Add64 Additive Synthesizer Documentation (c)2011-2019 by Matthias Nagorni

Document version 3.7.4-1

1. Introduction

Add64 implements Additive Synthesis for creating new sounds that are not accessible to any other synthesis method. Due to its large number of parameters, Additive Synthesis has never really reached much popularity. ***Add64*** is therefore designed to provide fast real-time access to all parameters. A set of two 24-stage filters has been added to map complex spectral alterations onto a set of only a few parameters. Emphasis has also been put on modulations, both in the filter as well as the spectral design sections.

Configuration settings are defined in the file ***Add64-MIDIconfig:***

```
4                         // Polyphony, set this to 0 to launch the Polyphony dialog
Add64Presets/             // Preset Folder
5                         // Noise step – glide effect for noise
system:playback_1         // Jack audio out connection
--
system:playback_2         // Jack audio in connection
--
system:midi_capture_2     // Jack MIDI in connections
system:midi_capture_3
Add64 Controller:add64-ctrl_midi_out
--
system:midi_playback_3    // Jack MIDI out connections
Add64 Controller:add64-ctrl_midi_in
--
_MIDI_       // MIDI connections – do not edit anything below this line
```

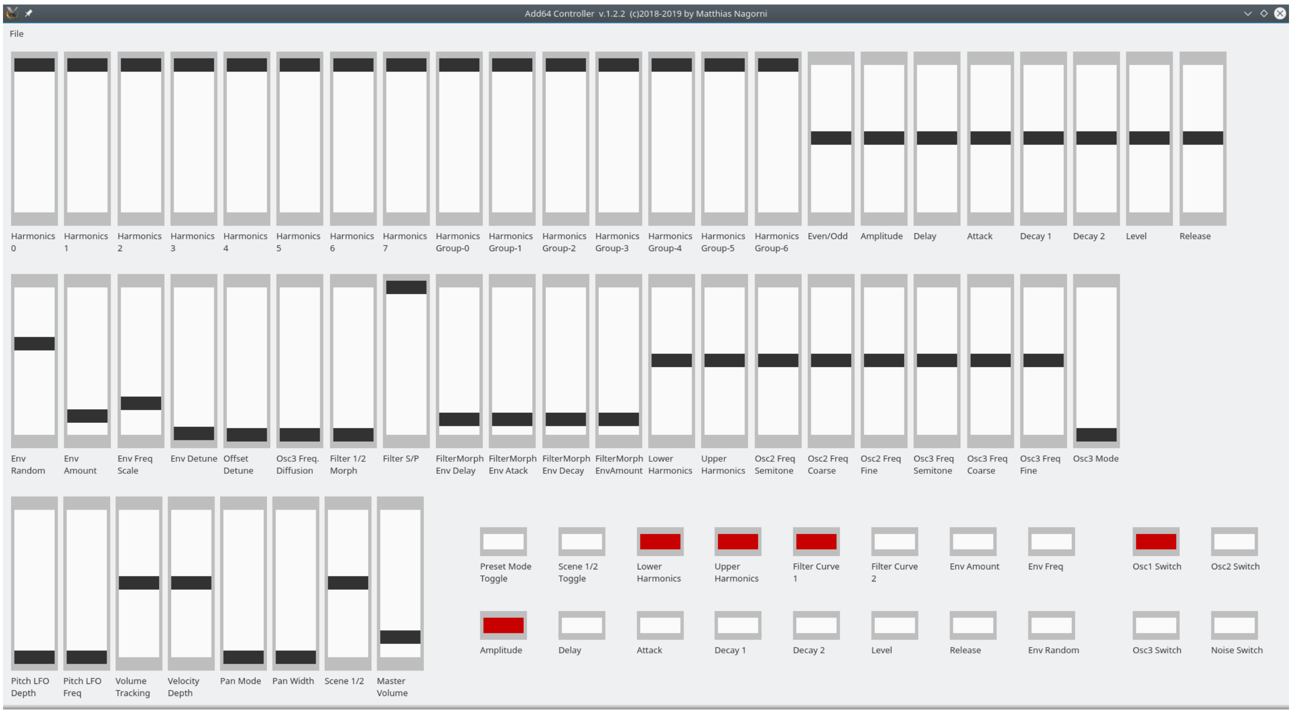


Figure 1.1 Add64-Ctrl, a virtual MIDI controller

Add64-Ctrl is a virtual MIDI controller for ***Add64***. The default MIDI configuration is matched to the respective definitions in ***Add64-Ctrl***. Since it both send and receives MIDI, it is recommended to start it first and then launch ***Add64*** so that the proper MIDI layer is selected and all controllers are initialized to the values of the initial patch.

To configure ***Add64-Ctrl*** for touch screens set

```
const bool touch = true;
```

in ***data.h***

Add64-Ctrl will send MIDI note events whenever it has keyboard focus.
The default definitions are optimized for a german keyboard – you can change this in ***Gui::getNoteFromKey***.

Add64 implements several MIDI layers which are switched and toggled by corresponding MIDI controllers. In the ***Add64-Ctrl*** Gui these are:

Preset Mode Toggle

Switch between Preset Mode (LED on) and Edit Mode. Presets are organized in Banks and Folders. In Preset Mode, the buttons „Folder Down“ and „Folder Up“ navigate through Folder 0..15. The current folder is indicated in binary form by the four LEDs of this button group.

Scene 1/2 Toggle

Switch between Scene 1 and Scene 2 (LED on). Each Scene has its own set of Harmonics and Scale settings for all oscillators and noise. The Scene 1/2 Toggle selects the scene for editing.

Lower Harmonics on/off

Turn harmonics 1-8 in the currently selected Harmonics group on/off.The active Harmonics group depends on the selected Scene and Oscillator as well as the Harmonic tab switches which select e.g. the Amplitude, Decay 1 or Release spectrum.

Harmonics - Upper Harmonics on/off

Turn harmonics in the currently selected Harmonics group on/off.

Filter Curve 1

Select Filter Curve 1 for editing (LED on). Note that the Filter Mix (22) is set to 50% per default. For the filter to be fully effective, the Filter Mix has to be set to maximum for both filters.

Filter Curve 2

Select Filter Curve 2 for editing (LED on).

Amplitude Switch

Select the Amplitude spectrum for editing.

Delay Switch

Select the Delay spectrum for editing.
The Delay spectrum allows to delay some harmonics in the sound.

Attack Switch

Select the Attack spectrum for editing.

Decay 1 Switch

Select the Decay 1 spectrum for editing.
The envelope will decay to the level set with the Level parameter

Decay 2 Switch

Select the Decay 2 spectrum for editing.

Level Switch

Select the Level spectrum for editing.

Release Switch

Select the Release spectrum for editing.

Env Amount Switch

Select the Env Amount spectrum for editing with controllers. This is the amount of LFO/Random Envelopes which are multiplied on top of the Amplitude spectrum.

Env Freq Switch

Select the Env Frequency spectrum for editing. This is the base frequency for the LFO/Random Envelopes. A random component is added to the LFO frequency based on the Env Random spectrum and the Env Random Scale settings.

Env Random Switch

Select the Env Random spectrum for editing with controllers. This is the amount of the random component that is added to the regular LFO frequency.

Osc 1 Switch

Select oscillator 1 for editing.

Osc 2 Switch (106)

Select oscillator 2 for editing.

Osc 3 Switch (107)

Select oscillator 3 for editing.

Noise Switch (108)

Select Noise for editing.

2. Architectural Overview

Add64 features three spectral oscillators and a spectral noise generator. These sound generators are organized in two scenes. Each oscillator can be sent to a set of two filters which feature extensive routing and modulation features.
The user interface is organized in numerous tabs. It follows a natural signal flow scheme of Oscillators → Mixer → Filters → Main. The ***Harmonics*** tab provides access to the spectra of the oscillators and their spectral envelopes. Amplitude and envelope spectra can be selected via the tabs at the bottom of the ***Harmonics*** window.

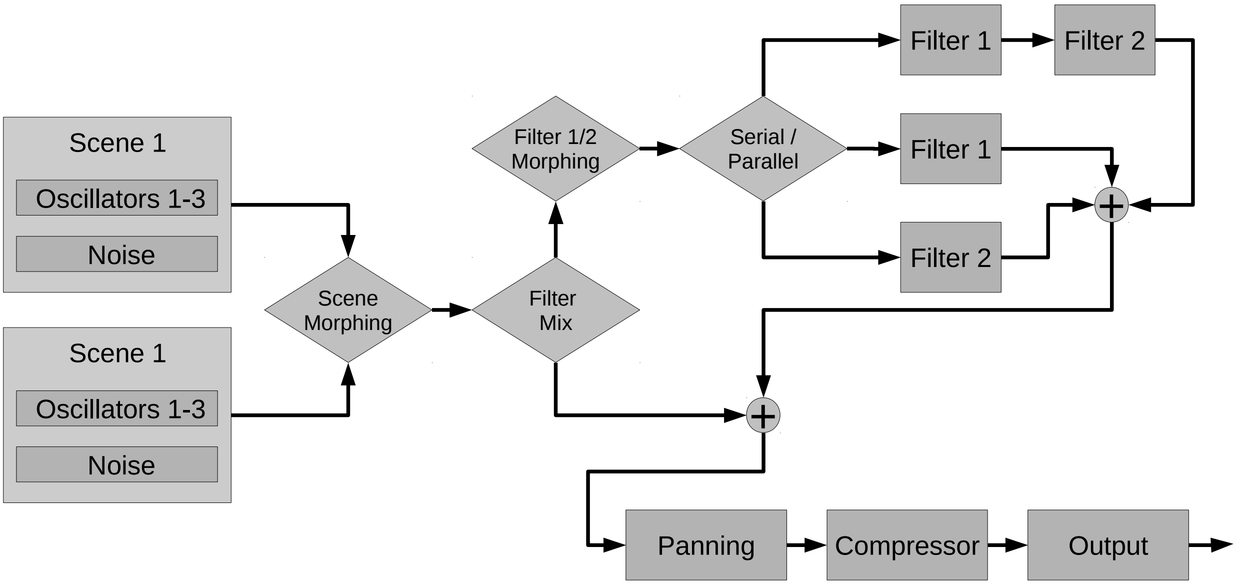
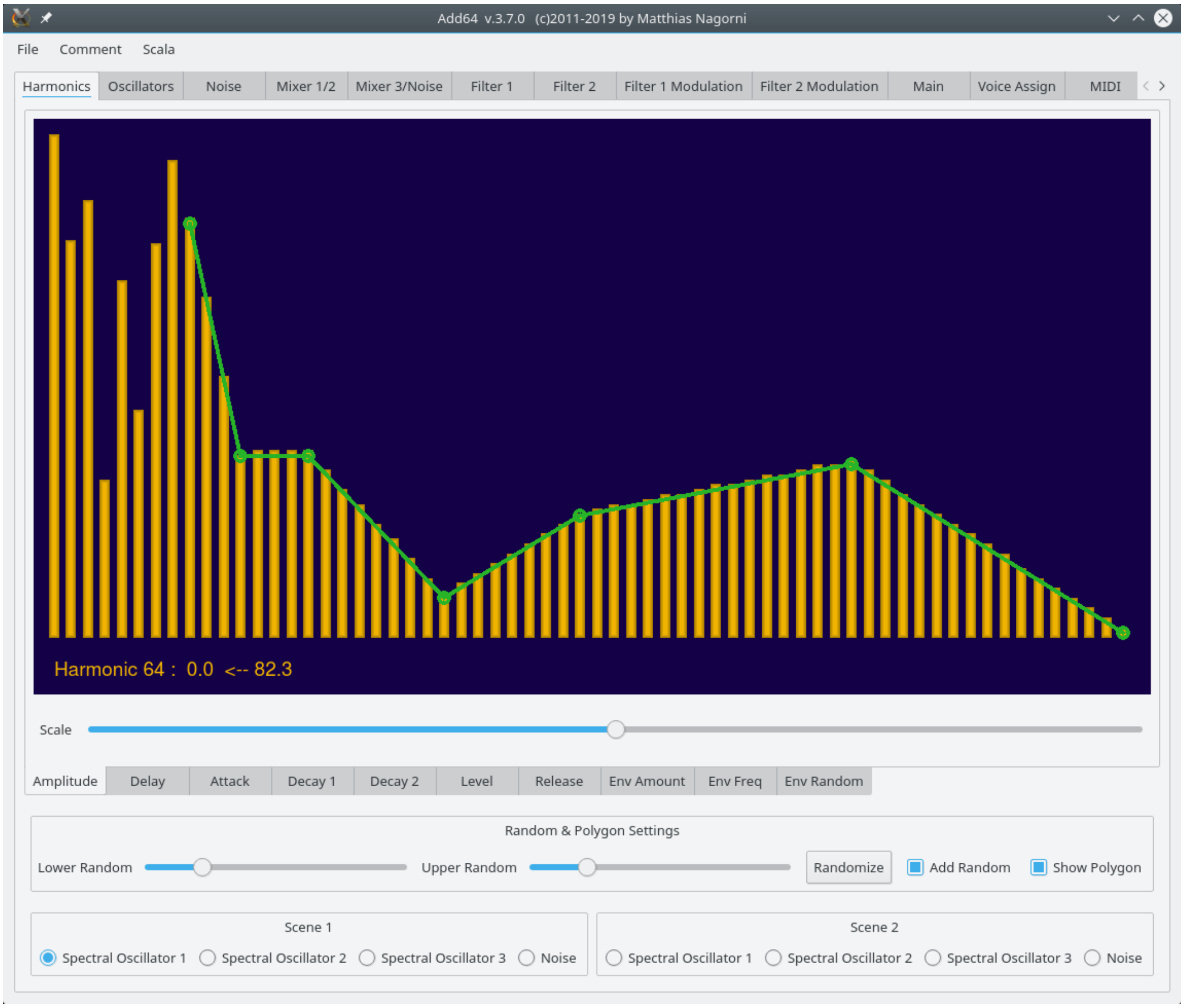


Figure 2.1 Add64, architecture and signal flow.

3. Harmonics

This tab allows to define the spectra for the oscillators and noise. Each harmonic has an amplitude parameter as well an envelope with attack, delay, double decay and release. The amplitude can further be modulated by an envelope LFO spectrum. The ***Scale*** slider allows to scale each spectrum by a constant factor. To generate any sound you need to at least draw a spectrum for ***Amplitude*** and ***Decay 1*** for at least one of the oscillators. Oscillators are selected using the radio buttons at the bottom of the window. Different spectra can be defined for Scene 1 and Scene 2 where the mix between these sounds is controlled by the ***Scene Morphing*** parameter in the ***Main*** tab (see section 9.).

Note that only ***Spectral Oscillator 1*** features an ordinary harmonic spectrum while ***Spectral Oscillator 2/3*** have variable spacing between harmonics which you can use to create metallic and other inharmonic sounds. See section 4. for more details on the relevant parameters.



Harmonic 64 : 0.0 <-- 82.3

☑ Add Random ☑ Show Polygon

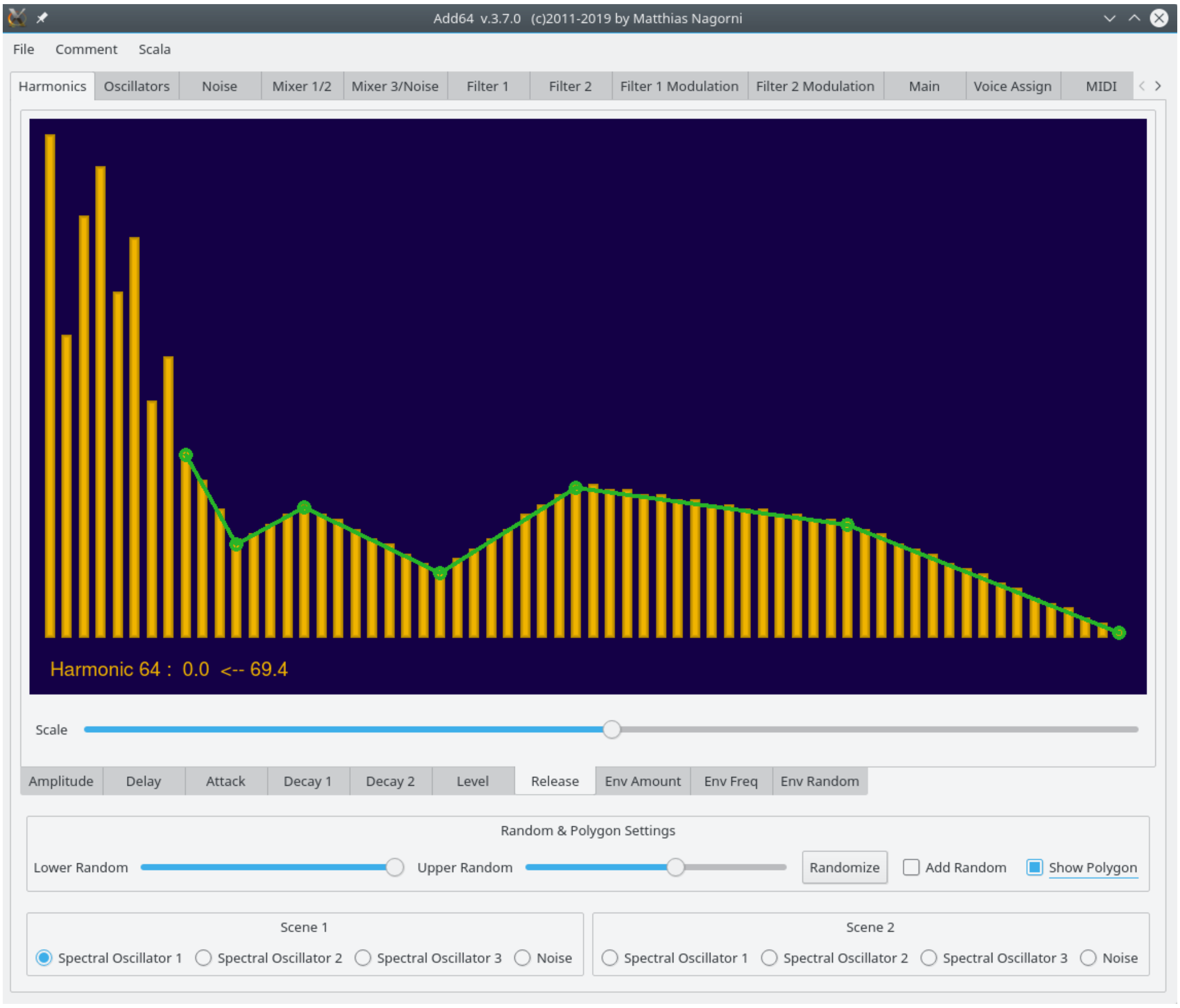
Scene 1: ● Spectral Oscillator 1 ○ Spectral Oscillator 2 ○ Spectral Oscillator 3 ○ Noise

Scene 2: ○ Spectral Oscillator 1 ○ Spectral Oscillator 2 ○ Spectral Oscillator 3 ○ Noise

Figure 3.1 A complex amplitude spectrum

If the checkbox ***Show Polygon*** is checked, the upper harmonics follow the polygon that is defined by the corresponding control points. Upper (9-64/32) and lower harmonics (1-8) can also be randomized separately. If the checkbox ***Add Random*** is checked, the random values are added to the current spectrum. Otherwise the spectrum is overwritten with random values whenever ***Randomize*** is clicked.

A double decay envelope, initially proposed by Fons Adriaensen for *AlsaModularSynth*, approximates the envelope of an acoustic instrument like piano, harpsichord or any other plucked string much better than the usual ADSR-Evelope with its static sustain level. ***Decay 1*** corresponds to the initial fast decay of the sound after the hammer hits the string or it is plucked. The ***Level*** parameter defines the level for the transition between ***Decay 1*** and ***Decay 2***. For large values of ***Decay 2*** the ***Level*** parameter is close to the sustain in traditional ADSR envelopes. Using the ***Delay*** parameter, some harmonics can be delayed relative to the start of the sound. This effect is found for example in large organ pipes, where the fundamental is delayed relative to higher harmonics. While in real acoustic instruments the delay of harmonics is usually a subtle effect, it offers an exciting playground for new electronic sounds. Especially pad sounds can get surprising dynamic components when some of their harmonics are delayed. It is fun to experiment with the ***Delay*** parameter at various levels of ***Scale.***



Harmonic 64 : 0.0 <-- 69.4

☐ Add Random ☑ Show Polygon

Scene 1

● Spectral Oscillator 1 ○ Spectral Oscillator 2 ○ Spectral Oscillator 3 ○ Noise

Scene 2

○ Spectral Oscillator 1 ○ Spectral Oscillator 2 ○ Spectral Oscillator 3 ○ Noise

Figure 3.2 According to this complex release spectrum, the sound will change colour when the key is released.

The ***Delay*** spectrum can also be combined with an ***Attack*** spectrum so that some harmonics will enter the sound more smoothly. Interesting effects can also be achieved with the ***Release*** spectrum. The ***Release*** parameter defines how rapidly a certain harmonic will fade out after the key has been released. A nonuniform spectrum as depicted in **Figure 3.2** will lead to significant changes in sound colour in the release phase since some harmonics will decay much faster than others. Note however that the release does not introduce new harmonics. Each harmonic will just decay from the level it had when the key was released.

In addition to the envelope, the amplitude of each harmonic can be modulated by a Low Frequency Oscillator. This is an advanced feature that you can safely ignore for your first experiments. Its purpose is to add further dynamics to sustained sounds. The level of LFO modulation can be set in the ***Env Amount*** tab. The frequency is defined in ***Env Freq***. The LFO frequency can be randomized in the ***Env Random*** tab. As a first experiment you can create a shimmering pad sound by setting all harmonics in the ***Amplitude, Decay 1 Env Amount*** tabs to maximum and then creating some nonuniform ***Env Freq*** spectrum. This way, the harmonics of the sound will fade in and out at different frequencies leading to an ever-changing sound colour. The periodic LFO can be randomized by defining a spectrum in the ***EnvRandom*** tab.

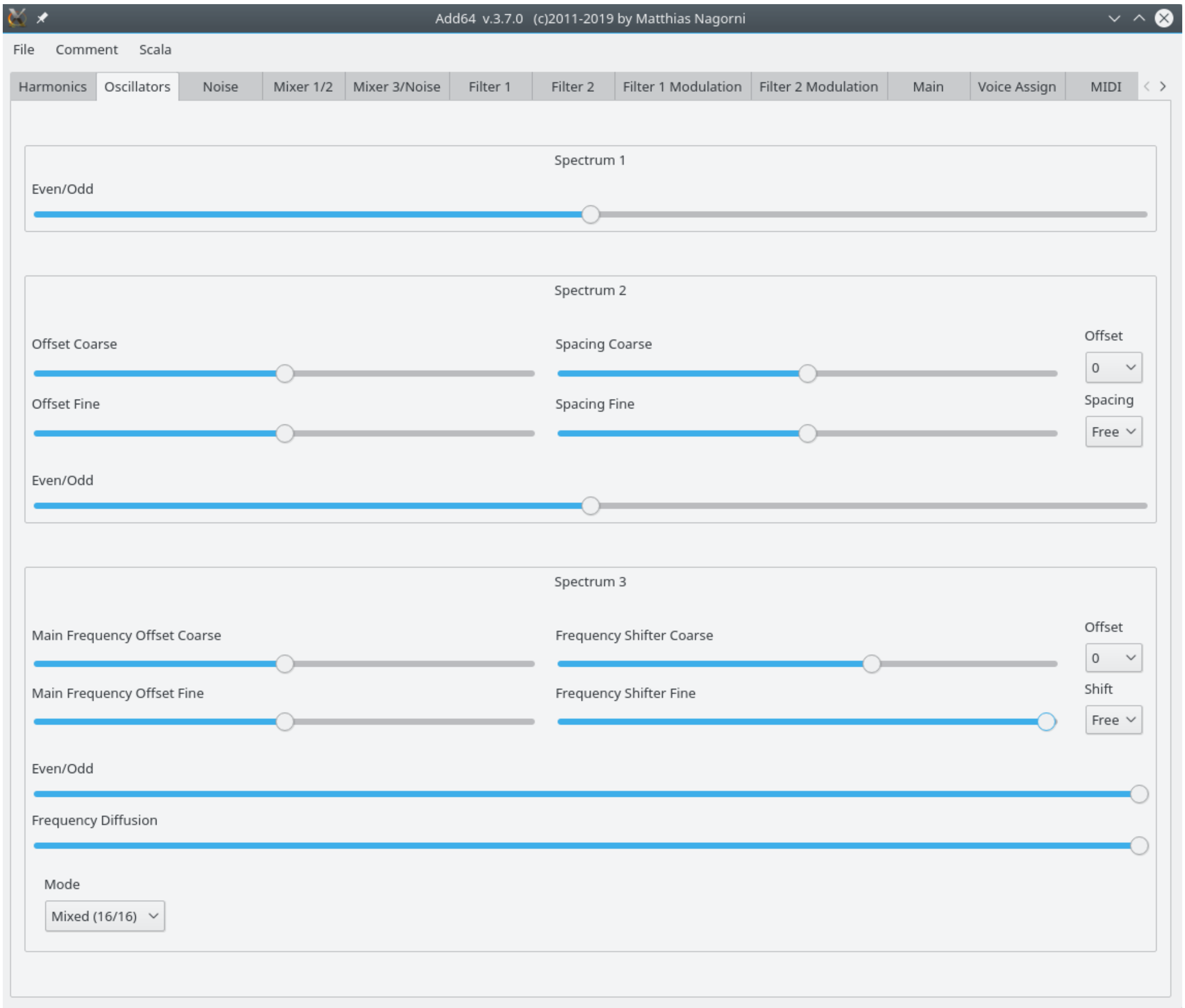
At first glance the many parameters in the ***Harmonics*** tab may seem a bit discouraging. You will however soon discover that it is relatively straightforward to create innovative sounds with ***Add64*** since most parameters can safely be set to their maximum level as a starting point. Refinements can then either be made in the spectrum itself or by using the parameters in the ***Oscillators*** and ***Filters*** tabs.

4. Oscillators

Spectral Oscillator 1 features an ordinary harmonic spectrum. The ***Even/Odd*** parameter allows to emphasize the even and odd harmonics. ***Spectral Oscillator 2*** can have variable spacing between harmonics. This leads to inharmonic sounds somewhat similar to ring modulation or a frequency shifter. The spectra can also be shifted using the ***Offset*** parameter. Both the ***Spacing*** and ***Offset*** parameters can be adjusted with the Coarse and Fine sliders. In addition, two selector boxes are provided for easy access to integer values. In these selectors, ***Offset*** 1 and ***Spacing*** 1.0 will lead to an ordinary harmonic spectrum. ***Spacing*** 2.0 will double the spacing between harmonics so that harmonic 2 corresponds to the 3rd harmonic of the harmonic series. The ***Offset*** parameter shifts the harmonics within the harmonic spectrum. At ***Spacing*** 1.0 an ***Offset*** of 3 will shift harmonic 1 to the 3rd harmonic of the harmonic series.

The spectrum of ***Spectral Oscillator 3*** can be shifted using the ***Frequency Shifter*** sliders. In addition, the frequency of each harmonic can be modulated by the corresponding envelope LFO, as specified in the ***Env Amount / Freq / Random*** tabs. The parameter ***Frequency Diffusion*** specifies the level of modulation. ***Spectral Oscillator 3*** features 4 different modes: Harmonic (n*f0), Subharmonic (f0/n) and two mixed modes with both Harmonics and Subharmonics.

Note: While Spectral Oscillators 1/2 use a recursive scheme to derive the spectrum from few sine/cosine computations, Spectral Oscillator 3 features up to 32 independend sine oscillators. These computations are quite CPU-intensive. Therefore all oscillators will calculate their spectrum only up to the harmonic which has the highest non-zero amplitude.



Offset 0

Spacing Free

Offset 0

Shift Free

Mode Mixed (16/16)

Figure 4.1 The *Oscillators* tab has parameters for the balance of even and odd harmonics. For oscillators 2 and 3, offset and spacing of the harmonics can be specified.

5. Noise

Noise is an important part of many acoustic sounds, like for example organ pipes and wind instruments. ***Add64*** features a noise generator which allows to create complex noise sounds based on the noise spectrum defined in the ***Harmonics*** tab. The noise spectrum is generated by passing white noise through a set of bandpass filters. When the ***Offset*** and ***Spacing*** sliders are in their center positions, the center frequencies of the bandpass filters follow an ordinary harmonic series. As in the oscillators, the ***Even/Odd*** parameter allows to balance even and odd harmonics. The ***Bandwidth*** parameter controls the width of the bandpass filter curve. If the bandwidth is large, the sound will be noisy, if it is small, the noise generator will almost sound like an oscillator. Note that in this case the bandpass filters will sound quite percussive if large intervals are played. A glide effect on noise pitch minimizes this. The corresponding ***Noise step*** parameter is set in ***Add64-MIDIconfig***. Since this is a step parameter, smaller values increase the glide effect.

To get started with the noise generator, you need to define at least the ***Amplitude*** and ***Decay 1*** spectra in the ***Harmonics*** tab (make sure the ***Noise*** radio button is selected).

As a first experiment you could set both spectra to uniform maximum values and then see what happens if you set the noise ***Bandwidth*** parameter to its minimum value. If you now also increase the ***Spacing*** parameter, the sound will get pretty spooky. As a next step you could play with the ***Attack*** parameter in the ***Harmonics*** tab so that some noise components will enter the sound more smoothly. If you remember that all envelope features of the ***Harmonics*** tab are also available for the noise generator, you can imagine how complex the noise sounds can get. The noise spectrum can also be set to subharmonic mode by checking the corresponding checkbox.

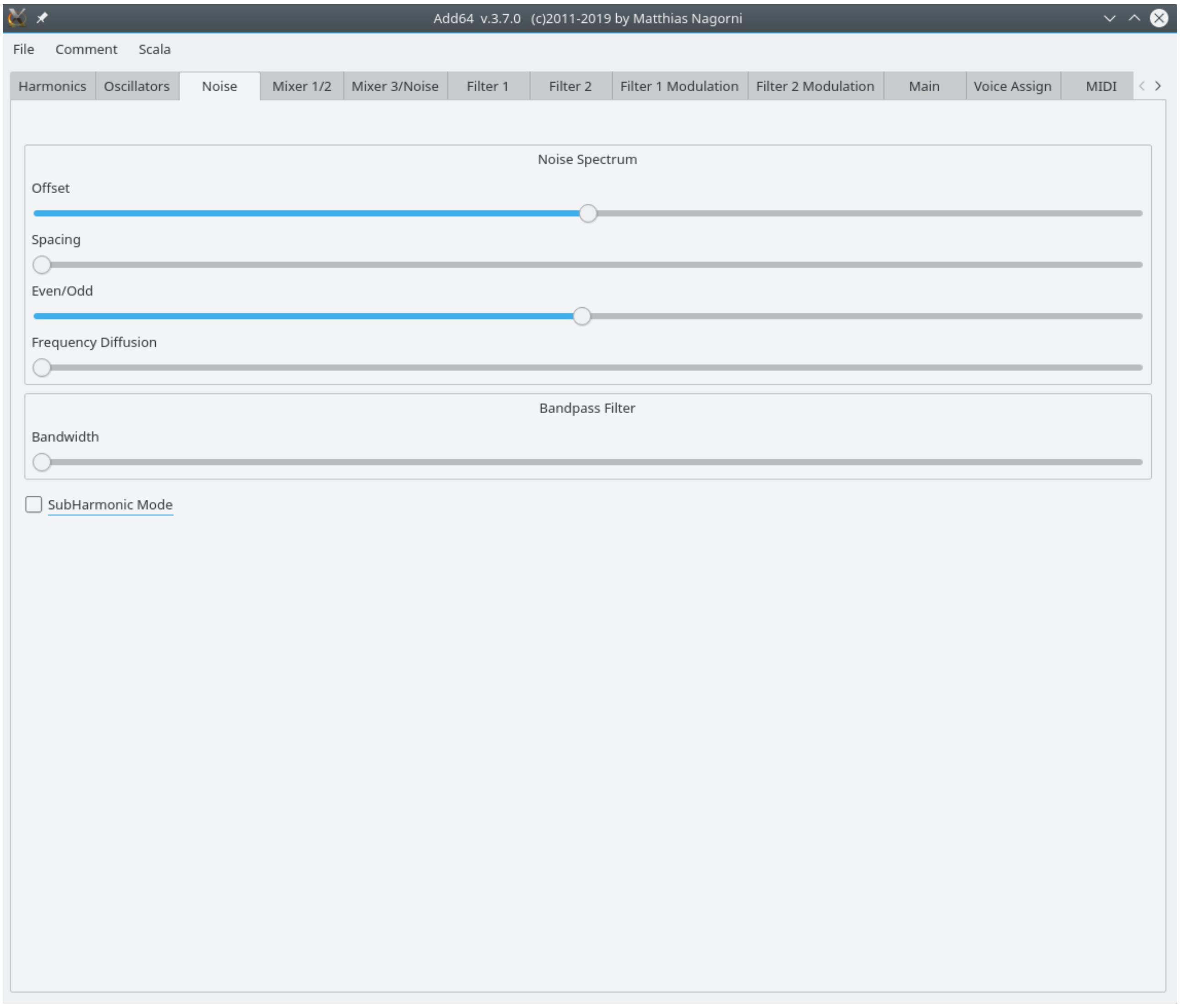


Figure 5.1 The *Noise* tab.

6. Mixer

If you used ***Add64*** for pure additive synthesis you could stop reading here. However the sound spectra defined in the ***Harmonics*** tab can also be sent through a set of two filters (see ***Figure 2.1***). The exact routing is defined in the ***Mixer*** tab. ***Filter Morph*** defines the morphing between Filter 1 and Filter 2. The ***Serial/Parallel*** parameter balances the amount of signal that is sent through both filters applied in series as well as to both filters in parallel arrangement. Note that each filter also has a ***Mix*** slider in the ***Filter 1/2*** tab. The ***Mix*** parameter defines the balance between filtered and raw signal. If ***Mix*** is set to maximum, the entire input signal is sent through the filter.

The filter morph can be modulated with a simple AD envelope. If ***LFO Mode*** is on, the AD envelope will oscillate as LFO where the rising edge is defined by ***Attack*** and the falling edge by ***Decay***. If ***Hold*** is on, the ***Delay*** parameter will also define the hold time. If ***LFO Retrigger*** is on, the LFO will restart on each new note.

You may be missing a way to balance the oscillator levels in the ***Mixer*** tab. Remember however that amplitude levels of the oscillators are defined in the ***Harmonics*** tab where you can use the ***Scale*** slider below the ***Amplitude*** spectrum to set the overall level of an oscillator. This slider can also be bound to a MIDI hardware controller (see section 11.).

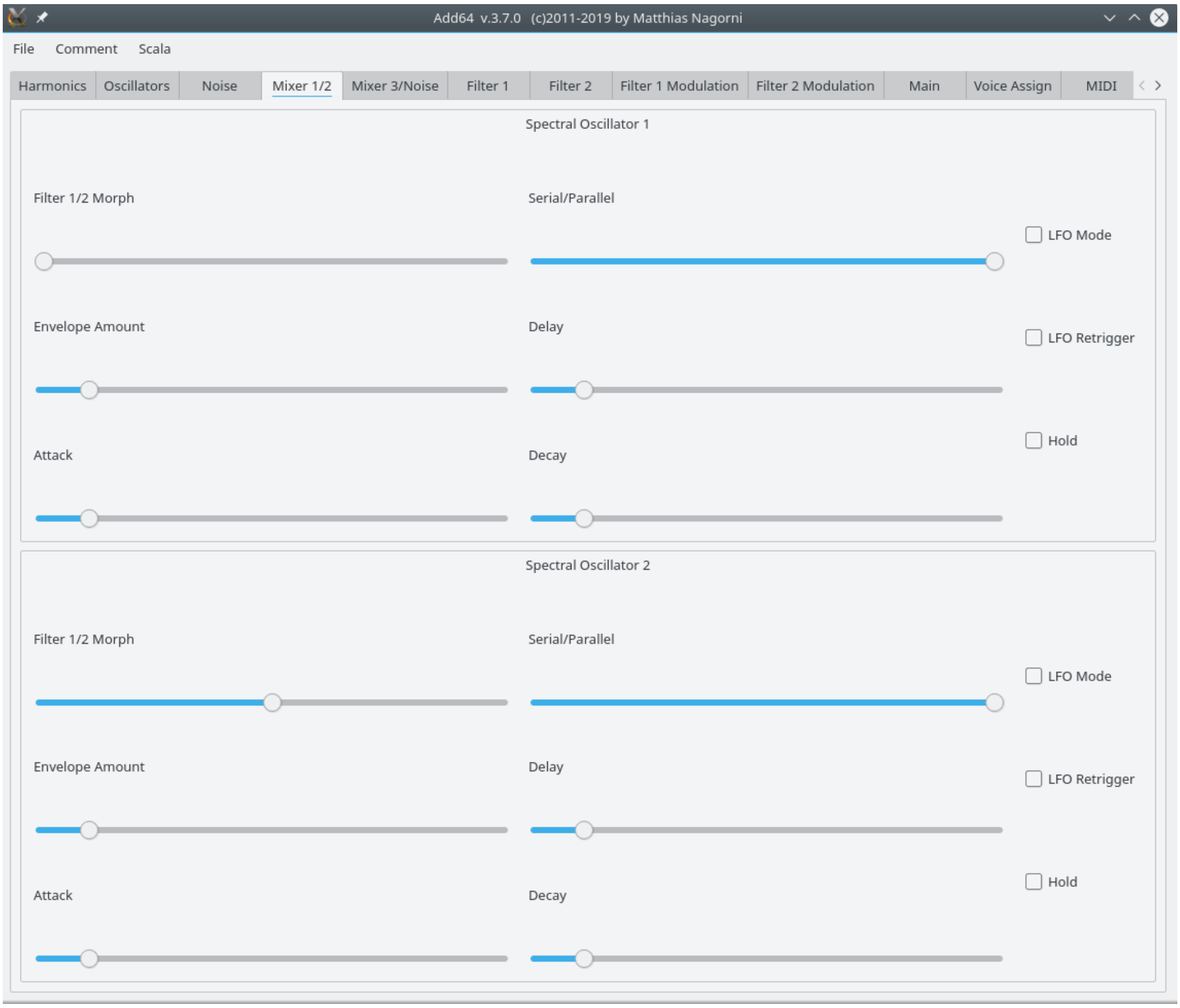
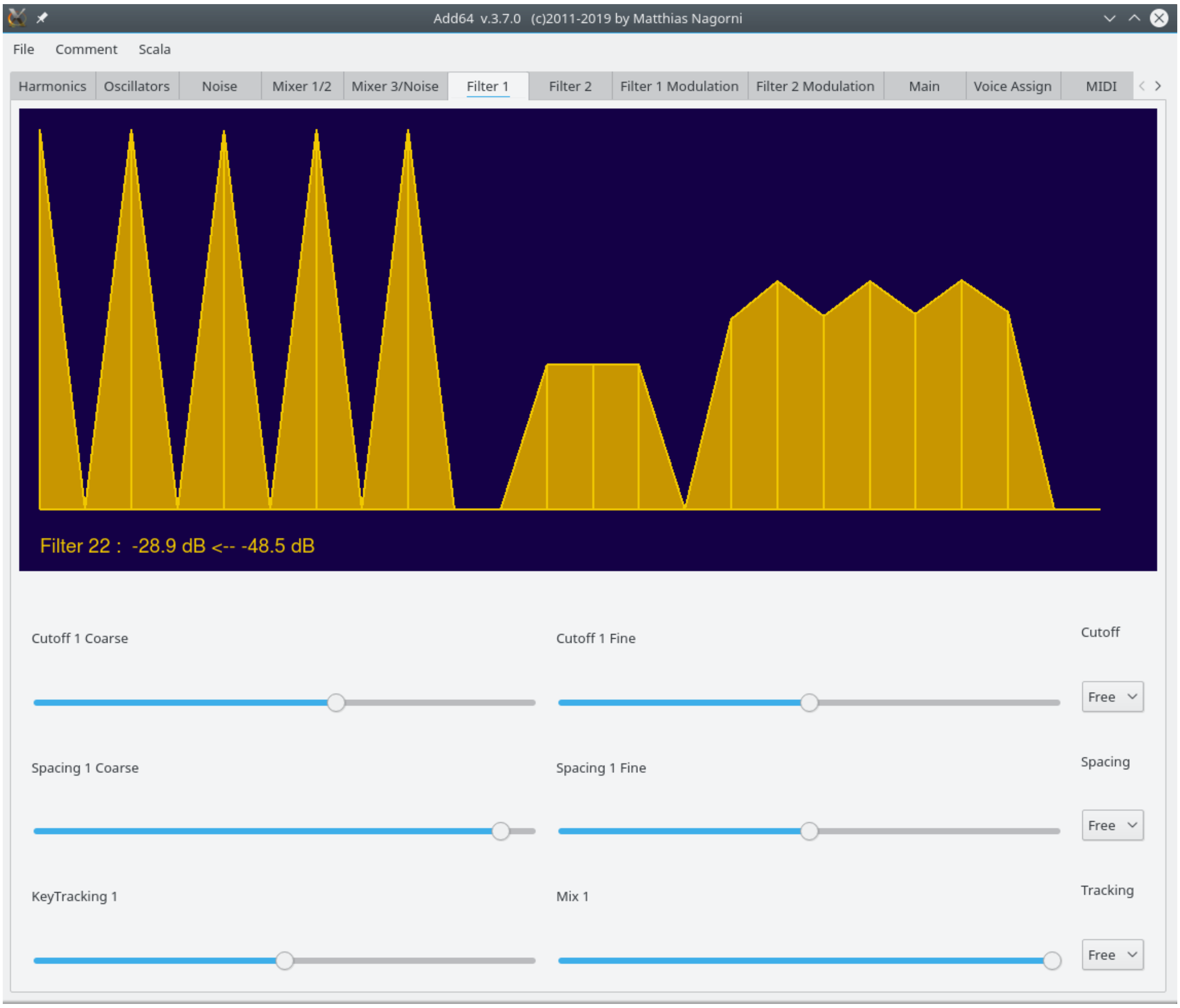


Figure 6.1 The filter routing parameters are found in the *Mixer* tab. The AD envelope has effect on the *Filter Morph* parameter.

7. Filters

The power of subtractive synthesis lies in the fact that the sound depends only on a small set of parameters, such as filter cutoff, filter resonance, filter envelope amount and a few others for oscillators and their relative frequencies and balance. In contrast, additive synthesis provides much more flexibility in sound design at the prize of hundreds of parameters where many of them have only a subtle effect on the sound. To combine the best of the two worlds and also provide further dynamic elements to the sound,

Add64 features two 24 step filters. This allows to graphically define almost arbitrary filter shapes.
The ***Mix*** parameter determines the balance between input signal and filter output. If it is set to maximum, the sound will entirely be shaped by the filter. The default value for the ***Mix*** parameter is different for filter 1 and 2. This is to ensure that you can use ***Add64*** as a pure additive synthesizer without paying attention to the filter tabs. The spacing between filter points is either defined with the ***Spacing Coarse / Fine*** sliders or the ***Spacing*** selector. The latter sets the spacing to half tones while adjusting the sliders correspondingly. A similar set of sliders and selector is used for the ***Cutoff*** parameter.
Key Tracking defines how much the filter cutoff is shifted based on the played note. At its maximum value of 1.0, the filter will exactly follow the pitch so that the sound is independend of the played note.



Add64 v.3.7.0 (c)2011-2019 by Matthias Nagorni

Filter 22 : -28.9 dB <-- -48.5 dB

Cutoff 1 Coarse

Cutoff 1 Fine

Cutoff: Free

Spacing 1 Coarse

Spacing 1 Fine

Spacing: Free

KeyTracking 1

Mix 1

Tracking: Free

Figure 7.1 This complex filter combines a lowpass with phaser and formant elements

The first and last elements of the filter are special in that their level is also applied to all frequencies below and above the limit. The first element is thus a low pass filter, the last element a high pass.
The easiest way to play with the filters is as follows: Go to the ***Harmonics*** tab and set all amplitudes of oscillator 1 as well as its decay 1 to maximum. In addition set the level slider

for decay 1 to maximum. Then set ***Mix 1*** in ***Filter 1*** to maximum and draw some filter. Obviously, this way you will end up with pure subtractive synthesis.

8. Filter Modulation

A filter envelope as well as LFOs for Cutoff and Spacing are found in the ***Filter Modulation*** tabs. Each filter has its own modulation sources and the filter mix can be modulated as well (see section 6.) so that you get a total of 5 LFOs for the filters. Remember that each harmonic has an individual LFO as well (see section 3.) so that there a few limits for creating dynamic and shimmering pads with ***Add64***.

As usual, the filter envelope has effect on the cutoff frequency. It is a double decay envelope where the maximum value of ***Decay 1*** is shorter than the maximum of ***Decay 2***. The ***Level*** parameter determines the level at which the decay 1 curve transitions into the decay 2 curve. If it is set to maximum, ***Decay 1*** will not have any effect and the decay depends only on ***Decay 2***. If the ***Inverse*** toggle is on, the envelope will be subtracted from the cutoff frequency instead of being added to it.

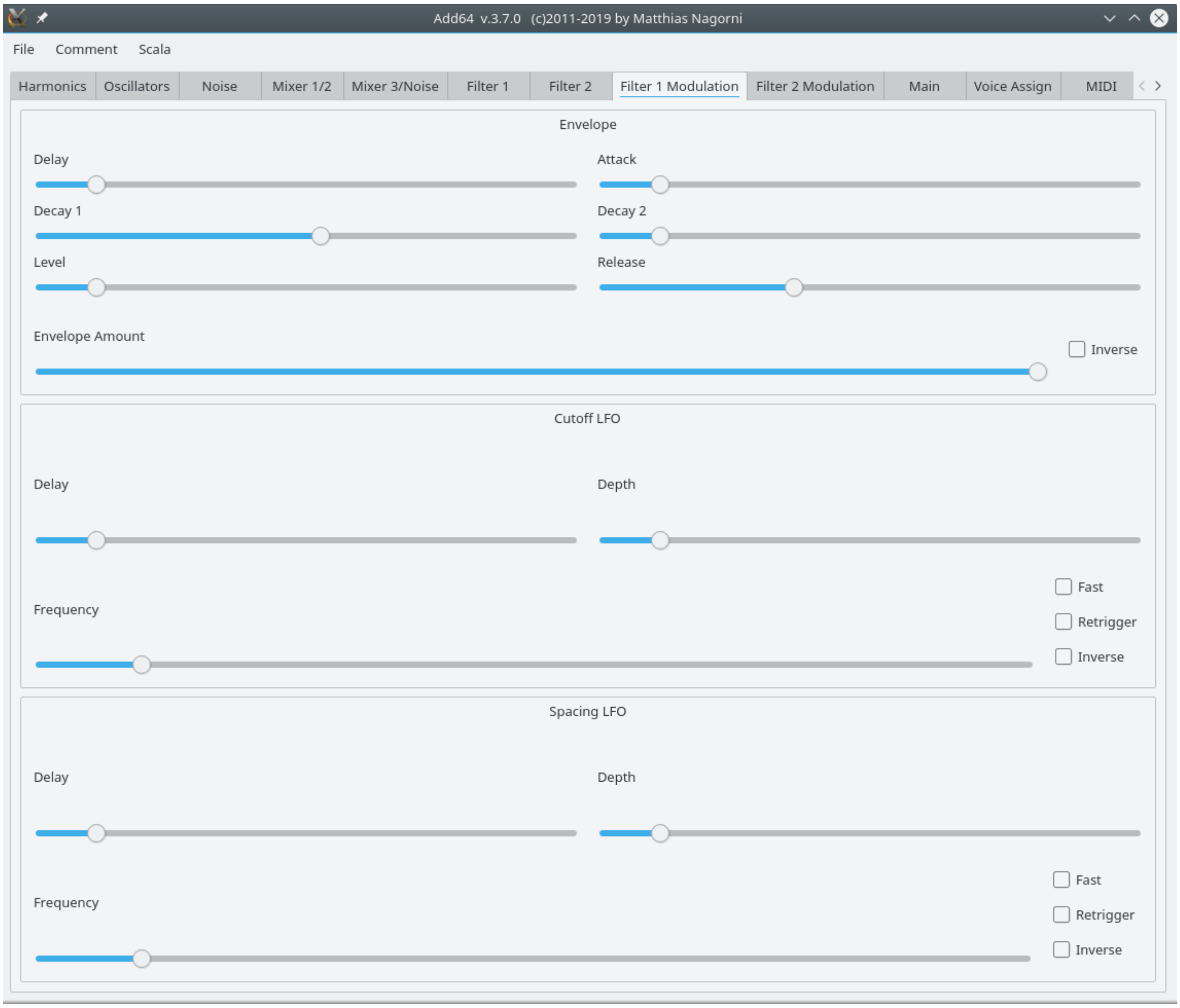


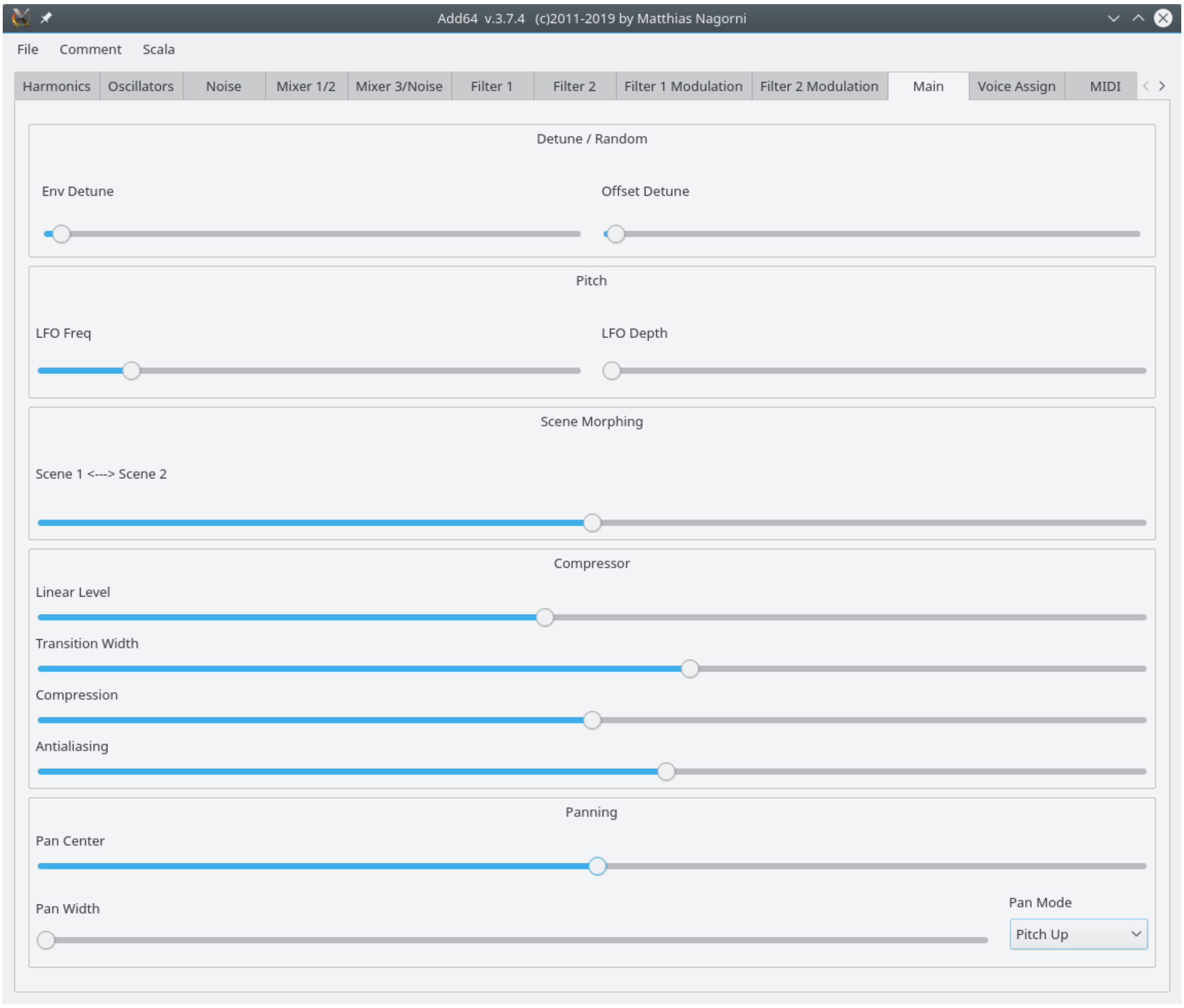
Figure 8.1 The *Filter Modulation* tab with parameters for filter envelope and LFOs for Cutoff and spacing of the filter points.

The LFOs for Cutoff and Spcaing have a similar set of parameters. Each LFO can be delayed and its amplitude is defined by ***Depth***. The frequency range can be boosted with the ***Fast*** toggle. If ***Retrigger*** is on, the LFO will be restarted whenever a new note is played.

9. Main

Detune features, panning, scene morphing as well as a simple compressor are found in the ***Main*** tab. Note that the ***Master Volume*** slider has been moved to the ***Volume*** tab where ***Volume Tracking*** defines the dependency between volume and pitch and ***Velocity Depth*** specifies the volume change based on MIDI velocity.

Env Detune adds pitch modulation based on the envelope LFO spectra. For each spectral oscillator a separate modulation is applied, based on the sum of the respective env modulation LFO. ***Offset Detune*** adds a constant random pitch offset for each note.



Pan Mode
Pitch Up

Figure 9.1 The *Main* tab has various parameters for detune, panning, scene morphing and the compressor.

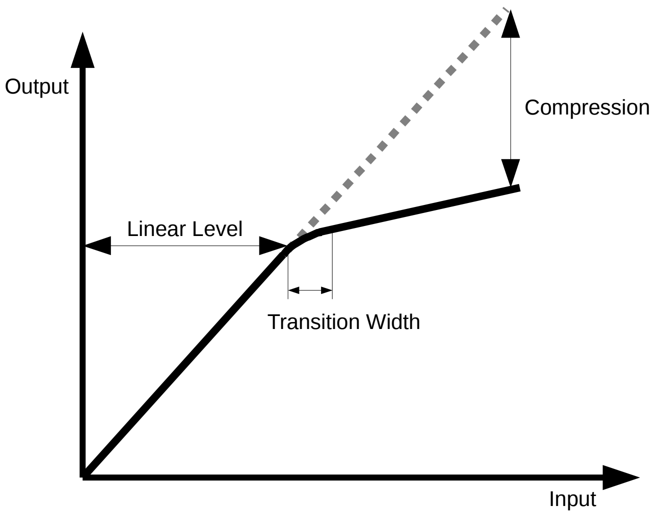


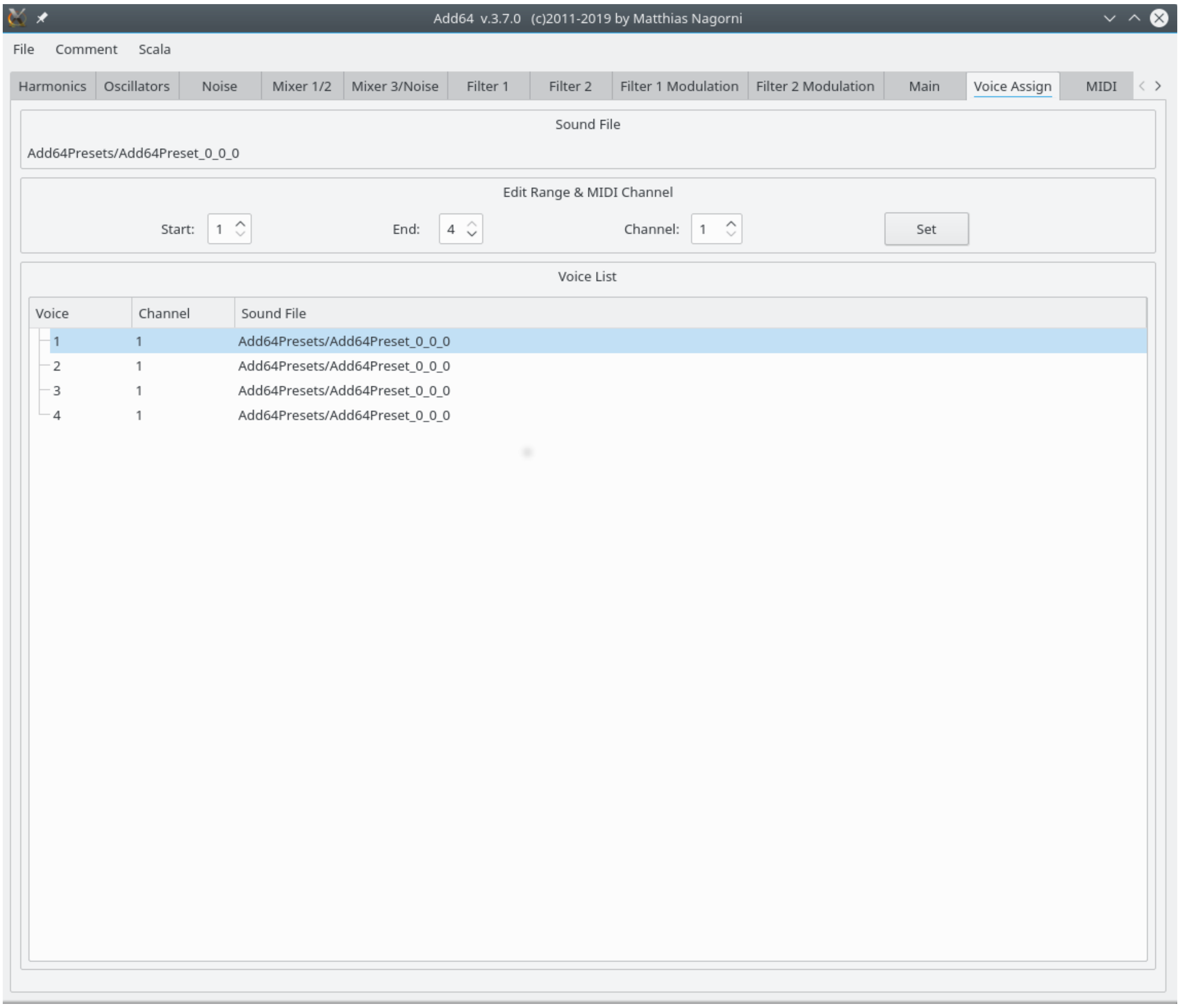
Figure 9.2 The compressor curve and its parameters.

The parameter ***Scene Morphing*** defines the balance between the Scene 1 and Scene 2 spectra as defined in the ***Harmonics*** tab. This can not only be used for simple morphing via MIDI knob. You can also connect ***Scene Morphing*** with the MIDI velocity or MIDI note events (see section 11.).

While the ***Compressor*** initially has just been added to prevent clippling, it can also be used to experiment with combinations of filters and distortion. ***Linear Level*** is the level up to which this stage has a neutral gain of 1. Input signals above this level will experience a gain below 1. The higher the ***Compression*** the lower the gain. The linear sections of unit gain and compression are connected by a continuous transition (**see Figure 9.2**) where ***Transition Width*** defines its width. Since the compression adds higher harmonics to the sound, you may need to use the parameter ***Antialiasing*** which controls to which extend higher harmonics are muted as you play in the upper region of the keyboard. Note that ***Add64*** is free of aliasing for any signal below the ***Linear Level*** threshold since harmonics are only added up to the Nyquist frequency.

10. Voice Assign

Add64 is multitimbral. In the ***Voice Assign*** tab you can assign sound files to a range of voices and define the associated midi channel. This is how it works: first, you define start, end and MIDI channel for each range of voices. When you click ***Set***, the changes will take effect in the ***Voice List***. Once the ranges are defined, you can assign sound files to them by selected a range and then loading the file with ***File*** → ***Load Sound***. The parameters in the ***Harmonics*** and other tabs will always reflect the sound which is currently selected in the ***Voice List***. If you select a different voice range in the list, you will be asked if you would like to save the current sound.



Sound File

Add64Presets/Add64Preset_0_0_0

Edit Range & MIDI Channel

Start: 1 End: 4 Channel: 1

Voice List

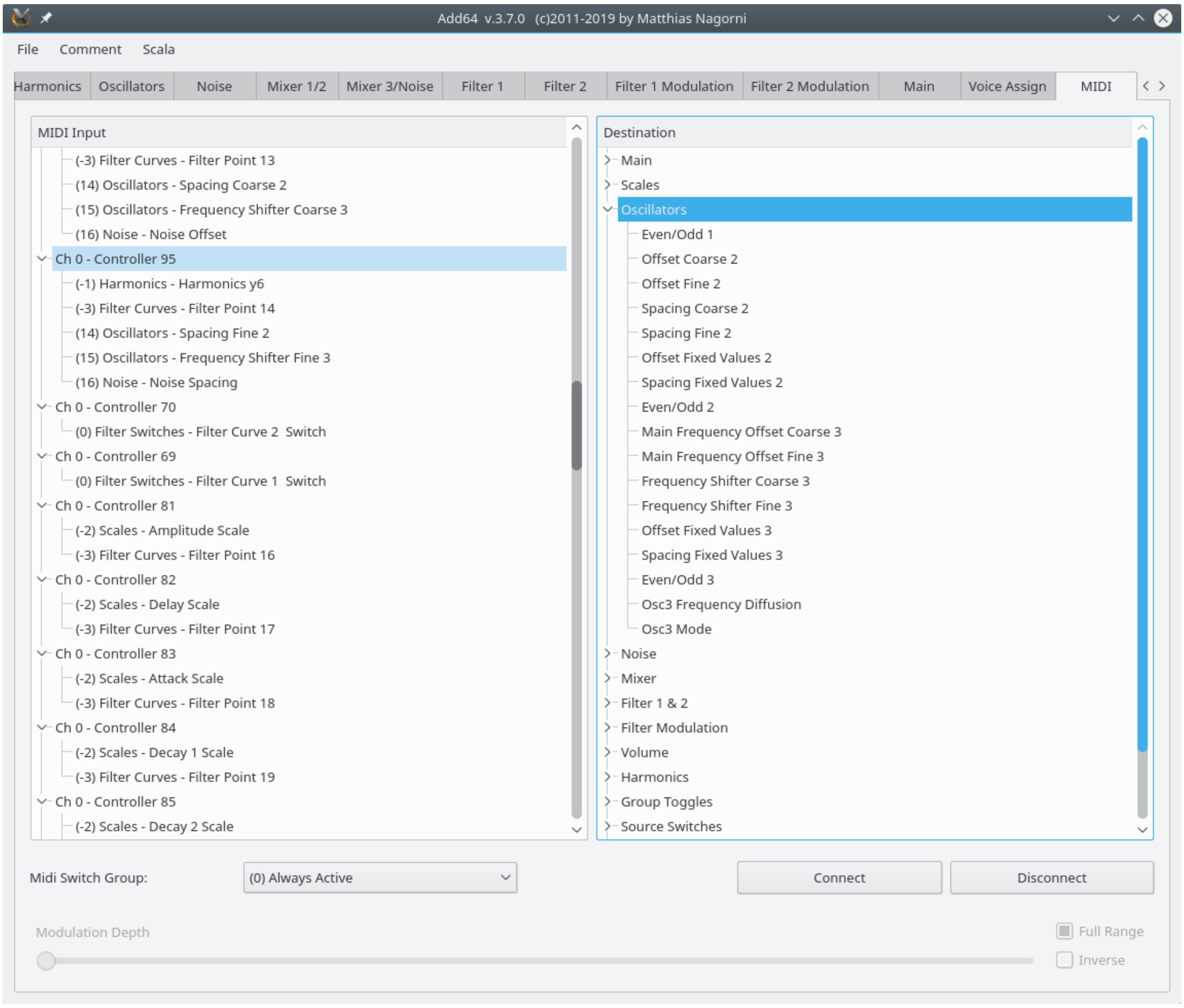
Voice	Channel	Sound File
1	1	Add64Presets/Add64Preset_0_0_0
2	1	Add64Presets/Add64Preset_0_0_0
3	1	Add64Presets/Add64Preset_0_0_0
4	1	Add64Presets/Add64Preset_0_0_0

Figure 10.1 The *Voice Assign* tab: Different sound files can be assigned to ranges and midi channels.

11. MIDI

Any parameter of Add64 can be asigned to a MIDI controller as well as note, velocity and pressure/aftertouch. By selecting a specific MIDI binding in the ***MIDI Input*** list, the ***Modulation Depth*** and ***Full Range*** parameters are enabled and let you define the MIDI range for the selected parameter.

Each MIDI controller can control several layers of parameters. Layer switching is based on toggles and switches. This way, ***Add64*** can be remote controlled by either a MIDI controller like BCR2000 or the virtual controller ***Add64-Ctrl***.



MIDI Input

- (-3) Filter Curves - Filter Point 13
- (14) Oscillators - Spacing Coarse 2
- (15) Oscillators - Frequency Shifter Coarse 3
- (16) Noise - Noise Offset

Ch 0 - Controller 95

- (-1) Harmonics - Harmonics y6
- (-3) Filter Curves - Filter Point 14
- (14) Oscillators - Spacing Fine 2
- (15) Oscillators - Frequency Shifter Fine 3
- (16) Noise - Noise Spacing

Ch 0 - Controller 70

- (0) Filter Switches - Filter Curve 2 Switch

Ch 0 - Controller 69

- (0) Filter Switches - Filter Curve 1 Switch

Ch 0 - Controller 81

- (-2) Scales - Amplitude Scale
- (-3) Filter Curves - Filter Point 16

Ch 0 - Controller 82

- (-2) Scales - Delay Scale
- (-3) Filter Curves - Filter Point 17

Ch 0 - Controller 83

- (-2) Scales - Attack Scale
- (-3) Filter Curves - Filter Point 18

Ch 0 - Controller 84

- (-2) Scales - Decay 1 Scale
- (-3) Filter Curves - Filter Point 19

Ch 0 - Controller 85

- (-2) Scales - Decay 2 Scale

Destination

- Main
- Scales
- Oscillators
  - Even/Odd 1
  - Offset Coarse 2
  - Offset Fine 2
  - Spacing Coarse 2
  - Spacing Fine 2
  - Offset Fixed Values 2
  - Spacing Fixed Values 2
  - Even/Odd 2
  - Main Frequency Offset Coarse 3
  - Main Frequency Offset Fine 3
  - Frequency Shifter Coarse 3
  - Frequency Shifter Fine 3
  - Offset Fixed Values 3
  - Spacing Fixed Values 3
  - Even/Odd 3
  - Osc3 Frequency Diffusion
  - Osc3 Mode
- Noise
- Mixer
- Filter 1 & 2
- Filter Modulation
- Volume
- Harmonics
- Group Toggles
- Source Switches

Midi Switch Group: (0) Always Active

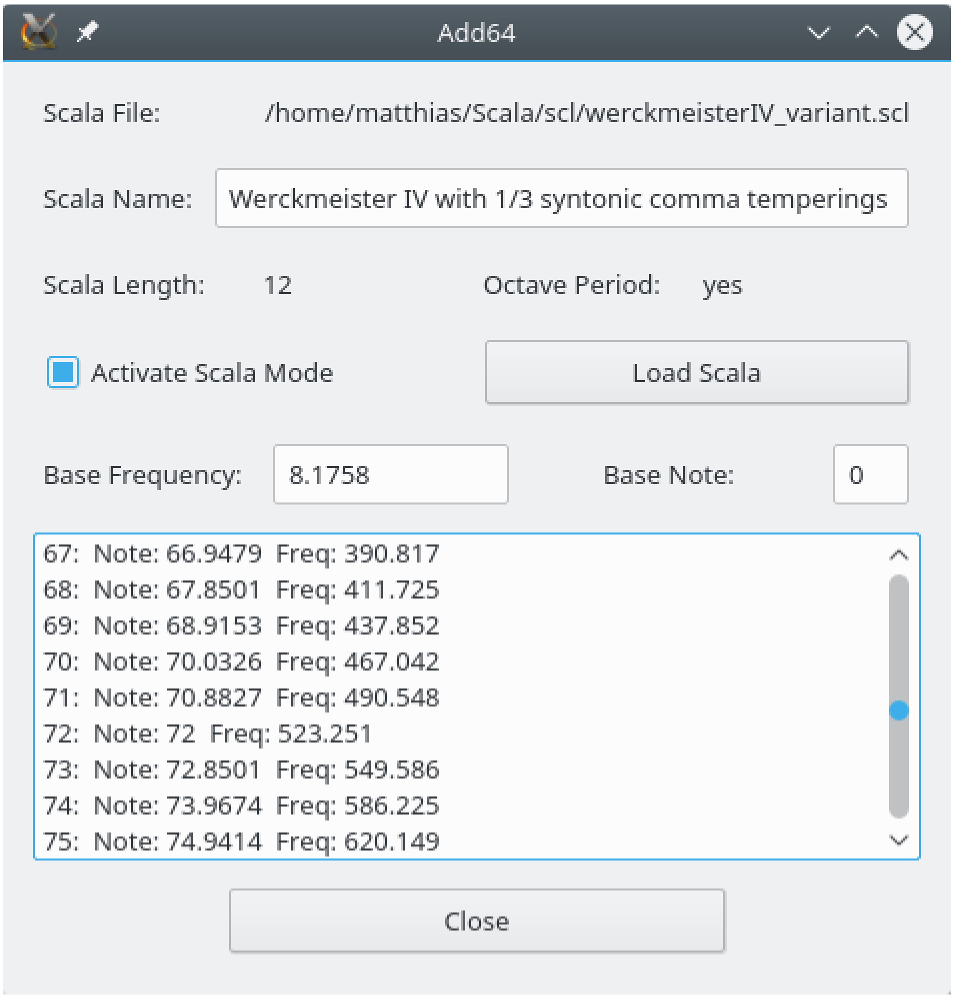
☑ Full Range

☐ Inverse

Figure 11.1 In the *MIDI* tab, every parameter can be bound to MIDI controller, note and pitchbend events.

11. Microtuning with Scala

Add64 supports microtuning with Scala .scl files. These files can be loaded using the ***Scala Dialog***. This dialog displays the length of the scala and the frequencies for all MIDI notes. You can change the ***Base Note*** if you would like to transpose the scala. The ***Base Frequency*** can be modified for tuning.



Scala File: /home/matthias/Scala/scl/werckmeisterIV_variant.scl

Scala Name: Werckmeister IV with 1/3 syntonic comma temperings

Scala Length: 12

Octave Period: yes

☑ Activate Scala Mode

Base Frequency: 8.1758

Base Note: 0

```
67:  Note: 66.9479  Freq: 390.817
68:  Note: 67.8501  Freq: 411.725
69:  Note: 68.9153  Freq: 437.852
70:  Note: 70.0326  Freq: 467.042
71:  Note: 70.8827  Freq: 490.548
72:  Note: 72  Freq: 523.251
73:  Note: 72.8501  Freq: 549.586
74:  Note: 73.9674  Freq: 586.225
75:  Note: 74.9414  Freq: 620.149
```

Figure 12.1 The Scala dialog for microtuning. In this example, Note 69, which corresponds to a^1, is tuned to 437.85 Hz.