The Second-Order Digital Waveguide Oscillator

Julius O. Smith III
Perry R. Cook
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

Abstract

A digital sinusoidal oscillator derived from digital waveguide theory is described which has good properties for VLSI implementation. Its main features are *no wavetable* and a computational complexity of only *one multiply* per sample when amplitude and frequency are constant. Three additions are required per sample. A piecewise exponential amplitude envelope is available for the cost of a second multiplication per sample, which need not be as expensive as the tuning multiply. In the presence of frequency modulation (FM), the amplitude coefficient can be varied to exactly cancel amplitude modulation (AM) caused by changing the frequency of oscillation.

Contents

1 Introduction

One of the very first computer music techniques introduced was *additive synthesis* [3]. It is based on Fourier's theorem which states that any sound can be constructed from elementary sinusoids, such as are approximately produced by carefully struck tuning forks. Additive synthesis attempts to apply this theorem to the synthesis of sound by employing large banks of sinusoidal oscillators, each having independent amplitude and frequency controls. Many analysis methods, e.g., the phase vocoder, have been developed to support additive synthesis. A summary is given in [5].

While additive synthesis is very powerful and general, it has been held back from widespread usage due to its computational expense. For example, on a single DSP56001 digital signal-processing chip, clocked at 33 MHz, only about 60 sinusoidal partials can be synthesized in real time using non-interpolated, table-lookup oscillators. Interpolated table-lookup oscillators are much more expensive, and when all the bells and whistles are added, and system overhead is accounted for, only around 12 fully general, high-quality partials are sustainable at 44.1 KHz on a $33 MHz$ DSP56001 (based on analysis of implementations provided by the NeXT Music Kit).

At CD-quality sampling rates, the note A1 on the piano requires $22050/55 \approx 400$ sinusoidal partials, and at least the low-frequency partials should use interpolated lookups. Assuming a worst-case average of 100 partials per voice, providing 32-voice polyphony requires 3200 partials, or around 64 DSP chips, assuming we can pack an average of 50 partials into each DSP. A more reasonable complement of 8 DSP chips would provide only 4-voice polyphony which is simply not enough for a piano synthesis. However, since DSP chips are getting faster and cheaper, DSP-based additive synthesis looks viable in the future.

The cost of additive synthesis can be greatly reduced by making special purpose VLSI optimized for sinusoidal synthesis. In a VLSI environment, major bottlenecks are *wavetables* and *multiplications*. Even if a single sinusoidal wavetable is shared, it must be accessed sequentially, inhibiting parallelism. The wavetable can be eliminated entirely if *recursive algorithms* are used to synthesize sinusoids directly.

In [1], three techniques were examined for generating sinusoids digitally by means of recursive algorithms. The recursions can be interpreted as implementations of second-order digital resonators in which the damping is set to zero. The three methods considered were

1. the *coupled form* which is identical to a two-dimensional vector rotation,
2. the *modified coupled form*, or "*magic circle*" algorithm, which is similar to (1) but has ideal numerical behavior, and
3. the *direct-form, second-order, digital resonator* with its poles set to the unit circle.

These three recursions are defined as follows:

$$
\begin{array}{llcll}
(1) & x_n & = & c_n x_{n-1} + s_n y_{n-1} & \\
 & y_n & = & -s_n x_{n-1} + c_n y_{n-1} & \text{(Coupled Form)} \\
(2) & x_n & = & x_{n-1} + \epsilon y_{n-1} & \\
 & y_n & = & -\epsilon x_n + y_{n-1} & \text{(``Magic Circle'')} \\
(3) & x_n & = & 2c_n x_{n-1} - y_{n-1} & \\
 & y_n & = & x_{n-1} & \text{(Direct-Form Resonator)}
\end{array}
$$

where $c_n \triangleq \cos(2\pi f_n T)$, $s_n \triangleq \sin(2\pi f_n T)$, f_n is the instantaneous frequency of oscillation (Hz) at time sample n, and T is the sampling period in seconds. The magic circle parameter is $\epsilon = 2\sin(\pi f_n T)$.

The digital waveguide oscillator appears to have the best overall properties yet seen for VLSI implementation. The new structure was derived as a spin-off from recent results in the theory and implementation of digital waveguides [6, 7]. Any second-order digital filter structure can be used as a starting point for developing a corresponding sinusoidal signal generator, so in this case we begin with the second-order waveguide filter.

2 The Second-Order Waveguide Filter

The first step is to make a second-order digital filter with zero damping by abutting two unit-sample sections of waveguide medium, and terminating on the left and right with perfect reflections, as shown in Fig. 1. The wave impedance in section i is given by $R_i = \rho c/A_i$, where ρ is air density, A_i is the cross-sectional area of tube section i, and c is sound speed. The reflection coefficient is determined by the impedance discontinuity via $k = (R_1 - R_2)/(R_1 + R_2)$. It turns out that to obtain sinusoidal oscillation, one of the terminations must provide an inverting reflection while the other is non-inverting.

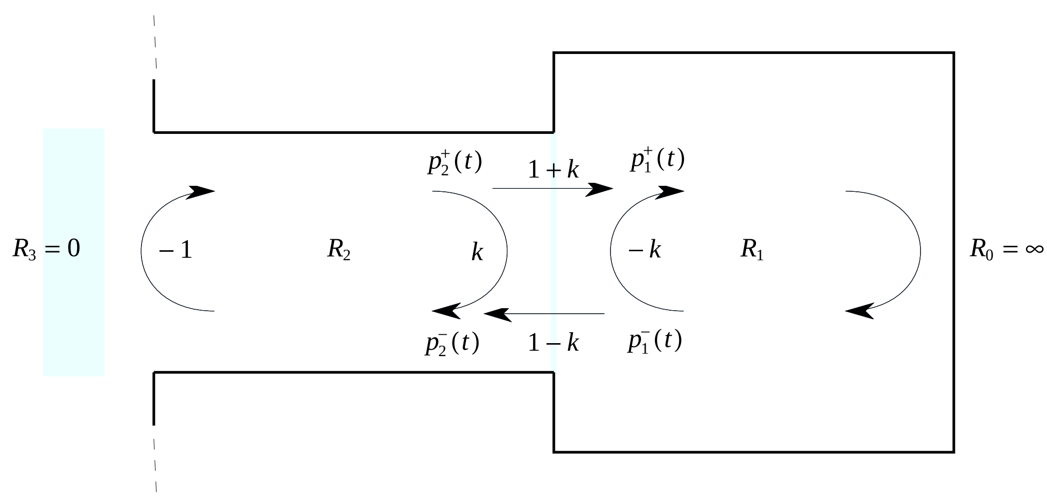

Figure 1: The second-order, lossless, digital waveguide oscillator, built using two *acoustic tube* sections.

At the junction between sections 1 and 2, the signal is partially transmitted and partially reflected such that energy is conserved, i.e., we have *lossless scattering*. The formula for the reflection coefficient k can be derived from the physical constraints that (1) pressure is continuous across the junction, and (2) there is no net flow into or out of the junction. For traveling pressure waves $p^{\pm}(t)$ and volume-velocity waves $u^{\pm}(t)$, we have $p^+(t) = Ru^+(t)$ and $p^-(t) = -Ru^-(t)$. The physical pressure and volume velocity are obtained by summing the traveling-wave components.

The discrete-time simulation for the physical system of Fig. 1 is shown in Fig. 2. The propagation time from the junction to a reflecting termination and back is one sample period. The half sample delay from the junction to the reflecting termination has been *commuted* with the termination and combined with the half sample delay *to* the termination. This is a special case of a "half-rate" waveguide filter [6].

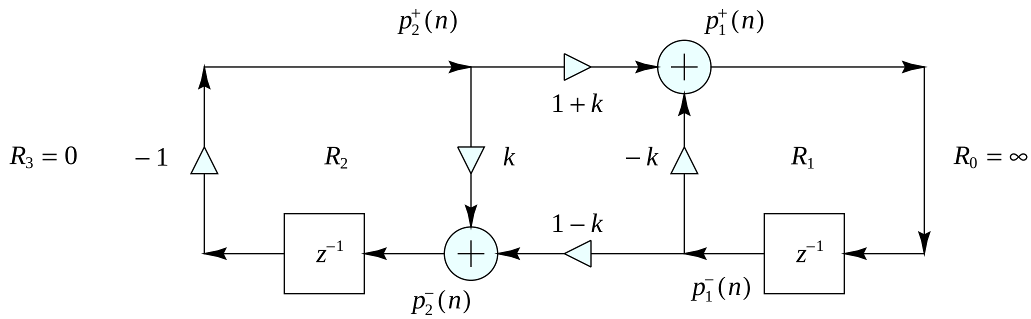

Figure 2: The second-order, lossless, waveguide filter.

Since only two samples of delay are present, the digital system is at most second order, and since the coefficients are real, at most one frequency of oscillation is possible in $(0, \pi)$.

The scattering junction shown in the figure is called the Kelly-Lochbaum junction in the literature on lattice and ladder digital filters [2]. While it is the most natural from a physical point of view, it requires four multiplies and two additions for its implementation.

It is well known that lossless scattering junctions can be implemented in a variety of equivalent forms, such as the two-multiply and even one-multiply junctions. However, most have the disadvantage of not being *normalized* in the sense that changing the reflection coefficient k changes the amplitude of oscillation. This can be understood physically by noting that a change in k implies a change in R_2/R_1. Since the signal power contained in a waveguide variable, say $p_1^+(n)$, is $\left[p_1^+(n)\right]^2/R_1$, we find that modulating the reflection coefficient corresponds to modulating the signal energy represented by the signal sample in at least one of the two delay elements. Since energy is proportional to amplitude squared, energy modulation implies amplitude modulation.

The well-known normalization procedure is to replace the traveling pressure waves $p^\pm$ by "root-power" pressure waves $\tilde{p}^\pm = p^\pm/\sqrt{R}$ so that signal power is just the square of a signal sample $(\tilde{p}^\pm)^2$. When this is done, the scattering junction transforms from the Kelly-Lochbaum or one-multiply form into the *normalized ladder* junction in which the reflection coefficients are again $\pm k$, but the forward and reverse transmission coefficients become $\sqrt{1-k^2}$. Defining $k = \sin(\theta)$, the transmission coefficients can be seen as $\cos(\theta)$, and we arrive essentially at the *coupled form*, or two-dimensional vector rotation considered in [1].

An alternative normalization technique is based on the digital waveguide *transformer* [6]. The purpose of a "transformer" is to "step" the force variable (pressure in our example) by some factor g without scattering and without affecting signal energy. Since traveling signal power is proportional to pressure times velocity p^+u^+, it follows that velocity must be stepped by the inverse factor $1/g$ to keep power constant. This is the familiar behavior of transformers for analog electrical circuits: voltage is stepped up by the "turns ratio" and current is stepped down by the reciprocal factor. Now, since $p^+ = Ru^+$, traveling signal power is equal to $p^+u^+ = (p^+)^2/R$. Therefore, stepping up pressure through a transformer by the factor g corresponds to stepping up the wave impedance R by the factor g^2. In other words, the transformer raises pressure and decreases volume velocity by raising the wave impedance (narrowing the acoustic tube) like a converging cone.

If a transformer is inserted in a waveguide immediately to the left, say, of a scattering junction, it can be used to modulate the the wave impedance "seen" to the left by the junction without having to use root-power waves in the simulation. As a result, the one-multiply junction can be used for the scattering junction, since the junction itself is not normalized. Since the transformer requires two multiplies, a total of three multiplies can effectively implement a normalized junction, where four were needed before. Finally, in just this special case, one of the transformer coefficients can be commuted with the delay element on the left and combined with the other transformer coefficient. For convenience, the -1 coefficient on the left is commuted into the junction so it merely toggles the signs of inputs to existing summers. These transformations lead to the final form shown in Fig. 3.

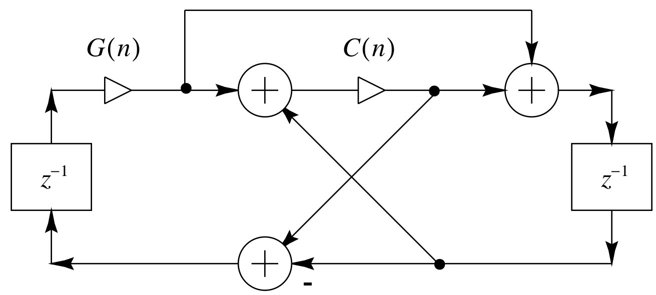

Figure 3: The transformer-normalized, digital waveguide oscillator.

The "tuning coefficient" is given by $C(n) = \cos(2\pi f_n T)$, where f_n is the desired oscillation frequency in Hz at sample n, and T is the sampling period in seconds. The "amplitude coefficient" is $G(n) = r_n g_n / g_{n-1}$, where $r_n = e^{-T/\tau_n}$ is the exponential growth or decay per sample ($r_n \equiv 1$ for constant amplitude), and g_n is the normalizing transformer "turns ratio" given by $g_n = \sqrt{[1 - C(n)]/[1 + C(n)]}$. When both amplitude and frequency are constant, we have $G(n) \equiv 1$, and only the tuning multiply is operational. When frequency changes, the amplitude coefficient deviates from unity for only one time sample to normalize the oscillation amplitude.

When amplitude and frequency are constant, there is no gradual exponential growth or decay due to round-off error. This happens because the only rounding is at the output of the tuning multiply, and all other computations are exact. Therefore, quantization in the tuning coefficient can only cause quantization in the frequency of oscillation. Note that any one-multiply digital oscillator should have this property. In contrast, the only other known normalized oscillator, the coupled form, *does* exhibit exponential amplitude drift because it has *two* coefficients $c = \cos(\theta)$ and $s = \sin(\theta)$ which, after quantization, no longer obey $c^2 + s^2 = 1$ for most tunings.

3 Conclusions

A recursive algorithm was presented for digital sinusoid generation that has excellent properties for VLSI implementation. It is like the coupled form in that it offers instantaneous amplitude from its state and constant amplitude in the presence of frequency modulation. However, its implementation requires only one or two multiplies per sample instead of four.

While these properties make the new oscillator appear ideally suited for FM applications in

VLSI, there are issues to be resolved regarding conversion from modulator output to carrier coefficients. Preliminary experiments indicate that FM indices less than 1 are well behaved when the output of a modulating oscillator simply adds to the coefficient of the carrier oscillator (bypassing the exact FM formulas). Approximate amplitude normalizing coefficients have also been derived which provide a first-order approximation to the exact AM compensation at low cost. For music synthesis applications, we believe a distortion in the details of the FM instantaneous frequency trajectory and a moderate amount of incidental AM can be tolerated since they produce only second-order timbral effects in many situations.

References

[1] J. W. Gordon and J. O. Smith, "A sine generation algorithm for VLSI applications," in *Proceedings of the 1985 International Computer Music Conference, Vancouver*, Computer Music Association, 1985.

[2] A. H. Gray and J. D. Markel, "Digital lattice and ladder filter synthesis," in *Selected Papers in Digital Signal Processing* (Digital Signal Processing Committee, ed.), IEEE Press, 1975.

[3] J.-C. Risset, "Computer music experiments, 1964—...," *Computer Music Journal*, vol. 9, no. 1, pp. 11–18, 1985, Reprinted in [4].

[4] C. Roads, ed., *The Music Machine*, Cambridge, MA: MIT Press, 1989.

[5] X. Serra and J. O. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990, Software and recent papers URL: `//www.iua.upf.es/~xserra/`.

[6] J. O. Smith, "Music applications of digital waveguides," Tech. Rep. STAN-M-39, CCRMA, Music Department, Stanford University, 1987, CCRMA Technical Report STAN-M-39, `http://-ccrma.stanford.edu/STANM/stanm39/`.

[7] J. O. Smith, "Waveguide filter tutorial," in *Proceedings of the 1987 International Computer Music Conference, Champaign-Urbana*, pp. 9–16, Computer Music Association, 1987.