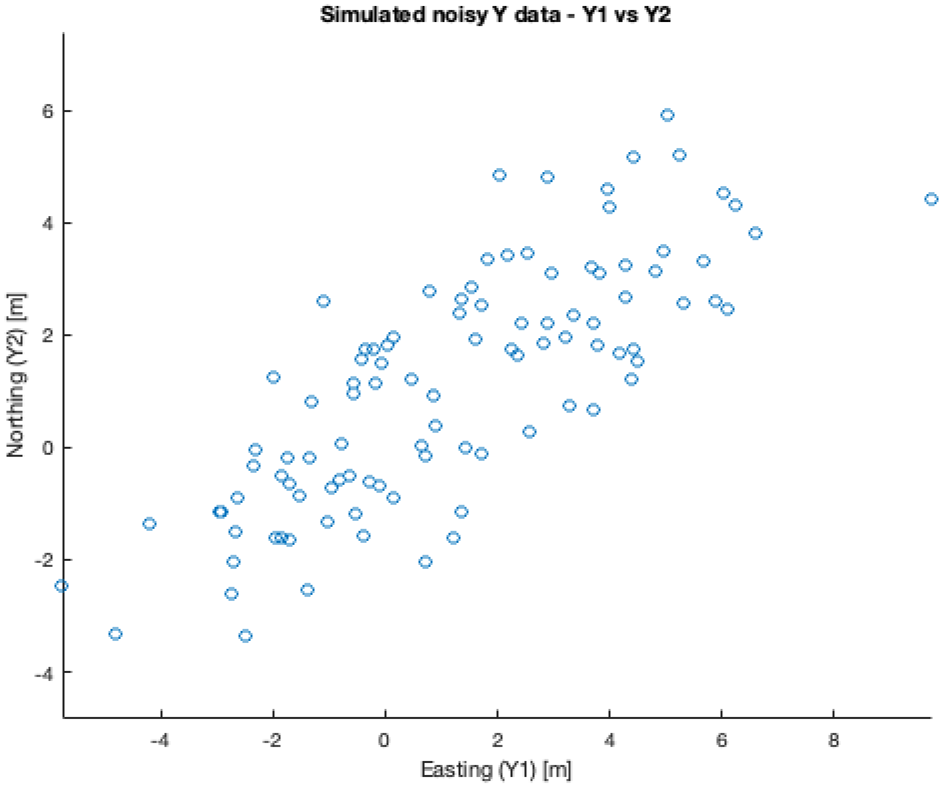
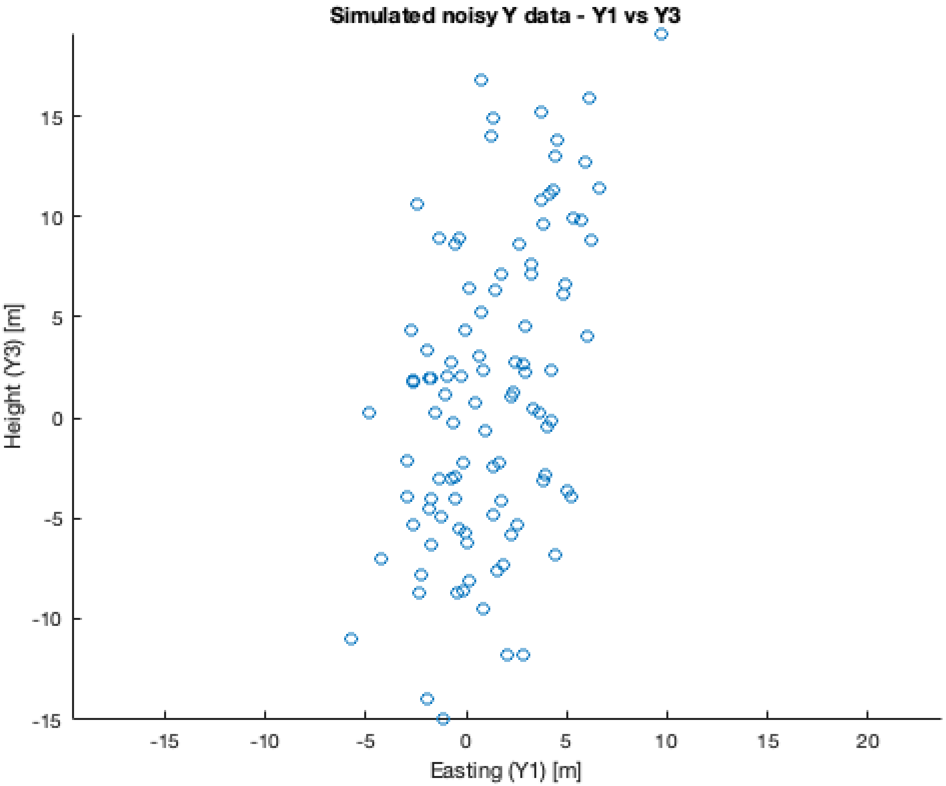
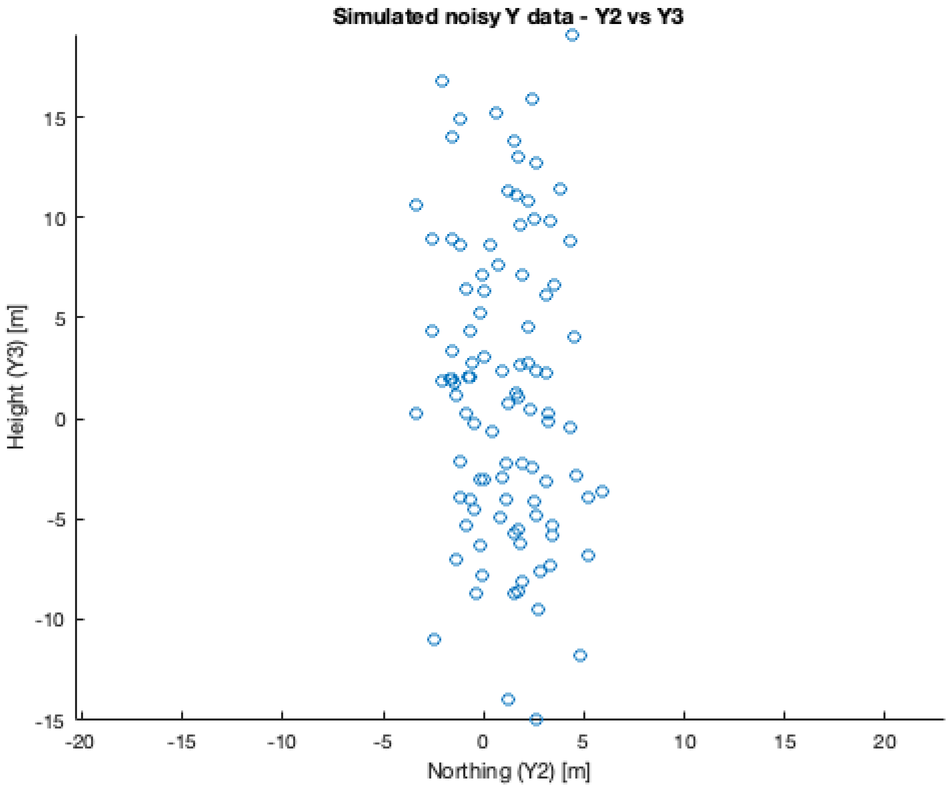
```
clear; clc; close all;

% 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

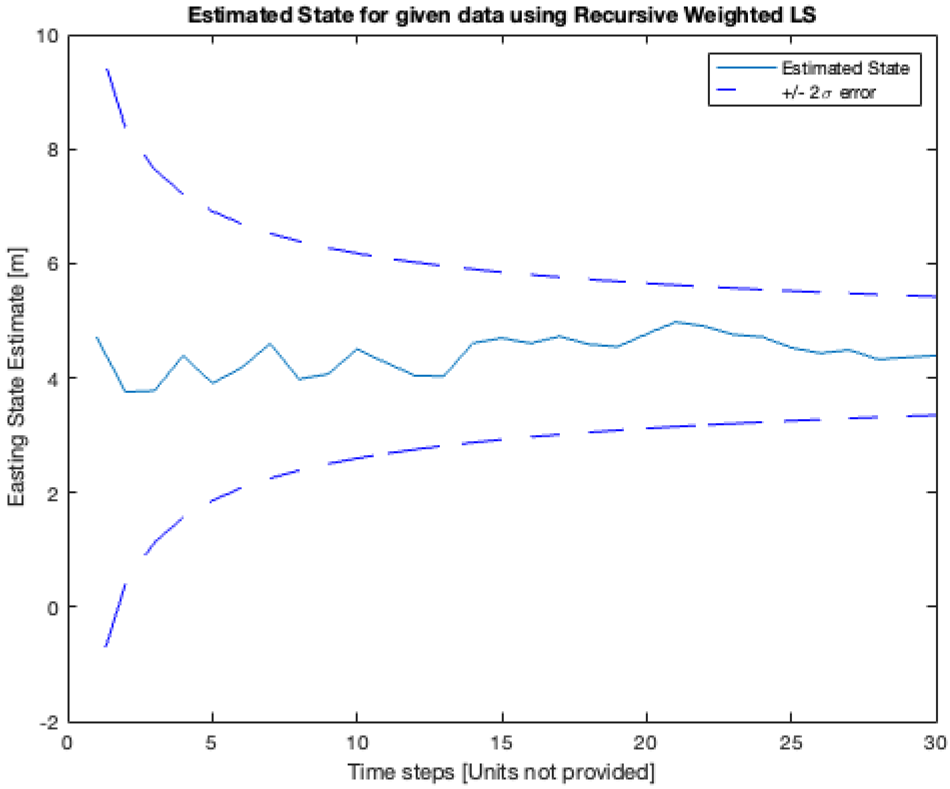
```
R_uw = eye(n*length(data));
```

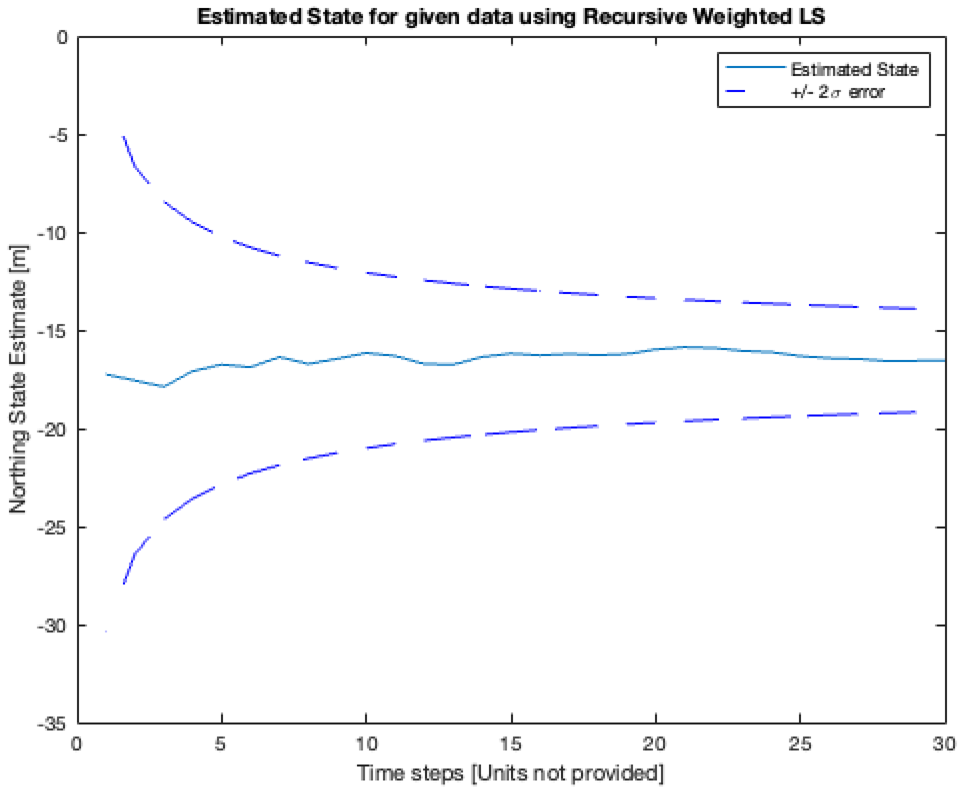
```
P_LS_data_uw = inv(H_data' * R_uw * H_data);
x_hat_data_uw = P_LS_data * H_data' * R_uw * y_data;
```

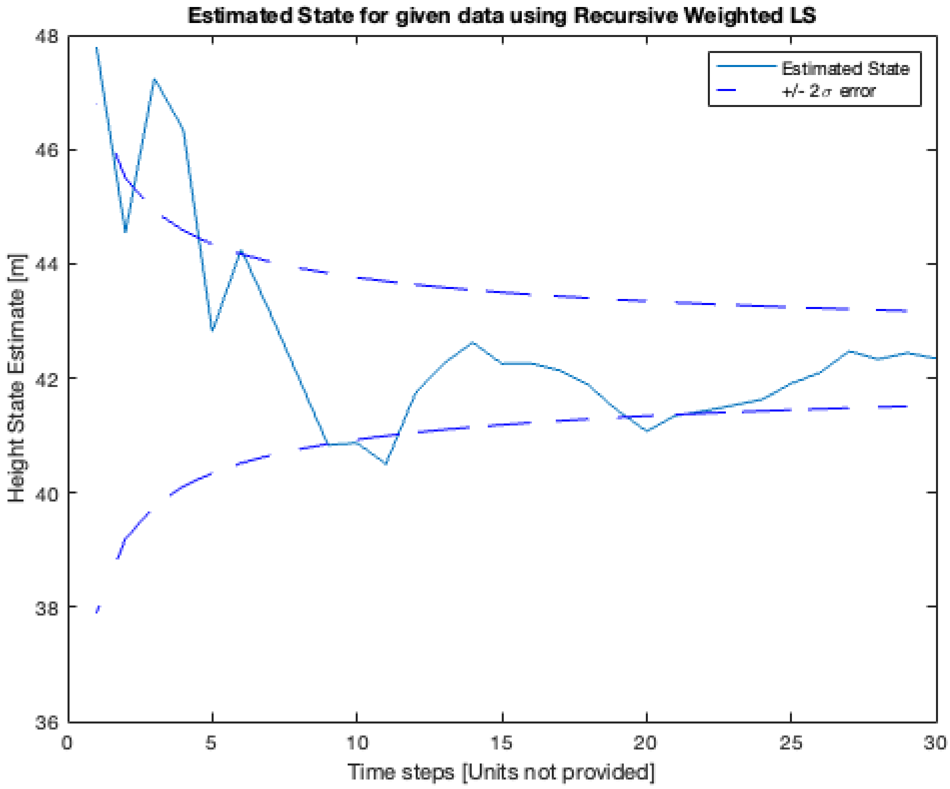
Part f

```
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]
0
-5
-10
-15
-20
-25
-30
-35
0
5
10
15
20
25
30
Time steps [Units not provided]


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

Published with MATLAB® R2024a