

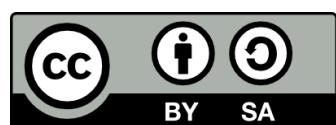
Sound Treatments in *RTcmix*~

A MaxMSP Patching Odyssey

ENT3320 — *Interactive VR, Music, Sound, and Memory*

**New York College of Technology, CUNY City Tech
Spring 2020**

Professor Louis Goldford
LGoldford@citytech.cuny.edu



*Creative Commons
Attribution-ShareAlike 4.0 International*

CONTENTS

Sound Treatments in <i>RTcmix~</i>	1
CONTENTS	2
1. What is a “Treatment”?	3
2. Download test.sounds Folder	4
3. Playback from Disk	4
4. Outskip / Inskip Pfields	5
5. Outsk/Insk With a For Loop	6
6. Tape Transposition	7
7. Transposition With Correct Duration	8
8. Transposed Grains	9
9. Low Pass Filter (LPF)	10
10. Dynamic LPF "Sweeping"	11
11. Basic Schroeder Reverb	12
12. A Basic Signal Chain:	13
13. Audio Input From Max	14
13.1 rtininput("AUDIO")	15
14. Amplitude Modulation	16
14.1 Makeconnection	17
15. Ping Pong Delay	18
15.1 PANECHO()	19

1. What is a “Treatment”?

Historically, the term “**treatment**” was used within electrical engineering to mean a modification imposed on an analog signal — a continuous, electrical circuit whose current or charge we can modulate with crafty engineering. Amplification and filtering are among these “basic operations.”

Until the advent of digital computers, such analog “treatments” were the only way to manipulate a signal. This mechanical treatment of sound was made possible through classic radio-telecommunications signal processing: time-domain audio filters and EQ (“equalization” filters in parallel), compression, and reverberation (filters in series), among others.

Significantly, [reel-to-reel tape machines](#) were commonly found in early radio and TV studios. Tape, in addition to its main use as a recording medium, extended treatments into the realm of time and pitch manipulation, and the re-ordering or shuffling of sonic events. Think about this: with tape, you not only can store an electro-magnetic signal but also you can slow the tape down, or run it backwards. When you change a reel’s speed, it also raises or lowers the pitch of your captured material. When you run it backwards, you also reverse time’s flow and with it, a sound’s entire ADSR envelope — first, its release runs backwards, followed by its sustain, decay, and finally its attack. With the ability to chop up little pieces of tape (called “splicing”) and reconnect them, small components of sound can be reordered, making elaborate collages possible. Such was the creative research of Pierre Schaeffer and his followers.

Today we continue to use digital emulation of these classic analog techniques as basic and often “pure” sonic treatments. We tend to create interest and variety by personalizing each method of manipulation, or by combining them, one after another, into a signal chain.

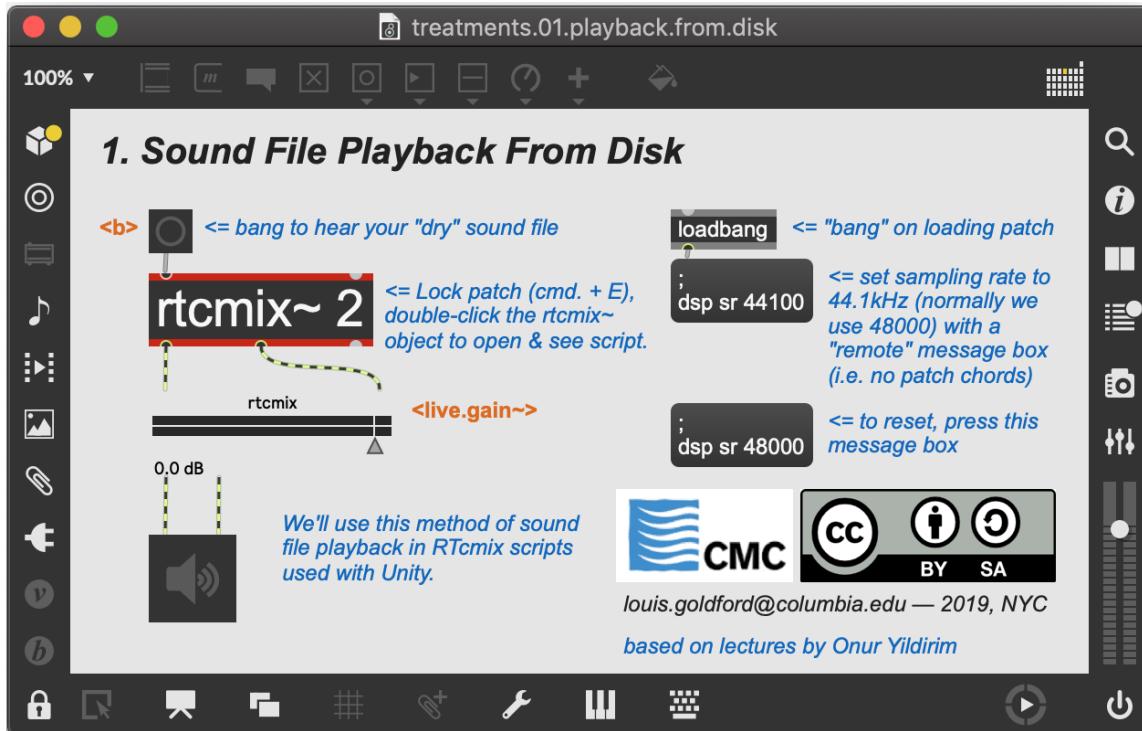
Perhaps it’s not enough simply to call a treatment a “sound effect.” People tend to think of “effects” (a.k.a. “FX”) as a layer of processing that somehow “enhances” the raw/dry sound source, or doesn’t alter its character much. While a treatment can certainly provide this, we cannot limit ourselves to this naive definition. Each of these basic operations, separate or combined in some way, makes possible a limitless range of expressive signal processing that could entirely change the sonic character of our original sound sources! We could invent wholly other sound worlds from a single, quite different, source. The composer Tristan Murail wrote about this problem of “merely effects,” urging composers to think of treatments as their own musical source material, as the foundation of an entirely new, expressive musical object in its own right. For example, if you use a distortion pedal on your guitar, listen for and try to shape the quality of the distortion itself; do not “insist” on hearing the guitar through the distortion. If we insist on effects alone, we are sure never to realize the full potential of sound treatments.

We’ll examine how some of these treatments work and how they sound using RTcmix~.

2. Download test.sounds Folder

Be sure to click the link, “[Download a library of audio samples for testing](#)” on our Website. You’ll be using the sound samples here to test the treatments we build.

3. Playback from Disk



```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtininput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the STEREO Instrument:
9 // STEREO(outsk, insk, dur, AMP, P4-N: input/output channel pan assigns(0-1 stereo; 0.5 is middle))
10 // source: http://rtcmix.org/reference/instruments/STEREO.php
11
12 STEREO(outsk=0.0, insk=0.0, dur=3.00, amp=1.0, 1.0, 0.0)
13
14 // Here we must use 2 pan arguments for each channel (1.0, 0.0).
15 // LEFT channel (flute/drums) is panned hard left (1.0),
16 // RIGHT channel (piano) panned hard right (0.0).
17 // Total number of pfields is therefore 6.
```

Cursor Line: 5 Insertion Point Line: 1

4. Outskip / Inskip Pfields

script_0

```
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtininput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif
7
8 // Syntax for the STEREO Instrument:
9 // STEREO(outsk, insk, dur, AMP, P4-N: input/output channel pan assigns(0-1 stereo; 0.5 is middle))
10 // source: http://rtcmix.org/reference/instruments/STEREO.php
11
12 // LEFT channel in left output (1), muting right (-1)
13 // starting 1.55 seconds into sound file (insk=1.55 skipping flute lick)
14 // starting playback after 1.55 seconds after bang msg. (outsk=1.55)
15
16 STEREO(outsk=1.55, insk=1.55, dur=1.45, amp=1.0, 1, -1)
17
18 // left channel muted (-1), RIGHT channel (piano) in right output (0)
19 // entire sound file of 3 seconds (dur=3.0)
20
21 STEREO(outsk=0, insk=0, dur=3.0, amp=1.0, -1, 0)
22
```

Cursor Line: 17 Insertion Point Line: 1

5. Outsk/Insk With a For Loop

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the STEREO Instrument:
9 // STEREO(outsk, insk, dur, AMP, P4-N: input/output channel pan assigns(0-1 stereo; 0.5 is middle))
10 // source: http://rtcmix.org/reference/instruments/STEREO.php
11
12 // Random pan + fade out in 12 easy steps
13 // Each iteration of the FOR LOOP plays one small "grain" of sound...
14 // Incrementing forward through outsk (each "grain" starts playback successively)
15 // Decrementing insk (counting backwards through location of audio file)
16
17 amp = 1.0
18 outsk = 0.0
19 insk = 3.0
20
21 for (i = 0; i < 12; i = i + 1) {
22     STEREO(outsk, insk, 0.5, amp, random())
23     outsk = outsk + 0.25
24     insk = insk - 0.25
25     amp = amp - 0.1
26 }
```

Cursor Line: 12 Insertion Point Line: 1

6. Tape Transposition

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtininput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the TRANS Instrument:
9 // TRANS(outsk, insk, dur, AMP, TRANSP[, inputchan, PAN])
10 // source: http://rtcmix.org/reference/instruments/TRANS.php
11 // TRANS only does one channel at a time
12 // uses oct.pc notation, so 0.03 = up 3 half-steps; -0.05 = down 5 half-steps
13 // half-steps are the successive black-and-white keys on a keyboard...
14
15 TRANS(outsk=0.0, insk=0.0, dur=6.0, amp=1.0, transp=-0.05, inputchan=0, pan=1.0)
16 TRANS(outsk=0.0, insk=0.0, dur=6.0, amp=1.0, transp=0.0, inputchan=0, pan=0.5)
17 TRANS(outsk=0.0, insk=0.0, dur=6.0, amp=1.0, transp=0.03, inputchan=0, pan=0.0)
18
19 // Scheduled for later...
20
21 TRANS(outsk=3.0, insk=0.0, dur=6.0, amp=1.0, transp=0.07, inputchan=0, pan=0.5)
22 TRANS(outsk=4.0, insk=0.0, dur=6.0, amp=1.0, transp=0.10, inputchan=0, pan=0.5)
23 TRANS(outsk=5.0, insk=0.0, dur=6.0, amp=1.0, transp=1.02, inputchan=0, pan=0.5)
24 TRANS(outsk=5.5, insk=0.0, dur=6.0, amp=1.0, transp=1.04, inputchan=0, pan=0.5)
25 TRANS(outsk=6.0, insk=0.0, dur=6.0, amp=1.0, transp=1.06, inputchan=0, pan=0.5)
26 TRANS(outsk=6.25, insk=0.0, dur=6.0, amp=1.0, transp=1.08, inputchan=0, pan=0.5)
```

Cursor Line: 18 Insertion Point Line: 1

7. Transposition With Correct Duration

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the TRANS Instrument:
9 // TRANS(outsk, insk, dur, AMP, TRANSP[, inputchan, PAN])
10 // source: http://rtcmix.org/reference/instruments/TRANS.php
11 // TRANS only does one channel at a time
12 // uses oct.pc notation, so 0.03 = up 3 half-steps; -0.05 = down 5 half-steps
13 // half-steps are the successive black-and-white keys on a keyboard...
14
15 // TRANS only speeds up or slows down the soundfile...
16 // Use the translen() scorefile command to get the correct duration...
17 // i.e. to target a specific pitch transposition!
18
19 basedur = 3.0
20 transposition = -0.12 // 12 half-steps; equal to -1.00 in oct.pc notation (one octave!)
21
22 dur = translen(basedur, transposition)
23
24 TRANS(outsk=0.0, insk=0.0, dur, amp=1.0, transposition, inputchan=0, 0.5)
25 TRANS(outsk=0.0, insk=0.0, 3.0, amp=1.0, transp=0.0, inputchan=0, 0.5)
```

Cursor Line: 14 Insertion Point Line: 1

8. Transposed Grains

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the TRANS Instrument:
9 // TRANS(outsk, insk, dur, AMP, TRANSP[, inputchan, PAN])
10 // source: http://rtcmix.org/reference/instruments/TRANS.php
11 // TRANS only does one channel at a time
12 // uses oct.pc notation, so 0.03 = up 3 half-steps; -0.05 = down 5 half-steps
13 // half-steps are the successive black-and-white keys on a keyboard...
14
15 amp = 1.0
16 ampenv = maketable("line", 1000, 0,0, 5,1, 50,0)
17
18 basedur = 0.32
19 outsk = 0.0
20
21 for (i = 0; i < 300; i = i + 1) {
22     insk = irand(0.0, 2.8) // a range within the file targetting the piano sound
23     transp = irand(-0.11, 0.11) // +/- a Major 7th
24     dur = translens(basedur, transp)
25
26 // Random stereo panning using only the RIGHT channel (piano)
27     TRANS(outsk, insk, dur, amp*ampenv, transp, 1, random())
28     outsk = outsk + irand(0.08, 0.16)
29 }
30
```

Cursor Line: 16 Insertion Point Line: 1

9. Low Pass Filter (LPF)

```
script_0
1 // To load a sound file from disk, FULL_PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the MOOGVCF Instrument:
9 // MOOGVCF(outsk, insk, dur, AMP, inputchan, PAN, BYPASS, FILTFREQTABLE, FILTRESONTABLE)
10 // source: http://rtcmix.org/reference/instruments/MOOGVCF.php
11
12 // "Moog" lowpass filter
13
14 // BYPASS -> bypass this filter (0: bypass off, 1: bypass on)
15 // FILTFREQTABLE -> filter cutoff frequency (Hz)
16 // FILTRESONTABLE -> filter resonance (0-1, 1 is more resonant)
17
18 MOOGVCF(outsk=0.0, insk=0.0, dur=3.0, amp=1.0, inputchan=0, pan=1.0, bypass=0, filterfreqtable=1000.0, filtressontable=0.3)
19 MOOGVCF(outsk=0.0, insk=0.0, dur=3.0, amp=1.0, inputchan=1, pan=0.0, bypass=0, filterfreqtable=800.0, filtressontable=0.8)
```

Cursor Line: 12 Insertion Point Line: 1

10. Dynamic LPF "Sweeping"

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the MOOGVCF Instrument:
9 // MOOGVCF(outsk, insk, dur, AMP, inputchan, PAN, BYPASS, FILTFREQTABLE, FILTRESONTABLE)
10 // source: http://rtcmix.org/reference/instruments/MOOGVCF.php
11
12 // "Moog" lowpass filter
13
14 // BYPASS -> bypass this filter (0: bypass off, 1: bypass on)
15 // FILTFREQTABLE -> filter cutoff frequency (Hz)
16 // FILTRESONTABLE -> filter resonance (0-1, 1 is more resonant)
17
18 // the cutoff frequency can be controlled by a dynamic maketable()
19
20 freq = maketable("line", "nonorm", 1000, 0,600, 40,1900, 100,400)
21
22 // Optional arg. "nonorm" turns off normalization (scaling of table b/t 0.0-1.0).
23 // We need this b/c we are specifying EXACT frequencies values for FILTFREQTABLE!
24
25 MOOGVCF(outsk=0.0, insk=0.0, dur=3.0, amp=1.0, inputchan=0, pan=0.5, bypass=0, filterfreqtable=freq, filtresontable=0.9)
26 MOOGVCF(outsk=0.0, insk=0.0, dur=3.0, amp=1.0, inputchan=1, pan=0.0, bypass=0, filterfreqtable=freq, filtresontable=0.9)
27
28 // Notice we give the filter a very high resonance (0.9) so that we clearly hear the "sweep."
```

Cursor Line: 15 Insertion Point Line: 1

11. Basic Schroeder Reverb

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtinput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // Syntax for the REVERBIT Instrument:
9 // REVERBIT(outsk, insk, dur, AMP, RVBTIME, RVBAMT, chan0delay, FILTFREQ[, dcblock, ringdowndur])
10 // source: http://rtcmix.org/reference/instruments/REVERBIT.php
11
12 REVERBIT(outskip=0.0, inskip=0.0, dur=9999, amp=1.0, revtime=2.6, revpct=0.4, rtchanel=0.02, cutoff=5000.0)
```

Cursor Line: 11 Insertion Point Line: 1

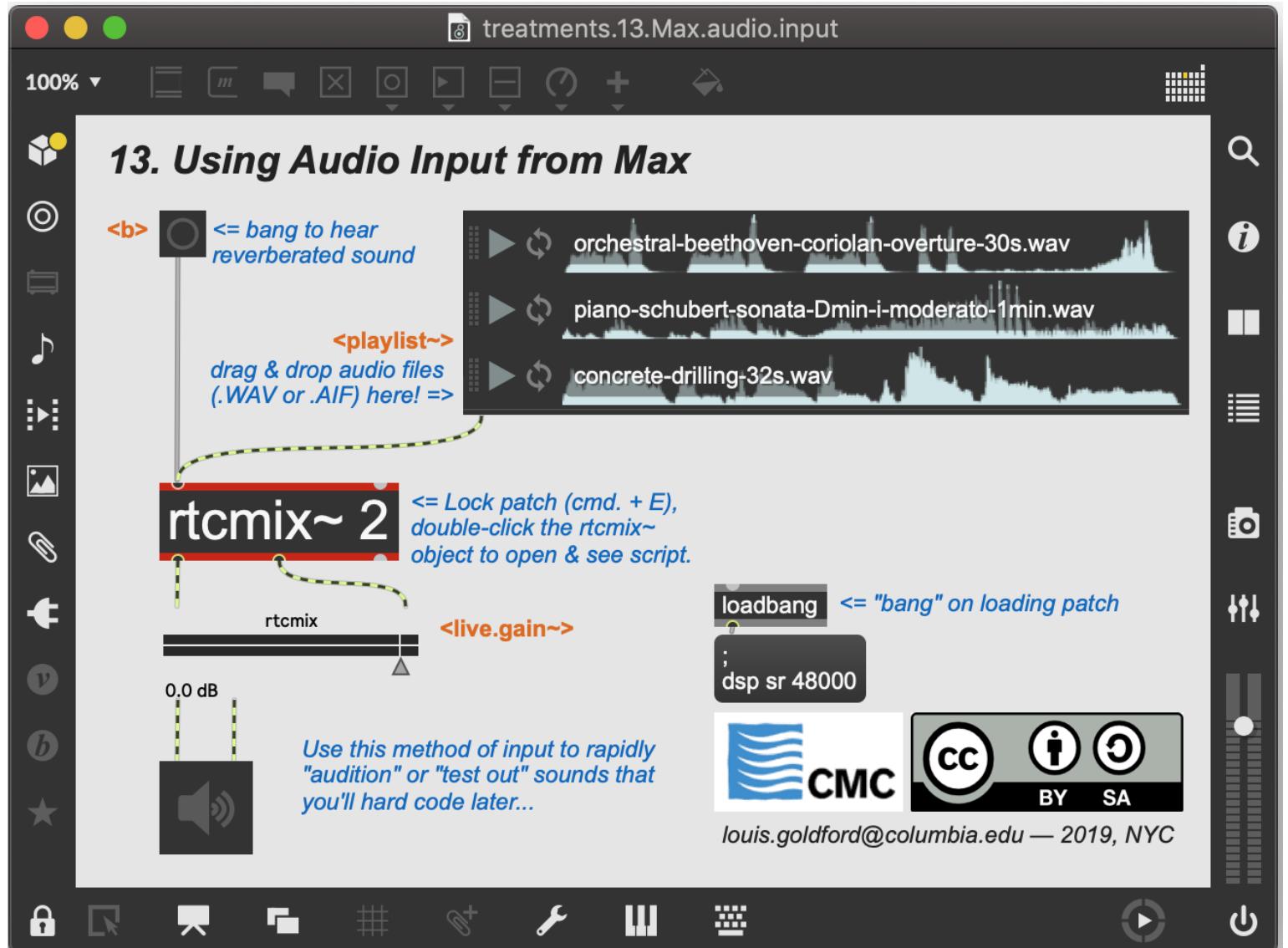
12. A Basic Signal Chain:

Filter → Reverb

```
script_0
1 // To load a sound file from disk, FULL PATH are required for the rtcmix~ object in Max.
2 // A relative pathname (e.g. "test.samples/stereo-test.aif") will not work. :(
3 // Remember: You can drag/drop a sound file onto a fresh terminal window to get its full path!
4 // Replace the following FULL PATH to this sound file with one on YOUR computer:
5
6 rtininput("/Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.03.treatments/test.samples/stereo-test.aif")
7
8 // We use bus_config() like a mixing board; to connect the soundfile -> MOOGVCF -> REVERBIT -> outlets in Max patch.
9
10 bus_config("MOOGVCF", "in 0-1", "aux 0-1 out")
11     // "in 0-1" input from sound file (rtininput)
12     // "aux 0-1 out" auxiliary "audio busses" allowing us to connect RTcmix instruments
13 bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
14     // "aux 0-1 in" receive input from MOOGVCF
15     // "out 0-1" output to the Max object's outlets (in our outside patch)
16
17 freq = maketable("line", "nonorm", 1000, 0, 2000, 25, 200, 50, 500, 100, 1100)
18
19 // Syntax for the MOOGVCF Instrument:
20 // MOOGVCF(outsk, insk, dur, AMP, inputchan, PAN, BYPASS, FILTFREQTABLE, FILTRESONTABLE)
21
22 MOOGVCF(outsk=0.0, insk=0.0, dur=3.0, amp=1.0, inputchan=0.0, pan=0.5, bypass=0, freq, filtressontable=0.9)
23
24 // Syntax for the REVERBIT Instrument:
25 // REVERBIT(outsk, insk, dur, AMP, RVBTIME, RVBAMT, chan0delay, FILTFREQ[], dcblock, ringdowndur)
26
27 REVERBIT(outskip=0.0, inskip=0.0, dur=9999, amp=1.0, revtime=2.6, revpct=0.4, rtchanel=0.02, cutoff=5000.0)
```

Cursor Line: 18 Insertion Point Line: 1

13. Audio Input From Max



13.1 rtinput("AUDIO")

```
script_0
1 // We can alternatively process live a input audio stream coming directly from Max.
2 // Instead of a path to an audio file, we can specify:
3
4 rtinput("AUDIO")
5
6 // We use bus_config() like a mixing board; to connect the soundfile -> MOOGVCF -> REVERBIT -> outlets in Max patch.
7
8 bus_config("MOOGVCF", "in 0-1", "aux 0-1 out")
9     // "in 0-1" input from sound file (rtinput)
10    // "aux 0-1 out" auxiliary "audio busses" allowing us to connect RTcmix instruments
11 bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
12     // "aux 0-1 in" receive input from MOOGVCF
13     // "out 0-1" output to the Max object's outlets (in our outside patch)
14
15 freq = maketable("line", "nonorm", 1000, 0, 2000, 25, 200, 50, 500, 100, 1100)
16
17 // Syntax for the MOOGVCF Instrument:
18 // MOOGVCF(outsk, insk, dur, AMP, inputchan, PAN, BYPASS, FILTFREQTABLE, FILTRESONTABLE)
19
20 MOOGVCF(outsk=0.0, insk=0.0, dur=9999, amp=1.0, inputchan=0.0, pan=0.5, bypass=0, freq, filtressontable=0.9)
21
22 // ...increased dur to 9999 here as well; to allow for indefinite real-time processing
23
24 // Syntax for the REVERBIT Instrument:
25 // REVERBIT(outsk, insk, dur, AMP, RVBTIME, RVBAMT, chan0delay, FILTFREQ[], dcblock, ringdowndur[])
26
27 REVERBIT(outskip=0.0, inskip=0.0, dur=9999, amp=1.0, revtime=2.6, revpct=0.4, rtchanel=0.02, cutoff=5000.0)
```

Cursor Line: 19 Insertion Point Line: 1

14. Amplitude Modulation

treatments.14.amplitude.modulation

100% □ ⊞ ⊖ ⊘ ⊙ ⊛ ⊕ ⊖ ⊗

14. AM (Amplitude Modulation)

 bang to hear reverberated sound

<playlist>
drag & drop audio files (.WAV or .AIF here! =>

orchestral-beethoven-coriolan-overture-30s.wav
piano-schubert-sonata-Dmin-i-moderato-1min.wav
concrete-drilling-32s.wav

loadmess 378.

378. <= ring modulator frequency in Hz.
Change this dynamically in your patch (works well with piano sounds...)

rtcmix~ 2 1
rtcmix
<live.gain~>

0.0 dB

Use this method of input to rapidly "audition" or "test out" sounds that you'll hard code later...

loadbang <= "bang" on loading patch
dsp sr 48000

CMC CC BY SA

louis.goldford@columbia.edu — 2019, NYC

14. AM (Amplitude Modulation)

 bang to hear reverberated sound

<playlist>
drag & drop audio files (.WAV or .AIF here! =>

orchestral-beethoven-coriolan-overture-30s.wav
piano-schubert-sonata-Dmin-i-moderato-1min.wav
concrete-drilling-32s.wav

loadmess 378.

378. <= ring modulator frequency in Hz.
Change this dynamically in your patch (works well with piano sounds...)

rtcmix~ 2 1
rtcmix
<live.gain~>

0.0 dB

Use this method of input to rapidly "audition" or "test out" sounds that you'll hard code later...

loadbang <= "bang" on loading patch
dsp sr 48000

CMC CC BY SA

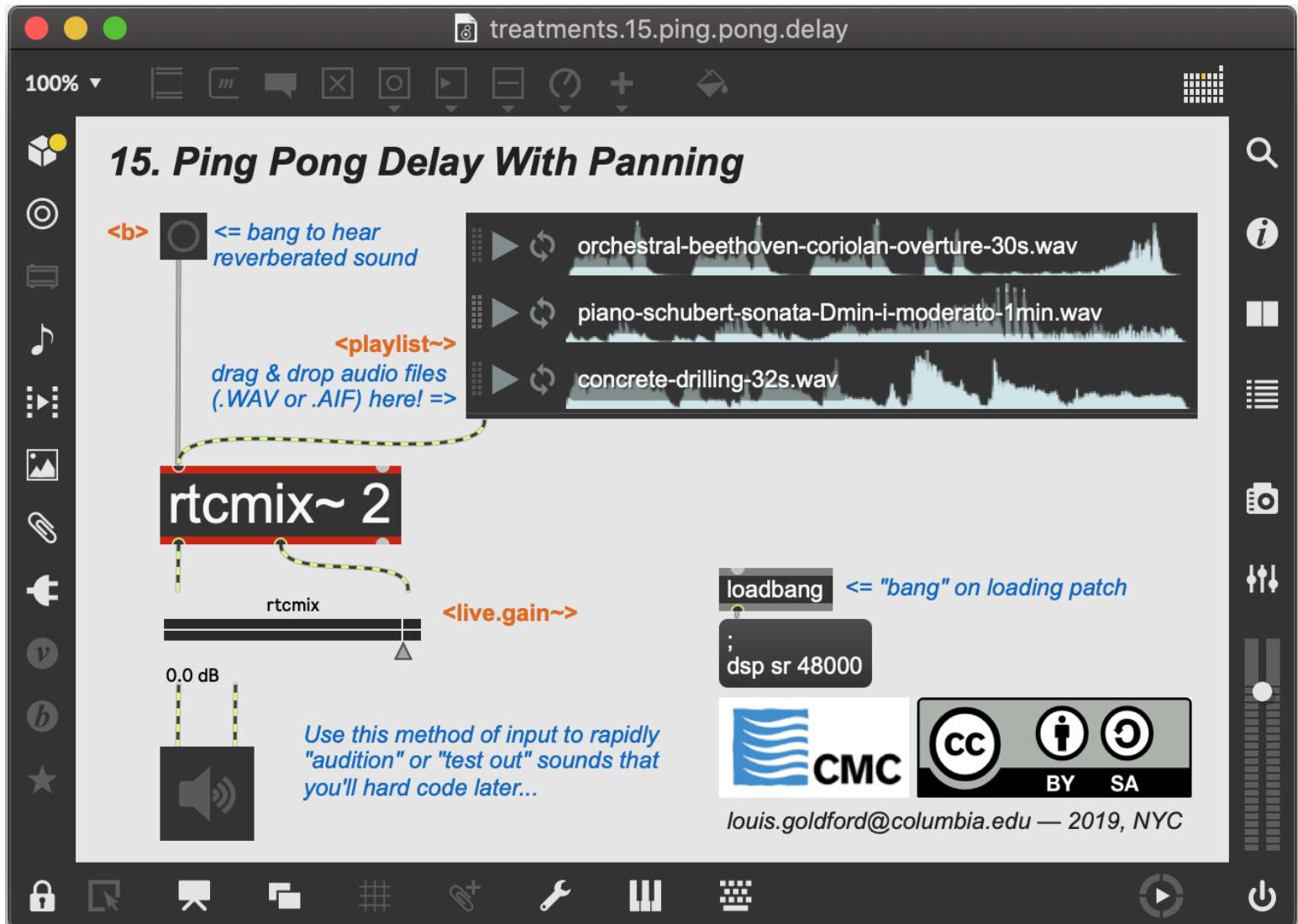
louis.goldford@columbia.edu — 2019, NYC

14.1 Makeconnection

```
script_0
1 // We can alternatively process live a input audio stream coming directly from Max.
2 // Instead of a path to an audio file, we can specify:
3
4 rtinput("AUDIO")
5
6 // We use bus_config() like a mixing board; to connect the soundfile -> MOOGVCF -> REVERBIT -> outlets in Max patch.
7
8 bus_config("AM", "in 0-1", "aux 0-1 out")
9     // "in 0-1" input from sound file (rtinput)
10    // "aux 0-1 out" auxiliary "audio busses" allowing us to connect RTcmix instruments
11 bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
12     // "aux 0-1 in" receive input from MOOGVCF
13     // "out 0-1" output to the Max object's outlets (in our outside patch)
14
15 freq = maketable("line", "nonorm", 1000, 0, 2000, 25, 200, 50, 500, 100, 1100)
16
17 // Syntax for the AM Instrument:
18 // AM(outsk, insk, dur, AMP, MODFREQ[, inputchan, PAN, MODWAVETABLE])
19 // ring modulate the entire soundfile with a 378 Hz modulator
20
21 modulator = makeconnection("inlet", 1, 378.0)
22     // Here we declare a variable controlled by live data input from within the Max patch...
23
24 AM(0.0, 0.0, dur=9999, 1.0, modulator)
25     // Now, our live input will control the modulator frequency. Try different frequencies in the patch...
26
27 // ...increased dur to 9999 here as well; to allow for indefinite real-time processing
28
29 // Syntax for the REVERBIT Instrument:
30 // REVERBIT(outsk, insk, dur, AMP, RVBTIME, RVBAMT, chan0delay, FILTFREQ[, dcblock, ringdowndur])
31
32 REVERBIT(outskip=0.0, inskip=0.0, dur=9999, amp=1.0, revtime=2.6, revpct=0.4, rchanel=0.02, cutoff=5000.0)
```

Cursor Line: 19 Insertion Point Line: 1

15. Ping Pong Delay



15.1 PANECHO()

```
script_0
1 rtinput("AUDIO")
2
3 start = 0
4 instart = 0
5 duration = 60
6 amplitude = 1.0
7 envelope = maketable("window", 1000, "hanning")
8 channel0delay = 0.25
9 // in seconds
10 channel1delay = 0.50
11 delay_feedback = 0.8
12 // ALWAYS less than 1.0 or a crazy feedback loop ensues!
13 ringdowndur = 1.0
14
15 // Syntax for PANECHO Instrument:
16 // source: http://rtcmix.org/reference/instruments/PANECHO.php
17
18 PANECHO(start, instart, duration, amplitude*envelope, channel0delay, channel1delay, delay_feedback, ringdowndur)
19
20 // source: 'Computer Music Composition with RTcmix' by Jerod Sommerfeldt, p. 86.
21 // http://sites.music.columbia.edu/cmc/courses/g6611/spring2017/week2/2015-0626_RTcmixbookrough1.pdf
22
```

Cursor Line: 7 Insertion Point Line: 1