```
function coverSIM()

clc
close all

%%% seleccion del tipo de modelo de propagacion
selection=1;

%%% Sensibilidad (por debajo no puede haber comunicaci?n)
%%% en el sistema
objetivo= -120;


%%% posici?n de la estaciones base
L=[ 0 0;1500 0;750 1300;750 -1300;-750 -1300;-1500 0;-750 1300];

%%% distancia de referencia
d0=100;
%d0=2;
%obs=[300 400 -300 200];

%%% altura antenna
h=[180,160, 150, 190,130,150,180];


%%% exponente de perdida
n=[2.5,2.6,2.7,2.2,2.5,2.6,2.5];


%%%% std de las Gaussianas (modelo log-normal)
sigma=[5,6,4,4,5,8,5];

%%% Potencia transmitida
Pt=-65*[1 1 1 1 1 1 1];


%%%% posici?n puntos aleatorio en un cuadrado de -1500 por 1500
a=random('unif', -1500,1500,10000,2);

%%% distancia puntos aleatorios - base
d=abs([sqrt((a(:,1)-L(1,1)).^2+(a(:,2)-L(1,2)).^2), sqrt((a(:,1)-L(2,1)).^2+(a(:,2)-L(2,2)).^2),sqrt((a(:,1)-L(3,1)).^2+(a
(:,2)-L(3,2)).^2),sqrt((a(:,1)-L(4,1)).^2+(a(:,2)-L(4,2)).^2),sqrt((a(:,1)-L(5,1)).^2+(a(:,2)-L(5,2)).^2),sqrt((a(:,1)-L
(6,1)).^2+(a(:,2)-L(6,2)).^2),sqrt((a(:,1)-L(7,1)).^2+(a(:,2)-L(7,2)).^2)]);
switch selection
    %%% potencia que llega a cada estacion base
    case 1
        %modelo log-distancia
        % llama la funci?n auxiliar 'Pr'(mira abajo)
        power=[Pr(Pt(1),n(1),d(:,1),d0) Pr(Pt(2),n(2),d(:,2),d0) Pr(Pt(3),n(3),d(:,3),d0),Pr(Pt(4),n(4),d(:,4),d0),Pr(Pt
(5),n(5),d(:,5),d0),Pr(Pt(6),n(6),d(:,6),d0),Pr(Pt(7),n(7),d(:,7),d0)];
    case 2

        %Okumura Hata
        % llama la funci?n auxiliar 'Pok' (mira abajo)
        power=[Pok(Pt(1),h(1),d(:,1)/1000), Pok(Pt(2),h(2),d(:,2)/1000), Pok(Pt(3),h(3),d(:,3)/1000), Pok(Pt(4),h(4),d(:,
4)/1000),Pok(Pt(5),h(5),d(:,5)/1000),Pok(Pt(6),h(6),d(:,6)/1000),Pok(Pt(7),h(7),d(:,7)/1000)];
end
% power=power +Pray(a,L,obs);
% power

z=1.67 %%% TABLA Gaussiana, TRASPARENCIA 16 (o mejor 20), TEMA 3 de la asignatura

%%%% la variable 'power' representa la media de la Gaussiana
ZZ=power-z*(ones(size(power,1),1)*sigma);


% decision1=(erfc(z/sqrt(2))/2)
% decision=zeros(1,size(power, 2));


for i=1:size(power,1)
    if(sum(ZZ(i,:)<objetivo)==7)
        %%%% si ningun ZZ es < que la Sensibilidad (objetivo)
        decision(i)=0;
    else
        decision(i)=find(max(ZZ(i,(ZZ(i,:)>objetivo)))==ZZ(i,:));
    end
end


%%%%% dibuja la figura (puntos con distintos color seg?n la decisi?n)
scatter(a(:,1),a(:,2),30,4*decision.')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% funciones auxiliares
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    function x=Pr(Pt,n,d,d0)
        %modelo log-distancia
        x=Pt-20-10.*n.*log10(d./d0);
    end
    function x=Pok(Pt,H,d)
        %Okumura Hata
        f=1800;
        h2=1.5;
        x=Pt+40-(69.55+26.16.*log10(f)-13.82*log10(H)+(44.9-6.545*log10(H))*log10(d)-((1.1*log10(f)-.7)*h2-(1.56*log10
(f)-0.8)));%
    end
    function x=P2ray(a,L,h,h2)
        x=zeros(size(a,1),7);

        for i=1:7
            for k=1:size(a,1)
                u1=((obs(3)-obs(1))*(L(i,2)-obs(2))-(obs(4)-obs(2))*(L(i,1)-obs(1)))/((obs(4)-obs(2))*(a(k,1)-L(i,1))-(obs
(3)-obs(1))*(a(k,2)-L(i,2)) );
                u2=((a(k,1)-L(i,1))*(L(i,2)-obs(2))-(a(k,2)-obs(2))*(L(i,1)-obs(1)))/((obs(4)-obs(2))*(a(k,1)-L(i,1))-(obs
(3)-obs(1))*(a(k,2)-L(i,2)) );

                if (u1>=0 && u1<=1 && u2>=0 && u2<=1)
                    x1=L(i,1)+u1*(a(k,1)-L(i,1));
                    y1=L(i,2)+u2*(a(k,2)-L(i,2));



                    d1=sqrt((L(i,1)-x1).^2+(L(i,2)-y1).^2);
                    d2=sqrt((a(k,1)-x1).^2+(a(k,2)-y1).^2);

                    x(k,i)=20*log10(4*pi/(d1+d2)*h*h2*1800/3e2);
                else x(k,i)=0;
                end

            end
        end

    end
    function x=Pray(a,L,obs)
        h=-1.5;
        f=1800;
        x=zeros(size(a,1),7);
        for i=1:7
            for k=1:size(a,1)
                %     x1=(L(i,1).*a(:,2)-L(i,2).*a(:,1)*(obs(1)-obs(3))-(L(i,1)-a(:,1))*(obs(1)*obs(4)-obs(2)*obs(3)))/((L
(i,1)-a(:,1))*(obs(2)-obs(4))-(L(i,2)-a(:,2)).*(obs(1)-obs(3)));
                %     y1=(L(i,1).*a(:,2)-L(i,2).*a(:,1)*(obs(2)-obs(4))-(L(i,2)-a(:,2))*(obs(1)*obs(4)-obs(2)*obs(3)))/((L
(i,1)-a(:,1))*(obs(2)-obs(4))-(L(i,2)-a(:,2)).*(obs(1)-obs(3)));
                u1=((obs(3)-obs(1))*(L(i,2)-obs(2))-(obs(4)-obs(2))*(L(i,1)-obs(1)))/((obs(4)-obs(2))*(a(k,1)-L(i,1))-(obs
(3)-obs(1))*(a(k,2)-L(i,2)) );
                u2=((a(k,1)-L(i,1))*(L(i,2)-obs(2))-(a(k,2)-obs(2))*(L(i,1)-obs(1)))/((obs(4)-obs(2))*(a(k,1)-L(i,1))-(obs
(3)-obs(1))*(a(k,2)-L(i,2)) );

                if (u1>=0 && u1<=1 && u2>=0 && u2<=1)
                    x1=L(i,1)+u1*(a(k,1)-L(i,1));
                    y1=L(i,2)+u2*(a(k,2)-L(i,2));



                    d1=sqrt((L(i,1)-x1).^2+(L(i,2)-y1).^2);
                    d2=sqrt((a(k,1)-x1).^2+(a(k,2)-y1).^2);
                    %     v=10*sqrt(2*f/3e8*(d1+d2)/(d1*d2));
                    m=(obs(4)-obs(2))/(obs(3)-obs(1));

                    %     F=(FresnelC(v)+sqrt(-1)*FresnelS(v))/2;
                    v=h*sqrt(2*f/3e2*(d1+d2)/(d1*d2));
% %
% %                       F=quad('exp((-j*pi*x.^2)/2)',v,35); %Integration of the function used in integral part of Complex
Fresnel Integral
% %                       fe=abs((0.5+0.5*sqrt(-1))*F ); %Complex Fresnel Integral.
% %                       x(k,i)=20*log10(fe); %attenuation factor introduced by Knife-edge diffraction
                    if(v < -1.0)
                        x(k,i)=0;
                    elseif( v <= 0)
                        x(k,i)=20*log10(0.5-0.62*v);
                    elseif(v <= 1)
                        x(k,i)=20*log10(0.5*exp(-0.95*v));
```

```
                elseif(v <= 2.4)
                    x(k,i)=20*log10(0.4-sqrt(0.1184-(0.38-0.1*v).^2));
                else
                    x(k,i)=20*log10(0.225/v);
                end
            else x(k,i)=0;
            end
        end
    end

end
end
```