

CSCI 4302/5302 Advanced Robotics

Assignment 2: Extended Kalman Filter

Due: 4/11

1 Introduction

In this assignment, you will implement an Extended Kalman Filter (EKF) for robot localization in a continuous world with nonlinear dynamics. The EKF extends the Kalman filter to handle nonlinear systems by linearizing around the current state estimate.

2 Background

The EKF maintains a Gaussian belief over continuous states. For a robot in a continuous world:

- State space: (x, y, θ) representing position and orientation
- Actions: forward movement and turns with continuous velocities
- Measurements: range and bearing to landmarks

The filter consists of two main steps:

1. Prediction:

$$\begin{aligned}\mu_t &= g(u_t, \mu_{t-1}) \\ \Sigma_t &= G_t \Sigma_{t-1} G_t^T + R_t\end{aligned}$$

where G_t is the Jacobian of the motion model.

2. Update:

$$\begin{aligned}K_t &= \Sigma_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1} \\ \mu_t &= \mu_t + K_t (z_t - h(\mu_t)) \\ \Sigma_t &= (I - K_t H_t) \Sigma_t\end{aligned}$$

where H_t is the Jacobian of the measurement model.

3 Implementation Tasks

3.1 Filter Implementation

Implement the main filter methods:

- Prediction step using linearized motion model
- Update step using linearized measurement model
- Kalman gain computation
- State and covariance updates

Note: The EKF can be sensitive to tuning. Try to find parameters that allow your filter to track the target well, but do not be concerned if the tracking is poor. This is difficult to localize with just a few lidar beams with nonlinear motion.

3.2 Analysis and Visualization

- Plot estimated trajectory vs. ground truth
- Visualize uncertainty ellipses
- Analyze linearization errors
- Compare performance with different noise parameters

4 Getting Started

4.1 Installation

Install the package with student dependencies:

```
pip install -e .
```

4.2 Code Structure

The template file `extended_kalman_filter.py` contains:

- Class definition with required methods
- Docstrings explaining expected behavior
- TODO comments marking implementation points
- Helper functions for matrix operations

4.3 Testing

Run the provided test suite:

```
pytest tests/test_extended_kalman_filter.py -v  
pytest tests/test_ekf_vis.py -v
```

5 Submission Instructions

Link to github branch.

<https://github.com/gyanigk/Advanced-Robotics/tree/homework-5-part-2.gyanigkumar>

