RTSPA1: a program and some technical & practical remarks for real time sound spatialization

Prof. Oscar Pablo Di Liscia
Carrera de Composición con Medios Electroacústicos
Universidad Nacional de Quilmes (Argentina)
odiliscia@unq.edu.ar

Abstract
Real time spatialization of sound involve not only DSP processes, but also the design of useful user interfaces to produce meaningful movement paths effectively connected with musical ideas. The program RTSPA1 addresses this problem through the control -via MIDI data- of the software and hardware Session8 (1). In addition to this, the program could be used also to control other real time DSP programs and devices, among of them the Csound program. Future developments are addressed to convert it on a general-purpose graphic interface to control spatial location of sound and music.

1 GENERAL COMMENTS
This program - among others- was written as a part of the research projects: Design and application of digital filters for artificial reverberation emulation, (UNQ, 1996-97) and Sound and Music spatialization (UNQ, 1997-98). It may be freely copied, used and distributed for non-commercial purposes.¹

RTSPA1 allows real time control, via MIDI data, of the Software and Hardware Session8 (1), to provide quadraphonic sound spatialization. It should run on any Windoze OS system, which can support the latter software and hardware.

¹ The most updated version may be obtained at: www.fortunecity.com/timpan/queen/353/home.htm
It is also possible to use it to control the Csound program (Barry Vercoe, MIT), or any program or device which may read MIDI control values and assign them to gain control of several audio channels.

The first version was specially written to be used on the musical drama "el hecho", by the argentine composer Oscar Edelstein (Premiered at Buenos Aires, June, 1998). RTSAP1 simulated spatial movement of the voice of one of the most important characters on the piece, "voz de Paz", performed by Mrs. Paula Ortega.

Each screen object of the mixer of Session8 is assigned to a specific MIDI controller number (the file midimap.txt of software Session specify the associations). As one example: the channel # 1 gain fader is associated with # 0 controller number, thus, sending MIDI commands of control # 0 will move the fader changing the amplitude of the signal being reproduced. Unfortunately the resolution of the values are only 7 bits, so the movements must be slow enough to avoid unwanted clicks due to amplitude discontinuities.

To improve the spatial effect, a copy of the direct sound (without gain scaling) was sent to a hardware reverberator, and its output ("wet") was mixed together with the four channels ("dry", or direct with gain scaling). Therefore, as distance increases, direct amplitude decreases while global reverberation amplitude ("wet") keeps the same, providing a powerful cue for judgment of distance between listener and virtual source.

2 SPECIFIC FEATURES OF RTSAP1

RTSPA1 assumes the listener is located at the origin of a Cartesian plane, and surrounded by four equidistant loudspeakers, having angles of 45, 135, 225 and 315 degrees, respectively.

The user may change the location of the virtual source using polar coordinates (angle & distance) on several ways, including a set of "automated movement presets" which allows both control and effectiveness under real time performance.

Figure 1 shows the screen of RTSAP1, with numbers for each of its objects.

Follows a brief explanation of each one.

(1) If this box is unchecked, the program works, but does not send any MIDI command. Useful to setup all the stuff before the
performance.

(2) With this scrollbar you choose by which MIDI channel is the command to be sent to Session8. (It must agree with Session8's MIDI Setup)

(3) Here you select the MIDI port which will receive the commands. (It must agree with Session8's MIDI Setup)

(4) Here you associate the MIDI # control for the four faders you will control, plus two more controllers which can be used to move the Master faders, or any other object. The distribution is:

```
front left  front right  masterch1
rear left   rear right  masterch2
```

The default # control mapping is:

```
1  0  8
2  3  9
```

(5) This scrollbar modifies the angle between the virtual sound source and the listener. Angle is indicated in degrees (counterclockwise). Its position is auto updated for the case of any automated movement (see (8) and (9))

(6) This scrollbar modifies the distance between the virtual sound source and the listener. It multiplies the normalized gain value for each channel by 128 - (V+1) where V is the current distance value. The number resulting (truncated to 7 bits unsigned int) is the second data byte of the correspondent controller. There is no strict correspondence between this value and distance in meters.

(7) This scrollbar modifies both controller numbers indicated on (4), left row. It may be set to control any other screen object of Session software.

(8) This panel provides several presets to perform automated controlled movement. Because RTSPA1 was meant to be used on live performance and since the mouse is not a versatile device for this task, the challenge was to find some compromise between "controlled" and "automated" movement.

The panel has 12 buttons, and each one triggers a different kind of process.

The four one closer to center (labeled: 1- 2, 2- 3, 3- 4 and 4- 1)
triggers panoramic movement of the source between the designated channels. Clicking the same button you stop the process. It is possible to change the distance gradually (using scrollbar (6)) while the process is running, to simulate zigzag movement.

The eight extern buttons (labeled: 1 - FR - 2 - LE - 3 - RE - 4 - RI) provide fix location of the source, performed "by jump" (on a similar fashion of the "flying sounds" of K. Stockhausen's Kontakte). The process is as follows: when any of these button is clicked, a "fade out" process is started, when 0 amplitude is reached (then, no direct output), the angle is suddenly changed according with the position required, and a "fade in" process is started. Thus, the sound source seems to disappear from its initial location, and popup at a different one.

(9) This button triggers automated circular movement. Clicking it again the process is stopped. It is possible to change the distance gradually (using scrollbar (see (6)) while the process is running, to simulate a spiral shape path.

(10) This scrollbar sets the size of increment of the values on every automated process, thus modifying the speed of them. It is possible to modify this value while some process is running to change gradually its speed.

(11) This button modify the direction of the circular movement (clockwise- counterclockwise)

(12) This button sends either a "key on" (C5, KV=127) MIDI order or (if the latter was already sent) a "key off" (C5, KV=0) MIDI order. It is only useful to trigger a note if you are using RTSPA1 to control Csound (see below).

3 USE OF MULTIPLE INSTANCES OF RTSPA1

It is possible to run more than one instance of RTSPA1 (most usually one, to control the remaining four channels of Session8). The output port must be the same for both instances (since Session8 does not allow more that one port receiving control data). This sometimes creates problems, because some MIDI drivers do not allow a multi-open feature (one might ask: what are these drivers doing on a multitasking OS...???). Those problems are not due to RTSPA1, and can be solved using a proper MIDI driver.

It is a good idea to spatialize two copies of the same input signal on different ways. Doing so (say, for instance, producing complementary spectra, using the Session8 Lowpass / Highpass filters) the effect is outstanding, because the sound seems to be
“splitted” when the two audio streams do no meet the same spatial location and “restored” when they do.

4 USING RTSPA1 WITH CSOUND

Csound (Barry Vercoe, MIT) is a widely known digital signal compiler with capacities of real time DSP and MIDI processing. Of course, its capacities depend on the efficiency and speed of the hardware and OS used, but at present I have tested it on a 133 MHz PC getting a very acceptable performance. I assume for the following section the knowledge of the reader of the basic features of Csound.

Here follows a simple model of orchestra and score for Csound, to be used together with RTSPA1:

First the orchestra:

```plaintext
; /******************************************************************************/
; rt1.orc
sr =44100          ;header
kr =441
ksmps =100
chnls =4

instr 1
ifac =1./127.       ;normalization factor

kch1 midictrl 0,0   ;channel 1 reads controller #0
ach1 interp kch1    ;interpolation to smooth values
kch2 midictrl 1,0   ;channel 2 reads controller #1
ach2 interp kch2    ;ditto...
kch3 midictrl 2,0   ;channel 3 reads controller #2
ach3 interp kch3    ;ditto...
kch4 midictrl 3,0   ;channel 4 reads controller #3
ach4 interp kch4    ;ditto...

asig in            ;read input signal
asig=asig*ifac     ;scale it
arev reverb2 asig*.1,2,.9 ;send a copy to reverb
arev tone arev,5000 ;lowpass reverb

outq (asig*ach1)+arev, ;output direct + reverb
     (asig*ach2)+arev,
     (asig*ach3)+arev,
```

The score is quite simple:

Of course, more powerful cues for spatialization (early echoes, Doppler shift, H.R.T.F...etc.) could be used. It seems, however, that for real time performance, the two I have used are the most effective, and takes no big effort to the computer´s CPU.

To start execution, the command line should look like this: csound -o dac -d rt1.orc rt1.sco -M sbmidi

Csound will wait for MIDI events to process. These will be read on the MIDI port specified when required, and you suppose to send it using RTSPA1. However, to set active instrument 1 you should send a key on MIDI order using the button designed for this on RTSPA1. The signal to be spatialized must be input to the required audio device, which should be full dual.

The ":-o dac" (or ":-o devaudio") flag tells to Csound that its output should be sent to the audio hardware. However, audio drivers for Windoze OS systems are usually prone to regard all audio devices as combinations of stereo devices, so quad output is not possible most of the cases.

Solutions for the latter problem may be: 1)To write a fair audio driver, or 2)To modify Csound to output audio by more than one device.

5 CONCLUSIONS

The software is simple, robust and perceptually effective, and it constitutes a useful tool to experiment with spatial location of sound. It will be updated to support many other systems and/or devices.
It was also proved, through its use on several performances, that effectiveness on spatial location of sound (especially on real time situation) depends not only on reliability of the cues and audio processes used on simulation. It also depends on the capacity of production and control of movement paths strongly related with our movement experience, and to organize them in differentiate layers with musical significance.

NOTES

(1) Session8 Software and Hardware are both registered trademarks of Digidesign.

BIBLIOGRAPHY


Chowning, J., Grey, J. Moorer, A. : Computer simulation of music instrument tones in reverberant environments, Report N° STAN-M-1, CCRMA, Stanford University,


Figure 1: the basic screen of RTSPA1