NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278

EXP. NO: 7

DATE: 08/03/2022

PPM, PAM AND PWM SIGNALS

AIM : Write a MATLAB program to execute and display the output for PPM (Pulse position modulation), PAM (Pulse amplitude modulation) and PWM (Pulse width modulation) signals.

SOFTWARE REQUIRED: MATLAB

THEORY:

a) Pulse position modulation (PPM)

Pulse Width Modulation (PWM) or Pulse Duration Modulation (PDM) or Pulse Time Modulation (PTM) is an analog modulating scheme in which the duration or width or time of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The width of the pulse varies in this method, but the amplitude of the signal remains constant. Amplitude limiters are used to make the amplitude of the signal constant. These circuits clip off the amplitude, to a desired level and hence the noise is limited.

b) Pulse amplitude modulation (PAM)

Pulse Amplitude Modulation (PAM) is an analog modulating scheme in which the amplitude of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The pulse amplitude modulated signal, will follow the amplitude of the original signal, as the signal traces out the path of the whole wave. In natural PAM, a signal sampled at the Nyquist rate is reconstructed, by passing it through an efficient Low Pass Frequency (LPF) with exact cutoff frequency.

c) Pulse width modulation (PWM)

Pulse Position Modulation (PPM) is an analog modulating scheme in which the amplitude and width of the pulses are kept constant, while the position of each pulse, with reference to the position of a reference pulse varies according to the instantaneous sampled value of the message signal.

Pulse position modulation is done in accordance with the pulse width modulated signal. Each trailing of the pulse width modulated signal becomes the starting point for pulses in PPM signal. Hence, the position of these pulses is proportional to the width of the PWM pulses.

ALGORITHM:

a) Pulse position modulation (PPM)

Step 1: Define the value for carrier frequency (fc), sampling frequency (fs) and modulating frequency (fm).

Step 2: Generate a message signal defined by X= 0.5*cos(2*pi*fm*t)+0.5.

Step 3: Modulate the message signal by using the modulate(x,fc,fs,method) command (modulation technique specified by method) in Matlab.

Step 4: Plot the message and modulated signal using the plot () command.

Step 5: Finally, save and click on Run to obtain the output graphs.

b) Pulse amplitude modulation (PAM)

Step 1: Define the value for amplitude and frequency.

Step 2: Generate an impulse signal using the stem(t) command and a sine signal using the sin (2 * pi * f * t) command in Matlab.

Step 3: Pulse amplitude modulation is achieved by multiplying pulse signal and message signal.

Step 4: Plot the impulse signal and sine signal using the plot () command. Plot the Pulse amplitude modulation signal by using the stem () command.

Step 5: Finally, save and click on Run to obtain the output graphs.

c) Pulse width modulation (PWM)

Step 1: Define the value for amplitude and frequency.

Step 2: Generate a sawtooth signal using the sawtooth(t) command and a sine signal using the sin (2 * pi * f * t) command in Matlab.

Step 3: Now compare the message signal and the sawtooth signal. If the message signal is greater, modulated signal is made as 1 or else it is set as 0.

Step 4: Plot the impulse signal, sine signal and modulated signal using the plot ()command.

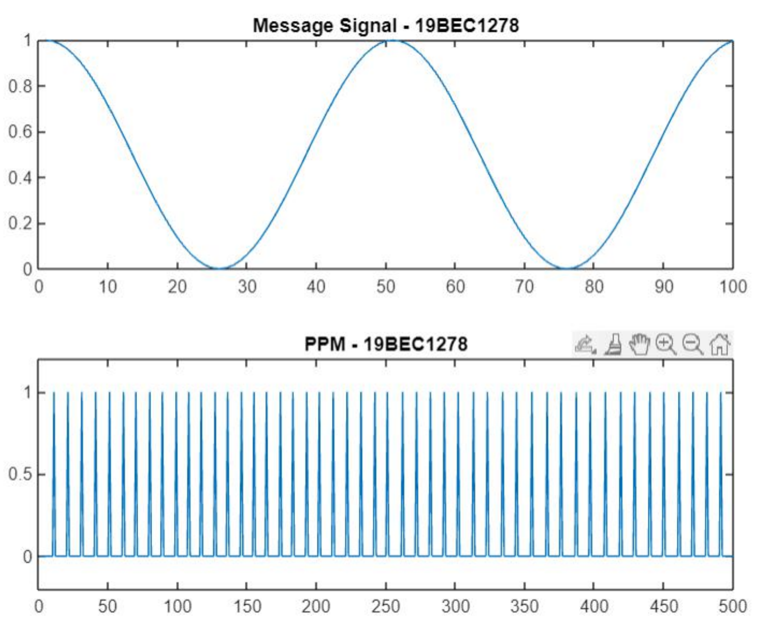
Step 5: Finally, save and click on Run to obtain the output graphs.

PPM:

Matlab Code:

```
clc;
clear all;
close all;
fc=100;
fs=1000;
fm=20;
t=0:1/fs:((2/fm)-(1/fs));
X=  0.5*cos(2*pi*fm*t)+0.5;
Y= modulate(X,fc,fs,'PPM');
subplot(2,1,1);
plot(X);
title('Message Signal - 19BEC1278');
subplot(2,1,2);
plot(Y);
axis([0 500 -0.2 1.2]);
title('PPM - 19BEC1278');
```

Output:

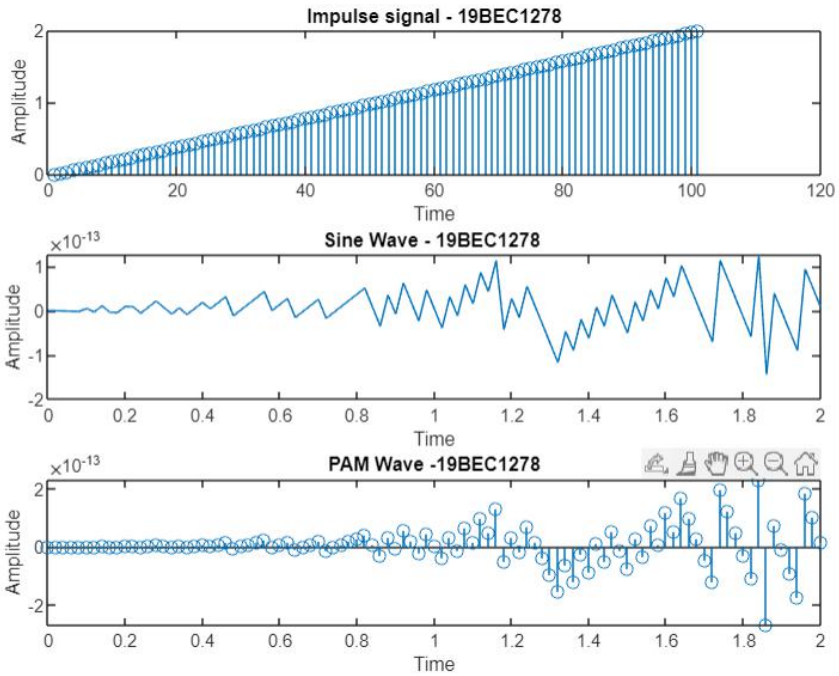


PAM:

Code:

```
clc
close all
clear all
a=input("Enter the amplitude: ");
f=input("Enter the frequency: ")'
t=0:0.02:2;
x1=stem(t);
x2=sin(2*pi*f*t);
y=t.*x2;
subplot(3,1,1);
stem(t);
title("Impulse signal - 19BEC1278");
xlabel("Time");   ylabel("Amplitude");
subplot(3,1,2);
plot(t,x2);
title("Sine Wave - 19BEC1278");
xlabel("Time");
ylabel("Amplitude");
subplot(3,1,3);
stem(t,y);
title("PAM Wave -19BEC1278");
xlabel("Time");
ylabel("Amplitude");
```

Output:



PWM:

Code:

```
clc;
clear;
close all;
t=0:0.001:10;
y=sin(2*pi*t);
z=sawtooth(4*pi*t);
subplot(3,1,1)
plot(y);
title("sine signal - 19BEC1278");
subplot(3,1,2)
plot(z);
```

```
title("saw signal -19BEC1278");
for i=1:length(t)
    if(y(i)>=z(i))
        u(i)=1;
    else
        u(i)=0;
    end
end
subplot(3,1,3)
plot(u);
```

Output: