CMSC828T Midterm: Extended Kalman Filter

Yash Manian

I. Introduction

This project takes into account the Optical flow velocity computed in P2Ph2, and the Poses computed in P2Ph1, and combines them with IMU readings using an extended Kalman filter to estimate the states. The Kalman filter is a tractable version of the Bayes Filter. Here, an extended Kalman filter is used, which linearizes the system about a previous state, and then proceeds with the standard estimation procedure.

II. The Kalman Filter

The Kalman filter allows the estimation of state vector x and its distribution at time t from the estimate at $t-1$. The Kalman filter makes three basic assumptions:

1) The process model $p(x_t|u_t, x_{t-1})$ is linear with additive Gaussian noise
2) The measurement model $p(z_t|x_t$ is linear with additive Gaussian noise
3) The prior on state x is $p(x_0)$, is normally distributed with a known mean μ_0 and covariance Σ_0.

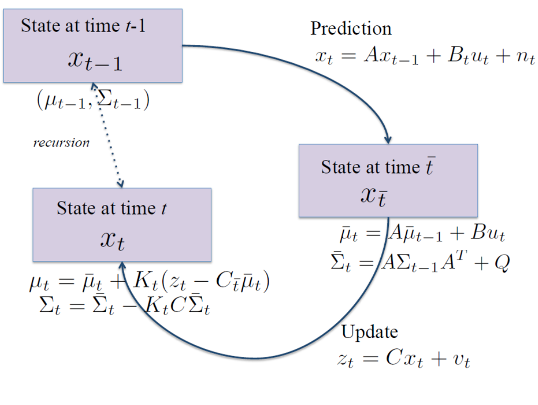

Fig. 1. The Kalman Filter

III. System Equations

A. Process Model

The Kalman Filter uses the state equations to make estimations. However, since there assumed to be a reliable IMU on the quadrotor, the dynamics of the quadrotor can be bypassed, and a process model constructed from the following state vector can be used:

$$x = \begin{bmatrix} p \\ q \\ \dot{p} \\ b_g \\ b_a \end{bmatrix} \tag{1}$$

Where p is the position vector, q is the orientation vector, $\dot{p}$ is the velocity vector,bg is the gyro bias and ba is the accelerometer bias.The process model based on this vector will be as follows:

$$f = \begin{bmatrix} \dot{p} \\ G(q)^{-1}(\omega_m - b_g - n_g) \\ g + R(Q)(a_m - b_a - n_a) \\ n_{bg} \\ n_{ba} \end{bmatrix} \tag{2}$$

Where $G(q)$ transforms the components of angular velocity in the body frame, $R(q)$ is a rotation matrix following the Z-X-Y Euler angle notation. ω_m is the previous measurement of angular velocity from the gyro, and a_m is the previous measurement of acceleration from the accelerometer.n_{ba} and n_{bg} denote changes in the bias over time.

B. Measurement Model

The measurement model determines what states are being measured, and what kind of noise the sensors experience. This model is used to determine the update to the estimates based on the measurements. The measurement matrix C is given as follows:

$$Ct = \begin{bmatrix} I_{3x3} & 0 & 0 & 0 & 0 \\ 0 & I_{3x3} & 0 & 0 & 0 \\ 0 & 0 & I_{3x3} & 0 & 0 \end{bmatrix} \tag{3}$$

The measurement model can then be given by:

$$\begin{bmatrix} x_t \\ z_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ C_t & W_t \end{bmatrix} \begin{bmatrix} x_{\bar{t}} \\ v_t \end{bmatrix} - \begin{bmatrix} 0 \\ h(\bar{\mu}_t, 0) - C_t\bar{\mu}_t \end{bmatrix} \tag{4}$$

IV. Linearizing

Since the Kalman Filter makes the assumption that the system is linear in nature, and the system of equation available is non-linear, a basic Kalman filter would fail here. The Extended Kalman Filter solves this problem by linearizing the process model about the current state, as follows:

$$A_t = \frac{\partial f}{\partial x} \tag{5}$$

$$U_t = \frac{\partial f}{\partial n} \tag{6}$$

Where x is the state, and n is the noise.

$$F_t = I + A_t \delta_t \tag{7}$$

$$V_t = U_t \delta_t \tag{8}$$

For the measurement model,

$$C_t = \frac{\partial h}{\partial x} \tag{9}$$

$$W_t = \frac{\partial f}{\partial v} \tag{10}$$

V. Kalman gain and Estimation

A. Prediction

After linearizing the process function, there are two steps to the estimation, i.e. prediction and update. The prediction takes into account the inputs and the previous state to predict what the next step would be. The noise in the process is characterized by the covariance matrix Q. The prediction then estimates a predicted mean and covariance.

$$\bar{\mu_t} = \mu_{t-1} + f\delta_t \tag{11}$$

$$\bar{\Sigma_t} = F_t \Sigma_{t-1} F_t^T + V_t Q_t V_t^T \tag{12}$$

Here, F_t and V_t are obtained from linearizing the process models.

B. Update

The Update computes the estimated next value based on the estimated mean and covariance from the process model with additive Gaussian noise, and the measured states at time t, with Gaussian measurement noise. We compute a Kalman gain to represent the trade-off between the importance given to the estimation and the measurement. The Kalman gain depends on the Q and R matrices. If Q is larger, then more the measurements will influence the output more as opposed to the predictions and vice-versa. Thus the values for Q and R need to be carefully chosen to estimate with reasonable accuracy. The Kalman gain can be defined as follows:

$$K_t = \bar{\Sigma_t} C_t^T (C_t \bar{\Sigma_t} C_t^T + W_t R_t W_t^T)^{-1} \tag{13}$$

Using the Kalman gain, the estimated mean and covariance can be calculated as follows:

$$\mu_t = \bar{\mu_t} + K_t(Z_t - h) \tag{14}$$

$$\Sigma_t = \bar{\Sigma_t} - K_t C_t \bar{\Sigma_t} \tag{15}$$

VI. Results

The EKF was run for six trajectories, i.e. Square, Mapping, Slow Circle, Fast Circle, Mountain and Straight Line. The states measured were Position, Velocity and orientation.

A. Mapping

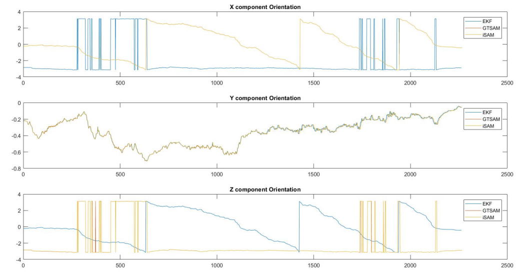

Fig. 2. Orientation estimates for Mapping trajectory

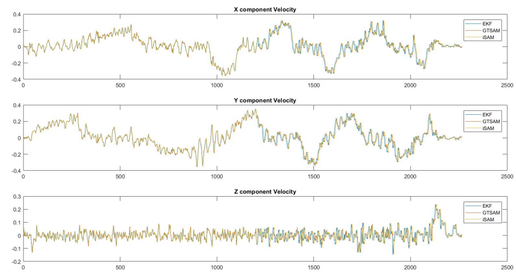

Fig. 3. Velocity estimates for Mapping trajectory

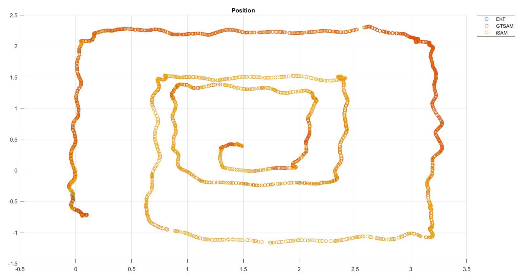

Fig. 4. XY Position estimates for Mapping trajectory

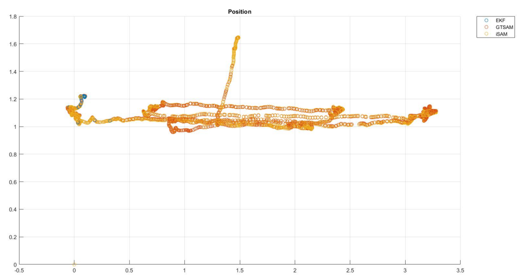

Fig. 5. XZ Position estimates for Mapping trajectory

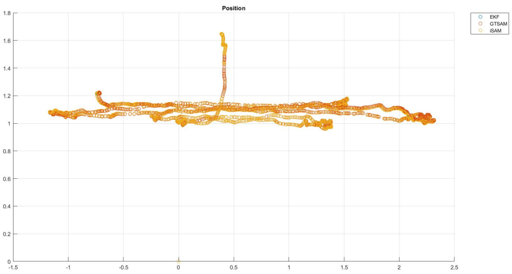

Fig. 6. YZ Position estimates for Mapping trajectory

B. Square

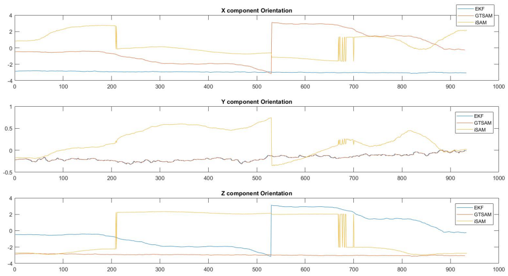

Fig. 7. Orientation estimates for Square trajectory

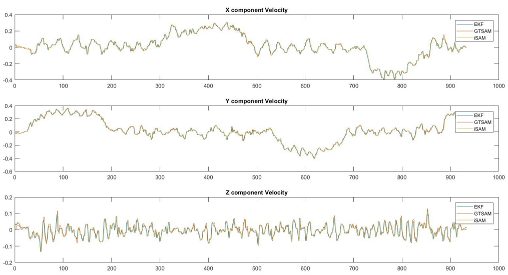

Fig. 8. Velocity estimates for Square trajectory

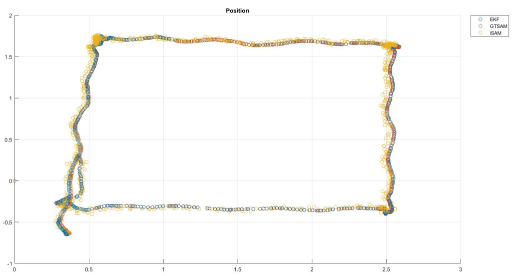

Fig. 9. XY Position estimates for Square trajectory

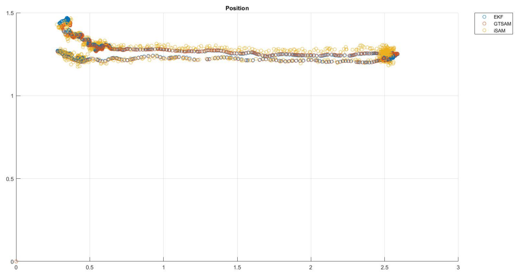

Fig. 10. XZ Position estimates for Square trajectory

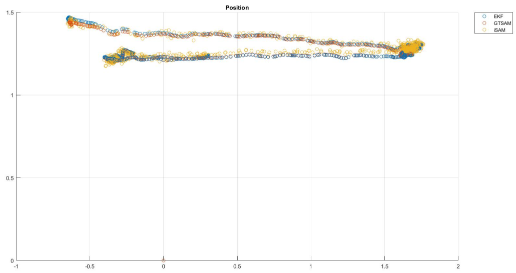

Fig. 11. YZ Position estimates for Square trajectory

C. Mountain

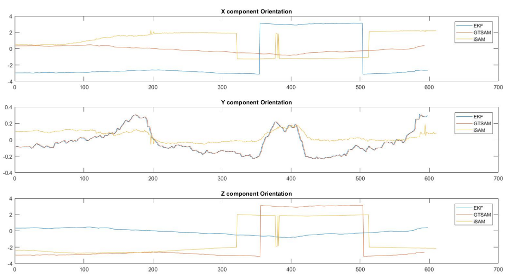

Fig. 12. Orientation estimates for Mountain trajectory

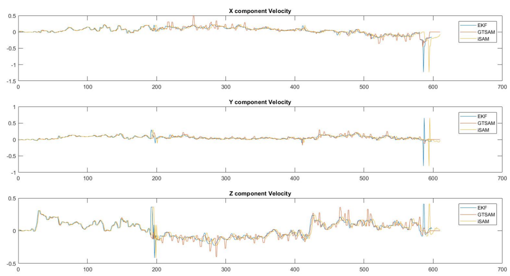

Fig. 13. Velocity estimates for Mountain trajectory

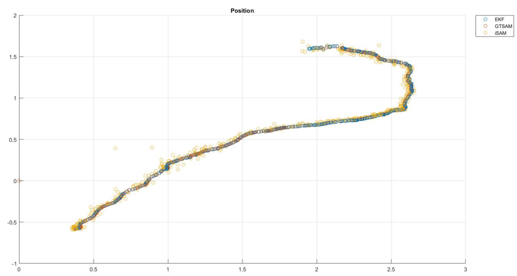

Fig. 14. XY Position estimates for Mountain trajectory

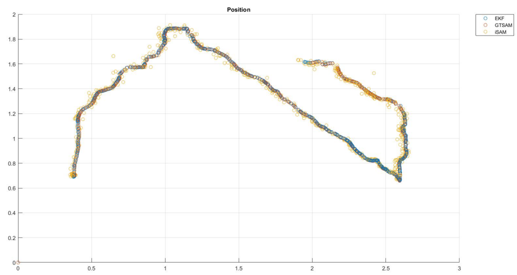

Fig. 15. XZ Position estimates for Mountain trajectory

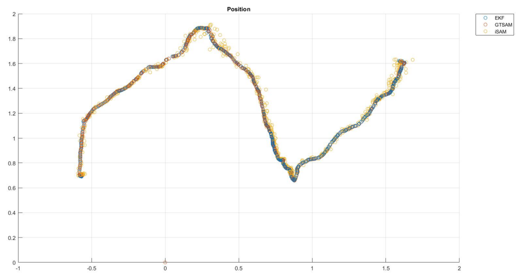

Fig. 16. YZ Position estimates for Mountain trajectory

D. Slow Circle

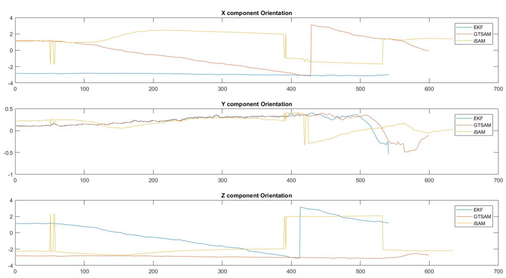

Fig. 17. Orientation estimates for Slow Circle trajectory

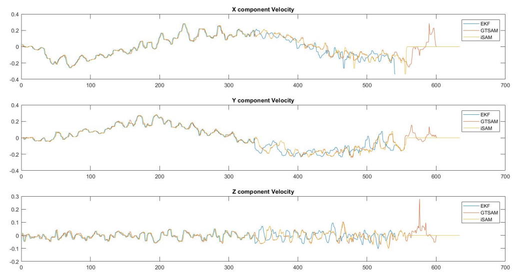

Fig. 18. Velocity estimates for Slow Circle trajectory

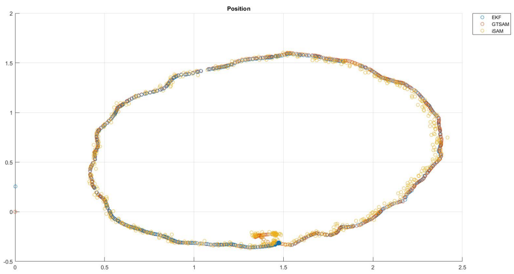

Fig. 19. XY Position estimates for Slow Circle trajectory

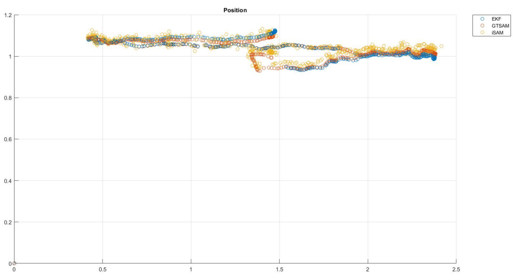

Fig. 20. XZ Position estimates for Slow Circle trajectory

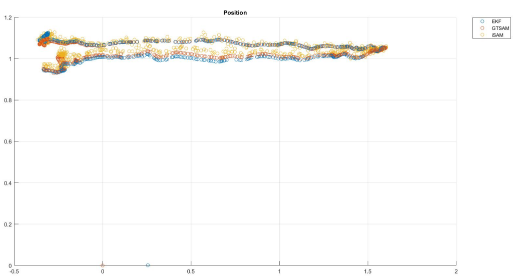

Fig. 21. YZ Position estimates for Slow Circle trajectory

E. Fast Circle

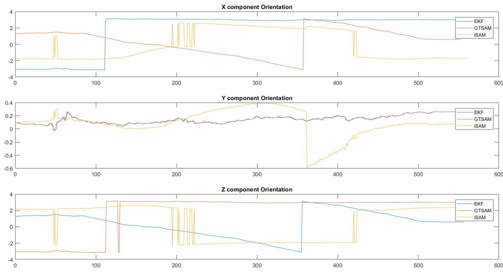

Fig. 22. Orientation estimates for Fast Circle trajectory

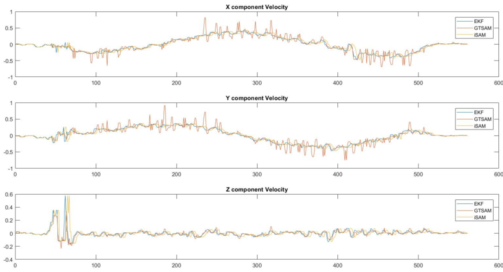

Fig. 23. Velocity estimates for Fast Circle trajectory

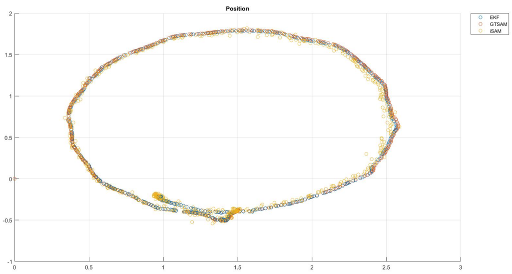

Fig. 24. XY Position estimates for Fast Circle trajectory

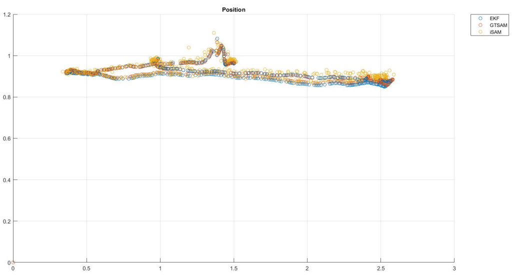

Fig. 25. XZ Position estimates for Fast Circle trajectory

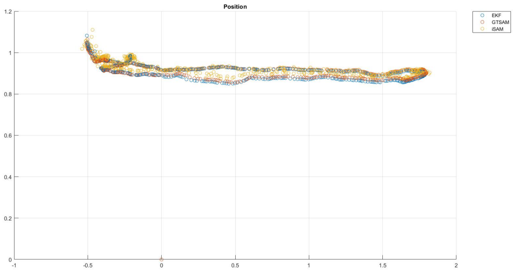

Fig. 26. YZ Position estimates for Fast Circle trajectory

F. Straight Line

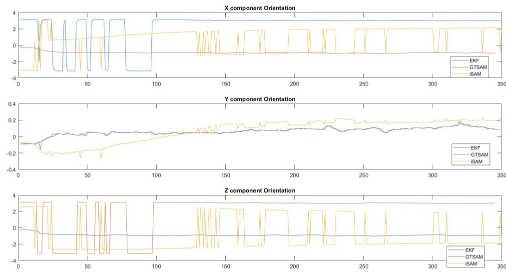

Fig. 27. Orientation estimates for Straight Line trajectory

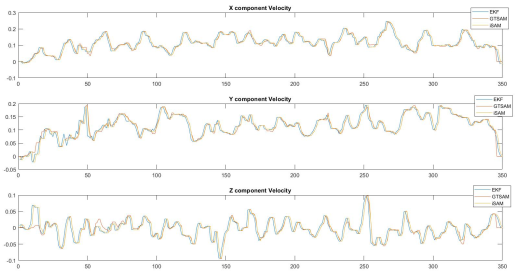

Fig. 28. Velocity estimates for Straight Line trajectory

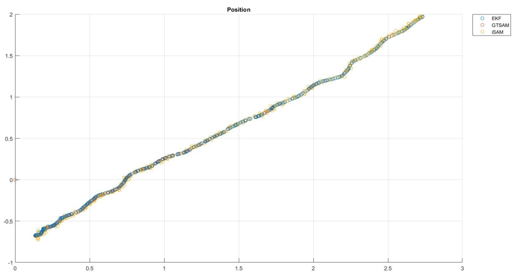

Fig. 29. XY Position estimates for Straight Line trajectory

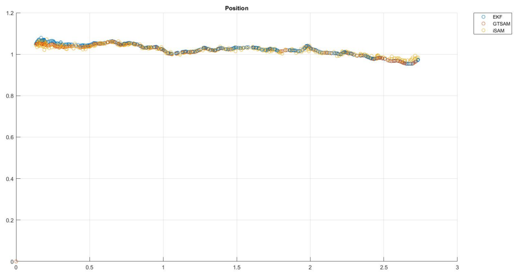

Fig. 30. XZ Position estimates for Straight Line trajectory

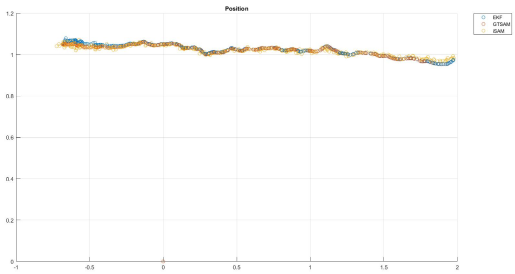

Fig. 31. YZ Position estimates for Straight Line trajectory