Project 2: Kalman Filter
RBE 595: Advanced Robot Navigation

Miheer Diwan
Robotics Engineering
Worcester Polytechnic Institute
msdiwan@wpi.edu

I. Introduction

In this project, we utilize a Kalman filter to estimate the motion of a drone flying in an open space. The drone's motion is tracked using measurements from a Qualisys motion capture system, providing either position or velocity data. Our objective is to implement the Kalman filter to accurately estimate the drone's state based on the provided measurements.

II. Task 1: System Modeling

The system is modeled using a state vector consisting of the drone's position and velocity. Let $\mathbf{p} = [x, y, z]^T$ denote the position vector and $\dot{\mathbf{p}} = [\dot{x}, \dot{y}, \dot{z}]^T$ denote the velocity vector. With this representation, the state vector becomes $x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. The noise is assumed to be zero.

A. State Extrapolation (Process Model)

Since the Kalman filter is a discrete filter, we need to first, discretize the given system using one step Euler integration method. the first-order Markov assumption makes the Kalman Filter system model a first order differential equation.

$$\dot{x} = f(x, u, \mathcal{N}(0, Q)) = Ax + Bu + E\mathcal{N}(0, Q)$$

The state transition model is given by:

$$\dot{x} = Ax + Bu$$

The state transition matrix A remains the same as before:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The input matrix B becomes:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix}$$

Using one-step Euler integration we go from Continuous time to Discrete time:

$$F = I + \delta t A = \begin{bmatrix} 1 & 0 & 0 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G = \delta t B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\Delta t}{m} & 0 & 0 \\ 0 & \frac{\Delta t}{m} & 0 \\ 0 & 0 & \frac{\Delta t}{m} \end{bmatrix}$$

$$V = \delta t E$$

B. Measurement Model

Depending on whether the measurements are position or velocity, the measurement matrix C will be different.

If the measurement is position ($\mathbf{z} = \mathbf{p}$), then:

$$C_{\text{position}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

If the measurement is velocity ($\mathbf{z} = \dot{\mathbf{p}}$), then:

$$C_{\text{velocity}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

These are the updated state-space equations considering the position vector $\mathbf{p}$ in its x, y, and z components.

III. Kalman Filter Algorithm

$$\begin{aligned} \bar{\mu}_t &= F\mu_{t-1} + Gu_t \\ \bar{\Sigma}_t &= F\Sigma_{t-1}F^T + VQV^T \\ K &= \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1} \\ \mu_t &= \bar{\mu}_t + K(z_t - C\bar{\mu}_t) \\ \Sigma_t &= (\mathbf{I} - KC)\bar{\Sigma}_t \end{aligned}$$

A. Parameters Used

The Q matrix and the process model was calculated taking the system dynamics into consideration using the following formula:

$$Q = \sigma^2(G.G^T)$$

$$where, G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\Delta t}{m} & 0 & 0 \\ 0 & \frac{\Delta t}{m} & 0 \\ 0 & 0 & \frac{\Delta t}{m} \end{bmatrix}$$

For the data files:

- **Low Noise**: $\sigma = 0.01,\ R = I * 0.5^2$
- **High Noise**: $\sigma = 0.01,\ R = I * 1.0^2$
- **Velocity**: $\sigma = 0.01,\ R = I * 0.5^2$
- **Motion Capture**: $\sigma = 0.001,\ R = I * 0.01^2$

IV. Conclusion

We apply the Kalman filter to real and hybrid data files, evaluating its performance in tracking the drone's motion. By tuning the filter parameters, we improve the accuracy of state estimation despite noisy measurements. Visualization of the drone's three-dimensional trajectory demonstrates the effectiveness of the Kalman filter.

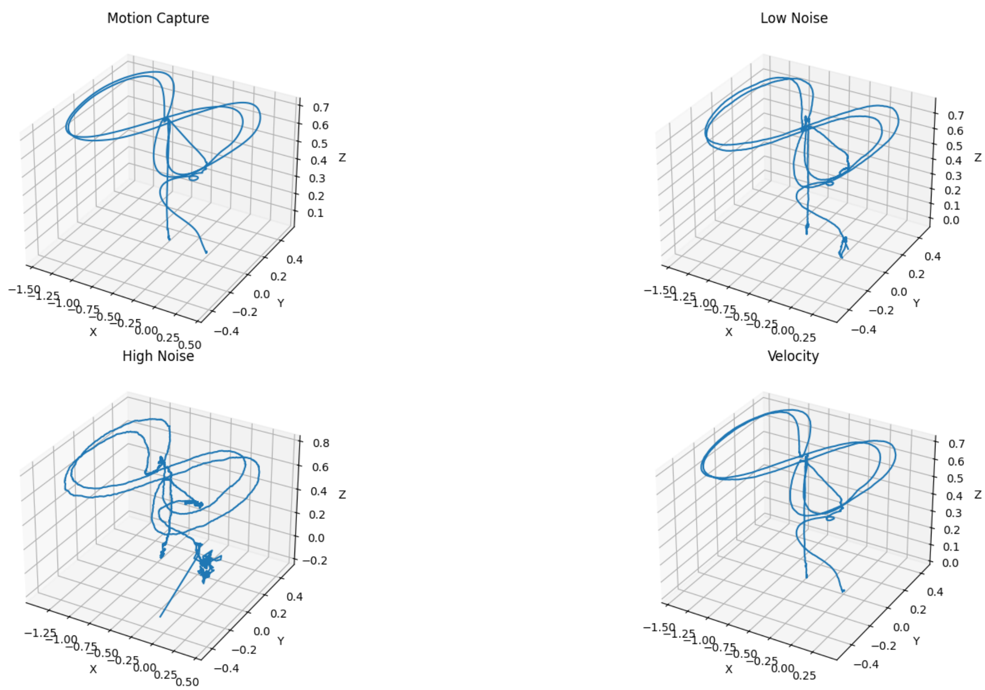

Fig. 1. 3D Trajectory Plots for the drone