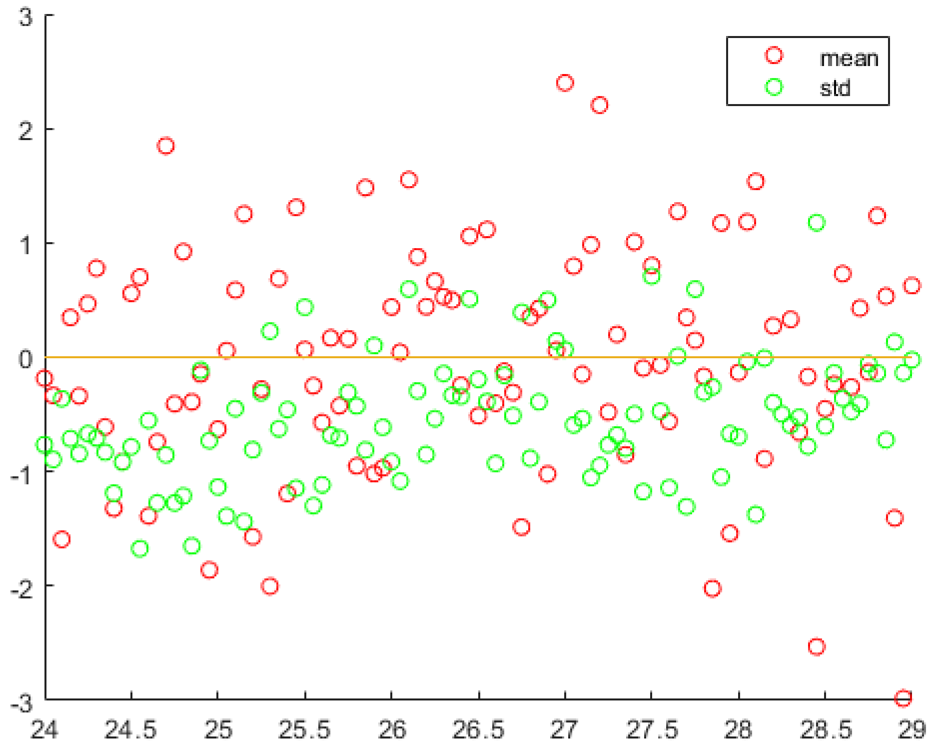
```
clear;clc;
```

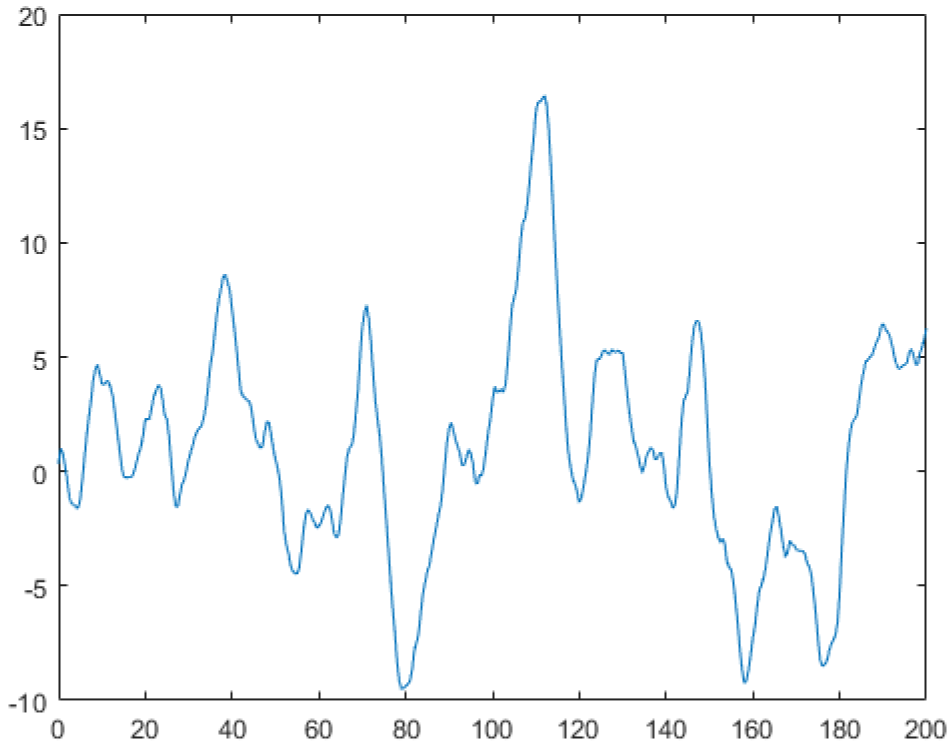
Wind Model

```
[numb,denb]=butter(2,.05);
numb=numb(1:2);
wi=filter(numb,denb,randn(500,1));
t=[0:.5:200]';
[T,~] = size(t);
[As,Bs,Cs,Ds]=tf2ss(numb,denb);
Wind_sys = ss(As,Bs,Cs,Ds);
%
x_w = zeros(2,T+1);
y_w = zeros(T,1);
v_w = randn(T,1);
%u_w0 = randn(T,1);
ms_rec = zeros(100,3);

i = 1;

for Qs = 24:0.05:29
    u_w = randn(T,1)*Qs;
    for k = 1:T
        x_w(:,k+1) = As * x_w(:,k) + Bs * u_w(k);
        y_w(k) = Cs * x_w(:,k) + Ds * v_w(k);
    end
    ms_rec(i,:) = [Qs, mean(y_w)+0.038, std(y_w)-5];
    i = i+1;
end
figure(1);
scatter(ms_rec(:,1), ms_rec(:,2), 'r');
hold on;
scatter(ms_rec(:,1), ms_rec(:,3), 'g');
plot(ms_rec(:,1), zeros(101,1))
legend('mean', 'std');
hold off;
```



Part iv

```
Qs_det = 27;
u_w = randn(T,1)*Qs_det;
wi_Qs = filter(numb,denb,u_w);
figure(2);
plot(t,wi_Qs);
```



Published with MATLAB® R2021a

Table of Contents


```
clear;clc;
```

Wind Model

```
[numb, denb]=butter(2,.05);
t=[0:.5:200]';
[T,~] = size(t);
Qs_det=1e-3;
u_w = randn(T,1)*Qs_det;
wi=filter(numb,denb,u_w); % wi is the output of wind velocity.
numb=numb(1:2); % This is the delay process, making the Ds zero.
[As,Bs,Cs,Ds]=tf2ss(numb,denb); % The wind system As,Bs,Cs,Ds.
Wind_force_ratio = 10; %This is a coefficient of relation between wind
 velocity and force.
```

Cart Model

```
[Ac,Bc,Cc,Dc]=ssdata(c2d(ss([0 1;0 -.2],
[0;1],eye(2),zeros(2,1)),.5)); % This is the cart model given by the
 prof.
%Gc=[1;10];
Gc=[1e-2;10];
Motor = sin(2*pi*t/10);
```

Combined Model

[Xs;Xc] is 4x1; F=[As 0; Gc*Cs Ac] is 4x4; G=[0;Bc] is 4x1; B1=[Bs;Gc*Ds] is 4x1; H=[0 Cc] is 2x4; J=[0;0]; B2=[1;1]; X=F*X+G*u+B1*process_noise_wind; Y=H*X+J*u+B2*measurement_noise_cart;

```
F = [As/Wind_force_ratio, zeros(2); Gc*Cs/Wind_force_ratio, Ac];
%F = [As, zeros(2); Gc*Cs, Ac];
G = [zeros(2,1);Bc];
B1 = [Bs zeros(2,3);zeros(2,3) Gc*Ds]/Wind_force_ratio;
%B1 = [Bs;Gc*Ds];
%B1 = [Bs zeros(2,3);zeros(2,3) Gc*Ds];
H = [zeros(2), Cc];
```

```
J = [0;0];
B2 = eye(2);
```

Compute the real data

```
X0 = [0;0;2;1];

% The measurement noise v is a 2x1 vector, the first element
 represents the
% postion and the second element represents the velocity.
% v = [v_pos; v_vel];
% To get a v.
v_pos = 0.5;
v_vel = 0.5;
v_all = [v_pos; v_vel];
% To realize these measurement noise.
v_pos_real = randn(1,T)*sqrt(v_pos);
v_vel_real = randn(1,T)*sqrt(v_vel);
v_all_real = [v_pos_real; v_vel_real];

% The process noise is actually the noise in wind model, which is a
 randn
% number.
% Qs = 27 is computed in Question 1,
%Qs=27;
Qs=1e-3;
wind_process_noise = randn(T,2)*Gc*sqrt(Qs);
wind_meas_noise = randn(2,T);
% %
d = [wind_process_noise';zeros(1,T); wind_meas_noise];
% %
[X_ss, Y_ss] = myss(F,G,H,J,B1,B2,Motor,X0,v_all_real,d);
```

Using lsim to compute real data

v_all_real = v_vel_real'; Qs=27; wv=randn(T,1)*Gc'*sqrt(Qs); Y_ss = lsim(ss(F,G,H,J),Motor,t,X0); Y_ss = Y_ss' + [v_pos_real; v_vel_real];

The Kalman Predictor process

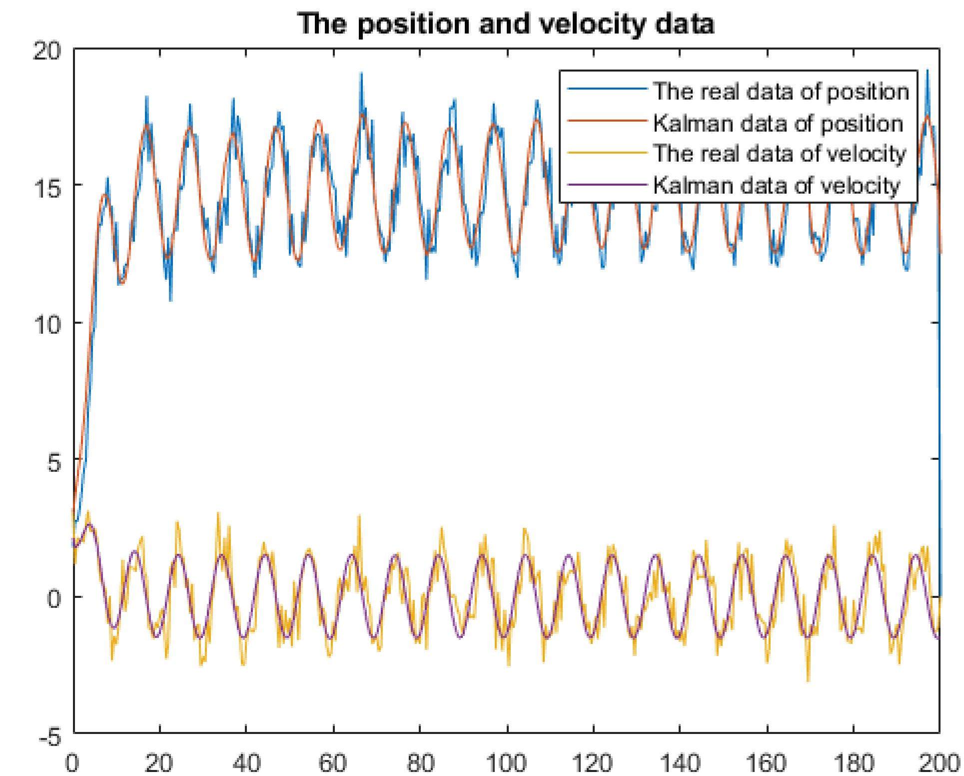
```
% We need to get Ro and Qo.
% Ro is the measurement noise matrix which is 2x2.
% Qo is the process noise matrix which is 4x4;
% In the combined model, the measurement noise only has v_pos and
 v_vel,
% thus the Ro should be diag([v_pos v_vel])
% Given that the process noise only has the wind noise Qs. Thus Qo
 should
% be diag([0,Qs,0,0]);

Ro = diag([v_pos, v_vel]);
```

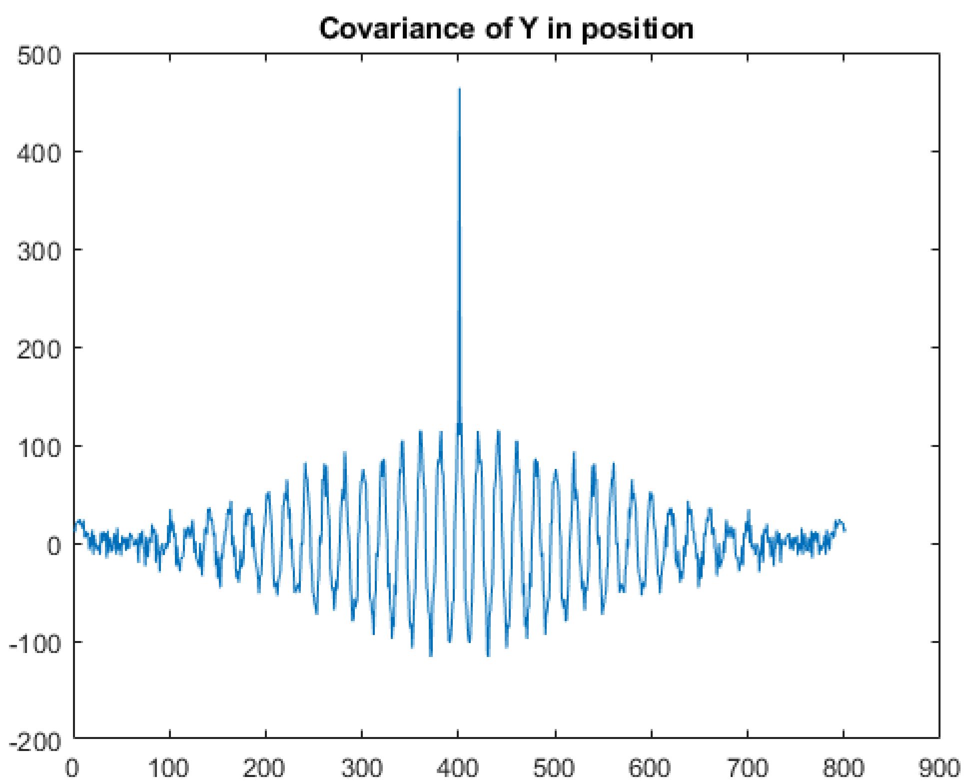
```
Qs = 1e-2;
Qo = diag([7,Qs,1e-4,1e-5]);

% Get X0 and P0
x0 = [0;0;2;1];
P0 = eye(4);

[X_hatrec, Y_hatrec, K_rec, P_rec] =
 myKalman(F,G,H,J,x0,Motor,Y_ss,P0,Qo,Ro);
```

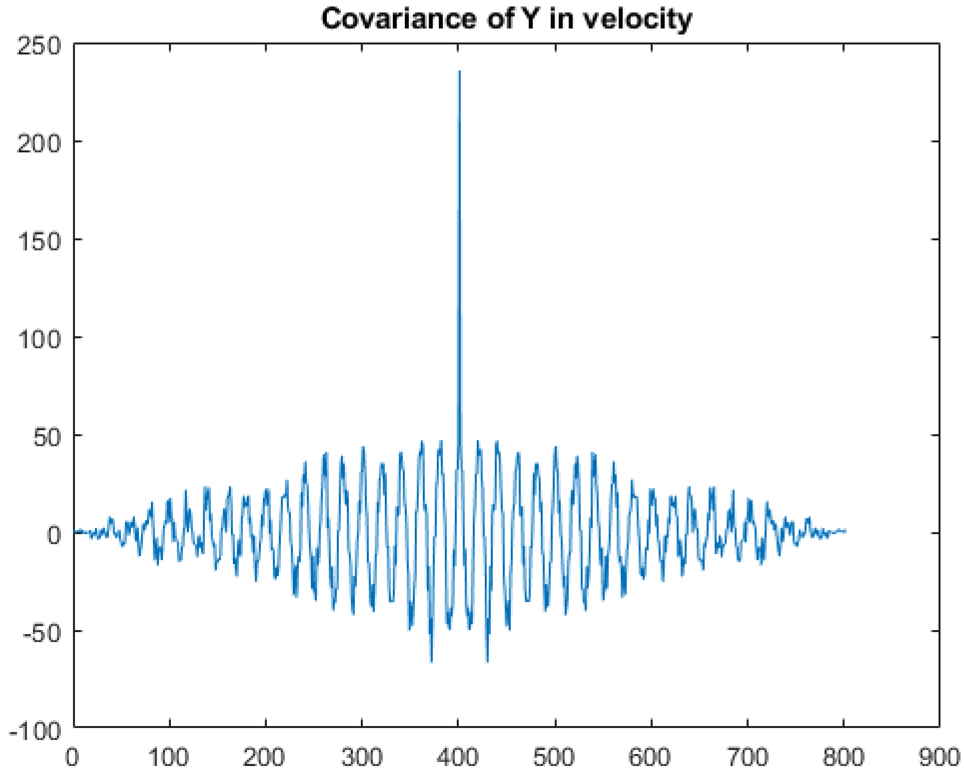
Plotting

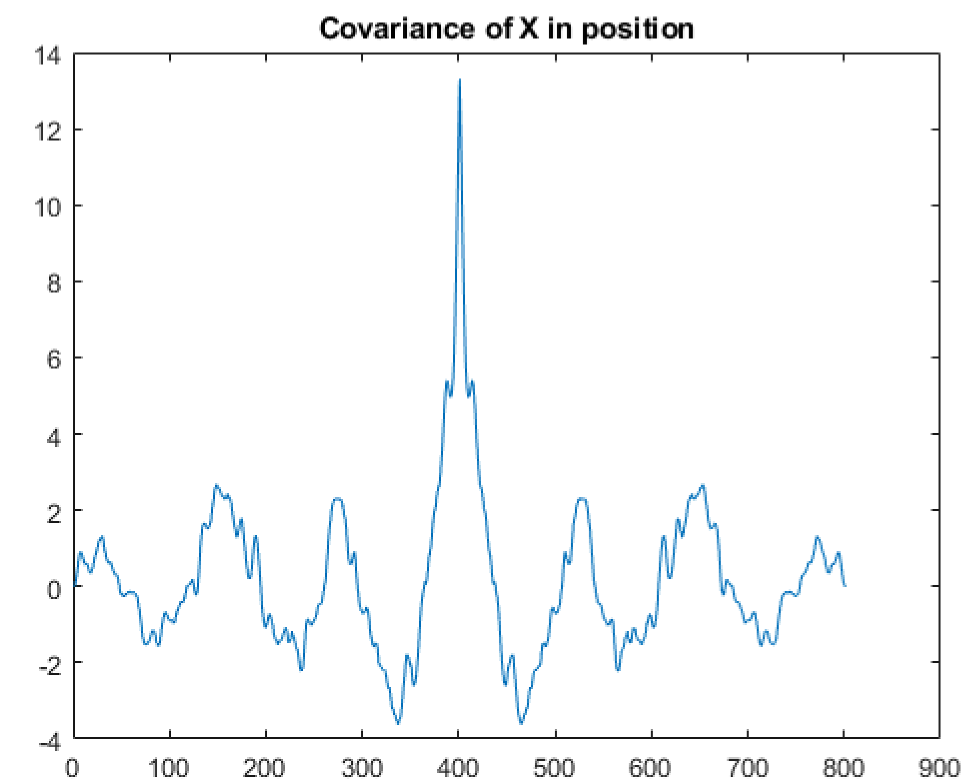
Now we have the real data y_ss and estimated Y_hatrec.

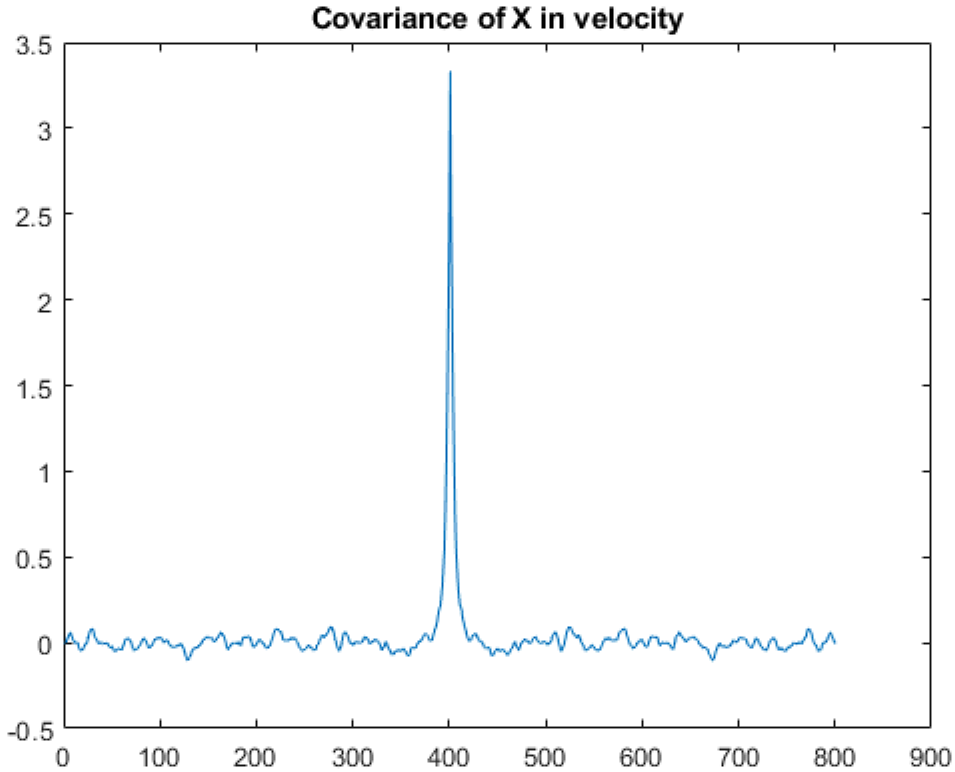
```
figure(1);
plot(t, Y_ss(1,:));
hold on
plot(t, Y_hatrec(1,:));
plot(t, Y_ss(2,:))
plot(t, Y_hatrec(2,:));
hold off;
legend('The real data of position','Kalman data of position','The real
 data of velocity','Kalman data of velocity');
title('The position and velocity data');
% figure(2);
% plot(t, X_ss(3,:), t, X_hatrec(3,:));
% hold on;
% plot(t, X_ss(4,:), t, X_hatrec(4,:));

%Y_tilda = Y_ss - Y_hatrec;
Y_tilda = Y_hatrec - Y_ss;
%X_tilda = X_hatrec - X_ss;
%Y_tilda_2_c=xcorr(Y_tilda(2,:)',Y_tilda(2,:)');
figure(3);
plot(xcorr(Y_tilda(1,:)));
title('Covariance of Y in position');
figure(4);
plot(xcorr(Y_tilda(2,:)));
title('Covariance of Y in velocity');

X_tilda = X_hatrec - X_ss;
figure(5);
plot(xcorr(X_tilda(3,:)));
title('Covariance of X in position');
figure(6);
plot(xcorr(X_tilda(4,:)));
title('Covariance of X in velocity');
```


The position and velocity data
The real data of position
Kalman data of position
The real data of velocity
Kalman data of velocity
20
15
10
5
0
-5
0
20
40
60
80
100
120
140
160
180
200


Covariance of Y in position
500
400
300
200
100
0
-100
-200
0
100
200
300
400
500
600
700
800
900


Covariance of Y in velocity
250
200
150
100
50
0
-50
-100
0
100
200
300
400
500
600
700
800
900


Covariance of X in position
14
12
10
8
6
4
2
0
-2
-4
0
100
200
300
400
500
600
700
800
900



Published with MATLAB® R2021a

Table of Contents


```
clear;clc;
```

Wind Model

```
[numb, denb]=butter(2,.05);
t=[0:.5:200]';
[T,~] = size(t);
Qs_det=1.3;
u_w = randn(T,1)*Qs_det;
wi=filter(numb,denb,u_w); % wi is the output of wind velocity.
numb=numb(1:2); % This is the delay process, making the Ds zero.
[As,Bs,Cs,Ds]=tf2ss(numb,denb); % The wind system As,Bs,Cs,Ds.
Wind_force_ratio = 10; %This is a coefficient of relation between wind
 velocity and force.
```

Cart Model

```
[Ac,Bc,Cc,Dc]=ssdata(c2d(ss([0 1;0 -.2],
[0;1],eye(2),zeros(2,1)),.5)); % This is the cart model given by the
 prof.
%Gc=[1;10];
Gc=[0;1];
Motor = sin(2*pi*t/10);
```

Combined Model

[Xs;Xc] is 4x1; F=[As 0; Gc*Cs Ac] is 4x4; G=[0;Bc] is 4x1; B1=[Bs;Gc*Ds] is 4x1; H=[0 Cc] is 2x4; J=[0;0]; B2=[1;1]; X=F*X+G*u+B1*process_noise_wind; Y=H*X+J*u+B2*measurement_noise_cart;

```
%F = [As/Wind_force_ratio, zeros(2); Gc*Cs, Ac];
F = [As/Wind_force_ratio, zeros(2); Gc*Cs/Wind_force_ratio, Ac];
G = [zeros(2,1);Bc];
B1 = [Bs zeros(2,3);zeros(2,3) Gc*Ds]/Wind_force_ratio;
H = [zeros(2), Cc];
J = [0;0];
B2 = eye(2);
```

Compute the real data

```
X0 = [0.5;0.5;2;1];
```

```
% The measurement noise v is a 2x1 vector, the first element
 represents the
% postion and the second element represents the velocity.
% v = [v_pos; v_vel];
% To get a v.
v_pos = 0.05;
v_vel = 0.05;
v_all = [v_pos; v_vel];
% To realize these measurement noise.
v_pos_real = randn(1,T)*sqrt(v_pos);
v_vel_real = randn(1,T)*sqrt(v_vel);
v_all_real = [v_pos_real; v_vel_real];

% The process noise is actually the noise in wind model, which is a
 randn
% number.
Qs=3;
wind_process_noise = randn(2,T)*Qs;
wind_meas_noise = randn(2,T);

d = [wind_process_noise; wind_meas_noise];

Ro = diag([v_pos,v_vel]);
Qs = 5;
Qo = diag([Qs,0,1e-5,1e-5]);



[L,~,~] = dlqr(F,G,Qo,v_vel);
Fnew = F - G*L;
Gnew = zeros(4,1);
Hnew = H;
Jnew = J;

[X_ss, Y_ss] = myss(Fnew,Gnew,Hnew,Jnew,B1,B2,Motor,X0,v_all_real,d);
```

The Kalman Predictor process with the feed-back control

```
% We need to get Ro and Qo.
% Ro is the measurement noise matrix which is 2x2.
% Qo is the process noise matrix which is 4x4;
% In the combined model, the measurement noise only has v_pos and
 v_vel,
% thus the Ro should be diag([v_pos v_vel])
% Given that the process noise only has the wind noise Qs. Thus Qo
 should
% be diag([0,Qs,0,0]);
```

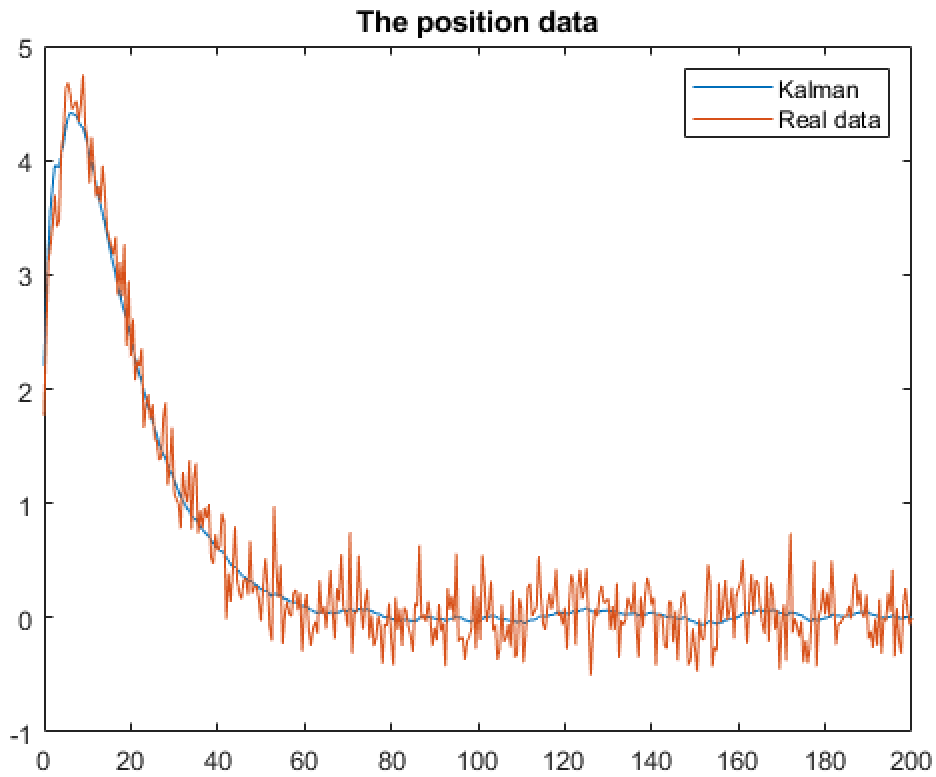
```
% Get X0 and P0
x0 = [0.5;0.5;2;1];
P0 = eye(4);

% Compute L
% Qc = [0,0,0,0;0,0,0,0;0,0,1,0;0,0,0,0];
% Rc = 0.003;

% Update the matrices

% Compute K
%K = dlqr(F,H',Qo,Ro);

[X_hatrec, Y_hatrec, K_rec, P_rec] =
 myKalman(Fnew,Gnew,Hnew,Jnew,x0,Motor,Y_ss,P0,Qo,Ro);
```

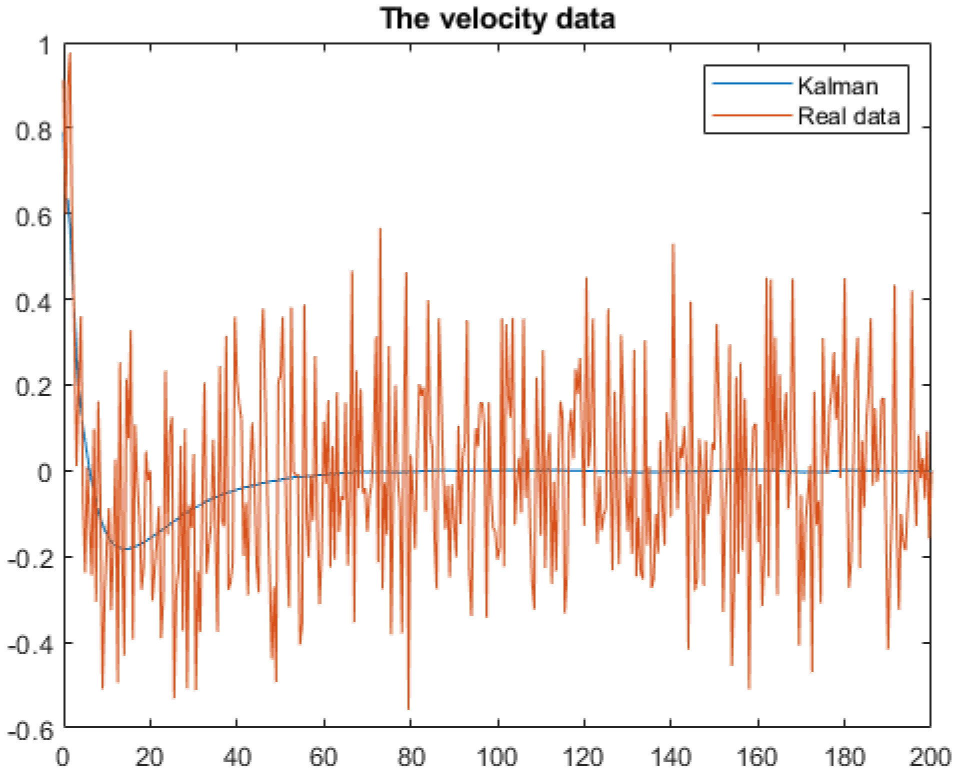
Plotting

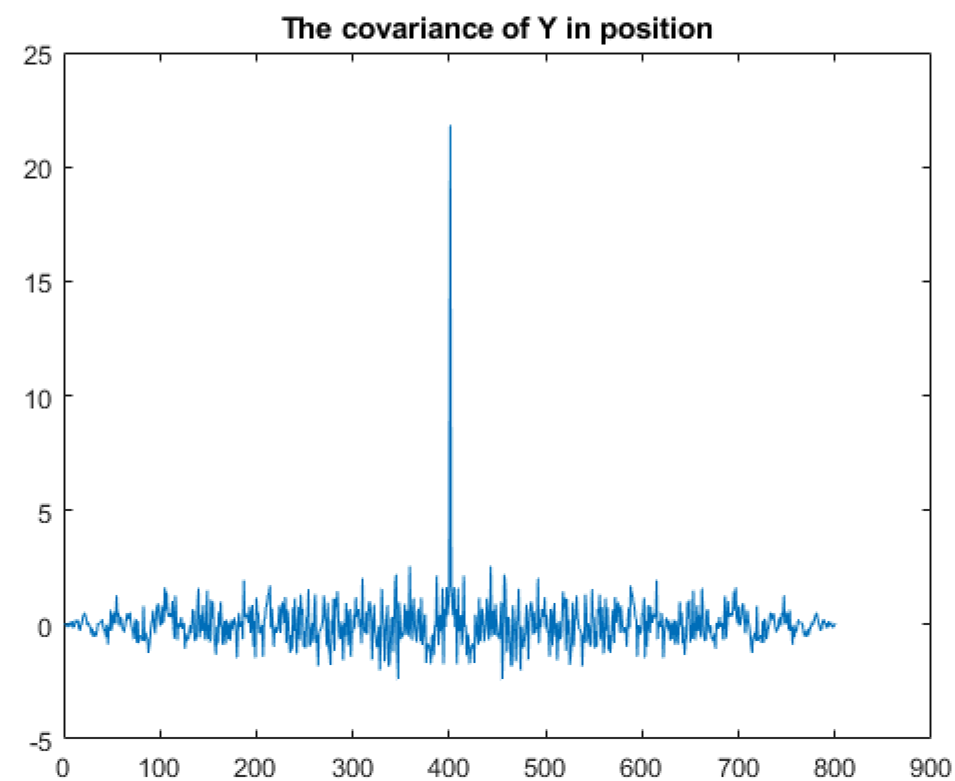
Now we have the real data y_ss and estimated Y_hatrec. figure(1); plot(t, Y_ss(1,:), t, Y_hatrec(1,:)); hold on; plot(t, Y_ss(2,:), t, Y_hatrec(2,:)); figure(2); plot(t, X_ss(3,:), t, X_hatrec(3,:)); hold on; plot(t, X_ss(4,:), t, X_hatrec(4,:));

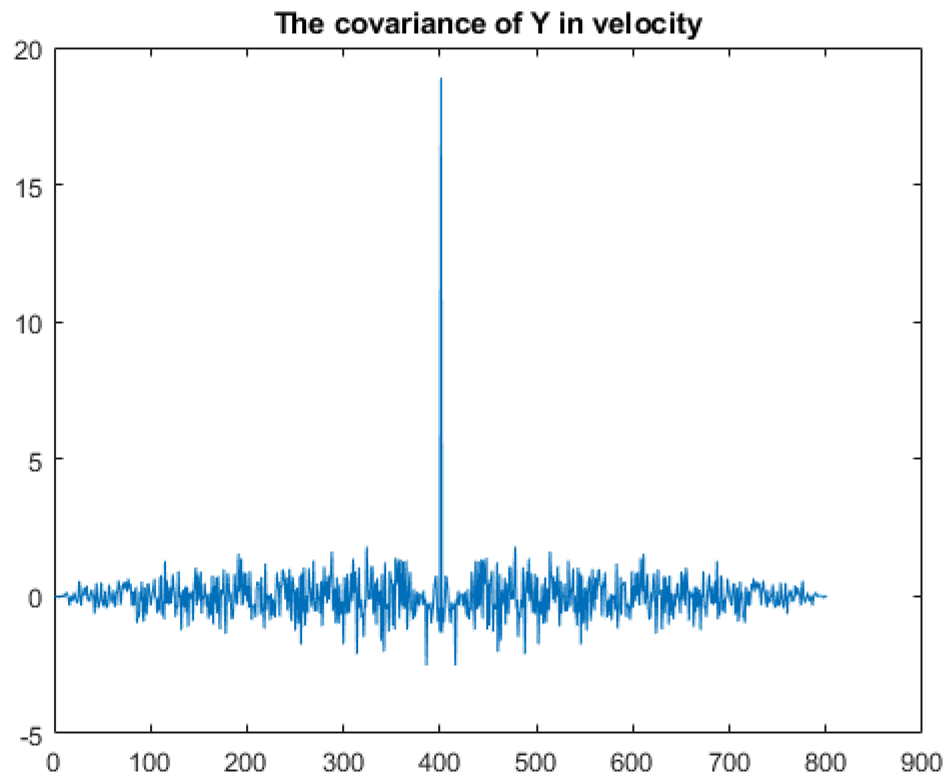
```
% figure(1);
% plot(t,X_hatrec(3,:));
% hold on;
% plot(t,Y_ss(1,:));
% hold off;

figure(1);
plot(t,Y_hatrec(1,:));
hold on;
plot(t,Y_ss(1,:));
hold off;
legend('Kalman','Real data');
title('The position data');

figure(2);
plot(t,Y_hatrec(2,:));
hold on;
plot(t,Y_ss(2,:));
hold off;
legend('Kalman','Real data');
title('The velocity data');
%
% figure(2);
% plot(t,X_hatrec(2,:));
% hold on;
% plot(t,Y_ss(2,:));
% hold off;

Y_tilda = Y_hatrec - Y_ss;
%Y_tilda_2_c=xcorr(Y_tilda(2,:)',Y_tilda(2,:)');
figure(3);
```

```
plot(xcorr(Y_tilda(1,:)));
title('The covariance of Y in position');
figure(4);
plot(xcorr(Y_tilda(2,:)));
title('The covariance of Y in velocity');

% X_tilda = X_hatrec - X_ss;
% %Y_tilda_2_c=xcorr(Y_tilda(2,:)',Y_tilda(2,:)');
% figure(5);
% plot(xcorr(X_tilda(3,:)));
% figure(6);
% plot(xcorr(X_tilda(4,:)));
```




The velocity data
Kalman
Real data


The covariance of Y in position



Published with MATLAB® R2021a

```
function [X,Y]=myss(A,B,C,D,G,H,u,x0,v,d)

Size=length(u);

X=zeros(length(A),Size);
Y=zeros(length(D),Size);
X(1:length(A),1) = x0;

for i = 1:Size - 1
    Y(1:length(D),i) = C * X(1:length(A), i) + D * u(i) + H * v(:,i);
    X(1:length(A), i+1) = A * X(1:length(A), i) + B * u(i) + G *
 d(:,i);
end
end
```

```
Not enough input arguments.

Error in myss (line 3)
Size=length(u);
```

Published with MATLAB® R2021a

```
function [X_hatrec, Y_hatrec, K_rec, P_rec] =
 myKalman(A,B,C,D,x0,u,y,P0,Qo,Ro)

Size = length(u);
X_hatrec = zeros(length(A), Size);
Y_hatrec = zeros(length(D), Size);
P_rec = zeros(Size, 2*length(A));
K_rec = zeros(Size, length(A));

X_hat = x0;
Pp = P0;

for i = 1:Size
    X_hatrec(:,i) = X_hat;
    P_rec(i,:) = [Pp(1,:), Pp(2,:)];
    K = A * Pp * C' / (C * Pp * C' + Ro);
    K_rec(i,:) = [K(1,:), K(2,:)];
    X_hat = (A - K * C) * X_hat + B * u(i) + K * y(:,i);
    Pp = A * Pp * A' - A * Pp * C' / (C * Pp * C' + Ro) * C * Pp * A'
 + Qo;
    Y_hatrec(:,i) = (C * X_hat);
end
end
```

```
Not enough input arguments.

Error in myKalman (line 3)
Size = length(u);
```

Published with MATLAB® R2021a

Part ii

Using the lyapunov function

$$X = AXA^T + Q.$$

$$x_{s,k+1} - \bar{x}_{s,k+1} = A_s x_{s,k} + B_s w_{s,k} - [A_s\bar{x}_{s,k} + B_s\bar{w}_{s,k}]$$

$$= A_s x_{s,k} + B_s w_{s,k} - A_s\bar{x}_{s,k} - B_s\bar{w}_{s,k}$$

$$= A_s(x_{s,k} - \bar{x}_{s,k}) + B_s(w_{s,k} - \bar{w}_{s,k})$$

$$E[(x_{s,k+1} - \bar{x}_{s,k})(x_{s,k+1} - \bar{x}_{s,k})^T]$$

$$= [A_s(x_{s,k} - \bar{x}_{s,k}) + B_s(w_{s,k} - \bar{w}_{s,k})]^T$$

$$= (x_{s,k} - \bar{x}_{s,k})^T A_s^T + (w_{s,k} - \bar{w}_{s,k})^T B_s^T$$

$$C_x = A_s C_x A^T + BQB^T$$

Similarly,

$$E([y_{s,k} - \bar{y}_{s,k}][y_{s,k} - y_{s,k}]^T) = [C_s(y_{s,k} - \bar{y}_{s,k}) + D_s(V_{s,k} - \bar{V}_{s,k})]^T$$

$$= (y_{s,k} - \bar{y}_{s,k})^T C_s^T + (V_{s,k} - \bar{V}_{s,k})^T D_s^T$$

$D_s = 0$. thus $C_y = C_s C_y C_s^T$

```
%Check observability
rank(obsv(F,H));
Ans = 4;
```

Thus this system is observable, means that it can build the initial state of wind and cart given the current condition.

Part iii

$$x_{s,k-\tau+1} = A_s x_{s,k-\tau} + B_s w_{s,k-\tau}$$

$$x_{s,k-\tau+2} = A_s x_{s,k-\tau+1} + B_s w_{s,k-\tau+1}$$

$$= A_s (A_s x_{s,k-\tau} + B_s w_{s,k-\tau}) + B_s w_{s,k-\tau+1}$$

$$= A_s^2 x_{s,k-\tau} + A_s B_s w_{s,k-\tau} + B_s w_{s,k-\tau+1}$$

$\ldots$

$$x_{s,k} = A_s^\tau x_{s,k-\tau} + \sum_{i=0}^{\tau-1} A_s^i B_s w_{s,k-\tau+i}$$

$$E(x_{s,k} \cdot x_{s,k-\tau}^T) = A_s^\tau E(x_{s,k-\tau} \cdot x_{s,k-\tau}^T) + \sum_{i=0}^{\tau-1} A_s^i B_s E(w_{s,k-\tau+i} \cdot x_{s,k-\tau}^T)$$

$$= A_s^\tau E(x_{s,k-\tau} x_{s,k-\tau}^T)$$

$$y_{s,k-\tau} = C_s \cdot x_{s,k-\tau}$$

$$y_{s,k-\tau+1} = C_s x_{s,k-\tau+1}$$

$$= C_s (A_s x_{s,k-\tau} + B_s w_{s,k-\tau})$$

$$= C_s A_s x_{s,k-\tau} + C_s B_s w_{s,k-\tau}$$

$$y_{s,k} = C_s x_{s,k}$$

$$= C_s \left(A_s^\tau x_{s,k-\tau} + \sum_{i=0}^{\tau-1} A_s^i B_s w_{s,k-\tau+i}\right)$$

$$E(y_{s,k} \cdot y_{s,k-\tau}^T) = C_s \cdot E(x_{s,k} x_{s,k-\tau}^T) C_s^T$$

Wind: $x_{s,k+1} = A_s x_{s,k} + B_s w_{s,k}$

$$y_{s,k} = C_s x_{s,k} + D_s v_{s,k}$$

Cart: $x_{c,k+1} = A_c x_{c,k} + B_c u_k + G_c w_{i,k}$

$$y_{c,k} = C_c x_{c,k} + v_{c,k}$$

Combined model

$$\begin{bmatrix} x_s \\ x_c \end{bmatrix}_{k+1} = \begin{bmatrix} A_s & 0 \\ G_c C_s & A_c \end{bmatrix} \begin{bmatrix} x_s \\ x_c \end{bmatrix}_k + \begin{bmatrix} 0 \\ B_c \end{bmatrix} u_k + \begin{bmatrix} B_s & 0 \\ 0 & G_c D_s \end{bmatrix} \begin{bmatrix} w_{s,k} \\ v_{s,k} \end{bmatrix}$$

$$y_{c,k} = \begin{bmatrix} 0 & C_c \end{bmatrix} \begin{bmatrix} x_s \\ x_c \end{bmatrix}_k + v_{c,k}$$

$$F = \begin{bmatrix} A_s & 0 \\ G_c C_s & A_c \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ B_c \end{bmatrix}, \quad B = \begin{bmatrix} B_s & 0 \\ 0 & G_s D_s \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & C_c \end{bmatrix}$$