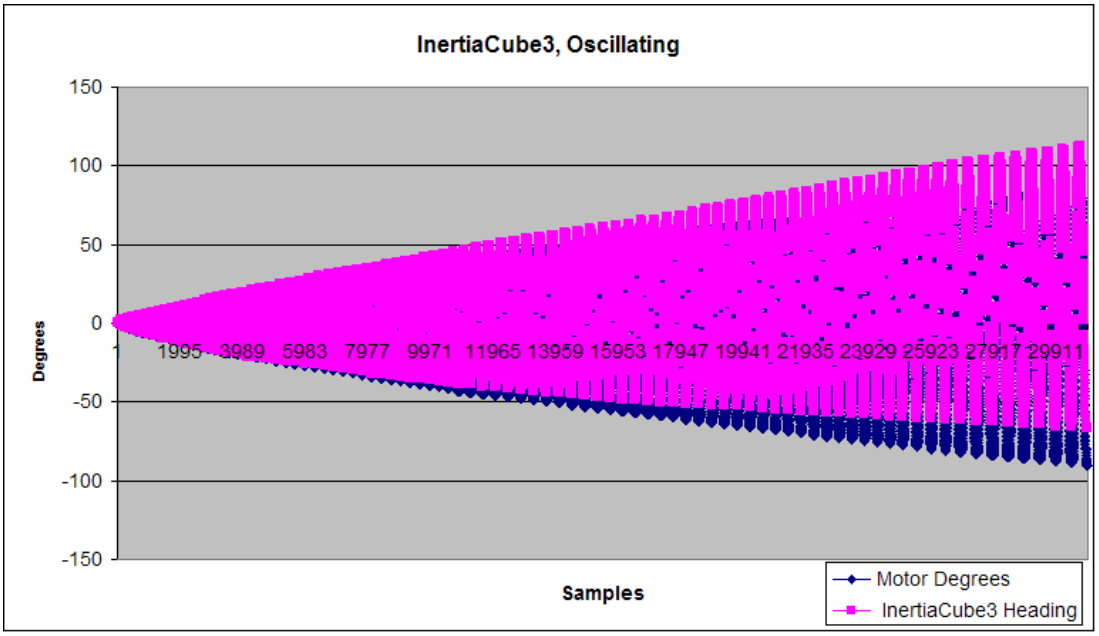

Figure 3.9: The path of the oscillating routine along with the headings recorded from the InertiaCube3.

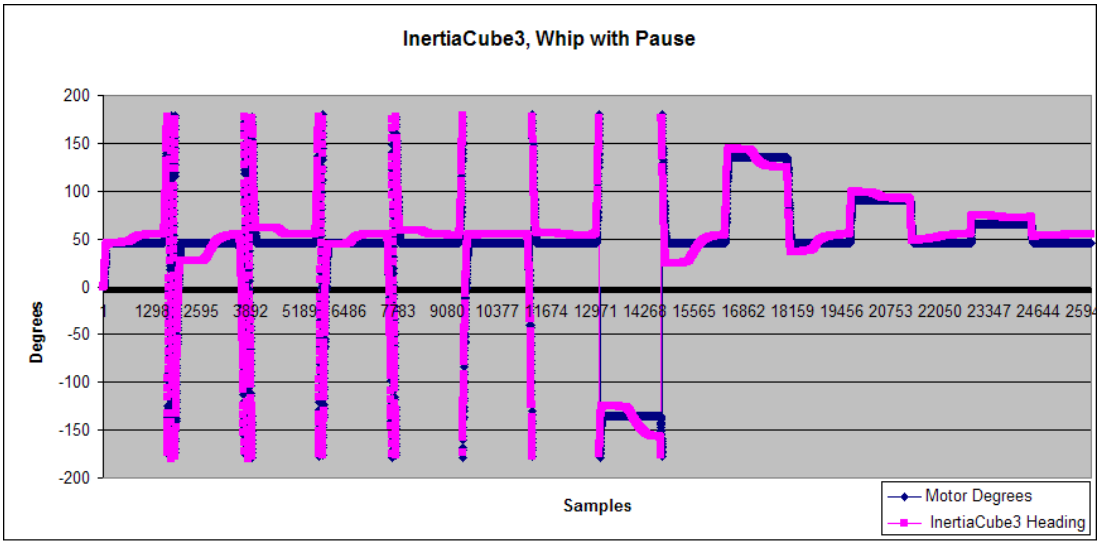

Figure 3.10: The path of the whip with pause routine along with the headings recorded from the InertiaCube3.

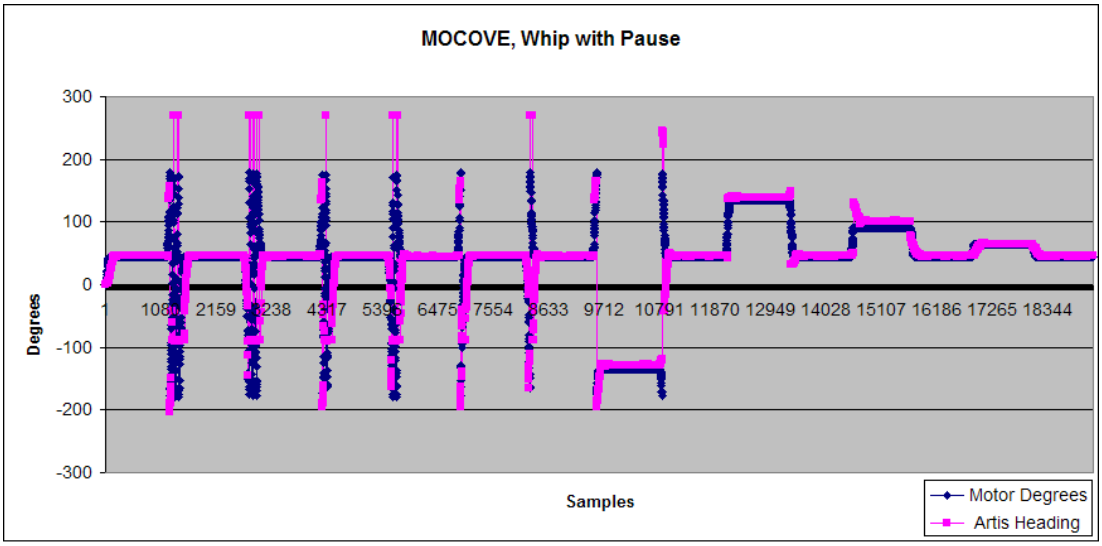

Figure 3.11: The path of the whip with pause routine along with the headings recorded from the MOCOVE.

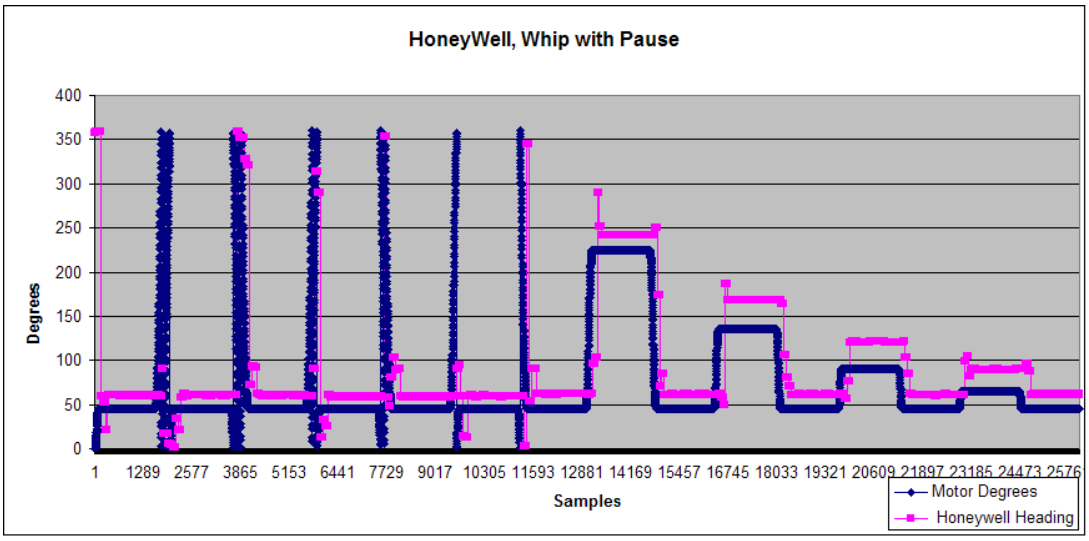

Figure 3.12: The path of the whip with pause routine along with the headings recorded from the HMR3300.

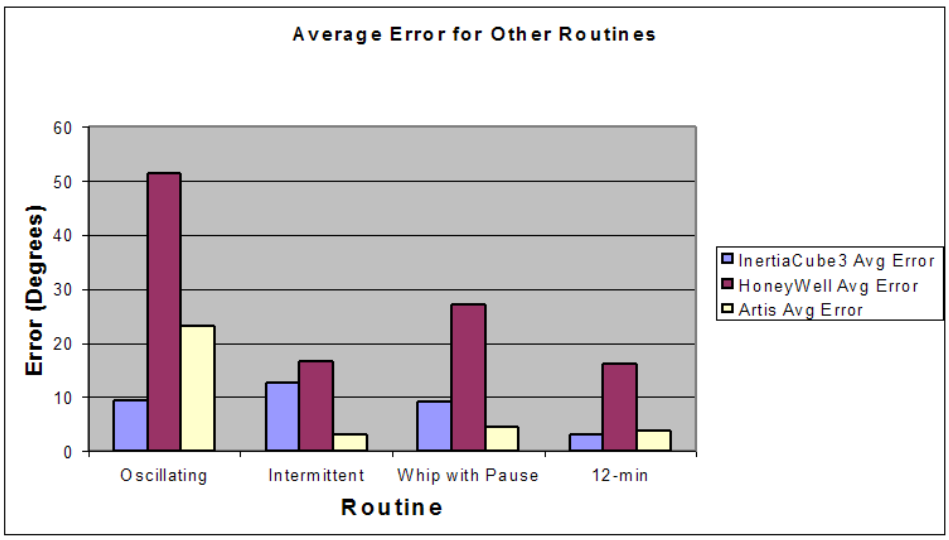

Figure 3.13: The average error calculated for each sensor running the other routines.

3.3 Max velocity

After observing the results of the routine tests, we were interested in knowing how fast each sensor could be spun and still produce a relatively accurate measurement. Each sensor was spun 3 revolutions with an initial acceleration of 20 $\frac{rev}{s^2}$, and varying velocities. Figure 3.14 summarizes the results of the test. As previously mentioned, the graph is a little misleading due to the range of each sensor. To further show this, Figure 3.15 shows the results of spinning the MOCOVE at 3 $\frac{rev}{sec}$. It should be obvious that the sensor is overwhelmed and confused, and rarely has a valid position measurement. The important thing to observe in Figure 3.14 is not the height of the average error, but rather where the slope increases for each sensor. This is the point at which the sensors are starting to fail. It shows that both the MOCOVE and the Honeywell are accurate to about .5 $\frac{rev}{sec}$, while the InertiaCube3 is accurate all the way up to 5 $\frac{rev}{sec}$. It is also interesting to note that, in accordance with the routine tests performed, at slow speeds the cheapest sensor, the MOCOVE, outperformed both of the other sensors.

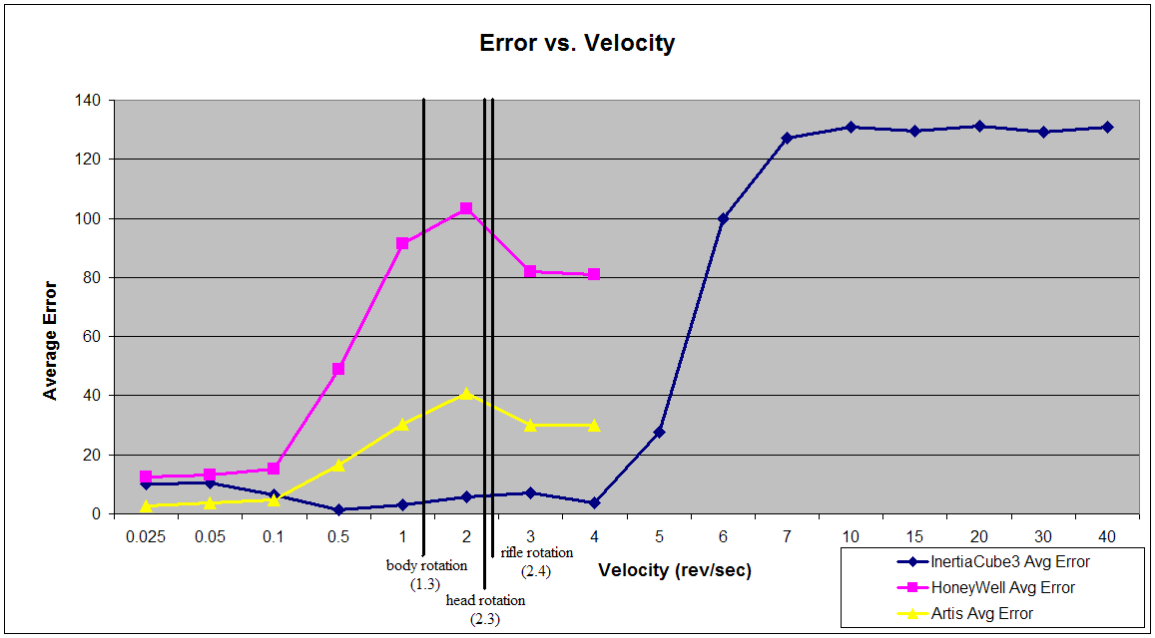

Figure 3.14: The error vs. velocity plot for all three orientation sensors.

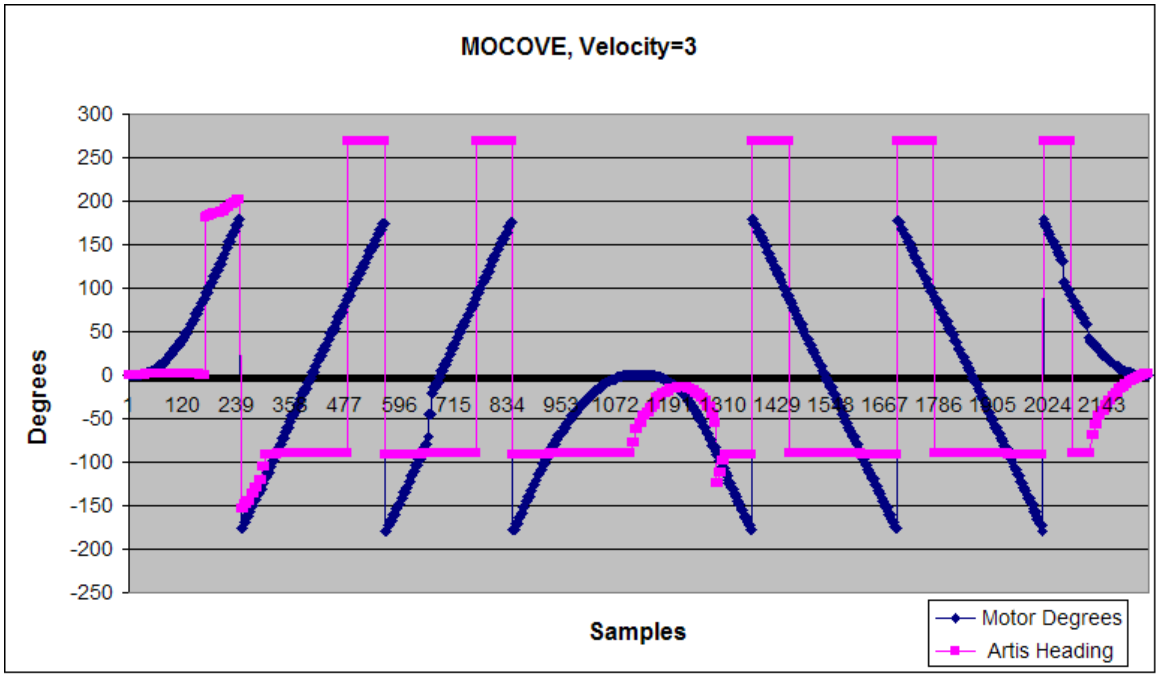

Figure 3.15: The MOCOVE orientation center spun 3 revolutions at 3 $\frac{rev}{sec}$.

The graph also shows rough estimates of how fast people can spin, turn their head, and swing a rifle. These speeds were calculated by having a participant, with the InertiaCube3 mounted, perform the particular movement measured. The derivative of the recorded positions was calculated, and the max angular rate was found. The velocity verses error test could really help consumers fit an orientation sensor to their particular application.

Chapter 4

Conclusions

This paper presented a way to test a MEMS orientation sensor under different types of motion in an effort to better understand the performance characteristics of the sensor. These sensors are produced and sold with data sheets that outline their performance, but lack the conditions under which the testing takes place. In this research, a testing apparatus was developed, and testing routines were designed to evaluate the different characteristics of orientation sensors under different motion conditions. Three orientation sensors, each in a different price range, were evaluated with the benchmark suite. The developed testing apparatus is a turntable that can precisely spin an orientation sensor via a stepper motor, and can record its exact orientation along with the heading read from an orientation device. Sets of movements we call benchmark routines were designed and implemented to test different properties of the sensors.

The results of this research show that the turntable performs correctly, and is a viable way to test orientation sensors. The three orientation sensors tested show that as expected sensors with similar data sheet specifications perform very differently. Under smooth, low-speed conditions (.1 $\frac{rev}{s}$), the cheaper MOCOVE orientation sensor outperformed the more expensive InertiaCube3 orientation sensor. However, as the angular velocity increased, the error of the MOCOVE quickly increased. The error of the Honeywell HMR3300 increased

as well. Only the InertiaCube3 could accurately measure quicker angular velocities (up to 5-6 $\frac{rev}{s}$) commonly found when tracking the movements of human head and arm movements. Also, different amounts of lag and accuracy were uncovered by different motion routines. Table 4.1 summarizes which sensor would be a good choice for each type of motion. The average error of a sensor must be under 10 degrees to be considered accurate

Routine	**Honeywell**	**InertiaCube3**	**MOCOVE**	**Modeled Motion**
Smooth-slow		X	X	Vehicle digital compass
Smooth-fast		X		Spinning wheel / winch
Smooth-varied		X		Accelerating wheel
Jerky-slow		X	X	Body movements
Jerky-fast		X		Arm/head movements
Jerky-varied		X		Arm/head movements
Oscillating				Pendulum
Intermittent			X	Conveyor belt
Whip with pause		X	X	mobile robot wheels
12-minute		X	X	Long-term measurement

Table 4.1: A summary of which sensor is applicable for each type of motion.

enough to work for an application. Overall, this research shows the need for such a benchmark suite, and will aid in the process of choosing an orientation sensor for a particular application.

Bibliography

[1] C. Aimone, J. Fung, and S. Mann. An eyetap video-based featureless projective motion estimation assisted by gyroscopic tracking for wearable computer mediated reality. *Personal and Ubiquitous Computing*, 7(5):236–248, Oct 2003.

[2] W. Ang, P. Khosla, and C. Riviere. Kalman filtering for real-time orientation tracking of handheld microsurgical instrument. *proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3:2574–2580, Oct 2004.

[3] Applied Motion Products, 404 Westridge Drive, Watsonville, CA 95076. *Si2035 Hardware Manual*, 2004.

[4] B. Avery, B. Thomas, J. Velikovsky, and W. Piekarski. Outdoor augmented reality gaming on five dollars a day. *proceedings of the Sixth Australasian conference on User interface*, 40:79–88, 2005.

[5] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through hmd. Master's thesis, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC 27599-3175, 1994.

[6] E. Bachmann and X. Yun. Design and implementation of marg sensors for 3-dof orientation measurement of rigid bodies. *proceedings of 2003 IEEE International Conference on Robotics and Automation*, 1:1171–1178, Sept 2003.

[7] E. Bachmann, X. Yun, and C. Peterson. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. *proceedings of 2004 IEEE International Conference on Robotics and Automation*, 2:1115–1122, April 2004.

[8] J. Bernstein. An overview of mems inertial sensing technology. Available on: `http://www.sensorsmag.com/articles/0203/14/`, February 1, 2003.

[9] J. Bers. A body model server for human motion capture and representation. *Presence: Teleoperators and Virtual Environments*, 5(4):381–392, 1996.

[10] Honeywell, 12001 State Highway 55, Plymouth, MN 55441. *1- and 2-Axis Magnetic Sensors*, 2002.

[11] D. Keymeulen, W. Fink, M. Ferguson, C. Peay, B. Oks, R. Terrile, and K. Yee. Evolutionary computation applied to the tuning of mems gyroscopes. *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 927–932, 2005.

[12] S. Lee, G. Nam, J. Chae, H. Kim, and A. Drake. Two-dimensional position detection system with mems accelerometer for mouse applications. Master's thesis, University of Michigan, Ann Arbor, MI 48109-2122, 2001.

[13] J. Marti, K. Brendley, P. DiZio, and J. Lackner. Motion-coupled virtual environment (mocove). Available on: `http://www.artisllc.com/solutions/bio/mocove/phase1/index.html`, 2000.

[14] E. Muth, A. Hoover, and M. Loughry. Developing an augmented cognition sensor for the operational environment: the wearable arousal meter. *proceedings of the 1st International Conference on Augmented Cognition*, 2005.

[15] W. Putman and B. Knapp. Input/data acquisition system design for human computer interfacing. Available on: `http://ccrma.stanford.edu/CCRMA/Courses/252/sensors/sensors.html`, Oct 1996.

[16] K. Waller, J. Luck, A. Hoover, and E. Muth. A trackable laser tag system. *proceedings of WORLDCOMP'06 The 2006 International Conference on Computer Games*, June 2006.

[17] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Collucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. Master's thesis, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC 27599-3175, 1999.

[18] Wikipedia. Giant magnetoresistive effect. Available on: `http://en.wikipedia.org/wiki/Giant_magnetoresistive_effect`, 2006. Online; Last modified Oct. 18, 2006.

[19] D. Wormell and E. Foxlin. Advancements in 3d interactive devices for virtual environments. *ACM International Conference Preceeding Series; Proceedings of the Eurographics Workshop on Virtual Environments*, 39:47–56, 2003.

[20] J. Wu and M. Ouhyoung. A 3d tracking experiment on latency and its compensation methods in virtual environments. Master's thesis, National Taiwan University, Taipei, Taiwan. R.O.C. 106, 1995.

[21] S. You and U. Neumann. Orientation tracking for outdoor augmented reality registration. *proceedings of 8th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Digital Interactive Media (WSCG 2000)*, page 38, Dec 1999.

[22] X. Yun, E. Bachmann, A. Kavousanakis, F. Yildiz, and R. McGhee. Design and implementation of the marg human body motion tracking system. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:625–630, Oct 2004.