Date:2021-12-09

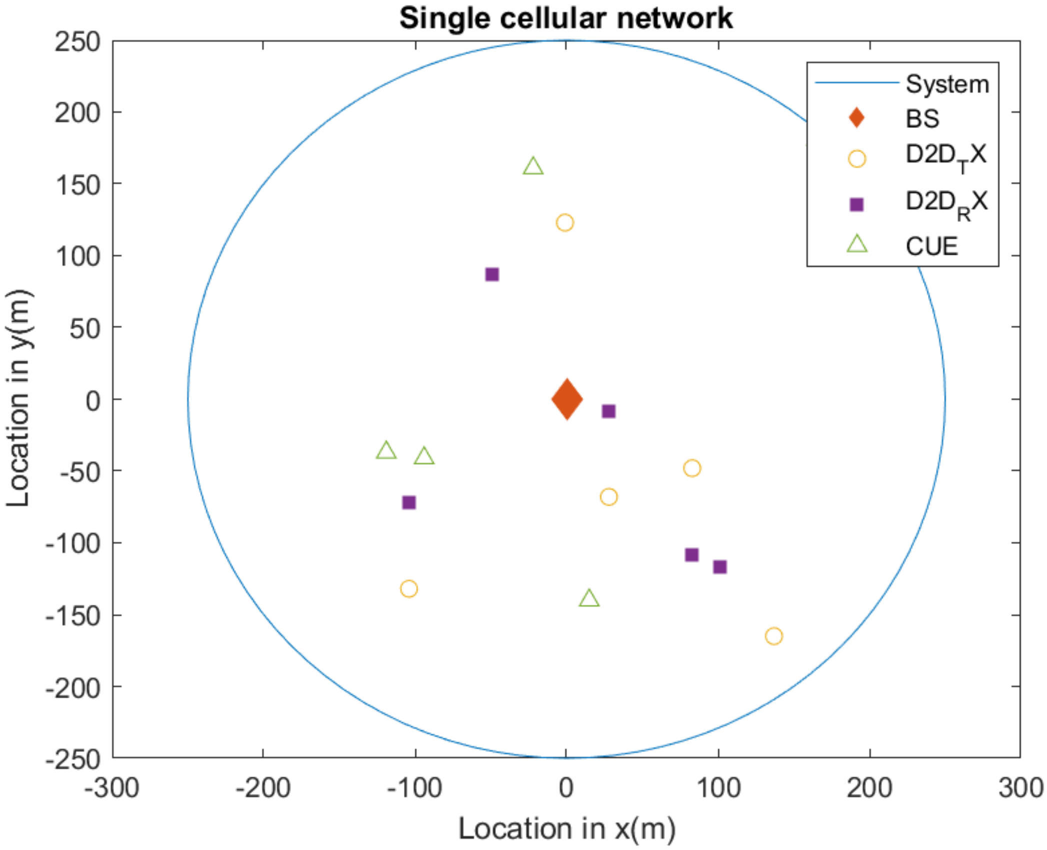
Cellular Network plotting

Implement a function which can be used to draw the cellular Network. It takes the coordinates of the two points (D2D TX and D2D RX) on D2D link and CUE. Then draw the random points to modelize the system

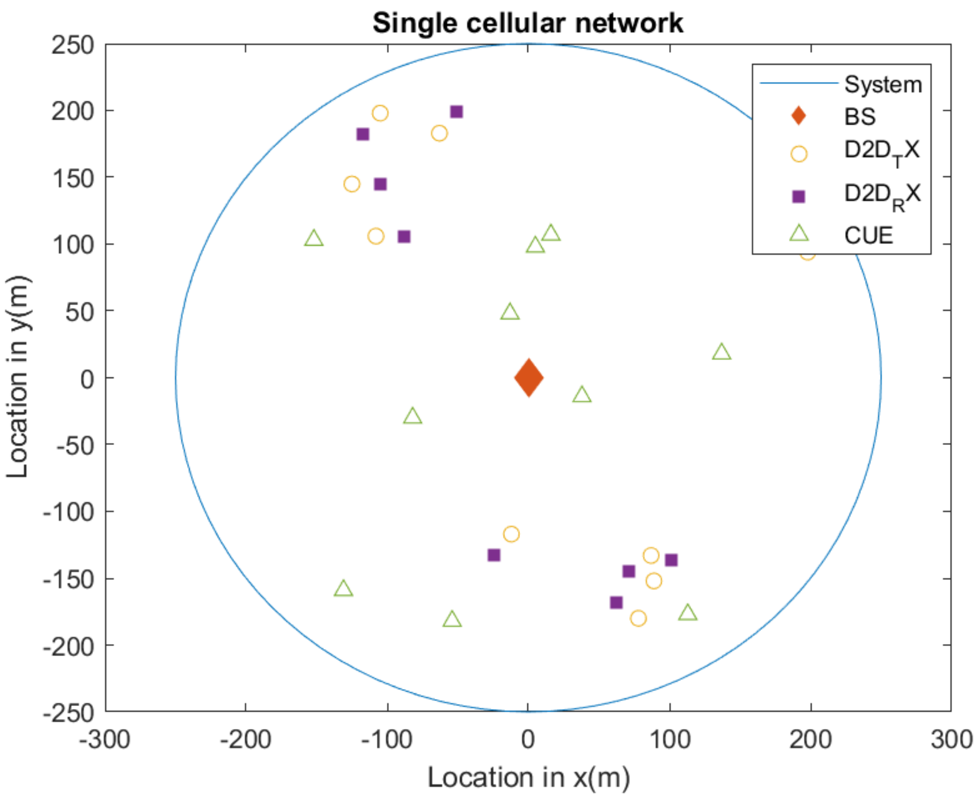
```
function plot_model(D2D,CUE)
D2D_TX_x=[];
D2D_TX_y=[];
D2D_RX_x=[];
D2D_RX_y=[];
CUE_x=[];
CUE_y=[];
for i=1:size(D2D,1)
    D2D_TX_point=D2D{i,1};
    D2D_RX_point=D2D{i,2};
    D2D_TX_x(end+1)=D2D_TX_point(1,1);
    D2D_TX_y(end+1)=D2D_TX_point(1,2);
    D2D_RX_x(end+1)=D2D_RX_point(1,1);
    D2D_RX_y(end+1)=D2D_RX_point(1,2);
end

for k=1:size(CUE,1)
    CUE_x(end+1)=CUE(k,1);
    CUE_y(end+1)=CUE(k,2);
end
th = 0:pi/50:2*pi;
x=0;
y=0;
r=250;
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
BS=0;
plot(xunit,yunit)
hold on
scatter([0],BS,140,'filled','d');
hold on
scatter(D2D_TX_x,D2D_TX_y);
hold on
scatter(D2D_RX_x,D2D_RX_y,'filled','s');
hold on
scatter(CUE_x,CUE_y,'^');
hold off
legend('System','BS','D2D_TX','D2D_RX','CUE');
title('Single cellular network')
xlabel('Location in x(m)')
ylabel('Location in y(m)')
end
```

This the Cellular Network when the number of D2D links is 5 and the number of CUE is 5, and the length of each D2D link is 60m



And this the Cellular Network when the number of D2D link is 10 and the number of CUE is 10 when the length of D2D link is 20m.



Rayleigh channel gain calculation

The channel gain of rayleigh fading channel can be calculated in two ways:

(1):*Modelling and simulation of Rayleigh fading, path loss, and shadowing fading for wireless mobile networks*

$$h_n^k = 20log_{10}(\frac{\lambda}{4\pi d_0}) * (\frac{d_0}{d})^\gamma$$

where

- λ is the wavelength
- d_0 is a reference distance for the antenna far filed and usually set to 10m
- d is the distance between the sender and receiver
- γ is the pass loss exponent that depends on the propagation environment environment. For urban environment it ranges 3.7-6.5, which in my case I would set to 4 or 3, and in the reference paper it should be 3

In this case, the λ will be different for each channel, and is calculated as follow $\lambda = \frac{c}{f}$ where c is the speed of light and f is the channel frequency.

(2) Literature Review: *Success Probability Analysis of C-V2X Communications on Irregular Manhattan Grids*

The vehicles in the network select to connect to either an MBS or an RSU according to Maximum Power-based Scheme. In this scheme, the vehicle connects to the transmitter (an MBS or an RSU) from which it receives the highest average power.

The vehicle chooses the closest RSU and MBS first and then selects to receive data from the one that can supply a higher time-average power.

To find the received power, we should introduce the channel model next. We use a general power-law path-loss model with exponent $\alpha > 2$ to describe the long time-scale channel effects. To describe the short time-scale channel effects, we use the Rayleigh fading in which the channel gain is exponentially distributed with unit mean. Thus, the instantaneous power received can be expressed by $P_{rec} = Phr^{-\alpha}$, where P is the transmit power, $h \sim \exp(1)$ models Rayleigh fading, and r is the distance between transmitter and receiver. We use time-average value of received power as the metric to choose transmitter, because it is undesirable to switch from one transmitter to another frequently due to the rapidly varying instantaneous power. If the average time interval is much larger than the coherence time of channel, the average received power is considered to be independent of h, which can be given by $P_{av} = Pr^{-\alpha}$. We assume all MBSs have the same transmit power P_c, and all RSUs have the same transmit power P_u.

The channel gain between TX and RX for a channel k is $h_n^k = (\frac{e^1}{r^\gamma})$

where

- e is a number drawn from exponential distribution where its mean is 1
- γ is the pass loss exponent that depends on the propagation environment

In my reference paper the second solution will be used to calculate the channel gain.

Prematching simulation

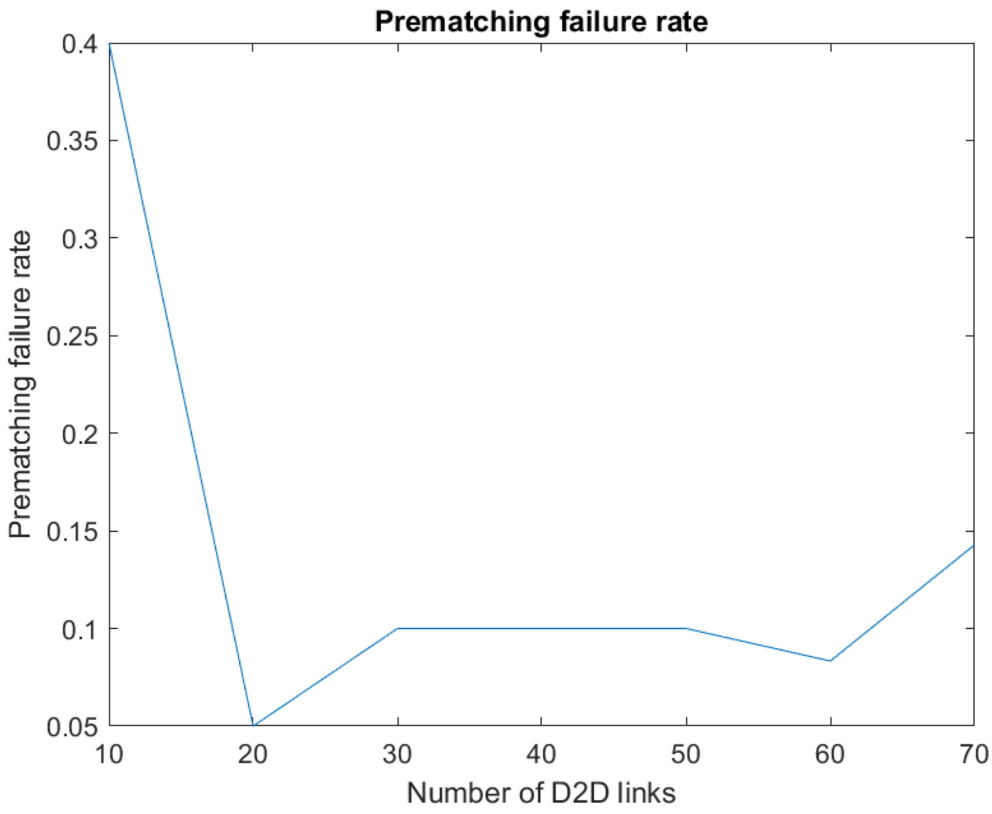
In this experiment, the distance of each D2D link i was set to 15m, while the distance between CUE k to the current distance was obtained by using another function ***point_to_line*** to calculate the distance

Point_to_line function

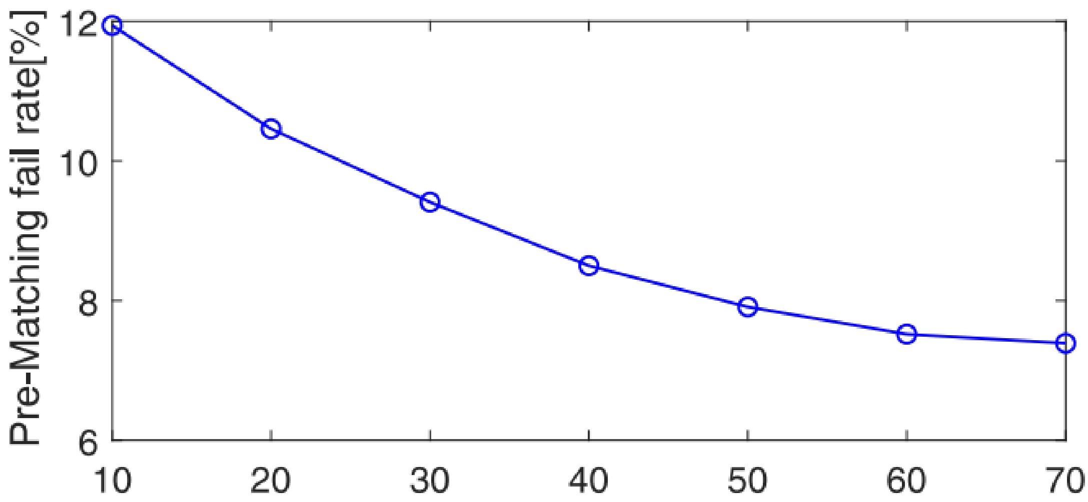
```
function d = point_to_line(v1,v2,CUE)
      v1(1,3)=0;
      v2(1,3)=0;
      CUE(1,3)=0;
      a = v1 - v2;
      b = CUE- v2;
      d = norm(cross(a,b)) / norm(a);
end
```

script for prematching algorithm

```
Pkc=0.1995262315;
Pth1=10*10^(-6);
Pmax=0.1995262315;
Tmin=2;
number=10:10:70;
fail_rate=[];
for i=10:10:70
[D2D,CUE]=system(i,i,15);
[Sid,InfD,EhaD]=Prematch(D2D,CUE,Pkc,Pth1,Pmax,Tmin,15);
fail=size(InfD,2)/i;
fail_rate(end+1)=fail;
end
plot(number,fail_rate)
title('Prematching failure rate')
xlabel('Number of D2D links')
ylabel('Prematching failure rate')
```



The result look a bit weird, since there is a huge fluctuation when increasing the number of D2D links, and what i feel confused about is the result from the reference paper, the when the total number of is 10, the PFMR is 12% which means that the link which can not perform SWIPT is 1.2, it is floating number.



The reason for that might due they set the distance from CUE k to D2D link i to a certain value, this is something that i need to ask my supervisor.