CS545 ROBOTICS HW2: **Kalman Filters**

Stephen Iota
iota@usc.edu
SID: 6862013543
October 7, 2021

Problem 1.

(a) *Do robot actions always increase uncertainty?* – No, not necessarily. If the robot's observations of the current state aligns well with its initial belief of the state (after taking action but before incorporating current observation), then the uncertainty will decrease that iteration.

(b) *What happens if at any point in Bayesian filtering the probability of a state assignment becomes 1? What are ways to avoid that?* – If the state assignment probability becomes 1, the filter has become over confident in its predictions. This may prevent the filter from predicting the robot's actual state, if there is any error at all.

One way to overcome this is to add some ficticious process noise when updating the belief over the state. The ground truth would still remain deterministic, but the added noise prevents the model from being overconfident in its predictions.

(c) *If an earthquake occurs, or there is a burglary, the alarim is likely to go off. If the alarm goes off, a police may arrive. Design a Bayesian network illustrating the causal relationship.* – See Figure 1.

(d) *In the recursive estimation case, what if the controls were dependent of observations? Visualize a Baysesian network illustrating the causal relationship.* – See Figure 2.

(e) *Why do Extended Kalman Filters fail in handling multiple hypotheses?* – EKFs model the transition probability distribution as a *unimodal* distribution. Suppose we would like the model to consider multiple hypotheses, but the *mean* of these hypotheses is not actually a candidate. Under these conditions, the EKF would still represent the situation as a unimodal distribution, centered about the mean of the hypotheses, and would frequently predict states that are not highly likely.

Problem 2.

(a) The vanilla Kalman Filter's (no process noise) position estimate and MSE can be seen in Figure 3, and in ∼/code/ directory of the Github repo.

(b) See Figure 4b for MSE plot.

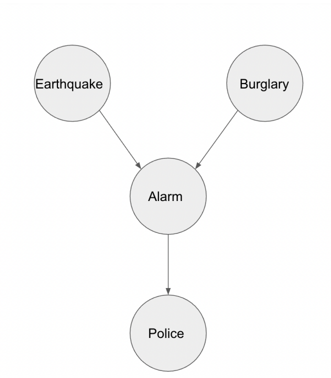


Figure 1: Bayesian network describing the causal relationship between the causal factors.

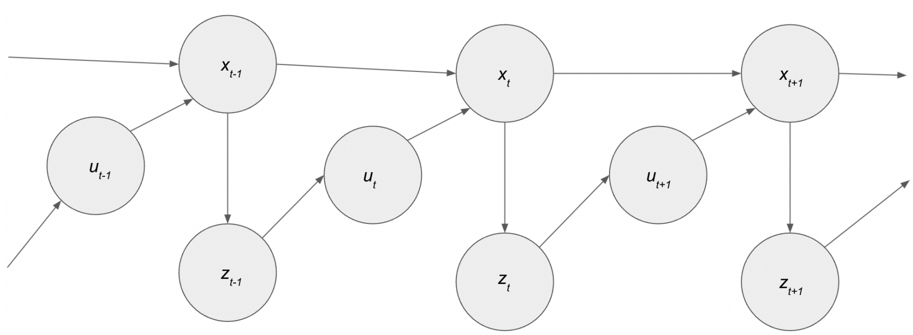
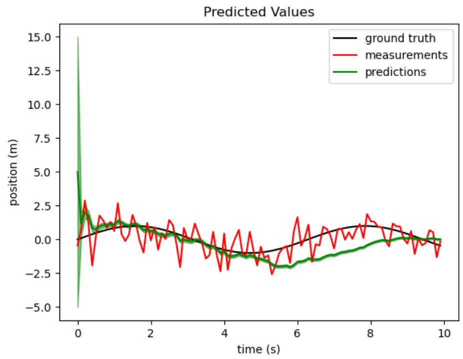
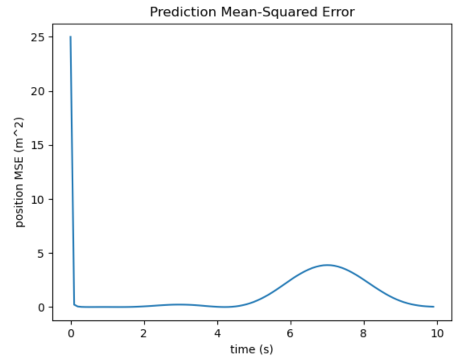


Figure 2: Bayesian network illustrating recursive state estimation if controls were influenced by observations.

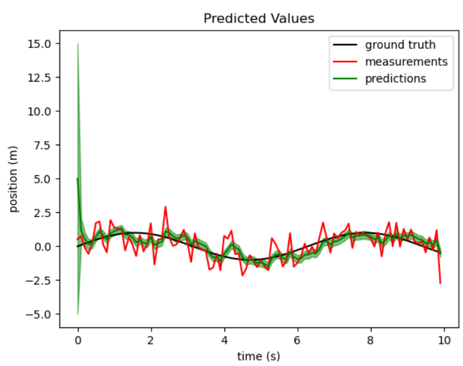


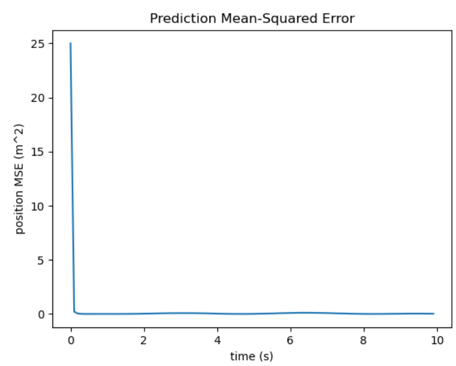


(a) Evolution of the estimated position and confidence over time.

(b) Mean Squared Error of the position of the object, average over $N = 10,000$ trials.

Figure 3: Figures for Problem 2(a).





(a) Evolution of the estimated position and confidence over time.

(b) Mean Squared Error of the position of the object, average over $N = 10,000$ trials.

Figure 4: Figures for Problem 2(b).

Problem 3.

(a) We can estimate the parameter α of the scalar system

$$
\begin{aligned}
x_{t+1} &= \alpha x_t + \omega(t) \\
z_t &= (x_t^2 + 1)^{1/2} + \nu(t)
\end{aligned}
$$

by modeling the system as an Extended Kalman Filter.

Let $\mathbf{x}_t = \begin{pmatrix} x_t & \alpha_t \end{pmatrix}^T$ denote the state vector at time t. Let $\mu_t = \begin{pmatrix} \mu_t^{(x)} & \alpha_t \end{pmatrix}^T$ be a random variable from a normal distribution denoting our belief of the state at time t. Then our system is characterized by the following nonlinear system.

$$
\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{g}(\mathbf{x}_t) + \omega(t) \\
\mathbf{z}_t &= \mathbf{h}(\mathbf{x}_t) + \nu(t) \\
\mathbf{g}(\mathbf{x}_t) &= \begin{pmatrix} \alpha_t x_t \\ \alpha_t \end{pmatrix} \\
\mathbf{h}(\mathbf{x}_t) &= \begin{pmatrix} (x_t^2 + 1)^{1/2} \\ 0 \end{pmatrix}
\end{aligned}
$$

In order to linearize our system, we can perform a first-order Taylor expansion about the current belief at that timestep, μ_t.

$$
\mathbf{g}(\mathbf{x}_t) \approx \mathbf{g}(\mu_t) + G_t(\mathbf{x_{t-1}} - \mu_{\mathbf{t-1}}) \tag{1}
$$

$$
\mathbf{h}(\mathbf{x}_t) \approx \mathbf{z}(\bar{\mu}_t) + H_t(\mathbf{x_{t-1}} - \bar{\mu}_{\mathbf{t-1}}) \tag{2}
$$

where G_t, H_t are the gradients of $\mathbf{g}, \mathbf{h}$ evaluated at the current belief μ_t, respectively.

$$
G_t = \begin{pmatrix} \alpha_t & \mu_t \\ 0 & 1 \end{pmatrix} \tag{3}
$$

$$
H_t = \begin{pmatrix} \frac{\bar{\mu}_t}{\sqrt{\bar{\mu}_t^2 + 1}} & 0 \\ 0 & 0 \end{pmatrix} \tag{4}
$$

Note that $\mathbf{h}$ and its gradient are evaluated at $\bar{\mu}_t$, which is the current belief of the model at the time of linearizing the function.

Now we can use $\mathbf{g}, \mathbf{h}, G, H$ in the EKF algorithm to estimate the parameter α, which would be the second element in the predicted state vector.

(b) Since the ground truth itself is stochastic, the performance of the EKF varies drastically from one run to another. One issue that arises is, as the predicted position approaches zero, the model can no longer update its estimate of α. Since position is used to estimate the parameter α (in $\mathbf{g}$), the filter can no longer update its estimate after a few timesteps when the predicted position goes to zero. See Figure 5 for estimate plot.

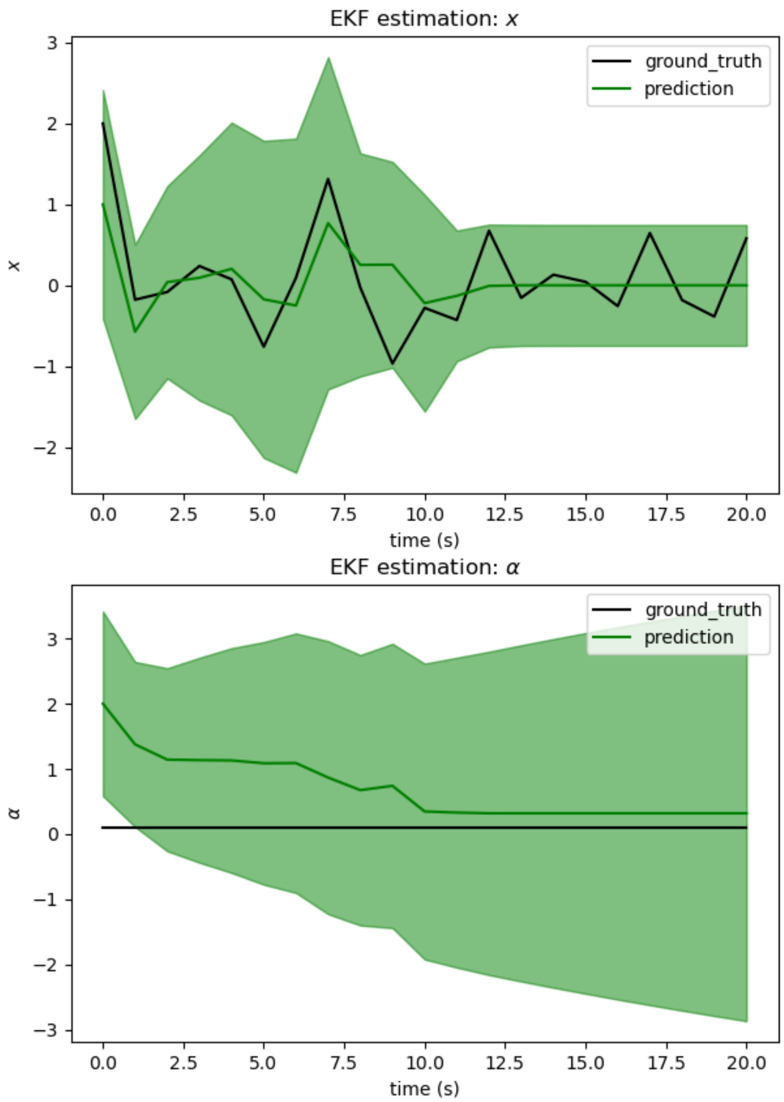


Figure 5: Extented Kalman Filter estimate of position x_t and parameter α.