Homework 3 - Kalman Filter

Darren Biskup

2025-09-21

1. (20pts) Consider a drone taking off vertically by producing a constant thrust force with its propellers. The drone has an infrared sensor that shoots downwards and measures the altitude. The drone mass (m) and sampling rate (Δt) are parameters known to the system.

Assume that the drone can control the thrust directly (this is a simplifying assumption, usually the model is more complex and includes the motor dynamics and the propeller model), but the signal is noisy. The input noise is Gaussian with standard deviation σ_T. The altitude measurement coming from the infrared sensor is also a Gaussian process with standard deviation σ_h.

Write the discrete-time model in canonical form that is to be used to implement the Kalman filter that estimates the drone altitude. Specify the A, B, C matrices of the model and covariance matrices of the input (Σ_u) and measurement (Σ_z). What is the dimension of state covariance? Show all derivations starting from first principles of laws of motion.

First start by writing the equations of motion of the 1-DOF system. From Newton's laws:

$$m\ddot{h} = u - mg + \text{noise}$$

we are also told that our measurement of the altitude is known to contain some noise ϵ_h

$$z = h + \epsilon_h$$

The Kalman Filter algorithm expects state transition dynamics to be *Linear Gaussian*, and right now due to the gravity term the dynamics equations can't direclty be transcribed to this form. Therefore, if we instead define our state variable as

$$x = \begin{bmatrix} h & \dot{h} & 1 \end{bmatrix}^\top$$

then the dynamics can then be expressed in continuous time as follows:

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

expanded:

$$\dot{x} = \begin{bmatrix} \dot{h} \\ \ddot{h} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -g \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h \\ \dot{h} \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \\ 0 \end{bmatrix} u + \epsilon_t$$

$$\epsilon_t \sim \mathcal{N}(0, \Sigma_u), \quad \Sigma_u = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and the observation/measurement is updated from the state as:

$$z = Cx + \epsilon_h = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h \\ \dot{h} \\ 1 \end{bmatrix} + \epsilon_h$$

$$\epsilon_h \sim \mathcal{N}(0, \Sigma_z), \quad \Sigma_z = \begin{bmatrix} \sigma_z \end{bmatrix}$$

To convert to a discrete time model suitable for usage in an online recursive Kalman Filter implementation, we can apply a first order Euler discretization as follows:

$$A_d = I + \Delta t \cdot A$$
$$B_d = \Delta t \cdot B$$

then:

$$x_{t+1} = A_d x_{t-1} + B_d u_t + \epsilon_t$$

$$A_d = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & -g\Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ \frac{\Delta t}{m} \\ 0 \end{bmatrix}$$

$$\epsilon_t \sim \mathcal{N}(0, \Sigma_u), \quad \Sigma_u = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The measurement model matrices remain unchanged when discretized:

$$z = Cx + \epsilon_h = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h \\ \dot{h} \\ 1 \end{bmatrix} + \epsilon_h$$

$$\epsilon_h \sim \mathcal{N}(0, \Sigma_z), \quad \Sigma_z = \begin{bmatrix} \sigma_z \end{bmatrix}$$

(Abuse of notation: reused the symbols ϵ_t, Σ_u, ϵ_h, and Σ_z to represent both the continuous-time and discrete-time noise terms and their covariances, which can change under discretization.)

The state covariance matrix Σ_u in this case is $\mathbb{R}^{3\times3}$ due to the augmentation of the state matrix, however the nonzero upper left portion representing the covariance matrix of the "relevant" states is $\mathbb{R}^{2\times2}$

2. (30pts) Using the cartesian robot code from class as the reference, modify the program to simulate the drone liftoff and use the Kalman filter to estimate the altitude. Assume the following parameters:

- Drone mass is 0.25kg
- Sampling rate is 200Hz
- Propellers are producing a constant thrust of 2.7N with variance of 0.25N².
- Measurement sample uncertainty is Gaussian but the variance value changes for each sample with uniform distribution between 0.01m² and 0.5m². Assume that the sensor correctly reports the variance.
- Only the first 5 seconds of simulation are of interest.

Simulate the following cases:

- Altitude is obtained by "stealing" the ground truth from the simulator.
- Altitude is taken at "face value" from the sensor without the Kalman filter.
- Altitude is estimated by the Kalman filter and the models are matched.
- Altitude is estimated by the Kalman filter, but the estimator has the mass parameter incorrectly set to 10% higher than the actual mass.

Plot the altitude for all four cases above (four plots on one figure, use different colors for each signal, do not use large markers that obscure the readability of the plot). Also plot the difference (error) between the ground truth and the estimate for the three cases above (three plots on one figure). Your plots must either include the legend or you must list which signal is which in your submission. Comment on the effect of the model mismatch. How does it impact the filter output?