UNIVERSITY OF CALIFORNIA

SANTA CRUZ

WAVESHAPING WITH DC OFFSET:

A COMPUTATIONALLY INEXPENSIVE METHOD OF MUSICAL SYNTHESIS

A thesis submitted in partial satisfaction

of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

By David Van Brink

June 1993

ABSTRACT

ones with complex spectral evolutions by several computationally cheap methods. One of these, “waveshaping”, is traditionally performed by submitting a sine wave, of varying amplitude, to a polynomial function. Both the sine wave and the polynomial function, or “shaping function”, are implemented via table lookups. The properties of the resulting waveform have been well analyzed and understood. The present work expands the technique to include a sine wave of varying amplitude *summed with a varying DC offset*. The properties of the resulting waveform are analyzed.

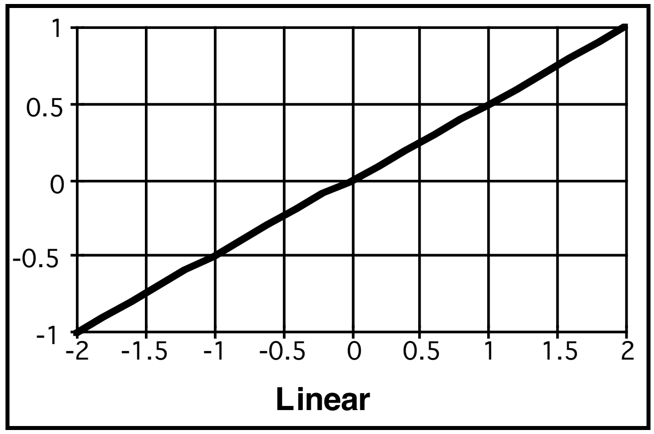
Also, a user interface for controlling these two parameters is proposed and demonstrated. The principles of this interface may be applied to other difficult-to-control methods of musical synthesis.

Introduction

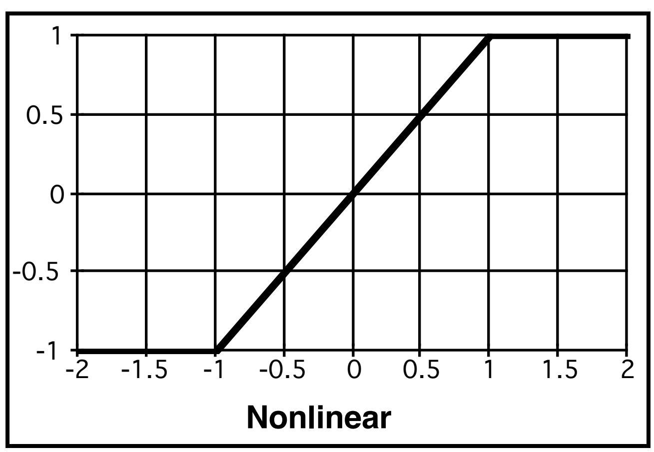
Background

Recently, I was peripherally involved with some software engineering work for a sound-playback system on a commercial, general-purpose microcomputer. It was desired to play between one and several sampled sounds simultaneously (depending upon the available CPU bandwidth), using software to resample and mix. The output channel and the sampled sounds were both 8 bits wide, linearly encoded. The problem was that if several channels were summed, and then divided down into the 8-bit range again, there was a significant loss of data. The solution arrived at, quite naively, was to use a nonlinear transfer function that retained full resolution for small amplitudes, and rolled off towards the extrema.

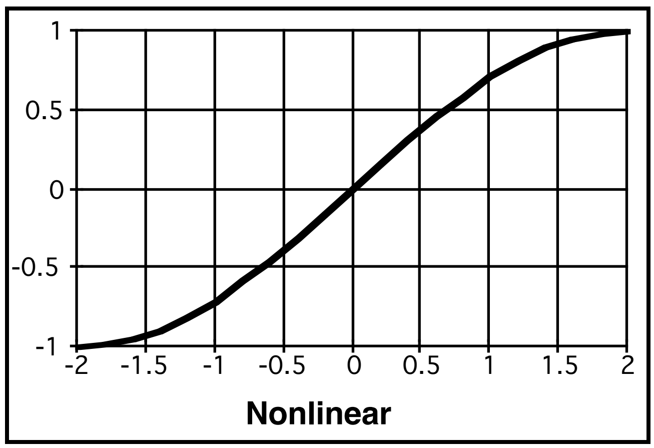
For example, suppose we wish to mix two signals. If we consider the 8-bit range to represent the amplitudes from -1 to +1, then summing the two signals yields a value with a range from -2 to +2. Dividing by two brings the signal back into a legal range for output. The result is that each signal is played at one half its original volume, and with only 7 bits of effective resolution. This was undesirable because most of the time the audio played too quietly, and 8 bits is already a very poor resolution of signal for audio use; 7 or fewer becomes nearly intolerable. The transfer function (for division-by-two) may be graphed as follows.



The sampled sounds we were playing, however, only occasionally required the full available signal range. That is, when two 8-bit signals were summed, they rarely exceeded an 8-bit value. One solution is to simply limit the output range to the allowable values. This results in full volume playback, and no loss of resolution, except when occasional “clipping” of the output channel occurs, which is aurally undesirable. (Hard clipping sounds quite harsh and buzzy; it is the sound produced by an overdriven transistor audio amplifier.) The transfer function for this form of clipping is shown in the following graph.



A compromise between these two methods is to use a “soft clipping” function. This function should retain full amplitude for small signals, and gradually clip for higher amplitude signals. While this introduces some harmonic distortion for high-amplitude signals, it was a viable tradeoff for the particular, moderately low-quality audio system involved. The final transfer function used is graphed below.

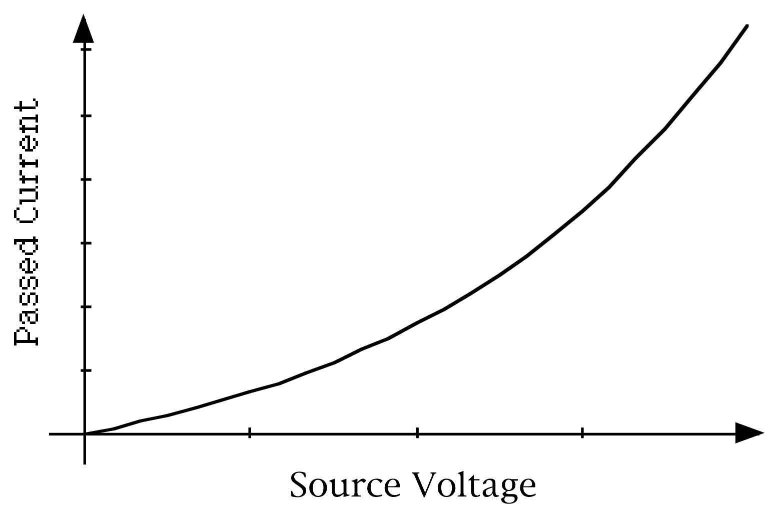


The transfer function was implemented as a lookup table. The thought occurred to me that by putting arbitrary functions into that lookup table, various forms of harmonic distortion could be induced on a submitted signal. The interesting feature of the distortion was that harmonics were *added* to the submitted signal. The method was clearly cheap to implement, and might possibly be useful in the context of musical synthesis.

Subsequent conversations and research revealed that, like most ideas with any merit, this one had been previously explored. However, as the topic still seemed intriguing, I looked for some missing bits of the published work on the subject, and have, I hope, filled in a minor crevice in the available literature.

Previous Work

Nonlinear functions have long been a factor in audio electronics: a negative factor. Audio work often consists ensuring fidelity to an input signal, rather than altering it. In 1970, R. A. Schaefer described the harmonics produced by passing a normalized sine wave through a transfer function specified by a polynomial or power series (Schaefer 1970). Schaefer paid particular attention to the transfer function produced by a semiconductor's *p-n* junction (the output current is $I_o (e^{kv} - 1)$ where v is the input voltage, and I_o and k are constants for the particular junction). Since he was employed by the Rodgers Organ Company, we may presume that combinations of diode junctions were being utilized to produce various organ timbres. Schaefer also gives a method for computing the polynomial transfer function given a particular desired output spectrum, which requires computing the coefficients of Chebyshev polynomials.



Typical semiconductor transfer function

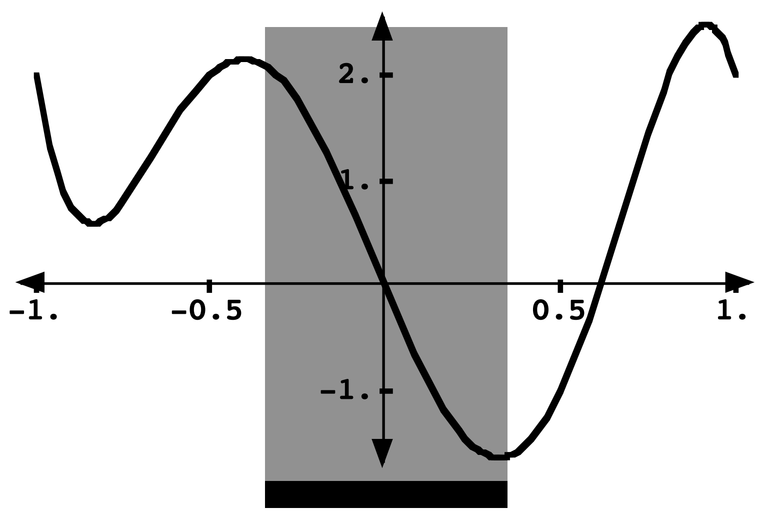
C. Y. Suen expanded upon Schaefer's equations by allowing a non-normalized sinusoidal input (Suen 1970). Symbolically, this involved adding a scalar to the input sine wave. This technique came to be known as nonlinear distortion, or "waveshaping" synthesis, and has been well analyzed in a digital music context since 1979 (Le Brun 1979, Arfib 1979).

Generally, the method described consists of passing a sine wave of varying amplitude through a polynomial shaping function (the "shaping function" is the "transfer function" described above). Changing the amplitude of the sine wave affects how much of the shaping function is used. A polynomial is convenient, because the output signal will have no harmonics higher than the highest degree of the polynomial. Certain power functions are nearly band limited when used as a shaping function. Sin x , for example, has factorially small coefficients, and so, while there are infinitely many harmonics in the produced signal, the higher ones are insignificant (and, in fact, disappear within the resolution of any digital audio system).

Several variations on this technique have been detailed in the literature. The published analyses describe what spectrum is produced given a particular shaping function, and submitted signal. The simplest case is a single submitted sine wave, although other functions have been analyzed. In some cases, a second stage of amplitude modulation is described, to decouple the shaping function's effective amplitude from the resultant signal volume. J. Beauchamp makes use of a digital filter to provide a realistic instrument formant for brass tone synthesis using waveshaping (Beauchamp 1979). Another possible post-process is ring modulation, to transpose the harmonics to an arbitrary frequency, and to add inharmonic frequency components.

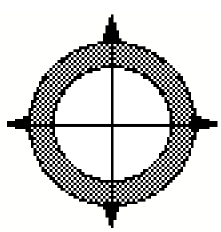
Without ring modulation, all harmonics produced are integer multiples of the frequency of the submitted sine wave. The ratios among these harmonics are the same regardless of the sine wave's frequency. De Poli analyzes the use of multiple waveshaping components of varying phases which are summed, to produce spectra that vary as a function of the frequency of the source sine wave (De Poli 1984).

One way to view waveshaping synthesis is to think of marching back and forth, sinusoidally, across a section of a shaping function. By modulating the amplitude of the submitted sine wave, different parts of the shaping function are used.



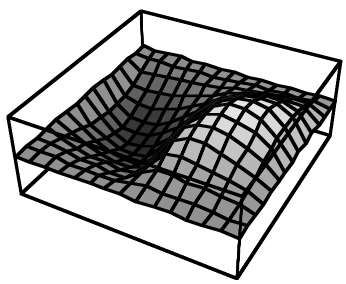
If 0.4 sin x is submitted to this shaping function, only part of the function is used.

One possible generalization of this technique is to use a shaping function with more than one dimension. D. Freed (Freed 1984) considers the case where the shaping function is realized in the complex plane, and a “phasor,” rather than a sinusoid, is submitted to it. A phasor is a signal of the form $e^{i\omega t}$. Thus, rather than using a portion of the linear shaping function, for a particular amplitude, a “ring” of the complex plane is utilized. The resultant waveform is then analyzed in terms of its phasor content, with harmonics of the form $e^{ih\omega t}$. Since varying the amplitude of the submitted phasor results in completely different rings being used, sharing no coordinates in common, complete control of all harmonic phasors for each amplitude may be realized with a sufficiently high-order shaping polynomial. Furthermore, the analysis using complex numbers becomes much simplified in several ways. Conveniently, a complex phasor can be interpreted as a real-valued audio signal by simply ignoring the imaginary component. This seems somewhat exciting, until one realizes that because the concentric rings used by various amplitudes are nonintersecting, the method is equivalent to using a large number of wavetables. Thus, this variation of waveshaping lacks the feature of reusing portions of the shaping function.



Area of the complex plane used by varying A slightly for the phasor $Ae^{i\omega t}$. Each value of A sweeps an independent set of points.

A more straightforward approach was taken by Y. Mitsuhashi (Mitsuhashi, 1982) and further analyzed by A. Borgonovo and G. Haus (Borgonovo & Haus 1986). This method consists of traversing a modified-ellipsoid "orbit" through a function of two variables. The function – analogous to the shaping function in standard waveshaping – is chosen such that the boundaries along the unit square, x or $y = \pm 1$, evaluate to zero, as well as the 1st derivative along that boundary. Thus, the orbit is free to "wrap around" the edges of this function, while still producing a smooth waveform. Their analysis was performed experimentally: a wave form was numerically calculated, and then analyzed for frequency content.



Function used for "Synthesis by Function of Two Variables." The boundary's value and first derivative are zero.

In the area of waveshaping synthesis, there is little formal method for synthesizing specific sounds. J. Beauchamp has laboriously synthesized brass tones, through much analysis for the specific instrument, and with A. Horner has implemented genetic algorithms to search for good matches to particular sounds (Beauchamp & Horner 1992).

D. Arfib (Arfib 1980) has used a number of waveshaping techniques in his compositions, including standard waveshaping to produce clarinet tones, and ring modulation to produce harmonic percussion sounds. By passing a sine wave through two consecutive shaping functions, each a Chebyshev polynomial, a complex evolution is rendered, which terminates on a single frequency some multiple of the submitted sine wave.

According to G. De Poli (De Poli 1983), "With waveshaping, listening and graphic considerations have more relevance than purely mathematical formulations," and "There is a large and not intuitive problem in choosing the [polynomial] coefficients, however, and further research is required."

The Present Work

I found two specific problems that seemed inadequately illuminated. First, several articles mention that adding a DC offset to the submitted signal can also affect timbre. This uses different horizontal sections of the shaping function. Using a DC offset to modify timbre is particularly appealing, because of its computational cheapness. However, the published analyses did not include this parameter; that omission is rectified in this document.

Second, I have developed a novel user interface, consisting of a means to naturally manipulate the evolution of two parameters over time. This requires a reasonable way to map the "timbre space" for a given shaping function to a human-interpretable form. I have started by mapping two parameters of waveshaping (amplitude and DC offset, or "shift") to the two axes of a graph. Within the graph, each point represents the spectrum produced by those two parameters. The challenge, then, is to find a good mapping from the many-dimensional space encompassed by an audio spectrum to the three-dimensional space afforded by possible colorings.

Superimposed on this spectral mapping is a path through that space, representing the evolution of the two parameters over time.

This mapping may perhaps be usefully applied to other spectral spaces, such as those of FM synthesis.

The next sections detail analyses of various waveshaping synthesis techniques which include the DC offset parameter. Following those are a description and discussion of the implementation and user interface that has been built to experiment with standard waveshaping with DC offset.

Analysis

Analyses of several variations on waveshaping with DC offset will be presented. These variations share two important features in common. First, they can all be implemented quite cheaply via the generation of a sinusoid (or two, for the complex and two-dimensional variations) of variable amplitude, added to a constant, and passed through a lookup table. Second, the resultant spectrum is completely known, based upon straightforward numerical methods – at least for the ideal case, where sampling and digitization errors are ignored.

The three methods are 1) Standard Waveshaping With DC Offset, which adds a single new parameter to waveshaping as described by Arfib, Le Brun, and others, 2) Complex Waveshaping, which adds a complex DC offset parameter to waveshaping as described by Freed, and 3) Two-Dimensional Waveshaping, which has some similarity to the methods described by Mitsuhashi and others, but is defined in such a way as to be more fully susceptible to Fourier analysis.

The first method, Standard Waveshaping With DC Offset, will be the most thoroughly explored. The formulae derived have been implemented in a realtime synthesis application, as described in the Implementation section.

Standard Waveshaping With DC Offset

Harmonics Produced By A Polynomial Shaping Function

The following section details the solution for the harmonic content of a sinusoid passed through a polynomial nonlinear transfer function, where the sinusoid has been modified in amplitude and DC offset.

We can construct a polynomial nonlinear transfer function,

$$s(x) = \sum_{n=0}^{d} p_n x^n \quad , \tag{1}$$

where

$s(x)$ is the shaping function (applied to x),
d is the highest degree of the polynomial,
p_n are the polynomial coefficients.

The function (resultant waveform) for a cosine wave, with amplitude scaling and DC offset applied, submitted to a polynomial shaping function is

$$f(t) = s(\cos t) = \sum_{n=0}^{d} p_n (A \cos t + S)^n \tag{2}$$

where

t is time (sans pitch ratio),
$f(t)$ is the resultant waveform as a function of time,
A is the amplitude of the cosine wave submitted to the shaping function,
S is the DC offset, or “shift” through the function.

Since $f(t)$ is an even function, it can be described as a sum of cosines. Furthermore, since the function is periodic (with period 2π), it may be described as a sum of cosines of integer frequency ratios:

$$f(t) = \sum_{h=0}^{\infty} a_h \cos ht \quad , \tag{3}$$

where a_h is the amplitude of the h-th harmonic.

The Fourier series will be used to determine the amplitude of each harmonic of the base frequency. Only the cosine transform need be used, as the phase of each harmonic is known. The formula for the amplitude of each frequency component is given by

$$\boxed{a_h = \frac{1}{\pi}\int_{-\pi}^{\pi} \sum_{n=0}^{d} p_n (A \cos t + S)^n \cos ht \, dt} \tag{4}$$

A bit of manipulation can be done on this. First, we can express $(A \cos t + S)^n$ as a combinatorial sum.

$$a_h = \frac{1}{\pi}\int_{-\pi}^{\pi} \sum_{n=0}^{d} p_n \sum_{k=0}^{n} \binom{n}{k} (A \cos t)^k S^{n-k} \cos ht \, dt \tag{5}$$

The nested summations can be rearranged by swapping the indices, giving

$$a_h = \frac{1}{\pi}\int_{-\pi}^{\pi} \sum_{k=0}^{d} \left[(A \cos t)^k \sum_{n=k}^{d} \binom{n}{k} p_n S^{n-k} \right] \cos ht \, dt \quad .$$

[6]

The first summation may be placed outside the integral.

$$a_h = \frac{1}{\pi} \sum_{k=0}^{d} \int_{-\pi}^{\pi} \left[(A \cos t)^k \sum_{n=k}^{d} \binom{n}{k} p_n S^{\,n-k} \right] \cos ht \, dt$$

[7]

Several of inner terms may similarly be factored out.

$$a_h = \frac{1}{\pi} \sum_{k=0}^{d} \left[A^{\,k} \left(\sum_{n=k}^{d} \binom{n}{k} p_n S^{\,n-k} \right) \int_{-\pi}^{\pi} \cos^k t \;\; \cos ht \, dt \right]$$

[8]

The definite integral of [8] has a solution (Gradshteyn & Ryzhik 1965). (See also Appendix A for the solution to this integral.)

$$\int_{-\pi}^{\pi} \cos^k t \;\; \cos ht \, dt = \begin{cases} k < h \text{ or } k + h \text{ is odd:} & 0 \\ \text{otherwise:} & \dfrac{\pi}{2^{k-1}} \dbinom{k}{\frac{k-h}{2}} \end{cases}$$

[9]

Substituting this into [8], we have

$$a_h = \frac{1}{\pi} \sum_{k=0}^{d} \left[A^{\,k} \left(\sum_{n=k}^{d} \binom{n}{k} p_n S^{\,n-k} \right) \begin{cases} n < h \text{ or } n + h \text{ is odd:} & 0 \\ \text{otherwise:} & \dfrac{\pi}{2^{k-1}} \dbinom{k}{\frac{k-h}{2}} \end{cases} \right].$$

[10]

More rearrangement yields

$$a_h = \frac{1}{\pi} \sum_{k=0}^{\frac{d-h}{2}} \left[A^{\,h+2k} \left(\sum_{n=h+2k}^{d} \binom{n}{h+2k} p_n S^{\,n-(h+2k)} \right) \frac{\pi}{2^{h+2k-1}} \binom{h+2k}{k} \right].$$

[11]

Finally, the π's are canceled, and the inner summation is made zero-based, to give a concise and computable (if not particularly closed form) solution.

$$a_h = 2 \sum_{k=0}^{\frac{d-h}{2}} \left[\frac{A^{h+2k}}{2^{h+2k}} \binom{h+2k}{k} \sum_{n=0}^{d-(h+2k)} \binom{n+h+2k}{n} S^n p_{n+h+2k} \right].$$

[12]

It is useful to note from equation [12] that the amplitude for any harmonic numbered greater than the degree of the polynomial is zero. Thus, the spectrum produced is band limited to the degree of the polynomial.

Comparison With Standard Result

If we take S as zero, we should have the solution for standard waveshaping. When $S =$ 0, the S^n term in the inner summation drives all terms to zero, except when $n = 0$. Hence, the inner summation may be removed, and zero substituted for n, yielding

$$a_h = 2 \sum_{k=0}^{\frac{d-h}{2}} \left[\frac{A^{h+2k}}{2^{h+2k}} \binom{h+2k}{k} p_{h+2k} \right].$$

[13]

This is identical to the formula for standard waveshaping given by Le Brun (Le Brun 1979).

Harmonics Produced In Terms Of Initial Harmonics

When designing an instrument for waveshaping synthesis, a known harmonic spectrum might be a useful starting point. From there, modulations to amplitude and shift might be explored.

As has been well documented in the literature regarding waveshaping, we can obtain a particular harmonic spectrum for the case $A = 1$ (and $S = 0$, by default) by constructing the shaping polynomial as a sum of Chebyshev polynomials of the first kind.

$$s(x) = \sum_{m=0}^{d} b_m T_m(x),$$

[14]

where

b_m is the amplitude of the m-th harmonic when $A = 1$ and $S = 0$.

T_m is the m-th Chebyshev polynomial.

This works because of the property of Chebyshev polynomials that $T_n(\cos t) = \cos nt$. Thus, when a cosine is submitted to this shaping function, at unity amplitude and no shift, we have

$$s(\cos t) = \sum_{m=0}^{d} b_m T_m(\cos t) = \sum_{m=0}^{d} b_m \cos mt.$$

[15]

The rightmost part of equation [18] matches equation [3]; by expressing the shaping function as a sum of Chebyshev polynomials, the harmonic amplitudes for unity amplitude and no shift follow immediately.

To express the dynamic spectral behavior in terms of the initial spectrum, we must derive the polynomial coefficients from the Chebyshev coefficients, which can then be applied to the previous analysis [12].

The coefficient of x^j for the m-th Chebyshev polynomial, T_m (Spanier & Oldham 1987) is

$$t_m^{(j)} = \begin{cases} \text{if } j > m \text{ or } j+m \text{ is odd: } 0 \\ \text{otherwise: } \dfrac{(-1)^{\frac{m-j}{2}} 2^j\, m \binom{\frac{m+j}{2}}{j}}{m+j} \end{cases} .$$

[16]

The coefficient for degree x^j for the polynomial expansion of the shaping function is then

$$p_j = \sum_{m=0}^{d} b_m t_m^{(j)} = \sum_{m=0}^{d} b_m \begin{cases} \text{if } j > m \text{ or } j+m \text{ is odd: } 0 \\ \text{otherwise: } \dfrac{(-1)^{\frac{m-j}{2}} 2^j\, m \binom{\frac{m+j}{2}}{j}}{m+j} \end{cases} .$$

[17]

Eliminating the conditional clause, we get

$$p_j = \sum_{m=0}^{\frac{d-j}{2}} b_{2m+j} \frac{(-1)^m 2^{j-1} (2m+j) \binom{m+j}{j}}{m+j} .$$

[18]

Now, we substitute this for the polynomial coefficients in equation [12].

$$a_h = 2 \sum_{k=0}^{\frac{d-h}{2}} \Bigg(\frac{A^{h+2k}}{2^{h+2k}} \binom{h+2k}{k} \sum_{n=0}^{d-(h+2k)} \binom{n+h+2k}{n} S^n \times \\ \sum_{m=0}^{\frac{d-(n+h+2k)}{2}} \left(b_{2m+n+h+2k} \frac{(-1)^m 2^{(n+h+2k-1} (2m+n+h+2k) \binom{m+n+h+2k}{m}}{m+n+h+2k} \right) \Bigg) .$$

[19]

Canceling some of the powers of 2 leaves

$$a_h = \sum_{k=0}^{\frac{d-h}{2}} A^{h+2k} \binom{h+2k}{k} \sum_{n=0}^{d-(h+2k)} 2^n \binom{n+h+2k}{n} S^n \times \sum_{m=0}^{\frac{d-(n+h+2k)}{2}} \left(b_{2m+n+h+2k} \frac{(-1)^m \ (2m+n+h+2k) \binom{m+n+h+2k}{m}}{m+n+h+2k} \right).$$

[20]

Comparison With Standard Result

Setting S to zero in equation [20] should produce Le Brun's result for standard waveshaping. Setting S to zero eliminates one of the summations yielding

$$a_h = \sum_{k=0}^{\frac{d-h}{2}} \left[A^{h+2k} \binom{h+2k}{k} \sum_{m=0}^{\frac{d-(n+h+2k)}{2}} \left(b_{2m+h+2k} \frac{(-1)^m \ (2m+h+2k) \binom{m+h+2k}{m}}{m+h+2k} \right) \right]$$

.

[21]

This is not exactly the same form as Le Brun's result. However, the rightmost part of the above equation can be manipulated to get the following transformation:

$$\frac{(2m+h+2k)\binom{m+h+2k}{m}}{m+h+2k} \rightarrow \binom{m+h+2k-1}{m-1} + \binom{m+h+2k}{m}$$

[22]

which, when substituted into equation [21], exactly matches Le Brun's form:

$$a_h = \sum_{k=0}^{\frac{d-h}{2}} \left[A^{h+2k} \binom{h+2k}{k} \sum_{m=0}^{\frac{d-(n+h+2k)}{2}} \left[b_{2m+h+2k} (-1)^m \left(\binom{m+h+2k-1}{m-1} + \binom{m+h+2k}{m} \right) \right] \right].$$

[23]

Harmonics Produced By Shaping Function of Sin X

If we use a single sine wave as a shaping function, we can substitute its polynomial expansion into the equation for waveshaping with shift. The polynomial (or power series) expansion of sin x is

$$\sin x = \frac{x^{1}}{1!} - \frac{x^{3}}{3!} + \frac{x^{5}}{5!} - \frac{x^{7}}{7!} \cdots .$$

[24]

If the sine wave is to be of arbitrary frequency, then we have

$$\sin \mathrm{F}x = \frac{F^{1}x^{1}}{1!} - \frac{F^{3}x^{3}}{3!} + \frac{F^{5}x^{5}}{5!} - \frac{F^{7}x^{7}}{7!} \cdots$$

[25]

where

F is the frequency ratio of the sine wave.

The polynomial coefficients may be easily stated from this.

$$p_i = \begin{cases} i \text{ is even} : 0 \\ i \text{ is odd} : \dfrac{F^{i}(-1)^{\left(\frac{i-1}{2}\right)}}{i!} \end{cases}$$

[26]

These coefficients are substituted into the waveshaping solution [12], where the maximum degree has been set to ∞.

$$a_h = 2 \sum_{k=0}^{\infty} \left[\left(\frac{A}{2}\right)^{h+2k} \binom{h+2k}{k} \sum_{n=0}^{\infty} \binom{n+h+2k}{h+2k} S^{n} \times \begin{cases} n+h \text{ is even} : 0 \\ i \text{ is odd} : \dfrac{F^{n+h+2k}(-1)^{\left(\frac{n+h-1}{2}\right)}}{(n+h+2k)!} \end{cases} \right]$$

[27]

A few factorials cancel out, and the term F^{n+h+2k} is broken into F^{n} and F^{h+2k}.

$$a_h = 2 \sum_{k=0}^{\infty} \left[\frac{\left(\frac{A\,F}{2}\right)^{h+2k}}{k\,!\,(h+k\,)!} \sum_{n=0}^{\infty} \begin{cases} n+h \quad \text{is even} : 0 \\ \quad\quad \text{otherwise} : \dfrac{(FS\,)^n(-1)^{\left(\frac{n+h\,-1}{2}\right)}}{n\,!} \end{cases} \right]$$

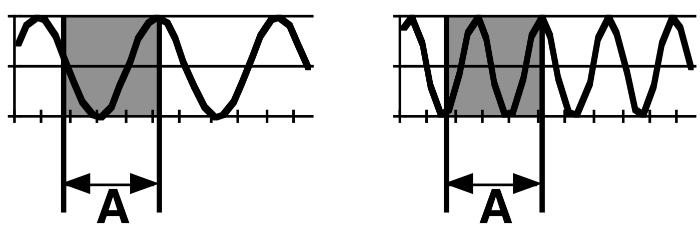
[28]

The right-hand summation is, in fact, an expansion for sine and cosine. Replacing it gives us a straightforward solution.

$$a_h = 2 \sum_{k=0}^{\infty} \left[\frac{\left(\frac{A\,F}{2}\right)^{h+2k}}{k\,!\,(h+k\,)!} \right] \times \begin{cases} h \quad \text{is even} : \sin SF \\ h \quad \text{is odd} : \cos SF \end{cases}$$

[29]

It is interesting to note that when the shift, S , is zero, increasing the amplitude, A , of the submitted cosine wave is *exactly* the same as increasing the frequency, F , of the shaping function sine wave. Similarly, the greater the frequency of the shaping function, the more rapid an effect changing the shift has.



Submitting a sine wave through a sinusoidal shaping function is, incidentally, equivalent to FM synthesis, where the carrier frequency is zero, the modulation frequency is that of the submitted sine wave, and the index of modulation is A .

Generation of Shaping Function for Arbitrary Shift and Amplitude

If we wish to define the resultant spectrum for particular values of A and S, we can construct a shaping function by taking the shaping function that produces the desired spectrum for $A = 1$ and $S = 0$, and then stretching and sliding the function to the desired position.

Equation [18] specifies the polynomial coefficients for a shaping function that produces a particular spectrum for $A = 1$ and $S = 0$.

$$p_j = \sum_{h=0}^{\frac{d-j}{2}} b_{2h+j} \frac{(-1)^h \, 2^{j-1} \, (2h+j) \binom{h+j}{j}}{h+j} \quad . \tag{18}$$

This is based directly upon a sum of Chebyshev polynomials. These coefficients are used to establish the shaping function, as given in equation [1].

$$s(x) = \sum_{n=0}^{d} p_n x^n \tag{1}$$

We can express the polynomial shaping function that produces a particular spectrum for arbitrary values of A and S in terms of the shaping function that produces that spectrum for $A = 1$ and $S = 0$. We would like the following relation to hold true:

$$\hat{s}(Ax + S) = s(x) \quad , \tag{30}$$

where $\hat{s}(x)$ is a polynomial shaping function having the spectral amplitudes b_n for amplitude A and shift S. Altering this into an operation on $s(x)$, we have $\hat{s}(x) = s\left(\frac{x}{A} - \frac{S}{A}\right)$. Substituting this into the polynomial summation of equation [1] gives us

$$\hat{s}(x) = \sum_{n=0}^{d} p_n \left(\frac{x}{A} - \frac{S}{A}\right)^n \quad . \tag{31}$$

We can change the inner exponentiation to a summation:

$$\hat{s}(x) = \sum_{n=0}^{d} p_n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{x}{A}\right)^j \left(\frac{-S}{A}\right)^{n-j} .$$

[32]

This double summation may be rearranged to group like powers of x , yielding

$$\hat{s}(x) = \sum_{j=0}^{d} \left[\frac{1}{A^j} \sum_{n=j}^{d} p_n \binom{n}{j} \left(\frac{-S}{A}\right)^{n-j} \right] x^j ,$$

[33]

which gives an expression of the coefficients of the desired shaping function, $\hat{s}(x)$, in terms of the coefficients of $s(x)$:

$$\hat{p}_j = \frac{1}{A^j} \sum_{n=j}^{d} p_n \binom{n}{j} \left(\frac{-S}{A}\right)^{n-j} ,$$

[34]

where $\hat{p}_j$ are the polynomial coefficients of the shaping function $\hat{s}(x)$.

The p_n's of this equation [34] could be replaced with the right-hand side of equation [18] (with the appropriate variable change) to give a solution for the polynomial coefficients of a shaping function directly in terms of the spectrum desired to be produced for values of A and S , but the form is ungainly. In practice, it is preferable to compute, first, the coefficients given by [18], and use these to compute the coefficients given by [34]. Conveniently, each $\hat{p}_j$ depends only upon the like-and-greater numbered p_n 's, so this transformation may be performed in-place.

It turns out, however, that in practice, the waveform resulting from most shaping functions grows rapidly out of range as A is increased from the value at which the shaping function was initially generated. For general applications, it is probably best to define the spectrum only at point of maximum amplitude, such as $A = 1$, $S = 0$.

Complex Waveshaping

D. Freed (Freed 1984) presents an analysis of waveshaping with complex arithmetic. In that analysis, the harmonics are determined in terms of “phasors,” which are, in some sense, the complex-number analog of a real sinusoid. A phasor has the form $e^{i\omega t}$, where ω is the angular frequency (and i is √-1). Although a phasor is a complex signal, it has a straightforward interpretation as an audio signal: simply throw away the imaginary part. A phasor can be thought of as a circular path in the complex plane. Thus, the real part of the phasor traces a cosine function, while the imaginary part traces a sine function. Discarding the imaginary part of the phasor changes its signal content: the Fourier transform of a phasor contains only a single negative frequency component, while the signal without the imaginary component contains positive and negative frequencies of equal value.

For the sake of completeness, the equivalent analysis is presented here, with the additional parameter of a complex DC offset parameter, S, or shift.

As with real-valued waveshaping, the shaping function is represented as a polynomial.

$$s(x) = \sum_{n=0}^{d} p_n x^n$$

where

$s(x)$ is the shaping function (applied to x),
d is the highest degree of the polynomial,
p_n are the polynomial coefficients, each a complex value.

Where a cosine wave was submitted to the shaping function in real-valued waveshaping, a phasor, with amplitude scaling and DC offset will be submitted to the complex shaping function.

$$f(t) = \sum_{n=0}^{d} p_n \left(Ae^{it} + S\right)^n$$

[35]

where

A is a scaling factor of the submitted phasor (it may be complex; this will only affect the phase of the resultant signal), and

S is a DC offset applied to the submitted phasor.

The "harmonics" produced are terms of the form e^{iht} where h is the number of the harmonic produced. The analysis consists of merely separating out those terms. First we apply the binomial theorem.

$$f(t) = \sum_{n=0}^{d} p_n \sum_{j=0}^{n} \binom{n}{j} S^{n-j} A^j e^{itj}$$

[36]

Next the summations are swapped, to group all like phasors together.

$$f(t) = \sum_{j=0}^{n} \left(\sum_{n=j}^{d} p_n \binom{n}{j} S^{n-j} A^j \right) e^{itj}$$

[36]

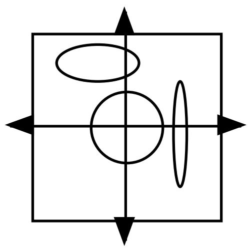
And, so, the amplitude of each harmonic phasor is simply the contents of the inner summation.

$$a_j = \sum_{n=j}^{d} p_n \binom{n}{j} S^{n-j} A^j$$

[37]

Two-Dimensional Waveshaping

Another musically applicable two-dimensional variation on waveshaping is to extend the polynomial with a second variable. The technique is to submit an ellipsoid to a polynomial of two variables.



Three possible submitted ellipsoids, plotted atop the range of a function of 2 variables.

As with one-dimensional waveshaping, the use of a polynomial ensures that the resultant waveform is band-limited, in this case to twice the greatest degree of the polynomial. This method is somewhat different than that described by Mitsuhashi (Mitsuhashi 1982) and A. Borgonovo and G. Haus (Borgonovo & Haus 1986), in that their shaping function was very simple, and was traversed with a more complex “orbit.”

The analysis proceeds quite analogously to the one-dimensional case already described. First, the shaping function is expressed in terms of x and y :

$$s(x,y) = \sum_{j=0}^{d} \sum_{k=0}^{e} p_{jk} x^j y^k ,\qquad [38]$$

where p_{jk} are the polynomial coefficients.

Next, the waveform produced is specified by substituting an ellipsoid for x and y .

$$f(t) = \sum_{j=0}^{d} \sum_{k=0}^{e} p_{jk} \left(A_x \cos\ \ t + S_x\right)^j \left(A_y \sin\ \ t + Sy\right)^k ,\qquad [39]$$

where

A_x and A_y are the two radii of the submitted ellipsoid, and
S_x and S_y specify the center of the submitted ellipsoid.

The exponentiated pieces are binomially expanded:

$$f(t) = \sum_{j=0}^{d}\sum_{k=0}^{e} p_{jk}\left(\sum_{m=0}^{j}\binom{j}{m} A_x^m \cos^m t \;\; S_x^{j-m}\right)\left(\sum_{n=0}^{k}\binom{k}{n} A_y^n \sin^n t \;\; S_y^{k-n}\right).$$

[40]

This can be rearranged to a more convenient form,

$$f(t) = \sum_{j=0}^{d}\sum_{k=0}^{e} p_{jk}\sum_{m=0}^{j}\;\sum_{n=0}^{k}\binom{j}{m}\binom{k}{n} A_x^m S_x^{j-m}\binom{k}{n} A_y^n S_y^{k-n} \cos^m t \;\; \sin^n t\,.$$

[41]

As with one-dimensional waveshaping, for fixed values of the parameters, in this case, A_x, A_y, S_x, and S_y, the resulting waveform is fixed with period 2π, and all frequencies present are harmonics of that fundamental. To determine the amplitudes of these harmonics, the Fourier series can be used. Unlike one-dimensional waveshaping, however, the various harmonics have no definite phase relationship; we cannot merely take the cosine transform. Taking the Fourier series, we have

$$a_h = \sum_{j=0}^{d}\sum_{k=0}^{e} p_{jk}\sum_{m=0}^{j}\;\sum_{n=0}^{k}\binom{j}{m}\binom{k}{n} A_x^m S_x^{j-m}\binom{k}{n} A_y^n S_y^{k-n}\,\frac{1}{\pi}\int_{-\pi}^{\pi} \cos^m t \;\; \sin^n t \;\; e^{iht} dt$$

[42]

where a_h is the amplitude of the h-th harmonic.

The solution to the integral part of this equation is given in Appendix A, as Integral III. We merely substitute it in here, and cancel the single occurrence of π.

$$a_h = \sum_{j=0}^{d}\sum_{k=0}^{e} p_{jk} \sum_{m=0}^{j}\sum_{n=0}^{k} \binom{j}{m}\binom{k}{n} A_x^m S_x^{j-m} \binom{k}{n} A_y^n S_y^{k-n} \;\; \mathbf{X}$$

$$\begin{cases} m+n+h \text{ is odd: } 0 \\ \text{else: } \dfrac{1}{i^n 2^{m+n-1}} \displaystyle\sum_{r=\max(0,\frac{m-n-h}{2})}^{\min(m,\frac{m+n-h}{2})} \binom{m}{r}\binom{n}{\frac{m+n-h}{2}-r}(-1)^{\frac{m+n-h}{2}-r} \end{cases}$$

[43]

A marvel of elegance and symmetry this form is not. It seems likely that there may be some possible simplifications to this result; alas, they shall not be here explored. However, we may at least establish the straightforward computability of the harmonic content of an ellipsoid submitted to a polynomial shaping function of two variables.

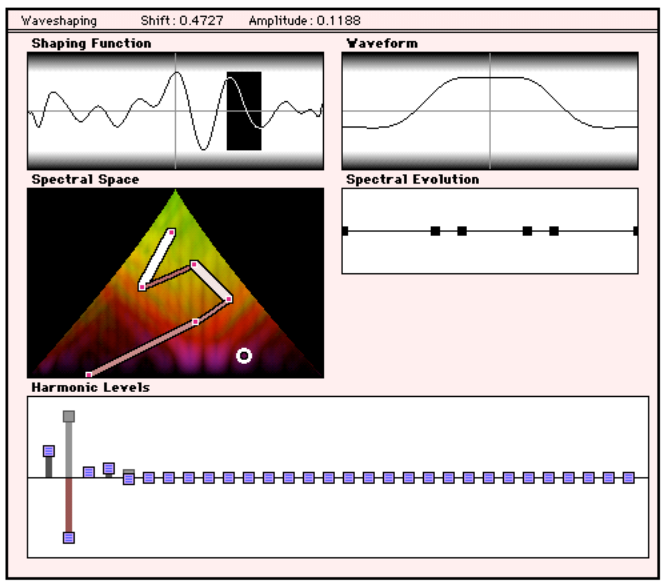
Implementation

I have written an application for the Macintosh personal computer which implements the technique of Standard Waveshaping With DC Offset. To give the user access to the various capabilities of this synthesis technique, a user interface had to be designed. Unfortunately, little support for the user interface for this sort of application could be found in the literature; common sense and good engineering have been used in its design. The justification for these interface elements will consist primarily of appeals to common sense in the User Interface section.

Since the implementation was on a low-powered personal computer, several implementation shortcuts have been used. Although these do not represent new theoretical work, they will nonetheless be presented here, as they are certainly of great practical value to anyone interested in using this synthesis method.

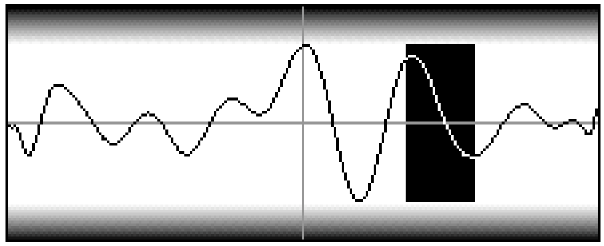
User Interface

To experiment with the technique of waveshaping synthesis, it was necessary to construct a software application that allowed all of the relevant parameters to be adjusted by the user. Some amount of feedback was desired, in order to render the process, as much as possible, understandable to the user. To this end, a user interface was designed and implemented that allows the user to modify the shaping function, and determine the evolution over time of the realtime parameters of amplitude and shift.



Above is a screen image of a window taken from the waveshaping application. The following sections will describe each pane of this window.

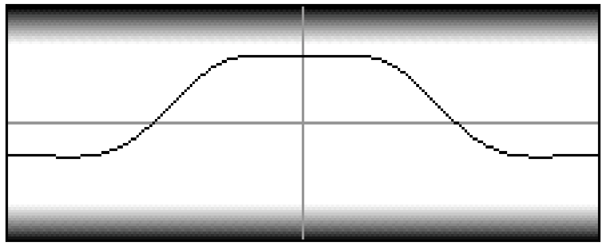
Shaping Function



The Shaping Function pane displays the shaping polynomial. This function is computed/modified based upon the harmonic level sliders in the Harmonic Levels pane. Both the horizontal and vertical axes represent the range [0, 1]. The highlighted area (black) shows the portion of the shaping function used by the “position marker” (white circle) in the Spectral Space pane. That is, the center of this highlighted section is the value of the shift parameter, and one half its width is the value of the amplitude parameter. (Alternatively, the left and right edges represent the range [S-A, S+A].)

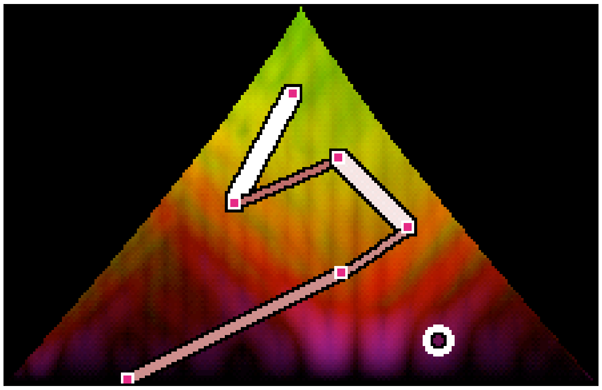
The shaping function is stored as a lookup table, and is recomputed as necessary when the sliders in the Harmonic Levels pane are adjusted.

Waveform



The Waveform pane shows the steady state waveform produced from the values for amplitude and shift specified by the “position marker” in the Spectral Space pane. Also, the waveform shown corresponds to the highlighted section of the Shaping Function pane. The waveform represents one full cycle of a cosine wave submitted to the highlighted portion of the shaping function.

Spectral Space



This Spectral Space pane is the densest pane, in terms of information presented. In this pane, the realtime parameters of amplitude and shift are placed in a two-dimensional space. The horizontal axis denotes the value of the shift parameter, ranging from -1 on the left to +1 on the right. This pane is aligned with the Shaping Function pane such that the value for the shift parameter have a spatial correspondence. That is, the two panes have identical horizontal extent, and represent the same [-1,1] range.

The vertical axis is the range of the amplitude parameter, varying from zero at the bottom, to unity at the top.

The colored triangular area (which is the full pane's width at the bottom, and peaked at the top) represents the allowable values for amplitude and shift. Because the shaping function is only defined for the range [-1,1], and the range of the shaping function used for a particular pair of values for amplitude and shift is [S-A, S+A], the sum of the magnitudes of amplitude and shift must not exceed unity. Only the points within the triangle satisfy that condition. Thus, at full amplitude, the only allowable shift value is zero.

The triangle of allowable values has slightly curved edges. This is because the amplitude scaling is not linear. To increase the efficiency of the implementation, a phase-offset-cancellation method is used to modulate the amplitude of the submitted sinusoid: this costs an addition and a memory reference rather than a multiplication. The vertical axis, then, represents not exactly the amplitude, but the phase offset used, which is related to the cosine of the amplitude. (This is described more fully in the “Sine Wave Amplitude Modulation” section.)

The white circle is a “position marker.” Whatever coordinates this marker lies on – it moves when the user clicks any place in the pane – are indicated by the selected area of the shaping function, the steady state waveform, and the relative harmonic values shown in the Harmonic Levels pane. While the user is clicking and dragging the position marker, the tone corresponding to that point in the spectral space is produced on the computer’s speaker.

The line segments specify a temporal evolution of the two realtime parameters. Starting from the topmost “knot,” the values for amplitude and shift are evenly and slowly altered over time, until they correspond to the coordinates of the 2nd knot, and so on, to the last knot near the bottom. The rate of travel along the segment is partially indicated by the width of the line segment: the thicker the line, the slower the travel. The duration of each segment can be changed in the Spectral Evolution pane; each segment has constant duration while being manipulated in the Spectral Space pane. Thus, when the user clicks on one of the knots and drags it, the overall area of each line segment remains constant (within the pixel resolution of the display, at least). The effect is very much like stretching a rubber band.

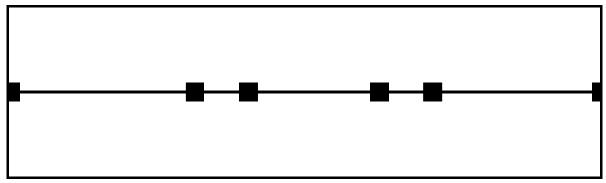
The choice of colorings for the Spectral Space is, perhaps, the most interesting part of this pane. The intent was to specify something meaningful about the perceived character of the spectrum produced at each coordinate in the spectral space. The values that determine the tone at each point are the quantities present of each harmonic multiple of the base frequency. Thus, several dozen (or more, if still higher harmonics are used!) values completely specify the tone. However, the color space visible to the human eye is generally considered to be three-dimensional. Thus, some values must be extracted from the specification of the tone, and collapsed down to the three available for color. In the present implementation, I have simply mapped the mean of the logs of harmonic numbers, standard deviation of the harmonic numbers, and overall volume to hue, intensity, and brightness, respectively (where "intensity" is the inverse of saturation).

Mapping volume to brightness seems reasonable: the louder the sound it, the easier it is to spot on the screen. Areas of the parameter space that produce no sound are black.

Changes in hue are very easy to see. Changes in overall harmonic content are verye asy to hear. For these reasons, I have mapped average harmonic to hue. The logs are taken of the harmonic numbers because our perception of pitch is logarithmic. That is, the aural difference between the fourth harmonic and the second harmonic is like the difference between the second and the first. The hue value for this representation of color is a circular scale; I have arbitrarily chosen to start the scale at blue, for low values, and end it at green, fir high values. The visual effect of this is to color the more "shrill" tones -- those with more upper harmonics -- with the brighter parts of the hue wheel.

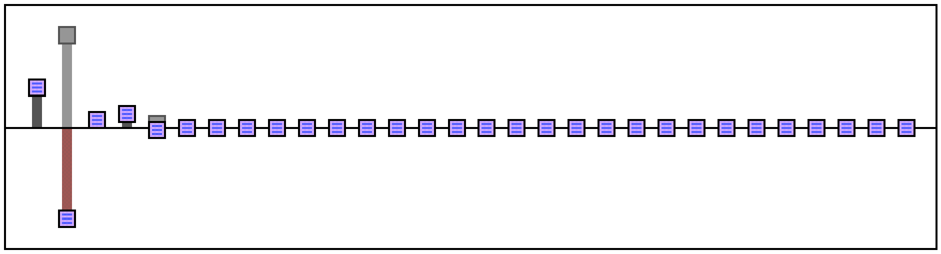
There are undoubtedly better spectrum-to-color mappings possible. There has been some research performed on dimensional reduction of timbre space (Wessel 1979), but this has focused primarily upon complete instrument timbres, and not extracted spectra. A set of experiments to determine perceptual distance between various spectra could be performed, which would, hopefully, lead to a more perceptually useful color mapping.

Spectral Evolution



The Spectral Evolution pane determines the timing between the knots drawn in the Spectral Space pane. The complete width of the pane represents a fixed duration, changeable from a menu item. The first, leftmost, knot is anchored, as is the rightmost one. The other knots may be moved left and right. The distance between any two knots represents the time necessary for the parameters to evolve from the knot on the left to the knot on the right.

Harmonic Levels



The Harmonic Levels pane shows the harmonic content of the steady state tone that would be produced given the values for amplitude and shift specified by the position marker in the Spectral Space pane. The vertical range of each slider is from -1 at the bottom to +1 at the top. Since “negative amplitudes” of a frequency component are equivalent to a phase offset of π, the negative amplitude *sounds* the same as a positive amplitude. However, arithmetically, a negative frequency component may lead to very different results in synthesis. To reflect these two contradictory views of negative frequency components, the sliders will display negative values both above and below the zero line.

The harmonic sliders can be adjusted, which specifies a particular harmonic spectrum to be produced by the present values (as shown by the position marker) for amplitude and shift. In practice, however, it is almost always necessary to adjust the harmonic sliders for the parameter values A = 1, S = 0. Adjusting the sliders for any other position in the spectral space tends to produce shaping functions that are wildly out of range.

Waveform Generation

Part of the appeal of waveshaping synthesis is that, while it doesn't model any particular physical process, it does produce complex spectra, and it is computationally very cheap to implement.

Sine Wave Amplitude Modulation

The synthesis described here requires a sinusoid of controllable amplitude. In order to avoid a multiplication operation (which was relatively slow on the platform used), a phase-cancellation technique was used to modulate the sinusoid's amplitude.

When a sinusoid is averaged with an out-of-phase copy of that sinusoid, the result is also a sinusoid, of a diminished amplitude:

$$\frac{\sin(t+p)+\sin(t-p)}{2}=\cos p\ \sin t\ ,$$

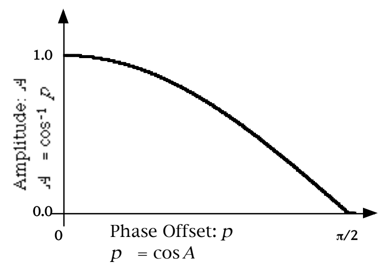
where

t is time (unscaled by frequency, here), and
p is phase offset.

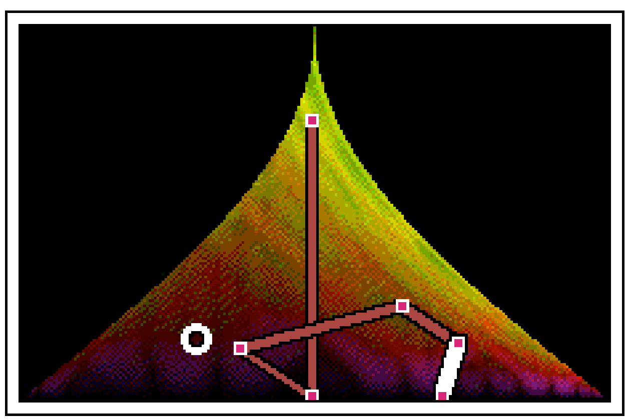
Since the sinusoid is being dephased forward and backward by equal amounts, before being averaged, the phase is preserved. To produce a sine wave of an amplitude between 0 and 1, we must choose p such that its cosine is the desired amplitude. The phase offset, p, ranges from 0 to $\pi/2$, to produce amplitudes from 1 to 0. Borrowing A from the earlier waveshaping equations,

$$A\ \sin t=\frac{\sin\left(t+\left(\cos^{-1}A\right)\right)+\sin\left(t-\left(\cos^{-1}A\right)\right)}{2}\ .$$

Because of the user interface used to control the waveshaping parameters, there is no reason that the amplitude adjustment must respond linearly. That is, the user may manipulate $p = \cos^{-1} A$ directly. However, because the slope of $\cos^{-1} A$ is infinite at $A = 1$, it is desirable to use only part of the inverse cosine curve.

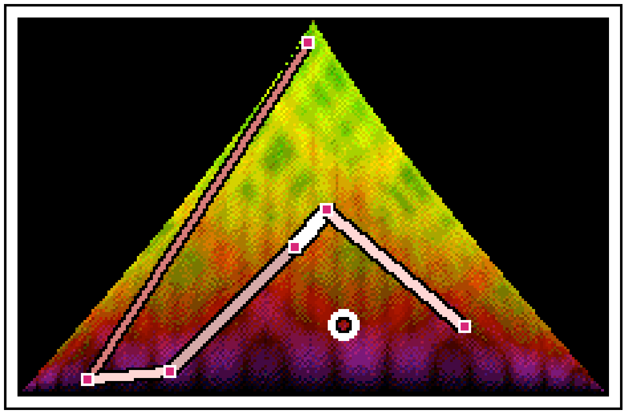


That is, when p is near zero, p must be changed a lot to make a small change in A; when p is near $\pi/2$, the relation between p and A is close to linear. (If the whole inverse cosine curve is utilized, then all timbre-paths must use a vertical portion to pass through the maximum amplitude point, which is inconvenient for the user.)



Minimum p value of $p_0 = 0$

If p ranges all the way to zero (no attenuation at all) then the resulting spectral space is very concave, and any approach or departure to the apex of the space must be vertical (A parameter change only), to avoid passing outside the allowable parameter combinations.



Minimum p value of $p_0 = \pi/4$

The edges of the usable part of the spectral space are slightly curved, but very nearly straight. Most importantly, the apex comes to a fairly wide angle, so there are many directions to approach and depart from that part of the space.

To address this, we constrain the values of p from p_0 to $\pi/2$. The sinusoid amplitudes produced then range from $\cos^{-1} p_0$ down to 0. To regain the full range from 0 to 1, we multiply the result by $1/\cos^{-1} p_0$ .The method finally used, then is

$$A \sin t = \frac{\sin(t + p) + \sin(t - p)}{2 \cos p_0}$$

where p ranges from p_0 to $\pi/2$. Thus, the amplitude ranges from 1 down to 0. For the implementation, the sine is computed with a table lookup, and the values in the table have been prescaled by $1/(2 \cos p_0)$. Another optimization in the implementation is to allow a phase error, by offsetting from t on only one of the sine computations.

$$A \sin(t + p) = \frac{\sin t + \sin(t + 2p)}{2 \cos p_0}$$

This saves a subtraction in one of the inner loops, at the expense of phase accuracy. So, as p is changed, the phase of the sine wave generated changes; this can be interpreted as a pitch disturbance. However, in practice, this does not seem to be a problem; indeed, rapidly changing either parameter (amplitude or shift) can result in pitch perturbations, but these effects do not sound unnatural.

Waveform Generation

The waveform generated has the form

$$f(t) = s(A \sin \omega t + S),$$

where

t is time,
f(t) is the resultant waveform,
s(x) is the shaping polynomial,
A is the amplitude of the submitted sinusoid,
S is the shift (DC offset) of the submitted sinusoid, and
ω is the base frequency of the produced tone.

Each sample requires an addition to "bump" the sine-lookup pointer by an increment appropriate to generate the desired frequency. The sine operation is a table lookup. The scaled sinusoid $A \sin \omega t$ is thus computed as described above, using two table lookups and two additions. Adding the shift is another addition, and passing this through the transfer function s(x) is one more table lookup.

I implemented this using 16 bit fixed point arithmetic. Following is the section of code that produces sampled audio output.

```
*w++ = shapeTable[ (
        (sineTable[(sample + dephase)& kSineTableMask)>>16]
```

```
        + sineTable[sample>>16])
        + (shift>>16)];
    sample = (sample + sampleStep) & kSineTableMask;
```

where

w is a pointer to the buffer being filled with digital audio sample data,

sineTable[] is a lookup table containing one complete sine wave,

shapeTable[] is the shaping polynomial,

sample[] is the sine-lookup pointer,

dephase is the amount to dephase the sine wave to attenuate its amplitude,

kSineTableMask is a mask to remove the upper bits of the sine table pointer, to keep the pointer within the table,

shift is the shift parameter for waveshaping, and

sampleStep is the amount to advance through the sine table on each sample to produce the desired pitch.

Thus, each sample requires a total of 4 additions, 2 AND operations, 3 shifts, and 3 table lookups. (Standard waveshaping without DC offset takes one fewer addition.)

The table sizes were chosen by trial and error, to achieve reasonable fidelity. If some form of interpolation were used, the tables could undoubtedly be made smaller. The shaping function table was set to 8192 entries long, by 8 bits deep (because the sound output hardware was an 8-bit DAC). In order to easily accommodate maximum amplitude, the sine table was scaled to have a maximum value of +/- 4095 (actually, larger, since there was some minimum amplitude attenuation for the reasons described earlier). The value -4096 was omitted to allow for a symmetrical table, and a true zero crossing. The sine table length was set to 4096, which, according to (Moore 1977) should have a signal-to-noise ratio of better than 60 dB at worst-case frequency stepping values (those near unity).

Conclusion

Presented here are all the necessary formulae for determining the spectra producible by several variations of the synthesis technique of nonlinear distortion, or waveshaping. The simplest of these techniques, Standard Waveshaping with DC Offset – submitting a sine wave to a polynomial shaping function, with the available realtime parameters of amplitude and shift – has been the most thoroughly analyzed. The algorithm has been implemented on a low-power personal computer (Macintosh CI, 25 MHz 68030), producing several simultaneous voices), demonstrating the algorithm's computational efficiency.

Waveshaping is an appealing method for sound synthesis, since it can be digitally implemented quite cheaply, and produces rich sounds with complex spectral evolutions. Like other methods of synthesis (frequency modulation, for example), there is some analytic difficulty in controlling the resultant waveform. The introduction of another parameter, "shift," provides a larger search space for particular spectra.

Although parameters for simulations of some instruments have been determined in previous research by analytic means and by automated parameter searches, an interactive means to adjust parameters can contribute greatly to the usefulness of waveshaping for musical composition. The interface and implementation described here is an initial attempt at that goal, making waveshaping synthesis even more practical for musical applications.

Bibliography

Arfib, D. 1979. "Digital Synthesis of Complex Spectra by Means of Multiplication of Nonlinear Distorted Sine Waves." *Journal of the Audio Engineering Society* 27(10):757-768.

Arfib, D. 1980. "The Musical Use of Nonlinear Distortion." *International Computer Music Conference Proceedings* 1980.

Beauchamp, J. 1979. "Brass Tone Synthesis by Spectrum Evolution Matching with Nonlinear Functions." *Computer Music Journal* 3(2): 35-43.

Beauchamp, J. and A. Horner. 1992. "Extended Nonlinear Waveshaping Analysis/Synthesis Technique." *International Computer Music Conference Proceedings* 1992.

Borgonovo, A. and G. Haus. 1986. "Sound Synthesis by Means of Two-Variable Functions: Experimental Criteria and Results." *Computer Music Journal* 10(3): 56-65.

De Poli, G. 1983. "A Tutorial on Digital Synthesis Techniques." *Computer Music Journal* 7(4):8-26.

De Poli, G. 1984. "Frequency Dependent Waveshaping." *International Computer Music Conference Proceedings* 1984.

Freed, D. 1984. "Waveshaping Analysis and Implementation: A Generalized Approach Using Complex Arithmetic." *International Computer Music Conference Proceedings* 1984.

Gradshteyn, I. S., and I. M. Ryzhik. 1965. Table of Integrals, Series, and Products. 4th Edition. New York: Harcourt Brace Jovanovich.

Le Brun, M. 1979. "Digital Waveshaping Synthesis." *Journal of the Audio Engineering Society* 27(4):250-265.

Mitsuhashi, Y. 1982. "Audio Signal Synthesis by Functions of Two Variables." *Journal of the Audio Engineering Society* 30(10):701-706.

Moore, F. R. 1977. "Table Lookup Noise for Sinusoidal Digital Oscillator." *Computer Music Journal* 1(2):26-29.

Schaefer, R. A. 1970. “Electronic Musical Tone Production by Nonlinear Waveshaping.” *Journal of the Audio Engineering Society* 18(6):675-676.

Spanier, J., and K. B. Oldham. 1987. An Atlas Of Functions. Washington: Hemisphere Publishing.

Suen, C. Y. 1970. “Derivation of Harmonic Equations in Nonlinear Circuits.” *Journal of the Audio Engineering Society* 18(6):675-676.

Wessel, D. L. 1977. “Timbre Space as a Musical Control Structure.” *Computer Music Journal* 3(2):45-52

Appendix: Solution To Integrals

Integral I

$$\int_{-\pi}^{\pi} e^{ikt}dt$$

[44]

The solution to this form, where k is an integer, will be useful in the forms to follow. Since the solution to $\int e^{ax}\,dx$ is e^{ax}/a, it will be useful to separate out the case where ikt is zero.

$$\int_{-\pi}^{\pi} e^{ikt}dt = \int_{-\pi}^{\pi} \begin{cases} k = 0 : 1 \\ k \neq 0 : e^{ikt} \end{cases} dt$$

[45]

Next, the two parts are solved, retaining the condition.

$$\int_{-\pi}^{\pi} e^{ikt}dt = \begin{cases} k = 0 : t \\ k \neq 0 : \dfrac{e^{ikt}}{ik} \end{cases} \Bigg|_{-\pi}^{\pi}$$

[46]

Substituting in the values of t gives us:

$$\int_{-\pi}^{\pi} e^{ikt}dt = \begin{cases} k = 0 : 2\pi \\ k \neq 0 : \dfrac{e^{ik\pi} - e^{-ik\pi}}{ik} \end{cases}$$

[47]

This may be simplified with the identity e^{it} - e^{-it} = (1/2i)sin t .

$$\int_{-\pi}^{\pi} e^{\,ikt}dt \;=\begin{cases} k = 0 : 2\pi \\ k \neq 0 : \dfrac{\sin k\,\pi}{-2k} \end{cases}$$

[48]

Since k is an integer, the sine will always evaluate to zero. Thus, we have the solution.

$$\int_{-\pi}^{\pi} e^{\,ikt}dt \;=\begin{cases} k = 0 : 2\pi \\ k \neq 0 : 0 \end{cases}$$

[49]

In use, this integral will be used in the solution to a Fourier transformation, which requires dividing by half the range of the integration. The π factor will usually cancel out.

Integral II

$$\int_{-\pi}^{\pi} \cos^{k} t \;\cos ht \;\; dt$$

[50]

This form is used to calculate the harmonics present in standard waveshaping, and in waveshaping with a DC offset. The variable k is an integer. First, the cosines are replaced by their exponential representations:

$$\int_{-\pi}^{\pi} \cos^{k} t \;\cos ht \;\; dt \;=\int_{-\pi}^{\pi} \frac{1}{2^{k}}\left(e^{it} + e^{-it}\right)^{k} \frac{1}{2}\left(e^{iht} + e^{-iht}\right) dt$$

[51]

The k - power expression may be replaced with its binomial expansion:

$$\int_{-\pi}^{\pi} \cos^k t \ \cos ht \ dt = \int_{-\pi}^{\pi} \frac{1}{2^{k+1}} \sum_{r=0}^{k} \binom{k}{r} e^{itr} e^{-it(k-r)} \left(e^{iht} + e^{-iht}\right) dt$$

[52]

This can be multiplied through, and expressed as a sum of two integrals with the form described above, Integral I.

$$\int_{-\pi}^{\pi} \cos^k t \ \cos ht \ dt = \frac{1}{2^{k+1}} \sum_{r=0}^{k} \binom{k}{r} \left(\int_{-\pi}^{\pi} e^{it(2r-k+h)} \, dt + \int_{-\pi}^{\pi} e^{it(2r-k-h)} \, dt \right)$$

[53]

The two integrals are replaced with their solutions, via Integral I.

$$\int_{-\pi}^{\pi} \cos^k t \ \cos ht \ dt = \frac{1}{2^{k+1}} \sum_{r=0}^{k} \binom{k}{r} \left(\begin{cases} r = \frac{k+h}{2} : 2\pi \\ else: 0 \end{cases} + \begin{cases} r = \frac{k-h}{2} : 2\pi \\ else: 0 \end{cases} \right)$$

[54]

The conditions, $r = (k + h)/2$ and $r = (k - h)/2$, will be true over the range for r if and only if $k + h$ is even, and $k \geq h$, and, in that case, they will both be true. That fact can be used to remove the summation, and simplify the form to a single condition, which completes the solution.

$$\int_{-\pi}^{\pi} \cos^k t \ \cos ht \ dt = \begin{cases} k \geq h \text{ and } k+h \text{ is even: } \dfrac{\pi}{2^{k-1}} \dbinom{k}{\frac{k-h}{2}} \\ \text{else: } 0 \end{cases}$$

[55]

This result is given in a rather more roundabout form by (Gradshteyn & Ryzhik 1965).

Integral III

$$\int_{-\pi}^{\pi} \cos^m t \ \ \sin^n t \ e^{iht} \ dt$$

[56]

This form arises when evaluating the harmonics produced by waveshaping with a polynomial of two variables, a two-dimensional form of waveshaping. m and n are integers. We proceed similarly to the evaluation of Integral II. First, the cosine and sine are replaced by their exponential forms.

$$\int_{-\pi}^{\pi} \cos^m t \ \ \sin^n t \ e^{iht} \ dt \ = \int_{-\pi}^{\pi} \frac{1}{2^m}\left(e^{it} + e^{-it}\right)^m \frac{1}{i^n\, 2^n}\left(e^{it} - e^{-it}\right)^n \ e^{iht} \ dt$$

[57]

The two power expressions are replaced by their binomial summations.

$$\int_{-\pi}^{\pi} \cos^m t \ \ \sin^n t \ e^{iht} \ dt \ = \int_{-\pi}^{\pi} \frac{1}{2^m} \sum_{r=0}^{m} \binom{m}{r} e^{itr} \ e^{-it(m-r)} \ \times$$

$$\frac{1}{i^n\, 2^n} \sum_{s=0}^{n} \binom{n}{s} e^{its} e^{-it(n-s)} (-1)^{n-s} \ e^{iht} \ dt$$

[58]

This can be regrouped to a more compact form, with the exponentials all combined.

$$\int_{-\pi}^{\pi} \cos^m t \quad \sin^n t \ e^{iht} \ dt = \frac{1}{i^n \, 2^{m+n}} \sum_{r=0}^{m} \sum_{s=0}^{n} (-1)^{n-s} \binom{n}{s} \binom{m}{r} \times \int_{-\pi}^{\pi} e^{it(2r+2s-m-n+h)} \, dt$$

[59]

This matches the form solved as Integral I, which may be substituted in.

$$\int_{-\pi}^{\pi} \cos^m t \quad \sin^n t \ e^{iht} \ dt = \frac{1}{i^n \, 2^{m+n}} \sum_{r=0}^{m} \sum_{s=0}^{n} (-1)^{n-s} \binom{n}{s} \binom{m}{r} \times \begin{cases} r+s = \dfrac{m+n-h}{2} : 2\pi \\ \text{otherwise} : 0 \end{cases}$$

[60]

The conditional will always evaluate to zero unless $m + n - h$ is even, and $m + n \geq h$. This can be simplified, then, to a single summation and condition, which is the solution:

$$\int_{-\pi}^{\pi} \cos^m t \quad \sin^n t \ e^{iht} \ dt =$$

$$\begin{cases} \begin{matrix} m+n+h \text{ is odd} \\ \text{or } h > m+n \end{matrix} : 0 \\ \text{else} : \dfrac{\pi}{i^n \, 2^{m+n-1}} \displaystyle\sum_{r=\max\left(0, \frac{m-n-h}{2}\right)}^{\min\left(m, \frac{m+n-h}{2}\right)} \binom{m}{r} \binom{n}{\frac{m+n-h}{2} - r} (-1)^{\frac{m+n-h}{2} - r} \end{cases}$$

[61]

If we set n to zero, then we should have, immediately, Integral II. The only value for the range of index r is $(m - h)/2$; the summation is removed.

$$\int_{-\pi}^{\pi} \cos^m t \ \ \sin^0 t \ \ e^{iht} \ dt \ =$$

$$\begin{cases} \begin{matrix} m+h \text{ is odd} \\ \text{or } h > m \end{matrix} : 0 \\ \text{else} : \dfrac{\pi}{2^{m-1}} \dbinom{m}{\frac{m-h}{2}} \dbinom{0}{\frac{m-h}{2} - \frac{m-h}{2}} (-1)^{\frac{m-h}{2} - \frac{m-h}{2}} \end{cases}$$

[62]

This is then trivially reduced to the result of Integral II.

$$\int_{-\pi}^{\pi} \cos^m t \ \ e^{iht} \ dt \ = \begin{cases} \begin{matrix} m+h \text{ is odd} \\ \text{or } h > m \end{matrix} : 0 \\ \text{else} : \dfrac{\pi}{2^{m-1}} \dbinom{m}{\frac{m-h}{2}} \end{cases}$$

[63]