Lab 11

**Discrete Fourier Transform (DFT)**

**OBJECTIVES**

• To study and implement the discrete Fourier transform

• To learn how to analyze discrete-time signal using DFT

• System analysis using DFT

**INTRODUCTION**

The discrete Fourier transform (DFT) provides a discrete representation in the frequency-domain, *X*[*k*], of a discrete-time signal *x*[*n*]. The DFT *X*[*k*] of a discrete time signal *x*[*n*] is defined as

*X*[*K*] consists of samples every Ω0 of the DTFT *X*(Ω) of the signal *x*[*n*], where Ω0 is the frequency resolution. In general *X*[*k*] is a representation of *X*[*k*Ω0] but for notation simplicity Ω0 is dropped. For example *X*[1] and X[2] represent the strengths and phase shifts of the sinusoidal signals with frequency components Ω0 and 2Ω0 in the discrete signal *x*[*n*]. *X*[0] represents the DC component. *X*[*k*] in general is a complex function of the real variable *k*.

where and are, respectively, the real and imaginary parts of 𝑋[*k*], and are real functions of *k*. 𝑋[*k*] can alternately be expressed in the form

The quantity |𝑋[*k*]| is called the magnitude function and the quantity ∠𝑋[*k*] is called the phase function.

The inverse discrete Fourier transform (IDFT) is the process of getting the discrete time signal *x*[*n*] from its discrete Fourier transform 𝑋[*k*]. The IDFT is defined as follow

The DTFT satisfies several useful properties that are often utilized in several applications. Figure 1 below summarizes the relationship between the Fourier transform (FT) *X*(*ω*) of a continuous time signal *x*(*t*) with the discrete time Fourier transform (DTFT) *X*(Ω) of a discrete time signal *x*[*n*] with the discrete Fourier transform (DFT) of periodic discrete time signal *x*[*n*].

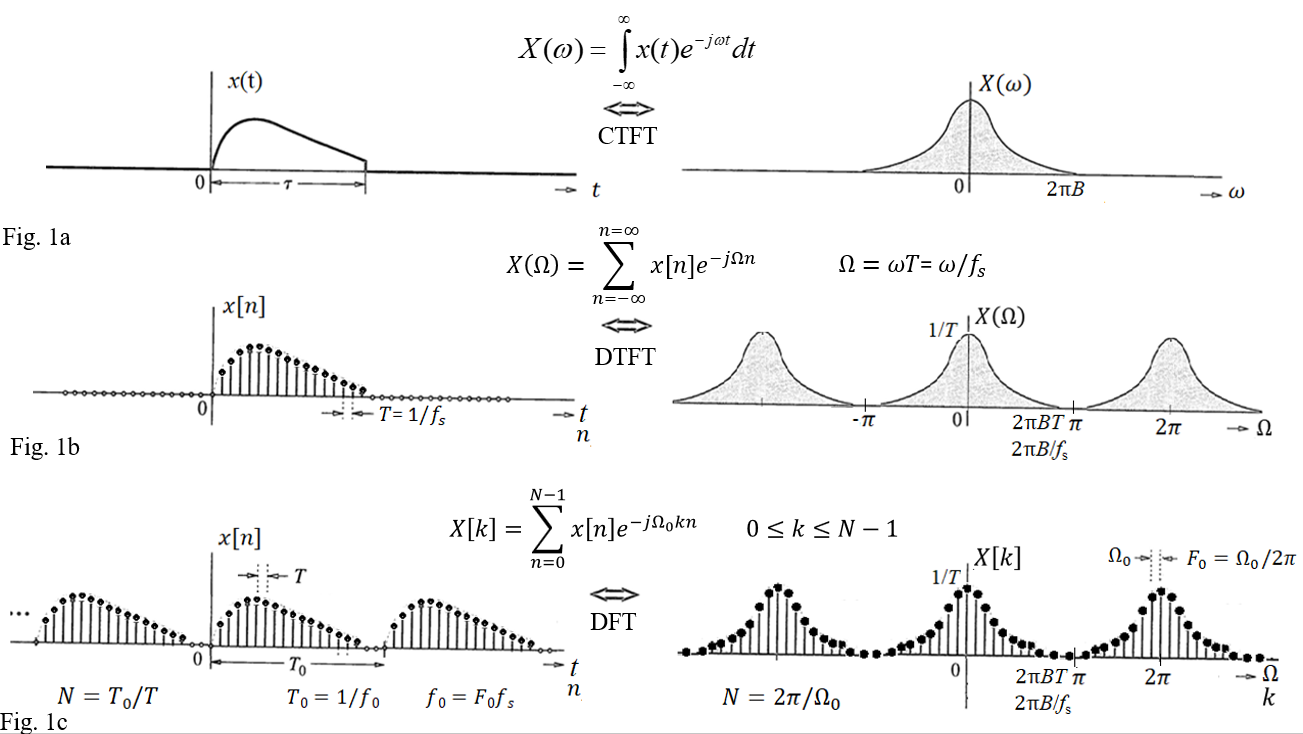


Fig. 1

Figure 1a shows a continuous time signal *x*(*t*) and its continuous spectrum *X*(*ω*) with bandwidth *B*. Figure 1b shows the discrete signal *x*[*n*] and its continuous spectrum *X*(*Ω*) with bandwidth *B*/*f*s. Figure 1c shows the sampled spectrum *X*[*k*] and its discrete time signal *x*[*n*] with periodicity *T*0.

To convert a signal from a continuous to a discrete in the time domain or the frequency domain then the following should be considered:

1. To avoid aliasing when sampling a continuous time signal to a discrete time signal then the sampling rate *f*s should be equal or larger than twice the bandwidth *B*, *f*s ≥ 2*B*.
2. To avoid aliasing when sampling a continuous spectrum to a discrete spectrum then the digital sampling interval *F*0 should be equal or less than the fundamental frequency of the discrete time signal *f*0, *F*0 ≤ *f*0/ *f*s.
3. To see more sample of the continuous spectrum then pads the discrete time signal *x*[*n*] by zeros. For example, if the original number of samples is 10 and we need to see 15 samples of the spectrum then pads the original samples by 5 zeros.

**Prelab Exercises**

For the FIR filters *h*1[*n*] = [1 -2 2 -2 1] and *h*2[*n*] = [1/3 -2/3 1/3] find the following:

1. The DFT of the impulse response *h*1[*n*] and *h*2[*n*] in term of *k*.
2. Use stem plot to plot the magnitude and phase for *H*1[*k*] and *H*2[*k*] versus *k*.

**lab Exercises**

**Task 1: Linear Phase FIR Type 1.**

For the FIR filters *h*1[*n*] = [1 -2 2 -2 1] do the following:

1. Use the fft function in Matlab to find *H*1[*k*] and plot the magnitude and phase. Did you get the same answer as in the prelab?
2. How to improve the frequency resolution Ω0 to 0.2π in *H*1[*k*]. Find and plot *H*1[*k*] with the new frequency resolution 0.2π.
3. What type of filter is *h*1[*n*]?
4. If the sampling rate is 8000 Hz, then the different components of *H*1[*k*] found in part 2 represent what frequencies in Hz?

**Task 2: Finding *X*[*k*] using fft in Matlab**

Consider the signal *x*(*t*) below as the output of a sensor. A one second duration of the signal is sampled with sampling rate 7 samples/second. The 7 samples of the 1 second duration are

*x*[*n*] = [2.7071 3.2431 -0.2023 -0.9326 0.8160 2.6390 -1.2702].

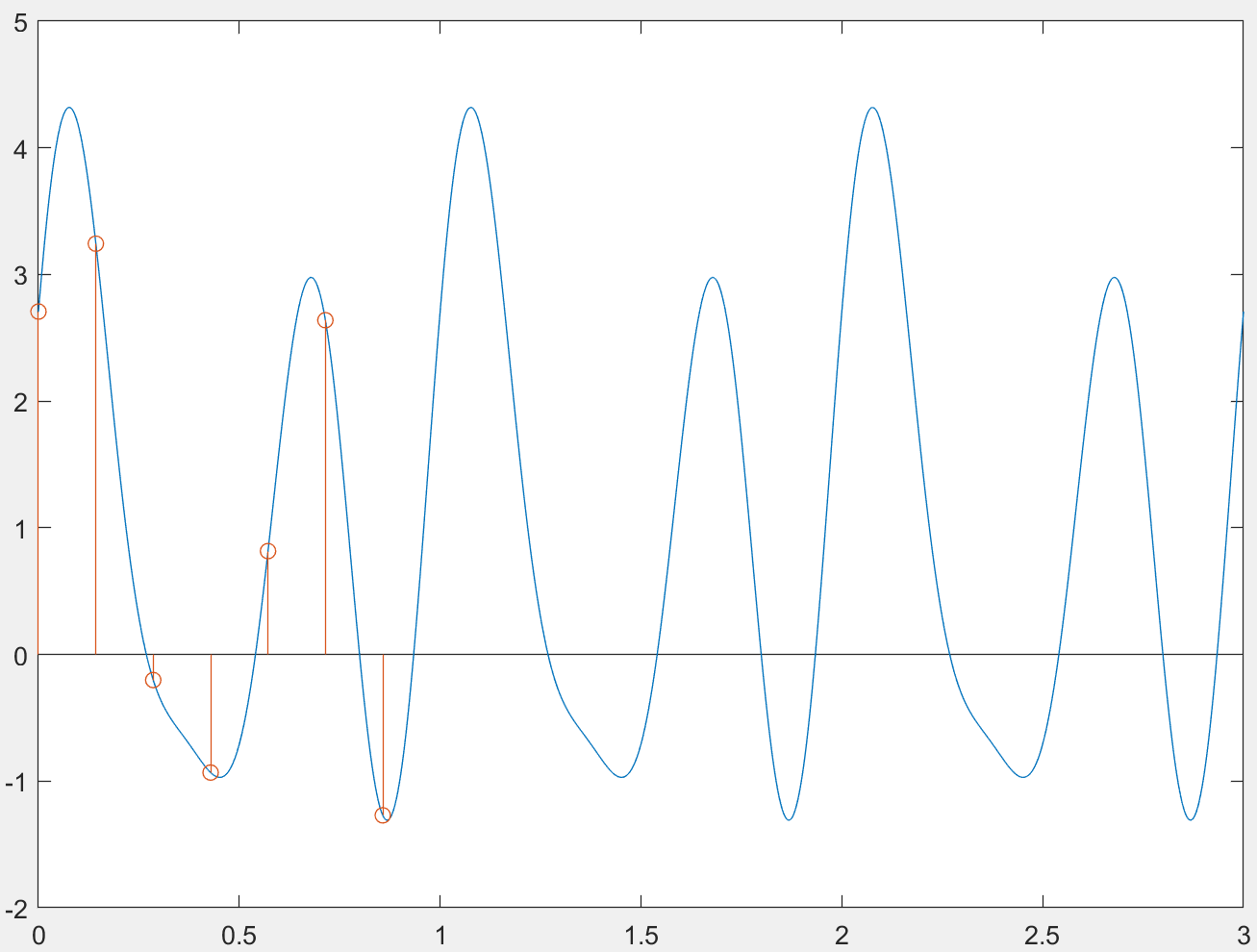


Fig. 2

1. Use Matlab function fft to find the DFT *X*[*k*] of the 7 samples.
2. Based on the frequency content *X*[*k*], express the signal *x*[*n*] and *x*(*t*) as

*x*[*n*] = *A*0 + *A*1cos(Ω1*n* + *θ*1) + *A*2cos(Ω2*n* + *θ*2) + ….+ *A*Ncos(ΩN*n* + *θ*N)

*x*(*t*) = *A*0 + *A*1cos(2π*f*1*t* + *θ*1) + *A*2cos(2π*f*2*t* + *θ*2) + ….+ *A*Ncos(2π*f*N*t* + *θ*N)

***Hint***: According to the inverse DFT equation

*A*0 = *X*[0]/*N* and *A*k = 2(*X*[*k*]/*N*). The analog frequency (ꞷ) is related to the digital frequency (Ω) by

Ω = ꞷT = ꞷ/*f*s = 2π*f* / *f*s .

1. Plot the signal *x*(*t*) for the time interval *t* = 0:0.01:3. Is the plot like the plot in Fig. 2 above?

**Task 3: Filtering Using the DFT Analysis**

The signal *x*[*n*] in task 2 is the input of the filter *h*2[*n*] = [1/3 -2/3 1/3].

1. Use the DFT analysis to find the output *Y*[*k*]. Make sure to pad the signal *x*[*n*] and *h*2[*n*] with right number of zeros to give you the right size of the output signal *Y*[*k*].
2. Use the IDFT of *Y*[*k*] found above to find *y*[*n*].
3. Find *y*[*n*] using linear convolution of *x*[*n*] with *h*2[*n*].

yl = conv(x,h2);

1. Find *y*[*n*] using circular convolution of *x*[*n*] with *h*2[*n*].

yc = cconv(x,h2,ny); where ny is the length of the output signal *y*[*n*].

1. Compare the three outputs found in part 2, 3 and 4 of this task.
2. How did the filter impact the different frequency components in the input signal *x*[*n*]?

**LAB REPORT FORMAT**

Please use D2L to submit the lab report electronically. In case there is a technical issue with D2L, please email the lab report to the course instructor before the cutoff time. The lab report should include: (1) Title Page; (2) Introduction; (3) Results (Include programs, answers, and figures to all steps in Lab Procedure); (4) Conclusion/Discussion.