Chapter 5
Window Functions

5.1 Introduction

As discussed in section (3.7.5), the DTFS assumes that the input waveform is periodic with a period of *N* (number of samples). This is observed in table (3.1). Therefore, if the time-domain waveform completes and integer number of cycles (no greater than half the sampling frequency) over the course of the *N* samples, then the waveform is properly prepared for DTFS, or FFT, analysis. Often times though, this is not the case, and the resulting sampled data set represents a truncated version of the periodic signal. For example, consider a sinusoid that completes one full cycle over a period of 360 samples. If only 270 samples are taken (starting at the rising edge of the sinusoid), then the resulting waveform is displayed in figure (5.1).

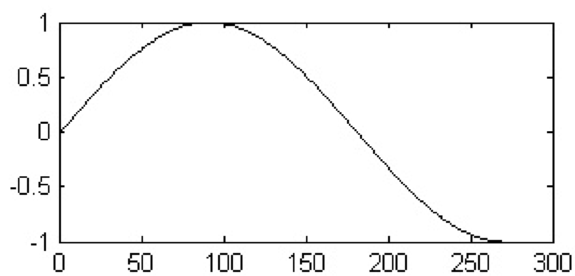

Figure 5.1 Truncated Sinusoid

Recall that the DTFS assumes the input signal is periodic over the *N* samples. Therefore, the DTFS periodically extends the waveform assuming a period of *N*, and the DTFS actually "sees" the waveform in figure (5.2). As a result, there are transients (discontinuities) at integer multiples of *N*. These discontinuities are artificially created by the DTFS. As discussed in section (3.7.5), the DTFS is not well suited for transient signal analysis. These transients correspond to high-frequency sinusoids that were not present in the original signal. If these frequencies are higher than the Nyquist frequency as discussed in section (3.4.6), they will appear to have frequencies between 0 Hz and half the sampling rate. In other words, they are aliased to within the unambiguous frequency range of the DTFS. This is spectral leakage at work as the energy from high frequencies due to the artificial transients leak into the frequency range of the DTFS. As discussed in section (3.7.4), spectral leakage is also caused by a sinusoid's frequency not lying directly at the center frequency of a DTFS bandpass filter [2].

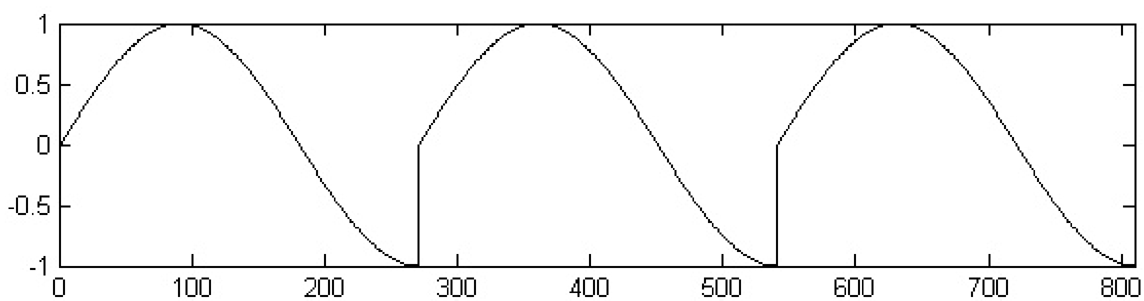

Figure 5.2 Periodically Extended Version of Figure 5.1

Spectral leakage will cause the energy of the FFT spectrum to spread out instead of being concentrated at the actual frequencies that are present in the time-domain signal. The FFT spectrum will not have highly distinct peaks at the true frequencies. Instead, it will have small peaks at the true frequencies and will have a slow roll-off rate outside of these frequencies. Figure (5.3) is a sinusoid that has been sampled over an integer number of cycles. The corresponding FFT of this sinusoid is shown in figure (5.4). Notice the distinct peaks and the fast roll-off rate. In contrast, figure (5.5) represents the case where the number of samples taken does not cover an integer number of cycles; therefore, a truncated sinusoid is processed by the FFT. The FFT of this truncated sinusoid is shown in figure (5.6). Notice the figure (5.6) has wider responses at the peaks and has a slower roll-off rate.

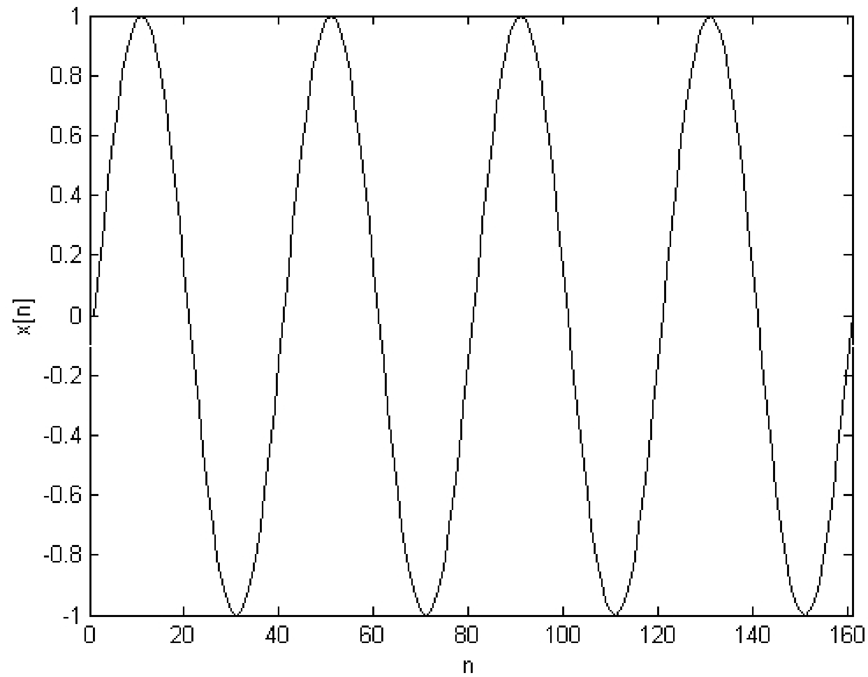

Figure 5.3 Sinusoid with Four Complete Cycles

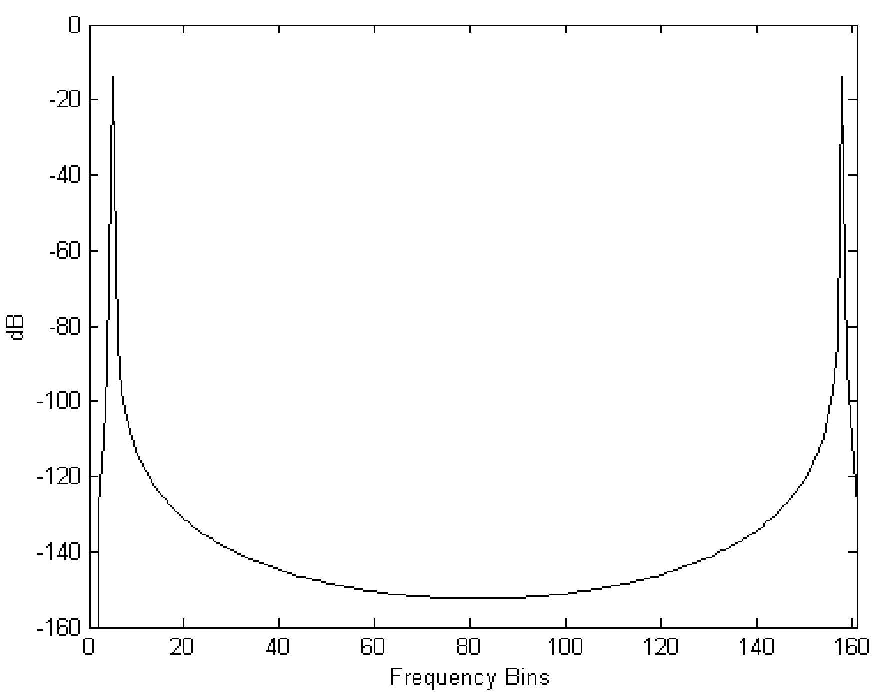

Figure 5.4 FFT without Spectral Leakage

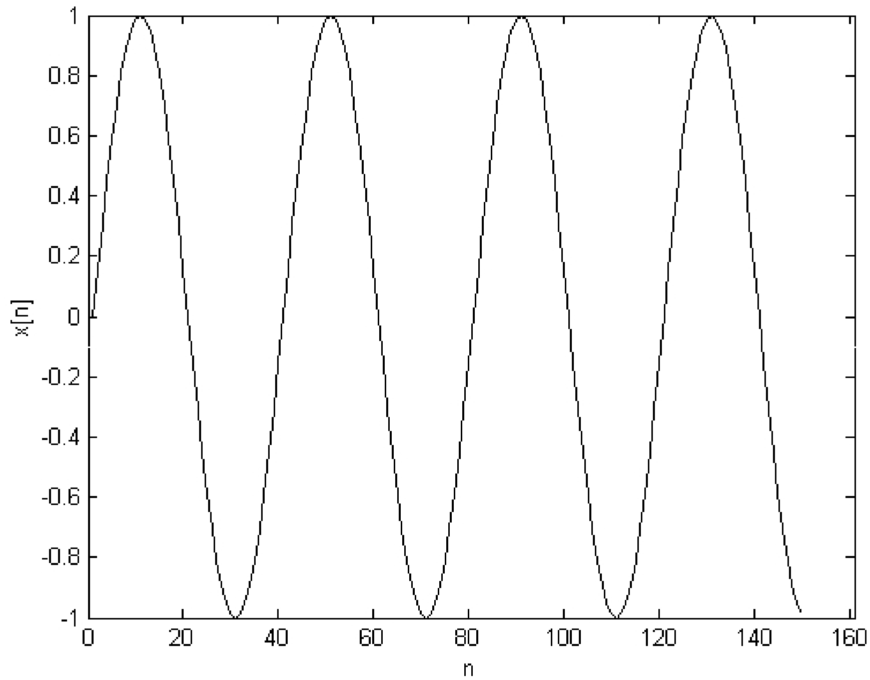

Figure 5.5 Sinusoid with Incomplete Cycle

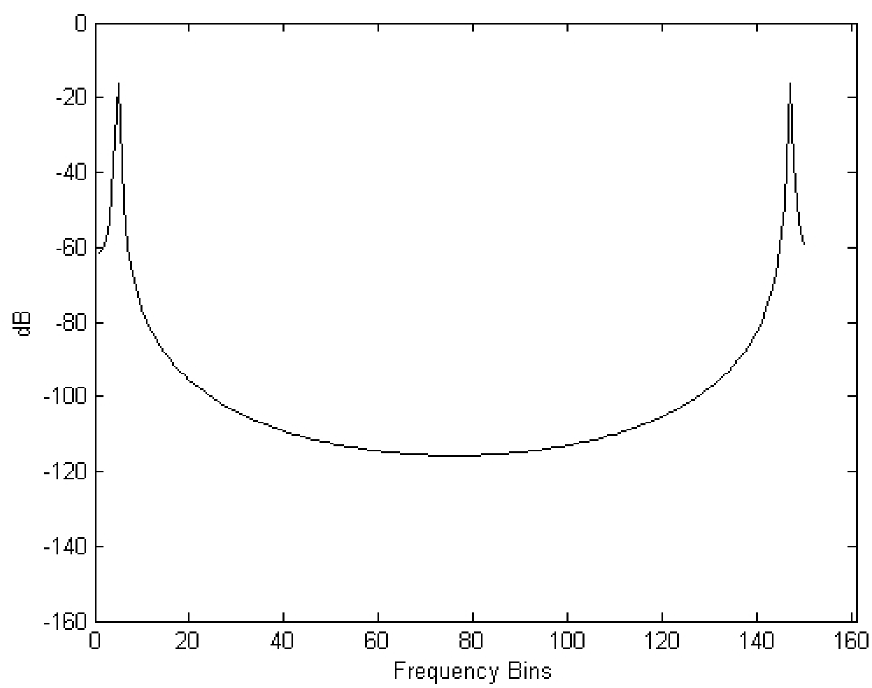

Figure 5.6 FFT with Spectral Leakage

In order to help minimize the effects of spectral leakage, window functions (or weighting functions) can be applied to the input signal [2].

5.2 Window Functions

A window function can be applied to an input data set in order to improve the FFT response. The time-domain data is simply multiplied by another discrete, time-domain function called a window function. These window functions tend to taper off the outside edges of the input signal. This diminishes the effect of the artificial transients produced by the periodic extension of the FFT.

Several types of window functions exist and each have their own unique characteristics. Each window function is better suited for a specific type of application with a certain input signal and a certain type of desired analysis. In order to select the optimum window for an application, the user must be aware of the behavior of the input signal as well as the type of frequency analysis that is required [2].

An important aspect of window functions is that the processes of selecting the window function and the FFT algorithm are independent of each other. Since FFT algorithms are not approximations to the DTFS but are alternate methods of calculating it, all FFT algorithms produce the same result. They go about the calculations using different methods but ultimately reach the same result. In consequence, window functions can be used with any FFT algorithm. Also, the window functions can be used with any length FFT, since the input data sequence can be zero-padded to fit the FFT length as discussed in section (3.5). Another important aspect of window functions is that they do not alter the resolution of the FFT. Therefore, applying a window function to the input data set does not alter the bin spacing, or frequency spacing, in the output spectrum of the FFT [2].

Recall that the input data set is multiplied by the window function in the time domain. This corresponds to convolution in the frequency domain. The convolution takes the window function's spectral content, reverses it, and copies it to every point in the original signal's discrete spectral content. In other words, reversed replicas of the spectral content of the window function are centered about every frequency bin (or DTFS coefficient) of the original signal's FFT [5].

5.3 Characteristics of Window Functions

The following six characteristics are used as quality factors of each window function. By assessing the characteristics of each window function, an appropriate determination can be made as to which window function to use for a specific application.

5.3.1 Coherent Integration Gain

As discussed in section (3.6.4) the DTFS leads to a coherent integration gain of N (using equation (3.1) without the normalization constant). All of the window functions, with exception to the rectangular, lead to a coherent integration gain less than N [2].

5.3.2 Highest Sidelobe Level

Each DTFS coefficient acts as a bandpass filter. As discussed in section (3.7.3), the frequency response of these filters is not ideal. Instead, the filters have a sinc-shaped frequency response with a main lobe in the center and smaller sidelobes on either side. The high sidelobe levels allow a signal that is not centered at the center frequency of the DTFS filter to produce a significant response from that particular DTFS output coefficient. A lower the sidelobe level leads to higher rejection of signals that are not centered at the center frequency of the filter [2].

5.3.3 Sidelobe Rolloff Ratio

As frequency increases further away from the center frequency of a DTFS filter, the sidelobes discussed in section (5.3.2) begin to decrease in amplitude or remain constant. This rolloff rate usually has dimensions of decibels per decade or decibels per octave [5]. A fast rolloff, or fall-off, rate diminishes the ability for an off-center frequency component to stimulate a significant response from a DTFS filter. Therefore, a higher rolloff rate leads to higher credibility as to where the frequency component lies on the frequency axis. In other words, the DTFS becomes more frequency-selective, allowing only signals that are extremely close to the center frequency of the DTFS filter to pass through. The main-lobe width discussed in section (5.3.6) is also a contributing factor to frequency selectivity. [2].

5.3.4 Frequency Straddle Loss

Frequency straddle loss was described in section (3.7.4) as the difference between the maximum coherent integration gain and the actual achieved coherent integration gain. If a signal does not lie at the center of the DTFS bandpass filter then its contribution to the DTFS component will be reduced. This is due to the sidelobe rolloff ratio discussed in section (5.3.3). A signal that lies exactly between two adjacent filters has the lowest coherent integration gain [2].

5.3.5 Equivalent Noise Bandwidth

Noise is an inherent part of any system, and therefore its impact on the system must be understood. Generally, noise is spread across the frequency domain. White noise is evenly distributed across the frequency domain, meaning that the noise has the same magnitude at all frequencies in the spectrum. Since the DTFS filters pass the noise as well as the signal, it is important to understand how much of this noise reaches the output of the filter. Consider an ideal bandpass filter with a rectangular frequency response and a gain equal to the peak gain of the DTFS filter. The equivalent noise bandwidth of the DTFS filter is the bandwidth of the ideal bandpass filter that would allow the same amount of white noise power to pass through as the DTFS filter [2].

5.3.6 Three-dB Main Lobe Bandwidth

As discussed in section (5.3.2), the DTFS filters have a sinc-shaped response with a main-lobe in the center and smaller sidelobes as frequency increases away from the center frequency of the filter. The width of this main lobe determines the range of frequencies about the center frequency of the filter which can pass through without being attenuated significantly. The 3-dB bandwidth is the range of frequencies which a signal can pass without its power being attenuated by more than a factor of 2 (or $10*LOG_{10}(2)$ dB). Having a narrower main lobe improves the accuracy of the DTFS to resolve frequencies. In contrast, a narrower main lobe reduces the range of frequencies about the center frequency that can be resolved by the DTFS [2].

5.4 Common Window Functions

Many different window functions exist and allow the user to choose from a broad range of characteristics. This section will discuss some of the more common window functions including their equations and characteristics.

5.4.1 Rectangular

The rectangular window function is applied simply by not modifying the input data set before it is processed by the FFT. It is inherent in the DTFS/FFT. The equation for the rectangular window function is

$$w(n) = \begin{cases} 1, & for\ n = 0\ to\ N\text{ - }1 \\ 0, & elsewhere \end{cases} \tag{5.1}$$

where N is the total number of samples.

The peak of the highest sidelobe is 13 dB below the peak of the main lobe. This corresponds to an attenuation of approximately 4.5 (or $10^{(13dB/20)}$) times less than that of the peak main-lobe attenuation [2]. This attenuation is in terms of the magnitude of the frequency component as opposed to power. This is considered a poor sidelobe performance compared to the other window functions.

The rectangular window function redeems itself by having a narrower main lobe and higher coherent integration gain than any of the other weighting functions. This causes the rectangular window to give the smallest output noise power but also the highest straddle loss. Since the rectangular window function has the lowest output noise power and highest coherent integration gain, it is better suited for applications where maximum signal-to-noise ratios are required [2].

5.4.2 Triangular

The triangular window function is defined in the time domain by equation (5.2):

$$w(n) = \begin{cases} 2 * \dfrac{n}{N}, & for\ n = 0\ to\ \dfrac{N}{2} \\ 2 * \dfrac{N-n}{N}, & for\ n = \dfrac{N}{2} + 1\ to\ N\text{ - }1 \\ 0, & elsewhere \end{cases} \tag{5.2}$$

The triangular window function has a highest sidelobe level of -27 dB, an improvement over the rectangular window function. It also has a higher sidelobe level rolloff ratio of –12 and a lower straddle loss. As shown in table (5.1) the triangular window function has a higher output noise power and wider 3-dB main lobe width [2].

5.4.3 Sine Lobe

The sine lobe window function is defined in the time domain by equation (5.3):

$$w(n) = \begin{cases} \sin\left(\dfrac{\pi n}{N}\right), & \textit{for } n = 0 \textit{ to } N - 1 \\ 0, & \textit{elsewhere} \end{cases} \tag{5.3}$$

The sine lobe window function improves the coherent integration gain over the triangular window function. The highest sidelobe level is between those of the rectangular and triangular window functions. The sine lobe window function has approximately the same sidelobe rolloff ratio as the triangular window function [2].

5.4.4 Hanning

The Hanning window function is defined in the time domain by equation (5.4):

$$w(n) = \begin{cases} \left(\dfrac{1}{2}\right)\left(1 - \cos\left(\dfrac{\pi n}{N}\right)\right), & \textit{for } n = 0 \textit{ to } N - 1 \\ 0, & \textit{elsewhere} \end{cases} \tag{5.4}$$

The Hanning window function is accompanied by an improvement in the highest sidelobe level and sidelobe rolloff ratio over the previous window functions [2].

5.4.5 Hamming

The Hamming window function is defined in the time domain by equation (5.5):

$$w(n) = \begin{cases} \left(0.54 - 0.46 * \cos\left(\dfrac{2\pi n}{N}\right)\right), & \textit{for } n = 0 \textit{ to } N - 1 \\ 0, & \textit{elsewhere} \end{cases} \tag{5.5}$$

The Hamming window function provides an even greater improvement to the highest sidelobe level but is accompanied by a poorer sidelobe rolloff ratio (equal to that of the rectangular window function).

5.4.6 Blackman

The Blackman window function is defined in the time domain by equation (5.6):

$$w(n) = \begin{cases} \left(0.42 - 0.50 * \cos\left(\frac{2\pi n}{N}\right) + 0.08 * \cos\left(\frac{4\pi n}{N}\right), & \textit{for } n = 0 \textit{ to } N - 1 \\ 0, & \textit{elsewhere} \end{cases} \tag{5.6}$$

The Blackman window function has the lowest coherent integration gain out of the six window functions discussed in this chapter, but it also has the lowest of the sidelobe level responses. The sidelobe rolloff ratio is equal to that of the Hanning window function. This window function provides high rejection of signals outside of its main lobe [2].

The characteristics discussed above about the window functions can be observed in table (5.1) below [2].

Window Function	Coherent Integration Gain	Highest Sidelobe Level (dB)	Sidelobe Rolloff Ratio	Frequency Straddle Loss (dB)	Equivalent Noise Bandwidth	3-dB Bandwidth
Rectangular	1.00	-13	-6	3.92	1.00	0.89
Triangular	0.50	-27	-12	1.82	1.33	1.28
Sine Lobe	0.64	-23	-12	2.10	1.23	1.20
Hanning	0.50	-32	-18	1.42	1.50	1.44
Hamming	0.54	-43	-6	1.78	1.36	1.30
Blackman	0.42	-58	-18	1.10	1.73	1.68

Table 5.1 Window Function Comparison Chart

5.5 Summary

The weaknesses of the DTFS discussed in chapter 3 as well as the usual truncation of a periodic signal provide a need to manipulate the data before it is processed by the DTFS or FFT. This can be done easily by using window functions, or discrete time-domain functions that are designed to minimize the weaknesses inherent in the DTFS. These improvements, though, do not come without flaw. As with most engineering processes, tradeoffs are involved in applying window functions. The user must weigh each characteristic of the window functions and determine which are most important and which are least important to the specific application. Several other window functions can be found in [2] in addition to the window functions discussed above in section (5.4).