

Notes on SYNTAL (V.00.2) for OSX

Chapter 3: Running Stereo

Wayne Slawson

May 21, 2004

“Stereo” and “Two-Voiced” Sound

WARNING: This has not been tested lately. I’ll send on a new version when I do get to test it.

Stereo means, in ordinary audiophile parlance, a system by which a spatial image of a sound source is reproduced for a listener in a “stereo” audio system in which (at least) two channels of recorded sound are played back through (at least) two loudspeakers. It’s what you get when you play a CD through your stereo hi-fi system.

This is to be distinguished—not entirely, but nevertheless distinguished—from a piece with two simultaneous musical voices that combine together to make counterpoint. It’s possible to compose two voices, record one on the Left channel of a cassette tape or writable CDR, the other on the Right channel and there you have two voices in stereo. But it’s also possible to mix together the two voices on, say, the left channel and then you have two-voiced *mono*. Another way of handling two voices is to set things up so that sound of, say, the first voice sounds like it’s coming from the middle of the sound space and the second voice sounds like it’s *moving* back and forth from the left to the right speakers.

Let’s set aside, for the moment, the problems of making mixed two-voice sound or sound that apparently is moving. Let’s instead make two voices, one of which is in the left channel and the other in the right channel.

First, make two separate files. You need to compose the two voices as separate SYNTAL files. Then you can compile, synthesize, and play back each of the two files separately to make sure that each one sounds the way you want it to sound. You would use **csp** as usual.

Now, mix them together. There are some special programs that help you to mix together two (or more—later for that) files that are related by a special naming convention. Suppose you want to call the two-voice result “Prj”. Then the first of the two SYNTAL source files should be named “1Prj.c”; the second, “2Prj.c”.

Let’s assume that you’ve made these two files (with exactly those two names) and you want to mix them together. The first thing you need to do is to make a Csound “.sco” file that specifies *how* the mixing is to take place.

Here’s an example of such a “.sco” file:

```
; This is a Csound .sco file that uses "mixrev2.orc"
; It's for mixing together two files that are 7 seconds in duration.
i 11    0  .01          0.
i 11    +  6.98         0.
i 11    +  .01          0.
i 12    0  .01          1.
i 12    +  6.98         1.
i 12    +  .01          1.
i 50    0  7.00   1 2 0 0
e
```

In .sco files, anything on a line after a “;” is a comment; the first line above is therefore such a comment. And as always it’s a good idea to put in a comment just to remind you of what this file you’re working is for. Each line (after the comment) starts with an “i” until you get to the last line, which consists entirely of an “e”. And to take the simplest issue first, that “e” line stands for “this is the end”. The only thing to remember is that the “e” line should have a `jreturnj` after the “e”. (Note the difference, by the way, from the end of a SYNTAL file, which ends with “END”) The “i” statements have what are called parameters, or “p-fields”, with names labeled in order: p1, p2 p3, p4, etc. In this case each “i” statement has only four p-fields, the first three of which are “instrument number” (p1), “start time” (p2), “duration” (p3), and “panning parameter”. At this point I might as well observe that the first three p-fields in “i” statements are *always* instrument number, start time, and duration, in that order. From p4 on, the parameter is peculiar to the Csound orchestra. In this case the orchestra is called ‘mixskp2to2.orc’, and ‘mixskp2to2.orc’ expects that p4 will represent the

location in the sound field that you want the two strands of “Prj” to appear to be coming from. In p4, 0 means left and 1 means right. Now let’s refer back to the “Prj.sco” file shown above. By convention (in this particular orchestra), instrument 11, referred to by all “i 11” statements, controls the first voice, and instrument 12, referred to by all “i 12” statements controls the second voice. In ‘Prj.sco’ voice 1, controlled by the “i ii” statements starts in the left channel and stays there; voice 2, controlled by the “i 12” statements starts in the right channel and stays there. That is to say, p4 in all the “i 11” statements are 0. and p4 in all the “i 12” statements are 1. Pretty simple, you say. OK, as a thought exercise, how would you get voice one to appear to come from the middle of the stereo sound field, and voice 2 to appear to move from the right speaker to the left speaker over the 7 seconds of Prj?

Another little point: You may not know exactly how long, in seconds, the ‘Prj’ actually is. You can figure it out by counting up all the durations in the SYNTAL statements, but there is an easier way. Run **csp** on each file and look at the printout. In a couple of places, it will tell you the durations—at the end of the SYNTAL processing, before Csound starts its work, or at the end of the Csound processing. It’s always better to let the computer do the work for you. And you can check to make sure both voices are the same duration. (If they aren’t, you may need a rest at the beginning or end of one of the voices.)

OK. Now you’ve got two SYNTAL files—‘1Prj.c’ and ‘2Prj.c’, and one Csound score file—‘Prj.sco’. You’ve run **csp** on ‘1Prj.c’ and ‘2Prj.c’ so there must be sound files in your directory called ‘1Prj.aiff’ and ‘2Prj.aiff’. How do you mix them together? Here’s how:

roger25% cmx Prj

You should get a little bit of calculation shown on the screen, and then you can hear the result by means of Quicktime.

A shortcut: Suppose you’ve prepared the two SYNTAL source files with the names ‘1Quick.c’ and ‘2Quick.c’ and a .sco file with the name ‘Quick.sco’. You can run everything all with one command, without running **csp**. Here’s how you do that:

roger25% do2 Quick

That’s all for now. Happy mixing!