ASEN 5050 HW 6 Main Script

Table of Contents

By: Ian Faber

Housekeeping

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

Problem 1

Part a

```
A = [
            0   1;
         -100 -10
    ];

B = zeros(2,1);
C = eye(2);
D = zeros(2,1);
Gamma = [
             0;
             1
         ];
W = 10;

% Part b
dT = 0.2; % sec
    % Discretize stochastic part
z = dT*[-A Gamma*W*Gamma'; zeros(size(A)) A'];
matExp = expm(z);
F = matExp(3:4, 3:4)'
Q = F*matExp(1:2, 3:4)
    % Discretize deterministic part
Ahat = [A B; zeros((size(A,2) + size(B,2))-size(A,1), size(A,2) +
size(B,2))];
matExp2 = expm(Ahat*dT);
epsilon = 1e-8;
if abs(matExp2(1:2,1:2) - F) < epsilon*ones(size(F)); fprintf("F matrix
matches!\n"); else; fprintf("F doesn't match...\n"); return; end
G = matExp2(1:2,3)

% Part d
```

```
R = 3/0.2
H = [1 0.2];


F =

    0.1506    0.0419
   -4.1928   -0.2687


Q =

    0.0040    0.0088
    0.0088    0.3760

F matrix matches!

G =

     0
     0


R =

    15
```

Problem 3

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

    % Part a
R = Rgiven;
numMeas = 100;
Sv = chol(R, 'lower')
my = [1; 1; 1]; % easting - northing - height
q = mvnrnd(zeros(3,1), eye(3), numMeas)';

y = my + Sv*q;

figure;
lims = [min(y([1 2], :), [], 'all'), max(y([1 2], :), [], 'all')];
hold on; grid on;
title("y_k(1) vs. y_k(2)")
plot(y(1,:), y(2,:), '.')
xlabel("y_k(1) [m]"); ylabel("y_k(2) [m]")
xlim(lims); ylim(lims)
```

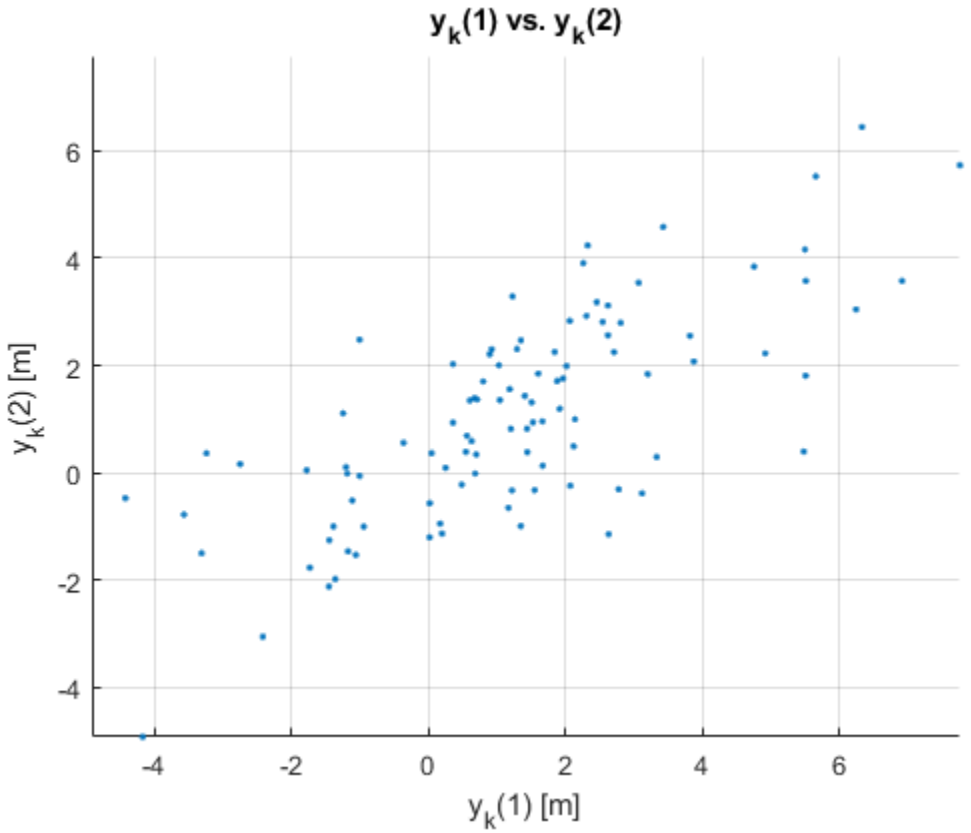
```
figure;
lims = [min(y([1 3], :), [], 'all'), max(y([1 3], :), [], 'all')];
hold on; grid on;
title("y_k(1) vs. y_k(3)")
plot(y(1,:), y(3,:), '.')
xlabel("y_k(1) [m]"); ylabel("y_k(3) [m]")
xlim(lims); ylim(lims)

figure;
lims = [min(y([2 3], :), [], 'all'), max(y([2 3], :), [], 'all')];
hold on; grid on;
title("y_k(2) vs. y_k(3)")
plot(y(2,:), y(3,:), '.')
xlabel("y_k(2) [m]"); ylabel("y_k(3) [m]")
xlim(lims); ylim(lims)

    % Part b
bigY = reshape(y, size(y,1)*size(y,2), 1);
bigH = repmat(eye(3), [size(y,2), 1]);
bigR = [];
for k = 1:length(y)
    bigR = blkdiag(bigR, R);
end

C = cov(y') - my*my'

    % Part c
for k = [3 10 numMeas]
    idx = 1:k*size(y,1);
    Y = bigY(idx);
    H = bigH(idx,:);
    R = bigR(idx, idx);
    xLS = ((H'*(R^-1)*H)^-1)*H'*(R^-1)*Y
    Pls = (H'*(R^-1)*H)^-1
end

    % Part d
y = load("hw6problem3data.csv");
R = Rgiven;

bigY = reshape(y, size(y,1)*size(y,2), 1);
bigH = repmat(eye(3), [size(y,2), 1]);
bigR = [];
for k = 1:length(y)
    bigR = blkdiag(bigR, R);
end

xLS = ((bigH'*(bigR^-1)*bigH)^-1)*bigH'*(bigR^-1)*bigY
Pls = (bigH'*(bigR^-1)*bigH)^-1

    % Part e
R = eye(3);

bigY = reshape(y, size(y,1)*size(y,2), 1);
```

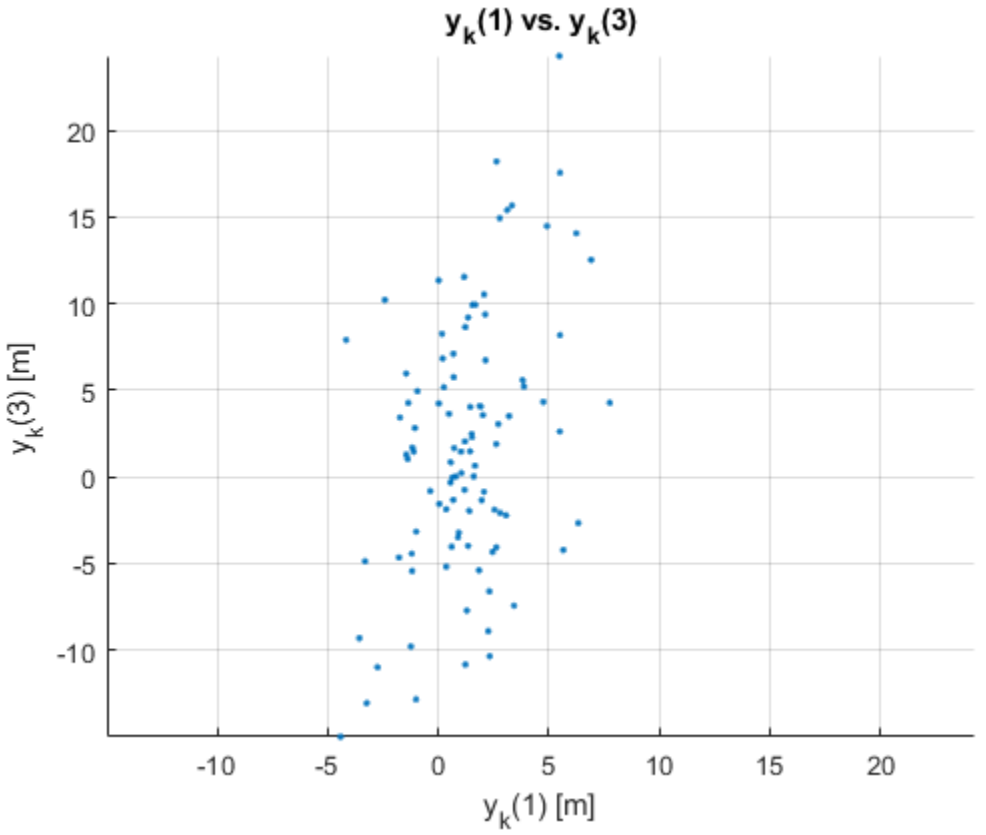
```
bigH = repmat(eye(3), [size(y,2), 1]);
expV = [];
for k = 1:length(y)
    expV = blkdiag(expV, Rgiven);
end

xLS = ((bigH'*bigH)^-1)*bigH'*bigY
Pls = ((bigH'*bigH)^-1)*bigH'*expV*bigH*((bigH'*bigH)^-1)

    % Part f
R = Rgiven;

xEst = [];
Pest = [];
xLS = zeros(3,1);
Pls = 1000*eye(3);
H = eye(3);
for k = 1:length(y)
    % Propagate estimator
    K = Pls*H'*((R + H*Pls*H')^-1);
    xLS = xLS + K*(y(:,k) - H*xLS);
    Pls = (eye(3) - K*H)*Pls*((eye(3) - K*H)') + K*R*K';

    % Save most recent estimate
    xEst = [xEst, xLS];
    Pest = [Pest, Pls];
end

for k = 1:length(y)
    xTrue = xEst(:,k);
    sigma1 = sqrt(Pest(1,3*(k-1)+1));
    sigma2 = sqrt(Pest(2,3*(k-1)+2));
    sigma3 = sqrt(Pest(3,3*(k-1)+3));
    xPlus2sig(:,k) = xTrue + 2*[sigma1; sigma2; sigma3];
    xMin2sig(:,k) = xTrue - 2*[sigma1; sigma2; sigma3];
end


    % plot states
figure;
hold on; grid on;
title("x^1_{LS} vs. k")
estimate = plot(xEst(1,:), 'b-');
bound = plot(xPlus2sig(1,:), 'r--');
plot(xMin2sig(1,:), 'r--')
truth = plot(xTrue(1)*ones(length(y),1), 'k:');
xlabel("k"); ylabel("x^1_{LS}")
legend([estimate, bound, truth], ["Estimated state", "2 sigma bound",
"'Truth' state from part d"])

figure;
hold on; grid on;
title("x^2_{LS} vs. k")
estimate = plot(xEst(2,:), 'b-');
```

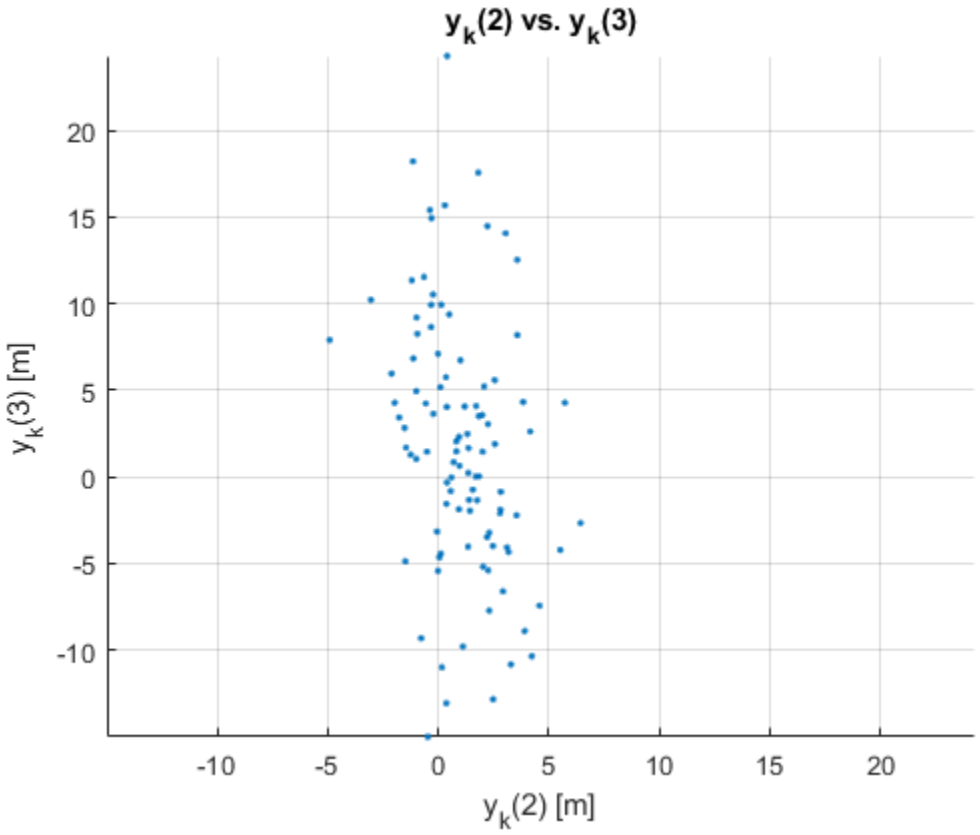
```
bound = plot(xPlus2sig(2,:), 'r--');
plot(xMin2sig(2,:), 'r--')
truth = plot(xTrue(2)*ones(length(y),1), 'k:');
xlabel("k"); ylabel("x^2_{LS}")
legend([estimate, bound, truth], ["Estimated state", "2 sigma bound",
"'Truth' state from part d"])

figure;
hold on; grid on;
title("x^3_{LS} vs. k")
estimate = plot(xEst(3,:), 'b-');
bound = plot(xPlus2sig(3,:), 'r--');
plot(xMin2sig(3,:), 'r--')
truth = plot(xTrue(3)*ones(length(y),1), 'k:');
xlabel("k"); ylabel("x^3_{LS}")
legend([estimate, bound, truth], ["Estimated state", "2 sigma bound",
"'Truth' state from part d"])
```

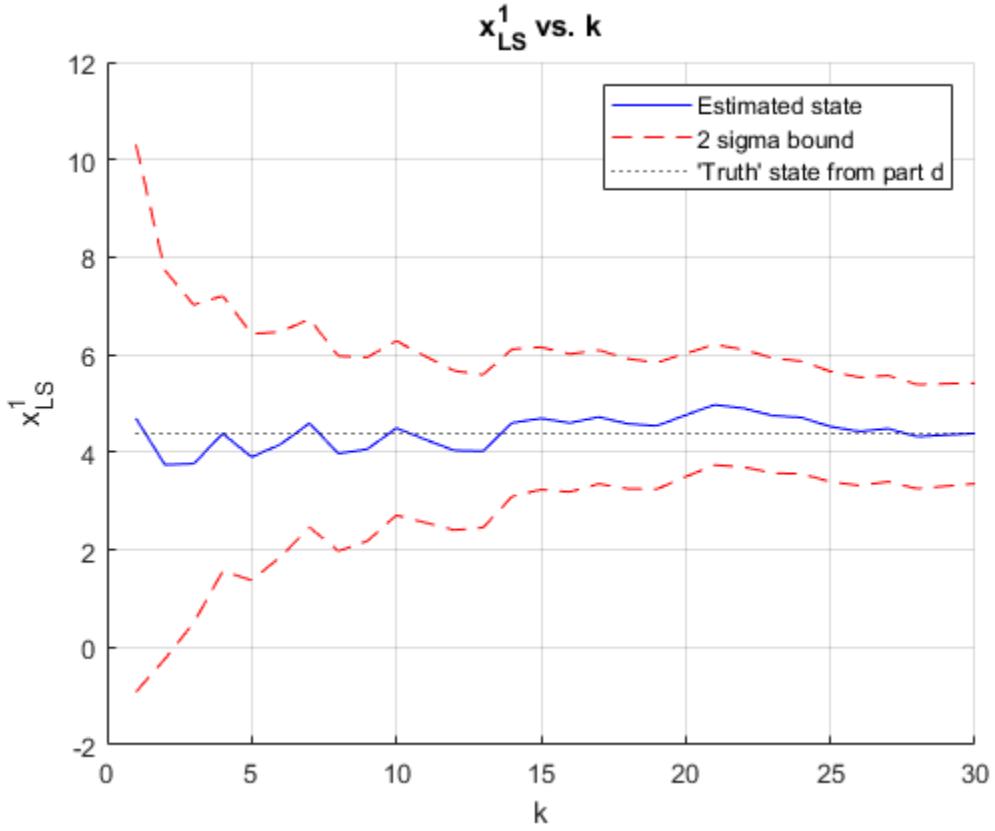
```
Sv =

    2.8284         0         0
    1.8208    1.2980         0
    2.2981   -6.3595    2.0677


C =

    4.7428    2.3768    6.2251
    2.3768    2.6961   -5.2708
    6.2251   -5.2708   54.0997


xLS =

   -0.7312
   -0.3455
   -1.0062


Pls =

    2.6667    1.7167    2.1667
    1.7167    1.6667   -1.3567
    2.1667   -1.3567   16.6667


xLS =

    0.4872
    0.6746
   -0.5582
```

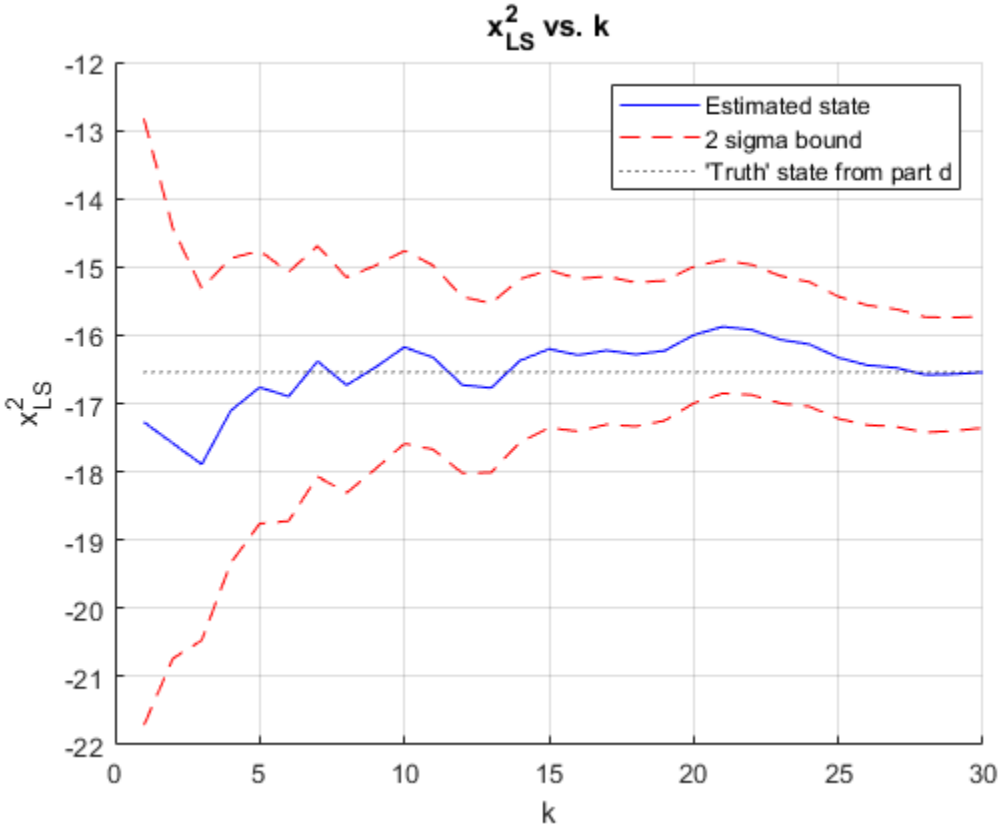
```
Pls =

    0.8000    0.5150    0.6500
    0.5150    0.5000   -0.4070
    0.6500   -0.4070    5.0000


xLS =

    1.2655
    1.0653
    1.8511


Pls =

    0.0800    0.0515    0.0650
    0.0515    0.0500   -0.0407
    0.0650   -0.0407    0.5000


xLS =

    4.3935
  -16.5495
   42.4159


Pls =

    0.2667    0.1717    0.2167
    0.1717    0.1667   -0.1357
    0.2167   -0.1357    1.6667


xLS =

    4.3935
  -16.5495
   42.4159


Pls =

    0.2667    0.1717    0.2167
    0.1717    0.1667   -0.1357
    0.2167   -0.1357    1.6667
```

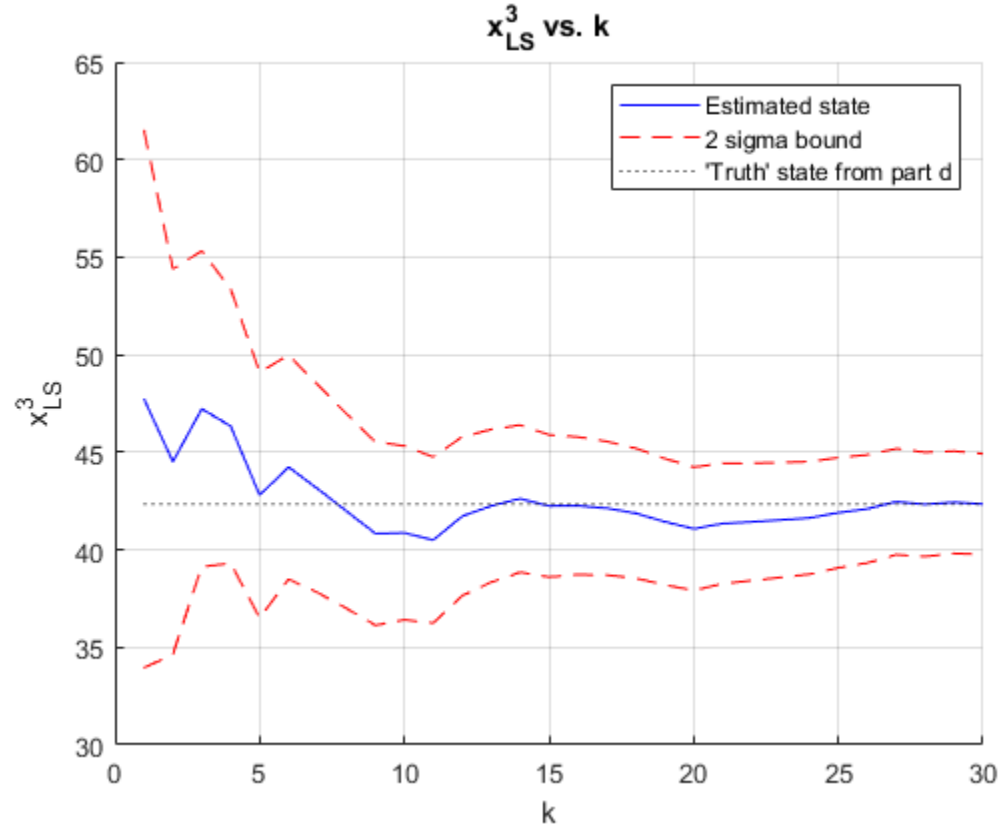

$y_k(1)$ vs. $y_k(2)$
$y_k(2)$ [m]
$y_k(1)$ [m]
6
4
2
0
-2
-4
-4
-2
0
2
4
6


$y_k(1)$ vs. $y_k(3)$
$y_k(3)$ [m]
$y_k(1)$ [m]
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20


$y_k(2)$ vs. $y_k(3)$
$y_k(3)$ [m]
$y_k(2)$ [m]
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20


x^1_{LS} vs. k
Estimated state
2 sigma bound
'Truth' state from part d
x^1_{LS}
k
12
10
8
6
4
2
0
-2
0
5
10
15
20
25
30


x^2_{LS} vs. k
Estimated state
2 sigma bound
'Truth' state from part d
x^2_{LS}
k


x^3_{LS} vs. k
Estimated state
2 sigma bound
'Truth' state from part d
x^3_{LS}
k

Published with MATLAB® R2023b