HW#3 Simulations

WIRELESS COMMUNICATIONS

Fateme Noorzad | 810198271 | spring 99

Question 7:

PART A:

This part is done with 10^4, 10^5 and 10^6 bits.

For any number of bits, using 'randi' command, 0 and 1 are generated. These bits create a n*4 matrix, where each row represents a symbol block. Then each row is changed to decimal. Its decimal value represents what coded symbol should be sent.

Then these coded symbols are modulated. For QPSK modulation as was said in problem, $e^{jn\pi/2}$ is used where n belongs to {0, 1, 2, 3} set.

Then 2 channels should be considered. In scenario one all symbols go through a fading channel, which has Rayleigh fading and an AWGN as its noise. In scenario two all symbols go through another channel which has no fading, has a channel gain of one and an AWGN as its noise.

Scenario 1: Fading Channel

For simulating the fading channel, 'normrnd' is used. h is a complex normal random variable. Its mean is 0 and its variance is 1. So its real and imaginary part are normal random variable with 0 as their mean and ½ as their variance. All created symbols go through this fading channel. For simulating it, these symbols are multiplied to h.

Then symbols which went through fading channel, go through an AWGN channel. For simulating this channel, since we want to plot P_e with respect to SNR, different SNRs should be considered. For this goal, AWGN noise is created with different variance representing different SNR values. This noise is also a complex normal random variable with 1/SNR variance and 0 as its mean. (We supposed symbols' energy is one) So noise has a real and imaginary part, each has 0 mean and 1/2SNR variance. This noise is added to symbols.

Now it's time to detect the received symbols. For detecting we use maximum likelihood. If we represent received symbols with Y, the reference symbols with X (give code words in question.) and the fading channel with h, then we have:

$$X_{ML} = \min \left\{ \left| |Y - Xh| \right|^{٢} \right\}$$

Scenario 2: AWGN Channel:

Then symbols go through an AWGN channel. For simulating this channel, since we want to plot P_e with respect to SNR, different SNRs should be considered. For this goal, AWGN noise is created with different variance representing different SNR values. This noise is also a complex normal random variable with 1/SNR variance and 0 as its mean. (We supposed symbols' energy is one) So noise has a real and imaginary part, each has 0 mean and 1/2SNR variance. This noise is added to symbols. In this scenario channel is considered to be 1.

Now it's time to detect the received symbols. For detecting we use maximum likelihood. If we represent received symbols with Y, the reference symbols with X (give code words in question.), then we have:

$$X_{ML} = \min\left\{ \left|\left|Y - X\right|\right|^{\text{۲}} \right\}$$

After detecting symbols in both scenarios, probability of error should be found. So the detected symbol matrix is subtracted from the sent symbols and the number of none zero indexes in this matrix is considered to be the errors. The fraction of these bits to all bits gives the probability of error.

Below plots show P_e with respect to SNR for different number of bits.

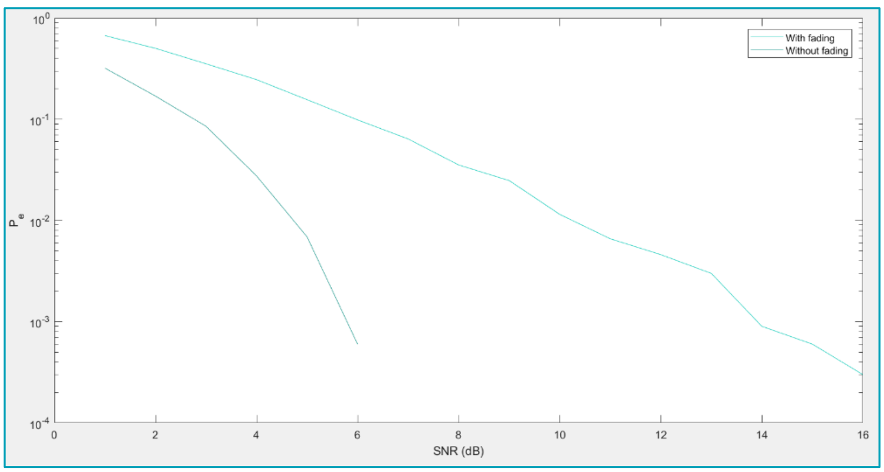


Figure 1. Channel performance curve for 10^4 bits

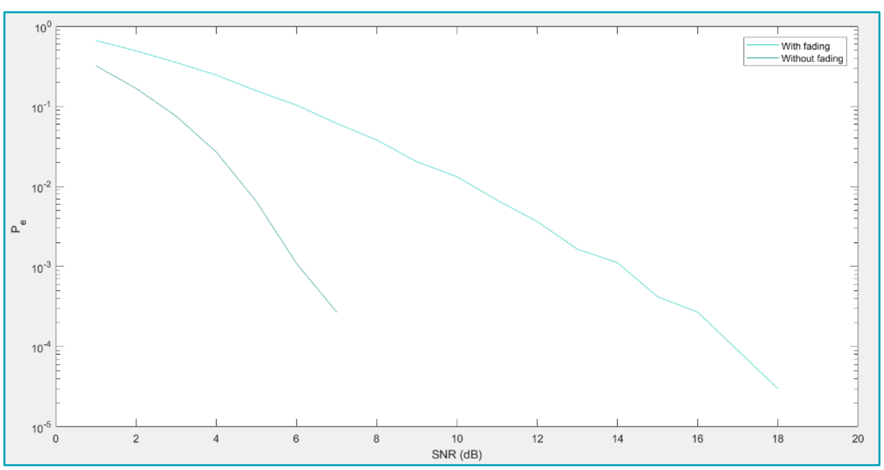


Figure 2. Channel performance curve for 10^5 bits

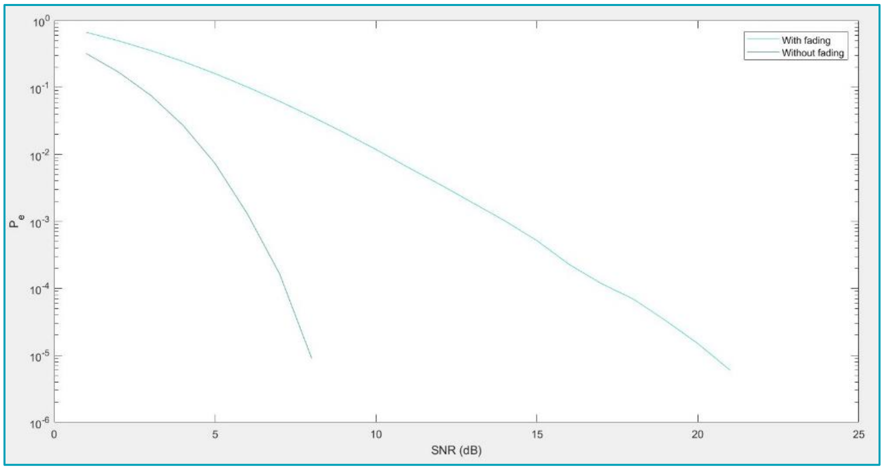


Figure 3. Channel performance curve for 10^6 bits

PART B:

In this part diversity order should be found in 2 different methods.

System Analysis:

First the given bits are changed into a 4*4 matrix where they lay on the diagonal of the matrix. Each matrix represents a code word. Then rank of all 2 different code words are found. Diversity order is the minimum rank which for this problem equals 3.

Performance Curve:

The slope of this curve in high SNRs represent diversity order. In this problem it is found 3 as well. Sometimes due to MATLAB's round ups, this slope is 4 which happens in a few cases.

Question 8:

PART A:

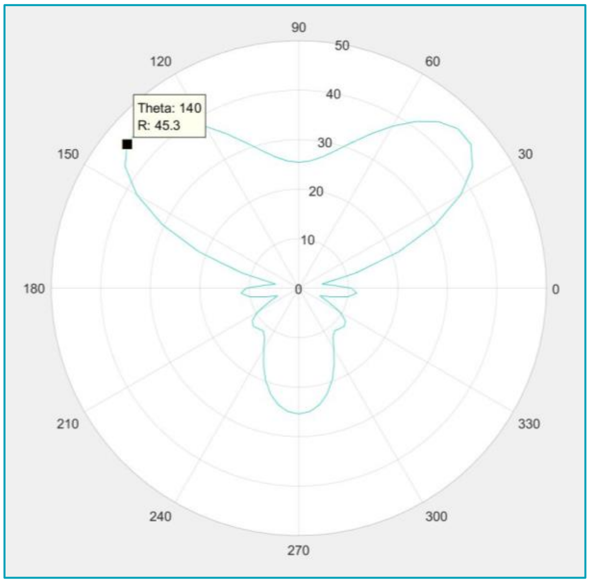
As was discussed in lectures for a ray coming from θ degrees as shown in the plot below, for MRC diversity, weight of an antenna is calculated with respect to its distance from the first antenna. So for the i^{th} antenna we have:

$$w_i = \sum_{j=١}^{n} |h_j| \exp(-j2\pi(i-١)\sin(\theta_j)\frac{d}{\lambda})$$

In this question d is $\frac{\lambda}{٢}$ and $|h_i|$ is assumed to be the given SNRs. Also n is 3 and θ_j is 55, *100* and *140* degrees.

With all that in mind weight of each antenna is calculated. Then all weights are used to calculate array function in different angles.

As been said, all antennas are omnidirectional, so it's supposed that $A(\theta) = 1$ for any θ from $[0,2\pi]$. With considering the delays between rays coming from each angel, the resulting pattern is:



As can be seen the main lobe happens near *140 degrees* which had the largest SNR. For other angles, the pattern is pretty much like we expected.

For finding the total SNR, since noise has the same distribution in all three rays, signals' energy is found using SNR of all three rays.

$$E_{s1} = SNR_i * N_{oi} = SNR_i$$

Formula for finding SNR total is:

$$SNR_T = \frac{E_{s1}\left|\sum_{i=١}^{٤} w_i e^{j\varphi_{١}{}_i}\right|^{٢} + E_{s2}\left|\sum_{i=١}^{٤} w_i e^{j\varphi_{٢}{}_i}\right|^{٢} + E_{s3}\left|\sum_{i=١}^{٤} w_i e^{j\varphi_{٣}{}_i}\right|^{٢}}{\sum_{i=١}^{٤} |w_i|^{٢}}$$

Where: $\varphi_{١}{}_i = \pi(i-١)\sin(١٤٠)\ .\ \varphi_{٢}{}_i = \pi(i-١)\sin(١٠٠).\ \varphi_{٣}{}_i = \pi(i-١)\sin(٥٥)$

```
The total SNR is: 25.851738 (dB)
```

PART B:

In this part since we want to maximize pattern in the angel of desired signal and have nulls in other angels, the below linear equations should be solved.

$$w_{١} + w_{٢}e^{jsin(٤٥)} + w_{٣}e^{j2sin(٤٥)} + w_{٤}e^{j3sin(٤٥)} = ١$$

$$w_{١} + w_{٢}e^{jsin(0)} + w_{٣}e^{j2sin(0)} + w_{٤}e^{j3sin(0)} = 0$$

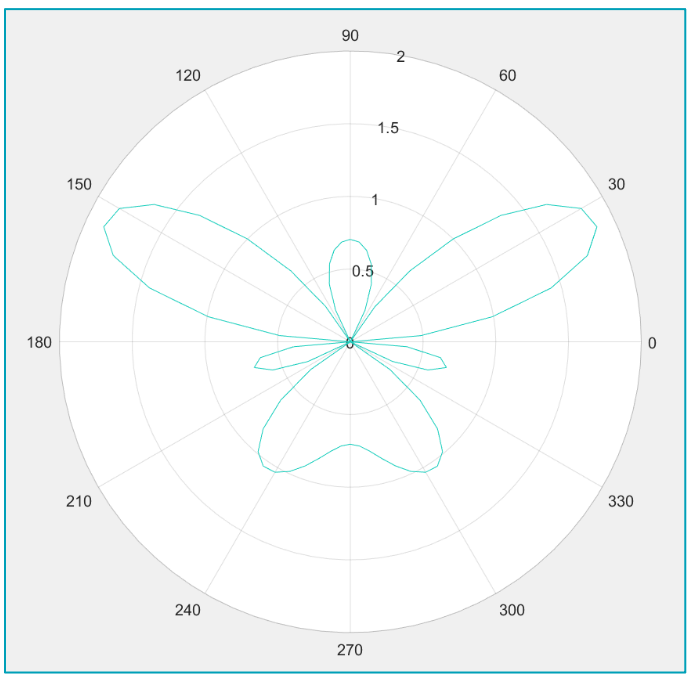
$$w_{١} + w_{٢}e^{jsin(٣٣٠)} + w_{٣}e^{j2sin(٣٣٠)} + w_{٤}e^{j3sin(٣٣٠)} = 0$$

$$w_{١} + w_{٢}e^{jsin(١٢٠)} + w_{٣}e^{j2sin(١٢٠)} + w_{٤}e^{j3sin(١٢٠)} = 0$$

So in MATLAB the below matrixes are created to form a linear equation:

$$\begin{bmatrix} ١ & e^{jsin(٤٥)} & e^{j2sin(٤٥)} & e^{j3sin(٤٥)} \\ ١ & e^{jsin(0)} & e^{j2sin(0)} & e^{j3sin(0)} \\ ١ & e^{jsin(٣٣٠)} & e^{j2sin(٣٣٠)} & e^{j3sin(٣٣٠)} \\ ١ & e^{jsin(١٢٠)} & e^{j2sin(١٢٠)} & e^{j3sin(١٢٠)} \end{bmatrix} \begin{bmatrix} w_{١} \\ w_{٢} \\ w_{٣} \\ w_{٤} \end{bmatrix} = \begin{bmatrix} ١ \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Then using 'linsolve' these weights are calculated. Same as last part, using these weights, antenna pattern is created. The resulting patter can be seen in the next page.



In this part we wanted to create nulls in *0*, *120* and *330 degrees*. As can be seen in the above plot, the desired goal is achieved. But because of loss and the interference, the maximum value didn't happen exactly in *45 degrees*.

For finding the total SNR, the formula mentioned in the last part is used.

```
The total SNR is: 20.158285 (dB)
```

As can be seen SNR is near the SNR value of the desired signal.

PART C:

In this part we want to maximize SINR. SINR is the signal to interference and noise ratio. As was explained is problem, the interference signal comes in the direction of *-60 degrees (150 degrees)*, and the desired signal comes in direction of *90 degrees (0 degrees)*. So SINR is:

$$SINR = \frac{E_S |\sum_{i=١}^{٤} w_i e^{j\varphi_{١\,i}}|^{٢}}{E_I |\sum_{i=١}^{٤} w_i e^{j\varphi_{٢\,i}}|^{٢} + \sum_{i=١}^{٤} |w_i|^{٢}}$$

Where: $\varphi_{١\,i} = \pi(i-١)\sin(o) = o\,.\,\varphi_{٢\,i} = \pi(i-١)\sin(١٥٠)$

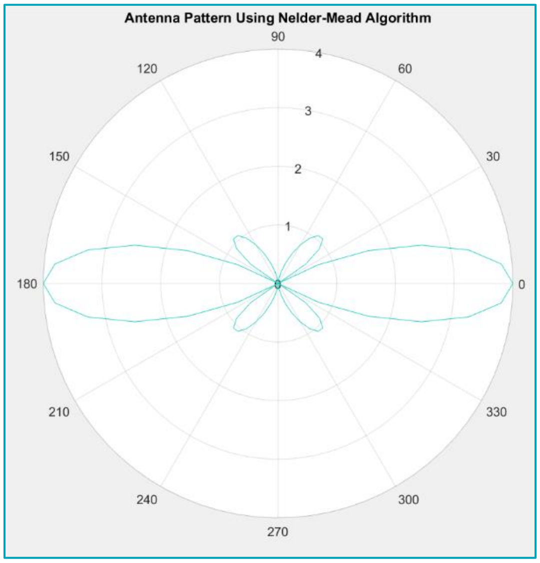
For finding weights 2 different optimization algorithms are used.

Nelder-Mead Algorithm:

For implementing this algorithm, ‘fminsearch’ command in matlab is used. Since this function finds minimum value of a function, 1/SINR is given to this command. In addition to SINR function, we should set a starting point for this command’s input. This starting point has a contribution on the amplitude of the lobes, and on SINR value. As it’s obvious, if antennas’ weights be the same or very close to each other, the resulting pattern will have a bigger main lobe and smaller side lobes resulting in better SINR. So it’s better to choose the starting point for weights close to each other and bigger as possible.

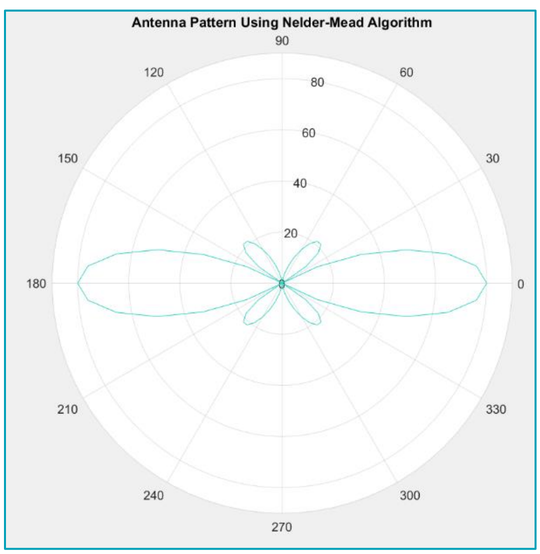
Below plots are for 3 different stating points and shows their SNRs as well.

- Starting point = [1 1 1 1]



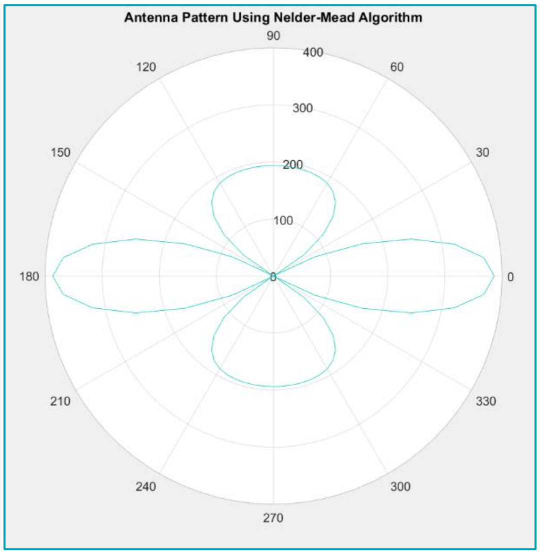

```
The total SINR using Nelder-Mead algorithm is: 26.020600 (dB)
```

- Starting point = [20 20 20 20]




```
The total SINR using Nelder-Mead algorithm is: 26.020600 (dB)
```

- Starting point = [0.1 10 -20 -0.01]



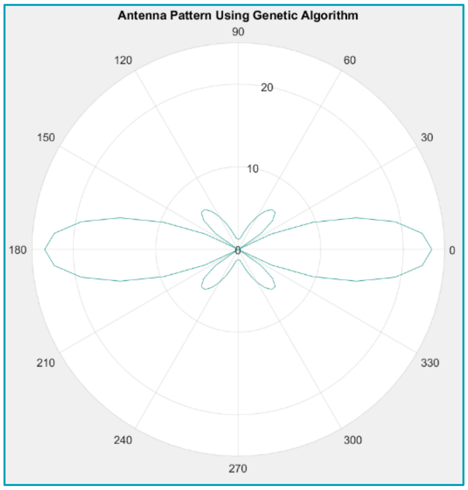

```
The total SINR using Nelder-Mead algorithm is: 25.041993 (dB)
```

As can be seen when all antennas have the same weights, the starting point has no contribution on SINR's value (as in the first and second scenario SINR is the same) but has a huge contribution on the amplitude of the main and side lobes. So based on the desired output, the starting point should be set. This difference happens because in this algorithm the local minimum is found.

In the third plot, side lobes have bigger amplitudes than the first and second plot and as a result SNR is less than the other 2. We expected to see this result. Because when antennas have different weights, the angels where side lobes happen don't cancel each other completely as they did with the same weights.

Genetic Algorithm:

For solving this problem using this algorithm, 'ga' command is used. Again 'ga' finds the minimum value of the goal function, so 1/SINR is given to it. The resulting pattern and SINR can be seen in the next page. Using this algorithm also gives a smaller SINR than the last algorithm.



The total SINR using genetic algorithm is: 25.996873 (dB)