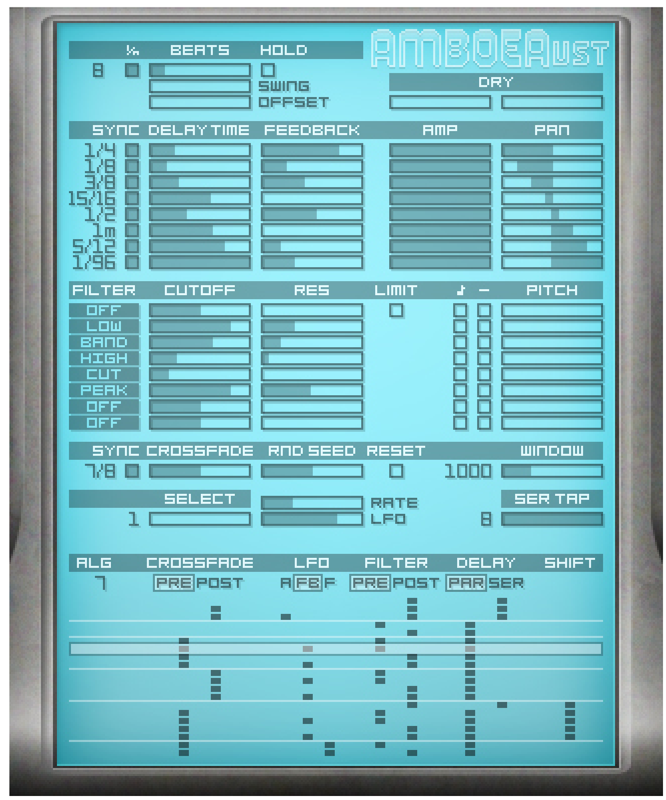
Amboea VST

This effect was birthed from a discussion on kvraudio. I was considering a multitap delay that produced irregularly timed echoes. During the discussion, kvr member spirit requested an effect for ambient music with varying synchronised delay taps and modulated feedback. I saw the idea as an ambient audio amoeba producing a panoramic landscape of incidental echoes.

Spirit also mentioned the option for some delay taps to include pitch shifting. I considered (biquad) filters to be a given. I envisioned spirit's idea functioning as a bank of delays in parallel with an algorithmic process mixing the signal into each tap.. but could the mixing go after the delay taps, similar to my older Discipline VST effect? And how about running the delays in serial? Do the filters go inside the delay loop, or afterwards?

One part was easy.. use integer value delay lines since the length itself isn't being modulated... this gives the best possible signal reproduction. Obviously stereo delay lines were necessary. Before long it was clear that not only was the idea a monster.. with endless, appealing choices for configuration.. cpu resourcing would be as well in order to run all the stereo, filtered delay lines, with pitch shifting and independent modulation simultaneously.



Amboea VST includes 20 algorithms opportuning its eight stereo delay lines. Some algorithms, eg. 1 and 4, are simple serial and parallel delays, which use a small amount of cpu and are more efficient than using several instances of other VST to accomplish the same purpose. Other algorithms have increasingly larger cpu loads.

Each delay line has a maximum length of 65535 samples, making the actual time dependent on the host sample rate. The output of each delay may be panned and filtered in all modes.

The lower quarter of the gui displays the configuration options. To change the algorithm number, drag vertically anywhere in this region. The white bar indicator will highlight the current selection on the graph, and the boxed selection indicators will also highlight the current settings.

Filtering

The biquad filters may be applied inside the delay loop or afterwards in the signal chain. Resonant filters in delay loops present the issue of increasing gain, which can produce very loud signals. As this tonality is desirable to some musicians, limiting is optional. Limiting is available in all pre-filtered modes. In peak mode, resonance increases the gain and narrows the resonant band. This filter has no effect when no resonance is applied.

Serial and Parallel Configurations

The primary delineation between algorithms is whether the delays are run in serial or parallel configuration. In serial arrangement, the original signal is routed to the first delay, routes to the second, and so on. Delay time is cumulative so that if each delay tap is 1 second in length, the output from the eighth delay occurs eight seconds after the input event.

For serial modes, only one tap is selected to be fed back into the input. The **ser tap** parameter in the lower right corner indicates the currently selected tap. Note that all serial modes are post-filtered and have post-crossfading. Because all delay lines are integer length (having significantly lower cpu than any appreciable interpolation scheme allowing finer length resolution) note that bpm-synced delay patterns will drift out of sync with hosts after a few minutes because of the rounding discrepancy.

In parallel modes, all delay lines feed back to themselves, and, apart from algorithm 4 which is a simple bank of eight parallel delays, all of them use the algorithmic crossfading feature.

Algorithmic Crossfade Mixer

The heart of Amboea is the algorithmic crossfader. This algorithm is based around a host synced clock. At every clock event, a new set of delay taps are selected. The previous set fade out while the new set are mixed in.

The parameters for this clock are at the top of the gui. The clock runs at an interval specified in host synced beats, or quarter notes. The clock rate is either integer multiples of the beat, eg. every 2 beats, 3 beats et c. or integer divisions, eg. every 1/2 beat, every 1/3 beat. **The crossfade will not operate when the host transport is not running.** Also note that if using a long fade time, all delays start mixed out at plugin initialisation.

The clock section is similar to my previous algorithmic MIDI VST (with the exception that normal selection is for multiples of beats and not divisions of beats) and similarly features a **hold** button - (previously iconised as a wrench) this can freeze the current clock values while a new value is set so that the clock rate may be adjusted during performance, eg. from 1/2 to 1/8 without dragging the clock through intermediate values. The clock may also be swung (shifting every 2nd pulse forward or backwards) and offset (shifting all pulses forwards and backwards).

At every clock event, up to eight of the delay taps are selected to be audible. The **select** parameter in the lower left indicates how many of the taps will be chosen. Those taps will be raised to full amplification at a rate set by the **crossfade** parameter. The crossfade time may either be set in seconds, or as a ratio of the clock interval, eg. 1/2 the clock interval, et c. Ratios up to 4 times the clock rate are allowed - coefficients longer than the interval, ie. greater than 1, will slowly raise or lower the volume of taps as they are selected and deselected.

The pattern of tap selections is set by the **random seed** value. Adjust this parameter to select a different pattern. To reinitialise the pattern, hold the **reset** button down during a clock pulse.

Pre-crossfaded mixing modulates the level of the dry signal routed to each delay. Post-crossfaded mixing feeds the dry signal to all delays and selects a number of delay taps for the output. In either mode, one (or more) of the delays can be set to zero so that the dry signal can be handled by the algorithmic crossfader.

Patching Crossfading Algorithms   * VERY IMPORTANT!!! *****

It is difficult to patch what you can't hear. Patching Amboea, or any multidelay, requires focus to remember which delay tap is doing what, otherwise you'll be scrambling to figure out which of the eight delay taps is the one you want to adjust. It is recommended that crossfading be the last element to patch. **While patching any crossfade algorithm, set select to 8 so that all taps are audible, then use the amp sliders to solo each delay and patch them.** Solo buttons were omitted from the design because, although convenient for patching, inclusion would have added an extra sixteen multiplies to the cpu load on every sample of operation. The same functionality can be created using an automation VST if required.

LFO Modulation

The LFO is applied to feedback amount in most patches. It uses a smoothed, randomised contour to apply noncyclic variation to each tap. Different modulation is applied to each delay.

The LFO is not included in all algorithms because it is cpu intensive - LFOs do not shut off when the amount is reduced to zero in case continuation is required when the level is raised again.

Apart from feedback, two other destinations obviated themselves - amplitude of taps in serial mode, which can add variation to host synced rhythmic patterns, and filter cutoff. The latter is more cpu intensive as filter coefficients must be recomputed on every sample for a smooth response. Duplicate this eight times and you have a lot of computations :) The filter cutoff mode is available with pre and post filtering for discretely different effects.

Pitch Shifting

A simple two-read-pointer windowed pitch shifting algorithm is available in both serial and parallel modes. The simplest application is to use the serial delay as an audio arpeggiator. This algorithm features adaptive processing so that delays with no pitch shift do not use extra cpu.

The pitch shifting algorithm works by changing the speed of the read location of the delay. Two read pointers are crossfaded so that the pointer doesn't exceed the delay loop boundaries. The window parameter determines the length in samples of the read location. It is advisable to set the window parameter longer than the wavelength of the lowest note in the performance.

Using short window times has the disadvantage of sounding robotic, using long window times sounds less grainy, but introduces an echo effect produced by the use of two read pointers. The echo is more noticeable on very short sounds like percussion. Generally it's best to use the shortest time that affects the lowest frequency you intend to use.

The range of the pitch shifter slider is plus or minus one octave. Pitch values may be quantised to 12ET pitch values. Quantisation ranges from -12 to +12 semitones and includes -/+24 semitones (two octaves) at the highest setting.

Parameter Resolution

Because of the high cpu load, several of the parameters are recomputed at buffer rate instead of per sample. Feedback and amplification are computed on each sample for smooth modulation. Filter coefficients and panning are computed at buffer rate. As this is often in excess of 440Hz, modulation of these parameters will generally not suffer from zipper noise.

Sparklies

Do not be alarmed! Sparklies do not use any extra cpu after the first few seconds of VST activation.

Footnote

Amboea is the kind of circuit one could easily create in a modular environment, which isn't convenient for all playing or composition styles. More intense configurations use over 10% of my 1.6g processor.. which is not a point of pride as a developer. As soon as I've finished patching and releasing it I do intend to visit it as a one-stop solution for instant ambient music production :) This is kind of a tough release.. delays are such a simple concept, and, especially during finalisation, there are many apparent options for amendment.

In this way, this VST is more of a hypothesis and convenience than solute product. I do not think it is possible to encapsulate all options for something as widely applicable as the delay, and I expect that it will promote the investigation of nonlinear composition for some users. I have produced hours of recordings using algorithmic signal paths and MIDI generators. Amboea may also add the right amount of randomness to patches in linear composition.