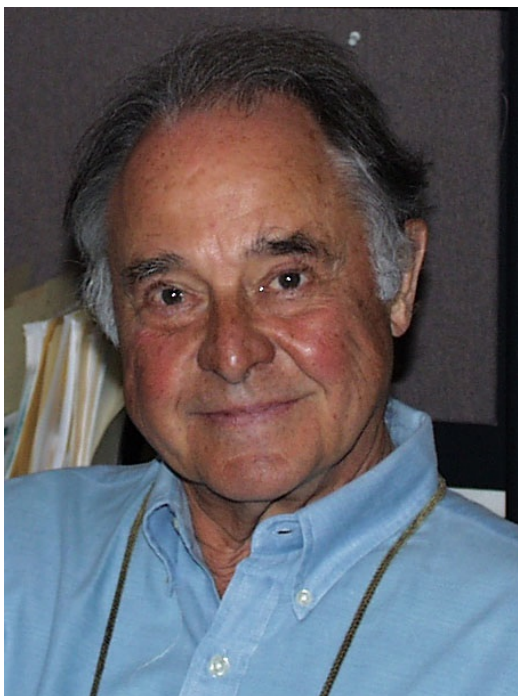


Interview - John Chowning

14 July 2013



John Chowning with a Yamaha DX7 (Chuck Painter)



John Chowning now (John Chowning)

John Chowning was instrumental in establishing a computer music programme at Stanford University's Artificial Intelligence Laboratory in 1964. He discovered the frequency modulation (FM) synthesis algorithm in 1967 and used this discovery extensively in his compositions. In 1973 Stanford licensed the FM synthesis patent to Yamaha in Japan, leading to the hugely successful synthesis engine used most notably in the DX-7 from 1983. In 1974, with John Grey, Andy Moorer, Loren Rush and Leland Smith, he founded Stanford University's Center for Computer Research in Music and Acoustics (CCRMA).

Stanford University page about John Chowning
(<http://ccrma.stanford.edu/people/john-chowning>)

This is a composite interview made from interviews conducted on 27 May and 10 June 2010. Updated and corrected by John Chowning and James Gardner, May 2012.

James Gardner: What are you working on now?

John Chowning: Currently I'm working on the last version of *Voices*, a piece for soprano and laptop computer, an interactive piece using the programming language Max/MSP, which is a kind of *lingua franca* of the computer music world. This is a piece that was commissioned by the Groupe de Recherche Musicales in Paris, and version 1 was performed in March 2005, and version 2 has been performed in this country, Argentina and France. I've had a couple of years of health problems so I've had to put things away a bit. But now I'm back on track finishing it up and about to begin a piece based upon the same program structure for french horn and interactive laptop.

Is *Voices* available on CD?

Not yet, but we plan to make a DVD and publish a 5.1 version plus the score and the actual Max/MSP patches. I have a very strong feeling about composers making available the details of their work to the point where other composers and interested musicians, or even lay people, can get access to the details of the actual sound production in the same way we can when we look at an orchestra score, say of a Strauss symphonic poem.

And this is also, presumably, to fight against obsolescence, too?

Yes, that's right.

It must be interesting for you to be working on a laptop now, with Max/MSP, after those years of working with mainframes and so forth. What are your feelings on working in this way now—does it feel like a summation of where you were heading many years ago, or does it just feel like another evolutionary step?

Well, it's quite an astonishing experience, because I'm one of the old guys and have a strong memory of the way we made music in the early '60s with long delay times between musical decisions and musical results. To be working with a laptop, where I have more computing power than all the computers and Samson Boxes [1] and synthesizers than ever existed all at once right before me, is quite an amazing experience from my point of view. And in addition to the fact that it's a kind of window to unlimited amounts of knowledge, access to resources etc. Quite unlike computers of the '60s, which were, as you've noted, mainframe, although we had some vision of the future because the computer that I worked on in the '60s and '70s was an A.I. Lab computer that was time-shared and had access to the ARPANET [2], so there were features such as e-mail and access to computers remotely.

What was the process, back then, for creating music on the computer?

Well, the typical case in computer music in those years was based on batch processing where one submitted cards that contained program and data. These were then run through the computer and the results—the sampled waveforms, in the case of music—were written on to magnetic tapes. Those magnetic tapes, at Bell Labs for example, were taken to another device that read the samples from the tape records—with buffering to eliminate gaps in between the records—and converted into sound. So...we were rather fortunate in that my first system, the large IBM 7090 was connected to a disc that was shared by a PDP-1, a DEC computer—their very first computer in fact—and that computer had a DECscope and the X/Y co-ordinates for the beam deflector served as my digital-to-analogue convertors. And so the data was written from the IBM 7090 on to a disc and then continuously from the disc in another step to the D-to-A convertors—the DECscope—and recorded on to tape. So it was probably the first online system in a sense.

That was unusual for the time because usually you would have to wait for a very long time to hear what you'd actually done.

Yes. Typically a 24-hour turnaround.

That long delay must have been very frustrating.

Well, as there was no alternative, there was no frustration. But the fact that there was such a delay made us think carefully about what we did, and take a lot of care in the detail of the card specification or code specification, in the case of the time-share system, because mistakes are so costly. So all the time that we waited for sounds to emanate from the computer we thought about the next step: “if this, then what would I do otherwise—what are my choices?” So it was hardly wasted time, and it’s developed habits that I maintain to this day—being careful about the choices that I make in program code.

It gave you time to reflect.

Exactly.

How do you feel about the way one works now, where the feedback loop is much quicker, and you’re working more or less directly with the sounds themselves?

Well, in some cases that’s very useful, but I tend to step away from that and continue the habits that I developed years ago and reflect on the choices I’m making. There are some things, of course, where being able to tune parameters in real time, and pick carefully based upon immediate feedback gives a certain advantage. But not always.

There’s a tendency, when discussing electronic music, for the technical aspects of the work to take centre stage and this can detract from the fact that you, for instance, were trained first and foremost as a musician, not as a technician.

That’s true. I was trained as a musician, absolutely—I have no formal training in technology or science or engineering. But I found that the context of computing, or rather the environment of the Stanford A.I. lab, was a very rich one—a population of users including linguists, philosophers, psychologists, and of course computer scientists and engineers. So the *lingua franca* was, of course, the language of the computer, which first of all was FORTRAN [3], I guess, and then SAIL, which was the Stanford Artificial Intelligence Language, and that was a procedural language that had some very rich features. For example I discovered the idea of recursion in that language—an idea that I used in a piece, *Stria*, which I composed in 1977. So I found that the ideas—what one might think would be barriers to musical thought—in my case turned out to be amplifiers of musical thought.

That's partly because you had this cross-disciplinary social network, but also because it allowed you to perhaps translate some of the computing ideas into musical or compositional ones.

Right. And I had a population of people to whom I could go who would explain concepts to me, and a very supportive environment. There was a kind of an unwritten rule, which was: I could go back and ask as many questions as I chose—as long as it was not the same question.

You went to Stanford after studying for three years with Nadia Boulanger in Paris, and presumably you heard some electronic music at some point. Can you recall your impressions of hearing that music for the first time, and what it was?

When I was studying in Paris with Nadia Boulanger between 1959 and 1962, Paris was a very rich environment as far as new music was concerned. Pierre Boulez had the new music series, the *Domaine musical*, at the Théâtre de l'Odéon and so those concerts were lively and often included electronic music. I'm not sure whether it was there that I first heard *Kontakte* and *Gesang der Jünglinge* by Stockhausen. *Trois Visages de Liège* by Pousseur I think I heard there. So I was quite struck with the idea of using speakers in a way that was creative, rather than using them as a means of reproducing music written for a totally different context. And when I went to Stanford, then, I was unhappy to discover that there was no possibility of doing any sort of electronic music there. And then I read the article [4] by Max Mathews, and discovered that we had computers that could run those programs. That was in my second year of graduate school. That was a great moment.

This was an article where he described the open-ended nature—at least in theory—of what the computer could do as far as generating sounds was concerned.

That's true. Basically the idea of information theory—being able to represent any signal by a set of numbers as long as the sampling rate and the accuracy of the numbers were great enough. And so that was the first publication that was in a public journal—I think that was in *Science*, November 1963 and I came upon it quite by chance a month or two later.

And that was the spark.

Yeah, that was the key: such a stunning statement in the article that one could—with a computer and loudspeakers—produce any perceptible sound as long as one assumed loudspeakers as a

source.

But at Stanford the music department didn't actually have any computing facilities.

They had no interest at all in electroacoustic music. Correct.

You had to go to the A.I. department.

I went to the Computer Science department. Exactly.

Did you feel like a loner there?

Let's say I did feel alone, but kind of wonderfully alone. I was tutored, more or less, by a young undergraduate who played the tuba in the Stanford Orchestra. And he was a budding hacker in the old, good sense—that is, someone who is completely devoted to computers. And he's the one that helped me get started.

One of the things I think you started working with was reverberation and spatial movement within four channels. Is that right?

Yes that's correct. Well, we began with two channels but by 1966 we were able to do four-channel spatial illusions. And the idea was based on freeing sounds from point-source locations and freeing them from their positions at loudspeakers to create illusions of sounds that come from not only between loudspeakers but also at varying distances. And also motion that included, of course, Doppler shift. Now this was very much dependent upon reverberation for the distance cue. That is, in order to perceive sounds at varying distances we have to control the amount of direct signal—or the apparent signal that comes from a sound source— independent of the reflected signal; that is the sum of which make up the reverberant signal. So the work of Manfred Schroeder figured very prominently in my work. He had developed an all-pass reverberator and I discovered that in one of the journals. And we implemented those reverberation schemes in that first research.

Would that have been in Music IV?

Yes that's correct—Music IV.

What was the first piece that came out of that?

The first piece that I did finally was in 1971. I mean I did a number of trials, of course, but the first piece that I felt...well, let's say...I must point out that in 1967 I discovered FM [5]...

Yes!

I stumbled upon that, and so that plus the work in spatial music became a rich place to work. By 1971 I had developed FM and the localization to the point where I felt I could make music that met the criteria that I held.

That was *Sabelithe*, a short piece that used FM—it's not the first piece to have used FM, however, but it was the first piece that I had done using it.

What was the first piece to use FM?

Well, until recently I thought it was Jean-Claude Risset's piece *Mutations*. In the latter part of that piece he uses FM synthesis. But since then I've been reminded that Martin Bresnick, then a graduate student in composition at Stanford, composed music for a short film, *Pour*, where he used FM synthesis under my guidance [6]. That was in the Spring of 1969, whereas *Mutations* was composed in the Autumn of the same year.

FM synthesis is, of course, what you're best known for. This came out of your experiments with extreme vibrato, didn't it?

That's correct, yes. I was searching for sounds that had some internal dynamism, because for localization one has to have sounds that are dynamic in order to perceive—especially—their distance, because the direct signal and the reverberant signal have to have some phase differences in order for us to perceive that there are two different signals that you are sensitive to.

This is what you get your depth cues from.

That's correct. Vibrato is one of the ways that one can do that. The reverberant signal is a collection of dense iterations of the direct signal, and by creating tones that have vibrato one makes the mix of the reverberation and the direct signal distinguishable.

I was experimenting with vibrato, so in the course of those experiments, I increased what I then called the vibrato depth, then the vibrato rate until it was beyond the normal physical boundaries that one would encounter in the case of vocal vibrato or string vibrato or wind vibrato.

This is the equivalent of moving your finger faster and further than you could physically do in the real world.

Correct. That's right. As if we had some magical violin that had a greased fingerboard, and we could zip up and down the fingerboard with our finger tens and hundreds of times of times per second.

So there was a certain moment in these experiments where I realized that I was no longer listening to a change in instantaneous frequency but rather I was hearing a timbral change or a change of spectrum.

Now it's important to remember that in those days, in experiments of this sort, a second or two of sound would be preceded by many minutes of computation, so there was lots of time to think about what I was doing.

There were two oscillators: I was using a sinusoid as a source that I was modulating with another sinusoid. And I realized that I was producing a complex spectrum using these two oscillators. I became rather, let's say, animated; and realized that this was something that was surprising and I was achieving a kind of aural complexity that I knew was difficult to do other than with lots and lots of oscillators, or perhaps a filter and a complex wave, both of which were computationally expensive. The last fact was not very important to me at the time, but it became important as this idea developed.

Just to clarify, so we've got a better image of this: at this time you weren't turning four or five knobs to change the modulation index and the modulator and carrier frequencies—you wrote lines of computer code in order to specify such things. So there was no direct feedback about the relationship between those changes and the sound you heard. You had to wait a long time until you heard the results of those specifications.

Exactly. One had to wait before one could advance a parameter. For example, the parameter of modulating frequency (or vibrato frequency). Let's say I went through the following steps: 5 times per second, then 10 times per second, then 20 times per second, then 40 times per second. And at about 40 times per second—and then doubling it again to 80—that's the region where I realized I was no longer hearing instantaneous frequency, but I was hearing a complex timbre.

How long did it take you to start to intuit the relationship between the modulation index and the modulator and carrier frequencies? It's quite difficult to get a feel for that sort of thing unless you've got your hands directly on the controls.

Yes...it actually was quite easy because, as I say, there was so much time between trials. While I was listening to, for example, a modulating frequency of 100Hz and a carrier frequency of 100Hz and perhaps a frequency deviation of 100Hz, I was

thinking “OK, well depending on how this sounds, I’ll try doubling everything and see if I have an octave transposition”. So as I waited, and heard the result, then I doubled everything and indeed it transposed. So that was very encouraging because that meant there was something orderly in this process that was therefore intuitive.

So I did a set of experiments based upon maintaining two parameters constant and changing one of them, and especially sweeping the deviation. As the deviation went from zero to non-zero the frequency didn’t change, but the richness of the spectrum changed: that is, the amplitudes of the partials. So by going through these steps and then doing such things as transpositions or changing the ratio and maintaining a constant ratio and changing the frequencies I realized that this was orderly, and I was able to produce both harmonic and inharmonic spectra, and produce some bell-like, chime-like tones.

I then went and showed what I had done to a young mathematics undergraduate and together we looked at an engineering text, found the chapter on frequency modulation and observed that what I was doing would be predicted by the equations in the book that explained frequency modulation. All the basic explanations are, of course, based upon a sinusoid modulating a sinusoid—exactly what I was doing—the only difference being that instead of the carrier frequency being in MHz it was in the audio band and that we were listening to the “un-demodulated” signal.

What was described in the engineering text was FM at radio frequencies, then?

That’s correct, yeah. The way they explained it was just using a modulating frequency that is a sinusoid—because it’s a simple explanation. One could then extrapolate this to a complex wave—which of course is what’s actually broadcast in FM radio. So the simple explanation in the engineering text was exactly what I was doing.

But it had not, until then, been applied in such a precisely controlled way in musical applications.

Yeah. And all the analogue synthesis was, of course, based upon voltage and that’s logarithmic. So this was linear modulation, and voltage-controlled devices could not produce the same effect.

Because the control range was different—it was exponential rather than linear.

Exactly.

After my initial discovery I did a set of more or less rigorous experiments and demonstrated for myself that this was not a phenomenon that was chaotic but it was predictable, transposable, and so it was really an ear discovery. That's a very important point, I think. It did not fall out of the equations. In fact two or three years ago I was giving a talk about this at the University of California, Berkeley to an engineering class and the professor, after my presentations, said "well, you know I could look at those equations"—that FM equation, which of course he knew very well—"for many, many hours and never be able to see that one could produce sounds of the sort that you've produced". And I said "well, I couldn't either!"

How did you make the leap from these initial intriguing sounds to the synthesis of sounds that had the dynamic and timbral richness that you were seeking?

Well, actually that was one of the early steps in the first set of experiments. I had a linear interpolation between a deviation of zero and non-zero because I wanted to see what the effect was in a slow change in, say, vibrato amplitude—which is really the deviation, and indirectly it's the modulation index. So I would sweep the index from zero to a large number, say in the case of carrier-to-modulating ratio of frequencies of 100, both, and sweeping the index from zero to 100 it went from a sinusoid to a well-behaved harmonic spectrum. And then we made a change in the oscillator such that when the deviation became greater than the carrier frequency the phase changed—that is, the oscillator would step backwards through the look-up table in the stored sine-wave function. So it was just a simple change in the program allowed me to have full-blown FM.

So if you had a deviation greater than the original frequency it wouldn't just stop at zero it would, as it were, go 'through' zero.

That's correct, yes. Just with a phase change. And that was just watching the sign. Until then, the original Music IV oscillator ignored the sign of the increment to the oscillator, the way Max originally set up the table look-up scheme. So in the book [7] about Music V that Max published in 1969 it did not include the phase change. It was in my article [8], published in 1973 that I have a footnote with the change to the Music V code that's required.

This was a modification that you had to do in order to implement FM in the way that you had been exploring it.

Yes, to have a full implementation. In the existing oscillator one could have a deviation that descended to zero but not beyond zero.

It was my ear that led the discovery because my ear was, in a sense, extraordinarily hungry for sounds that had internal dynamism. The only other work that had been done at that time in this regard was by Jean-Claude Risset and Max Mathews at Bell Labs. They were using additive synthesis, which was very, very expensive. Risset had done some extraordinarily beautiful and elegant simulations of musical instrument tones.

He'd worked on trumpet tones, hadn't he?

Yeah, that was the analysis that he did, which I remembered a couple of years later and implemented in FM, which was kind of a breakthrough moment. But with that additive synthesis technique—using multiple oscillators each of which had independent control over amplitude and frequency—he had simulated flutes and some percussive tones and gong tones, all of which were extremely elegant, but very, very expensive to produce.

That's because for each harmonic you had to have a separate oscillator.

That's right. Each partial had to have a separate oscillator and controls over amplitude and frequency.

So one of the breakthroughs of FM was its computational efficiency.

Exactly. Well, the breakthrough was in several domains. There was a musical breakthrough in the sense that the few parameters were salient parameters with regard to perception. That is, the ratio of the carrier and modulating frequencies determined the harmonic or inharmonic nature of the resulting spectrum, and there were a huge variety of choices within each of those. And the fact that the modulation index is a function of time produced a change in the spectrum in time. So those were extremely important, musically. But computationally, it demanded little memory—just piecewise linear functions and bits of data control to specify frequencies.

So it was very efficient in memory and computation. And one attribute that was very important—which Yamaha understood straight away when we explained this technology to them—had

to do with aliasing. That is, the bandwidth of the spectrum could be controlled by one number—the modulation index. Aliasing was a big problem in the early considerations of computer music and real-time digital synthesis because the sampling rates were relatively low and the way that one generated tones typically was a complex wave like a square wave, sawtooth wave, triangle wave, which have lots and lots of harmonics—well, basically an infinite number—and so this produced spurious frequencies that were audible and problematic. So they had to have filters or band-limited stored waveforms. There were lots of complications that the FM synthesis technique solved.

Now this all might remain a footnote in the annals of audio engineering or academic electronic music, but the point about your discovery was that it had such an incredible impact in the popular music world. How do we get from you discovering these things at Stanford to Yamaha picking up on it? I think you first hawked it to a number of organ companies—is that correct?

That's correct.

So who first saw the potential of this as an economical and new way of generating electronic sound?

Well, it was Yamaha. First of all I signed over the rights to the patent to Stanford University who then assumed all responsibility for patent searches and costs and whatnot. And they did the initial investigation into what music industry members might be interested in this. Of course the obvious one was electronic organs, and one of the big names was Hammond. They already had a successful line of studio organs, or home organs. And they sent engineers and musicians to Stanford to look and listen. All I had was a computer program and some sound examples—some very impressive sound examples for the day—some brass-like tones and percussive tones, all done with the simple FM algorithm.

And of course they were interested in the implementation, but their whole world was analogue. And I could not help them. I could show them the program and the logic—well basically the circuit, the block diagram circuit that I used to generate these tones. But they didn't understand basic digital representation, even sampling. And this was the case also with other organ companies, Rogers and Lowrey—these were the big organ companies. The Allen organ company was already working in the digital domain but I don't remember whether we contacted them. I suspect not, but I'm not sure about that.

But the other manufacturers weren't able to see that if they were to change to a digital implementation they could pick up on this.

That's correct.

Were Yamaha particularly forward-thinking as a company or did it just happen that someone there was bright enough to see the potential in FM?

Well, Yamaha was already working on basic research in digital sampling and representation. They were working on hybrid systems, like the digital control of analogue synthesizers. But I think they also saw that there was a possible future in direct digital synthesis. So when the Yamaha engineer [9] visited, and I gave the same examples and showed the same code—with a brief explanation—in ten or so minutes this engineer understood exactly what I was doing, because he was already well-schooled in the digital domain. They saw lots of problems that existed, for example aliasing. Sampling rates that they could envision were of the order of 20kHz or 25, 30, all of which present aliasing problems if one uses a complex wave.

These are pretty low sampling rates by today's standards.

Right. Exactly. And he saw that in frequency modulation synthesis, with just one parameter—the modulation index, or indirectly the frequency deviation—one could control the amount of significant spectral components. So by frequency scaling as a function of pitch height he could quickly see that they could avoid the whole aliasing problem. And then, of course, there was the economy of the algorithm—two oscillators to produce a wide variety of different timbral categories, from brass to woodwind to percussive etc. And small memory—that was the other thing. The envelopes were just piecewise linear and therefore there were only a handful of data points per function.

So this required relatively simple computation and used very little memory.

Exactly. And gave a huge timbral space. So that was what they saw immediately. That was in 1971 or 1972, when I first showed these examples. Of course the technology at the time—the LSI technology—was evolving rapidly, and while they could not have envisioned a commercial instrument in 1971, 72, they could see that maybe in a decade that the technology would have

evolved such that they could—with some special skills, and the development of their own LSI capability—build an affordable instrument. And in 1983 they did!

So the time delay between your demonstration to Yamaha, and their production of a relatively affordable instrument was due to the fact that they had to design the chips and do the research, and that took a lot of R&D time and money.

Yes, that's right. I visited probably 15 or so times in those ten years and worked with their engineers developing tones, and they developed various forms of the algorithm. All the while, of course, the large-scale integration was doubling its computational capacity and halving its size every year or so. So there was a lot of research trying to anticipate the convergence point of what they knew and what the technology would allow. And that moment was 1981 when they introduced the GS-1, a large 88-key keyboard, which was used by one group famously. I think Toto was the name of the group.

Yes, on their song *Africa*.

...the tune *Africa*.

After the GS-1 there was the GS-2 and then in 1983 the DX-7, which was the first FM [10] synthesizer to really take off. Do you think the fact that Yamaha implemented a great deal of touch sensitivity and finger control in those instruments was as much responsible for their success as the FM synthesis generating technique itself?

Well I think they're directly related. That was one of the things that was so very attractive—that they could couple the strike force on a key to the synthesis algorithm such that it not only affected intensity but also spectral bandwidth. And that was very important because the spectral bandwidth, or shifting the centroid of the spectrum up in frequency with increased strike force, is critical to our notion of loudness. So they saw on the DX-7 that good musicians would make use of that in a way that not such good keyboard players, would not. And I guess one of the British rock bands of the day had David Bristow—who was one of the factory voice people [11] for the DX-7—kind of advising them. And David pointed out that you're not going to punch through the drums and guitars just by increasing the overall intensity. But if you increase the amount of modulation index as a function of key velocity (which equals strike force) then you'll get through. And that was a revelation for this musician.

So the harder you hit the keys the brighter the sound gets. Just like a piano.

Exactly. Happens on all musical instruments except for the organ. All the traditional musical instruments—whether it's the greater the breath pressure in wind instruments or the pressure against the vocal folds the—greater the brightness. So it's a part of every natural, every acoustical instrument.

And it's an intuitive thing for most musicians.

Yeah. And deprived of that, one loses a lot in the notion of loudness, and loudness, of course, is one of the basic musical parameters.

I'd just like to go back to the transition again from the experiments you were working on with FM and Yamaha picking it up. Presumably at some point you played the demonstration examples to other people at Stanford who said "we think you're on to something". What kind of encouragement did you get?

Well, I was working on music, on compositions, and my whole environment was not one that was developing technology for the music industry but trying to make use of computers in a way that was expressive for my own interests in composition.

I first generated, for example, the brass tones in 1970 or '71, remembering some work the Jean-Claude Risset had done at Bell Labs in the analysis and synthesis of brass tones, where in the attack portion of the tone his analysis revealed that the harmonic energy increases as the centroid of the spectrum rapidly shifts up in the first 20 or 30 milliseconds of the attack. That's too fast to hear in detail, but it's a critical signature of the brass tone. So in '71 I remembered that and coupled the modulation index envelope and the intensity envelopes (a typical brass-like amplitude envelope), found the correct scaling and had some very elegant-sounding brass tones that worked through all the ranges from high brasses to low brasses. At that moment I knew myself that I was on to something, without any confirmation from anyone else—although of course I played it for musician friends and computer science friends in the A.I. Lab. And everyone was astounded that we could get so close with such a simple algorithm.

What did you feel about FM synthesis being yoked to a keyboard? The pieces you'd produced in the '70s using this technique were very fluid sounding, and yet Yamaha hooked it up to a keyboard and equal temperament. What did you feel about it being used in that way?

Well, my musical reference was not popular music so I didn't know a lot of the work that was done using the DX-7. I do now, because I guess it defines an era in a way. But I was not so interested in the DX-7 as a musical instrument for the reasons that you just mentioned. One lacked a kind of control—over pitch frequencies especially—that I had been very interested in. Curiously, [György] Ligeti was in Paris during the mid-'80s occasionally and we talked with Yamaha about producing microtuning. Ligeti was very interested in that, and we made the case to Yamaha. Of course their sales force had little interest in microtuning because most of their world was in common-practice tuning—the pop music world. But we explained to them that they could improve the piano tones if they had microtuning because of the stretched octave phenomenon etc. So in the DX-7II they provided it, and Ligeti was a very strong voice in that effort at persuading them.

Simha Arom, a musician and ethnomusicologist in Paris, used the DX-7II to do some experiments with the Aka pygmies in Africa and their tuning system. That's perhaps not too well known, but it's a very interesting early application of using the DX-7II keyboard with microtuning to determine preferences of this community of people whose music was very much dependent upon non-Western tuning systems.

Ligeti was a friend of Simha, and a couple of years later Arom dedicated a paper—that paper, I think—to Ligeti. But he did take this DX-7II to Africa to do these experiments.

You mentioned that Max Mathews' 1963 paper 'The Digital Computer as a Musical Instrument' was an inspiration to you, and one of the things that I find remarkable about it is that he's basically got the whole notion of digital synthesis and sampling sewn up pretty quickly.

Yes.

Do you think his achievements are slightly underrated, in that he was able to see that so early—at a time when actually implementing those ideas was so time-consuming and costly?

Well we certainly credit him with enormous insight. It's not only that he saw it early but that he implemented it in a way that people like me—without any engineering or scientific background—could understand it. That's the idea of the block diagram; the unit generator, the signal flow diagram. It was basically like Max/MSP—which is a popular program today, an object-oriented program. So Max's original conception, the way

he represented, was an object-oriented program, which allows a simple visualization of signal flow and processing. So he not only saw that this was powerful and that sampling was a general way—I mean that was what caught my attention: his assertion that any sound that can be produced by a loudspeaker can be synthesized using a computer. And that was such a stunning statement because we know that loudspeakers have an enormous range of possible sounds. But it was not only that. It was his insight that psychoacoustics would be an important discipline in the evolution of this as an artistic medium. And indeed it was. Perception, psychoacoustics turned out to be critical in a way that few people foresaw.

And that's what you were exploring in one of your early pieces with the psychoacoustics of reverberation.

That's right, and also the whole notion of loudness. Typically, if an electronic musician today wants things to be louder, they just push up the pots on the gain. In fact, loudness is a very complicated percept, as I sort of suggested, related to spectral centroid or brightness as well as intensity and also distance—the idea that sounds can be loud but distant or at an intensity that is measurably low because they're loud at a distance, which means that they sound loud but they're far away. So it gets pretty complicated.

You can tell the difference between a very distant, loud trumpet and a close-up quiet flute even though they may have the same sound pressure level when they reach your ear.

That's right—they could be the same measurable intensity. Their RMS intensity could be the same except that we perceive one as being loud and the other soft because we know something about auditory context.

David Bristow, who programmed many of the original DX-7 presets, discovered that if he put more of what he called 'stuff' into his FM synthesis emulations of acoustic instruments, people perceived them differently. Could you talk a little about that?

Exactly. Well, I think that was in his work on the simulation of the piano tones. 'Stuff' was the kind of clack and noise that happens when the key hits the keybed and the hammer...the mechanism, the noises that come from the mechanism, not just the vibrating string. So he separated that out and included it in one of his versions of the piano tone. Some noises would increase with key velocity as if one were hitting the key harder.

And he played this example for a keyboard player who had played the DX-7, I think it was, and then showed him the new one with this new voicing and the player thought it was a different keyboard—it was more responsive to velocity, or strike force. So that was an interesting example of how the percept can influence one's evaluation of a physical system.

The program that bears Max Mathews' name these days—Max/MSP [12]—is something you've been using in your more recent pieces. Could you talk about the way you use Max/MSP, and perhaps why you use it rather than some other piece of software or hardware.

Well, let's see. There are two models for sound synthesis these days, I guess. One is like Max/MSP, which is object-oriented, and the other would be Csound, which is code, like the kinds of code that we wrote in the earlier years beginning with Music IV and then Music V. Max/MSP has the attribute that one can be listening to the sound as one changes parameters or changes the structure of the system. It looks like signal flow, and one can change or modify the signal flow as one is listening. So that's interesting, quite different than the early years of computer music which I described to you in relationship to developing FM for example, where there was always a time interval. So this is more like an analogue synthesizer in that sense, because one has real-time control over the sound as one listens to it.

I think each of them is equally powerful, that is the code model—say Csound—compared to Max/MSP, but I think it's a question of personal preference, in part. I can reach a degree of complexity with object-oriented, graphics programming that I don't think I can reach in code writing. Musical complexity, that is. In Max/MSP one has access to the information in patches and sub-patches, where one can open up a sub-patch that represents some process that one has worked on, decided upon, debugged and has an input or an output, or several inputs and several outputs. But then you close it and one has to name it, it's a mnemonic and it's a conceptual entity then, where we hide the complexity, and then that becomes part of another level of complexity, if you will. So it's sort of like a visual flow chart of nested complexities. That's very easily done in Max/MSP, and I find it very attractive, both at the control level and at the synthesis level. The whole idea of Max/MSP is interaction. so it's a platform that is ideal if one is writing music that involves real-time performance interaction. The piece that I've finished, and the next that I'm working on, are based upon—in the first case—a soprano, where the program senses the pitches, and a subset

of all the pitches she sings are target pitches that launch events and processes in the Max/MSP program. So it's very simple to implement, because it involves a small head-microphone, a small A-to-D convertor, D-to-A convertor and a firewire to a computer. So with a couple of wires, a laptop and a little box, one has a very powerful complete system that's easily transported—and moving the soprano becomes the big problem.

Are you using the Max/MSP to process the sound in ways that have not been possible before, or is it more a question of using the program to in effect listen to what the singer's doing and implement that in a way that is much easier now...or a bit of both?

Well, both. I'm using Max/MSP to sense the singer, then amplify the singer, then realize an artificial acoustic environment — reverberant space in which the singer's voice is projected through the speakers. But also the voice is used to trigger the synthesis from the Max/MSP as well. So the accompaniment, as it were, is triggered by the soprano's voice. And I'm using frequency modulation synthesis, but in a pretty complicated way. In this case it produces sounds that are quite un-FM-like, if one thinks about typical DX-7 sounds. The idea in this piece for soprano is to create what I call structured spectra, inharmonic spectra that are orderly and predictable and unlike most natural inharmonic spectra, which have lots of partials that are not determined by the people who build or make the instruments; bells, gongs, things of that sort. So with FM I'm able to create these complex orderly inharmonic spectra. And in this case they're based upon carrier-to-modulator frequency ratios that are powers of the golden section or golden ratio—much like I used in a piece in 1977, *Stria*. But I wanted to see if I could create non-*Stria*-sounding inharmonic spectra and have the soprano sing in a scale system that was complemented by these spectral components, and create a piece that is non-traditional in both its tuning and spectral space but doesn't sound out of tune.

You were saying that the sounds are not obviously FM-like. Is that because of the particular algorithms that you're using?

Well, I use FM much like additive synthesis. That is, I create complex spectra by using iterations of fairly simple FM algorithms added to itself. That's to be able to achieve spectral density—and change within that spectral density—without having to have large sweeps of the index, which produce a very typical FM sound—sort of like a filter being swept. So to achieve

a kind of nuance, then treating FM as if it were additive synthesis, building up a complex timbre by multiple iterations of the algorithm at different frequencies and adding them together, seems to be a very effective way.

Are you doing similar things in the forthcoming piece for horn, too?

Yes, that's right.

And perhaps using the horn spectrum in there somewhere—the particular pitch possibilities inherent in the horn?

Yes, which are unique of course. So that's one of the ideas. I'm just getting going on that piece.

I gather your new pieces are not the only works of yours that have benefitted from advances in technology—you now have cleaner versions of some of your older works.

Yes, the Samson Box has been reconstructed so we have a pristine version of *Turenas*, for instance.

Could you talk us through that? I think the version that was released on CD [13] was generated at a relatively low sample rate.

Yes, and also that version on Wergo is a digitized version, that was made from an analogue tape recording so...

It's two generations removed, then...

Yeah, yeah. Exactly.

Talk us through the revision of that piece...

I redid *Turenas* in 1978, I think, for the Samson Box. It was composed in '71, '72, so we made a version for the Samson Box, which was a real-time digital synthesizer with 14 bits of DAC, I think, and about 25kHz sampling rate.

About the same as the DX-7?

Well, no the DX-7 was greater. The first DX-7 was like 57kHz—some high but odd sampling rate, around 60kHz, then the DX-7II was 44.1. But the Samson Box was lower by an octave. In 2009 Bill Schottstaedt built a software version of the Samson Box such that all of our input files for the original Samson Box could be regenerated. So now I have a version of *Turenas* in four-channel sound files that are absolutely pristine—32-bit floating point—so it's quite a different listening experience.

Now I suppose the question that comes to mind, if we're thinking about 'historically informed performance practice' is whether you consider the new, pristine version to be closer to your original intention, or were you actually working with the 'grain' of the original hardware in your composition in the first place?

I would have wished for a noise-free version, yes.

So back then you would have preferred the current version because it's closer to what you were after?

Yes.

And the limitation was in the technology.

That's right. Tape hiss and digital sampling noise, the low-order bit flipping 0101, like in reverberation tails.

That was not part of the charm.

No, that was not desired.

What do you make of people who hanker after early analogue synthesizer sounds—and indeed early digital ones?

I think it's great. I'm glad. I don't think it affects one way or another the work that we've done. It renews a lot of music that was not being played, and I think it's a kind of interesting revival. There's something about the inherent warmth of analogue, which I think we understand now—that that kind of warmth, or the inaccuracies in analogue synthesis, the fact that nothing ever was quite in tune or stable and therefore these wandering frequencies produce a kind of ear-friendly effect that we worked very hard to understand in the early days of computer music.

This comes back to your early concerns with internal dynamism, as you call it.

Exactly, exactly. That's right, Jim. And it was kind of a natural part of the analogue world. So I think this rediscovery that's going on by this current generation of those sounds from 40, 50 years ago is kind of healthy.

Do any of the synthesis techniques that followed FM, such as waveguide synthesis and physical modelling have any interest for you?

Well, they do...sampling, of course, was what replaced most all synthesis, and that's partly because it's easy. One gets what one gets. And with slight modifications one can own it. With synthesis we had to understand a lot of very basic stuff that is no longer required with sampling. But in struggling with those synthesis issues we learned to understand aspects of music perception that we would otherwise not have understood. And that understanding now is part of the general knowledge—maybe of some of the people who use sampling—but they've never had to go back to buckets of zeroes and buckets of ones and deal with it at that level. But that some of us did is certainly good. It was an important payoff.

Do you see any future in things like physical modelling or even FM to generate sounds that can't be achieved by just manipulating samples?

Well I do, and that's why I use it when I compose! One aspect of composition that's really interesting to me is what I call structured spectra; the kind of control of spectrum and scale systems as I mentioned in the case of this piece for soprano. That was first done by Jean-Claude Risset in his piece *Mutations*. I think we mentioned that it opens up with this wonderful series of pitches that we then hear as harmony and then they're reiterated as a gong tone, but the gong is made not of typical inharmonic partials but actually the pitch frequencies of the preceding chord and melodic pattern. So there's a kind of connection between timbre and pitch that can't be achieved by any other means. Only with a computer could one do that.

There's a degree of control you have over all those parameters.

Right, and the fact that the spectral components are absolutely ordered. Well that's what I did in *Voices*, that's what I did in *Stria*, also *Phoné* to an extent—using this idea of creating sound where the link between the pitch system and the spectrum is tighter than that which exists—and can exist—in the natural world of acoustics, although that's part of the richness of tonal music: that low-order harmonics are complementary to the primary pitches of the harmonic system.

Or indeed in conflict with them.

Yes, right. Exactly.

That means also, from your point of view, that the compositional intent and the technological means to achieve them are closely knit.

Exactly, yes.

You mentioned earlier that in *Stria* the compositional procedure was analogous to a computational procedure—could you tell us something about that?

Stria is where I first used recursion. And that grew out of the fact that I asked “what's a recursive procedure?”, because it was part of the Stanford Artificial Intelligence language (SAIL). Somebody explained it to me...this was in the late 1970s. SAIL was an ALGOL [14]-like language, but it had some special features that had been developed at the A.I. lab. And in the manual I saw “recursive procedures”, so I asked someone and this was explained to me. Then I realized that I could make musical use of that, and so I used recursion in the piece, where a set of pitches, or a spectral structure, included an element that was then the root for an imitative version of that which it was a part—another set of pitches/spectral structure, kind of like the self-similarity in Mandelbrot images. And so yes, the idea of algorithmic composition was very interesting to me and is still. In both *Stria*, and *Phoné* I used the SAIL language into which I could put a set of parameters that then were used to generate many many many note events; what we used to call note events.

Are there any areas you'd like to explore in more detail now?

You've mentioned these relatively recent pieces that are combinations of a solo performer and real-time electronics.

What do you see as being fertile ground for experimentation or research?

Oh, well I think there are all sorts of combinations that I—given a long enough life—would like to explore. Not just solo instruments but combinations of instruments. I'm a percussionist and I have lots of interest in percussion instruments, and I watch with interest the development of controllers. Don Buchla and I run into one another often, so I keep track of what's going on in that world. So yes, I find it all very interesting.

Who do you think is doing especially interesting work or work that seems to have potential?

Well, at CCRMA [15] there's lots of work being done on controllers, also by David Wessel at CNMAT. [16] At CCRMA there are people doing very interesting work on controllers, the Haptic Drum [17] for example, which is a force-feedback system under a traditional drum head that allows a percussionist to

play a roll with one stick, for example. Well—lots of other things—but there's a force-feedback element there that's very exciting.

This is to do with the pressure on the drum head, presumably.

Yeah, that's right. The instantaneous pressure is sensed and run through a computer processor and then using a voice coil it forces the drum head back up, so it can produce a response at various rates depending on the program and the strike force etc. There are quite a lot...I mean Max Mathews; his radio baton is used—we see a lot of that around CCRMA.

It sounds like a lot of the work is on the physical controllers rather than the synthesis engine itself. Would that be fair to say?

Yeah, I would say that's true. In general, I think people use sampling, or controllers to activate or process sampling. It's probably the most common use of controllers, really.

My own feeling is that the evolution of controllers—that is the devices with which we control this enormous power that we have available to us—can't be very distant from the existing musical technologies because it's driven by the notion of virtuosity. That is, most of us who are musicians heard some fabulous player when we were young and we try to emulate and we decided that it was worth the hours and hours and years of study to learn to play well an instrument like a violin or a piano or a trumpet or drum or whatever, because the payoff was so great in repertoire, especially the well-known instruments like piano, and stringed instruments, guitar. The repertoire could be performance traditions, as in jazz, but there's some body of music that drives us to spend the time to learn to play well. So I think that controllers that are based upon existing techniques...for example there's a new one [18] that's just been produced in England that looks like a cello fingerboard but it has a breath controller and maybe a continuous slider, all of which can be useful, but it's based upon things that people have learned for other reasons, because there's a large repertoire that makes it worth while spending the time to learn to play well.

And it responds well to extremely fine motor control.

Exactly, yeah. And that's the other important thing. I think that new controllers have to have at least one dimension of control that is highly nuanced, so that fine distinctions can be made between gestures. A good example of that is the piano. It's very simple...you know, there's onset time, release time, and then the strike force. But that strike force is an extremely nuanced

dimension, And of course there's onset time and release time. So a piano has just three dimensions on any key but multiply that by ten fingers and a couple of feet and the dimensionality is very great, and very rich.

How did CCRMA come into being and what was your involvement with it?

CCRMA was founded about ten years after the work started in 1964, so in fact in 1975 the four of us—Andy Moorer, John Gray, Loren Rush and myself, and Leland Smith who was then working on the Score music notation programs. So although the work had been going on since I began in '64 with people joining along the way, we formed CCRMA to have our own administrative entity, which made it easier to get funding and manage things than having to go through the music department. And also the fact that one has an entity...it gave a kind of unity to an effort. That was helpful, I think, in attracting students, and people from other institutions.

What do you think are the key things that have come out of CCRMA over the years?

Well from my point of view, the music is probably the most important, but I mean certainly the technologies that have been developed—the work in signal processing, beginning with my own work in FM synthesis, but Andy Moorer's work in reverberation, Julius Smith's work in physical modelling, the work in psychoacoustics that's been produced, like John Gray's seminal dissertation paper *An Exploration of Musical Timbre* was really important. And gee, the number of students, and what great music that's been done. Bill Schoedstatt, and Mike McNabb's *Dreamsong* is the piece that inspired me to do *Phoné*.

[1] The Samson Box was the nickname for the Systems Concepts Digital Synthesizer, (1977) designed by Peter Samson.

[2] The Advanced Research Projects Agency Network, a precursor of the internet that was established in 1969 to allow computers at different institutions to communicate with each other.

[3] The IBM Mathematical **Formula Translating System**: an early computer language, introduced in 1957.

[4] 'The Digital Computer as a Musical Instrument', published in *Science*, November 1963.

- [5] Frequency Modulation, specifically as applied to sound synthesis.
- [6] For Bresnick's recollections of this film score, see: <http://opinionator.blogs.nytimes.com/2011/05/25/prague-1970-music-in-spring/>
(<http://opinionator.blogs.nytimes.com/2011/05/25/prague-1970-music-in-spring/>)
- [7] *The Technology of Computer Music*.
- [8] 'The Synthesis of Complex Audio Spectra by Means of Frequency Modulation', *Journal of the Audio Engineering Society*. The footnote mentioned is the second in the article.
- [9] Mr. Ichimura.
- [10] Strictly speaking, Yamaha's 'FM' was an implementation of phase modulation, but 'FM' is the term universally associated with the synthesis technique, the Yamaha instruments and the sounds they typically produced.
- [11] The other programmer of the DX-7's factory voices was Gary Leuenberger.
- [12] Now called simply 'Max'.
- [13] WERGO WER 2012-50, released in 1988.
- [14] **ALGO**rithmic Language, another early computer language which appeared in 1958.
- [15] The Center for Computer Research in Music and Acoustics at Stanford University.
- [16] The Center for New Music & Audio Technologies at the University of California, Berkeley.
- [17] See <https://ccrma.stanford.edu/~eberdahl/Projects/HapticDrum/>
(<https://ccrma.stanford.edu/~eberdahl/Projects/HapticDrum/>)
- [18] The Eigenlabs Eigenharp. See <http://www.eigenlabs.com/product/alpha/>
(<http://www.eigenlabs.com/product/alpha/>)