Homework 7

Chien-Pin Chen
cchen144

The HW6 figure shows a robot moving left. While moving, it measures each second ($\Delta T = 1$s) the distance from the obstacle behind it (the corner) and the obstacle in front of it (the wall). In this scenario, we consider the corner as the reference point, i.e., with the coordinate 0. As shown in class, in order to simultaneously estimate the robot position and the position of the wall, we use the state vector $\underline{x}$ with the following three components: x_1 is the robot position measured from the corner, x_2 is the robot velocity with the positive direction towards the wall and x_3 is the wall position measured from the corner. The measurement y_1 is the distance from the robot to the corner and y_2 is the distance from the robot to the wall. A linear model describing this scenario is

$$\underline{x}(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix} w(k) \tag{1}$$

where the Gaussian random variable w(k) has zero mean and variance 1. The observation model is

$$\underline{y}(k) = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \underline{x}(k) + \underline{\theta}(k) \tag{2}$$

where the covariance matrix of Gaussian measurement noise $\underline{\theta}$ is $diag(10, 10)$. All variables in the above equations are expressed in cm and $cm = s$.

The course webpage (HW6) provides you the sequence of the measurements y_1 and y_2 recorded for $k = 1, 2, ...$ in the fille roboMes.mat. The data are organized so that $y(1, k)$ denotes the measurement $y_1(k)$ and $y(2, k)$ denotes the measurement $y_2(k)$, where $k = 1, 2, 3.....$

Design the Kalman smoother that takes the measurement sequences and produces the estimation sequences of the robot position, its velocity and the distance between the wall and the corner given all available data.

a) Plot on the same diagram the result of the forward Kalman filter and RTS backward iterations for **the robot position**.

b) Plot on the same diagram the result of the forward Kalman filter and RTS backward iterations for **the robot velocity**.

d) Plot on the same diagram the result of the forward Kalman filter and RTS backward iterations for **the wall position**.

e) For the robot position, compare the variance resulting from the Kalman filter and the variance resulting from the smoother.

Comment your results and trends in data. **Note:** Please keep in mind that you can use your Kalman filter code from HW6 with a slight modification, i.e., after removing 0.8 in the equation defining the velocity. Choose your initial

condition for $x_1(0)$ and $x_3(0)$, as well as corresponding variances the same way you did for the Kalman filter in HW6. **However, for $x_2(0)$ use the initial condition 0.2 or 1.6 and the variance 1.**
ANS:
For forward Kalman filter, I could use equations from homework 5 with slightly modify:

$$x_{k+1} = \Phi x_k + \Gamma w_k \tag{3}$$
$$y_k = H x_k + v_k \tag{4}$$

then, for computing in homework 7, I could set:

$$\Phi = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}, \text{and} Q = \text{var}(w(k)) = 1 \tag{5}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \text{and} R = \text{var}(\theta(k)) = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{6}$$

Now, I could use the same steps in homework 5 for computing forward Kalman filter in MATLAB. After forward Kalman filter, I get the sequence of the result of $\hat{x}_k$ and $\underline{P}_k$, and I could use the last elements in both sequence as initial value (condition) to do Kalman smoother with RTS algorithms:

$$\hat{\underline{x}}_{k|N} = \hat{\underline{x}}_k(+) + A_k(\hat{\underline{x}}_{k+1|N} - \hat{\underline{x}}_{k+1}(-)), \ \hat{\underline{x}}_{N|N} = \hat{\underline{x}}_N(+) \tag{7}$$
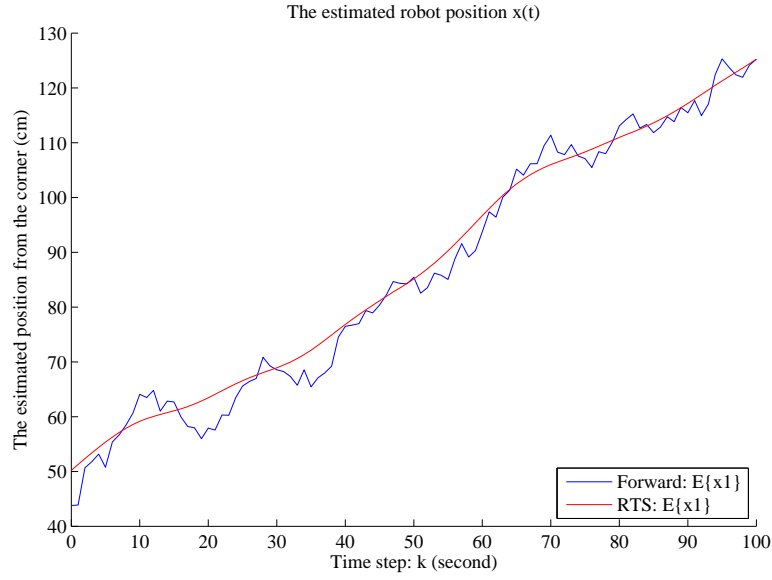
$$\underline{A}_k = \underline{P}_k(+)\Phi_k^T \underline{P}_{k+1}^{-1}(-) \tag{8}$$

$$\underline{P}_{k|N} = \underline{P}_k(+) + \underline{A}_k(\underline{P}_{k+1|N} - \underline{P}_{k+1}(-))\underline{A}_k^T, \ \underline{P}_{N|N} = \underline{P}_N(+) \tag{9}$$

And, after computing Kalman smoother and forward Kalman filter, I could plot the results as below:
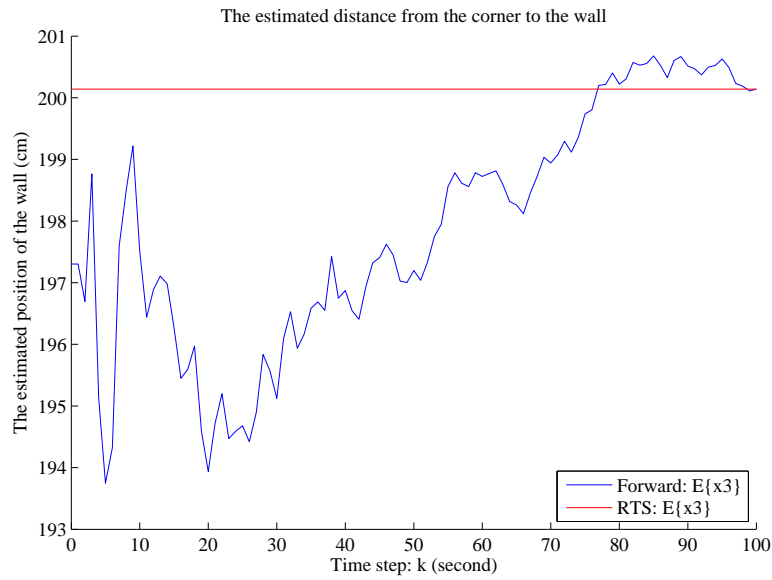**(a)** The diagram of results of the robot position.
From the below diagram, the result of Kalman smoother is smooth and each data point is not diverse a lot from the real data.

The estimated robot position x(t)

(b) The diagram of results of the robot velocity.
The result of velocity is also smooth, but it has more obviously turning curve. And, as indicated in "Applied Optimal Estimation", the result of smoother tends to lag (from backward) the measurement data whenever it has large changing curve.

The estimated robot position dx(t)
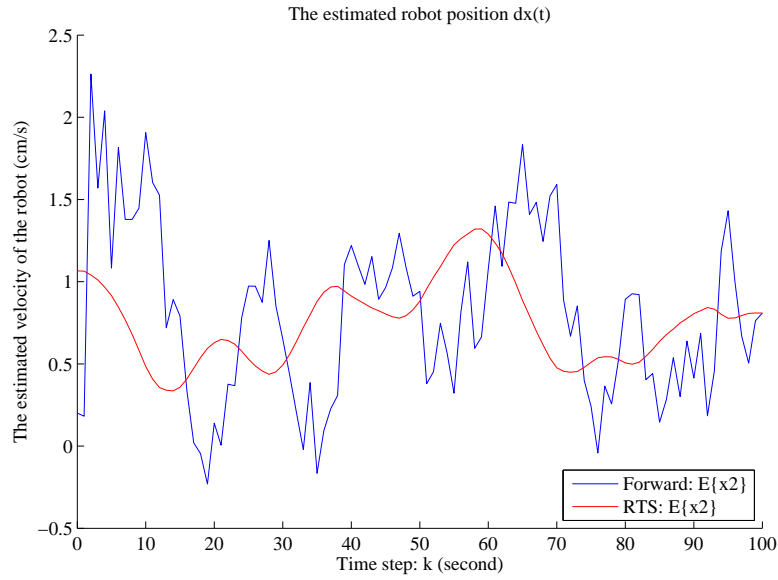
(d) The diagram of results of the wall position.

The result of wall position, after Kalman smoother, is fixed even when the measurement data are fluctuated.

The estimated distance from the corner to the wall

(e) The diagram of results of the variance of the robot position

From the below results, the variance from forward Kalman filter start to decrease and converge after 30 second. And, for RTS backward smoother, the variance start to decrease from 100 second and converge after 90 second. Although the variance of the backward after 10 second start to increase, it is still smaller than the variance of forward in the same time interval.

The vairance of the robot position from Kalman filter and smoother