

# *Bob Ostertag Plays the Aalto*

## Tracks:

Machine State 1	3:27
Machine State 2	2:53
Machine State 3	5:33
Machine State 4	4:32
Machine State 5	5:43
Machine State 6	4:51
Machine State 7	1:51
Machine State 8	6:04
Machine State 9	2:01
Machine State 10	2:20
Machine State 11	1:23



Cover art by  
**Haiyang Wang**

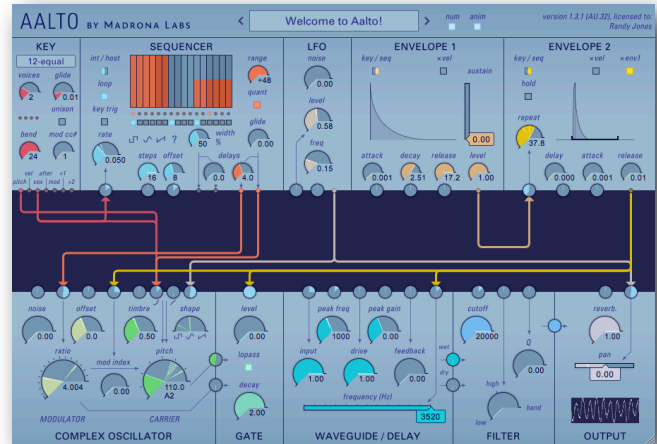
All this music was made with *Aalto*, a software synthesizer created by Randy Jones and his Madrona Labs, which was in turn inspired by the modular analog synthesizers of the 1970s.

Those early synthesizers were my first electronic instruments, beginning with the ARP 2600, then Don Buchla's *Buchla 200*, and then Serge Tcherepnin's *Serge*, which I played for several years. I used these on my first recordings: *The Surgeon General* (recorded 1977, released 2012); Anthony Braxton's *Creative Orchestra Koln* (recorded 1978, released 1995); Eugene Chadbourne's *The English Channel* (1978); John Zorn's *Pool* (1979); Fall Mountain's *Early Fall* (1979); and *Voice of America* with Fred Frith and Phil Minton (1982).

These synthesizers consisted of collections of discrete electronic modules which could be interconnected with cables, like a telephone switchboard. Each module either generated a pattern or modified the patterns generated by other modules. Moog, ARP and a few others put keyboards on their instruments and encouraged you to see them as a sort of next-generation pipe organ. But others suggested a different way of making music. Instead of conceiving of music as a series of events distributed across a timeline (a musical staff on paper back then, and the ubiquitous timelines that run left to right across computer screens today), you could configure the machine into a state overlapping patterns that *generated* interesting music, into a particular state and let it run.

This involved coming to terms with a very different notion of “playing” an instrument, as a given machine state was only interesting if allowed to run long enough for the listener to discern the shapes of the patterns. If you moved too many knobs too far too often, you wrecked it. You might move a knob a tiny bit every now and then, or plug in a cable here and there. But the most interesting music happened when you didn’t do too much.

But you did have to turn some knobs sometimes, because even the most well-conceived synthesizers could not be configured into a state that was so interesting you would want to listen to it indefinitely. So you had to touch the machines, but at the right time and not too much. And generally when you touched them it was to make a single slight change to one parameter. If you experimented with a given state long enough, you would usually find that there was one or two parameters that, when changed just once in a very small increment, would trigger cascades of changes in the patterns being generated that were deeply interesting.



Aalto screenshot

Hearing the patterns would challenge your ear in a most pleasing way. The practice of “playing an instrument” shifted in a way that made simple listening a much larger component and while physical activity played a smaller role. Playing became a sort of meditational practice of keeping the mind focused on the smallest details of complex patterns of sound.



Me with my Serge, 1979

This was such a different notion of what constituted “playing an instrument” that many people never got comfortable with it. They wanted to *do* stuff. There were decades of experimenting with “alternate controllers,” non-keyboard input devices that would be more idiomatically appropriate to this new music. But the real issue was not the elegance of the input device. The issue was that if you tried to control too much, the music lost its substance.

It took a long time working with a given synthesizer to discover how to create machine states which were interesting at the level I am describing. Of those which I played extensively, the Arp 2600 just couldn’t do it. The designers of that instrument were not thinking on that level. Neither could the Moog (though the Moog’s oscillators and filters sounded so good that the Moog quickly became the favorite of musicians who wanted to use a synthesizer to play conventional music in a

conventional way). The best instruments for the kind of music I am describing were the *Buchla* and *Serge*. And I always considered the music I made with those instruments to be collaborations with Don Buchla and Serge Tcherepnin.

(Note, however, that when playing improvised music with John Zorn and Eugene Chadbourne in the 1970s, I tried to play the Serge in a very different way. John and Eugene's music centered on sudden, extreme discontinuities, and to keep up I tried a completely different approach in which my hands moved from knob to knob and patch cord to patch cord as fast as I possibly could. You can hear that on those early records.)



Don Buchla with his instruments, 1970s

The old synthesizers had serious limitations. They were expensive. It was impossible to recall a particular configuration of the machine. And the knobs used to set the parameters were not sufficiently exact to set precise ratios between different parameters. These instruments were soon overwhelmed by digital synthesizers with keyboards controllable by MIDI sequencers, which were less expensive, allowed creating and recalling presets, and had parameters which could be precisely set. But these instruments also brought back the conventional world of the keyboard (and later the drum pad) and the timeline with a vengeance.



Serge Tcherepnin with a *Serge*, 1970s

When personal computers achieved the computing speeds required for audio synthesis, many assumed that the experience of working with an old modular analog synthesizer would be recreated in a more flexible and pleasing way on computers. But this never happened. The reasons for this are too long to elaborate here.

In recent years there has been a renaissance of modular analog synthesizers, and there is now a vibrant subculture of young analog synth aficionados who design, build, and trade hardware modules. These instruments still have all the limitations they had back in the day, but these young musicians prefer those obstacles to the set of problems presented by computer and software. They perceive the sound of the

analog synthesizers to be superior, and prefer working with knobs and hardware over trackpads and mice.

But the latest software synthesizers sound every bit as good as the old analog stuff, or even better to my ear. And I can remember when those of us who played those old instruments couldn't wait to work with something other than knobs. What is interesting about knobs?

Yet when I work with computers, I miss two things about old synthesizers. First, when you made music on an old synthesizer, you were *using your ears* from the moment you sat down until the moment you got up. Always listening. Intently. That was the fundamental thing you did: listened. Go into any computer music studio today and it sounds like a library. Silence! Everyone is busy writing code, which will *eventually* make sound. But for the most part you don't want anything to distract yourself from the important job of writing code. And second, an old synthesizer is a fixed world of possibility. It has, say, two oscillators. So you are going to have to figure out how to make the music you want with two oscillators. With a computer, you have as many oscillators as your computer can run, which at today's computer speeds is lots and lots. So, how many do you want? Why that number? Is your computer fast enough to do that? Maybe you should upgrade. Upgrading will take a long time. During all that time, the room will be silent, because you have to concentrate in order to avoid mistakes that will make the process of upgrading even longer. Maybe if you write some particularly clever code you can get the system you have now, without upgrading, to run that many oscillators. But writing that code will take longer than anticipated. Another long silence. The silences add up. In terms of *making music*, it turns out the question of how to make music with two oscillators was more interesting than how to program a computer to do whatever your great idea was.

Thus encountering Randy Jones' *Aalto* software synthesizer was a joy. No, it doesn't have knobs, but: (1) it gives me a fixed world of tools, that are (2) sufficiently complex to facilitate the approach to music I am talking about; (3) it makes interesting noise from the moment I sit down until the moment I get up, and (4) cannot be upgraded or expanded. Everything I liked about the old synthesizers is here. And I can specify precise ratios in an extremely intuitive way, and easily recall machine states. Working with *Aalto* I realize the inability of the old instruments to recall machine states was an even bigger obstacle than I had thought, because you simply cannot begin to master the complexities of a given state in one sitting, or even two or three.

I can now at long last work in the way the old synthesizers promised but never quite delivered. I consider this release as a collaboration with Randy Jones.



Arp 2600