SECTION 2

The Discrete Fourier Transform

2.1 The Discrete-Time Fourier Transform (DTFT)

"... the results need to be available within a finite time period, and the infinite summation must somehow be reduced to a finite summation."

In order to compute the Fourier transform using digital hardware, Eqn. 1-2 needs to be approximated in a manner which makes machine computation feasible. The first step in this process consists of eliminating the theoretical integral symbol, and replacing it by a computable sum:

$$X(f) \approx \tilde{X}(f) = T \sum_{n=-\infty}^{+\infty} \chi(nT)e^{-j2\pi fnT} \qquad \text{Eqn. 2-1}$$

The above expression uses a sampled signal $\chi(nT)$, where the sampling period T is made as small as possible to reduce approximation errors. Appropriately, $\tilde{X}(f)$ is called the discrete-time Fourier transform (DT-FT). As T (the sampling period) becomes infinitely small, the previous summation approaches the original Fourier transform in Eqn. 1-2. To assess the accuracy of this approximation, note that the resulting expression for $\tilde{X}(f)$ is a periodic function of frequency:

$$\tilde{X}(f) = \tilde{X}\left(f + \frac{1}{T}\right) \qquad \text{Eqn. 2-2}$$

because:

$$e^{-\left(j2\pi fnT+j2\pi n\frac{T}{T}\right)} = e^{-j2\pi fnT}e^{-j2\pi n} = e^{-j2\pi fnT} \qquad \text{Eqn. 2-3}$$

In general, the original spectrum X(f) is not periodic, and the approximation is only justified for a range of small values of f. In Figure 2-1, the DTFT magnitude and the Fourier transform magnitude of a simple rectangular function are shown for several values of the sample rate $f_S = 1/T$. Note the periodic nature of the resulting function, as well as the approximation errors due to the sampling process.

The Nyquist sampling theorem gives a well accepted criterion for the sampling rate. It states that a signal needs to be sampled faster than twice its highest frequency. In other words, if:

$$X(f) = 0 \qquad \text{Eqn. 2-4}$$

for $|f| \geq B$ (B is referred to as the bandwidth of the signal), then the sampling frequency needs to satisfy:

$$f_S \geq 2B \qquad \text{Eqn. 2-5}$$

In practice, signals rarely satisfy Eqn. 2-5, and some error, called the aliasing error, can be expected in the evaluation of X(f). The aliasing error is generated by frequency components at higher frequencies, which manifest themselves at lower frequencies because of the periodic nature of $\tilde{X}(f)$ (aliasing). The aliasing error can be reduced by filtering out the higher-frequency components of the signal using a low-pass anti-aliasing filter and/or by increasing the sampling rate.

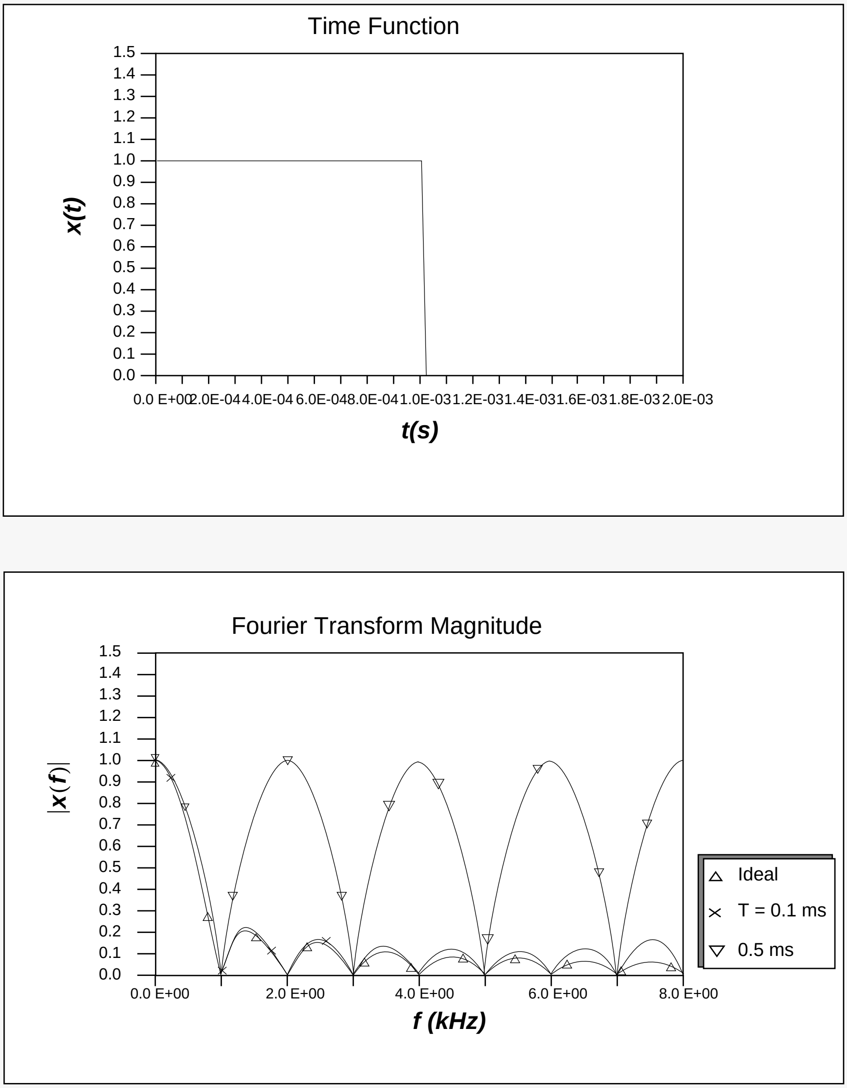

Figure 2-1 Fourier Transform of a Rectangular Function

2.2 Windowing and Windowing Effects

The discussion of aliasing errors illustrates how the Fourier transform can be approximated by an infinite summation. In practice, the results need to be available within a finite time period, and the infinite summation must somehow be reduced to a finite summation. One obvious way to reduce the infinite summation is by simply truncating the sum in Eqn. 2-2 to N terms as:

$$\tilde{X}_W(f) = T\sum_{n=0}^{N-1}\chi(nT)e^{-j2\pi fnT} \qquad \text{Eqn. 2-6}$$

This truncation is frequently referred to as “windowing” because an infinite summation is viewed through a finite window. The resulting transform is called the windowed discrete-time Fourier transform (WDTFT). In mathematical terms, windowing is simply the multiplication of the signal by a window sequence of finite-length, w(n). In the simple case above, w(n)=1 for $0 \le n \le N-1$; otherwise, w(n)=0. Because of its rectangular shape, the window shown above is called the rectangular window.

Unless the signal in question is of finite duration, this truncation will introduce other errors, resulting in a number of artifacts in the spectrum. To assess the effect of the windowing operation, a simple sine wave of the form:

$$\chi(t) = \sin(2\pi 1000t) \qquad \text{Eqn. 2-7}$$

is sampled with a sampling frequency of 4000 Hz, and the windowed DTFT is computed with N=20.

Figure 2-2 shows the result of windowing a sine wave by a rectangular window. Windowing causes the following errors:

1. Leakage — Even though the input signal consists of a single-frequency component at 1000 Hz, the result clearly shows components at frequencies other than 1000 Hz. This is called the leakage effect: it appears as if energy has “leaked” from 1000 Hz to the rest of the spectrum.

2. Smoothing — Although the theoretical transform exhibits an infinitely narrow, and infinitely large peak at 1000 Hz, the actual peak has finite magnitude and exhibits finite width. It appears that the narrow peak has been “smeared” out in the frequency domain as a result of the windowing function in the time domain. This effect is appropriately termed the smoothing effect.

3. Ripple — The overall magnitude plot in Figure 2-2 shows an oscillatory character not present in the original Fourier transform: this is called the ripple effect. The origin of the ripple effect lies in the discontinuity (abrupt start and end) introduced in the signal by the window. Windows with more gradual transitions generally have lower sidelobes and less ripple.

In general, a tradeoff exists between these different effects, and the advantages of an appropriate windowing function can be chosen for a specific application. For an excellent summary of existing windowing functions and their properties, see Reference 2.

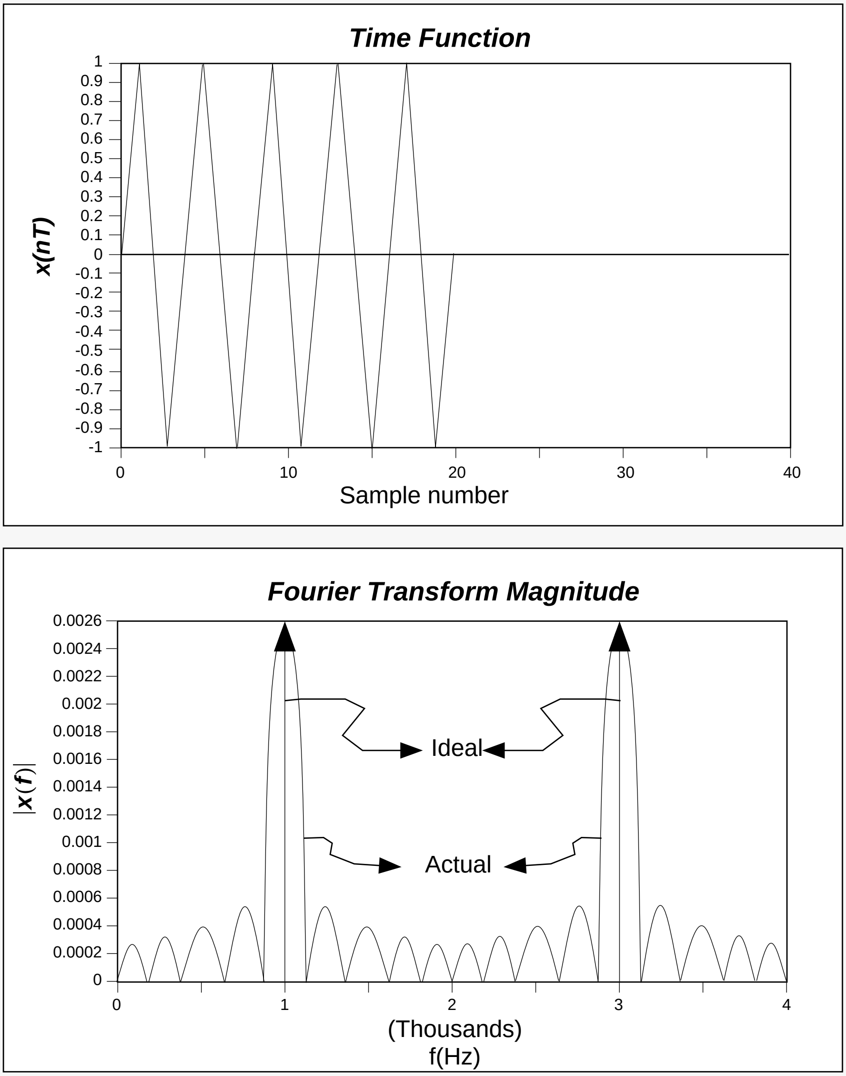

Figure 2-2 Windowing Effects When Windowing a Single Sine

2.3 Sampling the Frequency Function

The windowed DTFT is now ready for machine computation, with one exception: the independent frequency variable f is still a continuous variable, and needs to be captured in discrete intervals, or sampled. Since the DTFT is periodic in the frequency domain with period f_S, only values of f from 0 to f_S (the sampling frequency) need to be computed. Although there are similar arguments concerning the distance between successive frequency samples as in the case of time-sampling, it turns out that when the WDTFT is sampled every f_S/N Hz, fast algorithms for computing the transform can be derived. Note that in this case, the number of samples in the window (N) and the number of samples in the frequency domain (N) are equal. The resulting transform is called the discrete-time Fourier series (DTFS):

$$\tilde{X}_N(k) = T\sum_{n=0}^{N-1}\chi(nT)e^{-j\frac{2\pi}{N}nk} \qquad \text{Eqn. 2-8}$$

The inverse DTFS is given by:

$$\chi_N(k) = \frac{1}{NT}\sum_{k=0}^{N-1}\tilde{X}_N(k)e^{j\frac{2\pi}{N}nk} \qquad \text{Eqn. 2-9}$$

Keep in mind that the values of the frequency samples of f_k are equal to $[f_S/N]\ k$.

Note that many textbooks simply define the Discrete Fourier transform (DFT) $X_N(k)$:

$$X_N(k) = \sum_{n=0}^{N-1} \chi(nT) e^{-j\frac{2\pi}{N}nk} \qquad \text{Eqn. 2-10}$$

with inverse transform:

$$\chi_N(n) = \frac{1}{N} \sum_{n=0}^{N-1} X_N(k) e^{j\frac{2\pi}{N}nk} \qquad \text{Eqn. 2-11}$$

Obviously, the DFT and DTFS differ only by a scaling factor of T, making the spectrum independent of the sampling period. Consequently, explicit T dependence can be dropped from Eqn. 2-11.

Although the sequence $x_N(n)$ corresponds to the original sampled and windowed sequence $\chi(nT)$ for sampling instants 0 through N-1, the complete sampled sequence $\chi(nT)$ for any n cannot necessarily be recovered from it. Indeed, $x_N(n)$ appears to be periodic with period N due to

the periodicity of $e^{j\frac{2\pi}{N}nk}$, whereas the original sampled signal was not assumed to be periodic.[1]

---

[1] The error introduced in the time domain by sampling a frequency function is termed “aliasing in time” which is analogous to the “aliasing in frequency” caused by sampling a time function. (See SECTION 2.1 The Discrete-Time Fourier Transform (DTFT)). That is, if a frequency spectrum is not sampled densely or closely enough, the signal constructed in the time domain through the inverse “discrete-frequency Fourier transform” will show some distortion.

This must be kept in mind in convolution-based applications, where the forward as well as inverse transforms are used; the incoming signal stream needs to be segmented, and the computed signal segments need to be pieced together to construct the complete output stream. Most basic text-books on digital signal processing discuss techniques for piecing together the output stream (see Reference 3). ■