Table of Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     ASEN 5070-Problem 4
%
% Zach Dischner
%   Exam 3
%       Problem 4
%
%
%
% Solves and answers questions relating to problem 4 of the STATOD final
% exam
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Prepare Workspace

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

Load Data

```
tmp=load('Exam3_Problem4_data.txt');
```

Form into useful datasets

```
for ii=1:8
    data(ii).xy = tmp(ii,2:3);
    x(ii) = tmp(ii,2);
    y(ii) = tmp(ii,3);
    data(ii).P  = reshape(tmp(ii,4:end),2,2);
    data(ii).sigma = data(ii).P(1,1)^.5;
end
```

a - Find best guess of xhat, and covariance matrix

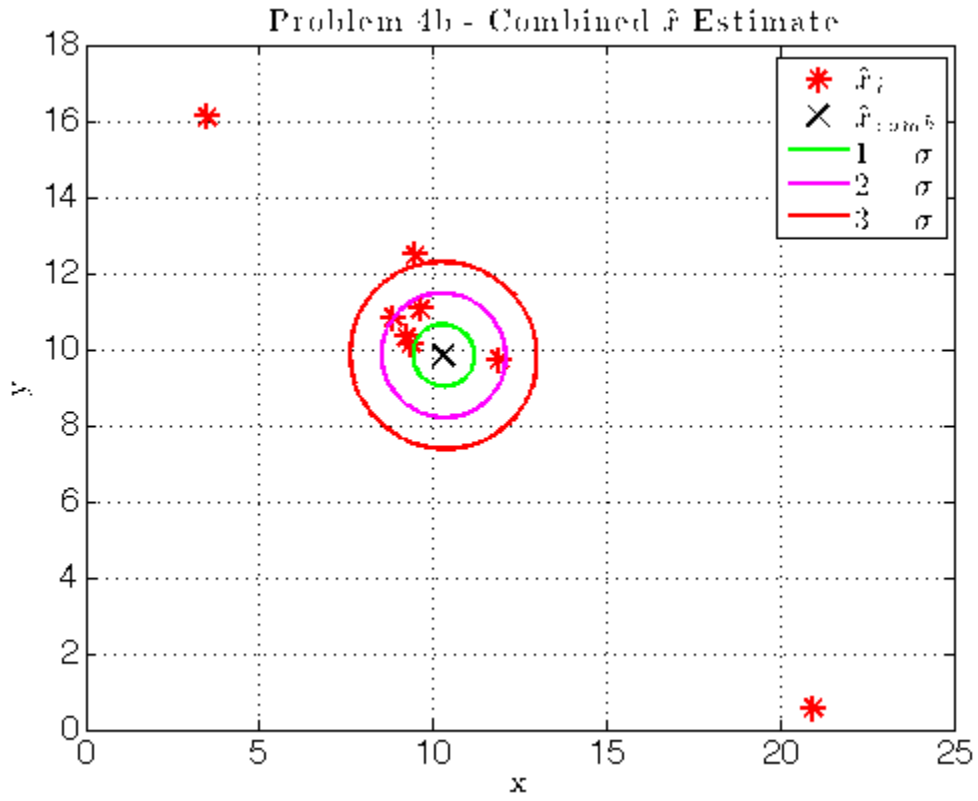
```
sumP_inv=0;
sumx=0;
xhat = [0 0]';
for ii=1:8
    sumP_inv = sumP_inv + inv(data(ii).P);
    sumx = sumx + inv(data(ii).P)*data(ii).xy';
end

xhat = inv(sumP_inv)*sumx;
Pfinal = inv(sumP_inv);

fprintf('The guess for xhat is: [%3.3f,%3.3f]',xhat(1),xhat(2))
fprintf('\n\nWith a covariance matrix of:  [%3.3f,%3.3f]',Pfinal(1),Pfinal(2))
fprintf('\n.............................[%3.3f,%3.3f]\n',Pfinal(3),Pfinal(4))
```

```
The guess for xhat is: [10.300,9.848]

With a covariance matrix of:  [0.813,-0.083]
.............................[-0.083,0.675]
```

b - Plot points and ellipses

```
figure
plot(x,y,'r*','MarkerSize',12)
hold on
for ii = 1:8
%     error_ellipse(data(ii).P,[data(ii).xy]);
end
plot(xhat(1),xhat(2),'xk','MarkerSize',15)

cov_ellipse(Pfinal,xhat(1),xhat(2),'G');
cov_ellipse(Pfinal*2^2,xhat(1),xhat(2),'m');
cov_ellipse(Pfinal*3^2,xhat(1),xhat(2),'r');
% error_ellipse(Pfinal,xhat)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','green')
%
%
% error_ellipse(Pfinal*4,xhat)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','magenta')
%
% error_ellipse(Pfinal*9,xhat)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','red')

legend('$\hat{x}_i$','$\hat{x}_{comb}$','$1-\sigma$','$2-\sigma$','$3-\sigma$')
xlabel('x');ylabel('y');title('Problem 4b - Combined $\hat{x}$ Estimate')
```

```
% 1sigma, 2sigma,3sigma are given by :
%   error_ellipse(P*1)
%   error_ellipse(P*4)
%   error_ellipse(P*9)
```



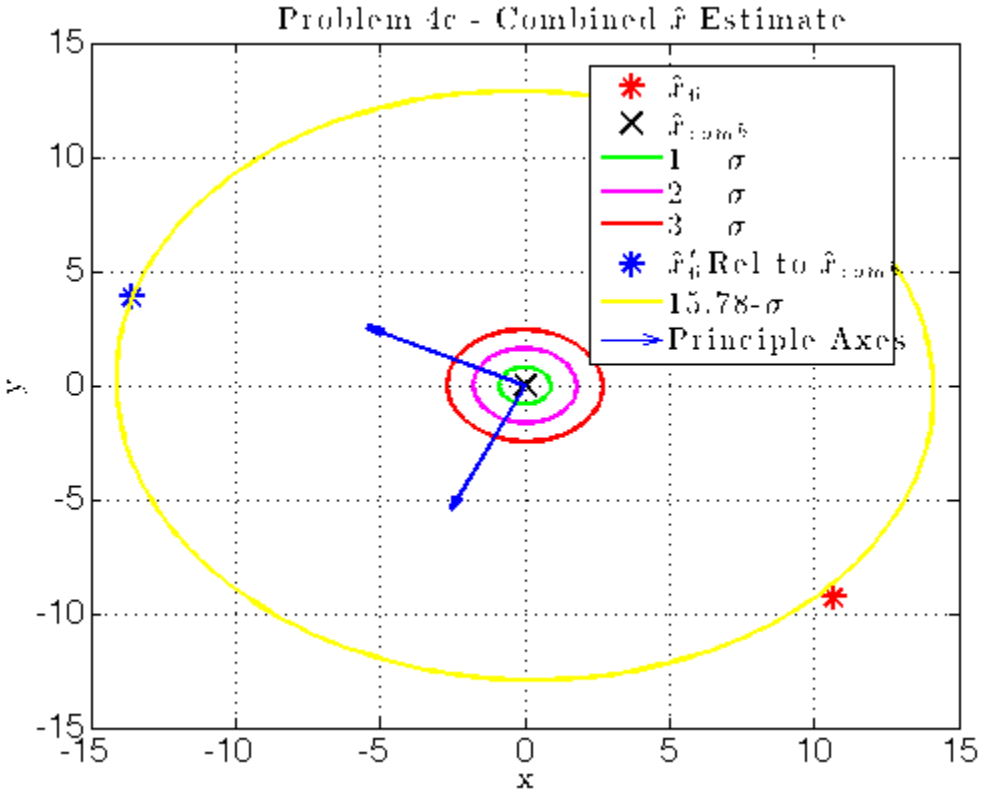
C - Look at x_6

```
xhat2 = xhat - xhat;
xhat6 =[x(6),y(6)]' - xhat;
figure
plot(xhat6(1),xhat6(2),'r*','MarkerSize',12)
hold on
plot(xhat2(1),xhat2(2),'xk','MarkerSize',15)

% error_ellipse(Pfinal,xhat2)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','green')
%
% error_ellipse(Pfinal*4,xhat2)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','magenta')
%
% error_ellipse(Pfinal*9,xhat2)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','red')
```

```
cov_ellipse(Pfinal,xhat2(1),xhat2(2),'G');
cov_ellipse(Pfinal*2^2,xhat2(1),xhat2(2),'m');
cov_ellipse(Pfinal*3^2,xhat2(1),xhat2(2),'r');

a= Pfinal(1,1)^.5;
b=Pfinal(2,2)^.5;


% Get U, the rotation vector
% xhat6=[x(6),y(6)]';
[vec,val]=eigs(Pfinal);
U=vec;
%Rotate point into principle axis
xhat6prime = U'*xhat6;
% xhat6prime = -[xhat6prime(2),xhat6prime(1)]' + xhat;
plot(xhat6prime(1) ,xhat6prime(2),'b*','Markersize',12)

% Scale
a= Pfinal(1,1)^.5;
b=Pfinal(2,2)^.5;

% delsigx = (xhat6prime(1))/a;
% delsigy = (xhat6prime(2))/b;
%
% nsigma = sqrt(delsigx^2 + delsigy^2);


% error_ellipse(Pfinal*13.157268^2,xhat2)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','yellow')


phi = atan2(U(2,1),U(1,1));
% dist = norm(xhat6-xhat2);
% delx = dist*U(1,1);
% dely = dist*(U(1,2));
% delsigx = delx/a;
% delsigy=dely/b;
% delsigma = norm([delsigx delsigy]);

th=atan2(xhat6prime(2),xhat6prime(1));
rel=th-phi;
d=norm(xhat6prime);
delx=d*cos(rel);
dely=d*sin(rel);
delxstd=delx/a;
delystd=dely/b;

delstd = sqrt(delxstd^2 + delystd^2);

cov_ellipse(Pfinal*delstd^2,xhat2(1),xhat2(2),'y')

% plot principle axis
quiver([xhat2(1),xhat2(1)]',[xhat2(2) xhat2(2)]',vec(:,2),vec(:,1),6)
```

```
legend('$\hat{x}_6$','$\hat{x}_{comb}$','$1-\sigma$','$2-\sigma$','$3-\sigma$','$\
xlabel('x');ylabel('y');title('Problem 4c - Combined $\hat{x}$ Estimate')

fprintf('\n\nxhat6 is %3.3f STD from xhat in X',delxstd);
fprintf('\nxhat6 is %3.3f STD from xhat in Y',delystd);
fprintf('\nFor a total sigma distance of %3.3f\n\n',delstd)
```

xhat6 is 15.455 STD from xhat in X
xhat6 is 2.774 STD from xhat in Y
For a total sigma distance of 15.702

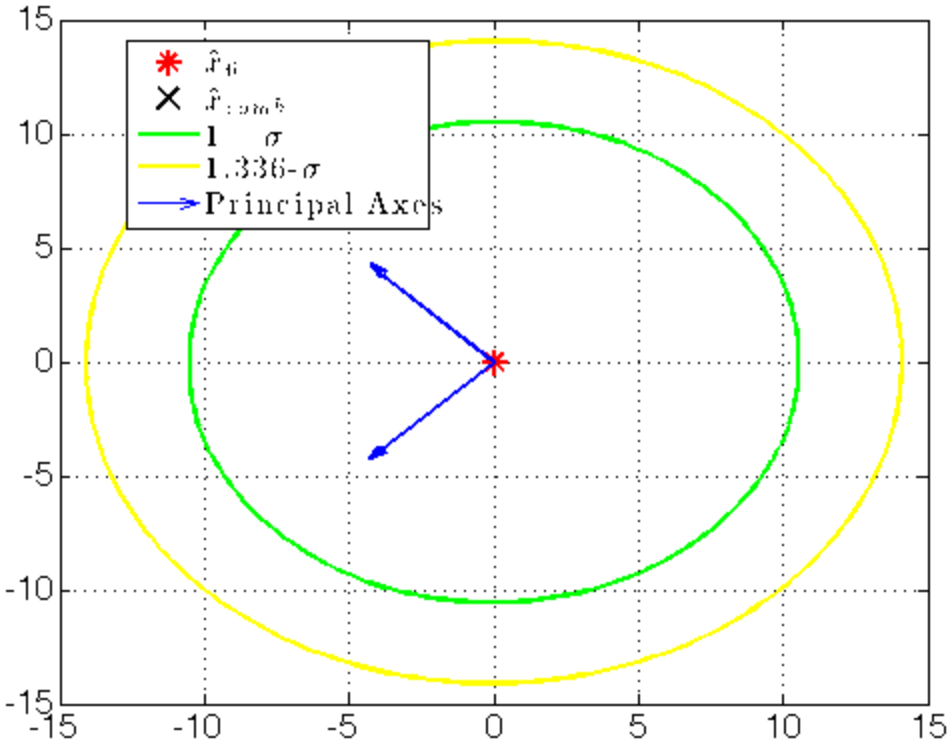


d - find dist from x6 to xhat

```
xhat6 =[x(6),y(6)]' - [x(6),y(6)]';
xhat2 = xhat - [x(6);y(6)];

P6 = data(6).P;
figure
plot(xhat6(1),xhat6(2),'r*','MarkerSize',12)
hold on
plot(xhat2(1),xhat2(2),'xk','MarkerSize',15)
```

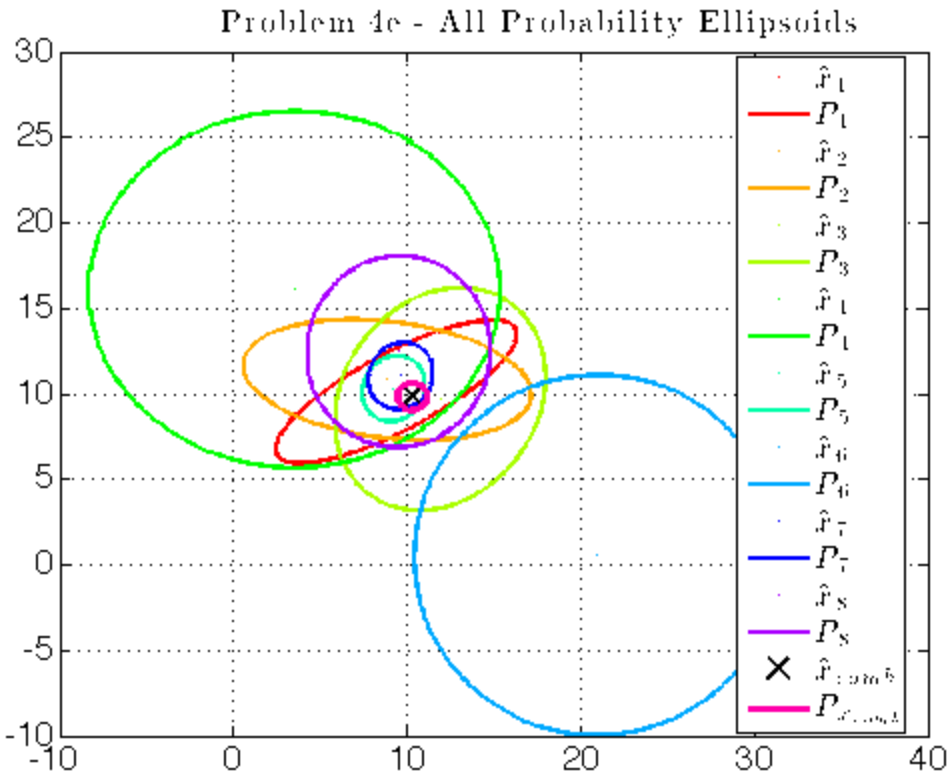
```
cov_ellipse(P6,xhat6(1),xhat6(2),'g')
% error_ellipse(P6,xhat6)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','green')

a= P6(1,1)^.5;
b= P6(2,2)^.5;

[vec,val]=eigs(P6);
U=vec;
phi = atan2(U(2,1),U(1,1));
% dist = norm(xhat-xhat6);
% delx = dist*U(1,1);
% dely = dist*(U(1,2));
% delsigx = delx/a;
% delsigy=dely/b;
del=U'*[xhat2-xhat6];

xrel=del(1)*cos(phi);
yrel=del(2)*sin(phi);
delsigx=xrel/a;
delsigy=yrel/b;
delsigma = norm([delsigx delsigy]);

xhat2prime = U'*(xhat2);
th=atan2(xhat2prime(2),xhat2prime(1));
rel=th-phi;
d=norm(xhat2prime);
delx=d*cos(rel);
dely=d*sin(rel);
delxstd=delx/a;
delystd=dely/b;

delstd = sqrt(delxstd^2 + delystd^2);

cov_ellipse(P6*delstd^2,xhat6(1),xhat6(2),'y');
% error_ellipse(P6*delstd^2,xhat6)
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','yellow')
quiver([xhat6(1),xhat6(1)]',[xhat6(2) xhat6(2)]',vec(:,2),vec(:,1),6)
fprintf('\n\nxhat6 is %3.3f STD from xhat in X',delxstd);
fprintf('\nxhat6 is %3.3f STD from xhat in Y',delystd);
fprintf('\nFor a total sigma distance of %3.3f\n\n',delstd)

legend('$\hat{x}_6$','$\hat{x}_{comb}$','$1-\sigma$','1.336-$\sigma$','Principal A
```

```
xhat6 is -0.873 STD from xhat in X
xhat6 is -1.011 STD from xhat in Y
For a total sigma distance of 1.336
```



e - The plot for all covariance matrices

```
figure
hold on
cmap = hsv(9);
for ii = 1:8
    plot(x(ii),y(ii),'.','Color',cmap(ii,:))
%     error_ellipse(data(ii).P,[data(ii).xy]);
%     lines = sort(findobj(gca,'Type','line'));
%     set(lines(end),'color',cmap(ii,:))
    cov_ellipse(data(ii).P,x(ii),y(ii),cmap(ii,:));
end

plot(xhat(1),xhat(2),'xk','MarkerSize',15)
% error_ellipse(Pfinal,xhat);
cov_ellipse(Pfinal,xhat(1),xhat(2),cmap(9,:));
lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color',cmap(9,:))
set(lines(end),'linewidth',3)

title('Problem 4e - All Probability Ellipsoids')

legend('$\hat{x}_1$','$P_1$','$\hat{x}_2$','$P_2$','$\hat{x}_3$','$P_3$',...
        '$\hat{x}_4$','$P_4$','$\hat{x}_5$','$P_5$','$\hat{x}_6$','$P_6$','$\hat{x
        '$P_7$','$\hat{x}_8$','$P_8$','$\hat{x}_{comb}$','$P_{\hat{x}_{comb}}$','l
```

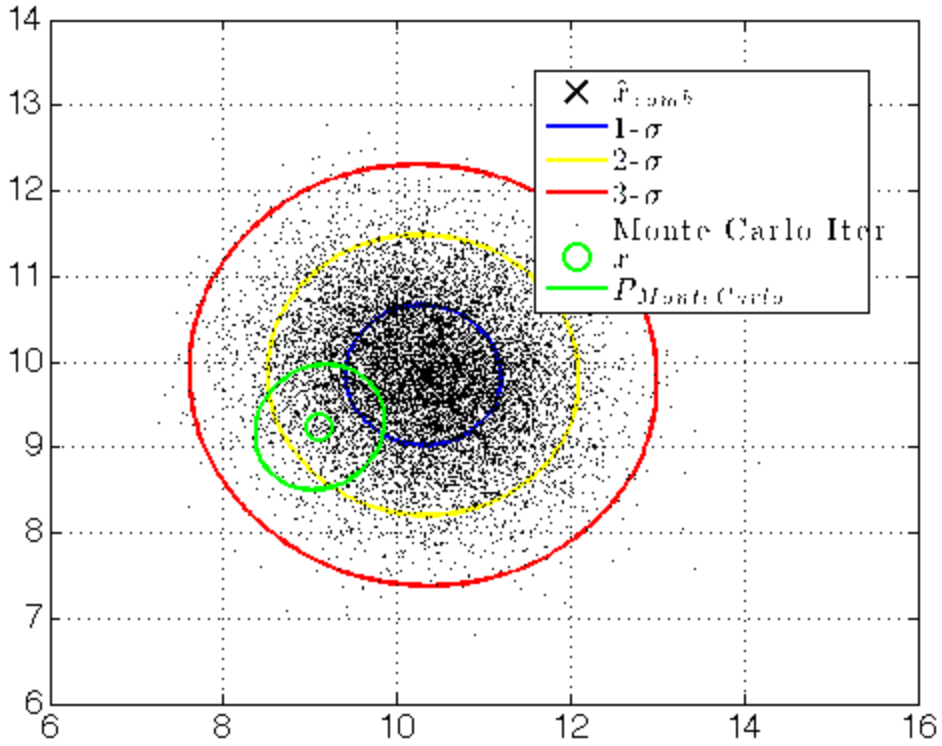


g - Monte Carlo

```
figure
hold on
plot(xhat(1),xhat(2),'xk','MarkerSize',15)
% error_ellipse(Pfinal,xhat);
%
% error_ellipse(Pfinal*4,xhat);
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','yellow')
%
% error_ellipse(Pfinal*9,xhat);
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','red')
cov_ellipse(Pfinal,xhat(1),xhat(2),'b');
cov_ellipse(Pfinal*2^2,xhat(1),xhat(2),'y');
cov_ellipse(Pfinal*3^2,xhat(1),xhat(2),'r');


S = (chol(Pfinal))';
x_montbar = [0;0];
P_mont = [0 0;0 0];
for ii = 1:10000
    e=randn(2,1);
    x_mont(:,ii)=(S'*e)+xhat;
```

```
    x_montbar = (x_montbar + x_mont(:,ii))/2;
    P_mont = (P_mont + (x_mont(:,ii) -x_montbar)*(x_mont(:,ii)-x_montbar)')/2;
end
plot(x_mont(1,:),x_mont(2,:),'k.','markersize',1)

plot(x_montbar(1),x_montbar(2),'og','Markersize',15)


cov_ellipse(P_mont,x_montbar(1),x_montbar(2),'G');
% error_ellipse(P_mont,x_montbar);
% lines = sort(findobj(gca,'Type','line'));
% set(lines(end),'color','green')

% x_montbar = sum(x_mont,2)/ii;

legend('$\hat{x}_{comb}$','1-$\sigma$','2-$\sigma$','3-$\sigma$','Monte Carlo Iter
```

```
% [evecs,evals] = eig([Pfinal,[0;0];[0 0 0]]);
% semi(1) = sqrt(evals(1,1));
% semi(2) = sqrt(evals(2,2));
% semi(3) = sqrt(evals(3,3));
% semi = sort(semi);
% semi = semi([3,2,1]);
% plotEllipsoid(evecs,semi)
```



Published with MATLAB® 8.0