Chalmers University of Technology

SSY130 Signal Processing

Hand in Problem 3

Yu-hsi LI

student_id: 20021026 *secret_key: 'Meganium'*

Task1

According to the state equation: $s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ s_4(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ y(t) \\ \dot{y}(t) \end{bmatrix}$

apply finite-difference approximation(T is the sampling time 0.01 s): $\dot{x}(t)|_{t=kT} = \frac{x(kT+T)-x(kT)}{T}$

The discrete state-space formulation can be derived by solving finite difference approximation in state equation: $\begin{aligned} x(k+1) - x(k) &= Tv_x(k) \\ v_x(k+1) - v_x(k) &= 0 \\ y(k+1) - y(k) &= Tv_y(k) \\ v_y(k+1) - v_y(k) &= 0 \end{aligned}$, will result in $\begin{aligned} s(k+1) &= As(k) + w(k) \\ z(k) &= Cs(k) + v(k) \end{aligned}$, where

$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, and $\begin{cases} w(k) = WGN(0, Q) \\ v(k) = WGN(0, R) \end{cases}$, Q is the variance of $w(k)$ and R is the variance of $v(k)$

The matrix C is responsible for selecting and extracting the position variables $x(k)$ and $y(k)$ from the state-space variables that are being measured. These measurements are affected by additive noise $v(k)$. Moreover, the process is influenced by zero mean discrete random noise $w(k)$.

Task2

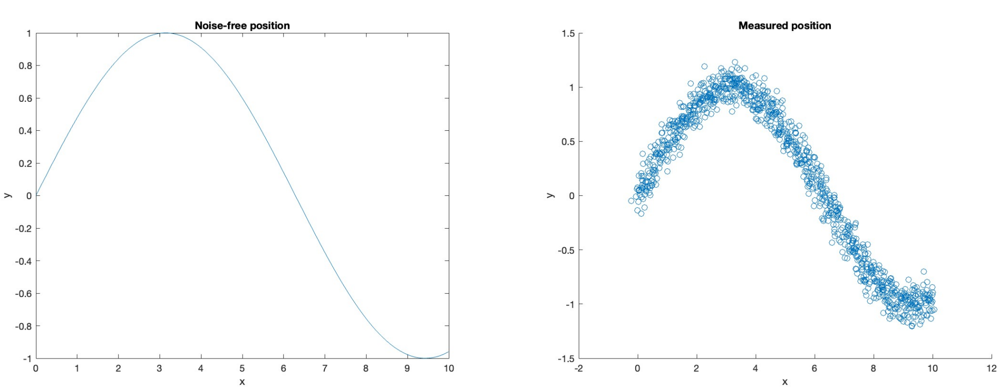

Figure 1: Noise-free position and measured position

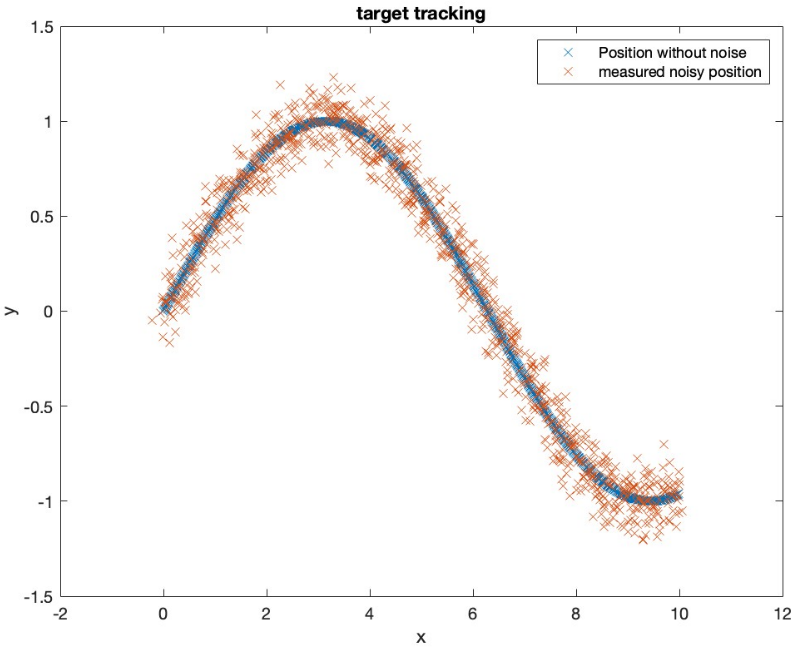

Figure 2: combination graph

Task3

```
for t=1:N
    % Filter update based on measurement Y(:,t)
    Xfilt(:,t) = Xpred(:,t) + P*C'*inv(C*P*C'+R) * (Y(:,t) - C*Xpred(:,t)); %TODO: This line is missing some code!
    % Covariance after measurement update
    Pplus = P - P*C'*inv(C*P*C'+R)*C*P; %TODO: This line is missing some code!
    % Prediction
    Xpred(:,t+1) = A * Xfilt(:,t); %TODO: This line is missing some code!
    % Covariance increase after prediction
    P = A*Pplus*A' + Q; %TODO: This line is missing some code!
end
```

Figure 3: Matlab code implementation for Kalman filtered

Task4

In order to utilize the Kalman filter, we need to define the matrices P_0, R, and Q. In this case, we assume an initial state vector of zero and set $P_0 = 10^{-6} \cdot I$, to be the initial state covariance matrix. We also know that Q is the covariance matrix of the process noise, R is the covariance matrix of the measurement noise, and considering they are uncorrelated.
The R matrix becomes diagonal, with each element representing the noise variance associated with each measurement. As the figure shows below, we can estimate the covariance matrix R using the known real position by analyzing statistically the measurement error.

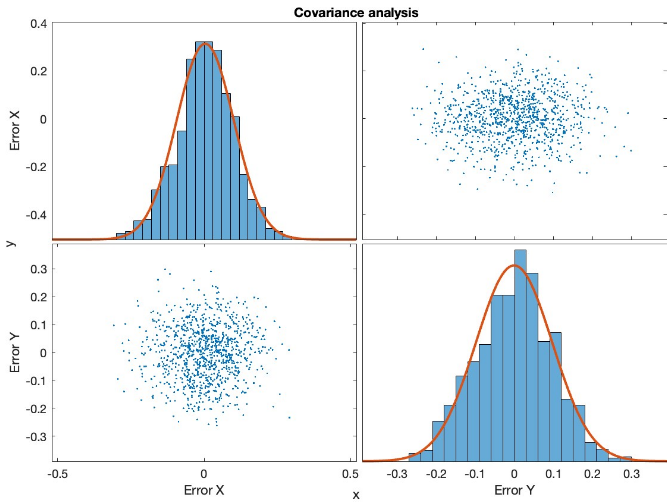

Figure 4: Measure error for X and Y

Fitting a zero-mean normal distribution, from the Matlab compilation, we get $R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} 9.6 * 10^{-3} & 0 \\ 0 & 9.5 * 10^{-3} \end{bmatrix}$. On the other hand, Q is defined to be the covariance of process noise. However, in our case, there is no way to directly determine the matrix values.
So, assuming that the process noises are only influenced by the velocities and we assume they are uncorrelated with each velocity, we can define $Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_{v_x^2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_y^2} \end{bmatrix} = \alpha \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ where α is an adjustable parameter. If we vary the α value and plot the position tracking and velocity tracking figure:

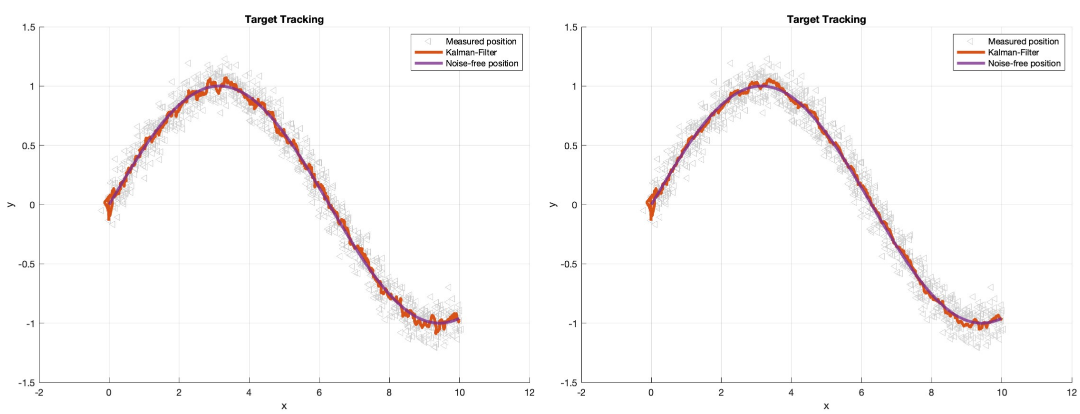

Figure 5: Kalman filter position tracking for $\alpha = 10^{-2}$ and $\alpha = 10^{-3}$

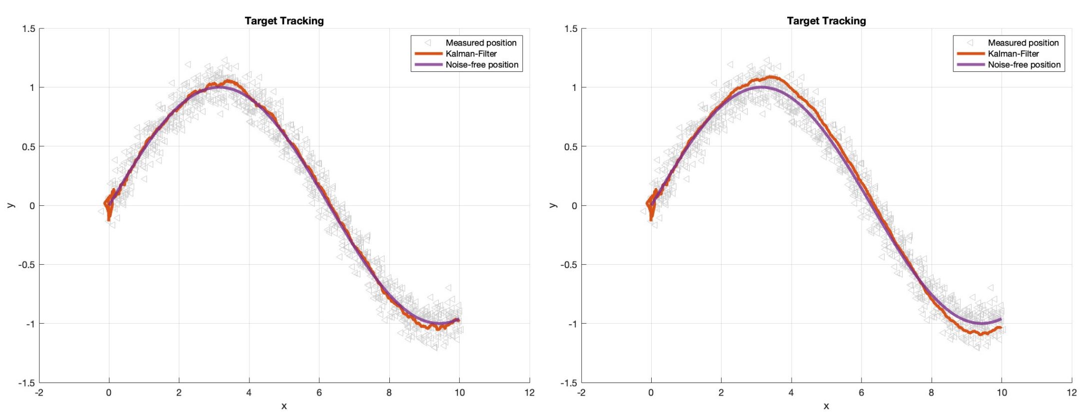

Figure 6: Kalman filter position tracking for $\alpha = 10^{-4}$ and $\alpha = 10^{-5}$

As we can see, the different α values affect the performance of the graph, while position tracking has little effect on the α value.

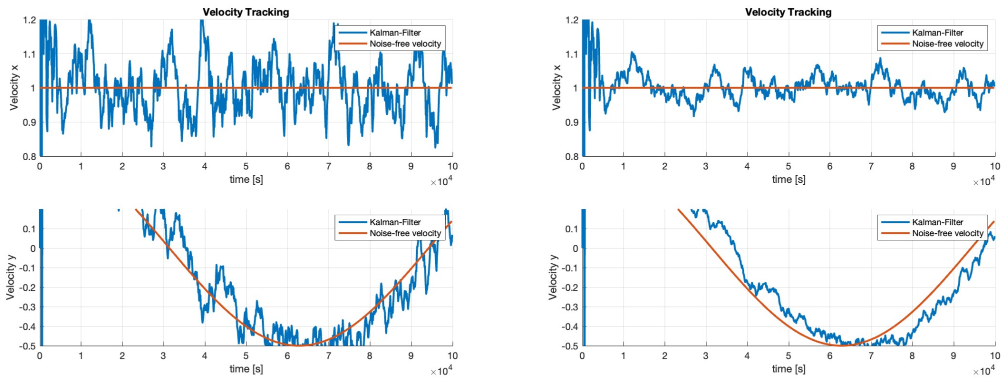

Figure 7: Kalman filter velocity tracking for $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$

A good value should be chosen to strike a good balance between noise filtering and response speed. While a smaller α value may have better performance in noise reduction but slower respond to velocity. After many tests, I found that $\alpha = 10^{-4}$ would be the best answer to it.

Also, I provided the estimated velocity tracking figures in the x and y direction comparing when $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$, we can see that using the 10^{-4} the result will be better. As mentioned earlier, this value effectively filters out noise but limits rapid speed variations.