## Table of Contents

Clean ........................................ 1
Problem 1 ........................................ 1
Problem 2 ........................................ 2
Problem 3 ........................................ 7

# Clean

```
close all; clear; clc;
```

# Problem 1

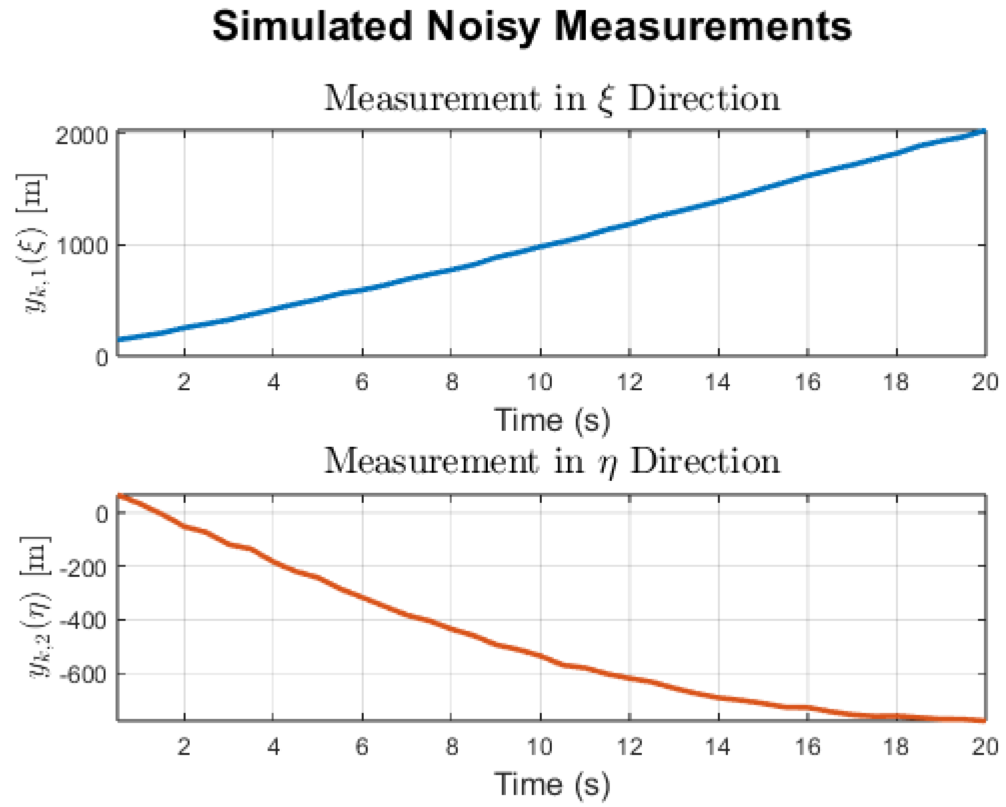
```
sigma_a = 0.045;
sigma_b = -0.045;
qw = 10;
dt = 0.5;
W = qw .* [2, 0.05; 0.05, 0.5];

gamma = [0,0;1,0;0,0;0,1];

Aa = [0,1,0,0;
    0,0,0,-sigma_a;
    0,0,0,1;
    0,sigma_a,0,0];

Ab = [0,1,0,0;
    0,0,0,-sigma_b;
    0,0,0,1;
    0,sigma_b,0,0];

Fa = [1, sin(sigma_a*dt)/sigma_a, 0, -(1-cos(sigma_a*dt))/sigma_a;
    0, cos(sigma_a*dt), 0, -sin(sigma_a*dt);
    0, (1-cos(sigma_a*dt))/sigma_a, 1, sin(sigma_a*dt)/sigma_a;
    0, sin(sigma_a*dt), 0, cos(sigma_a*dt)];

Fb = [1, sin(sigma_b*dt)/sigma_b, 0, -(1-cos(sigma_b*dt))/sigma_b;
    0, cos(sigma_b*dt), 0, -sin(sigma_b*dt);
    0, (1-cos(sigma_b*dt))/sigma_b, 1, sin(sigma_b*dt)/sigma_b;
    0, sin(sigma_b*dt), 0, cos(sigma_b*dt)];

% Van loans method
za = dt .* [-Aa, gamma*W*gamma'; zeros(4,4), Aa'];
ez_a = expm(za);
Qa = ez_a(5:8, 5:8)' * ez_a(1:4, 5:8);

zb = dt .* [-Ab, gamma*W*gamma'; zeros(4,4), Ab'];
ez_b = expm(zb);
Qb = ez_b(5:8, 5:8)' * ez_b(1:4, 5:8);
```

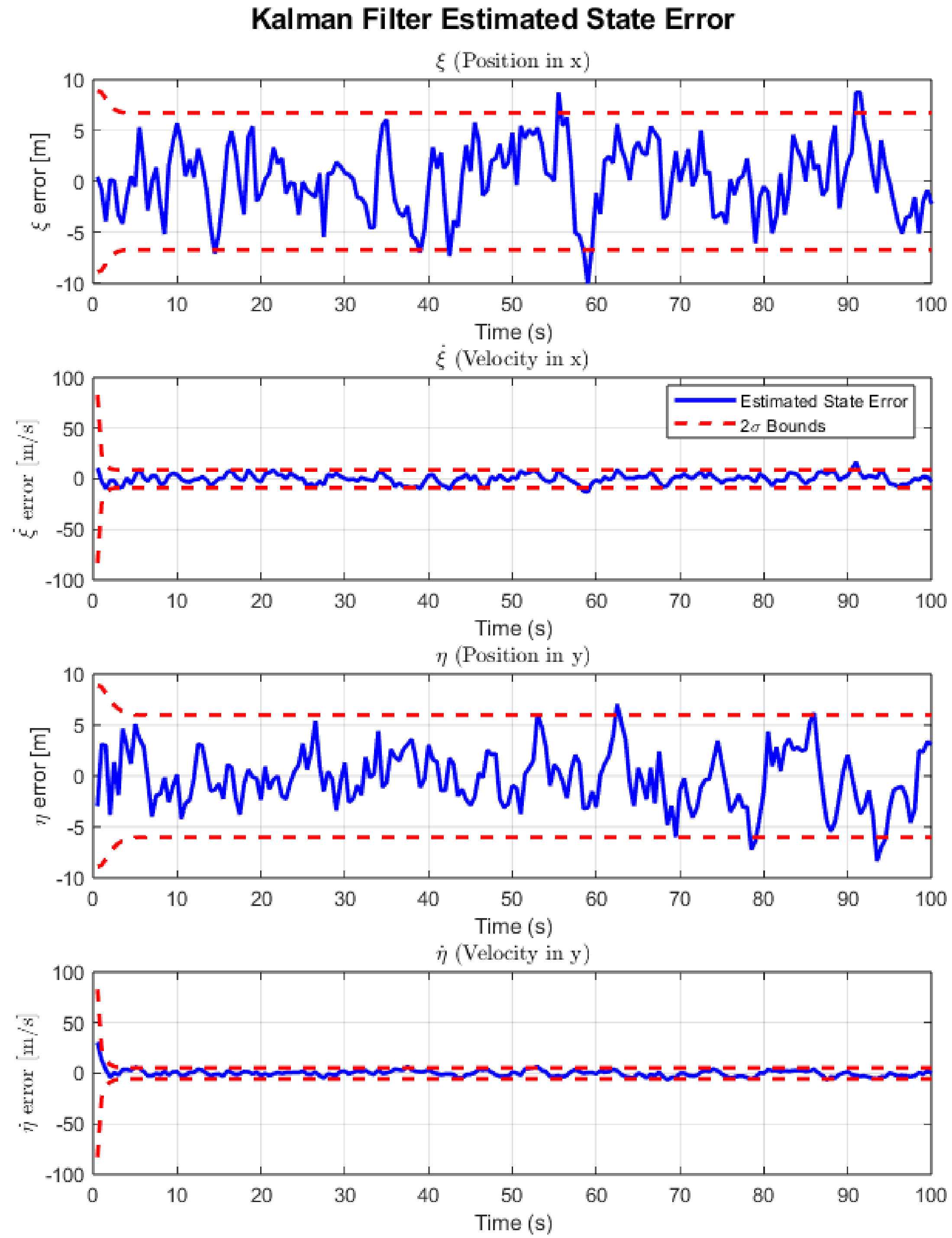
# Problem 2

```
rng(100);

% Load data, xadouble_truth, xasingle_truth, xbdouble_truth
load('hw8problemdata.mat');

% Measurement model
H = [1,0,0,0; 0,0,1,0];
Ra = [20, 0.05; 0.05, 20];

% Simulate noisy measurements
time_vec = 0.5:0.5:100;
vk_mat = zeros(2, 200);
for i = 2:length(xasingle_truth(1,:))
    pos_xa = H * xasingle_truth(:,i);
    vk_mat(:,i-1) = mvnrnd(pos_xa', Ra, 1);
end

% Try using the cholesky deomposition
Sv = chol(Ra, "lower");
vk_mat_chol = zeros(2, 200);
for i = 2:length(xasingle_truth(1,:))
    pos_xa = H * xasingle_truth(:,i);
    q = randn(2,1);
    vk_mat_chol(:,i-1) = pos_xa + Sv*q;
end
%vk_mat = vk_mat_chol;


% Plot for the first 20 seconds
time_20 = time_vec(time_vec <= 20);
vk_mat_20 = vk_mat(:,time_vec <= 20);


% Kalman filter initialization
mu_a_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
Pa_0 = 900 .* diag([10, 2, 10, 2]);
xa_meas = mu_a_0;
Pa_meas = Pa_0;

% Recursive kalman filter loop
xhat_mat = zeros(4, length(vk_mat(1,:)));
sigma_mat = zeros(4, length(vk_mat(1,:)));
for i = 1:length(vk_mat(1,:))
    % Time update
    xa_pred = Fa*xa_meas;
    Pa_pred = Fa*Pa_meas*Fa' + Qa;
    K_gain = Pa_pred * H' * inv(H * Pa_pred * H' + Ra);

    % Measurement update
    xa_meas = xa_pred + K_gain*(vk_mat(:,i) - H * xa_pred);
    Pa_meas = (eye(4) - K_gain*H) * Pa_pred;
    % Store the estimates and uncertainties
    xhat_mat(:,i) = xa_meas;
    sigma_mat(:,i) = [sqrt(Pa_meas(1,1)); sqrt(Pa_meas(2,2));
sqrt(Pa_meas(3,3)); sqrt(Pa_meas(4,4))];
end


% Calculate the estimated state error
est_error = xhat_mat - xasingle_truth(:,2:end);

%%%% NOISY MEASUREMENTS PLOT
figure();
% Title for the entire figure
sgtitle('Simulated Noisy Measurements', 'FontSize', 16, 'FontWeight',
'bold');

% First subplot
subplot(2, 1, 1);
plot(time_20, vk_mat_20(1, :), 'LineWidth', 2, 'Color', [0 0.4470 0.7410]);
% Use MATLAB default blue
grid on;
xlabel('Time (s)', 'FontSize', 12);
ylabel('$y_{k,1} (\xi)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
title('Measurement in $\xi$ Direction', 'Interpreter', 'latex', 'FontSize',
14);
xlim([min(time_20), max(time_20)]); % Set consistent x-axis limits

% Second subplot
subplot(2, 1, 2);
plot(time_20, vk_mat_20(2, :), 'LineWidth', 2, 'Color', [0.8500 0.3250
0.0980]); % Use MATLAB default red
grid on;
xlabel('Time (s)', 'FontSize', 12);
ylabel('$y_{k,2} (\eta)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
title('Measurement in $\eta$ Direction', 'Interpreter', 'latex', 'FontSize',
14);
xlim([min(time_20), max(time_20)]); % Set consistent x-axis limits

% Additional formatting
set(gcf, 'Color', 'w'); % Set background color to white


%%%% Kalman filter estimated state error plot
mu_mat = xhat_mat';
unc_mat = sigma_mat';
figure();

% Set line properties for clarity
mainLineColor = 'b';
boundLineColor = 'r';
lineWidth = 2;

% Main title for the figure
```

```
sgtitle('Kalman Filter Estimated State Error', 'FontSize', 14, 'FontWeight',
'bold');

% xi
subplot(4,1,1);
plot(time_vec, est_error(1,:), 'LineWidth', lineWidth, 'Color',
mainLineColor);
hold on;
grid on;
plot(time_vec, 2 .* unc_mat(:,1), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
plot(time_vec, -2 .* unc_mat(:,1), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('\xi error [m]');
title('$\xi$ (Position in x)', 'Interpreter','latex');

% xi dot
subplot(4,1,2);
plot(time_vec, est_error(2,:), 'LineWidth', lineWidth, 'Color',
mainLineColor);
hold on;
grid on;
plot(time_vec, 2 .* unc_mat(:,2), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
plot(time_vec, -2 .* unc_mat(:,2), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('$\dot{\xi}$ error [m/s]', 'Interpreter', 'latex');
title('$\dot{\xi}$ (Velocity in x)', 'Interpreter','latex');
legend('Estimated State Error', '2\sigma Bounds', 'Location', 'best');

% eta
subplot(4,1,3);
plot(time_vec, est_error(3,:), 'LineWidth', lineWidth, 'Color',
mainLineColor);
hold on;
grid on;
plot(time_vec, 2 .* unc_mat(:,3), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
plot(time_vec, -2 .* unc_mat(:,3), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('\eta error [m]');
title('$\eta$ (Position in y)', 'Interpreter','latex');

% eta dot
subplot(4,1,4);
plot(time_vec, est_error(4,:), 'LineWidth', lineWidth, 'Color',
mainLineColor);
hold on;
grid on;
plot(time_vec, 2 .* unc_mat(:,4), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
```

```
plot(time_vec, -2 .* unc_mat(:,4), 'Color', boundLineColor, 'LineStyle',
'--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('$\dot{\eta}$ error [m/s]', 'Interpreter', 'latex');
title('$\dot{\eta}$ (Velocity in y)', 'Interpreter','latex');

% Adjust layout and size
set(gcf, 'Position', [100, 100, 700, 900]);
```




Kalman Filter Estimated State Error
$\xi$ (Position in x)
$\xi$ error [m]
10
5
0
-5
-10
0
10
20
30
40
50
60
70
80
90
100
Time (s)
$\dot{\xi}$ (Velocity in x)
$\dot{\xi}$ error [m/s]
100
50
0
-50
-100
Estimated State Error
$2\sigma$ Bounds
Time (s)
$\eta$ (Position in y)
$\eta$ error [m]
Time (s)
$\dot{\eta}$ (Velocity in y)
$\dot{\eta}$ error [m/s]
Time (s)

# Problem 3

State transition matrices

```
F = blkdiag(Fa, Fb);
Q = blkdiag(Qa, Qb);

% Measurement matrix
Ha = [1, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 1, 0, 0, 0, 0, 0];

Hd = [1, 0, 0, 0, -1, 0, 0, 0;
    0, 0, 1, 0, 0, 0, -1, 0];

H = [Ha; Hd];

% Sensor noise covariance matrix
Rd = [10, 0.15; 0.15, 10];
R = blkdiag(Ra, Rd);

% Kalman filter initialization
mu_a_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
mu_b_0 = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)];
mu_s_0 = [mu_a_0; mu_b_0];

Pa_0 = 900 .* diag([10, 2, 10, 2]);
Pb_0 = 900 .* diag([11, 4, 11, 4]);
P_0 = blkdiag(Pa_0, Pb_0);

% Generate noisy measurements
time_vec = 0.5:0.5:100;
ys_mat = zeros(4, length(time_vec));
for i = 2:length(xadouble_truth(1,:))
    % Noisy measurement for aircraft A
    pos_xa = [xadouble_truth(1,i); xadouble_truth(3,i)];
    ya = mvnrnd(pos_xa', Ra, 1);

    % Noisy measurement for relative vector between A and B
    pos_xb = [xbdouble_truth(1,i); xbdouble_truth(3,i)];
    rd = pos_xa - pos_xb;
    yd = mvnrnd(rd', Rd, 1);

    % Combine into a stacked measurement
    ys_mat(:,i-1) = [ya'; yd'];
end

% Initialization
P_meas = P_0;
x_meas = mu_s_0;

% New KF to estimate both aircraft states
xhat_s_mat = zeros(8, length(ys_mat(1,:)));
sigma_s_mat = zeros(8, length(ys_mat(1,:)));
```

```
for i = 1:length(ys_mat(1,:))
    % Time update
    x_pred = F*x_meas;
    P_pred = F*P_meas*F' + Q;
    K_gain = P_pred * H' * inv(H * P_pred * H' + R);

    % Measurement update
    x_meas = x_pred + K_gain*(ys_mat(:,i) - H * x_pred);
    P_meas = (eye(8) - K_gain*H) * P_pred;

    % Store the estimates and uncertainties
    xhat_s_mat(:,i) = x_meas;
    sigma_s_mat(:,i) = sqrt(diag(P_meas));
end

% Calculate the state errors
true_mat = [xadouble_truth(:,2:end); xbdouble_truth(:,2:end)];
state_error_mat = xhat_s_mat - true_mat;


% Repeat with only the transponder measurements
% Initialization
P_meas = P_0;
x_meas = mu_s_0;

% New KF to estimate both aircraft states
xhat_d_mat = zeros(8, length(ys_mat(1,:)));
sigma_d_mat = zeros(8, length(ys_mat(1,:)));
for i = 1:length(ys_mat(1,:))
    % Time update
    x_pred = F*x_meas;
    P_pred = F*P_meas*F' + Q;
    K_gain = P_pred * Hd' * inv(Hd * P_pred * Hd' + Rd);

    % Measurement update
    x_meas = x_pred + K_gain*(ys_mat(3:4,i) - Hd * x_pred);
    P_meas = (eye(8) - K_gain*Hd) * P_pred;

    % Store the estimates and uncertainties
    xhat_d_mat(:,i) = x_meas;
    sigma_d_mat(:,i) = sqrt(diag(P_meas));
end

% Calculate the state errors
state_error_d_mat = xhat_d_mat - true_mat;



% Repeat with pure prediction
% Initialization
P_pred = P_0;
x_pred = mu_s_0;

% New KF to estimate both aircraft states
```

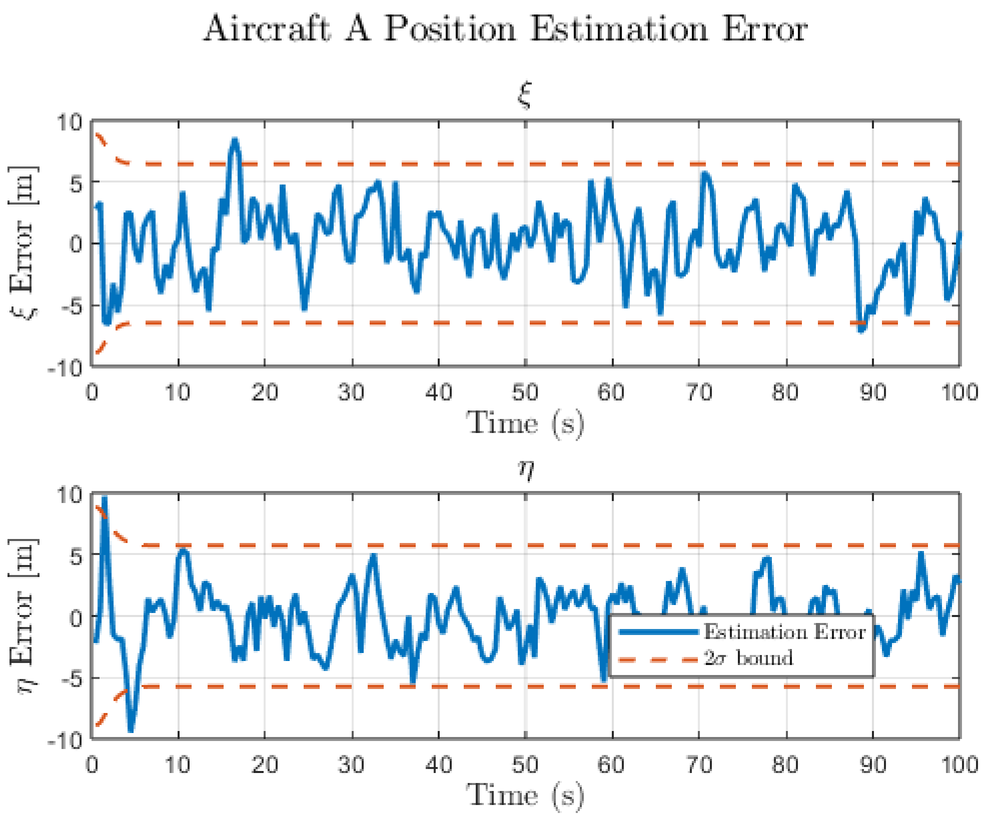
```
xhat_pred_mat = zeros(8, length(ys_mat(1,:)));
sigma_pred_mat = zeros(8, length(ys_mat(1,:)));
for i = 1:length(ys_mat(1,:))
    % Time update
    x_pred = F*x_pred;
    P_pred = F*P_pred*F' + Q;

    % Store the estimates and uncertainties
    xhat_pred_mat(:,i) = x_pred;
    sigma_pred_mat(:,i) = sqrt(diag(P_pred));
end

% Calculate the state errors
state_error_pred_mat = xhat_pred_mat - true_mat;

%%%% All measurements positional errors plot
% Aircraft A
% Positional Errors Plot for Aircraft A
figure();
sgtitle('Aircraft A Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_mat(1,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_s_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_s_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_mat(3,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_s_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_s_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');

% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white
```

```
% Aircraft B
% Positional Errors Plot for Aircraft B
figure();
sgtitle('Aircraft B Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_mat(5,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_s_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_s_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_mat(7,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_s_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_s_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');

% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white



%%%% Only transponder measurements positional errors plot
% Aircraft A
% Positional Errors Plot for Aircraft A
figure();
sgtitle('Aircraft A Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_d_mat(1,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
```

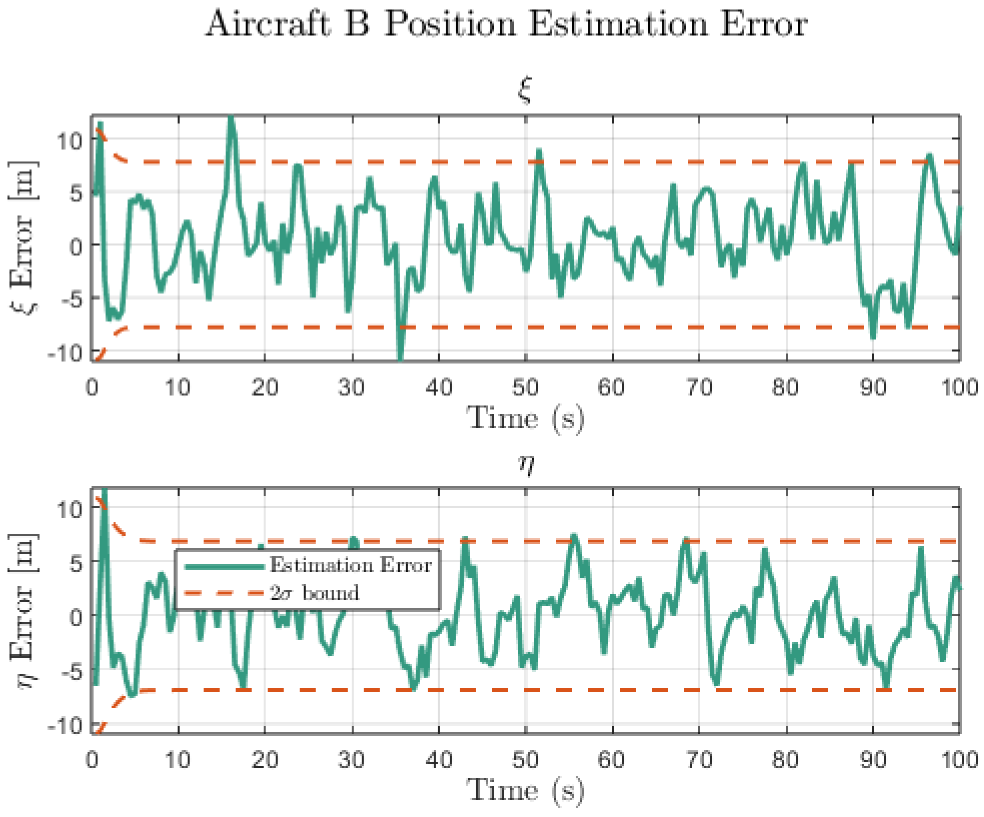
```
hold on;
plot(time_vec, 2 .* sigma_d_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_d_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_d_mat(3,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_d_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_d_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');

% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white


% Aircraft B
% Positional Errors Plot for Aircraft B
figure();
sgtitle('Aircraft B Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_d_mat(5,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_d_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_d_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_d_mat(7,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
```

```
hold on;
plot(time_vec, 2 .* sigma_d_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_d_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');

% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white



%%%% Pure prediction plot
% Aircraft A
% Positional Errors Plot for Aircraft A
figure();
sgtitle('Aircraft A Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_pred_mat(1,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_pred_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_pred_mat(1,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_pred_mat(3,:), 'LineWidth', 2, 'Color', [0 0.4470
0.7410]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_pred_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_pred_mat(3,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');
```

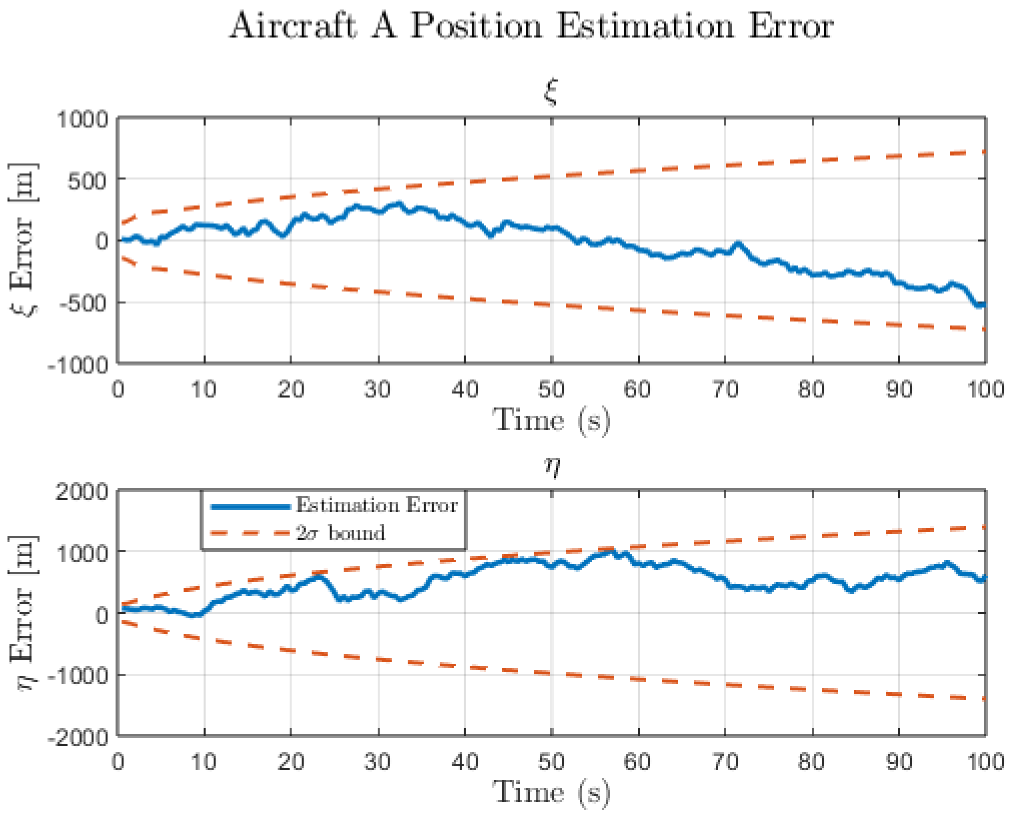
```
% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white


% Aircraft B
% Positional Errors Plot for Aircraft B
figure();
sgtitle('Aircraft B Position Estimation Error', 'FontSize', 14,
'Interpreter', 'latex')

% Subplot 1: Position error in x (ξ)
subplot(2,1,1)
plot(time_vec, state_error_pred_mat(5,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_pred_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_pred_mat(5,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\xi$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\xi$', 'FontSize', 12, 'Interpreter', 'latex');

% Subplot 2: Position error in y (η)
subplot(2,1,2)
plot(time_vec, state_error_pred_mat(7,:), 'LineWidth', 2, 'Color', [0.2 0.6
0.5]); % Default MATLAB blue
hold on;
plot(time_vec, 2 .* sigma_pred_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for +2σ
plot(time_vec, -2 .* sigma_pred_mat(7,:), '--', 'LineWidth', 1.5, 'Color',
[0.8500 0.3250 0.0980]); % Dashed red for -2σ
grid on;
xlabel('Time (s)', 'FontSize', 12, 'Interpreter', 'latex');
ylabel('$\eta$ Error [m]', 'FontSize', 12, 'Interpreter', 'latex');
title('$\eta$', 'FontSize', 12, 'Interpreter', 'latex');
legend('Estimation Error', '2$\sigma$ bound', 'Location', 'Best',
'Interpreter', 'latex');

% General Enhancements
set(gcf, 'Color', 'w'); % Set figure background to white
```

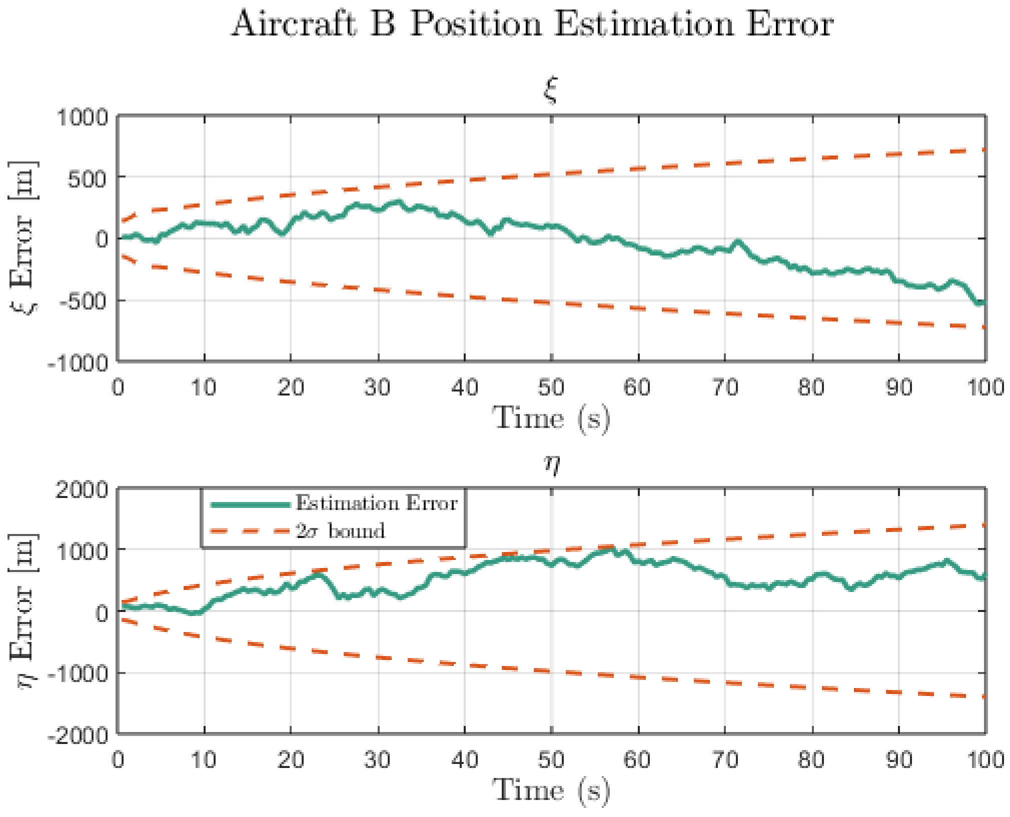


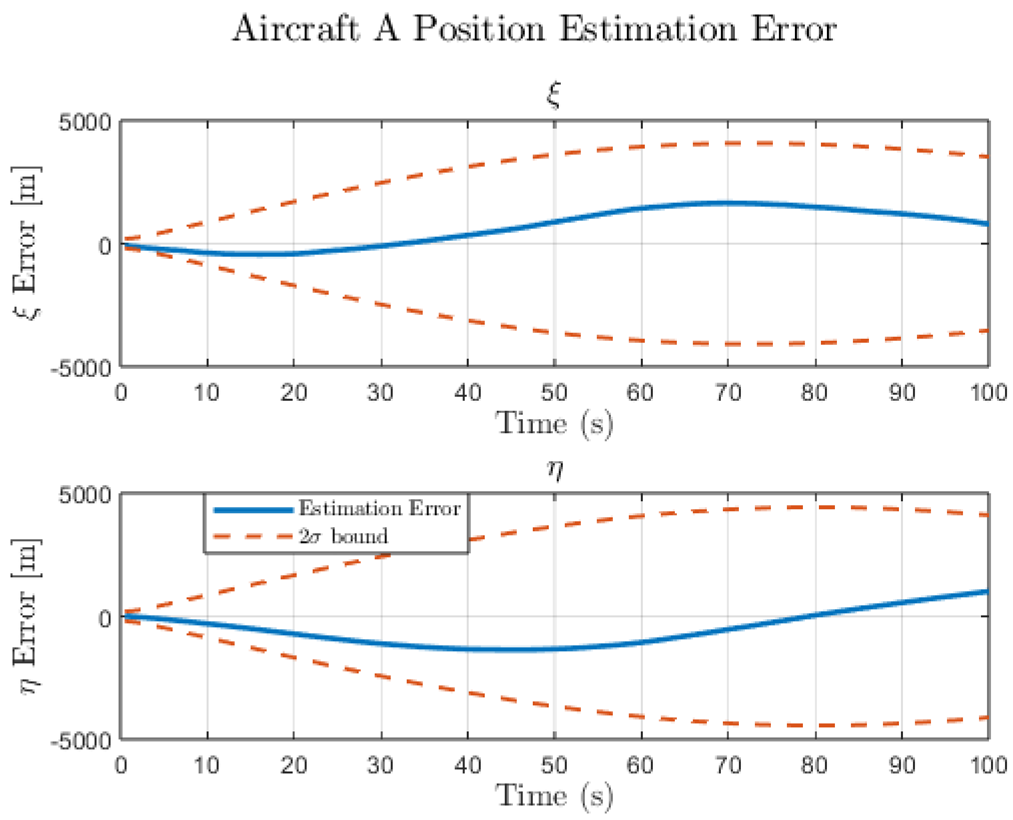
Aircraft A Position Estimation Error



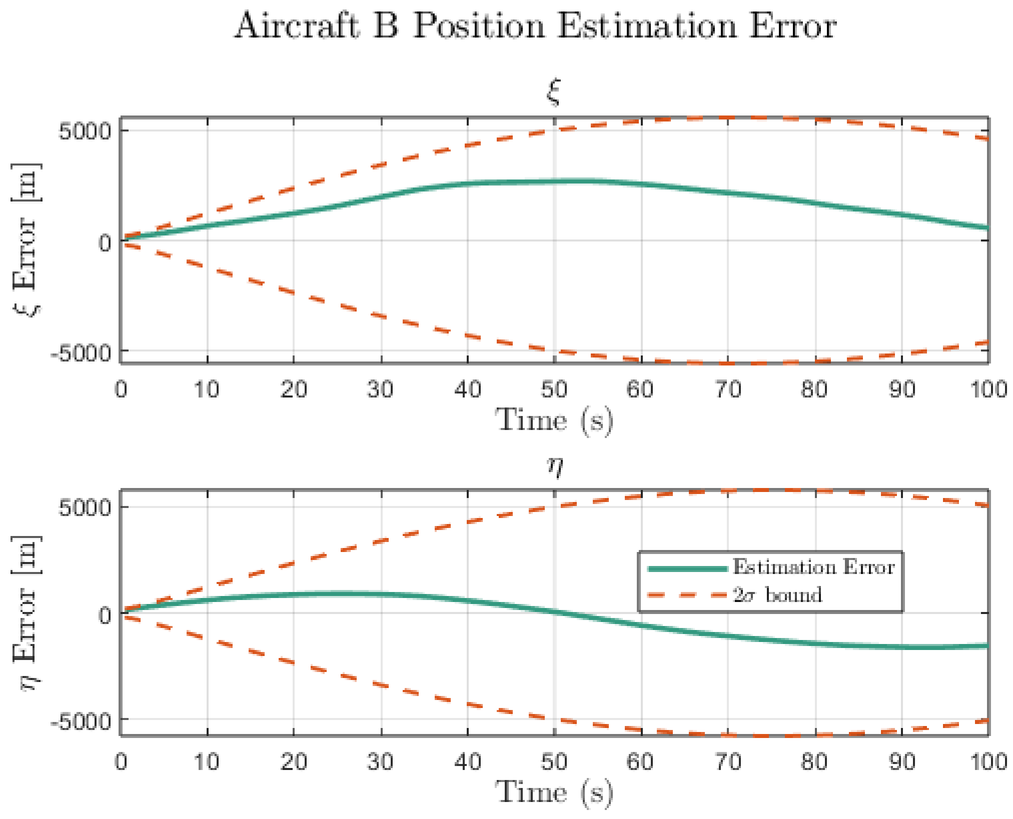
Aircraft B Position Estimation Error


Aircraft A Position Estimation Error
$\xi$
1000
500
0
-500
-1000
$\xi$ Error [m]
0
10
20
30
40
50
60
70
80
90
100
Time (s)
$\eta$
Estimation Error
$2\sigma$ bound
2000
1000
0
-1000
-2000
$\eta$ Error [m]
0
10
20
30
40
50
60
70
80
90
100
Time (s)


Aircraft B Position Estimation Error
$\xi$
1000
500
0
-500
-1000
$\xi$ Error [m]
0
10
20
30
40
50
60
70
80
90
100
Time (s)
$\eta$
Estimation Error
$2\sigma$ bound
2000
1000
0
-1000
-2000
$\eta$ Error [m]
0
10
20
30
40
50
60
70
80
90
100
Time (s)



Aircraft A Position Estimation Error



Aircraft B Position Estimation Error

*Published with MATLAB® R2023b*