International Institute of Information Technology
Bangalore

Principles of Communication Systems Lab

EC 303P

Lab 2: Review of Signals and Systems

Mohd. Rizwan Shaikh
(IMT2019513)

August 23, 2021

Lab 2: Review of Signals and Systems

Mohd. Rizwan Shaikh

August 23, 2021

Abstract

In this lab report, we explore the basics of MATLAB and various signal properties.

1 Introduction

In this lab report, we explore the basics of MATLAB and various signal properties.

1.1 Question 1

Consider a square wave with fundamental frequency 50 Hz with the maximum and minimum amplitude to be +1 and -1 respectively. Generate and plot (both continuous time and discrete time) 5 complete cycles of it with the following duty cycles.

(a) Duty Cycle 50%.

(b) Duty Cycle 25%.

(c) Duty Cycle 75%.

1.2 Question 2

Consider the signals $x_1[n] = 1$, for $n_{11} \leq n \leq n_{12}$, and $x_2[n] = 1$, for $n_{21} \leq n \leq n_{22}$ and perform the following operations:

(a) Find and plot (both continuous time and discrete time) the convolution between $x_1[n]$ and $x_2[n]$ for $n_{11} = n_{21} = -2$ and $n_{12} = n_{22} = 2$. Do not use the inbuilt function 'conv'!

(b) Find and plot (both continuous time and discrete time) the convolution between $x_1[n]$ and $x_2[n]$ for $n_{11} = -2$, $n_{12} = 2$, $n_{21} = 0$ and $n_{22} = 4$. Do not use the inbuilt function 'conv'!

(c) Find and plot (both continuous time and discrete time) the convolution between $x_1[n]$ and $x_2[n]$ for $n_{11} = -5$, $n_{12} = 5$, $n_{21} = -1$ and $_{n22} = 6$. Do not use the inbuilt function 'conv'!

(d) Verify and plot the above results using the inbuilt function 'conv'.

1.3 Question 3

Here we consider two signals, $m_1(t) = A_1 cos(2\pi f_1 t)$ and $m_2(t) = A_1 cos(2\pi f_1 t) + A_2 cos(2\pi f_2 t)$ with $A_1 = 2$, $A_2 = 6$, $f_1 = 50$ and $f_2 = 100$. We do the following:

(a) Plot $m_1(t)$ and $m_2(t)$.

(b) Plot the frequency spectrum $m_1(t)$ and $m_2(t)$, and identify the tones.

(c) Plot the phase spectrum of $m_1(t)$ and $m_2(t)$.

2 Method

2.1 Question 1

I implemented a function whose output is the signal vector and input is duty cycle, time period of signal and time vector. The function is as follows:

```
%<===== Function to find signal vector based on duty cycle =====>
function [x] = findCycle(x, Duty, Ts, t)
   T_ = Ts*Duty/100;     %T_ is the time at which amplitude goes from 1 to -1
   TP = Ts;              %TP is the time at which amplitude goes from -1 to 1
   mul = -1;
   for i = 1:length(t)  %We iterate through the signal vector and assign it ...
       the value 1 or -1 based on time.
       x(i) = x(i) * mul;
       if t(i) ≥ TP
           TP = TP + Ts;
           mul = mul * -1;
       end
       if t(i) ≥ T_
           T_ = T_ + Ts;
           mul = mul * -1;
       end
   end
end
```

Before plotting, we call the function by varying duty cycle and get the desired signal vector based on the duty cycle. We use `stem(x, y)` to plot discrete time signal and `plot(x, y)` to plot continuous time signal.
By using `subplot`, we plot all the sub parts in the same plot.

2.2 Question 2

In order to solve this question I implemented a function that takes n_{11}, n_{12}, n_{21} and n_{22} as the input and gives the discrete time convolution vector as output. The function is as shown below:

```
function [y, ny] = FindConvolution(n11, n12, n21, n22)
    x1 = ones(1, n12 - n11 + 1);
    x2 = ones(1, n22 - n21 + 1);
    ny = (n11+n21):1:(n12+n22);
    y = zeros(1, (n12+n22)-(n11+n21)+1);

    n1 = length(x1);
```

```
    n2 = length(x2);
    for n = 1:(n1+n2-1)
        for k = 1:n1
            if(n-k ≥ 0 && n-k < n2)
                y(n) = y(n) + x1(k)*x2(n-k+1);
            end
        end
    end
end
```

Using the above function, the discrete time convolution is plotted using `stem(x, y)` command. To plot continuous time convolution, we use the same function. For plotting, we use `plot(x, y)` command. To verify the results, we use the inbuilt `conv(x1, x2)` command to find the convolution. We plot the discrete time convolution using `stem(x, y)` and continuous time convolution using `plot(x, y)`.
By using `subplot`, we plot all the sub parts in the same plot.

2.3 Question 3

(a) To plot $m_1(t)$, `plot(t1, m1)` command is used. Here `t1` is time vector and `m1` is the signal vector for $m_1(t)$ signal. Using same command, $m_2(t)$ signal is plotted.

(b) To plot the Fourier spectrum of the signals, we first use `fft` to find the Fourier transform of the signals. After that, we use `fftshift` to perform zero-centered circular shift on the Fourier transform. We plot absolute value of signal to get the frequency spectrum. Before plotting, we divide the resultant vector by length of signal vector to get the plot that we are used to.

(c) To plot phase spectrum, we just use `angle(Y)` where `Y` is `fftshift(fft(X))` (Here `X` is the signal vector).

By using `subplot`, we plot all the sub parts in the same plot.

3 Results and Analysis

3.1 Question 1

The MATLAB code for the question is shown below:

```
clear all;
close all;
clc;

fs = 50;
Ts = 1/fs;

tc = 0:1/10000:5*Ts;
xc = -1*(tc ≥ 0 & tc ≤ 5*Ts);      %For continuous time signal
td = 0:1/2000:5*Ts;
xd = -1*(td ≥ 0 & td ≤ 5*Ts);      %For discrete time signal

%≤===== 1(a) ======>
x1 = findCycle(xc, 50, Ts, tc);
```

```
    subplot(3,2,1);
    plot(tc, x1);              ylim([-1.5 1.5]);
    xlabel('---> time(s)');    ylabel('---> x(t)');
    title('1(a1) Continuous Time, Duty Cycle = 50%');
    grid on;

    x1 = findCycle(xd, 50, Ts, td);
    subplot(3,2,2);
    stem(td, x1);               ylim([-1.5 1.5]);
    xlabel('---> time(s)');     ylabel('---> x(t)');
    title('1(a2) Discrete Time, Duty Cycle = 50%');
    grid on;

    %≤===== 1(b) ======>
    x1 = findCycle(xc, 25, Ts, tc);
    subplot(3,2,3);
    plot(tc, x1);              ylim([-1.5 1.5]);
    xlabel('---> time(s)');    ylabel('---> x(t)');
    title('1(b1) Continuous Time, Duty Cycle = 25%');
    grid on;

    x1 = findCycle(xd, 25, Ts, td);
    subplot(3,2,4);
    stem(td, x1);              ylim([-1.5 1.5]);
    xlabel('---> time(s)');    ylabel('---> x(t)');
    title('1(b2) Discrete Time, Duty Cycle = 25%');
    grid on;


    %≤===== 1(c) ======>
    x1 = findCycle(xc, 75, Ts, tc);
    subplot(3,2,5);
    plot(tc, x1);              ylim([-1.5 1.5]);
    xlabel('---> time(s)');    ylabel('---> x(t)');
    title('1(c1) Continuous Time, Duty Cycle = 75%');
    grid on;

    x1 = findCycle(xd, 75, Ts, td);
    subplot(3,2,6);
    stem(td, x1);               ylim([-1.5 1.5]);
    xlabel('---> time(s)');     ylabel('---> x(t)');
    title('1(c2) Discrete Time, Duty Cycle = 75%');
    grid on;

    %≤===== Function to find signal vector based on duty cycle =====>
    function [x] = findCycle(x, Duty, Ts, t)
       T_ = Ts*Duty/100;
       TP = Ts;
       mul = -1;
       for i = 1:length(t)
           x(i) = x(i) * mul;
           if t(i) ≥ TP
               TP = TP + Ts;
               mul = mul * -1;
           end
           if t(i) ≥ T_
               T_ = T_ + Ts;
               mul = mul * -1;
           end
```

```
        end
    end
```

The plot obtained is as follows:

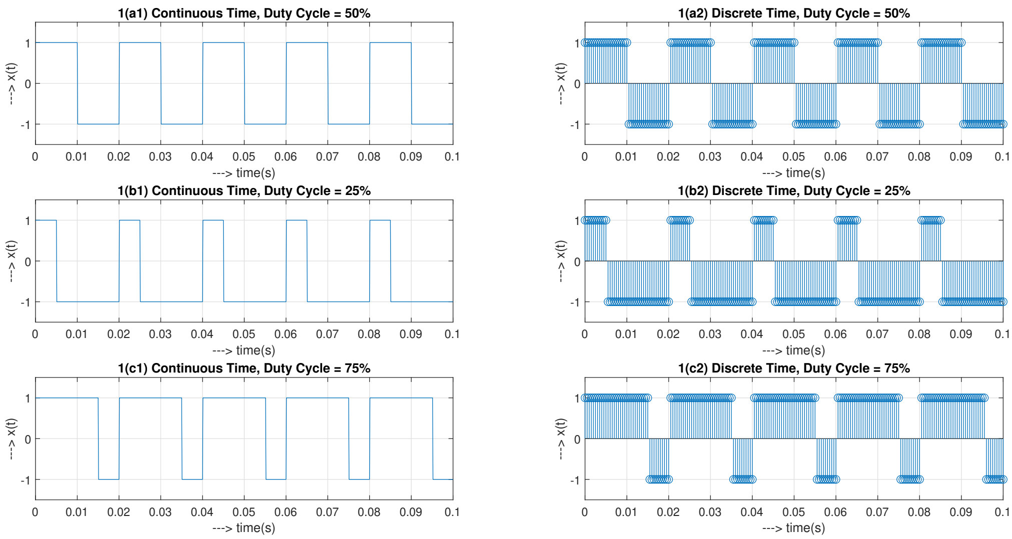

Figure 1: Plot obtained for Question 1

In the above plots, we observe that the time period for which the signal's value is 1 is directly proportional to the duty cycle. In first two plots with duty cycle of 50%, the signal is 1 for 0.01 seconds and -1 for 0.01 seconds in each cycle. Similarly, in next two plots with duty cycle of 25%, the signal is 1 for 0.005 seconds and -1 for 0.015 seconds. In the last two plots with duty cycle of 75%, the signal is 1 for 0.015 seconds and -1 for 0.005 seconds.

3.2 Question 2

The MATLAB code for the question is shown below:

```
clear all;
close all;
clc;

%<==== 2(a) =====>
n11 = -2;       n12 = 2;
n21 = -2;       n22 = 2;
subplot(3,4,1);
[y1, ny1] = FindConvolution(n11, n12, n21, n22);
stem(ny1, y1);
xlabel('---> n');    ylabel('---> convolution');
yticks(0:1:6);
title('2(a) Discrete-Time');
grid on;

subplot(3,4,2);
plot(ny1, y1);
```

```
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:6);       ylim([0 6]);
    title('2(a) Continuous-Time');
    grid on;

    %<==== 2(d(a)) =====>
    x1 = ones(1, n12 - n11 + 1);
    x2 = ones(1, n22 - n21 + 1);
    ny1 = (n11+n21):1:(n12+n22);
    subplot(3,4,3);
    stem(ny1, conv(x1, x2));
    xlabel('---> n');    ylabel('---> convolution');
    yticks(0:1:6);
    title('2(d(a)) Discrete-Time');
    grid on;

    y1 = conv(x1, x2);
    subplot(3,4,4);
    plot(ny1, y1);
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:6);       ylim([0 6]);
    title('2(d(a)) Continuous-Time');
    grid on;

    %<==== 2(b) =====>
    n11 = -2;       n12 = 2;
    n21 = 0;        n22 = 4;
    subplot(3,4,5);
    [y1, ny1] = FindConvolution(n11, n12, n21, n22);
    stem(ny1, y1);
    xlabel('---> n');    ylabel('---> convolution');
    yticks(0:1:6);
    title('2(b) Discrete-Time');
    grid on;

    subplot(3,4,6);
    plot(ny1, y1);
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:6);       ylim([0 6]);
    title('2(b) Continuous-Time');
    grid on;

    %<==== 2(d(b)) =====>
    x1 = ones(1, n12 - n11 + 1);
    x2 = ones(1, n22 - n21 + 1);
    ny1 = (n11+n21):1:(n12+n22);
    subplot(3,4,7);
    stem(ny1, conv(x1, x2));
    xlabel('---> n');    ylabel('---> convolution');
    yticks(0:1:6);
    title('2(d(b)) Discrete-Time');
    grid on;

    y1 = conv(x1, x2);
    subplot(3,4,8);
    plot(ny1, y1);
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:6);       ylim([0 6]);
    title('2(d(b)) Continuous-Time');
```

```
    grid on;

    %≤==== 2(c) =====>
    n11 = -5;      n12 = 5;
    n21 = -1;       n22 = 6;
    subplot(3,4,9);
    [y1, ny1] = FindConvolution(n11, n12, n21, n22);
    stem(ny1, y1);
    xlabel('---> n');    ylabel('---> convolution');
    yticks(0:1:8);       xticks(-6:2:12);
    xlim([-6 11]);
    title('2(c) Discrete-Time');
    grid on;

    subplot(3,4,10);
    plot(ny1, y1);
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:8);       xticks(-6:2:12);
    xlim([-6 11]);
    title('2(c) Continuous-Time');
    grid on;

    %≤==== 2(d(c)) =====>
    x1 = ones(1, n12 - n11 + 1);
    x2 = ones(1, n22 - n21 + 1);
    ny1 = (n11+n21):1:(n12+n22);
    subplot(3,4,11);
    stem(ny1, conv(x1, x2));
    xlabel('---> n');    ylabel('---> convolution');
    yticks(0:1:8);       xticks(-6:2:12);
    xlim([-6 11]);
    title('2(d(c)) Discrete-Time');
    grid on;

    y1 = conv(x1, x2);
    subplot(3,4,12);
    plot(ny1, y1);
    xlabel('---> t');    ylabel('---> convolution');
    yticks(0:1:8);       xticks(-6:2:12);
    xlim([-6 11]);
    title('2(d(c)) Continuous-Time');
    grid on;

    %≤==Function to find discrete time convolution ===>
    function [y, ny] = FindConvolution(n11, n12, n21, n22)
        x1 = ones(1, n12 - n11 + 1);
        x2 = ones(1, n22 - n21 + 1);
        ny = (n11+n21):1:(n12+n22);
        y = zeros(1, (n12+n22)-(n11+n21)+1);

        n1 = length(x1);
        n2 = length(x2);
        for n = 1:(n1+n2-1)
            for k = 1:n1
                if(n-k ≥ 0 && n-k < n2)
                    y(n) = y(n) + x1(k)*x2(n-k+1);
                end
            end
        end
```

```
end
```

The plot obtained is as follows:

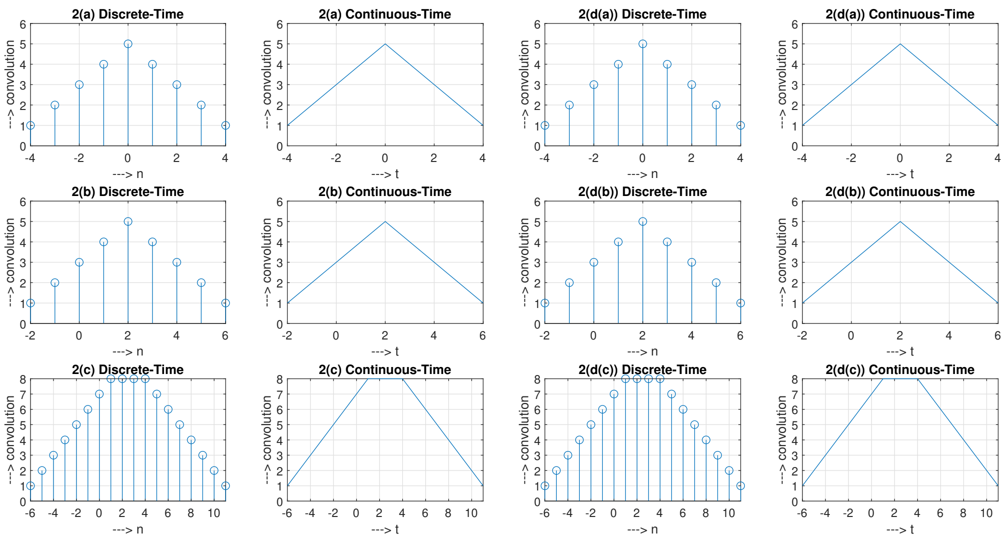

Figure 2: Plot obtained for Question 2

In the above figure, the plots in the first two columns are obtained by using the `FindConvolution(n11, ... n12, n21, n22)` function. The plots in third and fourth column are plotted using inbuilt `conv(x1, ... x2)`. The plots in the first row correspond to (a) part. The plots in the second row correspond to (b) part and the plots in third row correspond to (c) part. Here, we observe that the plots plotted using `conv(x1, x2)` are identical to the ones plotted using the `FindConvolution(n11, n12, ... n21, n22)` function.

3.3 Question 3

The MATLAB code for the question is shown below:

```
clear all;
close all;
clc;

A1 = 2;  A2 = 6;            %Given Parameters
f1 = 50; f2 = 100;

fs = 5000;
t1 = 0:1/fs:0.1;

m1 = A1*cos(2*pi*f1*t1);
m2 = A1*cos(2*pi*f1*t1) + A2*cos(2*pi*f2*t1);

%<==== 3(a1) =====>
subplot(3,2,1);      plot(t1, m1);
xlabel('---> time(s)');     ylabel('---> m_1(t)');
title('3(a1) m_1(t) signal');
```

```
grid on;

%≤==== 3(a2) =====>
subplot(3,2,2);     plot(t1, m2);
xlabel('---> time(s)');    ylabel('---> m_2(t)');
ylim([-8 10]);             yticks(-8:2:10);
title('3(a2) m_2(t) signal');
grid on;

%≤==== 3(b1) =====>
N1 = length(fft(m1));        %Number of samples
ts1 = 10/fs;                 %time
df1 = 1/(N1*ts1);            %frequency gap
t1 = [0:N1-1]*ts1;           %Time vector
m1 = A1*cos(2*pi*f1*t1);
m2 = A1*cos(2*pi*f1*t1) + A2*cos(2*pi*f2*t1);
Xf_shift1 = fftshift(fft(m1))/length(m1);  %perform zero-centered circular shift on fourier transform and normalize it
Ff_shift1 = [-N1/2:N1/2-1]*df1;

subplot(3,2,3);
plot(Ff_shift1, abs(Xf_shift1));
xlabel('---> Frequency(Hz)');    ylabel('---> M_1(f)');
title('3(b1) Frequency spectrum of m_1(t) signal');
xlim([-200 200]);
grid on;

%≤==== 3(b2) =====>
Xf_shift2 = fftshift(fft(m2))/length(m2);  %perform zero-centered circular shift on fourier transform and normalize it

subplot(3,2,4);
plot(Ff_shift1, abs(Xf_shift2));
xlabel('---> Frequency(Hz)');    ylabel('---> M_2(f)');
title('3(b2) Frequency spectrum of m_2(t) signal');
xlim([-200 200]);
grid on;

%≤==== 3(c1) =====>
subplot(3,2,5);
plot(Ff_shift1, angle(Xf_shift1));
xlabel('---> Frequency(Hz)');    ylabel('---> Phase(rad)');
title('3(c1) Phase spectrum of m_1(t) signal');
set(gca,'ytick',[-pi:pi/2:pi]);
set(gca,'yticklabels',{'-\pi','-\pi/2','0','\pi/2','\pi'});
xlim([-100 100]);
grid on;

%≤==== 3(c2) =====>
subplot(3,2,6);
plot(Ff_shift1, angle(Xf_shift2));
xlabel('---> Frequency(Hz)');    ylabel('---> Phase(rad)');
title('3(c2) Phase spectrum of m_2(t) signal');
xlim([-200 200]);
set(gca,'ytick',[-pi:pi/2:pi]);
set(gca,'yticklabels',{'-\pi','-\pi/2','0','\pi/2','\pi'});
grid on;
```

The plot obtained is as follows:

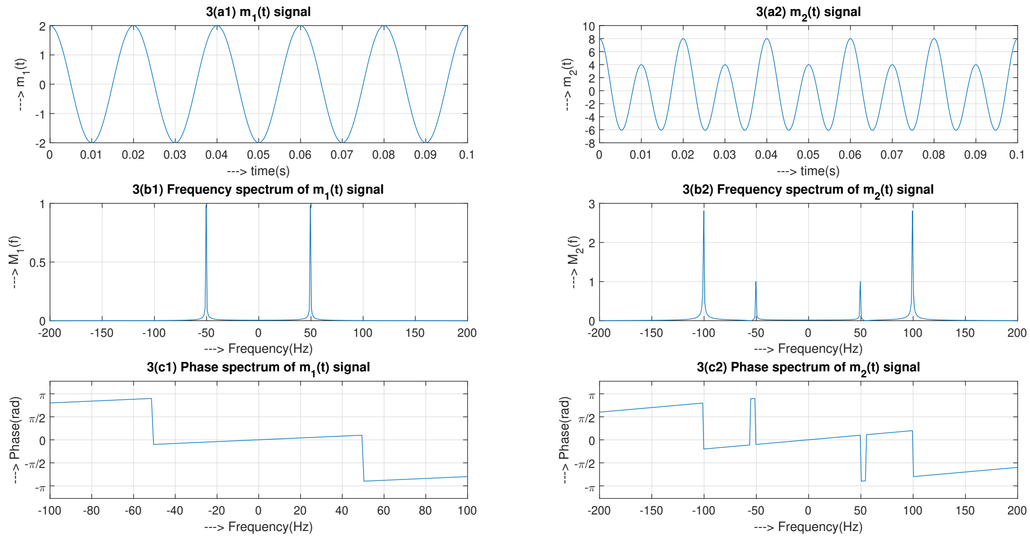

Figure 3: Plot obtained for Question 3

(a) We plotted $m_1(t)$ and $m_2(t)$ signal. Here, in the first plot, we observe that the time period of $m_1(t)$ is 0.02 seconds and its frequency is 50 Hz. This is a single tone signal. On the other hand, in second plot, we see a periodic signal whose time period is 0.02 seconds. This signal is combination of two signals namely $2cos(2\pi 50t)$ (time period $= 1/50$) and $6cos(2\pi 100t)$ (time period $= 1/100$). Mathematically, the time period should be LCM(1/50, 1/100) which is 0.02 seconds. Hence, the time period of the plotted signal corresponds to its mathematically calculated value. Moreover, this signal is a multi-tone signal.

(b) In the frequency spectrum of $m_1(t)$ signal, we see two spikes at ± 50 Hz. The amplitude of spikes is 1. **The tone of $\mathbf{m_1(t)}$ signal is 50 Hz**. In the frequency spectrum of $m_2(t)$ signal, we see four spikes, two at ± 50 Hz (corresponding to $2cos(2\pi 50t)$) and other two at ± 100 Hz (corresponding to $6cos(2\pi 100t)$). The amplitude of spikes at ± 50 Hz is 1 and the amplitude of spikes at ± 100 Hz is 3. **The tone of $\mathbf{m_2(t)}$ signal is at 50, 100 Hz.**

(c) In phase spectrum of $m_1(t)$ signal, we see a remarkable drop in phase at ± 50 Hz. Similarly, in phase spectrum of $m_2(t)$ signal, we see phase drop at ± 100 Hz along with a slight spike at ± 50 Hz.

4 Discussion and Conclusion

4.1 Question 1

The plotted signals are in accordance with the expected values.

4.2 Question 2

The plots obtained by `conv(x1, x2)` is identical to the ones plotted by calculating the value (both discrete time and continuous time). Hence, the function implemented to find convolution is correct.

4.3 Question 3

(a) Both the plotted signals ($m_1(t)$ and $m_2(t)$) are periodic in nature with time period of 0.02 seconds.

(b) Mathematically, the Fourier transform of $m_1(t)$ signal is:

$$2cos(2\pi 50t) <-> \delta(f-50) + \delta(f+50) \tag{1}$$

The obtained frequency spectrum of $m_1(t)$ signal has two spikes of height 1 at ± 50 Hz. Thus, the obtained plot is in agreement with the mathematical expression of Fourier transform.
The Fourier transform of $2cos(2\pi 50t)$ is:

$$2cos(2\pi 50t) <-> \delta(f-50) + \delta(f+50) \tag{2}$$

The Fourier transform of $6cos(2\pi 100t)$ is:

$$6cos(2\pi 100t) <-> 3(\delta(f-100) + \delta(f+100)) \tag{3}$$

Hence, the Fourier transform of $m_2(t)$ signal is:

$$2cos(2\pi 50t) + 6cos(2\pi 100t) <-> (\delta(f-50) + \delta(f+50)) + 3(\delta(f-100) + \delta(f+100)) \tag{4}$$

The obtained frequency spectrum of $m_2(t)$ signal has two spikes of height 1 at ± 50 Hz and two spikes of height 3 at ± 100 Hz. Thus, the obtained plot is in agreement with the mathematical expression of Fourier transform.
The tone for $m_1(t)$ signal is at ± 50 Hz.
The tone for $m_2(t)$ signal is at ± 50 and ± 100 Hz.