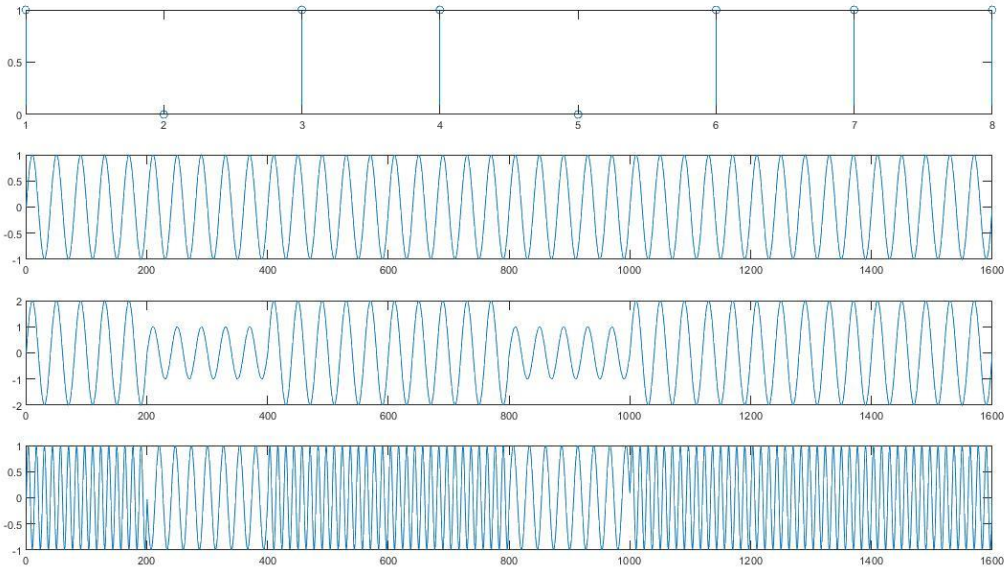
```
%%bit stream
subplot(4,1,1);
signal=round(rand(1,8));
stem(signal);
array=repmat(signal,200,1);
signal_reshaped=reshape(array,1,1600);

%%carrier
f=4;
t=linspace(0,10,1600);
carriar= sin(2*pi*f*t);
subplot(4,1,2);
plot(carriar);

%%ask
subplot(4,1,3);
res_ASK=(carriar.*(signal_reshaped+1));
plot(res_ASK);

%msk
subplot(4,1,4);
res_msk=sin(2*pi*f*((signal_reshaped+1)*1.5).*t);
plot(res_msk);
```



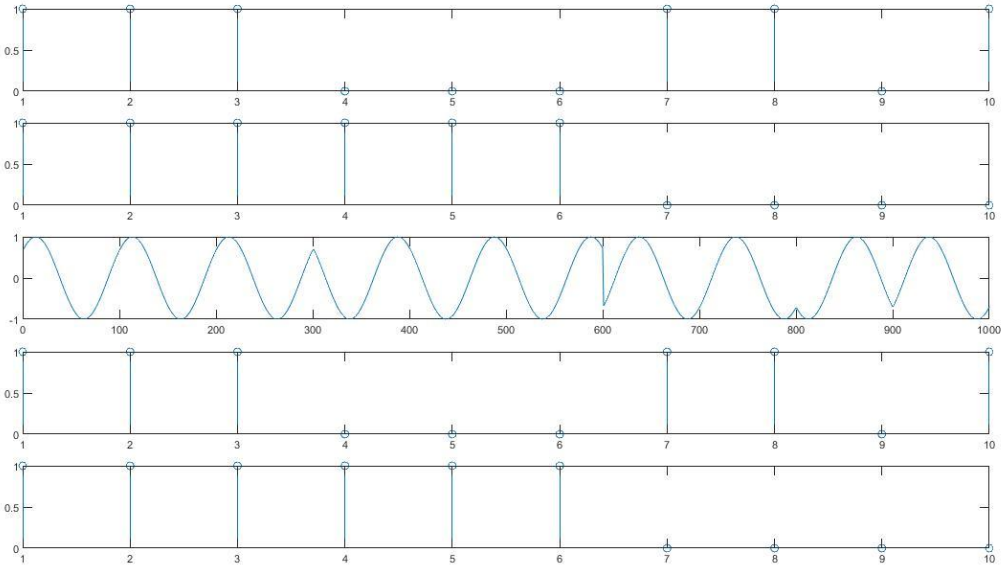
QPSK MODULATION LAB 2

QPSK FUNCTION

```
function [syp] = MY_QPSK_SYS(I,Q)

I=(I*2)-1;
Q=(Q*2)-1;
syp = I+ j*Q


end
```

QPSK MAIN

```
bitstream=randi([0,1],1,20);
I_bits=bitstream(1:10);
Q_bits=bitstream(11:20);

time=linspace(0,10,1000);

ans_qpsk = MY_QPSK_SYS(I_bits,Q_bits);

ans_qpsk=angle(ans_qpsk);

ans_qpsk=reshape(repmat(ans_qpsk,100,1),1,1000);

mod_signal=sin(2*pi*time + ans_qpsk);

I_COM=sin(2*pi*time);
Q_com=cos(2*pi*time);

demo_signal_I =ceil(intdump(mod_signal.*I_COM ,100));
demo_signal_Q =ceil(intdump(mod_signal.*Q_com ,100));


subplot(5,1,1);
stem(I_bits);
subplot(5,1,2);
stem(Q_bits);

subplot(5,1,3);
plot(mod_signal);

subplot(5,1,4);
stem(demo_signal_I);

subplot(5,1,5);
stem(demo_signal_Q);
```

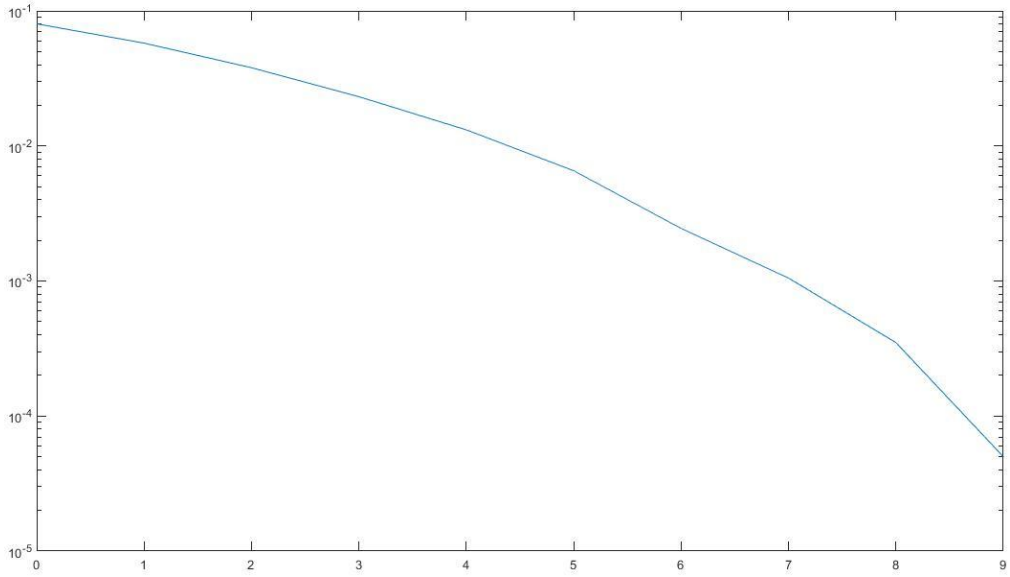

1
0.5
0
1 2 3 4 5 6 7 8 9 10
1
0.5
0
1 2 3 4 5 6 7 8 9 10
1
0
-1
0 100 200 300 400 500 600 700 800 900 1000
1
0.5
0
1 2 3 4 5 6 7 8 9 10
1
0.5
0
1 2 3 4 5 6 7 8 9 10

LAB 3 – bit error rate

QPSK

The main code :

```
close all;
I_bitstream=randi([0,1],1,10000);
Q_bitstream=randi([0,1],1,10000);

%%QPSK SYM
I=(I_bitstream*2)-1;
Q=(Q_bitstream*2)-1;
SYMBOLES =sqrt(1/2)*(I+ j*Q);


%%% noise
NOISE_I = randn(1,10000);
NOISE_Q = randn(1,10000);

noise_G = NOISE_I + j*NOISE_Q;

Var=[0.1, 0.1259,0.1585, 0.1995, 0.2512, 0.3162, 0.3981, 0.5012, 0.6310,
0.7943, 1];
emp_bits=[];
for v = Var

    Noise_G = noise_G .* sqrt(v)/2;

    signal_out = SYMBOLES+ Noise_G ;

    reBITS_I =ones(1,10000);
    reBITS_Q=ones(1,10000);

     reBITS_I(find(real(signal_out)<0))=0;
     reBITS_Q(find(imag(signal_out)<0))=0;

     b_error_1= sum(I_bitstream ~= reBITS_I);
      b_error_2= sum(Q_bitstream ~= reBITS_Q);

     sum_total=  ((b_error_1+b_error_2)/20000);



  emp_bits=[emp_bits sum_total ];

end

SNR_COMP=10*log10(1./Var);
semilogy(SNR_COMP,emp_bits);
```


10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
1
2
3
4
5
6
7
8
9