Formant-based audio synthesis using nonlinear distortion

Miller Puckette

April 16, 1996

Abstract

An audio synthesis technique ("Phase Aligned Formant synthesis") is presented which is aimed at real-time musical applications. A desired sound is specified in terms of one or several time-varying formants, each with specified center frequencies, bandwidths, and amplitudes. The sound produced may be periodic or noisy. The relative merits of other known real-time techniques for synthesizing sounds with desired formants are also discussed. As an example, a spoken word is analyzed and resynthesized.

1 Introduction

Designing an instrumental voice for a synthesizer often involves planning how the various partials of the sound will vary in frequency and amplitude over the life of a note. In many natural sounds the timbre is best described in terms of the spectral envelope, which gives a pitch-independent specification of the evolution of a sound's spectrum in time. This is especially true of the spoken or sung voice, whose spectral evolution determines (among other things) the phoneme or phonemes being uttered or sung. The changing spectra of speech and other sounds are crucial to our perception of them.

Sounds—especially but not exclusively speech—are often described in terms of their fundamental pitch and several *formants*, or peaks in the spectral envelope. A formant's contribution to the perceived timbre of the sound is roughly determined by the formant's peak strength, center frequency, and bandwidth. One can approximately describe a sound's spectral envelope by locating its most important peaks and representing them as formants.

In most sounds, the spectral envelope changes dynamically, and to synthesize them it is not enough merely to be able to create a sound with a static spectrum, but rather one must be able to change the spectral envelope over time. For example, a short-term Fourier analysis of a spoken word is shown in Figure 1. As the sound's spectrum changes over time, the number of important spectral peaks varies, as do their center frequencies, bandwidths, and amplitudes. That certain of these formants are not voiced is invisible in the figure but clearly perceived in the sound. A realistic resynthesis must imitate the evolution of the spectrum and also take this voiced/unvoiced distinction into account.

Many attempts have been made to synthesize the voice and other sounds from a knowledge of how their formants change in time. To date, none has been proposed which is perfectly suited for real-time musical performance; either the data must be prepared in a way which requires a great deal of pre-calculation

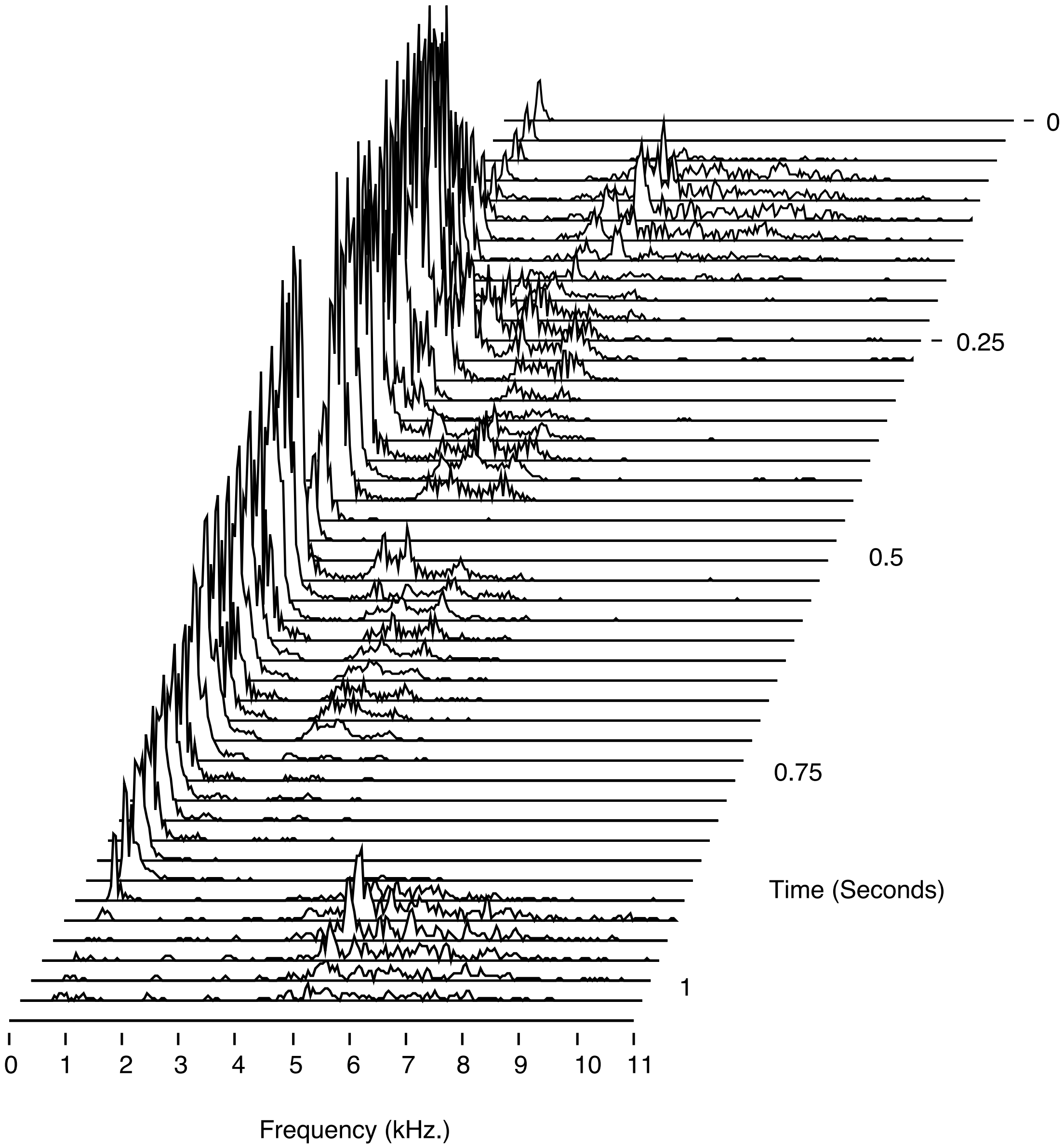

Figure 1: A 3d spectral plot of a recorded word of speech.

(and hence one can't make decisions on-the-fly as a result of real-time inputs), or else the sound quality has not been high enough for musical use.

The PAF (*Phase-Aligned Formant*) generator, proposed here, is an inexpensive method for generating sounds with a desired pitch and set of formants. The PAF is well adapted to real-time musical use because of its relative ease of computation, even in real-time situations where the formants and/or pitches required are not known in advance.

1.1 Known methods for generating complex spectra

Until 1973, spectra were usually synthesized using either *additive* [1] or *subtractive* synthesis [2]. In additive synthesis, a separate sinewave oscillator is used for each partial of the sound; thus the amplitudes and frequencies of all partials may be independently controlled. The main drawback to additive synthesis is the expense of dedicating oscillators to all the partials (there can easily be more than 100 in a single sound) and of computing, on the basis of some simpler specification, the desired amplitudes and frequencies of all the partials.

In subtractive synthesis, a broadband signal is filtered, often using a simple bandpass filter. Subtractive synthesis is less expensive than additive synthesis, but has three serious shortcomings. First, the phases of the partials of a signal are changed by filtering (unless one uses phase-linear filtering, but that would be very expensive in this context). Thus, if one adds the outputs of two filters their spectra do not superimpose in an easily predictable or controllable way. One could alternatively try to compound the effects of several filters by applying them in series, but in that case it is hard to calculate the coefficients required in the individual filters to give a desired final result.

A second problem posed by subtractive synthesis is that of predicting the amplitude of the output of a filter, especially in transient situations, but even in the steady state. In general, a filter whose bandwidth is smaller than the spacing of the partials of a sound will have a larger output when its center frequency is aligned with a partial than otherwise. If the filter coefficients are time-varying the problems are compounded.

The third problem is numerical accuracy. The difficulties of obtaining numerically accurate outputs from recursive filters are well documented elsewhere [3] and need not be detailed here.

The introduction of the Frequency Modulation technique for sound synthesis [4] marked the first inexpensive and numerically tractable digital technique for generating complex, dynamically-changing spectra. In its simplest form, the FM technique is to calculate samples of the function,

$$F(t) = \sin(\omega_c t + x \sin(\omega_m t)), \tag{1}$$

for discrete values of the time t. The parameter x, called the *index of modulation*, as well as the carrier and modulating frequencies ω_c and ω_m, may be varied in time to change the spectrum. The spectrum itself may be found from the Bessel function identity,

$$F(t) = \sum_{k=0}^{\infty} \mathrm{J}_{k+1}(x) \sin((\omega_c + k\omega_m)\, t) + \sum_{k=1}^{\infty} (-1)^k \mathrm{J}_{k+1}(x) \sin((\omega_c - k\omega_m)\, t), \tag{2}$$

that is, the sound has components of frequency $\omega_c \pm k\omega_m$ and amplitude $\mathrm{J}_{|k+1|}(x)$. As the index is changed, the various components increase and decrease in strength. One FM-generated sound with a steadily increasing index was analyzed to give the spectrum shown in Figure 2. It can be very hard to obtain a desired spectrum out of elementary spectra such as this one.

Since the original appearance of FM many variations and generalizations have been proposed, such as including more than one modulating sine wave in Equation 1, or considering functions of the form $f(xg(\omega_0 t))$ where f and g are arbitrary functions calculated by wavetable lookup and x is an index of modulation [5]. The results to date have all suffered from the unmanageable spectra that arise.

At least two other synthesis techniques have been proposed which allow direct specification of center frequency and bandwidth: VOSIM [6], and the FOF [7]. The VOSIM and FOF synthesis methods address the second and third problems encountered in subtractive synthesis but not the first. Both VOSIM and the FOF have outputs with spectra which are more complicated than what may be obtained using subtractive synthesis, and with harder-to-predict phases. In addition, the FOF poses an additional problem: its computational expense can vary without bound depending on its synthesis parameters, complicating

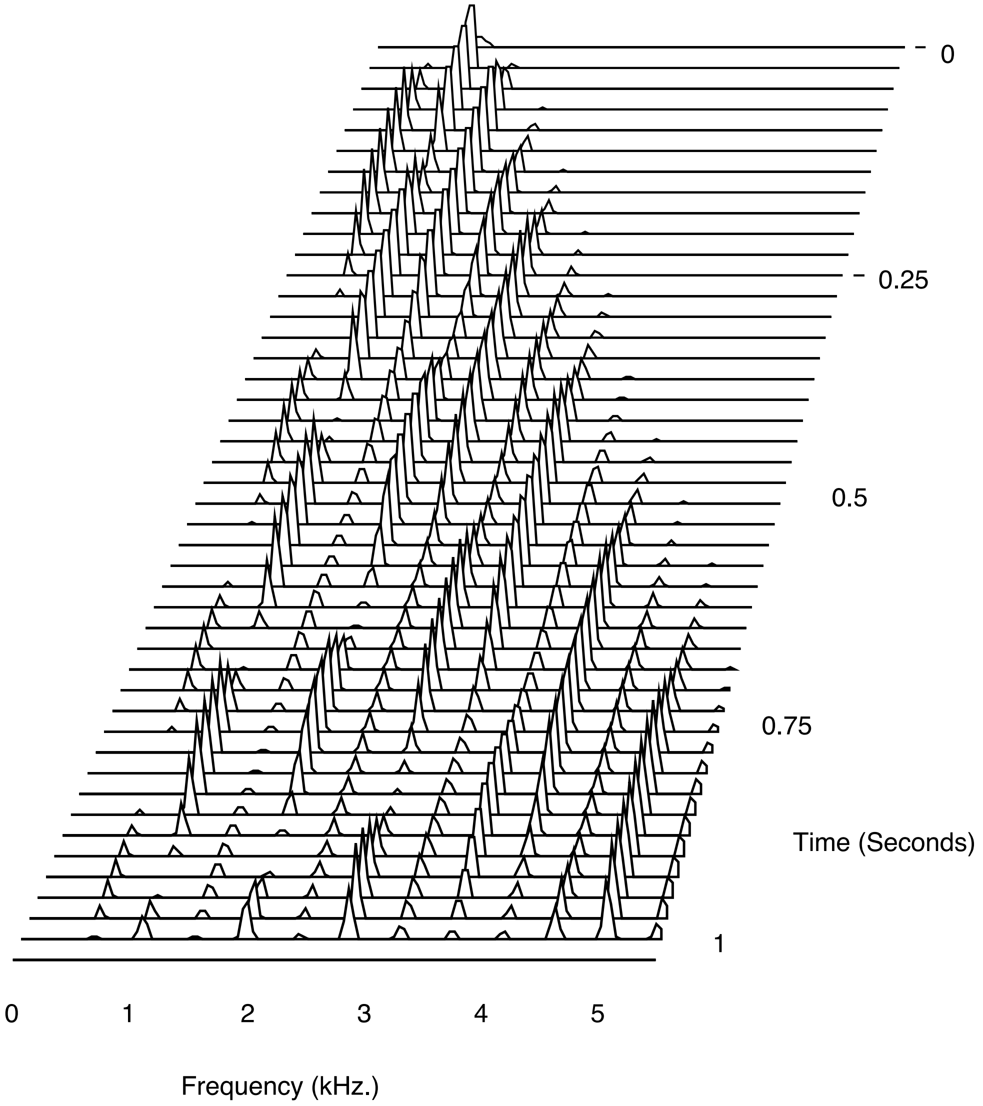

Figure 2: Analyzed output from the FM synthesis technique, showing the complicated evolution of the formants.

the task of adapting it for real-time synthesis.

1.2 Definition of the PAF

The PAF algorithm is proposed as a synthesis technique which satisfies all the criteria mentioned above: efficiency of computation, good numerical behavior, and an easily described spectrum whose phases may be controlled to allow easy superposition of simple spectra into desired ones. The PAF's timbral parameters are simply the desired center frequency and bandwidth; these can easily be changed over time with stable and predictable results.

The PAF algorithm numerically computes samples of the complex-valued function of time,

$$X(t) = \sum_{k=-\infty}^{\infty} e^{i(k\omega_0+\omega_s)t-\frac{|(k\omega_0+\omega_s)-\omega_c|}{\delta}} \tag{3}$$

where ω_0 is a fundamental frequency in radians per second, ω_s is a frequency shift, ω_c the center frequency of a formant in radians per second, and δ its bandwidth. Rewriting Equation 3 as,

$$X(t) = \sum_{k=-\infty}^{\infty} \left(\cos((k\omega_0 + \omega_s)t) + i\sin((k\omega_0 + \omega_s)t)\right) e^{-\frac{|(k\omega_0+\omega_s)-\omega_c|}{\delta}} \tag{4}$$

shows that the real and imaginary parts of $X(t)$ can be regarded as sums of cosine and sine waves, respectively, whose angular frequencies are all of the form $k\omega_0 + \omega_s$ and whose amplitude for the partial with angular frequency ω is given by,

$$e^{-\frac{|\omega-\omega_c|}{\delta}}. \tag{5}$$

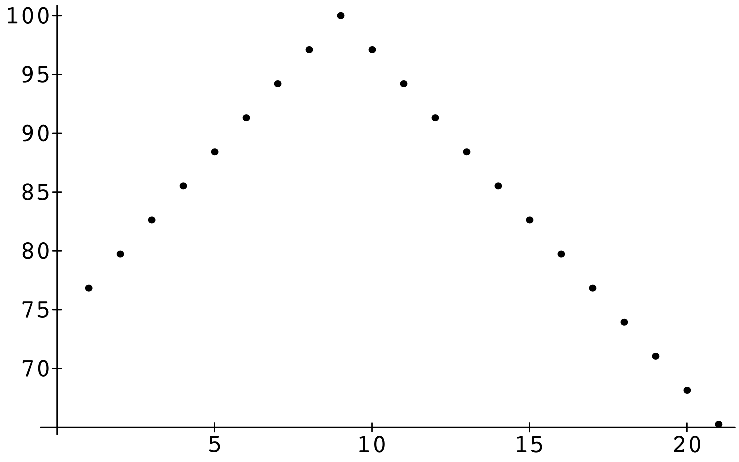

Figure 3: Component strengths for a PAF with $\omega_c = 8\omega_0$ and $\delta = 3\omega_0$. The horizontal axis is the number of the partial; the vertical axis is in DB.

If we choose for example $\omega_c = 8\omega_0$, $\omega_s = 0$, and $\delta = 3\omega_0$, the spectrum in decibels (taking 100 dB to represent unit amplitude) is graphed in Figure 3. In practice we will usually evaluate only the real or the imaginary part of Equation 3. The spectra of the real and imaginary parts are obtained by adding or subtracting from the above spectrum, its reflection about $\omega = 0$, as shown in Figure 4.

Since the center frequency and bandwidth enter naturally as parameters, it is easy to vary them independently. For instance, sweeping the center frequency while holding the bandwidth constant can yield the result shown in Figure 5. Compared to the output of the FM synthesis technique shown in Figure 2, it is clear that the PAF's spectral evolution is much simpler. This greatly simplifies the problem of finding synthesis parameters to approach a desired sonic result.

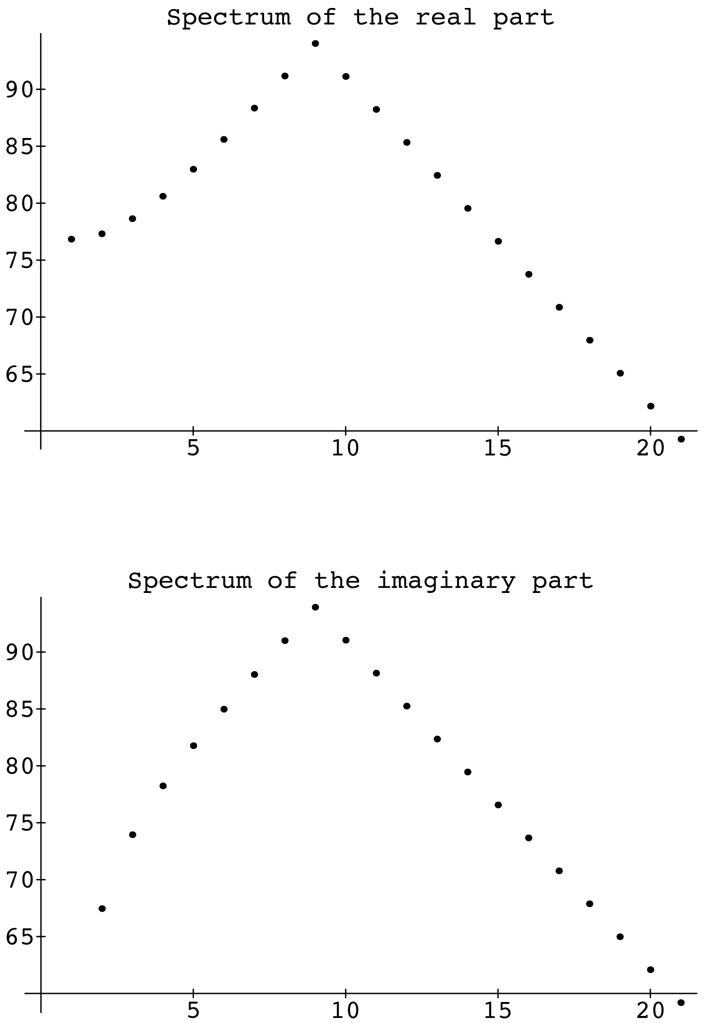

Figure 4: Component strengths for the real and imaginary parts of the PAF with the same parameters and coordinate axes as before.

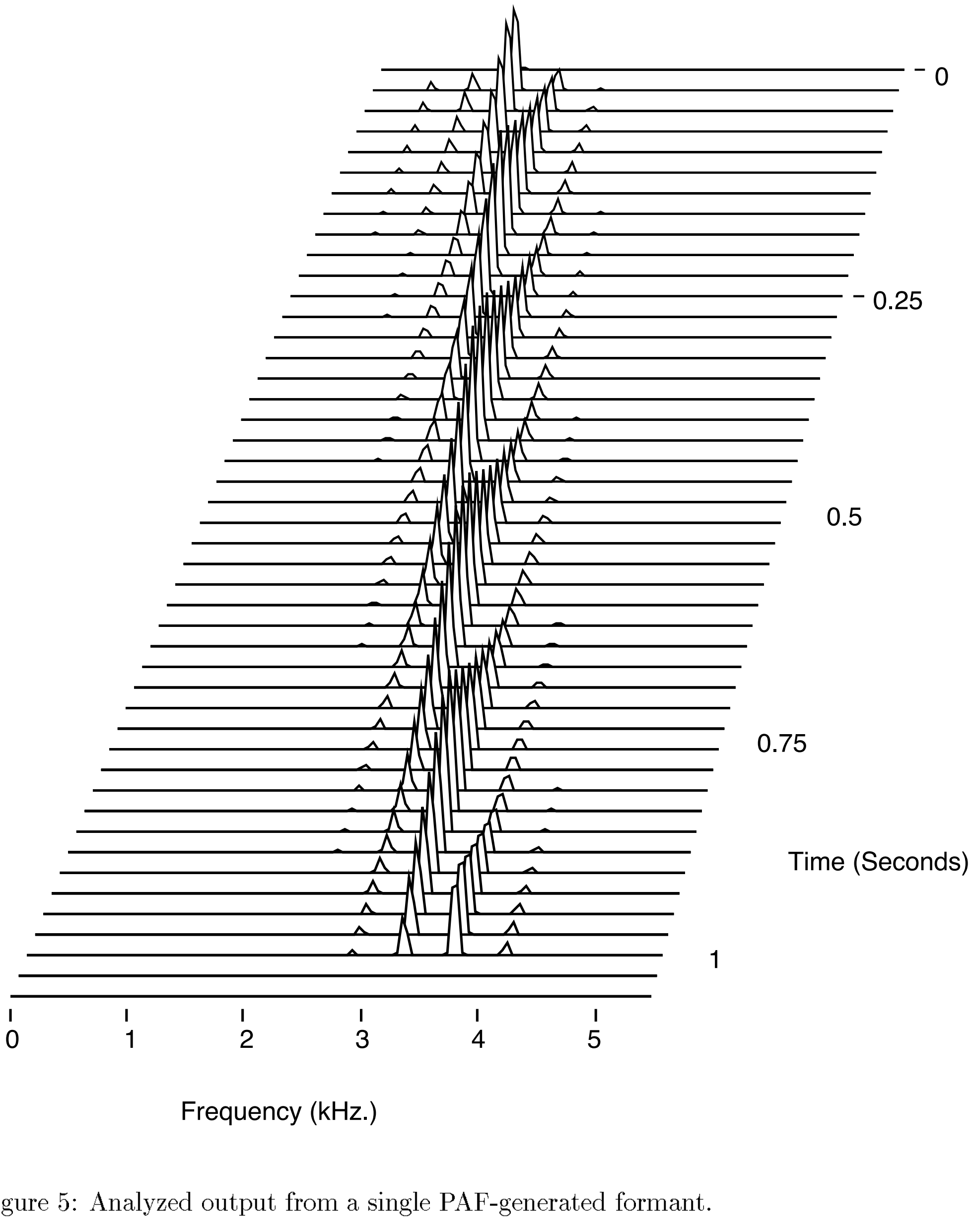

Figure 5: Analyzed output from a single PAF-generated formant.

2 Efficient computation of the PAF

In general, we will wish to compute values of $X(t)$ defined in Equation 3 for discrete values, $t = \{\tau, 2\tau, ...\}$, where τ is the sample period, and where the parameters ω_0, ω_c, ω_s, and δ, and functions of them, are regarded as "control signals" to be taken from envelope generators.

We start by considering the special case in which $\omega_s = 0$ and $\omega_c = n\omega_0$, where n is an integer. Equation 3 then becomes,

$$X(t) = \sum_{k=-\infty}^{\infty} e^{\omega_0\left(ikt - \frac{|k-n|}{\delta}\right)} \tag{6}$$

$$= \sum_{k=-\infty}^{\infty} e^{\omega_0\left(i(k+n)t - \frac{|k|}{\delta}\right)} \tag{7}$$

$$= C(t)M(t), \tag{8}$$

where

$$C(t) = e^{in\omega_0 t}, \tag{9}$$

$$M(t) = \sum_{k=-\infty}^{\infty} \left(ge^{i\omega_0 t}\right)^k, \tag{10}$$

$$g = e^{-\frac{\omega_0}{\delta}}. \tag{11}$$

We will refer to the quantity $C(t)$ as the *carrier* and $M(t)$ as the *modulator*. Following the example of [8], we rewrite the modulator as,

$$M(t) = 2\mathrm{Re}\left(\frac{1}{1 - ge^{i\omega_0 t}}\right) - 1 \tag{12}$$

$$= 2\mathrm{Re}\left(\frac{1 - ge^{-i\omega_0 t}}{1 + g^2 - 2g\cos(\omega_0 t)}\right) - 1 \tag{13}$$

$$= \frac{1-g^2}{1+g^2-2g\cos(\omega_0 t)}. \tag{14}$$

In [5] it is noted that this can be rewritten as,

$$M(t) = \frac{1-g^2}{(1+g^2)\left(1-\frac{2g\cos(\omega_0 t)}{1+g^2}\right)} \tag{15}$$

$$= p_1(t)s_1(x_1(t)\cos(\omega_0 t)), \tag{16}$$

where

$$s_1(z) = \frac{1}{1+z}, \tag{17}$$

$$p_1(t) = \frac{1-g^2}{1+g^2}, \tag{18}$$

$$x_1(t) = \frac{-2g}{1+g^2}. \tag{19}$$

Values of $M(t)$ can therefore be calculated efficiently using precalculated tables containing values of the cosine function and of the function s_1. (LeBrun actually proposes a slightly different, but equivalent, formula.) The computation requires two table lookups and two multiplications per output sample, plus two envelope generators to approximate the values $p_1(t)$ and $x_1(t)$.

However, this formula works badly in practice because of truncation error propagation. If we introduce a small truncation error ϵ to the calculated value of $\cos(\omega_0 t)$, and if we assume the worst case,

$$\cos(\omega_0 t) = 1, \tag{20}$$

the magnitude of the error introduced in the calculation of $M(t)$ will be approximately,

$$e \approx p_1(t){s_1}'(x_1(t))\epsilon. \tag{21}$$

If we take g close to 1, say $g = 1 - h$ where $h << 1$, the relative magnitude of the error becomes

$$\begin{aligned}
|(\frac{e}{M(t)})| &\approx |\frac{{s_1}'(x_1(t))}{s_1(x_1(t))}|\epsilon & (22)\\
&= \frac{\epsilon}{1 - \frac{2g}{1+g^2}} & (23)\\
&= \frac{\epsilon}{1 - \frac{2(1-h)}{2-2h+h^2}} & (24)\\
&\approx \frac{2\epsilon}{h^2}. & (25)
\end{aligned}$$

This relative error figure, based on the worst-case assumption (Equation 20), may be regarded as typical for small values of h, since it is just at the corresponding phase that the modulator attains a narrow peak. Since in practice h often descends to values of 0.1 and lower, applying the function s_1 can produce an error growth of 100 or more. This puts a high requirement of precision on the calculation of $\cos(\omega_0 t)$ and the ensuing arithmetic, as compared to other music synthesis techniques such as FM.

This difficulty is overcome by rewriting Equation 14 as,

$$\begin{aligned}
M(t) &= \frac{1-g^2}{(1-g)^2 + 2g\,(1 - \cos(\omega_0 t))} & (26)\\
&= \frac{1-g^2}{(1-g)^2 + 4g\sin^2(\frac{\omega_0 t}{2})} & (27)
\end{aligned}$$

$$= \frac{1+g}{(1-g)\left(1+\frac{4g\sin^2(\frac{\omega_0 t}{2})}{(1-g)^2}\right)} \tag{28}$$

$$= p(t)s(x(t)\sin(\frac{\omega_0 t}{2})), \tag{29}$$

where

$$s(z) = \frac{1}{1+z^2}, \tag{30}$$

$$p(t) = \frac{1+g}{1-g}, \tag{31}$$

$$x(t) = \frac{2\sqrt{g}}{1-g}, \tag{32}$$

with the same low cost of calculation as before.

Analyzing the error propagation in this calculation gives better results. The maximum absolute value of $s'(x)/s(x)$ is 1; thus no error growth takes place during the application of s.

2.1 The general case

The above calculations were based on the assumption that the center frequency ω_c was an integer multiple of the fundamental frequency ω_0, and $\omega_s = 0$. We now show how to calculate values of $X(t)$ in Equation 3 with an arbitrary center frequency and nonzero frequency shift. Choosing an integer n and $0 \leq a < 1$ so that

$$\omega_c - \omega_s = (n+a)\omega_0, \tag{33}$$

and setting $\gamma = \omega_0/\delta$, Equation 3 becomes,

$$X(t) = \sum_{k=-\infty}^{\infty} e^{i(k\omega_0+\omega_s)t-|(k-n-a)\gamma|}. \tag{34}$$

We wish to rewrite the quantity,

$$S = e^{-|(k-n-a)\gamma|}, \tag{35}$$

as a weighted sum:

$$T = ue^{-|(k-n)\gamma|} + ve^{-|(k-n-1)\gamma|}. \tag{36}$$

To do this, we note that for $k \leq n$, we have,

$$|\,(k-n-a)\,\gamma| = a\gamma + |\,(k-n)\,\gamma| = -b\gamma + |\,(k-n-1)\,\gamma|, \tag{37}$$

(where for brevity we take $b = 1 - a$), and thus,

$$T = \left(ue^{a\gamma} + ve^{-b\gamma}\right) e^{-|(k-n-a)\gamma|}. \tag{38}$$

In the same way, for $k > n$, we have,

$$T = \left(ue^{-a\gamma} + ve^{b\gamma}\right) e^{-|(k-n-a)\gamma|}. \tag{39}$$

Thus we will have $S = T$ for all integral k if we choose u, v so that,

$$ue^{a\gamma} + ve^{-b\gamma} = ue^{-a\gamma} + ve^{b\gamma} = 1. \tag{40}$$

While we could easily solve these equations exactly, it is simpler to approximate values of e^x by $1 + x$; in this approximation the solution is simply given by $u = b, v = a$. The approximation is closest for small values of g, which correspond to bandwidths that are several times the fundamental frequency. In practice, the approximation is an improvement over the exact result for smaller bandwidths, since the signal power of a swept formant becomes more nearly constant in time.

Equation 34 then becomes,

$$X(t) \approx b \sum_{k=-\infty}^{\infty} e^{i(k\omega_0+\omega_s)t-|(k-n)g|} + a \sum_{k=-\infty}^{\infty} e^{i(k\omega_0+\omega_s)t-|(k-n-1)g|} \tag{41}$$

$$= \frac{(1+g)\left(be^{i(n\omega_0+\omega_s)t} + ae^{i((n+1)\omega_0+\omega_s)t}\right)}{(1-g)\left(1+\frac{4g\sin^2(\frac{\omega_0 t}{2})}{(1-g)^2}\right)} \tag{42}$$

$$= p(t)s(x(t)\sin(\frac{\omega_0 t}{2}))\left(be^{i(n\omega_0+\omega_s)t} + ae^{i((n+1)\omega_0+\omega_s)t}\right). \tag{43}$$

This is the form in which the PAF is actually computed.

A possible generalization is to replace the waveshaping function, defined in Equation 30, by a sum of the form

$$\sum_{k=1}^{n} \frac{a_k}{1+c_k z^2}. \tag{44}$$

The resulting spectrum will thus be a sum of exponentials with different coefficients g. As a limiting special case of this, we can obtain the product of a polynomial by an exponential by taking the limit of the above as more than one of the c_k approach the same value. The possibilities raised by these alternate waveshaping functions have yet to be explored.

2.2 Making noisy formants

In the spoken and sung voice, and also in many acoustic instruments, parts of the spectrum at certain points in time are either partially or wholly "noisy," i.e., better approximated by spectrally enveloped noise than as a sum of equally spaced harmonics. This feature of natural sounds can be of great importance, and so in synthetic sounds, it is useful to be able to imitate it.

The simplest, and perhaps the most effective, way to use the PAF to generate noisy signals is not really specific to the PAF at all: it consists of providing a post-processor which modulates its input with band-limited noise. In practice, a good result can be obtained using the network shown in Figure 6.

3 Realization

A single PAF may be generated in hardware or in software following the block diagram shown in Figure 7. Here the operations "cos", "sin", and "s" are the cosine and sine functions and the waveshaping function of Equation 30, with the appropriate input normalizations; they may be evaluated using table lookup, either with or without interpolation. If more than one PAF is to be used to construct a multi-formant structure, they should be combined as in Figure 8, which shows a two-formant PAF configuration. Figure 8 also includes a noise post-processor configured to give independent control over the noisiness of each formant.

3.1 Control issues

The parameters n, a, b, x, and p in the block diagram may not be changed discontinuously without causing audible clicks in the output. The values of p and x may be ramped up and down, but the values of n, a, and b cannot be

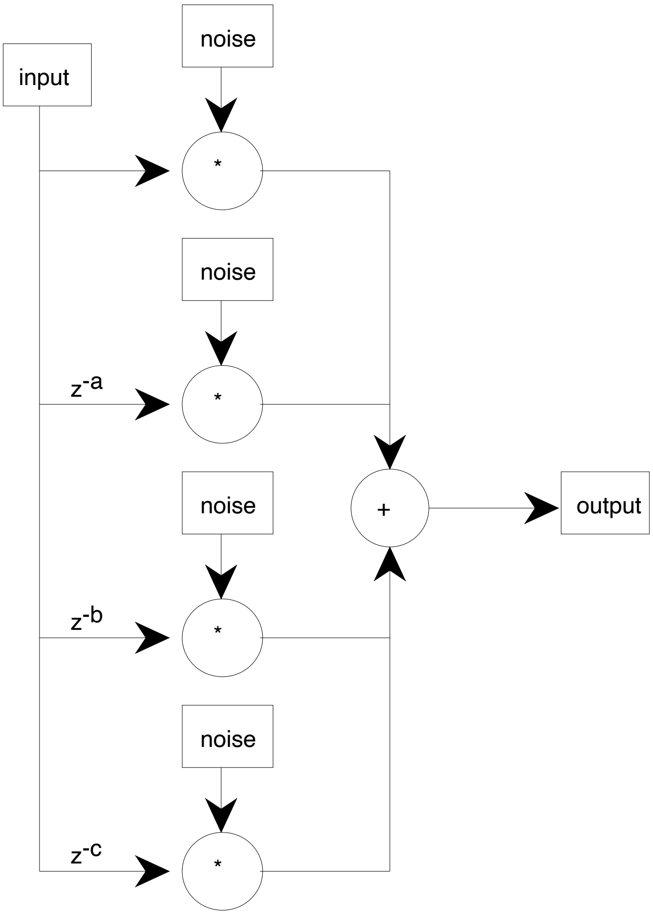

Figure 6: A signal processing network to realize a noise post-operator.

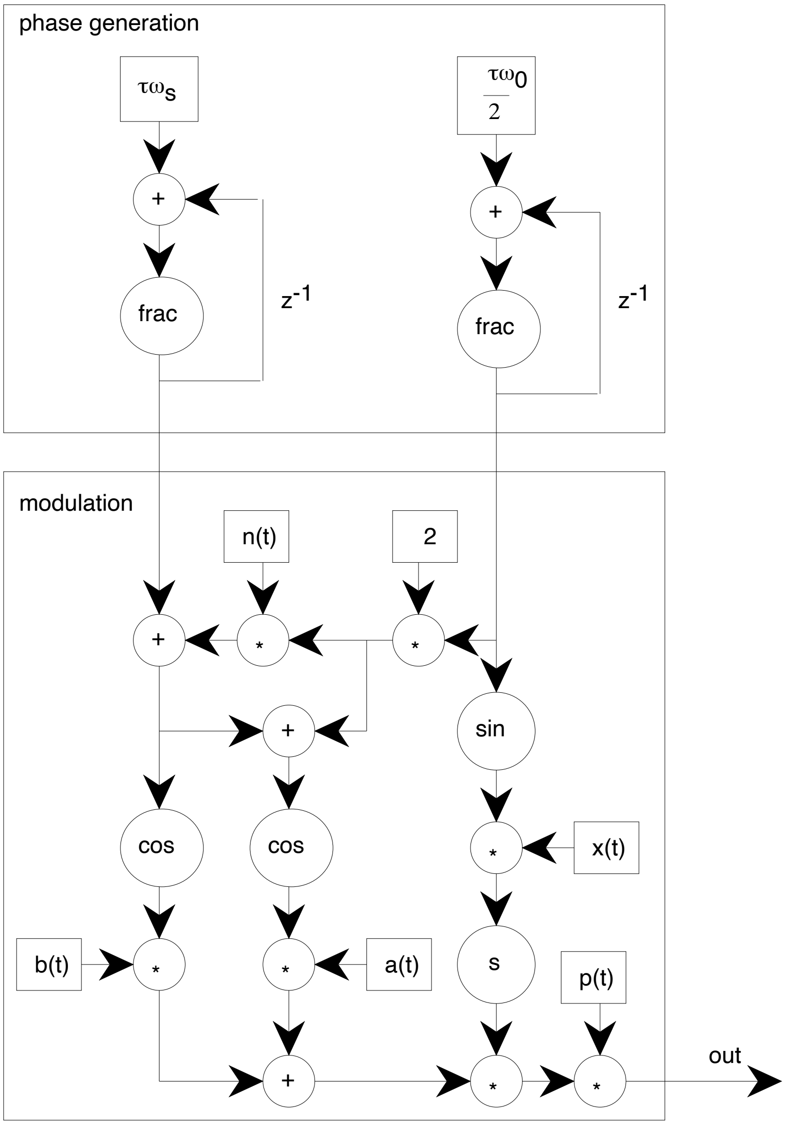

Figure 7: Block diagram of the PAF generator.

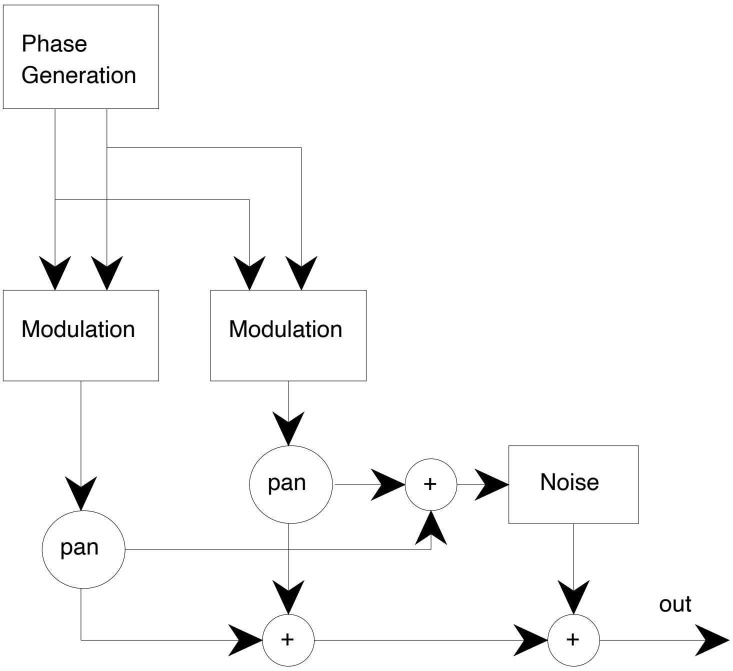

Figure 8: Two phase-locked PAFs and the noise modulation post-operator.

treated so simply; since n is an integer it cannot be changed continuously. If the center frequency ω_c or the fundamental ω_0 is swept, a and b can oscillate rapidly up and down, with an unpleasant effect on the sound. The solution is to change n, a, and b discontinuously at zero crossings of the phase, i.e., at samples where $\omega_0 t$ crosses a multiple of $2pi$, where the value shown in Equation 43 is independent of n, a, and b.

A subtlety arises for low values of ω_0, say below 120 radians per second: if a rapid change is desired in any parameter, the widely-separated parameter updates may sound like a discrete sequence instead of as a continuous change. Also, the reaction time can be unacceptably large since nothing can be changed until the next period. In these cases it is sometimes best to ramp the gain quickly to zero, effect the desired update, and ramp the gain back up.

3.2 Example

The recorded voice shown in Figure 1 was resynthesized using six PAF formants. The formants themselves were entered by hand using the Explode editor [9]. Resynthesis resulted in the spectrum shown in Figure 9, which is visibly similar to the original. The deviations are partly due to the fact that the formants were altered to make the word sound more natural, even if the resulting spectrum diverged from the original.

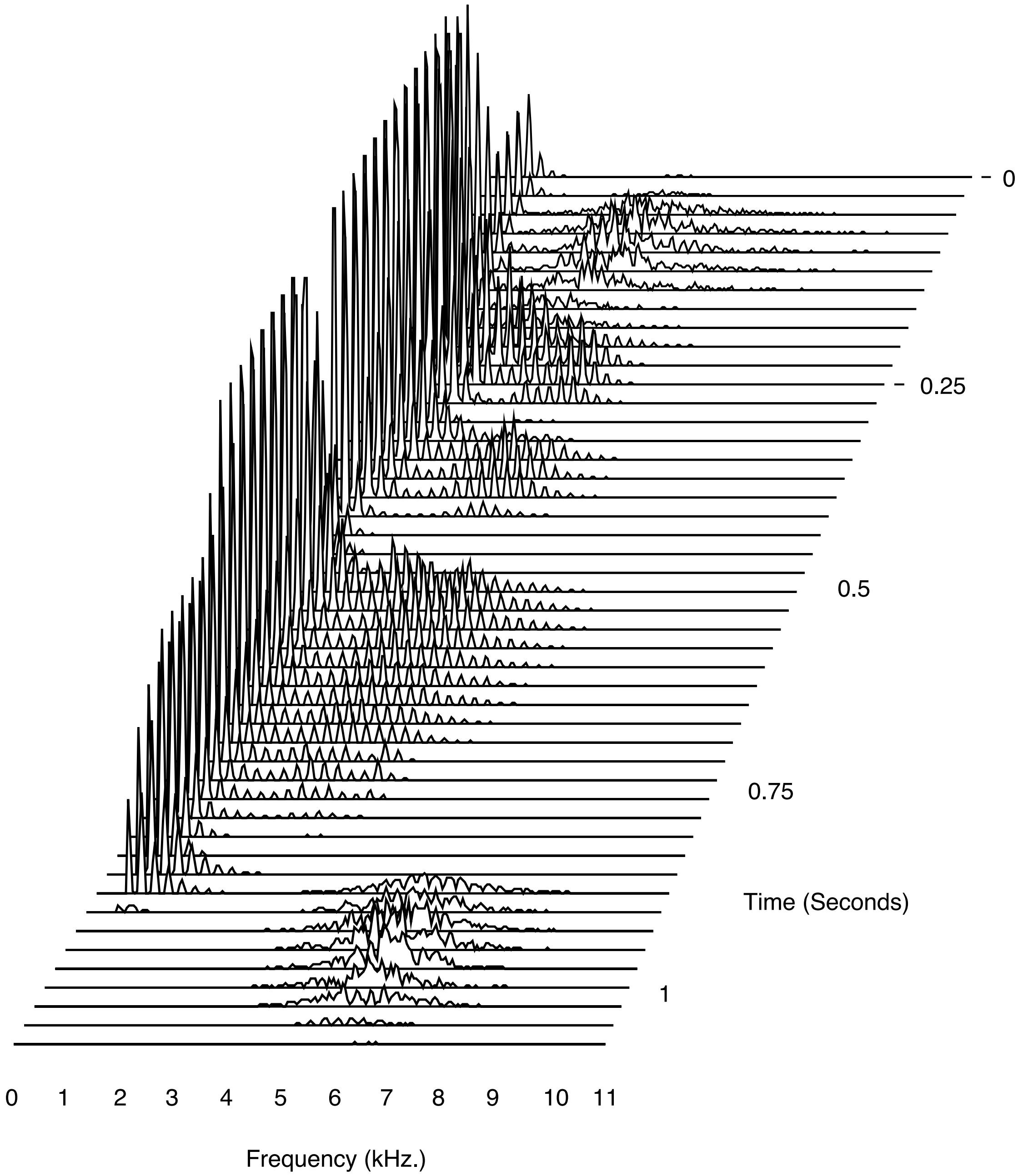

Figure 9: 3d spectral plot of the resynthesized sound.

4 Conclusion

The PAF generator has been implemented in real-time on the IRCAM Signal Processing Workstation [10] and a patent has been applied for in France. In the meantime, three pieces of music using the PAF generator have been produced at IRCAM and more will doubtless follow. The first one, Philippe Manoury's *La Partition du Ciel et de l'Enfer* for orchestra and live electronics (1988), was recently conducted by Pierre Boulez in Carnegie Hall [11].

The list of published audio synthesis techniques is long; perhaps only half a dozen of them have become standard techniques in computer music. It is too early to say whether the PAF will become part of this canon or not. The PAF's unique feature is the ease with which it can be used to generate sounds with specified time-varying spectral envelopes. The PAF will succeed if this ability proves relevant in a wide enough range of musical situations.

5 Acknowledgements

This work was carried out while the author was a member of the research staff at IRCAM (l'Institut de Recherche et Coordination Musique/Acoustique.) The author is indebted to Philippe Manoury who was the first to explore the PAF generator's musical potential. Stefan Bilbao, Zack Settel, Cort Lippe, Leslie Stuck, and Jonathan Bachrach have all contributed to the realization of the PAF generator within IRCAM's music production environment.

References

[1] Dodge, C. and Jerse, T., 1985. *Computer Music*, Shirmer (New York), pp. 78-79.

[2] *ibid.*, pp. 155-194.

[3] Rabiner, L., *et al.*, 1974. "Some comparisons between FIR and IIR digital filters," *Bell System Technical Journal* 53, pp. 305-331.

[4] Chowning, J., 1973. "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society* 21/7, pp. 526-534.

[5] LeBrun, M., 1979. "Digital waveshaping synthesis," *Journal of the Audio Engineering Society* 27/4, pp. 250-266.

[6] Templaars, S., 1977. "The VOSIM signal spectrum," *Interface* 6, pp. 81-96.

[7] Rodet, X., Potard, Y., and Barriere, J.-B., 1984. "The CHANT project: from the synthesis of the singing voice to synthesis in general." *Computer Music Journal* 8/3, pp. 15-31.

[8] Moorer, J. A., 1976. "The synthesis of complex audio spectra by means of discrete summation formulae," *Journal of the Audio Engineering Society* 24/8, pp. 717-727.

[9] Puckette, M., 1990. "EXPLODE: a user interface for sequencing and score following," *Proceedings*, International Computer Music Conference, pp. 259-261.

[10] Lindemann, E. et al. 1991. "The Architecture of the IRCAM Music Workstation." Computer Music Journal 15(3), pp. 41-49.

[11] Holland, B., 1993. "Boulez musters his high-tech Parisian forces," *New York Times* Monday, Nov. 15, p. B3.