# ASEN 5044 - HW 8, Fall 2024, Jash Bhalavat

## Problem 1

①

ASEN 5044
Fall 2024
Jash Bhalavat

$\boxed{\text{HW 8}}$

Problem 1 →

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix}, \quad \Gamma_A = \Gamma_B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad W = q_W \begin{bmatrix} 2 & 0.05 \\ 0.05 & 0.5 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \\ 0 & 0 & 0 & 1 \\ 0 & \Omega & 0 & 0 \end{bmatrix}$$

$$q_W = 10 \left(\frac{m}{s}\right)^2$$

$\Omega_a = 0.045 \frac{rad}{s}$, $\Omega_b = -0.045$ rad/s

$\Delta t = 0.5$ s

Using Van Loan's method:

$A_a$ uses $\Omega_a$
$A_b$ uses $\Omega_b$

$$Z_a = \Delta t \left[\begin{array}{c|c} -A_a & \Gamma_a W \Gamma_a^T \\ \hline 0 & A_a^T \end{array}\right] \xrightarrow[\text{matlab}]{\text{Using}} e^{Z_a} = \left[\begin{array}{c|c} (\cdots) & F_a^{-1} Q_a \\ \hline 0 & F_a^T \end{array}\right] \rightarrow F_a = (F_a^T)^T$$

$$F_a = \begin{bmatrix} 1 & 0.5 & 0 & -0.0056 \\ 0 & 0.9997 & 0 & -0.0225 \\ 0 & 0.0056 & 1 & 0.5 \\ 0 & 0.0225 & 0 & 0.9997 \end{bmatrix}, \quad Q_a = F_a \cdot F_a^{-1} Q_a = \begin{bmatrix} 0.8329 & 2.4983 & 0.0261 & 0.0953 \\ 2.4983 & 9.9931 & 0.0719 & 0.3343 \\ 0.0261 & 0.0719 & 0.2087 & 0.6266 \\ 0.0953 & 0.3343 & 0.6266 & 2.5069 \end{bmatrix}$$

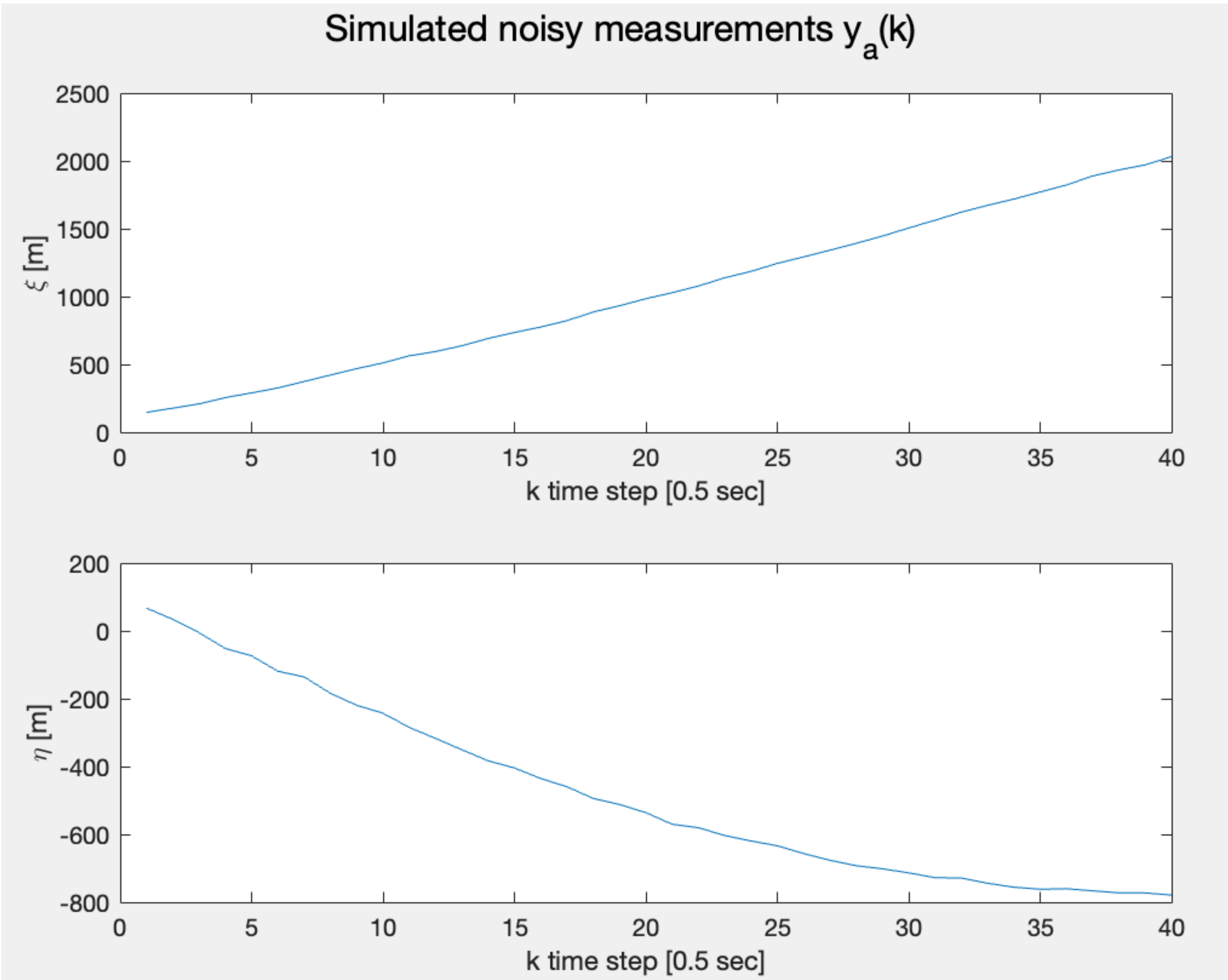
Similarly,

$$Z_b = \Delta t \left[\begin{array}{c|c} -A_b & \Gamma_b W \Gamma_b^T \\ \hline 0 & A_b^T \end{array}\right] \xrightarrow[\text{matlab}]{\text{Using}} e^{Z_b} = \left[\begin{array}{c|c} (\cdots)^{-1} & F_b^{-1} Q_b \\ \hline 0 & F_b^T \end{array}\right] \rightarrow F_b = (F_b^T)^T$$

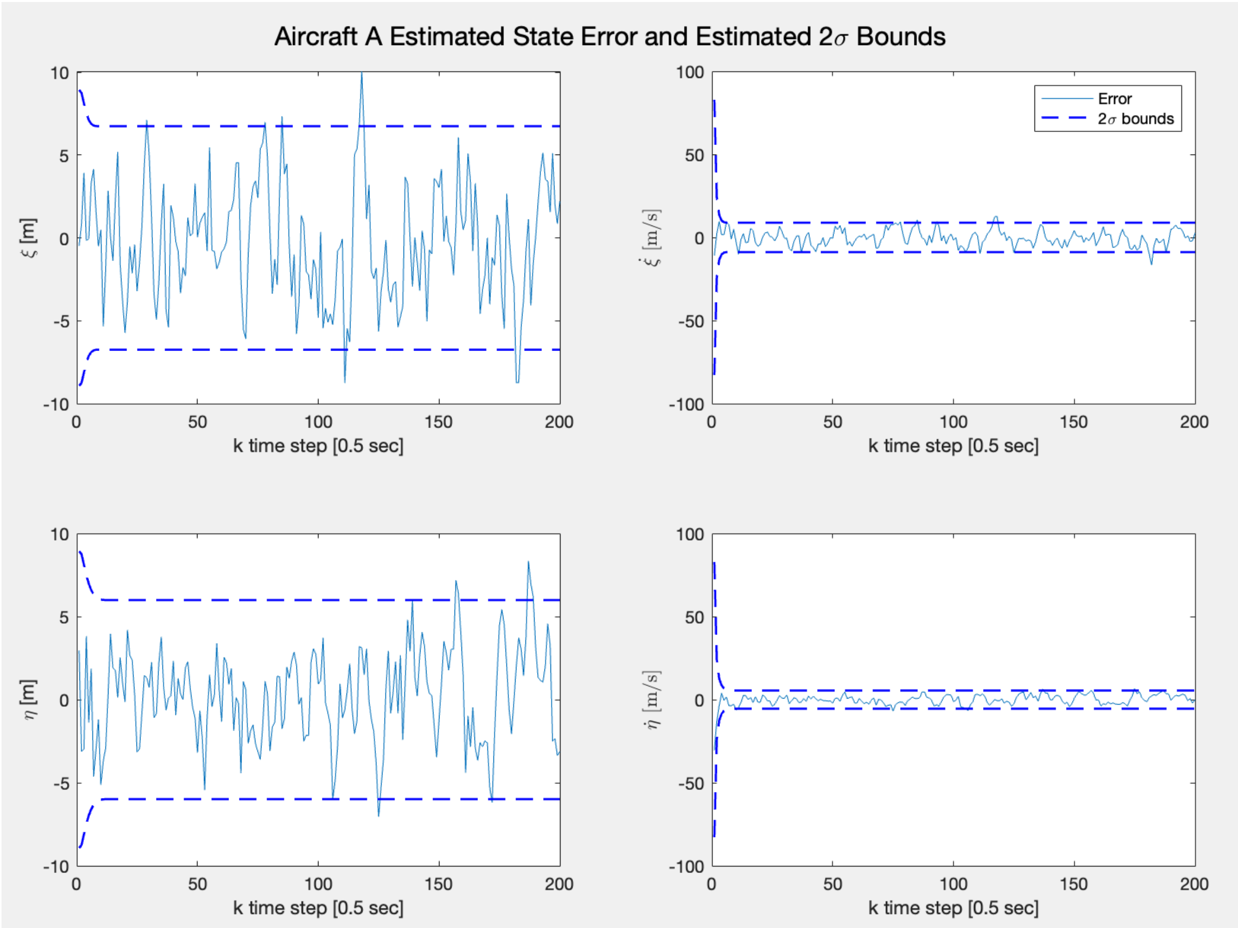
$$F_b = \begin{bmatrix} 1 & 0.5 & 0 & 0.0056 \\ 0 & 0.9997 & 0 & 0.0225 \\ 0 & -0.0056 & 1 & 0.5 \\ 0 & -0.0225 & 0 & 0.9997 \end{bmatrix}, \quad Q_b = F_b \cdot F_b^{-1} Q_b = \begin{bmatrix} 0.8336 & 2.5011 & 0.0156 & 0.0297 \\ 2.5011 & 10.044 & 0.0531 & 0.1656 \\ 0.0156 & 0.0531 & 0.2080 & 0.6238 \\ 0.0297 & 0.1656 & 0.6238 & 2.4956 \end{bmatrix}$$

## Problem 2

- Part a



- Part b




  - The kalman filter output is more certain about the position states (ξ and η) than the remaining two velocity states. That is because both the position states are measured by the ground tracking station monitors while the kalman filter has to predict the velocity states by the dynamics and no measurements. Hence, the kalman filter is more certain about the position states.

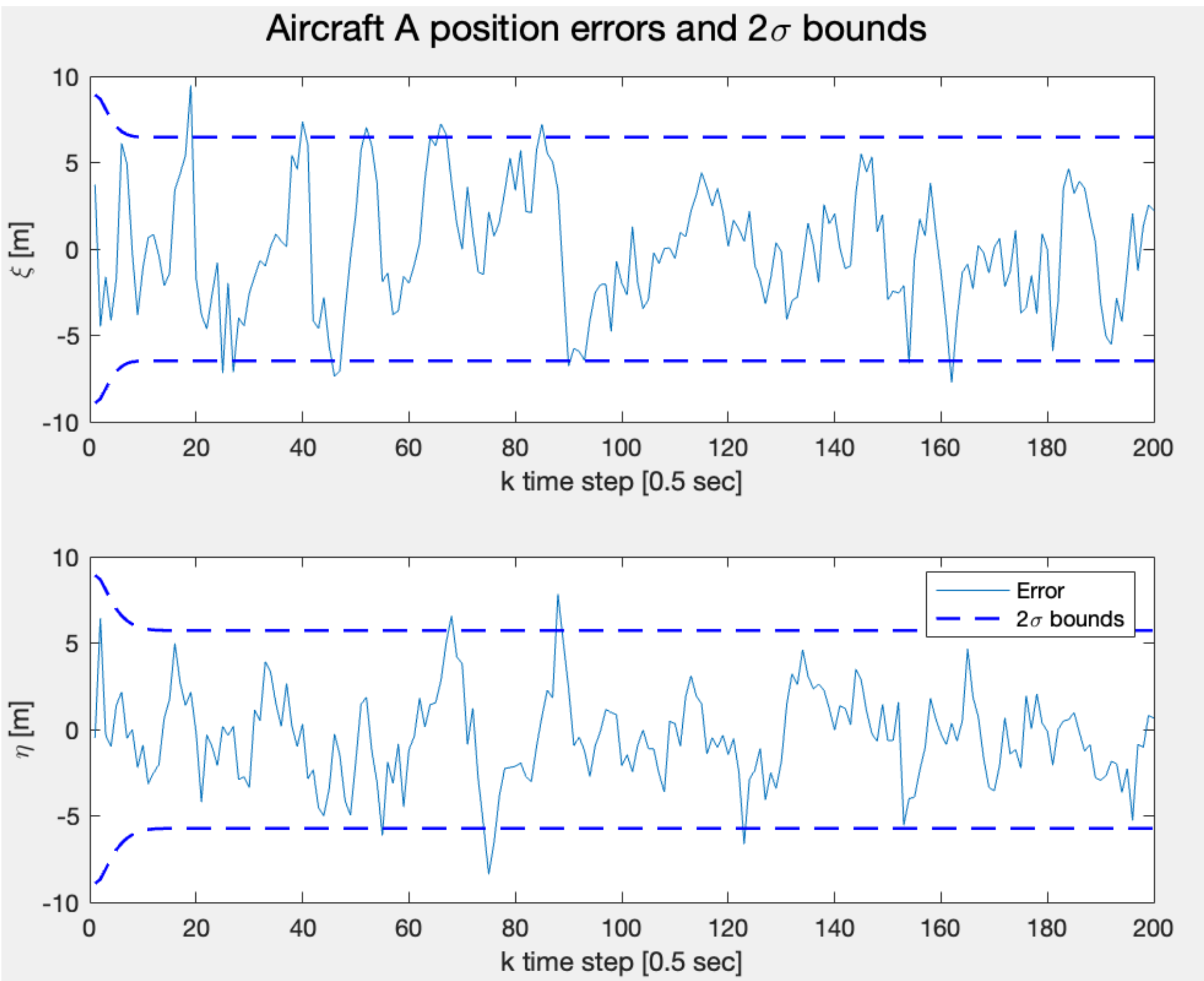
## Problem 3

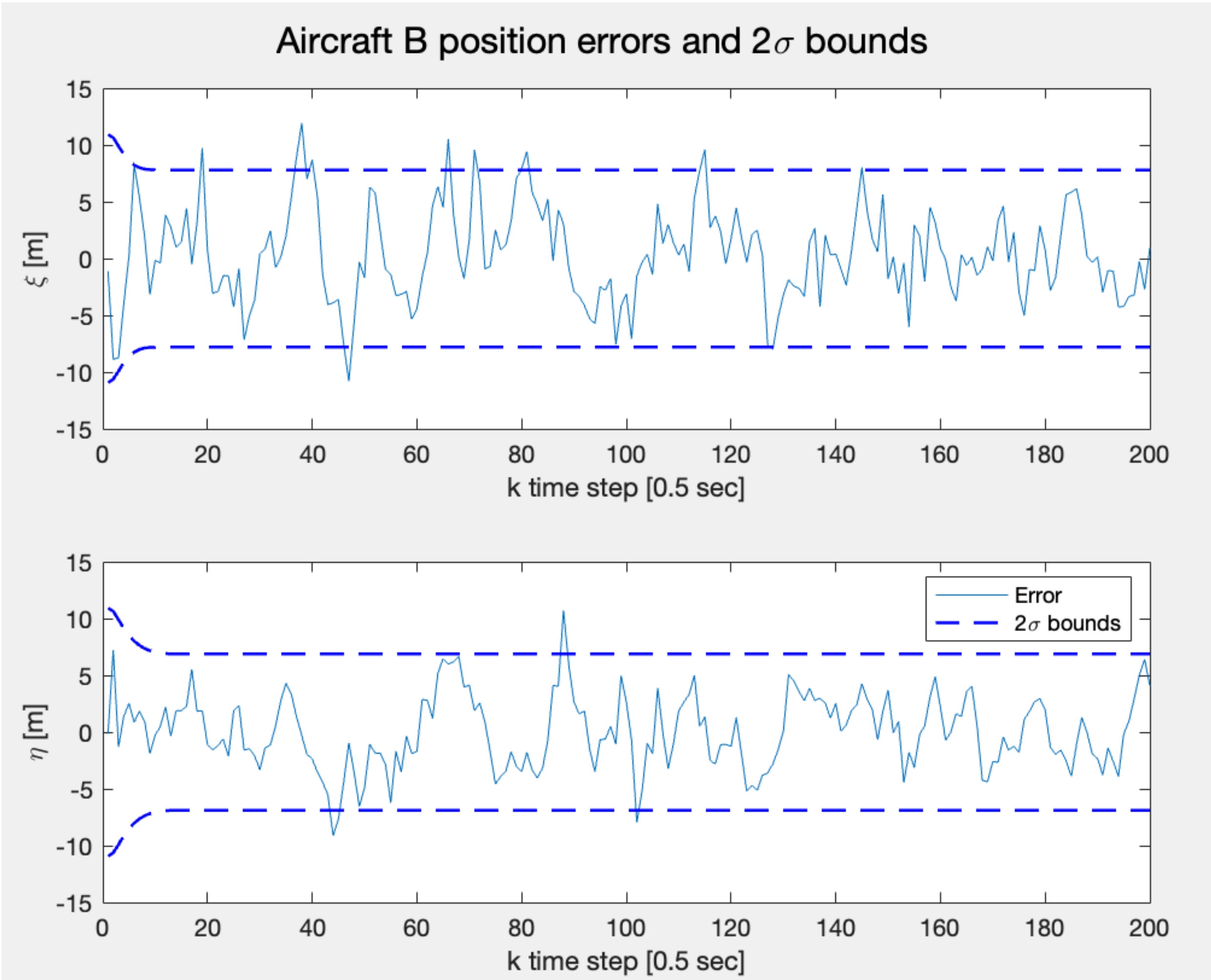
- Part a

Problem 3 → a) $x_s(k) = \begin{bmatrix} x_A(k) \\ x_B(k) \end{bmatrix}$, $F = \begin{bmatrix} F_A & 0 \\ 0 & F_B \end{bmatrix}$, $Q = \begin{bmatrix} Q_A & 0 \\ 0 & Q_B \end{bmatrix}$, $P(0) = \begin{bmatrix} P_A(0) & 0 \\ 0 & P_B(0) \end{bmatrix}$

$H_D = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$, $H_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow H = \begin{bmatrix} H_A & 0 \\ H_D & \end{bmatrix}$

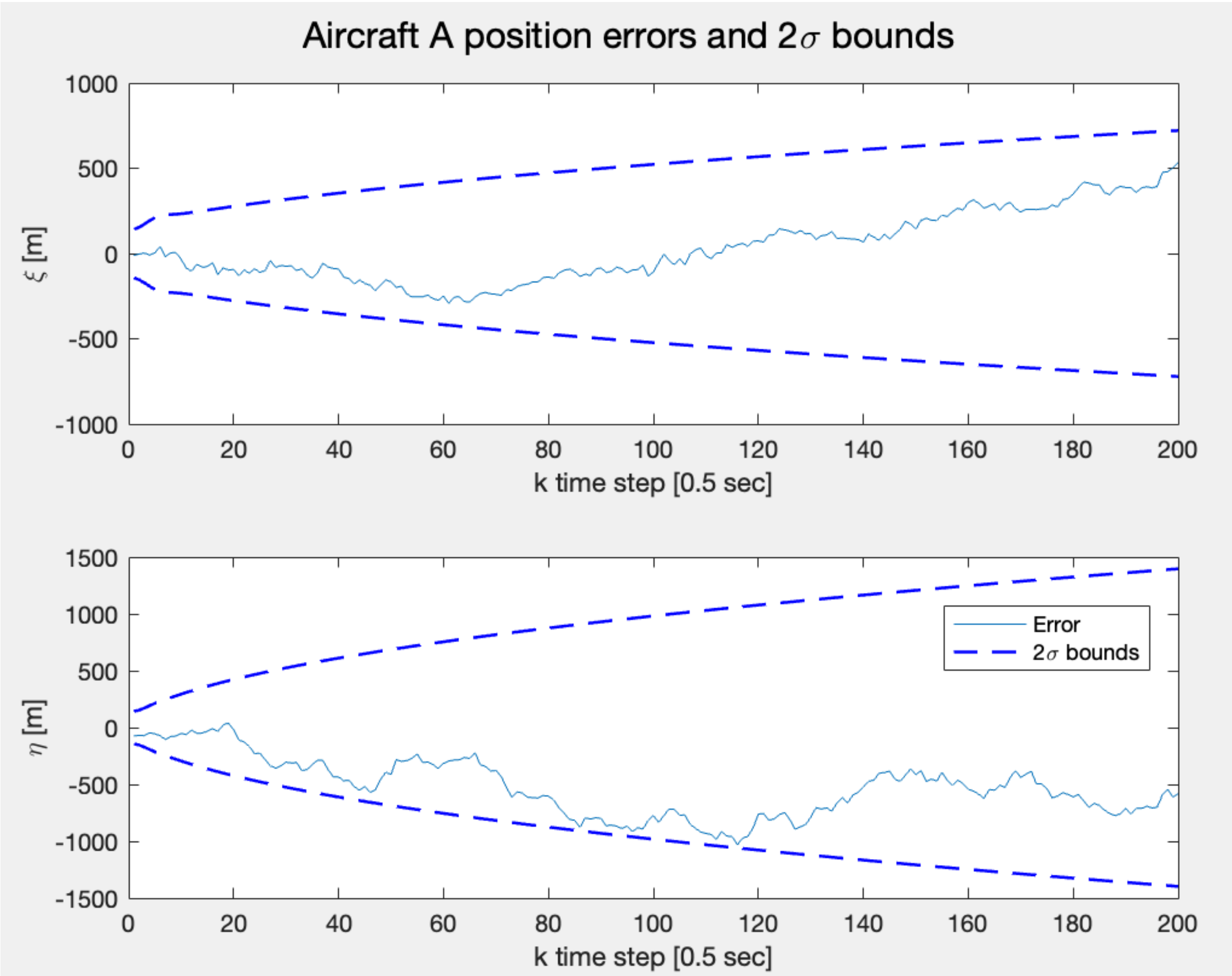
$R_A = \begin{bmatrix} 20 & 0.05 \\ 0.05 & 20 \end{bmatrix}$, $R_D = \begin{bmatrix} 10 & 0.15 \\ 0.15 & 10 \end{bmatrix} \rightarrow R = \begin{bmatrix} R_A & 0 \\ 0 & R_D \end{bmatrix}$



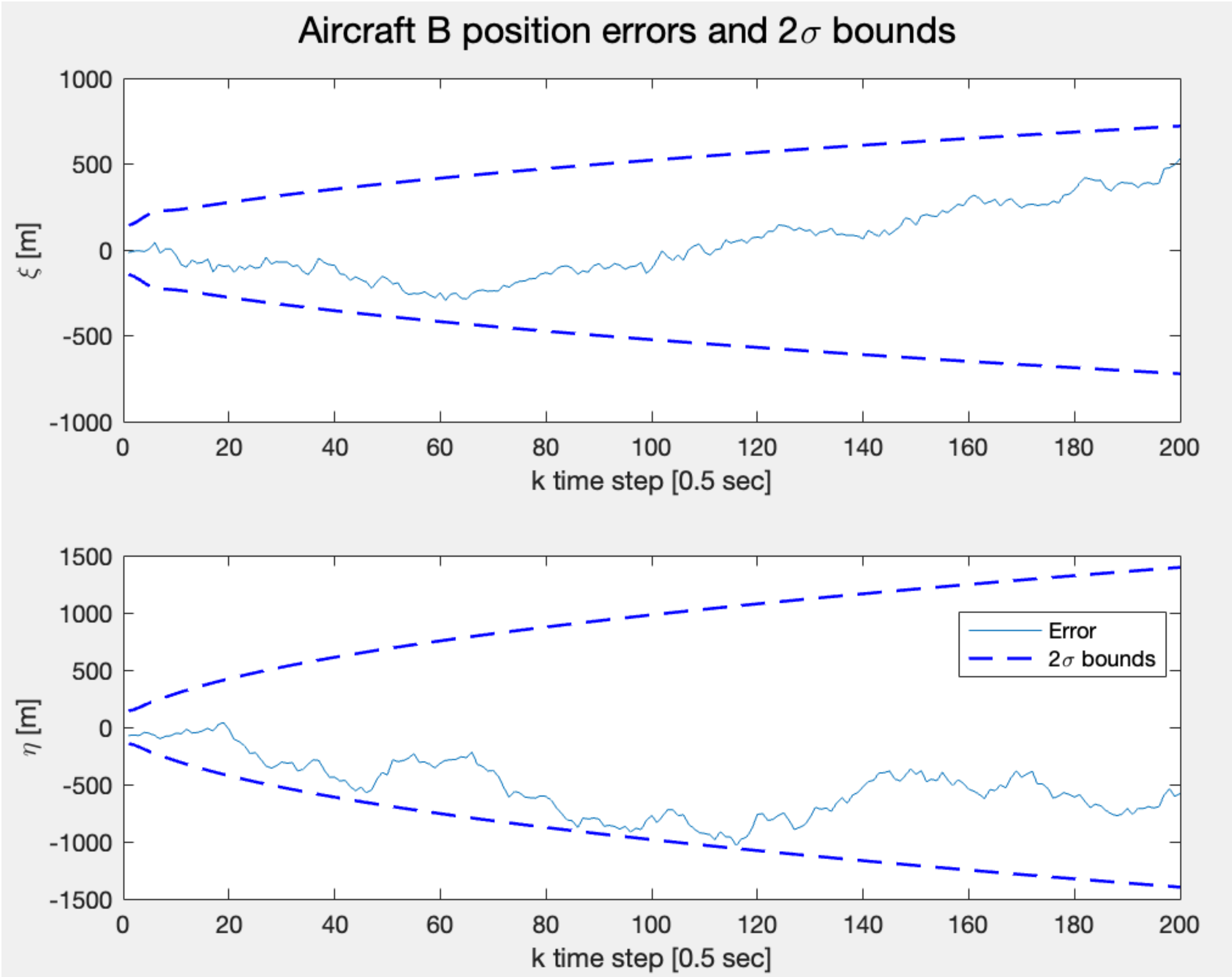


Aircraft B position errors and 2$\sigma$ bounds
15
10
5
0
-5
-10
-15
$\xi$ [m]
0
20
40
60
80
100
120
140
160
180
200
k time step [0.5 sec]
Error
2$\sigma$ bounds
$\eta$ [m]
k time step [0.5 sec]

- Part b



Aircraft A position errors and 2σ bounds
ξ [m]
1000
500
0
-500
-1000
0 20 40 60 80 100 120 140 160 180 200
k time step [0.5 sec]
η [m]
1500
1000
500
0
-500
-1000
-1500
0 20 40 60 80 100 120 140 160 180 200
k time step [0.5 sec]
Error
2σ bounds




- The errors for both the aircrafts are much higher compared to part a. That may be due to the fact that only relative measurements are available such as the difference between the position of the two aircrafts. Additionally, the certainty bounds seem to be increasing with every state indicating that the kalman filter is underconfident.
  - The observability matrix [H, HF, HF^2, HF^3]^T has a rank of 6, so this system is weekly observable, further showing why the uncertainty increases.

- Part c
  - The Kalman Filter covariance matrix should settle down after some initial steps as seen in part a (if the system is fully observable). In contrast, the covariance matrix in a pure prediction system increases, indicating that without measurements it is increasingly uncertain of the prediction of the state.

## Table of Contents



```
clear; clc; close all;

% ASEN 5044 - HW 8
% Fall 2024
% Jash Bhalavat
```

# Problem 1

Given

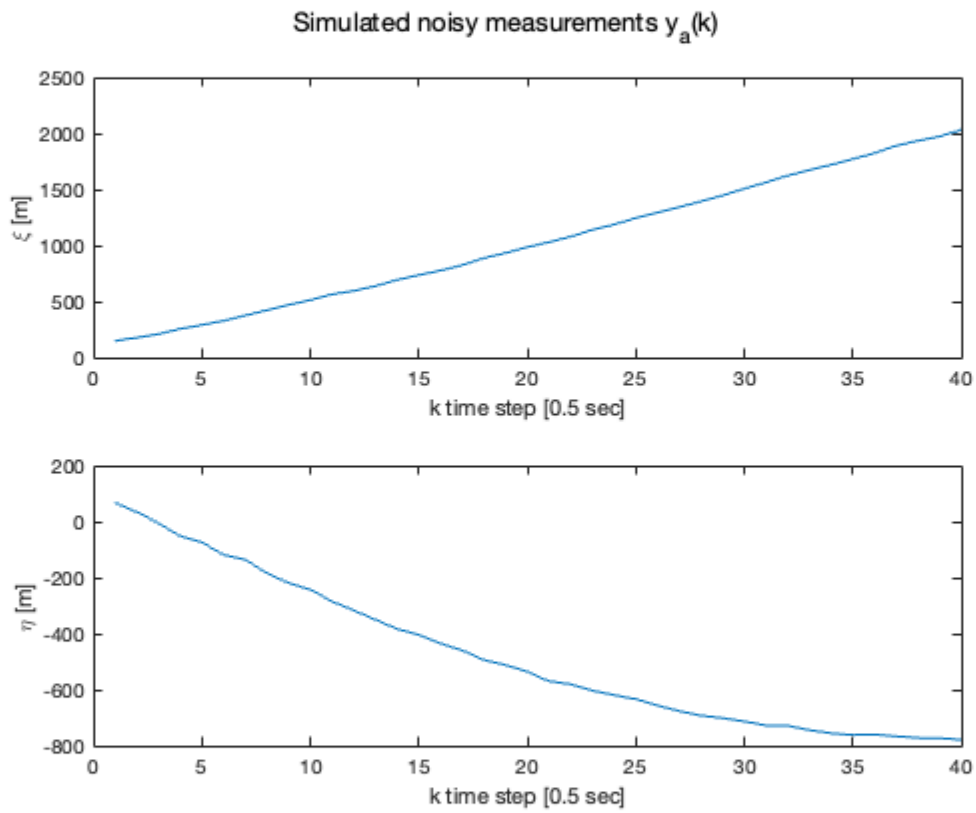
```
delta_t = 0.5;
omega_a = 0.045;
odt_a = delta_t*omega_a;
omega_b = -0.045;
odt_b = delta_t*omega_b;

A_a = [0 1 0 0; 0 0 0 -omega_a; 0 0 0 1; 0 omega_a 0 0];
A_b = [0 1 0 0; 0 0 0 -omega_b; 0 0 0 1; 0 omega_b 0 0];

n = length(A_a);

% Construct F_a, F_b matrices
F_a = [1 sin(odt_a)/omega_a 0 -(1-cos(odt_a))/omega_a;
        0 cos(odt_a) 0 -sin(odt_a);
        0 (1-cos(odt_a))/omega_a 1 sin(odt_a)/omega_a;
        0 sin(odt_a) 0 cos(odt_a)];

F_b = [1 sin(odt_b)/omega_b 0 -(1-cos(odt_b))/omega_b;
        0 cos(odt_b) 0 -sin(odt_b);
        0 (1-cos(odt_b))/omega_b 1 sin(odt_b)/omega_b;
        0 sin(odt_b) 0 cos(odt_b)];

q_omega = 10;
W = q_omega*[2 0.05; 0.05 0.5];

gamma_a = [0 0; 1 0; 0 0; 0 1];
gamma_b = [0 0; 1 0; 0 0; 0 1];

Z_a = delta_t * [-A_a gamma_a*W*gamma_a'; zeros(n), A_a'];
Z_b = delta_t * [-A_b gamma_b*W*gamma_b'; zeros(n), A_b'];

e_z_a = expm(Z_a);
e_z_b = expm(Z_b);
```

```
F_inv_Q_a = e_z_a(1:4, 5:8);
F_inv_Q_b = e_z_b(1:4, 5:8);
F_a_t = e_z_a(5:8, 5:8);
F_b_t = e_z_b(5:8, 5:8);

Q_a = F_a_t' * F_inv_Q_a;
Q_b = F_b_t' * F_inv_Q_b;
```

# Problem 2

```
rng(100);
H = [1 0 0 0; 0 0 1 0];
R_a = [20 0.05; 0.05 20];

data = load("hw8problemdata.mat");
x_a_single_truth = data.xasingle_truth;

p = size(H,1);
% Subtracting 1 because x_a_single_truth starts at 0
T = size(x_a_single_truth, 2) - 1;

% Part a
S_v_a = chol(R_a, 'lower');

% Necessary variables
I_p = eye(p);
zeros_p = zeros(p,1);

for i = 1:T
    q_k_a = mvnrnd(zeros_p, I_p)';
    % Using x(:,i+1) because x starts at 0
    y_a_k(:,i) = H*x_a_single_truth(:,i+1) + S_v_a*q_k_a;
end

k_20_sec = 1:40;

figure()
subplot(2,1,1)
plot(k_20_sec, y_a_k(1,1:40))
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")
% hold on
% plot(time_20_sec, x_a_single_truth(1,2:41))
% hold off

subplot(2,1,2)
plot(k_20_sec, y_a_k(2,1:40))
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")
% hold on
% plot(time_20_sec, x_a_single_truth(3,2:41))
```

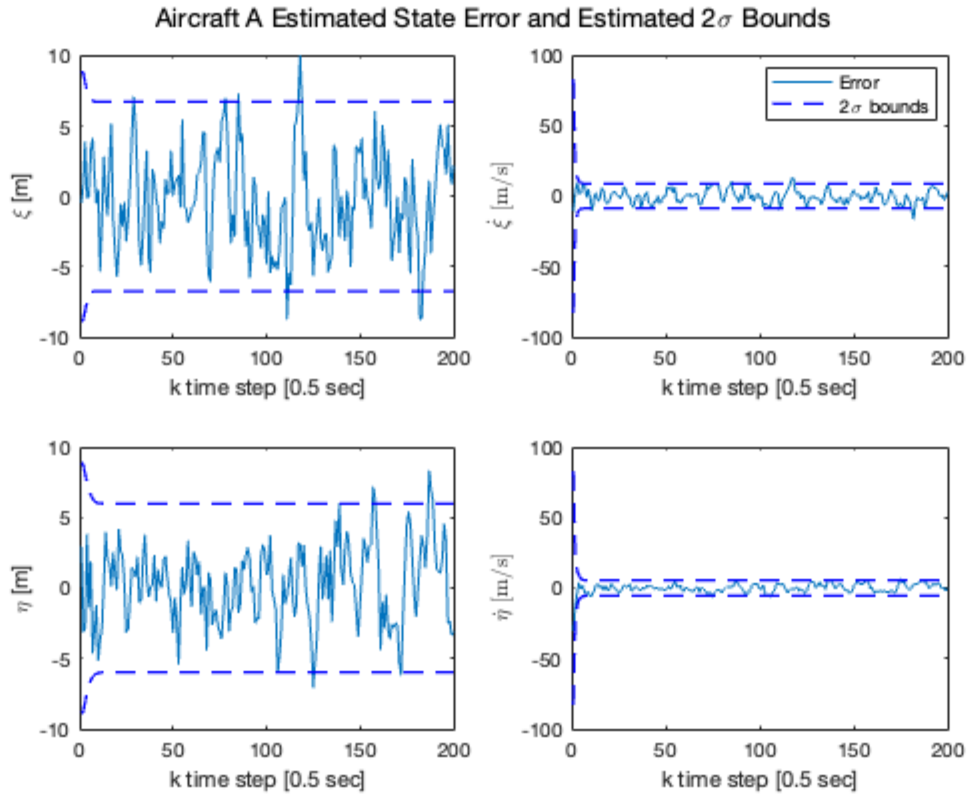
```
% hold off
sgtitle("Simulated noisy measurements y_a(k)")
```




# Part b

```
mu_a_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
P_a_0 = 900 * diag([10, 2, 10, 2]);

x_a_k = mu_a_0;
P_a_k = P_a_0;

Qkf_a = Q_a;
Rkf_a = R_a;

tvec = 1:T;

G_a = zeros(4,1);
u_a = zeros(1,T);

[x_a_kf, P_a_kf] = kalman_filter_hw8(tvec, F_a, G_a, x_a_k, u_a, P_a_k,
Qkf_a, Rkf_a, y_a_k, H);

figure()
subplot(2,2,1)
plot(tvec, x_a_single_truth(1,2:end)-x_a_kf(1,:))
hold on
```

```
plot(tvec,2*sqrt(squeeze(P_a_kf(1,1,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_a_kf(1,1,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")

subplot(2,2,2)
plot(tvec, x_a_single_truth(2,2:end)-x_a_kf(2,:))
hold on
plot(tvec,2*sqrt(squeeze(P_a_kf(2,2,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_a_kf(2,2,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel('$\dot{\xi}$ [m/s]', 'Interpreter','latex')
legend("Error", "2\sigma bounds")

subplot(2,2,3)
plot(tvec, x_a_single_truth(3,2:end)-x_a_kf(3,:))
hold on
plot(tvec,2*sqrt(squeeze(P_a_kf(3,3,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_a_kf(3,3,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")

subplot(2,2,4)
plot(tvec, x_a_single_truth(4,2:end)-x_a_kf(4,:))
hold on
plot(tvec,2*sqrt(squeeze(P_a_kf(4,4,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_a_kf(4,4,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel('$\dot{\eta}$ [m/s]', 'Interpreter','latex')
sgtitle("Aircraft A Estimated State Error and Estimated 2\sigma Bounds")
```




*Published with MATLAB® R2024a*

# Table of Contents



```
clear; clc; close all;
rng(100);

% ASEN 5044 - HW 8 Problem 3
% Fall 2024, Jash Bhalavat

% From problem 1
% Given
delta_t = 0.5;
omega_a = 0.045;
odt_a = delta_t*omega_a;
omega_b = -0.045;
odt_b = delta_t*omega_b;

A_a = [0 1 0 0; 0 0 0 -omega_a; 0 0 0 1; 0 omega_a 0 0];
A_b = [0 1 0 0; 0 0 0 -omega_b; 0 0 0 1; 0 omega_b 0 0];

n = length(A_a);

% Construct F_a, F_b matrices
F_a = [1 sin(odt_a)/omega_a 0 -(1-cos(odt_a))/omega_a;
        0 cos(odt_a) 0 -sin(odt_a);
        0 (1-cos(odt_a))/omega_a 1 sin(odt_a)/omega_a;
        0 sin(odt_a) 0 cos(odt_a)];

F_b = [1 sin(odt_b)/omega_b 0 -(1-cos(odt_b))/omega_b;
        0 cos(odt_b) 0 -sin(odt_b);
        0 (1-cos(odt_b))/omega_b 1 sin(odt_b)/omega_b;
        0 sin(odt_b) 0 cos(odt_b)];

q_omega = 10;
W = q_omega*[2 0.05; 0.05 0.5];

gamma_a = [0 0; 1 0; 0 0; 0 1];
gamma_b = [0 0; 1 0; 0 0; 0 1];

Z_a = delta_t * [-A_a gamma_a*W*gamma_a'; zeros(n), A_a'];
Z_b = delta_t * [-A_b gamma_b*W*gamma_b'; zeros(n), A_b'];

e_z_a = expm(Z_a);
e_z_b = expm(Z_b);

F_inv_Q_a = e_z_a(1:4, 5:8);
F_inv_Q_b = e_z_b(1:4, 5:8);
F_a_t = e_z_a(5:8, 5:8);
```

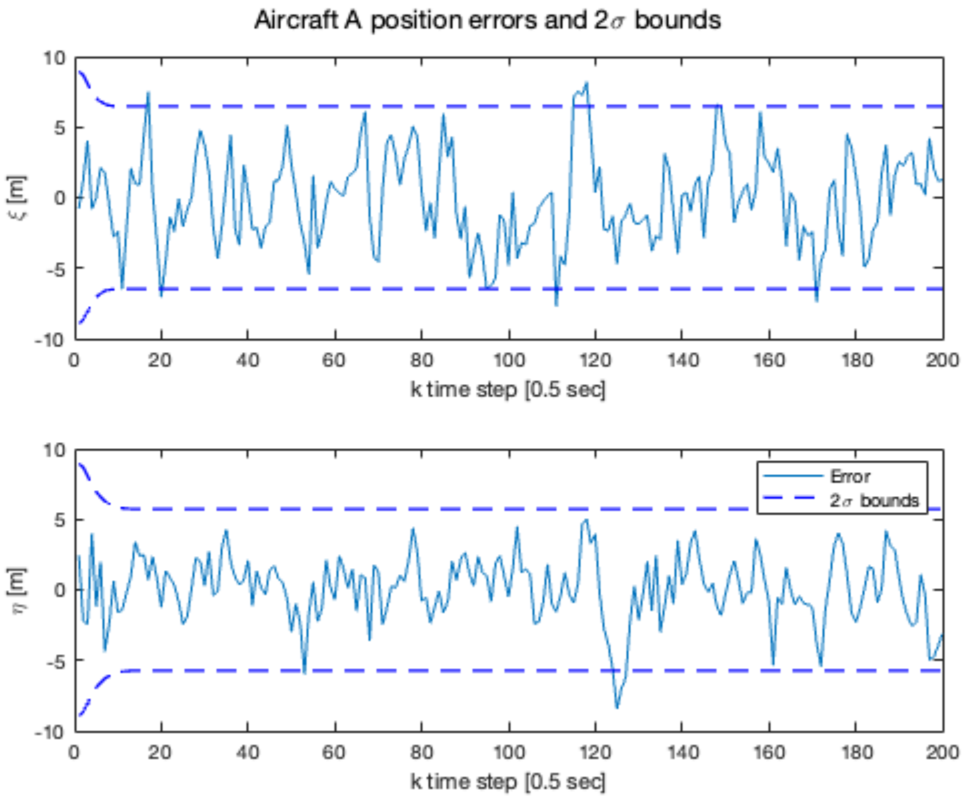
```
F_b_t = e_z_b(5:8, 5:8);

Q_a = F_a_t' * F_inv_Q_a;
Q_b = F_b_t' * F_inv_Q_b;
```

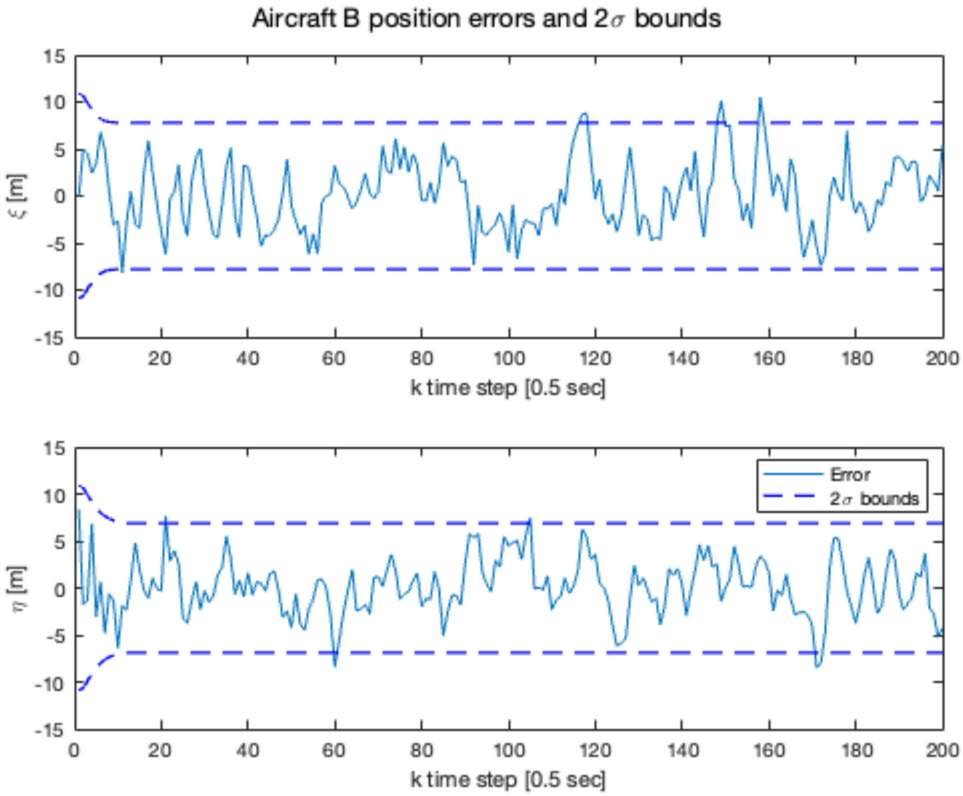
# Problem 3 Part a

```
data = load("hw8problemdata.mat");
x_a_double_truth = data.xadouble_truth;
x_b_double_truth = data.xbdouble_truth;

% Simulate noisy measurements for a'
H_a = [1 0 0 0; 0 0 1 0];
R_a = [20 0.05; 0.05 20];

p = size(H_a,1);
% Subtracting 1 because x_a_single_truth starts at 0
T = size(x_a_double_truth, 2) - 1;

% Part a
S_v_a = chol(R_a, 'lower');

% Necessary variables
I_p = eye(p);
zeros_p = zeros(p,1);

for i = 1:T
    q_k_a = mvnrnd(zeros_p, I_p)';
    % Using x(:,i+1) because x starts at 0
    y_a_k(:,i) = H_a*x_a_double_truth(:,i+1) + S_v_a*q_k_a;
end

% Simulate y_d noisy measurements
x_truth = [x_a_double_truth; x_b_double_truth];
H_d = [1 0 0 0 -1 0 0 0; 0 0 1 0 0 0 -1 0];
R_d = [10 0.15; 0.15 10];

S_v_d = chol(R_d, 'lower');

for i = 1:T
    q_k_d = mvnrnd(zeros_p, I_p)';
    y_d_k(:,i) = H_d*x_truth(:,i+1) + S_v_d*q_k_d;
end

y_s = [y_a_k; y_d_k];

tvec = 1:T;

% figure()
% plot(tvec, y_d_k(1,:))
% hold on
% plot(tvec, x_a_double_truth(1,2:end)-x_b_double_truth(1,2:end))
% hold off
```

```
mu_a_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
P_a_0 = 900 * diag([10, 2, 10, 2]);
mu_b_0 = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)];
P_b_0 = 900 * diag([11, 4, 11, 4]);

F = [F_a, zeros(4,4); zeros(4,4), F_b];
G = zeros(8,1);
u = zeros(1,T);

xk = [mu_a_0; mu_b_0];
Pk = [P_a_0 zeros(4,4); zeros(4,4) P_b_0];

Qkf = [Q_a, zeros(4,4); zeros(4,4) Q_b];
Rkf = [R_a, zeros(2,2); zeros(2,2) R_d];

H_s = [H_a, zeros(2,4); H_d];

[x_kf, P_kf] = kalman_filter_hw8(tvec, F, G, xk, u, Pk, Qkf, Rkf, y_s, H_s);

figure()
subplot(2,1,1)
plot(tvec, x_a_double_truth(1,2:end)-x_kf(1,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf(1,1,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf(1,1,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")

subplot(2,1,2)
plot(tvec, x_a_double_truth(3,2:end)-x_kf(3,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf(3,3,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf(3,3,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")
sgtitle("Aircraft A position errors and 2\sigma bounds")
legend("Error", "2\sigma bounds")

figure()
subplot(2,1,1)
plot(tvec, x_b_double_truth(1,2:end)-x_kf(5,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf(5,5,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf(5,5,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")

subplot(2,1,2)
plot(tvec, x_b_double_truth(3,2:end)-x_kf(7,:))
hold on
```

```
plot(tvec,2*sqrt(squeeze(P_kf(7,7,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf(7,7,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")
sgtitle("Aircraft B position errors and 2\sigma bounds")
legend("Error", "2\sigma bounds")
```

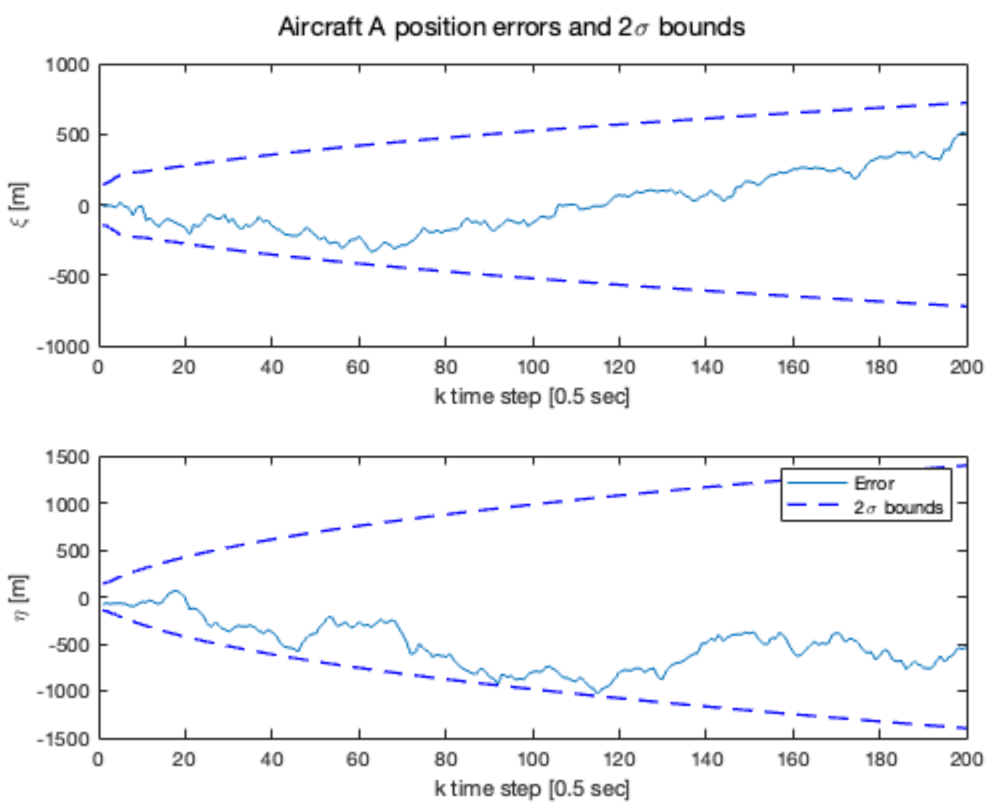


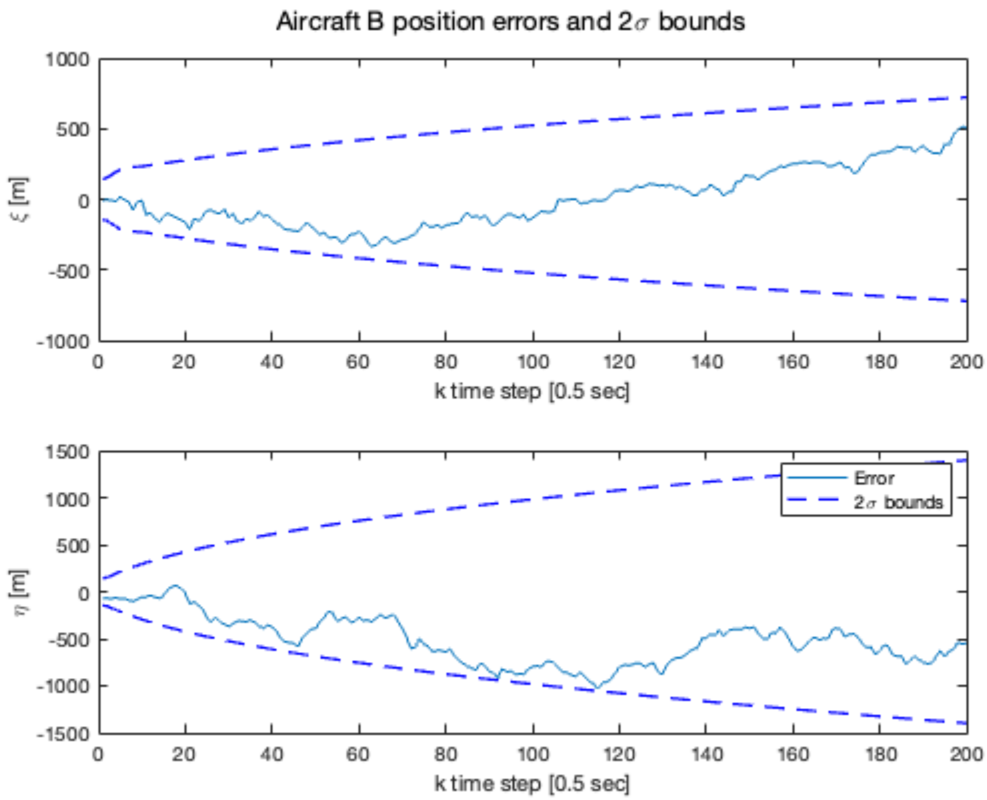



# Problem 3 Part b

```
[x_kf_partb, P_kf_partb] = kalman_filter_hw8(tvec, F, G, xk, u, Pk, Qkf, R_d,
y_d_k, H_d);

figure()
subplot(2,1,1)
plot(tvec, x_a_double_truth(1,2:end)-x_kf_partb(1,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf_partb(1,1,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf_partb(1,1,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")

subplot(2,1,2)
plot(tvec, x_a_double_truth(3,2:end)-x_kf_partb(3,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf_partb(3,3,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf_partb(3,3,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")
sgtitle("Aircraft A position errors and 2\sigma bounds")
legend("Error", "2\sigma bounds")
```

```
figure()
subplot(2,1,1)
plot(tvec, x_b_double_truth(1,2:end)-x_kf_partb(5,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf_partb(5,5,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf_partb(5,5,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\xi [m]")

subplot(2,1,2)
plot(tvec, x_b_double_truth(3,2:end)-x_kf_partb(7,:))
hold on
plot(tvec,2*sqrt(squeeze(P_kf_partb(7,7,:))'),'b--','LineWidth',1.25)
plot(tvec,-2*sqrt(squeeze(P_kf_partb(7,7,:))'),'b--','LineWidth',1.25)
hold off
xlabel("k time step [0.5 sec]")
ylabel("\eta [m]")
sgtitle("Aircraft B position errors and 2\sigma bounds")
legend("Error", "2\sigma bounds")
```






*Published with MATLAB® R2024a*

# Table of Contents



```
function [xk_filt_hist, Pk_filt_hist] = kalman_filter_hw8(tvec, F, G, xk, u,
Pk, Qkf, Rkf, ykhist, H)
    % Linearized Kalman Filter as presented in Lec26, Slide 6 of ASEN 5044,
    % Fall 2024 and copied from L26_1Drobotstatefilter.m code
    % Inputs
    % tvec - vector of time steps where T is total time steps [Tx1]
    % F - DT STM matrix [nxn]
    % G - DT control matrix [nxm] where m is total number of control inputs
    % xk - initial xk [nx1]
    % u - DT control vector [mxT]
    % Pk - initial Pk [nxn]
    % Qkf - Kalman Filter Process noise [nxn]
    % Rkf - Kalman Filter measurement covariance [pxp]
    % ykhist - Measurement history [pxT]
    % H - DT measurement matrix [pxn]
    %
    % Outputs
    % xk_filt_hist - Kalman filter estimation states [nxT]
    % Pk_filt_hist - Kalman filter estimation state covar [nxnxT]

    n = length(F);

    for k=1:length(tvec)

        %%Perform prediction step
        xkp1_minus = F*xk + G*u(:,k);
        Pkp1_minus = F*Pk*F' + Qkf;

        %%Compute Kalman gain
        Kkp1 = Pkp1_minus*H'*inv(H*Pkp1_minus*H' + Rkf);

        %%Perform measurement update step
        ykp1_report = ykhist(:,k); %pull report of actual data from sensor
        ykp1_pred = H*xkp1_minus; %predicted measurement
        innov_kp1 = ykp1_report - ykp1_pred; %compute innovation
        xkp1_plus = xkp1_minus + Kkp1*innov_kp1; %compute update to state mean
        Pkp1_plus = (eye(n) - Kkp1*H)*Pkp1_minus; %compute update to covar

        %%store results and cycle for next iteration
        xk = xkp1_plus;
        xk_filt_hist(:,k) = xkp1_plus;
        Pk = Pkp1_plus;
        Pk_filt_hist(:,:,k)=Pkp1_plus;
    end
end
```

```
Not enough input arguments.
```