Pose-Based EKF SLAM (PEKFSLAM) Using ICP Laser Scan Matching

Pravin Oli (u1999699), Gebrecherkos G. (u1999542)

Abstract—This paper presents a pose-based Extended Kalman Filter (EKF) SLAM system using laser scan matching with the Iterative Closest Point (ICP) algorithm. The method is designed for mobile robots operating in indoor environments and is validated in both simulation (RViz and Stonefish) and real-world TurtleBot platforms. The system fuses odometry and IMU data for prediction and corrects the robot pose using LIDAR-based ICP registration. Experimental results demonstrate accurate localization and consistent mapping under realistic conditions.

Index Terms—SLAM, EKF, ICP, LIDAR, Mobile Robots, ROS, Pose Graph, Loop Closure

1 Introduction

SIMULTANEOUS Localization and Mapping (SLAM) is a fundamental capability for autonomous mobile robots operating in unknown environments. The core challenge lies in building a consistent map of the environment while concurrently localizing the robot within it. Among the various approaches to SLAM, pose-based methods using the Extended Kalman Filter (EKF) are widely adopted for their probabilistic formulation and incremental update structure.

This paper presents a Pose-based EKF SLAM (PEKF-SLAM) system that integrates wheel odometry, IMU heading measurements, and 2D LIDAR scans to estimate the robot's trajectory and build a map. The system is designed for real-time operation in structured indoor environments, where geometric features such as walls and corners enable accurate scan registration.

The proposed method employs a differential-drive motion model for prediction using encoder inputs. To correct orientation drift, low-rate heading measurements from a compass or IMU are incorporated via a dedicated yaw correction EKF update. Each new LIDAR scan is stored along with the corresponding pose, forming an evolving pose-graph structure.

For data association, the system performs scan-to-scan registration using the Iterative Closest Point (ICP) algorithm. Only scans with sufficient geometric overlap are considered for matching. The resulting relative transformations are validated using the Individual Compatibility Nearest Neighbor (ICNN) test based on Mahalanobis distance and a chi-squared threshold. Compatible matches are used as observations to correct the robot's pose via a full EKF update.

Quantitative and qualitative results demonstrate improved pose accuracy, reduced uncertainty, and map consistency over dead reckoning alone. The modularity of the pipeline allows easy integration with additional sensors or future enhancements such as loop closure and graph optimization.

- *Master students at the University of Girona, Spain (IFRoS) Email: pravin.oli.08@gmail.com, chereg2016@gmail.com*

The remainder of this paper details the SLAM system architecture, mathematical formulation, scan registration pipeline, and experimental validation.

2 Related Works

Simultaneous Localization and Mapping (SLAM) has been an active research area for over two decades. Early SLAM methods used feature-based representations with Extended Kalman Filters (EKF) [1], [2], enabling real-time pose and landmark estimation. However, these approaches struggled with scalability and data association in large or unstructured environments.

Pose-based SLAM, where robot poses are the only estimated states and observations are relative transformations (e.g., from scan matching), has gained popularity due to its compact state representation and better convergence behavior. Works such as [3] and [4] provide a comprehensive overview of these methods.

Iterative Closest Point (ICP) is a widely used algorithm for scan registration in 2D and 3D SLAM. Research such as [5] and [6] compares various ICP variants, including point-to-plane and generalized ICP, and highlights their strengths in different environments. More recently, fast ICP implementations using GPU acceleration [7] and outlier rejection techniques such as TEASER++ [8] have improved robustness and speed.

For probabilistic scan registration, fusing ICP with EKF has shown promising results in mobile robotics. For instance, [9] and [10] explored graph-based SLAM and occupancy mapping using scan alignment. Mahalanobis distance-based outlier rejection techniques, such as the Individual Compatibility Nearest Neighbor (ICNN) filter [11], are commonly used in EKF pipelines for robust data association.

In recent years, research has focused on hybrid SLAM systems that fuse LIDAR, IMU, and visual information for improved accuracy and robustness. Systems like Cartographer [12] and LIO-SAM [13] integrate scan registration with inertial priors using factor graphs or tightly coupled filters.

Unlike many graph-based systems, our approach maintains an EKF framework for real-time performance and simplicity while integrating geometric scan matching for loop-consistent pose refinement.

3 System Overview

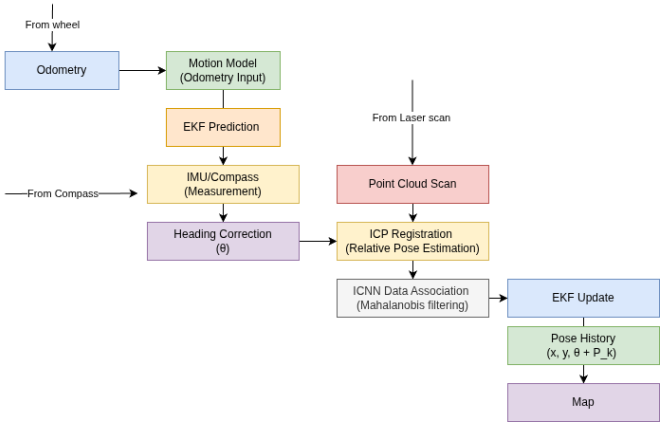

Fig. 1: System architecture showing data flow from odometry, IMU, and LIDAR into the EKF-SLAM pipeline.

Figure 1 illustrates the overall architecture of the implemented Pose-based EKF SLAM system. The pipeline begins with odometry data obtained from wheel encoders, which are processed through a differential-drive motion model to estimate the robot's pose prediction. Simultaneously, heading measurements from the IMU or compass sensor are used to correct the predicted yaw using a simple EKF update, reducing drift over time.

Upon receiving a new LIDAR scan, the pose and scan are appended to the state vector and map, respectively. The current scan is then matched with previous scans stored in the map using ICP (Iterative Closest Point) registration, estimating the relative transformation between viewpoints.

To ensure only reliable measurements are used, the registered transformation is validated against the EKF prediction using Mahalanobis distance within the ICNN (Individual Compatibility Nearest Neighbor) test. If the match passes the chi-squared threshold, the observation is accepted and used to update the state via the EKF correction step.

Finally, the updated pose is stored, and the aligned scans are integrated into the global map. This architecture enables continuous pose estimation and mapping in real time by tightly integrating sensor fusion, scan matching, and probabilistic filtering.

4 EKF SLAM Formulation

4.1 Spatial Relationship

In this SLAM formulation, the robot's trajectory is represented as a sequence of 2D poses defined in the global reference frame $\mathcal{N}$. Each robot pose at time step k, denoted ${}^{\mathcal{N}}\mathbf{x}_{B_k}$, is modeled in the Special Euclidean group $SE(2)$ as:

$$ {}^{\mathcal{N}}\mathbf{x}_{B_k} = \begin{bmatrix} {}^{\mathcal{N}}x_k \\ {}^{\mathcal{N}}y_k \\ {}^{\mathcal{N}}\theta_k \end{bmatrix} \in SE(2), \quad \text{where } \theta_k \text{ is the heading (yaw)} $$

The estimate of this pose, along with its local uncertainty, is expressed as:

$$ {}^{\mathcal{N}}\mathbf{x}_{B_k} \sim \mathcal{N}\left({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k},\, P^{\mathcal{N}}_{B_k} \right), \quad P^{\mathcal{N}}_{B_k} \in \mathbb{R}^{3\times 3} $$

Here, $P^{\mathcal{N}}_{B_k}$ denotes the covariance of the individual pose ${}^{\mathcal{N}}\mathbf{x}_{B_k}$.

The full SLAM state vector up to time step k accumulates all previous poses:

$$ \hat{\mathbf{x}}^{\mathcal{N}}_k = \begin{bmatrix} {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_0} \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_1} \\ \vdots \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k} \end{bmatrix} $$

The full state uncertainty is encoded as a block covariance matrix capturing both individual and correlated pose uncertainties:

$$ P^{\mathcal{N}}_k = \begin{bmatrix} P_{B_0} & P_{B_0B_1} & \cdots & P_{B_0B_k} \\ P_{B_1B_0} & P_{B_1} & \cdots & P_{B_1B_k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{B_kB_0} & P_{B_kB_1} & \cdots & P_{B_k} \end{bmatrix} $$

4.2 Motion Model

The motion of the TurtleBot is modeled using differential-drive kinematics, where its pose in the global frame $\mathcal{N}$ is updated using wheel encoder measurements. The robot's pose at time step k is denoted by:

$$ {}^{\mathcal{N}}\mathbf{x}_{B_k} = \begin{bmatrix} {}^{\mathcal{N}}x_k \\ {}^{\mathcal{N}}y_k \\ {}^{\mathcal{N}}\theta_k \end{bmatrix} \in SE(2) $$

The control input and additive Gaussian process noise are:

$$ u_k = \begin{bmatrix} \omega^k_L \\ \omega^k_R \end{bmatrix}, \quad w_k = \begin{bmatrix} \delta^k_L \\ \delta^k_R \end{bmatrix} \sim \mathcal{N}(0, Q_k) $$

where ω^k_L, ω^k_R are the angular velocities of the left and right wheels, and δ^k_L, δ^k_R are corresponding zero-mean Gaussian noise terms. The noise covariance is:

$$ Q_k = \begin{bmatrix} \sigma^2_L & 0 \\ 0 & \sigma^2_R \end{bmatrix} $$

Let Δt denote the time interval, r_w the wheel radius, and d the wheelbase. The motion update is expressed as:

$$ {}^{\mathcal{N}}\mathbf{x}_{B_k} = f({}^{\mathcal{N}}\mathbf{x}_{B_{k-1}}, u_k, w_k) $$

Specifically, the pose evolves via:

$$ {}^{\mathcal{N}}\mathbf{x}_{B_k} = \begin{bmatrix} {}^{\mathcal{N}}x_{k-1} \\ {}^{\mathcal{N}}y_{k-1} \\ {}^{\mathcal{N}}\theta_{k-1} \end{bmatrix} \oplus \begin{bmatrix} \frac{r_w\Delta t}{2}\left((\omega^k_L+\delta^k_L)+(\omega^k_R+\delta^k_R)\right)\cos({}^{\mathcal{N}}\theta_{k-1}) \\ \frac{r_w\Delta t}{2}\left((\omega^k_L+\delta^k_L)+(\omega^k_R+\delta^k_R)\right)\sin({}^{\mathcal{N}}\theta_{k-1}) \\ \frac{r_w\Delta t}{d}\left((\omega^k_R+\delta^k_R)-(\omega^k_L+\delta^k_L)\right) \end{bmatrix} $$

To propagate uncertainty, we compute the following Jacobians.

$$ F_{1_k} = \frac{\partial f_p({}^{\mathcal{N}}\mathbf{x}_{k-1}, u_k, w_k)}{\partial {}^{\mathcal{N}}\mathbf{x}_{k-1}} = \begin{bmatrix} I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \\ 0 & \cdots & 0 & J_{f_x} \end{bmatrix} \quad (12) $$

where

$$J_{f_x} = \begin{bmatrix} 1 & 0 & -V\Delta t \sin({}^{\mathcal{N}}\theta_{k-1}) \\ 0 & 1 & V\Delta t \cos({}^{\mathcal{N}}\theta_{k-1}) \\ 0 & 0 & 1 \end{bmatrix}, \quad V = \frac{r_w}{2}(\omega_L^k + \omega_R^k) \tag{13}$$

The Jacobian with respect to noise is:

$$F_{2_k} = \frac{\partial f_p({}^{\mathcal{N}}\mathbf{x}_{k-1}, u_k, w_k)}{\partial w_k} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ J_{f_w} \end{bmatrix} \tag{14}$$

with

$$J_{f_w} = \begin{bmatrix} \frac{r_w\Delta t}{2}\cos({}^{\mathcal{N}}\theta_{k-1}) & \frac{r_w\Delta t}{2}\cos({}^{\mathcal{N}}\theta_{k-1}) \\ \frac{r_w\Delta t}{2}\sin({}^{\mathcal{N}}\theta_{k-1}) & \frac{r_w\Delta t}{2}\sin({}^{\mathcal{N}}\theta_{k-1}) \\ -\frac{r_w\Delta t}{d} & \frac{r_w\Delta t}{d} \end{bmatrix} \tag{15}$$

Finally, the EKF prediction step updates the full state covariance as:

$$P_k = F_{1_k} P_{k-1} F_{1_k}^\top + F_{2_k} Q_k F_{2_k}^\top$$

This formulation accurately captures the nonlinearity and uncertainty of differential-drive motion, enabling robust prediction in PEKF-SLAM.

4.3 Scan Acquisition and Pose Addition

As the robot explores its environment, it continuously receives LIDAR scans. Each scan is acquired at a specific pose and must be stored in the map while updating the SLAM state. To enable this, the system grows both the state vector and covariance matrix by cloning the robot's current pose and inserting it as a new viewpoint.

Let the estimated pose at time step k be ${}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}$ with corresponding covariance $P_{B_k}^{\mathcal{N}}$. To grow the SLAM state, we clone the pose:

$${}^{\mathcal{N}}\hat{\mathbf{x}}_k^+ = \begin{bmatrix} {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_0} \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_1} \\ \vdots \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k} \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k} \end{bmatrix}$$

The state covariance matrix is expanded accordingly:

Method `AddNewPose`$({}^{\mathcal{N}}\hat{\mathbf{x}}_k, P_k)$

$$\left.\begin{aligned} {}^{\mathcal{N}}\hat{\mathbf{x}}_k^+ &= \begin{bmatrix} {}^{\mathcal{N}}\hat{\mathbf{x}}_k \\ {}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k} \end{bmatrix} \\ P_k^+ &= \left[\begin{array}{c|c} P_k & P_{0\ldots k,B_k} \\ \hline P_{B_k,0\ldots k} & P_{B_k} \end{array}\right] \end{aligned}\right\} \quad \texttt{return } ({}^{\mathcal{N}}\hat{\mathbf{x}}_k^+, P_k^+)$$

In parallel, the scan S_k is acquired and appended to the global map:

$$\mathcal{M} = [B_0S_0,\ B_1S_1,\ \ldots,\ B_kS_k]$$

Each scan S_k contains n_p points measured in the local robot frame B_k:

$$S_k = \left[{}^{B_k}p_0,\ {}^{B_k}p_1,\ \ldots,\ {}^{B_k}p_{n_p}\right], \quad {}^{B_k}p_i \in \mathbb{R}^2$$

These points are transformed into the global frame using the robot's pose:

$${}^{\mathcal{N}}p_i = T_{\mathcal{N}}^{B_k} \cdot {}^{B_k}p_i$$

This mechanism ensures each scan is indexed by a unique pose, allowing revisiting and alignment. The covariance structure is preserved for future correction. Scan-to-scan registration (ICP) can be efficiently applied using stored pose transformations.

Relative motion between any two robot poses can be recovered by:

$$T_{B_j}^{B_i} = \left(T_{\mathcal{N}}^{B_j}\right)^{-1} T_{\mathcal{N}}^{B_i}$$

The combined augmentation of pose and scan is critical for enabling map growth, data association, and loop closure in PEKF-SLAM.

4.4 Compass-Based Heading Correction

To correct for orientation drift from the motion model, we integrate heading measurements from a compass or IMU sensor. These measurements are fused using a simplified EKF update on the yaw angle alone.

The measurement model is:

$$z_\theta^k = \theta_k + v_\theta^k, \quad v_\theta^k \sim \mathcal{N}(0, R_\theta^k)$$

where:

- z_θ^k is the measured heading from the compass,
- $h_\theta(x_k) = \theta_k$ extracts the yaw from the pose,
- v_θ^k is the measurement noise,
- $R_\theta^k = \sigma_\theta^2$ is the heading noise variance.

The observation function and its Jacobian are:

$$h_\theta(x_k) = \theta_k, \quad H_\theta^k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

Given the predicted state $\hat{x}_k^-$ and covariance P_k^-, the EKF heading update becomes:

$$\begin{aligned} K_\theta^k &= P_k^- H_\theta^{k\top} \left(H_\theta^k P_k^- H_\theta^{k\top} + R_\theta^k\right)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_\theta^k \left(z_\theta^k - \hat{\theta}_k^-\right) \\ P_k^+ &= (I - K_\theta^k H_\theta^k) P_k^- \end{aligned}$$

This update adjusts only the yaw component of the state vector while preserving the position estimates (x, y). Full 2D pose corrections are applied separately using ICP-based registration during observation updates.

4.5 ICP Scan Matching

Once a new scan B_kS_k is added to the SLAM state, the system attempts to align it with previously acquired scans $\{B_jS_j\}$ that are spatially nearby. This alignment is performed using the Iterative Closest Point (ICP) algorithm, which estimates the relative rigid-body transformation between two scans.

Overlapping Scan Detection

To reduce unnecessary computations, we identify which past scans potentially overlap with the current view:

$$\mathcal{H}_p = \texttt{OverlappingScans}(\hat{\mathbf{x}}_k^{\mathcal{N}}, \mathcal{M})$$

Each detected overlapping pair $(B_jS_j, B_kS_k) \in \mathcal{H}_p$ is registered via scan matching.

ICP Registration

Let $B_kS_k = \{{}^{B_k}r_1, \ldots, {}^{B_k}r_N\}$ and $B_jS_j = \{{}^{B_j}b_1, \ldots, {}^{B_j}b_M\}$. The ICP objective is to find ${}^{B_j}\mathbf{x}_{B_k} \in SE(2)$ minimizing the sum of squared distances between each transformed point ${}^{B_k}r_i$ and its nearest neighbor in B_jS_j:

$${}^{B_j}\mathbf{x}^{\star}_{B_k} = {}_{{}^{B_j}\mathbf{x}_{B_k}} \sum_{i=1}^{N} \left\| {}^{B_j}c_i - {}^{B_j}\mathbf{x}_{B_k} \oplus {}^{B_k}r_i \right\|^2$$

Algorithm 1 ICP: Iterative Closest Point

function ICP(B_kS_k, B_jS_j, ${}^{B_j}\mathbf{x}_{B_k}^{(0)}$)
  $i \leftarrow 1$
  while $i <$ `maxIterations` and not converged **do**
    for $j = 1$ to N **do**
      ${}^{B_j}c_j \leftarrow_{{}^{B_j}b \in B_jS_j} \|{}^{B_j}b - {}^{B_j}\mathbf{x}_{B_k}^{(i-1)} \oplus {}^{B_k}r_j\|$
    end for
    Compute error: $e_j = {}^{B_j}\mathbf{x}_{B_k}^{(i-1)} \oplus {}^{B_k}r_j - {}^{B_j}c_j$
    Solve: ${}^{B_j}\mathbf{x}_{B_k}^{(i)} = \sum_j \|e_j\|^2$
    $i \leftarrow i + 1$
  end while
  return ${}^{B_j}\hat{\mathbf{x}}_{B_k}$
end function

The output is the mean transformation ${}^{B_j}\hat{\mathbf{x}}_{B_k}$ and its covariance R_r, forming a relative pose observation:

$$z_r = {}^{B_j}\hat{\mathbf{x}}_{B_k}, \quad R_r = \text{Cov}_{\text{ICP}}$$

4.6 Data Association with ICNN

To avoid incorrect associations, we evaluate the geometric consistency of the ICP result using the ICNN (Individual Compatibility Nearest Neighbor) criterion.

Mahalanobis Distance Test

Given:

- Predicted pose: ${}^{B_j}\hat{\mathbf{x}}_{B_k}$ with covariance $P_{B_k}^{B_j}$
- Observed pose (from ICP): z_r with covariance R_r

We define the innovation:

$$e = z_r - {}^{B_j}\hat{\mathbf{x}}_{B_k}, \quad S = P_{B_k}^{B_j} + R_r$$

The Mahalanobis distance is then:

$$D^2 = e^\top S^{-1} e$$

A match is accepted if:

$$D^2 < \chi^2_{3,\alpha}$$

where α is the confidence level (e.g., 0.95) and $\chi^2_{3,\alpha}$ is the chi-square threshold with 3 degrees of freedom.

ICNN Association Algorithm

Algorithm 2 ICNN: Mahalanobis-Based Scan Compatibility

$\mathcal{H}_o \leftarrow$ Overlapping scan indices
$c \leftarrow 1$; $\mathcal{H}_p \leftarrow [\,]$
for $i = 1$ to length($\mathcal{H}_o$) **do**
  $j \leftarrow \mathcal{H}_o[i]$
  Retrieve $[B_jS_j, B_jR_j]$ from map
  Predict relative pose: ${}^{B_j}\hat{\mathbf{x}}_{B_k} = ({}^{\mathcal{N}}\mathbf{x}_{B_j})^{-1} \oplus {}^{\mathcal{N}}\mathbf{x}_{B_k}$
  Compute $P_{B_k}^{B_j} = J_1 P_{B_j} J_1^\top + J_2 P_{B_k} J_2^\top$
  $[z_r, R_r] \leftarrow$ REGISTER($B_jS_j, B_kS_k, {}^{B_j}\hat{\mathbf{x}}_{B_k}$)
  $e \leftarrow z_r - {}^{B_j}\hat{\mathbf{x}}_{B_k}$; $S \leftarrow P_{B_k}^{B_j} + R_r$
  if $e^\top S^{-1} e < \chi^2_{3,\alpha}$ **then**
    Accept: $z_p[c] \leftarrow z_r$; $R_p[c] \leftarrow R_r$; $\mathcal{H}_p[c] \leftarrow i$; $c \leftarrow c + 1$
  end if
end for
return $\mathcal{H}_p$, z_p, R_p

This test filters out unreliable matches and ensures that only geometrically consistent scan registrations are passed to the EKF update step, improving the robustness and stability of the SLAM system.

4.7 Observation Model

The observation model in Pose-based EKF SLAM integrates relative pose measurements obtained from scan matching into the SLAM state. These measurements are expressed as constraints between the current pose ${}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}$ and previously stored poses ${}^{\mathcal{N}}\hat{\mathbf{x}}_{B_j}$.

Let the accepted measurement be:

$$z_p[i] = {}^{B_k}\hat{\mathbf{x}}_{B_j}, \quad R_p[i] = \text{Covariance from ICP}$$

These correspond to the registered transformation and its uncertainty from the current scan B_kS_k to reference scan B_jS_j.

Observation Function

For a set of n_p accepted pairwise registrations, the observation function is defined as:

$$z_r = h({}^{\mathcal{N}}\hat{\mathbf{x}}_k, v_k)$$

with the stacked residuals:

$$z_r = \begin{bmatrix} z_m \\ z_p \end{bmatrix} = \begin{bmatrix} h_r({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}, {}^{\mathcal{N}}\hat{\mathbf{x}}_Y, v_y) \\ h_r({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}, {}^{\mathcal{N}}\hat{\mathbf{x}}_B, v_b) \\ h_r({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}, {}^{\mathcal{N}}\hat{\mathbf{x}}_R, v_r) \end{bmatrix}$$

The residual function h_r is the relative transformation between two poses under noise v_k, defined as:

$$h_r({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}, {}^{\mathcal{N}}\hat{\mathbf{x}}_j, v_k) = ({}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k})^{-1} \oplus {}^{\mathcal{N}}\hat{\mathbf{x}}_j + v_k$$

Linearization and Jacobians

The measurement Jacobian H_k and noise Jacobian V_k are obtained by linearizing the observation function:

$$H_k = \left.\frac{\partial h({}^{\mathcal{N}}\hat{\mathbf{x}}_k, v_k)}{\partial {}^{\mathcal{N}}\hat{\mathbf{x}}_k}\right|_{v_k=0}, \quad V_k = \left.\frac{\partial h({}^{\mathcal{N}}\hat{\mathbf{x}}_k, v_k)}{\partial v_k}\right|_{v_k=0}$$

As shown in the figure, for each accepted measurement between ${}^{\mathcal{N}}\hat{\mathbf{x}}_{B_k}$ and a past pose ${}^{\mathcal{N}}\hat{\mathbf{x}}_j$, the observation Jacobians are built by inserting local blocks:

$$H_k = \begin{bmatrix} \cdots & J_1 J_{\ominus} & \cdots & J_2 J_{\oplus} & \cdots \end{bmatrix}$$

Where: - $J_1 = \frac{\partial \ominus x}{\partial x_1}$, $J_2 = \frac{\partial \oplus x}{\partial x_2}$ - $J_{\ominus}$ and $J_{\oplus}$ are Jacobians of the inverse and composition operations in $SE(2)$ - $V_k = I_{3\times 3}$ for each relative measurement noise component

4.8 EKF Update

Once valid relative pose measurements are obtained and validated (e.g., via ICNN), they are incorporated into the SLAM state using the standard Extended Kalman Filter (EKF) correction step.

Let the observation model be defined as:

$$z_k = h({}^{\mathcal{N}}\hat{\mathbf{x}}_k) + v_k, \quad v_k \sim \mathcal{N}(0, R_k)$$

The innovation (or residual) between the actual measurement and predicted observation is:

$$\tilde{z}_k = z_k - h({}^{\mathcal{N}}\hat{\mathbf{x}}_k)$$

The innovation covariance is computed as:

$$S_k = H_k P_k H_k^{\top} + V_k R_k V_k^{\top}$$

Then, the Kalman gain is:

$$K_k = P_k H_k^{\top} S_k^{-1}$$

The state estimate and covariance are updated as:

$${}^{\mathcal{N}}\hat{\mathbf{x}}_k \leftarrow {}^{\mathcal{N}}\hat{\mathbf{x}}_k \oplus K_k \tilde{z}_k$$

$$P_k \leftarrow (I - K_k H_k) P_k (I - K_k H_k)^{\top} + K_k V_k R_k V_k^{\top} K_k^{\top}$$

This update step integrates all accepted relative pose constraints into the SLAM state. It adjusts both the mean pose and the uncertainty by accounting for the observation Jacobians H_k, the measurement noise R_k, and the noise influence matrix V_k.

5 Experiment and Results

To validate the performance of the proposed PEKF-SLAM system, we conducted experiments in three different setups: RViz simulation, the Stonefish simulator, and real-world TurtleBot deployment. The system was evaluated in terms of scan alignment, trajectory accuracy, uncertainty evolution, and registration quality.

5.1 Environment Setup and Robot Platform

We used a TurtleBot platform equipped with a 2D LIDAR and IMU for all tests. The environments included grid-based RViz maps and a realistic 3D world in Stonefish. As shown in Fig. 2, this hardware setup was consistent across all trials. The simulation layout is shown in Fig. 3, while the resulting map reconstructed using RViz is shown in Fig. 4.

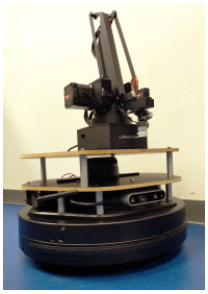

Fig. 2: TurtleBot used in real-world SLAM testing.

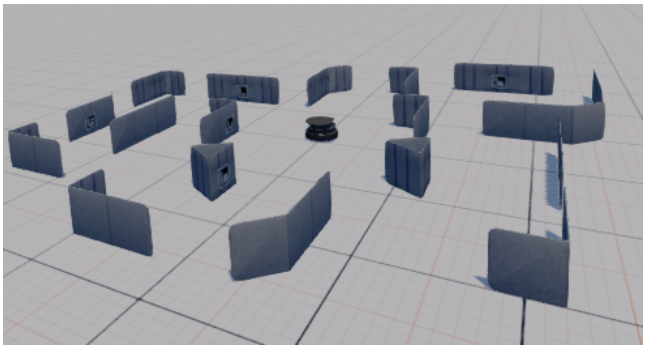

Fig. 3: Stonefish simulation environment used for virtual experiments.

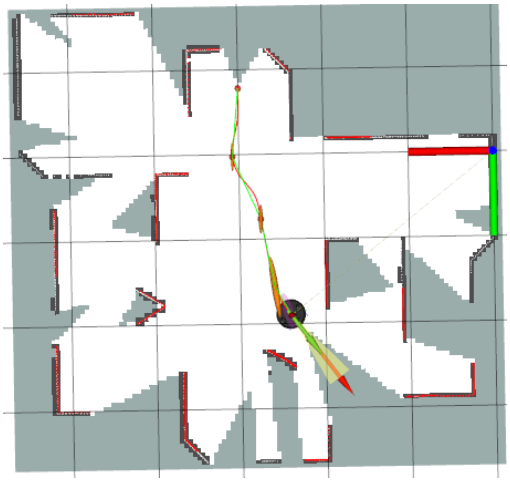

Fig. 4: Reconstructed map in RViz during SLAM operation.

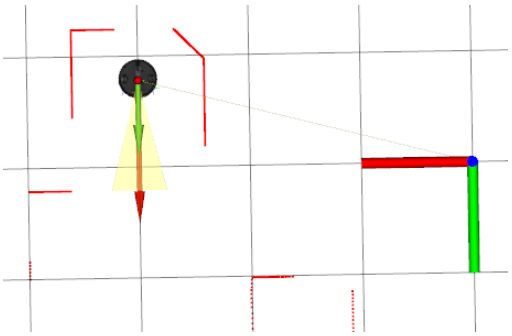

Fig. 5: Initial pose of TurtleBot before navigation.

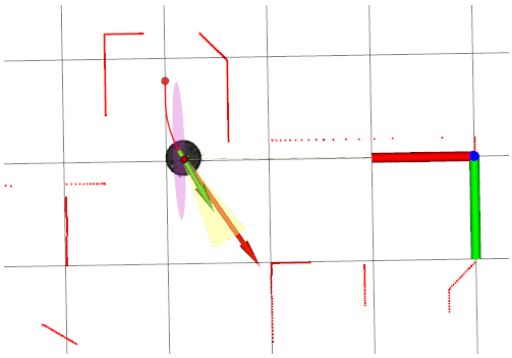

Fig. 6: Initial EKF covariance ellipses visualized in RViz.

5.2 State and Covariance Initialization

Initial pose and covariance are visualized in Fig. 5 and Fig. 6, respectively. The robot started with low uncertainty, shown by small ellipses centered around the initial position.

5.3 Scan Alignment Quality

Correct scan alignment is essential for accurate pose estimation. As shown in Fig. 7, ICP successfully aligns white point clouds scan data. In contrast, Fig. 8 shows misalignment due to poor registration or invalid data association.

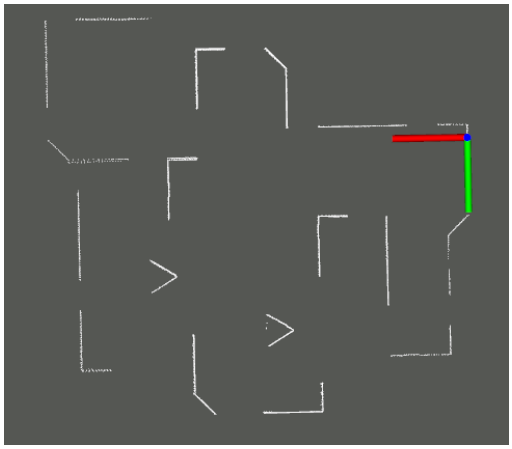

Fig. 7: Correctly aligned scans: green and red segments match.

5.4 Pose Uncertainty and Trajectory Accuracy

As shown in Fig. 9, EKF SLAM maintains a bounded 3σ uncertainty over time, while dead reckoning accumulates significant error. In Fig. 10, SLAM results closely follow the ground truth for x, y, and θ, confirming accurate tracking performance.

5.5 Fitness Score and RMSE Analysis

Scan registration quality was evaluated using two metrics:

- **Fitness Score:**

$$\text{Fitness} = \frac{\text{Matched scan points}}{\text{Total points}} \times 100$$

Fig. 8: Misaligned scans due to ICP failure or incorrect association.

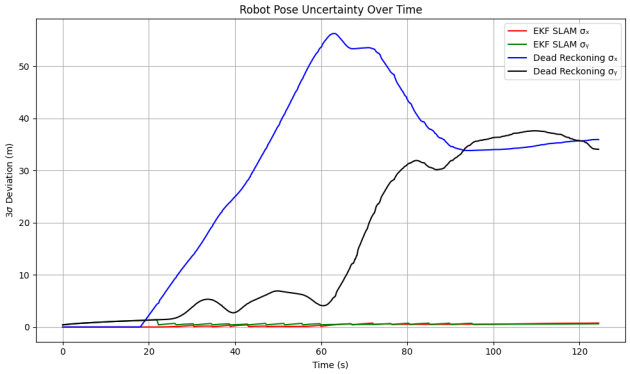

Fig. 9: 3σ uncertainty over time: SLAM (red/green) vs. Dead Reckoning (blue/black).

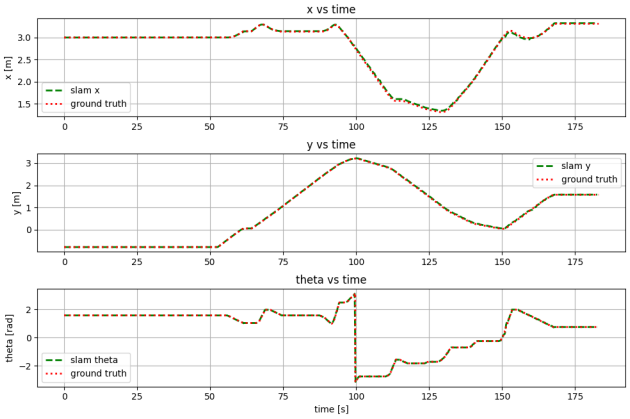

Fig. 10: Pose estimation vs. ground truth for x, y, and θ.

Values consistently exceeded 80%, indicating strong overlap between scans.

- **RMSE:**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2}$$

Values remained below 0.01 m, reflecting highly ac-

curate scan alignment.

5.6 Loop Closure Result

Finally, Fig. 11 shows the result after a loop closure event. The robot trajectory aligns tightly with previous scans, demonstrating the effectiveness of ICNN-based validation and EKF update during global correction.

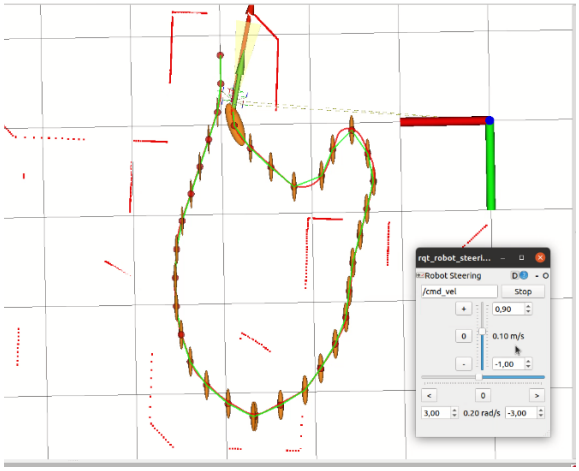

Fig. 11: Trajectory after loop closure: well-aligned scan history.

6 Conclusion

This paper presented a modular Pose-Based EKF SLAM system that integrates wheel odometry, IMU heading correction, and LIDAR scan matching using the ICP algorithm. The system supports real-time simulation pose estimation and mapping in structured indoor environments, leveraging EKF prediction, scan registration, and Mahalanobis-based data association for robustness.

Experimental results from simulation deployments demonstrate that the proposed approach improves localization accuracy and maintains bounded pose uncertainty compared to dead reckoning alone.

Future work will focus on extending the system with real robot to improve performance in dynamic or large-scale environments.

Click here Link to video
Click here Link to Github

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *AAAI*, vol. 593, 2002, p. 598.

[3] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[6] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems*, vol. 2, 2009, p. 435.

[7] P. J. Besl, Z. Zhang *et al.*, "A survey of icp variants and applications in robotics," *Robotics and Autonomous Systems*, vol. 166, p. 104407, 2023.

[8] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2021.

[9] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3607–3613.

[10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[11] S. J. Julier and J. K. Uhlmann, "Using the marginalized likelihood ratio for multi-hypothesis data association," *Proceedings of SPIE*, vol. 6206, 2006.

[12] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[13] T. Shan and B. Englot, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.