

A New Point-Cloud-Based LiDAR/IMU Localization Method with Uncertainty Evaluation

Ali Hassani and Mathieu Joerger
Virginia Tech, Blacksburg, VA

Abstract

This paper describes the design, analysis, and experimental evaluation of a new spherical-grid-based (SGB) localization algorithm. This method combines a light detection and ranging (LiDAR)'s spherically-parametrized point cloud with measurements from an inertial measurement units (IMU) to estimate the position and orientation of a moving vehicle. It also quantifies navigation uncertainty. This grid-based method does not require feature extraction and data association, which are necessary steps in landmark-based localization. In addition, we developed an automated testbed to analyze the probabilistic performance of a landmark-based method and of the new spherical grid-based algorithm. The sample and analytical error distributions for both methods are evaluated in a lab environment.

1 Introduction

Ensuring the safety of automated driving systems (ADS) requires the quantification of navigation uncertainty. In this paper, we design a new spherical grid-based localization (SGBL) method, and we evaluate its error distribution, together with that of an existing landmark-based localization (LBL) algorithm. The new method combines light detection and ranging (LiDAR) point clouds (PC) with inertial measurement unit (IMU) data. It evaluates ADS pose corrections by comparing a measured PC to a map-based computed PC, which differs from other implementations in [1]. SGBL departs from our prior research on LBL in that it does not require feature extraction (FE) and fault-prone data association (DA) [2, 3]. We quantify the pose estimation uncertainty of both SGBL and LBL for three-dimensional LiDAR data collected on a moving platform roving in a cluttered laboratory environment.

This research focuses on LiDAR-based localization uncertainty quantification for future use in high-integrity navigation applications. One of the most widely implemented LiDAR-based pose estimator in robotic applications is Simultaneous Localization and Mapping (SLAM). SLAM, however, is insufficient for safety-critical navigation applications because SLAM errors drift over distance, and the loop-closures that are often used to limit this drift are trajectory-constraining [4, 5]. Instead, we assume that an a-priori map is available in the form of a detailed PC or of a parametric representation of geometric features (planes for walls, cylinders for lamp poles, etc.).

We can distinguish three categories of map-based LiDAR localization algorithms. First, LBL methods require two intermediary pre-estimator steps of FE and DA. The FE finds recognizable, viewpoint-invariant landmarks in the LiDAR PC, then DA assigns these measured landmarks to the mapped ones [6-10]. Measurable features (i.e. location, return light intensity, etc.) of associated landmarks can effectively be used for pose estimation, e.g., using an extended Kalman filter (EKF), but the risk of incorrect association can be high in cluttered environments [11]. A second category of LiDAR localization includes grid-based methods. These methods don't use FE and DA [12-14]. Occupancy grid maps (OGM) are a widely implemented approach. An OGM is a probabilistic, tessellated representation of the PC's spatial information [15]. In two dimensions, each OGM cell is assigned a value of the probability that it may be occupied [16, 17]. OGMs are used in path planning and collision avoidance, but there are no analytical approaches for determining the OGM-based pose estimation error distribution. OGMs typically use Cartesian coordinate representations. The third LiDAR localization category includes the iterative closest point (ICP) algorithm, which can be used to match three-dimensional (3D) PCs over successive time epochs [1,18]. Current LiDAR technology can provide hundreds of thousands of data point per 360-degree PC, which makes ICP's computational cost prohibitive.

In response, in this work, (1) we design an automated testbed to evaluate pose estimation uncertainty by collecting hours of data on a moving platform, (2) we use the testbed to experimentally quantify the performance of LBL, and (3) we design a new IMU/LiDAR grid-based pose estimation process by combining OGM and ICP. SGBL reduces computation load and facilitates processing of LiDAR spherical measurements. The proposed algorithm matches the

measured PC with a computed PC using the IMU-predicted LiDAR pose and a map of the surroundings. We quantify pose estimation uncertainty using the experimental testbed.

The organization of this paper is as follows. Section 2 presents the experimental testbed. In Section 3, we describe the LBL algorithm and evaluate FE errors using overbounding theory [19-21]. In Section 4, we design the SGBL method and outline an analytical estimation error modeling process. Section 5 is an experimental probabilistic performance analysis of the SGM estimation uncertainty. Section six presents our conclusions.

2 Automated experimental testbed

In this section, we describe the experimental testbed designed and built to quantify pose estimation errors. The testbed is shown in Figure 1. It includes a rover equipped with sensors. The rover moves on a figure-eight track. It can operate for hours unattended to collect LiDAR and IMU data over repeated trajectories. The rover’s Velodyne VLP-16 Puck LTE LiDAR and NovAtel IMU-IGM-A1 are stacked vertically in order to minimize the lever arm and misalignments between sensor frames. In addition, an infrared (IR) camera motion capture system provides reference-truth trajectory. Sixteen Optitrack Prime 13W cameras, provide sub-centimeter-level rover positioning by tracking retro-reflective markers fixed on the rover and LiDAR. IR cameras, LiDAR, and IMU data are time-tagged using a common computer clock. Figure 2 shows pictures of the sensors. Cardboard cylinders serve as landmarks and facilitate FE for LBL. The SGBL algorithm not only uses LiDAR returns from the landmarks but also from the unstructured background lab-environment.

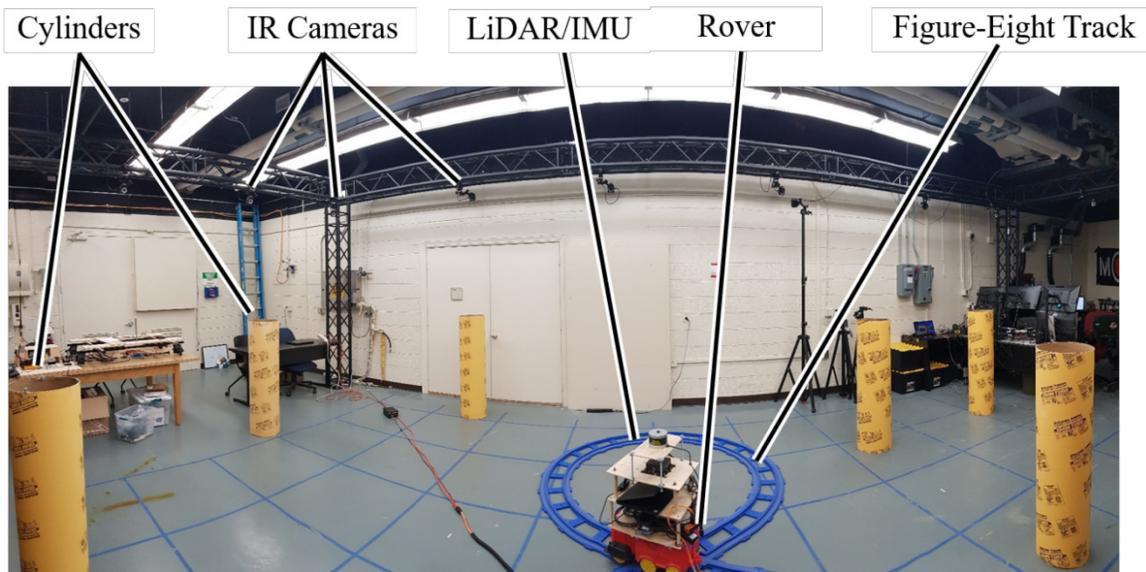


Figure 1. Testbed overview.

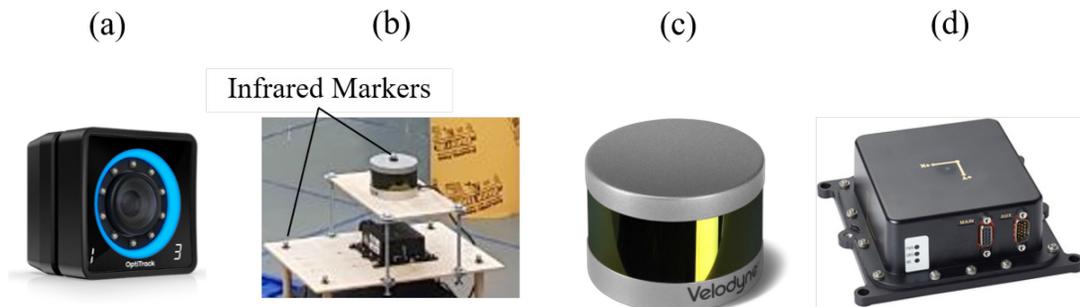


Figure 2. (a) IR camera; (b) IR markers on sensor platform; (c) LiDAR -VLP-16 Puck; (d) IMU-IGM-A1

3 Landmark-based localization method

The LBL method is based on a tightly integrated LiDAR/IMU estimator using an extended Kalman filter (EKF). This section provides an overview of the algorithm. More details can be found in [2, 3].

3.1 Linearized IMU measurements error model

The IMU state consists of ADS position, velocity, orientation, and IMU biases. The nonlinear continuous-time model of vehicle acceleration and sensor error can be found in [3]. The notation ‘ δ ’ indicates deviations of state parameters relative to reference values. We can write a continuous-time linearized state propagation model as [3]:

$$\delta\dot{\mathbf{x}} = \mathbf{F} \delta\mathbf{x} + \delta\mathbf{w} \quad (1)$$

$$\delta\mathbf{x} = [\delta\mathbf{x}_{ADS}^T \quad \delta\mathbf{v}_{ADS}^T \quad \delta\mathbf{e}_{ADS}^T \quad \delta\mathbf{b}_g^T \quad \delta\mathbf{b}_a^T] \quad (2)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{H2V} & \mathbf{0} & [{}^N\bar{\mathbf{f}}^1 \times] & \mathbf{0} & \mathbf{C}_B^N \\ \mathbf{0} & \mathbf{F}_{V2T} & -[{}^N\boldsymbol{\omega}^{IN} \times] & -\mathbf{C}_B^N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tau_g^{-1} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tau_a^{-1} \mathbf{I} \end{bmatrix}, \quad \delta\mathbf{w} = \begin{bmatrix} \mathbf{0} \\ \mathbf{C}_B^N (-\delta\mathbf{S}_a^B \bar{\mathbf{f}}^1 - \delta\mathbf{M}_a^B \bar{\mathbf{f}}^1 - \mathbf{v}_a) \\ -\mathbf{C}_B^N (-\delta\mathbf{S}_g^B \bar{\boldsymbol{\omega}}^{IB} - \delta\mathbf{M}_g^B \bar{\boldsymbol{\omega}}^{IB} - \mathbf{v}_g) \\ \bar{\mathbf{n}}_g \\ \bar{\mathbf{n}}_a \end{bmatrix} \quad (3)$$

where

$\delta\mathbf{x}_{ADS}$ is the vehicle position expressed in navigation frame N,

$\delta\mathbf{v}_{ADS}$ is the vehicle velocity with respect to earth expressed in frame N,

$\delta\mathbf{e}_{ADS}$ is the attitude of the vehicle with respect to earth expressed in body frame B,

$\delta\mathbf{b}_g, \delta\mathbf{b}_a$ are the gyroscope’s and accelerometer’s time-varying bias vectors in frame B respectively,

\mathbf{F}_{V2T} and \mathbf{F}_{H2V} are defined in Appendix A,

${}^N\boldsymbol{\omega}^{IN}$ is the angular velocity vector of the inertial frame I with respect to the frame N expressed in frame N [3],

${}^N\bar{\mathbf{f}}^1$ is the estimated specific force expressed in frame N,

\mathbf{C}_B^N is the rotation matrix from frame B to frame-N [22],

${}^B\bar{\mathbf{f}}^1$ is the measured specific force vector at the IMU axis center w.r.t. frame I expressed in frame B [22],

${}^B\bar{\boldsymbol{\omega}}^{IB}$ is the measured angular velocity vector of frame B w.r.t frame I expressed in frame B,

τ_g, τ_a are the gyro and accelerometer GMP time constants,

$\mathbf{S}_g, \mathbf{S}_a$ are the estimated gyroscope and accelerometer scale factors in frame B,

$\mathbf{M}_g, \mathbf{M}_a$ are the estimated gyroscope and accelerometer misalignment matrices in frame B,

$\mathbf{v}_g, \mathbf{v}_a$ are the gyroscope and accelerometer measurement white noise error components expressed in frame B,

$\mathbf{n}_g, \mathbf{n}_a$ are the gyroscope and accelerometer GMP time-uncorrelated driving noise vectors.

Considering the discrete-time expressions of the terms in equation (3) given in Appendix B, the discrete-time realization of equation (1) can be written as:

$$\delta\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \delta\mathbf{x}_k + \delta\mathbf{w}_k \quad (4)$$

where $\boldsymbol{\Phi}_k$ is the state transition matrix between time step ‘ k ’ and ‘ $k+1$ ’ [23].

3.2 Feature extraction process

A LiDAR provides a PC in the sensor's frame. FE finds consistently identifiable landmark features in the PC. Figure 3 shows an example LiDAR PC collected in the experimental testbed. The testbed includes easy-to-distinguish static vertical cylinders. The color code represents return light intensity measurements, with intensity decreasing from red to blue.

Figure 4 illustrates the three-step FE algorithm implemented to extract landmark information from LiDAR point clouds. FE aims at finding the center of the quasi-circular ellipses formed by the projection of vertical cylinders in the LiDAR's zero-elevation plane. The FE algorithm consists of the following steps. (1) *Segmentation*: we use the predicted vehicle pose to place the LiDAR on the landmark map and identify the data corresponding to cylinders. (2) *Model-Fitting*: we then project each point set onto the LiDAR zero-elevation plane and fit a circle through each point set. (3) *Feature Parameter Estimation*: the center of the best-fit circle is the extracted point-feature, whose relative location with respect to the LiDAR is captured with a range and bearing angle measurements.

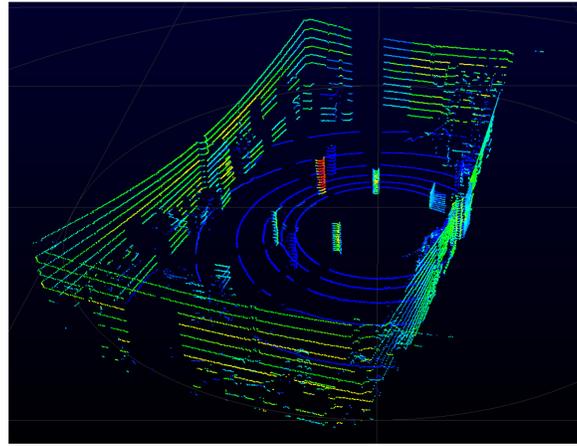


Figure 3. LiDAR Point Cloud Showing Return-Light Intensity (color-coded from blue to red, from low intensity to high intensity).

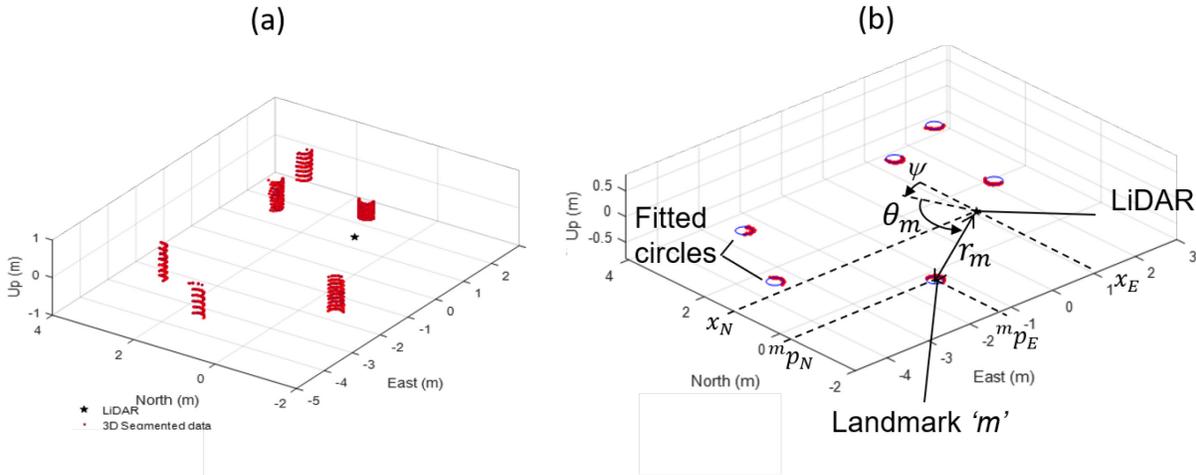


Figure 4. (a) 3D Segmentation of LiDAR Point cloud - (b) Circle Fitting and Point-Feature Measurement Extraction

In the following derivation, we express the extracted point-feature range and bearing angle measurement equations. First, let n_l be the number of extracted landmarks. We define r_m and θ_m as the range and bearing angle measurements in LiDAR frame for landmark 'm', for m ranging from 1 to n_l . In the landmark map's navigation frame, the horizontal position of the cylinder's center is time-invariant. Its Cartesian East and North coordinates are noted ${}^m p_E$ and ${}^m p_N$ for landmark m . , parameters in our IMU/LiDAR state space realization include the ADS position

\mathbf{x}_{ADS} and orientation \mathbf{e}_{ADS} in navigation frame (deviations in \mathbf{x}_{ADS} and \mathbf{e}_{ADS} appear in the state error equation (2)) are expressed as:

$$\bar{\mathbf{x}}_{ADS} = [x_E \quad x_N \quad x_U]^T \quad (5)$$

$$\bar{\mathbf{e}}_{ADS} = [\phi \quad \gamma \quad \psi] \quad (6)$$

The nonlinear LiDAR range and angular measurements can be written respectively as:

$$r_m = \sqrt{({}^m p_E - x_E)^2 + ({}^m p_N - x_N)^2} + v_r \quad (7)$$

$$\theta_m = \tan^{-1} \left(\frac{{}^m p_N - x_N}{{}^m p_E - x_E} \right) - \psi + v_\theta \quad (8)$$

where v_r and v_θ are random feature measurement errors.

The distributions of v_r and v_θ are not Gaussian, but can be overbounded in the CDF-sense [19, 20] by zero-mean normal distributions. Section 3.3 summarizes the overbounding theory and its impact on pose estimation error distribution. We use overbounding distributions to model the point-feature measurement error distributions.

Now, we can stack range and bearing measurements for all visible landmarks to obtain the following $2n_l \times 1$ nonlinear LiDAR measurement equations:

$$\hat{\mathbf{z}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (9)$$

$$\hat{\mathbf{z}}_k = [r_1 \quad \cdots \quad r_{n_l} \quad \theta_1 \quad \cdots \quad \theta_{n_l}]^T \quad (10)$$

$$\mathbf{v}_k = [v_{r_1} \quad \cdots \quad v_{r_{n_l}} \quad v_{\theta_1} \quad \cdots \quad v_{\theta_{n_l}}]^T \quad (11)$$

where

\mathbf{x}_k is the state vector whose error vector is defined in equation (2),

\mathbf{v}_k is the $2n_l \times 1$ feature measurement error vector modeled as a vector of normally distributed random variables as $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_k)$.

We linearize equation (9) about our best prediction of the vehicle and landmark positions, resulting in the linearized range and angular measurement and measurement error vectors, respectively denoted as $\delta\mathbf{R}$, $\delta\boldsymbol{\theta}$ and \mathbf{v}_r and \mathbf{v}_θ . The linearized LiDAR measurement equation can be written as:

$$\begin{bmatrix} \delta\mathbf{R} \\ \delta\boldsymbol{\theta} \end{bmatrix}_k = \begin{bmatrix} \mathbf{F}_{r,x} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{\theta,x} & \mathbf{0} & -\mathbf{F}_{\theta,e} & \mathbf{0} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \delta\mathbf{x}_{ADS} \\ \delta\mathbf{v}_{ADS} \\ \delta\mathbf{e}_{ADS} \\ \delta\mathbf{b}_g \\ \delta\mathbf{b}_a \end{bmatrix}_k + \begin{bmatrix} \mathbf{v}_r \\ \mathbf{v}_\theta \end{bmatrix}_k \quad (12)$$

where the coefficient matrices $\mathbf{F}_{r,x}$, $\mathbf{F}_{\theta,x}$ and $\mathbf{F}_{\theta,e}$ are determined using the state prediction vector and assuming correct association as described in Appendix A. More information about DA can be found in [11].

3.3 Measurement error modeling using overbounding theory

This section describes a robust method to derive probabilistic models of the extracted feature measurements. This method is based on overbounding theory in [19-21]. We collected LiDAR point cloud data for 4250 of sensor-to-landmark geometries, processed them using our feature extractor, and stored the estimated point-feature range and bearing angle measurements.

Figure 5 shows the range and bearing angle measurement error CDF on quantile-to-quantile plots. The plot's x-axis scales with theoretical standard normal distribution quantiles. The y-axis scales with the sample measurement

error distribution quantiles. If the empirical measurement error distribution were a normal distribution, the sample points would lie on a straight line with a slope equal to the sample standard deviation, with y-intercept equal to the sample mean. Figure 5 shows that the core of the distribution behaves like a normal distribution within $\pm 2\sigma$ on the x-axis, i.e., 95% of the time. But the sample distributions have wide tails.

Overbounding theory is used in aviation navigation to model non-Gaussian sample distributions [19-21]. The black lines in Figure 5 are overbounding Gaussian functions which account for errors occurring 99.5% of the time (the 4250 samples limit the reach of the overbound). The bounding standard deviations are 0.12 m for the range measurement error (versus 0.03 m for the sample standard deviation), and 2 degrees for the bearing angle measurement error (versus 1 deg for the sample standard deviation). Thus, Gaussian overbounds are conservative as compared to sample measurement error distributions, which will impact the pose estimation error distribution in the next section.

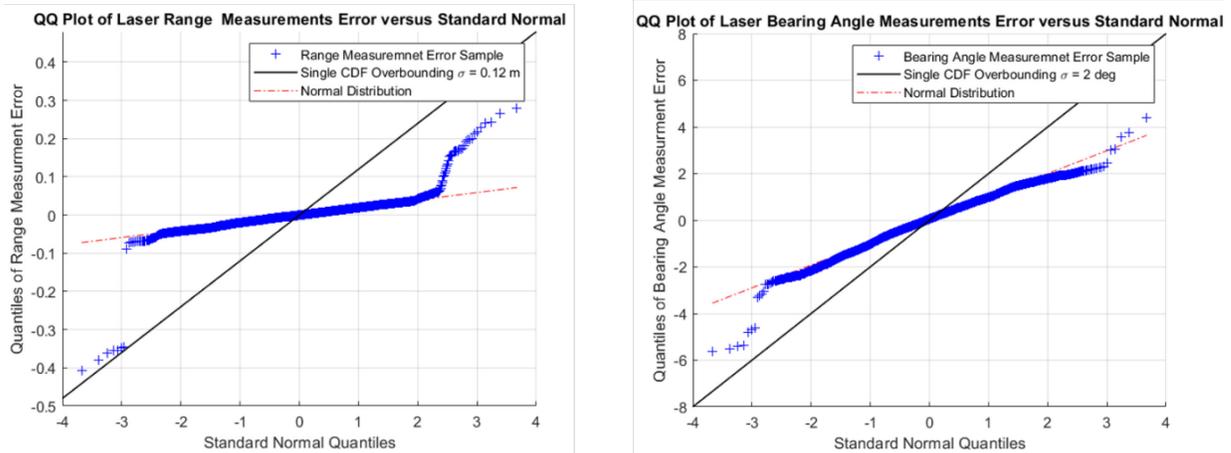


Figure 5. Range and Bearing Angle Measurement Error Distribution and Gaussian Overbounding Model

3.4 LBL pose estimation performance of 100 laps of trajectory

In this section, we describe the probabilistic pose estimation performance of the LBL method. The automated experimental testbed in Section 2 enables determination of estimation error distributions in the position domain, for comparison with analytical EKF error distributions, which we have not found in the literature. Figure 6 shows the estimated trajectory with a black solid line, the true trajectory with a black dashed line, the analytical 1σ covariance ellipses in blue, and the 1σ sample covariance ellipses in red. These covariance ellipses are inflated by a factor 5 for better visualization. The zoomed-in window shows that the sample and analytical covariance ellipses differ both in size and orientation. Potential sources of discrepancy include nonlinearities in process and measurement equations, inaccuracies in experimental data collection, and the conservativeness of the Gaussian FE error model in Section 3.3.

Figure 7 display gray lines representing sample error curves, defined as estimated-minus-true pose, over 100 laps. These curves show both the IMU-based pose prediction errors and the impact of LiDAR LBL updates. The upper chart in Figure 7(a) presents cross track errors which often are of primary concern in path-following safety evaluation. Below that, the second, third, and fourth charts show errors along the x-direction, along the y-direction and in heading. The 1σ analytical and sample covariance envelopes are respectively shown in blue and red. Figure 7(b) focuses on the time-interval highlighted in Figure 7(a). Analytical and sample covariance envelopes capture the IMU drift during state propagation, which is limited by LiDAR measurement updates.

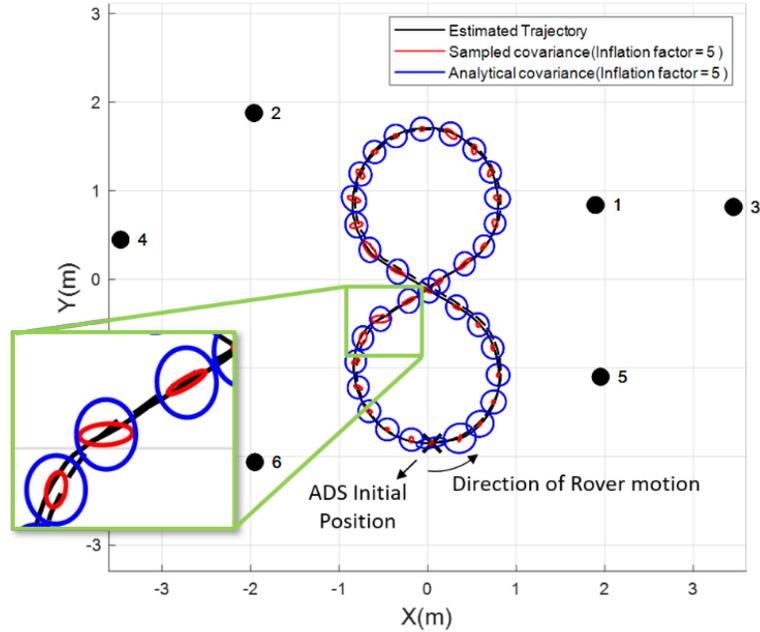


Figure 6. Landmark-Based Analytical and Sample Covariance Ellipses over 100 Laps

4 Spherical grid-based localization

In this section we introduce a new localization algorithm called spherical grid-based localization (SGBL). The goals with this SBGL method are: a) to utilize all LiDAR cloud points rather than the few corresponding to extracted and associated landmarks, b) to eliminate the errors caused by extraction and association, and c) to provide a computationally efficient approach to quantify navigation uncertainty.

4.1 Spherical grid-based estimator

The estimator part of SGBL algorithm includes two main procedures: a) a spherical gridding algorithm and b) a pose candidate search algorithm.

4.1.1 Spherical gridding algorithm

We use a sensor-centric spherical gridding approach, using range measurements in azimuth-elevation bins, which facilitates processing of spherical LiDAR data as compared to using Cartesian grids. Using a sensor-centric representation automatically accounts for sensor-location-dependent occlusions in the mapped data. In SGML, IMU-based sensor pose prediction on the map is used to compute a model LiDAR PC in sensor frame. In this work, the map comes in the form of a point cloud, but other geometric or occupancy grid maps could be used with a ray-tracing function to achieve the same result.

The spherical grid is made of azimuth-elevation bins at regular angular intervals called grid cells. Figures 8(a) and (b) show examples of 2D and 3D spherical gridding algorithms. For a better visualization purposes, the 3D grid representation only includes the data measured in the black dashed box in Figure 8(a).

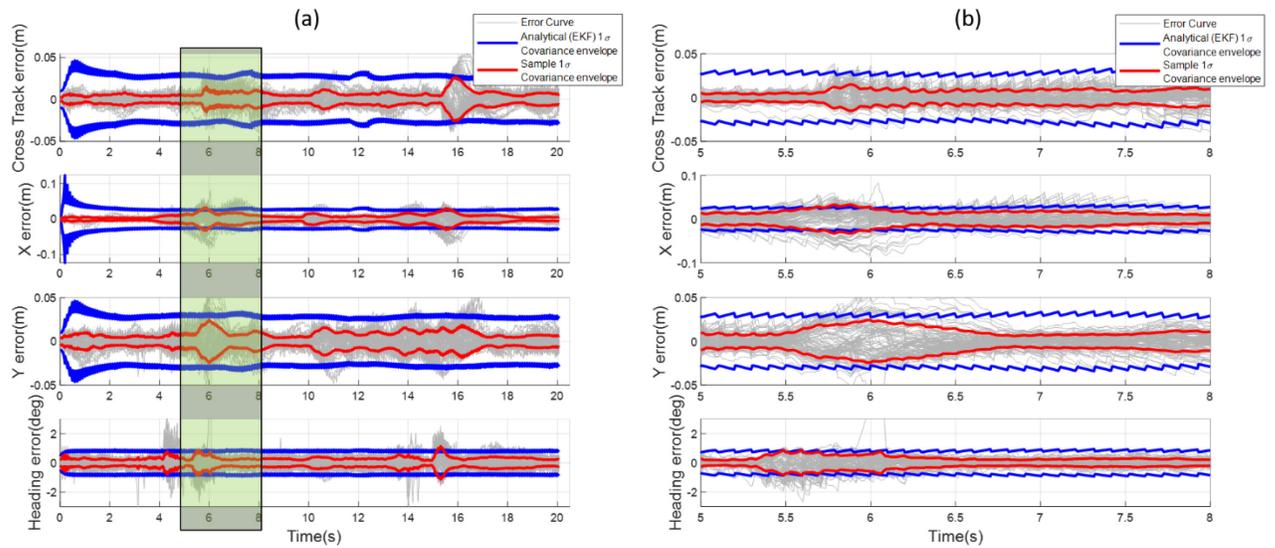


Figure 7. (a) Error Curves and Covariance Envelopes for Vehicle State Estimates - (b) IMU Drift Between Two LiDAR Measurements

In this first SGBL implementation, the shortest-range points, closest to the sensor in each azimuth-elevation bin are selected. (Additional processing could be performed, but is beyond the scope of this paper.) The selection of a single point per grid cell makes SGBL computationally more efficient than ICP. Point features in corresponding grid cells are matched, unless they are obvious outliers. Figure 9 is a zoomed-in version of the area inside of the green box in Figure 8(a). A selection process causes some grid cells to be empty, which does not appear to be an issue because the LiDAR provides plenty of redundant data.

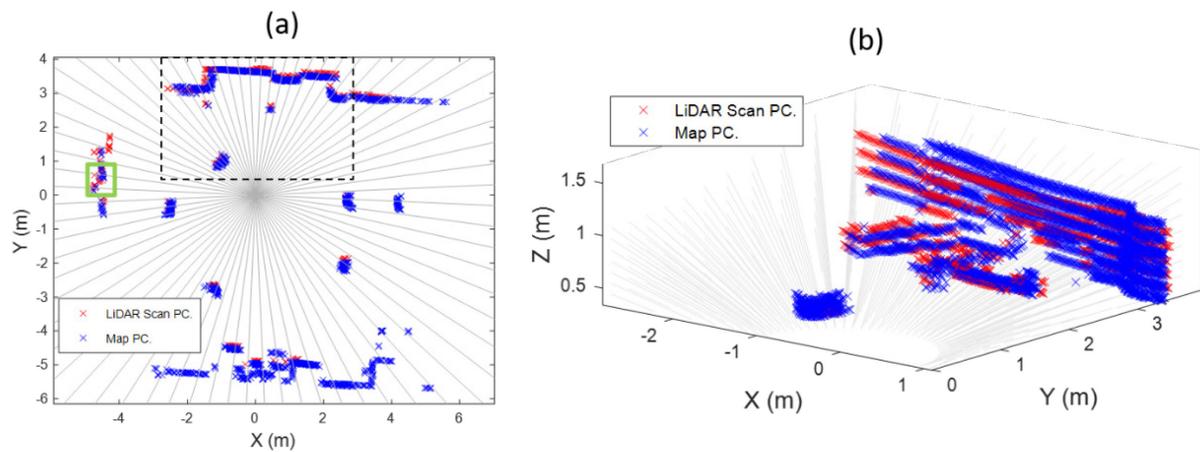


Figure 8. Spherical Gridding (a) 2D Spherical Grid - (b) 3D Spherical Grid of the Data Contained Inside the Dashed Black Box

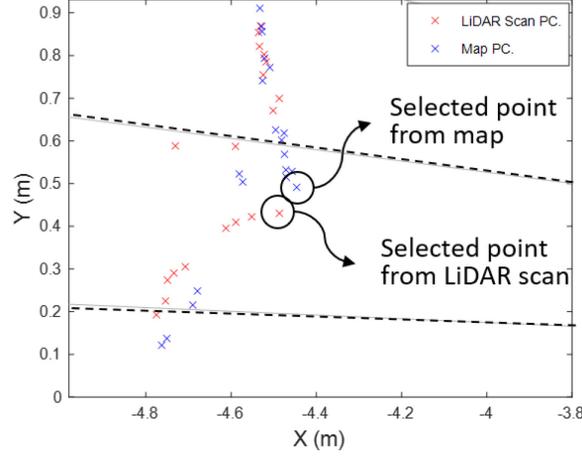


Figure 9. The Process of Selecting the Closest Points to the Sensor as Point Features

4.1.2 Pose candidate search algorithm

We use a straightforward search algorithm to estimate ADS state variables. First, for a predefined state parameter resolution, we define candidate ADS state variables $\mathbf{x}_{ADS,j}$ included in a search domain surrounding the ADS predicted pose $\bar{\mathbf{x}}_{ADS}$. Then, we define the measurement equations. Equation (13) shows the transformation of the map PC from the navigation frame to the sensor frame. The range and bearing angle of point ‘m’ from map will be considered as the computed measurements and can be written as a nonlinear function ξ of the map PC in sensor frame shown in equation (14). It is worth noting that the proposed approach can be extended to the 3D measurements model easily.

$${}^S\mathcal{P} = \mathbf{C}_N^S ({}^N\mathcal{P} - {}^N\mathbf{t}) \quad (13)$$

$$\begin{bmatrix} r_m \\ \theta_m \end{bmatrix} = \xi({}^S\mathcal{P}) \quad (14)$$

where

- ${}^S\mathcal{P}$ is map PC in sensor frame S,
- ${}^N\mathcal{P}$ is map PC in frame N,
- \mathbf{C}_N^S is rotation matrix from navigation frame to sensor frame,
- ${}^N\mathbf{t}$ is translation vector between frames S and N, using candidate j’s position in frame N,
- r_m is computed range from point ‘m’ to the origin of the frame S,
- θ_m is the computed bearing angle of point ‘m’ in S,

In the third step, we assume one of the candidates is the best representor of the ADS true position at each time step ‘k’ that will minimize the nonlinear least square (NLS) equation. We formulate NLS in equation (15) as the sum of the squared weighted norms of the measured minus computed range and measured minus computed bearing angle.

$$j^* = \arg \min_j \left(\left\| \mathbf{r}_{L_k} - \bar{\mathbf{r}}_{M_{j,k}} \right\|_{\mathbf{V}_r^{-1}}^2 + \left\| \boldsymbol{\theta}_{L_k} - \bar{\boldsymbol{\theta}}_{M_{j,k}} \right\|_{\mathbf{V}_\theta^{-1}}^2 \right) \quad (15)$$

$$\mathbf{r}_L = [r_{l_1} \quad \cdots \quad r_{l_{n_f}}]^T, \quad \boldsymbol{\theta}_L = [\theta_{l_1} \quad \cdots \quad \theta_{l_{n_f}}]^T$$

$$\bar{\mathbf{r}}_M = [r_{m_1} \quad \cdots \quad r_{m_{n_f}}]^T, \quad \bar{\boldsymbol{\theta}}_M = [\theta_{m_1} \quad \cdots \quad \theta_{m_{n_f}}]^T$$

where

- j^* is the best candidate,
- \mathbf{V}_r^{-1} is range measurement error covariance,
- \mathbf{V}_θ^{-1} is bearing angle measurement error covariance,
- n_f is number of point-features,
- r_l is measured range in sensor frame,
- θ_l is measured bearing angle in sensor frame.

Finally, we consider the best candidate as ADS state estimation at time ‘ k ’.

$$\hat{\mathbf{x}}_{ADS_k} = \mathbf{x}_{ADS_{j^*}} \quad (16)$$

Figure 10 is a block diagram of SGBL showing the spherical gridding algorithm and the pose estimation algorithm.

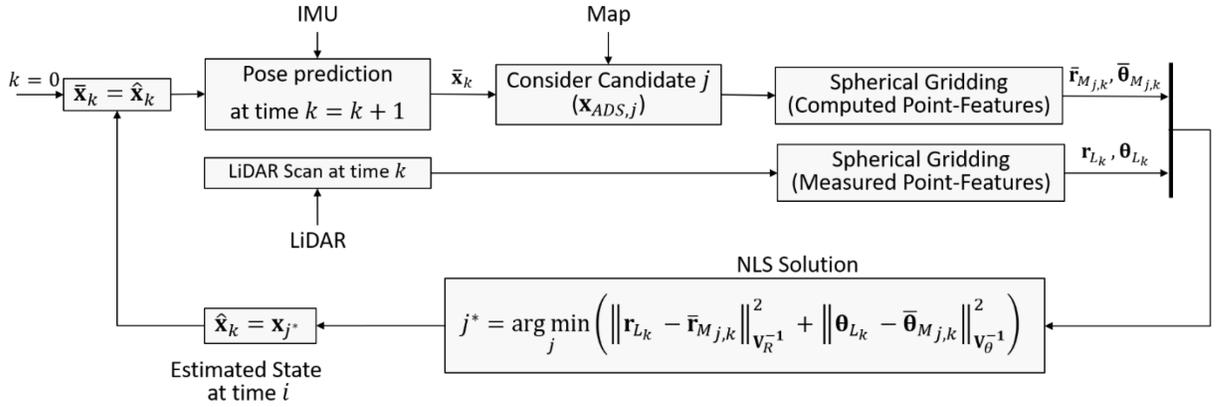


Figure 10. SGBL Block Diagram

4.2 Uncertainty quantification of spherical grid-based localization

The SGBL method combines OGM and ICP methods. The PC and map are partitioned into spherical grid cells, a single point-feature is considered per cell. SGBL uncertainty quantification is achieved using models of these point-features ranging and angular measurements errors, and a model of the error caused by the spherical grid resolution. For now, the pose candidate search resolution is assumed high enough that the resulting errors can be neglected. We can write the error equations as:

$$\varepsilon_{r_T} = \varepsilon_{r_M} \quad (17)$$

$$\varepsilon_{\theta_T} = \varepsilon_{\theta_G} + \varepsilon_{\theta_M} \quad (18)$$

where

- ε_{r_T} is the total range error,
- ε_{r_M} is the point-feature range error,
- ε_{θ_T} is the total bearing angle error,
- ε_{θ_G} is the spherical gridding error,
- ε_{θ_M} is the point-feature bearing angle error.

The distribution of ε_{r_M} , ε_{θ_M} and ε_{θ_G} can be evaluated using data and modeled using Gaussian overbounding. We assume zero mean normally distributed errors ε_{r_T} and ε_{θ_T} with variances $\sigma_{r_M}^2$ and $\sigma_{\theta_G}^2 + \sigma_{\theta_M}^2$, respectively.

The SGBL estimator is designed to find the solution of weighted NLS in equation (15). Since we formulated the SGBL estimator as a weighted NLS (it is really a search procedure), it seems reasonable to consider linearized ranging and angular measurement equations that would be implemented in an NLS estimator [18, 24]. The corresponding analytical covariance equation, assuming that the IMU-based prediction is only used to define the search space, is expressed as:

$$\mathbf{P}_k = (\mathbf{H}_k^T \mathbf{W}_{r,\theta} \mathbf{H}_k)^{-1} \quad (19)$$

$$\mathbf{H}_{k_{[2h \times n]}} = \begin{bmatrix} \vdots \\ \frac{\partial r_m}{\partial \mathbf{x}_{ADS}} \\ \frac{\partial \theta_m}{\partial \theta_m} \\ \frac{\partial \mathbf{x}_{ADS}}{\partial \theta_m} \\ \vdots \end{bmatrix}_{\bar{\mathbf{x}}_k}, \quad \mathbf{W}_{r,\theta} = \begin{bmatrix} \ddots & & & & \mathbf{0} \\ & \sigma_{r_M}^{-2} & & & \\ & & 1/(\sigma_{\theta_G}^2 + \sigma_{\theta_M}^2) & & \\ \mathbf{0} & & & & \ddots \end{bmatrix}_{[2h \times 2h]}$$

where

- \mathbf{P}_k is the SGBL analytical covariance at time k ,
- \mathbf{H}_k is the geometry matrix at time k ,
- $\mathbf{W}_{r,\theta}$ is the inverse of the measurement error covariance matrix,
- h is the number of point-features,
- n is the number of ADS states.

5 Experimental results of spherical grid-based localization

5.1 SGBL performance over a single lap

Figure 11(a) and (b) shows the 2D and 3D SGML's estimated trajectories versus true trajectory. Both methods achieve similar performance: they follow the true trajectory closely. Figure 11(b) focuses on heading estimation. Figure 11(c) shows error curves for both 2D and 3D SGBL along the cross-track, x and y directions and for the heading angle. These results show that all data points of a LiDAR PC in a cluttered environment can be used to match the pose estimation performance of LBL without requiring error-prone FE and DA.

5.2 SGB 100 trajectory laps estimation performance

In this section, the empirical SBGL pose estimation error distribution is evaluated for comparison with the analytical error model in equation (19). We use the same data set as for LBL in Section 3. Rover pose is estimated over 100 laps. Figure (12) shows the estimated trajectory with a black solid line, the true trajectory with a black dashed line (they overlap), the 1σ sample covariance ellipses in red, and the 1σ analytical covariance ellipses in blue. The analytical covariance ellipses are bounding the sample covariance ellipses. However, similar to LBL, the sample and analytical covariance ellipses differ in size and orientation (we speculate that sources of discrepancy are the same as for LBL). Figure (13) shows 100 pose estimation error curves in gray, the 1σ analytical covariance envelope in blue, and the 1σ sample covariance envelope in red. The sample covariance envelope is bounded by the analytical one during most of the figure-eight trajectory for all estimated states.

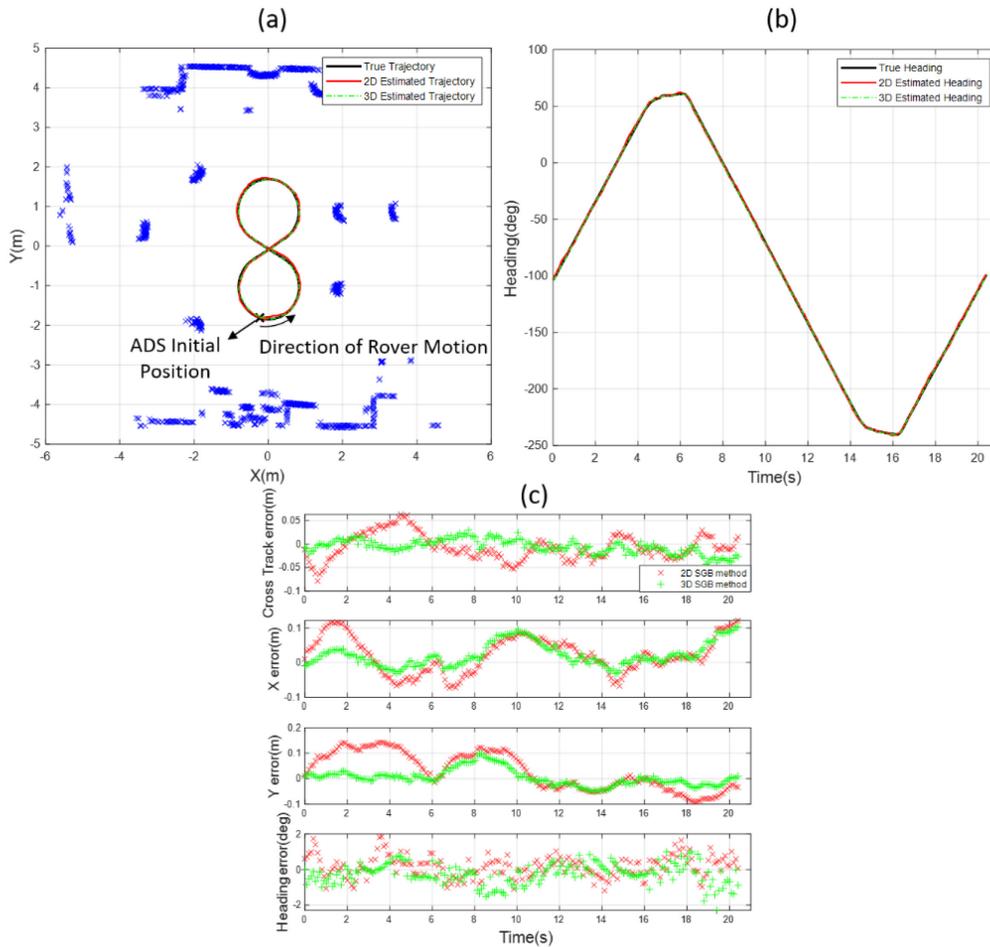


Figure 11. SGBL Performance for a Single Lap (a) Estimation of ADS Position - (b) Estimation of ADS Heading - (c) Sample State Estimate Error Curves.

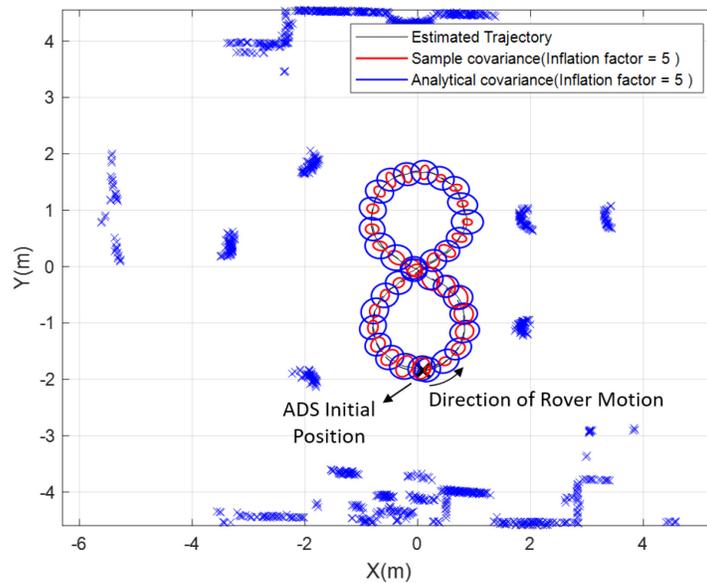


Figure 12. Estimated Trajectory, Analytical and Sample Covariances Using SGBL over 100 Laps.

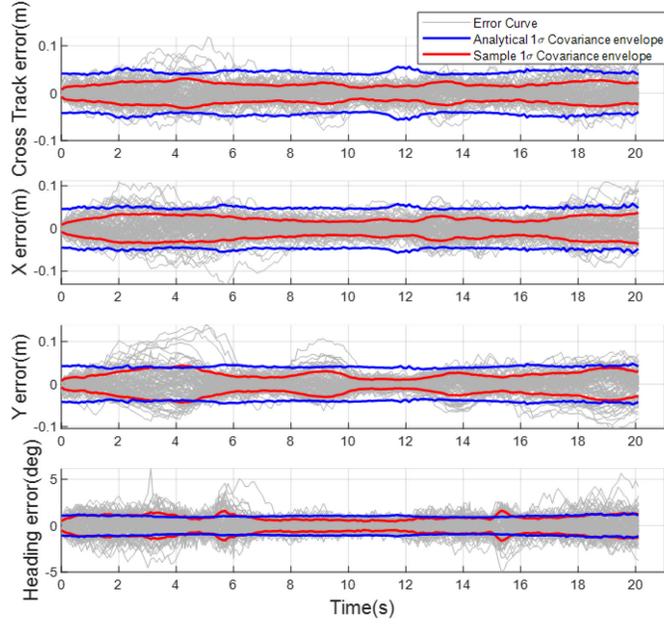


Figure 13. State Estimate Error Curves and $1\text{-}\sigma$ Analytical and Sample Covariance Envelopes Using SGML over 100 Laps.

6 Conclusions

In this paper, we designed an automated testbed for experimental evaluation of LiDAR/IMU position and orientation (or pose) estimation error distributions. We implemented this testbed to assess a landmark-based localization (LBL) method, which revealed significant discrepancies between modeled versus sample error distributions. Also, we developed and implemented a new spherical grid-based localization (SGBL) method together with its error model. Experimental evaluation showed that SGBL can provide the same level of pose estimation performance as LBL. In the next step of this work, we will refine the SGBL and LBL error models to obtain tighter bounds on pose estimation error distributions.

Appendix A – Linearized IMU and LiDAR Measurement Equations Coefficients

The coefficient matrices corresponding to IMU measurements in equation (3) can be defined as [22]:

$$\mathbf{F}_{V2r} = \begin{bmatrix} 0 & \frac{1}{R+h} & 0 \\ \frac{1}{R+h} & 0 & 0 \\ 0 & \frac{-\tan(\lambda)}{R+h} & 0 \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{F}_{H2V} = \begin{bmatrix} 0 & \frac{-2g_0}{R} & 0 \\ \frac{2g_0}{R} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.2})$$

where

R is the earth's radius,
 h is the vehicle's altitude,
 λ is the vehicle's latitude
 g_0 is the acceleration of gravity at zero altitude.

The coefficients matrices corresponding to the LiDAR measurements in equations (12) are [25]:

$$\mathbf{F}_{r,x} = \begin{bmatrix} -\frac{{}^1p_E - x_E}{\|{}^1\mathbf{p} - \bar{\mathbf{x}}_{EN}\|} & \cdots & -\frac{{}^{nL}p_E - x_E}{\|{}^{nL}\mathbf{p} - \bar{\mathbf{x}}_{EN}\|} \\ -\frac{{}^1p_N - x_N}{\|{}^1\mathbf{p} - \bar{\mathbf{x}}_{EN}\|} & \cdots & -\frac{{}^{nL}p_N - x_N}{\|{}^{nL}\mathbf{p} - \bar{\mathbf{x}}_{EN}\|} \\ 0 & \cdots & 0 \end{bmatrix}^T \quad (\text{A.3})$$

$$\mathbf{F}_{\theta,x} = \begin{bmatrix} \frac{{}^1p_N - x_N}{\|{}^1\mathbf{p} - \bar{\mathbf{x}}_{EN}\|^2} & \cdots & \frac{{}^{nL}p_N - x_N}{\|{}^{nL}\mathbf{p} - \bar{\mathbf{x}}_{EN}\|^2} \\ -\frac{{}^1p_E - x_E}{\|{}^1\mathbf{p} - \bar{\mathbf{x}}_{EN}\|^2} & \cdots & -\frac{{}^{nL}p_E - x_E}{\|{}^{nL}\mathbf{p} - \bar{\mathbf{x}}_{EN}\|^2} \\ 0 & \cdots & 0 \end{bmatrix}^T \quad (\text{A.4})$$

Where $\bar{\mathbf{x}}_{EN} = [x_E \quad x_N]^T$ (also appearing in ADS position states equation (5)) and $\mathbf{p} = [p_E \quad p_N]$

$$\mathbf{F}_{\theta,e} = \begin{bmatrix} 0 & 0 & 1 \\ \vdots & & \\ 0 & 0 & 1 \end{bmatrix}_{n_L \times 3} \quad (\text{A.5})$$

Appendix B –Discrete-Time Equations of IMU

We use the Van Loan algorithm to determine the discrete-time state propagation matrix Φ_k based on the continuous-time matrices \mathbf{F} and \mathbf{w} [23]. The following equations are the discrete-time form of terms in equation (3).

$${}^B\tilde{\boldsymbol{\omega}}_k^{\text{IB}} = [\mathbf{I} + \mathbf{S}_g + \mathbf{M}_g] {}^B\boldsymbol{\omega}_k^{\text{IB}} + \mathbf{b}_{g,k} + \mathbf{v}_{g,k} \quad (\text{B.1})$$

$${}^B\bar{\boldsymbol{\omega}}_k^{\text{IB}} = [\mathbf{I} + \hat{\mathbf{S}}_g + \hat{\mathbf{M}}_g]^{-1} ({}^B\tilde{\boldsymbol{\omega}}_k^{\text{IB}} - \hat{\mathbf{b}}_{g,k}) \quad (\text{B.2})$$

$${}^B\tilde{\mathbf{f}}_k^{\text{I}} = [\mathbf{I} + \mathbf{S}_a + \mathbf{M}_a] {}^B\mathbf{f}_k^{\text{I}} + \mathbf{b}_{a,k} + \mathbf{v}_{a,k} \quad (\text{B.3})$$

$${}^B\bar{\mathbf{f}}_k^{\text{I}} = [\mathbf{I} + \hat{\mathbf{S}}_a + \hat{\mathbf{M}}_a]^{-1} ({}^B\tilde{\mathbf{f}}_k^{\text{I}} - \hat{\mathbf{b}}_{a,k}) \quad (\text{B.4})$$

$$\mathbf{b}_{g,k+l} = \mathbf{e}^{-\frac{T_s}{\tau_g}} \mathbf{b}_{g,k} + \mathbf{n}_{a,k} \quad (\text{B.5})$$

$$\mathbf{b}_{a,k+l} = \mathbf{e}^{-\frac{T_s}{\tau_a}} \mathbf{b}_{a,k} + \mathbf{n}_{a,k} \quad (\text{B.6})$$

References

- [1] F. Wang and Z. Zhao, "A Survey of Iterative Closest Point Algorithm," 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 4395-4399, doi: 10.1109/CAC.2017.8243553.

- [2] A. Hassani, G. Duenas Arena, M. Spenko, and M. Joerger, "LiDAR Data Association Risk Reduction Using Tight Integration with INS," Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, FL, 2018.
- [3] A. Hassani, N. Morris, M. Spenko, and M. Joerger, "Experimental Integrity Evaluation of Tightly Integrated IMU/LiDAR Including Return-Light Intensity Data," Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, FL, 2019.
- [4] M. Joerger, M. Jamoom, M. Spenko, B. Pervan, "Integrity of Laser-Based Feature Extraction and Data Association," Proceedings of IEEE/ION PLANS 2016, Savannah, GA, April 2016, pp. 557-571.
- [5] M. Joerger, B. Pervan, "Continuity Risk of Feature Extraction for Laser-Based Navigation," Proceedings of the 2017 International Technical Meeting of the Institute of Navigation, Monterey, California, January 2017, pp. 839-855.
- [6] A. Hata and D. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," in 17th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014, pp. 584-589.
- [7] T. Ort, L. Paull, and D. Rus, "Autonomous vehicle navigation in rural environments without detailed prior maps," in International Conference on Robotics and Automation, 2018.
- [8] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based lowcost vehicle localization system for complex urban environments," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp. 1078-1086, 2017.
- [9] D. Gruyer, R. Belaroussi, and M. Revilloud, "Accurate lateral positioning from map data and road marking detection," Expert Systems with Applications, vol. 43, pp. 1-8, 2016.
- [10] X. Qu, B. Soheilian, and N. Paparoditis, "Vehicle localization using mono-camera and geo-referenced traffic signs," in Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE, 2015, pp. 605-610.
- [11] M. Joerger and A. Hassani, "A New Data Association Method Using Kalman Filter Innovation Vector Projections," 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 2020, pp. 318-327, doi: 10.1109/PLANS46316.2020.9110229.
- [12] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2010, pp. 4372-4378.
- [13] J. Castorena and S. Agarwal, "Ground-edge-based LIDAR localization without a reflectivity calibration for autonomous driving," IEEE Robot. Autom. Lett., vol. 3, no. 1, pp. 344-351, Jan. 2018.
- [14] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 2814-2821.
- [15] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," in Computer, vol. 22, no. 6, pp. 46-57, June 1989, doi: 10.1109/2.30720.
- [16] Z. Luo, M. V. Mohrenschildt, S. Habibi, "A Probability Occupancy Grid Based Approach for Real-Time LiDAR Ground Segmentation," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 998-1010, March 2020, doi: 10.1109/TITS.2019.2900548.
- [17] S. Oh, H. Kang, "Fast Occupancy Grid Filtering Using Grid Cell Clusters from LIDAR and Stereo Vision Sensor Data," in IEEE Sensors Journal, vol. 16, no. 19, pp. 7258-7266, Oct.1, 2016, doi: 10.1109/JSEN.2016.2598600.
- [18] S. Rusinkiewicz, M. Levoy. "Efficient Variants of the ICP Algorithm," Proceedings of the Third Intl. Conference on 3D Digital Imaging and Modeling, 145-152, 2001.
- [19] B. DeCleene, "Defining Pseudorange Integrity - Overbounding," Proc. ION GPS 2000, Salt Lake City, UT., 2000, pp. 1916-1924.
- [20] J. Rife et al., "Paired Overbounding for Nonideal LAAS and WAAS Error Distributions." IEEE TAES, Vol. 42, No. 4, 2006 pp. 1386-1395.
- [21] J. Blanch et al., "Gaussian Bounds of Sample Distributions for Integrity Analysis," IEEE Transactions on Aerospace and Electronic Systems, vol. 55, no. 4, pp. 1806-1815, 2019
- [22] D. Titterton, J. Weston, "Strapdown inertial navigation technology." Stevenage: The Inst. of Engineering and Technology, 2009
- [23] R. Brown, P. Hwang, "Introduction to Random Signals and Applied Kalman Filtering. Hoboken." NJ: John Wiley and Sons, 2012

- [24] A. Diosi, L. Kleeman “Fast Laser Scan Matching using Polar Coordinates.” *The International Journal of Robotics Research*. 2007, 26(10):1125-1153, doi:10.1177/0278364907082042
- [25] M. Joerger, “Carrier Phase GPS Augmentation Using Laser Scanners and Using Low Earth Orbiting Satellites,” Ph.D. Dissertation, Illinois Institute of Technology, 2009.