Pre – Requirements

1. System with internet connection.
2. Basic knowledge of MATLAB and its software or online MATLAB access.

Syntax Used in programs

- **plot (X, Y):** creates a 2-D line plot of the data in Y versus the corresponding values in X.
- **stem(signal):** plots a continuous time signal using the given time constraint and signal.
- **subplot (m, n, p):** divides the current figure into an m -by- n grid and creates axes in the position specified by p.
- **xlabel (txt)**: labels the x-axis of the current axes or standalone visualization.
- **ylabel (txt):** labels the y-axis of the current axes or standalone visualization.
- **title (title text):** adds the specified title to the current axes or standalone visualization.
- **clc(Clear command window): clc** clears all input and output from the Command Window display, giving you a "clean screen".
- **clear all: clear** removes all variables from the current workspace, releasing them from system memory.

AIM: For a frequency of 1000Hz, generate sawtooth, rectangular and triangular pulses within width 0.3.

CODE:

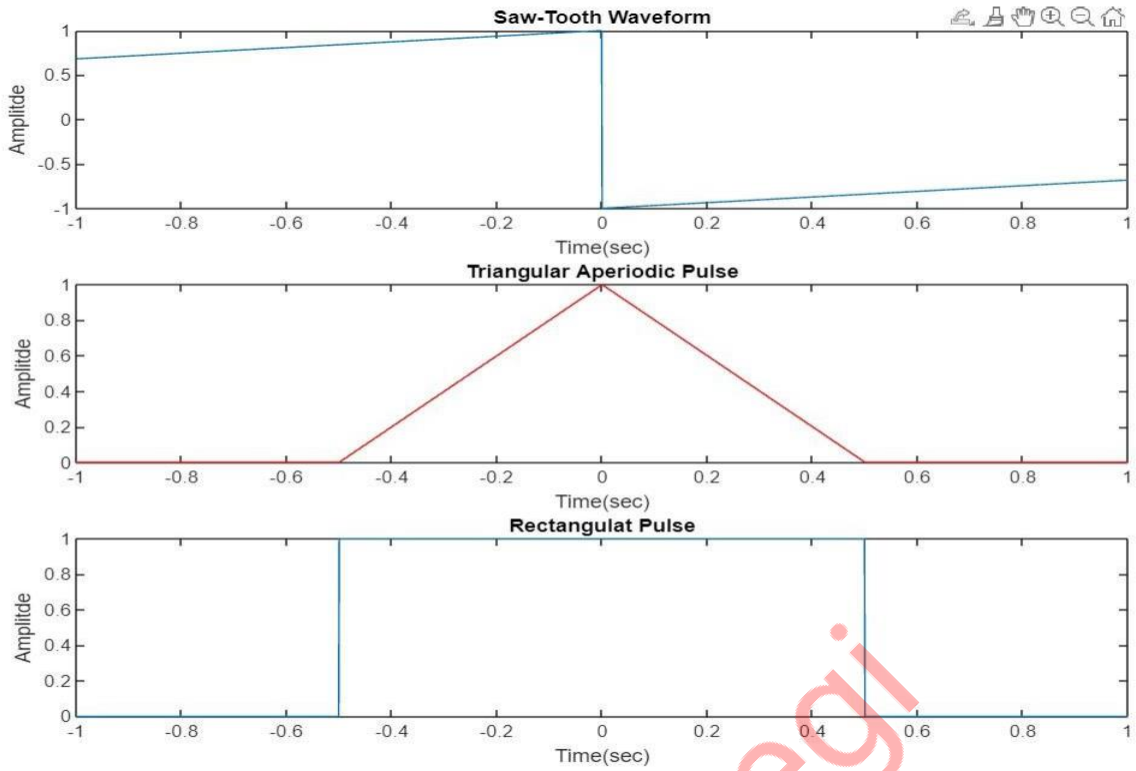
```
f=1000;
t=-1:1/f:1;

% Sawtooth
k=sawtooth(t);
subplot(3,1,1);
plot(t,k);
xlabel('Time(sec)');
ylabel('Amplitde');
title ('Saw-Tooth Waveform');

% Triangular pulse
k=tripuls(t,w);
subplot(3,1,2);
plot(t,k,'r');
xlabel('Time(sec)');
ylabel('Amplitde');
title ('Triangular Aperiodic Pulse');

% Rectangular Pulse
k=rectpuls(t,w);
subplot(3,1,3);
plot(t,k);
xlabel('Time(sec)');
ylabel ('Amplitde');
title ('Rectangular Pulse');
```

OUTPUT:



ANALYSIS:

In this experiment, I got to know the various keywords used in the first experiment and about the characteristics of sawtooth, triangular pulse and rectangular pulse.

AIM: Consider any two sinusoidal input signals of your own and carry out addition two input signal.

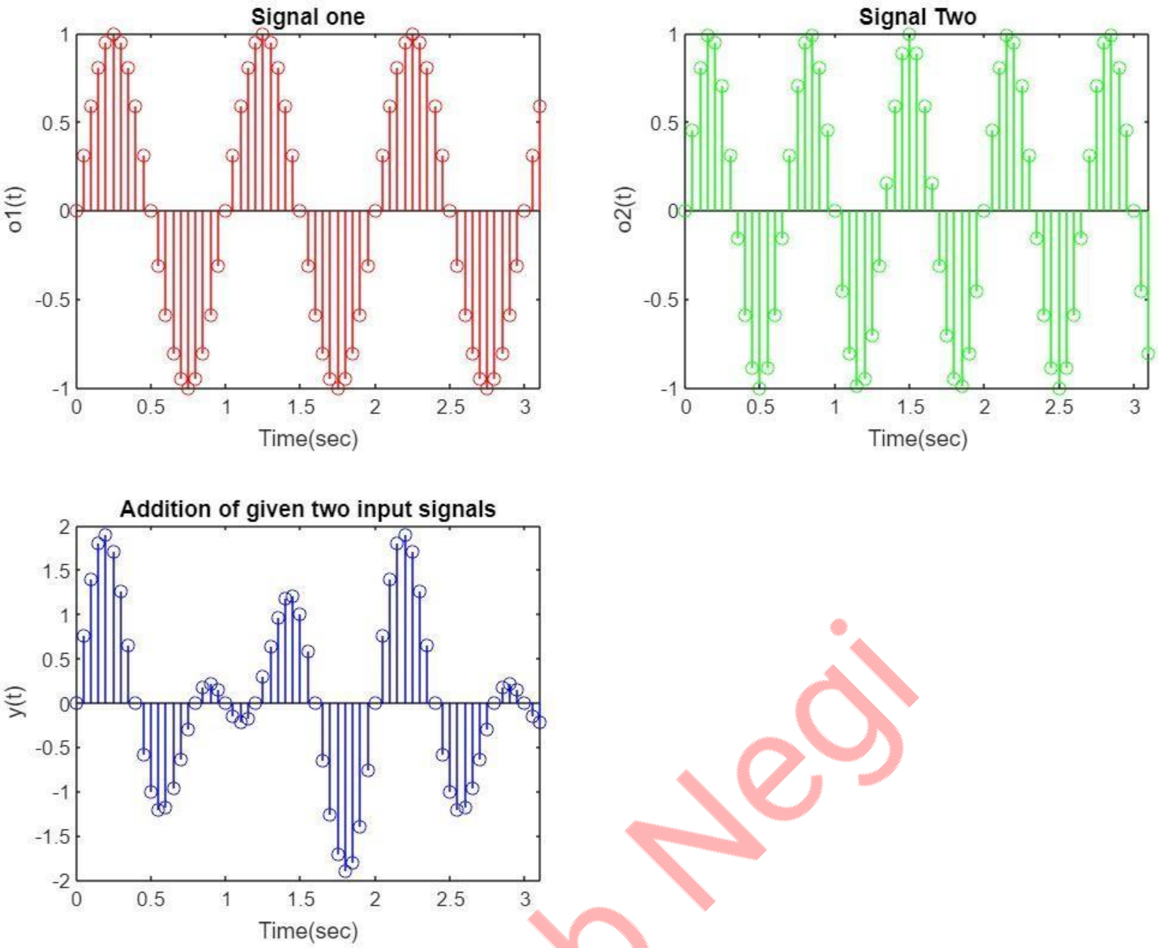
CODE:

```
clc; clear All; t=0:0.01: pi;
o1=sin(2*pi*t);
subplot(2,2,1);
stem (t, o1, "r ");
xlabel('Time(sec)');
ylabel('o1(t)');
title ("Signal one");

o2=sin(3*pi*t);
subplot(2,2,2);
stem (t, o2, "g ");
xlabel('Time(sec)');
ylabel('o2(t)');
title ("Signal Two");

%adding two signals
y= o1 + o2;
subplot(2,2,3);
stem (t, y, " b");
xlabel('Time(sec)');
ylabel('y(t)');
title ("Addition of given two input signals");
```

OUTPUT:



ANALYSIS:

In this experiment, we plotted the sinusoidal waves of different functions with different time periods. We used stem function to get the waves as discrete functions. And then added the two waveforms to get the resultant waveform using the "+" operator.

AIM: Perform cross-correlation and auto-correlation as given signals applying MATLAB function.

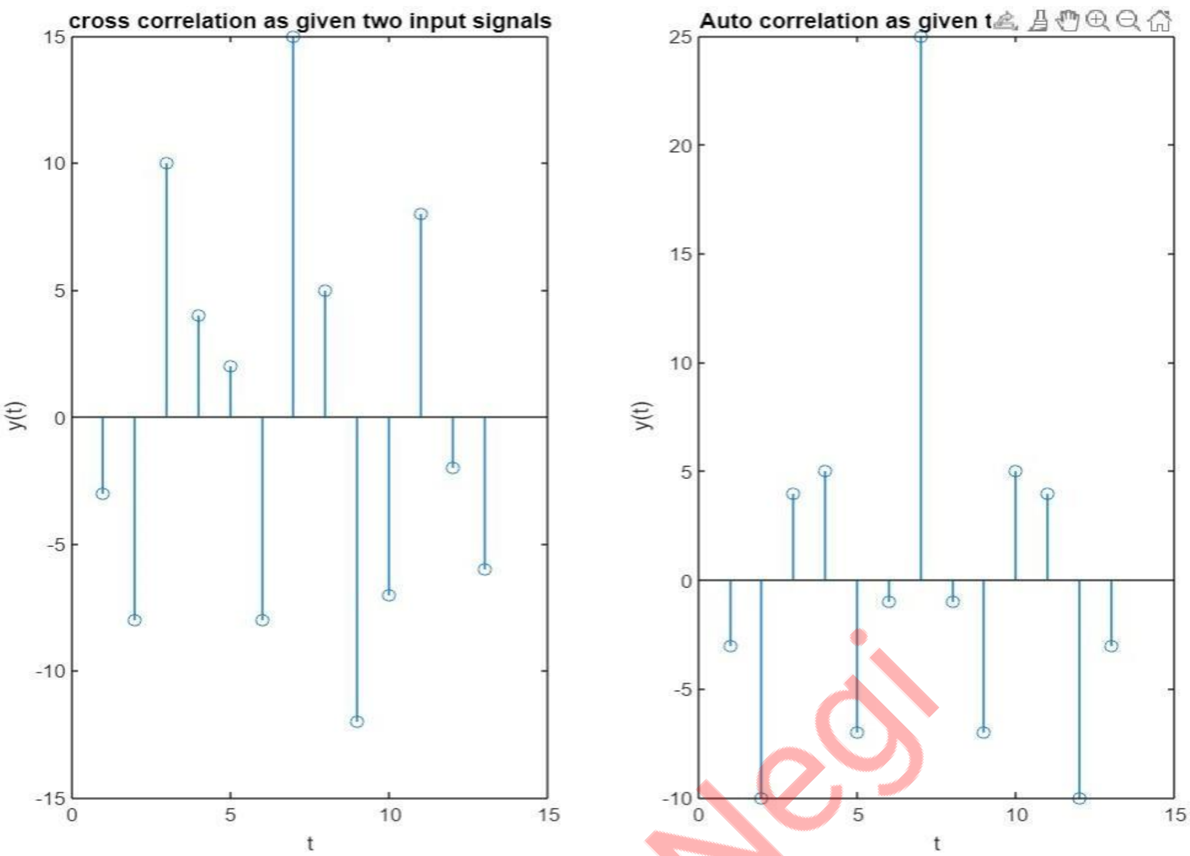
CODE:

```
a=input('enter 1st value:');
b=input('enter 2nd value:');
disp('cross correlation as given two input signals');
y=xcorr(a,b);
subplot(1,2,1);
stem(y);
xlabel('t'); ylabel('y(t)');
title('cross correlation as given two input signals');

disp('Auto Correlation as given two Input Signals');
y=xcorr(a,b);
subplot(1,2,2);
stem(y);
xlabel('t');
ylabel('y(t)');
title('Auto correlation as given two input signals');
```

INPUT: a: [-1,-3,2,0,-1,1,3] :: b: [-2,0,2,-3,-1,-1,3]

OUTPUT:



ANALYSIS:

In this experiment, I got to know about correlation i.e; the degree of similarity between the two curves, and we implemented it using the xcorr() keyword. In cross correlation, similarity is checked between two different curves. In auto correlation, similarity is checked between a signal and the time-shifted same signal. In the output we get the plots same in both the curves.

<u>AIM:</u> Perform circular-convolution of given two sequences applying MATLAB function.

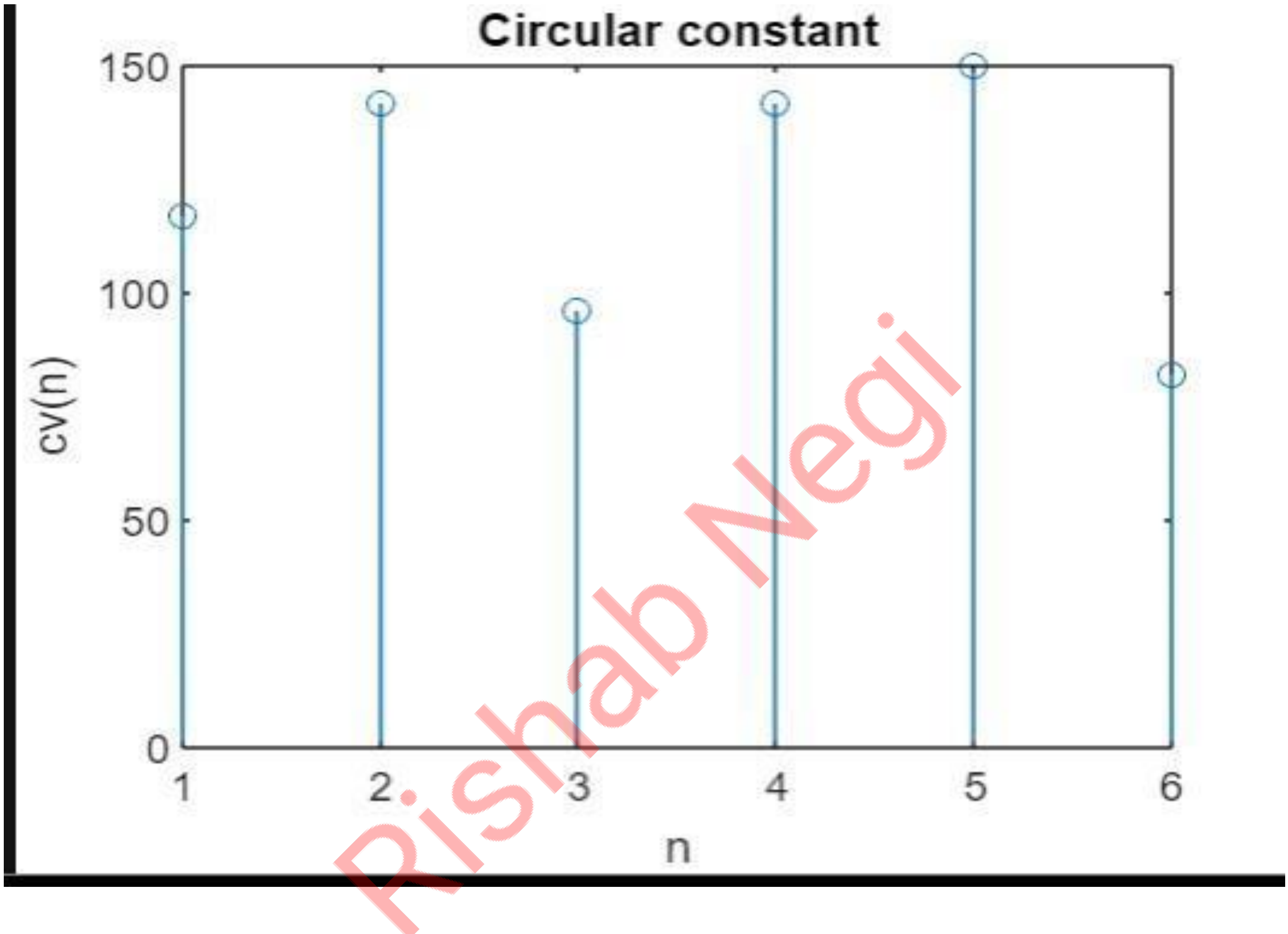
<u>CODE:</u>

```
a=[1,3,5,2,7,9];
b=[3,1,7,2,5,4];
n1=length(a);
n2=length(b);
nn=max(n1,n2);
cc=[a zeros(1,(nn-n1))];

for i=1:nn
k=i;
for j=1:n2
o(i,j)=cc(k)*b(j);
k=k-1;
y (k==o);
k=nn;
end
end
cv=zeros(1,nn);
m=0;
for j=1:nn
for i=1:n2
cv(j)=m(i,j)+cv(j);
end
end
disp('the output sequence is ');
disp(cv);
```

```
stem(cv);
title('circular convolution');
xlabel ('n');
ylabel ('cv(n)');
```

OUTPUT:



ANALYSIS:

In this experiment, I got to know about circular convolution.

AIM: Write a MATLAB program to compute DRT And IDRT of a sequence.

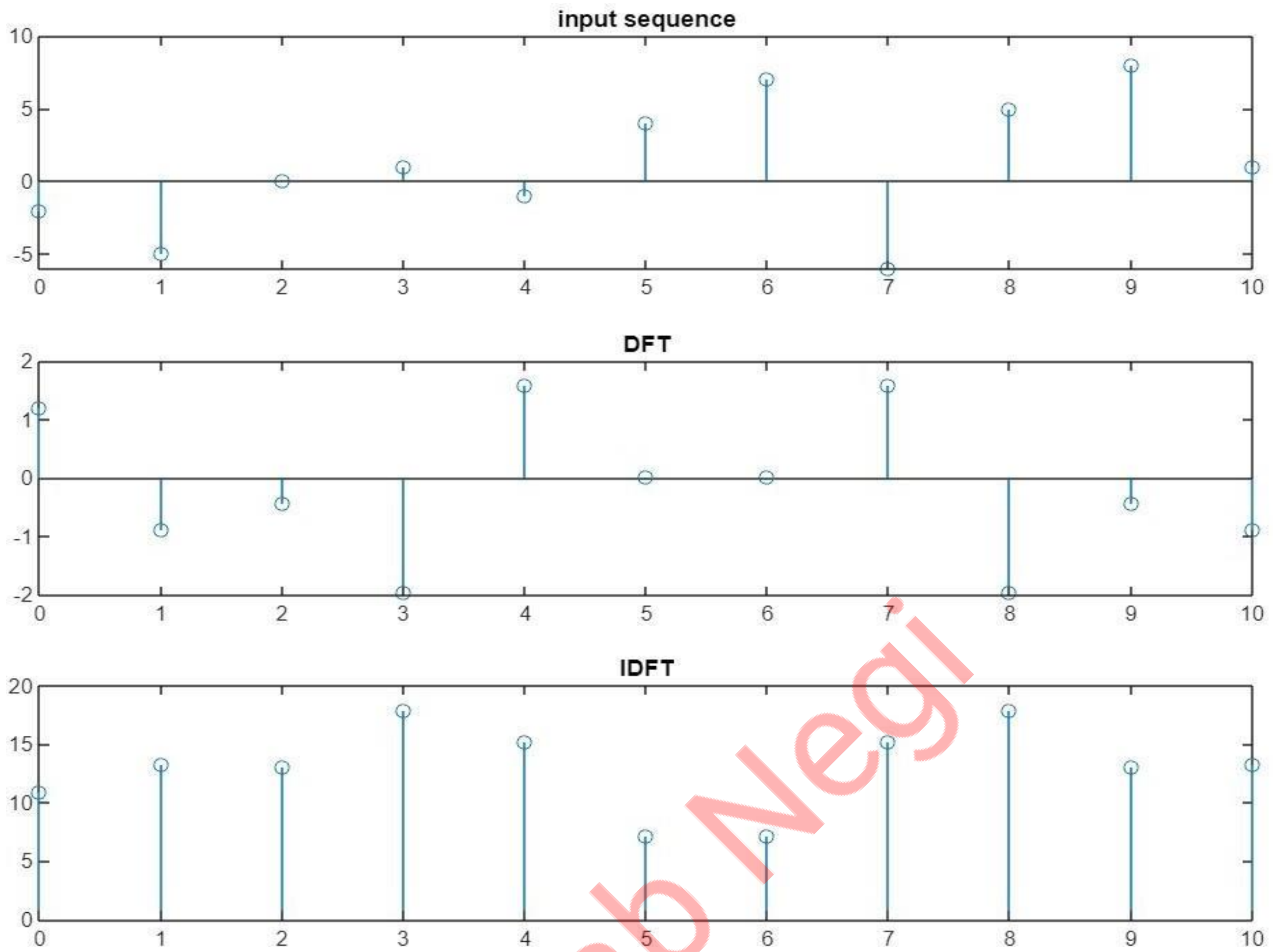
CODE:

```
clc;
clear All;
x=input("enter the sequence");
ln=input("enter the length of DFT");
y=fft(x)/ln;
n=0:length(x)-1;
xr=ln*ifft(x);
subplot(3,1,1);
stem(n,x);
title("input sequence");
subplot(3,1,2)
stem(n,y);
title("DFT");
subplot(3,1,3);
stem(n,abs(xr));
title("IDFT");
```


INPUT: enter the sequence: [-2,-5,0,1,-1,4,7,-6,5,8,1]

enter the length of DFT: 10

OUTPUT:


input sequence
10
5
0
-5
0 1 2 3 4 5 6 7 8 9 10
DFT
2
1
0
-1
-2
0 1 2 3 4 5 6 7 8 9 10
IDFT
20
15
10
5
0
0 1 2 3 4 5 6 7 8 9 10
Rishab Negi

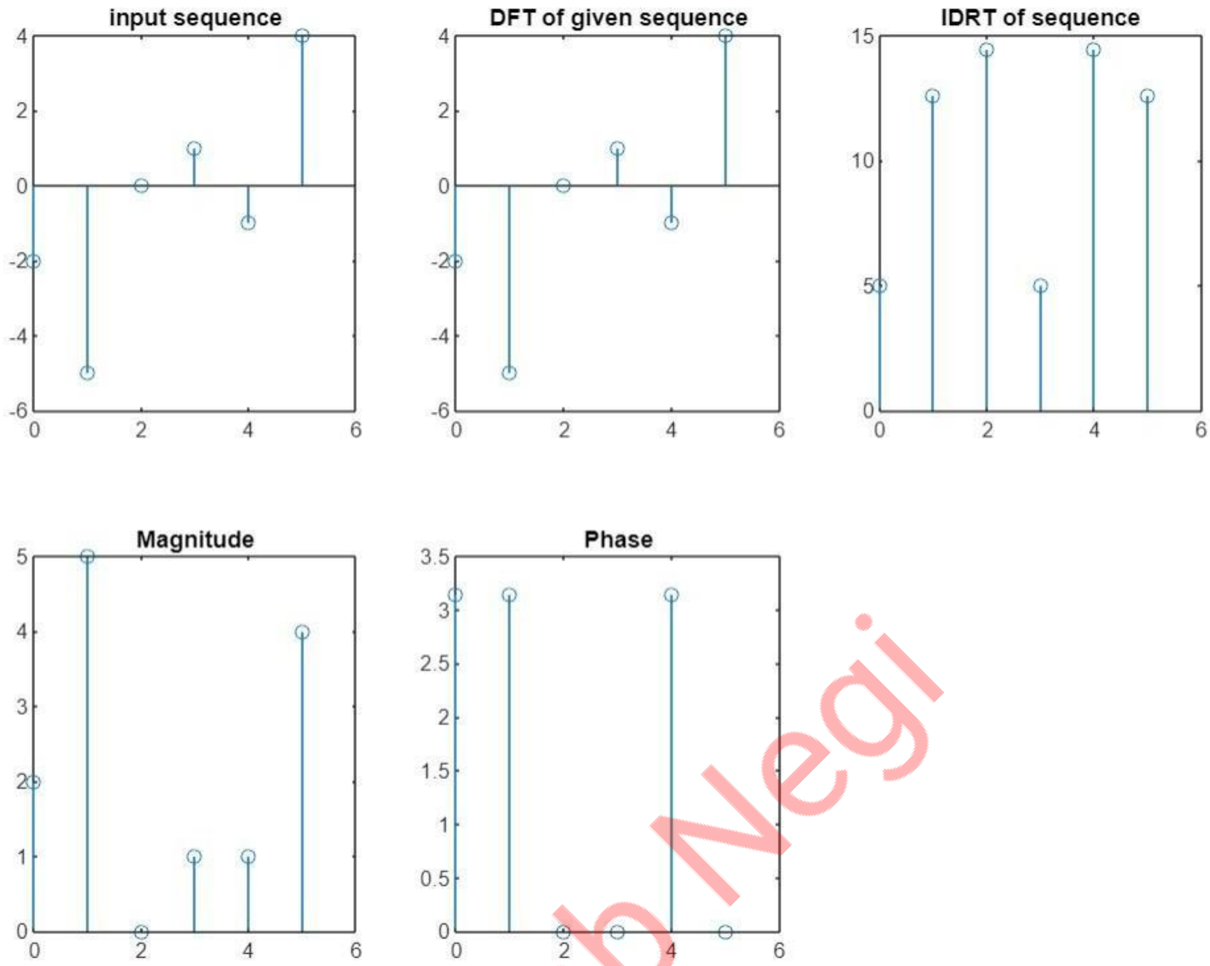
AIM: Write a MATLAB program to compute DFT and IDFT with their frequency and magnitude of a sequence.

CODE:

```
clc; clear All;
x=input("enter the sequence");
nn=input("enter the length of DFT");
y=fft(x)/ln; n=0:length(x)-1;
xinv=nn*ifft(x);
subplot(2,3,1);
stem(n,x);
title("input sequence");
subplot(2,3,2)
stem(n,x);
title("DFT of given sequence");
subplot(2,3,3);
stem(n,abs(xinv));
title("IDFT of sequence");
subplot(2,3,4);
stem(n,abs(x));
title("Magnitude");
subplot(2,3,5);
stem(n,angle(x));
title("Phase");
```


INPUT: enter the sequence: [-2,-5,0,1,-1,4]
enter the length of DRT: 10

OUTPUT:


input sequence
DFT of given sequence
IDRT of sequence
Magnitude
Phase

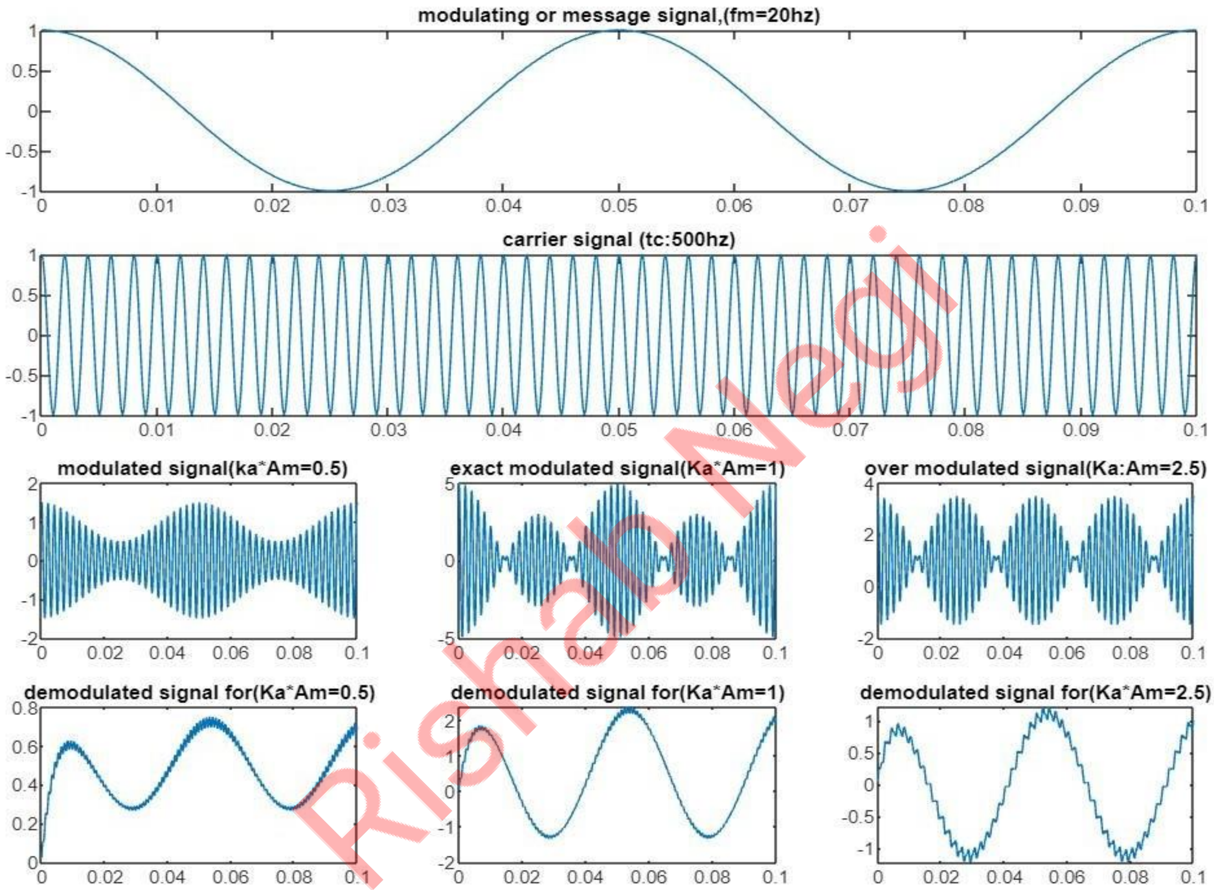
AIM: To perform Amplitude modulation and Demodulation of a signal.

CODE:

```
fs=8000; fm=20;
fc=500; Am=1; Ac=1;
t=(0:0.1*fs)/fs;
m=Am*cos(2*pi*fm*t);
c=Ac*cos(2*pi*fc*t);
Ka=0.5; U=Ka*Am;
s1=Ac*(1+U*cos(2*pi*fm*t)).*(cos(2*pi*fc*t));
subplot(4,3,1:3);
plot(t,m);
title('modulating or message signal,(fm=20hz)');
subplot(4,3,4:6);
plot(t,c);
title('carrier signal (tc:500hz)');
subplot(4,3,7);
plot(t,s1);
title(' modulated signal(ka*Am=0.5)');
Am=2; Ka=0.5; u=Ka*Am;
s2=Ac*(1+4*cos(2*pi*fm*t)).*(cos(2*pi*fc*t));
subplot(4,3,8);
plot(t,s2);
title('exact modulated signal(Ka*Am=1)');
Am=5;
Ka=0.5; U=Ka*Am;
s3=Ac*(1+U*cos(2*pi*fm*t).*cos(2*pi*fc*t));
subplot(4,3,9);
plot(t,s3);
title('over modulated signal(Ka:Am=2.5)');
r1=s1.*c;
[b,a]=butter(1,0.01);
mr1=filter(b,a,r1);
subplot(4,3,10);
plot(t,mr1);
title('demodulated signal for(Ka*Am=0.5)');
r2=s2.*c;
[b,a]=butter(1,0.01);
mr2=filter(b,a,r2);
subplot(4,3,11);
```

```
plot(t,mr2);
title('demodulated signal for(Ka*Am=1)');
r3=s3.*c;
[b,a]=butter(1,0.01);
mr3=filter(b,a,r3);
subplot(4,3,12);
plot(t,mr3);
title('demodulated signal for(Ka*Am=2.5)');
```

OUTPUT:

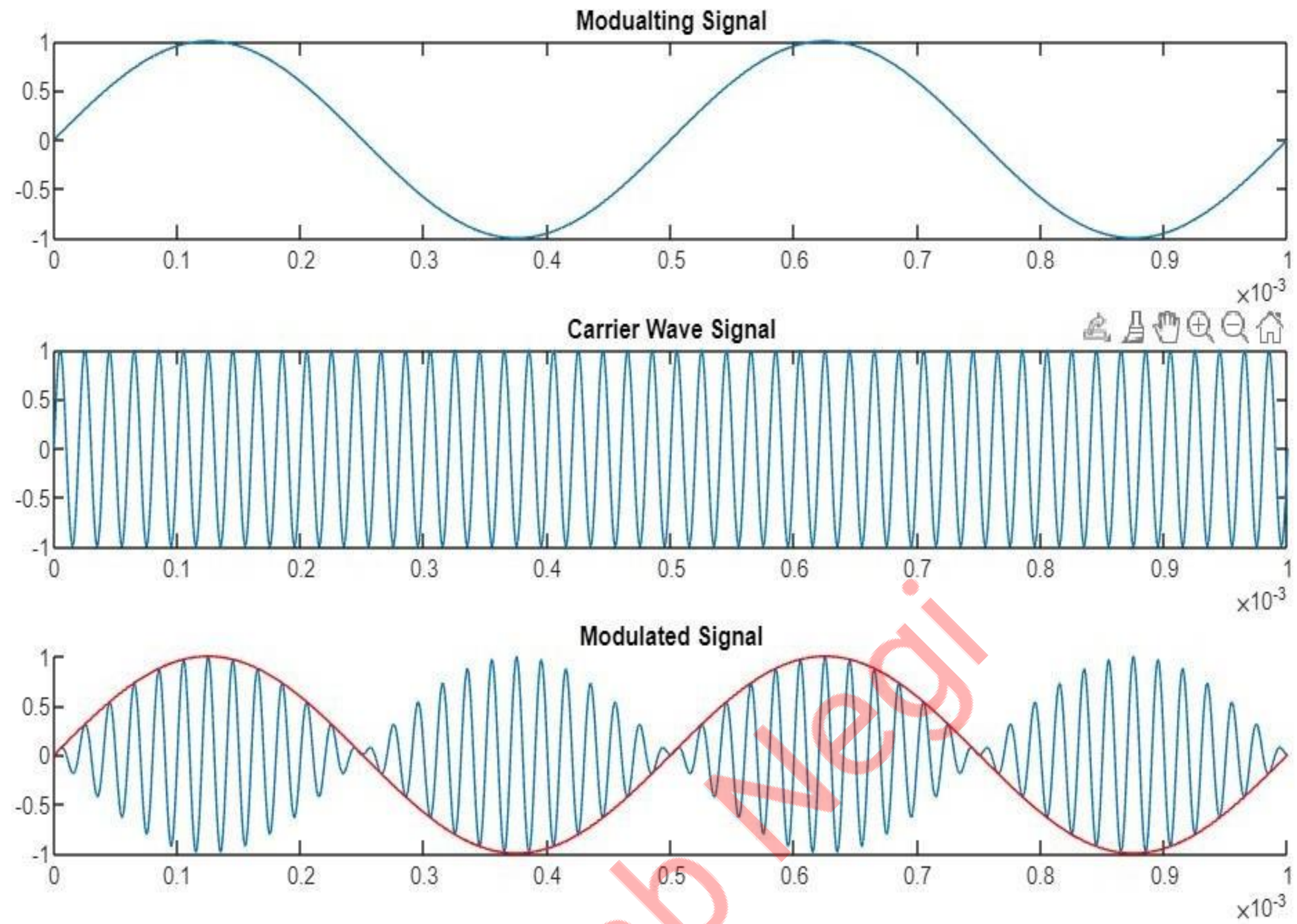


AIM: To perform experiment on DSB-SC MODULATOR AND DETECTOR of a signal.

CODE:

```
t=0:0.000001:0.001;
vm=1; vc=1; fm=2000;
fc=50000;
m_t=vm*sin(2*pi*fm*t);
subplot(4,1,1);
plot(t,m_t);
title("Modualting Signal");
c_t=vc*sin(2*pi*fc*t);
subplot(4,1,2);
plot(t,c_t);
title("Carrier Wave Signal");
s_t=m_t.*c_t;
subplot(4,1,3);
hold on;
plot(t,s_t);
plot(t,m_t,'r');
title("Modulated Signal");
hold off;
r=s_t*c_t;
[b,a]=butter(1,0.01);
mr=filter(b,a,r);
subplot(4,1,4);
plot(t,mr);
```

OUTPUT:



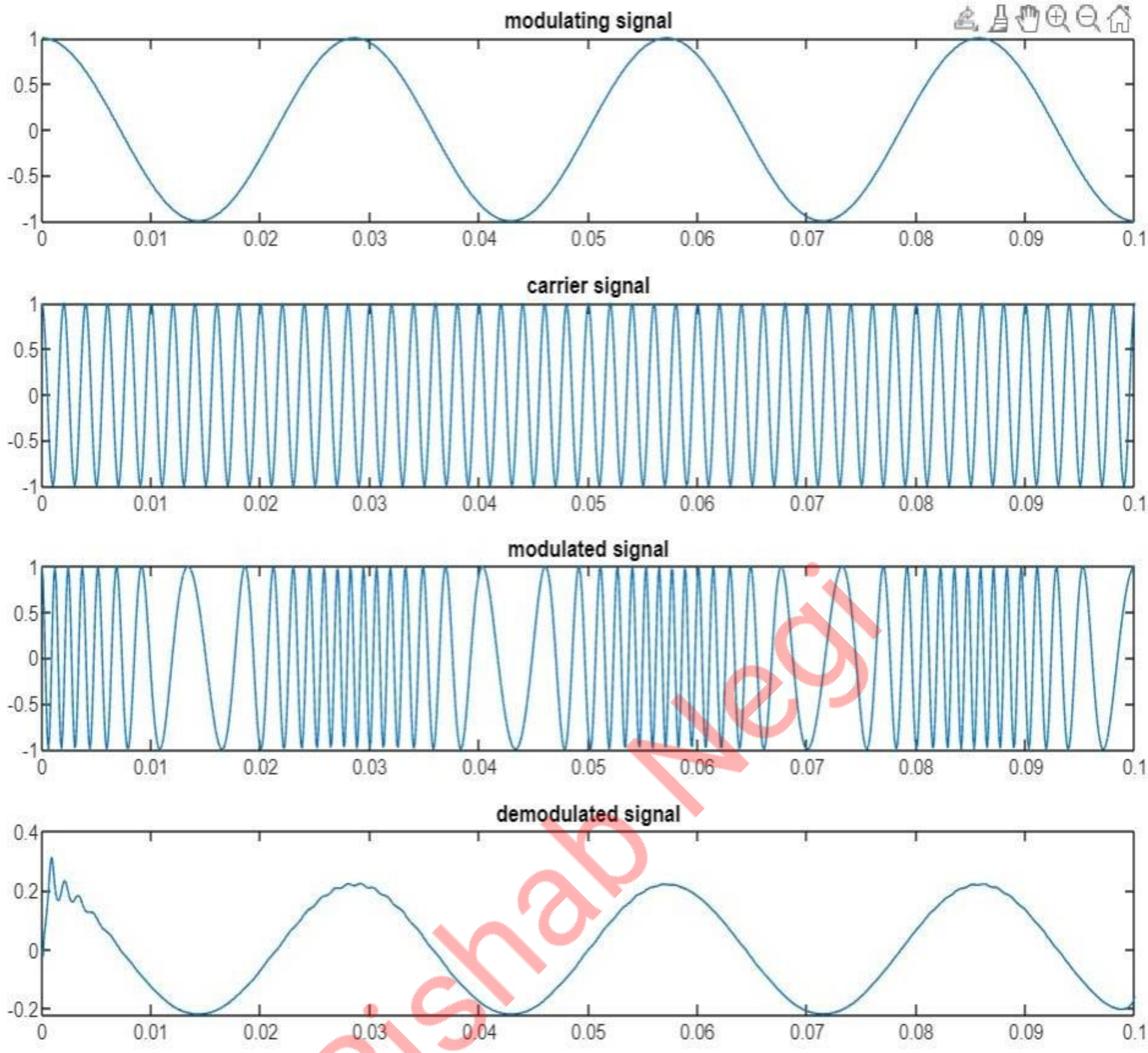
Modualting Signal
Carrier Wave Signal
Modulated Signal
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
×10^-3
Rishab Negi

AIM: To perform Amplitude modulation and Demodulation of a signal.

CODE:

```
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(4,1,1);
plot(t,m_t);
title('modulating signal');
c_t=ac*cos(wc*t);
subplot(4,1,2);
plot(t,c_t);
title('carrier signal');
s_t=c_t+b*sin(wm*t);
subplot(4,1,3);
plot(t,st);
title('modulated signal');
d=demod(st,fc,fs,'fm');
subplot(4,1,4);
plot(t,d);
title('demodulated signal');
```

OUTPUT:



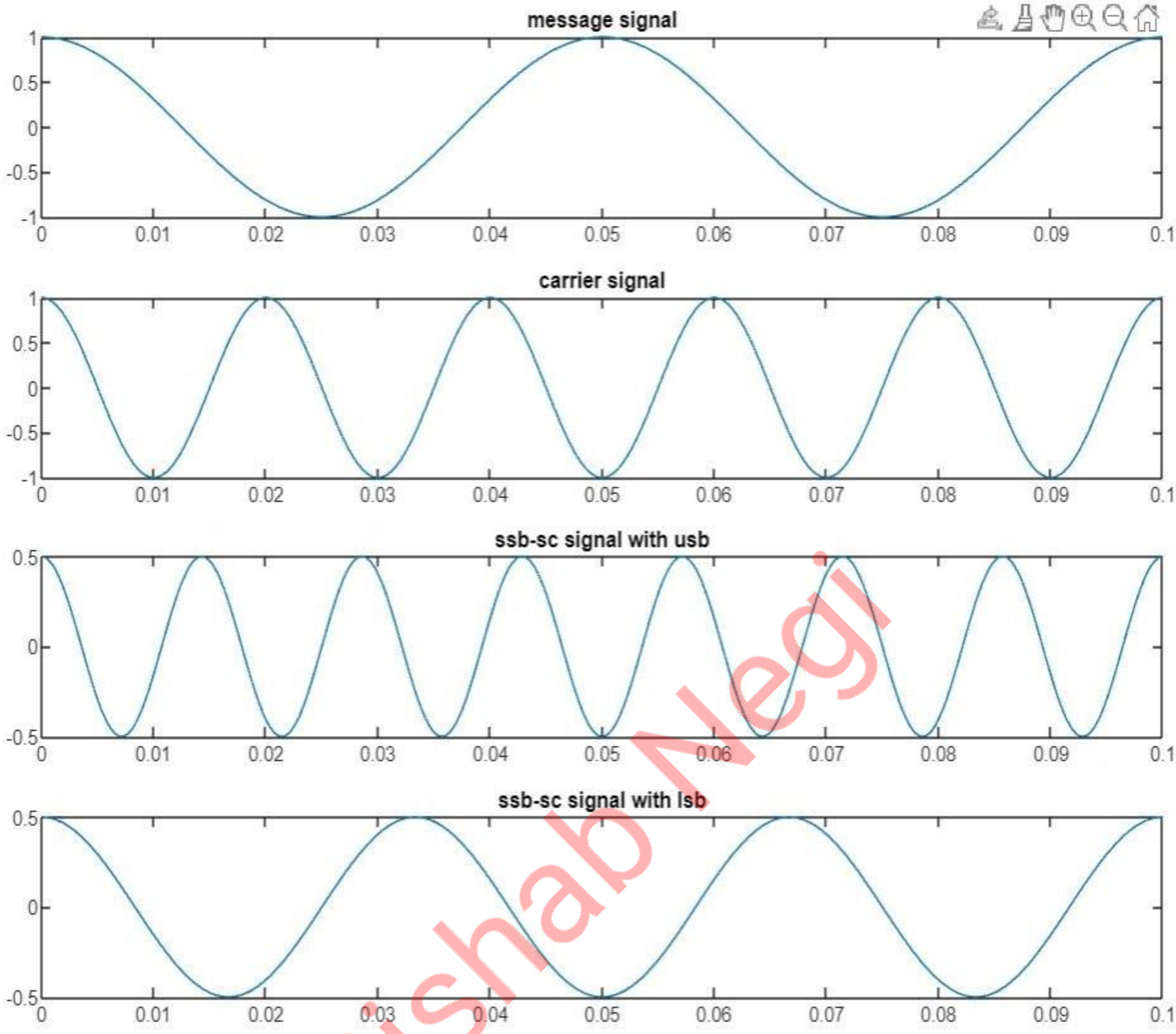
modulating signal
carrier signal
modulated signal
demodulated signal

AIM: To perform experiment on SSB MODULATOR AND DETECTOR of a signal.

CODE:

```
fs=8000; fm=20; fc=50;
am=1; ac=1;
t=(0:0.1*fs)/fs;
subplot(4,1,1);
m1=am*cos(2*pi*fm*t);
plot(t,m1);
title('message signal');
m2=am*sin(2*pi*fm*t);
subplot(4,1,2);
c1=ac*cos(2*pi*fc*t);
plot(t,c1);
title('carrier signal');
c2=ac*sin(2*pi*fc*t);
subplot(4,1,3);
Susb=(0.5*m1.*c1)-(0.5*m2.*c2);
plot(t,Susb);
title('ssb-sc signal with usb');
Slsb=0.5*m1.*c1+0.5*m2.*c2;
subplot(4,1,4);
plot(t,Slsb);
title('ssb-sc signal with lsb');
```

OUTPUT:



message signal
carrier signal
ssb-sc signal with usb
ssb-sc signal with lsb
1
0.5
0
-0.5
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Rishab Negi

AIM: To perform Pre-Emphasis and De-Emphasis of a given signal.

CODE:

```
num_samples=2^13;
fs=5000; ts=1/fs;
fm1=20; fm2=30; fc=200;
t=(0:num_samples-1)*ts;
f=(-num_samples/2:num_samples/2-1)*fs/num_samples;
mt=sin(2*pi*fm1*t);
mf=fftshift(abs(fft(mt)));
f_cutoff_pe=15;
wn_pe=f_cutoff_pe/(fs/2);
[b_pe,a_pe]=butter(1,wn_pe);
[h_pe,w]=freqz(a_pe,b_pe);
a_de=b_pe; b_de=a_pe;
[h_de,w]=freqz(a_de,b_de);
mt_pe=filter(a_pe,b_pe,mt);
mf_pe=fftshift(abs(fft(mt_pe)));
figure(1) grid on;
subplot(2,1,1);
plot(t,mt);
title('modulatting signal(time domain)');
axis([0 0.6 min(mt)-1 max(mt)+1]);
grid on;
subplot(2,1,2);
plot(f,mf);
title('modulating signal(frequency domain)');
axis([-50 50 0 max(mf)+100])
figure(2) subplot(2,1,1);
semilogx(w*pi*(fs/2),abs(h_pe),'m','linewidth',2)
axis([0 fs/2 0 50]) grid on;
title('pre-emphasis filter magnitude response');
subplot(2,1,2);
semilogx(w*pi*(fs/2),abs(h_de),'m','linewidth',2)
axis([0 fs/2 0 1]) grid on;
title('de-emphasis filter magnitude response');
```

OUTPUT:

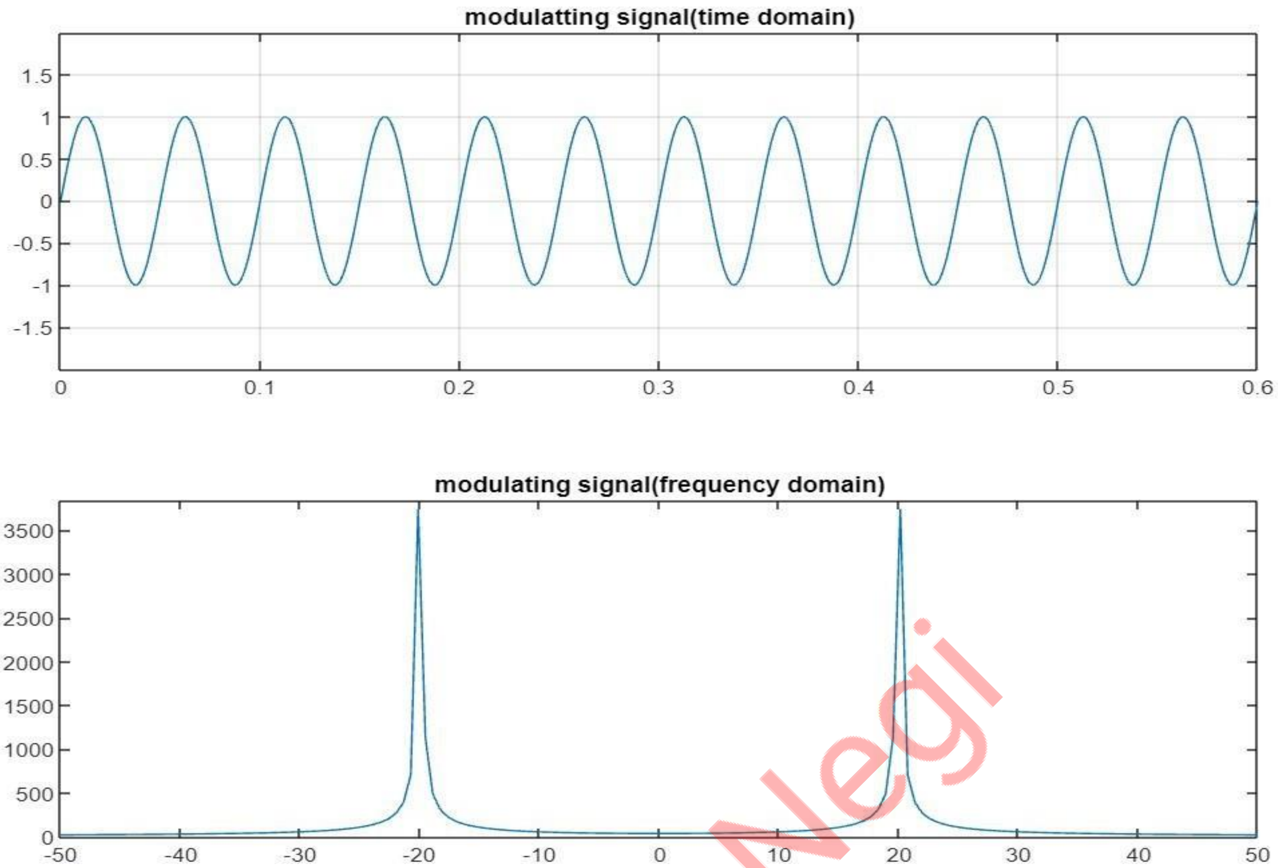


Fig 1

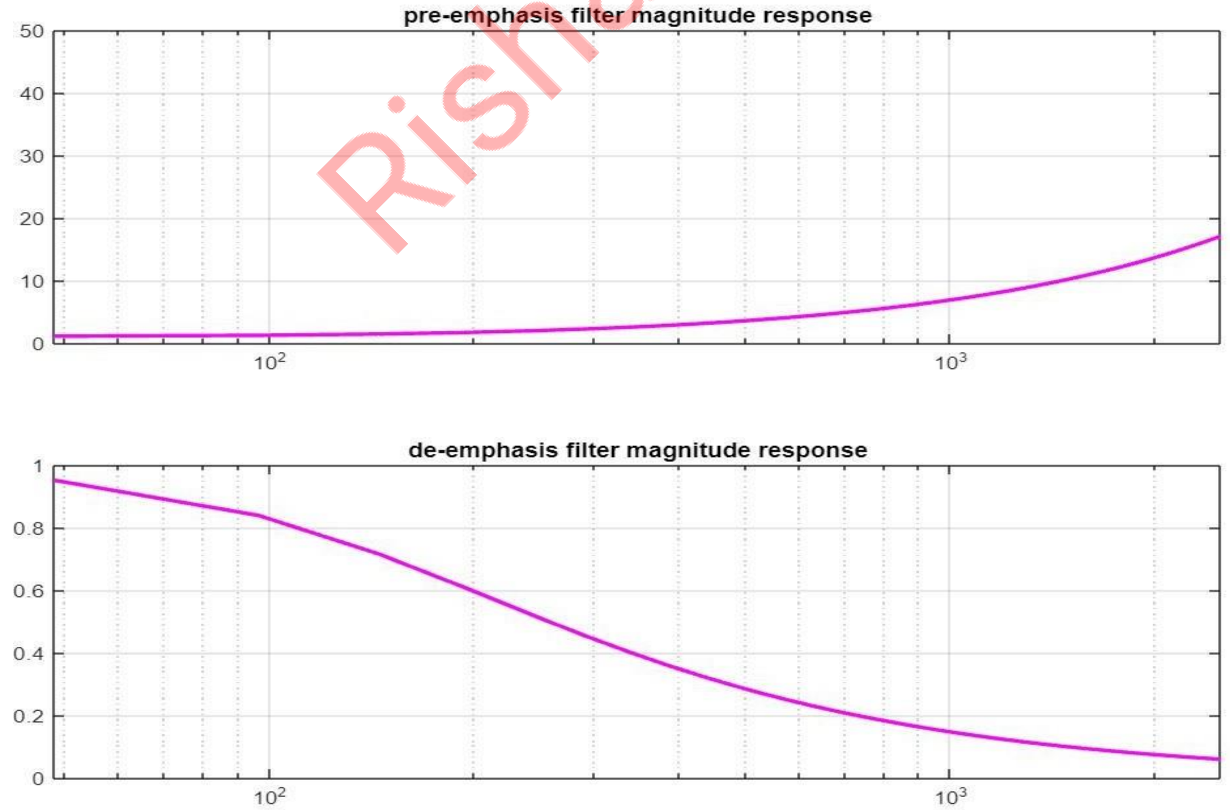


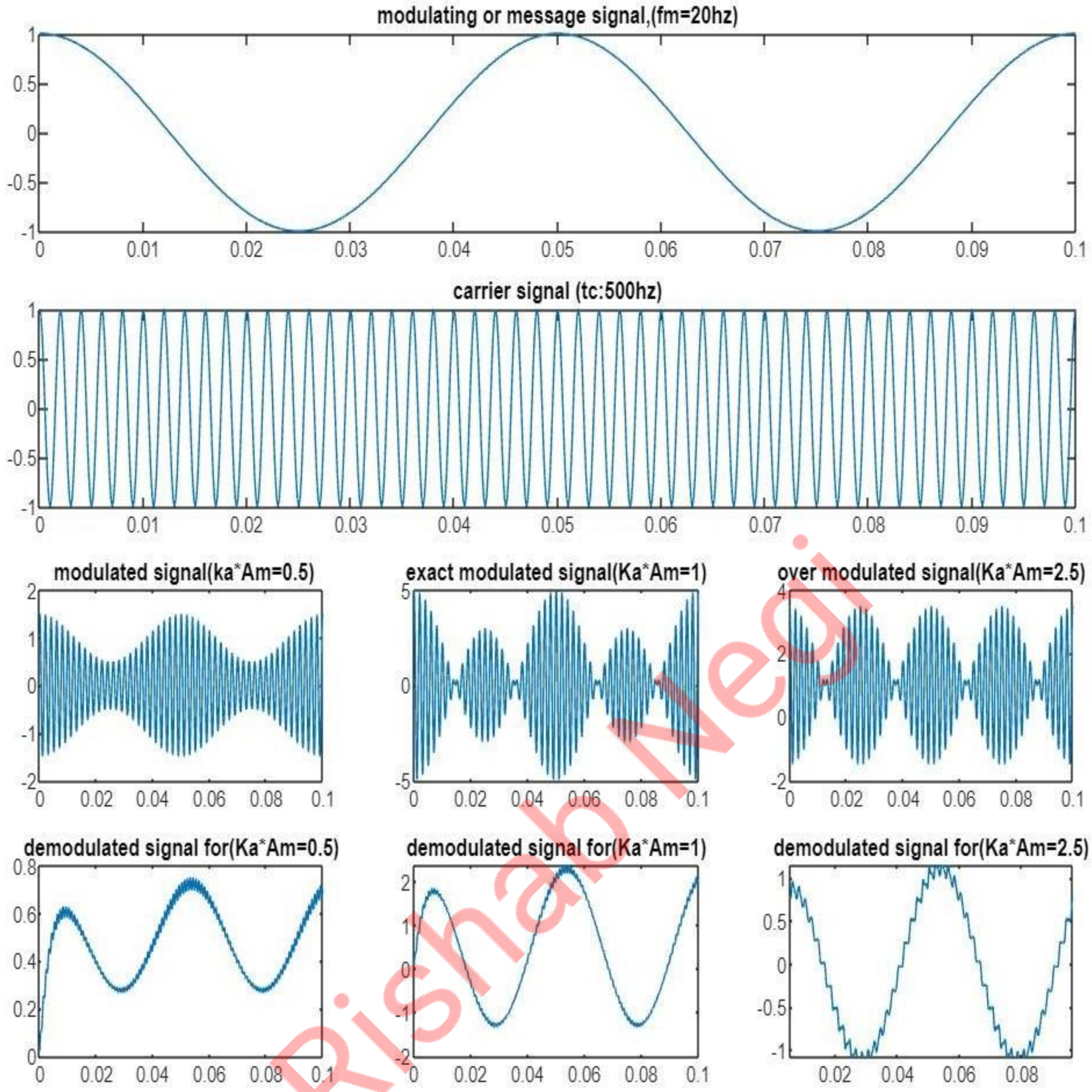
Fig 2

AIM: To perform Pulse Amplitude modulation and Demodulation of a signal.

CODE:

```
fs=8000; fm=20;
fc=500; Am=1; Ac=1;
t=(0:0.1*fs)/fs;
m=Am*cos(2*pi*fm*t);
c=Ac*cos(2*pi*fc*t);
Ka=0.5; U=Ka*Am;
s1=Ac*(1+U*cos(2*pi*fm*t)).*(cos(2*pi*fc*t));
subplot(4,3,1:3); plot(t,m);
title('modulating or message signal,(fm=20hz)');
subplot(4,3,4:6); plot(t,c);
title('carrier signal (tc:500hz)');
subplot(4,3,7); plot(t,s1);
title(' modulated signal(ka*Am=0.5)');
Am=2;
Ka=0.5; U=Ka*Am;
s2=Ac*(1+4*cos(2*pi*fm*t)).*(cos(2*pi*fc*t));
subplot(4,3,8); plot(t,s2);
title('exact modulated signal(Ka*Am=1)');
Am=5;
Ka=0.5; U=Ka*Am;
s3=Ac*(1+U*cos(2*pi*fm*t).*cos(2*pi*fc*t));
subplot(4,3,9); plot(t,s3);
title('over modulated signal(Ka*Am=2.5)');
r1=s1.*c;
[b,a]=butter(1,0.01);
mr1=filter(b,a,r1);
subplot(4,3,10); plot(t,mr1);
title('demodulated signal for(Ka*Am=0.5)');
r2=s2.*c;
[b,a]=butter(1,0.01);
mr2=filter(b,a,r2);
subplot(4,3,11); plot(t,mr2);
title('demodulated signal for(Ka*Am=1)');
r3=s3.*c;
[b,a]=butter(1,0.01);
mr3=filter(b,a,r3);
subplot(4,3,12); plot(t,mr3);
title('demodulated signal for(Ka*Am=2.5)');
```

OUTPUT:



Created by: Rishab Negi