Observability-aware online multi-lidar extrinsic calibration

Sandipan Das[1,2], Ludvig af Klinteberg[2], Maurice Fallon[3], Saikat Chatterjee[1]

***Abstract*— Accurate and robust extrinsic calibration is necessary for deploying autonomous systems which need multiple sensors for perception. In this paper, we present a robust system for real-time extrinsic calibration of multiple lidars in vehicle base frame without the need for any fiducial markers or features. We base our approach on matching absolute GNSS and estimated lidar poses in real-time. Comparing rotation components allows us to improve the robustness of the solution than traditional least-square approach comparing translation components only. Additionally, instead of comparing all corresponding poses, we select poses comprising maximum mutual information based on our novel observability criteria. This allows us to identify a subset of the poses helpful for real-time calibration. We also provide stopping criteria for ensuring calibration completion. To validate our approach extensive tests were carried out on data collected using Scania test vehicles (7 sequences for a total of $\approx$ 6.5 Km). The results presented in this paper show that our approach is able to accurately determine the extrinsic calibration for various combinations of sensor setups.**

I. Introduction

Autonomous vehicles require multiple sensors to create 360° sensing coverage for navigating safely and redundancy. Thus it is important to determine the accurate mounting position of the sensors in the vehicle which is referred to as extrinsic calibration. It is not possible to fuse the sensing data in a common reference frame without correct extrinsic calibration parameters. In our work we focused on extrinsic calibration of multi-lidar systems.

Extrinsic calibration can be performed in 2 ways: Offline and Online. Offline procedure refers to matching features between the lidars by placing fiducial markers in their common field-of-view (FoV) of both the sensors [1], [2]. Online calibration is performed by matching the estimated states of the sensor in its corresponding sensor frame [3], [4], [5], [6], [7], [8]. Its desirable to perform online calibration since the sensors may not have common FoV and at the same time it reduces engineering effort compared to offline process.

Online calibration methods rely on motion of the mobile platform. At the same time it is well known [9], [10], [11], all motion segments do not provide enough information necessary for extrinsic calibration. Hence, identifying the degenerate motion segments help to discard poses which are not necessary for extrinsic calibration computation.

We provide an observability-aware online calibration framework which can be run in the vehicle for calibrating multiple lidars in real-time. We also provide an analytical study on the effect noisy state estimates in the online calibration process.

[1] KTH EECS, Sweden. {sandipan,sach}@kth.se
[2] Autonomous Transport Solutions Research, Scania CV AB, Sweden. {sandipan.das, ludvig.af.klinteberg}@scania.com
[3] Oxford Robotics Institute, UK. mfallon@robots.ox.ac.uk

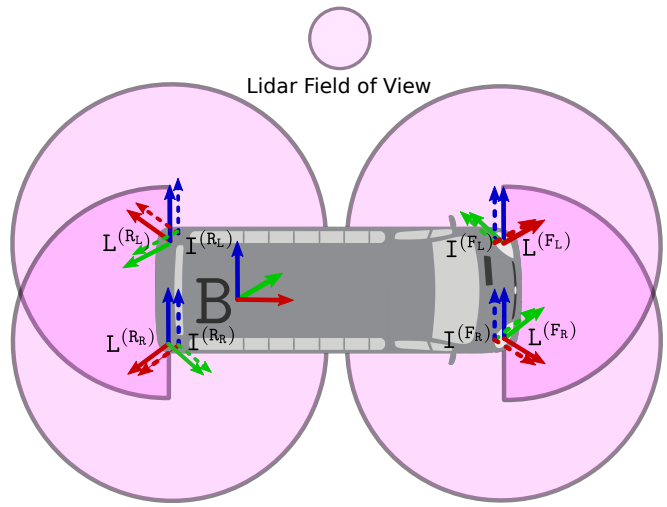

Fig. 1. Illustration of the four lidars with their embedded IMUs positioned around one data collection vehicle, when the sensors are calibrated. The vehicle base frame B, is located at the center of the rear axle. The sensor frames of the lidars are: $L^{(F_L)}$, $L^{(F_R)}$, $L^{(R_R)}$ and $L^{(F_L)}$, represented in solid lines; whereas, the sensor frames of the IMUs are: $I^{(F_L)}$, $I^{(F_R)}$, $I^{(R_R)}$ and $I^{(R_L)}$, represented in dashed lines. (F_L): front-left, (F_R): front-right, (R_R): rear-right and (R_L): rear-left.

A. Motivation

The mobile platforms used in our experiments have multitude of sensors as shown in Fig. 1. Moreover, before any autonomous run it is necessary to ensure that the sensors are properly calibrated. So we need to perform real-time calibration of all the sensors *wrt* to a common reference frame, usually located in center of the rear-axle. We also need to identify motion segments which provide enough excitation for different degrees of freedom required for calibration. Finally we need a mechanism to understand when the online calibration performance is satisfactory.

To achieve this, we calibrate all the lidars *wrt* a GNSS system which reports pose of the common reference frame in global frame. The GNSS system additionally reports the inertial measurements of the vehicle. All the lidar sensors used in our experiments also have an embedded IMU. Hence, for observability analysis we compare only the raw angular velocity signals between the GNSS system and the embedded IMU within the lidar(s) unlike comparing estimated states which inherently has process noise. Finally, we introduce a cost function to stop the calibration process when the results are satisfactory.

B. Contribution

Our work is motivated by the broad literature in motion based calibration. Our proposed contributions are:

- An extrinsic calibration algorithm to calibrate multiple lidars in real-time using GNSS reference poses with a calibration completion criteria.
- A novel approach for observability analysis based on comparing raw angular velocity signals.
- We also provide our analytical study on sensitivity of the estimated states used in our online calibration algorithm.
- Verification of our method based on data collected from Scania autonomous vehicles with the sensor setup shown in Fig. 2, with FoV schematics similar to Fig. 1.

II. Related Work

Online extrinsic calibration based on different strategies is an widely studied area. In our discussion we briefly review the relevant literature and motivate our choice of method adapted in our work.

A. Motion-based methods for online extrinsic calibration

Online extrinsic calibration based on the Hand-Eye method [12], [13] is an extensively studied topic. Initial research on motion-based calibration aligned a gripper (hand) and a camera (eye) by estimating the motion of the gripper along with the camera and constraining their poses with a fixed rigid body transformation, this approach is called Hand-Eye calibration and has been reviewed by Shah et al. [14]. Multi-camera extrinsic calibration with the Hand-Eye method have been explored in works like Camodocal [4] and recently extended to multi-lidar extrinsic calibration [15], [16].

Instead of performing Hand-Eye calibration, in Kalibr [9] 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 unobservant directions and degenerate scenarios. Furgale et al. [5] further generalized Kalibr for spatio-temporal calibration. We build upon the philosophy of Kalibr and minimize the error between the poses estimated by the lidar(s) and the GNSS in a batch optimization process to recover the extrinsic calibration parameters.

1) Types of motion models: Huang et al. [6] proposed a framework to evaluate different motion based strategies with their work on noise sensitivity analysis. They categorized the motion-based methods as:

- Model A: Comparing absolute poses in a common frame.
- Model B: Comparing absolute poses in separate frames.
- Model C: Comparing relative poses – Hand-Eye method.

They show that Model A performs well for all the relative angles between the pose pairs and has minimum number of degeneration zones. Hence, we base our cost function to recover transformation based on Model A.

Fig. 2. Reference frames conventions for our bus and truck platforms. In both cases, 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 lidar frame $\mathtt{L}$.

2) Types of cost functions: We can formulate the cost function in 2 different approaches:

- Minimize the absolute translation in a least-square approach [17], [18].
- Assume additive Gaussian noise in the estimated poses and minimize the weighted noise [19], [7] component.

In our work we formulate our cost function as a least-square problem. Unlike comparing only the translation component for comparing the poses we also compare the rotation components using the Q-method by Davenport [20]. Additionally, we analytically show that optimization problem remains invariant under certain noisy conditions. We also base our novel observability criteria based on comparing the raw angular velocity signals and stop the calibration process when satisfactory performance is reached.

B. Lidar odometry

There have been vast array of works on lidar odometry after the LOAM paper by Zhang et al. [21]. Often geometric features are extracted from the point clouds and tracked between frames for computational efficiency of various state estimation algorithms [22], [23], [24]. However, this principle doesn't work for featureless environments. Hence, instead of feature extraction, the whole point cloud is often processed, which is known as direct estimation, shown in works like Fast-LIO2 [25] and Faster-LIO [26]. We adapted Fast-LIO2 for estimating lidar odometry in our online calibration algorithm.

III. Problem Statement

A. Sensor platform and reference frames

The sensor configuration used is as shown in Fig. 2 for a bus and truck. We also show illustrations of the fields-of-view of the sensors for the bus in Fig. 1. Each of the sensor housings contains a lidar with an integrated IMU and two cameras. Please note the cameras are not used for this work.

First we will describe the necessary notation and the reference frames used in our system. The vehicle base frame, $\mathtt{B}$ is located on the center of the rear-axle of the vehicle. Sensor readings from GNSS, lidars and IMUs are represented in their respective sensor frames as $\mathtt{G}$, $\mathtt{L}^{(k)}$ and $\mathtt{I}^{(k)}$ respectively. Here, $k \in [\mathtt{F_L}, \mathtt{F_R}, \mathtt{R_L}, \mathtt{R_R}]$ denotes the location of the sensor in the vehicle corresponding to front-left, front-right, rear-left and rear-right respectively. The GNSS measurements are reported in world fixed frame, $\mathtt{W}$ and transformed to $\mathtt{B}$ frame by performing a calibration routine outside the scope of this work. 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 $\mathbf{R}\mathbf{R}^T = \mathbf{I}_{3\times3}$, since the rotation matrix is orthogonal.

B. Problem formulation

Our primary goal is to estimate the extrinsic calibration of multiple lidar sensors *wrt* the base frame and to recover $\mathbf{T}_{\mathtt{BL}^{(k)}}$ in real-time without the need for any fiducial markers. Since, our method is based on a motion-based approach, we first estimate the pose of the lidar sensor, $\mathbf{T}_{\mathtt{L}_i^{(k)}}$, at time t_i in its corresponding sensor frame, $\mathtt{L}^{(k)}$, prior to the online calibration. We also study the effect of process noise in our algorithm and identify the states which provide maximum excitation for different degrees of freedom, based on which online extrinsic calibration is performed.

IV. Methodology

A. Initialization

All the GNSS poses in base frame, $\mathtt{B}$ are normalized *wrt* to the initial pose, ${}_{\mathtt{W}}\mathbf{T}_{\mathtt{WG}}$. The GNSS unit also provides IMU measurements in the $\mathtt{G}$ frame using its internally embedded IMU. These measurement are then transformed to base frame, $\mathtt{B}$. For gravity alignment we use the equations 25 and 26 from [27] to estimate roll and pitch after collecting IMU data when the vehicle is static for few seconds according to the GNSS estimate.

Note that the noise processes and starting bias estimates for the embedded IMU sensor within the lidars and the GNSS unit were characterized in advance by estimating the Allan Variance [28] parameters using logs collected while the vehicle was stationary.

B. Lidar-IMU calibration

The integrated IMU within the lidar is utilized to rectify motion-induced lidar distortion as well as to pre-compute the initial guess for scan matching within the lidar odometry. Hence, it is important to calibrate the lidar and its corresponding integrated IMU. If the sensor supplier provided extrinsics are not available, the calibration can be formulated as a Hand-Eye problem to estimate the relative IMU and lidar poses separately. We integrate the IMU measurements between estimated lidar poses using IMU preintegration [29]. The state propagation equations are based on the IMU data $[\boldsymbol{\omega}$: angular velocity, $\mathbf{f}$: linear acceleration$]$ between two timestamps $[t_i, t_j]$ for all the Δt (inverse of sensor frequency) intervals in the $\mathrm{SE}(3)$ manifold is derived in [29] as:

$$\begin{aligned} \Delta\mathtt{R}_{ij} &= \prod_{k=i}^{j-1} \mathrm{Exp}\left(\boldsymbol{\omega}_{\mathtt{I}_k}\Delta t\right), \\ \Delta\mathbf{v}_{ij} &= \sum_{k=i}^{j-1} \Delta\mathrm{R}_{ik}\mathbf{f}_{\mathtt{I}_k}\Delta t, \\ \Delta\mathbf{p}_{ij} &= \sum_{k=i}^{j-1} \left[\Delta\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\mathrm{R}_{ik}\mathbf{f}_{\mathtt{I}_k}\Delta t^2\right], \end{aligned} \tag{1}$$

where, $\Delta\mathtt{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{p}_{ij}$ represent the relative rotation, velocity and position respectively. Based on the computed lidar II-B and IMU relative poses (Eq. 1) we formulate a Hand-Eye problem as,

$$\mathbf{T}_{\mathtt{L}_i\mathtt{L}_j}\mathbf{T}_{\mathtt{LI}} = \mathbf{T}_{\mathtt{LI}}\mathbf{T}_{\mathtt{I}_i\mathtt{I}_j}. \tag{2}$$

Since, we use the relative transforms from the IMU propagation, $\mathbf{R}^\star_{\mathtt{LI}}$ is determined by equating the rotation parts in Eq. 2 in quaternion form, similar to [8] as,

$$\begin{aligned} &\mathbf{q}_{\mathtt{L}_i\mathtt{L}_j} \otimes \mathbf{q}_{\mathtt{LI}} = \mathbf{q}_{\mathtt{LI}} \otimes \mathbf{q}_{\mathtt{I}_i\mathtt{I}_j} \\ \implies &\underbrace{\left[\mathcal{L}(\mathbf{q}_{\mathtt{L}_i\mathtt{L}_j}) - \mathcal{R}(\mathbf{q}_{\mathtt{I}_i\mathtt{I}_j})\right]}_{\mathbf{A}_k}\mathbf{q}_{\mathtt{LI}} = 0, \end{aligned} \tag{3}$$

where, the matrices, $\mathcal{L}(\mathbf{q})$ and $\mathcal{R}(\mathbf{q})$ are defined as,

$$\begin{aligned} \mathcal{L}(\mathbf{q}) &= \begin{bmatrix} q_s\mathbf{I}_3 - [\mathbf{q}_v]_\times & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_s \end{bmatrix} \\ \mathcal{R}(\mathbf{q}) &= \begin{bmatrix} q_s\mathbf{I}_3 + [\mathbf{q}_v]_\times & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_s \end{bmatrix}, \end{aligned}$$

and, $\mathbf{q}$ is a quaternion represented as,

$$\mathbf{q} = \underbrace{q_s}_{\text{Scalar component}} + \underbrace{q_x\hat{i} + q_y\hat{j} + q_z\hat{k}}_{\text{Vector component}} = [q_s, \mathbf{q}_v]. \tag{4}$$

For a set of relative poses we can form a system of linear equations of the form, $\mathbf{A}\mathbf{x} = \mathbf{0}$, where $\mathbf{A} = [\mathbf{A}_0^T\ \mathbf{A}_1^T\ \dots\ \mathbf{A}_n^T]^T$. $\mathbf{q}^\star_{\mathtt{LI}}$ lies in the nullspace of $\mathbf{A}$, which corresponds to the right unit singular vector with the smallest singular value of $\mathbf{A}$.

After finding $\mathbf{q}^\star_{\mathtt{LI}}$, for the translation component we get,

$$\underbrace{\left(\mathbf{R}_{\mathtt{L}_i\mathtt{L}_j} - \mathbf{I}\right)}_{\mathbf{A}_k}\mathbf{t}_{\mathtt{LI}} \overset{Eq.\ 2}{=} \underbrace{\left(\mathbf{R}^\star_{\mathtt{LI}}\mathbf{t}_{\mathtt{I}_i\mathtt{I}_j} - \mathbf{t}_{\mathtt{L}_i\mathtt{L}_j}\right)}_{\mathbf{B}_k}, \tag{5}$$

which is a system of linear equations of the form, $\mathbf{A}\mathbf{x} = \mathbf{B}$, where, $\mathbf{A} = [\mathbf{A}_0^T\ \mathbf{A}_1^T\ \dots\ \mathbf{A}_n^T]^T$, $\mathbf{B}^T = [\mathbf{B}_0^T\ \mathbf{B}_1^T\ \dots\ \mathbf{B}_n^T]^T$ and $\mathbf{x} = \mathbf{t}_{\mathtt{LI}}$. This is an over-determined system which can be solved using a least-square approach as,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{B}\|^2_{\boldsymbol{\Sigma}}, \tag{6}$$

where, $\boldsymbol{\Sigma}$ denotes the covariance of the residual. For these problems, the general strategy [30] is to iteratively approximate the original problem 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 $F(\mathbf{x})$. Thus, $\hat{\mathbf{x}}$ is updated in the current iteration as, $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} \boxplus \boldsymbol{\Delta}\mathbf{x}$, where $\boxplus$ is an additional operator on the manifold and the problem then becomes,

$$\boldsymbol{\Delta}\mathbf{x}^\star = \arg\min_{\boldsymbol{\Delta}\mathbf{x}} \frac{1}{2}\|(\mathbf{A}\hat{\mathbf{x}} - \mathbf{B}) + \mathbf{A}^T\boldsymbol{\Delta}\mathbf{x}\|^2_{\boldsymbol{\Sigma}}. \tag{7}$$

The optimal solution is given by,

$$\underbrace{\left(\mathbf{J}^\top\boldsymbol{\Sigma}^{-1}\mathbf{J}\right)}_{\text{Fisher information matrix}}\boldsymbol{\Delta}\mathbf{x} = -\mathbf{J}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{B}). \tag{8}$$

In practice, since we can take a reliable initial estimate for the translation component from the vehicle's CAD model, we search for $\mathbf{x}^\star$ only in a local neighborhood within reasonable

bounds. Thus, we solve a bounded variable least squares problem as,

$$\mathbf{x}^\star = \underset{\mathbf{L}\leq\mathbf{x}\leq\mathbf{U}}{\arg\min} \|\mathbf{A}\mathbf{x}-\mathbf{B}\|^2_{\mathbf{\Sigma}}, \tag{9}$$

where, $\mathbf{L}$ and $\mathbf{U}$ are the lower and upper bounds of $\mathbf{x}$ respectively. Thus the solution space in Eq. 7 is modified with an additional constraint as,

$$\begin{aligned}\mathbf{\Delta x}^\star &= \underset{\mathbf{\Delta x}}{\arg\min} \frac{1}{2}\|(\mathbf{A}\hat{\mathbf{x}}-\mathbf{B})+\mathbf{A}^T\mathbf{\Delta x}\|^2_{\mathbf{\Sigma}},\\ &s.t.\ \mathbf{L}\leq\hat{\mathbf{x}}+\mathbf{\Delta x}\leq\mathbf{U}.\end{aligned} \tag{10}$$

C. Lidar calibration in base frame

Let, $\mathbf{T}_{\mathtt{B}_i}$ be the pose of the base frame obtained using the GNSS measurements at timestamp t_i. And, $\mathbf{T}_{\mathtt{L}_i^{(k)}}$ be the pose of the kth lidar in its corresponding sensor frame obtained from a state of the art state estimator [25] at timestamp t_i. Then our problem is to estimate the optimal transformation matrix between 2 pair of poses accumulated over N timestamps:

$$\begin{aligned}\mathbf{T}^\star_{\mathtt{BL}^{(k)}} &= \begin{bmatrix}\mathbf{R}^*_{3\times3} & \mathbf{t}^*_{3\times1}\\ \mathbf{0}^\top & 1\end{bmatrix}\\ &= \underset{\mathbf{T}_{\mathtt{BL}^{(k)}}}{\arg\min}\sum_{i=1}^{N}\|\mathbf{T}_{\mathtt{BL}^{(k)}}\mathbf{T}_{\mathtt{B}_i}-\mathbf{T}_{\mathtt{L}_i^{(k)}}\|^2_F \qquad (11)\\ &= \underset{\mathbf{R}_{\mathtt{BL}^{(k)}},\mathbf{t}_{\mathtt{BL}^{(k)}}}{\arg\min}\sum_{i=1}^{N}\left\|\begin{array}{c|c}\mathbf{R}_{\mathtt{BL}^{(k)}}\mathbf{R}_{\mathtt{B}_i}-\mathbf{R}_{\mathtt{L}_i} & \mathbf{R}_{\mathtt{BL}^{(k)}}\mathbf{t}_{\mathtt{B}_i}+\mathbf{t}_{\mathtt{BL}^{(k)}}-\mathbf{t}_{\mathtt{L}_i}\\ \hline \mathbf{0}^\top & \mathbf{0}^\top\end{array}\right\|^2_F. \qquad (12)\end{aligned}$$

1) Initial calibration estimation: The initial extrinsics can be recovered by aligning the initial set of translation components, $\mathbf{t}_{\mathtt{B}_i}$ and $\mathbf{t}_{\mathtt{L}_i}$, $\forall i = 1(1)N$, using a 3D-point cloud alignment method proposed by Kabsch et al. [17] or Horn et al. [18], by breaking the problem into rotation and translation parts. Note that we omit the kth superscript for brevity in the rest of our discussion.

2) Rotation estimation: For Frobenius norm,

$$\|\mathbf{A}+\mathbf{B}\|^2_{\mathrm{F}} \overset{def}{=} \|\mathbf{A}\|^2_{\mathrm{F}}+\|\mathbf{B}\|^2_{\mathrm{F}}+2\langle\mathbf{A},\mathbf{B}\rangle_{\mathrm{F}} \tag{13}$$

$$\text{where, } \langle\mathbf{A},\mathbf{B}\rangle_{\mathbf{F}} \overset{def}{=} \sum_{i,j}\overline{A_{ij}}B_{ij} = \mathrm{Tr}\left(\mathbf{A}^T\mathbf{B}\right). \tag{14}$$

After recovering the initial parameters, the optimal rotation optimization can be re-written as,

$$\begin{aligned}\mathbf{R}^\star_{\mathtt{BL}} &= \underset{\mathbf{R}_{\mathtt{BL}}}{\arg\min}\sum_{i=1}^{N}\|\mathbf{R}_{\mathtt{BL}}\mathbf{R}_{\mathtt{B}_i}-\mathbf{R}_{\mathtt{L}_i}\|^2_F \qquad (15)\\ &= \underset{\mathbf{R}_{\mathtt{BL}}}{\arg\min}\textstyle\sum_{i=1}^{N}\left[\|\mathbf{R}^T_{\mathtt{B}_i}\mathbf{R}_{\mathtt{B}_i}\|+\|\mathbf{R}^T_{\mathtt{L}_i}\mathbf{R}_{\mathtt{L}_i}\|-2\,\mathrm{Tr}\left(\mathbf{R}_{\mathtt{BL}}\mathbf{R}_{\mathtt{B}_i}\mathbf{R}^T_{\mathtt{L}_i}\right)\right]\\ &= \underset{\mathbf{R}_{\mathtt{BL}}}{\arg\min}\sum_{i=1}^{N}2\left[1-\mathrm{Tr}\left(\mathbf{R}_{\mathtt{BL}}\mathbf{R}_{\mathtt{B}_i}\mathbf{R}^T_{\mathtt{L}_i}\right)\right], \qquad (16)\end{aligned}$$

where, the rotation components are in $\mathrm{SE}(3)$, $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. Because the constant terms do not affect the argmin operator, we can discard those and introduce our proposed cost function as,

$$J(\mathbf{R}_{\mathtt{BL}}) = -\sum_{i=1}^{N}\mathrm{Tr}\left(\mathbf{R}_{\mathtt{BL}}\mathbf{R}_{\mathtt{B}_i}\mathbf{R}^T_{\mathtt{L}_i}\right). \tag{17}$$

3) Q-method formulation: We formulate our problem as Q-method [20] and follow their ideas to convert the cost function based on rotation matrices to quaternion form as follows,

$$J(\mathbf{R}_{\mathtt{BL}}) \overset{Eq.\ 17}{=} -\mathrm{Tr}\left[\mathbf{R}_{\mathtt{BL}}\textstyle\sum_{i=1}^{N}\left(\mathbf{R}_{\mathtt{B}_i}\mathbf{R}^T_{\mathtt{L}_i}\right)\right] = -\mathrm{Tr}\left[\mathbf{R}_{\mathtt{BL}}\mathbf{\Delta}_{3\times3}\right], \tag{18}$$

where, $\mathbf{\Delta}_{3\times3} = \sum_{i=1}^{N}\left(\mathbf{R}_{\mathtt{B}_i}\mathbf{R}^T_{\mathtt{L}_i}\right)$. The rotation matrix can be expressed as a quaternion as,

$$\mathbf{R}_{\mathtt{BL}} = \left(q_s^2-\mathbf{q}_v^T\mathbf{q}_v\right)\mathbf{I}+2\mathbf{q}_v\mathbf{q}_v^T-2q_s\left[\mathbf{q}_v\right]_\times, \tag{19}$$

where, $[.]_\times$ is the skew-symmetric matrix. So, if we rewrite the cost function of Eq. 18 in quaternion space we get,

$$\begin{aligned}J(\mathbf{q}) &\overset{Eq.\ 19}{=} -\mathrm{Tr}\left[\left((q_s^2-\mathbf{q}_v^T\mathbf{q}_v)\mathbf{I}+2\mathbf{q}_v\mathbf{q}_v^T-2q_s\left[\mathbf{q}_v\right]_\times\right)\mathbf{\Delta}\right]\\ &= -\left(q_s^2-\mathbf{q}_v^T\mathbf{q}_v\right)\mathrm{Tr}\left(\mathbf{\Delta}\right)-\mathbf{q}_v^T\left(\mathbf{\Delta}^T+\mathbf{\Delta}\right)\mathbf{q}_v+2q_s\,\mathrm{Tr}\left[(\mathbf{q}_v)_\times\mathbf{\Delta}\right].\end{aligned} \tag{20}$$

Expanding the last term in the Eq. 20 yields,

$$\begin{aligned}&\mathrm{Tr}\left[(\mathbf{q}_v)_\times\mathbf{\Delta}\right]\\ &= q_x(\mathbf{\Delta}(3,2)-\mathbf{\Delta}(2,3))+q_y(\mathbf{\Delta}(1,3)-\mathbf{\Delta}(3,1))+q_z(\mathbf{\Delta}(2,1)-\mathbf{\Delta}(1,2))\\ &= \mathbf{\Lambda}\mathbf{q}_v,\end{aligned} \tag{21}$$

where, $\mathbf{\Lambda} = \begin{bmatrix}\mathbf{\Delta}(2,3)-\mathbf{\Delta}(3,2)\\ \mathbf{\Delta}(3,1)-\mathbf{\Delta}(1,3)\\ \mathbf{\Delta}(1,2)-\mathbf{\Delta}(2,1)\end{bmatrix}$. Hence, in quaternion form the cost function can be re-written as,

$$J(\mathbf{q}) = -\left(q_s^2-\mathbf{q}_v^T\mathbf{q}_v\right)\mathrm{Tr}\left(\mathbf{\Delta}\right)-\mathbf{q}_v^T\left(\mathbf{\Delta}^T+\mathbf{\Delta}\right)\mathbf{q}_v+2q_s\mathbf{\Lambda}\mathbf{q}_v \tag{22}$$

Defining, $\mathbf{\Gamma} = \mathbf{\Delta}^T+\mathbf{\Delta}$ and $\mu = \mathrm{Tr}(\mathbf{\Delta})$ we get,

$$\begin{aligned}J(\mathbf{q}) &\overset{Eq.\ 22}{=} -\left(q_s^2-\mathbf{q}_v^T\mathbf{q}_v\right)\mathrm{Tr}\left(\mathbf{\Delta}\right)-\mathbf{q}_v^T\left(\mathbf{\Delta}^T+\mathbf{\Delta}\right)\mathbf{q}_v+2q_s\mathbf{\Lambda}\mathbf{q}_v\\ &= -\left(\mathbf{q}_v^T(\mathbf{\Gamma}-\mu\mathbf{I})\mathbf{q}_v+q_s\mathbf{\Lambda}\mathbf{q}_v+q_s\mathbf{\Lambda}^T+q_s^2\mu\right)\\ &= -\begin{bmatrix}\mathbf{q}_v^T & q_s\end{bmatrix}\begin{bmatrix}\mathbf{\Gamma}-\mu\mathbf{I} & \mathbf{\Lambda}\\ \mathbf{\Lambda}^T & \mu\end{bmatrix}\begin{bmatrix}\mathbf{q}_v\\ q_s\end{bmatrix}\\ &= -\mathbf{q}^T\mathbf{K}\mathbf{q},\end{aligned} \tag{23}$$

where, $\mathbf{q}$ is the attitude quaternion (vector first), and $\mathbf{K}$ is the Davenport matrix denoted as $\mathbf{K} = \begin{bmatrix}\mathbf{\Gamma}-\mu\mathbf{I} & \mathbf{\Lambda}\\ \mathbf{\Lambda}^T & \mu\end{bmatrix}$. Since, $\mathbf{\Gamma}$ is symmetric, the Davenport matrix is also symmetric. From spectral theorem [31], we know that for any symmetric matrix, the eigenvalues are real and the associated eigenvectors form an orthonormal basis. We use this property to ensure a deterministic solution in the next steps of the process.

4) Q-method optimization with Lagrange multiplier: We can now perform a constrained optimization by considering the orthogonal constraints of the rotation, $\mathbf{q}\mathbf{q}^T = \mathbf{I}$, which can be introduced using a Lagrange multiplier, λ. The optimization function for Eq. 23 can then be re-written as,

$$\min_{\mathbf{q},\lambda} J(\mathbf{q},\lambda) = -\mathbf{q}^T\mathbf{K}\mathbf{q} + \lambda\left(\mathbf{q}^T\mathbf{q} - 1\right) \tag{24}$$

$$\text{For minima, } \frac{\partial J}{\partial \mathbf{q}} = 0 \implies \mathbf{K}\mathbf{q} = \lambda\mathbf{q}. \tag{25}$$

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).

$$\mathbf{q}^\star = \max_{\text{Eigenvalues}(\mathbf{K})} \text{Eigenvectors}(\mathbf{K}) \tag{26}$$

$$\implies \min_{\mathbf{q},\lambda} J(\mathbf{q},\lambda) \overset{Eq.\ 24}{=} -\lambda\mathbf{I} + \lambda(\mathbf{I} - 1) = -\lambda. \tag{27}$$

We only update the estimated extrinsic rotation parameters if it improves the rotational cost function, as described in IV-C.7.

5) Translation estimation: Note that, because the motion of the lidar is estimated in the sensor frame, it is not possible to solve for the translation component without the relative transformations and the rigid body constraint provided by Hand-Eye method [12]. From Fig. 3 we can write:

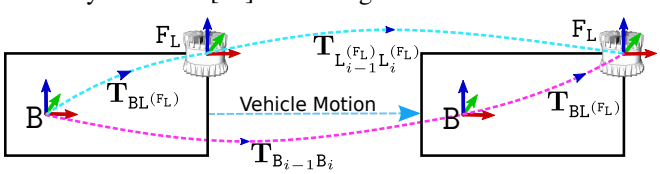

Fig. 3. Illustration of Hand-eye calibration between $\mathtt{B}$ and $\mathtt{L}^{(\mathtt{F_L})}$ frame.

$$\mathbf{T}_{\mathtt{B}_{i-1}\mathtt{B}_i}\mathbf{T}_{\mathtt{BL}} = \mathbf{T}_{\mathtt{BL}}\mathbf{T}_{\mathtt{L}_{i-1}\mathtt{L}_i}, \tag{28}$$

which can be further decomposed into the translation part with the pre-computed optimal rotation in previous step as,

$$\underbrace{\left(\mathbf{R}_{\mathtt{B}_{i-1}\mathtt{B}_i} - \mathbf{I}\right)}_{\mathbf{A}_i}\mathbf{t}_{\mathtt{BL}} \overset{Eq.\ 26}{=} \underbrace{\left(\mathbf{R}^\star_{\mathtt{BL}}\mathbf{t}_{\mathtt{L}_{i-1}\mathtt{L}_i} - \mathbf{t}_{\mathtt{B}_{i-1}\mathtt{B}_i}\right)}_{\mathbf{B}_i}, \tag{29}$$

where, $\forall i = 1(1)N$, which can be computed using similar principle of solving Eq. 9 with known prior bounds.

6) Observability analysis: Not all motions of the test vehicle sufficiently excite the degrees of freedom of the calibration technique. To ensure that online calibration is reliable it is necessary to identify the dataset segments which is suitable and ignore the others.

The extrinsics of $\mathtt{I}^{(k)}$ *wrt* $\mathtt{B}$ frame is strongly correlated with the extrinsics of $\mathtt{L}^{(k)}$ *wrt* $\mathtt{B}$ frame. Moreover, the angular velocity at all the points in a rigid body are identical and can be correlated to the motion estimate of the vehicle. Thus, we formulate a cost function for maximizing the likelihood of matching the angular velocity between $\mathtt{I}^{(k)}$ and $\mathtt{B}$ frame as,

$$\mathbf{R}^\star_{\mathtt{BI}} = \arg\min_{\mathbf{R}_{\mathtt{BI}}} \sum_{i=1}^{N} \|\mathbf{R}_{\mathtt{BI}}\boldsymbol{\omega}_{\mathtt{B}_i} - \boldsymbol{\omega}_{\mathtt{I}_i}\|^2_{\Sigma_i}, \quad \text{s.t. } \mathbf{R}_{\mathtt{BI}}\mathbf{R}_{\mathtt{BI}}^T = \mathbf{I}_3, \tag{30}$$

Algorithm 1: Online lidar extrinsic calibration

Input: Lidar point cloud in $\mathtt{L}$ frame, GNSS reference poses in $\mathtt{B}$ frame = $\mathbf{T}_{\mathtt{B}_i}$, Angular velocity signals of IMUs embedded in lidar and GNSS = $\{\boldsymbol{\omega}_{\mathtt{I}_i}, \boldsymbol{\omega}_{\mathtt{B}_i}\}$, Batch size = N, Pose error threshold = β, Information threshold = ϵ.

Output: $\mathbf{T}^\star_{\mathtt{BL}}$.

if *not initialized* **then**
  if $\mathbf{T}_{\mathtt{LI}}$ *is unknown* **then**
    // Estimate $\mathbf{T}_{\mathtt{L}_1\mathtt{L}_N}$ with lidar odometry and $\mathbf{T}_{\mathtt{I}_1\mathtt{I}_N}$ using Eq. 1 for Hand-Eye
    Estimate $\mathbf{T}_{\mathtt{LI}}$ using Eq. 3 and Eq. 5.
  Estimate $\mathbf{T}_{\mathtt{L}_i}$ using lidar odometry.
  // Since, initial $\mathbf{t}_{\mathtt{BL}}$ can be measured, only initial $\mathbf{R}_{\mathtt{BL}}$ computation is necessary
  Compute initial $\mathbf{R}_{\mathtt{BL}}$ and $\mathbf{t}_{\mathtt{BL}}$ by aligning the set of $\mathbf{t}_{\mathtt{L}_i}$ *wrt* $\mathbf{t}_{\mathtt{B}_i}$, $\forall i = 1(1)N$ by Umeyama [32] or Horn [18] alignment.
Let, $\mathtt{Error}(\mathbf{T}_{\mathtt{BL}}) \overset{Eq.\ 33}{=} \frac{1}{N}\sqrt{\sum_{i=1}^{N}\xi_{\mathbf{T}_{\mathtt{BL}}}(\mathbf{T}_{\mathtt{B}_i}, \mathbf{T}_{\mathtt{L}_i})}$
and, $\mathtt{R}_{\mathtt{Error}}(\mathbf{R}_{\mathtt{BL}}) \overset{Eq.\ 33}{=} \frac{1}{N}\sqrt{\sum_{i=1}^{N}\xi_{\mathbf{R}_{\mathtt{BL}}}(\mathbf{R}_{\mathtt{B}_i}, \mathbf{R}_{\mathtt{L}_i})}$.
Let, $\mathbf{T}_{\mathtt{B}} = \{\}$ and $\mathbf{T}_{\mathtt{L}} = \{\}$, be the set of all GNSS and lidar poses which satisfy observability criteria.
while $\mathtt{Error}(\mathbf{T}_{\mathtt{BL}}) \geq \beta$ **do**
  Estimate $\mathbf{T}_{\mathtt{L}_i}$ using lidar odometry.
  Find closest $\mathbf{T}_{\mathtt{B}_i}$ *wrt* $\mathbf{T}_{\mathtt{L}_i}$ based on timestamp correspondance of the poses.
  if *No. of corresponding poses* = N **then**
    Compute $\mathcal{I}_{N\times N}(\boldsymbol{\omega}_{\mathtt{B}_i}, \boldsymbol{\omega}_{\mathtt{I}_i})$ using Eq. 31.
    if *Minimum singular value of* $\mathcal{I}_{N\times N} \geq \epsilon$ **then**
      $\mathbf{T}_{\mathtt{B}} := \mathbf{T}_{\mathtt{B}} \cup \{\mathbf{T}_{\mathtt{B}_i}\}$, $\forall i = 1(1)N$.
      $\mathbf{T}_{\mathtt{L}} := \mathbf{T}_{\mathtt{L}} \cup \{\mathbf{T}_{\mathtt{L}_i}\}$, $\forall i = 1(1)N$.
      Compute $\hat{\mathbf{R}}_{\mathtt{BL}}$ between $\mathbf{T}_{\mathtt{B}}$ and $\mathbf{T}_{\mathtt{L}}$ by solving for $\mathbf{K}$ using Eq. 26.
      if $\mathtt{R}_{\mathtt{Error}}(\hat{\mathbf{R}}_{\mathtt{BL}}) \leq \mathtt{R}_{\mathtt{Error}}(\mathbf{R}_{\mathtt{BL}})$ **then**
        $\mathbf{R}_{\mathtt{BL}} := \hat{\mathbf{R}}_{\mathtt{BL}}$.
        Solve for $\hat{\mathbf{t}}$ using Eq. 29.
        if $\mathtt{Error}(\hat{\mathbf{T}}_{\mathtt{BL}}) \leq \mathtt{Error}(\mathbf{T}_{\mathtt{BL}})$ **then**
          // Update current extrinsics
          $\mathbf{T}_{\mathtt{BL}} := \hat{\mathbf{T}}_{\mathtt{BL}}$.
    else
      Discard the set of current N poses.
$\mathbf{T}^\star_{\mathtt{BL}} := \mathbf{T}_{\mathtt{BL}}$

where, $\boldsymbol{\omega}_{\mathtt{I}_i^{(k)}}$ and $\boldsymbol{\omega}_{\mathtt{B}_i}$ are the angular velocities of $\mathtt{I}^{(k)}$ and $\mathtt{B}$ at timestamp t_i. Since, Eq. 30 has similar form as Eq. 6 we solve it iteratively with the update step as,

$$\underbrace{\left(\sum_i \mathbf{J}_i^\top \Sigma_i^{-1}\mathbf{J}_i\right)}_{\text{Fisher information matrix}} \boldsymbol{\Delta}\mathbf{R}_{\mathtt{BI}} \overset{Eq.\ 8}{=} -\sum_i \mathbf{J}_i^\top \Sigma_i^{-1}(\hat{\mathbf{R}}_{\mathtt{BI}}\boldsymbol{\omega}_{\mathtt{B}_i} - \boldsymbol{\omega}_{\mathtt{I}_i}) \tag{31}$$

where, $\boldsymbol{\Sigma}_i = \text{cov}(\hat{\mathbf{R}}_{\mathtt{BI}}\boldsymbol{\omega}_{\mathtt{B}_i} - \boldsymbol{\omega}_{\mathtt{I}_i})$ 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. Since, we compute this online we do processing in a pre-defined batch size of N. We perform a Singular Value Decomposition of $\mathcal{I}_{N\times N}$ for each batch as:

$$\mathcal{I}_{N\times N} = \mathbf{U}\mathbf{S}\mathbf{U}^T, \tag{32}$$

where, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2 \ldots, \mathbf{u}_N]$ and $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2 \ldots, \sigma_N)$ is a diagonal matrix of singular values in decreasing order. The value of the minimum singular value gives an indication about the information in the batch data. If the minimum singular value is more than an threshold (design choice) we conclude the data in the batch is sufficiently excited for reliable extrinsics computation. If the batch of IMU signals provide enough information, the estimated poses from the lidar odometry are selected for online calibration computation as described in Algorithm 1.

7) Stopping criteria: In our Algorithm 1, we trigger the optimization process when a pre-defined batch of poses have been accumulated. However, instead of considering only the current batch of poses we consider the full set of accumulated poses from the beginning of operation to encapsulate wider excitation of different degrees of freedom. At each iteration we compute the normalized cost function and update the estimated extrinsic parameters, if it improves the cost function. Based on our empirical analysis we stop the process once our cost function is below a certain threshold. We define our cost function in $\text{SE}(3)$ using properties from Riemannian geometry [33] as,

$$\xi_{\mathbf{T}^\star_{\text{BL}}}(\mathbf{T}_{\text{B}_i}, \mathbf{T}_{\text{L}_i}) = \underbrace{\|\log(\mathbf{R}^\star_{\text{BL}}\mathbf{R}_{\text{B}_i}\mathbf{R}^T_{\text{L}_i})\|^2}_{\texttt{Rotation Cost}=\xi_{\mathbf{R}_{\text{BL}}}} + \underbrace{\|\mathbf{R}^\star_{\text{BL}}\mathbf{t}_{\text{B}_i} + \mathbf{t}^\star_{\text{BL}} - \mathbf{t}_{\text{L}_i}\|^2}_{\texttt{Translation Cost}=\xi_{\mathbf{t}_{\text{BL}}}}, \tag{33}$$

$$\text{where, } \log(\mathbf{R}) = \begin{cases} 0, & \theta = 0 \\ \frac{\theta}{2\sin\theta}\left(\mathbf{R} - \mathbf{R}^\top\right) & |\theta| \in (0, \pi) \end{cases}$$

and, $\text{Tr}(\mathbf{R}) = 1 + 2\cos\theta$.

8) Sensitivity analysis: Here we provide an analytical study of the effect of additive noise on our optimization problem.

(i) *Rotation noise component*: First we consider the rotational noisy components. We assume that the additive noise still constructs a rotation matrix in $\text{SE}(3)$. Considering the GNSS and the estimated lidar rotational components with noise as $\mathbf{R}_{\text{B}_i} + \boldsymbol{\eta}_{\text{B}_i}$ and $\mathbf{R}_{\text{BL}_i} + \boldsymbol{\eta}_{\text{L}_i}$, our optimization problem becomes,

$$\mathbf{R}^\star_{\text{BL}} = \underset{\mathbf{R}_{\text{BL}}, \boldsymbol{\eta}_{\text{B}_i}, \boldsymbol{\eta}_{\text{L}_i}}{\arg\min} \sum_{i=1}^{N} \left(\|\boldsymbol{\eta}_{\text{B}_i}\|^2_{\Sigma^{-1}_{\text{BB}}} + \|\boldsymbol{\eta}_{\text{L}_i}\|^2_{\Sigma^{-1}_{\text{LL}}} \right) \tag{34}$$

$$s.t. \quad \mathbf{R}_{\text{BL}}(\mathbf{R}_{\text{B}_i} + \boldsymbol{\eta}_{\text{B}_i}) = (\mathbf{R}_{\text{L}_i} + \boldsymbol{\eta}_{\text{L}_i}) \ \ \forall i. \tag{35}$$

Thus, rearranging Eq. 35 would yield,

$$\boldsymbol{\eta}_{\text{L}_i} = (\mathbf{R}_{\text{BL}}\mathbf{R}_{\text{B}_i} - \mathbf{R}_{\text{L}_i}) + \mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i}. \tag{36}$$

Considering equal co-variance norms we can re-write the error term, Ξ at t_i as,

$$\Xi(\boldsymbol{\eta}_{\text{B}_i}, \boldsymbol{\eta}_{\text{L}_i}) = \|\boldsymbol{\eta}_{\text{B}_i}\|^2 + \|\boldsymbol{\eta}_{\text{L}_i}\|^2 \tag{37}$$

$$= 2\|\boldsymbol{\eta}_{\text{B}_i}\|^2 + 2(\mathbf{R}_{\text{BL}}\mathbf{R}_{\text{B}_i} - \mathbf{R}_{\text{L}_i})^T\mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i} + \|\mathbf{R}_{\text{BL}}\mathbf{R}_{\text{B}_i} - \mathbf{R}_{\text{L}_i}\|^2,$$

since, $\|\mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i}\|^2 = (\mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i})^T\mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i} = \boldsymbol{\eta}^T_{\text{B}_i}\mathbf{R}^T_{\text{BL}}\mathbf{R}_{\text{BL}}\boldsymbol{\eta}_{\text{B}_i} = \|\boldsymbol{\eta}_{\text{B}_i}\|^2$.

To minimize, we equate the Jacobian of $\Xi(\boldsymbol{\eta}_{\text{B}_i}, \boldsymbol{\eta}_{\text{L}_i})$ to zero.

$$\implies \quad \frac{\partial \Xi}{\partial \boldsymbol{\eta}_{\text{B}_i}} = 0$$

$$\text{or,} \quad 4\boldsymbol{\eta}_{\text{B}_i} + 2\mathbf{R}^T_{\text{BL}}(\mathbf{R}_{\text{BL}}\mathbf{R}_{\text{B}_i} - \mathbf{R}_{\text{L}_i}) = 0$$

$$\text{or,} \quad \boldsymbol{\eta}_{\text{B}_i} = -\frac{1}{2}(\mathbf{R}_{\text{B}_i} - \mathbf{R}^{-1}_{\text{BL}}\mathbf{R}_{\text{L}_i}). \tag{38}$$

So, it turns out that under noisy conditions the optimal rotation finding problem has a similar form as our original rotation optimization problem in Eq. 15.

(ii) *Translation noise component*: Let the relative noisy translation components for the lidar and the GNSS measurements between timestamps t_{i-1} and t_i be $\boldsymbol{\delta}_{\text{L}_i}$ and $\boldsymbol{\delta}_{\text{B}_i}$. Let, $\boldsymbol{\tau}_{\text{L}_i} = \mathbf{t}_{\text{L}_{i-1}\text{L}_i}$ and $\boldsymbol{\tau}_{\text{B}_i} = \mathbf{t}_{\text{B}_{i-1}\text{B}_i}$. Our optimization problem becomes,

$$\mathbf{t}^\star_{\text{BL}} = \underset{\mathbf{t}_{\text{BL}}, \boldsymbol{\delta}_{\text{B}_i}, \boldsymbol{\delta}_{\text{L}_i}}{\arg\min} \sum_{i=1}^{N-1} \left(\|\boldsymbol{\delta}_{\text{B}_i}\|^2_{\Sigma^{-1}_{\text{BB}}} + \|\boldsymbol{\delta}_{\text{L}_i}\|^2_{\Sigma^{-1}_{\text{LL}}} \right) \tag{39}$$

$$s.t. \quad (\mathbf{R}_{\text{B}_{i-1}\text{B}_i} - \mathbf{I})\mathbf{t}_{\text{BL}} = \mathbf{R}_{\text{BL}}(\tau_{\text{L}_i} + \boldsymbol{\delta}_{\text{L}_i}) + (\tau_{\text{B}_i} + \boldsymbol{\delta}_{\text{B}_i}) \ \ \forall i. \tag{40}$$

Rearranging Eq. 40 we get,

$$\boldsymbol{\delta}_{\text{L}_i} = \mathbf{R}^T_{\text{BL}}(\mathbf{R}_{\text{B}_{i-1}\text{B}_i} - \mathbf{I})\mathbf{t}_{\text{BL}} + \mathbf{R}^T_{\text{BL}}(\tau_{\text{B}_i} + \boldsymbol{\delta}_{\text{B}_i}) - \tau_{\text{L}_i}. \tag{41}$$

Considering equal co-variance norms we can minimize the error term, $\Xi(\boldsymbol{\delta}_{\text{B}_i}, \boldsymbol{\delta}_{\text{L}_i})$ by setting the Jacobians equal to zero.

$$\implies \frac{\partial \Xi}{\partial \boldsymbol{\delta}_{\text{B}_i}} = 0$$

$$\text{or, } 2\boldsymbol{\delta}_{\text{B}_i} + 2\mathbf{R}_{\text{BL}}\left[\mathbf{R}^T_{\text{BL}}(\mathbf{R}_{\text{B}_{i-1}\text{B}_i} - \mathbf{I})\mathbf{t}_{\text{BL}} + \mathbf{R}^T_{\text{BL}}(\tau_{\text{B}_i} + \boldsymbol{\delta}_{\text{B}_i}) - \tau_{\text{L}_i}\right] = 0$$

$$\text{or, } 2\boldsymbol{\delta}_{\text{B}_i} + (\mathbf{R}_{\text{B}_{i-1}\text{B}_i} - \mathbf{I})\mathbf{t}_{\text{BL}} - (\mathbf{R}_{\text{BL}}\tau_{\text{L}_i} - \tau_{\text{B}_i}) = 0$$

$$\text{or, } \boldsymbol{\delta}_{\text{B}_i} = -\frac{1}{2}\left[(\mathbf{R}_{\text{B}_{i-1}\text{B}_i} - \mathbf{I})\mathbf{t}_{\text{BL}} - (\mathbf{R}_{\text{BL}}\mathbf{t}_{\text{L}_{i-1}\text{L}_i} - \mathbf{t}_{\text{B}_{i-1}\text{B}_i})\right]. \tag{42}$$

Thus, for the translation case we can also show that the original optimization problem described in Eq. 29 holds true under noisy conditions as well.

V. Experimental Results

To demonstrate the perforance of the online calibration system, we conducted several experiments using real world data collected from the two vehicles illustrated in Fig. 2 which have a different configuration.

A. Dataset

Dataset collection details for the experiments

Data	Vehicle	Sensor setup	Length (Km)	Duration (secs)
Seq-1	Bus	$\text{F}_\text{L}, \text{F}_\text{R}, \text{R}_\text{L}, \text{R}_\text{R}$	1.473	136
Seq-2	Bus	$\text{F}_\text{L}, \text{F}_\text{R}$	2.086	135
Seq-3	Bus	$\text{F}_\text{L}, \text{F}_\text{R}$	1.089	50
Seq-4	Truck	$\text{F}_\text{L}, \text{F}_\text{R}, \text{R}_\text{L}, \text{R}_\text{R}$	0.478	67
Seq-5	Truck	$\text{F}_\text{L}, \text{F}_\text{R}, \text{R}_\text{L}, \text{R}_\text{R}$	1.212	73
Seq-6†	Truck	$\text{F}_\text{L} - \text{Top}, \text{F}_\text{R} - \text{Top}$	0.224	164
Seq-7†	Truck	$\text{F}_\text{L} - \text{Top}, \text{F}_\text{R} - \text{Top}$	1.088	263

†*Front 2 top lidars mounted on the cab were used here.*

TABLE I

We use data from a GNSS receiver to produce ground truth (GT) motion trajectories of the B frame. We analyze our results on 7 collected test sequences described in Table I.

B. Online calibration results and repeatability analysis

For the online calibration process we start from a set of un-calibrated lidars for which only the translation components from the vehicle CAD parameters are known. A GT calibration is performed offline by refining the vehicle CAD parameters. The refinement process analyzes known static feature positions around the vehicle and matches the corresponding detected features between the lidars. We compare our results to these GT parameters for both vehicles. We measured translation and rotation error as:

$$\Delta t = \frac{1}{3}\sqrt{\|\widehat{\mathbf{t}} - \mathbf{t}_{\mathrm{GT}}\|_F^2}, \tag{43}$$

$$\Delta R = \frac{180}{\pi}\cos^{-1}\left[\frac{1}{2}\left(\mathrm{Tr}(\widehat{\mathbf{R}}^{-1}\mathbf{R}_{\mathrm{GT}}) - 1\right)\right], \tag{44}$$

where, ΔR is the rotation along the principle eigenvector of $(\widehat{\mathbf{R}}^{-1}\mathbf{R}_{\mathrm{GT}})$. We summarize our results in Table II and compare our performance to the Kabsch alignment [17] technique.

In Table II, we show the average of Δt and ΔR for the bus and truck sequences. It can be seen that our online method produces better results for translation components because of the bounded variable least square optimization 9. Our method also produces comparable results for rotation components outperforming Kabsh alignment in most scenarios.

Average performance comparison of calibration routines

Vehicle	Lidar position	Kabsch alignment		Ours online method	
		Δt [m]	ΔR [°]	Δt [m]	ΔR [°]
Bus	$\mathrm{F_L}$	10.603	**4.370**	**0.766**	4.619
	$\mathrm{F_R}$	16.484	2.167	**0.333**	**2.069**
	$\mathrm{R_R}$	3.301	**0.249**	**0.413**	0.285
	$\mathrm{R_L}$	3.246	0.257	**0.359**	**0.500**
Truck	$\mathrm{F_L}$	9.384	11.968	**0.318**	**5.301**
	$\mathrm{F_R}$	8.583	2.566	**0.340**	**2.006**
	$\mathrm{R_R}$	2.536	**1.192**	**0.311**	2.113
	$\mathrm{R_L}$	2.453	**0.594**	**0.506**	1.118
	$\mathrm{F_L}$ − Top	9.637	2.462	**0.346**	**2.091**
	$\mathrm{F_R}$ − Top	8.907	3.387	**0.388**	**2.141**

TABLE II

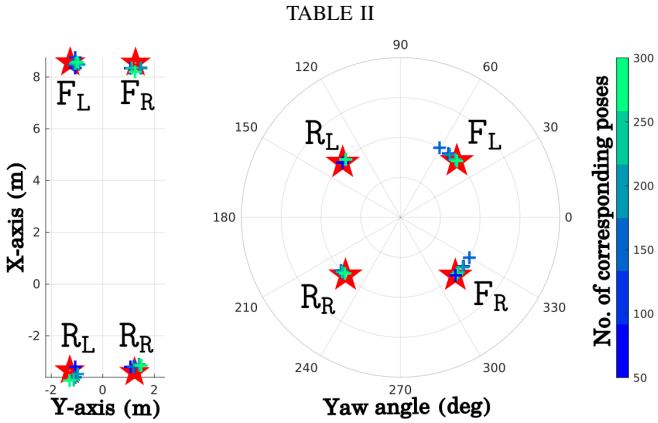

Fig. 4. Estimated t_x, t_y and Yaw for bus sequence-1 over different pose sizes. The GT calibration parameters are represented in red color.

We also show the calibration results for the individual transformation components in Fig. 4 for bus and Fig. 5 truck for different trajectories. Since, we compare the estimated lidar pose in sensor frame to the corresponding GNSS pose in B frame for calibration we study the effect of the size of the associated corresponding poses used for calibration.

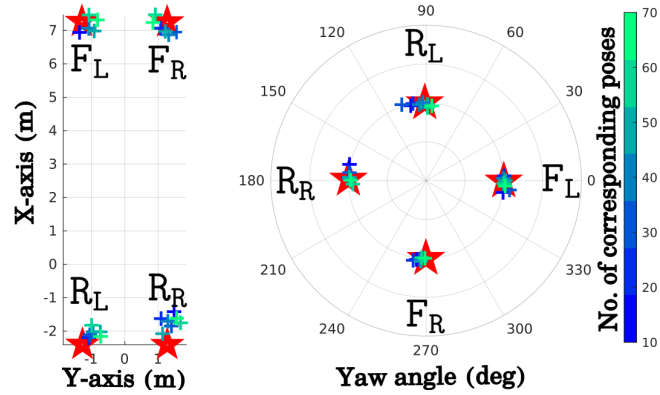

Fig. 5. Estimated t_x, t_y and Yaw for truck sequence-4 over different pose sizes. The GT calibration parameters are represented in red color.

As seen in both the sequences, the rotation components converges to the prior GT calibration as the number of corresponding poses in the optimization increases. We solved for the translation components using a bounded variable problem (± 0.3m) since the initial CAD estimates are reliable and found it converged towards the GT parameters as well.

After running our algorithm we calibrate the lidars online as seen in Fig. 6. Additionally, we demonstrate the online calibration process in real-time in our supplementary video.

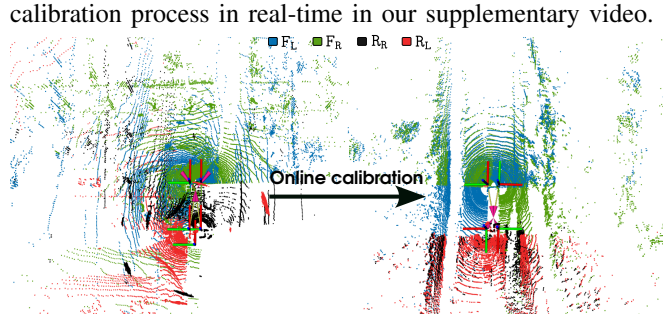

Fig. 6. Online calibration results for sequence-4.

C. Observability analysis

For our observability analysis we extract the information matrix from the angular velocity signals. We only choose to estimate calibration in segments for which there is enough excitation in the angular velocity. These segments are highlighted in green in Fig. 7.

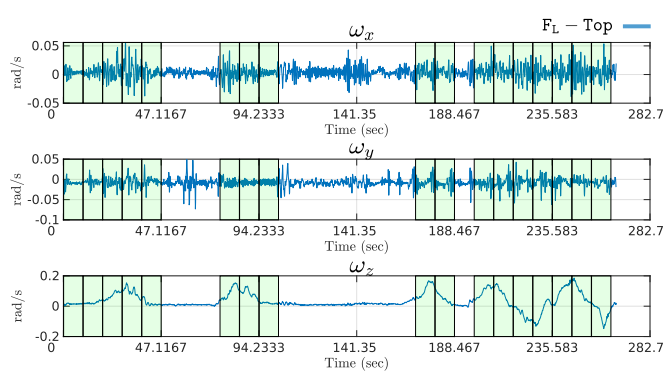

Fig. 7. Illustration of observability analysis for Seq-7 with angular velocity.

The corresponding trajectory is shown in Fig. 8. We can see that the trajectory segments selected for calibration correspond are those with significant turning maneuvers.

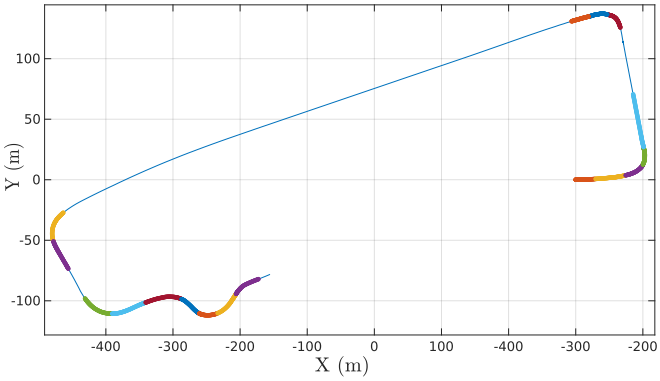

Fig. 8. Illustration of selected poses with observability analysis for Seq-7.

VI. Conclusion

We have presented an observability-aware online multi-lidar calibration algorithm. The observability aware module selects trajectory sequences that carry enough excitation for the different degrees of freedom required for calibration. Additionally, we provide an analytical study of the effect of noise on the estimated states used for calibration. We also provide a stopping measure which halts the calibration when a satisfactory performance has been achieved.

Our method offers equivalent performance for rotation estimate and superior outcomes for translation estimate when compared to trajectory alignment method. For future work we will develop a method which can generate a driving route containing observability-aware trajectories which the vehicle can follow to perform online calibration in a minimum amount of time possible.

VII. Acknowledgements

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

References

[1] 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/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1828–1835.

[2] 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.

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

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

[5] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.

[6] K. Huang and C. Stachniss, "On geometric models and their accuracy for extrinsic sensor calibration," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4948–4954.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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, no. 10, pp. 3846–3860, 2019.

[12] 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.

[13] R. Horaud and F. Dornaika, "Hand-eye calibration," *International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[14] M. Shah, R. D. Eastman, and T. Hong, "An overview of robot-sensor calibration methods for evaluation of perception systems," in *Workshop on Performance Metrics for Intelligent Systems*, 2012, pp. 15–20.

[15] J. Jiao, Y. Yu, Q. Liao, H. Ye, R. Fan, and M. Liu, "Automatic calibration of multiple 3D lidars in urban environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 15–20.

[16] S. Das, N. Mahabadi, A. Djikic, C. Nassir, S. Chatterjee, and M. Fallon, "Extrinsic calibration and verification of multiple non-overlapping field of view lidar sensors," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 919–925.

[17] 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.

[18] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

[19] K. Huang and C. Stachniss, "Extrinsic multi-sensor calibration for mobile robots using the Gauss-Helmert model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1490–1496.

[20] P. B. Davenport, *A vector approach to the algebra of rotations with applications*. National Aeronautics & Space Admin., 1968, vol. 4696.

[21] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[22] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, pp. 1–18, 2022.

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

[24] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5135–5142.

[25] 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.

[26] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

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

[28] D. W. Allan, "Statistics of atomic frequency standards," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 221–230, 1966.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

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

[31] K. Hoffman, *Linear algebra*. Prentice-Hall, 1971.

[32] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[33] P. Petersen, *Riemannian geometry*. Springer, 2006, vol. 171.