
Table of Contents

Clean	1
Problem 1	1
Problem 3a	1
Problem 3b	5
Problem 3c	5
Problem 3d	7
Problem 3e	7
Problem 3d	8
CHAD	10

Clean

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

Problem 1

```
A = [0, 1; -100, -10];
gamma = [0; 1];
dt = 0.2;
W = 10;

Z = dt .* [-A, gamma*W*gamma'; zeros(2,2), A'];
expmz = expm(Z);

F = expmz(3:4, 3:4)';
FinvQ = expmz(1:2, 3:4);

Q = F * FinvQ;
```

Problem 3a

```
R = [8, 5.15, 6.5; 5.15, 5, -4.07; 6.5, -4.07, 50];

% Cholesky decomposition
Sv = chol(R, "lower");

% Mean value
mx = [1; 1; 1];

numMeas = 100;
yk = zeros(numMeas, 3);
for i = 1:numMeas
    qk = randn(3, 1);
    yk(i,:) = (mx + Sv*qk)';
end
```

```

% Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% xi vs eta
figure();
scatter(yk(:,1), yk(:,2), 'filled', 'MarkerFaceColor', 'r',
'MarkerEdgeColor', 'k'); % red markers with black edge
hold on
scatter(1, 1, 'filled', 'MarkerFaceColor', 'g', 'MarkerEdgeColor', 'g');
xlabel('$y_{k}(1) (\xi)$ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Xi
ylabel('$y_{k}(2) (\eta)$ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Eta
title('$\xi$ vs $\eta$ Position', 'Interpreter', 'latex', 'FontSize', 16);

% Set grid and axis properties
grid on;
axis equal; % Equal scaling for both axes
ax = gca;
ax.FontSize = 12;
ax.LineWidth = 1.2;

% Adjust axis limits and add minor grid lines
xlim([min(yk(:,1)) - 0.1, max(yk(:,1)) + 0.1]); % Adding padding
ylim([min(yk(:,2)) - 0.1, max(yk(:,2)) + 0.1]);
ax.XMinorGrid = 'on';
ax.YMinorGrid = 'on';

% Add legend if needed
legend('measurement', '$m_{y}$', 'Interpreter', 'latex', 'FontSize', 12,
'Location', 'best');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% xi vs z
figure();
scatter(yk(:,1), yk(:,3), 'filled', 'MarkerFaceColor', 'r',
'MarkerEdgeColor', 'k'); % red markers with black edge
hold on
scatter(1, 1, 'filled', 'MarkerFaceColor', 'g', 'MarkerEdgeColor', 'g');
xlabel('$y_{k}(1) (\xi)$ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Xi
ylabel('$y_{k}(3) (z)$ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Eta
title('$\xi$ vs $z$ Position', 'Interpreter', 'latex', 'FontSize', 16);

% Set grid and axis properties
grid on;
axis equal; % Equal scaling for both axes
ax = gca;
ax.FontSize = 12;
ax.LineWidth = 1.2;

% Adjust axis limits and add minor grid lines
% xlim([min(yk(:,1)) - 0.1, max(yk(:,1)) + 0.1]); % Adding padding
xlim([min(yk(:,3)) - 0.1, max(yk(:,3)) + 0.1]); % Adding padding
ylim([min(yk(:,3)) - 0.1, max(yk(:,3)) + 0.1]);
ax.XMinorGrid = 'on';

```

```

ax.YMinorGrid = 'on';

% Add legend if needed
legend('measurement', '$m_{y}$', 'Interpreter', 'latex', 'FontSize', 12,
'Location', 'best');

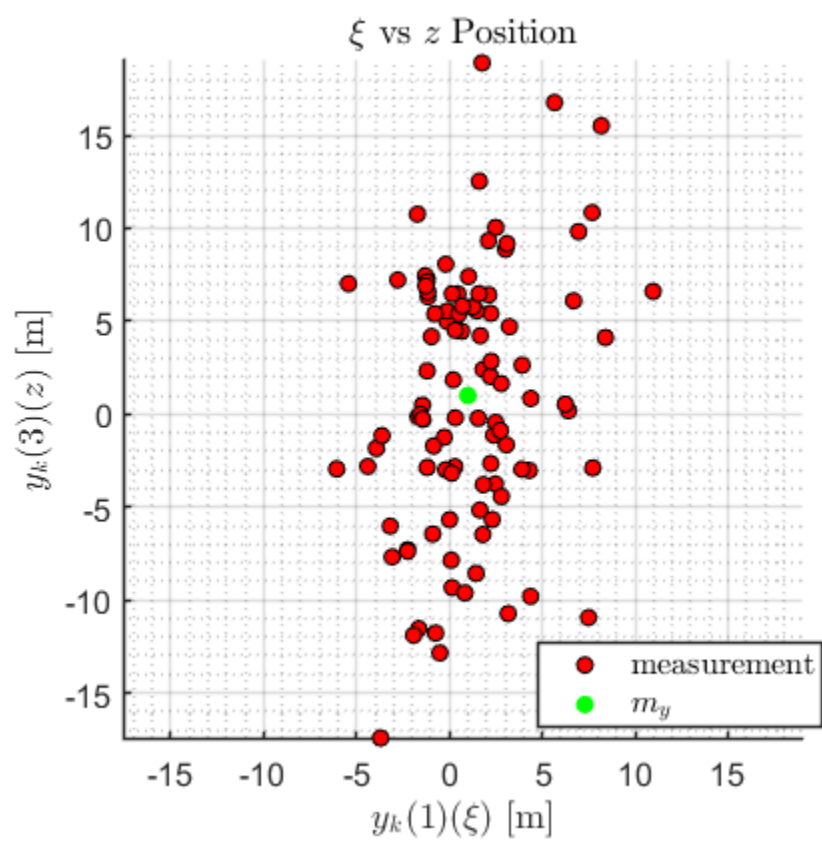
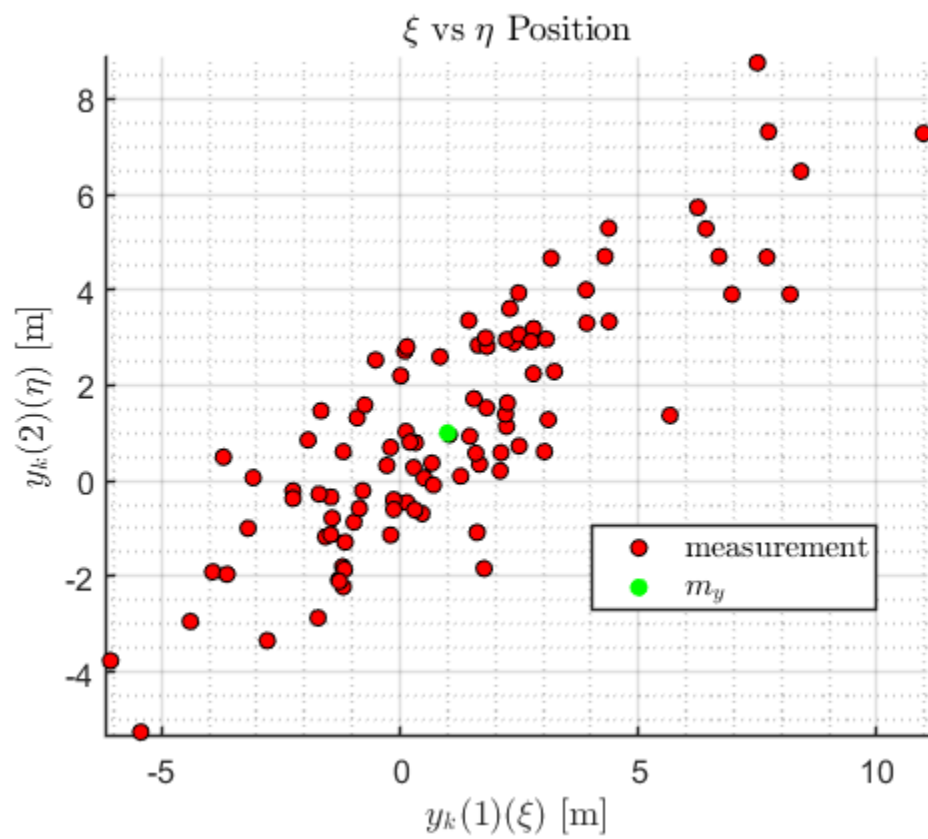
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% eta vs z
figure();
scatter(yk(:,2), yk(:,3), 'filled', 'MarkerFaceColor', 'r',
'MarkerEdgeColor', 'k'); % red markers with black edge
hold on
scatter(1, 1, 'filled', 'MarkerFaceColor', 'g', 'MarkerEdgeColor', 'g');
xlabel('$y_{k}(2)$ (\eta) $ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Xi
ylabel('$y_{k}(3)$ , (z) $ [m]', 'Interpreter', 'latex', 'FontSize', 14); %
Greek letter Eta
title('$\eta$ vs $z$ Position', 'Interpreter', 'latex', 'FontSize', 16);

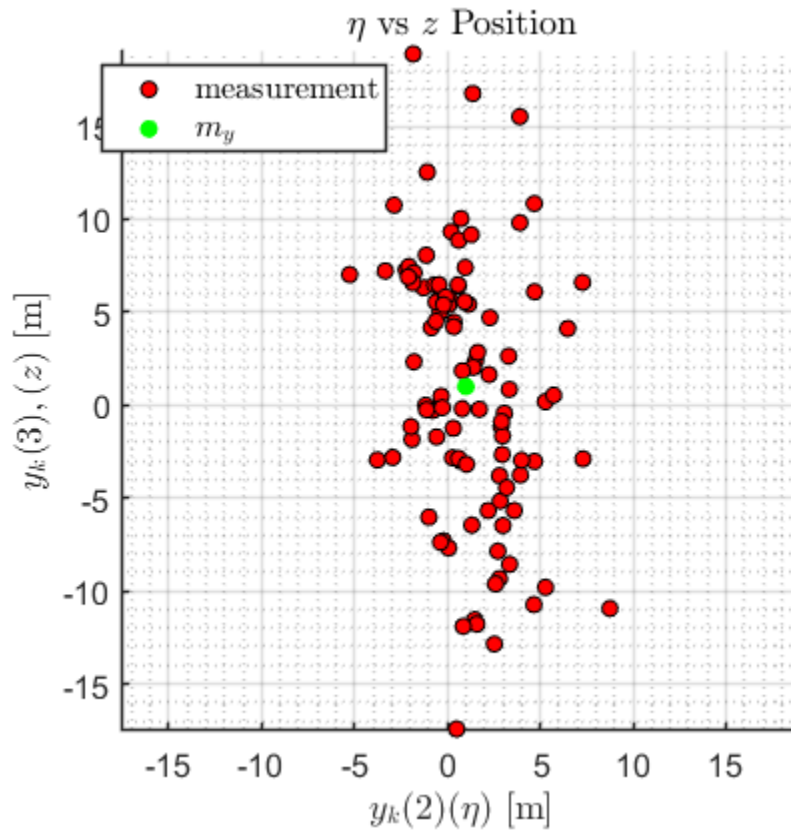
% Set grid and axis properties
grid on;
axis equal; % Equal scaling for both axes
ax = gca;
ax.FontSize = 12;
ax.LineWidth = 1.2;

% Adjust axis limits and add minor grid lines
%xlim([min(yk(:,2)) - 0.1, max(yk(:,2)) + 0.1]); % Adding padding
xlim([min(yk(:,3)) - 0.1, max(yk(:,3)) + 0.1]); % Adding padding
ylim([min(yk(:,3)) - 0.1, max(yk(:,3)) + 0.1]);
ax.XMinorGrid = 'on';
ax.YMinorGrid = 'on';

% Add legend if needed
legend('measurement', '$m_{y}$', 'Interpreter', 'latex', 'FontSize', 12,
'Location', 'best');

```





Problem 3b

```
covMat = cov(yk);
```

Problem 3c

3 measurements

```
T_ls = 3;
y_vec = zeros(T_ls*3, 1);
Hmat = zeros(3*T_ls, 3);
Rmat = zeros(3*T_ls);
for i = 1:T_ls
    % Stack y measurements
    y_vec(3*i - 2) = yk(i, 1);
    y_vec(3*i - 1) = yk(i, 2);
    y_vec(3*i) = yk(i, 3);

    % Stack H matrices
    Hmat(3*i - 2 : 3*i, :) = eye(3);

    % Stack R matrices
    Rmat(3*i - 2 : 3*i, 3*i - 2 : 3*i) = R;
end
```

```

% Least squares
xls_3 = inv(Hmat' * inv(Rmat) * Hmat) * Hmat' * inv(Rmat) * y_vec;
Pls_3 = inv(Hmat' * inv(Rmat) * Hmat);

% 10 measurements
T_ls = 10;
y_vec = zeros(T_ls*3, 1);
Hmat = zeros(3*T_ls, 3);
Rmat = zeros(3*T_ls);
for i = 1:T_ls
    % Stack y measurements
    y_vec(3*i - 2) = yk(i, 1);
    y_vec(3*i - 1) = yk(i, 2);
    y_vec(3*i) = yk(i, 3);

    % Stack H matrices
    Hmat(3*i - 2 : 3*i, :) = eye(3);

    % Stack R matrices
    Rmat(3*i - 2 : 3*i, 3*i - 2 : 3*i) = R;
end

% Least squares
xls_10 = inv(Hmat' * inv(Rmat) * Hmat) * Hmat' * inv(Rmat) * y_vec;
Pls_10 = inv(Hmat' * inv(Rmat) * Hmat);

% 100 measurements
T_ls = 100;
y_vec = zeros(T_ls*3, 1);
Hmat = zeros(3*T_ls, 3);
Rmat = zeros(3*T_ls);
for i = 1:T_ls
    % Stack y measurements
    y_vec(3*i - 2) = yk(i, 1);
    y_vec(3*i - 1) = yk(i, 2);
    y_vec(3*i) = yk(i, 3);

    % Stack H matrices
    Hmat(3*i - 2 : 3*i, :) = eye(3);

    % Stack R matrices
    Rmat(3*i - 2 : 3*i, 3*i - 2 : 3*i) = R;
end

% Least squares
xls_100 = inv(Hmat' * inv(Rmat) * Hmat) * Hmat' * inv(Rmat) * y_vec;
Pls_100 = inv(Hmat' * inv(Rmat) * Hmat);

% Calculate errors
x_ls_3 = [-1.4136; 0.9825; -8.5505];
x_ls_10 = [0.666; 1.3411; -1.753];
x_ls_100 = [1.06446; 0.9244; 1.3991];

```

```
x_true = [1;1;1];

e_3 = norm(x_ls_3 - x_true);
e_10 = norm(x_ls_10 - x_true);
e_100 = norm(x_ls_100 - x_true);
```

Problem 3d

```
data = readmatrix('hw6problem3data.csv');

% 30 measurements
T_ls = 30;
y_vec = zeros(T_ls*3, 1);
Hmat = zeros(3*T_ls, 3);
Rmat = zeros(3*T_ls);
for i = 1:T_ls
    % Stack y measurements
    y_vec(3*i - 2) = data(1, i);
    y_vec(3*i - 1) = data(2, i);
    y_vec(3*i) = data(3, i);

    % Stack H matrices
    Hmat(3*i - 2 : 3*i, :) = eye(3);

    % Stack R matrices
    Rmat(3*i - 2 : 3*i, 3*i - 2 : 3*i) = R;
end

% Least squares
xls_30 = inv(Hmat' * inv(Rmat) * Hmat) * Hmat' * inv(Rmat) * y_vec;
Pls_30 = inv(Hmat' * inv(Rmat) * Hmat);
```

Problem 3e

30 measurements

```
T_ls = 30;
y_vec = zeros(T_ls*3, 1);
Hmat = zeros(3*T_ls, 3);

for i = 1:T_ls
    % Stack y measurements
    y_vec(3*i - 2) = data(1, i);
    y_vec(3*i - 1) = data(2, i);
    y_vec(3*i) = data(3, i);

    % Stack H matrices
    Hmat(3*i - 2 : 3*i, :) = eye(3);
end

% Least squares
```

```

xls_30 = sum(data, 2) ./ length(data(1,:));
Pls_30 = inv(Hmat' * Hmat);

```

Problem 3d

```

Hk = eye(3);
Kl = eye(3);

x0 = [0;0;0];
x1 = x0 + (data(:,1) - x0);
P1 = R;

% Recursive least squares estimation
xhat_vec = x0;
xhat_vec(:,2) = x1;
sigma_vec = [sqrt(P1(1,1)); sqrt(P1(2,2)); sqrt(P1(3,3))];
Pk = P1;
I = eye(3);
for i = 2:length(data(1,:))
    Kk = Pk*Hk*inv(Hk*Pk*Hk + R);
    xhat_vec(:,i+1) = xhat_vec(:,i) + Kk*(data(:,i) - Hk*xhat_vec(:,i));
    Pk = (I - Kk*Hk)*Pk*(I - Kk*Hk)' + Kk*R*Kk';
    sigma_vec(:,i) = [sqrt(Pk(1,1)); sqrt(Pk(2,2)); sqrt(Pk(3,3))];
end

% Get 2 sigma bounds
sigma2_p = [xhat_vec(1,2:end) + 2*sigma_vec(1,:); xhat_vec(2,2:end) +
2*sigma_vec(2,:); xhat_vec(3,2:end) + 2*sigma_vec(3,:)];
sigma2_m = [xhat_vec(1,2:end) - 2*sigma_vec(1,:); xhat_vec(2,2:end) -
2*sigma_vec(2,:); xhat_vec(3,2:end) - 2*sigma_vec(3,:)];

% Plotting
figure();
%%% XI
subplot(3,1,1)
sgtitle('Estimated Position Over Time')
plot(0:30, xhat_vec(1,:), 'linewidth', 2, 'color', 'r')
hold on
plot(1:30, sigma2_p(1,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')
plot(1:30, sigma2_m(1,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')

xlabel('Time step k')
ylabel('$\xi$ [m]', 'Interpreter', 'latex', 'FontSize', 14); % Greek letter
Eta

%%% ETA
subplot(3,1,2)
plot(0:30, xhat_vec(2,:), 'linewidth', 2, 'color', 'b')
hold on
plot(1:30, sigma2_p(2,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')
plot(1:30, sigma2_m(2,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')

```

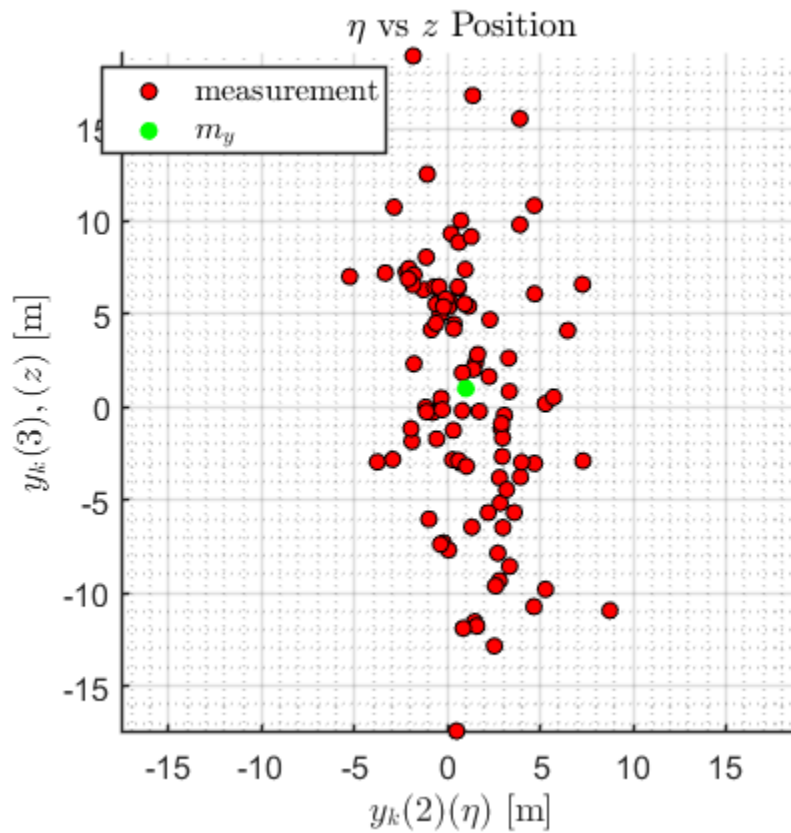
```

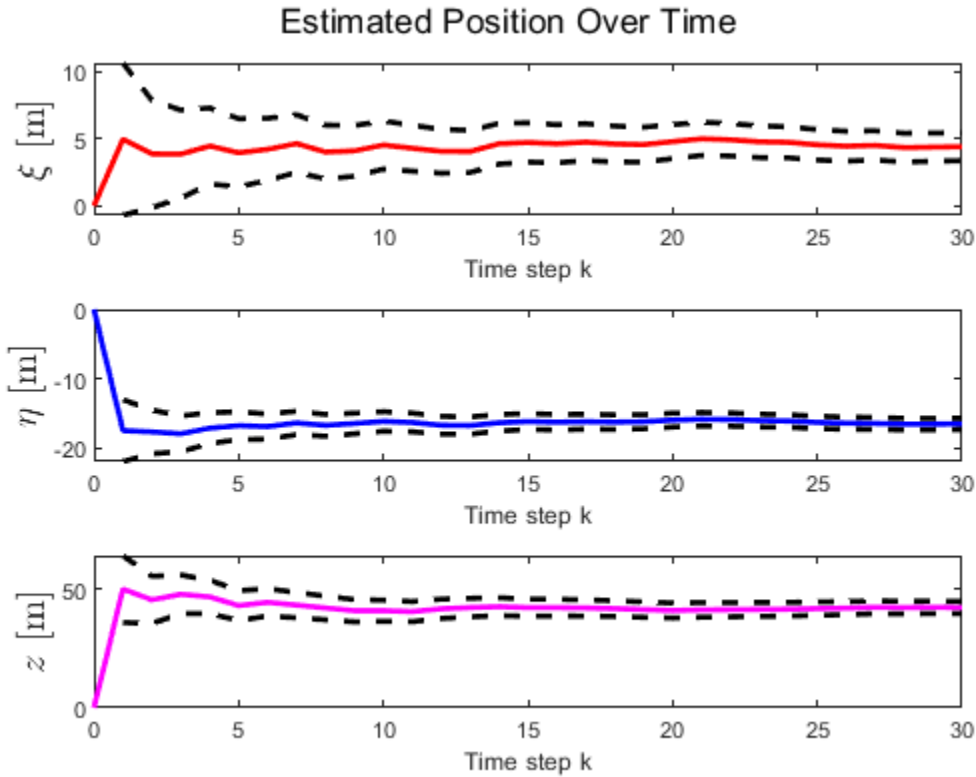
xlabel('Time step k')
ylabel('$\eta$ [m]', 'Interpreter', 'latex', 'FontSize', 14); % Greek letter
Eta

%%% Z
subplot(3,1,3)
plot(0:30, xhat_vec(3,:), 'linewidth', 2, 'color', 'm')
hold on
plot(1:30, sigma2_p(3,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')
plot(1:30, sigma2_m(3,:), 'linewidth', 2, 'color', 'k', 'LineStyle','--')

xlabel('Time step k')
ylabel('$z$ [m]', 'Interpreter', 'latex', 'FontSize', 14); % Greek letter Eta

```





CHAD

```
figure();
sgtitle('Estimated Position Over Time', 'FontSize', 16, 'FontWeight', 'bold')

% Define colors
estimateColor = {'r', 'b', 'm'};
boundColor = [0.2, 0.2, 0.2]; % light gray for bounds

% XI plot
subplot(3,1,1)
plot(1:30, xhat_vec(1,2:end), 'LineWidth', 2, 'Color', estimateColor{1})
hold on
plot(1:30, sigma2_p(1,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
plot(1:30, sigma2_m(1,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
xlabel('Time step k', 'FontSize', 12)
ylabel('$\xi$ [m]', 'Interpreter', 'latex', 'FontSize', 14)
legend('Estimate', '$2\sigma$ Bound', 'Location', 'Best', 'Interpreter',
'latex')
grid on

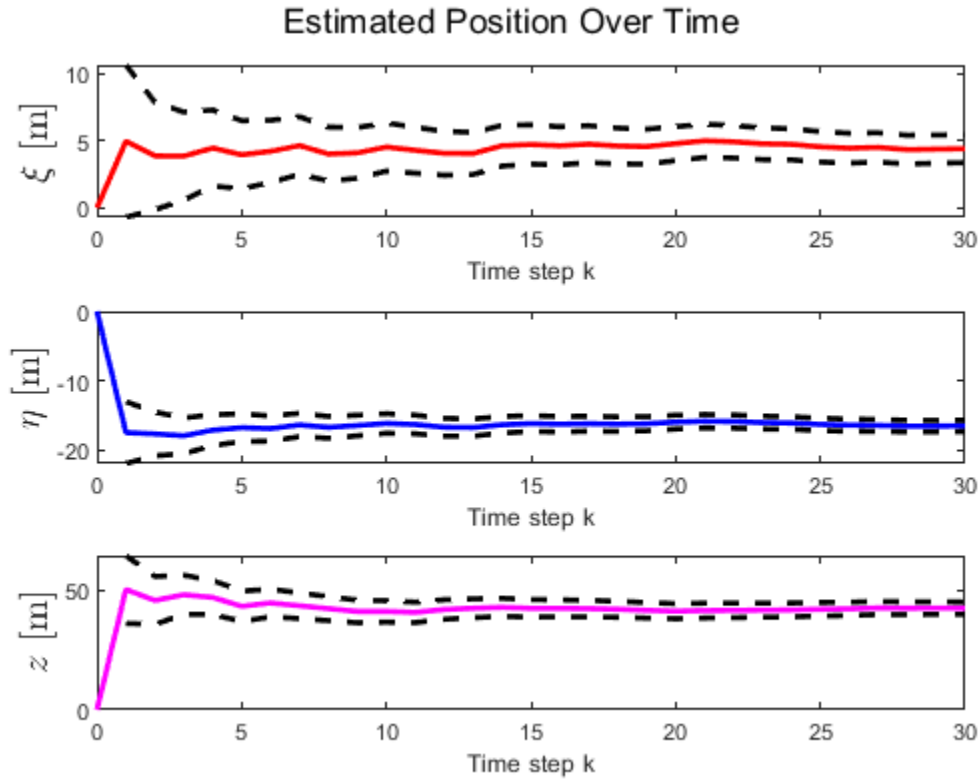
% ETA plot
subplot(3,1,2)
```

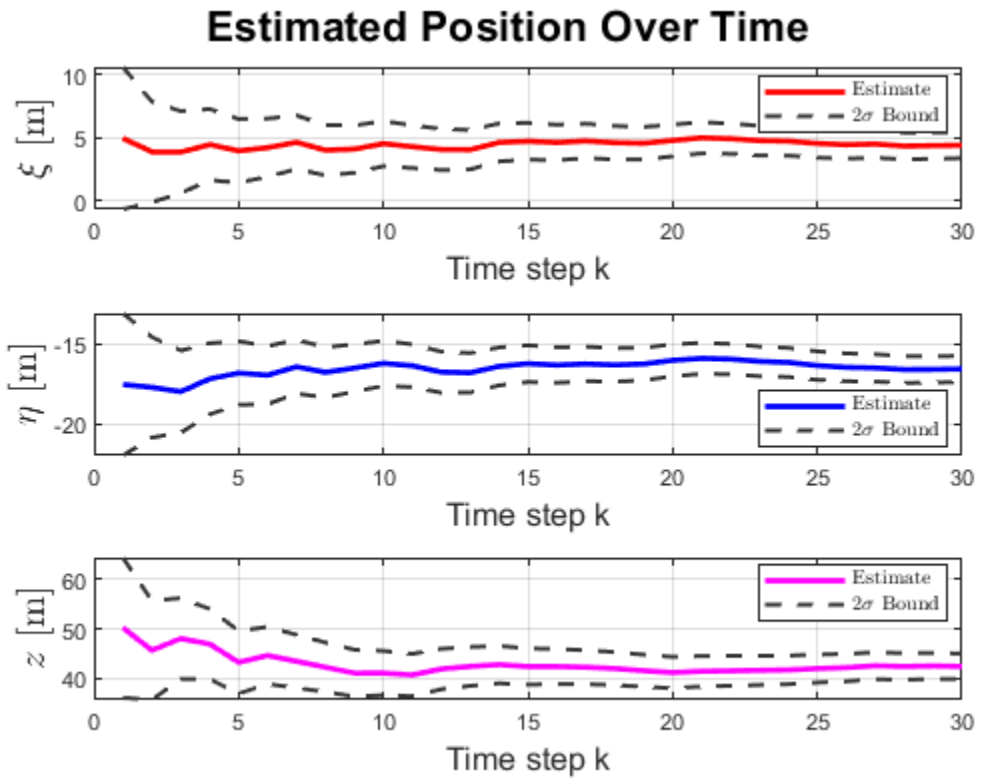
```

plot(1:30, xhat_vec(2,2:end), 'LineWidth', 2, 'Color', estimateColor{2})
hold on
plot(1:30, sigma2_p(2,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
plot(1:30, sigma2_m(2,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
xlabel('Time step k', 'FontSize', 12)
ylabel('$\eta$ [m]', 'Interpreter', 'latex', 'FontSize', 14)
legend('Estimate', '$2\sigma$ Bound', 'Location', 'Best', 'Interpreter',
'latex')
grid on

% Z plot
subplot(3,1,3)
plot(1:30, xhat_vec(3,2:end), 'LineWidth', 2, 'Color', estimateColor{3})
hold on
plot(1:30, sigma2_p(3,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
plot(1:30, sigma2_m(3,:), 'LineWidth', 1.5, 'Color', boundColor,
'LineStyle', '--')
xlabel('Time step k', 'FontSize', 12)
ylabel('$z$ [m]', 'Interpreter', 'latex', 'FontSize', 14)
legend('Estimate', '$2\sigma$ Bound', 'Location', 'Best', 'Interpreter',
'latex')
grid on

```





Published with MATLAB® R2024a