ROB-UY 3303 Project 1

sk11361 / Seoyoon Kim

1. State-space model

In this project, the vehicle depicted in Figure 1 with m = 0.5kg, k = 0.5 N/m, b = 2 N·s/m is used.

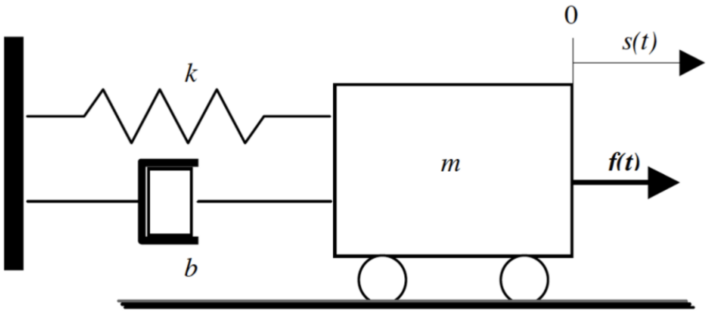


Figure 1. Robot Cart

Using basic equation of motion, I obtained the following equation.

$$m\ddot{s} = f(t) - ks - b\dot{s}$$

To represent the system as a state space model, I used the following relationships.

$$x_1 = s$$

$$x_2 = \dot{s}$$

Then, it can be represented as follows.

$$\dot{x_1} = x_2$$

$$\dot{x_2} = \frac{1}{m}(f - kx_1 - bx_2)$$

In matrix form, it becomes

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} f$$

Adding additive Gaussian white noise,

$$\dot{x} = Ax + Bu + \eta$$

where

$$x = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \quad \eta \sim N(0, Q)$$

2. Discretized system using Euler approach

Using one-step Euler integration,

$$x_t = x_{t-1} + f(x_{t-1}, u_t, n_t)\delta t = x_{t-1} + (Ax_{t-1} + Bu_t + n_t)\delta t$$

$$x_t = (I + \delta t A)x_{t-1} + \delta t B u_t + n_d$$

$$x_t = A_d x_{t-1} + B_d u_t + n_d$$

In the previous system,

$$A_d = I + \delta t \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}$$

$$B_d = \delta t \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$$

Therefore, discretized system is

$$x_t = (I + \delta t \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix})x_{t-1} + \delta t \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u_t + n_d$$

Mean and the covariance in the prediction step become as follows.

$$\overline{\mu_{t+1}} = A_d \mu_t + B_d u_t$$

$$\overline{\Sigma_{t+1}} = A_d \Sigma_t A_d^T + Q\delta t$$

Applying this to the system, I wrote the code of prediction part as Figure 2. uprev and covarPrev mean the previous mean and covariance respectively. Also, uEst and covarEst refer to estimated mean and covariance respectively. I chose noise Q to be [1;1].

```
function [covarEst,uEst] = pred_step(uPrev,covarPrev,ut,dt)
%covarPrev and uPrev are the previous mean and covariance respectively
%acc is the acceleration
%dt is the sampling time
m = 0.5; k = 3.5; b = 2;

A = [0 1; -k/m -b/m];
B = [0; 1/m];
Q = [1; 1];
I = eye(2);
Ad = I + A*dt;
Bd = B*dt;

uEst = Ad*uPrev + Bd*ut;
covarEst = Ad*covarPrev*transpose(Ad) + Q*dt;

 end
```

Figure 2. pred_step.m

Now, the observation model is as follows.

$$z = Cx + v,\ \ v \sim N(0, R_P)$$

In part 1, the measurement update will be given by the cart position, so C = [1 0]. Whereas in part 2, the update will be given by the velocity, so C = [0 1].

Since affline Transformations of Gaussian distributions are Gaussian, the result is a jointly normal distribution. Therefore, the conditional density $P(x_t|z_t)$ is a multivariate normal distribution.

$$K_t = \overline{\Sigma}_t C^T (C\overline{\Sigma}_t C^T + R_P)^{-1}$$

$$\mu_t = \overline{\mu}_t + K_t(z_t - C\overline{\mu}_t)$$

$$\Sigma_t = \overline{\Sigma}_t - K_t C\overline{\Sigma}_t$$

Applying this to the system, I wrote the code of update part as Figure 3. uCurr and covarCurr are updated mean and covariance respectively. uEst and covarEst refer to estimated mean and covariance, which are obtained at the prediction part.

```
function [uCurr,covarCurr] = upd_step(z_t,covarEst,uEst)
%z_t is the measurement
%covarEst and uEst are the predicted covariance and mean respectively
%uCurr and covarCurr are the updated mean and covariance respectively

C = [0 1];
Rp = 1;
K = covarEst*transpose(C)*inv(C*covarEst*transpose(C)+Rp);
uCurr = uEst + K*(z_t - C*uEst);
covarCurr = covarEst - K*C*covarEst;

end
```

Figure 3. upd_step.m

3. Result

Figure 4 is the result of Part 1. The actual and predicted motion mostly overlap.

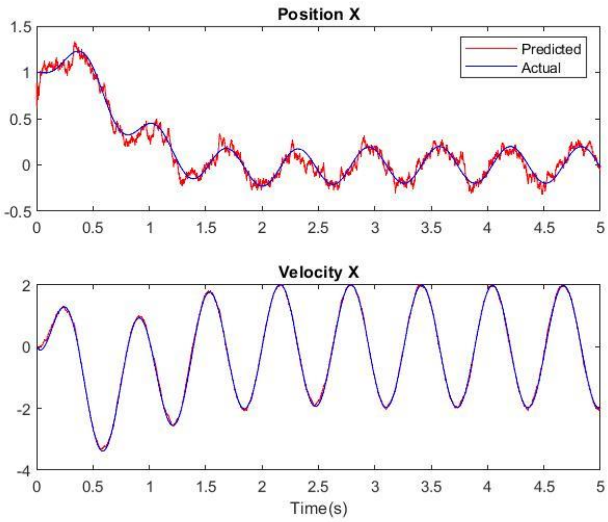


Figure 4. Result of Part 1

Figure 5 is the result of Part 2. The actual and predicted motion mostly overlap.

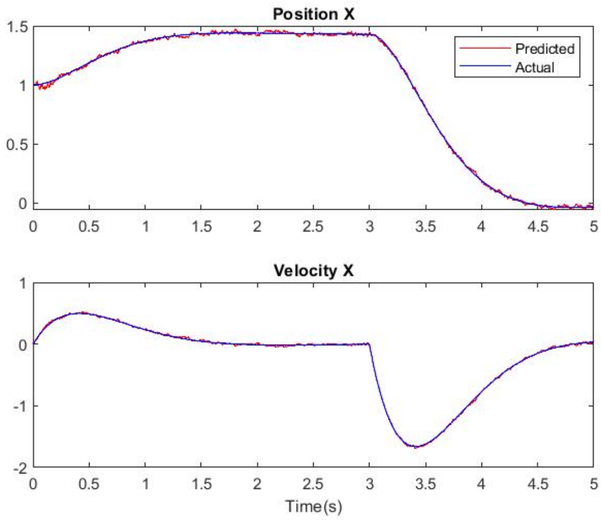
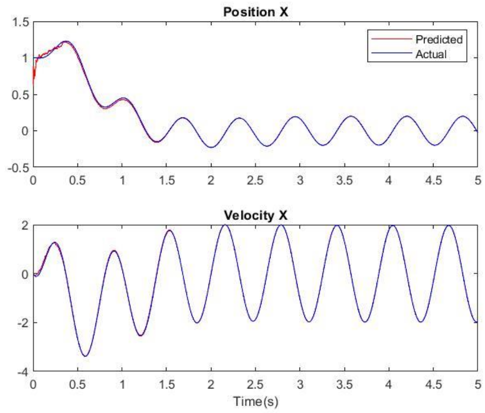
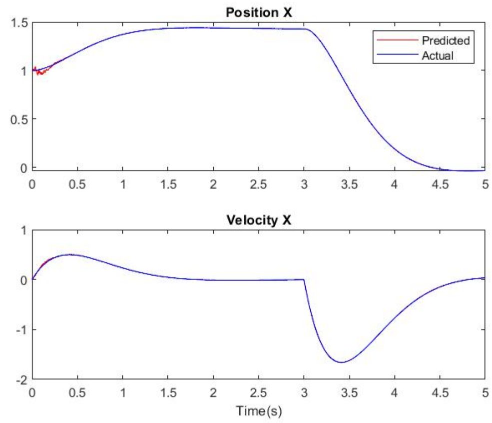


Figure 5. Result of Part 2

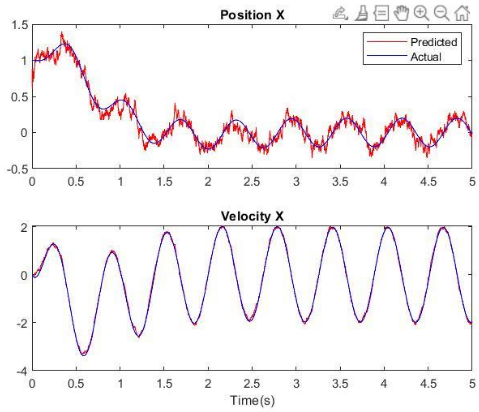
As I increase the noise Q or Rp, the error grows as expected. These are the results when the noise Q is [0; 0], [2; 2], [4; 4].



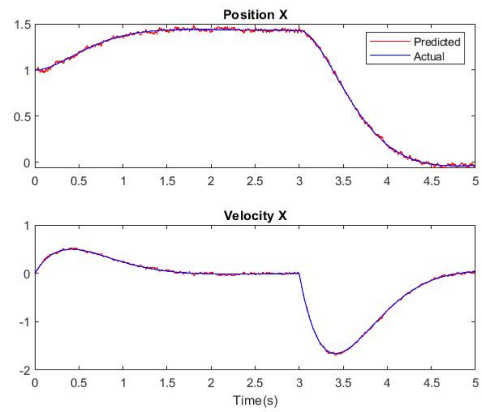
Part 1 (Q = [0; 0])



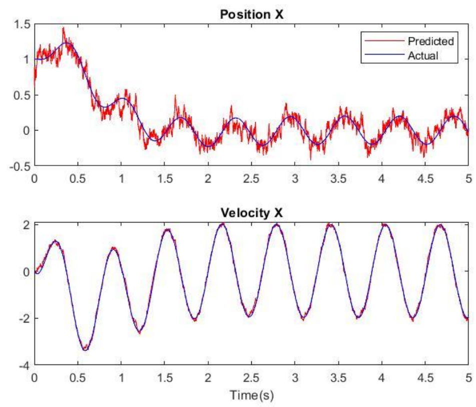
Part 2 (Q = [0; 0])



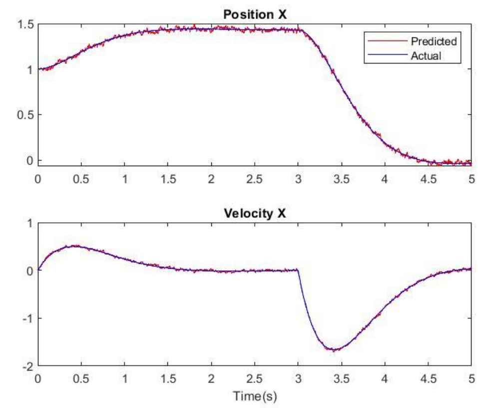
Part 1 (Q = [2; 2])



Part 2 (Q = [2; 2])



Part 1 (Q = [4; 4])



Part 2 (Q = [4; 4])