Uncertainty-aware Hand-Eye Decalibration Detection via the Gauss-Helmert Model

Marta Čolaković-Bencerić Ivan Marković
Fabio Bonsignorio Ivan Petrović *

* *University of Zagreb Faculty of Electrical Engineering and Computing, Croatia (e-mail: marta.colakovic@fer.unizg.hr) (e-mail: ivan.markovic@fer.unizg.hr) (e-mail: fabio.bonsignorio@fer.unizg.hr) (e-mail: ivan.petrovic@fer.unizg.hr)*

Abstract: Detecting decalibration is crucial to maintain perception accuracy and prevent performance degradation in long-term autonomous operations, where environmental changes, mechanical wear, and external forces can compromise sensor alignment. In this paper, we propose a motion-based framework that leverages uncertainty-aware optimization for accurate decalibration detection and correction. Key contributions include a batch calibration procedure based on the Gauss-Helmert model with variance component estimation and a probabilistic detection criterion using the Mahalanobis distance. Finally, we develop an online detection system employing a sliding-window approach for continuous monitoring, demonstrating particularly rapid response to rotational misalignment.

Keywords: decalibration detection; extrinsic calibration; Gauss-Helmert model; perception and sensing; mobile robots

1. INTRODUCTION

Modern robotic systems, designed to operate in highly dynamic environments, must reliably solve complex tasks such as environment perception, motion prediction, motion planning, and control. Environment perception, as the foundation of the autonomy pipeline, is heavily dependent on data from multiple heterogeneous sensors. Extrinsic calibration is responsible for establishing the spatial relationships between the coordinate systems of various sensors and robot components, enabling effective sensor fusion. However, the accuracy of these estimates can degrade over time due to factors such as environmental changes, mechanical wear, or external forces, resulting in decalibration.

Targetless methods of extrinsic calibration, including feature-based and motion-based approaches, utilize information available in the robot's environment, making them particularly well-suited for decalibration detection. Feature-based methods establish correspondences using natural environmental landmarks, while motion-based methods, such as odometry-based calibration, estimate sensor alignments based on ego-motion. Odometry-based methods face challenges like motion noise and lower accuracy compared to target-based methods. However, odometry-based calibration allows for easy integration of data from heterogeneous sensors, is sensor-agnostic, and utilizes odometry information, which is commonly available in robotic systems.

* This research has been supported by the European Union's Horizon 2020 research and innovation program under the Grant Agreement No. 952275 (AIFORS).

Given the complexities of multisensor systems and the importance of maintaining accurate calibration, robust methods for detecting and correcting decalibration are essential. This paper introduces a motion-based framework for hand-eye decalibration detection and correction, leveraging uncertainty-aware optimization.

Key contributions of this work include:

- A batch calibration procedure utilizing the Gauss-Helmert model, incorporating variance component estimation for improved robustness.
- A probabilistic decalibration detection criterion based on the Mahalanobis distance.
- An online decalibration detection system using a sliding window approach for continuous monitoring.

2. RELATED WORK

Targetless calibration methods can be broadly categorized into feature-based and odometry-based approaches, each with distinct underlying assumptions, data requirements, and operational constraints. Feature-based calibration methods rely on identifiable environmental structures, such as edges, corners, or visual patterns, to align sensor outputs (Levinson and Thrun (2013); Jiang et al. (2018)). These methods are highly accurate when the environment is structured and features are clearly identifiable. However, feature-based methods often require specific sensor combinations, limiting their general applicability (Scaramuzza et al. (2007)).

In contrast, odometry-based calibration methods leverage sensor motion to estimate calibration parameters (Deray et al. (2019); Huang and Stachniss (2017)). These methods

typically require only temporal synchronization and metrically scaled 6DoF motion data, making them more flexible across a wide range of heterogeneous sensors. However, odometry-based approaches often exhibit lower accuracy compared to feature-based methods due to reliance on motion data, which can be susceptible to noise.

A critical consideration for robotic systems is achieving continuous operation. One class of approaches involves continuous batch calibration (Lv et al. (2020); Ishikawa et al. (2018)), which focuses on recent data and marginalizes old data to perform calibration. Other methods incorporate calibration estimation into a filtering framework, enabling dynamic calibration in real time (Schneider et al. (2013)). Finally, online calibration can be achieved through decalibration monitoring and detection, emphasizing the detection of decalibration events and triggering recalibration only when necessary. This proactive strategy reduces unnecessary calibration efforts and allocates computational resources efficiently by focusing on significant deviations.

Approaches to decalibration detection vary and are closely tied to the underlying calibration procedure. For example, in (Peršić et al. (2021)), the authors propose a computationally efficient method that solves the orthogonal Procrustes problem over a window of tracked object positions to obtain pairwise calibration. The system monitors rotational decalibration and, when the magnitude exceeds a predefined threshold, recalibration is initiated. Similarly, (Deray et al. (2019)) achieves decalibration detection by monitoring the total cost over a sliding window of variable length, triggering recalibration when a heuristic threshold is exceeded. In (Levinson and Thrun (2013)), decalibration is detected by tracking the overlap of edges detected in lidar point clouds and camera images, with a probabilistic threshold used to assess deviations. Likewise, the method in (Jiang et al. (2018)) utilizes the average projection error of a point cloud to trigger recalibration. The authors of (Moravec and Šára (2024)) combine a joint saliency objective function with kernel correlation to quantify the alignment, providing a robust measure for decalibration detection.

Despite the growing interest in decalibration detection for feature-based methods, it remains less explored for odometry-based calibration. This paper addresses this gap by leveraging the calibration parameter and covariance estimation capabilities of an estimator based on the Gauss-Helmert (GH) model (Huang and Stachniss (2017)). The GH model has been extensively developed and studied for applications such in geodesy and scan matching (Wang et al. (2023); Vogel et al. (2022)), and stereo-camera calibration (Dang et al. (2009)). However, its application to odometry-based self-calibration has received limited attention. To formulate a probabilistic decalibration criterion based on covariance propagation, it is necessary to estimate the unknown measurement variances. Variance component estimation for the Gauss-Helmert model has been addressed in (Wang et al. (2024)) and (Teunissen and Amiri-Simkooei (2008)), providing a robust foundation for accurate covariance estimation in this context.

3. HAND-EYE CALIBRATION

Odometry-based extrinsic calibration determines the rigid transformations between sensor coordinate frames by leveraging odometry data. A common approach to this problem is the hand-eye problem formulation, which estimates the rigid transformations using their ego-motion data. This method is applicable to any sensor capable of determining its own motion, such as lidars, cameras, or IMUs.

3.1 Problem formulation

We denote the sensors of our system as $A, B, \ldots, M$, each rigidly attached to the robot. Ego-motion of a sensor between time instances t and $t+1$, is defined as:

$$\mathbf{A}_t = (\mathbf{M}_t^A)^{-1}\mathbf{M}_{t+1}^A, \quad \mathrm{Log}(\mathbf{A}_t) = \mathbf{l}_t^A, \tag{1}$$

where $\mathbf{M}_t^A$ represents the pose of sensor A at time t. At each time-step i, we have a set of ego-motions of each sensor consisting of an element of the vector space isomorphic to the tangent Lie algebra of the *SO(3)*, $\mathbf{r} = \mathrm{Log}(\mathbf{R}) \in \mathbb{R}^3 \cong \mathfrak{so}(3) = TSO(3)$, and a vector $\mathbf{t} \in \mathbb{R}^3$,

$$\mathbf{l}_i = [\, \mathbf{t}_{Ai}\ \mathbf{r}_{Ai}\ \mathbf{t}_{Bi}\ \mathbf{r}_{Bi}\ \ldots\ \mathbf{t}_{Mi}\ \mathbf{r}_{Mi}\,]^\top \tag{2}$$

where $i = 1, \ldots, N$, with N representing the number of ego-motions. The logarithmic mapping, Log, transforms elements of the Lie group into vectors belonging to a vector space isomorphic to its tangent Lie algebra, while the exponential map, Exp, performs the inverse operation,

$$\mathrm{Log}(\mathbf{R}) = \frac{r(\mathbf{R} - \mathbf{R}^\top)^\vee}{2\sin(r)}, \tag{3}$$

$$\mathrm{Exp}(\mathbf{r}) = \exp(\mathbf{r}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!}(\mathbf{r}^\wedge)^n, \tag{4}$$

where

$$\mathbf{r}^\wedge = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{5}$$

$(\mathbf{r}^\wedge)^\vee = \mathbf{r}$ and $r = |\mathbf{r}|$ (Barfoot (2017)). We define the incrementation strategy on the *SO(3)* group as multiplicative and establish an additive equivalent,

$$\delta\mathbf{R} \cdot \mathbf{R} = \mathrm{Exp}(\delta\mathbf{r}) \cdot \mathrm{Exp}(\mathbf{r}) \simeq \mathrm{Exp}\left(\mathbf{r} + \mathbf{J}_l^{-1}(\mathbf{r})\delta\mathbf{r}\right), \tag{6}$$

where $\mathbf{J}_l^{-1}$ is the inverse of the *SO(3)* left group Jacobian,

$$\mathbf{J}_l^{-1}(\mathbf{r}) = \mathbf{I}_3 - \frac{1}{2}\mathbf{r}^\wedge + \left(\frac{1}{r^2} - \frac{1+\cos(r)}{2r\sin(r)}\right)\mathbf{r}^{\wedge 2}. \tag{7}$$

The *hand-eye* problem, depicted in Fig.1, is expressed by the equation

$$\mathbf{AX} = \mathbf{XB}, \tag{8}$$

where $\mathbf{A}$ and $\mathbf{B}$ denote sensors' ego-motion, and $\mathbf{X}$ the rigid transformation between their coordinate frames. Following (Huang and Stachniss (2017)), we define the sensor A as the base sensor and reach the *hand-eye* constraints

$$\mathbf{g}_i(\boldsymbol{\xi}, \mathbf{l}_i) = \begin{bmatrix} (\mathbf{R}_{Ai} - \mathbf{I}_3)\mathbf{t} - \mathbf{R}\mathbf{t}_{Bi} + \mathbf{t}_{Ai} \\ \mathbf{r}_{Ai} - \mathbf{R}\mathbf{r}_{Bi} \\ \vdots \\ (\mathbf{R}_{Ai} - \mathbf{I}_3)\mathbf{t} - \mathbf{R}\mathbf{t}_{Mi} + \mathbf{t}_{Ai} \\ \mathbf{r}_{Ai} - \mathbf{R}\mathbf{r}_{Mi} \end{bmatrix}, \tag{9}$$

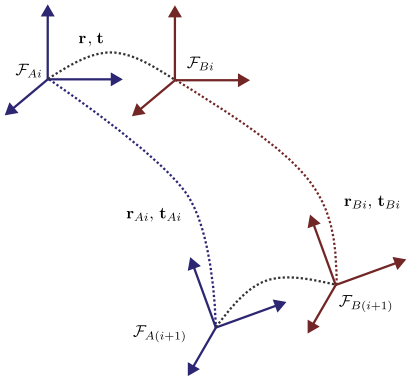


Fig. 1. The proposed method calculates the transformation between frames of sensors A and B, denoted by $\mathbf{r}$, $\mathbf{t}$. The measurements or ego-motions are symbolized by $\mathbf{r}_{Ai}$, $\mathbf{t}_{Ai}$ and $\mathbf{r}_{Bi}$, $\mathbf{t}_{Bi}$. The ego-motions denote the relative motion of sensors between timestamps $i+1$ and i.

with the unknown calibration parameters are denoted by

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{r}_1 & \dots & \mathbf{t}_{Nsensor} & \mathbf{r}_{Nsensor} \end{bmatrix}^\top, \tag{10}$$

where N_{sensor} denotes the number of sensors of the system.

3.2 Hand-eye calibration via Gauss-Helmert model

Following (Huang and Stachniss (2017)), we utilize the Gauss-Helmert model to address calibration deterioration in high-noise scenarios, which iteratively adjusts both unknown parameters and observations. By strictly enforcing constraints, the model minimizes observation corrections while refining parameter estimates. Additionally, we incorporate uncertainty-aware weighting to achieve a probabilistic and robust calibration approach.

With the total observation corrections defined as

$$\boldsymbol{\epsilon}_i = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{t}Ai} & \boldsymbol{\epsilon}_{\mathbf{r}Ai} & \boldsymbol{\epsilon}_{\mathbf{t}Bi} & \boldsymbol{\epsilon}_{\mathbf{r}Bi} & \dots & \boldsymbol{\epsilon}_{\mathbf{t}Mi} & \boldsymbol{\epsilon}_{\mathbf{r}Mi} \end{bmatrix}^\top, \tag{11}$$

the unknown parameters are estimated by minimizing

$$\begin{aligned} \underset{\Delta\boldsymbol{\xi},\, \Delta\mathbf{l}_i}{\operatorname{argmin}} & \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\epsilon}_i^\top \Sigma_{\mathbf{ll}\,i}^{-1} \boldsymbol{\epsilon}_i \\ \text{s.t. } & \mathbf{g}_i(\boldsymbol{\xi}, \mathbf{l}_i) = \mathbf{0}, \end{aligned} \tag{12}$$

while strictly enforcing constraints $\mathbf{g}_i(\boldsymbol{\xi}, \mathbf{l}_i) = \mathbf{0}$. Here $\Delta\mathbf{l}_i$ represents optimal corrections to observation i and $\Delta\boldsymbol{\xi}$ optimal corrections to the calibration parameters. These corrections allow us to calculate the updated measurements $\mathbf{l}_i^{new} = \Delta\mathbf{l}_i \oplus \mathbf{l}_i^{prev}$ and the updated total accumulated corrections to observations $\boldsymbol{\epsilon}_i^{new} = \Delta\mathbf{l}_i \oplus \boldsymbol{\epsilon}_i^{prev}$, as well as $\boldsymbol{\xi}^{new} = \Delta\boldsymbol{\xi} \oplus \boldsymbol{\xi}^{prev}$. The $\oplus$ operator is defined as

$$\Delta\mathbf{l} \oplus \mathbf{l} = \begin{bmatrix} \mathbf{t} + \delta\mathbf{t} \\ \mathbf{r} + \mathbf{J}_l^{-1}(\mathbf{r})\delta\mathbf{r} \end{bmatrix}. \tag{13}$$

The constraints are then linearized

$$\mathbf{Y}_i \Delta\boldsymbol{\xi} + \mathbf{Z}_i \Delta\mathbf{l}_i + \mathbf{g}(\boldsymbol{\xi}^{prev}, \mathbf{l}_i^{prev}) = 0, \tag{14}$$

with $\mathbf{Y}_i$ and $\mathbf{Z}_i$ being Jacobians of the constraints with respect to the unknown parameters and the observations. Finally, the optimal solution is found by formulating the Lagrange function using (14),

$$\Delta\boldsymbol{\xi} = \left(\sum_{i=1}^{N} \mathbf{Y}_i^\top \mathbf{W}_i \mathbf{Y}_i \right)^{-1} \cdot \sum_{i=1}^{N} \mathbf{Y}_i^\top \mathbf{W}_i \mathbf{c}_i, \tag{15}$$

$$\Delta\mathbf{l}_i = \Sigma_{\mathbf{ll}\,i} \mathbf{Z}_i^\top \mathbf{W}_i \cdot \left(\mathbf{c}_i - \mathbf{Y}_i \Delta\boldsymbol{\xi} \right) - \mathbf{U}_i^{-1} \boldsymbol{\epsilon}_i. \tag{16}$$

In (15) and (16) $\mathbf{W}_i$ represents the weighting matrices $\left(\mathbf{Z}_i \Sigma_{\mathbf{ll}\,i} \mathbf{Z}_i^\top \right)^{-1}$, while vector $\mathbf{c}_i$ equals $\mathbf{c}_i = \mathbf{Z}_i \mathbf{U}_i^{-1} \boldsymbol{\epsilon}_i - \mathbf{g}_i$,

$$\mathbf{U}_i = \operatorname{diag}\left(\mathbf{I}^3 \;\; \mathbf{J}_l^{-1}(\boldsymbol{\epsilon}_{\mathbf{r}Ai}) \;\; \mathbf{I}^3 \;\; \mathbf{J}_l^{-1}(\boldsymbol{\epsilon}_{\mathbf{r}Bi}) \;\; \dots \right). \tag{17}$$

The covariance of estimated parameters equals

$$\Sigma_{\boldsymbol{\xi}\boldsymbol{\xi}} = \left(\sum_{i=1}^{N} \mathbf{Y}_i^\top \mathbf{W}_i \mathbf{Y}_i \right)^{-1}. \tag{18}$$

3.3 Least squares variance component estimation

In extrinsic calibration, accurate weighting of observations is critical to achieving reliable parameter estimation. Accurate covariance information is rarely provided alongside odometry data, despite its role in extrinsic calibration. Most odometry estimation procedures either omit covariance estimates entirely or rely on simplistic models that fail to capture the true uncertainties. A robust solution to the absence of accurate covariance information in odometry data is the application of least squares variance component estimation.

Following the representation of ego-motion from Subsection 3.1 we define the stochastic model of ego-motion noise for a single sensor as

$$\delta\mathbf{R} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\mathbf{R}}^2 \mathbf{I}_3\right), \quad \delta\mathbf{t} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\mathbf{t}}^2 \mathbf{I}_3\right), \tag{19}$$

Additionally, we assume that ego-motion noise is uncorrelated between different time instances and sensors, with $\sigma_{\mathbf{t}}^2$, $\sigma_{\mathbf{R}}^2$ permitted to be different for each individual sensor. As a consequence of multiplicative nature of on manifold incrementation of rotation

$$\delta\mathbf{l}_i \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma_{\mathbf{t}}^2 \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_{\mathbf{R}}^2 \mathbf{J}_l^{-1}(\mathbf{r}_i) \mathbf{J}_l^{-1\top}(\mathbf{r}_i) \end{bmatrix}\right). \tag{20}$$

From the analytic expression for $\mathbf{J}_l^{-1}(\mathbf{r}_i)\mathbf{J}_l^{-1\top}(\mathbf{r}_i)$, we observe that for $r_i \in [0°, 40°]$ the diagonal entries do not deviate more than 5% from 1. Due to average magnitude of rotation being 3°, we can approximate $\mathbf{J}_l^{-1}(\mathbf{r}_i)\mathbf{J}_l^{-1\top}(\mathbf{r}_i)$ with the identity matrix. This approximation simplifies the computation and improves the efficiency of the calibration process by disregarding the off diagonal elements, while maintaining acceptable accuracy for our application. Because we employ least-squares variance component estimation (LS-VCE), the calibration procedure inherently accounts for any inflation of the diagonal elements, making the stochastic model

$$\delta\mathbf{l} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma_{\mathbf{t}}^2 \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_{\mathbf{R}}^2 \mathbf{I}_3 \end{bmatrix}\right) \tag{21}$$

or more precisely

$$\begin{aligned} \Sigma_{\mathbf{ll}} &= \operatorname{diag}(\sigma_{\mathbf{t}A}^2 \mathbf{I}_3, \sigma_{\mathbf{R}A}^2 \mathbf{I}_3, \sigma_{\mathbf{t}B}^2 \mathbf{I}_3, \sigma_{\mathbf{R}B}^2 \mathbf{I}_3, \dots) \\ &= \operatorname{diag}(\sigma_1^2 \mathbf{I}_3, \sigma_2^2 \mathbf{I}_3, \sigma_3^2 \mathbf{I}_3, \sigma_4^2 \mathbf{I}_3, \dots) \end{aligned} \tag{22}$$

The classical LS-VCE formulae cannot be directly applied to estimate variance components in the nonlinear Gauss-Helmert (GH) model due to differences in their mathematical frameworks (Wang et al. (2024)). To address this we follow (Wang et al. (2024)) and reformulate the parameter solution of the nonlinear GH model (16) into an equivalent standard least-squares structure

$$\mathbf{e}_i = \mathbf{c}_i - \mathbf{Y}_i \Delta\boldsymbol{\xi} = \mathbf{P}_i \mathbf{c}_i \tag{23}$$

with

$$\Sigma_{\mathbf{e}i} = \mathbf{Z}_i \Sigma_{\mathbf{ll}} \mathbf{Z}_i^\top, \tag{24}$$

and

$$\mathbf{P}_i = \mathbf{I} - \mathbf{Y}_i \Sigma_{\boldsymbol{\xi}\boldsymbol{\xi}} \mathbf{Y}_i^\top \mathbf{W}_i, \tag{25}$$

where $\mathbf{P}_i$ is an orthogonal projector. We express the observation covariance as

$$\Sigma_{\mathbf{ll}} = \sum_{k=1}^{2Nsensor} \sigma_k^2 \mathbf{Q}_{\mathbf{l}k}, \tag{26}$$

where $\mathbf{Q}_{\mathbf{l}k}$ denote the a priori cofactor matrices and σ_k^2 the unknown variance components. It follows that

$$\mathbf{W}_i^{-1} = \Sigma_{\mathbf{e}i} = \mathbf{Z}_i \Sigma_{\mathbf{ll}} \mathbf{Z}_i^\top = \sum_{k=1}^{2N_{\text{sensor}}} \sigma_k^2 \mathbf{Z}_i \mathbf{Q}_{\mathbf{l}k} \mathbf{Z}_i^\top, \tag{27}$$

and $\mathbf{Q}_{\mathbf{e}ki} = \mathbf{Z}_i \mathbf{Q}_{\mathbf{l}k} \mathbf{Z}_i^\top$. Per (Teunissen and Amiri-Simkooei (2008)) variance components $\boldsymbol{\sigma}^2 = [\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \dots]$ can now be estimated as

$$\hat{\boldsymbol{\sigma}}^2 = \mathbf{N}^{-1} \mathbf{v}, \tag{28}$$

with components of $\mathbf{N}$ and $\mathbf{v}$ given by

$$n_{j,k} = \frac{1}{2} \sum_i \operatorname{tr} \left(\mathbf{Q}_{\mathbf{e}ji} \mathbf{W}_i \mathbf{P}_i \mathbf{Q}_{\mathbf{e}ki} \mathbf{W}_i \mathbf{P}_i \right), \tag{29}$$

$$v_j = \frac{1}{2} \sum_i \mathbf{e}_i^\top \mathbf{W}_i \mathbf{Q}_{\mathbf{e}ji} \mathbf{W}_i \mathbf{e}_i, \tag{30}$$

and $j, k = 1, \dots, 2N_{sensor}$. Finally, to achieve non-negative variance, estimation was performed under the inequality constraint $\boldsymbol{\sigma}^2 \geq 10^{-12}[1, 1, 1, 1, \dots]^\top$. Joint calibration and variance component estimation are described in Alg. 1.

Algorithm 1 GH-Based Algorithm for Hand-Eye Calibration and Variance Component Estimation

Input: Ego-motions $\mathbf{l}_i$, $i = 1, \dots, N$

Output:

- Calibration $\boldsymbol{\xi}$ and uncertainty $\Sigma_{\boldsymbol{\xi}\boldsymbol{\xi}}$
- Variance components of input ego-motion $\hat{\boldsymbol{\sigma}}^2$

Procedure:

(1) **Initialization:**
- Set $\boldsymbol{\xi} \leftarrow \mathbf{0}$, $\boldsymbol{\epsilon} \leftarrow \mathbf{0}$, $\boldsymbol{\sigma}^2 \leftarrow [1, 1, 1, 1 \dots]^\top$

(2) **Iterate until convergence $\hat{\boldsymbol{\sigma}}^2$:**
- (a) Generate $\Sigma_{\mathbf{ll}}$ from $\hat{\boldsymbol{\sigma}}^2$ using (26)
- (b) **Iterate until convergence $\boldsymbol{\xi}$:**
  - (i) For each measurement $i = 1, \dots, N$:
    - Compute auxiliary quantities: $\mathbf{Y}_i$, $\mathbf{Z}_i$, $\mathbf{W}_i$, $\mathbf{c}_i$, $\mathbf{U}_i$, $\mathbf{e}_i$.
  - (ii) Compute uncertainty:
    - $\Sigma_{\boldsymbol{\xi}\boldsymbol{\xi}} \leftarrow \sum_{i=1}^N \mathbf{Y}_i^\top \mathbf{W}_i \mathbf{Y}_i$.
  - (iii) Update calibration:
    - Compute $\Delta\boldsymbol{\xi}$ as in Eq. (15).
  - (iv) For each measurement $i = 1, \dots, N$:
    - Compute $\Delta\mathbf{l}_i$ as in Eq. (16).
    - Update measurement: $\mathbf{l}_i \leftarrow \Delta\mathbf{l}_i \oplus \mathbf{l}_i$.
    - Update total correction: $\boldsymbol{\epsilon}_i \leftarrow \Delta\mathbf{l}_i \oplus \boldsymbol{\epsilon}_i$.
  - (v) Update $\boldsymbol{\xi} \leftarrow \Delta\boldsymbol{\xi} \oplus \boldsymbol{\xi}$.
- (c) For each measurement $i = 1, \dots, N$ calculate $\mathbf{P}_i$
- (d) Calculate $\mathbf{N}$ and $\mathbf{v}$ according to (29) and (30)
- (e) Estimate and update $\hat{\boldsymbol{\sigma}}^2$

3.4 Decalibration detection

Decalibration detection and outlier removal can be effectively addressed using the Mahalanobis distance. To fully leverage the probabilistic nature of calibration, we utilize the outputs of the calibration and variance estimation algorithm (Alg. 1). Our decalibration detection method computes the Mahalanobis distance of the constraint function for incoming ego-motion measurements, utilizing the estimated ego-motion and calibration covariances, and calibration parameters. This provides a probabilistic consistency measure for the data.

We examine the Mahalanobis distance separately for the translational and rotational components of the constraints, accounting for each sensor individually. This approach helps remove the effect of different measurement units and allows us to determine which sensor suffered decalibration. The theoretical mean of the constraint function for a well-calibrated system is $\boldsymbol{\mu} = \mathbf{0}$, with the covariance given by

$$\Sigma_{\mathbf{g}i} = \mathbf{Z}_i \Sigma_{\mathbf{ll}} \mathbf{Z}_i^\top + \mathbf{Y}_i \Sigma_{\boldsymbol{\xi}\boldsymbol{\xi}} \mathbf{Y}_i^\top. \tag{31}$$

While the covariance of each data point varies, the variation remains minimal for small incremental ego-motion, allowing us to treat it as constant until the next batch calibration procedure. Thus, we can approximate the covariance using the value from the last data point of the batch-calibrated data. This approximation enables efficient calculation of the Mahalanobis distance for incoming data

$$d_M(\mathbf{g}_i) = \sqrt{(\mathbf{g}_i - \boldsymbol{\mu})^T \Sigma_{\mathbf{g}}^{-1} (\mathbf{g}_i - \boldsymbol{\mu})}. \tag{32}$$

By setting the cutoff value for the Mahalanobis distance at a significance level of $\alpha = 0.05$, we can identify potential outliers that deviate significantly from the expected distribution. To distinguish between decalibrated points and outliers, we use a threshold based on the percentage of points with a Mahalanobis distance exceeding the heuristically set threshold of 10%, and window length set at $N = 20$. If the percentage of flagged points exceeds the threshold, a new batch-calibration procedure is initialized. These choices were guided by a trade-off observed in preliminary simulations. Shorter windows or lower α thresholds increase false alarm rates, while longer windows or higher thresholds delay detection. In practice, these parameters can be tuned when prior knowledge of noise levels, proportion of outliers or motion profiles is available.

4. EXPERIMENTS

To evaluate the performance of our calibration and decalibration detection methods, we designed a synthetic experiment by generating a simulated dataset for a three sensor system of ego-motion samples in three-dimensional space. The dataset was constructed to have an average translational magnitude of ego-motion of 0.05 m and an average rotational magnitude of 3°, ensuring that the directions of these vectors were uniformly distributed on the unit sphere. The magnitudes of the translation and rotational vectors were generated using a χ-distribution. The scale parameter of the χ-distribution was determined to match the desired average magnitudes of the vectors. The ground truth translational calibration parameters were drawn from a uniform distribution over the range $[-1, 1]$ m, whereas the rotational calibration parameters

were obtained by generating a random 3D rotation. Subsequently, the ego-motion of the second and third sensor was obtained by transforming the generated motion using the ground truth calibration parameters. Finally, ego-motion noise base values of the sensors are equal to $(1\%, 1\%)$, $(1\%, 5\%)$ and $(5\%, 1\%)$ in terms of percentages of average translational and rotational displacement. The Gaussian noise base values have been scaled with multiplicative factors ranging from 1 to 5 and added to the ego-motion vectors based on the stochastic model in Subsection 3.3 and the incrementation scheme in Subsection 3.1.

We conducted 100 experiments for each combination of noise levels and decalibration levels. Each experiment began with performing batch calibration on the first 100 ego-motions of a dataset. The decalibration detection method was tested by first introducing 100 new noisy but non-decalibrated data points. Following this, the decalibration detection method was tested by introducing decalibrated data until the recalibration flag was triggered or we reached the total of 200 newly introduced datapoints. We introduced decalibration in three different scenarios for the evaluation of the detection method. First, decalibration was introduced solely in the translation vector by perturbing the calibration parameters with a displacement of 0.03 m. Second, decalibration was introduced solely in the rotation vector by perturbing the calibration parameters with a change of 3°. Lastly, decalibration was introduced in both the translation and rotation vectors, with a translation perturbation of 0.03 m and a rotational perturbation of 3°. Decalibration was introduced by perturbing the ground truth calibration parameters. The translation and rotation vectors were independently perturbed to ensure controlled variability in both components. The translation vector was perturbed by adding a random displacement sampled uniformly and scaled to achieve a predefined norm. Similarly, the rotation vector was perturbed on the manifold as described in Eq. 6. Notably, in 50 percent of the experiments, decalibration was introduced, ensuring a balanced evaluation of the method's performance in both decalibrated and non-decalibrated scenarios.

To evaluate the performance of decalibration detection, we define two key metrics, detection accuracy and detection promptness. The detection accuracy metric measures the system's ability to correctly identify decalibration events while accounting for false positives and false negatives. A true positive occurs when the recalibration flag is correctly triggered after decalibration is introduced, while a true negative occurs if the decalibration was not flagged on an accurately calibrated dataset. A false positive occurs when the recalibration flag is triggered when no decalibration has occurred, while a false negative occurs when the system fails to trigger the flag despite the presence of decalibration. The accuracy metric is defined as the proportion of correctly detected events over the sum of all events and is expressed as

$$A = \frac{\text{Accurate flagging}}{\text{All events}}. \tag{33}$$

The detection promptness metric evaluates how quickly the system responds to decalibration. It measures the the datapoint index distance between when the decalibration is introduced and when the recalibration flag is triggered.

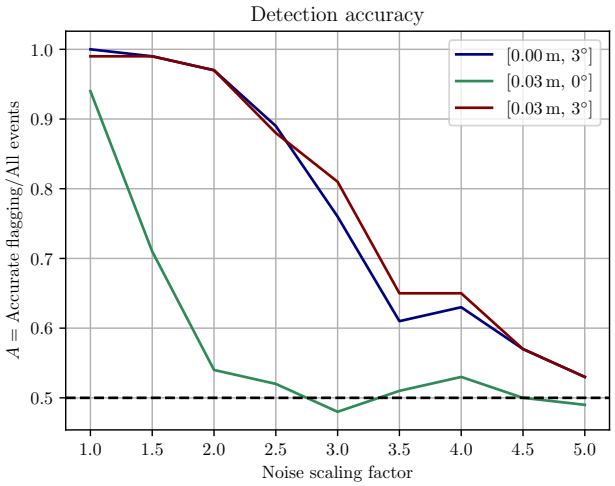


Fig. 2. Decalibration detection accuracy across three scenarios: translational decalibration, rotational decalibration, and combined translational and rotational decalibration. The dashed horizontal line at $y = 0.5$ represents the proportion of datasets without decalibration.

A smaller distance indicates a faster response, reflecting a more responsive system. The promptness is defined as

$$P = |i_{\text{triggered}} - i_{\text{introduced}}|. \tag{34}$$

where $i_{\text{triggered}}$ is the index at which the recalibration flag is triggered and $i_{\text{introduced}}$ is the index when decalibration was introduced. Each metric was averaged over 100 experiments with varying noise levels and decalibration severity.

In Figure 2, we present the accuracy of decalibration detection across three distinct perturbation scenarios: decalibration in rotation, decalibration in translation, and combined decalibration in both rotation and translation. The dashed horizontal line at $y = 0.5$ indicates the proportion of datasets that did not experience decalibration, serving as a baseline reference for the algorithm's performance. In particular, the detection algorithm has rarely experienced false positives in any of the scenarios. Furthermore, the results demonstrate that scenarios that involve decalibration in rotation, as well as those with combined decalibration in both rotation and translation, retain acceptable accuracy levels above 90% up to 2.5 times the set noise levels. This indicates the system's reliability under moderate to high noise conditions in these scenarios. In contrast, scenarios involving decalibration in translation alone exhibit lower detection accuracy, with reliable performance only on lowest noise levels, underscoring the greater challenge in detecting translational decalibration. This discrepancy arises because a pure translational offset only enters three of the six residual components per measurement pair, whereas a rotational offset perturbs all six. Making the traced Mahalanobis distances inherently smaller. Additional improvements to translational decalibration detection accuracy could be explored via observability analysis. In Figure 3, we observe the detection promptness for the three decalibration scenarios. For the two scenarios, rotational decalibration and combined decalibration, that can be reliably detected within the noise range where the detection accuracy exceeds 90%, the detection oc-

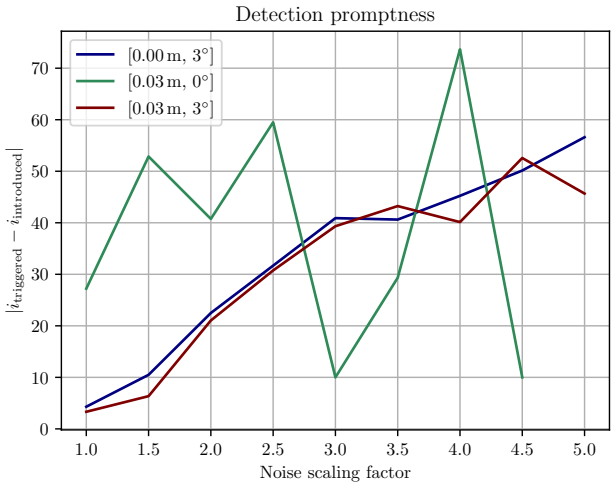


Fig. 3. Decalibration detection promptness for translational, rotational, and combined scenarios, measured as the index difference between the occurrence of decalibration and its detection.

curs within 30 indices of the decalibration event. At 20 frames per second, this corresponds to a detection time of 1.5 s, demonstrating responsiveness under high noise in odometry-only conditions common in fixed-base robot arms or slow-moving commercial mobile robots. Both the promptness and accuracy of decalibration detection, as shown in Figures 2 and 3, are strongly influenced by the quality of the initial covariance and parameter estimates, and information density of the data. One way to improve the performance of the algorithm would be to perform the initial calibration on a dataset longer than the 100 data points currently used. Finally, improving detection promptness could conceivably be achieved by employing a variable window length in the algorithm.

5. CONCLUSION

This work presents an odometry-based framework for decalibration detection in multisensor systems, leveraging uncertainty-aware optimization and Mahalanobis distance-based decalibration criteria. Through synthetic experiments, we evaluated the framework's performance across different decalibration scenarios and noise levels, demonstrating high detection accuracy and promptness in rotational and combined decalibration cases, even in moderate to high noise conditions. Translational decalibration proved challenging, highlighting the need for the development of new methods for detecting translational decalibration in odometry-based systems. We also identified that expanding the initial calibration dataset and adopting variable window lengths could further improve performance. These findings underscore the need for continued development of uncertainty-aware decalibration detection and tracking solutions to enhance robustness in real-world applications.

REFERENCES

Barfoot, T.D. (2017). *State Estimation for Robotics*. Cambridge: Cambridge University Press.

Dang, T., Hoffmann, C., and Stiller, C. (2009). Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing*, 18(7), 1536–1550.

Deray, J., Solà, J., and Andrade-Cetto, J. (2019). Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. In *2019 European Conference on Mobile Robots (ECMR)*, 1–6.

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

Ishikawa, R., Oishi, T., and Ikeuchi, K. (2018). Lidar and camera calibration using motions estimated by sensor fusion odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7342–7349.

Jiang, J., Xue, P., Chen, S., Liu, Z., Zhang, X., and Zheng, N. (2018). Line feature based extrinsic calibration of LiDAR and camera. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 1–6.

Levinson, J. and Thrun, S. (2013). Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*.

Lv, J., Xu, J., Hu, K., Liu, Y., and Zuo, X. (2020). Targetless calibration of LiDAR-IMU system based on continuous-time batch estimation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9968–9975.

Moravec, J. and Šára, R. (2024). Online camera–LiDAR calibration monitoring and rotational drift tracking. *IEEE Transactions on Robotics*, 40, 1527–1545.

Peršić, J., Petrović, L., Marković, I., and Petrović, I. (2021). Online multi-sensor calibration based on moving object tracking. *Advanced Robotics*, 35(3), 130–140.

Scaramuzza, D., Harati, A., and Siegwart, R. (2007). Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4164–4169.

Schneider, S., Luettel, T., and Wuensche, H.J. (2013). Odometry-based online extrinsic sensor calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1287–1292.

Teunissen, P.J.G. and Amiri-Simkooei, A.R. (2008). Least-squares variance component estimation. *Journal of Geodesy*, 82(2), 65–82.

Vogel, S., Ernst, D., Neumann, I., and Alkhatib, H. (2022). Recursive Gauss-Helmert model with equality constraints applied to the efficient system calibration of a 3D laser scanner. *Journal of Applied Geodesy*, 16(1), 37–57.

Wang, B., Zhao, Z., Chen, Y., and Yu, J. (2023). A novel robust point cloud fitting algorithm based on nonlinear Gauss–Helmert model. *IEEE Transactions on Instrumentation and Measurement*, 72, 1–12.

Wang, B., Zhao, Z., Wang, S., Yu, J., and Chen, Y. (2024). Robust LS-VCE for the nonlinear Gauss–Helmert model. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1–13.