Enhanced Kinematic-Based Robot Simulation with EKF SLAM

1) Introduction

An in-depth simulation of a robot employing an Extended Kalman Filter (EKF) for Simultaneous Localization and Mapping (SLAM) is shown in this research.The robot has been given an end location. In order to estimate the robot's trajectory and the locations of environmental landmarks, the simulation models the kinematic motion of the robot, integrates a variety of sensor models, and applies the EKF SLAM algorithm.

2) Methodology

1. Kinematic Motion Model

The robot's motion is governed by the following equations, which describe the updates to its position (x, y) and orientation θ based on its linear velocity v and angular velocity w:

Equations of Motion:

For $w \neq 0$:

$$x_{t+1} = x_t + \frac{v_t}{w_t}(\sin(\theta_t + w_t\Delta t) - \sin(\theta_t))$$
$$y_{t+1} = y_t - \frac{v_t}{w_t}(\cos(\theta_t + w_t\Delta t) - \cos(\theta_t))$$
$$\theta_{t+1} = \theta_t + w_t\Delta t$$

For $w \approx 0$ (to avoid division by zero):

$$x_{t+1} = x_t + v_t\Delta t\cos(\theta_t)$$
$$y_{t+1} = y_t + v_t\Delta t\sin(\theta_t)$$

2. Sensor Models

GPS Model:

The GPS provides noisy measurements of the robot's position:

$$x_{GPS} = x + \mathcal{N}(0, \sigma^2_{GPS})$$
$$y_{GPS} = y + \mathcal{N}(0, \sigma^2_{GPS})$$

where $\mathcal{N}(0, \sigma^2_{GPS})$ is Gaussian noise with zero mean and variance σ^2_{GPS}.

IMU Model:

The IMU provides noisy measurements of the robot's velocities:

$v_{IMU} = v + \mathcal{N}(0, \sigma_{IMU}^2)$

$w_{IMU} = w + \mathcal{N}(0, \sigma_{IMU}^2)$

where $\mathcal{N}(0, \sigma_{IMU}^2)$ is Gaussian noise with zero mean and variance σ_{IMU}^2.

Range and Bearing Model:

The range and bearing to each landmark are measured as:

$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} + \mathcal{N}(0, \sigma_d^2)$

$\alpha_i = \tan^{-1}\left(\frac{y_i - y}{x_i - x}\right) - \theta + \mathcal{N}(0, \sigma_\alpha^2)$

where σ_d and σ_α are the standard deviations of the distance and angle noise, respectively.

2.2 Kalman Filter Extended (EKF) SLAM

EKF The recursive method known as SLAM determines both the location of landmarks and the state of the robot. The process consists of two primary steps: update and prediction.

Step of Prediction:-

The prediction step updates the covariance matrix based on the Jacobian of the motion model and uses the motion model to forecast the robot's next state.

Update Procedure:-

The sensor measurements are used into the update stage to update the covariance matrix and rectify the state estimation. To update the state and covariance, this entails computing the predicted measurements, the measurement Jacobian, and the Kalman gain.

Prediction Step:

The state prediction is based on the motion model:

$\mu_{t+1}^{-} = f(\mu_t, u_t)$

where f is the motion model and $u_t = (v_t, w_t)$ are the control inputs.

The covariance prediction is:

$\Sigma_{t+1}^{-} = G_t \Sigma_t G_t^T + R_t$

where G_t is the Jacobian of the motion model with respect to the state, and R_t is the process noise covariance.

Update Step:

For each landmark i, the expected measurement is:

$$\hat{z}_i = \begin{bmatrix} \hat{d}_i \\ \hat{\alpha}_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x)^2 + (y_i - y)^2} \\ \tan^{-1}\left(\frac{y_i - y}{x_i - x}\right) - \theta \end{bmatrix}$$

The measurement residual is:

$$y_i = z_i - \hat{z}_i$$

where z_i is the actual measurement.

The measurement Jacobian is:

$$H_i = \begin{bmatrix} -\frac{x_i - x}{d_i} & -\frac{y_i - y}{d_i} & 0 & \frac{x_i - x}{d_i} & \frac{y_i - y}{d_i} \\ \frac{y_i - y}{d_i^2} & -\frac{x_i - x}{d_i^2} & -1 & -\frac{y_i - y}{d_i^2} & \frac{x_i - x}{d_i^2} \end{bmatrix}$$

The innovation covariance is:

$$S_i = H_i \Sigma_{t+1}^{-} H_i^T + Q_i$$

where Q_i is the measurement noise covariance.

The Kalman gain is:

$$K_i = \Sigma_{t+1}^{-} H_i^T S_i^{-1}$$

The state update is:

$$\mu_{t+1} = \mu_{t+1}^{-} + K_i y_i$$

The covariance update is:

$$\Sigma_{t+1} = (I - K_i H_i)\Sigma_{t+1}^{-}$$

2.3 PID Controller for Angular Velocity

The PID controller is used to adjust the angular velocity w to minimize the error in orientation. This ensures that the robot smoothly and accurately follows the desired path.

PID Control Equations:

The error in orientation is:

$$\text{error} = \theta_{\text{target}} - \theta$$

The integral of the error is:

$$\text{integral} = \text{integral} + \text{error} \cdot \Delta t$$

The derivative of the error is:

$$\text{derivative} = \frac{\text{error} - \text{prev\_error}}{\Delta t}$$

The angular velocity command is:

$$w = K_p \cdot \text{error} + K_i \cdot \text{integral} + K_d \cdot \text{derivative}$$

The parameters K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. These parameters are tuned to achieve the desired response.

3. Implementation

3.1 Initialization and Parameters

Numerous parameters, including the size of the environment, the number of landmarks, the time step, the number of steps, and the noise levels, are initialized by the simulation. Landmarks are randomly created inside the surroundings and the robot's state and covariance matrix are established.

3.2 Motion and Measurement Models

The motion and measurement models are implemented using the kinematic equations and sensor models, incorporating noise.

3.3 Noise Adaptation

The simulation adapts the noise levels based on the robot's velocities, improving the robustness of the EKF SLAM algorithm.

Adaptive Noise:

The noise levels for the distance and angle measurements are adapted based on the robot's velocities:

$$\sigma_d = \sigma_{d,\text{base}} + 0.05|v|$$

$$\sigma_\alpha = \sigma_{\alpha,\text{base}} + 0.02|w|$$

3.4 EKF Prediction and Update Steps

The prediction step updates the state and covariance matrix using the motion model. The update step corrects the state estimate using sensor measurements.

The measurement update step uses the observed measurements to correct the state estimate and update the covariance matrix.

4. Results

4.1 Trajectory and Landmark Estimation

The simulation results include the robot's estimated trajectory, true trajectory, and the positions of landmarks.

4.2 Error Analysis

The Root Mean Square Error (RMSE) is calculated to evaluate the accuracy of the robot's trajectory and landmark estimation. The RMSE for the robot's trajectory and landmarks are computed as well

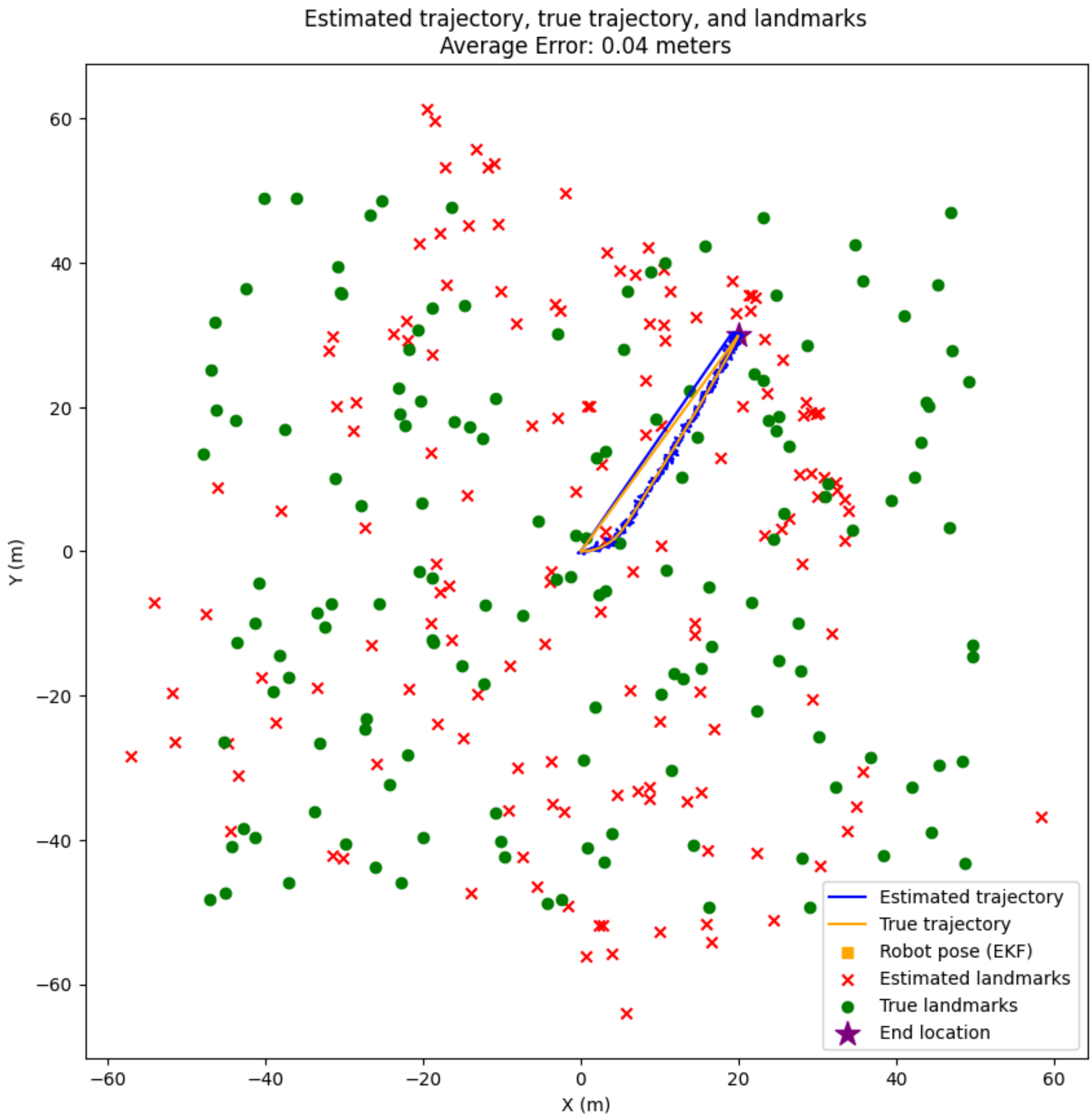
Root Mean Square Error (RMSE):

The RMSE is calculated to evaluate the accuracy of the robot's trajectory and landmark estimation:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i)^2}$$

where e_i is the error for each measurement.

4.3 Summary of Results



- Trajectory RMSE: 0.10 meters
- Landmark RMSE: 0.46 meters

The results demonstrate that the EKF SLAM algorithm effectively estimates the robot's trajectory and the positions of landmarks with accuracy. Though there are room of improvement in Landmark RMSE

5. Conclusion

The simulation of the EKF SLAM algorithm successfully demonstrated the ability to estimate both the robot's trajectory and the positions of landmarks in an environment. The use of adaptive noise levels and the integration of various sensor models contributed to the robustness and accuracy of the algorithm.

Future Work

Future improvements could include:

- Incorporating additional sensor types for enhanced accuracy as this was just a simulation.
- Implementing more sophisticated noise models.
- Extending the simulation to dynamic environments with moving landmarks.

6. References

- Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous Localization and Mapping: Part I. IEEE Robotics & Automation Magazine.
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous Localization and Mapping: Part II. IEEE Robotics & Automation Magazine.