Homework 6 - Wireless Systems

-By Siddhesh Deodhar

src :

-drawCell : This function is used to draw the cell.

-main : This is the main function

Doc :

graph1_hw6 : This is pic of cell

graph2_hw6 : This is graph of received signal with and without beam forming

graph3_hw6 : This is pseudo spectrum estimate via MUSIC algorithm

graph4_hw6 : This is graph of Received Signal Power with Beamforming with 45 degree Angle of Arrival

Code explanation :

Editor - /Users/siddhesh/Downloads/src 2/main.m *

readlabels.m | activation.m | training.m | mainMultiUser.m | pathLoss.m | main.m *

```
1   close all
2   clear all
3   clc
4   cellRadius = 100;
5   centerPos = 0;
6   iValue = 1;
7   jValue = 2;
8   global M
9   M = containers.Map('KeyType','char','ValueType','any')
10  %%
11  %% 1.Create data structure for matlab movie
12  video_flag = 1; % 0 (NO) || 1 (YES)
13  numFrames = 50;
14  if video_flag
15      videoName = 'demo';
16      movieFrames = moviein(numFrames);
17      frame_interf = [];
18  end
19
20  servingCellName = '';
21  servingCellArray = [];
22  distCentArray = {};
23  indArray = {};
24  y = {};
25
26  fc = 1.8e9;
27  lambda = 3.8*10^8/fc;
28  array = phased.ULA('NumElements',10,'ElementSpacing',lambda/2);
29  array.Element.FrequencyRange = [8e8 2e9];
30  t = (1:5:5.*numFrames)';
31  testsig = ones(size(t));
32
```

The picture above indicates the initialization of various parameters like wavelength lambda and cell radius.

Line 28 and 29 calls toolbox function to plot array

```
37
38  for index = 1:numFrames    % Draw each frame in the movie
39
40
41      % Draw the serving cell and label it
42      %drawCell( 0, 100, 'A_1' );
43      clf;
44      drawCell(centerPos, 400);
45      hold on;
46      axis off;
47      % Draw the mobile user at the appropriate location
48      plot(mobilePos(index), 'x' );
49      distToCent = findComplexDistance(mobilePos(index),centerPos);
50      distCentArray{end+1} = distToCent;
51      indArray{end+1} = index;
52      diffReal = real((mobilePos(index)) - centerPos);
53      diffImag = imag((mobilePos(index)) - centerPos);
54      mobAngle = atand(diffImag./diffReal);
55      angle_of_arrival = [mobAngle;0];
56      x = collectPlaneWave(array,testsig,angle_of_arrival,fc);
57      beamformer = phased.PhaseShiftBeamformer('SensorArray',array, ...
58      'OperatingFrequency',1e9,'Direction',angle_of_arrival, ...
59      'WeightsOutputPort',true);
60      [yTemp,wTemp] = beamformer(x);
61      y{end+1} = yTemp(1);
62      % Capture the frame for the movie
63      movieFrames(:, index) = getframe(gcf);
64
```

Line 49 plots the distance to center

Line 55 sets the angle of arrival

Lines 56 and 57 are functions within toolbox which simulate beam former array

```
% %%5. Plot received signal power
y = cell2mat(y);
wavelength = cellRadius/10;  %wavelength
distCentArray = cell2mat(distCentArray);
indArray = cell2mat(indArray);
receivedPowerArray = (1.5e-6).*((wavelength./(4.*pi.*distCentArray)).^2); %received power
beamformedPower = receivedPowerArray.*abs(y); % power of beamformer
[v, w] = unique(servingCellArray,'stable');
startInd = 1;
figure;
    subplot(211)
    plot(t,receivedPowerArray)
    axis tight
    title('Received Signal Power without Beamforming')
    ylabel('Power')
    subplot(212)
    plot(indArray,beamformedPower);
    hold on;
    title('Received Signal Power with Beamforming')
    ylabel('Power')
    xlabel('Seconds')
    hold on;
    figure;
    pmusic(receivedPowerArray,4)
figure;
```

The comments in the code indicate what that lines do.

Plots are made for the received signal with and without beam forming

```
105
106  for space = 1:10
107      dist = 70;
108      angle_of_arrival = [45;0];
109      receivedPowerArray = (1.5e-6).*((wavelength./(4.*pi.*distCentArray)).^2);
110      fc = 1.8e9;
111      lambda = physconst('LightSpeed')/fc;
112      array = phased.ULA('NumElements',10,'ElementSpacing',space*lambda/10);
113      array.Element.FrequencyRange = [8e8 2e9];
114      testsig = ones(size(t));
115      x = collectPlaneWave(array,testsig,angle_of_arrival,fc);
116      beamformer = phased.PhaseShiftBeamformer('SensorArray',array, ...
117      'OperatingFrequency',1e9,'Direction',angle_of_arrival, ...
118      'WeightsOutputPort',true);
119      [y,w] = beamformer(x);
120      beamformedPower = (1.5e-6).*((wavelength./(4.*pi.*dist)).^2)*abs(y);
121      [v, w] = unique(servingCellArray,'stable');
122      startInd = 1;
123      plot(space,beamformedPower,'x');
124      hold on;
125      title('Received Signal Power with Beamforming 45 Angle of Arrival')
126      ylabel('Magnitude')
127      xlabel('Interelement spacing in increments of lambda/10')
128      hold on;
129  end
```

This is the main logic of the code

For loop is used to iterate across the entire array.

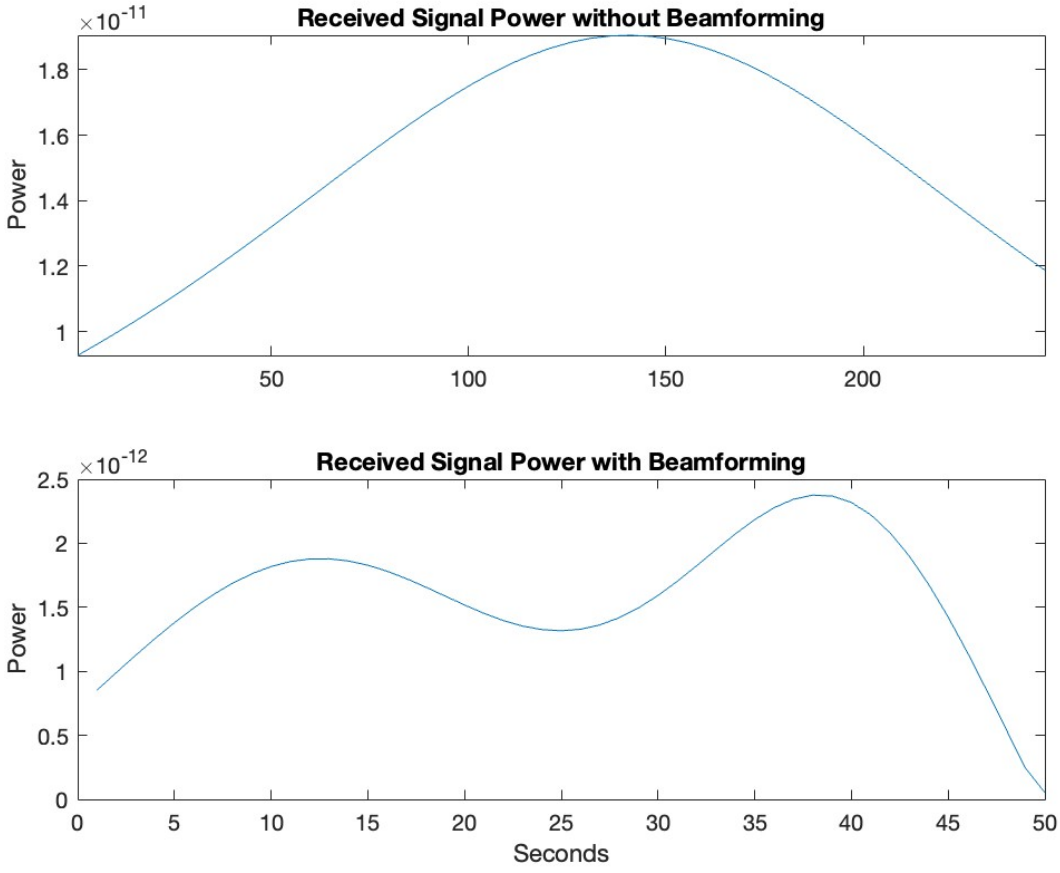
First the Distance and angle of arrival are set in lines 107 and 108. Then the received power is plotted in line 109.

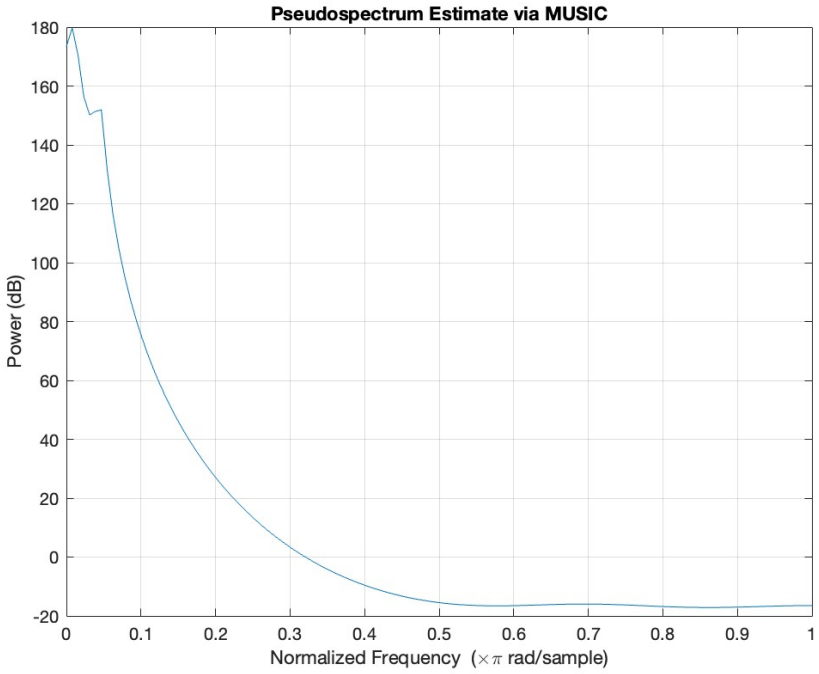
Lines 112 and 113 are used to call toolbox functions to create the array.

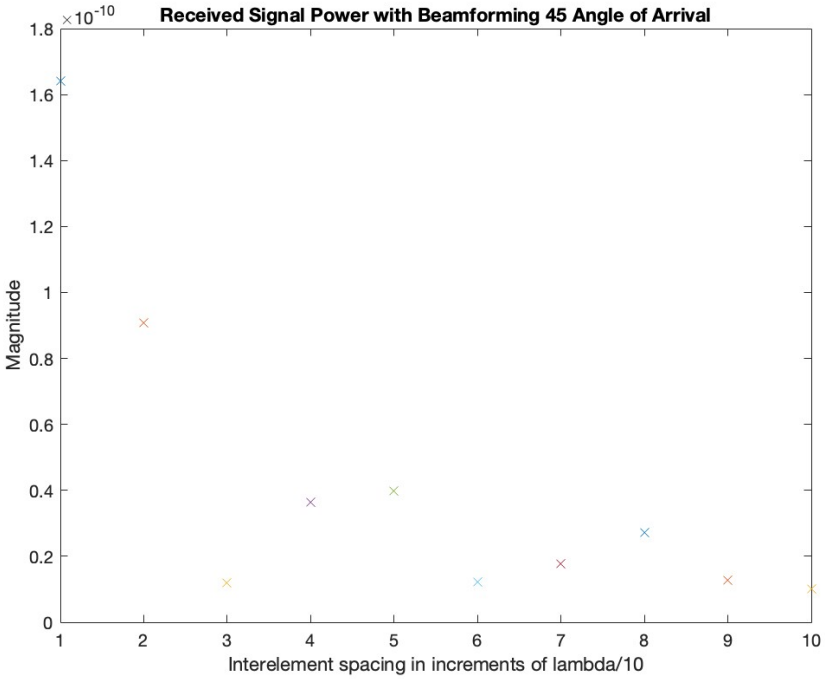
Lines 115 and 116 are used to generate the beam former by calling toolbox.

Beam former power is calculated in line 120

Graphs : The following graphs were generated for the homework-




Pseudospectrum Estimate via MUSIC
Power (dB)
Normalized Frequency (×π rad/sample)
180
160
140
120
100
80
60
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Received Signal Power with Beamforming 45 Angle of Arrival
×10^-10
Magnitude
Interelement spacing in increments of lambda/10
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10

Code execution :

Run the code main.m to generate the simulations. Download all the required toolboxes.