

Ing. Franz Johan Bierschwale Tytler

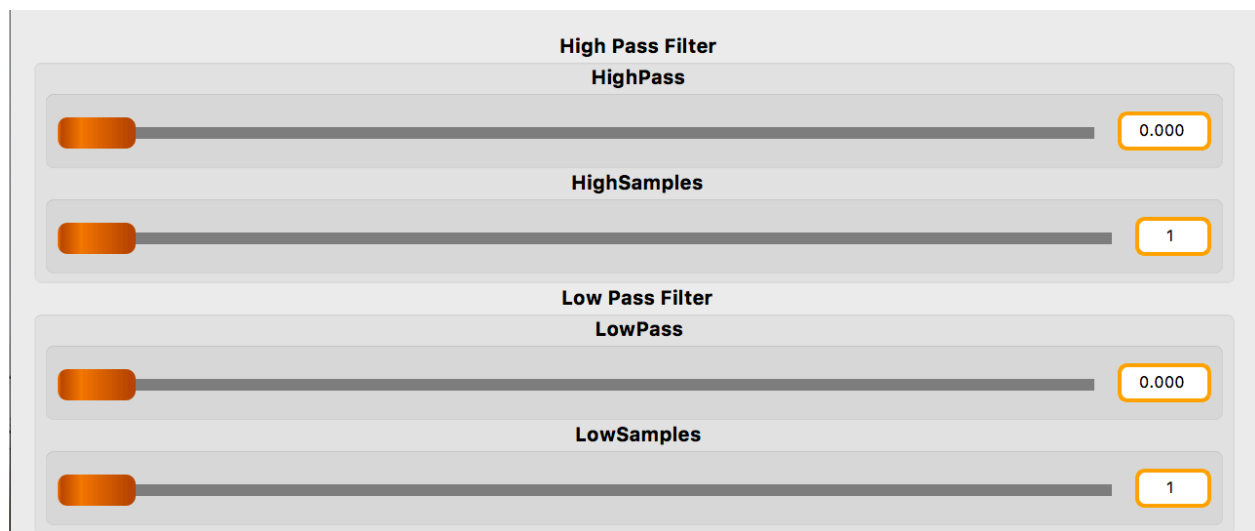
Project Name: Vox FX Station

Programmed in FAUST

Vox FX Station is a mobile effects station which uses the phones input microphone to put the original signal through a series of effects. These include:

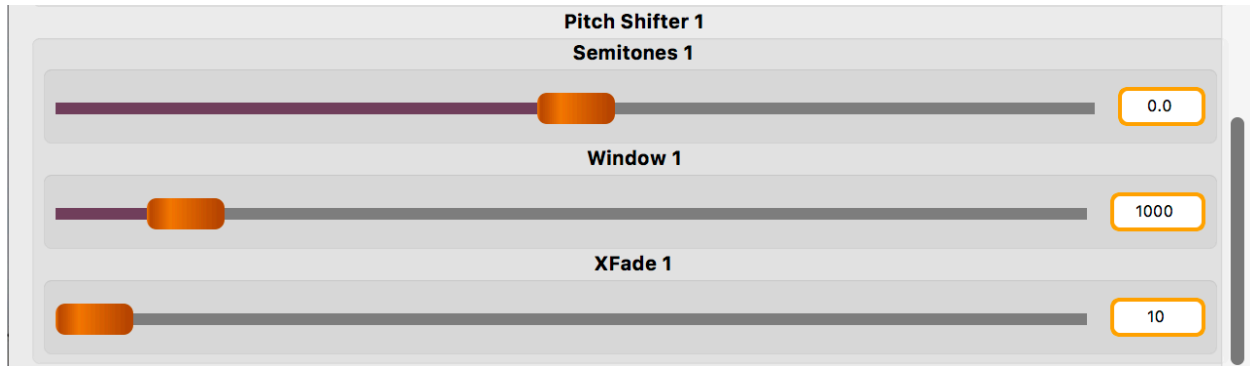
- Low Pass Filter
- High Pass Filter
- 2 octave parallel Pitch-shifter
- Reverb Design Station with a 6-band EQ adjustment
- Simple Echo Delay

Filters



Both filters were created using a OneZero Filter with an amplifier modulating the attenuation applied by said filter.

Pitch-Shifter

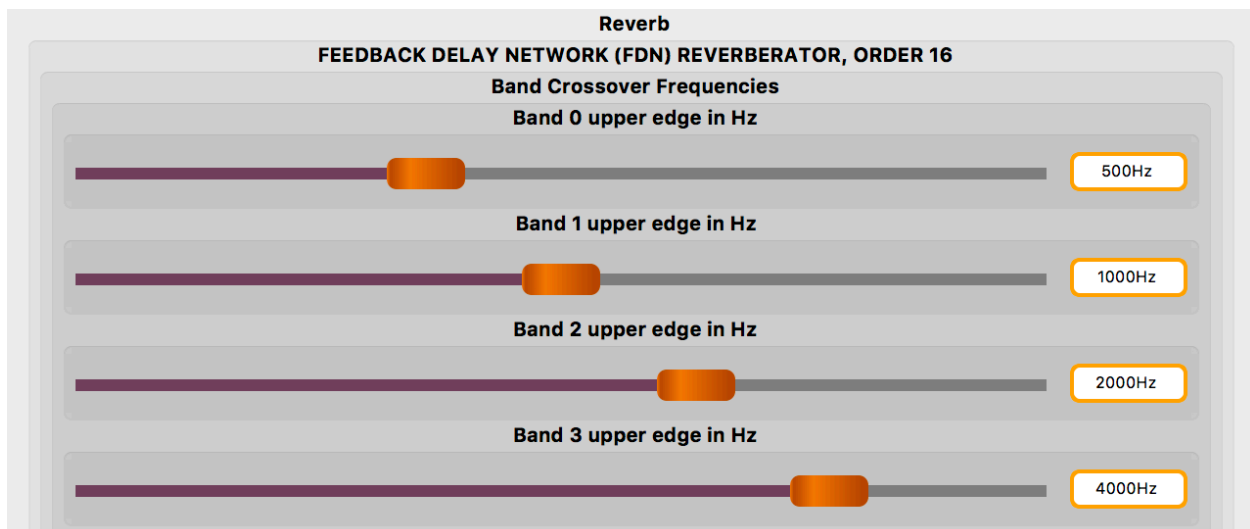


Pitchshifter used the `ef.transpose()` command and receives the order of 12 semitones above and below the root note provided by the signal. Window and X-Fade are signal delay and distortion algorithms which are achieved through merging the original signal and the processed one.

FDN Reverb Design Station

Faust has an integrated function for a complete Reverb design station which uses the Feedback Delay Network (FDN) Digital Reverberation algorithms in order to process the original signal. This method mainly consists of a 4x4 delay matrix which sets the delay times and decays for a typical feedback delay effect, creating the escalated short series of delays which build up the reverb effect.

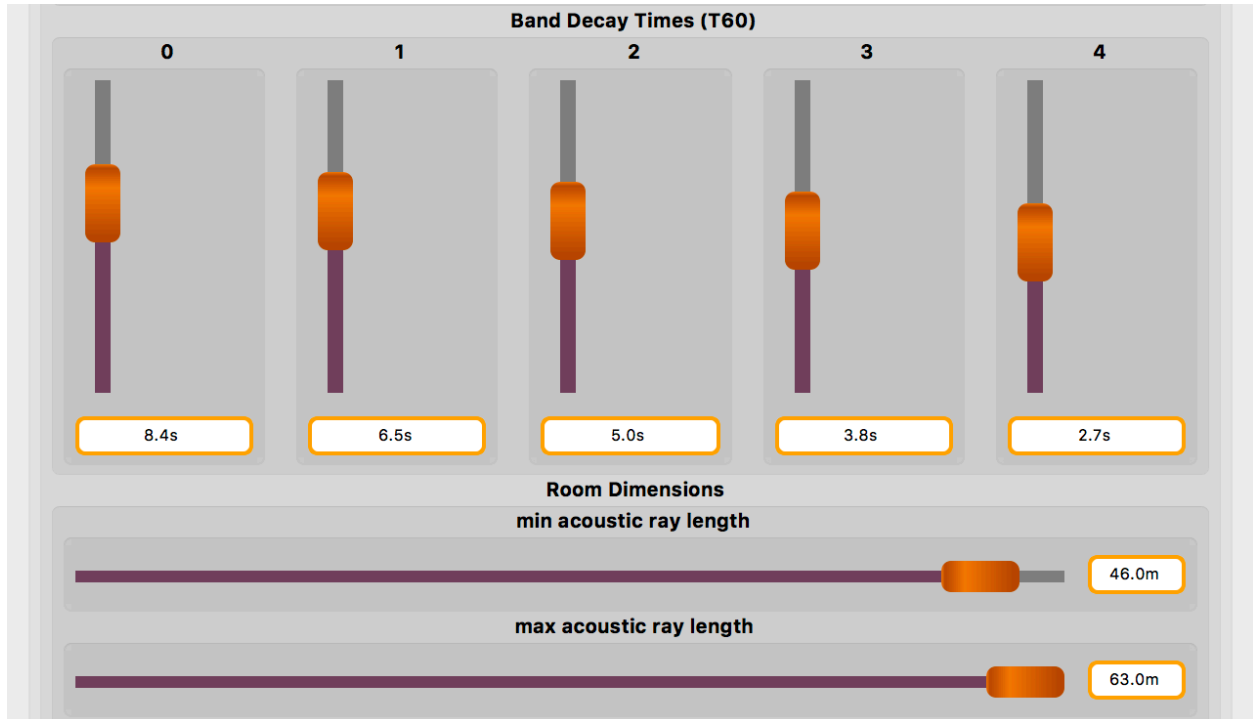
The first part of this station presents the user with a band crossover edge selector where they can fine tune the equalization of the room.



The next section allows the tweaking of $1/3^{\text{rd}}$ band decay times (in seconds) in compliance with the RT60 reverberation time defined by the ISO 3382-1 standard for performance spaces, and

the ISO 3382-2 standard for ordinary rooms. The bands go in 1/3rd octave steps from 50Hz to 8kHz.

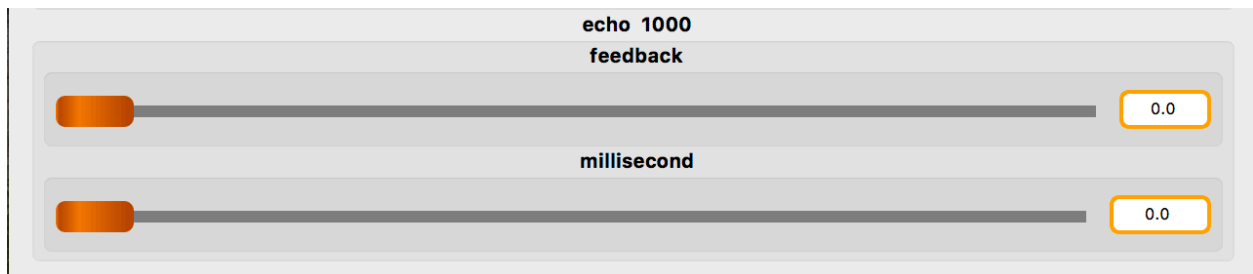
It also allows tweaking of the rooms dimension in meters with a minimum and a maximum reflection ray line.



It has a bypass option for the external input and a pink noise generation for frequency band reverberation testing, a source position selector (Lefter, center and right) and finally an output “Dry/Wet” control to mix the processed signal wieth the original.

Echo Delay

The simple echo delay effect is a feedback delay with a decay filter applied to it. The feedback is repeated through the decay filter in order to create the volume decay effect.



Code sample

```
declare name "VocalFXmodule";
declare copyright "(c) Franz Johan Bierschwale 2017";

//Library imports
import("stdfaust.lib");
import("noises.lib");
import("delays.lib");
import("oscillators.lib");
import("demos.lib");
import("misceffects.lib");
//import("music.lib");
//import("effect.lib");

//-----
//Var declaration

//PitchShifter1
Samp1 = hslider("Window 1", 1000, 50, 10000, 1);
Fade1 = hslider("XFade 1", 10, 1, 10000, 1);
Semi1 = hslider("Semitones 1", 0, -12, +12, 0.1);

//PitchShifter2
//Samp2 = hslider("Window 2", 1000, 50, 10000, 1);
//Fade2 = hslider("XFade 2", 10, 1, 10000, 1);
//Semi2 = hslider("Semitones 2", 0, -12, +12, 0.1);

//Reverb
N = 16; // Feedback Delay Network (FDN) order (power of 2, 2 to 16)
NB = 5; // Number of T60-controlled frequency-bands (3 or more)
BS0 = 3; // Order of each lowpass/highpass bandsplit (odd positive integer)

//LoPassHiPassOneZero
LoAmp = hslider("LowPass",0,0,1,0.001);
HiAmp = hslider("HighPass",0,-1,0,0.001);
LoSamp = hslider("LowSamples",1,1,100,1);
HiSamp = hslider("HighSamples",1,1,100,1);

//-----
//Process declaration

//PitchShifters
pitchshifter1 = vgroup("Pitch Shifter 1", ef.transpose(Samp1, Fade1, Semi1));
//pitchshifter2 = vgroup("Pitch Shifter 2", ef.transpose(Samp2, Fade2, Semi2));

//Reverb
VerbUI = vgroup("Reverb", fdnrev0_demo(N,NB,BS0));

//EQ
//LoPass
onezeroLo = vgroup("Low Pass Filter", (_ <: _ , (@LoSamp*LoAmp) :> _));

//HiPass
onezeroHi = vgroup("High Pass Filter", (_ <: _ , (@HiSamp*HiAmp) :> _));

//Echo
DelayUI = echo1s :> _;

//-----
process = _ <: pitchshifter1, onezeroLo, onezeroHi, VerbUI, DelayUI :> _, _;
```