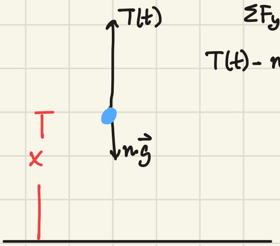
Probabilistic Robotics Hw #3

1.) Drone mass, m

Sampling Rate, Δt — Modeling the drone altitude as $x(t)$



$\Sigma F_y = ma$

$T(t) - m\vec{g} = m\vec{a}$

$\dot{x} = \vec{v}$

$\ddot{x} = \vec{a}$

State Vector

$x = \begin{bmatrix} x \\ v \end{bmatrix}$ $\quad u = T(t)$

Control input

Continuous time dynamics

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$T(t) - mg = m\ddot{x}(t)$

$\ddot{x}(t) = \frac{1}{m} T(t) - \vec{g}$

$$\dot{x}(t) = A \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + B\,u(t)$$

$$\begin{bmatrix} v(t) \\ \ddot{x}(t) \end{bmatrix} = \underset{\text{2x2 matrix}}{A} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + B\,u(t) + d_c$$

2x1 since $u(t) = T(t)$ only

d_c: modeling gravity

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) + d_c$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \quad d_c = \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

Discretize the continuous time dynamics

$$x_{k+1} = F(x_k)$$

$$x_k = x(k\Delta t)$$

$$x_{k+1} = x_k + \int_{k\Delta t}^{(k+1)\Delta t} f(x(\tau))\, d\tau$$

Discrete A Matrix

↳ $A_k = e^{A_c \Delta t}$

$$= I + A_c \Delta t = I + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Delta t$$

$$= I + \begin{bmatrix} 0 & \Delta t \\ 0 & 0 \end{bmatrix}$$

$$A_c = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

constant term

For a dynamical system of the form $\dot{x} = Ax + Bu + d_c$

$$x(t+\Delta t) = e^{A_c \Delta t} x_k + \int_0^{\Delta t} e^{A_c \tau} B_c u_k \, d\tau + \int_0^{\Delta t} e^{A_c \tau} d_c \, d\tau$$

LAST TERM

Taylor Expansion of exp (Matrix) → $e^{A} \overset{\text{def}}{=} I + A + \frac{A^2}{2!} + \frac{A^3}{6!} + \dots \frac{1}{k!} A^k + \dots$

$$e^{A_c \tau} = \left[I + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tau \right] B_c u_k$$

$$= \int_0^{\Delta t} \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} B_c u_k \, d\tau = \int_0^{\Delta t} \underset{2x2}{\begin{bmatrix} 0 & \tau \\ 0 & 1 \end{bmatrix}} \underset{2x1}{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}} d\tau = \int_0^{\Delta t} \begin{bmatrix} \frac{\tau}{m} \\ \frac{1}{m} \end{bmatrix} d\tau \rightarrow \begin{bmatrix} \frac{\tau^2}{2m} \\ \frac{\tau}{m} \end{bmatrix}_0^{\Delta t}$$

$$B_k = \begin{bmatrix} \frac{\Delta t^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix}$$

LAST TERM

$$e^{A_c\tau} d_c \Rightarrow \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ -g \end{bmatrix} = \begin{bmatrix} -g\tau \\ -g \end{bmatrix}$$

$$G = \int_0^{\Delta t} \begin{bmatrix} -g\tau \\ -g \end{bmatrix} d\tau = \begin{bmatrix} -\frac{1}{2} g \Delta t^2 \\ -g\Delta t \end{bmatrix}$$

Final Discrete time system:

$$x_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{\Delta t^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} u_k + \begin{bmatrix} -\frac{1}{2} g \Delta t^2 \\ -g\Delta t \end{bmatrix}$$

Furthermore, we can model gravity vector into the control input itself since $\vec{g}$ is fixed.

$$x_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{\Delta t^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} u_k$$

where

$u_k = T_k - mg$

T_k: Thrust @ k step

This is the form

$$x_{k+1} = A_d x_k + B_d u_k = A_d x_k + B_d (T_k - mg)$$

State Noise Covariance (Σ_x)

Process Noise Covariance (Q)

Measurement noise covariance (Σ_z)

Also noted as R in my code

input noise covariance (Σ_u)

$$\Sigma_z = \sigma_h^2 \quad \text{Scalar } (1\times1)$$

Input noise covariance

$$u_k = T_k + v_k \qquad v_k \sim N(0, \Sigma_u)$$

$$\Sigma_u = \sigma_T^2 \quad (\text{Scalar})$$

Process Noise

$$w_k = B v_k \qquad = E[w_k w_k^T]$$

expectation of input noise sq

$$Q = B \Sigma_u B^T$$

$$Q = \sigma_T^2 \begin{bmatrix} \left(\frac{\Delta t^2}{2m}\right)^2 & \frac{\Delta t^2}{2m}\frac{\Delta t}{m} \\ \frac{\Delta t^2}{2m}\frac{\Delta t}{m} & \left(\frac{\Delta t}{m}\right)^2 \end{bmatrix}$$

$$Q = \sigma_T^2 \begin{bmatrix} \frac{\Delta t^4}{4m^2} & \frac{\Delta t^3}{2m^2} \\ \frac{\Delta t^3}{2m^2} & \frac{\Delta t^2}{m^2} \end{bmatrix}$$

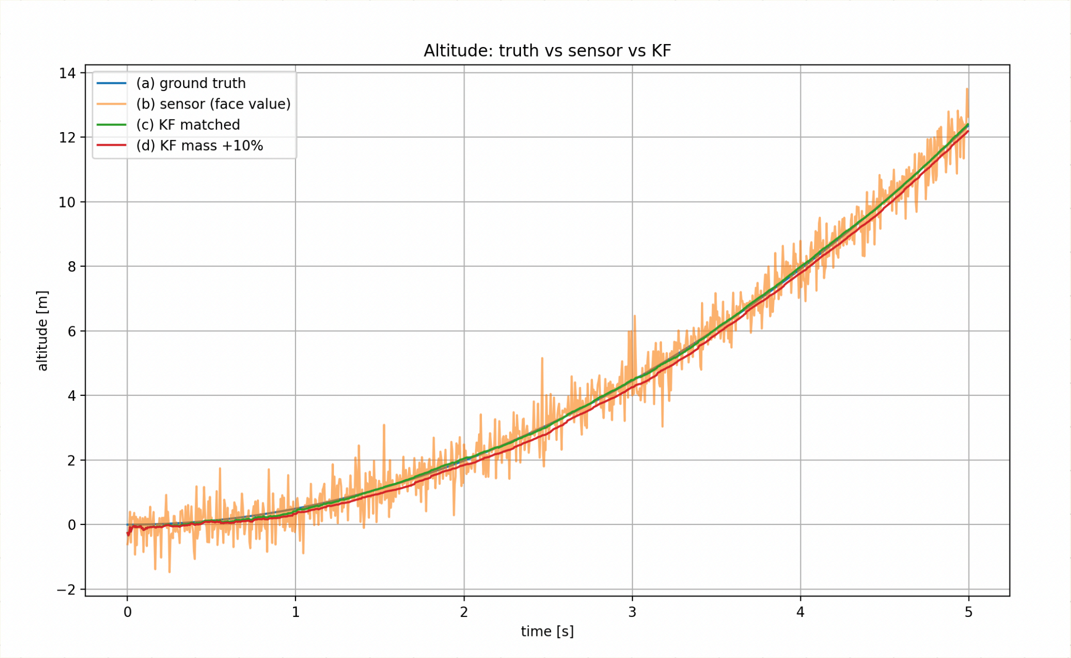
The State Covariance (P_k)

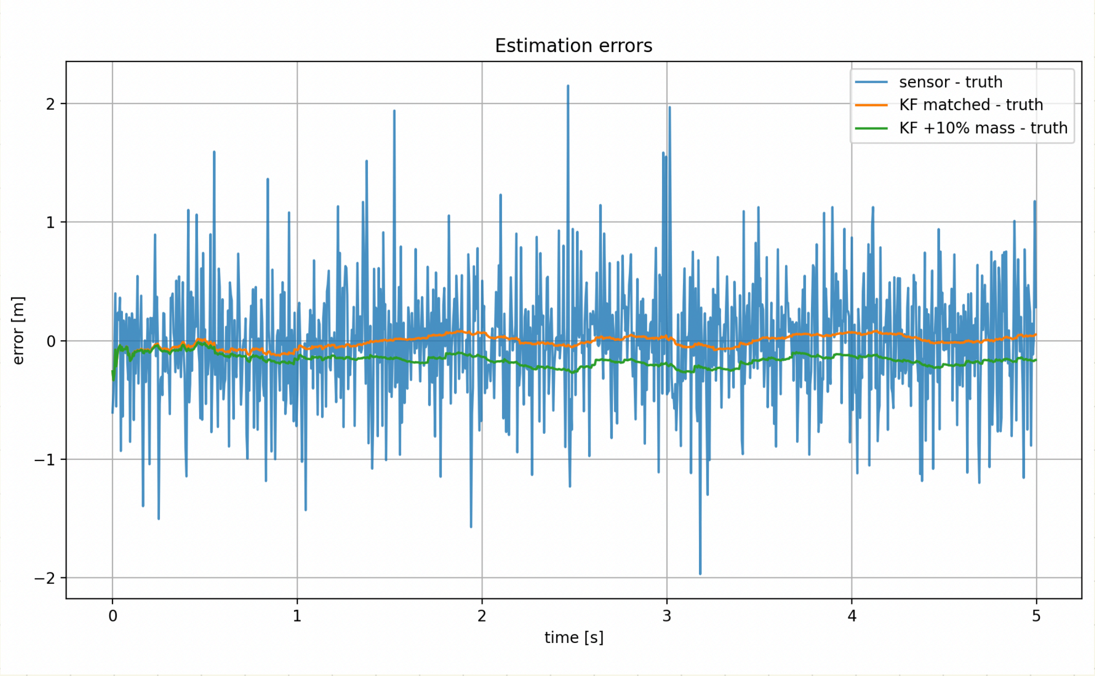
P_k based on the state vector which is 2×1 hence $P_k \subset \mathbb{R}^{2\times2}$

2×2 Matrix

2.)

Plots:


Altitude: truth vs sensor vs KF
(a) ground truth
(b) sensor (face value)
(c) KF matched
(d) KF mass +10%
altitude [m]
time [s]
14
12
10
8
6
4
2
0
−2
0
1
2
3
4
5


Estimation errors
sensor - truth
KF matched - truth
KF +10% mass - truth
error [m]
time [s]
2
1
0
−1
−2
0
1
2
3
4
5

The estimator thinks that the drone is heavier (0.275 kg). This makes it "believe" that the same thrust would produces less acceleration.

- In the plot, the red curve (corresponding to this +10%) slightly lags behind the ground truth.
- The mismatch is evident as a small but systematic bias where the filter is underestimating the altitude.

ERROR PLOT:

You can see a consistent offset (negative bias) in the red trace compared to the truth.

The +10% mass mismatch does Not destabalize the filter (it still does reduce noise) but brings in a systematic error. Instead of being centered at 0, the estimation error has a bias. The filter cant fix this by itself because the prediction model is wrong. This illustrates that the KF is dependent on its model. With a model mismatch, the filter smooths the noise but converges to the wrong trajectory.