Antenna Lab 1

Ahmed Sherif 19015255, Yahia Walid 19016891, Zyad Alaa 19015728

December 24, 2023

1 Part 1 : linear antenna (dipole of general length)

In this part we graph the antenna pattern of different linear dipole antenna. Figures 1-4 show the results for different inputs.

1.1 Examples

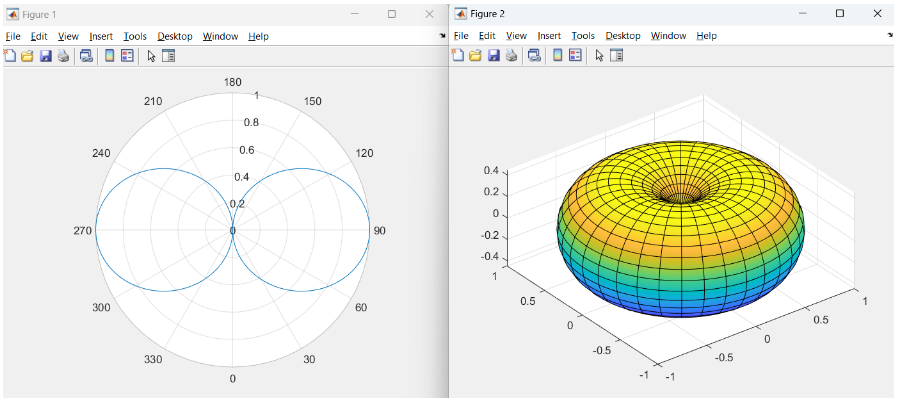

Figure 1: $l = \frac{\lambda}{2}$

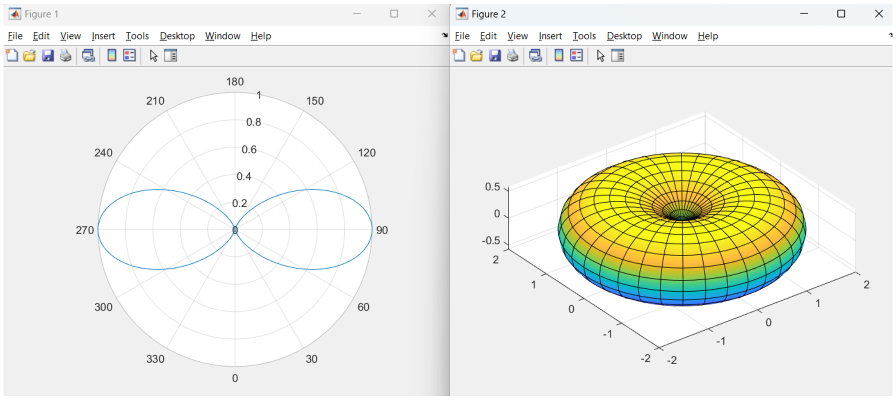
Figure 1
File Edit View Insert Tools Desktop Window Help
180
210
150
240
120
270
90
300
60
330
30
0
1
0.8
0.6
0.4
0.2
0
Figure 2
File Edit View Insert Tools Desktop Window Help
0.5
0
-0.5
2
1
0
-1
-2
-2
-1
0
1
2

Figure 2: $l = \lambda$

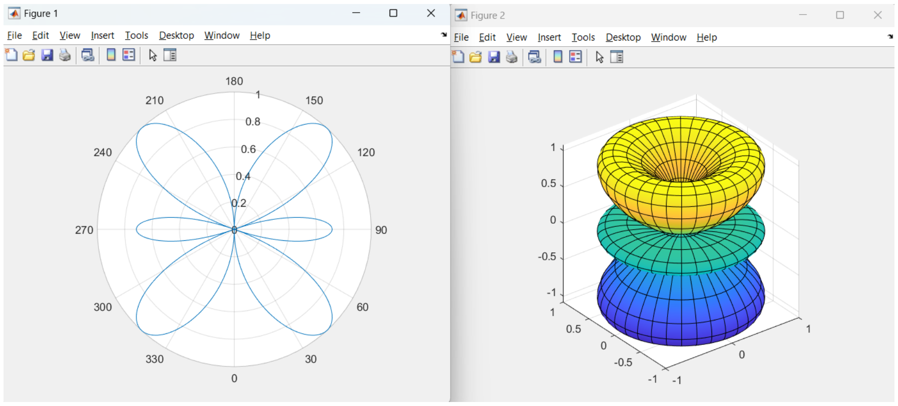
Figure 1
File Edit View Insert Tools Desktop Window Help
180
210
150
240
120
270
90
300
60
330
30
0
1
0.8
0.6
0.4
0.2
Figure 2
File Edit View Insert Tools Desktop Window Help
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
-1
0
1

Figure 3: $l = \frac{3\lambda}{2}$

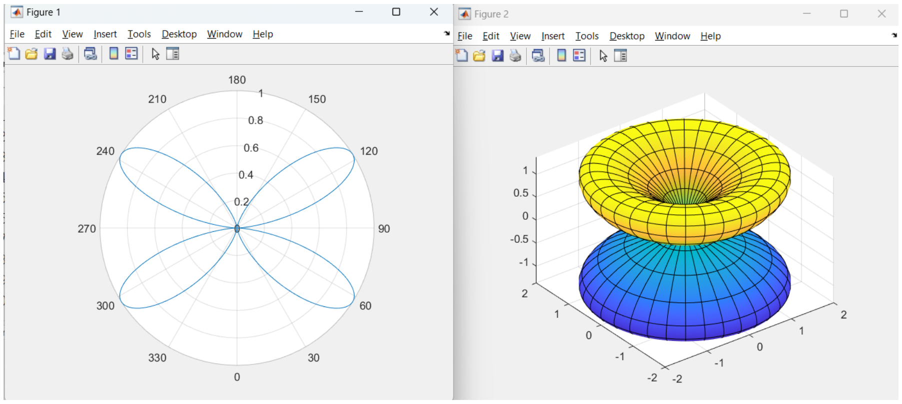
Figure 1
File Edit View Insert Tools Desktop Window Help
180
210
150
240
120
270
90
300
60
330
30
0
1
0.8
0.6
0.4
0.2
Figure 2
File Edit View Insert Tools Desktop Window Help
1
0.5
0
-0.5
-1
2
1
0
-1
-2
-2
-1
0
1
2

Figure 4: $l = 2\lambda$

2 Part 2 : Uniform linear antenna array

In this part we will be graphing the Array Factor of uniform linear antenna arrays. Figures 5-7 show the results for different inputs.

2.1 Examples

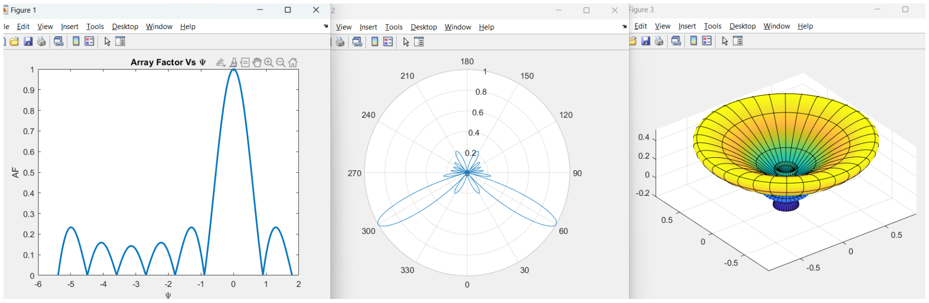

Figure 5: $d = \frac{4\lambda}{7}$, $N = 7$, $\alpha = \frac{-4\pi}{7}$

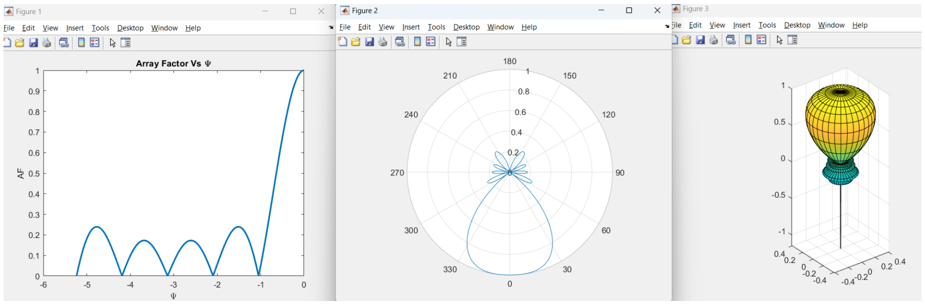

Figure 6: $d = \frac{5\lambda}{12}$, $N = 6$, $\alpha = \frac{-5\pi}{12}$

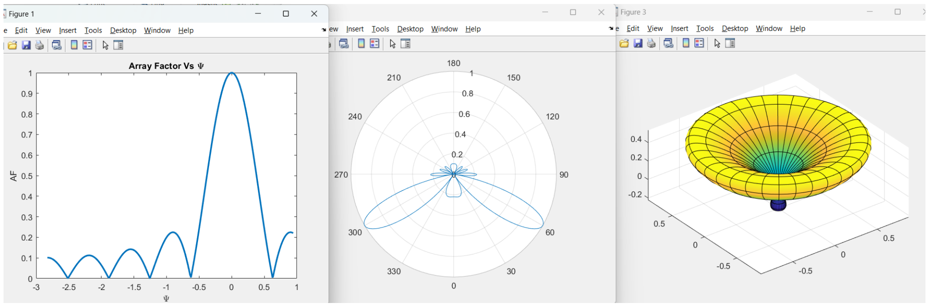

Figure 7: $d = \frac{3\lambda}{10}$, $N = 10$, $\alpha = \frac{-3\pi}{10}$

3 Code

3.1 Main

Code 1 shows the code used in the main file, this code makes use of five custom functions, each of which will be discussed subsequently.

```
%% Part 1: linear antenna (dipole of general length)
L_Lamda_ratio = input("Enter length of dipole relative to lamda: ");
E_plot_2D(L_Lamda_ratio);
E_plot_3D(L_Lamda_ratio);

%% Part 2: Uniform linear antenna array(ULA)
d_lamda_ratio = input("Enter spacing w.r.t lamda (d): ");
N = input("Enter number of elements (N): ");
alpha = input("Enter the progressive shift (alpha): ");

AF_cartesian_2D(N, alpha, d_lamda_ratio);
AF_polar_2D(N, alpha, d_lamda_ratio);
AF_cartesian_3D(N, alpha, d_lamda_ratio);
```

Code 1: Main code

3.2 E_plot_2D

This function plots the E_n pattern in polar co-ordinates in 2D.

```
% This function plots the normalized E pattern in polar co-ordinates in 2D
function E_plot_2D(L_Lamda_ratio)
    % Define constants
    step = 0.01;
    theta = -pi:step:pi;                                       % Theta
    lamda = 0.2;                                               % Wavelength
    B = 2*pi/lamda;                                          % Beta
    L = L_Lamda_ratio * lamda;                        % Wire length


    E_n_theta = (cos(B* L * cos(theta)/2) - cos(B*L/2)) ./ sin(theta);
    E_n_theta = E_n_theta/ max(abs(E_n_theta));
    E_n_theta = E_n_theta .* sign(theta);

    figure
    polarplot(theta, E_n_theta)
    pax = gca;
    pax.ThetaZeroLocation = 'bottom';
end
```

Code 2: E_plot_2D

3.3 E_plot_3D

This function plots the E_n pattern in 3D.

```
% This function plots the E pattern in 3D
function E_plot_3D(L_Lamda_ratio)
    syms phi theta
    % Define constants
    lamda = 0.2;                                    % Wavelength
    B = 2*pi/lamda;                                 % Beta
    L = L_Lamda_ratio * lamda;                      % Wire length

    E_theta = ( cos(B*L *cos(phi)/2) - cos(B*L/2) ) / sin(phi);

    %Converting to Cartesian Coordinates for plotting.
    x = E_theta*sin(phi)*cos(theta);
    y = E_theta*sin(phi)*sin(theta);
    z = E_theta*cos(phi);

    %Plotting E pattern with the defined ranges for x, y
    figure
    fsurf(x,y,z,[0 pi 0 2*pi])

    %for uniform plotting
    axis equal

end
```

Code 3: E_plot_3D

4 AF_cartesian_2D

This function draws the graph of the AF in 2D cartesian co-ordinates.

```
% This function plots the normalized Array Factor Vs Psi in cartesian
% co-ordinates
function AF_cartesian_2D(N, alpha, d_lamda_ratio)
    % Define constants
    step = 0.01;
    gamma = -pi:step:pi;
    psi = alpha + 2*pi*d_lamda_ratio * cos(gamma);

    AF = sin(N*psi/2) ./ (N * sin(psi/2));
    AF = abs(AF);

    figure
    plot(psi, AF, 'linewidth', 2);
    xlabel('\Psi')
    ylabel('AF')
    title('Array Factor Vs \Psi')
end
```

Code 4: AF_cartesian_2D

4.1 AF_polar_2D

This function draws the 2D pattern of AF in polar co-ordinates.

```
% This function plots the normalized Array factor pattern in polar co-ordinates in 2D
function AF_polar_2D(N, alpha, d_lamda_ratio)
    % Define constants
    step = 0.01;
    gamma = -pi:step:pi;
    psi = alpha + 2*pi*d_lamda_ratio * cos(gamma);

    AF = sin(N*psi/2) ./ (N * sin(psi/2));
    AF = abs(AF);

    figure
    polarplot(gamma, AF)
    pax = gca;
    pax.ThetaZeroLocation = 'bottom';
end
```

Code 5: AF_polar_2D

5 AF_cartesian_3D

This function draws the graph of the AF in 3D cartesian co-ordinates.

```
% This function plots the Array Factor in 3D
function AF_cartesian_3D(N, alpha, d_lamda_ratio)
    syms phi theta

    %Defining psi based on the inputs.
    psi = alpha + 2*pi*d_lamda_ratio * cos(phi);

    E_theta = sin(N*psi/2) ./ (N * sin(psi/2));

    %Converting to Cartesian Coordinates for plotting.
    x = E_theta*sin(phi).*cos(theta);
    y = E_theta*sin(phi).*sin(theta);
    z = E_theta.*cos(phi);

    %Plotting E pattern with the defined ranges for x, y
    figure
    fsurf(x,y,z,[0 pi 0 2*pi])

    %for uniform plotting
    axis equal
end
```

Code 6: AF_cartesian_3D