Extended Kalman Filter for Localization

Carolina Realinho Pires
ist1102574
MEAer
Instituto Superior Técnico
Lisbon, Portugal
carolinarealinhopires@tecnico.ulisboa.pt

Joshua Thomas Redelbach
ist1112470
MEIC - Erasmus
Instituto Superior Técnico
Lisbon, Portugal
joshua.redelbach@tecnico.ulisboa.pt

William Clarke
ist1111070
MEAer
Instituto Superior Técnico
Lisbon, Portugal
william.clarke@tecnico.ulisboa.pt

***Abstract*—This report presents an Extended Kalman Filter (EKF)-based localization framework for the TurtleBot3 mobile robot, evaluated under various conditions including accurate and inaccurate initial poses, as well as an approximation of the kidnapping scenario. Three Iterative Closest Point (ICP) variants are compared, with Pure ICP offering the best trade-off between accuracy and runtime. A simplified Globally-Optimal ICP (GO-ICP) is also integrated to enhance robustness in failure recovery, at the cost of increased computation. Including image-based pose estimates is investigated as well, however leading to no improvement in the filter's performance. Evaluation across multiple ground-truth trajectories demonstrates the reliability and limitations of the EKF and GO-ICP approach for indoor localization.**

***Index Terms*—EKF, Localization, Pose Estimation.**

I. Introduction

Mobile robot localization is the task of estimating a robot's pose $\mathbf{x} = (x, y, \theta)^T$ with respect to a known map of the environment. Accurate localization is essential for autonomous navigation and typically requires integrating data from multiple sensors over time.

This work focuses on the pose estimation of a TurtleBot3 Waffle Pi [1] using the available data and sensors: odometry, LiDAR and camera. Both the pose tracking problem (assuming a known initial pose) as well as the global localization problem (unknown initial pose) are considered. Additionally, the system's robustness is evaluated under a simulated relocation scenario to assess its ability to recover from severe pose estimation failures.

As part of the commitment of the authors to reproducibility, the source code for all experiments described throughout the report is publicly available.[1]

II. Theory

A. Extended Kalman Filter

The *Extended Kalman Filter (EKF)* is a recursive Bayesian estimator that extends the Linear Kalman Filter [2] to non-linear systems. It maintains a belief over the robot's pose $\mathbf{x} = (x, y, \theta)^T$ using a Gaussian distribution parametrized by a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

[1]See: https://github.com/JoshRedelbach/AutSys_EKF_Localization

The EKF recursion consists of two main stages. First, the *prediction step* uses the motion model g to estimate the new state $\bar{\mu}_t$ and its associated uncertainty:

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \tag{1}$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t \tag{2}$$

Next, the *update step* incorporates the new measurement z_t using the measurement model h to get a final estimate μ_t:

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} \tag{3}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \tag{4}$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t \tag{5}$$

Since g and h are non-linear, the EKF approximates them using first-order Taylor expansions around the current mean, with Jacobians G_t and H_t. The matrix Q_t represents the covariance of the process noise, while R_t captures the covariance of the measurement noise. Both are assumed to be Gaussian.

Between the prediction and update steps, it is essential to perform an additional verification known as the *matching step* [4]. In this phase, the predicted observation is compared to the actual sensor reading by computing the Mahalanobis distance, defined in equation 6, which gives a measure of how surprised the estimator is by the measurement. This allows the estimator to reject measurements that deviate significantly from the prediction, thereby improving robustness.

$$d^2 = (\mathbf{z}_t - h(\bar{\mu}_t))^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} (\mathbf{z}_t - h(\bar{\mu}_t)) \tag{6}$$

The filter is initialized with an initial pose estimate and a corresponding covariance matrix. The performance of the filter depends critically on this initialization as well as on the accuracy of the system functions g and h.

B. Odometry Model

Odometry provides estimates of the robot's motion by integrating wheel encoder readings. Although convenient, odometry suffers from drift and is only reliable over short durations. The motion between time steps $t-1$ and t is modelled as a combination of two rotations and a translation: $(\delta_{rot1}, \delta_{trans}, \delta_{rot2})$ [3]. Given two successive odometry read-

ings $(x_{t-1}, y_{t-1}, \theta_{t-1})$ and (x_t, y_t, θ_t), the motion increments are computed as:

$$\delta_{rot1} = \text{atan2}(y_t - y_{t-1}, x_t - x_{t-1}) - \theta_{t-1} \tag{7}$$

$$\delta_{trans} = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{8}$$

$$\delta_{rot2} = \theta_t - \theta_{t-1} - \delta_{rot1} \tag{9}$$

The pose is then predicted in the EKF prediction step by the non-linear function g, using the last estimated orientation θ_{est}, defined as:

$$\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} \delta_{trans} \cdot \cos(\theta_{est} + \delta_{rot1}) \\ \delta_{trans} \cdot \sin(\theta_{est} + \delta_{rot1}) \\ \delta_{rot1} + \delta_{rot2} \end{pmatrix} \tag{10}$$

with the Jacobian:

$$G_t = \begin{pmatrix} 1 & 0 & -\delta_{trans} \cdot \sin(\theta_{est} + \delta_{rot1}) \\ 0 & 1 & \delta_{trans} \cdot \cos(\theta_{est} + \delta_{rot1}) \\ 0 & 0 & 1 \end{pmatrix}$$

C. Iterative Closest Point (ICP)

While odometry enables dead reckoning, it cannot prevent error accumulation over time or provide absolute pose corrections, which highlights the need for external sensor-based corrections in the EKF. The *Iterative Closest Point (ICP)* algorithm aligns a scan (source point cloud, $P = \{p_1, p_2, \ldots, p_m\}$) with a map (target point cloud, $Q = \{q_1, q_2, \ldots, q_n\}$) by minimizing the distance between corresponding points [5]. The goal is to find a rigid transformation $T = [R|t]$, defined by a rotation matrix and a translation vector, that minimizes the error:

$$E(R, t) = \sum_{i=1}^{m} \|Rp_i + t - q_i\|^2 \tag{11}$$

The algorithm proceeds iteratively:

1) Initialize the transformation T (from the pose prediction).
2) For each $p_i \in P$, find the closest point $q_i \in Q$ using the k-d tree algorithm.
3) Solve for the transformation T that minimizes equation 11.
4) Update $P \leftarrow T(P)$ and repeat until convergence.

To improve robustness, outlier rejection schemes like *RANdom SAmple Consensus (RANSAC)* can be integrated into ICP. RANSAC identifies inliers by randomly sampling minimal sets, solving for model parameters, and accepting those with support above a threshold [6].

Three versions of ICP were tested in this work, with a trade-off between accuracy and computational cost. *Pure ICP* directly aligns the scan to the map without outlier filtering. The *ICP with RANSAC First* runs RANSAC first to detect and discard outliers of the scan point set. Then, it runs the ICP algorithm only on inliers with refined initial guess. Last, the *ICP with Nested RANSAC* applies RANSAC in each ICP-loop step. In each ICP version, first the scan points are downsampled by only taking every second point to improve runtime.

The output of the algorithm is a rigid transformation from which a measurement vector $\mathbf{z}_t = (t_x, t_y, \theta)^T$ is extracted and used as the observation. Since this measurement directly estimates the robot's pose, the measurement model is:

$$h(\bar{\mu}_t) = \bar{\mu}_t, \quad H_{icp} = I_{3\times 3}. \tag{12}$$

D. Globally Optimal Iterative Closest Point (GO-ICP)

The standard ICP algorithm, although widely adopted, suffers from several limitations: it is highly sensitive to initial pose estimates, prone to convergence to local minima, and lacks robustness in the presence of significant misalignments [5]. These issues render it unsuitable for global localization scenarios, where no prior information about the robot's pose is available, or where the robot is kidnapped.

To improve robustness, this work investigated the integration of a simplified version of the *Globally-Optimal ICP (GO-ICP)* algorithm [7]. GO-ICP employs a branch-and-bound strategy to systematically search the space of rigid-body transformations, in combination with local ICP refinement, to ensure convergence to the global minimum of the registration error. While the original formulation is computationally intensive and generally not suited for real-time applications, a quasi-global adaptation was adopted here. The search was limited to a discrete set of rotation candidates in the range $[-180°, 180°]$ with a fixed angular step of $90°$, and translation hypotheses were restricted to a set of uniformly sampled reference points from the map.

Incorporation into the EKF framework was designed to address tracking failures and to enable absolute pose corrections over the whole map. This was implemented by two modifications. First, at the beginning of each ICP registration, the number of inliers produced by the initial transformation T_0, computed from the current state estimate μ_t, was evaluated. If fewer than 60% of the scan points were classified as inliers, the initial pose was deemed unreliable. In such cases, the standard ICP procedure was replaced with the more robust GO-ICP algorithm, allowing the system to recover from poor initial estimates. Second, if more than five consecutive observations were rejected during the EKF update step due to excessive Mahalanobis distance, the state covariance matrix Σ_t was inflated to reflect the increased uncertainty. This strategy addressed scenarios where the filter may lose track of the correct pose, such as during a kidnapping event.

E. Image-based Pose Estimation

Additional to the odometry and scan data, images from the robot's camera can be used to estimate the robot's relative pose between consecutive frames. This can be achieved using feature detection algorithms such as SURF, ORB, or tracking methods like KLT. These methods identify visual correspondences between frames, used to compute the *Essential matrix* which encodes the relative rotation and the direction vector (up to scale) between the two views. Robust estimation of the Essential matrix often incorporates RANSAC to reject outliers.

To recover an absolute pose measurement, the following quantities are fused:

- The prior state estimate $\mu_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$.
- A relative pose $\Delta\tilde{\mathbf{x}} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})^T$, obtained from the Essential matrix (up to scale).
- The current linear velocity v_t from odometry to resolve the unknown scale.

The scale factor is recovered as:

$$s = v_t \cdot \Delta t \quad \Rightarrow \quad \Delta\mathbf{x} = s \cdot \begin{bmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \end{bmatrix}, \quad \Delta\theta = \Delta\theta$$

leading to the absolute pose measurement for the EKF:

$$\mathbf{z}_{cam} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix} \tag{13}$$

The measurement model is the identity mapping, as the measurement directly observes the pose:

$$h(\bar{\mu}_t) = \bar{\mu}_t, \quad H_{cam} = I_{3\times 3}. \tag{14}$$

In this work, the ORB algorithm [9] was applied on the incoming image frame and 100 features were extracted. After applying a brute-force matching, the best 20 correspondences were selected. Lens distortion was removed of these corresponding image points using the calibration matrix and distortion coefficients, which were obtained in advance. The Essential matrix was computed integrating the RANSAC algorithm. The resulting relative pose between the two camera coordinate frames was then transformed into a relative robot motion by applying the similarity transformation from camera to robot frame, obtained by measuring the robot's dimensions.

III. Experimental Setup

In order to evaluate the performance of the EKF and the different algorithms, data was recorded using a TurtleBot3 Waffle Pi platform running ROS 1 Noetic. A 2D occupancy grid map consisting of 3452 points was generated in a laboratory environment using the *gmapping SLAM* package.

Three evaluation trajectories were designed. Trajectory 1 describes a square path to allow an easy visual validation, while trajectories 2 and 3 are longer, with multiple turns covering the entire room. Control points were physically marked on the floor. Trajectory 1, 2 and 3 include 4, 12 and 18 control points respectively. Their positions were measured relative to a room-fixed reference frame to enable comparison with the map frame.

The data of the trajectory was recorded using *rosbag* for offline analysis. Bag durations were 357 s, 450 s, and 572 s respectively. All three bags include the odometry and laser scan data. As the camera was corrupted during the first two bags, only the third bag includes published camera images. The odometry, scan data and images were published with different rates (approximately 27 Hz, 5 Hz and 15 Hz, respectively) and so the filter worked asynchronously, predicting continuously with odometry and updating as soon as laser or camera data was available. The process and measurement covariance matrices were tuned in advance with final values of $Q = \text{diag}(0.001, 0.001, 2)^2$, $R_{icp} = \text{diag}(0.1, 0.1, 2)^2$ and $R_{cam} = \text{diag}(0.1, 0.1, 2)^2$.

TABLE I: True and good estimated initial poses. Position and angle are given in [m] and [deg] respectively.

Trajectory	True	Good Estimate
1	$(-3.8,\ 4.3,\ 6.9)^T$	$(-4.0,\ 4.0,\ 0.0)^T$
2	$(-0.5,\ 0.0,\ 186.9)^T$	$(-0.5,\ 0.0,\ 180.0)^T$
3	$(1.0,\ 2.6,\ 96.9)^T$	$(1.0,\ 2.5,\ 90.0)^T$

TABLE II: Evaluation metrics for all trajectories (good initialization)

Method	FPE [m]	RMS-ATE [m]	Max-ATE [m]
Trajectory 1			
Pure ICP	0.030	0.043	0.117
Nested RANSAC	0.031	0.043	0.119
RANSAC First	0.033	0.042	0.115
Trajectory 2			
Pure ICP	0.079	0.057	0.193
Nested RANSAC	0.081	0.058	0.197
RANSAC First	0.079	0.057	0.195
Trajectory 3			
Pure ICP	0.106	0.049	0.106
Nested RANSAC	0.057	0.047	0.103
RANSAC First	0.059	0.046	0.102

The entire work was implemented in *Python* and tested by replaying the *rosbags*. All algorithms were self-implemented except for image processing tasks such as feature detection, matching, and computation of the Essential matrix. For this, *OpenCV* was used. All simulations were run on an *Apple MacBook Pro 14" 2021* with an *Apple M1 Pro* chip and 16 GB unified memory.

IV. Results

A. Pose Tracking Problem

This experiment evaluates the performance of the three ICP variants described in section II-C in the context of pose tracking using the EKF. All trajectories were initialized with accurate pose estimates. No camera data was used in the filter.

The following three metrics are used to assess tracking performance:

- *Final Position Error (FPE)*: Euclidean distance between estimated and ground truth final position.
- *Maximum Absolute Trajectory Error (Max-ATE)*: Largest deviation between estimated and true position evaluated based on control points.
- *Root Mean Squared ATE (RMS-ATE)*: RMS value of the absolute trajectory error over all control points.

The true and estimated initial pose for each trajectory are given in table I. The initial covariance matrix for all runs was fixed as $\Sigma_0 = I_{3\times 3}$. The quantitative results for each trajectory are summarized in table II. Although some variations exist across trajectories, all three ICP variants provided consistent results in terms of accuracy. The average runtime per ICP iteration was recorded as well, shown in table III.

TABLE III: Average Runtime per ICP Method [ms]

Method	Average Runtime [ms]
Pure ICP	12.6
Nested RANSAC	17.3
RANSAC First	14.6

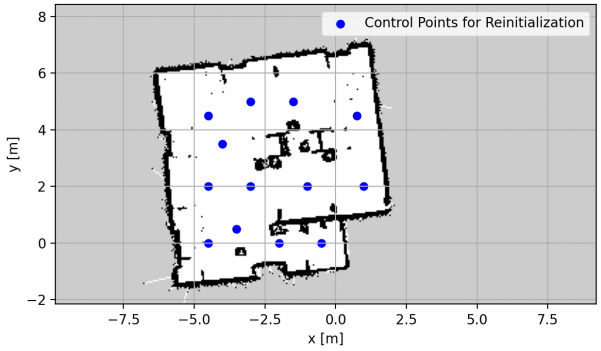


Fig. 1: 13 candidate points considered in GO-ICP.

TABLE IV: Evaluation metrics for all trajectories using GO-ICP (bad initialization)

Trajectory	FPE [m]	RMS-ATE [m]	Max-ATE [m]
1	0.074	0.063	0.153
2	0.056	0.049	0.159
3	0.106	0.049	0.106

Pure ICP consistently achieved the lowest computation time, with accuracy comparable to the more complex variants. As such, *Pure ICP* was selected for subsequent experiments due to its favourable trade-off between efficiency and performance.

B. Approximate Global Localization Problem

In order to test if the EKF is able to recover from a bad initial pose estimate by only using *Pure ICP* in the update step, all three trajectories were tested providing a bad initial estimate $\mu_0 = (-4\,\mathrm{m}, 4\,\mathrm{m}, 180\,\mathrm{deg})^T$ and setting the initial covariance matrix to $\Sigma_0 = 10 \cdot I_{3\times3}$. As can be seen in figure 2, for no trajectory the filter is able to recover. For that reason, the EKF is adapted by including a GO-ICP reinitialization routine within the standard ICP as described in section II-D. The 13 control points used for the reinitialization are depicted in figure 1. As can be seen in figure 2, the filter was now able to correct the bad initial estimate and tracks the position well over all three trajectories. This observation can be confirmed by evaluating the metrics given in table IV. However, a single GO-ICP execution took on average $679.8\,\mathrm{ms}$, which is significantly longer than the standard ICP routine.

C. Pose Recovery under Sensor Reset Scenario

To assess the system's robustness to severe localization failures, a sensor reset scenario was tested. This aimed to approximate the kidnapped robot problem without physical displacement. Two independently recorded rosbags were used: the rosbag from trajectory 1 was played until $t = 100\,\mathrm{s}$, and then the rosbag of path 2 started playing until its $t = 222\,\mathrm{s}$. By this approach the odometry data will be reset and will propagate a jump of the state. The pose estimate by the ICP will try to correct the current state even though the state's covariance is low. If the EKF can recover successfully, it is likely that it can also handle real kidnapping scenarios as well.

The EKF filter was initialized with the previous pose estimate $\mu_0 = (-4\,\mathrm{m}, 4\,\mathrm{m}, 180\,\mathrm{deg})^T$, with an inflated covariance $\Sigma_0 = 10 \cdot I_3$ to reflect the high uncertainty. Between the rosbags, the covariance matrix was not reset.

The resulting true and estimated trajectory can be observed in figure 3. Despite the large mismatch between the belief and true pose at the start of rosbag 2, the system was successfully recovered using the GO-ICP alignment module. The FPE was $0.025\,\mathrm{m}$, RMS-ATE was $0.054\,\mathrm{m}$, and the Max-ATE was $0.193\,\mathrm{m}$. These results confirm the system's ability to recover from a kidnapping-like scenario without manual reinitialization.

At the end of the run, the filter's confidence was reflected in the final covariance matrix:

$$\Sigma = \begin{bmatrix} 1.98 \times 10^{-4} & -9.93 \times 10^{-8} & -7.00 \times 10^{-6} \\ -9.93 \times 10^{-8} & 1.98 \times 10^{-4} & -5.46 \times 10^{-7} \\ -7.00 \times 10^{-6} & -5.46 \times 10^{-7} & 2.23 \times 10^{-3} \end{bmatrix}$$

This low-magnitude covariance indicates that the filter regained high confidence in its pose estimate. This behaviour was consistent across other trajectories, except during GO-ICP invocations, where the covariance was deliberately increased to enable outlier correction acceptance based on the Mahalanobis threshold. Overall, this scenario demonstrates the system's resilience, although the limitations of using sequential rosbags instead of continuous live data are acknowledged in the discussion section V.

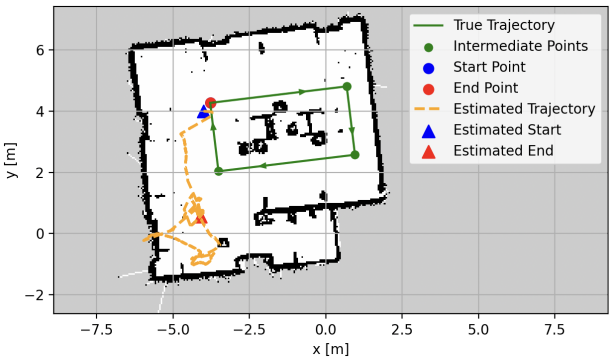
D. Camera Integration

To investigate if the estimation performance can be further improved, the image-based pose estimation update was integrated in the EKF as described in section II-E. The third trajectory was tested with the same poor initial estimate as in section IV-B. As the average runtime of the image-based pose estimate was about $10.0\,\mathrm{ms}$, only every tenth image was used for an update step to be able to run the EKF in real time. Throughout trajectory, the update step was called 833 times and only 5 image-based pose estimates were rejected due to excessive Mahalanobis distance, indicating the reliability of the algorithm. However, the results show that the performance of the estimator could not be improved with an FPE of $0.104\,\mathrm{m}$, an RMS-ATE of $0.050\,\mathrm{m}$ and a Max-ATE of $0.111\,\mathrm{m}$.

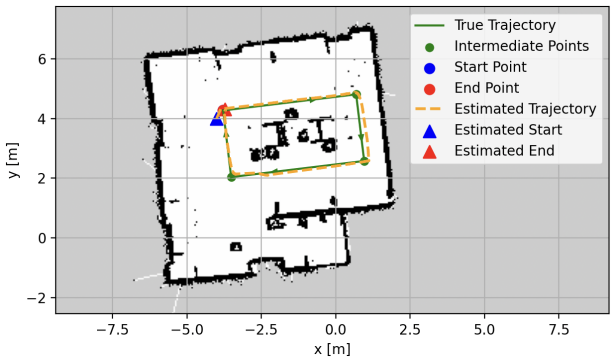
V. Discussion

This work aimed to evaluate an EKF-based localization framework integrating various ICP strategies, including a quasi-global GO-ICP routine, in scenarios with good initial pose estimates and under challenging recovery conditions.

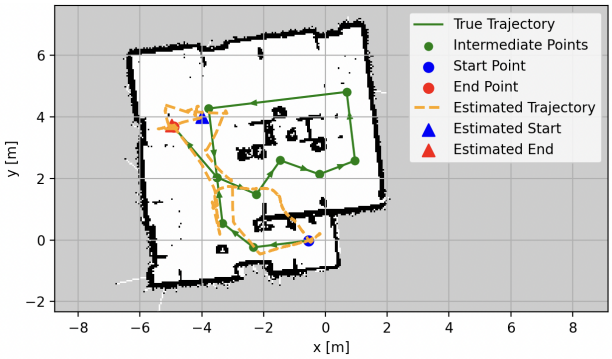
The quantitative evaluation focused on FPE, RMS-ATE, and Max-ATE across three trajectories. While these metrics offered



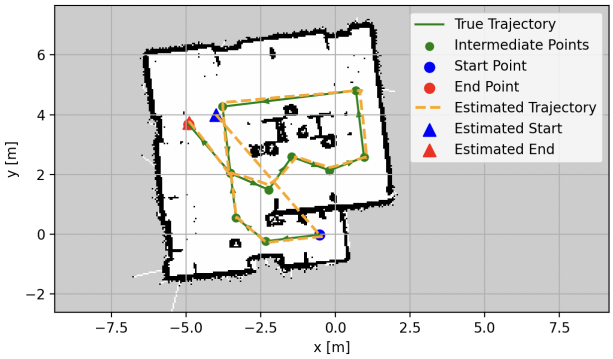
(a) Pure ICP Only - Trajectory 1



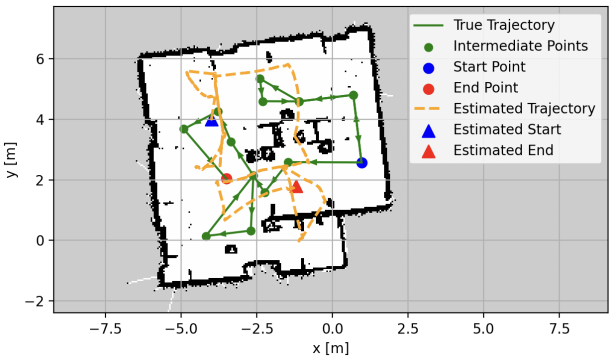
(b) GO-ICP - Trajectory 1



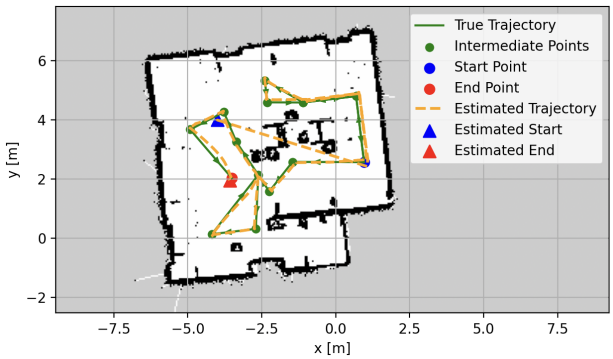
(c) Pure ICP Only - Trajectory 2



(d) GO-ICP - Trajectory 2



(e) Pure ICP Only - Trajectory 3



(f) GO-ICP - Trajectory 3

Fig. 2: Plots of the true and estimated trajectory using only ICP algorithm or the adapted GO-ICP algorithm.

insight into system accuracy, they are not without limitations. For instance, FPE only reflects the final estimation error, not how well the entire trajectory aligns. Similarly, taking the RMS-ATE of the minimal distances to manually marked control points may indicate spatial proximity, but not necessarily temporal alignment or trajectory correctness. Therefore, such metrics should be interpreted cautiously, especially for real-time, continuous localization.

Despite these caveats, the results demonstrate that all ICP variants tested - *Pure ICP*, *Nested RANSAC*, and *RANSAC First* - performed comparably under good initialization. For example, in trajectory 1, all methods achieved an FPE below $3.3\,\mathrm{cm}$, with RMS-ATE under $4.3\,\mathrm{cm}$. It should be noted that the Turtlebot3 has a length of $13.8\,\mathrm{cm}$ and a width of $17.8\,\mathrm{cm}$, so these errors are within the robot's limits. The differences across the three methods were marginal (less than $2\,\mathrm{mm}$ in RMS-ATE), highlighting that for tracking scenarios with good priors, the simpler *Pure ICP* algorithm is sufficient. However, using bad initial estimates, the filter could not recover when only using those standard ICP approaches.

This issue was successfully solved by introducing the GO-ICP reinitialization routine leading to RMS-ATE of less than $6.3\,\mathrm{cm}$ for all trajectories. Furthermore, GO-ICP significantly improved robustness in cases of severe pose uncertainty, recovering from a pose mismatch in the sensor reset scenario with a final FPE of $2.5\,\mathrm{cm}$ and RMS-ATE of $5.4\,\mathrm{cm}$. However, the ap-

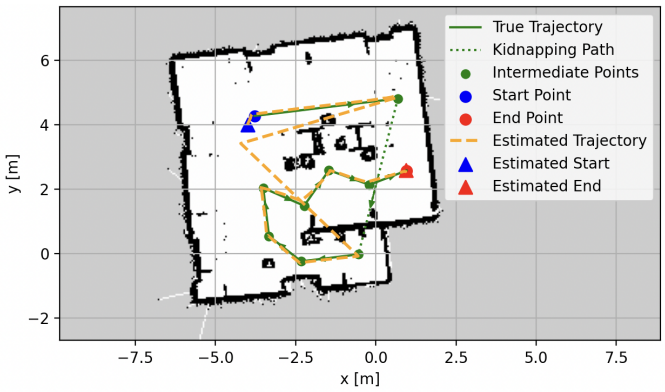


Fig. 3: Plot of true and estimated trajectory for the sensor reset scenario.

proach is not scalable. Its runtime increases sharply with map complexity and search resolution. In larger environments or with denser scan data, the reduced candidate set of translations and rotations may fail to capture the true alignment, preventing recovery. Furthermore, continuous use of GO-ICP degrades real-time performance. From the results, it took approximately 50 times more to run a GO-ICP iteration compared to a pure ICP iteration. For longer trajectories, even the usage of ICP can lead to instability, which can ultimately lead to losing the robot's position, especially on lower-performance hardware.

The simulated "kidnapping" scenario was implemented by playing two independent rosbags sequentially without resetting the filter, effectively introducing a pose discontinuity. While this did not replicate a true blindfolded displacement (with odometry and LiDAR both disconnected, for example), it reflected a realistic operational failure, such as a sensor blackout or system reboot. In such cases, the robot would resume with a mismatched belief. Thus, while not perfect, the scenario enabled controlled testing of the filter's recovery mechanisms.

Nonetheless, odometry inconsistencies at the bag transition can introduce leakage. Although both bags were independently recorded and not artificially truncated, the initial odometry message in the second bag might unintentionally guide the filter, introducing a bias. Future experiments could enforce continuity by overwriting or equalizing the odometry message across transitions to remove this bias.

In addition to the scan-based update step, an image-based pose estimation method was implemented and evaluated. Although the integration of these pseudo-absolute measurements did not lead to a significant improvement in overall performance, the method demonstrated potential. The filter consistently incorporated the image-based measurements, even though the achievable accuracy was limited due to the presence of calibration inaccuracies or poor measurements when determining the transformation between camera and robot coordinate frame.

Besides algorithmic limitations, other sources of error include odometry drift due to floor conditions and human teleoperation inconsistencies. Ground truth was manually measured and aligned using a known transformation, introducing further uncertainty. Nonetheless, the errors remained within reasonable bounds relative to the robot's physical dimensions. For example, the worst-case FPE was $10.6\,\mathrm{cm}$, and all RMS-ATE values remained below $6\,\mathrm{cm}$, suggesting the accuracy achieved is acceptable for indoor navigation.

Future work could explore more scalable alternatives to GO-ICP, such as the Nested Annealing ICP (NAICP) algorithm [8], or improve the integration of camera information. More realistic recovery scenarios, such as the true physical kidnapping with blindfolded sensors, should also be investigated. Additionally, extending to larger maps would require more intelligent candidate points selection and exploring different downsampling strategies to maintain online runtime feasibility.

VI. Conclusion

A robust and modular EKF-based localization framework for mobile robots was presented, incorporating odometry, LiDAR, and camera data. The system was evaluated using a TurtleBot3 platform under both favourable and challenging conditions. Experimental results showed that simple ICP strategies are sufficient in scenarios with accurate initial pose estimates, while the GO-ICP method enhances robustness in the presence of severe pose uncertainty, albeit with increased computational demands. Localization accuracy remained within acceptable limits for indoor environments, despite hardware constraints and sensor limitations. The results illustrate key trade-off between accuracy, robustness, and real-time performance in multi-sensor fusion systems. Future work should address scalability, improve visual data integration, and explore more realistic failure scenarios to further improve resilience in autonomous mobile robot localization.

References

[1] Open Source Robotics Foundation and ROBOTIS, *TurtleBot3*, TurtleBot3 – official website, https://www.turtlebot.com/turtlebot3/, accessed June 9, 2025.

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[4] F. Kong, Y. Chen, J. Xie, Z. Gang, and Z. Zhou, "Mobile robot localization based on extended Kalman filter," in *Proc. 6th World Congr. Intell. Control Autom. (WCICA)*, 2006, vol. 2, pp. 9242–9246, doi: 10.1109/WCICA.2006.1713789.

[5] H. Wang, Y. Yin, and Q. Jing, "Comparative analysis of 3D LiDAR scan-matching methods for state estimation of autonomous surface vessel," *J. Mar. Sci. Eng.*, vol. 11, no. 4, p. 840, Apr. 2023, doi: 10.3390/jmse11040840.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.

[7] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016, doi: 10.1109/TPAMI.2015.2513405.

[8] T. N. Linh and H. Hiroshi, "Global iterative closest point using nested annealing for initialization," *Procedia Comput. Sci.*, vol. 60, pp. 381–390, 2015, doi: 10.1016/j.procs.2015.08.147.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.