Gunjan K Khut
UID–115813083

Assignment 2

Task 1:

Signal Detection

1. Plotting the signals.
   - Selected file **'3.wav'** from the data folder. And used this for the entire assignment.
   - Used **'audio read function'** to read the sound signal and store it in a variable. Then found out the sample rate of the recording. Also, Played the file using the **'soundsc function'**.
   - Then plotted the time series of the transmitted and received signals with x-axis as time in seconds and the y-axis as amplitude.
2. Then computed the distance from base station using the transmitted and the received signals
   - Used Cross-correlation function (xcorr) to calculate the correlation between the transmitted signals and the received signal. This function returns the correlation and the index for each correlation as lag.

     ```
     [corr,lag] = xcorr(xr, xt);
     ```

     Where xt is one data set and xr is the other. With this computed the index of the max value for the correlation and then used that index to look up the lag. This is the delay in samples, within the precision of the sample rate:
   - The used the sample difference to calculate time difference and then computed the distance.
   - The reported distance is **442.2656 meters**.

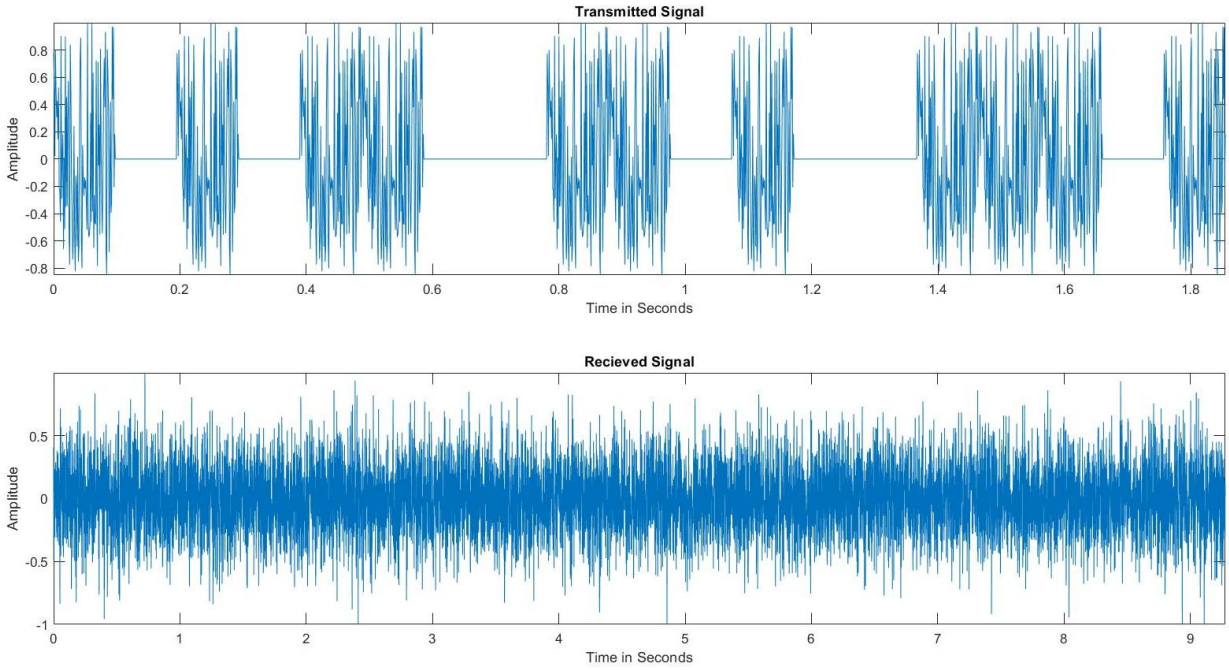


Figure 1: Transmitted and Received Signals

Gunjan K Khut
UID–115813083

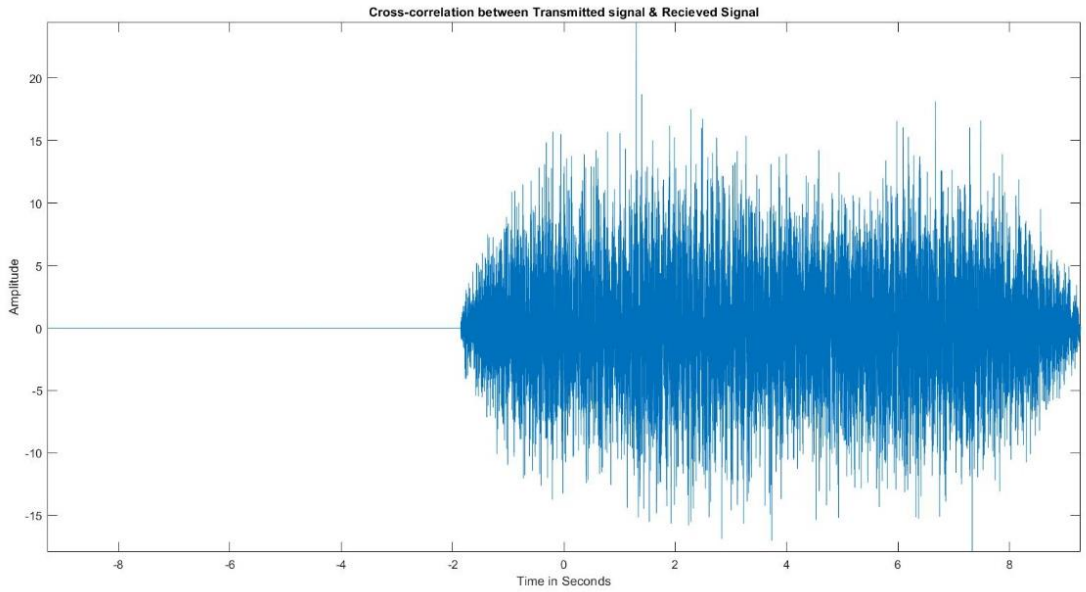


Figure 2: Cross-correlation between Transmitted signal & Received Signal

Task 2:

Directional Gain of ULAs

1. Used the given information to define the parameters and Plotted the directional gain of 10-element ULA for the target frequency 2 kHz. Here the inter-element distance of the ULA was equal to half of the wavelength of the target signal.

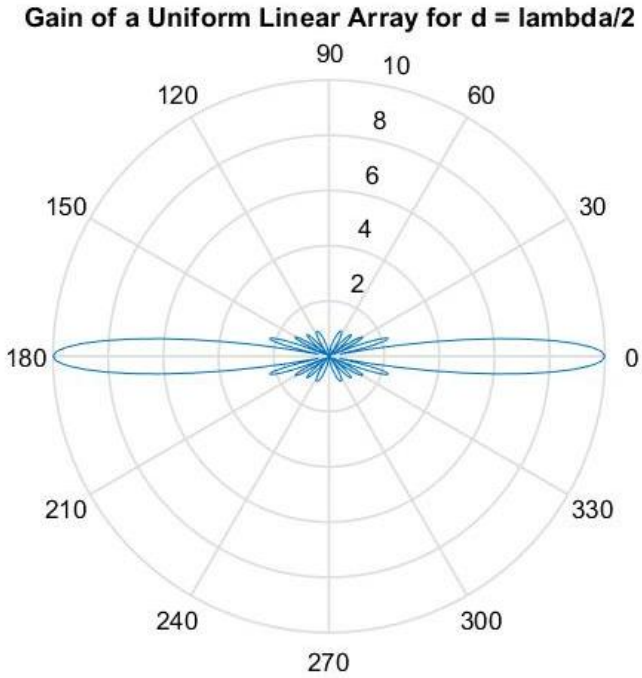


Figure 3. Directional gain of the ULA for interelement distance = lambda/2

Gunjan K Khut
UID–115813083

2. Used the given information to define the parameters and Plotted the directional gain of 10-element ULA for the target frequency 2 kHz. Here the inter-element distance of the ULA was equal to twice of the wavelength of the target signal.

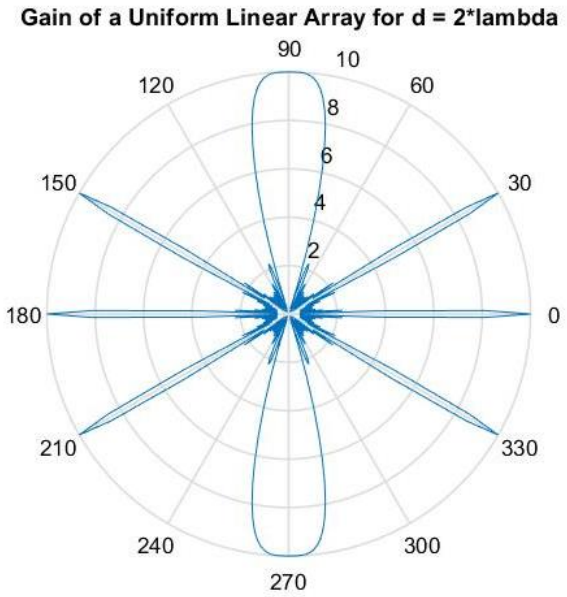


Figure 4. Directional gain of the ULA for interelement distance = 2*lambda

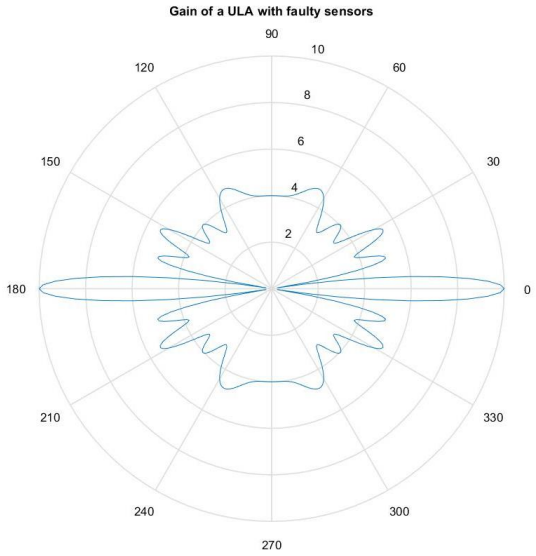
3. When the inter-element distance is twice the wavelength, we get a major lobe and few major end fire lobes
   When the inter-element distance is half of the wavelength, we get a major lobe and many end fire lobes.
   We can say that the width of the lobes or beam width of the antenna is related to the ratio of inter-element distance (d) along with the wavelength (lambda), such that if the ratio is smaller than the beam width is larger and vice-versa.

Task 2: Bonus

1. Considered that the 2nd, 3rd, and 4th microphones of the 10-element ULA are faulty and not recording any sound (output is zero for each sample).
2. The new array factor would be calculated by removing the antenna element all together. This increases the distance between the adjacent antenna elements of the removed antenna element. Here we are using 10 antenna elements with a uniform linear array with inter element spacing of lambda/2. And the 2th,3rd and 4th antennas are faulty. Then the spacing between element 1st and 5th would become 2*lambda. Now we calculate the new array factor.
3. Then plotted the directional gain of this faulty ULA for target frequency 2kHz and inter-element distance equal to the half of the wavelength.

Gunjan K Khut
UID–115813083

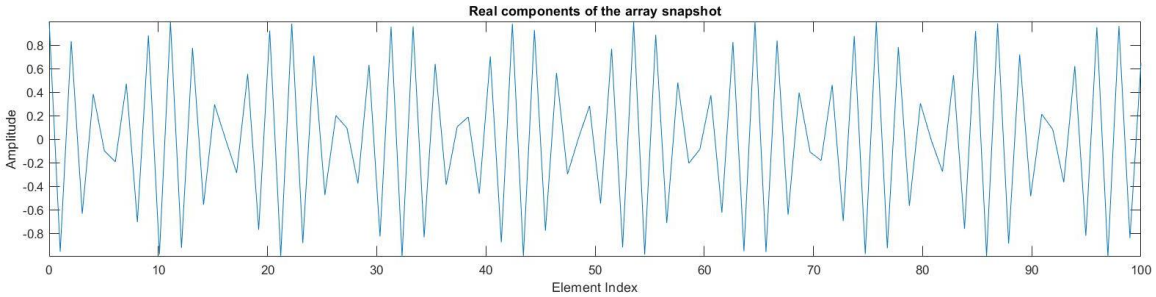
4. The difference of the gain of this faulty ULA compared to a standard ULA with the same configuration.
5. It is seen that as the number of array elements increases the Gain (or Directivity) of the array increases.

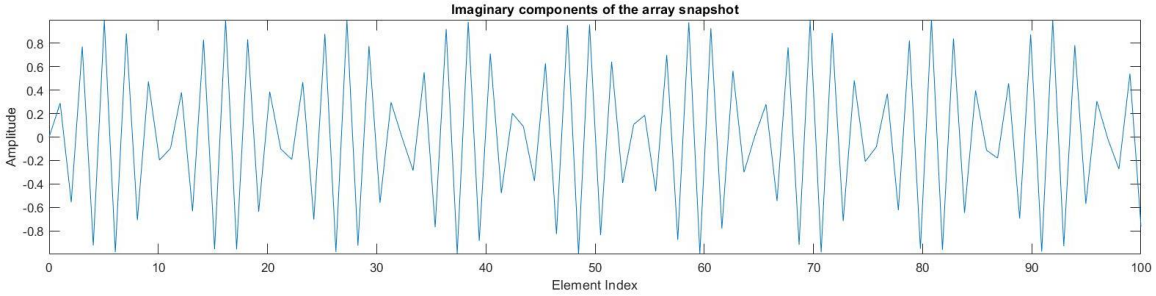


Task 3:

Direction of Arrival

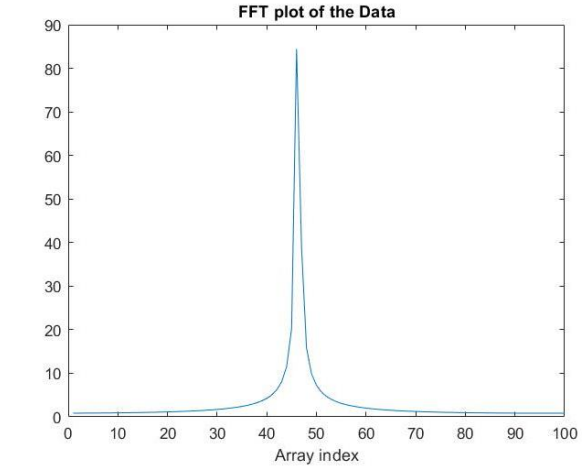
1. Selected file '3.csv' from the data folder. Used 'csvread' to read the file and extract the data. i.e., the samples of the incoming signal from all the elements of the array at a given instance of time. The data is in form of a complex vector. Then plotted the real and imaginary data

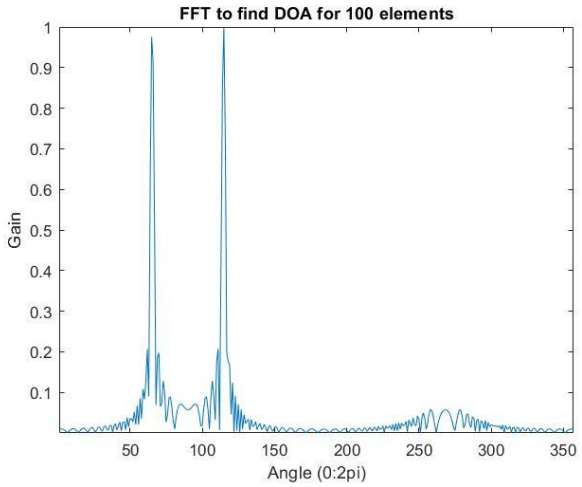




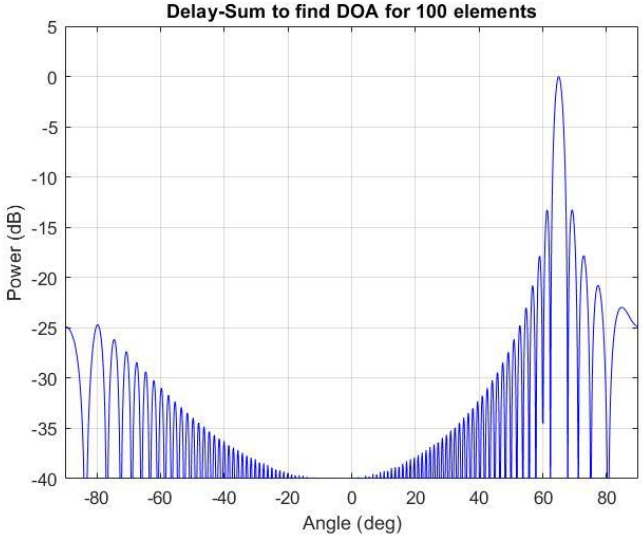
Gunjan K Khut
UID–115813083

2. Use the Fast Fourier Transform to find the direction of arrival (DoA) of the signal from the array snapshot.
   DoA was 65 degrees.





3. Used the Delay-Sum method to find the direction of arrival (DoA) of the signal from the array snapshot. **DoA was 64.69 degrees.**



Gunjan K Khut
UID–115813083

MATLAB CODE:

Task 1:

```
clear all;
close all;
clc;

%% Load the audio file
[xt, fst] = audioread('transmitSignal.wav');
[xr, fsr] = audioread('3.wav');
%% Play the sound
% wave contains the samples of the audio signal and fs is the sampling
frequency.
%soundsc(xt,fst); %play to check the correct fs
%soundsc(xr,fsr); %play to check the correct fs

%% Plotting the amplitude vs time plot
figure(1);
t=0:1/fst:(length(xt)-1)/fst;
subplot (2,1,1);
plot(t, xt); title('Transmitted Signal');axis tight;
xlabel('Time in Seconds'); ylabel('Amplitude');
% axis([0 1.8545 -1 1])

t=0:1/fsr:(length(xr)-1)/fsr;
subplot (2,1,2);
plot(t, xr); title('Received Signal');axis tight;
xlabel('Time in Seconds'); ylabel('Amplitude');
% axis([0 9.27764 -1 1])

%% Cross correlation:
% Where tx is one data set and rx is the other.
% The xcorr function will return the correlation and the index for each
% correlation as lag.
% With that you can find the index of the max value for the correlation and
% then use that index to look up the lag.
% This will be the delay in samples, within the precision of your sample
rate:

[corr,lag] = xcorr(xr, xt);
figure(2);
plot(lag/fsr,corr);axis tight;
xlabel('Time in Seconds'); ylabel('Amplitude')
title('Cross-correlation between Transmitted signal & Received Signal')

%% Distance Calculation
[~,I] = max(abs(corr));
sample_difference = lag(I);
time_difference = sample_difference/fsr;
distance = time_difference*340;
fprintf ('distance in meters = %d',distance);
```

Gunjan K Khut
UID–115813083

Task 2

```
clc;
clear all;
close all;

%% Defining the Parameters
ft = 2000; % Target frequency = 2 kHz
c = 340;   % Speed of sound = 340m/s
lambda = c/ft;  % Wavelength
no_of_elements = 10;
theta = -pi:pi/180:pi;

%% Inter-element distance is half of the wavelength (lambda/2)

Inter_dist_a = lambda/2;
r1 = zeros(1,length(theta));

for m = 1:no_of_elements
  dx(m,:) = (m-1)*Inter_dist_a*sin(theta);
  r1 = r1+exp(1i*2*pi*(dx(m,:)/lambda));
end

% Plot the directional gain of the ULAs
figure(3);
polar(theta,abs(r1))
title ('Gain of a Uniform Linear Array for d = lambda/2')

%% Inter-element distance is Twice of the wavelength (2*lambda)

Inter_dist_b = 2*lambda;
r2 = zeros(1,length(theta));

for m = 1:no_of_elements
  dx_new(m,:) = (m-1)*Inter_dist_b*sin(theta);
  r2 = r2+exp(-1i*2*pi*(dx_new(m,:)/lambda));
end

% Plot the directional gain of the ULAs
figure(4)
polar(theta,abs(r2))
title ('Gain of a Uniform Linear Array for d = 2*lambda')
```

Bonus Task

```
clc;
clear all;
close all;

%% Defining the Parameters
```

```
ft = 2000; % Target frequency = 2 kHz
c = 340;   % Speed of sound = 340m/s
lambda = c/ft;  % Wavelength
no_of_elements = 10;
theta = -pi:pi/180:pi;

%% Inter-element distance is (2*lambda) between 1 & 5 and (lambda/2) for the
rest.
d1=lambda/2;
d2=2*lambda;
n=10;
r1=zeros(1,length(theta));

for m = 1:n
  dx(m,:)=(m-1)*d1*sin(theta);
  dx(2,1:361) = 0;
  dx(3,1:361) = 0;
  dx(4,1:361) = 0;
  r1 = r1+exp(1i*2*pi*(dx(m,:)/lambda));
end

% Plot the directional gain of the ULAs
figure(1);
polar(theta,abs(r1))
title ('Gain of a ULA with faulty sensors')
```

Task 3

```
clear all;
close all;
clc;

%% Read the '3.csv' file

data = csvread('3.csv');
real_data = real(data); %read real data
imag_data = imag(data); %read imaginary data
t = linspace(0,100,100);

%% Plot the real and imaginary components of the array snapshot
figure(1);
subplot (2,1,1);
plot(t,real_data);  axis tight;
xlabel('Element Index'); ylabel('Amplitude');
set(gca,'YDir','normal')
title('Real components of the array snapshot');
subplot (2,1,2);
plot(t,imag_data), axis tight;
xlabel('Element Index'); ylabel('Amplitude');
set(gca,'YDir','normal')
title('Imaginary components of the array snapshot');
```

```
%% (b) Using FFT to find the direction of arrival (DoA) of the signal from
the array snapshot.

f = 2000;
c = 340;
lambda = c/f;      % Incoming Signal Wavelength in (m).
M = data; % Number of Array Elements.
d = lambda/2;    % Element Spacing.
No_of_elements = 100;
n = (1:No_of_elements).'; %ULA
%change the phi as per the source angle
% phi = 46; %source angle
%
% figure(2);
% K= 100;                           %fft length
% S1= abs(fftshift(fft(M,K))).^2;   %beamformed spectrum
% plot(S1/max(S1));                 %plot normalized spectrum
% xlabel('Array index'); ylabel('Gain');
% title('Direction of Arrival');


fft_data=fft(data);
figure(3);
plot(abs(fft_data));
title('FFT plot of the Data')

o = ((46*lambda)/(100*d));
Phi_fft = asind(o);
% figure(10);
% plot(Phi_fft);                %plot normalized spectrum
% xlabel('Array index'); ylabel('Gain');
% title('Direction of Arrival');

% z = exp(1j*2*180*d*K*sin(Phi_fft)/lambda); %steering vector
% figure(11);
% plot(z);


%% Array Pattern
% Define the angle grid.
phi_search = -pi:1.01*pi/180:pi;  %DOAs to search
% phi_search = -pi/2:1*pi/180:pi/2;  %DOAs to search
S2 = zeros(1,length(phi_search)); %preallocate array pattern
for i = 1:length(phi_search)
    v = exp(1j*2*pi*d*n*sin(phi_search(i))/lambda); %steering vector
    S2(i) = abs(M*v);                %spectrum
end
figure(3);
plot(S2/max(S2)); axis tight;         %plot normalized spectrum
xlabel('Angle (0:2pi)')
ylabel('Gain')
```

```
set(gca,'YDir','normal')
title('FFT to find DOA for 100 elements');

%% Use the Delay-Sum method to find the direction of arrival (DoA) of the
signal from the array snapshot.

phi_taskb = 65;
N_fft = 1024;
C1 = zeros(length(No_of_elements),N_fft);
% Define the angle grid.
angle = -90:180/N_fft:90-1/N_fft;  % Create N_fft angle samples between -90
to 90 deg.

for m=1:length(No_of_elements)
    u_s = (d/lambda)*sin(phi_taskb*pi/180);
    c_mf = exp(-1i*2*pi*u_s*(0:No_of_elements(m)-
1).')/sqrt(No_of_elements(m));
    for k=1:N_fft
        u = (d/lambda)*sin(angle(k)*pi/180);
        v = exp(-1i*2*pi*u*(0:No_of_elements(m)-
1)')/sqrt(No_of_elements(m)); % Azimuth Scanning Steering Vector.
        C1(m,k)= c_mf'*v;
    end
end

figure(4);
plot(angle,10*log10(abs(C1(1,:)).^2),'b');
title('Delay-Sum to find DOA for 100 elements');
xlabel('Angle (deg)'); ylabel('Power (dB)');
xlim([-90 90]); ylim([-40 5]);
grid on;
```