State Estimation for a Quadrotor System

Harishankar M

1 Introduction

Quadrotors are complex non-linear systems with multiple inputs and states, making them an ideal test bed for evaluating control and state estimation techniques. Accurate state estimation is crucial for control and navigation. This project aims to implement and compare the performances of the Kalman Filter(KF), Moving Horizon Estimator (MHE), and Unscented Kalman Filter (UKF) for estimating the states of a quadrotor around a predefined operating point.

2 System Modeling

2.1 Non Linear Model

We have utilized the mathematical framework used by Wu et. al in [2]. The structure and major forces acting on the quadrotor are shown in Fig 1.

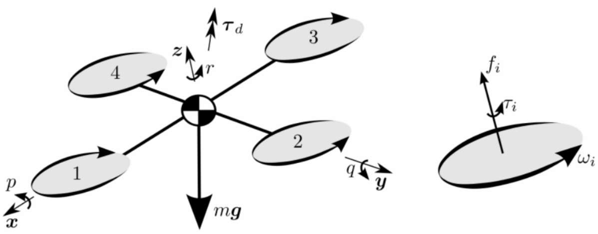

Figure 1: Various forces acting on quadrotor [?]

The mass of the quadrotor is denoted by $m \in R$, and the inertial tensor $I = diag(I_{xx}, I_{yy}, I_{zz}) \in \mathcal{R}^{3\times3}$ is defined in the body frame, which is symmetric. Let vector $\mathbf{p} = [x, y, z]^T \in R^3$ and $\mathbf{v} = [v_x, v_y, v_z]^T \in R^3$ represent the position and velocity respectively along the x, y, and z axes in the inertial frame. The attitude of the system (η) is defined by roll (rotation along x-axis, ϕ), pitch(rotation along y-axis, θ) and yaw (rotation along z-axis), $\eta = [\phi, \theta, \psi]^T \in R^3$.

The angular velocities around the x, y, and z axes are denoted by the elements of $\omega = [p, q, r]^T \in R^3$ in the body frame. The angular velocity vector $\dot{\eta}$ is related to the Euler angular velocity vector ω in the body-fixed frame, and the $\dot{\eta} = \mathbf{R_a}\omega$ can express relation ω, where $\mathbf{R_a}$ is a transformation matrix from the angular velocity around the axis to the Euler angular velocity is given by:

$$\boldsymbol{R_a} = \begin{bmatrix} 1 & \boldsymbol{s_\phi t_\theta} & \boldsymbol{c_\phi t_\theta} \\ 0 & \boldsymbol{c_\phi} & -\boldsymbol{s_\phi} \\ 0 & \boldsymbol{s_\phi / c_\theta} & \boldsymbol{c_\phi / c_\theta} \end{bmatrix}$$

In the equation above, $s_x. = sin(x), c_x. = cos(x), t_x. = tan(x)$ for simplicity. The linear transformation of a vector from the body-fixed frame to the inertial frame is represented by rotation matrix $R \in \mathcal{R}^{3\times3}$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{c_\theta c_\psi} & \boldsymbol{s_\phi s_\theta c_\psi} - \boldsymbol{c_\phi s_\psi} & \boldsymbol{c_\phi s_\theta c_\psi} + \boldsymbol{s_\phi s_\psi} \\ \boldsymbol{c_\theta s_\psi} & \boldsymbol{s_\phi s_\theta s_\psi} + \boldsymbol{c_\phi c_\psi} & \boldsymbol{c_\phi s_\theta s_\psi} - \boldsymbol{s_\phi c_\psi} \\ -\boldsymbol{s_\theta} & \boldsymbol{s_\phi c_\theta} & \boldsymbol{c_\phi c_\theta} \end{bmatrix}.$$

As shown in Fig.1, motors 1 and 3 spin in the clockwise direction with angular speeds ω_1 and ω_3, and motors 2 and 4 spin in a counterclockwise direction with angular speeds ω_2 and ω_4.The translational force on the quadrotor due to the rotors is F_l in the body frame, which is described by $F_l = \sum_{i=1}^{4} F_i$, where F_i is the thrust moment generated by each motor. F_i is calculated as $F_i = k\omega_i^2$ in the positive z direction, where k is the positive lumped constant parameter k denotes the thrust factor of the propeller. The torque on the body frame due to the motor's rotation is modelled as $T_i = b\omega_i^2$. The net torque acting with respect to roll, yaw, and pitch axes are τ_ϕ, τ_θ and τ_ψ respectively and are given by:

$$\boldsymbol{\tau_\eta} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l\,(F_2 - F_4) \\ l\,(F_3 - F_1) \\ T_1 - T_2 + T_3 - T_4 \end{bmatrix}$$

With l being the distance from the motor to the centre of gravity. The relation between the total control inputs $\boldsymbol{U} = [\tau_\phi, \tau_\theta, \tau_\psi, F_l]^T$ and the rotor's speeds $\boldsymbol{W} = [\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2]^T$ is bijective when there is no fault, expressed as

$$\boldsymbol{U} = \boldsymbol{GW}$$

with

$$\boldsymbol{G} = \begin{bmatrix} 0 & kl & 0 & -kl \\ -kl & 0 & kl & 0 \\ b & -b & b & -b \\ k & k & k & k \end{bmatrix}$$

The linear and angular dynamics are modelled as in [**?**]. The state dynamics of the quadrotor considering (drag components are considered negligible).

$$\begin{aligned} \dot{\mathbf{p}} &= v \\ \dot{\mathbf{v}} &= \frac{1}{m}\left(mg\boldsymbol{e}_3 - RF_l\boldsymbol{e}_3\right) \\ \dot{\eta} &= R_a\omega \\ \dot{\omega} &= I^{-1}\left(\tau_\eta - (\omega \times I\omega)\right) \end{aligned}$$

with $e_3 = [0, 0, 1]^T$. The combined states of the systems is denoted by $\mathbf{x} = [p v \eta \omega]^T$ and the motor inputs represented by $\mathbf{u} = \mathbf{W}$, the entire dynamics can be represented as $f(x, u)$ with state disturbance ($\mathbf{w}$) modelled as Gaussian with covariance matrix as Q'. We assume the translational and angular positions ($\mathbf{y}$) are available from the inertial measurement units and GPS with a sensor noise ($\mathbf{v}$) modelled as Gaussian with covariance matrix R'. The system dynamics can be represented as :

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{x} &= C\mathbf{x} + \mathbf{v} \end{aligned}$$

with

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

2.2 Linear Model

For obtaining the linear continuous model of the we linearize the quadrotor about hovering condition i.e $W = [\frac{mg}{4k} \frac{mg}{4k} \frac{mg}{4k} \frac{mg}{4k}]$. Upon substituting this, we get the fixed points of the system as $x^* = [1\ 11\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$. We linearize the system with respect to these fixed points and inputs using the Jacobian of the function and then further discretize it. The discrete form of the system is as given by :

The linearised state-space representation is:

$$x[k+1] = Ax[k] + Bu[k] + w, \quad y[k] = Cx[k] + v,$$

Where w and v are process and measurement noise, respectively, with covariance matrices Q and R respectively.

3 State Estimators

3.1 Kalman Filter (KF)

The KF assumes the system is linear and state disturbance and measurement noise to be Gaussian. The states are updated using the following algorithm

1. **Prediction:**

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k, \quad P_{k|k-1} = AP_{k-1|k-1}A^\top + Q.$$

2. **Update:**

$$\begin{aligned} L_k &= P_{k|k-1}C^\top(CP_{k|k-1}C^\top + R)^{-1}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k(y_k - C\hat{x}_{k|k-1}), \\ P_{k|k} &= (I - L_kC)P_{k|k-1}. \end{aligned}$$

3.2 Moving Horizon Estimator (MHE)

The MHE uses a window of past measurements to estimate states. The cost function which is minimised is :

$$J = \sum_{k-N+1}^{k} (x[k+1]-Ax[k]+Bu[k])W_{vn}(x_k-Ax[k]+Bu[k])^T+(y[k]-Cx[k])W_{dn}(y[k]-Cx[k])^T.$$

with W_{dn}, W_{vn} being weighting matrices, and N being the length of the horizon. Upon finding the optimal value of $x[k-N+1]$ and the state disturbances $w[k-n+1], w[k-n+2]...w[k-1]$, we use it to find $x[k]$ (estimate of current state of the quadrotor).

3.3 Unscented Kalman Filter (UKF)

The UKF is suitable for highly nonlinear systems. It uses a deterministic sampling technique to estimate the states and the error covariances. The Unscented Kalman Filter (UKF) differs from the standard Kalman Filter (KF) primarily in how it handles nonlinearities. The UKF avoids explicit linearization by using the unscented transform, a method that approximates the propagation of mean and covariance through a nonlinear function. It generates a set of sigma points that are deterministically chosen to capture the mean and covariance of the state distribution. These points are then propagated through the system nonlinear dynamics, allowing the UKF to achieve a higher degree of accuracy in estimating the state distribution . The formulation is as follows:([1])

$$\chi_0 = \hat{x}_{k-1},$$

$$\chi_i = \hat{x}_{k-1} + \sqrt{(n+\lambda)P_{k-1}}, \quad i = 1, \ldots, n,$$
$$\chi_{i+L} = \hat{x}_{k-1} - \sqrt{(n+\lambda)P_{k-1}}, \quad i = 1, \ldots, n,$$

where:

- n is the dimension of $x[k]$
- $\hat{x}_{k-1}$ is the estimate of previous state,
- P_{k-1} is the error covariance matrix from previous time step ,
- L is the state dimension,
- λ is a scaling parameter determined by *alpha* and β, $\lambda = \alpha^2(n+\kappa) - n$,

Each sigma point is propagated through the nonlinear system model:

$$\chi_i^* = f(\chi_i, u_k), \quad i = 0, \ldots, 2n,$$

where f_d is the discretised non-linear state evolution of the system,$x[k+1] = f_d(x[k], u[k]) + w[k]$, with $w[k]$ being state noise with covariance Q (we use Euler approximation).

The predicted state states are given by:

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} W_i^{(m)} \chi_i^*,$$

where $W_i^{(m)}$ are the weights for the states.

The predicted state covariance is:

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^{(c)} (\chi_i^* - \hat{x}_{k|k-1})(\chi_i^* - \hat{x}_{k|k-1})^\top + Q,$$

where $W_i^{(c)}$ are the weights for the covariance.

$$W_0^{(m)} = \frac{\lambda}{n+\lambda},$$

$$W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta),$$

where β is a parameter that incorporates prior distribution knowledge. For Gaussian distributions, we take $\beta = 2$.

For $i = 1, \ldots, 2n$, the weights are:

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)}.$$

Sigma points are passed through the measurement model:

$$y_i = C\chi_i^*, \quad i = 0, \ldots, 2L,$$

The predicted measurement mean is:

$$\hat{y}_k = \sum_{i=0}^{2n} W_i^{(m)} y_i.$$

The predicted measurement covariance is:

$$P_{yy} = \sum_{i=0}^{2n} W_i^{(c)} (y_i - \hat{y}_k)(y_i - \hat{y}_k)^\top + R,$$

The cross-covariance between the state and the measurement is:

$$P_{xy} = \sum_{i=0}^{2n} W_i^{(c)} (\chi_i^* - \hat{x}_{k|k-1})(\gamma_i - \hat{z}_k)^\top.$$

The Kalman gain is computed as:

$$K_k = P_{xy} P_{yy}^{-1}.$$

The state estimate is updated using:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k),$$

where y_k is the actual measurement.

The state covariance can be updated in multiple ways. We use a specific form called Josephs's form of update to ensure numerical stability.

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^\top + K_k R K_k^\top,$$

4 Simulation Results

4.1 Setup

The simulation is conducted using Matlab. We used a time step of 0.05 for numerical integration and simulated the system for 12s. The system parameters used are

Parameter	Description	Value
b	Torque coefficient	2.92×10^{-9}
L	Arm length (m)	0.222
k	Motor thrust coefficient	1.49×10^{-7}
m	Mass of the quadrotor (kg)	1.023
k_d	Drag coefficient	0.1
g	Gravitational acceleration (m/s^2)	9.81
I_{xx}	Moment of inertia about x-axis (kg·m^2)	9.5×10^{-3}
I_{yy}	Moment of inertia about y-axis (kg·m^2)	9.5×10^{-3}
I_{zz}	Moment of inertia about z-axis (kg·m^2)	18.6×10^{-3}

Table 1: Updated Quadrotor System Parameters([2])

4.2 Results

The input to the system, in terms of motor rpm is shown in Fig2

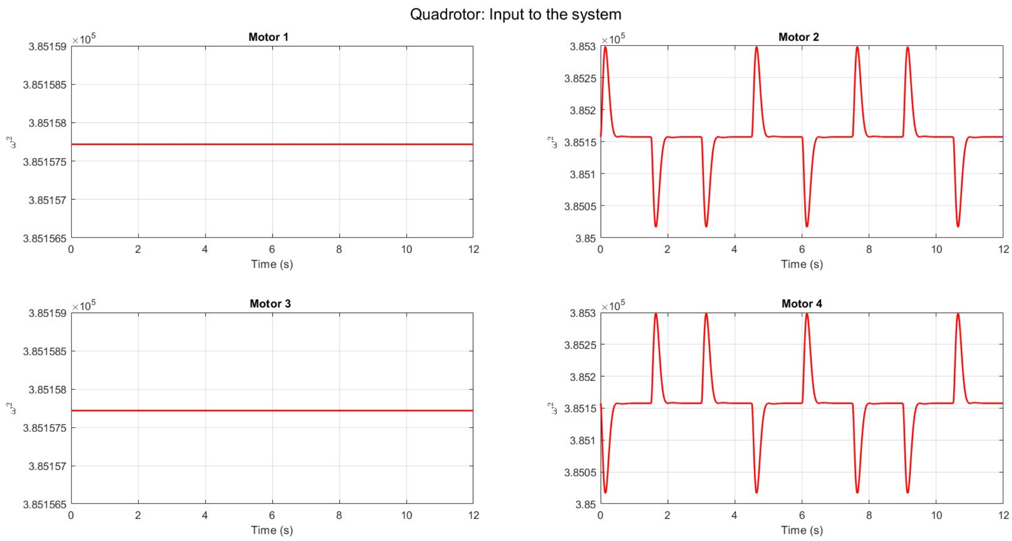

Figure 2: Motor RPM

Fig 3 shows the expected output from the system.

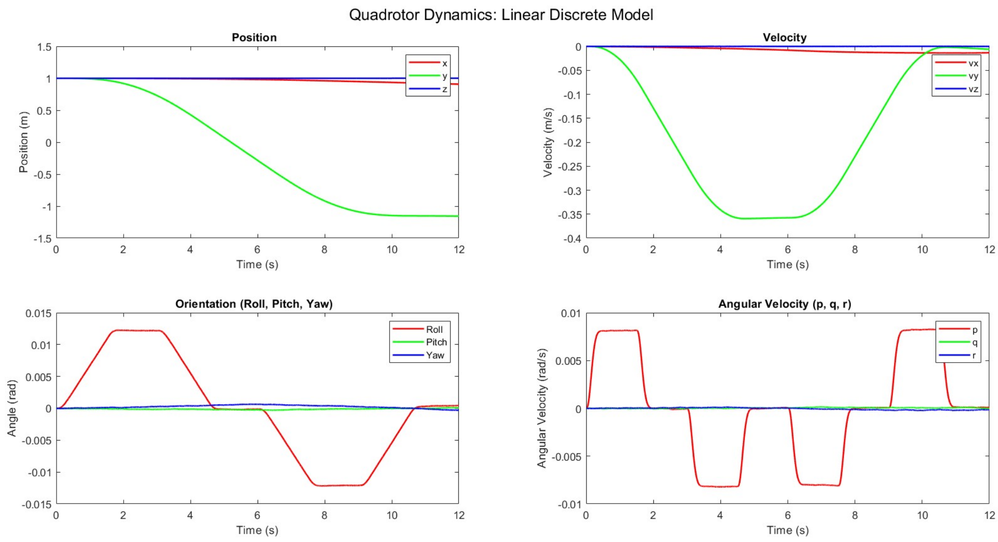

Figure 3: States of the system

4.3 Kalman Filter

Fig4 shows the output from the Kalman filter and Fig 5 shows the error graph.

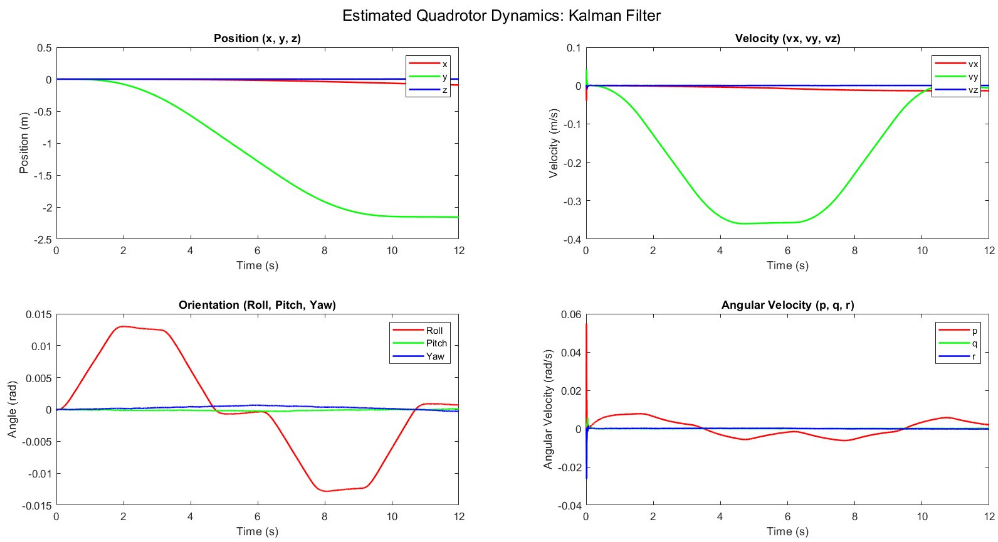

Figure 4: Estimates from Kalman Filter

4.4 Moving Horizon Estimator

Fig8 shows the output from the Kalman filter and Fig 9 shows the error graph.

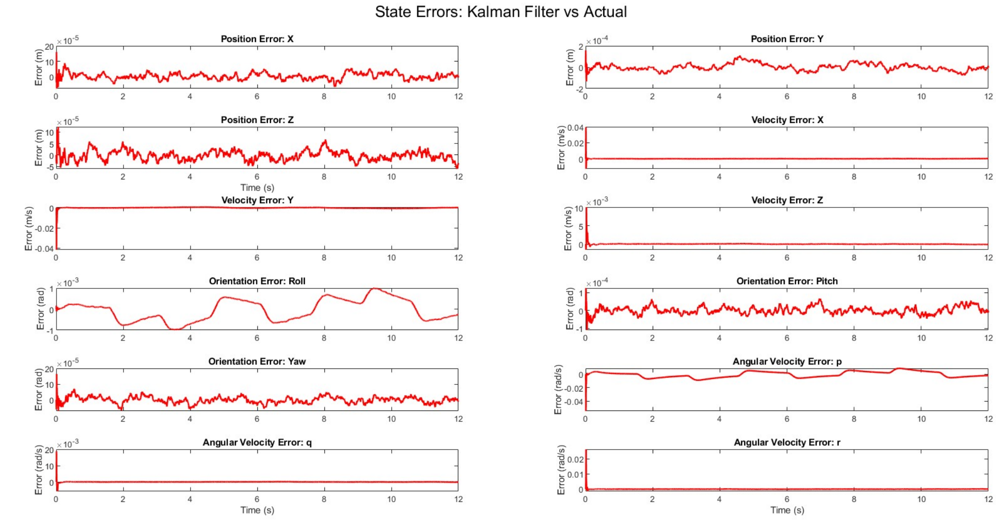

Figure 5: Error in Estimates of Kalman Filter

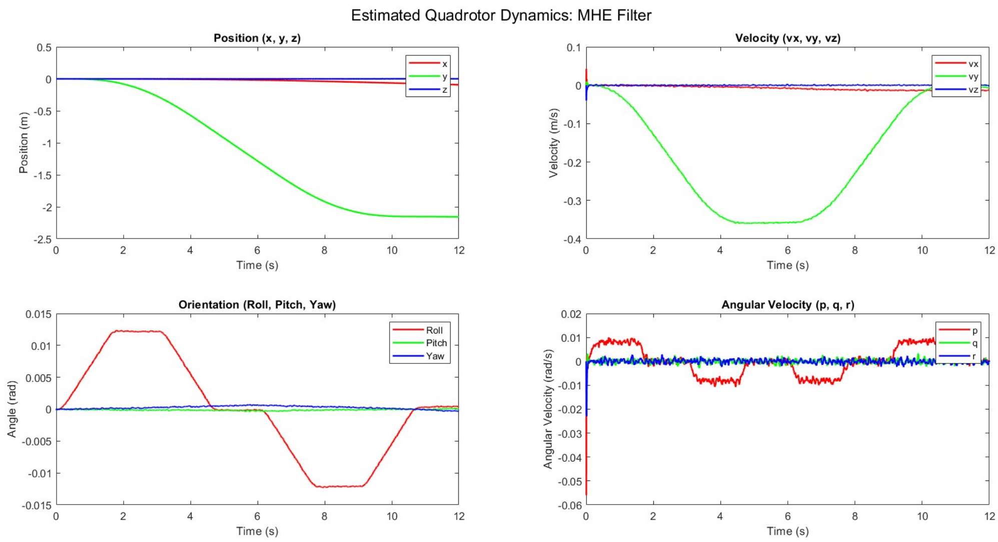

Figure 6: Estimates from MHE,window size = 20

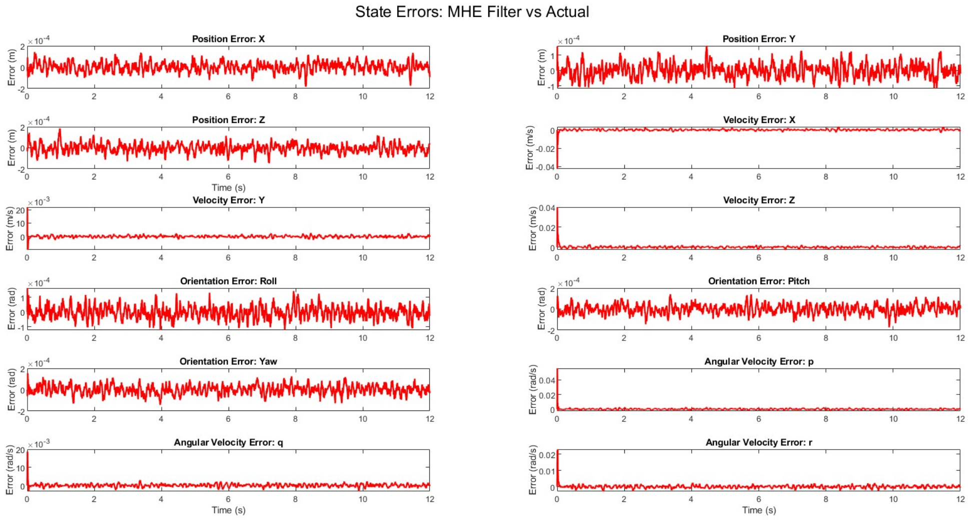

Figure 7: Error in Estimates of MHE,window size = 20

4.5 Unscented Kalman Filter

Fig8 shows the output from the Kalman filter and Fig 9 shows the error graph.

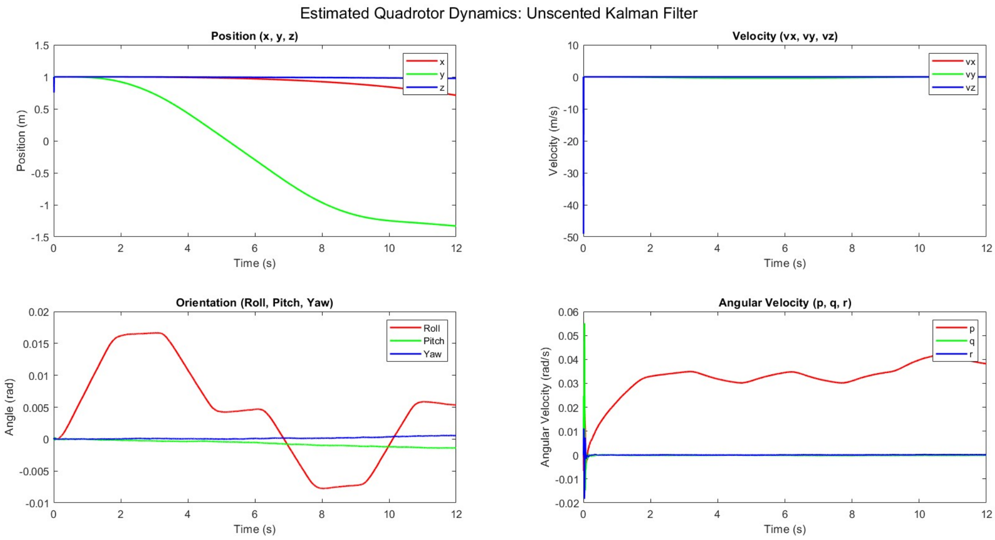

Figure 8: Estimates from UKF

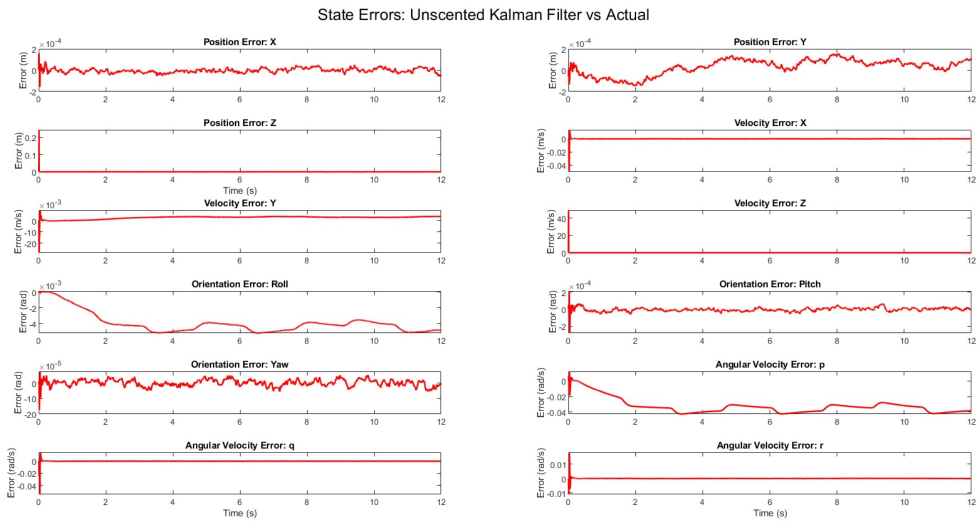

Figure 9: Error in Estimates of UKF

4.6 Computational Time

The time taken for complete state estimation of 12s with time step of 0.05s for

- Kalman Filter : 0.0626s
- MHE : 7.5422s
- Unscented Kalman Filter :1.134s

5 Inference and Conclusion

It can be observed that the MHE estimator is computationally more complex and time-consuming compared to the other two and, hence, may not be as ideal as the Kalman filter and UKF for real-time state estimation. The Kalman filter has the least computational cost since all the underlying operations are linear but may fail in heavily non-linear models. The UKF comes at a comparatively higher computational cost compared to the Kalman filter but is more reliable for highly non-linear models.

References

[1] H Bonyan Khamseh, S Ghorbani, and F Janabi-Sharifi. Unscented kalman filter state estimation for manipulating unmanned aerial vehicles. *Aerospace Science and Technology*, 92:446–463, 2019.

[2] Yuhu Wu, Kaijian Hu, Xi-Ming Sun, and Yanhua Ma. Nonlinear control of quadrotor for fault tolerance: A total failure of one actuator. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(5):2810–2820, 2021.