Spherical Grid-Based IMU/LiDAR Localization and Uncertainty Evaluation Using Signal Quantization

Ali Hassani and Mathieu Joerger
Virginia Tech, Blacksburg, VA

Abstract

This paper describes the design, analysis, and experimental evaluation of a spherical-grid-based localization algorithm that leverages quantization theory to bound navigation uncertainty. This algorithm integrates data from Light Detection And Ranging (LiDAR) and inertial measuring units (IMU) in an iterative extended Kalman filter to estimate the position and orientation of a moving vehicle. An analytical bound is derived on the vehicle's state estimation error, which accounts for both the random measurement noise and the loss of localization information caused by gridding. The performance of the proposed approach is analyzed and compared with that of a brute-force spherical grid-based method and of a landmark-based method in an indoor environment.

1 Introduction

In this paper, we develop, implement, and evaluate a new point cloud (PC)-based localization algorithm integrating Light Detection And ranging (LiDAR) and inertial measuring units (IMU) using a signal quantizer and an iterative extended Kalman filter (IEKF). In prior work [1], we introduced a spherical grid-based localization (SGL) method to estimate position and orientation (or pose) corrections between a measured three-dimensional PC and a map-based predicted PC. These corrections were obtained by searching over candidate vehicle poses. In this work, we achieve SGL using an IEKF, which enables tight integration of LiDAR PC with IMU and prediction of pose estimation uncertainty. We develop an automated approach that locally adjusts the size of the grid cells to minimize the information loss due to gridding. In addition, we derive an analytical bound on the spherical gridding error and evaluate its impact on pose estimation error. These methods are tested using experimental data collected in an indoor environment.

This research is intended for autonomous ground, air and space vehicle navigation applications where localization uncertainty quantification is key. In prior work [2, 3], we developed and evaluated a landmark-based localization (LL) method, which required two intermediary pre-processing steps: feature extraction (FE) and data association (DA). FE finds viewpoint-invariant landmarks in the LiDAR PC, and DA assigns these extracted landmarks to mapped ones. FE and DA are computationally expensive and prone to faults in cluttered environments [2, 4].

LiDAR localization methods that do not require FE and DA include PC-matching algorithms, such as the widely-implemented iterative closest point (ICP) [5, 6, 7]. ICP can be computationally expensive and does not enable reliable uncertainty prediction. To mitigate the computational burden of handling a three-dimensional LiDAR PC's thousands of high-update-rate data samples, we developed an SGL method. The spherical grid in [1] is made of azimuth-elevation bins at regular angular intervals. In each bin, a point feature is selected and its distance to the spherical grid's center point serves as range measurement. This gridding process is also applied to a PC-map seen from the perspective of the LiDAR's predicted pose. The measured PC can then be compared to the computed PC. The SGL's point-feature approach only provides a rough measure of navigation uncertainty [1]. Gridding and point selection can be improved, and rigorous error quantification has yet to be achieved.

In this paper, we develop a new PC spherical gridding approach that leverages a signal quantization technique used in data compression [8]. This approach locally modifies grid spacing to reduce information loss and improve pose estimation without increasing computation cost. It also provides a deterministic bound on the range quantization error in each azimuth-elevation bin. In addition, we design an IEKF that tightly integrates IMUs with LiDAR SGL. The IMU facilitates matching of measured versus computed PC whereas the SGL helps correct accelerometer and gyro biases. We validate these methods using experimental data collected on a sensor platform moving in an indoor environment.

The first part of the paper aims at developing an advanced LiDAR PC spherical gridding process. We leverage quantization theory which is a well-studied topic in data transmission, compression and classification [8, 9], information theory [10] and finite element method for mesh generation [11]. We use it here to rearrange a spherical grid's discretized representation of the environment [12, 13]. The algorithm assigns a greater number of azimuth-bins to parts of the environment that show larger geometric variations.

In the second part of the paper, we design an IEKF that incorporates the spherically gridded LiDAR PC, IMUs, and their error models for the estimation and uncertainty quantification of a moving platform's pose, and of the IMU's accelerometer and gyroscope biases.

In the third part, we derive an upper-bound on the impact of quantization errors on pose estimation uncertainty. In this preliminary algorithm, the estimated-state-level bound is achieved using a batch algorithm.

In the fourth part of the paper, we implement the IEKF-based IMU/LiDAR SGL algorithm by post-processing experimental data. All objects in the environment are static, but they can get occluded or can get out of the LiDAR's field of view as the sensor-platform moves. A probabilistic performance evaluation is conducted for the proposed IEKF-based SGL as well as for LL and brute-force SGL, to show the robustness of the new method in quantifying positioning uncertainty.

2 Spherical gridding using signal quantization

2.1 Quantization theory

Quantization is a process in which a large number of values is represented by a much smaller set of values [8]. If the values that we are quantizing are scalar, the process is called scalar quantization. The quantizer design aims at minimizing information loss. Figure 1 shows the parameters of a scalar quantizer.

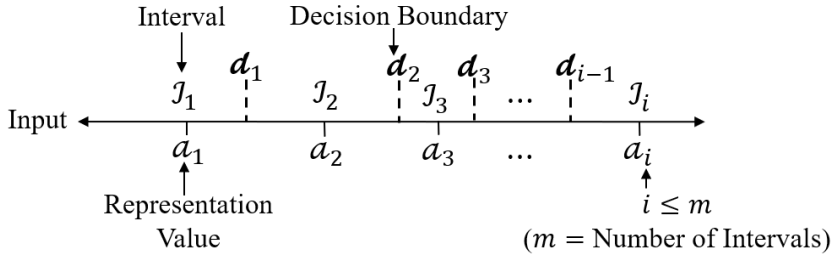

Figure 1: Scalar quantizer parameters

We define the following parameters:

- Input: The input can be a set of deterministic or random values with an associated probability density function (PDF), i.e. R with PDF $f_R(r)$.
- Interval: The interval partitions the input into separate ranges of values $\mathcal{I} = \{\mathcal{I}_i, \text{ for } i = 1,..,m\}$. Intervals are also known as Voronoi regions or encoders.
- Representation value: The representation value is a single value selected for each interval. It is the output of the quantizer. Representation values are also known as decoders.
- Decision Boundary: The limit value of an interval is a decision boundary. The number of decision boundaries is $m + 1$, where m is the number of intervals.

The quantization process $Q(r)$ for an input r can be described as:

$$d_{i-1} < r < d_i \Leftrightarrow Q(r) = a_i \tag{1}$$

In this work, the performance measure used to reduce information loss between quantizer input and output is the quantization distortion D defined as:

$$D = \frac{1}{M}\sum_{i=1}^{m}\sum_{r\in\mathcal{I}_i}(r - a_i)^2 \quad (2)$$

where the right-hand side of equation (2) is the sum of the squared errors in each interval $\mathcal{I}_i$ summed over all m intervals and divided by the total number of range measurements M. The number M is defined as $M = \sum_{i=1}^{m} M_i$ where M_i is the number of range measurements per interval $\mathcal{I}_i$, for $i = 1,..,m$.

The quantization process can be described as follow. Given an input r and a number of intervals m, find the values of d_i and a_i in each interval $\mathcal{I}_i$, for $i = 1,..,m$, that minimize the distortion D. There are two main categories of quantizers: uniform quantizers use fixed-size intervals, whereas non-uniform quantizers do not.

In this paper, we use a Lloyd-Max quantizer, which is a practical non-uniform scalar quantizer. We will use LiDAR range measurements as input r, as further explained in Section 2.2. The Lloyd-Max quantization algorithm finds the values of a_i and d_i that minimize the distortion by taking the derivatives of D with respect to these parameters and setting them equal to zero. The values of a_i and d_i for a set of inputs r can be found using the following equations [12]:

$$a_i = \frac{1}{M_i}\sum_{r\in\mathcal{I}_i} r \quad (3)$$

$$d_i = \frac{a_{i+1}+a_i}{2} \quad (4)$$

For a Lloyd-Max quantizer, the representation values a_i in equation (3) are equal to the mean of the input values in interval i, and d_i in equation (4) is the mid-point of two neighboring representation values. Considering initial values for d_i and a_i, the minimization of D can be iteratively achieved [8, 12].

2.2 Spherical grid design

This section aims at applying quantization techniques to LiDAR PCs. Data points are processed one elevation-cone at a time using the approach described below. Instead of the traditional Cartesian representation of a LiDAR PC, we consider a range-versus-bearing-angle representation for a single elevation angle, and for 360 degree azimuth angles – which we call a single frame at that elevation. It is represented as a blue line in Figure 2, where the LiDAR scan (blue-colored curve) is interpreted as a range signal varying over azimuth angle.

Then, we use quantization theory to find the optimal decision boundaries d_i and representation values a_i, for $i = 1,..,m$, which minimize the distortion D. These values are respectively represented in Figure 2 as horizontal dashed gray lines and red solid lines. The vertical solid gray lines show the boundaries of the azimuth bins, which are defined as bearing angle intervals with constant a_i- values (a change in a_i defines a new azimuth bin, even if that a_i-value defined a previous azimuth bin). Figure 3 shows the quantizer parameters and the azimuth bins in a zoomed-in region of the LiDAR range-versus-bearing-angle curve.

Figure 4 shows the output measurements in a cartesian coordinate system. This output is obtained after minor data trimming to exclude azimuth bins containing less than a minimum number of data points (which only seemed to add computation cost). The solid gray lines intersecting at the origin correspond to the vertical gray lines in Figure 2. In Figure 4, the LiDAR scan is represented with a blue curve and the selected points after quantization are shown with red crosses. The zoomed-in window shows raw azimuth-range measurements in each bin and their corresponding quantized values.

2.3 Quantization error

The quantization error is defined as the difference between quantizer input r and output a_i. The quantization error b_i, for $i = 1,..,m$, is defined as [8]:

$$b_i = d_{i-1} - d_i \quad (5)$$

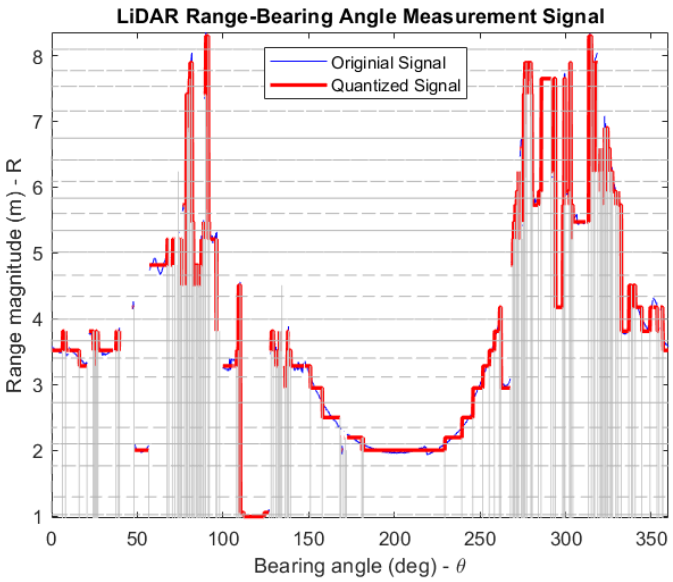

Figure 2: LiDAR scanned and quantized range measurements

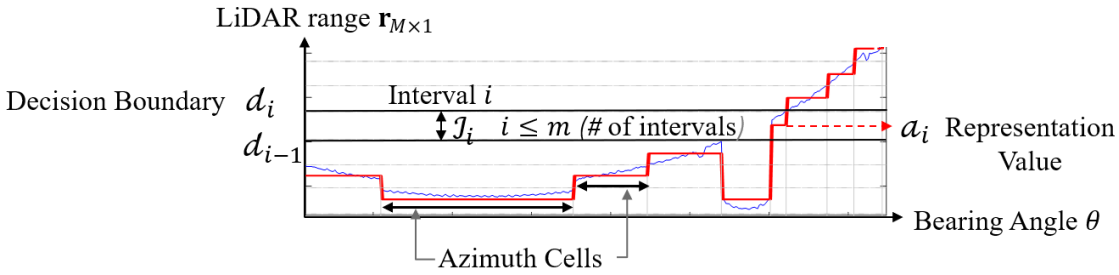

Figure 3: Quantizer parameter definition for LiDAR range vs. bearing angle measurements

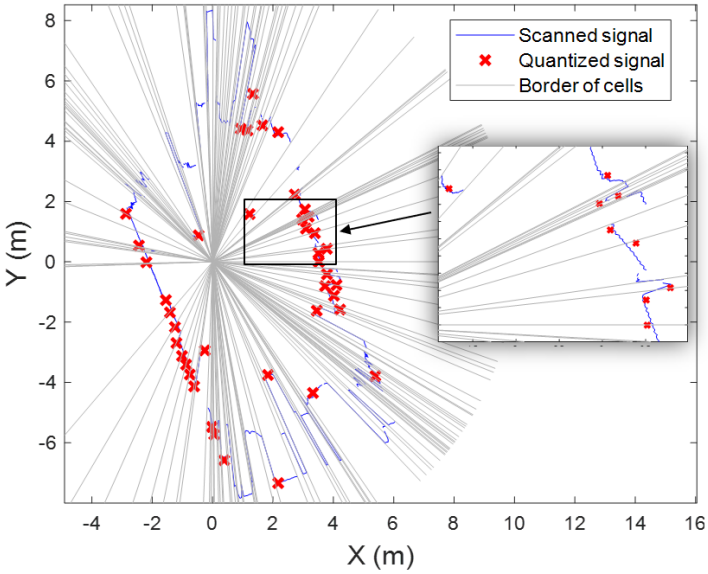

Figure 4: LiDAR scan and quantized signals

The decision boundaries d_i are given in equation (4). Quantization theory ensures that the following inequality is always satisfied:

$$|r - a_i| < b_i \tag{6}$$

Figure 5 shows quantization errors (blue) and error bounds (red) for all intervals in a frame. The red curve bounds the blue curve and is *known*.

The range measurement quantization error's bounding bias vector is defined as:

$$\mathbf{b}_k = [b_1 \quad \cdots \quad b_m \quad \mathbf{0}_{1\times m}]^T \tag{7}$$

where k is time epoch and b_i is a defined in equation (5). Zeros in equation (7) correspond to bearing angle measurements, which are not quantized in this work.

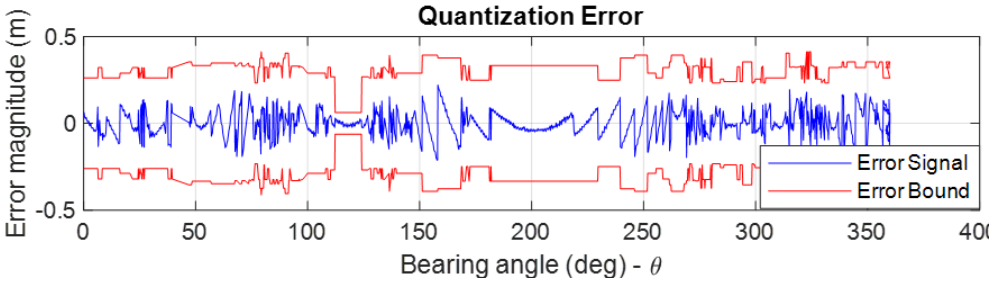

Figure 5: Quantization error and error bound for 360-degree measurements

2.4 Applying spherical grid to the map

We then apply the same spherical grid to the map PC. We first convert the map PC from the navigation frame to the sensor frame using the predicted state vector $\bar{\mathbf{x}}$ provided by the pose estimation algorithm. Then, in each azimuth-elevation bin, we select the mapped point closest to the sensor. This approach automatically addresses occlusions of objects that may not be visible from the LiDAR's current point of view. Figure 6 shows the LiDAR and map PCs after quantization. The azimuth bins are shown in gray.

3 Localization and analytical uncertainty quantification

In this section, we design an IEKF which uses the mapped and sensed PCs output by the spherical gridding process. We also design a batch algorithm to bound the impact of range quantizer errors on pose estimation.

3.1 Vehicle linearized state propagation model

The localization algorithm uses the IMU measurements and error model to propagate vehicle pose between two LiDAR updates. The IMU states consists of vehicle position, velocity, orientation, and IMU biases. This section describes the continuous-time linearized state propagation model. The complete nonlinear continuous and discrete-time equations and sensor error models can be found in [2,14]. In the following equations, the notation ' δ ' indicates deviations of the state parameters relative to reference values about which linearization is performed.

$$\delta\dot{\mathbf{x}} = \mathbf{F}\,\delta\mathbf{x} + \delta\mathbf{w} \tag{8}$$

$$\delta\mathbf{x} = [\delta\mathbf{x}_V^T \quad \delta\mathbf{v}_V^T \quad \delta\mathbf{e}_V^T \quad \delta\mathbf{b}_g^T \quad \delta\mathbf{b}_a^T] \tag{9}$$

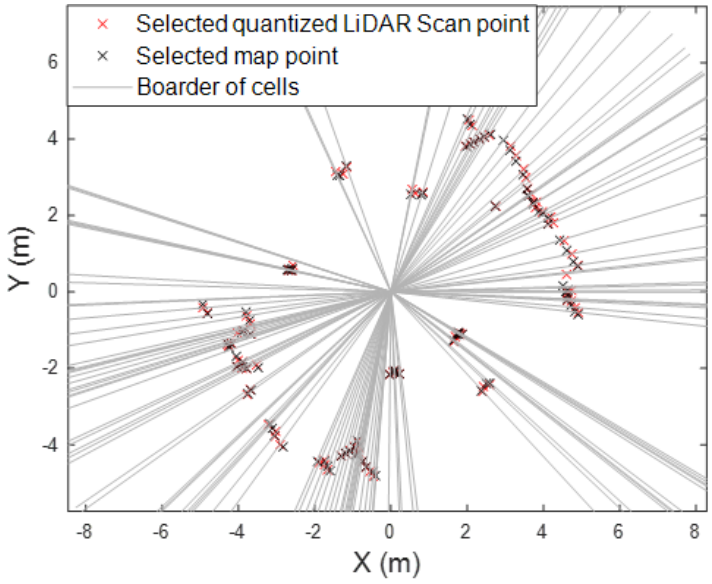

Figure 6: Applying the spherical grid to the map and selecting the points

$$\mathbf{F}=\begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{\text{H2V}} & \mathbf{0} & [{}^{\text{N}}\bar{\mathbf{f}}^{\text{I}}\times] & \mathbf{0} & \mathbf{C}_{\text{B}}^{\text{N}} \\ \mathbf{0} & \mathbf{F}_{\text{V2T}} & -[{}^{\text{N}}\boldsymbol{\omega}^{\text{IN}}\times] & -\mathbf{C}_{\text{B}}^{\text{N}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tau_g^{-1}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tau_a^{-1}\mathbf{I} \end{bmatrix},\quad \delta\mathbf{w}=\begin{bmatrix} \mathbf{0} \\ \mathbf{C}_{\text{B}}^{\text{N}}(-\delta\mathbf{S}_a\,{}^{\text{B}}\bar{\mathbf{f}}^{\text{I}}-\delta\mathbf{M}_a\,{}^{\text{B}}\bar{\mathbf{f}}^{\text{I}}-\boldsymbol{\upsilon}_a) \\ -\mathbf{C}_{\text{B}}^{\text{N}}(-\delta\mathbf{S}_g\,{}^{\text{B}}\bar{\boldsymbol{\omega}}^{\text{IB}}-\delta\mathbf{M}_g\,{}^{\text{B}}\bar{\boldsymbol{\omega}}^{\text{IB}}-\boldsymbol{\upsilon}_g) \\ \bar{\mathbf{n}}_g \\ \bar{\mathbf{n}}_a \end{bmatrix} \tag{10}$$

where

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

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

$\delta\mathbf{e}_V$ 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,

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

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

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

$\mathbf{C}_{\mathbf{B}}^{\mathbf{N}}$ is the rotation matrix from frame B to frame-N [14],

${}^{\text{B}}\bar{\mathbf{f}}^{\text{I}}$ is the measured specific force vector at the IMU axis center w.r.t. frame I expressed in frame B [14],

${}^{\text{B}}\bar{\boldsymbol{\omega}}^{\text{IB}}$ 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,

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

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

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

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

Considering the discrete-time expressions of the terms in equation (10) given in Appendix A, the discrete-time realization of equation (8) can be written as:

$$\delta\mathbf{x}_{k+1}=\boldsymbol{\Phi}_k\delta\mathbf{x}_k+\delta\mathbf{w}_k \tag{11}$$

where $\mathbf{\Phi}_k$ is the state transition matrix between time steps 'k' and 'k+1' [15].

3.2 Measurement models

In this section, we derive the nonlinear equations for the sensed and mapped PCs designed in Sections 2.2 and 2.4. Each mapped data point in navigation frame 'N' ${}^{\mathrm{N}}\mathbf{p}_j = [p_{E,j} \quad p_{N,j} \quad p_{U,j}]^T$ for $j = 1 \cdots N_m$, where N_m is the total number of mapped points, is projected in sensor frame using the following equations:

$$ {}^{\mathrm{S}}\mathbf{p}_j = \mathbf{C}_{\mathrm{N}}^{\mathrm{S}}\left({}^{\mathrm{N}}\mathbf{p}_j - \bar{\mathbf{x}}_V\right) \qquad \text{for } j = 1 \cdots N_m \tag{12} $$

where deviations on $\mathbf{x}_V$ and $\mathbf{e}_V$ appear in the state error equation (9), and ${}^{\mathrm{S}}\mathbf{p}_j = [p_{1,j} \quad p_{2,j} \quad p_{3,j}]^T$ is the mapped data point in sensor frame 'S', $\mathbf{C}_{\mathrm{N}}^{\mathrm{S}}$ is rotation matrix from navigation frame to sensor frame using $\bar{\mathbf{e}}_V$. We use the notations: $\bar{\mathbf{e}}_V = [\phi \quad \gamma \quad \psi]^T$ and $\bar{\mathbf{x}}_V = [x_E \quad x_N \quad x_U]^T$.

After applying the spherical grid to ${}^{\mathrm{S}}\mathbf{p}_{j=1\cdots N_m}$, for each elevation range, we define the gridded points' ranges $\bar{r}_i$ and bearing angles $\bar{\theta}_i$, for i ranging from 1 to m, as:

$$ \bar{r}_i = \sqrt{p_{1,j}^2 + p_{2,j}^2} + \upsilon_r \tag{13} $$

$$ \bar{\theta}_i = \tan^{-1}\left(\frac{p_{2,j}^2}{p_{1,j}^2}\right) + \upsilon_\theta \tag{14} $$

where υ_r and υ_θ respectively are random range and bearing angle measurement errors,

We then stack the computed range and bearing angle measurements and define the mapped measurements at time step k as:

$$ \mathbf{h}_k\left(\bar{\mathbf{x}}_k\right) = \begin{bmatrix} \bar{\mathbf{r}} \\ \bar{\boldsymbol{\theta}} \end{bmatrix} \tag{15} $$

$$ \bar{\mathbf{r}} = [\bar{r}_1 \quad \cdots \quad \bar{r}_m]^T, \ \bar{\boldsymbol{\theta}} = [\bar{\theta}_1 \quad \cdots \quad \bar{\theta}_m]^T $$

The LiDAR range r_i and bearing angle θ_i are provided in the sensor frame. The $2m$ sensed measurement vector can be written as:

$$ \hat{\mathbf{z}}_k = \mathbf{h}_k(\mathbf{x}_k) + \boldsymbol{\upsilon}_k \tag{16} $$

$$ \hat{\mathbf{z}}_k = [r_1 \quad \cdots \quad r_m \quad \theta_1 \quad \cdots \quad \theta_m]^T \tag{17} $$

$$ \boldsymbol{\upsilon}_k = [\upsilon_{r_1} \quad \cdots \quad \upsilon_{r_m} \quad \upsilon_{\theta_1} \quad \cdots \quad \upsilon_{\theta_m}]^T \tag{18} $$

where

$\mathbf{x}_k$ is the state vector whose error vector is defined in equation (9),

$\boldsymbol{\upsilon}_k$ is the $2m \times 1$ measurement error vector modeled as a vector of zero-mean normally distributed random variables with covariance matrix $\mathbf{V}_k$. We use the notation: $\boldsymbol{\upsilon}_k \sim \mathrm{N}(\mathbf{0}, \ \mathbf{V}_k)$.

We linearize equation (16) about our best pose prediction of the vehicle. The linearized range and angular measurement and measurement error vectors are respectively designated by $\delta\mathbf{R}$, $\delta\boldsymbol{\theta}$ and $\boldsymbol{\upsilon}_r$ and $\boldsymbol{\upsilon}_\theta$. The linearized LiDAR measurement equation can be written as:

$$\begin{bmatrix}\delta\mathbf{R}\\ \delta\boldsymbol{\theta}\end{bmatrix}_k = \begin{bmatrix}\mathbf{F}_{r,x} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{F}_{\theta,x} & \mathbf{0} & -\mathbf{F}_{\theta,e} & \mathbf{0} & \mathbf{0}\end{bmatrix}_k \begin{bmatrix}\delta\mathbf{x}_V\\ \delta\mathbf{v}_V\\ \delta\mathbf{e}_V\\ \delta\mathbf{b}_g\\ \delta\mathbf{b}_a\end{bmatrix}_k + \begin{bmatrix}\boldsymbol{\upsilon}_r\\ \boldsymbol{\upsilon}_\theta\end{bmatrix}_k \tag{19}$$

where the coefficient matrices $\mathbf{F}_{r,x}$, $\mathbf{F}_{\theta,x}$ and $\mathbf{F}_{\theta,e}$ are determined using the state prediction vector as described in Appendix B.

3.3 Model-based estimator design

We develop an IEKF to tightly integrate LiDAR and IMU measurements and estimate vehicle pose. Figure 7 is a diagram of IEKF SGL including the spherical gridding and pose estimation processes. In the IEKF pose estimation block, the last term in the state vector estimation equation improves convergence of the iterative solution. The contribution to pose estimation error due to quantization is explained in section 3.4 .

where

$\bar{\mathbf{x}}_k$ is the predicted state vector at time step k
$\bar{\mathbf{P}}_k$ is the predicted covariance matrix at time step k
$\mathbf{P}_k$ is the estimated covariance matrix at time step k
$\mathbf{H}_k$ is the observation matrix defined in equation (21) at time step k
$\mathbf{K}_k$ is the Kalman gain at time step k
$\widehat{\boldsymbol{\Phi}}_k$ is the discrete-time state transition matrix at time step k

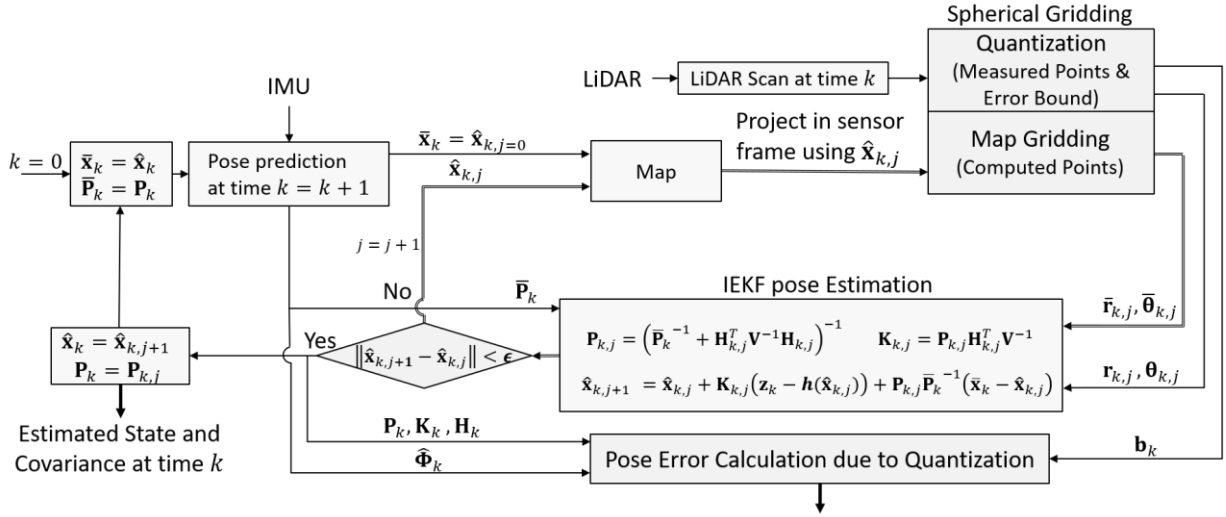

Figure 7: IEKF spherical grid-based localization diagram

3.4 Pose error due to quantization

In this section, we determine the impact of the quantization bias vector on state estimation error [16]. The expected value of the Kalman filter error in the presence of the bias $\mathbf{b}_k$ can be written as:

$$\mathrm{E}\left[\mathbf{x}_{k|k}\right] = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\widehat{\boldsymbol{\Phi}}_k \mathrm{E}\left[\mathbf{x}_{k-1|k-1}\right] + \mathbf{K}_k\mathbf{b}_k \tag{20}$$

The impact of the quantization bias on the a-posteriori estimation of the state vector at time step k is [17]:

$$\mathrm{E}[\mathbf{x}_k] = [\mathbf{A}_{1k} \quad \cdots \quad \mathbf{A}_{Kk}] \begin{bmatrix}\mathbf{b}_1\\ \vdots\\ \mathbf{b}_k\end{bmatrix} = \mathbf{A}_\mathrm{K}\mathbf{b}_\mathrm{K} \tag{21}$$

where

$$\mathbf{A}_{mn} = \begin{cases} \left(\prod_{t=n}^{m+1}(\mathrm{I} + \mathbf{K}_n\mathbf{H}_n^T)\widehat{\mathbf{\Phi}}_{k,m}\right)\mathbf{K}_m & if\ m < n \\ \mathbf{K}_m & if\ n = m \end{cases} \tag{22}$$

In equation (21), bias vectors $\mathbf{b}_i$, for $i = 1, \ldots, k$, are stacked from time step 1 to *k* in $\mathbf{b}_{\mathrm{K}}$**,** where K designates time steps 1,…,*k*. We can express the impact of the range quantization bias on a specific state of interest as:

$$c_{s,k}^2 = \mathbf{b}_{\mathrm{K}}^T\mathbf{A}_{\mathrm{K}}^T\boldsymbol{\alpha}_s\boldsymbol{\alpha}_s^T\mathbf{A}_{\mathrm{K}}\mathbf{b}_{\mathrm{K}} \tag{23}$$

where $\boldsymbol{\alpha}_s$ is a vector that extracts the state of interest from $\mathbf{x}_k$, i.e., consisting of a 1 for the desired state and zeros for all other states. Calculating the $c_{s,k}$ requires that we stack biases and estimator coefficients for time steps 1 to *k*. In future work, we will design a recursive approach to address this problem.

The state estimation error variance for the state of interest is given by:

$$\sigma_{s,k}^2 = \boldsymbol{\alpha}_s^T\mathbf{P}_k\boldsymbol{\alpha}_s \tag{24}$$

Both terms in equations (23) and (24) contribute to defining estimation error bounds for a desired confidence interval $\wp$. The estimation error bound on a state of interest is defined as:

$$L_{\wp,s,k} = \lambda c_{s,k} + \kappa_\wp\sigma_{s,k} \quad \text{with} \quad \lambda = \begin{cases} 1 & \wp \geq 50\% \\ -1 & \wp \leq 50\% \end{cases} \tag{25}$$

where $\kappa_\wp$ is a confidence multiplier for deviations with respect to median value ($\kappa_\wp = \Phi^{-1}(0.5 + \lambda\wp/2)$ and $\Phi^{-1}(\,)$ is the inverse of cumulative distribution function (CDF) for a standard normal distribution. For example, for a 68% confidence interval ($\wp = 0.68$), we can determine the 84% and 16% quantiles of the estimation error bounds as:

$$L_{84\%,s,k} = c_{s,k} + \sigma_{s,k} \tag{26}$$

$$L_{16\%,s,k} = -(c_{s,k} + \sigma_{s,k}) \tag{27}$$

where the impact of the quantization error $c_{s,k}$ is accounted for in the worst-contributing manner to guarantee a bound on the actual error quantiles. If $c_{s,k} = 0$, then the range $L_{16\%,s,k}$ to $L_{84\%,s,k}$ defines a 1-sigma (68%) error envelope.

4 Experimental results of spherical grid-based localization

4.1 Experimental testbed and settings

To quantify pose estimation errors, we use a testbed described in [2] and shown in Figure 8. It includes a sensor platform equipped with a Velodyne VLP-16 Puck LTE LiDAR, a NovAtel IMU-IGM-A1. Sixteen Optitrack Prime 13W infrared (IR) motion capture cameras provide sub-centimeter-level positioning by tracking retro-reflective markers fixed on the rover. The platform moves on a figure-eight track. The IEKF SGL uses the LiDAR PC of the entire lab including vertical cylinders which serve as landmarks in the LL method. The settings for the experiment are shown in Table 1. Then, we are introducing two performance measures and comparing the IEKF SGL performance with a brute force SGL and a LL performance.

4.2 IEKF SGL estimation performance over a single lap

Figure 9 shows the top-view true versus estimated rover trajectories which overlap most of the time. The mapped PC is shown with blue x-markers (crosses). A color-coded background is used to facilitate interpretation of subsequent figures over time: the upper loop's background is shown in white, the lower loop in light gray, and the straight parts of the trajectory in dark gray.

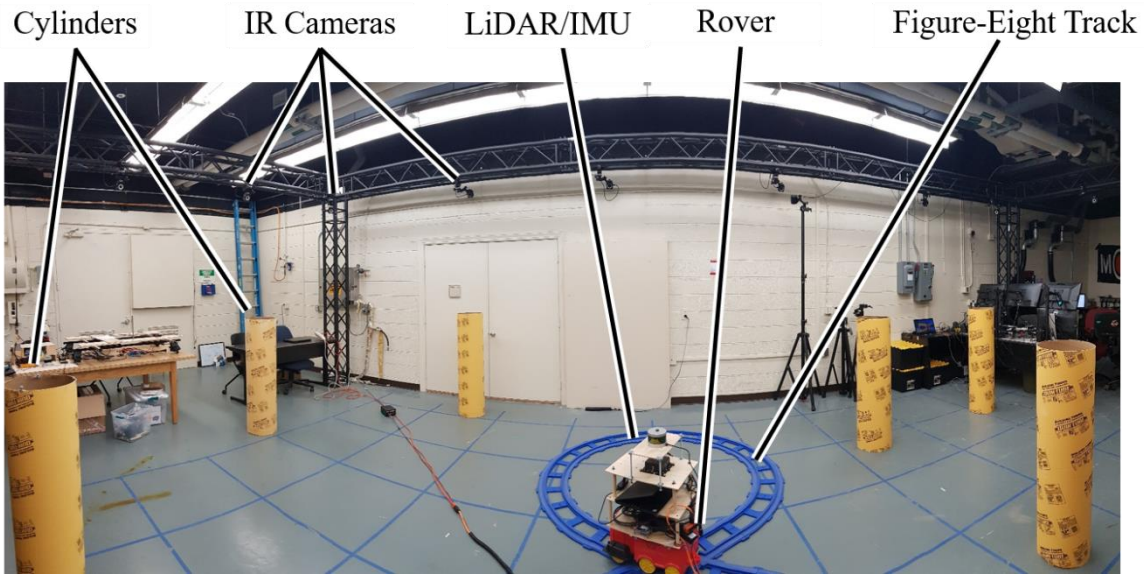

Figure 8. Testbed overview [2].

Table 1: Experiment settings

Parameters	Values	Parameters	Values
Range measurement standard deviation	0.04 m	Initial heading standard deviation	10 deg
Bearing angle measurement standard deviation	3 deg	Initial Roll/pitch standard deviation	1 deg
Initial position standard deviation	0.5 m	Number of iterations	5
Initial velocity standard deviation	0.05 m/s		

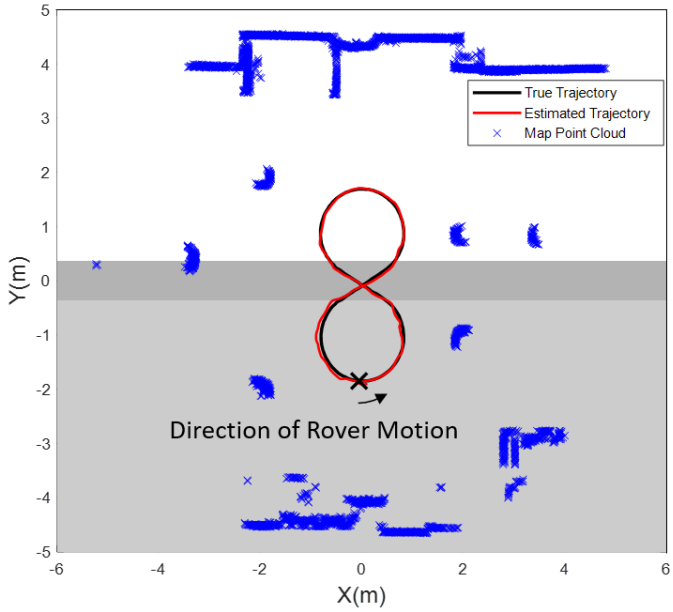

Figure 9: IEKF spherical grid-based localization performance for a single lap estimation of ADS Position

4.3 IEKF SGL estimation performance for 80 trajectory laps

This section analyzes IEKF SGL performance using experimental data collected over 80 laps. Figure 10(a) shows the cross-track error over time. The sample cross track positioning error is color coded in shades of gray, from white to black as the rover travels from the first to the last lap –the experimental sensor system exhibits a warm-up behavior which shows slight differences in the early laps as compared to the later ones. The blue-colored envelope represents the error-bounds in equation (26) and (27), each of which covers an area defined by the minimum and maximum $L_{84\%,s,k}$ and $L_{16\%,s,k}$ values over the 80 laps. The solid red line is the 68%-error envelope (i.e., the 84% and 16% sample quantiles), which is bounded by the analytical blue envelope for most of the 22 second-long trajectory. Figure 11(b) shows the corresponding curves for the heading angle of the vehicle. In this case again, the analytical envelope bounds the sample error envelope for most of the trajectory.

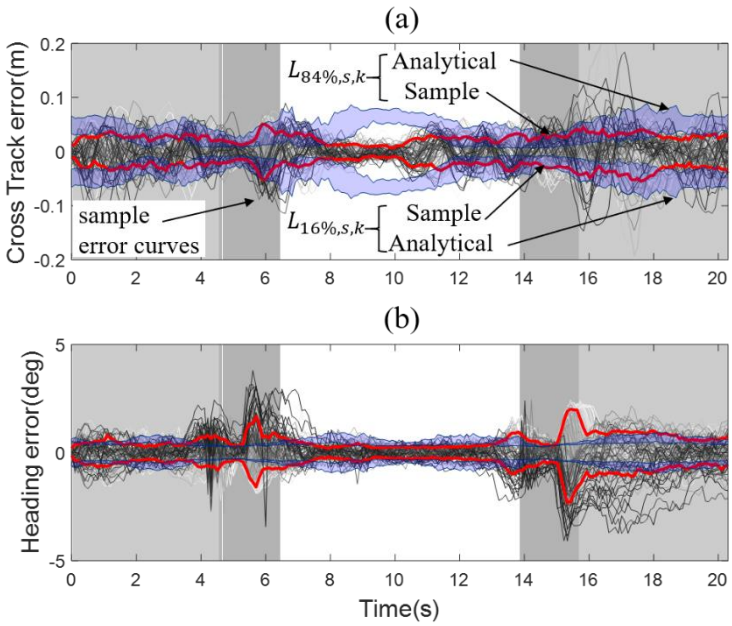

Figure 10: IEKF spherical grid-based localization error bounds for ADS (a) cross track error (b) heading error

Estimation performance comparison

In Table 2, we compare the performance of landmark-based localization (LL), brute-force spherical-grid-based localization (BF-SGL), and IEKF SGL. The performance measures for each approach include the maximum-over-the-trajectory of the 84% quantile cross-track error bounds (equal to the 1-sigma bound when quantization/gridding errors are neglected) and the corresponding single-lap-computation time. Computation time was evaluated using a 4.00 GHz processor, 32 GB RAM and x64 windows. The maximum error bound for LL is 5 cm: LL requires feature extraction (FE) and data association (DA) that introduce additional risks [1,2], which are not easily captured using the metrics in Table 2, and are therefore not accounted for in Table 2. The BF-SGL error bound is also 5cm: BF-SGL does not require FE and DA, but it only provides a 1-sigma bound with no means to account for gridding errors. Finally, the IEKF SGL provides a maximum 84% quantile bound of 8cm: in this case, spherical gridding errors are accounted for.

The longest computation time is for BF SGL and is determined by the size and resolution of the vehicle pose candidate space, and by that of the fixed-size azimuth-elevation grid. The automated grid size/resolution selection of IEKF SGL reduces the computation load to half that of BF SGL. The LL has the lowest computation time because it only processes the few data points corresponding to the selected landmarks.

Table 2: Comparison of performance measures for different localization algorithms

Max of 1σ Cross Track Error Bound		Computation Time (single lap)
Localization Methods	Value	
Landmark-based	0.05 m*	4 s
Brute Force SGB	0.05 m**	20 s
IEKF SGB	**0.08 m**	**9 s**

* : does not account for feature extraction and data association errors

** : does not account for gridding errors

5 Conclusion

In this paper, we developed a tightly-integrated, spherical-grid-based LiDAR/IMU localization algorithm which provides an analytical bound on the root-sum-squared positioning error. The algorithm leverages signal quantization theory to limit the amount of LiDAR data processing while minimizing loss of information. In addition, we designed an iterative extended Kalman filter (IEKF) to estimate vehicle pose and predict pose uncertainty. Experimental evaluation was performed in an indoor environment using data collected by a rover on tracks traveling 80 times along a same, repeated trajectory. Testing showed that IEKF spherical grid-based localization provides a more realistic positioning error bound than landmark-based and brute-force spherical grid localization approaches without causing significant additional computation costs.

Appendix A –Discrete-Time Equations of IMU

We use the Van Loan algorithm to determine the discrete-time state propagation matrix $\mathbf{\Phi}_k$ based on the continuous-time matrices $\mathbf{F}$ and $\mathbf{w}$ [15]. The following equations are the discrete-time form of terms in equation (10).

$$ {}^{\mathrm{B}}\tilde{\boldsymbol{\omega}}_k^{\mathrm{IB}} = [\mathbf{I} + \mathbf{S}_g + \mathbf{M}_g]\, {}^{\mathrm{B}}\boldsymbol{\omega}_k^{\mathrm{IB}} + \mathbf{b}_{g,k} + \boldsymbol{\upsilon}_{g,k} \tag{A.1} $$

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

$$ {}^{\mathrm{B}}\tilde{\mathbf{f}}_k^{\mathrm{I}} = [\mathbf{I} + \mathbf{S}_a + \mathbf{M}_a]\, {}^{\mathrm{B}}\mathbf{f}_k^{\mathrm{I}} + \mathbf{b}_{a,k} + \boldsymbol{\upsilon}_{a,k} \tag{A.3} $$

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

$$ \mathbf{b}_{g,k+1} = \mathrm{e}^{-\frac{T_S}{\tau_g}} \mathbf{b}_{g,k} + \mathbf{n}_{a,k} \tag{A.5} $$

$$ \mathbf{b}_{a,k+1} = \mathrm{e}^{-\frac{T_S}{\tau_a}} \mathbf{b}_{a,k} + \mathbf{n}_{a,k} \tag{A.6} $$

Appendix B – Linearized IMU and LiDAR Measurement Equations Coefficients

The coefficient matrices corresponding to IMU measurements in equation (10) can be defined as [14]:

$$F_{V2T} = \begin{bmatrix} 0 & \frac{1}{R+h} & 0 \\ \frac{1}{R+h} & 0 & 0 \\ 0 & \frac{-\tan(\lambda)}{R+h} & 0 \end{bmatrix} \tag{B.1}$$

$$\mathbf{F}_{H2V} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{2g_0}{R} \end{bmatrix} \tag{B.2}$$

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 (21) are:

$$\mathbf{F}_{r,x} = \begin{bmatrix} -\frac{{}^{\mathrm{N}}p_{E,1} - x_E}{\left\| {}^{\mathrm{N}}\mathbf{p}_1 - \bar{\mathbf{x}}_{EN} \right\|} & \cdots & -\frac{{}^{\mathrm{N}}p_{E,m} - x_E}{\left\| {}^{\mathrm{N}}\mathbf{p}_m - \bar{\mathbf{x}}_{EN} \right\|} \\ -\frac{{}^{\mathrm{N}}p_{N,1} - x_N}{\left\| {}^{\mathrm{N}}\mathbf{p}_1 - \bar{\mathbf{x}}_{EN} \right\|} & \cdots & -\frac{{}^{\mathrm{N}}p_{N,m} - x_N}{\left\| {}^{\mathrm{N}}\mathbf{p}_m - \bar{\mathbf{x}}_{EN} \right\|} \\ 0 & \cdots & 0 \end{bmatrix}^T \tag{B.3}$$

$$\mathbf{F}_{\theta,x} = \begin{bmatrix} \frac{{}^{\mathrm{N}}p_{N,1} - x_N}{\left\| {}^{\mathrm{N}}\mathbf{p}_1 - \bar{\mathbf{x}}_{EN} \right\|^2} & \cdots & \frac{{}^{\mathrm{N}}p_{N,m} - x_N}{\left\| {}^{\mathrm{N}}\mathbf{p}_m - \bar{\mathbf{x}}_{EN} \right\|^2} \\ -\frac{{}^{\mathrm{N}}p_{E,1} - x_E}{\left\| {}^{\mathrm{N}}\mathbf{p}_1 - \bar{\mathbf{x}}_{EN} \right\|^2} & \cdots & -\frac{{}^{\mathrm{N}}p_{E,m} - x_E}{\left\| {}^{\mathrm{N}}\mathbf{p}_m - \bar{\mathbf{x}}_{EN} \right\|^2} \\ 0 & \cdots & 0 \end{bmatrix}^T \tag{B.4}$$

where ${}^{\mathrm{N}}\mathbf{p} = [p_E \quad p_N]^T$ and $\bar{\mathrm{x}}_{EN} = [x_E \quad x_N]^T$

$$F_{\theta,e} = \begin{bmatrix} 0 & 0 & 1 \\ & \vdots & \\ 0 & 0 & 1 \end{bmatrix}_{m\times 3} \tag{B.5}$$

References

[1] A. Hassani, M. Joerger, "A New Point-Cloud-Based LiDAR/IMU Localization Method with Uncertainty Evaluation," Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri, September 2021, pp. 636-651.

[2] 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.

[3] 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.

[4] M. Joerger, 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.

[5] 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.

[6] A. Censi, "An ICP variant using a point-to-line metric," 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 19-25, doi: 10.1109/ROBOT.2008.4543181.

[7] 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

[8] K. Sayood, "Introduction to data compression," Morgan Kaufmann Publishers, San Francisco, 2000.

[9] J. Max, "Quantizing for minimum distortion," IRE Transactions on Information Theory, vol. 6, no. 1, pp. 7-12, March 1960, doi: 10.1109/TIT.1960.1057548.

[10] M. Sabin, R. Gray, "Global convergence and empirical consistency of the generalized Lloyd algorithm," IEEE Transactions on Information Theory, vol. 32, no. 2, pp. 148-155, March 1986, doi: 10.1109/TIT.1986.1057168.

[11] Q. Du, M. Gunzburger, "Grid generation and optimization based on centroidal Voronoi tessellations", Applied Mathematics and Computation, vol. 133, no. 2-3, pp. 591-607, 2002. doi: 10.1016/s0096-3003(01)00260-0.

[12] V. Madisetti, "The digital signal processing handbook," 2010 Boca Raton, FL, CRC Press.

[13] L. Zheng, R. Gallager, "Principles of Digital Communications I," Massachusetts Institute of Technology: MIT OpenCourseWare, Fall 2006.

[14] D. Titterton, J. Weston, "Strapdown inertial navigation technology," Stevenage: The Inst. of Engineering and Technology, 2009.

[15] R. Brown, P. Hwang, "Introduction to Random Signals and Applied Kalman Filtering. Hoboken," NJ: John Wiley and Sons, 2012.

[16] O. Garcia Crespillo, "GNSS/INS Kalman Filter Integrity Monitoring with Uncertain Time Correlated Error Processes," PhD dissertation, EPFL, 2022.

[17] Ç. Tanıl, S. Khanafseh, M. Joerger and B. Pervan, "An INS Monitor to Detect GNSS Spoofers Capable of Tracking Vehicle Position," IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 1, pp. 131-143, Feb. 2018, doi: 10.1109/TAES.2017.2739924.