ASSIGNMENT 5 REPORT

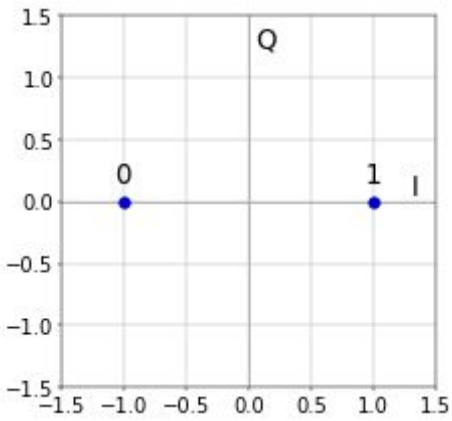
-JAYANT DUNEJA
2018102003

QUESTION-1:

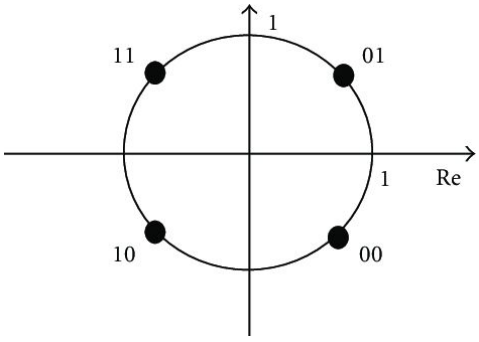
We generated the random bit by using the randsrc() function in Matlab. Both the probabilities were defined to be 0.5. This function returns a single bit and not a vector.

QUESTION-2:

1. **bpskmap():** This maps the given bits to -1,1. If the value of the bit is 0,then the value of the symbol is -1 and if the value of the bit is 1 then the value of the symbol is -1.I have written the given program using if statements nested inside a for loop.



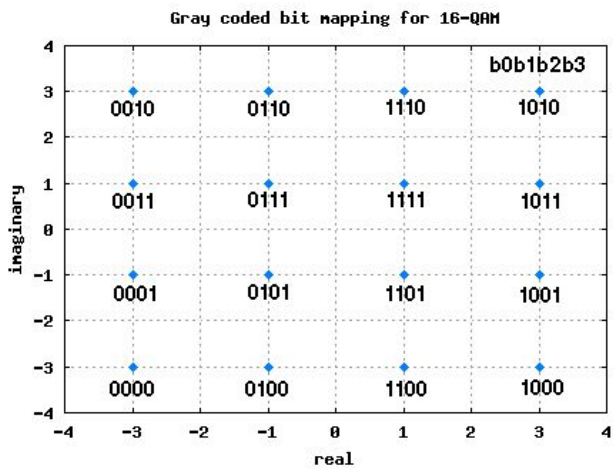
2. **qpskmap():** This function maps a sequence of 2 bits to 4 complex numbers. This was also implemented using the same technique,if statements nested in a for loop. This function takes its input as 2 vectors and the output is a single vector which contains the complex symbols.



3.fourpammap(): This function maps a sequence of 2 bits to 4 real values. This function takes 2 vectors as input and returns 1 output vector.The technique used is the same as described earlier.

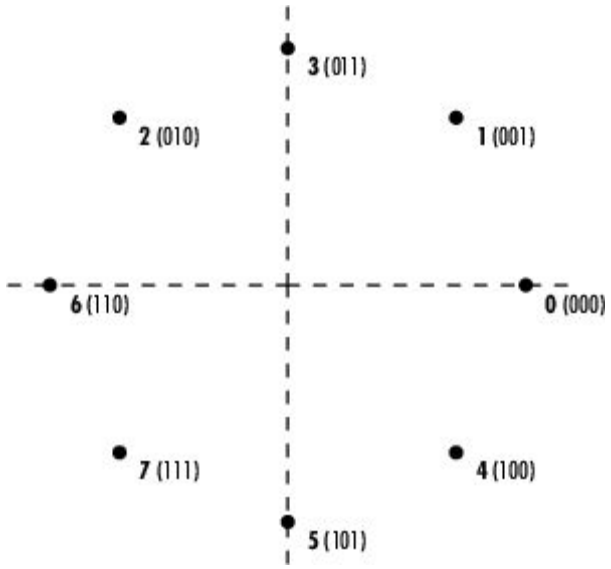
Bit Sequence	Value
00	-3
01	-1
11	1
10	3

4.sixteenqammap(): This function maps a sequence of 4 bits to 16 complex numbers. This was also implemented using the same technique,if statements nested in a for loop. This function takes its input as 4 vectors and the output is a single vector which contains the complex symbols.



5.eightpskmap(): This function maps a sequence of 3 bits to 8 complex numbers. This was also implemented using the same technique,if statements nested in a for loop. This function takes its input as 3 vectors and the output is a single vector which contains the complex symbols. The values taken are of the form

$$\exp(j*2*\pi*i/8) , i = 0, 1, ..., 7.$$



QUESTION-3:

- In this question firstly we define the Square Root Nyquist Pulse as our Transmitter and Receiver filter using the rcosdesign() inbuilt function in Matlab.
- Then 12000 random bits were generated and then these bits were mapped to 12000 symbols based on the qpsk mapping.
- These symbols were upsampled using the code snippet which was present in the textbook. Below is a paragraph which explains what upsampling is doing:
  The symbols come in every T seconds, while the samples of the transmit filter are spaced by T /m. For example,the nth symbol contributes b[n]p(t − nT) to the transmit filter output, and the (n + 1)st symbol contributes b[n + 1]p(t − (n + 1)T). Since p(t − nT) and p(t − (n + 1)T) are offset by T , they must be offset by m samples when sampling at a rate of m/T . Thus, if the symbols are input to
  a transmit filter whose discrete time impulse response is expressed at sampling rate m/T , then successive symbols at the input to the filter must be spaced by m samples. That is, in order to get the output as a convolution of the symbols with the transmit filter expressed at rate

m/T. We must insert m − 1 zeros between successive symbols to convert them to a sampling rate of m/T.

- After this the upsampled symbols are passed through the input transmitter filter and then passed through the receiver filter using the conv() function in Matlab.

QUESTION-4:

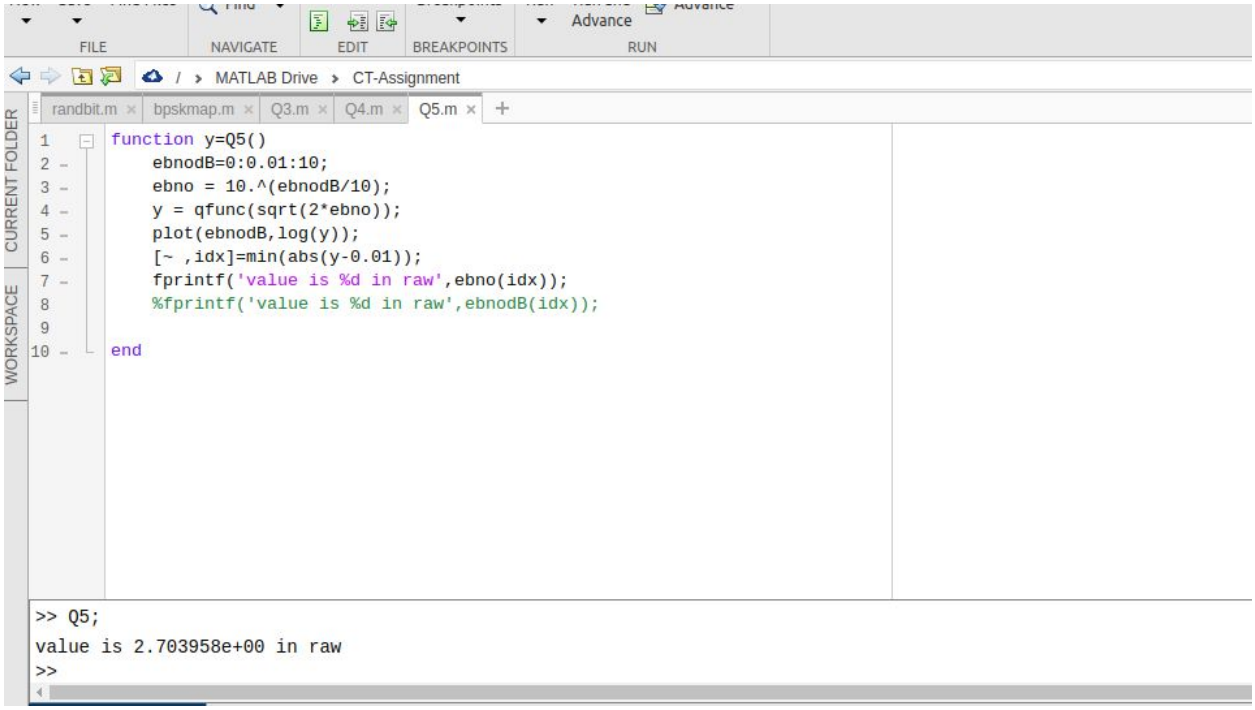
- In this question,the exact procedure of the preivous question is followed and the only difference is that we have added Noise in this question.
- The noise being added is AWGN gaussian noise with mean=0 and standard deviation=sigma(sigma is calculated in the next question).
- The noise added is complex and both the real and imaginary parts are gaussian random variables.
- We have used the Normrnd() inbuilt function of Matlab to generate the noise.
- These random variables were added to the signal generated after the transmitter filter and then the sum of the noise and the original symbols were passed through the receiver filter.
- I have attached a Code snippet of the process I just explained.

```
symbols_upsampled1(1:m:nsymbols_upsampled1)=symbols1;%insert symbols with spacing m

transmit_filter = rcosdesign(a,nsymbols_upsampled1,m,'sqrt');   %Generating square ro

%PASSING ORIGINAL SIGNAL  THROUGH THE  TRANSMITTER FILTER
tx_output=conv(symbols_upsampled1,transmit_filter);

%Gaussian noise generation
symbols= normrnd(0,sigma,size(tx_output)) + 1i*normrnd(0,sigma,size(tx_output));


temp1=symbols + tx_output;
y=conv(temp1,transmit_filter);
```

MMAND WINDOW

QUESTION-5:

- In this question we compute the ideal error probability for BPSK which is given by Q(sqrt(2*eb/no)).
- We have used the inbuilt qfunc to generate this value.
- Next we had to determine the value of Eb/No for which the value of probability was 0.01.
- As we were not sure whether the probability would take this given value, I minimised abs(prob - 0.01) using the min() function in Matlab and I got the value from there.A graph was plotted between log(prob) and Eb/No and the graph would be attached in the folder. Attached below is the code snippet and the output I got from this program.



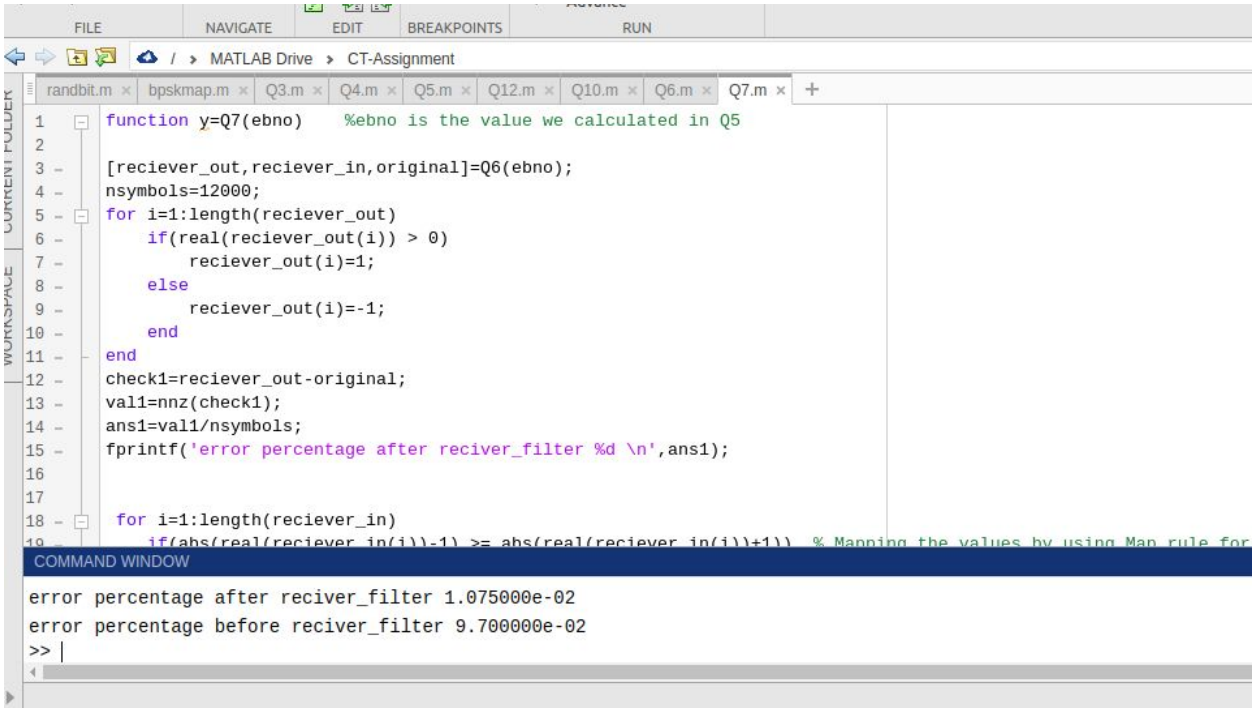
QUESTION-6:

- In this question we perform downsampling of the symbols we got after the receiver and the symbols we got before the receiver.
- We use the following formula for downsampling the symbols after the receiver(a in the given code snippet) and symbols before the receiver(b in the code snippet) respectively.
- Also in this question we calculated the value of sigma by using the fact that for BPSK Eb=1. Hence sigma=sqrt(½*ebno) where ebno is the parameter I am passing in the functions.

```
3
4 -   nymbols=12000;
5 -   a=zeros(1,nymbols);
6 -   m=4; %upsampling factor
7 -   for i=1:nymbols
8 -      a(i)=out(nsym_up*m +1 + (m*(i-1)));  % a(i) are the
9 -   end
10 -  b=zeros(1,nymbols);
11 -  for i=1:nymbols
12 -     b(i)=in(nsym_up*m/2 +1 + (m*(i-1)));  %b(i) are the
13 -  end
14 -  scatter(real(a),imag(a));
15    %scatter(real(b),imag(b));
16
17 - end
```

QUESTION-7:

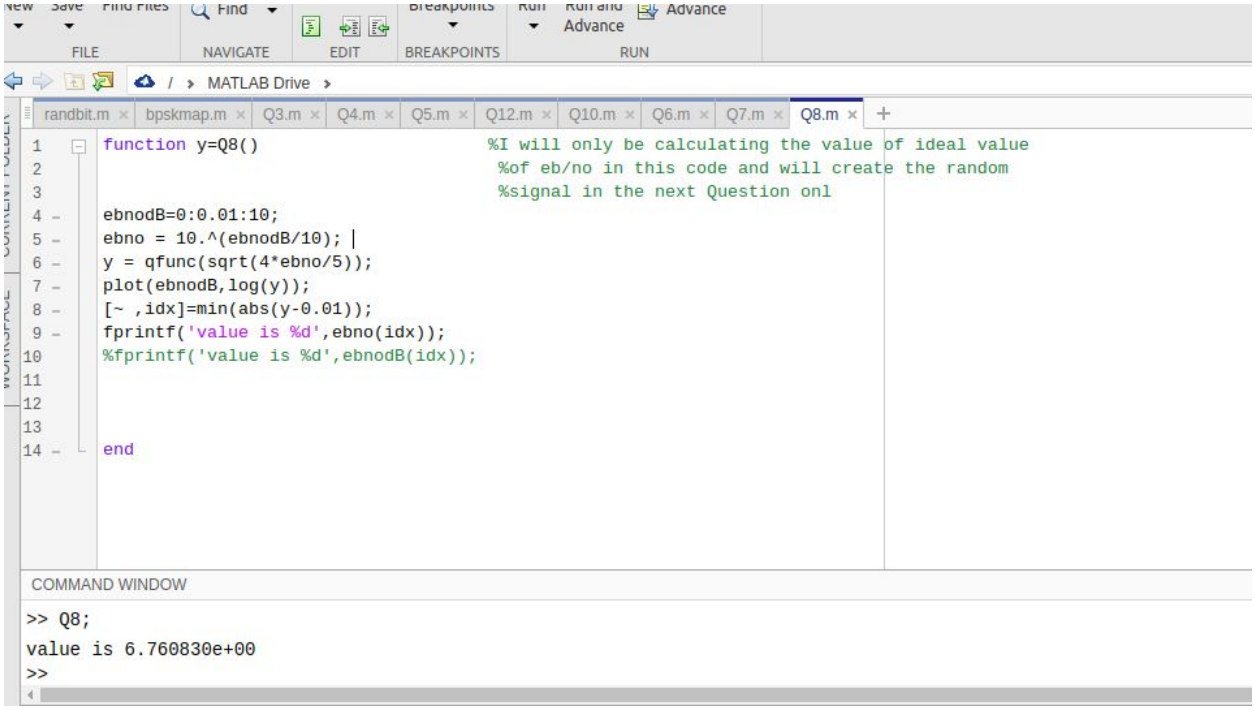
- In this question we made the decisions to decode the received symbols and then we calculated the error probability and compared that prob to the ideal value.
- We also calculated the error prob for the symbols before the receiver filter and that was significantly larger than the other prob, and the result was as we had predicted.
- The reason for this is that the ISI in the decision statistics.



- Above is a code snippet and answer for the values of the prob we got.
- We can also observe the decision rule that I have used for this question in the code snippet. I have used ML decoding rule.

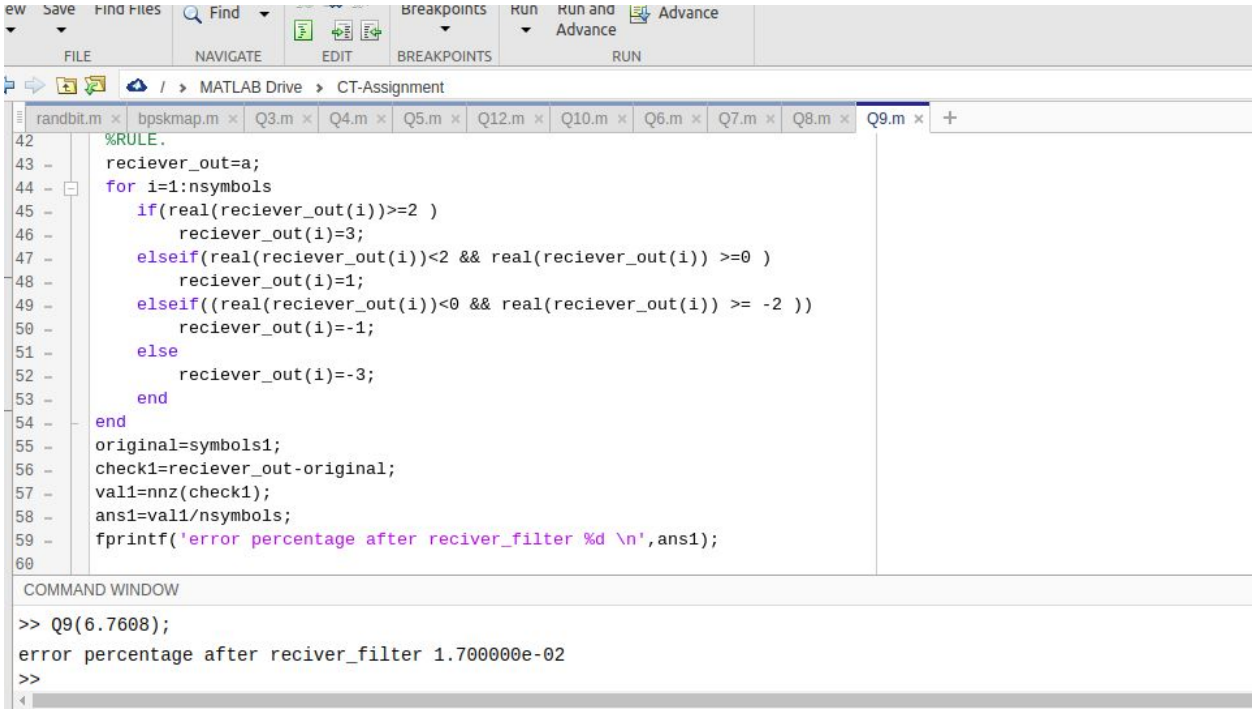
QUESTION-8:

- This question was very similar to Question 5, the only difference being that we have used a different expression for calculating the error prob.
- The expression this time was: prob=Q(sqrt(4*ebno/5))
- We used the same technique as Question 5 to find the value of Eb/No.
- Given below is the code snippet and the result we obtained.



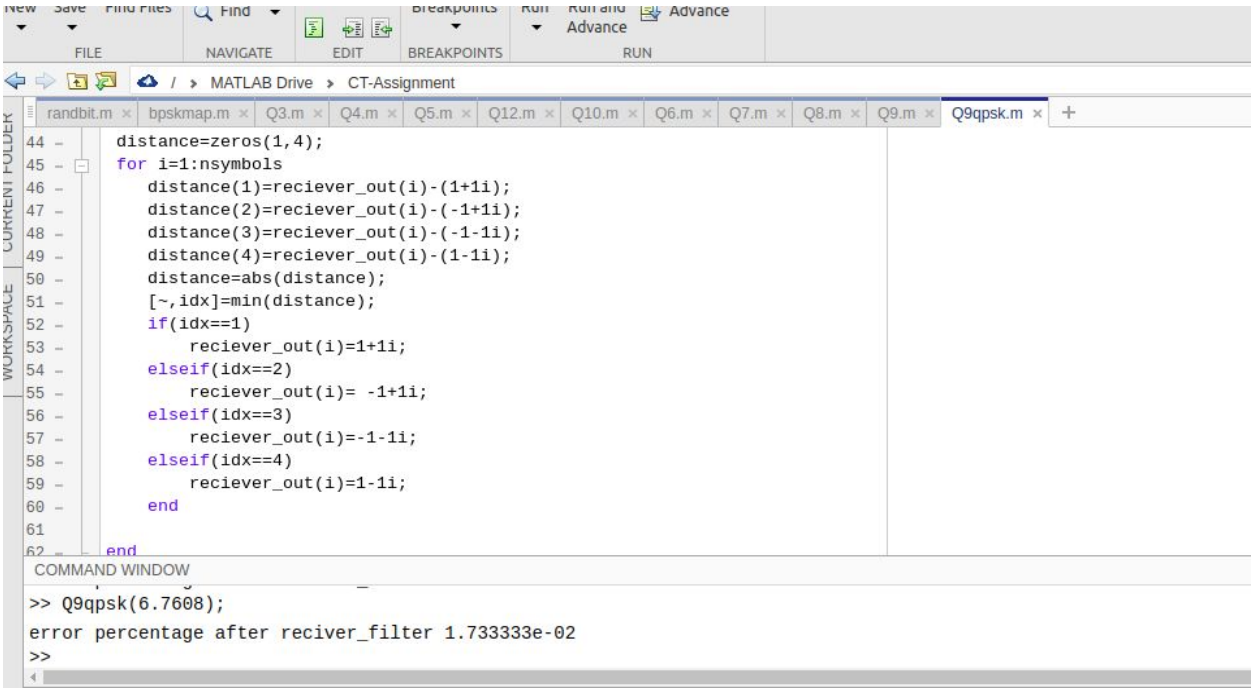
QUESTION-9:

- For this question I have combined the codes of the questions 4,6,7 and made the following changes:
    - ❖ Nysmbols are now 6000
    - ❖ The mapping function has changed and now fourpammap() was called for the mapping
    - ❖ The way in which the value of sigma was calculated was changed. In this code sigma=sqrt((5/(4*ebno))); We used this formula because we got Eb=5/2 for the 4-Pam constellation.
    - ❖ Decision rule also changed and the below code snippet shows the decision rule and the value of prob we got.



QUESTION -9 : (QPSK)

- The code for this was the same as Question 9 and only the mapping function and decision rule were changed.
- This time the qpskmap() function was called.

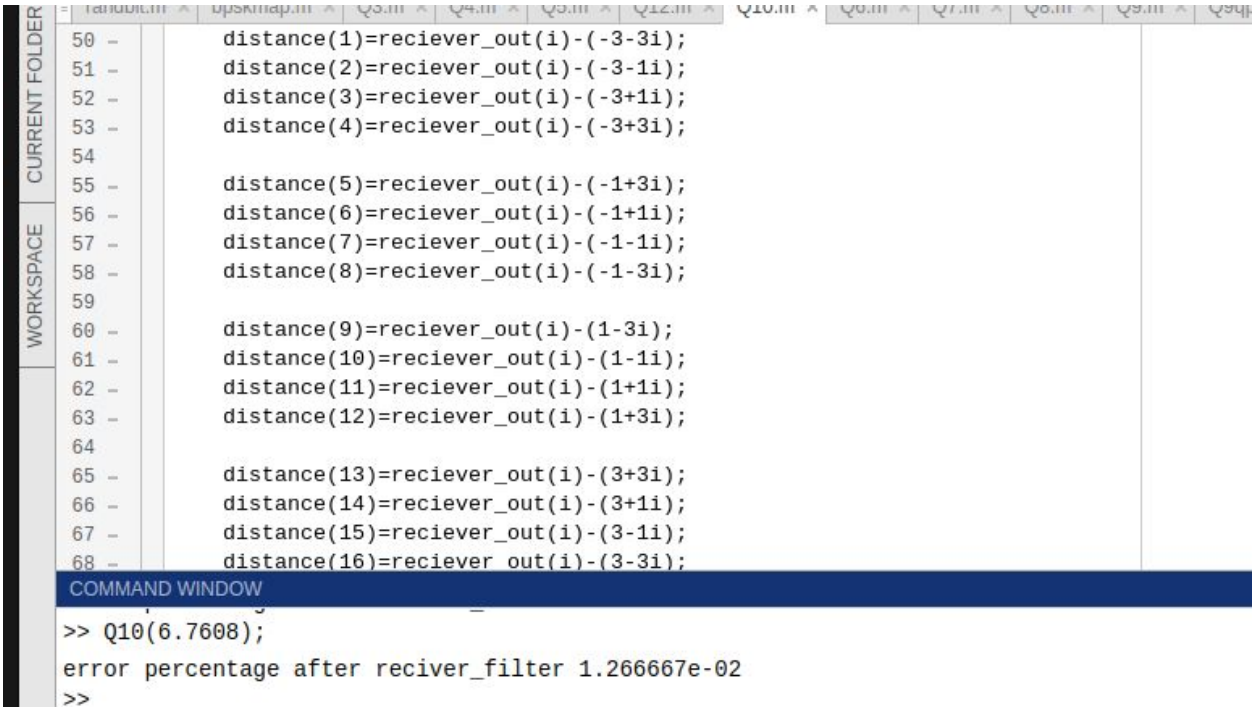


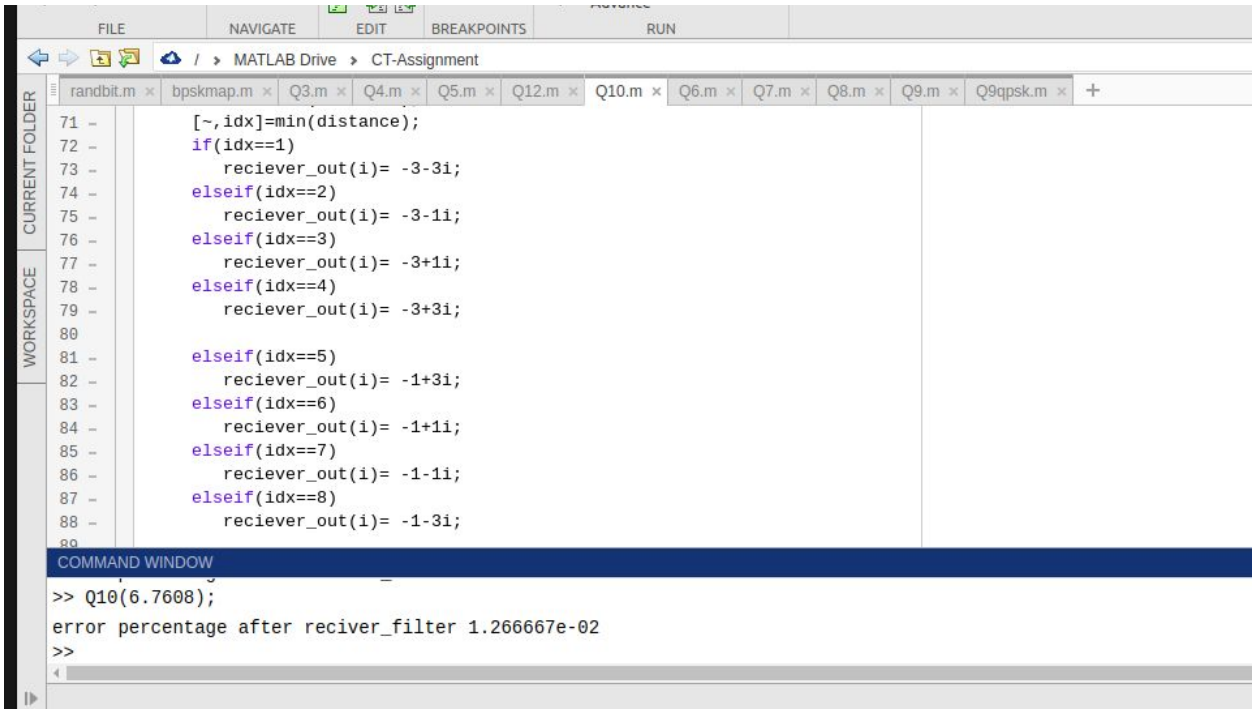
- Above is the code snippet of the decision rule and the value for the probability we got.

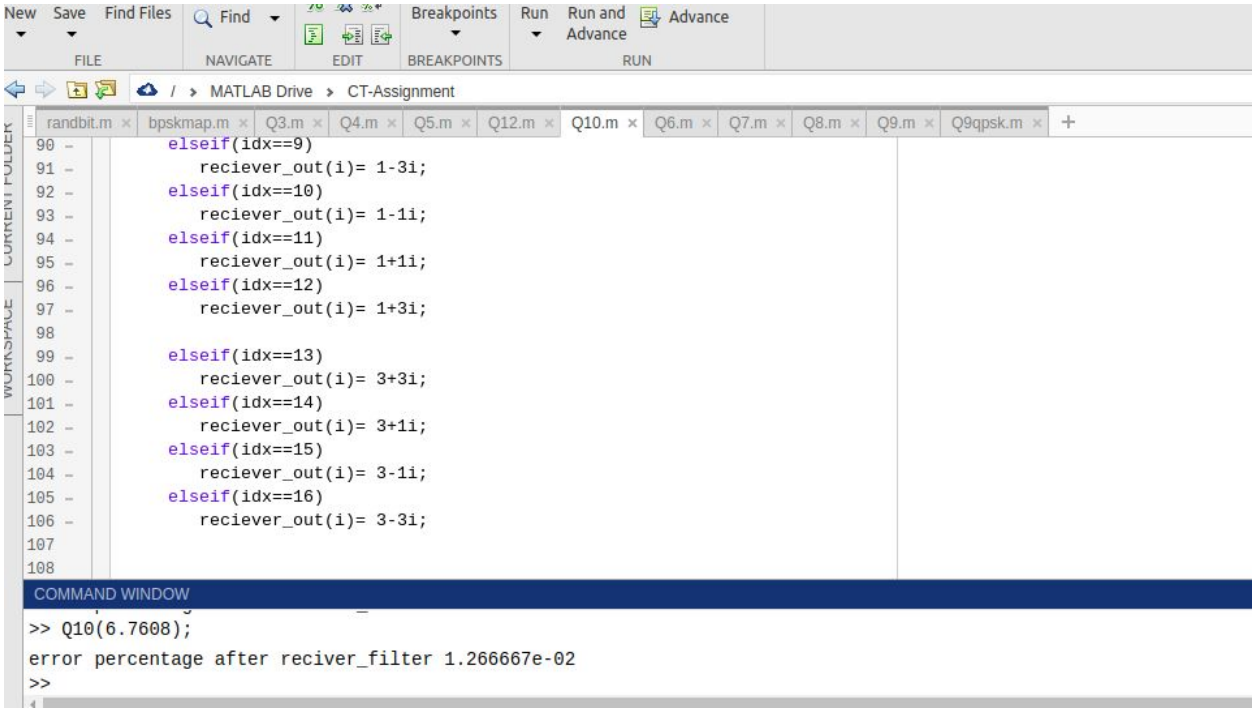
QUESTION-10:

- This code is similar to Question 9 with the following changes:
  - ❖ Nysmbols are now 3000
  - ❖ The mapping function has changed and now sixteenqammap() was called for the mapping
  - ❖ The way in which the value of sigma was calculated was changed. In this code sigma=sqrt((15/(16*ebno)));

  We used this formula because we got Eb=15/8 for the 16-Qam constellation.
  - ❖ Decision rule also changed and the below code snippet shows the decision rule and the value of prob we got.







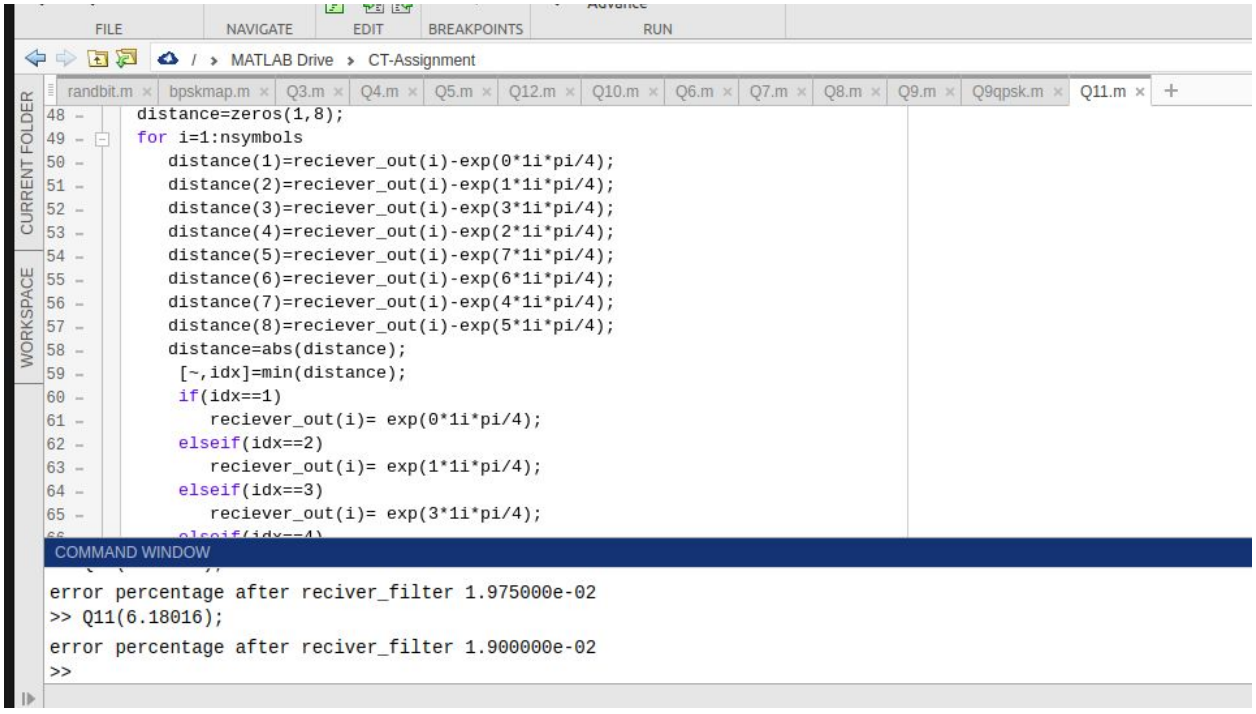
QUESTION-11:

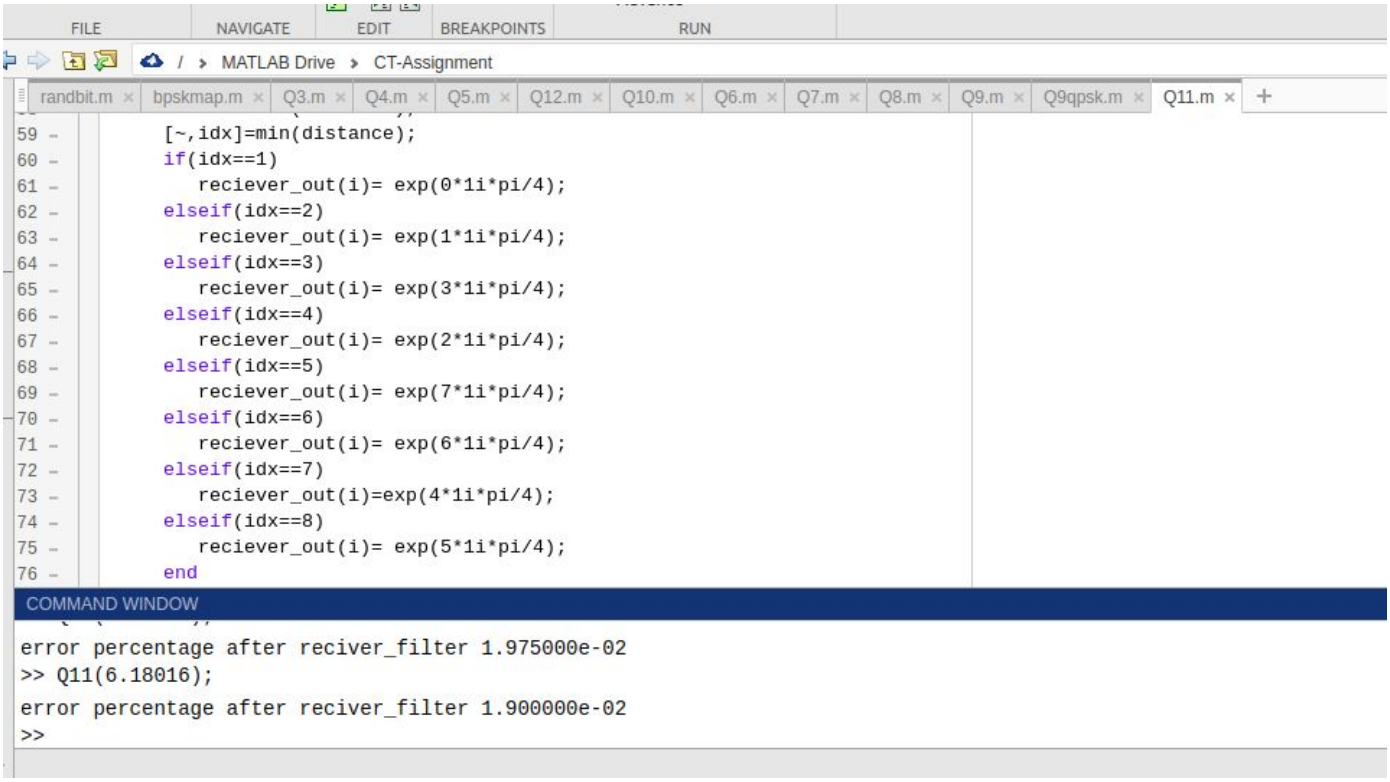
- This code is similar to Question 9 with the following changes:

- Nysmbols are now 4000
- The mapping function has changed and now EightpskMapping() was called for the mapping
- The way in which the value of sigma was calculated was changed. In this code sigma=sigma=sqrt((1/(6*ebno)));

We used this formula because we got Eb=1/3 for the 8-psk constellation.

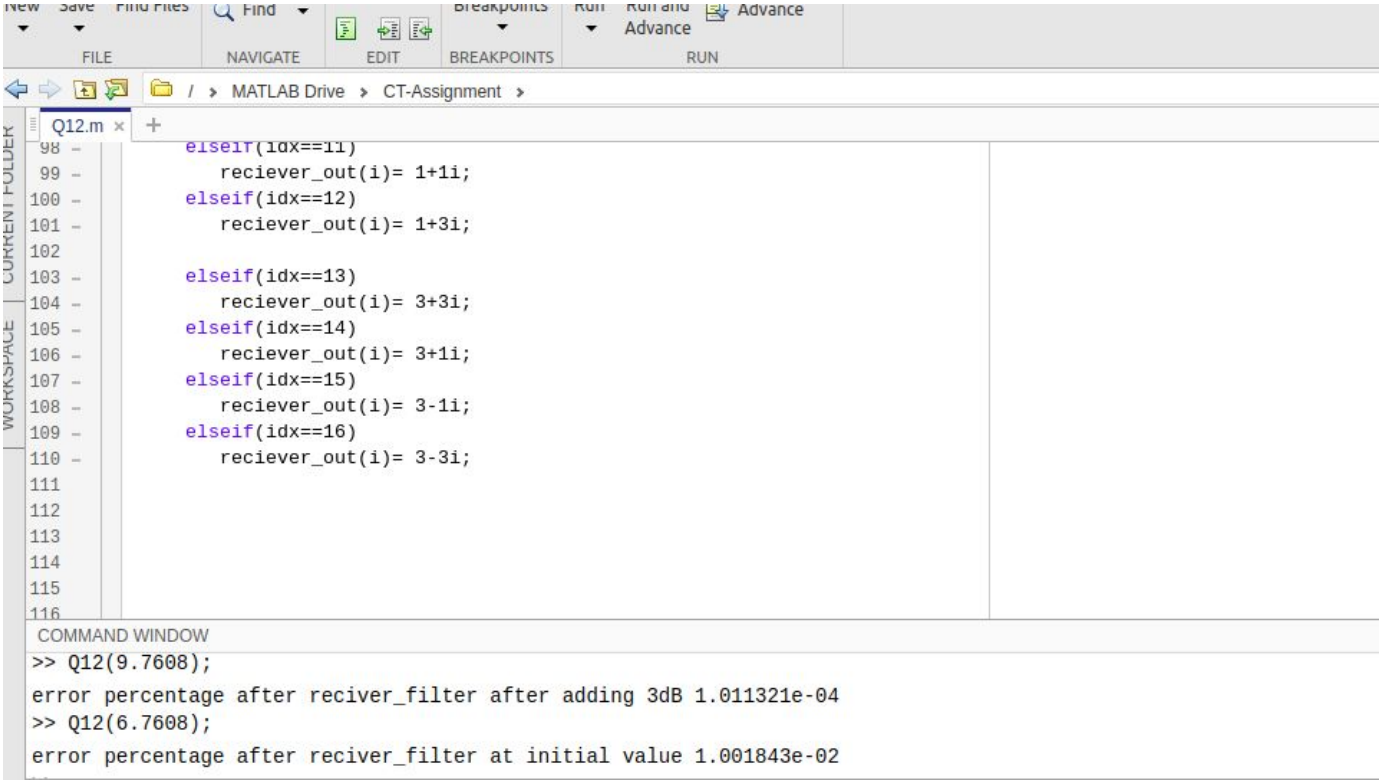
- Decision rule also changed and the below code snippet shows the decision rule and the value of prob we got.





QUESTION 12:

- In this question we have added the noise before we have convoluted with the transmitter filter.
- This helps in removing the noise and we see that the probability of error becomes very close to ideal after this.
- After this, we calculated the probability after adding 3dB to the earlier value of eb/no.
- The result was as expected and the probability of error dropped to approximately 0.0001.
- Also after making the scatter plots, we find out that the constellation comes closer when we run the code on the 3dB excess value.
- You can see the results in the below code snippet:


Find
Advance
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
MATLAB Drive
CT-Assignment
Q12.m
elseif(idx==11)
reciever_out(i)= 1+1i;
elseif(idx==12)
reciever_out(i)= 1+3i;
elseif(idx==13)
reciever_out(i)= 3+3i;
elseif(idx==14)
reciever_out(i)= 3+1i;
elseif(idx==15)
reciever_out(i)= 3-1i;
elseif(idx==16)
reciever_out(i)= 3-3i;
COMMAND WINDOW
>> Q12(9.7608);
error percentage after reciver_filter after adding 3dB 1.011321e-04
>> Q12(6.7608);
error percentage after reciver_filter at initial value 1.001843e-02

THE END