INSTITUTE OF ENGINEERING
CENTRAL CAMPUS,PULCHOWK

COMMUNICATION SYSTEM II

LAB #2

Sampling, TDM & Demodulation

Submitted BY:
AMRIT PRASAD PHUYAL
Roll: PULL074BEX004

Submitted To:
Suman Sharma
Department of Electronics and
Computer Engineering

August 8, 2021

Table of Contents

List of MATLAB codes

List of Figures

1 Title

Sampling, TDM & Demodulation

2 Objective

- To observe Signal Reconstruction using Samples
- To observe Time Division Multiplexing of two signals
- To observe Delta Modulation and its bits sequence

3 Theory

3.1 Sampling and Signal Reconstruction

Sampling is the process of taking a continuous signal and converting it into discrete signal. In other words, sampling is measuring the instantaneous value of continuous time signal.If $s(t)$ is a continuous time signal, then, the sampled discrete time signal having Sampling period T_s, denoted by $s[n]$, is given by:

$$s[n] = s(nT_s)$$

Interplotation technique is used to interpolate the sampled signal in order to recover the continuous time signal.

3.2 Time Division Multiplexing (TDM)

Time Division Multiplexing is a technique of multiplexing two signals in a single signal. Here each signal is allocated a definite amount of time such that the transmission appear to be parallel.

3.3 Delta Modulation

Delta Modulation is a technique of encoding a signal using a single bit. The signal is encoded using a single bit, which is either a 1 or a 0. The encoding is done by adding a delta to the original signal. The delta Δ is a small amount of the original signal. The delta is added to the original signal and the resulting signal is the encoded signal. It create stair-case approximation of output waveform. Delta modulation is deployed when transmission of voice is needed and quality is not the primary concern.

4 Problems

4.1 Signal Reconstruction using Sampling

```
clear all;
t=0:0.1:20;
f1=0.6;
f2=0.8;
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);

figure(1);
subplot(2,1,1);
plot(t,x);
hold on;
plot([0 20],[0 0],'black');
hold off;

title('Original signal');
xlabel('Time');
ylabel('Amplitude');

subplot(2,1,2);
x_samples = x(1:6:201);
stem(x_samples,'filled');
title('Sampled Signal');
xlabel('n');
ylabel('x_s[n]');
axis([0 20 -2 2]);
sgtitle('(PUL074BEX004)');


figure(2);
x_recon=0;
subplot(2,1,1);
stem(x_samples,'filled');
title('Sampled Signal');
xlabel('n');
ylabel('x_s[n]');

for k=0:length(x_samples)-1
    subplot(2,1,2);
stem(0:length(x_samples)-1,x_samples,'filled');
if k==length(x_samples)-1
title('Reconstruction finished');
else
title('Sample by sample reconstruction');
end
sgtitle('(PUL074BEX004)');
grid on;

l=k:-0.1:-20+k;
x_recon=x_recon+(x_samples(k+1)*sinc(l));
axis([0 20 -2 2]);
hold on;
plot(t,x_samples(k+1)*sinc(l),'r');
plot(t,x_recon,'b');
xlabel('Time, n');
ylabel('Amplitude, x_s[n]');
hold off;
waitforbuttonpress;
end
```

Code 1: MATLAB code Signal Reconstruction using Sampling

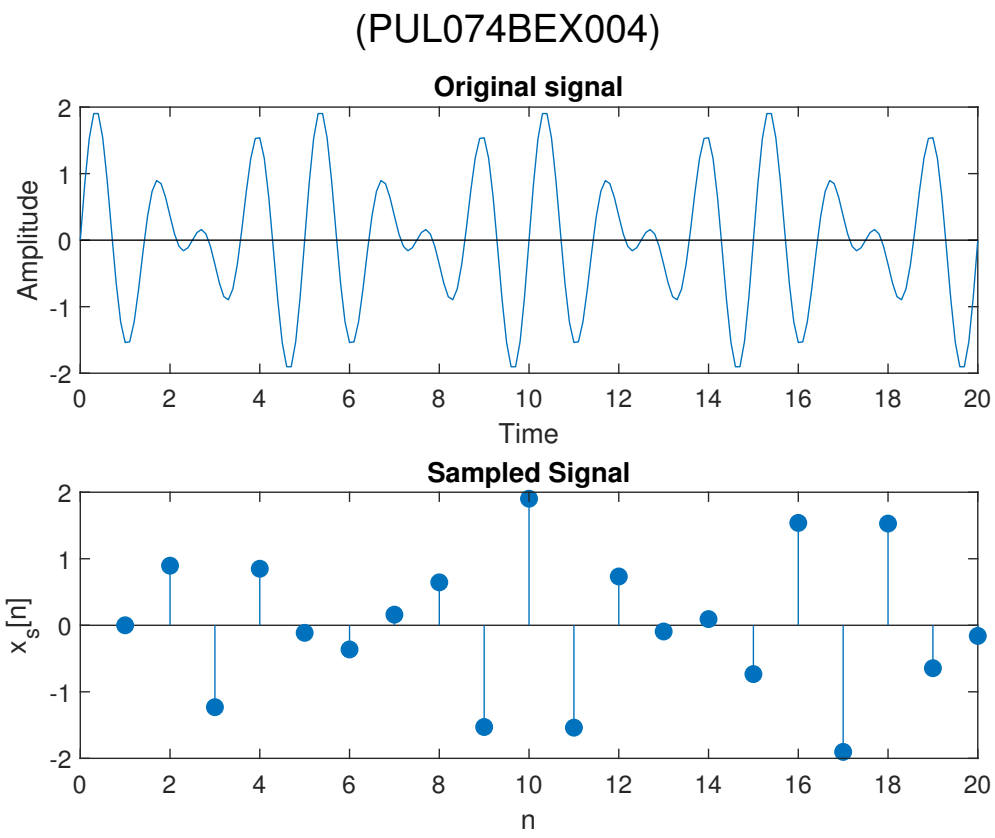

Figure 1: Orginal signal and its sampling

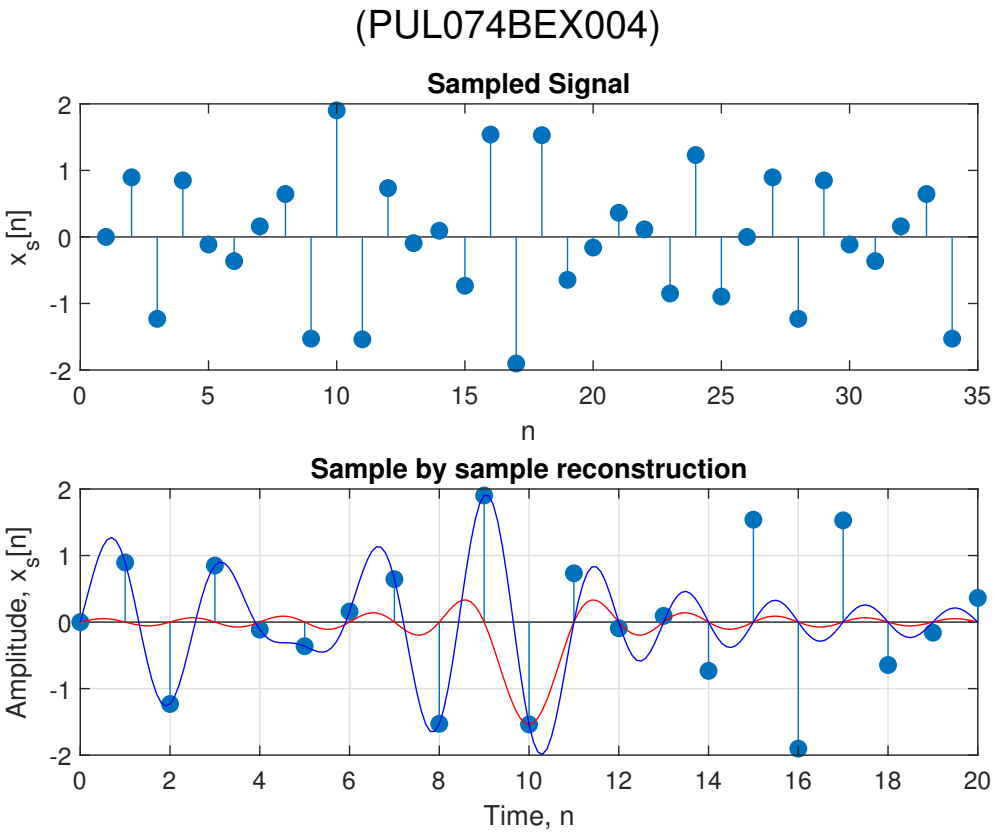

Figure 2: Sampled Signal and its Reconstruction Step at n=10

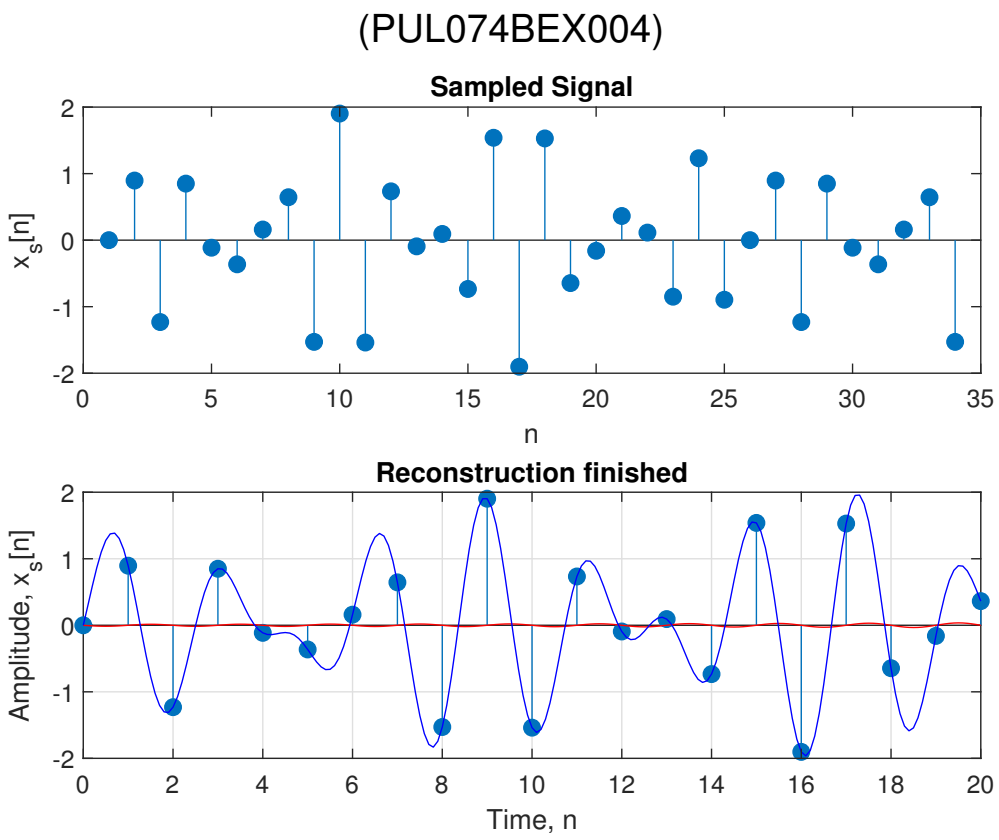

Figure 3: Sampled Signal and final Reconstructed Signal

4.2 Time Division Multiplexing (TDM)

```
clc;
close all;
clear all;

% Signal generation
x=0:0.2:4*pi; % siganal taken upto 4pi
sig1=8*sin(x); % generate 1st sinusoidal signal
l=length(sig1);
sig2=8*triang(l); % Generate 2nd traingular Sigal

% Display of Both Signal
subplot(2,2,1);
plot(sig1);
title('m_1(t) ');
xlabel('t');
ylabel('Amplitude');

grid on

subplot(2,2,2);
plot(sig2);
title('m_2(t) ');
xlabel('t');
ylabel('Amplitude');

grid on

% Display of Both Sampled Signal
subplot(2,2,3);
```

```
30
31  stem(sig1);
32  title('Sampled m_1(t) ');
33  xlabel('t');
34  ylabel('x_1[n]');
35  subplot(2,2,4);
36
37  stem(sig2);
38  title('Sampled m_2(t)');
39  xlabel('n');
40  ylabel('x_2[n]');
41  l1=length(sig1);
42  l2=length(sig2);
43  for i=1:l1
44  sig(1,i)=sig1(i); % Making Both row vector to a matrix
45  sig(2,i)=sig2(i);
46  end
47
48  % TDM of both quantize signal
49  tdmsig=reshape(sig,1,2*l1);
50
51  % Display of TDM Signal
52  figure
53  stem(tdmsig);
54  title('Multiplexed Signal ');
55  xlabel('n');
56  ylabel('Amplitude');
57
58  % Demultiplexing of TDM Signal
59  demux=reshape(tdmsig,2,l1);
60  for i=1:l1
61  sig3(i)=demux(1,i); % Converting The matrix into row vectors
62  sig4(i)=demux(2,i);
63  end
64
65  % display of demultiplexed signal
66  figure
67  subplot(2,1,1)
68  plot(sig3);
69  title('Demultiplexed m_1(t)');
70  xlabel('t');
71  ylabel('Amplitude');
72  subplot(2,1,2)
73  plot(sig4);
74  title('Demultiplexed m_2(t)');
75  xlabel('t');
76  ylabel('Amplitude');
```

Code 2: Time Division Multiplexing (TDM)

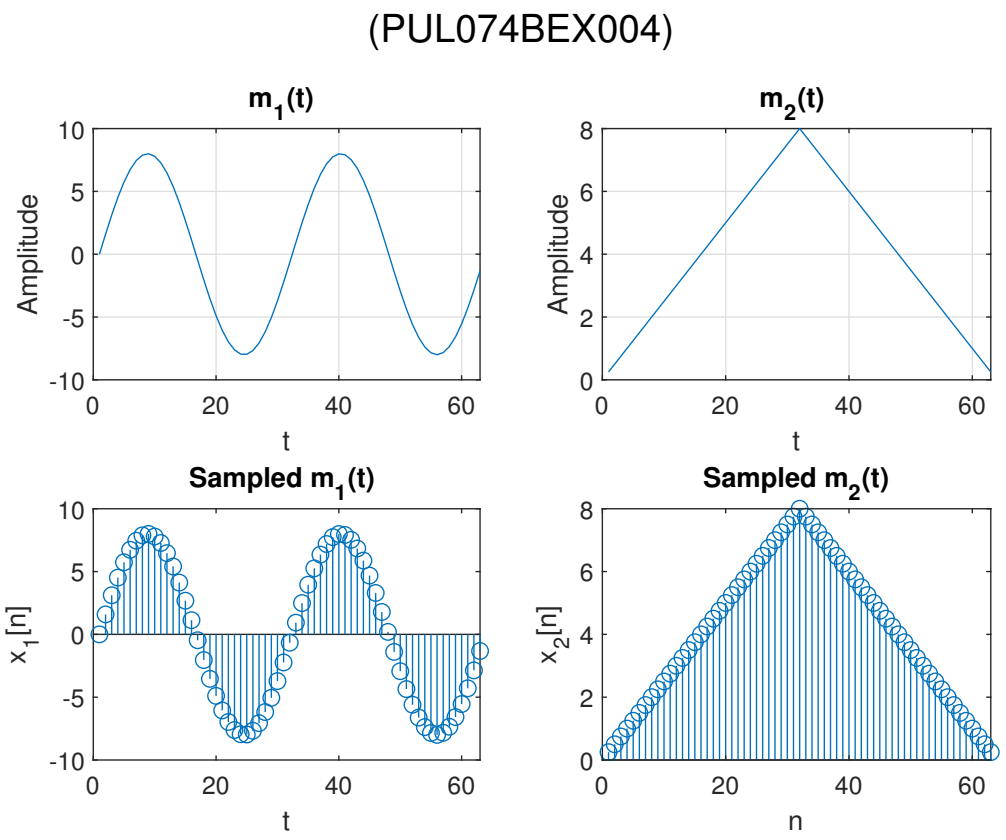

Figure 4: Orginal signals and their sampled form

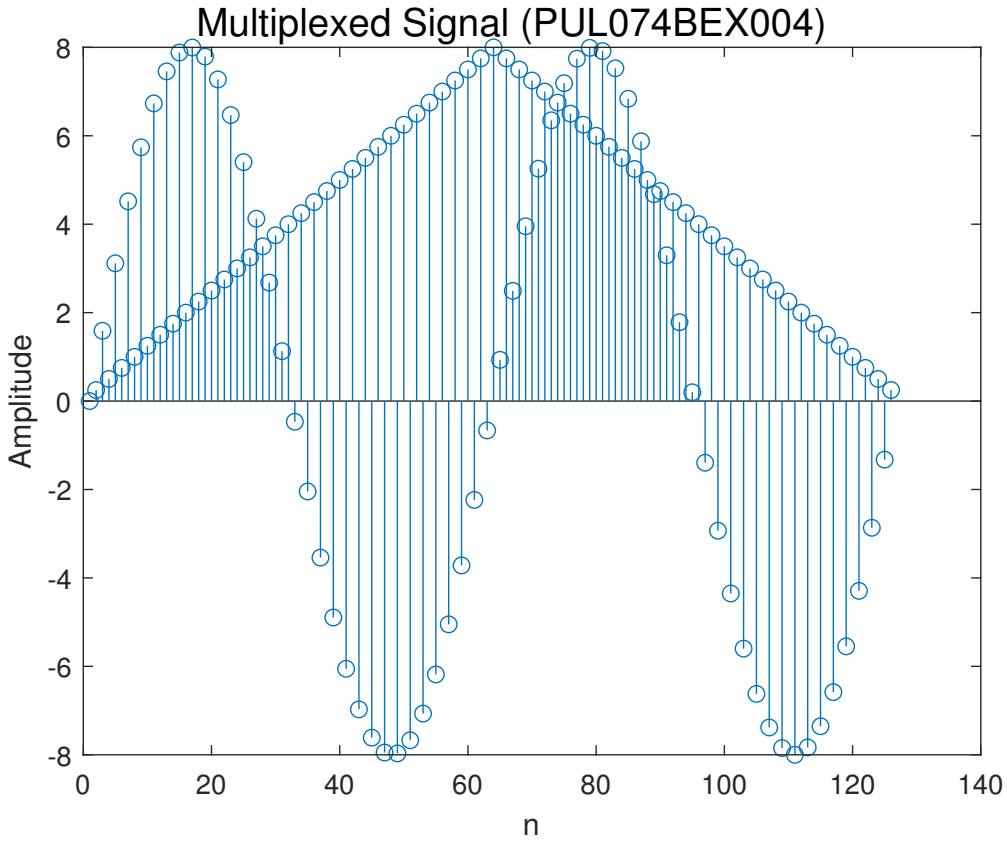

Figure 5: Multiplexed Signal

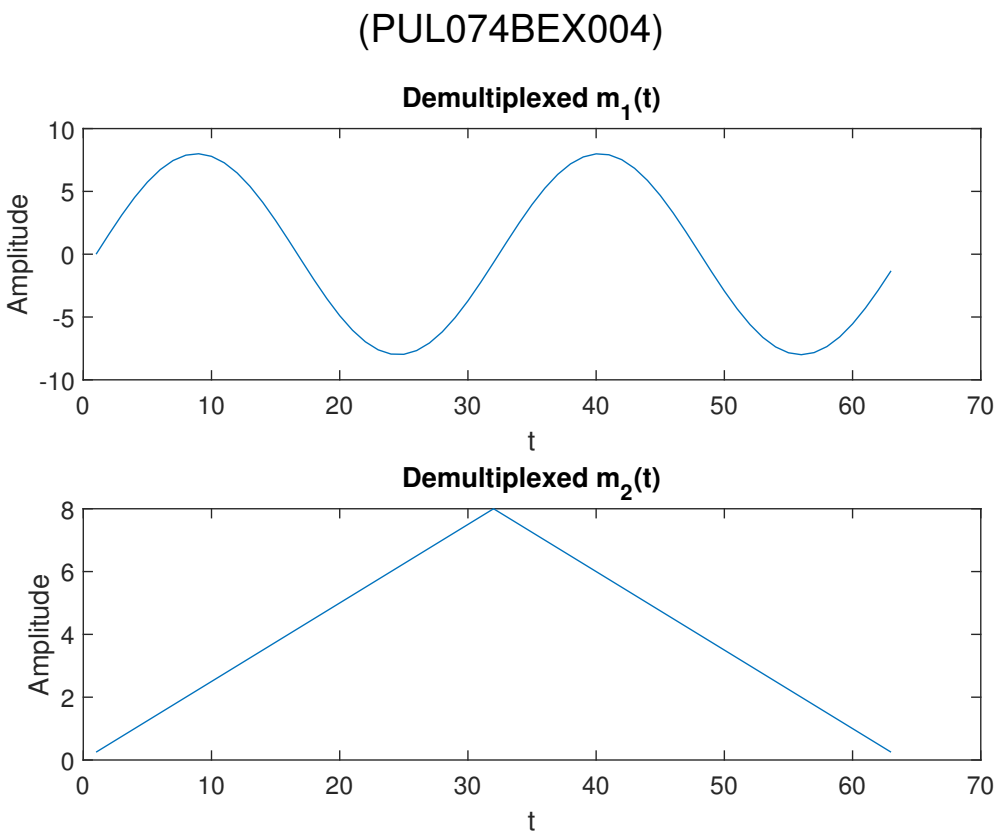

Figure 6: Demultiplexed Signals

4.3 Delta Modulation

```
del=input('Enter the step size = '); %step size
t=0:2*pi/100:2*pi;
x=2*sin(t); % amplitude of signal is 5
subplot(2,1,1)
plot(x)

hold on
y=[0]; % output binary sequencce
xr=0;
for i=1:length(x)-1
if xr(i)<=x(i)
d=1;
xr(i+1)=xr(i)+del;
else
d=0;
xr(i+1)=xr(i)-del;
end

y=[y d];
end

stairs(xr)
title('Delta Modulation ')
xlabel('Time (t)')
ylabel('Amplitude')
axis( [0 105 -3 3 ] )
subplot(2,1,2)
stairs(y)
title('Bit Output ')
```

```
30  xlabel('Time (t)')
31  ylabel('Amplitude')
32  axis( [0 105 0 1.5] )
33  sgtitle('(PUL074BEX004)');
34  hold off
```

Code 3: Delta Modulation and its bits sequence

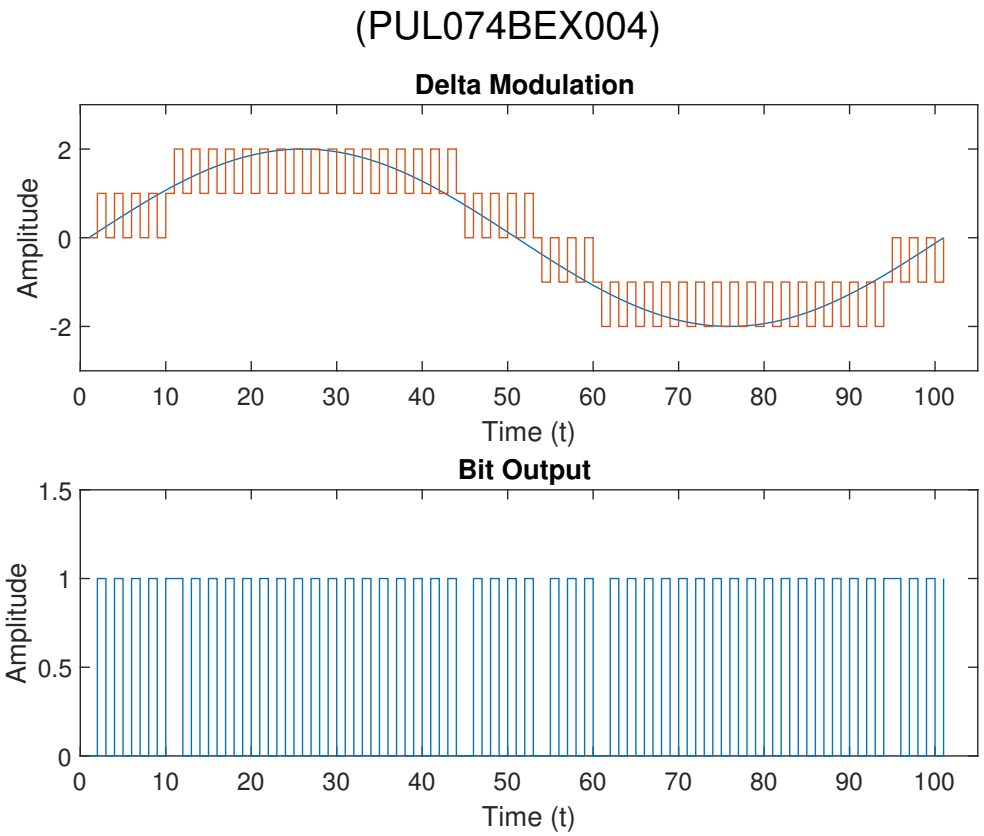

Figure 7: Delta Modulation and its bits sequence

5 Discussion and Conclusion

In this Lab we performed the Sampling of original signal and recover the measure signal from its sampled version. We also performed Time Division Multiplexing of two signals and Demultiplex the signals to recover the originals. We also performed Delta Modulation of a signal and its bits sequence. We used MATLAB, its modules and functions to perform the above tasks.