Aim:

To plot the wave forms for: - Sine, Cos, Exponential, Ramp, Unit Impulse, Unit Step Signal function in continuous time using MATLAB.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

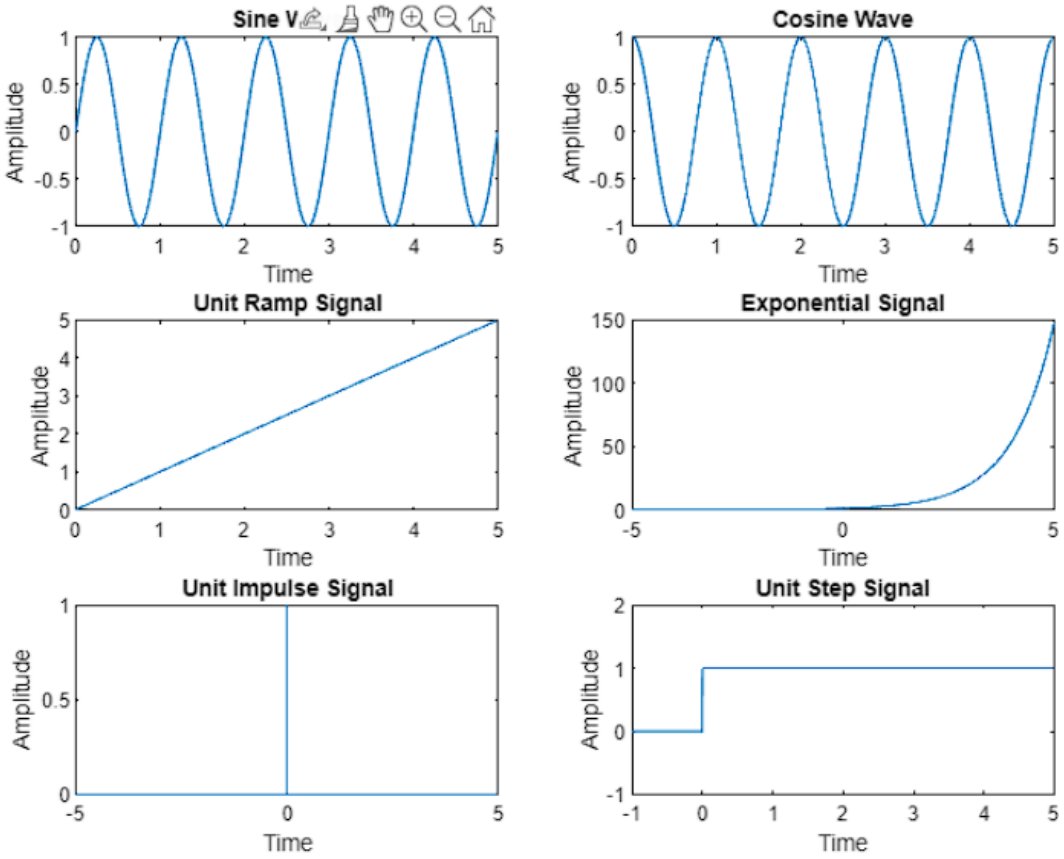
Personal Computer

MATLAB Code

```
t = [-5: 0.01: 5];
f = 1;
a = sin(2*pi*f*t);
b = cos(2*pi*f*t);
x = t;
y = (t==0);
z = (t>= 0);
c = exp(t);
subplot(321);
plot(t,a);
axis([0 5 -1 1]);
xlabel("Time");
ylabel("Amplitude");
title("Sine Wave");
subplot(322), plot(t,b);
axis([0 5 -1 1]);
xlabel("Time");
ylabel("Amplitude");
title("Cosine Wave");
subplot(323), plot(t,x);
axis([0 5 0 5]);
xlabel("Time");
ylabel("Amplitude");
```

```
title("Unit Ramp Signal");
subplot(324);
plot(t,c);
xlabel("Time");
ylabel("Amplitude");
title("Exponential Signal");
subplot(325);
plot(t,y);
xlabel("Time");
ylabel("Amplitude");
title("Unit Impulse Signal");
subplot(326);
plot(t,z);
axis([-1 5 -1 2]);
xlabel("Time");
ylabel('Amplitude');
title("Unit Step Signal");
```

Output


Sine V
Amplitude
Time
Cosine Wave
Amplitude
Time
Unit Ramp Signal
Amplitude
Time
Exponential Signal
Amplitude
Time
Unit Impulse Signal
Amplitude
Time
Unit Step Signal
Amplitude
Time

Aim:

To plot the wave forms for: - Sine, Cos, Exponential, Ramp, Unit Impulse, Unit Step Signal function in discrete time using MATLAB.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

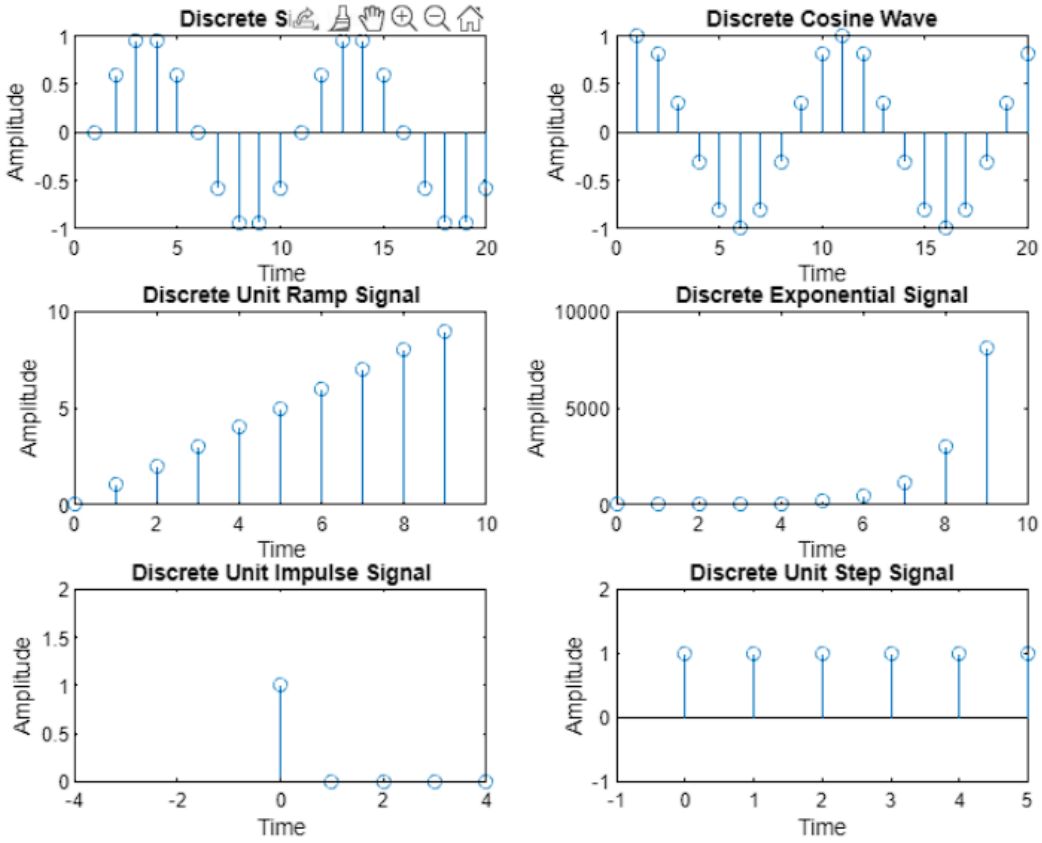
Personal Computer

MATLAB Code

```
N = 10;
n = 0:N-1;
n1 = 0:0.1:5;
f = 1;
a = sin(2*pi*f*n1);
b = cos(2*pi*f*n1);
x = n;
y = (n==0);
z = (n>= 0);
c = exp(n);
subplot(321);
stem(a); axis([0 20 -1 1]);
xlabel("Time");
ylabel("Amplitude");
title("Discrete Sine Wave");
subplot(322), stem(b);
axis([0 20 -1 1]);
xlabel("Time");
ylabel("Amplitude");
title("Discrete Cosine Wave");
subplot(323), stem(n,x);
axis([0 10 0 10]);
xlabel("Time");
```

```
ylabel("Amplitude");
title("Discrete Unit Ramp Signal");
subplot(324);
stem(n,c);
xlabel("Time");
ylabel("Amplitude");
title("Discrete Exponential Signal");
subplot(325);
stem(n,y);
axis([-4 4 0 2]);
xlabel("Time");
ylabel("Amplitude");
title("Discrete Unit Impulse Signal");
subplot(326);
stem(n,z);
axis([-1 5 -1 2]);
xlabel("Time");
ylabel('Amplitude');
title("Discrete Unit Step Signal");
```

Output


Discrete S
Amplitude
Time
Discrete Cosine Wave
Amplitude
Time
Discrete Unit Ramp Signal
Amplitude
Time
Discrete Exponential Signal
Amplitude
Time
Discrete Unit Impulse Signal
Amplitude
Time
Discrete Unit Step Signal
Amplitude
Time

Aim:

To simulate Amplitude Modulation (AM) Technique using MATLAB software.

Software and Hardware Required:

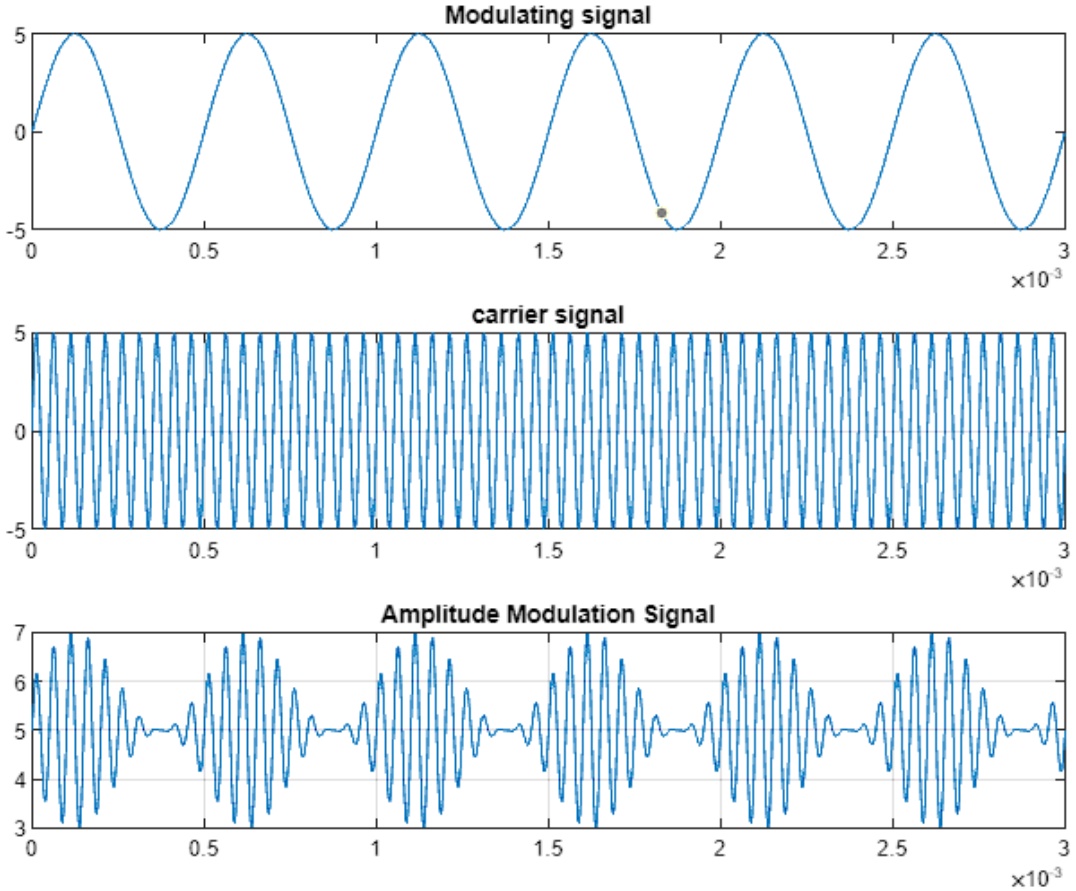
MTLAB Software (MATLAB R2022b)

Personal Computer

MATLAB Code

```
m=1;
Am = 5; %Amp. of modulating Signal
fa = 2000; % frequency of modulating signal
Ta = 1/fa;
t = 0:Ta/999:6*Ta;
ym = Am*sin(2*pi*fa*t);

subplot(3,1,1);
plot(t,ym)
title("Modulating signal")
%carrier signal
Ac = Am/m;
fc = fa*10;
Tc = 1/fc;
ye = Ac*sin(2*pi*fc*t);
subplot(3,1,2);
plot(t,ye)
grid on;
title("carrier signal");
%Am Modulation
y = Ac + (1 + m*sin(2*pi*fa*t)).*sin(2*pi*fc*t);
subplot(3,1,3);
plot(t,y);
title("Amplitude Modulation Signal")
grid on;
```

Output


Modulating signal
5
0
-5
0
0.5
1
1.5
2
2.5
3
×10-3
carrier signal
5
0
-5
0
0.5
1
1.5
2
2.5
3
×10-3
Amplitude Modulation Signal
7
6
5
4
3
0
0.5
1
1.5
2
2.5
3
×10-3

Aim:

To simulate Frequency Modulation(FM) using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

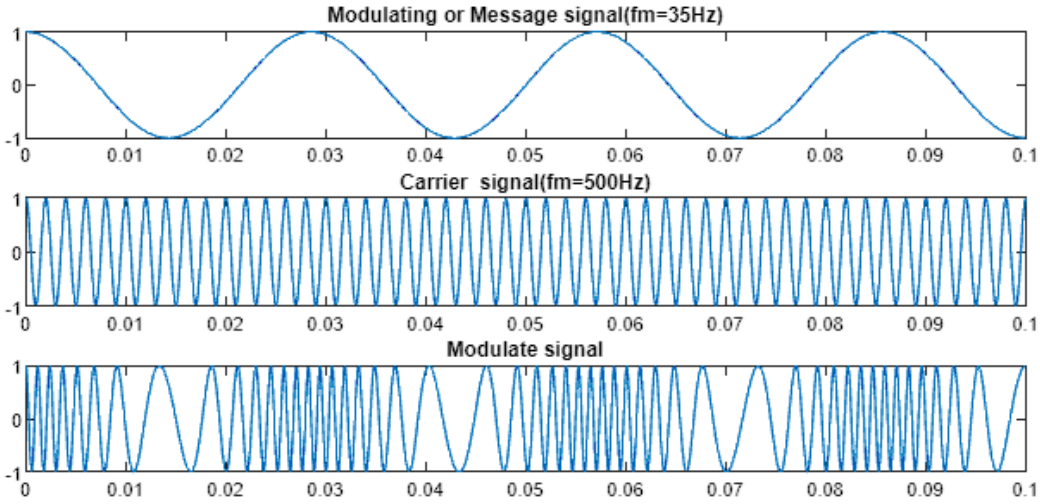
Personal Computer

MATLAB Code

%The frequency modulation (FM)waveform in time and frequency domain.
%fm=35HZ,fc=500HZ,Am=1V,Ac=1V,B=10

```
fs=10000;
Ac=1;
 Am=1;
 fm=35;
 fc=500;
 B=10;
t=(0:0.1*fs)/fs;
 wc=2*pi*fc;
 wm=2*pi*fm;
m_t=Am*cos(wm*t);
subplot(5,1,1);
plot(t,m_t); title('Modulating or Message signal(fm=35Hz)');
c_t=Ac*cos(wc*t);
subplot(5,1,2);
plot(t,c_t);
 title('Carrier  signal(fm=500Hz)');
s_t=Ac*cos((wc*t)+B*sin(wm*t));
subplot(5,1,3); plot(t,s_t);
 title('Modulate signal');
```

Output


Modulating or Message signal(fm=35Hz)
Carrier signal(fm=500Hz)
Modulate signal
1
0
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

Aim:

To simulate Pulse Code Modulation(PCM) technique using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

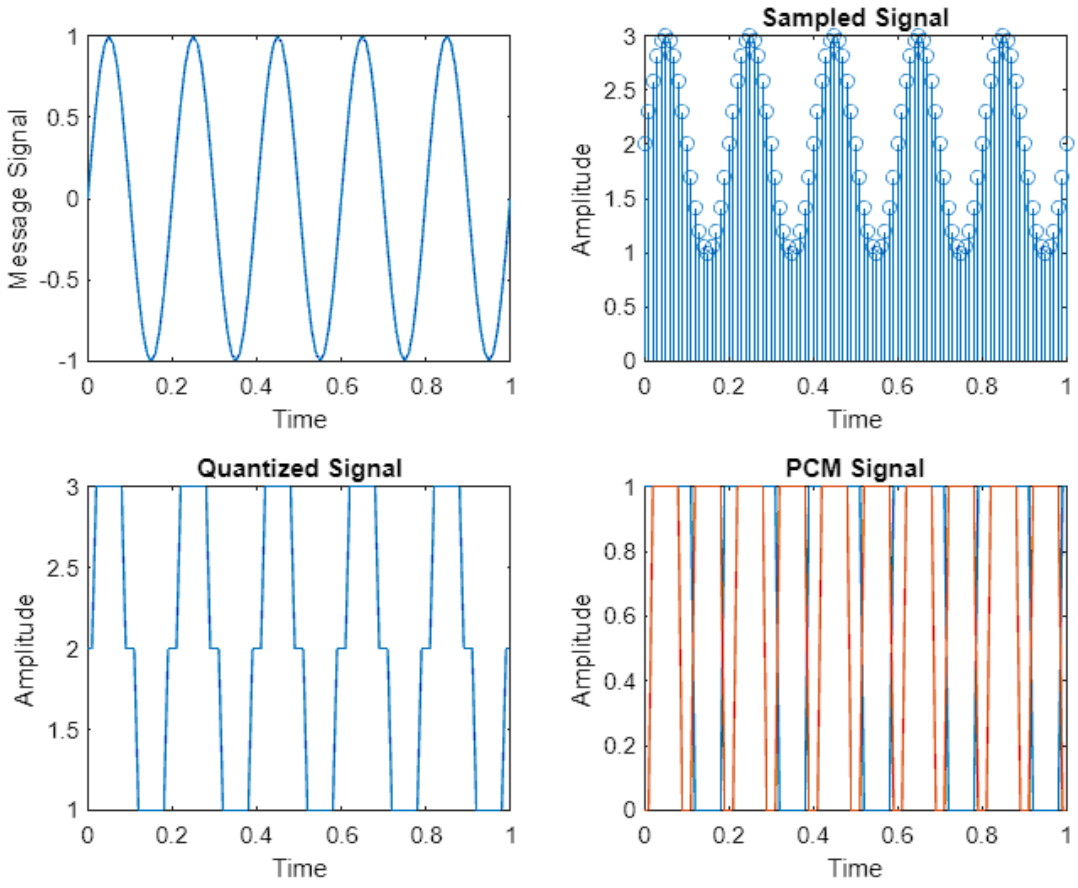
Personal Computer

MATLAB Code

```
t = 0:0.01:1;
f = 5;
a = sin(2*pi*f*t);
subplot(221);
plot(t,a);
xlabel('Time');
ylabel("Message Signal");
b = 2;
c = a + b;
subplot(222);
stem(t,c);
xlabel("Time")
ylabel("Amplitude");
title("Sampled Signal")
d = quant(c);
subplot(223);
plot(t,d);
xlabel('Time');
ylabel('Amplitude');
title('Quantized Signal');
e = de2bi(d,'left-msb');
subplot(224);
plot(t,e);
axis([0 1 0 1]);
```

```
xlabel("Time");
ylabel("Amplitude");
title("PCM Signal")
```

Output



Aim:

To simulate Amplitude Shift Keying(ASK) technique using MATLAB software.

Software and Hardware Required:

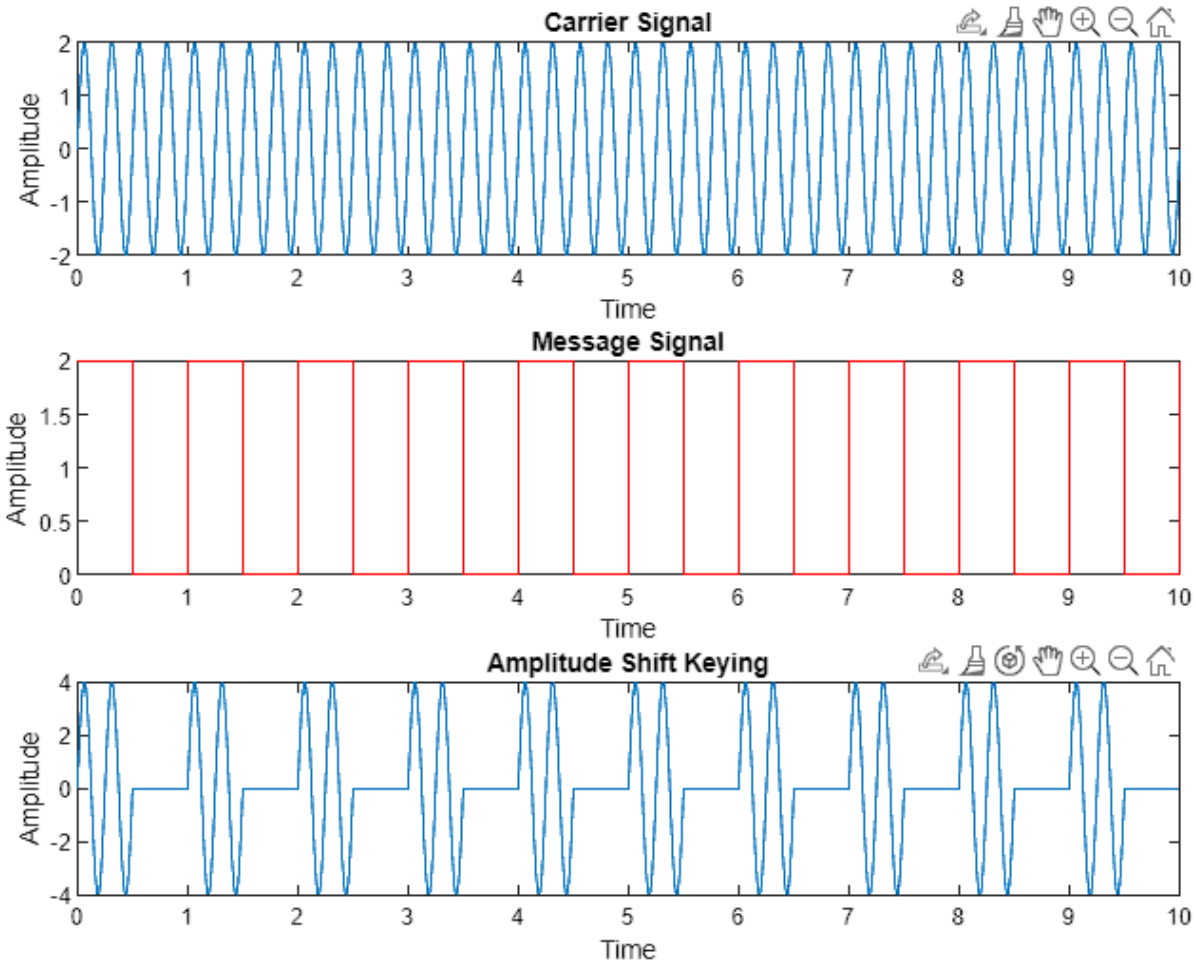
MTLAB Software (MATLAB R2022b)

Personal Computer

MATLAB Code

```
t = 0:0.001:10;
b = 2;
fc = 4 %sinusoidal signal
fm = 1 % pulse
a = b*(sin(2*pi*fc*t));
subplot(3, 1, 1);
plot(t,a);
title('Carrier Signal');
xlabel('Time');
ylabel('Amplitude');

y = (b/2)*square(2*pi*fm*t,50)+(b/2);
y1 = a.*y;
subplot(3,1,2);
plot(t,y,'r');
title('Message Signal');
xlabel('Time');
ylabel('Amplitude');
subplot(3,1,3);
plot(t,y1);
title("Amplitude Shift Keying");
xlabel("Time");
ylabel("Amplitude");
```

Output


Carrier Signal
Amplitude
Time
Message Signal
Amplitude
Time
Amplitude Shift Keying
Amplitude
Time

Aim:

To simulate Frequency Shift Keying(FSK) technique using MATLAB software.

Software and Hardware Required:

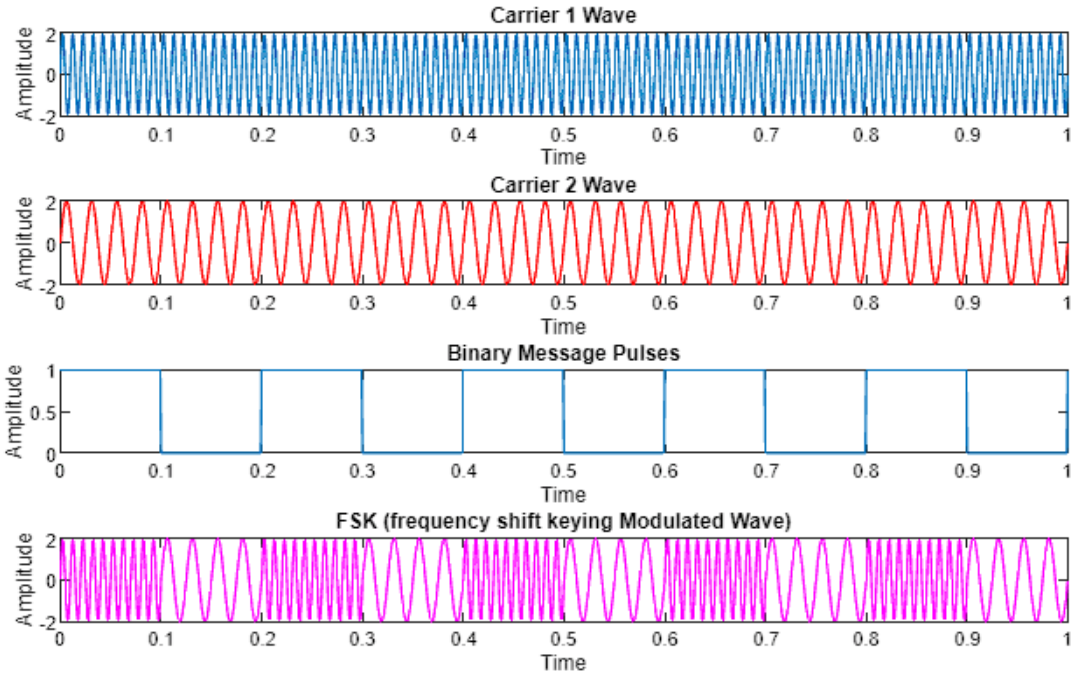
MTLAB Software (MATLAB R2022b)

Personal Computer

MATLAB Code

```
fc1 = input("Enter the freq 1st Sine Wave Carrier high frequency : ");
fc2 = input("Enter the freq of 2nd Sine Wave carrier lower frequency :");
fp = input("Enter the freq of Periodic Binary pulse(Message):");
amp = 2;
a = 0.5;
t = 0:0.001:1; % For setting the sampling interval
c1 = amp.*sin(2*pi*fc1*t); %For Generating 1st Carrier Sine Wave/1
c2 = amp.*sin(2*pi*fc2*t); %For Generating 2nd Carrier Sine Wave/0
subplot(5,1,1); %For Plotting the Carrier Wave
plot(t,c1);
xlabel("Time");
ylabel("Amplitude");
title("Carrier 1 Wave");
subplot(5,1,2); %For Plotting the Carrier Wave
plot(t,c2,'r')
xlabel("Time");
ylabel("Amplitude")
title("Carrier 2 Wave");
m = a.*square(2*pi*fp*t)+a; %For Generating Square wave message
subplot(5,1,3) %For plotting the Square Binary Pulse (Message)
plot(t,m);
xlabel("Time")
ylabel("Amplitude")
title('Binary Message Pulses')
n = length(t);
for i = 1:n % here we are generating the modulated wave
    if m(i) == 0
        f(i) = c2(i);
    else
        f(i) = c1(i);
    end
end

subplot(5,1,4) %For Plotting The Modulated Wave
plot(t,f,'m');
xlabel('Time');
ylabel('Amplitude');
title("FSK (frequency shift keying Modulated Wave)")
```

Output


Carrier 1 Wave
Amplitude
Time
Carrier 2 Wave
Amplitude
Time
Binary Message Pulses
Amplitude
Time
FSK (frequency shift keying Modulated Wave)
Amplitude
Time

Aim:

To simulate Phase Shift Keying (PSK) technique using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

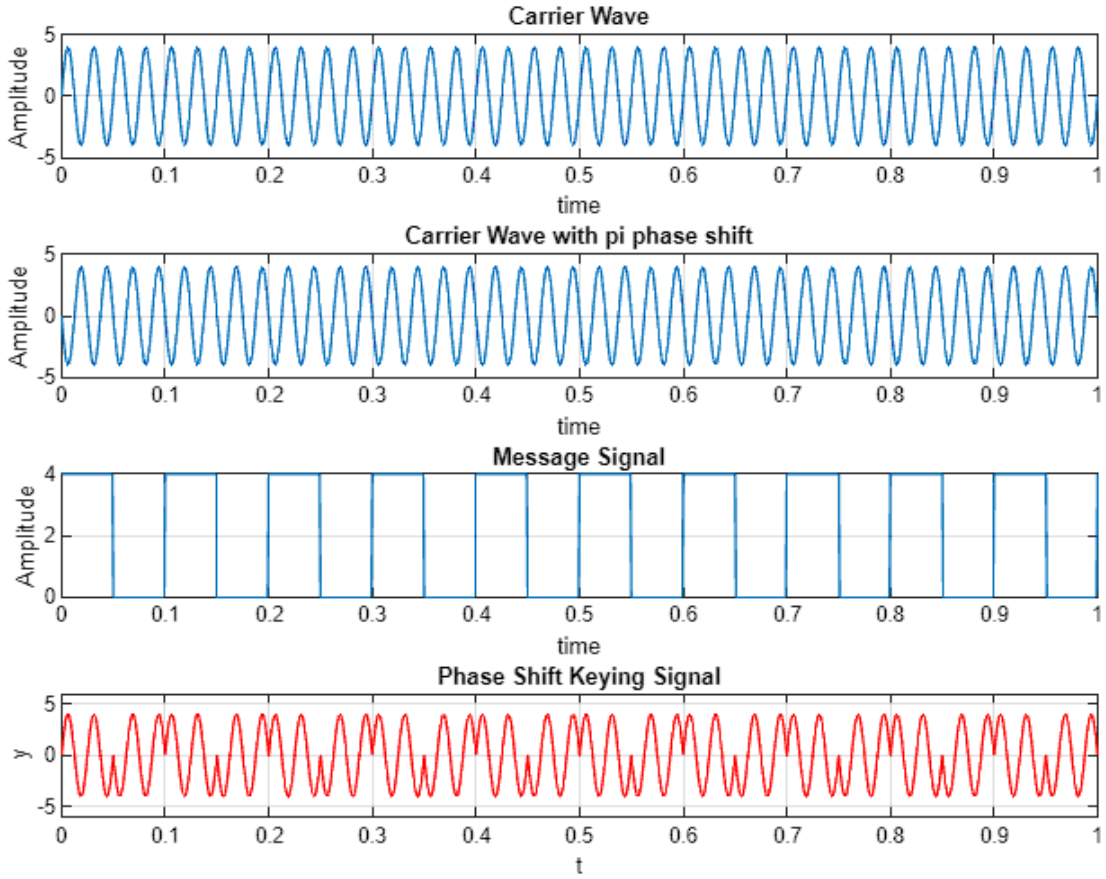
Personal Computer

MATLAB Code

```
A = 4;
t = 0:0.001:1;
f1 = input('Carrier Sine Wave Frequency = ');
f2 = input("Message frequency = ");
x = A.*sin(2*pi*f1*t); %Carrier Sine
y = A.*sin(2*pi*f1*t + pi);
subplot(4,1,1);
plot(t,x);
xlabel('time');
ylabel("Amplitude");
title("Carrier Wave");
grid on;
subplot(4,1,2);
plot(t,y);
xlabel('time');
ylabel('Amplitude');
title('Carrier Wave with pi phase shift');
grid on;
m = A/2.*square(2*pi*f2*t) + A/2 %Message Signal
subplot(4,1,3);
plot(t,m);
xlabel('time');
ylabel("Amplitude");
title('Message Signal');
grid on;
n = length(t);
for i=1:n %here we are generating the modulate wave
    if m(i) == 0;
        f(i) = y(i);
    else
        f(i) = x(i);
    end
end

subplot(4,1,4);
plot(t,f,'r')
axis([0 1 -6 6]);
xlabel('t');
ylabel('y');
title('Phase Shift Keying Signal');
grid on;
```

```
Carrier Sine Wave Frequency =
40
Message frequency =
10
```

Output



Aim:

To simulate Pulse Amplitude Modulation (PAM) technique using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

Personal Computer

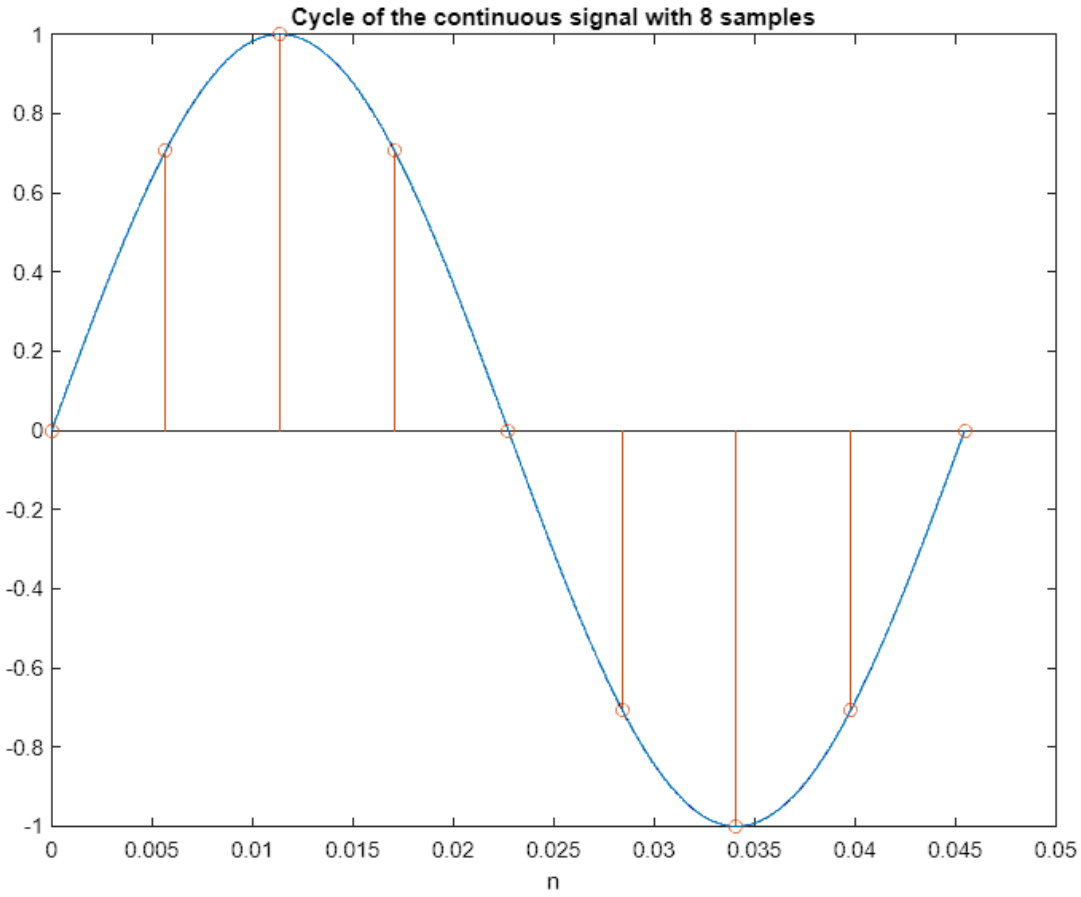
MATLAB Code

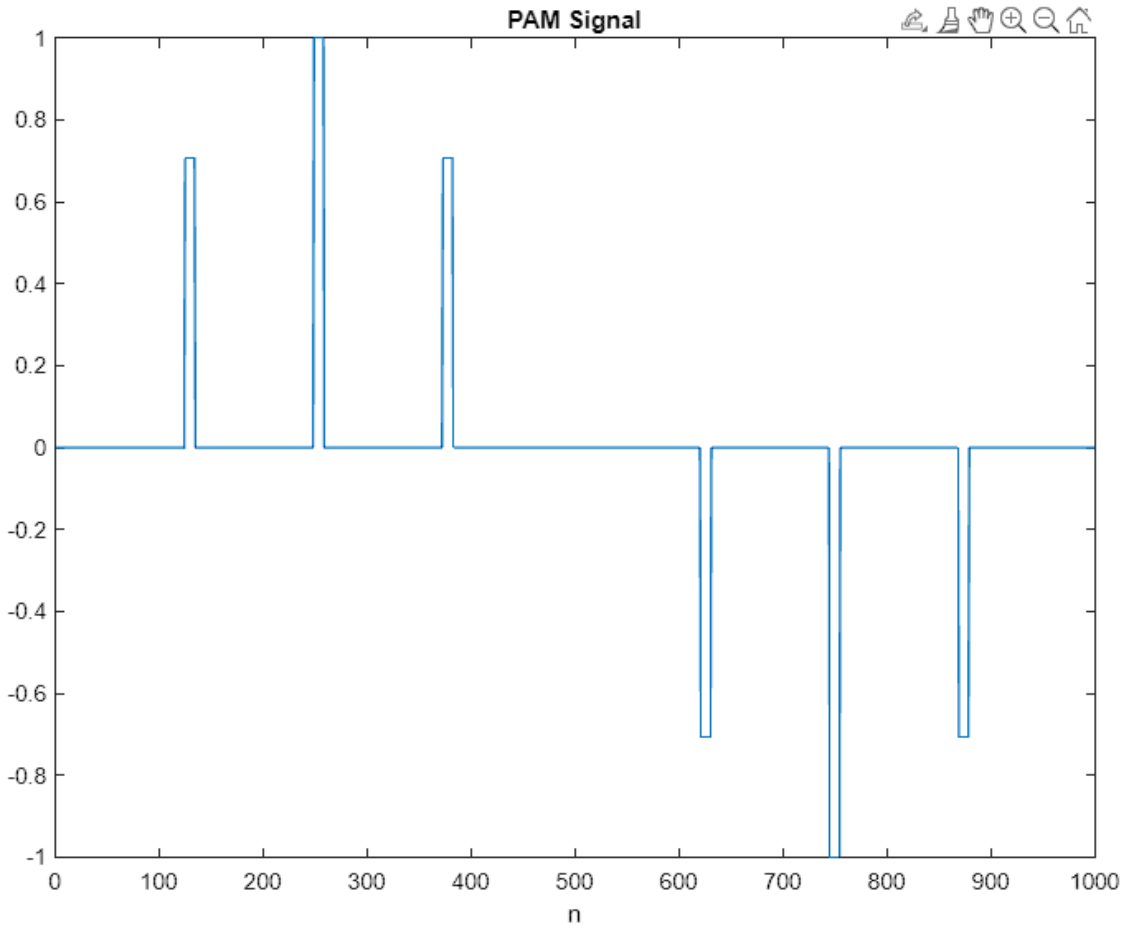
```
%continuous signal
L=1000;
fsim=22000; %sampling frequency of the continuous signal
fc=fsim/L; %fc allows to get 1 cycle of L samples
Tcycle=1/fc; % 1 period
Tcont=1/fsim; %sampling period
continuous_time_axis=[0:Tcont:Tcycle]; %a period of the signal
ycont=sin(2*pi*fc*continuous_time_axis); % a cycle of a continuos sinusoid
%ideal sampled signal
num_samples_cycle=8;
Tm=Tcont*L/num_samples_cycle; %sampling period of the sampled signal
discrete_time_axis=[0:Tm:Tcycle];
ysampled=sin(2*pi*fc*discrete_time_axis);
figure(1);
plot(continuous_time_axis,ycont); title('Cycle of the continuous signal
with 8 samples'); xlabel('n');
hold on;
stem(discrete_time_axis,ysampled);
hold off;
%PAM Signal
N=length(ycont); %we take the length of the continuous signal
z=zeros(1,N); % we create an N-length vector of zeros
n=floor(Tm/Tcont); % we divide the period of the discrete signal between
the
                    % period of the continuous signal (we round it). This
```

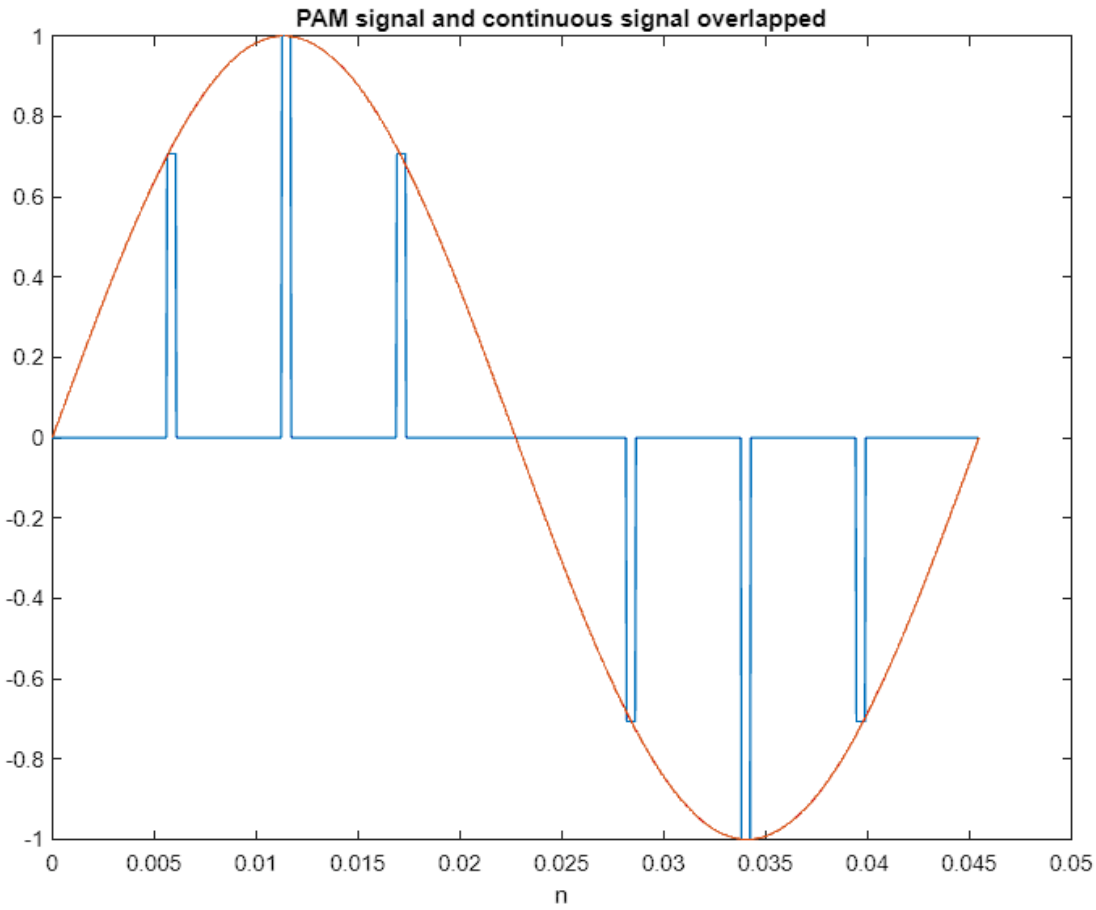
```
                    % will be the increment between the samples of the
                    % sampled signal
z(1:n:N)=ysampled; %we put the values of ysampled in the vector of zeros
every n samples;
                     %so we are adding zeros between the samples of the
                     %sampled signal
h=zeros(1,100);
h(1:10)=1; %we create a pulse with a duration of 10 samples
pamreal=conv(h,z); % we convolve the pulse with the sampled signal which
contains
                    % the zeros between each sample
figure()
plot(pamreal(1:L));title('PAM Signal'); xlabel('n')
figure(); %we are going to plot the continuous signal and the PAM signal
overlapped
plot(continuous_time_axis, pamreal(1:N));title('PAM signal and continuous
signal overlapped'); xlabel('n')
hold on;
plot(continuous_time_axis, ycont)
%FREQUENCY DOMAIN
freq_axis=[-fsim/2:fc:fsim/2];
YCONT=fft(ycont); %spectrum of the continuous signal
figure()
subplot(3,1,1)
stem(freq_axis, fftshift(abs(YCONT)));title('Spectrum of the continuous
sinusoide');xlabel('f')
freq_axis2=[-fsim/2:fsim/8:fsim/2];
YSAMPLED=fft(ysampled); %spectrum of the discrete signal which represents
8
                        %8 samples of the continous signal
subplot(3,1,2)
stem(freq_axis2,fftshift(abs(YSAMPLED)));title('Spectrum of the discrete
sinusoide');xlabel('f')
subplot(3,1,3)
```

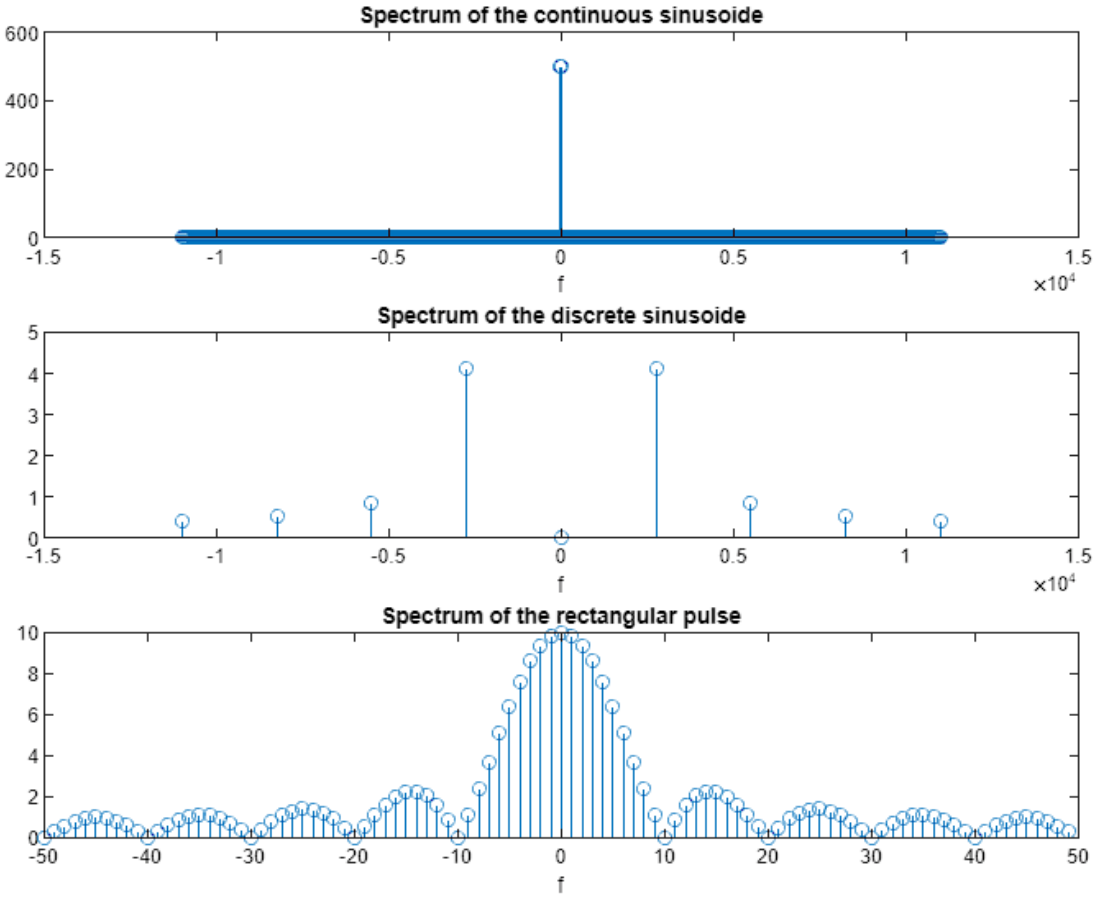
```
stem((-50:49),fftshift(abs(fft(h))));title('Spectrum of the rectangular pulse');xlabel('f')

figure()

stem((-L/2: L/2-1),fftshift(abs(fft(pamreal(1:L)))));title('Spectrum of the PAM signal'); xlabel('f')

figure()

stem(freq_axis, fftshift(abs(YCONT)));title('Spectrum of the continuous sinusoide and the PAM signal');xlabel('f')

hold on;

stem(freq_axis, fftshift(abs(fft(pamreal(1:N)))))
```

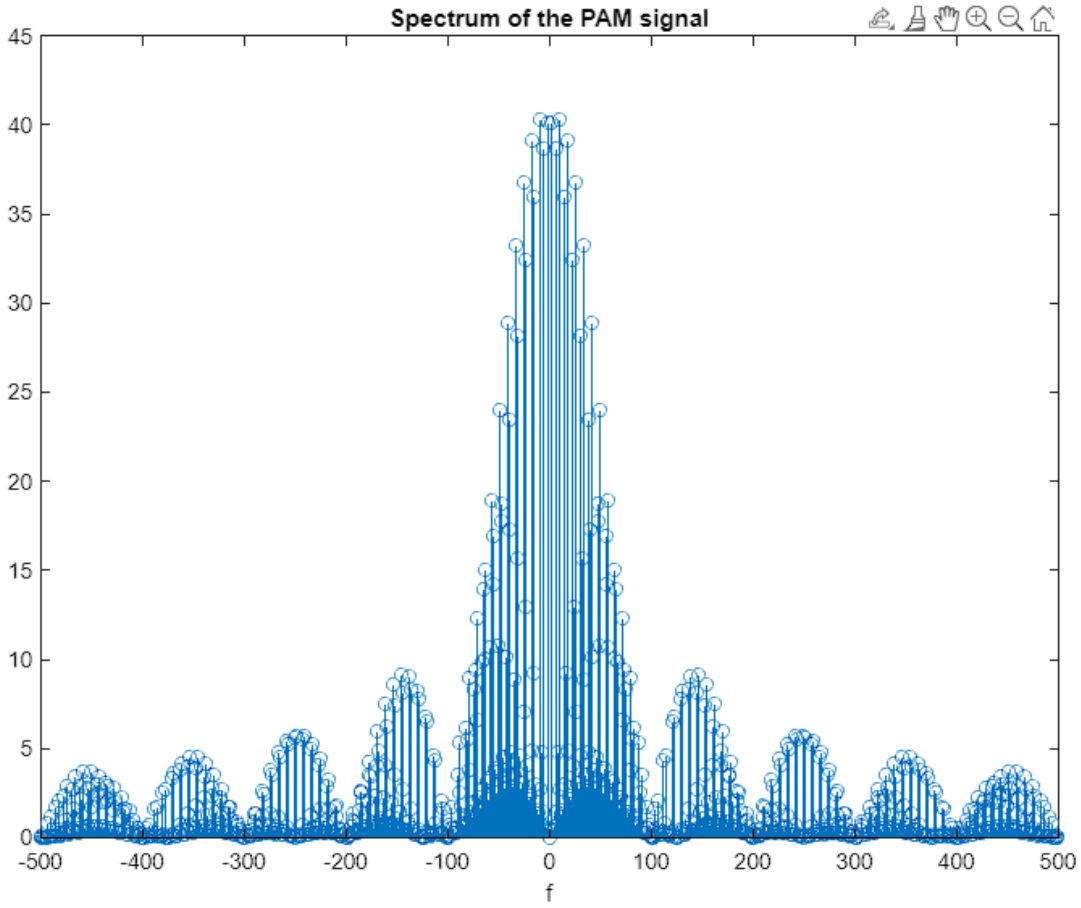
Output

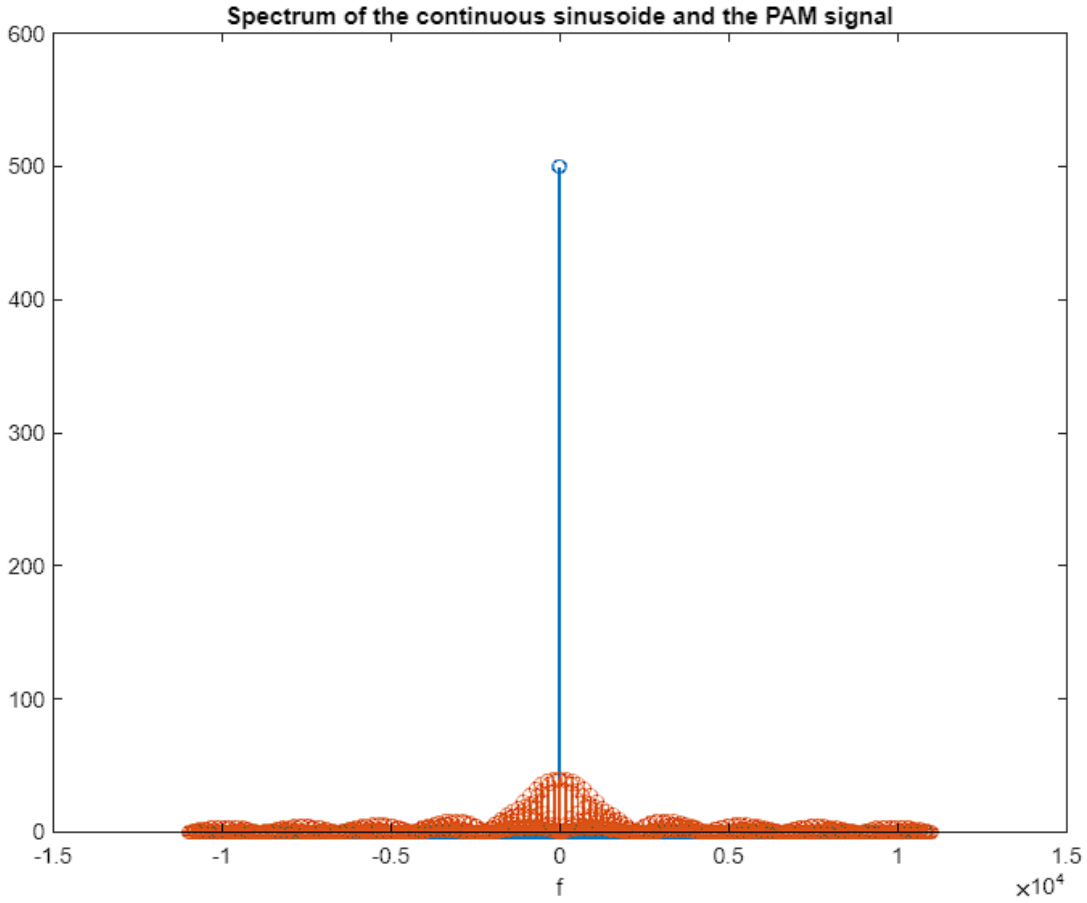

Cycle of the continuous signal with 8 samples
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
n


PAM Signal
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
100
200
300
400
500
600
700
800
900
1000
n


PAM signal and continuous signal overlapped
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
n


Spectrum of the continuous sinusoide
600
400
200
0
-1.5
-1
-0.5
0
0.5
1
1.5
f
×10^4
Spectrum of the discrete sinusoide
5
4
3
2
1
0
-1.5
-1
-0.5
0
0.5
1
1.5
f
×10^4
Spectrum of the rectangular pulse
10
8
6
4
2
0
-50
-40
-30
-20
-10
0
10
20
30
40
50
f


Spectrum of the PAM signal
45
40
35
30
25
20
15
10
5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500
f


Spectrum of the continuous sinusoide and the PAM signal
600
500
400
300
200
100
0
-1.5
-1
-0.5
0
0.5
1
1.5
f
×10^4

Aim:

To simulate Pulse Width Modulation (PWM) technique using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

Personal Computer

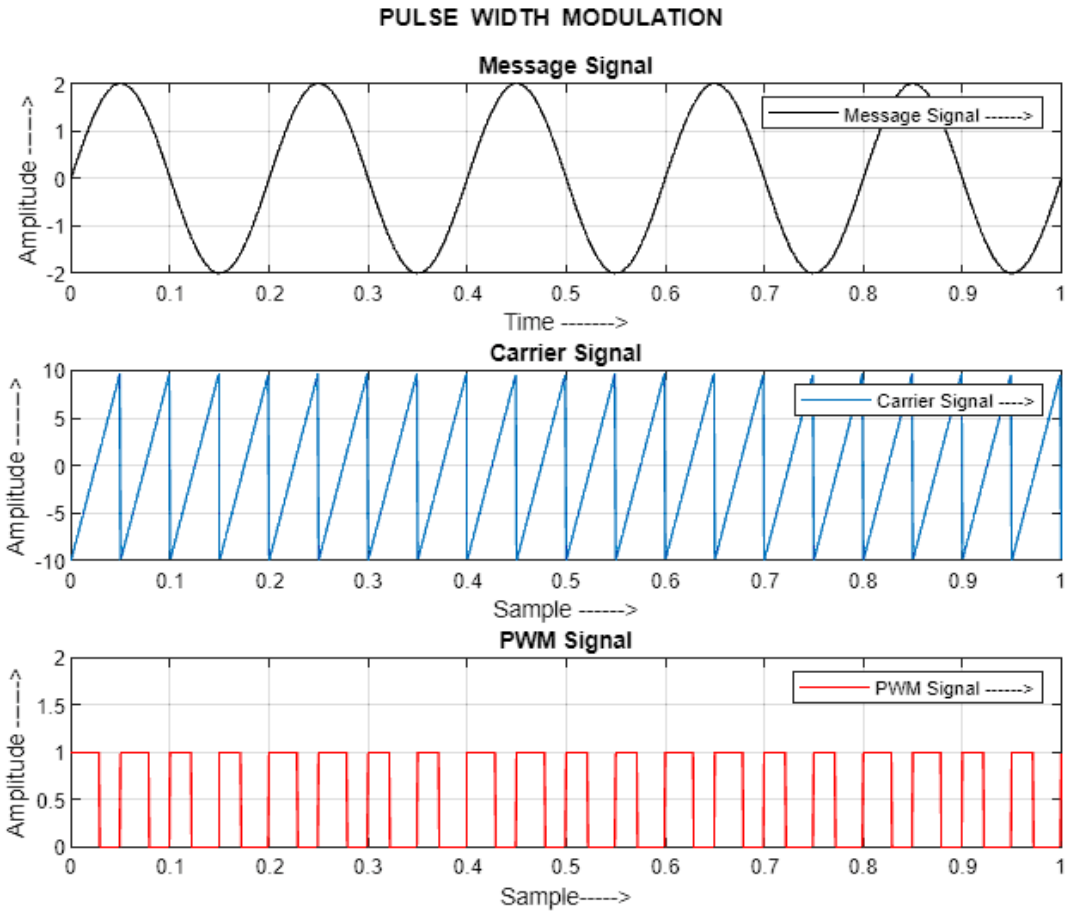
MATLAB Code

```
t = 0:0.001:1;
fc = input('Enter the Frequency of Carrier Signal(Sawtooth) = ');
fm = input("Enter the frequency of Message Signal(Sinusoidal) = ");
a = input("Enter the Amplitude of Carrier Signal = ");
b = input("Enter the Amplitude of Message Signal(should be < Carrier");
vc = a.*sawtooth(2*pi*fc*t);
vm  = b.*sin(2*pi*fm*t);
n = length(vc);

for i = 1:n
    if(vm(i) >= vc(i))
        pwm(i) = 1;
    else
        pwm(i) = 0;
    end
end
%Representation of the Message Signal
subplot(3,1,1);
plot(t,vm,'black');
xlabel("Time ------->");
ylabel('Amplitude ------>');
title('Message Signal');
legend('Message Signal ------>');
grid on;
```

```
%Representation of the Carrier Signal
subplot(3,1,2);
plot(t,vc);
xlabel('Sample ------>')
ylabel('Amplitude ------->')
title('Carrier Signal');
legend('Carrier Signal ---->');
grid on;
%Representation of the PWM Signal
subplot(3,1,3);
plot(t,pwm,'red');
xlabel('Sample----->');
ylabel('Amplitude ------>');
title('PWM Signal');;
legend('PWM Signal ------>');
axis([0 1 0 2]);
grid on;
%Add title to the Overall Plot
ha = axes ('Position',[0 0 1 1],'Xlim',[0 1],'Ylim',[0
1],'Box','off','Visible','off','Units','normalized','clipping','off');
text (0.5,1,'\bf PULSE WIDTH
MODULATION','HorizontalAlignment','center','VerticalAlignment','top');
```

```
Enter the Frequency of Carrier Signal(Sawtooth) =
20
Enter the frequency of Message Signal(Sinusoidal) =
5
Enter the Amplitude of Carrier Signal =
10
Enter the Amplitude of Message Signal(should be < Carrier
2
```

Output



Aim:

To simulate Time Division Multiplexing (TDM) technique using MATLAB software.

Software and Hardware Required:

MTLAB Software (MATLAB R2022b)

Personal Computer

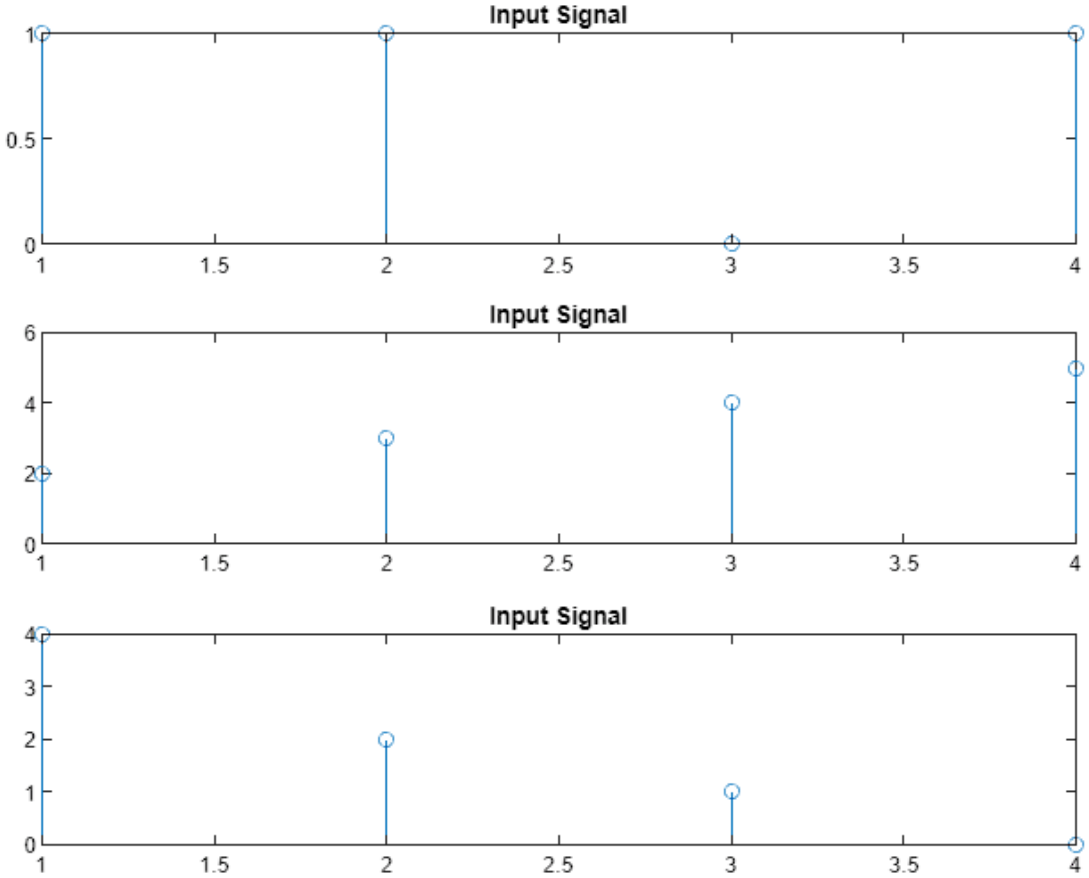
MATLAB Code

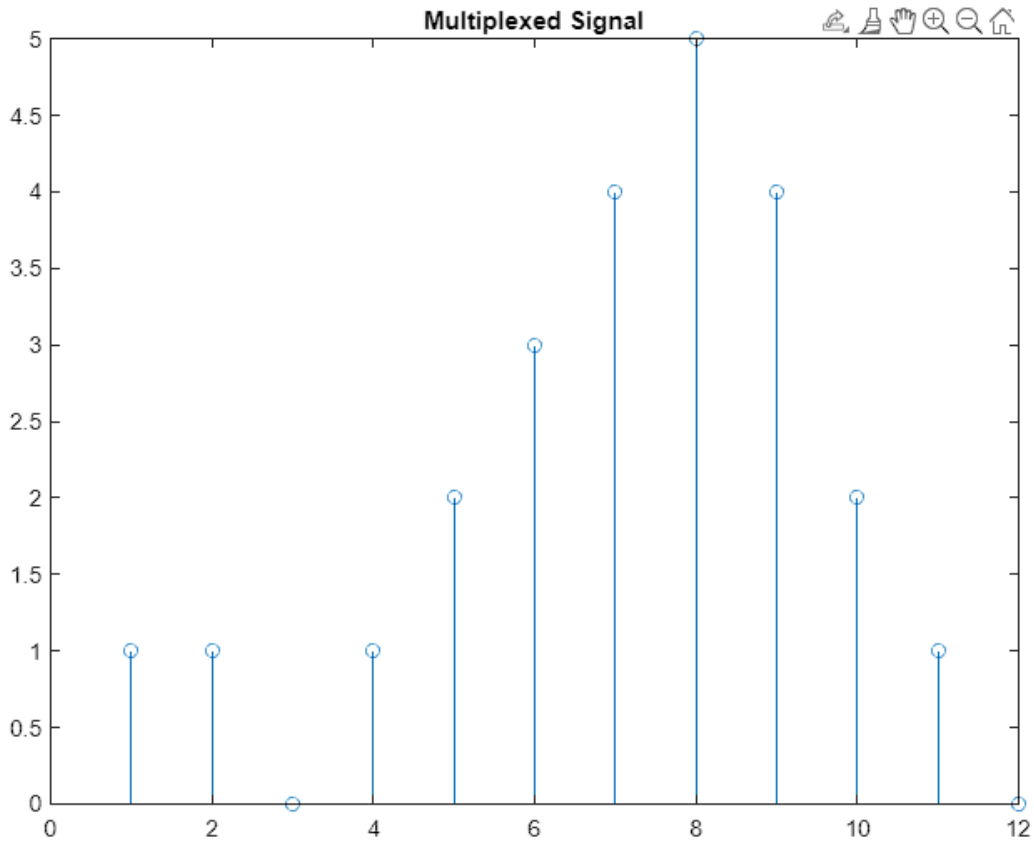
```
n=input('Enter the number the signal you multiplexed : ');
r=input('Enter the number of bits in each signal : ');
%signaling
for i=1:n
    a=input('enter data bits : ');
  for j=1:r
    a1(i,j)=a(1,j);
    j=j+1;
  end
  disp('Enter next signal data bits : ');
  i=i+1;
end
%displaying the signal
figure
for i=1:n
 for j=1:r
   a2(1,j)=a1(i,j)

   j=j+1;
 end
 subplot(n,1,i);
 stem(a2);title('Input Signal');
 i=i+1;
end
```

```
%multiplexed signal
figure
k=1;
for i=1:n
  for j=1:r
    m(1,k)=a1(i,j);
    j=j+1;
    k=k+1;
  end
  i=i+1;
end
stem(m);title('Multiplexed Signal');
```

```
Enter the number the signal you multiplexed :
3
Enter the number of bits in each signal :
4
enter data bits :
[1 1 0 1]
Enter next signal data bits :
enter data bits :
[2 3 4 5]
Enter next signal data bits :
enter data bits :
[4 2 1 0]
```

Output


Input Signal
1
0.5
0
1
1.5
2
2.5
3
3.5
4
Input Signal
6
4
2
0
1
1.5
2
2.5
3
3.5
4
Input Signal
4
3
2
1
0
1
1.5
2
2.5
3
3.5
4


Multiplexed Signal
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12