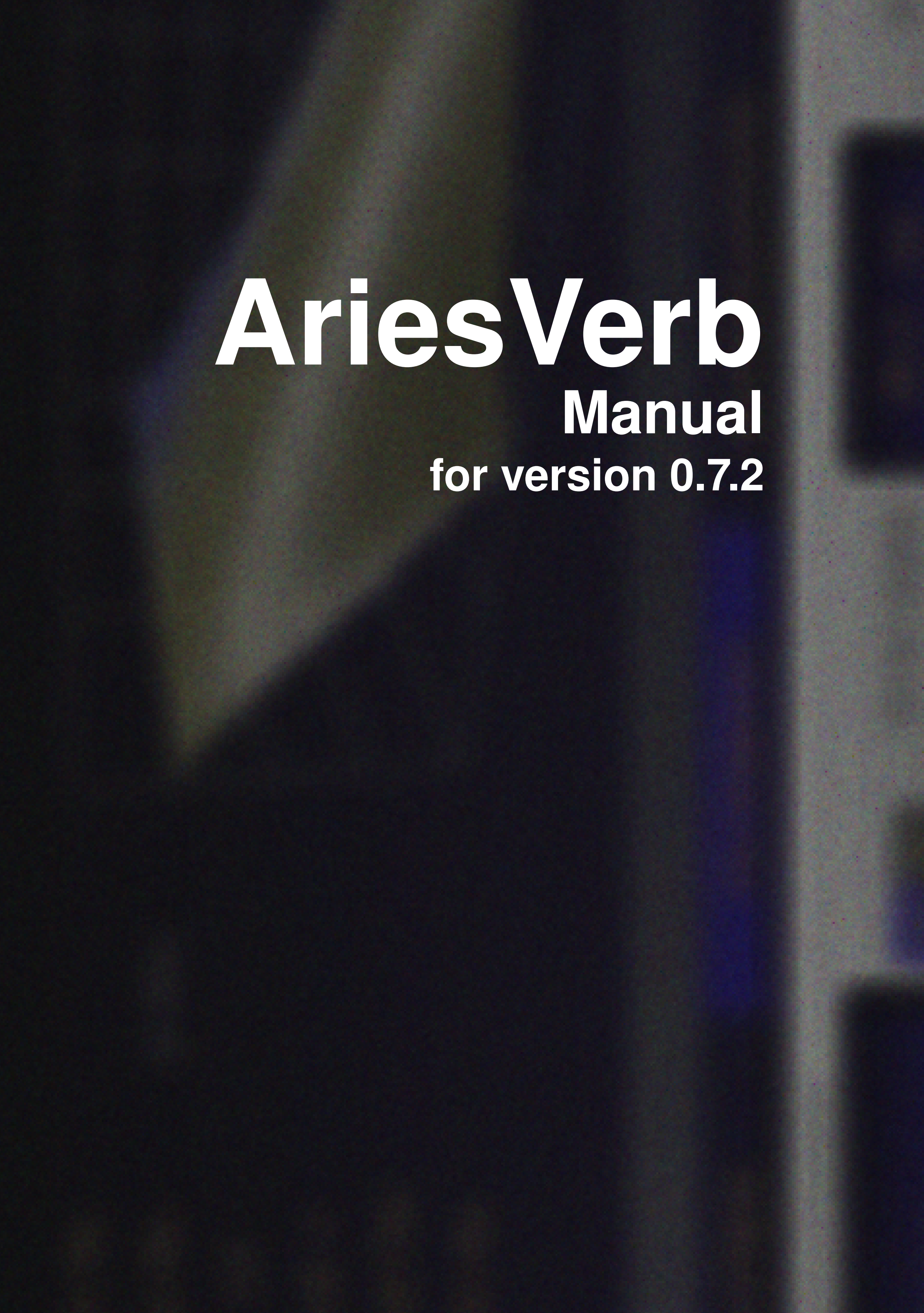


AriesVerb

Manual

for version 0.7.2

ARIESVERB Manual
for version 0.7.2
July 1, 2009

http://www.ariescode.com

License Agreement

Commercial license for AriesVerb 0.7.2 (the “Software”) between you (the “Licensee”) and the author, Christian Schüler.

You agree to be bound by this license by installing the Software. Within a 30 day period from the date of purchase, you may revoke the agreement for any reason and receive a refund, but in doing so you must uninstall the Software and delete all copies you made.

The software is licensed for use by a single natural person. You are allowed to make copies of this Software, including multiple installations and archival, provided that only you as the Licensee uses said copies. You are not allowed to redistribute the Software, including but not limited to, making the Software available for download, or making the Software part of a CD compilation. You are not allowed to resell or rent your license without permission.

You are allowed to use this software, for any artistic, scientific or commercial application, including music production.

This software is provided “as-is”, without any expressed or implied warranty. In no event will the author be held liable for damages arising from the use of the Software, to the maximum extent permitted by law. This applies to any use of the Software, whether the use was made in accordance with the license agreement or not.

Preface

ARIESVERB is a versatile sound effect processor—it can do reverb, delay, echo, flanger, chorus, phaser, comb filters, body resonances, pitch shifting and much more. It can also do hybrid effects by blending any of the above. This is possible because ARIESVERB in essence a pretty interface around a fully general Feedback Delay Network, which is a very generic algorithm able to unify all of said effects into one framework.

The development of ARIESVERB started in the early 2000's when the author was courious for a reverb algorithm that would be good for real time use in a game engine. Low CPU consumption would be the first design rule, as the processor time is very constrained when running a game, even more so back then at the beginning of the century. Feedback Delay Networks looked like a promising idea.

Soon it became apparent that an FDN has a much broader scope of application than just reverb. In order to capitalize on this, one has to think outside of the box and allow unusual parameter domains spanning several orders of magnitude. Delay lengths down to microseconds (µs) for instance, with good interpolation, as this would enable the phase effects of very short delays like in flangers. Or modulation rates up to the kHz range, as this would allow for FM-like effects. Also, the feedback matrix (a core element of any FDN) must not be hard wired, but freely adjustable.

Putting all of this into code that is still light on the CPU was a challenging excercise. Another challenge was the design of the human interaction and factorizing the right aspects into a user interface. ARIESVERB is the result of all this, delevered to you in form of an excellent VST plugin. Enjoy!

Contents

Operation

Download and Installation

Visit the download link that you have received via e-mail. Enter your license key into the online form and you should see a welcome message with your name and a list of download options.

Click to download the most recent version of ARIESVERB, which at this time should be at least version 0.7.2. It sould be found at the top of the list. When your browser asks you what to do, choose *save* (and if it asks you where, you should save it right onto your desktop).

After the download has finished, extract it's contents. The download is delivered as compressed archive ("zip file") and you should see an option to extract when right-clicking on it. After extraction is complete you should have these files:

presets
Ariesverb
readme

Open the file *readme* and look what it says about the version you downloaded. It shoud indeed be version 0.7.2 or higher. If everything went ok, move all extracted files to a location where your VST host application can find them. If you don't know where this is, you need to look this up in the documentation of your host. Alternatively you may try to open the hosts settings menu and look there, as most hosts allow you to configure plugin folders.

Running ARIESVERB

ARIESVERB is run as a VST plugin inside a host application, for instance by selecting it into a channel as an insert, or as a send. The first time ARIESVERB is run, the screen will look like shown below and ask you for your license key:

Click on the shaded input area on the right hand side, and type in your key or paste it (via `Ctrl-V`). This key is saved into the Windows registry and needs to be re-entered every time the information in the Windows registry is lost, for instance, after a re-install.

Upon success, you should now see the Front Panel of ARIESVERB. The Front Panel is a collection of the most fundamental controls and has switches to open the editor pages or the program library. Probably you want to check out some of the preset programs shipping with ARIESVERB now. Just go ahead, press the `P` button and browse the library. Refer to the User Interface chapter if anything on screen remains unclear. If on the other hand you want to learn more about the internals of ARIESVERB, read on.

Block Diagram

The overall structure of ARIESVERB is similar to that of a comb filter, which is simply a delay line with a feedback loop. In ARIESVERB however, all elements are multiplied by four, and a 4×4 matrix sits in the place of a simple feedback gain. A block diagram of this structure is shown in figure 1, with the user interface location corresponding to each element listed in table 1.

At the core of the system is the feedback loop with four delay lines ($z_1 \ldots z_4$) and a feedback matrix (A). Each delay line has an adjustable length and can feed back into each other delay line, via the feedback matrix. The sound signal circulates inside this loop, attenuated only by a combination of linear and nonlinear filters (F).

The input signal (x) enters from the left and is picked up by the input vectors (b). Each delay line is associated with an input vector listening along a specific direction in stereo space. In the same sense, the output vectors (c) assemble an output signal (y) by placing the signal from each delay line into a specific direction in stereo space. Both input- and output vectors are freely adjustable.

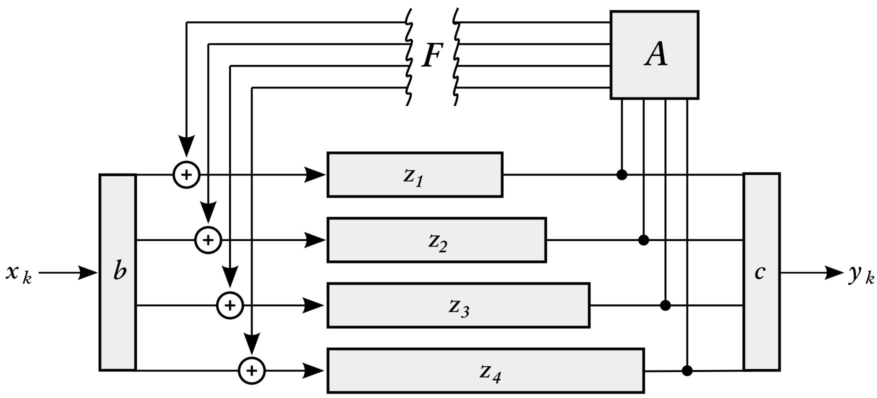


Figure 1: Block diagram of the signal flow inside AriesVerb. Feedback matrix (A), input vectors (b), output vectors (c), input signal (x), output signal (y), delay lines (z_i), scalar feedback gain, filters and saturation (F).

Element	User interface location
Delay lines ($z_1 \ldots z_4$)	Algorithm page and Front Panel (time base)
Filters (F)	Filtering page and Front Panel (half life)
Feedback matrix (A)	Matrix page
Input vectors (b)	In/Out page
Output vectors (c)	In/Out page

Table 1: Relationship between algorithm elements and their user interface location.

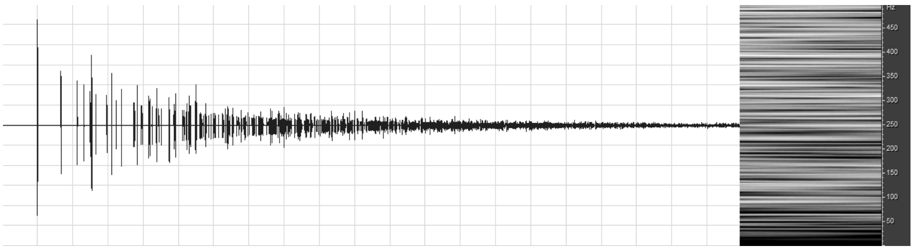


Figure 2: Impulse response of an ideal rectangular room, in time domain (left) and frequency domain (right).

Echo– and Mode Density

There are two measures of importance in a reverb application, namely echo density and mode density.

Echo density is the number of distinct echos per time interval, while mode density is the number of resonant peaks per frequency interval. For example, see the impulse response of an ideal rectangular room in figure 2. The echo density increases with time, while the mode density increases with frequency.

With the correct choices of delay lengths and feedback matrix, ARIESVERB can build a similar high echo density over time. This is evident in the recursive structure of the algorithm: The first generation spawns 4 echos, the second generation spawns 16 echos, the third generation 64, then 256, and so on.

There is however a conflict between echo density and mode density, as one of them always decreases when trying to raise the other. If, for instance, the mode density is too low, a reverb will sound metallic. Making the delay times longer (via the time base control) will fix this, but then the echo density may not be sufficient, resulting in flutter and graininess.

Multitap Mode

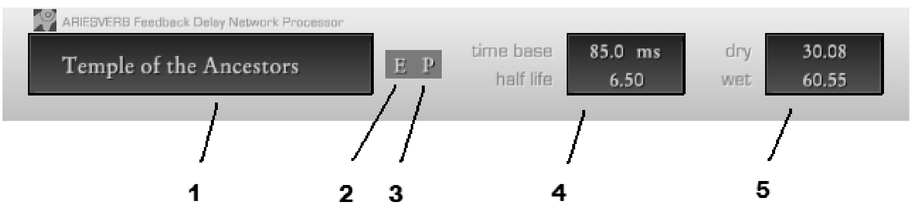
ARIESVERB offers a mechanism to solve the conflict between echo- and mode density, called multitap mode. It can provide sufficient density for the simulation of natural environments, while retaining the CPU efficiency of the original algorithm.

The multitap mode is activated by selecting this option in the configuration section of the Algorithm Page. In this case, each delay line is broken up into four parts with additional connections between the branch points. This allows for "cross mixing", a connection between corresponding taps on different delay lines, and "inner feedback", a connection between different taps on the same delay line. If both are enabled, ARIESVERB is essentially a 16×16 FDN system, comparable in quality to hardware reverb units of the early 1990's.

User Interface

Front Panel

The front panel is the part of the user interface that is always visible. From here you can open the editor, or the program library. Here are also the four fundamental parameters: time base, half life, dry- and wet mix.



1. Current program name.

2. Editor Button: Opens the program editor.

3. Program Library Button: Opens the program library.

4. Time Control Group:

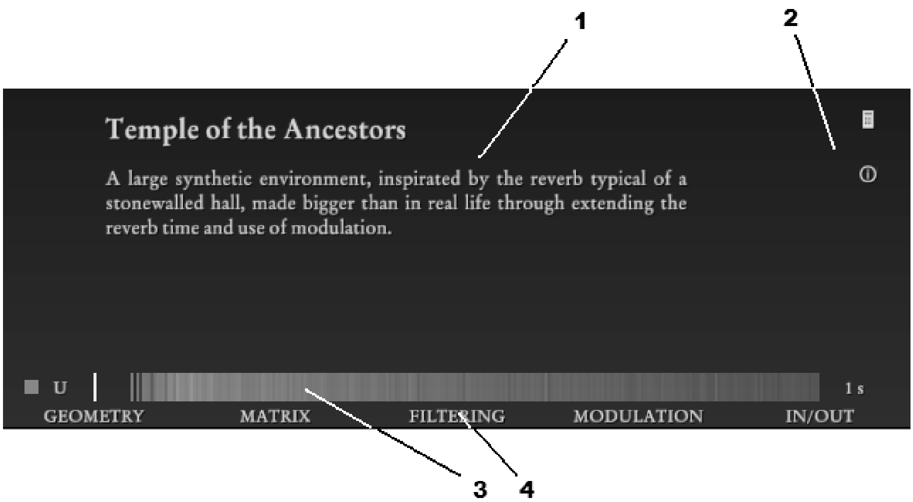
    - The time base is the basic unit of time for the current program, relative to which all other measurements are made. It can be specfied in three different units: seconds, Hertz and quarter notes. The time base affects all delay lengths and decay times. (See also the Algorithm Page).

- The half life is the time it takes for the amplitude of the reverb or echo tail to decay to a half (–6 dB). The half life is specified as a multiplier of the time base.

5. Mix Control Group: The amount of dry (original) and wet (processed) signal in the output mix, in percent.

Editor Window

The main editor window displays general information about the current program. It also functions as hub for the individual editor- and helper pages.



1. Program name and description: Both can be edited by clicking into the text.

2. Helper menu: The helper menu is always visible. It has a calculator symbol and an information symbol. Clicking on one of the symbols from anywhere in the user interface leads to the calculator or the system information. Clicking on the active symbol again exits that page.

3. Echogram: The echogram is always visible. It shows the instananeous impulse response of the current program, color coded to stereo phase and panorama. If the current program is time dependent due to modulation, the echogram is animating. To the left of the echogram is a switch to select one of a center (C), left (L), right (R), side (S) or uncorrelated (U) unit impulse. To the right of the echogram is a selector for the time scale.

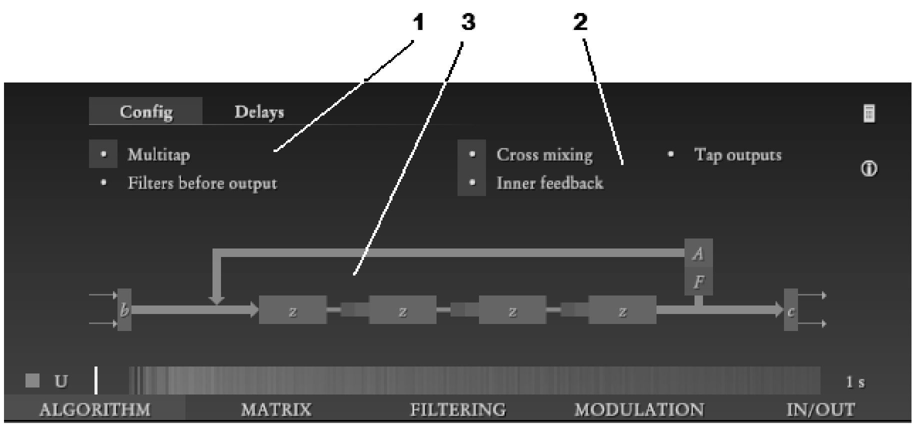
4. Editor menu: This menu is always visible. Clicking on one of the items leads to one of the available editor pages, as described in the following sections. Clicking on the active item returns you to the description page.

Algorithm Page

The overall algorithm of an AriesVerb program is edited in algorithm page. The algorithm page is divided into a config and a delays section.

Config Section

The config section displays a schematic of the current program configuration and has a number of switches to select between different configurations. Both global- and inner feedback are disabled if half life is set to zero.



1. Global options:

   - Multitap: Each delay line is subdivided into four sections, with branch points for matrix connections in between. Using this option increases the CPU load of ARIESVERB, but allows for the echo density required for most reverb applications.
   - Filters before output: The filters are placed before output is taken. This option influences the shape of the impulse response envelope. It also places non-linear saturation into the direct path from input to output.

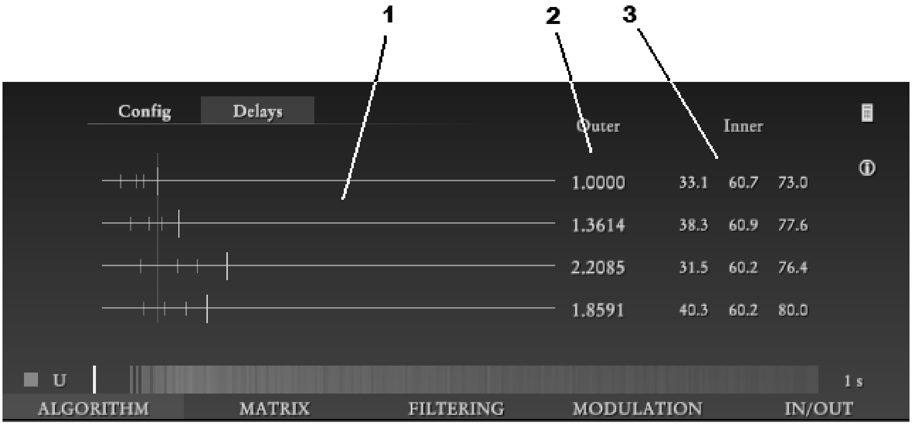
2. Multitap options: This is a collection of options only available if multitap is enabled.

   - Cross mixing: Conntects corresponding branch points of different delay lines via a matrix.
   - Inner feedback: Connects all branch points of the same delay line and the delay line entry point via a matrix.
   - Tap outputs: If enabled, all branch points are routed to the output as additional delay line taps. The per-segment amounts are adjusted via controls that appear on top of each branch point in the schematic.

3. Schematic: A schematic display of the current configuration, taking into account all options. Legend: matrix (A), input vector (b), output vector (c), delay line (z), filters and saturation (F).

Delays Section

The delays section has controls to adjust the lengths of the main delay lines ($z_1 \ldots z_4$), as well as the positions of branch points.



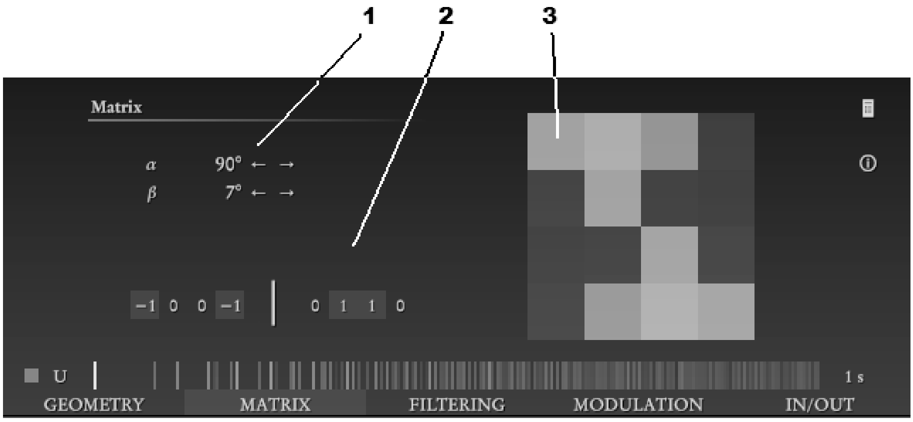
1. Delay length sliders: Graphical controls for each delay length in multiples of the time base (see Front Panel), from zero to a maximum of 8. The length of the time base is indicated by faint vertical line. The location of branch points is represented by a small green ticks if multitap mode is enabled.

2. Delay length controls: Numerical controls corresponding to the sliders.

3. Branch point controls: If multitap mode is enabled, these control the location of branch points as percentange of the total length.

Matrix Page

The matrix page has controls for the feedback matrix and optionally an inner feedback matrix. Matrices are controlled indirectly via rotation angles and a common plane of rotation. See also appendix “Rotation Matrices” for an explanation of this topic.



1. Rotation angle control: Numeric control for two rotation angles, alpha (α) and beta (β). These values control the amount of rotation that happens parallel and anti-parallel to the rotation plane, and thus, the intermixing between delay lines. If both angles are zero, the matrix reduces to an identity (non-operation).

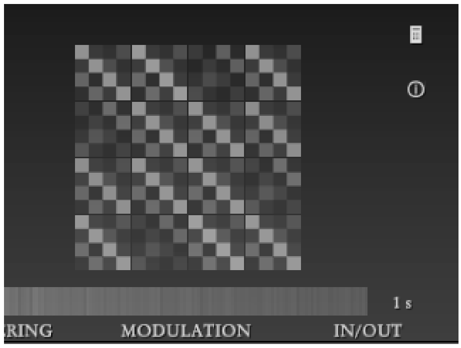
2. Rotation plane control: Numeric control for two axis vectors that span the plane of rotation. The possible values for each element are zero, one, and minus one, and the vectors are constrained to be orthogonal. If the same element is zero in both vectors, the corresponding delay line does not take part in a rotation parallel to that plane.

3. Matrix visualization: A visual representation of the resulting feedback matrix. The cell representing feedback from delay i to delay line j is found in row j, column i, as in the following layout:

$$\begin{array}{cccc} 1 \to 1 & 2 \to 1 & 3 \to 1 & 4 \to 1 \\ 1 \to 2 & 2 \to 2 & 3 \to 2 & 4 \to 2 \\ 1 \to 3 & 2 \to 3 & 3 \to 3 & 4 \to 3 \\ 1 \to 4 & 2 \to 4 & 3 \to 4 & 4 \to 4 \end{array}.$$

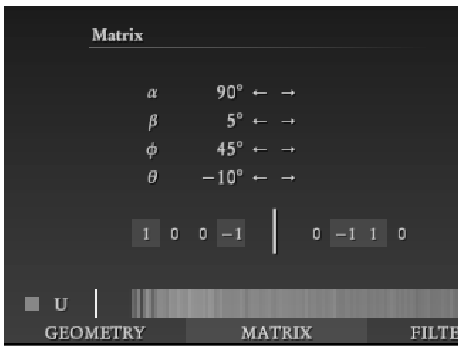
   Positive feedback is shown as white, and negative feedback is shown as red.

Multitap mode



The matrix display switches to a nested arrangement when multitap mode is enabled. In this mode there are 3 additional rows and 3 additional columns for each delay line, representing the 3 additional branch points. Thus, the cell representing feedback from delay line i, branch p to delay line j, branch q is found in row $4j+q-4$, column $4i+p-4$. The "first" branch is counted as the delay line entry point.

Inner feedback



Additional rotation angles phi (ϕ) and theta (θ) appear when inner feedback is enabled. These angles have a similar function for the inner feedback matrix as alpha and beta have for the global feedback matrix. Setting both phi and theta to zero has the same effect as disabling inner feedback altogether.

The inner feedback rotation controls the amount of intermixing between branch points on the same delay line, and can affect the density character, the attack character of the echo buildup and lead to metallic ringing.

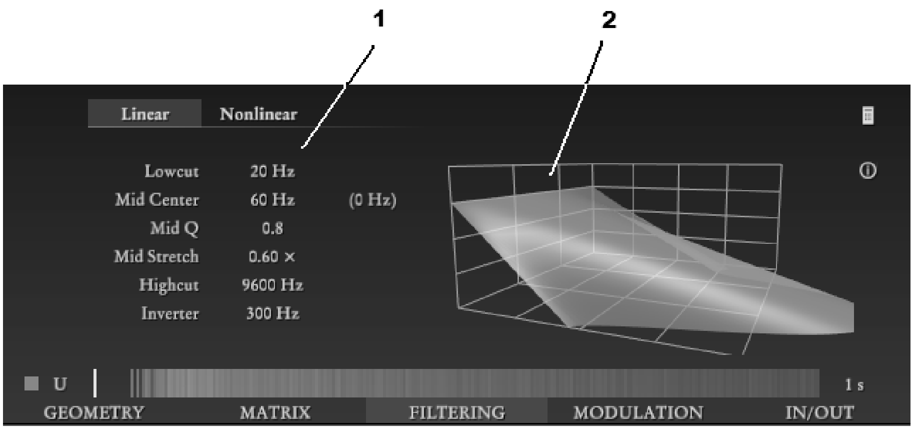
Filtering Page

The filtering page has controls for the filters (F, see block diagram in figure 1) inside the feedback loop. It is divided into a linear and a nonlinear section, with a menu on the top of the page to switch between sections.

Linear filtering section

The linear filtering section has controls for three linear filters, a 1st order low-cut filter, a 2nd order peaking filter, a 1st order high-cut filter and a frequency dependent inverter.

All filters have 6 dB per octave slopes, but depending on the current half life setting, the filters can translate into dramatic frequency dependent behaviour. See also appendix "Filters Inside a Feedback Loop" for an explanation of this topic.



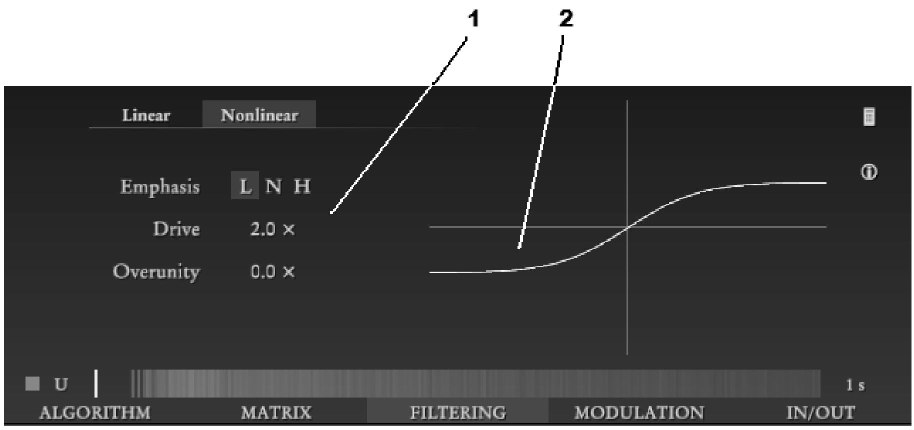
1. Filter controls:
    - Lowcut: Frequency at which the half life time is reduced to 50% by the lowcut filter.
    - Mid Center: Center frequency of the peaking filter. In parentheses, the center frequency of the resulting peak, taking into account all frequency dependent behaviour.

- Mid Q: The quality factor (bandwidth) of the peaking filter.
- Mid Stretch: The half life time factor for the peaking filter at its own center frequency.
- Highcut: Frequency at which the half life time is reduced to 50% by the highcut filter.
- Inverter: Frequency above which the phase is inverted (180°).

2. Waterfall diagram: This is a 3D visualization of frequency dependent decay, having a time axis (0 to 4 seconds, horizontally), a frequency axis (20 Hz to 20 kHz, deep) and a logarithmic amplitude axis (–60 to +20 dB, vertically). The waterfall diagram can be rotated freely by clicking and dragging it.

Nonlinear filtering section

The nonlinear filtering section has controls for a saturating waveshaper. The saturation generates odd-numbered harmonics (3rd, 5th, 7th, and so on).



1. Parameter controls:

    - Emphasis: Controls the type of a gentle emphasis filter placed before the saturation stage. The available types are low-frequency emphasis

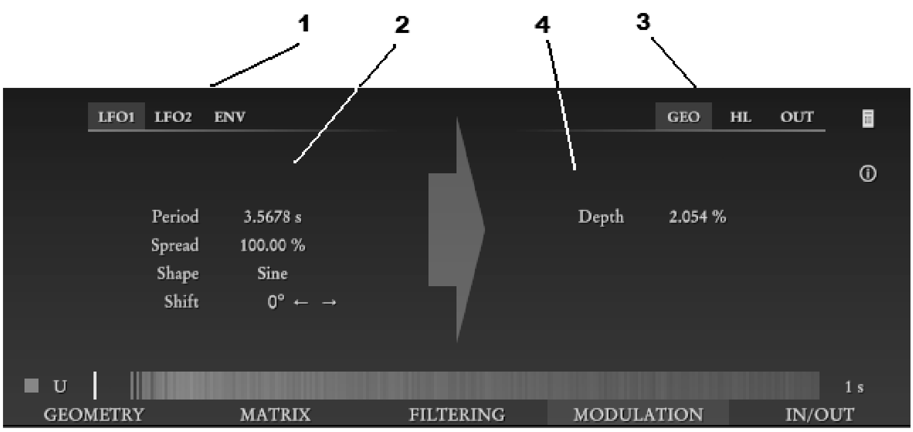
(L), neutral (N) and high-frequency emphasis (H). The low emphasis generally sounds the most aggressive, and was the default prior to version 0.7.2.

- Drive: Controls the aggressiveness of the waveshaper. The shape function saturates at a linear amplitude of ±1/drive.
- Overunity: Controls the slope around the origin, and therefore the additional energy introduced by the waveshaper. Anything else but zero may generate self-oscillation towards a limit cycle, if not attenuated through the linear filters.

2. Shape function graph: Both axes have a range of −1.41 to +1.41 (+3 dB in each direction).

Modulation Page

The modulation page is divided in two areas with multiple sections each; one area for the modulation source, and one for the modulation target. There are two LFOs (low frequency oscillators) and one envelope follower available as modulation sources. Modulation targets are either delay length, half life or output level.



1. Source Selector: One of three available modulation sources (LFO1, LFO2, ENV) is selected.

2. Source Editor: There are different editing options available depending on the source selection, explained below.

3. Target Selector. One of three avaiable modulation targets (GEO, HL, OUT) is selected.

4. Target Editor: There are different editing options available depending on target and source selection, explained below.

The LFO modulation soruce

There are two low frequency oscillators, and each of them generates an individual wave per delay line. The LFOs are controlled via four parameters: The wave period in seconds, the wave shape, a phase spread in percent and a constant phase shift, in degrees.

If the spread is zero, the LFO moves in sync for all delay lines. At full spread, the LFO is shifted 90 degrees from one delay line to the next. Full spread in combination with a sine or triangle shape means the average motion across all delay lines is zero.

The second LFO has a switch to sync with the first LFO. If this switch is active, there is only one period for both LFOs.

The ENV modulation source

There is an envelope follower that listens on the input signal, before the input signal goes through the input vectors. The envelope follower is controlled via two parameters: attack and release. Both parameters are half lives: the attack time is the time it would take the envelope to raise half the way to a constant signal level, and the release time is the time it takes to fall half the way to a constant signal level. The value of the envelope follower as modulation source is its current, linear level.

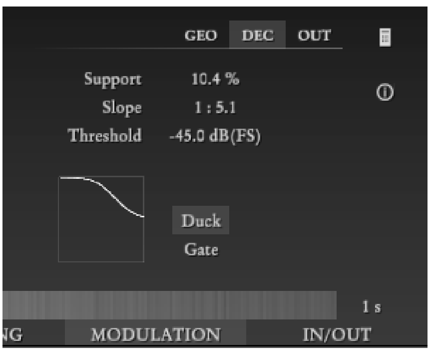
The GEO modulation target

All modulation sources can contract or extend the length of each delay line. The locations of branch points in multitap mode are however fixed and not modulated. This modulation target has only a single parameter, the depth, in percent. The depth specifies the maximal length change, at the point the modulation source has full amplitude.

Using an LFO as modulation source can lead to subtle chorusing or phasing effects, heavy pitch shifting or FM like effects, depending on the LFO parameters. Using the envelope follower as modulation source can be used for transient shaping.

The DEC modulation target

All modulation sources can shorten or prolong the decay time for each delay line. An LFO as modulation source will display a depth parameter that specifies the percentage of decay time change done by that LFO.



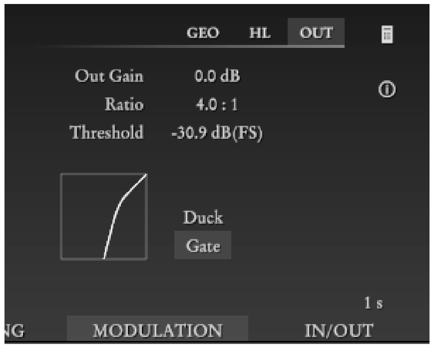
Selecting the envelope follower as modulation source enables a control field as shown on the side. The parameters in this field describe a function of decay time against envelope level: The support parameter controls a lower bound, the slope parameter controls the steepness of the dependency and the threshold parameter controls the level beyond which the effect sets in.

A switch next to the graph selects between ducking characteristic (effect is above the threshold) or gating characteristic (effect is below the threshold). The function graph has range of -90dB to 0dB on the horizontal and 1% to 100% on the vertical.

The OUT modulation target

All modulation sources can raise or lower the output level of each delay line. An LFO as modulation source will display a depth parameter that specifies the percentage of amplitude change done by that LFO.

Selecting the envelope follower as modulation source enables a control field as shown on the side. The parameters in this field describe a function of output level against envelope level: The output gain parameter controls an additional gain factor, the ratio parameter controls the ratio of envelope- to output level, and the threshold parameter controls the level beyond which the effect sets in.

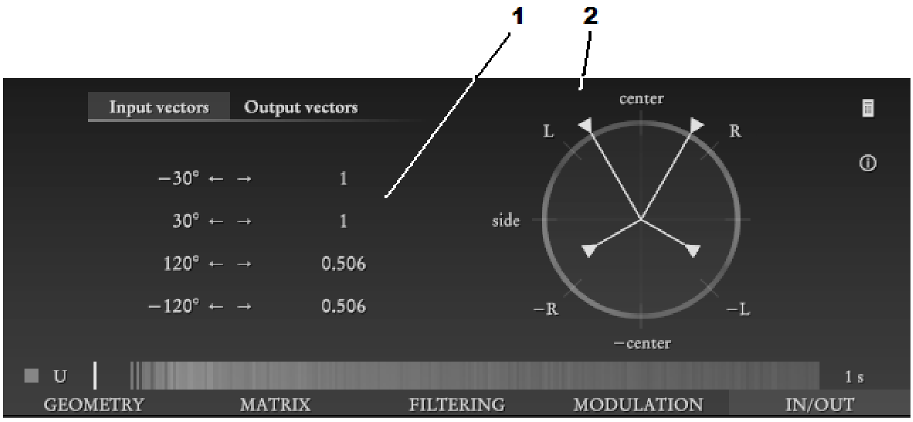


A switch next to the graph selects between ducking characteristic (effect is above the threshold) or gating characteristic (effect is below the threshold). The function graph has range of -90dB to 0dB on both axes.

Using the envelope follower on the output level is basically a compressor- or expander application. However, since the output can be delayed, this corresponds to a look-ahead compression.

In/Out Page

The In/Out Page has controls for orienting the input- and output vectors in stereo space and is divided in two identical sections, one for controlling the input- and one for controlling the output vectors.



1. Numerical vector control: This field displays the bearing angle and the radius of either input- or output vectors, depending on which ones are currently active. Which bearing angles correspond to which stereo directions is summarized in the table below. A turn of 180° always corresponds to a phase inversion.

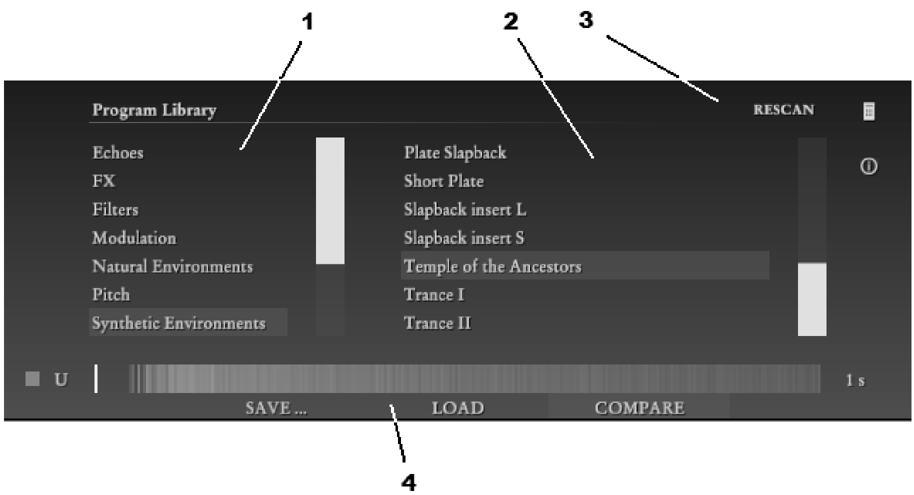
−90°	Side
−45°	Left channel
0°	Center
45°	Right channel

2. Graphical vector control: This field displays each input- or output vector as an arrow pointing from the origin into stereo space. The radius of each vector corresponds to the arrow length. The vectors can be dragged at their tips. Input vectors are displayed with inward pointing tips.

In addition to the numerical- and graphical controls, there is a color code shown on the rim around the vector field. Each bearing in stereo space has a color code that corresponds to the color shown in the echogram. Every delay line contributes to the output according to the bearing of its output vector, which shows in the echogram as impulses of the corresponding color. At the same time, every delay line is maximally sensitive for input components having the same bearing as its input vector, and not sensitive at all for input components orthogonal to that.

Program Library

The program library allows to browse and A/B compare programs. Programs are organized into categories and stored on disk as individual files, in a subfolder named `presets` that lives in the same place where the plugin is located. You can extend the library by adding your own programs, or managing files with the Windows Explorer.



1. Category List, with scrollbar: Selecting one of the available categories makes available the corresponding list of programs.

2. Program List, with scrollbar: Selecting a program enters *comparison mode*. The program is loaded temporarily, but not assigned to any of the four internal slots. You can hear the program through the output, and also see its echogram, but any attempt to modify a parameter will end the comparison mode. Leaving the program library also exists comparison mode.

3. Rescan Button: Pressing this button rescans the program library from disk to reflect any changes made externally, for instance, renaming, deleting or adding files.

4. Action Menu:

   - Save Button: Pressing this button opens a save dialog to save the currently loaded program as `FXP` file. The save dialog closes automatically when the library is exited or any modification to the current program is made.
   - Load Button: Pressing this button loads the program currently selected from the Program List into the active slot.
   - Compare Button: This button is highlighted when the compare mode is active. You can press it to toggle compare mode.

ARIESVERB is has four active program slots visible to the VST host application. These four slots can be populated with different programs for quick access or comparison. Upon startup, the first slot is populated with a default program (Temple of the Ancestors), while the other three slot are empty (Init). Switching the active slot is also done in the VST host application.

Calculator Page

The calculator page is activated via the calculator symbol on the helper menu to the right side. It offers a farily standard selection of operations, 14 digits of precision and a numerical range up to 99999999.999999. In addition, some operations are taylored for use with ARIESVERB:

- *dB* Button: This button converts decibel levels to linear levels. For instance, pressing *6* followed by *dB* yields *1.9952623149689*. Pressing *INV* before pressing *dB* reverses the operation.
- *ft* Button: This button converts feet to meters. For instance, pressing *1* followed by *ft* yields *0.3048*. Pressing *INV* before pressing *ft* reverses the operation.
- *BPM* Button: This button converts seconds to BPM and back to seconds. For instance, entering *125* followed by pressing *BPM* yields *0.48*.

Information Page

The information page is activated via the information symbol on the helper menu to the right side. It has three sections, titled Plugin, Host and System.

The Plugin section displays the version number of the plugin together with the licensee information.

The Host section displays information related to the current state of the VST host, like sample rate and tempo, as well as the host vendor as it is reported to the plugin.

The System section displays information related to the computer system, as well as the current CPU usage of ARIESVERB with respect to a single core, and the peak performance of the algorithm, in billion floating point operations per second (GFlop/s).

Appendix

Legacy parameter values

AriesVerb of version 0.4 and before had no graphical user interface and offered limited control over what can now be adjusted on the Algorithm- and Matrix Pages. Both relative delay lengths and the rotation plane were restricted to a selection of builtin table values.

Table 2 lists the geometry presets of AriesVerb 0.4. To emulate this parameter, set the length for delay lines 2 to 4 in the Algorithm Page to the values listed in the table. The first delay line must always be set equal to the time base.

Table 3 lists the rotation plane presets of AriesVerb 0.4. To emulate this parameter, set the axis vectors for the rotation plane in the Matrix Page to the values listed in the table. The Sparse presets cannot be emulated, because they had elements othen than zero and one. To emulate the block$_2$ preset correctly, set the beta angle to negative.

Rotation Matrices

This section was written to help explain the controls behind the feedback matrix. The job of the feedback matrix is to intermix sound between different channels while preserving the overall signal energy. This is accomplished in AriesVerb by means of a rotation matrix.

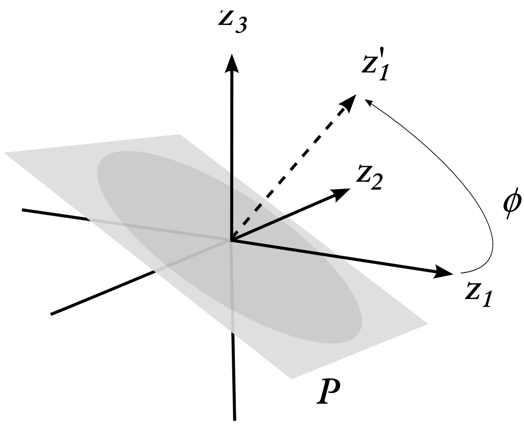
As long as there are only two channels, which is the case for instance in a stereo signal, a rotation can be described completely by a single parameter: the

Name	t_2	t_3	t_4
SPHERE	1	1	1
BOX$_1$	1.04	1.11	1.16
BOX$_2$	1.05	1.14	1.3
DENSE$_1$	1.22	1.34	1.72
DENSE$_2$	1.78	2.44	2.62
DRUM	1.15	1.63	2.41
HALLWAY	1.41	2.22	3.13
ROOM$_1$	1.67	2.52	3.11
ROOM$_2$	1.44	2.59	3.35
SPARSE$_1$	2.24	3.78	4.67
SPARSE$_2$	2.61	3.78	5.1
MINOR	$4/3$	$5/6$	2
MAJOR	$4/3$	$8/5$	2
HARMONIC	$4/3$	2	4
OCTAVES	2	4	8
FIFTHS	$3/2$	2	3
DETUNED	+0.01%	+0.3%	+0.31%

Table 2: Geometry presets used in ARIESVERB 0.4 for delay lines 2 to 4, in multiples of the time base

Name	$\vec{u}$	$\vec{v}$
DENSE$_1$	$(1,0,0,-1)$	$(0,1,-1,0)$
DENSE$_2$	$(1,0,0,-1)$	$(0,-1,-1,0)$
SPARSE$_1$	—	—
SPARSE$_2$	—	—
BLOCK$_1$	$(1,0,0,0)$	$(0,1,0,0)$
BLOCK$_2$	$(1,0,0,0)$	$(0,1,0,0)$

Table 3: Rotation plane presets used in ARIESVERB 0.4. See the text for more explanation.



(a) Feedback rotation.

Figure 3: Illustration of feedback rotation. Rotation plane (P), rotation angle (ϕ).

rotation angle. If more than two channels are involved however, there is additional freedom in choosing which pair of channels takes part in the rotation. It is even possible to choose a mixture of channels as taking part in a rotation. This leads to the concept of a rotation plane, which is a plane spanned by two axes, where each axis may be formed by a combination of channels.

Take for instance figure 3, a rotation involving three channels. The rotation is performed in the plane spanned by axis z_2, and an axis combining $z_1$and z_3.

In ARIESVERB all rotations involve four channels, and this is a situation that cannot be visualised on paper. Even more, it allows to rotate in two planes simultaneously, which is why ARIESVERB has controls for two independent rotation angles for each matrix.

Filters Inside a Feedback Loop

This section was written to help explain the behaviour that can be observed with certain filter parameter values, and may come unexpected to users with experience in filters or EQ curves.

ARIESVERB offers three types of filters to shape the characteristic of a frequency dependent decay time. Due to the fact that these filters operate inside a feedback loop, the standard rule of logarithmic addition for a series application (addition of decibel values) does not hold.

A numerical example: If the half life was set to 8, that corresponded to a feedback gain of about 92% such that 8 cycles through the feedback loop reduces the amplitude to 50%. On the other hand, if the half life was 16, that corresponded to a feedback gain of about 96%. If we wanted to have a half life of 16 only for a certain frequency band, we need a filter to boost that frequency band from 92% to 96%, which is an increase of just 0.4 dB.

As can be seen, the filter characteristic needed for a certain increase or decrease of decay time is dependent on all other factors that influence decay time. Especially at a regime close to unity, a filter response may look unusually steep, and moving a filter may cause side effects at frequencies far away.

Credits

Built on GCC 4.4.0 using wxDev-C++ 7.0 RC 4
Built using FreeType 2.1.8
Documentation written on LyX 1.6.3 using MiKTeX 2.7

VST Technology by Steinberg