Chapter 3
Discrete-Time Fourier Series

3.1 Introduction

The Fourier series and Fourier transforms are mathematical correlations between the time and frequency domains. They are the result of the heat-transfer work performed by the French mathematician Jean Baptiste Joseph Fourier in the early 1800’s. The Fourier representations are extremely important to engineering and science because they can take a time-domain function and convert it to the corresponding frequency-domain representation. The time-domain functions are not limited to a certain type of time-domain signal, or waveform, but may represent a wide variety of natural and man-made signals such as seismic vibrations, mechanical vibrations, electrical signals, and even stock fluctuations [2].

Fourier analysis can be an extremely useful tool in certain situations. Often times, converting a signal into another domain or transforming the signal can make the mathematics easier. Also, if a signal is passed through a linear time-invariant system (LTI), then the frequency components themselves are not shifted, but altered only in magnitude and phase. This means that the effect of the LTI system on the spectral content of the signal is much easier to predict than the effect on the signal’s time domain [4].

The Fourier series and Fourier transform take a continuous, time-domain function and represent it as a weighted sum of complex sinusoids. Mathematicians discovered after Fourier's work that the signals did not have to be continuous in the time domain but could instead be discrete data points. This led to the inception of the discrete-time Fourier series and the discrete-time Fourier transform [2].

Four different Fourier representations exist:

1. Fourier series (FS)
2. Fourier transform (FT)
3. Discrete-time Fourier series (DTFS)
4. Discrete-time Fourier transform (DTFT)

The discrete-time Fourier series that this chapter will primarily focus on is often referred to as the discrete Fourier transform, or DFT. This terminology is not to be confused with the discrete-time Fourier transform.

The factor that determines whether to use a series or a transform is periodicity in the time-domain signal. For a periodic signal in the time domain, the series is used. For a non-periodic signal in the time domain, the transform is used.

The Fourier series is used to convert a continuous and periodic time-domain signal into the frequency domain. The resulting frequency domain representation from performing the Fourier series is discrete and non-periodic.

The Fourier transform is used to convert a continuous and non-periodic time-domain signal into the frequency domain. The resulting frequency domain representation from performing the Fourier transform is continuous and non-periodic.

The discrete-time Fourier series is used to convert a discrete and periodic time-domain signal into the frequency domain. The resulting frequency domain representation from performing the discrete Fourier series is discrete and periodic.

The discrete-time Fourier transform is used to convert a discrete and non-periodic time-domain signal into the frequency domain. The resulting frequency domain representation from performing the discrete Fourier transform is continuous and periodic. These relationships are shown below in table (3.1) [1].

Time Domain	**Periodic**	**Nonperiodic**	
Continuous	FS	FT	**Nonperiodic**
Discrete	DTFS	DTFT	**Periodic**
	Discrete	**Continuous**	**Frequency Domain**

Table 3.1 Time and Frequency Properties of the Four Fourier Representations

3.2 Definition of the DTFS

Since the discrete-time Fourier series is the only Fourier representation with a finite number of discrete components in both the time and frequency domains, it is often used as a computational tool for analyzing signals. The DTFS is the only Fourier representation that can be evaluated numerically by a computer or a digital system.

The discrete-time Fourier series (DTFS) is given by

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j\Omega_0 kn} \tag{3.1}$$

where $\Omega_0 = \frac{2\pi}{N}$ and N is the number of data points in the data set.

The inverse DTFS uses the DTFS coefficients to represent or reconstruct the original time-domain signal $x[n]$.

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j\Omega_0 kn} \tag{3.2}$$

It should be observed that the DTFS and the inverse DTFS are fundamentally the same operation. The only difference is the normalization factor (1/*N*) and the sign of the complex exponential.

Some texts place the normalization constant (1/*N*) in front of the summation in equation (3.2) and remove it from equation (3.1). By doing this, each DTFS coefficient *X*[*k*] will be *N*-times as large as the DTFS coefficients solved by using equation (3.1). As a result, the calculated magnitude of the sinusoid at each frequency will have an amplitude *N*-times greater than the sinusoid's actual amplitude. This is a part of c*oherent integration gain* and will be discussed in section (3.6.4). In this chapter, equations (3.1) and (3.2) will be used to describe the DTFS and the inverse DTFS, respectively.

Also note that by Euler's identity:

$$e^{j\Omega_0 kn} = \cos(\Omega_0 kn) + j\sin(\Omega_0 kn) \tag{3.3}$$

$$e^{-j\Omega_0 kn} = \cos(\Omega_0 kn) - j\sin(\Omega_0 kn) \tag{3.4}$$

The magnitude of the coefficients *X*[*k*] are used to generate the magnitude spectrum of the time-domain signal. The DC coefficient, *X*[0], is equal to the magnitude of the DC component. The magnitude of the other coefficients is equal to the half the amplitude of the input sinusoid at that specific frequency. The phase angles of the coefficients are used to generate the phase plot. The magnitude of *X*[*k*] = *a* + *jb* is given by the following [8]:

$$|X[k]| = \sqrt{a^2 + b^2} \tag{3.5}$$

and the phase angle of *X*[*k*] is given by

$$\angle X[k] = \arctan\left(\frac{b}{a}\right) \tag{3.6}$$

3.3 Twiddle Factor

In equation (3.1) the complex exponential in the summation is called the twiddle factor [4].

$$\text{Twiddle Factor} = W_N^{nk} = e^{-j\Omega_0 kn} = e^{\frac{-j2\pi kn}{N}} \tag{3.7}$$

The twiddle factor can be described as a vector that rotates about a unit circle that lies in the real-imaginary plane. From equation (3.4) the real and imaginary parts of the complex exponential can be evaluated as follows:

$$\mathrm{Re}(e^{-j\Omega_0 kn}) = \cos(-\Omega_0 kn) \tag{3.8}$$

$$\mathrm{Im}(e^{-j\Omega_0 kn}) = -\sin(-\Omega_0 kn) \tag{3.9}$$

For example, let *N* = 8 and *k* = 1. This means that there are eight data points and the coefficient *X*[1] is being solved for. Figure (3.1) illustrates the rotating nature of the complex exponential around the unit circle as *n* increases from zero to seven.

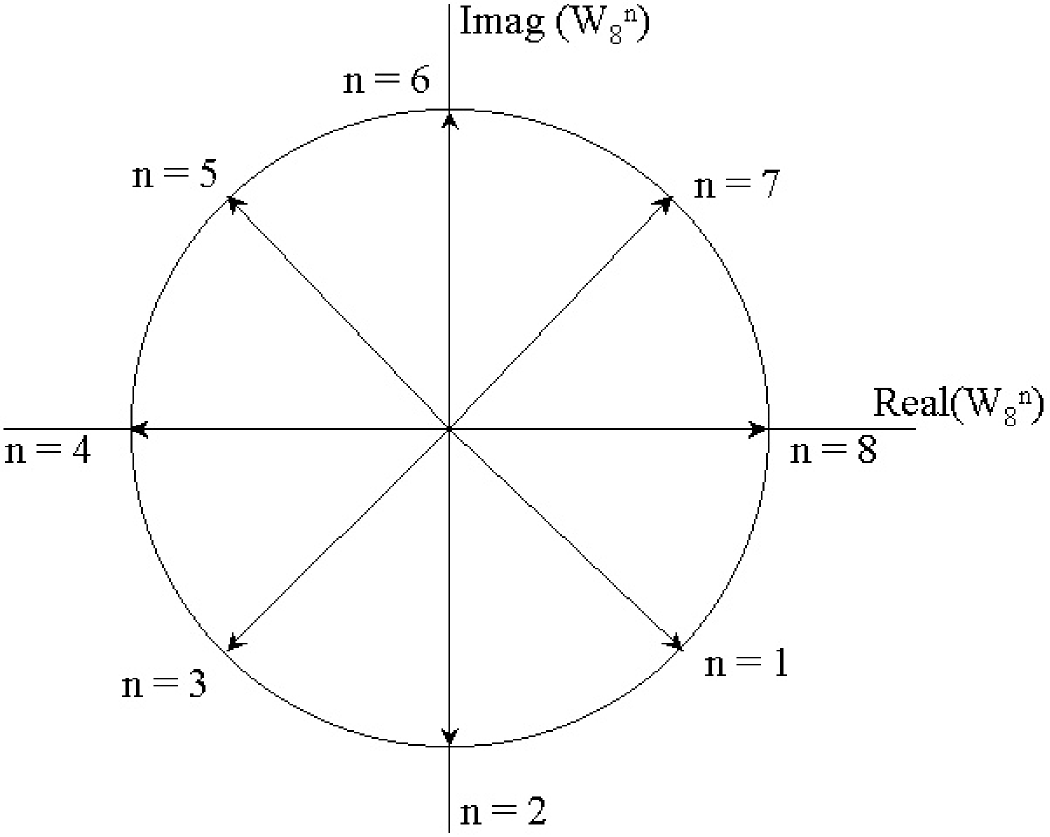

Figure 3.1 Twiddle Factor

In this example, eight data points cover the 2π radians. The vectors around the unit circle are all equally spaced by 2π radians divided by *N*. Increasing *N* will increase the number of discrete values that the complex exponential takes on to cover the 2π radians. As illustrated in section (3.4.3), increasing *N* decreases the frequency spacing of the DTFS coefficients. When solving for *X*[2], *k* is equal to 2. This means that the complex exponential rotates around the unit circle at twice the rate as when *k* was 1. Intuitively, it can be thought of as solving for the frequency component that is twice as fast as *X*[1], or completes the 2π radians in half the time as *X*[1]. When solving for *X*[3], *k* is equal to 3, and the complex exponential rotates three times as fast as if *k* was 1 and so on. In a more general description, the complex exponential in solving for *X*[*k*] rotates *k*-times as fast as the complex exponential in solving for *X*[1].

The complex exponential is inversely symmetric about the origin. Mathematically this is represented as

$$e^{\frac{-j2\pi kn}{N}} = -e^{\frac{-j2\pi k}{N}(n\pm\frac{N}{2})} \tag{3.10}$$

Equation (3.10) can also be written as follows:

$$e^{\frac{-j2\pi kn}{N}} = -e^{\frac{-j2\pi|nk|_N}{N}} \tag{3.11}$$

where $|nk|_N$ is the modulo-*N* of *nk*.

Equation (3.10) represents the circularity property of the complex exponential. The cyclic nature of the complex exponential plays a primary role in removing redundancy from the DTFS [8]. The inverse of a point on the unit circle is the point π radians away. For the discrete case, one full revolution around the unit circle is equal to *N* points; therefore a displacement of π radians is equal to *N*/2 points. The inverse property of the complex sinusoid can be seen from equation (3.4) and the following trigonometric identities:

$$\cos(x+\pi) = -\cos(x) \tag{3.12}$$

$$\sin(x+\pi) = -\sin(x) \tag{3.13}$$

This implies that in the discrete case, the first half of the *N* output samples contain all of the necessary information. The second half is only an inverse of the first half.

3.4 Properties of the DTFS

3.4.1 Linearity

The linearity property states that the DTFS of a sum of inputs is equal to the sum of the DTFS for each input. This relationship is shown in equation (3.14).

$$C[k] = \frac{1}{N}\sum_{k=0}^{N-1}\left[a[n]+b[n]\right]e^{-j\Omega_0 kn} \tag{3.14}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}a[n]e^{-j\Omega_0 kn} + \frac{1}{N}\sum_{k=0}^{N-1}b[n]e^{-j\Omega_0 kn} \tag{3.15}$$

$$= A[k] + B[k] \tag{3.16}$$

3.4.2 Symmetry

Let the DTFS of a time-domain signal *a*[*n*] equal *A*[*k*]. The symmetry property states that the DTFS of a time-domain signal *b*[*n*], where *b*[*n*] = *A*[*n*], is equal to *a*[*N*-*k*].

3.4.3 Time and Frequency Scaling

From equation (3.1) it is observed that the DTFS coefficients *X*[*k*] are not a function of the sampling frequency. This means that if two separate sets of data points are equal, then their DTFS coefficients are also equal even if the two sets of data points were acquired at different sample rates. The distinction becomes apparent when plotting the DTFS coefficients. The scale on the horizontal (frequency) axis depends on the sampling rate and the number of data points in the data set. The frequency spacing called the *frequency resolution*, or *bin spacing,* is the distance between the DTFS coefficients on the frequency axis. The bin spacing also represents the smallest frequency that can be resolved through the DTFS with a given sample rate and number of data points. The DTFS coefficient *X*[0] represents the DC component of the time-domain signal while *X*[1] represents the 0 + bin spacing sinusoidal component of the time-domain signal. The frequency spacing between DTFS coefficients is determined by the following relationship:

$$\text{Bin Spacing} = \frac{\text{sample rate}}{N} = \frac{\text{samples/sec}}{N} \tag{3.17}$$

This implies that as the sample rate increases, the frequency spacing between *X*[*k*-1] and *X*[*k*] increases. Also, if the number of data points in the data set (*N*) increases, then the frequency spacing between *X*[*k*-1] and *X*[*k*] decreases [2].

3.4.4 Time and Frequency Shifting

Shifting the signal in the time-domain is the equivalent to multiplying the signal's frequency domain by a linear phase shift. This can be represented mathematically as follows [4]:

$$\text{DTFS}\{x[n - n_0]\} = e^{-j\Omega_0 k n_0} X[k] \tag{3.18}$$

As discussed in section (3.2), the DTFS and the inverse DTFS are basically the same operation except for the normalization factor and the sign of the complex exponential. Because of symmetry, multiplying the time domain signal by the complex exponential in equation (3.9) will result in shifting the signal's frequency domain. This can be described mathematically as follows [4]:

$$\text{DTFS}\{e^{j\Omega_0 k_0 n} x[n]\} = X[k - k_0] \tag{3.19}$$

3.4.5 Parseval's Theorem

Parseval's theorem can be used to determine the power of an input data set once it has already been converted to the frequency domain. The power of the input data set in the time domain is the sum of the squares of each value in the data set. Parseval's theorem states that the power of the input data sequence is also equal to the sum of the squares of each of the DTFS coefficients of the data set divided by the number of data points. This relationship is shown in equation (3.20):

$$\sum_{n=0}^{N-1} \left| x[n] \right|^2 = \frac{1}{N} \sum_{n=0}^{N-1} \left| X[k] \right|^2 \tag{3.20}$$

Using equation (3.20) the power of the input data set can be easily calculated with the original data set or with the DTFS coefficients [2]. Also, note that the conservation of energy is upheld through the DTFS [1].

3.4.6 Periodicity

Let $x[n]$ be a set of N complex numbers. If the DTFS is performed on $x[n]$ for $0 \leq k \leq N$-1, then the output of the DTFS will be the set of corresponding frequencies $X[k]$. If k takes on values less than zero or greater than N-1, then the complex exponential in equation (3.1) will repeat itself with a period N; therefore, the DTFS coefficients generated are periodic repetitions of the original set. This is due to the circularity property of the complex exponential discussed in section (3.3). This leads to the conclusion that equation (3.1) is a periodic function of k with period N. Equation (3.2) reflects the same type of periodicity [8]. The set $x[n]$ is defined only over the range $0 \leq n \leq N$-1, but the inverse-DTFS can be calculated using equation (3.2) for values of n outside this range

[4]. As n ranges through the integer values, the set $x[n]$ repeats itself with period N [8]. Mathematically, this can be represented as follows [4]:

$$x[n] = x[n+N] \qquad \text{for all } n \tag{3.21}$$

$$X[k] = X[k+N] \qquad \text{for all } k \tag{3.22}$$

Note that the equations (3.21) and (3.22) are the results of using a complex set of input data. When the input data set is complex, two samples are required for each input coefficient $x[n]$ (one for the real part and one for the imaginary part). Thus, the sampling rate must be two times the Nyquist rate (four times f_{max}). This is assuming that acquiring both the real and imaginary parts requires two samples.

In many situations the input data set is purely real. By Shannon's sampling theorem, discussed in chapter 2, the sampling frequency should be a minimum of the Nyquist rate, or 2*f_{max}. The DTFS can only determine frequencies unambiguously from zero to half the sampling rate. Outside this range, frequencies are not determined accurately by the DTFS. Frequencies outside this range appear to be within this range (aliased) and distort the resulting spectral content. The effect of this is discussed in section (5.1).

In direct consequence of the circularity property discussed in section (3.4), only half of the output coefficients from the DTFS contain unique information. The outputs of the DTFS are symmetric about half the sampling rate, or mathematically stated as

$$X[k] = \begin{cases} X[N-k], & \text{for } 1 \le k \le N\text{ -}1 \\ X[k], & k = 0 \end{cases} \tag{3.23}$$

Therefore, from zero to half the sampling rate, frequencies are unambiguously determined by the DTFS [2]. Frequencies from half the sampling rate plus one to the sampling rate are only a reflection of the first half of DTFS coefficients where k ranges from 0 to $\left(\frac{N}{2}-1\right)$. In other words, the last half of the DTFS coefficients can be acquired from computing only the first half of DTFS coefficients. As will be discussed in chapter 4, the fast Fourier transform coefficients cannot be computed one at a time as with the DTFS coefficients.

3.5 Zero Padding

Although the DTFS can be evaluated over any number of N samples, many fast Fourier transform (FFT) algorithms require a specific number of samples. In many circumstances, the length (N) of the input data set $x[n]$ is smaller than the number of samples required by the FFT algorithm being used. In this case, zeroes can be added to the end of the input data set in order to obtain the required length. For example, if the length N is not a power of two but is required to be by the FFT algorithm, then zeroes can be appended to the data set until it is a power of two. By doing this, it is possible to analyze multiple input data sets of different lengths by using the same FFT algorithm. A higher frequency resolution is also achieved by zero-padding the input data set. As seen in equation (3.17), by increasing the length of the input data set, the bin spacing (the distance between adjacent frequency components) is decreased.

The process of zero padding is not impeccable as it causes the center frequencies of each DTFS coefficient to differ from the center frequencies of the DTFS coefficients calculated using the original length-N input data set. This is due to the fact of the different bin spacing achieved by different length N input data sets. Thus, the DTFS or FFT response of zero padded input data sets will differ slightly than that of the unpadded input data set. This effect can be minimized by using the window (or weighting) functions discussed in chapter 5 [2].

3.6 Strengths of the DTFS

3.6.1 Analyzing Periodic Signals

The DTFS is used to convert a time-domain signal into the frequency domain. The DTFS coefficients represent specific frequencies from zero to the sampling rate spaced equally by the bin spacing. This implies that the DTFS coefficients represent sinusoids that go through 0 to N-1 full cycles over the course of the N samples. Also note that, any sum of these frequencies must also repeat itself an integer number of times over the course of the N samples. In that sense, the DTFS is an excellent tool for analyzing signals that are periodic an integer number of times during the course of the N samples.

Even if the input signal is not periodic such as a frequency-shift-keyed (FSK) signal, the corresponding DTFS output does give coefficients that can be used to reconstruct the original signal in the time-domain. These corresponding frequencies in the DTFS output, though, are not the actual frequencies in the original signal [2].

3.6.2 Input Flexibility

The DTFS can accept either a purely real, purely imaginary, or complex input data set. The complex exponential in equation (3.1) guarantees that the coefficients of the DTFS output will be complex. This in turn, guarantees that if an input data set is purely real, then the output data set will be complex and allow for both magnitude and phase information to be calculated using equations (3.5) and (3.6).

3.6.3 Sets of Data

Consider an input data set of samples 0 to N-1. This data set can be divided into smaller data sets, and the DTFS can be calculated over each set. The data sets can be continuous or overlapping.

This also allows for a specific portion of a time-domain signal to be selected for DTFS analysis while the rest of the time-domain information is not needed. For example, if a radar-return pulse is received, then only the portion of the time-domain signal in which the pulse is present is important. This portion can be selected for spectral analysis while the time-domain information between pulses is discarded.

3.6.4 Coherent Integration Gain

As discussed in section (3.2), some texts place the normalization constant in front of the summation in equation (3.2) instead of equation (3.1). By doing this, the first aspect of coherent integration gain is observed. Performing the DTFS on a given input data set will give the output DTFS coefficients. These coefficients will lend themselves to amplitudes that are N times greater than the actual amplitudes of the sinusoids that make up the original time-domain signal [2].

For example, consider a 1-volt DC signal that is sampled 16 times ($N = 16$). Solving for the DC component $X[0]$, k is zero and the complex exponential is equal to 1. Using equation (3.1) without the normalization constant in front of the summation, it is found that $X[0]$ is equal to 16. Clearly, this is N times as large as the actual amplitude of the original DC signal.

Note that the normalization constant N, does not affect the phase spectrum calculated from the DTFS coefficients using equation (3.6). The reason is due to the fact that the phase is calculated using the arctangent of the ratio of the imaginary portion (b) to the real portion (a). Therefore, if both the imaginary and real parts are scaled by the same factor, then the ratio of b to a remains the same, and the phase is unaltered.

The second aspect of coherent integration gain deals with the signal-to-noise ratio. For best performance, the original signal is conditioned before it is sampled using *anti-aliasing* filters in order to remove all of the frequency content above half the sampling frequency (for purely real signals). As discussed in section (3.4.6), frequency components outside the range of zero to half the sampling frequency actually appear to be within that range after the DTFS is performed [2]. Anti-alias filtering also restricts the *noise bandwidth* to half the sampling frequency. The DTFS divides this input noise bandwidth by N (the number of output coefficients or frequency bins) because each frequency bin acts as a bandpass filter. As a result, each output coefficient can only have $1/N$ of the noise power on the input. Since additive white gaussian noise has a frequency spectrum that is continuous and uniform across the available frequency band, the noise bandwidth of each DTFS output is $1/N$ of the input noise bandwidth. As a corollary, the signal-to-noise ratio of a signal sine wave plus white noise is increased by a factor of N at the DTFS output [2].

3.7 Weaknesses of the DTFS

3.7.1 Computational Requirements

The principle weakness of the DTFS is the amount of computation required to perform it. The computational load of performing the DTFS directly can be evaluated by analyzing equation (3.1) directly. Evaluation of a single DTFS coefficient requires N-1 complex additions and N complex multiplications. Therefore, solving for all N DTFS coefficients requires $N(N\text{-}1)$, or $N^2\text{-}N$, complex additions and $N(N)$, or N^2, complex multiplications [1]. Since each complex addition requires two real additions and each complex multiplication requires four real multiplies and two real additions, the actual computational load for the DTFS is defined as follows [2]:

Number of real additions = $N(2(N\text{-}1) + 2(N)) = 4N^2\text{-}2N$ (3.24)

Number of real multiplications = $N(4N) = 4N^2$ (3.25)

From equations (3.24) and (3.25), the computational load grows quickly as the size of the data set (N) increases. For example, performing the DTFS on a data set of 1024 samples requires 4,192,256 complex additions and 4,194,304 complex multiplications, while performing the DTFS on a data set of twice as many samples requires 16,773,120 complex additions and 16,777,216 complex multiplications.

3.7.2 Quantization Noise Error

As discussed in section (2.7), there is quantization error involved with quantizing a sampled signal. The maximum error associated with uniform quantization is defined by equation (2.23). In section (2.7) the quantization error was represented by the variable X. In this chapter the variable X is used to represent the DTFS coefficients. Therefore in this section, to avoid confusion, let the quantization error be represented by the variable q. Also, let $q[n]$ represent the quantization error associated with the n^{th} sample and let $\hat{x}[n]$ equal the actual value of the signal at the time of the n^{th} sample. Now equation (3.1) can be rewritten as follows:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1}(\hat{x}[n] + q[n])e^{-j\Omega_0 kn} = \frac{1}{N}\sum_{n=0}^{N-1}\hat{x}[n]e^{-j\Omega_0 kn} + \frac{1}{N}\sum_{n=0}^{N-1}q[n])e^{-j\Omega_0 kn} \qquad (3.26)$$

The second term in the right-hand side of equation (3.26) is the error introduced to the calculation of each DTFS coefficient due to the quantization error in each sample. This error is rewritten as equation (3.27) below:

$$Q[k] = \frac{1}{N}\sum_{n=0}^{N-1} q[n]e^{-j\Omega_0 kn} \tag{3.27}$$

It becomes evident that as the size of the input data set increases, the error in each DTFS coefficient due to quantization noise is increased. Since FFT algorithms reduce the number of computations required to calculate the DTFS coefficients, this error is reduced [2].

3.7.3 High Sidelobes

Each of the DTFS output coefficients acts as a bandpass filter where the frequency corresponding to that certain coefficient represents the center frequency of the bandpass filter. The frequency responses of these DTFS filters does not represent ideal bandpass filters. Instead, the frequency response is that of a sinc function where the lobe closest to the main lobe is only 13 dB below that of the main lobe. As a result, if a time-domain function has a significant sinusoidal component that is not centered exactly on the center frequency of the DTFS filter, then that corresponding DTFS coefficient will still show significant amplitude. Using window (weighting) functions can alter this effect. This will be discussed in chapter 5.

3.7.4 Spectral Leakage

As discussed in section (3.7.3), each DTFS coefficient acts as a bandpass filter. If a sinusoidal component lies exactly at the center frequency of a specific DTFS filter, then that frequency component will have a coherent integration gain of *N* (using equation 3.1 without the normalization constant). If the sinusoidal component does not lie exactly at the center frequency of a specific bandpass filter, then the coherent integration gain will be less than *N*. The reason for this is the non-ideal frequency response of the DTFS bandpass filters. *Frequency straddle loss* is the difference between maximum coherent integration gain (N) and the actual coherent integration gain achieved by a sinusoidal component [2].

Also, as a sinusoidal component's frequency gets closer to the halfway mark between two adjacent filters, it will begin to contribute to both DTFS coefficients corresponding to those filters. Frequency straddle loss is greatest when a signal lies directly in the middle of two adjacent frequency bins. The sinusoid's energy will be attenuated and distributed between the two DTFS coefficients [2]. This is called *spectral leakage* because some of the energy from one frequency component leaks into the adjacent frequency bins.

3.7.5 Analyzing Transient Signals

As shown in table (3.1), the DTFS is ideal for analyzing time-domain signals that are periodic over the N samples. Signals that are non-periodic, or transient signals, are poorly analyzed by the DTFS. This is also true for transient sinusoids, signals that are periodic but change frequencies such as the FSK signal discussed in section (3.6.1) [2]. Ways of dealing with or minimizing the effects of these transients will be discussed in chapter 5 where spectral leakage also comes into play.

Notice from table (3.1) that the DTFT is better suited for analyzing non-periodic signals in the time-domain. Also, note that the DTFS is the only Fourier representation that can be computed numerically by a computer or digital system. As a result, the DTFS must be related to the DTFT in order to analyze non-periodic signals in the time-domain [1].

3.8 Summary

The DTFS, often referred to as the discrete Fourier transform (DFT), is an extremely useful tool for analyzing periodic signals. The DTFS can be used to analyze non-periodic time-domain signals, but the user must understand the implications of doing so and be able to interpret the results.

Given a purely real input data set, the DTFS can accurately resolve frequency components from zero to half the sampling rate. With a fixed sampling frequency, the frequency resolution achieved by the DTFS is improved as the number of samples is increased. With a fixed number of samples, the frequency resolution is improved by decreasing the sampling frequency.

The weaknesses of the DTFS described in section (3.7) give motivation for using a fast Fourier transform (FFT) algorithm, which will be discussed in chapter 4. These algorithms maintain the strengths of the DTFS while diminishing the weaknesses. The FFT algorithms are not approximations to the DTFS, but are more efficient methods in evaluating the DTFS [2].