CT Assignment-6 report

U.S.S.Sasanka

2019102036

Important points to note:

- All the questions have been answered.
- Plots are attached wherever asked. Theory parts and explanations have been given when required.
- Variance calculations from the plots have also been added.
- In matlab scripts:
  CLEAR before running

  scriptA has the code for Part-I

  For part-II the name is script<question number>

Part-I

We have to observe the effect of noise on digital communication system.

Consider two orthogonal signals

$(r0, r1) = (\sqrt{E} + n0, n1)$ (or) $(r0, r1) = (n0, \sqrt{E} + n1)$ where n0 and n1 are two **iid normal** random variables. E=1 for easy calculations.

We have used **Monte Carlo simulations** to generate 100 samples of (r0,r1)

The MATLAB code snippet is

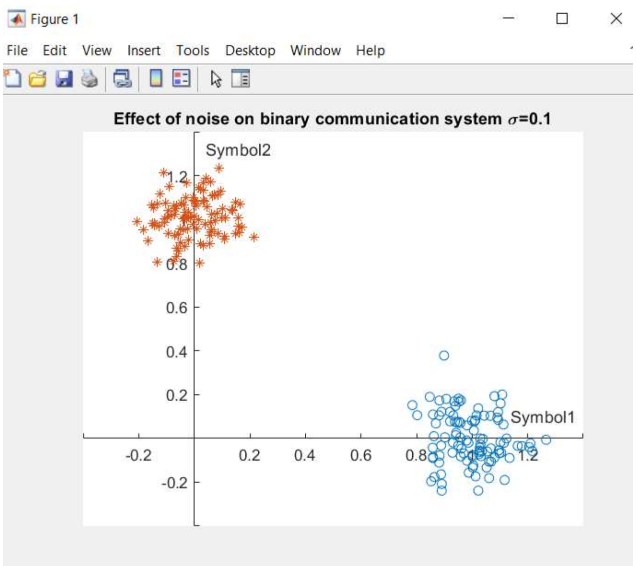
```
n0=sigma.*randn(n,1);
n1=sigma.*randn(n,1);
```

Parameters:

n=100;

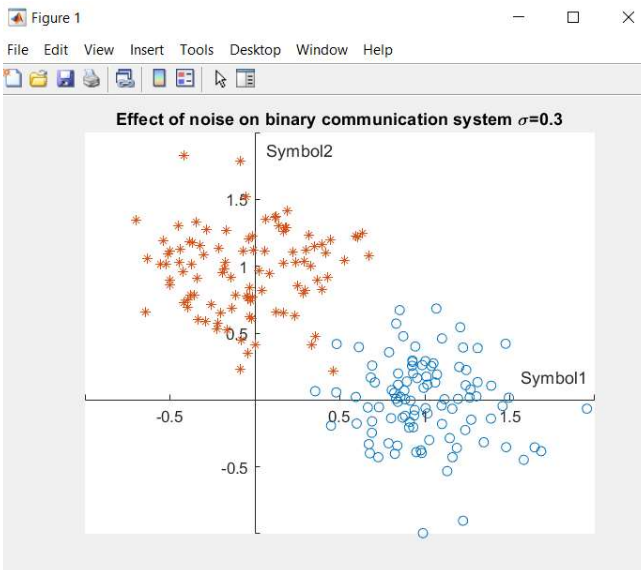
Plots:

For σ=0.1



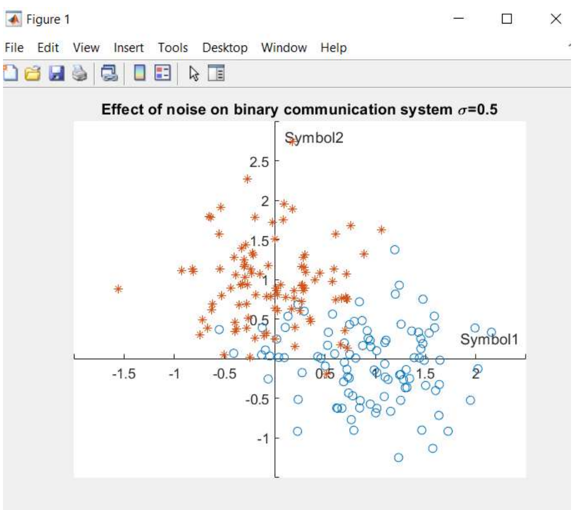
Blue circle(around x-axis)- Symbol1, Red cross(around y-axis)-Symbol2

For σ=0.3



Blue circle(around x-axis)- Symbol1, Red cross(around y-axis)-Symbol2

For σ=0.5



Blue circle(around x-axis)- Symbol1, Red cross(around y-axis)-Symbol2

A1.) We can use **binary orthogonal detector** following **MAP rule**

A2.) As we increase variance, the signal to noise ration decreases. This means that the strength of the signal decreases and noise strength increases.

For less variance, the graph is dense whereas for higher values, the graph is less dense and there is a distortion.

Clearly this is observed in the above plots where we get more distorted plots due to the more pronounced effect of noise caused by the increase in variance.

This might result in a situation where correct detection has less probability and **probability of error**(wrong detection occuring) **increases as variance increases**.

Part-II

Question-1)

We have to generate ‘randbit’ function that gives 0 and 1 with equal probability.

We can do this by using the inbuilt function `randi([0 1],1,n);`
This function generates random bits 0, 1 with equal probability but it is important to note that equi-probable does not mean equal number of 0s and 1s.

Question-2)

The given bpsk, qpsk, 8psk, 4pam, 16qam have been implemented using gray code.

Gray code: It is a code in which two consecutive numbers(nearest neighbours) differ by exactly one bit.

BPSK:

0→-1

1→ 1

QPSK:

00→ -1-i

01→-1+i

10→1+i

11→1-i

4-PAM:
00→ -3

01→-1

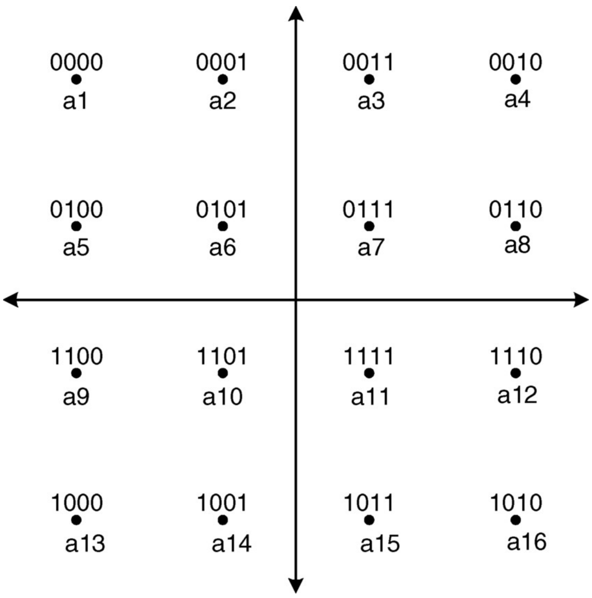
10→3

11→1

8-PSK:

Modulator Input	Modulator Output
000	exp(0)
001	exp($j\pi/4$)
010	exp($j3\pi/4$)
011	exp($j\pi/2$) = exp($j2\pi/4$)
100	exp($j7\pi/4$)
101	exp($j3\pi/2$) = exp($j6\pi/4$)
110	exp($j\pi$) = exp($j4\pi/4$)
111	exp($j5\pi/4$)

16-QAM:



This idea is used for 16QAM

Question-3)

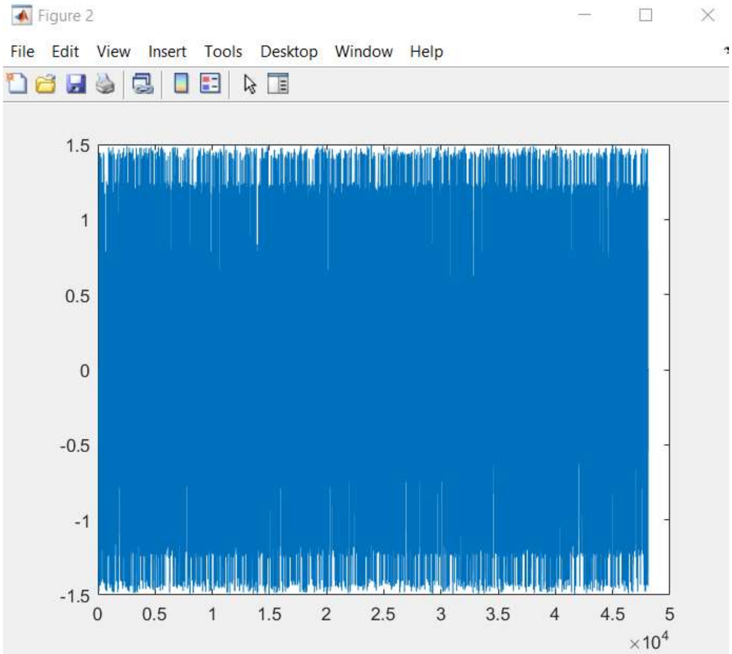
We need to generate 12000 0/1 bit and map them to BPSK symbols. Then, we pass these symbols to 4.B.2 reference code and make some changes in it. The output of the code is noiseless samples at rate 4/T. raised cosine function has also been taken from the text book.

In this part of the question we have taken the channel to be noiseless. So we just the transmit and receive filters are required.

Note that we have also upscaled the sampling frequency by a factor of 4. Excess bandwidth is taken to be 0.5 for convenience.

Plot(after the receive filter without noise):

This is the plot of rx_output vs samples



Clearly we notice that the graph is without any distortions due to the fact that we did not add any noise.

Question-4)

We have to do the same process as mentioned in the previous question but this time we have noise.

To the previous code, we have added the following code snippet so that we added noise to the ideal noiseless samples.

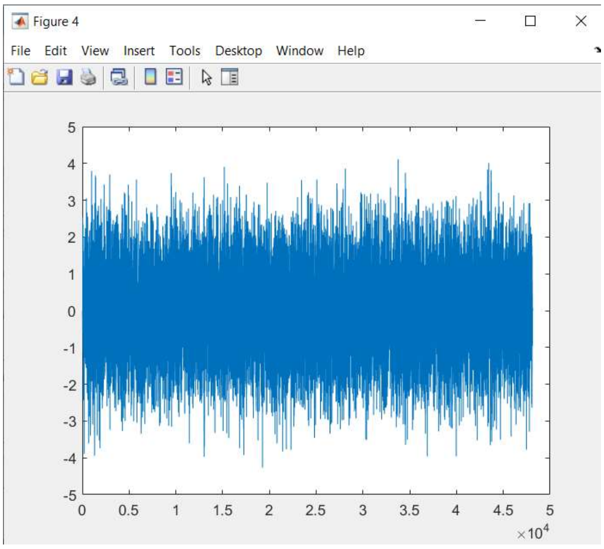
```
awgn1=normrnd(0, sqrt(variance), [size(tx_output)]);
awgn2=normrnd(0, sqrt(variance), [size(tx_output)]);
```

The variance value used is discussed in question 6).

We add awgn1+iawgn2. The noise is generated through a normrnd function which is inbuilt in matlab and takes arguments as mean, standard deviation and length required.

Plot(after the receive filter after adding noise):

This is the plot of real(rx_output) vs samples



Note that we have plotted only real part so that we can compare it with the previous signal and thus understand that noise has been added.

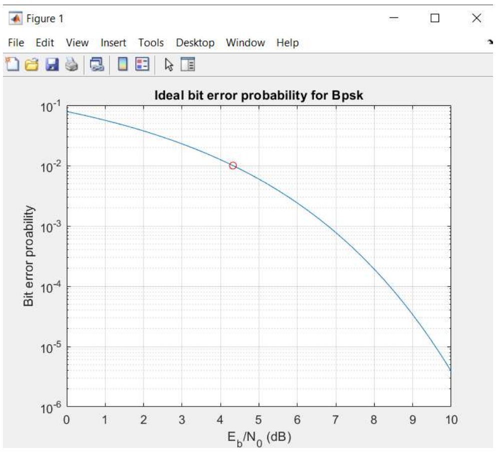
Question-5)

We have to plot Bit error Probability(for BPSK) vs Eb/N0 which are in log scale.We have used the inbuilt function efrc instead of the familiar function Q.

$$Q(x) = \frac{1}{2}\left(\frac{2}{\sqrt{\pi}}\int_{x/\sqrt{2}}^{\infty}\exp(-t^2)\,dt\right)$$
$$= \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \text{ -or-}$$
$$= \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right).$$

This relation is an extremely useful relation which helps us to determine the value of Q(sqrt(2Eb/N0)).

Plot(Ideal bit error probability for BPSK):



x-axis: Eb/N0   y-axis: BEP

The value at which the probability is 10^-2 is **Eb/N0=4.3232dB** which is marked as circle.

Question-6)

Eb/N0=4.3232 in dB

In normal scale, **Eb/N0=$10^{(4.32/10)}$=2.7040(Normal scale).**

We are dealing with BPSK. Es=1/2((-1)^2 +1^2)=1. **Eb=Es/$\log_2 2$=1**

- ⇨ (1)/N0=2.7040

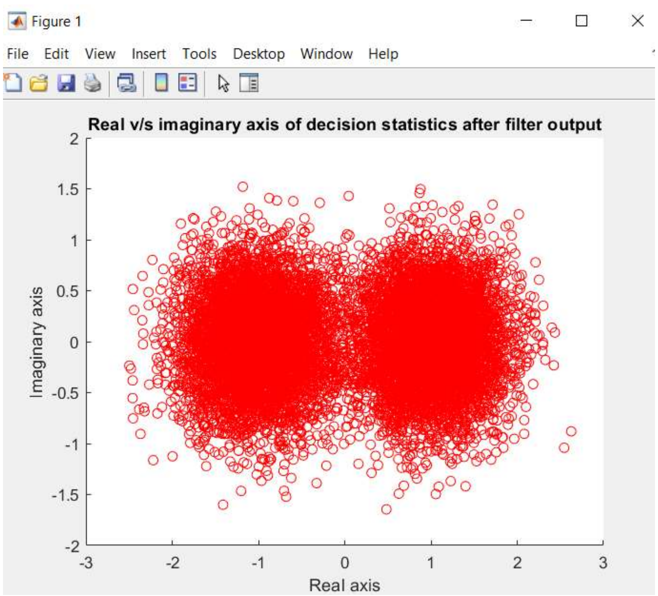
But we know variance=N0/2

- ⇨ **Variance=1/(2*2.704)=0.1849**
- ⇨ **Standard deviation=0.4300**

This standard deviation value is used in the previous question4 and will also be used for few other upcoming questions.

In addition to these calculations, we have to plot the constellations with that variance value.

Plot(of decision statistics of BPSK):



x-axis: Real axis

y-axis: Imaginary axis

Clearly, we find that this is a noisy version of BPSK constellation.

Question-7)

Parameter values are same as before question because we continue with BPSK

Decision rule for BPSK:

If(Array(i))>=0 Map to 1

If(Array(i))<0 Map to -1.

We do this for both the receiver input and receiver output.

Bit error probability of BPSK:

The bit error probability at receiver input is **~ 0.1007**

The bit error probability at receiver output is **~0.0225**

The ideal bit error probability is **0.01**

```
    0.1003

    0.0225

>> script7
    0.0994

    0.0225

>> script7
    0.1007

    0.0213
```

Therefore we can say that the bit error probability at input is **significantly worse** and the probability of error at receiver output is **little worse** than ideal value.

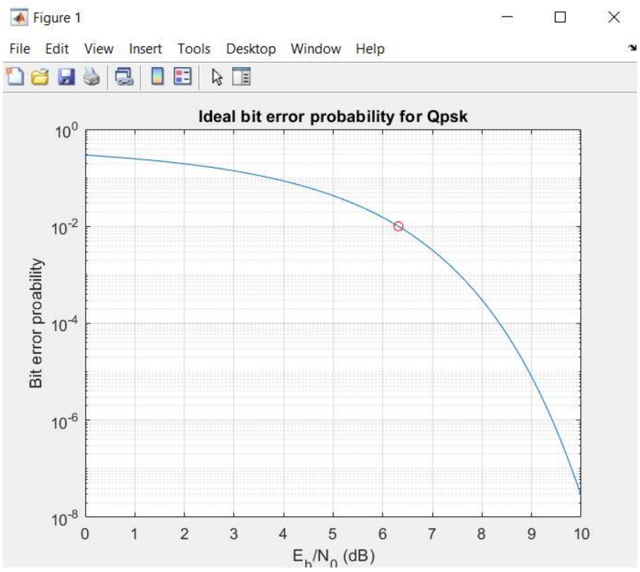
Question-8)

We have initially generated the required 4-PAM array.

Then, we followed the same procedure in question-5 to obtain bit error probability graph.

Here, the given expression is $Q(eE_b/5N_0)$. So equivalently in erfc, the expression is $0.5*erfc(eE_b/5*sqrt(2)*N_0)$.

Plot(Ideal bit error probability for QPSK):



x-axis: Eb/N0   y-axis: BEP

The value at which the probability is 10^-2 is **Eb/N0=6.3135dB** which is marked as circle.

Question-9)

Eb/N0=6.3135 in dB

In normal scale, **Eb/N0=$10^{(6.3135/10)}$=4.279(Normal scale).**

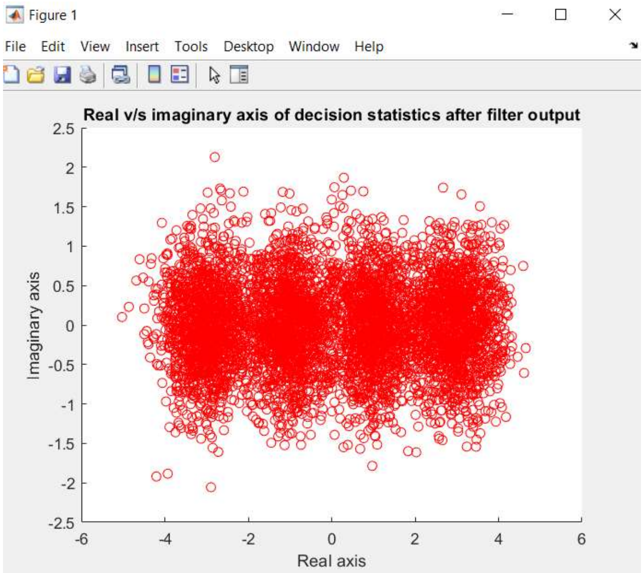
We are dealing with 4-PAM. Es=1/4(2*1^2+2*3^2)=5. **Eb=Es/$\log_2$4=5/2**

- ⇨ (5/2)/N0=4.279

But we know variance=N0/2

- ⇨ **Variance=5/(4*4.279)=0.292**
- ⇨ **Standard deviation=0.54**

Plot(Decision Statistics of 4-PAM):



x-axis: Real axis

y-axis: Imaginary axis

Decision rule for 4-PAM:

If(array(i)>=2) Map to 3

If(0<=array(i)<2) Map to 1

If(-2<=array(i)<0) Map to -1

If(array(i)<-2) Map to -3

Bit error probability(4-PAM):

```
>> script9a
    0.0482

>> script9a
    0.0490

>> script9a
    0.0488

>> script9a
    0.0458
```

The bit error probability is **~0.048.** The ideal value is 0.01. We notice that the probability of error **increased due to the ISI in the decision statistics**.

For QPSK:

We are given that the value of bit error probability is same for BPSK and QPSK.

Eb/N0=4.3232 in dB

In normal scale, **Eb/N0=$10^{(4.32/10)}$=2.7040(Normal scale)**.

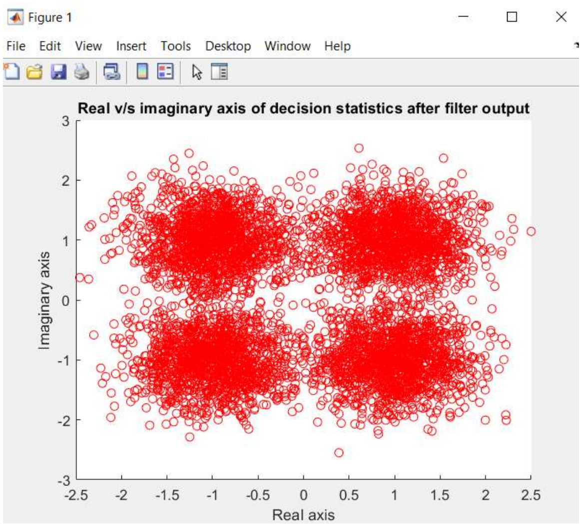
We are dealing with QPSK. Es=1/4(4*2)=2. **Eb=Es/$\log_2$4=1**

- ⇨ (1)/N0=2.7040

But we know variance=N0/2

- ⇨ **Variance=1/(2*2.704)=0.1849**
- ⇨ **Standard deviation=0.4300**

Plot(Decision Statistics of QPSK):



x-axis: Real axis

y-axis: Imaginary axis

Decision rule for QPSK:

```
if(real(arr(i))>=0 && imag(arr(i))>=0) Map to 1+i
if(real(arr(i))>=0 && imag(arr(i))<0) Map to 1-i
if(real(arr(i))<0 && imag(arr(i))<0) Map to -1-i
if(real(arr(i))<0 && imag(arr(i))>=0) Map to -1+i
```

Bit error probability(QPSK):

```
>> script9b
    0.0227

>> script9b
    0.0237

>> script9b
    0.0195
```

The bit error probability is **~0.022.** The ideal value is 0.01. We notice that the probability of error **increased due to the ISI in the decision statistics**.

Question-10)

Eb/N0=6.3135 in dB

In normal scale, **Eb/N0=$10^{(6.3135/10)}$=4.279(Normal scale)**.

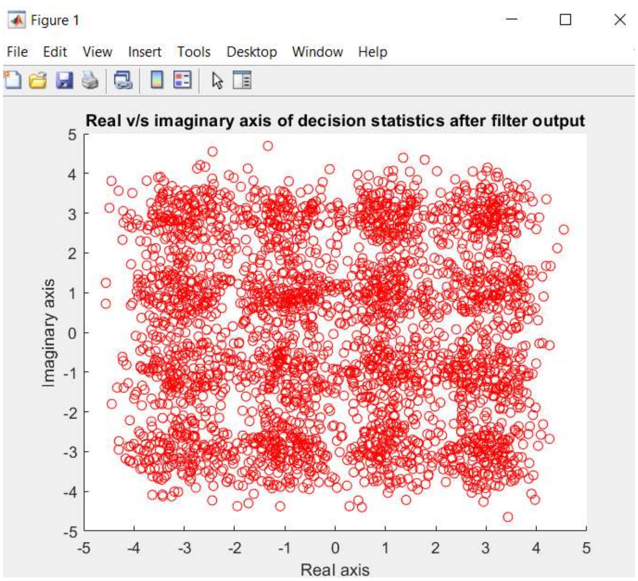
We are dealing with 16-QAM. Es=1/16(4*2+8*10+4*18)=10. **Eb=Es/$\log_2$16=5/2**

- ⇨ (5/2)/N0=4.279

But we know variance=N0/2

- ⇨ **Variance=5/(4*4.279)=0.292**
- ⇨ **Standard deviation=0.54**

Plot(Decision Statistics of 16-QAM):



x-axis: Real axis

y-axis: Imaginary axis

Decision rule for 16-QAM:

To write the entire decision rule is lengthy. The idea is just take the value in a particular box/ interval and map it to the original value.

Bit error probability(16-QAM):

```
    0.0893

>> clear
>> script10
    0.1043

>> script10
    0.0957

>> script10
    0.0907
```

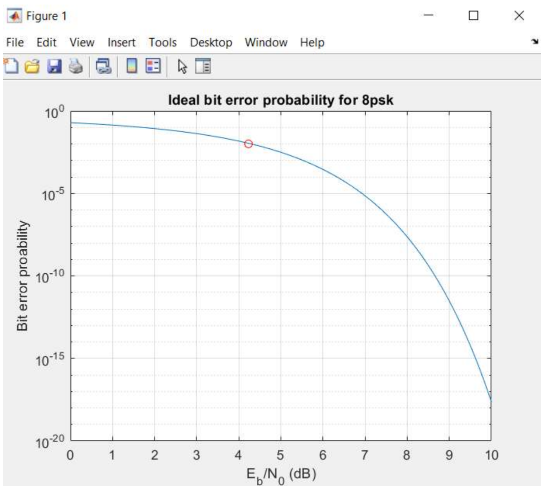
The bit error probability is **~0.0907.** The ideal value is 0.01. We notice that the probability of error **increased due to the ISI in the decision statistics**.

Question-11)

We have the given expression for ideal bit error for 8-psk. Then we used the equivalent erfc function which is as follows

```
1/2*erfc((( 6-3*sqrt(2))/2*sqrt(2))*Eb/No)
```

Plot(Ideal bit error probability 8-PSK):



x-axis: Eb/N0   y-axis: BEP

The value at which the probability is 10^-2 is **Eb/N0=4.2286dB** which is marked as circle.

Eb/N0=4.2286 in dB

In normal scale, Eb/N0=$10^{(4.2286/10)}$=**2.647(Normal scale)**.

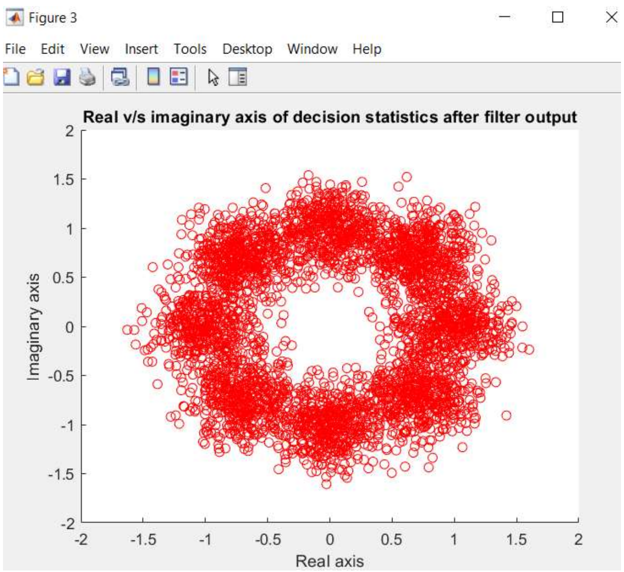
We are dealing with 8-PSK. Es=1/8(8*1)=1. **Eb=Es/$\log_2$8=1/3**

- ⇨ (1/3)/N0=2.647

But we know variance=N0/2

- ⇨ **Variance=1/(6*2.647)**

Plot(Decision Statistics of 8-PSK):



x-axis: Real axis

y-axis: Imaginary axis

Decision rule for 8-PSK:

To write the entire decision rule is lengthy. The idea is to divide the scatter plot based on angles into interval of pi/4 and map all the values in that particular interval to the corresponding number.

Bit error probability(8-PSK):

```
>> script11
    0.0770

>> script11
    0.0770

>> script11
    0.0762

>>
```

The bit error probability is **~0.077.** The ideal value is 0.01. We notice that the probability of error **increased due to the ISI in the decision statistics**.

Question-12)

The code is run and the plot is generated. The nearest neighbour approximation is a really good approximation as it is observed that the bit error probability found is nearly equal to the ideal bit error probability.