State Estimation Lab

Lab 3

I. Introduction

- Implementing Kalman Filter for estimating position of drone in the world coordinate system with:

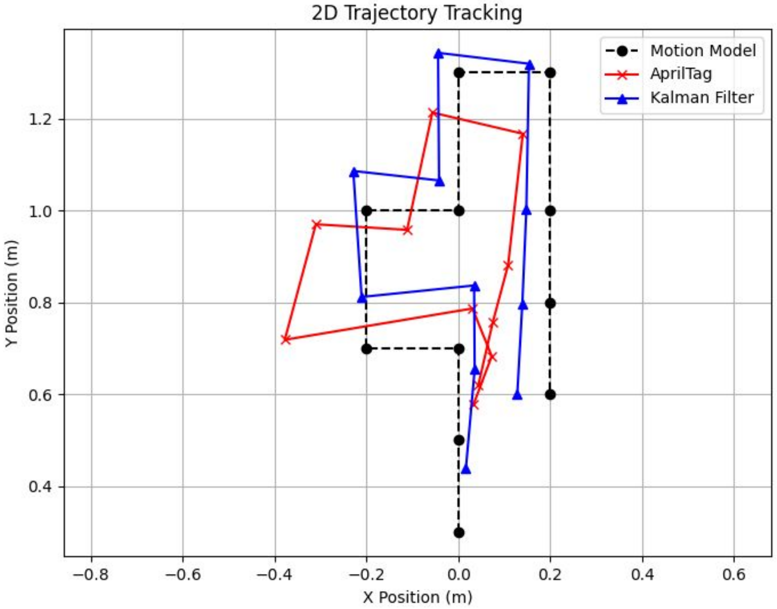
+ State vector $\mu_t = [x\ y\ z]^T$
+ Control input $u_t = [dx\ dy\ dz]^T$ with dx, dy, dz are moving distances.
+ Observation $z_t = [X\ Y\ Z]^T$ obtained from April Tag position and drone camera pose relative to the tag.

II. Coding

- The python code workflow:
  + Send a command to the drone and estimate its pose based on the motion model.
  + Detect April Tag and estimate the drone’s pose.
  + Kalman filter updates the drone’s pose based on these two estimated poses.
- Students are requested to complete ***KalmanFilter class***, enter the variables ***camera_params, tag_size, at_word*** (April Tag pose). (You can design your own movement sequence)

II. Coding

- Demo:



Thank you !!!

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t