**Name: Sangalang, Jamal Meldrick G.**

**Section: CpE-401**

**Date Assigned: 27/09/2023**

**Date Submitted:**

**EXPERIMENT 6**

**DISCRETE FOURIER TRANSFORM (DFT)**

**Z-TRANSFORM**

**OBJECTIVES**

1. To learn how to evaluate and plot the DFT.
2. To be able to write a MATLAB program that will compute and display the poles and zeros of a Z-transform.

**MATERIALS AND EQUIPMENT**

Computer with installed MATLAB 2013 or higher

**INTRODUCTION**

I. The discrete Fourier transform (DFT), *X*[*k*], of a finite-length sequence *x*[*n*] can be computed using MATLAB. The length of *X*[*k*] is the same as that of *x*[*n*]. The algorithm for *N*-point DFT of a finite-length sequence *x*[*n*], defined for 0 *≤ n ≤ N −* 1, is given by

II. The *z-transform X*(*z*) of a sequence x[*n*] is defined as

and expressed in the form of a ratio of polynomials in or in factored form. Some of the operations that are of interest in practice are as follows: (1) Evaluate the *z*-transform X(*z*) on the unit circle; (2) develop the pole-zero plot of X(*z*); and (3) determine the inverse *z*-transform x[*n*] of X(*z*).

**PROCEDURES**

**DFT**

**STEP 1:** Run the following program that obtains the DFT of a finite sequence x[n]. Use x[n]={1,1,2,3,3}.

NOTE: The MATLAB ***input*** command prompts for user input. Upon running the program, go to the command window to enter your data.

% Program Exp6\_1

%Computing the DFT using the for loop

clf;

close all;

clear all;

N=input('How many point DFT do you want?'); % 5 for x[n]=[1 1 2 3 3]

x2=input('Enter the sequence='); % input using the format [1 1 2 3 3]

for k=1:N %this is the outer loop

for n=1:N %this is the inner loop

w=exp((-2\*pi\*i\*(k-1)\*(n-1))/N);

x(n)=w;

end

c(k,:)=x;

end

r=[c]\*[x2']

%plotting magnitude and angle

subplot(2 1 1)

stem(abs(r),'filled','-r','LineWidth',2); %plot of the magnitude

title('DFT-magnitude');

grid;

subplot(2 1 2)

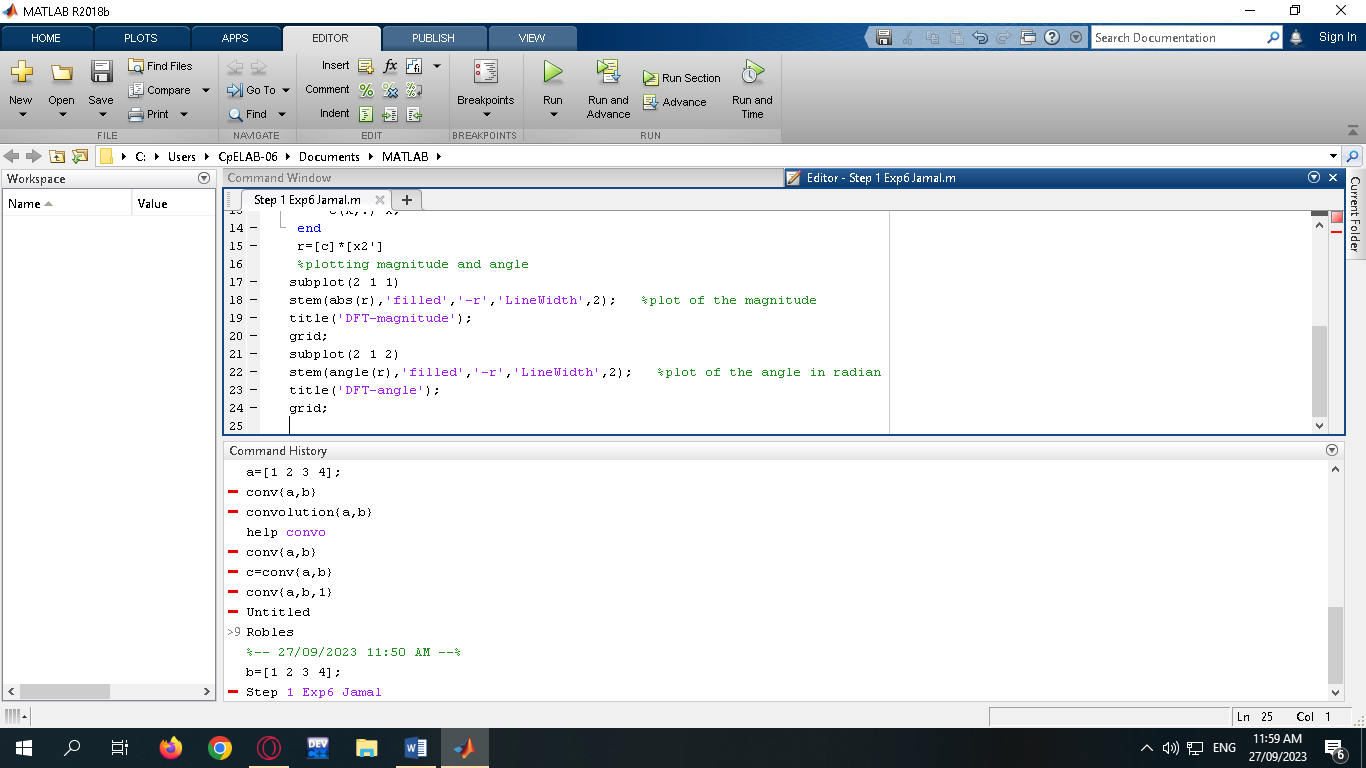
stem(angle(r),'filled','-r','LineWidth',2); %plot of the angle in radian

title('DFT-angle');

grid;

Write the DFT coefficients displayed on the command window (rectangular form).

**r = [10.0000 + 0i; -1.8090 + 2.4899i; -0.6910 + 0.2245i; - 0.2245i; -0.6910 - 0.2245i; -1.8090 - 2.4899i]**



On the space provided, write the magnitude and angle (in radian) for each of the DFTs obtained above. The result must agree with the values on the displayed Figure.

**10∠0°**

**3.077674936 ∠ 2.199110063°**

**0.7265543682 ∠ 2.827458927°**

**0.7265543682 ∠ -2.827458927°**

**3.077674936∠ - 2.199110063°**

Answer the following. From the given program,

What is the name of the variable that holds the input [1 1 2 3 3]? **x2**

Which command computes the exponential part of the algorithm? Write the entire command.

**w = exp((-2\*pi\*i\*(k-1)\*(n-1))/N)**

Which variable contains all the results done in the inner loop (values of the exponential part)? **x[n]**

Which variable contains the DFTs? **r**

What does the command **r=[c]\*[x2']** do? **The matrix c and the input x2 is multiplied.**

Can we remove the symbol (' ) on the command r=[c]\*[x2'] without causing any error?

**No, because it is important in the command, it cause error if we remove the symbol (‘).**

What does the symbol do? **Matrix transpose**

What is the purpose of the MATLAB command **abs**? **It then returns an array Y, where each element in the array represents the absolute value of the corresponding element in the input array X.**

What is the purpose of the MATLAB command **angle? it gives the phase angles for every component of complex array Z, expressed in radians.**

**STEP 2**: The DFT coefficients may also be computed using the MATLAB command **fft**. At this point, x2 contains [1 1 2 3 3]. On the command window type fft(x2) . Did you get the same DFT coefficients? **Yes**

**II. Analysis of Z-Transform**

The function ***tf2zpk*** can be used to determine the zeros and poles of a rational *z*-transform *G*(*z*) . The program statement to use is **[z, p, k] = tf2zpk(num,den)** where num and den are row vectors containing the coefficients of the numerator and denominator polynomials of *G*(*z*). The output file contains the gain constant k and the computed zeros, z, and poles, p. The pole zero plot is generated by the command **zplane(z,p)**.

The reverse process of converting a *z*-transform given in the form of zeros, poles, and the gain constant to a rational form can be implemented using the function zp2tf. The program statement to use is **[num,den] = zp2tf(z,p,k)**.

**STEP 3:**  Write a MATLAB code that will compute and plot the values of the poles and zeros of G(z), H(z), and Q(z). Be sure not to terminate the tf2zpk command line with a semicolon so that you can view the result on the command window.

What are the poles and zeros of the given G(z), H(z), and Q(z)?

**1)**

p = -2.0

2.0

0.50

z = 1.0 + 0.0i

-0.0 + 1.0i

-0.0 – 1.0i

K = 1

**2)**

p = 0.50 + 0.50i

0.50 – 0.50i

z = 0

-1

k = 1

**3)**

p = 0.50 + 0.50i

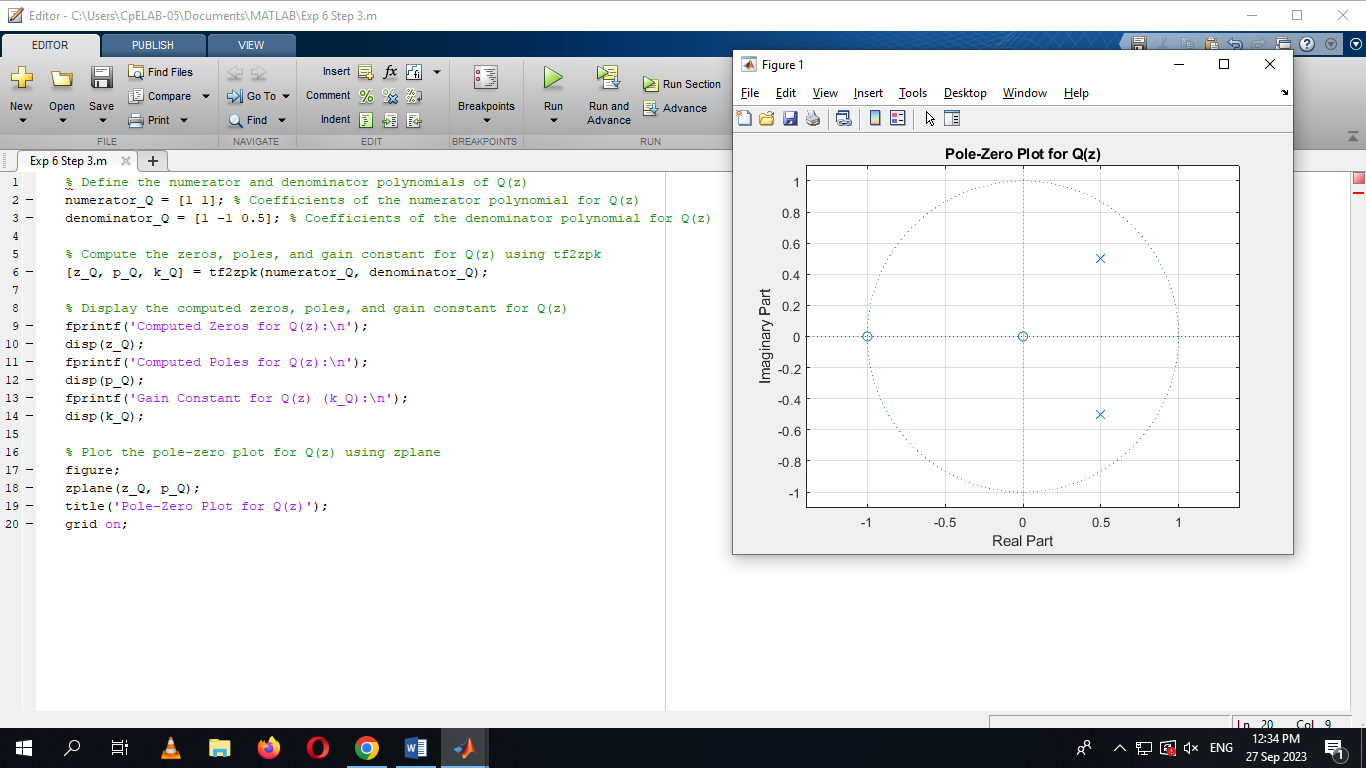
0.50 – 0.50i

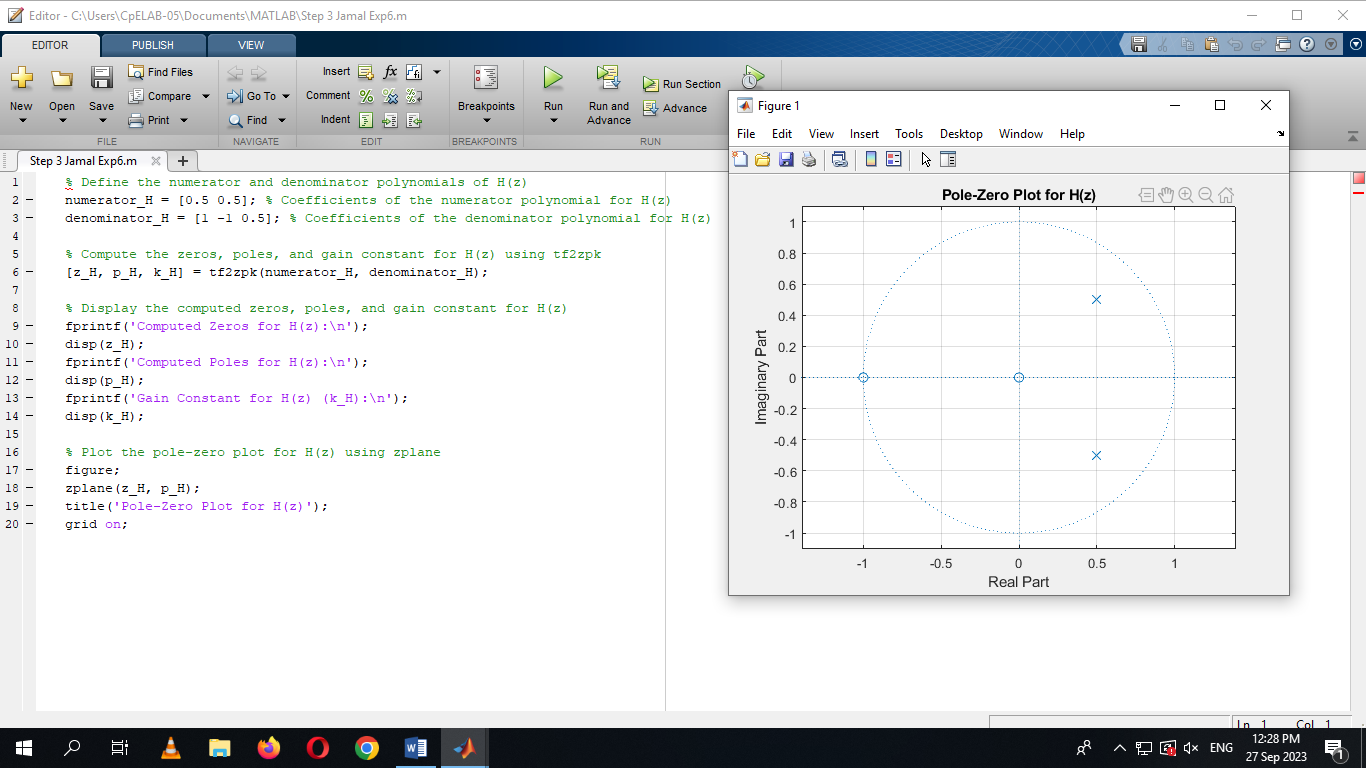
z = 0

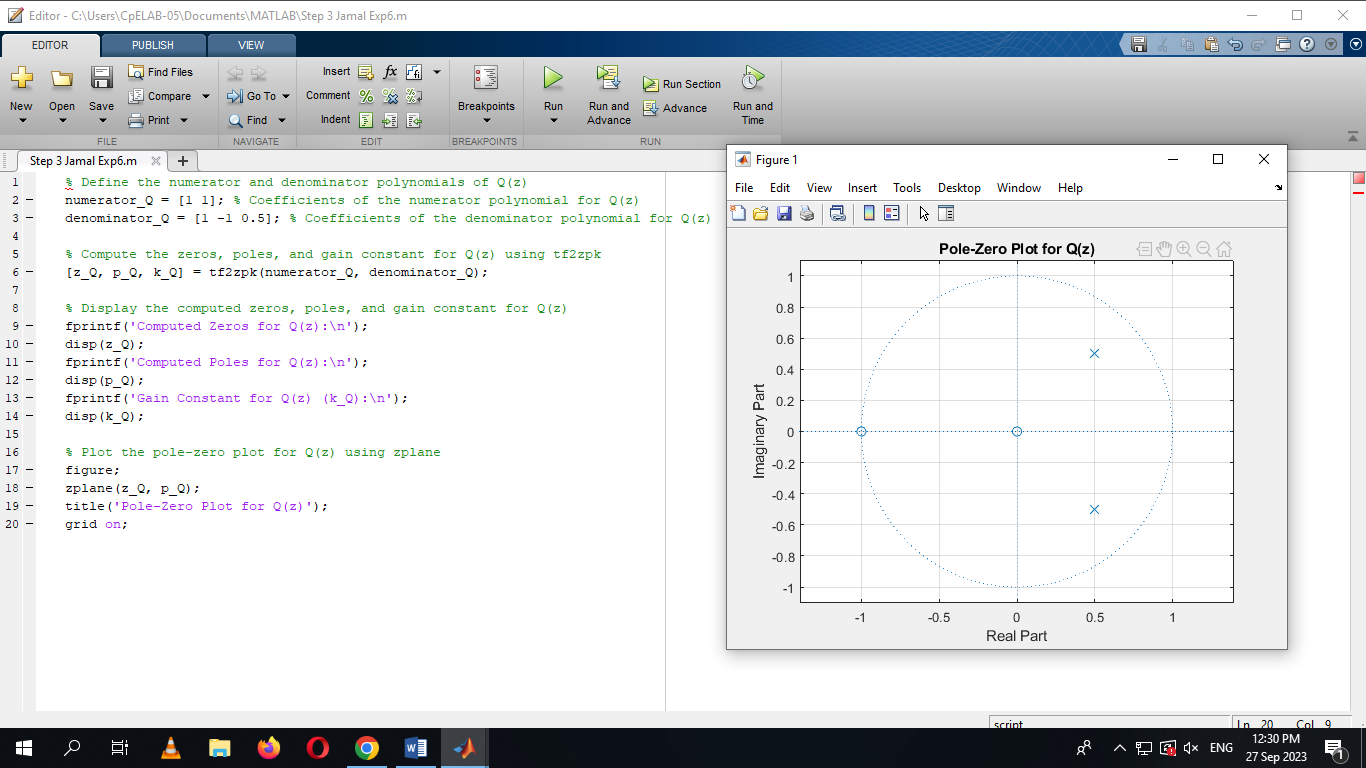
-1

k = 1

Attach the screenshot of the code and plot showing the poles and zeros for each transfer function:







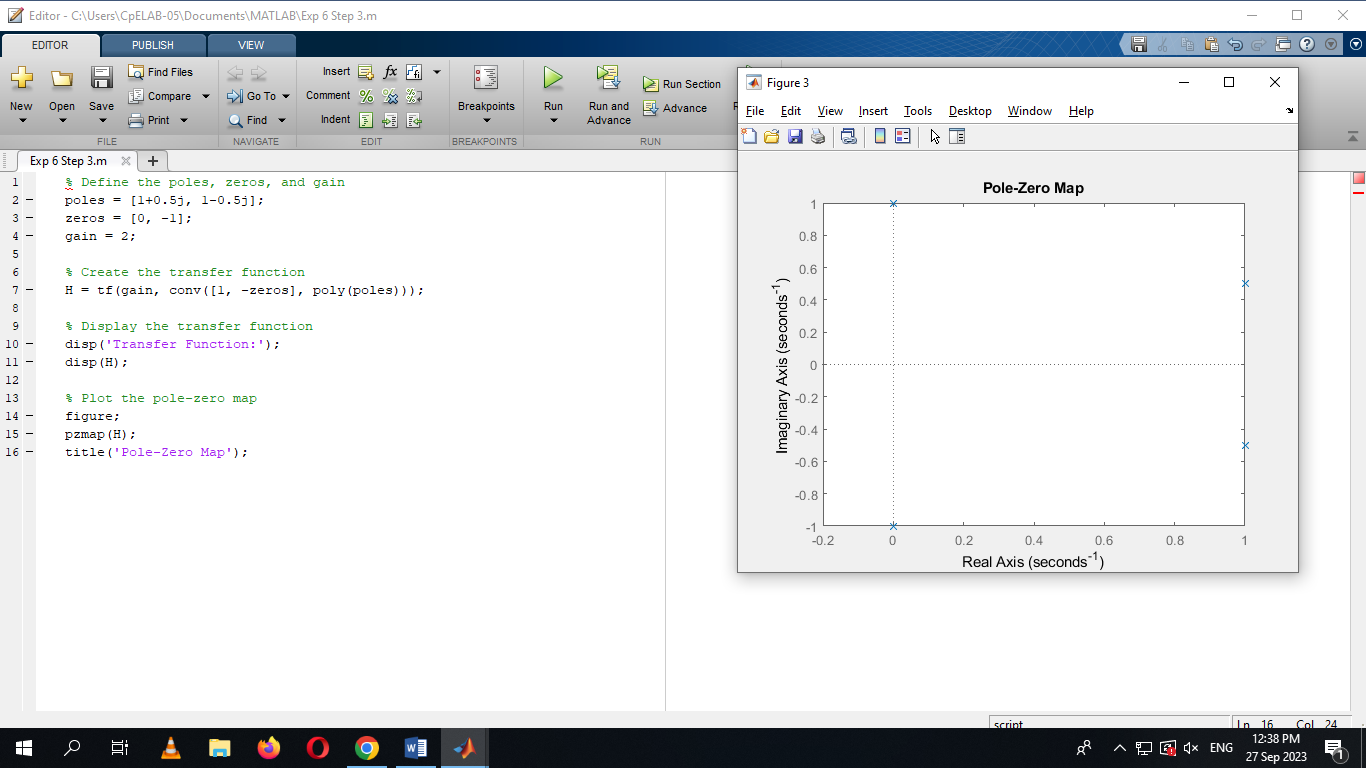
**STEP 4:**  Write a MATLAB code that will compute the transfer function given the poles,zeros, and gain.

Poles: z=1+0.5j , z=1-0.5j

Zeros: z=0, z=-1

Gain=2

Attach the code with the result below:



Write the equivalent transfer function below

**REMARKS AND CONCLUSION:**