# Analog & Digital Communication Lab (IT-351)

**Submitted in partial fulfilment of the requirement of the degree of**

**BACHELOR IN TECHNOLOGY**

**COMPUTER SCIENCE & ENGINEERING**

**By**


## Anirudh Gautam

**ENROLLMENT NO. 01116403220**

**(5th Sem.)**




**University School of Information & Communication Technology**

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY DWARKA, NEW DELHI**


**2022-23**

**Under the Supervision of**


**Dr. Mansi Jhamb**

**USICT , GGSIPU**

# INDEX

# Experiment-1

**Aim**: To plot the waveforms for: Sine, Cosine, Exponential, Ramp, Unit step signal, Unit impulse signal in continuous time.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

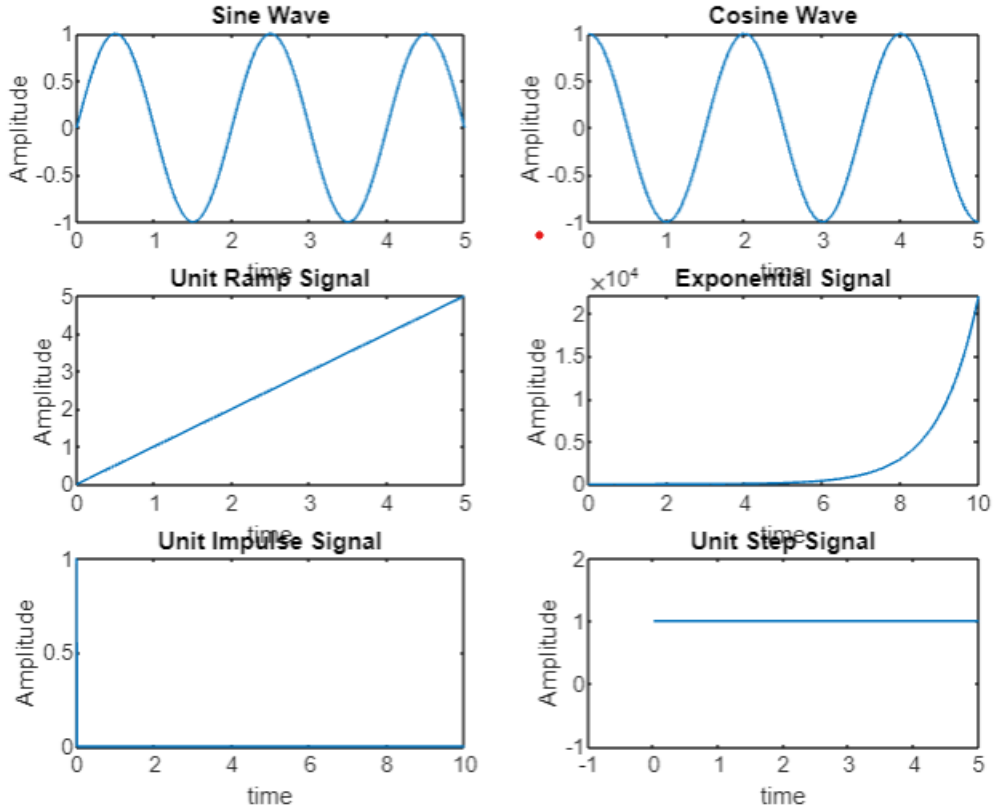
Personal Computer

## Code:

```
clc;
clear;
t=5:0.1:5;
f = 1000;
a=sin(2*pi*f*t);
b=cos(2*pi*f*t);
x=t;
y=(t==0);
z=(t>=0);
c=exp(t);
subplot(3,2,1);
plot(t,a);
axis([0 5 -1 1]);
xlabel('time');
ylabel('Amplitude');
title('Sine Wave');
subplot(3,2,2);
plot(t,b);
axis([0 5 -1 1]);
xlabel('time');
ylabel('Amplitude');
title('Cosine Wave');
subplot(3,2,3);
plot(t, x);
axis([0 5 0 5]);
xlabel('time');
ylabel('Amplitude');
title('Unit Ramp Signal')
subplot(3,2,4);
plot(t,c);
xlabel('time');
ylabel('Amplitude')
title('Exponential Signal')
subplot(3,2,5);
plot(t,y);
xlabel('time');
ylabel('Amplitude');
title('Unit Impulse Signal');
subplot(3,2,6);
plot(t,z);
axis([-1 5 -1 2]);
xlabel('time');
ylabel('Amplitude');
```

```
title('Unit Step Signal');
```

# Output:



# Experiment-2

**Aim:** To plot the waveforms for: Sine, Cosine, Exponential, Ramp, Unit step signal, Unit impulse signal in discrete time.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

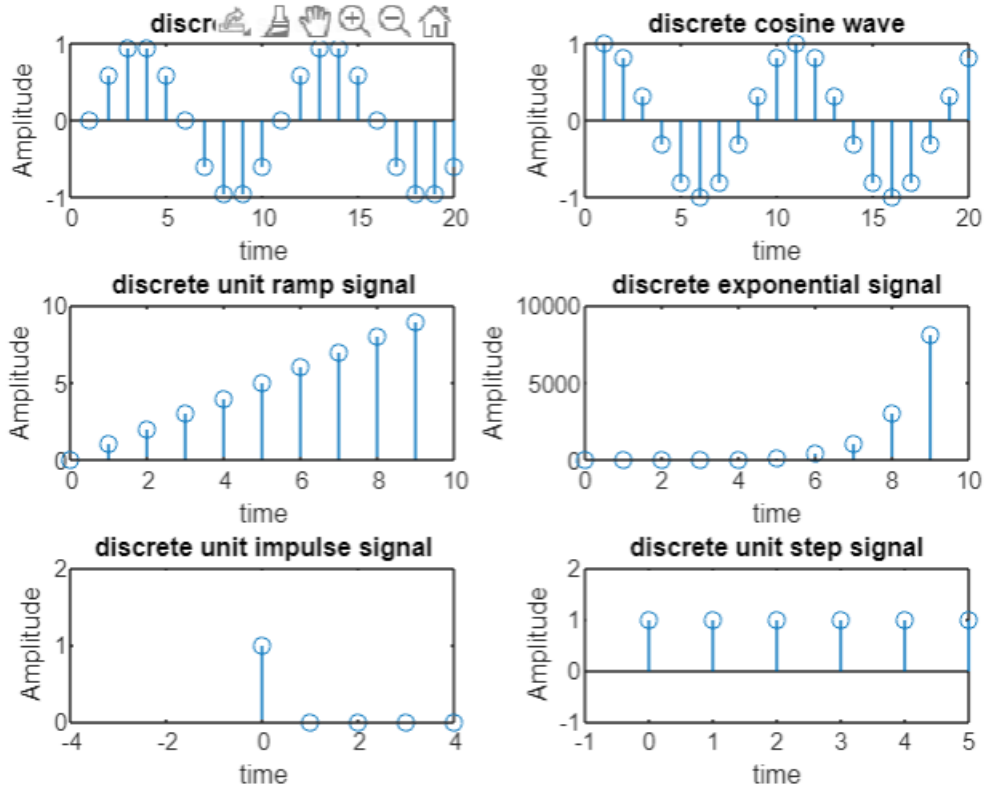
Personal Computer

## Code:

```
clc;
clear;
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(3,2,1);
stem(a);
axis([0 20 -1 1]);
xlabel('time');
ylabel('Amplitude');
title('discrete sine wave');
subplot(3,2,2);
stem(b);
axis([0 20 -1 1]);
xlabel('time');
ylabel('Amplitude');
title('discrete cosine wave');
subplot(3,2,3);
stem(n,x);
axis([0 10 0 10]);
xlabel('time');
ylabel('Amplitude');
title('discrete unit ramp signal');
subplot(3,2,4);
stem(n,c);
xlabel('time');
ylabel('Amplitude');
title('discrete exponential signal');
subplot(3,2, 5);
stem(n,y);
axis([-4 4 0 2]);
xlabel('time');
ylabel('Amplitude');
title('discrete unit impulse signal');
subplot(3,2,6);
stem(n,z);
axis([-1 5 -1 2]);
xlabel('time');
```

```
ylabel('Amplitude');
title('discrete unit step signal');
```

# Output:



# Experiment-3

**Aim:** To perform amplitude modulation using MATLAB.

## Software & Hardware Required:

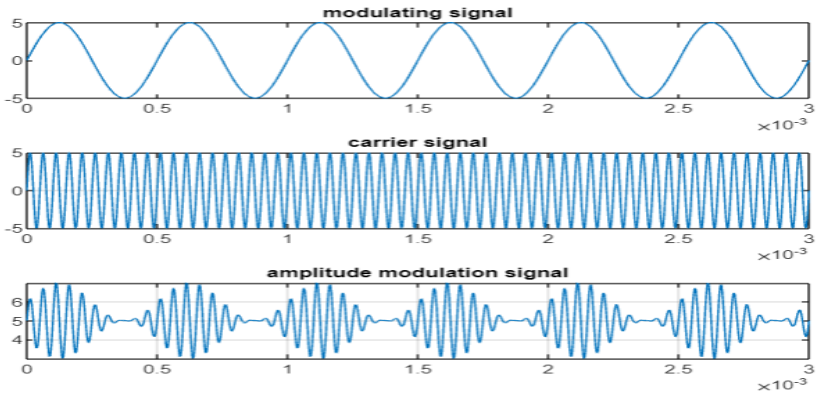
MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

```
clc;
clear;
close all;
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:



# Experiment-4

**Aim:** To perform frequency modulation using MATLAB.

## Software & Hardware Required:

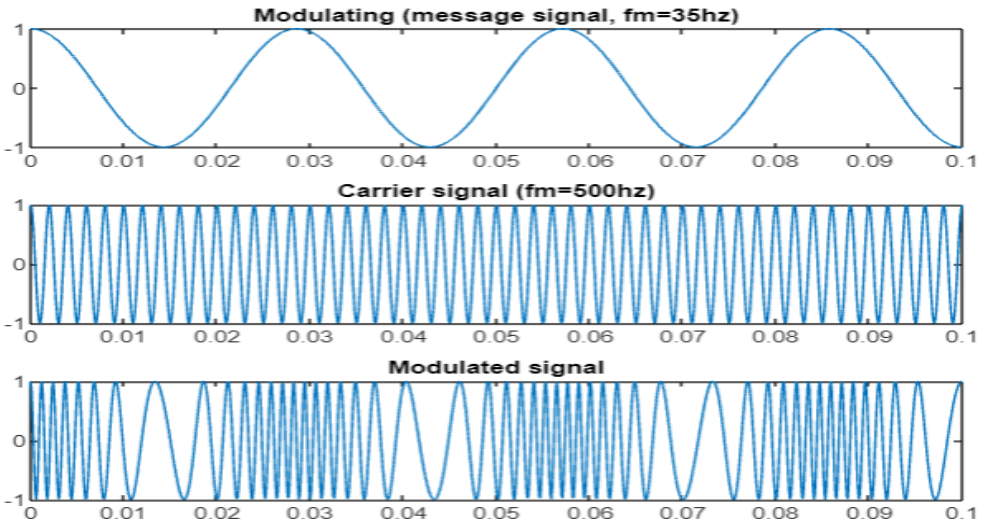
MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

```
% code for FM
%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(3,1,1);
plot(t, m_t);
title('Modulating (message signal, fm=35hz)');
c_t = Ac*cos(wc*t);
subplot(3,1,2);
plot(t,c_t);
title('Carrier signal (fm=500hz)');
s_t = Ac*cos((wc*t)+B*sin(wm*t));
subplot(3,1,3);
plot(t,s_t);
title('Modulated signal');
```

## Output:




# Experiment-5

**Aim:** To simulate Pulse code modulation (PCM) technique using MATLAB.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

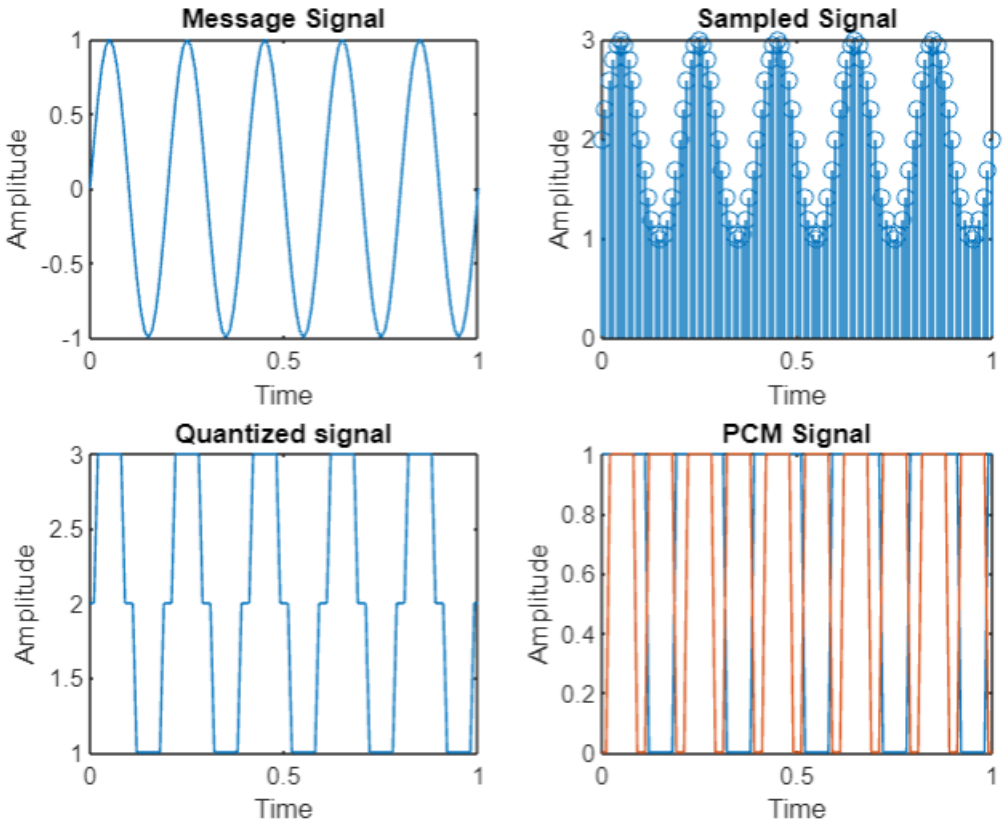
Personal Computer

## Code:

```
clc;
clear;
t = 0:0.01:1;
f = 5;
a = sin(2*pi*f*t);
subplot(2,2,1);
plot(t,a);
xlabel('Time');
ylabel('Amplitude');
title('Message Signal');
b = 2;
c = a+b;
subplot(2,2,2);
stem(t,c);
xlabel('Time');
ylabel('Amplitude');
title('Sampled Signal');
d = quant(c);
subplot(2,2,3);
plot(t,d);
xlabel('Time');
ylabel('Amplitude');
```

```
title('Quantized signal');
e = de2bi(d, 'left-msb');
subplot(2,2,4);
plot(t,e);
axis([0 1 0 1]);
xlabel('Time');
ylabel('Amplitude');
title('PCM Signal');
```

# Output:



# Experiment-6

**Aim:** To simulate amplitude shift key (ASK) technique using MATLAB.

## Software & Hardware Required:

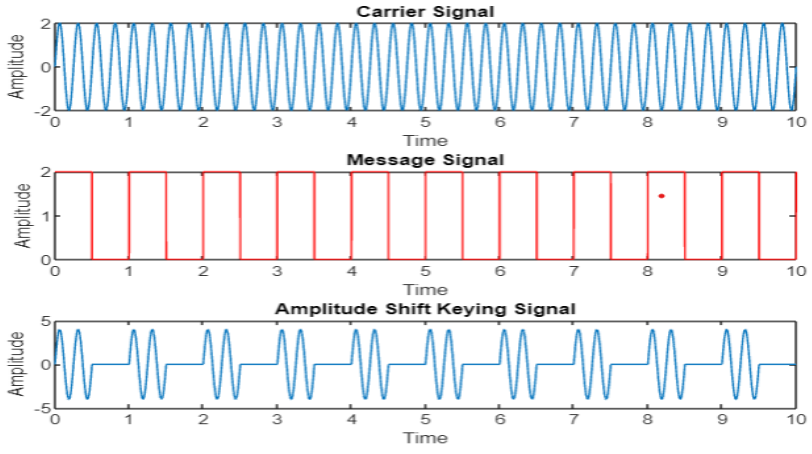
MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

```
clc;
clear;
t = 0:0.001:10;
b = 2;
fc = 4; %sinusoidal signal
fm = 1; % pulse signal
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 Signal');
xlabel('Time');
ylabel('Amplitude');
```

## Output:



# Experiment-7

**Aim:** To simulate frequency shift keying (FSK) technique using MATLAB.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

Personal Computer

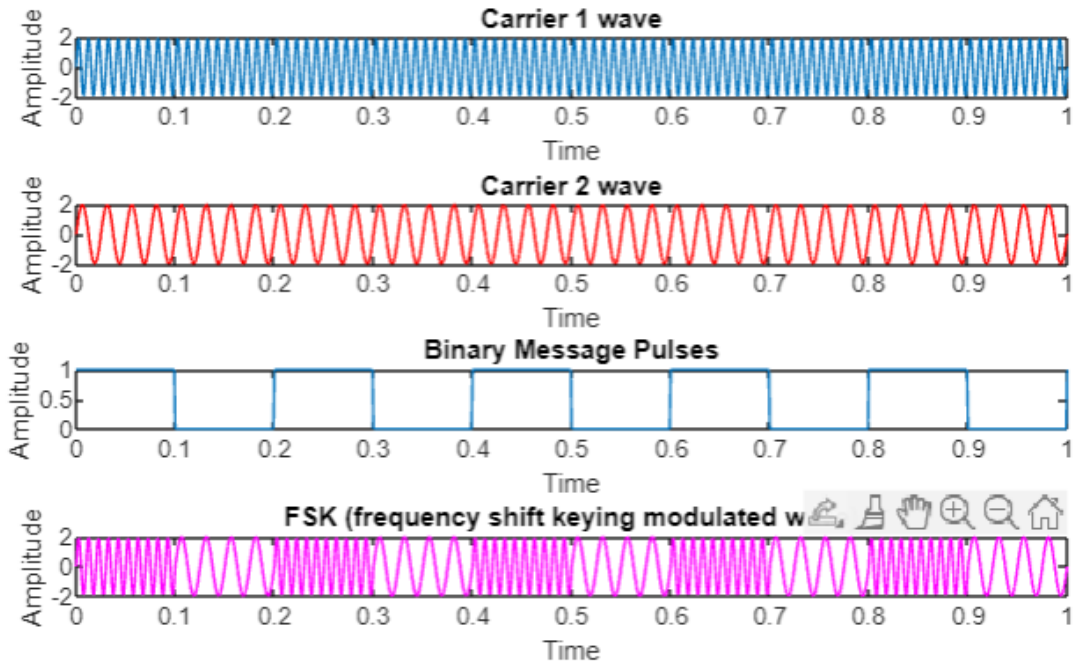
## Code:

```
clc;
clear;
fc1 = input('Enter the frequency of first sine wave carrier high frequency: ');
fc2 = input('Enter the frequency of second 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 sampling interval
c1 = amp.*sin(2*pi*fc1*t); %First carrier sine wave
c2 = amp.*sin(2*pi*fc2*t); %second carrier sine wave
subplot(5,1,1);
plot(t,c1);
xlabel('Time');
ylabel('Amplitude');
title('Carrier 1 wave');
subplot(5,1,2);
plot(t,c2,'r');
xlabel('Time');
ylabel('Amplitude');
title('Carrier 2 wave');
m = a.*square(2*pi*fp*t)+a;
subplot(5,1,3);
plot(t,m);
xlabel('Time');
ylabel('Amplitude');
title('Binary Message Pulses');
n = length(t);
for i=1:n
   if m(i)==0
      f(i)=c2(i);
   else
      f(i)=c1(i);
   end
end
subplot(5,1,4);
plot(t,f,'m');
xlabel('Time');
ylabel('Amplitude');
title('FSK (frequency shift keying modulated wave)');
```

# Output:

```
Enter the frequency of first sine wave carrier high frequency:
100
Enter the frequency of second sine wave carrier lower frequency:
40
Enter the freq. of periodic binary pulse (message):
```

```
Enter the freq. of periodic binary pulse (message):
5
```



# Experiment-8

**Aim:** To simulate phase shift keying (PSK) using MATLAB.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

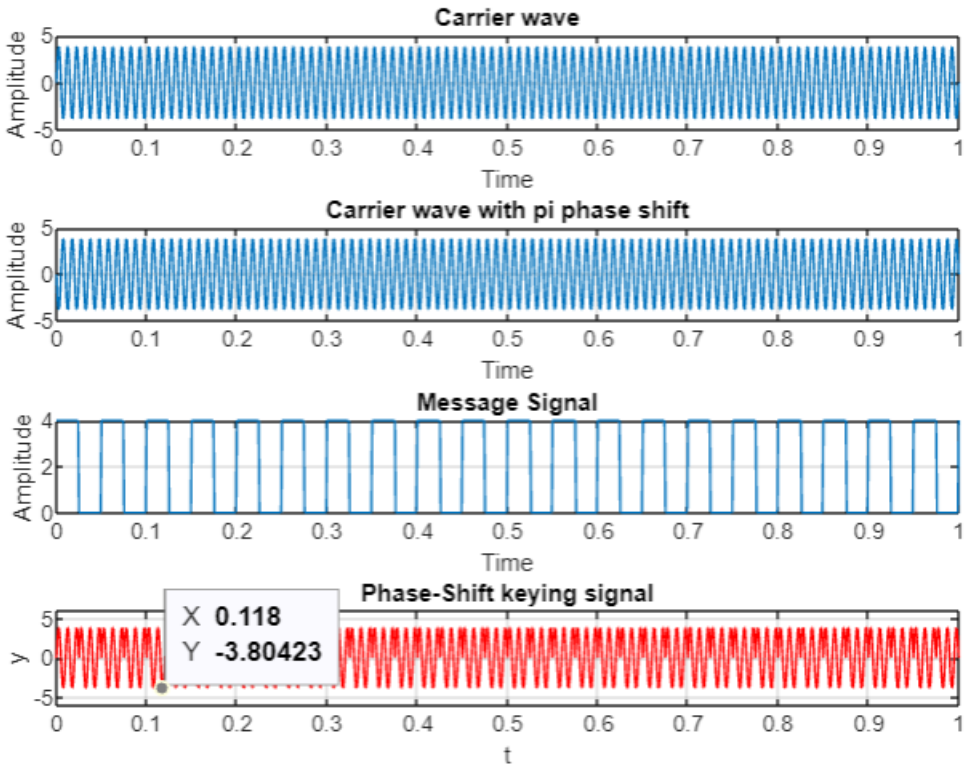
Personal Computer

## Code:

```
clc;
clear;
A = 4;
t = 0:0.001:1;
f1 = input('Carrier Sine wave frequency= ');
f2 = input('Message Frequency= ');
x = A.*sin(2*pi*f1*t);
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;
subplot(4,1,3);
plot(t,m);
xlabel('Time');
ylabel('Amplitude');
title('Message Signal');
grid on;

n = length(t);
for i=1:n
   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;
```

# Output:

```
Carrier Sine wave frequency=
100
Message Frequency=
20
>>
```



# Experiment-9

**Aim:** To simulate Pulse Amplitude Modulation (PAM) technique using MATLAB.

## Software & Hardware Required:

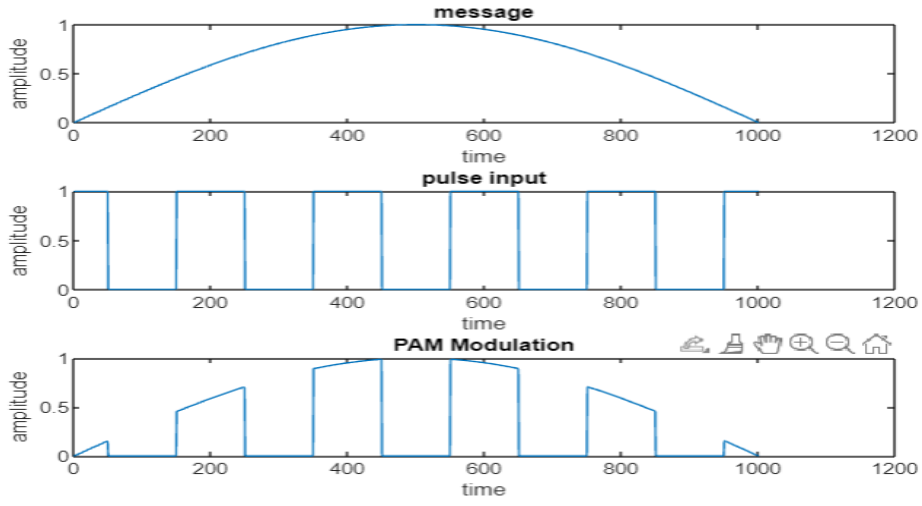
MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

```
clc;
clear;
t = 0:1/1e3:1;
d = 0:1/5:1;
x = sin(2*pi/4*2*t);
figure;
subplot(3,1,1);
plot(x);
title('message');
xlabel('time');
ylabel('amplitude');
y = pulstran(t,d,'rectpuls',0.1);
subplot(3,1,2);
plot(y);
title('pulse input');
xlabel('time');
ylabel('amplitude');
z = x.*y;
subplot(3,1,3);
plot(z);
title('PAM Modulation');
xlabel('time');
ylabel('amplitude');
```

## Output:



# Experiment-10

**Aim:** To simulate Pulse Width Modulation (PWM) using MATLAB.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

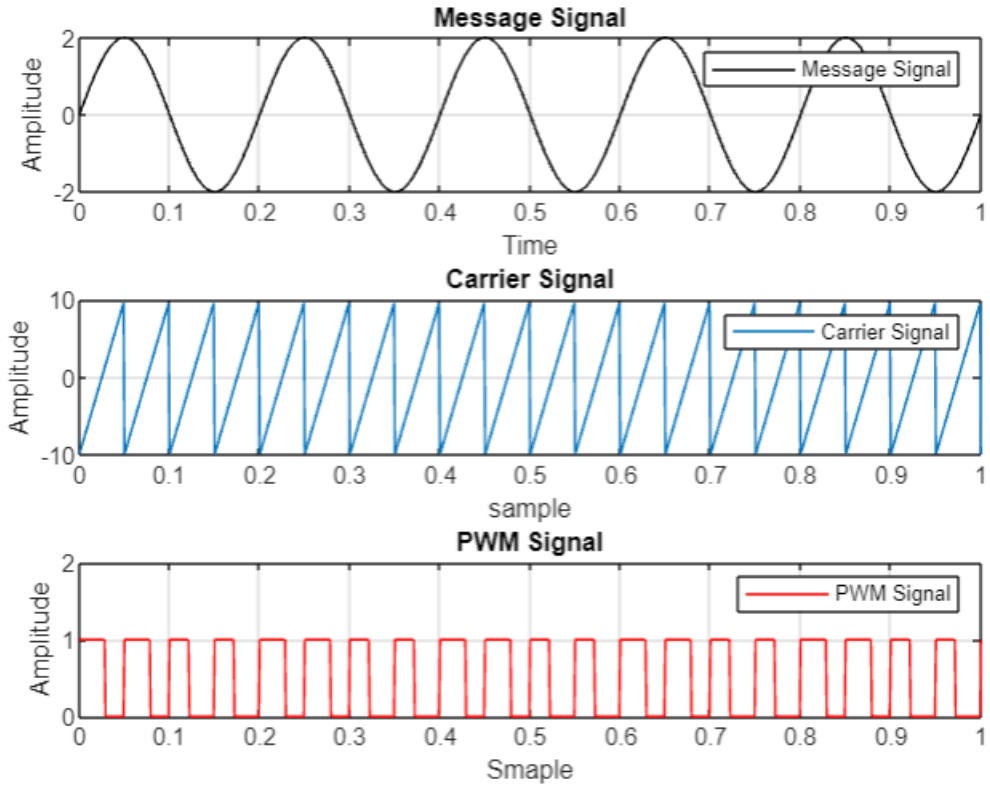
```
clc;
clear;
t = 0:0.001:1;
fc = input('Enter the frequency of carrier signal=> ');
fm = input('Enter the frequency of message signal=> ');
a = input('Enter the amplitude of carrier signal=> ');
b = input('Enter the Amplitude of meesgae signal (< carrier)=> ');
vc = a.*sawtooth(2*pi*fc*t);
vm = b.*sin(2*pi*fm*t);
n = length(vc);
for i=1:n
    if vm(i)>=vc(i)
        pwm(i) = 1;
    else
        pwm(i) = 0;
    end
end
subplot(3,1,1);
plot(t, vm, 'black');
xlabel('Time');
ylabel('Amplitude');
title('Message Signal');
legend('Message Signal');
grid on;

subplot(3,1,2);
plot(t, vc);
xlabel('sample');
ylabel('Amplitude');
title('Carrier Signal');
legend('Carrier Signal');
grid on;

subplot(3,1,3);
plot(t, pwm, 'red');
xlabel('Smaple');
ylabel('Amplitude');
title('PWM Signal');
legend('PWM Signal');
axis([0 1 0 2]);
grid on;
```

```
ha = axes('Position', [0 0 1 1], 'Xlim', [0 1], 'Ylim', [0 1], 'Box', 'off',
'visible','off','Units','normalized','clipping','off');
```

# Output:

```
Enter the frequency of carrier signal=>
20
Enter the frequency of message signal=>
5
```

```
Enter the amplitude of carrier signal=>
10
Enter the Amplitude of meesgae signal (< carrier)=>
2
>>
```



# Experiment-11

**Aim:** To simulate Time Division Multiplexing (TDM) using MATLAB.

## Software & Hardware Required:

MATLAB Software (MATLAB R2017a)

Personal Computer

## Code:

```
clc;
clear;
n = input('Enter the number of signal multiplexed: ');
r = input('Enter the number of bits in each signal: ');
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
for i=1:n
    for j=1:r
        a2(1,j) = a1(i,j);
        j = j+1;
    end
    subplot(n,1,i);
    stem(a2, 'r');
    title('Input Signal');
    i = i+1;
end
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');
```

## Output:

```
Enter the number of signal 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]
Enter next signal data bits:
>>
```