Localization of a Mobile Robot Using Extended Kalman Filter and Particle Filter

Muhammad Haseeb Khan, Muhammad Osama Fawad

May 16, 2025

Abstract

This report presents a study and implementation of two probabilistic algorithms for mobile robot localization: the Extended Kalman Filter (EKF) and the Particle Filter (PF). The goal is to estimate the robot's pose in a 2D environment using noisy odometry and landmark-based bearing observations. We describe the theoretical foundations, the motion and observation models, the implementation details, and analyze the performance of both filters through experimental results.

Contents

1 Introduction

Robot localization is a fundamental problem in mobile robotics, where the robot estimates its position and orientation (pose) in a given environment. Due to inherent noise in sensors and control commands, probabilistic methods are employed to maintain and update belief over the robot's state. This report focuses on implementing and analyzing two widely-used probabilistic localization techniques: the Extended Kalman Filter (EKF) and the Particle Filter (PF).

Both algorithms use a combination of motion models and sensor observations to iteratively improve the robot's pose estimate. EKF approximates the belief as a Gaussian distribution and linearizes nonlinear models, while PF represents the belief with a set of weighted samples (particles) to handle more complex distributions.

2 Theoretical Background

2.1 Robot State and Control

The robot's state at time t is represented as the pose vector:

$$\mathbf{s}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

where x_t, y_t are Cartesian coordinates, and θ_t is the orientation angle.

Control inputs $\mathbf{u}_t$, derived from odometry, are given by:

$$\mathbf{u}_t = \begin{bmatrix} \delta_{rot1} \\ \delta_{trans} \\ \delta_{rot2} \end{bmatrix}$$

representing an initial rotation, translation, and final rotation.

2.2 Motion Model

The robot motion model predicts the next state based on the current state and control input:

$$\begin{cases} x_{t+1} = x_t + \delta_{trans} \cos(\theta_t + \delta_{rot1}) \\ y_{t+1} = y_t + \delta_{trans} \sin(\theta_t + \delta_{rot1}) \\ \theta_{t+1} = \theta_t + \delta_{rot1} + \delta_{rot2} \end{cases}$$

This nonlinear model is implemented in the `soccerfield.py` module and utilized by both EKF and PF during the prediction step.

2.3 Observation Model

The robot observes bearing angles $\theta_{bearing}$ to known landmarks corrupted by noise parameter β. Landmark identities are assumed known and noise-free. This observation model is defined in `soccerfield.py` and forms the basis for measurement updates in both filters.

2.4 Extended Kalman Filter (EKF)

The EKF approximates nonlinear state and observation models by linearizing them about the current estimate using Jacobians:

$$G = \frac{\partial g}{\partial s}, \quad V = \frac{\partial g}{\partial u}$$

where g represents the motion model function.

The EKF algorithm proceeds with:

- **Prediction:** Computing the predicted state $\hat{\mathbf{s}}_{t+1}$ and covariance Σ_{t+1} incorporating process noise.
- **Update:** Calculating the Kalman gain

$$K = \Sigma_{pred} H^T (H \Sigma_{pred} H^T + R)^{-1}$$

  and correcting the predicted state and covariance based on observations, where H is the observation Jacobian and R the measurement noise covariance.

These steps are implemented in `ekf.py`, where the Jacobians and Kalman gain computations directly correspond to the mathematical formulation.

2.5 Particle Filter (PF)

The PF represents the belief distribution with a set of weighted particles $\{(s_i, w_i)\}$. It performs:

- Sampling each particle forward using the motion model with noise.
- Computing particle weights based on observation likelihood.
- Resampling particles using systematic low-variance resampling to focus on high-weight particles.
- Estimating the robot pose as the weighted mean of particles.

This approach, implemented in `pf.py`, better handles nonlinear and non-Gaussian distributions than EKF, at the expense of increased computational complexity.

2.6 Code Integration

The motion and observation models are encapsulated in `soccerfield.py` and utilized by both filter implementations. The main script, `localization.py`, orchestrates experiments, accepts noise scaling parameters via command-line arguments, executes EKF or PF algorithms, and visualizes results. This modular design aligns theoretical concepts with practical implementation for clear traceability.

3 Implementation Details

The implementation is done in Python, utilizing NumPy for computations and Matplotlib for visualization.

- localization.py — main script to run experiments and visualize results.
- ekf.py — contains EKF implementation.
- `pf.py` — contains PF implementation.
- `soccerfield.py` — defines motion and observation models, noise, and Jacobians.
- `utils.py` — helper functions for angle normalization and plotting.

Command-line usage examples:

```
python localization.py --plot none       # plots ground truth and odometry
python localization.py --plot ekf        # plots EKF estimate
python localization.py --plot pf         # plots PF estimate
```

4 Experimental Results and Analysis

4.1 EKF Results

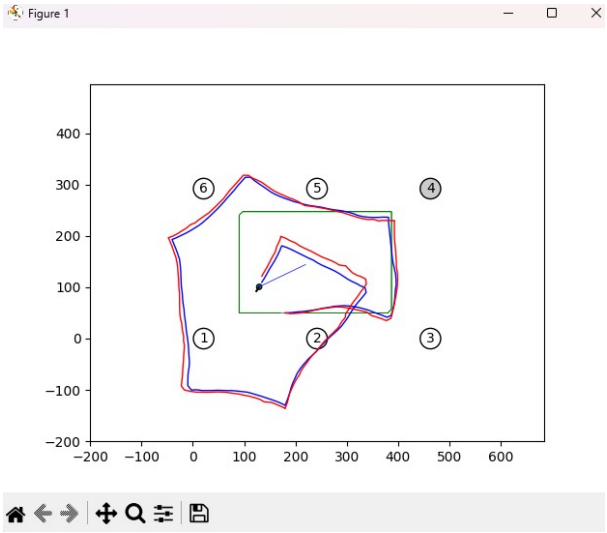

Figure 1: Robot path estimation by EKF (blue) compared to ground truth (red) and odometry (green rectangle).

The EKF provides a smooth estimate of the robot trajectory. The reported error metrics are:

- Mean Position Error: 8.9936753
- Mean Mahalanobis Error: 4.4164182
- ANEES: 1.472139

```
Mean position error: 8.998367536084691
Mean Mahalanobis error: 4.416418248584292
ANEES: 1.472139416194764
```

Figure 2: EKF numerical results printed from the console.

4.2 Particle Filter Results

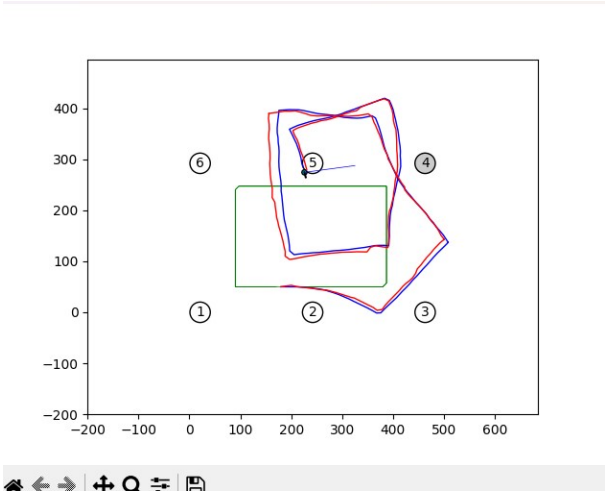

Figure 3: Robot path estimation by Particle Filter (blue) compared to ground truth (red) and odometry (green rectangle).

PF error metrics:

- Mean Position Error: 8.56726437
- Mean Mahalanobis Error: 14.74225277
- ANEES: 4.91408425

```
Mean position error: 8.567264372950909
Mean Mahalanobis error: 14.742252771106532
ANEES: 4.914084257035511
```

Figure 4: Particle Filter numerical results printed from the console.

4.3 Impact of Noise Scale on Position Error and ANEES

Figure 5 Noise scaling impacts both EKF and PF localization: when input and filter noise increase together, position error stays low at moderate noise but rises sharply at high levels, showing sensitivity to large uncertainties. ANEES remains near ideal under moderate noise but worsens at extremes, indicating less reliable uncertainty estimates. Varying only filter noise results in more stable position error, emphasizing the importance of accurately modeling input noise for effective localization.

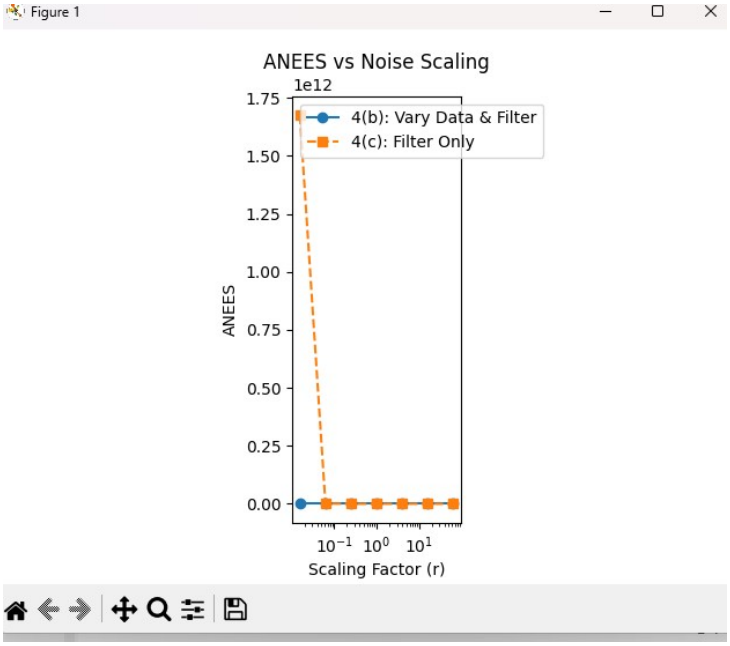

Figure 5: Effect of noise scale on mean position error and ANEES for EKF and PF. The blue line corresponds to experiments varying both input and filter noise parameters, while the orange dashed line corresponds to varying filter noise only.

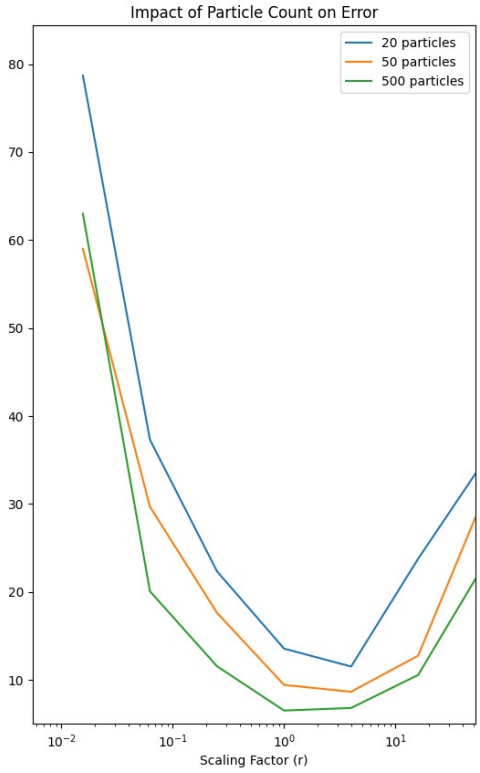

Figure 6: Mean position error as a function of noise scaling factor for different numbers of particles in the Particle Filter. Higher particle counts yield improved accuracy.

4.4 Impact of Particle Count on Particle Filter Performance

As shown in Figure 6, The Particle Filter's accuracy depends heavily on the number of particles used. Low counts (e.g., 20) lead to poor pose tracking and high errors, especially under noise. Increasing particles to 50 and 500 reduces position error by better approximating the posterior distribution. This illustrates the trade-off between accuracy and computational cost, requiring careful selection of particle count based on real-time and precision needs.

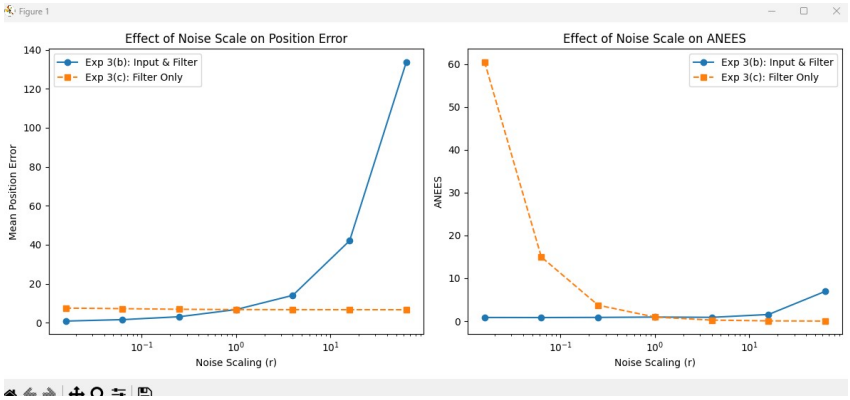

Figure 7: Position error and ANEES variation with noise scale for Extended Kalman Filter: comparison between varying both input and filter noise (solid line) and varying filter noise only (dashed line).

Figure 7 The Extended Kalman Filter is highly sensitive to noise scaling: increasing both input and filter noise sharply raises position error, while varying only filter noise causes moderate changes. Accurate input noise modeling is crucial for EKF reliability. ANEES values confirm consistent estimation at moderate noise but degrade as noise increases.

4.5 Position Error Across Noise Scales for Particle Filter

As seen in Figure 8, The Particle Filter maintains low position error at small noise levels when both data and filter noise vary, but error rises sharply as noise increases. When only filter noise varies, error follows a U-shaped curve—decreasing then increasing—showing the PF's robustness yet the challenge of tuning noise parameters for optimal accuracy.

4.6 Analysis Summary

These additional experimental results reinforce key insights into the behavior of EKF and PF under varying noise conditions:

- EKF is more sensitive to input noise modeling, with sharp degradation in position accuracy as noise scales up, confirming the importance of precise noise parameter tuning.
- PF exhibits a more complex response, with noise scale impacting performance non-linearly. Its particle-based representation allows it to maintain lower error at moderate noise but requires careful balance of noise parameters.

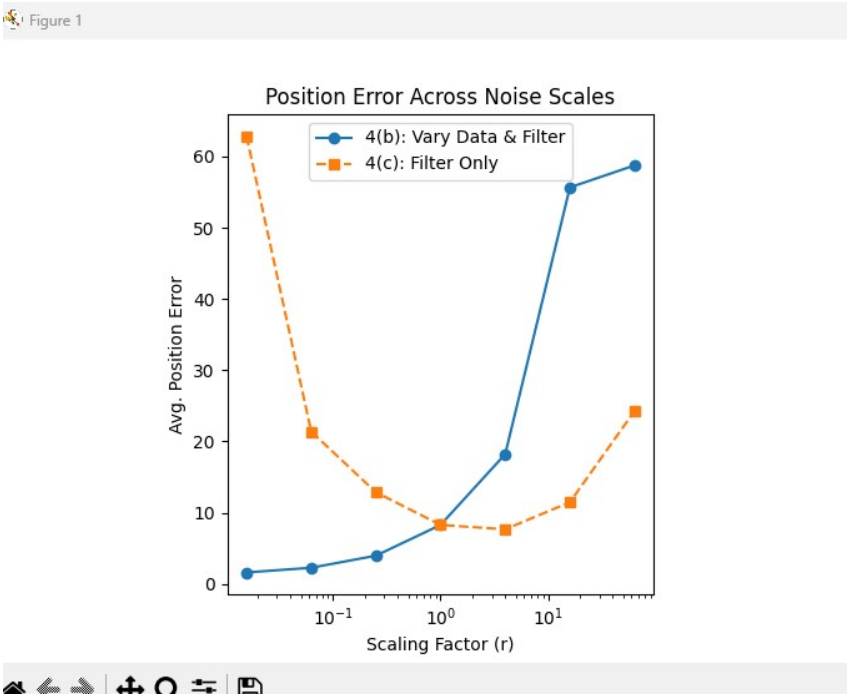

Figure 8: Average position error for Particle Filter across varying noise scales comparing experiments with varying data and filter noise versus filter noise only.

- Both filters demonstrate that ignoring or underestimating noise in motion inputs can lead to substantial localization errors.
- These findings emphasize the trade-offs in filter design and noise characterization necessary for robust mobile robot localization.

4.7 EKF Performance under Varying Data and Filter Noise Factors

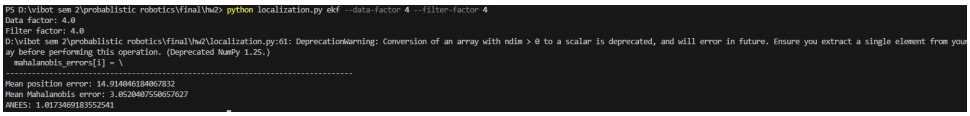

Figure 9: EKF output for Data Factor = 4 and Filter Factor = 4: Mean position error = 14.91, Mahalanobis error = 3.05, ANEES = 1.02.

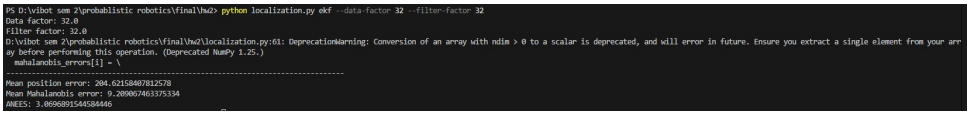

Figure 10: EKF output for Data Factor = 32 and Filter Factor = 32: Mean position error = 204.62, Mahalanobis error = 9.21, ANEES = 3.07.

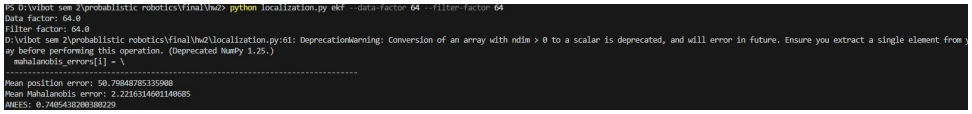

Figure 11: EKF output for Data Factor = 64 and Filter Factor = 64: Mean position error = 50.80, Mahalanobis error = 2.22, ANEES = 0.74.

Analysis and Comparison

Figures 9 to 11 present the EKF performance metrics under increasing noise scaling factors for both data and filter noise parameters.

Key observations:

- At low noise scaling (4.0), EKF maintains relatively low mean position error (14.91) and stable consistency (ANEES 1.02), indicating accurate and reliable pose estimation.
- Increasing noise scale to 32 results in a dramatic spike in mean position error (204.62) and Mahalanobis error (9.21), while ANEES increases above 3, suggesting a significant loss in filter accuracy and degraded uncertainty estimation.
- Surprisingly, at an even higher noise scale of 64, the mean position error decreases to 50.80, and ANEES drops below 1 (0.74), indicating the filter's estimation error and covariance are somewhat better aligned, although error remains substantially larger than at low noise.

The EKF's non-monotonic error behavior likely results from sensitivity to noise tuning and nonlinear or numerical instabilities at extreme noise levels. The error spike at moderate-high noise shows reduced filter reliability, while improvement at the highest noise may be due to covariance inflation or compensations. This highlights the need for careful noise parameter selection, as noise increase doesn't always cause a straightforward decline in performance.

Comparison to Previous Noise Analysis

Compared to earlier noise scale experiments (Figure 5), these results provide more granularity at specific scaling factors and confirm that:

- EKF accuracy and consistency deteriorate quickly with increasing noise.
- Filter performance is not strictly monotonic with noise increase, indicating potential for filter tuning to mitigate noise impact.
- Monitoring ANEES alongside position error is crucial to assess filter reliability and covariance consistency.

5 Discussion

- EKF performs well with Gaussian noise and smooth trajectories, offering low computational cost.

- PF can model complex, multimodal beliefs but requires more particles for accuracy.
- Increasing noise scaling factors beyond a threshold drastically increases position error for both filters.
- Particle count significantly influences PF accuracy and computational cost trade-off.

6 Conclusion

This work successfully implemented and compared the EKF and PF for mobile robot localization in a landmark-based 2D environment. Both filters show acceptable localization accuracy under nominal noise conditions. EKF is computationally efficient but limited by linearization assumptions, while PF offers robustness to nonlinearities at higher computational cost. Future extensions may involve adaptive noise modeling, 3D localization, and integration of additional sensor modalities.

References

- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
- Relevant lecture slides and class notes.