Namal Institute Mianwali

EE 322 (L) – Analog and Digital Communications (Lab)

Lab – 11–13 (30% of total lab marks)

Dated: August 15, 2021

(Submitted by)

Muhammad Faizan Ikram

Roll No: 2018-UET-NML-Elect-27

(Submitted to)

Mr. Arbaz

Lab Engineer

Electrical Engineering Department

Namal Institute Mianwali

Namal Institute Mianwali

Table of Contents

Namal Institute Mianwali

Introduction:

In this lab session we will understand the working of a digital communication system and design our own problem related to the digital communication system. We will analyze the results and output graphs using MATLAB.

Experimental Tool:

- MATLAB

Environment (IDE):

- MATLAB Programming IDE

Task -01

Methodology:

We have used MATLAB software to write code for BASK Modulation and Demodulation for an audio signal in task 1. In this task, we have used a voice signal for a digital communication through Binary Amplitude Shift Keying Modulation and Demodulation Scheme. We integrated each communication block and observed the output. The output was almost similar after demodulation.

Code:

Plotting Input Signal:

```
clc; close all; clear all;

%if the input is x with sampling frequency fs
%%%%%%%%%%%%%% Input Signal %%%%%%%%%%%%%%%%%%

[x, fs] = audioread('counting0.wav');       % Input Signal
%-----------------------------------------------------------------
% Plotting Input Signal
l1=length(x);
t1=-((l1-1)/2):1:((l1-1)/2);
t1=t1';
figure
subplot(2,1,1)
plot(t1,x);
title('Input Signal'), xlabel('time (sec)'), ylabel('y(t)');
%-----------------------------------------------------------------
```

Quantization of Input Signal:

```
%%%%%%%%%%%%%%% Quantization and PCM Encoding %%%%%%%%%%%%%%%%%%

n = 4;                          %the number of bits for PCM encoding;
L =2^n;

t = [0:1:length(x)-1]/fs;

xmax=max(max(x));
xmin=min(min(x));
del=(xmax-xmin)/L;
partition=xmin:del:xmax;
codebook=xmin-(del/2):del:xmax+(del/2);
[indx1,quantv1]=quantiz(x(:,1),partition,codebook);
%[indx2,quantv2]=quantiz(x(:,2),partition,codebook); % if two channels are
                                                     % to be quantized

for i=1:length(indx1)
    if(indx1(i)~=0)
        indx1(i) = indx1(i)-1;
    end
end
for i=1:length(quantv1)
    if(quantv1(i)==xmin-(del/2))

        quantv1(i)=xmin+(del/2);
    end
end

code=de2bi(indx1,'left-msb');
k=1;
xpcm = zeros(length(indx1)*n,1);
for i=1:length(indx1)
    for j=1:n
        xpcm(k)=code(i,j);
        k=k+1;
    end
end
```

Channel Encoding:

```
%--------------------------------------------------
%%%%%%%%%%%%%%% Channel Encoding %%%%%%%%%%%%%%%%%%

M = 3;
n1 = 2^M-1;                    % Code length
k1 = n1-M;                      % Message length

data = reshape(xpcm,length(xpcm)/k1, k1);
encData = encode(data,n1,k1,'hamming/binary');
encData1 = reshape(encData, length(encData)*n1,1);
```

BASK Modulation:

```
%-------------------------------------------------
%%%%%%%%%%%%%% BASK Modulation %%%%%%%%%%%%%%%%%%

%%%input is encoder output;

Tb = 0.1; %bit interval
ts = Tb/100;
fs = 1/ts;
nb = Tb/ts; %number of samples in one bit interval
fc = fs/10;

xmod = zeros(length(encData1)*nb,1);
j = 1;
for i = 1:nb:length(xmod)
    xmod(i:i+(nb-1)) = encData1(j);
    j = j+1;
end

t = 0:ts:(length(xmod)*ts)-ts;
car1 = cos(2*pi*fc*t);
xmodbask = xmod.*car1';
```

BASK Demodulation:

```
%---------------------------------------------------
%%%%%%%%%%%%%% BASK Demodulation %%%%%%%%%%%%%%%%%%

xdemodbask = abs(xmodbask);

%nb is the number of samples in one bit interval

xdemod = zeros(length(xdemodbask)/nb,1);
thresh1 = (max(xdemodbask)-min(xdemodbask))/2;
j = 1;
for i = 1:nb:length(xdemodbask)
    av = sum(xdemodbask(i:i+(nb-1)))/nb;
    if av >= thresh1
        xdemod(j) = 1;
    else
        xdemod(j) = 0;
    end
    j = j+1;
end
```

Namal Institute Mianwali

NAMAL INSTITUTE

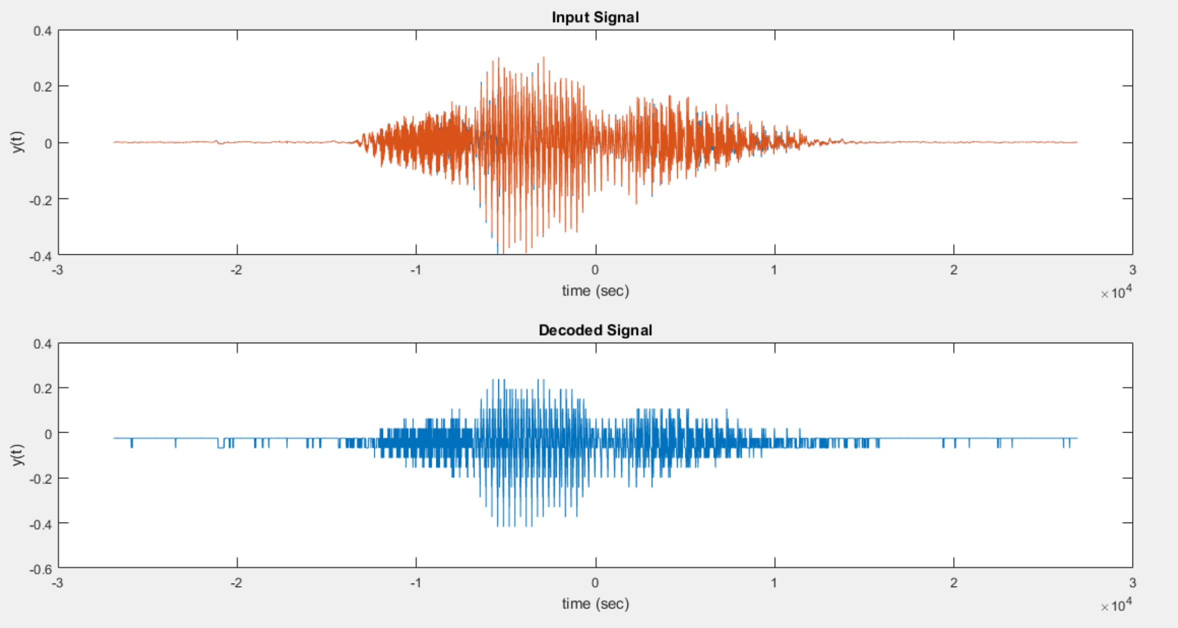
Channel Decoding:

```
%---------------------------------------------------
%%%%%%%%%%%%%% Channel Decoding %%%%%%%%%%%%%%%%%%

%if the input is named xdemod

decData1 = reshape(xdemod,length(xdemod)/n1,n1);
decData2 = decode(decData1,n1,k1,'hamming/binary');
numerr = biterr(data,decData2)
decData = reshape(decData2,length(decData2)*k1,1);
```

PCM Decoding:

```
%---------------------------------------------------
%%%%%%%%%%%%%% PCM Decoding %%%%%%%%%%%%%%%%%%

%if the input is named decData

xpcm1 =reshape(decData,n,length(decData)/n);
index =bi2de(xpcm1','left-msb');
xdecoded = codebook(index+1);

% Plotting Decoded Signal
l1=length(xdecoded);
t1=-((l1-1)/2):1:((l1-1)/2);
t1=t1';
subplot(2,1,2)
plot(t1,xdecoded);
title('Decoded Signal'), xlabel('time (sec)'), ylabel('y(t)');
```

Plots:



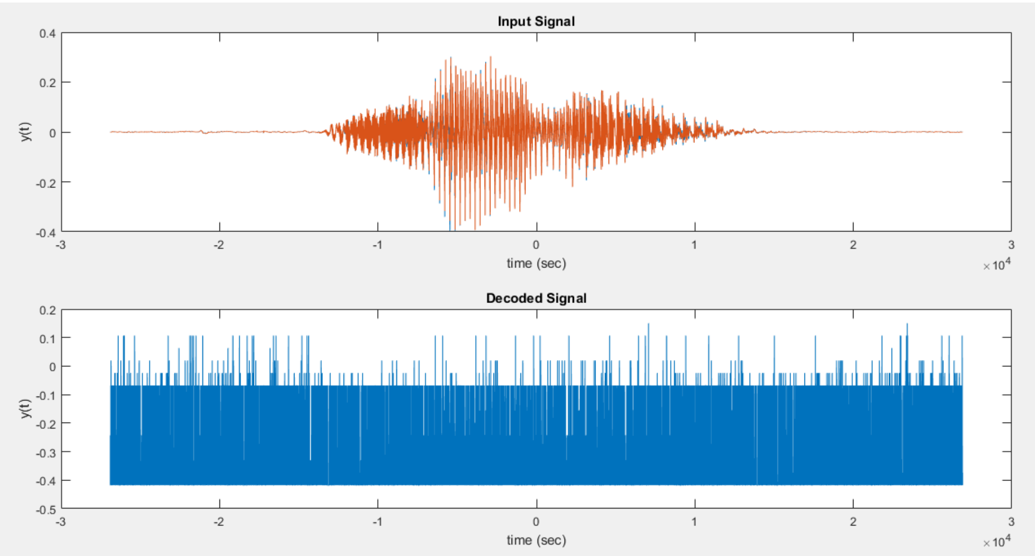
Comments: As we can see that after modulation and demodulation, the signal receive is an audio signal with pretty much same properties as input signal.

Now we will add a Gaussian Noise in the signal and see the effect.

Adding Gaussian Noise:

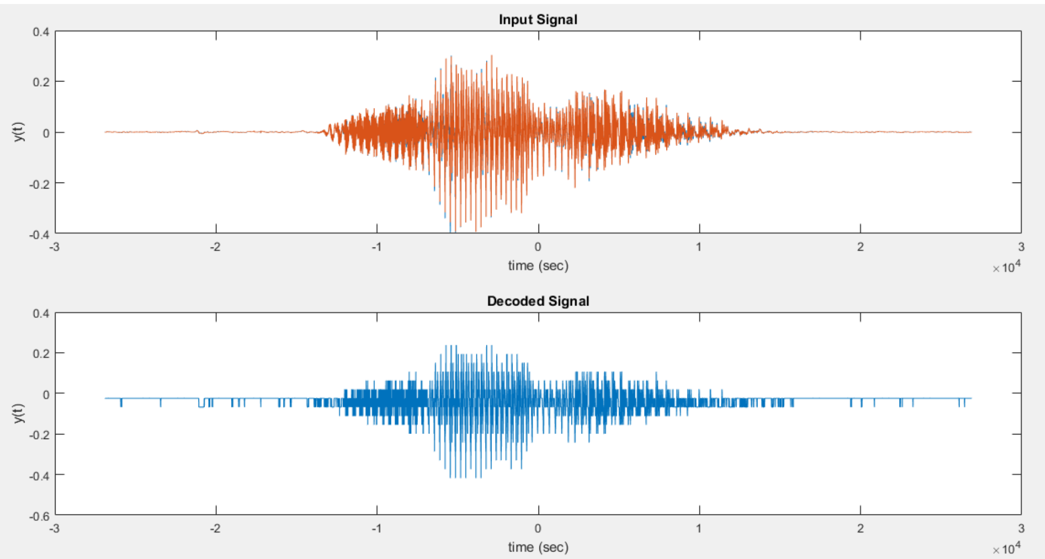
```
%--------------------------------------------------
%%%%%%%%%%%%%% Adding Gausian Noise %%%%%%%%%%%%%%%

xmodbask = awgn(xmodbask,25);
```

Plot with Noise:



Comments: The SNR has affected the input signal after modulation and the resulted output is noisy. If the SNR is increased, we get better output as shown in figure below:

Plot High SNR = 50:



Namal Institute Mianwali

Task-02

Problem Statement:

"To study the effect of BFSK Modulation and Demodulation on a Random Binary Signal."

Methodology:

We have used MATLAB software to write code for BFSK Modulation and Demodulation for a discrete random binary signal. In this task, we have used a random discrete signal and did bit stream. We used code for representation of this random binary stream. After this we have used BFSK modulation by using MATLAB built-in command ***fskmod*** [1] and then demodulated the signal using built-in command ***fskdemod.*** After demodulation we represented the original signal again with bit streaming. We have also analyzed the results by adding Gaussian Noise.

Code:

Input Binary Signal and Bit Streaming:

```
clc; close all; clear all;

%%%%%%%%%%%%%% Input Signal %%%%%%%%%%%%%%%%%%

N = 8;    % The number of bits
x = round(rand(1,N));  % Generate a random bit stream as Input Signal
bp=.000001;              % bit period
disp(' Binary information at Trans mitter :');
disp(x);
n = length(x);
t = 0:.01:n;
%-----------------------------------------------------------------
% Bit Streaming
bit=[];
for n=1:1:length(x)
    if x(n)==1;
       se=ones(1,100);
    else x(n)==0;
        se=zeros(1,100);
    end
     bit=[bit se];

end
```

Plotting Input Signal:

```
% Plotting Input Signal
t1=bp/100:bp/100:100*length(x)*(bp/100);
subplot(211);
plot(t1,bit,'lineWidth',2.5);grid on;
axis([ 0 bp*length(x) -.5 1.5]);
title('Binary Input Signal'), ylabel('amplitude(volt)'), xlabel('time(sec)');
```

BFSK Modulation:

```
%-------------------------------------------------
%%%%%%%%%%%%%% BFSK Modulation %%%%%%%%%%%%%%%%%%

% input is bit output;

Tb = 0.1; %bit interval
ts = Tb/100;
fs = 1/ts;  % Sample rate (Hz)
nb = Tb/ts; %number of samples in one bit interval
fc = fs/10;

M = 2;          % Modulation order
freqsep = 10; % Frequency separation (Hz)
nsamp = 8;     % Number of samples per symbol

xmodfask = fskmod(bit,M,freqsep,nsamp,fs);
```

BFSK Demodulation:

```
%---------------------------------------------------
%%%%%%%%%%%%%% FASK Demodulation %%%%%%%%%%%%%%%%%%

xdemod = fskdemod(xmodfask,M,freqsep,nsamp,fs);
```

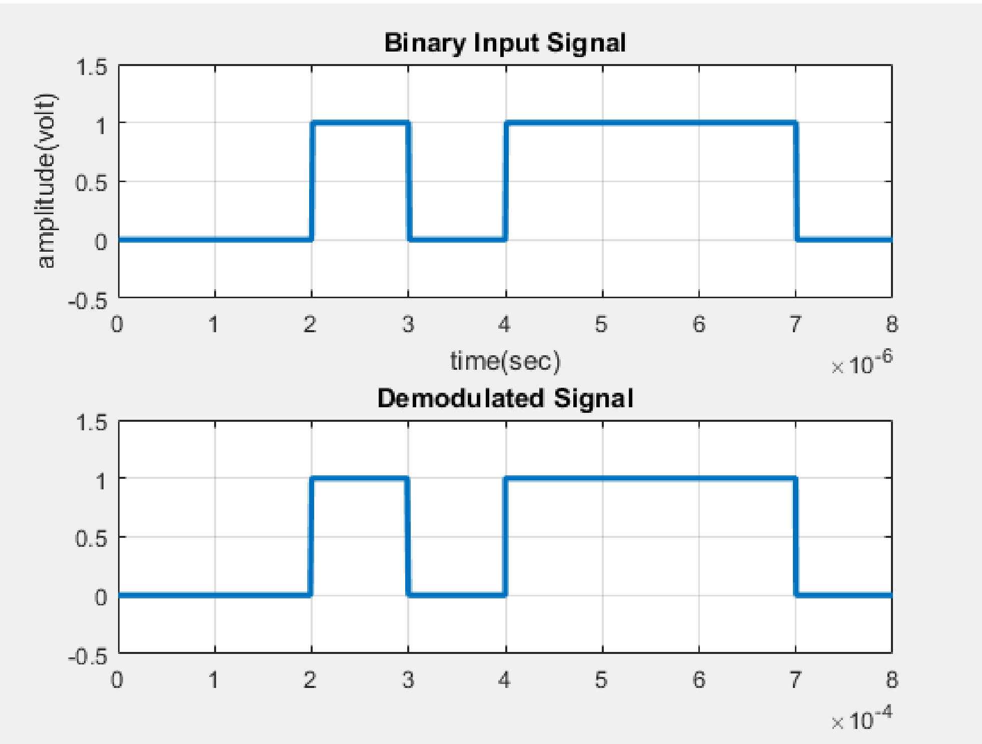
Output Signal Bit Streaming and Plot Code:

```
%-----------------------------------------------------
%%%%%%%%%%%%%% Demodulation Signal Represenatation %%%%%%%%%%%%%%%%%%

bit=[];
for n=1:length(xdemod);
    if xdemod(n)==1;
       se=ones(1,100);
    else xdemod(n)==0;
        se=zeros(1,100);
    end
     bit=[bit se];

end

t4=bp/100:bp/100:100*length(xdemod)*(bp/100);
subplot(212)
plot(t4,bit,'LineWidth',2.5);grid on;
axis([ 0 bp*length(xdemod) -.5 1.5]);
title('Demodulated Signal');
```

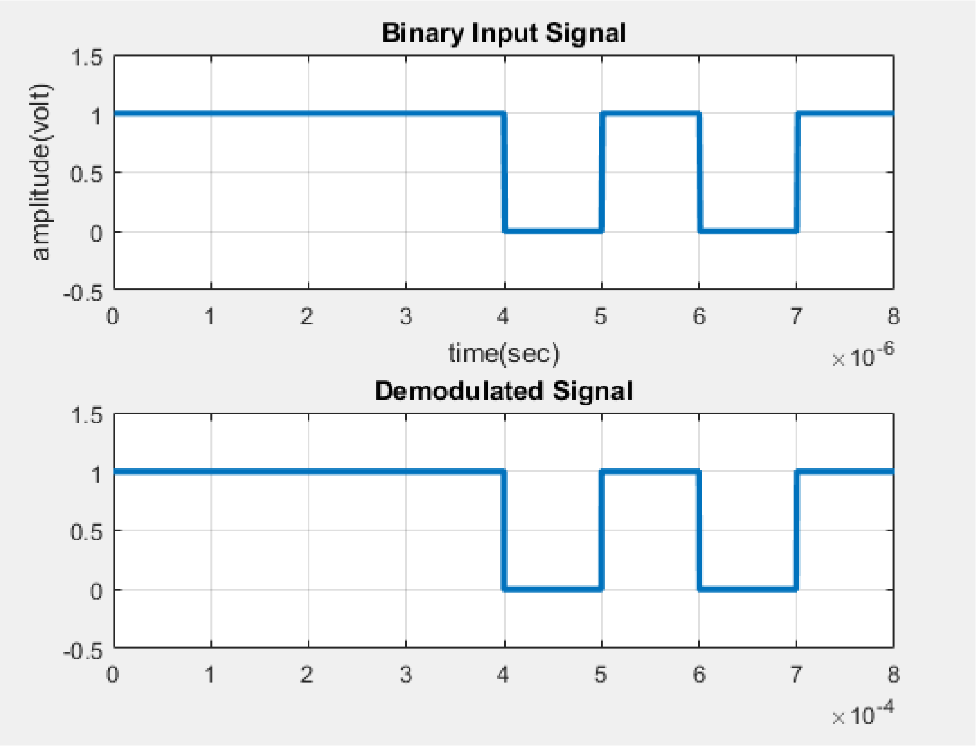
Plots:

These are the plots with random binary stream:

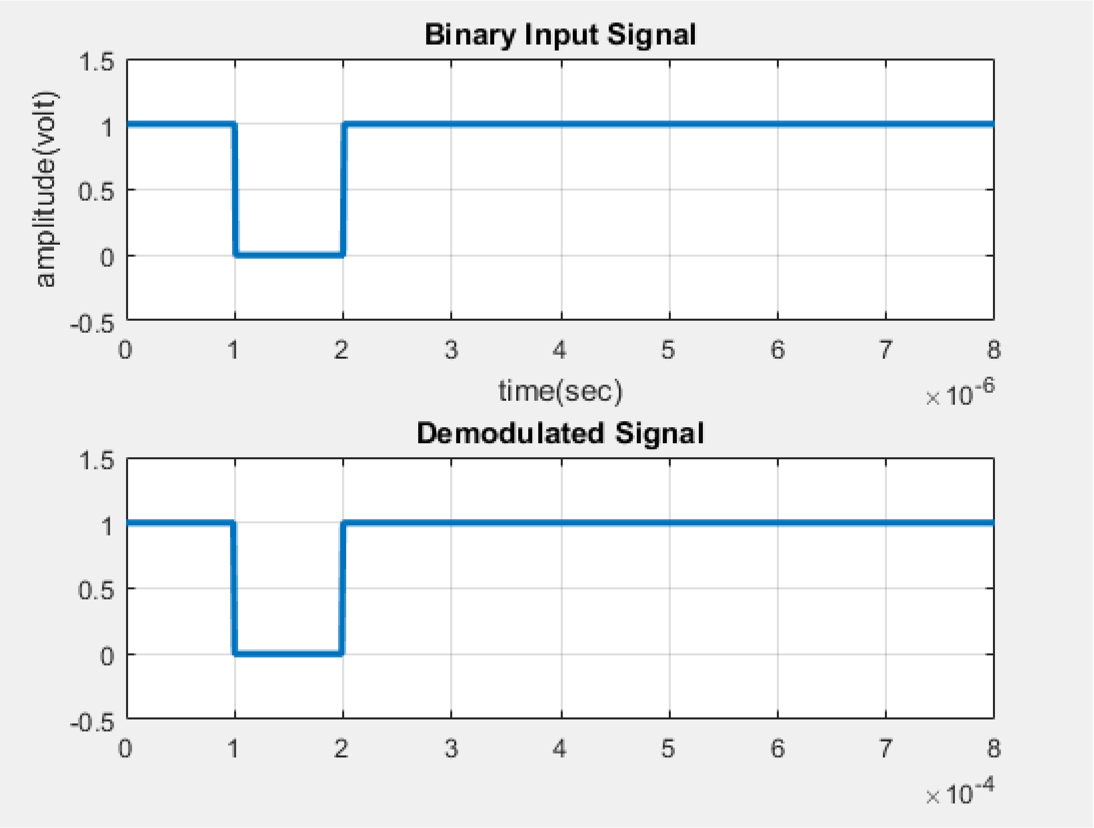
1st Run:



2[nd] Run:



3[rd] Run:



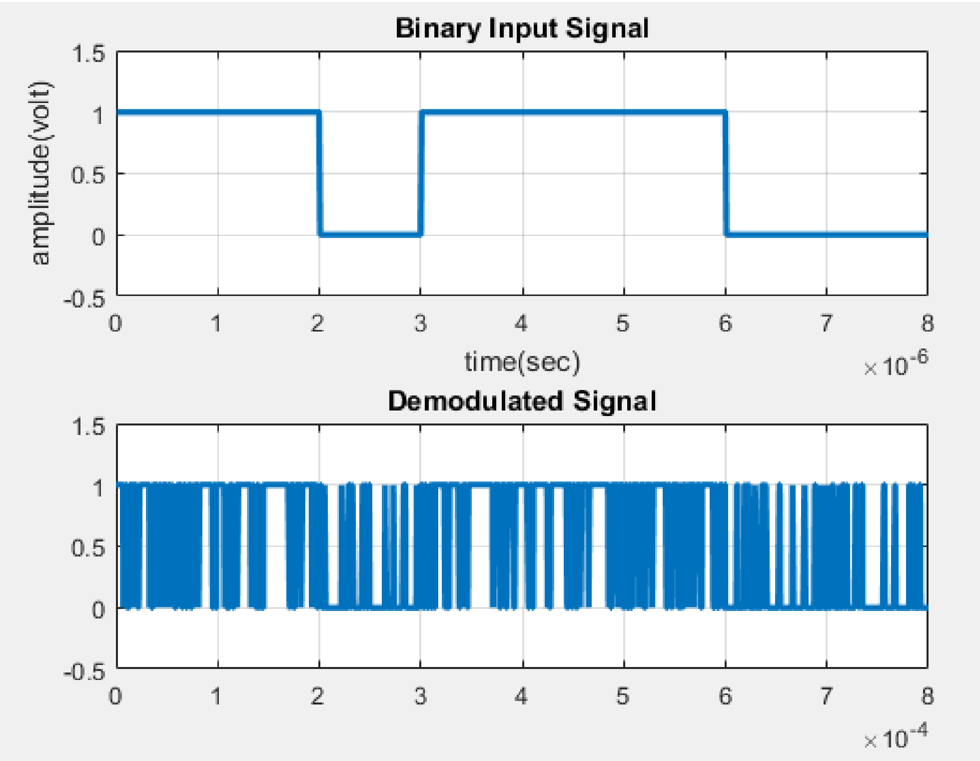
Comments: It is shown in graph that by doing Binary Frequency Shift Keying (BFSK) modulation and demodulation, the output at receiver is exactly same as the input at transmitter. It also shows that using built-in command in MATLAB we can get better results.

Now we will add Gaussian Noise after Modulation and see the effect.

Adding Gaussian Noise:

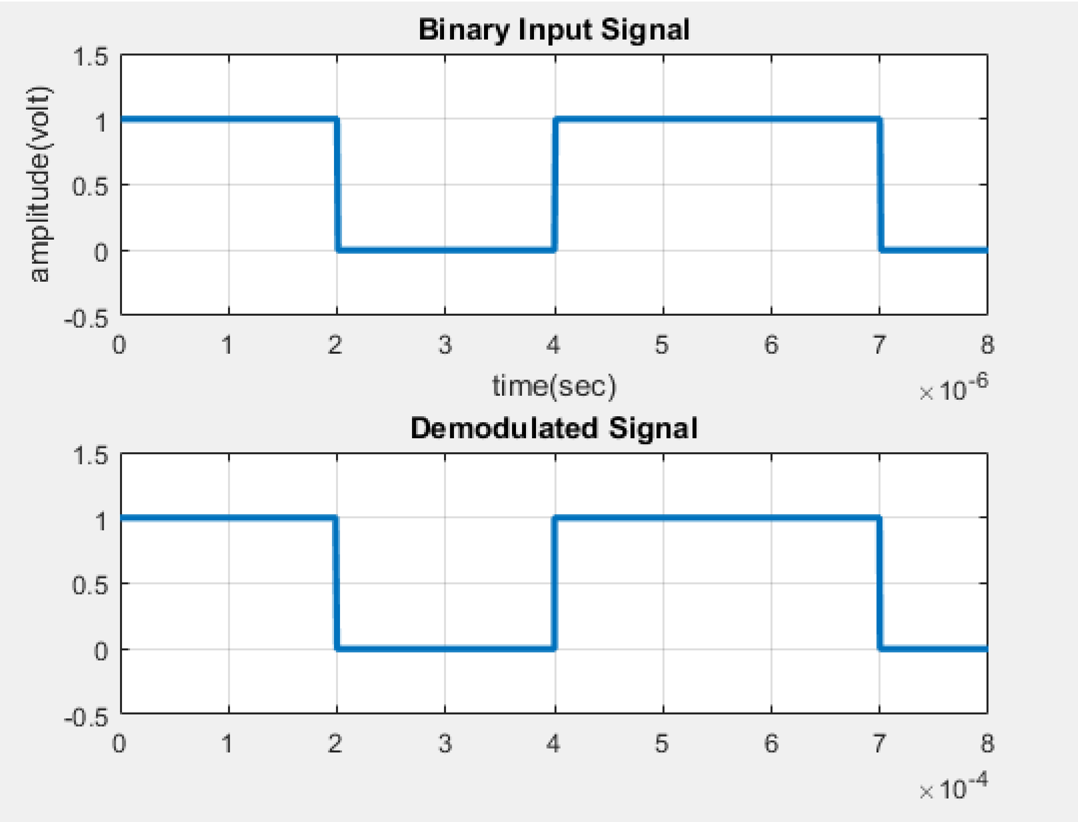
```
%---------------------------------------------------
%%%%%%%%%%%%%%% Adding Gausian Noise %%%%%%%%%%%%%%%

xmodfask = awgn(xmodfask,10);

-
```

Plot with Noise:



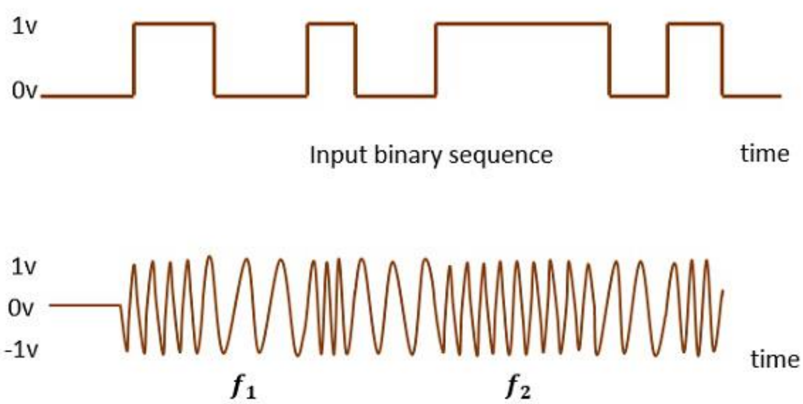
Comments: The SNR has affected the input signal after modulation and the resulted output is noisy. If the SNR is increased, we get better output as shown in figure below:

Plot High SNR = 25:



Discussion:

Frequency Shift Keying (FSK) is a type of digital modulation technique in which the frequency of the carrier signal varies according to the discrete digital changes. In this modulation scheme, when the amplitude of binary input signal is high, the frequency of FSK Modulated signal is high. Similarly when the binary input signal is low, the frequency of FSK Modulated signal is low. The BFSK figure is shown below [2]:



FSK Modulated output wave

In this scheme, the binary **1s** and **0s** are called **Mark** and **Space** frequencies respectively **[2].**

Hence, using this modulation technique, we can transmit and receive our signal through a digital communication system.

In our code, we have used a binary signal for which the demodulated output produce exactly same results. Using built-in command for BFSK modulation and demodulation, we can transmit this binary signal without any loss of information. So the error probability rate is also low in BFSLK modulation scheme. In other words, error-free reception is possible with FSK **[3].**

It is also shown in graph that the amplitude variation is zero in Binary FSK which produces good results as compared to BASK. We have also analyzed the SNR effect on the resulted demodulated output which shows that it has High SNR (signal to noise ratio) because as compared to BASK, the low SNR value produces better results in case of BFSK as seen above. This binary scheme can be useful in low-speed digital systems where high-frequency communication is required. **[3]**

However, we see that the bandwidth requirement is high for the BFSK as compared to the ASK and PSK **[3].** Because of this large bandwidth requirement, this modulation scheme is used in low-speed digital systems.

Conclusion:

At the end of this project, we conclude that we can use MATLAB to develop a complete modulation and demodulation technique whether it is FSK, PSK, or ASK for digital communication system. We have observed that every modulation technique has its own advantages are disadvantages. So there is trade off to use a particular modulation technique in transmission and communication of signals.

Namal Institute Mianwali

References:

[1] *fskmod MATLAB.* [Online].
Available at: Frequency shift keying modulation - MATLAB fskmod (mathworks.com)

[2] *Digital Modulation Techniques.* [Online].
Available at: Digital Modulation Techniques (tutorialspoint.com)

[3] *Frequency Shift Keying (FSK) Working & Applications*. [Online].
Available at: Frequency Shift Keying (FSK) :Working, Advantages and Disadvantages (elprocus.com)