Lab 02

Arithmetic Operations, Discrete-time and Continuous-time Signals and Processing

Name: basil khowaja Muhammad Aqeel mehdi	Student ID: bk08432 mm08442

2.1 Objective

Performing complex-valued arithmetic, discrete-time and continuous-time signals and processing using MATLAB

In-Lab:

Task 1: Complex arithmetic in MATLAB to learn **real**, **imag**, **conj**, **abs**, **angle**, etc.
Task 2: Plotting a sinusoid tone in discrete time.
Task 3: Plotting unit step, sawtooth wave and triangular wave.
Task 4: Filtering data arrays using signal processing function **filter.**

Post-lab

Task 5: Further manipulation of complex numbers. Plotting exponential function, unit ramp and unit impulse signal.
Task 6: Using MATLAB arrays and manipulating arrays to represent continuous time signals, showing amplitude modulation.

2.2 Introduction

In this lab, elementary signals in continuous-time and discrete-time systems are introduced. We learn how to use MATLAB tools to generate and graphically represent various signals.

2.3 Complex Numbers:

A complex number is a number of the form $a + bi$, where a and b are real numbers and i is the imaginary unit, satisfying $i^2 = -1$. Consider a complex number $x = 10 + 20i$

real(x) % returns 10 as the real part of complex number

imag(x) % returns 20 as the imaginary part of complex number

conj(x) % returns 10 - 20i as conjugate of complex number

abs(x) % returns the magnitude of complex number

angle(x) % returns the angle of complex number in radian

[X, Y] = pol2cart(theta, rho) %Transforms polar coordinates to Cartesian, theta is in radian

[theta, rho] = cart2pol(X, Y) %Transform Cartesian coordinates to polar coordinates

Task 1: Complex Arithmetic in MATLAB

Find (a) real value, (b) imaginary value, (c) complex conjugate, (d) phase angle, (e) magnitude of the given complex numbers using MATLAB commands:

1. $15.5 + 3.175i$

```
x = 15.5000 + 3.1750i
a = 15.5000
b = 3.1750
c = 15.5000 - 3.1750i
d = 15.8218
e = 0.2020
```

```
%part1
x=15.5+3.175i
a=real(x)
b=imag(x)
c=conj(x)
d=abs(x)
e=angle(x)
```

2. $5\angle 45°$

   Code:

```
mag=5;
radians=deg2rad(45);
[X,Y] = pol2cart(radians,mag);
x=X+Y*i;
a=real(x)
b=imag(x)
c=conj(x)
d=abs(x)
e=angle(x)
```

Result:

```
a = 3.5355
b = 3.5355
c = 3.5355 - 3.5355i
d = 5
e = 0.7854
```

dis

2.4 Discrete time Signals

A signal is called a **discrete-time signal** if the independent variable (time) is defined in a discrete interval (e.g., the set of integer numbers), while the dependent variable is defined in a continuous set of values. In the following example, the discrete-time signal $x[n] = \cos[n]$ is plotted. Note that when referring to discrete time the variable **n** is typically used to represent the time. A discrete-time signal $x[n]$ is obtained by sampling a continuous-time signal $x(t)$ at a constant rate. Suppose that T_s is the sampling period, that is every T_s second, we sample the value of $x(t)$. Suppose also that $n \in \mathbb{Z}$, i.e., $n = 0, \pm 1, \pm 2, \pm 3, \dots$ The sequence of the samples $x[nT_s]$, $n \in \mathbb{Z}$ derived from the continuous-time signal $x(t)$ is sometimes called time series and denotes a discrete-time signal. The sampling period T_s is constant and thus can be omitted from the notation. In this lab, we assume that $T_s = 1$. In order to plot discrete-time signals, we use MATLAB command **stem**. Type **help stem** in MATLAB command window to learn more about **stem**.

Try 1: An example code for plotting a sinusoid with unit sampling time ($T_s = 1$)

```
n = 0:10;                  % Discrete-time n is defined which goes from 0 to 10 in unity sampling interval.
y = cos(n);                % A sinusoidal function y[n] is defined attaining values −1 ≤ y[n] ≤ 1.

figure; plot(n, y, ':o');  % A continuous-time graphical plot of the signal y[n] where circular markers are used to represent values connected with a dotted line.
figure; stem(n, y);        % A more appropriate discrete-time graphical representation.
```

Try 2: An example code for plotting a sinusoid with non-unit sampling time ($T_s \neq 1$) Suppose that the <u>sampling interval</u> is not unity, say $T_s = 0.1$ (i.e sampling frequency $f_s = 10Hz$), then the discrete time is defined as:

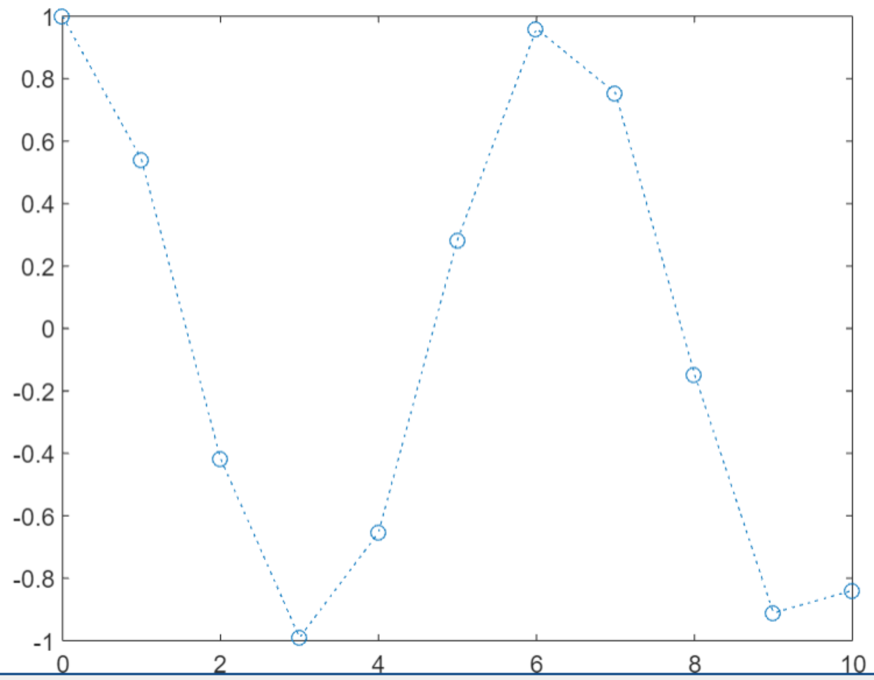
```
Ts = 0.1;      % Defining sampling time, the time interval between two consecutive samples
Fs = 1/Ts;     % Defining sampling frequency, the number of samples per second
t = 0:Ts:10;   % Defining time span
y = cos(t);    % Obtaining a discrete-time signal
figure;        % Using subplots to depict two plots in one figure
subplot 121; plot(t, y, ':o');
subplot 122; stem(t, y);
```

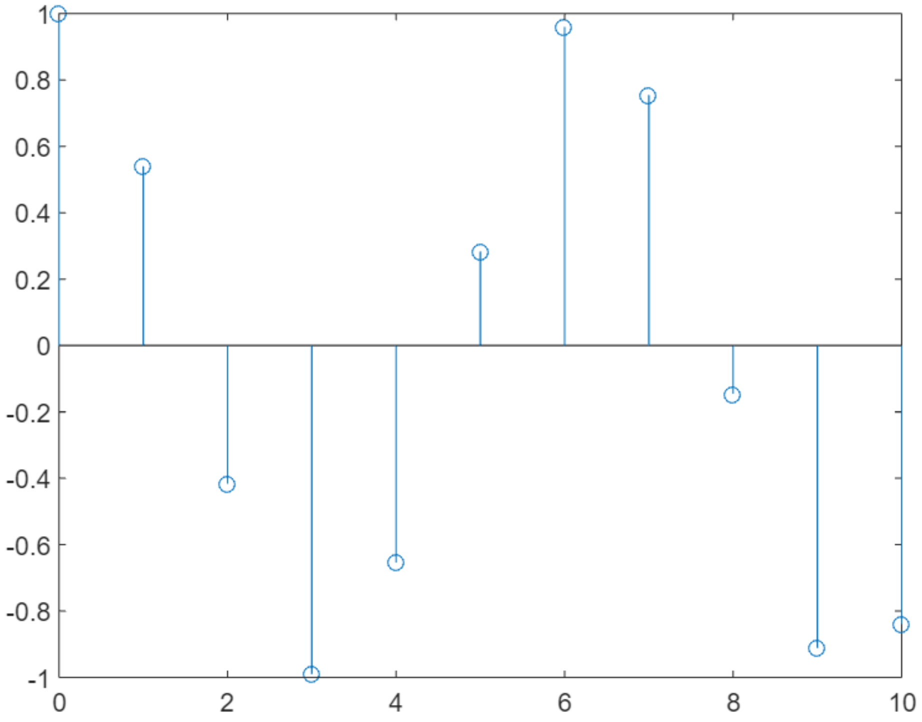
Task 2(a):

Show plots with proper titles, axis labelling and compare the results of Try 1 and Try 2, and briefly report your observation about the selection of sampling time and also state the difference between using **plot** and **stem** commands.

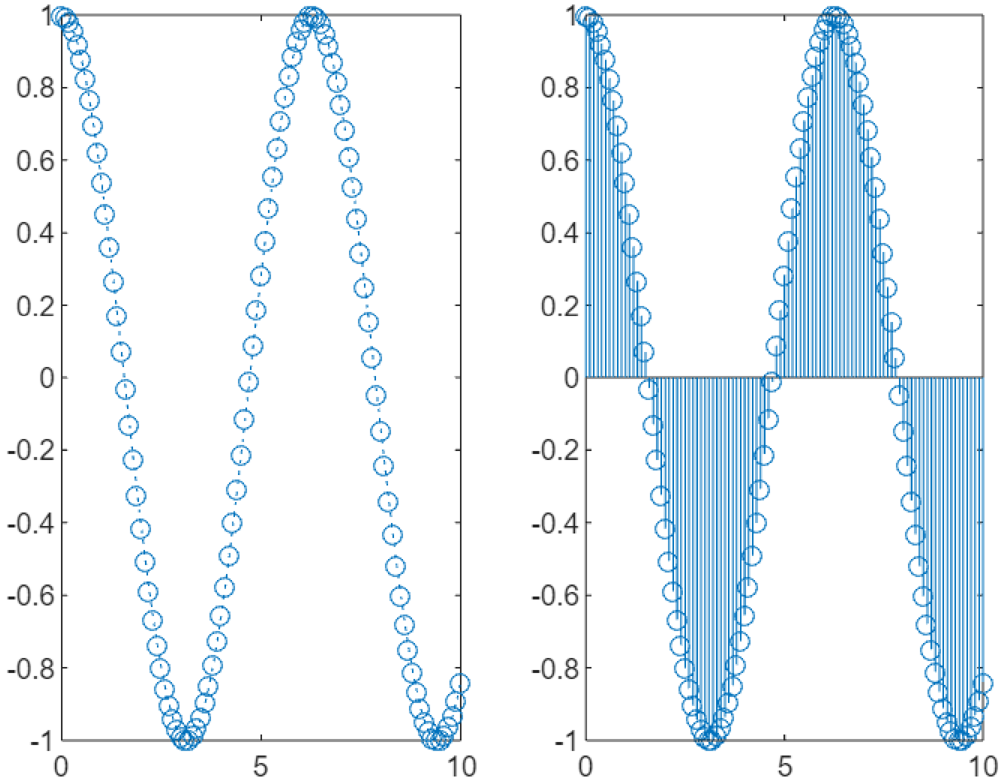
Ans: The difference between plot and stem is that the plot command only plots the points and joins them whereas the stem command also makes a line between the points and the x-axis.

```
y = 1×11
      1.0000    0.5403   -0.4161   -0.9900   -0.6536    0.2837    0.9602    0.7539   -0.1455 ...
```




1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10

```
y = 1×101
      1.0000    0.9950    0.9801    0.9553    0.9211    0.8776 ...
```



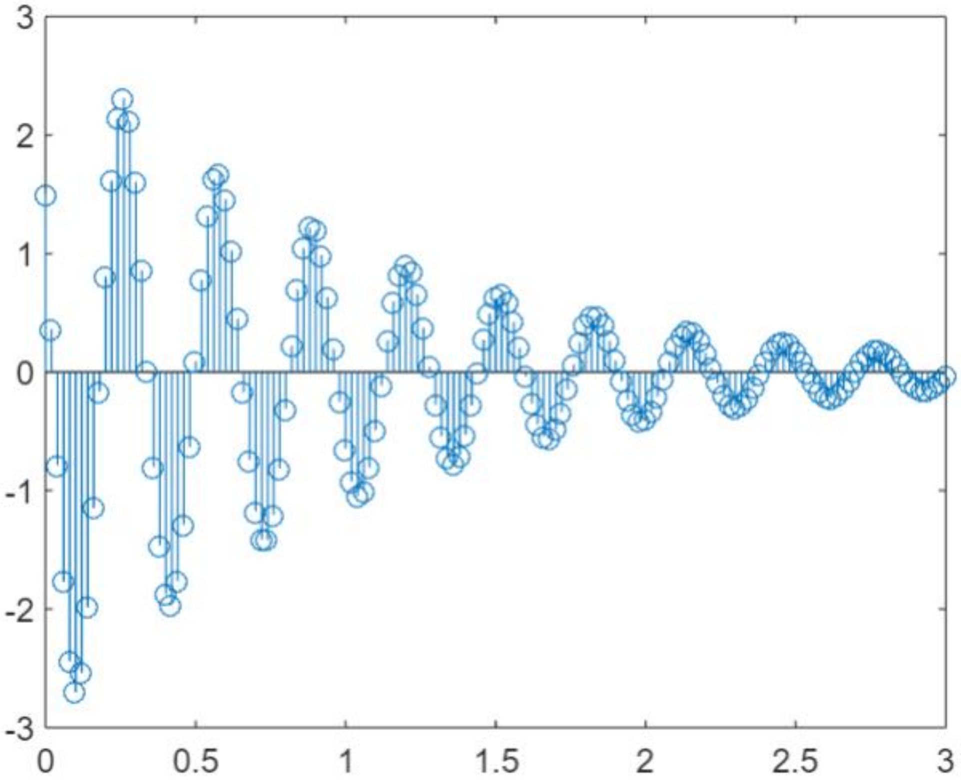
Task 2(b):

Sampling a continuous signal with defined sampling rate $f_{S.}$ ($T_s = 1/f_S$)

Given f_S (the number of samples per second), the sampling time is obtained as $T_s = 1/f_S$.

☞Sample the continuous signal $x(t) = 3e^{-t}\cos(20t + \pi/3)$ at the rate 50 Hertz (that is 50 samples per second), obtain a plot using **stem** for $0 \le t \le 3$ second.

```
%Task 2(b)
Ts=0.02;
t=0:Ts:3;
y=3*exp(-t).*cos(20*t+pi/3);
figure;stem(t,y);
```



2.5 Continuous-Time Signals

Continuous-time signals are defined for a continuum of values of the independent variable *time*. Examples of some continuous-time signals are speech signal and atmospheric pressure. A continuous time signal is denoted by $x(t)$, where t refers to independent variable time. The following MATLAB commands are useful in plotting continuous time signals:

sawtooth(t,xmax) % where xmax is a scalar number between 0-1 representing angle $0 - 2\pi$

stairs(X,Y) %draws a stair step graph of the elements of Y, at the locations specified in

mod(a,m) %returns the remainder after division of a by m, where a is the dividend and m is the divisor

stem(n, y); % Using stem is more appropriate when dealing with discrete-time signals.

Try 3: Plotting Unit Step and Sawtooth Functions

The unit step function is defined as: $u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$

Use the following MATLAB code to generate a unit-step function and sawtooth wave of frequency 1Hz, keeping time interval of 1 second.

☞ Show proper comments stating the function and explanation of each new command used.

```
clc; clear
t = -20:1:20;
y = mod(t,5); %the time vector is divided by 5 and remainder is used to plot the values resulting in a sawooth wave when plotted w.r.t time, whereas the second argument 5 represents the time period/duration of one sawtooth pulse
z = (t>=0); % t>=0 returns logic ‘1’ or ‘0’
subplot(1,2,1);
stairs(t,z,'linewidth',2)
title('Unit-step signal');
xlabel('Time');
ylabel('Amplitude');
axis([-20 20 0 1.1]);
subplot(1,2,2);
plot(t,y,'linewidth',1.5)
title('Sawtooth signal');
xlabel('Time');
ylabel('Amplitude');
axis([-20 20 0 5.2])
```

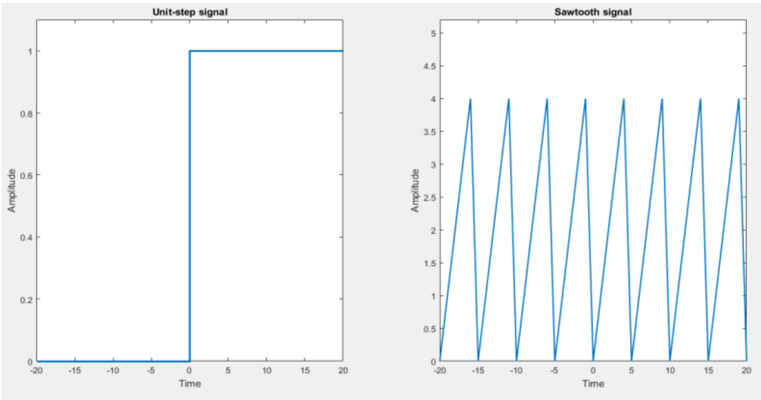
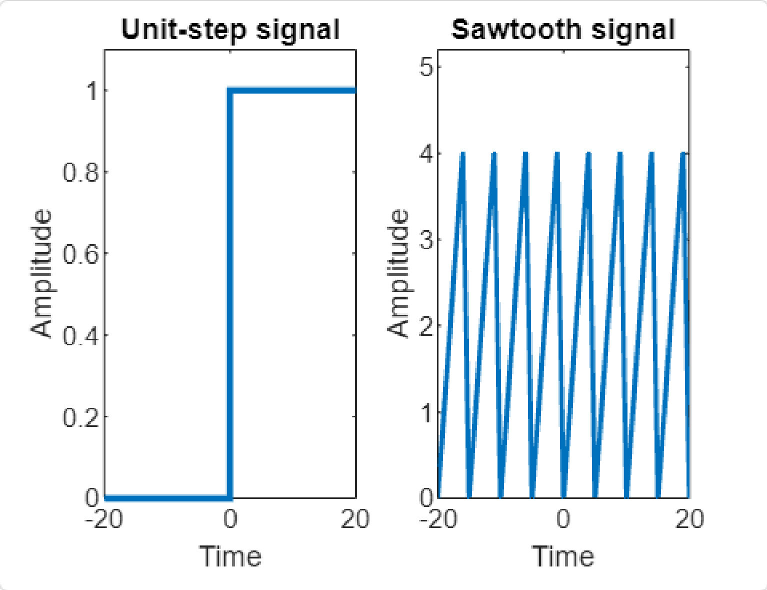


Figure 1: Obtaining side-by-side plots using **subplot**



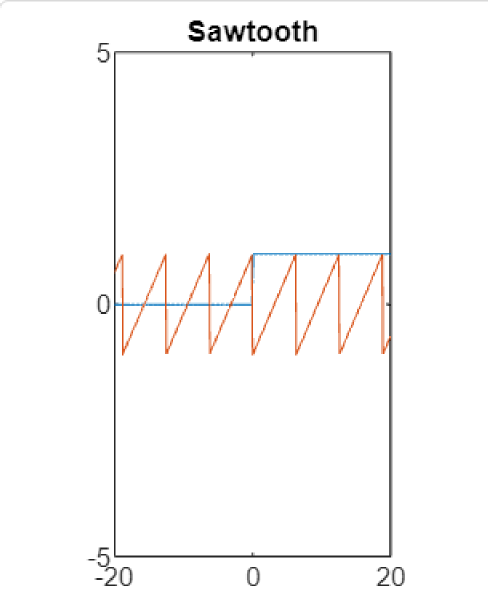

```
%Try 3:
t = -20:1:20; % Defining a time vector from -20 to 20 with a step of 1
y = mod(t,5); %calculating modulus of each element in t with respect to 5
z = (t >= 0); %making a binary signal z for indicating whether each element in 't' is non-negative

% Figure 1: Obtaining side-by-side plots using subplot
figure;

subplot(1,2,1);
stairs(t, z, 'linewidth', 2);
title('Unit-step signal'); %title of the plot
xlabel('Time'); %giving x label name
ylabel('Amplitude'); %giving y label name
axis([-20 20 0 1.1]);

subplot(1,2,2);
plot(t, y, 'linewidth', 1.5);
title('Sawtooth signal'); %title of the plot
xlabel('Time'); %x label name
ylabel('Amplitude'); %y label name
axis([-20 20 0 5.2]);
```

Task 3:

Plot the unit step function and sawtooth wave using command **sawtooth** on same figure and use proper axis labeling, legends and axis limits.

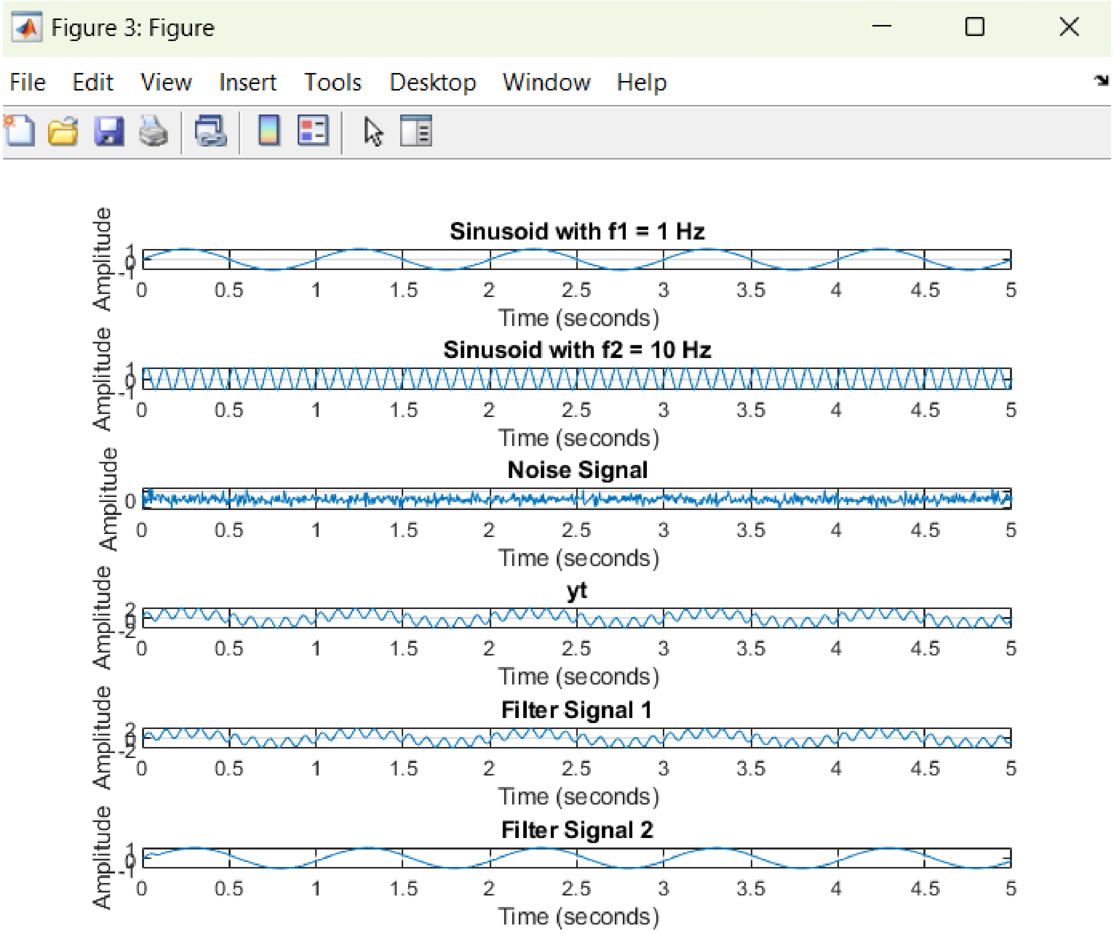
☞ Show proper comments stating the function and explanation of each new command used.

```
figure;
x = -20:0.1:20
subplot(1,2,1);
plot(x,heaviside(x));
title("Sawtooth");
hold on
plot(x,sawtooth(x))
axis([-20,20,-5,5])
hold off
```



task 4: Processing (Filtering) Signals with MATLAB

```
%Task 4
% Time scale from 0 to 5 seconds with interval of 0.005
t = 0:0.005:5;

% Frequencies of the sinusoids
f1 = 1;  % 1 Hz
f2 = 10; % 10 Hz

% Generating sinusoids
xt1 = sin(2*pi*f1*t);
xt2 = sin(2*pi*f2*t);
noise=randn(length(t))*0.2
yt=xt1+xt2+noise*0.2;
zt1=filter(ones(1,5)/5,1,yt)
zt2=filter(ones(1,20)/20,1,yt)
% Plotting
figure;

subplot(6,1,1);
plot(t, xt1);
title('Sinusoid with f1 = 1 Hz');
xlabel('Time (seconds)');
ylabel('Amplitude');
grid on;

subplot(6,1,2);
plot(t, xt2);
title('Sinusoid with f2 = 10 Hz');
xlabel('Time (seconds)');
ylabel('Amplitude');
grid on;

subplot(6,1,3)
plot(t,noise);
title('Noise Signal');
xlabel('Time (seconds)');
ylabel('Amplitude');
grid on;

subplot(6,1,4)
plot(t,yt);
title('yt');
xlabel('Time (seconds)');
ylabel('Amplitude');
```



Observation: In this filter, the noisy signal is effectively eliminated, resulting in a much clearer sine signal. This is achieved by removing the high frequency noise that was present in the original signal. When N is set to 5, only the extremely high frequencies are eliminated. However, when N is increased to 20, the less high frequencies are filtered out, resulting in a low frequency output.

Report your observations and understanding about the role of averaging filter in this experiment.

Post-Lab

Task 5:

1. For the given complex numbers, find the real value, the imaginary value, the complex conjugate, the phase in degrees**,** and the magnitude using MATLAB commands:

a) $(10.09 + 55.3i) + (20 \angle 85.5°)$
b) $15\angle 10° - 4 + 10i$

2. Plot unit ramp and unit impulse signal.

Task 5:

a) $(10.09 + 55.3i) + (20 \angle 85.5°)$

Code:

```
%Postlab task1
x=(10.09+55.3i)
magnitude=20;
radians=deg2rad(85.5);
[X,Y] = pol2cart(radians,mag);
r = real(x) + X;
s = imag(x) + Y;


c = r+s*i;
a=real(c)
b=imag(c)
c=conj(c)
d=abs(c)
e=angle(c)
```

Result:

```
x = 10.0900 +55.3000i
a = 10.4823
b = 60.2846
c = 10.4823 -60.2846i
d = 61.1891
e = -1.3986
```

b) $15\angle 10° - 4 + 10i$

Code:

```
radi = deg2rad(10);
[X,Y] = pol2cart(radi,15);
a = -4+10i;
r = X-real(a);
img = Y - imag(a);
c = r+img*i;
a1=real(c)
b=imag(c)
c=conj(c)
d=abs(c)
e=angle(c)
```

Result:

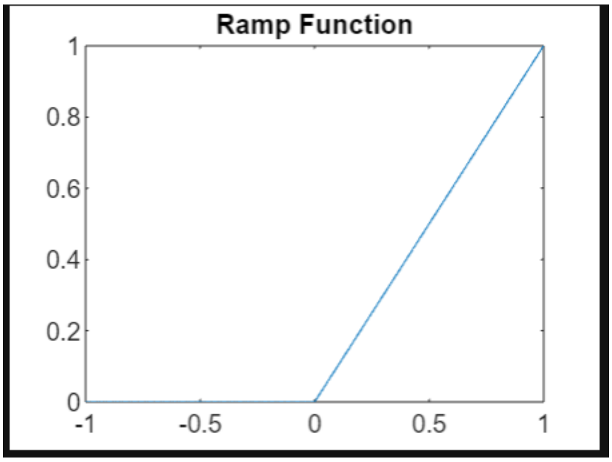
```
a1 = 18.7721
b = -7.3953
c = 18.7721 + 7.3953i
d = 20.1763
e = 0.3753
```

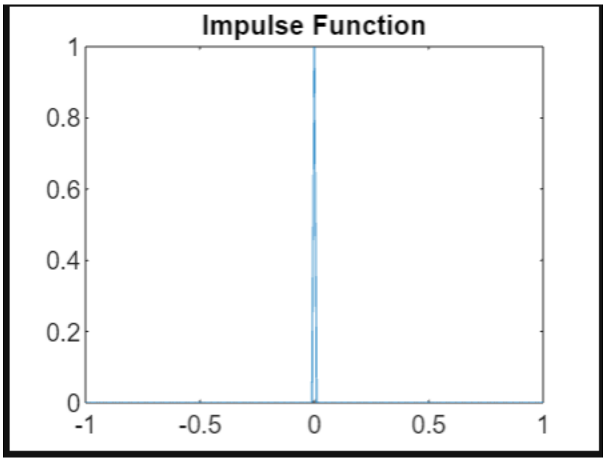
Task 5(b)

Code:

```
clc
close all
clear all
t = (-1:0.01:1)';
impulse = t==0;
unitstep = t>=0;


ramp = t.*unitstep; %ramp function multiplying it with unit step so it  exists
for t greater than or equal to zero
plot(t,ramp),title("Ramp Function")
% plotting the impulse
plot(t,impulse),title("Impulse Function")
```

Results:





Task 6: Amplitude Modulation

Analog communication is the process of conveying (sending, receiving, and processing) of information through radio, telephone, television etc. by using continuous signals or analog signals. Modulation is the process of superimposing the information contents of a modulating signal on a carrier signal (which is of high frequency).

Amplitude modulation (**AM**) is a modulation method used in electronic communication, most commonly for transmitting information via a radio carrier wave signal. Consider two sine wave signals -- one is the message signal of frequency f_m = 1Hz and the other is the carrier signal of frequency f_c = 10Hz – which are multiplied together in a specific manner, we get a modulated signal.

Note that, in MATLAB, to multiply two arrays, we use a "dot" before the multiplication symbol. Like, to multiply two arrays, *A* and *B*, we use *A* .**B*

Message signal is given by:	$m(t) = A_m \sin(2\pi f_m \text{ t})$
Carrier signal is given by:	$c(t) = A_c \sin(2\pi f_c \text{ t})$
Modulated signal is given by:	$s(t) = (1 + \mu A_m \sin(2\pi f_m \text{ t})) .* A_c \sin(2\pi f_c \text{ t})$

where A_m and A_c are the magnitudes of message and carrier signals, respectively. The symbol μ denotes modulation index and it is a positive number, which is usually, less than unity. For this experiment, consider $\mu = 0.5$.

- Plot message signal, carrier signal and the modulated signal in subplots 1, 2, and 3, respectively (using subplot obtain a grid of three-by-one plots). Label the axis and give proper titles.

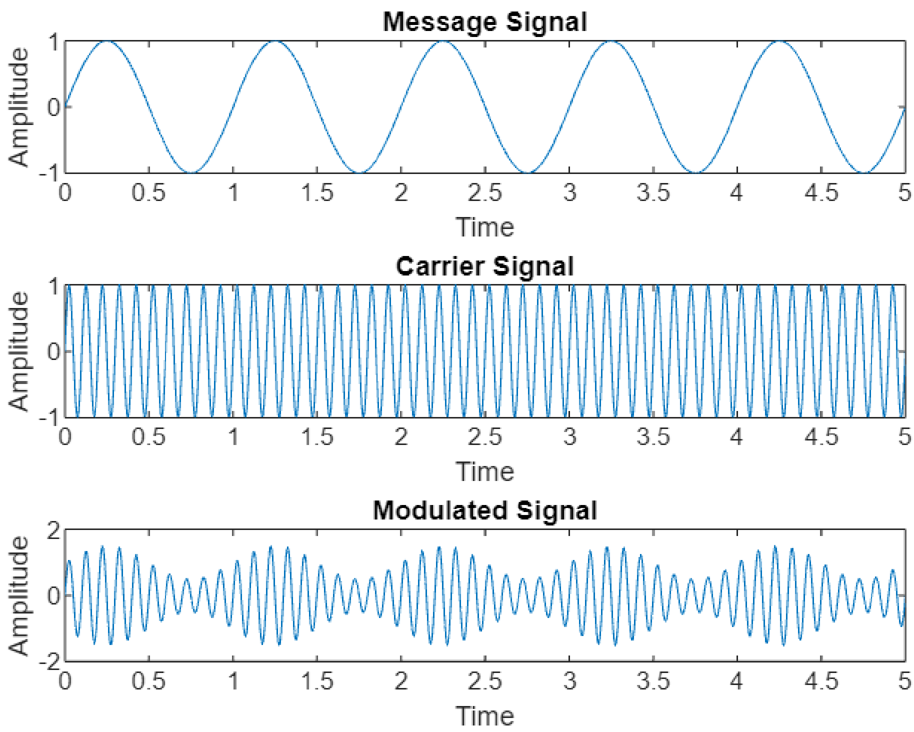
Note: In MATLAB, we simply write **x** or **xt** to represent a signal $x(t)$; and avoid writing **x(t).**

Code:

Live Editor - C:\Users\Dell\Documents\MATLAB\lab2.mlx *

lab1post.mlx | untitled.m | lab2.mlx *

```matlab
ylabel('Amplitude');

t = 0:0.005:5;
fm = 1;
fc = 10;
mu = 0.5;
mt = sin(2*pi*fm*t);
ct = sin(2*pi*fc*t);
st = (1 + (mu*mt)).*ct;
figure;
subplot 311; plot (t, mt)
title('Message Signal');
xlabel('Time');
ylabel('Amplitude');
subplot 312; plot (t, ct)
title('Carrier Signal');
xlabel('Time');
ylabel('Amplitude');
subplot 313; plot (t, st)
title('Modulated Signal');
xlabel('Time');
ylabel('Amplitude');
```



Assessment Rubric

Lab 02

Lab 2: Arithmetic Operations, Discrete-time and Continuous-time Signals and Processing

Name:	**Student ID:**

Marks distribution:

<table>
<tr><th></th><th>Tasks</th><th>LR2
Code</th><th>LR5
Results</th><th>AR4
Report
Submission</th><th>AR7
Comments</th></tr>
<tr><td rowspan="4">In-Lab</td><td>Task 1</td><td>/8</td><td>/4</td><td rowspan="6">/4</td><td>/2</td></tr>
<tr><td>Task 2</td><td>/4</td><td>/12</td><td></td></tr>
<tr><td>Task 3</td><td>/4</td><td>/4</td><td>/4</td></tr>
<tr><td>Task 4</td><td>/12</td><td>/8</td><td>/4</td></tr>
<tr><td>Post Lab</td><td>Task 5</td><td>/8</td><td>/8</td><td></td></tr>
<tr><td></td><td>Task 6</td><td>/4</td><td>/8</td><td>/2</td></tr>
<tr><td colspan="2">Total =100</td><td>/40</td><td>/44</td><td>/4</td><td>/12</td></tr>
<tr><td colspan="2">CLO Mapped</td><td>CLO1</td><td>CLO1</td><td>CLO3</td><td>CLO3</td></tr>
</table>

CLO	Total Points	Points Obtained
1	96	
3	4	
Total	**100**	

For description of different levels of the mapped rubrics, please refer the provided Lab Evaluation Assessment Rubrics and Affective Domain Assessment Rubrics.

Lab Evaluation Assessment Rubric

#	Assessment Elements	Level 1: Unsatisfactory Points 0-1	Level 2: Developing Points 2	Level 3:Good Points 3	Level 4:Exemplary Points 4
LR2	Program/Code/ Simulation Model/ Network Model	Program/code/simulation model/network model does not implement the required functionality and has several errors. The student is not able to utilize even the basic tools of the software.	Program/code/simulation model/network model has some errors and does not produce completely accurate results. Student has limited command on the basic tools of the software.	Program/code/simulation model/network model gives correct output but not efficiently implemented or implemented by computationally complex routine.	Program/code/simulation /network model is efficiently implemented and gives correct output. Student has full command on the basic tools of the software.
LR3	Troubleshooting	Unable to identify the fault/minimal effort show in troubleshooting.	Able to identify the fault but unable to remove it.	Able to identify the fault but partially removes it.	Able to identify the fault and takes necessary steps and actions to correct it.
LR4	Data Collection	Measurements are incomplete, inaccurate and imprecise. Observations are incomplete or not included. Symbols, units and significant figures are not included.	Measurements are somewhat inaccurate and imprecise. Observations are incomplete or vague. Major errors are there in using symbols, units and significant digits.	Measurements are mostly accurate. Observations are generally complete. Minor errors are present in using symbols, units and significant digits.	Measurements are both accurate and precise. Data collection is systematic. Observations are very thorough and include appropriate symbols, units and significant digits and task completed in due time.
LR5	Results & Plots	Figures/ graphs / tables are not developed or are poorly constructed with erroneous results. Titles, captions, units are not mentioned. Data is presented in an obscure manner.	Figures, graphs and tables are drawn but contain errors. Titles, captions, units are not accurate. Data presentation is not too clear.	All figures, graphs, tables are correctly drawn but contain minor errors or some of the details are missing.	Figures / graphs / tables are correctly drawn and appropriate titles/captions and proper units are mentioned. Data presentation is systematic.
LR6	Calculations	Formulae used and/or calculations are incorrect. Units are not mentioned with results.	Formulae used are correct but there are few mistakes in calculation which lead to incorrect results.	Formulae used and end results are correct. Complete working steps are not shown. Proper units are not mentioned with results.	Formulae used, end results and all calculations are correct with all the intermediate steps clearly shown. Units are mentioned with results.
LR9	Report	All the in-lab tasks are not included in report and / or the report is submitted too late.	Most of the tasks are included in report but are not well explained. All the necessary figures / plots are not included. Report is submitted after due date.	Good summary of most the in-lab tasks is included in report. The work is supported by figures and plots with explanations. The report is submitted timely.	Detailed summary of the in-lab tasks is provided. All tasks are included and explained well. Data is presented clearly including all the necessary figures, plots and tables.
LR11	Design	Proposed design is unsubstantiated or does not satisfy most of given constraints. The breakdown of system into features is incomplete and still at lower resolution.	Justification for proposed design is weak, and it only satisfies some constraints. The breakdown of system is missing some features and resolution is not appropriate at some places.	Proposed design is substantiated, preferably through proper analysis, and satisfies most given constraints. The breakdown of system into features seems complete, but resolution is not appropriate at some places.	Proposed design is substantiated, preferably through proper analysis, and satisfies all given constraints. The breakdown of system into features seems complete and at appropriate resolution, given students' level.
AR4	*Report Submission	Not accepted after 1 week.	Late submission after 2 days and within 1 week.	Late submission after the lab timing and within 2 days of the due date.	Timely submission of the report and in the lab time.
AR7	Report Content/Code comments	Most of the questions are not answered / figures are not labelled/ titles are not mentioned / units are not mentioned. No comments are present in the code.	Some of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Few comments are stated in the code.	Majority of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Comments are stated in the code.	All the questions are answered, figures are labelled, titles are mentioned and units are properly mentioned. Proper comments are stated in the code.