Pedestrian Navigation using Particle Filtering and *a priori* Building Maps

by

Tanner Norman Ray

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
August 3, 2019

Keywords: Pedestrian Dead Reckoning, Indoor Positioning, Particle Filtering, Kalman Filtering, Inertial Measurement Unit, Map-matching

Approved by

David Bevly, Chair, Professor of Mechanical Engineering
Scott Martin, Assistant Research Professor of Mechanical Engineering
Michael Zabala, Assistant Professor of Mechanical Engineering

Abstract

This thesis presents a new particle filter (PF) weight update method that improves the performance of indoor positioning systems. In standalone inertial pedestrian-dead-reckoning (PDR) systems, the position error grows with time due to the inertial measurement unit's (IMU) sensor errors. Often external measurements from GPS or radio networks (e.g. wireless local area network (WLAN), ultra-wide-band (UWB), Bluetooth low energy (BLE) etc.) are used to restrict the error growth. External measurements from infrastructure-based systems have inherent high costs and deployment time, thus they are not easily implemented. The presented work focuses on the development of a standalone wearable navigation system that does not depend on physical infrastructure. Therefore, to constrain error growth without external measurements, other techniques have been developed that utilize building map information as a measurement. One method uses the building to provide heading measurements to reduce the drift in the heading solution. This is based upon the fact that pedestrians typically walk straight when walking in building corridors. Another method constrains the error based upon the knowledge that pedestrian motion is limited by building floorplans, (e.g. walls, floors, and other features). This technique uses PF estimation to fuse standalone PDR with map measurements to perform accurate pedestrian localization. These techniques along with the current PDR techniques and underlying algorithms are discussed in detail. Lastly, this work presents a comparison of PFs that utilize different particle propagation and weight update methods for indoor positioning systems. A new type of weight update is also introduced that provides more accurate localization. A performance evaluation of the new weight update method is shown with both simulated and experimental data. The conclusions drawn from the evaluation results are discussed and a summary is provided.

Acknowledgments

First and foremost, I would like to thank my parents Kim and Joan for their constant love and support and many words of wisdom. Without them, the completion of this degree would not have been possible. I hope that oneday I will be able to repay them for the love they have shown me. This thesis is dedicated to them. I would also like to thank my siblings Cameron, Caroline, and Gannon, for each helping shape me into the person I am today. Gannon should receive special thanks for always answering his phone at any time of day or night to talk about the topics of the day.

A person who deserves huge thanks is my Honors Precalculus and AP Calculus AB teacher Mrs. Amanda Bittinger. She is a huge source of inspiration to me because of her passion for teaching and her students. She truly changed my life for the better when she fostered enjoyment of both mathematics and learning in me.

I must mention the many coworkers I have had the pleasure to work with during my time in the GAVLAB. They are what make a graduate program a rich experience. The first person I must thank is Amos Smith, he was the first person to work with me when I entered the lab, and provided me much help. A debt of gratitude goes to Dan Pierce for his many hours of help and advice. Without it, this work would surely not have been possible. Similarly, I would like to thank Grant Apperson for his help, willingness to share his knowledge, and similar appreciation of tools. I would also like to mention Stephen Geiger, Thomas 'Troupe' Tabb, Ethan Edwards, and Joe Selikoff for being true friends I value considerably. The other lads (Patrick Smith, Luke Kamrath, Robert Brothers) that made up the 'Siberia' office also deserve thanks for the many interesting discussions and helpful input that always made work fun. Lasty, my friend and fellow name bearer Tanner Watts, deserves mention for the interesting discussions about the topics of the day, GPS, and the NFL.

Dr. Howard Chen and Dr. Mark Schall both also deserve thanks, as they were always willing to lend equipment and help me in my research. I would also like to thank the members

of my committee Dr. Michael Zabala and Dr. Scott Martin for reviewing this document and providing me with helpful and encouraging comments.

I would like to thank my favorite YouTube channels PewDiePie and MrBeast for their humorous content that helped lighten the mood when the completion of this thesis seemed miles away.

I also would like to thank Dr. David Bevly for the opportunity to work in his lab as it has changed the course of my life for the better. I have had the chance to work on many interesting projects and learn about topics, I would not have been able to otherwise. This opportunity precipitated from taking his MECH 3140 class, which I have had the pleasure of continually being involved with through teaching a recitation session.

"For I know the plans I have for you, declares the LORD, plans for welfare and not for evil, to give you a future and a hope." – Jeremiah 29:11. I am deeply gladdened by the fact that God's plans for my life are not my own as I never would have pictured myself where I am today. I am also deeply thankful for God's provision for sinners like myself that we may be saved and know Jesus Christ as our Savior and Lord. "By grace you have been saved through faith; and that not of yourselves, it is the gift of God; not as a result of works that no one should boast" – Ephesians 2:8-9

Table of Contents

Abstract ii

Acknowledgments iii

List of Acronyms xii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Research Aims and Objectives 3

1.3 Thesis Outline 4

2 Coordinate Frames and Transformations 5

2.1 Introduction 5

2.2 Coordinate Frames 5

2.2.1 Inertial Frame 5

2.2.2 Body Frame 6

2.2.3 Local Navigation Frame 7

2.2.4 Pedestrian Frame 8

2.3 Coordinate Frame Transformations 9

2.3.1 Transformation from the Local Navigation Frame to the Body Frame . . 11

2.3.2 Transformation from the Body Frame to the Local Navigation Frame . . 11

2.4 Relationship between Euler Angles and Coordinate Transform Matrices 12

2.4.1 Local Navigation Frame to Body Frame CTM 12

2.4.2 Body Frame to Local Navigation Frame CTM 13
2.4.3 Euler Angle Computation Issues 13

3 Pedestrian Dead Reckoning: An INS Approach 14
3.1 Introduction 14
3.2 Pedestrian Gait Cycle 15
3.3 PDR-INS Systems 16
3.3.1 Inertial Measurement Unit Error Model 17
3.3.2 Gait Monitoring 18
3.3.3 Straight Walking Detection 25
3.3.4 The Discrete-Time Linear Kalman Filter 25
3.3.5 Extended Kalman Filter 28
3.3.6 Error-State Kalman Filter 28
3.3.7 INS Mechanization 40

4 Pedestrian Dead Reckoning: A Step Detection Approach 45
4.1 Introduction 45
4.2 PDR-SD Systems 45
4.3 Step Detection 45
4.3.1 The Effects of IMU Location on Step Detection 52
4.4 Step Length Determination 57
4.4.1 Linear Least Squares Estimation 61
4.4.2 Empirical Step Length Formula Comparison 64
4.5 Heading Determination 71
4.5.1 Magnetometer Heading 71
4.5.2 Heading from a Single Gyroscope 72
4.5.3 Gyro + Magnetometer Heading 74

4.5.4 Attitude Heading Reference Systems . . . 75
4.6 Navigation System Update . . . 75

5 Particle Filtering for Indoor Positioning Systems . . . 77
5.1 Introduction . . . 77
5.2 Estimation Framework . . . 77
5.3 Particle Filtering . . . 77
5.3.1 Initialization . . . 80
5.3.2 Propagation with PDR . . . 81
5.3.3 Propagation with PDR + Map . . . 86
5.3.4 Update . . . 86
5.3.5 Resampling . . . 89
5.4 Particle Inbounds Check . . . 91
5.5 Particle Filters . . . 93
5.5.1 Collision Particle Filter (PFc) . . . 94
5.5.2 Map-based measurement likelihood (PFw) . . . 94
5.6 New Weight Update . . . 94
5.7 Performance Evaluation . . . 99
5.7.1 Error Statistics . . . 99
5.7.2 Simulated Data Generation . . . 101
5.7.3 Simulated Results . . . 101
5.7.4 Experimental Data Collection . . . 104
5.7.5 Experimental Results . . . 105

6 Summary, Conclusions, and Future Work . . . 108
6.1 Summary . . . 108
6.2 Conclusions . . . 110

6.3 Future Work 110

References 112

List of Figures

2.1 Body Frame (Foot) 7

2.2 Body Frame (Torso) 7

2.3 Local Navigation Frame (NED) [41] 8

2.4 Pedestrian Coordinate Frame 9

2.5 Euler Angle Rotations 10

3.1 Walking Gait Cycle [25] 15

3.2 Step Versus Stride [25] 16

3.3 Gyro Output during Stance and Swing Phase 19

3.4 Zero-Velocity Detection 21

3.5 Still Detection 22

3.6 Stride Detection 24

3.7 Position Estimates with ZVU Method 37

3.8 Position Estimates with ZVU+ZARU Method 38

3.9 Position Estimates with ZVU+HDR Method 38

3.10 Position Estimates with ZVU+ZARU+CHAIN Method 39

4.1 Raw Accelerometer Measurements During Walking 47

4.2 Basic Step Detection [3, 47, 49, 51] 48

4.3 Step Detection 49

4.4 Possible fixed IMU locations on pedestrian body [3] 52

4.5 MVN Awinda System (Image courtesy of Auburn Univ. Industrial Eng. Dept.) . 54

4.6 Test #1 Step Detection 55

4.7 Test #1 Step Detection 55
4.8 Test #2 Step Detection Head 56
4.9 Test #2 Step Detection Pelvis 56
4.10 Pedestrian Path Used for Estimation of Step Length 69
5.1 Indoor Positioning System Estimation Framework 78
5.2 Two states represented with a bivariate Gaussian distribution using a mean and covariance (left), and a set of particles (right) [3] 79
5.3 Particle Filter Steps [3] 79
5.4 Particle Inbounds Check [13] 92
5.5 PFc Error in Corridor 95
5.6 PFc Error in Room 96
5.7 Inbounds Particle Filter 97
5.8 Environment Animation 101
5.9 User Path of a Simulated Pedestrian 101
5.10 Position Errors 102
5.11 Position RMS Error 103
5.12 Test #1 User Path 106
5.13 Test #1 Position Error 106
5.14 Test # 2 User Path 107
5.15 Test # 2 Position Error 107

List of Tables

3.1 Discrete-Time Linear Kalman Filter 27

4.1 Algorithm 1 Parameter Definitions 50

4.2 SL_3 Definition 58

4.3 OLS Estimates 69

4.4 WLS Estimates 70

4.5 Step Length RMS Error 70

5.1 Test # 1 and # 2 RMS Error 106

List of Acronyms

1D One Dimensional

2D Two Dimensional

3D Three Dimensional

AoA Angle of Arrival

BF Bootstrap Filter

BLE Bluetooth Low Energy

CHAIN Cardinal Heading Aided for Inertial Navigation

CTM Coordinate Transform Matrix

DoF Degree of Freedom

DR Dead Reckoning

ECI Earth Centered Inertial

EKF Extended Kalman Filter

ENU East North Up

ESKF Error State Kalman Filter

GAVLAB GPS and Vehicle Dynamics Lab

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

HDR Heuristic Drift Reduction

IMU Inertial Measurement Unit

INS Inertial Navigation System

JOSM Java Open Street Map

KF Kalman Filter

MEMS Micro-Electromechanical Systems

NED North East Down

OLS Ordinary Least Squares

OSM Open Street Map

PDF Probability Density Function

PDR Pedestrian Dead Reckoning

PDR-INS Pedestrian Dead Reckoning, INS approach

PDR-SD Pedestrian Dead Reckoning, SD approach

PF Particle Filter

PFc Collision Particle Filter

PFci Collision Inbounds Particle Filter

PFi Inbounds Particle Filter

PPP Precise Point Positioning

RFID Radio Frequency Indentification

RLS Recursive Least Squares

RMSE Root Mean Square Error

RSS Received Signal Strength

RTK	Real Time Kinematic
SD	Step Detection
SL	Step Length
TDCP	Time Differenced Carrier Phase
ToF	Time of Flight
UWB	Ultra Wide Band
WLAN	Wireless Local Area Network
WLS	Weighted Least Squares
ZARU	Zero Angular Rate Update
ZVU	Zero Velocity Update

Chapter 1

Introduction

This chapter begins with a brief background on the current state of pedestrian navigation systems. The current challenges of the field are discussed and some of the solutions proposed to solve them. Next, the remaining chapters are outlined.

1.1 Background and Motivation

The problem of determining *where we are* is not a new matter of concern. Since roughly 3000 BC, sailors have used positioning techniques to navigate the open seas [1]. The American Practical Navigator defines navigation as, "A field of study that focuses on the process of monitoring and controlling the movement of a craft or vehicle from one place to another" [2]. This definition, albeit outdated, has two distinct parts that are still extremely relevant today. The first part is concerned with the position determination of a moving body with respect to a well-defined reference frame. This is a subset of navigation that is known as *positioning* [3]. Note the term *localization* is often used synonymously with positioning in many works. Localization is more appropriately defined as indoor positioning (i.e. knowing your location indoors). The second part refers to the determination of the desired path of travel from one location to another. This other branch of navigation is called *guidance*, which is not discussed further in this work.

Reliable and accurate positioning is needed for many applications such as orbit determination of satellites, pedestrian navigation, autonomous vehicles, and guided munitions. Pedestrian navigation has received a lot of interest in the research community in recent years for three reasons. The first is the increase of processing power of mobile computing platforms. The second is the availability of sensors that have a reduced size, weight, and power requirements. Lastly, it has many useful applications which include, but are not limited to: geolocation of dismounted soldiers, navigating the blind and visually impaired, and tracking first responders [3, 4]. In the

next section, the current techniques of the field and the challenges that are still faced will be discussed.

The most common application of pedestrian navigation seen today is navigation in urban areas using smartphones. Smartphones contain a global positioning system (GPS) receiver, which provides users with reliable ubiquitous positioning as GPS has long term stability and its position error does not grow unbounded with time. Despite this, obstacles still exist in navigating pedestrians within challenging environments. Examples of challenging environments are: inside buildings, urban canyons, areas with dense foliage, or hostile warzones where jamming may occur. Global navigation satellite systems (GNSS) (e.g. GPS, GLONASS, Galileo, and Beidou) and other long-range radio navigation signals are either degraded or are completely unavailable as the power levels of these signals are extremely weak [5]. GPS for instance has a received signal power of $1 \times 10^{-16}\,\mathrm{W}$ [5, 6]. These conditions are why pedestrian navigation is one of the most challenging applications of navigation science today, presenting various problems that are nontrivial to solve.

The constraint of having to navigate in GNSS denied environments resulted in the development of foot and body mounted inertial pedestrian-dead-reckoning (PDR) systems. These were made possible with the advent of micro-electromechanical systems (MEMS) technology. MEMS sensors are small and light and exhibit high shock tolerance along with low power consumption. This is highly practical for being used as a body worn sensor. Moreover, they are mass produced which makes them low cost and widely available [3]. An inertial based system is considered to be a self-positioning system as it does not depend upon external measurements, except when initializing the system. The main drawback of an INS is unbounded error growth with time due to the inertial measurement unit (IMU) sensor errors. This is especially true for MEMS IMUs as they are low cost. Note that all of the IMUs used in this work are tactical grade IMUs as specified by Groves in [3]. In place of GNSS measurements, many use external measurements from other infrastructure-based systems to restrict the INS error growth. Examples of such systems are short-range radio networks (e.g. wireless local area network (WLAN), ultra-wide-band (UWB) radios, bluetooth low energy (BLE) etc.) and radio-frequency identification (RFID). BLE, WLAN, and UWB radios can use the strategies

of received signal strength (RSS), time of flight (ToF), or angle of arrival (AoA) to determine the location of the transmitting node [3, 4, 7–10]. With three or more locations of transmitting nodes known in regard to the receiving node, an estimate of its location can be found with either trilateration or triangulation. RFID systems leverage the knowlege of the RFID tag locations, therefore when a user is in proximity to a tag, the position is said to be near the position of the tag [3]. These systems have inherent high costs and deployment time, thus they are not easily implemented. The work presented in this thesis, focuses on the development of a standalone wearable navigation system that does not depend on physical infrastructure.

In order to constrain error growth without external measurements, other techniques must be utilized. One technique that has been popularized uses an *a priori* building map as a measurement [3,4,8–18]. This is based upon the knowledge that indoor pedestrian motion is limited by building floorplans, (e.g. walls, floors, and other features). The techniques that use building map information will be explored and improved within this thesis.

1.2 Research Aims and Objectives

The goals that this thesis aims to complete are listed below.

- Provide a thorough background on pedestrian-dead-reckoning methods, including INS and step detection methods.
- Investigate current methods of fusing *a priori* building map information with PDR systems.
- Introduce a new particle filtering weight update method that improves the performance of indoor positioning systems.
- Propose a new method of obtaining experimental data to evaluate indoor positioning systems using an accurate external reference.

1.3 Thesis Outline

This thesis has five remaining chapters. Chapter 2 discusses the coordinate frames used and how to transform between different coordinate frames. Chapter 3 discusses pedestrian-dead-reckoning methods that utilize an INS approach. The following chapter investigates pedestrian dead reckoning using the newer step detecting method. Chapter 5 discusses the steps of particle filters applied to indoor positioning systems. Next, the current methods and the new weight update method are introduced. These methods are compared in a detailed analysis that shows that the new weight update provides improved performance. The final chapter includes a summary, conclusions, and future work.

Chapter 2

Coordinate Frames and Transformations

2.1 Introduction

Navigation systems require various frames of reference and the transformations between them in order to produce a navigation solution. For instance, an INS uses an IMU to measure the changes of the body (i.e. acceleration and angular velocity) relative to the inertial frame. The changes on their own are not particulary useful for navigation purposes, therefore they need to be transformed to a navigation frame. The remainder of this chapter will discuss the coordinate frames used in this thesis and how to convert values between the different frames.

2.2 Coordinate Frames

This section describes the coordinate frames and the associated notation used in this thesis. Note that the body and local navigation coordinate frames are all right-handed orthogonal Cartesian systems.

2.2.1 Inertial Frame

The inertial frame is a reference frame that is non-accelerating. This is the fundamental assumption that Newton's laws are based upon. For global navigation, the most popular solution is to define an inertial frame at a given initial time that is coincident with the earth's center of mass. This frame is called earth centered inertial (ECI). Inertial measurements of the body of interest are taken with respect to the inertial reference frame. The symbol i is used to denote the inertial frame.

2.2.2 Body Frame

The body frame is attached to the object of interest that the navigation solution refers to. The term *body frame* is ambiguously used in navigation literature, as it is used to describe different things. In vehicle navigation for example, body frame can be referring to the vehicle frame or the sensor (instrument) frame of the IMU that is mounted to the vehicle. In this work, it is used to describe the sensor frame of the IMU (i.e. the sensor's sensitive axes). Note that these axes remain fixed with respect to the body. The most common convention is adopted in this work where the x-axis is in the forward direction (i.e. in the direction of travel), the y-axis is to the right, and the z-axis is in the downward direction. Note that the x-axis direction is commonly referred to as the longitudinal direction and the y-axis direction is referred to as the lateral direction. For angular motion, the rotation about the x-axis is known as roll (ϕ), the rotation about the y-axis is known as pitch (θ), and the rotation about the z-axis is known as yaw (ψ). The right hand rule applies such that if the axis is pointing away then the positive rotation direction is clockwise. The symbol *b* is used to denote the body frame.

An IMU's accelerometers measure specific force and its gyroscopes (gyros) measure the angular velocity of the body to which it is attached. The specific force readings for the x, y, and z axes are $\{\tilde{f}^b_{ib,x}, \tilde{f}^b_{ib,y}, \tilde{f}^b_{ib,z}\}$, respectively. The subscript ib should be read as the specific force of the body frame (b) with respect to the inertial frame (i). The superscript is the resolving frame (i.e. the frame the measurements are in), in this case, the body frame (b). Similarly, the angular velocity readings for the x, y, and z axes are $\{\tilde{\omega}^b_{ib,x}, \tilde{\omega}^b_{ib,y}, \tilde{\omega}^b_{ib,z}\}$, respectively. The subscript ib should be read as the angular velocity of the body with respect to the inertial frame. The superscript is the resolving frame, which in this case it is again the body frame (b).

The IMU in this thesis is either located on the user's torso near the sternum or on top of the user's shoe. This location is dependent upon which pedestrian-dead-reckoning technique is used, which is discussed at length in Chapter 4. The body frame of a shoe-mounted IMU is illustrated by Figure 2.1. The body frame the torso mounted IMU is shown in Figure 2.2.

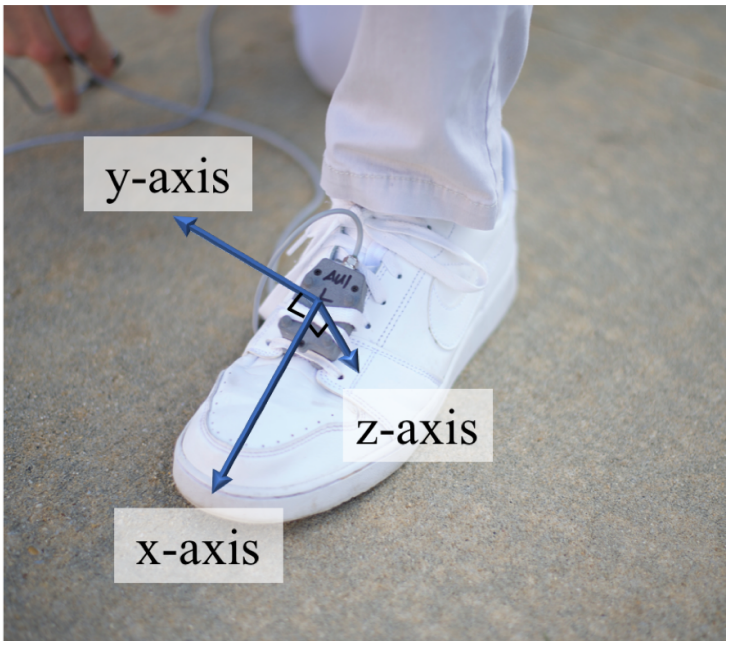

Figure 2.1: Body Frame (Foot)

Figure 2.2: Body Frame (Torso)

2.2.3 Local Navigation Frame

The local navigation frame has a fixed origin on the Earth, most commonly a point on the surface. This frame is found by fitting a tangent plane to the geodetic reference ellipsoid (e.g.

WGS84) at a point of interest [19]. The point of interest is a location convenient for local measurements and is the origin of the local tangent plane. This plane is used for local navigation, with applications such as indoor positioning and aircraft landing [3]. The two most popular coordinate conventions for local navigation frames are East-North-Up (ENU) and North-East-Down (NED). For this work, the NED convention is used where the x-axis points towards North, the y-axis points East, and the z-axis points Down. When this frame is aligned with the topographic directions (i.e. North and East) it is also be referred to as the local geodetic frame or topcentric frame [3]. Figure 2.3 illustrates the local navigation frame with the NED convention. The symbol *n* is used to denote the local navigation frame.

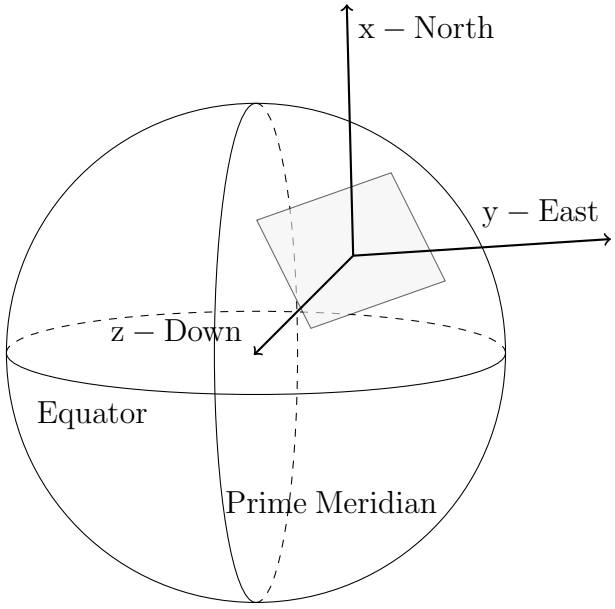

Figure 2.3: Local Navigation Frame (NED) [41]

2.2.4 Pedestrian Frame

The three axes of motion for an individual can be seen in Figure 2.4, where the origin is at the pedestrian's center of gravity. This is the true frame of a pedestrian. This frame is not used in this thesis as a frame of reference, but will be referred to in discussion of IMU location in Section 4.3.1.

Figure 2.4: Pedestrian Coordinate Frame

2.3 Coordinate Frame Transformations

Coordinate transformations are needed to transform both measured and computed values between various frames of reference [19]. The relative angular orientation of two coordinate frames is known as *attitude*, it can be represented by an orthonormal coordinate transform matrix (CTM) [20, 21]. The CTM can be broken down into three successive rotations about different axes, with each axis orthogonal to the previous and or following axis. A transformation from a coordinate frame to a new coordinate frame is as follows: rotate by angle ψ about the current z-axis, rotate by angle θ about the new y-axis, and rotate by angle ϕ about the new x-axis. These three angles are known as Euler angles (pronounced "oiler"), where $\{\psi, \theta, \phi\}$ are known as yaw, pitch, and roll, respectively. This representation is popular as it is the most intuitive way to represent attitude. Note that Euler angles exhibit singularity errors at $\pm 90°$ pitch causing the solution for roll and yaw to be indeterminate. This problem is further discussed in Section 2.4.3. The three rotations for an Euler attitude representation can be

expressed mathematically in Equations (2.1-2.3).

$$\text{rotation about the z-axis, } \mathbf{C}_1 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$$\text{rotation about the y-axis, } \mathbf{C}_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.2}$$

$$\text{rotation about the x-axis, } \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{2.3}$$

Each of these rotations are illustrated by Figure 2.5. The axes denoted with a prime symbol ($'$), are the coordinates after the rotation has occured.

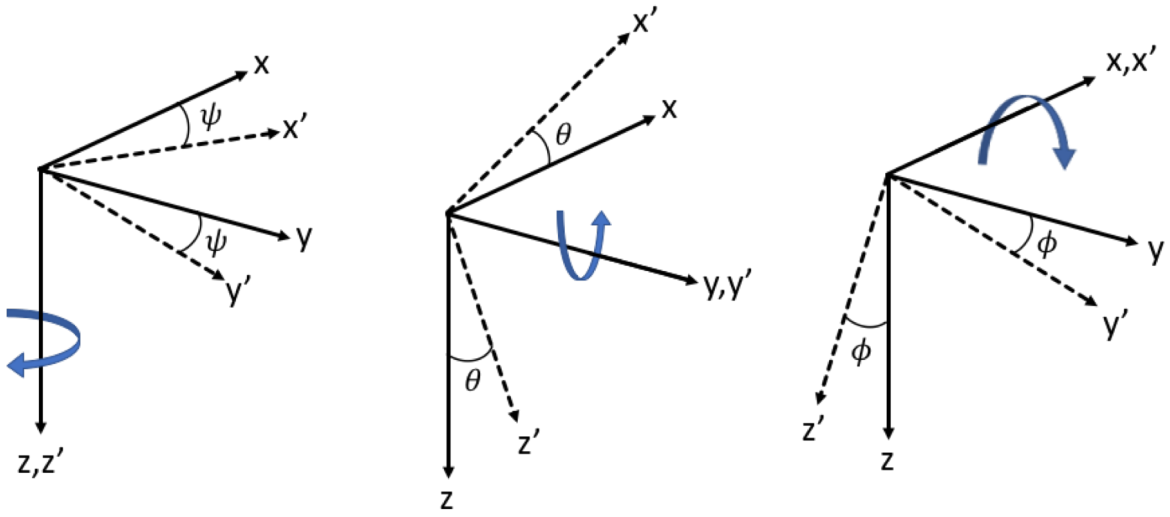

Figure 2.5: Euler Angle Rotations

2.3.1 Transformation from the Local Navigation Frame to the Body Frame

The rotation from the local navigation frame to the body frame is the multiplication of the three rotation matrices ($\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$) as shown in Equations (2.4-2.5).

$$\mathbf{C}_n^b = \mathbf{C}_3\mathbf{C}_2\mathbf{C}_1 \tag{2.4}$$

$$\begin{aligned}\mathbf{C}_n^b &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}\end{aligned} \tag{2.5}$$

The term $\mathbf{C}_n^b$ is the CTM that represents the inverse of the attitude matrix. The subscript indicates the "from" frame and the superscript indicates the "to" frame. Note that the order of the rotation matrices is critical and they must be applied in the correct order as the Euler angle matrices are not commutative. An example transformation of an arbitrary (3×1) vector $\mathbf{x}$ from the local navigation frame to the body frame is shown in Equation (2.6).

$$\mathbf{x}^b = \mathbf{C}_n^b\mathbf{x}^n \tag{2.6}$$

2.3.2 Transformation from the Body Frame to the Local Navigation Frame

In similar fashion, the inverse transformation from the body to the local navigation frame is given by in Equations (2.7-2.8).

$$\mathbf{C}_b^n = \mathbf{C}_n^{bT} = \mathbf{C}_1^T\mathbf{C}_2^T\mathbf{C}_3^T \tag{2.7}$$

$$
\begin{aligned}
\mathbf{C}_b^n &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}
\end{aligned}
\tag{2.8}
$$

The term $\mathbf{C}_b^n$ is the CTM that represents attitude. It is equivalent to the transpose of the CTM representing the rotation from the local navigation frame to the body frame $\left(\mathbf{C}_n^{bT}\right)$. The superscript T, represents the matrix transpose. An example transformation of an arbitrary (3×1) vector $\mathbf{x}$ from the body frame to the local navigation frame is shown in Equation (2.9).

$$
\mathbf{x}^n = \mathbf{C}_b^n \mathbf{x}^b \tag{2.9}
$$

2.4 Relationship between Euler Angles and Coordinate Transform Matrices

2.4.1 Local Navigation Frame to Body Frame CTM

The Euler angles can be computed from the rotation matrix $\mathbf{C}_n^b$ as follows

$$
\phi = \arctan2(\mathbf{C}_n^b(2,3), \mathbf{C}_n^b(3,3)) \tag{2.10}
$$

$$
\theta = -\arcsin(\mathbf{C}_n^b(1,3)) \tag{2.11}
$$

$$
\psi = \arctan2(\mathbf{C}_n^b(1,2), \mathbf{C}_n^b(1,1)) \tag{2.12}
$$

where the term $\mathbf{C}_n^b$(row, column) is read as the row and column entry of the CTM. The angles $\{\phi, \theta, \psi\}$ are again the roll, pitch, and yaw, respectively, and term $\arctan2$ is the four-quadrant inverse tangent and $\arcsin$ is the inverse sine.

2.4.2 Body Frame to Local Navigation Frame CTM

Similarly the Euler angles can be computed from the attitude representation ($\mathbf{C}_b^n$) in Equations (2.13-2.15).

$$\phi = \arctan2(\mathbf{C}_b^n(3,2), \mathbf{C}_b^n(3,3)) \tag{2.13}$$

$$\theta = -\arcsin(\mathbf{C}_b^n(3,1)) \tag{2.14}$$

$$\psi = \arctan2(\mathbf{C}_b^n(2,1), \mathbf{C}_b^n(1,1)) \tag{2.15}$$

2.4.3 Euler Angle Computation Issues

When θ approaches $\pm 90°$, Equations (2.10), (2.12), (2.13), and (2.15) become indeterminate as the numerator and the denominator approach zero simultaneously [21]. If this problem occurs alternate representations of attitude should be used, such as quaternion parameters or direction cosines. Note that this work used direction cosines for its attitude representation which is further discussed in Section 3.3.7. For further discussion of attitude representations, readers are referred to [3, 21, 41].

Chapter 3

Pedestrian Dead Reckoning: An INS Approach

3.1 Introduction

Dead reckoning (DR) is traditionally defined as integrating a measurement of velocity or measuring the change in position and adding it to the previous position to compute ones current position [3]. This technique has been used since as early as 3000 BC to navigate polynesian vessels and was also used to measure the distance traveled by Roman chariots around 23 BC [1, 3]. DR is still used today, albeit now with computers and more advanced sensors. As stated above, DR systems estimate the *change in positon* of the user and the new position is found by adding the change in position to the previously known position. DR systems use sensors mounted on the user to estimate the *relative position* and not the absolute position. Therefore, the initial position of the user must be known to initialize a DR system. DR is an alternative way to provide positioning information when external references are unavailable.

Inertial based pedestrian-dead-reckoning (PDR) systems are a topic of huge interest, as MEMS sensors are now sufficiently small and accurate to enable the practical tracking of pedestrians. PDR systems in their most general form can be broken into three steps:

1. Gait Monitoring.
2. Estimation of change in position and heading of the user.
3. Navigation system update.

Note that the term PDR is ambiguously used by authors [4]. Some authors use PDR to describe a system that performs step detection and estimates step length and heading of the pedestrian [3, 22]. Others use the term PDR to describe a foot-mounted inertial navigation approach with zero-velocity-updates (ZVUs), which are also called 'ZUPTs' by some authors [23,24]. A foot-mounted INS estimates the full 3D trajectory of the sensor at all moments in time. Therefore, in

order to differentiate the two PDR methods, the foot or shoe mounted approach will be referred to as PDR-INS while the step detection (SD) method will be called PDR-SD.

In the next section, the gait cycle and the terms step and stride are formally defined. In the following sections, the PDR-INS methods will be discussed. PDR-SD methods will be discussed in the next chapter.

3.2 Pedestrian Gait Cycle

Gait cycle monitoring is highly dependent upon the assumption that a pedestrian will exhibit ambulatory motion. Gait is the pedestrian's manner of ambulation or locomotion, while ambulation is the alternating pushing up of the body over each stiffened leg each time a pedestrian steps. The gait cycle can be broken into two phases, namely the stance phase and the swing phase. The *stance phase* is defined as the period from initial contact of the foot (i.e. heel strike) with the floor to the toe-off (i.e. lift off or peel off of foot). More intuitively, the stance phase can be thought of as the period that the foot is in contact with the ground. This can occur while walking or standing stationary. These two phases are illustrated in Figure 3.1 [25, 26]. The *swing phase* is the period from the toe-off to the heel strike of the same foot. The swing phase intuitively is the period in which the foot is 'swinging' in the air and is not in contact with the ground. The transition point from swing phase to stance phase is referred to as *footfall* by some authors [4, 24].

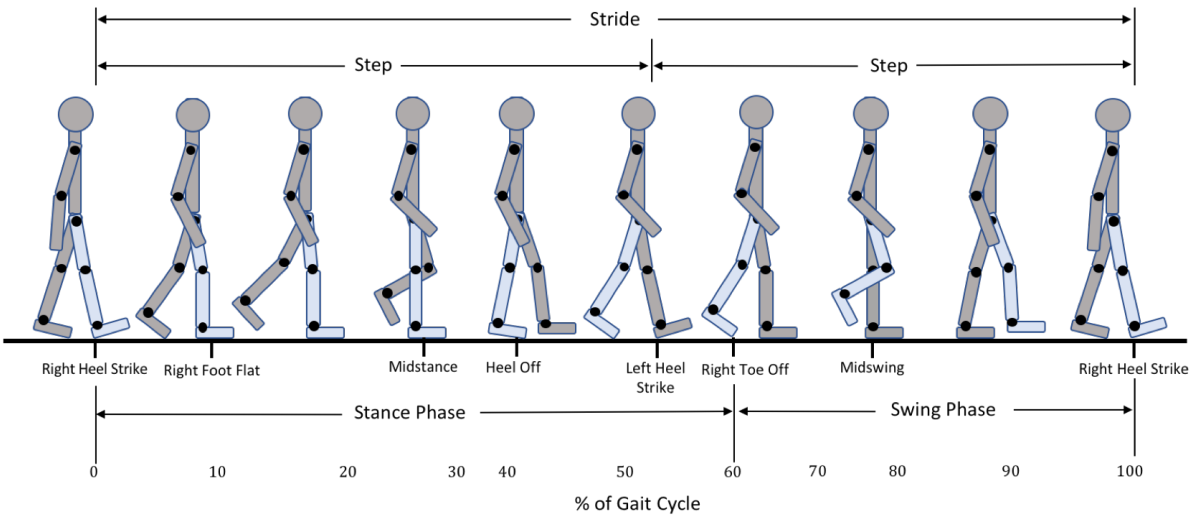

Figure 3.1: Walking Gait Cycle [25]

What is referred to as a step in publications is often ambiguous and not well defined by authors within the pedestrian navigation community. A *step* is defined as the period between heel strikes of the opposite foot [4,25,27]. This is not to be confused with a *stride*, which is the period between footfalls of the same foot. Steps and strides are illustrated in both Figure 3.1 and 3.2.

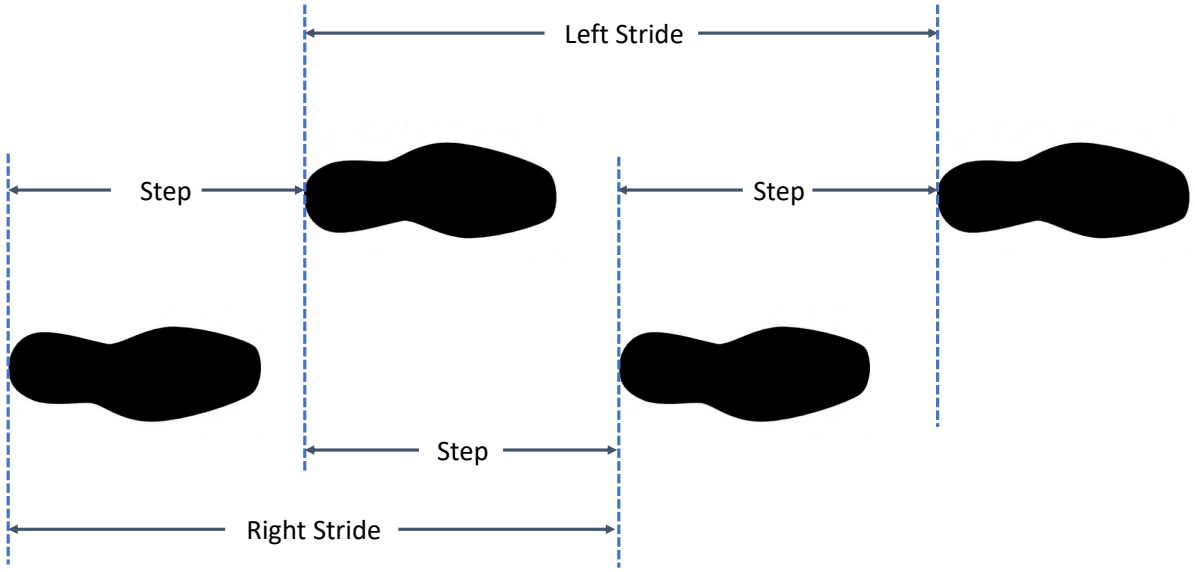

Figure 3.2: Step Versus Stride [25, 27]

3.3 PDR-INS Systems

This section describes performing pedestrian dead reckoning (PDR) with a PDR-INS approach. This approach is popular, as it takes advantage of the IMU location on the pedestrian's foot. An INS mechanizes a foot-mounted IMU by the double and single integration of the IMU accelerations and rotation rates, respectively. This produces a 6 degree-of-freedom (DoF) navigation solution of the device that includes position, velocity, and attitude [3]. An error-state Kalman Filter (ESKF) is then used to intelligently combine the inertial navigation system (INS) with pseudo measurements in the form of zero-velocity updates (ZVUs) and zero-angular-rate updates (ZARUs). This is done to estimate and correct the errors in the localization solution. Other pseudo measurements can also be created using the assumption of straight walking and known directions of building corridors. The first method is presented by Jiménez et al., which is based upon work by Borenstein et al. and is termed heuristic drift reduction (HDR) [23, 28].

It uses the assumption of straight walking to create a pseudo measurement of heading error to reduce the heading drift. The second method uses the known building corridor directions (i.e. cardinal directions) to create a heading measurement. This technique created by Abdulrahim et. al. is termed cardinal heading aided for inertial navigation (CHAIN) [29, 30]. These measurement updates allow for the estimation of internal IMU errors (e.g. sensor biases). The INS states are then corrected, using the estimated errors which significantly reduces the error drift of the navigation solution. In order to apply the ZVU, ZARU, HDR, and CHAIN measurement updates, the zero-velocity, still, and straight-walking points of the pedestrian gait cycle must first be detected. The remainder of this chapter is organized as follows. The next section will introduce the IMU error model used, and the next section introduces and quantitatively describes what zero-velocity, still, and straight-walking points are and how they are detected. Lastly, the next sections will introduce a brief overview of the Kalman filter and error-state Kalman filter framework used in this thesis.

3.3.1 Inertial Measurement Unit Error Model

Inertial measurement units typically contain accelerometer and gyro triads, which measure the specific force and angular rates of the unit, respectively. A triad has three axes mounted orthogonally to one another. The IMU measurements can be modeled as

$$\tilde{\mathbf{f}}_{ib}^{b} = (\mathbf{I}_3 + \mathbf{M}_a)\mathbf{f}_{ib}^{b} + \mathbf{b}_a + \mathbf{w}_a \tag{3.1}$$

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = (\mathbf{I}_3 + \mathbf{M}_g)\boldsymbol{\omega}_{ib}^{b} + \mathbf{G}_g\mathbf{f}_{ib}^{b} + \mathbf{b}_g + \mathbf{w}_g \tag{3.2}$$

where $\{\tilde{\mathbf{f}}_{ib}^{b}, \tilde{\boldsymbol{\omega}}_{ib}^{b}\}$ are the measured specific force and angular rate, and $\{\mathbf{f}_{ib}^{b}, \boldsymbol{\omega}_{ib}^{b}\}$ are the true values of specific force and angular rate, respectively [3]. The terms $\{\mathbf{M}_a, \mathbf{M}_g\}$ are (3×3) matrices that represent misalignment between axes, and $\mathbf{I}_3$ is a (3×3) identity matrix. The term $\mathbf{G}_g$ accounts for the gyro sensitivity (g-sensitivity) to specific force and $\{\mathbf{w}_a, \mathbf{w}_g\}$ are the white noise of the accelerometers and gyros, respectively. These models are often simplified for MEMS IMUs to be

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{\omega}_{ib}^{b} + \mathbf{b}_g + \mathbf{w}_g \tag{3.3}$$

$$\tilde{\mathbf{f}}_{ib}^{b} = \mathbf{f}_{ib}^{b} + \mathbf{b}_a + \mathbf{w}_a \tag{3.4}$$

where both the misalignment and the g-sensitivity errors have been neglected, as they have been removed by the manufacturer through calibration.

The accelerometer and the gyro biases are not static in time. This work models them as the following random walk processes

$$\dot{\mathbf{b}}_a = \mathbf{w}_{\mathbf{b}_a} \tag{3.5}$$

$$\dot{\mathbf{b}}_g = \mathbf{w}_{\mathbf{b}_g} \tag{3.6}$$

where $\{\mathbf{w}_{\mathbf{b}_a}, \mathbf{w}_{\mathbf{b}_g}\}$ are the driving noise of the accelerometer bias and gyro bias, respectively.

3.3.2 Gait Monitoring

Gait Monitoring has three parts, the first part is determining the zero-velocity points within the stance phase. The second part is determining the still periods that occur when the pedestrian is stationary in the stance phase. The third part is determining when the pedestrian's strides occur. The three parts will be discussed in the order that they were introduced.

When the foot is planted on the ground during the stance phase, the velocity of the foot is approximately zero. The foot is never detected to have exactly zero velocity, due to sensor errors and the fact that the foot is never perfectly still. This makes stance detection a nontrivial problem. Zero-velocity detection can be performed using different sensors with the most popular choices being either the magnitude of accelerometer or gyro signals [23, 24, 31]. This thesis used gyros, as they provide better performance by being less noisy. The magnitude of the gyros is calculated by finding the Euclidean norm of the measurements which is

$$\left\| \tilde{\boldsymbol{\omega}}_{ib}^{b} \right\| = \sqrt{(\tilde{\omega}_{ib,x}^{b})^2 + (\tilde{\omega}_{ib,y}^{b})^2 + (\tilde{\omega}_{ib,z}^{b})^2} \tag{3.7}$$

where $\{\tilde{\omega}_{ib,x}^{b}, \tilde{\omega}_{ib,y}^{b}, \tilde{\omega}_{ib,z}^{b}\}$ are the angular rate measurements of the x, y, and z axes, respectively. The stance and swing phase along with strides are identified qualitatively using the gyro magnitude in Figure 3.3.

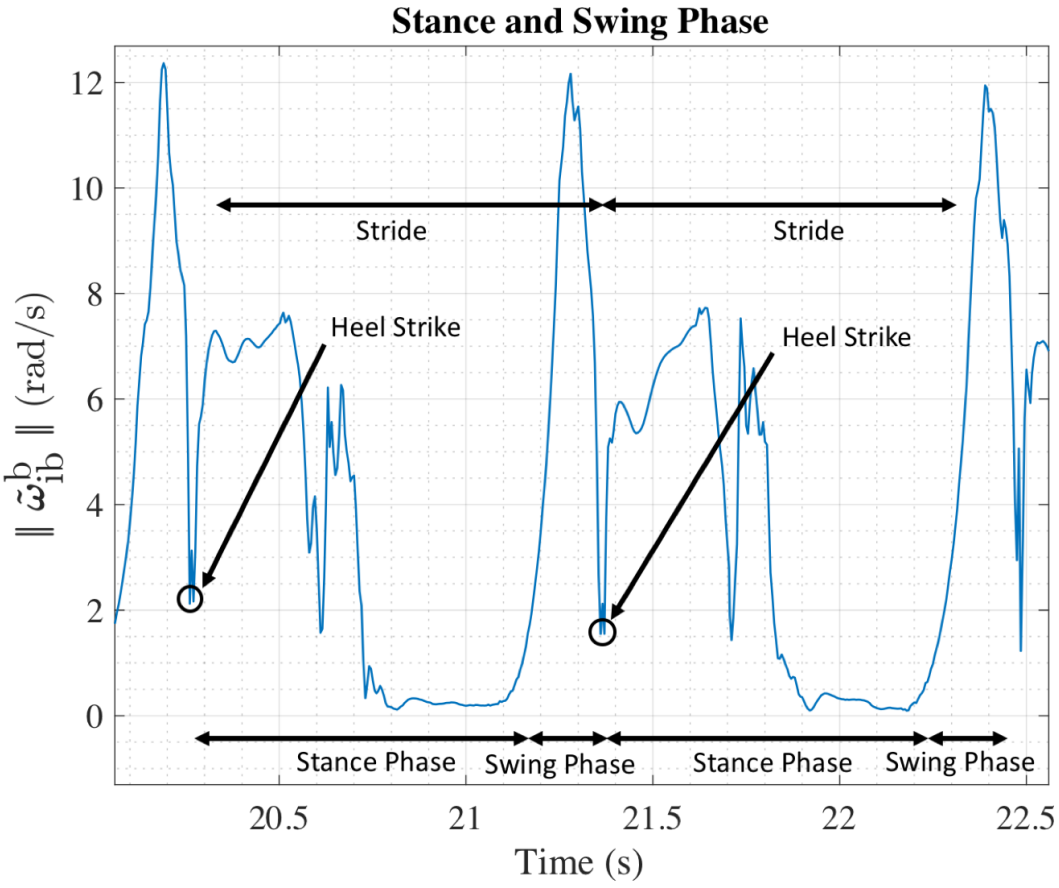

Figure 3.3: Gyro Output during Stance and Swing Phase

The stance phase begins with the heel strike which causes a short period of oscillations. Next, the foot can be seen to be still for a period of approximately 0.5 seconds. The swing phase follows this and then the heel strike begins the next stance phase. The technique for detecting the zero-velocity points is a thresholding method. This is based upon methods developed by Pierce [32] and Jiménez et al. [23]. The two conditions that must be true for the foot to be at zero-velocity are:

1. The norm (i.e. magnitude) of the gyros must be below a certain threshold ($TH_{zv} = 0.7$ (rad/s)). Note that some other methods use a maximum and minimum threshold, only a maximum threshold is needed as the norm is always positive. The first condition for zero-velocity detection written mathematically is

$$C1 = \begin{cases} 1, & \sim \text{any}\left(\left\| \tilde{\boldsymbol{\omega}}^{b}_{ib,window} \right\| > TH_{zv} \right) \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

where the term $\sim$ is a logical NOT and the function any() determines if any of the vector elements are nonzero. The conditional statement can be more intuitively thought of as, if none of the gyro norm measurements in the window are above the threshold than the conditional statement is true. A window length of 21 samples was used, with a sample rate of 100 Hz.

2. The local standard deviation of the signal is also taken into consideration. The second condition for zero-velocity detection written mathematically is

$$C2 = \begin{cases} 1, & \left(\left\|\tilde{\boldsymbol{\omega}}_{ib,k}^{b}\right\| - \mu_{window}\right) < 1.5 \cdot \sigma_{window} \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

where $\left\|\boldsymbol{\omega}_{ib,k}^{b}\right\|$, is the gyro magnitude at the current step in time (t_k) and μ_{window} is the mean of the current window. The term σ_{window} is the standard deviation of the current window. A window length of 21 samples was also used, with a sample rate of 100 Hz. The result of these conditional statements is shown in a plot in Figure 3.4.

A *still* is defined as when the user is standing still and not moving. Still detection has the added constraint that the pedestrian must also be stationary. The foot of the pedestrian is never measured to be perfectly stationary due to the pedestrian never being perfectly still and the errors in the sensor measurements. The two conditions that must be true for the foot to be still are:

1. The norm (i.e. magnitude) of the gyros must be below a certain threshold ($TH_{still} = 0.2$ (rad/s)). Notice that this is a much lower threshold than the zero-velocity threshold. The first condition for still detection written mathematically is

$$C3 = \begin{cases} 1, & \sim \text{any}\left(\left\|\tilde{\boldsymbol{\omega}}_{ib,window}^{b}\right\| > TH_{still} \right) \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

where the term $\sim$ is a logical NOT and the function any() determines if any of the vector elements are nonzero. The conditional statement can be more intuitively thought of as, if none of the gyro norm measurements in the window are above the threshold than the

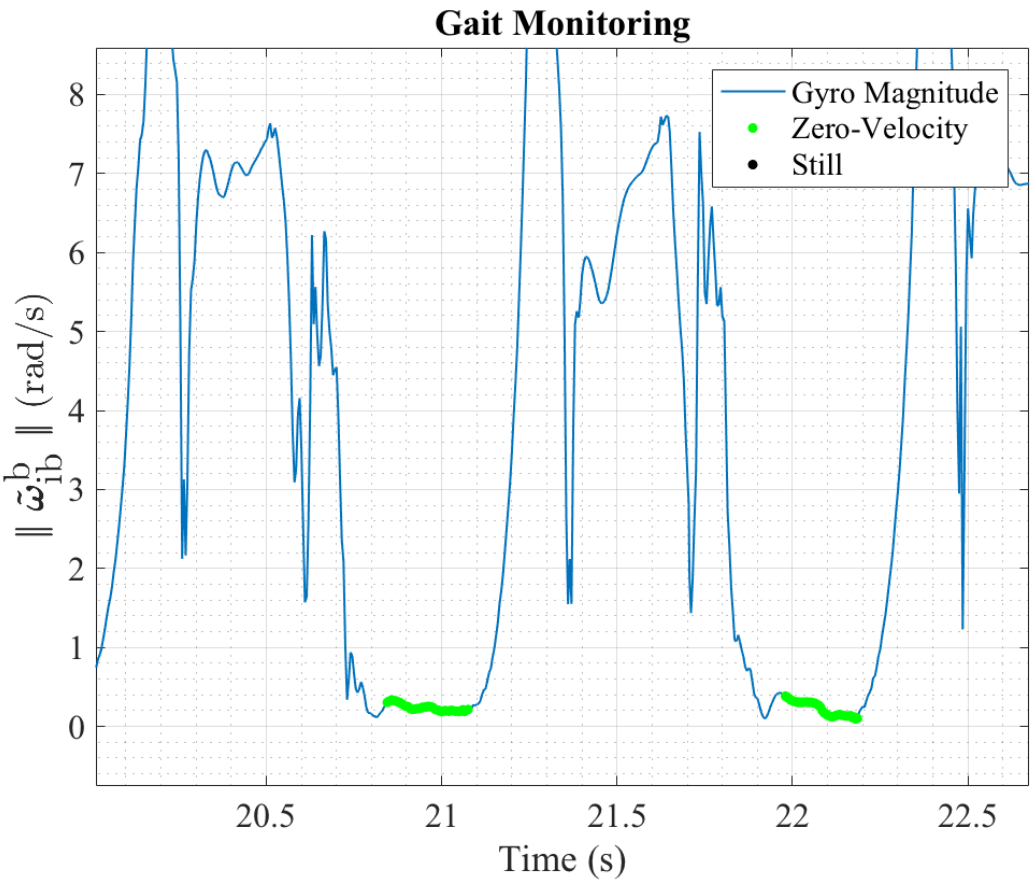

Figure 3.4: Zero-Velocity Detection

conditional statement is true. A window length of 21 samples was used, with a sample rate of 100 Hz.

2. The local standard deviation of the signal is also taken into consideration. The second condition for still detection written mathematically is

$$C4 = \begin{cases} 1, & \left(\left\| \tilde{\boldsymbol{\omega}}_{ib,k}^{b} \right\| - \mu_{window} \right) < 2 \cdot \sigma_{window} \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

where $\left\| \boldsymbol{\omega}_{ib,k}^{b} \right\|$, is the gyro magnitude at the current step in time (t_k) and μ_{window} is the mean of the current window. The term σ_{window} is the standard deviation of the current window. A window length of 21 samples was also used. The result of these conditional statements is shown in a plot in Figure 3.5.

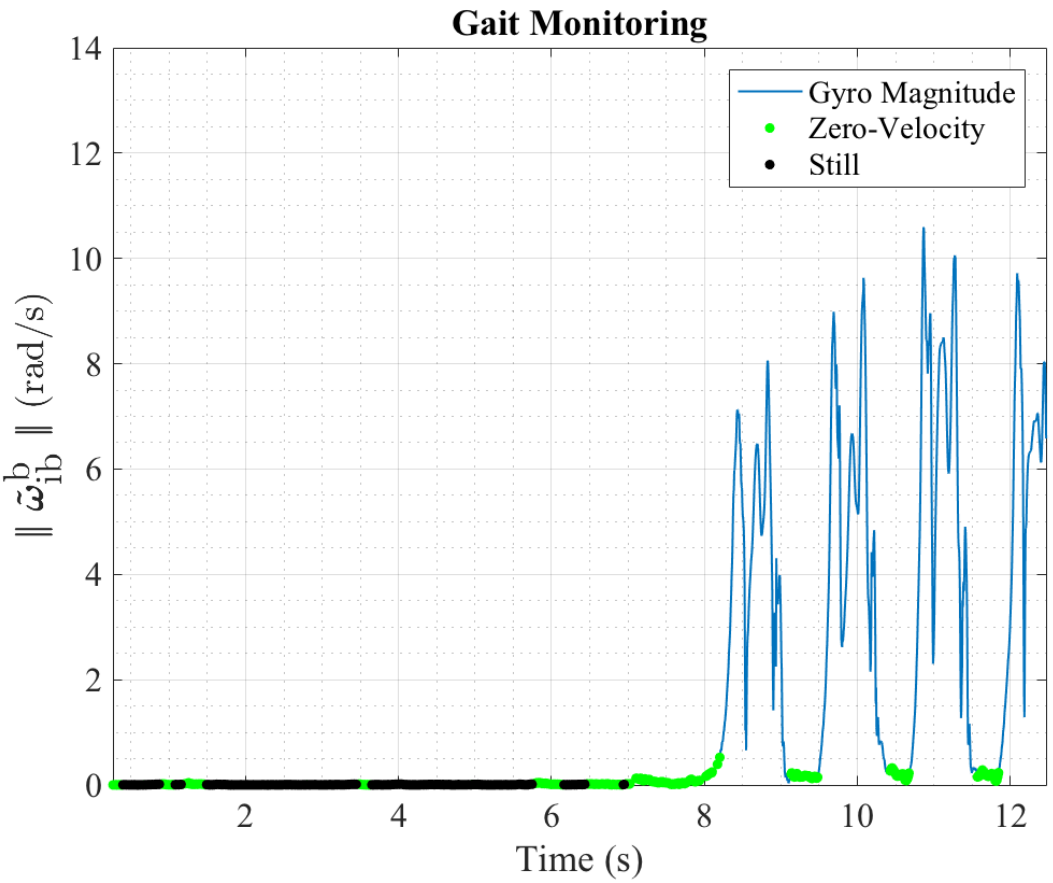

Figure 3.5: Still Detection

A new stride formally begins when footfall of the same foot occurs. As it occurs there is a period of high oscillations as the foot experiences a high amount of acceleration. Because of this, the point at which a stride begins is difficult to detect with a foot-mounted IMU, thus this work makes the approximation of saying a stride begins at the first moment that the foot is detected to be at zero velocity. The three conditions that must be true for there to be a stride detected are:

1. For there to be a stride at the moment in time the foot must be at zero velocity, thus conditions $\{C1, C2\}$ must both be true. The first condition for stride detection written mathematically is

$$C5 = \begin{cases} 1, & C1 \ \& \ C2 \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

where the term $\&$ is a logical AND operator.

2. The time between strides must be above a certain threshold (t_{min}). Therefore, the second condition for stride detection written mathematically is

$$C6 = \begin{cases} 1, & t_s > t_{min} \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

where t_s is the time of the current position minus the time at the epoch [1] when a previous stride was detected. The minimum time between strides was defined as $t_{min} = 0.5\ (s)$.

3. The stride length must be above a certain threshold ($stride_{min}$) or the change in heading above a certain threshold ($\Delta\psi_{min}$). This results in the following condition for stride detection as

$$C7 = \begin{cases} 1, & stride > stride_{min} \ || \ \Delta\psi_s > \Delta\psi_{min} \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

where the term $||$ is a logical OR operator and $stride$ is the distance between the current position and the epoch where a previous stide was detected. The term $\Delta\psi_s$ is the change in heading between the current heading and the heading where a previous stride was detected. The result of these conditional statements is shown in a plot in Figure 3.6.

[1] An epoch is defined as single moment in time (t_k).

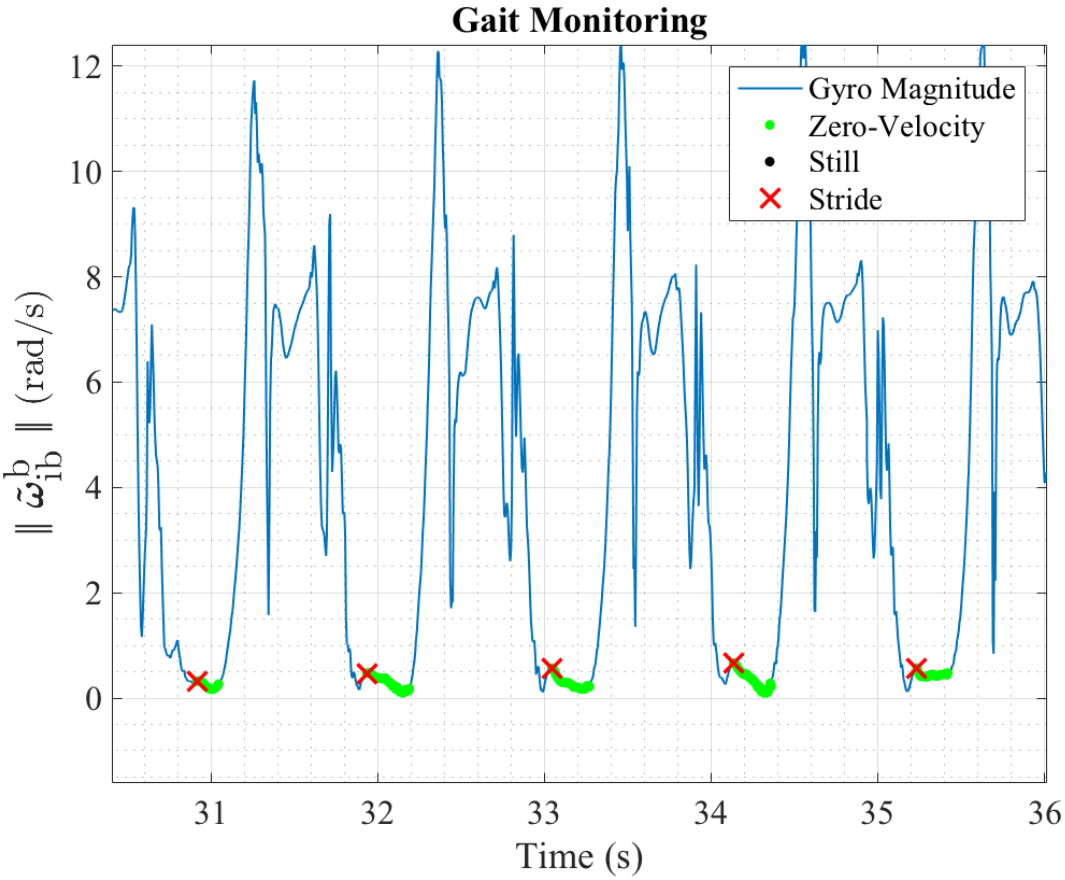

Figure 3.6: Stride Detection

3.3.3 Straight Walking Detection

Buildings often have corridors or paths that are straight. When a pedestrian walks inside a building they often follow these straight line paths in a predictable manner. Knowledge of this behavior can be taken advantage of and used as a measurement to help reduce the heading drift. This section will highlight how to detect when a pedestrian is walking straight. The method presented here is based upon a method presented by Jiménez et al. [23]. The direction of the pedestrian is quantified by computing

$$\Delta\psi = \psi_k - (\psi_{ks} + \psi_{ks-1})/2 \tag{3.15}$$

where ψ_k is the current heading, ψ_{ks} is the heading at the previous stride detected, and ψ_{ks-1} is the heading at stride detected one previous to the one at t_{ks}. The term $\Delta\psi$ must be below a certain threshold ($TH_{\Delta\psi}$) for the pedestrian to be walking staight. This condition for straight walking is

$$C8 = \begin{cases} 1, & \Delta\psi < TH_{\Delta\psi} \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

where $TH_{\Psi} = 10^{\circ}$.

3.3.4 The Discrete-Time Linear Kalman Filter

The Kalman filter (KF) was first introduced in the 1960's by Rudolf E. Kálmán in his landmark publication [33]. Its optimization criterion is the minimization of the mean-square error. The KF and its variants are widely used in many fields today, some of which include navigation systems, economic time series analysis, and weather forecasting. The application of this thesis is estimating INS errors. As practical applications are implemented using a digital computer, the discrete form of the linear Kalman filter is presented. The system model of a discrete system with only stochastic inputs is

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{d,k-1}, \quad \mathbf{w}_{d,k-1} \sim N\big(\mathbf{0}, \mathbf{Q}_{d,k-1}\big) \tag{3.17}$$

where $\mathbf{x}_k$ is the $(n \times 1)$ state vector at time t_k and $\mathbf{\Phi}_{k-1}$ is the $(n \times n)$ state transition matrix at time t_{k-1}. The term $\mathbf{w}_{d,k-1}$ is the $(n \times 1)$ discrete process noise vector at time t_{k-1}. The process noise $(\mathbf{w}_{k-1})$ is zero mean Gaussian noise with a covariance matrix of $\mathbf{Q}_{d,k-1}$. The discrete process noise covariance matrix $(\mathbf{Q}_{d,k-1})$ is defined as

$$\mathbf{Q}_{d,k-1} = E[\mathbf{w}_{d,k-1}\mathbf{w}_{d,k-1}^T] \tag{3.18}$$

The discrete-time linear measurement (observation) model is

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{3.19}$$

where $\mathbf{y}_k$ is the $(m \times 1)$ measurement vector at time t_k. The term $\mathbf{H}_k$ is the $(m \times n)$ measurement matrix, that models the measurements as a linear function of the true state vector at time t_k. Lastly, $\mathbf{v}_k$ is the $(m \times 1)$ measurement noise vector at time t_k. The measurement noise $\mathbf{v}_k$ is modeled as zero mean Gaussian noise with a covariance matrix of $\mathbf{R}_k$. The measurement noise covariance matrix $\mathbf{R}_k$ is a $(m \times m)$ matrix defined by Equation (3.20).

$$\mathbf{R}_k = E[\mathbf{v}_k\mathbf{v}_k^T] \tag{3.20}$$

The Kalman filter has three distinct steps: initialization, time update, and the measurement update. The KF is first initialized with an initial state and error covariance matrix. The time update uses the system model to produce a state estimate of the current state using the previous state estimate. Next, the state estimate is corrected using the measurement update, which optimally weights the measurement and the current state estimate using the Kalman gain. The Kalman filter is recursive, therefore the time and measurement update steps theoretically can repeat forever. Below in Table 3.1, the equations for the discrete-time linear KF are listed. A full derivation and further explanation of the Kalman filter and its applications are available in many texts [3, 34–36].

Table 3.1: Discrete-Time Linear Kalman Filter

Initialization	$\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_{t_0}$ $\mathbf{P}_0 = E[\hat{\mathbf{x}}_0 \hat{\mathbf{x}}_0^T]$
Time Update	$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1} \hat{\mathbf{x}}_{k-1}$ $\mathbf{P}_k^- = \mathbf{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{d,k-1}$
Measurement Update	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$ $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$

3.3.5 Extended Kalman Filter

A Kalman filter requires that the system and the measurement models both be linear. An extended Kalman filter (EKF) allows for the use of nonlinear system and measurement models. Consider the following nonlinear system and measurement models

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{w}(t)\big) \tag{3.21}$$

$$\dot{\mathbf{y}}(t) = \mathbf{h}\big(\mathbf{x}(t), \mathbf{v}(t)\big) \tag{3.22}$$

where $\mathbf{f}\big(\mathbf{x}(t), \mathbf{w}(t)\big)$ and $\mathbf{h}\big(\mathbf{x}(t), \mathbf{v}(t)\big)$are both assumed to be continuously differentiable functions. If the system and measurement models are nonlinear, the extended Kalman filter linearizes the nonlinear functions about the current state estimate. This is done by performing a Taylor series expansion and truncating to the first term. This first term is a matrix of partial derivatives as shown in Equations (3.23) and (3.24) is referred to as the Jacobian.

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.23}$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.24}$$

The models are linearized in order to calculate the covariance matrix (**P**) and the Kalman gain (**K**) which allows the measurement update to be performed. Note that linearizations need to be accurate or else the EKF will suffer from stability problems [3, 19, 34].

3.3.6 Error-State Kalman Filter

The method presented in this thesis has been shown by Pierce et al. in [37]. It is based upon the INS-EKF-ZVU (IEZ) framework developed by Jiménez et al. [23] and work by Foxlin [38]. A similar framework is presented by Rajagopal [24]. The approach uses a discrete-time error-state Kalman filter (ESKF). An error-state or indirect formulation is one in which the system model and measurement model are a function of error states and not the actual states themselves. Estimators that estimate the actual state (i.e. position, velocity, and attitude) are known as total

state or direct formulations. If a total state approach had been used, an EKF would be needed as the system model is nonlinear [3, 32]. For the ESKF, error is defined as $\delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$, where $\hat{\mathbf{x}}$ is the estimated state and $\mathbf{x}$ is the true state. The following sections will introduce the system model and the measurement model for the ESKF.

I. System Model

The continuous-time inertial navigation error equations for navigating in the local navigation frame are

$$\delta\dot{\mathbf{r}} = \delta\mathbf{v} \tag{3.25}$$

$$\delta\dot{\mathbf{v}} = -\mathbf{S}(\mathbf{a}_n)\delta\boldsymbol{\varphi} + \mathbf{C}_b^n(\mathbf{b}_a + \mathbf{w}_a) \tag{3.26}$$

$$\delta\dot{\boldsymbol{\varphi}} = \mathbf{C}_b^n(\mathbf{b}_g + \mathbf{w}_g) \tag{3.27}$$

where $\delta\mathbf{r}$ is the position error vector, $\delta\mathbf{v}$ is the velocity error vector, and $\delta\boldsymbol{\varphi}$ is the attitude error vector. The term $\mathbf{S}(\mathbf{a}_n)$ is the (3×3) skew symmetric matrix for accelerations in the navigation frame. It is defined as

$$\mathbf{S}(\mathbf{a}_n) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{3.28}$$

where $\mathbf{a}_n$ is the bias corrected acceleration that has been transformed into the navigation frame. The terms $\{\mathbf{w}_a, \mathbf{w}_g\}$ are the white noise terms of the accelerometers and gyros, respectively. Lastly, the term $\mathbf{C}_b^n$ is the CTM from the body to the navigation frame. Note that the Coriolis and gravity error terms have been neglected from the navigation error equations. The Coriolis error term can be neglected as it is much smaller than the other terms and the gravitation error term is neglected as the application is for local applications only in this thesis. A simple gravity error model would require knowledge of the pedestrian's global position. A thorough derivation of these error equations can be found in [21,39]. The system model is augmented to also include the accelerometer and gyro bias states. Section 3.3.1 modeled the biases as

$$\dot{\mathbf{b}}_a = \mathbf{w}_{\mathbf{b}_a} \tag{3.29}$$

$$\dot{\mathbf{b}}_g = \mathbf{w}_{\mathbf{b}_g} \tag{3.30}$$

where both the biases are modeled as random walk processes and $\{\mathbf{w}_{\mathbf{b}_a}, \mathbf{w}_{\mathbf{b}_g}\}$ are the driving noise. Notice that Equations (3.25-3.27, 3.29-3.30) are always linear, so they do not need to be linearlized and can be put directly into state-space form as

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{3.31}$$

where $\mathbf{F}$ is the state matrix, $\delta\mathbf{x}$ is the error state vector, $\mathbf{G}$ is the noise input matrix, and $\mathbf{w}$ is the continuous process noise.

The state vector is made up of 15 states that include 9 navigation error states and 6 bias states. The state vector is

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{r} & \delta\mathbf{v} & \delta\boldsymbol{\varphi} & \mathbf{b}_a & \mathbf{b}_g\end{bmatrix}^T \tag{3.32}$$

where navigation error states are the (3×1) position error vector ($\delta\mathbf{r}$), (3×1) velocity error vector ($\delta\mathbf{v}$), and the (3×1) attitude error vector ($\delta\boldsymbol{\varphi}$). The state vector has been augmented to include the (3×1) accelerometer bias vector ($\mathbf{b}_a$) and the (3×1) gyro bias vector ($\mathbf{b}_g$). The components of each of the navigation states and bias states can be seen in Equations (3.33) and (3.34), respectively.

$$\delta\mathbf{r} = \begin{bmatrix}\delta N \\ \delta E \\ \delta D\end{bmatrix} \quad \delta\mathbf{v} = \begin{bmatrix}\delta\dot{N} \\ \delta\dot{E} \\ \delta\dot{D}\end{bmatrix} \quad \delta\boldsymbol{\varphi} = \begin{bmatrix}\delta\phi \\ \delta\theta \\ \delta\psi\end{bmatrix} \tag{3.33}$$

$$\mathbf{b}_a = \begin{bmatrix}b_{a,x} \\ b_{a,y} \\ b_{a,z}\end{bmatrix} \quad \mathbf{b}_g = \begin{bmatrix}b_{g,x} \\ b_{g,y} \\ b_{g,z}\end{bmatrix} \tag{3.34}$$

The terms $\{\delta N, \delta E, \delta D\}$ are the position errors and $\{\delta\dot{N}, \delta\dot{E}, \delta\dot{D}\}$ are the velocity errors of the North, East, and Down directions, respectively. Following this, the terms $\{\delta\phi, \delta\theta, \delta\psi\}$ are the roll, pitch, and yaw errors, respectively. Lastly, the terms $\{b_{a,x}, b_{a,y}, b_{a,z}\}$ are the accelerometer biases and $\{b_{g,x}, b_{g,y}, b_{g,z}\}$ are the gyro biases of the x, y, and z axes, respectively. The state

matrix $\mathbf{F}$ is a (15×15) matrix defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{S}(\mathbf{a}_n) & \mathbf{C}_b^n & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{C}_b^n \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{3.35}$$

where $\mathbf{0}_3$ is a (3×3) zero matrix and $\mathbf{I}_3$ is a (3×3) identity matrix. The matrix $\mathbf{G}$ is a (15×12) continuous process noise input matrix, it is defined as

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{C}_b^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{C}_b^n & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.36}$$

The vector $\mathbf{w}$ is the (12×1) continuous process (system) noise vector and is defined as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_a & \mathbf{w}_g & \mathbf{w}_{b_a} & \mathbf{w}_{b_g} \end{bmatrix}^T \tag{3.37}$$

where $\mathbf{w}_a$ denotes the accelerometer noise and $\mathbf{w}_g$ the gyro noise. The terms $\{\mathbf{w}_{\mathbf{b}_a}, \mathbf{w}_{\mathbf{b}_g}\}$ are the accelerometer and gyro bias noise, respectively. The continuous process noise covariance is a (12×12) matrix defined as

$$\mathbf{Q} = \begin{bmatrix} \sigma_a^2 \cdot \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_g^2 \cdot \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \sigma_{\mathbf{b}_a}^2 \cdot \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \sigma_{\mathbf{b}_g}^2 \cdot \mathbf{I}_3 \end{bmatrix} \tag{3.38}$$

where $\{\sigma_a^2, \sigma_g^2\}$ are the variance of noise of the acceleration and gyro measurements, respectively. The terms $\{\sigma_{\mathbf{b}_a}^2, \sigma_{\mathbf{b}_g}^2\}$ are the variance of the accelerometer and gyro biases, respectively. The term $\mathbf{I}_3$ is a (3×3) identity matrix. Further discussion on how to find the sensor and bias variances can be found in [3, 40, 41]

In order to implement this KF in the discrete form, the system model must be discretized. The linear error equations in the discrete form are

$$\delta\mathbf{x}_k = \mathbf{\Phi}_{k-1}\delta\mathbf{x}_{k-1} + \mathbf{w}_{d,k-1} \tag{3.39}$$

where $\mathbf{\Phi}_{k-1}$ is the state transition matrix at time t_{k-1} and $\mathbf{w}_{d,k-1}$ is the discrete system noise. The state transition matrix is defined as being the unique solution to Equation (3.40) [42].

$$\dot{\mathbf{\Phi}} = \mathbf{F}\mathbf{\Phi}. \tag{3.40}$$

If the state transition matrix is assumed to be time invariant, the solution to the differential equation is found to be the matrix exponential of the system matrix as shown in Equation (3.41).

$$\mathbf{\Phi} = exp(\mathbf{F}\Delta t). \tag{3.41}$$

In this case, $\mathbf{F}$ is time varying as the acceleration term $-\mathbf{S}(\mathbf{a}_n)$ and the attitude represented as coordinate transform matrix $\mathbf{C}_b^n$ are updated each iteration in time. The matrix $\mathbf{F}$ can be approximated as being time invariant (i.e. a constant matrix) between each iteration in time if the sample rate of the IMU is high compared to the rate of change of $\mathbf{F}$. The matrix exponential can be approximated with a power series expansion truncated to the first term as

$$\mathbf{\Phi} \approx \mathbf{I}_{15} + \mathbf{F}\Delta t \tag{3.42}$$

where $\mathbf{I}_{15}$ is a (15×15) identity matrix and Δt is the time interval between samples. Therefore, the state transition matrix is

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & -\Delta t \cdot \mathbf{S}(\mathbf{a}_n) & \Delta t \cdot \mathbf{C}_b^n & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \Delta t \cdot \mathbf{C}_b^n \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.43}$$

The process noise covariance matrix is discretized with the equation

$$\mathbf{Q}_d = \mathbf{G}\mathbf{Q}\mathbf{G}^T \Delta t \tag{3.44}$$

where $\mathbf{G}$ is the continuous process noise input matrix and $\mathbf{Q}_d$ is the (15×15) discrete process noise matrix. The inertial error model equations presented in this thesis are linear, hence the discrete-time linear KF equations in Table 3.1 were utilized.

II. Measurement Model

The discrete-time linear measurement model of the ESKF is

$$\delta \mathbf{y}_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{3.45}$$

where $\delta \mathbf{y}_k$ is the error measurement, $\mathbf{H}_k$ is the measurement matrix, and $\mathbf{v}_k$ is the measurement noise at time t_k. Note the measurement model is linear and does not need to be linearized to be put into state-space form.

The ZVU, ZARU, HDR, and CHAIN updates are applied in the measurement update of the ESKF. When the foot is the swing phase of the gait cycle the measurement vector is zero. When zero-velocity is detected in the stance phase of the gait cycle, a ZVU can be applied. The

measurement vector when applying just a ZVU is

$$\delta \mathbf{y} = \begin{bmatrix} \delta \mathbf{v} \end{bmatrix} \tag{3.46}$$

where $\delta \mathbf{v}$ is the velocity error. It is defined as, $\delta \mathbf{v} = \hat{\mathbf{v}} - \mathbf{v}$, where $\hat{\mathbf{v}}$ is the estimated state and $\mathbf{v}$ is the true state. When the foot is detected to be at zero velocity, the velocity error is defined as $\delta \mathbf{v} = \hat{\mathbf{v}} - [0, 0, 0]^T$, where the value of the true state is a vector of zeros. Thus, the measurement is simplified to $\delta \mathbf{v} = \hat{\mathbf{v}}$. The measurement matrix is a (3×15) matrix for a ZVU and is

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{3.47}$$

where $\mathbf{0}_3$ is a (3×3) matrix of zeros and $\mathbf{I}_3$ is a (3×3) identity matrix. When the pedestrian's foot is still, it has zero velocity and is stationary allowing a ZVU and ZARU to be applied. Because the angular rates are not states, the application of the ZARU is not as straightforward. The gyro measurements that are modeled in Equation (3.3) are repeated here in Equation (3.48)

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{\omega}_{ib}^{b} + \mathbf{b}_g + \mathbf{w}_g \tag{3.48}$$

When a still is detected the foot is said to have a zero angular rate, therefore the true value of $\boldsymbol{\omega}_{ib}^{b}$ is equal to zero, $\boldsymbol{\omega}_{ib}^{b} = \mathbf{0}_{3\times 1}$. The measurement of the gyro bias is then said to be equal to the measured angular rates, $\mathbf{b}_g = \tilde{\boldsymbol{\omega}}_{ib}^{b}$. The measurement vector for both the ZVU and ZARU is Equation (3.49).

$$\delta \mathbf{y} = \begin{bmatrix} \delta \mathbf{v} & \mathbf{b}_g \end{bmatrix}^T \tag{3.49}$$

The measurement matrix is a (6×15) matrix for a ZVU and ZARU and is

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.50}$$

where $\mathbf{0}_3$ is a (3×3) matrix of zeros and $\mathbf{I}_3$ is a (3×3) identity matrix.

When straightline walking is detected either an HDR or CHAIN update can be applied. The measurement vector for both is

$$\delta \mathbf{y} = \begin{bmatrix} \delta\psi \end{bmatrix} \tag{3.51}$$

where $\delta\psi$ is the heading error. For the HDR method, the heading error is computed as

$$\delta\psi_k = \psi_k - \psi_{k-1} \tag{3.52}$$

where ψ_k is the current heading at time t_k and ψ_{k-1} is the heading at time t_{k-1}. For the CHAIN method the heading error is given by

$$\delta\psi_k = COG - \psi_{cardinal} \tag{3.53}$$

where $\psi_{cardinal}$ is one of the building cardinal directions. The term COG is the course over ground. It is defined as

$$COG = \arctan2(E_{ks} - E_{ks-1}, N_{ks} - N_{ks-1}) \tag{3.54}$$

where $\{E_{ks}, N_{ks}\}$ are the East and North positions at the current stride and $\{E_{ks-1}, N_{ks-1}\}$ are the East and North positions at the previous stride, respectively. The measurement matrix is a (3×15) matrix for a HDR or CHAIN update

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{I}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{bmatrix} \tag{3.55}$$

where $\mathbf{0}_{1\times3}$ is a (1×3) matrix of zeros and $\mathbf{I}_{1\times3}$ is a (1×3) identity matrix.

The measurement noise $\mathbf{v}_k$ is zero mean and Gaussian with a covariance matrix of $\mathbf{R}_k$. The measurement noise covariance matrix ($\mathbf{R}_k$) associated with the ZVU must be chosen as it is a pseudo measurement and not provided from an outside source. The method used in this thesis was developed in [38] by Foxlin and shown in [32] by Pierce. By making the update covariance no smaller than the current velocity variance, the corrections will occur gradually

over the update period. This is represented mathematically as

$$\mathbf{R}_k = \mathbf{I}_3 \cdot \text{trace}(\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T) \tag{3.56}$$

where the function trace is the summation of the diagonal elements of the matrix, and $\mathbf{I}_3$ is a (3×3) identity matrix. The part of the measurement noise covariance matrix ($\mathbf{R}_k$) associated with the ZARU is similarly found by

$$\mathbf{R}_k = \mathbf{I}_3 \cdot \text{trace}(\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T) + \mathbf{Q}_{4:6,4:6} \tag{3.57}$$

where the measurement covariance also accounts for the noise in the inertial measurements with the gyroscope portion of input covariance matrix ($\mathbf{Q}_{4:6,4:6}$). When applying both ZVU and ZARU, Equations (3.56-3.57) should be concatenated to make $\mathbf{R}_k$ a (6×6) matrix. Note that ZVU measurements do not allow for the estimation of all the gyro biases. The z-axis gyro bias is unobservable in low dynamic situations (e.g. slow turns, straight paths) [23, 32]. To estimate the z-axis gyro bias, a ZARU, HDR, or CHAIN method must be used. The x-axis and y-axis gyro biases are observable with just a ZVU due to their relation with the pitch (θ) and roll (ϕ) angle errors. Note that the method used to compute variance for the ZARU was also used for the HDR and CHAIN methods.

III. Commentary on Measurement Update Type

The yaw axis gyro bias has been shown to be unobservable with only a ZVU as a measurement source. This effect is shown in Figure 3.7. The ZARU has been shown to be an effective method in estimating the yaw axis gyro bias, though drift in the localization solution will still persist. The localization solution with both the ZVU and ZARU method is seen in Figure 3.8. The HDR measurement is another method used to help estimate the yaw axis gyro bias, and reduce the heading drift. The solution with the ZVU and HDR method is seen in Figure 3.9. Note the straight walking points are the epochs (shown with green circles) in which the measurement update was applied. The CHAIN method was introduced as a way to reduce the drift that is

still present when the ZARU and HDR methods are used. The results of this method are shown in Figure 3.10. This update was only applied at three straight walking epochs to demonstrate how the solution is corrected. Notice that after the updates stop, the solution begins to drift again. It should be noted that these results are preliminary and should be treated as such. A thorough analysis of each of these methods is beyond the scope of this work.

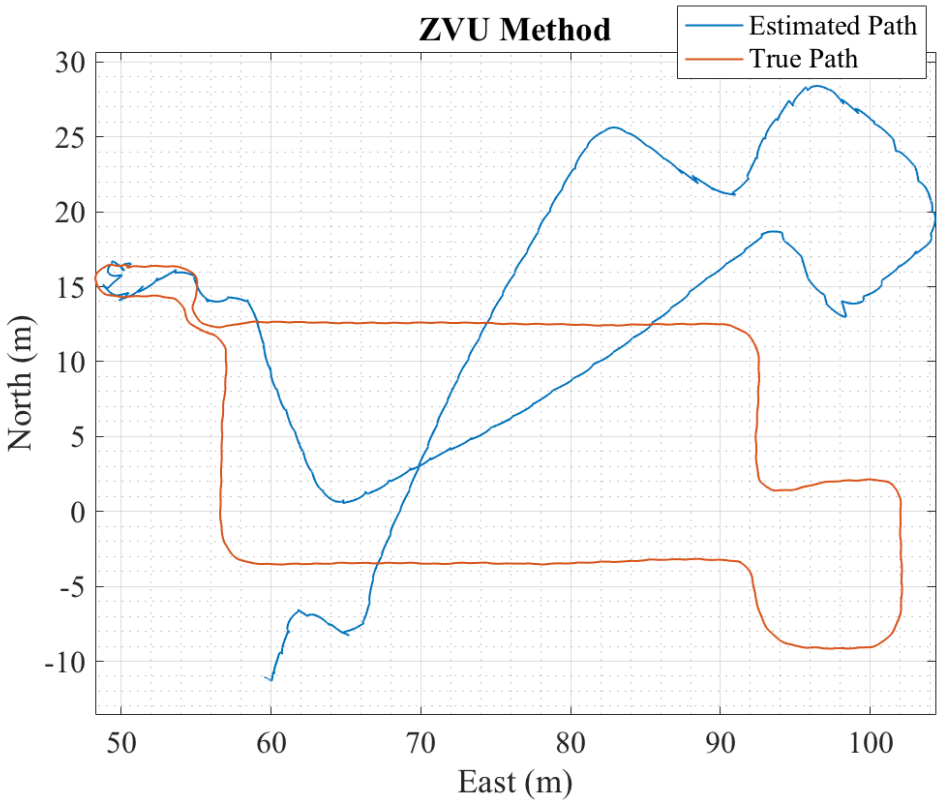

Figure 3.7: Position Estimates with ZVU Method

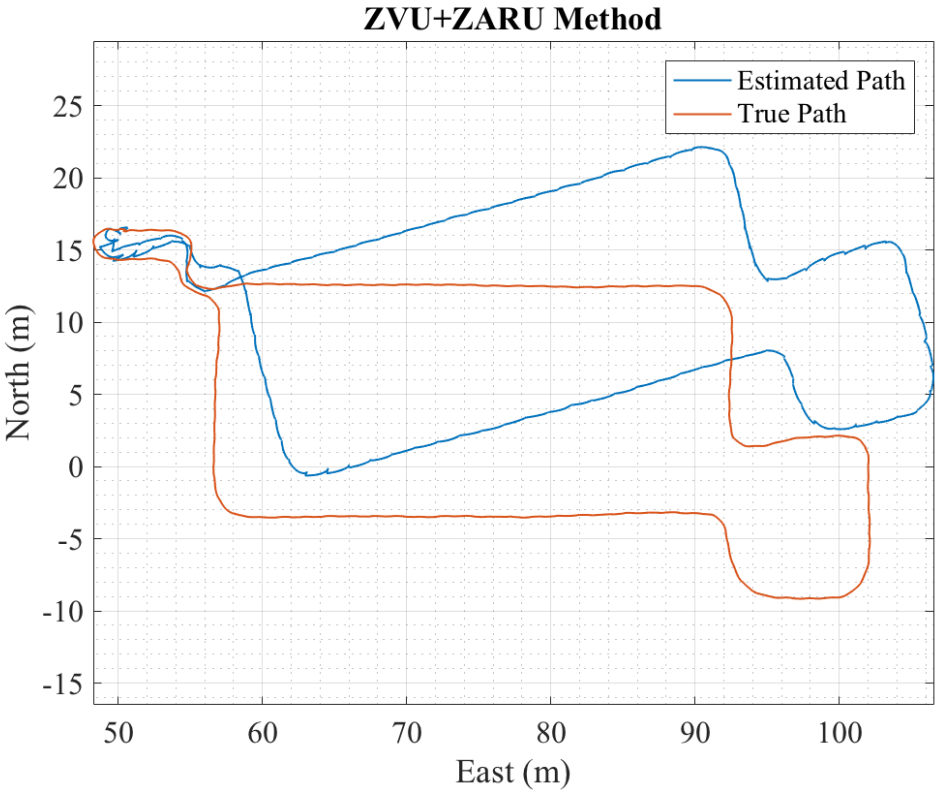

Figure 3.8: Position Estimates with ZVU+ZARU Method

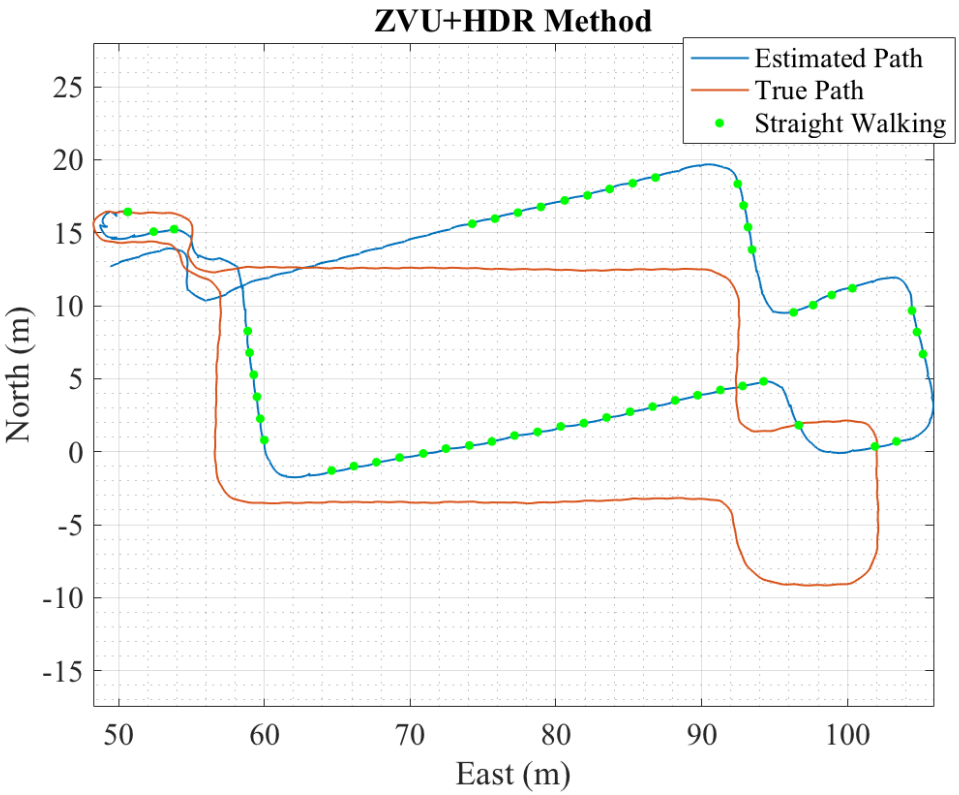

Figure 3.9: Position Estimates with ZVU+HDR Method

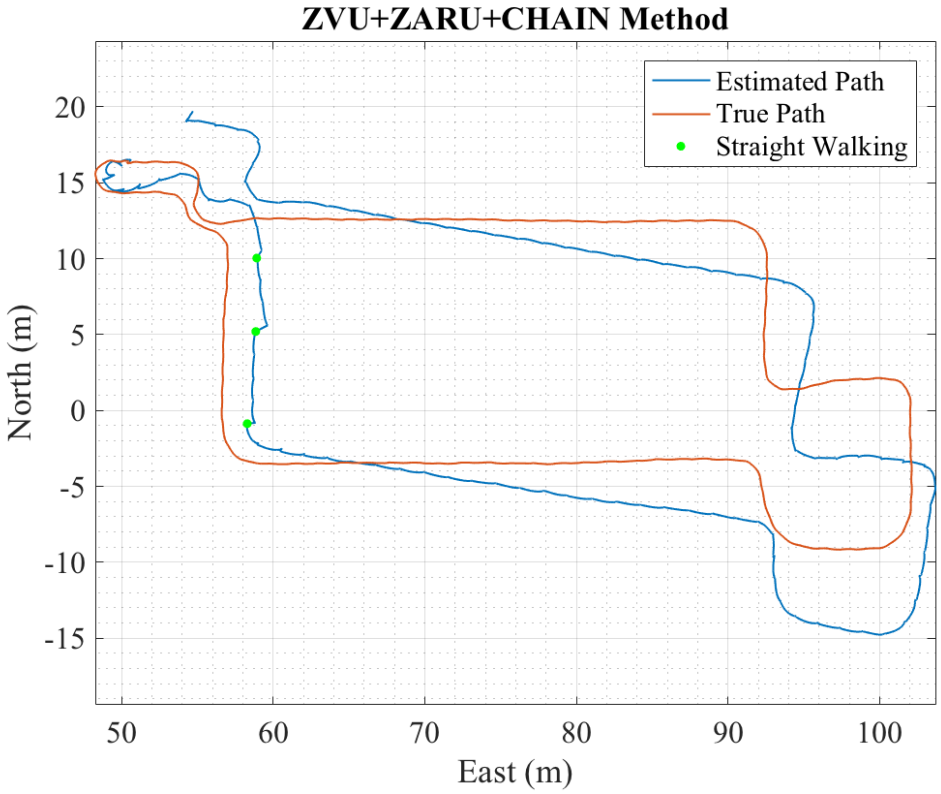

Figure 3.10: Position Estimates with ZVU+ZARU+CHAIN Method

3.3.7 INS Mechanization

The overall framework for a classical KF based INS mechanization is presented in this section. This INS mechanization process has seven steps that are presented in the order that they occur in the following sections. These are based upon work by Jiménez et al. [23].

1. Initialization of Position, Velocity, and Attitude

INS position and velocity must be initialized using an external reference. For this thesis the initial position of the user was found using GPS and the velocity of the foot was assumed to be zero as the user was stationary relative to the ground.

When an INS is stationary, *self-alignment* can be used to initialize the pitch, roll with *leveling* and the heading (yaw) initialized using *gyrocompassing* [3]. The leveling process is based upon the fact that when an INS is stationary, the specific force measured by the accelerometers is the reaction to gravity, which is approximately the down direction in the local navigation frame. Based upon this principle the pitch (θ) and roll (ϕ) can be initialized as

$$\theta_o = \arctan2(-f^b_{ib,y}, -f^b_{ib,z}) \tag{3.58}$$

$$\phi_o = \arctan2\left(f^b_{ib,x}, \sqrt{(f^b_{ib,y})^2 + (f^b_{ib,z})^2}\right) \tag{3.59}$$

where $\{f^b_{ib,x}, f^b_{ib,y}, f^b_{ib,z}\}$ are the measured specific force of the x, y, and z axes, respectively. The term $\arctan2$ is the four-quadrant inverse tangent. For further discussion of how Equations (3.58-3.59) are derived see [3].

Gyrocompassing requires that the IMU be aviation grade or better [3]. Thus it could not be performed in this thesis as a tactical grade IMU was used. The heading was instead initialized using a magnetometer. For details on how to calculate heading using a magnetometer see Section 4.5.1.

2. Bias Compensation

Bias compensation is the removal of the bias from raw accelerometers and gyropscope measurements. Section 3.3.1 modeled the inertial errors as

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{\omega}_{ib}^{b} + \mathbf{b}_g + \mathbf{w}_g \tag{3.60}$$

$$\tilde{\mathbf{f}}_{ib}^{b} = \mathbf{f}_{ib}^{b} + \mathbf{b}_a + \mathbf{w}_a \tag{3.61}$$

where $\{\tilde{\boldsymbol{\omega}}_{ib}^{b}, \tilde{\mathbf{f}}_{ib}^{b}\}$ are the measured (raw) values and $\{\boldsymbol{\omega}_{ib}^{b}, \mathbf{f}_{ib}^{b}\}$ are the true values of the angular rates and specific force, respectively. The bias terms $\{\mathbf{b}_g, \mathbf{b}_a\}$ estimated by the Kalman filter are removed from the raw measurements by subtracting them out. This is mathematically shown as

$$\boldsymbol{\omega}_{ib}^{b} = \tilde{\boldsymbol{\omega}}_{ib}^{b} - \hat{\mathbf{b}}_g \tag{3.62}$$

$$\mathbf{f}_{ib}^{b} = \tilde{\mathbf{f}}_{ib}^{b} - \hat{\mathbf{b}}_a \tag{3.63}$$

where $\{\hat{\mathbf{b}}_g, \hat{\mathbf{b}}_a\}$ are the estimated gyro and accelerometer biases, respectively. Note that the 'true' values $\{\boldsymbol{\omega}_{ib}^{b}, \mathbf{f}_{ib}^{b}\}$ solved for in Equations (3.62-3.63) are not the true values of the body in motion, as they can never be perfectly known. They are instead the optimal estimate of them.

3. Attitude Estimation

After the gyro measurements have been bias compensated, the attitude of the body frame relative to the local navigation frame must be updated. The time derivative of the CTM $\mathbf{C}_b^n$ is given by

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^{b} \tag{3.64}$$

where $\boldsymbol{\Omega}_{ib}^{b}$ is the skew symmetric matrix for angular rates. It is defined as

$$\boldsymbol{\Omega}_{ib}^{b} = \begin{bmatrix} 0 & -\omega_{ib,z}^{b} & \omega_{ib,y}^{b} \\ \omega_{ib,z}^{b} & 0 & -\omega_{ib,x}^{b} \\ -\omega_{ib,y}^{b} & \omega_{ib,x}^{b} & 0 \end{bmatrix} \tag{3.65}$$

where $\{\omega^b_{ib,x}, \omega^b_{ib,y}, \omega^b_{ib,z}\}$ are the bias compensated gyro measurements of the x, y, and z axes, respectively. The solution to Equation (3.65) can be approximated by assuming $\mathbf{\Omega}^b_{ib}$ is time invariant during the interval from time t_{k-1} to t_k. What this means practically is that $\mathbf{\Omega}^b_{ib}$ is assumed to be constant over the time interval, resulting in the following approximation

$$\mathbf{C}^n_{b,k} = \mathbf{C}^n_{b,k-1} exp(\mathbf{\Omega}^b_{ib}\Delta t) \tag{3.66}$$

where Δt is the time interval between samples at time t_{k-1} and t_k. A *Padé* approximation of the matrix exponential in Equation (3.66) is used for the attitude update [23, 43]. This approximation is used to maintain the orthoganality of the CTM [44]

$$\mathbf{C}^{n,-}_{b,k} = \mathbf{C}^n_{b,k-1} \frac{(2 \cdot \mathbf{I}_3 + \mathbf{\Omega}^b_{ib,k}\Delta t)}{(2 \cdot \mathbf{I}_3 - \mathbf{\Omega}^b_{ib,k}\Delta t)} \tag{3.67}$$

where $\mathbf{C}^{n,-}_{b,k}$ is the CTM that represents the attitude of the body frame (b) with respect to the local navigation frame (n) at time t_k. The superscript minus $(-)$ indicates that the value is prior to being corrected with the Kalman filter error estimate. The term $\mathbf{\Omega}^b_{ib,k}$ is the skew symmetric matrix of the angular rates at time t_k.

4. Gravity Compensation

In this step, the acceleration due to the gravitational force is removed from the accelerometer measurements. This must be done as accelerometers measure specific force and not acceleration [3]. The accelerometer measurements are first rotated into the local navigation frame and then the value of $g = 9.81\ (\mathrm{m/s^2})$ is added to the vertical component of the down component of the acceleration. This is shown as

$$\mathbf{a}^n_{ib,k} = \mathbf{C}^n_{b,k}\mathbf{f}^b_{ib,k} + [0, 0, g]^T \tag{3.68}$$

where $\mathbf{a}^n_{ib}$ is the acceleration of the body in the local navigation frame. It should be noted that this step is responsible for the largest amount of error in an inertial navigation solution. When $\mathbf{C}^n_{b,k}$ is not perfectly known, the gravitational acceleration will not be correctly removed.

Because the magnitude of the gravitational acceleration is large compared to the measured acceleration, this will cause a low cost unaided inertial navigation solution to quickly diverge. For further discussion of this topic, see [3, 40].

5. Estimation of Velocity and Position

Next, the acceleration measurements are numerically integrated with Euler (rectangular) integration to obtain the velocity of the body in the local navigation frame. This is given by

$$\mathbf{v}_{ib,k}^{n,-} = \mathbf{v}_{ib,k-1}^{n} + \mathbf{a}_{ib,k}^{n}\Delta t \tag{3.69}$$

where $\mathbf{v}_{ib,k}^{n,-}$ is the velocity of the body prior to being corrected by the Kalman filter estimate of error. This velocity is then integrated to obtain the position of the body in the local navigation frame. This is shown by

$$\mathbf{r}_{ib,k}^{n,-} = \mathbf{r}_{ib,k-1}^{n} + \mathbf{v}_{ib,k}^{n,-}\Delta t \tag{3.70}$$

where $\mathbf{r}_{ib,k}^{n,-}$ is the position of the body prior to being corrected by the Kalman filter error estimate.

6. Applying Corrections to Position, Velocity, and Attitude Estimates

The position, velocity, and attitude estimates are corrected using the error estimates from the Kalman filter. Error is defined as $\delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$, to get the corrected or 'true' state this equation is rearranged to be $\mathbf{x} = \hat{\mathbf{x}} - \delta\mathbf{x}$. Therefore the corrected position estimate is

$$\mathbf{r}_{ib,k}^{n} = \mathbf{r}_{ib,k}^{n,-} - \delta\hat{\mathbf{r}}_k \tag{3.71}$$

where $\mathbf{r}_{ib,k}^{n}$ is the corrected position estimate at time t_k and $\delta\hat{\mathbf{r}}_k$ is the Kalman filter position error estimate at time t_k. Similarly, the corrected velocity estimate is

$$\mathbf{v}_{ib,k}^{n} = \mathbf{v}_{ib,k}^{n,-} - \delta\hat{\mathbf{v}}_k \tag{3.72}$$

where $\mathbf{v}_{ib,k}^{n}$ is the corrected velocity estimate at time t_k and $\delta\hat{\mathbf{v}}_k$ is the Kalman filter velocity error estimate at time t_k.

The attitude correction (refinement) is applied assuming that the attitude errors are small. The correction is given using another *Padé* approximation as

$$\mathbf{C}_{b,k}^{n} = \frac{(2 \cdot \mathbf{I}_3 + \mathbf{\Psi}_k \Delta t)}{(2 \cdot \mathbf{I}_3 - \mathbf{\Psi}_k \Delta t)} \mathbf{C}_{b,k}^{n,-} \tag{3.73}$$

where $\mathbf{C}_{b,k}^{n}$ is the corrected rotation matrix. The term $\mathbf{\Psi}_k$ is the skew symmetric matrix of the attitude error estimated by the Kalman filter. It is defined as

$$\mathbf{\Psi}_k = \begin{bmatrix} 0 & -\delta\hat{\psi}_k & \delta\hat{\theta}_k \\ \delta\hat{\psi}_k & 0 & -\delta\hat{\phi}_k \\ -\delta\hat{\theta}_k & \delta\hat{\phi}_k & 0 \end{bmatrix} \tag{3.74}$$

where $\{\delta\hat{\phi}_k, \delta\hat{\theta}_k, \delta\hat{\psi}_k\}$ are the estimated roll, pitch, and yaw errors, respectively.

7. Zeroing Out Position, Velocity, and Attitude States

The navigation states $\{\delta\mathbf{r}_k, \delta\mathbf{v}_k, \delta\boldsymbol{\varphi}_k\}$ must be zeroed out after the corrections have been applied. This is done to ensure that the same error is not removed more than once. This process is shown by the following equations

$$\delta\hat{\mathbf{r}} = \mathbf{0}_{3\times 1} \tag{3.75}$$

$$\delta\hat{\mathbf{v}} = \mathbf{0}_{3\times 1} \tag{3.76}$$

$$\delta\hat{\boldsymbol{\varphi}} = \mathbf{0}_{3\times 1} \tag{3.77}$$

where $\mathbf{0}_{3\times 1}$ is a (3×1) vector of zeros. After this occurs, the steps repeat starting with step two.

Chapter 4

Pedestrian Dead Reckoning: A Step Detection Approach

4.1 Introduction

This chapter introduces an alternative method of PDR, that performs step detection and estimates step length and heading of the pedestrian [3]. This method has been shown in many works, some of which are [22, 27, 45–55]. The steps of this method are discussed in the remaining sections.

4.2 PDR-SD Systems

When pedestrians exhibit normal movement such as walking, IMUs that are not foot-mounted will always have specific force readings much greater than zero. This makes it impossible to regularly apply ZVUs as the IMU only has 'zero-velocity' when the pedestrian is stationary. PDR with step detection (PDR-SD) allows for the IMU to be mounted in multiple locations on the pedestrian's body as it *does not* rely upon inertial navigation with ZVUs. The PDR-SD methods can be broken into four parts [51]. They are: step detection, step length estimation, heading determination, and navigation system update. These parts will be explained in the following sections.

4.3 Step Detection

Most often step detection is performed using just accelerometers [3, 4, 47, 50, 53, 56, 57]. Step detection can be accomplished by using either measurements from a single accelerometer (i.e. $\tilde{f}^b_{ib,x}, \tilde{f}^b_{ib,y}, or \tilde{f}^b_{ib,z}$) or the magnitude of the accelerometers. The magnitude of the accelerometer

measurements is found by calculating the Euclidean norm shown in Equation (4.1).

$$\left\|\tilde{\mathbf{f}}^b_{ib}\right\| = \sqrt{(\tilde{f}^b_{ib,x})^2 + (\tilde{f}^b_{ib,y})^2 + (\tilde{f}^b_{ib,z})^2} \tag{4.1}$$

This is effectively using the accelerometers as a *pedometer*. If only single accelerometer is used, it must be roughly parallel with the pedestrian coordinate frame (Figure 2.4) z-axis for steps to be easily identified. If the orientation of the sensor relative to the pedestrian is unknown or it is not mounted in such a manner that an axis aligns with the pedestrian frame z-axis then the magnitude of all the accelerometer measurements should be used [3, 50]. In this thesis, the magnitude of all the accelerometers was chosen as it is the more robust approach.

The accelerometer measurement exhibits a double-oscillatory pattern due to the ambulatory motion pedestrians exhibit. A typical pattern of the raw measurements while a pedestrian is walking, can be seen in Figure 4.1. The z-axis can be seen to have large periodic motion relative to the other two axes. This is because the z-axis of the IMU was roughly parallel with the pedestrian z-axis. The magnitude of the accelerometer triad at other locations on the body have similar periodic signals. This allows for consistent step detection as the measurements have a predictable pattern. The most popular method uses thresholding to detect steps [50, 51, 58, 59]. Another commonly used method is peak detection. This method is more prone to error, thus was not considered in this work [47, 53, 60]. Furthermore, more advanced methods exist using correlation and spectral analysis algorithms. These are beyond the scope of this thesis, but further discussion can be found in [4, 18]. Step detection methods not only apply to accelerometer measurements, but other sensors can be used as well. Some other sensors that have been used successfully are: gyros, magnetometers, and pressure sensors [4, 18, 48]. For this thesis, the thresholding method using accelerometers was chosen for its effectiveness and simplicity.

For the threshold method, a new step is detected when there is an 'acceleration zero crossing'. For the most basic threshold detection method, acceleration zero crossings occur when the magnitude of the specific force readings cross the magnitude of the acceleration due to gravity ($|g| = 9.81\ \mathrm{m/s^2}$) [3]. This is based upon the knowledge that when a step begins at footfall, the magnitude of accelerometers for a torso mounted IMU is approximately the magnitude of

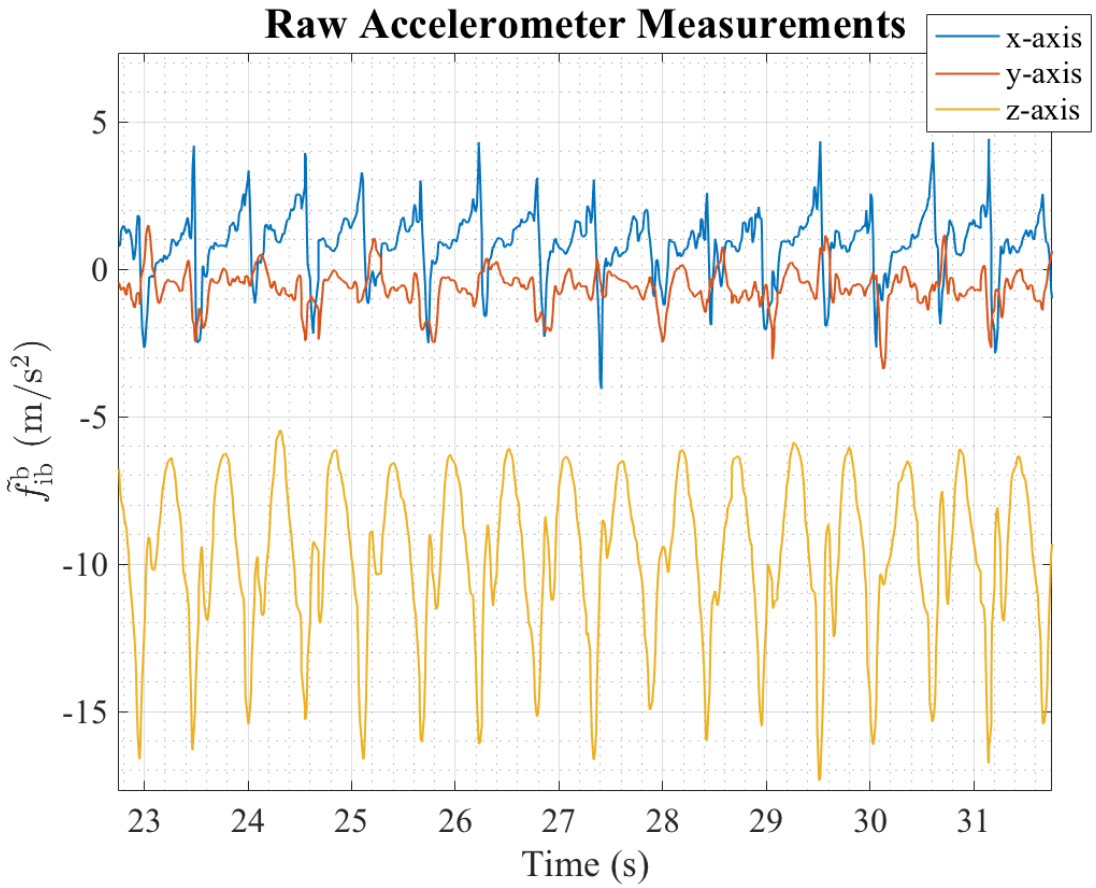

Figure 4.1: Raw Accelerometer Measurements During Walking

the acceleration due to gravity $\left\|\tilde{\mathbf{f}}_{ib}^{b}\right\| \approx |g|$. Note that this is only true for normal walking on a flat surface. Figure 4.2 illustrates step detection with a simplistic thresholding method along with the relevant parts of the gait cycle.

A more advanced thresholding method calculates a maximum, minimum, and average threshold for a window of samples [50]. The size of the window is determined by the IMU sample rate. For example, if the IMU was sampled at 100 Hz the thresholds would be updated every 100 samples. Four conditions have to be met for there to be a detected step. The conditions qualitatively are:

1. The accelerometer signal must have a negative slope. This means the current measurement must be less than the previous measurement.
2. The previous epoch must be above the average threshold and the current epoch must be below the average threshold. These are called acceleration crossings.

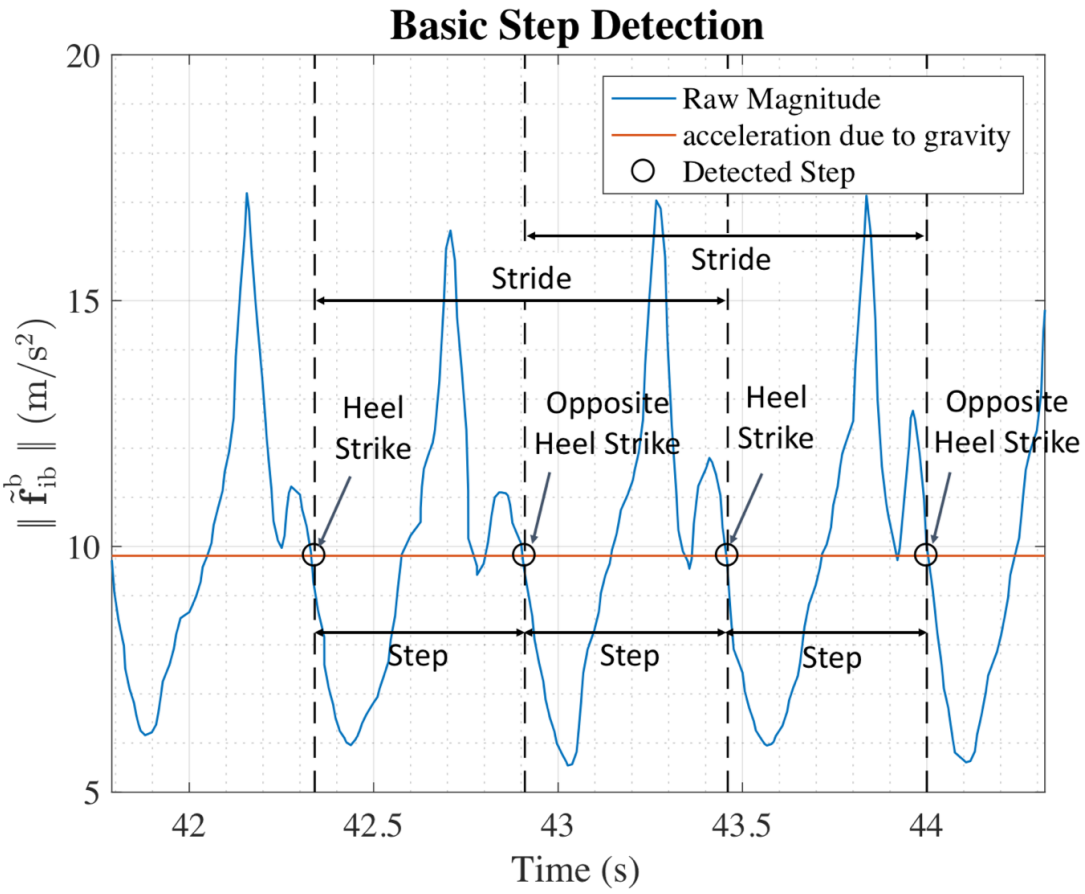

Figure 4.2: Basic Step Detection [3, 47, 49, 51]

3. There must be a minimum time between each detected step. The minimum time between each step varies based on the type motion (e.g. walking, running), the type of IMU and specific algorithm implementation.
4. The minimum and maximum thresholds must be values that indicate pedestrian movement. This is an additional condition, that was added to prevent false step detections during quasi-stationary periods.

This method applied to a segment of a walking data set is shown in Figure 4.3, showing that steps are consistently detected for a chest mounted IMU. This dynamic thresholding method is more robust at detecting steps for irregular walking and walking on inclined surfaces. The pseudo code for a post-process step detection algorithm is presented in Algorithm 1 at the end of this section. This algorithm can be modified and implemented in real time. The parameters used and their respective abbreviations are listed in Table 4.1. It should be noted that the accelerometer data was prefiltered in post-process using a zero-lag moving average filter.

This was done to ensure the accelerometer magnitude signal only had a single threshold crossing during each step. Other methods that could be used are band-pass filtering or a wavelet transform [47, 61].

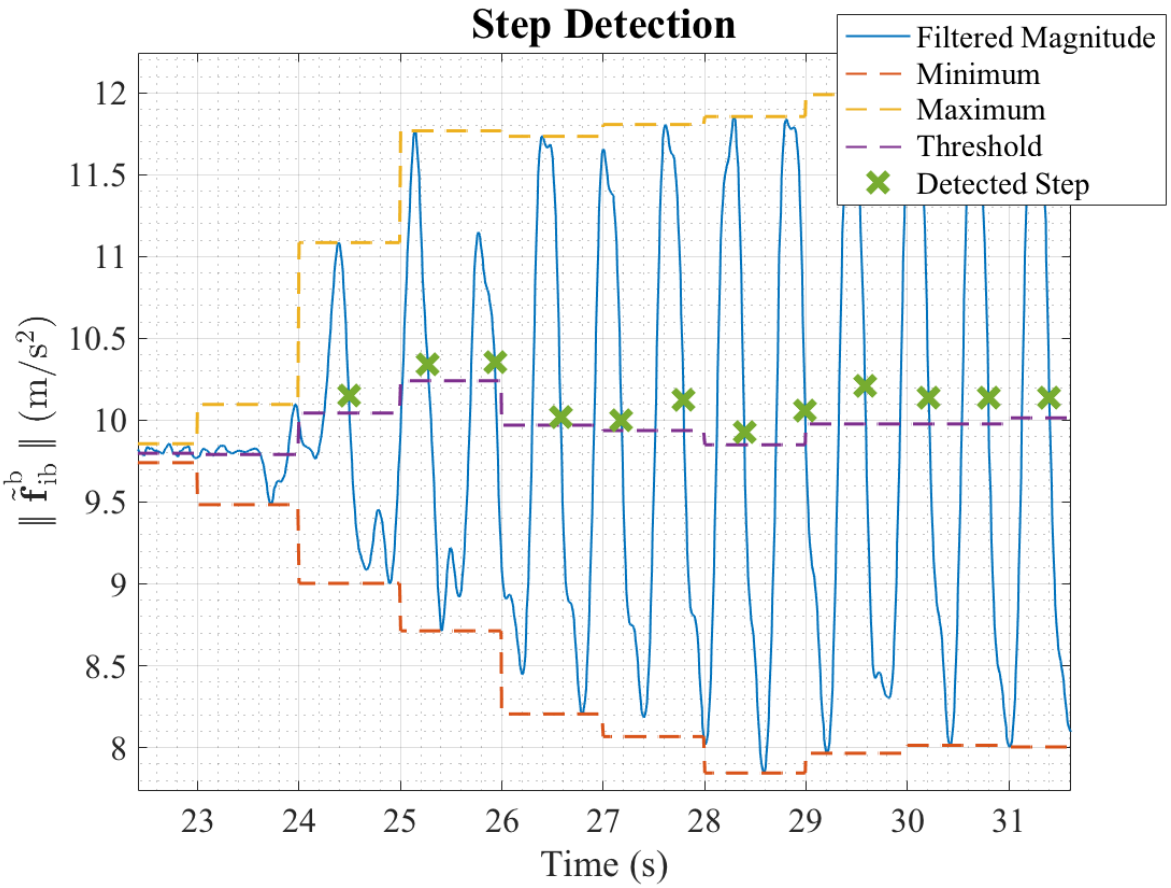

Figure 4.3: Step Detection

Table 4.1: Algorithm 1 Parameter Definitions

Algorithm 1 Parameter Definitions	
Parameter (abbreviation)	Value (if applicable)
Accelerometer Magnitude (accel)	N/A
Sample Rate (SR)	100 (Hz)
Minimum Time Between Steps (t_{min})	0.2 (s)
Period Between Samples (Δt)	1/SR (s)
Time Window (TW)	0.2 (s)
Sample Window (SW)	$round(TW/\Delta t)$
Number of steps ($stepCount$)	N/A
Time when step detected ($stepTimes$)	N/A
True or false if step detected (stepDetect)	1 (true), 0 (false)
Filtering Window (M)	25 samples
Minimum Threshold (TH_{min})	N/A
MaximumThreshold (TH_{max})	N/A
Average Threshold (TH_{avg})	N/A
Checks if all conditions are true ($logical$)	N/A
Finds index of where condition is true ($find$)	N/A

Algorithm 1: Step Detector

Input: $accel_{raw}, time_{accel}$
Output: $stepIndex, stepTimes$
Initialization: $SR = 100$ Hz; $t_{min} = 0.2$ (s); $accel_{filt} = accel_{raw}$;$\Delta t = 1/SR$ (s); $TW = 0.2$ (s);
$SW = round(TW/\Delta t)$; $stepCount = 1$; $stepTimes = 0$; $stepDetect = 0$; M = 25 (must be odd number)
% Zero-lag Moving Average Filter
for $i = (M+1) : (length(accel_{raw}) - 1)$ **do**
    $ii = (i - M) + (M - 1)/2$; %Center of filtering window
    $accel_{filt}(ii) = mean(accel_{raw}((i - M) : i - 1))$;
end
% Calculate Thresholds
for $i = (1 : SR : (length(accel_{raw}) - 1)$ **do**
    $TH_{min}(i : i + SR - 1) = min(accel_{filt}(i : (i + SR - 1)))$;
    $TH_{max}(i : i + SR - 1) = max(accel_{filt}(i : (i + SR - 1)))$;
    $TH_{avg}(i : i + SR - 1) = (TH_{min}(i : i + SR - 1) + TH_{max}(i : i + SR - 1))/2$;
end
for $i = 1 : ((length(accel_{raw}) - SW)$ **do**
    % Condition 1
    if $accel_{filt}(i + 1) < accel_{filt}(i)$ **then**
        $C1 = true$
    else
        $C1 = false$
    end
    % Condition 2
    if $accel_{filt}(i) > TH_{avg}(i) \quad \& \quad accel_{filt}(i + 1) < TH_{avg}(i)$ **then**
        $C2 = true$
    else
        $C2 = false$
    end
    % Condition 3
    if $(time_{accel}(i) - stepTimes(stepCount)) > t_{min}$ **then**
        $C3 = true$
    else
        $C3 = false$
    end
    % Condition 4
    if $TH_{min} < 9.75 \quad \& \quad TH_{max} > 10.25$ **then**
        $C4 = true$
    else
        $C4 = false$
    end
    $stepDetect(i) = logical(C1 \quad \& \quad C2 \quad \& \quad C3 \quad \& \quad C4)$
    if $stepDetect(i) = true$ **then**
        $stepCount = stepCount + 1$;
        $stepTimes(stepCount) = time_{accel}(i)$
    else
    end
    $stepIndex = find(stepDetect)$;
end

4.3.1 The Effects of IMU Location on Step Detection

When using the PDR-SD method, IMUs can be mounted at multiple locations on the pedestrian's body. Some of these possible IMU locations in a fixed setting are illustrated in Figure 4.4. A fixed setting is one in which the IMU is mounted to a certain area and does not change location on the pedestrian. PDR-SD with the foot-mounted IMU has been exhibited by Cho, S.Y. et al. as it easily identifies steps when the stance phase and swing phase are known [62]. Although a foot-mounted IMU is an option, it is preferable to use inertial navigation techniques when mounting on the foot as ZVUs help restrict the error growth [22,48,58]. When an IMU is not foot-mounted and mounted elsewhere on the body, it is sometimes referred to as a *body-mounted* IMU [22]. Some examples of the body-mounted IMU approach include, but are not limited to: IMU mounted on a backpack [22], waist-mounted IMUs [12,53], and an IMU mounted on a pedestrian's head using a helmet [56].

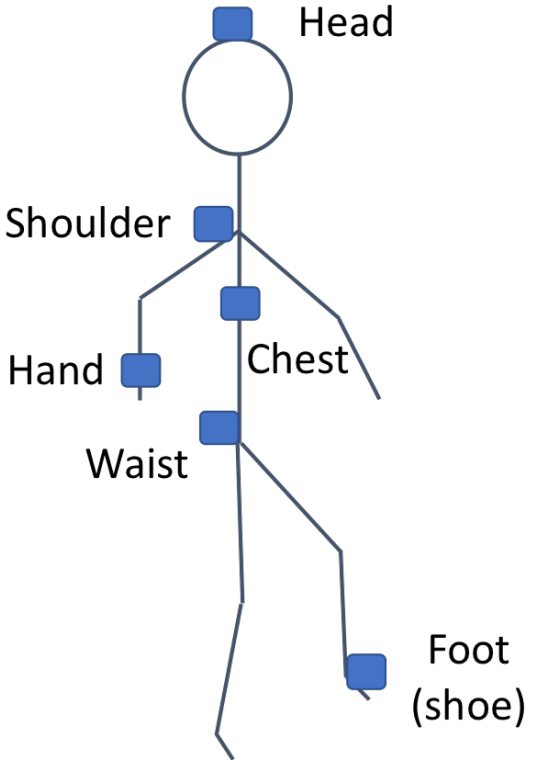

Figure 4.4: Possible fixed IMU locations on pedestrian body [3]

If sensors are attached to appendages (e.g. arms and legs), they can be 'excited' independently from the pedestrian. In other words, they exhibit movement while the pedestrian's center

of gravity (CoG) does not move. This is why it is commonly accepted that sensors should be attached near the body's center of mass, because movement of the center of mass more often reflects movements of the entire body [63, 64]. Note here, that the terms center of mass and center of gravity are used synonymously and these points are exactly the same on a pedestrian standing upright. The anotomical position for the CoG of a pedestrian when standing, lies approximately anterior to the second sacral vertebra and is the origin of the pedestrian frame as shown in Figure 2.4 [65]. When an IMU attached to an appendage moves independently, false steps can be detected.

Tests were conducted to verify this using a Xsens MVN Awinda system, shown in Figure 4.5. This is an inertial motion capture system used to perform full-body human motion capture. The results of Test #1 are shown in Figures 4.6 and 4.7. Figure 4.6 shows that around the 17 second mark a false step is detected for the motion captured for the user's right hand. This is known to be a false step detection, as the test subject is stationary at the 17 second mark. This was determined by analyzing the step detection performed on motion captured for the user's pelvis in Figure 4.7. This figure shows no movement or steps detected.

Test #2 showed that mounting an IMU on the head can produce similar issues. This is indicated by the false step detections from a head mounted IMU at the 20 second mark in Figure 4.8. This is known to be a false step detection as motion captured for the pelvis in Figure 4.9 indicates the pedestrian is stationary.

In this thesis, the IMU was placed on the pedestrian's torso. There are multiple locations on the pedestrian's torso that could have been used, including the waist (pelvis), chest, back, and shoulders. The choice of which location produces the best result is beyond the scope of this thesis. Note that when PDR-SD is used in this thesis, the IMU was mounted on the pedestrian's chest. For further discussion of IMU locations used to perform step detection, refer to [4].

Figure 4.5: MVN Awinda System
(Image courtesy of Auburn Univ. Industrial Eng. Dept.)

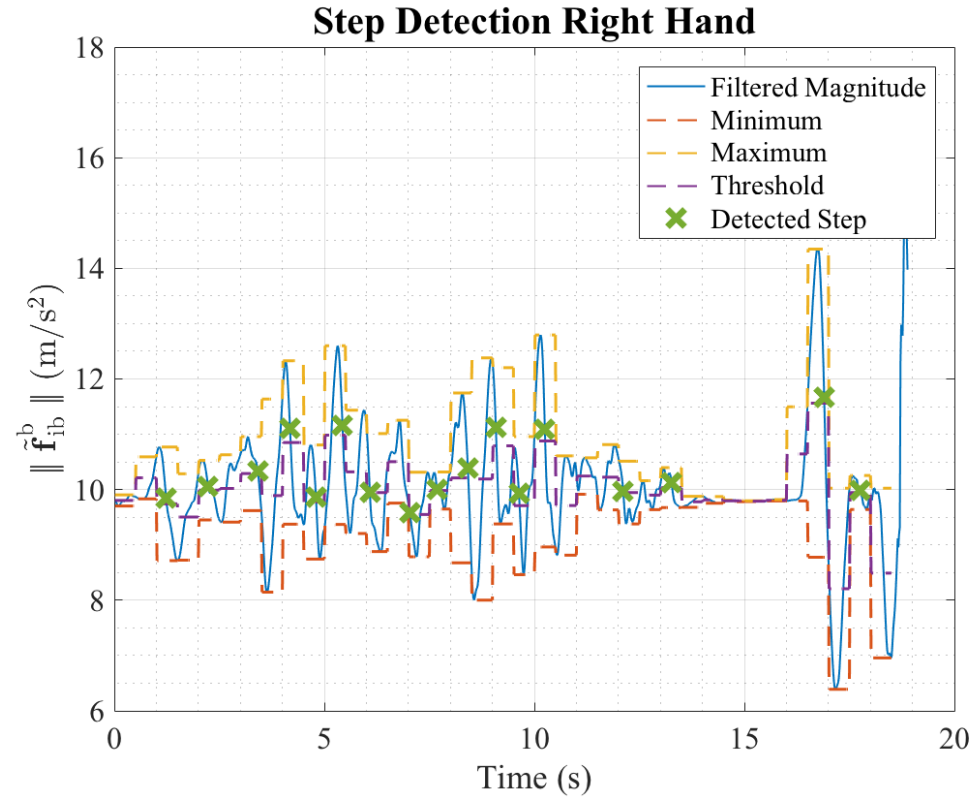

Figure 4.6: Test #1 Step Detection

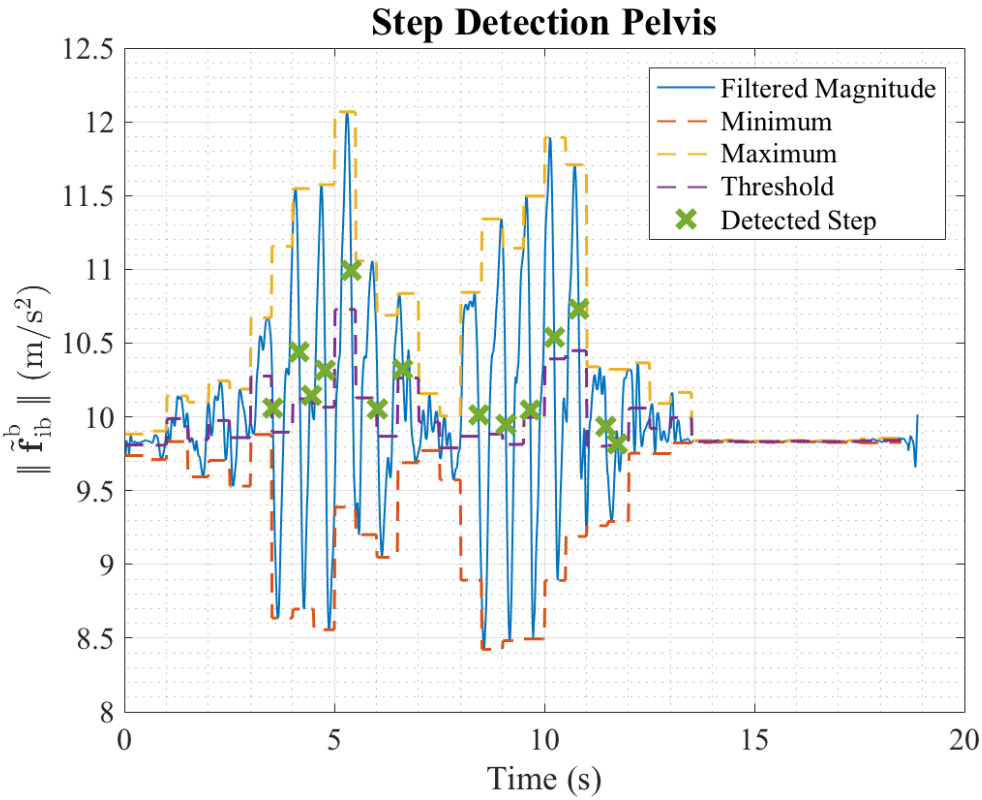

Figure 4.7: Test #1 Step Detection

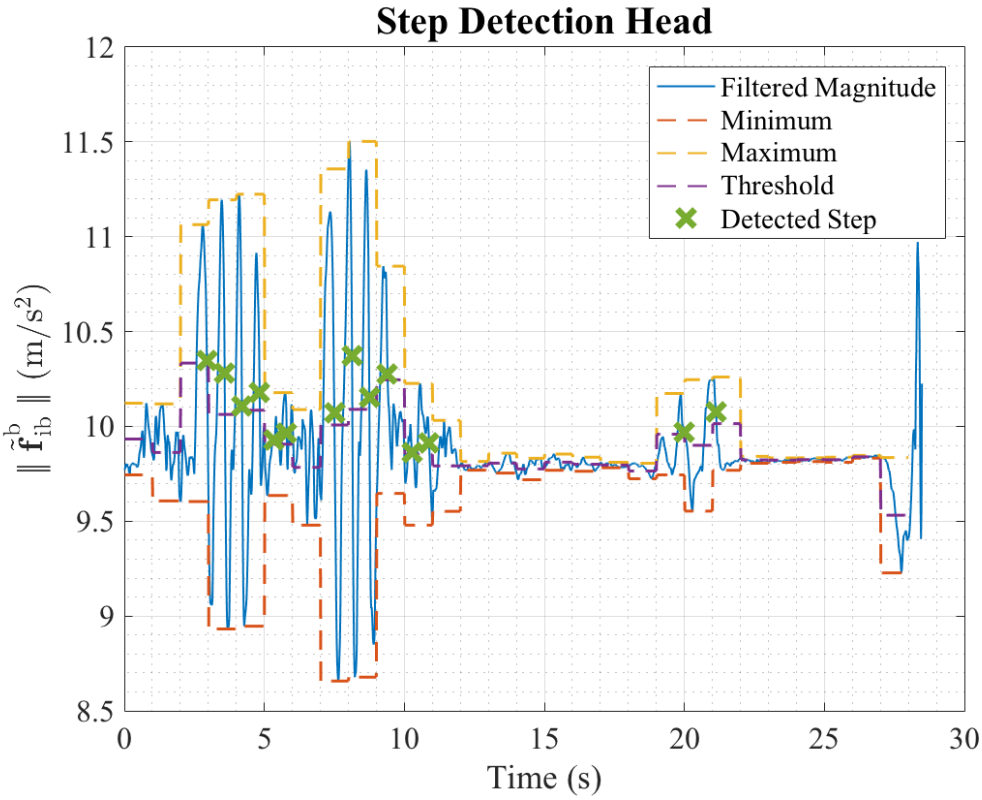

Figure 4.8: Test #2 Step Detection Head

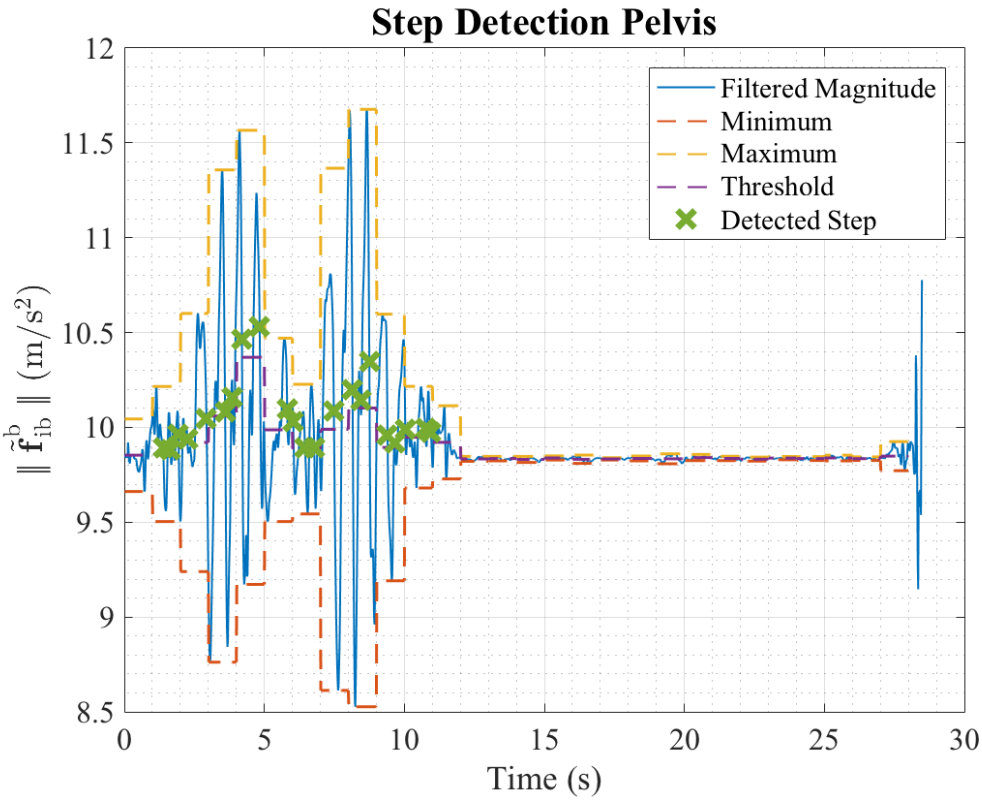

Figure 4.9: Test #2 Step Detection Pelvis

4.4 Step Length Determination

The most simplistic algorithms assume a constant step length (SL), as naturally walking pedestrians have an almost constant step length. This is an easy approach to implement as only the average step length for a user needs to be identified. This can be easily found by recording distance traveled and number of steps taken by a pedestian and using the simple formula

$$SL_1 = (x)/(n) \tag{4.2}$$

where x is the distance traveled and n is the number of steps taken. Some pedometer manufacturers list step length as a function of height [27]. This relation is

$$SL_2 = (h) \cdot (K) \tag{4.3}$$

where h is the height of the user and K is a constant based upon the user's height. The often quoted constant for the pedestrian height in inches is $K = 0.413$. This relation is created with a survey of multiple pedestrians walking naturally. Unfortunately, step length can vary as much as 40% from person to person, as it is largely dependent upon leg length. Furthermore, step length can vary as much as 50% as a pedestrian changes their gait (i.e. walking slower and faster than a natural pace) [57]. Unless the pedestrian assumes a natural walking pace for the entire distance traveled, a fixed step length is not adequate and will introduce error into the positioning solution.

A slightly more complex method is presented by Zhao, which uses a simple tabular method for computing step length based upon step frequency and height [50]. Note that the author calls a step length a stride in his work. The table of values for this method is presented in Table 4.2.

The following two step length formulas are intended to be used with a sensor mounted on the pedestrian's hip [18]. Consequently, the most popular application has been to use a smartphone, containing an IMU, in a pant or jacket pocket. Each formula uses the vertical acceleration to calculate the step length. Thus the smartphone or sensor must either be aligned such a way that the vertical component of acceleration is directly measured, or the orientation

Table 4.2: SL_3 Definition

Steps per 2 seconds	SL_3 (m)
0-2	height/5
2-3	height/4
3-4	height/3
4-5	height/2
5-6	height/1.2
6-8	height/4
$\geq$8	1.2 $\times$ height

of the smartphone must be known. Weinburg's work created a dynamic step length estimation formula based upon the maximum vertical displacement ('Bounce') of the hip (or upper body) [18, 57, 66]. It was shown that stride length was a function of the vertical displacement and the angle between the maximum and minimum hip position. This angle is set as a constant even though it is typically not. The formula has been modified to estimate step length instead of stride length. This results in the empirical relation

$$SL_4 = K\sqrt[4]{(a_{max} - a_{min})} \tag{4.4}$$

where a_{max}, a_{min} are the maximum and minimum measured values of the vertical acceleration during a step. The relation assumes the angle between the maximum and minimum hip position is constant which is generally not true. The formula contains a constant K to tune it for each individual user. This method also reports step lengths to be within 8% of their true values. The second step length formula for hip located sensors is

$$SL_5 = 0.1\sqrt[2.7]{\frac{\sum_{i=1}^{k=N}\left\|a_{vert,i}\right\|}{N}}\sqrt{\frac{K}{\sqrt{T \cdot a_{peak}}}} \tag{4.5}$$

where T is the step duration, a_{vert} is the vector of measured vertical accelerations during a single step. The term a_{peak} is the difference between the maximum and minimum vertical accelerations during the step and N is the number of samples during one step. Similar to Equation (4.4), K is a calibration constant. The empirical relation in Equation (4.5) was created

by [27] and is based upon step length formula by Kim et al. [45]. This formula is

$$SL_6 = 0.98\sqrt[3]{\frac{\sum_{i=1}^{k=N}\left\|a_{vert,i}\right\|}{N}} \tag{4.6}$$

where a_{vert} is the vector of measured vertical accelerations during a single step. Note that this formula is intended to be used for a foot-mounted IMU.

Another method uses a simple biomechanical model of a pedestrian, it is based upon work by [67] and is presented by [66]. This model assumes the pedestrian is a kneeless biped which is modeled as an inverted pendulum. This is given by the following function

$$SL_7 = K\sqrt{2LY - Y^2} \tag{4.7}$$

where Y is the vertical displacement of the pedestrian's hip and L is the length of the pedestrian's leg. The estimates are each scaled with a constant K to calibrate the results to the individual user.

The last method of step length determination models step length as a function of certain signal characteristics such as step frequency and variance of specific force. Step frequency has been shown to be strongly related to step length [68, 69]. Gusenbauer et al. chose to relate the step frequency to step length using the linear function [54]

$$SL_8 = K_1 + K_2 f \tag{4.8}$$

where f is step frequency and K_1 and K_2 are the model coefficients [54]. Other works include additional terms to their models. Ladetto models step length as a function of both step frequency and variance of the specific force, resulting in the function

$$SL_9 = K_1 + K_2 f + K_3 \sigma_{\mathrm{f}}^2 \tag{4.9}$$

where f is step frequency, σ_{f}^2 is the variance of the magnitude of the specific force, and K_1, K_2, and K_3 are the model coefficients [47]. Groves et al. also include a slope term in their work,

resulting in the function

$$SL_{10} = K_1 + K_2 f + K_3 \sigma_{\mathrm{f}}^2 + K_4 S \tag{4.10}$$

where f is step frequency, σ_{f}^2 is the variance of the magnitude of the specific force, and S is the slope term. K_1, K_2, K_3, and K_4 are the model coefficients [22]. Note that Groves et al. found the step frequency coefficient K_2 and the slope coefficient K_4 were both poorly estimated from their data. Hence they reduced the number of terms in their model, resulting in the equation

$$SL_{11} = K_1 + K_3 \sigma_{\mathrm{f}}^2 \tag{4.11}$$

where σ_{f}^2 is the variance of the magnitude of the specific force and K_1 and K_3 are the model coefficients.

The step frequency is calculated by

$$f_k = 1/(t_{k_{stop}} - t_{k_{start}}) \tag{4.12}$$

where $t_{k_{start}}$ is the time at the beginning of the step, and $t_{k_{stop}}$ is the time at the end of the step. In order to calculate the variance of the magnitude of the specific force, the magnitude is first found by computing the Euclidean norm with

$$\left\| \tilde{\mathbf{f}}^b_{ib,(k_{start}:k_{stop})} \right\| = \sqrt{(\tilde{f}^b_{ib,x,(k_{start}:k_{stop})})^2 + (\tilde{f}^b_{ib,y,(k_{start}:k_{stop})})^2 + (\tilde{f}^b_{ib,z,(k_{start}:k_{stop})})^2} \tag{4.13}$$

where k_{start} is the first epoch of the step and k_{stop} is the last epoch of the step. Then the variance of the magnitude of the specific force is computed as

$$\sigma_{\mathrm{f}}^2 = \frac{\left(\sum_{k=k_{start}}^{k_{stop}} \left(\left\| \tilde{\mathbf{f}}^b_{ib,k} \right\| - \mu_{(k_{start}:k_{stop})} \right) \right)^2}{k_{stop} - k_{start}} \tag{4.14}$$

where $\mu_{(k_{start}:k_{stop})}$ is the mean of Equation (4.13).

A comprehensive comparison of all the above step length formulas is beyond the scope of this thesis. A comparison of step length formulas: SL_8, SL_9, and SL_{11} is presented by Jahn et

al. [66]. The next section will introduce a short discussion of the linear least squares methods used to solve for the model coefficients in Equations (4.8-4.11). In the following section, least squares estimation will be introduced followed by a short study of the step length functions that are based upon empirical data. Within this study it will be shown how to estimate their model parameters.

4.4.1 Linear Least Squares Estimation

The model coefficients of each formula are estimated using linear least squares techniques. This is formally called parameter identification, and in the field of statistics the process of fitting a model to measured data is known as *regression*. Least squares computes the estimate ($\hat{\mathbf{x}}$) that minimizes the sum of the square of the residual errors [34, 35]. The following linear least squares techniques will be discussed: ordinary least squares, weighted least squares, and recursive least squares.

I. Ordinary Least Squares

The simplest form of linear least squares is ordinary least squares (OLS). It computes an optimal estimate of **x** by obtaining the solution to

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \tag{4.15}$$

where $\hat{\mathbf{x}}$ is the $(n \times 1)$ vector of estimates. The term **H** is the $(m \times n)$ measurement matrix, that relates the measurements to the state vector with a linear function. Lastly, **y** is the $(m \times 1)$ measurement vector. The estimate of the error covariance matrix is given as

$$\hat{\mathbf{P}} = (\mathbf{H}^T\mathbf{H})^{-1}\sigma^2 \tag{4.16}$$

where $\hat{\mathbf{P}}$ is the $(n \times n)$ estimated error covariance matrix, **H** is the $(m \times n)$ measurement matrix, and σ^2 is the variance of the measurement noise. The estimated covariance matrix provides an

estimate of the accuracy and correlation of state estimates. The estimate of σ^2 is computed by

$$\hat{\sigma}^2 = \frac{\mathbf{e}^T\mathbf{e}}{m-n} \tag{4.17}$$

where $\mathbf{e}$ is the $(m \times 1)$ vector of residuals. The vector of residuals is given by

$$\mathbf{e} = \mathbf{y} - \mathbf{H}\hat{\mathbf{x}} \tag{4.18}$$

For the derivation of Equation (4.17) see [70].

II. Weighted Least Squares

In the previous section, Equations (4.15) and (4.16) place an equal amount of confidence on each measurement. In the unlikely event that each measurement is made with equal confidence, the OLS approach is sufficient. However, the weighted least squares (WLS) approach, solves the problem of unequal measurements by enabling the ability to place more confidence in certain measurements over others. This results in the equations

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y} \tag{4.19}$$

$$\hat{\mathbf{P}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} \tag{4.20}$$

where $\mathbf{W}$ is the $(n \times m)$ weighting matrix. This weighting matrix can either be set empirically or can be found by inverting the measurement noise matrix ($\mathbf{R}$), and is expressed as $\mathbf{W} = \mathbf{R}^{-1}$. The OLS and WLS method each process all of the measurements simultaneously and are known as *batch* methods. The next section will introduce a method of least squares, that processes measurements as they become available sequentially in time.

III. Recursive Least Squares

Using OLS or WLS may not be possible over long periods of time as the measurement matrix ($\mathbf{H}$) may grow too large to be computed effeciently [35]. Therefore, as new measurements

become available, new estimates need to be computed without completely recalculating Equations (4.19-4.20). Recursive least squares (RLS) allows for a new estimate ($\hat{\mathbf{x}}_k$) to be computed from just the previous state estimate ($\hat{\mathbf{x}}_{k-1}$) and the current measurement ($\mathbf{y}_k$). Note that the previously discussed KF is also a recursive estimator. The KF in Table 3.1 is equivalent to RLS with the process noise covariance is set equal to zero ($\mathbf{Q}_d = 0$) and the state transition matrix is an identity matrix ($\mathbf{\Phi}_{k-1} = \mathbf{I}$) [34]. RLS calculates a recursive least squares estimate with the following equations. First the ($n \times m$) optimal gain matrix ($\mathbf{K}_k$) is computed by

$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{4.21}$$

where $\mathbf{P}_{k-1}$ is the ($n \times n$) estimate covariance matrix at time t_{k-1}, $\mathbf{H}_k$ is the ($m \times n$) measurement matrix at time t_k, and $\mathbf{R}_k$ is the ($m \times m$) measurement noise covariance matrix at time t_k. The state estimates are computed by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}) \tag{4.22}$$

where $\hat{\mathbf{x}}_k$ is the ($n \times 1$) vector of state estimates and $\mathbf{y}_k$ is the ($m \times 1$) vector of measurements all at time t_k. The estimate covariance matrix is updated by

$$\mathbf{P}_k = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k-1}(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k{\mathbf{K}_k}^T \tag{4.23}$$

where $\mathbf{I}_n$ is the ($n \times n$) identity matrix. Note that Equation (4.23) is just one way to update the estimate covariance matrix, another form was shown in the measurement update section in Table 3.1. These equations are mathematically identical thus they can be used with equal validity.

If no knowlege of the states ($\mathbf{x}$) is available beforehand the covariance of the estimate should be theoretically initialized as $\mathbf{P}_0 = \infty \cdot \mathbf{I}_{n\times n}$. However, the initialization can be computed in a more deterministic fashion by Equation (4.24) along with the initial state estimate in Equation (4.25)

$$\mathbf{P}_0 = \left(\frac{1}{\alpha^2} \cdot \mathbf{I}_{n\times n} + \mathbf{H}_0^T\mathbf{W}_0\mathbf{H}_0\right)^{-1} \tag{4.24}$$

$$\hat{\mathbf{x}}_0 = \mathbf{P}_0 \left(\frac{1}{\alpha}\beta + \mathbf{H}_0^T \mathbf{W}_0 \mathbf{H}_0 \right) \tag{4.25}$$

where α is a very large number, β is a very small number, and $\mathbf{I}_{n\times n}$ is a $(n \times n)$ identity matrix [34]. Equations (4.24) and (4.25) are often approximated by simply setting the initial covariance to a large number, $\mathbf{P}_o = 10^6 \cdot \mathbf{I}_{n\times n}$ and setting the initial state vector equal to zero, $\hat{\mathbf{x}}_0 = \mathbf{0}_{n\times 1}$. This thesis takes this approach.

4.4.2 Empirical Step Length Formula Comparison

This section will first introduce how the measurement noise covariance matrix was obtained. Next, the measurement matrices for each model are discussed. Lastly, the estimation of step length model coefficients for the models: SL_8, SL_9, and SL_{11} will be performed and then compared.

I. Step Length Measurement and Variance

The measurements used to solve for the step length function coefficients are not the North or East positions themselves, but the magnitude of the change in position (i.e. step length). The GPS receiver used also provides a covariance of the position measurements. This covariance does not represent the covariance of the relative position measurements. Finding the variance for this measurement is not obvious, so it is discussed here. For two random variables X and Y, the variance of the sum or difference of the random variables given by

$$\sigma^2_{X\pm Y} = \sigma^2_X + \sigma^2_Y - E[(X - \mu_X), (Y - \mu_Y)] \tag{4.26}$$

where $\sigma^2_{X\pm Y}$ is the variance of the summed or differenced measurements and $\{\sigma^2_X, \sigma^2_Y\}$ are the variance of the X and Y random variables, respectively. The term $E[(X-\mu_X), (Y-\mu_Y)]$, is the the covariance between the two random variables X and Y [71]. The GPS measurements can be assumed to be independent, therefore the covariance term will be zero and Equation (4.26) simplifies to Equation (4.27). Derivations of Equations (4.26-4.27) can be found in Appendix

B of [3].

$$\sigma^2_{X\pm Y} = \sigma^2_X + \sigma^2_Y \tag{4.27}$$

The true step length of the pedestrian is computed from the GPS measurements. It is the distance between two epochs, where a step was detected. It is computed with the Euclidean norm as

$$SL = \sqrt{(N_{ks} - N_{ks-1})^2 + (E_{ks} - E_{ks-1})^2} \tag{4.28}$$

where SL is the GPS step length, $\{N_{ks}, E_{ks}\}$ are the North and East positions at the current step detected, and $\{N_{ks-1}, E_{ks-1}\}$ are the North and East positions of the previous step detected. The variance of the step length measurement is found with Equation (4.27) and observing how the SL is computed. It is given by

$$\sigma^2_{SL,k} = (\sigma^2_{N,ks} + \sigma^2_{N,ks-1})^2 + (\sigma^2_{E,ks} + \sigma^2_{E,ks-1})^2 \tag{4.29}$$

where σ^2_{SL} is the variance of the SL measurement. For the OLS and WLS case the $(m \times m)$ diagonal measurement covariance matrix is given by Equation (4.30).

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{SL,1} & 0 & \dots & 0 \\ 0 & \sigma^2_{SL,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2_{SL,m} \end{bmatrix} \tag{4.30}$$

For RLS, the measurement covariance matrix at time t_k is $\mathbf{R}_k = \sigma^2_{SL,k}$.

II. Measurement Matrices and Vector

For OLS and WLS the measurements can be modeled by the equation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \quad \mathbf{v} \sim N(\mathbf{0}, \mathbf{R}) \tag{4.31}$$

where **H** is the measurement matrix and **v** is the measurement noise. The term **x** is the $(n \times 1)$ state vector which contains the model coefficients, and **y** is the $(m \times 1)$ measurement vector which contains the GPS step length measurements. The $(m \times m)$ measurement covariance matrix **R** is defined as

$$\mathbf{R} = E[\mathbf{v}\mathbf{v}^T] \tag{4.32}$$

as the noise for each measurement is assumed to be zero mean and independent [35]. For the RLS case the measurements are modeled as

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{4.33}$$

where the measurement covariance matrix $\mathbf{R}_k$ is defined as

$$\mathbf{R}_k = E[\mathbf{v}_k\mathbf{v}_k^T] \tag{4.34}$$

where the noise is again assumed to be zero mean and independent.

For SL_8, presented by Gusenbauer et al. [54], the $(m \times n)$ measurement matrix for OLS and WLS is

$$\mathbf{H} = \begin{bmatrix} 1 & f_1 \\ 1 & f_2 \\ \vdots & \vdots \\ 1 & f_m \end{bmatrix} \tag{4.35}$$

where the first column is a vector of ones and in the second column are the step frequencies (f) at the step epochs. For the RLS case, the $(1 \times n)$ measurement matrix is Equation (4.36).

$$\mathbf{H}_k = \begin{bmatrix} 1 & f_{ks} \end{bmatrix} \tag{4.36}$$

For SL_9 presented by Ladetto et al. [47], the $(m \times n)$ measurement matrix for OLS and WLS is

$$\mathbf{H} = \begin{bmatrix} 1 & f_1 & \sigma^2_{\mathrm{f},1} \\ 1 & f_2 & \sigma^2_{\mathrm{f},2} \\ \vdots & \vdots & \vdots \\ 1 & f_m & \sigma^2_{\mathrm{f,m}} \end{bmatrix} \tag{4.37}$$

where the first column is a vector of ones, in the second column are the step frequencies (f) for each step, and the third column is the variance of the magnitude of the specific force (σ_{f}) at each step. For the RLS case, the $(1 \times n)$ measurement matrix is Equation (4.38).

$$\mathbf{H}_k = \begin{bmatrix} 1 & f_{ks} & \sigma^2_{\mathrm{f,ks}} \end{bmatrix} \tag{4.38}$$

For SL_{11} presented by Groves et al. [22], the $(m \times n)$ measurement matrix for OLS and WLS is

$$\mathbf{H} = \begin{bmatrix} 1 & \sigma^2_{\mathrm{f},1} \\ 1 & \sigma^2_{\mathrm{f},2} \\ \vdots & \vdots \\ 1 & \sigma^2_{\mathrm{f,m}} \end{bmatrix} \tag{4.39}$$

where the first column is vector of ones and in the second column is the variance of the magnitude of the specific force (σ_{f}) at each step. For the RLS case, the $(1 \times n)$ measurement matrix is Equation (4.40).

$$\mathbf{H}_k = \begin{bmatrix} 1 & \sigma^2_{\mathrm{f,ks}} \end{bmatrix} \tag{4.40}$$

IV. Performance Evaluation

A short path was walked by the pedestrian as shown in Figure 4.10. Using OLS and WLS, the model coefficients for each of the different models were estimated along with their associated

confidence intervals. The general form of the covariance matrix is

$$\mathbf{P} = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \dots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_2^2 & \dots & \sigma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \dots & \sigma_{n,n}^2 \end{bmatrix} \tag{4.41}$$

where σ_1^2 is the variance of the first state estimate and $\sigma_{n,n}^2$ is the variance of the n^{th} state estimate. The standard deviations that form the confidence intervals are found by taking the square root of the diagonal entries of the error covariance matrix in Equation (4.41). The standard deviation of the first state estimate is $\sigma_1 = \sqrt{\mathbf{P}_{11}}$ the standard deviation of the second state estimate is $\sigma_2 = \sqrt{\mathbf{P}_{22}}$ and so on. The state can then be represented in the form, $\mathbf{x}_n = \hat{\mathbf{x}}_n \pm \sigma_n$. The estimates using OLS are shown in Table 4.3. The results shown for SL_8 show that both coefficients (K_1, K_2) were estimated with reasonable confidence. The results for SL_9 show that while coefficients $\{K_1, K_2\}$ are estimated reasonably well, the coefficient K_3 is not. The results for SL_{11} are similar, with coefficient K_1 estimated with reasonable certainty while coefficient K_2 is not. The coefficients from $\{SL_9, SL_{11}\}$ that are both poorly identifiable are the variance of specific force coefficients for each model. It may be possible to better identify these coefficients with longer data sets. The results from the WLS estimates are shown in Table 4.4. The results for $\{SL_8, SL_9, SL_{11}\}$ show that all of the coefficients for each model are estimated well. In order to better understand these results a second similar path was walked. Using the calibration from the first test dataset, the step length was then calculated using the second test dataset and compared to the true step length computed from the GPS position measurements. The root-means-square (RMS) error (defined in Section 5.7.1) is computed for each model. The results in Table 4.5 show that each of the models estimated with OLS perform slightly better or approximately the same as the WLS estimates. Also, the SL_{11} performs slightly better than the others. Note that these results are preliminary and should be treated as such. Future tests should estimate the models using a measurement source with higher accuracy. This could be achieved using the real-time kinematic (RTK) positioning technique

or other positioning techniques (e.g. precise point positioning (PPP), time differenced carrier phase (TDCP)). RTK-GPS can be used to obtain horizontal position measurements that are accurate to $\sim$2 (cm). Additionally, the data sets used should be much longer, to ensure a proper calibration and calculation of RMS error.

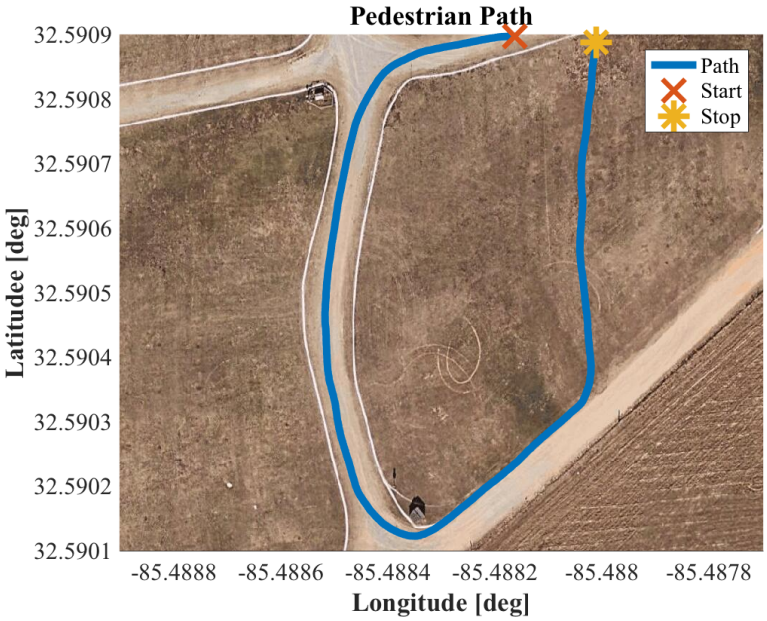

Figure 4.10: Pedestrian Path Used for Estimation of Step Length

Table 4.3: OLS Estimates

OLS Estimates	
Step Length Formula	K = estimate $\pm$ standard deviation
SL_8	$K_1 = 1.69154 \pm 0.11232$ (m) $K_2 = -0.44957 \pm 0.05949$ (Hz)
SL_9	$K_1 = 1.72391 \pm 0.11591$ (m) $K_2 = -0.45401 \pm 0.05959$ (Hz) $K_3 = -0.00190 \pm 0.00169$ $(\mathrm{m}^2/\mathrm{s}^4)$
SL_{11}	$K_1 = 0.85747 \pm 0.02483$ (m) $K_2 = -0.00104 \pm 0.00187$ $(\mathrm{m}^2/\mathrm{s}^4)$

Table 4.4: WLS Estimates

WLS Estimates	
Step Length Formula	K = estimate $\pm$ standard deviation
SL_8	$K_1 = 1.89943 \pm 0.02522$ (m) $K_2 = -0.55420 \pm 0.01324$ (Hz)
SL_9	$K_1 = 1.91497 \pm 0.02540$ (m) $K_2 = -0.55344 \pm 0.01324$ (Hz) $K_3 = -0.00184 \pm 0.00036$ (m^2/s^4)
SL_{11}	$K_1 = 0.86664 \pm 0.00398$ (m) $K_2 = -0.00201 \pm 0.00036$ (m^2/s^4)

Table 4.5: Step Length RMS Error

Step Length RMS Error		
Step Length Formula	Least Squares Method	RMS Error (m)
SL_8	OLS WLS	0.3173 0.3436
SL_9	OLS WLS	0.3307 0.3461
SL_{11}	OLS WLS	0.2771 0.2792

4.5 Heading Determination

Most often for PDR-SD, the heading is determined similar to a PDR-INS system, as the same sensors are used. Some systems only use a single gyro or a magnetometer mounted parallel to the torso and assume that it stays parallel. In addition, more advanced systems fuse a magnetometer with inertial sensors to estimate heading or attitude. In the following sections, the methods for determining heading with a gyro or magnetometer are detailed. It then will be shown how to fuse both the magnetometer and gyro measurements. Lasty, the subject of attitude heading reference systems will be briefly discussed.

4.5.1 Magnetometer Heading

The heading for a calibrated magnetometer is computed as

$$\psi_m = \arctan2(-m^b_{m,y}, m^b_{m,x}) \tag{4.42}$$

where $\{m^b_{m,y}, m^b_{m,x}\}$ are the calibrated measurements of the magnetic field in the body frame's y and x axes, respectively. The function $\arctan2$ is the four-quadrant inverse tangent. Note that this computation assumes that the magnetometer is level. If the magnetometer is not level, the measurements need to be compensated for pitch and roll effects [3, 23]. The expression for computing heading from a magnetometer becomes

$$\psi_m = \arctan2(-m^b_{m,y}\cos\phi + m^b_{m,z}\sin\phi, m^b_{m,x}\cos\theta + m^b_{m,y}\sin\phi\sin\theta + m^b_{m,z}\cos\phi\sin\theta) \tag{4.43}$$

where $\{\phi, \theta\}$ are the roll and pitch angles, respectively.

The magnetometer heading must also be corrected, as it is computed with respect to magnetic North and not true North. This is done by

$$\psi_n = \psi_m + \alpha_m \tag{4.44}$$

where ψ_n is the true heading and α_m is the declination angle. For further discussion of heading from a magnetometer, the reader is referred to [3, 21].

Many calibration algorithms exist to calibrate a magnetometer [72–74]. Most of these involve fitting either an ellipse or ellisoid to raw measurements. For further discussion of calibration algorithms and routines see [30, 74, 75].

4.5.2 Heading from a Single Gyroscope

Obtaining heading from a single gyroscope can be done several ways. The simplest method is to numerically integrate the yaw axis gyro to propagate the heading after initializing it from an external source. This method could be used in conjunction with an error-state Kalman filter. The method presented here uses a *kinematic estimator*. It is termed a kinematic estimator as the system model is the kinematic relationship of the sensor and not a dynamic model [76]. The following section will introduce the system model and then the measurement model will be presented.

I. System Model

The model of the gyroscope in Equation (3.3) is used for the kinematic estimator and can be put into state-space form as

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \tag{4.45}$$

where $\mathbf{x}_k$ is the (2×1) state vector, $\mathbf{F}$ is the (2×2) system matrix, and $\mathbf{G}$ is the (2×2) noise input matrix. The term $\mathbf{u}_k$ is the (1×1) input vector and $\mathbf{B}$ is the (2×1) input matrix. Equations (3.3) and (3.6) in the form of Equation (4.45) is

$$\begin{bmatrix} \dot{\psi} \\ \dot{b}_{g,z} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ b_{g,z} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{\omega}^b_{ib,z} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{w}_g \\ \mathrm{w}_{b_g} \end{bmatrix} \tag{4.46}$$

The elements of the state vector $\mathbf{x}$ are heading (ψ) and yaw axis gyro bias (ψ), $\big(\mathbf{x} = [\psi \quad b_{g,z}]^T\big)$. The element of the input vector ($\mathbf{u}$) is the measurement of z-axis angular rate ($\tilde{\omega}^b_{ib,z}$), $\big(\mathbf{u} = [\tilde{\omega}^b_{ib,z}]\big)$. The elements of the continuous noise vector ($\mathbf{w}$) are the gyro noise ($\mathbf{w}_g$) and the gyro

bias noise ($\mathbf{w}_{\mathbf{b}_g}$). The continuous process noise covariance is a (2×2) matrix defined as

$$\mathbf{Q} = \begin{bmatrix} \sigma_g^2 & 0 \\ 0 & \sigma_{\mathbf{b}_g}^2 \end{bmatrix} \tag{4.47}$$

where $\{\sigma_g^2, \sigma_{b_g}^2\}$ are the variance of the gyro noise and gyro bias, respectively.

In order to implement this KF in the discrete form, the system model must be discretized. The linear error equations in discrete form are

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{d,k-1} \tag{4.48}$$

where $\mathbf{\Phi}_{k-1}$ is the state transition matrix at time t_{k-1} and $\mathbf{\Gamma}_{k-1}$ is the discretized input matrix. The term $\mathbf{w}_{d,k-1}$ is the discrete system noise. The (2×2) state transition matrix is

$$\mathbf{\Phi} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \tag{4.49}$$

where Δt is the time interval between measurements. The discrete process noise covariance matrix is approximated with the equation

$$\mathbf{Q}_d = \mathbf{G}\mathbf{Q}\mathbf{G}^T\Delta t \tag{4.50}$$

where $\mathbf{G}$ is the continuous process noise input matrix and $\mathbf{Q}_d$ is the (2×2) discrete process noise matrix. Further discussion of the discretization process can be found in multiple texts [3, 19, 34, 41].

II. Measurement Model

The discrete-time linear measurement model is

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{4.51}$$

where $\mathbf{H}_k$ is the measurement matrix and $\mathbf{v}_k$ is the measurement noise at time t_k. Note the measurement model is linear and does not need to be linearized.

Unlike the PDR-INS approach, pseudo measurements cannot often be applied. Only ZARU can be applied if the pedestrian is standing stationary. Details on how to apply this update were previously provided in Section 3.3.6. Because of this, the (1×2) measurement matrix for a standalone gyro is typically empty as shown by Equation (4.52).

$$\mathbf{H} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{4.52}$$

4.5.3 Gyro + Magnetometer Heading

The heading determination methods in the two previous sections exhibit different error characteristics. This is because they utilize different sensors. Magnetic measurements are subject to interference from local ferrous and magnetic objects, but can provide an absolute heading measurement. Gyros can provide an accurate heading update measurement for short periods of time, but will drift over time. Together these sensors can be combined to remove the anomalies from the magnetometer measurements and reduce the heading drift from the gyros. Using the KF presented in Section 4.5.2 these measurements can be integrated as follows. When a magnetometer is available, the measurement vector is

$$\mathbf{y} = \begin{bmatrix} \psi_n \\ 0 \end{bmatrix} \tag{4.53}$$

where ψ_n is the magnetometer heading. The measurement matrix is Equation (4.54).

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.54}$$

This measurement update is often done with innovation filtering [3, 30, 77]. This is to remove outlier magnetometer measurements that have been corrupted by magnetic perturbations. Magnetic interference comes from ferrous material and magnetic fields (other than the Earth's)

that magnetometers are in close proximity to. Further discussion of this and other ways to mitigate magnetic interference are discussed by Abdulrahim [30].

4.5.4 Attitude Heading Reference Systems

Attitude heading and heading reference systems (AHRSs) are used to provide the user with orientation information (i.e. attitude). They often comprise of a tactical-grade IMU that includes an accelerometer, gyro, and magnetometer triads. AHRSs are used for applications that require higher heading determination accuracy than the two aforementioned methods. For an AHRS, attitude is computed in the same manner as it would be for an INS. A KF or KF variant is used to integrate the sensors and estimate the attitude or attitude errors. For further discussion and implementation of AHRSs see [3, 21, 41, 78, 79].

4.6 Navigation System Update

Once the steps, step length, and heading have been determined the navigation states can be updated. The system model for a PDR-SD system is given by

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} (SL_k) \cdot \cos\psi_k \\ (SL_k) \cdot \sin\psi_k \\ \Delta\psi_k \end{bmatrix} \tag{4.55}$$

where $\mathbf{x}_k$ is the state vector and ψ_k is the heading at time t_k. The term $\Delta\psi_k$ is the change in heading at time t_k. The elements of the state vector ($\mathbf{x}_k$) are North position (N_k) , East position (E_k), and heading (ψ_k), $\left(\mathbf{x} = [\mathrm{N} \quad \mathrm{E} \quad \psi]^T\right)$. This process in state-space format is governed by a linear difference equation in the form

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{\Gamma}\mathbf{u}_k \tag{4.56}$$

where $\mathbf{\Phi}_{k-1}$ is the (3×3) state transition matrix and $\mathbf{\Gamma}$ is the (3×3) input distribution matrix. The term $\mathbf{u}$ is the (3×1) input vector at time t_k. Equation (4.55) written in the form of Equation

(4.56) is

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k \tag{4.57}$$

$$\begin{bmatrix} N_k \\ E_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_{k-1} \\ E_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l_k \\ \Delta\psi_k \end{bmatrix} \tag{4.58}$$

where the elements of the input vector ($\mathbf{u}_k$) are step length (l_k) and the change in heading ($\Delta\psi_k$), $\left(\mathbf{u} = [l \quad \Delta\psi]^T\right)$.

Chapter 5

Particle Filtering for Indoor Positioning Systems

5.1 Introduction

This chapter introduces the overall framework for an indoor positioning system using a particle filter (PF). Subsequently, each of the PF steps are explained in detail. Next the current state of the art PF methods for pedestrian navigation are discussed and a new weight update method is presented. Lastly, a performance evaluation of the new and current methods is presented with both simulated and experimental data.

5.2 Estimation Framework

The estimation framework for a body worn system (i.e. no physical infrastructure) used in this thesis is described as follows. The sensor used is a tactical grade MEMS IMU, which is mounted on the pedestrian's foot. Thus the PDR method used is a PDR-INS approach with ZVUs and ZARUs [3]. The PDR-INS produces a 6 degree-of-freedom (DoF) navigation solution of the device (i.e. position, velocity, attitude). A 2D odometry measurement (i.e. change in position, change in heading) is extracted from the 6 DoF solution and used as the source of PDR and input into the PF along with the *a priori* map. The PF fuses the map and PDR, outputting a more accurate localization solution. The overall framework is portrayed in Figure 5.1. This framework was first demonstrated independently by Woodman and Harle [11], Klepal et al. [80], and Krach and Robertson [81] in 2008 [4]. Subsequently, it has been explored by others, some of which are [4, 8, 9, 12–18].

5.3 Particle Filtering

The PF was first introduced in 1993, as a nonlinear, non-Gaussian, Bayesian estimation technique that numerically approximates the Bayesian filter with a recursive state-space approach

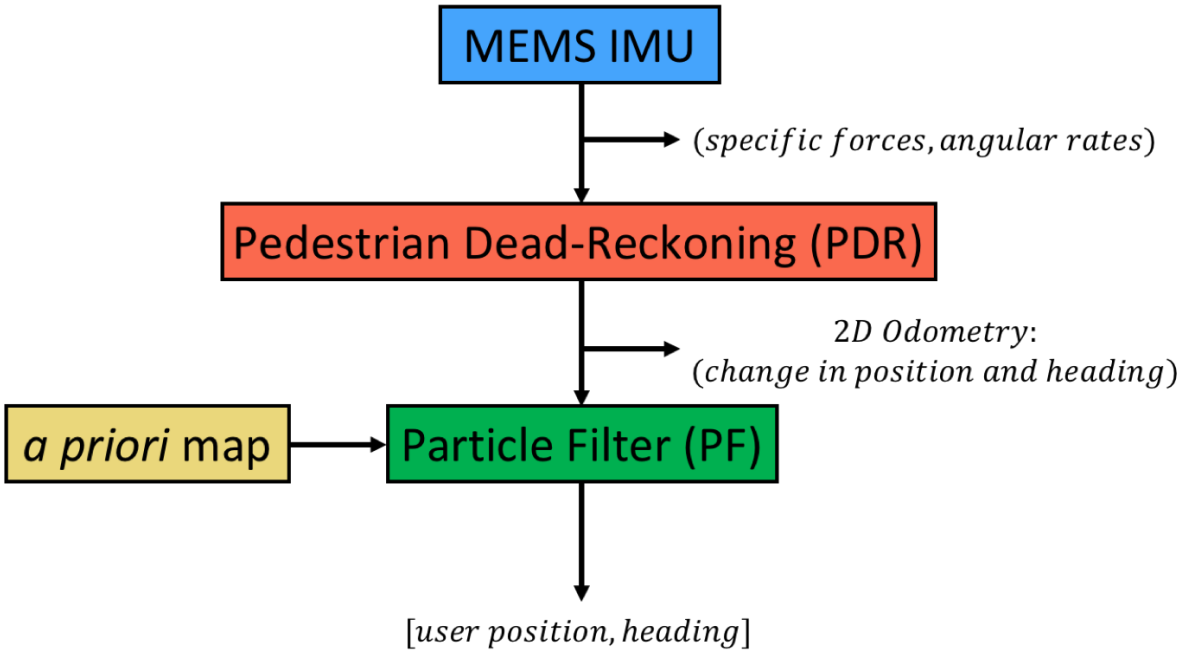

Figure 5.1: Indoor Positioning System Estimation Framework

[34, 82, 83]. The PF estimates the continuous posterior probability density function (PDF) with multiple finite random samples (particles), which differs from traditional filtering techniques like Kalman filtering [34]. For instance, Kalman filtering represents a bivariate Gaussian distribution with a mean and covariance, whereas the PF spreads a set of discrete state vectors (particles) to create a joint distribution to represent the distribution [3]. This is represented visually in Figure 5.2.

The PF is widely used for problems that are difficult to solve with conventional Kalman filtering techniques, which rely on Gaussian measurement distributions. The KF and its variants cannot easily incorporate map measurements as they are highly non-Gaussian in nature. The PF incorporates map information by placing direct constraints upon the position states which are known to be restricted by the building floorplan [12]. This is accomplished by eliminating particles that cross map boundaries and is referred to as *map-matching* [4, 9].

The PF consists of four steps: initialization, propagation, update, and resampling. These are illustrated in Figure 5.3. Each of the PF steps can be implemented in various ways, which will affect the overall performance of the PF. The main focus of this chapter is to describe

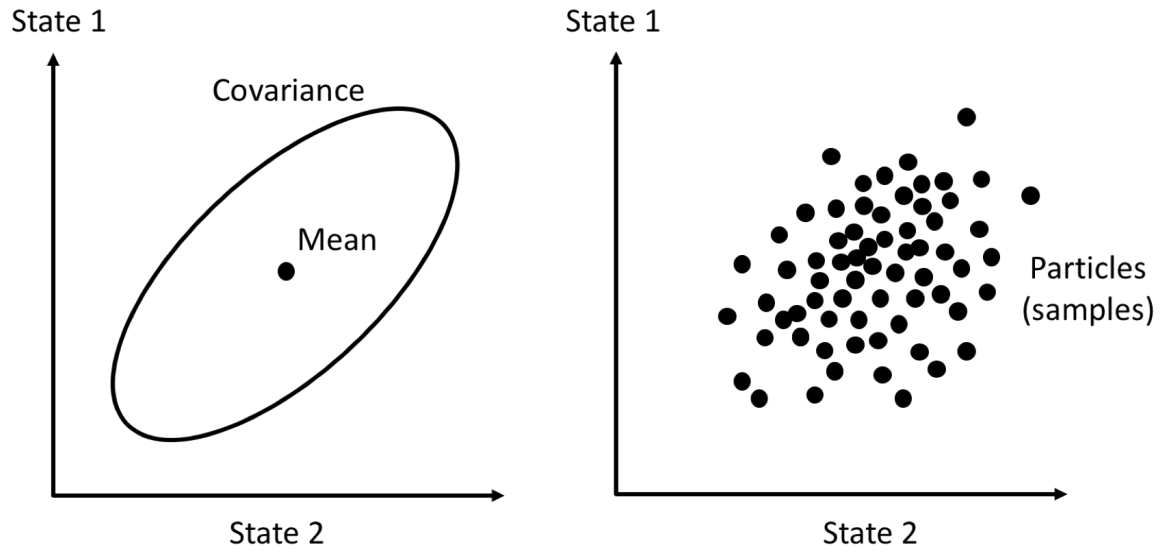

Figure 5.2: Two states represented with a bivariate Gaussian distribution using a mean and covariance (left), and a set of particles (right) [3]

various propagation and weight update methods. The methods differ based upon the selection of the motion model, measurement likelihood, and the proposal distribution.

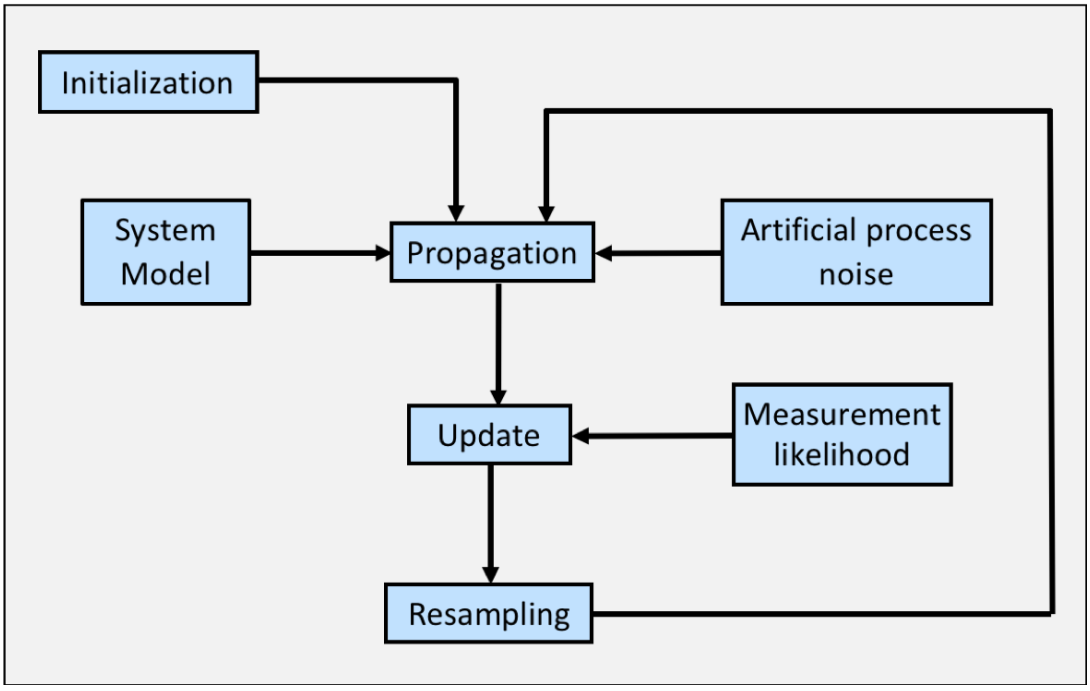

Figure 5.3: Particle Filter Steps [3]

5.3.1 Initialization

The first step is to initialize the PF with a fixed number of particles, N_p, with equal normalized weights. The PFs in this thesis used $N_p = 1000$ particles. This choice is based upon the standard practice of having a minimum of 1000 particles [34]. This is somewhat arbitrary, as currently no method exists to calculate how many particles should be used for a given state vector of a certain type and dimension. The approximation of the posterior density approaches the true posterior density as $N_p \rightarrow \infty$, but as the number of particles increases, so does the computational cost. In order to approximate the posterior density well, sample impoverishment (depletion) must be avoided. Sample impoverishment is when only a small number of particles (or particle) remains to contribute to the state estimate [83]. For this thesis, 1000 particles were found to be adequate at avoiding sample impoverishment while also not being too computationally expensive. Further study on the optimal number of particles is beyond the scope of this work but is further discussed in [83].

A standalone PDR navigation system requires an external reference to determine the initial position and heading states of the particles. An example of an external reference would be a WLAN to obtain the pedestrian's initial position and a magnetometer to determine initial heading. More advanced techniques assume that there is no knowledge of the pedestrian's pose within a building and adopt a multi-modal approach. This is accomplished by spreading particles over the entire building or area of interest [10, 11]. The initial spread of particles corresponds to the confidence in the initial state estimates.

In this thesis, the initial position and heading are found using GPS and a magnetometer, respectively. The initial particles are spread with the distribution $N(\mathbf{x}|\mathbf{x}_o, \Sigma_{\mathbf{x}_o})$, where $N(\mathbf{x}|\boldsymbol{\mu}, \mathbf{P})$ is a Normal (Gaussian) distribution of $\mathbf{x}$, with a mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$. The term $\Sigma_{\mathbf{x}_o}$ is a (3×3) diagonal initial covariance matrix. The elements of the initial state vector $\mathbf{x}_o$ are North position (N_o), East position (E_o), and heading (ψ_o). The standard deviations chosen to represent the uncertainty of the initial position coordinates and heading are $\{\sigma_{N_o} = 0.6$ m, $\sigma_{E_o} = 0.6$ m, $\sigma_{\psi_o} = 0.5^\circ\}$.

5.3.2 Propagation with PDR

Pedestrian dead-reckoning is the most popular method used to propagate particles. Each time the pedestrian steps or changes direction, the particle states are propagated with the change in position and change in heading measurements. Artificial process noise is added to particle states to prevent particle degeneracy and sample impoverishment by spreading the particles. This *ad-hoc* process is called *roughening* or *jittering* and is used to increase the particle diversity [8, 34]. This ensures that there are distinct particles (i.e. not identical) to approximate the posterior distribution, thus the PF is a type of sequential Monte Carlo algorithm [3, 34, 71]. For the PFs in this work, roughening is performed each time the particles are propagated. Note that this may be not necessary, but the analysis of the frequency of roughening is beyond the scope of this work. The following sections outline the system models used for propagation for pedestrian navigation and the relationship between them.

I. Delta Positions System Model

The conventional 2D dead-reckoning equations in the North-East-Down (NED) frame that form the system model are

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} \Delta x_k \cos\psi_k - \Delta y_k \sin\psi_k \\ \Delta x_k \sin\psi_k + \Delta y_k \cos\psi_k \\ \psi_k + \Delta\psi_k \end{bmatrix} \tag{5.1}$$

where the elements of the (3×1) state vector $\mathbf{x}_k$ are North position (N_k) , East position (E_k), and heading (ψ_k), $\left(\mathbf{x} = [\mathrm{N} \quad \mathrm{E} \quad \psi]^T\right)$. These equations show that the changes in position $(\Delta x_k, \Delta y_k)$ of the body are rotated into the local navigation frame (n) and then added to the previous state vector $\mathbf{x}_{k-1}$ to obtain the current state vector $\mathbf{x}_k$. This rotation into the local navigation frame is shown in Equations (5.2-5.4).

$$\begin{bmatrix} \Delta\mathrm{N} \\ \Delta\mathrm{E} \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{5.2}$$

$$\begin{bmatrix} \Delta \mathrm{N} \\ \Delta \mathrm{E} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{5.3}$$

$$\begin{bmatrix} \Delta \mathrm{N} \\ \Delta \mathrm{E} \end{bmatrix} = \begin{bmatrix} \Delta x \cos\psi - \Delta y \sin\psi \\ \Delta x \sin\psi + \Delta y \cos\psi \end{bmatrix} \tag{5.4}$$

The matrix $\mathbf{C}_b^n$ is the 2D CTM from the body frame (b) to the navigation frame (n). This model assumes motion only occurs in the 2D plane (i.e. planar motion), so it is commonly referred to as a planar model [19, 41, 84]. These equations are typically used when extracting a 2D odometry solution from a 6 DoF INS solution.

When the PDR is used to propagate particles, discrete samples of artificial noise ($\mathrm{w}_{\Delta x}, \mathrm{w}_{\Delta y}, \mathrm{w}_{\Delta \psi}$) are added to the longitudinal change in position (Δx), lateral change in position (Δy), and the change in heading ($\Delta \psi_k$) respectively, at each time step (t_k). This is done to create a distribution of state estimates as in Equation (5.5).

$$\mathbf{x}_k^i = \mathbf{x}_{k-1}^i + \begin{bmatrix} (\Delta x_k + \mathrm{w}_{\Delta x}^i)\cos\psi_k - (\Delta y_k + \mathrm{w}_{\Delta y}^i)\sin\psi_k \\ (\Delta x_k + \mathrm{w}_{\Delta x}^i)\sin\psi_k + (\Delta y_k + \mathrm{w}_{\Delta y}^i)\cos\psi_k \\ \psi_k + (\Delta \psi_k + \mathrm{w}_{\Delta \psi}^i) \end{bmatrix} \tag{5.5}$$

The vector $\mathbf{x}_k^i$ is the state vector of the i^{th} particle at time t_k. The system model in Equations (5.1) and (5.5) allows for the pedestrian to move in any planar direction. The artificial process noise is sampled from the following distributions $\{\mathrm{w}_{\Delta x} \sim N(0, \sigma_{\Delta x}^2), \mathrm{w}_{\Delta y} \sim N(0, \sigma_{\Delta y}^2), \mathrm{w}_{\Delta \psi} \sim N(0, \sigma_{\Delta \psi}^2)\}$ where $N(\mu, \sigma^2)$ is a Normal (Gaussian) distribution with a mean μ and a variance of σ^2. The standard deviations of $\{\Delta x, \Delta y, \Delta \psi_k\}$ were selected empirically to be $\{\sigma_{\Delta x}^2 = 0.1$ m, $\sigma_{\Delta y}^2 = 0.1$ m, $\sigma_{\Delta \psi}^2 = 0.5^\circ\}$, respectively. This process in state-space format is represented by a linear difference equation of the form

$$\mathbf{x}_k^i = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1}^i + \mathbf{\Gamma}\mathbf{u}_k + \mathbf{\Upsilon}\mathbf{w}_{d,k}^i \tag{5.6}$$

where $\mathbf{\Phi}_{k-1}$ is the (3×3) state transition matrix at time t_{k-1}, $\mathbf{\Gamma}$ is the (3×3) input distribution matrix, and $\mathbf{\Upsilon}$ is the (3×3) discrete noise distribution matrix. The term $\mathbf{u}_k$ is the input vector at

time t_k and $\mathbf{w}_{d,k}^i$ is the discrete noise vector of the i^{th} particle at time t_k. Equation (5.5) written in the form of Equation (5.6) is

$$\mathbf{x}_k^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1}^i + \begin{bmatrix} \cos\psi_k & -\sin\psi_k & 0 \\ \sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u}_k + \begin{bmatrix} \cos\psi_k & -\sin\psi_k & 0 \\ \sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{w}_{d,k}^i \tag{5.7}$$

$$\begin{bmatrix} N_k^i \\ E_k^i \\ \psi_k^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_{k-1}^i \\ E_{k-1}^i \\ \psi_{k-1}^i \end{bmatrix} + \begin{bmatrix} \cos\psi_k & -\sin\psi_k & 0 \\ \sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta\psi_k \end{bmatrix} + \begin{bmatrix} \cos\psi_k & -\sin\psi_k & 0 \\ \sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{w}_{\Delta x_k}^i \\ \mathrm{w}_{\Delta y_k}^i \\ \mathrm{w}_{\Delta\psi_k}^i \end{bmatrix} \tag{5.8}$$

where the elements of the input vector ($\mathbf{u}_k$) are the longitudinal change in position (Δx), lateral change in position (Δy), and the change in heading ($\Delta\psi_k$), $\left(\mathbf{u} = [\Delta x \;\; \Delta y \;\; \Delta\psi]^T\right)$. The elements of the ($3 \times 1$) discrete noise vector ($\mathbf{w}_{d,k}$) are the Δx noise ($\mathrm{w}_{\Delta x_k}$), Δy noise ($\mathrm{w}_{\Delta y_k}$), and $\Delta\psi$ noise ($\mathrm{w}_{\Delta\psi_k}$), $\left(\mathbf{w}_d = [\mathrm{w}_{\Delta x} \;\; \mathrm{w}_{\Delta y} \;\; \mathrm{w}_{\Delta\psi}]^T\right)$. The PDF of the system model in Equation (5.5) is

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1} + \begin{bmatrix} \Delta N_k \\ \Delta E_k \\ \Delta\psi_k \end{bmatrix}, \Sigma_{\mathbf{x}}\right) \tag{5.9}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1} + \begin{bmatrix} \Delta x_k\cos\psi_k - \Delta y_k\sin\psi_k \\ \Delta x_k\sin\psi_k + \Delta y_k\cos\psi_k \\ \Delta\psi_k \end{bmatrix}, \begin{bmatrix} \sigma_{\Delta x}^2 & & O \\ & \sigma_{\Delta y}^2 & \\ O & & \sigma_{\Delta\psi}^2 \end{bmatrix}\right) \tag{5.10}$$

where $N(\mathbf{x}|\boldsymbol{\mu}, \mathbf{P})$ is a Normal (Gaussian) distribution of $\mathbf{x}$ with a mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$. The term $\Sigma_{\mathbf{x}}$ is a (3×3) diagonal covariance matrix and the term O indicates that all of the off-diagonal terms are zero. The PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is referred to as the motion model. Note that this name is a bit of a misnomer as it may cause the reader to mistake it as the system model. Note that $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is also known as the transition density and the prior density by other authors [34–36]. Though these are more appropriate names, the pedestrian navigation community commonly refers to the PDF of the system model as the motion model, thus this thesis follows that precedent [9, 11, 12].

II. Distance and Heading System Model

The distance and heading system model is a simplification of the previous model. Instead of incorporating the longitudinal and lateral changes in position (Δx, Δy), the distance traveled (l) is used. Distance traveled is defined as the Euclidean distance from the previous position to the current position and is either a step or stride length depending upon the PDR method used. If the PDR-INS method is used, the distance traveled is the stride length. It is found by computing the Euclidean norm of the delta positions ($l = \sqrt{(\Delta x)^2 + (\Delta y)^2}$). If the PDR-SD method is used the distance traveled is the step length. The distance and heading system model is given by Equation (5.11).

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} (l_k) \cdot \cos \psi_k \\ (l_k) \cdot \sin \psi_k \\ \Delta \psi_k \end{bmatrix} \tag{5.11}$$

When PDR is used to propagate particles within a PF, discrete samples of artificial noise ($\mathrm{w}_l, \mathrm{w}_{\Delta\psi}$) are added to the distance traveled (l_k) and change in heading (ψ_k), respectively at time t_k to create a distribution of state estimates. This is seen in Equation (5.12).

$$\mathbf{x}_k^i = \mathbf{x}_{k-1}^i + \begin{bmatrix} (l_k + \mathrm{w}_l^i) \cdot \cos \psi_k \\ (l_k + \mathrm{w}_l^i) \cdot \sin \psi_k \\ (\Delta \psi_k + \mathrm{w}_{\Delta\psi}^i) \end{bmatrix} \tag{5.12}$$

The vector $\mathbf{x}_k^i$ is the state vector of the i^{th} particle at time t_k. The system model in Equations (5.11-5.12) assumes the pedestrian *only* moves forward and does not account for sideways or backwards motion. The artificial process noise is sampled from the distributions $\{\mathrm{w}_l \sim N(0, \sigma_l^2), \mathrm{w}_{\Delta\psi} \sim N(0, \sigma_{\Delta\psi}^2)\}$ where $N(\mu, \sigma^2)$ is a Normal (Gaussian) distribution with a mean μ and a variance of σ^2. The standard deviations of the step length and heading were selected empirically as $\{\sigma_l^2 = 0.14$ m, $\sigma_{\Delta\psi}^2 = 0.5^\circ\}$, respectively. This process in state-space format is governed by a linear difference equation of the form

$$\mathbf{x}_k^i = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1}^i + \mathbf{\Gamma}\mathbf{u}_k + \mathbf{\Upsilon}\mathbf{w}_{d,k}^i \tag{5.13}$$

where $\mathbf{\Phi}_{k-1}$ is the (3×3) state transition matrix at time t_{k-1}, $\mathbf{\Gamma}$ is the input distribution matrix, and $\mathbf{\Upsilon}$ is the (3×3) discrete noise distribution matrix. The term $\mathbf{u}_k$ is the input vector at time t_k and $\mathbf{w}_{d,k}^i$ is the discrete noise vector of the i^{th} particle at time t_k. Equation (5.11) written in the form of Equation (5.13) is

$$\mathbf{x}_k^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1}^i + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \mathbf{w}_{d,k}^i \tag{5.14}$$

$$\begin{bmatrix} N_k \\ E_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_{k-1} \\ E_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l_k \\ \Delta\psi_k \end{bmatrix} + \begin{bmatrix} \cos\psi_k & 0 \\ \sin\psi_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{w}_{l_k}^i \\ \mathrm{w}_{\Delta\psi_k}^i \end{bmatrix} \tag{5.15}$$

where the elements of the input vector ($\mathbf{u}_k$) are the distance traveled (l_k) and the change in heading ($\Delta\psi_k$), $\left(\mathbf{u} = [l \quad \Delta\psi]^T\right)$. The elements of the (3×1) discrete noise vector ($\mathbf{w}_{d,k}$) are the l_k noise (w_{l_k}) and $\Delta\psi_k$ noise ($\mathrm{w}_{\Delta\psi_k}$), $\left(\mathbf{w}_d = [\mathrm{w}_l \ \ \mathrm{w}_{\Delta\psi}]^T\right)$. The PDF of the system model in Equation (5.12) is

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1} + \begin{bmatrix} \Delta N_k \\ \Delta E_k \\ \Delta\psi_k \end{bmatrix}, \Sigma_{\mathbf{x}}\right) \tag{5.16}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1} + \begin{bmatrix} l_k \cdot \cos\psi_k \\ l_k \cdot \sin\psi_k \\ \Delta\psi_k \end{bmatrix}, \begin{bmatrix} \sigma_l^2 \cdot \mathbf{I}_2 & O \\ O & \sigma_{\Delta\psi}^2 \end{bmatrix}\right) \tag{5.17}$$

where $N(\mathbf{x}|\boldsymbol{\mu}, \mathbf{P})$ is a Normal (Gaussian) distribution of $\mathbf{x}$ with a mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$. The term $\Sigma_{\mathbf{x}}$ is a (3×3) diagonal covariance matrix and $\mathbf{I}_2$ is a (2×2) identity matrix. The term O indicates that all of the off-diagonal terms are zero. The elements of the state vector $\mathbf{x}_k$ are North position (N_k) , East position (E_k), and heading (ψ_k) $\left(\mathbf{x} = [\mathrm{N} \quad \mathrm{E} \quad \psi]^T\right)$. The PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is again referred to as the motion model.

III. Relationship of PDR Models

The model used in this chapter was the delta positions model in Equation (5.8). The simpler distance and heading model could have also been used, as the pedestrian in tests only exhibited

forward motion. The distance traveled and the delta positions are related by the Pythagorean theorem, as are the standard deviations. The Euclidean norm of the delta positions and their standard deviations are shown in Equations (5.18) and (5.19), respectively.

$$l = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{5.18}$$

$$\sigma_l = \sqrt{(\sigma_{\Delta x})^2 + (\sigma_{\Delta y})^2} \tag{5.19}$$

5.3.3 Propagation with PDR + Map

Newer PF methods include map information in the propagation step [9]. Map information is quantified by creating what is referred to as the angular PDF. The angular PDF weights the particles based on their proximity to the boundaries of the map. Distances shorter than an average distance traveled (l) are given small non-zero weights, whereas distances as they extend longer than the distance traveled are assigned increasingly larger weights. This heavily weights the particles toward open space directions and gives little to no weight in the direction of map boundaries. Mathematically this method is implemented using a truncated PDF $N_{[a,b]}(\psi_k|\psi_{k-1} + \Delta\psi_k, \sigma^2_{\Delta\psi})$ with the truncation interval [a, b] calculated with the angular PDF. Truncation reduces the uncertainty (i.e. variance) of the heading distribution and puts more probability in open space directions. Therefore, when the distribution is sampled the particles will spread towards open space. The result is less particles colliding with walls which will reduce the need for resampling.

5.3.4 Update

The weight update depends upon the motion model $p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})$, measurement likelihood $p(\mathrm{y}_k|\mathbf{x}^i_{k-1})$, and proposal distribution $q(\mathbf{x}^i_k|\mathbf{x}_{0:k-1}, \mathrm{y}_{1:k})$. For a Markov process, the proposal distribution only depends upon the current state ($\mathbf{x}^i_{k-1}$) and measurement (y_k) and simplifies to $q(\mathbf{x}^i_k|\mathbf{x}_{k-1}, \mathrm{y}_k)$ [12, 34, 82]. From Bayes' rule and the law of total probability, the weight update equation is

formed for this Markov process is

$$w_k^i = \frac{p(\mathrm{y}_k|\mathbf{x}_{k-1}^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}, \mathrm{y}_k)} w_{k-1}^i \tag{5.20}$$

where w_k^i is the weight of the i^{th} particle and y_k is the map measurement at time t_k. The particle weights (w_k^i) should be interpreted to be the probability of the states each particle represents. After the weight update the particle weights are renormalized with Equation (5.21).

$$w_k^i = \frac{w_k^i}{\sum_{j=1}^{N_p} w_k^j} \tag{5.21}$$

The posterior probability density function is approximated as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{5.22}$$

where w_k^i is defined by Equation (5.20) and $\delta(\cdot)$ is a Dirac delta function. Note that as $N_p \to \infty$ the approximation in Equation (5.22) approaches the true posterior density function $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ [82]. The states are estimated by computing the expected valued (mean) after the update at every epoch (t_k) in Equation (5.23).

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k] \approx \sum_{i=1}^{N_p} w_k^i \mathbf{x}_k^i \tag{5.23}$$

The measurement likelihood $p(\mathrm{y}_k|\mathbf{x}_{k-1}^i)$ is also known as the likelihood or likelihood function [34,36,82]. The proposal distribution $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}, \mathrm{y}_k)$ is also known as the importance distribution, importance density, and proposal density [12,34,36,82]. Note that calling $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}, \mathrm{y}_k)$ the proposal distribution is a misnomer as it is a density and not a distribution. Even though there are more correct names than proposal distribution, this work will follow the precedent of previous works and call it as such [9, 13].

Two methods used to implement the weight update are presented below.

I. Bootstrap Filter

A bootstrap filter (BF) makes the convenient choice of the motion model as the proposal distribution. The advantage of this is that the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and proposal distribution $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathrm{y}_k)$ terms cancel, and the weight update equation becomes Equation (5.24).

$$w_k^i = p(\mathrm{y}_k|\mathbf{x}_k^i)w_{k-1}^i \tag{5.24}$$

The motion model is a suboptimal choice for a proposal distribution as it does not take the current measurement (y_k) into account at time t_k [9, 34, 82]. For a non-infrastructure-based system with no absolute position measurement, the measurement likelihood is defined with the binary weighting scheme described in Equation (5.25).

$$p(\mathrm{y}_k|\mathbf{x}_k^i) = \begin{cases} \epsilon, & \text{particle crossing} \\ 1-\epsilon, & \text{otherwise} \end{cases} \tag{5.25}$$

The particles that 'collide' with a wall (i.e. the relative position vector between the *a priori* and *a posteriori* particle passes through a wall) are said to 'die' and given small or zero weights (ϵ). The other particles are said to 'live' and retain their weights, and then the particle weights are renormalized. This binary weight update is a popular choice to perform map-matching as it only relies on the measurement likelihood. In this thesis, the value $\epsilon = 0$ was assigned. The technique used to check for wall crossings is based upon work by Perttula et al. [13]. It is presented in full in Section 5.4.

II. Map-based proposal distribution, motion model, and measurement likelihood

Newer methods propose including map information into the weight update equation, namely the proposal distribution, motion model, measurement likelihood or a combination thereof [9, 85]. The map information is included in the weight update equation using the angular PDF. This creates a two-step weight update. The particles are first re-weighted using the angular PDF. Next, map-matching is performed with the binary weight update and the particle weights renormalized. The performance of the PF is affected by which parts of the weight update

equation include the angular PDF. Depending on the which parts are chosen, the particles will be either weighted towards open space or in the direction of the map. The more accurate the PDR, the less of an effect the angular PDF has as less particles will collide with walls.

5.3.5 Resampling

Resampling is required to prevent sample impoverishment and particle degeneracy. Sample impoverishment occurs when the effective sample size N_{eff} reduces and approaches $N_{eff} \approx 1$ as calculated by Equation (5.26).

$$N_{eff} = \frac{1}{\sum_{j=1}^{N_p} w_k^j} \tag{5.26}$$

N_{eff} is reduced when particle weights are set to zero after a wall crossing has been detected.

Particle degeneracy occurs when the normalized weights tend to concentrate to one or few particles after several recursive steps. This lack of particle diversity is also indicated by $N_{eff} \approx 1$. If N_{eff} becomes too small the PF will no longer approximate the posterior distribution accurately. When almost all of the particles are equally weighted then $N_{eff} \approx N_p$. A large N_{eff} is desired for good PF performance. To produce this result, particles are resampled when the constraint

$$N_{eff} < 0.5 \cdot N_p \tag{5.27}$$

is met, where 0.5 is the resample percent. Note that a large N_{eff} alone does not ensure that the particles are diverse, as they may be identical due to previous resampling steps [34]. This is why particle filters, especially bootstrap filters, implement both resampling and roughening (jittering) to ensure good performance.

The basic idea of resampling is to eliminate particles with small weights (zero in this case) and duplicate the particles that have retained their weights. It is implemented by sampling with replacement from $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p}$ to create N_p equally weighted particles $\{\mathbf{x}_k^j, w_k^j = 1/N_p\}_{j=1}^{N_p}$. The number of particles remains unchanged from resampling. Several resampling schemes exist, for this work the systematic resampling method was used. It was chosen as it is the most popular method and is straightforward to implement. Resampling and other resampling methods are further discussed in [34,82,83,86]. The steps for systematic resampling are shown

below with pseudocode provided in Algorithm 2 and is based upon pseudocode in [34, 82].

Systematic Resampling

1. Compute the cumulative distribution function, where c_i is the i^{th} cumulative sum element of the weights: $c_i = \sum_{i=1}^{N_p} w_k^i$.
2. Draw a uniform sample (u_1) on the interval $(0, 1/N_p]$: $u_1 \sim uniform(0, 1/N_p]$
3. For $j = 1 : N_p$ compute:

$$u_j = u_1 + (1/N_p) \cdot (j - 1)$$

4. Set $i = 1$. Perform next steps for $j = 1 : N_p$.

   if $u_j < c_i$

   $\mathbf{x}_k^j = \mathbf{x}_k^i$

   $w_k^j = w_k^i$

   $j = j + 1$

   else

   $i = i + 1$

   end

Algorithm 2: Systematic Resampling

Input: $\mathbf{x}_k^i, w_k^i$
Output: $\mathbf{x}_k^j, w_k^j$
Initialization: $c_1 = 0$
for $i = 2 : N_p$ **do**
    $c_i = c_{i-1} + w_k^i$
end
$i = 1$
$u_1 = rand/N_p$
for $j = 1 : N_p$ **do**
    $u_j = u_1 + (1/N_p) \cdot (j - 1)$
    while $u_j > c_i$ **do**
        if $(i + 1) > length(c)$ **then**
            break
        else
            $i = i + 1$
        end
    end
    $\mathbf{x}_k^j = \mathbf{x}_k^i$
    $w_k^j = 1/N_p$
end
%Note the function $rand$ returns a single uniformly distributed random sample in the interval (0,1].

5.4 Particle Inbounds Check

The particle inbounds check is a very important piece of the update step as it checks to see if the particle at the current time is within the map boundary. In other words, has the particle crossed a map boundary (e.g. wall) when it was propagated to a new position. As previously mentioned, the technique used to check for wall crossings is based upon work by Perttula et al. [13]. This method checks if the vector from the previous to the current particle position intersects a map boundary vector. The particles' 2D positions are each represented with (2×1) column vectors. The vector $\mathbf{r}_a$ represents the particle position at the previous epoch t_{k-1} and $\mathbf{r}_b$ is the particle position at the current epoch t_k. Likewise, the 2D map boundary positions are represented with (2×1) column vectors. The vector $\mathbf{r}_c$ is the node that is the start map boundary and the vector $\mathbf{r}_d$ is the node that ends the map boundary. The vector from the previous position ($\mathbf{r}_a$) to the current particle position ($\mathbf{r}_b$) is $\mathbf{r}_{b/a}$ and is defined in Equation (5.28), where b/a is read as: b from the perspective of a.

$$\mathbf{r}_{b/a} = \mathbf{r}_b - \mathbf{r}_a \tag{5.28}$$

The vector from the start of the map boundary ($\mathbf{r}_c$) to the end of the map boundary ($\mathbf{r}_d$) is $\mathbf{r}_{d/c}$ and is defined in Equation (5.29).

$$\mathbf{r}_{d/c} = \mathbf{r}_d - \mathbf{r}_c \tag{5.29}$$

The particles inbounds check is presented graphically in Figure 5.4, where the vectors $\mathbf{r}_a$, $\mathbf{r}_b$, $\mathbf{r}_c$, $\mathbf{r}_d$, $\mathbf{r}_{b/a}$, and $\mathbf{r}_{d/c}$ are illustrated. Additionally, the point of intersection for each vector $\mathbf{r}_{b/a}$ and $\mathbf{r}_{d/c}$ is defined as $\mathbf{p}_{ab}$ and $\mathbf{p}_{cd}$, respectively. The equations that define these points are

$$\mathbf{p}_{ab} = \mathbf{r}_a + \mathrm{q}_1 \mathbf{r}_{b/a}, \quad 0 \leq \mathrm{q}_1 \leq 1 \tag{5.30}$$

$$\mathbf{p}_{cd} = \mathbf{r}_c + \mathrm{q}_2 \mathbf{r}_{d/c}, \quad 0 \leq \mathrm{q}_2 \leq 1 \tag{5.31}$$

where q_1 and q_2 are scalar coefficients.

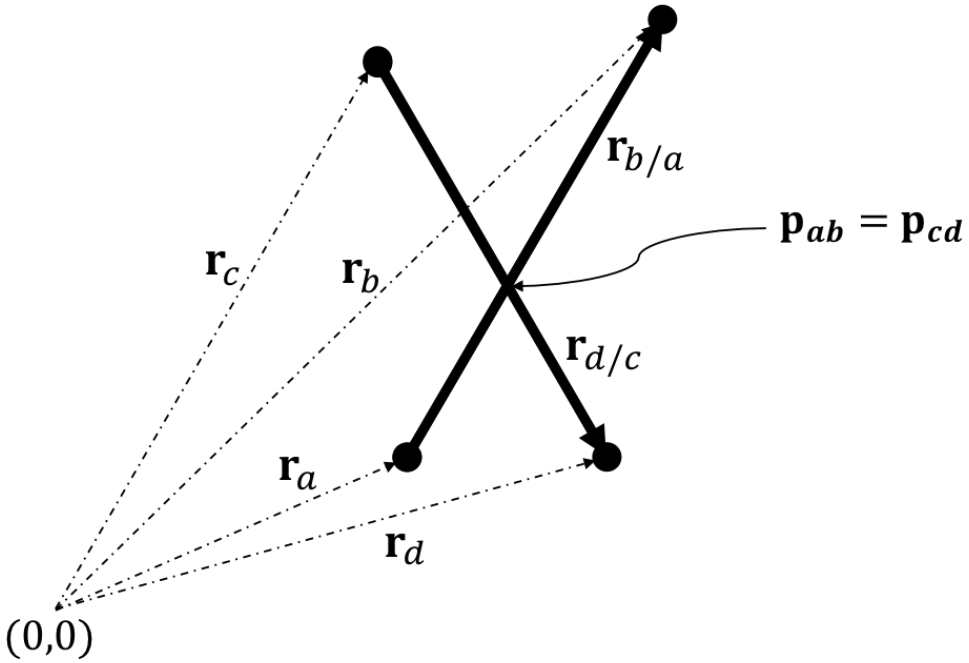

Figure 5.4: Particle Inbounds Check [13]

In order to check to see if an intersection exists, the scalar coefficients q_1 and q_2 must be calculated. First the Equations (5.30) and (5.31) are set equal. This can be done as the points $\mathbf{p}_{ab}$ and $\mathbf{p}_{cd}$ are equivalent if there is an intersection. The resultant equation is shown below.

$$\mathbf{r}_a + \mathrm{q}_1\mathbf{r}_{b/a} = \mathbf{r}_c + \mathrm{q}_2\mathbf{r}_{d/c}. \tag{5.32}$$

Next, Equation (5.32) is rearranged by placing the terms that contain the scalar coefficients q_1 and q_2 on the left-hand side and the other terms on the right-hand side. This is shown in Equation (5.33).

$$\mathrm{q}_1\mathbf{r}_{b/a} - \mathrm{q}_2\mathbf{r}_{d/c} = \mathbf{r}_c - \mathbf{r}_a \tag{5.33}$$

This equation can be written in matrix form by defining a new column vector $\mathbf{q} = [\mathrm{q}_1, \mathrm{q}_2]^T$ and a (2×2) matrix $\mathbf{M} = [\mathbf{r}_{b/a}, -\mathbf{r}_{d/c}]$. These substituted into Equation (5.33) result in Equation (5.34).

$$\mathbf{M}\mathbf{q} = \mathbf{r}_c - \mathbf{r}_a. \tag{5.34}$$

The column vector $\mathbf{q}$ is solved for by multiplying Equation (5.34) by the matrix inverse $\mathbf{M}^{-1}$ resulting in Equation (5.35).

$$\mathbf{q} = \mathbf{M}^{-1}(\mathbf{r}_c - \mathbf{r}_a) \tag{5.35}$$

There is an intersection of the particle path ($\mathbf{r}_{b/a}$) and the map boundary ($\mathbf{r}_{d/c}$) if $0 \leq \mathrm{q}_1 \leq 1$ and $0 \leq \mathrm{q}_2 \leq 1$. If these two conditions are met then the particle is said to be out of bounds. If not, then the particle is inbounds and no wall has been crossed. The pseudocode for the particle inbounds check is represented in Algorithm 3.

Algorithm 3: Particle Inbounds Check

Input: $\mathbf{r}_{k-1}, \mathbf{r}_k$, mapStart, mapStop
Output: inbounds
Initialization: $\mathbf{r}_a = \mathbf{r}_{k-1}, \mathbf{r}_b = \mathbf{r}_k, \mathbf{r}_c =$ mapStart, $\mathbf{r}_d =$ mapStop
% Calculate the position and map vectors
$\mathbf{r}_{b/a} = \mathbf{r}_b - \mathbf{r}_a$
$\mathbf{r}_{d/c} = \mathbf{r}_d - \mathbf{r}_c$
% Calculate **M** and **q**
$\mathbf{M} = [\mathbf{r}_{b/a}, -\mathbf{r}_{d/c}]$
$\mathbf{q} = \mathbf{M}^{-1}(\mathbf{r}_c - \mathbf{r}_a)$
% Conditions for Intersect
C1 $= (0 \leq \mathbf{q}(1) \leq 1)$ % Condition 1
C2 $= (0 \leq \mathbf{q}(2) \leq 1)$ % Condition 2
intersect $= logical(C1 \quad \& \quad C2)$
% Condition if particle is inbounds
if intersect $=$ true **then**
    inbounds $=$ false
else
    inbounds $=$ true
end

5.5 Particle Filters

Different PFs result from what propagation and weight update methods are selected in the design. In the work by Nurminen et al., five PFs are compared with four of those including map information via the angular PDF [9]. The filter PF1 in his work includes the angular PDF solely in the proposal distribution. PF2 includes map information in both the proposal distribution and the motion model. PF3 includes map information in the proposal distribution and the measurement likelihood. Lastly, PFw includes the angular PDF only in the measurement likelihood. PF1 and PF2 assign more probability towards open space while PF3 and PFw assign more probability in the direction of the map boundaries. The PFc and PFw are the filters of interest and are presented below. PFw is the only filter that incorporates the angular PDF to

be evaluated in this work due to its performance presented by Nurminen et al. PFw was shown to perform well when the PDR closely matched the map in corridor and room situations while using $N_p = 1000$ particles and a change of heading measurement perturbed by noise with a small variance. The fifth PF presented by Nurminen et al. was the collision particle filter (PFc). The details of the other filters are not further discussed here, but are described in [9].

5.5.1 Collision Particle Filter (PFc)

The most common form of the PF used in the field of indoor positioning is the collision particle filter (PFc). This is due to its relative simplicity. The PFc propagates particle states with PDR and is a BF, using the motion model as the proposal distribution. The weight update is a single step with the binary weight update used to perform map-matching.

5.5.2 Map-based measurement likelihood (PFw)

PFw also uses PDR to propagate particles. The difference between the two filters is that the PFw includes the angular PDF in the measurement likelihood. The measurement likelihood $p(\mathrm{y}_k|\mathbf{x}_k^i)$ is replaced with the angular PDF $p(\psi_k|\mathbf{x}_k^i)$ and the weight update equation becomes Equation (5.36).

$$w_k^i = p(\mathrm{y}_k|\mathbf{x}_k^i)p(\psi_k|\mathbf{x}_k^i)w_{k-1}^i \tag{5.36}$$

After the particles are re-weighted with the angular PDF, map-matching is performed with the binary weight update.

5.6 New Weight Update

This thesis proposes a new weight update method. The motivation for this new method is based upon the weaknesses of the current PF methods in certain situations. The situation where a map only constrains a single side of the particle cloud is of particular interest in this work. This can occur when pedestrians walk closely to the wall of a wide corridor, or in a room where the outer wall is the only feature constraining the pedestrian's motion. The PFc does not purposely

weight particle directions, instead it eliminates particles that have crossed map boundaries. This shifts the mean of the particle distribution away from the wall towards open space producing an undesired estimate. This solution drift can be seen in Figures 5.5 and 5.6. PF1 and PF2 suffer from the same problem as PFc, as they also perform map-matching and assign more weight to open space directions. PFw and PF3 weight map directions more heavily so they perform well in these situations as the PDR path closely follows the building map. Though these filters perform well in those instances, this assumption can be impractical in many other situations [9]. A new weight update method is proposed to mitigate the limitations of prior methods.

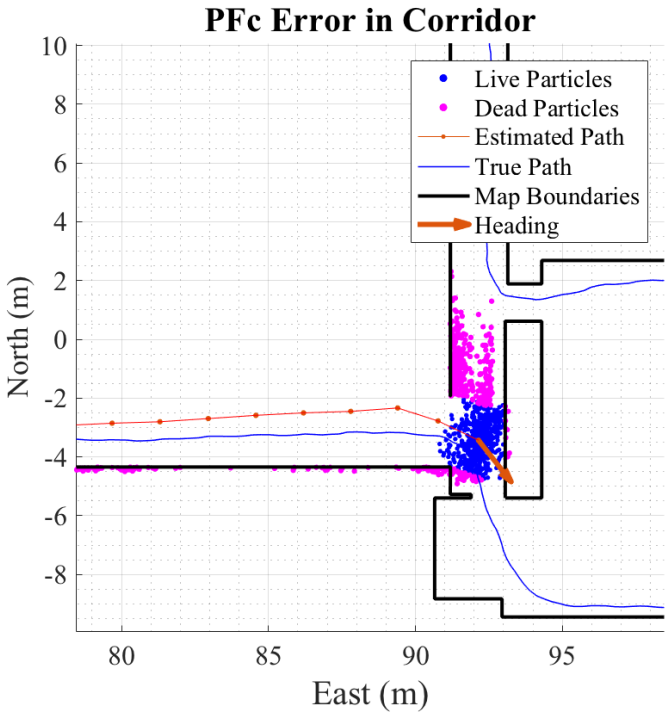

Figure 5.5: PFc Error in Corridor

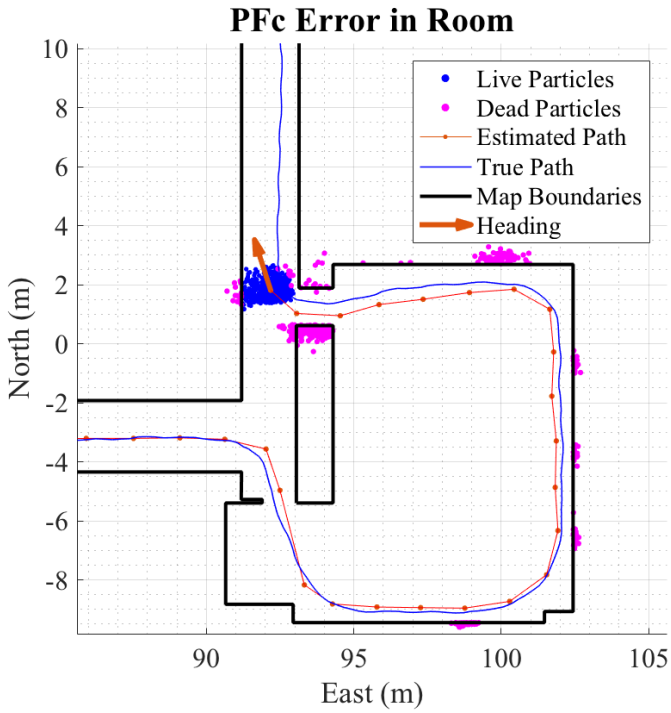

Figure 5.6: PFc Error in Room

The proposed method is similar to the PFc. The particles are propagated with PDR, but a new weight update is used. Instead of weighting particles that collide with the wall to zero or a small number, particles are placed on the inside of the wall that they crossed, and their weights are retained. These particles are now no longer in violation and produce an estimate which reduces the drift and improves the navigation solution. The steps of the new method are described as follows. The vector from the previous position to the current position ($\mathbf{r}_{k/k-1}$) is found by subtracting the previous position ($\mathbf{r}_{k-1}$) at time t_{k-1} from the current position ($\mathbf{r}_k$) at time t_k. It is then rotated by angle α and becomes $\mathbf{r}^*_{k/k-1}$. This is then added to $\mathbf{r}^*_k$ so that the particle is no longer outside of the map boundary. The new method is illustrated in Figure 5.7 and is detailed in Algorithm 4.

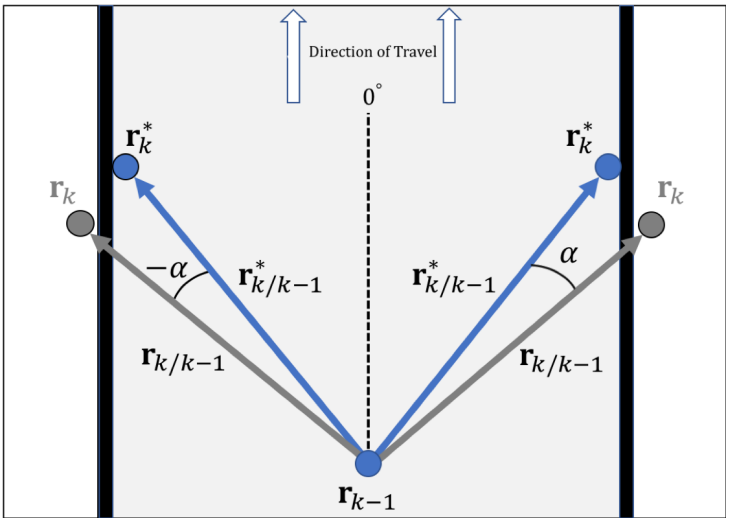

Figure 5.7: Inbounds Particle Filter

The particle is either rotated in the clockwise (CW) or counter clockwise (CCW) direction depending on its out of bounds position. The new update uses a signum function $\text{sign}(x)$ to change the sign of angle α each iteration, to check both directions. Angle $\delta\alpha$ is added to the absolute value of angle α until a wall cross is no longer detected. Also, angle $\delta\alpha$ is set to a value that balances finding the minimum rotation angle with reasonable error and computational time. The angle α is initialized as angle $\delta\alpha$. The value of angle α is limited to the range $-\alpha_{max} \leq \alpha \leq \alpha_{max}$. The term α_{max} cannot be greater than $90°$ otherwise the positive and negative angles will overlap. For this thesis α_{max} was empirically tuned to $45°$. The PF that implements this new method is called the inbounds particle filter (PFi).

Algorithm 4: Rotate Particle Inbounds

Input: $\mathbf{r}_{k-1}, \mathbf{r}_k$, mapStart, mapStop
Output: $\mathbf{r}_k^*$

Initialization: $\alpha = 0.1°$; $\alpha_{max} = 45°$; valid = true;
while $\text{inbounds} = \text{false} \quad \& \quad \text{valid} = \text{true}$ **do**
    % Calculate vector from previous to current position
    $\mathbf{r}_{k/k-1} = \mathbf{r}_k - \mathbf{r}_{k-1}$
    % Construct a rotation matrix
    $\mathbf{R} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$
    %Rotate Vector
    $\mathbf{r}^*_{k/k-1} = \mathbf{R} \cdot \mathbf{r}_{k/k-1}$
    %Calculate new particle position
    $\mathbf{r}^*_k = \mathbf{r}_{k-1} + \mathbf{r}^*_{k/k-1}$
    %Take the absolute value of α and add $\delta\alpha$:
    $\alpha = (|\alpha| + \delta\alpha) \cdot \text{sign}(-\alpha)$
    %Check if particle is inbounds
    [inbounds]=paricleInboundsCheck($\mathbf{r}_{k-1}, \mathbf{r}_k$, mapStart, mapStop)
    %Check if α is Valid
    if $\alpha > \alpha_{max}$ **then**
        valid = false
    end
end
%The signum function (sign(x)) is defined as:

$$\text{sign(x)} = \begin{cases} -1, & x < 0 \\ 1, & x > 0 \end{cases}$$

5.7 Performance Evaluation

Multiple types of data can be produced to evaluate the performance of a localization algorithm. One type is simulated data which is generated using a predetermined path which acts as the truth reference [9]. Simulated data is useful as it allows the user to perform analysis of algorithms without actually having to perform data collection with sensors. It is the author's opinion that simulated data should always be coupled with experimental data. This section first presents the error statistics used to perform the evaluation, then presents analysis using simulated data followed by an experimental data analysis to verify the simulated results.

5.7.1 Error Statistics

In order to determine the accuracy of a positioning system, error statistics must be computed. A simple definition of statistical error is the difference between a measured value and the actual value of the parameter. When evaluating the performance of an indoor positioning system, the parameter of interest is position error. Multiple types of position error can be computed to provide a more comprehensive performance evaluation. The different types of position errors used in this work are presented in the remainder of this section.

The first type of error is 2D position error, it is the Euclidean distance between the estimated position and the true position. It is defined as

$$\text{2D position error} = \sqrt{(\delta \text{N})^2 + (\delta \text{E})^2} \tag{5.37}$$

where $\{\delta \text{N}, \delta \text{E}\}$ are the North and East position errors, respectively. The North and East position errors are defined as

$$\delta \text{N} = \hat{\text{N}} - \text{N} \tag{5.38}$$

$$\delta \text{E} = \hat{\text{E}} - \text{E} \tag{5.39}$$

where $\{\hat{\text{N}}, \hat{\text{E}}\}$ are the estimated and $\{\text{N}, \text{E}\}$ are the true North and East positions, respectively.

The North and East position errors separate are not particulary useful for analyzing the system performance, as the orientation of the body frame is constantly changing relative to the

navigation frame. Therefore, the error in the longitudinal and lateral directions of the body frame is used instead. The North and East errors are transformed into the body frame to compute the longitudinal and lateral error. This is shown in Equations (5.40-5.42).

$$\begin{bmatrix} \delta \mathrm{x} \\ \delta \mathrm{y} \end{bmatrix} = \mathbf{C}_n^b \begin{bmatrix} \delta \mathrm{N} \\ \delta \mathrm{E} \end{bmatrix} \tag{5.40}$$

$$\begin{bmatrix} \delta \mathrm{x} \\ \delta \mathrm{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \delta \mathrm{N} \\ \delta \mathrm{E} \end{bmatrix} \tag{5.41}$$

$$\begin{bmatrix} \delta \mathrm{x} \\ \delta \mathrm{y} \end{bmatrix} = \begin{bmatrix} \delta \mathrm{N}\cos\psi + \delta \mathrm{E}\sin\psi \\ -\delta \mathrm{N}\sin\psi + \delta \mathrm{E}\cos\psi \end{bmatrix} \tag{5.42}$$

The matrix $\mathbf{C}_n^b$ is the 2D CTM from the navigation frame (n) to the body frame (b). The terms $\{\delta \mathrm{x}, \delta \mathrm{y}\}$ are the longitudinal and lateral position errors, respectively.

Another type of error is root-mean-square (RMS) error. It is also referred to as RMS deviation by some [87]. Its purpose is to aggregate the error computed at various times into a single measure of error. In other words, it provides a measure of accuracy with a single number for an entire dataset. A general expression for RMS error (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{\mathrm{x}}_k - \mathrm{x}_k)^2} \tag{5.43}$$

where $\{\hat{\mathrm{x}}_k, \mathrm{x}_k\}$ are the estimated and true variables at time t_k, respectively and N is the total number of epochs in time. RMS error can be computed for 2D, longitudinal and lateral position errors as follows

$$\text{2D Position RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[(\delta \mathrm{N}_k)^2 + (\delta \mathrm{E}_k)^2\right]} \tag{5.44}$$

$$\text{Longitudinal Position RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\delta \mathrm{x}_k)^2} \tag{5.45}$$

$$\text{Lateral Position RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\delta \mathrm{y}_k)^2} \tag{5.46}$$

where $\{\delta \mathrm{N}_k, \delta \mathrm{E}_k\}$ are the North and East position errors, respectively and $\{\delta \mathrm{x}_k, \delta \mathrm{y}_k\}$ are the longitudinal and lateral position errors, respectively all at time t_k.

5.7.2 Simulated Data Generation

A simple 2D trajectory was created to simulate a pedestrian walking in a straight line alongside a single wall. This situation is illustrated by Figure 5.8 and the pedestrian's path is shown in Figure 5.9. The estimated positions from PFc and PFi were compared.

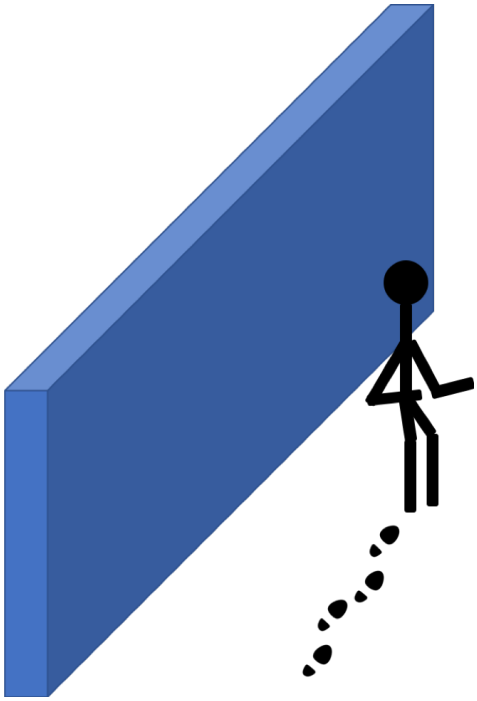

Figure 5.8: Environment Animation

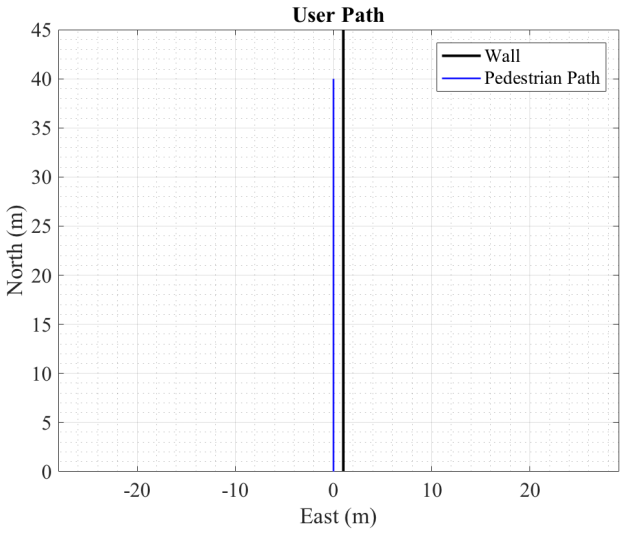

Figure 5.9: User Path of a Simulated Pedestrian

5.7.3 Simulated Results

Due to the stochastic nature of the filters, a Monte Carlo simulation was performed with 5000 iterations of each filter [36,71]. The error at every stride epoch can be seen in Figure 5.10. Note that this is the mean error of the 5000 iterations at each epoch. A stride epoch is defined as a single instance in which the pedestrian takes a new stride. The PFc is shown to have more 2D and lateral position error than PFi, while the PFi is shown to have slightly more longitudinal

error. The lateral, longitudinal, and 2D position RMS errors are plotted in Figure 5.11 and can be seen to converge to a steady state error. The PFi is shown to have more lateral and 2D RMS error than PFc, while PFi and PFc have approximately the same longitudinal error.

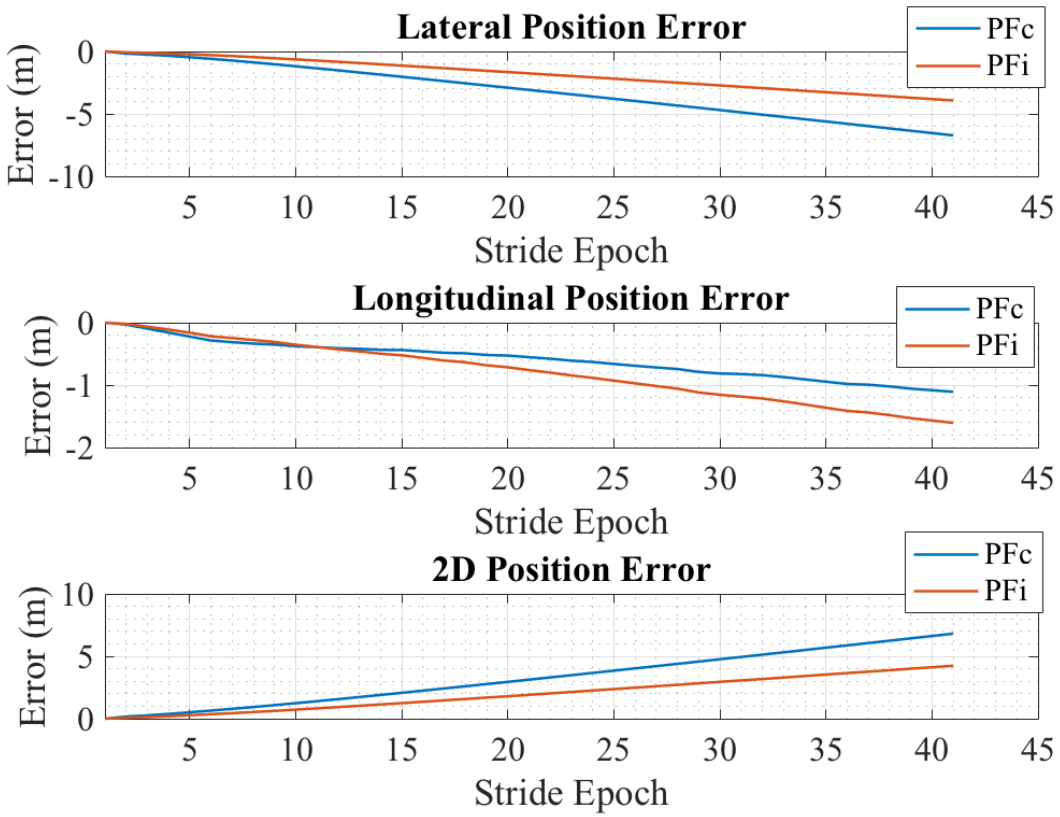

Figure 5.10: Position Errors

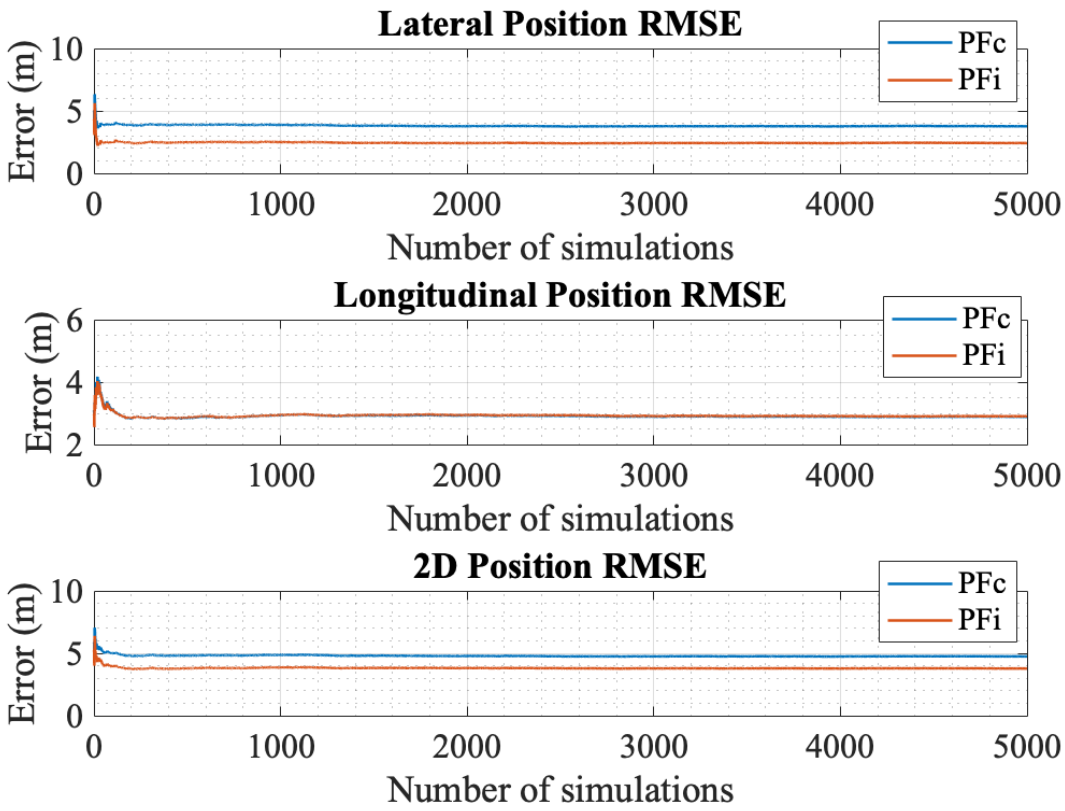

Figure 5.11: Position RMS Error

5.7.4 Experimental Data Collection

Though simulated data is useful for preliminary analysis, it is not desirable as it is difficult to represent the authentic motion of a pedestrian. A second method obtains experimental data by pausing at chosen points along a predetermined path [13]. These true positions of these points are known and allow for error between the true and predicted position to be calculated. This method is also undesirable as the error is only calculated at certain points and not every predicted position in the pedestrian's trajectory. Another method uses absolute position measurements from indoor measurement systems (e.g. ultrasonic positioning) [11]. These systems detailed in [10], are extremely beneficial to performing analysis, but are costly to obtain and time prohibitive to deploy [4]. This thesis proposes simulating indoor walking to obtain experimental data to use in lieu of the above-mentioned methods.

Building interiors are GNSS denied environments, therefore this thesis proposes simulating a building floorplan in an outdoor environment. This would make GPS available as an external measurement. Standalone GPS is $\sim$5m accurate in the horizontal plane. This does not meet the accuracy requirements to survey out a virtual building floorplan and to evaluate PF performance. To amend this, real-time kinematic (RTK) positioning technique was used to obtain horizontal position measurements that are accurate to $\sim$2 cm. With the accuracy requirements satisfied, the steps for simulating a building floorplan outdoors are presented below.

The building floorplan of interest must first be either surveyed or obtained from a source. If the building floorplan is surveyed, the survey data can easily be implemented as a graphical style map (i.e. nodes and ways). Otherwise if building floorplan images are obtained, more steps are required to create a graphical style map. First, a satellite view map of the relevant building should be obtained from Open Street Maps (OSM) or similar equivalent. If OSM is used, the floorplan image can be georeferenced with Java Open Street Maps (JOSM). Once georeferenced, the floorplan image is digitized by overlaying the image with nodes and ways.

A virtual building should then be created in an outdoor space, by first surveying points with RTK-GPS and plotting the building walls and features on the ground. With this 2D representation of a building floor plan now available, the pedestrian can now walk realistic trajectories while using RTK-GPS as an external reference to evaluate algorithm performance.

For this thesis, the floor plan of the 2nd floor of the Woltosz Engineering Research Laboratory building at Auburn University was surveyed. Inertial measurements were taken with a MEMSENSE 3020 IMU which contains a triaxial accelerometers and gyroscopes. Two Novatel ProPak GPS receivers were used. One acted as a base station to broadcast corrections to the second receiver that was mounted on the rover (i.e. pedestrian). The sensors mounted on the pedestrian were interfaced to a central computer and the data collected using the Robot Operating System (ROS).

5.7.5 Experimental Results

Test #1 was performed to validate the proposed PFi against PFc. The test trajectory chosen was a wide corridor in which the building map only constrains the user's movement on one side as seen in Figure 5.12. The true building has an inner wall, which was removed from the map to emulate a large open corridor or atrium. The performance of the filters was characterized by computing lateral, longitudinal, and 2D position error. The 2D position error is calculated from the Euclidean distances between the RTK-GPS and PF positions in time. Due to the probabilistic nature of PF's, a Monte Carlo simulation was performed running each filter 10 times. The 2D position error results from Test #1 can be seen in Figure 5.13. The PFc position solution diverges as time progresses because particles are shifted towards open space when map-matching is performed. The PFi solution error remains relatively constant over the trajectory as the new weight update method restricts the error growth. Table 5.1 shows the RMS error of both filters for Test #1 showing PFi to have the best performance with the lowest RMSE.

The inbounds PF (PFi) is only practical when used in situations when only a single map boundary restricts the pedestrian's position (e.g. wide corridor, room). If used in other situations, like a narrow corridor the new weight method can introduce error into the solution. The PFc works well in almost every situation except those illustrated by Figures 5.5-5.6 and Test #1

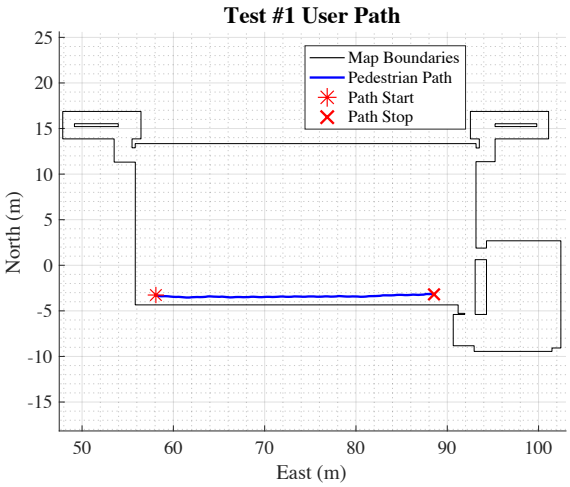

Figure 5.12: Test #1 User Path

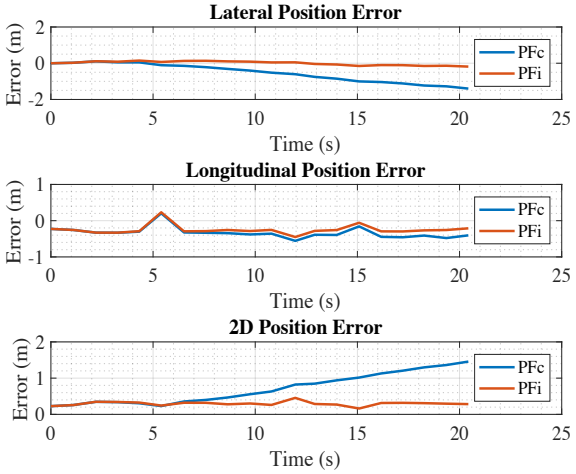

Figure 5.13: Test #1 Position Error

Table 5.1: Test # 1 and # 2 RMS Error

Test 1		Test 2	
Particle Filter	RMSE (m)	Particle Filter	RMSE (m)
PFc	0.819	PFc	0.704
PFi	0.302	PFci	0.605
		PFw	1.097

results. Therefore this thesis proposes using the PFc and PFi in conjunction with one another to utilize the strengths of the two different weight updates. This new filter is termed the collision inbounds particle filter (PFci).

Test #2 was performed to compare three filters: PFc, PFci, and PFw. The length of the test was approximately 2 minutes long and its path can be seen in Figure 5.14. Because of the stochastic behavior of the PFs, each filter was run 10 times. The intervals at which the different PFci weight updates were used was predetermined empirically. When the user's position is constrained with only the exterior corridor wall and while inside the room, the new weight update is performed. In all other instances, map-matching is performed with the binary weight update. The time intervals at which the new weight update was used for these are from 32 to 52 seconds and 58 to 76 seconds, respectively. The 2D error results can be seen in Figure 5.15. The results reveal that the PFw is the worst performing filter. This is most likely due to the fact that this filter is intended to be used for a system with low quality PDR [9]. Figure 5.15 also shows that PFci has less error growth than PFc during the two periods that the new weight

update is used. Table 5.1 shows that PFci is the best performing PF overall, as it converges to the lowest RMSE.

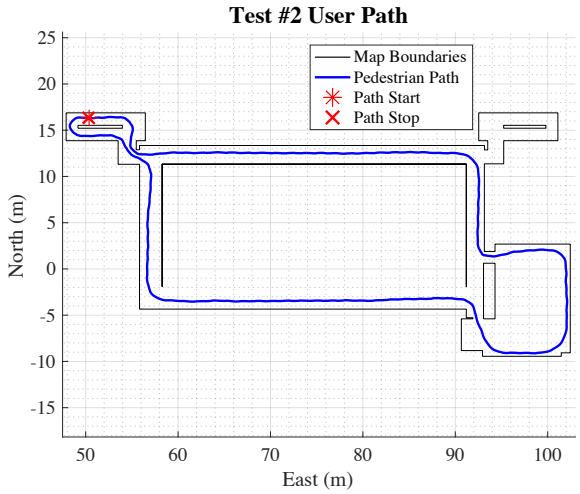

Figure 5.14: Test # 2 User Path

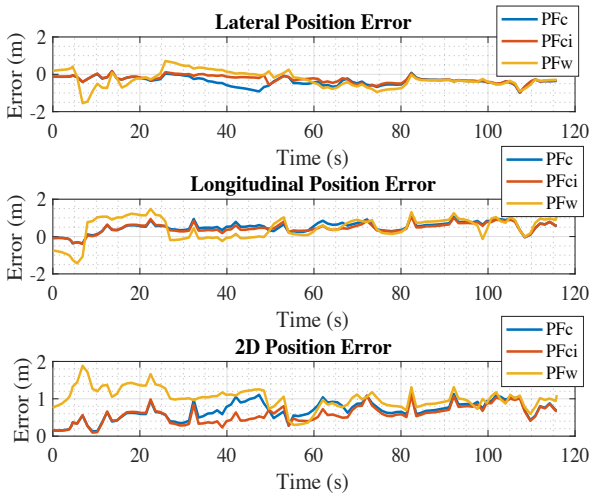

Figure 5.15: Test # 2 Position Error

Chapter 6

Summary, Conclusions, and Future Work

6.1 Summary

Chapter 1 began by motivating why pedestrian navigation is one of the most challenging navigation problems seen today. This was conferred to be because GNSS signals are either degraded or completely unvailable in certain environments thus the user must utilize alternative positioning systems. These systems use body worn inertial sensors that provide relative positioning information using PDR. The problem of unbounded error growth associated with PDR systems was introduced along with the methods to mitigate it. The first method discussed uses an infrastructure based system, that utilizes beacons or networks to provide absolute positioning information. It was decided that these systems are not yet practical as they are expensive and time intensive to deploy. The second method which uses *a priori* map information, was chosen to be the preferred method to aid PDR systems in this thesis. Next, the objectives of this work were introduced. They were as follows: provide background information on current PDR methods including both the classical INS approach and the newer step detecting approach, investigate the current methods of integrating *a priori* with PDR systems, and introduce a new weight update method.

Chapter 2 first introduced the coordinate frames used in this thesis. This included definitions of the body and navigation frames used. Next, the term attitude was defined along with methods used to represent it. Lastly, the transformations to and from different coordinate frames were discussed.

Chapter 3 began by defining pedestrian dead reckoning and its variants and describing the PDR-INS approach. Next, the pedestrian gait cycle along with the terms step and stride were defined, followed by the IMU error model. Following, the gait monitoring process to

detect zero-velocity, still, and stride epochs was described followed by straight walking detection. Subsequently, the error-state Kalman filter was presented along with the system and measurement models as well as the effects of different measurement updates upon the localization solution. The last portion of Chapter 3 discussed each of the steps required to perform an INS mechanization in detail.

Chapter 4 started by motivating the use of PDR-SD as opposed to the classical INS approach. This was followed by a discussion of common step detection methods and how they are implemented. Next, a short study on the effect of IMU mounting location was performed. This revealed that an IMU needs to be mounted on the pedestrian's torso. Following, a survey of step length determination methods was presented as well as the method of linear least squares and its variants. The next section presented a short study of different empirical step length formulas. This study showed that all of the parameter estimates with OLS outperformed those estimated with WLS. This indicated that the measurement variances used were not adequate in weighting the measurements. The study also showed that SL_{11} performed slightly better than the other two formulas (SL_8, SL_9) compared. Lastly, the methods of heading determination were discussed along with the system model used to perform the navigation system update.

Chapter 5 first introduced the estimation framework of fusing PDR and a PF that uses map measurements. Next, a qualitative description of PF estimation was presented which included initialization, propagation, update, and resampling. Subsequently, the system models used to propagate the particles were discussed as well as the methods used to perform the weight update and resampling. Following, the method used to check if particles are inbounds was derived and its implementation shown. In the next section, the current PFs used for indoor positioning systems were introduced. This was followed by the motivation for a new weight update which is based upon the weaknesses of the current methods. This new weight update was explained and the PF that uses it was termed the inbounds particle filter (PFi). Lastly, a performance evaluation was performed using both simulated and experimental data to asses the performance of the PFs. In the next section, the performance evalution is further discussed along with the conclusions that can be drawn from it.

6.2 Conclusions

The first goal of this thesis was to provide a thorough background on current PDR methods. This was effectively done with both classical PDR-INS and the newer PDR-SD approaches. Next, the current particle propagation and weight update methods used in indoor positioning systems were presented. Additionally, the problems associated with each of the different PF methods were discussed motivating the proposal of a new weight update method. This new weight update was introduced to reduce position solution drift in situations where only a single map boundary constrains the user's motion. In addition, a new approach for collecting experimental data was introduced, that uses a high accuracy GNSS reference to evaluate indoor positioning systems. This allowed for experimental validation of an indoor positioning system without the use of expensive indoor systems. Test #1 evaluated PFi and PFc showing that PFi reduced the position drift when only a single wall was used as a measurement update. In Test #2 PFc, PFci, and PFw were assessed within a more general building environment. The test demonstrated the proposed filter PFci to be the best performing filter, as it had the lowest overall error. Therefore, it is concluded that the new PF is an improvement upon previous methods. The following section presents potential future work based upon the work in this thesis.

6.3 Future Work

Some areas of future work are listed below. This includes the continuation of the work in this thesis and additional topics related to the field of pedestrian navigation.

- Create an algorithm to generalize the process of deciding when the two different weight updates for the PFci should be used. This algorithm will detect how many map boundaries are near the user and choose the appropriate weight update.
- A detailed comparison of the performance of PDR-INS and PDR-SD methods should be completed like those seen in [22, 48].

- A comprehensive comparison of step detection and step length determination methods should be performed. The pedestrian navigation research community suffers from a lack of repeatability and comparable results.

- More advanced biomechanics based step length models could be developed. This would be an improvement upon the kneeless biped model presented in Section 4.4.

- The gait monitoring and step detection techniques used in this work should be compared with more advanced methods such as correlation and spectral analysis, as well as machine learning [4, 18].

- Further study of fusing a foot and body mounted IMU should be performed [88].

- Develop machine learning algorithms to perform activity detection (i.e. crawling, walking, running) to use in conjuction with navigation systems to increase the robustness of systems. [3].

References

[1] K. R. Howe, *Vaka moana: Voyages of the ancestors: The discovery and settlement of the Pacific*. University of Hawaii Press, 2006.

[2] J. I. Bowditch, *American practical navigator*. No. 9, US Government Printing Office, 1906.

[3] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[4] R. Harle, "A survey of indoor inertial positioning systems for pedestrians.," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[5] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: principles and applications*. Artech House, 2017.

[6] P. Misra and P. Enge, "Global positioning system: signals, measurements and performance second edition," *Massachusetts: Ganga-Jamuna Press*, 2006.

[7] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE signal processing magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[8] H. Nurminen, A. Ristimaki, S. Ali-Loytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pp. 1–10, IEEE, 2013.

[9] H. Nurminen, M. Raitoharju, and R. Piché, "An efficient indoor positioning particle filter using a floor-plan based proposal distribution.," in *FUSION*, pp. 541–548, 2016.

[10] O. Woodman, *Pedestrian Localisation for Indoor Environments*. PhD thesis, University of Cambridge, Computer Laboratory, Sept. 2010. PhD Thesis.

[11] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 114–123, ACM, 2008.

[12] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, pp. 1–4, IEEE, 2010.

[13] A. Perttula, H. Leppäkoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching.," *IEEE Trans. Instrumentation and Measurement*, vol. 63, no. 11, pp. 2682–2695, 2014.

[14] T. N. Ray, J. D. Pierce, and D. M. Bevly, "A comparison of particle propagation and weight update methods for indoor positioning systems," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, pp. 3398–3408, 2018.

[15] J. Racko, P. Brida, A. Perttula, J. Parviainen, and J. Collin, "Pedestrian dead reckoning with particle filter for handheld smartphone," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, IEEE, 2016.

[16] P. Peltola, C. Hill, and T. Moore, "Particle filter for context sensitive indoor pedestrian navigation," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–6, IEEE, 2016.

[17] J. Pinchin, C. Hide, and T. Moore, "A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–10, IEEE, 2012.

[18] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2017.

[19] J. Farrell and M. Barth, *The global positioning system and inertial navigation*, vol. 61. Mcgraw-hill New York, 1999.

[20] J. Ginsberg, *Engineering dynamics*, vol. 10. Cambridge University Press, 2008.

[21] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*, vol. 17. IET, 2004.

[22] P. D. Groves, G. W. Pulford, C. A. Littlefield, D. L. Nash, and C. J. Mather, "Inertial navigation versus pedestrian dead reckoning: Optimizing the integration," in *Proc. ION GNSS*, pp. 2043–2055, 2007.

[23] A. R. Jiménez, F. Seco, J. C. 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*, pp. 135–143, IEEE, 2010.

[24] S. Rajagopal, "Personal dead reckoning system with shoe mounted inertial sensors," *Master's Degree Project, Stockholm, Sweden*, 2008.

[25] D. Levine, J. Richards, and M. W. Whittle, *Whittle's gait analysis*. Elsevier Health Sciences, 2012.

[26] S. J. Cuccurullo, *Physical medicine and rehabilitation board review*. Demos Medical Publishing, 2014.

[27] I. Bylemans, M. Weyn, and M. Klepal, "Mobile phone-based displacement estimation for opportunistic localisation systems," in *2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 113–118, IEEE, 2009.

[28] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *The Journal of Navigation*, vol. 62, no. 1, pp. 41–58, 2009.

[29] K. Abdulrahim, C. Hide, T. Moore, and C. Hill, "Aiding low cost inertial navigation with building heading for pedestrian navigation," *The Journal of Navigation*, vol. 64, no. 2, pp. 219–233, 2011.

[30] K. A. Rahim, *Heading drift mitigation for low-cost inertial pedestrian navigation*. PhD thesis, University of Nottingham, 2012.

[31] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detection—an algorithm evaluation," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2657–2666, 2010.

[32] D. Pierce, "Incorporation of a foot-mounted imu for multi-sensor pedestrian navigation," Master's thesis, Auburn University, 2016.

[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[34] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems, Second Edition (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science)*. Chapman & Hall/CRC, 2nd ed., 2011.

[35] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[36] R. G. Brown, P. Y. Hwang, *et al.*, *Introduction to random signals and applied Kalman filtering*, vol. 3. Wiley New York, 1992.

[37] J. D. Pierce and D. M. Bevly, "A centralized approach to pedestrian localization using multiple odometry sources," in *Proc. ION ITM*, 2015.

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

[39] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[40] O. J. Woodman, "An introduction to inertial navigation," tech. rep., University of Cambridge, Computer Laboratory, 2007.

[41] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.

[42] B. Schutz, B. Tapley, and G. H. Born, *Statistical orbit determination*. Elsevier, 2004.

[43] A. Schumacher, "Integration of a gps aided strapdown inertial navigation system for land vehicles," *Master of Science Thesis, Stockholm, Sweeden*, pp. 1–57, 2006.

[44] I. Skog, *GNSS-aided INS for land vehicle positioning and navigation*. PhD thesis, KTH, 2007.

[45] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Positioning*, vol. 1, no. 08, p. 0, 2004.

[46] S. H. Shin and C. G. Park, "Adaptive step length estimation algorithm using optimal parameters and movement status awareness," *Medical engineering & physics*, vol. 33, no. 9, pp. 1064–1071, 2011.

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

[48] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu," in *2009 IEEE International Symposium on Intelligent Signal Processing*, pp. 37–42, IEEE, 2009.

[49] H. Leppakoski, J. Kappi, J. Syrjarinne, and J. Takala, "Error analysis of step length estimation in pedestrian dead reckoning," in *ION GPS 2002: 15 th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2002.

[50] N. Zhao, "Full-featured pedometer design realized with 3-axis digital accelerometer," *Analog Dialogue*, vol. 44, no. 06, pp. 1–5, 2010.

[51] J. Kappi, J. Syrjarinne, and J. Saarinen, "Mems-imu based pedestrian navigator for handheld devices," in *Proceedings of the 14th international technical meeting of the satellite division of the institute of navigation (ION GPS 2001)*, pp. 1369–1373, 2001.

[52] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pp. 27–35, 2006.

[53] C. Randell, C. Djiallis, and H. Muller, "Personal position measurement using dead reckoning," in *null*, p. 166, IEEE, 2003.

[54] D. Gusenbauer, C. Isert, and J. Krösche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–9, IEEE, 2010.

[55] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[56] S. Beauregard, "A helmet-mounted pedestrian dead reckoning system," in *3rd International Forum on Applied Wearable Computing 2006*, pp. 1–11, VDE, 2006.

[57] H. Weinberg, "Using the adxl202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, 2002.

[58] S. Godha, G. Lachapelle, and M. E. Cannon, "Integrated gps/ins system for pedestrian navigation in a signal degraded environment," in *Ion gnss*, vol. 2006, 2006.

[59] R. Faragher and R. Harle, "Smartslam-an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing," in *ION GNSS*, vol. 13, pp. 1–14, 2013.

[60] T. Judd, "A personal dead reckoning module," in *ION GPS*, vol. 97, pp. 1–5, 1997.

[61] G. Thonet, O. Blanc, P. Vandergheynst, E. Pruvot, J.-M. Vesin, and J.-P. Antoine, "Wavelet-based detection of ventricular ectopic beats in heart rate signals," *Applied Signal Processing*, vol. 5, no. 3, pp. 170–181, 1998.

[62] S. Y. Cho, K. W. Lee, C. G. Park, and J. G. Lee, "A personal navigation system using low-cost mems/gps/fluxgate," *Proc. ION 59th*, 2003.

[63] C. V. Bouten, A. A. Sauren, M. Verduin, and J. Janssen, "Effects of placement and orientation of body-fixed accelerometers on the assessment of energy expenditure during walking," *Medical and biological engineering and computing*, vol. 35, no. 1, pp. 50–56, 1997.

[64] D. Farris, "Prototype development of a system providing for initial a.˜ sessment of the dynamics/kinematic of bipedal motion.'," *Prec. lOth Ann, Coati RESNA*, pp. 726–728, 1987.

[65] N. P. Hamilton, *Kinesiology: Scientific basis of human motion*. Brown & Benchmark, 2011.

[66] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. G. Boronat, "Comparison and evaluation of acceleration based step length estimators for handheld devices," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–6, IEEE, 2010.

[67] W. Zijlstra and A. L. Hof, "Displacement of the pelvis during human walking: experimental data and model predictions," *Gait & posture*, vol. 6, no. 3, pp. 249–262, 1997.

[68] J. E. Bertram and A. Ruina, "Multiple walking speed–frequency relations are predicted by constrained optimization," *Journal of theoretical Biology*, vol. 209, no. 4, pp. 445–453, 2001.

[69] S. Yang and Q. Li, "Ambulatory walking speed estimation under different step lengths and frequencies," in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 658–663, IEEE, 2010.

[70] W. H. Greene, *Econometric analysis*. Pearson Education India, 2003.

[71] J. A. Gubner, *Probability and random processes for electrical and computer engineers*. Cambridge University Press, 2006.

[72] D. Gebre-Egziabher, G. H. Elkaim, J. David Powell, and B. W. Parkinson, "Calibration of strapdown magnetometers in magnetic field domain," *Journal of Aerospace Engineering*, vol. 19, no. 2, pp. 87–102, 2006.

[73] M. J. Caruso, "Applications of magnetic sensors for low cost compass systems," in *Position Location and Navigation Symposium, IEEE 2000*, pp. 177–184, IEEE, 2000.

[74] J. A. Shockley, "Ground vehicle navigation using magnetic field variation," tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF . . . , 2012.

[75] G. Morales, "Magnetometer aided navigation filters for improved observability and estimation on ground vehicles," Master's thesis, Auburn University, 2016.

[76] D. M. Bevly, "Global positioning system (gps): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles," *Journal of dynamic systems, measurement, and control*, vol. 126, no. 2, pp. 255–264, 2004.

[77] B. Brumback and M. Srinath, "A chi-square test for fault-detection in kalman filters," *IEEE Transactions on Automatic Control*, vol. 32, no. 6, pp. 552–554, 1987.

[78] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[79] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 4, pp. 2003–2011, IEEE, 2001.

[80] M. Klepal, S. Beauregard, *et al.*, "A backtracking particle filter for fusing building plans with pdr displacement estimates," in *2008 5th Workshop on Positioning, Navigation and Communication*, pp. 207–212, IEEE, 2008.

[81] B. Krach and P. Robertson, "Integration of foot-mounted inertial sensors into a bayesian location estimation framework," in *2008 5th Workshop on Positioning, Navigation and Communication*, pp. 55–61, IEEE, 2008.

[82] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[83] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[84] R. N. Jazar, *Vehicle dynamics: theory and application*. Springer, 2017.

[85] S. Kaiser, M. Khider, and P. Robertson, "A human motion model based on maps for navigation systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, p. 60, 2011.

[86] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 64–69, IEEE, 2005.

[87] J. R. Taylor, "An introduction to error analysis 2nd edn (sausalito, ca," 1997.

[88] T. Gädeke, J. Schmid, M. Zahnlecker, W. Stork, and K. D. Müller-Glaser, "Smartphone pedestrian navigation by foot-imu sensor fusion," in *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pp. 1–8, IEEE, 2012.