

# I Dig A Pygmy — an homage to John Lennon

## Composition notes

By Paul D. Lehrman

(originally printed in [Sound on Sound](#) magazine, December, 1993.)

This is the story of the creation, evolution, and performance of a new kind of piece, one that takes full advantage of some of the latest sampling, synthesis, and processing technology. Like most stories about new technology, it doesn't really have an ending, but so far, it's been a fine education, not to mention a genuine pleasure, to experience.



The story starts in the fall of 1990, when the University of Massachusetts-Lowell, where I was teaching, announced plans to produce a tribute concert to John Lennon, on the 10th anniversary of his death. Will Moylan, the coordinator of the Sound Recording Technology program, asked me if I wanted to compose and perform something. I had recently acquired a Roland S-770 sampler, and thought this would be an excellent opportunity to see what it could do. The result was "I Dig a Pygmy", a piece of "live musique concrète" whose sound sources consisted entirely of the utterances and recordings of John Lennon and the Beatles.

### A little historical context

In the early days of electronic music, before synthesizers, composers got sounds wherever they could. They travelled around to places like factories, zoos, and airports with portable tape recorders, collecting sonic material that could be processed and spliced together in unusual ways back in their studios. Many of these "musique concrète" composers experimented with the human voice, modifying it in various ways. Two noteworthy examples were Karlheinz Stockhausen's "Gesang der Jünglinge", which combined the voices of a boys' choir with other natural and electronic sounds, and Luciano Berio's "Thema (Omaggio a Joyce)", which used a single sound source — the voice of singer Cathy Berberian reading a passage from James Joyce's "Ulysses" — to create an unearthly, evocative soundscape.

For the tribute to Lennon, I wanted to use spoken and sung materials in the same way as Stockhausen and Berio, but to add two other dimensions. Those early pieces exploited the sound of the voices for their emotional and musical content, but the literal meaning of the text (if there even was one) was incidental — the Joyce passage

that Berio used, for example, is highly abstract. I wanted to make sure the samples I used were intelligible, so they would have literal value for the audience, as well as aural: what Lennon was saying or singing was to be as important to the piece as the sound of him saying it. Maintaining a balance between intelligibility and sonic interest created a tension in the design of the piece I found quite challenging.

The other added dimension I envisioned was that while tape composers labored over their creations for months to emerge with a recording, I wanted to do a live piece, with as much improvisation as possible, in front of an audience. In essence, I wanted to create a "hyper-instrument", made up of samples of Lennon. First I would build it, then I would learn it, and then I would attempt to perform on it.

## The samples

First I went to the recordings of the Beatles and Lennon, and sampled selections from a number of CDs, through the S-770's digital inputs. I was particularly interested in samples that had strong, loopable, rhythmic qualities, like the guitar licks that open "And Your Bird Can Sing" and "Hey Bulldog", or the opening drone, percussion, and bird sounds from "Tomorrow Never Knows". I also looked for samples whose words had particular relevance to Lennon's life story, like "the man in the crowd" from "Happiness is a Warm Gun" and the "Mama don't go" scream from "Mother". Plus I looked for sections of music that were abstract and non-verbal, which could lend themselves well to real-time processing like speed changing and filtering, as well as looping. These included the weird section in the middle of "I'm the Walrus", the cascading bells from "...Mr. Kite", and the guitar feedback at the end of "Helter Skelter".



For the spoken samples, I found copies of the videos of the "Live-Peace" concert in 1970 in Toronto, the film "The Compleat Beatles", and the prophetic interview Lennon did with Tom Snyder in New York for NBC television, shortly before his death. I sampled fragments of them, as well as some other broadcast interviews I came across.

By now I had amassed about 60 megabytes of samples. The CDs were sampled in stereo at 44.1 kHz, while the spoken materials were sampled at 22.05 kHz in mono. As the S-770 can only hold a maximum of 16 Mbytes of sample RAM, this was a tad much. Since I was planning to do all sorts of panning tricks in the piece, I decided stereo samples were unnecessary, so I made a big dent in the storage requirements by resampling the CDs in mono. I also did a lot of trimming to make sure I wasn't using any more memory than I had to.

## **Structuring the piece**

My next task was to come up with an overall structure for the piece. Turning 10 minutes of samples, no matter how brilliant, into a coherent piece which can hold the attention of an audience, is not a trivial undertaking. I stewed over the possibilities for several weeks, fooling around with the samples and juxtaposing them in different ways, until very early one morning I woke up with literally the complete structure in my head. I dashed to my computer, wrote it down, and went back to bed. When I awoke again three hours later, I was pleased to see that what I had written still made sense.

The structure I came up with consisted of five sections, each with a different set of samples mapped across the keyboard. Program changes would be used to go from one section to the next. The sections would be played in a pre-determined order, and the beginning and end of each section would be worked out, so that the transitions between them would be predictable. What happened within each section, however, I would be able to improvise. The length of each section was therefore very flexible. If something was working great, I could keep it going for a while, but if it was getting boring, I could exit at any time. I figured on 2-3 minutes for each section, making for a 10-15 minute piece.

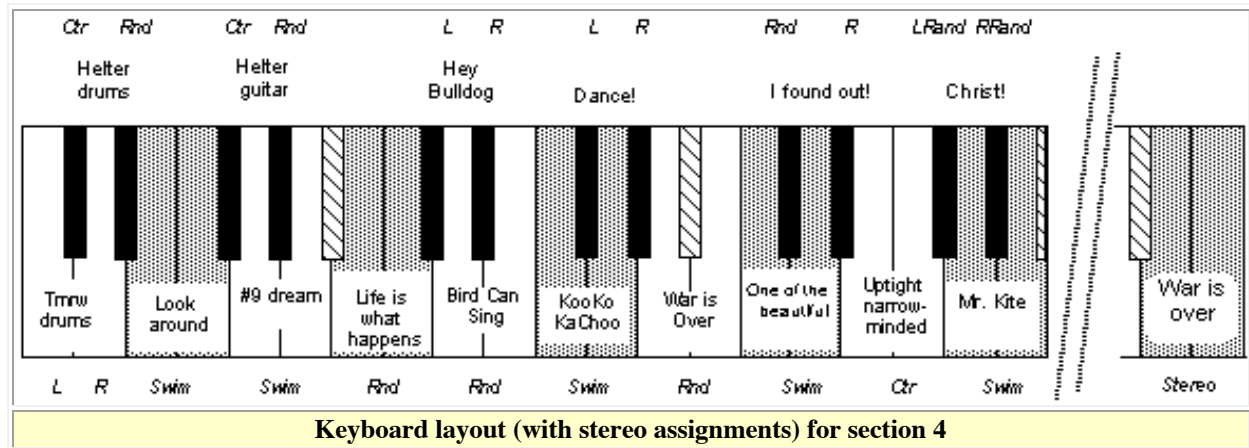
The sections alternated as to the kind of material they contained: sections 1, 3, and 5 were mostly spoken samples, while 2 and 4 were all music. Each section's samples were joined by a unifying theme. Roughly described, they were, in order: ironic humor; mystery and dread; Lennon's relationship with his audience; the opposition of anger and pacifism; and religion, violence, and death.

Once I had the structure determined, the piece nearly wrote itself. It became easy to decide which samples to use, and how to map them. To conserve space, I downsampled some of the music samples, and eventually got everything I needed to fit into 16 Megs.

## **Working with the samples**

Next came the task of making the samples interesting. Playing them straight would get boring after a couple of minutes, both for me and the audience. So I created loops in almost every sample, plus I gave them fairly long release times, which allowed them to crossfade into each other. The S-770 lets you design two loops for any sample: sustain and release. I set up the loops and envelopes in many cases so that while I held down a key, the middle of a phrase would repeat, and then when I released the key, the end of the phrase would repeat and fade.

Panning was another feature I used extensively. The S-770 lets you place individual samples anywhere within the stereo field, and has a "random" pan function, which puts the sound in a different location every time it fires. The pan is static: the sound stays put once it starts. I took great advantage of this, flying things all over the room.



I noticed an interesting phenomenon as I laid down the samples, particularly the heavily rhythmic ones, on the keyboard. Many of the tempos of the different sample loops worked very well with each other. They weren't all the same length Ñ some were 4 beats, some 5, some 8, and so on Ñ but they often fit together. Either the beats were the same, or there was an obvious rhythmic ratio between them, like 3 over 2. By fine-tuning the pitch of the samples, I could make them fit perfectly. For example, by tuning the "Hey Bulldog" loop up about a quarter-tone, I could make its 8 beats fit exactly twice into the 16 beats of "Tomorrow Never Knows". I could also make a 6-beat loop of the word "dance!", from "Woman is the Nigger of the World", fit into that same space, as I could a 12-beat loop of the verse from "Baby, You're a Rich Man". This allowed me to design section 4, which consisted entirely of music loops, so that it had a constant, highly complex internal rhythm, as the loops piled on top of each other.

I decided a couple of samples could benefit from real-time filtering control, so I mapped a foot pedal to the center frequency of a bandpass filter laid on top of those samples.

## Processing

The S-770 has six individual outputs in addition to the main stereo outs. I used one of these individual outputs for special signal processing. This was handled by an AKG ADR 68K, which has two inputs and four outputs, and can operate as two independent processors. The main outputs of the sampler were mixed and fed to one side of the ADR 68K, which was set to reverb, while the individual output was fed to the other

side, which changed from chorus to delays, flanges, and other effects. Each section had its own type of processing: the program changes used to select the sample map also were used to select the processing program on the ADR 68K.

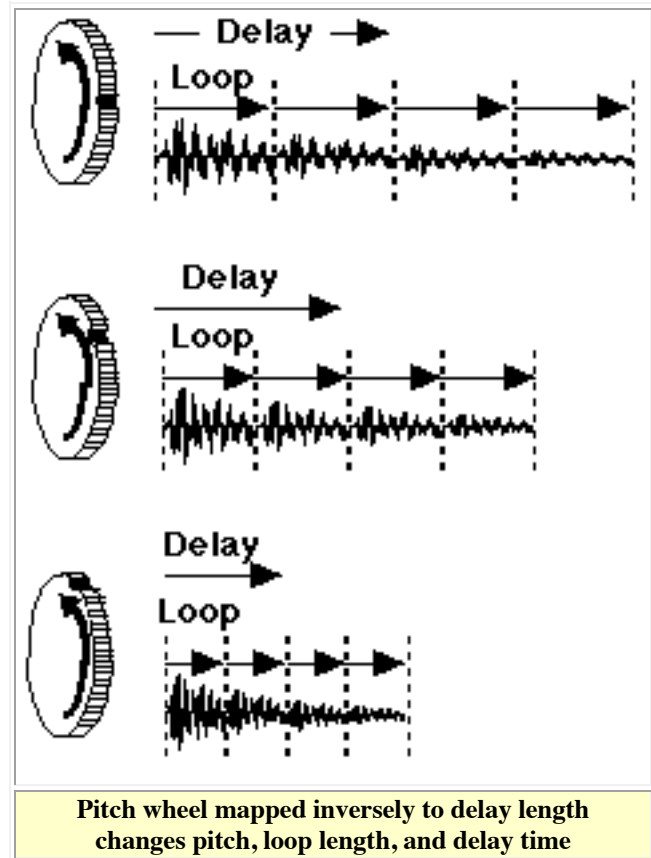
In addition, several real-time MIDI controls were mapped to effects parameters. For example, at the end of section 3, I played a loop of "Mama don't go", and brought up the pitch with the pitchbend wheel. At the same time, the loop was fed through a delay with heavy feedback on the ADR 68K. I mapped the pitchbend wheel inversely to the delay time, so as the pitch went up and the loop got faster, the delay window got shorter. It took a little while to get them to track exactly, but the result was very effective.

### Practice and performance

After finishing the sample tweaking and mapping, I had about a week to rehearse the piece and get used to this monster I had created. I made charts for myself showing the key assignments in the five sections. I dug up an old DX7 to use for the controller keyboard, and had to explain to the audience that, no, it wasn't making any of the sounds. The performance went quite well, and got a very enthusiastic reception. I noticed some glitching when I changed effects programs that hadn't seemed to be a problem in my studio, but otherwise I was quite happy with it.

### Updating for the K2000

In early 1993, I got another opportunity to perform the piece, and I decided to take the opportunity to make a few changes in it. The most significant change was that I wanted to port it to a Kurzweil K2000. The K2000 has built-in effects and a keyboard, plus room for an internal drive. Instead of hauling around a sampler, a video monitor, a control keyboard, a mixer, and the two parts of the reverb unit, I could now bring the piece on the road with only the K2000. Not only does it fit very nicely into a DX7's road case, but it weighs less than the DX7!



So then I collected all of my original materials and resampled them on the K2000. **Not!!!**



Theoretically, I could have transferred the samples from the S-770 to the K2000 with MIDI Sample Dump Standard, but with 167 samples taking up 16 Megabytes, that process would have been exceptionally tedious. SMDI was not an option, as the S-770, even though it has a SCSI port, doesn't support it. Fortunately, the Kurzweil operating system allows the K2000 to read S-770 files on SCSI disks directly. In some ways, the K2000 does a better job reading S-770 disks than it does its own disks: from its own disks, it can only load in data as part of an entire bank, containing samples, keymaps, and programs; while from a Roland disk you can have it load in anything from a single sample to an entire volume containing thousands of samples.

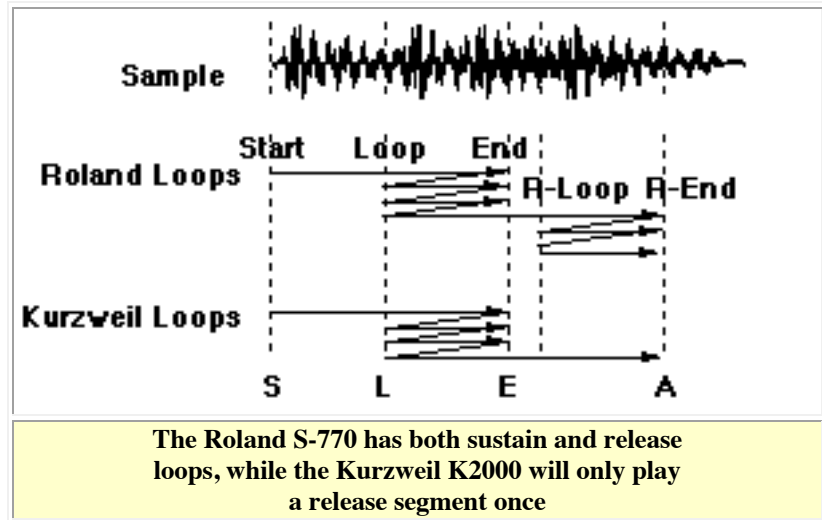
Since I had the whole piece on a single Syquest 44-Megabyte cartridge, I could get it into the K2000 in a few minutes. Now I could edit the sounds on the K2000, and save them on its internal disk. The samples translated nearly flawlessly, regardless of their length or sample rate. There were a couple that came across inexplicably truncated or with odd garbage in the middle, but a quick second pass (of only the samples in question Ñ something I couldn't have done if they were K2000 samples!) cleaned that up. The envelopes also survived the transfer well.

What didn't come across very well, however, were the tuning and key assignments of the samples. The Kurzweil tried hard, and I suppose if these were "normal" instruments with just a few multisamples mapped across the keyboard, it might have been more successful, but the complexity of these keymaps was too much. The S-770 has three different ways to tune a sample, and so does the K2000, but they're not quite the same. They're close enough that some of the samples did show up in the right place at the right pitch, but most of the samples had to be retuned and re-assigned. This actually served to give me a little more creative freedom, as I'll explain in a moment.

## **Loop-de-loops**

The sample loop points were all imported properly, which was a relief: resetting 300 loops was not a job I relished. However, there was a problem. Unlike the S-770, the K2000 doesn't have a release loop, only a sustain loop. It does, however, let you play the portion of the sample that comes after the loop, although only once. A sample will loop as long as

you hold down its key or the sustain pedal, and when you let it go, you'll hear the rest of the sample, provided you've given the envelope a long enough release time.



I discovered a strange thing happening: under certain circumstances, holding down the sustain pedal did not keep the sustain loop going — only holding the key did. I called Kurzweil to see if they could help me make sense out of this, but their engineers had not seen it before. Together, however, we eventually arrived at what was going on: some samples, when they were imported, set a switch in the K2000's sample editor called "Alt Sense" to "reverse". This caused the sustain pedal to ignore the loop, but still to sustain the envelope until the end of the sample, regardless of the length of the envelope.

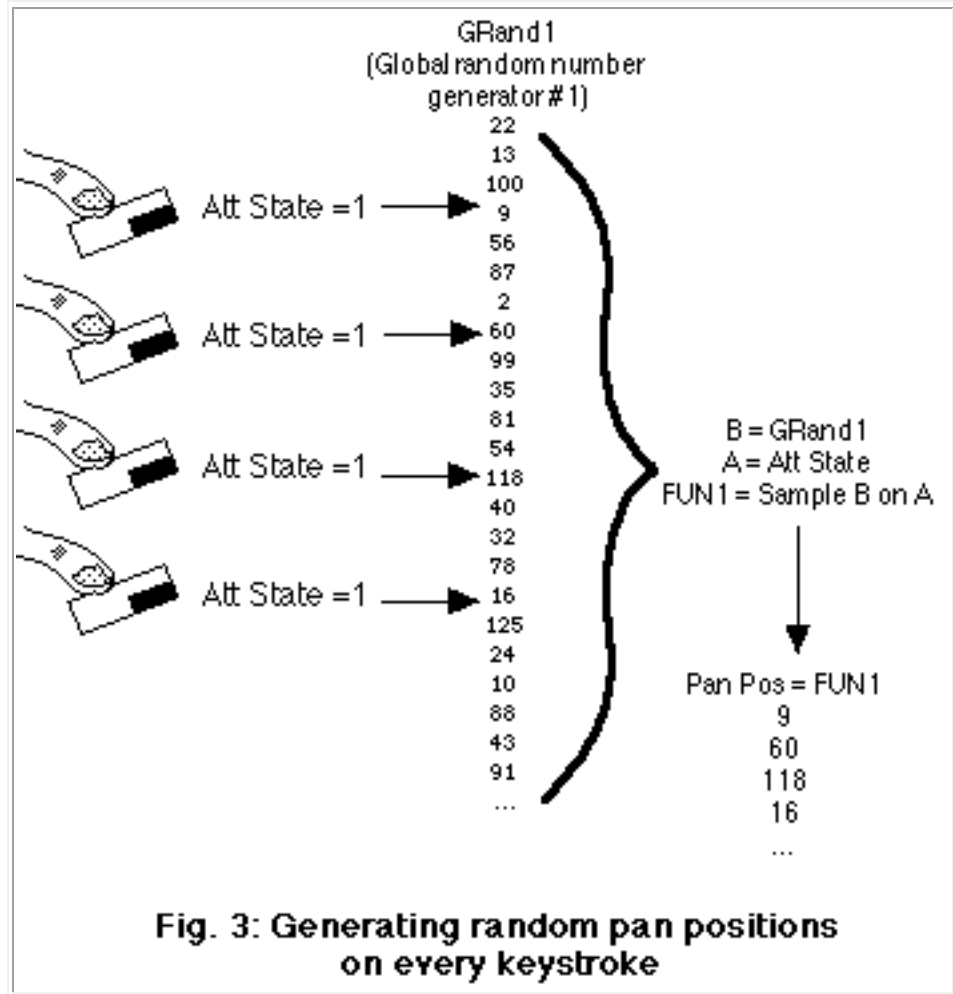
So what do you do when you find an interesting bug like this? Turn it into a feature! I set up a number of sounds to take advantage of this oddity. For example, I had a sample of Lennon saying, "They're jumping around naked all over the place." I programmed this so it looped on the word "naked" until I let go of the key. Then, if I was holding the sustain pedal, you heard "all over the place"; otherwise the sound stopped.

## It's Just the Motion

Another difference between the two instruments is how they handle panning. As I mentioned earlier, the S-770 can only pan the beginnings of sounds, which then play through at that particular spot in the stereo spectrum. The K2000, however, also can pan sounds continuously, so that they move around the stereo field ("swim") as they play. On the other hand, while there are lots of ways to control pan position in the K2000, making it random, which is a snap on the S-770, is not at all intuitive. Here's

how I figured out how to do this: The K2000 has a ridiculously large array of sources for controlling sound parameters in real time — LFOs, envelopes, timers, etc. — and also has several logical "Functions" (abbreviated "FUNs") for combining control sources. These let you do things like add channel pressure to modulation wheel for extra emphasis on a filter, or sync an LFO to MIDI clocks.

One source is a random number generator (actually there are four of them). Another is "Attack State", which goes to 1 (full value) when a note starts, and back to 0 as soon as the attack segment is finished. Among the Functions is "Sample B on A". This means that whenever A is positive, it will grab whatever value B happens to be at that time. By making A in this Function the Attack State, and B one of the random number generators, I could get a new random



**Fig. 3: Generating random pan positions on every keystroke**

Generating random pan positions on every key stroke

number every time I hit a key. This number could be assigned to the pan position, so that each keystroke put the resulting sound in a different spot.

I wanted several sounds, such as a loop from "Don't Wanna Be a Soldier", to "swim" by themselves. For these, I assigned a very slow (0.2 Hz) LFO to the panner. For some one-shot sounds which I wanted to move across the stereo field and then stay there, I assigned an simple envelope, triggered by the Attack State, to the panner.



Early on, I made the decision to take advantage of the K2000's ability to have up to 32 "layers" in a program. Each sample could have its own layer, and thus its own individual synth-style processing. (Normally, the K2000 only lets you assign 3 layers to a program, except that you can create special "drum programs", which have 32. You can only have drum programs on one MIDI channel, but since I was only using one MIDI channel anyway, I could make all five of my programs drum programs.) I spent a lot of happy, creative time designing pans, notch filters, vibratos, and filter sweeps for all of the sounds, both to clean them up and to bring them to life.

Another technique I stumbled on that made for even more interesting sounds was to use two or more keys for each sample. Originally, I had assigned any sample whose pitch was not going to change (which included most of the spoken samples and the rhythmic loops) to a single key. But since I was re-assigning all the samples anyway, I decided to try setting a layer's "pitch key tracking" parameter to 0, and assigning the sample to several keys, which would then all play at the same pitch. Hitting two keys with the same sample assigned to them produced a very clean, controllable delay: playing them at exactly the same time resulted in a comb filter; playing them slightly apart gave me a digital delay-like sound; and playing them like grace notes sounded like overdubbing or ADR.

For a couple of sounds, I set the pitch key tracking not to 0 but to 1 cent/note, which detuned adjacent keys by 1/100th of a semitone. Playing two keys together set up this way created a gorgeous sweeping chorus. Add that effect to a swimming pan and an LFO-driven bandpass filter, and you've got a sound with an amazing amount of motion.

In some cases, instead of putting adjacent keys on the same layer, I put them on different layers, detuned them slightly, and placed them in random pan positions using two different random number generators. This resulted in a stereo sound that not only appeared in a different place each time it played, but also its apparent width would change, depending on how far apart the two random numbers happened to be.

## **Internal Processing**

The K2000's internal effects processor is a sophisticated chip made by Digitech. It has a single audio input, and only can do one effect at a time (although some of them are combinations, like chorus+delay+reverb), so every sound being run through the processor will be effected the same way. While you can adjust the wet/dry mix of an entire program, there's no way to adjust the effects levels of different layers individually. A layer either goes through the effects processor (if it is assigned to the "A" outputs) or it doesn't (the "B" outputs).

However, if you have extra layers available, you can overcome this by creating two layers for each sound, and run one layer through the A outputs and the other through B. The effects mix can then be adjusted by changing the relative balance of the two layers. I used the K2000's data slider to do this: whenever I wanted real-time effects control on an individual sound, I set up the slider so that as I moved it higher, the level of the "A" layer went down and the level of the "B" layer went up. A bit complex, but it worked.

There was a more serious problem when it came time to switch effects. I gave each of the five programs its own effect, like a sweeping hall, multitap chorussed delays, normal room reverb, and "Into the Abyss" reverb. But then I found that changing programs, particularly if there was a lot of signal going through the effects processor, created serious glitches. Since most of the transitions between sections involved sounds from one section decaying slowly while the next section started, this would not do at all.

Before	After
501 Intro (effect = Warm Plate)	501 Intro (effect = Warm Plate)
502 Abstract Loops (effect = Flange Room)	502 Abstract Loops1 (effect = Warm Plate)
503 Murmur (effect = Sweeping Hall)	503 Abstract Loops2 (effect = Flange Room)
504 Rhythm Loops (effect = MultiChorus Delay)	504 Murmur1 (effect = Flange Room)
505 Finale (effect = Into the Abyss)	505 Murmur2 (effect = Sweeping Hall)
	506 Rhythm Loops1 (effect = Sweeping Hall)
	507 Rhythm Loops2 (effect = MultiChorus Delay)
	508 Finale1 (effect = MultiChorus Delay)
	509 Finale2 (effect = Into the Abyss)

**Modifying program changes for smoother effects transitions**

The solution was twofold. As it happened, each section had minimal effects at the beginning, and the effects level would generally increase as the section progressed. I created a duplicate of each of programs 2 through 5, and inserted the duplicate in the program list just below its original. Then I assigned the same effect to the duplicate as was assigned to the previous program. Now when I switched from the previous program into the copy, there would be no glitching. After the sound from the previous section had faded completely, I would switch into the original version of the program, which had the correct effect assigned. Since there was little signal going through the processor at that point, there was no audible discontinuity, and later in the section, when the effects level came up, the right effect would be there.

## Performing with the K2000

The concert in which I presented this newest version was a very complex affair. A fellow faculty member was doing a multimedia presentation with three slide projectors, live narration, a sequencer, several keyboards, and a Casio MIDI horn, all

going through a six-channel surround-sound system. Also on the program was a 10-piece student ensemble featuring three guitarists, two percussionists, and five synth players. Since my setup was by far the simplest, and we only had a couple of hours to set up and test everything, I mostly stayed out of the way. I ended up using only two of the K2000's outputs, going directly to a power amp and the main stereo speakers, and relied on the instrument's internal mixer to maintain the balance. It was the correct choice, because the sound was quite clean and the stereo imaging, which I had worked on so hard, was perfect.

No audio recording was being made of the concert, but I knew I could create a "virtual" recording of my performance by connecting a MIDI sequencer to the output of the K2000 and recording the key presses and controller movements. So I made arrangements to get a computer on the stage. In the pre-show rush, however, even this seemed like too much trouble, so I hit upon the revolutionary idea of using the K2000's internal sequencer. The sequencer was very rudimentary: it was essentially one track, with overdubbing but no editing. The memory was battery-backed RAM (it shares this with the program memory, separate from the sample RAM), and there was about 50K of it available. Theoretically, I could record the performance, shut off the K2000, and then play it back later at home. So I tried it.

The piece, 11 minutes long, used up 89% of the available memory. I saved it on a floppy disk before going off stage just in case, and hoped for the best. When I got home the next day, I re-loaded the samples, started the sequencer, and heard an exact replication of the performance I had played the night before, mistakes and all. I hooked to K2000 up to my Mac, and played the sequence into a MIDI sequencing program, Passport Designs' Master Tracks Pro 5. Now I could edit and clean it up to my heart's content, and make a CD-quality recording of my concert performance in my own studio.

In the creation of "I Dig A Pygmy," I took advantage of so many of the K2000's capabilities that now it could be performed anywhere a K2000 can go — and I don't even have to show up. But I would never do it that way: the whole point of the piece is that it is designed for a live musician to play in concert, improvising in response to the audience and the space. In that way, it combines aesthetics of the 50s, the 60s, and the 90s into a coherent piece of performance art. Not only that, but boy, is it fun! •

---

*postscript:* On May 5, 2001, I premiered a new version of the piece that used four channels of audio, for a full-surround experience. In this version, the second pair of channels is generated by the K2000's "B" outputs and is sent through an Alesis Wedge reverb, which receives MIDI program changes from the K2000. The piece was

performed at the marathon concert of the Brandeis Electro-Acoustic Music Studio ("BEAMS") at Brandeis University in Waltham, Mass., as part of the Boston Cyberarts Festival. The four-channel version was performed again in April 2004 at a Tufts Composers concert at Tufts University in Medford, Mass.

---