NETW-705

Mobile Communication Networks

Localization Project - Matlab

By:

- Farouk El Gohary 49- 0744
- Yazeed Ahmad Kamal 49- 11916
- Yusuf El Kilany 49- 3712
- Hussein Salama 49- 4179

Project Review

This code is a script that is used to simulate a floor plan with walls and access points (APs). It first initializes the positions of the walls and APs in the floor plan. It then creates a figure and uses a loop to draw the walls and APs on the figure. The walls are drawn in white and the APs are drawn in blue.

After creating the floor plan figure, the code initializes an array called "ref_points" which will store a set of reference points in the floor plan. It then uses a loop to populate the "ref_points" array with points at every 0.5 interval from (0,0) to (52,20).

The code then aggregates the positions of all the APs into a single array called "APs_array" and initializes an array called "walls" which stores the positions of the walls in the floor plan.

The code then defines several constants: "n", "c", "f", "lamda", and "PAF". These constants are used in the calculation of the received signal strength (RSS) at each reference point in the floor plan.

The code then enters a loop where it iterates through each reference point and calculates the RSS at that point using the equation:

RSS = PAF - (10 * n * log10(d/lambda))

where "d" is the distance between the reference point and the AP.

The code then finds the AP with the highest RSS at each reference point and marks that point with a circle on the floor plan figure.

After the loop that calculates the received signal strength (RSS) at each reference point, the code enters another loop which iterates through each reference point. For each reference point, the code calculates the distance between the reference point and each of the APs using the Euclidean distance formula:

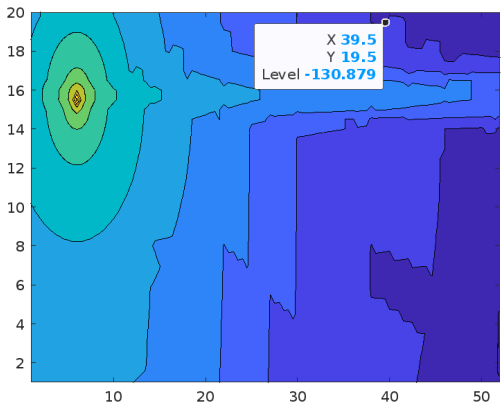
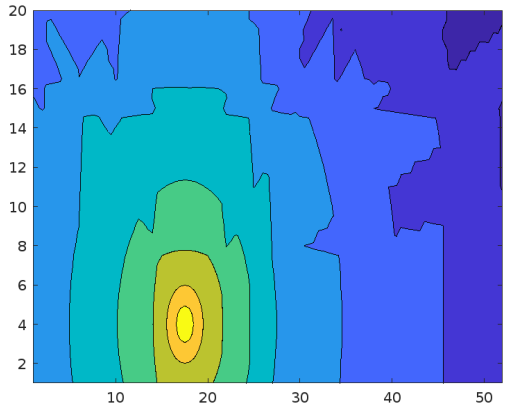
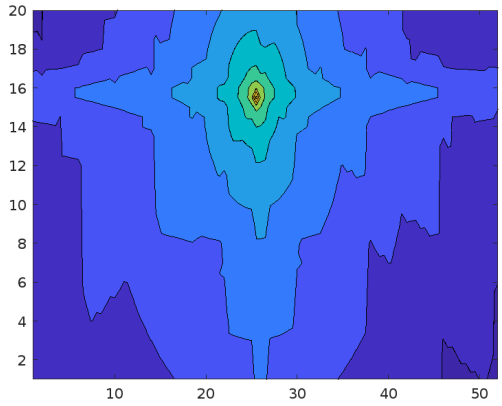
distance = sqrt((x2-x1)^2 + (y2-y1)^2)

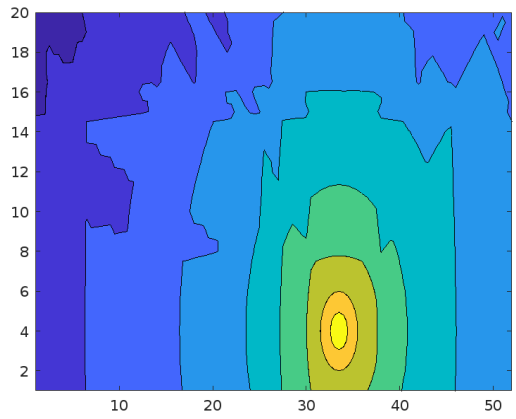
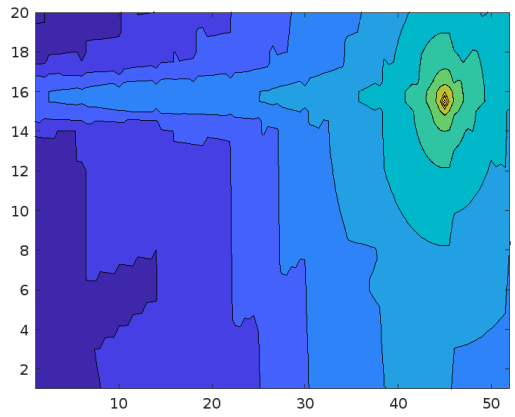
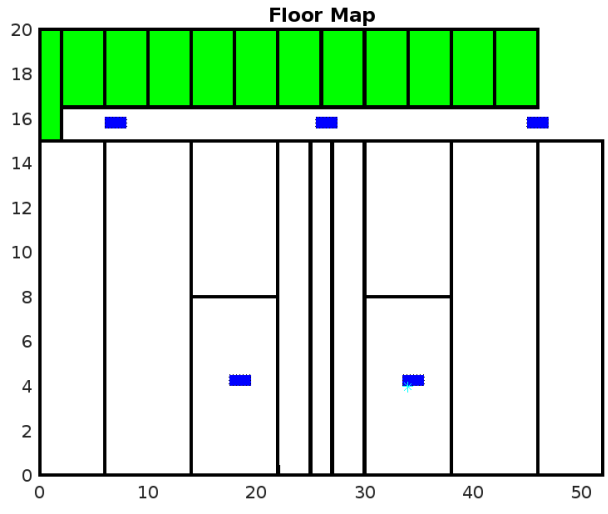
where (x1, y1) are the coordinates of the reference point and (x2, y2) are the coordinates of the AP.

The code then finds the AP with the minimum distance to the reference point and marks that point with a cross on the floor plan figure.

Finally, the code displays the floor plan figure with the walls, APs, and the user’s location.

Project Graphs Outputs

Contour Map of Access Point 1	
Contour Map of Access Point 2	
Contour Map of Access Point 3	

Contour Map of Access Point 4	
Contour Map of Access Point 5	
Floor Map	

Project Code

```
clear all

clc

clf

%Walls initialization for drawing

Walls_rect=[0,0,6,15;

   6,0,8,15;

   14,0,8,8;

   14,8,8,7;

   22,0,3,15;

   25,0,2,15;

   27,0,3,15;

   30,0,8,8;

   30,8,8,7;

   38,0,8,15;

   46,0,6,15;

   0,15,2,5;

   2,16.5,4,3.5;

   6,16.5,4,3.5;

   10,16.5,4,3.5;

   14,16.5,4,3.5;

   18,16.5,4,3.5;

   22,16.5,4,3.5;

   26,16.5,4,3.5;

   30,16.5,4,3.5;

   34,16.5,4,3.5;

   38,16.5,4,3.5;

   42,16.5,4,3.5];
```

```
AP1 = [6,15.5625];

AP2 = [17.5,4];

AP3 = [25.5,15.5625];

AP4 = [33.5,4];

AP5 = [45,15.5625];

% Defining the Access Points for drawing last 2 coordinates are length and

% width.

APs = [6,15.5625,2,0.5;

      17.5,4,2,0.5;

      25.5,15.5625,2,0.5;

      33.5,4,2,0.5;

      45,15.5625,2,0.5];

%------------------------------ Start of Localization----------------

% initializing the reference points.

ref_points=[];

for i=1:0.5:52

   for j=1:0.5:20

       ref_points=[ref_points;i j];

   end

end

%aggregating all access points in one array for iteration.

APs_array = [AP1;AP2;AP3;AP4;AP5];

walls = [0 52 15 15;

        6 6 0 15;

        14 14 0 15;

        22 22 0 15;

        25 25 0 15;

        27 27 0 15;

        30 30 0 15;
```

```
        38 38 0 15;
        46 46 0 15;
        52 52 0 15;
        14 22 8 8;
        30 38 8 8;
        2 46 16.5 16.5;
        6 6 16.5 20;
        10 10 16.5 20;
        14 14 16.5 20;
        18 18 16.5 20;
        22 22 16.5 20;
        26 26 16.5 20;
        30 30 16.5 20;
        34 34 16.5 20;
        38 38 16.5 20;
        42 42 16.5 20;
        46 46 16.5 20;
        2 2 15 20];
%Givens
n = 3;
c = 3*(10^8);
f = 2.4*(10^9);
lamda = c/f;
PAF = 3;
%the row indices represent the reference point no.
%the columns represent the power received from each access point.
ref_powers = [];
for i = 1:1:length(ref_points)
   x = ref_points(i,1);
```

```
   y = ref_points(i,2);


   power_access_point =[];
   for j = 1:1:length(APs_array)
       x_ap = APs_array(j,1);
       y_ap = APs_array(j,2);


       d =sqrt((x_ap-x)^2+(y_ap-y)^2);
       path_loss = ((2*pi*d)/lamda) ^n;
       p_loss_dB=10*log10(path_loss);



       wall_sum = 0;
       for wall = 1:1:length(walls)
              wall_ax = walls(wall,:);
              x_wall = wall_ax(1, [1 2]);
              y_wall = wall_ax(1, [3 4]);


              [x_intersect, y_intersect] = curveintersect(x_wall,y_wall,[x
x_ap],[y y_ap]);
               % summing up the number of walls
              points = length(x_intersect);
              wall_sum = wall_sum + points;
       end
        % calculating the power received at the
        % reference points from the access points
       power = -10 - p_loss_dB - (PAF*wall_sum);
       power_access_point = [power_access_point, power];
   end
   ref_powers = [ref_powers; power_access_point];
```

```
end
%---------- Contour map drawing -------------------------
for apoint = 1:1:length(APs_array)
   accessPointPowers = [ref_points(:,1), ref_points(:,2), ref_powers(:,apoint)];

   [x_values, x_values_indices] = unique(accessPointPowers(:,1));
   % calculating the difference in power
   % between each point and the one after it
   % to get how the power travels across the room
   indexDifference = diff(x_values_indices);
   % Reshape the array to separate the x coordinates,
   % y coordinates and power for the contour fx

   apReshaped = reshape(accessPointPowers,
indexDifference(1),[],size(accessPointPowers,2));
   % x_coo are the x coordinates
   x_coo = apReshaped(:,:,1);
   % y_coo are the y coordinates
   y_coo = apReshaped(:,:,2);
   % p_val are the power values at these coordinates
   p_val = apReshaped(:,:,3);
   % time to plot the power heat map for each Access Point
   figure('name', sprintf('Access Point Power Heat Map %d', apoint))
   contourf(x_coo,y_coo,p_val)
end
% ready to obtain received powers from Access Points
% to find where the user is
AP1 = input('Enter received power from AP1 = ');
AP2 = input('Enter received power from AP2 = ');
AP3 = input('Enter received power from AP3 = ');
```

```
AP4 = input('Enter received power from AP4 = ');

AP5 = input('Enter received power from AP5 = ');

user = [];

userInput = [AP1 AP2 AP3 AP4 AP5]

for p = 1:1:length(ref_powers)

   userPowerDiff1 = (userInput(1)-ref_powers(p,1))^2;

   userPowerDiff2 = (userInput(2)-ref_powers(p,2))^2;

   userPowerDiff3 = (userInput(3)-ref_powers(p,3))^2;

   userPowerDiff4 = (userInput(4)-ref_powers(p,4))^2;

   userPowerDiff5 = (userInput(5)-ref_powers(p,5))^2;


   userSum = userPowerDiff1 + userPowerDiff2 + userPowerDiff3 + userPowerDiff4 +
userPowerDiff5;

   user = [user,userSum];

end

% finding the closest position in the fingerprint

% to the user to approximate their location in the floor

closest = find(user == min(user))

userPosition = ref_points(closest,:)

%------------------ Plotting the Floor with the user's location
------------------

figure('name', sprintf('Floor Map %d',''))

%putting a title for the Floor Map

title('Floor Map')

% Drawing the walls

for i = 1:1:length(Walls_rect)

   if (i <=11)

       rectangle('Position',Walls_rect(i,:),'FaceColor','w','LineWidth',2)

       axis([0 52 0 20])
```

```
   else
       rectangle('Position',Walls_rect(i,:),'FaceColor','g','LineWidth',2)
       axis([0 52 0 20])
   end
end
% Drawing the Access Points
hold on
for i = 1:1:length(APs)
   rectangle('Position', APs(i,:),'FaceColor','b','LineStyle',':')
end
hold on
plot(userPosition(1,1),userPosition(1,2),'c*')
```