Hand in Problem 3 – Applied Signal Processing

Fikri Farhan Witjaksono (19950822-Tyrantrum)

December 20, 2019

Task 1.

Process Model Equation

We define the initial states in the Continuous Time model.

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ s_4(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ y(t) \\ \dot{y}(t) \end{bmatrix} \tag{1}$$

$$\dot{s}(t) = \begin{cases} \dot{s}_1(t) = s_2(t) \\ \dot{s}_2(t) = 0 \\ \dot{s}_3(t) = s_1(t) \\ \dot{s}_4(t) = 0 \end{cases} \tag{2}$$

$$\dot{s}(t) = A_c s(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} s(t) \tag{3}$$

If we want to convert the Continuous Time model into Discrete Time model as shown below,

$$s(k+1) = A_d s(k) + w(k) \tag{4}$$

We have to use The Finite Difference Approximation to get the velocity is generally done by the equation below

$$\dot{x}(t)|_{t=kT} \approx \frac{x(kT+T) - x(kT)}{T} \tag{5}$$

where T = Sampling Time = 0.01 s

Let we assume $x(t) = s(t)$

$$\dot{s}(kT) \approx \frac{s(kT+T) - s(kT)}{T} \tag{6}$$

Moving T to the Left Hand Side (LHS) multiply with $\dot{s}(kT)$ as well as moving the term $s(kT)$ to LHS,

$$T\dot{s}(kT) + s(kT) = s(kT + T) \tag{7}$$

Flipping the LHS to RHS equation and RHS to LHS equation,

$$s(kT + T) = T\dot{s}(kT) + s(kT) = T\begin{bmatrix}\dot{s}_1(t)\\ \dot{s}_2(t)\\ \dot{s}_3(t)\\ \dot{s}_4(t)\end{bmatrix} + \begin{bmatrix}s_1(kT)\\ s_2(kT)\\ s_3(kT)\\ s_4(kT)\end{bmatrix} \tag{8}$$

Assuming that equation (2) works well for equation (8),

$$s(kT + T) = T\begin{bmatrix}s_2(kT)\\ 0\\ s_2(kT)\\ 0\end{bmatrix} + \begin{bmatrix}s_1(kT)\\ s_2(kT)\\ s_3(kT)\\ s_4(kT)\end{bmatrix} = \begin{bmatrix}Ts_2(kT) + s_1(kT)\\ s_2(kT)\\ Ts_4(kT) + s_3(kT)\\ s_4(kT)\end{bmatrix} \tag{9}$$

Rearranging equation (9) gives us

$$s(kT + T) = \begin{bmatrix}1 & T & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{10}$$

Finally we will obtain the Discrete time matrix A_d if we assume T=1

$$s(k + 1) = \begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

<u>Measurement Model Equation</u>

$$z(k) = Cs(k) + v(k)$$

$z(k)$ = Measurement of position

$v(k)$= additive noise to the measurement

$v(k)$ itself could be divided into 2 different components

$$v_x = additive\ noise\ to\ velocity\ in\ x - axis$$

$$v_y = additive\ noise\ to\ velocity\ in\ y - axis$$

$$z(k) = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix} s(k) + v(k)$$

Task 2.

```
x = 0:0.01:9.99; y = sin(0.5*x);
Y = [x;y]; Z = Y + 0.1*randn(size(Y));
```

where x and y represent the noise free position coordinates which is grouped in Y in the MATLAB

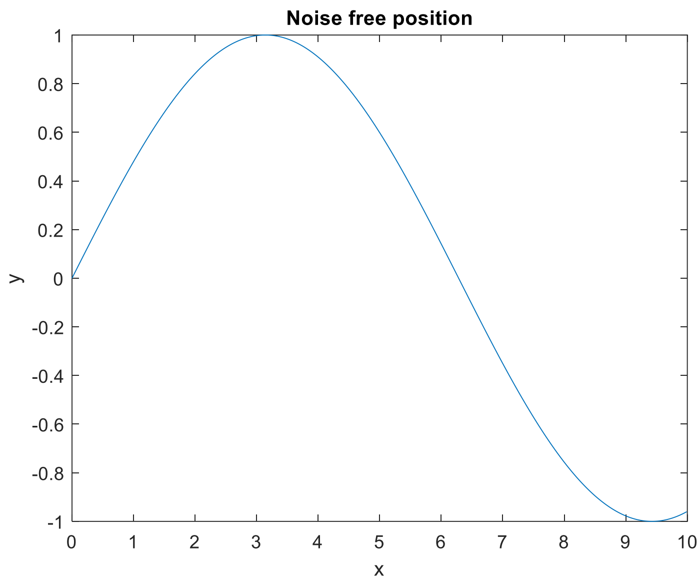

Figure 1. Noise Free of Position Measurement

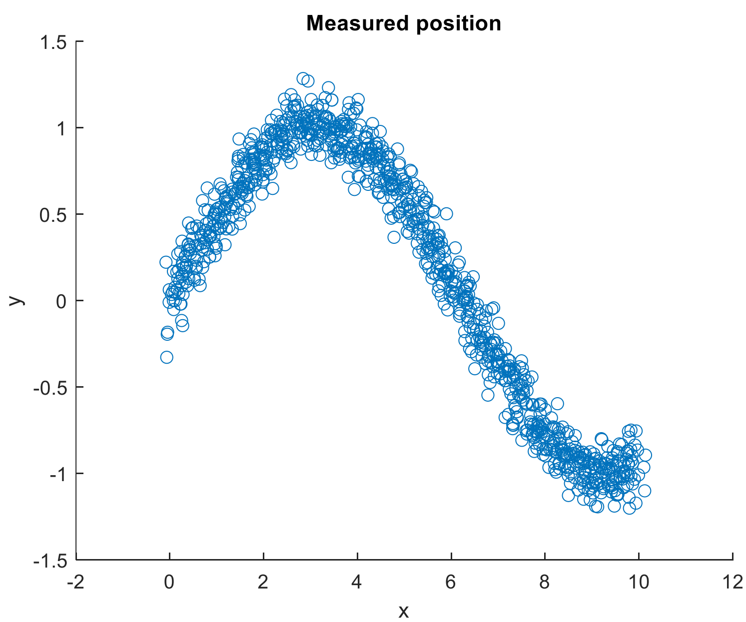

Figure 2. Crude Measurement of position

Task 3

Input Matrices in Matlab

(1). Y : size = [2x1000]

(2). $A = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

(3). C = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$,

(4). Let x is a stochastic process, w_n zero mean discrete time process noise of state n ($denoted\ by\ variance\ \sigma_n$). Here n = 1,2,3,4 since we have 4 states. Moreover, Q is covariance matrix of process noise where noise only affect the velocity without affecting the position. w_{nm} is the cross-covariance of the discrete time process noise. This imply that w1 = w3 = 0. Hence, we obtain Q matrix as shown below

$$Q = E[x \cdot x^T] = E\left(\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} [w_1 \quad w_2 \quad w_3 \quad w_4]\right) = E\begin{bmatrix} {w_1}^2 & \cancel{w_1 w_2} & \cancel{w_1 w_3} & \cancel{w_1 w_4} \\ \cancel{w_1 w_2} & {w_2}^2 & \cancel{w_2 w_3} & \cancel{w_2 w_4} \\ \cancel{w_1 w_3} & \cancel{w_2 w_3} & {w_3}^2 & \cancel{w_3 w_4} \\ \cancel{w_1 w_4} & \cancel{w_2 w_4} & \cancel{w_3 w_4} & {w_4}^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & {w_2}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & {w_4}^2 \end{bmatrix}$$

(5). Let v_n zero mean discrete time measurement noise of input n (σ_n) where n = [1,2]. R is covariance matrix of the measurement noise. We will obtain R as shown below

$$R = E\left[\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \cdot [v_1 \quad v_2]\right] = \pm \begin{pmatrix} v_1^2 & \cancel{v_1 v_2} \\ \cancel{v_1 v_2} & v_2^2 \end{pmatrix}$$

(6). Let x0 is estimate of x(0) assume by zeros.

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(7). Let P0 is estimate covariance of x(0) and I is identity matrix of size [4x4]

$$P_0 = 10^6 \cdot I$$

(8). Let Xfilt is the Kalman Filter equation update of the measurement.

$$\hat{x}_k^+ = \hat{x}_k + P_k C^T (C P_k C^T + R)^{-1} C P_k$$

(9). Let Pplus is the uncertainty update which is expressed by reduced variance

$$P_k^+ = P_k - P_k C^T (C P_k C^T + R)^{-1} C P_k$$

and the Xpred is the prediction of the state up to time k whose function is to project the state ahead

$$\hat{x}_{k+1} = A \hat{x}_k^+$$

Moreover, the prediction update whose function is to project the error covariance matrix ahead

$$P_{k+1} = AP_k^+ A^T + Q$$

Task 4

Kalman Filter Tuning

Table 1. Tuning the scale of P_0, Q, R and its Error comparing to the Noise Free position

Case	P_0	Q	R	Error RMS (x)	Error RMS (y)
1	10^6	1	1	**0.031**	**0.034**
2	10^6	1	10	**0.0242**	**0.0289**
3	10^6	1	10^6	**0.0344**	**0.388**
4	10^6	1	0.1	**0.0409**	**0.0425**
5	10^6	1	10^{-6}	**0.0973**	**0.0994**
6	10^5	0.1	1	**0.0242**	**0.0289**

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} |x_{true} - x_{estimated}|^2}, y_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} |y_{true} - y_{estimated}|^2} \quad (11)$$

where N = 1000 points

The tuning is done by changing the scale of R by increasing/decreasing with a factor of 10. By tuning the scale of R, we could observe that it is inversely proportional to changing the scale of P_0 and Q (increasing R corresponds to decreasing both, decreasing R corresponding to increasing both) as shown above in Table 1 comparing case 1 and case 6.

In case 4, we set Q to be in bigger in terms of scale compare to R (Q>R), this means that we trust the measurement of the position (camera based sensor) more that the discrete-time state space model that we have (see Figure 4 below for plot).

On the other hand, **in case 2**, setting R to be bigger in terms of scale compare to Q (R>Q) implies that we trust the model more than the sensor. Comparing between the cases that we have above concludes that it is more desirable to set (R>Q) where R is 10 times bigger in scale than the Q since the Error RMS (done by eq (11)) is the smallest of all cases, hence we could say that *the model is more trustworthy than the sensor to some degree* (see Figure 3 for plot).

However, setting R to be very large compare to Q or vice versa, will give us very bad accuracy estimation especially **in y-coordinates for case 3** and **both-coordinates for case 5** where R>>Q and Q>>R respectively (see Figure 5 and 6). Moreover, we observed that case 3 will give better precision in estimation than case 5. Therefore, we could conclude that setting either covariance to be too large compared to the other will result in overtrusting on either the model and the sensor will gives inaccurate and unprecise estimation.

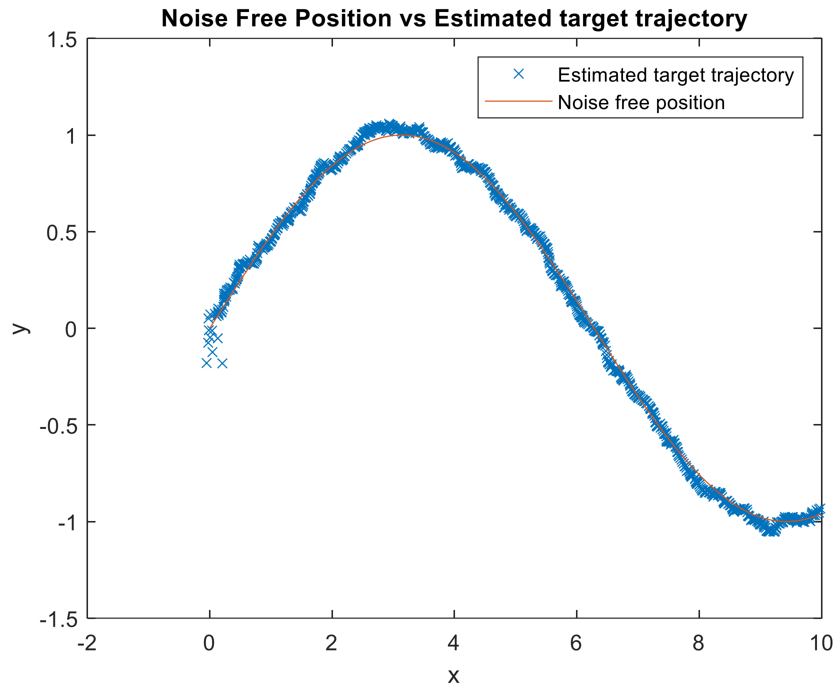

Figure 3. Noise Free Position vs Estimated target trajectory **(Case 2)**

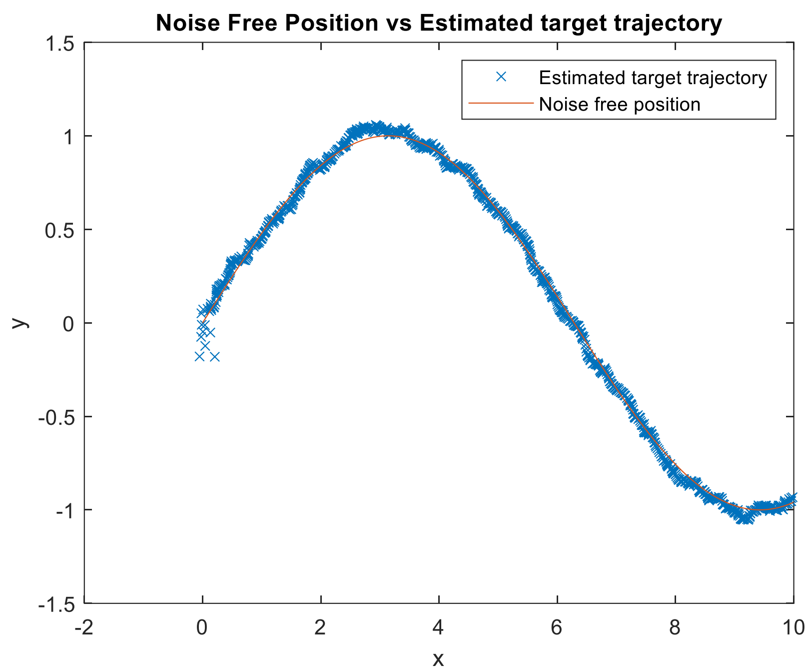

Figure 4. Noise Free Position vs Estimated target trajectory **(Case 4)**

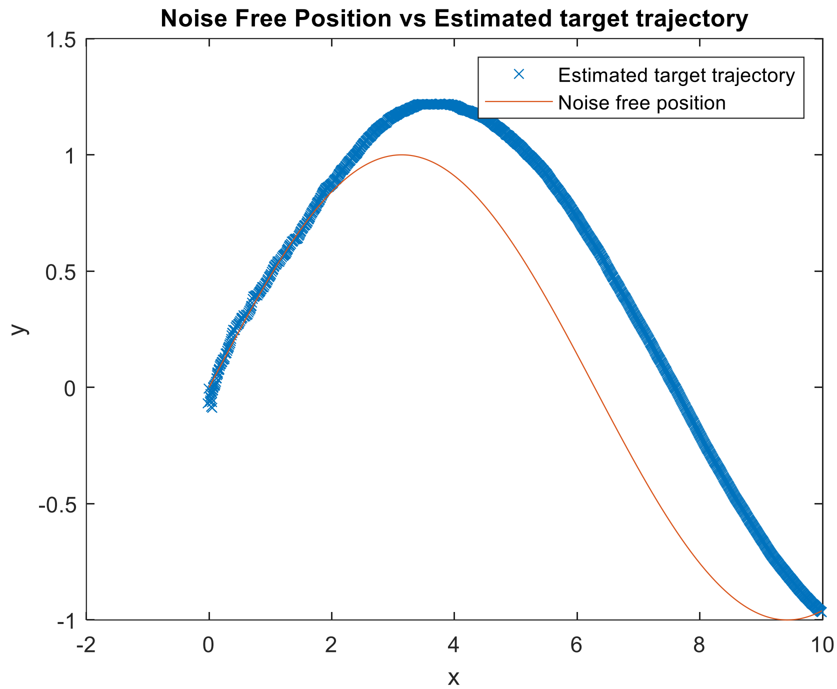

Figure 5. Noise Free Position vs Estimated target trajectory **(Case 3)**

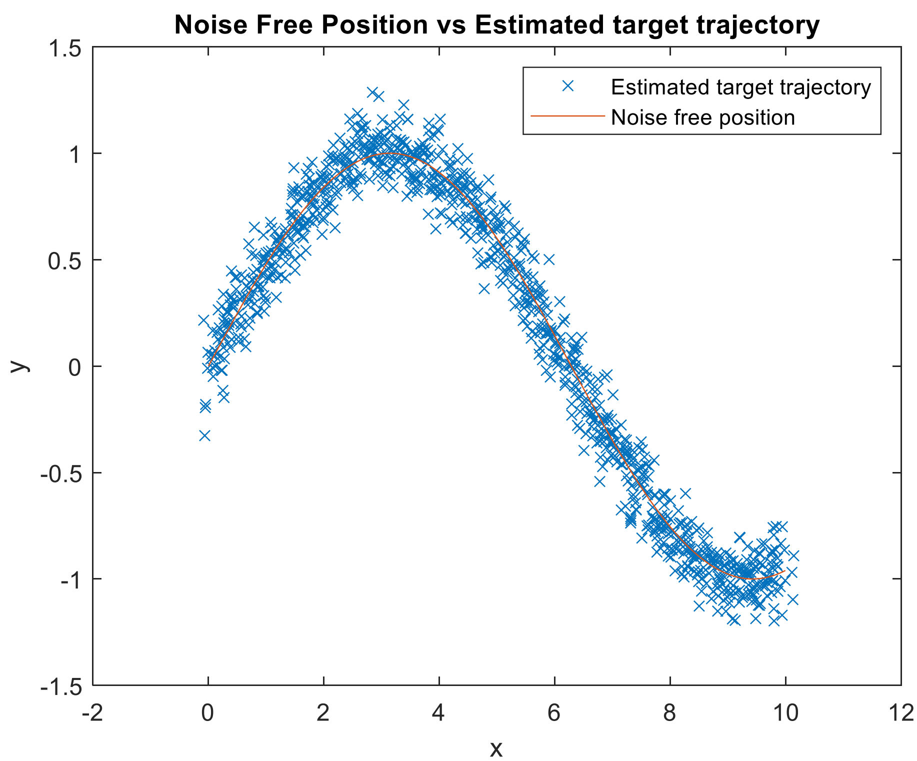

Figure 6. Noise Free Position vs Estimated target trajectory **(Case 5)**