Composers' Desktop Project

CDP Functions for STRANGE Effects

(with Command Line Usage)

Functions to Create STRANGE Relationships Between Partials

GLIS
Create glissandi inside the (changing) spectral envelope of the original sound

INVERT
Invert the spectrum

SHIFT
Linear frequency shift of (part of) the spectrum

WAVER
Oscillate between harmonic and inharmonic state

STRANGE GLIS – create glissandi inside the (changing) spectral envelope of the original sound

Usage

strange glis 1 *infile outfile* **-f***N*| **-p***N* [**-i**] *glisrate* **-t***topfrq*
strange glis 2 *infile outfile* **-f***N*| **-p***N* [**-i**] *glisrate hzstep* **-t***topfrq*
strange glis 3 *infile outfile* **-f***N*| **-p***N* [**-i**] *glisrate* **-t***topfrq*

Modes

1 Shepard tones
2 inharmonic glide
3 self-glissando

Parameters

infile – input analysis file made with PVOC
outfile – output analysis file
-f extract formant envelope linear frequency-wise, using 1 point for every *N* equally-spaced frequency-channels
-p extract formant envelope linear pitchwise, using *N* equally-spaced pitch-bands per octave
-i quicksearch for formants (less accurate)
glisrate rate of glissing in semitones per second (negative value for downward glissando)
hzstep partials spacing in inharmonic glide. Range: FROM channel-frq-width (usually 43Hz) TO Nyquist/2 (i.e., ¼ the sample rate)
-t*topfrq* top of spectrum: must be > 2*channel width of the analysis. Default: Nyquist (i.e., ½ the sample rate)

Understanding the STRANGE GLIS Process

This program extracts the time-changing spectral contour (the spectral trajectory) of the sound. It therefore retains any spectral articulation, such as patterns of speech, but replaces the signal by an endlessly rising (positive values for *glissrate*) or falling (negative values for *glissrate*) shepard tone or inharmonic glissando. It uses linear interpolation in determining the spectral contour.

In Mode **1** (Shepard tones)

the key parameter appears to be *glissrate*. A high positive or negative value (e.g., + or -50) introduces quite a bit of noise into the sound, whereas a low value such as 0.3 results in a twangy vibrating sound.

In Mode **2** (inharmonic glissando)

> the two most salient parameters are *glissrate* and *spacing*. Higher postive or lower negative values for *glissrate*, e.g., +24 or -24, +100 or -100, gives a very rapidly cycling glissando (a pretty mushy twang). Low values for *spacing* (e.g., 45) gives a fairly deep twang, whereas high values, e.g. 1101, results in a rather hollow sound.
>
> The number of channels (frequency-wise formant extraction) or bands per octave (pitchwise) affects the resolution of the analysis without changing the resulting sound effect very much, and adjusting the top frequency can offer a certain degree of filtering control.

In Mode **3** (self-glissando)

> the original tonal quality of the sound is largely preserved, especially with low values for *glissrate*. With higher values of *glissrate*, a higher, somewhat hollow glissando is added to the sound, otherwise very much like the original input.

Musical Applications

In effect, the original sound is replaced by the glisandoing tone, but the articulations of the original are retained, i.e., the rhythm and attack transients of the spectral envelope are notably the same. However, the tone of the sound is greatly altered, acquiring in Mode **2** as noted above a mushy twang with low values for *spacing* and a rather hollow vibrating sound with higher values for *spacing*. One set of trials on a low drum sound transformed it into something not unlike a didgerdoo.

Because formants are involved, input sounds with significant resonance features work well with this function, e.g., vocal and some drum sounds, especially those lower in the frequency range.

End of STRANGE GLIS

STRANGE INVERT – invert the spectral envelope

Usage

strange invert mode *infile outfile*

Modes

1 Normal inversion
2 Output sound retains the amplitude envelope of the source sound

Parameters

infile – input analysis file made with PVOC
outfile – output analysis file

Understanding the STRANGE INVERT Process

STRANGE INVERT is derived from an idea developed by Allen Strange. It inverts the spectral envelope, relative to the overall spectral envelope. This means that the energy asociated with the highest frequency bands is transferred to the lowest ones and vice versa. As the lowest partials in many sounds have the greatest amplitude and the highest ones the least, the result is typically a much brighter timbre. Mode 2 retains the *amplitude* envelope of the sound, so will sound closer to the original, with just the timbral change.

STRANGE INVERT does NOT invert the frequencies themselves. For this see SPECNU SLICE, Mode 5.

This function was formerly SPEC INV, which required that its *infile* be generated by SPEC BARE (zeros channels which don't contain harmonics). The SPEC BARE function is now built into STRANGE INVERT, meaning that the result may have more of a 'harmonic' quality than it might have. This will be related to the degree that the original sound was distinctly pitched. The main reason for including the BARE function, however, was simply to make the spectrum 'easier to manipulate'; it doesn't go so far as to harmonise the sound as such (see the PITCH functions for this).

Musical Applications

The spectral envelope is shaped by the amplitude level of the various frequencies, changing through time. Inverting this shape should reduce the level of some frequencies, while making others more prominent. So what should be most noticeable would be a timbral change in the sound, perhaps deeper and richer in character. But it will be important to explore the possibilities by submitting different types of source to it, ranging from strong, focussed pitches, to sounds of a more amorphous nature.

End of STRANGE INVERT

STRANGE SHIFT – Linear frequency shift of (part of) the spectrum

Usage

strange shift 1 *infile outfile frqshift* [**-l**]
strange shift 2-3 *infile outfile frqshift frqdivide* [**-l**]
strange shift 4-5 *infile outfile frqshift frqlo frqhi* [**-l**]

Modes

1 Shift the whole spectrum
2 Shift the spectrum above frq_divide
3 Shift the spectrum below frq_divide
4 Shift the spectrum only inside the range *frqlo frqhi*
5 Shift the spectrum only outside the range *frqlo frqhi*

Parameters

infile – input analysis file made with PVOC
outfile – output analysis file
frqshift linear shift in Hz of spectral frequencies (same for all); for downward shifts, use -*frqshift*, i.e., a negative value (Range: -22050 to 22050)
frq_divide frequency at which the shifting starts or stops
frqlo frqhi define a range inside or outside of which shifting takes place
-l logarithmic interpolation between varying frequency values (Default: linear). (The **-l** option is useful only if any of the above input parameters are time-varying, AND *frqshift* values must be all positive or all negative for this to work.)

frqshift, frq_divide, frqlo and *frqhi* may vary over time

Understanding the STRANGE SHIFT Process

The fact that this shift is linear means that it will produce inharmonic results: it will alter any harmonic relationships between partials. The linear (i.e., additive) change distorts harmonic relationships and musical intervals, which are based on multiplication. An upwards shift reduces the intervals between partials, while a downwards shift stretches them.

Musical Applications

This function can be used to create major alterations in the timbral character of a sound. A shift upwards will emphasise high frequencies in a way which makes a dense, inharmonic sound. A shift downwards will produce a deeper, more gong-like sound.

End of STRANGE SHIFT

STRANGE WAVER – Oscillate between harmonic and inharmonic state

Usage

strange waver 1 *infile outfile vibfrq stretch botfrq*
strange waver 2 *infile outfile vibfrq stretch botfrq expon*

Modes

1 Standard spectral stretching for inharmonic state
2 Specify spectral stretching for inharmonic state

Parameters

infile – input analysis file made with PVOC
outfile – output analysis file
vibfrq the frequency of oscillation, i.e., number of oscillations per second. (Range: 1/infile_duration to 172.265628)
stretch the maximum spectral stretch in the inharmonic state (Range: 1 to 2205)
botfrq the frequency above which spectral stretching happens
expon defines the type of stretch (must be > 0)

vibfrq and *stretch* may vary over time

Understanding the STRANGE WAVER Process

STRANGE WAVER introduces an oscillation towards and away from 'inharmonicness' in the spectrum of a sound. The program moves towards inharmonicness by means of the *stretch* factor. The rate of oscillation, *vibfrq*, can be fixed or time-varying, using a *time vibfrq* breakpoint file. The effect can be very much like a frequency vibrato – it depends on the relationship between *vibfrq* and *stretch*.

The minimum value for *vibfrq* is related to the duration of the *infile*: **1/infile_duration**. Thus, durations less than 1 result in a minimum *vibfrq* which is > 1, and durations greater than 1 result in a minimum *vibfrq* which is less than 1. *Vibfrq* will always be greater than 0. The 0.32 minimum value recommended in *SoundShaper* is for an input file of about 3 sec duration. Longer input files will result in a lower a lower acceptable minimum value. For example, a 10 second file will have a minimum value of 0.1 sec. Whether you would want the effect to operate at this rate is another matter.

Vibfrq values greater than 24 start to produce a 'flutter' effect.

Higher *stretch* values will raise the perceived pitch of the output.

The *stretch* factor affects the spectral character of the sound, but does not alter the length – it is not a time-stretch. Take care with the *stretch* parameter: this function can produce 'wispy' artefacts which higher values will tend to increase.

The *vibfrq* and *stretch* factors interact in a very interesting way, which is well worth exploring.

Musical Applications

This progam is a useful way to create vibrato and flutter effects, especially via the *vibfrq* parameter. However, a very low *vibfrq* combined with a rather high *stretch* produces something quite different.

End of STRANGE WAVER

Last Updated 18 June 2015
Documentation: Archer Endrich, revised R. Fraser