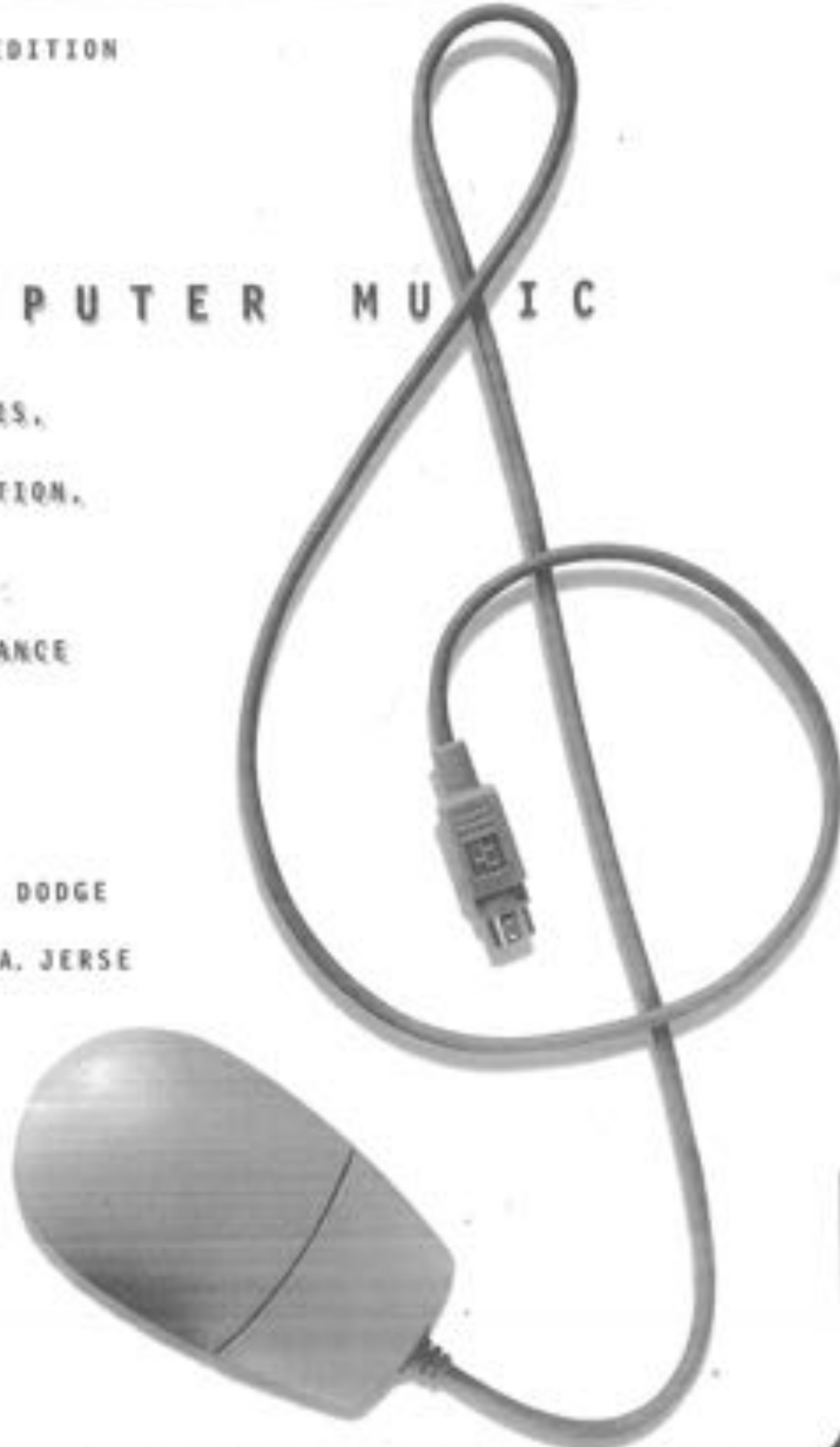
SECOND EDITION

COMPUTER MUSIC

SYNTHESIS, COMPOSITION, AND PERFORMANCE

CHARLES DODGE

THOMAS A. JERSE



$42.00

First published in 1985, ***Computer Music*** has been embraced worldwide as the best available introduction to the use of computer synthesis in musical composition. This new edition of ***Computer Music*** builds on the foundation of the original book to address the revolution in computing technology that has put computer music within the reach of all, including the availability of powerful personal computers at low cost, the development of user-friendly software, and the establishment of the MIDI interface for digital control of music hardware. The text is informally divided into five parts. In the first, the fundamentals of computers, psychoacoustics, and digital audio are clearly set forth. The second part describes a wide range of techniques for sound synthesis, including recent developments in synthesis-from-analysis, granular synthesis, and synthesis using physical models. In the third part, approaches to sound modification are discussed, including recent research on sound source localization and new techniques for ***musique concrète***. The fourth section is devoted to methods of computer-aided composition, with programming examples now presented in the widely used language C++. The closing chapter presents modes and techniques of computer usage in live performance.

Many new compositional examples are included, and current trends in performance practice are described. In short, this book serves as a thorough introduction to the world of computer music—synthesis, composition, and performance—written in such a way that music students can understand it.

CHARLES DODGE first gained recognition as a composer of orchestral and chamber music, then became one of the earliest composers to realize the vast musical potential of the computer. His ***Speech Songs*** (1972) is recognized as a classic of computer music. He has also composed works combining acoustic instruments with computer sound, including the widely performed ***Any Resemblance Is Purely Coincidental***, which features the voice of Enrico Caruso as reproduced and altered by computer synthesis. He is visiting professor of music at Dartmouth College.

THOMAS A. JERSE has worked extensively in the development of hardware and software for musical applications. He is a former assistant professor of music at Brooklyn College of the City of New York and was the first technical director of the Center for Computer Music there. He is currently a principal engineer with the Boeing Defense and Space Group.

Cover concept/design: reitdesign
Digital photograph: Andy Darlow-Unidigital/Cardinal Corp.

SCHIRMER BOOKS
An Imprint of Simon & Schuster Macmillan
1633 Broadway
New York, NY 10019

http://www.mlr.com/schirmer

ISBN 0-02-864682-7

9 780028 646824 90000

trolled by the output of linen. The frequency of the oscil is taken from p4 of the note statement. The third argument, the number of the wave table used for the waveform of the oscil, indicates that the stored wave table number 2 will be used. The out statement's audio rate argument, a2, causes the result of the oscil operation to be sent to the output of the orchestra.

```
instr     1
k1  linen             p5,p6,p3,p8
a2  oscil             k1,p4,2
out a2
endin
```

EXAMPLE 4.1 Csound code for the instrument shown in figure 4.11a

F. Richard Moore worked on the creation of Music 5 at Bell Laboratories with Max Mathews in the late 1960s. He created Cmusic around 1980 in the C programming language in order to make computer music in a UNIX operating system environment. Cmusic resembles Music 5 in the way the score and orchestra are encoded. In Cmusic, as in Music 5, the structure of the execution stage calls for each unit generator to contribute a number of successive outputs to a block of output samples (b1 and b2 in our example). The output block provides the means for interconnecting the unit generators.

```
ins 0 SIMPLE;
osc b2 p5 p10 f3 d;
osc b1 b2 p6 f1 d;
out b1;
end;
```

EXAMPLE 4.2 Cmusic code for the instrument shown in figure 4.11b.

The ins and end statements serve essentially the same purpose in Cmusic as in Csound—to delimit the definition of the instrument. In our Cmusic example, we call for instrument SIMPLE to be defined at time 0. We use an osc unit generator to perform the function of an envelope control unit. Its arguments are b2, the number of the location for its output; p5, its amplitude; p10, the duration of the envelope; f3, the number of the table storing the envelope shape; and d, the phase of the oscillator. The second osc statement in the example refers to another oscillator, this one used to generate the signal. The first argument is b1, the number of the block for its output; b2, the input amplitude taken from the output of the previous unit generator; p6, the frequency; f1, the stored waveform for the oscillator; and again, d. The out statement calls for the output of the second oscillator, b1, to be sent to the output of the orchestra.

4.7 ADDITIVE SYNTHESIS

The simple instrument that was shown in figures 4.5 and 4.11 is the first configuration used to synthesize musical sound. It is based on a simplified Helmholtz model of musical sound, which consists of a waveform at a constant frequency enclosed in an envelope. The choice of waveform may be made in many ways. Early attempts at using this instrument to approximate natural sounds analyzed the steady-state spectrum of the tone to be matched.

The sound produced by this instrument differs from natural sound in two important respects. First, the amplitudes of all the spectral components are varied equally by the envelope, so that amplitudes of the components relative to each other do not change during the course of the tone. Thus, the sound lacks independent temporal evolution of the harmonics, an important characteristic of natural sound. Second, all the spectral components are exact-integer harmonics of the fundamental frequency, not the slightly mistuned partials that often occur in acoustically generated sounds.

As explained in chapter 2, each spectral component of a sound can be represented by its own independent amplitude and frequency functions. The synthesis of a tone based on this model (figure 4.12) requires a separate sinusoidal oscillator for each partial, with the appropriate amplitude and frequency functions applied to it. The output from each of the oscillators is added together to obtain the complete sound. Hence, the name additive synthesis is used to designate this technique.

Additive synthesis provides the musician with maximum flexibility in the types of sound that can be synthesized. Given enough oscillators, any set of independent spectral components can be synthesized, and so virtually any sound can be generated.

The amplitude and frequency functions can be obtained from the analysis of real sounds as described in chapter 2. The name *Fourier recomposition* is sometimes used to describe the synthesis from analysis, because it can be thought of as the reconstitution

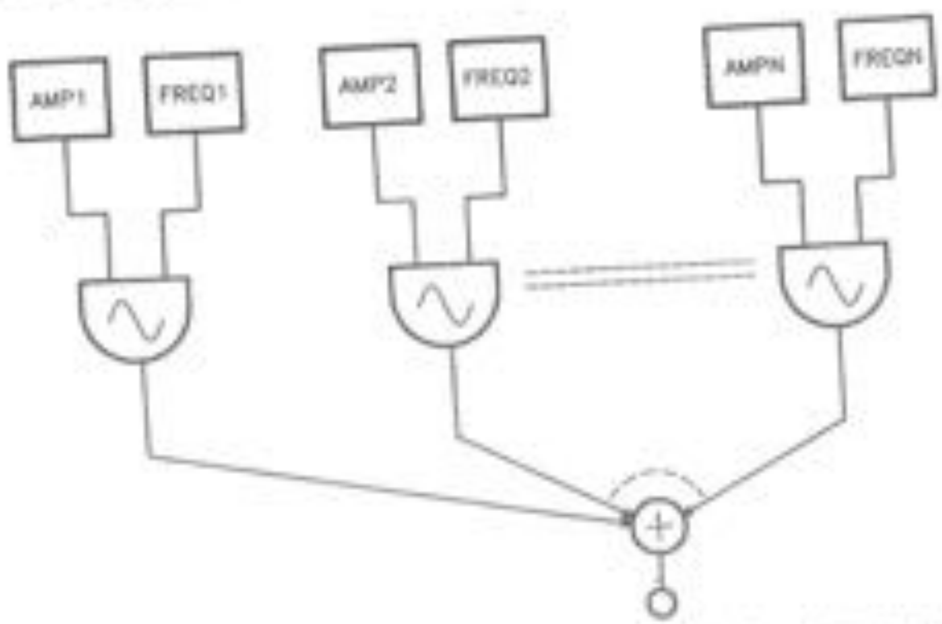


FIGURE 4.12 Basic configuration for additive synthesis. The amplitude and frequency inputs of each oscillator derive from independent function generators, which usually take the form of envelope generators or oscillators.

of the time-varying Fourier components of a sound. Additive synthesis has proven capable of realizing tones that are "indistinguishable from real tones by skilled musicians."[7]

Of course, the instrument designer is not restricted to using functions obtained from analyses. When choosing functions, however, it is helpful to have a knowledge of the behavior of functions that describe natural sounds. For example, in many acoustic instruments, the higher harmonics attack last and decay first. Knowing this, a musician might choose to synthesize an unusual sound using a set of functions with the opposite characteristic.

New, natural-sounding functions can be generated by interpolating between sets of functions that have been derived by analysis.[8] Musicians can use this technique to transform gradually the sound of one instrument into that of another. The normal way to interpolate between two functions is to take a weighted average between comparable points on each function. For example, to generate a function of time that is 30% of the way between function 1 (F1) and function 2 (F2), the new function is formed as the sum of 70% of F1 and 30% of F2. However, applying this technique to two amplitude functions that peak at different times creates a new function, which lacks a single, sharp maximum. Instead, the peak of the function broadens, spread between times of the peaks of the original functions. Upon synthesis, this discrepancy introduces an additional timbral element that is unrelated to either of the original sounds. To preserve a clear, single maximum, and hence more timbral similarity with the original sounds, the time at which the maximum of the new function occurs is interpolated between the times of the original functions. For example, suppose F1 peaks at 0.1 second and F2 peaks at 0.15 second. The function that is 30% of the way between F1 and F2 would peak at $0.7 \times 0.1 + 0.3 \times 0.15 = 0.115$ second. Having established the time of the maximum, the attack portion of the new function is interpolated between the attack portions of the original functions. Similarly, the decay portion is interpolated using the decay portions of the originals.

Additive synthesis produces high-quality sound but requires a comparatively large amount of data to describe a sound because each of the many oscillators requires two functions. A further complication arises because a given set of functions is normally useful only for limited ranges of pitch and loudness. If a set is determined by analysis for a specific pitch, then it will produce the timbral quality of the original source in only a small pitch interval around that point. Any formants present in the spectrum will move directly with the fundamental frequency. Thus, much of the timbral similarity between the tones of different pitches will be lost. In addition, the functions are highly sensitive to dynamic level, so that a set determined for a *mezzo forte* will produce an unrealistic *pianissimo*. To fully realize the benefits of additive synthesis, it is necessary to have either a large library of function sets or a complex scheme for altering a function set on the basis of pitch and amplitude during performance. (The spectral interpolation technique described in section 4.10 reduces the data storage requirements without a perceptible loss in quality.)

An advantage of additive synthesis is that it provides complete, independent control over the behavior of each spectral component. However, such a large number of controls on the timbre can make it difficult for the musician to know how to achieve a particular sound. A practical disadvantage of additive synthesis is that it requires a large number of unit generators. When synthesizing complex sounds, it is not unusual to employ 10 or more oscillators with their associated function generators in the synthesis of a single voice. This characteristic differs from and motivates the synthesis techniques presented in subsequent chapters, which use fewer unit generators.

4.8 MODULATION

Modulation is the alteration of the amplitude, phase, or frequency of an oscillator in accordance with another signal. Modulation has been used for many years in radio communications to transmit information efficiently. Musicians have exploited various modulation techniques in electronic music to create distinctive sounds efficiently.

The oscillator that is being modulated is called the *carrier oscillator*. If it were run without modulation, it would generate a continuous waveform called the *carrier wave*. When modulation is applied, the carrier wave is changed in some way. The changes are in sympathy with the modulating signal, so that the output of the carrier oscillator may be thought of as a combination of the two signals. The nature of this combination depends on the modulation technique used and will be examined below.

The spectral components of a modulated signal are classified into two types: *carrier components* and *sidebands*. The frequency of a carrier component is determined only by the frequency of the carrier oscillator. The frequency of a sideband is determined by both the carrier frequency and the frequency of the modulation.

4.8A Amplitude Modulation

There are three main techniques of amplitude modulation: "classical" amplitude modulation, ring modulation, and single-sideband modulation. The letters AM are most often used to denote the first type. Ring modulation finds use in several techniques of computer music and will be presented in section 4.8B; the application of single-sideband (SSB) modulation to computer music is rare and will not be discussed here.

Figure 4.13 diagrams an instrument that implements classical amplitude modulation (AM). The carrier oscillator has a constant frequency of f_c and the modulating oscillator a frequency of f_m. For this example, the waveform of each oscillator is a sinusoid. The output from the modulating oscillator is added to a value that expresses the amplitude the carrier oscillator would have if there were no modulation. The amplitude of the modulating oscillator is expressed as a proportion of the unmodulated amplitude of the carrier oscillator. This proportion is denoted by the variable m, which is called the *modulation index*. When $m = 0$, there is no modulation and the carrier oscillator generates a sinusoid with a constant amplitude of AMP. When m is larger than 0, the carrier wave will take an envelope with a sinusoidal variation (figure 4.13b). When $m = 1$, the amplitude of the modulating oscillator equals the unmodulated amplitude of the carrier oscillator and 100% modulation is said to take place.

When both the carrier and the modulating waveforms are sinusoids, the spectrum of an AM signal (figure 4.14) contains energy at three frequencies: the carrier frequency (f_c) and two sidebands ($f_c + f_m$ and $f_c - f_m$). The amplitude of the component at the carrier frequency does not vary with the modulation index. The amplitude of each sideband is a factor of $m/2$ less than the amplitude of the carrier, showing that this modulation process splits the energy between equally upper and lower sidebands. For example,

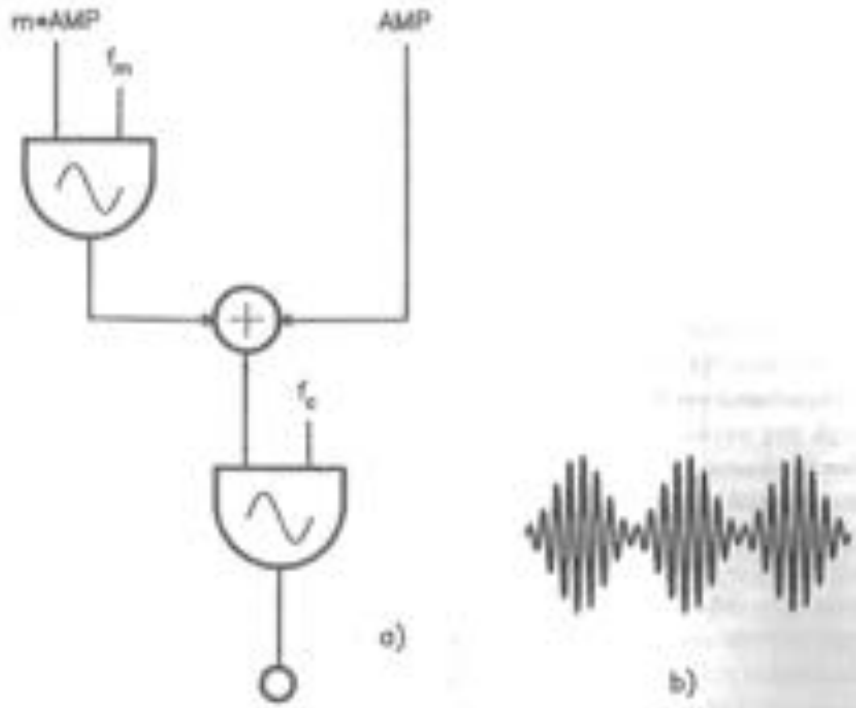


FIGURE 4.13 Simple instrument (a) that implements amplitude modulation and its output waveform (b).

when $m = 1$, the sidebands will have one-half the amplitude of the carrier, and therefore will be 6 dB below the level of the carrier.

The frequency of the modulation determines how a listener perceives the AM sound. If f_m is less than about 10 Hz, the ear will track the individual amplitude variations. When f_m is greater than 10 Hz, but small enough that the carrier and both sidebands fall within the same critical band, the tone will sound with a loudness proportional to the average amplitude of the modulating waveform. A value of f_m that exceeds one-half the critical band causes the sidebands to be perceived individually, creating the sensation of additional loudness. Musicians have used amplitude modulation to create electronic "tremolo"

FIGURE 4.14 Spectrum of the AM signal produced by the instrument of figure 4.13.

by using a small modulation index and subaudio modulating frequency. When the modulation index is close to unity and f_m is small, a markedly pulsating sound will be produced.

4.5B Ring Modulation

When modulation is applied directly to the amplitude input of a carrier oscillator, the process created is known as *ring modulation*. Other names for it are *balanced modulation* and *double-sideband (DSB) modulation*. Figure 4.15 illustrates the signal flowchart for an instrument where one oscillator ring modulates another. The amplitude of the carrier oscillator is determined only by the modulating signal, so that an absence of modulation ($A = 0$) causes the output from the instrument to go to 0. This characteristic is a noticeable departure from the configuration used in the AM technique described above.

Although ring modulation operates on the amplitude of the carrier, it is most often used to alter the frequency of a sound. When both the carrier (f_c) and modulating (f_m) signals are sinusoidal, the spectrum of the modulated signal contains only two frequencies: $f_c + f_m$ and $f_c - f_m$. In other words, ring modulation produces sidebands but no carrier. Because neither f_c nor f_m appears directly in the spectrum, the frequency of the sound can be quite different. For example, if $f_m = 440$ Hz and $f_c = 261$ Hz, the resulting spectrum contains energy only at 179 Hz and 701 Hz, frequencies that are not harmonically related to the originals or to each other. Given the amplitude of the modulating signal in figure 4.15 as A, both sidebands have amplitudes of A/2.

Ring modulation is often used for sound modification. All frequencies in a sound that is applied directly to the amplitude input of an oscillator (figure 4.16) are changed by ring modulation. Suppose a speech sound with a fundamental frequency of 100 Hz ring-modulates a sinusoidal oscillator with a frequency of 1123 Hz. The sound that emerges contains the sum

FIGURE 4.15 Simple instrument that implements ring modulation of one oscillator by another

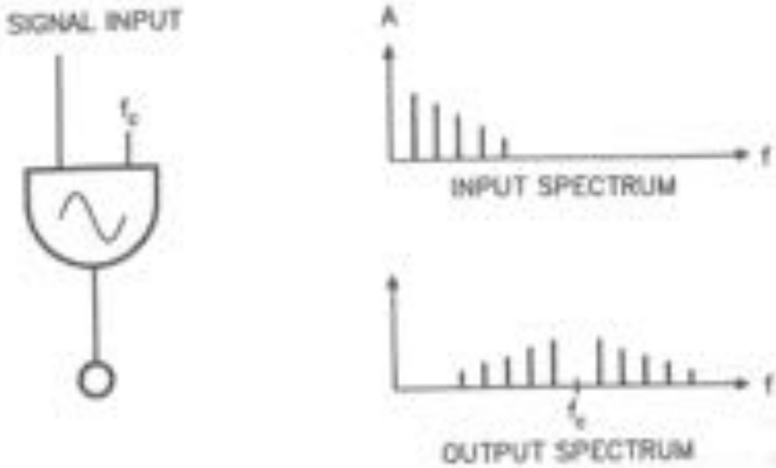


FIGURE 4.16 Alteration of the spectrum of a signal by ring modulation.

and difference between each harmonic of the speech and 1123 Hz. Thus, the spectral component that was the fundamental of the speech sound is output at both 1023 Hz and 1223 Hz, the former second harmonic (originally at 200 Hz) appears at 923 Hz and 1323 Hz, and so on. The formerly harmonic speech now sounds inharmonic and may not be intelligible.

Ring modulation may be realized without oscillators just by multiplying two signals together. Thus, the multiplier shown in figure 4.17 is a general-purpose ring modulator. Two signals are often combined in this way for the purpose of frequency alteration. Suppose that two sine waves, with amplitudes A_1 and A_2 and frequencies f_1 and f_2, respectively, are multiplied together. The resulting spectrum will contain frequencies of $f_1 - f_2$ and $f_1 + f_2$, and the amplitude of each component will be $A_1A_2/2$. Observe that if either signal has an amplitude of 0, there will be no output from the modulator. Composers such as Jean-Claude Risset (see section 4.12) and James Dashow[3] have used this form of ring modulation for the creation of chordal structures.

The multiplication of two complex sounds produces a spectrum containing frequencies that are the sum and difference between the frequencies of each component in the

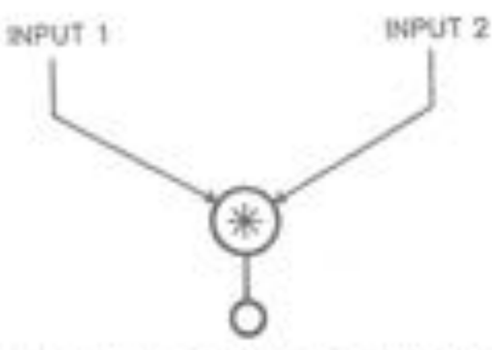


FIGURE 4.17 A multiplier is a general-purpose ring modulator.

first sound and those of each component in the second. If there are p components in the first sound and q components in the second, as many as $2pq$ components can appear in the output. Thus, multiplication can be used to create dense spectra. For example, if two signals, each with four components, are multiplied together (figure 4.18), the resulting sound will have as many as 32 components. There would be fewer components if the two signals were harmonically related because some of the sidebands would have the same frequencies, reducing the overall number of observed spectral components. To avoid aliasing, it should be noted that the highest frequency produced by this process is the sum of the highest frequency contained in the first sound and the highest in the second.

4.8C Vibrato Simulation by Frequency Modulation

When a modulating signal is applied to the frequency input of a carrier oscillator, frequency modulation occurs. Vibrato, a slight wavering of pitch, can be simulated using the instrument in figure 4.19. The carrier oscillator generates a tone at the specified amplitude and frequency (f_c), and the vibrato oscillator varies that frequency, at the vibrato rate, by a maximum amount equal to the vibrato width. Thus, the instantaneous frequency of the carrier oscillator changes on every sample, varying between f_c plus the vibrato width and f_c minus the vibrato width. Its average frequency is f_c.

The vibrato width is usually specified as a proportion of the fundamental frequency

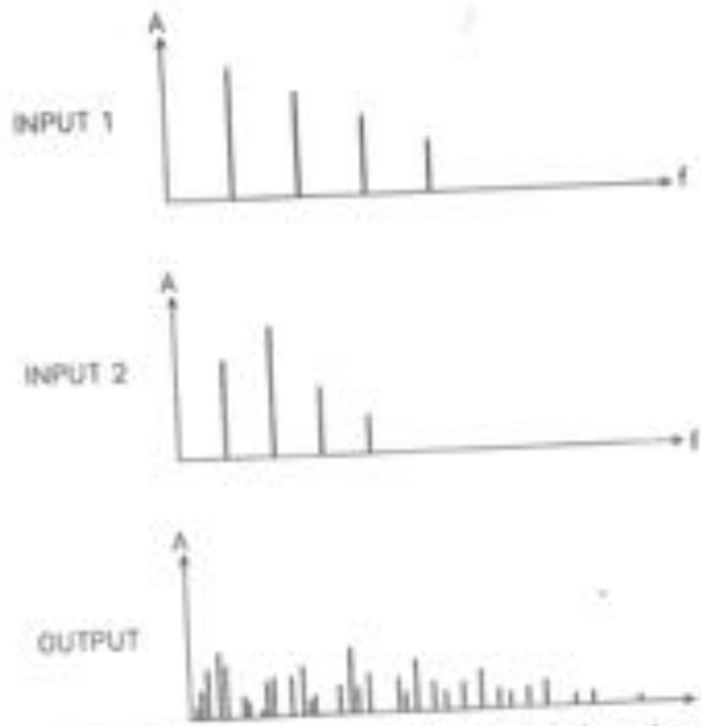


FIGURE 4.18 Ring modulation of two signals to produce a dense spectrum.

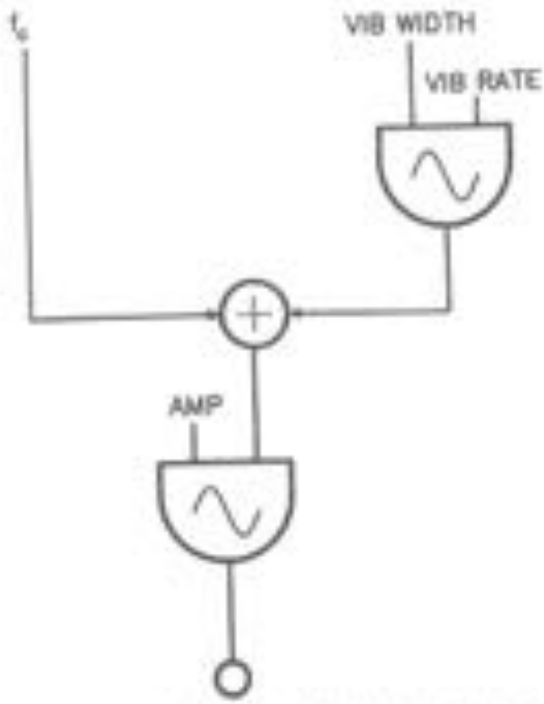


FIGURE 4.19 Simulation of a simple, periodic vibrato.

of the tone and is ordinarily no more than a few percent of f_c. In order for the frequency modulation to be perceived as vibrato, the vibrato rate must be restricted to frequencies below the audio range. The vibrato found in natural sounds can be quite complex. Its rate and width often changes during the course of a tone and frequently contains a certain amount of randomness. Chapter 7 will discuss one of the more complicated forms of natural vibrato, that of the singing voice.

When a large vibrato width at an audio rate is used, the aural effect is no longer that of a simple vibrato. Under these conditions, frequency modulation becomes a powerful synthesis technique capable of producing a wide variety of timbres. Chapter 5 will cover the theory and applications of frequency modulation synthesis in detail.

4.9 NOISE GENERATORS

An oscillator is designed to produce a periodic waveform with well-defined spectral components. The spectrum is a *discrete spectrum*; that is, the energy is found at specific, harmonically related frequencies. The opposite of a discrete spectrum is a *distributed spectrum*, in which energy exists everywhere within a range of frequencies. Most of the noise sounds found in nature have distributed spectra, and thus algorithms designed to generate distributed spectra are called *noise generators*.

Certain phenomena have the characteristic that their repeated occurrence, even under the same set of conditions, will not always lead to the same result. Members of

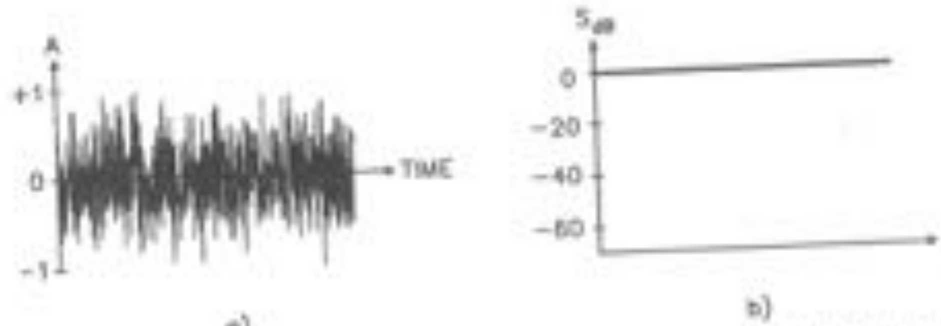


FIGURE 4.20 (a) Segment of the waveform of white noise and (b) the spectrum of ideal white noise.

this class are called *random phenomena*. Even though the exact outcome cannot be predicted, they exhibit a certain amount of statistical regularity that can be used to describe them and to predict the probability of any given occurrence. The statistical characterization of a random signal is used to determine its frequency. (Additional explanation of random processes can be found in chapter 11 as they are applied to composition.)

In sound synthesis, randomness is used to generate distributed spectra. The waveform pictured in figure 4.20a is a segment of the waveform of white noise. If it were digitized, there would be no recognizable pattern of sample values; in fact they would appear to be randomly distributed. The amplitude of the digitized white noise is characterized by a range—the interval within which the maximum and minimum sample values occur. In the figure, the range is −1 to +1. Because, unlike a periodic waveform, a repeating pattern of samples cannot be identified, signals of this type are referred to as aperiodic. White noise has a uniformly distributed spectrum as shown in figure 4.20b. Between any two frequencies a fixed distance apart, there is a constant amount of noise power. For instance, there is the same amount of noise power in the band between 100 and 200 Hz, as there is between 7900 and 8000 Hz. White noise makes the "hissing" sound often associated with white noise generated by electronic means.

The unit generator that produces nearly white noise is often called RAND and has an amplitude input. Its symbol is shown in figure 4.21a. The amplitude input sets the range of the permissible output sample values, and hence the amplitude of the noise. If a value AMP is applied to the input, the sampled noise will range between −AMP and +AMP.

The basic algorithm used to generate white noise simply draws a random number on each sample. This makes a good, but not perfect, white-noise source. The spectral distribution of such a generator is shown in figure 4.21b. It deviates slightly from a uniform distribution because of a frequency bias inherent in the process of sample generation. The actual spectral distribution $S(f)$ at frequency f is given by

$$S(f) = \frac{\sin\left(\pi \frac{f}{f_s}\right)}{\pi \frac{f}{f_s}}$$

FIGURE 4.21 Digital white-noise generator and its spectrum.

True white noise would have $S(f) = 1$. Over most of the spectrum, the generated noise power is very close to this ideal. At the Nyquist frequency, $S(f_s / 2) = 0.6366$, showing that the power is down less than 4 dB from a true uniform distribution at the top of the frequency range.

White noise has a very large bandwidth. Sometimes it is desirable to narrow the bandwidth by reducing the amount of high-frequency energy in the noise. A noise source with most of its power at low frequencies will make a rumbling sound. An algorithm that synthesizes this kind of spectrum draws random numbers at a rate less than the sampling rate. The unit generator that accomplishes this is often called RANDH (figure 4.22a) and has two inputs: amplitude and the frequency (f_R) at which random numbers are drawn. (On some systems, the frequency is not specified directly, but by a number proportional to f_R, in the same way that the sampling increment is proportional to the frequency of an oscillator.) Choosing random numbers at a rate lower than f_s implies that a random number is held for a few samples, until the next one is to be drawn. For example, if $f_s = 40$ kHz and $f_R = 4$ kHz, the algorithm chooses a random value, outputs it for the next 10 samples, and then chooses another.

When noise is generated by this process, many of the samples are related to each other because their value is the same as the previous sample. This relatedness reduces the noise power in the higher frequencies. The lower the frequency f_R, the smaller the amount of high-frequency energy that will be contained in the noise. Thus, f_R can be thought of as a control on the "bandwidth" of the noise. Figure 4.22b illustrates the spectrum when $f_R = f_s / 6$. The shape of the spectrum in the general case is given by the product of two functions as

$$S(f) = \frac{\sin\left(\pi \frac{f}{f_R}\right)}{\pi \frac{f}{f_R}} \frac{\sin\left(\pi \frac{f}{f_s}\right)}{\pi \frac{f}{f_s}}$$

A variation on this technique, one that provides noise spectra with even greater attenuation of the high frequencies, involves interpolation. As before, random numbers are drawn at a rate (f_R) that is lower than the sampling rate. Instead of holding the value

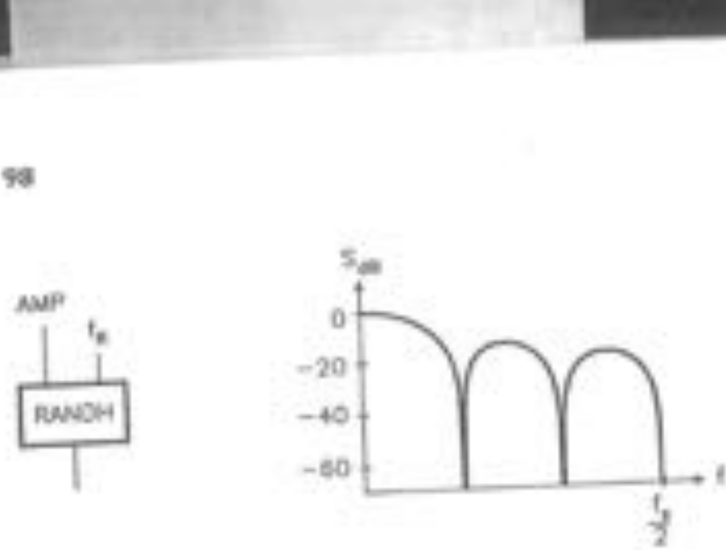


FIGURE 4.22 Digital noise generator (a) in which noise samples are generated at a rate lower than the sampling rate and its spectrum (b).

of the last random number until the next one is drawn, the samples in between draws are interpolated linearly between successive random numbers. This can be visualized as connecting successive random numbers with straight lines. The unit generator that performs this algorithm is often called RANDI (figure 4.23a).

Figure 4.23b illustrates the spectrum of such a noise generator when $f_R = f_s / 6$. Observe the diminished amount of high-frequency energy. The general shape of the spectrum is given by

$$S(f) = \frac{f_R^2\left[1 - \cos\left(2\pi\frac{f}{f_R}\right)\right]}{2\pi f f_s \sin\left(\pi\frac{f}{f_s}\right)}$$

It is possible to realize noises with other types of spectral distributions such as 1/f (see section 11.1G), in which the spectrum is distributed in inverse proportion to the frequency. Other techniques, such as the one proposed by Siegel, Steiglitz, and Zuckerman,[a] are available for generating random signals with specifiable spectral densities.

How does the computer, which is designed to store and process numbers accurately and reproducibly, obtain seemingly unpredictable random numbers? One way is to sample

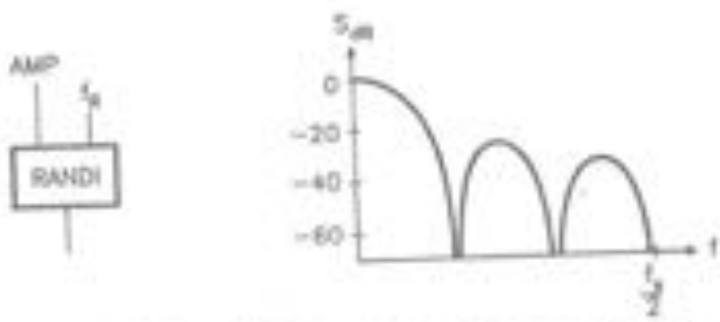


FIGURE 4.23 Interpolating noise generator (a) and its spectrum (b).

an external random physical process such as thermal noise, but this requires additional hardware. A less expensive and more commonly used approach is to employ an algorithm called a pseudo-random number *generator*,[7] which produces a sequence of numbers that satisfy most of the criteria for randomness, with the notable exception that the sequence repeats itself. Fortunately, it is possible to make the period of the sequence so long that for most purposes it can be considered random. A pseudo-random number generator actually creates a discrete harmonic spectrum, but the spectrum is so extraordinarily dense that, for most musical applications, it is indistinguishable from a truly distributed one.

Pseudo-random number generators generally use the most recently generated random number as a basis for calculating the next. As a result, most algorithms give the user the option of specifying a "seed" value. When a seed is specified, the first random number will be calculated on the basis of the seed value. Therefore, starting from the same seed will always yield the same sequence of random numbers. This facility allows two different computer runs involving the generation of pseudo-random numbers to have exactly the same results.

4.10 SYNTHESIS BY MEANS OF SPECTRAL INTERPOLATION

The additive synthesis of tones using the data from prior spectral analyses can re-create tones that are very close to the original. As stated above, the principal drawback to this technique is the large data sets required to represent each timbre. When given the need for a single instrument to have separate data sets for tones in different pitch regions and at different dynamic levels, the impetus for data reduction is strong indeed. One approach, previously described in section 2.7, is to approximate the records of the dynamic spectra with a small number of straight-line segments. In this way, each partial of the tone would be described by simple linear amplitude and frequency envelopes. Beginning with the work of Risset, this method has proven effective in synthesizing realistic tones, but further data reduction can be achieved by the method of spectral interpolation.[8]

As diagrammed in figure 4.24, spectral-interpolation synthesis requires two oscillators, each with a different waveform stored in its wave table. The mix between the two oscillators is controlled by the interpolation parameter x. When x is set to 0, the output consists of only the waveform (WF0) produced by the left oscillator. By contrast, setting x to unity results in the output waveform (WF1) from the right oscillator because the signal contributed by the left oscillator is both added to and subtracted from the output resulting cancellation. The intermediate value of $x = 0.4$, for example, would produce an output that combined 60% of WF0 with 40% of WF1. Figure 4.25 shows an example of the constantly changing waveform produced when x is linearly advanced between 0 and 1.

The method takes advantage of the relatively slow changes exhibited in the waveform during most of the course of a tone. A succession of short-term spectra of the tone to be synthesized is derived. From this analysis, a few representative breakpoints are designated between which the spectra will be interpolated. The density of these breakpoints depends on how quickly the tone is changing. During the attack portion, many individual spectra are required, but during the sustain portion, the harmonic content is changing so slowly that only a few breakpoints can be used.

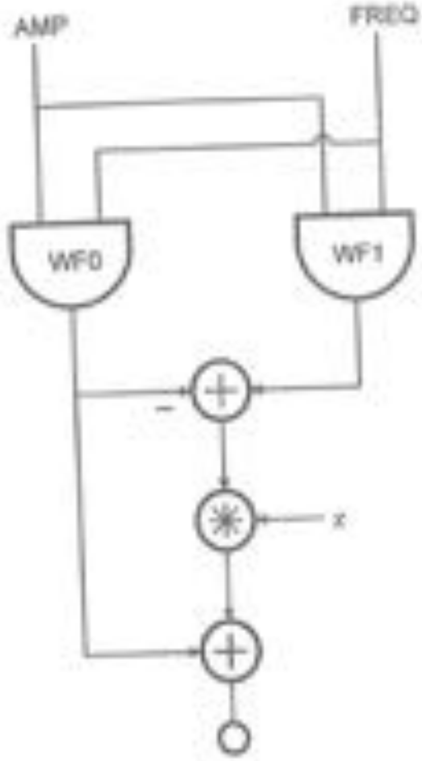


FIGURE 4.24 An instrument that interpolates between the outputs of two oscillators with parameter x controlling the mix.

The data used for spectral-interpolation synthesis are most frequently obtained by the analysis of acoustic sounds. The input is broken down into a series of short-term spectra (see section 7.2), and an analysis algorithm is applied to determine the number and temporal position of the breakpoints required. The algorithm positions the breakpoints by applying a routine that minimizes the spectral error over the course of the tone between the original signal and the one produced by spectral interpolation. To simplify the analysis and storage requirements, it is assumed that the phase of the partials relative to each other has a negligible aural effect. When the spectral-interpolation technique is used to create instruments that simulate natural ones, data must be taken at many different pitch and dynamic levels; but the considerable data compression inherent in this technique at each level makes such an approach practical.

4.11 INSTRUMENT DESIGNS

Each of the chapters of this book in which sound synthesis techniques are discussed includes a number of instrument designs. Because it is anticipated that the readers will be using a variety of musical programming languages, we have used flowcharts to express the instrument designs. The instrument designs are offered as a guide to what ts of two oscillators with

frequently obtained by
a series of short-term
determine the number
ms positions the break-
r the course of the tone
terpolation. To simplify
use of the partials rela-
tral-interpolation tech-
data must be taken at
lata compression inher-
tical.

chniques are discussed
ed that the readers will
ve used flowcharts to
rred as a guide to what

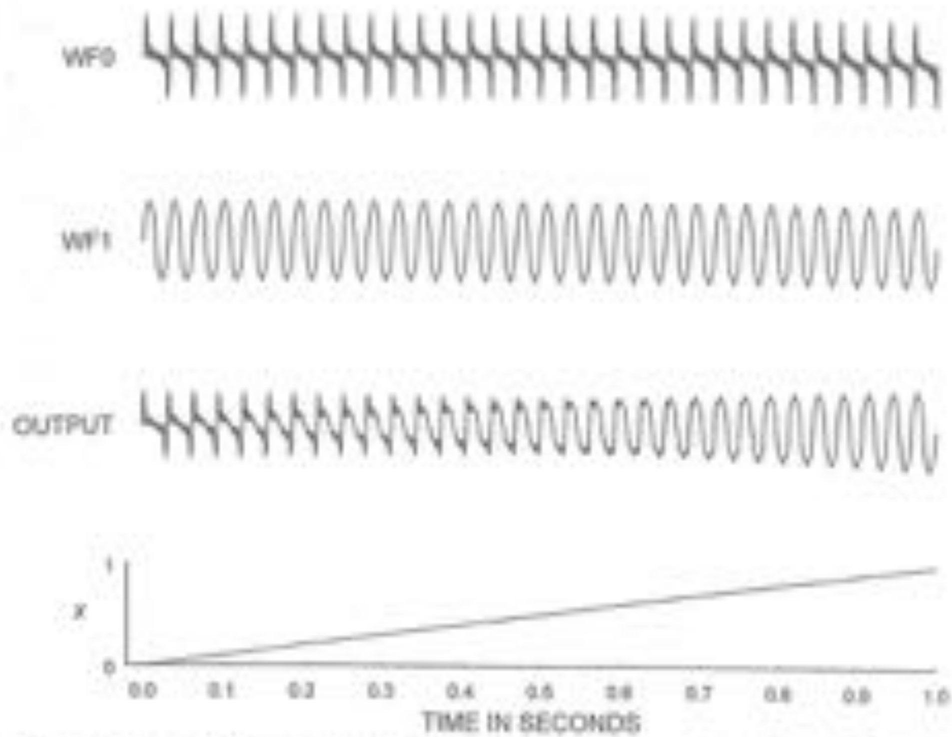


FIGURE 4.25 Waveform produced by interpolating between waveforms WF0 and WF1 as parameter x linearly increases as shown.

has been done with a particular sound synthesis technique. The instrument designs are neither exhaustive nor definitive; they are simply offered here as a starting point for the reader to develop a personal vocabulary of designs for computer synthesis.

Some instrument designs are likened to traditional instruments to give the reader a sense of their sound. The timbral impression evoked by these instruments on isolated notes is similar to the stated instrument. In a composition, their identification and their ability to hold the listeners' interest may also depend on the phrasing used and whether or not their musical part is idiomatic to that instrument.

4.11A Translating Flowchart Diagrams into Music Synthesis Code

Translating flowchart diagrams into written instrument definitions is a task that many musicians find initially difficult. Following is a general guide to the process. It is divided into two stages: analysis of the flowchart and coding of the instrument.

There are three steps involved in analyzing a flowchart. The first step is to find the output or outputs from the instrument. This helps show the basic structure of the instrument. Step 2 consists of designating the separate branches and subbranches of the instrument. By doing this, the musician divides the instrument into its component parts in order to know how to direct the flow of the signal from one unit generator to the next.

In step 3, the musician finds the sources for all the inputs and the destinations of the outputs of all the branches of the instrument.

When starting to encode the instrument design into the sound synthesis language, it is essential to make certain that the use of all the unit generators in the design is understood as well as the meaning of their inputs and outputs. Consult the manual for the particular sound synthesis language to be sure. Start the encoding with the uppermost unit generator either in the flowchart or in a branch of the flowchart. Write out the unit generator name, label its output (if appropriate in the language used), and fill its inputs from the initialization values. It is good practice for most sound synthesis languages to list the initialization values in a separate section at the head of the instrument.

Next, follow the same procedure for the subsequent unit generators of the branch or subbranch until the instrument is coded completely. Keep in mind that the inputs of unit generators to which the outputs of other unit generators are connected ordinarily get their values at performance time. Inputs not fed from other unit generators obtain their values at initialization time.

After all the branches of the instrument are coded, interconnect them by means appropriate to the language used. Finally, direct the results of the instrument into the output(s) by means of an output statement.

Following are some hints for proofreading the code that describes an instrument: (1) check that no unit generators have been omitted; (2) make certain that all unit generators input are of the correct form (e.g., that an input expecting a frequency in hertz is not given a frequency in some other notation); (3) make sure that all unit generators are given the required number of inputs; and (4) check to be certain that all stored functions referred to by the unit generator have the right contents.

Common mistakes in instrument coding include sending the output of one unit generator to the wrong input of the next, or sending it to the wrong unit generator entirely. Be meticulous in checking every input of every unit generator and in carefully labeling the branches of the instrument. Ample comments should appear at the head of the instrument to identify its function and characteristics.

After encoding the instrument design, check the code for correct syntax by invoking the orchestra translation program. The translator will make a trial translation of the code into machine language and give error messages if the syntax is faulty. Next, the musician should try out the instrument on a few typical notes in order to hear whether the instrument does what is wanted. It is possible, and indeed common, for a design to be syntactically correct but not to give the desired results. The trial tones will also be helpful in establishing the limits of the instrument's usefulness. Most instruments show great differences in sound, depending on such factors as note length, register, and amplitude.

Finally, the instrument must be tested in a musical context to find out whether it is appropriate for the musical articulation desired. At this point, such issues as the balance of the instrument with copies of itself in different registers, the balance of the instrument with other instruments in the same and other registers, and masking become important. It is often necessary at this point to recast parts of the instrument to fit the demands of the context in which it will be used.

4.11B Instrument Design Examples

Our first instrument design uses ring modulation to produce a band of noise. Controls on both the center frequency and the width of the band are provided. While more focused noise spectra can be synthesized by filtering white noise (see section 6.8), this method is both efficient and useful for many musical purposes.

As shown in figure 4.26, a noise generator ring-modulates a sinusoidal oscillator. This process translates the noise generator's low-frequency noise to a higher-frequency region, centering the noise band at the frequency (FREQ) of the oscillator. The amplitude input to the noise generator (AMP) directly controls the amplitude of the noise band. The frequency at which the random noise is generated, f_R, determines the bandwidth of the noise (see section 4.9). If the bandwidth is sufficiently small, the noise will be perceived as pitched. A noise band with a width of 20% of the center frequency will produce a good sensation of pitch. A noise band with a bandwidth of 5% of the center frequency will sound less "noisy" and have a highly focused pitch. A glissando of a noise band can be synthesized by programming the oscillator to a glissando. James Tenney realized the glissandoing noise bands of his *Noise Study* in this way.[9]

Figure 4.27 shows another use of a noise band created with random ring modulation. Jean-Claude Risset used this technique to simulate the sound of the snares in a drum instrument.[10] The three oscillators each contribute different components at different amplitudes. The decay of F2 is steeper than that of F1, so that the two oscillators on the left side (labeled NOISE and INHARM), which contain the higher-frequency components of the sound, die away sooner than the oscillator on the right (FUND). The latter oscillator samples a stored sine tone producing a tone at the fundamental frequency. The INHARM oscillator samples a stored waveform (F3) consisting of partials 10, 16, 22, and 23 with relative amplitudes 1, 1.5, 2, and 1.5, respectively. When the frequency of the INHARM oscillator is set to 1/10 that of

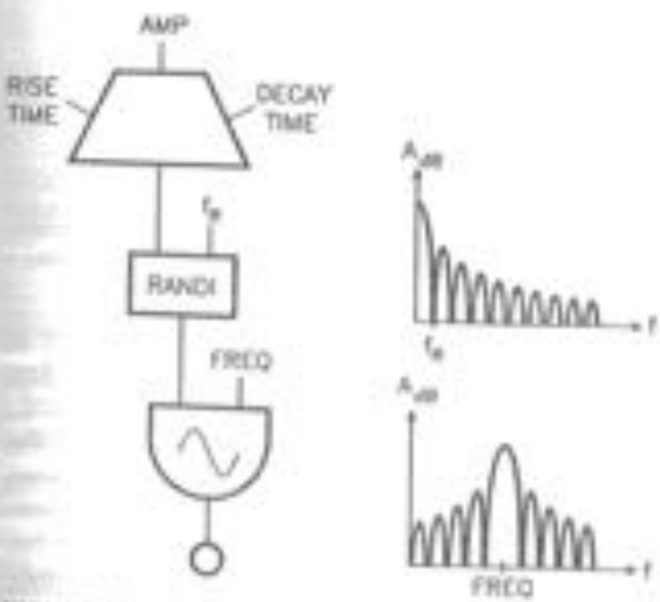


FIGURE 4.26 Generation of bands of noise by means of ring modulation.

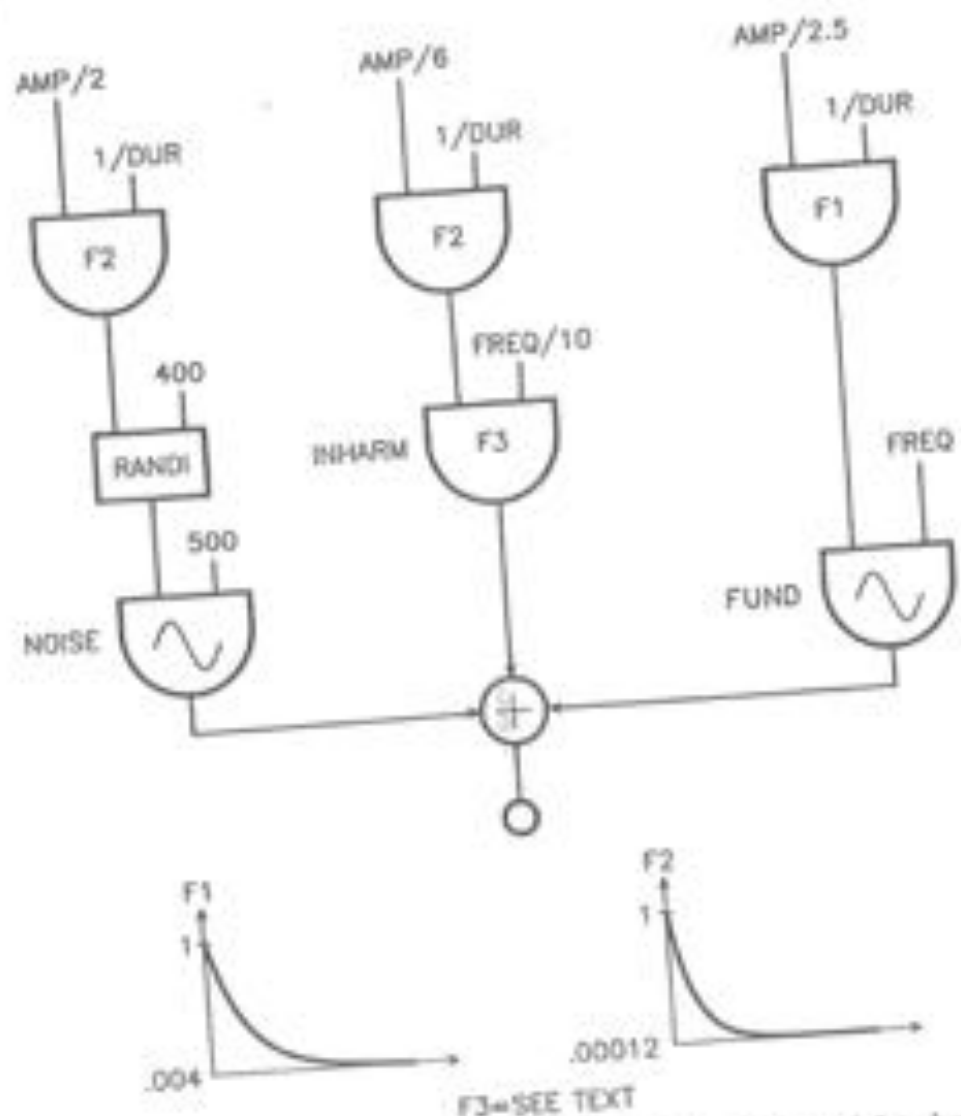
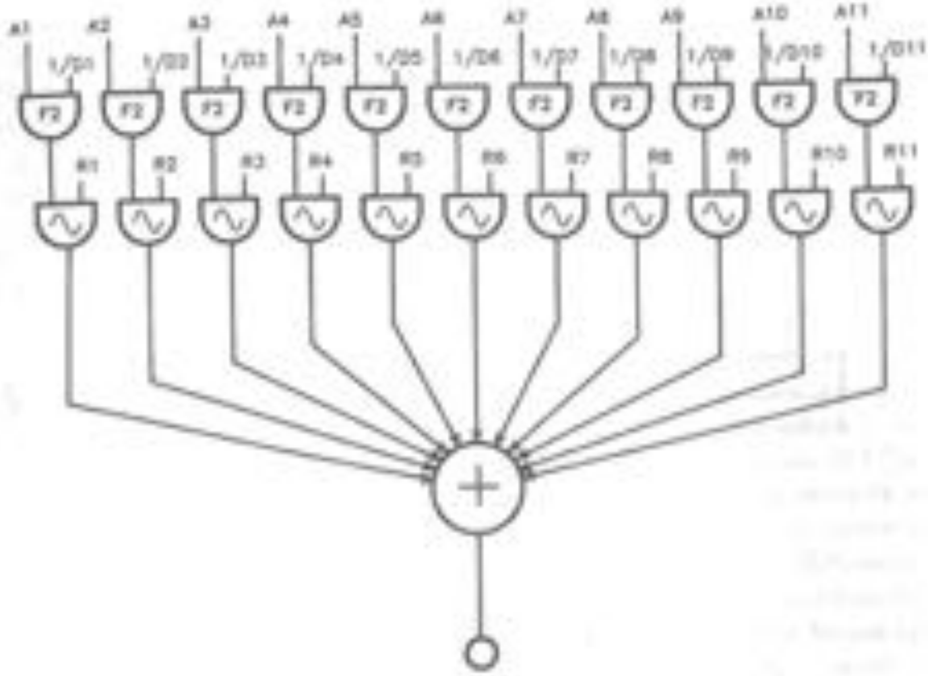


FIGURE 4.27 Drum instrument based on Risset. (Based on example in Risset's *Introductory Catalogue of Computer-Synthesized Sounds*. Reprinted with permission of Jean-Claude Risset.)

the FUND oscillator, its partials sound at 1, 1.6, 2.2, and 2.3 times the frequency of the fundamental, producing in this way a group of partials that is nonharmonic to the fundamental.

Risset has employed additive synthesis in a number of his works to produce bell-like sounds. A design based on one of the bell sounds of the Computer Sound Catalog is shown in figure 4.28. The three principal features that contribute to the bell-like sound are: (1) nonharmonic partials; (2) decay times of the partials roughly inversely proportional to their frequency; and (3) beating of pairs of components, slightly mistuned on the lowest two partials.

Risset points out that while the partials are inharmonic, they are not tuned arbitrar-



Amplitudes		Durations		Frequencies	
A1	AMP	D1	DUR	R1	FREQ*.56
A2	AMP*.67	D2	DUR*.9	R2	FREQ*.56+1
A3	AMP	D3	DUR*.65	R3	FREQ*.92
A4	AMP*1.8	D4	DUR*.55	R4	FREQ*.92+1.7
A5	AMP*2.67	D5	DUR*.325	R5	FREQ*1.19
A6	AMP*1.67	D6	DUR*.35	R6	FREQ*1.7
A7	AMP*1.46	D7	DUR*.25	R7	FREQ*2.
A8	AMP*1.33	D8	DUR*.2	R8	FREQ*2.74
A9	AMP*1.33	D9	DUR*.15	R9	FREQ*3
A10	AMP	D10	DUR*.1	R10	FREQ*3.76
A11	AMP*1.33	D11	DUR*.075	R11	FREQ*4.07

FIGURE 4.28 Bell instrument based on Risset. *(Based on example in Risset's Introductory Catalogue of Computer-Synthesized Sounds. Reprinted with permission of Jean-Claude Risset.)*

ily. The first five partials of bell tones approximate the following: a fundamental, a minor third, a perfect fifth, a "hum tone" at an octave below the fundamental, and the "nominal" at an octave above the fundamental. The ratios in frequency for this grouping of partials are 1:1.2:1.5:0.5:2. In his design, Risset extends the series to include higher partials, and tunes the partials to the following ratios—0.56:0.92:1.19:1.70:2:2.74:3:3.76:4.07.

The waveform of each component is a sinusoid and the envelope (F2) is an exponential decay from 1 to 2^{-10}. The duration used in the Sound Catalog is 20 seconds. When implementing the design suggested in the figure, it is advisable to use a method of "turning off" the oscillator pairs after their playing time has elapsed, in order to save computation time.

Another of Risset's designs from the *Computer Sound Catalog* is shown in figure 4.29. It represents a computer instrument that produces an "endless glissando" or Shepard tone. Psychologist Roger Shepard discovered that the apparent register of tones in musical scales could be made ambiguous by carefully controlling the amplitude of the partials of the tones. Shepard produced scales that were perceived as "circular" in pitch—while appearing to move continuously in one direction along the scale, they actually never left the register in which they began. Risset extended this principle to achieve the same effect with glissandoing tones as well.

The design is a highly controlled glissando configuration in which 10 interpolating oscillators track the same amplitude and frequency functions. Each sinusoidal oscillator is controlled by two interpolating oscillators sampling amplitude and frequency functions, respectively. The function F3, which controls the frequency, is exponential. This produces a constant change of musical interval per unit time. F3 decays from 1 to 2^{-10}, producing a frequency change of 10 octaves over its duration. Each pair of controlling oscillators has the same initial phase. However, their phase is offset by 1/10 of a cycle from the phase of a neighboring pair. This corresponds to a phase offset of 51.2 when using a wave table of 512 locations. Because F3 exponentially decays from 1 to 2^{-10}, the phase offset of 1/10 cycle results in the 10 oscillators glissandoing downward in parallel octaves. When an oscillator reaches the end of F3, it "wraps around" to the beginning of the function and continues. Ordinarily, such a large discontinuity in frequency (a 10-octave jump) would cause a click and destroy the effect of smooth glissandoing. However, during the transition, the amplitude function (F2) is at its minimum value, preventing our hearing the click. On the other hand, when a tone passes through the midrange, F2 greatly emphasizes it. The effect of summing the 10 sinusoidal oscillators together is that of a continually glissandoing tone in which no change of register occurs.

Risset has observed that the computer must have sufficient word length to accurately represent the phase in order to prevent noticeable roundoff error. For the acoustical illusion to be effective, a sufficient duration must be used. Risset chose 120 seconds for the completion of the entire cycle of 10 glissandos. He used the design and other closely related ones in his composition, *Mutations I* (see section 4.12).

A useful class of sounds for certain kinds of musical textures is choral tone, which is analogous to the effect in acoustic music of more than one instrument or voice playing a line in unison. A spectral analysis of a group of instruments playing in unison reveals a significant amount of spreading in components of the spectrum; that is, the energy of each component will be more widely distributed about its average frequency than when a single instrument is playing. This is the result of slight mistunings of the instruments and the lack of correlation among their vibratos.

The effect can be approximated by adding another copy of the computer instrument design at 1 or 2 Hz away from the original and then applying a small (approximately 1%) amount of random frequency deviation to both instruments. The randomness is best implemented with a noise generator that has most of its energy below 20 Hz. Also, because voices do not enter and exit at exactly the same times, a small amount of random deviation in their starting times and durations, as well as to the breakpoints of their envelopes, is desirable. Another method that uses delay lines is described in chapter 10.

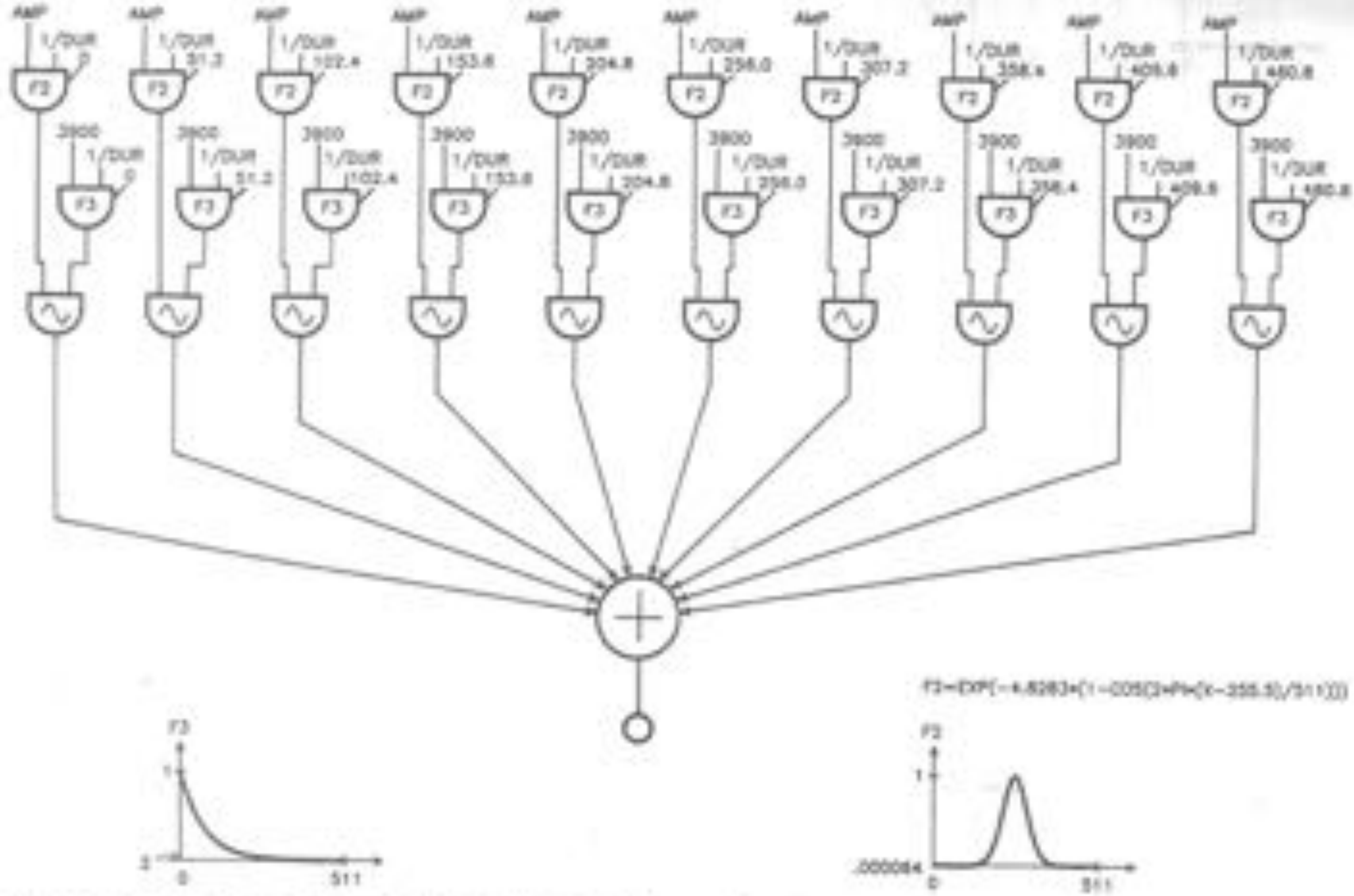


FIGURE 4.29 Design for endless glissando based on Risset. (Based on example in Risset's Introductory Catalogue of Computer-Synthesized Sounds. Reprinted with permission of Jean-Claude Risset.)

assumed to have an initial value of 0 because they start at the left side of the sine wave shown. This configuration, based on the pioneering work of Chowning, produces the particular spectrum described in this section. When other phase relationships exist between the oscillator waveforms, such as when one oscillator uses a sine wave and the other a cosine wave, the spectral components generated have different phases than the ones described here.[7] The difference in the resulting spectrum of the sound will be particularly audible when the N_1:N_2 ratio is such that the negative-frequency components combine with positive ones. The FM instrument designs appearing in this text produce the results described when the specific waveforms shown are used. Synthesis with other phase relationships between the carrier and modulating waveforms will not, in many cases, produce the same sound.

5.1C Obtaining Dynamic Spectra

Because FM has a single index for controlling the spectral richness of a sound, its use simplifies the synthesis of time-varying, dynamic spectra. The index of modulation controls the spectral content of an FM signal, and so an envelope applied to the index will cause the spectrum to change with time. Figure 5.7 illustrates a simple instrument that produces dynamic

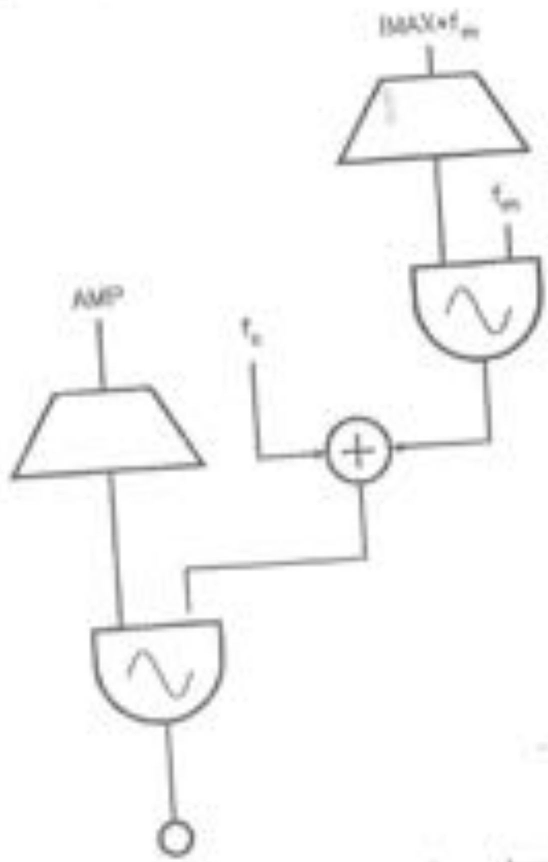


FIGURE 5.7 Simple FM instrument that produces time-varying spectra.

spectra. Notice that there are two separate envelopes, one for the spectrum and one for the amplitude. IMAX is the maximum value that the index will assume. To drive the modulating oscillator, IMAX is converted to a deviation by multiplying it by the modulating frequency.

The progression of the spectral components with index can be complicated when the effects of the folded negative sidebands are taken into account. By examining the shape of the Bessel functions in figure 5.3, it is not hard to see that the evolution of an FM signal generally has a certain amount of "ripple" in it. That is, as the index increases, the amplitude of any particular component will not increase smoothly, but instead will alternately increase and decrease, sometimes passing through 0. The amount of ripple is somewhat proportional to the maximum value of the modulation index. To demonstrate, figure 5.8 plots the time-varying spectrum produced by the instrument of figure 5.7 with the parameter values indicated. Observe the ripple in the evolution of the spectrum.

Unlike additive synthesis, frequency modulation allows only certain types of spectral evolutions. Ordinarily, it will not be possible for a musician to match on a point-by-point basis the component amplitudes of a spectrum obtained from an acoustic analysis. However, an effective strategy can be to select the spectral envelope that will realize the desired evolution of the overall richness, the bandwidth, of the spectrum. Because time evolution of the richness of the spectrum is an important element in the perception of timbre, a wide variety of tones can be synthesized by this technique.

5.1D Simple FM Instrument Designs

Figures 5.9 and 5.10 illustrate John Chowning's designs[8] for producing a variety of instrument-like tones with simple FM. On many systems, the functions F1 and F2 could be realized by using envelope generators instead of oscillators. The tone quality produced by the design can be varied by altering any of three factors: the ratio $f_c:f_m$, the maximum value of the modulation index (IMAX), and the function shapes for the

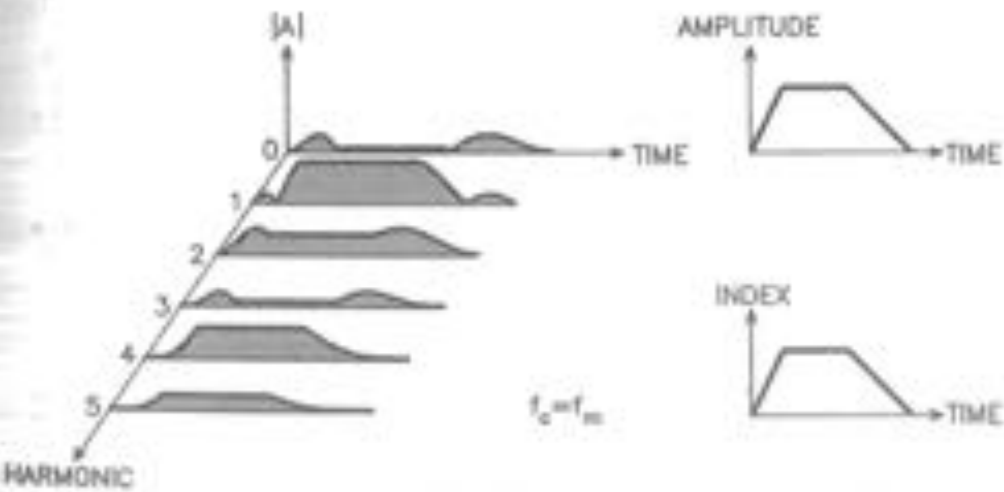


FIGURE 5.8 Dynamic spectrum produced by the instrument of figure 5.7.

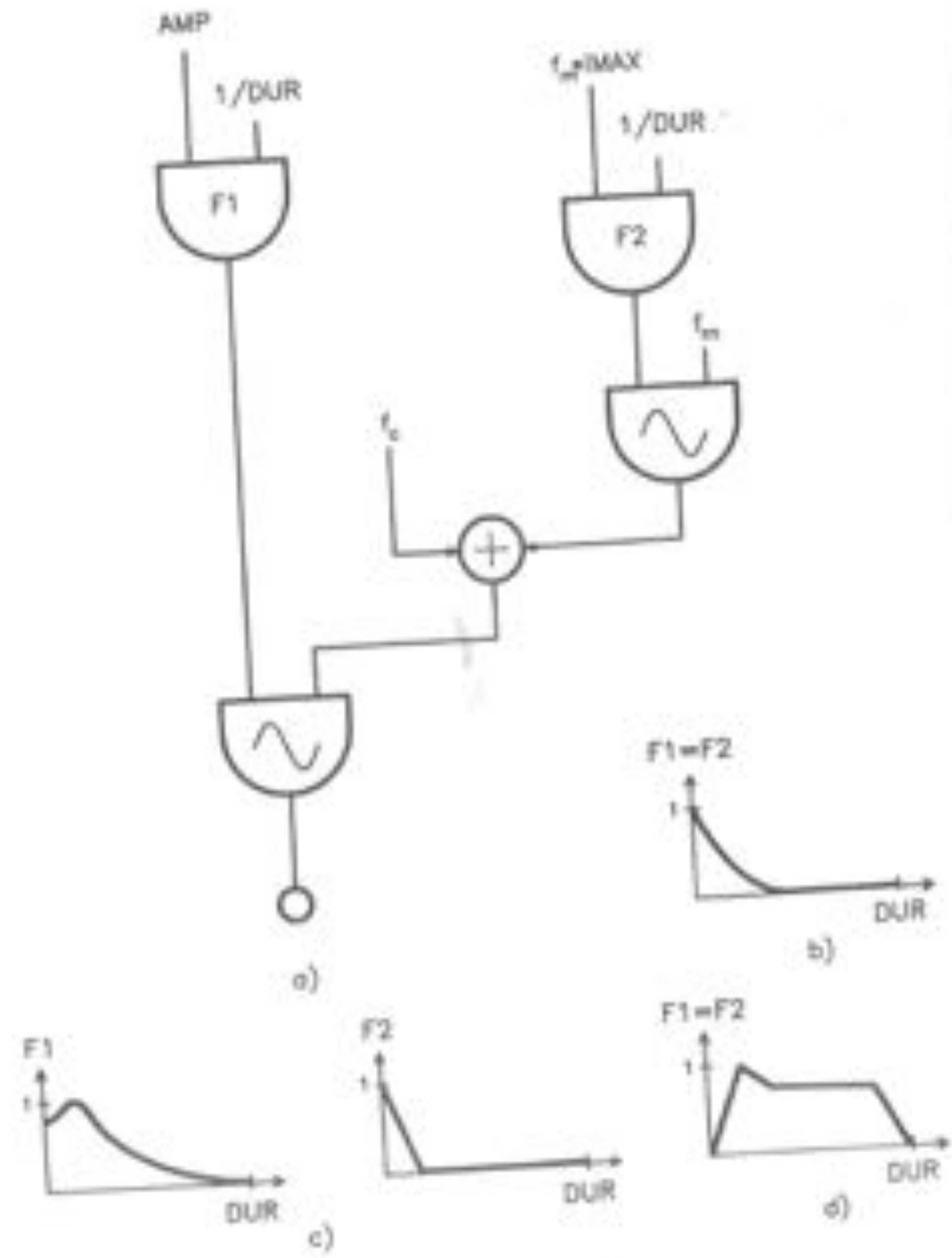


FIGURE 5.9 (a) Basic Chowning FM instrument; (b) function for bell-like timbre; (c) function for wood drum sound; and (d) function for brasslike timbre. (Based on design in "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," by John Chowning. Published in Journal of the Audio Engineering Society, 21(7), 1973. Reprinted with permission of the author.)

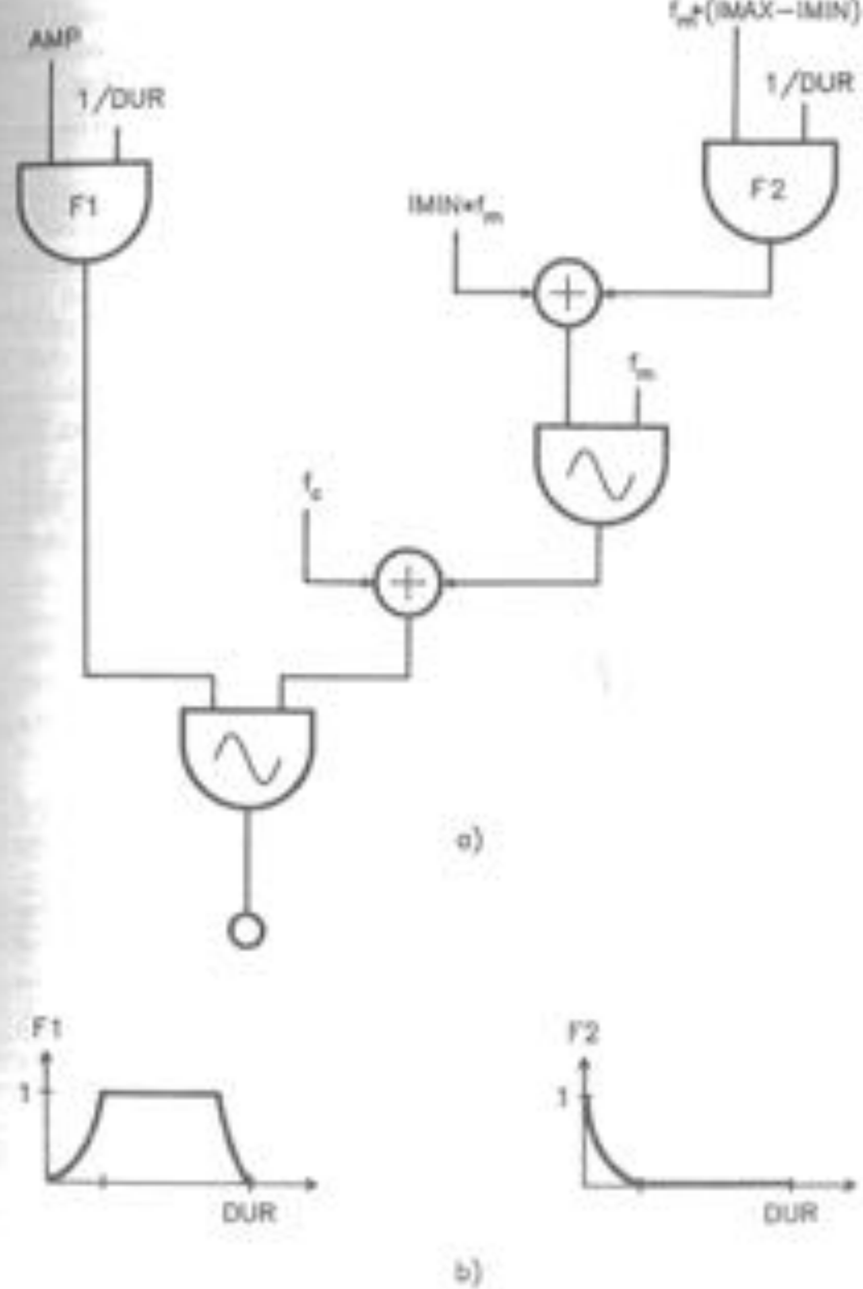


FIGURE 5.10 FM design based on Chowning for producing a clarinet-like timbre. (Based on design in *"The Synthesis of Complex Audio Spectra by Means of Frequency Modulation,"* by John Chowning. *Published* in Journal of the Audio Engineering Society, 21(7), 1973. *Reprinted with permission of the author.*)

amplitude and for the index of modulation. The amplitude parameter (AMP) should be scaled for all the examples to a value appropriate to the particular system used. The topology of figure 5.9 is used for the first three FM instrument designs that follow.

For bell-like tones, Chowning suggests the following parametric values:

DUR = 15 seconds
f_c = 200 Hz
f_m = 280 Hz (i.e., an f_c:f_m ratio of 5:7)
IMAX = 10

The function shown in figure 5.9b is used for both the amplitude envelope and the envelope applied to the index of modulation. The exponential decay of the amplitude is characteristic of bell sounds. The shape of the function applied to the index of modulation creates a rich, inharmonic spectrum at the beginning of the tone. During the decay, the bandwidth of the spectrum continually diminishes until, near the end of the tone, the sound is essentially a sine wave at the carrier frequency. To obtain a bell-like sound, the duration must not be made too short. If the "bell" is not allowed to ring out for at least 2 seconds, listeners will perceive this tone as more of a "clank."

For FM wood-drum like tones, Chowning recommends:

DUR = 0.2 second
f_c = 80 Hz
f_m = 55 Hz
IMAX = 25

The functions for wood-drum tones are shown in figure 5.9c. The function to control the index of modulation causes an inharmonic spectrum with wide bandwidth during the attack. After a short time, the index drops to 0 and the drum tone becomes simply a decaying sine wave at the carrier frequency, so that a clear pitch will be perceived at that frequency. The duration is a critical cue for these drum tones and cannot be longer than about 0.25 second without destroying the percussive effect. Raising the carrier frequency in the range of 200 Hz with the same f_c:f_m ratio produces a sound closer to that of a wood block.

FM brass-like tones can be produced with:

DUR = 0.6 second
f_c = 440 Hz
f_m = 440 Hz (an f_c:f_m ratio of 1:1)
IMAX = 5

The same envelope function (figure 5.9d) is applied to both the amplitude and the index of modulation. Lowering the value of IMAX to 3 yields a more muted brass tone.

Figure 5.10a shows the design for obtaining FM clarinet-like tones. Chowning suggests using the following values:

DUR = 0.5 second
f_c = 900 Hz
f_m = 600 Hz (an f_c:f_m ratio of 3:2)

meter (AMP) should be
cular system used. The
designs that follow.
netric values:

e envelope and the enve-
of the amplitude is char-
the index of modulation
se. During the decay, the
the end of the tone, the
obtain a bell-like sound,
allowed to ring out for at
clank."

3c. The function to control
vide bandwidth during the
e becomes simply a decay-
ll be perceived at that fre-
cannot be longer than about
he carrier frequency in the
ser to that of a wood block.

he amplitude and the index
ne muted brass tone.
-like tones. Chowning sug-

IMIN = 2
IMAX = 4

The shapes of the functions are shown in figure 5.10b. The fundamental frequency produced by this instrument will be f_c / 3 (300 Hz when using the values above). Notice that because the denominator of the $f_c : f_m$ ratio is 2, the resulting tone will contain no even harmonics. The use of two modulation indices ensures that the resulting modulation index will never drop below the value of IMIN. Increasing IMAX to 6 produces a more strident attack. A small, constant value may be added to the modulating frequency, causing the folded sidebands to beat with the upper sidebands. This technique can result in a more realistic tone.

5.1E Use of Two Carrier Oscillators

An important characteristic of many natural sounds is the presence of fixed formants. Without provision for them, several classes of sounds cannot be satisfactorily synthesized. Even when great care is exercised in choosing the parameters of an FM instrument so that a peak is placed at some desired point in the spectrum, the peak will be valid only for a small range of the values of the index of modulation. Also, the peak in the spectrum is not fixed; it will move with the fundamental frequency of the sound. Passing the signal from any instrument through a band-pass filter (see chapter 6) will yield an accurate, immobile formant, but a more economical and, in many systems, a more practical approach is described below. This method can only approximate fixed formants, but the results are often satisfactory.

The use of two carrier oscillators driven by a single modulating oscillator (figure 5.11) provides a means for formant simulation. The index of modulation of the first carrier oscillator is I1. The modulating signal delivered to the second carrier oscillator is multiplied by a constant (I2/I1) in order to provide a second index of modulation with the same time variation as the first. The second carrier oscillator produces a spectrum that is centered around the second carrier frequency. Because its index of modulation (I2) is typically small, the spectrum has its strongest component at the second carrier frequency. When the two FM signals are added together, the overall spectrum has a peak at the second carrier frequency. The audible effect is to add a formant to the sound. The amplitude of the second carrier oscillator is proportional to (and usually less than) the amplitude of the first by the factor A2. The relative strength of the formant can be adjusted by changing this parameter.

The second carrier frequency (f_{c2}) is chosen to be the harmonic of the fundamental frequency (f_0) that is closest to the desired formant frequency (f_f). Mathematically stated,

$$f_{c2} = nf_0 = \mathrm{int}\left(\frac{f_f}{f_0} + 0.5\right) f_0$$

That is, n is the ratio, rounded to the nearest integer, of the desired formant frequency to the fundamental frequency. The second carrier frequency remains harmonically related to f_0; the value of n changes with f_0 in order to keep the second carrier frequency as

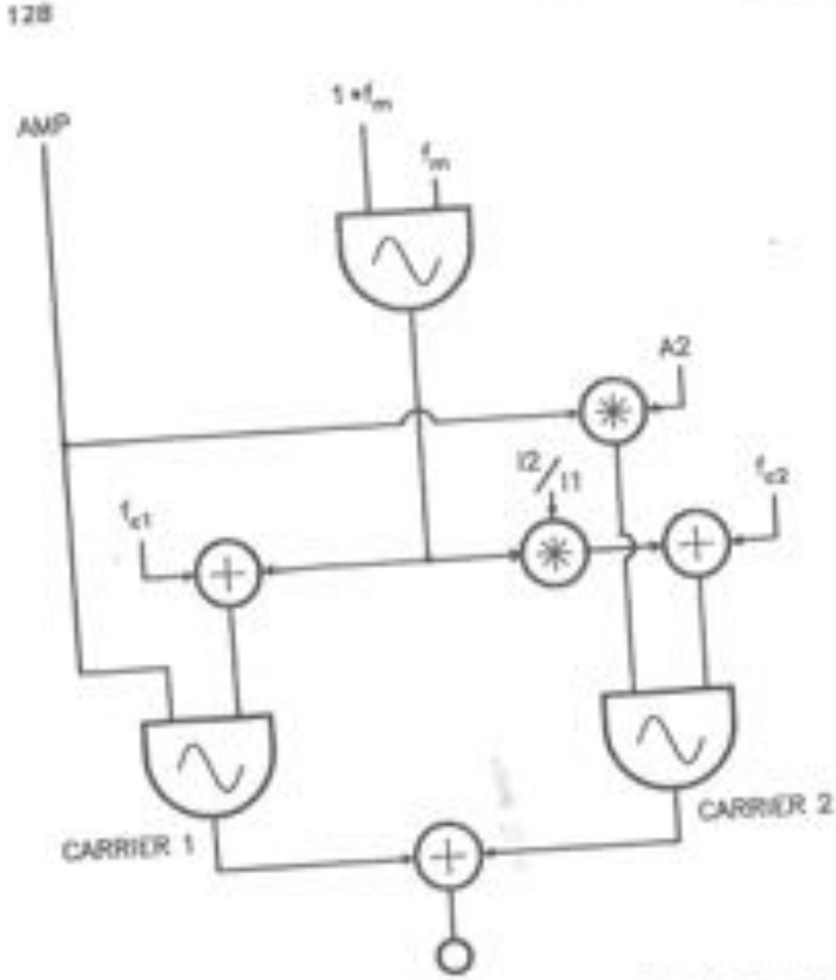


FIGURE 5.11 FM instrument employing two carrier oscillators for formant simulation.

close as possible to the desired formant frequency. For example, if the desired frequency is 2000 Hz, then for $f_0 = 400$ Hz, the fifth harmonic will be used. As f_0 is increased, f_{c2} will remain the fifth harmonic until f_0 becomes greater than 444.4 Hz, when the fourth harmonic will be closer to 2000 Hz.

5.1F Double-Carrier FM Instruments

Dexter Morrill has made extensive use of *double-carrier* FM in his computer synthesis of trumpet tones.[7] The design shown in figure 5.12 is based on one of his algorithms. For convenience, it is divided into a main instrument and a vibrato generator.

In the main instrument, the two carrier oscillators have frequencies at the fundamental frequency and the first formant frequency, respectively. The maximum value of the index of modulation for the first carrier oscillator is IMAX. The peak index of modulation for the second carrier oscillator is obtained by scaling the output of the modulating oscillator by the ratio of the second index to the first, IRATIO. The amplitude of

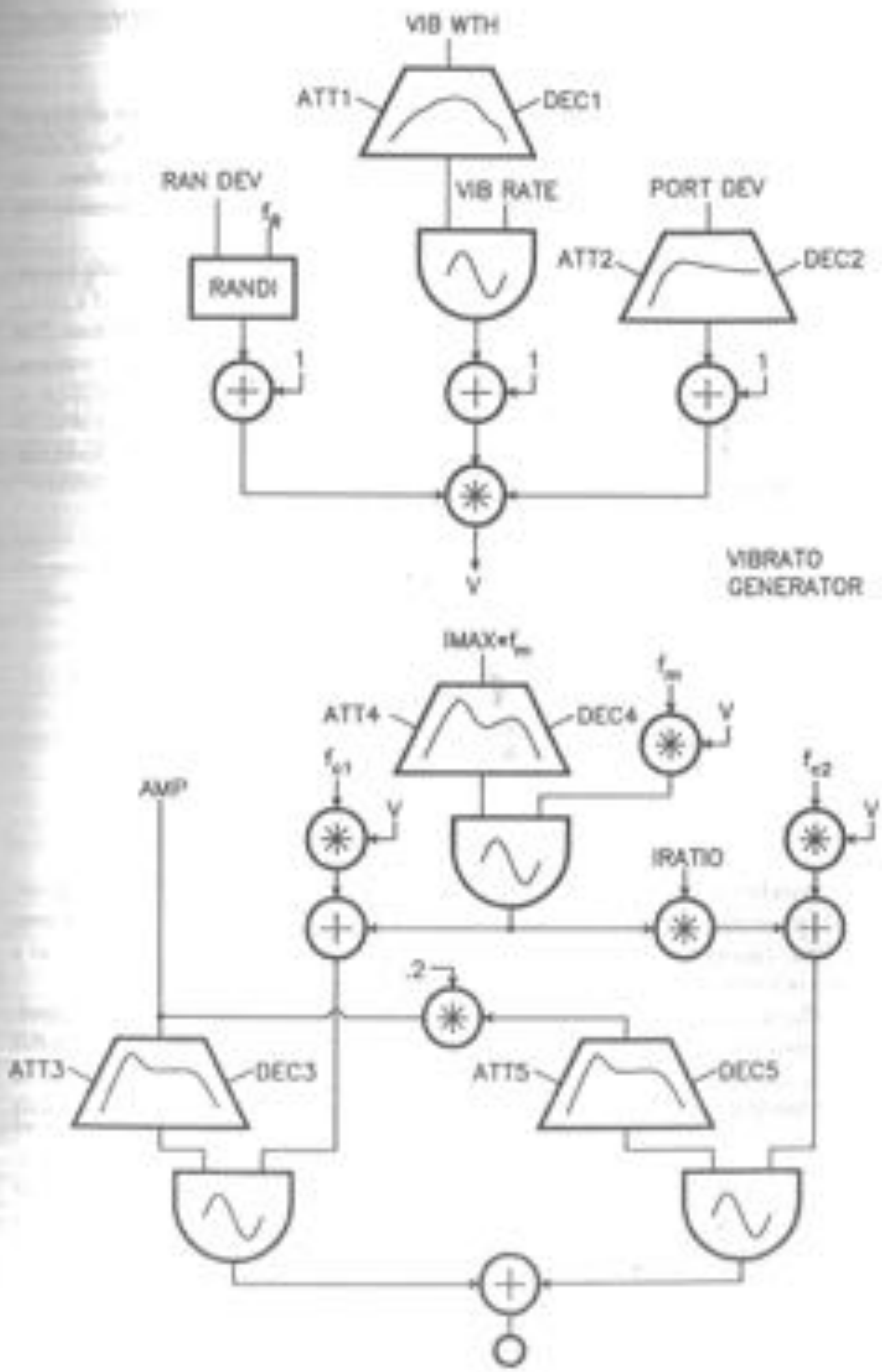


FIGURE 5.12 Double-carrier trumpet instrument. (*Adapted from Morrill's design. Reprinted with permission of Computer Music Journal.*)

the second carrier is 20% that of the first, thus setting the amplitude of the simulated formant at the desired level. The harmonics associated with the formant decay more quickly than the rest of the components of the trumpet tone. On many systems, the envelopes will have to be realized with function generators or oscillators because the shapes are too complex for a simple three-segment envelope generator.

The vibrato generator provides an additional frequency modulation of both carrier systems. It imparts a periodic vibrato, a small, random, frequency deviation, and a portamento to each trumpet tone. This part of the instrument ensures a more lifelike tone. The portamento frequency deviation keeps the pitch of the note from sounding too uniform. Its function shape determines the pattern of deviation, and its maximum deviation is PORT DEV. The random frequency deviation simulates one of the characteristics of trumpet tones that Risset describes. Vibrato is especially important on longer tones. Some values for the parameters of the instrument adapted from Morrill's example are as follows:

DUR = 1 second	DEC5 = 0.3 second
f_{c1} = 250 Hz	RANDEV = 0.007
f_{c2} = 1500 Hz	f_R = 125 Hz
f_m = 250 Hz	VIB WTH = 0.007
IMAX = 2.66	VIB RATE = 7 Hz
IRATIO = 1.8 / 2.66	PORT DEV = 0.03
ATT3 = 0.03 second	ATT1 = 0.6 second
ATT4 = 0.03 s econd	ATT2 = 0.06 second
ATT5 = 0.03 second	DEC1 = 0.2 second
DEC3 = 0.15 second	DEC2 = 0.01 second
DEC4 = 0.01 second	

In coding a score for this instrument, Morrill used Leland Smith's SCORE program to obtain, from one note to the next, random deviation of certain values of his parameter lists. This ensured that the succession of notes would not sound mechanical as a result of too great a uniformity of parameter values.

The principle of using more than one carrier frequency to simulate formants is extended in John Chowning's FM design to realize the female singing voice on vowels. In this design, Chowning uses two carrier oscillators and one modulating oscillator. The oscillators are all tuned in whole-number multiples of the same frequency. Chowning has discerned six characters of the singing soprano voice which guided his design for synthesis. They are:

1. There is a weighting of the spectral energy around the low-order harmonics with the fundamental as the strongest harmonic, thus supporting the theory that the lowest formant tracks the pitch period.
2. There are one or more secondary peaks in the spectrum, depending on the vowel and fundamental pitch, which corresponds to the resonances on the vocal tract or upper formants.
3. The formants are not necessarily at constant frequencies independent of the fundamental pitch, but rather follow formant trajectories which may either ascend or descend, depending on the vowel, as a function of the fundamental frequency.

4. The upper formants decrease in energy more rapidly than does the lowest formant when a tone is sung at a decreasing loudness.
5. Only the lowest formant is prominent at the amplitude thresholds of the attack and decay portions, while the upper formants only become pronounced as the overall amplitude of the signal approaches the quasi-steady state.
6. There is a small but discernible fluctuation of the pitch period even in the singing condition without vibrato.

Using these principles as a guide, Chowning designed the instrument shown in figure 5.13. The design resembles the trumpet of the previous example in that it uses two carrier oscillators and makes provision for vibrato with random deviation. The singing-soprano design also includes a slight portamento, but only during the attack portion of the note. The design includes a set of arrays that use the pitch of the note to determine values for the second formant frequency, the amplitude of the second carrier (A2), and the modulation indices for both carriers. A different set of arrays is used for each vowel. Following is an example of the sorts of values used for the design:

AMP = in the range of 0 < AMP < 1
PITCH = in the range of G3 < PITCH < G6
f_m = PITCH
A2 = the relative amplitude of the second carrier, between 0 and 1 in value—see figure 5.14
A1 = the relative amplitude of the first carrier = 1 − A2
f_{c1} = first carrier frequency = PITCH
f_{c2} = second carrier frequency = INT (f_2 / PITCH + 0.5), where f_2 is the frequency of the upper formant, obtained from figure 5.14
I1 = modulation index for the first carrier, computed from the data in figure 5.14
I2 = modulation index for the second carrier, computed from the data in figure 5.14
VIB WTH = 0.2 $\log_2$PITCH
VIB RATE = between 5 and 6.5 Hz, depending on PITCH

One of the striking features of Chowning's tapes produced with this instrument can be heard in the examples where the two carriers enter one at a time, 10 seconds apart. Ten seconds after the entrance of the second carrier, the vibrato with its random deviation is applied equally to the two carriers. Until the application of the vibrato, the two carriers do not fuse into a single aural image of a "voice." Chowning observes that by itself, "the spectral envelope does not make a voice."

5.1G Complex Modulating Waves

Up to this point, the only FM instruments that have been considered are those in which the waveform of each oscillator is a sine wave. While many interesting timbres can be synthesized this way, certain others require a more complicated modulating waveform. Because

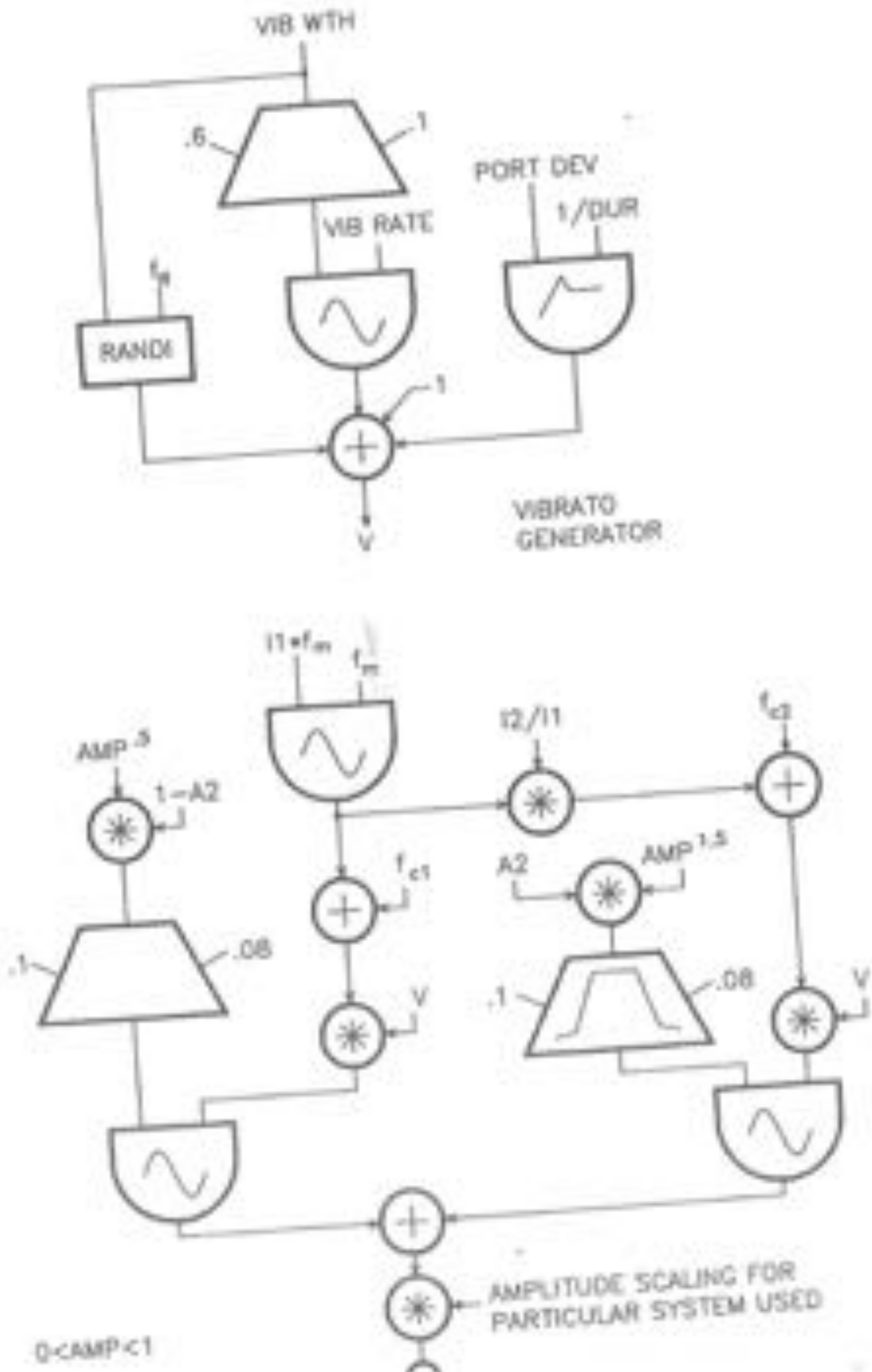


FIGURE 5.13 FM soprano instrument based on Chowning.

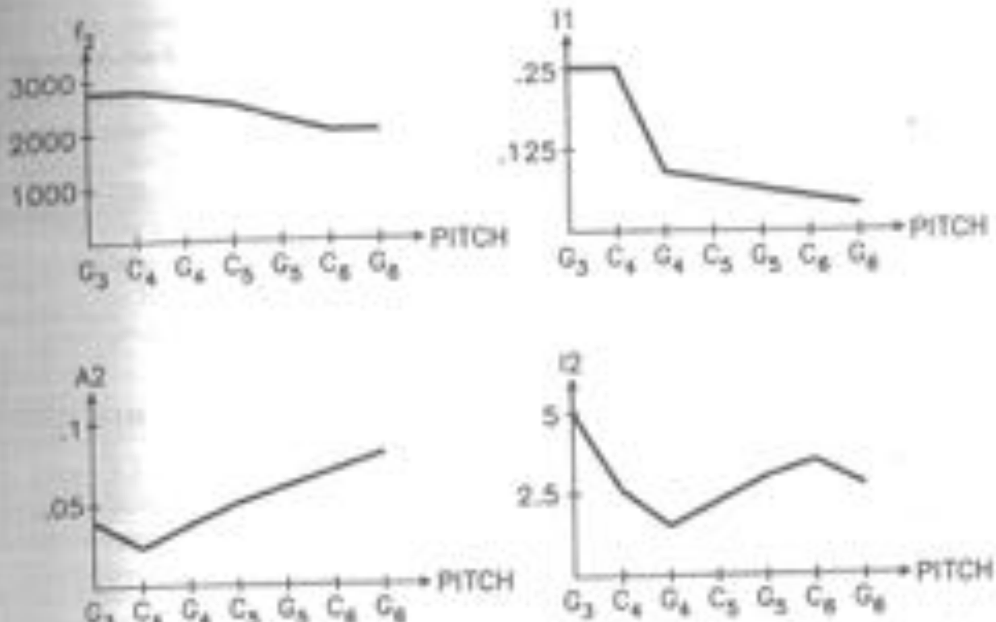


FIGURE 5.14 Parameters used by Chowning for FM soprano. (*From "Computer Synthesis of the Singing Voice," by John Chowning, in Sound Generation in Winds, Strings, Computers, Publication No. 29 of the Royal Swedish Academy of Music, edited by Johan Sundberg. Reprinted with permission of the editor.*)

the process of frequency modulation produces such rich spectra, a complex modulating wave generally needs to consist of no more than two or three spectral components. Figure 5.15 illustrates an instrument in which the frequency of the carrier oscillator is modulated by a complex wave that is the sum of two sine waves. In principle, this instrument could have been realized with a single modulating oscillator whose waveform was the appropriate combination of components, but employing independent oscillators permits variation of the relative amplitudes and frequencies of the modulation components.

The spectrum of this instrument will contain a large number of frequencies. If the carrier frequency is f_c and the modulating frequencies are f_{m1} and f_{m2}, then the resulting spectrum will contain components at the frequencies given by $f_c \pm i f_{m1} \pm k f_{m2}$, where i and k are integers greater than or equal to 0. To indicate when a minus sign is used, the value of i or k will be superscripted with a minus sign. For instance, the sideband with a frequency of $f_c + 2f_{m1} - 3f_{m2}$ will be denoted by the pair: $i = 2, k = 3^-$.

Independent indices of modulation can be defined for each component. I_1 is the index that characterizes the modulation that would be produced if only the first modulating oscillator were present; I_2 is that of the second. The amplitude of the ith, kth sideband ($A_{i,k}$) is given by the product of Bessel functions[11] as

$$A_{i,k} = J_i(I_1)J_k(I_2)$$

When i or k is odd and the minus sign is taken, the corresponding Bessel function assumes

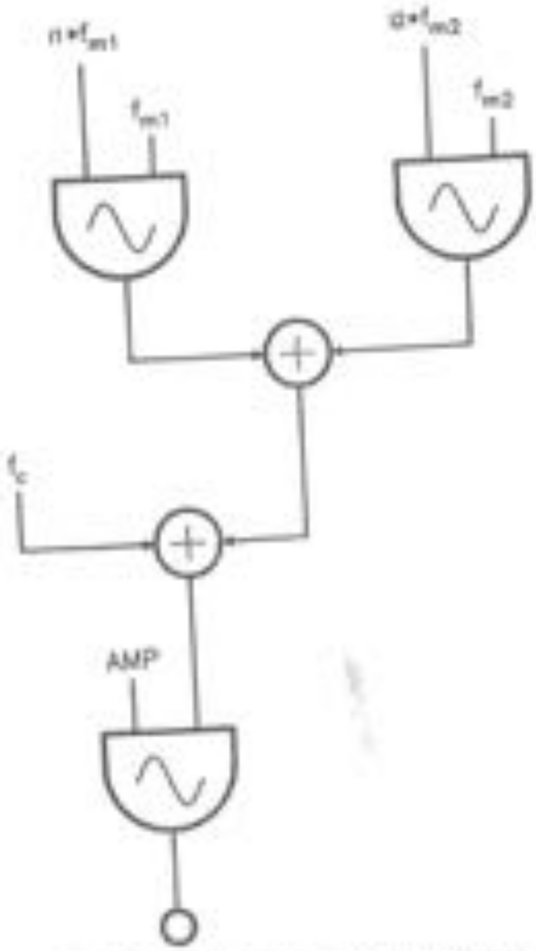


FIGURE 5.15 FM instrument with a complex modulating wave.

the opposite sign. For example, if $i = 2$ and $k = 3^-$, the amplitude is $A_{2,3} = -J_2(I_1)J_3(I_2)$. For $i = 3^-$, $k = 1^-$, the amplitude is $[-J_3(I_1)][-J_1(I_2)]$, and the two negative signs cancel when the two factors are multiplied.

In a harmonic spectrum, the net amplitude of a component at any frequency is the combination of many sidebands. As before, when a sideband has a negative frequency, it is folded around 0 Hz with a change of sign in its amplitude. For example, when f_c = 100 Hz, f_{m1} = 100 Hz, and f_{m2} = 300 Hz, the spectral component present in the sound at 400 Hz is the combination of sidebands given by the pairs: $i = 3, k = 0$; $i = 0, k = 1$; $i = 3^-, k = 2$; and so on. The components at −400 Hz come from $i = 2^-, k = 1^-$; $i = 1, k = 2^-$; $i = 5^-, k = 0$; and so on. The overall amplitude at 400 Hz, $A(400)$, is the sum of the amplitudes of all contributing components.

$$
\begin{aligned}
A(400) = {} & J_3(I_1)J_0(I_2) + J_0(I_1)J_1(I_2) - J_3(I_1)J_2(I_2) \\
& + J_2(I_1)J_1(I_2) - J_1(I_1)J_2(I_2) + J_5(I_1)J_0(I_2) \cdots
\end{aligned}
$$

The above expression shows only the three lowest-order terms for both the positive- and negative-frequency components. In fact, there are an infinite number of i, k dyads that produce a sideband at ±400 Hz. The actual number that contributes significantly to the overall amplitude is determined by the modulation indices. For given values of I_1 and I_2, one can calculate the maximum values of i and k for which the Bessel functions of that order have significant values. This information predicts the frequency of the highest significant component in the resulting spectrum, and so can be used to avoid aliasing.

Because so many sidebands are produced by this technique, lower indices of modulation can be used to obtain the same amount of spectral richness. This approach can be advantageous because the time evolution of the harmonics is smoother at lower indices.

5.1H FM Instrument with Complex Modulation

Figure 5.16 shows the flowchart for an instrument based on Bill Schottstaedt's simulation of stringlike tones with an FM design. The design entails a single-carrier oscillator and three modulating oscillators. It sounds best in the range of a cello when using the following parametric values.

f_c = pitch of the note	$I1 = 7.5 / \log_e f_c$
$f_{m1} = f_c$	$I2 = 15 / \sqrt{f_c}$
$f_{m2} = 3f_c$	$I3 = 1.25 / \sqrt{f_c}$
$f_{m3} = 4f_c$	

Notice that the indices of modulation vary with the fundamental frequency of the tone. Multiplying the modulation indices by 2 or 3 produces more strident tones, according to the designer.

A vibrato is implemented, similar to the one used in the trumpet design, except that here it is without the undershoot of the pitch of the tone. The *chiff*, or attack noise of the bow, is simulated with a noise band centered at 2000 Hz using a bandwidth of 20% of the carrier frequency. The function applied to the noise causes it to cease 0.2 second after the attack of the note.

The increase in spectral richness of the tone during the attack is simulated by adding the envelope F1 to each index of modulation. In his article, Schottstaedt also includes some advice for making other string instrument effects, such as *sul ponticello*, pizzicato, and the choral effect.

5.1I Compositional Examples

The compositions by John Chowning represent distillations of extensive research into computer techniques for the synthesis of sound. His expertise in acoustics and psychoacoustics plays a major role in formulating his pieces, as does his interest in expressing physical phenomena mathematically. Yet, artistically, each of his compositions is highly unified—usually around a particular technique or relationship. For example, *Turenas* (1972) demonstrates the travel of sound in a quadraphonic space (see section 10.3B); *Stria* (1976) illustrates the use of computer synthesis of sound to interrelate the small-scale sound design of

applied to the input will be transmitted to the output with its amplitude reduced by one-half.
A transfer function is nonlinear when it has terms that contain x raised to a power other than
1, or when it contains other mathematical functions of x, such as the logarithm or tangent. A
simple example of a nonlinear transfer function is $F(x) = x^3$ (figure 5.23b). In this case, the
output will be a distortion of the input because the value of each input sample is cubed as it
is passed to the output. Thus, doubling the input causes the output to increase by eight times.

To control the maximum harmonic in the spectrum and, hence, avoid aliasing in
waveshaping synthesis, a transfer function is normally expressed as a polynomial, which
has the following form:

$$F(x) = d_0 + d_1x + d_2x^2 + \cdots + d_Nx^N$$

The value of each coefficient d_i ($d_0, d_1, d_2, \ldots$) is chosen by one of the methods given below.
The degree or order of the polynomial is N, the value of the largest exponent. When driven
with a sine or cosine wave, a waveshaper with a transfer function of order N produces no
harmonics above the Nth harmonic. The musician can, therefore, predict the highest harmonic frequency generated and avoid aliasing either by limiting the frequency of the oscillator that drives the waveshaper or by tailoring the transfer function accordingly.

5.2C Calculating an Output Spectrum from a Transfer Function

When the nonlinear processor is driven by a sinusoidal waveform, the amplitudes of the
various harmonics of the output can be calculated from the transfer polynomial using
table 5.2. The table shows the amplitudes of the harmonics produced by a term in the
polynomial when the amplitude of the driving sinusoid is 1. The symbols along the top

	DIV	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
x^0	0.5	1											
x^1	1		1										
x^2	2	2		1									
x^3	4		3		1								
x^4	8	6		4		1							
x^5	16		10		5		1						
x^6	32	20		15		6		1					
x^7	64		35		21		7		1				
x^8	128	70		56		28		8		1			
x^9	256		126		84		36		9		1		
x^{10}	512	252		210		120		45		10		1	
x^{11}	1024		462		330		165		55		11		1

TABLE 5.2 Harmonic amplitudes produced by each algebraic term of a transfer function

of the table, denoted in the form h_j, represent the amplitude of the jth harmonic. Each line in the table has an associated divisor (DIV); the true value of any entry is the listed value divided by the divisor for that line.

As an example, suppose the transfer function $F(x) = x^5$ is driven by a cosine wave with an amplitude of 1. That is, the variable x signifies the cosine wave. The table shows that the output will contain the first, third, and fifth harmonics with the following amplitudes:

$$h_1 = \tfrac{1}{16}(10) = 0.625$$

$$h_3 = \tfrac{1}{16}(5) = 0.3125$$

$$h_5 = \tfrac{1}{16}(1) = 0.0625$$

For mathematical convenience, the amplitude of the zeroth harmonic is equal to $h_0/2$. Thus, the value in the table is two times larger than the actual amplitude of this particular harmonic. This term has a frequency of 0 Hz, and therefore serves to offset the output signal by a fixed amount. That is, it is a constant value added to each sample of the signal. Since a constant does not fluctuate, it contributes nothing to the perceived sound. This phenomenon needs to be taken into account, however, since a large offset can cause the sample values to exceed the range of the system without causing the signal to sound louder.

When a polynomial has multiple terms, the output is the sum of the contributions of each term. For instance, if $F(x) = x + x^2 + x^3 + x^4 + x^5$ is driven with a unit-amplitude sinusoid, the amplitudes of the components in the output spectrum are

$$\frac{h_0}{2} = \tfrac{1}{2}\left[\tfrac{1}{2}(2) + \tfrac{1}{8}(6)\right] = 0.875$$

$$h_1 = 1 + \tfrac{1}{4}(3) + \tfrac{1}{16}(10) = 2.375$$

$$h_2 = \tfrac{1}{2}(1) + \tfrac{1}{8}(4) = 1.0$$

$$h_3 = \tfrac{1}{4}(1) + \tfrac{1}{16}(5) = 0.5625$$

$$h_4 = \tfrac{1}{8}(1) = 0.125$$

$$h_5 = \tfrac{1}{16}(1) = 0.0625$$

The reader might recognize table 5.2 as the right side of Pascal's triangle and the entries to be the binomial coefficients. The table can be extended by adding two adjacent numbers on the same line and writing the sum below the space between them. The value of h_0 is twice the value of h_1 from the previous line, and the divisor is increased by a factor of two each time. For example, for x^{12}, the divisor would be 2048, $h_0 = 2 \times 462 = 924$, $h_2 = 462 + 330 = 792$, and so on.

As expected, the table shows that a particular term of the polynomial does not produce harmonics with numbers greater than its exponent. It further indicates that an even power of x will produce only even harmonics and that an odd power generates only odd harmonics. This characteristic is another advantage of polynomial representation

because it affords the instrument designer independent control of the odd and even har-
monics of a sound.
Thus far, the analysis has been applied for a sinusoidal input whose amplitude is 1.
What, then, is the general relationship between the amplitude of the input sinusoidal
waveform and the amplitude of a given harmonic? Let the input to a waveshaper be a
cosine wave with an amplitude of a. The output in polynomial form becomes

$$F(ax) = d_0 + d_1 ax + d_2 a^2 x^2 + \ldots d_N a^N x^N$$

where x, as above, symbolizes a cosine wave with an amplitude of 1. The harmonic
amplitudes can still be determined using table 5.2, but the contribution of each term is
multiplied by a, raised to the appropriate power. The dependence of the amplitude of
any harmonic on the value a is usually indicated by writing the amplitude of the jth har-
monic as a function, $h_j(a)$. Using the example of $F(x) = x + x^3 + x^5$ and substituting ax
for x, the harmonics at the output will be calculated as:

$$h_1(a) = a + \frac{1}{4} a^3(3) + \frac{1}{16} a^5(10)$$

$$h_3(a) = \frac{1}{4} a^3 + \frac{1}{16} a^5(5)$$

$$h_5(a) = \frac{1}{16} a^5$$

These harmonic amplitudes are plotted against the value of a in figure 5.24. Notice that
the spectrum becomes richer as the value of a increases. Because it exercises so much
control over the richness of the spectrum, a is called a *distortion index*. It is analogous to
the index of modulation of FM synthesis, in that it is a single parameter that determines
the spectral content. For simplicity, the index is often limited to values between 0 and 1.
When the waveform applied to the waveshaper is nonsinusoidal, the resulting spec-
trum is more difficult to predict and is often not band-limited. The spectrum of a wave-
form produced by passing a complex waveform through a nonlinear processor cannot be
calculated simply as the sum of the distortion applied to the individual spectral compo-
nents of the input waveform. Instead, a Fourier transform is performed on the output
waveform to determine its spectrum. Thus, for reasons of conceptual simplicity and to
obtain band-limited spectra, the majority of waveshaping instruments use a sinusoidal
oscillator to drive the nonlinear processor.

5.2D Selecting a Transfer Function

The success of an instrument that uses waveshaping synthesis is largely dependent on
the choice of transfer function. There are three principal approaches to the choice: spec-
tral matching, graphical, and heuristic.
In *spectral matching*, a transfer function is determined that will make the output of
the waveshaper match a desired steady-state spectrum for a particular value of the distor-
tion index. This can be accomplished through the use of *Chebyshev polynomials*, usually
denoted as $T_k(x)$, where k is the order of the polynomial. (T signifies that it is a Chebyshev
polynomial of the first kind.) These polynomials have the useful property that when the

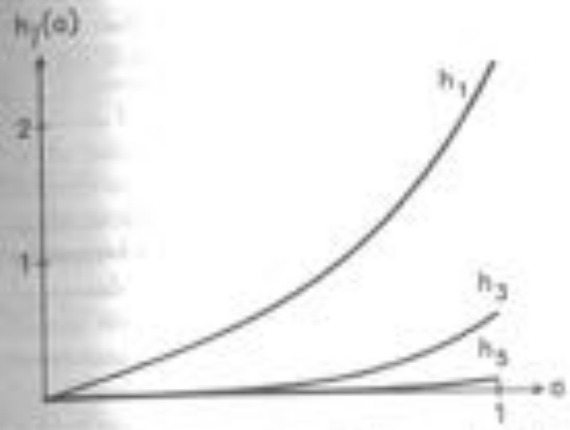


FIGURE 5.24 Harmonics amplitude versus distortion index for $F(x) = x^5 + x^3 + x$.

transfer function of a waveshaper is $T_k(x)$ and a cosine wave with an amplitude of 1 is applied to the input, the output signal contains only the kth harmonic.* For example, a transfer function given by the seventh-order Chebyshev polynomial results in an output of a sinusoid at seven times the frequency of the input. (This relationship exists only when the amplitude of the input is 1.) Table 5.3 lists the Chebyshev polynomials through the 11th order. Higher-order polynomials can be generated from the relationship:

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$$

A spectrum containing many harmonics can be matched by combining the appropriate Chebyshev polynomial for each desired harmonic into a single transfer function.

$$T_0(x)=1$$
$$T_1(x)=x$$
$$T_2(x)=2x^2-1$$
$$T_3(x)=4x^3-3x$$
$$T_4(x)=8x^4-8x^2+1$$
$$T_5(x)=16x^5-20x^3+5x$$
$$T_6(x)=32x^6-48x^4+18x^2-1$$
$$T_7(x)=64x^7-112x^5+56x^3-7x$$
$$T_8(x)=128x^8-256x^6+160x^4-32x^2+1$$
$$T_9(x)=256x^9-576x^7+432x^5-120x^3+9x$$
$$T_{10}(x)=512x^{10}-1280x^8+1120x^6-400x^4+50x^2-1$$
$$T_{11}(x)=1024x^{11}-2816x^9+2816x^7-1232x^5+220x^3-11x$$

TABLE 5.3 Chebyshev polynomials through the 11th order

Let h_j represent the amplitude of the jth harmonic in the spectrum to be matched and let N be the highest harmonic in that spectrum. The transfer function is then calculated as

$$F(x) = \frac{h_0}{2}T_0(x) + h_1T_1(x) + h_2T_2(x) + \cdots + h_NT_N(x)$$

As an example of spectral matching, suppose that when the distortion index equals 1, it is desired that the spectrum contain only the first, second, fourth, and fifth harmonics with amplitudes of 5, 1, 4, and 3, respectively. The transfer function to realize this is:

$$\begin{aligned} F(x) &= 5T_1(x) + 1T_2(x) + 4T_4(x) + 3T_5(x) \\ &= 5x + 1(2x^2 - 1) + 4(8x^4 - 8x^2 + 1) + 3(16x^5 - 20x^3 + 5x) \\ &= 48x^5 + 32x^4 - 60x^3 - 30x^2 + 20x + 3 \end{aligned}$$

When the distortion index assumes a value other than 1, this transfer function generates different spectra. The relationships between the various harmonic amplitudes and the distortion index, a, can be calculated from the transfer function using table 5.2, as before. These amplitudes are plotted three-dimensionally versus index and harmonic number in figure 5.25. Observe that the third and zeroth harmonics are not present when $a = 1$, but at other values of the distortion index, they are no longer balanced out. At small values of the distortion index, the spectrum is dominated by the fundamental and zero-frequency term.

Another more intuitive means of selecting the transfer function is based on the choice of its graphical shape. Some general principles relating the shape of the transfer function to the harmonics produced were given above. To avoid aliasing, the selected graphical shape should be approximated by a polynomial, with the order of the approxi-

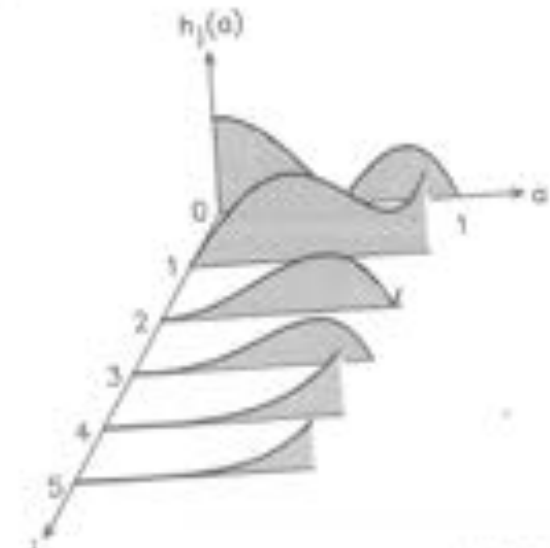


FIGURE 5.25 Harmonic amplitudes versus distortion index for $F(x) = 48x^5 + 32x^4 - 60x^3 - 30x^2 + 20x + 3$.

imation set to the highest harmonic number desired. The computer can be programmed to perform this task.

A heuristic selection of the coefficients of the transfer polynomial can sometimes be effective. For best results, the instrument designer should choose the signs of the terms of the polynomial carefully. If the signs of all the terms are the same, the function will become increasingly steep with the distortion index, resulting in exceptionally bright timbres. More subdued timbres can be obtained by following the pattern found in the Chebyshev polynomials where the signs of the even-and-odd-order terms alternate independently. This will yield a flatter transfer function that produces a less brilliant sound. With experience, the musician can develop an intuitive understanding of the relationship between the coefficients and the sound produced. Interactively altering a set of coefficients originally produced by the combination of Chebyshev polynomials is a good way to begin to develop this skill.

5.2E Considerations for Dynamic Spectra

As in FM synthesis, waveshaping facilitates the production of dynamic spectra because the distortion index can be varied with time. Of course, the transfer function used determines the amount and type of spectral controls possible. It would be simplest and probably the most musically useful if the index increased the amplitudes of all the harmonics smoothly. Polynomials that do this without producing extraordinarily brassy sounds are uncommon. If the polynomial is chosen for a less brilliant spectrum, there is often considerable ripple in the evolution of the harmonics with distortion index (unless the spectrum has an unusually small amount of harmonic energy). The higher the order of the transfer function, the harder it will be to obtain smooth spectral evolutions. Also, polynomials that are obtained by matching a spectrum in which the highest harmonics have relatively large amplitudes tend to produce much more spectral ripple than those where the harmonic amplitudes diminish with harmonic number.

If spectral matching is used to determine the transfer function, the smoothness of the spectral evolution can be improved by selecting the signs of the harmonic of the desired spectrum such that the signs of the even and odd harmonics are alternated independently.[*] Thus, the even harmonics would have the following pattern, starting with the zeroth: +, −, +, −, ... The odd harmonics, starting with the first, would take the same form. When the even and odd harmonics are combined to make a complete spectrum, the overall pattern becomes: +, +, −, −, +, +, −, −, ... Applying this method to the example of spectral matching given above yields a spectrum to be matched that contains only the first, second, fourth, and fifth harmonics with amplitudes of 5, −1, 4, and 3, respectively. The resulting transfer function is

$$F(x) = 48x^5 + 32x^4 - 60x^3 - 34x^2 + 20x + 5$$

While the spectrum at the value of the distortion index where the matching takes place is not audibly affected, this method results in a smoother spectral evolution.

As another example, the evolution of the harmonics with distortion index is plotted for two different cases in figure 5.26. Figure 5.26a shows the result when the spectral

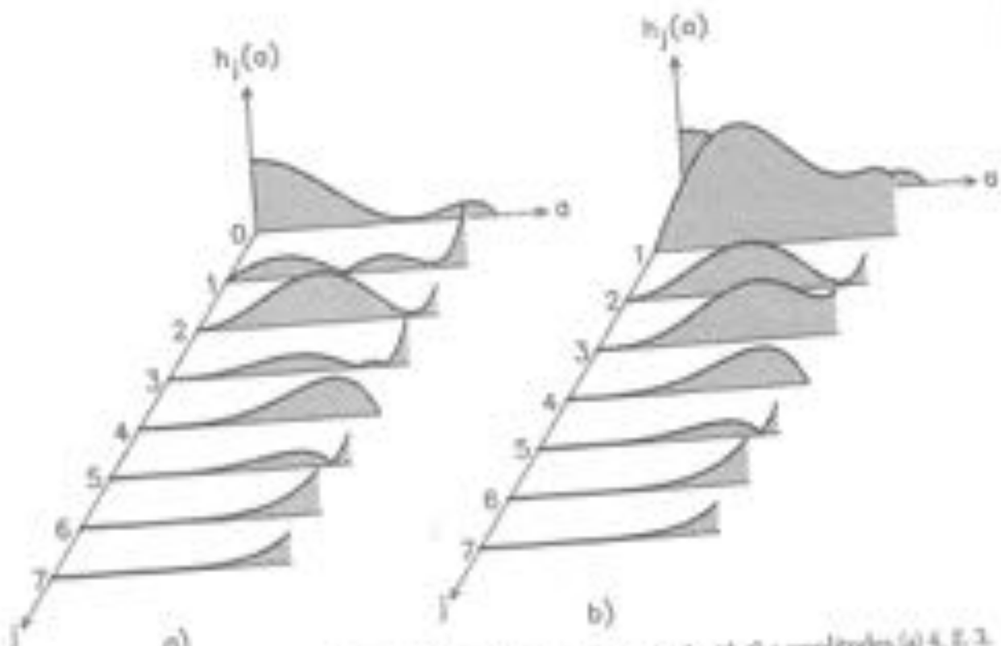


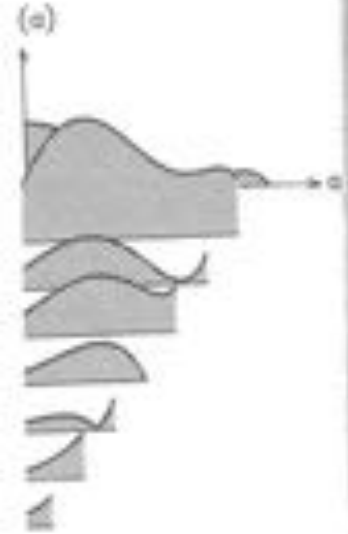
FIGURE 5.20 Harmonic evolution when a spectrum is matched with the amplitudes (a) 4, 2, 3, 0, 2, 3, 2 and (b) 4, −2, −3, 0, 2, −3, −2.

amplitudes, starting with the first harmonic, 4, 2, 3, 0, 2, 3, 2, are matched. Figure 5.20b illustrates the evolution when 4, −2, −3, 0, 2, −3, −2 are matched. Observe the vastly different behavior of the first and third harmonics with distortion index.

5.2F Implementation of a Nonlinear Processor

It is not practical to have the computer calculate the value of the output of the waveshaper by directly evaluating the transfer function for each sample. It is more efficient to calculate values of the transfer function for a number of input amplitudes and to store them in a table prior to performance. (This approach is quite similar to storing a waveform in a wave table that is referenced by a digital oscillator.) In the waveshaper, location in the table of the output value is obtained from the amplitude of the input signal. The instrument designer must ensure that the samples entering a waveshaper fall within the range of locations of the table. For example, if a 512-entry table has its locations marked by the integers 0 to 511, then the output of the oscillator driving the waveshaper must be scaled accordingly. Examples of this appear in section 5.2H.

As demonstrated for the digital oscillator (section 4.3), the use of a finite table size causes small inaccuracies in the output value when the input amplitude falls between entries in the table. In a waveshaper, these inaccuracies are heard as slight increases in the amount of harmonic distortion, and therefore are seldom objectionable.



with the amplitudes (a) 4, 2, 3,

, are matched. Figure 5.26b
hed. Observe the vastly dif-
ion index.

of the output of the wave-
sample. It is more efficient
nput amplitudes and to store
te similar to storing a wave-
x.) In the waveshaper, loca-
mplitude of the input signal
ring a waveshaper fall with-
-entry table has its locations
lator driving the waveshaper
tion 5.2H.
the use of a finite table size
put amplitude falls between
e heard as slight increases in
an objectionable.

5.2G Amplitude Scaling

The distortion index also controls the amplitude of the sound, so that changes in spectrum will be accompanied by changes in loudness. This relationship corresponds to the behavior of acoustic instruments, where louder playing ordinarily produces more overtones. Although this general relationship between amplitude and spectrum is correct, it is rare that the desired spectral and amplitude envelopes can both be obtained from a single envelope, particularly when the transfer function provides large amounts of distortion. Excessive loudness variation with spectral content often results, producing excessively bright tones. The amplitude variation with spectral content can be so extreme that the effective dynamic range of the tone of the sound exceeds the limitations of the system. When a secondary amplitude envelope is placed on the output of the waveshaper, the overall envelope of the sound becomes the product of the distortion index and the amplitude envelopes. When the interdependence between amplitude and spectrum is tempered, the amplitude envelope can be more easily chosen.

One method of compensation is to multiply the output of the waveshaper by a *scaling factor* which is a function of the distortion index.[17] If $S(a)$ is the scaling function, then the output is $OUT(ax) = F(ax)S(a)$. Because the scaling function is almost always complicated, it is impractical to calculate its value every time the distortion index is changed. Instead, it is stored as the transfer function of a second nonlinear processor, which is driven directly by the distortion index, *a*. Thus, in a waveshaping instrument like the one shown in figure 5.27, the distortion index is applied to both the sinusoidal oscillator and the processor containing the scaling function.

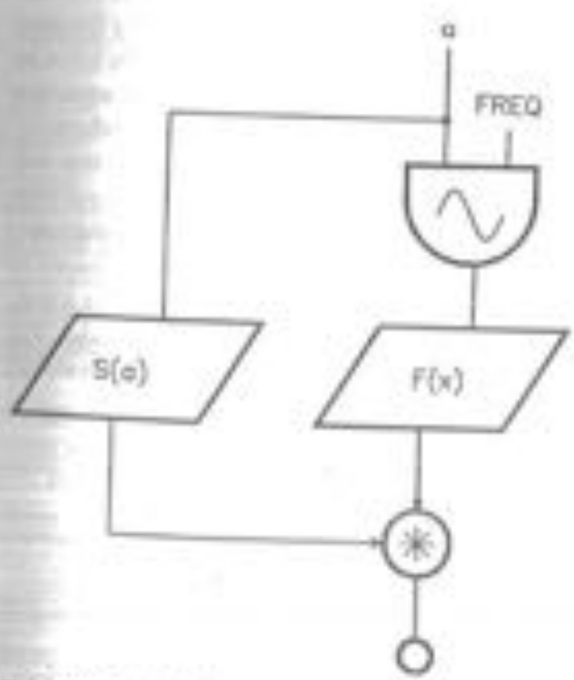


FIGURE 5.27 Implementation of a scaling function by means of second nonlinear processor.

The scaling is applied to the output signal according to some criterion. A practical method is to equalize the power in the signal to a constant value. For true power scaling, the scaling function is chosen as

$$S(a) = \frac{1}{\sqrt{\frac{1}{4}h_0^2(a) + h_1^2(a) + \cdots + h_N^2(a)}}$$

The relative strengths of the harmonics still vary with the distortion index just as they would without scaling, but the amplitude of the entire spectrum is raised or lowered by the appropriate scaling factor to keep the output power constant. However, this method does not work when the distortion index equals 0 because the scaling function becomes infinite. The instrument designer may choose to work around this by setting S(0) to a large, but noninfinite, value.

Peak scaling defines a scaling function where the peak amplitude of the output waveform is always the same. With a distortion index of a, the input to the waveshaper assumes values between −a and +a. To peak-scale the output, it is divided by the maximum magnitude of the transfer function for that range of inputs. Thus, the scaling function is

$$S(a) = \frac{1}{\max|F(\alpha)|_{-a \le \alpha \le a}}$$

The audible effect of this kind of scaling varies tremendously with the shape of the transfer function. When the transfer function is either continuously increasing or decreasing (i.e., has no peaks or valleys), this scaling method tends to equalize the power to some extent, although not with the accuracy of true power scaling. When the transfer function has ripples, an unusual relationship between amplitude and spectrum results because the scaling function has several plateaus yielding a limited amount of power equalization for some regions of distortion index and considerable power variation in others.

The use and type of a scaling function depends on the choice of transfer function and the type of sound to be synthesized. As in FM synthesis, waveshaping with true power scaling allows the amplitude and spectral envelopes to be independent, but at the same time eliminates one of the musical advantages of waveshaping synthesis—the change in spectral content with amplitude. A strategy that is often more effective is to choose a scaling function providing a more restrictive variation of loudness with distortion index. For some transfer functions, this can be done with peak scaling; in other cases, a modified power scaling function is more effective.

5.2H Example Waveshaping Instruments

Jean-Claude Risset designed and used in composition a nonlinear waveshaping instrument in the late 1960s.[*] The design, shown in figure 5.28, creates clarinet-like tones through nonlinear distortion of sine tones into tones with harmonic series containing only the odd-numbered partials.

The transfer function of the waveshaper is stored in a table with a length of 512 loca-

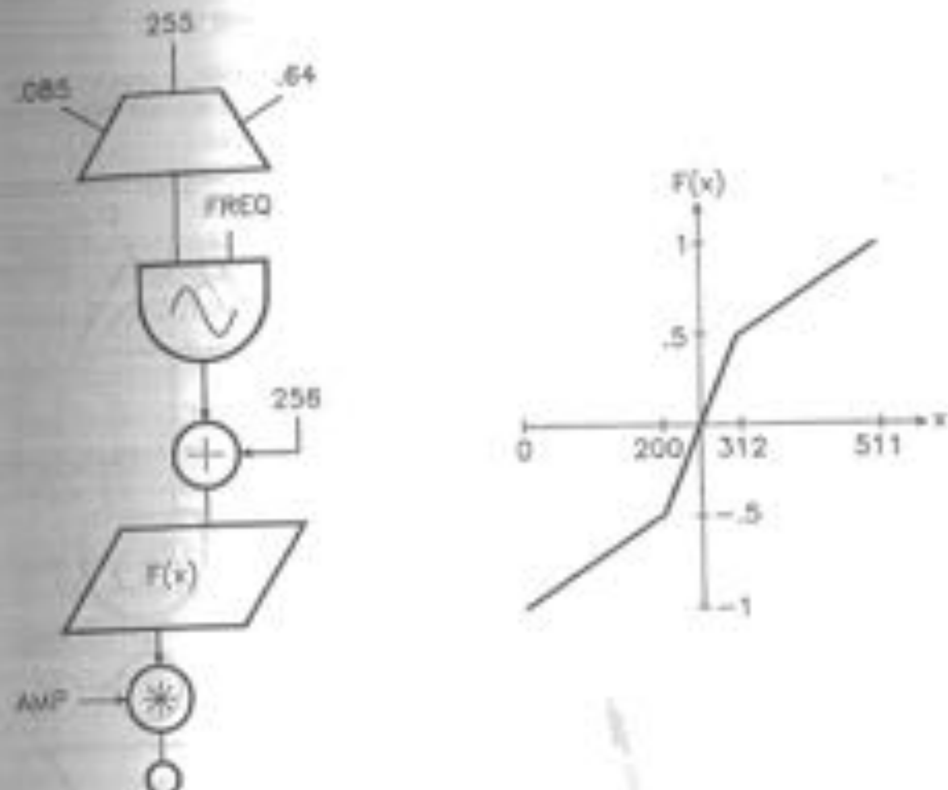
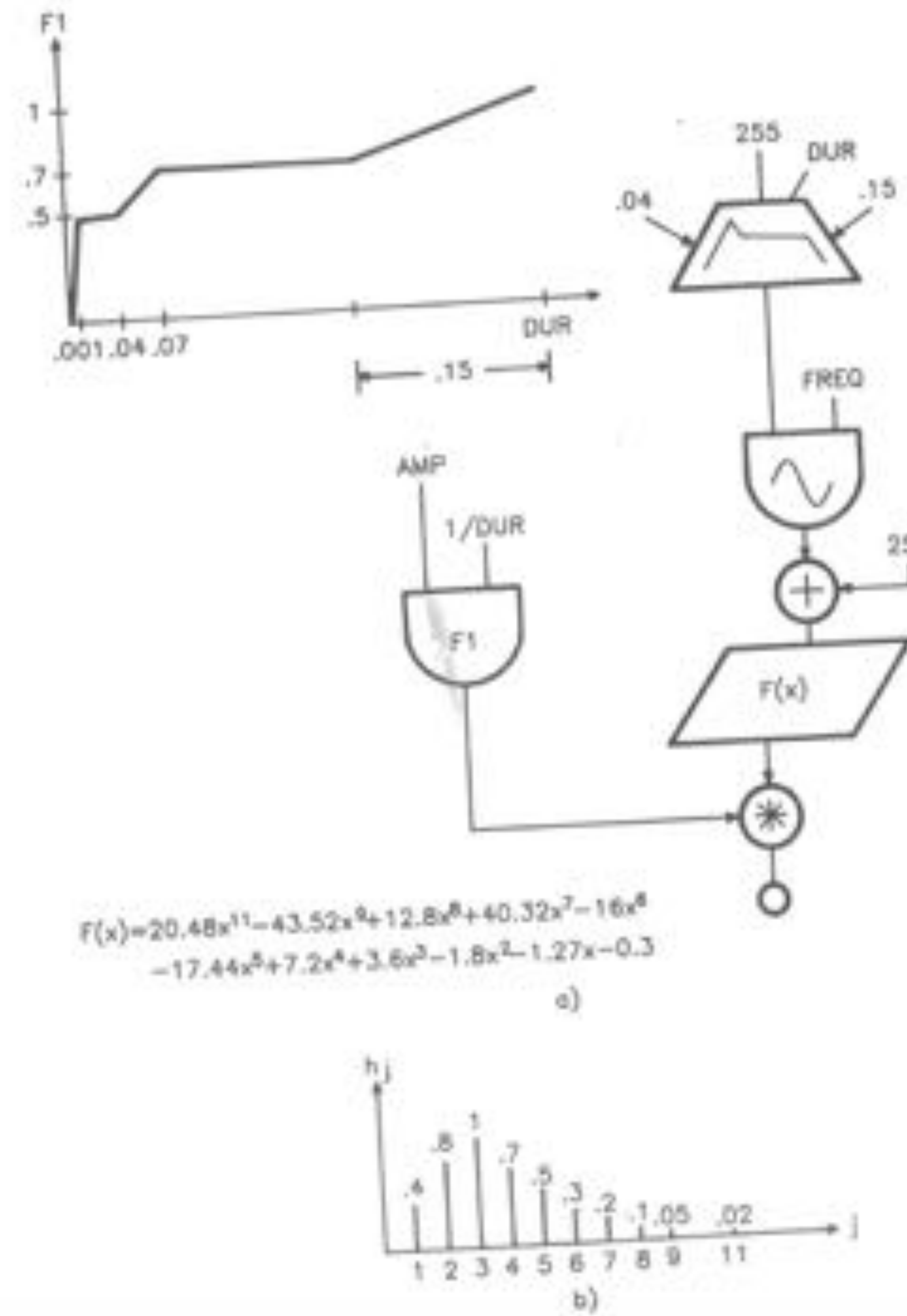


FIGURE 5.28 Waveshaping instrument that produces a clarinet-like timbre, based on Risset. (*Based on example in Risset's* Introductory Catalogue of Computer-Synthesized Sounds. *Reprinted with permission of Jean-Claude Risset.*)

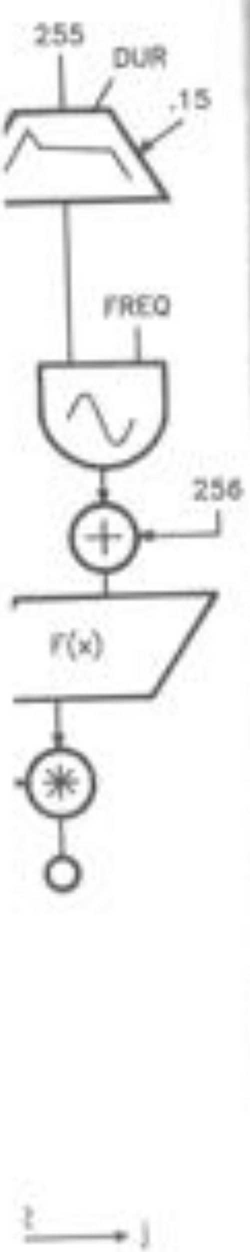
tions. The transfer function is specified graphically in a piecewise linear manner. The samples from the output of the oscillator fall in the range ±255, which is one-half the table length minus 1. To this value is added the constant value of 256. Thus, the values used to reference the waveshaper oscillate in the range from 1 to 511. The entries in the waveshaper table are scaled to the range ±1 and subsequently multiplied by the desired amplitude. As Risset points out, because the transfer function is not band-limited, this design, when used at a sampling rate of 20 kHz, will generate objectionable aliasing on tones with fundamental frequencies above about 1500 Hz.

Figure 5.29a shows the design for a nonlinear waveshaping instrument that produces simple, "brassy" tones. The design includes a scaling function (F1) to temper the variation of amplitude with spectrum. The linear rise in the distortion index causes distortion in the tone to increase rapidly during the attack. During the steady state, the distortion index has a value of 0.7, so that the harmonic content is less brilliant. During the decay, the harmonic content at the high frequencies falls off first. The transfer function was obtained by matching the spectrum shown in figure 5.29b.



$$F(x) = 20.48x^{11} - 43.52x^{9} + 12.8x^{8} + 40.32x^{7} - 16x^{6} - 17.44x^{5} + 7.2x^{4} + 3.6x^{3} - 1.8x^{2} - 1.27x - 0.3$$

FIGURE 5.20 (a) Waveshaping instrument with a "brassy" timbre and (b) spectrum that was matched.



ectrum that was matched.

5.2I Use of Ring Modulation with Waveshapers

The technique of waveshaping described above provides strictly harmonic spectra that, in general, become richer with increasing distortion index. This makes it easy to synthesize tones in which the lower harmonics attack first and decay last. However, the basic technique does not readily provide for the synthesis of certain other harmonic evolutions. A simple variation on the basic technique increases the types of sounds that can be synthesized, makes possible inharmonic spectra, and facilitates formant simulation.[19] In this approach, the output of a waveshaper is multiplied by a sinusoidal tone as shown in figure 5.30a. This results in ring modulation, which produces a spectrum containing a replicated image of the spectrum of the shaped tone both above and below the modulating frequency. The lower image is reversed (figure 5.30b). If the shaped and modulating frequency tones have fundamental frequencies of f_1 and f_2, respectively, the frequencies produced are the sum and difference of f_2 with each harmonic of f_1—that is, with an Nth-order transfer function, $f_2 \pm jf_1$, where $j = 0, 1, 2, \ldots, N$. The corresponding amplitude of each component is $0.5(A2)h_j$, where $A2$ is the amplitude of the modulating wave and h_j is the amplitude of the jth harmonic of the shaped wave.

The result bears some similarities to simple FM synthesis. If the ratio of the frequencies is given as

$$\frac{f_1}{f_2} = \frac{N_1}{N_2}$$

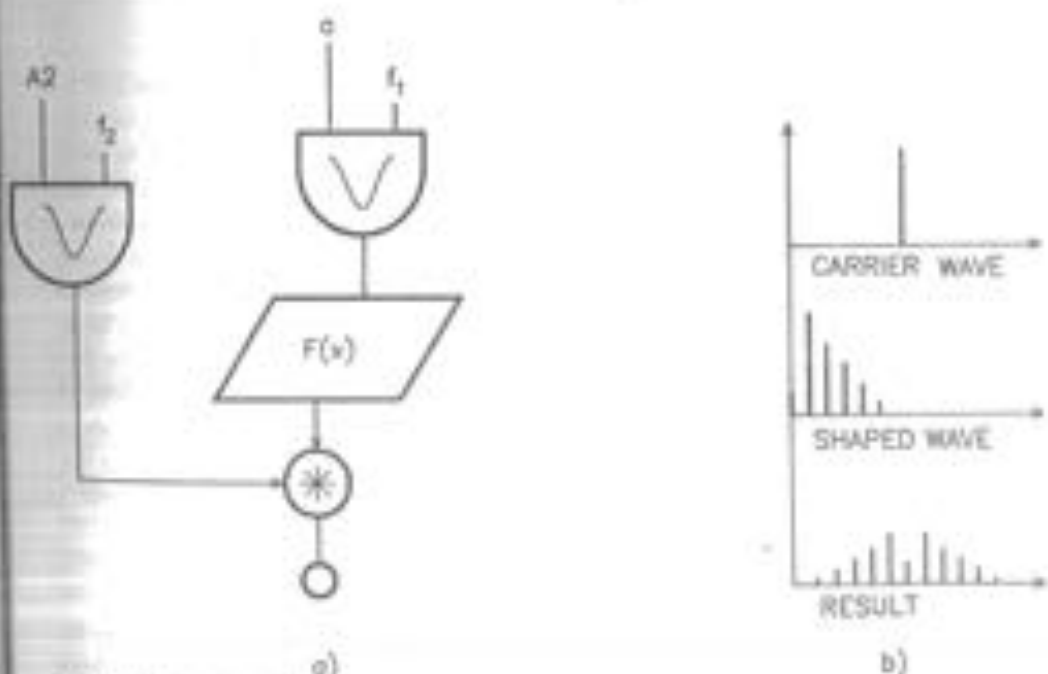


FIGURE 5.30 Use of ring modulation in a waveshaping instrument (a) and its spectrum (b).

where N_1 and N_2 are integers with no common factors, then the fundamental frequency (f_0) of the resulting waveform is

$$f_0 = \frac{f_1}{N_1} = \frac{f_2}{N_2}$$

When N_2 is even, only odd harmonics are produced. If f_1 or f_2 is an irrational number, then N_1 and N_2 cannot be defined as integers and an inharmonic spectrum results.

If f_2 is less than f_1 times the order of the transfer function (i.e., $f_2 < Nf_1$), negative frequencies result. These are transformed into positive frequencies by folding them around 0 Hz (see section 5.1B). If there is a component at the corresponding positive frequency, the folded component is added to it to form the net spectrum. The phase of the components must be considered when they are combined. Unlike FM, the folding process does not necessarily reverse the phase of a component. If the modulating waveform and the input to the waveshaper are both cosine waves, the folded components are added in phase. If the modulating wave is a sine wave, the components are out of phase and must be subtracted.

Placing an envelope on the distortion index realizes a dynamic spectrum. In addition, it is usually necessary to place an amplitude envelope on the modulating oscillator to obtain the desired overall envelope of the sound. If the transfer function has a constant term, the modulating tone is heard even when the distortion index drops to 0. This can be useful in realizing certain timbres.

5.2J A Waveshaping Instrument Using Ring Modulation

Figure 5.31 shows a waveshaper that uses the ring modulation technique of section 5.2I to produce a pitched-percussion sound. The instrument has separate envelopes for the distortion index and the amplitude. The two oscillators are tuned inharmonically to each other to produce an inharmonic spectrum. The modulating oscillator is controlled by the amplitude envelope F2, and generates a sine tone of frequency FREQ. As is characteristic of most percussive sounds, F2 has a rapid rise and a much longer, exponential decay. The oscillator driving the waveshaper is controlled by the distortion index and produces a sine tone of frequency 0.7071FREQ. During the first part of the tone, the relatively large distortion index causes the shaped wave to occupy a rather wide bandwidth. When the distortion index falls to 0, the constant term in the transfer function allows the pure tone of the modulating oscillator to come through. The duration of the tones should be kept short (< 0.25 second), to retain the characteristic percussive timbre. At lower frequencies the timbre is drumlike; at higher frequencies the sensation of pitch becomes much clearer.

5.2K Use of a High-Pass Filter

Another variation on the basic waveshaping technique involves the use of a high-pass filter in conjunction with a waveshaper. A *high-pass filter* is an element that allows high frequencies to pass through and attenuates low ones (see chapter 6). This method has

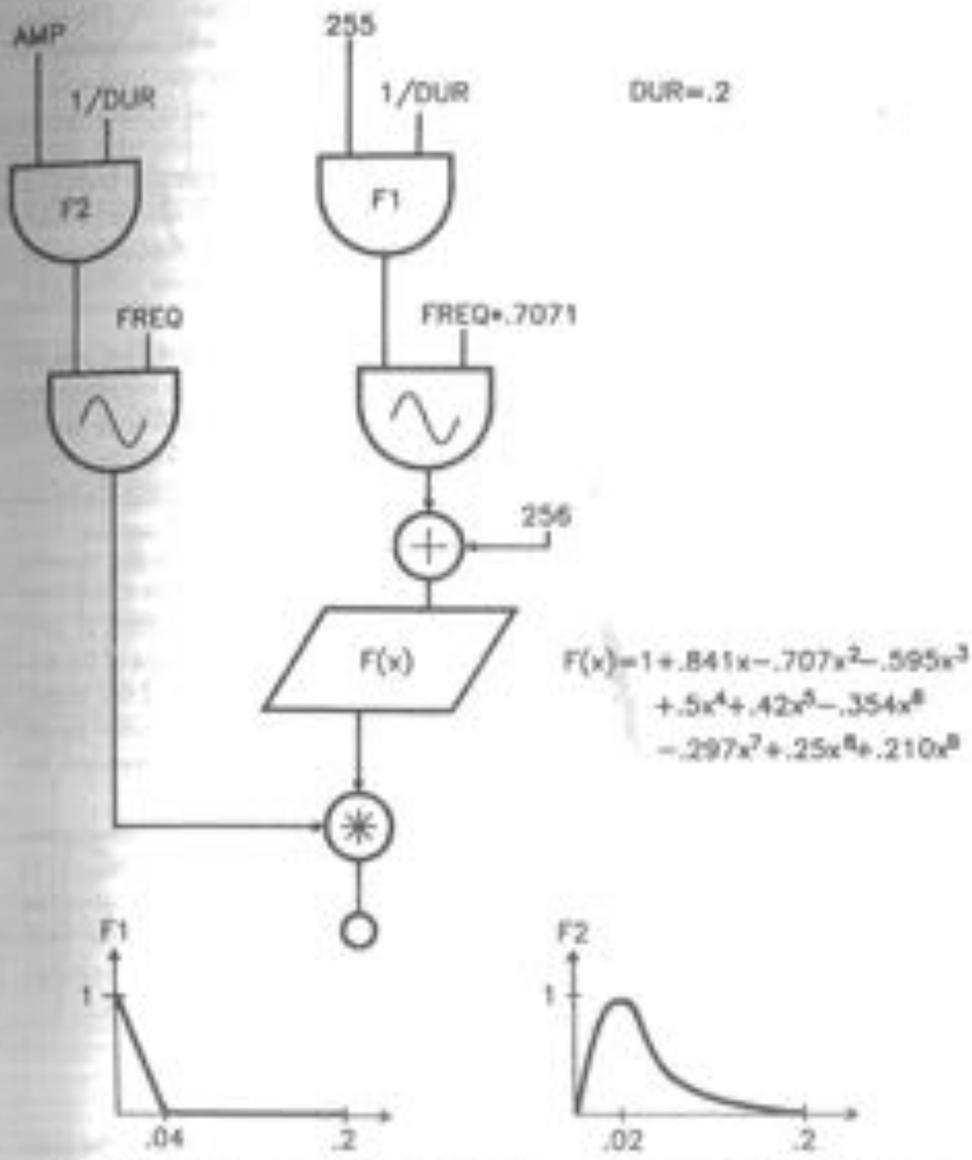


FIGURE 5.31 Waveshaping instrument that uses ring modulation to produce a drumlike sound.

been demonstrated by James Beauchamp and has been successful in the synthesis of realistic brass tones. The method is based on a mathematical model of the acoustical processes used to produce a tone on a brass instrument. The parameters of the model are determined from the analysis of actual instrument characteristics.

The spectrum of the sound produced by blowing into an unattached mouthpiece is a nonlinear function of its loudness—the harder the player blows, the richer the spectrum of the sound. The body of the natural brass instrument serves not only to resonate the sound from the mouthpiece into a clear pitch, but also acts as a high-pass filter. The computer instrument of figure 5.32 approximates the mouthpiece action with the sine-wave oscillator and the waveshaper. The transfer function is determined by matching a spectrum that is obtained by analysis of actual mouthpiece spectra at different dynamic levels.

The parameters of the high-pass filter are chosen to provide characteristics determined from acoustical measurements on the body of the brass instrument. The high-pass filter emphasizes the higher harmonics, and so the waveshaper does not need to provide as much high-frequency energy. Thus, the transfer function will give a smoother evolution with distortion index, and it would be unlikely that such an instrument will require dynamic amplitude scaling.

Waveshaping with a high-pass filter is generally a useful technique that can be adapted to the synthesis of other types of sound where the parameters are not necessarily determined from analysis. It does, however, require that the digital hardware have a large dynamic range in the numbers that it can represent, because the high-pass filter greatly reduces the peak amplitude of the waveform. The amplitude is reconstituted by the multiplication that is performed on the output of the filter.

In his use of this design to make brasslike tones, Beauchamp chose the transfer function (figure 5.32a) to simulate the nonlinear behavior of the spectrum of a cornet mouthpiece played alone. The cutoff frequency of the high-pass filter was fixed at 1500 Hz. The high-pass filter, with a constant setting of its parameters irrespective of the frequency of the instrument, contributes to the most important characteristic of this instrument design—the instrument maintains the appropriate timbre over its entire range. The particular filter algorithm that Beauchamp uses is a second-order Butterworth filter realized with the bilinear transform (see section 6.13C).

One of the primary features of this instrument design is that it produces richer spectra at higher dynamic levels. In addition, Beauchamp changes the envelope shape (F1) with the dynamic level of the desired tone. Figure 5.32b, c, and d, respectively, gives the envelope shapes for three different dynamic levels: *pp*, *mf*, and *ff*. The louder the tone, the less the amplitude of the tone diminishes during the steady state. These envelope shapes were also derived from the analysis of recordings of cornet mouthpieces.

5.3 SYNTHESIS USING DISCRETE SUMMATION FORMULAS

In addition to frequency modulation synthesis and synthesis by waveshaping, the class of techniques known as distortion synthesis includes synthesis by the explicit use of *discrete summation formulas*. This category encompasses a wide variety of algorithms. Both harmonic and inharmonic spectra in either band-limited or unlimited form can be synthesized by means of various formulas.

What is a discrete summation formula? The sum of the first N integers can be written as

$$1+2+3+\cdots+N=\sum_{k=1}^{N}k$$

where k is an arbitrary index, ranging from 1 to N. To evaluate this expression, one could