Glitch Free FM Vocal Synthesis

Chris Chafe
Center for Computer Research in Music and Acoustics, Stanford University
cc@ccrma.stanford.edu

ABSTRACT

Frequency Modulation (FM) and other audio rate non-linear modulation techniques like Waveshaping Digital Synthesis, Amplitude Modulation (AM) and their variants are well-known techniques for generating complex sound spectra. Kleimola [1] provides a comprehensive and up-to-date description of the entire family. One shared trait is that synthesizing vocal sounds and other harmonically-structured sounds comprised of formants can be problematic because of an obstacle which causes distortions when cranking up time-varying controls.

Large deflections of pitch or phoneme parameters cause jumps in the required integer approximations of formant center frequencies. Trying to imitate human vocal behavior with its often wide prosodic and expressive excursions causes audible clicks. A partial solution lay buried in some code from the 80's. This, combined with a new kind of oscillator bank which produces uniform phase harmonic components ensures artifact-free, exact formant spectra even under the most extreme dynamic conditions. The paper revisits singing and speech synthesis using the classic FM single modulator / multiple-carrier structure pioneered by Chowning [2]. The revised method has been demonstrated as software written in Faust and is as efficient as its predecessor technique. FM for singing synthesis can now be "abused" with radical time-varying controls. It also has potential as an efficient means for low-bandwidth analysis – resynthesis speech coding.

1. INTRODUCTION

Synthesis of singing voice by computer has a history which begins in the very first years of computer music. The song *Daisy Bell (Bicycle Built for Two)* was sung by a computer in 1961 in an arrangement by Max Mathews and Joan Miller with vocal synthesis by John Kelly and Carol Lochbaum when the Bell Telephone Laboratories experiments with digital music synthesis were only 4 years old. It was an early case of analysis – resynthesis speech coding providing a means for singing synthesis. Over the decades, literally every known synthesis technique has been applied to emulate the singing voice (additive, subtractive, physical model, FOF, etc.). The quest continues more than fifty years later with composers attracted to vocal synthesizers like Yamaha's Vocaloid [1] where they can explore a fascination with musical personalities of singers which never existed. This paper joins the thread which was started in the late 70's, early 80's involving FM for vocal synthesis and which has been virtually languishing since it's early use in a few musical works.

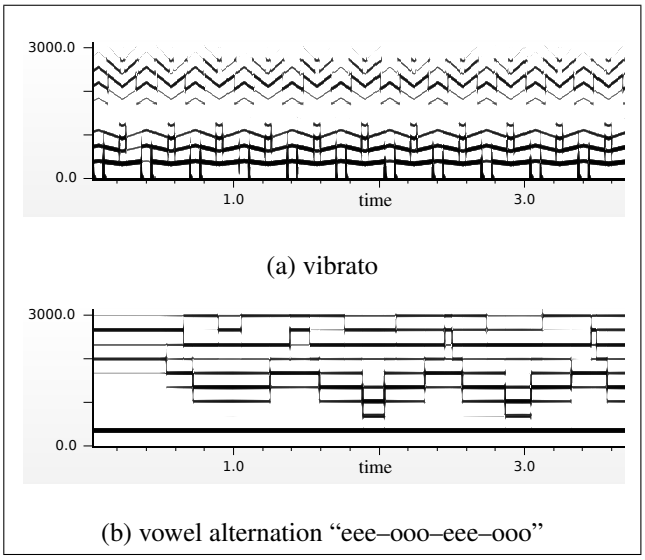


Figure 1. Clicks always occur when transitions to a new formant center frequency f_c forces a carrier oscillator to change its harmonic ratio. FM vocal formants use a $c : m$ ratio where $c \in \mathbb{N}_{\geq 1}$ and $m = f_p$, where f_p is the desired pitch.

John Chowning's FM singing voice method was first described in his 1980 article [2] prior to completing *Phonē* at IRCAM (1981). The multi-channel tape piece features a wide variety of singing voices and morphing of vocal timbres with other FM-generated timbres such as gongs. The technique creates multiple formants with independent tunings using multiple carriers and a shared modulator. Two formants are used for his version of a soprano voice "eee" and three formants for his spectrally-rich basso *profondissimo*. A later version adds a third formant to the soprano model in a synthesis of the vowel "ahh" [3]. Pitch vibrato which causes synchronous spectral modulation is especially effective and Chowning has often demonstrated how crucial this is to rendering vowels convincingly. "It is striking that the tone only fuses and becomes a unitary percept with the addition of the pitch fluctuations, thus *spectral envelope does not make a voice!*" [2].

[1] Vocaloid3 uses a triphone frequency-domain concatenative synthesis engine.

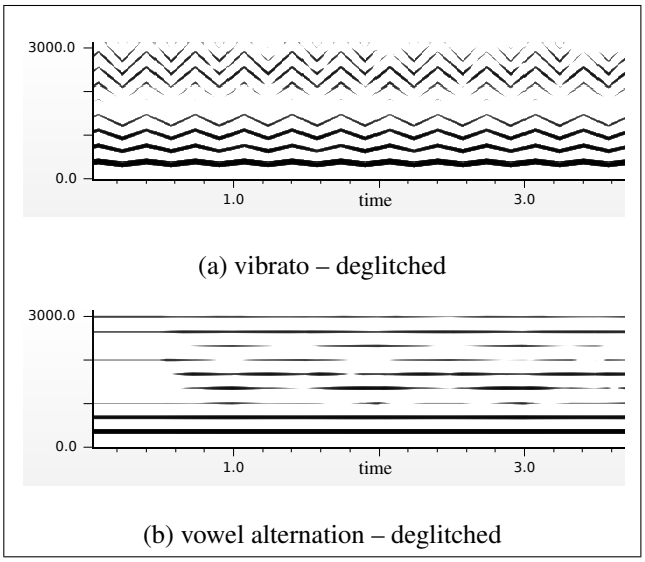
The method has an inherent shortcoming which limits the amount of vibrato excursion and limits phoneme transitions to nearby phonemes. Beyond these limits noticeable artifacts occur which are caused by discrete shifts of formant center frequency. Discontinuities are perceived as clicks and result from integer shifts in the carrier to modulator ratio $c : m$ which are required in order to track a desired formant center frequency f_c for a given pitch f_p. The modulating oscillator is always set to f_p, so $m = 1.0$. The carrier ratio c is an *integer approximation* and quantization of the actual *real* ratio f_c/f_p.

Formant synthesis with FM is essentially contradictory to the physics. The harmonic nature of voiced sound allows only harmonic number ratios $c \in \mathbb{N}_{\geq 1}$ for the carrier. Where physical sound production is an excitation – resonance mechanism with independent tuning of both elements, FM models can only approximate the resonance frequencies of the latter when constrained to produce harmonic spectra. The inherent problem is that these approximations are discontinuous in frequency. In practice, this severely limits the amount of pitch skew not only constraining vibrato, but also portamento and glissando to small ranges. In the method's original form, it is impossible to shift ratios without causing glitches like those shown in the spectrograms of Fig. 1.

I have encountered the problem in my attempts to use the method as the synthesis engine for a sonfication project involving EEG data. FM singing offers advantages for this body of work which attempts to fashion a singing choir direct from the mind. The goal isn't what one probably first imagines e.g., an ensemble of mind-controlled voices. Instead, this is a technology for auditory display of the rapid fluctuations of brain signals. Singing voice synthesis has its attractions in that it can allude to imagery of "inner voices" but it is also particularly apt because of the ease with which listeners lock on to patterns of phonemic and other voice-like timbral transitions. The range of data encountered in EEG (from quiescent to seizure) and my desire to have a very flexible mapping strategy have been motivations for the present investigation into solving the discontinuity problem. The completed work will reach the public as a gallery installation (exploring recorded data) and as a medical monitoring device for detecting seizures (with the singing voice controlled directly from electrodes in real time).

2. EARLY SOLUTION

Marc Le Brun described digital waveshaping synthesis in 1979 as a generalized paradigm for non-linear modulation synthesis [4]. FM is a special case of waveshaping synthesis and in devising a way to avoid the discontinuity problem for waveshaping, Le Brun also solved it for the FM case. Le Brun's solution remains unpublished (until now) with one exception: Bill Schottstaedt has preserved it as a synthesis instrument in the Common Lisp Music (CLM) project [5]. From the code comment, "**Vox**, an elaborate multi-carrier FM instrument is the voice instrument written by Marc Le Brun, used in *Colony* and other pieces."



(a) vibrato – deglitched

(b) vowel alternation – deglitched

Figure 2. Result of applying the solution adopted from Le Brun to the synthesis shown in Fig.1.

Vox avoids the integer ratio shift discontinuities by implementing a cross-fading solution. Two carriers corresponding to even and odd harmonic numbers are assigned to each formant "bracketing" the true formant center frequency. Their assignments are made from the two nearest harmonics $f_{lower} = \lfloor f_c/f_p \rfloor$ while the other is the nearest upper harmonic $f_{upper} = \lceil f_c/f_p \rceil$. The assignment of harmonics to individual oscillators is dynamic and depends on whether they are even numbered or odd numbered. When an oscillator is required to change its harmonic number the other will be approaching the actual target f_c/f_p. The two carrier oscillators' amplitudes sum to unity in a mixture whose gains are complementary and linearly determined by proximity to the target. The key feature which makes this work is that it ensures that the oscillator which is having its frequency changed will be muted. As a nice sideeffect, it also sharpens the accuracy with which the target formant center frequency is being synthesized.

Le Brun's paper describes "a unified conceptual framework for a number of nonlinear techniques, including frequency-modulation synthesis. Both the theory and practice of the method are developed fairly extensively, beginning with simple but useful forms and proceeding to more complex and richer variations." The cross-fade solution however only existed in code from the same era. To detail the historical record precisely, its first implementation was written in the MUS10 compiler (Stanford Artificial Intelligence Laboratory's version of Bell Laboratories' MusicN compilers). Later, it was ported to CLM as **pqw-vox** a "translation from MUS10 of MLB's waveshaping voice instrument (using phase quadrature waveshaping)." Today, both **pqw-vox** and the FM version **vox** can be found translated to Scheme in Schottstaedt's Snd project [5] as instruments defined the file *clm-ins.scm*. [2]

The cross-fade solution has not been incorporated in other

[2] One caution: some implementation versions belonging to this family have mistakenly labeled carrier oscillators as "modulators" and the reverse: their "carrier" is actually the modulator.

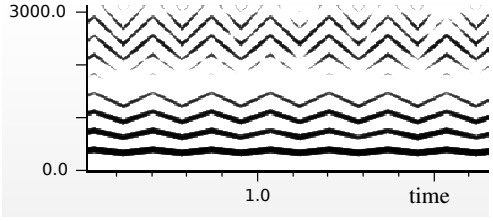
FM vocal synthesis implementations. Today, the most notable is the **FMVoice** instrument included in the Synthesis Tool Kit (STK) [6]. The class *FMvoices.cpp* can be freely downloaded as part of STK's source code and has been ported to various platforms e.g., Chuck [7] and Max / MSP / PeRColate [8]. In porting this class to Faust [9] and dealing with the discontinuity problem, I "rediscovered" for myself Le Brun's early solution.

3. NEW PROBLEM

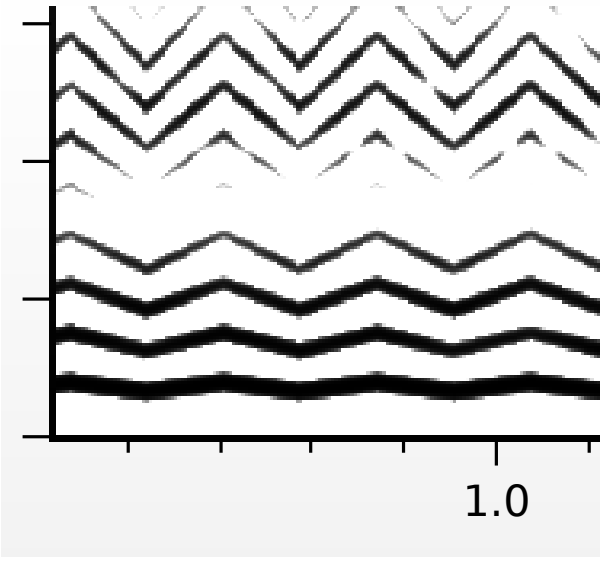
The discontinuity's clicks are gone (Fig. 2) but like sometimes happens, you think you've fixed something but the fix introduces a new problem. This next challenge is again an audible artifact plaguing vibrato. Not clicks, but a new kind of artifact – perfectly periodic vibrato should elicit perfectly periodic spectral modulation but in fact, doesn't. From one vibrato cycle to the next an overlayed pattern of spectral modulation can be heard. The problem arises from phase mismatches in the pair of formants (even and odd harmonic numbers) being mixed for each formant. These are the pair being cross-faded to combat the clicks in what I will now label as the "first-order problem."

The cross-fade technique assumes that the energy of all coincident pairs of spectral lines will sum arithmetically. However, this assumption does not take phase into account. A "second-order problem" is caused by phase interference of coincident spectral lines. These are the spectral lines (carrier and sideband frequencies) of the two overlapping formant generators which fill out the spectral envelope of the formant. They are identical spectra which are shifted relative to one another by one harmonic number. All phases are generated relative to their respective carrier oscillators rather than to the ensemble of frequencies as a whole. And phases of the carriers are arbitrary in time since they are independently determined by control changes.

As discussed in Sec. 2 the first-order artifact is only apparent under changing conditions of pitch and phoneme target. Similarly, the second-order effect may remain unnoticed under steady-state conditions. With no change to carrier frequencies, carrier phases will also be constant and so will the resultant spectral mix. Nevertheless, interference between unrelated sets of phases can have an effect which alters the static spectral envelope and is perceived as a quality mismatch away from a target steady-state vowel. The problem becomes more apparent when carrier frequencies are being shifted dynamically, especially if these changes are happening periodically. Vibrato is a good way to "tickle" the problem. Spectral distortions which may be imperceptible under other conditions are easier to hear with control changes that are repeating. The ear can pick out the distortion effect as a kind of spectral "isorhythm" or aliased pattern which is superimposed. Vibrato with a given period will generate a longer-period pattern of spectral modulation as seen in Fig.3. If you study the region around 2 kHz, you will notice a pattern in which phase-related Moiré fringing is inscribed on the amplitudes of the actual partials of the sound.



(a) fringing pattern changing over several vibrato periods



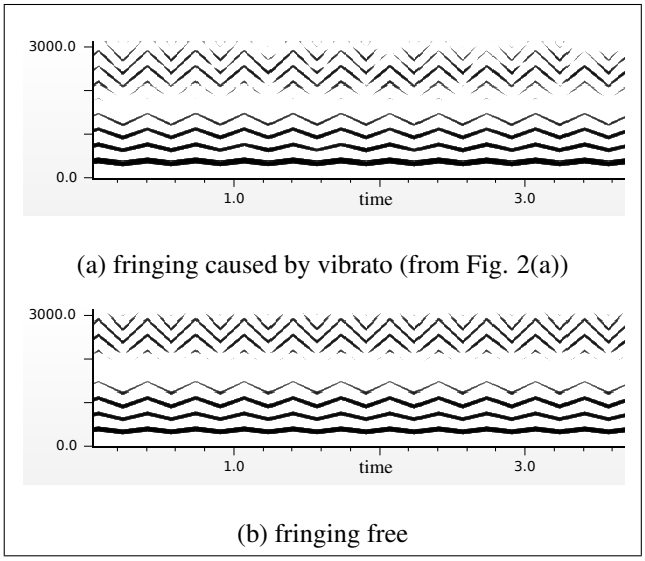
(b) fringing detail

Figure 3. Phase-related Moiré fringing, zoomed in from Fig. 2(a). When either of the two cross-faded carriers resets its frequency because the harmonic ratio needs to shift, it also resets its phase with respect to the other carrier. The result is a time-varying Moiré fringing effect visible in spectrograms.

3.1 Minimizing Fringing

An initial attempt to minimize the audible effect of phase fringing is worth mentioning even though it isn't ultimately the solution being adopted. It exploits the fact that phase interference is most notable when the cross-fade mix of the two carriers approaches equal portions (when interaction will be greatest). This is the point at which the carriers are equidistant from the target center frequency. Conversely, the least interference occurs when one of them is closest to the target and the other is essentially muted. Taking advantage of this proportion where one oscillator dominates, by expanding its time in the (vibrato-related) duty cycle, is one way to minimize fringing.

In listening tests, it was found that the cross-fade ramp can be made non-linear and still mask the first-order discontinuity perfectly. By using a power law for the ramp



(a) fringing caused by vibrato (from Fig. 2(a))

(b) fringing free

Figure 4. Re-rendering the vibrato example from 2(b) with UPHO's eliminates the fringing artifact.

slope, fringing is reduced by causing less time to be spent in the portion of the duty cycle with the problematic mix ratio. Initial experiments under periodic vibrato conditions indicated that even a very significant exponent can be used e.g., $f(x) = x^7$ and still mute the first-order artifact smoothly enough to avoid a click. This greatly reduces the time spent in "phase interference mode" and all but eliminates the audible effect of the second-order problem.

This fix has its drawbacks. Some fringing still remains during the portion of the duty cycle where the cross-fade mix briefly crosses through the equal portion region. More significantly though, is that altering the temporal dynamics of the mix distorts the mix away from the target i.e., away from the mix which best approximates the intended formant center frequency.

3.2 A Better Way

The root of the problem is in the use of independent oscillators. The way around this is to employ a bank of linked oscillators of a novel type called "uniform phase harmonic oscillator" (UPHO's). The bank can be constructed with any number of outputs and all will be tapped off of a single common phasor.

The familiar sampled sinusoid generates the fundamental (pitch) frequency signal for the bank to be constructed:

$$x(t) = Asin(\omega t + \phi) \tag{1}$$

$$\omega = rad/sec \tag{2}$$

$$= 2pif \tag{3}$$

where A is oscillator amplitude and f is frequency.

Expressed in pseudo-code, Eq. 1 can be implemented with the modulo function:

```
w = f / SR
mp = 0.0
for i = 0 to N
   y[n] = a * sin(2pi * mp)
   mp = (mp + w) mod 1.0
end
```

The constant SR is the sample rate and the variable mp is the fundamental's instantaneous phase.

The key to the next step is sharing the instantaneous phase mp with any other oscillators, where o specifies oscillator number, cp_o its instantaneous phase and h_o its harmonic number:

```
cp[o] =  mp mod (1/h[o])
y[o][n]
    = h[o] * a[o] * sin(2pi * cp[o])
```

and since we're interested in doing FM

```
m[o] = y[n] * i[o]
cp[o] =  mp mod (1/h[o])
y[o][n]
    = h[o] * a[o]
        * sin(2pi * cp[o] + m[0])
```

The above pseudo-code implements one FM pair consisting of an independent carrier and shared modulator which produces a formant centered at harmonic h of pitch frequency f with modulation index i. The latter coefficient determines formant bandwidth and is typically used in a low range (< 2.0). Phase depends on the master (shared) phasor mp and cannot be affected by h. In practice, a bank of six (or more) oscillators of this kind will be used to generate a vocal sound. These will create phonemes of 3 (or more) formants represented by a time-varying distribution of h, a, and i coefficients.

4. CONCLUSION

The completed glitch-free method consists of Chowning FM singing voice + Le Brun cross-fade algorithm (from Sec. 2) + UPHO oscillator bank (from Sec. 3). Fig.4(b) displays a spectrogram of vibrato rendered using the fully-realized solution. Classic phoneme table synthesis using Chowning's method can now be extended to arbitrary dynamic behavior.

Speech synthesis, with its widely varying pitch and phoneme transitions, provides a good "real-life" test of the revised technique. The test has been created with a "toy" analysis – resynthesis platform driving synthesis from digitized singing and speech. The formant tracking analyzer is written in Chuck and the UPHO-based formant synthesizer is a Chuck UGen (unit generator) written in Faust. The analysis portion is FFT-based and uses a relatively long (4096 sample) window for formant accuracy (at 48 kHz sample rate). An example speech input fragment and the method's resynthesized output are compared in the spectrograms of Fig.5. Signal coding in this version consists

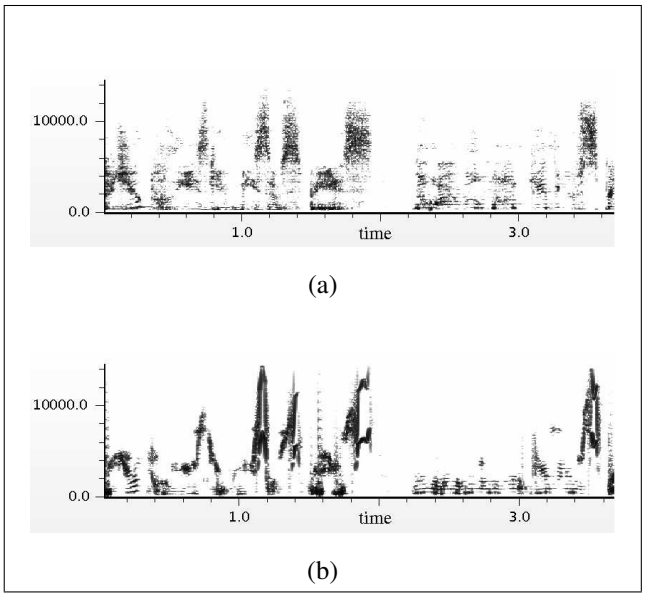


Figure 5. Analysis – resynthesis of dialog: (a) is the input from an argument between a teenage daughter and her mother, "can't you please give me some space," and "no, I will not give you some space.", (b) FM resynthesis

simply of recording formant parameter updates which are relatively sparse (and could be greatly optimized). The results are promising for developing this into an FM-based speech coder – the example consists of two different speakers in a heated, emotional dialog. Their voices and identities are preserved, as is their expressive prosody and intelligibility. The analysis tracks populations of short-lived formants which in the example are limited to 4 at a time (using 9 oscillators total). This style of modeling formant behavior has advantages over classical phoneme table modeling for capturing articulations and prosody. One goal of the analysis system going forward is to create a large database of acquired vocal sounds in order to structure more complex phoneme-based singing synthesis from a vastly expanded table.

The present synthesis method can be used for non-vocal sounds whose acoustic structures are also represented with formant-like resonances. Horner has explored timbre matching for a sampled trumpet using a genetic algorithm to find suitable FM formant parameters [10].

The two improvements to FM vocal synthesis detailed in this paper can be extended to other audio rate modulation schemes, in particular those which also employ single modulator / multiple carrier structures. A glitch-free AM vocal synthesis "cousin" has also been implemented in Faust. AM has the advantage of simplicity in prediction of dynamic sideband behavior (AM sidebands are free of the Bessel function which determines FM sidebands).

Acknowledgments

Many thanks to John Chowning for his inventions and encouragement, musical and technical. Bill Schottstaedt continues to passage into the future comprehensive sets of synthesis instruments and analysis tools. His Snd project preserves and provides essential computer music algorithms without which much of the present work would not have been possible.

5. REFERENCES

[1] J. Kleimola, "Nonlinear abstract sound synthesis algorithms," Ph.D. dissertation, Aalto University, Helsinki, Finland, 2013.

[2] J. Chowning, "Computer synthesis of the singing voice," in *Sound Generation in Winds, Strings, Computers*, J. Sundberg, Ed. Royal Swedish Academy of Music, 1980, pp. 4–13.

[3] ——, "Frequency modulation synthesis of the singing voice," in *Current Directions in Computer Music Research*, M. Mathews and J. Pierce, Eds. MIT Press, 1989, pp. 57–64.

[4] M. L. Brun, "Digital waveshaping synthesis," *J. of Audio Eng. Soc.*, vol. 27, no. 4, pp. 250–266, 1979.

[5] "Scheme, Ruby, and Forth Functions included with Snd," last viewed 29 Mar. 2013. [Online]. Available: https://ccrma.stanford.edu/software/snd/snd/sndscm.html

[6] "FMVoices Class Reference, in The Synthesis ToolKit in C++," last viewed 29 Mar. 2013. [Online]. Available: https://ccrma.stanford.edu/software/stk/

[7] "Chuck : Strongly-timed, concurrent, and on-the-fly audio programming language," last viewed 29 Mar. 2013. [Online]. Available: http://chuck.cs.princeton.edu/

[8] "PeRColate, A collection of synthesis, signal processing, and image processing objects for Max/MSP," last viewed 29 Mar. 2013. [Online]. Available: http://music.columbia.edu/percolate/

[9] "FAUST (Functional Audio Stream)," last viewed 29 Mar. 2013. [Online]. Available: http://faust.grame.fr/

[10] J. B. A. Horner and L. Haken, "Machine Tongues XVI: Genetic algorithms and their application to FM matching synthesis," *Computer Music J.*, vol. 17, no. 4, pp. 17–29, 1993.