

# Implementing Efficient Frequency Variation in Coupled-Mode Synthesis and Other Cosine-Frequency Systems

Timothy Stilson (stilti@staccatosys.com)  
Scott A. Van Duyne (savad@staccatosys.com)  
Staccato Systems, Inc.  
2155 Park Blvd, Palo Alto, CA, 94306

## Abstract

Many oscillator and filter algorithms are tuned via coefficients that are of the form  $\cos(2\pi f/f_s)$ . This can make implementing frequency modulation, and pitch-bend expensive, at least one table lookup per frequency modification. In this paper, an approximation to the cosine is explored which allows for direct modulation without cosine re-computation or table lookup. With this ability, fast and/or smooth time-variation of mode frequencies is possible without significant increase in expense, while retaining good low-deviation tuning accuracy.

## 1 Introduction

Quite a few oscillator and filter algorithms, including the direct-form second-order filter, the second-order digital waveguide oscillator [Smith and Cook, 1992] and, more recently, the coupled-mode synthesis filter [Van Duyne, 1997], are tuned via coefficients that are related to frequency by a cosine operation. This can make implementing frequency variation, such as FM, vibrato, and pitch-bend expensive, requiring at least a good table lookup per frequency modification. In this paper, approximations to the cosine are explored, which allow for direct modification of the coefficients without cosine re-computation or table lookup.

With this ability, fast and/or smooth time-variation of mode frequencies in Coupled-Mode Synthesis is possible, allowing for some interesting effects, both physical and non-physical. The range of non-physical effects is quite wide, while possible physical effects may include: simulating amplitude dependent modal frequencies in drums; simulating the pitch sliding effects of moving the pedal on a kettle drum; and simulating the sounds of water-filled shells, whose modal frequencies change as the water moves about.

In Section 2, the approximation will be derived. In Section 3, its frequency-deviation characteristics will be explored. In Section 4, more trivial deviations will be compared.

## 2 Derivation

Our goal is to implement the cosine coefficient in such a way as to be able to modify the coefficient to implement frequency variations in as cheap a way as possible while still keeping as much useful control over the exact variation as possible (i.e. the cosine approximation should be good). In keeping with the cheapness goal, it will be assumed that the frequency variation is based on a signal input to the system, so that it changes as fast as once per sample (this creates a context for the definition of “cheap”). It is also assumed that a table lookup is too expensive (either due to interpolation computation or memory access), so as to motivate a search for another method.

First we will define the frequency variation. We have chosen a multiplicative deviation, as it fits musical requirements: new freq =  $\alpha f$ , where  $f$  is the base frequency at which the original coefficients are computed ( $f$  is assumed to be change “rarely”, so that exact cosine computation can be done when it changes), and  $\alpha$  is the

frequency deviation ratio. Thus our base coefficient  $c$  is defined as  $c = \cos(2\pi f/f_s)$ , and our deviated coefficient  $\tilde{c}$  should approximate the value  $\cos(\alpha 2\pi f/f_s)$ . For simplicity, let  $\theta = 2\pi f/f_s$ , so that  $c = \cos(\theta)$ .  $\theta$  varies by the same ratios as  $f$ , so that  $\alpha f$  corresponds to  $\alpha\theta$  (i.e.  $\tilde{c}$  must approximate  $\cos(\alpha\theta)$ ).

First, we look at the polynomial expansion of cosine ([Abramowitz 1964], for example):

$$c = \cos(\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} - \dots$$

The deviated true coefficient is:

$$\cos(\alpha\theta) = 1 - \frac{\alpha^2\theta^2}{2!} + \frac{\alpha^4\theta^4}{4!} - \frac{\alpha^6\theta^6}{6!} + \frac{\alpha^8\theta^8}{8!} - \dots$$

Define a new coefficient  $d = c - 1$ , so that  $c = 1 + d$ , thus:

$$d = -\frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} - \dots$$

We now introduce our approximation:

$$\tilde{c} = 1 + \alpha^2 d = 1 - \frac{\alpha^2\theta^2}{2!} + \frac{\alpha^2\theta^4}{4!} - \frac{\alpha^2\theta^6}{6!} + \frac{\alpha^2\theta^8}{8!} - \dots$$

This approximation deviates first all the way at the fourth order, and if  $\alpha$  is near 1.0 (i.e. small deviations),  $a^n \approx a^2$  for small  $n$ , thus increasing the accuracy of the approximation a further. Also, this approximation is better for small  $\theta$ , which works well because most useful frequencies correspond to rather small  $\theta$  at  $f_s = 44100\text{Hz}$ .

We can use this approximation by updating  $c$  via  $1 + \alpha^2 d$ , or if updating coefficients is difficult (as it can be in some situations), by replacing all multiplications by  $c$  with an add and another multiply (see Figure 1). Note that if  $\alpha$  is updated every sample, there is no reason to use the former method over the latter, since in the former case, the coefficients would have to be stored to memory. This storage is a useless operation since the new value is useful only in the current sample, because it will be recomputed in next sample anyways.

In multi-mode systems, like coupled-mode synthesis, the same value of  $\alpha$  is likely to be used for all modes, which means  $\alpha^2$  need only be computed once per sample rather than once per mode per sample.

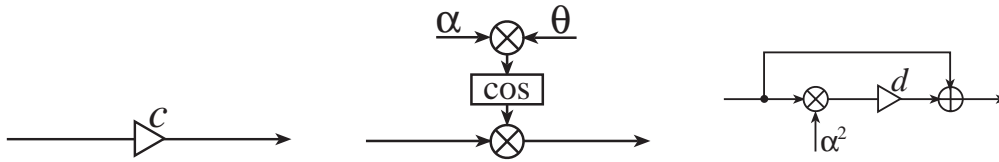


Figure 1: Modulating  $c$ , l to r: no modulation, exact modulation,  $1 + \alpha^2 d$  approximation.

When used to modulate high-Q filters, care must be taken in the modulation, because the time-varying filter could become unstable (even if all intermediate states are stable in steady state). If this is likely to be a problem, the filter should have safeguards implemented, such as state clipping. Also, when performing large deviations near  $c = 1$  or  $c = -1$ ,  $|\tilde{c}|$  may get larger than 1.0. In many systems, this will represent an unstable system, so coefficient clipping may also be necessary in certain situations.

### 3 Evaluating the Approximation

The accuracy of the approximation depends mainly on two parameters:  $\theta$  and  $\alpha$ . At low base frequencies (small  $\theta$ ), and/or at low frequency deviations ( $\alpha$  near 1.0), the approximation will be very good, getting progressively worse as the base frequency increases and the modulation deviation increases.

In Figure 2, we compare the approximation to the desired values for  $\alpha$  in the range  $[0,2]$ , corresponding to the range from DC to up one octave. Since the approximation also depends on  $\theta$ , we show the approximations at various base frequencies. The first set of graphs show compare  $c$  and  $\tilde{c}$ , the next compare actual tuning based on  $\tilde{c}$  against desired tuning, and the third set shows the deviation in the actual tuning from the desired tuning, in cents.

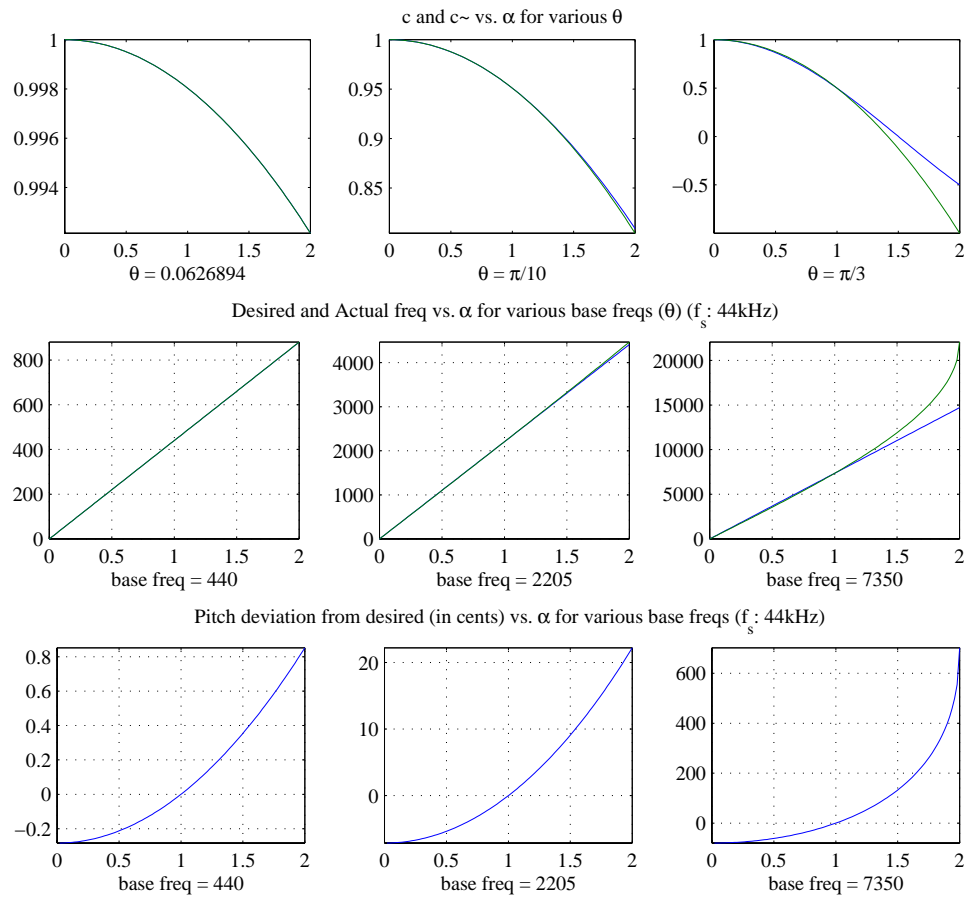


Figure 2: Evaluating the approximation across frequency ratios ( $\alpha$ ) from 0 to 2 (up one octave)

The figure shows that for musically useful fundamental frequencies, the approximation works well, especially for frequency deviations of less than an octave. If, however, we are using this approximation to deviate a harmonic of a musical signal, the approximation may be less useful. As an example, if we take a set of partials that are harmonically related at the base frequency, and deviate them, they could end up significantly inharmonic at the extremes of the deviation.

#### 4 Other Methods

In most of the algorithms mentioned at the top of the paper, the tuning coefficient enters into the system in such a way that any value of  $-1 \leq c \leq 1$  will represent a valid frequency. Therefore, even more trivial ways of varying  $c$  will produce frequency deviation, such as  $\tilde{c} = c + \alpha$  or  $\tilde{c} = \alpha c$  (with appropriate clipping to  $[-1,1]$ ). These variations will be much more approximate than the above approximation, as shown in Figure 3. Note, in particular, that the range of useful  $\alpha$  now becomes a function of the base frequency, which makes the use of these variations more tricky. Certain applications don't require accurate tuning, though, so these trivial variations need not be rejected offhand.

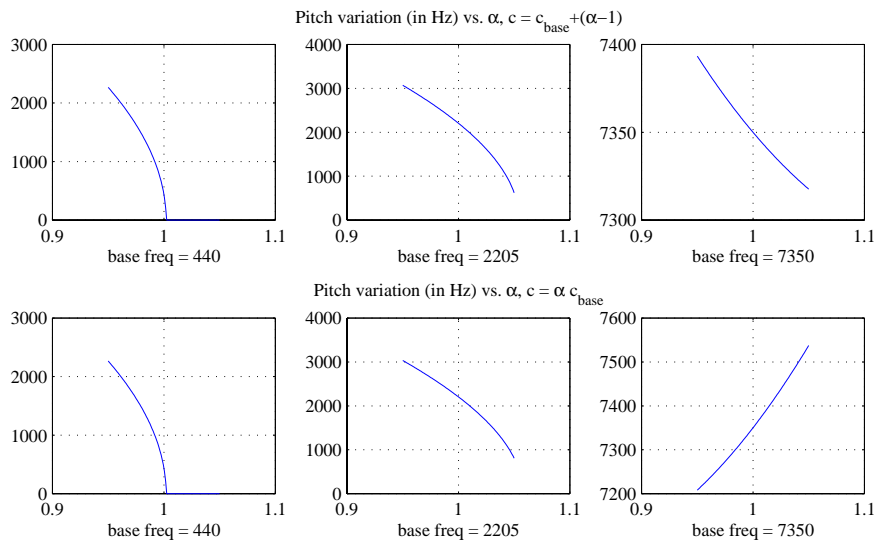


Figure 3: *Frequency deviation characteristics for more trivial coefficient variations.*

## 5 Summary

An approximation of cosine, useful for small-ratio deviations from precomputed values, was presented. The approximation is intended for use in implementing inexpensive frequency modulations in cosine-tuned oscillators and filters. Numerical evaluation of the approximation shows its usefulness for most musically useful frequencies and variations (such as vibrato).

## 6 References

- [Abramowitz 1964] , M. Abramowitz and I. A. Stegun, **Handbook of Mathematical Functions**, Dover Publications, 1964.
- [Smith and Cook, 1992] J. O. Smith and P. R. Cook, "The Second-Order Digital Waveguide Oscillator," International Computer Music Conference, San Jose, Oct., 1992.
- [Van Duyne 1997] Scott A. Van Duyne, "Coupled Mode Synthesis," International Computer Music Conference, Thessaloniki, Oct., 1997.