Summer 2020

Wireless Wearable System for the Assessment of Gait

Abdallah Jabr
Southern Methodist University, abdallah.jabr@gmail.com

WIRELESS WEARABLE SYSTEM
FOR THE ASSESSMENT OF GAIT

Approved by:

Dr. Paul Krueger
Professor

Dr. Yildirim Hurmuzlu
Professor

Dr. Dario Villarreal
Assistant Professor

WIRELESS WEARABLE SYSTEM FOR THE ASSESSMENT OF GAIT

A Thesis Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science

with a

Major in Mechanical Engineering

by

Abdallah Jabr

(B.Sc., Mechatronics Engineering, University of Jordan, Jordan)

August 4, 2020

ACKNOWLEDGMENTS

This work could not have been accomplished without the help and constant support of my advisor, Prof. Krueger, who constantly provided feedback and advice throughout the writing of my thesis, and Prof. Villarreal, who provided consistent encouragement and an environment of sharing in his laboratory. I am also grateful for my colleagues in the Mechanical Engineering department who were always ready to provide support and advice.

Jabr, Abdallah (B.Sc., Mechatronics Engineering, University of Jordan, Jordan)

Wireless Wearable System
for the Assessment of Gait

Advisor: Dr. Paul Krueger
Master of Science degree conferred August 4, 2020
Thesis completed August 4, 2020

This work investigates the development and use of a wireless wearable system for the assessment of gait. The system proposed consists of a sensor module that is attached to the foot. The sensor proposed is an inertial measurement unit, often abbreviated as **IMU** - a 9-axis System in Package (SiP) including a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer, as well as a fusion engine for signal processing. While the focus of this work is on evaluating gait metrics, the performance of the proposed IMU in evaluating orientation (perhaps for joint angle measurement) is quantified. In doing this work, we try to address the issue of cost. While inertial-based wearable(s) are typically cheaper than optical-based measurement systems, they are still relatively expensive ranging from $3500-$11000 depending on the number of modules and manufacturer. Examples include *MTw Awinda Research Bundle* (XSENS) and *Blue Trident IMU* (VICON). In this work, I use an off-the-shelf commercial IMU unit that costs $10, a wireless IoT chip that costs $10, and a Li-Ion battery that costs $7 for a total of $27 for one wireless sensor module.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Dedicated to my parents.

Chapter 1

INTRODUCTION

1.1. Objectives and Motivation

MEMS inertial sensor technology has seen ubiquitous uses in engineering, aviation, sports and fitness monitoring, robotics, and perhaps in other industrial sectors. Advancements in wireless communication technology have afforded the emergence of wireless wearable(s) where inertial sensors have seen extensive usage. Usage of inertial sensors have not extended as commonly in the medical field, however, despite the high interest of the engineering community in inertial-enabled medical diagnostics [2]. Not only can these sensors be more easily adopted in clinical diagnosis and treatment procedures than their current counterparts, but they can also monitor gait continuously outside clinics [2]. The roadblocks that seemingly prevent standardized usage of inertial sensors in clinical settings include wear-ability, practicality, and medical context-lacking presentation of sensor outputs that may not be understandable by the persons operating the equipment [2]. However, wear-ability is being addressed by the fast introduction of increasingly smaller sensors, the advent of flexible and 3D-printed electronics, and recent advancements in battery technology. As a proof of concept, this work aims to develop a low-cost wearable system that addresses the issue of sensor placement through use of a neural model and the evaluation of gait metrics in a Kalman-based framework. Some background on motion capture techniques is presented first to establish the context of this work.

1.2. Motion Capture Techniques

Optical, marker-based measurement systems are the gold standard for motion capture. Current solutions offer sub-millimeter accuracy, robust operation, and high-frequency out-

puts reaching 1 KHz. However, these systems are expensive, costing in the tens or even hundreds of thousands depending on the number of cameras, specifications, and manufacturer. Optical systems are generally not portable, in need of constant calibration, confined to the field of view of the cameras, and need an unobstructed line of view to capture the motion of markers. Inertial-based motion capture relies on sensor fusion of an array of sensors to estimate motion, accelerometers and gyroscopes making the basic building blocks for such systems. Inertial-based motion capture systems offer a cheaper alternative to optical systems as well as practicality through portability, ease of use, and small profile. However, inertial-based systems suffer from drifting measurements, sensitivity to high-frequency motion, and still rely on magnetic measurements to reference the heading angular direction, which depends on the detection of the faint magnetic field of the Earth.

1.3. Inertial Motion Capture

As mentioned above, inertial-based motion capture relies on the fusion of multiple sources of sensory information to evaluate the best motion estimate. Accelerometers output linear accelerations in the sensor frame of reference while gyroscopes output angular velocities. Theoretically speaking, integrating angular velocities over time should yield angular position while twice integration of linear accelerations should yield linear position. In practice, however, a plethora of problems plague the estimation task: accelerometers suffer from noisy measurements, gyroscopes suffer from fast-drifting measurements, and lastly, a good reference for the heading requires a magnetic compass, which causes the largest source of error in the estimation problem. Rotating linear accelerations from the sensor frame to an inertial frame of reference to integrate for velocity and position requires accurate estimation of the angular position of the sensor, which in turn requires integrating the drifting gyroscope measurements. A digital compass or a magnetometer is also needed if the application requires an accurate heading estimate. Magnetometers produce an angular position estimate based on the magnetic field of the earth, which varies depending on geographical location. Metallic structures, electric lines, monitors, phones, and a myriad of other magnetic field-

inducing sources corrupt measurement of the Earth's magnetic field intensity needed for the estimation problem. Outdoor applications utilize **GNS** (Global Navigation System) satellite signals to aid in heading corrections, but it remains very problematic to infer an accurate estimate of the sensor's yaw in indoor applications.

1.3.1. Accelerometers

Given that the accelerometer measures linear accelerations in all three sensitive axes of the sensor, including the acceleration of Earth's gravity, then if the gravitational acceleration g is estimated from the IMU (Inertial Measurement Unit) signals, the rotated gravity vector measured in the sensor body frame of reference can provide an estimate of the tilt of the sensor. However, since the gravity vector is parallel with one of the axes in the inertial (*local frame* for this application), the accelerometer is unable to give complete information about the rotation of the sensor. Specifically, the rotation obtained does not produce a unique solution for the heading - the angular rotation about the gravity vector. Noise is the main source of error for accelerometers. The sensor's output is either in gravities (multiples of g - Earth's gravitational acceleration) or $\frac{m}{s^2}$.

1.3.2. Gyroscopes

Gyroscopes measure the angular velocities undergone by the rotating sensor in the three sensitive axes, from which angular position can be evaluated by integrating the gyroscopic signals over time. Gyroscopes suffer from a time-varying bias, which is the dominant source of error, as well as noise. The output is usually expressed in $\frac{\text{deg}}{s}$ but some sensors may report the output in $\frac{rad}{s}$.

1.3.3. Magnetometers

Magnetometers measure the magnetic field intensity along the three directions of the sensor's sensitive axes. Using the magnetic field intensities they output, magnetometers provide complete orientation information of the IMU sensor. Magnetometers suffer from sensitiv-

ity to local magnetic fields that distort the Earth's magnetic field measurement needed to extract information about the sensor orientation. This distortion manifests in a simple constant 1×3 measurement offset, termed as the *hard-iron* bias, and a more complex 3×3 scale matrix affecting the orthogonality of the 1×3 measurement, termed the *soft-iron* bias. The output is usually expressed in Gauss G or Tesla T.

1.4. Outline

1. This work investigates developing a prototype based on MEMS inertial technology for the assessment of gait. A wireless wearable for the foot is considered to capture motion that can be used to evaluate important gait metrics: speed, cadence, and stride length.

2. The Kalman framework is utilized for the estimation problems. An Extended Kalman Filter is used for orientation estimation of the sensor module.

3. For gait speed evaluation, estimating the linear velocity of the foot-attached sensor module is a challenging task. The drifting angular measurements used to rotate the linear acceleration vector into the local reference frame (*inertial frame* for this application) will quickly corrupt the linear velocity estimates. To mitigate this issue, ZUPT(s) (*Zero-Velocity Updates*) are used to aid in this task. ZUPT(s) are incorporated in an error-state Kalman filter that estimates the position error, velocity error, and orientation error, which are then used to correct motion states. ZUPT(s) are applied when the foot lays flat on the ground where it is assumed that the foot velocity vector is the zero vector. Linear acceleration peaks provide the time it takes to complete a full step. Using this time period with the corrected linear velocity estimates, it becomes possible to obtain the best estimates for gait speed, the stride length, and cadence in the local inertial frame.

4. For arbitrary placement on the foot, a regression neural network models the sensor orientation *with respect to* the foot. The input to the model is a 1×9 vector of linear accelerations, angular velocities, and magnetic field measurements from the sensor while the output is the sensor orientation.

Chapter 2

LITERATURE REVIEW

2.1. Estimation

2.1.1. Introduction

Inertial-based motion estimation has been explored extensively in the engineering literature [19] [27] [21]. Many different filtering, smoothing, and optimization-based algorithms have been developed to solve the problem of kinematic estimation for inertial-based sensors [8]. While accelerometers suffer from noise, their measurements do not drift, making them suitable to be used as a reference. On the other hand, gyroscopic measurements are highly accurate but suffer from drift. Kalman filters can be used to correct orientation estimates by using accelerometer tilt estimates as a reference against those estimated from the gyroscopes.

2.1.2. Kalman Filtering

The accepted basis for orientation, velocity, and position estimation in navigation systems for aviation is the Kalman filter [12]. Inertial-based, indoor motion estimation is very similar to motion estimation for aircraft navigation, but simplified by the fact that geographical location (longitude and latitude) can be assumed constant and hence do not enter into the mathematical model. The Kalman filter is a statistical estimator that is able to adapt to the noise inherent in the operation of inertial sensors. It does so by explicit modeling of the noise in the filter process model, generally assumed to be a linear stochastic function of white Gaussian noise of zero-mean and a finite covariance.

The Kalman filter can be completely described using second-order statistics - the mean, describing the state, and covariance, describing the uncertainty in the state. The Kalman filter relies on a process model describing the evolution of the state over time to find a *prediction* of the state at the next time step, and sensor measurements that can infer information about the states. The predicted state estimate is compared with the available measurements to produce a better final state estimate depending on knowledge of the noise present in the system, both in the process model and the sensor measurements. Formulas for the time-propagation of the state mean and state covariance that minimize uncertainty in the system can be found explicitly by solving a simple weighted least-squares problem [5]. The assumption of Gaussian noise is appropriate due to the central-limit theorem, stating that the convolution of many small sources of error will approximate a Gaussian distribution, making the Kalman filter an optimal choice for estimation problems. However, the Kalman filter in its simplest form is a linear filter. Here, a constant gain, called the Kalman gain, can be solved for once and then used in the filter implementation. The problem of kinematic estimation is highly non-linear, however, making the use of nonlinear variations of the Kalman filter a necessity.

2.1.2.1. Extended Kalman Filter

In this work, the Extended Kalman filter; commonly called the EKF - an extension to the original Kalman filter, is used. The EKF [5] linearizes the nonlinear process model and measurement function at every time step, evaluates a new Kalman gain, and produces a state estimate. Simply put, the EKF evaluates a linear Kalman filter at every time step. The EKF is the most commonly used form of the Kalman filter. It is relatively easy to implement, offers intuitive addition of new states to the process model, and is well-understood. However, it requires evaluation of state and error Jacobians for linearization; the derivation of which can be a tedious and difficult task if the state vector is large.

2.2. IMU-Foot Auto-Calibration

Auto-calibration refers to the evaluation of the proper spatial transformation to apply to the sensor module's estimated orientation that would align the arbitrarily-placed module to the foot it is attached to. Use of current MEMS, inertial-based, wearable systems requires proper manual placement calibration of the sensor module on the foot, which in turn requires expert knowledge and is prone to human error. Misplacing sensor modules on body segments can result in large errors [31]. For practical use by non-specialists, inertial-based assessment systems should be cordless, placement-insensitive (auto-calibrating), and be able to adapt to small motion artifacts, basically, a plug and play system. However, current commercial systems require manual alignment or functional calibration through static poses [3] or other difficult to perform methods [1]. Other error-contributing factors are less intuitive and difficult to model, such as loose garments and clothing material [33].

Neural models offer the ability to learn complex mappings between inputs (sensor measurements) and outputs (kinematics) given a reliable source for outputs. Supervised learning has the ability to generalize the performance of neural models given enough data. Using a multi-layer perceptron (MLP) architecture to regress the sensor placement is the solution proposed for this problem. Utilizing a reliable source for orientation outputs, the supervised learning problem can be formulated and solved for the needed model, which could then be deployed online, allowing for arbitrary placement of the sensor module on the foot in the final application. In [33], the authors used a multi-layer perceptron (MLP) network to learn sensor-displacement patterns in loose garments using 3 hours of motion data collected from 12 subjects. This work proved the ability of deep supervised learning to generalize performance when hard-to-model errors are present (loose garments). However, despite improved performance compared with regular estimation, orientation errors are still relatively large (10° - 14°). Better accuracy can be achieved assuming rigid fitting of the sensor modules on the body segments. In [25], the authors used deep learning architectures to auto-classify and auto-calibrate the sensor modules placed arbitrarily on body segments. The average accuracy for the auto-classification of segments based on inertial measurements was 98.57% while

the mean orientation error for auto-calibration was 15.21°. However, for auto-calibration, the authors assumed 360° variability for two rotational degrees of freedom of each sensor attached to a segment. While this approach offers large leeway in sensor placement, it comes at the expense of accuracy. While the focus of [33] and [25] was on the ability of common deep learning architectures to generalize model performance, this work attempts to create a solution that only models angular variability in the axis normal to the foot to help increase accuracy, a requirement for specialty uses [2], that is easy to use, and robust to small human error and motion artifacts (from clothing), all while considering the nature of gait and its metrics in the design.

Chapter 3

EXPERIMENTAL SETUP AND METHODS

3.1. Setup

This chapter investigates the methods used to quantify the quality and accuracy of the sensor module in measuring the orientation and the aforementioned gait metrics. It also aims to make clear the techniques used to evaluate these quantities for both the optical measurement system, which is used as the source for ground-truth measurements, and also for the sensor module.

3.1.1. Sensor Module

The sensor module shown in figure 3.1 consists of an IoT (Internet of Things) device that comprises a 120Mhz low-power micro-controller (Particle Photon), a 1000mAh Lithium-ion battery, and an inertial measurement unit (BNO080), packaged in a compact 3D-printed enclosure. The inertial sensor is attached vertically to one of the walls of the enclosure to maximize the utilized volume and ensure a compact size of the assembly. The IoT device reads measurements from the sensor at a variable-rate (up to 0.4Khz) using an I^2C (Inter-Integrated Circuit) serial interface, serializes the data into a fixed-size message, then sends the message over a TCP WiFi connection to a MATLAB TCP interface for data processing.

Figure 3.1: Sensor Module

3.1.2. Optical Measurement System

A 16-camera VICON optical system is mounted 1 foot below the lab ceiling and setup to maximize capture of the working lab volume. It uses what is referred to as *optical retro-reflective markers* to identify the rigid-body for which we have interest in finding inertial information: position, orientation, etc. Those markers are tracked by the infrared cameras mounted on the lab walls. The accuracy of the VICON optical system is assessed as part of the work done in [28] and is found to offer sub millimeter accuracy in estimating position of rigid-body objects. However, it is worth noting that during subject activity, potential movement of the markers on body segments due to skin motion lowers the accuracy of measurements for the actual segment. Data rate for the cameras is up to 330Hz and their resolution is 2.2 Mega-Pixel. The VICON system is used as the source for ground-truth measurements (reference measurements) to be used for comparison against those of the sensor module. Information collected from the cameras is then used by the VICON software to evaluate the position and orientation for the tracked foot-attached sensor module. Velocity is evaluated by finding the gradient of the position vector.

3.1.3. Data Acquisition

Here we investigate the methods used to acquire data from both the sensor module and the optical system. The infrared cameras are setup to track the retro-reflective markers at 100hz. Tracker is a software developed by the VICON manufacturer to track rigid-body objects (the foot-attached sensor module in this application). Three retro-reflective markers are placed non-concurrently on a plane surface to construct a body coordinate system. The global coordinate system (to which the cameras are measuring with respect to) is also setup in the software using a hand-held wand. The wand allows for setting up the origin and other parameters needed for the global coordinate system.

3.2. Experimental Procedure

1. The cameras are calibrated using a standard procedure where a wand is moved in the control volume captured by the cameras. The Tracker software then evaluates whether or not the errors introduced by camera placement is acceptable.

2. Once the camera calibration procedure is done, the wand is placed somewhere at the center of the lab to designate the origin of the global coordinate system. Positions and orientations will be referenced to this coordinate system.

3. The foot-attached module is fitted with optical markers on its surface to be tracked by the infrared cameras.

4. The sensor module body coordinate system constructed from the optical markers is aligned with the global coordinate system in the Tracker software. This implies that the information measured by both the optical system and the sensor module share reference coordinate systems that are aligned to one another.

5. A MATLAB script is run on the central computer to capture position and attitude data from both the optical system (via a streaming MATLAB library developed by VICON) and from the sensor module using the WiFi TCP link. Data for both acquisition systems is sent at a rate of 100Hz.

3.3. Data

Table 3.1 below summarizes the quantities supplied by the sensor and the optical measurement system. It also lists the quantities we are interested in estimating. $\mathbf{a}$ is the linear acceleration vector, $\boldsymbol{\omega}$ is the angular velocity vector, $\mathbf{m}$ is the magnetic intensity vector of the Earth, $\mathbf{v}$ is the linear velocity vector, $\mathbf{p}$ is the linear position vector, and $\mathbf{q}$ is the orientation quaternion. Gait speed is evaluated from the estimated linear velocity $\mathbf{v}$, cadence is evaluated from the acceleration signal $\mathbf{a}$, stride length is evaluated from both $\mathbf{v}$ and $\mathbf{a}$.

Parameters	
Sensor Module Measured Quantities	$\mathbf{a}$, $\boldsymbol{\omega}$, $\mathbf{m}$
Sensor Module Estimated Quantities	$\mathbf{q}$, $\mathbf{v}$, $\mathbf{p}$
Optical System Measured Quantities	$\mathbf{q}$, $\mathbf{p}$

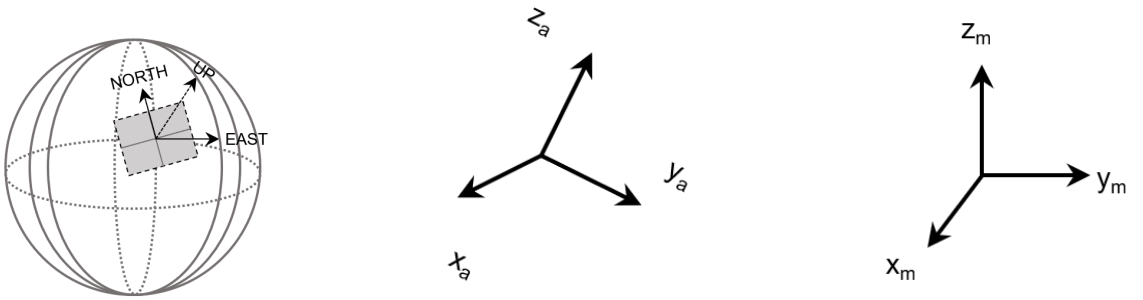
Table 3.1: Measured and Estimated Quantities

Chapter 4

Filter Modeling and Kinematics

4.1. Coordinate Frames

This section introduces the coordinate systems used to define the kinematic models for the estimation problem, as well as the inertial frame for the sensor module. Since the wearable module developed is an indoor application and will not be used across large distances around the globe, there is no need to use a global inertial frame; meaning, the geographical location defined by latitude, longitude, and altitude does not need to enter the transformation equations between the module body frame and the inertial frame (defined locally and termed the *local frame*). Figure 4.2 shows the sensor body frame when the module is mounted on the foot.



(a) Local Frame (b) Accelerometer Axes (c) Magnetometer Axes

Figure 4.1: Sensor Coordinate Frames

4.1.1. World Frame: Local Frame

The IMU (inertial measurement unit) used in this work aligns the sensor measurements using an *ENU* or East-North-Up body frame. Since no geographical jumps are likely to

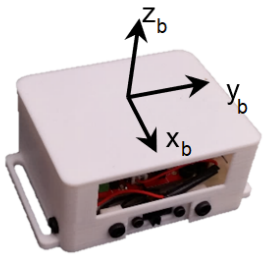


Figure 4.2: Foot Direction Aligns with the x_a Axis

happen during use of the wearable module, a *stationary local* inertial frame is used that discards variation in geographical location in the model. The application also neglects the effect of Earth's rotation about its own axis since it is very small. The ENU convention is used for the *local* frame in this work to avoid transforming the sensor measurements into a different body frame. The local frame in figure 4.1a defines a *right-handed coordinate system* with east as the x-axis, north as the y-axis, and upward as the z axis.

4.1.2. Magnetometer Sensor Frame

The magnetometer measures the magnetic field intensity of the magnetic field of the Earth. This field vector penetrates different points on the Earth's surface at a specific inclination and declination angles. The inclination angle defines the angle the magnetic field vector makes with the horizontal plane and the declination angle defines the angle between geographic north (in the *local* frame) and magnetic north. The inclination angle is 61° and the declination angle 3° in Dallas, Texas, USA, as found from the International Geomagnetic Reference Field (IGRF), from which the magnetic field vector can be evaluated. In figure 4.3, the green line represents the Earth's magnetic field intensity vector, X_{mGeo} represents the direction of geographic north, X_m represents the direction of magnetic north, D denotes the declination angle and I the inclination angle.

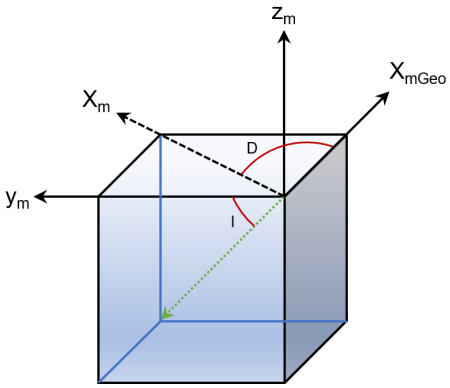


Figure 4.3: Inclination and Declination Angle in ENU Local Frame

4.1.3. IMU Sensor Frame

Technically speaking, an IMU houses *only* an accelerometer and a gyroscope. The magnetometer or digital compass is usually a separate chip. However, the term "IMU" is sometimes used to reference all three sensors on a PCB. The distinction has been made to make clear that the sensitive axes of the accelerometer and gyroscope (IMU) are usually aligned, while the sensitive axes of the latter and the magnetometer are not.

4.1.4. Sensor Body Frame

Due to the single chip design of the BNO080 [7], outputs of the sensor are all reported with respect to the sensor body frame **b**.

4.2. Sensor Models

This section defines the sensor models that will later be used in formulating different estimation problems. Each sensor will be characterized based on three aspects: the main sources of error, how the output measurements contribute to the different kinematic estimation problems or the role of each sensor, and lastly, a description of the nature of measurements.

4.2.1. Accelerometer

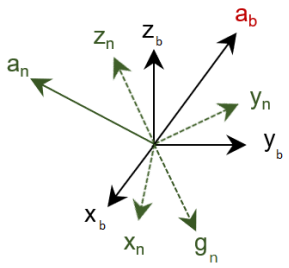


Figure 4.4: Linear Acceleration in the Sensor Body Frame **b** and Local Frame **n**

Linear acceleration is modeled as a sum of a stochastic noise process and the actual physical measurement as represented in the sensor body frame **b** [22]. Assuming $\mathbf{R}_n^b$ represents the rotation from the local frame **n** to the sensor body frame **b**, the model is expressed as:

$$\mathbf{a}_b = \mathbf{R}_n^b(\mathbf{a}_n - \mathbf{g}_n) + \mathbf{a}_e$$

where $\mathbf{a}_b$ is the sensor body acceleration, $\mathbf{a}_n$ is the acceleration as expressed in the local frame, $\mathbf{g}_n$ is the gravitational acceleration in vector form, $\mathbf{R}_n^b$ is the transformation matrix that rotates from the local frame to the sensor body frame, and $\mathbf{a}_e$ is a stochastic process model for the accelerometer error [8]. A common model for the error is called the Random Walk model, which consists of a white Gaussian noise component and a random time-varying bias. Figure 4.4 describes the frames involved as well as the variables of interest; namely, the *measured* sensor-body acceleration $\mathbf{a}_b$ and the rotation described by $\mathbf{R}_n^b$, which can be *partially* estimated from the latter.

4.2.2. Gyroscope

Gyro measurements are modeled as the sum of a stochastic noise process component and the actual rotational rate of the sensor. The output measurements are expressed in the sensor body frame **b** as reflected in figure 4.5. It is worthwhile to note that the gyroscope also measures the angular velocity of the earth rotation about its own axis, but it is very small

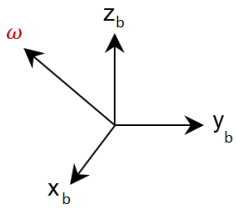


Figure 4.5: Angular Velocity in the Sensor Body Frame **b**

and negligible compared to the sensor rotation for this application justifying discarding it in calculations [8]. Accordingly, the Coriolis and centrifugal accelerations in the accelerometer model are also discarded and do not enter $\mathbf{a}_n$ - the acceleration in the local frame **n**. The model is expressed as:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_b + \boldsymbol{\omega}_e$$

The angular velocity output measured by the sensor is $\boldsymbol{\omega}$, $\boldsymbol{\omega}_b$ is the true angular velocity, and $\boldsymbol{\omega}_e$ is a stochastic process model for the gyroscopic error. The gyroscopic error is also often assumed to follow the Random Walk model [8], with a white noise component and a time-varying bias that is either given in datasheets or estimated.

4.2.3. Magnetometer

Similar to the accelerometer, the magnetic vector is modeled as a sum of the physical magnetic intensity measurement and a stochastic noise process term. The error in the magnetometer consists of two components: the first being the hard-iron bias or offset, and the second is what's called the soft-iron bias, which is much harder to quantify and is a matrix that multiplies the true magnetic field vector. Temperature, active magnetic fields, and other factors contribute to soft-iron bias. The soft-iron bias distorts the orthogonality of the sensor measurement. Many solutions have been proposed in the literature to quantify the soft-iron bias. However, the vast majority of the methods achieve static calibration, which is not very useful if the sensors experience a drastic change in the magnetic environment. Ref. [13] formulates the magnetic calibration task as the solution to a maximum-likelihood problem,

Ref. [10] exploits the in-variance of the dot product of the gravitational and magnetic fields vectors under rotation and uses a set of measurements to minimize an objective function and solve for the hard-iron and soft-iron biases. Ref. [29] also formulates the problem as a maximum likelihood estimator that operates in the *sensor* frame, decoupling the calibration problem from the attitude estimation problem. Finally, Ref. [30] collects a set of magnetic field measurements and fits them to an ellipsoid, the parameters of which give the biases. However, this fitting method may apply a rotation to the soft-iron bias matrix [10] that we have no knowledge of.

In this work, magnetometer calibration was done by first collecting a few tens of thousands of measurements around the laboratory and then determining an offset and a simple diagonal scale matrix without applying any transformations to the magnetic field measurements. The offset and the scale matrix evaluated depend on the local magnetic environment and should be evaluated for every new location the sensor module operates in as part of an initial calibration process. This method has proved to provide consistent heading accuracy estimates with a predictable error. The magnetometer model is expressed as:

$$\mathbf{m}_b = \mathbf{R}_n^b \mathbf{m}_n + \mathbf{m}_e$$

where $\mathbf{m}_b$ is the magnetic field intensity measured by the magnetometer in the sensor body frame $\mathbf{b}$, $\mathbf{m}_n$ is the magnetic field intensity in the local frame $\mathbf{n}$ (Earth magnetic field), $\mathbf{R}_n^b$ is as defined for the accelerometer, and $\mathbf{m}_e$ is a stochastic error process model to account for magnetic disturbance. The error process for the magnetometer is a sum of white Gaussian noise as well as a bias term that includes the effect of both the *hard-iron* and *soft-iron* biases.

4.2.4. Global Gravity and Magnetic Models

For the purposes of the application in this work, a constant gravitational model is suitable. The gravity vector is assumed to act downwards towards the center of the Earth. Assuming an ENU local frame, the gravity model is defined as:

$$\mathbf{g}_n = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

where g is the gravitational constant. It varies slightly from $9.81\frac{m}{s^2}$ across different locations on Earth. The simple global model considered for the magnetometer is defined in terms of the inclination and declination angles, which depend on the way the magnetic vector penetrates the specific location. Complex and recent magnetic models and calculators are available [16], but they are too complex to incorporate into the process models of estimators. Assuming an ENU local frame, the magnetic model considered is defined as:

$$\mathbf{m}_n = m_t \begin{bmatrix} \cos I \sin D \\ \cos I \cos D \\ -\sin I \end{bmatrix}$$

where m_t is the magnitude of the total field, I is the inclination angle, and D is the declination angle. The inclination and declination angles for a specific location can be found from the IGRF.

4.3. Orientation Estimation

The Extended Kalman Filter (EKF): the EKF is an extension to the original linear Kalman filter. Using the Jacobian of the nonlinear model *with respect to* the state, the EKF linearizes the nonlinear process model to evaluate a generic linear Kalman filter at every time step Δt. The convergence of the EKF is heavily dependent on the trueness of the initial state and covariance estimates [8]. The full derivation of the orientation EKF is provided in appendix A. The work flow in the EKF for orientation estimation proceeds in the following manner (outlined in Table 4.1):

1. Following selection of appropriate global gravity and magnetic field models, a first static measurement of acceleration $\mathbf{a}_1$ and magnetic field $\mathbf{m}_1$ is collected. This measurement set coupled with the Earth's fields $\mathbf{g}_n$ and $\mathbf{m}_n$ are used in the TRIAD [34] (Three Axis Attitude Determination) method to find the direction cosine matrix of the sensor body frame relative to the absolute local frame. This results in the initial orientation estimate, parameterized as a unit quaternion $\mathbf{q}_1^{nb}$ (Table 4.1). Notation of the aforementioned quaternion indicates a rotation from body frame $\mathbf{b}$ to local frame $\mathbf{n}$. In Table 4.1, $\mathbf{P}_1$ is the initial state covariance, $\mathbf{P}_e$ is the uncertainty associated with an assumed initial error in the rotation vector resulting from $\mathbf{q}_1^{nb}$. The error assumed with the initial measurement is 0.3 rad [8]. $\mathbf{G}_t$ is the continuous-time state error matrix as defined in equation A.7.

2. In the process model, the angular velocity measurement is converted into a unit quaternion $\exp_q(\boldsymbol{\omega}_b T)$, where T is the sampling period of the measurement, which is then applied to the initial quaternion using the quaternion product yielding the new predicted estimate. The state covariance $\mathbf{P}_{k-1}$ is time-propagated as explained in appendix A to evaluate the predicted state covariance $\hat{\mathbf{P}}_k$ at the current time step. The predicted orientation estimate $\hat{\mathbf{q}}_k^{nb}$ and the predicted orientation covariance estimate $\hat{\mathbf{P}}_k$ are the results of this step. See Table 4.1. The discretized process noise matrix $\mathbf{Q}_d$ and state transition matrix $\boldsymbol{\phi}_k$ is as defined in table A.3. Table 4.1 summarizes the steps.

3. The newly-measured acceleration and magnetic measurement $\mathbf{z}_k$ are compared against those evaluated from the measurement function $\mathbf{y}_k$ (evaluated from the predicted state $\hat{\mathbf{q}}_k^{nb}$) to evaluate the residual $\mathbf{r}_k$. The state Jacobian $\mathbf{H}_k$ of the measurement function $\mathbf{y}_k$ is evaluated, which then allows the evaluation of the Kalman gain $\mathbf{K}_k$. The measurement noise matrix $\mathbf{R}$ is defined by equation A.10 and depends on the sensor noise characteristics. The Kalman gain $\mathbf{K}_k$ is then used to find the corrected estimate $\mathbf{q}_k^{nb}$ and the corrected state covariance $\mathbf{P}_k$ (see Table 4.1). Figure 4.6 shows the EKF computations for orientation estimation. Table 4.1 also summarizes the equations needed for the initialization, prediction and update steps.

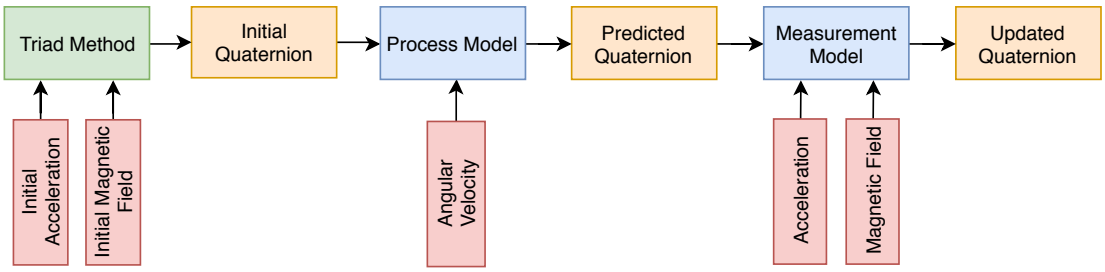


Figure 4.6: Orientation Estimation via the Extended Kalman Filter

Initialization	$\mathbf{q}_1^{nb} = \mathbf{f}_q(\mathbf{m}_1, \mathbf{a}_1)$ $\mathbf{P}_1 = \mathbf{f}_P(\boldsymbol{\omega}_e) = \mathbf{G}_t \mathbf{P}_e \mathbf{G}_t^\top$
Prediction	$\hat{\mathbf{q}}_k^{nb} = \mathbf{q}_{k-1}^{nb} \odot \exp_q(\boldsymbol{\omega} T)$ $\hat{\mathbf{P}}_k = \boldsymbol{\phi}_k \mathbf{P}_{k-1} \boldsymbol{\phi}_k^\top + \mathbf{Q}_d$
Update	$\mathbf{r}_k = \mathbf{z}_k - \mathbf{y}_k$ $\mathbf{q}_k^{nb} = \hat{\mathbf{q}}_k^{nb} + \mathbf{K}\mathbf{r}_k$ $\mathbf{K}_k = \hat{\mathbf{P}}_k {\mathbf{H}_k}^\top (\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^\top + \mathbf{R})^{-1}$ $\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}(\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^\top + \mathbf{R})\mathbf{K}_k^\top$

Table 4.1: EKF Computation Flow

4.4. Error-state Kalman Filter: Position and Velocity Estimation using ZUPT(s)

The evaluation of cadence, gait speed, and stride length requires the evaluation of linear velocity. It is generally difficult to obtain a good linear velocity estimate from low-cost IMU(s), as it is a function of orientation and any error of which will be integrated into the linear velocity term, resulting in a fast error accumulation if orientation is not accurate. To combat this issue, the concept of zero-velocity updates is used. Zero-Velocity Updates (ZUPT(s)) refer to resetting the linear velocity term to zero when the foot is flat on the ground during the bipedal gait cycle. The velocity reset is done in the framework of an *error-state Kalman filter* based on the work of [22]. A Kalman framework is beneficial, instead of naive integration because it allows the correlation of orientation errors in other states in an intuitive manner, which affords corrections to be made in the update step once the errors are observable [22] [6]. Figure 4.7 shows the computation flow in the ESKF. Table

4.2 also summarizes the evaluations needed for every time step.

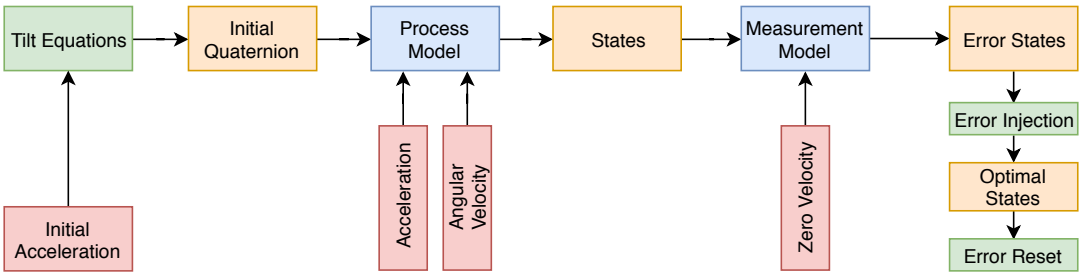


Figure 4.7: Kinematics Estimation Using the Error-State Kalman Filter

Computation steps of the linear ESKF (error-state Kalman filter) are summarized in Table 4.2 and described below:

1. Initializing the orientation state $\mathbf{q}_1^{nb}$ based on the first accelerometer measurement $\mathbf{a}_1$. The initial position $\mathbf{p}_1$, velocity $\mathbf{v}_1$, error state $\delta\mathbf{x}_1$, and error state covariance $\mathbf{P}_1$ are initialized to zero as shown in the initialization section of table 4.2.

2. In the prediction step of the filter, the orientation $\hat{\mathbf{q}}_k^{nb}$, linear velocity $\hat{\mathbf{v}}_k$, and linear position $\hat{\mathbf{p}}_k$ predicted *nominal* states are evaluated based on rigid-body kinematics using the sensor linear accelerations $\mathbf{a}_b$ and angular velocities $\boldsymbol{\omega}$ as inputs.

3. The process model for the error state is defined by $\mathbf{f}$. Knowledge of the Jacobian $\mathbf{F}_x$ ($\mathbf{f}$ with respect to $\delta\mathbf{x}$) and the Jacobian $\mathbf{F}_i$ ($\mathbf{f}$ with respect to $\mathbf{e}$) allows for evaluating the predicted covariance $\hat{\mathbf{P}}_k$ of the error state $\delta\mathbf{x}$. The process noise covariance matrix $\mathbf{Q}$ is as defined in equation B.3.

4. In the update step of the filter, the *pseudo* linear velocity *measurement* $\mathbf{z}_k$ of zero is used to update the error state. The measurement function $\mathbf{y}_k$ represents velocity while $\mathbf{Y}_i$ represents the uncertainty in the velocity measurement. The residual $\mathbf{r}_k$ is evaluated from the *pseudo* measurement and the evaluated measurement function. Evaluating the Jacobian $\mathbf{H}_k$ ($\mathbf{y}_k$ with respect to $\delta\mathbf{x}$) then allows for the evaluation of the Kalman gain $\mathbf{K}_k$. Thereafter, the error states $\delta\mathbf{x}_k$ become observable and are subsequently injected into the nominal states as shown in the last three entries in the update section of table 4.2 to obtain the corrected position estimate $\mathbf{p}_k$, the corrected velocity estimate

$\mathbf{v}_k$, and the corrected quaternion estimate $\mathbf{q}_k^{nb}$. Detailed computations of the ESKF are in appendix B.

Error State Model	$\delta\mathbf{p}_k = \delta\mathbf{p}_{k-1} + \delta\mathbf{v}_{k-1}\Delta t$ $\delta\mathbf{v}_k = \delta\mathbf{v}_{k-1} - \mathbf{R}_b^n[\mathbf{a}_b]_\times \delta\boldsymbol{\theta}_{k-1}\Delta t + \mathbf{v}_i$ $\delta\boldsymbol{\theta}_k = \mathbf{R}_b^{n\top}\delta\boldsymbol{\theta}_{k-1} + \boldsymbol{\theta}_i$ $\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta}\end{bmatrix}^\top$ $\mathbf{f} = \begin{bmatrix}\delta\mathbf{p}_k & \delta\mathbf{v}_k & \delta\boldsymbol{\theta}_k\end{bmatrix}^\top$ $\mathbf{e} = \begin{bmatrix}\mathbf{v}_i & \boldsymbol{\theta}_i\end{bmatrix}^\top$
Initialization	$\mathbf{q}_1^{nb} = \mathbf{f}_q(\mathbf{a}_1)$ $\mathbf{p}_1 = \mathbf{0}_{1\times 3}$ $\mathbf{v}_1 = \mathbf{0}_{1\times 3}$ $\boldsymbol{\delta x}_1 = \mathbf{0}_{1\times 9}$ $\mathbf{P}_1 = \mathbf{0}_{9\times 9}$
Prediction	$\hat{\mathbf{q}}_k^{nb} = \mathbf{q}_{k-1}^{nb} \odot \exp_q(\boldsymbol{\omega} T)$ $\hat{\mathbf{p}}_k = \mathbf{p}_{k-1} + \mathbf{v}_{k-1}\Delta t + \frac{1}{2}(\mathbf{R}_b^n\mathbf{a}_b + \mathbf{g}_n)\Delta^2 t$ $\hat{\mathbf{v}}_k = \mathbf{v}_{k-1} + (\mathbf{R}_b^n\mathbf{a}_b + \mathbf{g}_n)\Delta t$ $\hat{\mathbf{P}}_k = \mathbf{F}_x\mathbf{P}_{k-1}\mathbf{F}_x^\top + \mathbf{F}_i\mathbf{Q}\mathbf{F}_i^\top$
Update	$\mathbf{y}_k = \mathbf{v}_k + (\mathbf{R}_b^n\mathbf{a}_b + \mathbf{g}_n)\Delta t + \mathbf{Y}_i$ $\mathbf{z}_k = \mathbf{0}_{1\times 3}$ $\mathbf{r}_k = \mathbf{z}_k - \mathbf{y}_k$ $\mathbf{K}_k = \hat{\mathbf{P}}_k\mathbf{H}_k^\top(\mathbf{H}_k\hat{\mathbf{P}}_k\mathbf{H}_k^\top + \mathbf{Y}_i)^{-1}$ $\delta\mathbf{x}_k = \mathbf{K}_k(\mathbf{z}_k - \mathbf{y}_k)$ $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{P}}_k$ $\mathbf{p}_k = \hat{\mathbf{p}}_k + \delta\mathbf{p}_k$ $\mathbf{v}_k = \hat{\mathbf{v}}_k + \delta\mathbf{v}_k$ $\mathbf{q}_k^{nb} = \hat{\mathbf{q}}_k^{nb} \odot \exp_q(\delta\boldsymbol{\theta}_k)$

Table 4.2: ESKF Computation Flow

Chapter 5

System Auto-calibration

5.1. What is Auto-calibration?

A significant source of error in motion estimation using inertial sensors is the erroneous placement of the sensors on body segments (feet in this application). Assuming a correct module alignment $\mathbf{q}_{SI}$ with the foot using a functional calibration procedure [17], where S stands for segment (foot) and I stands for the IMU module, then auto-calibration refers to including the effect of arbitrary rotation of the sensor module on a tangent surface to the foot with respect to $\mathbf{q}_{SI}$, effectively modeling the degree of freedom that is likely to introduce errors. The goal of auto-calibration is to find a rotation paramterization for the IMU sensor module relative to the segment (foot), denoted by the quaternion $\hat{\mathbf{q}}_{SI}$ or rotation matrix $\hat{\mathbf{R}}_{SI}$. Figure 5.1 depicts the angular degree of freedom of interest; the rotation normal to a surface tangent of the segment (foot).

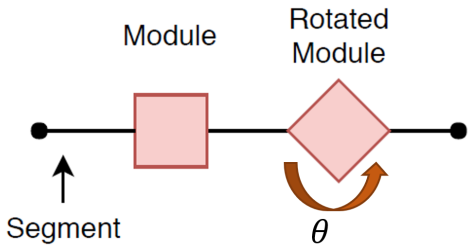


Figure 5.1: Modelled Degree of Freedom

Regression machine learning architectures have the ability to model complex parametric models. Acceleration, gyroscopic, and magnetic signals hold complete information about orientation, therefore, the proposed input to the regression problem is a $\mathbf{1} \times \mathbf{9}$ vector of the $\mathbf{1} \times \mathbf{3}$ acceleration signal, the $\mathbf{1} \times \mathbf{3}$ gyroscopic signal, and the $\mathbf{1} \times \mathbf{3}$ magnetic signal. The

magnetic signal is necessary to correct for gyroscopic drift. The proposed output is the $\mathbf{1} \times \mathbf{4}$ orientation quaternion $\hat{\mathbf{q}}_{SI}$ of the sensor module relative to the foot. Theoretically speaking, if motion capture data defining global position and orientation trajectories of an object (foot in this case) is available, then IMU data can be obtained by differentiation [23] [25] if corresponding IMU data is not available. The performance metric for the modeling problem is the RMSE (*Root Mean Squared Error*). A perfect neural model would result in a *zero* RMSE - implying a mathematical relationship that perfectly maps the $\mathbf{1} \times \mathbf{9}$ inputs to the $\mathbf{1} \times \mathbf{4}$ orientation quaternions $\hat{\mathbf{q}}_{SI}$. Such a model would provide a way to find the orientation of the foot relative to the sensor module as described below.

5.2. Methodology

1. Attaching a sensor module to a foot, $\mathbf{q}_{SI}$ is found using a functional calibration method [17]. In this method, two static poses are used to collect gravity vector measurements from the sensor module, one pose when the subject is standing up and the other when the subject is lying down. Coupled with a normal vector found from the cross-product of the previous two sets of measurements, it is now possible to evaluate $\mathbf{q}_{SI}$. In this work, $\mathbf{q}_{SI}$ is evaluated by knowledge of the orientation quaternion of the segment and the orientation quaternion of the IMU.

2. The sensor module is then rotated about its normal axis by θ such that its new $\hat{\mathbf{q}}_{SI}$ orientations are described by:

$$\hat{\mathbf{q}}_{SI} = \mathbf{q}_{SI} \odot \mathbf{q}_{\theta}$$

   where $\mathbf{q}_{\theta}$ parameterises the arbitrary degree of freedom we are interested in modelling. By manually rotating the sensor module to span 360° from the starting orientation $\mathbf{q}_{SI}$, we have information mapping the IMU sensor measurements; linear accelerations, angular velocities, and magnetic measurements to specific $\hat{\mathbf{q}}_{SI}$. Knowledge of $\hat{\mathbf{q}}_{SI}$ and $\mathbf{q}_{SI}$, it is now possible to evaluate $\mathbf{q}_{\theta}$.

3. Having a neural model to evaluate the angular distance $\mathbf{q}_{\theta}$ from a *correctly-aligned* sensor on the foot at $\mathbf{q}_{SI}$ allows for arbitrary orientation of the sensor module on the

foot. Multiplying the orientation quaternion of an arbitrarily-oriented sensor module on the foot by the inverse of $\mathbf{q}_\theta$ yields the orientation quaternion as would be obtained by a *correctly-aligned* sensor module.

4. IMU measurements constitute the input to the regression neural network and the corresponding $\hat{\mathbf{q}}_{SI}$ constitute the output.

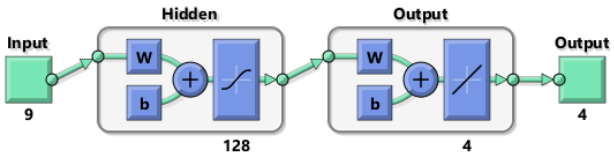


Figure 5.2: Proposed Neural Architecture

The MATLAB neural fitting tool is used to construct and train the neural network. The data set (TotalCapture) [26] used to train the network consists of 11 walking trials of five subjects. Data from 9 walking trials of three subjects are used to train and test the network while both walking trials of subjects 4 and 5 are used for further validation the network (unseen subjects). The data set contains motion capture data for all body segments but only data for the right foot is used in this work. In total, 31068 data points sampled at 60Hz are used for training, validation, and testing of the regression model while 7114 data points are used for testing on unseen subjects. Every data point contains the $\mathbf{1} \times \mathbf{3}$ global position and the $\mathbf{1} \times \mathbf{4}$ global orientation trajectories of the corresponding object. Synthesizing IMU data from the global trajectories as described above, the proposed network has one input layer of size 9, a hidden layer with 128 neurons or parameters, and an output layer of size 4. Details about the proposed network are shown in table 5.1 below.

Proposed Network Architecture	
Input Size	9
Output Size	4
Number of Neurons	128
Training Data Percent	70%
Testing Data Percent	15%
Validation Data Percent	15%
Performance Metrics	RMSE (Root Mean Square Error)

Table 5.1: Proposed Network Architecture

Chapter 6

RESULTS AND DISCUSSIONS

6.1. Data Evaluation

To verify the ability of the sensors to produce accurate orientation estimates, the performance of the EKF is verified against an optical measurement system. In these evaluations, rotations are parameterized using the Euler formulation. Sensor data as well as optical data are sampled at a 100Hz. Time-stamping for the sensor signals is evaluated such that delay components, such as communication delay and sensor reporting delay, are taken into account. Data is then re-sampled again to exactly match the sampling frequency of the optical system. It is worthwhile to note that the static and dynamic accuracy of the IMU sensor used is 2° and 3.5°, respectively, when the sensor is factory-calibrated [7], meaning that in the best of scenarios, those values for the errors should generally be assumed, on average. EKF-filtered estimates are compared with both unfiltered and optical (reference) estimates.

6.2. Orientation Estimation

In an effort to benchmark the quality of orientation estimates of the EKF, three sets of measurements are collected, each consisting of 2000 data points recorded at 100Hz. The data sets are collected off of a sensor module that is randomly rotated by hand. Example plots for one data set are shown in figures 6.1, 6.2 and 6.3. It is clear from the plot figures that the EKF-filtered estimates better track the reference estimates for all three Euler angles. "Reference" in the legend of the figures refers to optical data. Similar results for the other two sets are obtained.

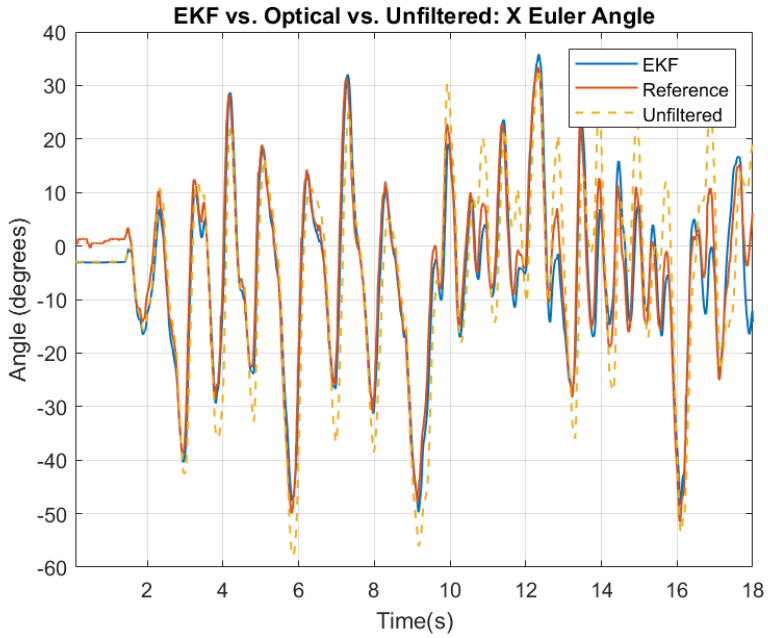


Figure 6.1: X-Euler Angle: EKF (blue) vs. Unfiltered (yellow) vs. Optical (orange)

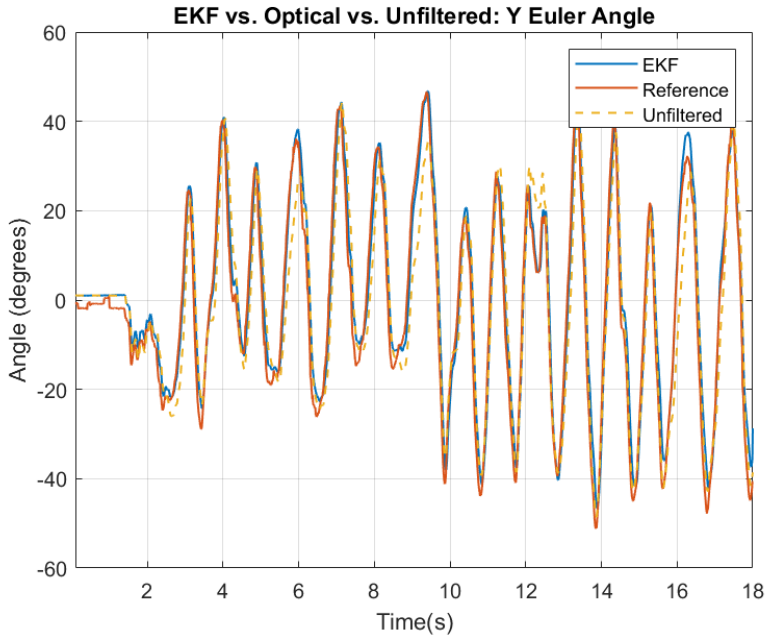


Figure 6.2: Y-Euler Angle: EKF (blue) vs. Unfiltered (yellow) vs. Optical (orange)

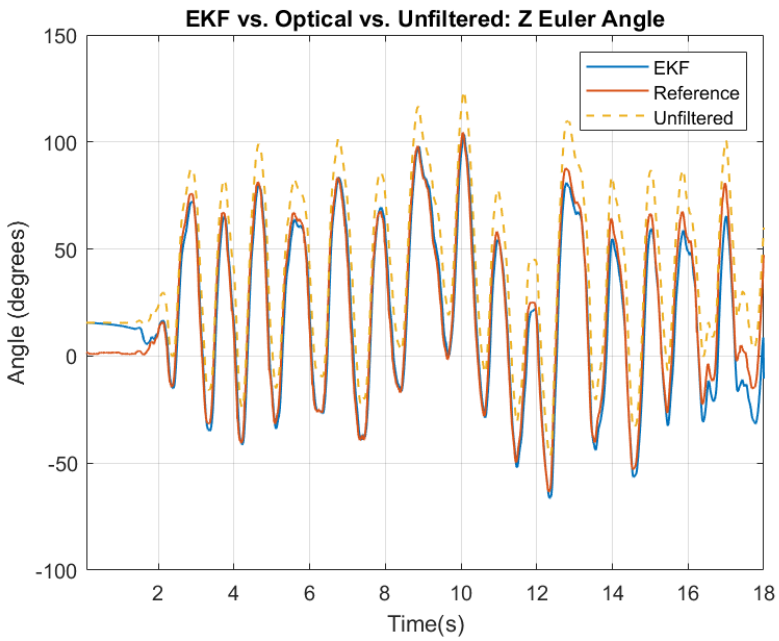


Figure 6.3: Z-Euler Angle: EKF (blue) vs. Unfiltered (yellow) vs. Optical (orange)

Trial no.	**X-Max. Error**	**Y-Max. Error**	**Z-Max. Error**	$\boldsymbol{\sigma}_x$	$\boldsymbol{\sigma}_y$	$\boldsymbol{\sigma}_z$
Trial 1	12.8°	5.4°	16.2°	3.6°	2.3°	4.9°
Trial 2	13.7°	5.8°	27°	7.9°	6.4°	24.6°
Trial 3	10°	5.4°	8.8°	18.1°	6.9°	13.1°

Table 6.1: EKF Orientation Errors

Large errors in the initial few estimates are a result of the initial covariance assumed for the Kalman filter, which is evaluated based on a 0.3 radian angular error. This is done to give the filter enough leeway to converge in case of large errors in the initial quaternion estimate [8]. It is also worth noting that the laboratory environment contains a large amount of metallic frames and material that severely corrupts the magnetic measurement, which in turn corrupts orientation. Results for the two other trials along with the aforementioned are summarized in table 6.1.

6.3. Position and Velocity Estimation

Procedure: Optical position measurements are plotted against the position evaluated from the sensor module to measure performance. The RMSE is the chosen metric to quantify the performance of the filter in estimating position and velocity in the walking direction. The premise is that if the sensor module is capable of producing accurate position estimates in the forward direction as compared to the optical system, then it can be faithfully used for gait evaluation. The data set used for tuning the filter consists of 50 seconds of walking data captured at 100Hz. The data is captured from a single subject (28 year old male, 5 feet 8 inches, 180 lbs) walking back and forth in the laboratory environment. Filter noise (from both the accelerometer and the gyroscope) as well as the ZUPT threshold are manually tuned until good position tracking is observed. The RMSE of the ESKF estimate (shown in figure 6.4) evaluated is $\approx 0.45m$ which translates to an NRMSE (normalized by peak) of $\approx 5.8\%$. Figure 6.5 shows the unfiltered position evaluation for the same data set. The unfiltered position is evaluated directly from the kinematic equations without ZUPT(s). The RMSE for the unfiltered evaluation is $\approx 129.3m$. It is clear from figure 6.5 that position errors accumulate very quickly without ZUPT(s) in the ESKF framework.

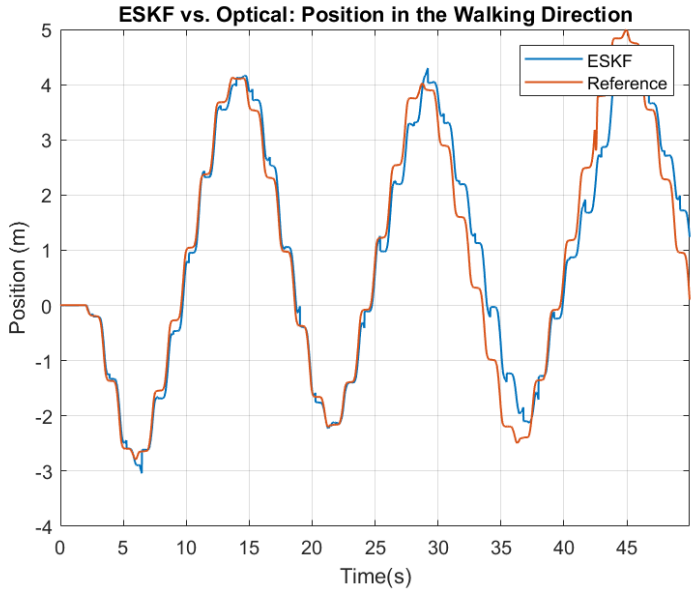


Figure 6.4: Forward Position - Trial 1: ESKF (blue) vs. Optical (orange)

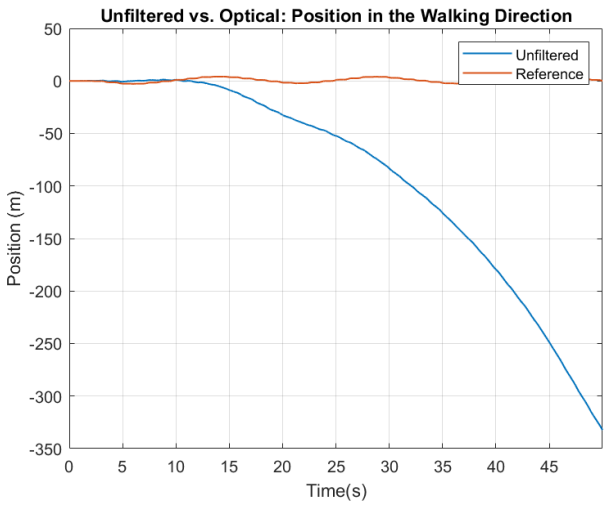


Figure 6.5: Forward Position - Trial 1: Unfiltered (blue) vs. Optical (orange)

Next is a comparison between the estimated walking velocity, the walking velocity evaluated from the optical position, and the walking velocity evaluated without ZUPT(s) for the same data set used above. Velocity is obtained from optical position by evaluating the gradient. Figure 6.6 shows the speed-time plot. It is clear from figure 6.6 the need for estimating velocity in the ESKF framework utilizing ZUPT(s). The speed RMSE of the ESKF estimate evaluated is $\approx 0.32\frac{m}{s}$, which translates to an NRMSE of $\approx 12.4\%$. The RMSE of the unfiltered estimate is $\approx 7.1\frac{m}{s}$.

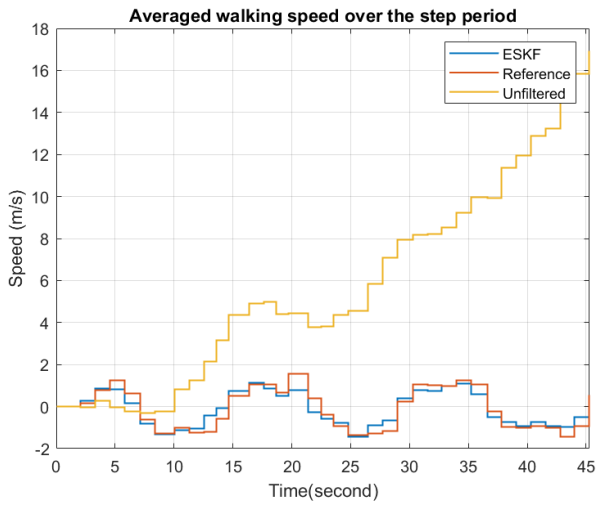


Figure 6.6: Forward Speed - Trial 1: ESKF (blue) vs. Optical (orange) vs. Unfiltered (yellow)

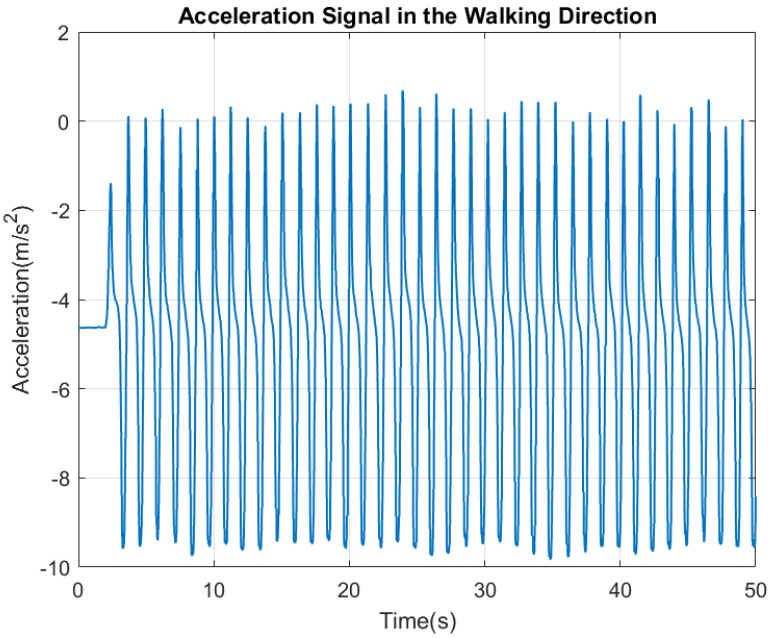


Figure 6.7: Acceleration Peaks during Ground Contact - Trial 1

Discussion: The tunable parameters in the ESKF used for gait evaluation are summarized in table 6.2. The accelerometer and gyroscope noise densities represent the standard deviation of measurement noise in the corresponding sensor, which is used to form the process noise matrix **Q** in the ESKF, while the noise density term for linear velocity represents

the standard deviation (uncertainty) in the *pseudo* linear velocity measurement and is used to form the **R** matrix in the ESKF. The ZUPT threshold for angular velocity represents the norm below which the foot is assumed to lay flat on the ground, triggering a ZUPT in the filter. Four trial data sets have been collected to verify the performance of the ESKF for position and velocity estimation. The Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE) are used to evaluate performance against optical reference measurements. The NRMSE normalizes the estimate by its peak. The results are shown in table 6.4. The NRMSE values for the forward velocity (walking speed) estimate show promise in using the sensor module for gait evaluation and assessment. Next we present results for cadence and stride.

Tunable ESKF Parameters		
Accelerometer Noise Density	0.030	m/s^2
Gyroscope Noise Density	0.002	rad/s
Linear Velocity Noise Density	0.001	m/s
Angular Velocity Norm ZUPT threshold	0.550	rad/s

Table 6.2: Values for the ESKF Tunable Parameters

RMSE and NRMSE of Forward Position and Velocity					
Trial	Pos. RMSE(m)	Pos. NRMSE(%)	Vel. RMSE(m/s)	Vel. NRMSE(%)	Duration(s)
Trial 1	0.4487	5.8	0.3197	12.4	50
Trial 2	0.2961	4.8	0.3589	17.9	20
Trial 3	0.5309	7.5	0.3538	13.6	30
Trial 4	0.1832	2.55	0.3036	15.22	20

Table 6.3: RMSE and NRMSE of Forward Position and Velocity

6.4. Gait Metrics

Evaluating the stride length, speed, and cadence requires knowledge of the time taken to complete a full step. During walking, ground contact of the foot causes the acceleration signals to hit a local peak [24]. By detection of those peaks seen in figure 6.7 and their associated timestamps, we are able to evaluate the step time.

6.4.1. Gait Speed

Gait speed is evaluated as the speed average value over the time interval of one step - the integral of the absolute velocity-time profile divided by the interval. It is expressed in $\frac{m}{s}$. Figure 6.6 shows the gait speed for the data set discussed above for position and velocity - trial 1 in table 6.4. Note that gait speed is represented by the estimated forward velocity discussed in the previous section. Table 6.4 lists the RMSE and NRMSE (normalized by the peak) values for the four recorded trials.

6.4.2. Cadence

Cadence expresses the number of steps taken per minute. Just as gait speed, this is also evaluated per step period. Figure 6.8 shows the cadence for the first trial. Cadence is evaluated as the inverse value of time between two acceleration peaks - the time it takes to complete a full step.

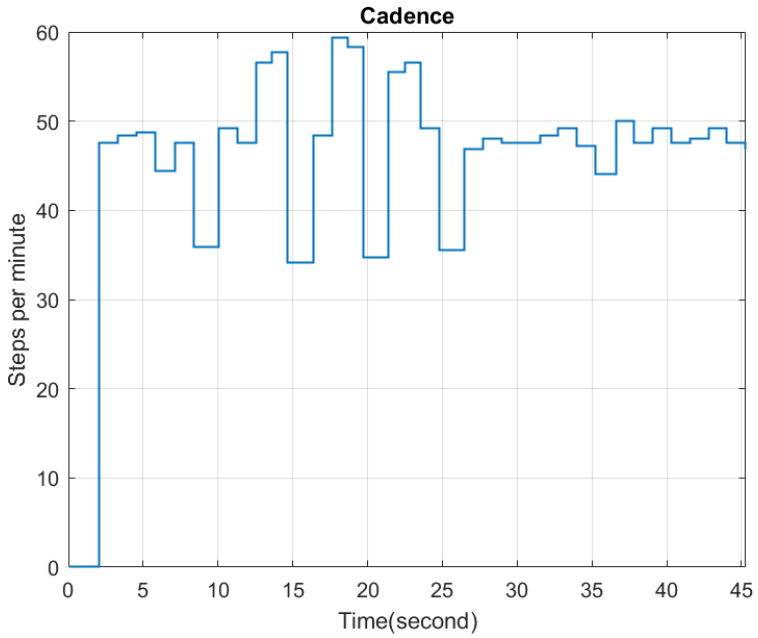


Figure 6.8: Cadence - Trial 1

6.4.3. Stride

Stride length represents the distance measured between two heel-strike events of the same foot. It is expressed in meters. Figure 6.9 shows the stride lengths for the first trial. The RMSE of the ESKF estimate is found to be $\approx 0.4371m$, which corresponds to an NRMSE (normalized by peak) of $\approx 10.9\%$. The unfiltered stride length evaluation has an RMSE of $\approx 9m$, which, again, highlights the value of ZUPT(s) in the ESKF framework.

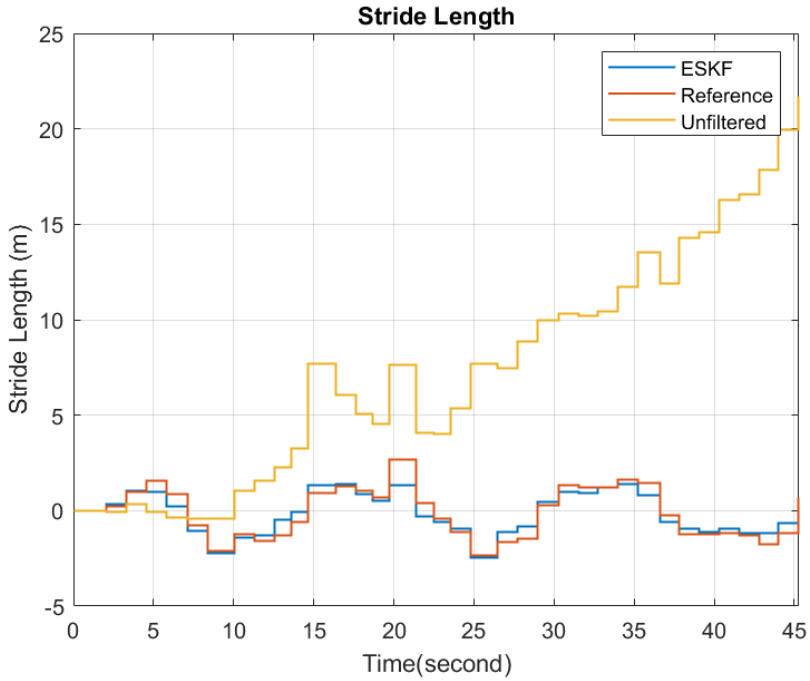


Figure 6.9: Stride Length - Trial 1

RMSE and NRMSE of Stride Length			
Trial	Stride Length RMSE(m)	Stride Length NRMSE(%)	Duration(s)
Trial 1	0.4210	10.9	50
Trial 2	0.4500	14.9	20
Trial 3	0.5030	11.3	30
Trial 4	0.4253	15.2	20

Table 6.4: RMSE and NRMSE of Stride Length

6.4.4. Discussion

In the four trials of walking data we have evaluated, position estimates yielded an average RMSE of $\approx 0.365m$ and an average NRMSE of $\approx 5.2\%$, speed estimates yielded an average RMSE of $\approx 0.334\frac{m}{s}$ and an average NRMSE of $\approx 14.8\%$, and stride length estimates yielded an average RMSE of $\approx 0.45m$ an an average NRMSE of $\approx 13\%$. In addition, the average

correlation coefficient is evaluated for each of the quantities in all four trials. The average correlation coefficient between the ESKF position estimates and reference measurements is found to be 0.99, 0.90 for the ESKF speed estimates, and 0.90 for the ESKF stride length estimates. This shows the viability of using IMU signals to estimate forward position, velocity, gait speed, cadence, and the stride length. By simply attaching a sensor module to the foot, important gait information are estimated for gait pathology.

6.5. Auto-calibration

Here we explore the performance of a neural model that captures the effect of rotational variability (angular offset from correct alignment) in the axis that is normal to the foot the sensor module is attached to. A neural network architecture with one input layer, one hidden layer, and one output layer is proposed. Having a model that accurately evaluates the rotational distance from a *correctly-aligned* sensor module using IMU signals (accelerations, angular velocities, and magnetic signals) would minimize human error when placing those sensor modules on feet.

6.5.1. Results

Model training is halted when validation performance indicates no fitting improvement of the model with the outputs supplied. To afford easier interpretation of errors, the Euler angles representation is used. Note that in the data set, the RMSE of the calibrated inertial orientation with respect to the orientation obtained from the optical system for all subjects and walking trials is found to be X: 4.2°, Y: 3.8°, and Z: 2.5° in Euler angles form. Following training the model, the RMSE(s) for different rotational variability ranges are summarized in table 6.5.

RMSE in Euler Angles Form		
Variability	RMSE (Train, Valid, Test)	RMSE (Unseen Subject Data)
0°	X: 5.4° - Y: 6.7° - Z: 3.9°	X: 7.1° - Y: 6.2° - Z: 14.7°
10°	X: 5.8° - Y: 7.2° - Z: 4.1°	X: 7.4° - Y: 6.7° - Z: 14.6°
45°	X: 8.4° - Y: 12.3° - Z: 7.1°	X: 11.1° - Y: 11.3° - Z: 14.5°
180°	X: 41.3° - Y: 25.3° - Z: 36°	X: 34.7° - Y: 24° - Z: 37.7°

Table 6.5: RMSE

Rotating the sensor module within 10° about the Z-axis results in an RMSE of 4.1°, 7.1° when rotated within 45°, and 36° when rotated within 180°. Unseen data (subjects 4

and 5) results in an RMSE of 14.6°, 14.5°, and 37.7° for the same variability ranges. As expected, the RMSE values for the 10° and the 45° cases are higher for the the unseen subjects data. A larger data set that includes more subjects would cover a wider range of human gait motion and result in a better model that is less likely to over-fit, which would improve testing performance on unseen data. It would also better highlight the feasibility and practicality of using neural models to predict angular offsets. The extreme case of 180° produces a larger angular error of 37.7° on unseen subjects data, highlighting the limits of a neural model based on a small data set.

Chapter 7

CONCLUSIONS

An inertial-based, wireless wearable is developed to aid in diagnostics of gait pathology through evaluation of gait parameters such as speed, stride, and cadence, as well as the orientation of the sensor. The module is also capable of estimating position $\mathbf{p}$ with an acceptable accuracy for a short period of time ($\approx 25s$). A neural model is developed to evaluate the orientation of an arbitrarily-placed sensor module on the foot with respect to the orientation of a *correctly-aligned* module. This is done in an attempt to minimize the effect of foot mounting errors in the normal axis. Allowing a 10° range variability produced $\approx 14.6^\circ$ RMSE while a 45° produced $\approx 14.5^\circ$ RMSE on unseen subjects. While this shows promise in using data-driven techniques to mitigate placement-induced errors of the sensor module, the magnitude of the errors highlights the challenges of using inertial sensors for accurate motion tracking. That being said, for short periods of time not exceeding 50s, the sensor module provided speed estimates with an average NRMSE of $\approx 14.8\%$ over 4 trials and an average NRMSE of $\approx 13\%$ for stride length estimates of the same trials. This suggests that while the module could be useful in providing qualitative assessments of gait (slower gait speed and shorter stride length in schizophrenia, rapid gait speed in mania, among others) [20], it lacks the high accuracy and precision needed in many robotics applications. In a larger scope of this effort, multiple subjects would be used to provide the sensor signals needed to train the network as a larger pool of people provide better data variability and better model generalization performance. An ESKF is developed to estimate linear velocity $\mathbf{v}$ and position $\mathbf{p}$ of the sensor module. Using linear velocity estimates $\mathbf{v}$ from the ESKF coupled with acceleration peaks, walking speed, cadence, and stride length are evaluated as demonstrated in earlier sections.

Appendix A

Derivation of the EKF for Orientation Estimation

A.1. Sensor Models

Accelerometer	$\mathbf{a}_b = \mathbf{R}_n^b(\mathbf{a}_n - \mathbf{g}_n) + \mathbf{a}_e$	Accelerometer Noise Density	$\mathbf{a}_e = \mathcal{N}(0, \sigma_a^2\mathbf{I}_3)$
Gyroscope	$\boldsymbol{\omega} = \boldsymbol{\omega}_b + \boldsymbol{\omega}_e$	Gyroscope Noise Density	$\boldsymbol{\omega}_e = \mathcal{N}(0, \sigma_\omega^2\mathbf{I}_3)$
Magnetometer	$\mathbf{m}_b = \mathbf{R}_n^b\mathbf{m}_n + \mathbf{m}_e$	Magnetometer Noise Density	$\mathbf{m}_e = \mathcal{N}(0, \sigma_m^2\mathbf{I}_3)$

Table A.1: Sensor Models for Orientation Estimation.

Notes regarding the sensor models:

1. For orientation estimation, the accelerometer model assumes $\mathbf{a}_n$ to be zero [8] (does not take into account external accelerations applied to the sensor). This implies that the accelerometer is assumed to only measure the gravitational acceleration $\mathbf{g}_n$. This can be justified when the motion is of relatively constant speed.

2. $\mathbf{a}_e$, $\boldsymbol{\omega}_e$, and $\mathbf{m}_e$ are Gaussian processes of zero mean and a non-zero covariance. $\mathcal{N}(0, \sigma_x^2\mathbf{I}_3)$ defines a 1×3 normal distribution of zero mean and a 3×3 diagonal covariance matrix of magnitude σ_x^2. In the context of this work, σ_a defines the standard deviation of the accelerometer noise, etc.

3. Perfect axial orthogonality is assumed in each of the 3-axes sensors. Therefore, the error covariance is expressed as a diagonal matrix representing the error or uncertainty in each axis separately.

4. Additionally, a constant uncertainty is assumed to affect all 3-axes of a given sensor, which is not necessarily the case. Different values can be chosen for the different axes as long as they are rotated appropriately in the estimation problem. Fixing them in all axes makes the covariance matrix invariant under rotation.

A.2. Global Reference Models and Process Model

Local Frame	**ENU** - East North Up
Process Model	$\mathbf{f}(\mathbf{q}_t^{nb}, \boldsymbol{\omega}, \boldsymbol{\omega_e}) = \mathbf{q}_t^{nb} \odot \exp_q(\boldsymbol{\omega}_b T) = \mathbf{q}_t^{nb} \odot \exp_q[(\boldsymbol{\omega} - \boldsymbol{\omega}_e)T]$
Earth Gravity Model	$\mathbf{g}_n = (0 \quad 0 \quad -g)$
Earth Magnetic Model	$\mathbf{m}_n = (\cos I \sin D \quad \cos I \cos D \quad -\sin I)$

Table A.2: Earth Field and Process Models

Notes regarding the above models:

1. The state in the continuous-time process model is the quaternion $\mathbf{q}_t^{nb}$ parameterizing a rotation from body frame **b** to local frame **n**. Quaternions are defined in appendix C. T is the sampling period of the IMU signals.

2. The local frame is represented by a right-handed coordinate system. The global field vectors are represented with respect to this reference frame as described in table A.2 above.

A.3. Model Linearization

The process model is a function of the state $\mathbf{q}_t^{nb}$, the gyroscope input $\boldsymbol{\omega}$, and the gyroscope error $\boldsymbol{\omega}_e$. The continuous-time process noise matrix $\mathbf{Q}$ comprises the covariance of $\boldsymbol{\omega}_e$ introduced by the gyroscope and is given as $\sigma_\omega^2 \mathbf{I}_3$.

The linearization of the model is as follows:

$$\mathbf{f}(\mathbf{q}_t^{nb}, \boldsymbol{\omega}, \boldsymbol{\omega_e}) \simeq \frac{\partial \mathbf{f}}{\partial \mathbf{q}_t^{nb}} \mathbf{q}_t^{nb} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\omega}_e} \boldsymbol{\omega}_e + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\omega}} \boldsymbol{\omega} = \mathbf{F}_t \mathbf{q}_t^{nb} + \mathbf{G}_t \boldsymbol{\omega}_e + \mathbf{U}_t \tag{A.1}$$

	Continuous-time	Discrete-time
State Transition Matrix	$\mathbf{F}_t$	$\boldsymbol{\phi}_k = \exp(T\mathbf{F}_t)$
State Error Matrix	$\mathbf{G}_t$	$\mathbf{Q}_d = \mathbf{G}_t\mathbf{Q}\mathbf{G}_t^\top T$

Table A.3: Discrete time Process Matrices

The continuous-time state transition matrix $\mathbf{F}_t$ is evaluated as:

$$\mathbf{F}_t = \frac{\partial \mathbf{f}}{\partial \mathbf{q}_t^{nb}} = \frac{\partial\Big(\mathbf{q}_t^{nb} \odot \exp_q(\boldsymbol{\omega}_b T)\Big)}{\partial \mathbf{q}_t^{nb}} = \frac{\partial\Big(\Big[\exp_q(\boldsymbol{\omega}_b T)\Big]_R \mathbf{q}_t^{nb}\Big)}{\partial \mathbf{q}_t^{nb}} = \Big[\exp \boldsymbol{\omega}_b T\Big]_R \tag{A.2}$$

This follows directly from the process model, which can be expressed as $(\exp_q(\boldsymbol{\omega}_b T))^R \mathbf{q}_t$, converting the quaternion product into a matrix product, from which the state Jaocbian is easily found to be $[\exp_q(\boldsymbol{\omega}_b T)]_R$ in continuous-time, with the definition of the $[\quad]_R$ given in appendix C, describing the right matrix factor in the quaternion product operation. This derivation follows from [8]. The continuous-time state transition matrix $\mathbf{F}_t$ is of a form which allows for a closed-form solution for the matrix exponential, which represents state evolution in discrete-time ϕ_k. It is given by the following evaluations:

$$\boldsymbol{\phi}_k = \exp(T\mathbf{F}_t) = \begin{bmatrix} a & -bp_2 & -bp_3 & -bp_4 \\ bp_2 & a & bp_4 & -bp_3 \\ bp_3 & -bp_4 & a & bp_2 \\ bp_4 & bp_3 & -bp_2 & a \end{bmatrix} \tag{A.3}$$

Where:

$$p_1 = \cos\left(\frac{nT}{2}\right) \quad p_2 = \frac{1}{n}\omega_1 \sin\left(\frac{nT}{2}\right) \quad p_3 = \frac{1}{n}\omega_2 \sin\left(\frac{nT}{2}\right) \quad p_4 = \frac{1}{n}\omega_3 \sin\left(\frac{nT}{2}\right) \tag{A.4}$$

$$a = \frac{1}{2}\exp\Big(p_1 - n\Big)\Big(1 + \exp\left(2n\right)\Big) \quad b = \frac{1}{2n}\exp\Big(p_1 - n\Big)\Big(\exp\left(2n\right) - 1\Big) \tag{A.5}$$

$$n = \sqrt{-\Big(p_2^2 + p_3^2 + p_4^2\Big)} \tag{A.6}$$

This follows directly from equations A.2, C.3 and C.7 in appendices A and C. The continuous-time state error matrix is similarly expressed as [8]:

$$\mathbf{G}_t = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\omega}_e} = \frac{\partial\Big(\mathbf{q}_t^{nb} \odot \exp_q(\boldsymbol{\omega}_b T)\Big)}{\partial \boldsymbol{\omega}_e} = [\mathbf{q}_t^{nb}]_L \frac{\partial\Big(\exp_q(\boldsymbol{\omega}_b T)\Big)}{\partial \boldsymbol{\omega}_e} \tag{A.7}$$

$$\frac{\partial\Big(\exp_q(\boldsymbol{\omega}_b T)\Big)}{\partial \boldsymbol{\omega}_e} = \begin{bmatrix} \frac{s\omega_1 T}{2n} & \frac{s\omega_2 T}{2n} & \frac{s\omega_3 T}{2n} \\ -\frac{T\omega_1^2 c}{2n^2} - \frac{s}{n} + \frac{\omega_1^2 s}{n^3} & -\frac{T\omega_1\omega_2 c}{2n^2} + \frac{\omega_1\omega_2 s}{n^3} & -\frac{T\omega_1\omega_3 c}{2n^2} + \frac{\omega_1\omega_3 s}{n^3} \\ -\frac{T\omega_1\omega_2 c}{2n^2} + \frac{\omega_1\omega_2 s}{n^3} & -\frac{T\omega_2^2 c}{2n^2} - \frac{s}{n} + \frac{\omega_2^2 s}{n^3} & -\frac{T\omega_2\omega_3 c}{2n^2} + \frac{\omega_2\omega_3 s}{n^3} \\ -\frac{T\omega_1\omega_3 c}{2n^2} + \frac{\omega_1\omega_3 s}{n^3} & -\frac{T\omega_2\omega_3 c}{2n^2} + \frac{\omega_2\omega_3 s}{n^3} & -\frac{T\omega_3^2 c}{2n^2} - \frac{s}{n} + \frac{\omega_3^2 s}{n^3} \end{bmatrix} \tag{A.8}$$

Where:

$$n = \| \boldsymbol{\omega}_b \|, \quad s = \sin\left(\frac{nT}{2}\right), \quad c = \cos\left(\frac{nT}{2}\right)$$

The derivation follows from [15]. Again, the $[\ \ ]_L$ is the left matrix factor in the quaternion product operation, also given in appendix C. The discrete-time representation of the state error matrix $\mathbf{G}_t$ is denoted by $\mathbf{Q}_d$ and is approximated by the first-order term $\mathbf{Q}_d \simeq \mathbf{G}_t \mathbf{Q} \mathbf{G}_t^\top T$ [5]. The measurement function of the EKF consists of the sensor models of the accelerometer and the magnetometer and is denoted by $\mathbf{y}$. The actual sensor measurements are denoted by $\mathbf{z}$, the measurement function Jacobian by $\mathbf{H}$, the system uncertainty by $\mathbf{S}$, the initial state covariance by $\mathbf{P}_1$, and the initial state (quaternion) estimate by $\mathbf{q}_1$:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_n^b(\mathbf{a}_n - \mathbf{g}_n) + \mathbf{a}_e \\ \\ \mathbf{R}_n^b\mathbf{m}_n + \mathbf{m}_e \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} \mathbf{z}_a \\ \\ \mathbf{z}_m \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \frac{\partial \mathbf{y}_1}{\partial \mathbf{q}_t^{nb}} \\ \\ \frac{\partial \mathbf{y}_2}{\partial \mathbf{q}_t^{nb}} \end{bmatrix} \tag{A.9}$$

A.4. Filter Computations

Here, $\mathbf{R}$ defines the measurement noise matrix. Since the accelerometer and magnetometer are used as sources of measurements for orientation estimation, matrix $\mathbf{R}$ is represented by the block diagonal matrix of the noise densities $\mathbf{a}_e$ and $\mathbf{m}_e$ defined in table A.1.

$$\mathbf{R} = \begin{bmatrix} \mathbf{a}_e & \mathbf{0} \\ \\ \mathbf{0} & \mathbf{m}_e \end{bmatrix} \tag{A.10}$$

Initialization	$\mathbf{q}_1 = \mathbf{f}(\mathbf{m}_1, \mathbf{a}_1)$ $\mathbf{P}_1 = \mathbf{f}(\boldsymbol{\omega}_e) = \mathbf{G}_t\mathbf{P}_e\mathbf{G}_t^\top$
Prediction	$\hat{\mathbf{q}}_k^{nb} = \mathbf{q}_{k-1}^{nb} \odot \exp_q(\boldsymbol{\omega} T)$ $\hat{\mathbf{P}}_k = \boldsymbol{\phi}_k\mathbf{P}_{k-1}\boldsymbol{\phi}_k^\top + \mathbf{Q}_d$
Update	$\mathbf{r}_k = \mathbf{z}_k - \mathbf{y}_k$ $\mathbf{q}_k^{nb} = \hat{\mathbf{q}}_k^{nb} + \mathbf{K}\mathbf{r}_k$ $\mathbf{K} = \hat{\mathbf{P}}_k\mathbf{H}^\top(\mathbf{H}\hat{\mathbf{P}}_k\mathbf{H}^\top + \mathbf{R})^{-1}$ $\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}(\mathbf{H}\hat{\mathbf{P}}_k\mathbf{H}^\top + \mathbf{R})\mathbf{K}^\top$

Table A.4: EKF Computation Flow

Appendix B

Derivation of the ESKF for Velocity Estimation

B.1. Process Model

The error-state Kalman filter utilizes two sets of equations, nominal state models assuming no sources of error and perfect sensor measurements, and error state models accounting for sensor errors, state-correlated errors, etc.

B.1.1. Continuous-time State Kinematics

Position	$\dot{\mathbf{p}} = \mathbf{v}$	Position Error	$\delta\dot{\mathbf{p}} = \delta\mathbf{v}$
Velocity	$\dot{\mathbf{v}} = \mathbf{R}_b^n \mathbf{a}_b + \mathbf{g}_n$	Velocity Error	$\delta\dot{\mathbf{v}} = -\mathbf{R}_b^n [\mathbf{a}_b]_\times \delta\boldsymbol{\theta} - \mathbf{R}_b^n \mathbf{a}_e$
Quaternion	$\dot{\mathbf{q}}^{nb} = \frac{1}{2}\mathbf{q}^{nb} \odot \boldsymbol{\omega}$	Angular Error	$\delta\dot{\boldsymbol{\theta}} = -[\boldsymbol{\omega}]_\times \delta\boldsymbol{\theta} - \boldsymbol{\omega}_e$

Table B.1: ESKF Process Model

Here, $\mathbf{a}_e$ and $\boldsymbol{\omega}_e$ represent sensor errors and are modelled as white Gaussian processes $\mathbf{a}_e = \mathcal{N}(0, \sigma_a^2 \mathbf{I}_3)$ and $\boldsymbol{\omega}_e = \mathcal{N}(0, \sigma_\omega^2 \mathbf{I}_3)$ as defined in appendix A. The first term in the velocity error correlates the orientation error of the sensor to the linear velocity error and the second term translates the effect of the accelerometer sensor Gaussian error into the velocity term.

B.1.2. Discrete-time state kinematics

Implementing the continuous-time state model in a computer algorithm requires integration of the three constituent differential equations over a sampling period or time interval T resulting in a discretized state model. The Euler method, where the state derivatives are assumed to be constant over the sampling period, is used.

Position	$\hat{\mathbf{p}}_k = \mathbf{p}_{k-1} + \mathbf{v}_{k-1}\Delta t + \frac{1}{2}(\mathbf{R}_b^n \mathbf{a}_b + \mathbf{g}_n)\Delta^2 t$	Position Err.	$\delta\hat{\mathbf{p}}_k = \delta\mathbf{p}_{k-1} + \delta\mathbf{v}_{k-1}\Delta t$
Velocity	$\hat{\mathbf{v}}_k = \mathbf{v}_{k-1} + (\mathbf{R}_b^n \mathbf{a}_b + \mathbf{g}_n)\Delta t$	Velocity Err.	$\delta\hat{\mathbf{v}}_k = \delta\mathbf{v}_{k-1} - \mathbf{R}_b^n[\mathbf{a}_b]_\times \delta\boldsymbol{\theta}_{k-1}\Delta t + \mathbf{v}_i$
Quaternion	$\hat{\mathbf{q}}_k^{nb} = \mathbf{q}_{k-1}^{nb} \odot \exp_q(\boldsymbol{\omega} T)$	Angular Err.	$\delta\hat{\boldsymbol{\theta}}_k = \mathbf{R}_b^{n\top}\delta\boldsymbol{\theta}_{k-1} + \boldsymbol{\theta}_i$

Table B.2: ESKF Discretized Process Model

Where $\mathbf{v}_i$ and $\boldsymbol{\theta}_i$ are the integrated white Gaussian noise of the accelerometer and gyroscope, respectively. $\mathbf{R}_b^n$ denotes the rotation matrix from local frame $\mathbf{n}$ to body frame $\mathbf{b}$.

B.1.3. Error-state estimation

Evaluation steps for the ESKF are as follows:

1. The process model, state, error-state, input, and error vectors are expressed respectively as:

$$\mathbf{f} = \begin{bmatrix} \delta\mathbf{p}_k & \delta\mathbf{v}_k & \delta\boldsymbol{\theta}_k \end{bmatrix}^\top \quad \mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{v} & \mathbf{q} \end{bmatrix}^\top \quad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta} \end{bmatrix}^\top \quad \mathbf{u} = \begin{bmatrix} \mathbf{a}_b & \boldsymbol{\omega} \end{bmatrix}^\top \quad \mathbf{e} = \begin{bmatrix} \mathbf{v}_i & \boldsymbol{\theta}_i \end{bmatrix}^\top \tag{B.1}$$

2. The state transition matrix $\mathbf{F}_x$ is the Jacobian of the process model $\mathbf{f}$ with respect to the error states while the state error matrix $\mathbf{F}_i$ is the Jacobian of the process model

with respect to the error vector $\mathbf{e}$:

$$\mathbf{F}_x = \frac{\partial \mathbf{f}(\mathbf{x}, \delta\mathbf{x}, \mathbf{u}, \mathbf{e})}{\partial \delta\mathbf{x}} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{R}_b^n[\mathbf{a}_b]_\times \Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_b^{n\top} \end{bmatrix} \quad \mathbf{F}_i = \frac{\partial \mathbf{f}(\mathbf{x}, \delta\mathbf{x}, \mathbf{u}, \mathbf{e})}{\partial \mathbf{e}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & 0 \\ \mathbf{I} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{I} & 0 \end{bmatrix} \tag{B.2}$$

3. The covariance of the Gaussian process error is $\mathbf{Q}$:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{V}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{\Theta}_i \end{bmatrix} \quad \mathbf{V}_i = \mathcal{N}(0, \sigma_a^2 \Delta t^2 \mathbf{I}_3) \quad \mathbf{\Theta}_i = \mathcal{N}(0, \sigma_\omega^2 \Delta t^2 \mathbf{I}_3) \tag{B.3}$$

Assuming uncorrelated sensor axes errors of equal magnitude in both the accelerometer and gyroscope of the IMU. However, the error variance terms (error magnitude) can be different for the different axes of the sensors.

4. At the start of the filter computations, the filter states and error states are initialized according to the following values:

$$\mathbf{p} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \quad \mathbf{v} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \tag{B.4}$$

Additionally, quaternion $\mathbf{q}$ is initialized using the accelerometer tilt equations [6]. This concludes initializations for the nominal state vector. The error states are all initialized to zero:

$$\delta\mathbf{p} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \quad \delta\mathbf{v} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \quad \delta\boldsymbol{\theta} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \tag{B.5}$$

5. $\mathbf{P}$ denotes the error state covariance and is propagated through time using the standard equation of the linear Kalman filter:

$$\hat{\mathbf{P}}_k = \mathbf{F}_x \mathbf{P}_{k-1} \mathbf{F}_x^\top + \mathbf{F}_i \mathbf{Q} \mathbf{F}_i^\top \tag{B.6}$$

6. The measurement function is the linear velocity given as:

$$\mathbf{y}_k = \mathbf{v}_k + (\mathbf{R}_b^n \mathbf{a}_b + \mathbf{g}_n)\Delta t + \mathbf{Y}_i \quad \mathbf{Y}_i = \mathcal{N}(0, \sigma_y^2 \mathbf{I}_3) \tag{B.7}$$

where $\mathbf{Y}_i$ represents the uncertainty in the velocity measurement. The measurement Jacobian with respect to the error states is evaluated using the chain rule:

$$\mathbf{H}_k = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \delta \mathbf{x}} \tag{B.8}$$

The first term is represented by the following matrix:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \Delta t \frac{\partial (\mathbf{R}_b^n \mathbf{a}_b)}{\partial \mathbf{q}^{nb}} \end{bmatrix} \quad \frac{\partial \mathbf{x}}{\partial \delta \mathbf{x}} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\delta \boldsymbol{\theta}} \end{bmatrix} \quad \mathbf{Q}_{\delta \boldsymbol{\theta}} = \frac{1}{2} [\mathbf{q}^{nb}]_L \begin{bmatrix} 0 & \boldsymbol{I}_3 \end{bmatrix} \tag{B.9}$$

The first term is evaluated from $\mathbf{R}_b^n(\mathbf{q})$ defined in appendix C. The second term follows from the work of [22]. The measurement Jacobian matrix with respect to the error states can now be evaluated.

7. The Kalman gain $\mathbf{K}_k$ is evaluated using the standard equation of the linear Kalman filter:

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{H}_k^\top (\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^\top + \mathbf{Y}_i)^{-1} \tag{B.10}$$

8. The pseudo measurement is the zero linear velocity of the IMU denoted by $\mathbf{z}$:

$$\mathbf{z}_k = \mathbf{0}_{1\times 3} \tag{B.11}$$

9. The corrected error states and error states covariance estimates are finally updated using the Kalman gain according to the standard gain equation:

$$\delta \mathbf{x}_k = \mathbf{K}_k (\mathbf{z}_k - \mathbf{y}_k) \quad \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k \tag{B.12}$$

10. The error states are then injected into the nominal states for correction:

$$\mathbf{p}_k = \hat{\mathbf{p}}_k + \delta\mathbf{p}_k \quad \mathbf{v}_k = \hat{\mathbf{v}}_k + \delta\mathbf{v}_k \quad \mathbf{q}_k^{nb} = \hat{\mathbf{q}}_k^{nb} \odot \exp_q(\delta\boldsymbol{\theta}_k) \tag{B.13}$$

11. Finally, the error states are reset to zero according to the formulation of [22]:

$$\delta\mathbf{x}_k = \mathbf{0}_{9\times 1} \tag{B.14}$$

Appendix C

Quaternion Math

C.1. Quaternion Algebra

1. Rotation can be parameterized by unit quaternions, a 4-component vector in the hypercomplex plane. A quaternion $\mathbf{q}$ is defined as:

$$\mathbf{q} = q_0 + \mathbf{q}_v = q_0 + iq_1 + jq_2 + kq_3 \tag{C.1}$$

where q_0 is a real scalar and $\mathbf{q}_v$ is an imaginary vector. $\mathbf{q}$ is a fourth dimensional vector used to parameterise rotation [8].

2. The product of two quaternions is bi-linear and can be expressed as [22]:

$$\mathbf{q} \odot \mathbf{p} = [\mathbf{q}]_L \mathbf{p} = [\mathbf{p}]_R \mathbf{q} \tag{C.2}$$

$$[\mathbf{q}]_L = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{C.3}$$

$$[\mathbf{p}]_R = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & p_3 & -p_2 \\ p_2 & -p_3 & p_0 & p_1 \\ p_3 & p_2 & -p_1 & p_0 \end{bmatrix} \tag{C.4}$$

3. Rotation matrix $\mathbf{R}$ to rotate an arbitrary vector $\mathbf{v}$ can be expressed in terms of the quaternion $\mathbf{q}$ as [9]:

$$\mathbf{R} = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix} \tag{C.5}$$

4. The skew-symmetric matrix of a vector $\mathbf{v}$ is denoted by $[\mathbf{v}]_x$ and is expressed as:

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{C.6}$$

5. The quaternion exponential map, an extension of the Euler formula, is used to represent a *rotation vector* of axis $\mathbf{u}$ and angle $\boldsymbol{\phi}$ as a quaternion:

$$\mathbf{q} = \exp_{\mathbf{q}}\left(\frac{\mathbf{v}}{2}\right) = \begin{bmatrix} \cos\left(\frac{\boldsymbol{\phi}}{2}\right) \\ \mathbf{u}\sin\frac{\boldsymbol{\phi}}{2} \end{bmatrix} \tag{C.7}$$

where the magnitude of $\mathbf{v}$ represents the angle of rotation and the unit vector of $\mathbf{v}$ represents the axis of rotation.

BIBLIOGRAPHY

[1] A. Cutti, A. G., and Rocchi, L. Ambulatory measurement of shoulder and elbow kinematics through inertial and magnetic sensors. *Medical Biological Engineering and Computing 46* (2008), 169–178.

[2] Chen, Shanshan, L. J. L. B. Y. G. Z. Toward pervasive gait analysis with wearable sensors: A systematic review. *IEEE Journal of Biomedical and Health Informatics 20* (2016), 1521–1537.

[3] D. Roetenberg, H. L., and Slycke, P. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Tech. rep., 2013.

[4] Elbaz, A. Lower extremity kinematic profile of gait of patients after ankle fracture: A case-control study. *Foot and Ankle Surgery 55* (2016), 918–921.

[5] FARREL, J. A. *AIDED NAVIGATION*. Mc.Graw Hill, 2008.

[6] Fischer, C. Tutorial: Implementing a pedestrian tracker using inertial sensors. *IEEE Pervasive Computing 12* (2013), 17–27.

[7] Hillcrest. Bno080 data sheet table of contents. Tech. rep., 2017.

[8] Kok, Hol, S. Using inertial sensors for position and orientation estimation.

[9] Kuipers, J. B. *QUATERNIONS and ROTATION SEQUENCES*. Princeton Univerity Press, 1999.

[10] Li, Xiang, L. Z. A new calibration method for tri-axial field sensors in strap-down navigation systems. *Measurement Science and Technology 23* (2012).

[11] M. Benedetti, F. Catani, A. L. E. P., and Giannini, S. data management in gait analysis for clinical applications). *Clin. Biomech. 13*, 3 (1998), 204–215.

[12] Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, R. Estimation of imu and marg orientation using a gradient descent algorithm. 1–7.

[13] Manon Kok, T. B. S. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal 16* (2016), 56795689.

[14] Miezal, Markus, T. B. B. G. On inertial body tracking in the presence of model calibration errors. *Sensors (Switzerland) 16* (2016), 1–34.

[15] Myoung-Jun Kim, M.-S. K. A compact differential formula for the first derivative of a unit quaternion curve. *J. of Visualization and Computer Animation A* (1996).

[16] NCEI. Magnetic field calculators.

[17] Palermo, Eduardo, R. S.-M. F. P. F. C. P. Experimental evaluation of accuracy and repeatability of a novel body-to-sensor calibration procedure for inertial sensor-based gait analysis. *Measurement: Journal of the International Measurement Confederation 52* (2014), 145–155.

[18] S. Tadano, R. Takeda, K. S., and Fujisawa, T. Gait characterization for osteoarthritis patients using wearable gait sensors (h-gait systems). *Foot and Ankle Surgery 49* (2016), 684–690.

[19] Sabatini, A. M. Kalman-filter-based orientation determination using inertial/magnetic sensors: Observability analysis and performance evaluation. *Sensors 11* (2011), 9182–9206.

[20] Sanders RD, G. P. Gait and its assessment in psychiatry. *Psychiatry (Edgmont) 7* (2010).

[21] Simon, D. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory Applications 4* (2010), 1303–1318.

[22] Sola, J. Quaternion kinematics for the error-state kalman filter. *IEEE Transactions on Aerospace and Electronic Systems* (2017).

[23] Stanley, M. Freescale semiconductor mems industry group accelerated innovation community.

[24] Strohrmann, C., H. H. T. G. H. S. M. R. . out of the lab and into the woods: kinematic analysis in running using wearable sensors. *13rd International Conference on Ubiquitous Computing* (2011).

[25] T. Zimmerman, T. B., and Bleser, G. Imu-to-segment assignment and orientation alignment for the lower body using deep learning. *MDPI Sensors 18* (2018), 1–35.

[26] Trumble, M., Gilbert, A., Malleson, C., Hilton, A., and Collomosse, J. Total capture: 3d human pose estimation fusing video and inertial sensors. In *2017 British Machine Vision Conference (BMVC)* (2017).

[27] Valenti, Dryanovski, X. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. *Sensors (Switzerland) 15* (2015), 19302–19330.

[28] van der Kruk, E. Accuracy of human motion capture systems for sport applications; state-of-the-art review. *BIOMECHANICS AND MOTOR CONTROL* (2018).

[29] Vasconcelos, J. F., E. G. S. C. O. P. C. B. Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic Systems 47* (2011), 1293–1306.

[30] Vitali, A. Ellipsoid or sphere fitting for sensor calibration. 2–7.

[31] W. De Vries, H. Veeger, A. C., and Baten, C. Functionally interpretable local coordinate systems for the upper ex- tremity using inertial magnetic measurement systems. *Journal of Biomechanics 43* (2010), 1983–1988.

[32] Wan, Merwe, R. The unscented kalman filter for nonlinear estimation.

[33] Xiao, X., and Zarar, S. Machine learning for placement-insensitive inertial motion capture. Tech. rep., 2018.

[34] Yizhou, W. Three dimensional attitude estimation via the triad algorithm and a time-varying complementary filter. *JSME 2012 11th Motion and Vibration Conference 6* (2012), 1–9.