# HW 2: Kalman Filters (Due Sep 29th 11:59pm)

1. Answer each of the following questions.

   Submit your answers as `studentid_lastname_hw2.pdf` in the root of the repository.

   (a) Do robot actions always increase uncertainty? Explain your answer in 2-3 sentences.

   (b) What happens if at any point in Bayesian filtering the probability of a state assignment becomes 1? What are ways to avoid that? Explain your answer in 2-3 sentences.

   (c) If an earthquake occurs, or there is a burglary, the alarm is likely to go off. If the alarm goes off, a police may arrive. Design a Bayesian network illustrating the causal relationships.

   (d) In the recursive estimation case, what if the controls were dependent on observations? Visualize a Bayesian network showing this dependence.

   (e) Why do Extended Kalman Filters (EKFs) fail in handling multiple hypotheses? Explain your answer in 2-3 sentences.

2. We want to track the position of an object of unknown dynamics. We assume that the object moves in one dimension. A common model for unkown dynamics is the *constant jerk* model, that is assume that the acceleration is linear. The constant jerk model can be represented as follows: $\mathbf{x}(t) = [p_t, v_t, a_t, j_t]^T$, with the elements being position, velocity, acceleration and jerk. Assuming discrete time-steps of $\Delta_t = 0.1$, the dynamics based on that model are:

$$\mathbf{x}(t+1) = A\mathbf{x}(t)$$

$$A = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 1 & 0.1 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We additionally assume that we can measure the object's position with some noise:

$$\mathbf{z}(t) = C\mathbf{x}(t) + \mathbf{v}(t) \tag{1}$$
$$C = [1\ 0\ 0\ 0] \tag{2}$$

$\mathbf{v}(t)$ is a zero-mean Gaussian sensor noise with variance $Q = 1.0$.

$\boldsymbol{\mu}_0$ and $\Sigma_0$ represent the initial belief state of the Kalman Filter. We have that:

$$\boldsymbol{\mu}_0 = (5, 1, 0, 0), \qquad \Sigma_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

The *true* position of the object changes as follows:

$$\mathbf{p}(t) = \sin(0.1 * t)$$

with $\mathbf{p}(0) = \mathbf{0}$. You must generate your own noisy sensor data $\tilde{\mathbf{p}}(t)$ by adding zero mean Gaussian noise, $\mathbf{v_m}(t)$, with variance $Q = 1.0$.

$$\tilde{\mathbf{p}}(t) = \mathbf{p}(t) + \mathbf{v_m}(t)$$

(a) Implement a Kalman Filter for $T = 100$ timesteps and plot how the error evolves over time. Save your figure as `problem2a_kf_estimation.png`. Compute the Mean Squared Error (MSE) of the position of the object, averaged over $N = 10000$ trials. Save your figure as `problem2a_kf_mse.png`.

(b) To deal with model uncertainty, a technique frequently used is adding "fictitious" process noise. That is, assume that there is noise in the state dynamics. Rewrite your system dynamics as:

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + \mathbf{w}(t)$$

Add fictitious noise in the form of $\mathbf{w}(t)$ in the dynamics, with zero mean and covariance[1]:

$$R = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

Run the KF again and compare the MSE error with the absence of fictitious noise. Save your figure as `problem2b_kf_mse.png`.

[1]Note that the discretized covariance matrix of the process noise has typically also non-diagonal elements, but we assume a diagonal matrix for simplicity

3. Consider the following scalar system:

$$\begin{aligned} x(t+1) &= \alpha x(t) + w(t) \\ z(t) &= \sqrt{x(t)^2 + 1} + v(t) \end{aligned}$$

$w(t)$ is zero-mean Gaussian process noise with variance $R$. $v(t)$ is zero-mean Gaussian sensor noise with variance $Q$.

   (a) Write the equations for the Kalman filter to estimate the unknown constant $\alpha$ given $z(t)$.
   *Hint:* The state should be augmented with the unknown parameter $\alpha$.

   (b) Using $Q = 1, R = 0.5$, write a script to test the algorithm in Python. Let *true* $x(0) = 2, \alpha = 0.1$. Assume initial estimates as follows, where $\hat{\alpha}$ is the initial estimate of $\alpha$:

$$\begin{aligned} \mu_0 = 1 \text{ and } E[(x(0) - \mu_0)(x(0) - \mu_0)] = 2 \\ \hat{\alpha} = 2 \text{ and } E[(x(0) - \hat{\alpha})(x(0) - \hat{\alpha})] = 2 \end{aligned}$$

   How well does it work? Visualize the results for $T = 20$, and save your figure as `problem3_ekf_estimation.png`.