National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering and Computer Science

Faculty Member: Dr. Salman Ghafoor

Dated: 3/10/2022

Semester: 5th

Section: BEE 12C

EE-232: Signals and Systems

Lab 4: Introduction to Complex Exponentials

Group Members

Name	Reg. No	PLO4 - CLO3 **Viva / Quiz / Lab Performance** **5 Marks**	PLO5 - CLO3 **Analysis of data in Lab Report** **5 Marks**	PLO8 - CLO4 **Modern Tool Usage** **5 Marks**	PLO9 - CLO4 **Ethics and Safety** **5 Marks**
Danial Ahmad	331388				
Muhammad Umer	345834				
Syeda Fatima Zahra	334379				

1 Table of Contents

2 Introduction to Complex Exponentials

2.1 Objectives

The goal of this laboratory is to gain familiarity with complex numbers and their use in representing sinusoidal signals such as $x(t) = A\cos(wt + \phi)$ as complex exponentials.

- How to work with complex numbers in MATLAB
- Familiarization with MATLAB function and commands for Complex Exponentials
- Sinusoid addition using complex exponentials

2.2 Equipment

Software

- *MATLAB*

2.3 Lab Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

- Lab objectives
- MATLAB codes
- Results (Graphs/Tables) duly commented and discussed
- Conclusion

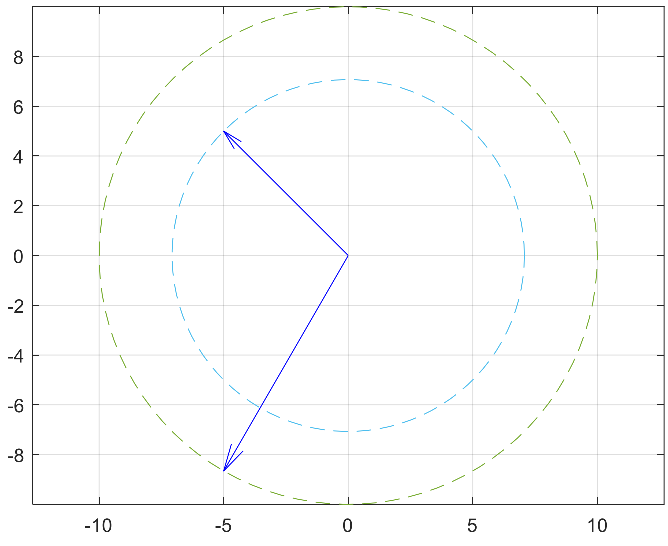
3 Pre-Lab

3.1 Complex Numbers

Use $z_1 = 10e^{-j2\pi/3}$ and $z_2 = -5 + j5$ for all parts of this section.

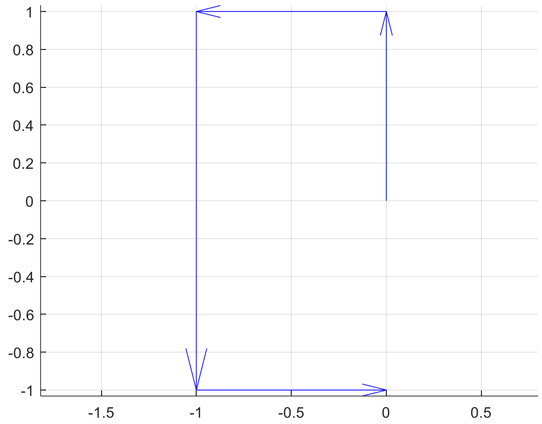
1. Enter the complex numbers z1 and z2 in MATLAB. Plot them with zvect() and print them with zprint(). When unsure about a command, use help. Whenever you make a plot with zvect() or zcat(), it is helpful to provide axes for reference. An x-y axis and the unit circle can be superimposed on your zvect() plot by doing the following: *hold on, zcoords, ucplot, hold off*

```
% Definition
z1 = 10 * exp(-1j * 2 * pi / 3);
z2 = -5 + 5j;
org = 0 + 0j;
zcoords('-');
zvect([z1 z2]);

% Plot
hold on
ucplot(abs(z1), org, '--');
ucplot(abs(z2), org, '--');
zprint([z1 z2])
```

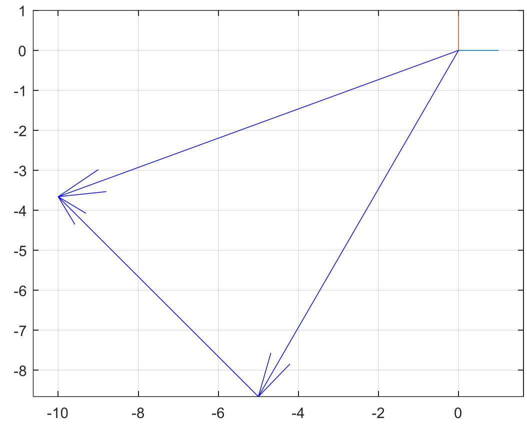



```
Z =      X     +      jY    Magnitude    Phase    Ph/pi    Ph(deg)
        -5          -8.66          10   -2.094   -0.667    -120.00
        -5              5       7.071    2.356    0.750     135.00
```

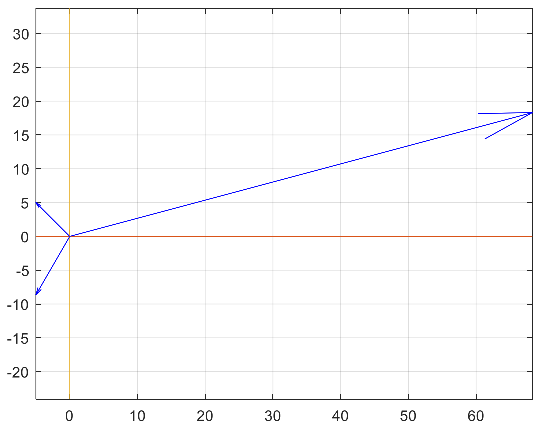
2. The function zcat() can be used to plot vectors in a “head-to-tail” format. Execute the statement zcat([j,-1,-2j,1]); to see how zcat() works when its input is a vector of complex numbers.



3. Compute z1 + z2 and plot the sum using zvect(). Then use zcat() to plot z1 and z2 as 2 vectors head-to-tail, thus illustrating the vector sum. Use hold on to put all 3 vectors on the same plot.

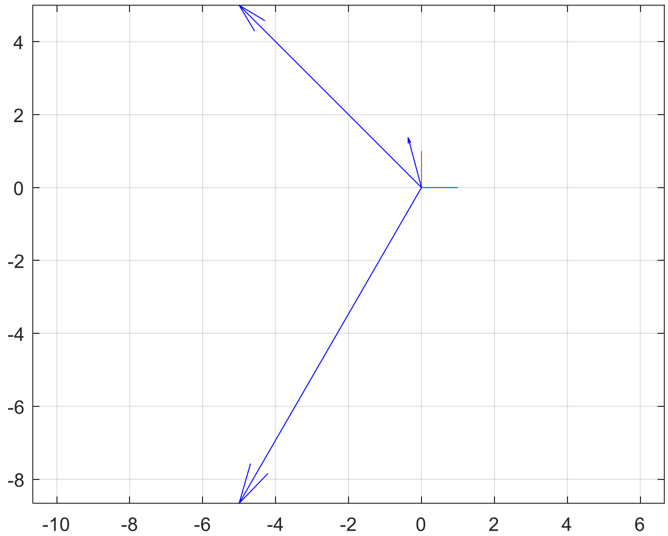


4. Compute z1z2 and plot the answer using zvect() to show how the angles of z1 and z2 determine the angle of the product. Use zprint() to display the result numerically.



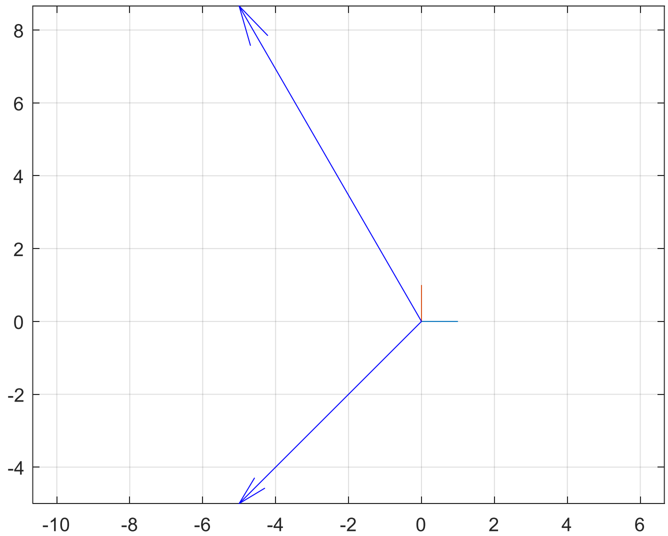
Z =	X	+	jY	Magnitude	Phase	Ph/pi	Ph(deg)
	-5		-8.66	10	-2.094	-0.667	-120.00
	-5		5	7.071	2.356	0.750	135.00
	68.3		18.3	70.71	0.262	0.083	15.00

5. Compute z2/z1 and plot the answers using zvect() to show how the angles of z1 and z2 determine the angle of the quotient. Use zprint() to display the result numerically.



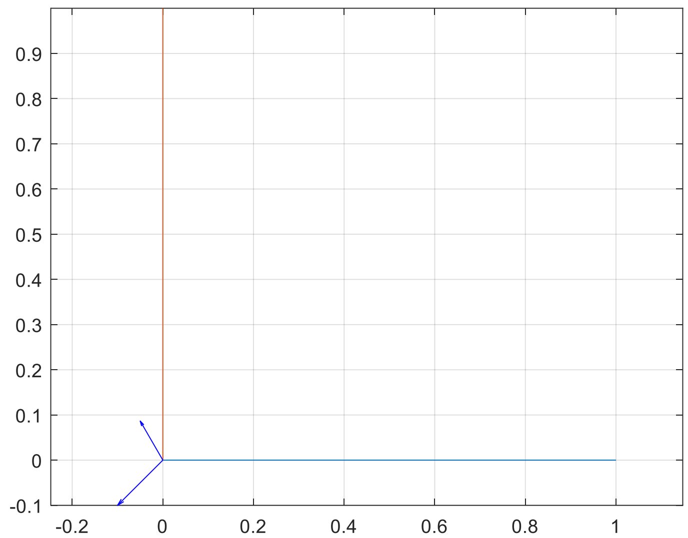
Z =	X	+	jY	Magnitude	Phase	Ph/pi	Ph(deg)
	-5		-8.66	10	-2.094	-0.667	-120.00
	-5		5	7.071	2.356	0.750	135.00
	-0.366		1.366	1.414	1.833	0.583	105.00

6. Compute the conjugate z^* for both z1 and z2 and plot the results. In MATLAB, see help conj. Display the results numerically with zprint.



Z =	X	+	jY	Magnitude	Phase	Ph/pi	Ph(deg)
	-5		8.66	10	2.094	0.667	120.00
	-5		-5	7.071	-2.356	-0.750	-135.00

7. Compute the inverse 1/z for both z_1 and z_2 and plot the results. Display the results numerically with zprint.




```
Z =       X      +       jY      Magnitude     Phase      Ph/pi     Ph(deg)
        -0.05          0.0866          0.1     2.094      0.667     120.00
         -0.1            -0.1       0.1414    -2.356     -0.750    -135.00
```

4 Lab Tasks

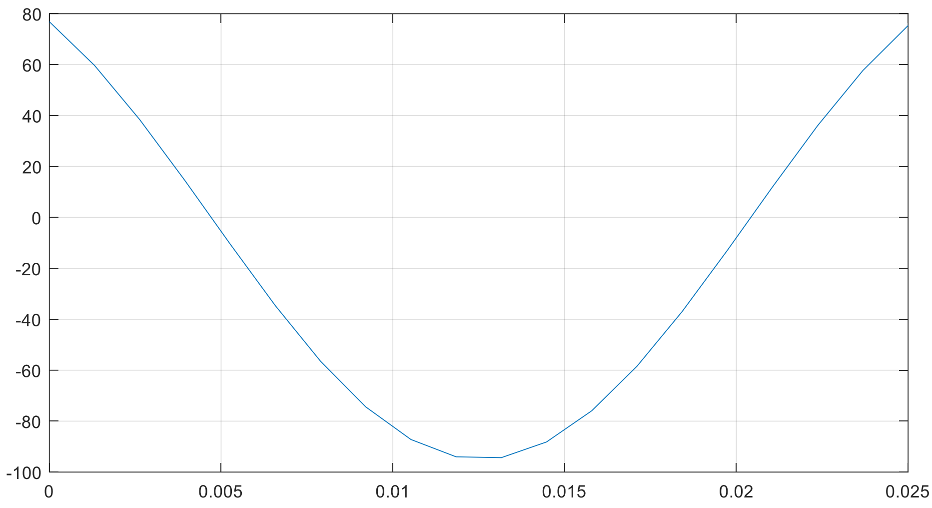
4.1 M-File to Generate a Sinusoid

Write a function that will generate a single sinusoid, $x(t) = A\cos(wt + \phi)$, by using four input arguments: amplitude (A), frequency (ω), phase (ϕ) and duration (dur). The function should return two outputs: the values of the sinusoidal signal (x) and corresponding times (t) at which the sinusoid values are known. Make sure that the function generates 20 values of the sinusoid per period. Call this function one cos(). Demonstrate that your one_cos() function works by plotting the output for the following parameters: A = 95, ω = 200 rad/sec, $\phi = \pi/5$ radians, and dur = 0.025 seconds. Be prepared to explain to the lab instructor features on the plot that indicate how the plot has the correct period and phase. What is the expected period in millisec?

```
function [x, t] = one_cos(A, w, phi, dur)
    t = linspace(0, dur, 20);
    x = A * cos(w * t + phi);
    plot(t, x)
    grid
end
```

Common Window

[x, t] = one_cos(95, 200, pi/5, 0.025);



→ w = 200 rad/sec → w = 2πf → f = 200/2π = 100/π
→ T = 1/f → 0.0314 → **31.4 ms**

4.2 Sinusoidal Synthesis with an M-File

Write an M-file called syn_sin.m that will synthesize a waveform in the form of:

$$x(t) = \Re e(\sum_{k=1}^{N} X_k e^{j2\pi f_k t})$$

Although 'for' loops are rather inefficient in MATLAB but you must write the function with one loop in this lab. The first few statements of the M-file are the comment lines—they should look like:

```
function [xx,tt] = syn_sin(fk, Xk, fs, dur, tstart)
%SYN_SIN Function to synthesize a sum of cosine waves
% usage:
% [xx,tt] = syn_sin(fk, Xk, fs, dur, tstart)
% fk = vector of frequencies
% (these could be negative or positive)
% Xk = vector of complex amplitudes: Amp*e^(j*phase)
% fs = the number of samples per second for the time axis
% dur = total time duration of the signal
% tstart = starting time (default is zero, if you make this input optional)
% xx = vector of sinusoidal values
% tt = vector of times, for the time axis
% Note: fk and Xk must be the same length.
% Xk(1) corresponds to frequency fk(1),
% Xk(2) corresponds to frequency fk(2), etc.
```

```
function [xx, tt] = syn_sin(fk, Xk, fs, dur, tstart)
    %SYN_SIN Function to synthesize a sum of cosine waves
    % usage:
    % [xx,tt] = syn_sin(fk, Xk, fs, dur, tstart)
```

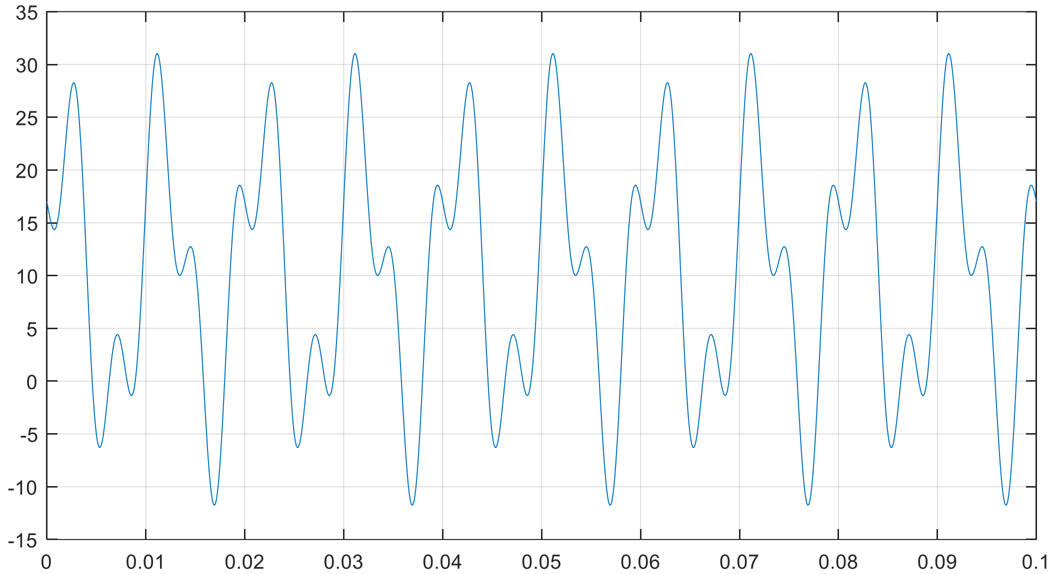
```
    % fk = vector of frequencies
    % (these could be negative or positive)
    % Xk = vector of complex amplitudes: Amp*eˆ(j*phase)
    % fs = the number of samples per second for the time axis
    % dur = total time duration of the signal
    % tstart = starting time (default is zero, if you make this input optional)
    % xx = vector of sinusoidal values
    % tt = vector of times, for the time axis
    % Note: fk and Xk must be the same length.
    % Xk(1) corresponds to frequency fk(1),
    % Xk(2) corresponds to frequency fk(2), etc.

    tt = tstart:1 / fs:dur;
    xx = 0;

    if not(eq(length(fk), length(Xk)))
        error('Length of fk and Xk must be equal.');
    end

    for iter = 1:length(fk)
        x_iter = Xk(iter);
        f_iter = fk(iter);
        xx = xx + real(x_iter * exp(2 * pi * 1i * f_iter * tt));
    end

    plot(tt, xx)
end
```

In order to use this M-file to synthesize harmonic waveforms, you must choose the entries in the frequency vector tobbe integer multiples of some desired fundamental frequency.

```
[xx0, tt0] = syn_sin([0, 100, 250], [10, 14*exp(-j*pi/3), 8*j], 10000, 0.1, 0);
%-Period =?
```

Measure the period of xx0 by hand. Then compare the period of xx0 to the periods of the three sinusoids that make up xx0 and write an explanation on the verification sheet of why the period of xx0 is longer.

The period of xx0 is larger than the signals which add up to it; the period of signal with f = 100 Hz is 10 ms, with f = 250 Hz is 4 ms and the resulting signal has a period which is the LCM of the components' fundamental period which is 20m seconds (as is verifiable by the output plot)

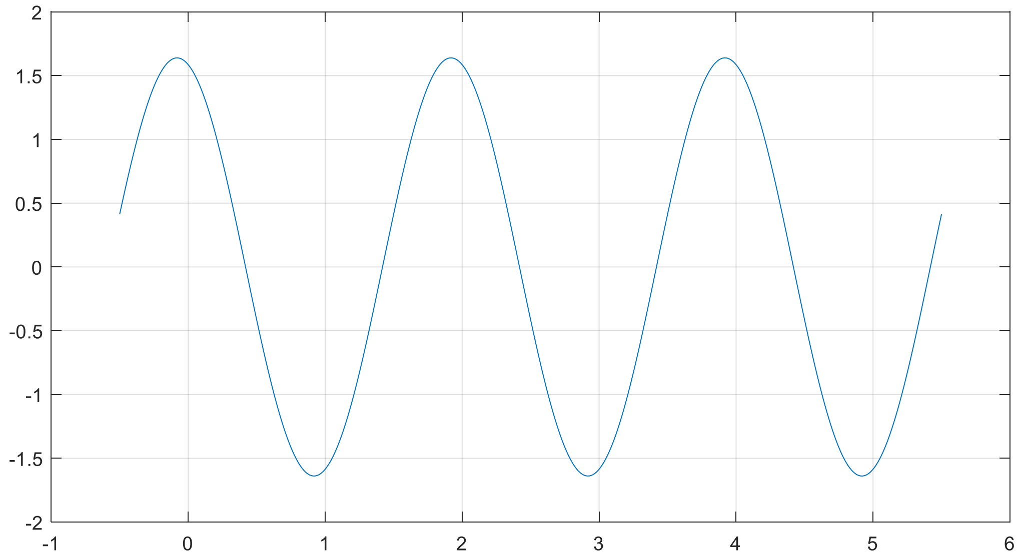


4.3 Representation of Sinusoids with Complex Exponentials

Generate the signal $x(t) = \Re e\{2e^{j\pi t} + 2e^{j\pi(t-1.25)} + (1-j)e^{j\pi t}\}$ and make a plot versus t. Use the syn sin function and take a range for t that will cover three periods starting at t = −0.5 secs. *Include the* MATLAB *code with your report.*

→ f = 1/2 Hz → T = 2 s → tstart = - 0.5 → range for three periods = 2 s * 3 = 6 s → dur = - 0.5 + 6 = 5.5 s

```
x1 = 2; x2 = 2*exp(j*pi*(-1.25)); x3 = (1-j); (at t=0)
[xx0, tt0] = syn_sin([1/2, 1/2, 1/2], [x1, x2, x3], 10000, 5.5, -0.5)
```



5 Conclusion

In this lab, we further familiarized ourselves with MATLAB, and we learned how to define and manipulate complex numbers. We used the exponential notation of the sinusoid family to both define and plot a cosine wave. We also synthesized a sinusoidal wave from multiple complex exponentials and verified that the time period of the resulting signal is the longer than the LCM of the frequencies of the fundamental components.