A New Recursive Quadrature Oscillator

Martin Vicanek

21. October 2015

1 Introduction

Digital harmonic oscillators are standard devices used in sound synthesis, control systems, test and measurement equipment, heterodyning, frequency conversion, and many other DSP applications. A harmonic or sine wave oscillator generates a pure tone without any overtones. In additive synthesis, a number sine waves are added to generate timbre the Hammond organ is a prominent example of this principle.

Methods for generating sine waves in the digital domain fall roughly into two categories:

1. Recursive oscillators, where the actual output value is computed from previous states
2. Direct evaluation as a function of a phase accumulator (= non band-limited ramp)

The scheme presented in this note belongs to the first category.

There are many algorithms for recursive oscillators, with different characteristics in terms of accuracy (especially at low frequencies), stability, computational complexity, etc.[1, 2]. In this note I present a new quadrature oscillator as the only hyperstable, equi-amplitude quadrature oscillator that I am aware of.

Recursive oscillators perform best at constant frequency. Modulation requires coefficient(s) updating and may often result in amplitude change. For an equi-amplitude quadrature oscillator, the situation is less problematic. It is somewhat difficult to add phase modulation (it can be done for a quadrature oscillator, but it is not very efficient).

2 Recursion Scheme

Let u_n and v_n denote the oscillator outputs. The goal is to provide recursion equations for a harmonic quadrature oscillation, i.e.

$$u_n = \cos(n\omega), \quad v_n = \sin(n\omega) \tag{1}$$

with a specified frequency ω in radians per sample. Given the state at some discrete point in time n, the next state may be found by a simple rotation. For reasons to become clear later, we will first advance from time n by half a sample to the intermediate time $n + 1/2$,:

$$u_{n+1/2} = cu_n - sv_n \tag{2}$$
$$v_{n+1/2} = su_n + cv_n \tag{3}$$

where we have used the abreviations $c = \cos(\omega/2)$ and $s = \sin(\omega/2)$. Advance another half step to time $n + 1$,

$$u_{n+1} = cu_{n+1/2} - sv_{n+1/2} \tag{4}$$
$$v_{n+1} = su_{n+1/2} + cv_{n+1/2} \tag{5}$$

Likewise, we may write down equations for the reverse transitions by half a sample frm $n + 1$ to $n + 1/2$,

$$u_{n+1/2} = cu_{n+1} + sv_{n+1} \tag{6}$$
$$v_{n+1/2} = -su_{n+1} + cv_{n+1} \tag{7}$$

and from $n + 1/2$ to n:

$$u_n = cu_{n+1/2} + sv_{n+1/2} \tag{8}$$
$$v_n = -su_{n+1/2} + cv_{n+1/2} \tag{9}$$

We will not make use of all eight equations (2)-(9), since they are redundant. It will be sufficient to use a selection of four, namely equations (2) and (6), and equations (5) and (9). Thus, equation (2) provides an expression for the intermediate state $u_{n+1/2}$. Then equation (6) advances u another half step to the time $n+1$. To propagate v from n to $n+1$, subtract equation (9) from equation (5) to obtain

$$v_{n+1} = v_n + 2su_{n+1/2} \tag{10}$$

Note that the intermediate state $u_{n+1/2}$ is only an internal temporary variable, a stepstone to leapfrog from v_n to v_{n+1}. We may save two multiplies if we work with the auxiliary quantity $w_n := u_{n+1/2}/c$. Then equations (2), (10) and (6) become

$$w_n = u_n - k_1 v_n \tag{11.1}$$
$$v_{n+1} = v_n + k_2 w_n \tag{11.2}$$
$$u_{n+1} = w_n - k_1 v_{n+1} \tag{11.3}$$

where the two constants k_1 and k_2 are

$$k_1 = \tan(\omega/2), \quad k_2 = 2\sin(\omega/2)\cos(\omega/2) = \sin\omega = 2k_1/(1+k_1^2) \tag{12}$$

Equations (11.1)-(11.3) constitute the main result of this paper. With initial values chosen as

$$u_0 = 1, \quad v_0 = 0 \tag{13}$$

the result of the recursion is a quadrature harmonic oscillation as in equation (1).

Figure 1 shows the block diagram corresponding to equations (11).

Below is a sample-by-sample processing algorithm in pseudocode.

```
// initialize u and v
at start do{
u = 1;
v = 0;
}

// update coefficients
if frequency changes do{
update w;
```

```
k1 = tan(0.5*w);
k2 = 2*k1/(1 + k1*k1);
}

// iterate filter
for every sample do{
tmp = u - k1*v;
v   = v + k2*tmp;
u   = tmp - k1*v;
}
```

3 Stability

For a stability analysis we take the z-transform of equations (11.1)-(11.3),

$$\begin{aligned} w(z) &= u(z) - k_1 v(z) \\ z v(z) &= v(z) + k_2 w(z) \\ z u(z) &= w(z) - z k_1 v(z) \end{aligned} \tag{14}$$

Eliminating $w(z)$ and rearranging, we obtain the following homogeneous system of equations for $u(z)$ and $v(z)$:

$$\begin{aligned} k_2 u + [(1 - k_1 k_2) - z] v &= 0 \\ (z-1) u + (z+1) k_1 v &= 0. \end{aligned} \tag{15}$$

Nontrivial solutions exist where the determinant vanishes,

$$z^2 - 2(1 - k_1 k_2) z + 1 = 0 \tag{16}$$

For $0 < k_1 k_2 < 2$, the roots $z_{1,2}$ of equation (16) are complex conjugate and lie exactly on the unit circle. We may write

$$z_{1,2} = e^{\pm i\omega} \tag{17}$$

with the frequency ω given by

$$\cos \omega = 1 - k_1 k_2. \tag{18}$$

Even if k_1 and k_2 are subject to quantization in a real implementation, the poles still lie exactly on the unit circle hence preventing exponential runaway: the presented oscillator is hyperstable, as opposed to the well-known coupled form quadrature oscillator.

Futrhermore, by virtue of equation (15),

$$u/v = \pm i \sin\omega/k_2 = \pm i k_1/\tan(\omega/2) = \pm i\sqrt{k_1\left(\frac{2}{k_2} - k_1\right)}, \qquad (19)$$

i.e. the eigenmodes exhibit a phase lag between u and v of exactly $\pi/2$, regardless of coefficient quantization. The ratio of the corresponding amplitudes is unity if

$$k_2 = 2k_1/(1 + k_1^2) \qquad (20)$$

In practice, equation (20) will be valid only within machine precision, resulting in slightly different amplitudes.

4 Modulation

An interesting question is how the system responds to frequency modulation, i.e. coefficient changes while running freely. Neglect for a moment roundoff errors introduced by finite precision arithmetics, then the trajectory in u-v-space will be an ellipse. A deviation from a circular trajectory occurs because equation (20) is fulfilled only to machine precision. Now if we change the values of k_1 and k_2, the system's trajectory will simply switch to another ellipse, possibly moving at a different speed. As long as equation (20) remains approximately valid, the new ellipse will be close to circular.

If coefficients are changed frequently, the deviations may accumulate in the long run, ultimately resulting in an altered amplitude. Whether or not this is a concern depends on the application. For sound generation in a musical context, this is not an issue at all.

5 Roundoff Error

A rigorous analysis of the effect of roundoff errors on the iteration equation (11) is beyond the scope of this note. Quantization is often modelled as additive noise, without taking into account a possible correlation to the signal itself. Such a model would result in a slow drift with the accumulated error increasing as the square root of the number of iterations. However, this behavior is not supported by empirical findings, which suggest that the system rather locks in into a limiting cycle. The oscillator has been run at 44.1 kHz sample rate for days, with varying initial conditiona and coefficient values. In no case has there been an unbounded drift. Again, this is in harsh contrast with the coupled form oscillator, which shows systematic deviations already in the first few seconds, and ultimately terminates in an exponential runaway.

Appendix: Some Common Oscillators

Table 1 lists some recursive oscillators with their respective properties. For each listed oscillator tyxpe we give the governing equations below.

Biquad Oscillator

The biquad oscillator is a direct form I realization, and is the most used oscillator form. With one multiply and one add it is a very economic scheme, however accuracy is bad at low frequencies.

Recurrence	Parameter	Initialization	Output
$u_{n+1} = ku_n - u_{n-1}$	$k = 2\cos\omega$	$u_1 = 0$	$u_n = \sin(n\omega)$
		$u_0 = -\sin\omega$	

Reinsch Oscillator

This is another CPU friendly oscillator, with good accuracy at low frequencies. It is the ideal static oscillator.

Recurrence	Parameter	Initialization	Output
$u_{n+1} = u_n + v_n$	$k = 4\sin^2(\frac{1}{2}\omega)$	$u_0 = 0$	$u_n = \sin(n\omega)$
$v_{n+1} = v_n - ku_{n+1}$		$v_0 = \sin\omega$	$v_n = A\cos[(n + \frac{1}{2})\omega]$
			$A = 2\sin(\frac{1}{2}\omega)$

Digital Waveguide Oscillator

Another one-multiply oscillator with quadrature output, however with bad accuracy at low frequencies.

Recurrence	Parameter	Initialization	Output
$s_n = k(u_n + v_n)$	$k = \cos\omega$	$u_0 = 0$	$u_n = A\sin(n\omega)$
$t_n = s_n + u_n$		$v_0 = 1$	$v_n = \cos(n\omega)$
$u_{n+1} = s_n - v_n$			$A = -\tan(\frac{1}{2}\omega)$
$v_{n+1} = t_n$			

Quadrature Oscillator with Staggered Update

Recurrence	Parameter	Initialization	Output
$v_{n+1} = u_n + kv_n$	$k = \cos\omega$	$u_0 = 0$	$u_n = A\sin(n\omega)$
$u_{n+1} = kv_{n+1} - v_n$		$v_0 = 1$	$v_n = \cos(n\omega)$
			$A = -\sin\omega$

Magic Circle Oscillator

A popular 'almost quadrature' oscillator. The outputs have equal amplitudes and a phase difference of 90 degrees plus half a sample.

Recurrence	Parameter	Initialization	Output
$u_{n+1} = u_n - kv_n$	$k = 2\sin(\frac{1}{2}\omega)$	$u_0 = \cos(\frac{1}{2}\omega)$	$u_n = \cos[(n - \frac{1}{2})\omega]$
$v_{n+1} = v_n + ku_{n+1}$		$v_0 = 0$	$v_n = \sin(n\omega)$

Coupled Form Oscillator

The standard quadrature, equal amplitudes oscillator. Unfortunaltely it is not numerically stable: if left run freely, amplitudes will grow or decay exponentially because the poles generally do not lie exactly on the unit circle. Therefore, some sort of automatic gain control has to be applied. Updating the parameters, i.e. frequency modulation, is somewhat CPU-demanding.

Recurrence	Parameters	Initialization	Output
$u_{n+1} = k_1 u_n - k_2 v_n$	$k_1 = \cos\omega$	$u_0 = 1$	$u_0 = \cos(n\omega)$
$v_{n+1} = k_2 u_n + k_1 v_n$	$k_2 = \sin\omega$	$v_0 = 0$	$v_0 = \sin(n\omega)$

Present Work

A stable quadrature oscillator with equal amplitudes, good accuracy at low frequencies, and reasonable CPU load.

Recurrence	Parameters	Initialization	Output
$w_n = u_n - k_1 v_n$	$k_1 = \tan(\frac{1}{2}\omega)$	$u_0 = 1$	$u_0 = \cos(n\omega)$
$v_{n+1} = v_n + k_2 w_n$	$k_2 = \sin\omega$	$v_0 = 0$	$v_0 = \sin(n\omega)$
$u_{n+1} = w_n - k_1 v_{n+1}$			

References

[1] P. Symons, "Digital Waveform Generation", Cambridge University Press 2013, pp. 90-108.

[2] Clay Turner, "Recursive Discrete Time Sinusoidal Oscillators", IEEE Signal Processing Mag, May 2003, p103-111.

[3] J. W. Gordon and J. O. Smith, A Sine Generation Algorithm for VLSI Applications, in Proc. Int. Computer Music Conf. (1985), pp. 165168.

[4] J. O. Smith and P. R. Cook, The Second-Order Digital Waveguide Oscillator, in Proc. Int. Computer Music Conf. (1992), pp. 150153.

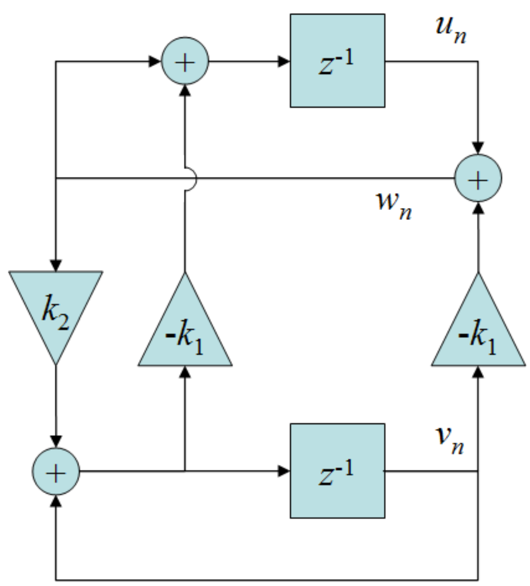

Figure 1: Block diagram of the oscillator corresponding to equations (11)

Oscillator type	Equal amplitudes	Quadrature	Stable	Low-frequency accurate
Biquad	yes	no	yes	no
Reinisch	no	nearly	yes	yes
Digital Waveguide	no	yes	yes	no
Quad Staggered Update	no	yes	yes	no
Magic Circle	yes	nearly	yes	yes
Coupled Quad	yes	yes	no	yes
This work	yes	yes	yes	yes

Table 1: Some common oscillators and their properties