Tecnológico de Monterrey **Campus Ciudad de México** **School of Engineering and Sciences** **Department of Mechatronics**	**TE2019 Digital Signal Processing Laboratory** Student's 1 name and ID: Alfredo Zhu Chen - A01651980 Student's 2 name and ID: Luis Arturo Dan Fong - A016506720 Student's 3 name and ID: Miguel Alejandro Cruz Banda - A01656527 Lab # 8 Fourier Analysis of discrete-time signals Date: 14/05/2020

1 INTRODUCTION

The Fourier analysis is named after Jean Baptiste Joseph Fourier, a French mathematician and physicist. One of Fourier's biggest contributions about Fourier analysis can be broken into four principal categories according to their characteristics, such as them being periodic or aperiodic and continuous or discrete. The combination of two of those characteristics give as a result the following categories [1].

- Aperiodic-continuous: These types of signals extend to both -∞ and ∞ without repeating a periodic pattern in the continuous time domain. The Fourier analysis for this category is known simply as the Fourier Transform(FT) [1].
- Periodic-continuous: This segment includes any waveform which repeats itself in a regular pattern from -∞ to ∞ in the continuous time domain. The Fourier analysis for this category is known as Fourier series (FS) [1].
- Aperiodic-discrete: These types of signals are only defined at discrete points between -∞ to ∞ and do not repeat itself in a periodic way. The Fourier analysis for this category is known as Discrete-time Fourier Transform (DTFT) [1].
- Periodic-discrete: These signals repeat themselves in a periodic fashion from -∞ to ∞. The Fourier analysis for this category is known as Discrete-time Fourier Series (DTFS) [1].

Fourier analysis is one of the most useful tools in signal processing. It is based on the decomposition of a signal in terms of a set of base functions (sinusoids of different frequencies). For discrete signals that are of interest in this laboratory session, the DTFS can be used for periodic signals, while the DTFT is used for aperiodic signals. When a discrete-time system is linear and time-invariant (LTI system), the DTFT is the only representation that stands out as the most useful. Is based on the set of complex exponential signals $e^{j\omega n}$.
The DTFT of a signal x[n] is given by equation (1) [2].

$$X[e^{j\omega}] \equiv F[x[n]] = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \tag{1}$$

It is rare to find purely periodic signals in digital signal processing. Non-periodic signals and data with a finite number of values is the most commonly used environment in practical LTI systems.Since the DTFT gives as result a continuum spectrum of frequency, the Direct Fourier Transform (DFT) of a signal x[n] is more practical to implement as it takes equally-spaced samples of the DTFT. The DFT of a signal defined in the range $0 \le n \le N - 1$, is defined as in equation (2)[2].

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N} \tag{2}$$

This equation allows us to transform a math function into another, obtaining a representation in the frequency domain, the original function being a function in the time domain. In particular, DFT is commonly used in digital signal processing and other related fields dedicated to analyzing the frequencies contained in a sampled signal, also to solve partial differential equations, and to carry out operations such as convolutions or multiplications of large integers[3].

OBJECTIVES

1. To define the DTFT, DFT and FFTand describe their similarities and differences
2. To apply the DTFT, DFT and FFT to the frequency analysis of discrete-time signals

2 MATERIALS AND METHODS

The materials used in this laboratory were:

- 1 PC or laptop with at least 4GB RAM and and 5kB(approximate size of five scripts) of space available
- MATLAB R2019a or newer versions

2.1 *Experiment 1*

In order to compute and plot the magnitude and phase of the DTFT $X(\omega)$ of the signal (3), using $0 \leq n \leq 10$ over the frequency intervals $-\pi \leq \omega \leq \pi$ and $-3\pi \leq \omega \leq 3\pi$.

$$x[n] = cos(\frac{\pi n}{3}) \tag{3}$$

Firstly it was necessary to use a symbolic variable *w*, so it is defined using MATLAB as *syms w* in order for it to be considered a continuous variable, as the spectrum in frequency given by DTFT is continuous.The $n$interval was defined in the interval $0 \leq n \leq 10$, while the signal was defined using the variable x. To calculate the DTFT of the signal (3) with (1), it was necessary to use the MATLAB functions $sum(A)$and *exp(x)* as in (4), considering the asterisks as a multiplication operation and not a convolution operation in the code.

$$X = sum(x.* exp(-j * w * n)) \tag{4}$$

Once all the variables were declared, $subplot(m, n, p)$, where it divides the figure in *m* by *n* grid and creates the plot in position *p*,and $ezplot(fun, [xmin, xmax])$, where the function is plotted within the interval *xmin* and *xmax* in the x axis, were used for plotting the DTFT graphs for both magnitude and phase in the same figure. This procedure was performed two times, one for each frequency interval($-\pi \leq \omega \leq \pi$ and $-3\pi \leq \omega \leq 3\pi$) and giving as result two figures, each having their own magnitude and phase according to their defined frequency intervals. The magnitude was computed using MATLAB function *abs(x)* and the phase was generated using *angle(x).*

2.2 *Experiment 2*

The experiment consist of calculating and plotting the magnitude and phase of the DFT X_k of the signal (3), using $0 \leq n \leq 10$. At first it is required to define the signal and its interval as in experiment one.To calculate the DFT of a signal, a function was firstly in a new script named as *dft(x)*, which it takes the signal *x* as the only parameter. The function determines the length *N* of the signal with the MATLAB function *length(x)*. Then it iterates with a *for* loop over *n* from 0 to *N-1* in order to calculate the vector of values before applying sum operation in (2) at the index k and this is iterated inside another *for* loop over *k* from 0 to *N-1* to sum up each vector for each *k*, the result of this would be the DFT of the signal *x*.

Once the *dft(x)* function was written, the DFT of the signal (3) was stored in the variable Xk in the MATLAB script of this experiment as in (5). Another variable *Xkshift* was used to store the DFT of the signal applied with MATLAB function $fftshift(X)$ for rearranging the Fourier Transform by shifting the zero-frequency component to the center of the array as in (6). As the zero-frequency is rearranged to the center of the signal vector, the x-axis to plot the signal was modified to shift it symmetrically by the center of 0. Furthermore, an implementation of DFT, MATLAB function $fft(x)$, was used and stored in the variable Xk_fft as in (7). Since the DFT gives a discrete data sequence, the magnitude and phase of these three implementations were plotted in the same figure with MATLAB function *stem(x,y)* and *subplot(m,n,p).*

$$Xk = dft(x) \tag{5}$$

$$Xkshift = fftshift(Xk) \tag{6}$$

$$Xk\_fft = fft(x) \tag{7}$$

2.3 *Experiment 3*

In this part of the procedure, the function $Xk = dft(x)$ was used to compute the DFT of a sequence x[n]. The discrete-time signal that the DFT is computed to is shown in the equation (8). The procedure is similarly as in experiment 2, the only two things that changed compared to experiment 2, apart from the variable defined as x and the interval called n using $0\leq n\leq 3$, were that a second parameter n in the MATLAB function *fft(x,n)* was added which represents the transform length.

$$x[n]=[1,-2,2,1] \tag{8}$$

2.4 *Experiment 4*

This experiment consist of plotting in the same graph the DTFT and DFT of the signal (9) in the interval $0\leq\omega\leq 2\pi$. First, the signal was defined in MATLAB with the interval $0\leq n\leq 19$, its DTFT was generated as in experiment 1 using (4) and its DFT by using the implementation of *dft(x)*. To plot both in the same figure, MATLAB *hold on* command was used to hold the graphs in the same figure and *hold off* was used to stop holding the graphs. The magnitude and phase for were plotted separately in two figures.

$$x[n]=0.7^{\,n}u[n] \tag{9}$$

3 RESULTS AND DISCUSSIONS

3.1 *Experiment 1*

As it can be seen in figure 1, the magnitude and phase of the DTFT of the signal (3) are shown. Although the signal (3) is discrete and limited in time, the DTFT gives as result the spectrum of frequency in a continuous range of frequency from $-\pi$ to π. Since the signal is a cosine function with frequency $\pi/3$, it is expected that the the spectrum in frequency of the magnitude has two remarkable pikes at $-\pi/3$ and $\pi/3$. Additionally, it is possible to see that graph of the magnitude is symmetrically to y-axis, since the negative frequencies can be reflected to the positive values, it is possible to say that the information of the magnitude is sufficient by only obtaining one side of the frequency, this might be commonly seen in some spectrum analyzers. The magnitude of the spectrum in frequency is an even function, on the contrary, the phase of the spectrum in frequency is an odd function as it is possible to see in the figure.

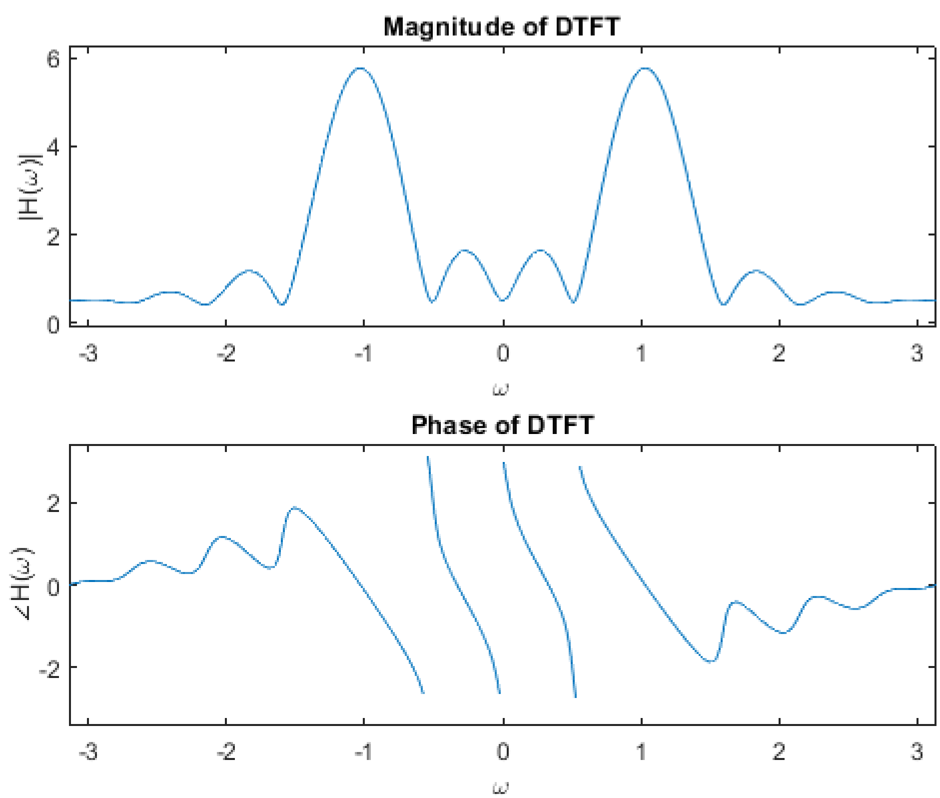

Figure 1. DTFT of signal (3) with in $-\pi\leq n\leq\pi$

The following figure 2 shows the DTFT of the same signal (3), the difference is that this time is contained in a more extended interval from -3π to 3π. Comparing this figure with the previous one, it can be seen that the magnitude and phase of the spectrum in frequency is repeated 3 times as it is now extended to 3 periods. From this information, it is possible to state that the spectrum is periodic, with period of 2π and so it is sufficient to analyse only one period. Taking the advantage that the magnitude of the spectrum is an even function, analysing half of the period is also possible to know all the information of the spectrum.

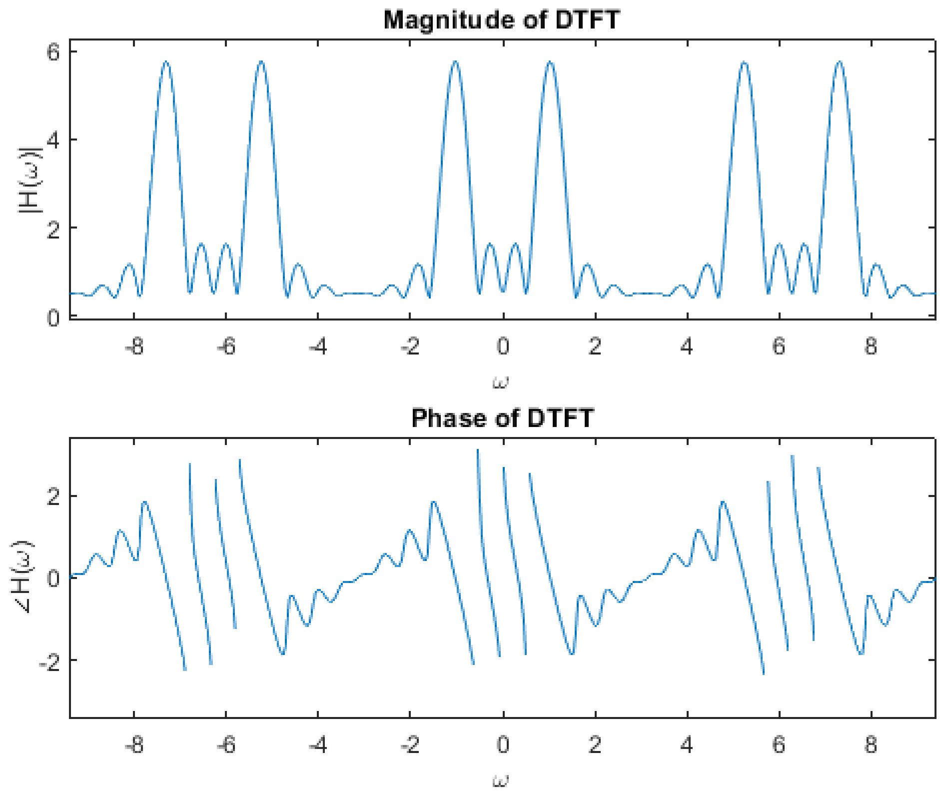

Figure 2. DTFT of signal (3) within $-3\pi \leq n \leq 3\pi$

3.2 *Experiment 2*

As it can be seen from figure 3, the magnitude and phase of the DFT of the signal (3) in different forms are shown.The pair of graphs, from top to bottom, correspond to the result of DFT by implementation of scratch using dft(x), MATLAB function *fft(x)*, and using *ffshift(x)* after *dft(x)*. Comparing the two graphs from the top and two at the center, the two pairs are identical, so this can verify that our implementation of *dft(x)* to calculate the DFT is correct. The pair at the bottom show the magnitude and phase of the spectrum applied with *fftshift(x)*, the main idea of this is to centralize the spectrum at zero. As it was said in 3.1, the spectrum is periodic, at least in the perspective from the property of circular shift in time and frequency of the DFT, so the information repeats after every period; in this case, the right half of the spectrum data vector is now passed to the negative part frequency ,so the same information is contained and it is another way to present the spectrum.Furthermore, it is also possible to see from the figure 3 that the DFT gives as result a discrete spectrum and it maintains the same properties like the DTFT, such as the symmetry and the periodicity.   .

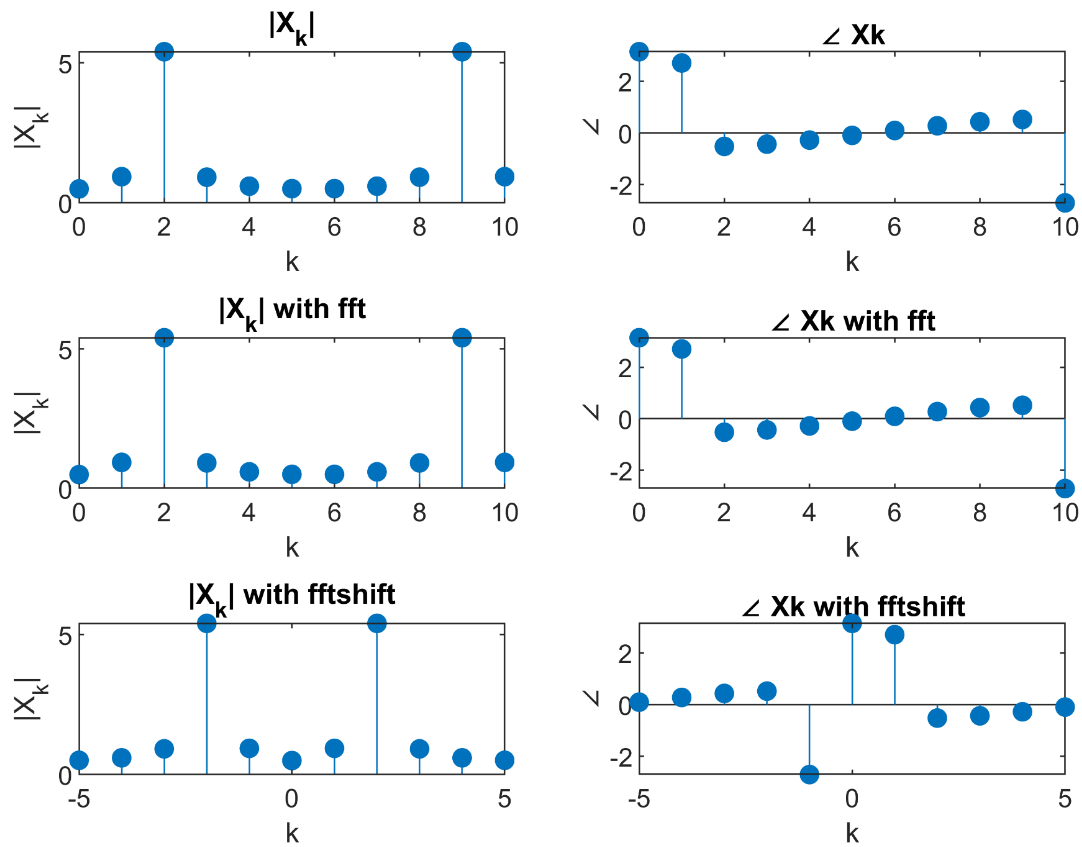

Figure 3. DFT of signal (3)

3.3 *Experiment 3*

The next figure 4 shows the magnitude and phase of the spectrum by applying the DFT to the discrete time data sequence (8). As it can be seen, a discrete spectrum is given and it represents the content in frequency of the signal. According to the two pairs of graphs of the figure, since there are only four samples in the signal, only four points of the spectrum are obtained. Having few samples of the signal can be a problem, as the result of the DFT would be also a few points in the spectrum and the frequency content of the signal would be hard to interpret. To tackle this problem, it is possible to indicate in the *fft(x, n)* function as a second parameter *n* of the transform length to generate more points than it generates by default, *n* is commonly specified as a power of 2 or a value that can be factored into a product of prime numbers to increase performance of computation . In this case, from the pair of graphs at the bottom, 32 points are generated and it is possible to obtain more information about the frequency content than by having only four points.

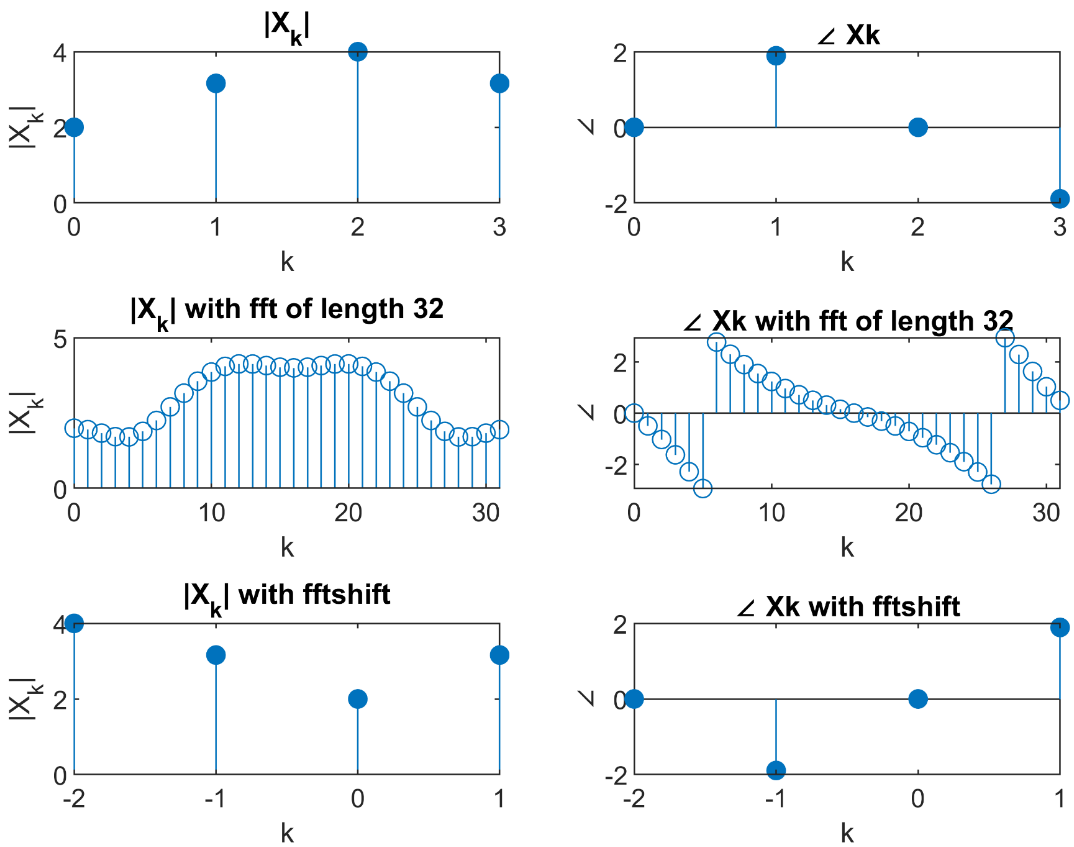

Figure 4. DFT of signal (8)

3.4 *Experiment 4*

From the figure 5, the results of the DTFT and the DFT are compared in the magnitude and phase of the spectrum. It is important to notice that the DTFT gives as results a continuous spectrum of the signal, while the DFT gives as result a discrete spectrum. Both give the same information about the spectrum of the signal. The DTFT uses symbolic function to represent the spectrum, this can be a problem because it is not possible to implement in some devices, so the DFT is a good alternative as it discretizes the DTFT.

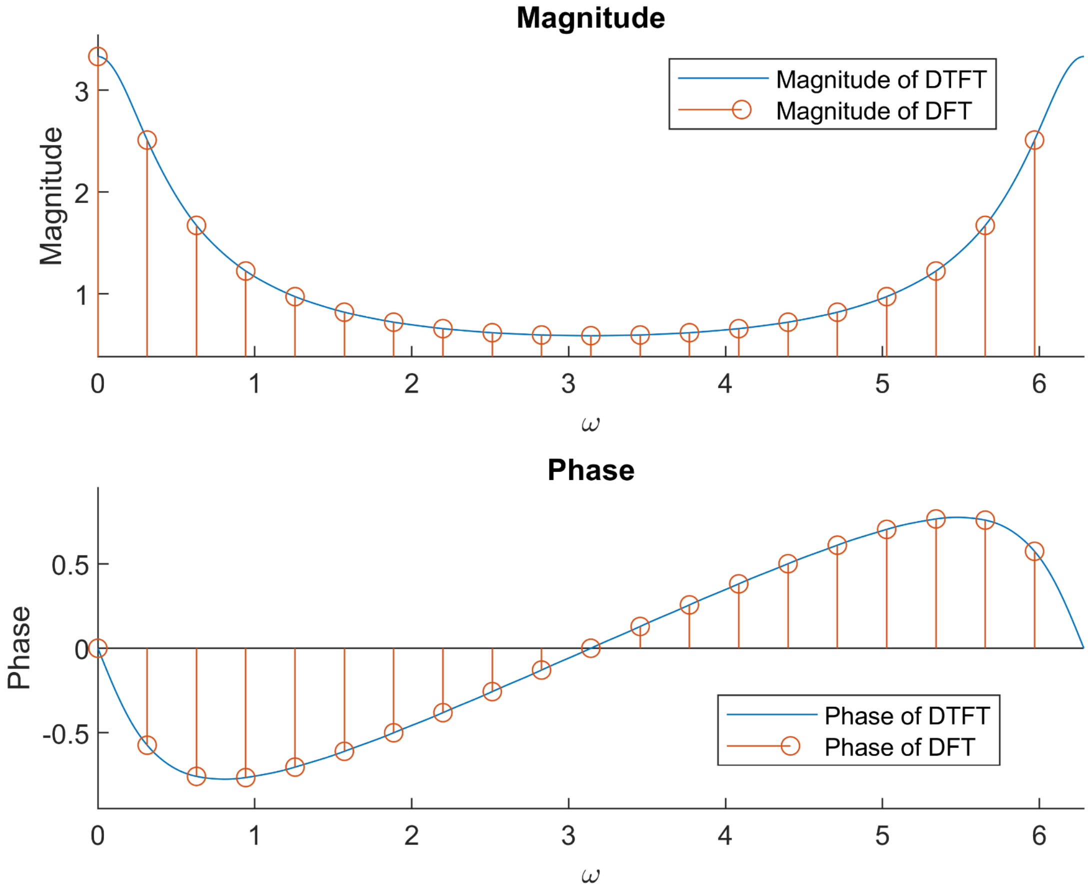

Figure 5. DTFT and DFT of the signal (9)

4 CONCLUSION

In conclusion, performing different analysis for the DFT and DTFT allows for a better comprehension of the differences and similarities between each of them. Likewise making a graphical analysis by plotting each of them provides a clear and better understanding of how signals behave in the frequency domain for both DFT and DTFT, by decomposing their magnitude and phase. From the results, it is possible to conclude that DTFT is a powerful tool to analyse a discrete aperiodic signal in the frequency domain, while the DFT is a very practical tool to obtain the frequency spectrum of the signal in a discrete form.

5 REFERENCES

[1] Smith, S. W. (1997). The scientist and engineer's guide to digital signal processing (Vol. 14, p. 626). San Diego: California Technical Pub.
[2] Stanley, W. D., Dougherty, G. R., Dougherty, R., & Saunders, H. (1988). Digital signal processing.
[3] Oppenheim, A. V. (1978). Applications of digital signal processing. *Englewood Cliffs*.

6 APPENDIX

1. Experiment 1

```
%Clear environment
clc
clear all
close all

syms w
n = 0:10; %interval
x = cos(n*pi/3); %defining signal x
X = sum(x.*exp(-j*w*n)); %calculating the DTFT of x

figure
subplot(2,1,1)
ezplot(abs(X),[-pi pi])  %DTFT magnitude form -pi to pi
title('Magnitude of DTFT')
xlabel('\omega')
ylabel('|H(\omega)|')
subplot(2,1,2)
ezplot(angle(X),[-pi pi]) %DTFT phase from -pi to pi
title('Phase of DTFT')
xlabel('\omega')
ylabel('\angleH(\omega)')

figure
subplot(2,1,1)
ezplot(abs(X),[-3*pi 3*pi]) %DTFT magnitude form -3pi to 3pi
title('Magnitude of DTFT')
xlabel('\omega')
ylabel('|H(\omega)|')
subplot(2,1,2)
ezplot(angle(X),[-3*pi 3*pi]) %DTFT phase from -3pi to 3pi
title('Phase of DTFT')
xlabel('\omega')
ylabel('\angleH(\omega)')
```

2. Experiment 2

```
%Clear environment
clc
clear all
close all

n = 0:10; %interval
x =cos(n*pi/3); %signal
Xk=dft(x); %DFT of x
Xkshift=fftshift(Xk); %Centralize zero-frequency
Xk_fft=fft(x); %DFT of x with fft

N=length(Xk); %size of DFT

figure
subplot(3,2,1)
stem(0:N-1,abs(Xk),'filled'); %DFT magnitude
title ('|X_k|')
ylabel("|X_k|")
xlabel("k")
subplot(3,2,2)
stem(0:N-1,angle(Xk),'filled');  %DFT phase
```

```
title ('\angle Xk')
ylabel("\angle")
xlabel("k")
subplot(3,2,3) %DFT
stem(0:N-1,abs(Xk_fft),'filled'); %DFT magniude with fft
title ('|X_k| with fft')
ylabel("|X_k|")
xlabel("k")
subplot(3,2,4)
stem(0:N-1,angle(Xk_fft),'filled');  %DFT phase with fft
title ('\angle Xk with fft')
ylabel("\angle")
xlabel("k")
subplot(3,2,5)
stem(-(N-1)/2:(N-1)/2,abs(Xkshift),'filled'); %DFT magnitude with fftshift
title ('|X_k| with fftshift')
ylabel("|X_k|")
xlabel("k")
subplot(3,2,6)
stem(-(N-1)/2:(N-1)/2,angle(Xkshift),'filled');%DFT phase with fftshift
title ('\angle Xk with fftshift')
ylabel("\angle")
xlabel("k")
```

3. Experiment 3

```
%Clear environment
clc
clear all
close all

n = 0:3; %interval
x =[1,-2,2,1]; %signal
Xk=dft(x); %DFT
Xk_fft32=fft(x,32); %DFT with fft of 32 length
Xkshift=fftshift(Xk); %DFT with fftshift
N=length(Xk); %length of DFT
N_fft=length(Xk_fft32); %fft 32 length
figure
subplot(3,2,1)
stem(0:N-1,abs(Xk),'filled'); %DFT magnitude
title ('|X_k|')
ylabel("|X_k|")
xlabel("k")
subplot(3,2,2)
stem(0:N-1,angle(Xk),'filled');%DFT phase
title ('\angle Xk')
ylabel("\angle")
xlabel("k")
subplot(3,2,3)
stem(0:N_fft-1,abs(Xk_fft32));
title ('|X_k| with fft of length 32') %fft magnitude with length 32
ylabel("|X_k|")
xlabel("k")
subplot(3,2,4)
stem(0:N_fft-1,angle(Xk_fft32)); %fft phase with length 32
title ('\angle Xk with fft of length 32')
```

```
ylabel("\angle")
xlabel("k")
subplot(3,2,5)
stem(-N/2:N/2-1,abs(Xkshift),'filled'); %DFT magnitude with fftshift
title ('|X_k| with fftshift')
ylabel("|X_k|")
xlabel("k")
subplot(3,2,6)
stem(-N/2:N/2-1,angle(Xkshift),'filled'); %DFT phase with fftshift
title ('\angle Xk with fftshift')
ylabel("\angle")
xlabel("k")
```

4. Experiment 4

```
%Clear environment
clc;
close all;
clear all;

syms w
n = 0:19; %interval
x = (0.7.^n); %signal
X_dtft = sum(x.*exp(-j*w*n)); %DTFT of signal x

X_fft=dft(x); %DFT with fft of signal x
k=n;
N=length(k);

figure
subplot(2,1,1)
hold on
ezplot(abs(X_dtft),[0 2*pi]); %DTFT magnitude
stem(2*pi*k/N,abs(X_fft)); %DTF magnitude
xlabel('\omega')
ylabel('Magnitude')
title('Magnitude')
legend('Magnitude of DTFT','Magnitude of DFT')
hold off

subplot(2,1,2)
hold on
ezplot(angle(X_dtft),[0 2*pi]);%DFT phase
stem(2*pi*k/N,angle(X_fft)); %DFT phases
xlabel('\omega')
ylabel('Phase')
title('Phase')
legend ('Phase of DTFT','Phase of DFT')
hold off
```

5. function dft(x)

```
function Xk=dft(x)
%Arguments:
%x:signal

N=length(x); %length of the signal

%DFT
```

```
for k=0:N-1
    for n=0:N-1
        X(n+1) = x(n+1) * exp(-j*2*pi*k*n/N);
    end
    Xk(k+1) = sum(X);
end
end
```