Incorporation of a Foot-Mounted IMU for Multi-Sensor Pedestrian Navigation

by

Daniel Pierce

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 7, 2016

Keywords: sensor fusion, inertial navigation, human gait analysis, foot-mounted IMU, pedestrian navigation, computer vision, relative state measurements

Approved by

David Bevly, Chair, Professor of Mechanical Engineering
John Y. Hung, Professor of Electrical and Computer Engineering
Dan Marghitu, Professor of Mechanical Engineering

Abstract

This thesis deals with pedestrian localization by way of multi-sensor fusion. A special focus is made on the use of a foot-mounted inertial measurement unit (IMU) and its incorporation into other pedestrian navigation systems. Mounting to the foot location is preferred due to the anticipated dynamics of the foot during normal walking motion, but poses difficulty when fusing with other body-worn sensors. One of the main challenges lies in the non-rigid relation between navigation sources. This thesis approaches the problem by characterizing the human gait during walking motion and detecting instances in which spatial relations can be made. Navigational information of a pedestrian is often provided as a relative state measurement, such as step length, walking pace, rate of turn, etc. The fusion of multiple relative state measurements is non-trivial and requires manipulation of standard sensor fusion frameworks. Therefore, this thesis discusses certain frameworks for processing multiple relative state measurements in detail.

Two algorithms are presented for fusing a foot-mounted IMU with other body-worn relative state measurement systems. The first algorithm operates in a cascade architecture, which allows for easy implementation while still showing promising results. Drawbacks to the cascade architecture are discussed, and a second, centralized architecture is presented that addresses these issues. To serve as an example, a particular problem is addressed: fusing measurements from the foot-mounted IMU with a chest-mounted visual odometry system. By analyzing the human gait through the raw IMU signals, a relation is made between the position and orientation of the chest-mounted camera and the foot-mounted IMU. Using a high precision motion capture system, the human gait is analyzed and motion profiles of a typical step are calculated. Using these motion profiles as a basis, a simulation environment is developed to replicate visual odometry and foot-mounted IMU measurements

and the navigation algorithms are applied to simulated data. Conclusions are drawn from simulation on the effectiveness of the respective algorithms and experimental data validates these findings. Experimental data is collected with an open source stereo visual odometry system and a MEMS grade IMU. In post-process, the experimental data is fed through the developed algorithms, and the results are compared to those found in simulation. The work presented in this thesis will inform the reader of the characteristics of a foot-mounted IMU solution and establish a methodology for fusing with any general pedestrian navigation device.

Acknowledgments

The research presented in this thesis was made possible through funding provided by the Federal Highway Administration. The project sponsored by the FHWA inspired me to contribute to the field of pedestrian navigation with the goal of improving the lives of blind and visually impaired individuals. Thanks to the generous sponsorship of the FHWA, my work towards the master's degree has been highly rewarding.

I would like to thank my fellow GAVLAB members. I am grateful to have the opportunity to work in a lab filled with such camaraderie. The GAVLAB members provided consistent encouragement and support throughout my time thus far in the lab. As I climbed my way up the steep learning curve of working in the GAVLAB, the experienced lab members were extremely helpful (and patient) along the way, especially Jordan Britt, Scott Martin, and Chris Rose. I would also like to thank Grant Apperson and Robert Cofield, the two GAVLAB members who have been with me from the start. We have always been there for each other to bounce ideas back and forth and we've gained some lasting memories from late nights spent in the lab. A special thanks goes out to those who helped review my thesis including Cody Salmon, Brently Nelson, and Andrew Eick.

I would also like to thank Dr. David Bevly, who has had the most direct impact on my graduate career. I am honored to have Dr. Bevly as my advisor throughout graduate school. He has a way of engaging my curiosity and has motivated me to achieve more than I ever thought was possible. Until receiving Dr. Bevly's encouragement, I had not considered pursuing a graduate degree, which has turned out to be one of the most rewarding (and challenging) experiences in my life so far.

My family has been tremendously supportive throughout my master's program. Throughout my life they have taught me values that have proven to be imperative in guiding me

towards my goals. I accredit my hard work and dedication to my father, who has always encouraged me to be a leader in every aspect of my life. I have my mother to thank for being a spiritual guide and encouraging me to have a close relationship with God. I would also like to thank my older brother, Will, who has always been there to lead by example and has supported me through all of the transitions in my life. I am extremely grateful for the support that I have received from each one of my grandparents. I am especially grateful for my grandfather, Charlie Pierce, who was and continues to be a major influence on my life as a mechanical engineer, a family member, and as a man.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

In recent years, much effort has gone into increasing the accuracy and reliability of navigation solutions. The desire for autonomous navigation has been a large motivator behind the increase of research in this field, resulting in unmanned aerial vehicles, unmanned ground vehicles, and more. These autonomous systems require a highly accurate and robust localization solution for feedback to a control algorithm, which calculates inputs for actuating the system. Improvements in navigation technology have opened the possibility for increased accuracy in pedestrian navigation as well. In the case of pedestrian navigation, the control system is replaced with a human subject who responds to the localization solution. For most pedestrian navigation tasks, a highly accurate localization solution is not necessary due to the user's ability to perceive the environment and navigate accordingly. However, certain scenarios exist in which the ability to perceive the environment is diminished and assistance from navigation technology is required. Such scenarios may include assisting in the navigation of visually impaired or coordinating first responders in searching a smoke filled building. Stand-alone Global Navigation Satellite System (GNSS) solutions are not sufficient in most of these cases due to the lack of availability and degraded accuracies in certain environments. To handle the lack of GNSS coverage, a fused sensor solution is proposed. Thanks to the increased availability of wearable technology such as smartphones, smartwatches, and fitness trackers, a large amount of sensors can now be used in this fusion. Often times, wearable technology provides information of the user's motion, such as step count, cadence, step length, etc. By fusing the redundant information of the user's motion, an accurate dead reckoning solution can be realized to navigate in times of GNSS outage.

1.1 Prior Research

The task of ubiquitous pedestrian localization is non-trivial and poses many challenges. While a great deal of progress has been made in GNSS positioning, even in challenged environments, there are many places in which a pedestrian may travel that will not allow for such a solution. Therefore, the method for navigating pedestrians should not be dependent on GNSS availability. Several other approaches are used for pedestrian navigation in prior research, each with their own attributes and setbacks. These methods and their characteristics will be discussed in this section.

1.1.1 Infrastructure-Based Systems

One choice for localization of a pedestrian without depending on GNSS is through infrastructure-based systems such as Wi-Fi, Bluetooth, Radio-frequency identification (RFID), etc. Systems such as Wi-Fi and Bluetooth use measurements of time of flight (ToF) or received signal strength (RSS) to calculate a distance from a transmitting station located inside the building [12] [13]. With three or more range signals from fixed stations of known positions, the position of the user can be calculated through trilateration and multilateration. Such systems suffer from high noise, multipath, and signal propagation delays from walls and other mediums. RFID systems are similar to Wi-Fi and Bluetooth positioning by measuring RSS, but require line of sight to the receiver [14]. Most often, RFID is used to provide information on the user's vicinity. By associating a unique radio frequency of the RFID tags with a specific location, the user can be positioned in the vicinity of this location when that specific radio frequency is received. Incorporating RFID positioning into an estimation framework can be difficult due to the probability distribution associated with such updates. Although the systems mentioned in this section can be very beneficial for indoor localization, the drawback lies in the inherent need for infrastructure. A dependence on infrastructure means high costs when scaling the system and the need for a standardized

communication protocol. The work presented in this thesis does not rely on infrastructure for these reasons.

1.1.2 Proprioceptive Sensors

Proprioceptive sensors are common for pedestrian navigation and can be found in most of today's wearable technologies. These sensors collect information of the user's motion and are generally unaffected by environmental conditions. Examples of proprioceptive sensors include accelerometers, gyroscopes, and step counters. One popular approach using proprioceptive sensors is pedestrian dead reckoning (PDR) using step detection [15]. The PDR algorithm operates by modeling the human gait and calibrating gait parameters with external measurement sources, most commonly GNSS. Step length is then formulated as a function of the gait parameters and inertial signal attributes, such as amplitude and frequency. Upon detecting a new step, the position solution is propagated using the inertial signal, calibrated gait parameters, and an external estimate of the user's heading [5]. These algorithms are shown to be surprisingly accurate, but suffer from the dependence on external measurements. The calibration phase requires the user to have high quality GNSS signal reception or other reliable positioning measurements. Heading information is often provided from a digital compass, which are known to provide biased measurements at times and are extremely susceptible to passing metal objects.

Another approach to pedestrian tracking with proprioceptive sensors is through the foot-mounted inertial measurement unit (IMU) [2] [3] [11]. For this method, a MEMS (Micro Electro Mechanical Systems) grade IMU is mounted to the shoe of the user. Standard IMU mechanization is used to propagate a 6 degrees of freedom (DOF) solution from an initial state. The foot reaches zero velocity during each step when the foot comes in contact with the ground. Internal errors of the IMU can be estimated to refine the navigation solution during these zero velocity periods. The output is a tracked 6 DOF pose that has significantly less drift when compared to stand-alone IMU mechanization. This approach is preferable to PDR

in that a calibration phase is not required. Additionally, an external heading measurement is not a strict requirement since the IMU tracks the orientation through the internal 3 axis gyroscopes.

1.1.3 Exteroceptive Sensors

Exteroceptive sensors are an increasingly common navigation source. These sensors derive the motion of the user through perception of the surrounding environment. Such sensors include cameras (both monocular and stereo), LiDARs, infrared devices, etc. Cameras rely on computer vision techniques to extract features from a 2D image, project the features into the 3D environment, and calculate the motion by comparing consecutive frames [16]. This technique is referred to as visual odometry and is covered in more detail in Chapter 2. Visual odometry can be computationally expensive, but the use of this technology has expanded in recent years due to increased processing capabilities. Visual odometry approaches do, however, suffer in poor lighting and other low visibility conditions. Even so, cameras are still a popular choice for pedestrian navigation due to low size, weight, and power consumption (SWaP). Other exteroceptive sensors, such as the LiDAR, operate similar to visual odometry, but can measure spatial information of the environment directly. LiDAR sensors do not depend on lighting conditions, but can be expensive and consume a relatively large amount of power. Other devices, such as the Microsoft Kinect are capable of measuring the environment in 3D. These devices are less expensive and popular among hobbyists, but are less accurate than stereo cameras and LiDARs and suffer from a low maximum range [17].

1.1.4 Relative State Measurement Frameworks

In order to fuse measurements from both proprioceptive and exteroceptive devices, an estimation algorithm must have the ability to process multiple relative state measurements. The process of fusing multiple relative state measurements is a non-trivial task, especially when the inputs are provided at different sampling rates. In 1999, Koifman and Bar-Itzhack

presented a method for aiding an IMU with an aircraft model [7]. In Koifman and Bar-Itzhack's work, the state vector is appended to include copies of the same state. Both states are allowed to evolve in the time update, one from the IMU and the other from the aircraft model. The measurement update includes a pseudo-measurement that equates the states, which aids the IMU navigation. More recently, Crocoll built off the work of Koifman and Bar-Itzhack by presenting a system that also aids IMU mechanization with an aircraft model [6]. Crocoll modifies the previous approach to gain computational advantages and the filter is referred to as the Unified Model.

In 2000, Roumeliotis and Burdick presented a method for processing relative state measurements in a Kalman filter framework [8]. The method was termed the stochastic cloning Kalman filter (SCKF) and operates by appending the original state vector with a cloned copy, just as in [7]. The cloned copy is held static through the time update, and the relative state measurement is processed in the measurement update of the Kalman filter. Later, Mourikis worked with Roumeliotis and Burdick in a publication that applies the SCKF for mobile robot localization [9]. Mourikis and his co-authors use the SCKF to fuse proprioceptive and exteroceptive relative state measurements, much like the work presented in this thesis.

As for applications of such algorithms in pedestrian navigation, Amirhosseini et. al. presents a stochastic cloning unscented Kalman filter for fusing visual odometry, an IMU, and magnetometer measurements [10]. The sensor suite used in Amirhosseini's work is mounted rigidly together and worn at the chest of the user. Amirhosseini's results show considerable improvements when fusing the data as compared to stand-alone visual odometry.

1.1.5 Foot-Mounted IMU Sensor Fusion

The use of a foot-mounted IMU for navigation was presented by Hutchings in 1997 [18], but it was Foxlin who popularized the method among the research community with his

publication in 2005 [2]. Since this time, many efforts have been made to fuse the foot-mounted IMU with other navigational devices for pedestrians. These works have dealt with the lever arm between the foot-mounted IMU and the other body-worn sensor in different ways with varying level of success.

In 2011, Chdid et. al. presented a method for fusing a waist-mounted visual odometry system with a foot-mounted IMU [19]. Chdid's approach used a constant jerk model for the time update of the high-level filter. Outputs from the visual odometry system and foot-mounted IMU were used to update the position, velocity, and acceleration states. While a constant jerk model is convenient for its simplicity, it does introduce a lag in the system, resulting in localization errors. In dead reckoning scenarios, the errors introduced by using such a model compound and lead to significant error drift with increasing distance traveled.

For fusing a foot-mounted IMU with GPS, Hide et. al. used GPS to update the IMU directly in a closely coupled fashion [20]. By mounting the GPS antenna rigidly to the foot-mounted IMU, the lever arm between the two sensors can be ignored. However, mounting a GPS antenna to the foot is problematic for two main reasons. The GPS antenna on the foot is likely to cause discomfort to the user and could impede the natural walking motion. Additionally, the GPS signals are likely to be corrupted from interference with the body when the antenna is mounted to the foot.

In 2014, Gao et. al. approached the same problem as Hide, but with the antenna mounted to the shoulder of the user [21]. Mounting to the shoulder is less impeding on the user's comfort and provides better line-of-sight to GPS signals. To account for the lever arm, the estimation algorithm was subject to inequality constraints, which limited the physical distance between the foot and the antenna position based on the user's height and leg length. The approach used two inertial sensors mounted to the same foot, and applied equality constraints relating the distance between the two.

The use of inequality constraints for pedestrian navigation was also used by Skog et. al. in 2012 to fuse dual foot-mounted IMU's with one IMU on each foot [22]. A maximum step

length was calculated based on the user's physical dimensions and an inequality constraint was imposed on the relative positions of the foot-mounted sensors. The approach successfully limited the position error drift compared to the non-constrained navigation solutions of the individual foot-mounted IMUs.

While the use of inequalities successfully fuses data, it requires the minimization of a nonlinear cost function, which can be non-trivial and specific to these scenarios. These cases also assumed prior knowledge of the user's physical dimensions for the best performance. The work in this thesis presents a general framework for fusing a foot-mounted IMU with other body-worn relative state measurements, which is both easy to implement and requires no *a priori* knowledge of the user.

1.2 Contributions

This thesis focuses on processing multiple relative state measurements that are not rigidly attached and periodically related, as is common for pedestrian navigation. The work presented in [10], [8], and [7] all assume that the two sources of relative state information are provided in the same coordinate frame and are rigidly attached. Unlike these publications in which pseudo-measurements are applied at every time step, this thesis applies the pseudo-measurements only once per human step when a spatial relationship can be made about the two systems.

Two different frameworks are considered for this fusion: a cascade and a centralized approach. As an example, the algorithm is applied to fusing a foot-mounted IMU with a chest-mounted visual odometry system. Preliminary results for the centralized approach were presented in an earlier publication by the author [1]. A simulation environment is developed for replicating signals from a foot-mounted IMU and a visual odometry system. The simulation environment allows for a thorough statistical analysis to be performed. The results are validated by processing the algorithm using experimental data. In summary, this thesis provides the following contributions to the field of pedestrian navigation:

- Development of two separate algorithms for fusing measurements of a foot-mounted IMU with relative state measurements from other body-worn systems.
- A simulation environment is developed for producing synthetic foot-mounted IMU signals and chest-mounted visual odometry measurements.
- A thorough analysis is made of the two approaches using Monte Carlo runs of simulated data.
- Simulations are validated by processing experimental data and comparing results of the two algorithms.

Chapter 2

Underlying Algorithms

The work presented in this thesis builds off of the state of the art for pedestrian navigation and relative state measurement processing. A great deal of research has been performed that focuses on each of these underlying processes. In order to present the contributions made in this work, an understanding of these existing methods must be made. The foot-mounted IMU algorithm in itself is an area of much research in recent years and this thesis utilizes the most widely adopted approach. Visual odometry is also popular in pedestrian navigation and a high-level understanding of its operations is presented in this chapter. An understanding of some existing frameworks for processing multiple relative state measurements is also presented.

2.1 The Foot-Mounted IMU

In recent years, the foot-mounted IMU has been a topic of interest for pedestrian localization. Mounting on the foot is ideal due to the anticipated behavior of the foot, specifically the periods of zero velocity experienced during a typical walking step. With knowledge of an inertial sensor having zero velocity, measurements can be made to estimate the internal errors that are inherent to inertial devices. Errors are estimated by tracking the navigation states along with their relation to the IMU errors in a Bayesian scheme with zero velocity updates (ZUPT) and zero angular rate updates (ZARU). The work presented in this thesis uses an extended Kalman filter (EKF) for estimation due to the filter's ability to operate on nonlinear systems. Devices that mechanize IMU readings for calculating a navigation solution are referred to as inertial navigation systems (INS). The process of updating an INS

using an EKF with zero velocity measurements will be referred to as INS-EKF-ZUPT (IEZ) for the remainder of this work.

2.1.1 Modeling the Inertial Measurement Unit

Inertial measurement units contain an array of accelerometers and gyroscopes that measure the motion of the device through specific force and rotation rate, respectively. All inertial measurements suffer from imperfections in their raw outputs. Over the years, many studies have modeled these errors to a high level of success. Some of the main sources of error in IMU readings include white noise, bias, scale factor, misalignment, and g-sensitivity. Although the scale factor and misalignment can be detrimental to a navigation solution, these are assumed to be deterministic errors that are accounted for *a priori* in a controlled calibration process. The white noise and bias errors, however, are stochastic in nature and must be handled accordingly. Another error source that exhibits stochastic properties is g-sensitivity, which acts on the inertial measurements as a function of the acceleration experienced by the sensor. While the foot location does experience high acceleration when walking (up to 5 g's), these instances are very brief and result in only about 5° of heading error after an hour of normal walking [23]. The inertial values are also affected by the rotation of the earth [5]. These effects can be significant when traveling long enough distances and depend on the user's location on the earth. While incorporating the earth rotation rate for IMU mechanization improves the navigation solution, it was not considered in this thesis. By excluding earth rotation, the algorithm can be developed in a local-level frame, reducing the complexity of the derivations presented later in this text. Therefore, the model used for the entirety of this thesis will only consider bias and white noise as error sources.

The gyroscopic readings ($\tilde{\omega}^b$) are described as a vector containing the true rotation rate (ω^b), the gyroscope bias (b_g), and white noise (w_g) as shown in Equation (2.1).

$$\tilde{\omega}^b = \omega^b + b_g + w_g \tag{2.1}$$

Similarly, the specific force readings from the accelerometers ($\tilde{f}^b$) are modeled as a vector containing the true specific force (f^b), the accelerometer bias (b_a), and white noise (w_a) as shown in Equation (2.2).

$$\tilde{f}^b = f^b + b_a + w_a \tag{2.2}$$

Measurements from the IMU are provided in the body frame (b) of the device and in order to propagate a navigation solution in global coordinates, a rotation from the body frame to the navigation frame (n) must be performed. The coordinate frame rotations are explained in detail in Section A.1 of the Appendix. It is important to note that the specific force contains both the true accelerations and the acceleration due to gravity [5]. The operations for removing gravity from the specific force readings will be described in detail later in this section.

The biases that act on both the accelerometer and gyroscope readings are not static. For the purpose of this thesis, bias values are modeled as a first order Markov process as shown in Equations (2.3) and (2.4) [24].

$$\dot{b}_g = -\frac{1}{\tau_g} b_g + w_{b_g} \tag{2.3}$$

$$\dot{b}_a = -\frac{1}{\tau_a} b_a + w_{b_a} \tag{2.4}$$

The first order Markov process models biases as drifting values with dynamics according to time constants (τ_g, τ_a) and driving noise (w_{b_g}, w_{b_g}).

Both the noise acting on the inertial readings and the bias variation noise are assumed to be zero-mean Gaussian as defined by Equation (2.5). The magnitude of the noise sources are described using the input covariance matrix (Q), which is also defined by Equation (2.5),

$$\mathrm{E}\left[w\right] = 0 \qquad \mathrm{E}\left[ww^T\right] = Q \tag{2.5}$$

where w is a vector containing all of the noise sources ($w = \left[w_a, w_g, w_{b_a}, w_{b_g}\right]$). With three accelerometers and three gyroscopes, the noise vector is length 12 and the covariance matrix is a 12×12 matrix.

2.1.2 Gait Monitoring

As a precursor to the tracking of a foot-mounted IMU, gait monitoring must be performed for automatic detection of zero velocity conditions. Most of the literature on foot-mounted IMU tracking uses the raw inertial measurements for gait monitoring to allow the process to be self-contained and independent of any external sources of information. The automated detection follows a set of logic that acts on a window of past inertial measurements [2] [3]. Two conditions must be met for the detection of zero velocity. To satisfy the first condition, the magnitude of the latest inertial measurement must be within a predetermined threshold. The second condition is met when the variance of the window of inertial measurements is below a certain threshold. Three parameters must be chosen when using this approach for gait monitoring: window length, magnitude threshold, and variance threshold. The parameters used for zero velocity detection in this work can be found in Table 2.1.

Table 2.1: Zero Velocity Detection Parameters

Window Length	20 samples
Magnitude Threshold	$\left[\begin{array}{cc} 9 & 11 \end{array}\right]$ m/s^2
Variance Threshold	.0064 (m/s^2)^2

Zero velocity is not the only information that is beneficial for the navigation solution, but also zero angular rate. With knowledge that the device is not rotating, zero angular rate updates can be applied to the EKF to estimate gyroscope biases. A similar method to that which detects zero velocity is used for detecting the zero angular rate condition. This

condition is determined by studying the raw rotation rates provided from the gyroscopes of the IMU. The values used for the detection of zero angular rate conditions can be found in Table 2.2.

Table 2.2: Zero Angular Rate Detection Parameters

Window Length	5 samples
Magnitude Threshold	$\begin{bmatrix} -0.05 & 0.05 \end{bmatrix}$ rad/s
Variance Threshold	1 $(\text{rad/s})^2$

The results of both zero velocity and zero angular rate detection can be seen in Figure 2.1. Notice that zero angular rate is detected much less than the zero velocity conditions. In a typical step, the zero angular rate conditions are only met for a very small time period, if they are met at all. However, this is not as critical for navigation since the gyroscopes experience less drift when compared to the accelerometers [5]. The use of zero angular rate update is most beneficial for initial stationary periods and other sections of extended still periods.

2.1.3 Estimation

Using raw inertial measurements and gait monitoring results, the foot is tracked using an EKF. This section describes the models used for the EKF while a detailed explanation of the EKF process can be found in Section (A.3) of the Appendix. The IEZ estimator tracks the 6 DOF navigation states of the foot (designated with f subscript) as well as the bias in the IMU. The states consist of position (r_f), velocity (v_f), attitude (φ_f), accelerometer bias (b_a), and gyroscope bias (b_g). To estimate a 6 DOF solution, each component of the state vector is three-dimensional, resulting in a 15 variable system. The full state vector can be seen in Equation (2.6).

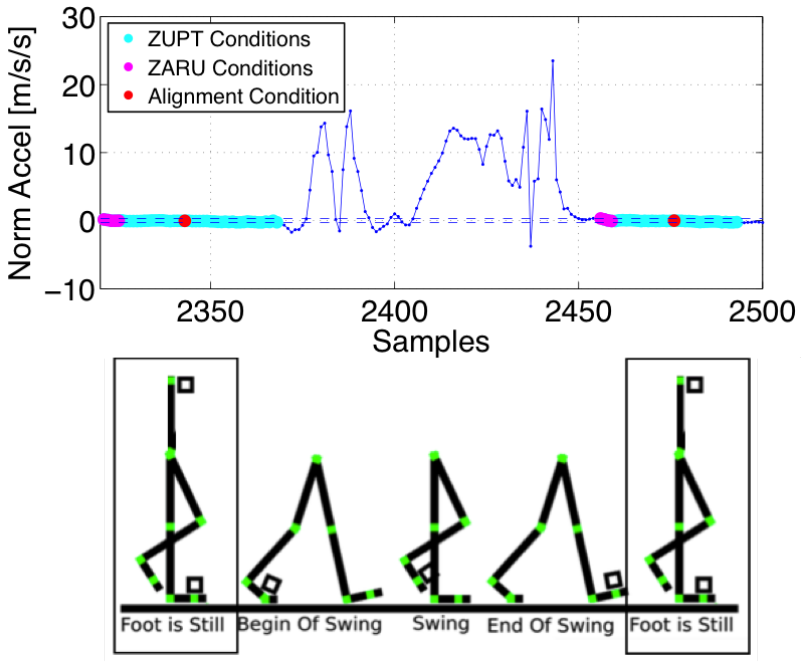

Figure 2.1: Gait Monitoring Results [11]

$$x_f = \begin{bmatrix} \mathrm{r}_f & \mathrm{v}_f & \varphi_f & b_a & b_g \end{bmatrix}^T \tag{2.6}$$

IEZ Initialization

To initialize the system, the position and heading are defined with respect to an arbitrary reference frame and are generally initialized to zero. It is assumed for this work, and for most foot-mounted IMU applications, that the foot starts from rest and, therefore, the velocity is also initialized to zero. In practice, the system can delay initialization until receiving zero velocity detection from the gait monitoring system. The roll and pitch are initialized to correspond with the local-level navigation frame. For this, the specific force readings from the IMU are used in an approach termed accelerometer leveling [5]. In zero velocity conditions, the accelerometer readings consist of only the accelerometer bias, effects of gravity ($g^n = \begin{bmatrix} 0 & 0 & 9.81\,\mathrm{m/s^2} \end{bmatrix}^T$), and noise. Knowing that the gravity acts only in the downward direction of the navigation frame, the relation shown in Equation (2.7) can be made,

$$\tilde{f}^b = f^b + b_a + w_a = C_n^b g^n + b_a + w_a \tag{2.7}$$

where the rotation matrix (C_n^b) rotates the gravity vector from the navigation frame to the body frame.

By solving for the scalar component of the accelerometer readings and setting the noise variable to zero, a relation can be made between tilt angles (roll/pitch) and the accelerometer readings. A least squares initialization process is used to account for the correlation introduced between the accelerometer bias estimates and tilt angle estimates. Using the nomenclature of Appendix A.2, the inputs to the least squares are shown in Equations (2.8) and (2.9).

$$z_{1:3} = \tilde{f}^b \tag{2.8a}$$

$$z_{4:6} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{2.8b}$$

$$h_1 = g\sin(\theta) + b_{a_x} \tag{2.9a}$$

$$h_2 = -g\cos(\theta)\sin(\phi) + b_{a_y} \tag{2.9b}$$

$$h_3 = -g\cos(\theta)\cos(\phi) + b_{a_z} \tag{2.9c}$$

$$h_{4:6} = b_a \tag{2.9d}$$

The measurements to the least squares process (z) are specific force readings and an initial estimate of zero for the accelerometer biases. The measurements are weighted by the accelerometer covariance ($Q_{1:3,1:3}$) and initial accelerometer bias uncertainty ($P_{b_a,i}$) as shown in Equation (2.10).

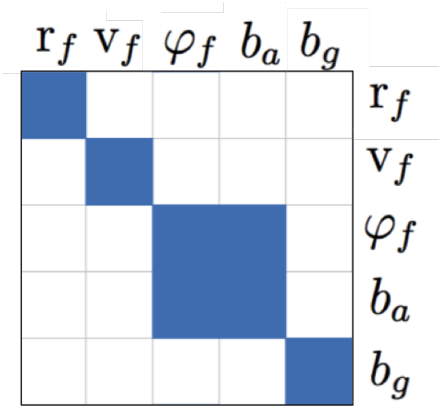

Figure 2.2: Correlations of Initial IEZ State Covariance

$$W^{-1} = C_z = \begin{bmatrix} Q_{1:3,1:3}/\Delta t & O \\ O & P_{b_a,i} \end{bmatrix} \tag{2.10}$$

In Equation (2.10), $Q_{1:3,1:3}$ represents a submatrix of Q from Equation (2.5), specifically rows and columns 1 through 3. Note that the accelerometer covariance is discretized by dividing by the IMU time step (Δt).

The result of the least squares process is an initial estimate for the tilt angles along with a covariance between the tilt angles and accelerometer bias estimates. The covariance matrix from least squares is combined with initial state estimate variances for position, velocity, yaw, and gyroscope bias. Figure 2.2 shows the full initial state estimate covariance and the correlations by representing the state estimate covariance matrix visually and shading in the populated sections.

IEZ Time Update

The inertial measurements are used to predict the navigation solution in the time between measurement updates. This is done by treating the inertial measurements as inputs ($u = [\tilde{f}^b, \tilde{\omega}^b]^T$) in the time update of the EKF. A nonlinear function ($\dot{x}_f = f(x_f, u, w)$) is used to describe the time rate of change of the 15 states. This is also known as the state

equations. The state equation for position ($f_{1:3}$) is simply equal to the velocity as shown in Equation (2.11).

$$f_{1:3} = \mathrm{v}_f \tag{2.11}$$

The rate change in velocity is equal to the true acceleration in the navigation frame. The true navigation frame acceleration is equated to the accelerometer inputs by 1) subtracting the bias and noise from the accelerometer readings 2) rotating the body frame measurements into the navigation frame using the rotation matrix and 3) removing the effect of gravity acting on the accelerometer readings [3]. This process is performed in the state equations for velocity ($f_{4:6}$) as shown in Equation (2.12).

$$f_{4:6} = C_b^n(f^b) + g^n = C_b^n(\tilde{f}^b - b_a - w_a) + g^n \tag{2.12}$$

For the function describing the rate of change in the orientation states, the gyroscope readings are used. First, the equation subtracts the bias and noise terms in the same manner as for the accelerometer readings. To transform the rotation rates from the body frame to the navigation frame, the mechanization matrix (M) is used [25]. A detailed description of the mechanization matrix can be found in Appendix A.1. The state equation for the orientation states ($f_{7:9}$) is shown in Equation (2.13).

$$f_{7:9} = M\omega^b = M(\tilde{\omega}^b - b_g - w_g) \tag{2.13}$$

The inertial bias states are modeled as a first order Markov process with time constants as shown in the previous section. For extended periods without measurement updates to the EKF, the bias estimates will tend towards zero at a rate that depends on the value of the time constants. Equations (2.14) and (2.15) show the state equations for the accelerometer and gyroscope biases ($f_{10:12}, f_{13:15}$).

$$f_{10:12} = -\frac{1}{\tau_a} b_a + w_{b_a} \tag{2.14}$$

$$f_{13:15} = -\frac{1}{\tau_g} b_g + w_{b_g} \tag{2.15}$$

The time update is performed upon receiving a new inertial measurement. Given the current state estimate ($\hat{x}_f$) and the system input, and setting noise variables to zero, the state equations are numerically integrated in order to propagate the states to the next time step. In addition to predicting the states at the next time step, the state estimate covariance (P) must evolve accordingly and, for this, the state transition matrix (Φ_f) is used. In order to calculate the state transition matrix at the current time step, a numerical integration is performed on the first derivative of the state transition matrix ($\dot{\Phi}_f$). The first derivative of the state transition matrix is found by taking the Jacobian of the state equations with respect to the states and evaluated at the current state estimate as shown in Equation (2.16).

$$\dot{\Phi}_f = \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_k, u_k, 0)} \tag{2.16}$$

IEZ Measurement Update

The time update is repeated until detection of either zero velocity or zero angular rate conditions. When zero velocity is detected, an EKF update is performed with the measurement and measurement matrix as shown in Equation (2.17),

$$y = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \quad H = \begin{bmatrix} O & I & O & O & O \end{bmatrix} \tag{2.17}$$

where I is a 3x3 identity matrix and O is a 3x3 matrix of zeros. This zero velocity update allows for the estimation of the accelerometer biases. In addition, the ZUPT also helps to estimate the tilt angles (roll and pitch) due to their correlation with the accelerometer bias states [3].

The measurement covariance (R) associated with the zero velocity update must be chosen, since this information is not provided from an outside source. In [2], Foxlin introduces a method for choosing the covariance used for this update. Foxlin says that by making the update covariance no smaller than the current velocity variance, the corrections will occur gradually over the update period. Equation (2.18) is used to ensure gradual corrections in this way.

$$R = I \cdot \text{trace}(HPH^T) \tag{2.18}$$

In addition to estimating the accelerometer biases, it is also beneficial to estimate the biases of the gyroscope. The use of the IEZ algorithm with only ZUPT measurements does not allow for full estimation of gyroscope biases. While x and y gyroscope biases can be estimated quite well due to their relation with the tilt angles, the z gyroscope bias is highly unobservable in low dynamics (straight paths, slow turns) [4]. For aiding in the estimation of the z gyroscope bias, a ZARU is used. Implementing a ZARU measurement is not as straightforward as for ZUPT. Since there are no angular rate states, a direct zero valued measurement cannot be used. However, a measurement of the gyroscope bias can be performed by solving Equation (2.1) of the gyroscope model for the bias with values of zero angular rate. This leads to measurement and measurement matrices according to Equation (2.19) [3].

$$y = \tilde{\omega}^b \quad H = \begin{bmatrix} O & O & O & O & I \end{bmatrix} \tag{2.19}$$

The covariance for the zero angular rate update is calculated in a similar manner to that of the zero velocity update, with one addition. The update covariance also accounts for the noise in the inertial readings by using the gyroscope portion of input covariance matrix ($Q_{4:6,4:6}$) as shown in Equation (2.20).

$$R = I \times \text{trace}(HPH^T) + Q_{4:6,4:6}/\Delta t \tag{2.20}$$

The input covariance matrix is discretized by dividing by the time step of the IMU. In most cases, the trace(HPH^T) portion of the summation shown in Equation (2.20) has a relatively small contribution to the total measurement uncertainty. However, the trace(HPH^T) portion's contribution to the total measurement uncertainty could be significant after long periods without zero angular rate updates where the z gyroscope bias has grown to a relatively high value.

2.2 Visual Odometry

Visual odometry is a valuable navigation source for many different applications. With recent advancements in processor capabilities, an increase in research of visual odometry has followed. The process of visual odometry operates by estimating egomotion of a camera between consecutive frames. Egomotion is defined as the motion of the observer relative to its environment. Egomotion estimates are used to propagate pose estimates from some initial condition, resulting in a dead reckoning navigation solution. Because visual odometry does not rely on GNSS signal availability, it makes for an ideal candidate for applications in pedestrian navigation. The following research deals specifically with stereo visual odometry, a subset of the many visual odometry methods. This section discusses the operation of stereo visual odometry at a high-level. In addition, a discussion will be made on some of the benefits and drawbacks to stereo visual odometry, a necessary understanding for the development of the fusion algorithm of topic.

Stereo visual odometry acts on a stream of images provided by two separate camera lenses. The two lenses are rigidly attached and related to one another through a rigid body transformation. Before any processing can take place on the stereo images, an estimate of the baseline transformation between the lens focal points is made. Assuming that the baseline

transformation is an unchanging, deterministic relation, a single *a priori* calibration process can be performed to estimate these parameters [26].

Each image that is provided by the stereo camera undergoes a detection scheme to extract distinguishable features in the image. Such features include corners, lines, or any points of high intensity gradients in the image. A correspondence is then made to relate each feature from one image of the stereo camera to the next. Due to the separation between the two stereo camera lenses, depth perception is attained. Using the pinhole camera model, a point cloud is generated from the features that correspond between the two images. In summary, each stereo image pair produces a new 3D point cloud of the stereo camera's perceived environment. This, of course, relies on a clear view of an environment with distinguishable features. Some non-ideal scenarios include dark scenery, poor weather conditions, uniform surfaces, etc. [26].

For each epoch in time, consecutive point clouds are compared to estimate the egomotion of the camera. Assuming that the environment is static, any change in consecutive point clouds is due to motion of the camera. An iterative minimization process is used to converge on the associated translation and rotation of the camera [16]. The assumption of a static environment is likely to fail in many scenarios, especially for pedestrian navigation. However, stereo camera point clouds are extremely dense and are usually capable of converging on an egomotion estimate even in the presence of dynamic features. The effect of dynamic features on egomotion is just one of the error sources of visual odometry along with erroneous feature locations and false correspondences between images [26].

This thesis uses the Libviso2 open source implementation of stereo visual odometry [16]. The outputs of visual odometry are 3D body frame translation ($\Delta\tilde{\mathrm{r}}^b$) and rotation ($\Delta\tilde{\varphi}^b$) between epochs in time. The Δ notation is used to emphasize that the measurements are a change from one time step to another. To truly model the errors associated with the visual odometry egomotion would be extremely involved and require delving into the low-level algorithm. For the purpose of this thesis, an assumption is made that these inputs are

affected only by white noise. Therefore, the egomotion measurements are modeled as a sum of the true egomotion ($\Delta\mathrm{r}^b$,$\Delta\varphi^b$) and white noise ($\epsilon_\mathrm{r}, \epsilon_\varphi$) as shown in Equation (2.21).

$$\Delta\tilde{\mathrm{r}}^b = \Delta\mathrm{r}^b + \epsilon_\mathrm{r} \tag{2.21a}$$

$$\Delta\tilde{\varphi}^b = \Delta\varphi^b + \epsilon_\varphi \tag{2.21b}$$

The egomotion of the visual odometry system can be used as inputs to a discrete difference function to propagate navigation states. The states required to model the system (x_t) include the 3D position (r_t) and orientation (φ_t) of the camera as shown in Equation (2.22).

$$x_t = \begin{bmatrix} \mathrm{r}_t & \varphi_t \end{bmatrix}^T \tag{2.22}$$

The t subscript is used to designate variables associated with the tracking of the torso. Unlike with IMU mechanization, the position and orientation are updated directly using egomotion and, accordingly, no velocity states are required. Also, egomotion from the visual odometry system is assumed to be unbiased and because of this, no bias states are included in the state vector.

The discrete difference function ($x_{t,k} = g(x_{t,k-1}, z_k, \epsilon_k)$) describes the change in states between time steps (k). The state at the next time step is a function of the previous state, egomotion inputs ($z_k = \begin{bmatrix} \Delta\tilde{\mathrm{r}}^b & \Delta\tilde{\varphi}^b \end{bmatrix}^T$), and the egomotion noise. The difference equation for position ($g_{1:3}$) subtracts noise from the translation input and transforms the result into the navigation frame before adding to the previous position as shown in Equation (2.23).

$$g_{1:3} = \mathrm{r}_{t,k-1} + C_b^n(\Delta\tilde{\mathrm{r}}^b - \epsilon_\mathrm{r}) \tag{2.23}$$

For transforming the egomotion translation into the navigation frame, the rotation matrix is used, which is a function of attitude.

The difference equation for attitude ($g_{4:6}$) is very similar to the difference equation for position. The noise is subtracted from the input of rotation and the result is transformed into the navigation frame before adding to the previous attitude as shown in Equation (2.24).

$$g_{4:6} = \varphi_{t,k-1} + M(\Delta\varphi^b - \epsilon_\varphi) \tag{2.24}$$

For transforming the egomotion rotation into the navigation frame, the mechanization matrix is used, which is a function of attitude.

The visual odometry egomotion can be used as the time update for an EKF. For this, the state transition matrix is found by taking the partial derivative of the discrete difference equation with respect to the states as shown in Equation (2.25).

$$\Phi_t = \left.\frac{\partial g}{\partial x}\right|_{(\hat{x}_k, z_k, 0)} \tag{2.25}$$

Note that since the state equations are already discrete, the state transition matrix can be calculated directly without integration [9].

2.3 Relative State Measurements

The previous two sections introduce particular methods for estimating a navigation solution. Both the foot-mounted IMU and visual odometry provide what will be referred to as relative state measurements. A relative state measurement describes the change in state from one point in time to the next and is generally a measurement of the motion of a body. Therefore, relative state measurements require some initial condition in order to propagate a navigation solution.

As opposed to the previous two sections that deal with specific systems, this section is more general and introduces methods for fusing arbitrary relative state measurements. By introducing general methods for relative state measurement fusion, a better understanding of these algorithms can be made. Specific applications of the algorithms introduced in

this chapter can be found in Chapter 3, where the foot-mounted IMU is fused with visual odometry.

Relative state measurements act as the basis for a navigation Kalman filter. The relative state measurements are used to predict the state estimate between measurement updates. For scenarios in which multiple relative state measurements are available, intuition says that the two can be used to refine the navigation solution, especially while dead reckoning. The process of fusing multiple relative state measurements into a Kalman filter framework, however, is not as straightforward as one may think. While the Kalman filter allows for any number of measurement updates, only a single source can be used for the time update.

The incorporation of multiple relative state measurements into a navigation solution is desired for multiple reasons. The added redundancy helps to improve the robustness of the solution. A faulty measurement is less likely to have detrimental effects on the navigation solution when fused with other redundant information. The redundancy also allows for a fault detection scheme to be implemented on the relative state information. In addition to increased robustness, fusion of relative state information theoretically reduces the amount of drift incurred in dead reckoning [6] [7] [8].

2.3.1 Stochastic Cloning Kalman Filter

One method for handling relative state measurements is by way of the stochastic cloning Kalman filter (SCKF) [8]. The SCKF processes relative state information in the measurement update of the Kalman filter. On initialization, the original state vector is appended with a static "cloned" state (x_s) to form the full state vector ($\breve{x}$) as shown in Equation (2.26),

$$\breve{x} = \begin{bmatrix} x \\ x_s \end{bmatrix} \tag{2.26}$$

where the s subscript designates static, unchanging values. These static states are the basis for the SCKF operation.

By making an exact clone of the original state vector, the corresponding states are fully correlated on initialization. The initial state estimate covariance must account for the correlation between the original and cloned state vectors. The off-diagonals of the full initial state estimate covariance ($\breve{P}$) are populated as shown in Equation (2.27),

$$\breve{P} = \begin{bmatrix} P & P \\ P & P \end{bmatrix} \tag{2.27}$$

where P is the initial state estimate covariance for the original state.

The relative state information that is provided at a higher sample rate is used as the time update. The cloned states remain static during the time update while the original state is allowed to evolve according to its state equation. The state transition matrix and input covariance matrix are appended to correspond with the static states. The state transition matrix is appended with an identity matrix and the discrete input covariance matrix is appended with zeros as shown in Equations (2.28) and (2.29).

$$\breve{\Phi}_k = \begin{bmatrix} \Phi_k & O \\ O & I \end{bmatrix} \tag{2.28}$$

$$\breve{Q}_k = \begin{bmatrix} Q_k & O \\ O & O \end{bmatrix} \tag{2.29}$$

The result of Equations (2.28) and (2.29) is the full state transition matrix ($\breve{\Phi}_k$) and full discrete input covariance matrix ($\breve{Q}_k$).

Upon receiving a measurement from the second relative state source, a Kalman pseudo-measurement update is performed. The measurement update is performed in the same fashion as a standard Kalman filter update. The only difference from a typical Kalman filter update is that the measurement model for the pseudo-measurement is a function of

both the current state and the cloned state. The measurement model is also a function of pseudo-measurement errors (v_k) as shown in Equation (2.30).

$$y_k = h(x, x_s, v_k) = h(\breve{x}, v_k) \tag{2.30}$$

After a pseudo-measurement update is performed, the cloned state is then reset to the original state vector. At this point, the two states are once again fully correlated. The full state estimate covariance matrix is recorrelated, just as on initialization as shown in Equation (2.27). The process is repeated for each relative state measurement and the result is a fused dead reckoning solution that experienced less drift than either of the two individual odometry sources.

2.3.2 Unified Model

Another method for processing multiple relative state information is by way of the Unified Model [6]. As opposed to the SCKF, the Unified Model processes relative state measurements in the time update. Similar to the SCKF, the state vector is appended with a clone of the original state vector to form the unified state vector ($\breve{x}$) as shown in Equation (2.31).

$$\breve{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.31}$$

Additionally, the state estimate covariance matrix is appended and correlated in the same manner as for the SCKF, shown previously in Equation (2.27).

In the time update of the Unified Model, both state vectors evolve alongside one another according their respective state equations. The unified state equation ($\breve{f}$) is formed using Equation (2.32),

$$\breve{f}\left(\breve{x}, u, w\right) = \begin{bmatrix} f_1\left(x_1, u_1, w_1\right) \\ f_2\left(x_2, u_2, w_2\right) \end{bmatrix} \tag{2.32}$$

where the two state equations (f_1, f_2) are combined into a single state equation vector. The unified state equation is a function of the current state, the inputs to the two systems (u_1, u_2), and the model disturbances of the two systems (w_1, w_2).

The unified state estimate covariance matrix is updated in the time update using the unified state transition matrix $(\breve{\Phi}_k)$ and the unified discrete input covariance $(\breve{Q}_k)$. The unified state transition matrix is formed as a diagonal of the two state transition matrices from the different models as shown in Equation (2.33).

$$\breve{\Phi}_k = \begin{bmatrix} \Phi_{1,k} & O \\ O & \Phi_{2,k} \end{bmatrix} \tag{2.33}$$

Similarly, the unified discrete input covariance matrix is formed as a diagonal of the two discrete input covariance matrices as as shown in Equation (2.34).

$$\breve{Q}_k = \begin{bmatrix} Q_{1,k} & O \\ O & Q_{2,k} \end{bmatrix} \tag{2.34}$$

The measurement update of the Unified Model is a pseudo-update that relates the two state vectors together. In a simple case, the two state vectors are identical and the measurement model subtracts the two states from one another. In this simple case, the measurement used in the update is a vector of zeros. Optionally, the two states may be related by some baseline separation, which will be used as the measurement vector. The covariance associated with this update will be the confidence in this baseline separation itself.

Unlike the SCKF, the state vector does not need to be reset after a pseudo-measurement. This is beneficial in that it allows for some continuity in the navigation solution, making it

easier to implement. It is also easier, in most cases, to formulate the measurement models using the unified approach. The measurement model for the unified approach is often linear and all of the nonlinearities of the relative state measurements are handled in the time update. With a linear measurement model, the measurement matrix is constant and does not require calculation of a Jacobian.

Chapter 3

Fusion of Foot-Mounted IMU With Visual Odometry

This chapter presents two methods for fusing a foot-mounted IMU with other relative state measurements for pedestrian navigation. To serve as an example, the algorithm will be developed for fusion with a chest-mounted visual odometry system. These two systems provide redundant odometry information, which can be fused for an improved dead reckoning solution. These systems are complementary for two main reasons. The first complementary trait is experienced during stationary periods, when the user is not walking and the torso is relatively motionless. In this time, the foot-mounted IMU position estimate remains virtually stationary, while the visual odometry error grows with time. A fusion of the two systems would utilize the stationary behavior for the fused solution. The two systems also complement each other in the walking phase. While the foot-mounted IMU solution suffers from z gyro bias drift, and consequentially heading drift, the visual odometry egomotion is unbiased and can be used to improve heading estimation performance.

The challenge of fusing such information lies in the non-rigid relationship between the two measurement sources. By making assumptions about the motion of the human body throughout typical walking gait, the two systems can be related. One assumption is that the relative position of the camera and foot-mounted IMU returns to the same value once per step. The assumption that the relative position takes on periodic behavior is reasonable considering that the foot must support and balance the upper body. The second assumption is that the change in heading of the chest-mounted camera and foot-mounted IMU from one step to the next is the same. Although the heading profiles of the two systems differ significantly throughout a step, the change in heading between the two systems returns to roughly the same value when the foot is contacting the ground. The instance when

the relative position and heading returns to the same value is referred to as the alignment instance. Variations in the relative position and heading at alignment are inevitable, but the assumption is that these variations are zero-mean. The instance of alignment between the foot-mounted IMU and chest-mounted camera is detected using information from gait monitoring when the foot is in the stance phase.

Upon detecting alignment, a pseudo-measurement is used to relate the systems. One method for relating the two systems is based on the stochastic cloning Kalman filter, which was presented earlier in Chapter 2. While the SCKF is comparatively easy to implement in this case, it is shown to have suboptimal performance due to certain characteristics of the IEZ algorithm. A second method is proposed to combat the suboptimal characteristics of the SCKF approach, which borrows from both the Unified Model and SCKF.

3.1 Coordinate Frames

Up to this point, two coordinate frames have been used to describe the system: the body frame and the navigation frame. In Chapter 2, the body frame was referred to as a coordinate frame attached to the sensor, whether that is the IMU or the stereo camera. In order to develop algorithms for fusing measurements from the two systems, it is necessary to distinguish the two frames. For this, the "foot frame" (f) will be a frame attached to the foot-mounted IMU, while the "torso frame" (t) is attached to the stereo camera. The two frames are separated by a rotation and a translation that changes throughout the step. The translation between frames is not important for the sensor fusion for reasons that will be discussed later in this chapter. It is necessary, however, to introduce the boresight angle (β), which is the difference between the foot and torso heading. In order to track the frames in the centralized filter, an estimate of this angle must be made. A visual representation of the two frames and the boresight angle can be seen in Figure 3.1.

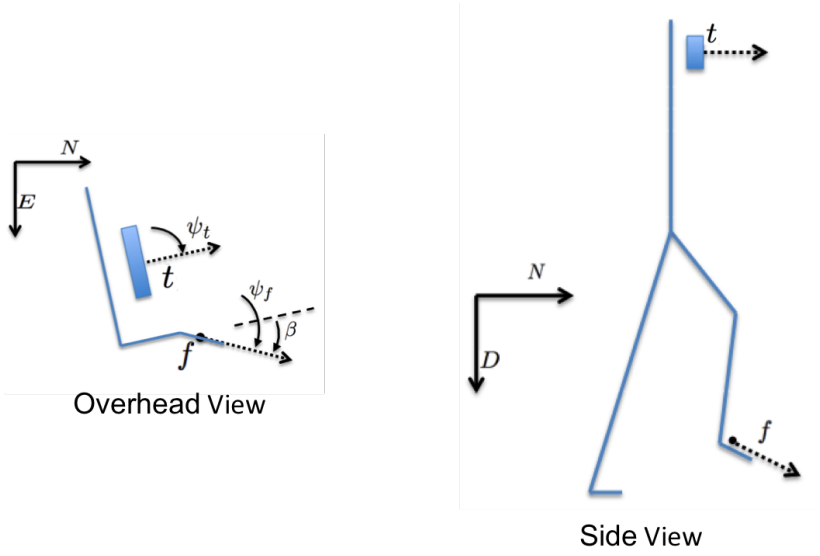

Figure 3.1: Foot and Torso Coordinate Frames

3.2 Cascade Fusion Approach

First, a method is presented that decentralizes the navigation states of the foot-mounted IMU and chest-mounted visual odometry system. In this approach, a cascade implementation is used in which the foot-mounted IMU is processed at the lowest level, which updates the high-level visual odometry system. The low-level IEZ algorithm is similar to that presented in Chapter 2 and tracks the foot's 6 DOF navigation states. At each step, the IEZ algorithm outputs the associated step length and a change in heading, which are used to update the high-level system. At the high-level, the visual odometry egomotion is used for the time update. The measurement update is processed using a stochastic cloning Kalman filter with relative state measurements from the foot-mounted IMU. External measurements can also be applied to the system using a standard Kalman filter update. A visual representation of this process can be seen in Figure 3.2.

3.2.1 Low-Level Foot Tracking

The foot states are tracked in similar fashion to that discussed in Chapter 2. In the cascade method, not all of the information from the foot-mounted IMU is useful for fusion

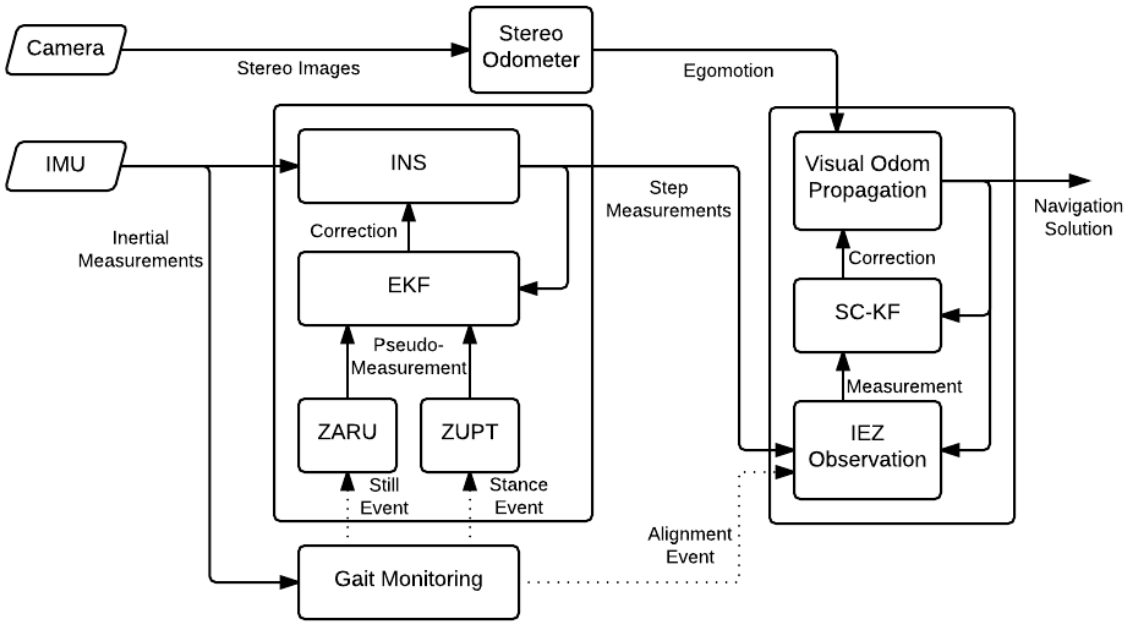

Figure 3.2: Cascade Filter Flowchart

at the higher level. Outputs of the IEZ algorithm can be extracted, providing information of each step taken. This can be beneficial from a communications standpoint. If measurements are only provided once per step, a much lower amount of data is required to pass to the integration filter. In this case, Bluetooth (or similar wireless technology) would have acceptable bandwidth for passing step information, which could eliminate the need for wires that may obstruct the user's motion [4]. For the cascade filter presented in this thesis, the step outputs of the IEZ system are step length and change in heading (delta heading).

Since these outputs will be used to update the torso states, their values should be fed to the high-level system at consistent times for each step. Ideally, this timing would occur when the 2D position of the foot-mounted IMU is perfectly aligned with that of the chest-mounted camera. Although no direct information is available for when this alignment occurs, the instance can be approximated by studying the raw inertial signals of the IMU. For this, the results from gait monitoring are used. Alignment is declared when the zero velocity condition has been satisfied for a specified amount of time. This specified amount of time

was chosen to be a tenth of a second for all of the results shown in this paper. This value was determined by studying a portion of the initial IEZ and visual odometry solutions and is small enough to where error drift is negligible. The alignment detection produces consistent alignment events at each step; however, its performance is not perfect. Later in this chapter an approach will be discussed for handling the errors associated with alignment detection.

In addition to calculating the values of step length and delta heading, the associated covariance in these values must also be provided for fusing with the high-level system. In order to generate the covariance of IEZ outputs, the state cloning approach is used. This approach is very similar to a stochastic cloning Kalman filter, but for calculating outputs as opposed to updating the system. The original 15 variable system is appended with a clone of the foot position and heading as shown in Equation (3.1).

$$x = \begin{bmatrix} x_f & | & \mathrm{r}_{f,s} & \psi_{f,s} \end{bmatrix}^T \tag{3.1}$$

On initialization, the cloned states are set equal to the original state and their covariance is fully correlated as explained in Chapter 2 for the SCKF. Throughout a stepping motion, the original states are allowed to evolve according to Equations (2.11-2.15), while the cloned states are held static. When propagating the state estimate covariance, the state transition matrix and discrete input covariance matrix are modified according to Equations (2.28) and (2.29).

Upon detecting a step, the IEZ outputs are calculated as a function of both the original state and the cloned state as shown in Equation (3.2).

$$z_{step} = \begin{bmatrix} |\mathrm{r}_f - \mathrm{r}_{f,s}| \\ \psi_f - \psi_{f,s} \end{bmatrix} \tag{3.2}$$

The covariance associated with this output is calculated using the output Jacobian (H) and state estimate covariance matrix as shown in Equation (3.3).

$$R_{step} = HPH^T \tag{3.3}$$

The output Jacobian is calculated using Equation (3.4).

$$H = \left.\frac{\partial z_{step}}{\partial x}\right|_{\hat{x}_k} \tag{3.4}$$

After the step output and associated covariance have been calculated, the cloned state is reset and covariance is again fully correlated as on initialization. The process is then repeated for each detected step.

3.2.2 High-Level System

The high-level system for the cascade filter approach tracks the 3D position and orientation of the stereo camera. The navigation solution is propagated in the time update using egomotion from the visual odometry system according to Equations (2.23) and (2.24). For the incorporation of the step information from the foot-mounted IMU, a stochastic cloning Kalman filter framework is used. The original state vector is appended with static copies of 3D position ($\mathrm{r}_{t,s}$) and heading ($\psi_{t,s}$) to allow for a step update at each detected alignment. The full state vector includes 10 variables and can be seen in Equation (3.5).

$$x = \begin{bmatrix} x_t & | & \mathrm{r}_{t,s} & \psi_{t,s} \end{bmatrix}^T \tag{3.5}$$

The step update includes the step length and delta heading values provided by the low-level IEZ algorithm. The system is updated according to the measurement and measurement equation shown in Equation (3.6).

$$y = z_{step} \qquad h\left(x, x_s\right) = \begin{bmatrix} |\mathrm{r}_t - \mathrm{r}_{t,s}| \\ \psi_t - \psi_{t,s} \end{bmatrix} \tag{3.6}$$

To deal with the uncertainty that exists in the alignment detection, the covariance reported from the IEZ output is summed with an expected alignment variance (R_{align}) as shown in Equation (3.7).

$$R = R_{step} + R_{align} \tag{3.7}$$

Intuitive values can be chosen for the alignment variance by considering the motion involved in a typical step. For the heading alignment variance, it can be assumed that that the two system's headings should not vary by more than around 45° due to physical limitations of the human body. Intuition of the human step can also be used to choose the variance associated with the position alignment. During a typical stepping motion, the entire body is supported solely through the planted foot. In order to maintain balance, the torso is positioned above the foot. Therefore, the maximum assumed misalignment error was chosen to be half that of a typical human step length, or 0.6 meters. Assuming that 99.9% of the alignment errors fall within these values, 45° and 0.6 meters were chosen for the 3σ values of the delta heading and step length updates, respectively. The alignment covariance (R_{align}) is set accordingly. The values chosen for the alignment variance are rather conservative. Even with these conservative values, the foot-mounted IMU improves the stand-alone performance of the visual odometry system, assuming the errors are unbiased. The alignment of the frames is likely to experience the most errors when the user undergoes maneuvers such as turning and during the transients of walking. An improvement to the choice of alignment variance could be made by inflating this variance when detecting such maneuvers. For the work presented in this thesis, however, a constant alignment variance is used.

In addition to the step updates, the framework is developed to allow for external measurements to be applied with ease. With external measurements, the states would be updated directly using a standard Kalman filter update. For example, with position measurements from GPS ($\tilde{\mathrm{r}}_{t,GPS}$), the measurement vector and measurement matrix shown in Equation (3.8) would be applied.

$$y = \tilde{\mathrm{r}}_{t,GPS} \quad H = \begin{bmatrix} I & O_{3\times 7} \end{bmatrix} \tag{3.8}$$

3.2.3 Further Discussion

One reason the cascade approach may be preferred in some cases is due to its segmented structure. The algorithm does not necessarily depend on both systems to be operational. If the foot-mounted IMU were to go offline, the high-level system could continue to propagate without relative state updates from the step information. In the case of lost visual odometry, the outputs of the IEZ system could be treated as the sole source of odometry. The cascade approach does, however, suffer from certain drawbacks. Since there is no correlation maintained between the foot states and the high-level states, information from an external measurement cannot be shared through all states. For example, a GPS position update, as the one shown in Equation (3.8), would not benefit the low-level IEZ filter. This lack of correlation between the two systems may be especially problematic due to certain undesirable characteristics of the foot-mounted IMU. As discussed in Chapter 2, the IEZ algorithm suffers from weak observability in the yaw and z gyro bias states under certain dynamics. This low observability may lead to biased delta heading outputs. If the system were allowed to share information, however, it could improve the observability of these states.

3.3 Centralized Fusion Approach

To deal with the drawbacks of the cascade approach, a centralized framework is proposed. The centralized algorithm joins both the torso and foot tracking states into one centralized state vector. By centralizing all involved states, the correlation between state estimates can be maintained. Therefore, the benefits of a measurement update will be shared throughout both systems. The time update of the centralized approach uses both inputs from the visual odometry and inertial measurements from the IMU. Due to the different sample rates of the two inputs, the torso states associated with the visual odometry updates

are held static between egomotion measurements, while the foot states are allowed to propagate. When an egomotion measurement is received, the torso states are updated alongside the foot states. Upon detecting alignment, a pseudo-measurement update relates the states of the foot and torso similar to the update used in the Unified Model. Similar to the cascade approach, external measurements can be applied to the system using the standard Kalman filter update. The difference here is that an external measurement will share its information with all states involved. A visual interpretation of the centralized approach can be seen in Figure 3.3.

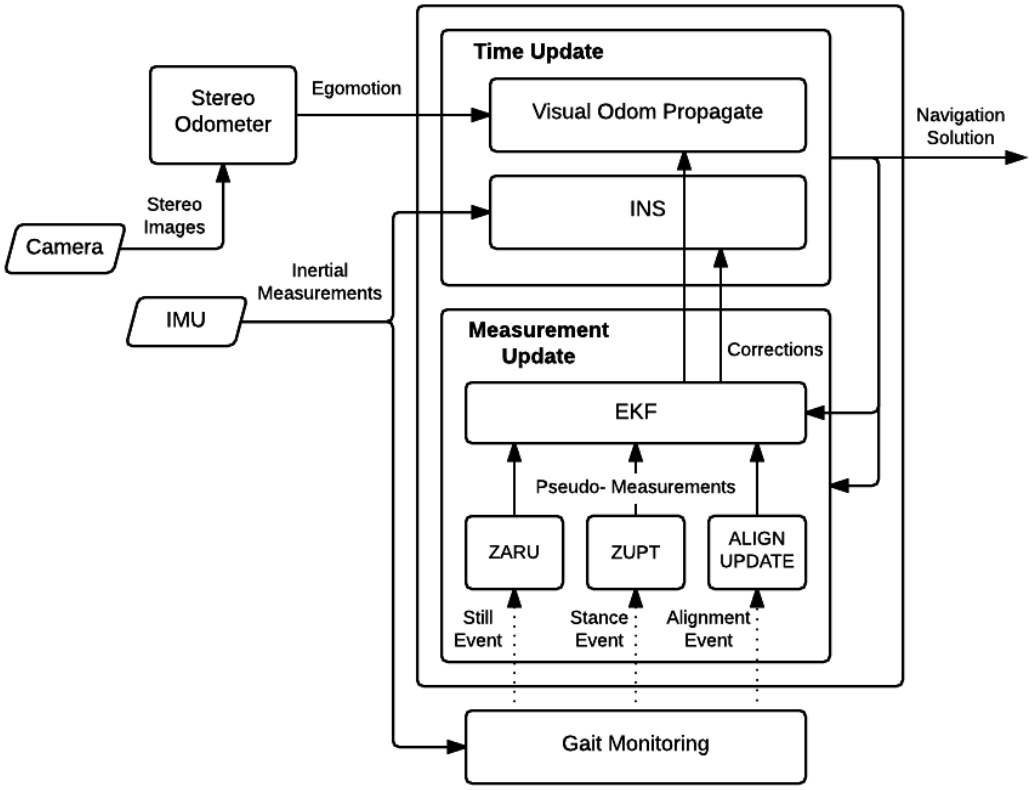

Figure 3.3: Centralized Filter Flowchart

The states involved in the centralized approach are torso position (r_t), torso orientation (φ_t), foot position (r_f), foot velocity (v_f), foot orientation (φ_f), accelerometer bias (b_a), gyroscope bias (b_g), and the boresight angle (β) as shown in Equation (3.9).

$$x = \left[\begin{array}{ccc|ccccc|c} \mathrm{r}_t & \varphi_t & | & \mathrm{r}_f & \mathrm{v}_f & \varphi_f & b_a & b_g & | & \beta \end{array} \right]^T \tag{3.9}$$

Each component of the state vector (position, orientation, velocity, bias) are 3D vectors with the exception of the boresight angle, which is a scalar value. By combining the torso and foot states and with the addition of the boresight angle, the result is a 22 variable system.

3.3.1 Initialization

To initialize the centralized states, the application must be taken into consideration. For pure dead reckoning scenarios, the position and heading can be set to zero. If the navigation is to occur with respect to some global reference frame, an initial position measurement can be used for initialization along with a guess of the initial heading. One choice may be to initialize the position with GPS position and the heading with a magnetometer.

In actuality, the foot frame and the torso frame are separated by some offset, which depends on the height of the user. However, the offset information is not necessary for the applications of this thesis. Because the offset is assumed constant, equating the foot frame and torso frame position on initialization has no adverse effect. One application in which the positions should be initialized independently is when position measurements are available for both the foot-mounted IMU and the visual odometry camera. Another example is if measurements are available on the relative pose of the two frames.

Along with an initial position and heading estimate, an estimate of the foot IMU's pitch and roll must be calculated to account for gravity's effect on the inertial measurements. For this, the same accelerometer leveling technique is used as described in Section 2.1 [5]. A correlation also exists between the torso positions, foot positions, torso heading, foot heading, and boresight angle on initialization. Since this relation is linear, an iterative least squares process is not required. The estimate covariance between the positions, headings, and boresight angle ($P_{\mathrm{r},\psi,\beta}$) is calculated using Equation (3.10)

$$P_{\mathrm{r},\psi,\beta} = \left(H^T C_z^{-1} H\right)^{-1} \tag{3.10}$$

with least squares covariance (C_z) and measurement matrix as shown in Equations (3.11) and (3.12),

$$C_z = \begin{bmatrix} P_{\mathrm{r}_t,i} & O & O & O \\ O & P_{\psi_t,i} & O & O \\ O & O & P_{\beta,i} & O \\ O & O & O & R_{align} \end{bmatrix} \tag{3.11}$$

$$H = \begin{bmatrix} I & O & O & O & O \\ O & 1 & O & O & O \\ O & O & O & O & 1 \\ I & O & -I & O & O \\ O & 1 & O & -1 & 1 \end{bmatrix} \tag{3.12}$$

where $P_{\mathrm{r}_t,i}$, $P_{\psi_t,i}$, and $P_{\beta,i}$ are the initial uncertainty in the torso position, torso heading, and boresight angle states, respectively. An alignment uncertainty is also introduced for the relation of the torso and foot states.

The resulting covariance matrices from both the accelerometer leveling and pose alignment initialization processes are used to generate the full initial state estimate covariance. In addition, an initial uncertainty in foot velocity and gyroscope bias estimates is included. The resulting full state estimate covariance matrix accounts for the cross correlations introduced upon initialization. Figure 3.4 shows these correlations by shading the cells in blue of which are populated in the initial state estimate covariance.

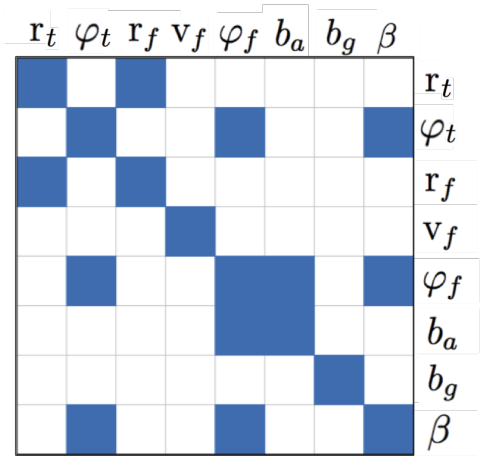

Figure 3.4: Initial State Estimate Correlations

3.3.2 Time Update

The time update for the centralized approach is very similar to that of the Unified Model as discussed in Chapter 2. Two models are used for the time update, one for the foot-mounted IMU and one for the visual odometry. Because the two inputs are provided at different rates, an adjustment must be made to the time update. For the time between inputs from visual odometry egomotion, the torso states (r_t and φ_t) are held static and the foot states are allowed to evolve according to the inertial measurement inputs. When the next visual odometry input is provided, the torso states are then updated alongside the foot states. In this method, the time update resembles both that from the Unified Model as well as the stochastic cloning Kalman filter. This process is summarized in Table 3.1.

The boresight angle is modeled as a random walk, which allows the estimate to be sensitive to updates. During long periods without updates, the boresight angle estimate will remain stationary, which is ideal for this particular scenario. The random walk model is preferred over a Markov process model in which the estimate would digress back to zero in the lack of updates [5].

Table 3.1: Centralized Time Update

	Without VO Inputs	With VO Input
Foot States	$\dot{x}_f = f(x_f, u_k, 0)$	$\dot{x}_f = f(x_f, u_k, 0)$
Torso States	$x_{t,k} = x_{t,k-1}$	$x_{t,k} = g(x_{t,k-1}, z_k, 0)$
State Transition Matrix	$\breve{\Phi}_k = \begin{bmatrix} \Phi_{f,k} & O \\ O & I \end{bmatrix}$	$\breve{\Phi}_k = \begin{bmatrix} \Phi_{f,k} & O \\ O & \Phi_{t,k} \end{bmatrix}$
Input Covariance Matrix	$\breve{Q}_k = \begin{bmatrix} Q_{f,k} & O \\ O & O \end{bmatrix}$	$\breve{Q}_k = \begin{bmatrix} Q_{f,k} & O \\ O & Q_{t,k} \end{bmatrix}$

3.3.3 Measurement Update

Just as with the cascade model, knowledge of torso and foot alignment is used. For this approach, however, the information is used as a pure pseudo-measurement. This pseudo-measurement has the effect of bringing the state estimates together while introducing cross correlation in the full state estimate covariance matrix. The pseudo-measurement equates the position of the foot frame with the position of the torso frame. The heading estimates are related such that the heading of the foot frame is equal to the torso frame heading plus the boresight angle. These pseudo-measurements can be summarized by the measurement vector and measurement matrix shown in Equation (3.13).

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \quad h(x, x_s) = \begin{bmatrix} \mathrm{r}_t - \mathrm{r}_f \\ \psi_t - \psi_f + \beta \end{bmatrix} \tag{3.13}$$

In addition to the alignment pseudo-measurement, the framework can easily incorporate external measurements. External measurements update the states directly just as with the

cascade filter approach. The main difference from the cascade approach is that external measurements affect all of the involved states. Whether the external measurements are applied directly to the foot or torso states, the correlation maintained in the state estimate covariance allows for the information to be shared throughout the entire system.

Chapter 4

Simulation

In order to assess the performance of the algorithms presented in Chapter 3, a simulation environment is developed. Using experimental data alone to assess the algorithms would be tedious, because it would involve repeating data collections multiples times to gauge the expectation of the performance. Even still, repeated experimental data collections would inevitably vary run to run. With simulated data, a trajectory can be repeated any number of times. Each run of the trajectory will have unique noise values of the same statistical parameters, as defined by the user. From this, an accurate expectation of the performance can be attained. This process of repeating simulated trajectories a high number of times with new noise values on each run is referred to as a Monte Carlo simulation.

This chapter will discuss the procedures involved in developing the simulation environment. As a precursor to developing the simulation, true human motion is analyzed using a motion capture system. By studying the motion of the foot throughout a typical step, motion profiles are developed. These profiles are used to simulate the inertial measurements of a foot-mounted IMU. In addition, a torso motion profile is developed to simulate the visual odometry egomotion. With these simulated measurements, the algorithms discussed in Chapter 3 will be used. From the results of the simulated data, an evaluation of the algorithms can be made.

4.1 Comparison With Motion Capture System

To ensure that the simulated signals are representative of true human motion, a highly accurate motion capture system is used to analyze the human gait. The motion capture system operates using an array of infrared cameras that observe the testing platform, as shown

in Figure 4.1. The cameras track the position of reflective markers through triangulation with high precision. The motion capture system used is accurate to millimeter level positioning and is the gold standard for gait monitoring tasks. A similar system has been used for tracking a foot-mounted IMU before, but in this case, only the position was tracked [27]. The orientation of the IMU throughout a step has a major impact on the IEZ algorithm's performance and therefore is important to track with the motion capture system. By placing multiple reflective markers on the foot, a calculation of the orientation can be made.

Figure 4.1: Motion Capture System

In order to track the position of the foot-mounted IMU, a reflective marker is placed on top of an Xsens MTi IMU, which is mounted to the shoe of the test subject. The tracking of the IMU's orientation requires additional markers. Due to the small size of the Xsens with respect to the reflective markers, there is not enough room to attach the additional markers rigidly to the IMU. Instead, two markers are placed on the user's ankle: one on the inside (medial) and one on the outside (lateral). Using this configuration of reflective markers, an assumption is made that the foot and IMU are rigidly connected and that the

foot rotates about an axis coinciding with the ankle. With this assumption in place, the pitch, and yaw are calculated from the 3D positions of the three markers. The relationship between the ankle and the foot does not allow for calculating an accurate roll measurement and, therefore, the roll of the foot is assumed constant throughout a step.

The results of the motion capture states of the foot through a single step can be seen in Figure 4.2. The x axis points in the stepping direction, while the y direction is lateral to the step, and z direction points downward with gravity. The x trajectory takes on a tangent hyperbolic shape, which peaks at the step length. The y trajectory rises only slightly and ends roughly where it began. The z value decreases, indicating a rise in the foot's height from the ground, and returns to roughly the same value. The pitch of the foot ends at roughly the same value at which it starts. The pitch has a non-zero initial value that corresponds with its orientation to a local level frame. The yaw also begins and ends at roughly the same value as a result of the straight step taken. The yaw, along with the positions, is initialized at zero to correspond with the choice of coordinate frame.

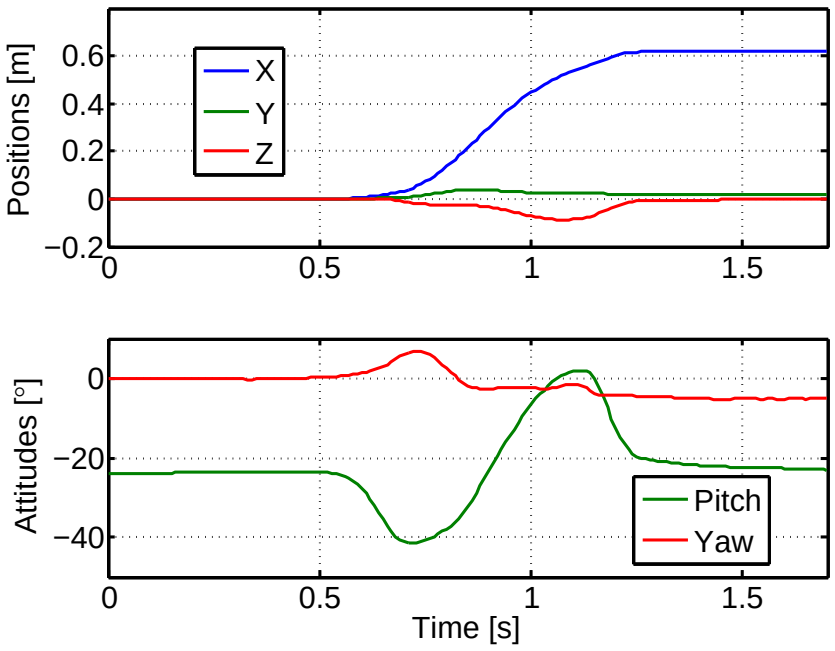

Figure 4.2: Motion Capture System Foot States

4.2 Simulation of the Foot-Mounted IMU Readings

With knowledge of the true motion of the foot throughout a typical step, a motion profile can be generated to replicate this motion in simulation. The true motion of the states seem to take on sinusoidal behavior. This knowledge can be used to generate the profiles using mathematical functions [28]. First, the x position profile ($\mathrm{r}_x(t)$), is formed using the first quarter of a sinusoid period. The profile is shaped using the step length (ℓ_s) and the step interval time (t_u) as shown in Equation (4.1).

$$\mathrm{r}_x(t) = \left(\frac{\ell_s}{2}\right)\left(1 - \cos\left(\frac{\pi t}{t_u}\right)\right) \tag{4.1}$$

The motion capture system shows that the lateral motion of the foot throughout a step is comparatively small. For this reason, the y position profile ($\mathrm{r}_y(t)$) is held constant. The z position profile ($\mathrm{r}_z(t)$) is generated in a similar fashion to that of the x. The z profile is shaped using step height (h_s) and the step interval time as shown in Equation (4.2).

$$\mathrm{r}_z(t) = -\left(\frac{h_s}{2}\right)\left(1 - \cos\left(\frac{2\pi t}{t_u}\right)\right) \tag{4.2}$$

The yaw profile ($\psi(t)$) is generated using an equation much like that of the z position. The shape of the yaw profile is determined by the max yaw (ψ_{max}), the interval time, and the yaw offset (ψ_0) as shown in Equation (4.3).

$$\psi(t) = \left(\frac{\psi_{max}}{2}\right)\left(1 - \cos\left(\frac{2\pi t}{t_u}\right)\right) + \psi_0 \tag{4.3}$$

The yaw offset is the difference between the walking direction and the yaw of the foot. This yaw offset is equivalent to the boresight angle, which was presented earlier in Chapter 3.

The equation for generating the pitch profile ($\theta(t)$) is slightly more involved. As shown in Figure 4.2, the pitch profile in a true step does not resemble a typical sinusoidal shape.

To replicate the more complex shape, a piecewise sinusoid function is used. Three profiles are defined $(\theta_1, \theta_2, \theta_3)$ as shown in Equation (4.4).

$$\theta_1(t) = \left(\frac{\theta_{max}}{2}\right)\left(\cos\left(\frac{3\pi t}{t_u}\right) - 1\right) + \theta_0 \tag{4.4a}$$

$$\theta_2(t) = (\theta_{max})\cos\left(\frac{3\pi t}{t_u}\right) + \theta_0 \tag{4.4b}$$

$$\theta_3(t) = \left(\frac{\theta_{max}}{2}\right)\left(\cos\left(\frac{3\pi t}{t_u}\right) + 1\right) + \theta_0 \tag{4.4c}$$

These profiles are shaped by max pitch (θ_{max}), the interval time, and the pitch offset (θ_0). Depending on the current time in the interval, one of the three equations will define the pitch profile. This piecewise relationship is shown in Equation (4.5).

$$\theta(t) = \begin{cases} \theta_1 \quad | \quad for \ 0 < t < t_u/3 \\ \theta_2 \quad | \quad for \ t_u/3 < t < 2t_u/3 \\ \theta_3 \quad | \quad for \ 2t_u/3 < t < t_u \end{cases} \tag{4.5}$$

Since the roll information cannot be calculated with the configuration of the motion capture reflectors described earlier, a prediction must be made of the roll of the foot throughout a step. It is assumed that the changes in roll experienced throughout a single step are small compared to the amount of pitch variation. With this assumption in place, the roll profile $(\phi(t))$ is simply held at a constant value of zero throughout a step.

At this point, a motion profile has been generated for the interval phase of a human gait. Another key aspect is the stance phase when the foot satisfies zero velocity conditions. In the stance phase, foot states are held static for a set amount of time, which will be referred to as stance time (t_s). The static period is appended to the interval profile to make up a profile for a single step. From here, the user can define how many steps are present in a simulation and stack the individual step profiles for a full simulated trajectory.

Table 4.1: Foot Simulation Variables

Step Interval Time	t_u	0.8 s
Stance Time	t_s	0.3 s
Step Length	ℓ_s	1.3 m
Step Height	h_s	0.14 m
Max Yaw	ψ_{max}	10 °
Yaw Offset	ψ_0	15 °
Max Pitch	θ_{max}	30 °
Pitch Offset	θ_0	−20 °

In order for the profiles to match that of a typical walking step, the step profile parameters shown in Table 4.1 are used. The values in Table 4.1 were chosen by examining the motion capture profiles in addition to experimentally collected IMU data. The generated profiles match closely to the profiles measured with the motion capture system and can be seen in Figure 4.3.

The inertial values are then generated from the full motion profile, which contains the 3D position and orientation of the foot with respect to the navigation frame. The initial step in generating inertial values is to take the first derivative of the orientations, which equates to the rotation rate in the navigation frame (ω_b^n) as shown in Equation (4.6),

$$\omega_b^n = \frac{d\varphi(t)}{dt} \tag{4.6}$$

where $\varphi(t)$ is a vector containing the three attitude profiles ($\varphi(t) = [\phi(t), \theta(t), \psi(t)]^T$). The rotation rates are then transformed into the body frame of the IMU using the mechanization matrix as discussed in detail in Appendix A.1. The inverse of the mechanization matrix is used to solve for the rotation rate in the body frame as shown in Equation (4.7).

$$\omega_b^b = M^{-1}\omega_b^n \tag{4.7}$$

The result of Equation (4.7) is a vector containing the true rotation rates provided by a perfect three-axis gyroscope.

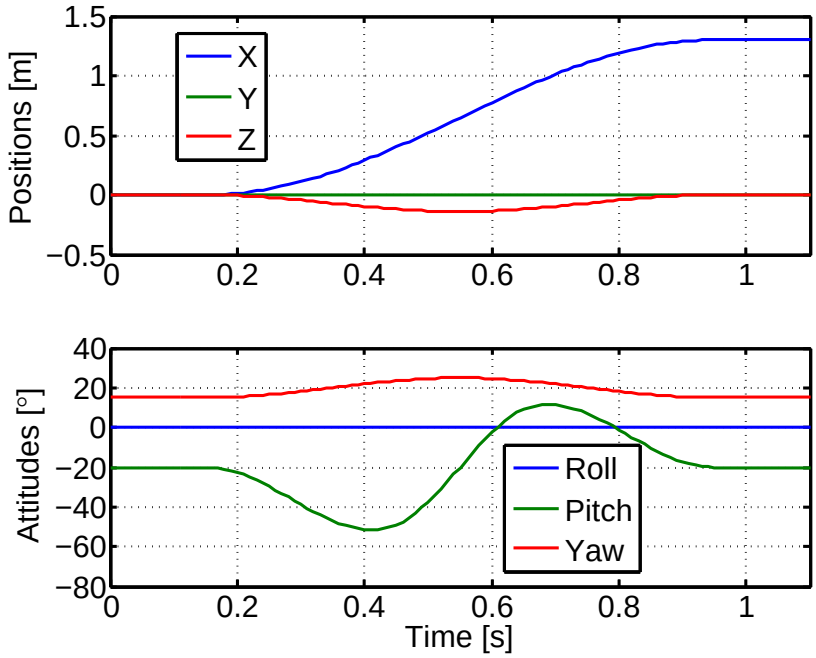

Figure 4.3: Simulated Foot States

Next, the specific force readings must be simulated. The second derivative of the 3D position is taken, which yields the acceleration of the body in the navigation frame as shown in Equation (4.8),

$$a_b^n = \frac{d^2 r(t)}{dt^2} \tag{4.8}$$

where $r(t)$ is a vector containing the three position profiles ($r(t) = [r_x(t), r_y(t), r_z(t)]^T$). Gravity is then accounted for, which acts in the down direction of the navigation frame. The result of adding gravity, as shown in Equation (4.9), is the specific force in the navigation frame.

$$f_b^n = a_b^n + g^n \tag{4.9}$$

A rotation matrix is formed using the true orientation and is used to rotate specific force from the navigation frame to the body frame as shown in Equation (4.10).

$$f_b^b = C_b^n f_b^n \tag{4.10}$$

Now that the values that correspond with a perfect IMU have been calculated, an error model is used to replicate realistic IMU readings. A bias and white noise are added to the true inertial values. The white noise is formed using a normal distribution and the bias is modeled as a first order Markov process, as discussed in Chapter 2.

4.3 Simulation of the Visual Odometry Egomotion

Similar to the IMU simulation, a motion profile is generated for the camera trajectory. For this, a constant longitudinal velocity is used. Motion is restricted in the lateral and vertical direction and changes in orientation are also restricted. The profile is generated in such a way that the camera aligns with the foot once per step, at the point where the step is in the stance phase. The relationship between the positions of the two frames can be seen in Figure 4.4.

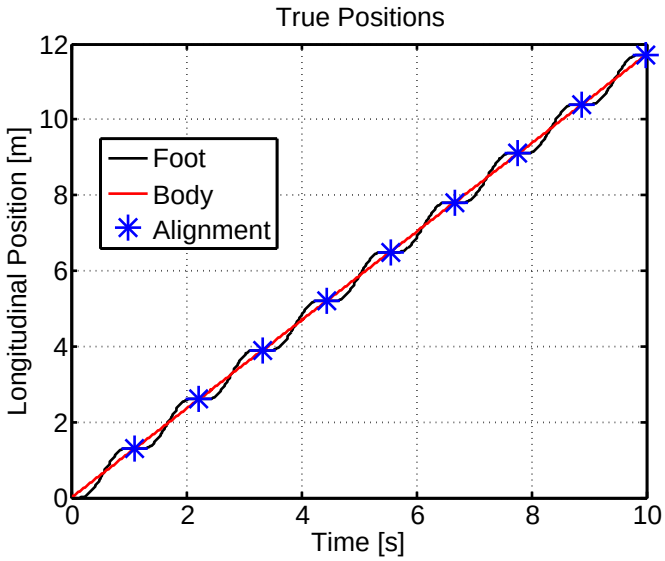

Figure 4.4: Simulated Positions

The visual odometry egomotion is calculated as the change in pose between consecutive frames. To simulate these inputs, a sample rate is chosen and the motion profile is differenced to calculate the true egomotion. Noise is then added to these values, which is chosen to be zero-mean Gaussian.

4.4 Results

The simulated signals are then processed through the algorithms introduced in Chapter 3. Monte Carlo simulations are performed to gain an expectation of each algorithm's performance. From this, a comparison is made of the four different localization solutions, including:

- Stand-alone IEZ
- Stand-alone visual odometry
- Cascade fusion
- Centralized fusion

In order for the proposed algorithms to be of any benefit, they should be able to outperform both stand-alone algorithms in terms of a position and orientation solution.

4.4.1 INS-EKF ZUPT

The Monte Carlo simulation results can be used to characterize the IEZ solution mentioned earlier in the text. The most impactful characteristic of the IEZ solution is the unobservable z gyroscope bias state during continuous walking. Figure 4.5 shows the root mean squared error (RMSE) of the gyroscope bias states and their evolution with time. In the initial stationary period, the RMS errors of all three gyro bias states decrease. From here, the x and y gyro bias errors remain bounded while that of the z gyro grows with time. This reinforces the belief that the x and y gyro bias states are observable and that the z

gyro bias state is not. The observability of the x and y gyro bias states is due to the tilt effect introduced by the gravity vector. Similar to the accelerometer leveling process for tilt angle initialization, the relationship between the tilt angles and gravity is maintained in the state transition matrix. The result bounds tilt angle errors and, therefore, bounds x and y gyro bias errors [3].

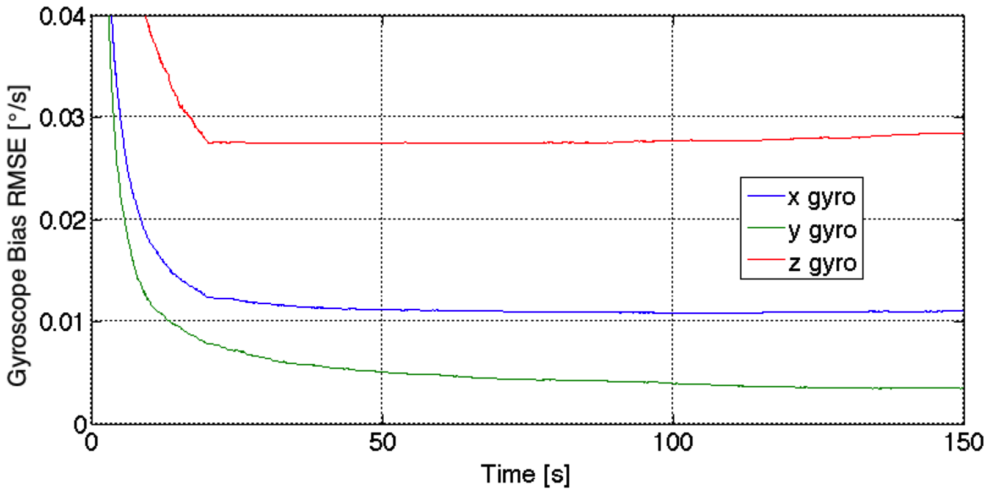

Figure 4.5: IEZ Gyroscope Bias RMS Errors

While the growth of the z gyro bias RMS error seems slight, the impact on the yaw state is significant. Any errors in the z gyro bias estimate are compounded in the yaw state due to integration. Figure 4.6 shows the RMS errors of the attitude state estimates with time. Here, the unbounded error growth of the yaw state can be seen. Yaw errors are also undesirable because of their influence on position errors. Figure 4.6 also shows that the roll and pitch estimate errors remain bounded for reasons discussed in the previous paragraph.

The simulation environment also allows for the analysis of the IEZ output extraction method. To confirm that the variances calculated for the IEZ outputs are viable, their values are compared to an empirical mean squared error. Figure 4.7 shows this comparison in terms of expected standard deviation and RMS error.

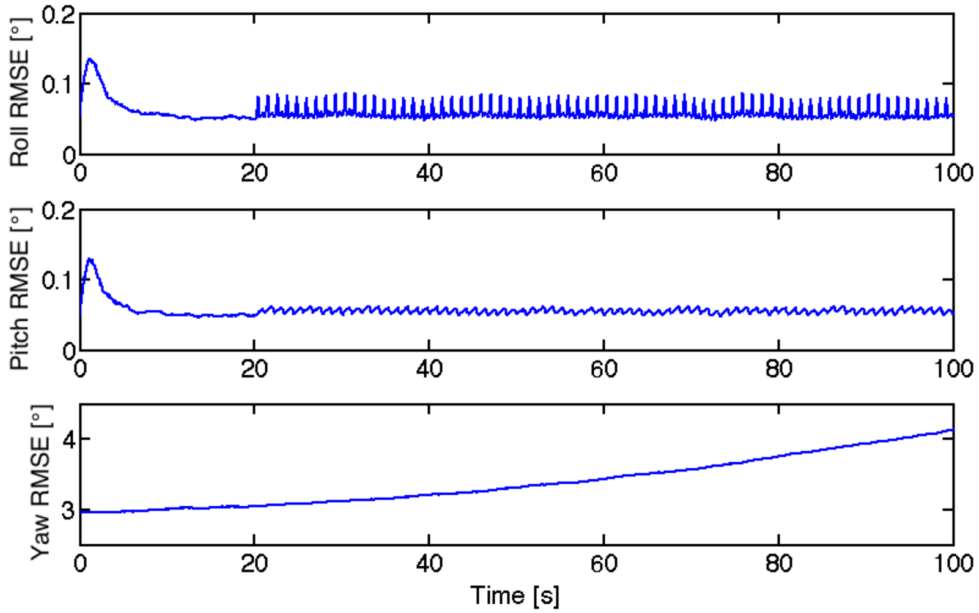

Figure 4.6: IEZ Attitude RMS Errors

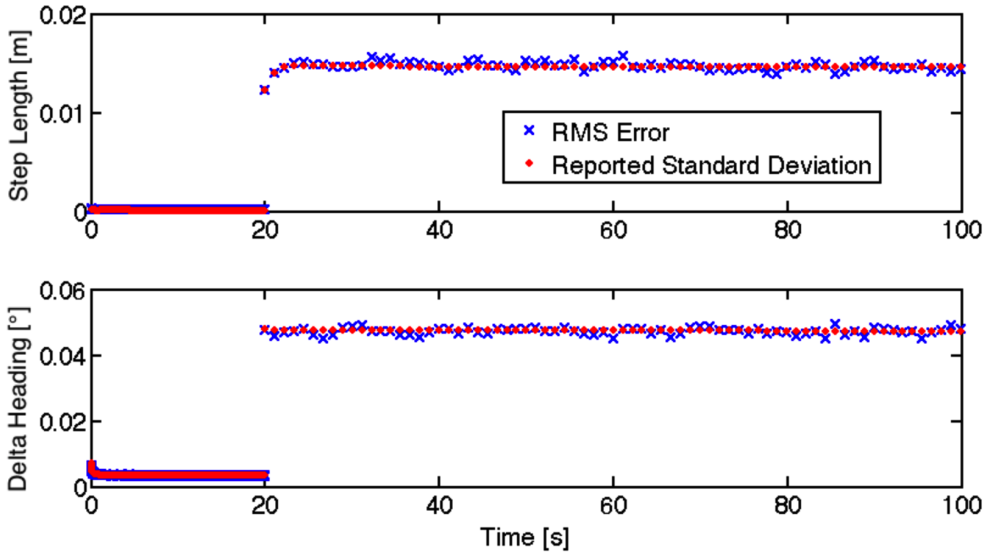

Figure 4.7: IEZ Output RMS Errors and Standard Deviation

4.4.2 Fusion Approaches - Dead Reckoning Performance

Now that a characterization has been made on the IEZ solution, the proposed sensor fusion approaches can be considered. First, an evaluation is made of the algorithms for pure dead reckoning. In this scenario, the only updates to the system are pseudo-updates

including 1) zero velocity updates 2) zero angular rate updates and 3) alignment updates between the foot frame and the torso frame.

One concern may be the ability of the centralized filter to estimate boresight angle in the dead reckoning scenario. To test this potential limitation, a simulation is performed in which the initial boresight estimate is set to zero with the true boresight angle as 15°. The transients of the boresight estimate exhibit a convergence towards the true boresight angle as shown in Figure 4.8. Note that for this particular set, it takes nearly 30 seconds to converge. This lengthy convergence time is due to the initial boresight error being well outside the initial boresight estimate uncertainty, meant to exaggerate the convergence of the state.

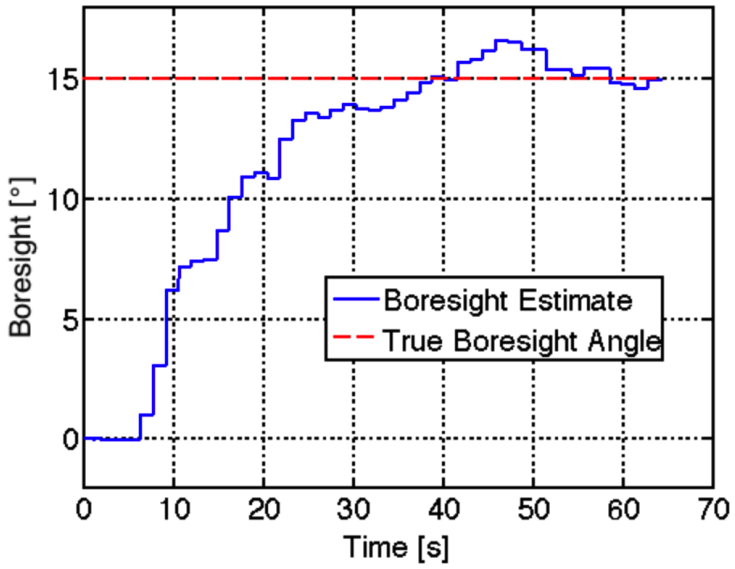

Figure 4.8: Dead Reckoning Boresight Estimation

By utilizing the Monte Carlo simulations, the expectation of the boresight angle estimation can be studied. Figure 4.9 shows the RMS errors of the boresight angle estimate with time. During the first 20 seconds of the simulation, while the system is stationary, the boresight estimate errors grow continuously. It is not until the walking motion that the boresight state can be estimated, at which point on the RMS errors remain bounded with time. The unobservability of the boresight state during stationary periods suggests that, in practice, stationary periods should be limited in time. A balance must be made to

allow for the estimation of gyroscope biases with zero angular rate updates, while limiting error growth in the boresight estimate. Considering both experimental and simulation data, stationary periods between 10 to 20 seconds were found to be ideal.

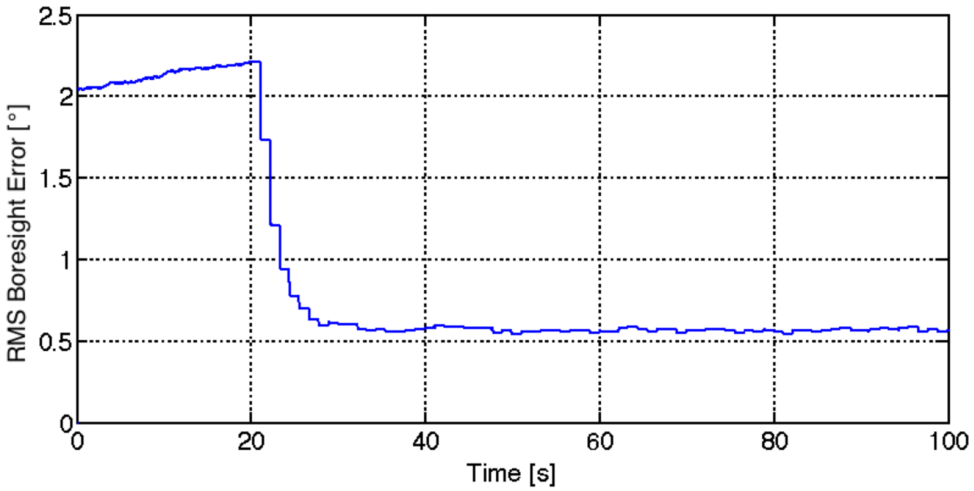

Figure 4.9: Boresight Angle RMS Errors

After confirming that the centralized filter is capable of estimating the boresight angle in pure dead reckoning, the localization results are compared against the other navigation solutions. Once again, the RMS errors of the fusion algorithms are analyzed to gain an expectation of the solution drift vs. time. Figure 4.10 shows the results of the notable states from the four algorithms (stand-alone IEZ, stand-alone visual odometry, cascade fusion, and centralized fusion).

In Figure 4.10, it can be seen that both the cascade and the centralized solution outperform the stand-alone systems it terms of drift rate. In fact, the two fusion algorithms have very similar errors, with the centralized only slightly outperforming the cascade. The centralized approach shows the most improvement over the cascade approach in the heading estimation. This is because of the heading alignment updates, which actually improve the estimation of the z gyro bias. The last subplot of Figure 4.10 shows the centralized z gyro bias RMS error compared to that of the IEZ filter. The z gyro bias error of the IEZ solution is comparatively higher and grows at a faster rate. This shows that the centralized algorithm

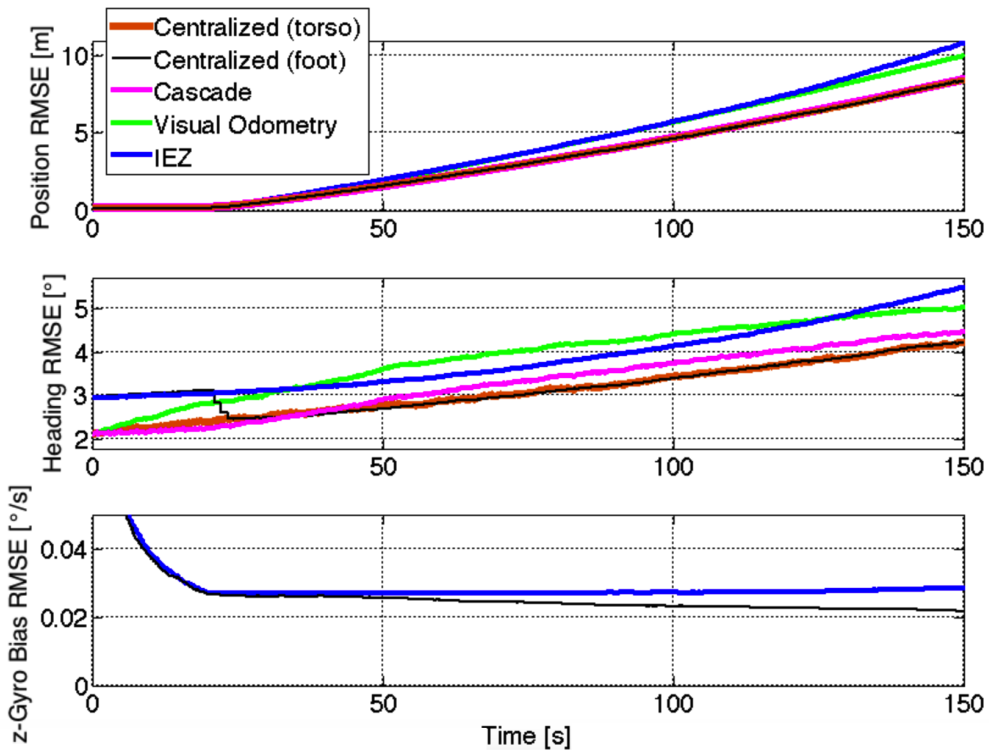

Figure 4.10: Monte Carlo RMS Error

is more apt to utilizing the complementary aspects of the two systems during the walking phase.

Another telling portion of this simulation is in the first 20 seconds, when the system is static. A close up of the RMS errors shown in Figure 4.10 can be seen in Figure 4.11. During this stationary period, alignment updates are applied to both the cascade and centralized filter once every second. The alignment updates are successful in significantly reducing position error growth for both fusion algorithms, while the visual odometry error grows at a much higher rate. As for the heading estimates, the cascade solution experiences the lowest amount of drift during the stationary period. While the centralized heading does drift at a lower rate than the visual odometry, it is outperformed by the cascade solution, likely due to the growth in boresight uncertainty. Notice that the foot and torso heading RMS error evolve separately during the stationary period. Once motion occurs, the foot heading RMS

error drops to meet the torso heading RMS error in roughly three steps, due to the increase of boresight observability.

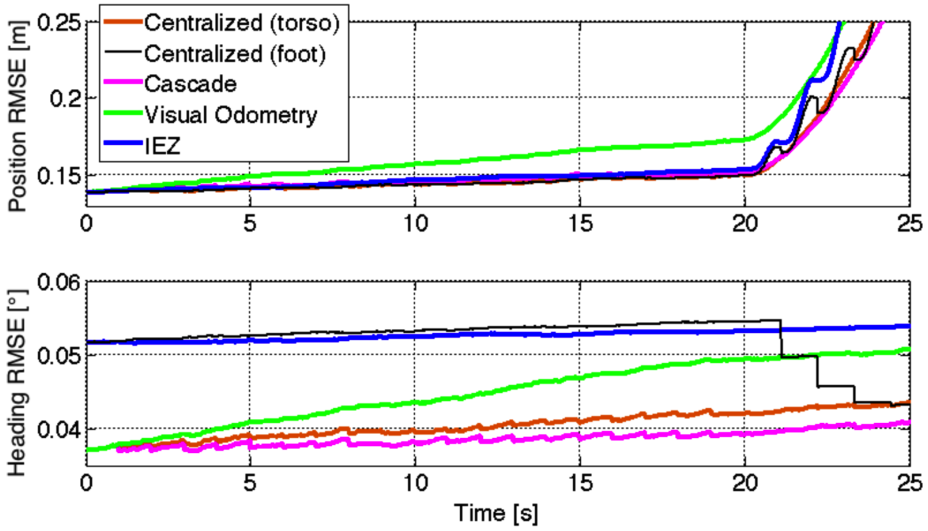

Figure 4.11: Stationary Period RMS Pose Error

4.4.3 External Measurements

To characterize the filter's response to external measurements, GPS position measurements were simulated and used to update the torso states of both the cascade and centralized systems. The position updates successfully bound the errors of the two fused solutions as shown in Figure 4.12. A close up of the RMS error can be seen in Figure 4.13, where vertical lines are drawn to indicate the alignment and GPS measurement epochs. By taking a closer look at the RMS error of the centralized solution, it can be seen that the position updates actually reduce the error of the foot states, even though the update is not applied directly to these states. The reduction in centralized foot state estimate errors is due to the correlation maintained between the torso and foot states in the state estimate covariance. The GPS updates even assist in the estimation of the z gyroscope bias. The z gyroscope bias RMS errors of both the centralized filter and stand-alone IEZ are compared in the last subplot of Figure 4.12, which shows the benefit of applying position updates. With the cascade approach lacking the ability to update the low-level IEZ algorithm, a higher steady state error

results when compared to the centralized solution, even though the same GPS measurements were used.

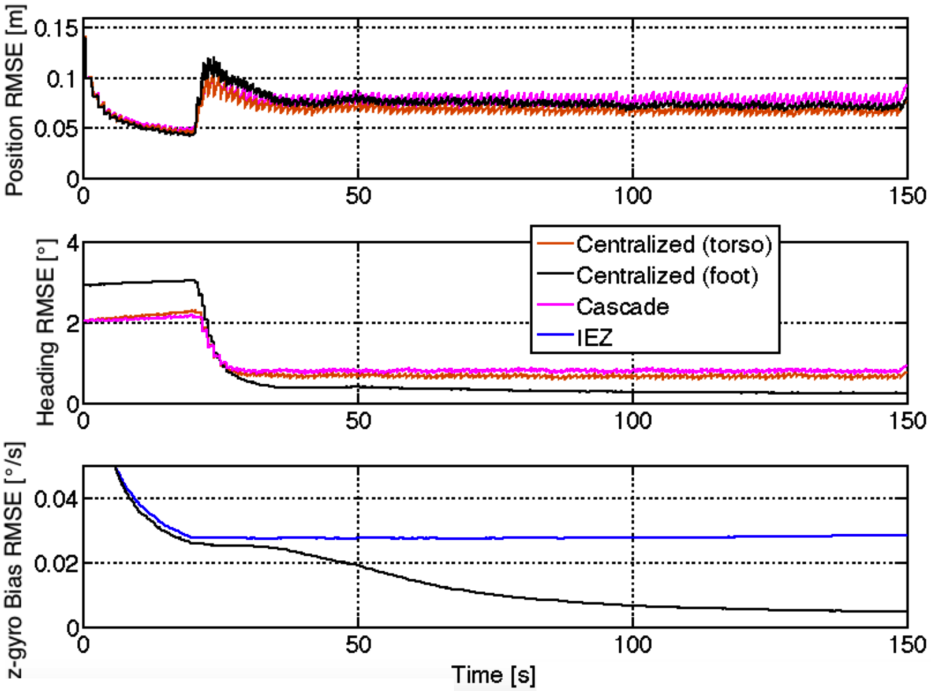

Figure 4.12: RMS Error With GPS Updates

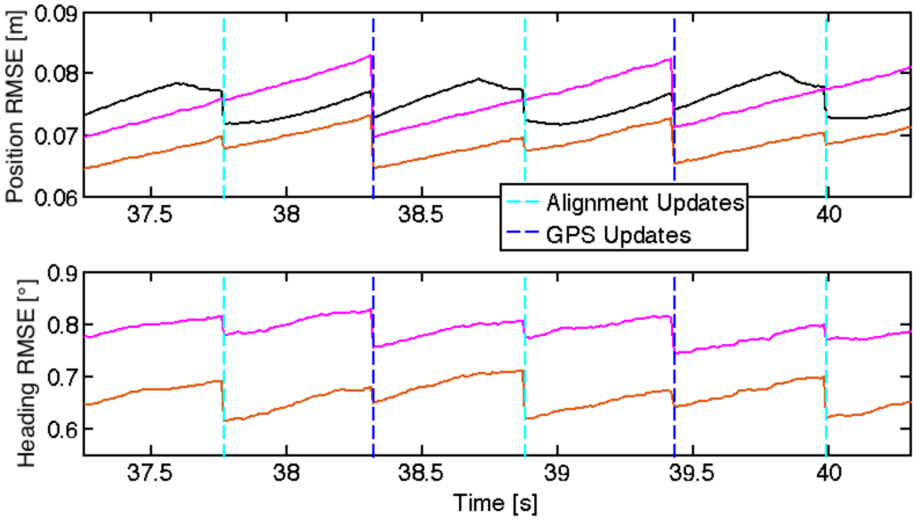

Figure 4.13: RMS Error With GPS Updates - Close Up

Chapter 5

Experimental Validation

In order to validate the findings of Chapter 4, the proposed cascade and centralized algorithms are tested with experimental data. It is difficult to draw conclusions about dead reckoning systems using experimental data, but it is beneficial to see how the systems react with real life disturbances. This chapter shows the results of the centralized and cascade fusion algorithms both in dead reckoning scenarios and with GPS position updates. The navigation solutions are compared to a reference in order to make an assessment on the performance of the two algorithms. For indoor scenarios, a floor plan of the building is used as a visual reference and for outdoor tests the solutions are compared with GPS.

5.1 Setup

An Xsens MTi is used for the foot-mounted IMU. This sensor provides triaxial specific force and rotation rate measurements in a compact packaging. The sensor is mounted to the user's foot in a secure fashion as shown in Figure 5.2. A Point Grey Bumblebee2 Stereo Camera is used for collecting stereo images for visual odometry. The stereo camera is mounted to the user by a chest mount, which can be seen in Figure 5.1. The stereo camera provides images to LibViso2 open source visual odometry library, which calculates the camera egomotion measurements [16].

A GPS unit is used for evaluating the localization performance of outdoor experiments. The GPS unit is used for comparing the dead reckoning solution to an absolute position. The drifts in the dead reckoning solutions (with enough distance traveled) will be significantly higher than the error in the GPS solution. For this reason, a stand-alone GPS solution is sufficient. The GPS solution is provided from the internal GPS receiver of the Xsens MTi-G.

Figure 5.1: Stereo Camera

Figure 5.2: Foot-mounted IMU

Two main experiments were performed: indoor and outdoor. The indoor test results are meant to evaluate the algorithm's performance on a path in which multiple turns are involved. In this test, a closed-loop path is taken in which the user ends at the start location. For closed-loop paths, the final position estimate is effectively the position error that was accumulated throughout the run. The estimated trajectory of the indoor tests is plotted over a floor plan of the building for making a qualitative assessment. The outdoor experiment allows for the localization solution to be compared with GPS. In order to calculate an accurate error as a percent of distance traveled, a straight path was taken with a total distance of approximately 95 meters. Another outdoor test involved walking the entirety of a 400 meter athletic track and was used to assess the navigation solution's long distance performance. Each experiment starts with an initial stationary period of 20 seconds to allow for the gyroscope biases to be estimated. After the initial stationary period, the user maintains a steady walking pace throughout the test, resulting in an absence of zero angular rate conditions.

5.2 Results

The bird's eye view of the position solutions of the indoor route along with the floor plan of the building can be seen in Figure 5.3. The IEZ solution drifts slightly more than the

visual odometry in this run. The fusion of the IEZ and visual odometry in the centralized filter seems to follow the floor plan for most of the trajectory until halfway through the final stretch where it is pulled towards the IEZ solution. The cascade solution relies heavily on the visual odometry throughout most of the run. The cascade solution's reliance on visual odometry results in poor positioning around the second turn, but ultimately leads to a reasonable final position error. The final position error for all navigation solutions is listed below.

- IEZ - 3.56 (m)
- Visual Odometry - 1.69 (m)
- Cascade - 1.12 (m)
- Centralized - 1.67 (m)

The position solutions for the first outdoor test can be seen in Figure 5.4 from a bird's eye view plotted over a satellite image of the test sight. The positions are plotted along with GPS, which is used as a reference for truth. This dataset is particularly interesting because the visual odometry and IEZ solution both drift in the same direction. While the cascade solution lies between the IEZ and visual odometry, the centralized solution actually tends towards the true path. With GPS as a reference throughout the run, it is possible to calculate a position error as a percentage of distance traveled, which is shown in Figure 5.5. A straight trajectory was chosen for this run because it provides the most accurate results for calculating the error as a function of distance traveled.

The second outdoor dataset was collected to study a large closed-loop trajectory with GPS again used only as a truth reference. This set consists of a full lap around a 400 meter long running track. A bird's eye view of the navigation solutions can be seen in Figure 5.6. The 2D position error is plotted versus time in Figure 5.7. From these results, it can be seen why final position error is not always the best way to assess a dead reckoning solution. While

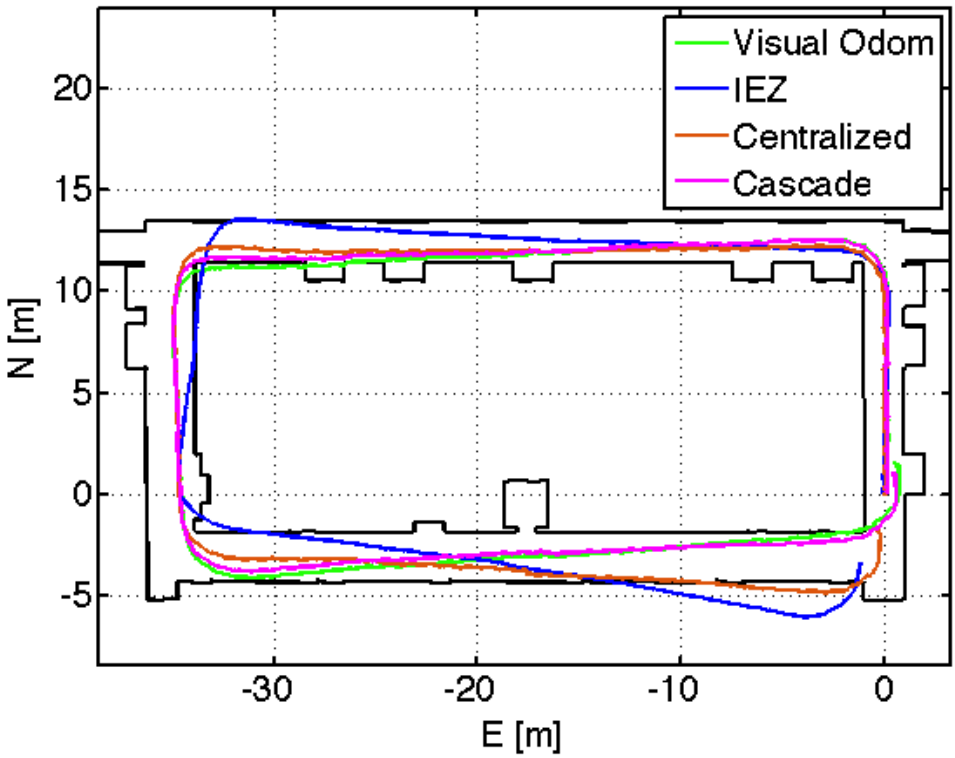

Figure 5.3: Bird's Eye View - Indoor

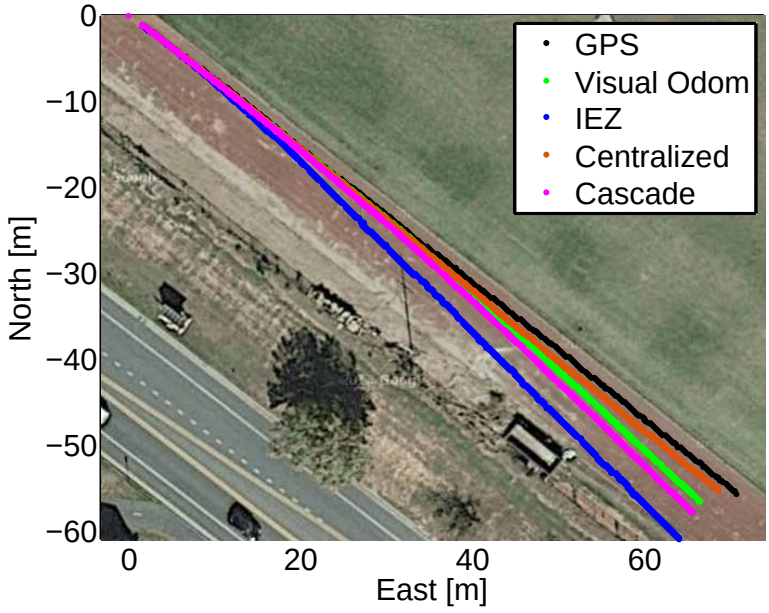

Figure 5.4: Bird's Eye View - Outdoor

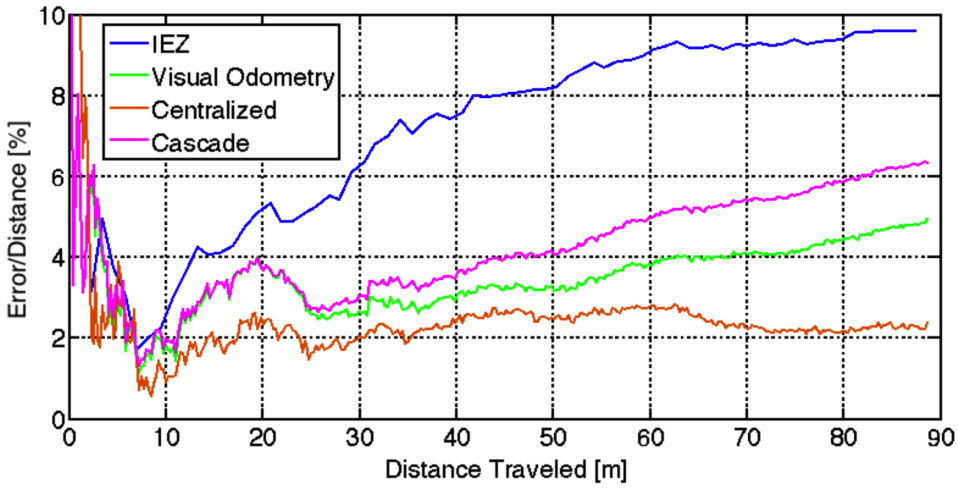

Figure 5.5: Error As Percentage of Distance Traveled

the cascade solution has the lowest error at the end, the centralized filter better estimates position throughout a majority of the run.

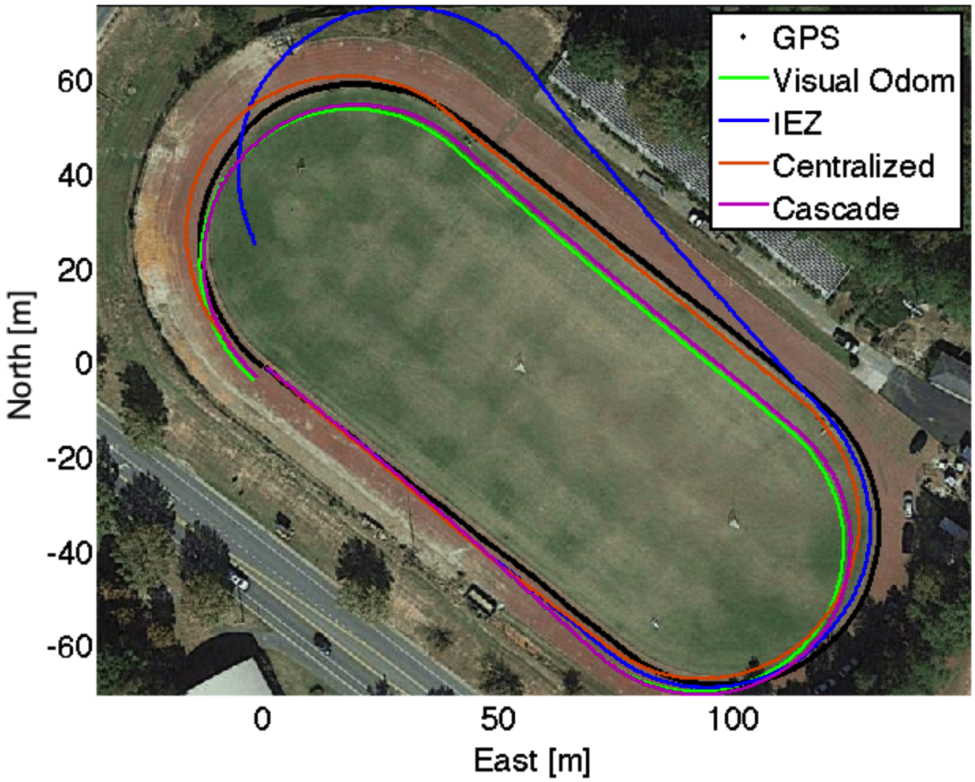

Figure 5.6: Bird's Eye View - Outdoor Loop

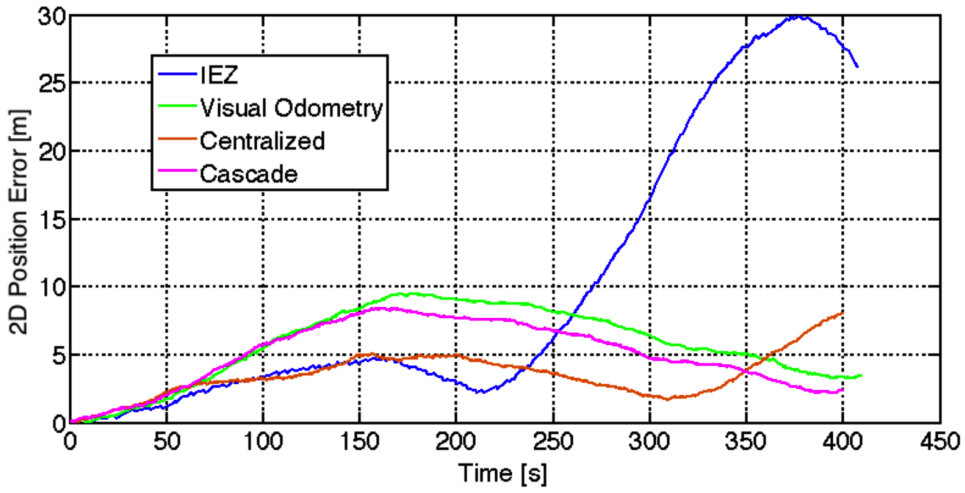

Figure 5.7: Position Error vs. Time - Outdoor Loop

With GPS position measurements, the algorithm's abilities to utilize external measurements can be assessed. The cascade and centralized filters were provided position updates for the first straightaway of the 400 meter track. After the initial phase of position updates, the systems used dead reckoning for the remainder of the path around the track. The purpose of the initial position update phase is to highlight the centralized filter's ability to share information throughout all states. With GPS position updates, the navigation solution of the centralized filter is not only aided throughout the update phase, but also in the time following. The results of this test can be seen from bird's eye view in Figure 5.8. In this figure, the centralized solution seems to maintain a good heading estimate, even after removing GPS updates. The 2D positional errors can be seen plotted versus time in Figure 5.9. This figure shows that the centralized solution experiences less drift throughout the run.

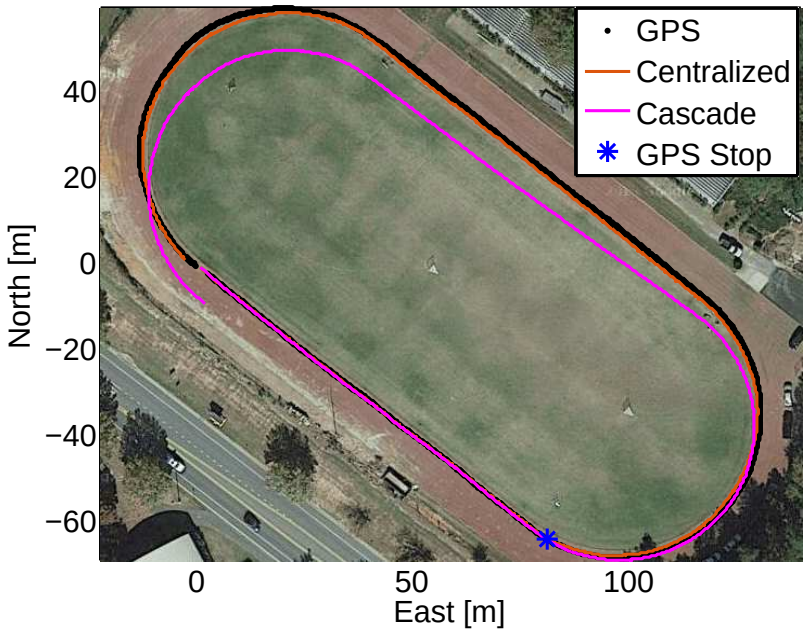

Figure 5.8: Bird's Eye View - Partial Trajectory GPS

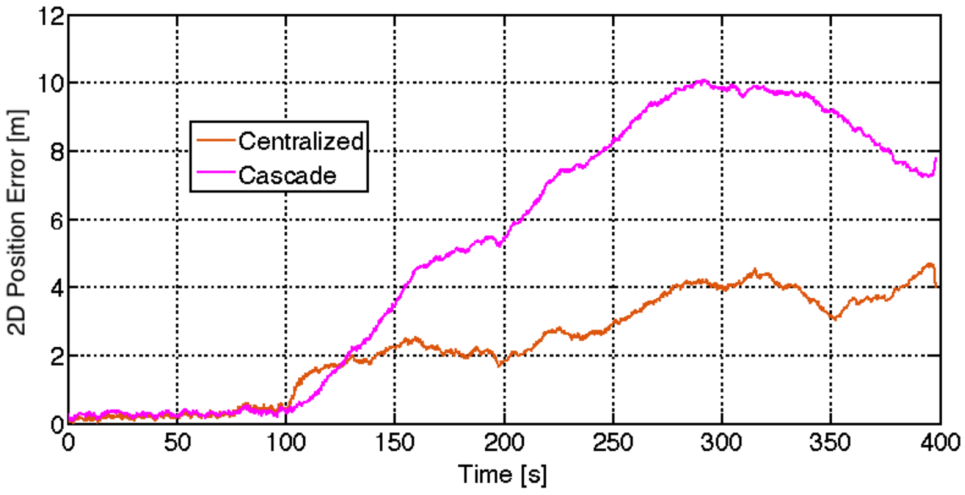

Figure 5.9: Position Error vs. Time - Partial Trajectory GPS

Chapter 6

Conclusions and Future Work

The problem of pedestrian navigation is a challenging task for many reasons. The motion of a human is complex and often difficult to predict. Pedestrians often require navigation in environments with little to no GNSS coverage. While an absolute position solution can be estimated using infrastructure-based systems, these methods can be costly when scaling for widespread usage. One option, presented in this thesis, is to utilize an array of body-worn devices providing information of the user's motion. By fusing motion information into a dead reckoning solution, a user's position and orientation can be tracked between absolute position updates. Theoretically, the more motion information used, the longer a dead reckoning solution will be valid. The fusion of relative state information is a challenge for any system, especially pedestrians. The additional challenge for localizing pedestrians comes from the use of body-worn motion devices, which are often located at different locations throughout the user's body.

A focus is made in this thesis on a popular technique for dead reckoning of a pedestrian: the mechanization of a foot-mounted IMU. The process was described in detail and the outputs of the system characterized. The foundation of the IEZ algorithm is the detection of zero velocity and zero angular rate conditions. By studying a window of inertial measurements, these conditions are monitored. With the sensor being mounted at the foot location, zero velocity conditions are detected at each step and used to update the internal errors of the IMU. Literature states that a major fallback to the IEZ algorithm is the unobservability of the z gyro bias while walking [3]. While implementing a zero angular rate update allows for the estimation of the gyroscope biases, these conditions are rarely met in typical walking

motion. This trait and its effects on the navigation solution were shown through simulation and analysis of Monte Carlo results.

While the IEZ algorithm tracks a 6 DOF pose of the foot-mounted IMU, this much information is not entirely relevant for high-level fusion. A technique was presented for calculating outputs of the IEZ along with a covariance of these values. By way of Monte Carlo analysis, this technique for calculating IEZ outputs was validated. The outputs include step length and delta heading, which alone can act as an odometry source. These outputs are also affected by poor z gyro bias observability. As the time between zero angular rate update increases, a bias is introduced to the delta heading values.

By fusing the foot-mounted IMU solution with additional navigation systems, the adverse effects can be reduced. The mounting of an IMU to the foot is ideal for estimating internal errors of the device, but makes it especially challenging to fuse with other body-worn systems. Utilizing the gait monitoring results, which act on the raw inertial measurements, the instance of torso-to-foot alignment can be approximated. Once torso alignment is detected, the information from the foot-mounted IMU can be combined with other body-worn systems.

Two algorithms for fusing the foot-mounted IMU with other body-worn sensors were presented: a cascade and a centralized approach. Each of the algorithms has both positive and negative characteristics, which were discussed in the algorithm's respective sections and when describing results. To serve as an example, a fusion of the foot-mounted IMU with a chest-mounted visual odometry system was considered.

In order to characterize the two methods, a simulation environment was developed. Studying the motion of the foot sensor with a high accuracy motion capture system allowed for mathematical formulas of the step motion to be developed, which mimic the position and orientation of the foot throughout a step. By differentiating these values and incorporating the effects of gravity, outputs of a perfect IMU are simulated. Outputs of a visual odometry system are simulated in a similar way, with the true trajectory aligning with the foot once per

step. After perfect measurements are simulated, error models are incorporated to replicate the outputs of the actual sensors. The development of a simulation environment allowed for Monte Carlo runs to be performed in order to make assessments of an algorithm's error drift performance.

The cascade approach uses the step length and delta heading outputs of the IEZ to update a visual odometry system in a high-level EKF. This scheme may be preferred in application due to its simplified architecture. It does not rely on both systems to be operational to continue providing a navigation solution. With implementation in a Kalman filter framework, the incorporation of external measurements is very straightforward. One drawback to the cascade filter approach, however, is that the correlation between the torso and foot states is not maintained. Monte Carlo simulations showed that this aspect is not critical for dead reckoning scenarios, but when external measurements are applied to either the foot or torso states, the cascade approach does not take full advantage of this information. In addition, the errors in the low level IEZ heading eventually introduce bias in the delta heading outputs. As time between ZARU measurements increases, the performance of the cascade filter degrades.

The second algorithm proposed combines the torso states and foot states into a centralized state vector for the sake of maintaining a correlation between all states. The Monte Carlo results show that the centralized approach aids in the estimation of the z gyroscope bias. As a result, the heading error drift rate is improved when compared to the cascade solution. The main drawback to the centralized approach is the complex architecture. The centralized approach is more difficult to implement, primarily due to the differing sample rates of the IMU and visual odometry system. Another characteristic of the centralized filter is the lack of boresight angle observability during stationary periods, which was shown using Monte Carlo analysis. While this trait negatively affected the results of the simulation, it may actually be a benefit in application. In real life, the user is prone to fidgeting in

place, causing actual uncertainty in the boresight angle. The centralized filter's boresight uncertainty growth might be more accurate in capturing this phenomenon.

To show how external measurements affect the systems, position updates were applied to both simulated and real data. Simulation showed how the position updates to the torso states indirectly improved the foot states. The foot state errors were reduced with GPS position updates, not only in position and heading, but also in the estimation of the z gyro bias. This assistance in z gyro bias estimation enhances the navigation solution for the centralized approach even after position measurements are removed. While the cascade filter also showed benefits from position measurements, the steady state error was higher than that of the centralized solution. The centralized filter was also the better performer in the experimental tests by better taking advantage of the GPS position measurements applied in the first 100 seconds of the run.

So when choosing between fusion techniques, the application must be considered. Because the centralized approach is more difficult to implement, it may be preferable to use the cascade approach. The Monte Carlo results show that this may be an acceptable option, but only if the time between ZARU conditions is limited. Also, when the application calls for extended stationary periods, it may be preferable to use the cascade approach, which does not rely on the estimation of a boresight angle.

There are still many improvements that could be made to build off the work presented in this thesis. For one, future work could be made to enhance some of the low-level operations involved in both the cascade and centralized algorithms. In this thesis, a focus was not made on optimizing factors such as the zero velocity/zero angular rate detection, visual odometry parameters, sensor mounting locations, and so on. Improvements made to any of these aspects will surely enhance the overall fusion result.

One simple adjustment that can also be made and may show significant overall improvements would be in the choice of covariances used in the alignment updates. The values

chosen for the alignment covariance in this thesis were conservative to account for the uncertainties in the transients of walking or when turning. A smart approach would be to detect the user's walking phase and to scale the alignment covariance values accordingly. For example, upon noticing a large increase in visual odometry rotation rate, the user could be assumed to be turning, in which case the alignment variance should be inflated. Also, a high alignment variance could be assigned on the first step after long stationary periods to account for uncertainty in the transients of walking.

Another improvement would be the incorporation of a model and/or measurement relating the foot and torso states. This thesis only considered the case in which pseudo-measurements between the torso and foot states occurred once per step. It has been proven that the more frequent these pseudo-measurements occur, the more the navigation solution will be improved [6]. The centralized architecture would be well suited for incorporating this additional information. As an example, consider the case in which a walking gait model predicts the relative pose between the foot and the torso throughout a step. In another example, a downward facing camera could track the foot's relative orientation and provide measurements of the boresight angle. It would also be feasible to utilize devices that provide range between the foot-mounted IMU and the camera. In any of these examples, the centralized approach could be slightly modified to include an additional measurement that relates the foot and torso states.

The work presented in this thesis lays a foundation for multi-sensor fusion for pedestrian navigation to come. The two proposed frameworks for fusion of a foot-mounted IMU are easy to implement and should therefore encourage future researchers to incorporate into other pedestrian navigation tasks. The two methods are also flexible enough to allow for additional measurements, which opens up the potential for continued improvements. The contribution of this thesis will advance the topic of pedestrian localization and of navigation in general.

Bibliography

[1] J. D. Pierce and D. M. Bevly, "A centralized approach to pedestrian localization using multiple odometry sources," in *Proceedings of the 2015 International Technical Meeting of The Institute of Navigation*, 2015.

[2] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, vol. 25, no. 6, pp. 38–46, Nov 2005.

[3] A. Jimenez, F. Seco, J. Prieto, and J. Guevara, "Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU," in *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, March 2010, pp. 135–143.

[4] J.-O. Nilsson, A. K. Gupta, and P. Handel, "Foot-mounted inertial navigation made easy," *In proc. IPIN 2014*, Oct. 28-30, 2014.

[5] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[6] P. Crocoll, L. Gorcke, G. F. Trommer, and F. Holzapfel, "Unified model technique for inertial navigation aided by vehicle dynamics model," *ION ITM*, 2013.

[7] M. Koifman and I. Bar-Itzhack, "Inertial navigation system aided by aircraft dynamics," *Control Systems Technology, IEEE Transactions on*, vol. 7, no. 4, pp. 487–493, Jul 1999.

[8] S. Roumeliotis and J. Burdick, "Stochastic cloning: a generalized framework for processing relative state measurements," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, 2002, pp. 1788–1795 vol.2.

[9] A. Mourikis, S. Roumeliotis, and J. Burdick, "SC-KF mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements," *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 717–730, Aug 2007.

[10] S. Amirhosseini, M. Romanovas, T. Schwarze, M. Schwaab, M. Traechtler, and Y. Manoli, "Stochastic cloning unscented Kalman filtering for pedestrian localization applications," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, Oct 2013, pp. 1–10.

[11] M. Romanovas, V. Goridko, A. Al-Jawad, M. Schwaab, M. Traechtler, L. Klingbeil, and Y. Manoli, "A study on indoor pedestrian localization algorithms with foot-mounted sensors," in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, Nov 2012, pp. 1–10.

[12] A. Hatami and K. Pahlavan, "A comparative performance evaluation of RSS-based positioning algorithms used in WLAN networks," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, March 2005, pp. 2331–2337 Vol. 4.

[13] P. Groves, L. Wang, D. Walter, H. Martin, K. Voutsis, and Z. Jiang, "The four key challenges of advanced multisensor navigation and positioning," in *Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION*, May 2014, pp. 773–792.

[14] A. Ruiz, F. Granja, J. Prieto Honorato, and J. Rosas, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 1, pp. 178–189, Jan 2012.

[15] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *Proceedings of ION GPS*, vol. 2000, 2000, pp. 1735–1740.

[16] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[17] J. Smisek, M. Jancosek, and T. Pajdla, "3d with Kinect," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 1154–1160.

[18] L. Hutchings, "System and method for measuring movement of objects," Jun. 19 1997, wO Patent App. PCT/US1996/019,648. [Online]. Available: http://www.google.com/patents/WO1997021983A1?cl=en

[19] D. Chdid, R. Oueis, H. Khoury, D. Asmar, and I. Elhajj, "Inertial-vision sensor fusion for pedestrian localization," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, Dec 2011, pp. 1695–1701.

[20] C. Hide, T. More, C. Hill, and J. Pinchin, "Investigating the integration of a foot-mounted IMU and GNSS antenna," in *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012*, Oct 2012, pp. 1–10.

[21] W. Gao, J. Li, F. Yu, G. Zhou, C. Yu, and M. Lin, "Data fusion with two nonlinear constraints on Kalman filtering," in *Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION*, May 2014, pp. 524–528.

[22] I. Skog, J.-O. Nilsson, D. Zachariah, and P. Handel, "Fusing the information from two navigation systems using an upper bound on their maximum spatial separation," in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, Nov 2012, pp. 1–5.

[23] J. B. Bancroft and G. Lachapelle, "Estimating MEMS gyroscope g-sensitivity errors in foot mounted navigation," in *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012*, Oct 2012, pp. 1–6.

[24] W. Flenniken, "Modeling inertial measurement units and anlyzing the effect of their errors in navigation applications," Master's thesis, Auburn University, 2005.

[25] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[27] M. Angermann, P. Robertson, T. Kemptner, and M. Khider, "A high precision reference data set for pedestrian navigation using foot-mounted inertial sensors," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, Sept 2010, pp. 1–6.

[28] F. Zampella, A. Jimenez, F. Seco, J. Prieto, and J. Guevara, "Simulation of foot-mounted IMU signals for the evaluation of PDR algorithms," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, Sept 2011, pp. 1–7.

[29] J. Ginsberg, *Engineering Dynamics*. Cambridge University Press, 2008, vol. 10.

Appendix

A.1 Rotations and Coordinate Frames

The work presented in this thesis uses a North-East-Down (NED) convention. In order to track an object within this frame, a rotation must be made from the coordinate frame attached to the object (body frame) to the NED frame (navigation frame). The body frame aligns with the navigation frame at zero roll, pitch, and yaw. When the two frames align, the North, East, Down directions correspond with the body's X, Y, and Z, respectively. A rotation matrix is used to rotate vectors from the body frame to the navigation frame (C_b^n). This includes accelerations (in the case of the IMU) and delta positions (in the case of visual odometry) as shown in Equations (A.1) and (A.2).

$$a^n = C_b^n a^b \tag{A.1}$$

$$\Delta r^n = C_b^n \Delta r^b \tag{A.2}$$

To perform this rotation in three dimensions, a series of three individual rotations are made. These three rotations are performed in the specific order shown in Equation (A.3) to produce the 3D rotation matrix. The rotation matrix from the navigation frame to the body frame (C_n^b) is found simply by taking the inverse of the rotation matrix from the body frame to the navigation frame. Since these matrices are symmetric, the inverse of that matrix is the same as the transpose [29].

$$C_b^n = \left(C_n^b\right)^{-1} = \left(C_\phi C_\theta C_\psi\right)^{-1} \tag{A.3}$$

$$C_\psi = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

$$C_\theta = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{A.5}$$

$$C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.6}$$

Angular rate vectors are transformed from the body frame to the navigation frame using the mechanization matrix (M). These vectors include the rotation rates (in the case of the IMU) and delta rotations (in the case of visual odometry) as shown in Equations (A.7) and (A.8),

$$\omega^n = M\omega^b \tag{A.7}$$

$$\Delta\varphi^n = M\Delta\varphi^b \tag{A.8}$$

where the mechanization matrix is a function of the body's roll and pitch as shown in Equation (A.9) [25].

$$M = \frac{1}{\cos(\theta)} \begin{bmatrix} 1 & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\sin(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.9}$$

A.2 Least Squares Estimation

The least squares algorithm is used to fit a set of noisy observations with a model. In other words, a set of noisy measurements is used to determine the most likely value of a system's state. Along with a state estimate, the least squares solution provides the covariance associated with that estimate. For a nonlinear system, an iterative process must be performed, updating the estimate on each iteration. When information of the measurement accuracy is available, a weighting can be assigned to each individual measurement and accounted for in the weighted least squares approach [5].

Consider the linear equation

$$\delta z^- = H\delta x + \delta z^+ \tag{A.10}$$

where δz^- is the measurement innovation, defined as

$$\delta z^- = z - h\left(\hat{x}^-\right) \tag{A.11}$$

δz^- is the measurement residual, defined as

$$\delta z^+ = z - h\left(\hat{x}^+\right) \tag{A.12}$$

and δx is the state vector innovation, defined as

$$\delta x = \hat{x}^+ - \hat{x}^- \tag{A.13}$$

The matrix H is the measurement matrix, which is defined as

$$H = \frac{dh}{dx} = \frac{dz}{dx} \tag{A.14}$$

The least squares estimation operates by minimizing the measurement residuals with respect to the states. From this, the following relation is made:

$$\frac{\partial}{\partial \delta x}\left(\delta z^{+^T} \delta z^{+}\right) = 0 \tag{A.15}$$

After performing substitutions and additional matrix algebra and simplification, the result is the following equation:

$$\delta x = \left(H^T H\right)^{-1} H^T \delta z^{-} \tag{A.16}$$

By applying the above equation, an optimal estimate of the states is made given the measurements provided.

A.2.1 Iterated Least Squares

For the derivation above, it is assumed that the measurement matrix is completely independent of the states. In this case, the solution to Equation (A.16) provides and optimal estimate. In the case of a nonlinear problem, the measurement matrix will contain the state values themselves. In this case, *a priori* estimates of the state are used to approximate the measurement matrix. By performing least squares in an iterative fashion and setting $\hat{x}^-$ to $\hat{x}^+$ after each iteration, a state estimate can be achieved. The iteration process can be discontinued when the value of δx converges below some predefined threshold.

A.2.2 Weighted Least Squares

In many cases of state estimation, the measurements provided have varying accuracy. In this case, it is sensible provide a higher weighting to the more accurate measurements. Similar to the linear least squares problem, the residuals are to be minimized, but in this case a weighting matrix (W) is applied:

$$\frac{\partial}{\partial \delta x}\left(\delta z^{+^T} W \delta z^{+}\right) = 0 \tag{A.17}$$

After performing substitutions and additional matrix algebra and simplification, the result is the following equation:

$$\delta x = \left(H^T W H\right)^{-1} H^T W \delta z^{-} \tag{A.18}$$

For many applications, the accuracy of a measurement is provided as a covariance matrix (C_z). The covariance matrix is simply the inverse of the weighting matrix. With this knowledge, the inverse of the covariance matrix can replace the weighting matrix in the above equation.

A.2.3 Least Squares Residuals and their Covariance

Since the measurements used in least squares are not perfect, there will inevitably be residuals after the estimation process. The residuals, show in Equation (A.12), have an associated covariance. The covariance of the residuals is calculated using Equation (A.19).

$$C_{\delta z}^{+} = C_z - H\left(H^T C_z^{-1} H\right)^{-1} H^T \tag{A.19}$$

Equation (A.19) is essentially the covariance of the measurement minus the covariance of the state estimate mapped into the measurement domain. The covariance of the state estimate itself is calculated using Equation (A.20). This covariance sheds light to the accuracy of the estimation and also the correlation between state estimates.

$$C_{\hat{x}}^{+} = \left(H^T C_z^{-1} H\right)^{-1} \tag{A.20}$$

A.3 The Extended Kalman Filter

The extended Kalman filter is a Bayesian estimation scheme that is very popular for nonlinear systems. The standard EKF consists of three major steps: initialization, the time update, and the measurement update. Each process will be discussed in detail in this section.

A.3.1 Initialization

The state vector (x) consists of all the variables needed to describe a system throughout a dynamic process. For example, a navigation based Kalman filter tracks the position, velocity, and orientation of a body and these values make up the state vector. To start the filter process, an initial estimate must be made of the state vector. The initial estimate ($\hat{x}$) will differ from the actual initial state value, as shown in Equation (A.21). A successful EKF will minimize the difference between the state estimate and the true state, also referred to as the state estimate error (δx).

$$\delta x = x - \hat{x} \tag{A.21}$$

To initialize the EKF, the covariance (P) for this initial estimate must be attained. The state estimate covariance is defined using the expectation operator as shown in Equation (A.22).

$$P = E[\delta x \delta x^T] \tag{A.22}$$

The state estimate covariance is a matrix of size $n \times n$, where n is the number of states. The diagonals of the state estimate covariance matrix contain the variance of the state estimate for each individual state. The off-diagonals of P represent the correlation between different state estimates. Often times, the system is initialized using the first received measurement. In this case, the initial state estimate covariance matrix will be populated with the covariance of that measurement. However, it is rarely the case that a measurement is available for the

full state vector and so the remainder of the state estimate covariance must be populated in another way.

A.3.2 Time Update

The time update is also referred to as the prediction stage because it calculates a prediction of what the state will be at the next measurement epoch. This is made possible by modeling the system and propagating the states through the model given inputs (u) to the system. A discrete difference equation (f) is used to calculate the propagated state as a function of the previous state, the system input (u), and noise (w) as shown in Equation (A.23). The noise term represents the uncertainties that exist in the model or system input.

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{A.23}$$

The state estimate is propagated in the time update using the difference equation and setting the noise terms to zero as seen in Equation (A.24). The result is an *a priori* state estimate, which is denoted by the '$-$' superscript. The '$+$' superscript denotes the best estimate at t_{k-1} and is referred to as the *a posteriori* state estimate.

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0) \tag{A.24}$$

The EKF assumes perfect knowledge of the system inputs and for the disturbances (w) to be zero-mean Gaussian. The expectation operator is again used to express this assumption as shown in Equation (A.25). The magnitude of the noise is represented by its covariance (Q_k) as shown in Equation (A.26).

$$E[w_k] = 0 \tag{A.25}$$

$$E[w_k w_k^T] = Q_k \tag{A.26}$$

Also involved in the time update is the prediction of the state estimate covariance. The *a priori* state estimate covariance is calculated using the discrete Lyapunov equation as shown in Equation (A.27)

$$P_k^- = \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + Q_{k-1} \tag{A.27}$$

where Φ_{k-1} is the state transition matrix, which is calculated by taking the partial derivative of f with respect to x evaluated at the most recent state estimate as shown in Equation (A.28).

$$\Phi_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{(\hat{x}_{k-1}^+, u_k, 0)} \tag{A.28}$$

A.3.3 Measurement Update

Now that an *a priori* state estimate has been calculated, this value will be refined using the measurement update of the EKF. A measurement (y) is provided and a model for this measurement is available and defined by Equation (A.29). The measurement model (h) is a function of the state itself and measurement uncertainties (v).

$$y_k = h(x_k, v_k) \tag{A.29}$$

The same assumptions are made on the measurement noise to be zero-mean Gaussian as shown in Equation (A.30). The measurement update also requires a covariance (R) associated with the measurement, which is defined in Equation (A.31).

$$E[v_k] = 0 \tag{A.30}$$

$$E[v_k v_k^T] = R \tag{A.31}$$

The correction is made to the state estimate using the Kalman gain (K), which is calculated as shown in Equation (A.32)

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R \right)^{-1} \tag{A.32}$$

where H_k is the measurement matrix. The measurement matrix is calculated by taking the Jacobian of the measurement model with respect to the state evaluated at the current state estimate as shown in Equation (A.33).

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{(\hat{x}_k^-, 0)} \tag{A.33}$$

The Kalman gain is applied to the measurement innovation to correct the state estimate as shown in Equation (A.34)

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left(y_k - h \left(\hat{x}_k^-, 0 \right) \right) \tag{A.34}$$

and the state estimate covariance is updated accordingly as shown in Equation (A.35).

$$P_k^+ = \left(I - K_k H_k \right) P_k^- \tag{A.35}$$

The process is then repeated, resulting in a continued estimate of the state.