AriesVerb

Manual

for version 0.7.7

ARIESVERB Manual
for version 0.7.7
May 22, 2017

`http://ariescode.com`

License

License for ARIESVERB 0.7.7 (the "Software") between you (the "Licensee") and the author, Christian Schüler. The Software comes in two embodiments, an evaluation version (the “Demo”) and a full version. The Demo version is for evaluation only. Within a 30 day period from the date of purchase of the full version, 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.

You agree to be bound by this license by installing the Software. 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 the Software, for any artistic, scientific or commercial application, including music production.

The 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. 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, would enable the phase effects of very short delays like in flangers. Or modulation rates up to the kHz range, 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

The evaluation version can be downloaded from the ARIESCODE homepage. If you have bought the full version of ARIESVERB, then you have received download instructions via e-mail. In either case, the download is delivered as a compressed archive ("zip file") and you should see an option to extract its contents when right-clicking on it. Some browsers may do this step automatically for you, in which case you'll see a folder already extracted for you. In any case you should end up with these files:

`Ariesverb.vst` (macOS) –or– `Ariesverb-32.dll` and `Ariesverb-64.dll` (Windows)	The plugin file
`readme.txt`	"Read-me" text with release information
`-> VST Plug-Ins`	A shortcut to the system's VST Plugins folder (macOS), for convenience

Open the *readme* file and look for what it says about the version you downloaded. It shoud indeed be version 0.7.7 or higher. If everything went ok, move the plugin file or files to a location where your VST host can find them. On macOS this is a standardized location and you can use the supplied shortcut to quickly go there.

On Windows this is typically `C:\Program Files\VSTPlugIns`, but if in doubt, you need to check the settings in your host application or configure plugin folders.

Running ARIESVERB

If you have bought the full version, then ARIESVERB will ask you for a license key the first time it is run:

Click on the shaded input area on the right hand side, and type in your key or copy and paste it. The license key is saved with the App Preferences on the computer and needs to be re-entered only if this information is lost, for instance, after a re-install.

Upon success, you should now see the Front Panel of ARIESVERB. Probably you want to check out some of the preset programs shipping with ARIESVERB now. Just go ahead, press the book symbol on the right side of the screen, which opens 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, there are four delay lines in parallel, and a 4×4 matrix sits in the place of a simple feedback gain (this is the "FDN" bit). 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 are the four delay lines ($z_1 \ldots z_4$). Each delay line has an adjustable length and can feed back into each other delay line, via the feedback matrix (A). The sound signal would circulate inside this loop forever, if it was not attenuated 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.

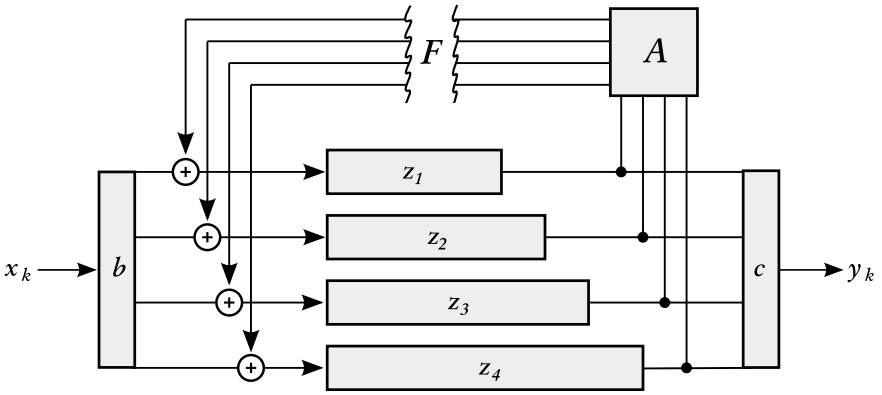


Figure 1: Block diagram of the signal flow inside ARIESVERB. Feedback matrix (A), input vectors (b), output vectors (c), input signal (x), output signal (y), delay lines (z_i), scalar feedback gain, filters and saturation (F).

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.

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. It's echo density increases with time, while its mode density increases with frequency.

Simulating an increasing echo density is easy for an algorithmic reverb to do. This is evident in the recursive structure of the algorithm: In case of a 4×4 structure, the first generation spawns 4 echos, the second generation spawns 16 echos, the third generation 64, then 256, and so on. The only thing to mind here is to avoid simple integer ratios of delay lengths, since then the echos would fall on top of each other.

Simulating an increasing mode density is more difficult to do, since algorithmic delay elements produce a constant spacing of resonant peaks in the spectrum, like a harmonic series. One old trick to this end would be modulation, which would produce side bands at a bandwidth increasing proportionally to frequency. This maybe explains why reverb with modulation is so popular. ARIESVERB proudly can do an excellent job of modulation.

There is also an eternal conflict between echo density and mode density, as one of them always decreases when trying to raise the other, like a yin and yang.

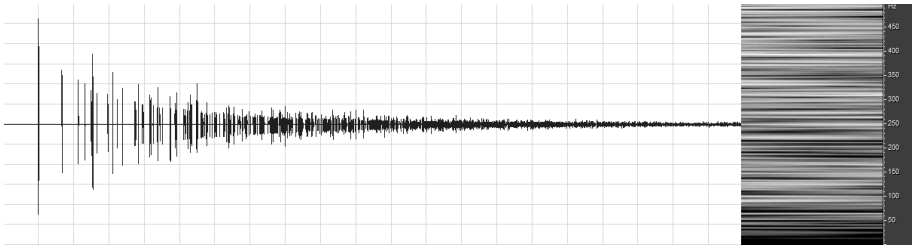


Figure 2: Impulse response of an ideal rectangular room, in time domain (left) and frequency domain (right).

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

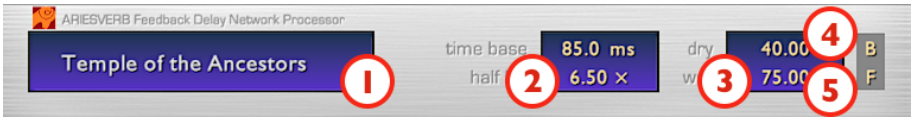
ARIESVERB offers a solution to break out of the aforementioned conflict simply by brute force. It is called *multitap* mode and provides sufficient density for the simulation of natural environments, while retaining the CPU efficiency of the original algorithm.

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 matrix 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 isomorphic to a 16×16 FDN system, comparable in quality to hardware reverb units of the 1990's. However, while the latter ones often employed hard-coded feedback matrices to keep costs down, ARIESVERB remains fully general.

User Interface

Front Panel

The front panel is in the lower part of the user interface and is always visible. It displays the name of the currently active program and has controls for the four most important parameters: time base, half life, dry- and wet mix.



1. Current program name.
2. 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 a reverb or echo tail to decay to half (−6 dB). Ten half lives therefore equal reverb time RT60. The half life however is not given directly, but as a multiple of the time base. In this way it becomes a measure of resonance, like the Q factor of a filter. It essentially the number of cycles, or repetitions, through the feedback loop in order to reach half amplitude.

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

4. Bypass button. When active, all processing is bypassed.

5. Hold button. When active, enables the *lossless prototype*. This means that all filtering is disabled, and the sound signal circulates infinitely.

Editor Window

The editor window sits on top of the front panel and by default, displays the summary page with a description of the current program. It also functions as a hub to navigate between the individual editor- and helper subpages.

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

2. Global navigation bar: The navigation bar has five symbols that can be clicked to activate other pages. From top to bottom: Editor subpages (pencil symbol), program library (book symbol), calculator (calculator symbol), information and statistics (information symbol) and settings (wrench symbol). Clicking on an active symbol again exits that page and returns to the summary page.

3. Undo and redo buttons: These buttons are always visible and allow to undo or redo any modification.

Editor Subpages

A number of additional controls become visible once the editor subpages are activated by clicking on the pencil symbol in the navigation bar.

1. Editor navigation bar: This is an additional navigation bar and is visible on all editor subpages. It selects between algorithm page (`ALGO`), matrix page (`MATR`), filtering page (`FILT`), modulation page (`MOD`), and input-output page (`I/O`).

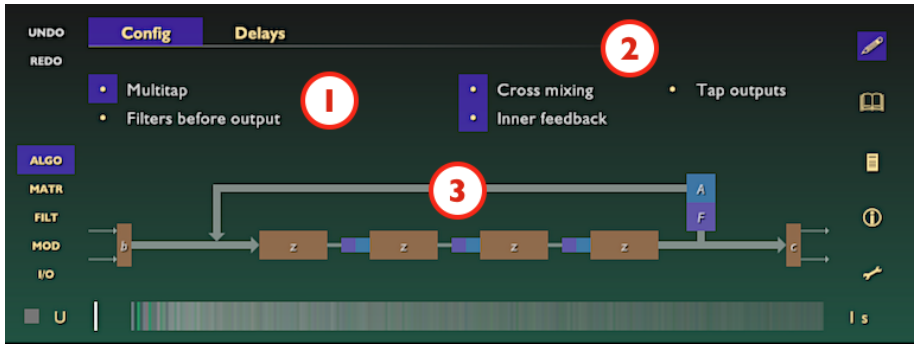
2. Echogram: The echogram is visible on all editor subpages. It shows the instananeous impulse response of the current program, color coded to stereo phase. If the current program is time dependent due to modulation, the echogram will be animating. To the left of the echogram is a switch to select the position of the test impulse. It can be center (`C`), left (`L`), right (`R`), side (`S`) or uncorrelated (`U`). To the right of the echogram is a selector for the time scale.

Algorithm Page

The overall algorithm of an ARIESVERB program is edited on the 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 algorithm configuration and has a number of switches to select between different configurations.



1. General switches:

   - Multitap: If enabled, the delay lines are further 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: If enabled, the filters are applied 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 switches: An additional three switches are only visible if multitap mode is enabled.

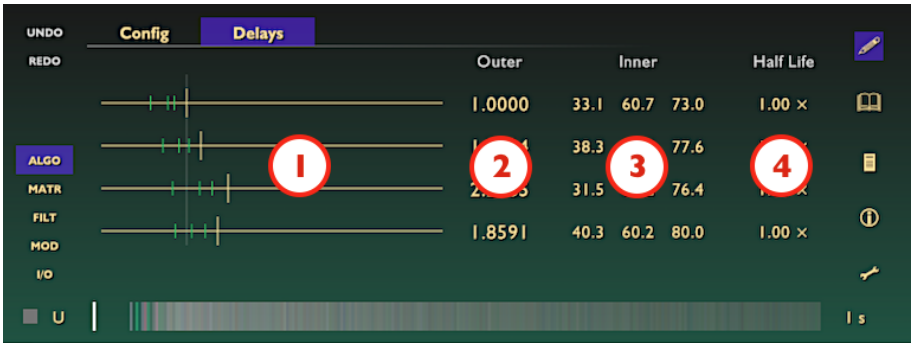
   - Cross mixing: If enabled, conntects *the same* branch points of *different* delay lines with one another.
   - Inner feedback: If enabled, connects *different* branch points of *the same* delay line with one another.
   - Tap outputs: If enabled, all branch points are routed to the output as additional 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. The signal flow is understood to be vectorized, so that a

single thick line actually represents four independent lines, and a delay line block actually repretents four delay lines that may have different lengths. Legend: feedback matrix A (blue), input vector b, output vector c, delay lines z (all orange) and filters F (purple).

Delays Section

The delays section has controls to adjust the overall lengths of individual delay lines ($z_1 \dots z_4$), 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 at the numerical value of 1. The location of branch points is represented by a small green ticks if multitap mode is enabled.

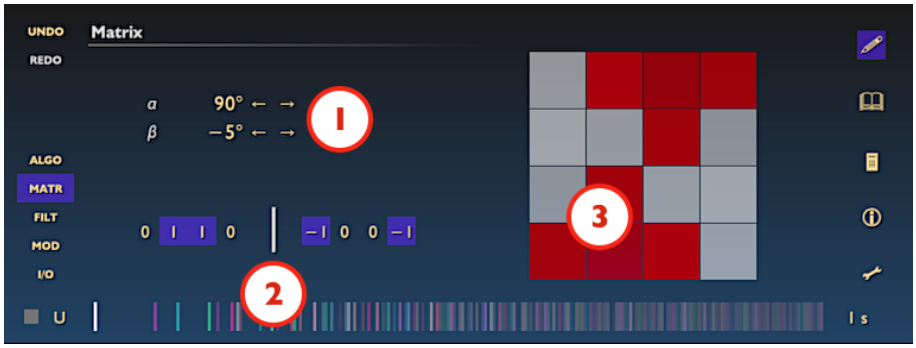
2. Delay length controls: Numerical controls corresponding to the sliders.

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

4. Half-life modifiers: Allows to adjust the half-life on individual delay lines as a multiple of the global half-life.

Matrix Page

The matrix page has controls for the feedback matrix. The feedback matrix is a special animal since it must be energy-conserving ("unitary"). Therefore, it is not possible to just enter any old values for it. The solution in ARIESVERB is to limit the choice to *rotation matrices* and have the user control the rotation *angles* and the rotation *plane*.



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

2. Rotation plane control: Numeric control for two direction vectors that, together, span the plane of rotation. For simplicity, the possible values for each element are limited to 0, 1, or -1 , and the vectors are constrained to be orthogonal. When the α angle is at 90°, the resulting matrix will exactly rotate the first vector into the second[1]. Under the hood there always exists a corresponding anti-parallel rotation plane, which is not shown. This plane is then controlled by the β angle.

[1]Don't believe me? Take the example shown in the figure and multiply the matrix columns with the elements of the first vector (0,1,1,0) and add everything up horizontally (the grey and red boxes in this case represent 0.5 and -0.5 resp.). The result equals the second vector!

3. Matrix visualization: A visual representation of the resulting feedback matrix, with positive feedback displayed as levels of grey, and negative feedback as levels of red. The cell representing feedback from delay line i to delay line j is found in row j, column i, as in the following layout:

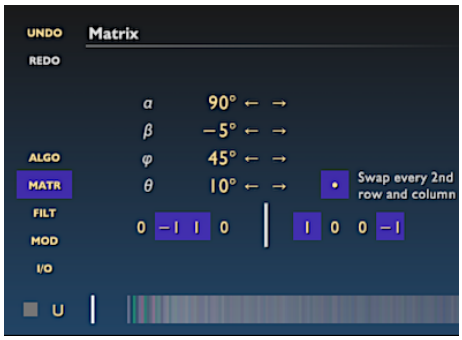
$$\begin{array}{cccc} 1\to 1 & 2\to 1 & 3\to 1 & 4\to 1 \\ 1\to 2 & 2\to 2 & 3\to 2 & 4\to 2 \\ 1\to 3 & 2\to 3 & 3\to 3 & 4\to 3 \\ 1\to 4 & 2\to 4 & 3\to 4 & 4\to 4 \end{array} .$$

Multitap mode

When multitap mode is enabled, the matrix switches to a different arrangement and becomes a matrix of matrices (essentially a 16×16 matrix). In multitap mode there are 3 additional rows and 3 additional columns for each delay line, representing the 3 additional branch points. If we label the branch points with letters a, b, c, d and assign the delay line entry point to branch a, the matrix layout is:

$$\begin{array}{cccc|c|c} 1a\to 1a & 1b\to 1a & 1c\to 1a & 1d\to 1a & 2a\to 1a & \\ 1a\to 1b & 1b\to 1b & 1c\to 1b & 1d\to 1b & 2a\to 1b & \\ 1a\to 1c & 1b\to 1c & 1c\to 1c & 1d\to 1c & 2a\to 1c & \cdots \\ 1a\to 1d & 1b\to 1d & 1c\to 1d & 1d\to 1d & 2a\to 1d & \\ \hline 1a\to 2a & 1b\to 2a & 1c\to 2a & 1d\to 2a & 2a\to 2a & \\ & & \vdots & & & \ddots \end{array} .$$

Inner feedback



Additional rotation angles phi (ϕ) and theta (θ) appear when *inner feedback* is enabled. These angles apply to all 4×4 sub-matrices in the same way as α and β apply to the parent matrix. They control the amount of intermixing between branch points on the same delay line, and can affect the density character or the attack character of the echo buildup. Setting both ϕ and θ to zero has the same effect as disabling inner feedback altogether.

The option to swap "every second row and column" was originally introduced for backward compatibility, since at one point in time these were indeed erroneously swapped. When a preset from such a version of ARIESVERB is loaded, this option is then automatically ticked in order to faithfully reproduce the sound. On the other hand, this is may be a useful option to shape the matrix, so it has been retained.

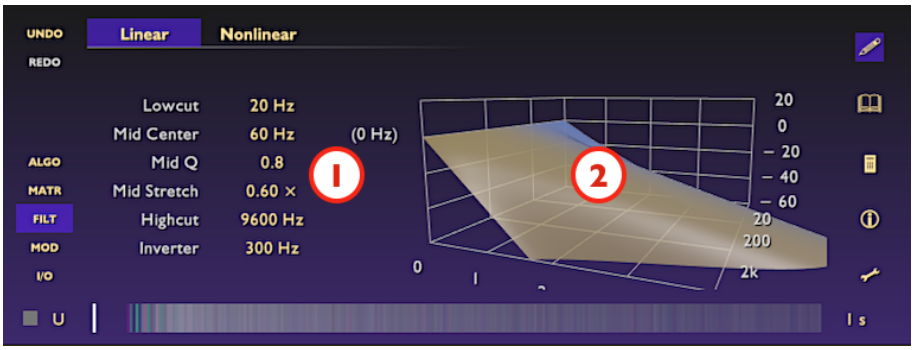
Filtering Page

The filtering page has controls for the filters (F in the block diagram at the beginning in figure 1). 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 1^{st} order low-cut filter (6 dB/oct), a 2^{nd} order peaking filter (6 dB/oct on both sides), and again a 1^{st} order high-cut filter. It also sports a frequency dependent inverter. The 6 dB slopes may not sound impressive, but these slopes do not directly translate to the spectrum, instead they modify the half-life. See the section about "Filters Inside a

Feedback Loop" in the appendix to understand why even shallow filters can have a dramatic effect.



1. Filter controls, from top to bottom:

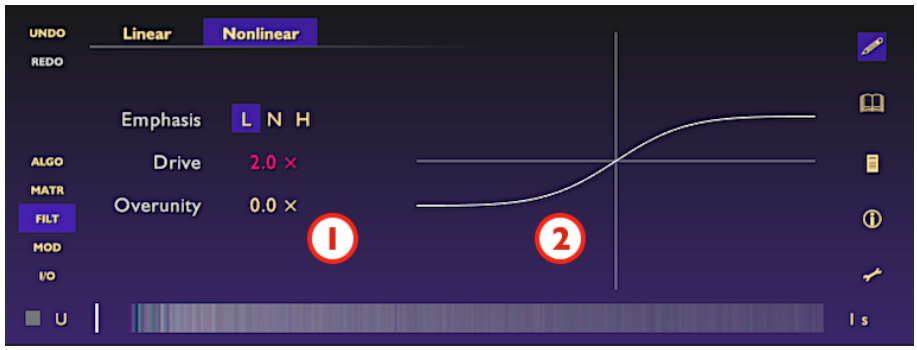
   - Lowcut: Frequency at which the half life is reduced to 50% by the lowcut filter.
   - Mid Center: Center frequency of the peaking filter. In parentheses, the center frequency of the actual peak.
   - Mid Q: The quality factor (bandwidth) of the peaking filter.
   - Mid Stretch: The change in half life for the peaking filter at its own center frequency.
   - Highcut: Frequency at which the half life is reduced to 50% by the highcut filter.
   - Inverter: Corner frequency, at which the phase is shifted by 90°. Below this frequency, the phase shift will tend to zero and above this frequency, the phase shift will tend to 180°.

2. Waterfall diagram: This is a 3-D visualization of the frequency dependent decay behavior, having a time axis (0 to 4 seconds, horizontally), a frequency axis (20 Hz to 20 kHz, along the depth) and a logarithmic amplitude axis (−60 to +20 dB, vertically). The action of the inverter is color-coded

into this diagram, with blue being neutral (0°) and orange representing inversion (180°). 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, modelled after a "tanh" function (hyperbolic tangent). The saturation generates odd-numbered harmonics (3rd, 5th, 7th, and so on).



1. Parameter controls, from top to bottom:

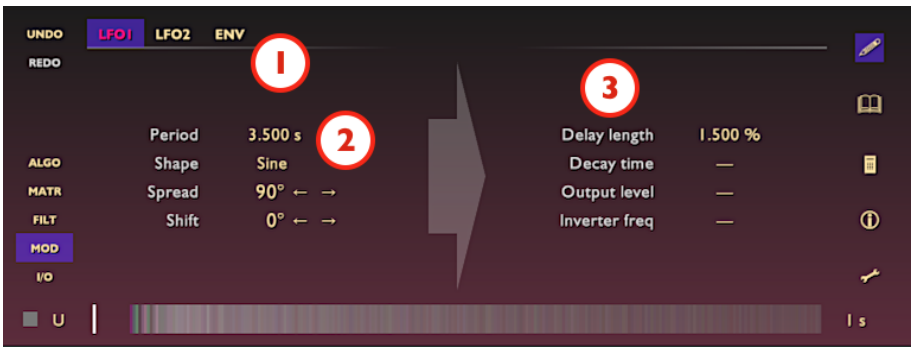
    - Emphasis: Controls the type of a gentle emphasis filter that is placed before the saturation stage. A matching de-emphasis filter is then automatically applied after the saturation. The available types are low-frequency emphasis (`L`), neutral (`N`) and high-frequency emphasis (`H`). The low emphasis will usually sound the most aggressive, because the corresponding de-emphasis filter amplifies the high frequency harmonics introduced by the waveshaper. Low emphasis was the default prior to version 0.7.2.
    - Drive: Controls the aggressiveness of the waveshaper.
    - Overunity: Controls the slope around the origin, and therefore the additional energy introduced by the waveshaper. Anything else but

zero may drive the feedback loop in self-oscillation towards a limit cycle, if not attenuated through the linear filters.

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

Modulation Page

The modulation page is divided into a left hand side for the modulation sources and a right hand side for the modulation targets. There are two LFOs (low frequency oscillators) and one envelope follower available as modulation sources. Modulation targets are the delay length, half life (decay time), output level or inverter frequency.



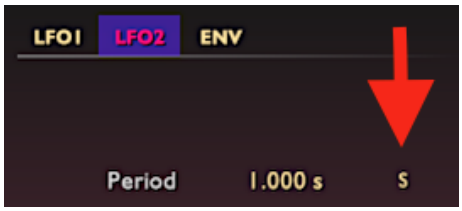
1. Source Selector: One of three available modulation sources (`LFO1`, `LFO2`, `ENV`) is selected.

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

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

The LFO modulation source

There are two low frequency oscillators available, with each of them actually generating 4 independent signals (one for each delay line). The LFOs are controlled via four parameters: The period in seconds, the wave shape, a phase spread and a phase shift, both in degrees. If the spread is zero, the LFO moves in sync for all delay lines. At 90° spread, the LFO is shifted in phase 90 degrees from one delay line to the next, etc.



The second LFO has a switch to sync with the first LFO (shown on the side). 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 delay length modulation target

All modulation sources can contract or extend the length of delay lines. The combined effect of multiple sources is additive. The locations of branch points in multitap mode are however fixed and not modulated, which means that in multitap mode, a delay line can only contract as far as its rightmost branch point, and no further.

This delay length modulation target has only a single parameter, the depth, in percent. The depth specifies the percentual length change when the modulation source has full amplitude.

What happens when a delay line gradually changes length is pitch shift. 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 create transient effects (employed by the program “Brass Attack Maker”, for example).

The inverter frequency modulation target

All modulation sources can modify the corner frequency of the inverter in the linear filter section. Modulation of the inverter creates *phase modulation*, in contrast to the delay length modulation described above, which is *frequency modulation*. Everything else is pretty similar though.

The decay time modulation target

All modulation sources can shorten or prolong the decay time (equivalently the half life) individually 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 for the decay time 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 corner level at 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. Gating is necessary to create a finite response when overunity is used in the non-linear filter section. Without gating, the limit cycle would be endless. Ducking can be used to create a *ducked reverb*, ie. a reverb that becomes active only at the end of a phrase.

The output level modulation target

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

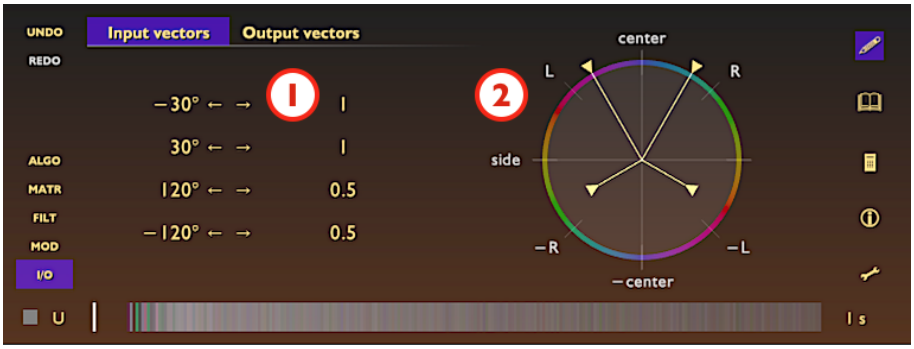
Selecting the envelope follower as modulation source enables a control field as shown on the side. The parameters in this field are the same as typically found in compressors and describe a function of output level against envelope level: The gain parameter is just that, an additional gain factor; the ratio parameter controls the compression or expansion ratio; and the threshold parameter controls the corner level at 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 against the input, this corresponds then to a look-ahead compression. The graph of the compressor in ARIESVERB has a very soft knee, which was done on purpose in an attempt to give it an analog feel.

Input-Output Page

The input-output page has controls for orienting the input- and output vectors in stereo space and is divided into two identical sections, one for the input side and one for the output side.



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. The correspondence between bearing angles and stereo directions is summarized in the table below. A turn of 180° always corresponds to a phase inversion.

±90°	Side
−45° or 135°	Left channel
0° or −180°	Center
45° or −135°	Right channel

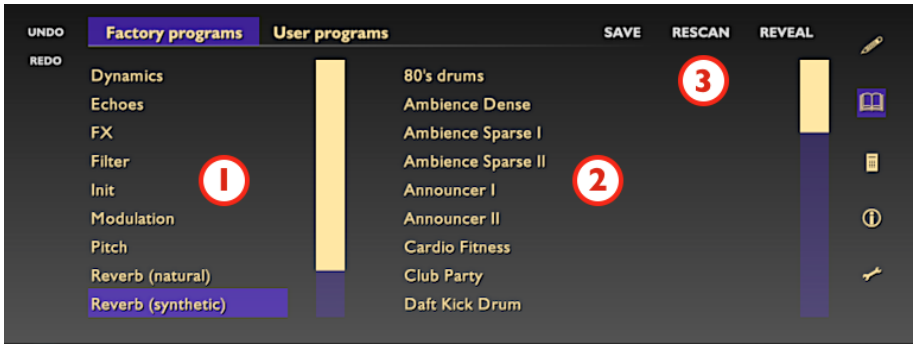
2. Graphical vector control: This field displays each input- or output vector as an arrow pointing from the origin into stereo space. The radius of each vector corresponds to the arrow length. The vectors can be dragged at their tips. Input vectors are displayed with inward pointing tips.

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 at all sensitive for input at an angle of 90° to that.

Program Library

The program library is activated by clicking on the book symbol on the right hand side navigation bar. Programs are organized into a one-level hierarchy of categories. The factory programs are stored inside the plugin and are immutable (technically, on macOS you could open the bundle and replace its resources). In contrast, the user programs are stored as individual files in a standard location depending on your operating system. You don't need to remember where this is, because the `REVEAL` button (see below) will always take you there.



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 immediately loads it and makes it current for the active slot. This action is undoable, so a quick A/B comparison can be done by loading a program and then undoing it.

3. Action Menu:

- Save Button: Pressing this button opens a save dialog to save the currently loaded program as *FXP* file in the currently selected category of your user library. If no category is selected, the "Default Category" will be used. If the default category not even exists, it will be created.
- Rescan Button: Pressing this button re-scans the entire user library. This is necessary to sync the display when files have been modified externally.
- Reveal Button: Pressing this button will open a Finder or Explorer window at the location where the currently selected program is stored on your disk.

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 done in the VST host application.

Calculator Page

The calculator page is activated via the calculator symbol on the right hand side navigation bar. 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.

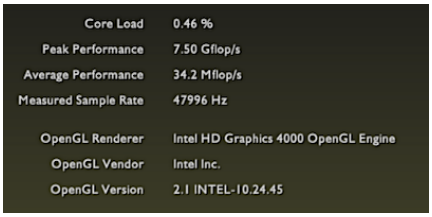
- *BPM* Button: This button converts seconds to BPM or vice versa. For instance, entering *125* then pressing *BPM* yields *0.48*, pressing it a second time will yield *125* again.

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.

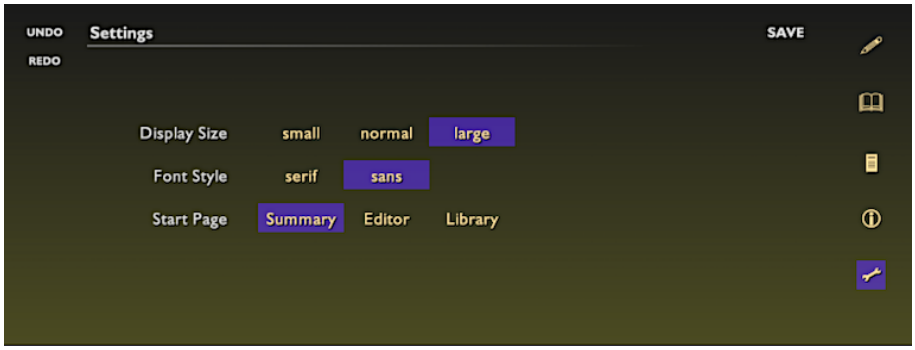
The Host section displays information that is reported by the VST host. These are, from top to bottom: the product name, version and vendor of the VST host; the supported VST version; the current sample rate and the current tempo in BPM.



The System section displays information related to the computer system. These are, from top to bottom: the current percentual CPU usage of ARIESVERB with respect to a single core; the numerical peak performance of the algorithm, in billion floating point operations per second (Gflop/s); the sustained average floating point performance, in million floating point operations per second (Mflop/s); the measured sample rate and information about the current OpenGL driver.

Settings Page

The settings page is activated by pressing the wrench symbol on the right hand side navigation bar. You can select display (window) size, font style and which page should be the start page.



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, negate the beta angle.

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.

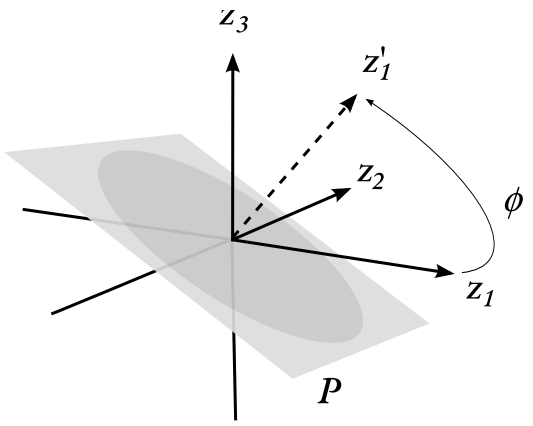
As long as there are only two channels, which is the case for instance in a stereo signal, a rotation can be described completely by a single parameter: the

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	$(0,1,-1,0)$	$(1,0,0,-1)$
DENSE_2	$(0,-1,-1,0)$	$(1,0,0,-1)$
SPARSE_1	—	—
SPARSE_2	—	—
BLOCK_1	$(0,1,0,0)$	$(1,0,0,0)$
BLOCK_2	$(0,1,0,0)$	$(1,0,0,0)$

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.

Acknowledgements

Built with GCC 6.3.0 and XCode
Font textures created using FreeType 2
Documentation written on LyX 2.2.0 using TeX Live 2016

VST Technology by Steinberg