ASEN 5044 - HW 6, Fall 2024, Jash Bhalavat

①

Jash Bhalavat
ASEN 5044
Fall 2024

$\boxed{\text{HW \#6}}$

Problem 1 → $\ddot{z} + 10\dot{z} + 100z = f(t)$, $z(0) = 0$, $\dot{z}(0) = 0$, $W = 10$ units

a) $x(t) = \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}$, $\dot{x}(t) = \begin{bmatrix} \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -100 & -10 \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [f(t)]$

$\boxed{\dot{x}(t) = Ax(t) + \Gamma\tilde{w}(t)}$

$\boxed{\text{where } A = \begin{bmatrix} 0 & 1 \\ -100 & -10 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x(t) = \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}, \tilde{w}(t) = f(t), W = 10}$

b) $t = 0$, $\Delta T = 0.2$ sec,

$F = e^{A\Delta T} = \underset{\text{(matlab)}}{\text{expm}}(A \cdot 0.2) = \begin{bmatrix} 0.1506 & 0.0419 \\ -4.1928 & -0.2687 \end{bmatrix}$

Use Van Loan's method to get Q

$Z = \Delta T \cdot \begin{bmatrix} -A & \Gamma W \Gamma^T \\ 0 & A^T \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 100 & 10 & 0 & 10 \\ 0 & 0 & 0 & -100 \\ 0 & 0 & 1 & -10 \end{bmatrix} \cdot 0.2 = \begin{bmatrix} 0 & -0.2 & 0 & 0 \\ 20 & 2 & 0 & 2 \\ 0 & 0 & 0 & -20 \\ 0 & 0 & 0.2 & -2 \end{bmatrix}$

$e^Z = \underset{\text{(matlab)}}{\text{expm}}(Z) = \begin{bmatrix} -1.9855 & -0.3098 & -0.0107 & -0.1339 \\ 30.9808 & 1.1126 & 0.1339 & 0.6907 \\ 0 & 0 & 0.1506 & -4.1928 \\ 0 & 0 & 0.0419 & -0.2687 \end{bmatrix} = \left[\begin{array}{c|c} (\dots) & F^{-1}Q \\ \hline 0 & F^T \end{array}\right]$

$Q = (F^T)^T \cdot [F^{-1}Q] = \begin{bmatrix} 0.1506 & 0.0419 \\ -4.1928 & -0.2687 \end{bmatrix} \begin{bmatrix} -0.0107 & -0.1339 \\ 0.1339 & 0.6907 \end{bmatrix} = \begin{bmatrix} 0.0040 & 0.0088 \\ 0.0088 & 0.3760 \end{bmatrix} = Q$

$x_{k+1} = Fx_k + w_k \quad \rightarrow \quad w_k = Q\,\delta(k,j)$

$\begin{bmatrix} z_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} = \begin{bmatrix} 0.1506 & 0.0419 \\ -4.1928 & -0.2687 \end{bmatrix} \begin{bmatrix} z_k \\ \dot{z}_k \end{bmatrix} + \begin{bmatrix} 0.004 & 0.0088 \\ 0.0088 & 0.376 \end{bmatrix} \delta(k,j)$

c) $\boxed{z \rightarrow m, \ \dot{z} \rightarrow m/s}$

$\boxed{Q \rightarrow \begin{bmatrix} (m)^2 & m \cdot m/s \\ m \cdot \frac{m}{s} & (m/s)^2 \end{bmatrix}}$ → ∵ Q is not diagonal, z and $\dot{z}$ are correlated, not independent. Off-diagonal terms in Q indicate cross covariance of z and $\dot{z}$ disturbance due to process noise. So, there's linear integration coupling.

d) $y(t) = z + \dot{z}\Delta T + \tilde{v}(t)$, $V = 3$

$y(t) = \begin{bmatrix} 1 & \Delta T \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \tilde{v}(t) = Cx(t) + \tilde{v}(t) \rightarrow C = \begin{bmatrix} 1 & \Delta T \end{bmatrix}$ where $\Delta T = 0.2$ s

$\boxed{H = C = \begin{bmatrix} 1 & 0.2 \end{bmatrix}}$

$R(k) = \frac{V}{\Delta T}\delta(k,j) = \frac{3}{0.2}\delta(k,j) = \boxed{15\delta(k,j) = R(k)}$

If $z \rightarrow$ meters $\rightarrow$ $\boxed{R \rightarrow \text{meters}^2}$

Problem 2 → <u>3-4</u>

$P_0 = \infty$ $\sigma_1^2 = 1$, $\sigma_2^2 = 4$ $\rightarrow R = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$, $R^{-1} = \frac{1}{|R|}\begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}$

a) $H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ $\rightarrow \hat{x}_{LS} = (H^T R^{-1} H)^{-1} H^T R^{-1} \vec{y}$

$\hat{x}_{LS} = \left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)^{-1}\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1/4 \end{bmatrix}\right)^{-1}\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2/4 \end{bmatrix}$

$= \left(\frac{5}{4}\right)^{-1}\left(y_1 + \frac{y_2}{4}\right) = \boxed{\frac{4y_1}{5} + \frac{y_2}{5} \text{ [Volts]} = \hat{x}_{LS}}$

b) Variance after 1st measurement $\rightarrow (H^T R^{-1} H)^{-1} = ([1][1]^{-1}[1])^{-1} = \boxed{1\ V^2}$

Variance after 2nd measurement $\rightarrow (H^T R^{-1} H)^{-1} =$ (from part a) $\rightarrow \boxed{\frac{4}{5}\ V^2}$

c) Voltage $\rightarrow (y_1 + y_2)\frac{1}{2}$ → unweighted average

$\hat{x}_{LS} = \left(\sum_{k=1}^{2}\frac{1}{\sigma_k^2}\right)^{-1}\left(\frac{y_1}{\sigma_1^2} + \frac{y_2}{\sigma_2^2}\right) = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}\left(\frac{y_1}{\sigma_1^2} + \frac{y_2}{\sigma_2^2}\right) = (y_1 + y_2)\frac{1}{2}$

$\therefore \sigma_1^2 = \sigma_2^2 = 1 \rightarrow$ $\therefore R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, $R^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$P_{LS} = (H^T R^{-1} H)^{-1} = \left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)^{-1} = \left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)^{-1} = \boxed{\frac{1}{2}\ V^2 = P_{LS}}$

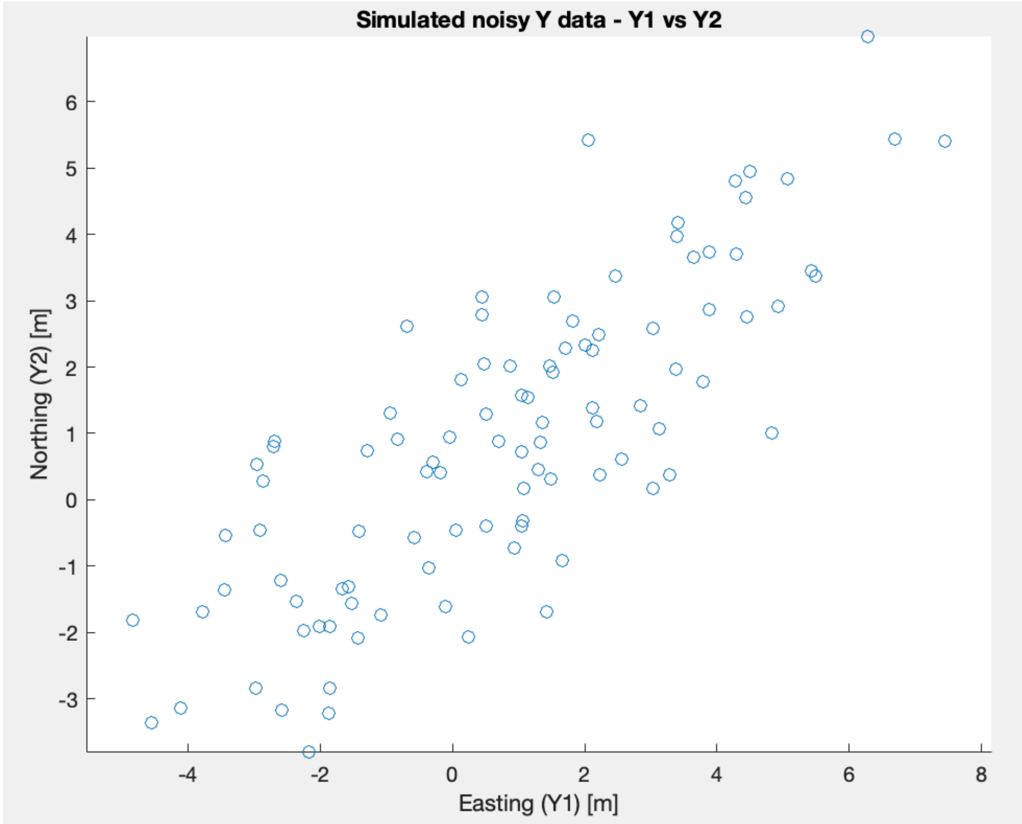
Problem 3 →

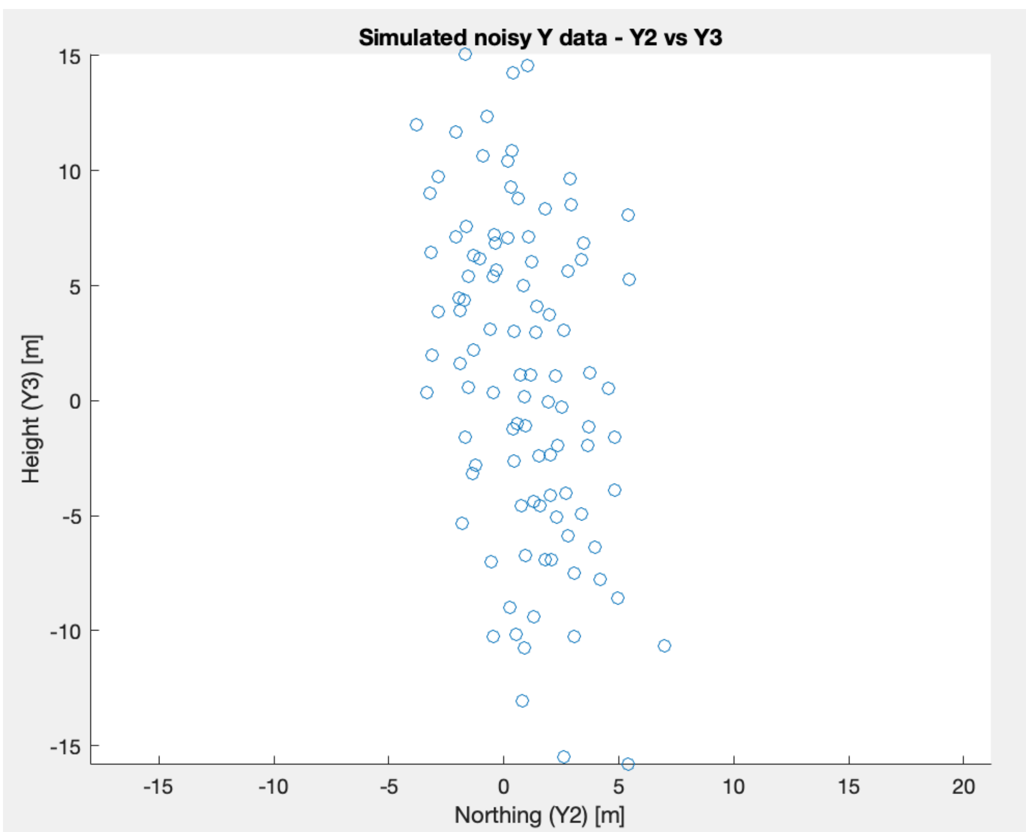
$R = \begin{bmatrix} 8 & 5.15 & 6.5 \\ 5.15 & 5 & -4.07 \\ 6.5 & -4.07 & 50 \end{bmatrix} m^2$, 3D static robot, $x(k+1) = x(k) = \begin{bmatrix} \xi_0 \\ \eta_0 \\ z_0 \end{bmatrix}$

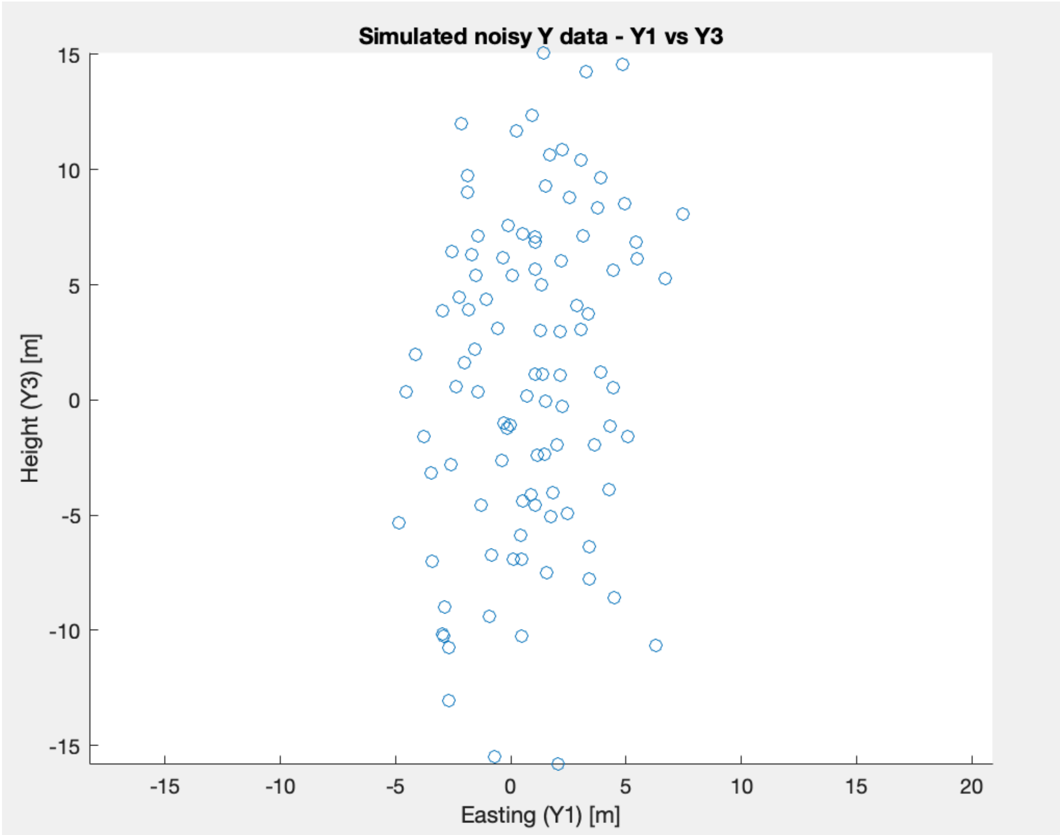
$y_{k+1} = x_{k+1} + v_{k+1} = x_0 + v_{k+1}$

a) $\xi_0 = \eta_0 = z_0 = 1m$, $T = 100$, $v_{k+1} \sim \mathcal{N}(0, R)$, $H = I_{3\times3}$

Using matlab to simulate scatterplots of resulting noisy y data.


Simulated noisy Y data - Y1 vs Y2
Northing (Y2) [m]
Easting (Y1) [m]
6
5
4
3
2
1
0
-1
-2
-3
-4
-2
0
2
4
6
8


Simulated noisy Y data - Y2 vs Y3
Height (Y3) [m]
Northing (Y2) [m]
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15
20



$\boxed{\text{HW 6}}$

→ 5) Using matlab cov() to calculate the sample covariance matrix from

$$\boxed{P = \begin{bmatrix} 7.9748 & 5.5198 & 3.9659 \\ 5.5198 & 5.3162 & -5.1068 \\ 3.9659 & -5.1068 & 48.7969 \end{bmatrix} [m^2]}$$

P is close to R but doesn't exactly match. The magnitudes and signs o

and P match. As $T \to \infty \to P \to R$

c] Using first 3 samples $\rightarrow$ $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ $\leftarrow$ simulated in part a)

$\bar{H} = \begin{bmatrix} H \\ H \\ H \end{bmatrix}$, $\bar{R} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix}$ where R and H are defined in part a)

T = 3 $\rightarrow$

$\hat{x}_{LS} = (\bar{H}^T \bar{R}^{-1} \bar{H})^{-1} \bar{H}^T \bar{R}^{-1} \vec{y}$

Using Matlab for all computations $\rightarrow$ $\boxed{\hat{x}_{LS} = \begin{bmatrix} 2.4136 \\ 1.7088 \\ 4.3021 \end{bmatrix} [m]}$

$P_{LS} = (\bar{H}^T \bar{R}^{-1} \bar{H})^{-1} = \boxed{\begin{bmatrix} 2.6667 & 1.7167 & 2.1667 \\ 1.7167 & 1.6667 & -1.3567 \\ 2.1667 & -1.3567 & 16.6667 \end{bmatrix} [m^2] = P_{LS}}$

Using matlab for 10 samples and 100 samples

T = 10 $\rightarrow$ $\hat{x}_{LS} = \begin{bmatrix} 0.7193 \\ 1.1083 \\ -0.2929 \end{bmatrix} [m]$, $P_{LS} = \begin{bmatrix} 0.8 & 0.5150 & 0.65 \\ 0.5150 & 0.5 & -0.4070 \\ 0.65 & -0.4070 & 5 \end{bmatrix} m^2$

T = 100 $\rightarrow$ $\hat{x}_{LS} = \begin{bmatrix} 0.9530 \\ 1.0718 \\ 0.3203 \end{bmatrix} [m]$, $P_{LS} = \begin{bmatrix} 0.08 & 0.0515 & 0.065 \\ 0.0515 & 0.05 & -0.0407 \\ 0.065 & -0.0407 & 0.5 \end{bmatrix} m^2$

These estimates are somewhat accurate (ground truth is [1; 1; 1])
When running this again in Matlab $\rightarrow$ different results are achieved. If T $\uparrow$ is increased, this gets closer to ground truth.

d] $\bar{H}$ is same as part c) expect length is now for T = 30, same goes for $\bar{R}$.
$\vec{y}$ is given. Using $\rightarrow$ $\hat{x}_{LS} = (\bar{H}^T \bar{R}^{-1} \bar{H})^{-1} \bar{H}^T \bar{R}^{-1} \vec{y}$

$$\boxed{\hat{x}_{LS} = \begin{bmatrix} 4.3935 \\ -16.5495 \\ 42.4159 \end{bmatrix} m, \quad P_{LS} = \begin{bmatrix} 0.2095 & 0.2173 & 0.0594 \\ 0.2173 & 0.2666 & -0.2314 \\ 0.0594 & -0.2314 & 1.5477 \end{bmatrix} m^2}$$

e) $\bar{R}$ is now $I_{90\times 90}$, $\bar{H}$, $\vec{y}$ remain the same (∵ Unweighted least squares)

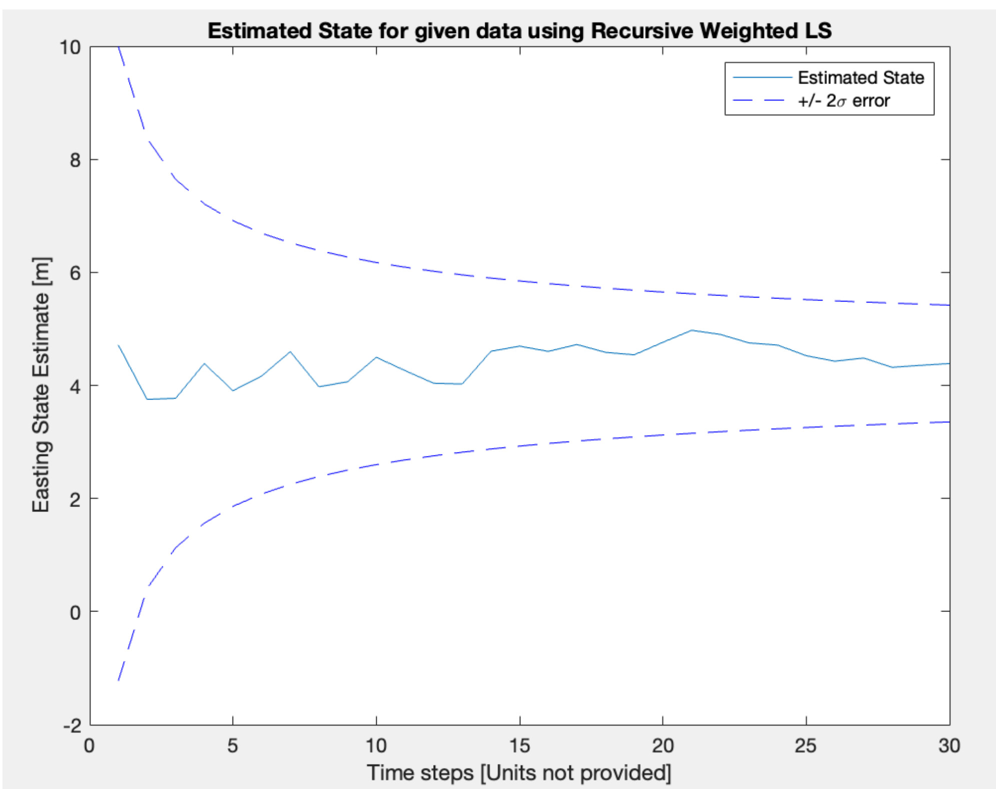
$$\therefore \hat{x}_{LS} = (\bar{H}^T\bar{H})^{-1}\bar{H}^T\vec{y} = \begin{bmatrix} 5.9909 \\ -373.3599 \\ 2.0921\times 10^3 \end{bmatrix} m, \quad P_{LS} = \begin{bmatrix} 0.0333 & 0 & 0 \\ 0 & 0.0333 & 0 \\ 0 & 0 & 0.0333 \end{bmatrix} m^2$$

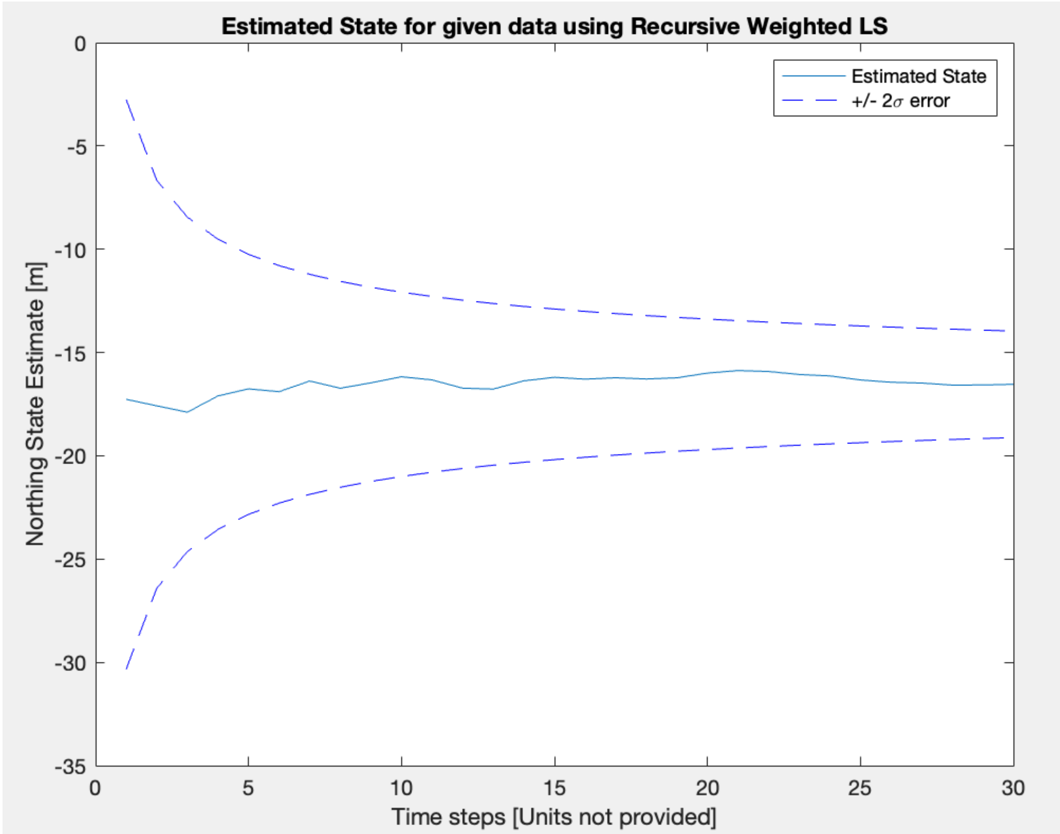
Unweighted P_{LS} < weighted P_{LS}

Unweighted P_{LS} has no cross-correlation terms. But, unweighted $\hat{x}_{LS}$ is greater than weighted $\hat{x}_{LS}$.

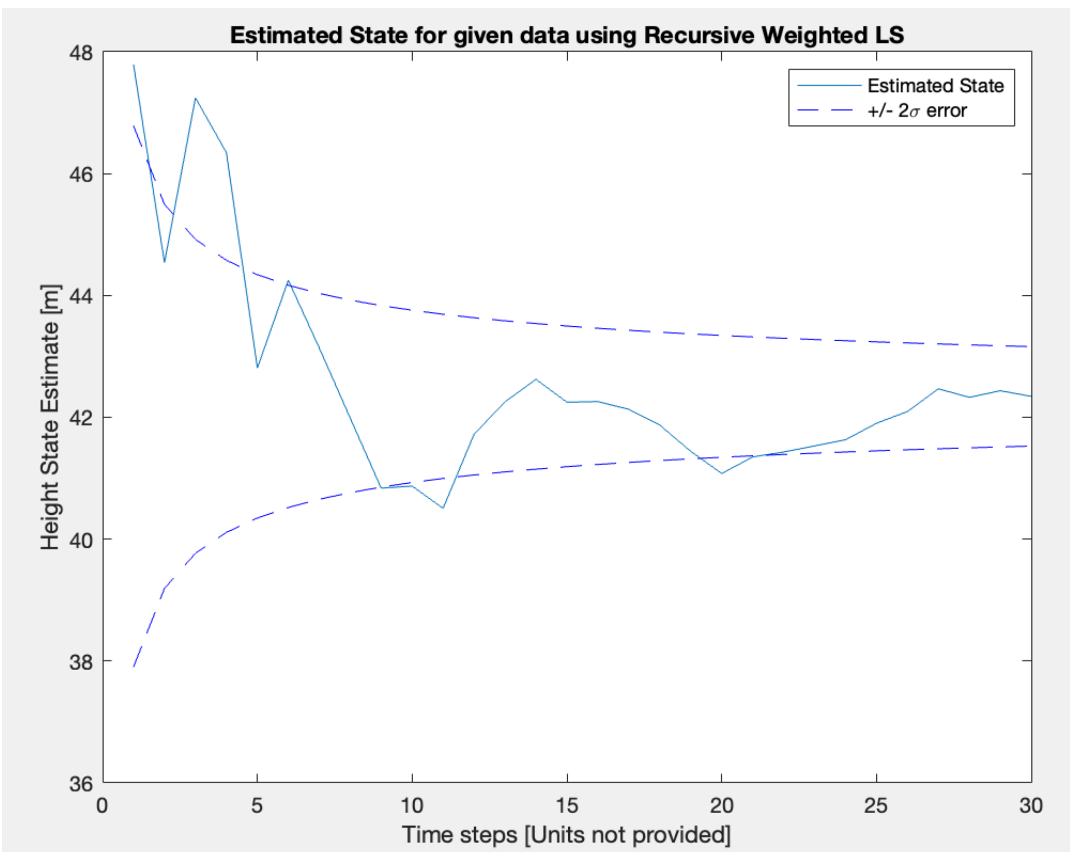
f) Let $x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $P_0 = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$ to use into recursive LS.

$$\left.\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I-K_kH_k)^T + K_kR_kK_k^T \end{aligned}\right\} \rightarrow \text{Implemented in matlab for } k = 1,2,\dots,30$$



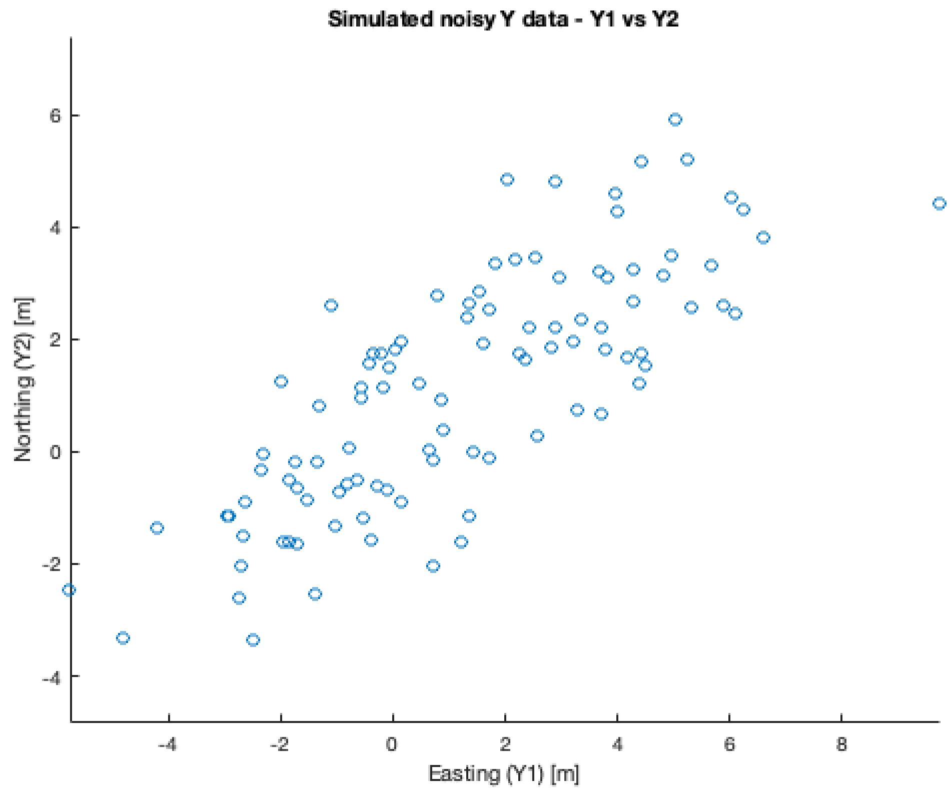


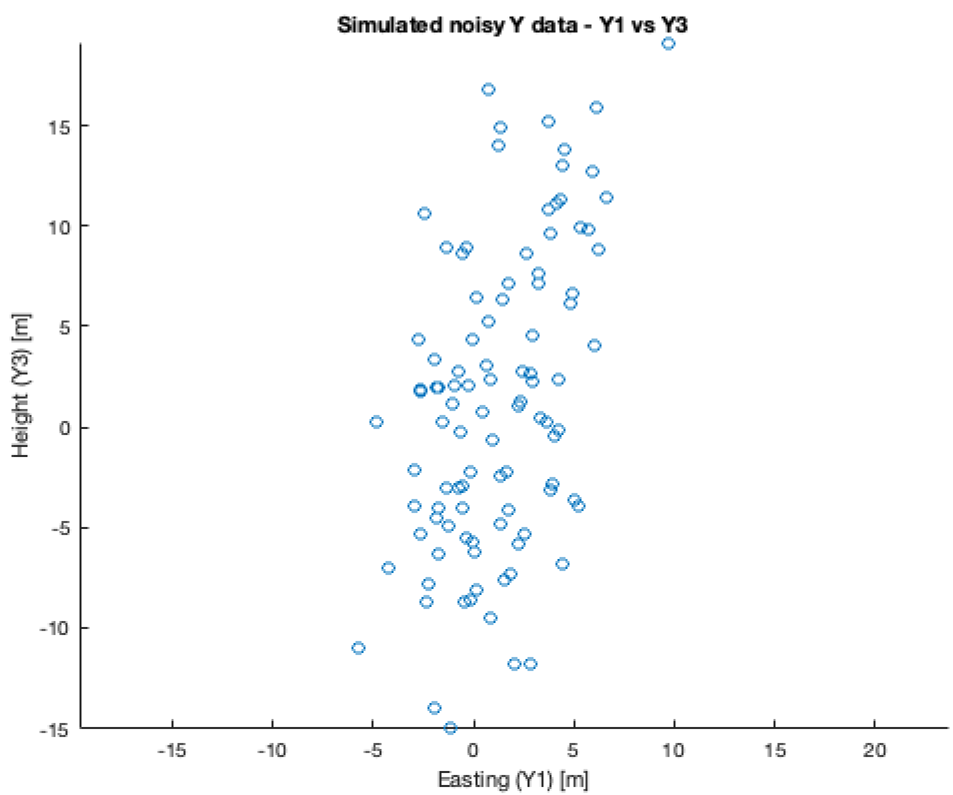
Estimated State for given data using Recursive Weighted LS
Estimated State
+/- 2σ error
Northing State Estimate [m]
Time steps [Units not provided]
0
-5
-10
-15
-20
-25
-30
-35
0
5
10
15
20
25
30

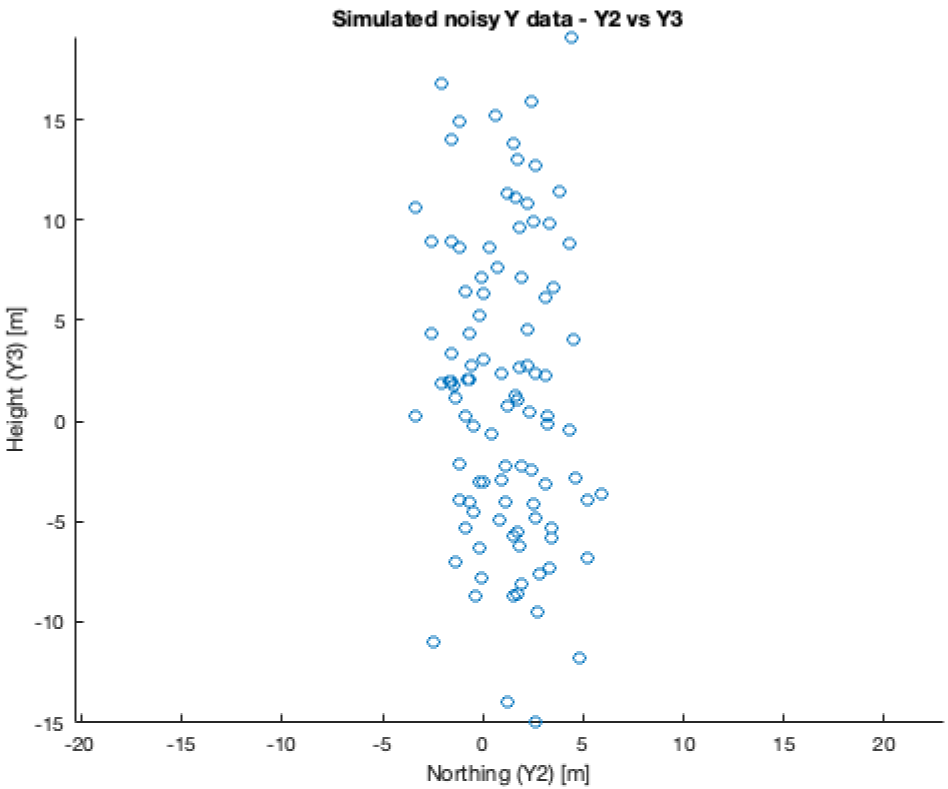


Estimated State for given data using Recursive Weighted LS
Estimated State
+/- 2σ error
Height State Estimate [m]
Time steps [Units not provided]
48
46
44
42
40
38
36
0
5
10
15
20
25
30

```
clear; clc; close all;

% ASEN 5044, Fall 2024
% HW 6, Problem 3
% Jash Bhalavat

% Part a
R = [8, 5.15, 6.5; 5.15, 5, -4.07; 6.5, -4.07, 50];
x0 = [1; 1; 1];
H = eye(3);
mu = [0; 0; 0];
T = 100;
for i = 1:T
    y(:,i) = x0 + mvnrnd(mu, R)';
end

figure()
scatter(y(1,:), y(2,:))
title("Simulated noisy Y data - Y1 vs Y2")
xlabel("Easting (Y1) [m]")
ylabel("Northing (Y2) [m]")
axis equal

figure()
scatter(y(1,:), y(3,:))
title("Simulated noisy Y data - Y1 vs Y3")
xlabel("Easting (Y1) [m]")
ylabel("Height (Y3) [m]")
axis equal

figure()
scatter(y(2,:), y(3,:))
title("Simulated noisy Y data - Y2 vs Y3")
xlabel("Northing (Y2) [m]")
ylabel("Height (Y3) [m]")
axis equal
```


Simulated noisy Y data - Y1 vs Y2
Northing (Y2) [m]
Easting (Y1) [m]
6
4
2
0
-2
-4
-4
-2
0
2
4
6
8


Simulated noisy Y data - Y1 vs Y3
Height (Y3) [m]
Easting (Y1) [m]
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15
20



Part b

```
cov12 = cov(y(1,:), y(2,:));
cov13 = cov(y(1,:), y(3,:));
cov23 = cov(y(2,:), y(3,:));
```

Part c

```
T = 3;
n = length(x0);
[y3, H3, R3] = part_c(T, n, y, H, R);
R3_inv = inv(R3);


P_LS_3 = inv(H3' * R3_inv * H3);
x_hat_3 = P_LS_3 * H3' * R3_inv * y3;


T = 10;
[y10, H10, R10] = part_c(T, n, y, H, R);
R10_inv = inv(R10);

P_LS_10 = inv(H10' * R10_inv * H10);
x_hat_10 = P_LS_10 * H10' * R10_inv * y10;
```

```
T = 100;
[y100, H100, R100] = part_c(T, n, y, H, R);
R100_inv = inv(R100);

P_LS_100 = inv(H100' * R100_inv * H100);
x_hat_100 = P_LS_100 * H100' * R100_inv * y100;

function [y_out, H_out, R_out] = part_c(T, n, y, H, R)
    size_y = size(y);

    % Y is Tp * 1
    p = size_y(1);
    y_out = reshape(y(:,1:T), [T*p, 1]);

    % R is Tp * Tp
    R_out = zeros(T*p);
    for i = 1:T
        lower_bound = n*(i-1) + 1;
        upper_bound = n*i;
        R_out(lower_bound:upper_bound, lower_bound:upper_bound) = R;

        % H is Tp * n
        H_out(lower_bound:upper_bound, :) = H;
    end

end
```

Part d

```
% Read data
data = table2array(readtable("hw6problem3data.csv"));
T = length(data);
cov_12_data = cov(data(1,:), data(2,:));
cov_13_data = cov(data(1,:), data(3,:));
cov_23_data = cov(data(2,:), data(3,:));
R_meas = [cov_12_data [cov_13_data(1,2); cov_23_data(1,2)]; cov_13_data(1,2),
cov_23_data(1,2), cov_13_data(2,2)];

[y_data, H_data, R_data] = part_c(T, n, data, H, R_meas);

R_data_inv = inv(R_data);

P_LS_data = inv(H_data' * R_data_inv * H_data);
x_hat_data = P_LS_data * H_data' * R_data_inv * y_data;
```

Part e

Unweighted LS - R is identity, its inverse is also R_uw

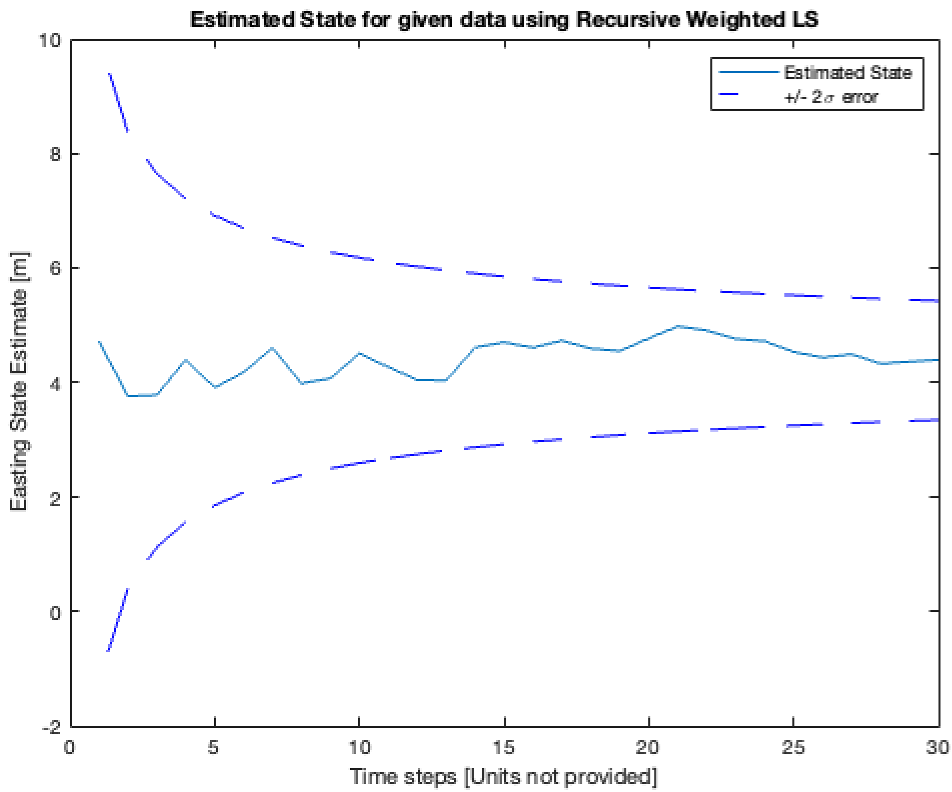
```
R_uw = eye(n*length(data));
```

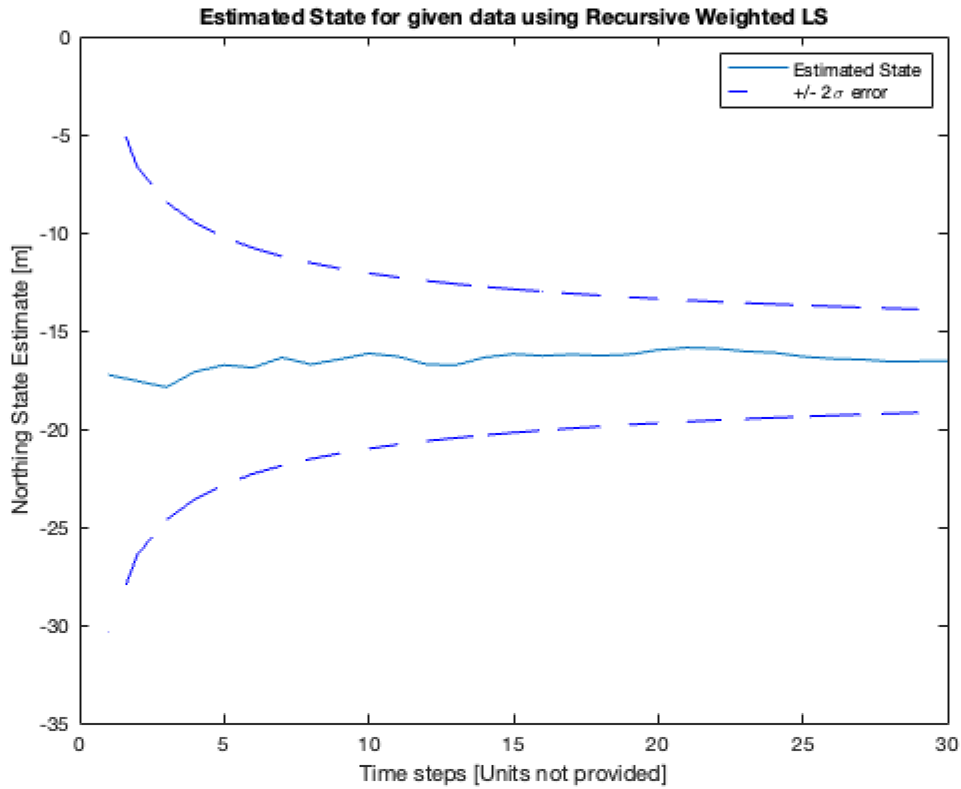
```
P_LS_data_uw = inv(H_data' * R_uw * H_data);
x_hat_data_uw = P_LS_data * H_data' * R_uw * y_data;
```

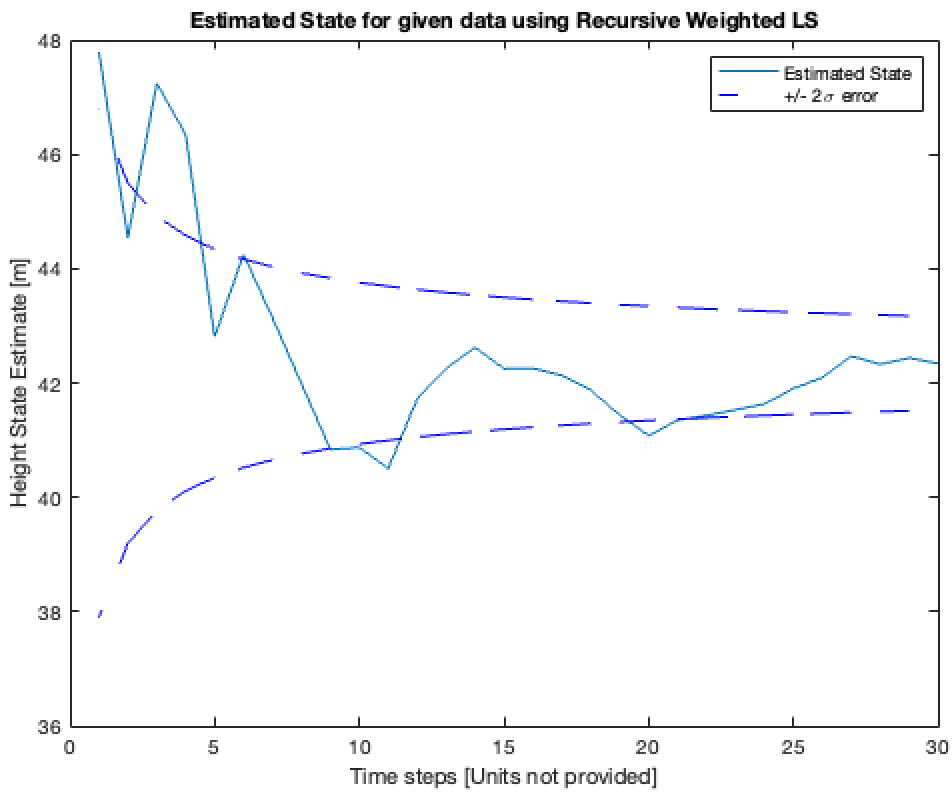
Part f

```
x_hat_0 = x0;
P_0 = eye(3) * 1000;

x_hat = [x_hat_0];
P = [P_0];

for i = 1:T
    lower_bound = 3*(i-1) + 1;
    upper_bound = 3*i;
    next_bound = 3*(i+1);

    P_k_minus_1 = P(lower_bound:upper_bound, :);
    K_k = P_k_minus_1 * H' * inv(H * P_k_minus_1 * H' + R);
    x_hat(:,i+1) = x_hat(:,i) + K_k * (data(:,i) - H * x_hat(:,i));
    P(upper_bound+1:next_bound, :) = (eye(3) - H*K_k) * P_k_minus_1 * (eye(3)
- H*K_k)' + K_k * R * K_k';

    sigma_x1_2(i) = sqrt(P(upper_bound+1, 1)) * 2;
    sigma_x3_2(i) = sqrt(P(upper_bound+2, 2)) * 2;
    sigma_x2_2(i) = sqrt(P(upper_bound+3, 3)) * 2;
end


k = 1:30;

figure()
plot(k, x_hat(1, 2:end))
hold on
plot(k, sigma_x1_2 + x_hat(1,end), 'b--')
plot(k, -sigma_x1_2 + x_hat(1,end), 'b--')
hold off
xlabel("Time steps [Units not provided]")
ylabel("Easting State Estimate [m]")
title("Estimated State for given data using Recursive Weighted LS")
legend("Estimated State", "+/- 2\sigma error")

figure()
plot(k, x_hat(2, 2:end))
hold on
plot(k, sigma_x2_2 + x_hat(2,end), 'b--')
plot(k, -sigma_x2_2 + x_hat(2,end), 'b--')
hold off
xlabel("Time steps [Units not provided]")
ylabel("Northing State Estimate [m]")
title("Estimated State for given data using Recursive Weighted LS")
legend("Estimated State", "+/- 2\sigma error")

figure()
```

```
plot(k, x_hat(3, 2:end))
hold on
plot(k, sigma_x3_2 + x_hat(3,end), 'b--')
plot(k, -sigma_x3_2 + x_hat(3,end), 'b--')
hold off
xlabel("Time steps [Units not provided]")
ylabel("Height State Estimate [m]")
title("Estimated State for given data using Recursive Weighted LS")
legend("Estimated State", "+/- 2\sigma error")
```




Estimated State for given data using Recursive Weighted LS
Estimated State
+/- 2σ error
Northing State Estimate [m]
Time steps [Units not provided]
0
-5
-10
-15
-20
-25
-30
-35
0
5
10
15
20
25
30


Estimated State for given data using Recursive Weighted LS
Estimated State
+/- 2σ error
Height State Estimate [m]
Time steps [Units not provided]
48
46
44
42
40
38
36
0
5
10
15
20
25
30