Real-Time Implementation of Asymmetrical Frequency-Modulation Synthesis*

B. T. G. TAN AND S. L. GAN

Department of Physics, National University of Singapore, Singapore 0511, Republic of Singapore

Asymmetrical frequency modulation (AFM) as proposed by Palamin et al. (1988) offers greater flexibility in the shaping of the spectra of synthesized waveforms as compared to normal frequency-modulation (FM) synthesis. The feasibility of using AFM synthesis for the real-time generation of waveforms of musical instruments is demonstrated using a DSP56000 digital signal processor to implement the AFM algorithm.

0 INTRODUCTION

The discovery by Chowning in 1973 that frequency modulation (FM) could be applied to the synthesis of complex sounds was a major advance in musical sound synthesis [1]. FM synthesis has since then proven to be a powerful and flexible method of musical sound synthesis [2]–[5]. In the simplest FM synthesis mode, the frequency of a sine wave (the carrier) is modulated by another sine wave (the modulator) to produce a complex waveform whose spectral characteristics depend on the characteristics of the two sine waves and the degree of modulation.

The basic equation for FM is

$$x(t) = A \sin [\omega_c t + I \sin (\omega_m t)] \tag{1}$$

where

A = maximum amplitude
ω_c = angular carrier frequency, $= 2\pi f_c$, f_c being the carrier frequency
ω_m = angular modulator frequency, $= 2\pi f_m$, f_m being the modulator frequency
I = modulation index
t = time

When expressed in Fourier series form, it can be seen that the amplitude is governed by the Bessel functions of the first type $J_n(I)$ as follows:

$$x(t) = A \sum_{n=-\infty}^{\infty} J_n(I) \sin (\omega_c t + n\omega_m t) . \tag{2}$$

1 ASYMMETRICAL SPECTRA FREQUENCY MODULATION

Although the FM method is very powerful, the degree to which the spectral envelope of a synthesized waveform can be modified is limited by the intrinsic envelope of the symmetrical sidebands of the FM spectrum. Palamin et al. have shown [7] that with a simple modification to Chowning's FM synthesis equation, it is possible to introduce and control spectral sideband asymmetry easily and thereby increase greatly the control of the shape of the spectral envelope of the synthesized waveform.

The FM spectral equation, Eq. (2), may be expressed as follows:

$$x(t) = A \sum_{n=-\infty}^{\infty} J_n(I) \sin (\alpha + n\beta) . \tag{3}$$

where $\omega_c t = \alpha$ and $\omega_m t = \beta$ for convenience.

Palamin et al. introduce a new parameter which modifies the spectral amplitudes in Eq. (3) as follows:

$$x(t) = A \sum_{n=-\infty}^{\infty} r^n J_n(I) \sin (\alpha + n\beta), \qquad r \neq 0 . \tag{4}$$

* Manuscript received 1992 September 24; revised 1993 February 18.

Each harmonic with amplitude $J_n(I)$ is to be multiplied by r^n. The Bessel functions decrease rapidly as their order increases, thus ensuring that the spectrum converges.

2 SYNTHESIS ALGORITHM

Palamin et al. derive the expression for the synthesis of the modified FM waveform which can be expressed thus:

$$x(t) = A \exp \left[\frac{I}{2} \left(r - \frac{1}{r} \right) \cos (\omega_{\mathrm{m}} t) \right] \times \sin \left[\omega_{\mathrm{c}} t + \frac{I}{2} \left(r + \frac{1}{r} \right) \sin (\omega_{\mathrm{m}} t) \right] . \quad (5)$$

It should be noted that when r is set to 1.0, Eq. (5) will be reduced to Chowning's simple FM, as expressed in Eq. (2).

To simplify our discussion, let us call this modified FM synthesis technique asymmetrical frequency modulation (AFM).

As can be seen in Eq. (5), the amplitude of the signal is amplitude modulated. Since $r^{2n} + r^{-2n} > 2$ for any $r \neq 1$ and any $n \neq 0$, the introduction of r in Eq. (3), which becomes Eq. (4), increases the power in the spectrum. To normalize the signal with respect to the power, each harmonic component of Eq. (4) should be multiplied by the factor

$$N = \frac{1}{\sqrt{J_0' \, [I(r - 1/r)]}} \quad (6)$$

where J_0' represents the modified Bessel function of the first kind and of order 0.

Palamin et al. propose that the normalization can be made simpler by including the logaritham of the normalization factor in the exponential amplitude term:

$$A \left\{ \frac{I}{2} \left(r - \frac{1}{r} \right) \cos \omega_{\mathrm{m}} t - \frac{1}{2} \ln \left(J_0' \left[I \left(r - \frac{1}{r} \right) \right] \right) \times \sin \left[\omega_{\mathrm{c}} t + \frac{I}{2} \left(r + \frac{1}{r} \right) \sin \omega_{\mathrm{m}} t \right] \right\}$$

$$= A \frac{1}{\sqrt{J_0' \, [I(r - 1/r)]}} \sum_{-\infty}^{+\infty} J_n(I) r^n \sin (\omega_{\mathrm{c}} t + n \omega_{\mathrm{m}} t) . \quad (7)$$

In this way the multiplication usually required for normalization is converted into a computationally simpler summation.

3 IMPLEMENTATION

We have implemented the AFM algorithm for real-time digital sound synthesis using a standard digital signal processor (DSP) integrated circuit to demonstrate the feasibility of AFM in practical applications such as keyboard synthesizers. With the Apple Macintosh IIci as a host computer, we used the Digidesign Sound Accelerator Card, which had a Motorola DSP56000 chip as its DSP. The DSP56000 was used to generate the required AFM waveform in real time by computing two AFM operators. The sum of the two operators was converted by a digital-to-analog converter to an analog signal which could be amplified, monitored, and analyzed (Fig. 1).

The AFM program is separated into two parts: the control program for the Macintosh host computer written in C language [8] and the computation program for the DSP56000 written in the DSP56000 assembly language. The control program prepares the various parameters, tables, and envelope functions and also controls the status of the DSP. The computation program, which performs the actual real-time AFM synthesis by using the various parameters, tables, and envelope functions stored in its memory, is written in DSP56000 assembly language [6].

On the Macintosh IIci the function of the control program written in C is to load the AFM parameters, constants, and tables which are required by the AFM algorithm into the DSP56000 memory on the Sound Accelerator Card ready for the DSP56000 to reference them. The sine and exponential tables required for the AFM computation are generated by the 68030 and loaded into the predefined areas of the DSP56000 memory. The execution of the DSP56000 AFM program is done via the control program, which is able to make the DSP56000 execute its program, wait, or stop, as desired.

The DSP56000 assembly programming was written in such a way that the key parameters of the AFM synthesis algorithm could be changed at any point during the synthesis of a musical note so as to achieve a dynamic spectrum. This was done by referring to the parameters stored as a table in the data memory of the DSP56000. If it was desired to change any of these parameter values, the Macintosh control program could alter these pa-

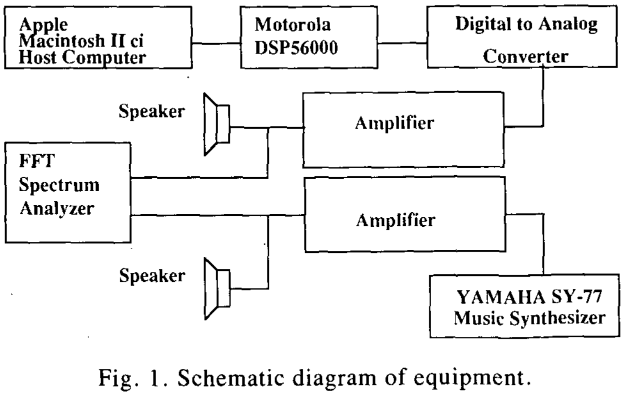

Apple Macintosh II ci Host Computer
Motorola DSP56000
Digital to Analog Converter
Speaker
Amplifier
FFT Spectrum Analyzer
Amplifier
Speaker
YAMAHA SY-77 Music Synthesizer

Fig. 1. Schematic diagram of equipment.

rameter values in the data memory of the DSP56000.

We were unable to calculate the AFM spectrum normalization factor in real time due to the fact that the DSP56000 is an integer processor. This made the real-time computation of the normalization factor which involves floating-point Bessel functions not possible. An alternative method of dealing with the normalization factor would have been to precompute it and store it in a table of values. However, since the RAM memory of the DSP56000 was used for storing the sine table, exponential table, AFM parameter table, and amplitude envelope table, we were not able to store the required table of normalization values.

Because of these restrictions, we have implemented Eq. (5), that is, we proceeded without normalization, instead of Eq. (7). In the absence of normalization it was necessary to control the amplitude of the synthesized AFM waveform so that it did not exceed the maximum allowed by the length of the binary word used, namely, 16 bits. A scaling factor was thus used to control the amplitude. The value of the scaling factor was adjusted by trial and error so that the peak amplitude of the waveform did not exceed the maximum limit of the digital-to-analog converter.

The flowchart algorithm for one AFM operator is shown in Fig. 2. The variables, such as ϕ_{ω_m}, ϕ_{ω_c}, and the temporary variables are declared as memory locations in the DSP. The purpose of the phase variables ϕ_{ω_m} and ϕ_{ω_c} is to simulate t (time) in the real-time synthesis. The temporary variables TEMP _ A, TEMP _ B, TEMP _C, and TEMP _ D store the various intermediate values of the computation. TEMP _ E stores the final value to be converted by the digital-to-analog (D/A) converter. For the second AFM operator, another set of ϕ_{ω_m} and ϕ_{ω_c} is needed. Real-time calculation of the second AFM operator is done immediately following that for the first AFM operator, that is, serially. The outputs of both AFM operators are added and the final value is sent to the D/A converter. After this, the whole process repeats continuously until we stop it through the control program of the host computer.

To synthesize the waveform of a musical instrument, the spectrum of the instrument to be synthesized is first obtained from its sampled waveform using a fast Fourier transform (FFT) spectrum analyzer. After the analysis, the AFM parameters were chosen for which the synthesized AFM waveform would have the closest spectrum to the real sample.

4 REAL-TIME AFM SYNTHESIS OF MUSICAL INSTRUMENTS

For the musical instruments to be synthesized, waveforms from the Yamaha SY-77 synthesizer's sampled waveform banks were used to determine the spectral characteristics of the target instruments. The SY-77's waveform banks contain sampled waveforms of real musical instruments; so they were suitable representations of the instruments which were to be synthesized. We note that between the waveforms of the real musical instrument and SY-77's stored waveforms, there are a series of unknown transitions due to the microphones used by Yamaha and also the frequency response of their amplifiers. We assume that these modifications to the original waveforms are negligible. The instruments selected from the SY-77's banks were the piano, pipe organ, soft saxophone, and harp.

For each instrument the waveform for the A3 pitch (440.0 Hz) was analyzed, and its spectral characteristics were determined using an FFT spectrum analyzer. We tried to ascertain from the Yamaha SY-77 manual [9] the pitches at which the original notes were sampled. The manual states that the piano waveform was sampled at various pitches, each pitch covering a certain range of the keyboard pitches. We examined the shape of the piano waveform across the keyboard and determined that there were actually 12 ranges of pitches having the same waveform: lower keys to F1, F#1 to C2, C#2 to F#2, G2 to A#2, B2 to D3, D#3 to F#3, G3 to A3, A#3 to D4, D#4 to G#4, A4 to D5, D#5 to F#5, and G5 to upper keys. The pitch we used, A3, was from the range of G3 to A3, and thus the original sampled pitch was very close or identical to the A3 pitch.

The other instruments we used were sampled at only one pitch for the entire keyboard range. Yamaha was not able to give us the original sampled pitch. However, we found that this does not really matter for the purposes

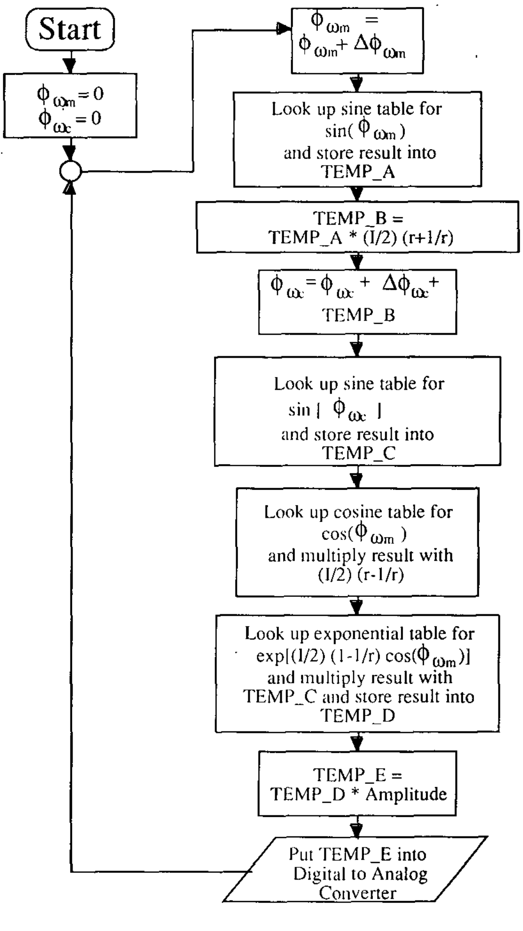

Start
$\phi_{\omega m} = 0$
$\phi_{\omega c} = 0$
$\phi_{\omega m} = \phi_{\omega m} + \Delta\phi_{\omega m}$
Look up sine table for sin($\phi_{\omega m}$) and store result into TEMP_A
TEMP_B = TEMP_A * (1/2) (r+1/r)
$\phi_{\omega c} = \phi_{\omega c} + \Delta\phi_{\omega c} +$ TEMP_B
Look up sine table for sin[$\phi_{\omega c}$] and store result into TEMP_C
Look up cosine table for cos($\phi_{\omega m}$) and multiply result with (1/2) (r-1/r)
Look up exponential table for exp[(1/2) (1-1/r) cos($\phi_{\omega m}$)] and multiply result with TEMP_C and store result into TEMP_D
TEMP_E = TEMP_D * Amplitude
Put TEMP_E into Digital to Analog Converter

Fig. 2. Asymmetrical frequency modulation (AFM) algorithm flowchart for one AFM operator.

of our experiments, since if we had used the original pitch, the synthesis could have been performed in the same way.

Once the spectral components had been determined, an attempt could then be made to reproduce it by AFM synthesis. Each attempt at AFM synthesis used at most only two AFM operators. In some cases one AFM operator sufficed and the other operator was either an FM operator or a sine wave. For comparison, an attempt was also made to synthesize each instrument with not more than two FM operators.

For each instrument being synthesized, the AFM and FM parameters that give the synthesized spectral components closest to that of the original SY-77 sample were determined. The final AFM and FM synthesized real-time waveforms that were generated by the DSP56000 were analyzed by the FFT spectrum analyzer. The spectrum plot for each instrument compares the spectra for FM and AFM synthesis with the spectrum of the SY-77 sampled waveform. The table accompanying each spectrum plot lists the values of the parameters used for each of the operators for both the FM and the AFM waveforms. The parameters listed are A, f_c, f_m, I, and r (only for AFM), with the appropriate subscript to indicate which of the two operators is referred to.

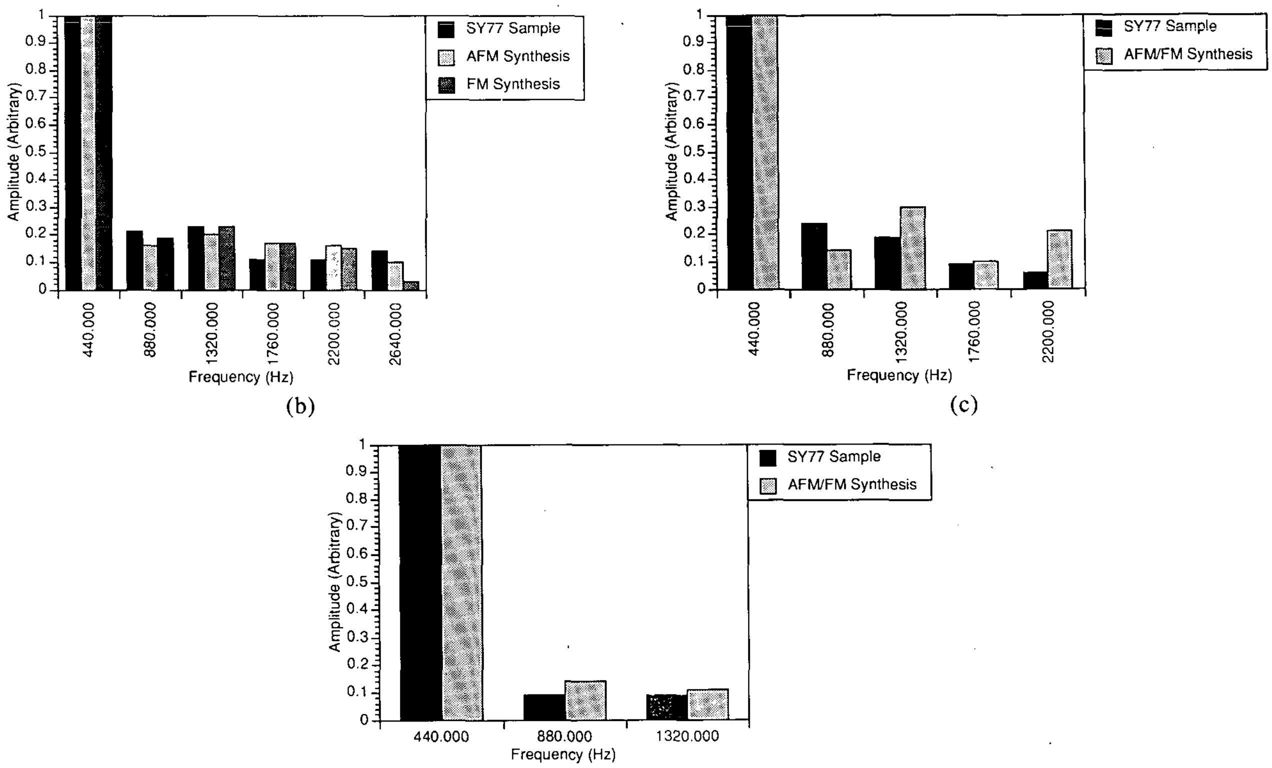
4.1 Piano

Fig. 3 shows our attempt to synthesize the spectrum of the piano note A3. Here we have used one AFM operator to generate the fundamental frequency 440.0 Hz, the third and fifth harmonics. The second, fourth, and sixth harmonics were generated by another AFM operator. The first AFM operator has r equal to 1.0, that is, the special case of AFM which makes it equivalent to an FM operator. This is an example of mixed FM and AFM operators. The amplitude of each operator is controlled individually so that the synthesized spectrum has the closest match to that of the real instrument.

In Fig. 3(b), where two FM operators were simply

Parameters	Attack (FM)	Attack (AFM)	Attack to Sustain (AFM)	Sustain (AFM)
A_1 (%)	100	100	100	100
f_{c1} (Hz)	440.0	440.0	440.0	440.0
f_{m1} (Hz)	880.0	880.0	880.0	880.0
I_1	1.200	2.200	2.360	1.650
r_1	-	1.000	1.000	1.000
A_2 (%)	40	22	23	18
f_{c2} (Hz)	880.0	880.0	880.0	880.0
f_{m2} (Hz)	880.0	880.0	880.0	880.0
I_2	1.100	0.700	0.800	0.000
r_2	-	2.700	1.000	1.000

(a)



(b) (c) (d)

Fig. 3. Piano A3. (a) Synthesis parameters. (b) Attack spectra. (c) Attack-to-sustain spectra. (d) Sustain spectra.

added together, we were not able to increase the sixth harmonic without either increasing the second and fourth harmonics or introducing higher harmonics. We were therefore forced to have a very small sixth harmonic, which affects the timbre of the sound we want to generate. With AFM operators, however, we were able to overcome this problem with the use of the parameter *r*, which was set to 2.7. The value of *r* here, which is more than 1.0, allowed a shift in power from the lower harmonics (namely, 880.0 and 1320.0 Hz) to the higher harmonics. This raised the amplitude of the 2200.0-Hz frequency component slightly and the amplitude of the 2640.0-Hz frequency component (that is, the sixth harmonic of this spectrum) to more than double as compared to when *r* was 1.0.

For the attack-to-sustain portion [Fig. 3(c)] and the sustain portion [Fig. 3(d)] of the piano note, the values of *r* for the AFM operators were set to 1.0, that is, the AFM operators effectively became FM operators. The spectra for these portions are much simpler than the attack portion, so the FM method alone is able to synthesize these spectra effectively.

4.2 Pipe Organ

The pipe organ (Fig. 4) was found to be one of the easier instruments to synthesize. Here only one AFM operator and a sine-wave generator were used. The AFM operator was used to generate the second, fourth, sixth, and eighth harmonics, while the sine-wave generator was set to the first harmonic. The sine-wave generator was necessary because both the AFM and FM operators were not able to generate the 220.0-Hz frequency component with both the carrier frequency f_c and the modulator frequency f_m set to 440.0 Hz.

Here we have used the parameter *r* of AFM to shift the power of the harmonics to the left ($r = 0.68 < 1.0$) to achieve a spectrum that is very close to that of the real sample, that is, the amplitudes of the 1760.0- and 1320.0-Hz frequency components have diminished and the amplitudes of the 440.0-Hz frequency component has increased. The sine-wave generator is not changed so that the first harmonic did not change in amplitude.

If, however, $r = 1.0$, that is, AFM becomes FM, the spectrum generated is very different from the real sample. The 440.0-Hz frequency component becomes only half that of the real sample and the 1320.0-Hz frequency component becomes very large as compared to that of the real sample.

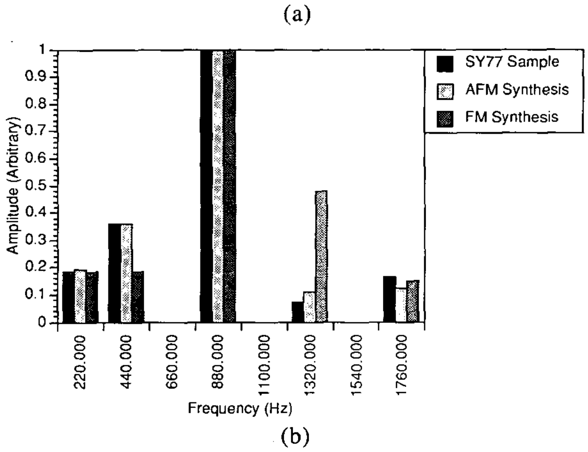
4.3 Soft Saxophone

The soft saxophone spectrum (Fig. 5) was also generated by one AFM operator and a sine-wave generator. The AFM operator was set to generate all five harmonics and the sine-wave generator was set to enhance the fifth harmonic (2200 Hz). The *r* parameter was set to 2.0 and was used to shift some of the power of the harmonics to the right, that is, decreasing the amplitude of the 440.0-Hz frequency component and increasing the amplitudes of the 880.0-, 1320.0-, and 1760.0-Hz frequency components, making the synthesized spectrum a good match to that of the real sample.

If an FM operator were used instead of the AFM operator to generate the soft saxophone spectrum [Fig. 5(b)], the amplitude of the 440.0-Hz frequency com-

Parameters	FM	AFM
A_1 (%)	100	100
f_{c1} (Hz)	440.0	440.0
f_{m1} (Hz)	440.0	440.0
I_1	1.700	1.700
r_1	-	0.680
A_2 (%)	8	8
f_{c2} (Hz)	220.0	220.0
f_{m2} (Hz)	0.0	0.0
I_2	0.000	0.000
r_2	-	1.000

(a)

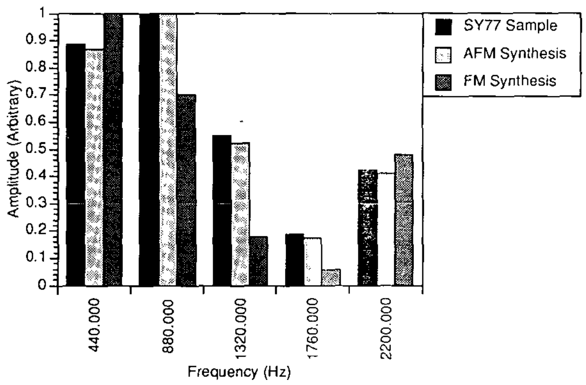


(b)

Fig. 4. Pipe organ A3. (a) Synthesis parameters. (b) Spectra.

Parameters	FM	AFM
A_1 (%)	100	100
f_{c1} (Hz)	440.0	440.0
f_{m1} (Hz)	440.0	440.0
I_1	1.050	1.050
r_1	-	2.000
A_2 (%)	30	30
f_{c2} (Hz)	2200.0	2200.0
f_{m2} (Hz)	0.0	0.0
I_2	0.000	0.000
r_2	-	1.000

(a)



(b)

Fig. 5. Soft saxophone A3. (a) Synthesis parameters. (b) Spectra.

ponent would be higher than that of the real sample and the amplitudes of the higher harmonics (such as 880 Hz and higher) would not be enough to match that of the real sample. The sine-wave generator was also used in the FM operator case to enhance the fifth harmonic (2200.0 Hz).

4.4 Harp

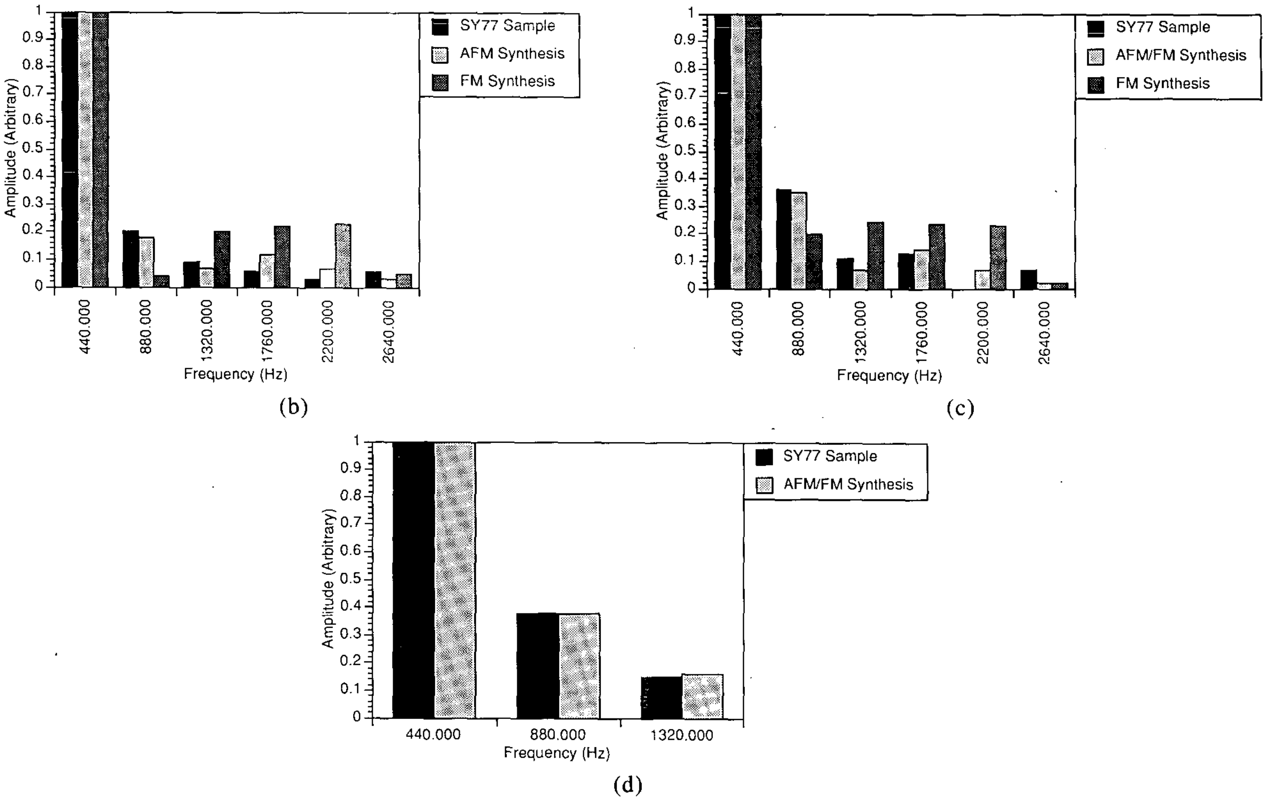
Fig. 6 shows the results for the harp. The attack portion of the harp A3 note spectrum in Fig. 6(b) was generated by two AFM operators. The first AFM operator was used for all the harmonics while the second AFM operator at negative amplitude corrected the amplitudes of the second, fourth, and sixth harmonics generated by the first operator. The second AFM operator had r equal to 1.0, that is, it was effectively an FM operator. This is another example of mixed FM and AFM synthesis operators.

If we used only the FM operators, the amplitudes of the 880.0-Hz frequency component would have been very small and the amplitudes of the 1320.0-, 1760.0-, and 2200.0-Hz frequency components would have been very large as compared to those of the real sample. The parameter r, which was set to 0.885, that is, less than 1.0, shifted the power of the higher harmonics down to the lower harmonics.

The attack-to-sustain portion of the harp A3 note [Fig. 6(c)] was generated similarly with mixed FM and AFM operators. Here the parameter r of the first AFM operator was still set to 0.885. The second AFM operator, which has $r = 1.0$, that is, AFM becomes FM, has a different modulation index ($I = 1.3$) to correct the amplitudes of the second, fourth, and sixth harmonics of the first AFM operator. The parameter r of the first AFM operator, which was 0.885, that is, less than 1.0, has shifted the power of the higher harmonics down to the lower harmonics. If only FM operators were used, the higher harmonics would be too large

Parameters	Attack (FM)	Attack (AFM)	Attack-to-Sustain (FM)	Attack-to-Sustain (AFM)	Sustain (AFM)
A_1 (%)	100	100	100	100	100
f_{c1} (Hz)	440.0	440.0	440.0	440.0	440.0
f_{m1} (Hz)	440.0	440.0	440.0	440.0	440.0
I_1	3.800	3.800	3.800	3.800	1.100
r_1	-	0.885	-	0.885	1.000
A_2 (%)	-75	-75	-60	-60	-25
f_{c2} (Hz)	880.0	880.0	880.0	880.0	880.0
f_{m2} (Hz)	880.0	880.0	880.0	880.0	880.0
I_2	1.100	1.100	1.300	1.300	0.000
r_2	-	1.000	-	1.000	1.000

(a)



(b)

(c)

(d)

Fig. 6. Harp A3. (a) Synthesis parameters. (b) Attack spectra. (c) Attack-to-sustain spectra. (d) Sustain spectra.

compared to those of the real sample. [The FM spectrum in Fig. 6(c) was generated by an FM spectrum generation program.]

The sustain portion [Fig. 6(d)], however, was generated with one FM operator and a sine-wave generator. The sine-wave generator was created by setting the modulation index of the second AFM operator to 0, that is, turning off the modulator, and setting r to 1.0. The sine-wave generator had a negative amplitude and was used to correct the amplitude of the second harmonic, that is, the 880.0-Hz frequency component generated by the FM operator.

5 CONCLUSION

We have shown that the asymmetrical frequency modulation (AFM) synthesis method can be implemented in real time on a modern DSP such as the Motorola DSP56000. We have also shown that the AFM synthesis method can be used to supplement the original FM synthesis method easily, thus achieving a more accurate spectrum. We therefore believe that AFM is a practical and powerful method of generating musical waveforms which would require a greater number of FM operators for similar accuracy.

If the normalization factor in Eq. (7) were also implemented, the parameter r could then be varied in real time to obtain a dynamically changing spectrum, thus enabling dynamic musical notes to be more accurately synthesized. This is not practical in real time with an integer processor such as the DSP56000. However, with a more sophisticated floating-point DSP, like the recently introduced Motorola DSP96000, this will be possible. We hope to investigate this further when we have acquired a DSP96000 system for the Macintosh.

6 REFERENCES

[1] J. M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *J. Audio Eng. Soc.*, vol. 21, pp. 526–534 (1973 Sept.).

[2] D. Morrill, "Trumpet Algorithms for Computer Composition," *Computer Music J.*, vol. 1, no. 1, pp. 46–52 (1977).

[3] S. Saunders, "Improved FM Audio Synthesis Methods for Real-Time Digital Music Generation," *Computer Music J.*, vol. 1, no. 1, pp. 53–55 (1977).

[4] B. Schottstaedt, "The Simulation of Natural Instrument Tones Using Frequency Modulation with a Complex Modulating Wave," *Computer Music J.*, vol. 1, no. 4, pp. 46–50 (1977).

[5] J. A. Moorer, "Signal Processing Aspects of Computer Music: A Survey," *Proc. IEEE*, vol. 65, pp. 1108–1137 (1977 Aug.).

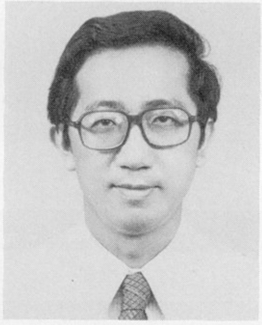
[6] *DSP56000/DSP56001 Digital Signal Processor User's Manual*, DSP56000UM/AD REV 1, Motorola, Phoenix, AZ.

[7] J.-P. Palamin, P. Palamin, and A. Ronveaux, "A Method of Generating and Controlling Musical Asymmetrical Spectra," *J. Audio Eng. Soc.*, vol. 36, pp. 671–685 (1988 Sept.).

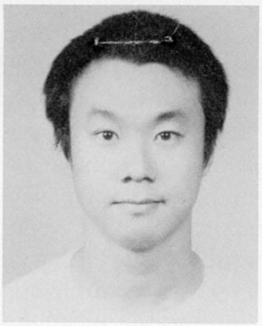
[8] *Think C User's Manual and Standard Libraries Reference Manual*, Symantec Corp., Cupertino, CA. (1989).

[9] *Yamaha Music Synthesizer Operating Manual*, Yamaha Corp., Tokyo, Japan.

THE AUTHORS



B.T.G. Tan



S.L. Gan

B.T.G. Tan graduated in 1965 with a B.Sc. (Hons) in physics from the University of Singapore, and in 1968 with a Ph.D. from Oxford University. He is a Chartered Engineer and a Member of the Institution of Electrical Engineers.

Since 1968 Dr. Tan has taught at the National University of Singapore where he is now an associate professor in physics. His research interests include the microwave properties of semiconductors and dielectrics, surface physics, image processing, and digital sound synthesis.

●

S.L. Gan obtained a B.Sc. (Hons) in physics in 1989 from the National University of Singapore. He is currently working toward an M.Sc. while concurrently working as a research assistant in acoustics with the National University of Singapore.