

Depth Camera Aided Dead-Reckoning Localization of Autonomous Mobile Robots in Unstructured Global Navigation Satellite System Denied Environments

David L. Olson, Stephen B. H. Bruder, Adam S. Watkins, Cleon E. Davis

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 precise IMU measurements with 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. However, 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 combats errors that arise due to gyroscope imperfections. This configuration for sensor fusion leads to a dramatic reduction of PVA error compared to traditional aiding sensor configurations. This paper provides the theoretical basis for the depth camera aiding sensor method, initial expectations of performance benefit via simulation, and hardware implementation thus verifying its veracity. Hardware implementation is performed on the Quanser Qbot 2™ mobile robot, with a VectorNav VN-200™ IMU and Kinect™ camera from Microsoft.

Keywords—Autonomous mobile robotics, dead reckoning, depth camera, inertial navigation, Kalman filtering, localization, sensor fusion.

I. INTRODUCTION AND BACKGROUND

THE proliferation of micro-electromechanical system (MEMS) IMUs over the past decade has become an enabling technology in the field of autonomous systems. 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]. 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 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. [3]. Unfortunately, some of these aiding sensors are not well suited to a mobile robotic platform operating indoors [4]. For example,

GNSS signals are unreliable indoors and magnetic fields generated by the robot's drive motors distort the magnetometer readings.

II. INERTIAL NAVIGATION BACKGROUND AND CHALLENGES

A single-axis gyroscope can be modeled as shown in Fig. 1. Extrapolating the model in Fig. 1, a triaxial IMU measures the inertial motion of the body coordinatized or resolved in the body frame (*b*-frame) as:

$$\begin{aligned}\tilde{\omega}_{ib}^b &= \bar{b}_g + (I + M_g)\bar{\omega}_{ib}^b + G_g \bar{f}_{ib}^b + \bar{w}_g \\ &= \bar{\omega}_{ib}^b + \Delta\bar{\omega}_{ib}^b \\ \tilde{f}_{ib}^b &= \bar{b}_a + (I + M_a)\bar{f}_{ib}^b + \bar{w}_a \\ &= \bar{f}_{ib}^b + \Delta\bar{f}_{ib}^b\end{aligned}\quad (1)$$

where $\bar{\omega}_{ib}^b$ and \bar{f}_{ib}^b are the “true” gyroscope and accelerometer vectors, M represents the scale factor and misalignment terms, G_g is the gyroscope g-sensitivity, \bar{w} denotes noise, and $\bar{b} = \bar{b}_{FB} + \bar{b}_{BS} + \bar{b}_{BI}$ denotes fixed, stability, and instability type bias terms.

A MATLAB application [5] was developed to accept sensor parameters in 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 Fig. 2.

Defining $\delta\bar{f}_{ib}^b = \bar{f}_{ib}^b - \hat{f}_{ib}^b$ where \hat{f}_{ib}^b is the post-calibration accelerometer estimate leads to:

$$\begin{aligned}\delta\bar{f}_{ib}^b &= (\bar{f}_{ib}^b - \hat{f}_{ib}^b) - \Delta\bar{f}_{ib}^b \\ &\triangleq \Delta\hat{f}_{ib}^b - \Delta\bar{f}_{ib}^b\end{aligned}\quad (2)$$

where $\Delta\bar{f}_{ib}^b$ represents the “true” sensor errors and $\Delta\hat{f}_{ib}^b$ is an estimate of such obtained via calibration. An equivalent expression can be obtained for the gyroscope.

David Olson is with Johns Hopkins University, United States (e-mail: divadnoslo@gmail.com).

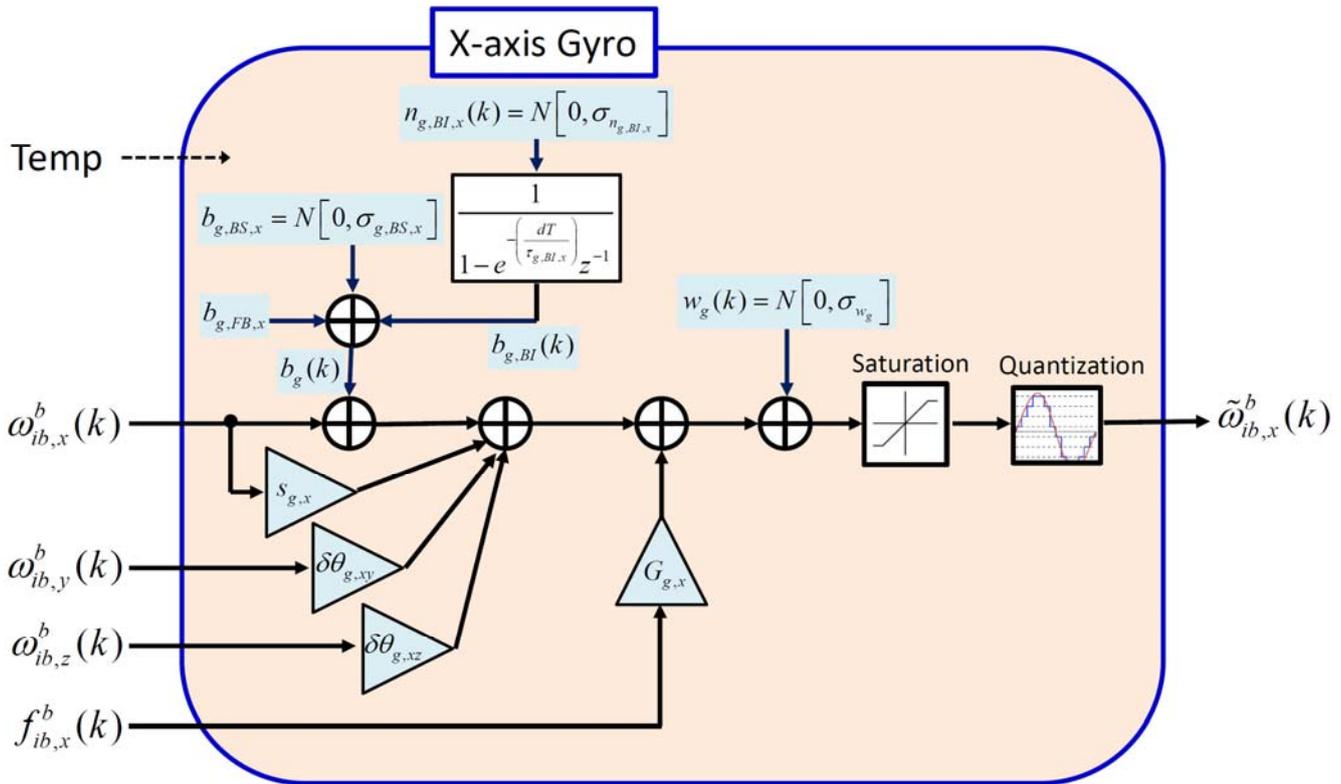


Fig. 1 Single-axis gyroscope data-flow model

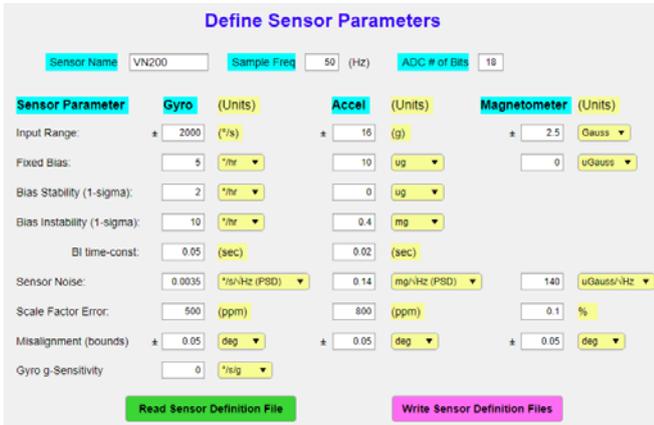


Fig. 2 Definition of parameters for a VectorNav VN-200™ IMU

Given the application domain, a relatively small region in the local vicinity, a locally level navigation coordinate frame often referred to as a tangential frame (t -frame) [2], is used to formulate the mechanization of the evolution of PVA from accelerometer and gyroscope measurements [5] as:

$$\begin{bmatrix} \dot{\vec{r}}_{ib}^t \\ \dot{\vec{v}}_{ib}^t \\ \dot{\hat{C}}_b^t \end{bmatrix} = \begin{bmatrix} \vec{v}_{ib}^t \\ C_b^t \vec{f}_{ib}^b + C_e^t \vec{g}_b^e - 2\Omega_{ie}^t \vec{v}_{ib}^t \\ C_b^t \Omega_{ib}^b \end{bmatrix} \quad (3)$$

where \vec{r}_{ib}^t is the position, \vec{v}_{ib}^t is the velocity, C_b^t conveys the attitude of the b -frame with respect to the t -frame, \vec{g}_b^t is the gravity vector (relatively constant), and $\Omega \triangleq [\hat{\omega} \times]$ is the skew-symmetric matrix version of an angular velocity vector.

Defining the position and velocity errors as truth minus estimate, e.g., $\delta \vec{r}_{ib}^t = \vec{r}_{ib}^t - \hat{\vec{r}}_{ib}^t$ and attitude errors as truth times estimate transposed $\Delta C_b^t = C_b^t [\hat{C}_b^t]^T = e^{\delta \Psi_{ib}^t}$, suggests that $\delta \Psi_{ib}^t \approx I - \hat{C}_b^t C_b^t = [\delta \vec{\psi}_{ib}^t \times]$. The error dynamics of this mechanization can be expressed in terms of the errors inherent to the IMU as [6]:

$$\begin{bmatrix} \delta \dot{\vec{\psi}}_{ib}^t \\ \delta \dot{\vec{v}}_{ib}^t \\ \delta \dot{\vec{r}}_{ib}^t \end{bmatrix} = \begin{bmatrix} -\Omega_{ie}^t & 0_{3 \times 3} & 0_{3 \times 3} \\ -[\hat{C}_b^t \hat{\vec{f}}_{ib}^b \times] & -2\Omega_{ie}^t & -\frac{2\hat{\vec{v}}_{ib}^t (\hat{\vec{r}}_{ib}^t)^T}{r_{eb}^e |\hat{\vec{r}}_{ib}^e|} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \vec{\psi}_{ib}^t \\ \delta \vec{v}_{ib}^t \\ \delta \vec{r}_{ib}^t \end{bmatrix} + \begin{bmatrix} \delta \vec{\omega}_{ib}^t \\ \delta \vec{f}_{ib}^t \\ \vec{0} \end{bmatrix} \quad (4)$$

The model presented in (4) can be modified to account for dynamics in the sensor error terms, such as a Gauss-Markov bias instability model. Many of the inertial sensor errors can be mitigated by careful calibration. However, some, such as gyroscope angle random walk (ARW) and bias instability (BI) cannot. Hence, these sensor error quantities provide intrinsic coordinates (see Fig. 3) for representing sensor performance

[5].

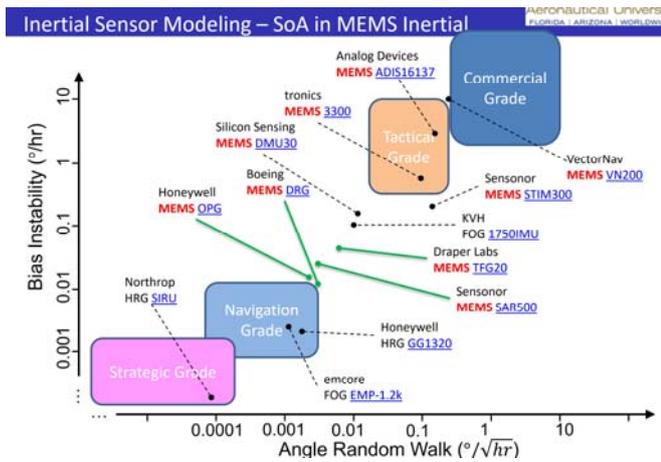


Fig. 3 A comparison of gyroscope performance

Examining the sources of PVA error growth leads to the consideration of gyroscope bias terms which give rise to a time-cubed (*i.e.*, $\propto t^3$) error term, in contrast to time-squared errors resulting from accelerometer bias terms, thus motivating the inclusion of an attitude aiding source as the highest priority.

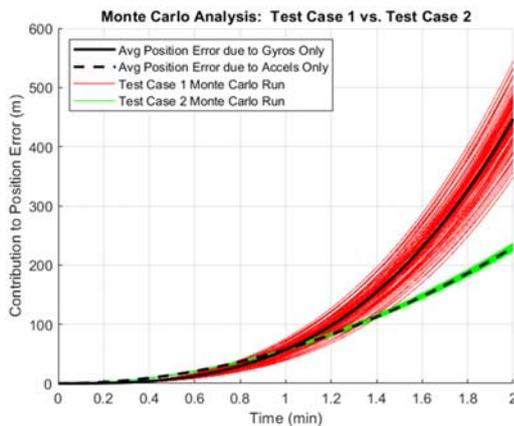


Fig. 4 Monte Carlo analysis of two test cases demonstrating the severe impact gyroscope errors have on position error growth

Fig. 4 presents a Monte Carlo analysis comparing two test cases of IMU error contributions to position error. We consider two test cases, each involving a calibrated VectorNav VN-200™ IMU at rest for two minutes collecting data. In test case 1, all accelerometer measurements are perfect while gyroscope measurement errors are present and 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 their own stochastic processes for time-varying error sources. The increase in position error growth due to gyroscope error sources ($\propto t^3$) versus

accelerometer error sources ($\propto t^2$) substantiates the need for including an attitude aiding source.

In summary, the selection of aiding sensors must be commensurate with their ability to mitigate dead reckoning error contributors. As such, our highest priority will be attitude aiding via a Kinetic™ camera followed by velocity aiding by way of odometry.

III. EXTRACTING ATTITUDE FROM DEPTH CAMERA IMAGES

A. Background and Motivation

Depth cameras such as the Kinect™ camera have become a popular sensor choice in the last decade, and their possible applications continue to grow [8]. Depth cameras return point clouds of their surrounding environment, such as the example in Fig. 5.

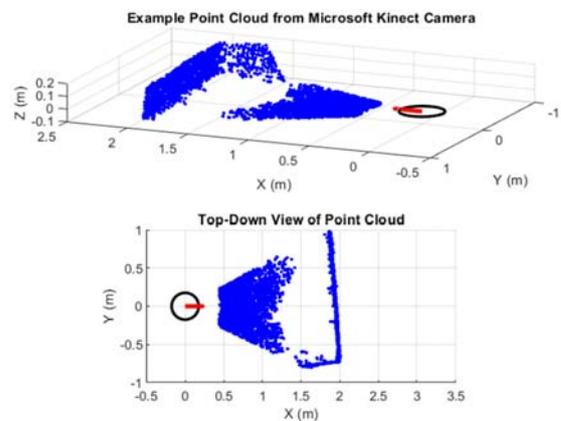


Fig. 5 Point Cloud from a Microsoft Kinect™ Camera onboard a Quanser Qbot 2™

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 [9], one of the most popular being the Hough transform [10]. 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 [11], but the core concept remains the same.

B. Hough Transform for Attitude Extraction

Viewing the point cloud from Fig. 5, we 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 which 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 2TM are of paramount interest. Due to the single axle and lack of suspension, the roll ϕ of the Qbot 2TM 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.

C. Surface Normal Hough Transformation Derivation

We consider the top-down 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 KinectTM camera sensor noise.

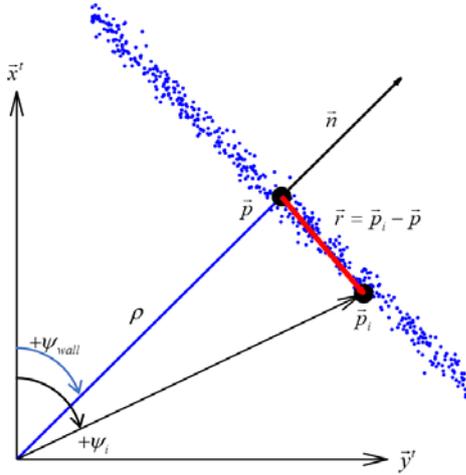


Fig. 6 Top-Down View of an Example Plane Feature Parameterized in the Proposed Hough Space.

To extract the surface normal vector \vec{n} from this feature, we 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 $\sin(\psi) \doteq s_\psi$.

$$\vec{n} = R_z(\psi) \cdot \vec{u}_x = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c_\psi \\ s_\psi \\ 0 \end{bmatrix} \quad (5)$$

Referring to Fig. 6, we 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 .

$$\vec{r} \cdot \vec{n} = (\vec{p}_i - \vec{p}) \cdot \vec{n} = (\vec{p}_i - \rho \vec{n}) \cdot \vec{n} = 0$$

$$\begin{bmatrix} x_i - \rho c_\psi \\ y_i - \rho s_\psi \\ 0 \end{bmatrix}^T \begin{bmatrix} c_\psi \\ s_\psi \\ 0 \end{bmatrix} = 0 \quad (6)$$

$$x_i c_\psi + y_i s_\psi = \rho$$

Equation (6) provides the transform of a Cartesian coordinate $\vec{p}_i = [x_i \ y_i \ 0]^T$ into the Hough space $[\psi \ \rho]$. 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 (7), respectively.

$$\begin{aligned} -x_i c_\psi + y_i s_\psi &= \rho \\ -x_i s_\theta + z_i c_\theta &= \rho \end{aligned} \quad (7)$$

D. 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. 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.

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 [11]; 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 I.

TABLE I
 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

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 Fig. 7.

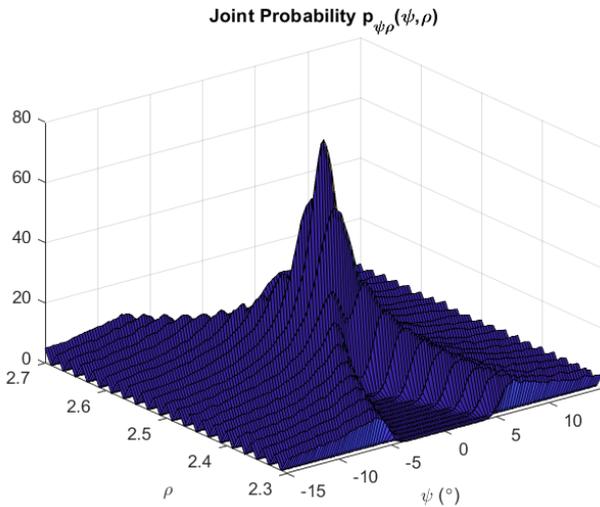


Fig. 7 Accumulator design yielding a joint pdf of ψ and ρ

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, $p(\psi | \rho = \rho_{ML})$, then provides the most likely value of the search angle, as well as the measurement uncertainty. An example of this is shown in Fig. 8. The fact that this approach to processing 3D depth data naturally provides the uncertainty of the measurement is a fundamental benefit of this technique.

E. Kinect™ Camera Noise Characteristics

Extensive work has gone into determining the noise characteristics of the Kinect™ camera [12]. Depth cameras acquire their measurements via different methods. The Kinect™ camera itself determines depth via triangulation [13]. A rigorous transformation from depth measurement uncertainty to Cartesian covariance ellipsoids is provided by [14]. 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.

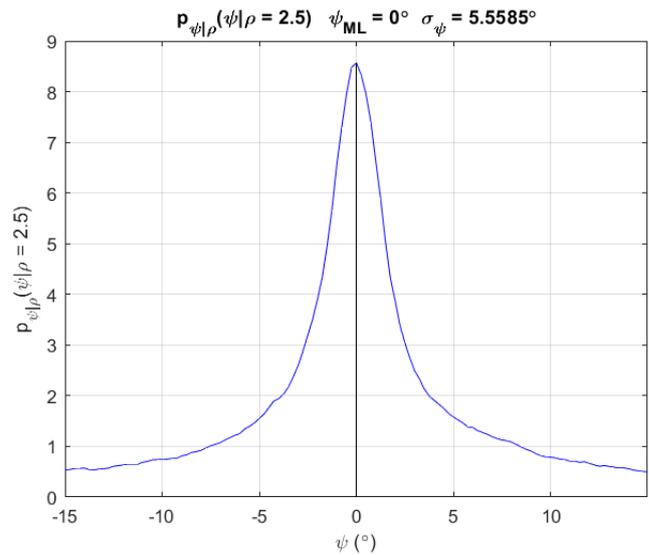


Fig. 8 Conditional probability of ψ for a given ρ , providing both a measurement and measurement uncertainty

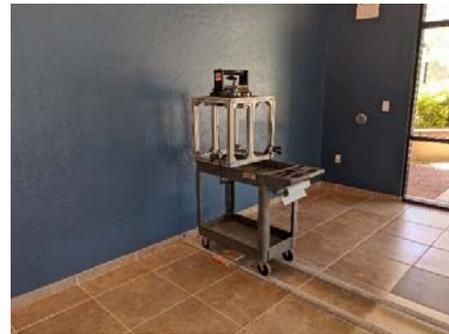


Fig. 9 Qbot 2™, Cart, Track, and Flat Wall for Noise Characterization

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 Fig. 10.

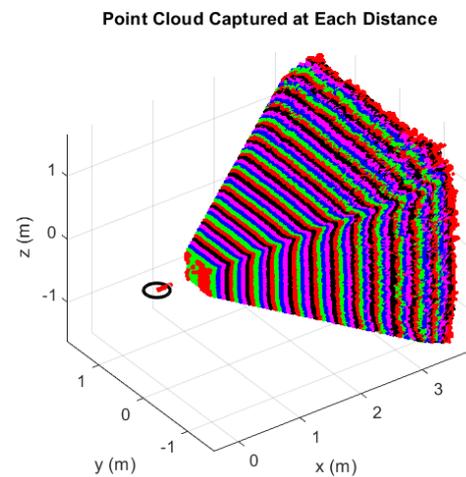


Fig. 10 Point clouds from the Kinect™ camera at various distances

Results from [12] were confirmed, in that, point cloud thickness grows quadratically with distance, as shown in Fig. 11.

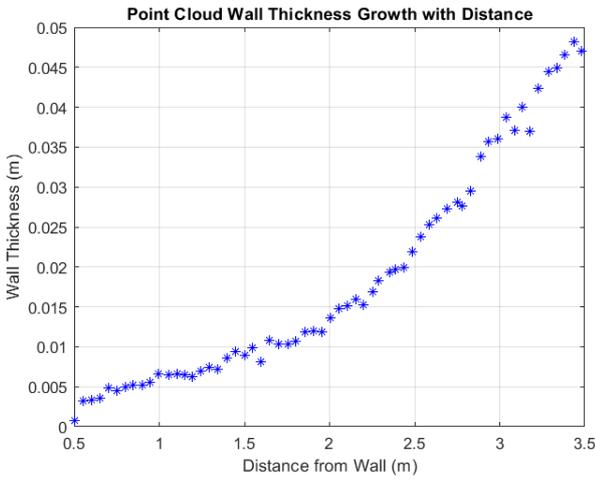


Fig. 11 Quadratic growth of point cloud wall thickness

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 Fig. 12.

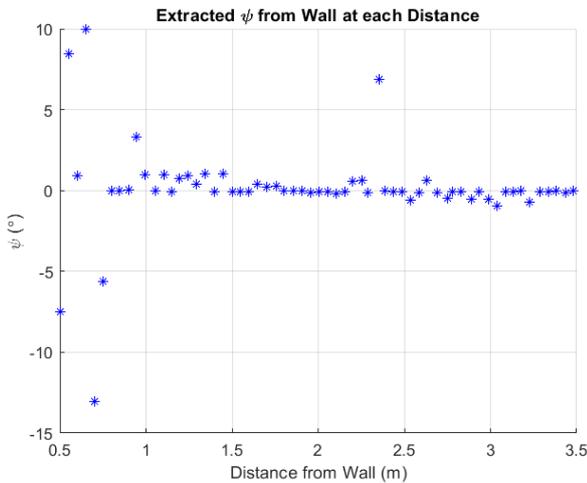


Fig. 12 Extracted ψ at each distance from the wall

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 (Fig. 13).

F. 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, the perpendicular walls will serve as the \bar{x}^t and \bar{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 $\hat{C}_{b, cam}^t$.

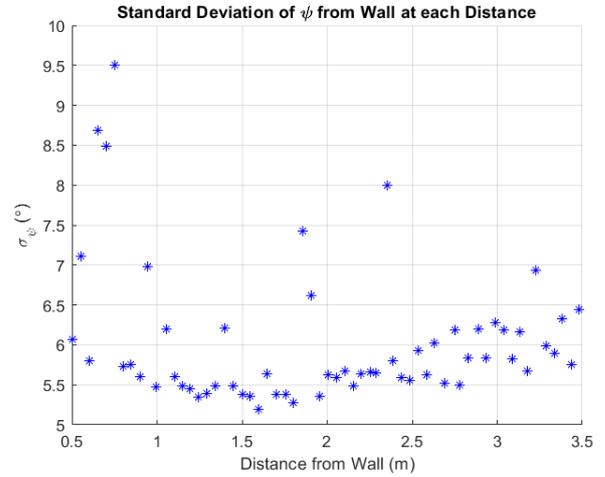


Fig. 13 Measurement uncertainty from each wall measurement



Fig. 14 Simple Box Test Environment

We assume that the mobile robot travels in a box with walls surrounding the travel path, as illustrated in Fig. 14.

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 \bar{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.

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

$$\bar{x}^{t+} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \bar{x}^{t-} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, y^{t+} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \bar{y}^{t-} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (8)$$

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

Equation (8) shows an example of computing the Mahalanobis distance for the likelihood of \tilde{x}_b^t (i.e., x-axis of the b -frame) being aligned with the positive t -frame axis \tilde{x}^{t+} . This process is repeated for the possible alignment of \tilde{y}_b^t also with \tilde{x}^{t+} , \tilde{x}^{t-} , \tilde{y}^{t+} , or \tilde{y}^{t-} . The smallest Mahalanobis distances computed determines the most likely axes to which the b -frame is aligned such that:

$$\begin{aligned} \text{If, } \min(d_{\tilde{x}^{t+}}, d_{\tilde{x}^{t-}}, d_{\tilde{y}^{t+}}, d_{\tilde{y}^{t-}}) = d_{\tilde{x}^{t+}} &\Rightarrow \hat{\tilde{x}}_b^t = \tilde{x}^{t+} \\ \min(d_{\tilde{x}^{t+}}, d_{\tilde{x}^{t-}}, d_{\tilde{y}^{t+}}, d_{\tilde{y}^{t-}}) = d_{\tilde{y}^{t+}} &\Rightarrow \hat{\tilde{y}}_b^t = \tilde{y}^{t+} \end{aligned} \quad (9)$$

Next, the attitude of the b -frame with respect to the t -frame is determined via the SNHT. Referring to Fig. 5, 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 $\begin{bmatrix} \theta_f^c & \psi_{fw}^c & \psi_{sw}^c \end{bmatrix}$ and their standard deviations $\begin{bmatrix} \sigma_f & \sigma_{fw} & \sigma_{sw} \end{bmatrix}$.

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

$$\begin{aligned} \theta_f^c &= -\theta_f^t, \quad \psi_{fw}^c = -\psi_{fw}^t, \quad \psi_{sw}^c = -\psi_{sw}^t \\ \vec{n}_f &= \begin{bmatrix} -s_{\theta_f^t} \\ 0 \\ c_{\theta_f^t} \end{bmatrix}, \quad \vec{n}_{fw} = \begin{bmatrix} c_{\psi_{fw}^t} \\ s_{\psi_{fw}^t} \\ 0 \end{bmatrix}, \quad \vec{n}_{sw} = \begin{bmatrix} -c_{\psi_{sw}^t} \\ s_{\psi_{sw}^t} \\ 0 \end{bmatrix} \end{aligned} \quad (10)$$

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 (11):

$$\begin{aligned} \begin{bmatrix} \tilde{x}_t^b \\ \tilde{y}_t^b \\ \tilde{z}_t^b \end{bmatrix} &= \begin{bmatrix} \vec{n}_{fw} \\ -(\vec{n}_{fw} \times \vec{n}_f) \\ \vec{n}_f \end{bmatrix}, \quad \sigma_{\psi_{fw}} \leq \sigma_{\psi_{sw}} \\ \begin{bmatrix} \tilde{x}_t^b \\ \tilde{y}_t^b \\ \tilde{z}_t^b \end{bmatrix} &= \begin{bmatrix} \vec{n}_{sw} \times \vec{n}_f \\ \vec{n}_{sw} \\ \vec{n}_f \end{bmatrix}, \quad \sigma_{\psi_{sw}} < \sigma_{\psi_{fw}} \end{aligned} \quad (11)$$

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

$$\begin{aligned} \tilde{C}_{t,cam}^b &= \begin{bmatrix} \tilde{x}_t^b \cdot \hat{\tilde{x}}_b^t & \tilde{y}_t^b \cdot \hat{\tilde{x}}_b^t & \tilde{z}_t^b \cdot \hat{\tilde{x}}_b^t \\ \tilde{x}_t^b \cdot \hat{\tilde{y}}_b^t & \tilde{y}_t^b \cdot \hat{\tilde{y}}_b^t & \tilde{z}_t^b \cdot \hat{\tilde{y}}_b^t \\ \tilde{x}_t^b \cdot \hat{\tilde{z}}_b^t & \tilde{y}_t^b \cdot \hat{\tilde{z}}_b^t & \tilde{z}_t^b \cdot \hat{\tilde{z}}_b^t \end{bmatrix} \\ \tilde{C}_{b,cam}^t &= [\tilde{C}_{t,cam}^b]^T \end{aligned} \quad (12)$$

IV. AIDING SENSOR IMPLEMENTATIONS

A. Estimating PVA Error

The synergy between the short-term precise inertial sensors and long-term accurate aiding sensors is compelling in determining PVA. The problem can be formulated to produce an estimate of PVA or an estimate of the error in PVA resulting from an inertial-only PVA solution (see (4)). The latter error-space approach is sometimes referred to as the "go-free" concept [7]. The dominant contributor to growth in the inertial-only PVA error is the gyroscope, but the accelerometer's impact should not be ignored. Some error terms, such as bias instability, possess dynamics that can augment the PVA error model of (4) [5].

B. Odometry Aiding

The odometer provides an accurate long-term measurement of linear and angular velocity in the body frame as:

$$\begin{aligned} \tilde{v}_{tb,odo}^b &= \left[(v_L + v_R) / 2 \quad 0 \quad 0 \right]^T + \vec{n}_{v,odo} \\ \tilde{\omega}_{tb,odo}^b &= \left[0 \quad 0 \quad (v_L - v_R) / d \right]^T + \vec{n}_{\omega,odo} \end{aligned} \quad (13)$$

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

$$\begin{aligned} \delta \tilde{v}_{tb}^t &= \hat{C}_{b,odo}^t \tilde{v}_{tb}^b - \hat{v}_{tb}^t \\ &= \hat{C}_b^t (\tilde{v}_{tb}^b + \vec{n}_{v,odo}) - (\tilde{v}_{tb}^t - \delta \tilde{v}_{tb}^t) \\ &\approx (I - \delta \Psi_{tb}^t) C_b^t \tilde{v}_{tb}^b + \hat{C}_b^t \vec{n}_{v,odo} - (\tilde{v}_{tb}^t - \delta \tilde{v}_{tb}^t) \\ &\approx \tilde{v}_{tb}^t - \delta \Psi_{tb}^t \tilde{v}_{tb}^t - \tilde{v}_{tb}^t + \delta \tilde{v}_{tb}^t + \hat{C}_b^t \vec{n}_{v,odo} \\ &\approx \delta \tilde{v}_{tb}^t + \hat{C}_b^t \vec{n}_{v,odo} - \delta \Psi_{tb}^t (\tilde{v}_{tb}^t + \delta \tilde{v}_{tb}^t) \\ &\approx \delta \tilde{v}_{tb}^t + \left[\hat{v}_{tb}^t \times \right] \delta \tilde{\Psi}_{tb}^t + \hat{C}_b^t \vec{n}_{v,odo} \end{aligned} \quad (14)$$

Unfortunately, the angular velocity measurement obtained from the odometry ($\tilde{\omega}_{tb,odo}^b$) cannot be used to provide an accurate measurement of PVA error. Alternatively, it can be coordinatized in the t -frame ($\hat{C}_b^t \tilde{\omega}_{tb,odo}^b$) and combined with the gyroscope 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 [5].

C. Depth Camera Aiding

The depth camera provides attitude measurements, as demonstrated in Section II. This aiding can be used to combat attitude drift due to gyroscope errors, which directly benefit the quality of velocity and position estimates.

Attitude error is captured as a DCM as shown in (15).

$$\begin{aligned} \delta \tilde{C}_b^t &= \tilde{C}_{b,cam}^t \left[\tilde{C}_{b,imu}^t \right]^T \\ &= e^{\delta \tilde{\Psi}_{tb}^t} \\ &\approx I_3 + \delta \tilde{\Psi}_{tb}^t \end{aligned} \quad (15)$$

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, from (15),

$$\begin{aligned} \delta \tilde{\Psi}_{tb}^t &\approx \tilde{C}_{b,cam}^t \left[\tilde{C}_{b,imu}^t \right]^T - I_3 \\ &\approx \left[(I_3 - \delta \Psi_{tb,cam}^t) C_b^t \right] \left[(I_3 - \delta \Psi_{tb,imu}^t) C_b^t \right]^T - I_3 \\ &\approx \delta \Psi_{tb,imu}^t - \delta \Psi_{tb,cam}^t \end{aligned}$$

Then, as an angle-axis vector:

$$\begin{aligned} sk \left[\delta \tilde{\psi}_{tb}^t \right] &= sk \left[\delta \tilde{\psi}_{tb,imu}^t \right] - sk \left[\delta \tilde{\psi}_{tb,cam}^t \right] \\ \Rightarrow \delta \tilde{\psi}_{tb}^t &\approx \delta \tilde{\psi}_{tb}^t - \tilde{\eta}_v \end{aligned} \quad (16)$$

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

$$R_{\tilde{\psi}} = E \left[\tilde{\eta}_v \tilde{\eta}_v^T \right] = \begin{bmatrix} \sigma_{\tilde{\psi}_1}^2 & 0 & 0 \\ 0 & \sigma_{\tilde{\psi}_2}^2 & 0 \\ 0 & 0 & \sigma_{\tilde{\psi}_3}^2 \end{bmatrix} \quad (17)$$

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 [5].

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

Hence,

$$R_{\tilde{\psi}} = J R_{Euler} J^T = J \begin{bmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix} J^T \quad (18)$$

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, providing an opportunity for 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. Outlier rejection solves this problem by computing the Mahalanobis distance between the current Kalman filter estimate of the Euler angle ψ_{KF} and the attitude measurement from the depth camera ψ_{cam} . Kalman filter estimate uncertainty $\sigma_{\psi,KF}$ and measurement uncertainty $\sigma_{\psi,cam}$ are added to normalize the Mahalanobis distance.

$$d = \sqrt{\frac{(\psi_{cam} - \psi_{KF})^2}{(\sigma_{\psi,cam} + \sigma_{\psi,KF})^2}} \quad (19)$$

By normalizing the Mahalanobis distance d , thresholding measurements for outlier rejection are greatly simplified. In the example shown in Fig. 15, ψ_{cam} measurements are either accepted or rejected if the Mahalanobis distance $d \leq 0.5$.

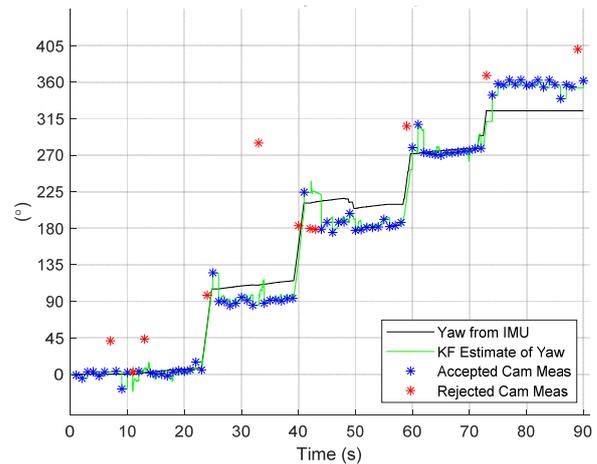


Fig. 15 Outlier Rejection Example

The example in Fig. 15 is derived from hardware data collected in the ideal environment shown in Fig. 14. While most measurements are accepted, outliers are promptly rejected and ignored by the Kalman filter leaving overall estimates intact.

V. AIDING SENSOR COMPARISON IN SIMULATION

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

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 Fig. 16.

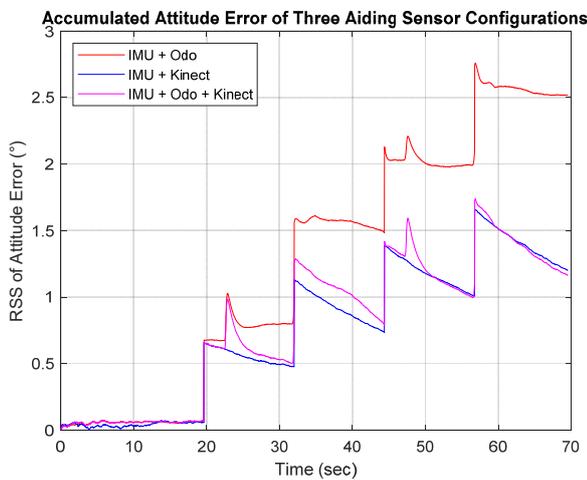


Fig. 16 Overall accumulated attitude error of three aiding sensor configurations in simulation

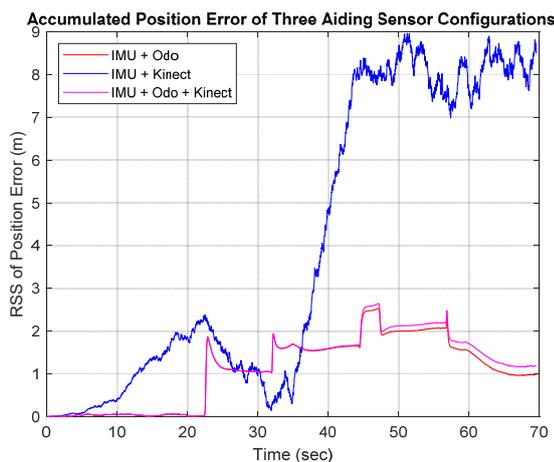


Fig. 17 Overall accumulated position error of three aiding sensor configurations in simulation

With attitude error reduced via the Kinect™ camera, one

would expect for position error to also decrease. This behavior is not reflected in simulation, which seems concerning at a first pass (see Fig. 17).

The lack of position error reduction is not due to Kinect™ camera aiding, but rather overly optimistic odometry aiding. The simulation adds white noise to simulated wheel velocities, which does not accurately capture real world odometry errors such as wheel slippage and imperfect wheels. Without a realistic error model for simulated odometry measurements, it becomes necessary to compare each aiding sensor configuration on real hardware.

VI. HARDWARE IMPLEMENTATION

A. Simple Box Test Preparation

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 an ideal environment in-line with the assumptions made for the SNHT algorithm regarding flat floors and vertical walls at orthogonal orientations, as shown in Fig. 18.

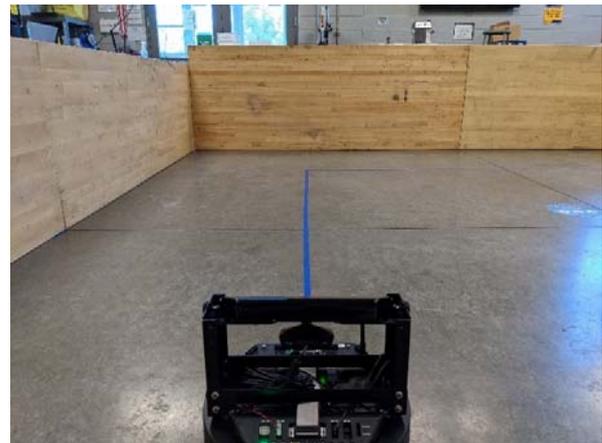


Fig. 18 SBT environment from the robot's perspective

Prior to constructing the environment for the SBT, shown in full view in Fig. 14, the VectorNav VN-200™ was mounted to the center of the Quanser Qbot 2™ body. A 6-DOF transfer alignment was performed to align the IMU axes to the body-frame of the robot. This was accomplished by fixing the Quanser Qbot 2™ to an extruded 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 accomplished to rotate captured point clouds appropriately to the Qbot 2™ body-frame.

Once ready, the robot is driven wirelessly in the SBT course along blue tape on the floor. The test begins by the having the robot remain quiescent for 10 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 are saved and post-processed

offline due to inadequate computational resources onboard the Quanser Qbot 2™. The drivers of the robot do their best to keep the robot on the blue tape path.



Fig. 19 6-DOF Transfer Alignment fixture

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.

B. SBT Post-Processing

Once all sensor data were 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 feature: a floor, a front wall, and a side wall. Each SNHT search is performed in accordance with Section II, returning Euler 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 Section IV 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 $\tilde{C}_{b,cam}^t$, and perform outlier rejection all before being processed by the Kalman filter algorithm. This process is explained thoroughly in Section II.

C. SBT Results

The estimated path from each aiding sensor configuration is shown in Fig. 20. Two of the aiding 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.

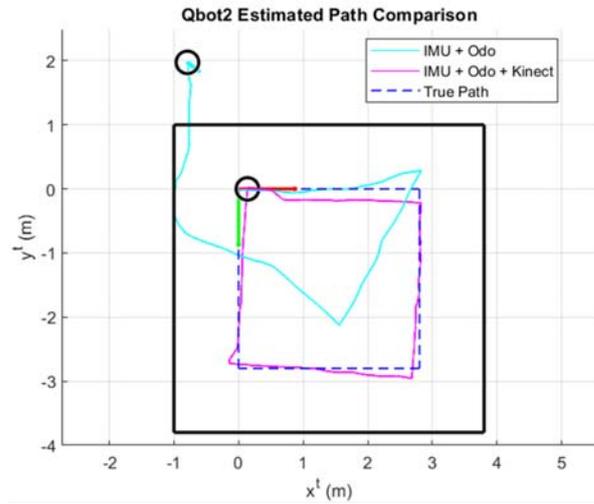


Fig. 20 A comparison of estimated positional results

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

TABLE II
 AIDING SENSOR CONFIGURATION COMPARISON

Quantity	IMU + Odo	IMU + Odo + Kinect™
$\hat{r}_{tb}^t = \begin{bmatrix} \hat{x}_{tb}^t \\ \hat{y}_{tb}^t \\ \hat{z}_{tb}^t \end{bmatrix} (m)$	$\begin{bmatrix} -0.792m \\ 1.975m \\ -0.168m \end{bmatrix}$	$\begin{bmatrix} 0.138m \\ 0.001m \\ -0.182m \end{bmatrix}$
$ \hat{r}_{tb}^t (m)$	2.134m	0.228m
$\hat{\psi}_{tb}^t = \begin{bmatrix} \hat{\phi}_{tb}^t \\ \hat{\theta}_{tb}^t \\ \hat{\psi}_{tb}^t \end{bmatrix} (^\circ)$	$\begin{bmatrix} 0.321^\circ \\ -1.342^\circ \\ -37.570^\circ \end{bmatrix}$	$\begin{bmatrix} 0.201^\circ \\ 0.702^\circ \\ -0.672^\circ \end{bmatrix}$
$ \hat{\psi}_{tb}^t (^\circ)$	37.595°	0.992°

Final position and attitude errors are reduced by approximately a factor of 10, indicating a profound performance benefit. This result affirms the benefit of depth camera aiding and prompts further investigation into depth camera aiding possibilities.

VII. 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 in computed PVA. PVA error accumulates mainly via measurement errors from the gyroscope, rather than the accelerometer. This fact establishes

the importance of combatting gyroscope error. Via 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 vividly 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. Non-ideal environments were initially planned in the development of this paper, however 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.

ACKNOWLEDGMENT

D. Olson would like to extend his appreciation to his co-authors for their mentorship and advice during the development of this paper. D. Olson also thanks Embry-Riddle Aeronautical University in Prescott AZ for lending their robotic platforms and testing environments during this project. D. Olson would also like to thank Johns Hopkins University for their financial support.

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] Paul D. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems 2e," Artech House, 2013.
- [3] 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).
- [4] 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.
- [5] <http://mercury.pr.erau.edu/~bruders/teaching/EE440/ee440.html>
- [6] T. Fauser, S. Bruder and A. El-Osery, "A comparison of inertial-based navigation algorithms for a low-cost indoor mobile robot," 2017 12th International Conference on Computer Science and Education (ICCSE), 2017.
- [7] Robert Grover Brown and Patrick Y. C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition," Wiley, February 2012.
- [8] 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.
- [9] Limberger, F. A. (2014). Real-Time Detection of Planar Regions in Unorganized Point Clouds (thesis). PPGC da UFRGS, Porto Alegre.
- [10] Hough, Paul V. C. "Machine Analysis of Bubble Chamber Pictures." (1959).
- [11] Bormann, 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](https://doi.org/10.1007/3DRes.02(2011)3)
- [12] 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.

- [13] 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>
- [14] 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>