Aengus Kennedy

2025 04 18

Tufts ME0134

Lab 4

**Implementation Notes**

* I wrote all the code in this submission from scratch without the starting templates. I reused the forward kinematics code I wrote in Lab 3.
* The Kalman filter is implemented in the sixth cell of the file [lab\_4/l4data\_processing.ipynb](https://github.com/aengusk/ME0134-Aengus-Kennedy/blob/main/lab_3/l3data_processing.ipynb) accessible in my ZIP submission or online at <https://github.com/aengusk/ME0134-Aengus-Kennedy/>.

**Experiment**

The robot was commanded to follow the following sequence of straight trajectories and point turns.

* forward 2 meters (speed 150 cm/s)
* turn -1.5 rotations (speed 150 cm/s) (where negative rotations are in the clockwise direction)
* forward 5 meters (speed 150 cm/s)
* turn -2.5 rotations (speed 150 cm/s)
* forward 1.5 meters (speed 150 cm/s)
* turn -8 rotations (speed 100 cm/s)
* forward 4 meters (speed 100 cm/s)

**Results**

A graph of different modeling methods

AI-generated content may be incorrect.

Fig. 1: Comparison of Actual and Predicted Paths

by Different Modeling Methods

**Analysis**

I tuned the Kalman filter’s parameters by paying attention to how the **final yaw value** (corresponding to the angle of the final straight stretch of the robot’s path) predicted by the model changed as I changed its paramters. The robot’s true final yaw was about −5.67 rotations, right in between the value predicted by pure FK integration (−5.89 rotations, plotted in red) and pure IMU integration (−5.41, plotted in orange). The Kalman filter values of *P* = 0.03, *Q* = 0.0001, *R* = 1.0 best aligned with the true path (plotted in black) and predicted a final yaw of −5.56 (plotted in blue).

I chose a very small *Q* value because the

**Analysis**

aa

guiding questions:

* How close is the final position from the Kalman filter to the ground truth? Is it better than using only encoders?
* If you disabled the Kalman update step (i.e., ignored the sensor measurements), what kind of drift would accumulate?
* How does this experiment illustrate the importance of fusing sensor and model information?
* What did the Kalman filter correct that the forward kinematic model alone could not? Give a specific moment in your trajectory when this occurred.
* The prediction step relies on idealized motion models. When the robot spins quickly or moves at high speeds, what assumptions of the model might break down? How does this impact the Kalman filter?
* After completing the rotations, how well does the predicted θ (Kalman filter versus model only) match the actual heading of the robot when it transitions to straight-line motion?
* Note: Kalman filter is not built for this, where sensor noise is an integrated value. Kalman filter assumes noise is Additive, Gaussian, and has zero mean