RAJALAKSHMI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

STUDENT RECORD

ACADEMIC YEAR: 2023 – 2024 (ODD SEMESTER)

Subject Code & Name: CEC335 – ANTENNA DESIGN

Year & Semester: III/V

STUDENT DETAILS			
ROLL NO	REGISTER NUMBER	STUDENT NAME	SECTION

RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

BONAFIDE CERTIFICATE

ACADEMIC YEAR-2023-2024 **YEAR/SEMESTER-III/V**

BRANCH –ECE

UNIVERSITY REGISTER NUMBER -

Certified that this is the bonafide record of work done by the above student in the **CEC-ANTENNA DESIGN during *the year* 2023-2024.**

Signature **Signature**

Faculty-in-Charge **HOD**

Submitted for the practical examination held on …………………………….

Internal Examiner **External Examiner**

INSTITUTE VISION

- To set a benchmark in the field of engineering education by providing quality technical education that fosters the spirit of learning, research and globally competent professionalism.

INSTITUTE MISSION

- To impart education that caters to the growing challenges of the industry and social needs of our nation.
- To constantly upgrade the standards of teaching and learning in the field of engineering and technology while promoting a healthy research atmosphere.
- To foster a healthy symbiosis with the industry through meaningful and dynamic interactions.

DEPARTMENT VISION

- To initiate high quality technical education and to nurture young minds towards creative thinking that inspires them to undertake innovations in the field of Electronics and Communication Engineering (ECE) and be competent in the global arena.
- To Emphasize on the student body to carry out research for the service of our Nation and to the Society at large.

DEPARTMENT MISSION

- Constantly upgrade engineering pedagogy that caters to the growing challenges of the industry.
- Develop conceptual learning that leads towards critical and innovative thinking.
- Establish good harmony with industry that fills the gap between academia and the outside world enabling the students to prepare for diverse and competitive career paths.

PROGRAMME EDUCATIONAL OBJECTIVES

- **PEO1:** To enable graduates to pursue research or have a successful career in academia or industries associated with Electronics and Communication Engineering, or as entrepreneurs.
- **PEO2:** To provide students with strong foundational concepts and also advanced techniques and tools in order to enable them to build solutions or systems of varying complexity.
- **PEO3:** To prepare students to critically analyze existing literature in an area of specialization and ethically develop innovative and research-oriented methodologies to solve the problems identified.

PROGRAMME SPECIFIC OUTCOMES

- To analyze, design and develop solutions by applying foundational concepts of electronics and communication engineering.
- To apply design principles and best practices for developing quality products for scientific and business applications
- To adapt to emerging information and communication technologies (ICT) to innovate ideas and solutions to existing/novel problems

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest.

SYLLABUS

CEC335 **ANTENNA DESIGN** **L T P C 2 0 2 3**

COURSE OBJECTIVES:

- To introduce the basic concepts of antenna arrays for smart antenna design.
- To discuss the random variables and processes for angle of arrival (AOA) estimation.
- To describe different algorithms used for AOA estimation.
- To introduce the concepts of fixed weight beamforming.
- To introduce the concept of adaptive beamforming.

LISTOF EXPERIMENTS

1. Write a MATLAB code to estimate the radiation pattern of a linear array
2. Write a MATLAB code to estimate the radiation pattern of a N element uniform array
3. Write a MATLAB code to estimate the weights of the array.
4. Write a MATLAB code to dynamically alter the main lobe direction based on the information of AOA.
5. Write a MATLAB code to estimate the AOA using ESPRIT algorithm
6. Write a MATLAB code to estimate the AOA using MUSIC algorithm
7. Write a MATLAB code to estimate the weights of the array factor and the mean square error.
8. Write a MATLAB code to estimate basic vector and Matrices properties.

COURSE OUTCOMES:

At the end of the course students will be able to:

- At the end of the course the students will be able to describe the basics of phased array antennas.
- Students will be able to understand random process and its application in Smart antennas.
- Student will able to estimate the weights of the antenna array based on the angle of arrival.
- Student will able to analyze the fixed weight beamforming in smart antennas
- Students will be able to analyze adaptive beamforming in smart antennas.
- Students will be able to design a microstrip patch antenna using the CST/HFSS tool.

LIST OF EXPERIMENTS

S.No	Name of Experiments
	SIMULATION USING MATLAB
1.	Write a MATLAB code to estimate the radiation pattern of a linear array.
2.	Write a MATLAB code to estimate the radiation pattern of a N element uniform array.
3.	Write a MATLAB code to estimate the weights of the array.
4.	Write a MATLAB code to dynamically alter the main lobe direction based on the information of AOA.
5.	Write a MATLAB code to estimate the AOA using ESPRIT algorithm
6.	Write a MATLAB code to estimate the AOA using MUSIC algorithm
7.	Write a MATLAB code to estimate the weights of the array factor and the mean square error.
	CONTENT BEYOND THE SYLLABUS
8.	Write a MATLAB code to estimate basic vector and Matrices properties.

RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

INDEX

S.No	Date	Name of Experiments	Sign
SIMULATION USING MATLAB			
1.		Write a MATLAB code to estimate the radiation pattern of a linear array.	
2.		Write a MATLAB code to estimate the radiation pattern of a N element uniform array.	
3.		Write a MATLAB code to estimate the weights of the array.	
4.		Write a MATLAB code to dynamically alter the main lobe direction based on the information of AOA.	
5.		Write a MATLAB code to estimate the AOA using ESPRIT algorithm	
6.		Write a MATLAB code to estimate the AOA using MUSIC algorithm	
7.		Write a MATLAB code to estimate the weights of the array factor and the mean square error.	
CONTENT BEYOND THE SYLLABUS			
8.		Write a MATLAB code to estimate basic vector and Matrices properties.	

MATLAB PROGRAMS

Expt. No:	LINEAR ARRAY	Date:

AIM:
To write a program for linear array and observe the beam pattern.

REQUIREMENTS:

MATLAB, Personal computer.

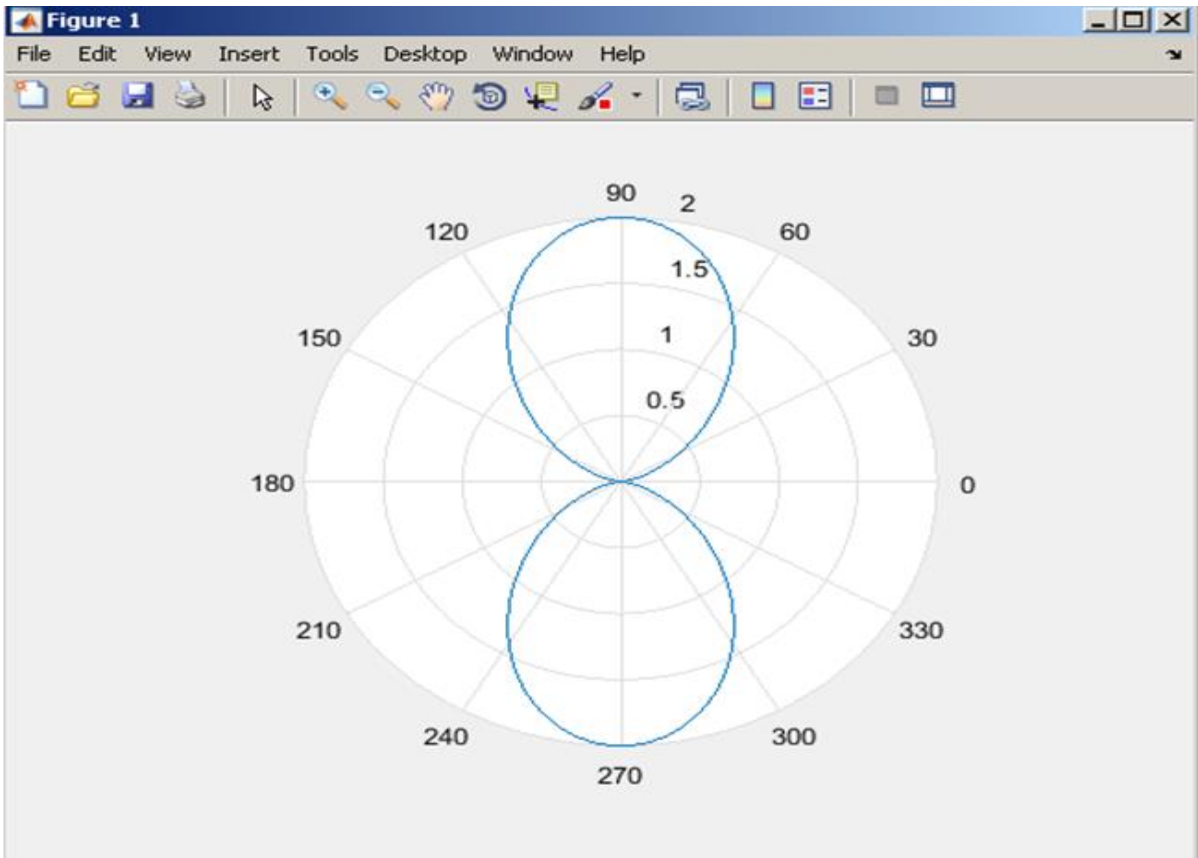
ALGORITHM:
1. Open the MATLAB software
2. Clear the command window and workspace using clc
3. Using input function obtain the inputs for lambda, alfa ,N and d from the user.
4. Calculate the value of B using the formula[B=(2*pi/lambda)]
5. Generate an array of angles theta from 0 to 2*pi with a step of pi/100.
6. Calculate the array w representing the phase difference for each angle using [w = alfa + B*d*cos(theta)]
7. Calculate the array AF representing the array factor [AF=sinc(N*(w./2))./sinc(w./2)]
8. Plot the beam pattern in polar coordinates using the polar function, with angles theta and array factor AF.

PROGRAM:
```
clc;
lambda=input('enter the value of wave length= ');
N=input('enter the no.of elements= ');
alfa=input('enter your phase= ');
d=input('enter the separation distance between elements= ');
B=(2*pi/lambda);
theta=0:0.01:2*pi;
w=alfa+B*d.*cos(theta);
AF=sin(N*(w./2))./sin(w./2);
polar(theta,AF)
```

OUTPUT:

enter the value of wave length= 1
enter the no. of elements= 2
enter your phase= 0
enter the separation distance between elements= 0.5

PLOT:



RESULT:

Thus the program for linear array was executed and the beam pattern was observed.

Expt. No:	UNIFORM NTH ELEMENT ARRAY	Date:

AIM:
To write a program for Nth element array and observe the beam pattern.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:
1. Clear the command window and workspace using clc.
2. Prompt the user to enter the number of elements N, the separation distance between elements d, and the wavelength lambda using input function.
3. Calculate the normalized spatial frequency B using the formula B = (2*pi/lambda).
4. Define the angle of arrival range theta_range from -pi to pi with a step of pi/180.
5. Create an array AF to store the Array Factor values, initialized with zeros and the same size as theta_range.
6. Loop through each angle theta in the theta_range.
7. Calculate the phase difference w for each angle using w = B*d*cos(theta).
8. Calculate the Array Factor value for that specific angle using the sum of complex exponentials and store it in the AF array.
9. Normalize the Array Factor values by dividing them by the maximum value to make the peak equal to 1.
10. Plot the polar diagram using the polarplot function, with theta_range as the angles and AF as the Array Factor values.

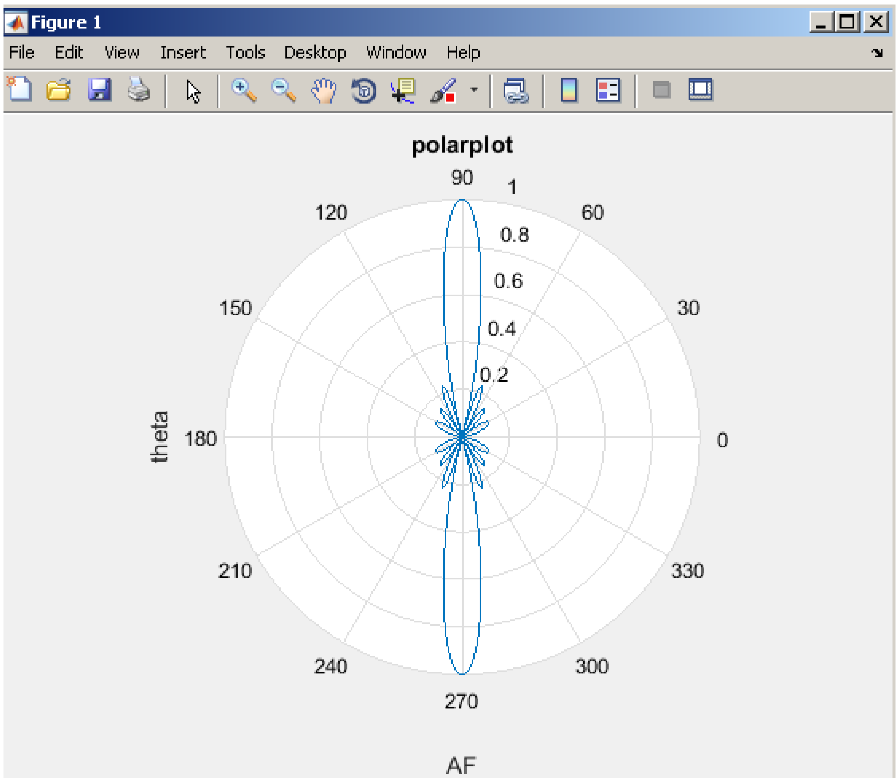
PROGRAM:
```
clc;
N=input('enter the number of elements(N):');
d=input('enter the separation distance between elements(d):');
lambda=input('enter the wavelength(lambda):');
Beta =(2*pi/lambda);
theta = 0:0.01:2*pi;
AF=zeros(size(theta));
for k=1:length(theta)
```

```
    w=Beta*d*cos(theta(k));
    AF(k)=abs(sum(exp(1i*(0:N-1)*w)))/N;
end
AF=AF/max(AF);
polar(theta,AF);
xlabel('AF');
ylabel('theta');
title('polarplot');
```

OUTPUT:

enter the number of elements(N):8
enter the separation distance between elements(d):0.5
enter the wavelength(lambda):1

PLOT:



RESULT:

Thus the program for Nth element array was executed and the beam pattern was observed.

Expt. No:	WEIGHT OF THE ARRAY ELEMENTS USING LEAST SQUARE METHOD	Date:

AIM:
To write a program for calculating the weight of the arrray elements using least square method.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1.Clear the command window and workspace using clc.

2.Enter the following parameters when prompted:

N: Number of elements in the array

d: Separation distance between elements

lambda: Wavelength of the signal

3.Calculate the array factor.

4.Use the least squares method to find the complex weights of the array elements

5.Normalize the weights to have a unit magnitude.

6.The array weights are displayed in the console window.

PROGRAM:

```
clc;
% Input parameters
N = input('Enter the number of elements (N): ');
d = input('Enter the separation distance between elements (d): ');
lambda = input('Enter the wavelength (lambda): ');
```

```
% Calculate the array factor (AF) for the ULA
theta_range = -pi:pi/180:pi;
B = (2*pi/lambda);
AF = exp(1i*(0:N-1)'*B*d*cos(theta_range));
% Desired radiation pattern (example: broadside response)
desired_response = ones(1, length(theta_range));
% Use the least squares method to find the weights
w = lscov(real(AF'), desired_response');
% Normalize the weights
W = w / norm(w);
% Display the complex weights of the array elements
title('Weights of the array elements:');
disp(W);
stem(W);
xlabel('Number of Elements');
ylabel('Array Factor');
```

OUTPUT:

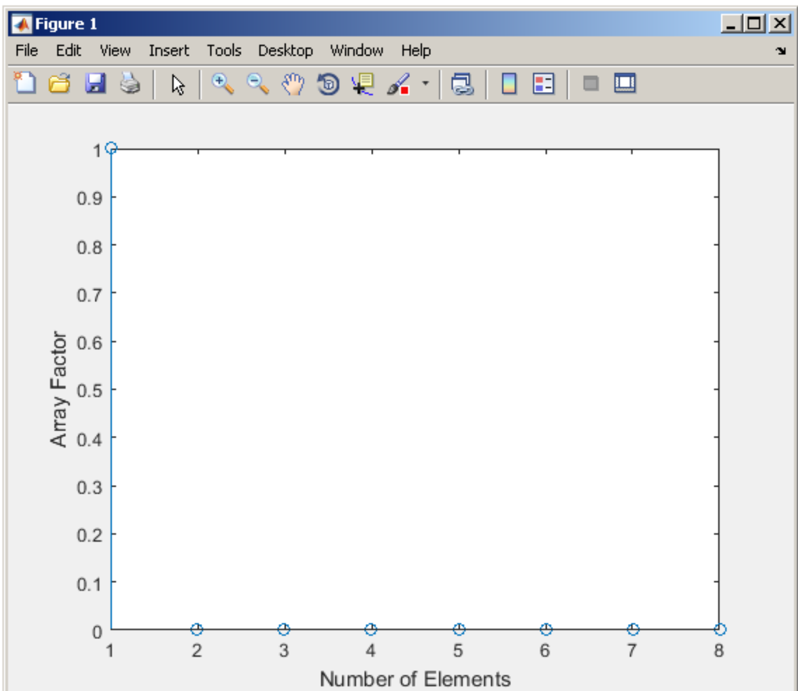
```
Enter the number of elements (N): 8
Enter the separation distance between elements (d): 0.5
Enter the wavelength (lambda): 1
    1.0000
   -0.0000
   -0.0000
   -0.0000
   -0.0000
```

0.0000

0.0000

-0.0000

PLOT:



RESULT:

Thus the program for calculating the weight of the array elements using binomial window method was executed and weight plot was observed.

Expt. No:	WEIGHT OF THE ARRAY ELEMENTS USING BINOMIAL METHOD	Date:

AIM:
To write a program for calculating the weight of the arrray elements using binomial window method.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1.Clear the command window and workspace using clc.
2.Enter the following parameters when prompted:
    N: Number of elements in the array
    d: Separation distance between elements
    lambda: Wavelength of the signal
3.Calculate the array factor.
4.Use the least squares method to find the complex weights of the array elements
5.Normalize the weights to have a unit magnitude.
6.The array weights are displayed in the console window.

PROGRAM:

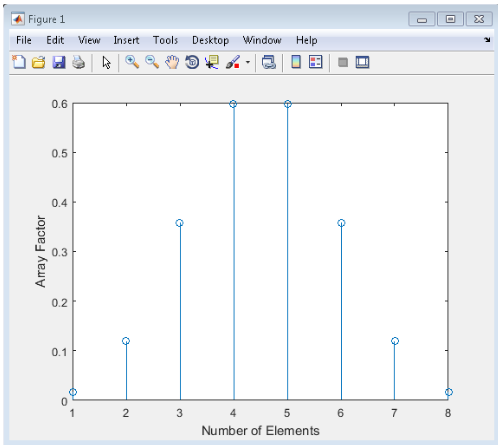
```
clc;
% Input parameters
N = input('Enter the number of elements (N): ');
d = input('Enter the separation distance between elements (d): ');
lambda = input('Enter the wavelength (lambda): ');
% Calculate the array factor (AF) for the ULA
theta_range = -pi:pi/180:pi;
B = (2*pi/lambda);
AF = exp(1i*(0:N-1)'*B*d*cos(theta_range));
% Desired radiation pattern (example: broadside response)
desired_response = ones(1, length(theta_range));
% Use the least squares method to find the weights
w = diag(rot90(pascal(N)));
```

```
% Normalize the weights
W = w / norm(w);
% Display the complex weights of the array elements
title('Weights of the array elements:');
disp(W);
%plot(W);
stem(W);
xlabel('Number of Elements');
ylabel('Array Factor');
```

OUTPUT:

```
Enter the number of elements (N): 8
Enter the separation distance between elements (d): 0.5
Enter the wavelength (lambda): 1
    0.0171
    0.1195
    0.3585
    0.5974
    0.5974
    0.3585
    0.1195
    0.017
```

PLOT:



RESULT:

Thus the program for calculating the weight of the array elements using binomial window method was executed and weight plot was observed.

Expt. No:	ALTER MAIN LOBE DIRECTION BASED ON AOA	Date:

AIM:
To write a program for altering the main lobe direction of the linear arrray elements using Angle of Arrival (AOA) method.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1. Define the parameters of the antenna array
2. Specify the angle of arrival (AOA)
3. Convert the AOA from degrees to radians
4. Calculate the phase shifts for each element in the array based on the AOA
5. Create an array of angles (theta) over which to evaluate the array pattern
6. Compute the array pattern by summing the contributions from each element
7. Normalize the array pattern to make it easier to interpret
8. Plot the array pattern.

PROGRAM:

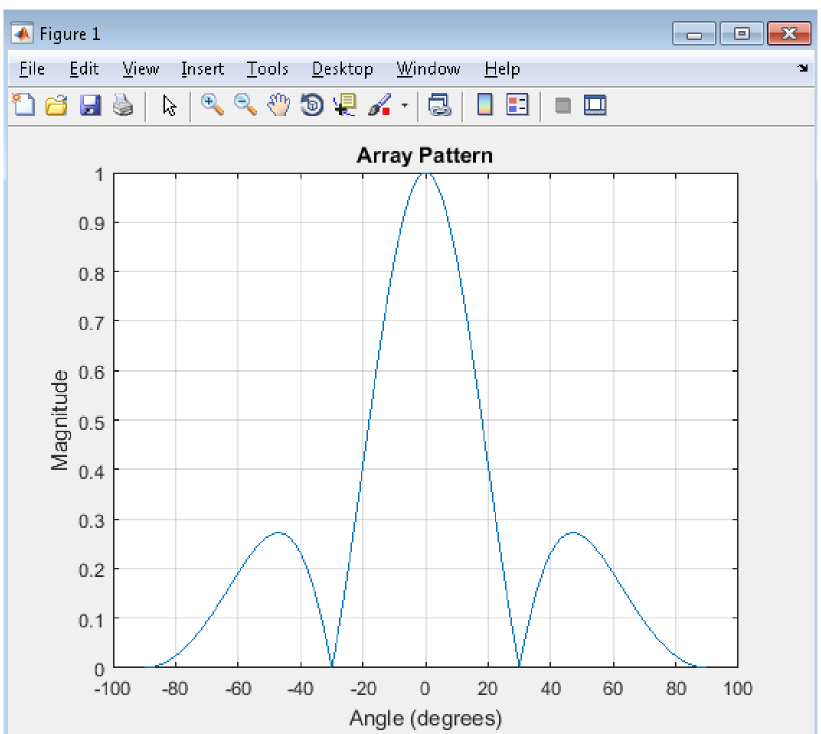
```
N = input('Enter the no of elements=   ');     % Number of elements in the linear array
wavelength = input('Enter the wavelength=  ');       % Wavelength of the signal (arbitrary units)
element_spacing = 0.5 * wavelength; % Spacing between elements (arbitrary units)
% AOA (Angle of Arrival) in degrees
D =input('Enter the angle in degree=  ');
% Convert AOA from degrees to radians
R = D*pi/180;
% Calculate the phase shifts for each element
phase_shifts = (0:N-1) * 2 * pi * element_spacing * sin(R) / wavelength;
% Create an array pattern using the pattern function
theta = linspace(-pi/2, pi/2, 1000); % Theta is the angle over which to evaluate the array pattern
array_pattern = abs(sum(exp(1i * phase_shifts.' * sin(theta)), 1));
```

```
% Normalize the array pattern
array_pattern = array_pattern / max(array_pattern);
% Plot the array pattern
figure;
P = theta*180/pi;
plot(P, array_pattern);
title('Array Pattern');
xlabel('Angle (degrees)');
ylabel('Magnitude');
grid on;
```

OUTPUT:

Enter the no of elements = 4
Enter the wavelength = 1
Enter the angle in degree = 90

PLOT:



RESULT:

Thus the program for altering the main lobe direction of the linear array elements using Angle of Arrival (AOA) method was executed and Array pattern was observed.

Expt. No:	AOA USING ESPRIT ALGORITHM	Date:

AIM:
To write a program for AOA using ESPRIT algorithm.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

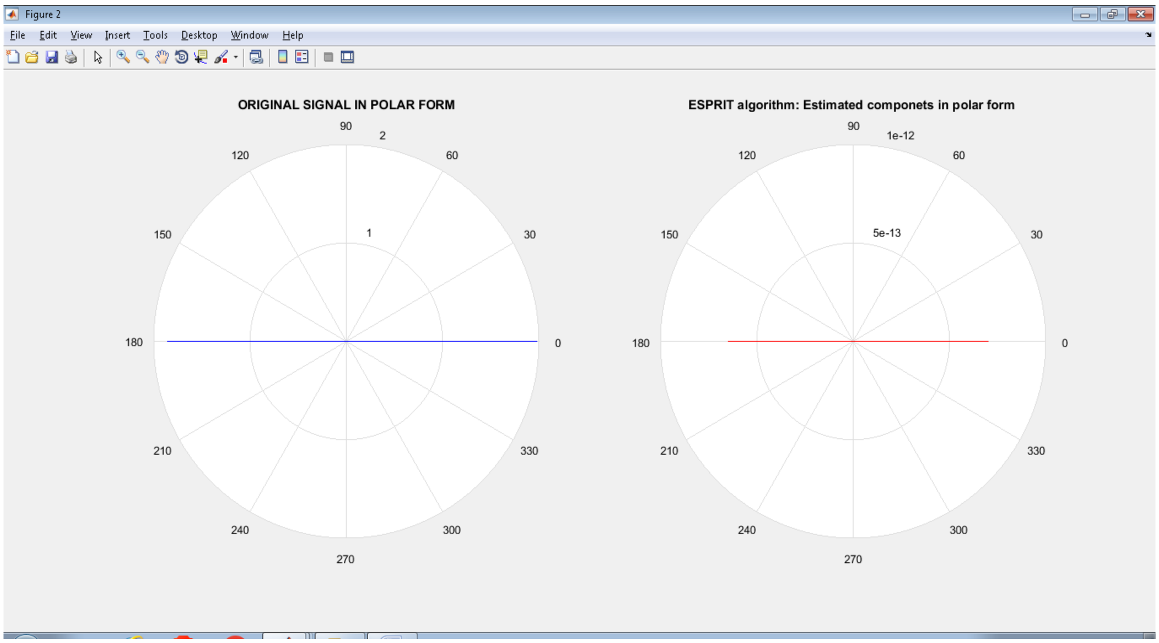
1.Generate the sample data.
2.Define the esprit algorithm.
3.Display estimated frequency.
4.Plot the signal and its estimated components in polar form.

PROGRAM:

```
fs=1000;
t=0:1/fs:1;8
frequencies=[50,150];
amplitudes=[1,0.5];
x=amplitudes(1)*sin(2*pi*frequencies(1)*t)+ amplitudes(2)*sin(2*pi*frequencies(2)*t);
noise_power=0.1;
x=x+sqrt(noise_power)*randn(size(t));
M=2;
n=length(x);
l=n-M+1;
X=toeplitz(x(1:l),x(l:n));
[U,n,N]=svd(X);
Un=U(:,1:M);
Ur=U(:,2:M+1);
theta = -angle(eig(Ur'*Un));
f_est=(fs/(2*pi))*sort(theta);
disp('estimated frequencies(HZ):');
disp(f_est);
figure;
```

```
subplot(1,2,1);
polar(angle(x),abs(x),'b');
title('ORIGINAL SIGNAL IN POLAR FORM');
subplot(1,2,2);
for i=1:M
  component=amplitudes(i)*sin(2*pi*f_est(i)*t);
  polar(angle(component), abs(component), 'r--');
  hold on;
end
title('ESPRIT algorithm: Estimated components in polar form ');
```

PLOT:



RESULT:

Thus the program elements for ESPRIT algorithm using Angle of Arrival (AOA) method was executed and Array pattern was observed.

RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

Expt. No:	ARRAY FACTOR AND MEAN SQUARE ERROR	Date:

AIM:
To write a program to calculate array factor and mean square error.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1.Generate the sample data.
2.Define the esprit algorithm.
3.Display estimated frequency.
4.Plot the signal and its estimated components in polar form.

PROGRAM:

```
N=input('Enter the Array Elements');
d=input('Enter the  separation distance between elements ');
theta_steer_deg=input('Enter the AOA');
theta_steer_rad=(theta_steer_deg)*(pi/180);
theta=linspace(-pi/2,pi/2,1000);
af=zeros(size(theta));
for k=1:N
   af=af+exp(1j*(k-1)*2*pi*d*sin(theta));
end
af=abs(af)/max(abs(af));
mean_error_rad=abs(theta-theta_steer_deg);
mean_error_deg=(mean_error_rad)*(180/pi);
```

```
mean_error=mean(mean_error_deg);
figure;
polar(theta,af);
title('array factor');
disp(['mean error(degree):',num2str(mean_error)]);
```

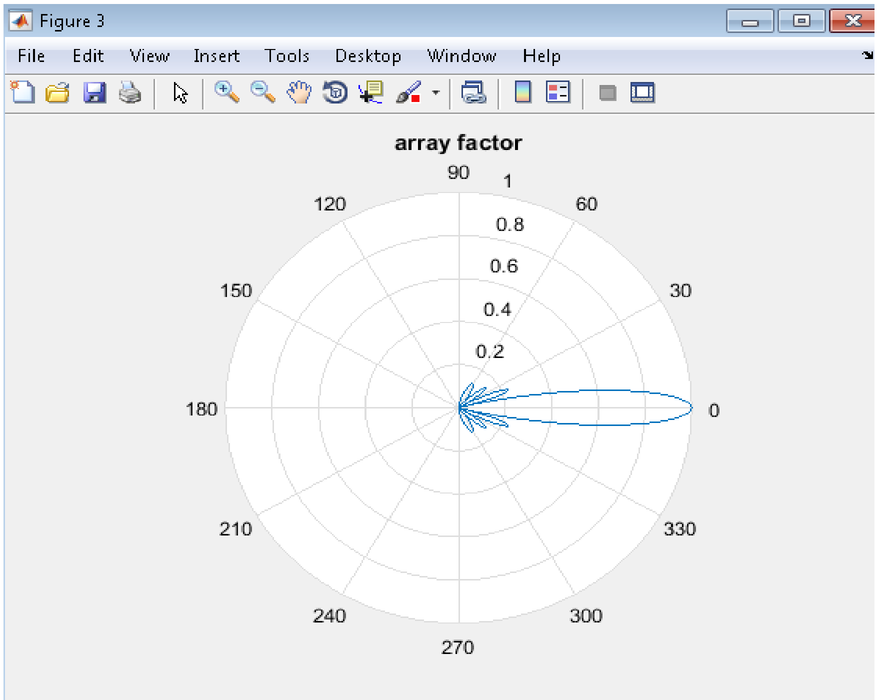
OUTPUT:

Enter the number of elements (N): 8

Enter the separation distance between elements (d): 0.5

Enter the AOA =30

PLOT:



RESULT:

Thus the program elements for array factor and mean square error algorithm method was executed and Array pattern was observed.

RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

Expt. No:	MUSIC AOA ESTIMATION	Date:

AIM:

To write a program for AOA using MUSIC algorithm.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1.Generate the sample data.
2.Define the music algorithm.
3.Display estimated frequency.
4.Plot the signal and its estimated components in polar form.

PROGRAM:

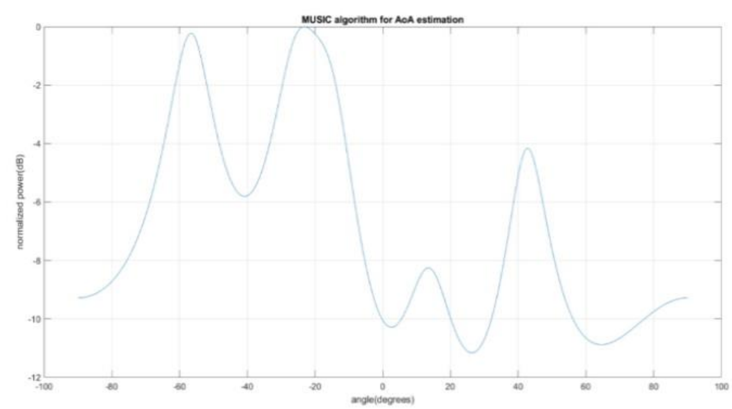
```
% Define your parameters
M = 6; % Number of sensors in the array
N = 4; % Number of sources
D = 0.5; % Sensor spacing (in wavelengths)
% Generate a ULA (Uniform Linear Array) steering matrix
A = @(theta) exp(-1i * 2 * pi * d * (0:M-1)' * sind(theta));
% Generate sample data for two sources at 30 and 60 degrees
Theta_sources = [30, 60]; % Actual AoA of sources
SNR = 10; % Signal-to-Noise Ratio
Num_samples = 1000; % Number of samples
Noise = sqrt(0.5 / 10^(SNR / 10)) * (randn(M, num_samples) + 1i *
randn(M, num_samples));
Signal = A(theta_sources(1)) + A(theta_sources(2));
X = signal + noise;
% Compute the covariance matrix
R = X * X' / num_samples;
% Estimate the signal subspace
[U, ~] = eig®;
% Define the angular grid for AoA estimation
```

```
Theta_grid = -90:0.5:90; % Adjust the range and step size as needed
% Initialize the spatial spectrum
Pmusic = zeros(size(theta_grid));
% Calculate the MUSIC spatial spectrum
For I = 1:length(theta_grid)
 A_theta = A(theta_grid(i));
 Pmusic(i) = 1 / (a_theta' * (U(:, N+1:end) * U(:, N+1:end)') * a_theta);
End
% Plot the spatial spectrum
Figure;
Plot(theta_grid, 10 * log10(abs(Pmusic) / max(abs(Pmusic))));
Xlabel('Angle (degrees)');
Ylabel('Normalized Power (dB)');
Title('MUSIC Algorithm for AoA Estimation');
Grid on
```

OUTPUT:

```
  84.8000  -60.2000    0.5000  -20.7000
```

PLOT:



RESULT:

Thus the program elements for MUSIC algorithm using Angle of Arrival (AOA) method was executed and Array pattern was observed.

RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

CONTENT BEYOND SYLLABUS

Expt. No:	CONTENT BEYOND SYLLABUS	Date:

AIM:
To determine the basic vector and matrix operation using matlab.

REQUIREMENTS:

MATLAB, Personal computer.

ALGORITHM:

1.Get the user data.

2.Use appropriate formulae and compute the operations.

3. Display the obtained output.

PROGRAM:

i. To find determinant of a matrix

```
>>A=[1 2 0 ; 3 2 1 ; 5 1 -1];
>>det(A)
```

ii. To find multiplication of two matrix

```
>>A=[1 -2 ; 3 4];
>>B=[7 3 ; -1 5];
>>A*B
```

iii. To find trace of a matrix

```
>>A=[1 -2 ; 3 4];
>>trace(A)
```

iv. To find transpose of a matrix

```
>> A=[1 -2 ; 3 4];
>>transpose(A)
```

v. To find inverse of a matrix

```
>>A=[1 3 ; -2 5];
>>inv(A)
```

vi. To find eigen values and eigen vectors of a matrix

```
>>A=[1 2  ; 3 5];
>>[EV , V] = eig(A);
>>EV
>>diag(V)
```

vii. To produce an identity matrix

```
>>n=”Enter the no. of rows and columns”;
>>I=eye(n)
>>n
```

OUTPUT:

i. ans = -8

ii. ans =

```
9     -7
17    29
```

iii. ans = 5

iv. ans =

```
0.4545    -0.2727
0.1818     0.0909
```

v. ans =

```
0.4545    -0.2727
0.1818     0.0909
```

vi. EV =

```
-0.8646    -0.3613
 0.5025    -0.9325
```

ans =

```
-0.1623
 6.1623
```

RESULT:

Thus, the basic vector and matrix operation was performed using MATLAB.