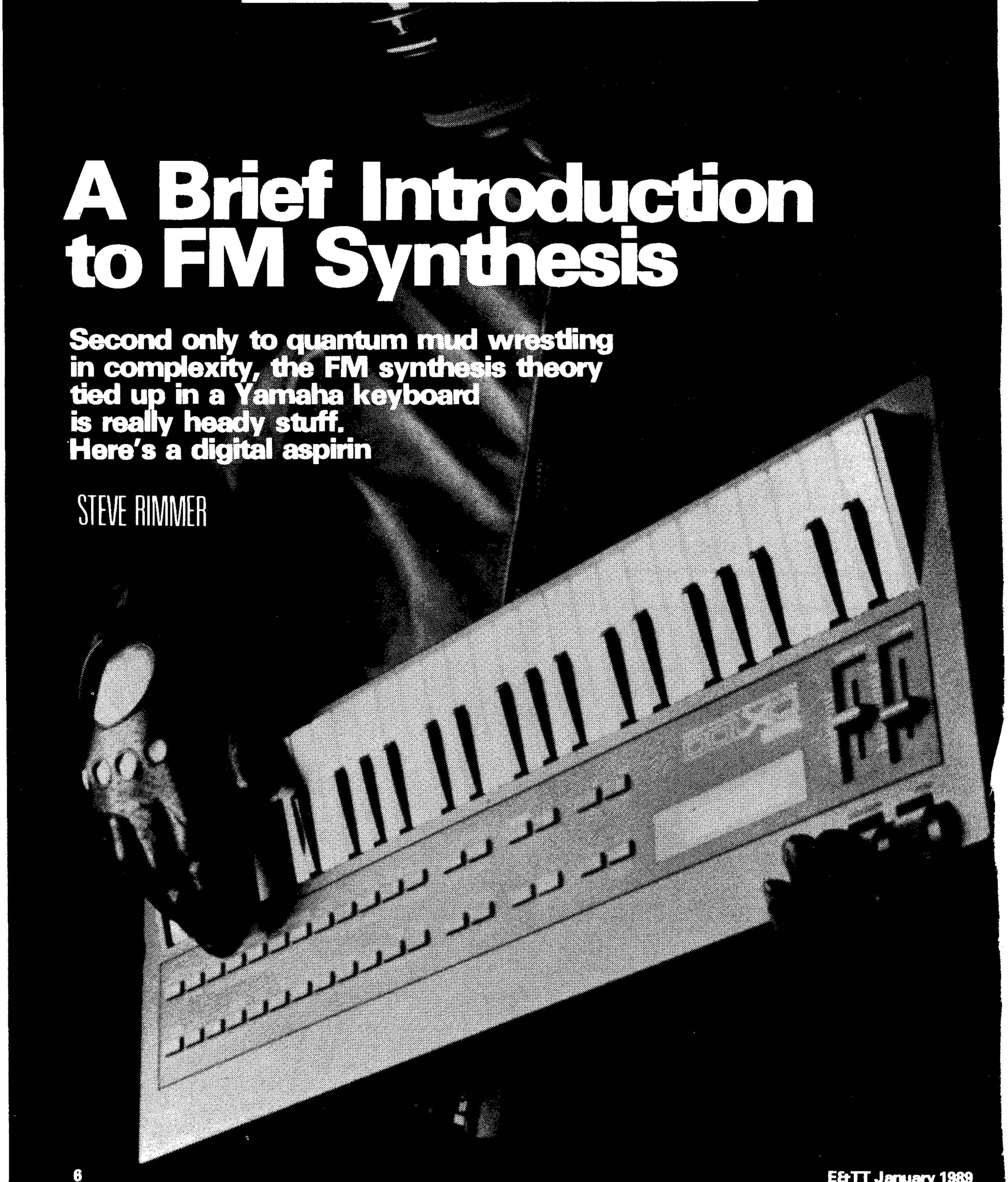
COVER FEATURE

A Brief Introduction to FM Synthesis

Second only to quantum mud wrestling in complexity, the FM synthesis theory tied up in a Yamaha keyboard is really heady stuff. Here's a digital aspirin

STEVE RIMMER



Synthesizers come and synthesizers go. Quite a few have come and gone from my space over the years. Being synthetic, the sounds which electronic instruments dream up seem to go stale more quickly than do real acoustic noises. They have a finite complexity, and the ear grows bored with them after it has heard them a few times.

Bored ears are worse than a Big Mac with too much special sauce.

One of the few synthesis systems which I've actually hung onto has been the FM synthesizers in the Yamaha "X" instruments. I have a number of these going at the moment, including an FB-01, a TX-81Z and, of course, a trusty old DX-7 keyboard. These boxes all use a rather peculiar strategy for making noise – it's extremely hard to understand, but it gets really interesting once you do.

FM synthesis is a pretty hairy subject for a single article – you won't come away from this one knowing everything there is to know about it. However, you should have a reasonable leg up onto it if you plough through to the end of this feature. If you're still curious about it when the dust settles, you might want to check out a book called *FM Theory & Applications* by John Chowning and David Bristow, available from most Yamaha dealers.

Dial 0

The Yamaha "X" instruments all use a pretty consistent *theory* of sound generation – even if none of their voice files are in the least bit compatible with one another. We'll talk about the DX-7 here, although what we get into will apply to any FM instrument you want to check out.

The simplest form of synthesis is basic subtractive analog noise making. It was really big stuff back in the sixties. In its most rudimentary sense, an analog synthesizer takes a complex wave form which contains all the harmonics of the sound we're after and uses filters to snuff the ones we don't like the look of. The result is an approximation of something we might want to listen to.

Analog synthesis of this sort is actually rather boring to listen to, because even complex patches don't really approximate the natural processes which make sound happen. They can't reproduce the complex interplay of energy that exists in natural sounds. If we set up a conventional analog synthesizer with enough filters and oscillators to start to approach the complexity of real noise, we begin running into problems like finding a dedicated hydroelectric project to power it and a team of roadies to plug in all the cords.

The reality of all this being as it may, however, we can still find some useful theory in classical analog synthesis. If you check out Fig. 1, for example, you'll find a pretty easily understood diagram of how square waves come to be. A square wave is the addition of a fundamental pitch and a spectrum of odd numbered harmonics. With enough oscillators going at the correct pitches and amplitudes, the resulting wave would, indeed, look square on an oscilloscope.

In fact, consider that our ears kind of peter out about sixteen kilohertz or so. If the fundamental note is 440Hz, the first odd numbered harmonic is 1320Hz (three times the fundamental), the next is 2200Hz (five times), the next is 3080Hz – it doesn't take long for the harmonics to be too high to be audible. As such, we can see that it's not actually necessary to include very many actual tonal components in a waveform to make it *sound* like a square wave.

Now, up until this point we've been speaking of analog oscillators. There are no analog oscillators in an "X" instrument, such as a DX-7. There are digital things which can be made to behave like analog oscillators, and we'll treat them as such because it's a lot easier to deal with sounds as sounds than as data. However, it's important to note that these instruments really are digital, with all their sounds produced by digital to analog converters. All the stuff we're going to look at in a moment regarding operators and algorithms and such involves the process by which the computer which runs FM synthesis goes about creating its data.

A DX-7 is essentially a dedicated computer with a touch sensitive organ keyboard for input – plus the odd membrane switch and a MIDI port – and a sixty kilohertz digital to analog converter for output. Theoretically, the computer can come up with any sort of data and, as such, can emit any sound imaginable with essentially perfect fidelity. In practice, this is not so, because aside from having the hardware to make the noises we want, we need the software process to generate the data that will produce the sounds.

An "X" instrument isn't quite so versatile as to be able to generate absolutely any sound. However, the vast range of sounds it can get together are a result of a program which implements classical FM synthesis on its internal microcomputer.

Under FM synthesis, a sound source has one or more program segments that generate sine waves – or, more correctly, that generate data which, if sent to the digital to analog converter of the synthesizer would result in sine waves spewing out from the audio jack at the back. These program segments are called "operators". The computer is able to control the frequency of the output of each operator, the amplitude of the resultant sine wave and, to a somewhat less flexible extent, how the output of multiple operators will be combined. That's about it – this is all there is to FM synthesis, at least at the organizational level.

The DX-7 has six operators, that is, essentially six digital oscillators. The output of these six operators is data which which will sound like sine waves when it's eventually converted to audio. However, before this happens the six data streams can be mathematically manipulated. For example, two digital sine waves can be "mixed" by simply adding each of the values of their respective samples to get a third data stream that's a combination of the first two. This is essentially what happens naturally when we mix to analog wave forms, but in this case we must have the computer do the actual math, rather than using handy physical laws which are already in place for analog stuff.

The specific details of combining the outputs of the six operators in a DX-7 are called "algorithms". There are thirty two predetermined algorithms in the DX-7. This may seem like a small subset of all the possible permutations of six entities – it is, in a way, but for practical purposes it turns out that the thirty two algorithms represent all the good bits. Most of the others either just don't sound very nice, or are effectively duplicated in the thirty two algorithms that *are* available.

The output of an operator can go to one of two places – either directly into the resulting sound, or into another operator to control some aspect of it. Typically, in the latter case, this would mean that one operator controls the pitch or amplitude of another. Figure two is a typical algorithm on a DX- 7 – this happens to be algorithm number one. In this case, the outputs of operators one and three are directly producing sound. Some aspect of operator one is being controlled or modified by operator two, and the same can be said of operator three by operator four. However, operator four is in turn being modified by operator five which is in turn being modified by operator six – which appears to be modifying itself as well.

Obviously, this algorithm does not itself define a specific sound, but, rather, one of thirty two basic strategies for generating sound. We still have lots of things which are variable in this basic structure. We get to define whether the parameters being controlled on the operators are amplitude or frequency. We also get to decide on the pitches of the controlling operators, and the amplitudes of their outputs as they affect the controlled operators.

As you can see, there's a good potential here for a really complex final waveform. Once again, it's important to remind ourselves that all of the actual mixing of sounds and modification of the parameters of the operators — the sound sources — is handled in software, and that all the things which we like to think of as being sonic are actually mathematical constructs until they finally hit the digital to analog converter.

No Static at All

We can now begin to understand how FM synthesis gets around to its uniquely complex sound. Consider a two operator algorithm — we don't actually have one, but we can arrive at the result of one by simply turning the amplitude right down on operators three though six on algorithm one. What's left is one operator producing sound and a second one modifying some parameter of the first. We'll set it up so that the parameter being modified in the first operator is the frequency of the resulting sound.

If the first operator is producing a note of four hundred and forty hertz and the second operator a waveform of, say, ten hertz, the result will be a note with vibrato — something which is pretty common even in many acoustic instruments. The amplitude of operator two determines the "depth" of the vibrato, that is, how much the frequency of the note generated by operator one moves around.

In a purely technical sense, the frequency of operator one is being modulated by the frequency of operator two — frequency modulation, which would seem to have something to do with the name for the synthesis technique used in "X" instruments.

If you increase the frequency of operator two — a lot — the rate of the vibrato of operator one will appear to increase. After a while, it will increase to the point where there's no longer any real perception of a single note swinging slightly in pitch. Instead, we experience a single, extremely complex sound with a lot of harmonics which are obviously related to the fundamental, but not in any immediately obvious way. A very interesting sound indeed results from operator two having the same frequency as operator one.

Think about that one for a sec —

A whole range of potentially interesting and useful sounds can be arrived at by making the pitch of operator two numerically related to that of operator one. Having them removed from each other by one or more octaves is a fairly obvious arrangement, but we can also experiment with fixed

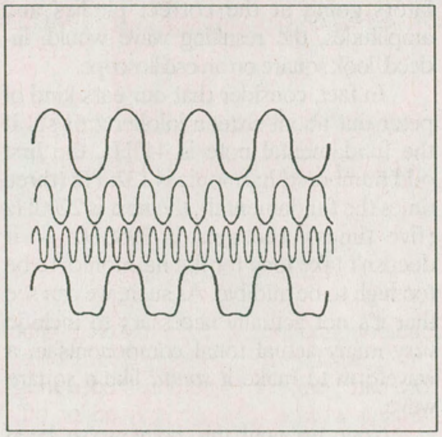


Fig. 1. If enough sine waves of the right pitch, phase and amplitude are combined, the result is a square wave.

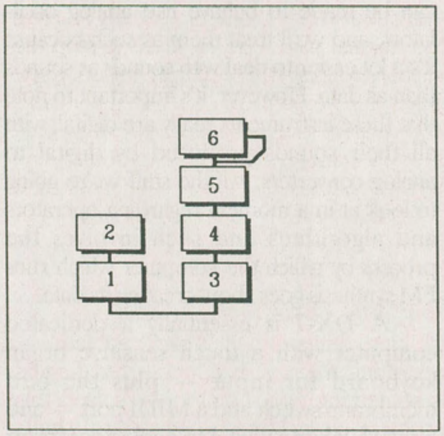


Fig. 2. An algorithm which guides the interaction among the operators in a synthesis device.

musical intervals, such as fifths and sevenths. In each case, a different set of complex harmonics will be generated, and different — potentially interesting — sounds will scream their way out of the digital to analog converter of the synthesizer.

Using the traditional process of backwards justification — finding a good reason for doing something after it has proven successful — we can see that the physical processes of producing acoustic noises usually do so though a mechanical analog to FM synthesis. A saxophone, for example, has a fundamental vibrating frequency generator — a reed — with all sorts of other mechanical "operators" acting upon it, including the cavity it vibrates in, the shape and dimensions of the horn it blows into and the velocity of the air blowing over it.

As such, FM synthesis produces a better synthesis of many acoustic sounds than one might expect because it turns out to use much the same process — if not the same tools — as the analog machines which originally generated the sounds. Having said this, I should note that a saxophone is one of the few instruments I've never been able to work up a satisfactory DX-7 voice for.

Obviously, if we combine all six operators of the DX-7 in this way, we'll have a very, very complex waveform indeed at our disposal. Allowing that we can actually make some sense of it — which turns out to be a lot easier to do in practice than it is on paper — we can approximate a lot of really decent sounding instruments quite closely.

We Did Say "Brief"

As you'll note if you look down this column a bit, this is where we're going to let go of FM synthesis for today. In fact, the theory of this rather complex subject really only begins here. It gets pretty mathematical from this point on, though. More to the point, the reading grows a bit thick.

It will be fairly obvious, however, that the rather simplistic looking algorithms of an "X" instrument are the basis for an almost infinite number of sounds. Predicting what the result of using these operators will sound like is, of course, the real trick of the exercise. Because of the complexity of FM synthesis, this isn't anything like easy. In fact, it's really only through experience that one can begin to predict what an FM voice will sound like prior to its actual completion.

While there are more techniques for digital synthesis than most people not living near a nuclear power plant have fingers and toes, the FM synthesis technique used in Yamaha's "X" instruments has the dual advantages of being almost infinitely flexible and very nearly incomprehensible for a long time after you get started with it. The latter has the effect of keeping the plebes away from your synthesizer — worth it for that alone.

Finally, when it's all over with and you just don't want to look at another operator until hell or Ottawa freezes over, you can take the sensible approach that I use. Nip out, buy yourself a couple of disks full of canned commercial voices and get back to just playing music. ■