Interview with Max Mathews
Author(s): C. Roads and Max Mathews
Source: *Computer Music Journal*, Vol. 4, No. 4 (Winter, 1980), pp. 15-22
Published by: The MIT Press
Stable URL: http://www.jstor.org/stable/3679463
Accessed: 28/01/2010 22:31

C. Roads

Interview with Max Mathews

Introduction

Max Mathews is a pioneer in computer music, having developed the first sound synthesis programs in the late 1950s at Bell Laboratories. He is the author of the classic text on the subject, *The Technology of Computer Music* (1969, The MIT Press), and of numerous papers on the application of computer technology to music and acoustics.

This interview took place in Cambridge, Massachusetts in late June 1980. Dr. Mathews had recently returned from a research visit to Institute de Recherche et Coordination Acoustique/Musique (IRCAM) in Paris, where he and Curtis Abbott had worked with his Sequential Drum and its interconnection to the 4C Machine.

Background

Roads: First, I'd like to ask you a little about your background; where you went to school, what you studied, and how you got interested in computer music.

Mathews: I studied electrical engineering in all my schools; Cal Tech first, and then M.I.T. I eventually got a Doctor's degree in electrical engineering. At that time and ever since I've been interested in large, complex systems and in computers; first in analog computers and then, when they became practical, in digital computers. Leaving M.I.T., I went to Bell Labs for my working career. I worked in the Acoustics Research Department applying digital techniques to speech transmission problems and, eventually, to music. I've always enjoyed music as an amateur musician.

Roads: I noticed in your demonstration and performance out at Stanford that you at some point learned to play the violin. Did you study violin as a child?

Mathews: I studied violin through high school and have continued to play it ever since, taking a few lessons but not many. I like the instrument very much personally, although I call it an "inefficient" music instrument in that you have to practice more to achieve a given musical performance than with almost any other instrument. Anyway I'm stuck with the violin; I'm certainly not going to learn any other instruments except the computer and new instruments I might invent myself!

Computer Music Journal, Vol. 4, No. 4, Winter 1980,
0148-9267/80/ 040015-08 $04.00/0

Music I

Roads: It must have been in the early and mid-1950s that you started applying computers toward musical goals. Can you tell us a little about your experiences with the program Music I? You brought that up on an IBM 704 computer?

Mathews: That was the only computer we had that was capable of doing sound processing.

Roads: That was at Bell Labs?

Mathews: Actually the 704 computer was in New York City at IBM World Headquarters on Madison Avenue. We used to go in there and run our programs and bring a digital magnetic tape back to Bell Labs. The digital-to-analog converter was at Bell Labs.

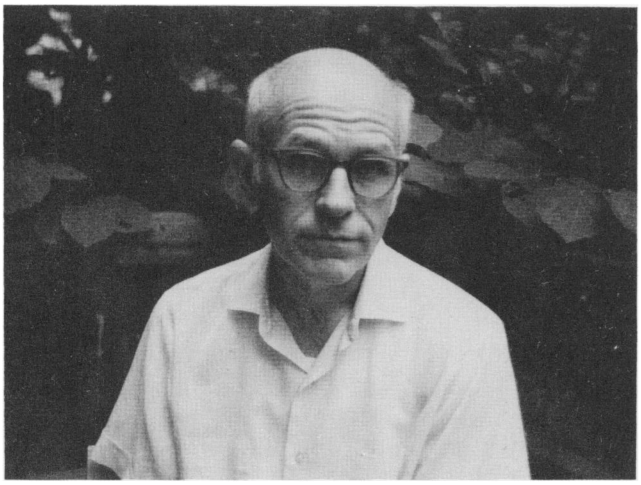
Roads: What was the converter like?

Mathews: We had a 12-bit vacuum tube converter; it was quite a nice machine. It was made by a company called Epsco, I believe.

Roads: Music I was capable of playing melodies, was it not?

Mathews: It generated one waveform, an equilateral triangular waveform, with the same rise as decay characteristics. You could specify a pitch, and an amplitude and a duration for each note and that was it.

Fig. 1. Max Mathews, June 1980, Cambridge, Massachusetts. (Photo by C. Roads.)



Roads: Had you heard of anyone else doing anything similar at around that time?

Mathews: No. There were some people, Lejaren Hiller in particular, who had done a bit with composing music—compositional algorithms—but as far as I know there were no attempts to perform music with a computer. In fact, we were the only ones in the world at the time who had the right kind of digital-to-analog converter hooked up to a digital tape transport that would play a computer tape. So we had a monopoly, if you will, on this process.

Roads: This was 1957?

Mathews: Yes.

Roads: Some people made compositions with Music I, is that not true?

Mathews: One brave soul, a psychologist named Newman Guttman, made one composition. But it was not hard to realize that we could do better. Music I sounded terrible and was very limited. It was clear that if we were going to get better music out we had to do better. It wasn't hard to see things that could be added and changed.

Music II

Roads: So you developed Music II. As I recall, Music II was capable of four independent voices of sound, and a choice of 16 waveforms stored in memory. Some hardware synthesizers today seem to give you this.

Mathews: True. The hardware synthesizers probably have to repeat the trajectory of the software programs.

Roads: That must have been 1958. Was this still on the 704 computer?

Mathews: By that time we had moved to the IBM 7094 at Bell Labs. That was a very, very effective machine. We used it for almost a decade, not only for musical purposes, but for our great preponderance of computations at Bell Labs in speech processing and visual signal processing. That was a fine machine and some of the notable early operating systems were developed for that machine, including Bellsys 1 and Bellsys 2.

Music III

Roads: In Music III, which you introduced in 1960, you introduced also the concept of the *unit generator*, which is certainly one of the major conceptual advances which has made computer music possible today. Can you describe why that became a necessary concept?

Mathews: I too think it's a very important concept, and more subtle than it appears on the surface. I wanted to give the musician a great deal of power and generality in making the musical sounds, but at the same time I wanted as simple a program as possible; I wanted the complexity of the program to vary with the complexity of the musician's desires. If the musician wanted to do something simple, he or she shouldn't have to do very much in order to achieve it. If the musician wanted something very elaborate there was the option of working harder to do the elaborate thing. The only answer I could see was not to make the instruments myself—not to impose my taste and ideas about instruments on the musicians—but rather to make a set of fairly universal building blocks and give the musician both the task and the freedom to put these together into his or her instruments. I made my building blocks correspond to many of the functions of the new analog synthesizers.

I wouldn't say that I copied the analog synthesizer building blocks; I think we actually developed them fairly simultaneously. In any case, that was an advantage because a musician who knew how to patch together Moog synthesizer units would have a pretty good idea how to put together unit generators in the computer.

Music IV

Roads: Yes. Music IV followed Music III in 1963. What was the advantage of Music IV over Music III?

Mathews: Music IV was simply a response to a change in the language and the computer. It had some technical advantages from a computer programming standpoint. It made heavy use of a macro assembly program which existed at the time.

Roads: So up until that point you had been programming in assembly language without a macro facility?

Mathews: Macro assemblers were just invented at that time. Indeed, Music IV debugged a lot of the macro assembler that was used at Bell Labs. It made very heavy and rather sophisticated use of the macro facilities, and I discovered a lot of bugs in them that the designers hadn't anticipated, and that they were glad to fix.

So in essence Music IV was musically no more powerful than Music III and was only a little more convenient to use, but it was computationally quite a bit more sophisticated.

Roads: From Music IV a number of people made their own versions, like at Princeton.

Mathews: Music IVB and IVBF and things like this were developed at least with the inspiration of Music IV.

Music V

Roads: Was the need to get these programs out into the world part of the motivation for developing a machine-independent version of Music IV, that is, Music V?

Mathews: Certainly. There were several motivations. One was that again we'd changed computers. So I had to contemplate rewriting the program once more and I wanted to make a universal program. Also, I wanted to make it as universally available as possible. At that time the Fortran compiler was available and was the most widely used compiler. I was able to work out what I still believe was a very ingenious technique so as to have almost all of the complexity of the program encoded in Fortran statements which are portable, but have the inner loops of the unit generators, which are computationally rather simple but get executed so many times that they put a heavy load on the computer, programmed in machine language. So the overall program was both simple in terms of the amount of coding required to put it on a new and different machine, and efficient in terms of running rapidly.

Roads: Even though you had developed Music V as a means of spreading computer music out into the world, a number of musicians had already got wind of what was going on at Bell Labs and were starting to go there to find out more about what you were up to. Some of your colleagues, including F. R. Moore, Jean-Claude Risset, and James Tenney, were able to do some very significant work at Bell Labs around that time. Can you describe the atmosphere at Bell Labs? Were composers visiting?

Mathews: When we first made these music programs the original users were not composers, they were the psychologist Guttman and John Pierce and myself, who are fundamentally scientists. We wanted to have musicians try the system to see if they could learn the language and express themselves with it. So we looked for adventurous musicians and composers who were willing to experiment. The first one was David Lewin, who was at Harvard at the time. We corresponded and he did a composition mostly by mail, which was a brave thing to do. Then John Pierce met Jim Tenney at the University of Illinois, where Tenney was studying with Hiller. Pierce was very much impressed with Tenney's music and his interest in computers. He invited Tenney to take a temporary job at the Laboratories to try out the music programs. Tenney took the job and developed some timbres of his own and also some pieces. To my mind, the most interesting music he did at the Laboratories involved the use of random noises of various sorts.

Roads: Yes, his *Noise Studies*.

Mathews: Then after Tenney, Jean-Claude Risset was sent to the Laboratories on a French scholarship to do a thesis in physics. Risset has a Ph.D. in physics. His thesis involved analyzing the timbre of the trumpet and developing a new technique for analysis called *analysis by synthesis*, which is still, I think, the most powerful technique for analyzing natural music sounds.

Early Reaction to Computer Music

Roads: Computer music today seems to be gaining acceptance at least in some limited quarters. There are computer music centers established, and modern composers recognize that some form of computer literacy is an important part of their compositional training. But what was the early reaction both of the public and musicians to your demonstrations of computer music? It must have been quite a shock to some people.

Mathews: The reaction amongst all but a handful of people was a combination of skepticism, fear, and a complete lack of comprehension. Amongst musicians, the group that was the most interested in the computer was the composers, while the group that was the most antagonistic to the computer was the performers. The group that understood the computer the least was "the general audience." The rock and popular musicians were willing to think about the possibilities, but they also have the same difficulties anyone dealing with a well-established musical technique has. They are used to well-known and hence powerful methods for making sounds. They're not very patient with "new" techniques which are quite weak and tedious to use, and which certainly require much more experimentation.

Graphics Interaction

Roads: That leads to my next question. When I look back over your work, I see a very logical progression. You developed this series of flexible sound synthesis programs culminating in Music V, which is still in use today and which essentially has not been surpassed in terms of flexibility. But there was another rigid aspect that was part of the Music V process which was the input process. I know that you turned to graphical systems of interaction at one point. What led you to that?

Mathews: I was led to it by a desire to broaden and make more facile the techniques for specifying compositions. I think my graphical experiments were very interesting; they did not, though, lead to a language which is as universally understood as the Music V language is. In some sense, I think today the Music V language is much more important than the Music V program, in that almost anyone involved in computer music can read a Music V score or read a description of a Music V instrument with unit generators and understand it, and translate it into whatever language he or she is using, whether it be Music 11, Music 10, or Music 360. It provides a well-documented and universally understood way of describing a sequence of notes and their interpretation and musical instruments.

The graphic languages didn't get to that level of generality. I could describe an accelerando with a single line, where the ordinate value of the line was the tempo, so if it sloped upward the tempo was increasing and if it sloped down it was decreasing, and that was quite a nice, easy way of making accelerandos. I had some experiments for drawing melodic lines; that made graphical sense.

GROOVE

Roads: After you developed this flexible sound synthesis system and a means of graphic interaction with this system, what did you turn to then?

Mathews: I got into graphics at the end of Music IV's history, and I did Music V as the next thing. My graphical input work was at the very end of the life of the 7094 computer. I spent a time getting Music V in operation, then Risset carried out the brunt of Music V work at Bell Labs. He worked on analysis-by-synthesis techniques and a catalog of computer-generated sound. After Music V I got interested in real-time work and developed the

GROOVE system. That was quite a different direction.

Roads: That was with F. R. Moore?

Mathews: Yes, he had come to Bell Labs by that time.

Roads: The GROOVE system, as I recall, was a hybrid system consisting of a minicomputer connected to an analog sound synthesis system, with a number of input devices such as joysticks and knobs for conducting a score which had been prepared beforehand and input to the computer. It also incorporated graphics, did it not?

Mathews: It had graphical displays. The point about GROOVE was that it looked upon the score as a recording of the control functions for the analog synthesizer. These are the functions of time which specify the way the frequencies and all the other things one controls in an analog synthesizer change with time to make music. Now, it basically recorded these time functions in sampled form at about 100 to 200 Hz, fast enough to record human gestures, and would store this on disk. The program was a flexible way of playing back these functions and combining them with other functions of time that were generated by the performer playing on the *sensors* of the instrument. These combined functions were then used to control the analog synthesizer. In addition, the program had very good editing facilities so you could go back to these stored functions and change them. You could change one sample of a single function without affecting anything else.

Roads: So you could get a display or printout of these functions, allowing musicians to perfect what they had done in real-time—edited improvisations, so to speak.

Mathews: Exactly. The functions were displayed on a scope and you could move to a particular sample in the function and you would hear the resulting sound as a sustained sound and you could flip the editing switch on and change the value of the function at that point and hear what it was doing to the sound.

Roads: Yes, editing with real-time feedback. As far as I know the most extensive use that was made of this machine was by Emmanual Ghent. Was there anyone else who worked with this machine?

Mathews: You're certainly right that Ghent worked with it more than anyone else. F. R. Moore made plenty of compositions with it. Boulez and I worked with it on the Conductor program which was based on the GROOVE system.

Roads: When was that?

Mathews: That was 1975 and 1976.

Roads: The GROOVE system, which was developed in 1968, ran, as far as I know, up until 1979, didn't it?

Mathews: Yes, eventually we couldn't maintain the computer it ran on. I haven't reprogrammed it because I think one should now use a digital synthesizer. Also I think a different way of storing the score is better and more appropriate now, such as the way Curtis Abbott has used at IRCAM.

The Sequential Drum and 4CED

Roads: I understand that you have recently returned from IRCAM, where you attached a new instrument called the Sequential Drum to the 4C digital synthesizer.

Mathews: Yes. The Sequential Drum is another sensor which you hit like a drum. The sensor sends three signals to the computer. One of these is a pulse which tells the computer when the drum is hit and how hard it's been hit, and the other two say *where* you hit the drum in terms of x and y coordinates. The computer and the synthesizer use these three pieces of information to synthesize a sound. The musician decides how to use this information in controlling the sound.

In my case, I tried an interesting principle in which the pitches to be played are a sequence in the computer memory. Each time you hit the drum you automatically get the next pitch in the sequence. I did this because most traditional music has a very rigid pitch line and the musician is not allowed to deviate from the composer's intention.

I actually built two drums; one I used at Bell Labs and the more recent one I took to Paris. The computer there was the PDP-11/34 and the synthesizer was the 4C Machine. I was very fortunate that Curtis Abbott had written a program which I could adapt to this sequential principle. It's a program one can do many things on. It is one of the first general real-time programs. I think it's a good program partly because it resembles Music V and Music 10. If you've learned these languages it becomes quite easy to learn Abbott's program. It's also a good program because he has figured out the right structure for describing a series of complex events in a score and for synchronizing and controlling events which are determined by a score with events determined by the input sensors like the drum.

Roads: As I understand it, in Curtis Abbott's program, wherein a traditional *pfield* in a language like Music 11 might specify a value which would be typed in by a composer, in the 4CED program an instrument, instead of being fed by a pfield would be fed by an external event which might occur at any time.

Mathews: That's right. A note can be initiated by a performer or by a score.

Roads: These scores themselves can be structured into rather complex entities. One score event might trigger a whole series of score events. One can, for instance, attach a subscore to a particular event. So not only do you have this complex instrument connection to real-time, you also have a complex score connection to real-time, as I understand it.

Mathews: That's quite correct. The score controls the real-time processing of events in the 11/34 computer. It allows one to put together a whole series of events in a treelike structure where one starts going up the trunk and when one reaches a side branch that branch starts whatever it is doing and at the same time one proceeds on up the trunk perhaps starting other side branches. So you can get a lot of things going simultaneously. The side branches all end, and the trunk continues on, so the overall synchronization is quite simple since it all depends on how fast you climb the trunk.

Electronic Violins

Roads: This Sequential Drum is only one in a number of new musical instruments you've developed, including a new violin.

Mathews: Yes, I've made a number of electronic violins. Some of them are closely related to normal violins, in that I've worked very carefully on the resonances of normal violins, and made electronic circuits which introduce these same resonances into the electronic violins. The electronic instruments all have regular violin strings and they all have bows. So the source of the vibrations is the same as in a regular violin. But thereafter the sound gets modified electronically. It is possible to tune the resonances very accurately, and it is, of course, possible to have as much energy as you want in the sound. It also makes it possible to use some of the things we know about timbres to change completely the timbre of the violin. Notably, we can make it sound like a brass instrument or like a human voice. The other thing I've most recently done is incorporate *volume expansion* to increase the dynamic range of the violin. The normal violin I think suffers from having a very narrow dynamic range. One can expand the entire spectrum of the violin, but it's more interesting to expand only a portion of the spectrum. This way you not only get a change in the dynamic range, but you get a marked change in the spectrum.

I've also made some physical changes in the violin. My latest instruments are played in a vertical position, as one plays a cello. I've learned to play the violin that way myself, and I'm convinced that that's a superior physiological position. In short, I'm going back to the old style of the viols. I've introduced some other human-engineering changes—some very low frets which one feels with the fingers but which do not constrain the pitch of the string to the pitch of the fret. So one can do vibrato or a glissando and for that matter, one can tune the scale to whatever variations one might want.

Intelligent Instruments

Roads: Do you see a time when families of instruments such as the ones you've developed would be used in concert for a new kind of musical performance—an orchestra of intelligent instruments?

Mathews: In truth, I don't know whether the intelligent instruments will be used in concert form or whether they'll be used in the home, or whether both will happen. Certainly one could use the increased capabilities to enhance performances by virtuosos. One could also use the capabilities to make a wider range of music available to the amateur player.

Roads: Do you also see new kinds of musical interaction resulting from the use of intelligent musical instruments? One of the implications of your Sequential Drum is that the performer is in a new kind of relationship with both the score and the instrument.

Mathews: That's certainly true. With the intelligent instruments both the score and the control from the player form separate inputs to the instrument. The score does not have to pass through the performer as it does with traditional instruments. Also, the score doesn't stand between the musician and the instrument, as with Music V. So I think this is a much more powerful and flexible arrangement.

Also, we will have very interesting situations when we have several intelligent instruments and several performers interacting with one another, where part of the interaction flows directly from one instrument to the other. That's a very pregnant situation.

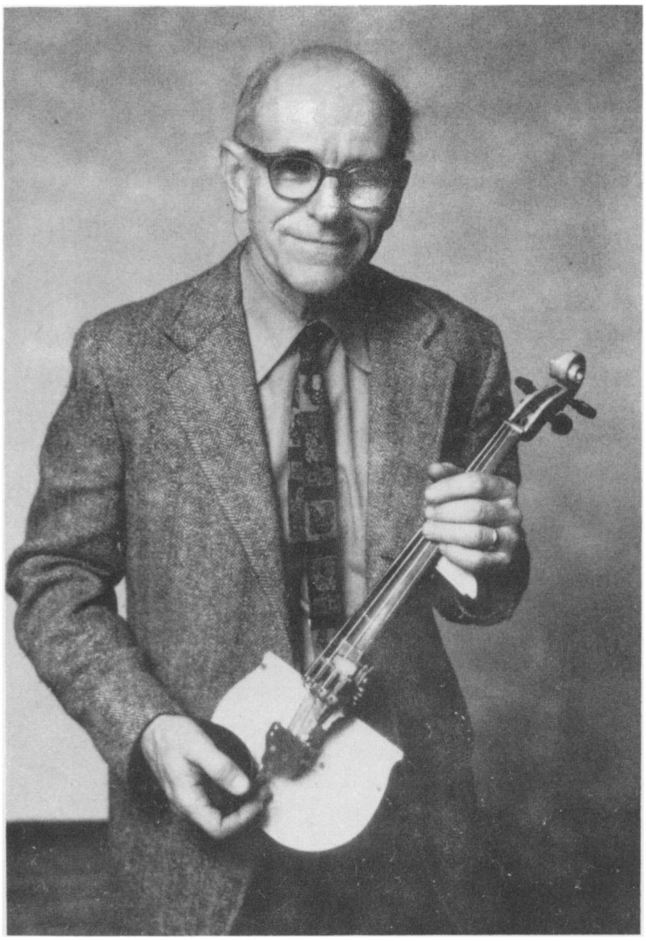
Roads: This implies that these intelligent instruments will be in some way cognizant of each other. Do you foresee a day when these instruments will actually be able to listen to one another the way that musicians listen to achieve ensemble quality?

Mathews: I don't think they'll listen with microphones. I think the basic information of what each instrument is doing will be transmitted to the other instruments over a digital channel. This gets into another important question of how one makes multiple, independent computers work together on a single problem. That's one of the most interesting unsolved computational problems today.

Experiments with Inharmonic Timbres

Roads: You've recently concluded a series of experiments dealing with "stretched" inharmonic tones. What was the motivation behind that project, and what were the results?

Fig. 2. Max Mathews with one of his electronic violins. (Photo by Nathaniel Tileston.)



Mathews: It's clear that inharmonic timbres are one of the richest sources of new sounds. At the same time they are a veritable jungle of possibilities so that some order has to be brought out of this rich chaos before it is to be musically useful. So John Pierce and I have been studying one small class of inharmonic sounds, those which have overtones like normal sounds except that the overtones are stretched farther apart than normal overtones or compressed closer together than normal overtones.

Roads: So the relationship of a pseudooctave might be 2.2/1 or 2.3/1, not 2/1?

Mathews: Exactly. Our initial experiments were aimed at finding out what properties of normal harmonic music carried over to music that was made with stretched overtones. We found some things carried over and some things did not. The sense of "key" carried over better than we expected.

Roads: So you can actually detect "keys" in sequences of completely inharmonic sounds.

Mathews: That's right. You play two samples and a person can reliably say whether they're in the same or a different key.

Other properties do not carry over. The sense of finality in a traditional cadence does not carry over. A person who hears a cadence with unstretched tones says, "That sounds very final to me." When he hears the same cadence played with stretched tones, he'll say, "That doesn't sound especially final." But we have been able to make other inharmonic materials which do convey a sense of cadence.

Roads: Yes, very strongly so in the one's I've heard. These modified sounds relied on the critical-band phenomenon, as I recall.

Mathews: They were related to a theory of the critical band and how it affects the consonance and dissonance of sounds, which was primarily developed by Plomp and Levelt.

Roads: If we can detect "keys" and some form of finality within a cadence or progressions within inharmonic tones, then some of the theories of harmony in the past must not be as cogent as some of their proponents have thought them to be.

Mathews: Our results are contradictory. We looked at two theories. One was the Rameau theory of the fundamental bass, and the other was the Helmholtz and Plomp theory of the consonance and dissonance of overtones. The destruction of the cadence would support the Rameau theory and the persistence of the sense of key would support the Helmholtz and Plomp theory. So we have one result which supports one theory and one which supports the other, with the overall conclusion that the world is a more complicated place than we had perhaps hoped it was. We will have to dig deeper before we can say what is causing the various perceptions we find meaningful to music. But unanswered questions such as these make life interesting.