SCHOOL OF ELECTRICAL ENGINEERING

COMMUNICATION ENGINEERING (EEE2006)

NAME : SHANT RAKSHIT
REG. NO. : 19BEE0154
SLOT : L 55+56

Experiment 4
Date : 08/10/21, 15/10/21

1. Generate and demodulate PAM using pspice and also demonstrate the generation of PAM using MATLAB involving the concepts of natural sampling, impulse sampling, and flat top sampling.

2. Generate and demodulate PWM with pspice and MATLAB. Also show the effects of Noise in PWM

3. Generate PPM modulation and demodulation. Also show the variations of demodulation with noise.

Theory Questions.

1. What are the differences you observed between these pulse modulations techniques.

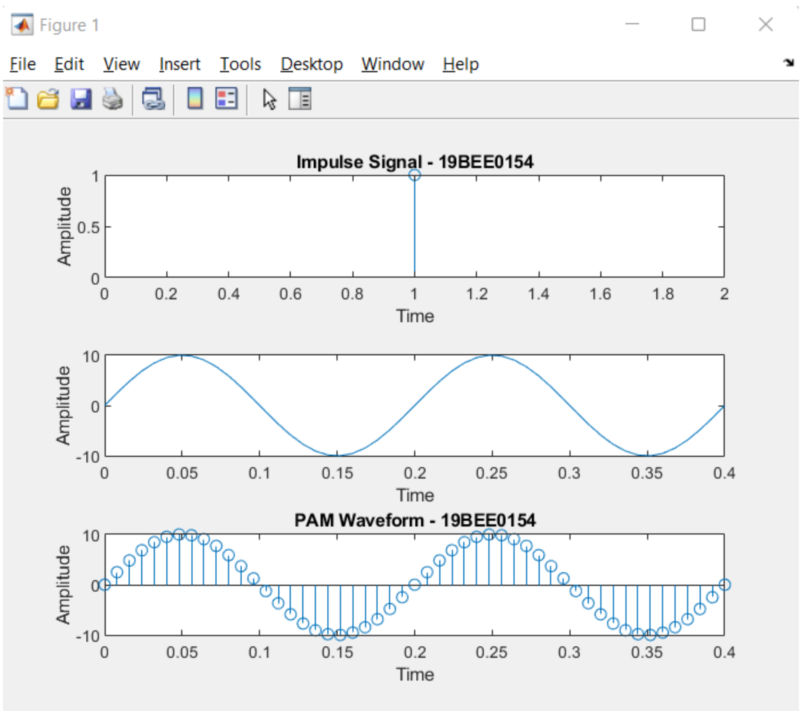
2. Define Nyquist sampling theorem and show how the spectrum is plotted for each of these modulations.
3. List out the practical applications of each of these modulations.

Generation and Demodulation PAM by Impulse Sampling

MATLAB Code

```
clc;
Am = input('Enter the amplitude:');
fm = input('Enter frequency:');
fs = 25*fm;
t = 0:1/fs:2/fm;
x1 = 1;
xm = Am*sin(2*pi*fm*t);
y = x1.*xm;
subplot(3,1,1);
stem(x1);
xlabel('Time');
ylabel('Amplitude');
title('Impulse Signal - 19BEE0154');
subplot(3,1,2);
title('Sinusoidal Signal - 19BEE0154');
plot(t,xm);
ylim([-Am-0.2 Am+0.2])
xlabel('Time');
ylabel('Amplitude');
subplot(3,1,3);
stem(t,y);
title('PAM Waveform - 19BEE0154');
xlabel('Time');
ylabel('Amplitude');
ylim([-Am-0.2 Am+0.2])
```

Simulation



Generation and Demodulation PAM by Natural Sampling

MATLAB Code

```
clc;
close all;
clear all;

fm = input('Enter the message frequency = ');
fc = input('Enter the carrier frequency = ');

t=0:0.001:1;
fs=1000;
m=cos(2*pi*fm*t);
c=0.5*square(2*pi*fc*t)+0.5;
s=m.*c;
dt=s.*c;
filter=fir1(200,fm/fs,'low');
d=conv(filter,dt);
l=length(d);
t1=linspace(0,1,l);

subplot(4,1,1);
plot(t,m,'r');
title('Message Signal 19BEE0154');
xlabel('Time');
ylabel('Amplitude ');

subplot(4,1,2);
plot(t,c,'g');
title('Carrier Signal 19BEE0154');
xlabel('Time');
ylabel('Amplitude ');

subplot(4,1,3);
plot(t,s,'b');
title('PAM Signal 19BEE0154');
xlabel('Time');
ylabel('Amplitude ');

subplot(4,1,4);
plot(t1,d,'m');
title('Demodulated Signal 19BEE0154');
xlabel('Time');
ylabel('Amplitude ');
```

Simulation

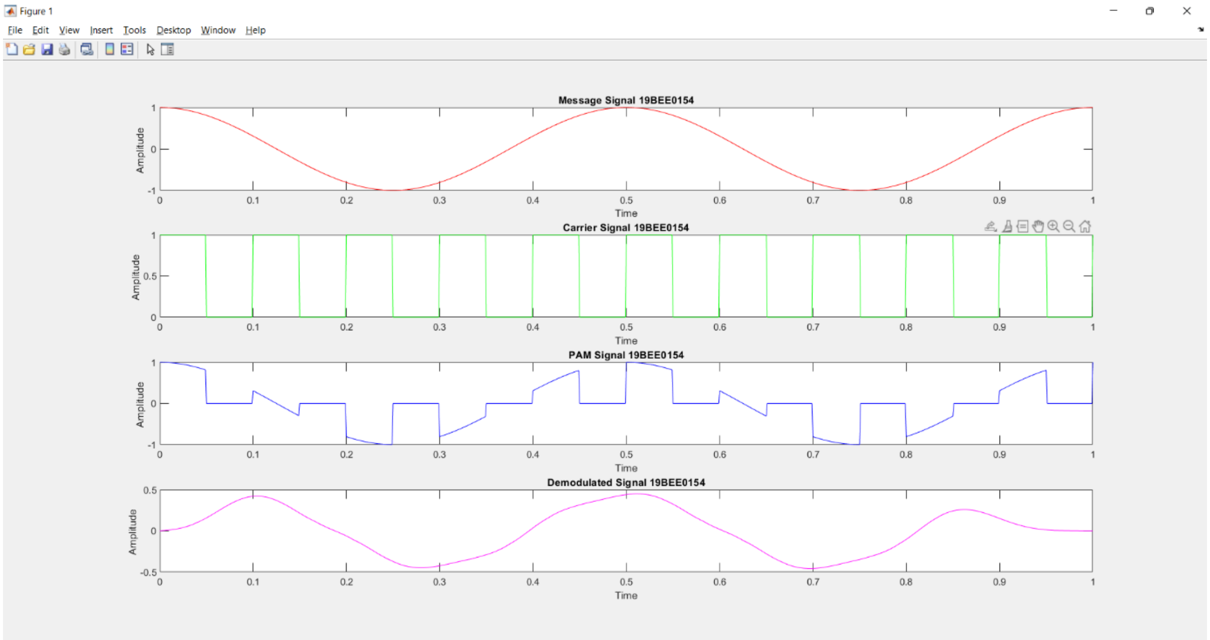


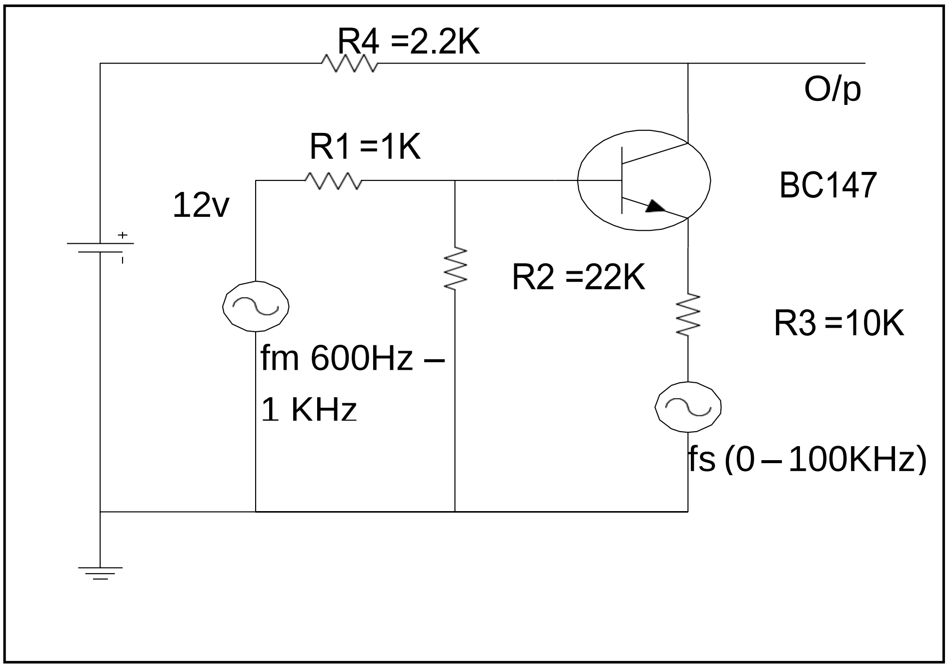
Figure 1
File Edit View Insert Tools Desktop Window Help
Message Signal 19BEE0154
Amplitude
Time
Carrier Signal 19BEE0154
Amplitude
Time
PAM Signal 19BEE0154
Amplitude
Time
Demodulated Signal 19BEE0154
Amplitude
Time

Generation and Demodulation PAM using Pspice

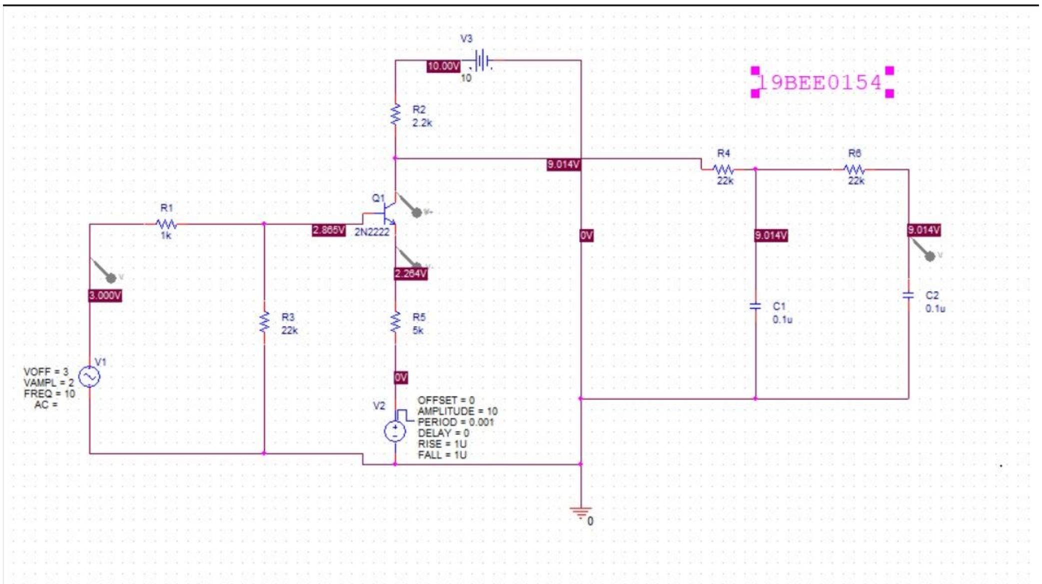
PROCEDURE

- Connections are made as per the circuit diagram.
- Apply a modulating signal with 1 KHz frequency and carrier pulse signal with 100 KHz frequency from the function generator.
- Apply 15 V dc supply to the collector.
- Take the modulated signal output at the emitter and observe the same on CRO
- Note the readings of amplitude and frequency of PAM output signal

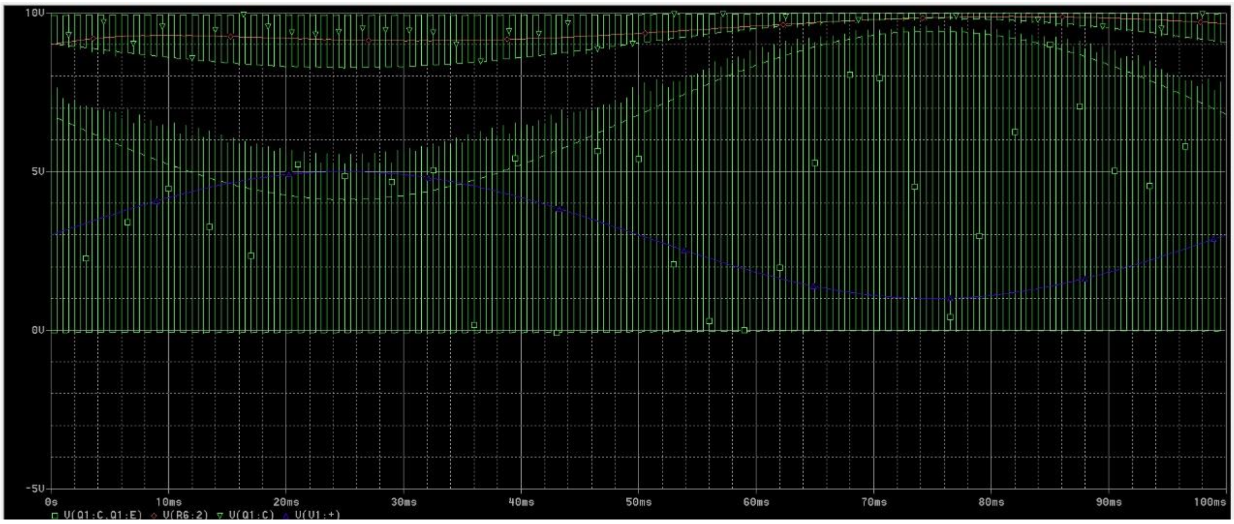
CIRCUIT DIAGRAM



Circuit on Pspice



Simulation



Inference

We have simulated the graphs of pulse amplitude modulated waves and its demodulation is also plotted. We have also made the Pspice circuit for PAM, and we also observed the practical graphs (input is with offset voltages). In graphs, we can see the variation of amplitude according to pulses.
Demodulation is also done properly.

Generation of PWM and PPM using MATLAB

MATLAB Code

```
% Generation of PWM and PPM
clc;
clear all;
close all;
fs = 1000;
fm = 1;
fc = 10;
vm = 2;
t = 0:1/fs:1;
message = vm*sin(2*pi*t);
subplot(4,1,1);
plot(t,message);
title("The modulating signal 19BEE0154");
vc = 3;
pulse = vc*sawtooth(2*pi*fc*t);
subplot(4,1,2);
plot(t,pulse);
title('Carrier Pulse 19BEE0154');
%Generation of PWm
n = length(pulse);
for i =1:n
 pwm(i)=0;
end
for i = 1:n
 if (message(i)>=pulse(i))
 pwm(i)=1;
 else
 pwm(i)=0;
 end
end
subplot(4,1,3)
plot(t,pwm);
title('PWM output 19BEE0154');
% Generation of PPM
PT = 1/fc;
CP = 0:1/fs:PT/4;
PL = length(CP);
for i =1:n
 ppm(i)= 0;
end
for i =1:n-1
 if (pwm(i)>pwm(i+1))
 for j = 1:PL-1
 ppm(i+j)=1;
 end
 end
end
subplot(4,1,4);
plot(t,ppm);
title('PPM output 19BEE0154');
```

Simulation

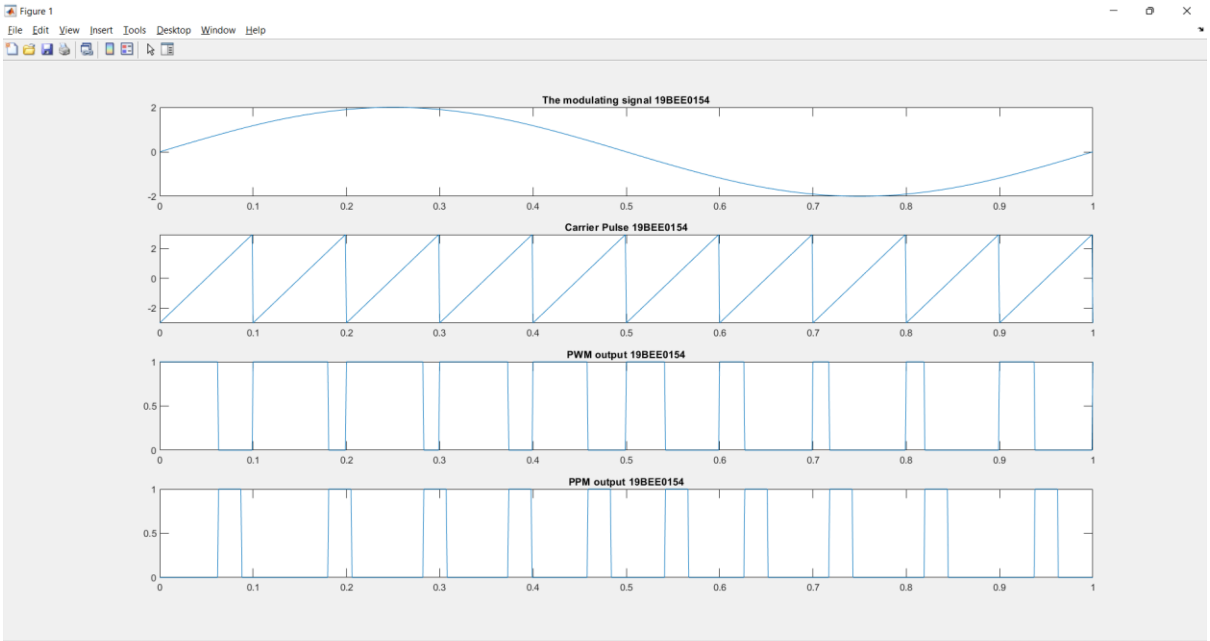


Figure 1
File Edit View Insert Tools Desktop Window Help
The modulating signal 19BEE0154
Carrier Pulse 19BEE0154
PWM output 19BEE0154
PPM output 19BEE0154

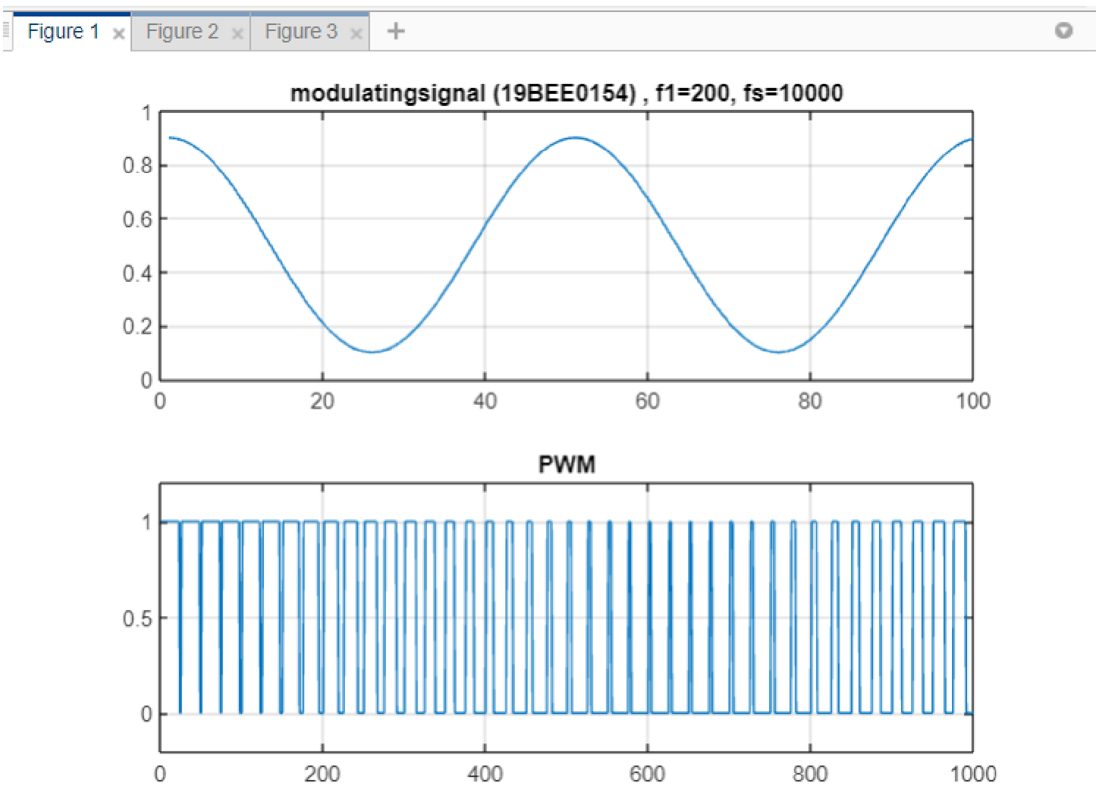
Generation of PWN and Recovery with the effect of Noise

MATLAB Code

```
%SHANT RAKSHIT
%19BEE0154
close all
clear all
clc
fc=400; %Carrier Frequency
fs=10000;%Sampling frequency
f1=200;%Pulse frequency
t=0:1/fs:((2/f1)-(1/fs));
x1=0.4*cos(2*pi*f1*t)+0.5;%Pulse signal
y1=modulate(x1,fc,fs,'PWM');%Modulated signal
y2=awgn(y1,0.8);
subplot(211)
plot(x1)
grid on;
axis([0 100 0 1]);
title('modulatingsignal (19BEE0154) , f1=200, fs=10000');
subplot(212)
plot(y1)
grid on;
axis([0 1000 -0.2 1.2]);
title('PWM');
grid on;
x1_recov=demod(y1,fc,fs,'PWM');
[den,num]=butter(1,2*pi*f1/fs);
figure(2);
subplot(311)
plot(x1_recov)
grid on;
title('Time domain recovered, single tone, f1=200');
axis([0 100 0 1]);
s11=filter(den,num,x1_recov);
subplot(312)
plot(s11)
grid on;
title('Filtered Output - 1 19BEE0154');
axis([0 100 0 1]);
s12=filter(den,num,s11);
subplot(313)
plot(s12)
grid on;
title('Filtered Output - 2 19BEE0154');
axis([0 100 0 1]);
x2_recov=demod(y2,fc,fs,'PWM');
[den1,num1]=butter(1,2*pi*f1/fs);
figure(3);
subplot(311)
plot(x2_recov)
grid on;
title('Time domain recovered(NOISY), single tone, f1=200');
```

```
axis([0 100 0 1]);
s111=filter(den1,num1,x2_recov);
subplot(312)
plot(s111)
grid on;
title('Filtered Output - 1 19BEE0154');
axis([0 100 0 1]);
s122=filter(den1,num1,s111);
subplot(313)
plot(s122)
grid on;
title('Filtered Output - 2 19BEE0154');
axis([0 100 0 1]);
```

Simulation



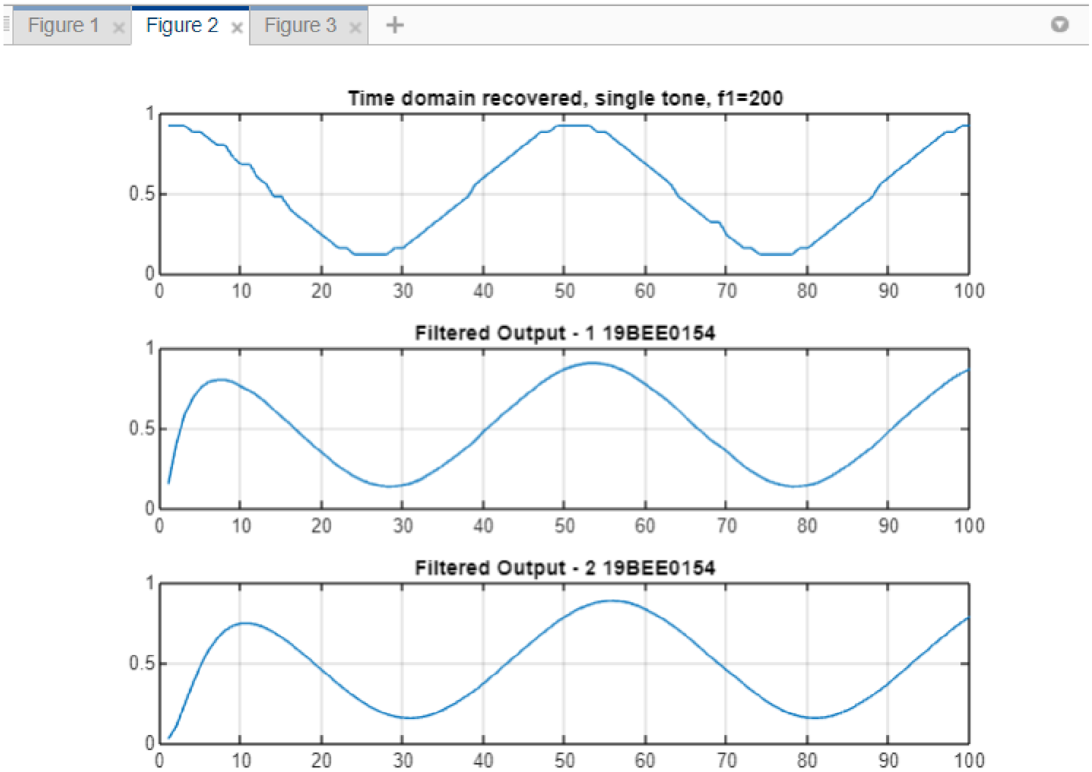


Figure 1
Figure 2
Figure 3
Time domain recovered, single tone, f1=200
Filtered Output - 1 19BEE0154
Filtered Output - 2 19BEE0154

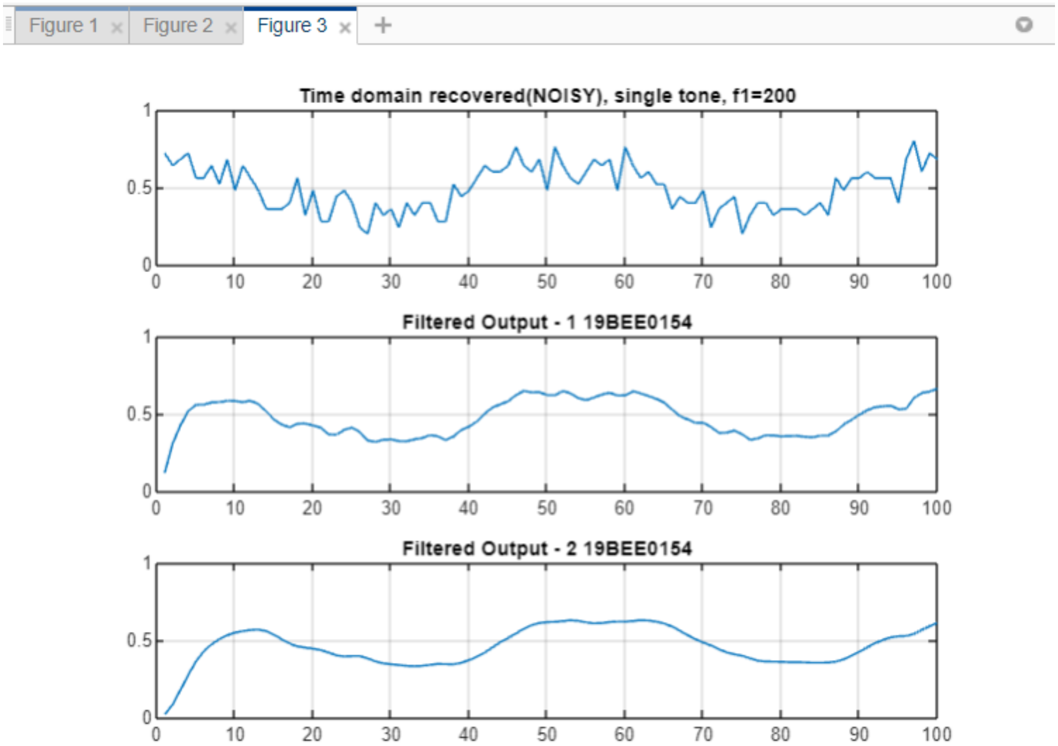


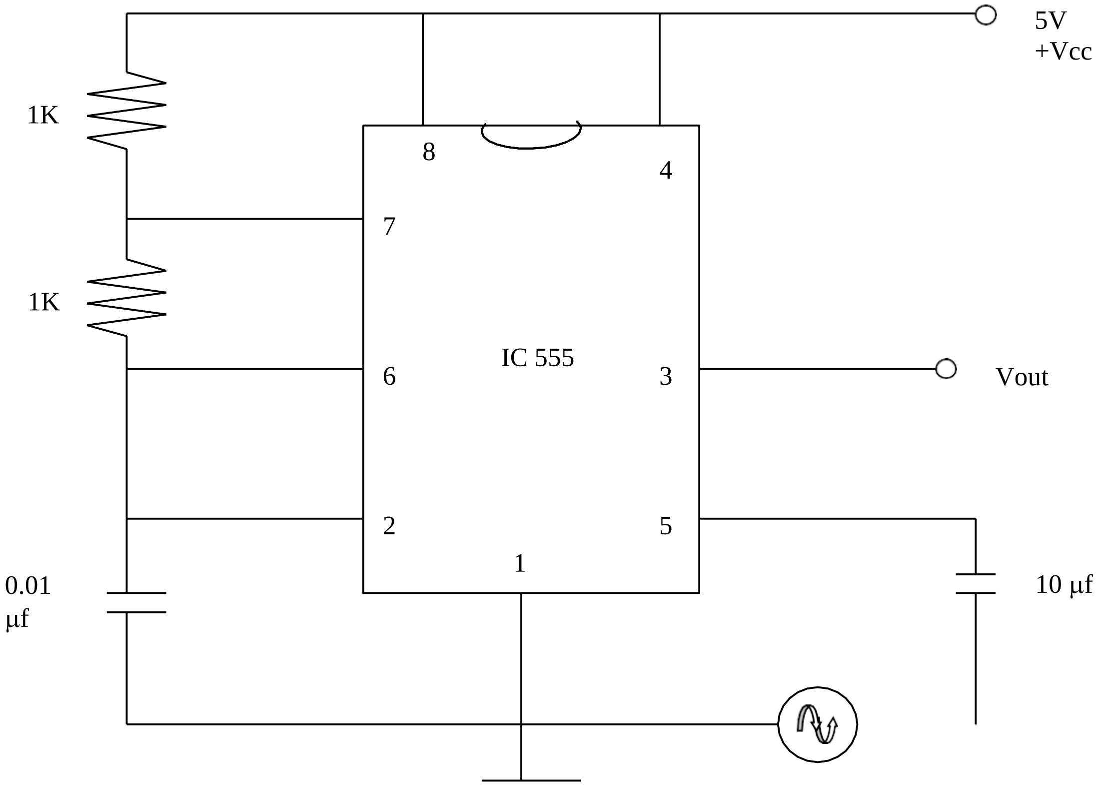
Figure 1
Figure 2
Figure 3
Time domain recovered(NOISY), single tone, f1=200
Filtered Output - 1 19BEE0154
Filtered Output - 2 19BEE0154

Generation of PWM and Demodulation using Pspice

PROCEDURE:

- Connect the circuit as per shown in the diagram
- Set the amplitude of the input signal 2volts
- Vary the input frequency from 100Hz to 5KHz to get the PWM output in pin no
- Plot the graph for both input and output signal.

Circuit Diagram



To get PWM Modulation

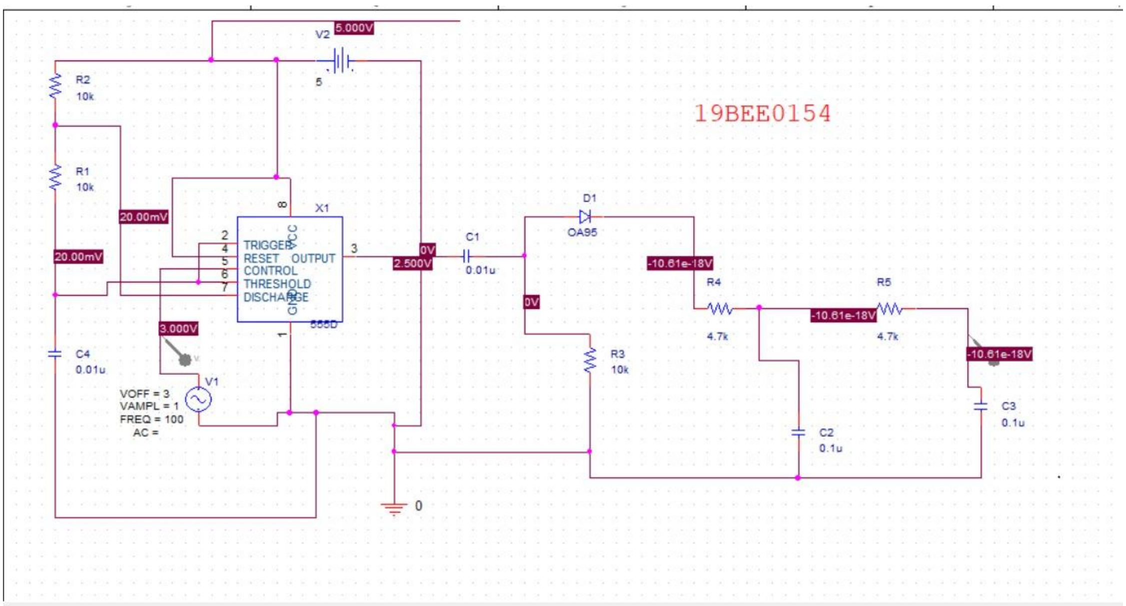
1. Connect the probe across message source and another probe at pin number 3 of 555 timer.
2. Run the simulation for 15 ms.

To get PWM Demodulation

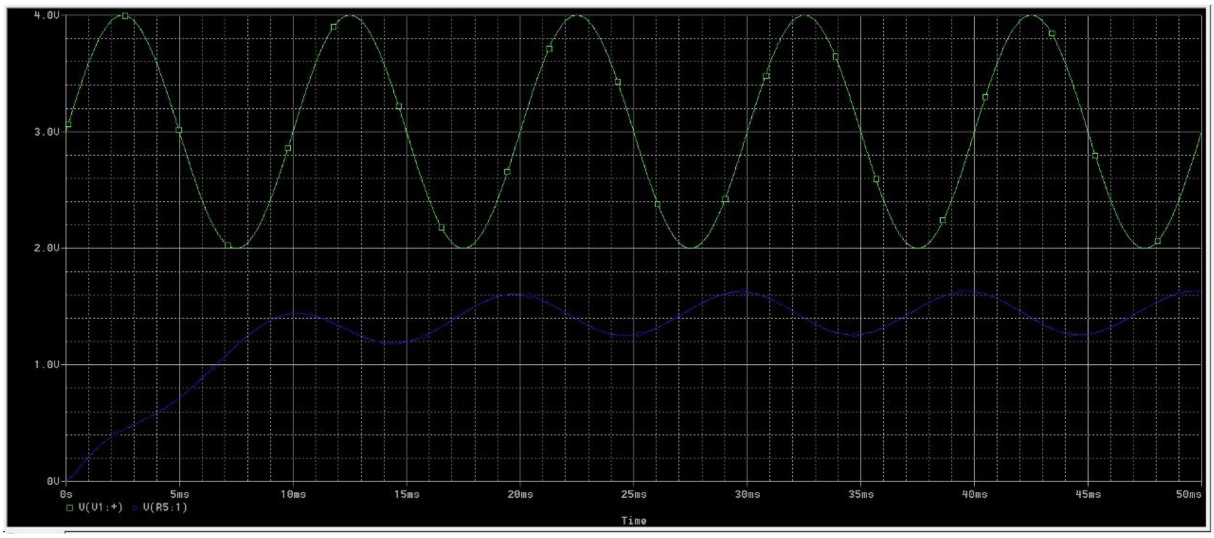
1. Connect the probe across message source and another probe at one end the capacitor(shown in the above diagram).

Run the simulation for 100 ms.

Pspice Circuit



Simulation



Inference

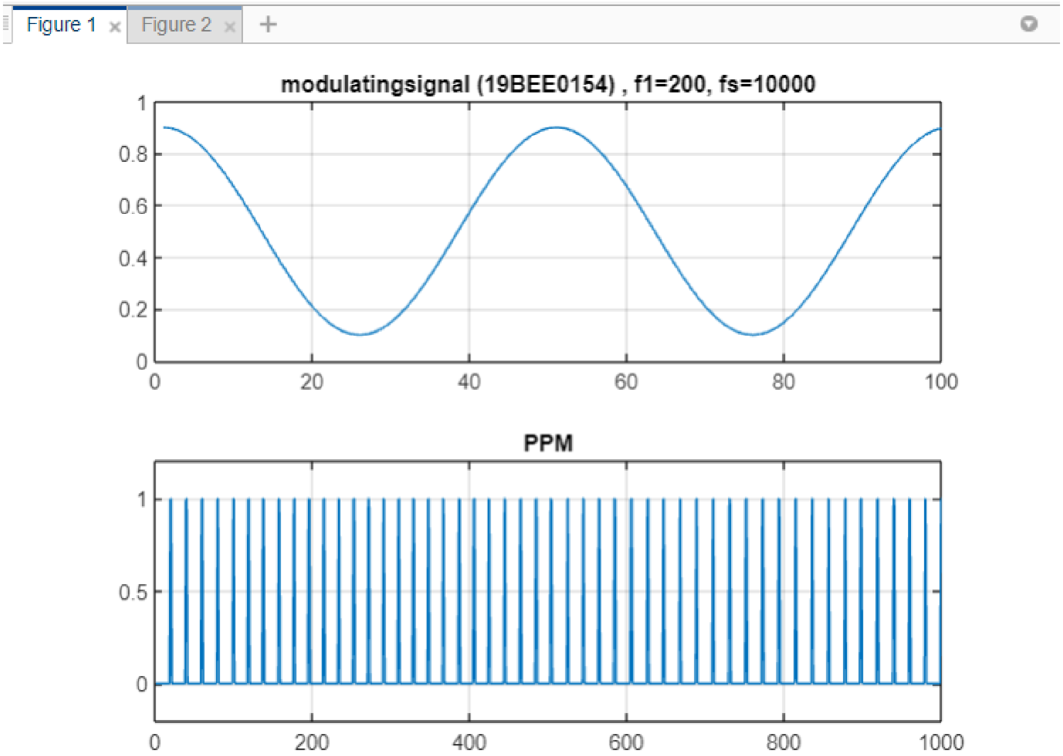
Here We have observed the PWM waves and also PPM wave using PWM pulses. We can observe the PWM pulse generation using MATLAB. We have also made the Pspice modulator (using 555 timer) and demodulator circuit and simulated the graphs which gives us a little realistic simulation. Also, we can observe the original signal filtration to get our original signal. Moreover, effect of noise is observed on demodulation. Thus, if any noise comes in way between transmitter and receiver, it can be demodulated to a fine extent. Yet we can see the disturbance.

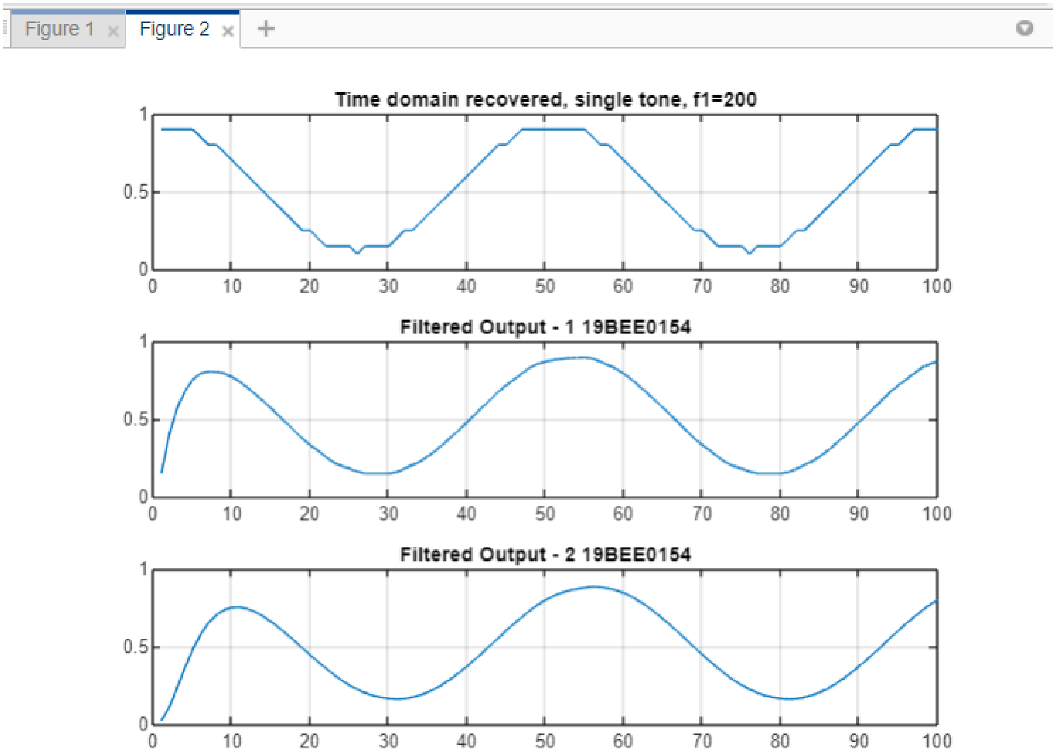
Generation of PPM and Demodulation

MATLAB Code

```
%19BEE0154
%Shant Rakshit
close all
clear all
clc
fc=500; %Carrier Frequency
fs=10000;%Sampling frequency
f1=200;%Pulse frequency
t=0:1/fs:((2/f1)-(1/fs));
x1=0.4*cos(2*pi*f1*t)+0.5;%Pulse signal
y1=modulate(x1,fc,fs,'PPM');%Modulated signal
subplot(211)
plot(x1)
grid on;
axis([0 100 0 1]);
title('modulatingsignal (19BEE0154) , f1=200, fs=10000');
subplot(212)
plot(y1)
grid on;
axis([0 1000 -0.2 1.2]);
title('PPM');
grid on;
x1_recov=demod(y1,fc,fs,'PPM');
[den,num]=butter(1,2*pi*f1/fs);
figure(2);
subplot(311)
plot(x1_recov)
grid on;
title('Time domain recovered, single tone, f1=200');
axis([0 100 0 1]);
s11=filter(den,num,x1_recov);
subplot(312)
plot(s11)
grid on;
title('Filtered Output - 1 19BEE0154');
axis([0 100 0 1]);
s12=filter(den,num,s11);
subplot(313)
plot(s12)
grid on;
title('Filtered Output - 2 19BEE0154');
axis([0 100 0 1]);
```

Simulation



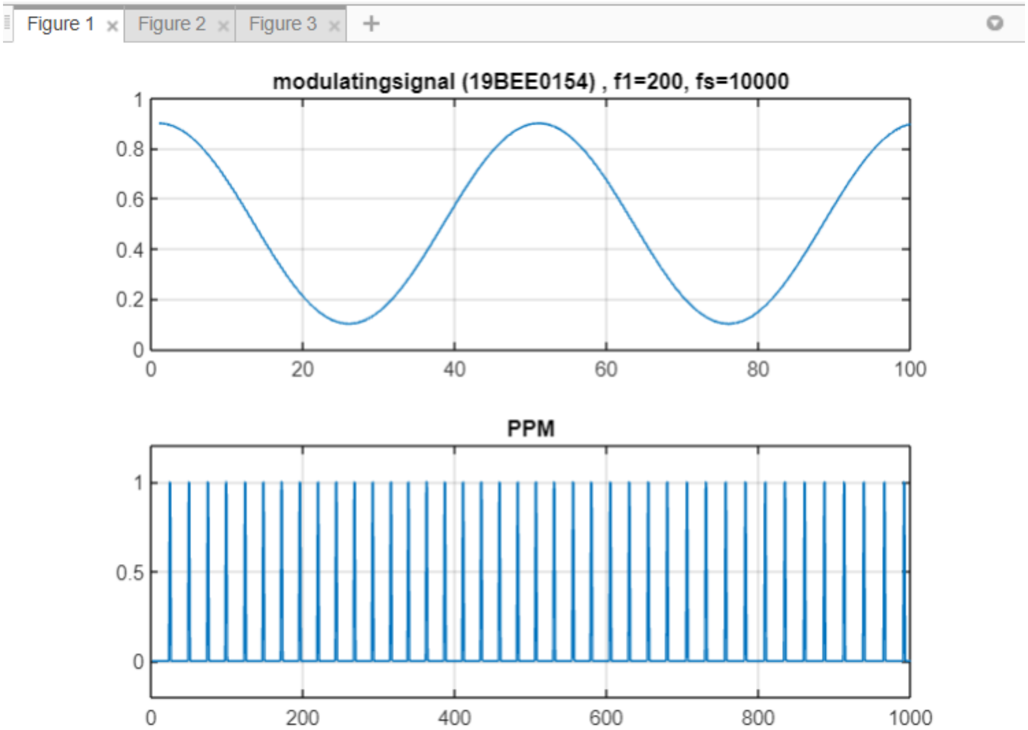


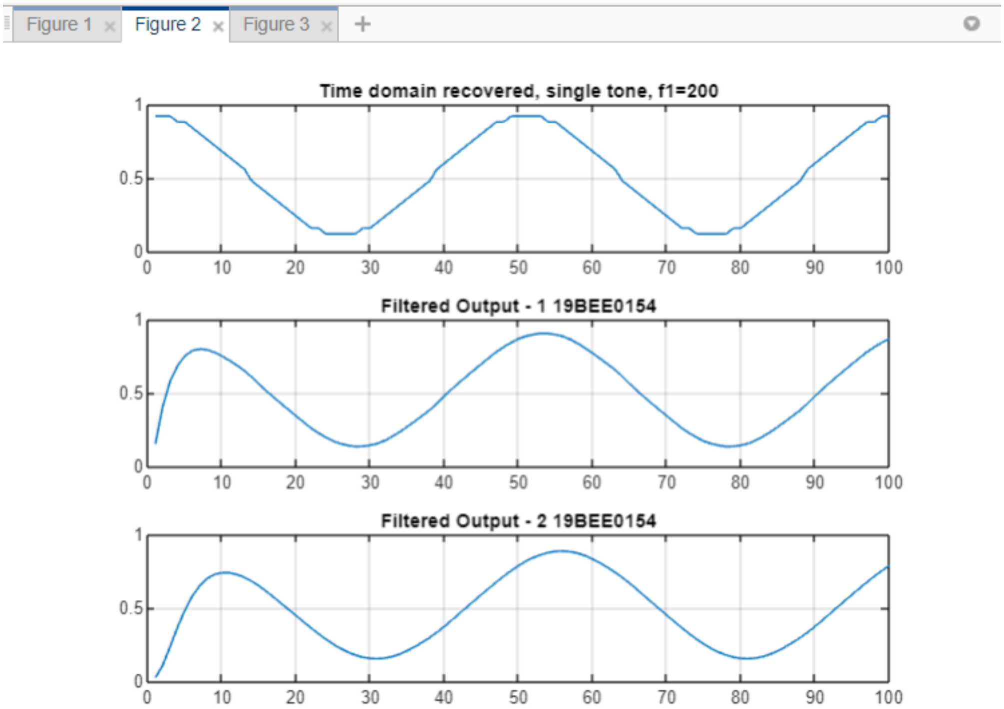
Generation of PPN and Recovery with the effect of Noise

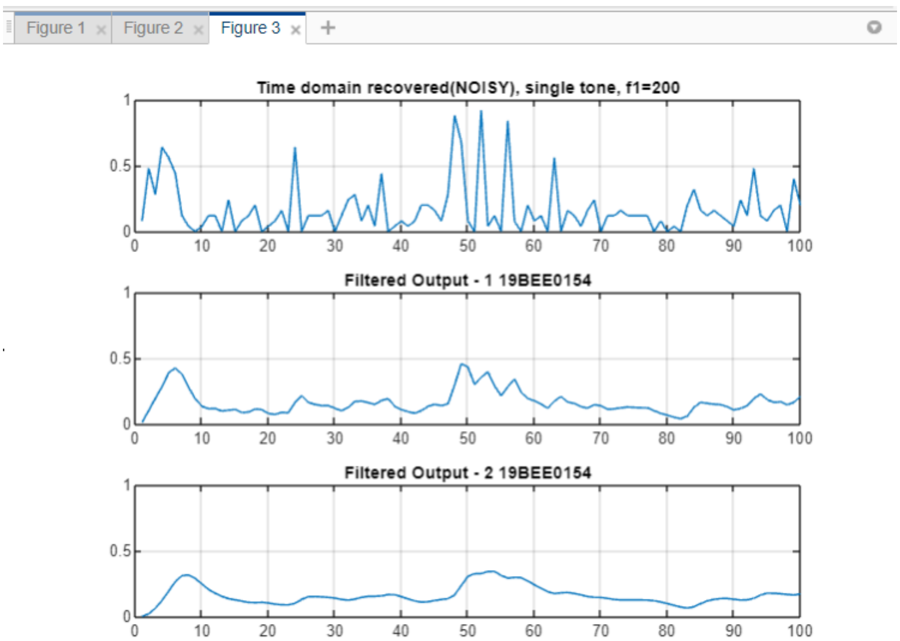
```
%19BEE0154
%Shant Rakshit
close all
clear all
clc
fc=400; %Carrier Frequency
fs=10000;%Sampling frequency
f1=200; %Pulse frequency
t=0:1/fs:((2/f1)-(1/fs));
x1=0.4*cos(2*pi*f1*t)+0.5;%Pulse signal
y1=modulate(x1,fc,fs,'PPM');%Modulated signal
y2=awgn(y1,5); %noisy signal

subplot(211)
plot(x1)
grid on;
axis([0 100 0 1]);
title('modulatingsignal (19BEE0154) , f1=200, fs=10000');
subplot(212)
plot(y1)
grid on;
axis([0 1000 -0.2 1.2]);
title('PPM');
grid on;
x1_recov=demod(y1,fc,fs,'PPM');
[den,num]=butter(1,2*pi*f1/fs);
figure(2);
subplot(311)
plot(x1_recov)
grid on;
title('Time domain recovered, single tone, f1=200');
axis([0 100 0 1]);
s11=filter(den,num,x1_recov);
subplot(312)
plot(s11)
grid on;
title('Filtered Output - 1 19BEE0154');
axis([0 100 0 1]);
s12=filter(den,num,s11);
subplot(313)
plot(s12)
grid on;
title('Filtered Output - 2 19BEE0154');
axis([0 100 0 1]);

%recovery of noisy signal
x2_recov=demod(y2,fc,fs,'PPM');
[den1,num1]=butter(1,2*pi*f1/fs);
figure(3);
subplot(311)
plot(x2_recov)
grid on;
title('Time domain recovered(NOISY), single tone, f1=200');
axis([0 100 0 1]);
s111=filter(den1,num1,x2_recov);
subplot(312)
plot(s111)
grid on;
```

```
title('Filtered Output - 1 19BEE0154');
axis([0 100 0 1]);
s122=filter(den1,num1,s111);
subplot(313)
plot(s122)
grid on;
title('Filtered Output - 2 19BEE0154');
axis([0 100 0 1]);
```

Simulation







<u>INFERENCE</u>

We have observed the PPM waves and also PPM wave using PWM pulses. We can observe the PWM pulse generation using MATLAB. And simulated the graphs. Also, we can observe the original signal filtration to get our original signal. Moreover, effect of noise is observed on demodulation. Thus, if any noise comes in way between transmitter and receiver, it can be demodulated to the neither extent. We can see the disturbance in the recovery. Here noise can’t be handled properly by this method.

Theory Questions

1. Differences observed between these pulse modulation techniques

   From PAM , we can observed in the output graphs that the amplitude of the pulses is varying with respect to the amplitude of analog modulating signal, like in case of amplitude modulation.

   From PWM , we can see that unlike PAM, in this technique the amplitude of the signal is constant and only the width is varied. PWM technique is similar to frequency modulation as, by the varying the width of the pulses, the frequency of the pulses in the PWM signal shows variation.

   From PPM , we can see that pulse amplitude and the pulse width are constant that is they do not show variation with the amplitude of the modulating signal but only the position is varied. It must be noted that the position of the pulse changes according to the reference pulses. And these reference pulses are nothing but PWM pulses. Basically, the dropping edge of PWM pulses acts as the starting of the PPM pulses.

2. Nyquist Sampling Theorem

   The Nyquist sampling theorem is a theorem in the field of signal processing which serves as a fundamental concept between continuous-time signals and discrete-time signals. It establishes a sufficient condition for a sample rate that permits a discrete sequence of samples to capture all the information from a continuous-time signal of finite bandwidth.

   Nyquist's theorem states that a periodic signal must be sampled at more than twice the highest frequency component of the signal. In practice, because of the finite time available, a sample rate somewhat higher than this is necessary.

   If the frequency spectra of a function x(t) contains no frequencies higher than B hertz, x(t) is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart.

   - Sampling is recording values of a function at certain times
   - Allows for transformation of a continuous time function to a discrete time function
   - This is obtained by multiplication of f(t) by a unit impulse train

3. Applications of PAM, PWM, PPM

Applications of PAM

- It is used in Ethernet communication
- It is used in micro-controllers for generating control system signals.
- It is used in Photo-biology.
- It is used as an electronic driver for lighting.
- PAM is also used in the Ethernet network which is used to connect systems and used to transfer data among those systems.
- The control signals can be generated in various microcontrollers by using PAM
- This modulation technique is mostly used in digital data transmission and applications changed by PCM &PPM. Mostly all phone modems which are faster.

Applications of (PWM) :

- PWM methods are used in Telecommunications for encoding purposes.
- Pulse Width Modulation helps in voltage regulation and thus used in controlling Brightness in Smart Lighting Systems and also controls the speed of motors.
- Computer Motherboard requires PWM Signals that controls the heat generated in the board. PWM header is embedded in the fan that helps to dissipate the heat from the motherboard
- It is also used in Audio/Video Amplification.

Applications of PPM :

- Pulse position modulation is especially used in radio frequency communications like remote-controlled aircraft, cars, and other vehicles where communications are done to the receiver by a transmitter
- These are used in non coherent detection where a receiver does not need any Phase lock loop for tracking the phase of the carrier
- These are also sed in radio frequency (RF) communication. Also used in contactless smart card, high frequency, RFID tags
- This involves transmitting electrical, electromagnetic or optical signals or pulses to communicate or transfer data to a computer or any other device.