spectral shapers

www.soundhack.com

Contents

End User License Agreement

CAREFULLY READ THE FOLLOWING LICENSE AGREEMENT. YOU ACCEPT AND AGREE TO BE BOUND BY THIS LICENSE AGREEMENT BY CLICKING THE ICON LABELED "I ACCEPT" THAT IS DISPLAYED BELOW. IF YOU DO NOT AGREE TO THIS LICENSE, CLICK THE ICON LABELED "CANCEL" AND YOUR ORDER WILL BE CANCELED, THE SOFTWARE WILL NOT BE DOWNLOADED AND YOU WILL NOT BE CHARGED.

License Grant

"You" means the person or company who is being licensed to use the Software or Documentation. "We," "us" and "our" means Tom Erbe, d.b.a. *SoundHack*. "Software" means *SoundHack Spectral Shapers* for Windows or for Macintosh.

We hereby grant you a nonexclusive license to use one copy of the Software on any single computer, provided the Software is in use on only one computer at any time. The Software is "in use" on a computer when it is loaded into temporary memory (RAM) or installed into the permanent memory of a computer—for example, a hard disk, CD-ROM or other storage device. If the Software is permanently installed on the hard disk or other storage device of a computer (other than a network server) and one person uses that computer more than 80% of the time, then that person may also use the Software on a portable or home computer.

Title

We remain the owner of all right, title and interest in the Software and related explanatory written materials ("Documentation").

Archival or Backup Copies

You may copy the Software for back up and archival purposes, provided that the original and each copy is kept in your possession and that your installation and use of the Software does not exceed that allowed in the "License Grant" section above.

Things You May Not Do

The Software and Documentation are protected by United States copyright laws and international treaties. You must treat the Software and Documentation like any other copyrighted material—for example, a book. You may not:

• copy the Documentation,
• copy the Software except to make archival or backup copies as provided above,
• modify or adapt the Software,
• reverse engineer, disassemble, decompile or make any attempt to discover the source code of the Software,
• place the Software onto a server so that it is accessible via a public network such as the Internet, or
• sublicense, rent, lease or lend any portion of the Software or Documentation.

Limited Warranty

We warrant that for a period of 90 days after delivery of this copy of the Software to you:

• the media on which this copy of the Software is provided to you will be free from defects in materials and workmanship under normal use, and
• the Software will perform in substantial accordance with the Documentation.

To the extent permitted by applicable law, THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND WE DISCLAIM ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING ANY IMPLIED WARRANTY OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, regardless of whether we know or had reason to know of your particular needs. No employee, agent, dealer or distributor of ours is authorized to modify this limited warranty, nor to make any additional warranties.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Limited Remedy
Our entire liability and your exclusive remedy for breach of the foregoing warranty shall be, at our option, to either:
• return the price you paid, or
• repair or replace the Software or media that does not meet the foregoing warranty if it is returned to us with a copy of your receipt.
IN NO EVENT WILL WE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR THE INABILITY TO USE THE SOFTWARE (EVEN IF WE OR AN AUTHORIZED DEALER OR DISTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF THESE DAMAGES), OR FOR ANY CLAIM BY ANY OTHER PARTY.
SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.
Term and Termination
This license agreement takes effect upon your use of the software and remains effective until terminated. You may terminate it at any time by destroying all copies of the Software and Documentation in your possession. It will also automatically terminate if you fail to comply with any term or condition of this license agreement. You agree on termination of this license to destroy all copies of the Software and Documentation in your possession.
Confidentiality
The Software contains trade secrets and proprietary know-how that belong to us and it is being made available to you in strict confidence. ANY USE OR DISCLOSURE OF THE SOFTWARE, OR OF ITS ALGORITHMS, PROTOCOLS OR INTERFACES, OTHER THAN IN STRICT ACCORDANCE WITH THIS LICENSE AGREEMENT, MAY BE ACTIONABLE AS A VIOLATION OF OUR TRADE SECRET RIGHTS.
General Provisions
1. This written license agreement is the exclusive agreement between you and us concerning the Software and Documentation and supersedes any prior purchase order, communication, advertising or representation concerning the Software.
2. This license agreement may be modified only by a writing signed by you and us.
3. In the event of litigation between you and us concerning the Software or Documentation, the prevailing party in the litigation will be entitled to recover attorney fees and expenses from the other party.
4. This license agreement is governed by the laws of the State of California.
5. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations.

1) Overview

SoundHack is a program that has been used in sound design for music and film for the last 12 years. It is a collection of spectral filters which emphasize the creative manipulation of sound. SoundHack has been used on many films – from “The Matrix” to “The City of Lost Children” – and by many recording artists – from Nine Inch Nails to Ry Cooder to J Lesser.

I am now introducing a series of plugins for VST and AU hosts based on the SoundHack program. This series of plugins, which will be released in three groups, will expand on the filters in SoundHack, and explore new spectral techniques as well:

- The first group is called **Spectral Shapers** and includes: **+morphfilter**, **+spectralgate**, **+spectralcompand** and **+binaural**. These are all filters which emphasize the reshaping of the timbre of sound.
- The second group will the emphasize the stretching, reordering and compressing of time.
- The third group will emphasize cross-synthesis, the combining and correlating of several sounds.

All of the plugins can be automated by MIDI controller messages. They are currently running under VST hosts on Mac OS 9, Mac OS X and Windows and under Audio Unit hosts on Mac OS X. I will also port them to the MOTU MAS and Digidesign RTAS format if there is enough interest.

Tom Erbe

SoundHack

2) Installation

SoundHack Spectral Shapers do not require an installation program. The user simply unpacks the downloaded archive, and copies the plugins in the unpacked folder to the VST or Audio Unit plugin directory. This directory is different under each operating system.

Macintosh OS X - VST

Copy the SoundHack Spectral Shapers from the OS X folder to /Library/Audio/Plug-Ins/VST/ on your system drive. This folder may be found easily using the Finder's "Go to Folder ..." command under the GO menu (Fig.1).

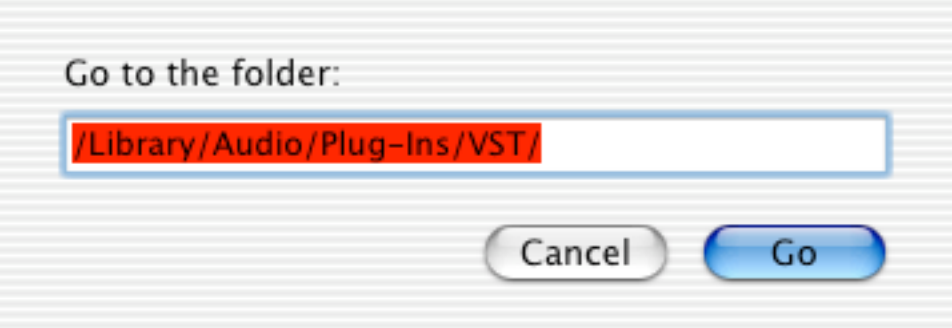


Figure 1: OS X Folder Dialog

Macintosh OS X - Audio Units

Copy the SoundHack Spectral Shapers from the OS X folder to /Library/Audio/Plug-Ins/Components/ on your system drive. This folder may be found easily using the Finder's "Go to Folder ..." command under the GO menu

Macintosh OS 9

Each host program in OS 9 will have its own plugin folder. Copy the SoundHack Spectral Shapers from the OS 9 folder

to the “VstPlugins” folder for each host program. (It’s OK to have multiple copies of the same plugin installed under several different hosts).

Windows

Copy the SoundHack Spectral Shapers from the Windows folder to the Vstplugins folder on your hard drive. This is usually *C:\Program Files\Steinberg\Vstplugins* but could be changed by the host program, or by editing the Windows registry (look in the registry for *HKEY_LOCAL_MACHINE\SOFTWARE\VST*)

IMPORTANT: The Windows version of SoundHack Spectral Shapers require that **Quicktime version 5 or later** be installed. Quicktime is installed by downloading and installing Quicktime player from http://www.apple.com/quicktime. **If you do not install Quicktime, the SoundHack Spectral Shapers will have a blank user interface.**

3) Registration

SoundHack Spectral Shapers will operate for a 14-day demonstration period without registration. To register, start the **Spectral Shapers Unlock** program you recieved when you purchased the plugins. When the registration box appears (Fig. 3), enter your user name and registration code and click Register.

Figure 2: Registration Bar on **+binaural** before registration.

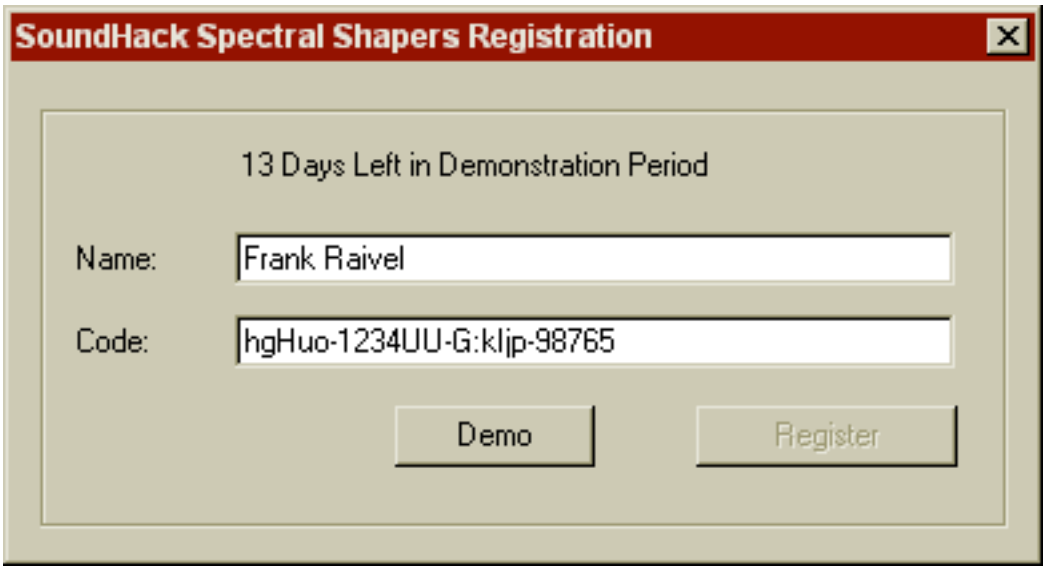


Figure 3: Registration Box (Windows)

The user name and registration code must be entered exactly as they appear on the Confirmation Email you received from SoundHack. If these are entered correctly, the registration box will disappear when you click the **Register** button, and the next time you load a plugin your name will appear at the bottom of the plugin window (Fig.4).

Figure 4: Registered user's name appears on plugin window

NOTE: SoundHack Spectral Shapers are licensed to a **single CPU only**. If you wish to license multiple CPUs, you must purchase a multiple machine license from SoundHack (http://www.soundhack.com).

4) Computer Requirements

SoundHack Spectral Shapers are CPU and memory-intensive plugins. To operate, they require at least a 500 MHz processor, and will, of course, benefit from faster speeds. If the plugins do not load, the host's memory allocation may need to be increased.

Hosts Tested

The following programs have been used extensively in the development of SoundHack Spectral Shapers, and are known to work well with them.

Macintosh OS 9:
Ableton Live
AudioEase VSTWrapper
Cycling `74 Max/MSP
Bias Deck
Bias Peak
Emagic Logic
Steinberg Cubase

Macintosh OS X:
Ableton Live
Bias Deck
Bias Peak
Cycling `74 Max/MSP
Plogue Bidule
Steinberg Cubase SX
Steinberg Nuendo
TC Electronic Spark XL
FXpansion VST to AudioUnit Adapter

Windows:

Ableton Live
AudioMulch Interactive Music Studio
Plogue Bidule
Steinberg Cubase SX
Steinberg Nuendo
Cakewalk VST Adapter

5) Controls And Displays

The following controls and displays are used in the SoundHack Spectral Shapers:

Rotary Knob

Figure 5

The rotary knob (Fig.5) changes value with straight mouse motion. If you click down and drag up or drag right, the value increases. If you click down and drag down or drag left, the value decreases. The current value of the control is displayed in the center of the knob.

PPM Meter

Figure 6

The PPM meter (Peak Programme Meter) indicates the output level of the plugin (Fig.6). This meter has an IEC standard 20 dB decay per 1.5 seconds. If you click on the PPM meter, numbers will appear to show the actual dB level.

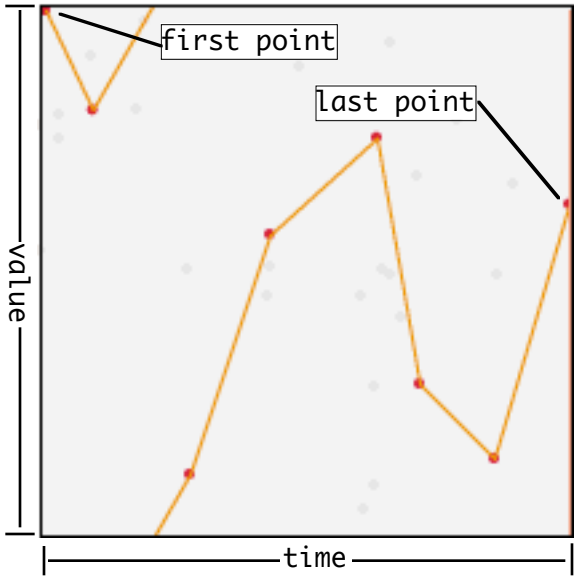
Toggle Button

Figure 7

A single click of the toggle button (Fig.7) will change the setting from on to off, or off to on. When the button is on, the small circle will have a dark outline. This button is sometimes used

to switch between different settings, such as **log/linear** or **gate/duck**. The toggle text will change to indicate the setting.

Envelope Editor



The envelope editor (Fig. 8) allows you to adjust a fixed number of points in the envelope table. When you click in the editor, the closest point will be adjusted to the clicked position. The first and last points have a fixed time at the beginning and end of the table, and can only be adjusted for value.

Figure 8: Envelope Editor

Spectral Display

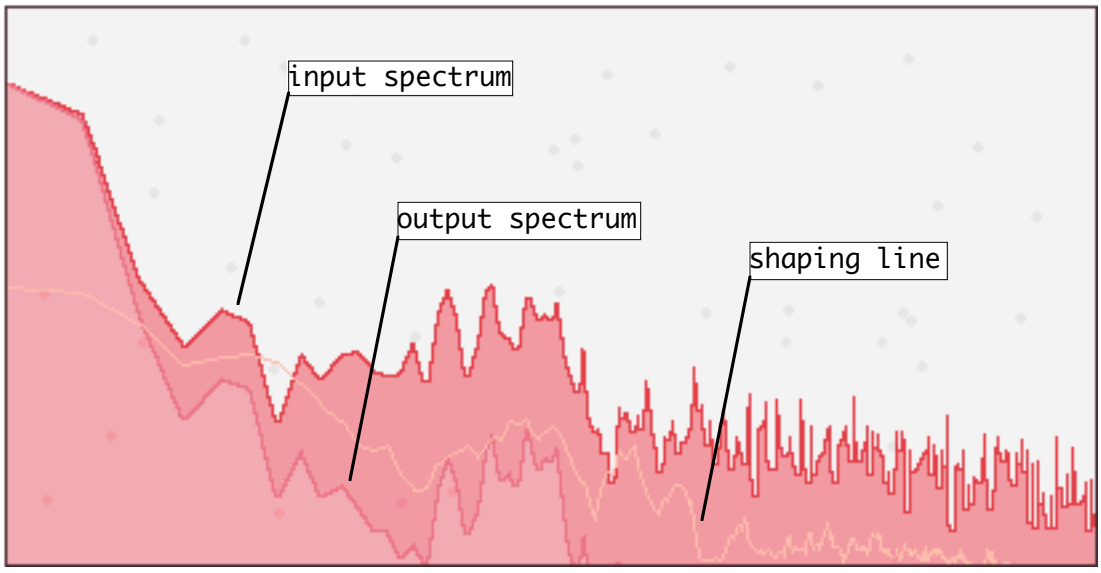


Figure 9: Spectral Display in single-channel mode

The spectral display is the most complex control in the

SoundHack Spectral Shapers. It shows the level of sound at all frequencies (the spectrum), for all channels, and for both input and output.

If the plugin is working on a single channel (mono-in, mono-out), the single channel will use the entire spectral display, with the highest amplitude (0 dB) on the top and the lowest amplitude (-96 or -120 dB) on the bottom (Fig. 9).

If the plugin is working on two channels (mono-in, stereo-out or stereo-in, stereo-out), the left channel will be shown on the top half of the display, with the highest amplitude on the top and the lowest in the middle (Fig. 10). The right channel will be shown inverted on the lower half of the display with the highest amplitude on the bottom and the lowest amplitude in the middle.

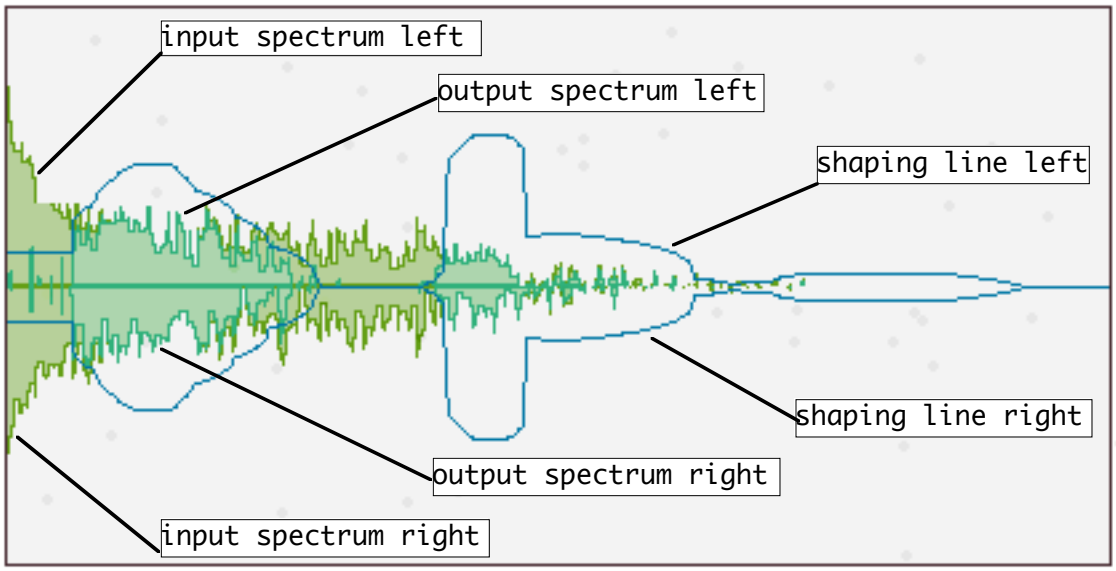


Figure 10: Spectral Display in two-channel mode

In both cases, the output spectrum is drawn on top of the input spectrum in a slightly different color.

Frequency is displayed with the lowest frequency (0 Hz) on

the left and the highest frequency (half the sample rate) on the right. In most of the Spectral Shapers, a "**linear/log**" button is provided to allow you to display the frequencies over a linear or logarithmic scale.

The spectral display will show you the precise amplitude and frequency of any point in the display if you move the mouse pointer over the display without clicking down.

In all of the Spectral Shapers there is a **shaping line** which you can edit to affect processing of separate frequencies. Edit this **shaping line** by clicking and dragging across the spectrum (see figure 6). It is useful to edit the **shaping line** while processing the sound to hear the effect of your changes. All Spectral Shapers have a "**stereolink**" button to allow you to edit two channels simultaneously.

6) +spectralgate

+spectralgate is a spectral version of the standard noise gate/ducker dynamics processor (Fig.11). It divides the frequency range into 513 bands, and applies a separate noise gate on each band. This allows you to gate or duck some bands without affecting others, thus affecting only specific frequencies. In addition, you can draw a threshold shape, setting a separate threshold for each frequency.

The combination of a noise gate/ducker and a multiband processor gives you a plugin that can used both as a frequency dependent gate or as a level dependent filter. Ducking will be more useful as a musical effect than gating, as it affects the louder harmonics (those higher than the threshold), while gating affects the quieter harmonics. I left the more extreme settings available on most of the controls so that you can explore the limits of this algorithm.

Controls

threshold – This is the point at which the gate starts to operate. It can be varied from –96 dB to 0 dB. The level of the input sound at every frequency is compared against the **threshold**. If gating, the **gain** is applied whenever the input sound is lower than the **threshold**. If ducking, the **gain** is applied whenever the input sound is higher than the **threshold**. If the **shaping line** in the spectral display is used to give the threshold a complex shape, the **threshold** knob represents the average threshold of the **shaping line**.

gain – This regulates the amount of gain applied when the **+spectralgate** is in **gate** mode and the sound is below the threshold, or in **duck** mode and the sound is above the threshold. **gain** can be varied from –60 dB to 60 dB.

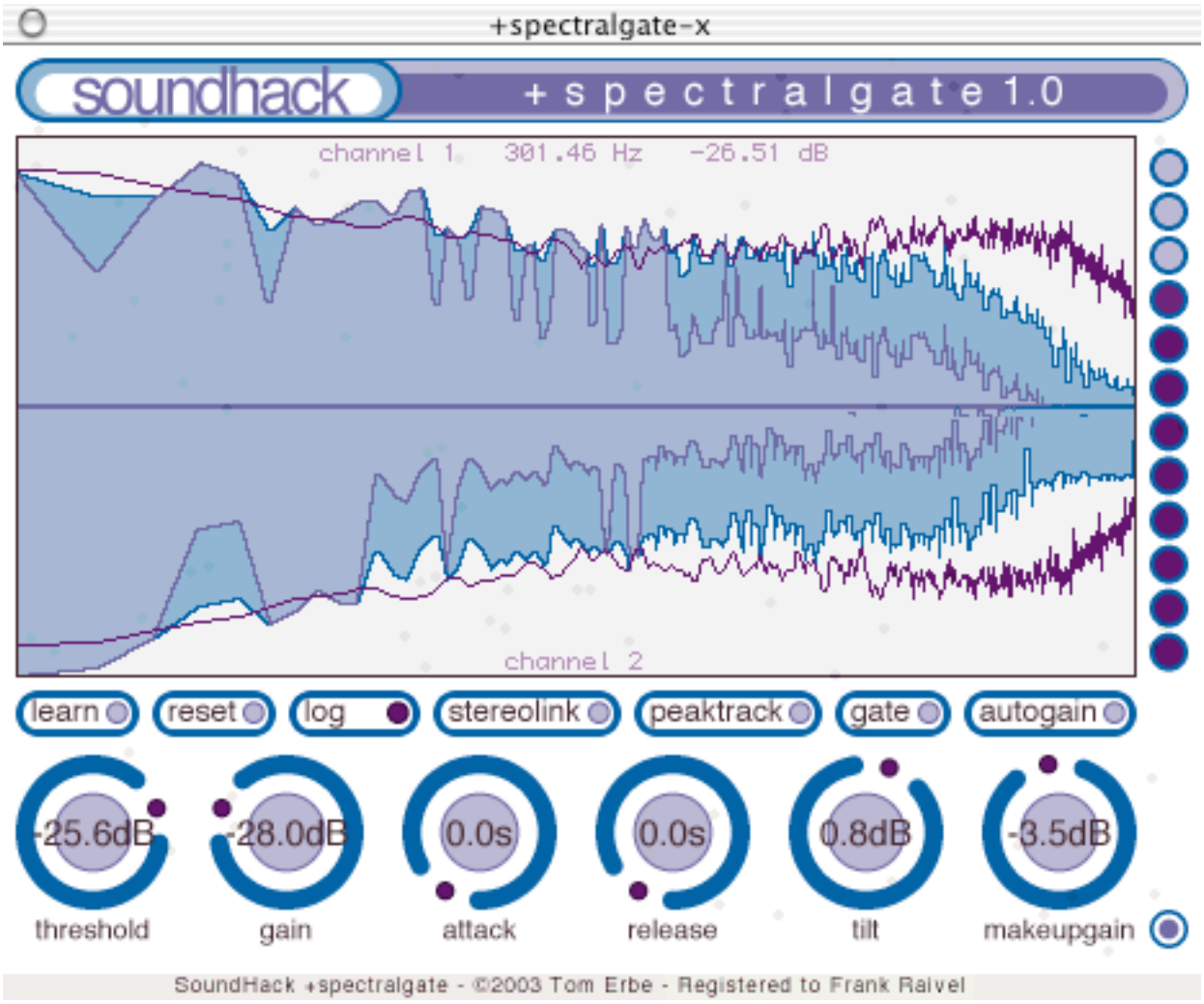


Figure 11: **+spectralgate**

attack and **release** – These controls specify how quickly the gate will turn on (**attack**) or turn off (**release**) after passing the **threshold**. They both can be varied from 0 to 1 second. A longer value will smooth the transition from ungated to gated, making the transition less

noticeable. A shorter value will make the transition more abrupt.

tilt – This control gives you an easy way to change the threshold shape, either by raising the high frequencies and lowering the low, or vice-versa. Tilt can go from a 6 dB per octave boost of the high frequencies to a –6 dB per octave cut of the high frequencies.

makeupgain – This is a simple level adjustment that affects all frequencies, whether gated or not. It is automatically adjusted as **gain** and **threshold** are adjusted to attempt to maintain a unity gain for the effect. The range is –60 dB to 60 dB.

learn – This button will take the spectral shape of the current input sound and set the threshold from it. This is useful if you are trying to gate a specific type of sound.

reset – This button resets the threshold shape to a straight line.

linear/log – This changes the display of the input and output spectra to either linear or log frequency.

stereolink – This gives you the same threshold shape for both channels.

peaktrack – This causes the threshold shape to track the peak sound. This is useful if you want to maintain the same spectral density regardless of the sound level.

gate/duck – This button switches between **gate** mode, where the gain is applied to bands in which the level is below the threshold, and **duck** mode, where the gain is

applied to bands in which the level is above the threshold.

autogain – When **autogain** is on, the **makeupgain** is set automatically, attempting to match the level of input and output signals. It is very helpful to have this on when you are playing with wide variations in settings.

MIDI Controls

The following MIDI controller to VST control mapping is used:

```
MIDI controller.....VST control
7...................makeupgain
72..................release
73..................attack
75..................threshold
76..................gain
77..................tilt
```

Suggested Uses For +spectralgate

It is hard to suggest common settings for **+spectralgate**, as the effect is very dependent on the type of sound being filtered. However, here are a couple places to start:

Harmonic reduction – In this example you can remove the low amplitude harmonics, leaving only the harmonic peaks. First, make sure the **+spectralgate** is in **gate** mode. Turn down the **gain** (to about –30 dB or less), and leave the **threshold** at around –48 dB. You will notice that only the

loudest components of the sound remain. For more removal, bring the **threshold** higher to about –24 dB. If brought lower, you can do a more subtle removal (and you can actually do some noise reduction or reverb reduction with this kind of a setting). Now adjust the **tilt** either way to favor either the high or the low frequencies. While adjusting all of these parameters, you should watch the spectral display to see the effect of what you are doing. If you put the display in **log** mode, you will find it easier to see individual harmonics. Finally, turn on **peaktrack**. This will cause the threshold to track the peak harmonic, so that the same number of harmonics are removed no matter what the overall volume is.

Peak reduction – Here you do the opposite of the previous effect, reduce the loudest spectral components. This is useful in treating recordings that have loud resonances, or to bring out some of the background of a recording. This time, put the **+spectralgate** in **duck** mode (in this mode, sounds above the **threshold** will be affected). Start out as in the previous example, with the **gain** at about –36 dB, but with the **threshold** at 0 dB. Now reduce the threshold, you will notice more and more of the harmonics dropping out. At a **threshold** of about –50 dB, you will be left with mostly background reverb. Between these two extremes there should be a lot of interesting sounds. Turning the **gain** down will result in a more subtle transformation. Using **tilt** will again emphasize either the higher or lower frequencies. At this point, you may try drawing some custom thresholds so that certain frequencies are affected more than others. Whenever adjusting these parameters, it is highly recommended that **autogain is turned on**. This will attempt to automatically adjust the **makeupgain** so that the output sound is the same volume as the input sound.

7) +spectralcompand

+spectralcompand is a spectral version of the standard expander/compressor (Fig.12). Like the **+spectralgate**, it divides the frequency spectrum into 513 bands, and processes each band individually. Each band is processed with a combination expander and compression unit (commonly known as a "compander") which can go smoothly from a compression ratio of 5:1 to an expansion ratio of 1:5.

This effect can be used in a similar fashion to **+spectralgate**. However, since expansion and compression are based on smoothly changing gain, the effect is much more gentle and subtle than **+spectralgate**. When in compression mode, it can be used to tame resonant frequencies; to diminish the fundamental; or for radical changes of timbre.

In expansion mode, **+spectralcompand** becomes a highly tunable broadband noise remover, capable of removing hiss, hum and machine noise, without damaging the original sound.

Controls

threshold – This is the point at which **+spectralcompand** turns on and off. When compressing, the effect is active when the sound is higher than the threshold. When expanding, the effect is active when the sound is lower than the threshold (this is known as "downward expansion"). The **threshold** can be varied from –120 dB to 0 dB. If the **shaping line** in the spectral display is used to give the threshold a complex shape, the **threshold** knob represents the average threshold of the **shaping line**.

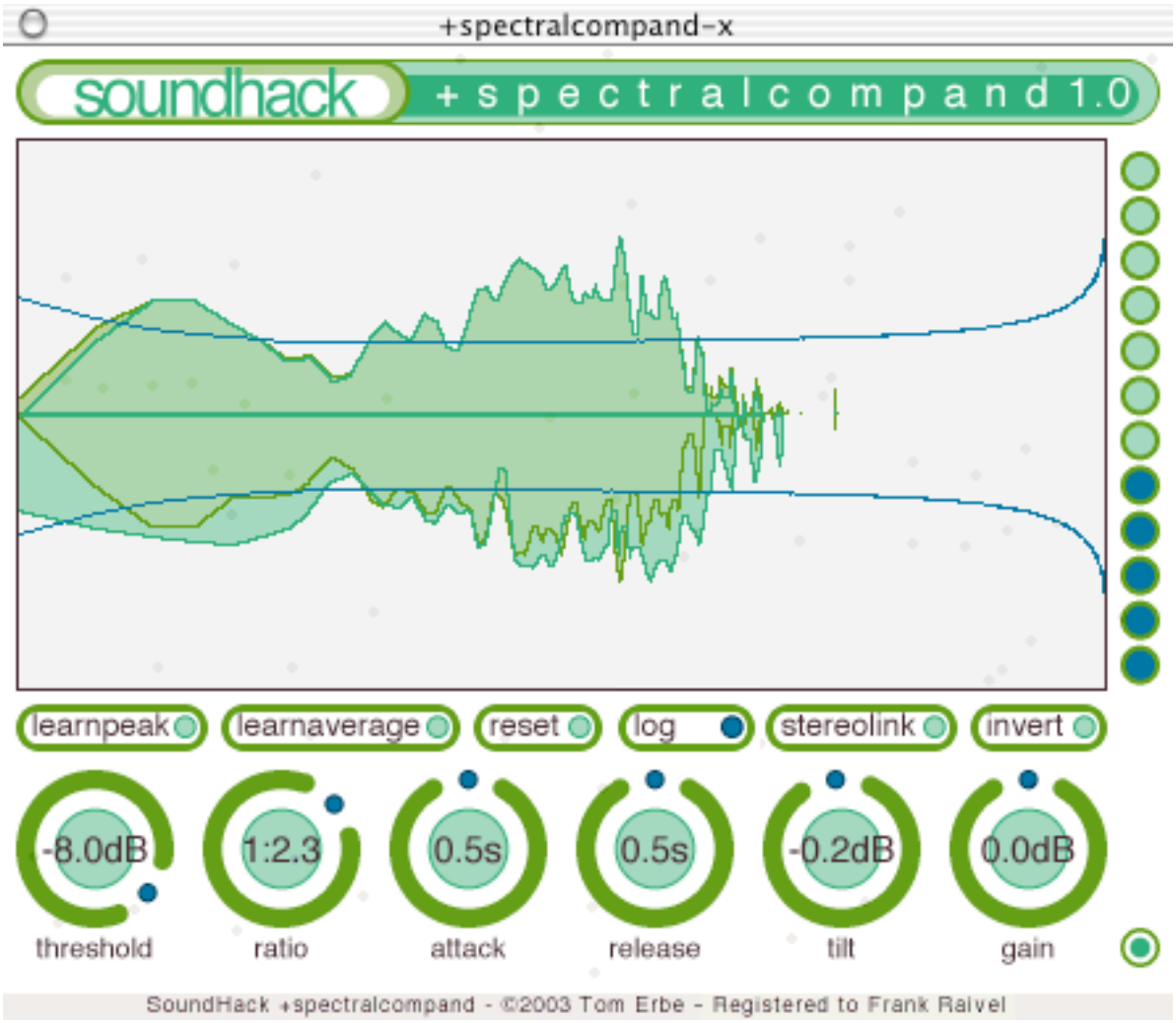


Figure 12: **+spectralcompand**

ratio – The **ratio** controls the amount of compression or expansion. The range from 5:1 to 1:1 is the compression range, and from 1:1 to 1:5 is the expansion range. Compression reduces the dynamic range of the sound, resulting in a smaller difference between the loudest and softest harmonic. Compression affects only those harmonics which are above the threshold. Conversely, expansion increases the dynamic range of the sound, resulting in a larger difference between the loudest and softest harmonics. Expansion only affects those harmonics which are below the threshold.

attack and **release** – These controls specify how quickly the gate will turn on (**attack**) or turn off (**release**) after passing the **threshold**. They both can be varied from 0 to 1 second. A longer value will smooth the transition, making the transition less noticeable. A shorter value will make the transition more abrupt.

tilt – This control gives you an easy way to change the threshold shape, either by raising the high frequencies and lowering the low, or vice-versa. Tilt can go from a 6 dB per octave boost of the high frequencies to a –6 dB per octave cut of the high frequencies.

makeupgain – This is a simple level adjustment that affects all frequencies. The range is –60 dB to 60 dB.

learnpeak – This button will set the threshold shape, using the largest (peak) sound between the current input sound and the previous threshold shape for each frequency.

learnaverage – This button will set the threshold shape, using the average of the current input sound and the previous threshold shape for each frequency.

reset – This button resets the threshold shape to a straight line.

linear/log – This changes the display of the input and output spectra and the threshold shape, to linear frequency or to log frequency.

stereolink – This gives you the same threshold shape for both channels.

invert – This control allows you to invert the

compression or expansion process and hear the difference between the processed sound and the input. This is very useful when using spectral expansion for **broadband noise reduction**, as you will be able to hear the sound that is being removed.

MIDI Controls

MIDI controller.....VST control
7.....gain
72.....release
73.....attack
75.....threshold
76.....ratio
77.....tilt

Suggested Uses For +spectralcompand

The expansion and compression modes in **+spectralcompand** have such different sounds, that it is like having two separate effects. The expansion mode is most useful as a **broadband noise remover**, able to flexibly remove hum or hiss. The compression mode can be used as a **spectral flattener** or **reshaper**, changing the spectral profile of a soundfile to another template.

Broadband noise remover – To separate and remove steady spectrum noise (e.g. hiss or hum) from a sound, you first need to play a portion of the noise alone. When this noise is playing, click either **learnpeak** or **learnaverage**. This will set the threshold to the level of noise for each frequency. Now set the **ratio** to 1:2, and raise the **threshold** until you hear the noise disappear. At this point, it is useful to click **invert** to hear what you are removing. Increasing the **ratio** will give you a more

abrupt noise reduction. To better preserve transients in the original sound, change the **attack** and **release** to match the typical attack and release of the sound being affected. Usually an **attack** of around 0.0 and a **release** between 0.3 and 0.5 will sound most natural. Often you will need to perform more noise reduction on hum (50 – 60 Hz) than other frequencies. If you do, put the spectral display in **log** mode and try to draw a higher threshold for these low (leftmost on the window) frequencies. Clicking **learnpeak** or **learnaverage** again will add to the current noise threshold profile.

Spectral flattener and reshaper – Spectral compression can be used to flatten the spectrum, that is, to make all harmonics the same volume. First, set the **ratio** around 2:1 to 4:1 and lower the **threshold** slowly. Louder harmonics will be affected first, automatically lowered in volume to match the **threshold** amplitude. The **gain** may need to be increased to make up for the reduction in overall sound level. Lowering the **attack** and **release** will cause the sound to more quickly drop to the level of the **threshold**. Excessive flattening will give you nothing but noise, as all the harmonics will drop to the same level. In addition, if you use the **learn** buttons to grab a spectral shape, or redraw the shape by hand, you can reshape the sound to this learned shape.

8) +morphfilter

+morphfilter is a basic filter in which you can draw or learn a filter shape (Fig.13). Each **+morphfilter** preset contains two filter shapes which you can morph between. Filter depth can be applied to increase, decrease or even invert the filter shape. This is an easily controllable, yet complex, filter plugin.

Controls

spectral display – To use the **+morphfilter**, you must first draw a filter shape in the **spectral display**. This curve will be used by the **filter depth** setting to create the actual filter.

filter depth – This changes the depth or strength of the filter. At 0.0 you get no filtering (a flat line). As you proceed to 1.0, you get closer to the filter specified by the filter shape. When you put this control in the negative region (0.0 to –2.0), you get an inversion of the filter shape. This control goes from –2.0 to 2.0. While changing **filter depth**, it is good to have **autogain** switched on, as certain filter depths can have much more gain than others.

filter number – This control lets you morph between the two stored filter shapes, filter 0 and filter 1. Changing the **filter number** will cause a smooth fade between the shapes. If you are drawing or learning a new shape, you will set filter 0 when the control is below 0.5 and you will set filter 1 when the control is above 0.5.

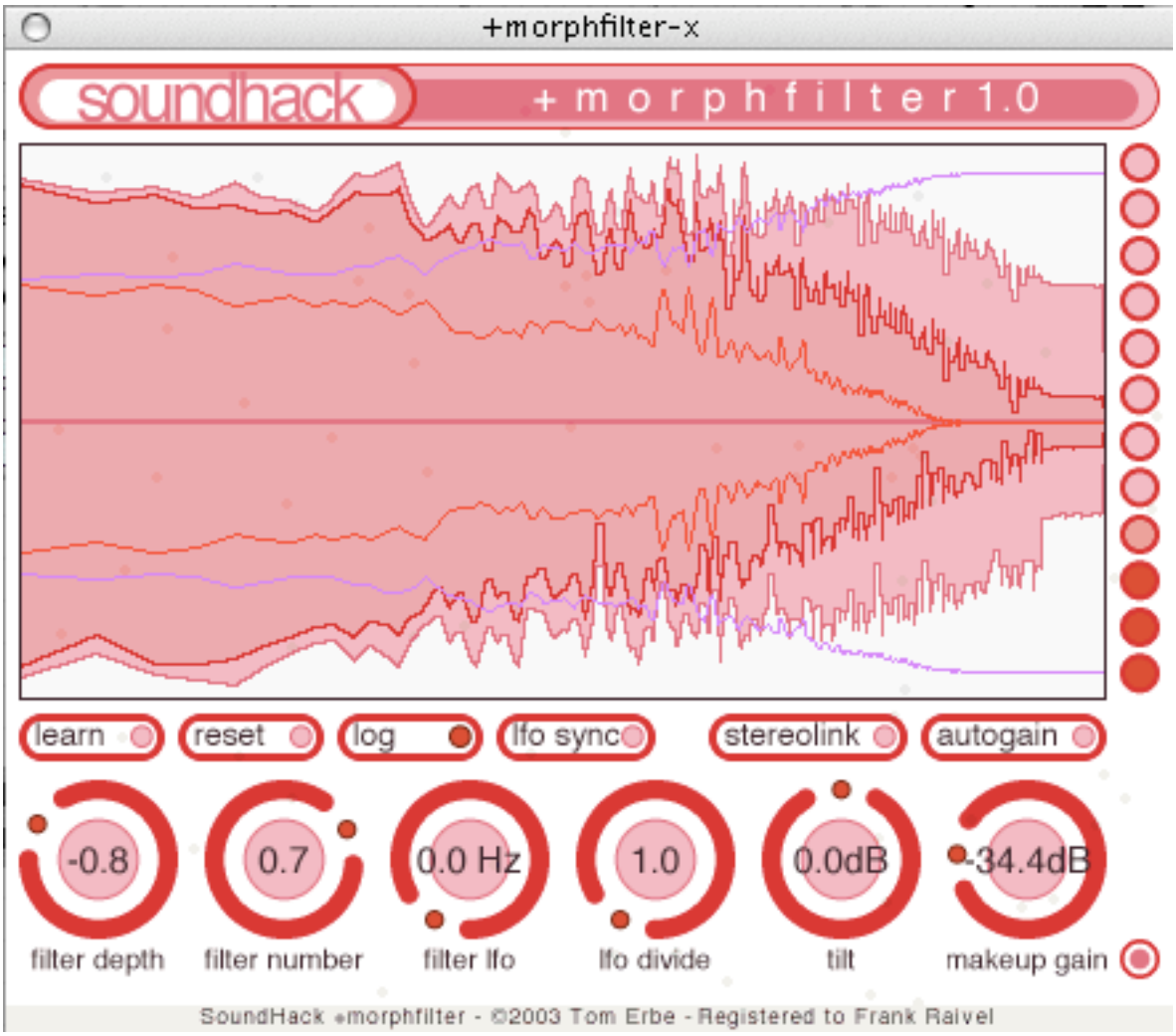


Figure 13: **+morphfilter**

filter lfo – You can modulate the **filter number** with a low frequency oscillator (LFO) by setting this control to something other than 0.0 Hz. This will cause the filter to morph back and forth between filter 0 and filter 1. This control goes from 0.0 to 4.0 Hz.

lfo divide – This allows you to quickly change the LFO frequency by dividing the LFO by the number set on the control. The range of **lfo divide** is 1.0 to 11.0.

tilt – This tilts the frequency response to boost highs

and cut lows, or boost lows and cut highs. The range of **tilt** is from –3.0 dB/octave to +3.0 dB/octave.

makeupgain – A gain control to compensate for the natural gain or loss in a given filter shape. It is better to have this control automatically controlled by the **autogain** function if you are dynamically changing the filter number and depth.

learn – This will capture the average of the current sound's frequency spectrum. Unlike the **learn** buttons in both **+spectralcompand** and **+spectralgate**, this button is not momentary, but remains on, adding more sound to the spectral average, until it is clicked a second time.

reset – This clears the current filter shape, leaving a flat 0dB line.

linear/log – This changes the display of the input and output spectra and the filter shape, to either **linear** or **log** frequency.

lfo sync – This button will synchronize the LFO to the host application's MIDI clock (if there is one). If there is no MIDI clock, this button won't do anything.

stereolink – This button gives you the same filter shape for both channels.

autogain – When **autogain** is on, the **makeupgain** is set automatically, attempting to match the level of input and output signals. It is very helpful to have this on with **+morphfilter**, as different filter shapes have widely varying gain characteristics.

MIDI Controls

MIDI controller.....VST control
7.....makeupgain
75.....depth
76.....tilt
77.....filter number
78.....lfo speed

Suggested Uses For +morphfilter

The application of **+morphfilter** is straightforward. First, draw a filter shape to boost or cut frequencies, then adjust **depth** to get the desired strength of filtering and use **tilt** to boost the low or high end. Once your filter is set, adjust the **filter number** to 1, define a second filter, and morph between the two filter shapes with the LFO or by manually adjusting **filter number**.

Inverse filtering – **+morphfilter**'s ability to learn a filter shape from the incoming sound, combined with the inverse range of the **depth** control, gives you a unique method for removing a frequency response from a sound.

In this first example, we will try decreasing the prominent timbral characteristic in a sound. Make sure **autogain** is switched on while you are adjusting the filter shape and **depth**. Click **learn** while the sound to be removed is playing and click again to stop learning. Now slowly turn **depth** from 0.0 to –1.0. You should hear the spectra of the sound flatten out as you adjust the depth.

In this second example, we will try to remove a room characteristic from a sound. To do this you will need to record white noise in the room in which the recording was made. Now, while playing the white noise, train the filter again by clicking **learn**. Make sure to take a large sample. Click **learn** again to finish taking the sample. This sampled filter shape will now contain the resonances of the room. Now, play the sound recorded in this room through this newly-created room filter. When you adjust the **depth** down toward -1.0, you will hear the room resonances diminish.

9) +binaural

+binaural is a filter which places a sound at a specific position around the listener's head (Fig.14). It does this by using filters which simulate the filtering effect of the head and outer ear for sounds at all angles. The filters in **+binaural** are optimized for headphones and are only for 0 degree elevation at a 44100 sample rate. Future versions of **+binaural** will contain speaker corrections and multiple elevations and sample rates.

Controls

angle – This is where you set the position of the sound. The angle can be varied from –180 degrees to 180 degrees. 0 degrees is straight ahead, 90 is to the right, –90 is to the left, 180 and –180 are both directly behind the listener.

gain – A simple volume control after the binaural processing. Occasionally a sound will resonate with the binaural filter, and it will be necessary to lower the gain. This control goes from –12 to 12 dB.

filter one/filter two – These buttons allow you to select between two binaural filter sets. **filter one** is derived from Bill Gardner and Keith Martin's measurements of the KEMAR dummy head microphone at the MIT Media Lab (http://sound.media.mit.edu/KEMAR.html). The diffuse-field equalized HRTFs are used in this plugin. **filter two** is the original SoundHack binaural filter developed by Dr. Durand Begault.

lfo sync – This button will synchronize the LFO to the

host application's MIDI clock (if there is one). If there is no MIDI clock, this button won't do anything.

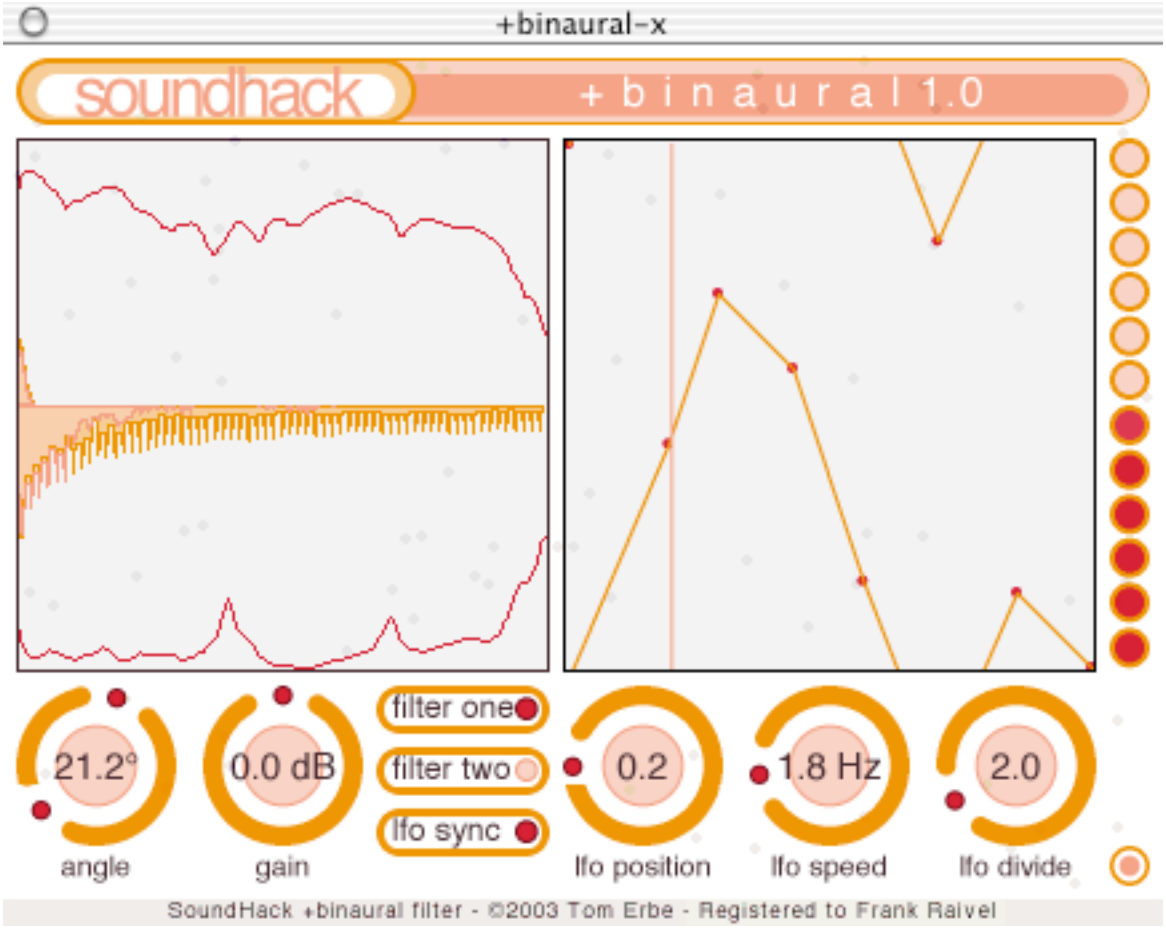


Figure 14: **+binaural**

lfo position – You can set the phase or position in the lfo envelope with this control. You will see the position line move in the envelope editor and the corresponding angle change when moving this control.

lfo speed – This sets the speed of the LFO from –1 Hz to 1 Hz. A negative speed causes a reverse rotation through the LFO envelope. If **lfo sync** is on, the **lfo speed** is set by the MIDI clock.

lfo divide – This control slows down the **lfo speed** by

dividing it by numbers from 1 to 11. This gives you an extremely slow LFO if desired.

envelope editor – This allows you to change the shape of the lfo envelope, giving you custom binaural trajectories.

MIDI Controls

MIDI controller.....VST control
7.....gain
10.....angle
12.....lfo speed
13.....lfo position

Suggested Uses for +binaural

+binaural will most often be used when a more accurate pan control with front and rear depth is needed. It can be used along with a reverb (to simulate distance) to recreate a sound stage or other virtual environment. When used with **lfo sync**, **+binaural** can be used to place various beats or parts of a loop in specific repeatable positions.

10) Acknowledgments

I'd like to thank the many people who have helped me on this project, but primarily Betsy Edwards, my wife and the editor of this manual, for her continual support, feedback, and for keeping things on course. Thanks to Sean Rooney for collaborating on the first version of +binaural; to Eloy Anzola for his web design skills; to Akira Rabelais for endless help in graphic design; and to Durand Begault for the HRTFs and drinks at Plouf. I'd also like to thank Phil Burk, Sara Roberts, Mark Trayle, Marc Porier, Ryan Francesconi, Stephan Mathieu and Joel Fox for all their help with both programming and graphic design. Thanks to CRCA at UCSD (Carole Hobson, Ted Apel, and Miller Puckette) and CEAIT at CalArts for hosting me during development. Thanks to my beta testers for putting Spectral Shapers through their paces: Simon Rogers, Steve Barry, Ryan Ball, Rick Brown, Clay Chaplin, Casey Rice, Jef Bear, Jef Lear, Adrian Moore, Andy Crosby, J Lesser, Jason Kahn, Marc Van Bork, Marshall Simmons, Michael Theodore and Nick Wallace. And finally, I'd like to thank the following people and companies for their support: Angus F. Hewlett, FXpansion; Christian Kleine, Ableton; Karl Steinberg, Steinberg; Sven Duwenhorst, TCWorks; Steve Berkley, BIAS; Ross Bencina, AudioMulch; and Sascha Kujawa, Emagic.