

List of tidal effects and their description

Basic effects

Delay

(#delay, #delaytime/#delayt, #delayfeedback/#delayfb)

Values from 0 to 1

Leslie

(#leslie, #lrate, #lsize)

"Leslie" controls dry/wet. "lrate" is the modulation rate (typical vintage rates would be 6.7 for "fast", 0.7 for "slow"). "lsize" is the physical size of the cabinet in meters, this mostly affects the Doppler amount (pitch warble)

Reverb

(parameters: #room, #size, #dry)

Thanks to Jost Muxfeld and James McCartney. Note that "size" is not room size, just a depth metaphor. Values from 0 to 1

Crush

(parameters: #crush)

Bit crushing, a pattern of numbers from 1 (for drastic reduction in bit-depth) to 16 (for barely no reduction).

Legato

(parameters: #legato)

Legato controls the amount of overlap between two adjacent synth sounds

Tremolo

(parameters: #tremolodepth/#tremdp, #tremolorate/#tremr)

Shape

(parameters: #shape)

A type of amplifier, values from 0 to 1

Coarse

(parameters: #coarse)

Fake-resampling, a pattern of numbers for lowering the sample rate, i.e. 1 for original 2 for half, 3 for a third and so on.

Phaser

(parameters: #phaserrate/#phasr, #phaserdepth/#phasdp)

Vowel

(parameters: #vowel)

Formant filter to make things sound like vowels, a pattern of either `a`, `e`, `i`, `o` or `u`. Use a rest (`~`) for no effect.

Lowpass filter

(#cutoff/#lpf, #resonance/#lpq)

Cutoff as Hz. Resonance from 0 to 1

Highpass filter

(#hcutoff/#hpf, #hresonance/#hpq)

Cutoff as Hz. Resonance from 0 to 1

Bandpass filter

(#bandf/#bpf, #bandq/#bpfq).

Cutoff as Hz. Resonance from 0 to 1

ASR-Envelope

(#attack/#att, #hold, #release/#rel)

Values in seconds.

MIDI specific parameters

(parameters: #dur, #modwheel, #expression, #sustainpedal)

Extra effects + SuperCollider help file description

Waveloss

(parameters: #waveloss)

(Calum Gunn)

Divides an audio stream into tiny segments, using the signal's zero-crossings as segment boundaries, and discards a fraction of them.

(SuperCollider documentation for Waveloss.ar:

Divide an audio stream into tiny segments, using the signal's zero-crossings as segment boundaries, and discard a fraction of them (i.e. replace them with silence of the same length). The technique was described by Trevor Wishart in a lecture.

Parameters: the filter drops drop out of out of chunks. mode can be 1 to drop chunks in a simple deterministic fashion (e.g. always dropping the first 30 out of a set of 40 segments), or 2 to drop chunks randomly but in an appropriate proportion.)

Squiz

(parameters: #squiz)

(Calum Gunn)

Reminiscent of some weird mixture of filter, ring-modulator and pitch-shifter

(SuperCollider documentation for Squiz.ar:

A simplistic pitch-raising algorithm. It's not meant to sound natural; its sound is reminiscent of some weird mixture of filter, ring-modulator and pitch-shifter, depending on the input. The algorithm works by cutting the signal into fragments (delimited by upwards-going zero-crossings) and squeezing those fragments in the time domain (i.e. simply playing them back faster than they came in), leaving silences inbetween. All the parameters apart from memlen can be modulated.

This UGen is dedicated to Suburban Base Records. (It doesn't sound like them, but was half-inspired by them.))

Frequency shifter

(parameters: #fshift, #fshiftnote, #fshiftphase)

(Ben Gold)

Total shift is sum of `fshift` (in Hz) and `fshiftnote` times the current note frequency.
`fshiftphase` allows control over the phase

(SuperCollider documentation for FreqShift.ar:

FreqShift implements single sideband amplitude modulation, also known as frequency shifting, but not to be confused with pitch shifting. Frequency shifting moves all the components of a signal by a fixed amount but does not preserve the original harmonic relationships.)

Triode

(parameters: #triode)

(Ben Gold)

Triode-like distortion, uses only the `triode` parameter

Krush

(parameters: #krush, #kcutoff)

(Ben Gold)

Sonic Pi's krush modified a bit so krush "0" is the same as dry signal uses `krush` and `kcutoff` as parameters

Octer

(parameters: #octer, #octersub, #octersubsub)

(Ben Gold)

Sonic Pi's octaver uses `octer` for octave harmonics, `octersub` for half-frequency harmonics, and `octersubsub` for quarter-frequency harmonics.

Ring

(parameters: #ring, #ringf, #ringdf)
(Ben Gold)

Ring modulation with `ring` (modulation amount), `ringf` (modulation frequency), and `ringdf` (slide in modulation frequency)

Distort

(parameters: #distort)
(Ben Gold)

A crunchy distortion with a lot of high harmonics, the only parameter is `distort`

Spectral delay

(parameters: #xsdelay, #tsdelay)
(Mads Kjeldgaard)

(No description)

Magnitude freeze

(parameters: #freeze)
(Mads Kjeldgaard)

Accepts integers: 1 freezes the audio, 0 doesn't.

(SuperCollider documentation for PV_Freeze:

Freezes magnitudes at current levels when freeze > 0, and advances phase according to difference between frames to try and maintain correct spectral quality.)

Spectral comb filter

(parameters: #comb)
(Mads Kjeldgaard)

Number of teeth and width of the comb are all controlled using one floating point number

(SuperCollider documentation for PV_RectComb:

Makes a series of gaps in a spectrum)

Bin scrambling

(parameters: #scram)

(Mads Kjeldgaard)

Accepts floats to control the width and placement of the scrambling in the spectrum

(SuperCollider documentation for PV_BinScramble:

Randomizes the order of the bins. The trigger will select a new random ordering.)

Magnitude smearing

(parameters: #smear)

(Mads Kjeldgaard)

Accepts floats to determine the amount of smearing.

(SuperCollider documentation for PV_MagSmear:

Average a bin's magnitude with its neighbors.)

Bin shifting

(parameters: #binshift)

(Mads Kjeldgaard)

Both the stretching and shifting of bins are controlled by supplying this parameter with floats.

(SuperCollider documentation for: PV_BinShift:

Shift and scale the positions of the bins. Can be used as a very crude frequency shifter/scaler.)

Spectral high pass filter

(parameters: #hbrick)

(Mads Kjeldgaard)

Accepts floats in values ranging from 0.0 to 1.0

(SuperCollider documentation for PV_BrickWall:

Clears bins above or below a cutoff point.)

Spectral low pass filter

(parameters: #lbrick)

(Mads Kjeldgaard)

Accepts floats in values ranging from 0.0 to 1.0

(SuperCollider documentation for PV_BrickWall:

Clears bins above or below a cutoff point.)

Spectral conformer

(parameters: #real, #imag)
(Mads Kjeldgaard)

(No Description)

(SuperCollider documentation for PV_ConformalMap:

Applies the conformal mapping

$$z \rightarrow (z - a) / (1 - za^*)$$

to the phase vocoder bins z with a given by the real and imag inputs to the UGen.

Makes a transformation of the complex plane so the output is full of phase vocoder artifacts but may be musically fun. Usually keep

$$|a| < 1$$

but you can of course try bigger values to make it really noisy.

$$a = 0$$

should give back the input mostly unperturbed.

See <http://mathworld.wolfram.com/ConformalMapping.html> .)

Spectral enhance

(parameters: #enhance)
(Mads Kjeldgaard)

(No description)

DJ Filter

(parameters: #djf)
(Alex McLean)

DJ filter, a low pass filter for the first half of the range, and a high pass for the rest.

Basic SuperDirt synths

Imp

(#imp parameters: #sustain, #accelerate)

Modulated band limited impulse

Psin

(#psin parameters: #sustain, #accelerate)

Modulated phase mod sines

Pmsin

(#pmsin parameters: #sustain, #accelerate)

Cursor modulated phase mod sines

In

(#in)

Live audio input

Inr

(#inr parameters: #accelerate)

Pitch shifted live audio input

Gabor

(#gabor parameters: #sustain)

Gabor grain

Cyclo

(#cyclo parameters: #sustain, #accelerate)

Shepard on a cycle

Extra SuperDirt synths

(Ben Gold)

Supermandolin

(#supermandolin parameters: #sustain, #accelerate)

Physical modeling of a vibrating string, using a delay line (CombL) excited by an initial pulse (Impulse). To make it a bit richer, I've combined two slightly detuned delay lines

"accelerate" is used for a pitch glide, and "sustain" changes the envelope timescale

Supergong

(#supergong parameters: #voice, #accelerate)

An example of additive synthesis, building up a gong-like noise from a sum of sine-wave harmonics. Notice how the envelope timescale and amplitude can be scaled as a function of the harmonic frequency. "voice" provides something like a tone knob, and "decay" adjusts how the harmonics decay as in the other SynthDefs, "sustain" affects the overall envelope timescale and "accelerate" for pitch glide.

For a demo, try this in Tidal:

```
d1 $ n (slow 2 $ fmap (*7) $ run 8) # s "supergong" # decay "[1 0.2]/4" # voice "[0.5 0]/8"
```

Superpiano

(#superpiano parameters: #velocity, #sustain)

Hooking into a nice synth piano already in Supercollider. Uses the "velocity" parameter to affect how hard the keys are pressed. "sustain" controls envelope and decay time

(Supercollider documentation for MdaPiano.ar:

A piano synthesiser (*originally a VST plugin by Paul Kellett, ported to SC by Dan Stowell*). This UGen is not polyphonic (but can be retriggered to play notes in sequence).

Superhex

(#superhex parameters: #sustain, #accelerate)

Waveguide mesh, hexagonal drum-like membrane. "sustain" controls envelope and decay time. "accelerate" controls tension.

(Supercollider documentation for MembraneHexagon.ar:

MembraneCircle and MembraneHexagon are rather similar, being triangular waveguide meshes of a drum-like membrane. You input some excitation, such as a pulse of noise, and can adjust the tension and loss while it plays.

They're named after the shape made out of triangular meshes. Obviously you can't make a circle out of triangles, but it tries. At the moment MembraneCircle is a bit bigger than MembraneHexagon, using more waveguides and therefore more CPU.)

Superkick

(#superkick parameters: #n, #sustain, #accelerate, #pitch1, #decay)

Kick Drum using Rumble-San's implementation as a starting point

(<http://blog.rumblesan.com/post/53271713518/drum-sounds-in-supercollider-part-1>). "n"

controls the kick frequency in a nonstandard way. "sustain" affects overall envelope timescale, "accelerate" sweeps the click filter freq, "pitch1" affects the click frequency, and "decay" changes the click duration relative to the overall timescale.

Super808

(#super808 parameters: #n, #sustain, #speed, #voice)

A vaguely 808-ish kick drum. "n" controls the chirp frequency, "sustain" the overall timescale, "speed" the filter sweep speed, and "voice" the sinewave feedback

Superhat

(#superhat parameters: #n, #sustain, #accelerate)

Hi-hat using Rumble-San's implementation as a starting point

(<http://blog.rumblesan.com/post/53271713518/drum-sounds-in-supercollider-part-1>). Using "n" in a weird way to provide some variation on the frequency. "sustain" affects the overall envelope rate, "accelerate" sweeps the filter.

Supersnare

(#supersnare parameters: #n, #decay, #sustain, #accelerate)

Snare drum using Rumble-San's implementation as a starting point

(<http://blog.rumblesan.com/post/53271713909/drum-sounds-in-supercollider-part-2>) Again using "n" for some variation on frequency, "decay" for scaling noise duration relative to tonal part, "sustain" for overall timescale, "accelerate" for tonal glide.

Superclap

(#superclap parameters: #n, #delay, #speed, #pitch1, #sustain)

Hand clap using Rumble-San's implementation as a starting point

(<http://blog.rumblesan.com/post/53271713909/drum-sounds-in-supercollider-part-2>) "delay" controls the echo delay, "speed" will affect the decay time, "n" changes how spread is calculated, "pitch1" will scale the bandpass frequency, and "sustain" the overall timescale.

Supersiren

(#supersiren parameters: #sustain)

A controllable synth siren, defaults to 1 second, draw it out with "sustain".

The next four synths respond to the following parameters in addition to gain, pan, n, and all the "effect" parameters (including attack, hold, and release). Default values in parentheses.

sustain - scales overall duration

decay(0) - amount of decay after initial attack

accelerate(0) - pitch glide

semitone(12) - how far off in pitch the secondary oscillator is (need not be integer)

pitch1(1) - filter frequency scaling multiplier, the frequency itself follows the pitch set by "n"

speed(1)- LFO rate

lfo(1) - how much the LFO affects the filter frequency

resonance(0.2) - filter resonance

voice(0.5) - depends on the individual synth

Supersquare

(#supersquare parameters: #voice + see above)

A moog-inspired square-wave synth; variable-width pulses with filter frequency modulated by an LFO. "voice" controls the pulse width (exactly zero or one will make no sound)

Supersaw

(#supersaw parameters: #voice + see above)

A moog-inspired sawtooth synth; slightly detuned saws with triangle harmonics, filter frequency modulated by LFO. "voice" controls a relative phase and detune amount

Superpwm

(#superpwm parameters: #voice + see above)

A moog-inspired PWM synth; pulses multiplied by phase-shifted pulses, double filtering with an envelope on the second "voice" controls the phase shift rate.

Supercomparator

(#supercomparator parameters: #voice + see above)

"voice" scales the comparator frequencies, higher values will sound "breathier".

Superchip

(#superchip parameters: #slide, #speed, #accelerate, #pitch2, #pitch3, #voice)

Uses the Atari ST emulation UGen with 3 oscillators. "slide" is for a linear frequency glide that will repeat "speed" times (can be fractional or negative). "accelerate" is for an overall glide, "pitch2" and "pitch3" control the ratio of harmonics. "voice" causes variations in the levels of the 3 oscillators

Supernoise

(#supernoise parameters: #n, #voice, #accelerate, #speed, #pitch1, #slide, #resonance)

Digital noise in several flavors with a bandpass filter. "voice" at 0 is a digital noise for which "n" controls rate, at 1 is Brown+White noise for which "n" controls knee frequency. "accelerate" causes glide in n, "speed" will cause it to repeat. "pitch1" scales the bandpass frequency (which tracks "n"). "slide" works like accelerate on the bandpass. "resonance" is the filter resonance.

Superfork

(#superfork parameters: #sustain, #accelerate)

Tuning fork from Ben Gold's experimentation and from "On the acoustics of tuning forks", Rossing Russell and Brown.

Superhammond

(#superhammond parameters: #voice, #decay)

Hammond B3 sim. Freq adjustments courtesy Tom Wiltshire (electricdruid.net). Drawbar settings selectable with "voice" and from many public domain sources, Google "Hammond drawbar settings" and add your own in the big SelectX block (in SC). "perc" "percf" and "decay" are an attempt at the percussion, no idea if it sounds at all reasonable. vintage Hammonds had percf as 2 or 3 (switchable), two perc levels (maybe roughly 0.7 and 1.2?), and two decay options (roughly 0 and maybe 1ish?). "vibrato", "vrate", "perc", "percf" are all new params you'll need to define in Tidal if you want to change them.

Supervibe

(#supervibe parameters: #decay, #velocity, #accelerate, #modamp, #modfreq, #detune)

Vibraphone simulation, adapted with some help from Kevin Larke's thesis Real Time Vibraphone Pitch and Timbre Classification.

"decay" - use larger values to damp higher harmonics (default 0)

"velocity" - higher velocity will brighten the sound a bit (default 1)

"accelerate" - for a linear pitch bend

"modamp" - amplitude of the tremolo (0-2 is OK, default 1)

"modfreq" - frequency of the tremolo (default 7)

"detune" - adjusts a high harmonic to give the sound a different character

Superhoover

(#superhoover parameters: #slide, #decay, #accelerate)

Hoover, adapted from Wouter Snoei's

(<http://superdupercollider.blogspot.com/2009/06/more-dominator-deconstruction.html>)

"slide" is used for the amount of initial pitch glide, positive slides up in pitch, negative slides down. also responds to "decay" for a different envelope shape and "accelerate" for constant pitch glide. uncomment the two lines with "rrand" for a different every time you evaluate it (in SC).

Superzow

(#superzow parameters: #decay, #accelerate, #slide, #detune)

Phased saws. "decay" and "accelerate" used as in some synths above, for envelope shaping and pitch bend, respectively. "slide" here is how fast it moves through the phase (default 1) "detune" is for oscillator detuning and defaults to 1.

Superstatic

(#superstatic parameters: #sustain)

Impulse noise with a fadein/out. "sustain" affects the overall envelope rate

Supergrind

(#supergrind parameters: #accelerate, #detune, #voice, #speed)

From synthdef.art fragment: 2018-08-16. Responds to accelerate (for pitch glide), detune (in Hz, but even small values are quite noticable), voice (changes harmonics), speed (impulse trigger rate)

Superprime

(#superprime parameters: #accelerate, #detune, #voice, #speed)

From synthdef.art: #nightofprimes. Responds to accelerate (for pitch glide), detune (fractional), voice (some subtle shaping and reverb changing), speed (impulse trigger rate).

Superwavemechanics

(#superwavemechanics parameters: #accelerate, #detune, #voice, #resonance)

From synthdef.art, #wavemechanics. Responds to accelerate (for pitch glide). detune (for pitch randomization). voice (to color/whiten the noise). resonance (affects reverberation)

Supertron

(#supertron parameters: #accelerate, #detune, #voice)

Feedback PWM. Can use "accelerate" "voice" and "detune" parameters.

Try ``d1 $ s "supertron" # octave 3 # accelerate "0.2"`

Superreese

(#parameters: #accelerate, #voice, #detune)

Vaguely Reese-like synth. Can use "accelerate" "voice" and "detune" parameters