

Summer 9-11-2019

Tracking Center of Mass with Limited Inertial Measurement Units

Connor Nathaniel Morrow
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Biomechanics Commons](#), and the [Mechanical Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Morrow, Connor Nathaniel, "Tracking Center of Mass with Limited Inertial Measurement Units" (2019).
Dissertations and Theses. Paper 5273.
<https://doi.org/10.15760/etd.7146>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Tracking Center of Mass with Limited Inertial Measurement Units

by

Connor Nathaniel Morrow

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Mechanical Engineering

Thesis Committee:
Alexander Hunt, Chair
David Turcic
Robert Peterka

Portland State University
2019

Abstract

Wearable motion tracking systems pose an opportunity to study and correct human balance and posture during movement. Currently, these observations are either being conducted in laboratories with the use of camera systems and markers placed on the body, or through the use of suits containing large numbers (15-20) of inertial measurement units. However, to aid with rehabilitation of individuals with impaired balance, there needs to be an option to collect these observations outside of clinics and without incurring much cost from the user. I have focused on three inertial measurement units, one placed on each shank and one placed on the upper torso, to record posture and estimate where an individual's center of mass is located. The use of three inertial measurement units significantly reduces the cost and encumbrance for studying balance in an individual's primary environment. This reduction comes at the cost of accuracy, which is explored further through this thesis. First, I looked at different methods used for determining IMU orientation, such as a biologically inspired orientation algorithm and a gradient descent algorithm. I then incorporated those methods to create a model of the body that calculates an estimate on the center of mass, based on the user's weight and height. Finally, this estimate is compared to a force plate derived method of finding center of mass to determine viability of sensor reduction for practical applications. The created algorithm is able to generate a center of mass location that is able to match the actual location within 3 cm.

Acknowledgements

I would like to express my thanks to the various faculty members at Portland State University for providing an encouraging and thoughtful academic environment. Dr. Alexander Hunt welcomed me to be a part of the lab with open arms and instructed me on how to lay the foundation for good academic contributions. Dr. David Turcic provided phenomenal classes which increased my curiosity and excitement around control systems and signal processing. Outside of Portland State University, Dr. Robert Peterka was willing to provide a guiding hand for some of the more demanding algorithms I ran into during the course of this thesis.

I would be remiss to not also thank my lab mates, in particular Cody Sharzenberger and Andreas Natsis. Cody was a font of inspiration and would always lend an open ear to the various struggles I faced during the thesis. Andreas continually provided the necessary encouragement to make it through the toughest days. Finally, I would like to thank Reggie Stecher for helping me with different design challenges that came around, and also always being optimistic to help me in any way he could.

Contents

Abstract	i
Acknowledgements	ii
List of Tables	vi
List of Figures	viii
List of Abbreviations	ix
Chapter 1: Introduction	1
1.1 Detecting Falls	1
1.2 Kinematics of Falls	2
1.3 Current Methods Monitoring Center of Mass	2
1.4 Objectives and Outline	4
Chapter 2: Background	5
2.1 Inertial Measurement Units	5
2.2 Describing Rotations	6
2.2.1 Rotation Matrices	7
2.2.2 Rotation Quaternions	8
2.3 Sensor Fusion Algorithms	10
2.3.1 Complementary Filter	10

2.3.2	Vestibular Emulating Filter	12
2.3.3	Gradient Descent	14
2.4	IMU Sensor Placement	16
Chapter 3: Equipment		20
3.1	Inertial Measurement Units	20
3.2	Microcontroller	21
3.3	I2C Multiplexer	21
3.4	IMU Sensor Brackets	21
3.5	Project Box	22
3.6	Goniometer Linkage	23
3.7	Total Cost	24
3.8	Wiring	24
Chapter 4: Leg Angle Experiment		26
4.1	Methods	26
4.2	Results	27
4.3	Discussion	30
Chapter 5: Constrained Hip Experiment		31
5.1	Methods	31
5.2	Results	33
5.3	Discussion	35
Chapter 6: Center of Mass Experiment		37
6.1	Methods	37
6.2	Results	39
6.3	Discussion	42

Chapter 7: Conclusion	43
7.1 Future Work	44
Bibliography	47
Appendix	50
Discrete Vestibular Emulating Filter Algorithm	50

List of Tables

2.1	Inertial parameters of various limb segments for males, documented by Dumas et al. [3].	17
3.1	Cost of the various components used during the construction of the sensor suite	25
4.1	RMSE values calculated for each trial of the leg angle experiment. . .	29
5.1	Dimensions used to model the lower body during constrained hip experiment.	33
6.1	Body segment center of mass locations in the Earth frame.	39
6.2	RMSE values for the center of mass experiment.	41

List of Figures

1.1	Goniometer that is visually read to determine the angle between two body segments.	3
2.1	Block diagram depicting the complementary filter, with the value of gains used during this research.	11
2.2	Block diagram depicting the algorithm of the vestibular emulating filter, developed by Mergner et al.	14
2.3	Block diagram depicting the gradient descent algorithm, developed by Madgwick et al.	16
2.4	Placement of IMU sensors on the human body, including their sensor frames with respect to a global Earth frame.	19
3.1	Solidworks rendering of the IMU sensor bracket.	22
3.2	Image of the IMU bracket with an IMU attached and wired, as well as a velcro strap threaded through it.	22
3.3	Solidworks rendering of the project box.	23
3.4	Image of the project box, containing the microcontroller inside, with wiring heading out to the IMU sensors.	23
3.5	Image of the linkage with a goniometer and IMU attached.	24
3.6	Diagram of the microcontroller wiring.	25
4.1	Diagram showing the schematic for the knee bending experiment. . .	27

4.2	Leg angle measurements from extending and flexing the leg at a frequency of 0.165 Hz.	28
4.3	Leg angle measurements from extending and flexing the leg at a frequency of 0.467 Hz.	28
4.4	Leg angle measurements from extending and flexing the leg at a frequency of 1.00 Hz.	29
5.1	Diagram representing the model and equipment for the constrained hip experiment.	32
5.2	Normalized hip height location, captured visually as well as calculated using a complementary filter and vestibular emulating filter.	34
6.1	Z axis position of center of mass calculated during slow side to side swaying.	40
6.2	Z axis position of center of mass calculated during fast side to side swaying.	40
6.3	Z axis position of center of mass calculated during hip swaying.	41

List of Abbreviations

BoS - Base of Support

CoM - Center of Mass

CoP - Center of Pressure

DoF - Degree of Freedom

DPS - Degrees per Second

IMU - Inertial Measurement Unit

MEMS - Microelectromechanical Systems

MUX - Multiplexer

RMSE - Root Mean Squared Error

WHO - World Health Organization

Chapter 1

Introduction

1.1 Detecting Falls

Falls pose a significant risk to our aging populace. Currently, the World Health Organization estimates 646,000 individuals suffer from a fatal fall every year (<https://www.who.int/news-room/fact-sheets/detail/falls>), placing it as the second leading cause of unintentional injury death, after road traffic injuries. 41% of these falls are caused by incorrect weight shifting, or movement of the individual's center of mass [14]. These statistics show a clear need for a way to help prevent falls.

Individuals who are at high risk for falling are asked to engage in exercises and strength building to help maintain balance. Those with repeated occurrence of falls are asked to also make changes in their life that will eliminate high risk activities [17]. Balance and gait training is estimated to reduce the risk of falling anywhere from 14-27%. These measures help lower the probability of falling, however wont prevent falls as they are occurring.

Current systems using accelerometers are able to detect when a fall has happened and can alert medical personnel. Falls can be quantified as when the acceleration detected at the torso exceeds 1.8g [6]. Commercial devices that implement this technology, such as Life Alert, produce fast response to a falling event, however, don't prevent the fall from happening.

While we have methods of lowering the probability of falling and methods of detecting when a fall has happened, there are no current means of detecting motion that will lead to a fall event.

1.2 Kinematics of Falls

When evaluating how to prevent falls, it is important to quantify how falls occur and what conditions need to be fulfilled to maintain balance. For static upright standing conditions, the center of mass (CoM) needs to be within the base of support (BoS) or maintain a velocity that will move it towards the BoS [11]. The BoS is defined as the possible range of the center of pressure, and the center of pressure is the centroid of the pressure that is distributed by a person's contact with a surface [5]. Individuals are at risk of falling when their CoM is no longer aligned with their CoP. In order to quantify and move towards preventing falls, researchers need to have a way of monitoring where an individual's CoM is located.

1.3 Current Methods Monitoring Center of Mass

The golden standard for tracking an individual's center of mass is through a camera system. These visual systems typically are accompanied with the use of markers, which are placed along key anatomical landmarks on a human. Visual tracking systems, like VICON or Optotrack, can accurately track human motion to 1 mm of error [21]. Visual systems fail to monitor human motion properly when body segments overlap and markers cannot be detected. This prevents 3D models from being rendered.

Goniometers are another option for monitoring body orientation. A goniometer is a device that measures the change in angle that joints go through during mo-

tion. Goniometers are typically used to measure the range of motion for patients [20]. Electrogoniometers are goniometers that measure rotation through the use of a potentiometer. They are sufficient at detecting uniplanar movement, but lack the ability to properly monitor movement of joints with multiple degrees of freedom, making them less desirable for full body motion tracking. Electrogoniometers allow for automated data collection by a microcontroller. Other goniometers, such as the one in Figure 1.1, must be visually read in order collect data.

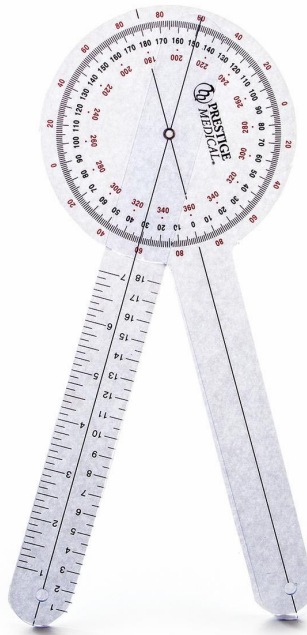


Figure 1.1: Goniometer that is visually read to determine the angle between two body segments.

Another method for monitoring human motion is through the use of full body suits that are embedded with inertial measurement units (IMU sensors). Xsens is a leading company that sells these suits. They are used for both full body tracking as well as animation capture. Their suits place 17 IMU sensors strategically on different limbs of an individual. Their proprietary sensor fusion algorithm and model construction monitors body motion with a high degree of accuracy [15]. The Xsens MVN Biomech

system was found to be able to monitor an individual's center of mass to within 5.45 mm [2]. However, with their cost being above \$10,000, they are impractical for individuals who don't require the degree of precision that Xsens provides.

1.4 Objectives and Outline

The primary objective of this research is to develop a low cost method for tracking an individual's center of mass. Inertial measurement units are a low cost device that are small and noninvasive. This makes them ideal for creating a method of monitoring human motion. The sensor system does not need to fully define an individual's range of motion, but rather, capture the important motions that contribute to a change in a person's center of mass. With a positive outcome, this research can continue to be refined to create a way of not only monitoring an individual's center of mass, but also alert the user if they are about to fall. With these objectives in mind, it is my hypothesis that center of mass can be tracked with three inertial measurement units, one placed on the upper torso, and one placed on each of the lower legs.

This thesis will begin with an overview of inertial measurement units, a discussion on how to quantify rotations, a description of the different sensor fusion algorithms that were used in this research, and an explanation of the sensor distribution across the body. Chapter 3 details what equipment was used for this research and why it was selected. Chapters 4 to 6 describe the different experiments that were performed to quantify the accuracy of the different sensor fusion algorithms that were used. The first two focus on modeling leg orientation of an individual, while testing which IMU sensor algorithm performs best. The last experiment tests the full IMU sensor system to detect where an individual's center of mass is located. Chapter 7 reflects on what was gained through this research and what future steps need to be taken to increase the viability of this sensor system.

Chapter 2

Background

2.1 Inertial Measurement Units

Inertial measurement units are sensors that detect linear accelerations and angular velocity. IMU sensors can also come equipped with a magnetometer, to measure magnetic field. Six degree of freedom (DoF) inertial measurement units include tri-axis accelerometers and gyroscopes. The accelerometer of the IMU sensor measures the sum of all accelerations and gravity, expressed in the sensor's reference frame. The gyroscopes measure rotational velocities about the axes of the reference frame. When IMU sensors include a magnetic angular rate and gravity sensors (MARG sensors), they are able to provide 9 DoF. By combining these measurements, IMU sensors provide an attitude and heading reference system (AHRS) which will describes the orientation of the IMU sensor. With the ability to generate an AHRS, IMU sensors have become an essential component for the navigation and guidance of land vehicles, aircrafts, and ships, both manned and autonomous.

IMU sensors have become much smaller in size over the past few years, making them ideal candidates for monitoring biomechanics of individuals [18]. The diminishing size is due to improvements to mecroelectromechanical systems (MEMS). MEMS technology has given them wider applicability than their previous iterations. Along with biomechanics, MEMS IMU sensors can be found in mobile phones, virtual and

augmented reality devices, game controllers, and many other small electronic systems.

The main complication that arises with using IMU sensors to determine orientation is the tendency to drift or deviate from their true angular value. The issue is due to gyroscopic drift. This drift is caused by continuously integrating the angular velocity data, where noise and bias in the signal continues to build up after repeated numerical integration (Equation 2.1). In order to combat this issue of drift, multiple sensor fusion algorithms have been created that increase the accuracy of determining the IMU sensor's orientation. Three of these algorithms will be discussed and tested in later sections.

$$\theta_{calculated} = (\omega_{measured} + \omega_{noise} + \omega_{bias}) \cdot \Delta t \quad (2.1)$$

2.2 Describing Rotations

There are multiple ways of describing the rotation of an IMU sensor. The first way is to collect the Euler angles that the IMU sensor experiences and to calculate rotation matrices from them. The other way is to describe the rotation in a four dimensional vector called a quaternion. Creating rotation matrices from Euler angles is the most common way for determining IMU sensor orientation. Euler angles are commonly taught and easy to generate based off of angular velocities. Rotation quaternions are more difficult to use, as it requires understanding specific quaternion mathematical operations, and have not seen much use in the past. However, with the advent of computer generated graphics, they have become increasingly more prominent since 1985 [16]. Rotation quaternions have made their way into IMU sensor research, primarily because they avoid the issue of gimbal lock. They are also more compact representations of rotation.

2.2.1 Rotation Matrices

Rotation matrices are orthogonal matrices, with real components, that are used to rotate vectors or rotate axes. Rotation matrices are calculated based off the Euler angles. The equation can be found in Equations 2.2 to 2.4, where θ , ϕ , and ψ are the Euler angles that represent rotation about the x axis, rotation about the y axis, and rotation about the z axis respectively.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (2.2)$$

$$R_y(\phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \quad (2.3)$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

To determine the overall rotation about all of the axes, the rotation matrices can be multiplied together.

$$R = R_z R_y R_x \quad (2.5)$$

An issue that arises with using Euler Angles to represent rotation is gimbal lock. Gimbal lock occurs when two axes are rotated into becoming parallel, losing a degree of freedom in the process, and creating a degenerate system. One way of avoiding this problem is to use quaternions to describe rotation instead.

2.2.2 Rotation Quaternions

Quaternions are vectors with four components written in complex number notation. Equation 2.6 shows the general form of a quaternion, where $\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = -1$ and $\hat{\mathbf{i}}\hat{\mathbf{j}} = \hat{\mathbf{k}}$, $\hat{\mathbf{j}}\hat{\mathbf{i}} = -\hat{\mathbf{k}}$, $\hat{\mathbf{k}}\hat{\mathbf{i}} = \hat{\mathbf{j}}$, $\hat{\mathbf{i}}\hat{\mathbf{k}} = -\hat{\mathbf{j}}$, $\hat{\mathbf{j}}\hat{\mathbf{k}} = \hat{\mathbf{i}}$, and $\hat{\mathbf{k}}\hat{\mathbf{j}} = -\hat{\mathbf{i}}$. In the following analysis, bold variables are representative of vectors.

$$\mathbf{q} = q_w + q_x\hat{\mathbf{i}} + q_y\hat{\mathbf{j}} + q_z\hat{\mathbf{k}} \quad (2.6)$$

Rotation quaternions can be used to describe the rotation of three component vectors. The three imaginary components of the rotation quaternion describe an axis that the vector rotates around, and the real component of the quaternion describe the magnitude of rotation. Knowing the angle of rotation, θ , and the unit vector describing the axis of rotation, $\mathbf{u} = u_x\hat{\mathbf{i}} + u_y\hat{\mathbf{j}} + u_z\hat{\mathbf{k}}$, a quaternion can be written as:

$$\mathbf{q} = \exp\left(\frac{\theta}{2}\mathbf{u}\right) = \cos\left(\frac{\theta}{2}\right) + (u_x\hat{\mathbf{i}} + u_y\hat{\mathbf{j}} + u_z\hat{\mathbf{k}})\sin\left(\frac{\theta}{2}\right) \quad (2.7)$$

To apply a rotation to a vector using a rotation quaternion, the vector is premultiplied by the quaternion and postmultiplied by the quaternion's reciprocal. Because quaternions are four element vectors, the vector to be rotated needs to be rewritten as having four elements, which means including a real component equal to 0 to the vector.

$$\mathbf{v} = 0 + v_x\hat{\mathbf{i}} + v_y\hat{\mathbf{j}} + v_z\hat{\mathbf{k}} \quad (2.8)$$

$$\mathbf{v}' = \mathbf{q}\mathbf{v}\mathbf{q}^{-1} \quad (2.9)$$

The multiplication used for this rotation is called the Hamilton or quaternion

product. The result is determined by multiplying each element through the distributive law. For example, for two generic quaternions $\mathbf{q}_1 = a_1 + b_1\hat{\mathbf{i}} + c_1\hat{\mathbf{j}} + d_1\hat{\mathbf{k}}$ and $\mathbf{q}_2 = a_2 + b_2\hat{\mathbf{i}} + c_2\hat{\mathbf{j}} + d_2\hat{\mathbf{k}}$, their Hamilton product, $\mathbf{q}_1\mathbf{q}_2$, would be equal to:

$$\begin{aligned}
& a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 + \\
& (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\hat{\mathbf{i}} \\
& (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)\hat{\mathbf{j}} \\
& (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)\hat{\mathbf{k}}
\end{aligned} \tag{2.10}$$

The reciprocal of a quaternion, seen in equation 2.9, is defined as[19]:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \tag{2.11}$$

The numerator in equation 2.11 is the conjugate of the quaternion.

$$\mathbf{q}^* = q_w - q_x\hat{\mathbf{i}} - q_y\hat{\mathbf{j}} - q_z\hat{\mathbf{k}} \tag{2.12}$$

The denominator in equation 2.11 is the norm of the quaternion.

$$\|\mathbf{q}\| = \sqrt{\mathbf{q}\mathbf{q}^*} \tag{2.13}$$

With a known angular velocity, $\boldsymbol{\omega}$, the time derivative of the quaternion can be calculated by taking the Hamilton product of the quaternion and the angular velocity:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q}\boldsymbol{\omega} \tag{2.14}$$

2.3 Sensor Fusion Algorithms

There are many sensor fusion techniques for calculating the angles of rotation or quaternions for an IMU sensor. The ones that were looked at during the course of this thesis were the complementary filter, a vestibular emulating filter created by Thomas Mergner et al. [9], and a gradient descent filter created by Madgwick et al. [7].

Other filters that were considered but not evaluated include the Kalman filter, Mahony filter, and other filters that are modifications on the filters described here. The Kalman filter is an accurate filter, but notorious for being too computationally intensive for most microcontrollers. The Mahony filter is a variation of the gradient descent filter, and it was decided to first implement the gradient descent filter to test it's viability before looking at variations.

2.3.1 Complementary Filter

One of the most common ways for filtering IMU data is the complementary filter [4]. It is straight forward to implement and comes at a low computational cost.

The algorithm is a numeric method for integrating gyroscope data. The angular velocity is multiplied by the sampling rate and added to the previously calculated angle. The filter then takes the tangent of the different accelerometer components to determine the angles of direction that the acceleration vector is in. Two gains are created, one for the gyroscope angle and one for the accelerometer angle. These gains are complementary, in that they sum to 1. After multiplying the different angles by their respective gains, the results are summed together to provide an estimated angle of rotation.

$$\theta_x(t) = Gain_{gyroscope} \cdot (\theta_x(t-1) + \omega_x(t) \cdot t_s) + Gain_{accelerometer} \cdot \tan\left(\frac{a_y(t)}{a_z(t)}\right) \quad (2.15)$$

$$\theta_y(t) = Gain_{gyroscope} \cdot (\theta_y(t-1) + \omega_y(t) \cdot t_s) + Gain_{accelerometer} \cdot \tan\left(\frac{a_x(t)}{a_z(t)}\right) \quad (2.16)$$

$$\theta_z(t) = Gain_{gyroscope} \cdot (\theta_z(t-1) + \omega_z(t) \cdot t_s) + Gain_{accelerometer} \cdot \tan\left(\frac{a_x(t)}{a_y(t)}\right) \quad (2.17)$$

To serve as a high pass filter, the gyroscope gain is kept to a value close to one. The accelerometer gain is closer to 0, to act as a simple low pass filter. When taking the tangent of the accelerometer, the aim is to take the tangent of only the measured gravitational vector, and to eliminate any higher frequency movements that are detected by the IMU sensor. Figure 2.1 shows this algorithm as a block diagram.

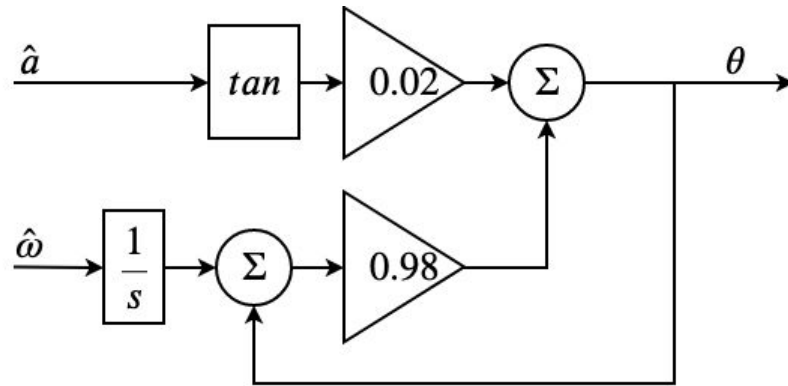


Figure 2.1: Block diagram depicting the complementary filter, with the value of gains used during this research.

2.3.2 Vestibular Emulating Filter

The vestibular emulating filter is a filter that was developed by Mergner et al [9, 10]. The human vestibular system uses biological analogs to IMU sensors for balance control. Inner ear otolith receptors detect inertial acceleration and the semicircular canals detect angular velocities in humans [1]. These organic sensors are similar to linear accelerometers and gyroscopes respectively. The vestibular system synthesizes information received from the otoliths and canals to determine an estimated angular velocity, linear acceleration, and gravitational direction. Transfer functions have been created to represent each stage of this filter.

The semicircular canals interpret angular velocity information with a time constant of 5 seconds [8]. In order to approximate how humans receive this information, gyroscope data is passed through a first order high pass filter with a time constant of 5 seconds.

$$\text{HP} = \frac{5s}{5s + 1} \quad (2.18)$$

The acceleration data is normalized and differentiated with respect to time. The normalized acceleration and its time derivative are crossed to find the rate of change in direction.

$$\mathbf{f} = \dot{\mathbf{a}} \times \frac{\mathbf{a}}{|\mathbf{a}|} \quad (2.19)$$

This signal passes through a first order low pass filter. This filter is complementary with the high pass filter used for the angular velocity data.

$$1 = \text{HP} + \text{LP} \quad (2.20)$$

$$\text{LP} = \frac{1}{5s + 1} \quad (2.21)$$

The outputs from these two filters are then summed together to give an improved estimate of the angular velocity, $\hat{\omega}$, that is not affected by gyroscopic bias. This angular velocity estimate is then used to update the direction of gravity. The previous direction of gravity is crossed with angular velocity to find the rate of change in direction of gravity, \dot{g} .

$$\dot{\mathbf{g}}_t = \mathbf{g}_{t-1} \times \hat{\boldsymbol{\omega}} \quad (2.22)$$

To find the new direction of gravity, the rate of change in the direction of gravity is passed through a leaky integrator, with a time constant of 20 seconds. To fill in for a portion of the output that is lost through the leaky integrator, some of the raw acceleration data is added to the input. The accelerometer data is multiplied by a gain of 0.05 before being added to $\dot{\mathbf{g}}$. This new signal then goes through the leaky integrator.

$$\text{Leaky Integrator} = \frac{20}{20s + 1} \quad (2.23)$$

The updated direction of gravity is subtracted from the raw acceleration data to eliminate the gravitational component that it measures. Figure 2.2 displays a block diagram representation of the algorithm.

The orientation of the sensor relative to the earth frame is calculated using the angles between $\hat{\mathbf{g}}$ and the original direction of gravity, determined at the beginning of the algorithm. These angles are then used to create a rotation matrix that determines the rotation of the sensor.

This process is described as a continuous system. In order to make it available to

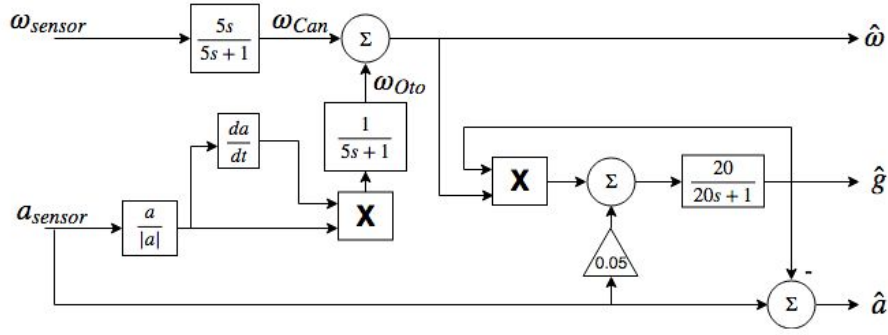


Figure 2.2: Block diagram depicting the algorithm of the vestibular emulating filter, developed by Mergner et al.

be placed onto a microcontroller, the system was converted into a discrete system of difference equations using MATLAB. The MATLAB code can be found in Appendix A1.

2.3.3 Gradient Descent

The gradient descent method for fusing IMU sensor data was developed by Madgwick et al. as a method of reducing the computational load of determining sensor orientation compared to the popular Kalman filter [7].

The gradient descent filter is a first order optimization algorithm that calculates a rotation vector that aligns a reference acceleration vector, in this case a gravity vector, with the IMU sensor measured acceleration vector. It first defines the optimization function as:

$$\mathbf{f} = \mathbf{q}_{Es} \mathbf{d}_E \mathbf{q}_{Es}^{-1} - \mathbf{a}_s \quad (2.24)$$

\mathbf{q}_{Es} describes the orientation of the sensor frame in the Earth frame, \mathbf{d}_E is the reference vector, and \mathbf{a}_s is the acceleration experienced in the sensor frame. The function is minimized when \mathbf{q}_{Es} rotates the reference vector to be in the same direction

as the measured acceleration vector.

The first step of the algorithm is to evaluate the objective function with the previously estimated \mathbf{q}_{Es} and multiplying that by the Jacobian of the objective function.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial}{\partial w} \mathbf{f} \\ \frac{\partial}{\partial x} \mathbf{f} \\ \frac{\partial}{\partial y} \mathbf{f} \\ \frac{\partial}{\partial z} \mathbf{f} \end{bmatrix} \quad (2.25)$$

$$\nabla \mathbf{f} = \mathbf{J}^T \mathbf{f} \quad (2.26)$$

The filter also takes the measured angular velocity, and performs a quaternion product with it and the previously estimated rotation quaternion to come up with the time derivative of the quaternion.

$$\dot{\mathbf{q}}_{Es} = \frac{1}{2} \mathbf{q}_{Es} \boldsymbol{\omega} \quad (2.27)$$

The gradient is then normalized and multiplied by a scaling factor, β . For this research, the value of β was set to 0.1. This value is then subtracted from the derivative. The scaling factor and gradient serve to reduce the error in the quaternion derivative that comes up due to gyroscopic bias and noise. The result is numerically integrated and normalized to form the new estimate of the rotation quaternion. The algorithm is visualized in Figure 2.3.

The gradient descent algorithm used during this thesis was written in Arduino and MATLAB by Sebastian Madgwick, and can be found at <https://github.com/arduino-libraries/MadgwickAHRS>. Minor changes were made to extract different information from the Madgwick filter class.

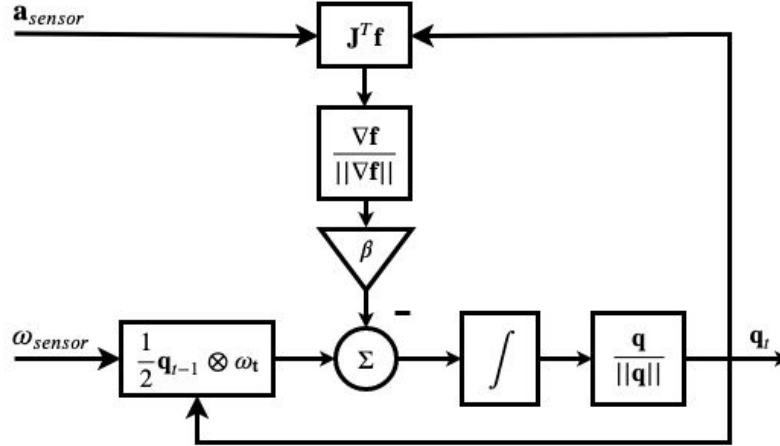


Figure 2.3: Block diagram depicting the gradient descent algorithm, developed by Madgwick et al.

2.4 IMU Sensor Placement

One of the goals of this thesis is to develop an IMU sensor system that minimizes the number of sensors needed. By lowering the amount of sensors needed, the cost of the system is lowered and provides instrumentation that won't burden or interfere with an individual's normal tasks.

In order to decide the number of sensors required and where to place them, it was necessary to look into which limbs contributed to the majority of an individual's center of mass. This was done by analyzing the results from Dumas et al.'s paper on human body segment inertial parameters [3]. This paper lists a number of body segments, such as the head, torso, legs, and describes where the center of mass is for each of those segments, as well the average percentage of mass of a person that is contained within those segments.

The body segment center of mass locations are given as a scaling value in the x, y, and z axis. The location can be found by multiplying these values by the length of the body segment. For example, if the torso has the following parameters:

$$\text{Torso Length} = 47.7 \text{ cm} \quad (2.28)$$

$$\text{Torso CoM Scale} = [-0.036, -0.42, 0.002]^T \quad (2.29)$$

then the center of mass location of the torso, in reference to the torso anatomical landmark, the suprasternale, is:

$$\text{Torso CoM} = [-1.717, -20.034, 0.0954] \text{ cm} \quad (2.30)$$

The segments that contribute most to the mass of an individual are the torso, pelvis, thigh's, and lower legs. The percentages can be found in Table 2.1.

Table 2.1: Inertial parameters of various limb segments for males, documented by Dumas et al. [3].

Limb Segment	Percent of Mass
Head and Neck	6.7%
Torso	30.3%
Pelvis	14.2%
Thigh	12.3%
Lower Leg	4.8%
Feet	1.2%
Arm	2.4%
Forearm	1.7%
Hand	0.6%

Using this information, by knowing the orientation of each limb segment, the overall center of mass coordinates, \mathbf{C}_{total} , can be found by summing all of the limb center of mass locations, $\mathbf{C}_{segment}$, multiplied by their percent of mass contribution, $\%_{mass}$.

$$\mathbf{C}_{total} = \Sigma(\mathbf{C}_{segment} \cdot \%_{mass}) \quad (2.31)$$

Because of their large contributions to an individual's center of mass, it was decided to focus on using three inertial measurement units, one on the torso and one on each lower leg. By knowing the position of these segments in space, the thigh, pelvis, and head position can be calculated by making assumption about their orientation relative to the segments that are being measured. The assumptions made were that the head is extended along the same orientation as the torso, the pelvis is at the base of the torso orientation, the feet are located at the bottom of the lower legs, and the thighs orientation begins at the end of the lower legs and connect to the pelvis. The arms, forearms, and hands are the only segments not included in this analysis. Combined, they contribute 9.4% to the overall mass of an individual. With this value being low, it was assumed their center of mass and position could be neglected, as the other body segments would contribute to the location of the center of mass more significantly. The accuracy of these assumptions was tested during the last experiment of this thesis. Figure 2.4 shows the location of each IMU sensor used to find the orientation of the targeted body segments, as well as showing how their sensor frames are oriented in the global Earth frame.

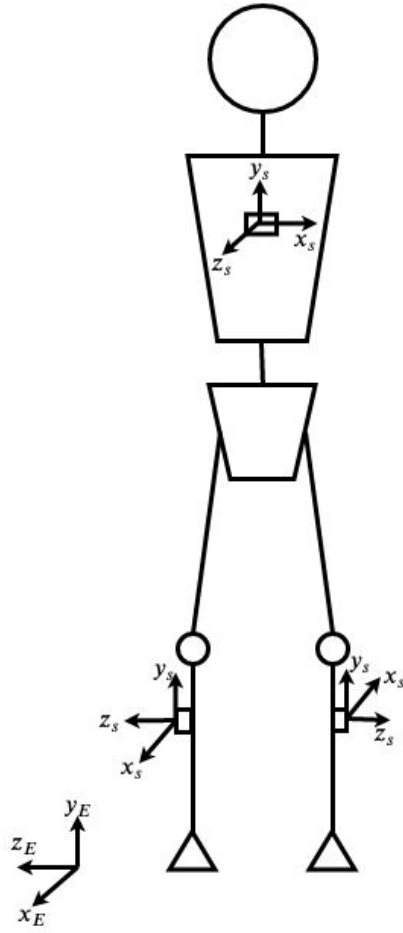


Figure 2.4: Placement of IMU sensors on the human body, including their sensor frames with respect to a global Earth frame.

Chapter 3

Equipment

This chapter describes the various pieces of equipment that were used. Sections 3.1 to 3.6 describe the equipment and its purpose. Section 3.7 describes the overall cost of the project, to see how well it meets the objective to keep costs minimal. Section 3.8 shows a wiring schematic that depicts how the electrical equipment fits together.

3.1 Inertial Measurement Units

The IMU sensors used for this research were the Adafruit NXP Precision 9 DoF IMU. They combine a FXAS21002C 3 axis gyroscope and a FXOS8700CQ 6 axis accelerometer and magnetometer. The magnetometer was not used during this research.

The gyroscope has a 16-bit digital resolution. It has a configurable angular velocity range. The range goes from ± 250 degrees per second (dps) to ± 2000 dps. The resolution at ± 250 dps is 0.39 dps and the resolution at ± 2000 dps is 3.125 dps. For this research, the range was set to ± 250 dps. The gyroscope also has a data output frequency between 12.5 and 800 Hz.

The accelerometer has a 14-bit ADC resolution. The adjustable acceleration range goes from $\pm 2g$ to $\pm 8g$. For this research, the range was set to $\pm 2g$. The accelerometer has a data output frequency between 1.56 to 800 Hz.

These sensors are low cost and able to provide readings accurate for testing this

research. Other IMU sensors could have been used, such as ones from Xsens and APDM, however, the increase in accuracy is offset by the dramatic increase in cost.

3.2 Microcontroller

The microcontroller used for this experiment was the Adafruit Feather M0 Adalogger. The Adalogger used an ATSAMd21g18 ARM Cortex M0 processor, with a clock speed of 48 MHz and 32KB of RAM. It has a built in MicroSD card slot for writing data to, which was used during experimentation to collect raw and filtered IMU sensor values. It also used an external lithium polymer battery to power it.

3.3 I2C Multiplexer

For the experimental setup, three of the same IMU sensors were used. They all shared the same I2C (Inter-Integrated Circuit) protocol address. Each IMU has two addresses, however, three would be necessary to communicate with them independently. In order to solve this issue, an I2C multiplexer (MUX) was used. An I2C multiplexer is a multiplexer that splits the I2C signal between multiple devices. It only communicates with one at a time, and is told which one to communicate with by the microcontroller. The I2C multiplexer used was an Adafruit TCA9548A.

3.4 IMU Sensor Brackets

Custom built brackets to hold the IMU sensors were 3D printed using a Markforged printer. The 3D printer uses a proprietary material called Onyx, which is primarily composed of nylon plastic and chopped strands of carbon fiber. The IMU sensor brackets were designed to match the shape of the IMU sensors. They include a loop in which to thread a velcro strap. This velcro strap attaches the IMU and bracket to

the individual's body segments. Figure 3.1 shows the SolidWorks model used for 3D printing and Figure 3.2 shows the final assembly for the IMU bracket.

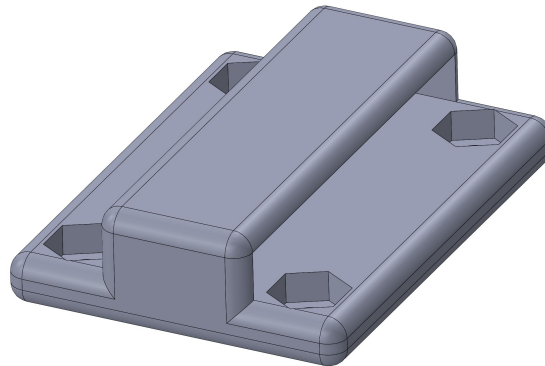


Figure 3.1: Solidworks rendering of the IMU sensor bracket.

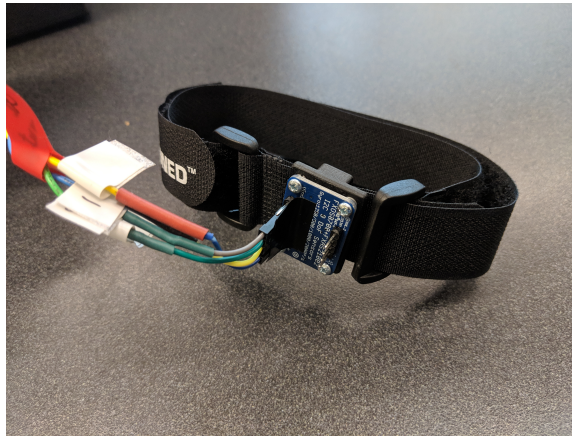


Figure 3.2: Image of the IMU bracket with an IMU attached and wired, as well as a velcro strap threaded through it.

3.5 Project Box

A custom built project box was 3D printed using a Markforged printer. The box housed all the microcontroller components, I2C multiplexer, and was embedded with LED's to indicate when experiment runs were in progress or completed.

The box has four through holes located along the bottom surface. The holes included recess in which to place a 2-56 nut. A 2-56 screw was threaded through the

microcontroller shield, to secure it and the components to the bottom of the project box.

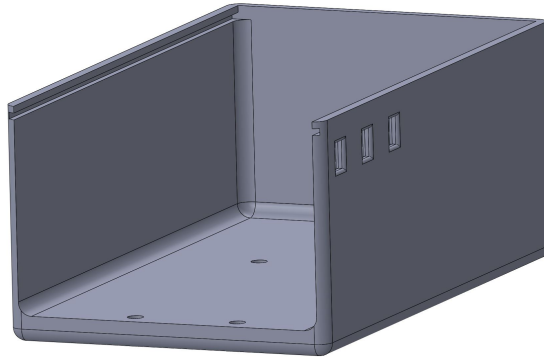


Figure 3.3: Solidworks rendering of the project box.



Figure 3.4: Image of the project box, containing the microcontroller inside, with wiring heading out to the IMU sensors.

3.6 Goniometer Linkage

The goniometer is constructed with a potentiometer. The potentiometer rotates with one bar of the linkage, sending a varying analog signal to the microcontroller as the linkage rotates. After calibration, a relationship between the voltage signal and the angle was calculated. The potentiometer used for this experiment was a TT

Electronics P160KN28K. An IMU sensor is also attached to the linkage that sits on the lower leg. The IMU sensor is located 6 cm below the potentiometer. This piece of equipment was used for the leg extension experiment (Chapter 4).

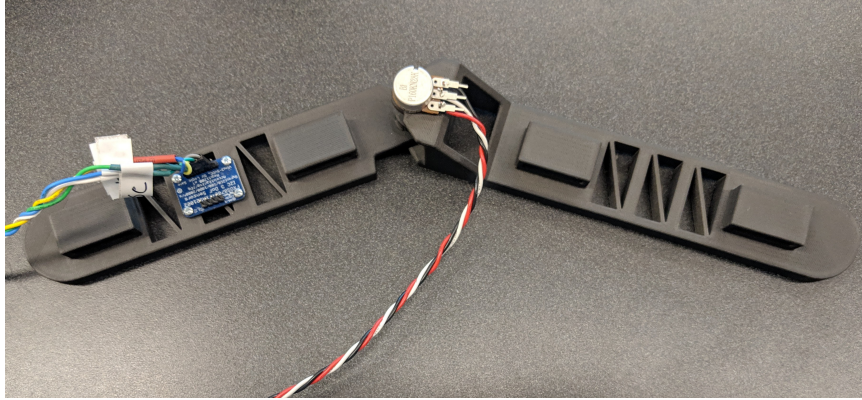


Figure 3.5: Image of the linkage with a goniometer and IMU attached.

3.7 Total Cost

The individual component cost and sum total for the full IMU sensor system is found in Table 3.1. The total cost for the resulting system is \$126.07. This price doesn't factor the engineer time developing the coding system, however it does provide the basis of a low cost device that can be used to monitor balance and center of mass for individuals. With the price of Xsens systems being around \$10,000, the goal of creating a system with substantially lower cost than the leading market devices has been achieved.

3.8 Wiring

The microcontroller was connected with the three IMU sensors through the I2C multiplexer. Communication across these devices occur along the SDA and SCL lines. The microcontroller was also connected to three LED's. Green indicated startup of

Table 3.1: Cost of the various components used during the construction of the sensor suite

Component	Quantity	Cost
NXP Precision 9 DoF Breakout	3	\$44.85
Adafruit Feather M0 Adalogger	1	\$19.95
Assembled Terminal Block	1	\$14.95
Lithium Ion Polymer Battery	1	\$5.95
16 GB micro SD Card	1	\$9.99
I2C Multiplexer	1	\$6.95
3D Printed Project Box	1	\$12.70
3D Printed IMU Brackets	3	\$1.71
Velcro Straps	8	\$8.99
Total	1	\$126.07

the microcontroller and when the data collection was complete, yellow indicated data was currently being captured, and red indicated an error with the system. Figure 3.6 shows how these pieces were connected.

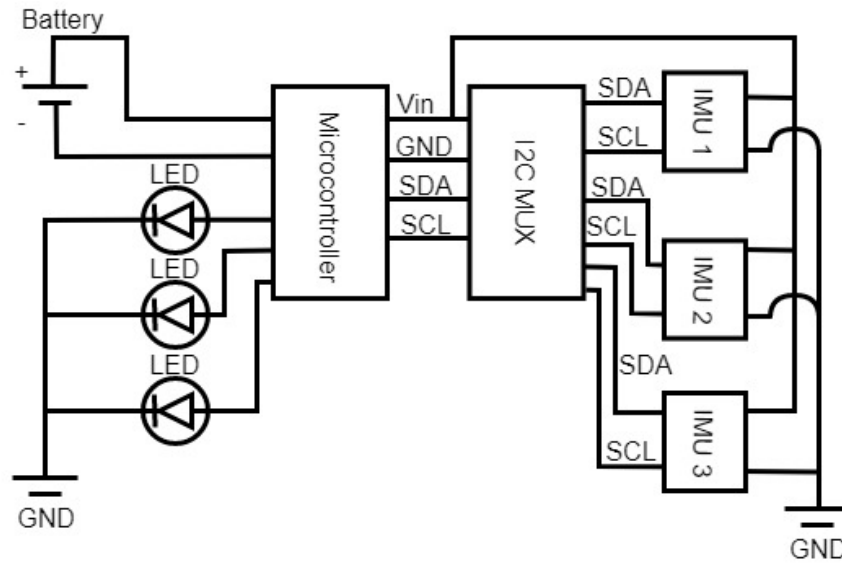


Figure 3.6: Diagram of the microcontroller wiring.

Chapter 4

Leg Angle Experiment

The leg angle experiment was the first experiment developed to observe how well the IMU sensor fusion algorithms would perform at determining limb orientation. The experiment compares goniometer data to data filtered through the complementary filter, gradient descent filter, and vestibular emulating filter (discussed in Section 2.3).

4.1 Methods

In this experiment, the subject attached the goniometer linkage and IMU holder onto their leg. Then, while sitting down with their leg bent to 90° , they extended their leg out horizontally and returned it to the original position. The participant performed this motion repeatedly within 30 seconds. The experiment was conducted three times, with varying frequencies of leg extension and flexion.

The IMU sensor captured angular velocity and linear acceleration data, which was post processed through the complementary filter and vestibular emulating filter using MATLAB. The gradient descent angles were calculated on the microcontroller and stored on the SD card to compare with the complementary filter.

The goniometer provided the validation reference measurement for this experiment. It was held in place, both with velcro straps and by holding it with the

participants hands. It provides an accurate measurement of the true angle that the leg swings through. Figure 4.1 shows a representation of the experimental set up.

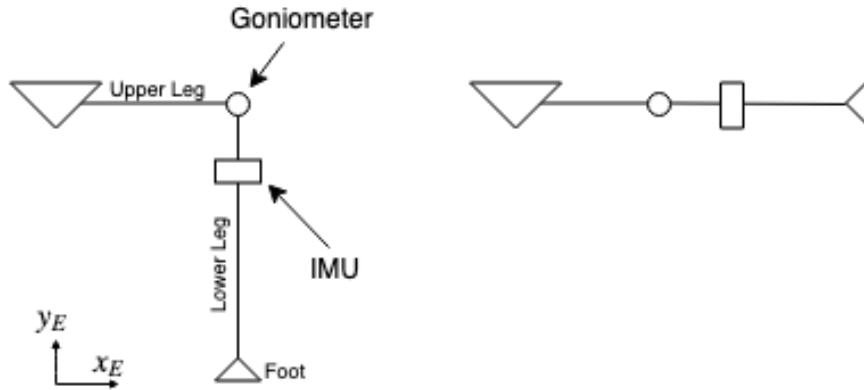


Figure 4.1: Diagram showing the schematic for the knee bending experiment.

4.2 Results

This experiment was performed three times. The first time it was performed, the leg extended and flexed at a frequency of 0.165 Hz. The results of this experiment can be seen in Figure 4.2. Both the complementary filter and the gradient descent filter were able to match the true angle generated by the goniometer. There is a period within the first two seconds that the gradient descent filter has dramatic error, as it is still optimizing to find the quaternion that matches the initial orientation of the inertial measurement unit in the Earth frame. The vestibular emulating filter shows errors early on. It has significantly diminished peaks compared to the actual angle the leg swings through, and is off phase by 1.01 seconds.

The second time the experiment was performed, the frequency was increased to 0.467 Hz. The complementary filter was unable to match the peak to peak range of the goniometer at this frequency, but was able to match it closer compared to the vestibular emulating filter. The gradient descent filter was still able to follow the angle

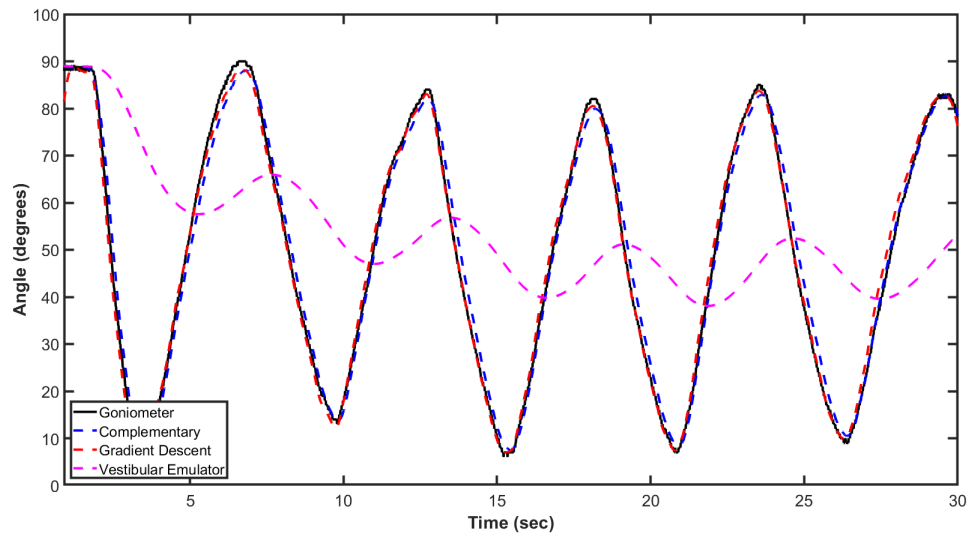


Figure 4.2: Leg angle measurements from extending and flexing the leg at a frequency of 0.165 Hz.

almost perfectly. The vestibular emulating filter continues to be unable to match the angle change, with a phase lag of 0.54 seconds. Figure 4.3 shows the results of this trial.

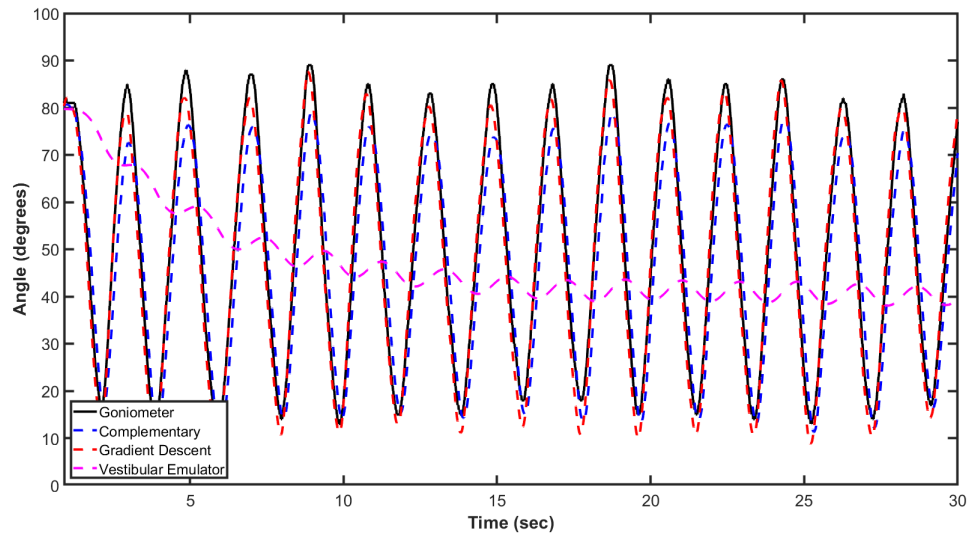


Figure 4.3: Leg angle measurements from extending and flexing the leg at a frequency of 0.467 Hz.

The last experiment was performed at 1Hz. This is where we saw dramatic drop off in the accuracy of the complementary filter. The gradient descent filter was able to easily match the true angle. The vestibular emulating filter shows a phase lag of 0.46 seconds for this experiment. Figure 4.4 shows the results of this trial.

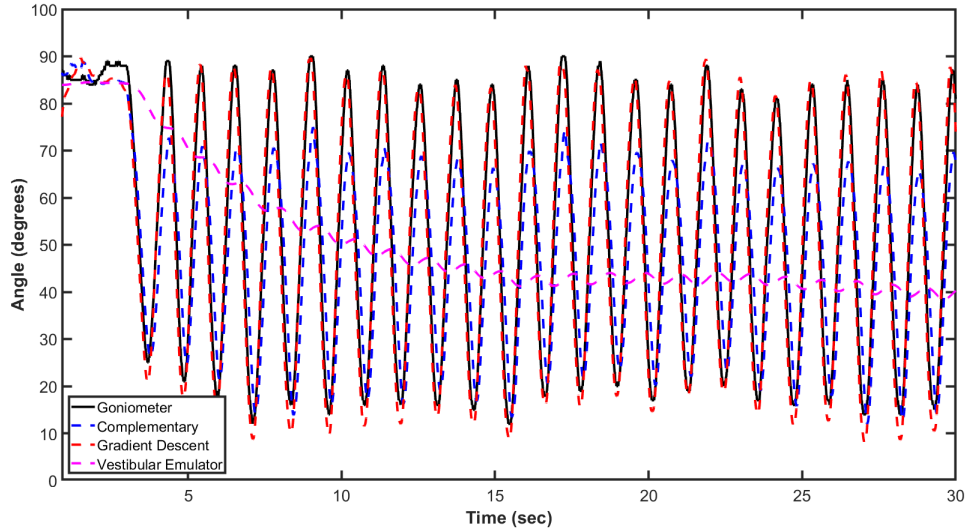


Figure 4.4: Leg angle measurements from extending and flexing the leg at a frequency of 1.00 Hz.

To evaluate the accuracy quantitatively, the root mean squared error (RMSE) of all three experiments was calculated. The equation used to calculate the RMSE can be found in equation 4.1. Table 4.1 lists the RMSE values calculated for each trial.

$$\text{RMSE} = \sqrt{\frac{1}{N} \left(\sum_i^N \theta_{i,goniometer} - \theta_{i,algorithm} \right)^2} \quad (4.1)$$

Table 4.1: RMSE values calculated for each trial of the leg angle experiment.

Frequency	Complementary Filter	Gradient Descent	Vestibular Filter
0.165 Hz	2.76°	1.72°	26.90°
0.467 Hz	7.54°	3.97°	25.36°
1.00 Hz	11.85°	5.00°	25.16°

4.3 Discussion

This experiment was able to identify how the changing dynamics of the human system would effect the results captured from the complementary filter, gradient descent filter, and vestibular emulating filter. We see that the complementary filter begins to experience drop off of its peak to peak range at higher frequencies. It maintains the same phase as the leg swing angle, but it has a low gain that continues to decrease as frequency increases. The gradient descent method doesn't experience this at high frequencies and is able to maintain both phase and magnitude with the goniometer angle.

Overall, results show that the gradient descent method is more accurate and robust than the complementary filter for dynamic motions. However, for ease of computational load, the complementary filter provides accurate measurements during low frequency activities.

The vestibular emulating filter shows significant error throughout all of the experiments. The results from Mergner et al. experiments were accurate for a change in angle with a peak to peak range of up to 8 degrees [10]. However, with a peak to peak range of 90 degrees, the vestibular emulating filter has issues maintaining accurate results for these frequencies. The filter will continue to be investigated, however, with this degree of error, it shows that it will likely not be robust enough to capture a person's posture during normal movement.

Chapter 5

Constrained Hip Experiment

The constrained hip experiment was designed to test how well the complementary filter, gradient descent filter, and the vestibular emulating filter would perform at modeling both lower legs. The hip was constrained in the horizontal plane, but allowed to move up and down vertically. This serves as a way to validate the sensor fusion algorithm, by recording the hip position with a stationary camera. This experiment was designed to build off the previous experiment. Instead of only evaluating one leg's position, this experiment calculates both leg positions. Results were meant to inform how to construct the final experiment, in which the full body orientation was calculated as well as calculating the center of mass.

After the first experiment, it was seen that the complementary filter performed very well during low frequency movements. Because of this and its ease of use, it was used as a baseline to compare it to the vestibular emulating filter.

5.1 Methods

The experiment was designed to see how accurate the IMU sensor algorithms would perform at monitoring lower leg position and angle. The experiment used the complementary filter, gradient descent filter, and vestibular emulating filter to track the shank of the participant while they enter a crouching position. With the shank angle

determined, the thigh position is calculated by connecting the knee position (located at the end of the shank) to the transverse location of the hip, which is constrained by the experimental apparatus. The angle of the thighs were then used to calculate an experimentally determined hip height.

One IMU sensor was attached to the outside of each shank, 8cm below the knee on the leg of the participant. The hip was attached to a measurement device, which was comprised of a platform with a measuring stick extending vertically upward off the platform with centimeter gradations along it. The gradations indicate the hip height above the platform. The hip attachment point is at approximately the location of the L5 spinal segment. The hip was allowed to slide up and down the measurement device, and using a video of the experiment, the hip height was determined manually at every second that the experiment was performed. This hip height measurement was used to validate the accuracy of the sensor fusion techniques. Figure 5.1 represents the experimental set up.

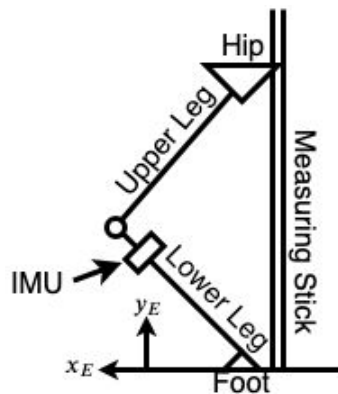


Figure 5.1: Diagram representing the model and equipment for the constrained hip experiment.

At the beginning of the experiment, the subject stood in an upright position, feet placed shoulder width apart. The camera began recording video of the experiment. Without moving their feet, the subject slowly bent their knees to enter a crouching

position. The motion of the lower legs was recorded through the IMU sensors and the position of their hip was recorded with the video. The experiment lasted for 30 seconds, during which the subjected moved into the crouching position and then returned to their initial upright position.

After completing the data collection, the sensor data was filtered through a complementary filter, gradient descent filter, and vestibular emulating filter. These filters calculated the angle of the lower legs relative to an Earth frame. These angles determine the experimental hip height location, based on the lengths and positions of the individual’s lower and upper legs and their orientation. This experimental result was compared to the data collected from the video. The video gave a direct location for the hip height, with measurements occurring once a second with a resolution of half a centimeter. The video and the collected IMU sensor data were synced by a green LED embedded in the project box. When data collection began, the LED illuminated and could be seen in the video recording. Table 5.1 lists the various model lengths that were used during the recreation of the lower limbs in MATLAB.

Table 5.1: Dimensions used to model the lower body during constrained hip experiment.

Segment	Length
Height from Floor to Ankle	5.0 cm
Shank Length	43.3 cm
Thigh Length	42.0 cm
Height from Hip Joint to L5	4.2 cm
Initial L5 Height	94.5 cm
Feet Width	21.0 cm
IMU Sensor Placement (below knee)	8.0 cm

5.2 Results

The results of this experiment are shown in Figure 5.2. The hip height data was normalized to the standing hip height. Results show that the complementary filter

and gradient descent filter were able to calculate a hip height that was close to the camera recorded hip height location. The vestibular emulating filter is offset from the true position, with phase lag of about 2 seconds.

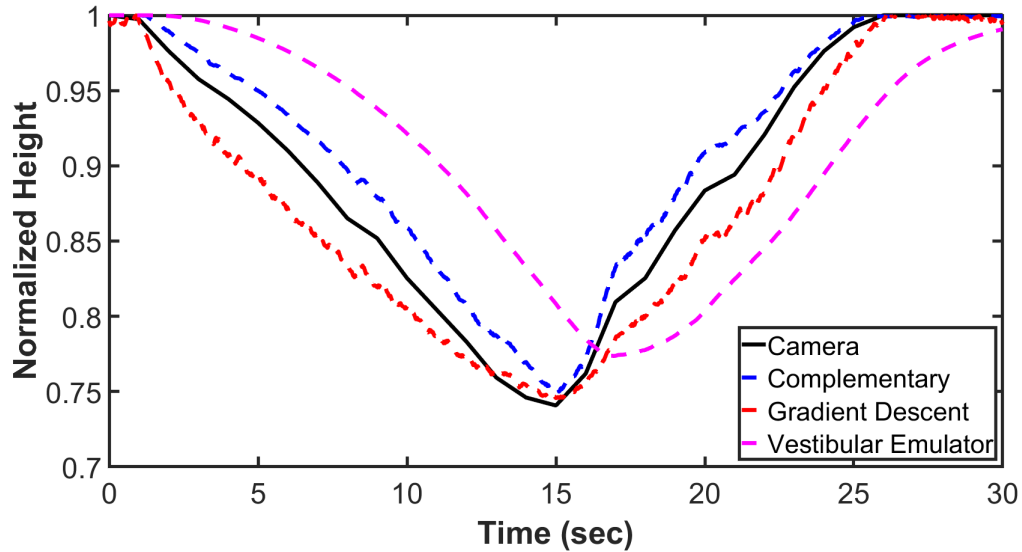


Figure 5.2: Normalized hip height location, captured visually as well as calculated using a complementary filter and vestibular emulating filter.

To quantify the the precision of these methods, the RMSE was calculated between the camera validation and each algorithm. The RMSE equation can be found in Section 4.1. The RMSE of the complementary filter was 0.0196, the RMSE of the gradient descent filter was 0.0198, and the RMSE of the vestibular emulating filter was 0.0663. These values were calculated based on the normalized hip height. By multiplying them by the initial hip height, they provide specific information about the error of the system. The complementary filter had a RMSE of 1.85 cm, the gradient descent filter had a RMSE of 1.87 cm, and the vestibular emulating filter had an RMSE of 6.26 cm.

5.3 Discussion

The results demonstrate how a limited number of IMU sensors can be used to determine hip height during motion. The complementary filter and gradient descent filter are more accurate than the vestibular algorithm for tracking the IMU sensor orientation. The error found with the complementary filter and gradient descent filter were below 2 cm, while the vestibular emulating filter was over three times that amount.

While not shown, gradient descent filter in this experiment experienced a longer period of convergence than it did with the first experiment performed. Without initialization, the convergence period was about 7 seconds. This is due to including a second IMU sensor in the algorithm. With the need to converge two optimization functions, it takes longer to arrive at a solution. This continues to come up in the next experiment where 3 IMU sensors are used.

The vestibular algorithm does not perform as well in the context of continuously tracking an IMU sensor in real time. The issue comes from the very large angle changes and slow dynamics of the system. The primary role of this algorithm is to emulate the vestibular system of a human, which is only one component that contributes to overall balance. The hope was that this algorithm would be an accurate way of monitoring limb position in a biologically inspired manner. The outcome here shows that when emulating biology for the stance determination of the limbs, proprioception should be the system to replicate.

The experimental set up itself could lend to the errors found in these results. The hip location was determined manually by watching the video of the experiment. During the experiment, the measuring rod tilted slightly due to the force applied when the hip pulled against it. While the change in angle was less than ± 2.5 degrees, it still could provide inaccurate readings of the hip location.

The complementary filter and gradient descent filter continue to show the highest accuracy results. Moving forward with this research, they will be the filters considered.

Chapter 6

Center of Mass Experiment

This was the final experiment conducted to determine the viability of using three IMU sensors to track an individual's center of mass. The data collected from the IMU sensors was validated against a one dimensional axis position of the center of mass, retrieved from a device created by Dr. R. Peterka [12]. The purpose was to test the validity of using three IMU sensors to track CoM of a person using the complementary filter and the gradient descent filter.

6.1 Methods

For this experiment, the participant wore the three IMU sensor suite proposed for the measurement of CoM sway. The systems consists of an IMU placed on the outside of each shank, 8 cm below the knee, as well an IMU sensor on the front of their chest, 6 cm below the clavicles. The subject also used the equipment developed by R. Peterka to locate center of mass of an individual [13]. This equipment included a single IMU sensor placed on the subject's back, two sway rods attached at the hip and shoulders that measured lateral body displacements, and a force plate that the participant stood on. A calibration procedure that related center of pressure measurements from the force plate to lateral motions at the hip and shoulder levels was used to derive a relationship used to calculate CoM displacement as a function of hip and shoulder

displacement. This corresponds with a location along the z axis from Figure 2.4. With all the equipment in place, the user performed three motions. Angular velocity and linear acceleration data was captured and stored on an SD card at a sampling frequency of 40 Hz. The three motions performed were a side to side full body sway, a second full body sway at a higher frequency, and finally a high frequency side to side hip sway.

Once all of the IMU sensor data was collected, it was transferred to MATLAB to build the human model. The first step was to create rotation matrices that would orient the IMU sensor frames into the Earth frame. The Earth frame's origin starts at the horizontal plane the participant stands on. It is centered along the medial plane. The x axis extends towards the anterior of the body, the y axis extends upward, orthogonal to the ground plane, and the z axis extends laterally away from the body. All the data was premultiplied by the matrices, so that the data from each sensor would be in the Earth frame.

$$\theta_E = \mathbf{R}_{Es}\theta_s \tag{6.1}$$

After all the data was oriented in the same reference frame, the angular data was used to create rotation matrices that the IMU sensor's went through during the experiment. The body segment CoM points were rotated by these rotation matrices, to find their new location in the Earth frame following the experimental motion. The initial CoM locations for each of the body segments while the participant was standing upright can be found in Table 6.1. These are the values that were found after multiplying the scaling data found in Dumas et al.'s paper [3] by the participants body segment lengths and moving them to be in reference with the Earth frame, as opposed to the anatomical landmark joint frames.

Table 6.1: Body segment center of mass locations in the Earth frame.

Segment	X Location	Y Location	Z Location
Torso	-1.7 cm	70.4 cm	0.095 cm
Pelvis	-1.8 cm	43.1 cm	1.01 cm
Left Thigh	-1.78 cm	86.3 cm	6.6 cm
Left Shank	-2.1 cm	43.8 cm	7.7 cm
Right Thigh	-1.78 cm	86.3 cm	-6.6 cm
Right Shank	-2.1 cm	43.8 cm	-7.7 cm

Once all of the new positions of the CoM locations were found, the overall CoM was calculated using Equation 2.31. The z axis location was then compared against the location retrieved from the validation equipment.

6.2 Results

The first motion performed was a full body side to side sway. The participant slowly tilted their legs and torso into the positive z direction, returned back to the standing position, and tilted in the negative z direction. This action was performed at a frequency of 0.02 Hz. The angles generated from the complementary and gradient descent filter of the torso and lower legs were used to recreate the human model and center of mass. The z axis position of the center of mass generated from the complementary and gradient descent filter can be found in Figure 6.1. We see that the complementary filter matches the shape of the CoM location, but only reaches 72.7% of the peak to peak range of the actual CoM location. The gradient descent filter also undershoots the CoM location, but is able to achieve up to 92% of the magnitude.

The second motion performed was the same side to side swaying as the first motion, however, with an increased frequency. The frequency of sway for this trial was 0.13 Hz. In Figure 6.2, we see a similar trend as the previous result, where both filters undershoot the CoM location. The complementary filter reaches 68% of the

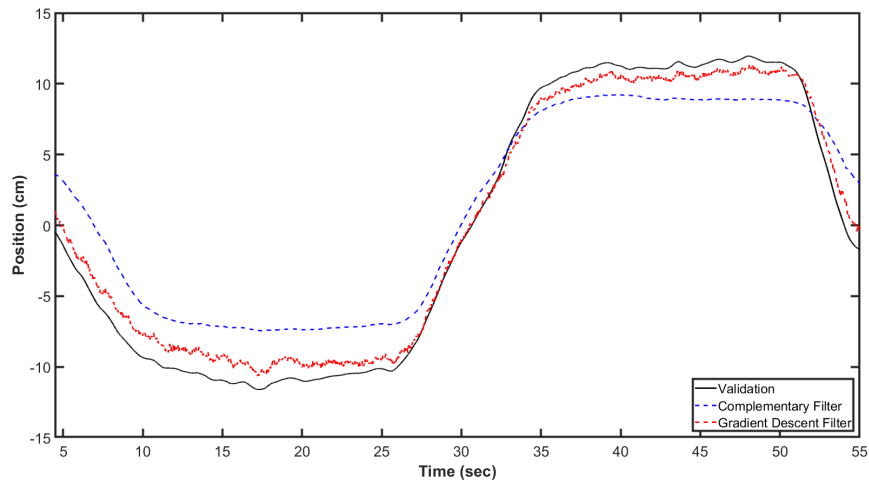


Figure 6.1: Z axis position of center of mass calculated during slow side to side swaying.

magnitude and the gradient descent filter reaches 80% of the magnitude.

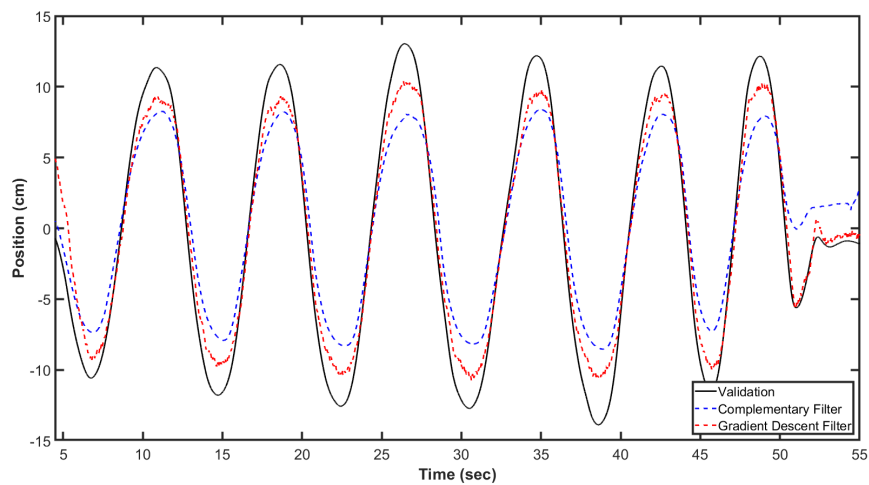


Figure 6.2: Z axis position of center of mass calculated during fast side to side swaying.

The final motion that was performed was a swaying of the hips from side to side. The hip was extended out in one direction, while the torso and head attempted to stay in the same horizontal position. The hip then swayed to the other side. This action was performed at a frequency of 0.13 Hz. Figure 6.3 shows the results from the filters. The complementary filter matched 53% of the peak to peak amplitude,

while the gradient descent matched 72%.

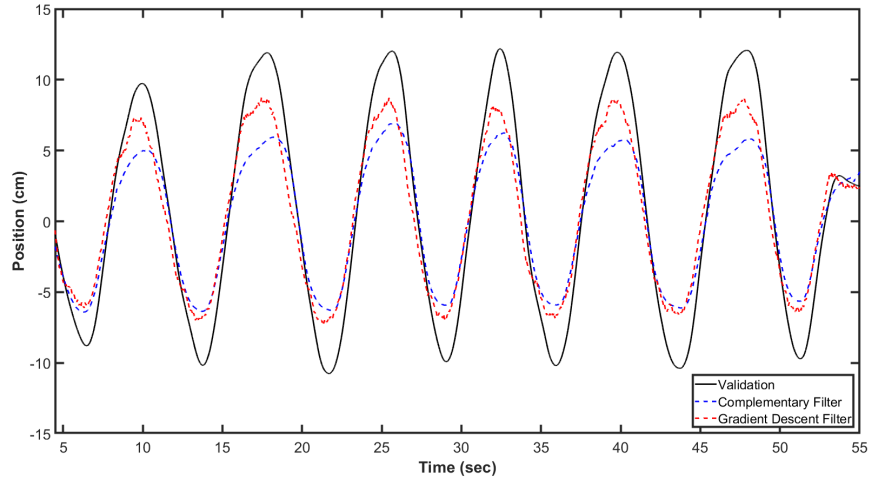


Figure 6.3: Z axis position of center of mass calculated during hip swaying.

To quantify the errors in the methods, the RMSE was calculated for the complementary and gradient descent filter. The RMSE values can be found in table 6.2. The RMSE confirms that there is a large amount of error in these methods. Both methods undershoot the actual CoM location by 0.5-3 cm.

Table 6.2: RMSE values for the center of mass experiment.

Motion	Complementary	Gradient Descent
Slow Swaying	2.28 cm	0.79 cm
Fast Swaying	2.79 cm	1.53 cm
Hip Swaying	3.06 cm	2.40 cm

6.3 Discussion

Both filters were unable to attain an accurate reading. The methods and assumptions used to model the body are too conservative. We see from the results that both the complementary filter and the gradient descent filter returned results that undershot the actual CoM location. Compared to a system like Xsens, who are able to return results within 5.45mm, these results are much worse. The error for the complementary filter ranged from 2-3 cm and the error for the gradient descent filter ranged from 0.8-2.5 cm.

The gradient descent filter performed much better than the complementary filter. For the first two experiments, the gradient descent filter had an RMSE that was about half of the RMSE of the complementary filter. The gradient descent filter was also able to match the peak to peak range of the actual CoM location by about 20% more than the complementary filter. Both struggled with the hip swaying motion, resulting in close RMSE values between the two filters.

One of the biggest assumptions the model makes is that the torso accounts for the mass that is held in the arms, forearms, and hands (Chapter 2.4). By summing those masses with the mass of the torso, the results are centralized closer to the medial plane of the body. To alleviate this issue, other assumptions can be made, but ultimately more IMU sensors should be used to monitor other body segment locations. This will provide more accurate results on where the CoM is located.

Chapter 7

Conclusion

This work has shown that there is potential in using a low number of inertial measurement units in capturing and recording data that describes where a person's center of mass is located, however, more consideration will need to be put into the model of the body that was used during analysis. This thesis focused on using three IMU sensors, one located on each lower leg and one on the upper torso, to locate an individual's center of mass. The center of mass location generated was based on the angle of these body segments. The final experiment in this thesis showed that the center of mass could be located within three centimeters of accuracy, depending on which sensor fusion algorithm was used. This was also done while the user was standing upright, with both feet planted on the ground. To detect the center of mass location of a person during gait, the algorithm used will need to be altered to take into account more limb motion.

The most difficult and crucial part of this work depends on the fusion sensor algorithm used for calculating rotation and angle of the inertial measurement units. With lower number of IMU sensors being used, the gradient descent filter is the best filter to use. It was able to track results more accurately than other filters during dynamic movement. However, as more IMU sensors are added to the system, the computational load becomes an issue. All of the analysis presented was done

within MATLAB. Attempting to perform the gradient descent filter on the board of a microcontroller can lead to errors due to the numerous calculations that are being performed. Surprisingly, the complementary filter was able to provide good results for such a simple algorithm. The complementary filter begins to lose accuracy during high frequency activities, which becomes an issue in the context of developing a system that can respond to falling actions and alert the user.

The assumptions made early when deciding where to place the IMU sensors (Chapter 2.4) were overly simplistic. The results from the final experiment show that even when excluding the forearms, arms, and hands from center of mass calculations, the calculated location undershoots where the actual center of mass location is. By grouping extra limb segments into the torso segment, the center of mass location becomes concentrated along the medial plane. This leads to inaccurate results. To develop a more robust system, two or more extra sensors should be included in the system that can monitor arm locations.

7.1 Future Work

The use of IMU sensors for motion and center of mass tracking is not a new concept, however, there haven't been major developments to making their systems easy to use for all individuals, and the cost of these systems are prohibitive to a majority of the populace. Because of that, this research poses new prospects to bringing this technology to many individual's who would benefit from a fall prevention method.

One of the next steps to take in this research would be to continue improving on the equipment to make a more robust and faster system. The first thing that can be changed is the choice in microcontroller. The Adafruit Adalogger was used in this research because of it's ease of use, low cost, and availability of on chip SD card storage. However, other microcontrollers can offer further improvements in processing

power, which would be necessary for performing IMU sensor filtering and center of mass calculations.

Another piece of equipment that could be improved upon is the inertial measurement units. The IMU sensors in this research were the lowest cost sensors found that provided enough information at a fast enough speed. There are a multitude of other IMU sensors that could be used, however their cost makes it prohibitive for a low cost system. However, if higher accuracy is desired, then cost can be sacrificed. The center of mass experiment performed showed that there were large errors between the calculated location and the actual location. While the error most likely lies with the model of the human body used during calculations, it is worth investigating if higher cost, higher performance IMU sensors would lead to more accurate results.

Other than testing new pieces of equipment, the next thing to test would be to investigate other sensor fusion algorithms. Some other filters were mentioned in this paper, specifically the Kalman filter, but were never tested. While the Kalman filter is a prevalent sensor fusion technique used in many applications, it is the most prohibitive in computational power and complexity. Because of that, it was not tested during the course of this research. However, now that a standard for center of mass tracking has been established, it would be beneficial to attempt to implement it and compare its results to the other filters used during this thesis. The filter could be tested on MATLAB. If it shows significant improvement, then it can be tested on a microcontroller with better computational power than the one used during this thesis.

In order to further validate the results of the center of mass location, other experiments can be created that are validated through visual motion tracking systems. As mentioned in Chapter 2, visual methods are the most common and accurate way of determining an individual's center of mass. By performing an experiment with this form of validation, the center of mass can be tracked in 3 dimensions, while the final

experiment of this thesis was only validated in one dimension.

Other experiments to conduct would be to observe how well these systems work in more dynamic situations, specifically during higher frequency tasks, such as during gait, running, walking up stairs, etc. The algorithms used focused primarily on determining the sensor orientation during slower movements and while standing in one place. The biggest issue with trying to monitor center of mass during gait is the issue of detecting limb position in space. When both feet are planted on the ground, or during slow movement, it is easy to detect where their position is. However, while walking there needs to be continuous computations must be done to measure translational motions of body segments, which would require an integration of the acceleration signal. As seen with the gyroscopes, integrating signal data leads to significant drift if not handled properly. Having to integrate the acceleration data twice to determine position would lead to significant errors.

The results shown in this thesis point to a possibility of creating a low cost center of mass tracking device. Future work would complete a device that can monitor and record center of mass location and posture, while also alerting the user when a fall is about to occur.

Bibliography

- [1] Dora E. Angelaki and Kathleen E. Cullen. Vestibular System: The Many Facets of a Multimodal Sense. *Annual Review of Neuroscience*, 31(1):125–150, July 2008.
- [2] Daniel Dinu, Martin Fayolas, Marine Jacquet, Elsa Leguy, Jean Slavinski, and Nicolas Houel. Accuracy of Postural Human-motion Tracking Using Miniature Inertial Sensors. *Procedia Engineering*, 147:655–658, 2016.
- [3] R. Dumas, L. Chèze, and J.-P. Verriest. Adjustments to McConville et al. and Young et al. body segment inertial parameters. *Journal of Biomechanics*, 40(3):543–553, January 2007.
- [4] Pengfei Gui, Liqiong Tang, and Subhas Mukhopadhyay. MEMS based IMU for tilting measurement: Comparison of complementary and kalman filter based data fusion. In *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, pages 2004–2009, Auckland, New Zealand, June 2015. IEEE.
- [5] A.L. Hof, M.G.J. Gazendam, and W.E. Sinke. The condition for dynamic stability. *Journal of Biomechanics*, 38(1):1–8, January 2005.
- [6] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a Real-Time Human Movement Classifier Using a Triaxial

- Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, January 2006.
- [7] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, Zurich, June 2011. IEEE.
- [8] Thomas Mergner. Modeling sensorimotor control of human upright stance. In Paul Cisek, Trevor Drew, and John F. Kalaska, editors, *Computational Neuroscience: Theoretical Insights into Brain Function*, volume 165 of *Progress in Brain Research*, pages 283 – 297. Elsevier, 2007.
- [9] Thomas Mergner and Stefan Glasauer. A Simple Model of Vestibular Canal-Otolith Signal Fusion. *Annals of the New York Academy of Sciences*, 871:430–434, 1999.
- [10] Thomas Mergner, Georg Schweigart, and Luminous Fennell. Vestibular humanoid postural control. *Journal of Physiology-Paris*, 103(3-5):178–194, May 2009.
- [11] Yi-Chung Pai and James Patton. Center of Mass Velocity Position Prediction for Balance Control.pdf. *Journal of Biomechanics*, 30(4):347–354, 1997.
- [12] R. J. Peterka. Sensorimotor Integration in Human Postural Control. *Journal of Neurophysiology*, 88(3):1097–1118, September 2002.
- [13] Robert J. Peterka, Charles F. Murchison, Lucy Parrington, Peter C. Fino, and Laurie A. King. Implementation of a Central Sensorimotor Integration Test for Characterization of Human Balance Control During Stance. *Frontiers in Neurology*, 9:1045, December 2018.

- [14] Stephen N Robinovitch, Fabio Feldman, Yijian Yang, Rebecca Schonnop, Pet Ming Leung, Thiago Sarraf, Joanie Sims-Gould, and Marie Loughin. Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study. *The Lancet*, 381(9860):47–54, January 2013.
- [15] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens MVN: Full 6dof Human Motion Tracking Using Miniature Inertial Sensors. *Xsens Technologies*, page 10, 2013.
- [16] Ken Shoemake. animating rotation with quaternion curves.pdf. *ACM SIG-GRAPH*, 19(3):245–254, 1985.
- [17] Mary E Tinetti. Preventing Falls in Elderly Persons. *The New England Journal of Medicine*, page 8, 2003.
- [18] Che-Chang Yang and Yeh-Liang Hsu. A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring. *Sensors*, 10(8):7772–7788, August 2010.
- [19] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear Algebra and its Applications*, 251:21–57, January 1997.
- [20] H. Zheng, N. D. Black, and N. D. Harris. Position-sensing technologies for movement analysis in stroke rehabilitation. *Medical & Biological Engineering & Computing*, 43(4):413–420, August 2005.
- [21] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation—A survey. *Biomedical Signal Processing and Control*, 3(1):1–18, January 2008.

Appendix

A.1 Discrete Vestibular Emulating Filter Algorithm

```
1 function [gphix, gphiy, gphiz] = VestibularAHRS(gyro, a, Ts)
2
3 %This function takes in angular velocity and linear acceleration
4 %measurements, as well as the sampling time, in order to derive the
5 %orientation of an inertial measurement unit. The orientation is
6 %described with three angles, generated from orientation of a
7 %calculated gravity vector compared with the orientation of a
8 %constant gravity vector. This algorithm is based off of the
9 %algorithms created by T. Mergner et al.
10
11 %Organizing gyroscope measurements
12 gyro_x = gyro(:, 1);
13 gyro_y = gyro(:, 2);
14 gyro_z = gyro(:, 3);
15
16 %High pass filter for Gyro measurements
17 HPtau = 5;
18 HP = tf([HPtau 0], [HPtau 1]);
19 DHP = c2d(HP, Ts, 'ZOH');
20 HPN1 = DHP.Numerator{1,1}(1);
21 HPN2 = DHP.Numerator{1,1}(2);
22 HPD1 = DHP.Denominator{1,1}(1);
23 HPD2 = DHP.Denominator{1,1}(2);
24
25 %Low pass filter for Accelerometer measurements
26 LP = 1 - HP;
```

```

27 DLP = c2d(LP, Ts, 'ZOH');
28 LPN1 = DLP.Numerator{1,1}(1);
29 LPN2 = DLP.Numerator{1,1}(2);
30 LPD1 = DLP.Denominator{1,1}(1);
31 LPD2 = DLP.Denominator{1,1}(2);
32
33 % Setting up leaky integrator
34 LIgain = 20;
35 LI = tf(LIgain, [LIgain 1]);
36 DLI = c2d(LI, Ts, 'ZOH');
37 LIN2 = DLI.Numerator{1,1}(2);
38 LID1 = DLI.Denominator{1,1}(1);
39 LID2 = DLI.Denominator{1,1}(2);
40
41 %Setting up iterative variables for canal estimation
42 gyroxm1 = 0; gyroym1 = 0; gyrozm1 = 0;
43 omegacxm1 = 0; omegacym1 = 0; omegaczml = 0;
44
45 %Setting up iterative variables for otolith estimation
46 axnormm1 = a(1, 1)/norm(a(1, :));
47 aynormm1 = a(1, 2)/norm(a(1, :));
48 aznormm1 = a(1, 3)/norm(a(1, :));
49 omegaoxm1 = 0; omegaoyml = 0; omegaozml = 0;
50 omegaestxm1 = 0; omegaestym1 = 0; omegaestzml = 0;
51
52 %Setting up variables for gravity estimation
53 accelgain = 1/LIgain;
54 g = 9.8067;
55 gxm1 = a(1, 1);
56 gym1 = a(1, 2);
57 gzm1 = a(1, 3);

```

```

58 gxdot = 0; gxdotm1 = 0;
59 gydot = 0; gydotm1 = 0;
60 gzdot = 0; gzdotm1 = 0;
61
62 %Gain Correction
63 meana = sqrt(mean(a(1:100, 1))^2+mean(a(1:100, 2))^2+mean(a(1:100, ←
        3))^2);
64 gc = -9.807/meana;
65 a(:, 1) = gc*a(:, 1);
66 a(:, 2) = gc*a(:, 2);
67 a(:, 3) = gc*a(:, 3);
68
69 for ii = 1:length(gyrox)
70     %Gyro high pass filter
71     omegacx = (HPN1*gyrox(ii) + HPN2*gyroxm1-HPD2*omegacxm1)/HPD1;
72     omegacy = (HPN1*gyroy(ii) + HPN2*gyroym1-HPD2*omegacym1)/HPD1;
73     omegacz = (HPN1*gyroz(ii) + HPN2*gyrozm1-HPD2*omegaczm1)/HPD1;
74     gyroxm1 = gyrox(ii); gyroym1 = gyroy(ii); gyrozm1 = gyroz(ii);
75     omegacxm1 = omegacx; omegacym1 = omegacy; omegaczm1 = omegacz;
76
77     %accel normalization
78     amag = norm(a(ii, :));
79     axnorm = a(ii, 1)/amag;
80     aynorm = a(ii, 2)/amag;
81     aznorm = a(ii, 3)/amag;
82
83     %accel differentiation
84     axder = (axnorm - axnormm1)/Ts;
85     ayder = (aynorm - aynormm1)/Ts;
86     azder = (aznorm - aznormm1)/Ts;
87     axnormm1 = axnorm;

```

```

88     aynormm1 = aynorm;
89     aznormm1 = aznorm;
90
91     %cross product for otolith angular velocity
92     omegaox = (ayder*aznorm - aynorm*azder);
93     omegaoy = -(axder*aznorm - axnorm*azder);
94     omegaoz = (axder*aynorm - axnorm*ayder);
95
96     %Otolith low pass filter
97     omegaestx = (LPN2*omegaoxm1 - LPD2*omegaestxm1)/LPD1;
98     omegaesty = (LPN2*omegaoym1 - LPD2*omegaestym1)/LPD1;
99     omegaestz = (LPN2*omegaozm1 - LPD2*omegaestzm1)/LPD1;
100    omegaoxm1 = omegaox; omegaoym1 = omegaoy; omegaozm1 = omegaoz;
101    omegaestxm1 = omegaestx;
102    omegaestym1 = omegaesty;
103    omegaestzm1 = omegaestz;
104
105    %Canal - Otolith fusion
106    omx = omegacx + omegaestx;
107    omy = omegacy + omegaesty;
108    omz = omegacz + omegaestz;
109
110    %Cross Product for change in direction of gravity
111    gxdot = gym1*omz - gzm1*omy + accelgain*a(ii , 1);
112    gydot = gzm1*omx - gxm1*omz + accelgain*a(ii , 2);
113    gzdot = gxm1*omy - gym1*omx + accelgain*a(ii , 3);
114
115    %Leaky Integrator
116    gravx = (LIN2*gxdotm1 - LID2*gxm1)/LID1;
117    gravy = (LIN2*gydotm1 - LID2*gym1)/LID1;
118    gravz = (LIN2*gzdotm1 - LID2*gzm1)/LID1;

```

```

119
120     gxml = gravx; gym1 = gravity; gzml = gravz;
121     gxdotml = gxdot; gydotml = gydot; gzdotml = gzdot;
122
123     %saving calculated values
124     grav(ii, :) = [gravx, gravity, gravz];
125 end
126
127 %Fusing gyro and accelerometer data to come up with an angle
128 accelBiasZ = atan2(a(1, 2), a(1, 1));
129 accelBiasX = atan2(a(1,2), a(1, 3));
130 accelBiasY = atan2(a(1, 3), a(1, 1));
131
132 for ii = 1:length(om)
133     gphix(ii) = atan2(grav(ii, 2), grav(ii, 3))-accelBiasX;
134     gphiy(ii) = atan2(grav(ii, 3), grav(ii, 1))-accelBiasY;
135     gphiz(ii) = atan2(grav(ii, 2), grav(ii, 1))-accelBiasZ;
136 end
137
138 end

```