Wavetable Synthesis 101, A Fundamental Perspective

Robert Bristow-Johnson
Wave Mechanics, Inc.
45 Kilburn St., Burlington VT 05401 USA
robert@wavemechanics.com

ABSTRACT: Wavetable synthesis is both simple and straightforward in implementation and sophisticated and subtle in optimization. For the case of quasi-periodic musical tones, wavetable synthesis can be as compact in data storage requirements and as general as additive synthesis but requires much less real-time computation. This paper shows this equivalence, explores some suboptimal methods of extracting wavetable data from a recorded tone, and proposes a perceptually relevant error metric and constraint when attempting to reduce the amount of stored wavetable data.

0 INTRODUCTION

Wavetable music synthesis (not to be confused with common PCM sample buffer playback) is similar to simple digital sine wave generation [1] [2] but extended at least two ways. First, the waveform lookup table contains samples for not just a single period of a sine function but for a single period of a more general waveshape. Second, a mechanism exists for dynamically changing the waveshape as the musical note evolves, thus generating a quasi-periodic function in time.

This mechanism can take on a few different forms, probably the simplest being linear crossfading from one wavetable to the next sequentially. More sophisticated methods are proposed by a few authors (recently Horner, et al. [3] [4]) such as mixing a set of well chosen basis wavetables each with their corresponding envelope function as in Fig. 1. The simple linear crossfading method can be thought of as a subclass of the more general basis mixing method where the envelopes are overlapping triangular pulse functions. In that case, only two wavetables are being mixed at any one instance of time as indicated in Fig. 2. In any case, the amount of data being stored and used for this synthesis method is far less than just the PCM sample file of the note. This is because wavetable synthesis takes advantage of the quasi-periodic nature of the waveform to remove redundancies and to reduce the data set.

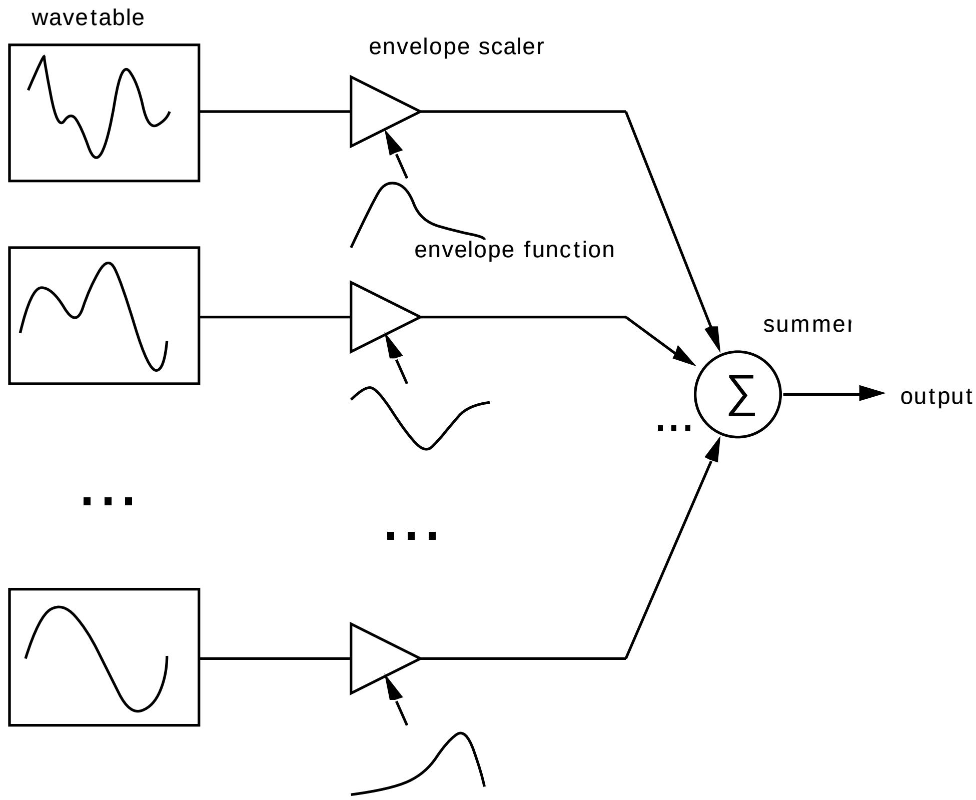

Figure 1 Arbitrary Wavetable Mixing

Wavetable synthesis is well suited for synthesizing quasi-periodic musical tones because wavetable synthesis can be as compact in data storage requirements and as general as additive synthesis but requires much less real-time computation. This is because it precomputes the inverse Discrete Fourier Transform (DFT) of the waveform spectrum before playback rather than computing the inverse DFT in real-time which is essentially what additive synthesis does when summing the outputs of many sine-wave oscillators. With waveform tables precomputed, real-time synthesis is reasonably simple to implement. That being the case, it seems odd that there are not many commercial examples of wavetable synthesizers implemented.

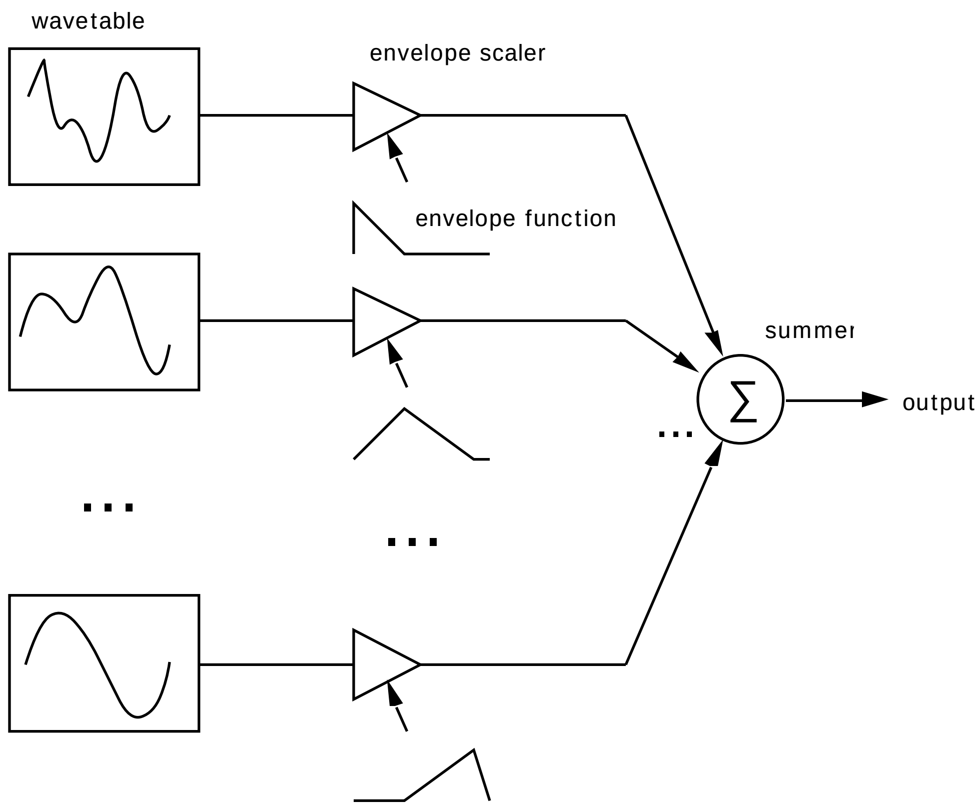

Figure 2 Sequential Crossfading of Wavetables

1 WAVETABLE MECHANICS

As mentioned above, the first ingredient in wavetable synthesis is a static waveform generator that uses a circular table of sequential waveform values, a phase accumulator for address generation, and some means of interpolation between neighboring wavetable samples because the address generated by the phase accumulator will not generally fall exactly on a wavetable midsample [1] [2]. Fig. 3 illustrates the waveform generator. Two issues concerning wavetable size, K, are the modulo arithmetic used in the circular address generator (usually indicating K being a power of 2) and the interpolation mechanism. If linear interpolation (or worse yet, drop-sample interpolation) is used, a larger wavetable is required to restrain interpolation error than if a more legitimate method of fractional sample interpolation [5] [6], such as that used in sample rate conversion, is being used. However if, because of limited real-time computational speed, a simpler interpolation method is to be used, a smaller wavetable that is archived and inactive can be expanded into larger wavetable, using higher quality interpolation, when it is loaded and made active.

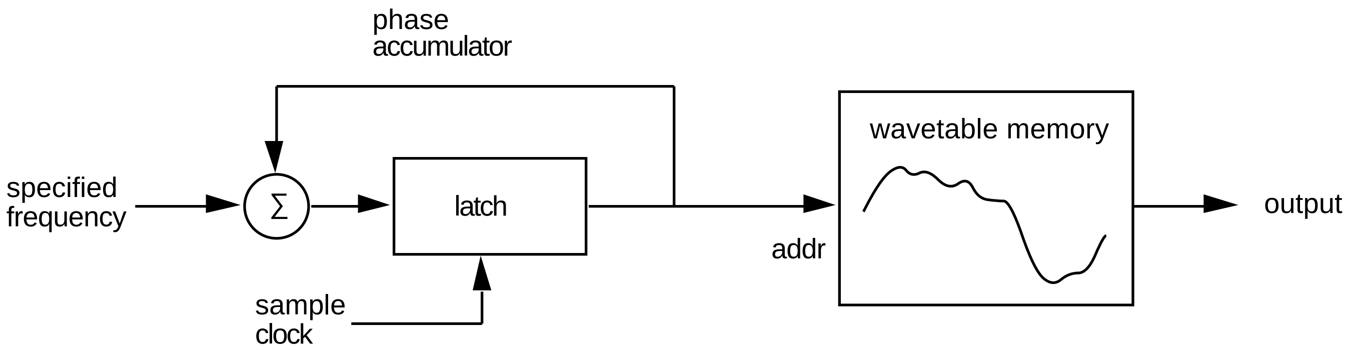

Figure 3 Phase Accumulator Wavetable Oscillator

The other ingredient for wavetable synthesis is a mechanism to dynamically change the waveform as the musical note proceeds in time. The method described by Horner, et al. [3] [4] that involves mixing a finite set of static and phase-locked wavetables, each scaled by individual envelope functions, appears to be conceptually the most general. It is shown in Fig. 1. At least one commercial product implements this method in hardware. The Sequential Circuits Prophet VS vector synthesizer which has four wavetables and envelopes which are derived from a nonstationary (or time varying), two dimensional vector. The remaining tasks, to both design the most appropriate set of wavetables (as well as selecting and minimizing the number of static wavetables) and to determine their envelope functions, are discussed by Horner. Two particular subclasses of this general wavetable mixing might be noted:

One is the case where each wavetable is an integer number of cycles (a harmonic) of sine and cosine functions up to a finite number of wavetables (and harmonics). Since the sine and cosine wave of the same frequency can be mixed to result in a sine wave of that frequency and any arbitrary amplitude and phase, this method is simply equivalent to sinusoidal additive synthesis. It is very general (if the wavetables are phase locked, the result will still be quasi-periodic), but requires many wavetables (twice the number of the highest harmonic) that must all be scaled and mixed in real time thus removing any computational advantage in efficiency that wavetable synthesis promises.

The other subclass is that of crossfading from one wavetable to the next (called “wavestacking” in [7]). In this case, the envelope functions are overlapping in such a way that no more than two are non-zero at any one instance of time. This means that, although there may be many wavetables to proceed through (memory is cheap), only two are mixed at any one time. This is computationally inexpensive and easy to conceptualize especially if linear crossfading is used. In this case, the envelope functions are overlapping triangular pulse functions as shown in Fig. 2.

It is reasonably easy to show how a set of sequential wavetables can be extracted directly from a quasi-periodic input note without having to go into the frequency domain. Some of the same principles used in pitch-synchronous short time Fourier analysis (STFT) stopping short of performing any Fourier Transform are used to do this.

The first step is determining the pitch or fundamental frequency or period of the quasi-periodic input at a given time in the note where a wavetable is to be extracted. Many papers ([8] - [11] are a few) and a few patents [12] exist for the problem of estimating pitch for difficult signals, but for the case of nearly periodic inputs, simple autocorrelation type of methods work fine. By evaluating the average magnitude (or magnitude squared) difference function (AMDF) Eq. (1) below as shown in Fig. 4, and correctly picking minima of the AMDF, one can safely infer the period length, τ, to a precision of fraction of a unit sample time and thus the fundamental frequency, f_0, in the vicinity of a given time, t_0.

$$\gamma_{t_0}(\delta) \equiv \int_{-\infty}^{\infty} \left| x\left(t + \tfrac{\delta}{2}\right) - x\left(t - \tfrac{\delta}{2}\right) \right|^2 w(t - t_0)\,dt \qquad (1)$$

$$\gamma_{t_0}\left(\tau(t_0)\right) = \min_{\Delta < \delta}\left\{\gamma_{t_0}(\delta)\right\}$$

$$\text{where} \quad \gamma'_{t_0}(\Delta) = 0$$

$$\text{and} \quad \gamma'_{t_0}(\delta) > 0 \quad \text{for} \quad 0 < \delta < \Delta$$

where $w(t - t_0) \geq 0$ is a window function centered at t_0 and width wider than any anticipated period τ. τ has an implied dependence on t_0 and is sometimes abbreviated from $\tau(t_0)$. In the example of Fig. 4, Δ is about 65 sample units (at the first local maximum) and a period τ of around 335 samples would be inferred since $\gamma(\tau)$ is the global minimum value of the AMDF for any lag greater than Δ. Interpolation around the global minimum would be employed to compute a value of τ to a precision of a fraction of a sample.

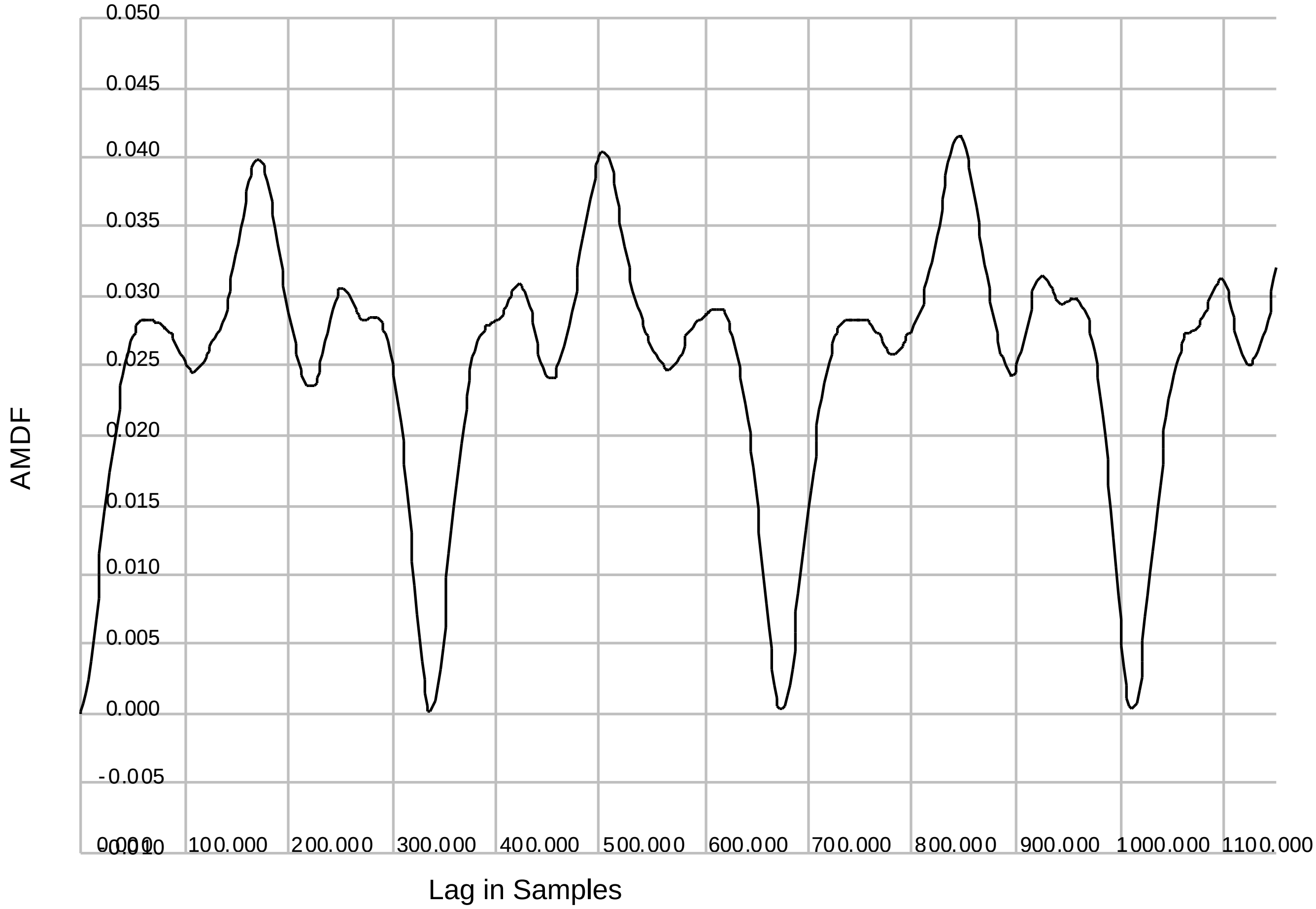

Figure 4 Average Magnitude (Squared) Difference Function

The second step is to periodically extend the quasi-periodic input from t_0 to infinity. This could be done by simply hacking off all of the input except for one period centered at t_0 (functionally equivalent to applying a rectangular window shown in Fig. 5) and then periodically repeating the one cycle in both directions of time, ad infinitum. Of course, this would introduce a discontinuity at each splice unless the input was perfectly periodic. To avoid this, one can use a window that more gracefully truncates the input outside of the period in the t_0 neighborhood. This window must have a complimentary fade-in and fade-out characteristic as discussed in [13]. The result is a single windowed period or wavelet defined as

$$\hat{x}_{t_0}(t) \equiv x(t) w_n\left(\frac{t-t_0}{\tau(t_0)}\right)$$

where $w_n(\beta)$ is a normalized, complimentary window function such that

$$
\begin{aligned}
&w_n(-\beta) = w_n(\beta) \\
&w_n(0) = 1 \\
&w_n(\beta) = 0 && \text{for} \quad |\beta| \geq 1 \\
&w_n(\beta - 1) + w_n(\beta) = 1 && \text{for} \quad 0 \leq \beta \leq 1
\end{aligned}
$$

Two suitable candidates for complimentary normalized windows are:

$$
w_n(\beta) = \begin{cases} \frac{1}{2}\left(1 + \cos(\pi\beta)\right) & |\beta| < 1 \\ 0 & |\beta| \geq 1 \end{cases} \qquad \text{Hann window}
$$

$$
w_n(\beta) = \begin{cases} \frac{1}{2}\left(1 + \frac{9}{8}\cos(\pi\beta) - \frac{1}{8}\cos(3\pi\beta)\right) & |\beta| < 1 \\ 0 & |\beta| \geq 1 \end{cases} \qquad \text{“Flattened Hann” window}
$$

The periodically extended waveform is constructed by summing together an infinite number of copies of the wavelet $\hat{x}_{t_0}(t)$, all equally spaced by the period $\tau(t_0)$.

$$
x_{t_0}(t) \equiv \sum_{m=-\infty}^{\infty} \hat{x}_{t_0}\left(t - m\tau(t_0)\right) \tag{2}
$$

Clearly $x_{t_0}(t)$ is periodic with period $\tau(t_0)$. Using a complimentary (e.g. a Hann window or the kind of “flattened Hann” window suggested in [13]) and an appropriately time-scaled window of exactly one period half-amplitude length (the non-zero length is two periods long), one can see that the periodically extended waveform in Fig. 5 is continuous everywhere and matches the input exactly at time t_0, that is

$$
x_{t_0}(t_0) = \hat{x}_{t_0}(t_0) = x(t_0), \tag{3}
$$

and also very closely matches the input in the vicinity of t_0. This simple but very important fact has a way of phase-locking the periodically extended waveforms as t_0 is swept from the beginning of the note to the end. This is important to insure phase-locking of adjacent wavetables. If adjacent wavetables which have otherwise similar appearance, but are not phase-locked, are crossfaded from one to the next, an unintended null will occur when the crossfade is half complete.

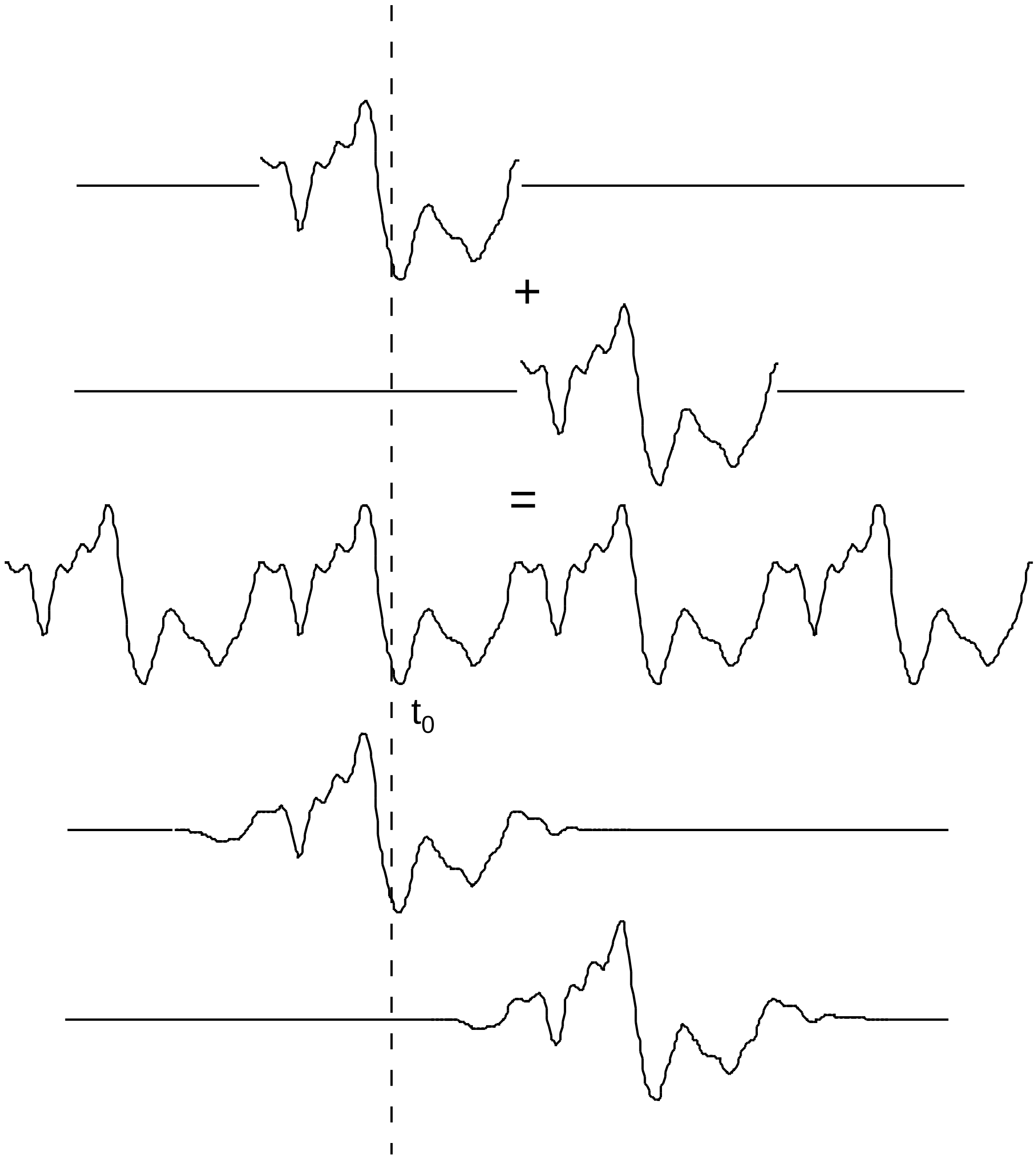

Figure 5 Period Extension of Quasi-Periodic Waveform
using Rectangular Window (top) and
"Flattened Hann" Window (bottom)

To rigorously of phase-align each wavetable might be more abstruse than it would first appear for quasi-periodic tones with nonstationary period or pitch. First the fundamental frequency $f_0(t_0)$ at time t_0 of the quasi-periodic tone is defined to be the reciprocal of the period $\tau(t_0)$ at the same time.

$$f_0(t_0) \equiv \frac{1}{\tau(t_0)}$$

If the wavetable phase at $t = 0$ is arbitrarily defined to be zero, then the (unwrapped) phase at time $t = t_0$ is

$$\theta(t_0) = 2\pi \int_0^{t_0} f_0(u)\,du$$

and $\theta(t_0)/2\pi$ is the precise number of cycles or periods that have elapsed from $t = 0$ to $t = t_0$. To back up in time to precisely zero phase would be to back up $\theta(t_0)/2\pi$ periods or $\left(\theta(t_0)/2\pi\right)\tau(t_0)$ in time.

Then in the final step, the wavetable (containing one cycle of the representative waveform of the input in the vicinity of t_0) is extracted from the periodic extension by resampling K points from $t = t_0 - \left(\theta(t_0)/2\pi\right)\tau$ to $t = t_0 - \left(\theta(t_0)/2\pi\right)\tau + \frac{K-1}{K}\tau$ using normal sampling (or resampling) techniques [5] [6].

$$\begin{aligned} x_{t_0}[k] &\equiv x_{t_0}\left(t_0 + \left(\frac{k}{K} - \int_0^{t_0} f_0(u)\,du\right)\tau\right) \qquad 0 \le k \le K-1 \\ &= \sum_{m=-\infty}^{\infty} \hat{x}_{t_0}\left(t_0 + \left(\frac{k}{K} - \int_0^{t_0} f_0(u)\,du\right)\tau - m\tau\right) \\ &= \sum_{m=-\infty}^{\infty} x\left(t_0 + \left(\frac{k}{K} - \int_0^{t_0} f_0(u)\,du - m\right)\right) w_n\left(\frac{k}{K} - \int_0^{t_0} f_0(u)\,du - m\right) \end{aligned}$$

There are only two terms of the summation in which the window $w_n()$ is nonzero.

$$-1 \le \frac{k}{K} - \int_0^{t_0} f_0(u)\,du - m < +1$$

$$\frac{k}{K} - \int_0^{t_0} f_0(u)du - 1 \; < \; m \; \le \; \frac{k}{K} - \int_0^{t_0} f_0(u)du + 1$$

$$m = \text{floor}\left(\frac{k}{K} - \int_0^{t_0} f_0(u)du\right), \; \text{floor}\left(\frac{k}{K} - \int_0^{t_0} f_0(u)du\right) + 1$$

$$x_{t_0}[k] = x\big(t_0 + \beta\tau\big)w_n(\beta) \; + \; x\big(t_0 + (\beta - 1)\tau\big)w_n(\beta - 1) \qquad (4)$$

where

$$\beta \equiv \frac{k}{K} - \int_0^{t_0} f_0(u)du - \text{floor}\left(\frac{k}{K} - \int_0^{t_0} f_0(u)du\right) = \text{fract}\left(\frac{k}{K} - \int_0^{t_0} f_0(u)du\right)$$

$$\text{fract}(u) \equiv u - \text{floor}(u) \; , \qquad 0 \le \text{fract}(u) < 1$$

The $\text{floor}()$ function maps the argument to the largest integer no greater than the argument. Eq. (4) defines explicitly how the discrete wavetable values are extracted from the input signal. Here the notation $x[k]$ is adopted for discrete functions of time (such as wavetable points) whereas $x(t)$ is the convention for continuous time functions. However, in this paper, the subscript t_0 of either $x_{t_0}[k]$ or $x_{t_0}(t)$ is a continuous value of time.

Perfect bandlimited reconstruction of the periodic extension, $x_{t_0}(t)$, from the extracted wavetable data, $x_{t_0}[k]$, (assuming K is even) would be accomplished from Eq. (5).

$$x_{t_0}\left(t + t_0 - \tau\int_0^{t_0} f_0(u)du\right) = \sum_{k=0}^{K-1} x_{t_0}[k] \frac{\sin\left(\frac{K\pi}{\tau}\left(t - \frac{k}{K}\tau\right)\right)}{K\tan\left(\frac{\pi}{\tau}\left(t - \frac{k}{K}\tau\right)\right)} \qquad (5)$$

However, it should be noted here that the reconstruction of $x_{t_0}(t)$ from the wavetable points should not, in practical resynthesis, require all K values of $x_{t_0}[k]$ to be summed as indicated above or any computational savings offered by using wavetable synthesis would be lost. A smaller number of the wavetable points (say, four to eight) in the neighborhood of $k = \left[\text{floor}\left(K\frac{t}{\tau}\right)\right] \text{mod}\, K$ would suffice if the form $\frac{\sin(K\pi\ )}{K\tan(\pi\ )}$ function in Eq. (5) were replaced by a properly windowed $\frac{\sin(K\pi\ )}{K\pi}$ or $\text{sinc}(K\ )$. If K is large enough (say 2048), linear interpolation may well suffice in lieu of the perfect reconstruction of Eq. (5).

The number of wavetable points necessary for proper sampling, K, is discussed in the following section. As t_0 moves slowly from beginning to end, a given point, $x_{t_0}[k]$, of the wavetable will change value slowly and will not change at all if the input is precisely periodic.

2 EQUIVALENCE TO HARMONIC ADDITIVE SYNTHESIS

A periodic function of time with period τ has, by definition, the following property:

$$x(t+\tau) = x(t), \qquad -\infty < t < \infty$$

This periodic function can also be written as a Fourier Series written as follows using any of the following three different forms.

$$\begin{aligned} x(t) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_0 t) - b_n \sin(2\pi n f_0 t) \\ &= c_0 + \sum_{n=1}^{\infty} r_n \cos(2\pi n f_0 t + \varphi_n) \\ &= \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_0 t} \end{aligned}$$

where

$$\begin{aligned} c_{-n} &= c_n \\ a_n &= 2\,\mathrm{Re}(c_n) \qquad b_n = 2\,\mathrm{Im}(c_n) \\ r_n &= 2|c_n| \qquad\quad \varphi_n = \arg(c_n) \end{aligned}$$

and

$$c_n = \frac{1}{\tau}\int_u^{u+\tau} x(t) e^{-j2\pi n f_0 t}\,dt \qquad f_0 \equiv \frac{1}{\tau}$$

The fourier coefficients, for a completely periodic function, are constant as above. For a quasi-periodic function, the periodicity property is only approximately true,

$$x\big(t+\tau(t)\big) \approx x(t)$$
$$(\,\tau(t) \text{ is the period around time } t\,)$$

and the fourier coefficients and period (or fundamental frequency) are not constant but slowly changing functions of time.

$$x(t) = c_0(t) + \sum_{n=1}^{\infty} r_n(t)\cos\left(2\pi n\int_0^t f_0(u)\,du + \varphi_n(t)\right) \tag{6}$$
$$= \sum_{n=-\infty}^{\infty} c_n(t)\, e^{\,j2\pi n\int_0^t f_0(u)\,du}$$

where
$$c_{-n}(t) = c_n(t)$$
$$r_n(t) = 2\left|c_n(t)\right| \qquad \varphi_n(t) = \arg\big(c_n(t)\big)$$
$$c_n(t) = \tfrac{1}{2}\, r_n(t)\, e^{j\varphi_n(t)} \qquad \text{for } n > 0$$

To obtain the values of the time variant but bandlimited Fourier “coefficients” at some time t_0, one would periodically extend the input $x(t)$ from t_0 outward as described in the previous section and then obtain fourier coefficients from that periodic extension. The periodic extension must hold the fourier coefficients constant at their values at time t_0, the fundamental frequency constant at $f_0(t_0)$, and must satisfy Eq. (3). The fourier coefficients in Eq. (8) are used to construct the periodic extension function of Eq. (2).

$$x_{t_0}(t) \equiv \sum_{m=-\infty}^{\infty} \hat{x}_{t_0}\big(t - m\tau(t_0)\big)$$
$$= c_0(t_0) + \sum_{n=1}^{\infty} r_n(t_0)\cos\left(2\pi n\left[\int_0^{t_0} f(u)\,du + \int_{t_0}^{t} f(t_0)\,du\right] + \varphi_n(t_0)\right)$$
$$= c_0(t_0) + \sum_{n=1}^{\infty} r_n(t_0)\cos\left(2\pi n f(t_0)\,t \;+\; 2\pi n\int_0^{t_0}\big(f(u) - f(t_0)\big)du + \varphi_n(t_0)\right) \tag{7}$$
$$= \sum_{n=-\infty}^{\infty} c_n(t_0)\, e^{\,j2\pi n\int_0^{t_0}\left(f(u)-f(t_0)\right)du}\, e^{\,j2\pi n f(t_0)\,t}$$

where

$$c_n(t_0)e^{j2\pi n\int_0^{t_0}(f(u)-f(t_0))du} = \frac{1}{\tau(t_0)}\int_u^{u+\tau(t_0)} x_{t_0}(t)e^{-j2\pi n\frac{t}{\tau(t_0)}}dt \qquad -\infty < u < \infty$$

$$= \frac{1}{\tau(t_0)}\int_0^{\tau(t_0)} x_{t_0}(t+t_0)e^{-j2\pi n\frac{t+t_0}{\tau(t_0)}}dt$$

$$= f_0(t_0)\int_0^{\tau(t_0)} x_{t_0}(t+t_0)e^{-j2\pi n f_0(t_0)(t+t_0)}dt$$

$$= f_0(t_0)e^{-j2\pi n f_0(t_0)t_0}\int_0^{\tau(t_0)} x_{t_0}(t+t_0)e^{-j2\pi n f_0(t_0)t}dt$$

$$= f_0(t_0)e^{-j2\pi n f_0(t_0)t_0}\int_0^{\tau(t_0)}\left[\hat{x}_{t_0}(t+t_0-\tau(t_0))+\hat{x}_{t_0}(t+t_0)\right]e^{-j2\pi n f_0(t_0)t}dt$$

$$c_n(t_0) = f_0(t_0)e^{-j2\pi n\int_0^{t_0} f(u)du}\left[\int_0^{\tau(t_0)} x(t+t_0-\tau(t_0))e^{-j2\pi n f_0(t_0)t}dt + \int_0^{\tau(t_0)}\left[x(t+t_0)-x(t+t_0-\tau(t_0))\right]w_n\left(\tfrac{t}{\tau(t_0)}\right)e^{-j2\pi n f_0(t_0)t}dt\right] \qquad (8)$$

The last equation is just to show that if $x(t)$ is truly periodic with period τ, then the second integral is zero and the fourier coefficients are no different from what we expect for a periodic $x(t)$. Also for quasi-periodic input, the fourier coefficients for a smoothly windowed wave extension differ from that of a rectangularly windowed extension by only the contribution of the second integral. The dependency of the fourier coefficients on time t_0 simply reflect that the periodic function on which they are derived depends on t_0. Given $x(t)$, and Eqs. (6) and (8), the amplitude and phase envelopes, $r_n(t)$ and $\varphi_n(t)$, can be determined for each harmonic. And given $\tau(t)$, determined from a pitch detection algorithm similar to Eq. (1) the instantaneous frequency $f_0(t)$ can be computed.

Musical Additive Synthesis amends Eq. (6) in a couple of ways. First, it eliminates the DC term and all terms above the Nth harmonic reflecting the bandlimited nature of the analyzed / synthesized tone.

$$x(t) = \sum_{n=1}^{N} r_n(t) \cos\left(2\pi n \int_0^t f_0(u)\,du + \varphi_n(t) \right) = \sum_{n=-N}^{N} c_n(t)\, e^{j2\pi n \int_0^t f_0(u)\,du} \tag{9}$$

where
$$\begin{aligned} c_{-n}(t) &= c_n(t) \\ r_n(t) &= 2|c_n(t)| \qquad \varphi_n(t) = \arg(c_n(t)) \\ c_n(t) &= \tfrac{1}{2} r_n(t) e^{j\varphi_n(t)} \qquad \text{for } n > 0 \\ c_0(t) &= 0 \end{aligned}$$

Secondly, the time variant phase term and the instantaneous fundamental frequency are collected together as a single nonstationary harmonic frequency expression below.

$$2\pi n \int_0^t f_0(u)\,du \ + \ \varphi_n(t) \ = \ 2\pi \int_0^t f_n(u)\,du \ + \ \varphi_n(0)$$

where
$$f_n(t) \equiv n f_0(t) + \tfrac{1}{2\pi} \varphi_n'(t)$$

$$x(t) = \sum_{n=1}^{N} r_n(t) \cos\left(2\pi \int_0^t f_n(u)\,du + \varphi_n(0) \right) \tag{10}$$

Eq. (10) can, for Additive Synthesis, be generalized further by removing any restriction that the instantaneous frequency of the nth harmonic or overtone, $f_n(t)$, **not** have to be close to n times the fundamental, however, the result (if that harmonic's amplitude was significant) would likely be less periodic or not quasi-periodic at all making this tone not suitable for normal wavetable synthesis. If Eq. (10) were to be converted back to the constant multiple fundamental frequency and time varying phase form as in Eq. (6), it would be apparent that the phase term, $\varphi_n(t)$, would have to vary rapidly to detune the overtone from its "harmonic" frequency $nf_0(t)$ making the fourier coefficients $c_n(t)$ no longer slowly varying in time. Thus we will restrict all overtones to be very nearly harmonic, that is with instantaneous frequencies nearly an integer multiple of some common fundamental frequency, $f_0(t)$.

If the tone is expressed in Additive Synthesis form, probably the simplest way to periodically extend the tone from time t_0 is to simply hold the slowly time-variant fundamental frequency and fourier coefficients constant at their values at time t_0 as is done in Eq. (7).

$$
\begin{aligned}
x_{t_0}(t) &= \sum_{n=1}^{N} r_n(t_0)\cos\left(2\pi n\left[\int_0^{t_0} f(u)du + \int_{t_0}^{t} f(t_0)du\right] + \varphi_n(t_0)\right) \\
&= \sum_{n=1}^{N} r_n(t_0)\cos\left(2\pi n f(t_0)t \;+\; 2\pi n\int_0^{t_0}\big(f(u) - f(t_0)\big)du + \varphi_n(t_0)\right) \\
&= \sum_{n=1}^{N} r_n(t_0)\cos\left(2\pi n f(t_0)\left(t - \int_0^{t_0}\left(1 - \tfrac{f(u)}{f(t_0)}\right)du\right) + \varphi_n(t_0)\right)
\end{aligned}
$$

Then the wavetable points extracted out of the periodic extension from time t_0 of input $x(t)$ expressed as Eq. (9) are

$$
\begin{aligned}
x_{t_0}[k] &\equiv x_{t_0}\left(t_0 + \left(\frac{k}{K} - \int_0^{t_0} f_0(u)du\right)\tau(t_0)\right) \qquad 0 \le k \le K-1 \\
&= x_{t_0}\left(\left(\frac{k}{K} - \int_0^{t_0}\big(f(u) - f(t_0)\big)du\right)\tau(t_0)\right) \\
&= \sum_{n=1}^{N} r_n(t_0)\cos\left(2\pi\tfrac{nk}{K} + \varphi_n(t_0)\right) \qquad (11) \\
&= \sum_{n=-N}^{N} c_n(t_0)e^{j2\pi\frac{nk}{K}}
\end{aligned}
$$

Eq. (11) is a fundamental mapping of additive synthesis specification data (usually expressed as each harmonic's amplitude $r_n(t)$ and frequency $f_n(t)$ **or** phase $\varphi_n(t)$) to wavetable synthesis data $x_t[k]$. In order for the Nyquist sampling theorem requirement to be satisfied, the number of wavetable points K must be larger than twice the index of the highest harmonic or $2N$. This means, at least theoretically, a 128 point wavetable can accurately represent a periodic function having arbitrarily defined magnitude and phase for each harmonic up to the 63rd. In practice, for the purpose of reducing spurious aliasing during interpolation, it might seem wise to limit the index of the highest nonzero harmonic (N) to something much less than $\frac{K}{2}$. For purposes of both simple modulo address arithmetic and convenient radix-2 DFT or FFT, it seems logical to choose K to be an integer power of two.

Another little tidbit to point out here is, in the limiting case, Eq. (11) maps $2N$ continuous and bandlimited time functions (N amplitude functions, $r_n(t)$, and N phase functions, $\varphi_n(t)$) governing frequency domain behavior into nearly the same number ($K = 2N + 2$) of continuous and bandlimited time functions, $x_t[k]$, governing time domain behavior. The difference of enumerating two functions is due to the implicit specification that the DC and Nyquist harmonics have zero amplitude adding two more functions governing frequency domain behavior.

3 ANY WAY YOU SLICE IT

Eq. (11) can be expressed in such a way that the wavetable values $x_t[k]$ at some time t are the inverse Discrete Fourier Transform of some data set $X_t[n]$.

$$\begin{aligned} x_t[k] &= \sum_{n=1}^{N} r_n(t)\cos\left(2\pi\tfrac{nk}{K} + \varphi_n(t)\right) \\ &= \sum_{n=-N}^{N} c_n(t)\, e^{j2\pi\frac{nk}{K}} \\ &= \frac{1}{K}\sum_{n=0}^{K-1} X_t[n]\, e^{j2\pi\frac{nk}{K}} \qquad \begin{matrix} 0 \le k \le K-1 \\ N < \tfrac{K}{2} \end{matrix} \end{aligned} \tag{12}$$

where

$$X_t[n] = \begin{cases} Kc_n(t) = \tfrac{K}{2} r_n(t) e^{j\varphi_n(t)} & \text{for } 1 \le n \le N \\ Kc_{n-K}(t) = \tfrac{K}{2} r_{K-n}(t) e^{-j\varphi_{K-n}(t)} & \text{for } K-N \le n \le K-1 \\ 0 & \text{otherwise} \end{cases}$$

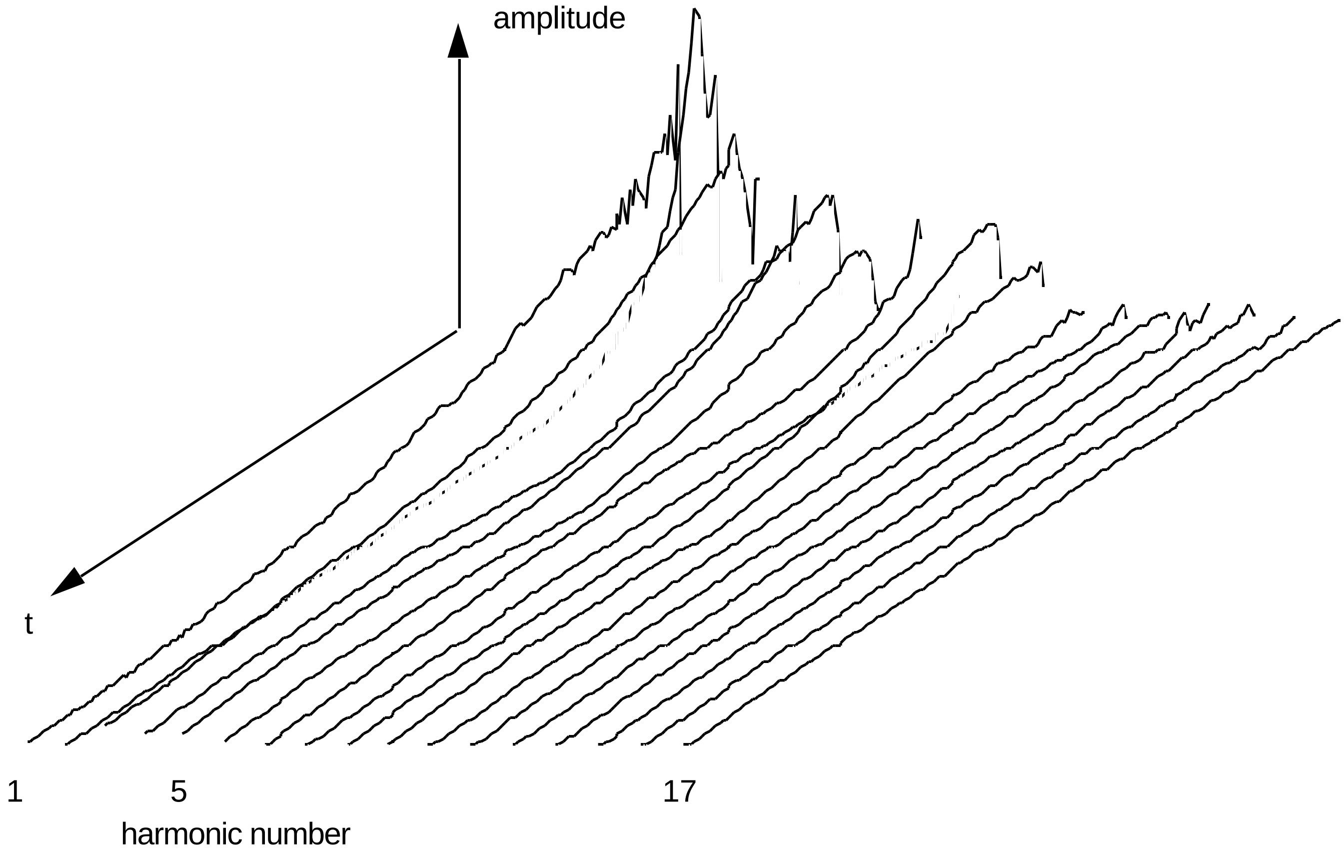

Figure 6   Family of Harmonic Amplitude Magnitude Curves

The use of the inverse DFT representation in Eq. (12) has one immediately useful consequence of yielding an inverse mapping to Eq. (11):

$$
\begin{aligned}
r_n(t) &= \tfrac{2}{K}\,|\,X_t[n]\,| = \tfrac{2}{K}\sqrt{X_t[n]X_t[K-n]} \\
\varphi_n(t) &= \arg\left(X_t[n]\right) = \tfrac{1}{2j}\log\left(\frac{X_t[n]}{X_t[K-n]}\right)
\end{aligned}
\tag{13}
$$

where $$X_t[n] = \sum_{k=0}^{K-1} x_t[k]\,e^{-j2\pi\frac{nk}{K}}$$

This tells us directly, given a set of wavetable data $x_t[k]$ at a given time, what the instantaneous magnitude $r_n(t)$ and phase $\varphi_n(t)$ (or frequency, if we so choose) of each harmonic is. This will be put to use in the next section.

It is common to plot the family of curves for all of the harmonic amplitudes $r_n(t)$ as shown in Fig. 6. Plots of harmonic phases $\varphi_n(t)$ can also be drawn but it is less common to do so. What might be illustrative would be plotting the complex harmonic amplitude function $X_t[n]$ versus time as a three dimensional trajectory as shown in Fig. 7 for one harmonic. At some time t the distance $X_t[n]$ is from the zero line is $\frac{K}{2} r_n(t)$ and the angle against the real plane is $\varphi_n(t)$ if $1 \le n \le N$. Another useful property is that at any time t, all of the complex amplitude curves $X_t[n]$ are (varied) linear combinations of the wavetable curves $x_t[k]$. This fact will also find service in the next section.

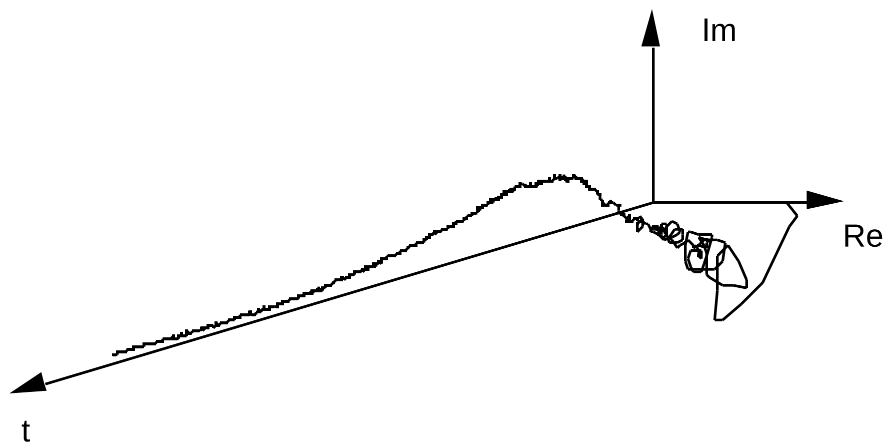

Figure 7 Complex Harmonic Amplitude (1st Harmonic)

By holding n constant and “slicing” the curve family in Fig. 6 along $r_n(t)$, one can readily observe in that cross-section, the envelope of the magnitude of the nth harmonic of the tone versus time. On the other hand, by holding t constant and “slicing” the family of curves perpendicular to the t axis, one can observe in that cross-section, the instantaneous line spectrum magnitude of the quasi-periodic tone at time t.

In the same manner of plotting the harmonic amplitude family of envelopes in Fig. 6, we can plot the family of envelopes of the time-variant wavetable points $x_t[k]$ versus time as shown in Fig. 8. Again, by holding k constant and “slicing” the curve family in Fig. 8 along $x_t[k]$, one can readily observe in that cross-section, the envelope of the value of the kth wavetable point versus time. And again, by holding t constant and “slicing” the family of curves perpendicular to the t axis, one can observe in that cross-section, the instantaneous, phase aligned, waveshape of the quasi-periodic tone at time t .

It should be restated here that, at a given time t the line spectrum cross-section of Fig. 6 and the waveshape cross-section of Fig. 8 are related to each other by the Discrete Fourier Transform and its inverse.

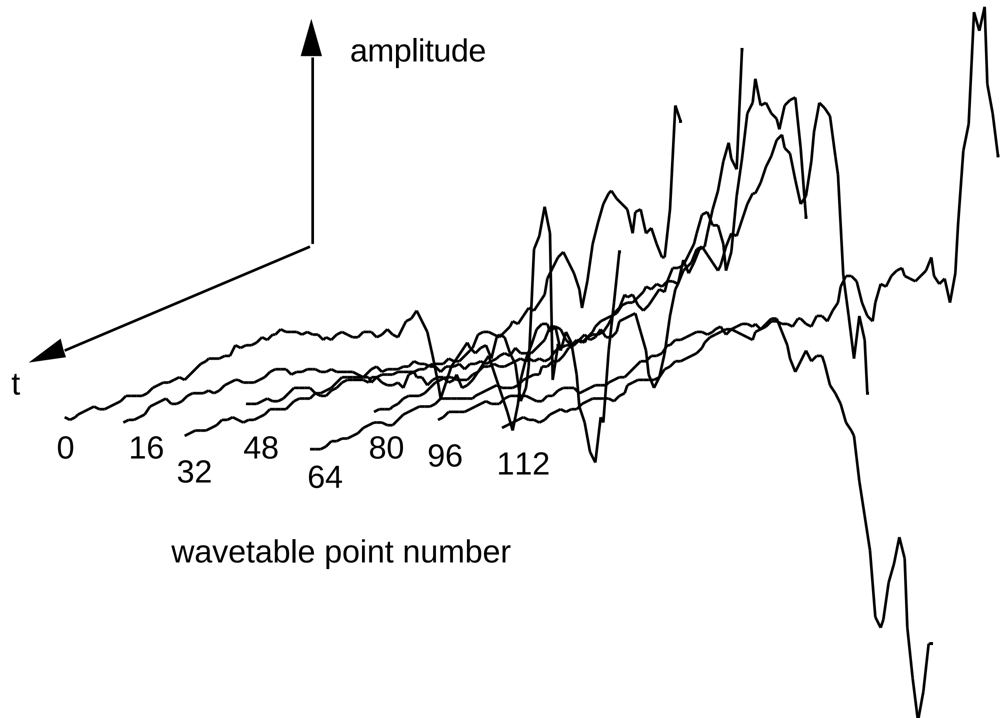

Figure 8 Family of Envelopes of Wavetable Points

4 DATA REDUCTION, APPROXIMATION, AND ERROR

It has been shown above that, given a quasi-periodic piece of sound and the means to measure pitch or period at any given neighborhood of time, the characteristic waveform at that neighborhood of time can be extracted. These neighborhoods could be spaced very close in time (say 1 ms) to each other resulting in wavetables that are very similar to their adjacent wavetables and, for real-time computation, this would have no additional cost since, still, only two wavetables are being crossfaded at any one time. There would be, however, a cost to expect in memory requirements: If 128 point wavetables were extracted every millisecond (44 or 48 samples), we would be increasing memory requirements at least 167% from the raw recorded PCM sample of the note. This does not take any advantage of the redundancy resulting from the near periodicity of the tone. The underlying reason for this redundancy is that information is (or, more precisely data storage resources are) wasted on many wavetables that are so nearly equal to their neighbors that some wavetables can and should be eliminated. A frequency domain perspective of this can be seen by performing a Fourier Transform on Eq. (6) (assume, for illustrative purposes that the fundamental frequency $f_0(t)$ is constant).

$$x(t) = \sum_{n=-\infty}^{\infty} c_n(t) e^{j2\pi n f_0 t}$$

$$X(f) \equiv \mathscr{F}\{x(t)\} = \mathscr{F}\left\{ \sum_{n=-\infty}^{\infty} c_n(t) e^{j2\pi n f_0 t} \right\} = \sum_{n=-\infty}^{\infty} C_n(f - nf_0)$$

$$C_n(f) \equiv \mathscr{F}\{c_n(t)\}$$

If $x(t)$ is nearly periodic, meaning that $c_n(t)$ and each wavetable point is very slowly moving in time and that each $C_n(f)$ is bandlimited to probably much less than f_0, then there is a lot of empty space, between the harmonic "spikes" $C_n(f)$, in the spectrum $X(f)$ that need not have any information (or data storage) wasted on it. If $x(t)$ was not at all periodic, adjacent wavetables would not be similar (no redundancy) and the spectrum $X(f)$ would have nonzero data throughout its frequency range.

Returning to the proposition of extracting a wavetable every millisecond (or some other small unit of time), it is clear that this essentially samples the envelope functions $x_t[k]$ every millisecond and consequently the frequency domain envelopes, $X_t[n]$, $c_n(t)$, $r_n(t)$, and $\varphi_n(t)$ at the same rate. If the input $x(t)$ is sufficiently quasi-periodic, all of these envelopes should be bandlimited to far less than what would required for this sampling rate (in this case 500 Hz) hence more information is present than necessary.

One can readily imagine many different approaches to reducing the amount of this information. However proposing an optimal method, or even dealing with any more approaches than the very simplest is beyond the scope of this paper. One can refer to Horner, et al. [3] [4] [14], Serra, et al. [15], and Sandell and Martens [16], for some novel work in this. Nonetheless, there are a few common considerations regarding *any* method of data representation and reduction for wavetable synthesis of quasi-periodic tones that are observed here, perhaps in the hope that meaningful and common goals and metrics of performance can be used for all methods. The simplistic method of data reduction given here as an example is that of simply eliminating adjacent wavetables (that were extracted, in the beginning, far more often than necessary), keeping a few for "endpoints", until the results become nearly discernibly different from the original tone (assuming linearly crossfading between the endpoint wavetables that remain).

What makes the result of any data reduction operation "discernible" or "not quite discernible" is a perceptual issue (or specifically a psychoacoustical issue if it is audio data). All that's involved with that broad issue is also beyond the scope of this paper yet a very general and hopefully perceptually relevant error metric is proposed here:

> Given a target quasi-periodic tone $x(t)$ that can be represented in the Additive Synthesis form of Eqs. (9) and (10) and the data reduced synthesized tone $\tilde{x}(t)$ that is represented in an identical way in Eqs. (9) and (10),
>
> - The amplitude of each synthesized harmonic must be constrained to deviate less than a given and possibly time-variant error bound in dB from the target:
>
> $$\left| \log\left(\tilde{r}_n(t)\right) - \log\left(r_n(t)\right) \right| < \varepsilon_n(t) \qquad (14)$$
>
> - Although absolute phase of each harmonic may not be perceptually salient, the instantaneous frequency (and thus the rate of change of phase) is salient (the importance of this seems to have been devalued in Horner, et al. [3] [4] and Serra, et al. [15]) and must also be constrained to deviate less than a given and possibly time-variant error bound in Hz from the target:
>
> $$2\pi\left| \tilde{f}_n(t) - f_n(t) \right| = \left| \tilde{\varphi}'_n(t) - \varphi'_n(t) \right| < \xi_n(t) \qquad (15)$$

The equality in Eq. (15) can be supported with Eq. (10) and by noting that the given input (or given target tone described by Eqs. (9) or (10)) refer to the same instantaneous fundamental frequency $f_0(t)$. If an overall error metric is desired (usually for the purpose of minimizing, given a fixed number of wavetables), summing (for $1 \le n \le N$) the difference expressions in Eqs. (14) and (15), weighted by the reciprocals of $\varepsilon_n(t)$ or $\xi_n(t)$ (possibly raised to a high power to de-emphasize any error below the constraints), might be useful but is not used here.

The error constraints of Eqs. (14) and (15) are more easily visualized by referring to the complex harmonic amplitude trajectories shown in Figs. 10 and 12 and noting that

$$r(t)e^{-\varepsilon_n(t)} \;<\; \tilde{r}(t) \;<\; r(t)e^{+\varepsilon_n(t)}$$

$$|X_t[n]|\, e^{-\varepsilon_n(t)} \;<\; \left|\tilde{X}_t[n]\right| \;<\; |X_t[n]|\, e^{+\varepsilon_n(t)} \tag{14a}$$

and

$$\varphi_n'(t) - \xi_n(t) \;<\; \tilde{\varphi}_n'(t) \;<\; \varphi_n'(t) + \xi_n(t)$$

$$\frac{\varphi_n(t) - \varphi_n(t_i)}{t - t_i} - \xi_n(t_i) \;<\; \frac{\tilde{\varphi}_n(t) - \tilde{\varphi}_n(t_i)}{t - t_i} \;<\; \frac{\varphi_n(t) - \varphi_n(t_i)}{t - t_i} + \xi_n(t_i)$$

$$\varphi_n(t) + \left[\tilde{\varphi}_n(t_i) - \varphi_n(t_i)\right] - \xi_n(t_i)(t - t_i)$$
$$< \; \tilde{\varphi}_n(t) \;<\; \varphi_n(t) + \left[\tilde{\varphi}_n(t_i) - \varphi_n(t_i)\right] + \xi_n(t_i)(t - t_i)$$

$$\arg(X_t[n]) + \left[\arg\left(\tilde{X}_{t_i}[n]\right) - \arg\left(X_{t_i}[n]\right)\right] - \xi_n(t_i)(t - t_i)$$
$$< \; \arg\left(\tilde{X}_t[n]\right) \tag{15a}$$
$$< \; \arg(X_t[n]) + \left[\arg\left(\tilde{X}_{t_i}[n]\right) - \arg\left(X_{t_i}[n]\right)\right] + \xi_n(t_i)(t - t_i)$$

for small $t - t_i$.

$\tilde{x}(t)$ is the synthesized output from a reduced data set and can be defined from some hypothetical harmonic amplitude and phase or frequency envelopes, $\tilde{r}_n(t)$, $\tilde{\varphi}_n(t)$, $\tilde{f}_n(t)$, using Eqs. (9) or (10). If $\tilde{x}(t)$ is computed by linearly crossfading between two "endpoint" wavetables of the original input $x(t)$ (at times t_i and t_{i+1}), then the envelope functions for the synthesized wavetable points at times between t_i and t_{i+1} are these linear functions:

$$\tilde{x}_t[k] = x_{t_i}[k] + \frac{t - t_i}{t_{i+1} - t_i}\left(x_{t_{i+1}}[k] - x_{t_i}[k]\right) \qquad t_i \le t \le t_{i+1} \tag{16}$$

Because of the linear mapping of the DFT in Eq. (13) is it clear that the same applies to the complex harmonic envelopes:

$$\tilde{X}_t[n] = X_{t_i}[n] + \frac{t - t_i}{t_{i+1} - t_i}\left(X_{t_{i+1}}[n] - X_{t_i}[n]\right) \qquad t_i \le t \le t_{i+1} \tag{17}$$

$$\tilde{c}_n(t) = \tfrac{1}{2}\,\tilde{r}_n(t)e^{j\tilde{\varphi}_n(t)} = \tfrac{1}{K}\,\tilde{X}_t[n] \qquad \text{for } 1 \le n \le N$$

or

$$= c_n(t_i) + \frac{t - t_i}{t_{i+1} - t_i}\left(c_n(t_{i+1}) - c_n(t_i)\right) \qquad t_i \le t \le t_{i+1}$$

The complex harmonic envelopes $\tilde{X}_t[n]$ or $\tilde{c}_n(t)$ of the synthesized tone are then connected straight lines as shown in Fig. 9.

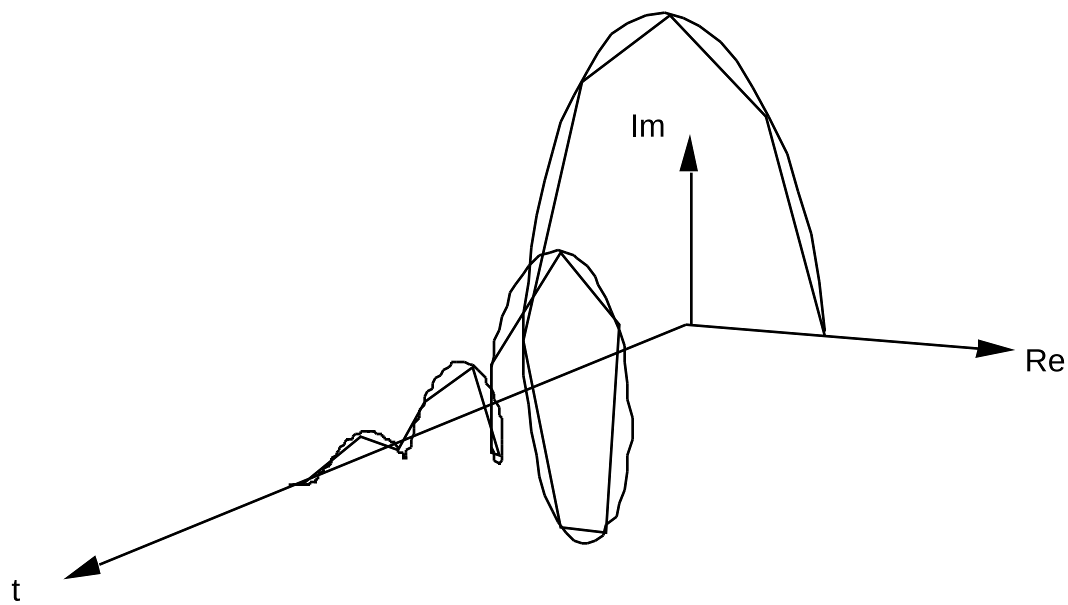

Figure 9 Complex Harmonic Envelope and Piecewise Linear Approximation

For a fixed k, it would seem unproductive to attempt to fit the piecewise linear curve $\tilde{x}_t[k]$ directly to the given curve $x_t[k]$ since there is no direct perceptual coupling of the exact value of $\tilde{x}_t[k]$ and how we hear $\tilde{x}(t)$. But that is not so much the case with the complex harmonic envelope curves $\tilde{c}_n(t)$ or $\tilde{X}_t[n]$ since they are directly related to the amplitude and phase (or frequency) of the nth harmonic of $\tilde{x}(t)$. Therefore fitting piecewise linear curves $\tilde{c}_n(t)$ to the $c_n(t)$ envelopes originally derived from $x(t)$ is the strategy taken here.

Now from a given quasi-periodic input $x(t)$, the complex harmonic envelope functions $X_t[n]$ are readily determined as shown in the previous sections and if the breakpoint times (which correspond with the "endpoint" wavetable times t_i) are known, the complex harmonic envelope $\tilde{X}_t[n]$ and the salient properties ($\tilde{r}_n(t)$, $\tilde{\varphi}_n(t)$, $\tilde{f}_n(t)$) of it for the synthesized output are known. If the simple data reduction method of eliminating redundant adjacent wavetables mentioned above is used, the following procedure for determining "endpoint" wavetable times t_i could be performed:

1. Start with $t = 0$, $i = 0$ and $t_i = t_0 = 0$.
2. Advance t by a small increment (such as 1 ms as suggested above) and evaluate both Eqs. (14a) and (15a) for every harmonic index $1 \leq n \leq N$.
3. Check to see if violation of either inequality (14a) and (15a) for any harmonic index is impending for any time between t_i and t. If no, repeat step 2. If yes, proceed to step 4.
4. Let $t_{i+1} = t$, $i \leftarrow i+1$, repeat procedure at step 2.

5 INTERPOLATION IN MULTIPLE DIMENSIONS

Because wavetable synthesis inherently "normalizes" frequency in its analysis (that is each wavetable represents the data for exactly one period of a waveform), it is not difficult to interpolate between two or more waveforms *for different notes*. Interpolating between notes of different instruments is not expected to be useful. However interpolating between different but relatively close pitches of the same instrument and interpolating between notes of the same pitch with varying loudness, attack, or key velocity has had some limited but interesting results.

The assumption made here, with no certain physical basis, is that it makes sense that the wavetables of the two or more notes being interpolated be phase aligned as much as possible. Since adjacent wavetables of the same note are already phase aligned to each other (because of the phase locking procedure of Section 1), what remains necessary is to phase align **all** of one note's wavetable against another given note's wavetables with a **single** phase adjustment or wavetable rotation. Rotating all wavetables of a given note preserves the relative phase coherency between adjacent wavetables of that note.

Probably the simplest criterion to phase align one note's wavetables, $y_t[k]$, against another's, $x_t[k]$, is to maximize the time average cross-correlation of corresponding wavetables extracted at the same times in the evolution of the note.

$$\max_{0 \le m < K} \left\{ \sum_i \sum_{k=0}^{K-1} y_{t_i}\left[(k+m) \bmod K\right] x_{t_i}[k] \right\}$$

$$\text{then redefine :} \quad y_{t_i}[k] \leftarrow y_{t_i}\left[(k+m) \bmod K\right] \tag{18}$$

If wavetables are removed for data reduction as done in the previous section, this rotate-and-align operation should be done first so that there exists, for each wavetable of a given note at times t_i, a corresponding wavetable of the other note(s) at the same times. K should be quite large (at least 2048) for the rotation of Eq. (18) to have sufficient resolution.

The methods of weighting and mixing different notes for interpolation is far from a settled issue but a first guess could be simple polynomial interpolation between the corresponding points of the different but corresponding wavetables. That is, linear interpolation would be used to change from one note definition to another, given only two notes. Quadratic interpolation would be used if three notes are used and so on.

Consider this example: given three recordings of the same pitch of the same instrument but at different and increasing intensities or key velocities, mf, f, and ff. Associate with these three loudness levels the control parameters, 0, 1, and 2, respectively (the parameters do not need to be equally spaced and normally would not be). At time t_0, the three notes have corresponding, phase aligned, wavetables, $x_{mf,t_0}[k]$, $x_{f,t_0}[k]$, $x_{ff,t_0}[k]$ and given a loudness control parameter, α, the resulting wavetable would be defined as:

$$\begin{aligned} x_{t_0}[k] &= \frac{(\alpha-1)(\alpha-2)}{(0-1)(0-2)} x_{mf,t_0}[k] + \frac{(\alpha-0)(\alpha-2)}{(1-0)(1-2)} x_{f,t_0}[k] + \frac{(\alpha-0)(\alpha-1)}{(2-0)(2-1)} x_{ff,t_0}[k] \\ &= y_{0,t_0}[k] + \alpha \; y_{1,t_0}[k] + \alpha^2 \; y_{2,t_0}[k] \end{aligned} \tag{19}$$

Eq. (19) uses Lagrange's interpolation formula [17] and normally α is approximately in the range of the given control parameters ($0 \le \alpha \le 2$). Eq. (19) also shows two different implementations, the lower equation probably being simpler to perform in real-time but requiring the $y_{\sim,t_0}[k]$ wavetables to be precomputed from the $x_{\sim,t_0}[k]$ wavetables. This method can also be applied in a similar manner, with varying results, to three notes at different pitches of the same instrument.

6 CONCLUSIONS

Recapitulating, wavetable synthesis data (the wavetables themselves) can be extracted directly from the time-domain waveform itself or computed from the additive synthesis representation (or some other synthesis model) of a quasi-periodic musical tone. In either case, adjacent wavetables, in time or other parametric dimensions, should normally be phase aligned as much as possible so that in mixing or crossfading, spurious nulls are avoided. At any given small neighborhood of time, the wavetable data and the additive synthesis data are related to each other by way of the Discrete Fourier Transform. As in additive synthesis, the salient perceptual features of the waveform to preserve, when any data reduction is performed, are the instantaneous amplitude and frequencies (to varying degrees of accuracy) of each harmonic. As noted in the introductory section, wavetable synthesis seems to be a natural and comparatively inexpensive alternative to additive synthesis of quasi-periodic tones.

7 ACKNOWLEDGMENTS

Thanks go to Scott Morgan of Interval Music Systems for supplying the author with some nice instrument sound files for analysis, for early discussions regarding this topic, and for providing a project related to the topic (the Prophet VS WaveWrangler software) for the author to contribute to.

8 REFERENCES

[1] F. R. Moore, *Elements of Computer Music*, pp. 159-169, (Prentice-Hall, Englewood Cliffs, NJ, 1990).

[2] J. M. Snell, “Design of a Digital Oscillator That Will Generate up to 256 Low-Distortion Sine Waves in Real Time”, *Foundations of Computer Music*, pp. 289-325, (MIT Press, Cambridge, MA, 1985).

[3] A. Horner, J. Beauchamp, and L. Haken, “Methods for Multiple Wavetable Synthesis of Musical Instrument Tones”, *J. Audio Eng. Soc.*, vol. 41, no. 5, pp. 336-356 (1993 May).

[4] A. Horner, “Wavetable Matching Synthesis of Dynamic Instruments with Genetic Algorithms”, *J. Audio Eng. Soc.*, vol. 43, no. 11, pp. 916-931 (1995 Nov.).

[5] R. W. Schafer and L. R. Rabiner, “A Digital Signal Processing Approach to Interpolation”, *Proc. IEEE,* vol. 61, no. 6, pp. 692-702, (1973 June).

[6] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, “Splitting the Unit Delay: Tools for Fractional Delay Filter Design”, *IEEE Signal Process. Mag.*, vol.13., no. 1, pp. 30-60 (1996 Jan.).

[7] C. Roads, *The Computer Music Tutorial,* p. 159, (MIT Press, Cambridge, MA, 1996).

[8] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals,* pp. 149-150, (Prentice-Hall, Englewood Cliffs, NJ, 1978).

[9] J. D. Wise, J. R. Caprio, and T. W. Parks, “Maximum Likelihood Pitch Estimation”, *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 5, pp. 418-422, (1976 Oct.).

[10] Y. Medan, E. Yair, and D. Chazan, “Super Resolution Pitch Determination of Speech Signals”, *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 40-48, (1991 Jan.).

[11] P. R. Cook, D. Morrill, and J. O. Smith, “A MIDI Control and Performance System for Brass Instruments”, *ICMC Proceedings 1993,* pp. 130-133.

[12] P. R. Cook and J. O. Smith, “Accurate Pitch Measurement and Tracking System and Method”, *U. S. Patent No. 5,353,372,* (1995).

[13] R. Bristow-Johnson, “A Detailed Analysis of a Time-Domain Formant-Corrected Pitch Shifting Algorithm”, *J. Audio Eng. Soc.*, vol. 43, no. 5, pp. 340-352 (1995 May).

[14] A. Horner and J. Beauchamp, “Piecewise-Linear Approximation of Additive Synthesis Envelopes, A Comparison of Various Methods”, *Computer Music J.,* vol. 20, no. 2, (1996 Summer).

[15] M. H. Serra, D. Rubine, and R. B. Dannenburg, “Analysis and Synthesis of Tones by Spectral Interpolation”, *J. Audio Eng. Soc.*, vol. 38, no. 3, pp. 111-128 (1995 Mar.).

[16] G. J. Sandell and W. L. Martens, “Perceptual Evaluation of Principal-Component-Based Synthesis of Musical Timbres”, *J. Audio Eng. Soc.*, vol. 43, no. 12, pp. 1013-1028 (1995 Dec.).

[17] E. Kreyszig, *Advanced Engineering Mathematics,* p. 652, (Wiley, New York, NY, 1972).