Instrument Modeling and Synthesis

Chapter · January 2009

DOI: 10.1007/978-0-387-30441-0_24

2 authors:

Andrew Horner
The Hong Kong University of Science and Technology
128 PUBLICATIONS **1,053** CITATIONS
SEE PROFILE

James Beauchamp
University of Illinois, Urbana-Champaign
119 PUBLICATIONS **983** CITATIONS
SEE PROFILE

Multiple Wavetable and Frequency Modulation Instrument Modeling and Synthesis Based on Spectral Analysis

Andrew Horner
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
horner@cs.ust.hk

James Beauchamp
School and Music and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
2136 Music Building, 1114 West Nevada Street, Urbana, Illinois 61801
jwbeauch@uiuc.edu

Abstract

Replicating musical instrument sounds is a classic problem of computer music. With the advent of digital synthesizers in the 1970s and 1980s, frequency modulation and multiple wavetable synthesis methods required trial-and-error optimization of synthesis parameters. Since the early 1990s automated techniques for optimizing the synthesis parameters have been developed. This paper surveys FM and wavetable techniques for matching synthesized to original sounds.

1 Introduction

During the 1970s and 1980s, before synthesizers based on direct sampling of musical sounds became popular, replicating musical instruments using frequency modulation (FM) or wavetable synthesis was one of the "holy grails" of music synthesis. Synthesizers such as the Yamaha DX7 allowed users great flexibility in mixing and matching sounds, but were notoriously difficult to coerce into producing sounds like those of a given instrument. Instrument design wizards practiced the mysteries of FM instrument design. Wavetable synthesis was less mysterious, but was limited to rather static organ-like sounds. With a single wavetable, you could easily attain a trumpet-like sound, but a realistic trumpet was still out of reach. Then, sampling came along and

soon even cheap synthesizers could sound like realistic pianos and trumpets, as long as the desired articulations happened to match those of the original recorded samples. Sample libraries quickly replaced sound wizards.

Ironically, at about the time that FM declined, researchers developed algorithms to optimize FM and wavetable parameters. The results were every bit as realistic as sampled sounds, with the added benefits of increased spectral and temporal control.

The basic procedure for FM and wavetable matching is shown in Fig. 1. Matching generally begins with a short-time (aka time-varying) spectrum analysis of the original sound. Typically, 500 to 2000 consecutive "spectral snapshots" or frames of the tone result from this analysis, each consisting of the instantaneous amplitudes and frequencies of the harmonics of the tone's time-varying fundamental frequency ($f_0(t)$). The next step is to find parameters for the synthesis algorithm that provide the average "best fit" of the spectrum of the tone over the set of these snapshots. The final step is to resynthesize the sound using the matched parameters.

The next section describes the short-time spectral analysis methods the authors have used, the first step of Fig. 1. Then, we describe the fundamentals of wavetable synthesis, followed by an overview of wavetable matching procedures. Next, we describe FM synthesis and FM matching. Finally, we conclude with some practical guidelines for selecting among these methods.

2 Spectral Analysis

Spectral analysis methods used for wavetable and FM matching synthesis are of necessity pitch-synchronous and were previously described by Beauchamp (1993). The most often used method is a variation on the *phase vocoder* (Flanagan 1966; Portnoff 1976). With our method, a digital single-voice signal, which is assumed to have a nearly constant fundamental frequency, is broken into overlapping segments, called windows or frames. For analysis purposes, the fundamental is chosen to be f_a, an approximation to the actual time-varying fundamental frequency $f_0(t)$. The window lengths are taken to be twice the period corresponding to the fundamental (i.e., $T = 2/f_a$), and the overlap time or time-between-frames is taken to be a half period (i.e., $\Delta T = 0.5/f_a = 0.25T$). The windowed signal segments are multiplied by a Hamming window function ($w(t) = 0.5 + 0.426\cos(2\pi t/T)$) before applying a Fast Fourier Transform (FFT) to the signal samples. However, the signal must generally be upsampled to provide an even power-of-two samples per window in order to satisfy the requirements of our version of the FFT. If the original sample rate is f_s, the new sample rate is $f_s' = 2^{\text{ceil}(\log_2(Tf_s))} / T$. After the FFT is taken, only the even components are retained, with the second component corresponding to the first harmonic, the fourth component corresponding to the second harmonic, and so forth. For each spectral snapshot or frame j, the FFT returns the real and imaginary parts of the analysis, and these must be converted to harmonic amplitudes and frequencies. For each harmonic k, the estimated amplitude $A_{k,j}$ is computed as the square root of the sum of the squares of the real and imaginary parts (i.e., the complex magnitude), whereas the estimated instantaneous frequency $f_{k,j}$ is computed using an arctangent formula based on the real and imaginary parts for two successive frames (Beauchamp 1993). The number of harmonics K analyzed and kept is given by the Nyquist frequency divided by f_a; thus, $K = \text{floor}(0.5\, f_s/\, f_a)$. The total number of data points kept is twice the number of samples in the sound signal analyzed.

An alternative analysis method is based on frequency tracking of spectral components (McAulay and Quatieri 1986; Smith and Serra 1987). This method works for signals whose fundamental frequency variation exceeds a small percentage of its mean value. A fixed window of 2^N samples length is chosen to be greater than three times the period of the lowest expected frequency. After a Kaiser-windowed FFT processes the signal, the magnitude spectrum is computed, and spectral peaks above a specified amplitude threshold are retained. Spectral peaks are connected from frame to frame using a nearest neighbor algorithm to form tracks. Next, fundamental frequency vs. time is computed using an "two-way mismatch algorithm" (Beauchamp *et al.* 1993; Maher and Beauchamp 1994). Finally, harmonics are separated from the total spectrum based on the computed time-varying f_0. For tones with little frequency variation, the result is very similar to that obtained using the phase vocoder, but the frequency-tracking method works much better for sounds with significant frequency deviation.

Although in general these methods result in harmonics which do not track each other perfectly (*i.e.*, $f_{k,j}/k_1 \neq f_{k,j}/k_2$ for $k_1 \neq k_2$), the methods described in this chapter require a common fundamental frequency such that $f_{k,j} = kf_{0,j}$. Calculation of $f_{0,j}$ is described by Beauchamp (1993). The result is a signal model given by

$$s[n] = \sum_{k=1}^{K} A_k[n]\cos(k\int f_o[n]dn / f_s + \phi_k), \qquad [1]$$

where n = sample no., and $A_k[n]$ and $f_o[n]$ are given as sample-time values interpolated from neighboring frame-time values $A_{k,j}$ and $f_{k,j}$ obtained from the spectral analysis. f_s is the sampling frequency in Hz.

3.1 Wavetable Synthesis

The music industry currently uses the term *wavetable synthesis* synonymously with *sampling synthesis*. However, in this article, *sampling* means recording an entire note and playing it back, whereas *wavetable synthesis* refers to storing only one period of a waveform in an oscillator table and synthesizing a signal by applying amplitude- and frequency-vs.-time envelopes (Mathews 1969). The waveform is generated by a sum of harmonically related sine waves, whose amplitudes and phases define the wavetable's spectrum. In our methods of wavetable synthesis, several wavetables, each corresponding to a *basis spectrum*, are scaled by time-varying amplitude envelopes (or weight functions) and summed to form the output signal. Fig. 2 shows a block diagram for this *multiple-wavetable synthesis* model.

Prior to synthesis, one cycle of each basis waveform is stored in a wavetable. The waveform is generated by a sum of harmonically-related sine waves. I.e., for the m^{th} waveform or table, the table entries are given by

$$table_m[i] = \sum_{k=1}^{K} A_{k,m} \cos\left(\frac{2\pi k i}{tableLength} + \phi_{k,m}\right), \quad \text{for } 0 \le i < tableLength \qquad [2]$$

where $table_m[i]$ is the i^{th} entry of the m^{th} wavetable, *tableLength* is the number of entries in the table, and $A_{k,m}$ and $\phi_{k,m}$ are the amplitude and phase of the k^{th} harmonic for the table. The A_k's corresponding to each table form the wavetable's basis spectrum. The phases generally don't usually affect the sound and are often set to zero, but, in any case, for a given harmonic, they should be the same for all tables in order to avoid phase cancellations. Then, for *M* equal-length wavetables, the output signal to be matched with the original of Eq. 1 is given by

$$\hat{s}[n] = \sum_{m=1}^{M} \alpha_m[n]\, table_m[\int f_o[n]dn \,/\, (f_s / tableLength)], \qquad [3]$$

where $\alpha_m[n]$ is the amplitude envelope of the m^{th} wavetable and $f_o[n]$ is the frequency envelope common to all wavetables. Although table entries are specified to be integers in Eq. 2, we assume here that the table values are interpolated according to the integer and fraction values of its argument and that its argument is taken modulo *tableLength*.

The main advantage of multiple wavetable synthesis is its efficiency for generating periodic waveforms. A possible disadvantage of this method is that each wavetable only produces a static spectrum, while real sounds produce continuously changing spectra. However, if we carefully select basis spectra and control amplitudes, we can fully reproduce the dynamic spectra of the original tone in wavetable synthesis.

Another possible disadvantage is that it is necessary that harmonics be strictly locked together in frequency so that $f_k[n] = k\, f_1[n]$. Since phase is the integral of frequency, this implies that

relative phases are also locked together. Frequency unlocking (inharmonicity) or phase unlocking certainly occurs in freely vibrating tones, such as struck or plucked string tones (Fletcher *et al.* 1962), and there is some evidence that phase unlocking (or aperiodicity) also occurs momentarily in sustained tones, especially during vibrato and transients (Beauchamp 1974; Schumacher 1992; Brown 1996; Dubnov and Rodet 2003). Extension of the multiple wavetable synthesis method for approximately handling inharmonic cases with slightly mistuned wavetables have been explored for the piano (Lee and Horner 1996; Zheng and Beauchamp 1999). Bristow-Johnson (1996) has proposed a wavetable method for changing relative phases as time progresses, which is tantamount to synthesizing unlocked phases. However, the perceptual advantage of unlocked phases in sustained tones has not been fully explored. In any case, synthesis with unlocked phases is beyond the scope of this chapter.

There are two basic types of multiple wavetable synthesis. *Wavetable interpolation synthesis* is a method whereby tables are interpolated from one (or possibly a few) to the next in succession, usually using straight line interpolation. With *wavetable index synthesis* all tables are active simultaneously, and the amplitude envelope weights applied to the tables are computed using *least squares* (Horner *et al.* 1993a). In this case, the amplitude envelopes tend to be quite complex and there is no restriction on their polarities. Thus, when basis spectra are combined (assuming they have identical phases), the weights can cause them to cancel each other in such a way that their combination converges to the original time-varying spectrum as more tables are added.

Basis spectra can be taken directly from the original time-varying spectrum at certain points in time, from spectral averages using a clustering method, or by using a method such as Principal Components Analysis (PCA) to compute basis spectra which decrease in importance as more are added. Optimal matching algorithms, such as genetic algorithms (GA) (Goldberg 1989), can be used to search for the "best" set of basis spectra.

3.2 Wavetable Matching

The objective of wavetable matching is to find the best set of parameters to synthesize a musical instrument tone using wavetable synthesis. A number of papers have explored methods for optimizing wavetable basis spectra and their amplitude-vs.-time envelopes (Horner et al. 1993a, Sandell and Martens 1995, Horner and Beauchamp 1996, Horner 2001). Usually a *cost metric* is used to determine the efficacy of the match and to steer the matching process. One metric we have found successful, called the *relative amplitude spectral error*, appears to correspond closely to perceptual similarity (Horner et al. 2006); it is given by

$$\varepsilon = \frac{\sum_{j=0}^{J-1}\sum_{k=1}^{K}\left|A_{k,j} - A'_{k,j}\right|^p}{\sum_{j=0}^{J-1}\sum_{k=1}^{K}\left|A_{k,j}\right|^p}, \qquad [4]$$

where J is the number of analysis frames used for the average. The error ε can be an average over all of the frames of the tone or a suitable subset of the frames. The power p can be either 1 or 2, and good results obtain for either case.

Maher and Beauchamp (1990) used a wavetable matching synthesis method in their investigation of vocal vibrato. They selected their basis spectra at the low and high points of the vibrato of sung tones, and crossfaded the wavetables as a function of the vibrato. Since the spectra at these points virtually repeated in cyclic fashion during the tone, only two wavetables were required for the synthesis.

With wavetable interpolation synthesis (Chamberlin 1980; Serra *et al.* 1990; Horner and Beauchamp 1996), the signal is represented by a consecutive series of basis spectra, which are generally spectral snapshots take from the tone itself at particular points in time or *breakpoints*, and synthesis proceeds by gradually crossfading from one spectrum (or waveform) to the next. While one basis spectrum ramps down to zero, a new basis spectrum ramps up to take its place.

Serra *et al.* (1990) gives two algorithms for determining spectral interpolation basis spectra. The first draws basis spectra from the original tone. The second uses a least-squares algorithm to compute the basis spectra which minimize the mean-squared error between original and synthetic spectra. Basis spectra are added until the maximum mean-squared error is brought below a prescribed threshold. Interpolation between basis spectra is done using a linear or nonlinear method. These algorithms cycle through basis spectra between 5 and 20 times per second.

Rather than first specifying a error threshold, Horner and Beauchamp (1996) used a genetic algorithm and a sequential enumeration (aka greedy) method to select a predetermined number of best times (breakpoints) to take basis spectra from the signal. Several questions were explored. It was found that the GA method was slightly better than the greedy method in terms of relative

error vs. number of breakpoints, but it was substantially better than random breakpoints or equally spaced breakpoints. While breakpoints optimized independently for each harmonic performed somewhat better than common breakpoints in terms of error vs. the number of breakpoints per harmonic, common breakpoints won in terms of error vs. data storage requirements. Also, independent harmonic breakpoints would not allow wavetable interpolation. Another method tried used quadratic rather than linear curves to interpolate between basis spectra. Like the independent harmonic breakpoint attempt, this resulted in an improvement for the same number of breakpoints, but when total data and computation requirements were taken into account, improvement was negligible or nill.

Genetic algorithms have also been used to optimize wavetable interpolation with multiple (more than two at a time) wavetables (Horner 1996b). In this case, each breakpoint spectrum is the weighted sum of two or more spectra (wavetables) which are crossfaded to the same number of tables at the next breakpoint.

Group additive synthesis (Kleczkowski 1989) is another wavetable variant at the opposite extreme of spectral interpolation. Group additive synthesis divides the spectrum into non-overlapping subsets of harmonics for the various wavetables. As an example, one wavetable might contain only the even harmonics while a second only the odd. Subsequent to Kleczkowski's initial study, researchers have optimized group additive synthesis parameters using an automated clustering scheme (Oates and Eaglestone 1997) and genetic algorithms (Cheung and Horner 1996; Horner and Ayers 1998; Lee and Horner 1999).

For wavetable index synthesis, researchers have used genetic algorithms to match the multiple wavetable model shown in Fig. 2 (Horner *et al.* 1993a; Horner 1995). Like the interpolation synthesis method, one approach is to use a genetic algorithm to select the best time values (breakpoints) for spectral snapshots to be taken from the original tone as the basis spectra (Horner *et al.* 1993a). However, unlike the interpolation method, a least-squares algorithm is used to compute the time-varying basis spectra amplitude weights which minimize the squared error between the original and synthetic spectra. This method generates an exact match at the time points of the selected snapshots and usually results in excellent matches at neighboring points as well. The average relative spectral error between the original and matched spectra typically serves as a fitness function to guide the GA's search for the best solution. Most matched instruments require 3 to 5 wavetables for a good match (less than 5% average relative spectral error), a considerable savings compared to sinusoidal additive synthesis or interpolation synthesis. However, in terms of total computation, interpolation synthesis can be just as efficient as wavetable index synthesis and is more intuitive in terms of the amplitude weight envelopes. This is because the amplitude weights for wavetable index synthesis can be either positive or negative and thus the sum of the amplitude-controlled basis spectra only *converge* to the least-error values.

Another approach for selecting basis spectra is based on Principal Components Analysis (PCA). PCA is a statistical procedure for optimizing both weights and basis vectors (Dunteman 1989) and has been applied to the wavetable matching problem (Zahorian and Rothenberg 1981; Sandell and Martens 1992; Horner *et al.* 1992; Horner *et al.* 1993a; Sandell and Martens 1995). In our case, PCA first finds a basis spectrum and time-varying amplitude control to minimize the

average mean-squared spectral error (rather than the relative spectral error). It then finds an amplitude and control to further minimize the error. It usually converges to a good approximation with five basis spectra (wavetables). If the number of PCA basis spectra equals the number of harmonics, convergence is perfect. Also, an alternative way to compute the control functions, which we actually use, is to first compute the basis spectra and then employ the same method we used for wavetable index synthesis, which is the least-squares solution. However, there are a couple of disadvantages to to the PCA method in comparison to using spectral snapshots: First, in selecting basis spectra, PCA strongly weights the high-amplitude steady-state spectra, which is unfortunate because the low-amplitude attack and release spectra are perceptually very important. However, this can be overcome by judicious use of spectra used in computing the basis spectra. Second, except for the first basis spectrum, the PCA basis spectra do not resemble real spectra because they are used to correct the first one (in analogous fashion to Fourier sine waves correcting each other to form a square wave) and are therefore less intuitive to work with than spectral snapshots. Nevertheless, PCA provides an interesting alternative approach to basis spectra generation.

Another method which uses basis spectra/wavetables is *cluster synthesis*. With this method snapshot spectra are sorted into clusters according to some measure of the spectra. One method is to sort the spectra according to their spectral centroids (Beauchamp and Horner 1995, Horner and Beauchamp 1995). The spectra within each cluster are amplitude-normalized and then averaged to produce a basis spectrum. For synthesis the control functions are amplitude, centroid, and f_0 vs. time. For each time frame, the centroid value is used to interpolate between two basis spectra whose centroids straddle this value to get an output spectrum which is then

synthesized at the appropriate amplitude and f_0. Like wavetable interpolation, it can be shown that interpolation in the frequency domain coupled with additive synthesis is equivalent to interpolation in the time domain with basis-spectra-loaded wavetables.

An alternative method of cluster synthesis comes from the technique of vector quantization (Ehmann and Beauchamp 2002). Here the spectra are clustered using a K-Means algorithm (Rabiner and Juang 1993). The basic synthesis method is similar to the centroid-clustering method above. However, with vector quantization, each cluster is represented by an average spectrum which is labeled by a "code-book number" or index rather than by its average spectral centroid. So the basis spectra must be retrieved by index-vs.-time data. For synthesis, the basis spectra can be ordered according to their spectral centroids or corresponding amplitudes in the original signal. Then for each synthesis frame, the best rms basis spectrum match is found, and the index-vs.-time data are generated. To avoid discontinuities, the index-vs.-time data are smoothed, and then interpolation is used to insure smooth transitions between the basis spectra. So the control parameters are time-varying amplitude, index, and f_0. Alternatively, the time-varying spectrum is first normalized in terms of both amplitude and spectral tilt, where tilt is the slope p of a straight line that forms the best least-squares fit to the spectrum plotted as log-amplitude-vs.-log-frequency. The K-Means algorithm is then used to cluster the resulting flattened spectra. Synthesis parameters now consist of amplitude, tilt, index, and f_0 vs. time, and synthesis proceeds in a fashion described above, except that now both amplitude and spectral tilt have to be imposed on the retrieved spectra before synthesis. Therefore, synthesis has to be done in the frequency domain before conversion to signal. However, if tilt normalization is omitted, wavetables can be used as in the centroid-clustering case.

Wavetable synthesis is an inherently harmonic synthesis method, so replicating sounds that are nearly harmonic, such as piano and plucked string tones, requires some enhancements of the method. By grouping partials into groups with similar frequency stretch factors (Fletcher *et al.* 1962), genetic algorithms have successfully optimized group additive synthesis parameters to simulate piano tones (Lee and Horner 1999; Zheng and Beauchamp 1999) and string tones (So and Horner 2002).

Sometimes using a complex method can lead to a simpler one. We have found that insights gained from exploring a problem first with the GA method often led to finding better or simpler solutions. Wavetable matching is such an example. Instead of using the GA to approximate the best match to all or a subset of the spectral snapshots of the original tone, an alternative method is to use a combinatorial method to find the best match for a subset of the spectral snapshots (Horner 2001). It turns about this approach is as effective and efficient as the GA method, and much simpler.

4.1 Frequency Modulation (FM) Synthesis Models

Like wavetable synthesis, FM synthesis can efficiently generate interesting sounds. There are several possible FM configurations, including those with multiple parallel modulators, nested (serial) modulators, and feedback (see Fig. 3). Several of these "modules" can be combined to form complex FM synthesizers. For example, as with multiple-wavetable synthesis, several carrier oscillators can be combined in parallel to form a single-modulator/multiple-carrier FM synthesizer (see Fig. 4). During the height of FM's popularity in the 1980s, synthesizers such as

the Yamaha DX7 allowed users great flexibility in mixing and matching models like these.

4.1.1 **Single Modulator/Single Carrier Model**

The original FM equation Chowning used for music synthesis consisted of a single sine wave modulating a carrier sine wave in a vibrato-like fashion (Chowning 1973):

$$A\sin(2\pi\int(f_c + \Delta f_m \cos(2\pi f_m t)dt = A\sin(2\pi f_c t + \alpha_m \sin(2\pi f_m t)), \qquad [5]$$

where A = amplitude, f_c = carrier frequency, Δf_m = modulator amplitude, f_m = modulation frequency, and $\alpha_m = \Delta f_m / f_m$ = modulation index. Vibrato results when the modulator frequency is low (< 20 Hz). However, with an audio-rate modulator frequency, a spectrum is heard whose frequencies depend on the carrier and modulator frequencies and whose amplitudes depend on the modulation index. This is made obvious by expanding Eq. 5 in terms of Bessel functions:

$$\sin(2\pi f_c t + \alpha_m \sin(2\pi f_m t)) = \sum_{k=-\infty}^{\infty} \mathrm{J}_k(\alpha_m)\sin(2\pi(f_c + kf_m)t). \qquad [6]$$

We see that the spectrum frequencies are given by $f_k = |f_c + k f_m|$, for k = …-3,-2,-1,0,1,2,3…, which makes it clear that the f_c is the center frequency with amplitude $\mathrm{J}_0(\alpha_m)$ and that this component is surrounded by positive and negative side bands of $f_c \pm |k| f_m$ with amplitudes $\mathrm{J}_k(\alpha_m)$. As k increases eventually a point is reached (when $k > f_c/f_m$) where the negative-k side-band frequencies become negative, causing the sine function to flip its sign. Negative frequencies beyond this point are said to "fold over zero". Thus, the frequencies are effectively just $|f_c \pm |\mathrm{k}| f_m|$ with amplitudes of $\pm|\mathrm{J}_{|k|}(\alpha_m)|$. Because, for the same value of $|k|$, the negative frequency component amplitudes are given by $\mathrm{J}_{-k}(\alpha) = (-1)^k \mathrm{J}_k(\alpha)$, Eq. 6 can be rewritten as

$$\begin{aligned}\sin(2\pi f_c t + \alpha_m \sin(2\pi f_m t)) &= J_0(\alpha_m)\sin(2\pi f_c t) \\ &+ \sum_{k=1}^{\infty} J_k(\alpha_m)[\sin(2\pi(f_c + kf_m)t) + (-1)^k \sin(2\pi(f_c - kf_m)t)]\end{aligned}, \qquad [7]$$

which shows a clear separation between the positive and negative sidebands. The actual sign of each side-band amplitude depends on a combination of a) whether it's a positive or negative side-band, b) whether the Bessel function itself is positive or negative at a particular value of α_m, and c) whether the component has folded-over (*i.e.*, $k > f_c/f_m$). Also, note that the left sides of Eqs. 6 and 7 are actually in the form of phase modulation (PM). FM and PM are closely related because the phase is the integral of the frequency, i.e., $\alpha_m \sin(2\pi f_m t) = \int 2\pi \Delta f_m \cos(2\pi f_m t) dt$. If some other sinusoid phase inside the integral is used (e.g., $\sin(2\pi f_m t)$), the result will be different, although basically the same type of spectra will result. For details see Beauchamp (1992).

The modulation index controls the amount of modulation and the precise shape of the resulting spectrum. Keeping f_m fixed, the spectrum bandwidth generally increases as the modulation index α_m increases. This effect shown in Fig. 5. For large α_m, the -40 dB bandwidth approaches $2\Delta f_m$. For small α_m, the bandwidth-to-Δf_m ratio gets larger; for example, at $\alpha_m = 1$, the ratio is approximately 6.2. Thus, a time-varying modulation index produces a dynamically changing spectrum from a single FM carrier-modulator oscillator pair. By contrast, single wavetable synthesis lacks this flexibility. Still, this simple FM model has the unfortutunate property that, due to the oscillatory nature of the Bessel functions, as the modulation index changes, individual spectral components fade in and out dramatically, in a way that is not characteristic of typical musical tone spectral evolution.

A special case of Eq. 7 called *formant FM* occurs when the carrier frequency is an integer multiple of the modulator frequency. First, the spectrum is harmonic with fundamental frequency f_m, and second, for limited values of α_m, the spectrum tends to form a symmetrical formant band

around the carrier frequency (see Fig. 5).

4.1.2 **Single Modulator/Multiple Carrier Synthesis Model**

Like wavetable synthesis, FM synthesis is very efficient in terms of computation and storage: A single carrier-modulator FM instrument requires about the same amount of computation as a pair of wavetables. However, only a single sine wavetable is required for all the modulators and carriers of a complex FM synthesizer; thus, FM is more storage-efficient than wavetable synthesis. Also, assuming that modulation indices are not time-varying, each carrier's output has a static spectrum, and so the FM model provides time-varying spectrum control in similar fashion as multiple-wavetable synthesis (see Fig. 2), i.e., by means of the time-varying amplitudes of the carriers (see Fig. 4). However, unlike wavetable synthesis, the spectrum produced by FM is not arbitrary, but is restricted to the subset of possible FM spectra, so that more modules are generally needed for FM to produce a result of the same quality.

An equation for single-modulator/multiple-carrier FM synthesis, as depicted in Fig. 4, is given by

$$s(t) = \sum_{n=1}^{N} A_n(t) \sin(2\pi(r_n f_m t + \alpha_{m_n} \sin(2\pi f_m t))). \qquad [8]$$

For each carrier oscillator n, the time-varying amplitude is given by $A_n(t)$, the modulator frequency by f_m, the carrier frequency by $r_n f_m$ (where r_n is an integer), and the modulation index by α_{mn}. The time-varying harmonic spectrum of $s(t)$ can be calculated by expanding each term of Eq. 8 using Eq. 7 and combining corresponding frequencies.

4.1.3 **More Complex FM Models**

A few years after Chowning's original work, Schottstaedt (1977) introduced a double

modulator/single carrier (DMSC) FM model (see Fig. 3), where the outputs of two oscillators are summed to modulate the carrier. Thus,

$$s(t) = A\sin(2\pi f_c t + \alpha_{m1}\sin(2\pi f_{m1} t) + \alpha_{m2}\sin(2\pi f_{m2} t)). \quad [9]$$

If the carrier and modulator frequencies are all related by integer multiples of the fundamental, a harmonic tone results. As shown by Le Brun (1977), the DMSC FM model with N modulators and a single carrier produces frequencies of $|f_c + k_1 f_{m1} + \ldots k_N f_{mN}|$ with Bessel function product amplitudes $J_{k1}(\alpha_{m1})\, J_{k2}(\alpha_{m2})\ldots J_{kN}(\alpha_{mN})$ for all combinations of (positive negative or zero) integers $k_1, \ldots, k_N$. This is a significantly more complicated model, even with just two modulators, than the single-modulator case, as described above, thus making parameter optimization more difficult than for the single modulator case.

Justice (1979) introduced a two modulator nested FM model (see Fig. 3), where one modulator modulates another, which, in turn, modulates the carrier:

$$s(t) = \sin(2\pi f_c t + \alpha_{m_1}\sin(2\pi f_{m_1} t + \alpha_{m_2}\sin(2\pi f_{m_2} t))). \quad [10]$$

Like double FM, if the carrier and modulator frequencies are all related by integer multiples of the fundamental, a harmonic tone results. By treating the modulator m_1 as a carrier, we see it can be expanded in terms of Bessel functions as in Eqs. 6 & 7 so that Eq. 10 could be expanded in the form of Eq. 9 with an infinite sum of modulators. Taking this a step further, as for the finite modulator sum case, results in an equation with infinite sums and products. This is a much more complicated relationship than the Bessel function expansion of double FM, where the modulators are summed instead of nested. However, if time samples of Eq. 10 are computed uniformly over the fundamental period (the inverse of the largest common divisor of f_c , f_{m1} , and f_{m2}), a discrete Fourier transform can be used to compute the harmonic amplitudes for any combination of the

five parameters, f_c, f_{m1}, f_{m2}, α_{m1}, and α_{m2}.

Another FM variant that proved useful in 1980s synthesizers was *feedback FM* (Tomisawa 1981; Mitsuhashi 1982). The output of the carrier at sample *n* modulates the following sample, scaled by a modulation index:

$$s(n+1) = \sin(2\pi f_c n / f_s + \alpha_m s(n)). \qquad [11]$$

Unlike the other FM methods discussed above, this is probably intractable for analytic solution. However, given a starting value, e.g., $s(0)=0$, it can be computed and converted into a spectrum. It turns out that when the modulation index is less than about 1.5, a monotonically decreasing spectrum results (Tomisawa 1981). Because of this, feedback FM is potentially more easily controlled than the other forms of FM, where the harmonics oscillate in amplitude as the modulation index changes. Another advantage of feedback FM over other FM models is that its harmonic amplitudes are strictly positive when the modulation index is less than 1.5 (other forms of FM produce both positive and negative amplitudes). This avoids cancellation when adding multiple carriers together. Still, the monotonically decreasing spectrum of feedback FM has a disadvantage: Many musical instruments which have strong formants at upper harmonics cannot be directly modelled by the monotonic spectra inherent with Feedback FM. Nevertheless, amplitude modulation of the feedback-FM carrier signal by a sine wave can shift frequencies upward to overcome this limitation.

4.2 Frequency Modulation Matching

One of the factors leading to FM's decline in popularity in the late 1980s was that matching an arbitrary musical instrument tone by FM synthesis is difficult, much more difficult than with

wavetable matching. A closed-form analytical solution for determining the best set of FM parameters does not exist, and some form of general-purpose optimization is necessary. Most previous work on FM utilized ad hoc and semi-automated techniques for matching instrument tones. However, hand tuning of multiple carriers quickly exceeds the limits of human ability and endurance. In the early 1990s Horner et al. (1993b) introduced systematic techniques for matching tones to FM models based on genetic algorithm (GA) optimization.

Several researchers have emulated particular instruments by tailoring the time-varying FM indices by hand, starting with Chowning's original FM paper (Chowning 1973). In addition to applying FM to music synthesis, Chowning gave parameters appropriate to various classes of instruments based on simulating various properties of those instruments. For instance, a brass tone's spectral centroid (a strong correlate of perceptual brightness) is usually proportional to its overall amplitude. Chowning simulated this behavior by taking advantage of the fact that the centroid of an FM spectrum generally increases as its modulation index increases. He then varied the modulation index in direct proportion to the amplitude of the carrier to approximate a brass instrument. He produced woodwind-like tones and percussive sounds using similar methods. Chowning also discussed a double carrier instrument near the end of his 1973 paper. In a later study, Chowning (1980) designed a double carrier FM instrument to simulate a singing soprano voice. Chowning centered the first carrier at the fundamental frequency and the second at an upper formant, intuitively deriving the parameters of the instrument. He identified vibrato as critically important for achieving a voice-like sound.

Morrill (1977) followed Chowning's lead in trying to determine FM parameters based on detailed

knowledge of trumpet sounds. Morrill outlined single and double carrier instrument designs for the trumpet and clearly identified the limitations of single carrier instruments. His double carrier instrument set carrier frequencies to the fundamental and its sixth harmonic, the latter corresponding to a known upper formant region in the trumpet. Morrill also pointed out the difficulty in predicting the spectral output of the double carrier instrument.

Schottstaedt (1977) changed the basic FM instrument design by using two modulators to simultaneously frequency-modulate a single carrier. After describing the spectral behavior of the double-modulator FM model, Schottstaedt gave parameters for simulating the piano and string instruments. He used instrument characteristics and trial-and-error to find the parameters. The technique found small modulation indices most useful.

Justice (1979) introduced the nested modulator FM model, and outlined a Hilbert transform procedure to decompose a signal into parameters for a single carrier FM instrument. The procedure attempted to produce a matched FM signal close to the original, leaving the user to tweak the parameters as desired. However, Justice matched FM-generated signals and not those of acoustic musical instruments. Payne (1987) extended Justice's technique to a pair of carriers with nested modulators. Each carrier contributed to an independent frequency region, giving a more accurate match than with a single carrier. In addition to matching contrived FM signals, Payne matched a cello sound. The result was reportedly string-like, but lacking properties of "liveliness." Payne reported that the matching procedure was computationally very expensive.

More recent work by Delprat and his collaborators used a wavelet analysis and a Gabor

transform to find spectral trajectories (Delprat et al. 1990; Delprat 1997). They used these trajectories to estimate modulation parameters. This approach was similar to that used by Justice and Payne except that it broke the frequency range into more component parts. Thus, it was also computationally expensive. They gave examples for a saxophone and trumpet using five carrier-modulator pairs, indicating the growing awareness that precise spectral control requires several carriers.

Meanwhile, Beauchamp (1982) developed a frequency-domain method to find FM parameters as part of a larger study on nonlinear synthesis based on time-varying spectral centroid (aka "brightness") matching. He used a single-modulator/single-carrier model with a centroid-controlled modulation index to match the time-varying spectral centroid of the original signal. Though the level of control was too coarse to provide a convincing synthesis, the technique was notable in its attempt to perform an automated spectral match.

In the early 1990s researchers introduced evolutionary matching techniques for the various FM models, first applying them to the single-modulator/multiple-carrier (formant FM) model (Horner et al. 1993b). A genetic algorithm (GA) procedure was used to optimize fixed modulation indices and modulator-to-carrier frequency ratios for various numbers of carriers in an effort to best match particular instrument sounds. Fixed rather than time-varying modulation indices were used because allowing them to vary resulted in radically different frame-to-frame indices with their accompanying audible artifacts. Using invariant indices also avoided the considerable extra expense of optimizing time-varying indices. As in wavetable matching, the relative spectral error between the original and matched spectra served as a fitness function in

guiding the GA's search for the best FM parameters. Matching results in terms of relative amplitude spectral error (see Eq. 4) vary with instrument. Recent measurements on a trumpet tone showed that 5 carriers achieved an error of 10%, whereas for a Chinese pipa tone an error of 25% was achieved for 6 carriers (Horner 2006). This performance is substantially worse than for the wavetable index method where 3 tables achieved an error of 7.5% for the trumpet and 6 tables achieved an error of 10% for the pipa.

Tan *et al.* (1994) introduced an enumerative procedure for optimizing a steady-state double modulator FM model. Because this model only produced static spectra, it did not effectively match instruments with dynamic spectra. Since then, genetic algorithms have successfully optimized the double FM problem (Horner 1996a; Tan and Lim 1996; Lim and Tan 1999). GA methods were used to optimize invariant modulation indices in the double FM model, and the modulation indices it found were relatively small. Double FM matches were worse than formant FM matches when compared against the same number of table lookups. However, double FM matches were better than formant FM matches for the same number of carriers, an advantage when double FM hardware is available.

Finally, the GA method was applied to nested modulator and feedback FM matching (Horner 1998). Fixed modulation indices were used for nested modulator FM, but time-varying modulation indices were allowed for feedback FM. Like double FM matching, the optimized parameters for nested modulator and feedback FM had relatively small modulation indices. Feedback FM often gave the best matches of all the FM models when compared against the same number of table lookups, indicating feedback FM is a good choice for software synthesis where

computation is the main factor. However, if nested modulator FM hardware is already available, then double or triple modulator FM gave the best results of all the FM models for the same number of carriers.

5 Conclusions

We have reviewed several techniques for instrument modeling/synthesis based on spectral analysis, with a particular focus on wavetable and FM matching. Among the various types of wavetable synthesis, the best method depends on the given situation. For simplicity, group additive synthesis has the advantage of being intuitive, since each harmonic is only in one wavetable. For memory-constrained systems where instruments have to compete for limited wavetable space, wavetable matching is a very good choice. Conversely, for real-time systems where memory is not a problem, wavetable interpolation is a good choice.

Alhough wavetable matching is simpler and more effective than FM matching in general (Horner 1997), FM synthesis provides real-time flexibility over wavetable synthesis when wavetable memory is limited. Among the various types of FM, the best method again depends on the given situation. For simplicity and ease of control, formant FM is best. For software synthesis where computation is the main factor, feedback FM is best. If FM hardware is available, nested modulator FM is best.

In any case, the various wavetable and FM matching procedures provide an interesting point of departure for instrument designers in applications such as *timbral interpolation* (Grey 1975; Beauchamp and Horner 1998), where the parameters of one spectral match are crossfaded to that

of another. The smoothness of the transformation depends on the synthesis technique. Wavetable synthesis gives a smoother interpolation than FM synthesis. Interpolating distantly spaced FM index values will likely produce wildly changing spectral results during the interpolation due to oscillation of the Bessel functions. However, such interpolations may still be interesting and musically useful.

Acknowledgements

The Hong Kong Research Grant Council's Projects 613505 and 613806 supported this work. James Beauchamp's sound analysis and display program *Sndan* and a variation of his listening test program *SameDiff* were used in this work.

References

Beauchamp, J. (1982). "Synthesis by Amplitude and 'Brightness' Matching of Analyzed Musical Instrument Tones", *J. Audio Eng. Soc.* **30**(6) pp. 396-406.

Beauchamp, J. (1992). "Will the Real FM Equation Please Stand Up"(letter), *Computer Music J.* **16** (4), pp. 6-7.

Beauchamp, J. (1993). "Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds", Audio Eng. Soc. Preprint No. 3479.

Beauchamp, J., & A. Horner (1998). "Spectral Modeling and Timbre Hybridization Programs for Computer Music", *Organised Sound* 2(3), pp. 253-258.

Chamberlin, H. (1980). "Advanced Real-Timbre Music Synthesis Techniques", *Byte Magazine* April, 1980, pp. 70-94, 180-196.

Cheung, Ngai-Man & A. Horner (1996). "Group Synthesis with Genetic Algorithms", *J. Audio Eng. Soc.* **44**(3), pp. 130-147.

Chowning, J. (1973). "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation", *J. Audio Eng. Soc.* **21**(7), pp. 526-534.

Chowning, J. (1980). "Computer Synthesis of the Singing Voice", *Sound Generation in Wind, Strings, Computers* (Stockholm: The Royal Swedish Academy of Music), pp. 4-13.

Delprat, N., P. Guillemain, & R. Kronland-Martinet (1990). "Parameter Estimation for Non-Linear Resynthesis Methods with the Help of a Time-Frequency Analysis of Natural Sounds", Proc. 1990 Int. Computer Music Conf. Glasgow, Scotland, pp. 88-90.

Delprat, N. (1997). "Global Frequency Modulation Law Extraction from the Gabor Transform of a Signal: A First Study of the Interacting Components Case", *IEEE Trans. Speech and Audio Processing* **5**(1), pp. 64-71.

Dunteman, G. (1989), *Principal Components Analysis*, Newbury Park, CA: Sage Publications.

Ehmann, A. F. and Beauchamp, J. W. (2002) "Musical sound analysis/synthesis using vector-quantized time-varying spectra" (abstract), *J. Acoust. Soc. Am.* **112** (5, pt. 2), p. 2239.

Flanagan, J. L. and Golden, R. M. (1966). "Phase Vocoder", *Bell System Technical J.* **45**, pp. 1493-1509.

Fletcher, H., Blackham, E. D., Stratton, R. (1962). "Quality of Piano Tones", *J. Acoust. Soc. Am.* **34**(6), pp. 749-761.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley.

Grey. J. (1975). "An Exploration of Musical Timbre", Ph.D. Dissertation (Stanford: Dept. of Music, Stanford Univ.).

Horner, A., Beauchamp, J., and Haken, L. (1992). "Wavetable and FM Matching Synthesis of Musical Instrument Tones," *Proc. 1992 Int. Computer Music Conf.*, San Jose, CA, pp. 18-21.

Horner, A., Beauchamp, J. & Haken, L. (1993a). "Methods for Multiple Wavetable Synthesis of Musical Instrument Tones", *J. Audio Eng. Soc.* **41**(5), pp. 336-356.

Horner, A., Beauchamp, J. & Haken, L. (1993b). "Machine Tongues XVI: Genetic Algorithms and Their Application to FM Matching Synthesis", *Computer Music J.* **17**(4), pp. 17-29.

Horner, A. (1995). "Wavetable Matching Synthesis of Dynamic Instruments with Genetic Algorithms", *J. Audio Eng. Soc.* **43**(11), pp. 916-931.

Horner, A. (1996a). "Double Modulator FM Matching of Instrument Tones", *Computer Music J.* **20**(2), pp. 57-71.

Horner, A. (1996b). "Computation and Memory Tradeoffs with Multiple Wavetable Interpolation", *J. Audio Eng. Soc.* **44**(6), pp. 481-496.

Horner, A. & Beauchamp, J. (1996). "Piecewise Linear Approximation of Additive Synthesis Envelopes: A Comparison of Various Methods", *Computer Music J.* **20**(2), pp.72-95.

Horner, A. (1997). "A Comparison of Wavetable and FM Parameter Spaces", *Computer Music J.* **21**(4), pp. 55-85.

Horner, A. (1998). "Nested Modulator and Feedback FM Matching of Instrument Tones", *IEEE Trans. Speech and Audio Processing* **6**(4), pp. 398-409.

Horner, A. & Ayers, L. (1998). "Modeling Acoustic Wind Instruments with Contiguous Group Synthesis", *J. Audio Eng. Soc.* **46**(10), pp. 868-879.

Horner, A. (2001). "A Simplified Wavetable Matching Method Using Combinatorial Basis Spectra Selection", *J. Audio Eng. Soc.* **49**(11), pp. 1060-1066.

Horner, A. (2007). "A Comparison of Wavetable and FM Data Reduction Methods for Resynthesis of Musical Sounds", in *Analysis, Synthesis, and Perception of Musical Sounds*, J. W. Beauchamp, ed., Springer, pp. 228-249.

Justice, J. (1979). "Analytic Signal Processing in Music Computation", *IEEE Trans. Acoustics, Speech, and Signal Processing* 27(6), pp. 670-684.

Kleczkowski, P. (1989). "Group Additive Synthesis", *Computer Music J.* **13**(1), pp. 12-20.

Le Brun, M. (1977). "A Derivation of the Spectrum of FM with a Complex Modulating Wave", *Computer Music J.* 1(4), pp. 51-52.

Lee, K., & Horner, A. (1999). "Modeling Piano Tones with Group Synthesis", *J. Audio Eng. Soc.* **47**(3), pp. 101-111.

Lim, S. M., & Tan, B. T. G. (1999). "Performance of the Genetic Annealing Algorithm in DFM Synthesis of Dynamic Musical Sound Samples", *J. Audio Eng. Soc.* **47**(5), pp. 339-354.

Maher, R., & Beauchamp, J. (1990). "An Investigation of Vocal Vibrato for Synthesis", *Applied Acoustics* **30,** pp. 219-245.

Mathews, M. V. (1969). *The Technology of Computer Music*, M. I. T. Press, p. 56.

McAulay, R. J. and Quatieri, T. F. (1986). "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. Acoustics, Speech, and Signal Processing* **34**(4), pp. 744-754.

Mitsuhashi, Y. (1982). "Musical Sound Synthesis by Forward Differences", *J. Audio Eng. Soc.* **30**(1/2), pp. 2-9.

Morrill, D. (1977). "Trumpet Algorithms for Computer Composition", *Computer Music J.* **1**(1), pp. 46-52.

Oates, S., & Eaglestone, B. (1997). "Analytic Methods for Group Additive Synthesis", *Computer Music J.* **21**(2), pp. 21-39.

Payne, R. (1987). "A Microcomputer Based Analysis/Resynthesis Scheme for Processing Sampled Sounds using FM", Proc. 1987 Int. Computer Music Conf., Urbana, IL, pp. 282-289.

Portnoff, M. R. (1976). "Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform", IEEE Trans. Acoustics, Speech, and Signal Processing **ASSP-24**(3), pp. 243-248.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*, Prentice Hall, pp. 122-132.

Sandell, G. and Martens, W. (1992). "Prototyping and Interpolation of Multiple Musical Timbres Using Principal Component-Bases Synthesis," *Proc. 1992 Int. Computer Music Conf.*, San Jose, CA, pp. 34-37.

Sandell, G. and Martens, W. (1995). "Perceptual Evaluation of Principal-Component-Based Synthesis of Musical Timbres," *J. Audio Eng. Soc.* **43**(12), pp. 1013-1028.

Schottstaedt, B. (1977). "The Simulation of Natural Instrument Tones using Frequency Modulation with a Complex Modulating Wave", *Computer Music* J. **1**(4), pp. 46-50

Serra, M.-H., Rubine, D. & Dannenberg, R. (1990). "Analysis and Synthesis of Tones by Spectral interpolation", *J. Audio Eng. Soc.* 38(3), pp. 111-128.

Smith, J. O. and Serra, X. (1987). "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation", *Proc. 1987 Int. Computer Music Conf.*, Urbana, IL, pp. 290-297.

So, K.F., & Horner, A. (2002). "Wavetable Matching of Inharmonic String Tones", *J. Audio Eng. Soc.* **50**(1/2), pp. 47-57.

Tan, B.T.G., Gan, S.L., Lim, S.M. & Tang, S.H. (1994). "Real-Time Implementation of Double Frequency Modulation (DFM) Synthesis", *J. Audio Eng. Soc.* **42**(11), pp. 918-926.

Tan, B. T. G. & Lim, S.M. (1996). "Automated Parameter Optimization for Double Frequency Modulation Synthesis Using the Genetic Annealing Algorithm", *J. Audio Eng. Soc.* **44**(1/2), pp. 3-15.

Tomisawa, N. (1981). "Tone Production Method for an Electronic Music Instrument", (U.S. Patent 4,249,447).

Zahorian, S. and Rothenberg, M. (1981). "Principal-Components Analysis for Low Redundancy Encoding of Speech Spectra", *J. Acoust. Soc. Am.*, **69**(3), pp. 832-845.

Zheng, H. and Beauchamp, J. (1999). "Analysis and Critical-Band-Based Group Wavetable Synthesis of Piano Tones", Proc. 1999 Int. Computer Music Conf., Beijing, China, pp. 9-12.

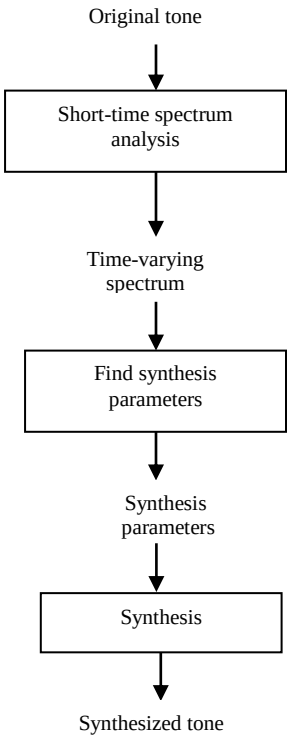


Fig. 1. Overview of the analysis/matching/resynthesis procedure.

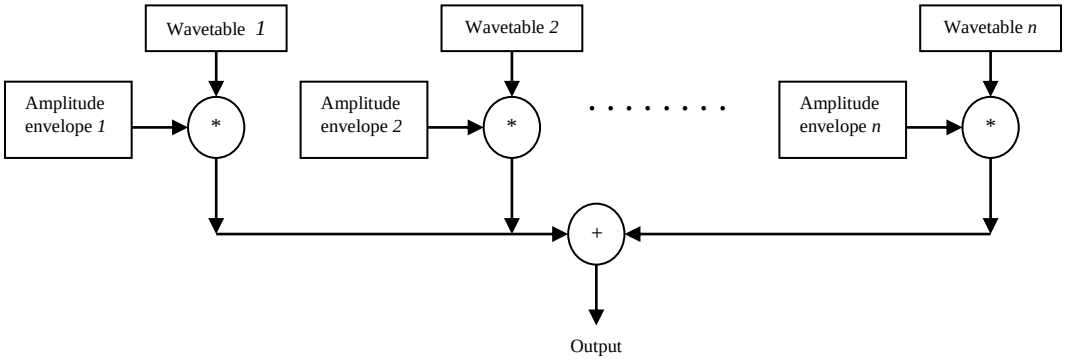


Fig. 2. A multiple wavetable synthesis block diagram.

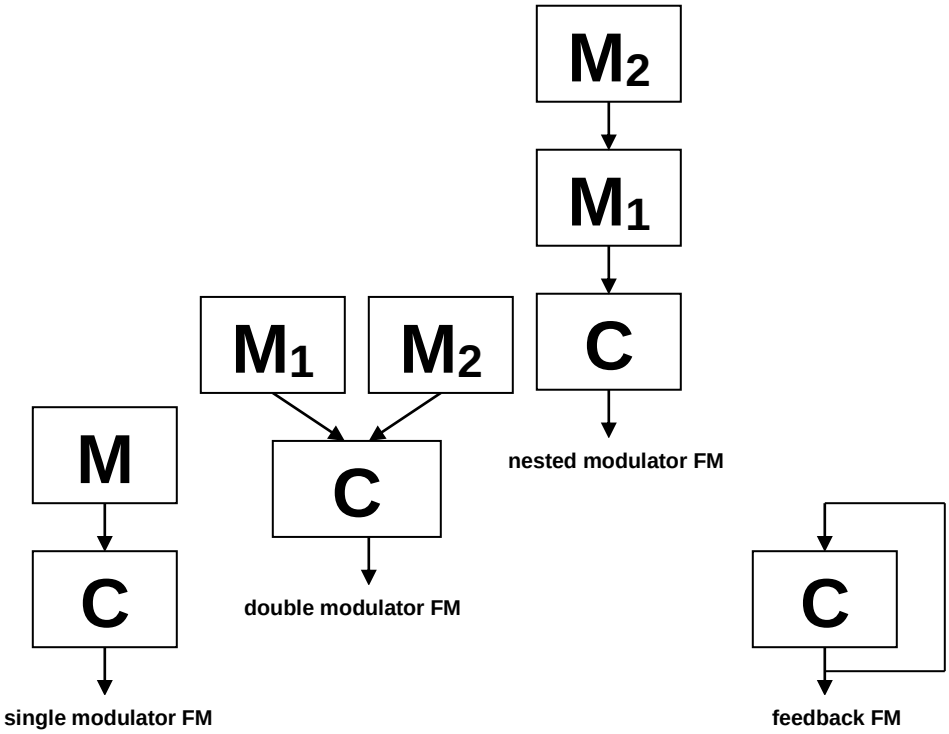
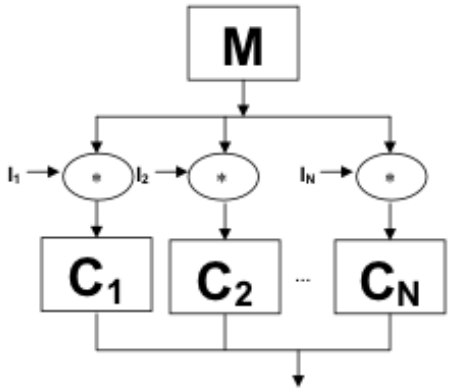
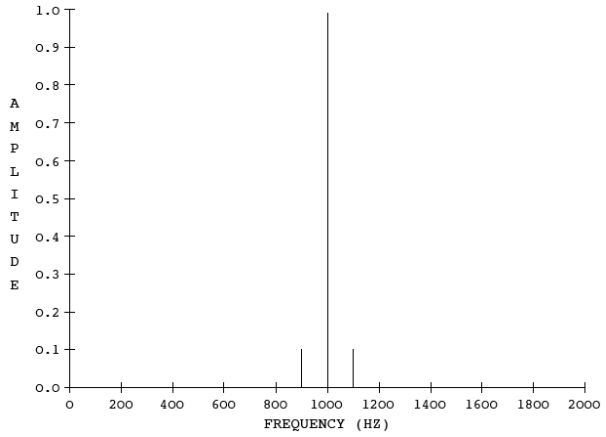


Fig. 3. Block diagrams for several basic types of FM synthesis.

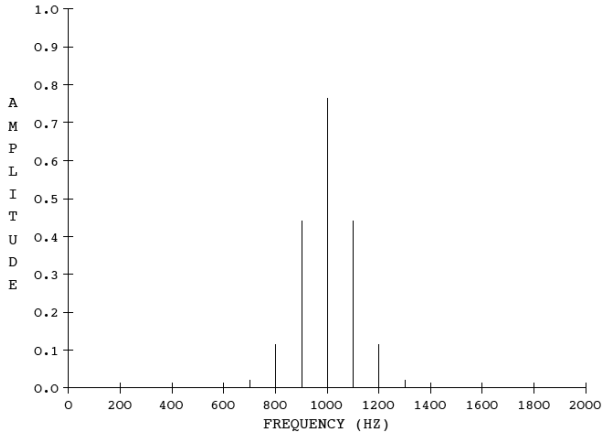


1

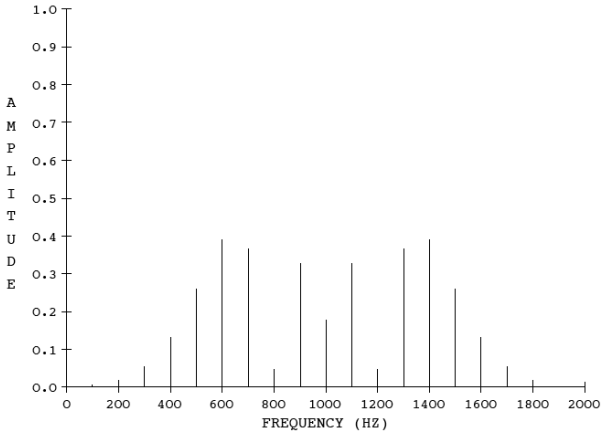
Fig. 4. Block diagram of single-modulator/multiple-carrier (formant) FM synthesizer.



(a)



(b)



(c)

Fig. 5. FM magnitude spectra for carrier frequency f_c, modulator frequency f_m, and index α = a) 0.2, b) 1.0, and c) 5.0.