Lab-3

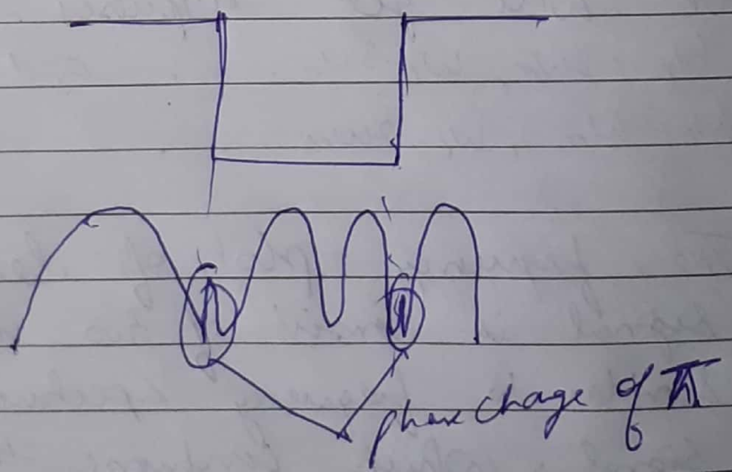
$c(t) = \sin(2\pi f t)$

message signal is ~~inform~~ an array of bits (1 or 0), where 1 is mapped to +1 and 0 to -1.

After mapping ~~(this~~ is) the information bit is converted to a square wave with amplitude +1 and -1.

For modulation, this square pulse is multiplied with carrier signal.

Wherever ~~ther~~ the amplitude of square signal changes from +1 to -1, there is a phase chance of π in the modulated signal.



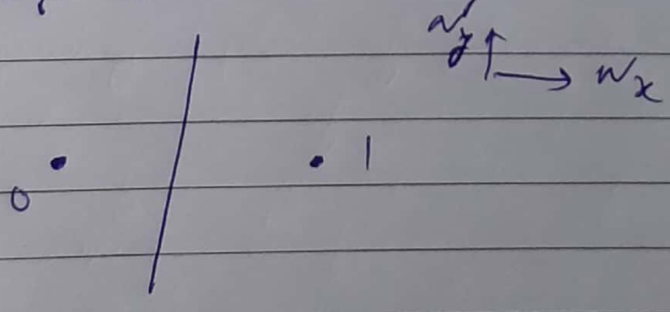
Then when this modulated signal is passed through channel some noise is added through it. Function 'awgn' in matlab is used for additive white gaussian noise. After this we get a noisy signal.

The demodulation of this noisy signal is ~~done~~ by multiplying it with carrier signal. But the demodulated message signal ~~is~~ also ~~noisy~~ contains noise.

So to find out whether the information bit was 1 or 0 the demodulated signal is decoded

To draw curve for BER vs SNR and to compare it with theoretical value i have followed the following steps:

1. generate a random signal $s \in (1,0)^T$
2. generate AWGN
3. received signal $(y) = S + N$
4. check whether the received signal is in left side or right side.



5. If there is an error increment the error count.

Now ~~[illegible]~~ to compare it with theoretical value which is $\frac{1}{2}\ \frac{1}{2}\sqrt{\frac{E_b}{N_0}}$

plot it on semilog plot.

```
clc;
clear all;
close all;

num_bit=10; %number of bit
data=randi([0,1],num_bit); %random bit generation(1 or 0)
pData=2*data-1;


%%Define Carrier
f=1000; %carrier frequency
fs=f*10; %sampling frequency
Ts=1/fs;
T=1/f;
M=2; %modulation index
n=M*length(data);
t=0:Ts:n*T; %defining x-range values
car=sin(2*pi*f*t); %carrier frequency
subplot(2,1,1);
stem(pData); %plotting data points after mapping 1 to 1 and 0 to -1
xlabel('Time');
ylabel('Amplitude');
title('mapped data points');
subplot(2,1,2);
plot(car);  %plotting carrier frequency
xlabel('Time');
ylabel('Amplitude');
title('Carrier signal');


%convert impulse data to square pulse
%converting the data points to square pulse with amplitude varying betwen 1
%and -1
tp=0:Ts:T*M;
exData=[];
for(i=1:length(data))
    for(j=1:length(tp)-1)
        exData=[exData pData(i)];
    end
end
exData=[exData 0];
figure;
subplot(2,1,1);
plot(exData,'r-','LineWidth',2); %plotting square wave
xlabel('Time');
ylabel('Amplitude');
title('Square Pulse');
grid on;


%%Modulation
%modulation is done by multiplying square wave by carrier frequency
mSig=exData.*car;
```

```
subplot(2,1,2);
plot(mSig,'b-','LineWidth',1);
xlabel('Time');
ylabel('Amplitude');
title('Modulated signal');


%%channel
%here i am adding additive white gaussian noise to the modulated signal
%with SNR varying between -10 and 10 in steps of 1
SNR=-10:10;
for(k=1:length(SNR))
    rx=awgn(mSig,SNR(k));
end
figure;
plot(rx,'g-','LineWidth',1); %plotting noisy signal
xlabel('Time');
ylabel('Amplitude');
title('Noisy signal');
grid on;


%%Demodulation
%demodulation can be simply done by multiplying nosiy signal with carrier
%signal.
%By this we will get a noisy version of orginal square pulse
dem=rx.*car;
figure;
plot(dem,'r-','LineWidth',1); %plotting demodulated signal
xlabel('Time');
ylabel('Amplitude');
title('Demodulated signal');
grid on;

%%decoding
%here i am deciding whether the incoming bit is 1 or 0 and is stored
%in rcv
k=1;
rcv=[];
for(i=1:length(data))
    sm=0;
    for(j=1:length(tp)-1)
        sm=sm+dem(k);
        k=k+1;
    end
    if(sm>0)
        rcv=[rcv 1];
    else
        rcv=[rcv 0];
    end
end
ber=sum(data~=rcv)/num_bit;
ber
```

```
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%XXXXXXXX BER performance annalysis of BPSK modulation Technique XXXXXXXXX
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
clc;
clear all;
close all;
num_bit=1500;%number of bit
data=randi([0,1],num_bit);%random bit generation (1 or 0)
s=2*data-1;%conversion of data for BPSK modulation
SNRdB=-10:1:10; % SNR in dB
SNR=10.^(SNRdB/10);

%calculation of error
for(k=1:length(SNRdB))%BER (error/bit) calculation for different SNR
y=awgn(complex(s),SNRdB(k));
error=0;
for(c=1:1:num_bit)
    if (y(c)>0&&data(c)==0)||(y(c)<0&&data(c)==1)%logic acording to BPSK
        error=error+1;
    end
end
error=error/num_bit; %Calculate error/bit
m(k)=error;
end


figure(1)
%plot start
semilogy(SNRdB,m,'o','linewidth',2.5);
grid on;
hold on;
BER_th=(1/2)*erfc(sqrt(SNR));
semilogy(SNRdB,BER_th,'r','linewidth',2);
title(' curve for Bit Error Rate verses  SNR for Binary PSK modulation');
xlabel(' SNR(dB)');
ylabel('BER');
legend('simulation','theorytical')
axis([-10 10 10^-5 1]);
%XXXXXXXXXXXXXXXXXXXX End of program XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

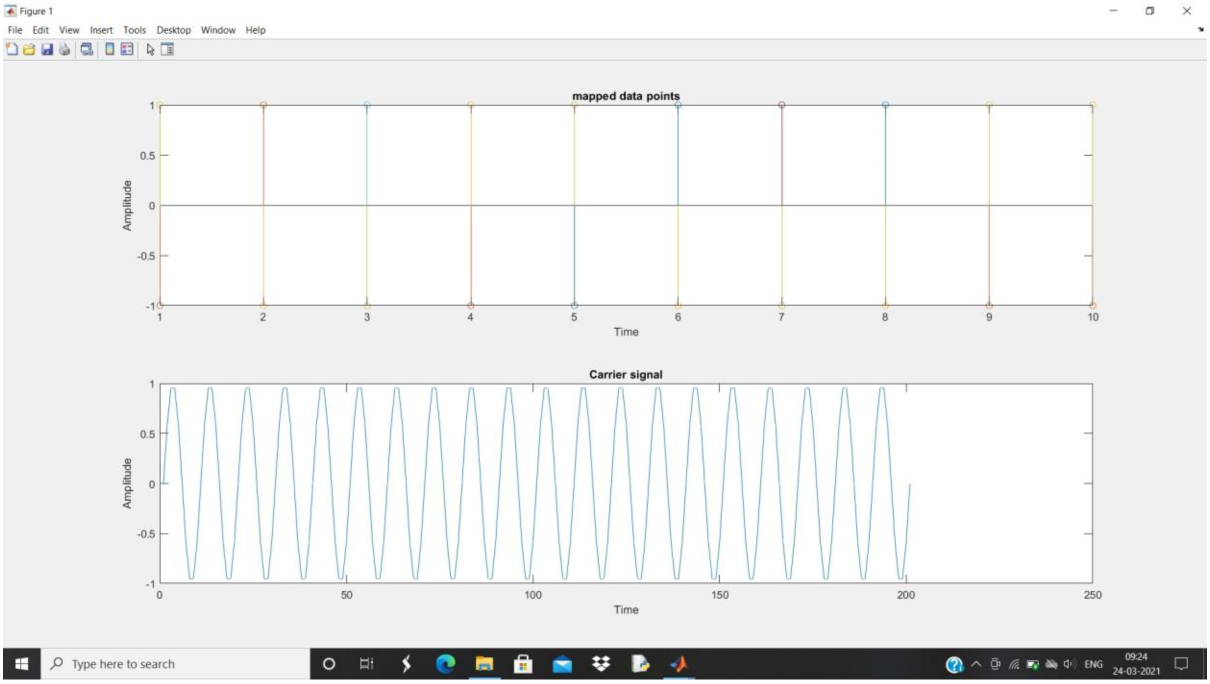


Figure 1
File Edit View Insert Tools Desktop Window Help
mapped data points
Amplitude
Time
Carrier signal
Amplitude
Time
Type here to search
ENG
09:24
24-03-2021

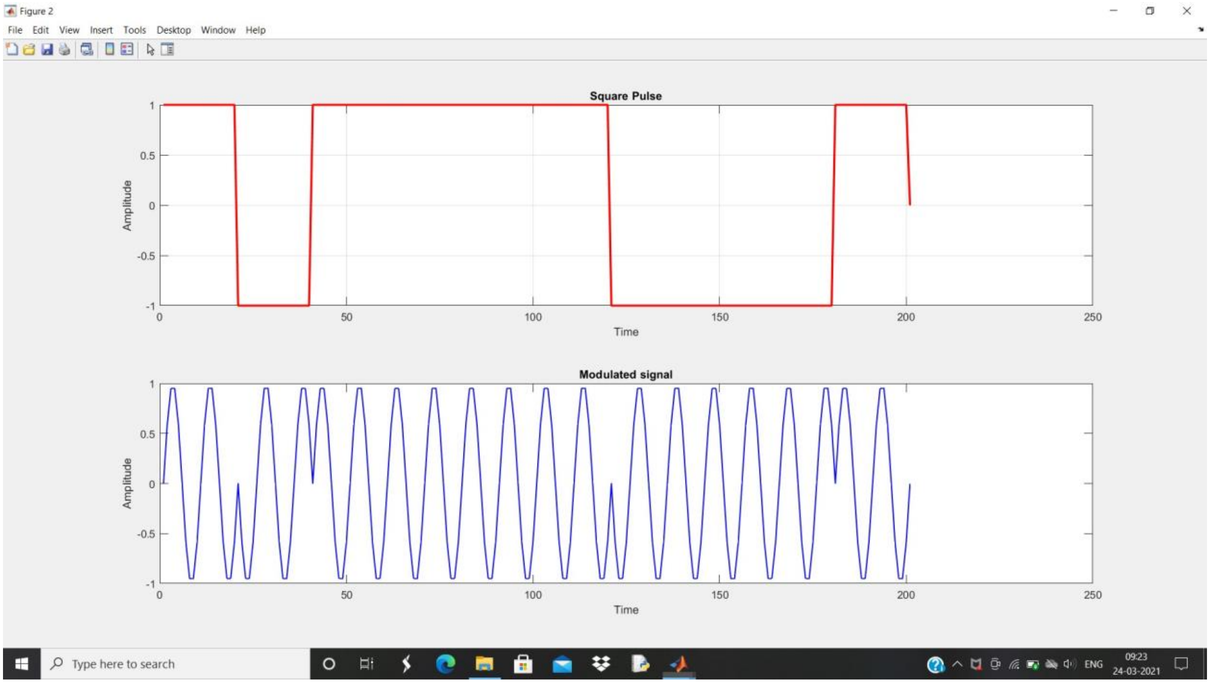


Figure 2
File Edit View Insert Tools Desktop Window Help
Square Pulse
Amplitude
Time
Modulated signal
Amplitude
Time
Type here to search
ENG
09:23
24-03-2021

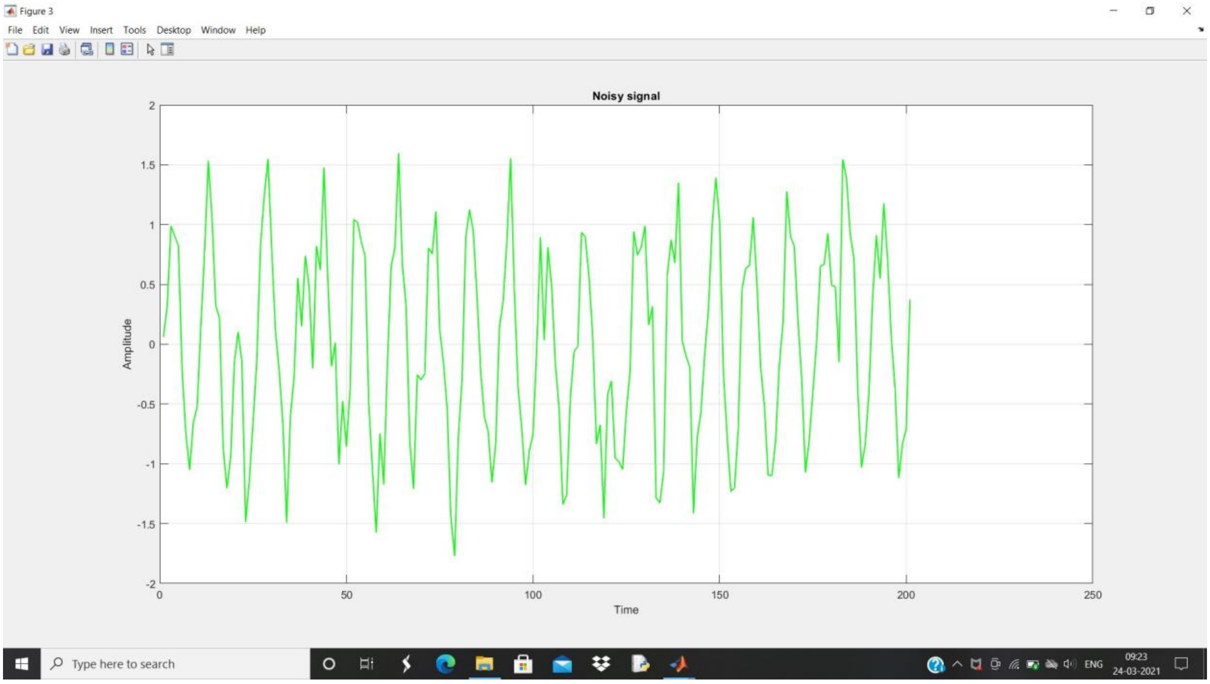


Figure 3
File Edit View Insert Tools Desktop Window Help
Noisy signal
Amplitude
Time
Type here to search
ENG
09:23
24-03-2021

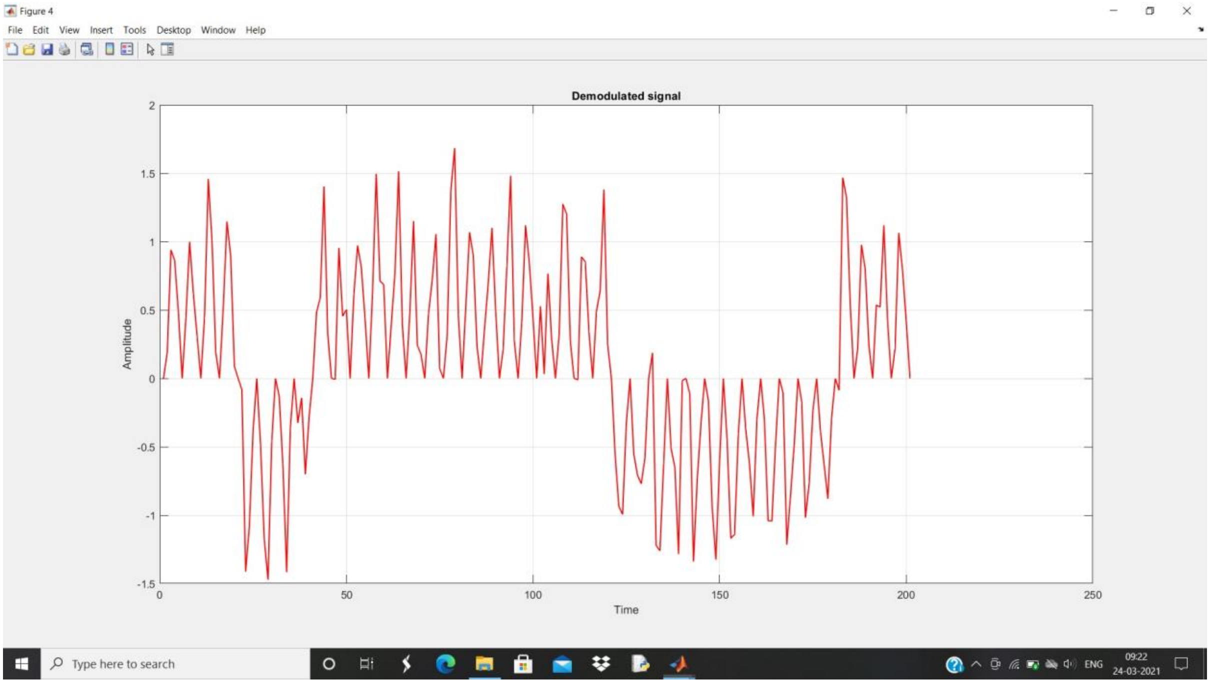


Figure 4
File Edit View Insert Tools Desktop Window Help
Demodulated signal
Amplitude
Time
Type here to search
ENG
09:22
24-03-2021

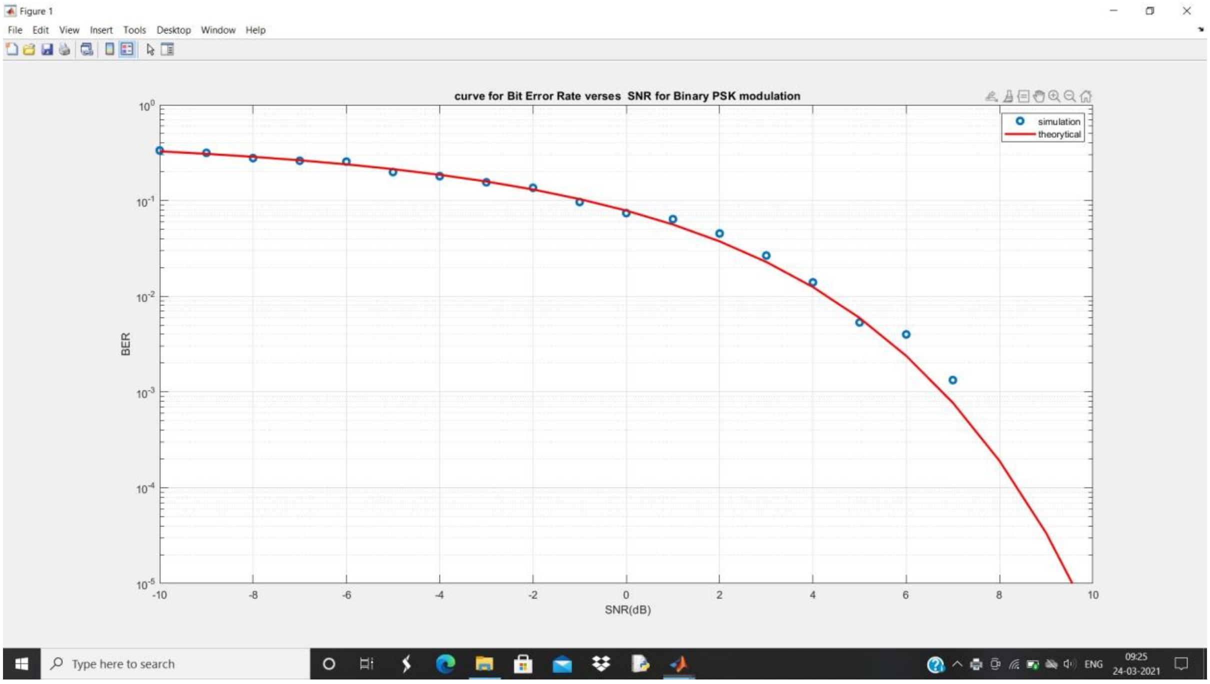


Figure 1
File Edit View Insert Tools Desktop Window Help
curve for Bit Error Rate verses SNR for Binary PSK modulation
simulation
theorytical
BER
SNR(dB)
Type here to search
ENG
09:25
24-03-2021