DEPTH CAMERA AIDED DEAD-RECKONING LOCALIZATION OF AUTONOMOUS MOBILE ROBOTS IN UNSTRUCTURED GNSS-DENIED ENVIRONMENTS

by
David Lawrence Olson

A thesis submitted to Johns Hopkins University in conformity with the requirements for

the degree of Master of Science

Baltimore, Maryland
December 2021

Abstract

In global navigation satellite system (GNSS) denied settings, such as indoor environments, autonomous mobile robots are often limited to dead-reckoning navigation techniques to determine their position, velocity, and attitude (PVA). Localization is typically accomplished by employing an inertial measurement unit (IMU), which, while precise in nature, accumulates errors rapidly, and severely degrades the localization solution. Standard sensor fusion methods, such as Kalman filtering, aim to fuse short-term precise IMU measurements with long-term accurate aiding sensors to establish a precise and accurate solution. In indoor environments, where GNSS and no other *a priori* information is known about the environment, effective sensor fusion is difficult to achieve, as accurate aiding sensor choices are sparse. Fortunately, an opportunity arises by employing a depth camera in the indoor environment. A depth camera can capture point clouds of the surrounding floors and walls. Extracting attitude from these surfaces can serve as an accurate aiding source, which directly mitigates errors that arise due to gyroscope imperfections. This configuration for sensor fusion leads to a dramatic reduction of PVA error compared to other traditional aiding sensor configurations. This paper provides the theoretical basis for the new aiding sensor method, initial expectations of performance benefit via simulation, and hardware implementation of the new algorithm, thereby verifying its veracity. Hardware implementation is performed on the Quanser Qbot 2™ mobile robot with a Vector-Nav VN-200™ IMU and Kinect™ camera from Microsoft.

Primary Advisor: Dr. Stephen B.H. Bruder
Second Advisor: Dr. Adam S. Watkins
Tertiary Advisor: Dr. Cleon E. Davis

Acknowledgments

I am very grateful for the lessons, advice, and mentorship of Dr. Stephen Bruder. He is the professor who started me down the path of navigation sciences. Dr. Bruder's approach to problems and attitude towards life has dramatically impacted my worldview. He has served as my biggest role model since the start of my academic and professional career.

I would also like to thank Dr. Adam Watkins and Dr. Cleon Davis for their support in pursuing this research. Dr. Watkins' course in Kalman filtering gave me the confidence to pursue this topic in detail. His advice, mentorship, and encouragement have been much appreciated.

Dedication

This thesis is dedicated to Mrs. Lowe, my pre-calculus teacher from my junior year of high school. When signing up for classes for my senior year of high school, I told Mrs. Lowe that I was planning on taking two "off periods" instead of any mathematics or science courses. She immediately ripped my sign-up sheet from my hands and wrote, *in pen*, AP Calculus BC and AP Physics. At the time I had no intention of pursuing engineering or mathematics, but I was eventually convinced that I should at least try the courses and see what I think. What I thought was a massive inconvenience at that moment is now what I consider one of the most extraordinary favors a teacher has done for my future. Thank you for the nudge, Mrs. Lowe.

Table of Contents

List of Tables

List of Figures

Chapter 1 - Introduction

In the development of reliable navigation systems [1], inertial sensors, specifically, accelerometers and gyroscopes, remain the primary information source as they are virtually impervious to external influences [2], [3]. Unfortunately, all inertial-only navigation solutions suffer from inertial drift, which is an inherent consequence of integrating imperfect acceleration and angular velocity measurements to determine position, velocity, and attitude (PVA). A durable approach to ameliorating this dilemma is to complement the short-term precise inertial-only PVA solution with long-term accurate aiding sensors such as a GNSS receiver, magnetometer, barometric altimeter, LIDAR, odometry, camera, *etc.* [4]. Unfortunately, some of these aiding sensors are not suited to a mobile robotic platform operating indoors [5]. For example, GNSS signals are unreliable indoors and magnetic fields generated by the robot's drive motors distort the magnetometer readings regardless of the environment.

Research in GNSS-denied navigation is becoming a focal point in many application spaces [3]. While GNSS-supported navigation is well suited in many applications, such as crop management applications demonstrated by the Australian Centre for Field Robotics [6], not all mobile robot applications will benefit from GNSS assistance. For example, robotic platforms traveling in indoor environments commonly suffer from multipath errors and signal outages [7]. The research in this thesis assumes the worst-case scenario, in that GNSS signals are consistently unavailable to support indoor navigation efforts. The common alternative to GNSS-focused navigation is to employ the inertial measurement unit (IMU) as the core of the overall inertial navigation system (INS).

This research analyzes the challenges of navigating via an IMU by identifying which IMU error sources have the most severe impact on the final PVA solution. Once identified, an appropriate aiding sensor option will be explored and ultimately included in the INS. Finally, simulation and hardware implementation will determine and verify the overall benefit of this unique aiding sensor to the final PVA solution.

IMU error analysis will be specific to the VectorNav VN-200™. Figure 1 has an image of the VectorNav VN-200™; this IMU is representative of the typical performance specifications of consumer-grade IMUs on the market.

Figure 1 - VectorNav VN-200™

This IMU will be onboard the Quanser Qbot 2™ shown in Figure 2. The Quanser Qbot 2™ serves as the mobile robotic platform that requires navigation. Onboard the Quanser Qbot 2™ is a Microsoft Kinect™ camera, which has both RGB and depth camera capabilities.

Figure 2 - Quanser Qbot 2™ with a Microsoft Kinect™ camera

Preliminary Background Topics

The following section covers the basics of the navigation sciences and the mathematical principles and notations used in this publication. Those unfamiliar with navigation principles are encouraged to read the next section, while those with adequate background might move on directly to Chapter 2 – Inertial Navigation Background and Challenges.

Common Coordinate Frames

Coordinate frames are applied to various settings, from describing the orientation of the Earth to mobile robots interacting with a local environment. Thus, understanding the background and role of each coordinate frame is crucial to the material ahead.

The Earth Centered Inertial (ECI) frame, also referred to as the i-frame, captures the inertial background of the surrounding universe. The origin of the ECI frame is set at the center of mass of the Earth. The $\vec{x}_i$ axis is established by pointing a vector from the Earth to the Sun during the vernal equinox, and the $\vec{z}_i$ axis is aligned with the spin-axis of the Earth. The $\vec{y}_i$ axis is orthogonal to both axes according to the right-hand rule. This is depicted in Figure 3.

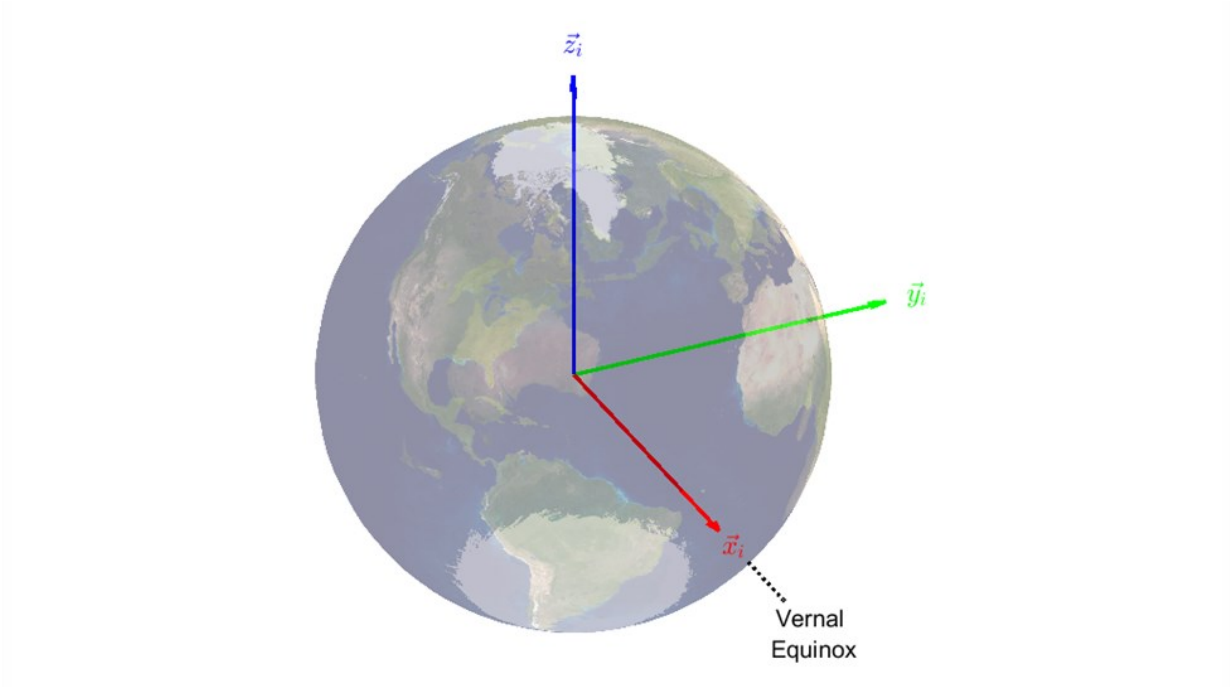

Figure 3 - Earth Centered Inertial (ECI) Frame Example

The Earth Centered Earth Fixed (ECEF frame), referred to as the e-frame, is defined similarly to the ECI frame; however, this coordinate frame is fixed to the Earth (Figure 4). While both the ECI and ECEF frames share the same origin, the ECEF frame rotates with the Earth, meaning the ECI and ECEF are only in alignment once approximately every twenty-four hours. The ECEF frame is commonly employed for global positioning measurements using geodetic latitude, longitude, and height coordinates.

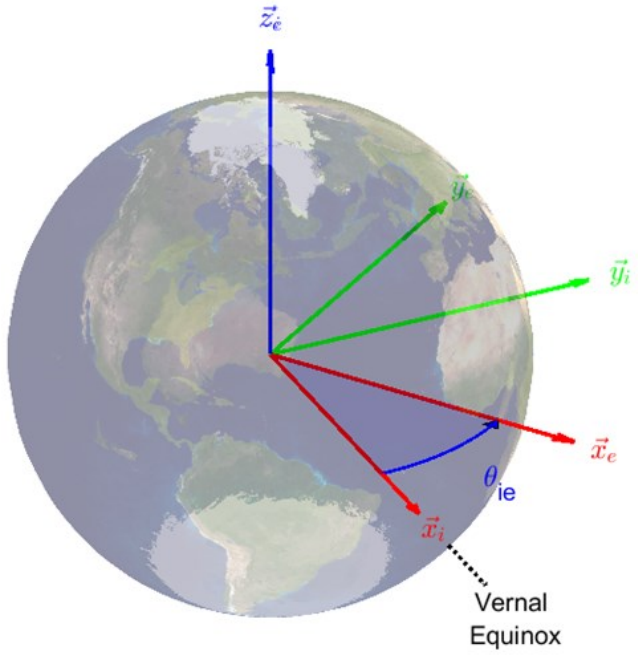

Figure 4 - Earth Centered Earth Fixed (ECEF) Frame Example

The tangential frame (or *t*-frame) is used for vehicles operating within local confines to define the local coordinate frame axes. These axes are commonly defined as locally-level in the North, East, and Down (NED) configuration. However, in some cases, the $\vec{x}_t$ and $\vec{y}_t$ axes are rotated to align with local environmental features, such as walls in an indoor building. For example, a tangential frame is placed locally in Minot, North Dakota in the NED configuration below.

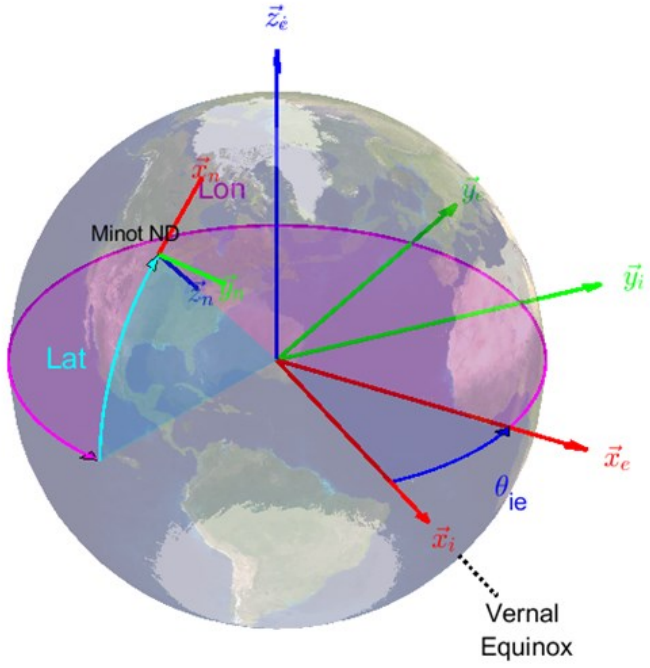

Figure 5 - Tangential Frame Example

Finally, the body frame (or *b*-frame) is used to describe the axes of the vehicle (Figure 6). In the material ahead, the desire is to determine the body frame's PVA with respect to the tangential frame. The *b*-frame's x, y, and z axes are typically aligned with the forward, right, and down directions of the vehicle, respectively.

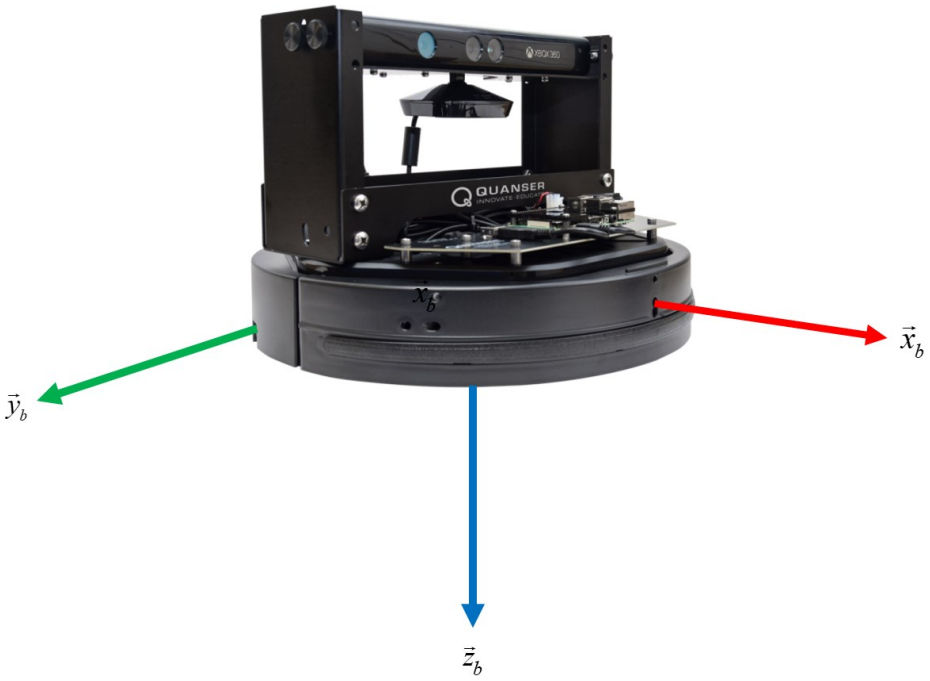

Figure 6 - Body Frame Example

Vector Frame Notation

Consider the position vector $\vec{r}$, of an object, described via x, y, z Cartesian coordinates.

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1.1}$$

This description of the position, however, is inadequate. This vector could describe position with respect to any reference point and in any direction. Instead, the three following items should be defined:

a.) The point whose position is of interest

b.) The coordinate frame from which the position is being described

c.) The resolving frame, the frame in which the measurement is coordinatized in

For example, consider the position of point {a} with respect to frame {b}, coordinatized in frame {c}. Figure 7 helps to visualize the interaction of the three coordinate frames.

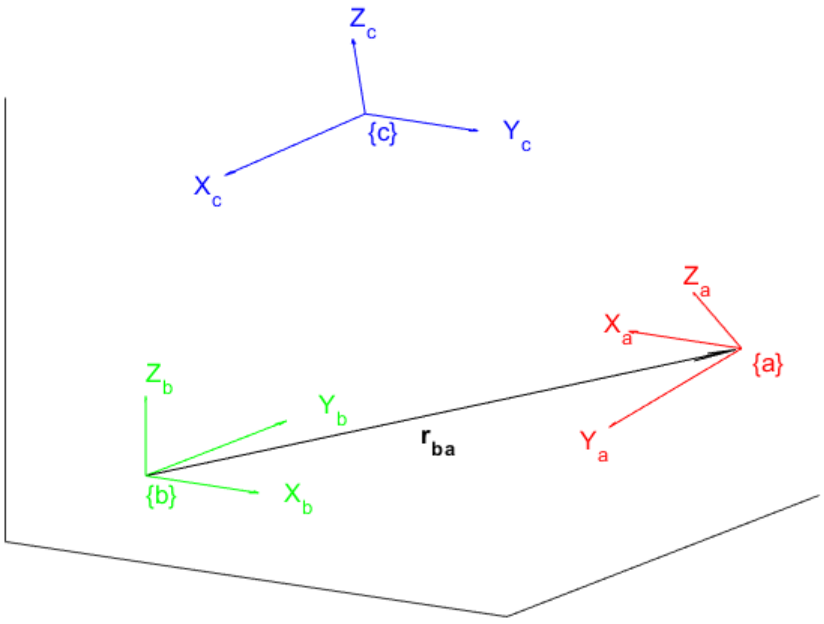

Figure 7 - Vector Frame Notation Example

This position vector would be written as:

$$\vec{r}_{ba}^{\,c} = \begin{bmatrix} x_{ba}^{c} \\ y_{ba}^{c} \\ z_{ba}^{c} \end{bmatrix} \tag{1.2}$$

Attitude Representations

The most common method for attitude representation is Euler angles, which describe a vehicle's roll ϕ, pitch θ, and yaw ψ. These angles describe rotation about each Cartesian axis, in a specified order, with positive rotation defined by the right-hand rule.

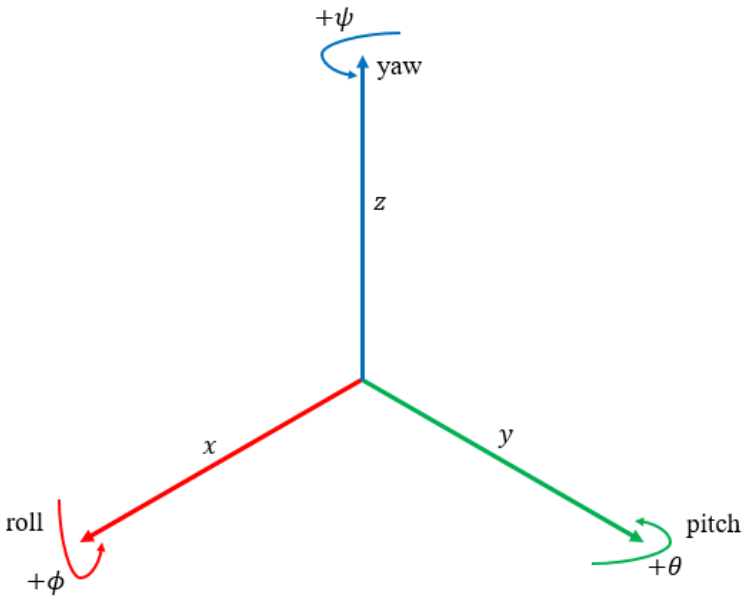

Figure 8 - Euler Angle Example

Another common representation is the Directional Cosine Matrix (DCM). This representation is commonly used to describe the attitude of one coordinate frame with respect to another. Consider two arbitrary coordinates frames {a} and {b}. The projection of the $\vec{x}_a$ axis onto the {b} frame is captured by each dot-product, as shown in Figure 9.

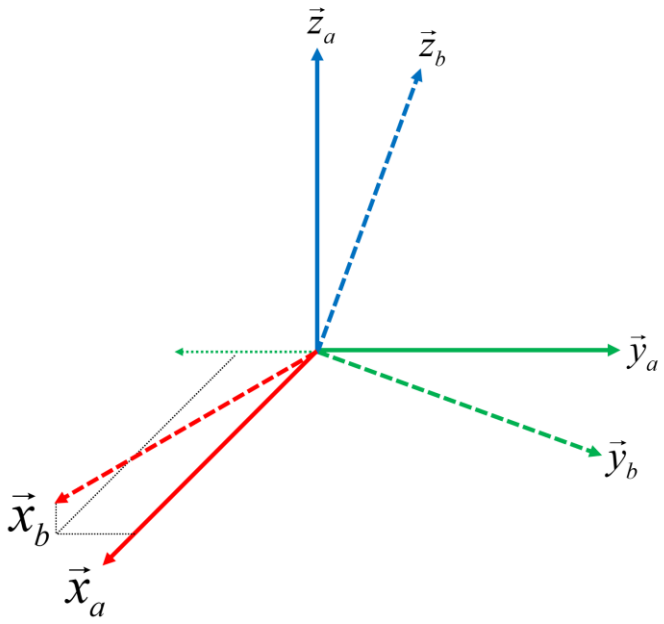

Figure 9 - DCM Example

Dot-product relationships express the projection of the frame {b} axes onto the frame {a} axis, which can alternatively be expressed as cosines of angles between each axis.

$$C_b^a = \begin{bmatrix} \vec{x}_b^a & \vec{y}_b^a & \vec{z}_b^a \end{bmatrix} = \begin{bmatrix} \vec{x}_b \cdot \vec{x}_a & \vec{y}_b \cdot \vec{x}_a & \vec{z}_b \cdot \vec{x}_a \\ \vec{x}_b \cdot \vec{y}_a & \vec{y}_b \cdot \vec{y}_a & \vec{z}_b \cdot \vec{y}_a \\ \vec{x}_b \cdot \vec{z}_a & \vec{y}_b \cdot \vec{z}_a & \vec{z}_b \cdot \vec{z}_a \end{bmatrix}$$

$$C_b^a = \begin{bmatrix} \cos(\theta_{\vec{x}_b \cdot \vec{x}_a}) & \cos(\theta_{\vec{y}_b \cdot \vec{x}_a}) & \cos(\theta_{\vec{z}_b \cdot \vec{x}_a}) \\ \cos(\theta_{\vec{x}_b \cdot \vec{y}_a}) & \cos(\theta_{\vec{y}_b \cdot \vec{y}_a}) & \cos(\theta_{\vec{z}_b \cdot \vec{y}_a}) \\ \cos(\theta_{\vec{x}_b \cdot \vec{z}_a}) & \cos(\theta_{\vec{y}_b \cdot \vec{z}_a}) & \cos(\theta_{\vec{z}_b \cdot \vec{z}_a}) \end{bmatrix} \tag{1.3}$$

DCMs are commonly used to rotate vectors resolved in one coordinate frame into another frame. Computations must involve vectors all resolved in the same coordinate frame. For example, referring to Figure 7, consider the addition of vectors $\vec{r}_{bc}^{\,b}$ and $\vec{r}_{ca}^{\,c}$ to obtain $\vec{r}_{ba}^{\,b}$.

This addition would require the use of a DCM to coordinatize the vector $\vec{r}_{ca}^{\,c}$ into the {b} frame prior to addition.

$$\begin{aligned} \vec{r}_{ba}^{\,b} &\neq \vec{r}_{bc}^{\,b} + \vec{r}_{ca}^{\,c} \\ \vec{r}_{ba}^{\,b} &= \vec{r}_{bc}^{\,b} + C_c^b\, \vec{r}_{ca}^{\,c} = \vec{r}_{bc}^{\,b} + \vec{r}_{ca}^{\,b} \end{aligned} \tag{1.4}$$

The DCM has a valuable mathematical property in that it is an orthonormal matrix. Due to this property, the inverse of a DCM and the transpose of a DCM are equivalent.

$$\begin{aligned} C_b^a &= \left[C_a^b\right]^{-1} = C_a^{b'} \\ C_a^b C_b^a &= C_a^b \left[C_a^b\right]^{-1} = C_a^b C_a^{b'} = I \end{aligned} \tag{1.5}$$

In the case of multiple rotations, recall that matrix multiplications are, in general, non-communitive. This property reflects that the sequence of rotations is significant and changing a sequence of rotations will result in different final attitudes.

Another attitude representation is the angle-axis representation. Given an angle-axis rotation vector $\vec{k}$, the vector can be split into its magnitude (amount of rotation) and a unit-vector defining an axis of rotation. The magnitude of the rotation vector encodes the angle rotated about the unit vector, while the unit vector defines the spin axis.

$$\begin{aligned} \vec{k} &= \bar{k}\,\theta = \theta \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \\ \theta &= \left\|\vec{k}\right\|,\ \bar{k} = \frac{\vec{k}}{\left\|\vec{k}\right\|} \end{aligned} \tag{1.6}$$

To describe a cross-product operation involving an angle-axis vector, it can be transformed into a skew-symmetric form, as demonstrated below.

$$\mathcal{K} = \left[\vec{k}\times\right] = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \left[\vec{k}\times\right]\vec{r} = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{1.7}$$

Angular Rates

Angular rates also require careful definition in a matter similar to attitude representations. For example, it is a common mistake to refer to Euler rates as angular velocity.

$$\frac{d}{dt}\begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} \dot{\phi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} \neq \vec{\omega} \tag{1.8}$$

Angular velocity is instead defined in a manner similar to the angle-axis format. Unfortunately, taking the derivative of an angle-axis format vector is non-differentiable with respect to time. When both the rotation rate and axis of rotation are changing simultaneously, it is not easy to define mathematically which change to apply first, as the order would lead to different results. Instead, it is assumed that the axis of rotation remains "relatively" constant for small time step sizes, which leads to the following definition for, so called, body-reference angular velocity.

$$\frac{d}{dt}\vec{k}(t)=\frac{d}{dt}\left[\bar{k}(t)\theta(t)\right]\approx\frac{d}{dt}\left[\bar{k}\,\theta(t)\right]=\bar{k}\,\dot{\theta}(t),\quad t\le t\le t+\Delta t$$
$$\bar{k}\,\dot{\theta}(t)\triangleq\vec{\omega}(t),\quad t\le t\le t+\Delta t \tag{1.9}$$

Angular velocity vectors are subject to the same properties as position and velocity vectors in terms of their addition, subtraction, and rotation. Additionally, the angular velocity vector can also be transformed into a skew-symmetric matrix.

$$\Omega^{c}_{ba}(t)=[\vec{\omega}^{c}_{ba}(t)\times]=\begin{bmatrix}0 & -\vec{\omega}^{c}_{ba,3}(t) & \vec{\omega}^{c}_{ba,2}(t)\\ \vec{\omega}^{c}_{ba,3}(t) & 0 & -\vec{\omega}^{c}_{ba,1}(t)\\ -\vec{\omega}^{c}_{ba,2}(t) & \vec{\omega}^{c}_{ba,1}(t) & 0\end{bmatrix} \tag{1.10}$$

where, $\vec{\omega}^{c}_{ba}\triangleq\begin{bmatrix}\vec{\omega}^{c}_{ba,1} & \vec{\omega}^{c}_{ba,2} & \vec{\omega}^{c}_{ba,3}\end{bmatrix}^{T}$

Skew-symmetric matrices exhibit the addition and rotation properties demonstrated below.

$$\begin{aligned}\Omega^{c}_{ba}(t)&=-\Omega^{c}_{ab}(t)\\ \Omega^{c}_{ba}(t)&=\Omega^{c}_{bd}(t)+\Omega^{c}_{da}(t)\\ \Omega^{c}_{ba}(t)&=C^{c}_{d}\Omega^{d}_{ba}(t)C^{d}_{c}\end{aligned} \tag{1.11}$$

The derivative of a DCM attitude quantity is also achievable and essentially captures the angular velocity of the rotating reference frame. The derivative of a DCM is derived below.

$$
\begin{aligned}
C_b^a(t+dt) &= C_{a(t)}^{a(t+dt)} C_b^a(t) \\
&= \left(I_3 - \left[\vec{k}_{a(t)}^{a(t+dt)} \times \right] \right) C_b^a(t) \\
&= \left(I_3 - dt \left[\vec{\omega}_{ba}^a \times \right] \right) C_b^a(t) \\
&= \left(I_3 - dt\, \Omega_{ba}^a \right) C_b^a(t)
\end{aligned}
$$

$$
\begin{aligned}
\dot{C}_b^a(t) &= \lim_{t \to 0} \left(\frac{C_b^a(t+dt) - C_b^a(t)}{dt} \right) \\
&= \lim_{t \to 0} \left(\frac{\left(I_3 - dt\, \Omega_{ba}^a \right) C_b^a(t) - C_b^a(t)}{dt} \right) \\
&= -\Omega_{ba}^a C_b^a(t)
\end{aligned}
\tag{1.12}
$$

Chapter 2 – Inertial Navigation Background and Challenges

IMU Basic Operation

In most navigation applications, the IMU is mounted to the body of a vehicle and provides measurements regarding its movement. An IMU consists mainly of accelerometers and gyroscopes, which measure specific force and rotational rates, respectively.

Specific force refers to any force applied to the body except for the force of gravity. Because the IMU is always acted upon by gravity, no external reference is available to measure its influence. The specific force quantity measured by the accelerometer is described mathematically in (2.1). This vector is observed in the body frame (*b*-frame) with respect to the ECI frame (subscript), coordinatized in the body frame (superscript).

$$\vec{f}_{ib}^{\,b} = \begin{bmatrix} \vec{f}_{ib,x}^{\,b} \\ \vec{f}_{ib,y}^{\,b} \\ \vec{f}_{ib,z}^{\,b} \end{bmatrix} \tag{2.1}$$

The gyroscope measures angular velocity in an angle-axis format. The angle-axis format is different from Euler rates in the sense that the angle-axis vector captures both the magnitude and axis of rotation. Thus, the angle-axis format is demonstrated in (2.2) with the magnitude of the rotation $|\omega|$ and the unit vector capturing the axis of rotation $\overline{k}_{ib}^{\,b}$.

$$\vec{\omega}_{ib}^{b} = \begin{bmatrix} \vec{\omega}_{ib,x}^{b} \\ \vec{\omega}_{ib,y}^{b} \\ \vec{\omega}_{ib,z}^{b} \end{bmatrix}$$
$$\vec{\omega}_{ib}^{b} = |\omega| \bar{k}_{ib}^{b} = |\omega| \begin{bmatrix} \bar{k}_{ib,x}^{b} \\ \bar{k}_{ib,y}^{b} \\ \bar{k}_{ib,z}^{b} \end{bmatrix} \quad (2.2)$$

Accelerometer and gyroscope measurements are integrated over time to compute position, velocity, and attitude (PVA), thus providing a complete navigation solution.

Computing PVA in the Tangential Frame

There are a variety of coordinate frames available for use in the navigation sciences. One coordinate frame of particular interest is the tangential frame (*t*-frame), commonly used for mobile robots traveling in local environments. Tangential frames are generally defined as locally-level for the space of interest, axes aligned either to the starting attitude of the robot body or to features in the environment, and with approximately constant magnitude and direction of the local gravity vector ($\vec{g}^{t}$) in the *t*-frame. One example of this would be inside a building that contains flat floors, walls and floors to align axes, and a local gravity vector that remains approximately constant regardless of location in the building.

However, the IMU records measurements with respect to the *i*-frame instead of the *t*-frame. Therefore, an intermediate step between the *i*-frame and the *t*-frame is required, known as the Earth Centered Earth Fixed frame (ECEF frame or *e*-frame). The tangential mechanization

is derived below to compute PVA in the *t*-frame (*i.e.*, $\vec{r}_{tb}^{\,t}$, $\vec{v}_{tb}^{\,t}$, C_b^t) from inertial measurements in the *i*-frame.

First, consider the position vector $\vec{r}_{ib}^{\,i}(t)$. This vector can be broken into separate vectors describing the position of one coordinate frame to another. Then, these position vectors are rotated by directional cosine matrices (DCMs) to align them to other coordinate frames.

$$\begin{aligned}\vec{r}_{ib}^{\,i}(t) &= \vec{r}_{ie}^{\,i}(t) + \vec{r}_{et}^{\,i}(t) + \vec{r}_{tb}^{\,i}(t) \\ &= \vec{0} + C_e^i(t)\vec{r}_{et}^{\,e} + C_e^i(t)\vec{r}_{tb}^{\,e}(t) \\ &= C_e^i(t)\left(\vec{r}_{et}^{\,e} + \vec{r}_{tb}^{\,e}(t)\right)\end{aligned} \tag{2.3}$$

As the ECI frame and ECEF frame share the exact origin, $\vec{r}_{ie}^{\,i}(t)$ is equal to zero. Additionally, the vector $\vec{r}_{et}^{\,e}$ is constant with respect to time, given that the vector points from the center of mass of the Earth to the origin of the tangential frame. Additionally, the Earth's angular velocity $\vec{\omega}_{ie}^{\,i}$ is also considered to be constant. Moving forward with these assumptions, taking the derivative of (2.3) leads to the following:

$$\begin{aligned}\dot{\vec{r}}_{ib}^{\,i}(t) &= \frac{d}{dt}\left[C_e^i(t)\left(\vec{r}_{et}^{\,e} + \vec{r}_{tb}^{\,e}(t)\right)\right] \\ &= \dot{C}_e^i(t)\vec{r}_{et}^{\,e} + \dot{C}_e^i(t)\vec{r}_{tb}^{\,e}(t) + C_e^i(t)\dot{\vec{r}}_{tb}^{\,e}(t) \\ &= \Omega_{ie}^i C_e^i(t)\vec{r}_{et}^{\,e} + \Omega_{ie}^i C_e^i(t)\vec{r}_{tb}^{\,e}(t) + C_e^i(t)\dot{\vec{r}}_{tb}^{\,e}(t) \\ &= \vec{\omega}_{ie}^{\,i} \times C_e^i(t)\vec{r}_{et}^{\,e} + \vec{\omega}_{ie}^{\,i} \times C_e^i(t)\vec{r}_{tb}^{\,e}(t) + C_e^i(t)\dot{\vec{r}}_{tb}^{\,e}(t)\end{aligned} \tag{2.4}$$

The derivative is repeated to reveal acceleration terms that capture the specific force phenomenon measured by the accelerometer.

$$
\begin{aligned}
\ddot{\vec{r}}^{\,i}_{ib}(t) &= \frac{d}{dt}\left[\vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\vec{r}^{\,e}_{et} + \vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\vec{r}^{\,e}_{tb}(t) + C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t)\right] \\
&= \vec{\omega}^{i}_{ie}\times \dot{C}^{i}_{e}(t)\vec{r}^{\,e}_{et} + \vec{\omega}^{i}_{ie}\times \dot{C}^{i}_{e}(t)\vec{r}^{\,e}_{tb}(t) + \vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + \ldots \\
&\quad \dot{C}^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + C^{i}_{e}(t)\ddot{\vec{r}}^{\,e}_{tb}(t) \\
&= \vec{\omega}^{i}_{ie}\times \Omega^{i}_{ie}C^{i}_{e}(t)\vec{r}^{\,e}_{et} + \vec{\omega}^{i}_{ie}\times \Omega^{i}_{ie}C^{i}_{e}(t)\vec{r}^{\,e}_{tb}(t) + \vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + \ldots \\
&\quad \Omega^{i}_{ie}C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + C^{i}_{e}(t)\ddot{\vec{r}}^{\,e}_{tb}(t) \\
&= \vec{\omega}^{i}_{ie}\times(\vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\vec{r}^{\,e}_{et}) + \vec{\omega}^{i}_{ie}\times\left(\vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\vec{r}^{\,e}_{tb}(t)\right) + \vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + \ldots \\
&\quad \vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t) + C^{i}_{e}(t)\ddot{\vec{r}}^{\,e}_{tb}(t) \\
\ddot{\vec{r}}^{\,i}_{ib}(t) &= \vec{\omega}^{i}_{ie}\times\left(\vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\vec{r}^{\,e}_{eb}(t)\right) + 2\left(\vec{\omega}^{i}_{ie}\times C^{i}_{e}(t)\dot{\vec{r}}^{\,e}_{tb}(t)\right) + C^{i}_{e}(t)\ddot{\vec{r}}^{\,e}_{tb}(t)
\end{aligned}
\tag{2.5}
$$

These three terms make up the different forms of acceleration seen from the ECI frame. The first term is the centripetal acceleration, the second is the acceleration due to the Coriolis effect, and the third is the linear acceleration experienced in the body frame with respect to the tangential frame.

The acceleration vector $\ddot{\vec{r}}^{\,i}_{ib}(t)$ is broken up into different terms in (2.6). This acceleration vector breaks up into the specific force $\vec{f}^{\,i}_{ib}$ that consists of all forces the body experiences that are not due to gravity and the mass gravity attraction vector $\vec{\gamma}^{\,i}_{ib}$ that consists of all the forces due to the mass attraction of gravity. The mass gravity attraction vector $\vec{\gamma}^{\,i}_{ib}$ also splits into two different pieces, the force due to gravity resulting from the mass of the robot $\vec{g}^{\,i}_{b}$ and the centripetal acceleration keeping the mobile robot moving around the Earth's axis (*i.e.*, $\vec{\omega}^{i}_{ie}\times\left(\vec{\omega}^{i}_{ie}\times\vec{r}^{\,i}_{ib}\right)$).

$$
\begin{aligned}
\ddot{\vec{r}}^{\,i}_{ib}(t) = \vec{a}^{\,i}_{ib}(t) &= \vec{f}^{\,i}_{ib}(t) + \vec{\gamma}^{\,i}_{ib} = \vec{f}^{\,i}_{ib} + \vec{g}^{\,i}_{b} + \vec{\omega}^{i}_{ie}\times\left(\vec{\omega}^{i}_{ie}\times\vec{r}^{\,i}_{ib}(t)\right) \\
\vec{a}^{\,i}_{ib}(t) &= \vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} + \vec{\omega}^{i}_{ie}\times\left(\vec{\omega}^{i}_{ie}\times\vec{r}^{\,i}_{ib}(t)\right)
\end{aligned}
\tag{2.6}
$$

With all acceleration terms present in the relationship between acceleration $\ddot{\vec{r}}^{\,i}_{ib}(t)$ and the specific force $\vec{f}^{\,i}_{ib}$, the acceleration in the *t*-frame can now be specified as a differential equation by bring (2.5) and (2.6) together.

$$\begin{gathered}
\ddot{\vec{r}}^{\,i}_{ib}(t) = \dot{\vec{v}}^{\,i}_{ib}(t) = \vec{\omega}^{i}_{ie} \times \left(\vec{\omega}^{i}_{ie} \times C^{i}_{e}(t) \vec{r}^{\,e}_{eb}(t) \right) + 2\left(\vec{\omega}^{i}_{ie} \times C^{i}_{e}(t) \dot{\vec{r}}^{\,e}_{tb}(t) \right) + C^{i}_{e}(t) \ddot{\vec{r}}^{\,e}_{tb}(t) \\
\vec{a}^{\,i}_{ib}(t) = \vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} + \vec{\omega}^{i}_{ie} \times \left(\vec{\omega}^{i}_{ie} \times \vec{r}^{\,i}_{ib}(t) \right) \\
\dot{\vec{v}}^{\,i}_{ib}(t) = \vec{a}^{\,i}_{ib}(t) \\
\cancel{\vec{\omega}^{i}_{ie} \times \left(\vec{\omega}^{i}_{ie} \times C^{i}_{e}(t) \vec{r}^{\,e}_{eb}(t) \right)} + 2\left(\vec{\omega}^{i}_{ie} \times C^{i}_{e}(t) \dot{\vec{r}}^{\,e}_{tb}(t) \right) + C^{i}_{e}(t) \ddot{\vec{r}}^{\,e}_{tb}(t) = \ldots \\
\vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} + \cancel{\vec{\omega}^{i}_{ie} \times \left(\vec{\omega}^{i}_{ie} \times \vec{r}^{\,i}_{ib}(t) \right)} \\
C^{i}_{e}(t) \ddot{\vec{r}}^{\,e}_{tb}(t) = \vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} - 2\left(\vec{\omega}^{i}_{ie} \times C^{i}_{e}(t) \dot{\vec{r}}^{\,e}_{tb}(t) \right) \\
C^{i}_{e}(t) C^{e}_{t} \dot{\vec{v}}^{\,t}_{ib}(t) = \vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} - 2\left(\Omega^{i}_{ie} \vec{v}^{\,i}_{tb}(t) \right) \\
C^{i}_{e}(t) C^{e}_{t} \dot{\vec{v}}^{\,t}_{ib}(t) \left[C^{i}_{e}(t) C^{e}_{t} \right]^{-1} = \left(\vec{f}^{\,i}_{ib}(t) + \vec{g}^{\,i}_{b} - 2\left(\Omega^{i}_{ie} \vec{v}^{\,i}_{tb}(t) \right) \right) \left[C^{i}_{e}(t) C^{e}_{t} \right]^{-1} \\
\dot{\vec{v}}^{\,t}_{ib}(t) = \vec{f}^{\,t}_{ib}(t) + \vec{g}^{\,t}_{b} - 2 C^{t}_{e} \Omega^{e}_{ie} C^{e}_{t} \vec{v}^{\,t}_{tb}(t) \\
\dot{\vec{v}}^{\,t}_{tb}(t) = C^{t}_{b}(t) \vec{f}^{\,b}_{ib} + \vec{g}^{\,t}_{b} - 2 C^{t}_{e} \Omega^{e}_{ie} C^{e}_{t} \vec{v}^{\,t}_{tb}(t)
\end{gathered} \tag{2.7}$$

A similar differential relationship is found between $C^{t}_{b}(t)$ and $\dot{C}^{t}_{b}(t)$, capturing how gyroscope measurements are propagated into attitude measurements.

$$
\begin{aligned}
\dot{C}_b^i(t) &= \frac{d}{dt}\left[C_e^i(t)C_t^eC_b^t(t)\right] \\
LHS &= C_b^i(t)\Omega_{ib}^b(t) \\
RHS &= \dot{C}_e^iC_t^eC_b^t(t) + C_e^i(t)C_t^e\dot{C}_b^t(t) \\
&= \Omega_{ie}^iC_e^i(t)C_t^eC_b^t(t) + C_t^i(t)\dot{C}_b^t(t) \\
C_t^i(t)\dot{C}_b^t(t) &= C_b^i(t)\Omega_{ib}^b(t) - \Omega_{ie}^iC_e^i(t)C_t^eC_b^t(t) \\
\dot{C}_b^t(t) &= C_i^t(t)C_b^i(t)\Omega_{ib}^b(t) - \left[C_b^t(t)C_t^b(t)\right]C_i^t(t)\Omega_{ie}^iC_b^i(t) \\
&= C_b^t(t)\Omega_{ib}^b(t) - C_b^t(t)C_i^b(t)\Omega_{ie}^iC_b^i(t) \\
&= C_b^t(t)\left[\Omega_{ib}^b(t) - \Omega_{ie}^b\right]
\end{aligned}
\tag{2.8}
$$

Results from equations (2.7) and (2.8) provide differential relationships to build a continuous-time model of how PVA propagates with inputs from the accelerometer and gyroscope, as shown in (2.9).

$$
\begin{bmatrix} \dot{\vec{r}}_{tb}^t(t) \\ \dot{\vec{v}}_{tb}^t(t) \\ \dot{C}_b^t(t) \end{bmatrix} = \begin{bmatrix} \vec{v}_{tb}^t(t) \\ C_b^t(t)\vec{f}_{ib}^b(t) + \vec{g}_b^t - 2C_e^t\Omega_{ie}^eC_t^e\vec{v}_{tb}^t(t) \\ C_b^t(t)(\Omega_{ib}^b(t) - \Omega_{ie}^b) \end{bmatrix} \tag{2.9}
$$

IMU Error Sources

Both inertial sensors are subject to a variety of error sources. Furthermore, each error source has a varying contribution to PVA error, making some error sources more severe than others. The relationship between "ground truth" quantities, sensor measurements (*i.e.*, $\tilde{\vec{\omega}}_{ib}^b$), and their error (*i.e.*, $\Delta\vec{\omega}_{ib}^b$) is defined in (2.10).

$$\tilde{\vec{f}}_{ib}^{b} = \vec{f}_{ib}^{b} + \Delta\vec{f}_{ib}^{b}$$
$$\tilde{\vec{\omega}}_{ib}^{b} = \vec{\omega}_{ib}^{b} + \Delta\vec{\omega}_{ib}^{b} \tag{2.10}$$

For consumer-grade micro-electrical mechanical systems (MEMS) IMUs, error sources in the accelerometer and gyroscope function similarly. Each sensor is subject to fixed biases ($\vec{b}_{a,FB}$ or $\vec{b}_{g,FB}$), bias stability ($\vec{b}_{a,BS}$ or $\vec{b}_{g,BS}$), and bias instability ($\vec{b}_{a,BI}$ or $\vec{b}_{g,BI}$). Fixed biases are consistent between each power cycle of the IMU, while bias stability errors change from turn-on to turn-on of the IMU. Bias instability errors vary in-run and are typically modeled via a first-order Gauss-Markov process shown in (2.12).

$$\vec{b}_{a} = \vec{b}_{a,FB} + \vec{b}_{a,BS} + \vec{b}_{a,BI}$$
$$\vec{b}_{g} = \vec{b}_{g,FB} + \vec{b}_{g,BS} + \vec{b}_{g,BI} \tag{2.11}$$

$$\dot{\vec{b}}_{a,BI}(t) = \left(\frac{-1}{\tau_{a,BI}}\right)\vec{b}_{a,BI}(t) + \vec{\eta}_{a,BI} \quad where\ \vec{\eta}_{a,BI} \sim N\left[0, \sigma_{a,BI}\right]$$
$$\dot{\vec{b}}_{g,BI}(t) = \left(\frac{-1}{\tau_{g,BI}}\right)\vec{b}_{g,BI}(t) + \vec{\eta}_{a,BI} \quad where\ \vec{\eta}_{g,BI} \sim N\left[0, \sigma_{g,BI}\right] \tag{2.12}$$

Each sensor is also subject to scale-factor and misalignment errors. Scale-factor errors occur when a sensing axis measures a value proportional to the true value and are typically minimal, yet can lead to PVA error accumulation. Misalignment errors capture the imperfect orthogonality of the sensing axes. Each error source is captured as a ratio and both are represented in a misalignment matrix demonstrated in (2.13).

$$M_a = \begin{bmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{bmatrix}$$
$$M_g = \begin{bmatrix} s_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & s_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & s_{g,z} \end{bmatrix} \tag{2.13}$$

The gyroscope contains one additional error source, namely gyroscope sensitivity G_g, where some accelerations are sensed by the gyroscope creating erroneous measurements.

The remaining unspecified error sources are approximated as white noise. White noise inputs into the accelerometer and gyroscope lead to velocity random walks (VRW) and angle random walks (ARW) once integrated. Each white-noise process is assumed to be zero-mean. A power spectral density is often captured to confirm the white-noise assumption and determine the amount of noise present.

Equation (2.14) provides the full error model of the accelerometer and gyroscope, depicting how each error source distorts the ground-truth value.

$$\begin{aligned} \tilde{\vec{f}}_{ib}^{b} &= \vec{f}_{ib}^{b} + \Delta\vec{f}_{ib}^{b} = \vec{b}_a + \left(I_{3x3} + M_a\right)\vec{f}_{ib}^{b} + \vec{w}_a \\ \tilde{\vec{\omega}}_{ib}^{b} &= \vec{\omega}_{ib}^{b} + \Delta\vec{\omega}_{ib}^{b} = \vec{b}_g + \left(I_{3x3} + M_g\right)\vec{\omega}_{ib}^{b} + G_g\vec{f}_{ib}^{b} + \vec{w}_g \end{aligned} \tag{2.14}$$

A MATLAB application [8] was developed to accept sensor parameters in Original Equipment Manufacturer (OEM) provided datasheet units and create standardized binary (.mat) descriptor files for a given IMU. An example for the VectorNav VN-200™ IMU is shown in Figure 10.

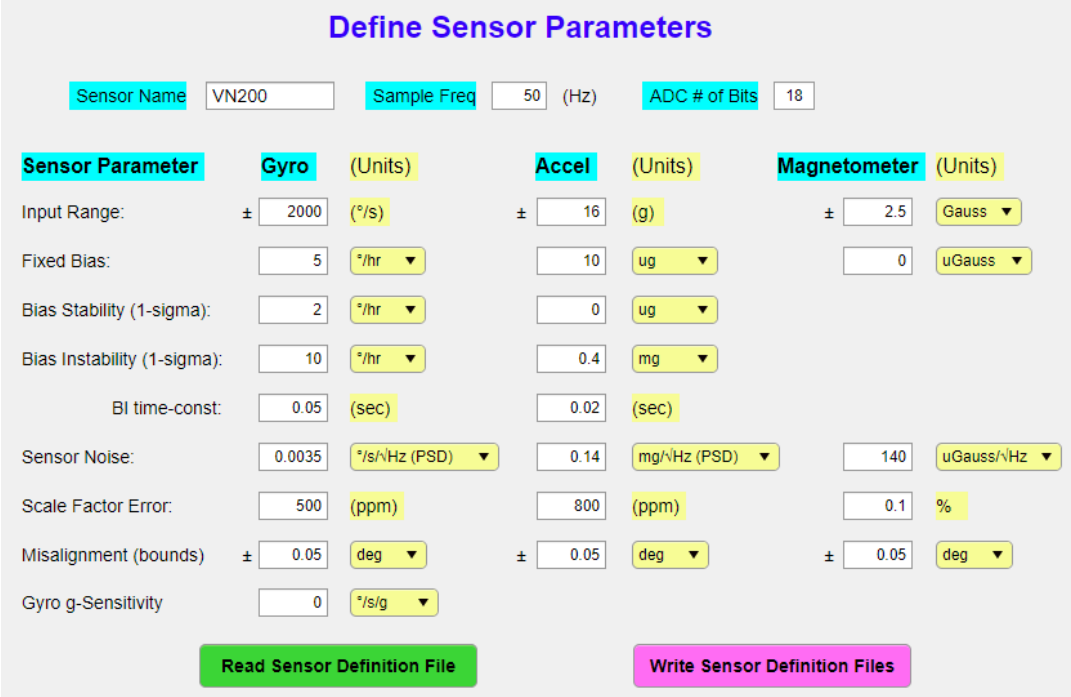

Figure 10 - VectorNav VN-200™ Error Characteristics

Figure 10 depicts the quality of the IMU in terms of expected error source quantities. Many error sources can be removed via calibration prior to operation, including fixed bias, scale factor, and misalignment. However, other error sources such as gyroscope ARW and bias instabilities cannot; hence, they provide intrinsic coordinates (see Figure 11) for representing sensor performance [8].

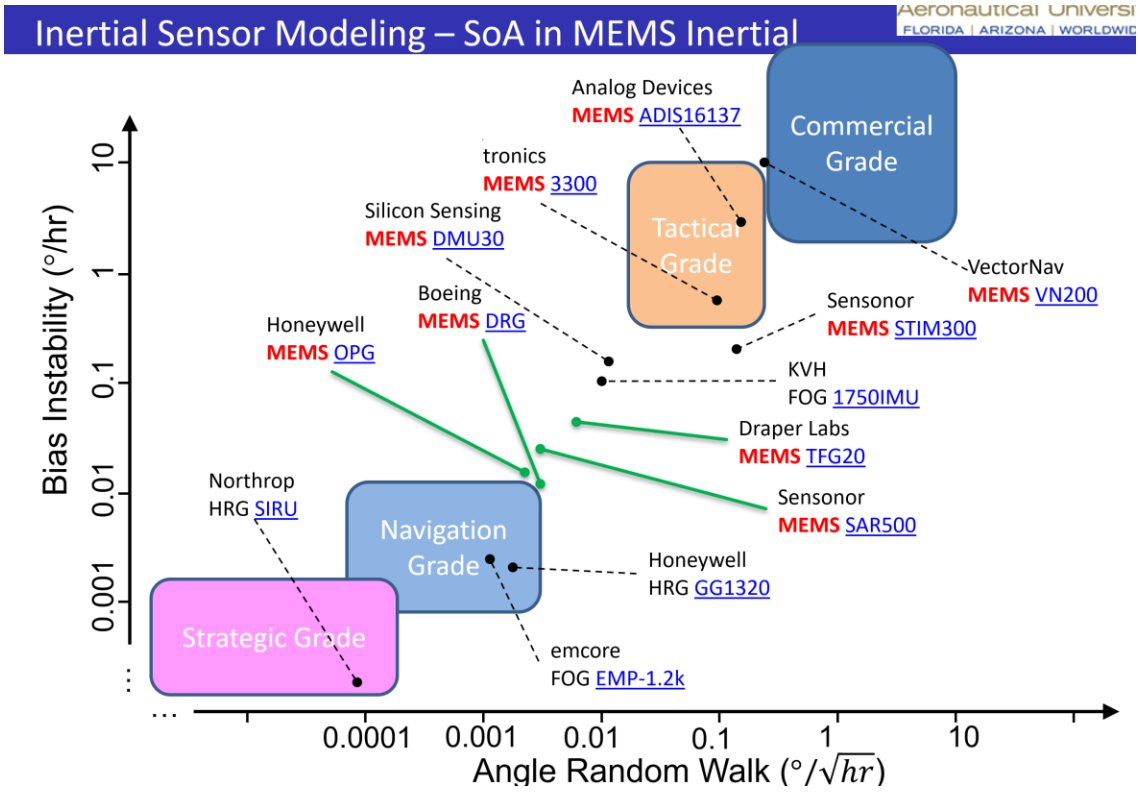

Figure 11 - A comparison of gyroscope performance

Different IMU grades serve different purposes and different applications. The VectorNav VN-200™ can be considered as a consumer-grade IMU.

IMU Error Contribution Comparison

During IMU operation, error sources cause error to accumulate in the PVA solution, known as inertial drift. Each error source causes PVA error accumulation at different rates, which can serve as a means to prioritize aiding sensor selection. The contribution of PVA error from each error source is compared in the following sections to justify aiding sensor selection.

Bias Error

Bias errors are commonly the most severe error sources when left uncalibrated. While fixed biases are easily calibrated and removed in software during IMU operation, bias stability is more challenging to account for during the IMU's operation. Imprecise calibrations can give rise to residual bias errors, thus leading to faster PVA error accumulation. Furthermore, bias instability itself cannot be removed via an off-line calibration due to its stochastic nature. Instead, this error source is typically mitigated by aiding sensors in the INS. For the simple analysis below, consider a worst-case scenario of when bias instability remains at its 1σ level.

Consider a one-dimensional example of integrating accelerometer bias (b_a) versus integrating gyroscope bias. When integrating accelerometer bias into position error, the error will grow at a rate $\propto t^2$.

$$\begin{aligned} e_a(t) &= b_a \\ e_v(t) &= \int_0^t b_a d\tau = b_a t \\ e_r(t) &= \int_0^t b_a t dt = \frac{1}{2} b_a t^2 \end{aligned} \tag{2.15}$$

However, when integrating gyroscope bias (b_g) into position error, the error will instead grow at a rate $\propto t^3$. The resulting angular error becomes $e_\theta(t) = \int_0^t b_g d\tau = b_g t$, causing an effective acceleration bias of $a_{avg} \sin\left(e_\theta(t)\right) \approx a_{avg} e_\theta(t) = a_{avg} b_g t$ (small angle assumption), hence,

$$
\begin{aligned}
e_a(t) &= a_{avg} b_g t \\
e_v(t) &= \int_0^t a_{avg} b_g t dt = \frac{1}{2} a_{avg} b_g t^2 \\
e_r(t) &= \int_0^t \frac{1}{2} a_{avg} b_g t^2 dt = \frac{1}{6} a_{avg} b_g t^3
\end{aligned}
\tag{2.16}
$$

The difference in position error accumulation rates in (2.15) and (2.16) make it clear that gyroscope bias errors can have a more significant contribution to PVA error than accelerometer bias errors.

Scale Factor Error

Scale factor errors also contribute to PVA error accumulation. While typically accounted for in calibration and removed in software, imprecise calibrations leave room for scale factor errors to cause faster inertial drift. Their contribution follows similar rates as bias error terms but at different proportions.

Consider a one-dimensional example of integrating accelerometer scale factor errors versus integrating gyroscope scale factor errors. When integrating accelerometer scale factor error into position error, the error will grow at a rate $\propto t^2$.

$$
\begin{aligned}
e_a(t) &= \left(1 + s_{a,x}\right) a_{avg} - a_{avg} = s_{a,x} a_{avg} \\
e_v(t) &= \int_0^t s_{a,x} a_{avg} d\tau = s_{a,x} a_{avg} t \\
e_r(t) &= \int_0^t s_{a,x} a_{avg} t dt = \frac{1}{2} s_{a,x} a_{avg} t^2
\end{aligned}
\tag{2.17}
$$

When integrating gyroscope scale factor error into position error, the error will instead grow at a rate $\propto t^3$.

$$
\begin{aligned}
e_\omega(t) &= \left(1+s_{g,x}\right)\omega_{avg} - \omega_{avg} = s_{g,x}\omega_{avg} \\
e_\theta(t) &= \int_0^t s_{g,x}\omega_{avg}\,d\tau = s_{g,x}\omega_{avg}t \\
e_a(t) &= a_{avg}\sin\left(e_\theta(t)\right) \approx a_{avg}e_\theta(t) \\
e_a(t) &= s_{g,x}a_{avg}\omega_{avg}t \\
e_v(t) &= \int_0^t s_{g,x}a_{avg}\omega_{avg}t\,dt \\
e_v(t) &= \frac{1}{2}s_{g,x}a_{avg}\omega_{avg}t^2 \\
e_r(t) &= \int_0^t \frac{1}{2}s_{g,x}a_{avg}\omega_{avg}t^2\,dt \\
e_r(t) &= \frac{1}{6}s_{a,x}a_{avg}\omega_{avg}t^3
\end{aligned}
\tag{2.18}
$$

The difference in position error accumulation rates in (2.17) and (2.18) make clear that gyroscope scale factor errors can have a more significant contribution to PVA error than accelerometer scale factor errors.

Misalignment Error

Misalignment error of the IMU sensing axes causes significant PVA error accumulation. While this error source is typically calibrated and removed before IMU operation, the position error accumulation rates still point to gyroscope misalignment having a more significant error contribution than accelerometer misalignment.

Consider a one-dimensional example of integrating accelerometer misalignment versus integrating gyroscope misalignment. When integrating accelerometer misalignment into position error, the error will grow at a rate $\propto t^2$.

$$\begin{aligned} e_a(t) &= a_{avg}\sin\left(\delta\theta_{a,xy}\right) \approx m_{a,xy}a_{avg} \\ e_v(t) &= \int_0^t m_{a,xy}a_{avg}d\tau = m_{a,xy}a_{avg}t \\ e_r(t) &= \int_0^t m_{a,xy}a_{avg}tdt = \frac{1}{2}m_{a,xy}a_{avg}t^2 \end{aligned} \tag{2.19}$$

When integrating gyroscope misalignment into position error, the error will instead grow at a rate $\propto t^3$.

$$\begin{aligned} e_\omega(t) &= \omega_{avg}\sin\left(\delta\theta_{g,xy}\right) \approx m_{g,xy}\omega_{avg} \\ e_\theta(t) &= \int_0^t m_{g,xy}\omega_{avg}d\tau = m_{g,xy}\omega_{avg}t \\ e_a(t) &= a_{avg}\sin\left(e_\theta(t)\right) \approx a_{avg}e_\theta(t) \\ e_a(t) &= m_{g,xy}a_{avg}\omega_{avg}t \\ e_v(t) &= \int_0^t m_{g,xy}a_{avg}\omega_{avg}tdt \\ e_v(t) &= \frac{1}{2}m_{g,xy}a_{avg}\omega_{avg}t^2 \\ e_r(t) &= \int_0^t \frac{1}{2}m_{g,xy}a_{avg}\omega_{avg}t^2dt \\ e_r(t) &= \frac{1}{6}m_{g,xy}a_{avg}\omega_{avg}t^3 \end{aligned} \tag{2.20}$$

The difference in position error accumulation rates in (2.19) and (2.20) make it clear that gyroscope misalignment has a more significant contribution to PVA error than accelerometer misalignment.

Velocity Random Walk and Angle Random Walk

VRW from accelerometer white noise and ARW from gyroscope white noise are the primary source of inertial drift in well calibrated IMUs. Integrating white noise leads to Brownian motion and random walks, denying the opportunity for removal via off-line calibration. Instead, inertial drift from these random walks is addressed via aiding sensors in the overall INS.

Consider a one-dimensional example of integrating accelerometer white noise versus integrating gyroscope white noise. When integrating VRW into position error, the error will grow at a rate $\propto t^{3/2}$.

$$\left[VRW\right] = \left[\frac{m}{s^2}\right] \Big/ \sqrt{Hz}$$

$$\begin{aligned} e_v(t) &= VRW\sqrt{t} \\ e_r(t) &= \int_0^t VRW\sqrt{t}\,dt = \frac{2}{3}VRWt^{3/2} \end{aligned} \tag{2.21}$$

When integrating ARW into position error, the error will instead grow at a rate $\propto t^{5/2}$.

$$\left[ARW\right] = \left[\frac{rad}{s}\right] \Big/ \sqrt{Hz}$$

$$\begin{aligned} e_\theta(t) &= ARW\sqrt{t} \\ e_a(t) &= a_{avg}\sin\left(e_\theta(t)\right) \approx a_{avg}ARW\sqrt{t} \\ e_v(t) &= \int_0^t a_{avg}ARW\sqrt{t}\,dt = \frac{2}{3}a_{avg}ARWt^{3/2} \\ e_r(t) &= \int_0^t \frac{2}{3}a_{avg}ARWt^{3/2}\,dt \\ e_r(t) &= \frac{4}{15}a_{avg}ARWt^{5/2} \end{aligned} \tag{2.22}$$

The difference in position error accumulation rates in (2.21) and (2.22) make it clear that ARW have a more detrimental impact on PVA error than VRW.

Monte Carlo Analysis of Inertial Drift

The analysis in the previous section assumes worst-case scenarios of inertial drift due to stochastic processes in bias instabilities and VRW/ARW. A Monte Carlo analysis is performed to properly compare the severity of inertial drift due to these stochastic processes.

Consider two test cases involving a calibrated VectorNav VN-200™ IMU at rest for two minutes to collect data. In test case 1, all accelerometer measurements are perfect, while gyroscope measurement errors solely contribute to position error. In test case 2, only accelerometer measurement errors contribute to position error, while all gyroscope measurements are perfect. Ideally, the IMU should measure that it remains at rest; however, different error sources will cause PVA error to accumulate and cause the entire PVA solution to drift away from its initial resting location. Each test case is simulated one hundred times, each with its own independent stochastic processes for time-varying error sources. The disparity in position error growth due to gyroscope error sources versus accelerometer error sources substantiates the need for including an attitude aiding source in the overall INS.

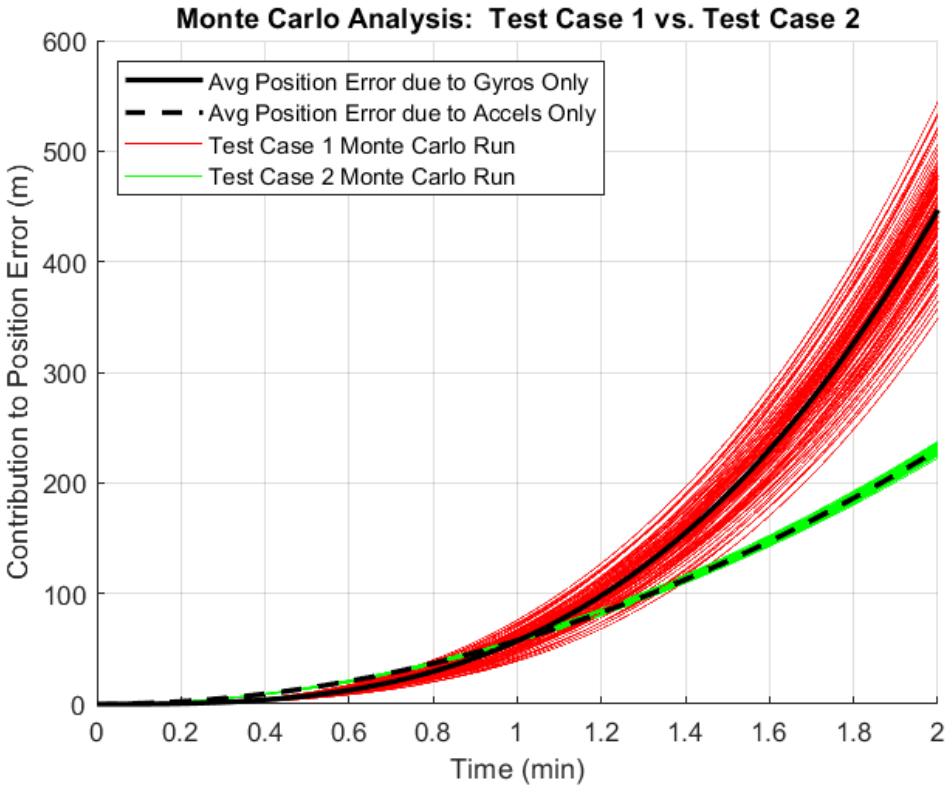

Figure 12 - Disparity in Inertial Drift due to separate inertial sensor error

Figure 12 confirms that gyroscope error is the dominant contributor to inertial drift compared to accelerometer error. Therefore, when considering which aiding sensors to incorporate into the overall INS, the need for an attitude aiding source is paramount.

This realization inspires the investigation of employing a depth camera in an indoor environment for attitude aiding and becomes the focus of the remainder of this thesis.

Chapter 3 - Extracting Attitude from Depth Camera Images

Depth cameras such as the Kinect™ camera have become a popular sensor choice in the last decade, and their possible applications continue to grow [9]. Depth cameras return point clouds of their surrounding environment, as shown in Figure 13.

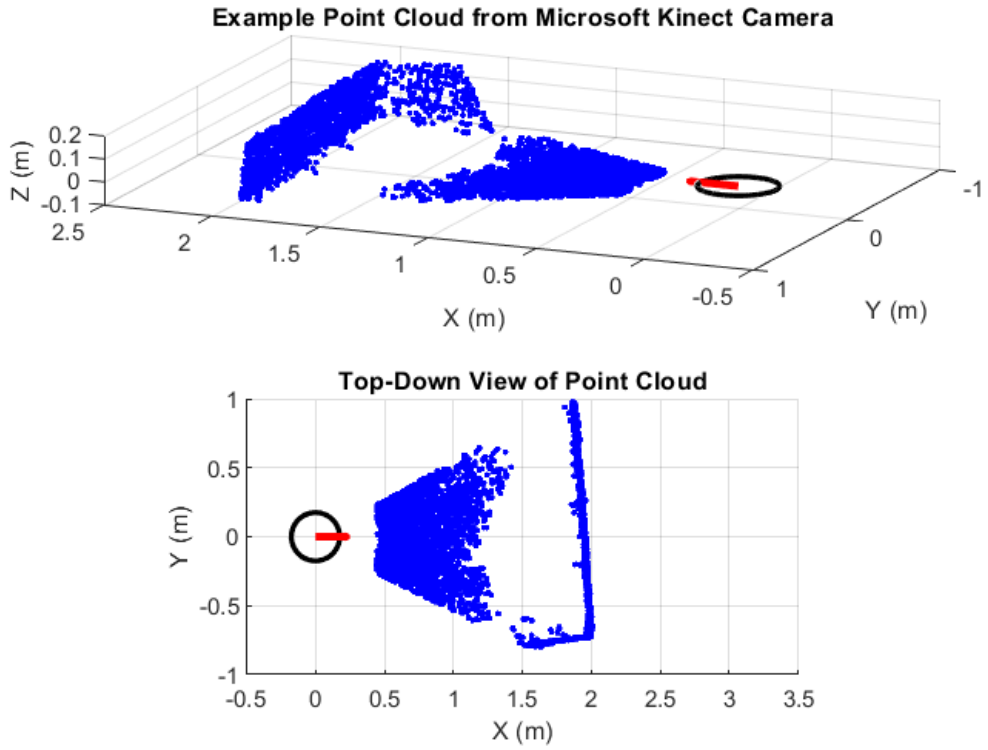

Figure 13 - Point Cloud from a Microsoft Kinect™ Camera onboard a Quanser Qbot 2™

For humans, identifying walls and floors in the image is a simple task. Defining an algorithm to robustly extract walls and floors from unorganized sensor data, however, is substantially more difficult. Several feature extraction techniques exist for point clouds [10], one of the most popular being the Hough Transform [11]. The Hough Transform employs a parameterization of the desired feature (*i.e.*, edge, plane) to transform each point in the dataset

into a new feature space. This new space is called the Hough space, in which votes are stored in an accumulator structure indicating the most likely parameterization for the feature given the provided data.

The Hough Transform has been modified and adapted to accomplish many different tasks [12], but the core concept remains the same.

Hough Transform for Attitude Extraction

Viewing the point cloud from Figure 13, consider the possibility of extracting attitude information from the floors and walls surrounding the Qbot 2™. The Qbot 2™ has a single axle with two wheels and a differential drive that allows the robot to translate and rotate in place. Two castor wheels are placed perpendicular to the axle, allowing the robot to pitch back and forth. Due to these kinematic constraints of this specific robotic platform, determining the pitch θ and yaw ψ of the Quanser Qbot 2™ are of paramount interest. Due to the single axle and lack of suspension, the roll ϕ of the Qbot 2™ remains nominally zero during its travel on a flat floor. It is assumed that the floor always remains locally level and flat in the indoor environment.

Using the Hough transform to extract attitude from the depth camera point clouds, directly addresses the problem of PVA error accumulation. Given that the gyroscope is the primary contributor to PVA error, having an accurate attitude aiding source is expected to improve PVA estimation performance dramatically [3].

Surface Normal Hough Transformation Derivation

Consider the top-down (*i.e.*, 2D) view of an example point cloud that resembles a wall at a given angle ψ_{wall} and a signed distance from the origin ρ. Upon visual inspection, the plane contains a surface normal vector $\vec{n}$ aligned with the angle ψ_{wall}; however, if only provided the point cloud data, determining $\vec{n}$ is not obvious. The points resemble a wall but are not perfectly coplanar due to Kinect™ camera sensor noise.

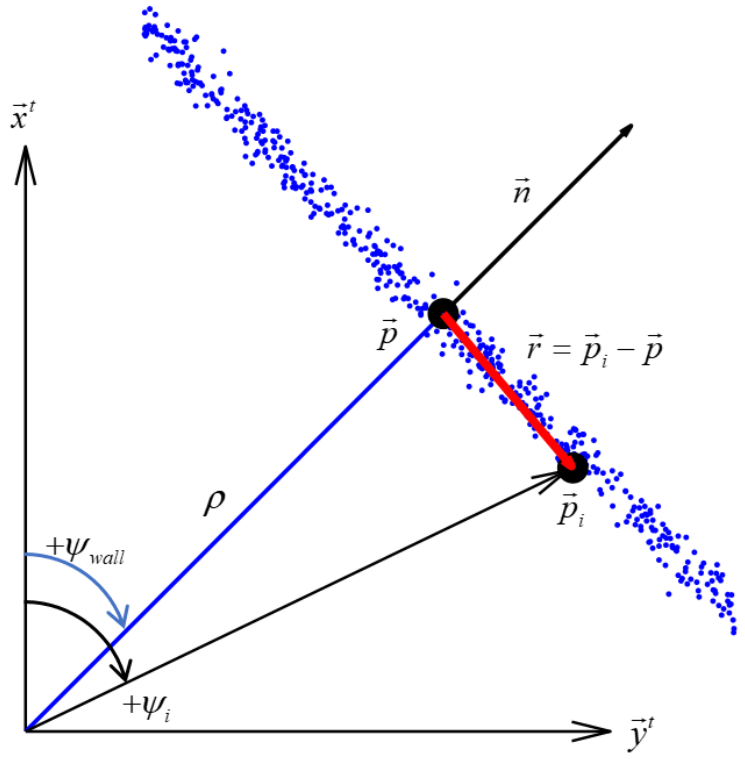

Figure 14 - Top-Down View of an Example Plane in the Proposed Hough Space.

To extract the surface normal vector $\vec{n}$ from this feature, consider a unit vector aligned with the x-axis, which is then rotated by some amount of yaw ψ, where $\cos(\psi) \doteq c_\psi$ and $\sin(\psi) \doteq s_\psi$.

$$\vec{n} = R_z(\psi) \cdot \vec{u}_x = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c_\psi \\ s_\psi \\ 0 \end{bmatrix} \tag{3.1}$$

Referring to Figure 14, consider the test point $\vec{p}_i$, which lies in the point cloud. If $\vec{p}_i$ truly lies in the plane, the vector $\vec{r} = \vec{p}_i - \vec{p}$ will be normal to the surface normal vector $\vec{n}$, and hence their dot product will equal zero. Furthermore, the point $\vec{p}$ is defined by the product of the surface normal vector $\vec{n}$ and the signed distance to the origin ρ.

$$\begin{gathered} \vec{r} \cdot \vec{n} = (\vec{p}_i - \vec{p}) \cdot \vec{n} = (\vec{p}_i - \rho\vec{n}) \cdot \vec{n} = 0 \\ \begin{bmatrix} x_i - \rho c_\psi \\ y_i - \rho s_\psi \\ 0 \end{bmatrix}^T \begin{bmatrix} c_\psi \\ s_\psi \\ 0 \end{bmatrix} = 0 \\ x_i c_\psi + y_i s_\psi = \rho \end{gathered} \tag{3.2}$$

Equation (3.2) provides the transform of a Cartesian coordinate $\vec{p}_i = \begin{bmatrix} x_i & y_i & 0 \end{bmatrix}^T$ into the Hough space $\begin{bmatrix} \psi & \rho \end{bmatrix}$.

It is important to note that this specific parameterization defines a front wall feature. Similar parameterizations are provided for side wall features and floor features provided in (3.3) respectively.

$$\begin{aligned} -x_i c_\psi + y_i s_\psi &= \rho \\ -x_i s_\theta + z_i c_\theta &= \rho \end{aligned} \tag{3.3}$$

Surface Normal Hough Transform Algorithm

The Surface Normal Hough Transform (SNHT) algorithm is performed three times, once for each feature type: floor, front wall, and side wall. Before beginning the transformation, the point cloud is split into two sections, points belonging to the floor and points belong to any wall. One SNHT search occurs for points belonging to the floor, while two SNHT searches occur for points belonging to the wall. The algorithm begins by defining a search space for the angle in question. Then, every Cartesian point in the point cloud is transformed into the Hough space according to its given feature parameterization for every search angle previously defined. Each transformation computes a value of ρ, which then fully parameterizes one possible plane.

Once the entire point cloud is processed, each possible plane defined by the search angle ψ_i or θ_i and the computed value ρ is stored as a vote in an accumulator. Many accumulator designs exist [12]; however, the accumulator design for this algorithm resembles a two-dimensional histogram. For each point in the point cloud, one vote is made for each test angle ψ_i or θ_i. Example pseudocode is shown in Table 1.

Table 1 - Example Surface Normal Hough Transform Pseudocode

Step	SNHT Algorithm Steps
1	Given a point cloud XYZ
2	Define search space $\psi \in [\psi_{min} < \psi_i < \psi_{max}]$
3	for every point in the point cloud XYZ
4	for every search angle ψ_i
5	Compute ρ according to Hough transformation
6	Store vote in the accumulator at location $[\psi_i, \rho]$
7	end for
8	end for
9	Normalize accumulator

Pseudocode of the SNHT for one feature

At the end of the SNHT algorithm, the accumulator will serve as the joint probability density function (pdf) of the search angle ψ_i or θ_i and ρ. The location of the peak in the joint pdf serves as the most likely (*i.e.*, mode) parameterization of the plane. Interpreting the accumulator as a joint pdf is critical for establishing the measurement uncertainty for the Kalman filter described in the next section. An example joint pdf produced by the SNHT wall algorithm is shown in Figure 15.

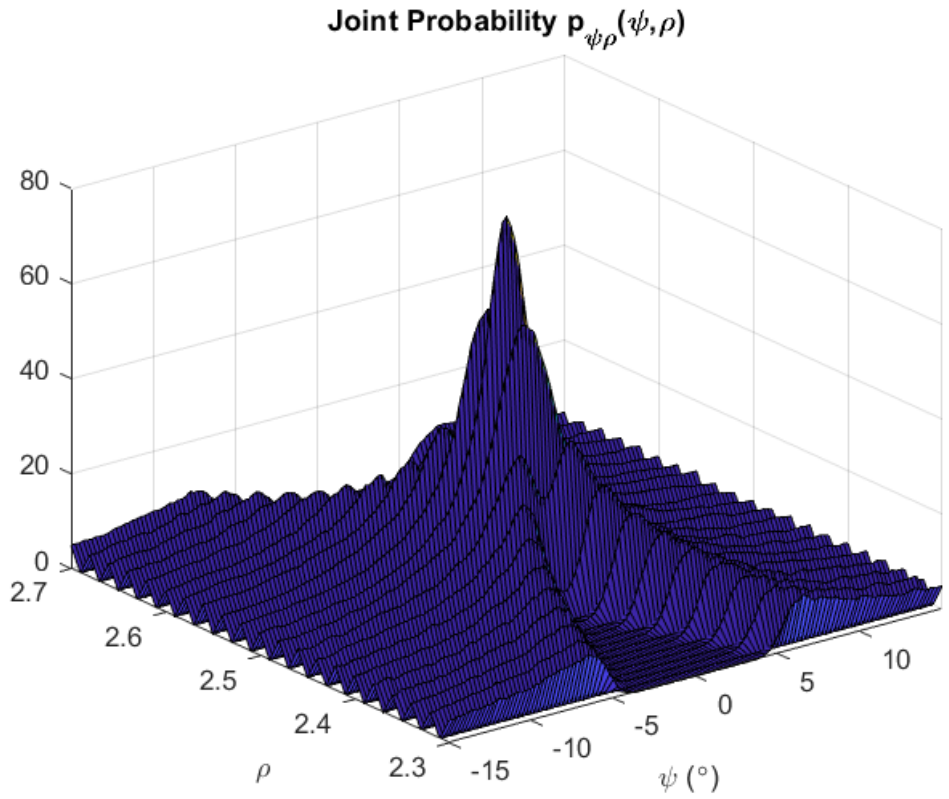

Figure 15 - Accumulator design yielding a joint pdf of ψ and ρ

To extract attitude and its uncertainty from the joint pdf, the marginal probability mass function (*pmf*) of ρ is first computed. The mode of the *pmf* will determine ρ_{ML}, the most likely value of ρ. Then, a conditional slice of the joint pdf will be taken at the location ρ_{ML}. This

conditional slice, $p(\psi \mid \rho = \rho_{ML})$, then provides the most likely value of the search angle, as well as the measurement uncertainty. An example of this is shown in Fig. 10. The fact that this approach to processing 3D depth data naturally provides the uncertainty of the measurement is a fundamental benefit of this technique.

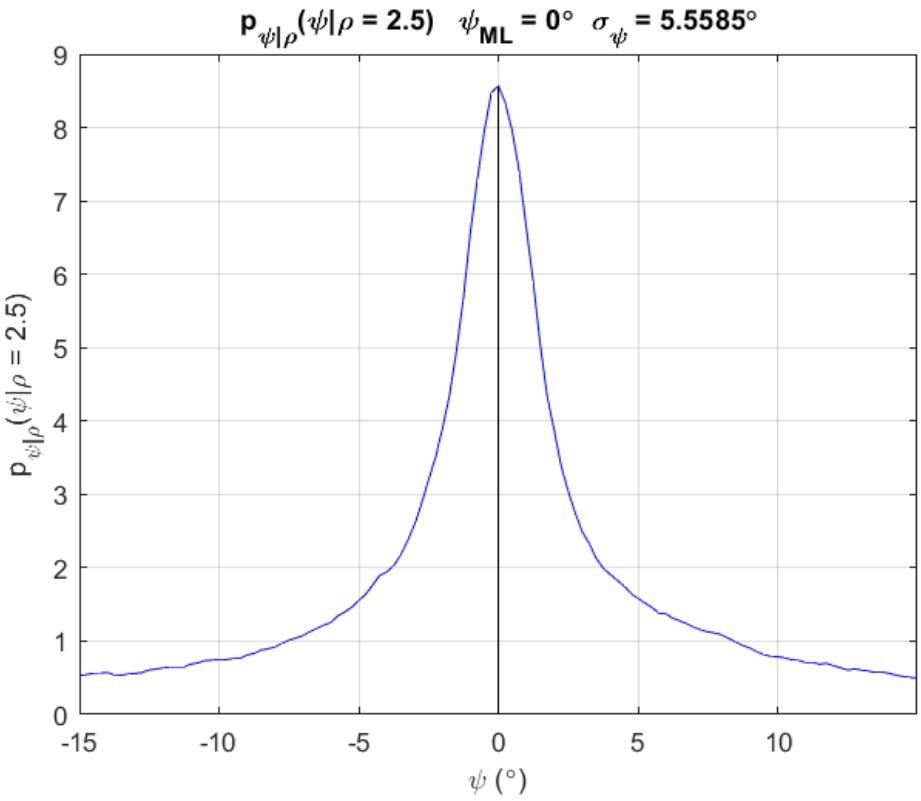

Figure 16 - Conditional probability of ψ given ρ, providing measurement and uncertainty

Kinect™ Camera Noise Characteristics

Extensive work has gone into determining the noise characteristics of the Kinect™ camera [13]. Depth cameras acquire their measurements via different methods. The Kinect™ camera itself determines depth via triangulation [14]. A rigorous transformation from depth measurement uncertainty to Cartesian covariance ellipsoids is provided by [15]. To establish confidence in the standard deviations produced by the SNHT algorithm, it would seem reasonable to determine the transformation from Cartesian covariance ellipsoids to SNHT standard deviations. Instead, the SNHT standard deviations prove to be invariant to noise in the Cartesian space within reason.

This claim was substantiated by the following experiment. A Qbot 2™ equipped with a Kinect™ camera was placed on a cart constrained to move along a track. The track was placed against a flat wall, and distances from the wall were measured.

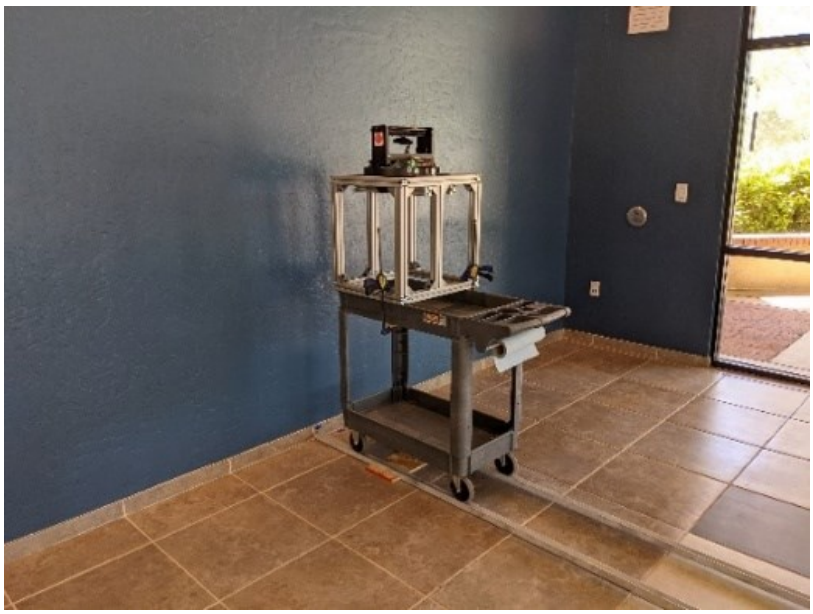

Figure 17 - Qbot 2™, Cart, Track, and Flat Wall for Noise Characterization

The Kinect™ camera started at 0.5 meters from the wall and point clouds of the wall were collected from 0.5 meters to 3 meters, in steps of 5 centimeters, as shown in Figure 17.

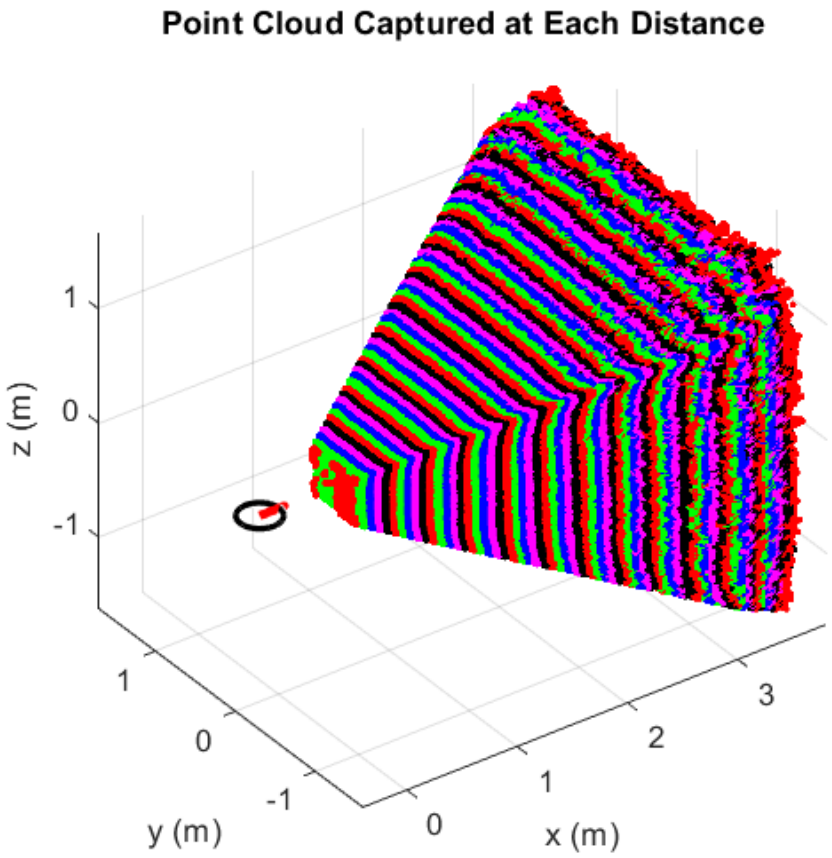

Figure 18 - Point clouds from the Kinect™ camera at various distances

Results from [13] were confirmed, in that, point cloud thickness grows quadratically with distance, as shown in Figure 19.

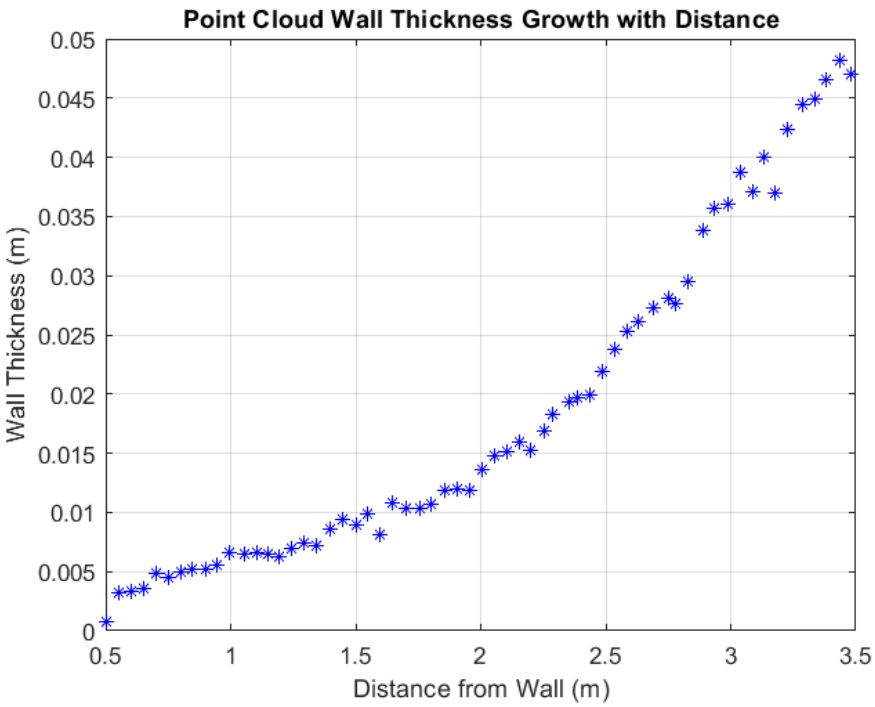

Figure 19 - Quadratic growth of point cloud wall thickness

Intuitively, one would expect that as thickness in the point cloud increased, SNHT standard deviations would also grow. Instead, the increase in thickness has no effect, as shown in Figure 20.

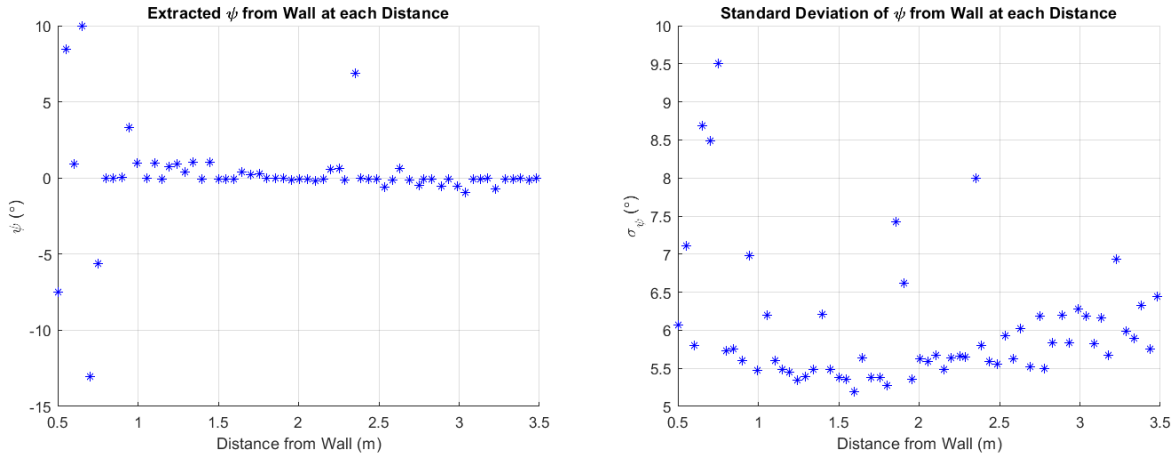

Figure 20 - Extracted ψ and measurement uncertainty from each distance from the wall

The results from this experiment provide confidence in the generated SNHT standard deviations and provide insight into what characteristics impact the final standard deviations produced by the final SNHT algorithm.

Extracting Attitude from an Indoor Environment

A typical indoor environment consists of predominantly mutually orthogonal walls and floors. While exceptions to this orthogonal configuration exist, it is assumed that all floors are flat, and all walls are vertical for the purposes of extracting attitude. Thus, surface normals to the perpendicular walls will serve as the $\vec{x}^t$ and $\vec{y}^t$ axes of the tangential frame, and their attitude relative to the mobile robot can provide an absolute measurement of the mobile robot's attitude $\tilde{C}^t_{b,cam}$.

Assume that the mobile robot travels in a box with walls surrounding the travel path, as illustrated in Figure 21.

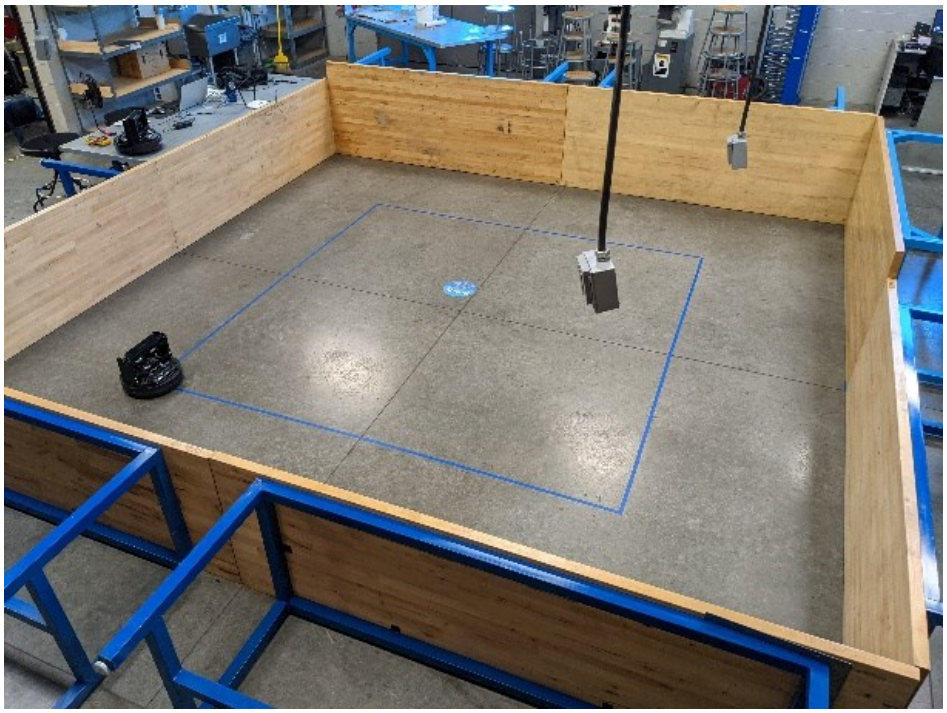

Figure 21 - Simple Box Test Environment

First, it is crucial to determine which axis the mobile robot is traveling along. It is assumed that the robot begins traveling along the positive $\vec{x}^t$ axis. Using the best available measurement or estimate of the mobile robot's attitude of the *b*-frame to the *t*-frame $\hat{C}_b^t$, each column of the directional cosine matrix (DCM) represents the local axis to which it is aligned. Using a variation of the Mahalanobis distance, one can determine which axis the mobile robot is traveling along.

$$\vec{x}^{t+} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ \vec{x}^{t-} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix},\ y^{t+} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \vec{y}^{t-} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\hat{C}_b^t = \begin{bmatrix} \hat{\vec{x}}_b^t & \hat{\vec{y}}_b^t & \hat{\vec{z}}_b^t \end{bmatrix}$$

$$d_{\vec{x}^{t+}} = \frac{\sqrt{(\hat{\vec{x}}_b^t - \vec{x}^{t+})^T \cdot (\hat{\vec{x}}_b^t - \vec{x}^{t+})}}{2}$$

$$d_{\vec{x}^{t-}} = \frac{\sqrt{(\hat{\vec{x}}_b^t - \vec{x}^{t-})^T \cdot (\hat{\vec{x}}_b^t - \vec{x}^{t-})}}{2}$$

$$d_{\vec{y}^{t+}} = \frac{\sqrt{(\hat{\vec{y}}_b^t - \vec{y}^{t+})^T \cdot (\hat{\vec{y}}_b^t - \vec{y}^{t+})}}{2}$$

$$d_{\vec{y}^{t-}} = \frac{\sqrt{(\hat{\vec{y}}_b^t - \vec{y}^{t-})^T \cdot (\hat{\vec{y}}_b^t - \vec{y}^{t-})}}{2} \tag{3.4}$$

Equation (3.4) demonstrates the computation of each Mahalanobis distance to determine which of the axes the mobile robot is most likely aligned. The smallest Mahalanobis distances computed are the most likely axes to which the *b*-frame is aligned such that:

$$
\begin{aligned}
&For\ \hat{\vec{x}}_b^t: \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{x}^{t+}}\right] \rightarrow\ \hat{\vec{x}}_b^t = \vec{x}^{t+} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{x}^{t-}}\right] \rightarrow\ \hat{\vec{x}}_b^t = \vec{x}^{t-} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{y}^{t+}}\right] \rightarrow\ \hat{\vec{x}}_b^t = \vec{y}^{t+} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{y}^{t-}}\right] \rightarrow\ \hat{\vec{x}}_b^t = \vec{y}^{t-} \\
&\\
&For\ \hat{\vec{y}}_b^t: \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{x}^{t+}}\right] \rightarrow\ \hat{\vec{y}}_b^t = \vec{x}^{t+} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{x}^{t-}}\right] \rightarrow\ \hat{\vec{y}}_b^t = \vec{x}^{t-} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{y}^{t+}}\right] \rightarrow\ \hat{\vec{y}}_b^t = \vec{y}^{t+} \\
&\left[\min(d_{\vec{x}^{t+}}, d_{\vec{x}^{t-}}, d_{\vec{y}^{t+}} d_{\vec{y}^{t-}}) = d_{\vec{y}^{t-}}\right] \rightarrow\ \hat{\vec{y}}_b^t = \vec{y}^{t-}
\end{aligned}
\tag{3.5}
$$

Next, the attitude of the *b*-frame with respect to the *t*-frame is determined via the SNHT. Referring to Figure 13, a front wall, a side, and a floor are present in the point cloud. The surface normal vector belonging to each surface will ideally align with the *t*-frame axes respectively; however, there is no guarantee that the mobile robot's body will be aligned perfectly to the walls at any time. Three SNHT searches are completed for each expected surface, returning the measurements $\left[\theta_f^c \quad \psi_{fw}^c \quad \psi_{sw}^c\right]$ and their standard deviations $\left[\sigma_f \quad \sigma_{fw} \quad \sigma_{sw}\right]$.

The Euler angles captured by each SNHT can be transformed into their respective surface normal vectors accordingly.

$$\theta_f^c = -\theta_f^t, \quad \psi_{fw}^c = -\psi_{fw}^t, \quad \psi_{sw}^c = -\psi_{sw}^t$$
$$\vec{n}_f = \begin{bmatrix} -s_{\theta_f^t} \\ 0 \\ c_{\theta_f^t} \end{bmatrix}, \ \vec{n}_{fw} = \begin{bmatrix} c_{\psi_{fw}^t} \\ s_{\psi_{fw}^t} \\ 0 \end{bmatrix}, \ \vec{n}_{sw} = \begin{bmatrix} -c_{\psi_{sw}^t} \\ s_{\psi_{sw}^t} \\ 0 \end{bmatrix} \tag{3.6}$$

Only one of the two wall surface normal vectors is used to extract attitude. The higher quality of the two measurements is chosen according to which measurement has a lower standard deviation. Then, the measured *t*-frame axes are computed as shown in (3.7).

$$\begin{bmatrix} \tilde{\vec{x}}_t^b \\ \tilde{\vec{y}}_t^b \\ \tilde{\vec{z}}_t^b \end{bmatrix} = \begin{bmatrix} \vec{n}_{fw} \\ -(\vec{n}_{fw} \times \vec{n}_f) \\ \vec{n}_f \end{bmatrix}, \quad \sigma_{\psi_{fw}} \leq \sigma_{\psi_{sw}}$$
$$\begin{bmatrix} \tilde{\vec{x}}_t^b \\ \tilde{\vec{y}}_t^b \\ \tilde{\vec{z}}_t^b \end{bmatrix} = \begin{bmatrix} \vec{n}_{sw} \times \vec{n}_f \\ \vec{n}_{sw} \\ \vec{n}_f \end{bmatrix}, \quad \sigma_{\psi_{sw}} < \sigma_{\psi_{fw}} \tag{3.7}$$

Once the body frame axes are computed, the DCM $\tilde{C}_{b,cam}^t$ can be constructed.

$$\tilde{C}_{t,cam}^b = \begin{bmatrix} \tilde{\vec{x}}_t^b \cdot \hat{\vec{x}}_b^t & \tilde{\vec{y}}_t^b \cdot \hat{\vec{x}}_b^t & \tilde{\vec{z}}_t^b \cdot \hat{\vec{x}}_b^t \\ \tilde{\vec{x}}_t^b \cdot \hat{\vec{y}}_b^t & \tilde{\vec{y}}_t^b \cdot \hat{\vec{y}}_b^t & \tilde{\vec{z}}_t^b \cdot \hat{\vec{y}}_b^t \\ \tilde{\vec{x}}_t^b \cdot \hat{\vec{z}}_b^t & \tilde{\vec{y}}_t^b \cdot \hat{\vec{z}}_b^t & \tilde{\vec{z}}_t^b \cdot \hat{\vec{z}}_b^t \end{bmatrix}$$
$$\tilde{C}_{b,cam}^t = \left[\tilde{C}_{t,cam}^b \right]^T \tag{3.8}$$

Chapter 4 – Aiding Sensor Configurations

The synergy between the short-term precise inertial sensors and long-term accurate aiding sensors is compelling in determining PVA [3]. The problem can be formulated to produce an estimate of PVA or the error in PVA resulting from an inertial-only PVA solution. The latter error-space approach is sometimes referred to as the "go-free" concept [16]. The dominant contributor to growth in the inertial-only PVA error is the gyroscope; however, the accelerometer's impact should not be ignored.

The entire PVA estimation takes place in two phases, PVA error prediction and PVA error measurement updates. PVA error prediction consists of building a model of how PVA error will likely evolve. PVA error measurement updates use aiding sensors to measure some combination of position, velocity, or attitude, compare that measurement to the PVA generated by the IMU, and update the overall estimate of PVA error. Finally, these PVA error measurement updates are blended with PVA error predictions according to model uncertainty and measurement uncertainty to return the most likely estimate of PVA error.

PVA Error Prediction

Performing PVA prediction via the "go-free" concept frees the state estimator of concerning itself with mobile robot's kinematics and dynamics. Instead of using PVA directly as the state variables, the "go-free" concept uses PVA error estimates and associated PVA error dynamics. To mathematically distinguish between ground-truth values, measurements, and estimates, the following notation is used.

$$\begin{aligned}\vec{x} &\leftrightarrow ground\ truth\\ \tilde{\vec{x}} &\leftrightarrow measurement\\ \hat{\vec{x}} &\leftrightarrow estimate\end{aligned} \tag{4.1}$$

The relationship between ground truth, measurements, and the error between the two is defined in (2.10). To estimate true error in PVA (*i.e.*, $\Delta\vec{r}_{tb}^{\,t}$, $\Delta\vec{v}_{tb}^{\,t}$, ΔC_b^t), the relationship between ground truth and estimates is defined as:

$$\begin{aligned}\vec{x} &= \hat{\vec{x}} + \delta\vec{x}\\ \delta\vec{x} &= \vec{x} - \hat{\vec{x}}\end{aligned} \tag{4.2}$$

Representing position and velocity error estimates follow the relationship described in (4.2); however, representing attitude error as a DCM creates problems for Kalman filter state vectors. So instead, attitude error estimates are represented in the angle-axis format shown in (4.3) as (the equivalent of truth - estimate)

$$\begin{aligned}\delta C_b^t &= C_b^t\left[\hat{C}_b^t\right]^T\\ &= e^{\delta\Psi_{tb}^t}\\ &\approx I_3 + \delta\Psi_{tb}^t\\ \delta\Psi_{tb}^t &\approx C_b^t\left[\hat{C}_b^t\right]^T - I_3\end{aligned} \tag{4.3}$$

where $\delta\Psi_{tb}^t \triangleq sk\left[\delta\vec{\psi}_{tb}^{\,t}\right]$

The approximation made in (4.3) is valid only for relatively small angles. Should estimates of attitude error grow beyond an acceptable range, the approximation will no longer be valid. Noting that $\left(\delta C_b^t\right)^{-1} = \left(\delta C_b^t\right)^T \approx \left(I_3 + \delta\Psi_{tb}^t\right)^T = I_3 - \delta\Psi_{tb}^t$, as $\left(\delta\Psi_{tb}^t\right)^T = -\delta\Psi_{tb}^t$, from (4.3),

estimates of attitude in a DCM format relate to their ground-truth quantity in the following manner:

$$\begin{aligned} C_b^t &= \left(I_3 + \delta\Psi_{tb}^t\right)\hat{C}_b^t \\ \hat{C}_b^t &= (I_3 - \delta\Psi_{tb}^t)C_b^t \end{aligned} \tag{4.4}$$

With error in PVA fully defined, the Kalman filter state estimate vector is defined in (4.5) as

$$\delta\vec{x} \triangleq \begin{bmatrix} \delta\vec{\psi}_{tb}^t \\ \delta\vec{v}_{tb}^t \\ \delta\vec{r}_{tb}^t \end{bmatrix} \tag{4.5}$$

Tangential Error Mechanization

Recall the tangential mechanization defined in (2.9), repeated below.

$$\begin{bmatrix} \dot{\vec{r}}_{tb}^t(t) \\ \dot{\vec{v}}_{tb}^t(t) \\ \dot{C}_b^t(t) \end{bmatrix} = \begin{bmatrix} \vec{v}_{tb}^t(t) \\ C_b^t(t)\vec{f}_{ib}^b + \vec{g}_b^t - 2C_e^t\Omega_{ie}^e C_t^e\vec{v}_{tb}^t(t) \\ C_b^t(t)(\Omega_{ib}^b - \Omega_{ie}^b) \end{bmatrix}$$

A similar continuous-time model can be defined to describe how PVA errors evolve according to error in IMU measurements. This model will serve as a means of predicting PVA error in the Kalman filter.

To determine how attitude error evolves in response to gyroscope error, $\dot{C}_b^t(t)$ is expanded and substituted for estimated quantities, as shown in (4.6).

$$\begin{aligned}\dot{C}_b^t(t) &= \frac{d}{dt}\left[\left(I_3 + \delta\Psi_{tb}^t(t)\right)\hat{C}_b^t(t)\right] \\ &= \delta\dot{\Psi}_{tb}^t(t)\hat{C}_b^t(t) + \left(I_3 + \delta\Psi_{tb}^t(t)\right)\dot{\hat{C}}_b^t(t)\end{aligned} \tag{4.6}$$

For consumer-grade IMU's, the Earth's rotation is well below the noise floor and unmeasurable. Thus, ignoring the effects of the Earth's rotation, the right-hand side of the tangential mechanization undergoes similar substitutions and expansions in (4.7). Similarly,

$$\begin{aligned}\dot{C}_b^t(t) &= C_b^t(t)\Omega_{tb}^b \\ &= \left(I_3 + \delta\Psi_{tb}^t(t)\right)\hat{C}_b^t(t)\Omega_{tb}^b \\ &= \left(I_3 + \delta\Psi_{tb}^t(t)\right)\hat{C}_b^t(t)\left(\hat{\Omega}_{tb}^b + \delta\Omega_{tb}^b\right) \\ &= \left(I_3 + \delta\Psi_{tb}^t(t)\right)\hat{C}_b^t(t)\hat{\Omega}_{tb}^b + \hat{C}_b^t(t)\delta\Omega_{tb}^b + \cancel{\delta\Psi_{tb}^t(t)\hat{C}_b^t(t)\delta\Omega_{tb}^b} \\ &\approx \left(I_3 + \delta\Psi_{tb}^t(t)\right)\hat{C}_b^t(t)\hat{\Omega}_{tb}^b + \hat{C}_b^t(t)\delta\Omega_{tb}^b \\ &\approx \left(I_3 + \delta\Psi_{tb}^t(t)\right)\dot{\hat{C}}_b^t(t) + \hat{C}_b^t(t)\delta\Omega_{tb}^b\end{aligned} \tag{4.7}$$

Equating . (4.6) and (4.7) reveals the following:

$$\begin{aligned}\delta\dot{\Psi}_{tb}^t(t)\hat{C}_b^t(t) + \left(I_3 + \delta\Psi_{tb}^t(t)\right)\dot{\hat{C}}_b^t(t) &= \left(I_3 + \delta\Psi_{tb}^t(t)\right)\dot{\hat{C}}_b^t(t) + \hat{C}_b^t(t)\delta\Omega_{tb}^b \\ \delta\dot{\Psi}_{tb}^t(t)\hat{C}_b^t(t) &= \hat{C}_b^t(t)\delta\Omega_{tb}^b \\ \delta\dot{\Psi}_{tb}^t(t) &= \hat{C}_b^t(t)\delta\Omega_{tb}^b\left[\hat{C}_b^t(t)\right]^T\end{aligned} \tag{4.8}$$

Transforming attitude error from a skew-symmetric matrix to an angle-axis vector provides the relationship between attitude error, it's derivative, and gyroscope error in (4.9).

$$
\begin{aligned}
\delta\dot{\vec{\psi}}^t_{tb}(t) &= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{tb} = \hat{C}^t_b(t)\left(\delta\vec{\omega}^b_{ib} - \delta\vec{\omega}^b_{ie}\right) \\
&= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{ib} - \hat{C}^t_b(t)\left(\vec{\omega}^b_{ie} - \hat{\vec{\omega}}^b_{ie}\right) \\
&= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{ib} - \hat{C}^t_b(t)\left(C^b_t(t)\,\vec{\omega}^t_{ie} - \hat{C}^b_t(t)\,\vec{\omega}^t_{ie}\right) \\
&= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{ib} - \left(\hat{C}^t_b(t)\,C^b_t(t) - I_3\right)\vec{\omega}^t_{ie} \\
&= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{ib} + \delta\Psi^t_{tb}(t)\,\vec{\omega}^t_{ie} \\
&= \hat{C}^t_b(t)\,\delta\vec{\omega}^b_{ib} - \Omega^t_{ie}\,\delta\vec{\psi}^t_{tb}(t)
\end{aligned}
\tag{4.9}
$$

To determine the relationship between velocity error, its derivative, and error in accelerometer measurements, ground-truth quantities are substituted for their estimate and estimate error quantities in the tangential mechanization system in (2.9). The left-hand side can be expanded as:

$$
\begin{aligned}
\dot{\vec{v}}^t_{tb}(t) &= \dot{\hat{\vec{v}}}^t_{tb}(t) + \delta\dot{\vec{v}}^t_{tb}(t) \\
&= \hat{C}^t_b(t)\,\hat{\vec{f}}^b_{ib} + \hat{\vec{g}}^t_b + 2\Omega^t_{ie}\,\hat{\vec{v}}^t_{tb}(t) + \delta\dot{\vec{v}}^t_{tb}(t)
\end{aligned}
\tag{4.10}
$$

The right-hand side can be expanded as:

$$
\begin{aligned}
C^t_b(t)\,\vec{f}^b_{ib} + \vec{g}^t_b - 2C^t_e\Omega^e_{ie}C^e_t\vec{v}^t_{tb}(t) &= \left(I_3 + \delta\Psi^t_{tb}(t)\right)\hat{C}^t_b(t)\left(\hat{\vec{f}}^b_{ib} + \delta\vec{f}^b_{ib}\right) + \ldots \\
&\quad \left(\hat{\vec{g}}^t_b + \delta\vec{g}^t_b\right) - 2\Omega^t_{ie}\left(\hat{\vec{v}}^t_{tb} + \delta\vec{v}^t_{tb}\right) \\
&= \hat{C}^t_b(t)\,\hat{\vec{f}}^b_{ib} + \hat{C}^t_b(t)\,\delta\vec{f}^b_{ib} + \delta\Psi^t_{tb}(t)\,\hat{C}^t_b(t)\,\hat{\vec{f}}^b_{ib} + \ldots \\
&\quad \delta\Psi^t_{tb}(t)\,\hat{C}^t_b(t)\,\delta\vec{f}^b_{ib} + \hat{\vec{g}}^t_b + \delta\vec{g}^t_b - 2\Omega^t_{ie}\,\hat{\vec{v}}^t_{tb} - 2\Omega^t_{ie}\,\delta\vec{v}^t_{tb} \\
&\approx \hat{C}^t_b(t)\,\hat{\vec{f}}^b_{ib} + \hat{C}^t_b(t)\,\delta\vec{f}^b_{ib} + \delta\Psi^t_{tb}(t)\,\hat{C}^t_b(t)\,\hat{\vec{f}}^b_{ib} + \ldots \\
&\quad \hat{\vec{g}}^t_b + \delta\vec{g}^t_b - 2\Omega^t_{ie}\,\hat{\vec{v}}^t_{tb} - 2\Omega^t_{ie}\,\delta\vec{v}^t_{tb}
\end{aligned}
\tag{4.11}
$$

Bringing both sides together leaves the remaining terms below in (4.12).

$$\begin{aligned}\delta\dot{\vec{v}}_{tb}^{t}(t) &= \hat{C}_{b}^{t}(t)\,\vec{f}_{ib}^{b} + \delta\Psi_{tb}^{t}(t)\,\hat{C}_{b}^{t}(t)\,\delta\vec{f}_{ib}^{b} + \delta\vec{g}_{b}^{t} - 2\Omega_{ie}^{t}\,\delta\vec{v}_{tb}^{t} \\ &= \left[\delta\vec{\psi}_{tb}^{t}(t)\times\right]\hat{C}_{b}^{t}(t)\,\delta\vec{f}_{ib}^{b} + \hat{C}_{b}^{t}(t)\,\delta\vec{f}_{ib}^{b} + \delta\vec{g}_{b}^{t} - 2\Omega_{ie}^{t}\,\delta\vec{v}_{tb}^{t}\end{aligned} \tag{4.12}$$

A gravity model is used from [3], which provides the magnitude and direction of the acceleration due to gravity for a given latitude and position on earth in (4.13). This model is transformed to the *t*-frame to substitute into (4.12).

$$\begin{aligned}\delta\vec{g}_{b}^{e} &= \frac{2g_0\left(\hat{L}_b\right)\vec{r}_{eb}^{\,e}}{r_{eS}^{e}\left(\hat{L}_b\right)\left|\vec{r}_{eb}^{\,e}\right|^2}\left(\vec{r}_{eb}^{\,e}\right)^{T}\delta\vec{r}_{eb}^{e} \\ \delta\vec{g}_{b}^{t} &= C_{e}^{t}\frac{2g_0\left(\hat{L}_b\right)\vec{r}_{eb}^{\,e}}{r_{eS}^{e}\left(\hat{L}_b\right)\left|\vec{r}_{eb}^{\,e}\right|^2}\left(\vec{r}_{eb}^{\,e}\right)^{T}C_{t}^{e}\,\delta\vec{r}_{tb}^{t}\end{aligned} \tag{4.13}$$

The relationship between velocity error, its derivative, and accelerometer error is expressed in (4.14).

$$\begin{aligned}\delta\dot{\vec{v}}_{tb}^{t}(t) = &-\left[\hat{C}_{b}^{t}(t)\,\delta\vec{f}_{ib}^{b}\times\right]\delta\vec{\psi}_{tb}^{t}(t) + \hat{C}_{b}^{t}(t)\,\delta\vec{f}_{ib}^{b} + \ldots \\ &\left[C_{e}^{t}\frac{2g_0\left(\hat{L}_b\right)\hat{\vec{r}}_{eb}^{\,e}}{r_{eS}^{e}\left(\hat{L}_b\right)\left|\hat{\vec{r}}_{eb}^{\,e}\right|^2}\left(\hat{\vec{r}}_{eb}^{\,e}\right)^{T}C_{t}^{e}\right]\delta\vec{r}_{tb}^{t} - 2\Omega_{ie}^{t}\,\delta\vec{v}_{tb}^{t}\end{aligned} \tag{4.14}$$

Given both (4.9) and (4.14), the full continuous-time model encompassing the whole PVA mechanization is given as:

$$\begin{bmatrix} \dot{\delta\vec{\psi}}^t_{tb} \\ \dot{\delta\vec{v}}^t_{tb} \\ \dot{\delta\vec{r}}^t_{tb} \end{bmatrix} = \begin{bmatrix} -\Omega^t_{ie} & 0_3 & 0_3 \\ -\left[\hat{C}^t_b \vec{f}^b_{ib} \times\right] & -2\Omega^t_{ie} & C^t_e \dfrac{2g_0\left(\hat{L}_b\right)\hat{\vec{r}}^e_{eb}}{r^e_{eS}\left(\hat{L}_b\right)\left|\hat{\vec{r}}^e_{eb}\right|^2}\left(\hat{\vec{r}}^e_{eb}\right)^T C^e_t \\ 0_3 & I_3 & 0_3 \end{bmatrix} \begin{bmatrix} \dot{\delta\vec{\psi}}^t_{tb} \\ \dot{\delta\vec{v}}^t_{tb} \\ \dot{\delta\vec{r}}^t_{tb} \end{bmatrix} + \begin{bmatrix} 0_3 & \hat{C}^t_b \\ \hat{C}^t_b & 0_3 \\ 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \delta\vec{f}^b_{ib} \\ \delta\vec{\omega}^b_{ib} \end{bmatrix} \quad (4.15)$$

Kalman Filter Prediction Model

The Kalman filter prediction model relies primarily on a dynamic model of the tangential error mechanization, however not entirely. The model inputs themselves consist of IMU errors, which are unknown during its operation. Instead, these IMU error terms must be augmented to reflect what IMU error sources are expected to be present.

Recall the IMU error models provided in (2.14). Most error sources such as fixed biases and misalignments are calibrated and removed prior to IMU operation. However, other error sources such as bias instabilities and white noise inputs that cannot be corrected by calibration are shown in (4.16).

$$\begin{aligned} \delta\vec{f}^b_{ib} &\approx -\vec{b}_{a,BI} - \vec{w}_a \\ \delta\vec{\omega}^b_{ib} &\approx -\vec{b}_{g,BI} - \vec{w}_g \end{aligned} \quad (4.16)$$

The first-order Gauss-Markov process used to model bias instability in (2.12) allows for the possibility to augment the state vector demonstrated in (4.17).

$$\dot{\vec{x}} = A \vec{x} + \vec{\varpi}$$

$$\begin{bmatrix} \delta\dot{\vec{\psi}}^t_{tb} \\ \delta\dot{\vec{v}}^t_{tb} \\ \delta\dot{\vec{r}}^t_{tb} \\ \dot{\vec{b}}_{a,BI} \\ \dot{\vec{b}}_{g,BI} \end{bmatrix} = \begin{bmatrix} -\Omega^t_{ie} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\hat{C}^t_b \\ -\left[\hat{C}^t_b \vec{f}^b_{ib} \times\right] & -2\Omega^t_{ie} & A_{2,3} & -\hat{C}^t_b & 0_{3\times3} \\ 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -I_3 \big/ \tau_{a,BI} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -I_3 \big/ \tau_{g,BI} \end{bmatrix} \begin{bmatrix} \delta\vec{\psi}^t_{tb} \\ \delta\vec{v}^t_{tb} \\ \delta\vec{r}^t_{tb} \\ \vec{b}_{a,BI} \\ \vec{b}_{g,BI} \end{bmatrix} + \begin{bmatrix} -\hat{C}^t_b \vec{w}_g \\ -\hat{C}^t_b \vec{w}_a \\ \vec{0}_{3\times1} \\ \vec{\eta}_{a,BI} \\ \vec{\eta}_{g,BI} \end{bmatrix} \tag{4.17}$$

$$where\ A_{2,3} = C^t_e \frac{2g_0\left(\hat{L}_b\right)\hat{\vec{r}}^e_{eb}}{r^e_{eS}\left(\hat{L}_b\right)\left|\hat{\vec{r}}^e_{eb}\right|^2}\left(\hat{\vec{r}}^e_{eb}\right)^T C^e_t$$

The model above is the full Kalman filter prediction model in continuous time, which captures all expected IMU error sources and how they will contribute to PVA error over time. Implementing this model on a computer requires this model to be converted to a discrete-time model shown in (4.18).

$$\dot{\vec{x}}(t) = A\vec{x}(t) + \vec{\varpi}(t)$$
$$F = e^{A\Delta T} \approx I_{3\times3} + A\Delta T$$
$$\vec{x}(k) \approx F\,\vec{x}(k-1) + \vec{w}(k)$$

$$\bar{Q}(k) \triangleq E\left[\vec{w}(k)\vec{w}(k)^T\right] = \begin{bmatrix} \sigma^2_g & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma^2_a & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \sigma^2_{a,BI} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \sigma^2_{g,BI} \end{bmatrix} \tag{4.18}$$

$$Q(k) \approx \frac{1}{2}\left(F\bar{Q}(k)F^T + \bar{Q}(k)\right)\Delta T$$

$$\vec{x}(k) = F\,\vec{x}(k-1) + Q(k)$$

PVA Error Measurement Updates

Measurement updates are used to aid the prediction model and correct estimates when information is available. These updates require the use of sensors external to the IMU, typically referred to as aiding sensors. When incorporating the "go-free" concept, aiding sensors are used not to provide absolute measurements of PVA but rather measurements of the error in PVA computed by the IMU.

The IMU is precise in nature and can capture high-dynamic motion in the short term. Inertial drift, however, prevents the IMU from remaining accurate. Therefore, it is vital to ensure that aiding sensor selection is accurate to compensate for the shortcomings of the IMU.

The two upcoming sections describe odometry and depth camera aiding, the first establishing a traditional aiding approach for mobile robotics followed by a new approach based on attitude measurement opportunities described in Chapter 3.

Odometry Aiding

The odometer is one of the most common aiding sensors in mobile robotics [17]. The odometer provides an accurate long-term measurement of linear and angular velocity in the body frame as

$$\begin{aligned} \tilde{\vec{v}}^{b}_{tb,odo} &= \begin{bmatrix} (v_L + v_R)/2 & 0 & 0 \end{bmatrix}^T + \vec{n}_{v,odo} \\ \tilde{\vec{\omega}}^{b}_{tb,odo} &= \begin{bmatrix} 0 & 0 & (v_L - v_R)/d \end{bmatrix}^T + \vec{n}_{\omega,odo} \end{aligned} \tag{4.19}$$

where, v_L / v_R are left / right wheel speeds and d is the axial separation between the wheels. Thus, a measurement of the velocity-error in the *t*-frame can be generated as

$$
\begin{aligned}
\delta\tilde{\vec{v}}_{tb}^{t} &= \hat{C}_b^t\,\tilde{\vec{v}}_{tb,odo}^{b} - \hat{\vec{v}}_{tb}^{t} \\
&= \hat{C}_b^t\left(\vec{v}_{tb}^{b} + \vec{n}_{v,odo}\right) - \left(\vec{v}_{tb}^{t} - \delta\vec{v}_{tb}^{t}\right) \\
&\simeq \left(I - \delta\Psi_{tb}^{t}\right)C_b^t\,\vec{v}_{tb}^{b} + \hat{C}_b^t\,\vec{n}_{v,odo} - \left(\vec{v}_{tb}^{t} - \delta\vec{v}_{tb}^{t}\right) \\
&\simeq \vec{v}_{tb}^{t} - \delta\Psi_{tb}^{t}\,\vec{v}_{tb}^{t} - \vec{v}_{tb}^{t} + \delta\vec{v}_{tb}^{t} + \hat{C}_b^t\,\vec{n}_{v,odo} \\
&\simeq \delta\vec{v}_{tb}^{t} + \hat{C}_b^t\,\vec{n}_{v,odo} - \delta\Psi_{tb}^{t}\left(\hat{\vec{v}}_{tb}^{t} + \delta\vec{v}_{tb}^{t}\right) \\
&\approx \delta\vec{v}_{tb}^{t} + \left[\hat{\vec{v}}_{tb}^{t}\times\right]\delta\vec{\psi}_{tb}^{t} + \hat{C}_b^t\,\vec{n}_{v,odo} \\
&= \left[\left[\hat{\vec{v}}_{tb}^{t}\times\right] \quad I_{3\times3} \quad 0_{3\times9}\right]\begin{bmatrix}\delta\vec{\psi}_{tb}^{t}\\ \delta\vec{v}_{tb}^{t}\\ \delta\vec{r}_{tb}^{t}\end{bmatrix} + \hat{C}_b^t\,\vec{n}_{v,odo}
\end{aligned}
\tag{4.20}
$$

Unfortunately, the angular velocity measurement obtained from the odometry ($\tilde{\vec{\omega}}_{tb,odo}^{b}$) cannot be used to provide an accurate measurement of PVA error. Alternatively, it can be coordinatized in the *t*-frame ($\hat{C}_b^t\,\tilde{\vec{\omega}}_{tb,odo}^{b}$) and combined with the gyro angular velocity measurement via a least-square or complementary filtering approach.

A Kalman filter provides a unified framework for fusing aiding sensors with the inertial-only PVA in error-space as each additional aiding sensor simply augments the measurement vector and associated measurement covariance matrix provided to the filter [16]. The resulting measurement (see (4.20)) update matrix H and the measurement covariance R are shown in (4.21).

$$
\begin{aligned}
H &= \left[\left[\hat{\vec{v}}_{tb}^{t}\times\right] \quad I_{3\times3} \quad 0_{3\times9}\right] \\
R &= E\left\{\hat{C}_b^t\,\vec{n}_{v,odo}\left(\hat{C}_b^t\,\vec{n}_{v,odo}\right)^T\right\} = \hat{C}_b^t\begin{bmatrix}\sigma^2_{\tilde{\vec{v}}_{tb,odo}^{b}} & 0_{3\times3} & 0_{3\times3}\\ 0_{3\times3} & \sigma^2_{\tilde{\vec{v}}_{tb,y}^{b}} & 0_{3\times3}\\ 0_{3\times3} & 0_{3\times3} & \sigma^2_{\tilde{\vec{v}}_{tb,z}^{b}}\end{bmatrix}\left(\hat{C}_b^t\right)^T
\end{aligned}
\tag{4.21}
$$

Depth Camera Aiding

The depth camera provides attitude measurements, as demonstrated in Chapter 3 - Extracting Attitude from Depth Camera Images. This aiding can be used to combat attitude drift due to gyroscope errors, which directly benefits the quality of velocity and position estimates.

Attitude error is captured as a DCM, as shown in (4.22).

$$\begin{aligned} \delta\tilde{C}^t_b &= \tilde{C}^t_{b,cam}\left[\tilde{C}^t_{b,imu}\right]^T \\ &= e^{\delta\tilde{\Psi}^t_{tb}} \\ &\simeq I_3 + \delta\tilde{\Psi}^t_{tb} \end{aligned} \tag{4.22}$$

However, attitude error in DCM representation does not lend itself to a meaningful state vector update. Instead, the measurement update model is formulated in an angle-axis format. First, as a skew-symmetric matrix form as shown in (4.23):

$$\begin{aligned} \delta\tilde{\Psi}^t_{tb} &\simeq \tilde{C}^t_{b,cam}\left[\tilde{C}^t_{b,imu}\right]^T - I_3 \\ &\simeq \left[(I_3 - \delta\Psi^t_{tb,cam})C^t_b\right]\left[(I_3 - \delta\Psi^t_{tb,imu})C^t_b\right]^T - I_3 \\ &\simeq \delta\Psi^t_{tb,imu} - \delta\Psi^t_{tb,cam} - \cancel{\delta\Psi^t_{tb,cam}\left(\delta\Psi^t_{tb,imu}\right)^T} \\ &\approx \delta\Psi^t_{tb,imu} - \delta\Psi^t_{tb,cam} \end{aligned} \tag{4.23}$$

Then as an angle-axis vector as shown below:

$$\begin{aligned} & sk\left[\delta\tilde{\vec{\psi}}^t_{tb}\right] = sk\left[\delta\vec{\psi}^t_{tb,imu}\right] - sk\left[\delta\vec{\psi}^t_{tb,cam}\right] \\ & \Rightarrow \delta\tilde{\vec{\psi}}^t_{tb} \simeq \delta\vec{\psi}^t_{tb} - \vec{\eta}_{\upsilon} \end{aligned} \tag{4.24}$$

Measurement uncertainty from the depth camera is parameterized in the following manner, where each diagonal element reflects the variance of each angle-axis representation element:

$$R_{\vec{\psi}} = E\left[\vec{\eta}_{\upsilon}\vec{\eta}_{\upsilon}^{\,T}\right] = \begin{bmatrix} \sigma_{\vec{\psi}_1}^2 & 0 & 0 \\ 0 & \sigma_{\vec{\psi}_2}^2 & 0 \\ 0 & 0 & \sigma_{\vec{\psi}_3}^2 \end{bmatrix} \tag{4.25}$$

The main advantage of employing the SNHT is that each measurement from the camera also returns its uncertainty. The SNHT, however, returns uncertainty in terms of Euler angles. Euler angle uncertainty can be transformed in angle-axis format via the following transformation [8]:

$$\vec{\omega}_{tb}^{\,b} = \begin{pmatrix} 1 & 0 & -\mathrm{s}_{\theta} \\ 0 & c_{\phi} & s_{\phi}c_{\theta} \\ 0 & -\mathrm{s}_{\phi} & c_{\phi}c_{\theta} \end{pmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Hence,

$$R_{\vec{\psi}} = J\, R_{Euler}\, J^T = J \begin{bmatrix} \sigma_{\phi}^2 & 0 & 0 \\ 0 & \sigma_{\theta}^2 & 0 \\ 0 & 0 & \sigma_{\psi}^2 \end{bmatrix} J^T \tag{4.26}$$

Attitude measurement uncertainty adjusts accordingly to feature accuracy. When genuine features are captured in the point cloud, SNHT measurement uncertainty remains low. However, when non-planar features appear in the point cloud, SNHT measurement uncertainty increases, necessitating outlier rejection.

Outlier rejection is accomplished within the Kalman filter algorithm. Regarding this specific application, floor features are assumed never to require outlier rejection. However, wall features vary in geometric quality and may contain walls not aligned to the tangential frame axes (*e.g.*, open doors). Outlier rejection solves this problem by computing the Mahalanobis distance between the current Kalman filter estimate of the yaw angle ψ_{KF} and the yaw

measurement from the depth camera ψ_{cam}. Kalman filter estimate uncertainty $\sigma_{\psi,KF}$ and measurement uncertainty $\sigma_{\psi,cam}$ are included to normalize the Mahalanobis distance.

$$d = \sqrt{\frac{(\psi_{cam} - \psi_{KF})^2}{(\sigma_{\psi,cam} + \sigma_{\psi,KF})^2}} \tag{4.27}$$

By normalizing the Mahalanobis distance, d, thresholding measurements for outlier rejection is greatly simplified [18]. In the example shown in Figure 22, ψ_{cam} measurements are either accepted or rejected if the Mahalanobis distance, d, which becomes a Chi-squared random variable with one degree of freedom, is below a value of 0.5.

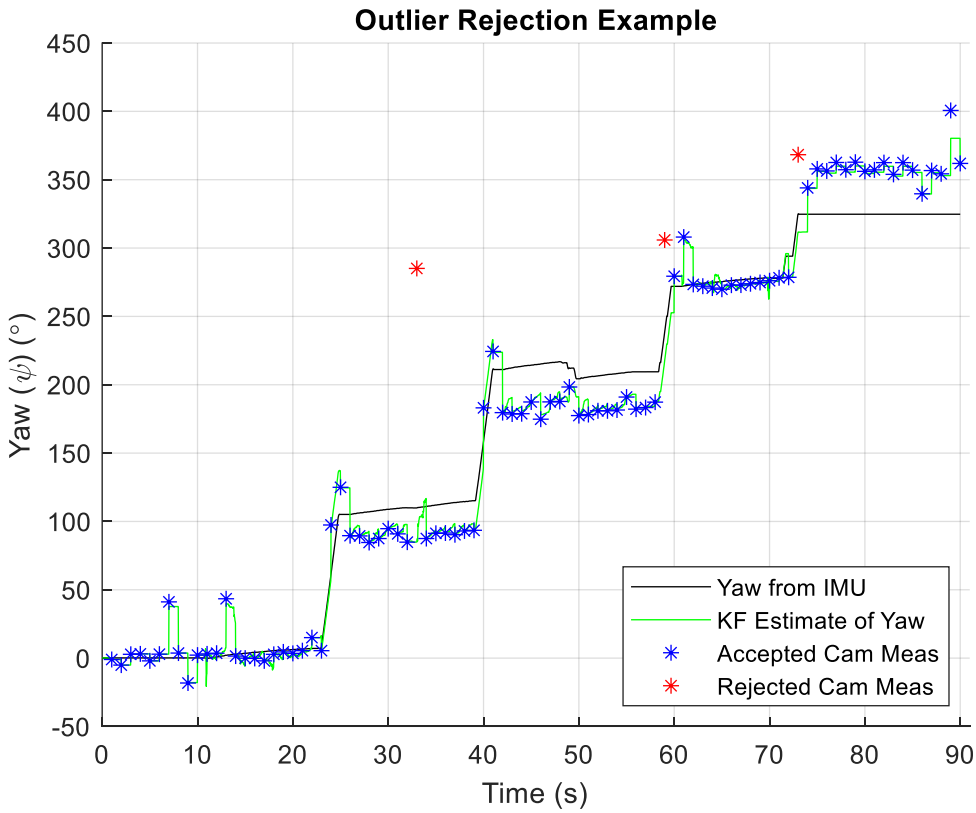

Figure 22 - Outlier Rejection performed on hardware data

The example in Figure 22 is derived from hardware data collected in an ideal environment shown in Figure 21. While most measurements are accepted, outliers are promptly rejected and ignored by the Kalman filter leaving overall estimates intact.

Aiding Sensor Configuration Comparison

While the benefit of odometry, the traditional aiding source, is well known, the performance of depth camera attitude aiding is undoubtedly not. Thus, while significant performance benefit from depth camera aiding is expected, just how much benefit should one expect? The following two chapters investigate this question in simulation and hardware implementation to determine the utility of depth camera aiding.

Chapter 5 – Simulation of PVA Estimation Improvement

A simulation was constructed to provide initial expectations of aiding sensor configuration performance provided from [8]. The simulation tests three aiding sensor configurations: IMU + Odo (Odometry), IMU + Kinect™, and IMU + Odo + Kinect™.

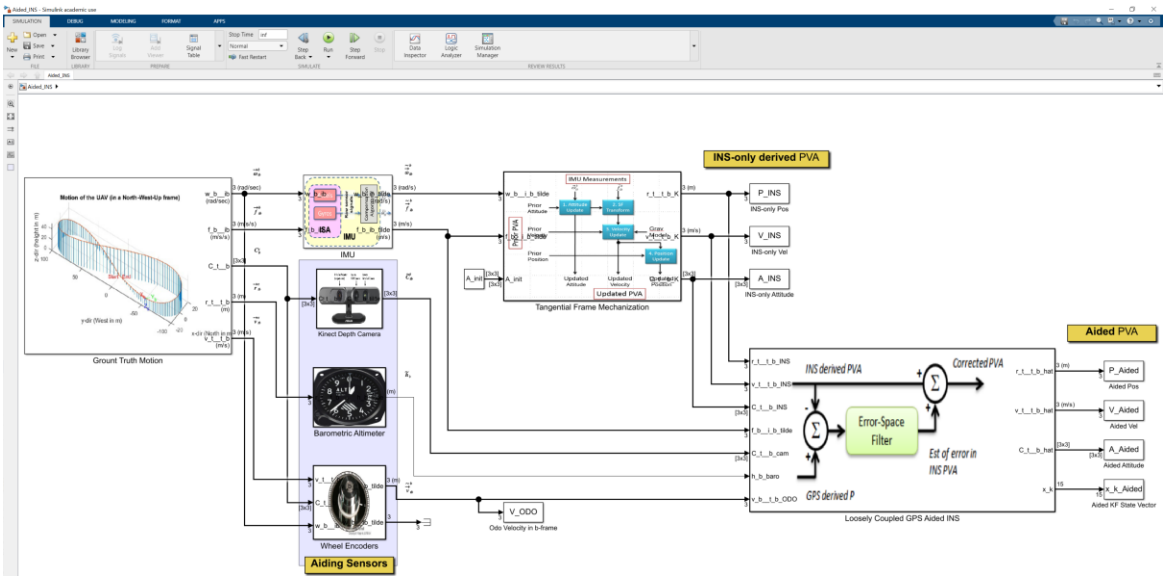

Figure 23 - Simulation Top-Level View

A motion profile of the Quanser Qbot 2™ was generated to produce "true" sensor measurement quantities for the IMU, odometry, and Kinect™ camera. Expected error quantities are added to each sensor measurement in accordance with datasheets and other noise characterization methods. This is then fed to the Kalman filter algorithm which predicts PVA error, which is used to return the overall best estimate of PVA for each aiding sensor configuration.

A comparison of the three aiding sensor configurations shows that Kinect™ camera aiding is effective in reducing overall attitude error and is relatively unaffected when odometry aiding is also included, as shown in Figure 24.

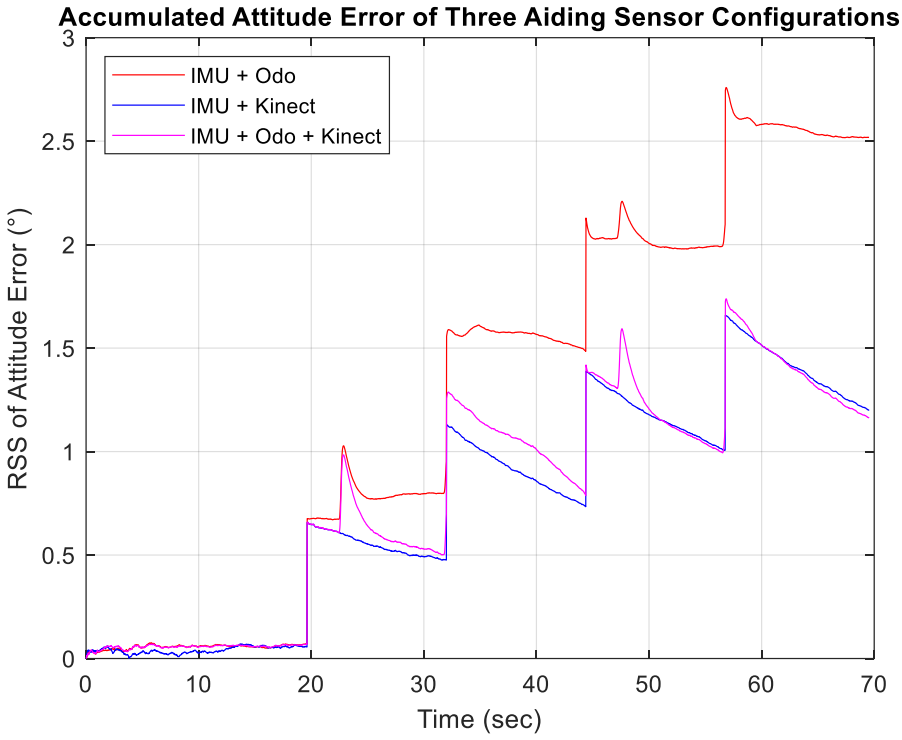

Figure 24 - Attitude Estimation Performance Comparison

With attitude error reduced via the Kinect™ camera, one would expect the position error to also decrease. This behavior is not reflected in simulation, which seems concerning at a first pass.

Figure 25 - Position Estimation Performance Comparison

The lack of position error reduction in Figure 25 is not due to Kinect™ camera aiding but rather overly optimistic odometry aiding. Odometry error is simulated by adding white noise to computed wheel velocities, which does not accurately capture real-world odometry errors such as wheel slippage and biases [17]. Without a realistic error model for simulated odometry measurements, comparing each aiding sensor configuration on real hardware becomes necessary.

Chapter 6 – Hardware Implementation

The Simple Box Test (SBT) serves as a proof-of-concept test to demonstrate increased PVA estimation performance when the Kinect™ camera is included as an aiding sensor. This test is designed to provide a close to ideal environment in-line with the assumptions made for the SNHT algorithm regarding flat floors and vertical walls at orthogonal orientations, as shown in Figure 26.

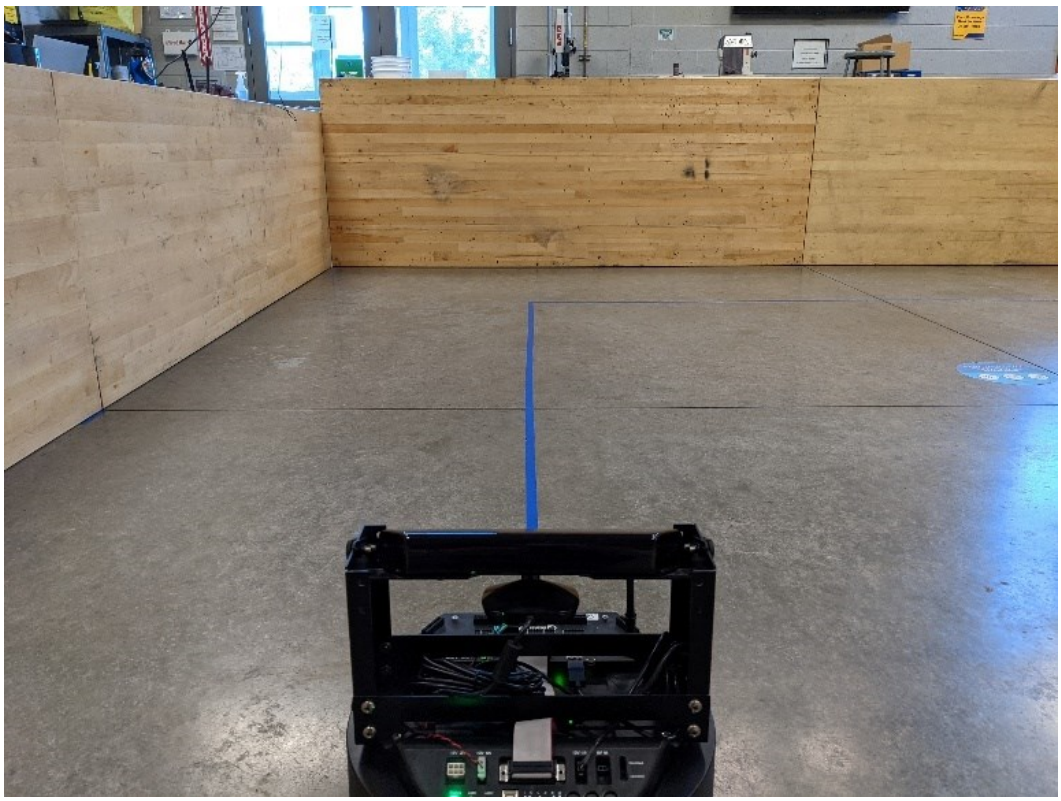

Figure 26 - Simple Box Test environment from the robot's perspective

Prior to constructing the environment for the SBT, shown in full view in Figure 21, the VectorNav VN-200™ was mounted to the center of the Quanser Qbot 2™ body. A 6-DOF transfer alignment was performed (see Figure 27) to align the IMU axes to the body-frame of the robot. This was accomplished by fixing the Quanser Qbot 2™ to an aluminum cage and

matching the local gravity vector passing through each face of the cage to each sensing axis of the IMU. A transfer alignment of the Kinect™ camera to the Quanser Qbot 2™ body was also performed to rotate captured point clouds appropriately to the Qbot 2™ body-frame.

Figure 27 - 6-DOF Transfer Alignment fixture

Once ready, the robot is driven wirelessly in the SBT course along blue tape on the floor (see Figure 21). The test begins by having the robot remain quiescent for ten seconds for initialization purposes. Once the blue tape path has been traversed, the robot is driven back to the initial position and orientation. The robot records IMU data and odometry data at 50 Hz and captures point clouds at 1 Hz. All data is saved and post-processed offline due to inadequate

computational resources onboard the Quanser Qbot 2™. The driver of the robot does their best to keep the robot on the blue tape path.

Serial port communication issues caused IMU measurements to lag behind odometry and Kinect™ camera measurements during data collection. To resolve this issue after data collection was complete, each motion event in the IMU, such as the straight-forward accelerations and turns, were manually aligned to the same motion events in the odometry data, reconciling the time alignment issue. While this solution is not ideal, it is an ethical solution for a proof-of-concept test.

Simple Box Test Post-Processing

Once all sensor data was collected and manually corrected for latency issues, post-processing began in two phases: point cloud processing and aiding sensor performance comparison. Point cloud processing consists of performing an SNHT search for each variety of the feature: a floor, a front wall, and a side wall. Each SNHT search is performed in accordance with Chapter 3 - Extracting Attitude from Depth Camera Images, returning yaw angle measurements and their uncertainties. These results are then saved and brought forth into the second phase of post-processing.

In the second phase, another simulation was built to emulate the navigation to be ideally performed onboard the Quanser Qbot 2™. Each variety of aiding sensor configuration tested in Chapter 4 – Aiding Sensor Configurations is also tested in post-processing. For configurations involving the Kinect™ camera, SNHT results are then emulated in "real time" to construct surface normal vectors, compute $\tilde{C}^{t}_{b,cam}$, and perform outlier rejection all before being processed by the

Kalman filter algorithm. This process is explained thoroughly in Chapter 3 - Extracting Attitude from Depth Camera Images.

Simple Box Test Results

The estimated path from each aiding sensor configuration is shown in Figure 28. Two of the sensor configurations, IMU Only and IMU + Kinect™, drift off well beyond the walls that make up the environment as expected. The two other aiding sensor configurations, IMU + Odo and IMU + Odo + Kinect™, remain somewhat bounded to the testing environment. The difference between the two estimated paths makes clear the performance benefit of including the Kinect™ camera in the aiding sensor package.

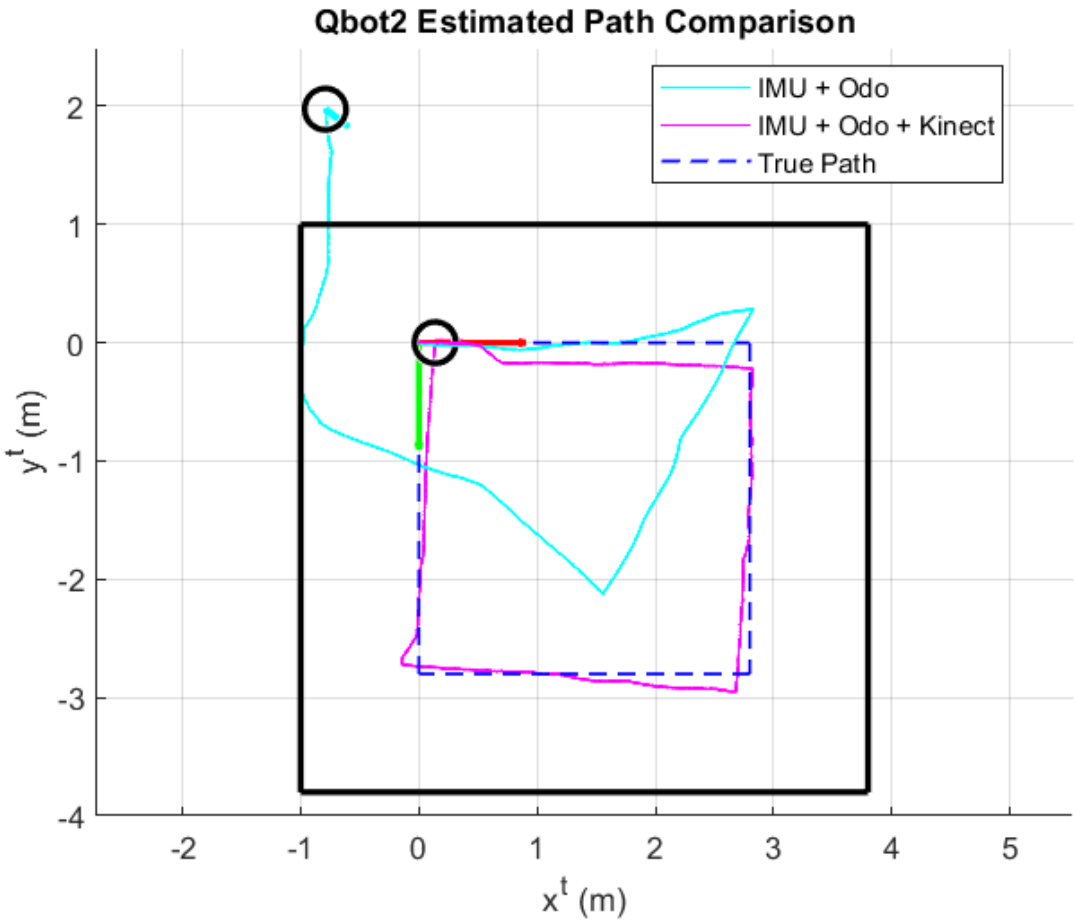

Figure 28 - A Comparison of Estimated Position Results

At the end of the SBT, the Qbot 2™ ends at the same position and attitude in which it began. The final position and attitude estimates from each aiding sensor comparison are shown in Figure 29, serving as a means of determining the final position and attitude error.

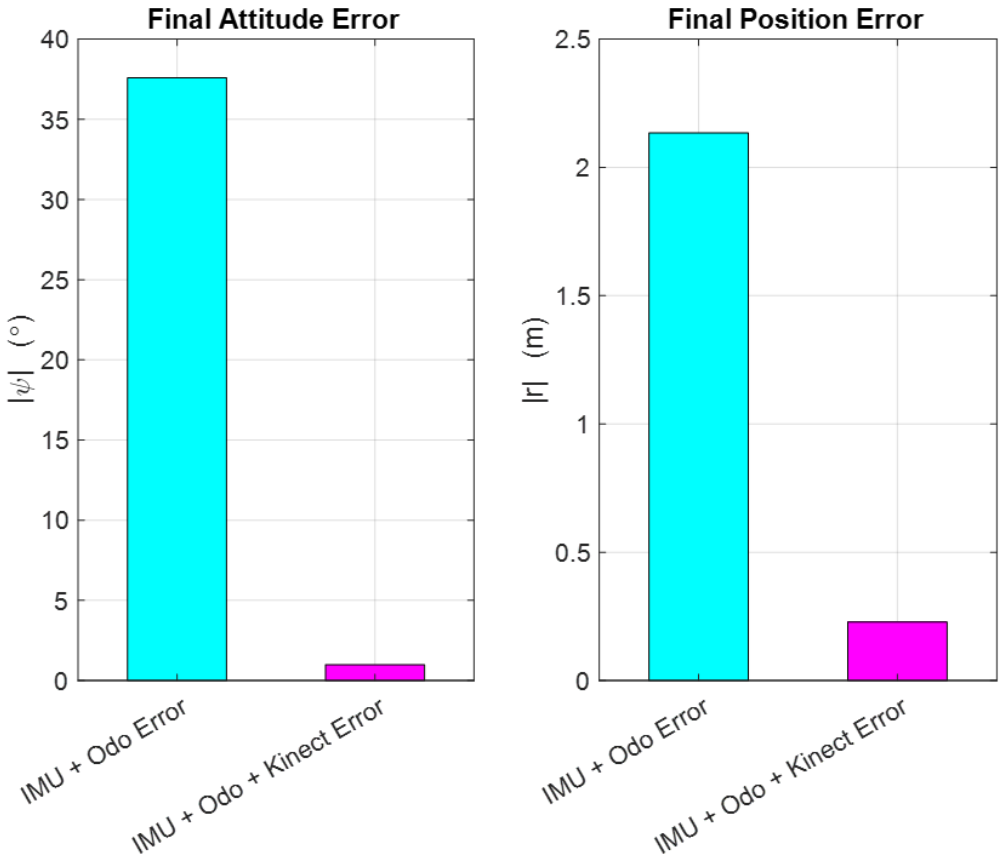

Figure 29 - Bar graph of final attitude and position error

The final position and attitude errors are reduced by approximately a factor of ten, indicating a profound performance benefit. This result affirms the benefit of depth camera aiding and prompts further investigation into depth camera aiding possibilities.

Chapter 7 – Conclusion

GNSS-based navigation is sometimes not feasible in certain situations such as indoor environments. This leads to adopting an IMU as the core of a navigation solution, although using an IMU alone results in significant error growth in computed PVA. PVA error accumulates mainly due to measurement errors from the gyroscope, rather than the accelerometer. This fact establishes the importance of mitigating gyroscope error. Employing a depth camera, such as the Kinect™ camera, and the SNHT algorithm, an opportunity arises to obtain accurate attitude information from the surrounding indoor environment. By incorporating this information into the full inertial navigation solution, attitude error is significantly reduced allowing for meaningful reconstruction of the robot's true path. The performance benefit of including depth camera aiding is abundantly clear in comparison to the traditional odometry aiding only approach.

For future work, finding and eliminating the source of the IMU data stream latency issues will allow for improved hardware implementation testing in non-ideal environments. Data collection in non-ideal environments beyond the SBT were initially planned in the development of this paper; however, the aforementioned data collection problems presented insurmountable challenges. This non-ideal environment included open doorways, trashcans, and other non-wall features. Generalizing the proposed approach to non-ideal environments will help solidify depth cameras as attitude aiding sources in dead-reckoning navigation settings.

References

[1] S. Y. Cho, M. S. Chae and K. H. Shin, "Reliability Analysis of the Integrated Navigation System Based on Real Trajectory and Calculation of Safety Margin Between Trains," in IEEE Access, vol. 9, pp. 32986-32996, 2021.

[2] David Olson, Stephen Bruder, Adam Watkins, Cleon Davis, " Depth Camera Aided Dead-Reckoning Localization of Autonomous Mobile Robots in Unstructured GNSS-Denied Environments," ICRLA004 2021: 15. International Conference on Robotics, Learning and Algorithms, Sydney, Australia, December 02-03, 2021.

[3] Paul D. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems 2e," Artech House, 2013.

[4] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali, "Indoor positioning and wayfinding systems: a survey," Human Centric Computing and Information Sciences, 10, 18 (2020).

[5] E. J. Alqahtani, F. H. Alshamrani, H. F. Syed and F. A. Alhaidari, "Survey on Algorithms and Techniques for Indoor Navigation Systems.," 2018 21st Saudi Computer Society National Computer Conference (NCC), 2018.

[6] https://www.sydney.edu.au/engineering/our-research/robotics-and-intelligent-systems/australian-centre-for-field-robotics.html

[7] P. Puricer and P. Kovar, "Technical Limitations of GNSS Receivers in Indoor Positioning," *2007 17th International Conference Radioelektronika*, 2007, pp. 1-5, doi: 10.1109/RADIOELEK.2007.371487.

[8] http://mercury.pr.erau.edu/~bruders/teaching/EE440/ee440.html

[9] A. M. Pinto, P. Costa, A. P. Moreira, L. F. Rocha, G. Veiga and E. Moreira, "Evaluation of Depth Sensors for Robotic Applications," *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015, pp. 139-143, doi: 10.1109/ICARSC.2015.24.

[10] Limberger, F. A. (2014). Real-Time Detection of Planar Regions in Unorganized Point Clouds (thesis). PPGC da UFRGS, Porto Alegre.

[11] Hough, Paul V. C. ."Machine Analysis of Bubble Chamber Pictures." (1959).

[12] Borrmann, D., Elseberg, J., Lingemann, K. *et al.* The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Res* **2,** 3 (2011). https://doi.org/10.1007/3DRes.02(2011)3

[13] T. Mallick, P. P. Das and A. K. Majumdar, "Characterizations of Noise in Kinect™ Depth Images: A Review," in *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1731-1740, June 2014, doi: 10.1109/JSEN.2014.2309987.

[14] Khoshelham, Kourosh, and Sander Oude Elberink. 2012. "Accuracy and Resolution of Kinect™ Depth Data for Indoor Mapping Applications" *Sensors* 12, no. 2: 1437-1454. https://doi.org/10.3390/s120201437

[15] Park, Jae-Han, Yong-Deuk Shin, Ji-Hun Bae, and Moon-Hong Baeg. 2012. "Spatial Uncertainty Model for Visual Features Using a Kinect™ Sensor" *Sensors* 12, no. 7: 8640-8662. https://doi.org/10.3390/s120708640

[16] Robert Grover Brown and Patrick Y. C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition," Wiley, February 2012.

[17] Borenstein, J., Everett, H. R., & Feng, L. (1996). *"Where am I?" -- Systems and Methods for Mobile Robot Positioning*. University of Michigan.

[18] Kim, M. G. (2000). "Multivariate outliers and decompositions of Mahalanobis distance". Communications in Statistics – Theory and Methods. 29 (7): 1511–1526