For 1(c) we use following codes.

clc;

clear all;

L1 = [0,0];

L2 = [4,2];

theta1 = -165/180\*pi;

theta2 = 150/180\*pi;

pos = get\_pos(theta1,theta2,L1,L2);

hold on

plot([L1(1),L2(1)],[L1(2),L2(2)],'go')

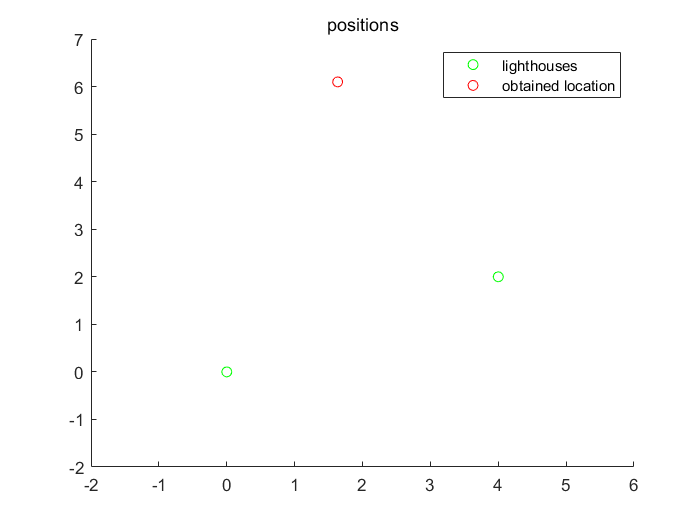
plot(pos(1),pos(2),'ro')

axis([-2,6,-2,7])

title('positions')

legend("lighthouses","obtained location")

And the picture look like below:



P1: position of light house and obtained location

For 1(d) we use following codes.

clc;

clear all;

L1 = [0,0];

L2 = [4,2];

L3 = [1,4];

theta1 = -140/180\*pi;

theta2 = 90/180\*pi;

theta3 = -30/180\*pi;

pos1 = get\_pos(theta1,theta2,L1,L2);

pos2 = get\_pos(theta2,theta3,L2,L3);

pos3 = get\_pos(theta1,theta3,L1,L3);

pos = [pos1, pos2,pos3];

hold on

plot([L1(1),L2(1),L3(1)],[L1(2),L2(2),L3(2)],'go')

plot(pos(1,:),pos(2,:),'bo')

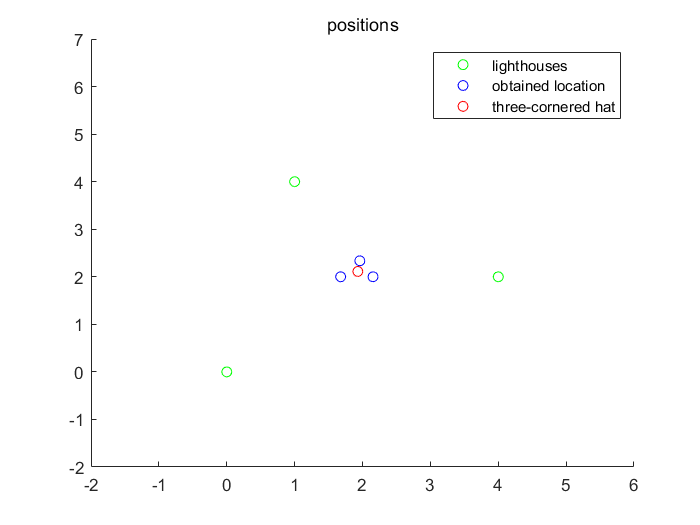
plot(sum(pos(1,:))/3,sum(pos(2,:))/3,'ro')

axis([-2,6,-2,7])

title('positions')

legend("lighthouses","obtained location","three-cornered hat");

And the picture look like below:



P2: position of light house and obtained location

For 2(a) we use following codes. It include a function “fun\_2a” . The function code is shown below:

function fun = fun\_2a(L1,L2,x,theta,phi)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...

\*norm(x-L2)\*cos(theta)-norm(x-L1)^2;

eq2 = dot(L2-x,[0;1])-cos(phi)\*norm(L2-x);

fun = eq1^2+eq2^2;

if(norm(x-L1)<0.1 || norm(x-L2)<0.1)

fun = fun+10001/(norm(x-L1)\*norm(x-L2)+0.01);

end

end

The other code is shown as follows:

clc;

clear all;

L1 = [0,0];

L2 = [4,2];

theta1 = -165/180\*pi;

theta2 = 150/180\*pi;

theta = 2\*pi+theta1-theta2;

phi = -theta1+theta;

%pos = get\_pos(theta1,theta2,L1,L2);

tt = fun\_2a(L1,L2,[4,2],theta,phi);

fun = @(x) fun\_2a(L1,L2,x,theta,phi);

fval\_list = [];

x\_all = [];

for i = -2:2:10

for j = -2:2:10

x0 = [i,j];

options = optimoptions('fminunc','OptimalityTolerance',10e-6);

[x\_ans,fval] = fminunc(fun,x0,options);

fval\_list = [fval\_list,fval];

x\_all = [x\_all;x\_ans];

end

end

fval\_list = fval\_list';

ind = find(fval\_list==min(min(fval\_list)));

x\_final = x\_all(ind,:);

hold on

plot([L1(1),L2(1)],[L1(2),L2(2)],'go')

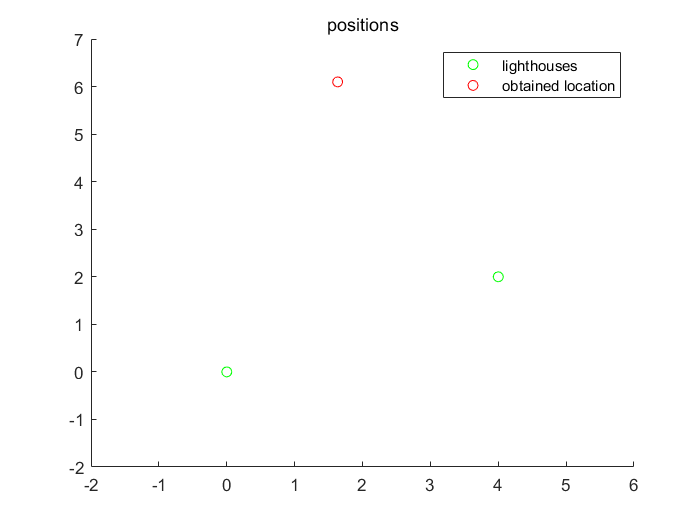
plot(x\_final(1),x\_final(2),'ro')

axis([-2,6,-2,7])

title('positions')

legend("lighthouses","obtained location")

And the picture look like below:



P3: position of light house and obtained location

For 2(b) we use following codes.

clc;

clear all;

L1 = [0,0];

L2 = [4,2];

L3 = [1,4];

theta1 = -140/180\*pi;

theta2 = 90/180\*pi;

theta3 = -30/180\*pi;

theta = 2\*pi+theta1-theta2;

theta\_2 = 2\*pi+theta3-theta2; % 3 and 2

theta\_3 = -theta1+theta3; %1 and 3

phi = -theta1+theta; %1 and 2

phi\_1 = theta2; %3 and 2

phi\_2 = theta3;%1 and 3

tt = fun\_2a(L1,L2,[4,2],theta,phi);

fun = @(x) fun\_2a(L1,L2,x,theta,phi);

fun2 = @(x) fun\_2a(L3,L2,x,theta\_2,phi\_1);

fun3 = @(x) fun\_2a(L1,L3,x,theta\_3,phi\_2);

fval\_list = [];

x\_all = [];

fval\_list2= [];

x\_all2 = [];

fval\_list3 = [];

x\_all3 = [];

for i = -2:2:10

for j = -2:2:10

x0 = [i,j];

options = optimoptions('fminunc','OptimalityTolerance',10e-6);

[x\_ans,fval] = fminunc(fun,x0,options);

fval\_list = [fval\_list,fval];

x\_all = [x\_all;x\_ans];

%3 and 2

[x\_ans2,fval2] = fminunc(fun2,x0,options);

fval\_list2 = [fval\_list2,fval2];

x\_all2 = [x\_all2;x\_ans2];

%1 and 3

[x\_ans3,fval3] = fminunc(fun3,x0,options);

fval\_list3 = [fval\_list3,fval3];

x\_all3 = [x\_all3;x\_ans3];

end

end

fval\_list = fval\_list';

ind = find(fval\_list==min(min(fval\_list)));

x\_final\_1 = x\_all(ind,:);

fval\_list2 = fval\_list2';

ind = find(fval\_list2==min(min(fval\_list2)));

x\_final\_2 = x\_all2(ind,:);

fval\_list3 = fval\_list3';

ind = find(fval\_list3==min(min(fval\_list3)));

x\_final\_3 = x\_all3(ind,:);

x\_lists\_all = [x\_final\_1;x\_final\_2;x\_final\_3];

hold on

plot([L1(1),L2(1),L3(1)],[L1(2),L2(2),L3(2)],'go')

plot(x\_lists\_all(:,1),x\_lists\_all(:,2),'bo')

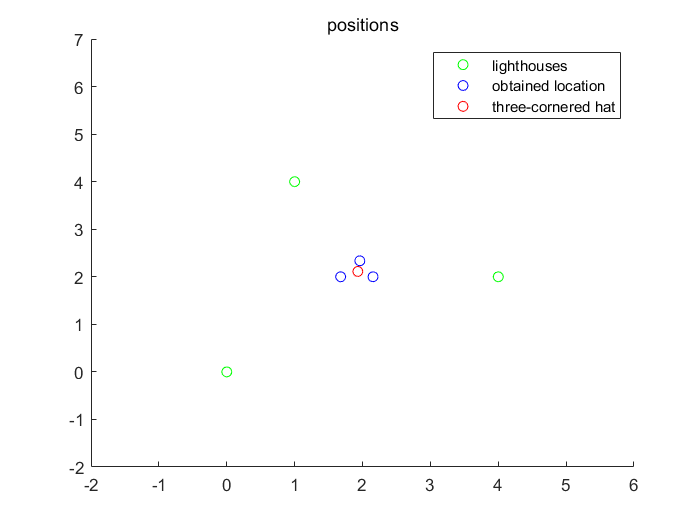
plot(sum(x\_lists\_all(:,1))/3,sum(x\_lists\_all(:,2))/3,'ro')

axis([-2,6,-2,7])

title('positions')

legend("lighthouses","obtained location","three-cornered hat")

And the picture look like below:



P4: position of light house and obtained location

For 3(a) we use following codes. And it includes a function “fun\_3”. The code of the function is shown below:

function fun = fun\_3(L1,L2,x,theta,phi)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...

\*norm(x-L2)\*cos(theta)-norm(x-L1)^2;

eq2 = dot(L2-x,[0,1,0])-cos(phi)\*norm(L2-x);

fun = eq1^2+eq2^2;

%if(norm(x-L1)<0.1 || norm(x-L2)<0.1)

%fun = fun+10001/(norm(x-L1)\*norm(x-L2)+0.01);

%end

end

The other code is shown as follows:

clc;

clear all;

L1 = [0,0,0];

L2 = [5,0,0];

theta = 0.5\*pi;

phi = 135/180\*pi;

%pos = get\_pos(theta1,theta2,L1,L2);

fun = @(x) fun\_3(L1,L2,x,theta,phi);

fval\_list = [];

x\_all = [];

for i = -5:2:5

for j = -5:2:5

for q = -5:2:5

x0 = [i,j,q];

options = optimoptions('fminunc','OptimalityTolerance',10e-16);

[x\_ans,fval] = fminunc(fun,x0,options);

fval\_list = [fval\_list,fval];

x\_all = [x\_all;x\_ans];

end

end

end

fval\_list = fval\_list';

ind = find(fval\_list==min(min(fval\_list)));

x\_final = x\_all(ind,:);

%3D plot

center = [2.5,2.5,0];

x = x\_all(:,1);

y = x\_all(:,2);

z = x\_all(:,3);

figure(1)

scatter3(x,y,z,50,'b','fill')

hold on

scatter3(center(1), center(2), center(3),50,'r','fill')

hold off

zlabel('$z$ (m)','interpreter','latex','fontsize',15)

ylabel('$y$ (m)','interpreter','latex','fontsize',15)

xlabel('$x$ (m)','interpreter','latex','fontsize',15)

%2D projections

figure(2)

subplot(2,2,1)

scatter(x,z,50,'b','fill')

hold on

scatter(center(1), center(3),50,'r','fill')

hold off

grid on

ylabel('$z$ (m)','interpreter','latex','fontsize',15)

%xlabel('$x$ (m)','interpreter','latex','fontsize',15)

subplot(2,2,3)

scatter(x,y,50,'b','fill')

hold on

scatter(center(1), center(2),50,'r','fill')

hold off

grid on

ylabel('$y$ (m)','interpreter','latex','fontsize',15)

xlabel('$x$ (m)','interpreter','latex','fontsize',15)

subplot(2,2,4)

scatter(z,y,50,'b','fill')

hold on

scatter(center(3), center(2),50,'r','fill')

hold off

%ylabel('$y$ (m)','interpreter','latex','fontsize',15)

xlabel('$z$ (m)','interpreter','latex','fontsize',15)

grid on

figure(3)

[Xsp,Ysp,Zsp] = sphere;

surf(2.5.\*Xsp+ 2.5,2.5.\*Ysp,2.5.\*Zsp,'FaceAlpha',0.25);

hold on

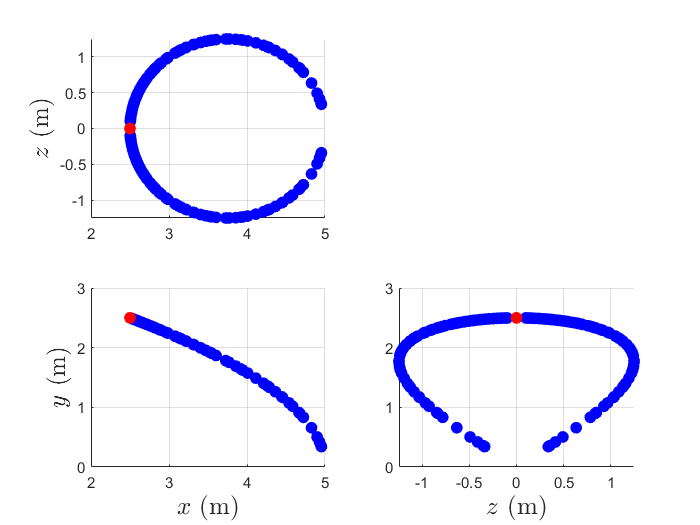
scatter3(x,y,z,50,'b','fill')

grid on

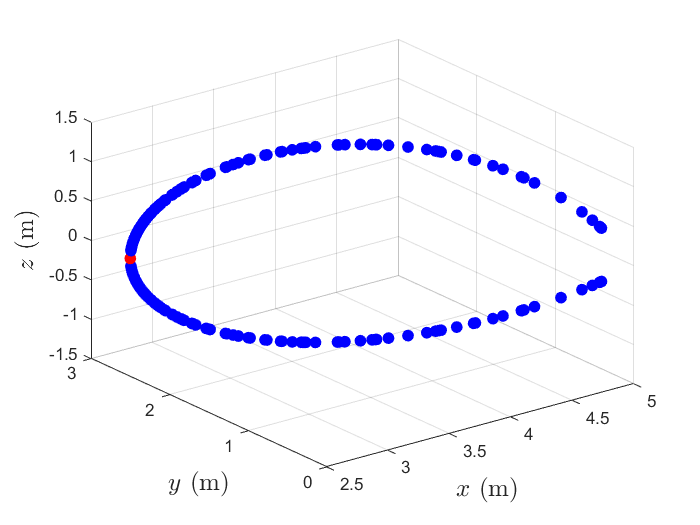
hold off

axis equal

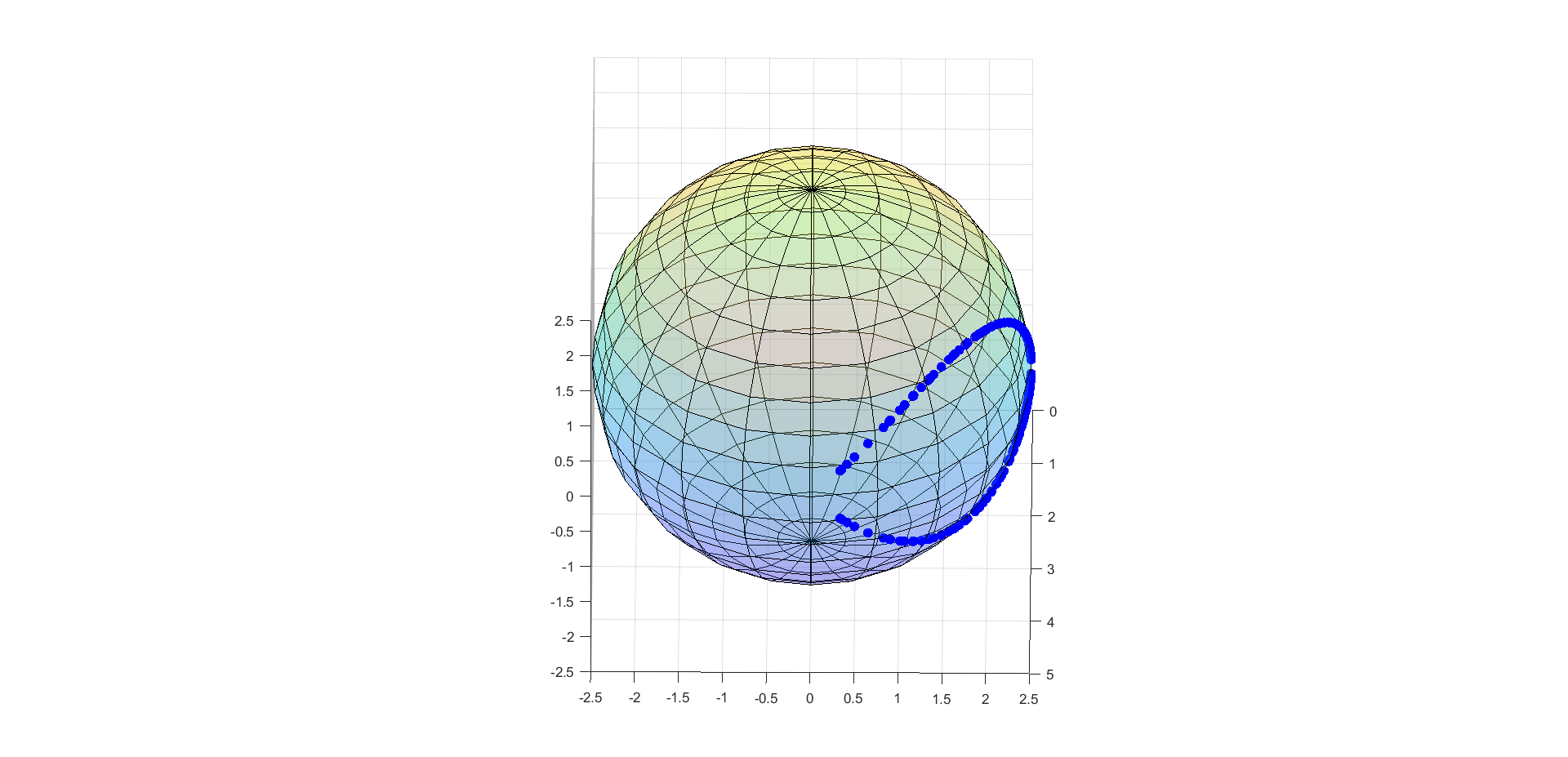
And the pictures look like below:



P5: 2D projection of solutions and the point



P6: solutions in actual 3D space



P7: sphere with solutions

We can observe that actually solutions are on the surface of a sphere. We get the solutions based on a subtended angle and a bearing angle. In 3D space, the same subtended angle within 2 lighthouses represents a sphere and the same bearing angle within a lighthouse represents a conical. Thus, these two constrictions form the solutions we have: a curve on the surface of a 3D sphere.

For 4 we use following codes. The codes includes a function call “fun\_4”. The function shows as follows:

function fun = fun\_4(L1,L2,x,theta,phi1,star1,phi2,star2)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...

\*norm(x-L2)\*cos(theta)-norm(x-L1)^2;

eq2 = dot(L2-x,star1)-cos(phi1)\*norm(L2-x);

eq3 = dot(L2-x,star2)-cos(phi2)\*norm(L2-x);

fun = eq1^2+eq2^2+eq3^2;

if(norm(x-L1)<0.1 || norm(x-L2)<0.1)

fun = fun+10001/(norm(x-L1)\*norm(x-L2)+0.01);

end

end

The other codes shows as follows:

clc;

clear all;

L1 = [0,0,0];

L2 = [5,0,0];

theta = 0.5\*pi;

phi1 = 135/180\*pi;

phi2 = 90/180\*pi;

star1 = [0,1,0];

star2 = [0,0,1];

fun = @(x) fun\_4(L1,L2,x,theta,phi1,star1,phi2,star2);

fval\_list = [];

x\_all = [];

for i = -5:2:5

for j = -5:2:5

for q = -5:2:5

x0 = [i,j,q];

options = optimoptions('fminunc','OptimalityTolerance',10e-16);

[x\_ans,fval] = fminunc(fun,x0,options);

fval\_list = [fval\_list,fval];

x\_all = [x\_all;x\_ans];

end

end

end

fval\_list = fval\_list';

ind = find(fval\_list==min(min(fval\_list)));

x\_final = x\_all(ind,:);

center = x\_final;

house = [L1;L2];

x = house(:,1);

y = house(:,2);

z = house(:,3);

figure(1)

subplot(2,2,1)

scatter(x,z,50,'g','fill')

hold on

scatter(center(1), center(3),50,'r','fill')

hold off

grid on

ylabel('$z$ (m)','interpreter','latex','fontsize',15)

%xlabel('$x$ (m)','interpreter','latex','fontsize',15)

subplot(2,2,3)

scatter(x,y,50,'g','fill')

hold on

scatter(center(1), center(2),50,'r','fill')

hold off

grid on

ylabel('$y$ (m)','interpreter','latex','fontsize',15)

xlabel('$x$ (m)','interpreter','latex','fontsize',15)

subplot(2,2,4)

scatter(z,y,50,'g','fill')

hold on

scatter(center(3), center(2),50,'r','fill')

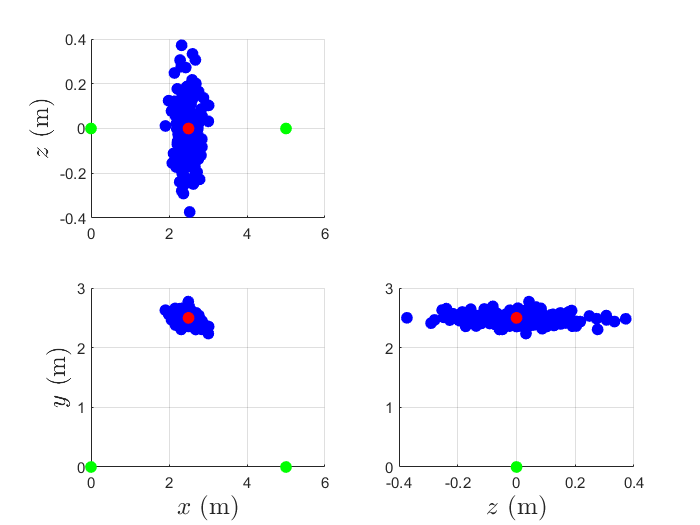
hold off

%ylabel('$y$ (m)','interpreter','latex','fontsize',15)

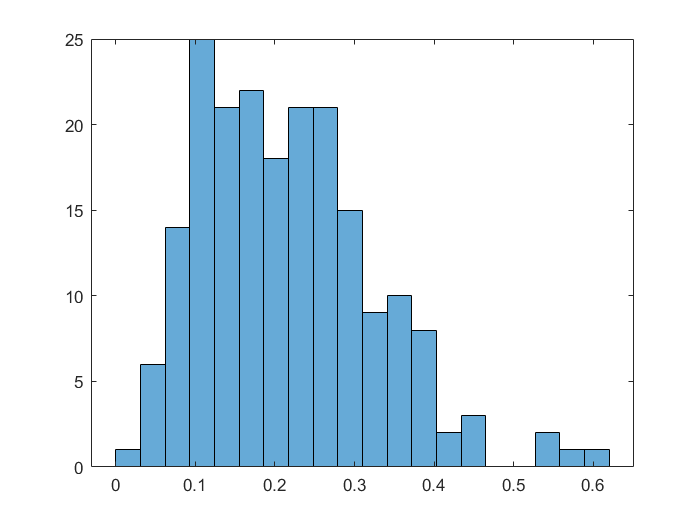
xlabel('$z$ (m)','interpreter','latex','fontsize',15)

grid on

And the picture looks like below:



P8: projection of obtained locations



P9: histogram of position fix errors

We get:

deviation of error

0.1089

mean of error

0.2152