

README

Overview

This repository contains a MATLAB script named **Robot2DLocalizationGUI.m** that demonstrates a Kalman Filter for 2D localization in a position-velocity model. The code uses simulated accelerometer data for prediction steps and noisy position measurements for correction steps. A real-time playback visualization is provided in a graphical user interface (GUI).

Features

1. 2D Position-Velocity Kalman Filter

- Predicts the robot's state (position and velocity) using accelerometer data.
- Updates (corrects) the state using noisy position measurements.

2. Real-Time Playback

- Displays the true trajectory, estimated trajectory, and noisy measurements in a GUI.
- Plots an uncertainty ellipse to visualize the covariance at each time step.

3. Interactive Sliders

- Adjust process noise and measurement noise in real time.
- Observe changes in the filter performance interactively.

4. Uncertainty Visualization

- Plots an ellipse representing the covariance around the estimate, giving insight into the confidence of the estimation.

Requirements

- **MATLAB** (R2019b or later recommended).
- **Statistics and Machine Learning Toolbox** (for functions like `mvnrnd`).

How to Run

1. **Open MATLAB.**
2. **Navigate** to the directory containing `Robot2DLocalizationGUI.m`.
3. **Run** the script

4. A figure window with a set of controls will appear:
 - **Process Noise Std Slider:** Adjusts the level of process noise (how uncertain the filter is about the dynamics).
 - **Measurement Noise Std Slider:** Adjusts the standard deviation of the measurement noise.
 - **Run:** Starts the simulation and displays the real-time animation.

Explanation of Key Components

- **Kalman Filter:** A recursive algorithm that estimates the state of a dynamic system from a series of incomplete and noisy measurements.
- **Predict Step (`kalman_predict`):** Uses current state and control input (acceleration) to predict the next state and covariance.
- **Update Step (`kalman_update`):** Incorporates new measurements to correct the prediction, reducing uncertainty.

Files and Functions

- **Robot2DLocalizationGUI.m**

Main script that:

- Creates the GUI and its elements.
- Generates a simulated ground truth motion.
- Runs the Kalman Filter in real time.
- Calls helper functions for prediction, update, and plotting.

- **Helper Functions**

1. `kalman_predict(x, P, u, A, B, Q)`: Implements the prediction step of the Kalman Filter.
2. `kalman_update(x_pred, P_pred, z, H, R)`: Implements the update step of the Kalman Filter.
3. `plot_uncertainty_ellipse(ax, mu, Sigma, color)`: Draws an ellipse representing the covariance matrix on the specified axes.

Adjusting the Simulation

- **Simulation Duration:** The variable `T` sets the total simulation time, and `dt` sets the time step.

- **Model Matrices:** A (state transition), B (control input), and H (measurement) can be changed to model different dynamics or measurement setups.
- **Initial Conditions:** $x_true(:,1)$ and $x_est(:,1)$ can be adjusted to change the starting true and estimated states.

Troubleshooting & Tips

- If the legend overlaps important parts of the plot, you can move it manually within the figure.
- If the animation runs too quickly or too slowly, adjust the `pause(dt)` value.
- To see the influence of process noise and measurement noise more clearly, try varying the sliders across their ranges.