IMU-based Online Multi-lidar Calibration

Sandipan Das[1,2], Bengt Boberg[2], Maurice Fallon[3], Saikat Chatterjee[1]

Modern autonomous systems typically use several sensors for perception. For best performance, accurate and reliable extrinsic calibration is necessary. In this research, we propose a reliable technique for the extrinsic calibration of several lidars on a vehicle without the need for odometry estimation or fiducial markers. First, our method generates an initial guess of the extrinsics by matching the raw signals of IMUs co-located with each lidar. This initial guess is then used in ICP and point cloud feature matching which refines and verifies this estimate. Furthermore, we can use observability criteria to choose a subset of the IMU measurements that have the highest mutual information — rather than comparing all the readings. We have successfully validated our methodology using data gathered from Scania test vehicles.

I. Introduction

For safe navigation and sensor redundancy, autonomous vehicles need numerous sensors to ensure 360° sensing coverage. For precise fusion, it is crucial to establish the accurate mounting location for each of the vehicle's sensors — a process known as extrinsic calibration. Without the proper extrinsic calibration parameters, it is impossible to fuse the sensing data into a single reference frame. In this work, we concentrated on multi-lidar system extrinsic calibration, though the same principle might also be applied to camera or radar extrinsic calibration.

There are two main types of multi-lidar extrinsic calibration algorithms: target-based and target-less. Target-based calibration refers to putting fiducial markers in the shared field-of-view (FoV) of the lidars and mutually matching the markers. Target-less calibration is carried out by comparing the estimated states of the sensors for initialization and refined using feature-matching. In another set of target-less methods, both state and the calibration parameters are estimated by solving a joint optimization step where the reprojection error between the tracked features is minimized. Since the sensors may not have an overlapping FoV, target-less calibration is preferable as it is more general than target-based methods. It is widely recognized that not all motion segments produce enough observable information for matching states used for the initialization of extrinsic calibration. Because of this identifying any degenerate motion segments is also important to discard information that is not helpful for extrinsic calibration initialization.

A. Motivation

As seen in Fig. 1 and Fig. 2, the mobile platforms we employed for our studies have multiple lidars. Therefore, it is essential to confirm that the sensors are correctly calibrated prior to any autonomous run. To properly excite the different degrees of freedom, motion-based calibration requires aggressive driving maneuvers which can cause state estimation drift. Moreover, state estimation itself adds additional complexity to the whole tool chain and it does not provide independent verification for the estimated extrinsics. Finally, we also need a process to verify the quality of the estimated extrinsics.

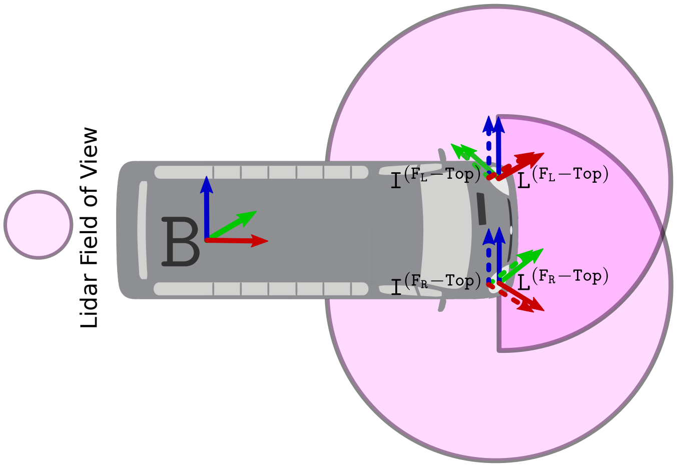

Lidar Field of View
B
$\mathtt{I}^{(\mathtt{F_L}-\mathtt{Top})}$
$\mathtt{L}^{(\mathtt{F_L}-\mathtt{Top})}$
$\mathtt{I}^{(\mathtt{F_R}-\mathtt{Top})}$
$\mathtt{L}^{(\mathtt{F_R}-\mathtt{Top})}$

Fig. 1: Illustration of 2 lidars with co-located IMUs positioned at the corners of our data collection vehicle. The vehicle base frame $\mathtt{B}$, is located at the center of the rear axle. The sensor frames of the lidars (solid lines) are: $\mathtt{L}^{(\mathtt{F_L}-\mathtt{Top})}$ and $\mathtt{L}^{(\mathtt{F_R}-\mathtt{Top})}$ whereas, the sensor frames of the IMUs (dotted lines) are: $\mathtt{I}^{(\mathtt{F_L}-\mathtt{Top})}$ and $\mathtt{I}^{(\mathtt{F_R}-\mathtt{Top})}$. $(\mathtt{F_L})$: front-left and $(\mathtt{F_R})$: front-right.

B. Contribution

We present the following contributions:

- A novel observability-aware, target-less extrinsic calibration algorithm to calibrate multiple lidars using co-located IMUs without the need for any lidar odometry estimation and extrinsics rotation initialization.
- Online extrinsics refinement based on line feature matching and online bias compensation of the IMU signals to improve the quality of the extrinsic calibration algorithm.
- Verification of our method in real-time based on plane feature association with datasets collected from Scania test vehicles in urban driving scenarios.
- Our implementation for the IMU-IMU calibration with observability analysis on a custom-made test board and sample datasets is open-sourced on Github[4].

II. Related Work

Extrinsic sensor calibration is a well studied area. In our discussion, we briefly review the relevant literature and motivate our choice of methods.

[1] KTH EECS, Sweden. {sandipan,sach}@kth.se
[2] Scania, Sweden. {sandipan.das, bengt.boberg}@scania.com
[3] ORI, University of Oxford, UK. mfallon@robots.ox.ac.uk
[4] https://github.com/mrsandipandas/imu-calibration

A. Target-based calibration

Target-based extrinsic calibration is performed by matching known markers or fiducials placed in the common FoV of multiple sensors. Researchers have explored target-based between camera and lidar using known geometric shapes like checkerboards, triangles, and diamonds [1] or geometric feature correspondences [2]. To calibrate a multi-lidar system, geometric elements including points, lines, and planes were retrieved and matched in [3], [4]. The most important aspect of a target-based system is there being overlapping FoV.

B. Target-less calibration

Target-less extrinsic calibration based on the Hand-Eye method [5], [6], [7] is an extensively studied topic. The term "Hand-Eye" refers to early motion-based calibration research that estimated the motion of the gripper (hand) and the camera (eye) while constricting their poses with a fixed rigid body transformation. Multi-camera extrinsic calibration with Hand-Eye method has been explored in works like Camodocal [8] and recently extended to multi-lidar extrinsic calibration [9].

Another technique for target-less calibration is pose alignment with Kabsch method [10], where the poses are obtained from lidar odometry [11], [12] in the corresponding sensor frame. Recent works like CROON [13] and adaptive voxelization [14] improve the pose alignment with additional feature matching techniques. In Kalibr [15] the authors automatically detected sets of measurements from which they could identify an observable parameter space and then performed a maximum likelihood estimate (MLE) by minimizing the errors between landmark observations and their known correspondences. They also discarded parameter updates for numerically unobserved directions and degenerate scenarios. In [16], a global bundle adjustment was performed to minimize the reprojection error between the tracked features to estimate the camera-lidar extrinsics. A similar principle can be applied for multiple lidar extrinsic calibration if the sensors have an overlapping FoV. However, this method would still require lidar odometry estimates, which is a computationally expensive process.

Because we have integrated IMUs within our lidars (with factory calibrations), we aim to use them to create a multi-IMU (MIMU) configuration to estimate initial extrinsics estimates without the need for any pose alignment. Kalibr was extended towards MIMU calibration [17] by matching poses fitted to a B-spline and further constraining motion with known image landmarks. In MIMC-VINS [18], an efficient multi-state constraint Kalman filter was used to jointly propagate all IMU states while enforcing rigid body constraints between the IMUs during the filter update stage, which produced the MIMU calibration. Unlike prior works, we used the fundamental property that a rigid body's angular velocity is constant around every point of the body to formulate our MIMU calibration. We used MLE to match the raw angular velocity between the IMU(s) embedded within the lidar(s) in accordance with the signal-to-signal match principle [19].

We then selected measurements that have contain sufficient signal excitation which is known as observability-aware criteria [20], [21], [22], [23]. We then maximized mutual information between the angular velocity signals to identify relevant motion segments. This method can be extended to other modalities and does not require overlapping FoV between the sensors.

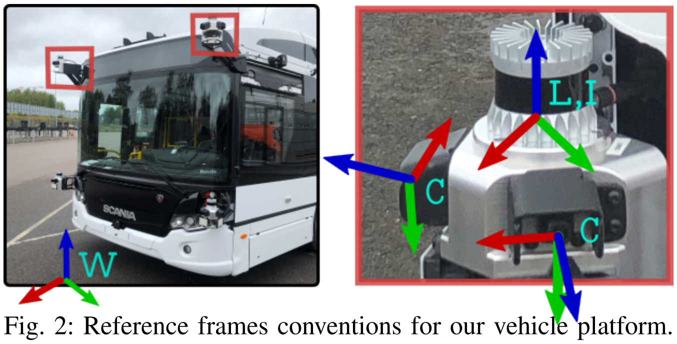


Fig. 2: Reference frames conventions for our vehicle platform. The world frame $\mathtt{W}$ is a fixed frame, while the base frame $\mathtt{B}$, as shown in Fig. 1, is located at the rear axle center of the vehicle. Each sensor unit contains the two optical frames $\mathtt{C}$, an IMU frame, $\mathtt{I}$, and a lidar frame $\mathtt{L}$.

III. Problem Statement

A. Sensor platform and reference frames

We evaluate our proposed method using data collected from a Scania test vehicle. The sensor platform of the vehicle with its illustrative sensor FoV and corresponding reference frames is shown in Fig. 1 and Fig. 2 respectively. Each sensor housing contains a lidar with an embedded IMU and two cameras. Note that the cameras were not used in this work.

Now we describe the necessary notation and reference frames used in our system. The vehicle base frame is denoted as $\mathtt{B}$ and the world frame is denoted as $\mathtt{W}$. Sensor readings from the lidars and IMUs are represented in their respective sensor frames as $\mathtt{L}^{(k)}$, and $\mathtt{I}^{(k)}$ respectively, where, $k \in [\mathtt{F_L}-\mathtt{Top}, \mathtt{F_R}-\mathtt{Top}]$ denotes the location of the sensor in the vehicle corresponding to front-left-top and front-right-top respectively. In our discussions, the transformation matrix is denoted as, $\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathrm{SE}(3)$ and $\mathbf{T}_{\mathtt{AB}}$ denotes the transformation matrix of frame $\mathtt{B}$ *wrt* frame $\mathtt{A}$.

B. Problem formulation

The primary goal is to estimate the extrinsic calibration between multiple lidar sensors, $\mathbf{T}_{\mathtt{L}^{(k_i)}\mathtt{L}^{(k_j)}}$, in real-time without the need for any calibration targets or odometry estimation. The system is initialized ($\mathbf{T}_{\mathtt{init}}$) only with rough translation component parameters from the vehicle and sensor CAD models. Finally, the extrinsics are also verified in real-time to ensure robustness.

IV. Methodology

Our method starts by matching the noise compensated IMU signals between the base frame, $\mathtt{B}$, and the IMU frame, $\mathtt{I}^{(k)}$, which estimates $\hat{\mathbf{T}}_{\mathtt{IMU}}$. Then a GICP-based [24] point cloud alignment is performed with the initial estimate to compute $\hat{\mathbf{T}}_{\mathtt{GICP}}$, which is further refined by feature matching to estimate $\hat{\mathbf{T}}_{\mathtt{Refined}}$. An additional online verification is performed to ensure the correctness of the calibration process.

Thus, $\mathbf{T}_{\mathtt{BL}^{(k)}} \simeq \hat{\mathbf{T}}_{\mathtt{Refined}} \times \hat{\mathbf{T}}_{\mathtt{GICP}} \times \hat{\mathbf{T}}_{\mathtt{IMU}} \times \mathbf{T}_{\mathtt{init}}$. For notional brevity we consider one of the lidars to be base and discuss a general calibration routine between the base frame, $\mathtt{B}$, and the lidar frame, $\mathtt{L}^{(k)}$.

A. IMU-based initial estimate

1) IMU initialization: The IMU measurements are in their corresponding sensor frame. For gravity alignment, we use equations eq. 25, 26 from [25] to estimate roll and pitch and obtain $\mathtt{R}_{\mathtt{WB}}$ after collecting IMU data when the vehicle is static for a few seconds.

2) IMU sensor model: We have considered a 6-DoF (degree of freedom) IMU such that it has a 3-axis accelerometer and 3-axis gyroscope. The IMU sensor data in its corresponding sensor frame can be represented as,

$$\begin{bmatrix} \boldsymbol{\omega}_{\mathtt{I}_t} \\ \mathbf{a}_{\mathtt{I}_t} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_{\mathtt{I}_t} \\ \hat{\mathbf{a}}_{\mathtt{I}_t} - \mathbf{R}_{\mathtt{WI}_t} \mathbf{g}_{\mathtt{W}} \end{bmatrix} + \begin{bmatrix} \mathbf{b}^{\omega}_{\mathtt{I}_t} \\ \mathbf{b}^{a}_{\mathtt{I}_t} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}^{\omega}_{\mathtt{I}_t} \\ \boldsymbol{\eta}^{a}_{\mathtt{I}_t} \end{bmatrix}. \quad (1)$$

Here, $[\boldsymbol{\omega}_{\mathtt{I}_t} \in \mathbb{R}^{3\times1}, \mathbf{a}_{\mathtt{I}_t} \in \mathbb{R}^{3\times1}]$ represents the measured angular velocity and linear acceleration at timestamp t and $[\hat{\boldsymbol{\omega}}_{\mathtt{I}_t}, \hat{\mathbf{a}}_{\mathtt{I}_t}]$ represent the latent ideal angular velocity and linear acceleration respectively. $\mathbf{b}^{\omega}_{\mathtt{I}_t} \in \mathbb{R}^{3\times1}$ and $\mathbf{b}^{a}_{\mathtt{I}_t} \in \mathbb{R}^{3\times1}$ represent the gyro and accelerometer biases which change with time and other factors like temperature. $\boldsymbol{\eta}^{\omega}_{\mathtt{I}_t} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\omega})$ and $\boldsymbol{\eta}^{a}_{\mathtt{I}_t} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{a})$ are the additive zero-mean white Gaussian noises for gyroscope and accelerometer with covariance $\boldsymbol{\Sigma}_{\omega} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Sigma}_{a} \in \mathbb{R}^{3\times3}$ respectively. $\mathbf{g}_{\mathtt{W}} \in \mathbb{R}^{3\times1}$, represents the gravity vector in the world frame, $\mathtt{W}$, and $\mathbf{R}_{\mathtt{BW}}$ represent the gravity alignment rotation matrix.

3) IMU bias characterization: IMU biases are estimated by collecting data sequences at rest. The accelerometer bias is unchanged till the next rest state detection whereas the gyro bias is tracked online with a Kalman filter [26]. The biases are recomputed whenever a rest state is detected which limits the bias covariance growth.

a) IMU state propagation: The IMU dynamical model based on angular velocity in quaternion form is shown as,

$$\mathbf{q}_{\mathtt{BI}_t} = \left[\cos\left(\frac{\Delta t}{2} \left\| \boldsymbol{\omega}_{\mathtt{I}_t} \right\| \right) \mathbf{I}_4 + \frac{\sin\left(\frac{\Delta t}{2} \left\| \boldsymbol{\omega}_{\mathtt{I}_t} \right\| \right)}{\left\| \boldsymbol{\omega}_{\mathtt{I}_t} \right\|} \boldsymbol{\Omega}_{\mathtt{I}_t} \right] \mathbf{q}_{\mathtt{BI}_{t-1}} \quad (2)$$

and,

$$\boldsymbol{\Omega}_{\mathtt{I}_t} = \begin{bmatrix} 0 & [\boldsymbol{\omega}_{\mathtt{I}_t}]_z & -[\boldsymbol{\omega}_{\mathtt{I}_t}]_y & [\boldsymbol{\omega}_{\mathtt{I}_t}]_x \\ -[\boldsymbol{\omega}_{\mathtt{I}_t}]_z & 0 & [\boldsymbol{\omega}_{\mathtt{I}_t}]_x & [\boldsymbol{\omega}_{\mathtt{I}_t}]_y \\ [\boldsymbol{\omega}_{\mathtt{I}_t}]_y & -[\boldsymbol{\omega}_{\mathtt{I}_t}]_x & 0 & [\boldsymbol{\omega}_{\mathtt{I}_t}]_z \\ -[\boldsymbol{\omega}_{\mathtt{I}_t}]_x & -[\boldsymbol{\omega}_{\mathtt{I}_t}]_y & -[\boldsymbol{\omega}_{\mathtt{I}_t}]_z & 0 \end{bmatrix}. \quad (3)$$

Δt denotes the sampling period of the IMU data. We used the Madgwick filter [27] to estimate the refined rotation which minimizes the difference between the measured acceleration and the aligned gravity vector using a gradient descent algorithm as,

$$\hat{\mathbf{q}}_{\mathtt{BI}_t} = \underset{\mathbf{q}_{\mathtt{BI}_t} \in \mathbb{R}^{4\times1}}{\arg\min} \left(\; \tilde{\mathbf{q}}_{\mathtt{BI}_t} \otimes \mathbf{R}^{-1}_{\mathtt{BW}} \mathbf{g}_{\mathtt{W}} \otimes \mathbf{q}_{\mathtt{BI}_t} - \hat{\mathbf{a}}_{\mathtt{I}_t} \; \right), \quad (4)$$

where, $\tilde{\mathbf{q}}$ represents the quaternion conjugate and $\otimes$ is the quaternion product operator. Note that we use the compensated accelerometer signals after bias compensation which is discussed in IV-A3c.

b) Rest state: The rest state is detected if the difference between the norm of the aligned gravity vector and the acceleration vector is less than a predefined threshold, τ for at least more than 2 seconds. Thus,

$$\mathtt{Rest\ detected} = \| \; \tilde{\hat{\mathbf{q}}}_{\mathtt{BI}_t} \otimes \mathbf{g}_{\mathtt{B}} \otimes \hat{\mathbf{q}}_{\mathtt{BI}_t} - \hat{\mathbf{a}}_{\mathtt{I}_t} \; \|_2 \leq \tau \quad (5)$$

c) Accelerometer bias estimation: The accelerometer bias is estimated by computing the mean of the signal at rest state and the standard deviation gives us the co-variance of the white Gaussian noise. We recompute these parameters whenever the rest state is detected and keep them unchanged until the next rest state detection. If the rest period duration is N seconds then,

$$\begin{aligned} \mathbf{b}^{a} &= \frac{\Delta t}{N} \sum_{i=1}^{N} (\mathbf{a}_{\mathtt{I}_i} - \hat{\mathbf{R}}_{\mathtt{BI}_t} \mathbf{R}^{-1}_{\mathtt{BW}} \mathbf{g}_{\mathtt{W}}), \\ \boldsymbol{\Sigma}_{a} &= \mathtt{diag} \left(\frac{\Delta t}{N} \sum_{i=1}^{N} |\mathbf{a}_{\mathtt{I}_i} - \mathbf{b}^{a}|^2 \right), \\ \implies \hat{\mathbf{a}}_{\mathtt{I}_t} &= \mathbf{a}_{\mathtt{I}_t} + \hat{\mathbf{R}}_{\mathtt{BI}_t} \mathbf{R}^{-1}_{\mathtt{BW}} \mathbf{g}_{\mathtt{W}} - \mathbf{b}^{a} - \mathcal{N}(0, \boldsymbol{\Sigma}_{a}). \end{aligned} \quad (6)$$

d) Gyro bias estimation: The initialization of the gyro bias is carried out in a similar fashion as we did for the accelerometer and recomputed whenever the rest state is detected. Thus,

$$\begin{aligned} \text{System State}, \hat{\mathbf{x}}_0 = \mathbf{b}^{\omega}_{\mathtt{I}_0} &= \frac{\Delta t}{N} \sum_{i=1}^{N} (\boldsymbol{\omega}_{\mathtt{I}_i}) \\ \text{Measurement Noise}, \mathbf{W}_0 &= \mathcal{N}(0, \boldsymbol{\Sigma}_{\omega}) \\ \text{Process Noise}, \mathbf{Q} &= (0.05^{\circ}/\mathtt{sec})^2 \mathbf{I}_{3\times3} \\ \text{Initial Covariance}, \mathbf{P}_0 &= (0.5^{\circ}/\mathtt{sec})^2 \mathbf{I}_{3\times3}. \end{aligned} \quad (7)$$

After that we use a Kalman filter to track the gyroscope bias as a state. The system is modeled by,

$$\begin{aligned} \mathbf{x}_t &= \hat{\mathbf{x}}_{t-1} + \mathcal{N}(0, \mathbf{Q}) \\ \mathbf{y}_t &= \hat{\mathbf{R}}_{\mathtt{BI}_t} \mathbf{x}_t + \mathbf{W}_t. \end{aligned} \quad (8)$$

The standard Kalman filter update equations become,

$$\begin{aligned} \mathbf{P}_t &= \mathbf{P}_{t-1} + \mathbf{Q} \\ \mathbf{K}_t &= \mathbf{P}_t \hat{\mathbf{R}}_{\mathtt{BI}_t} (\boldsymbol{\Sigma}_{\omega} + \hat{\mathbf{R}}_{\mathtt{BI}_t} \mathbf{P}_t \hat{\mathbf{R}}^{T}_{\mathtt{BI}_t})^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t-1} + \mathbf{K}_t (\mathbf{y}_t - \hat{\mathbf{R}}_{\mathtt{BI}_t} \hat{\mathbf{x}}_{t-1}) \\ \mathbf{P}_t &= \mathbf{P}_t - \mathbf{K}_t \hat{\mathbf{R}}_{\mathtt{BI}_t} \mathbf{P}_t. \end{aligned} \quad (9)$$

Thus after estimating the biases we can rearrange eq. 1 to compute $\hat{\boldsymbol{\omega}}_{\mathtt{I}_t}$.

4) IMU signal matching:

a) Rotation estimate: The extrinsics of $\mathtt{L}^{(k)}$ *wrt* $\mathtt{B}$ frame is closely related to with the extrinsics of $\mathtt{I}^{(k)}$ *wrt* $\mathtt{B}$ frame as they are co-located with known extrinsics from sensor supplier. A rigid body has uniform angular velocity throughout the body. The angular velocities of all the IMU sensors in our sensor arrangement must be equal if their orientations are the same because they are all securely affixed to the vehicle. Let, $[\hat{\boldsymbol{\omega}}_{\mathtt{I}^{(k)}_i}, \hat{\mathbf{a}}_{\mathtt{I}^{(k)}_i}]$ and $[\hat{\boldsymbol{\omega}}_{\mathtt{B}_i}, \hat{\mathbf{a}}_{\mathtt{B}_i}]$ be the estimated angular velocity and linear acceleration of the k^{th} IMU and the base IMU at timestamp t_i after bias compensation. By removing the k^{th} superscript for brevity, our optimization problem becomes,

$$\begin{aligned} \mathbf{R}^{\star}_{\mathtt{BI}} &= \arg\min_{\mathbf{R}_{\mathtt{BI}}} \sum_{i=1}^{N} \|\mathbf{R}_{\mathtt{BI}}\hat{\boldsymbol{\omega}}_{\mathtt{B}_i} - \hat{\boldsymbol{\omega}}_{\mathtt{I}_i}\|^2_{\sum_i}, \\ \textit{s.t.} \quad & \mathbf{R}_{\mathtt{BI}}\mathbf{R}^T_{\mathtt{BI}} = \mathbf{I}_3, \end{aligned} \tag{10}$$

which, can be simply solved with Kabsch alignment [10].

b) Translation estimate: For the acceleration components we account for the Coriolis forces as illustrated in Fig. 3 and equate the translation components as,

$$\begin{aligned} (\mathbf{R}^{\star}_{\mathtt{BI}})^{-1}\hat{\mathbf{a}}_{\mathtt{I}} &= \hat{\mathbf{a}}_{\mathtt{B}} + \underbrace{\hat{\boldsymbol{\omega}}_{\mathtt{B}} \times (\hat{\boldsymbol{\omega}}_{\mathtt{B}} \times \mathbf{t}_{\mathtt{BI}})}_{\text{Centrifugal force}} + \underbrace{\dot{\hat{\boldsymbol{\omega}}}_{\mathtt{B}} \times \mathbf{t}_{\mathtt{BI}}}_{\text{Euler force}} \\ &= \hat{\mathbf{a}}_{\mathtt{B}} + [\hat{\boldsymbol{\omega}}_{\mathtt{B}}]^2_{\times}\mathbf{t}_{\mathtt{BI}} + [\dot{\hat{\boldsymbol{\omega}}}_{\mathtt{B}}]_{\times}\mathbf{t}_{\mathtt{BI}}, \end{aligned} \tag{11}$$

where, $[.]_{\times}$ is a skew-symmetric matrix and $[\boldsymbol{a}]_{\times}\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$.

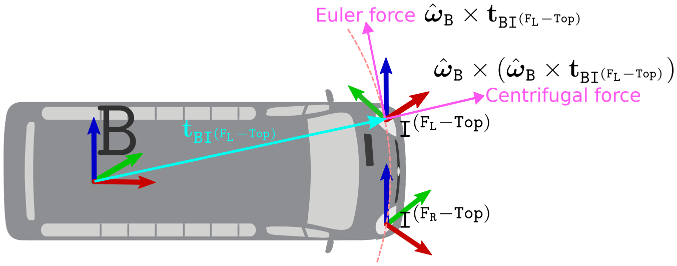

Euler force $\dot{\hat{\boldsymbol{\omega}}}_{\mathtt{B}} \times \mathbf{t}_{\mathtt{BI}(\mathtt{F_L-Top})}$
$\hat{\boldsymbol{\omega}}_{\mathtt{B}} \times (\hat{\boldsymbol{\omega}}_{\mathtt{B}} \times \mathbf{t}_{\mathtt{BI}(\mathtt{F_L-Top})})$
Centrifugal force
B
$\mathbf{t}_{\mathtt{BI}(\mathtt{F_L-Top})}$
$\mathtt{I}^{(\mathtt{F_L-Top})}$
$\mathtt{I}^{(\mathtt{F_R-Top})}$

Fig. 3: Illustration of IMU signal transformation between the base frame, $\mathtt{B}$, and the IMU frame, $\mathtt{I}^{(\mathtt{F_L-Top})}$.

Since, we have already computed $\mathbf{R}^{\star}_{\mathtt{BI}}$, our optimization problem for the translation component becomes,

$$\mathbf{t}^{\star}_{\mathtt{BI}} = \arg\min_{\mathbf{t}_{\mathtt{BI}}} \sum_{i=1}^{N} \| \underbrace{([\hat{\boldsymbol{\omega}}_{\mathtt{B}_i}]^2_{\times} + [\dot{\hat{\boldsymbol{\omega}}}_{\mathtt{B}_i}]_{\times})}_{\mathbf{A}_i} \mathbf{t}_{\mathtt{BI}} - \underbrace{(\mathbf{R}^{-1\star}_{\mathtt{BI}}\hat{\boldsymbol{a}}_{\mathtt{I}_i} - \hat{\boldsymbol{a}}_{\mathtt{B}_i})}_{\mathbf{B}_i} \|^2, \tag{12}$$

where, $\mathbf{A} = [\mathbf{A}_0 \ \mathbf{A}_1 \ \dots \ \mathbf{A}_n]^T$, $\mathbf{B} = [\mathbf{B}_0 \ \mathbf{B}_1 \ \dots \ \mathbf{B}_n]^T$ and $\mathbf{x} = \mathbf{t}_{\mathtt{BI}} \ \forall i \in [1, N]$. This is a system of linear equations of the form, $\mathbf{Ax} = \mathbf{B}$, which can be solved using a least-square approach as,

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{B}\|^2_{\mathbf{\Sigma}}, \tag{13}$$

where, $\mathbf{\Sigma}$ denotes the co-variance of the residual. The main approach used to tackle these problems [28] involves solving a series of approximations to the original problem repeatedly by linearizing as, $F(\mathbf{x}+\Delta\mathbf{x}) \approx F(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}$, where, J being the jacobian of $F(\mathbf{x})$. In this way $\hat{\mathbf{x}}$ is updated in the current iteration as, $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} \boxplus \mathbf{\Delta x}$, where $\boxplus$ is an addition operator in the manifold and the problem becomes,

$$\mathbf{\Delta x}^{\star} = \arg\min_{\mathbf{\Delta x}} \frac{1}{2}\|(\mathbf{A}\hat{\mathbf{x}} - \mathbf{B}) + \mathbf{A\Delta x}\|^2_{\mathbf{\Sigma}}. \tag{14}$$

The optimal solution is given by,

$$\underbrace{(\mathbf{J}^{\top}\mathbf{\Sigma}^{-1}\mathbf{J})}_{\text{Fisher information matrix}} \quad \mathbf{\Delta x} = -\mathbf{J}^{\top}\mathbf{\Sigma}^{-1}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{B}). \tag{15}$$

In practice, since we can take a reliable initial estimate for the translation component from the CAD parameters (within ≈10 cm), we search for $\mathbf{x}^{\star}$ only in a local neighborhood within reasonable bounds. Thus, we solve a bounded variable least squares problem as,

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{L} \leq \mathbf{x} \leq \mathbf{U}} \|\mathbf{Ax} - \mathbf{B}\|^2_{\mathbf{\Sigma}}, \tag{16}$$

where, $\mathbf{L}$ and $\mathbf{U}$ are the lower and upper bounds of $\mathbf{x}$ respectively. Thus the solution space in Eq. 14 is modified with an additional constraint as,

$$\begin{aligned} \mathbf{\Delta x}^{\star} &= \arg\min_{\mathbf{\Delta x}} \frac{1}{2}\|(\mathbf{A}\hat{\mathbf{x}} - \mathbf{B}) + \mathbf{A\Delta x}\|^2_{\mathbf{\Sigma}}, \\ \textit{s.t.} \ & \mathbf{L} \leq \hat{\mathbf{x}} + \mathbf{\Delta x} \leq \mathbf{U}. \end{aligned} \tag{17}$$

c) Observability analysis: Not all vehicle motions are suitable for exciting the degrees of freedom to allow calibration. As a result, it's important to identify suitable motion segments for information-aware calibration updates. We captured this information by comparing the angular velocity between $\mathtt{I}^{(k)}$ and $\mathtt{B}$ frame based on Eq.10 which can also be solved iteratively with the update step as,

$$\underbrace{\left(\sum_i \mathbf{J}_i^{\top}\Sigma_i^{-1}\mathbf{J}_i\right)}_{\text{Fisher information matrix}} \mathbf{\Delta R}_{\mathtt{BI}} \overset{Eq.\ 15}{=} -\sum_i \mathbf{J}_i^{\top}\Sigma_i^{-1}(\hat{\mathbf{R}}_{\mathtt{BI}}\boldsymbol{\omega}_{\mathtt{B}_i} - \boldsymbol{\omega}_{\mathtt{I}_i}) \tag{18}$$

where, $\mathbf{\Sigma}_i = \text{cov}(\hat{\mathbf{R}}_{\mathtt{BI}}\boldsymbol{\omega}_{\mathtt{B}_i} - \boldsymbol{\omega}_{\mathtt{I}_i})$ and $\mathbf{J}_i = \boldsymbol{\omega}^T_{\mathtt{B}_i}$. The Fisher information matrix, $\mathcal{I}_{N \times N}$ captures all the information contained in the measurements. We do the processing after arranging the data in a batch size of N. We perform a Singular Value Decomposition of $\mathcal{I}_{N \times N}$ for each batch as:

$$\mathcal{I}_{N \times N} = \mathbf{USU}^T, \tag{19}$$

where, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2 \dots, \mathbf{u}_N]$ and $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2 \dots, \sigma_N)$ is a diagonal matrix of singular values in decreasing order. Information about the data in the batch is indicated by the value of the minimal singular value. If the minimal singular value exceeds a certain threshold (design decision), we may say that there is sufficient excitation in the batch of data to allow for extrinsics computation and hence chosen for calibration.

B. Point cloud based alignment

a) Initial match: Based on our sensor setup we first filtered the point clouds with a box filter to retain the region comprising of maximum overlap between the two point clouds for faster processing. For our setup, we filtered the point clouds between $[0, -10, -10]m$ and $[50, 10, 10]m$. Then the initial guess from the IMU-based matching was used as a prior for the GICP-based [24] matching on the filtered point clouds.

b) Line and plane feature extraction and matching: After obtaining improved extrinsic parameters from GICP we further refined the extrinsics using line feature extraction and matching. First, segmentation [11] and local curvature [29] of each point in the filtered clouds are evaluated. The points with the highest and lowest curvature are allocated to the sets of line and plane clouds, respectively. Then, we fit the line and plane model to the corresponding line and plane cloud sets using PROSAC [30] which exploits the spatial coherence of the line and plane points. Thus, for each lidar point cloud, we obtained the line and plane feature set as $\mathcal{F}^L$ and $\mathcal{F}^P$, where a line and plane are modeled as,

$$\begin{aligned}\mathcal{F}^L_{\text{L}^{(k)}} &:= \left\{\langle \hat{\mathbf{n}}^{(l)}_{\text{L}^{(k)}}, \hat{\mathbf{c}}^{(l)}_{\text{L}^{(k)}} = (l_x, l_y, l_z)\rangle \in \text{SO}(3) \times \mathbb{R}^3\right\} \\ \mathcal{F}^P_{\text{L}^{(k)}} &:= \left\{\langle \hat{\mathbf{n}}^{(p)}_{\text{L}^{(k)}}, \hat{\mathbf{c}}^{(p)}_{\text{L}^{(k)}} = (p_x, p_y, p_z)\rangle \in \text{SO}(3) \times \mathbb{R}^3\right\}\end{aligned} \tag{20}$$

where, $\hat{\mathbf{n}}$ and $\hat{\mathbf{c}}$ denotes the estimated direction and center of the line and plane respectively. Finally, additional filtering is performed where only the closest line and planes between the set of different point clouds are kept for further refinement and verification. Two lines or planes are considered a close match if,

$$\begin{aligned}\alpha &= \|\cos^{-1}(\hat{\mathbf{n}}_{\text{B}} \cdot \hat{\mathbf{n}}_{\text{L}^{(k)}})\| && \leq \tau_\alpha \\ \delta &= \frac{|(\hat{\mathbf{n}}_{\text{B}} \times \hat{\mathbf{n}}_{\text{L}^{(k)}}) \cdot (\hat{\mathbf{c}}_{\text{B}} - \hat{\mathbf{c}}_{\text{L}^{(k)}})|}{\|(\hat{\mathbf{n}}_{\text{B}} \times \hat{\mathbf{n}}_{\text{L}^{(k)}})\|} && \leq \tau_\delta\end{aligned} \tag{21}$$

c) Extrinsic refinement: For refinement, we chose N set of corresponding line features from the point clouds for which $\tau^{(l)}_\alpha = 5°$ and $\tau^{(l)}_\delta = 0.5m$. By removing the k superscript for brevity, the optimal rotation can be computed by solving the following optimization,

$$\hat{\mathbf{R}}_{\texttt{Refined}} = \underset{\mathbf{R}_{\text{BL}}}{\arg\min} \sum_{i=1}^{N} \|\mathbf{R}_{\text{BL}}\hat{\mathbf{n}}_{\text{B}_i} - \hat{\mathbf{n}}_{\text{L}_i}\|^2_F \tag{22}$$

$$= \underset{\mathbf{R}_{\text{BL}}}{\arg\min} \sum_{i=1}^{N} 2\left[1 - \text{Tr}\left(\mathbf{R}_{\text{BL}}\hat{\mathbf{n}}_{\text{B}_i}\hat{\mathbf{n}}^T_{\text{L}_i}\right)\right]. \tag{23}$$

We formulate our problem as a Q-method and convert the cost function based on rotation matrices to quaternion form as discussed in our earlier work [23] as,

$$J(\mathbf{R}_{\text{BL}}) \overset{Eq.\ 23}{=} -\text{Tr}\left[\mathbf{R}_{\text{BL}} \textstyle\sum_{i=1}^{N}\left(\hat{\mathbf{n}}_{\text{B}_i}\hat{\mathbf{n}}^T_{\text{L}_i}\right)\right] = -\text{Tr}\left[\mathbf{R}_{\text{BL}}\mathbf{\Delta}_{3\times 3}\right]. \tag{24}$$

The attitude quaternion which minimizes the cost function is the unit eigenvector corresponding to the largest eigenvalue of $\mathbf{K}$ (eigenvalues of a symmetric matrix are real and hence can be sorted). Thus,

$$\hat{\mathbf{q}}(\mathbf{R}_{\text{BL}}) = \underset{\text{Eigenvalues}(\mathbf{K})}{\arg\max} \quad \text{Eigenvectors}(\mathbf{K}), \tag{25}$$

where, $\mathbf{K} = \begin{bmatrix} \mathbf{\Gamma} - \mu\mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^T & \mu \end{bmatrix}$, $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Delta}(2,3) - \mathbf{\Delta}(3,2) \\ \mathbf{\Delta}(3,1) - \mathbf{\Delta}(1,3) \\ \mathbf{\Delta}(1,2) - \mathbf{\Delta}(2,1) \end{bmatrix}$, $\mathbf{\Gamma} = \mathbf{\Delta}^T + \mathbf{\Delta}$ and $\mu = \text{Tr}(\mathbf{\Delta})$.

The optimal translation is obtained by matching the line centers as,

$$\hat{\mathbf{t}}_{\texttt{Refined}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\mathbf{c}}_{\text{L}} - \hat{\mathbf{c}}_{\text{B}}\right). \tag{26}$$

d) Extrinsics verification: After estimating the refined extrinsics we obtained the final extrinsic parameters as $\mathbf{T}_{\text{BL}^{(k)}} \simeq \hat{\mathbf{T}}_{\texttt{Refined}} \times \hat{\mathbf{T}}_{\texttt{GICP}} \times \hat{\mathbf{T}}_{\texttt{IMU}} \times \mathbf{T}_{\texttt{init}}$. At this stage, we applied the transformation to the point clouds to orient them to a common reference frame and extract plane features. If for the corresponding plane features, $\tau^{(p)}_\alpha = 1°$ and $\tau^{(p)}_\delta = 0.3m$, we accept the final extrinsic calibration, otherwise, we recompute the whole point-based alignment.

V. Experimental Results

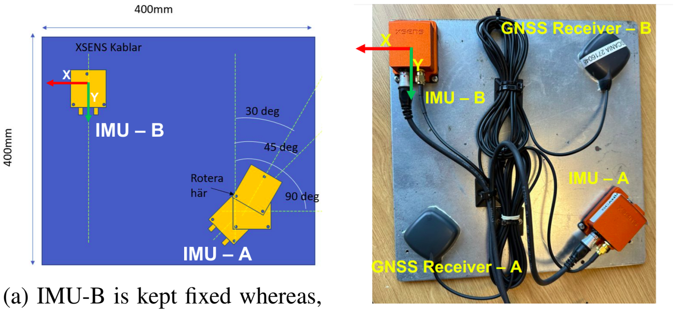
We carried out a series of experiments to test different aspects of algorithm — starting from a basic sensor rig and building up to a full demonstration on a test vehicle.

We first carried out MIMU calibration by mounting 2 IMUs on a custom made planar board. After this we collected sensor data from the test vehicle and computed the baseline extrinsics, where the (lidar) odometry was estimated using Fast-LIO2 [12] and matched using Kabsch alignment [10].

We then benchmark our results against CROON [13] where ground plane extraction and ICP (with normal) was performed between corresponding lidar frames and averaged our results across all the frames in the scene. Finally, we performed our calibration experiments with IMU-based initialization. The results are presented as the difference between ground truth (GT) and estimated extrinsics.

A. MIMU calibration on a test rig

We created the planar test rig seen in Fig. 4 to verify the MIMU calibration technique. The board contained 2 Xsens MTi-300 IMUs along with GNSS receivers to ensure time synchronization. $\texttt{IMU} - \texttt{B}$ was fixed to the board while we could adjust the pose of $\texttt{IMU} - \texttt{A}$ clockwise to three different orientations — $30°, 45°$ and $90°$. We collected IMU data at each orientation and computed the extrinsics by matching the IMU signals as discussed in IV-A4. We measured the GT translation components between the IMU centers using a measuring tape. The results are shown in Table I and an illustration of the signals before and after calibration of the $45°$ configuration are shown in Fig. 5 and Fig. 6 respectively. We see that after the calibration steps the IMU signals show good qualitative alignment when plotted in the same reference frame.



(a) IMU-B is kept fixed whereas, IMU-A is rotated clockwise at $30°, 45°$ and $90°$ configurations.

(b) $45°$ configuration example.

Fig. 4: Illustration of the test IMU board.

Performance comparison at different configurations: IMU test board dataset

Configuration	$\Delta x[m]$	$\Delta y[m]$	$\Delta z[m]$	$\Delta roll[°]$	$\Delta pitch[°]$	$\Delta yaw[°]$
30°	-0.0500	-0.0461	0.0015	-0.3268	-2.3129	-0.8808
45°	-0.0950	0.1018	0.0018	-0.4305	-2.2219	-1.2211
90°	-0.0003	0.0146	-0.0065	-0.3174	-2.3144	-0.0562

TABLE I

B. Lidar extrinsic calibration using IMU-based initialization on Scania datasets

After verifying the basic MIMU calibration approach on the test rig, we used the same method to pre-initialize our

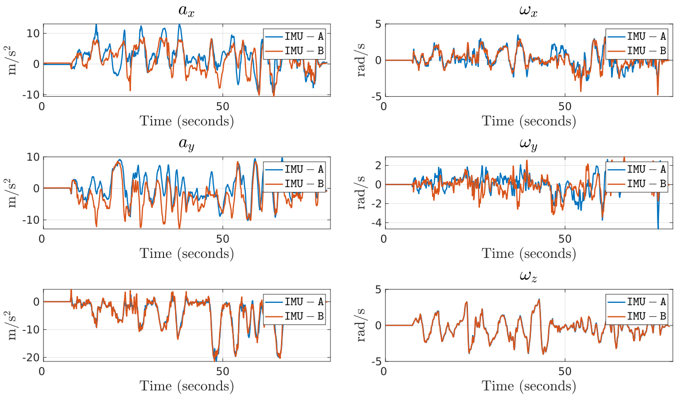


Fig. 5: Raw IMU signals from IMU – A and IMU – B at 45° configuration in individual sensor reference frame.

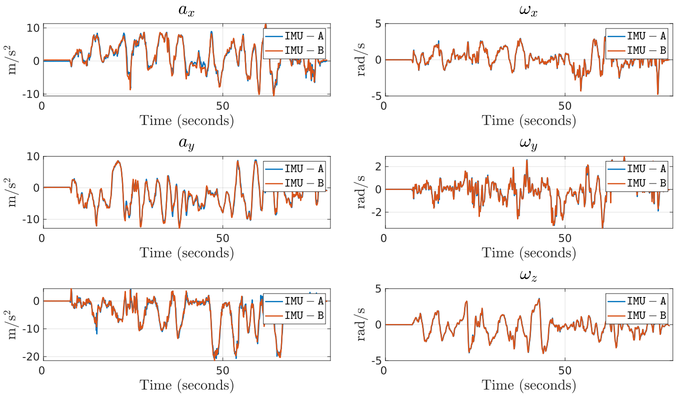


Fig. 6: Calibrated IMU signals from IMU – A and IMU – B at 45° configuration with bias compensation in IMU – B frame.

lidar extrinsic calibration algorithm. The GT for the extrinsic calibration parameters was pre-computed offline. We analyzed our lidar calibration results on 4 different test sequences which are described in Table II, covering different motion scenarios. We consider the feature richness level to be low if there are very few (≤ 3) geometric features detected per frame on average. In Seq-1, we performed aggressive maneuvers to excite all possible degrees of freedom in a parking lot. Seq-2 consists of very slow driving (≈10 km/h) in a feature rich environment.

Seq-3 and Seq-4 contain data from an urban driving scenario (≈40 km/h) in normal traffic conditions. We also computed the lidar odometry using FAST-LIO2 [12], for all the sequences and computed the RMSE of the absolute pose error (APE) between the lidars using GT extrinsics. This helped us to create the baseline for the extrinsics computation using Kabsch alignment. The high APE supports our claim that because the pose alignment technique has inherent drift and computationally heavy, it does not bring any additional value to the targetless extrinsics estimation process other than just a rough initialization.

Scania dataset collection details for the experiments

Data	Scenario	Length (Km)	Duration (secs)	Feature richness	APE
Seq-1	8-pattern in parking	0.325	72.5	Low	2.390
Seq-2	Slow turning	0.333	128.7	Medium	2.152
Seq-3	Urban driving	0.439	78.1	High	1.886
Seq-4	Urban driving	2.209	281.4	High	5.793

TABLE II

C. Bias estimation results

To improve our calibration performance, which relies on the signal-to-signal matching policy, we compensate for the biases in both the accelerometer and angular velocity signals. When a period of rest is identified, the accelerometer biases are recomputed and held constant for the following period. The angular velocity biases are updated based on the estimated orientation using the Madgwick filter that uses the bias-compensated accelerometer signals for gravity alignment. Whenever a period of rest is detected the covariances of the angular velocity biases converge.

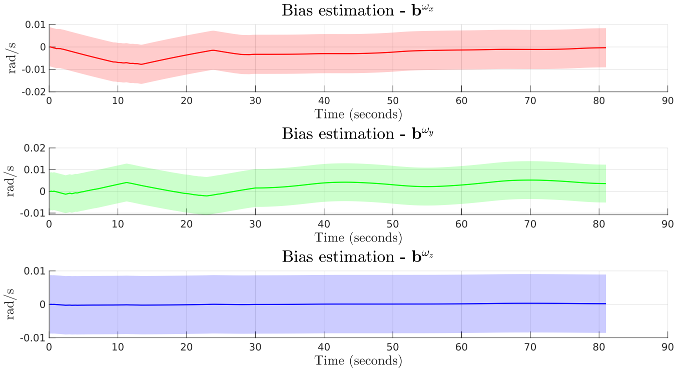


Fig. 7: Bias estimation for the $\mathtt{F_L}$ – Top angular velocity signals for Seq-1.

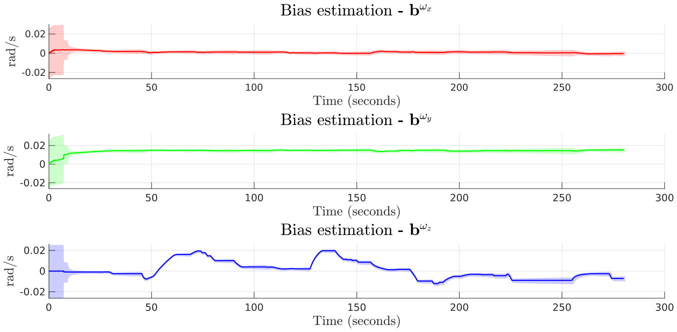


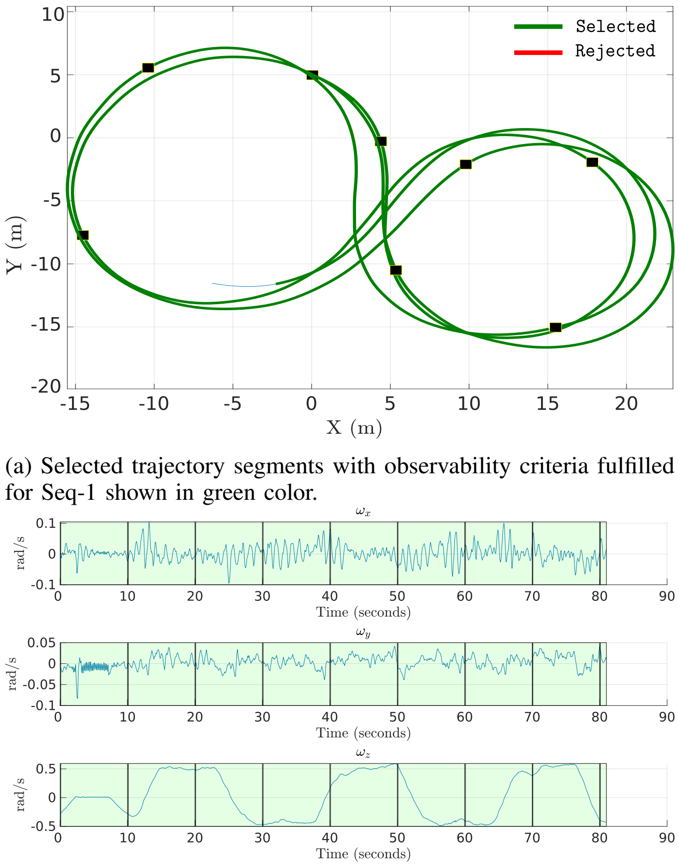
Fig. 8: Bias estimation for the $\mathtt{F_L}$ – Top angular velocity signals for Seq-4.

In Seq-1, we didn't have any periods of rest. Because of this, the angular velocity bias covariances did not converge as seen in Fig. 7. Whereas, in Fig. 8, we see that the angular velocity bias covariances converge after the detection of the rest period for a few seconds at the beginning of Seq-4.

D. Observability analysis

To study observability analysis, we extracted the information matrix by comparing the angular velocity signals between the IMU, $\mathtt{I}^{(k)}$, and the B frame by dividing the data into equal segments of 10 sec each. We analyze the SVD of the Fisher information matrix from Eq.18 and select the IMU data in the segment for calibration if the minimum singular value is greater than a threshold. For our experimental dataset, we set the threshold as 5^{-10}.

As seen in Fig. 9b, we selected all segments (highlighted in green) as there was enough excitation in the angular velocity

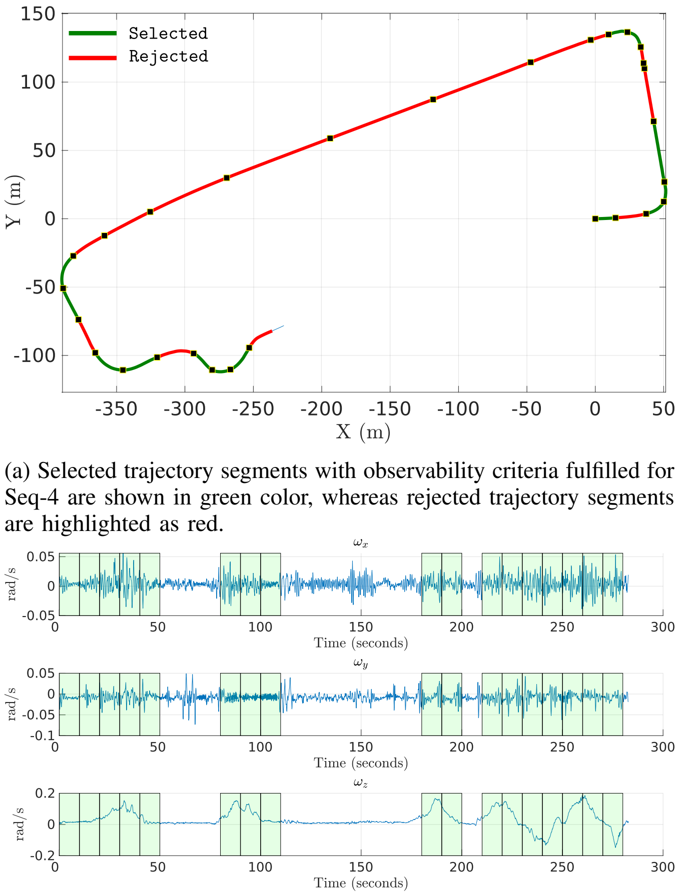


(a) Selected trajectory segments with observability criteria fulfilled for Seq-1 shown in green color.

(b) The angular velocity signals in different motion segments in a 10 sec window.

Fig. 9: Seq-1: The observability criteria selects all the motion primitives as there were enough excitation in the angular velocity signals in all the motion segments.

during all the time in Seq-1 as the vehicle was driving in a figure-of-8 pattern. The corresponding selected trajectory segments are shown in Fig. 9a. Similarly, for Seq-4, we can see that in Fig. 10a, relevant trajectories for calibration are selected only when there are strong turns maneuvers are observed in Fig. 10b as this is when there is enough excitation of the different degrees of freedom. This analysis helped us to understand that the best possible maneuvers for IMU-based initialization must contain aggressive turns like 8-patterns.



(a) Selected trajectory segments with observability criteria fulfilled for Seq-4 are shown in green color, whereas rejected trajectory segments are highlighted as red.

(b) We observe that in straight motion segments, there is not enough excitation in the angular velocity for Seq-4. Hence, the straight motion segments are not used from an observability perspective and are thus ignored for calibration.

Fig. 10: Seq-4: The observability criteria selects motion segments when there is enough excitation in the angular velocity signals, particularly during the turns.

Performance comparison of calibration routines: Scania dataset

Sequence	Method	$\Delta x[m]$	$\Delta y[m]$	$\Delta z[m]$	$\Delta roll[°]$	$\Delta pitch[°]$	$\Delta yaw[°]$
Seq-1	Kabsch	0.367	**0.099**	-0.309	-1.533	-2.647	-5.744
	CROON†	-0.844	-0.959	-0.340	-1.707	-2.783	1.918
	Ours*	**-0.164**	-0.184	**-0.153**	**-1.415**	**-2.759**	**-0.592**
Seq-2	Kabsch	0.734	-0.822	-0.342	**-1.654**	-2.860	-1.019
	CROON†	**-0.134**	-0.099	**-0.178**	-1.761	-2.722	**0.265**
	Ours*	0.187	**0.043**	-0.183	-2.105	**-2.562**	-1.889
Seq-3	Kabsch	0.675	-0.392	0.440	-2.474	-1.940	**0.083**
	CROON†	0.310	**-0.030**	0.081	-3.387	-1.762	-1.710
	Ours*	**0.261**	0.095	**-0.072**	**0.057**	**-1.561**	-0.159
Seq-4	Kabsch	1.260	-0.339	0.870	1.784	-4.947	0.304
	CROON†	0.494	0.403	**0.054**	-1.036	-1.447	-4.654
	Ours*	**0.118**	**-0.112**	-0.059	**0.438**	**-1.267**	**0.138**

* *Only translation initialization needed,* † *Both translation and rotation initialization needed.*

TABLE III

E. Online calibration experiments

GT calibration of the lidars is obtained by refining the vehicle CAD parameters. The refinement process analyzes known static feature positions around the vehicle in the world frame, $\texttt{W}$, and matches the corresponding detected features from the lidar perspective, using $\mathbf{T}_{\texttt{WB}}$ from GNSS. After that, we obtain the extrinsics of the lidars to the base frame, $\texttt{B}$ and compute the transformation matrix, $\mathbf{T}_{\texttt{GT}} = \mathbf{T}^{-1}_{\texttt{BL}(\texttt{F}_{\texttt{L}}-\texttt{Top})} \mathbf{T}_{\texttt{BL}(\texttt{F}_{\texttt{R}}-\texttt{Top})}$, between the 2 lidars. Ideally, GT calibration should be recomputed regularly. However, for our sequences, even though they were collected on different days we used the same GT parameters.

We compare the results after evaluating $\mathbf{T}_{\texttt{GT}} - \hat{\mathbf{T}}_{\texttt{Estimated}}$ for all the sequences using Kabsch alignment, CROON and our method. In the refinement step, we match a set of $N = 100$ lines before applying the optimization discussed in Sec. IV-B0c. Kabsch alignment requires lidar odometry and CROON requires ICP matching with normal estimation. Hence for both methods, the computation time is significantly higher. Also, our method does not need any rotation initialization and we can also verify the quality of the extrinsic parameters online which is not considered in these works. As seen in Table III, our method performs better in feature rich environments present in Seq-3

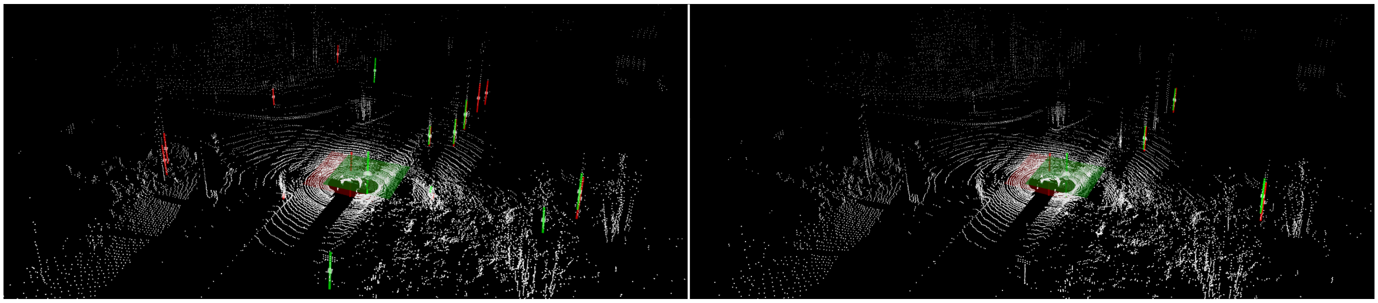


Fig. 11: Line and plane features: All (left) and closest (right) features after calibration refinement from Seq-4.

and Seq-4 and under aggressive maneuvers in Seq-1, where rich motion sequences are available. In Fig. 11 once can see the closest features from the two point clouds after refinement. The features extracted from $\mathtt{F_L-Top}$ and $\mathtt{F_R-Top}$ lidar are colored red and green respectively.

VI. Conclusion

In this work, we show that it is possible to initialize the extrinsic calibration of multiple lidars by matching the signals from IMUs co-located with the lidars. Unlike other methods which use odometry estimates for matching poses, this is a lightweight method that relies on raw signal matching. An observability-aware module informs us of the maneuvers needed to produce the excitations necessary for successful extrinsic calibration. Our method provides comparable extrinsics to the vehicle's CAD parameters when sufficient signal excitations are present in normal urban driving scenarios.

VII. Acknowledgements

This research has been jointly funded by the Swedish Foundation for Strategic Research (SSF) and Scania. The research is also affiliated with Wallenberg AI, Autonomous Systems and Software Program (WASP).

References

[1] J.-K. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," *IEEE Access*, vol. 8, pp. 134 101–134 110, 2020.

[2] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2018, pp. 5562–5569.

[3] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, "Pairwise lidar calibration using multi-type 3d geometric features in natural scene," in *IEEE Intl. Conference on Intelligent Robots and Systems*, 2013, pp. 1828–1835.

[4] D.-G. Choi, Y. Bok, J.-S. Kim, and I. S. Kweon, "Extrinsic calibration of 2-d lidars using two orthogonal planes," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 83–98, 2016.

[5] J. Brookshire and S. Teller, "Extrinsic calibration from per-sensor egomotion," *Robotics: Science and Systems VIII*, pp. 504–512, 2013.

[6] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.

[7] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[8] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1793–1800.

[9] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 351–371, 2021.

[10] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Crystallographica: Crystal Physics, Diffraction, Theoretical and General Crystallography*, pp. 827–828, 1978.

[11] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE Intl. Conference on Intelligent Robots and Systems*, 2018, pp. 4758–4765.

[12] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[13] P. Wei, G. Yan, Y. Li, K. Fang, X. Cai, J. Yang, and W. Liu, "Croon: Automatic multi-lidar calibration and refinement method in road scene," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 12 857–12 863.

[14] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small fov lidars and cameras using adaptive voxelization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[15] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 473–480.

[16] D. Tu, B. Wang, H. Cui, Y. Liu, and S. Shen, "Multi-camera-lidar auto-calibration by joint structure-from-motion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2242–2249.

[17] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 4304–4311.

[18] K. Eckenhoff, P. Geneva, and G. Huang, "Mimc-vins: A versatile and resilient multi-imu multi-camera visual-inertial navigation system," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1360–1380, 2021.

[19] M. E. Pittelkau, "Calibration and attitude determination with redundant inertial measurement units," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 743–752, 2005.

[20] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, 2017.

[21] T. Schneider, M. Li, C. Cadena, J. Nieto, and R. Siegwart, "Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation," *IEEE Sensors Journal*, vol. 19, pp. 3846–3860, 2019.

[22] J. Lv, X. Zuo, K. Hu, J. Xu, G. Huang, and Y. Liu, "Observability-aware intrinsic and extrinsic calibration of lidar-imu systems," *IEEE Transactions on Robotics*, 2022.

[23] S. Das, L. a. Klinteberg, M. Fallon, and S. Chatterjee, "Observability-aware online multi-lidar extrinsic calibration," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2860–2867, 2023.

[24] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[25] M. Pedley, "Tilt sensing using a three-axis accelerometer," *Freescale semiconductor application note*, vol. 1, pp. 2012–2013, 2013.

[26] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[27] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE Intl. Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[28] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[29] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 163–169, 2016.

[30] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *IEEE computer society conference on computer vision and pattern recognition*, vol. 1, 2005, pp. 220–226.