# COMSYS LAB4

## Name: Sidharth S Nair | ID: 2019A3PS0178P

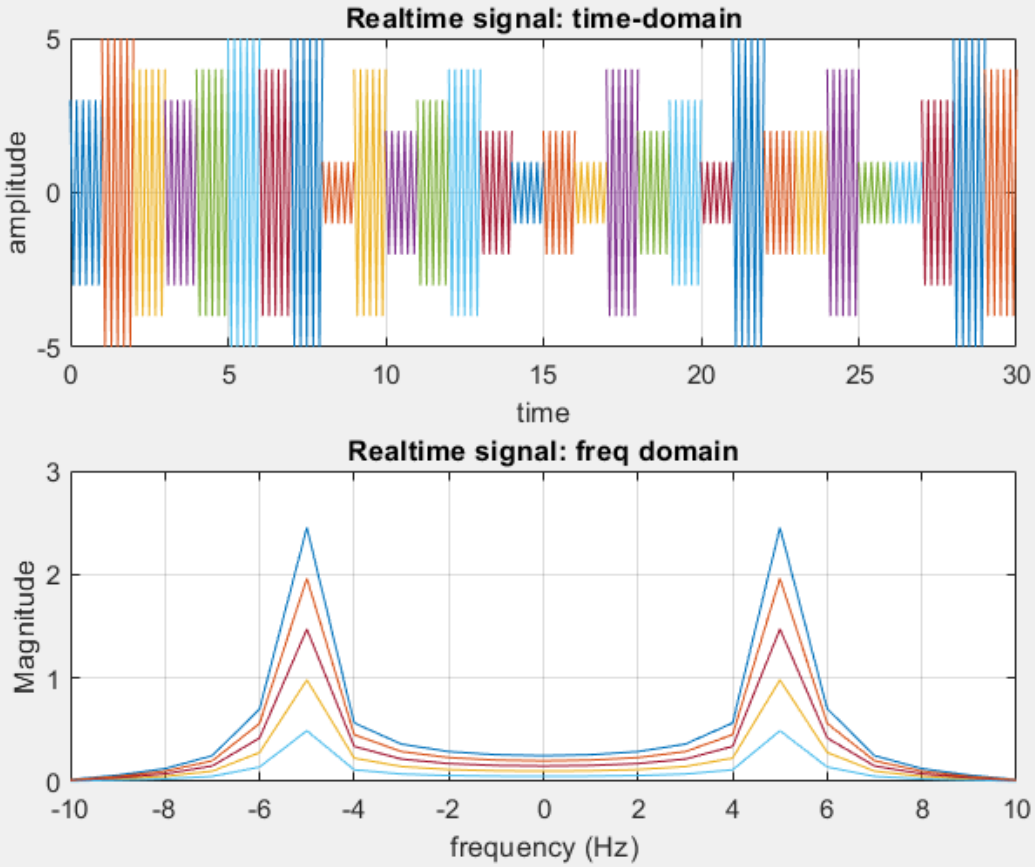
## Task1: Realtime signals

### Code

```
clear all
close all
duration_signal=29;
A=1;
for T = 0:duration_signal  %%%% Duration 30 seconds with interval of 1 sec.
    if T==0
    display('Transmission Started')
     display (T)
    elseif (T==duration_signal)
        display('Transmission ends: see the final result')
     display (T)
    else
            display('Transmission in progress: please wait')
     display (T)
    end
    freq=5;

   fs=4*freq;
    ts=1/fs;
     t=0:ts:1;

     U=randi(5);
    m_t=U*cos(2*pi*freq*t);
  N=length(m_t);
  m_f= fft(m_t,N)/fs;
  freqaxis=linspace(-fs/2, fs/2, N);
    figure(1)

    hold all  %%% keeps the previous plots and everytime changes the color
    subplot(2,1,1), plot(t+T,m_t);
    xlabel('time')
    ylabel('amplitude')
    grid on
    axis([0  inf -5 5])   %%% first two are limits for x-axis, the other two are limit
s for y-axis: observe why 0 inf , and -5 5 are used here.
    hold on     %%% keeps the previous plots
    subplot(2,1,2), plot(freqaxis,fftshift(abs(m_f)))
      xlabel('frequency (Hz)')
    ylabel('Magnitude')
    grid on
    axis([-inf inf 0 3])  %%% first two are limits for x-axis, the other two are limit
s for y-axis: observe why -inf inf , and 0 3 are used here.
```

```
    pause(2)  %%%%% pauses for 2 seconds and then go for next loop increment.
end
```

## Plot




# Task2: Real-time signals with choice and memory

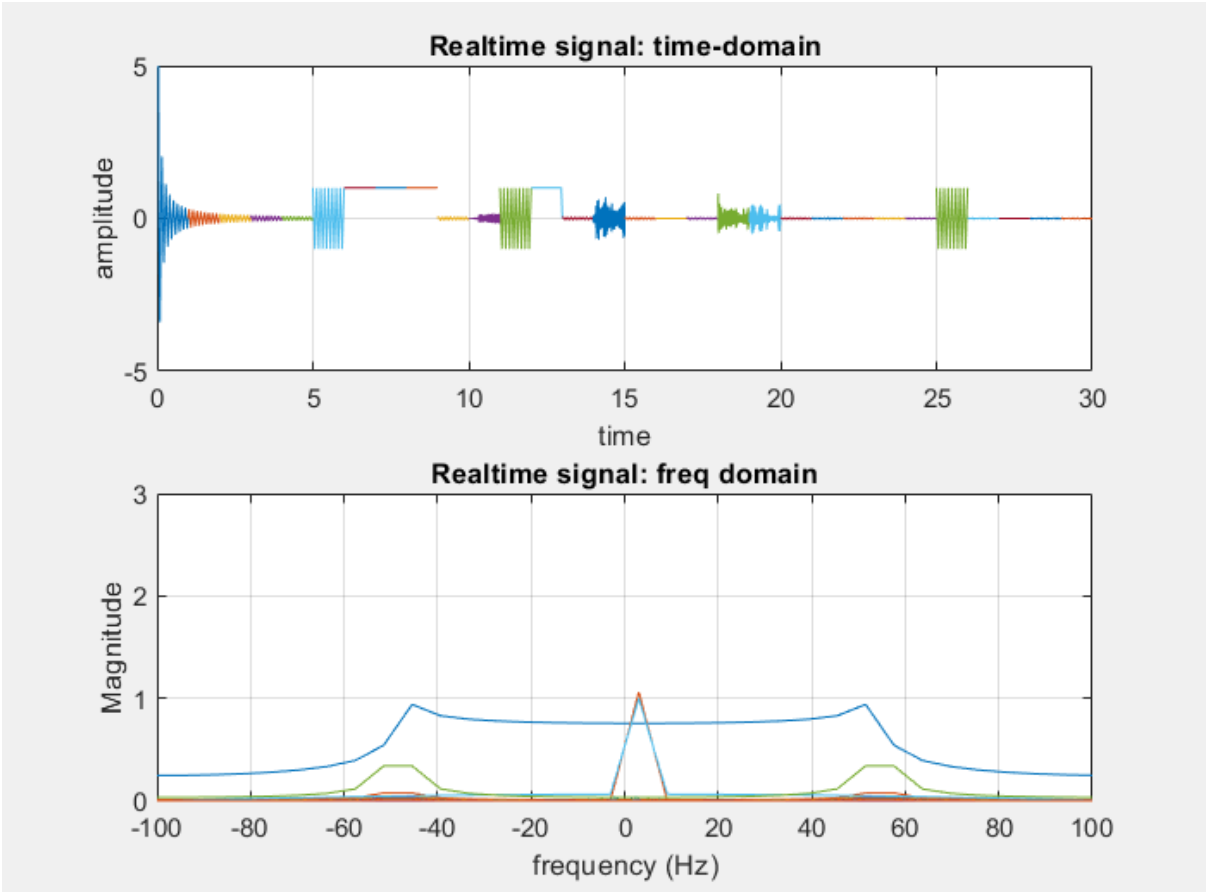
## code

```
clear all;
close all;
clc;
duration_signal=29;
A=1;
t_start=ones(1,4);
%tstop=ones(1,4);
freq=8;
N1=8;
fs=4*freq;
ts=1/fs;
t1=0:ts:30;
rect=@(T,t)(abs(t)<(T/2));
m1=cos(2*pi*freq*t1);
```

```
m2=2*N1*sinc(2*N1*t1);
m3=rect(N1,t1);
[y,Fs]=audioread('file_example_WAV_1MG.wav');
for T = 0:duration_signal  %%%% Duration 30 seconds with interval of 1 sec.
    if T==0
    display('Transmission Started')
     display (T)
    elseif (T==duration_signal)
        display('Transmission ends: see the final result')
     display (T)
    else
            display('Transmission in progress: please wait')
     display (T)
    end
    freq=8;

    fs=4*freq;
    ts=1/fs;
    t=0:ts:1;
    U=randi(4);
    %m_t=U*cos(2*pi*freq*t);
    if U==1
        m_t=m1(t_start(1):t_start(1)+fs+1);
        t_start(1)=t_start(1)+fs;
    elseif(U==2)
        m_t=m2(t_start(2):t_start(2)+fs+1);
        t_start(2)=t_start(2)+fs;
    elseif (U==3)
        m_t=m3(t_start(3):t_start(3)+fs+1);
        t_start(3)=t_start(3)+fs;
    else
        m_t=y(t_start(4):t_start(4)+Fs+1);
        t_start(4)=t_start(4)+Fs;
        fs=Fs;
    end
    N=length(m_t);
    m_f= fft(m_t,N)/fs;
    freqaxis=linspace(-100,100, N);
    figure(1)

    hold all  %%% keeps the previous plots and everytime changes the color
    subplot(2,1,1),
    if U==4
        t2=0+T:1/fs:1+T;
        plot(t2,m_t(1:size(t2,2)));
    else
        plot(t+T,m_t(1:size(t,2)));
    end

    xlabel('time')
    ylabel('amplitude')
    title('Realtime signal: time-domain');
    grid on
    axis([0  inf -5 5])   %%% first two are limits for x-axis, the other two are limit
s for y-axis: observe why 0 inf , and -5 5 are used here.
    hold on     %%% keeps the previous plots
    subplot(2,1,2), plot(freqaxis,fftshift(abs(m_f)))
    xlabel('frequency (Hz)')
```

```
    ylabel('Magnitude')
     title('Realtime signal: freq domain');
    grid on
    axis([-inf inf 0 3])  %%% first two are limits for x-axis, the other two are limit
s for y-axis: observe why -inf inf , and 0 3 are used here.
    pause(0.5)  %%%%% pauses for 2 seconds and then go for next loop increment.
end
```

### output




### Observations

- The signals were cached before transmission and randomly chosen and subsequent packets were sent each time by storing where the last packets transmission ended
- We can see this by following the rectangle function or the music file output which are different each time

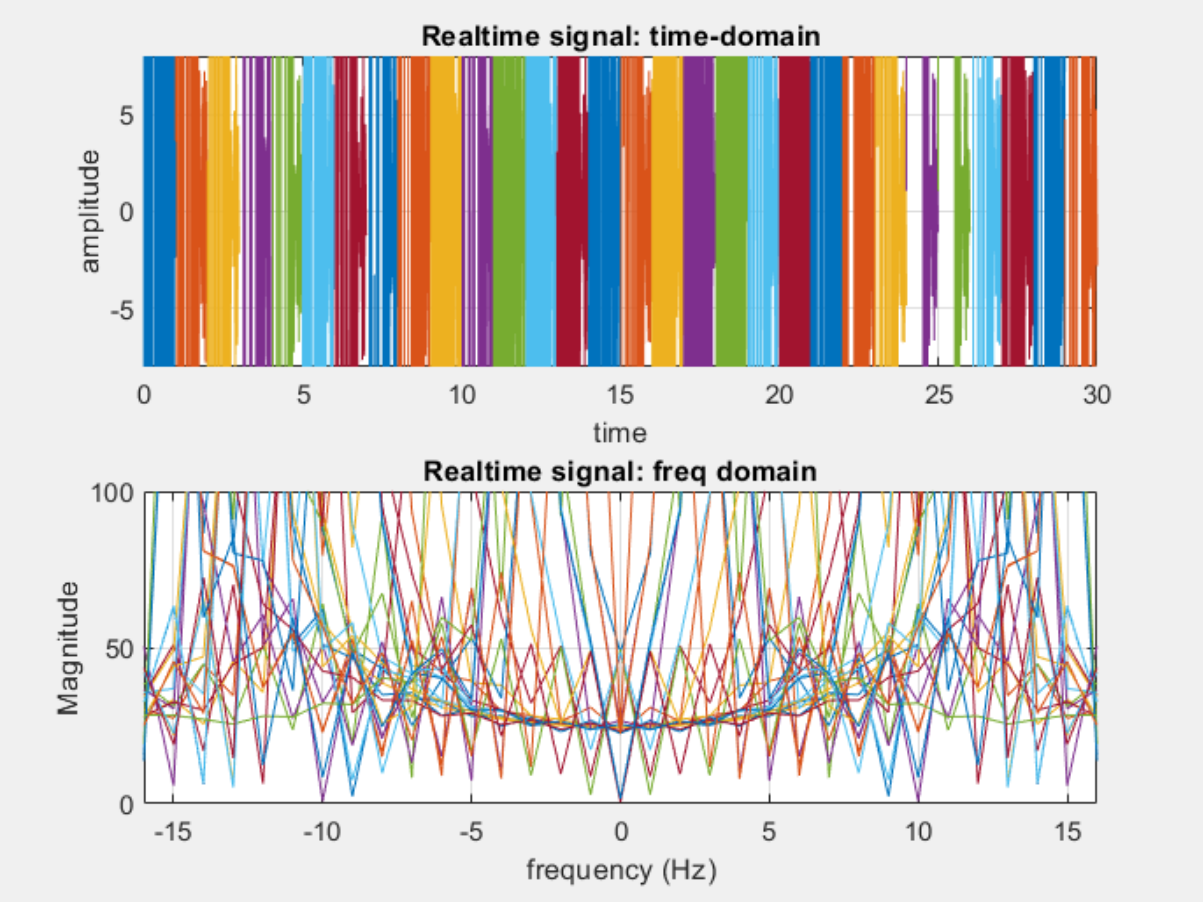
## Task-3: Cosine Realtime

### code

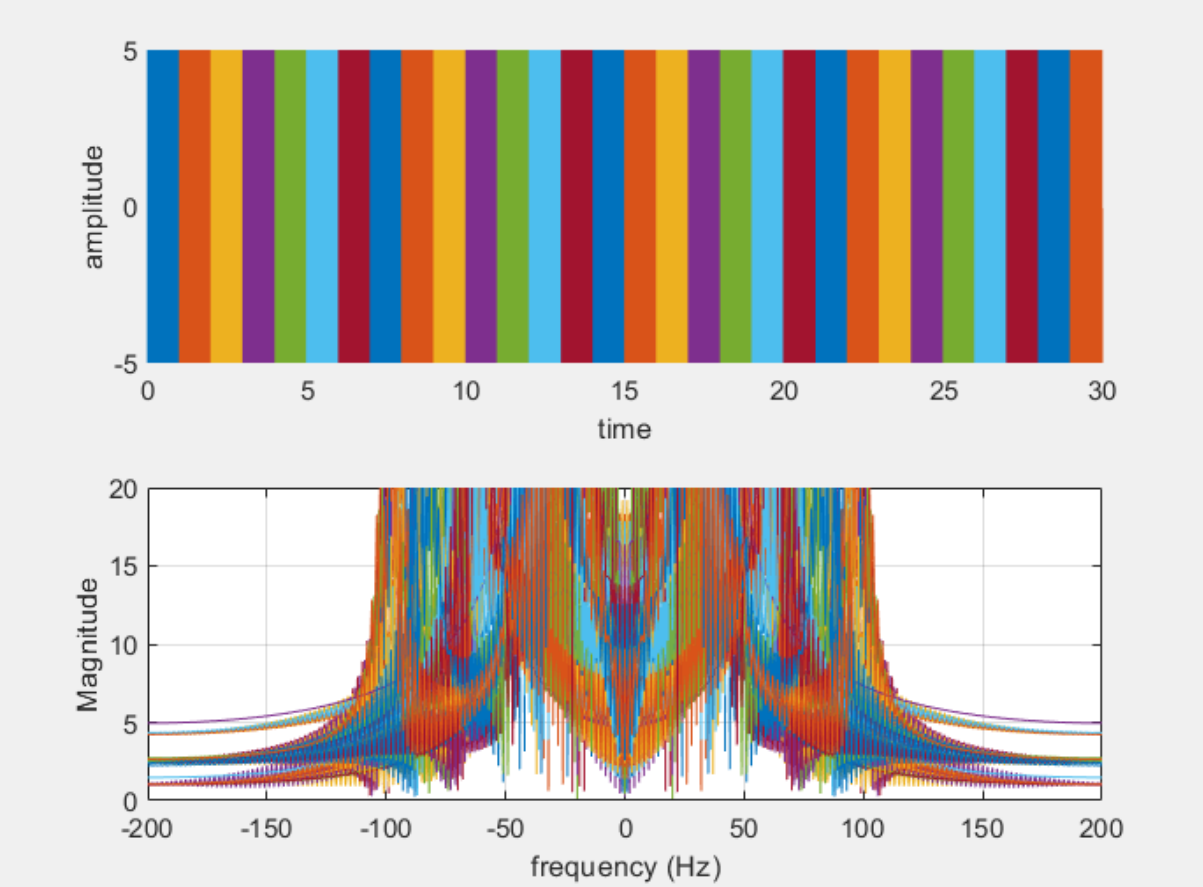
```
clear all;
close all;
clc;
duration_signal=29;
A=1;
for T = 0:duration_signal  %%%% Duration 30 seconds with interval of 1 sec.
    if T==0
    display('Transmission Started')
     display (T)
    elseif (T==duration_signal)
        display('Transmission ends: see the final result')
     display (T)
    else
            display('Transmission in progress: please wait')
     display (T)
    end
    freq=8;
    N=8;
    fs=50*freq;
    ts=1/fs;
     t=0:ts:1;
    f=10+mod(randi(100,1,2),90);
    U=randi(5);
    bw=50 ;%bandwidth=50Hz
    m_t= N*cos(2*pi*f(1)*t)+N*cos(2*pi*f(2)*t);
    channel=2*bw*sinc(2*bw*t);
    y=conv(m_t,channel,'same');
    N1=length(y);
    m_f= fft(y,N1)/fs;
    freqaxis=linspace(-fs/2,fs/2, N1);
    figure(1)

    hold all  %%% keeps the previous plots and everytime changes the color
    subplot(2,1,1), plot(t+T,y);
    xlabel('time')
    ylabel('amplitude')
    grid on
    axis([0  inf -5 5])   %%% first two are limits for x-axis, the other two are limit
s for y-axis: observe why 0 inf , and -5 5 are used here.
    hold on     %%% keeps the previous plots
    subplot(2,1,2);
    plot(freqaxis,fftshift(abs(m_f)));
      xlabel('frequency (Hz)')
    ylabel('Magnitude')

    grid on
    axis([-inf inf 0 20])  %%% first two are limits for x-axis, the other two are limi
ts for y-axis: observe why -inf inf , and 0 3 are used here.
    pause(0.5)  %%%%% pauses for 2 seconds and then go for next loop increment.
end
```

## Plot

I have increased the limits of y-axis for better visibility. The limit of x axis to be 8 since for me the value of N=8




On increasing Sampling frequency we get the following more detailed plot




# Task 4: Hilbert transform

## code

```
clc; clear all; close all;
N = 8;
fs = 10*N;  % 10 times the maximum frequency component in Hertz.
for T = 0:1:29
    t = 0:1/fs:1-1/fs;
    t = t+T;  %to get continuation each time
    m = 2*N*sinc(2*N*t);
    x1 = hilbert(m);

    mf = fft(x)/fs;
    mf_abs_sorted = fftshift(abs(mf));
    freq_axis = linspace(-fs/2, fs/2, length(mf));

    figure(1);
    subplot(2,1,1);
    plot(t,real(x1),'g',t,imag(x1),'red');hold on;
    title('Time Domain');
    xlabel('Time in seconds');
    ylabel('Amplitude');
    legend('Realpart of x(t))','Imaginary part of x(t)');

    subplot(2,1,2);
```

```
    plot(freq_axis,mf_abs_sorted);hold on;
    title('Frequency Domain');
    ylabel('Frequency in Hertz')
    pause(0.5);
end
```

## output