TABLE LOOKUP OSCILLATORS USING GENERIC INTEGRATED WAVETABLES

Günter Geiger

Music Technology Group
Universitat Pompeu Fabra, Barcelona, Spain
ggeiger@iua.upf.edu

ABSTRACT

Table lookup oscillators form a basic building block of most software synthesis systems. Several of the classical digital and analog synthesis techniques require their use. The classical table lookup oscillator [1, 2] is commonly discussed regarding its amplitude error, but another source of error is mostly disregarded, although it influences the sound quality to a much higher degree, especially when running the oscillators at elevated frequencies. This error is due to the aliasing effect that occurs when shifting (or resampling) the original lookup table at higher rates.

One obstacle that has been encountered when trying to emulate classical analog synthesizers, is that the naive implementation of waveforms for subtractive synthesis generate the same aliasing frequencies. Over the years, several solutions for the problem of generating high quality band-limited waveforms for emulating classical analog synthesis have been proposed, solving the problem for specific waveforms. Never-the-less, the general question of how to manage aliasing behavior efficiently in general table lookup oscillators is still not resolved. This article reviews existing techniques for analog synthesis and tries to expand them for the general case.

1. INTRODUCTION

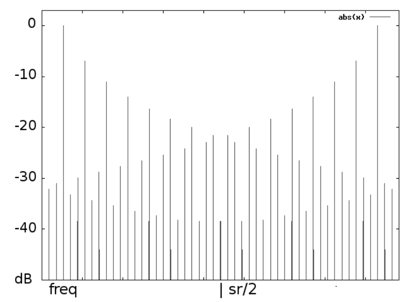
A general table lookup oscillator consists of two elements, the table, which specifies a sampled version of the oscillator function at a specific frequency (f_s/N, where N is the table size) and an interpolation scheme, which is used to resample the stored function in a way that it has a new duration $1/f$, the period of the desired oscillator. If N is the number of samples in the table, f_s the systems sampling frequency, f the oscillator frequency, a interpolation scheme I can be described in the following form.

$$x[n] = I\left(\frac{nf}{f_s N}\right) \tag{1}$$

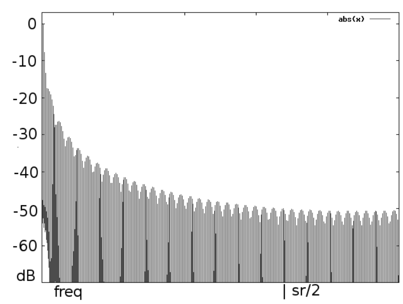
The interpolation scheme is commonly a lowpass filter that acts on several samples of the lookup table. Interpolating the table values this way leads to a reduction of the amplitude error, but because of the fact that samples of the original table get actually skipped, it acts only as a post-resampling low-pass filter, and therefore it doesn't improve the signal to noise ratio of "real" against aliased frequency components.

Implementing a table lookup oscillator leads to aliasing effects, even if the original signal in the table does not show aliasing. This is noticeable when operating the oscillator at higher frequencies, as for a naive implementation frequencies beyond the Nyquist frequency get mirrored around $f_s/2$ and reappear as aliasing frequencies.

Figure 1 shows a spectrum of such an aliased waveform. Our goal is to devise an interpolation scheme that is efficient and general and allows for the implementation of general wave-table lookup oscillators with minimal aliasing effects.



(a)



(b)

Figure 1: *(a) Aliasing through resampling of a wave-table. Partials are mirrored at half the sampling frequency and disturb the original signal. (b) Spectral content of a sawtooth at frequency* $44100/1024$*, the amplitude at half sampler-ate is about* 54 *dB lower.*

Several algorithms are known to reduce aliasing effects, mainly targeted at the recreation of analog synthesizer waveforms such as square and sawtooth. We try to give a short overview of these techniques.

- Oversampling
- Precalculation of band limited signals using resampling and table interpolation.
- Synthesis based on band limited impulse trains (BLIT) [3]
- band limited step (BLEP), based on the idea of BLIT [4]
- Differentiated Parabolic Wave (DPW) [5]
- full-wave rectified sine waves [6]

1.1. Oversampling

The method used when oversampling is straight forward, best results give sinc based FIR anti-aliasing filters which are used for the resampling step. This method is fairly expensive because of the high oversampling factors but it can be generally applied with high quality.

Prefiltering the lookup table in order to be able to shift to higher frequencies is a special case of the oversampling method. In this case the oversampling factor is determined by the length of the table and by the target frequency. The wave-table can be seen as such an oversampled representation of the function. Low-pass filtering the whole table at the desired frequency (kf_s/N) where $k = f_{\text{target}}N/f_s$ is the pitch shift factor and resampling [7] the resulting table in order to achieve a higher frequency leads to a high quality implementation. N samples have to be filtered, regardless of the resulting period which could be just a few samples for high frequencies. Therefore the method gets more expensive the higher the frequency. Oversampling lookup tables this way yields oscillators of constant high quality, but the quality has to be paid with efficiency losses at higher frequencies.

1.2. Precalculating band-limited wave-tables

For the general case of anti-aliased waveforms, this method can either be implemented by analytically computing the waveforms at different frequencies, by constructing their Fourier series with varying bandwidth, or by using the oversampling method as a precalculation step. It helps to decrease the performance problems of the oversampling method but increases memory consumption considerably [8].

1.3. Band Limited Impulse Trains

This method achieves to synthesize some of the analog waveforms by the interpolation of a band-limited Impulse Train, computed by the following formula:

$$x[n] = \frac{\sin(2\pi f/f_s)}{f/f_s} \tag{2}$$

The BLIT method is effective and can be computed without additional memory, but being an analytic method it is only applicable for specific waveforms.

1.4. The minBLEP method

An enhancement of the BLIT method is the band limited step function. In this method the discontinuity in the analog waveforms such as the sawtooth gets precalculated as such and stored in a wave-table. The advantage is a higher stability of the resulting waveform, the implementations efficiency is dependent on the frequency that has to be synthesized.

1.5. Differentiated parabolic wave

This method does not try to synthesize a perfect band-limited signal, but it allows some aliasing to occur, and minimizes it in those regions where it is most likely to be noticed by a listener. The method describes the synthesis of a sawtooth by integration of a parabolic wave, but it can be expanded to the general case. The method, being analytic can be expanded to the general numeric case as we will see later.

1.6. Full-wave rectified sine wave

Lane [6] proposes an efficient implementation of the sawtooth waveform based on full-wave rectifying a sine wave and filtering the resulting signal with two digital filters, one as a 1 pole filter (with a zero at DC) and a higher order butterworth low-pass. The goal of the method is reduction of aliasing and not its complete suppression. The methods proposed are very much focused on generating specific waveforms and are hard to apply to the general case.

2. THE INTEGRATED WAVE-TABLE

2.1. DPW analysis

We might take a closer look at the differentiated parabolic wave method. Besides the relatively expansive oversampling method, the DPW method is a candidate to be expanded for general waveforms, as its idea does not rely on dealing directly with the properties of specific waveforms, but on the low-pass characteristics of the integration.

First we will try to take a closer look at the DPW method in order to understand how it can be applied in the general case. We start from a sawtooth wave, described by $x[n] = nf/f_s \mod 1$, from Fourier analysis we know that this waveform has harmonics which decay at a ratio of $1/n$, which means 6dB per octave. To calculate the amplitude of the nearest aliased bin we use the ad-hoc formula

$$A_a(f_0) = 20 * \log\left(\frac{f}{f_s - f_0}\right) \tag{3}$$

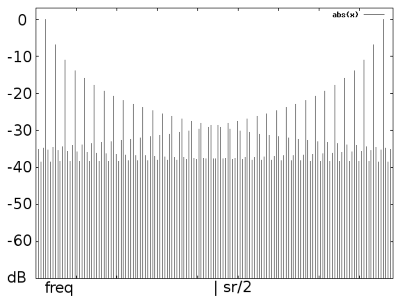
with f_0 being the fundamental frequency of our generated sawtooth wave, A_a the amplitude in dB of a aliased bin at f_0. We assume the closest aliased bin to appear in the first octave above f_0. For example, assuming a frequency f of 1323 Hz, and considering the mirror effect of aliasing at 22050 Hz the closest aliased bin appears at 1764 Hz with an RMS value 30 dB below the fundamental. Depending on the frequency of the fundamental these aliasing frequencies are either directly audible or produce roughness in the signal through interference. When sweeping the frequency of the oscillator one can hear the additional disturbing effect that the aliased components are moving in the opposite direction of the sweep.

The DPW method creates an analytically integrated waveform, hence filtering the sawtooth with an integrator, achieving an additional attenuation of 6 dB/octave, resulting in an amplitude of -60 dB for the aliasing bin in the previous example.

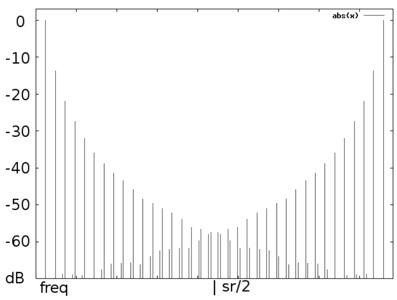
In order to restore the original signal, in DPW the generated waveform gets sampled and differentiated. As the amplitude changes for the integrator and differentiator are linearly dependent on the frequency, the aliasing effect gets reduced, because the attenuation took place at a higher frequency, whereas the boosting trough differentiation takes place at the lower, aliased frequency, resulting in an overall diminuition of aliasing. This effect is stronger the bigger the difference of the aliasing frequency and the Nyquist frequency.

2.2. The generic integrated wave-table

In order to demonstrate the effectiveness of the numerical integration Figure 2 shows a sawtooth wave stored in a lookup table and the spectral content of its integrated wave. Its not astonishing that the numerical method shows a similar improvement in terms

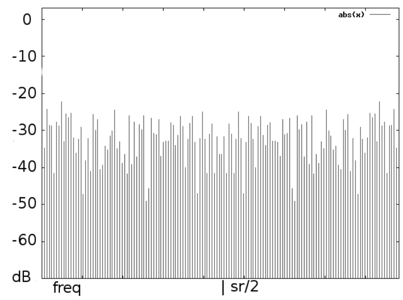


(a)

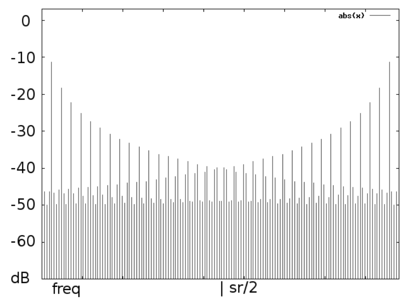


(b)

Figure 2: *(a) Generating a sawtooth waveform from a lookup table of size $N = 1024$, frequency f_0 is 1200 Hz. (b) Spectral content of a integrated lookup table of the same sawtooth.*



(a)



(b)

Figure 3: *(a) Generating a random periodic waveform from a lookup table of size $N = 1024$, frequency f_0 is 1200 Hz. (b) Spectral content of a integrated random lookup table of the same function, the fundamental frequency of 1200 Hz can be recognized.*

of suppression of aliasing than the analytic generation of the sawtooth used in DPW.

Figure 3 (a) shows the extreme case of a randomly generated function used as the basis of an oscillator. As can be seen the aliasing effect with this function is such, that the original fundamental frequency of 1200 Hz is not distinguishable anymore. Instead the frequency bins are more or less equal in amplitude and only show a harmonic pattern because of the mapping of aliased bins onto already existing ones, reflecting the real periodicity of the random wave-table, which is a common divisor of the Nyquist frequency and the oscillators frequency.

The integrated wave-table method can be compared to the method of using several band-limited wave-tables, the difference is that the low-pass function is not used to remove aliasing completely, but to diminish it over all frequencies. This way one low-pass filtered wave-table is enough.

2.3. Restoring the original wave-table

Using integrated wave-tables, the target waveform has to be restored by differentiating the synthesized waveform. Unlike the integration step this has to be done in real-time. As differentiation boils down to a simple subtraction in the digital domain, this method is very efficient.

The method will be better, the farther away from the Nyquist frequency the aliased components lie. The differentiation has the following properties as a function of frequency:

$$H(k) = kX(k) \tag{4}$$

Whereas the components close to the Nyquist frequency will be fully restored to their original value, the amplitude will fall off by $1/(n_{\text{nyquist}} - n)$, therefore achieving good results at lower frequencies where aliasing effects are more disturbing.

Figure 4 shows the spectral content of the differentiated signal, based on the integrated sawtooth. Compared with result with Figure 2 (a) the integrated wave-table method exposes a considerably improved SNR, where the aliasing noise floor of -35–40 dB is only visible at values lower than -50 dB and only for frequencies higher than about 4000 Hz.

The extreme case of a white noise based integrated wave-table is shown in Figure 4 (b). Although the noise floor is still very high in this case (-20 dB), at least the fundamental frequency is recognizable.

3. CONCLUSIONS

We presented a generic, efficient method to reduce aliasing artifacts in generic table lookup oscillators. By storing the desired wave-table function at a low frequency (long table) in its integrated form we achieve a considerable improvement of the signal to noise ratio. The method does not eliminate aliasing completely, but it improves the quality of table lookup synthesis considerably, and is directly applicable to implementations of oscillators.

We show that the problem of aliasing is not limited to the simulation of analog synthesis, but is a general problem of the table lookup oscillator, with a negative influence superior to the better investigated problem of sample accuracy in interpolation schemes.

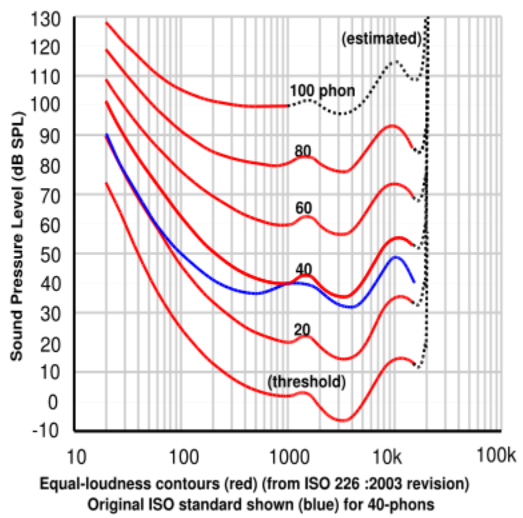
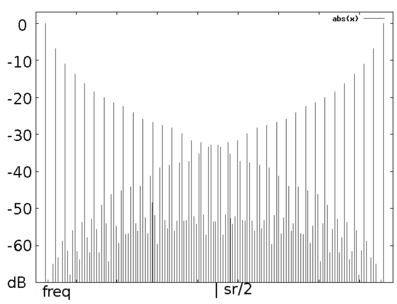
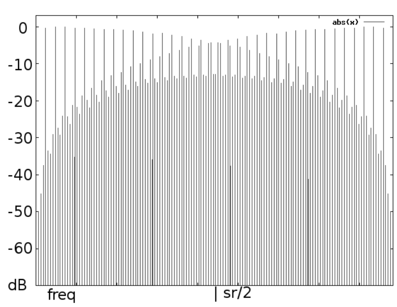


Figure 5: *Curves of equal loudness (diagram from wikipedia) suggest that aliasing that occurs above* 15 *kHz only has a limited influence upon the signal frequencies between* 200 *Hz and* 8 *kHz are critical.*



(a)



(b)

Figure 4: *(a) The spectrum of the resynthesized sawtooth, based on the integrated wave-table method. (b) The hardest case: a resynthesized noise wave-table.*

Our method is based on the Differentiated Parabolic Wave (DPW) method, but it expands it to the general case, and therefore we propose to name it generic integrated wavetable (GIW) method, reflecting how the wavetable is stored in memory (in its integrated form) than how it is resynthesized (by differentiation).

We conducted our work without any interpolation (except simple rounding) on wave-tables of a length of 1024 values in order to discard influences of interpolation and the resulting low-pass filtering. This interpolation scheme amounts to a maximal noise level due to amplitude errors of about -60 dB [9]. In a high quality implementation the table size has to be increased and a classic interpolation scheme has to be used in order to avoid amplitude errors.

4. FUTURE WORK

Although the generic integrated wave-table is a considerable improvement in the implementation of table lookup oscillators, the quality of the resulting signal might still be improved. The most obvious path for improvement is to work with higher sample rates. In this case a 6 dB SNR improvement would require the double of calculations.

We didn't make quantitative comparisons with other methods yet, although we consider it more important to check our method with listening tests, as pleasing the human ear is the final goal of our efforts.

Other methods that solve the aliasing problem, like precalculated waveforms still have to be investigated and compared to this method in terms of accuracy and speed.

5. ACKNOWLEDGEMENTS

Thanks go to Vesa Välimäki for his idea about differentiated parabolic waveforms [5], which was the main inspiration for this work.

6. REFERENCES

[1] C. Roads, *The Computer Music Tutorial*. Cambridge, MA.: MIT Press, 1996.

[2] F. R. Moore, *Elements of computer music*. Englewood Cliffs, N.J.: Prentice Hall, 1990.

[3] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proc. Int. Comp. Music Conf. (ICMC'96),* Hong Kong, 1996, pp. 332–335.

[4] E. Brandt, "Hard sync without aliasing," in *Proc. Int. Comp. Music Conf. (ICMC'01),* Havana, Cuba, 2001, pp. 365–368.

[5] V. Välimäki, "Discrete-time synthesis of the sawtooth waveform with reduced aliasing," *IEEE Sig. Proc. Letters*, vol. 12, no. 3, pp. 214–217, march 2005.

[6] J. Lane, D. Hoory, E. Martinez, and P. Wang, "Modeling analog synthesis with DSPs," *Computer Music J.*, vol. 21, no. 4, pp. 23–41, may 1997.

[7] J. O. Smith, "Digital audio resampling home page," [Online] http://www-ccrma.stanford.edu/~jos/resample/, Jan. 28 2002.

[8] J. Wright, "Synthesising band limited waveforms using wavetables," [Online] http://www.musicdsp.org/files/bandlimited.pdf, Retrieved June 29th, 2006.

[9] M. S. Puckette, "Theory and techniques of electronic music," [Online] http://www-crca.ucsd.edu/~msp/, 2006.