Lab 9

DFT and IDFT Computations

9.1 Introduction

In the previous lab, we saw that the frequency response of a system is calculated on the basis of Z-transform. The frequency response was a continuous function of z. Although the continuous function is a good candidate for visualization purposes, it does not stand well from computational point of view in digital hardware.

The DFT is a procedure to obtain a sampled frequency domain of a digital signal. The sampled version of frequency response is useful because it can be stored in memory of a digital system and can be utilized if we perform calculations in frequency domain. The DFT, $X(k)$, of a signal $x[n]$ with N samples is given by the following relation.

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn}, k = 0, 1, ..., N-1 \tag{9.1}$$

where the frequency resolution w_0 is given by:

$$w_0 = \frac{2\pi}{N} \tag{9.2}$$

The opposite of DFT is IDFT which is used to obtain the time domain signal from the frequency domain data.

9.1.1 The FFT and IFFT

The DFT given in (9.1) involves the computation of N complex multiplications and additions. So to compute the N outputs, we need N^2 complex multiplications and additions. This clearly shows that computing a DFT can become a daunting task for a processor.

A fast algorithm, FFT is used to compute the DFT of the time domain data of a signal. The ratio of computing cost (in terms of multiplication) in both the cases is approximately

$$\frac{FFT}{DFT} = \frac{log_2 N}{2N} \tag{9.3}$$

The number of samples for computation of FFT and IFFT must be some power of 2 for speedy calculation. To make the samples equal to some power of 2, we need to add trailing zeros to the input signals.

9.1.2 MATLAB's *fft*() Function

MATLAB also utilizes FFT to obtain the frequency domain of the signals. Same is the case for IDFT, where IFFT is utilized. To calculate the magnitude and phase of the FFT, we can use the MATLAB's *abs* and *angle* commands.

The frequency of a digital signal is represented in the limits from 0 to 2π rad, shown by f_s in Fig. 9.1. A digital signal cannot exist outside this frequency. So all the frequency domain plots in MATLAB are from 0 to 2π and it's the responsibility of the user to mark the axis correctly. Many of the times, input to an FFT block is a real signal and the real signal satisfy Hermition redundancy property, which states that the real output of the FFT is even and imaginary output of FFT is an odd function.

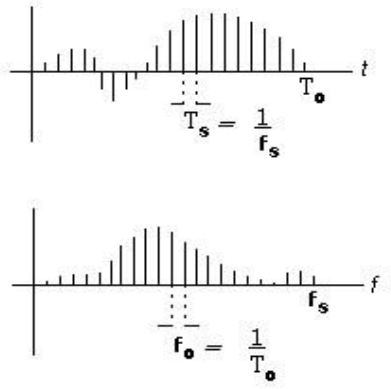

Figure 9.1: A digitized time domain signal and its DFT. Only magnitude of DFT is shown.

9.2 Practical

Generate 1024 samples of a signal containing two sinusoids of frequency 200 and 300 Hz, sampled at 1000 Hz. Obtain the frequency response of the resultant signal. Add some noise in the resultant signal and again compute the frequency response. Compare the two plots. Take IFFT of the contaminated signal and compare it to the original signal. Match your figure to the one given by Fig. 9.5.

9.3 Hints

The functions *fft*() and *ifft*() are used in MATLAB to calculate the frequency and time domain representations respectively. To contaminate a signal with noise, either use the function *awgn*() or *randn*(). The digital frequency *w* in rad is related to analog frequency *f* in Hz in the following manner.

$$w(rad) = 2\pi \frac{f(Hz)}{f_s(Hz)} \tag{9.4}$$

where f_s is the sampling frequency. Every index k on the frequency axis corresponds to the frequency kf_s/N (Hz). Also use MATLAB's help for *fftshift*() function.

9.4 Questions

1. Reduce the number of samples to 512. What happens to the signals? Why?
2. Using IFFT, generate 512 samples of a 1000 Hz sinusoid sampled at 4000 Hz.
3. Generate a contaminated sinusoid of frequency 100 Hz. Apply this signal to a moving-average filter. Plot the FFT of input to the filter and output of the filter.
4. Generate a signal containing two sinusoids of frequencies 400 Hz and 850 Hz, sampled at 4000 Hz. Plot its FFT for N = 100, 128, 200, 512.
5. Use fftshift() to represent the frequency in the range of -pi to pi. Use MATLAB's Help for this purpose.

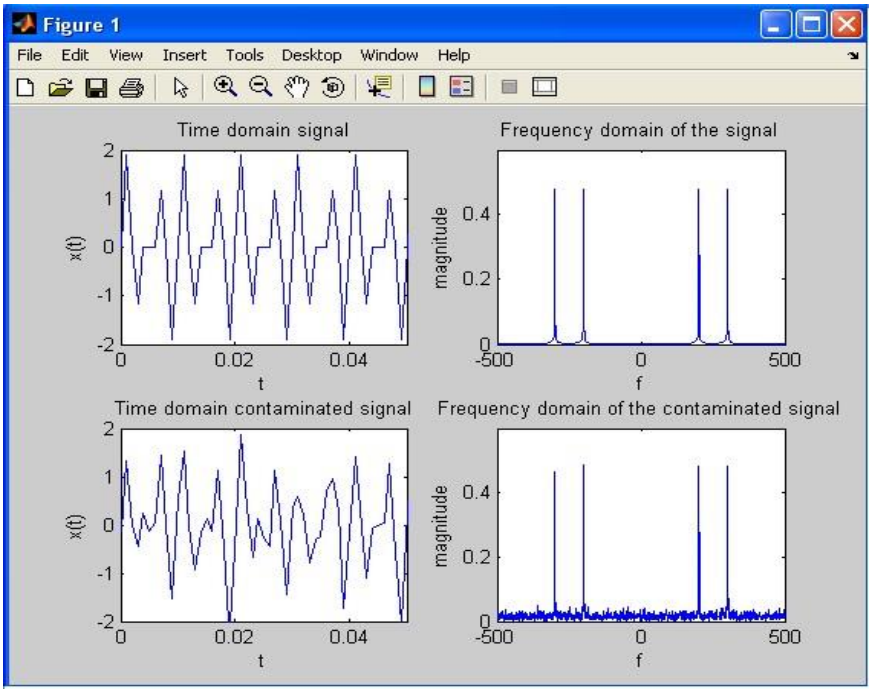

Figure 9.5: The Fourier Transform of a pure and noise-contaminated signal