ANALOG COMMUNICATION LAB REPORT (EC - 225)

ELECTRONICS AND COMMUNIACTION ENGINEARING

Submitted By

Himanta Biswa Deka

Scholar Id: 1914018

ECE(A)

EXPERIMENT 9

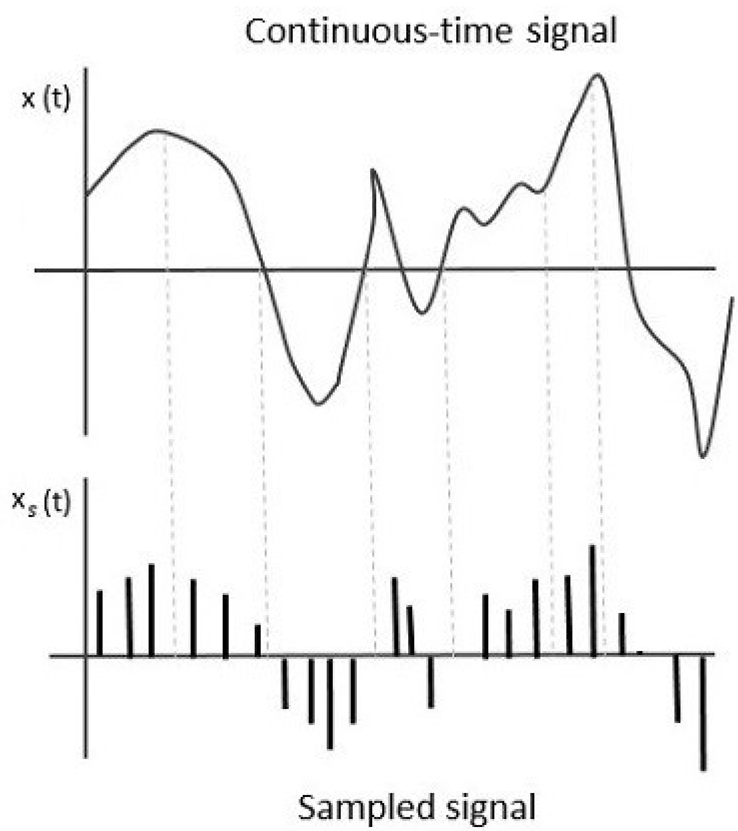
Aim: To verify the sampling theorem using MATLAB software.

Software Used : **MATLAB** software.

Brief Theory :

In real life scenario, we deal with the analog and the digital signals as well. But there should be a link between the two, for our convenience of analysis. An input signal is converted from some continuously varying physical value (e.g. pressure in air, or frequency or wavelength of light), by some electro-mechanical device into a continuously varying electrical signal, which is further converted to a sequence of digital values, called samples, by some analog to digital conversion circuit.

A sampling signal is a periodic train of pulses, having unit amplitude, sampled at equal intervals of time Ts, which is called as sampling time(fs = 1/Ts, Sampling rate). This data is transmitted at the time instants Ts and the carrier signal is transmitted at the remaining time.



The sampling rate should be such that the data in the message signal should neither be lost nor it should get over-lapped. The sampling theorem states that, "a signal can be exactly reproduced if it is sampled at the rate fs, which is greater than or equal to twice the maximum frequency of the given signal W."

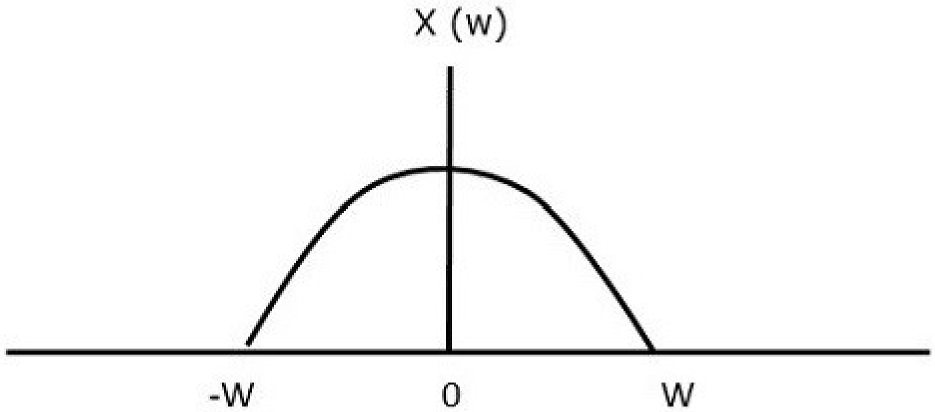
Mathematically, if fs is the sampling rate and W is the frequency of the given signal, then

$$fs \geq 2W$$

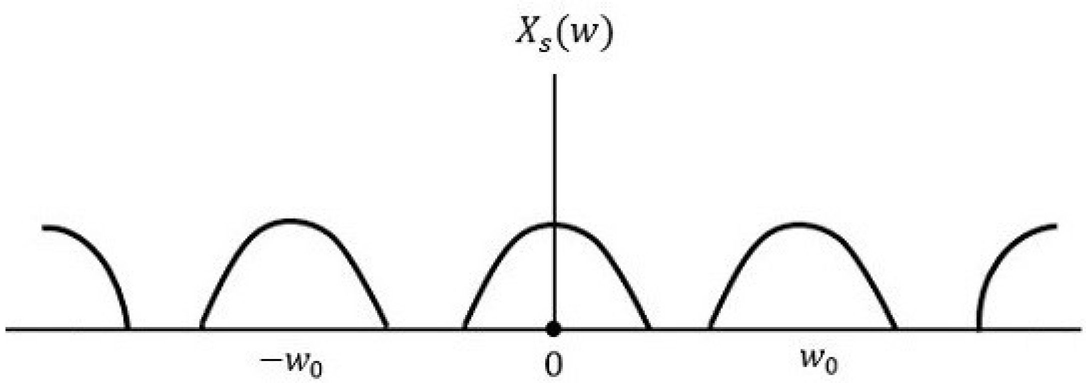
fs = 2W, the sampling rate is called as Nyquist rate.

Block Diagram of Sampling:

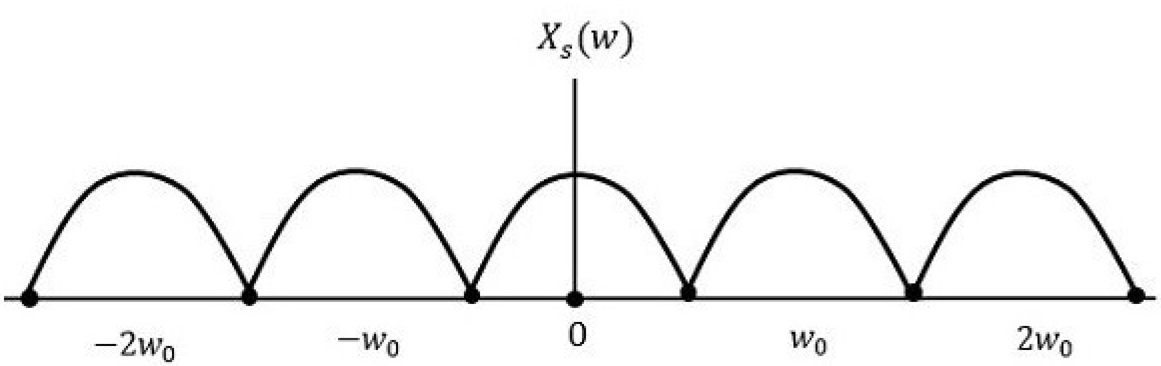
For continuous-time signal x(t), which is band-limited in the frequency domain:



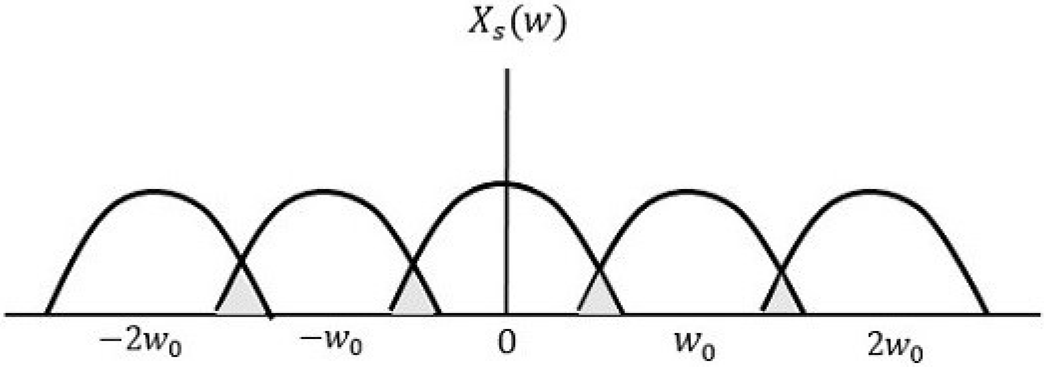
If fs > 2W,



If fs = 2W



If fs < 2W (Aliasing)



CODE AND EXECUTION:

The code attached herewith is for the sampling of the signals.

#Code:: sampling

CONCLUSION:

The MATLAB code for Sampling worked properly and the output curves were recorded and studied how they are varying with time. We try to cover almost the entire frequency range in sampling, although in practical there must be some ambiguity due to the limitations in the channel parameters. Several advanced methods are also there, which involves better mechanisms.

EXPERIMENT 10

Aim: To generate the Pulse Amplitude modulated signal using the given message signal in MATLAB and perform its demodulation as well.

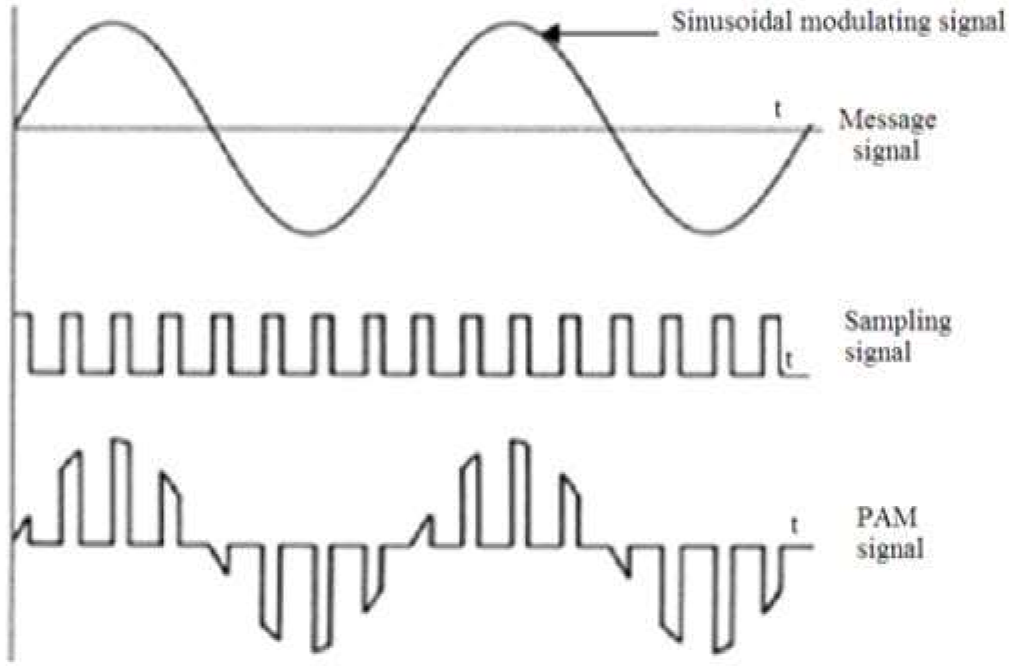
Software Used: MATLAB software.

Brief Theory :

Pulse-amplitude modulation is widely used in modulating signal transmission of digital data, with non-baseband applications. It is a modulation system in which the signal is sampled at regular intervals and each sample is made proportional to the amplitude of the signal at the instant of sampling. This technique transmits the data by encoding in the amplitude of a series of signal pulses. They are used in Ethernet communication, control of Light Emitting Diodes and broadcasting data for the Digital Television.

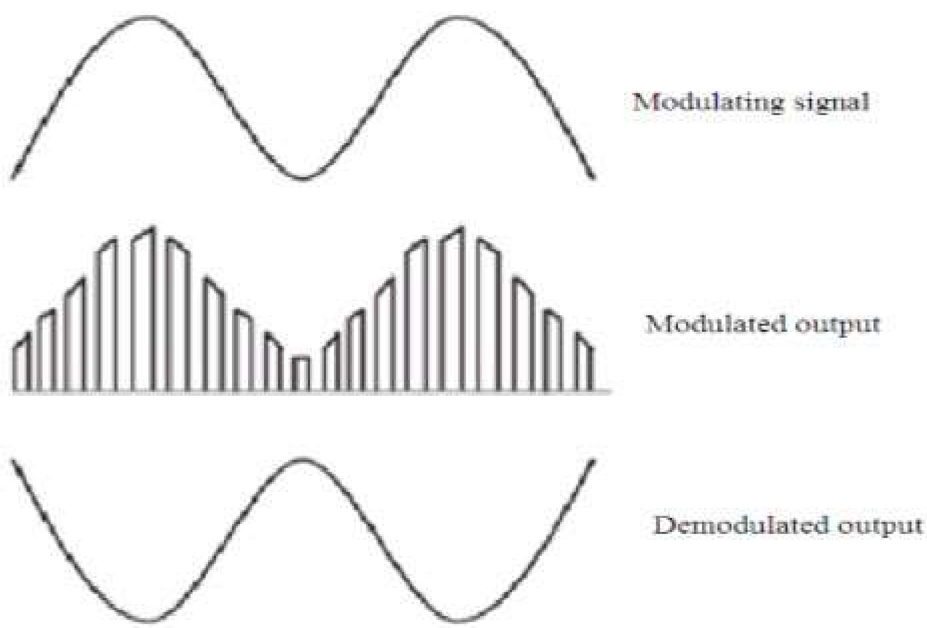
There are mainly two types of PAMs:

- single polarity PAM
- double polarity PAM



A PAM is generated from a pure sine wave modulating signal and a square wave generator which produces the carrier pulse and a PAM modulator circuit. We can also use the 555 IC for the modulation of the PAM signals.

For the purpose of demodulation the modulated signal is fed to a low pass filter which removes the high frequency ripples and then the later is applied to the inverting amplifier to amplify its signal level to have the demodulated output with almost equal amplitude with the modulating signal.



Block Diagrams of PAM and its Demodulation:

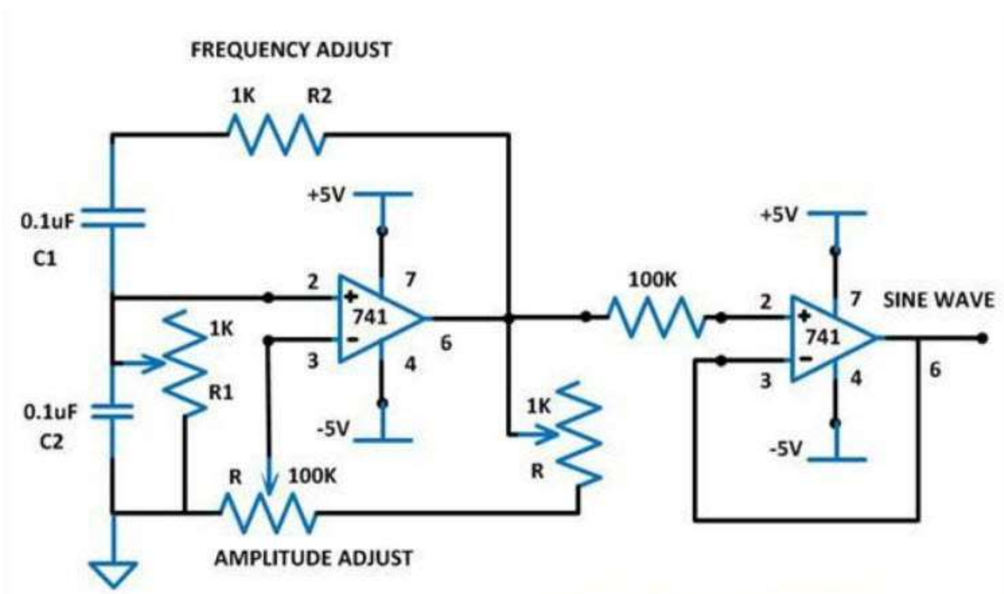


Fig: sine-wave generator

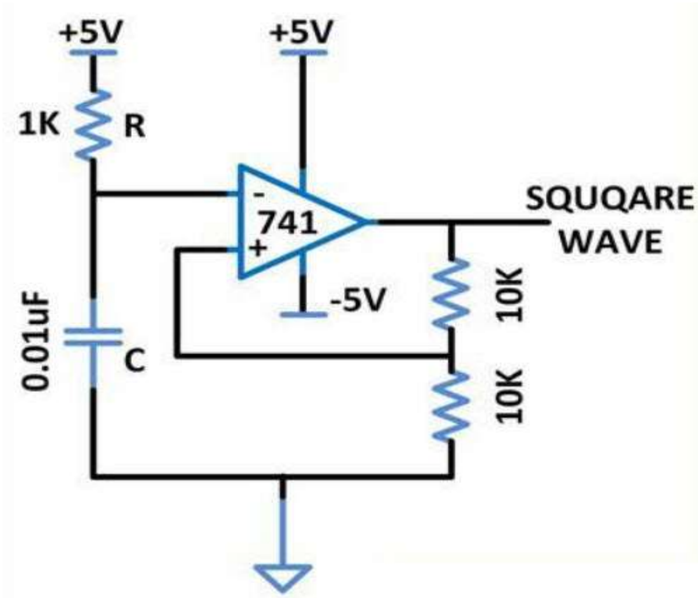


Fig: square-wave generator

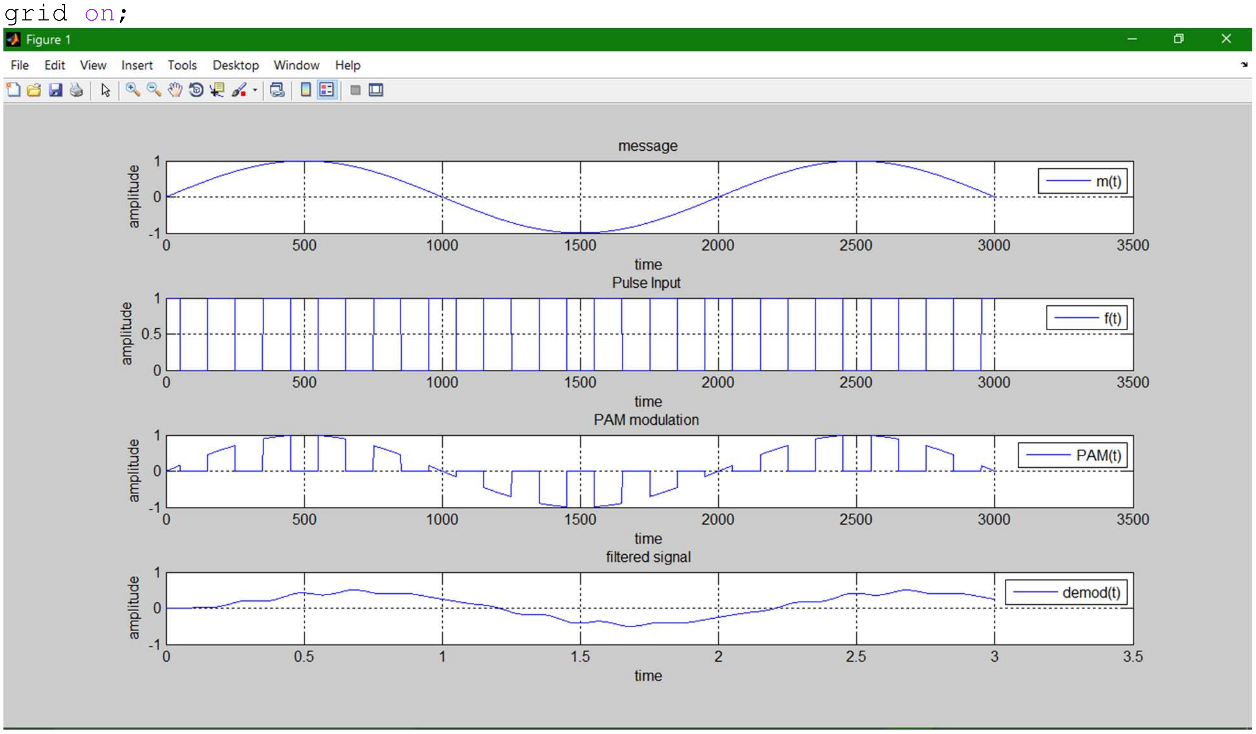
CODE AND EXECUTION:

The code for the PAM modulation and Demodulation with stimulation are attached below.

#Code::PAM Modulation & Demodulation

```
close all
clear all
clc
t = 0 : 1/1e3 : 3; % 1 kHz sample freq for 1 sec
d = 0 : 1/5 : 3;
x = sin(2*pi/4*2*t); %message signal
y = pulstran(t,d,'rectpuls',0.1); %generation of pulse input
z=x.*y; % PAM output

%demodulation
[den, num]=butter(1,2*pi*0.5/1000);
s11=filter(den,num,z);
s12=filter(den,num,s11);

figure;
subplot(4,1,1);
```

```
plot(x);
title('message');
xlabel('time');
ylabel('amplitude');
legend('m(t)');
grid on;

subplot(4,1,2)
plot(y);
title('Pulse Input ');
xlabel('time');
ylabel('amplitude');
legend('f(t)');
grid on;

subplot(4,1,3);
plot(z);
title('PAM modulation ');
xlabel('time');
ylabel('amplitude');
legend('PAM(t)');
grid on;

subplot(4,1,4)
plot(t,s12)
axis([0 3.5 -1 1]);
title('filtered signal')
xlabel('time');
ylabel('amplitude');
legend('demod(t)');
grid on;
```



CONCLUSION:

The MATLAB code for PAM modulation and demodulation worked properly and the output curves were recorded and studied how they are varying with time. In communication PAM finds a very vast use though bandwidth and power requirement are quite high. Several advanced methods are also there, which involves better mechanisms.

EXPERIMENT 11

Aim: To generate the Pulse Width Modulated signal using the given message signal in MATLAB and perform its demodulation as well.

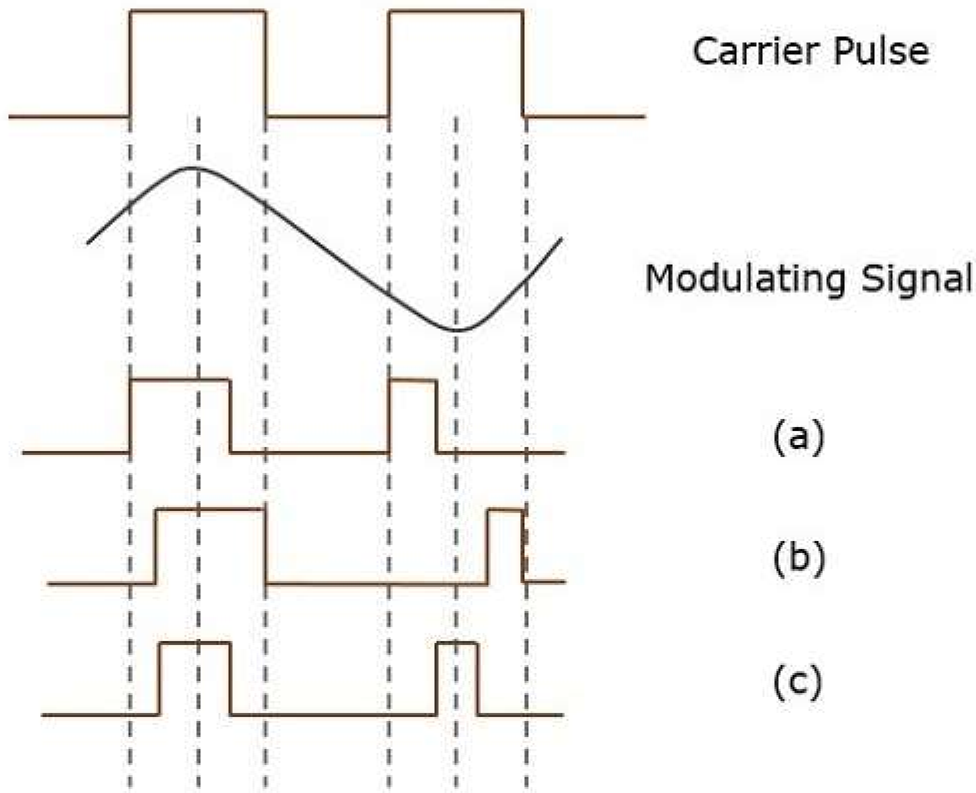
Software Used: MATLAB software.

Brief Theory :

In Pulse Width Modulation (PWM) or Pulse Duration Modulation (PDM) or Pulse Time Modulation (PTM) technique, the width or the duration or the time of the pulse carrier varies in accordance 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.

The following are the types of PWM:



A)The leading edge of the pulse being constant, the trailing edge varies according to the message signal. The waveform for this type of PWM is denoted as (a) in the above figure.

B)The trailing edge of the pulse being constant, the leading edge varies according to the message signal. The waveform for this type of PWM is denoted as (b) in the above figure.

C)The center of the pulse being constant, the leading edge and the trailing edge varies according to the message signal. The waveform for this type of PWM is denoted as (c) shown in the above figure.

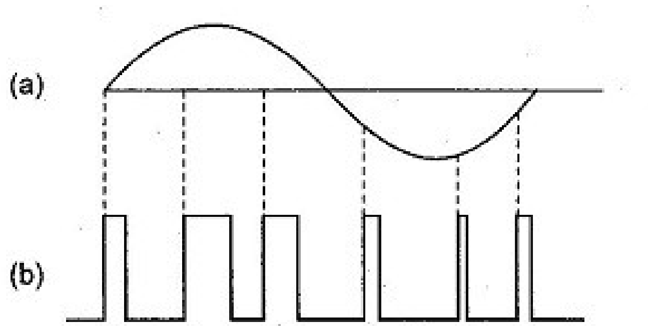


Fig. 18.8 Pulse Width Modulation (a) Signal (b) PWM

Advantages of PTM

- Low power consumption.
- It has an efficiency of about 90 per cent.
- Noise interference is less.
- High power handling capacity.

Disadvantages of PTM

- The circuit is more complex.
- Voltage spikes can be seen.
- The system is expensive as it uses semiconductor devices.
- Switching losses will be more due to high PWM frequency.

Applications of PTM

- Used in encoding purposes in the telecommunication system.
- Used to control brightness in a smart lighting system.
- Helps to prevent overheating in LED’s while maintaining it’s brightness.
- Used in audio and video amplifiers.

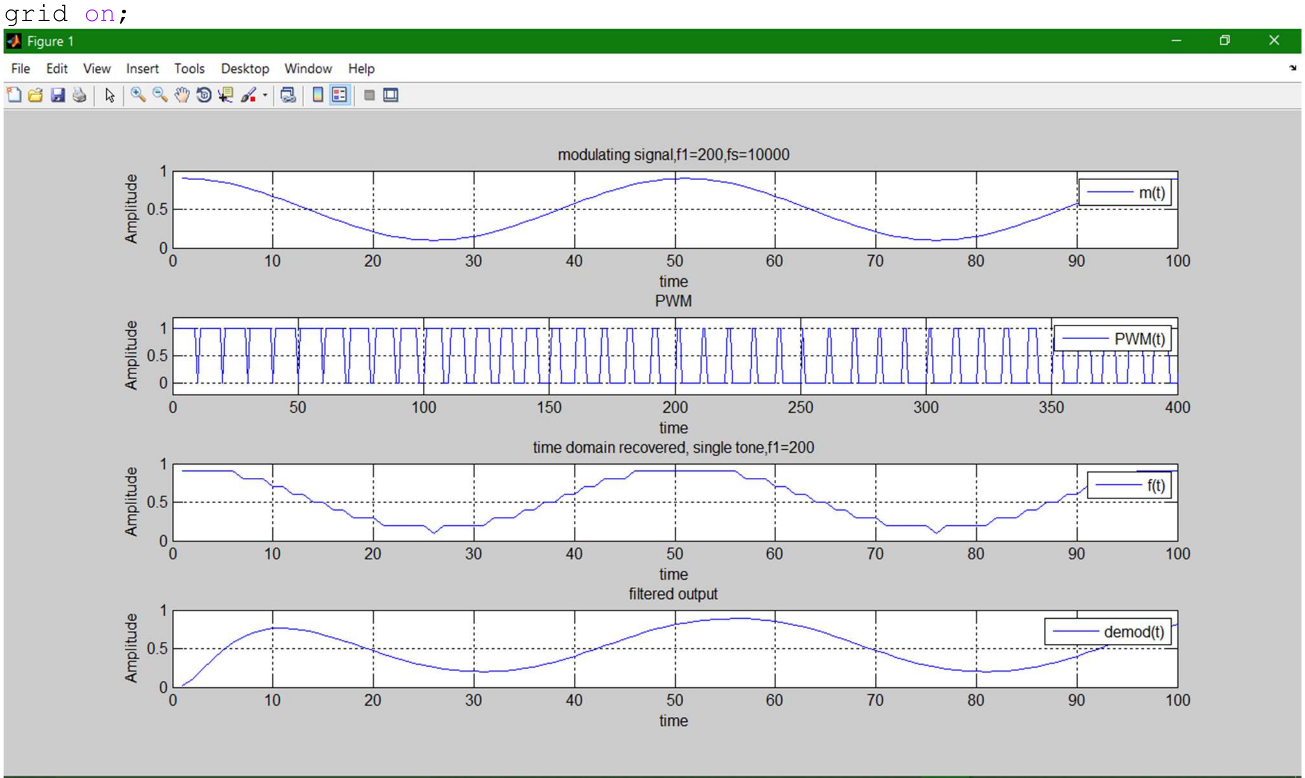
CODE AND EXECUTION:

The code for the PWM modulation and Demodulation with stimulation are attached below.

#Code::PWM Modulation & Demodulation

```
% pulse width modulation & demodulation

close all
clear all
clc
fc=1000;
fs=10000;
f1=200;
t=0:1/fs:((2/f1)-(1/fs));
x1=0.4*cos(2*pi*f1*t)+0.5;

%modulation
y1=modulate(x1,fc,fs,'pwm');

%demodulation
x1_recov=demod(y1,fc,fs,'pwm');
[den, num]=butter(1,2*pi*f1/fs);
s11=filter(den,num,x1_recov);
s12=filter(den,num,s11);

figure;
subplot(4,1,1);
plot(x1);
axis([0 100 0 1]);
title('modulating signal,f1=200,fs=10000');
xlabel('time');
ylabel('Amplitude');
legend('m(t)');
grid on;

subplot(4,1,2);
plot(y1);
axis([0 400 -0.2 1.2]);
title('PWM');
xlabel('time');
ylabel('Amplitude');
legend('PWM(t)');
grid on;

subplot(413);
plot(x1_recov);
title('time domain recovered, single tone,f1=200')
axis([0 100 0 1]);
xlabel('time');
ylabel('Amplitude');
legend('f(t)');
```

```
grid on;

subplot(414);
plot(s12);
title('filtered output')
axis([0 100 0 1]);
xlabel('time');
ylabel('Amplitude');
legend('demod(t)');
grid on;
```



CONCLUSION:

Thus, we have studied pulse width modulation and demodulation. It has lower noise and lower power consumption and it is highly efficient. It is used in telecommunication, power delivery etc.

EXPERIMENT 12

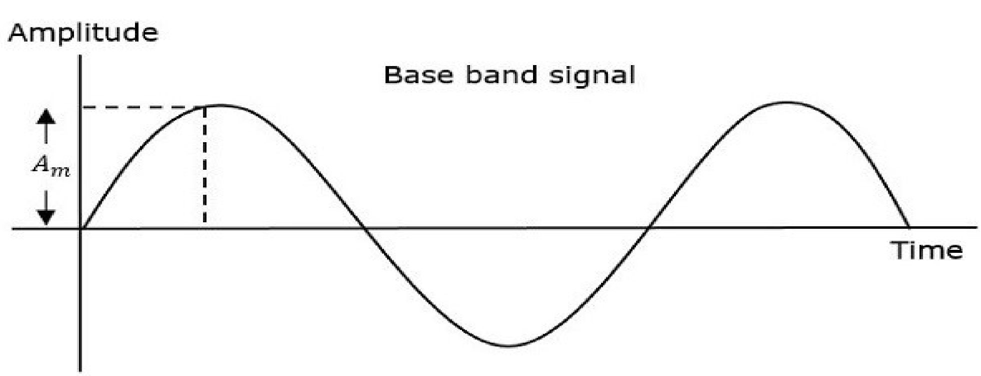
Aim: To generate the Pulse Position Modulated signal using the given message signal in MATLAB and perform its demodulation as well.

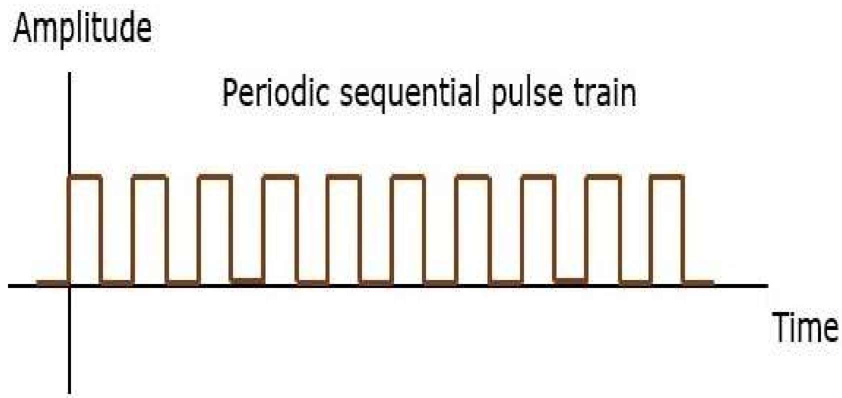
Software Used: MATLAB software.

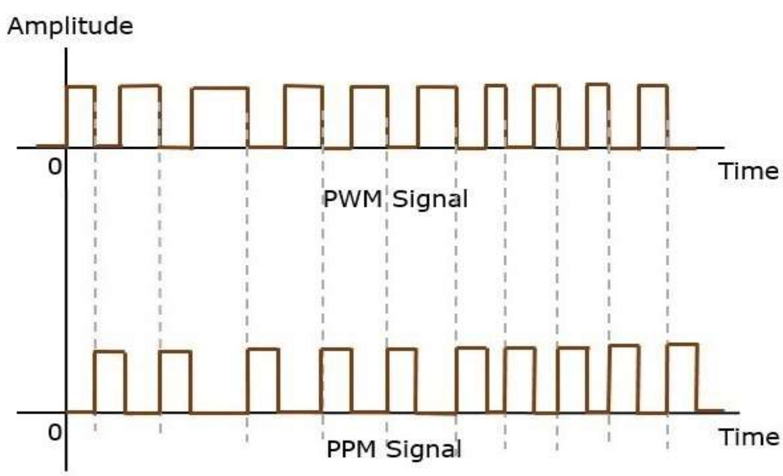
Brief Theory :

Pulse Position Modulation (PPM) is an analog modulation scheme in which, the amplitude and the 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.

The transmitter has to send synchronizing pulses (or simply sync pulses) to keep the transmitter and the receiver in sync. These sync pulses help to maintain the position of the pulses. The following figures explain the Pulse Position Modulation.







Thus,we can observe that for PPM signal only the position of pulses is changed but the amplitude and width is kept constant.

Pulse position modulation is done in accordance with the pulse width modulated signal. Each trailing edge 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.

Advantages of PPM

- As it has constant amplitude noise interference is less.
- We can easily separate signal from a noisy signal.
- Among all three types, it has the most power efficiency.
- Requires less power when compared to pulse amplitude modulation.

Disadvantages of PPM

- The system is highly complex.
- The system requires more bandwidth.

Applications of PPM

- It is used in the air traffic control system and telecommunication systems.
- Remote controlled cars, planes, trains use pulse code modulations.
- It is used to compress data and hence it is used for storage.

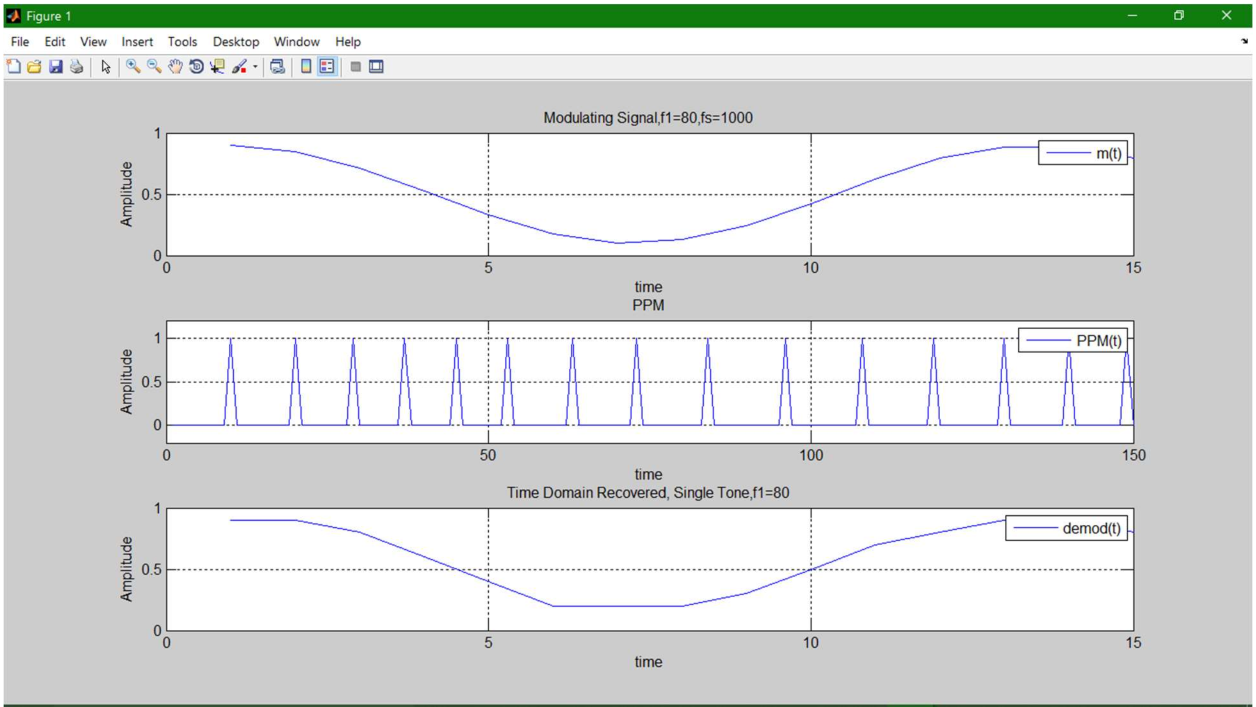
CODE AND EXECUTION:

The code for the PPM modulation and Demodulation with stimulation are attached below.

#Code::PPM Modulation & Demodulation

```
% pulse position modulation
close all
clear all
clc
```

```
fc=100;
fs=1000;
f1=80;
t=0:1/fs:((2/f1)-(1/fs));
x1=0.4*cos(2*pi*f1*t)+0.5;

%modulation
y1=modulate(x1,fc,fs,'ppm');

%demodulation
x1_recov=demod(y1,fc,fs,'ppm');

figure;
subplot(3,1,1);
plot(x1);
axis([0 15 0 1]);
title('Modulating Signal,f1=80,fs=1000');
xlabel('time');
ylabel('Amplitude');
legend('m(t)');
grid on;

subplot(3,1,2);
plot(y1);
axis([0 150 -0.2 1.2]);
title('PPM');
xlabel('time');
ylabel('Amplitude');
legend('PPM(t)');
grid on;

subplot(313);
plot(x1_recov);
title('Time Domain Recovered, Single Tone,f1=80')
axis([0 15 0 1]);
xlabel('time');
ylabel('Amplitude');
legend('demod(t)');
grid on;
```



CONCLUSION:

Thus, we have studied pulse position modulation and demodulation. Here, only position of the pulses are varied according to the instantaneous value of the message signal.