

system, as is the case with much popular music. But such losses are evident in musical material that exploits the full range of a fine audio system. It is not difficult to generate signals that reveal the weaknesses of the commercial coding models. The degradation is more pronounced when these signals are “copied” (or to be more precise, recoded) or further manipulated.

For creative purposes, we prefer data reductions that leave the analysis data in editable form. The literature of computer music includes a large body of research work on data reduction, including pioneering studies by Risset (1966), Freedman (1967), Beauchamp (1969, 1975), and Grey (1975). Techniques that have been used in computer music include line-segment approximation, principal components analysis, spectral interpolation synthesis, spectral modeling synthesis, and genetic algorithms.

Theory of Fourier Analysis

In order to comprehend many of the transformations presented later in this chapter, it is important to have a basic understanding of the theory of Fourier analysis. This section presents a capsule history and the essential points of the theory.

History of Fourier Analysis

In the early eighteenth century, the French engineer and aristocrat Jean-Baptiste Joseph, Baron de Fourier (1768–1830), formulated a theory stating that arbitrary periodic waveforms could be deconstructed into combinations of simple sine waves of different amplitudes, frequencies, and phases. Through the middle of the nineteenth century, Fourier analysis was a tedious task of manual calculation. In the 1870s, the British physicist Lord Kelvin and his brother built the first mechanical harmonic analyzer (Marple 1987). This elaborate gear-and-pulley contraption analyzed handtraced waveform segments. The analyzer acted as a mechanical integrator, finding the area under the sine and cosine waves for all harmonics of a fundamental period. The Michelson-Stratton harmonic analyzer (1898) was probably the most sophisticated machine of this type. Designed around a spiral spring mechanism, it could resolve up to eighty harmonics. It could also act as a waveform synthesizer, mechanically inverting the analysis to reconstruct the input signal.

In the twentieth century, mathematicians refined Fourier's method. Engineers designed analog filter banks to perform simple types of spectrum analysis. Following the development of stored-program computers in the 1940s, programmers created the first digital implementations of the *Fourier transform* (FT), but these consumed enormous amounts of computer time—a scarce commodity in that era. Finally, in the mid-1960s, a set of algorithms known as the *fast Fourier transform* or FFT, described by James Cooley at Princeton University and John Tukey at Bell Telephone Laboratories, greatly reduced the voluminous calculations required for Fourier analysis (Cooley and Tukey 1965).

Fourier Series

Fourier showed that a periodic function $x(t)$ of period T can be represented by the infinite summation series:

$$x(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega t + \phi_n)$$

That is, the function $x(t)$ is a sum of harmonically related sinusoidal functions with the frequency $\omega_n = n\omega = 2\pi/T$. C_0 is the offset or DC component; it shifts the waveform up or down. The first sinusoidal component C_1 is the *fundamental*; it has the same period as T . The numerical variables C_n and ϕ_n give the magnitude and phase of each component.

A Fourier series summation is a formula for reconstructing or synthesizing a periodic signal. But it does not tell us how to set the coefficients C_n and ϕ_n for an arbitrary input sound. For this, we need the analysis method called the Fourier transform.

Fourier Transform

This section takes advantage of the complex exponential representation of a sine wave at a given phase. This representation is based on these identities:

$$\cos(2\pi f + \phi) = \cos(2\pi f) + j \sin(2\pi f) = e^{j2\pi f}$$

So, a cosine at a given frequency and phase can also be represented as a complex number, or a complex exponential function. (See Roads 1996, appendix A.)

Suppose that we wish to analyze a continuous-time (analog) signal $x(t)$ of infinite extent and bandwidth. Fourier's theory says that $x(t)$ can be accurately reconstructed with an infinite number of pure sinusoidal waves of different amplitudes, frequencies, and initial phases. These waves make up the signal's Fourier transform spectrum. The FT spectrum represents all frequencies from 0 Hz (a constant) to infinity (∞) Hz, with a mirror image in the negative frequencies.

The formula for the FT or *Fourier integral* is as follows:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

This says that the FT at any particular frequency f is the integral of the multiplication of the input signal $x(t)$ by the pure sinusoid $e^{-j2\pi ft}$. Intuitively, we could surmise that this integral will be larger when the input signal is high in amplitude and rich in partials. $X(f)$ represents the magnitude of the Fourier transform of the time-domain signal $x(t)$. By magnitude we mean the absolute value of the amplitude of the frequencies in the spectrum. The capital letter X denotes a Fourier transform, and the f within parentheses indicates that we are now referring to a frequency-domain signal, as opposed to the time-domain signal $x(t)$. Each value of $X(f)$ is a complex number.

The magnitude is not a complete picture of the Fourier transform. It tells us just the amount of each complex frequency that must be combined to synthesize $x(t)$. It does not indicate the phase of each of these components. One can also plot the *phase spectrum*, as it is called, but this is less often shown.

The magnitude of the Fourier transform $X(f)$ is symmetric around 0 Hz. Thus the Fourier representation combines equal amounts of positive and negative frequencies. This is the case for any real-valued input signal. This dual-sided spectrum has no physical significance. (Note that the inverse Fourier transform takes a complex input signal—a spectrum—and generates a real-valued waveform as its output.)

The Discrete Fourier Transform

The one kind of signal that has a discrete frequency-domain representation (i.e., isolated spectral lines) is a periodic signal. A periodic signal repeats at every interval T . Such a signal has a Fourier transform containing components at a fundamental frequency ($1/T$) and its harmonics and its zero everywhere else.

A periodic signal, in the precise mathematical sense, must be defined from $t = -\infty$ to $t = \infty$. Colloquially, one speaks of signals as periodic if $x(t) = x(t + T)$ for an amount of time that is long relative to the period T . We can construct this kind of periodic signal by replicating a finite-length signal. Imagine that we infinitely replicate the finite-length signal $x(t)$ backwards and forwards in time. In the discrete-time (sampled) domain, this produces a periodic signal $x[n]$. The use of brackets rather than parentheses indicates that the signal is discrete, rather than continuous.

The frequency-domain representation of this replicated periodic signal $x[n]$ is called its *discrete Fourier transform* (DFT). The DFT provides a sampled look at both the magnitude and phase of the spectrum of $x[n]$, and is a central tool in musical signal processing. In effect, the DFT sets up a one-to-one correspondence between the number of input samples N and the number of frequencies that it resolves.

The Short-Time Fourier Transform

To adapt Fourier analysis to the practical world of sampled time-varying signals, researchers molded the FT into the short-time Fourier transform or STFT (Schroeder and Atal 1962; Flanagan 1972; Allen and Rabiner 1977; Schafer and Rabiner 1973).

Windowing the Input Signal

As a preparation for spectrum analysis, the STFT imposes a window upon the input signal. A window is nothing more than a simple amplitude envelope. Windowing breaks the input signal into a series of segments that are shaped in amplitude by the chosen window function and bounded in time according to the length of the window function. In audio applications, the duration of the window is usually in the range of 1 ms to 100 ms, the window envelope is bell-shaped, and the segments usually overlap. By analyzing the spectrum of each windowed segment separately, one obtains a sequence of measurements that constitute a time-varying spectrum.

Unfortunately, windowing has the side effect of distorting the spectrum measurement. This is because the spectrum analyzer is measuring not purely the input signal, but rather, the product of the input signal and the window.

The spectrum that results is the convolution of the spectra of the input and the window signals. We see the implications of this later.

Operation of the STFT

Adopting Dolson's (1986) notation, the equation for a DFT of an input signal $x[m]$ multiplied by a time-shifted window $h[n - m]$ is as follows:

$$X[n, k] = \sum_{m=-\infty}^{\infty} \{x[m]h[n - m]\}e^{-j(2\pi/N)km}$$

Thus the output $X[n, k]$ is the Fourier transform of the windowed input at each discrete time n for each discrete frequency band or bin k . The equation says that m can go from minus to plus infinity; this is a way of saying "for an arbitrary-length input signal." For a specific short-time window, the bounds of m are set to the appropriate length. Here, k is the index for the frequency bins, N is the number of points in the spectrum. The following relation sets the frequency corresponding to each bin k :

$$f_k = (k/N) \times f_s$$

where f_s is the sampling rate. So for a sampling rate of 44.1 kHz, an analysis window length N of 1024 samples, and a frequency bin $k = 1$, f_k is 43 Hz. The windowed DFT representation is particularly attractive because the fast Fourier transform or FFT can calculate it efficiently.

A discrete STFT formulation indicating the *hop size* or time advance of each window is:

$$X[l, k] = \sum_{m=0}^{M-1} h[m]x[m + (lH)]e^{-j(2\pi/N)km}$$

where M is the number of samples in the input sequence, $h[m]$ is the window that selects a block of data from the input signal $x[m]$, l is the *frame index*, and H is the hop size in samples (Serra 1989).

Each block of data generated by the STFT is called a *frame*, by analogy to the successive frames of a film. Each frame contains two spectra: (1) a magnitude spectrum that depicts the amplitude of every analyzed frequency component (figure 6.1c), and (2) a phase spectrum that shows the initial phase value for every frequency component (figure 6.1d). We can visualize each of these two spectra as histograms with a vertical line for each frequency component along the abscissa. The vertical line represents amplitude in the case of the magnitude

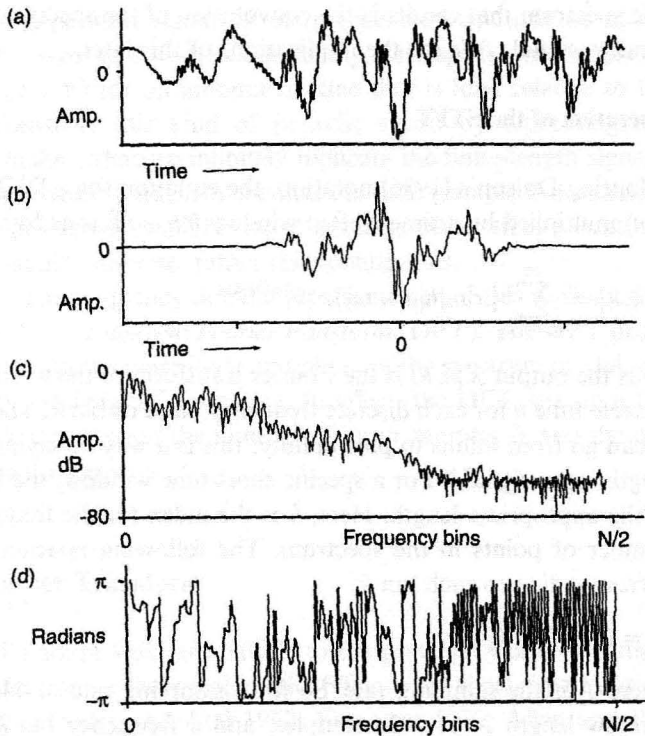


Figure 6.1 Magnitude and phase spectra. (a) Input waveform. (b) Windowed segment. (c) Magnitude spectrum plotted over the range 0 to -8 dB. (d) Phase spectrum plotted over the range $-\pi$ to π . (After Serra 1989.)

spectrum, and the starting phase (between $-\pi$ and π) in the case of the phase spectrum. The magnitude spectrum is relatively easy to read, the phase spectrum less so. When normalized to the range of $-\pi$ and π it is called the *wrapped phase* representation. For many signals, it appears to the eye like a random function. An *unwrapped phase* projection may be more meaningful visually. (See Roads 1996, appendix A.)

To summarize, applying the STFT to a stream of input samples results in a series of frames that make up a time-varying spectrum.

Justifications for Windowing

Theory says that we can analyze a segment of any length and exactly resynthesize the segment from the analysis data. For example, we can analyze in one

pass Stravinsky's *Le sacre du printemps* using a thirty-minute-long window, and reconstruct the entire piece from this analysis. This being the case, why bother to break the analysis into overlapping windows on a micro time scale?

The reasons are several. The analysis of a thirty-minute monaural sound sampled at 44.1 kHz would result in a spectrum of over seventy-nine million points. A visual inspection of this enormous spectrum would eventually tell us all the frequencies that occurred over a half hour, but would not tell us when they occurred. This temporal information is embedded deep in the mathematical combination of the magnitude and phase spectra, hidden to the eye. Thus the first thing that windowing helps with is the visualization of the spectrum. By limiting the analysis to micro segments (typically less than a twentieth of a second), each analysis plots fewer points, and we know more accurately when these frequencies occurred.

A second reason for using short-time windows is to conserve memory. Breaking the input into micro segments makes it easy to calculate the FFT in a limited memory space.

A third reason for short-time windows is that one obtains results more quickly. For *Le sacre du printemps* one would have to wait up to thirty minutes just to read in the input signal, plus however long it takes to calculate an FFT on a seventy-nine million point input signal. Windowing the input lets one obtain initial results quickly—after reading just a few milliseconds of the input. This opens up applications for real-time spectrum analysis.

Analysis Frequencies

One can think of the STFT as the application of a bank of filters at equally spaced frequency intervals to the windowed input signal. The frequencies are spaced at integer multiples (i.e., harmonics) of

$$\frac{\text{sampling frequency}}{N}$$

where N is the size of the analyzed segment. (As we will later see, the value of N is usually greater than the actual number of sound samples analyzed; for now we will assume they are the same length.) Thus if the sampling frequency is 50 kHz and the window length is one thousand samples, the analysis frequencies are spaced at intervals $50,000/1000 = 50$ Hz apart, starting at 0 Hz. The analyzer at 0 Hz measures the direct current or DC offset of the signal, a

constant that can shift the entire signal above or below the center point of zero amplitude.

Audio signals are bandlimited to half the sampling rate (25 kHz in this case) and so we are concerned with only half of the analysis bins. The effective frequency resolution of an STFT is thus $N/2$ bins spread equally across the audio bandwidth, starting at 0 Hz and ending at the Nyquist frequency. In our example, the number of usable audio frequency bins is five hundred, spaced 50 Hz apart.

Time-Frequency Uncertainty

The knowledge of the position of the particle is complementary to the knowledge of its velocity or momentum. If we know the one with high accuracy we cannot know the other with high accuracy. (Heisenberg 1958)

All windowed spectrum analyses are hampered by a fundamental uncertainty principle between time resolution and frequency resolution. This is directly analogous to a principle first recognized by quantum physicists such as Werner Heisenberg in the early part of the twentieth century. The *linear resolution principle* (Masri, et al 1997a) states that if we want high resolution in the time-domain (i.e., we want to know precisely when an event occurs), we sacrifice frequency resolution. In other words, we can tell that an event occurred at a precise time, but we cannot say exactly what frequencies it contained. Conversely, if we want high resolution in the frequency-domain (i.e., we want to know the precise frequency of a component), we sacrifice time resolution. This means that we can pinpoint frequency content only over a long time interval. It is important to grasp this fundamental relationship in order to interpret the results of Fourier analysis.

Fourier analysis starts from this abstract premise: if a signal contains only one frequency, then that signal must be a sinusoid that is infinite in duration. Purity of frequency—absolute periodicity—implies infinitude. As soon as one limits the duration of this sine wave, the only way that Fourier analysis can account for it is to consider the signal as a sum of many infinite-length sinusoids that just happen to cancel each other out in such a way as to result in a limited-duration sine wave! While this characterization of frequency neatens the mathematics, it does not jibe with our most basic experiences with sound. As Gabor (1946) pointed out, if the concept of frequency is used only to refer to infinitely long signals, then the concept of changing frequency is impossible.

Figure 6.2 shows the effects of time-frequency (TF) uncertainty at the juncture of an abrupt transition between two pure tones. Figure 6.2a portrays the actual spectrum of the signal fed into the analyzer. Figure 6.2b is the measured short-time Fourier transform of this signal. Notice the band-thickening and blurring, which are classic symptoms of TF uncertainty.

Time-Frequency Tradeoffs

The FFT divides the audible frequency space into $N/2$ frequency bins, where N is the length in samples of the analysis window. Hence there is a tradeoff between the number of frequency bins and the length of the analysis window. For example, if N is five hundred and twelve samples, then the number of frequencies that can be analyzed is limited to two hundred and fifty-six. Assuming a sampling rate of 44.1 kHz, we obtain two hundred and fifty-six bins equally spaced over the bandwidth 0 Hz to the Nyquist frequency 22.05 kHz. Increasing the sampling rate only widens the measurable bandwidth, it does not increase the frequency resolution of the analysis.

If we want high time accuracy (say 1 ms or about forty-four samples), we must be satisfied with only $44/2$ or twenty-two frequency bins. Dividing the audio bandwidth from 0 to 22.05 kHz by twenty-two frequency bins, we obtain $22,050/22$ or about 1000 Hz of frequency resolution. That is, if we want to know exactly when events occur on the scale of 1 ms, then our frequency resolution is limited to the gross scale of 1000-Hz-wide frequency bands. By sacrificing more time resolution, and widening the analysis interval to 30 ms, one can spot frequencies within a 33 Hz bandwidth. For high resolution in frequency (1 Hz), one must stretch the time interval to 1 second (44,100 samples)!

Because of this limitation in windowed STFT analysis, researchers are examining hybrids of time-domain and frequency-domain analysis, multiresolution analysis, or non-Fourier methods to try to resolve both dimensions at high resolution.

Frequencies in between Analysis Bins

The STFT knows only about a discrete set of frequencies spaced at equal intervals across the audio bandwidth. The spacing of these frequencies depends on the window size. This size corresponds to the “fundamental period” of the analysis. Such a model works well for sounds that are harmonic or quasi-harmonic where the harmonics align closely with the bins of the analysis. What

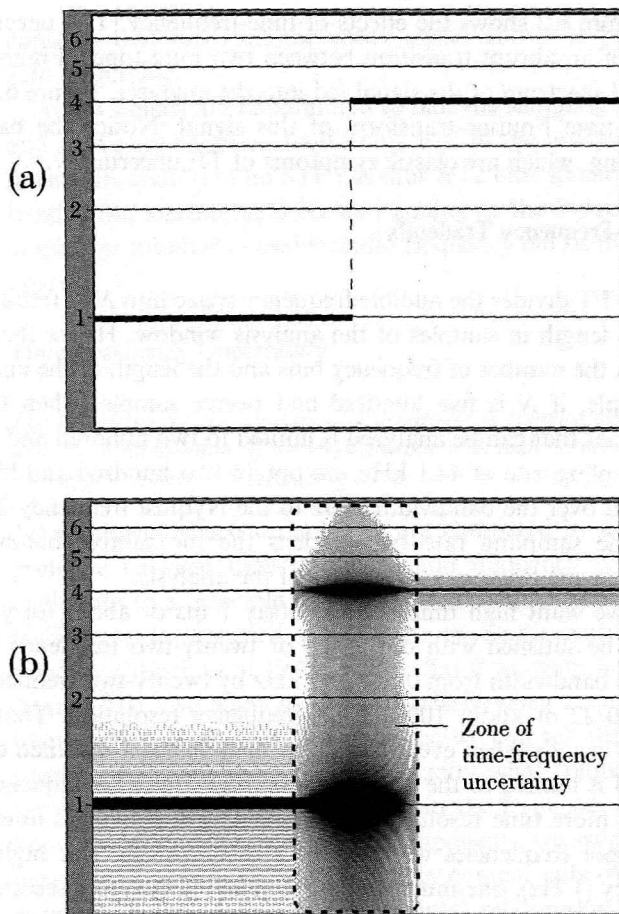


Figure 6.2 Time-frequency uncertainty in short-time Fourier analysis. (a) Idealized spectrum of the signal fed to the analyzer, consisting of a tone at 1000 Hz followed immediately by a tone at 4000 Hz. (b) Analysis sonogram. The blurring indicates the zone of time-frequency uncertainty. Both images are plotted on a logarithmic frequency scale from 500 Hz to 7000 Hz.

happens to frequencies that fall in between the equally spaced analysis bins of the STFT? This is the case for inharmonic sounds such as gongs or noisy sounds such as snare drums.

Let us call the frequency to be analyzed f . When f coincides with the center of an analysis channel, all its energy is concentrated in that channel, and so it is accurately measured. When f is close to but not precisely coincident with the center, energy leaks into all other analysis channels, but with a concentration remaining close to f . The leakage spilling into all frequency bins from components inbetween bins is a well-known source of unreliability in the spectrum estimates produced by the STFT. When more than one component is in between bins, *beating effects* (periodic cancellation and reinforcement) may occur in both the frequency and amplitude traces. The result is that the analysis shows fluctuating energy in frequency components that are not physically present in the input signal.

Significance of Clutter

If the signal is resynthesized directly from the analysis data, the extra frequency components and beating effects pose no problem. These effects are benign artifacts of the STFT analysis that are resolved in resynthesis. Beating effects are merely the way that the STFT represents a time-varying spectrum in the frequency-domain. In the resynthesis, some components add constructively and some add destructively (canceling each other out), so that the resynthesized result is a close approximation of the original.

Beating and other anomalies are harmless when the signal is directly resynthesized, but they obscure attempts to inspect the spectrum visually, or to transform it. For this reason, the artifacts of analysis are called *clutter*. Dolson (1983) and Strawn (1985) assayed the significance of clutter in analysis of musical instrument tones. Cross-term clutter is common in *higher-order analysis*, which can extract detailed phase and modulation laws embedded in the spectrum analysis (Masri, et al. 1997a, 1997b).

The Phase Vocoder

The *phase vocoder* (PV) uses the STFT to convert an audio signal into a complex Fourier representation. Since the STFT calculates the frequency domain representation of the signal on a fixed frequency grid, the actual frequencies of

the partial bins have to be found by converting the relative phase change between two STFT outputs to actual frequency changes. The term “phase” in phase vocoder refers to the fact that the temporal development of a sound is contained in its phase information, while the amplitudes denote that a specific frequency component is present in a sound. The phase contains the structural information (Sprenger 1999). The phase relationships between the different bins reconstruct time-limited events when the time-domain representation is resynthesized. The phase difference of each bin between two successive analysis frames determines that bin’s frequency deviation from its mid frequency. This provides information about the bin’s true frequency, and makes possible a resynthesis on a different time basis.

Phase Vocoder Parameters

The quality of a given PV analysis depends on the parameter settings chosen by the user. These settings must be adjusted according to the nature of the sounds being analyzed and the type of results that are expected. The main parameters of the PV are:

1. Window size (also called frame size)—number of input samples to be analyzed at a time.
2. FFT size—the actual number of samples fed to the FFT algorithm; usually the nearest power of two that is double the window size, where the unit of FFT size is referred to by *points*, as in a “1024-point FFT.”
3. Window type—selection of a window shape from among standard types.
4. Hop size or overlap factor—time advance from one window onset to the next.

Next we discuss each parameter in turn. Later we give rules of thumb for setting these parameters.

Window Size

The window size (in samples) determines one aspect of the tradeoff in TF resolution. The larger the window is, the greater the number of frequency bins, but the lower the time resolution, and vice versa. If we are trying to analyze sounds in the lower octaves with great frequency accuracy, we cannot avoid a large window size. Since the FFT computes the average spectrum content within a

given window, the precise onset time of any spectrum changes within the span of the window is lost when the spectrum is plotted or transformed. (If the signal is simply resynthesized, the temporal information is restored.) For high-frequency sounds, small windows are adequate, which are also more accurate in time resolution.

FFT Size and Hop Size

The FFT size is typically the nearest power of two that is double the window size. For example, a window size of 512 samples would mandate an FFT size of 1024. The other 512 samples in the FFT are set to zero—a process called *zero-padding*.

The *hop size* is the number of samples that the analyzer jumps along the input waveform each time it takes a new spectrum measurement. The shorter the hop size, the more successive windows overlap. This improves the resolution of the analysis, but requires more computation. Some PVs specify hop size as an overlap factor that describes how many analysis windows cover each other. An overlap of four, for example, means that one window follows another after 25% of the window length. Regardless of how it is specified, the hop size is usually a fraction of the window size. A certain amount of overlap (e.g., eight times) is necessary to ensure an accurate resynthesis. More overlap may improve accuracy when the analysis data is going to be transformed, but the computational cost is proportionally greater.

Window Type

A spectrum analyzer measures not just the input signal but the product of the input signal and the window envelope. The law of convolution, introduced in chapter 5, states that multiplication in the time-domain is equivalent to convolution in the frequency-domain. Thus the analyzed spectrum is the convolution of the spectra of the input and the window signals. In effect, the window modulates the input signal, and this introduces sidebands or clutter into the analyzed spectrum.

A smooth bell-shaped window minimizes the clutter. Most PVs let the user select a window from a family of standard window types, including Hamming, Hanning (or Hann; see Marple 1987), truncated Gaussian, Blackman-Harris, and Kaiser (Harris 1978; Nuttall 1981). All are bell-shaped, and all work reasonably well for general musical analysis-resynthesis. Each one is slightly

different, however, so it may be worth trying different windows when the results are critical. The one window to avoid is the rectangular or Dirichelet, which introduces a great deal of clutter or extraneous frequency components into the analyzed spectrum.

Typical PV Parameter Settings

No parameter settings of the PV are ideal for all sounds. Within a certain range, however, a variety of traditional instrumental sounds can be analyzed and resynthesized with reasonable fidelity. Here are some rules of thumb for PV parameter settings that may serve as a starting point for more tuned analyses:

1. Window size—large enough to capture four periods of the lowest frequency of interest. This is particularly important if the sound is time-stretched; too small a window size means that individual pitch bursts are moved apart, changing the pitch, although formants are preserved.
2. FFT size—double the window size, in samples.
3. Window type—any standard type except Dirichelet.
4. Hop size—Time advance of the analysis window. If the analysis data is going to be time-distorted, the recommended hop size is an eighth of the frame size, in samples (i.e., eight times overlap). The minimum technical criterion is that all windows add to a constant, that is, all data is equally weighted. This typically implies an overlap at the -3 dB point of the particular window type chosen, from which can be derived the hop size.

Any given setting of the window size results in an analysis biased toward harmonics of the period defined by that window size. Frequency components that fall outside the frequency bins associated with a given window size will be estimated incorrectly. Some analyzers try to estimate the pitch of the signal in order to determine the optimal window size. This is called *pitch-synchronous analysis* (Mathews, Miller, and David 1961). Pitch-synchronous analysis works well if the sound to be analyzed has a basically harmonic structure.

Resynthesis Techniques

Resynthesis constructs a time-domain signal from the analysis data. If the analysis data has not been altered, then the resynthesis should be a close simulacrum of the original signal. If the analysis data has been altered, the resyn-

thesized sound will be transformed. A variety of resynthesis techniques have been invented. Some are more efficient, some are more accurate, some are more robust under transformation, some are adapted for real time operation. This section presents three techniques. The first two appear in commonly available phase vocoders. The third is more experimental, but gives an idea of the optimizations that can be made.

Overlap-Add Resynthesis

To resynthesize the original time-domain signal, the STFT can reconstruct each windowed waveform segment from its spectrum components by applying the *inverse discrete Fourier transform* (IDFT) to each frame. The IDFT takes each magnitude and phase component and generates a corresponding time-domain signal with the same envelope as the analysis window. Then by overlapping and adding these resynthesized windows, typically at their half power or -3 dB points, one obtains a signal that is a close approximation of the original. This is called the *overlap-add* (OA) method of resynthesis.

We use the qualification “close approximation” as a way of comparing practical implementations of the STFT with mathematical theory. In theory, resynthesis from the STFT is an identity operation, replicating the input sample by sample (Portnoff 1976). If it were an identity operation in practice, we could copy signals through an STFT/IDFT any number of times with no generation loss. However, even good implementations of the STFT lose a small amount of information. This can be verified by a careful comparison between the input and output waveforms. The loss may not be audible after one pass without transformation through the STFT.

Many microsonic transformations manipulate the analysis frames before resynthesizing the sound with the OA method. The OA process, however, is designed for cases where the windows sum perfectly to a constant. As Allen and Rabiner (1977) showed, any additive or multiplicative transformations that disturb the perfect summation criterion at the final stage of the OA cause side effects that will probably be audible. Time expansion by stretching the distance between windows, for example, may introduce comb filter or reverberation effects, depending on the window size used in the analysis. Using speech or singing as a source, some transformations result in voices with robotic or ringing artifacts. One way to lessen unwanted artifacts is to stipulate a great deal of overlap among successive windows in the analysis stage. In selected cases, the distortion introduced by OA resynthesis can be exploited as a sonic effect.

Oscillator Bank Resynthesis

Oscillator bank (OB) resynthesis differs from the overlap-add approach. In contrast to the OA model, which sums the sine waves at each frame, OB resynthesis converts the analysis data from all analyzed frames into a set of amplitude and frequency envelopes for multiple oscillators. In effect, the envelopes convert the analysis data from the micro time scale to the time scale of the analyzed sound.

The advantage of OB resynthesis is that envelopes are much more robust under musical transformation than the spectrum frames. The perfect summation criterion of the OA model does not apply in OB resynthesis. Within broad limits, one can stretch, shrink, rescale, or shift the envelopes without worrying about artifacts in the resynthesis process. Another strength is that the OB representation facilitates graphical editing of the spectrum. A disadvantage of OB is that it is not as efficient computationally as OA methods.

Analysis-by-Synthesis/Overlap-Add Resynthesis

Analysis-by-synthesis/overlap-add (AS/OA) is an adaptive method designed for improved resolution and more robust transformations. AS/OA incorporates an error analysis procedure (George and Smith 1992). This procedure compares the original signal with the resynthesized signal. When the error is above a given threshold, the procedure adjusts the amplitudes, frequencies, and phases in the analysis frame to approximate the original more closely. This adaptive process can occur repeatedly until the signal is more-or-less precisely reconstructed. As a result, the AS/OA method can handle attack transients, inharmonic spectra, and effects such as vibrato with greater accuracy than the OA method. It also permits more robust musical transformations. Another method called the tracking phase vocoder, presented later in the chapter, has similar benefits.

Assessment of the Phase Vocoder

Present methods of windowed spectrum analysis owe a debt to Dennis Gabor's pioneering work (1946, 1947, 1952; see also chapter 2). The frames of the STFT are analogous to his acoustical quanta. The projection of the time-frequency plane onto the sonogram is analogous to a visual representation of the Gabor matrix.

The phase vocoder has emerged from the laboratory to become a popular tool. It is packaged in a variety of widely distributed musical software. The compositional interest of the PV lies in transforming the analysis data before resynthesis, producing variations of the original sound. What the composer seeks in the output is not a clone of the input, but a musical transformation that maintains a sense of the source's identity.

The weaknesses of the STFT and the PV as representations for sound are well known. The uncertainty principle pointed out by Gabor is embedded deeply within the STFT. Time-frequency information is smeared. Overlapping windows mean that it is impossible to modify a single time-frequency atom without affecting adjacent atoms. Such a change will most likely lead to a discontinuity in the resynthesized signal. Many transformations sound "blurry" or "sinusoidal" in quality, a common artifact of Fourier techniques in general. The tracking phase vocoder, described later, is a more secure transformation tool, but it has its own imperfections.

On the positive side, the PV is powerful. Good implementations of the PV offer the possibility of modifying pitch, time, and timbre independently.

Sound Transformation with Windowed Spectrum Operations

Inside a windowed spectrum analyzer is a granulated time-frequency representation of sound. By manipulating this representation, we can obtain many transformations that are difficult or impossible to achieve with time-domain procedures, including high-quality pitch-time changing, frequency-domain filtering, stable and transient extraction, multiband dynamics processing, cross-synthesis, and many other exotic effects. This section explores these techniques, many of which are implemented within phase vocoders.

Pitch-Time Changing

One of the most common windowed spectrum transformations is the altering of a signal's duration whilst maintaining its original pitch. Inversely, one can change pitch without altering duration. For these effects, the phase vocoder often achieves better sound quality than can be obtained with the time-domain granulation algorithms described in chapter 5.

In a PV with overlap-add resynthesis, the time stretching/shrinking algorithm moves the onset times of the overlapping frames farther apart (when stretching)

or closer together (when shrinking) in the resynthesis. For the smoothest transpositions, the PV should multiply the phase values by the same constant used in the time base changing (Arfib 1991).

Pitch-shifting alters the pitch without changing the time base. Pitch-transposition is a matter of scaling the frequencies of the resynthesis components. For speech signals in particular, however, a constant scale factor changes not only the pitch but also the formant frequencies. For upward shifts of an octave or more, this reduces the speech's intelligibility. Thus Dolson (1986) suggested a correction to the frequency scaling that reimposes the original spectral envelope on the transposed frequency spectrum. If the original spectrum had a formant at 2 kHz, for example, then so will the transposed version.

Frequency-Domain Filtering

Spectrum filters operate in the frequency domain by rescaling the amplitudes of selected frequency bins. Some of their controls are similar to traditional time-domain filters, such as center frequency, bandwidth, and gain or boost in dB.

Other controls apply to the windowed analysis, such as the window size, FFT size, and overlap factor. These controls affect the efficiency and quality of the analysis-resynthesis process. For example, longer windows and FFTs generally result in more pronounced filtering effects.

Differences between time-domain filters and spectral filters show up when the bandwidths are narrow and the filter Q is high. The spectral filter breaks down a broadband signal into individual sinusoidal components. A tell-tale "breebles" artefact may be heard, as individual components pop in and out. Breebles are characteristic of manipulations on windowed Fourier analyses in general, and appear in a number of other PV transformations.

Another approach to frequency-domain filtering provides a graphic interface, in which the user sees a sonogram display of the sound. The software provides a palette of drawing tools that let users erase or highlight selected regions of the sonogram image. The sound is then resynthesized on the basis of the altered sonogram image. (See the later section on sonographic transformations.)

Stable and Transient Extraction

This is a class of transformations that sorts audio waveforms on a micro time scale into two categories: stable and transient. Spoken vowels, for example, are relatively stable frequencies compared to the transient frequencies in conso-

nants. Once these frequencies are separated, the signals can be further manipulated individually.

In Erbe's (1995) implementation of stable and transient extraction, the user specifies:

1. Number of bands in the analyzer
2. Number of frames to analyze at a time
3. Frequency threshold for the transient part of the signal
4. Frequency threshold of the stable part of the signal

For example, the transient part could be specified as changing more than 30 Hz per frame, while the stable part is specified as changing less than 5 Hz per frame. Erbe's extraction algorithm takes the average of the change in instantaneous frequency over several FFT frames. If this average is greater than the stipulated value for transient information, the amplitude and phase from the source is assigned to the transient spectrum. Similarly, if the average change is less than the stipulated value for stable information the amplitude and phase from the source is assigned to the stable spectrum. Note that if the transient and stable thresholds are not identical, this leaves behind a part of the spectrum that is between the two—neither stable nor transient.

Another approach to stable and transient extraction is via *spectral tracing* (Wishart 1994; Norris 1997). Spectral tracing analyzes a sound and retains only the loudest or softest $N\%$ of the partials in the spectrum. To extract the transient part of a spoken voice, one retains only the softest 1% of the analyzed spectra. The sound quality of this 1% after the result is high-pass filtered, is like noisy whispering.

Dynamic Range Manipulations

Spectrum analysis makes it possible to manipulate the dynamic range of selected frequency bands. The reader is referred to the discussion of dynamics processing on a micro time scale in chapter 5.

Cross-Synthesis: Vocoding, Spectral Mutation, and Analysis-Based Formant Filtering

Cross-synthesis extracts characteristics from the spectrum of one signal and uses them to modify the spectrum of another signal. This can take a variety of forms, including, among others, vocoding, spectral mutation, and formant filtering.

Vocoding (as opposed to phase vocoding) analyzes one signal to adjust a bank of filters (or subbands) that are applied to a second signal. Its operation is relatively simple, meaning that it can be computed in real time. First, it extracts the amplitude envelope of the input signal coming through each filter. This is typically accomplished by rectification and lowpass filtering. The signal from which the amplitude is extracted is called the modulator; the signal that is affected is called the carrier. It then applies the amplitude envelopes from the modulator to a second filter bank through which the carrier is passing. In a typical vocoder, there are a relatively small number of filters (<50) and the center frequency of each of the filters is fixed. These constraints give the effect its characteristic sound quality.

An example of a vocoder is found in Arboretum's Hyperprism program (MacOS). This 26-band real-time vocoder performs an FFT on successive windows of the modulator signal. It adjusts the amplitude of its filter bands to match the spectrum of the incoming signal. That is, the modulating signal sets the gain for each of the filters. It then applies the 26-band filter bank to the carrier signal.

Another type of cross-synthesis involves mutations from one set of spectral data to another. The spectral mutation operations in the SoundHack program (MacOS) interpolate the sign or magnitude of a source set of spectral frames into the sign or magnitude of a target set (Polansky and Erbe 1996). The mutation functions operate on the phase and amplitude data of each analyzed frequency band. A single frame spans a microtemporal duration that is usually between 10 and 90 milliseconds. The degree of mutation is called Ω , which takes a value between 0 (source) and 1 (target). The software offers five mutation functions, some of which move linearly from one the source to the target, while others scramble the two.

A third possibility is creating a new spectrum by taking the intersection of two source spectra. This can be achieved in a variety of ways. One of the spectra, for example, might be derived from an analysis based on linear predictive coding or LPC (Roads 1996). LPC analysis calculates an overall spectrum envelope. One could use the LPC spectrum of one sound to shape the detailed spectrum of another sound, where the detailed analysis was derived by a phase vocoder (Serra 1997). U&I Software's MetaSynth program (MacOS) takes another approach to this cross-synthesis (Wenger and Spiegel 1999). Its Formants Filter effect uses a 128-band filter bank to find the formant peaks in the spectrum of one signal. It then applies this bank to boost the same formant regions in another signal. The effect thus emphasizes the frequencies that the two sounds have in common.

Other Operations on the Time-Frequency Plane

Over the past decade, an extensive catalog of transformations based on windowed spectrum analysis has been developed. Many of these were originally implemented in the Composer's Desktop Project (CDP) software package for Windows computers (Endrich 2000). Trevor Wishart's book *Audible Design* (1994) describes many of these effects using evocative drawings. Table 6.1 is a list of spectrum-based operations available in the CDP package, grouped into fourteen categories. Some of the operations are utilitarian, and do not transform the sound.

For those working on MacOS computers, at least a dozen software packages provide spectrum analysis and resynthesis capabilities. Many of the CDP spectrum operations, for example, were made available as plugins by Michael Norris (1997) and Alex Yermakov (1999) within the SoundMaker program (Ricci 1997).