

ZVORK

OBERON



User Manual

version 2.0.0

Table of Contents

First Contact.....	4	Curve display.....	15
Multi-point loop-able envelopes.....	4	Keyboard shortcuts.....	15
Editable waves.....	5	Edit Panel.....	16
Editable Filter response curves.....	5	Envelope controls.....	16
Sample loading.....	5	Attack.....	16
X-Morph between oscillators.....	5	Sustain.....	16
Unison and Octave copies.....	5	Release.....	16
Timbre bend.....	5	Time scaling knobs.....	17
Separate voice audio and gate outputs.....	5	Waveform controls.....	17
What's new with version 2.0?.....	6	Interpolation Mode.....	17
Sample loading.....	6	Modifiers.....	17
Envelope enhancements.....	6	Final Wave View.....	18
Wave form modifiers.....	6	Sample controls.....	18
LFO Delay & Rise.....	6	Importing a sample.....	18
PolyCV audio output mode.....	6	Sample edition.....	18
GUI improvements.....	6	Root Note.....	19
The Oberon engine.....	7	Filter controls.....	19
Oscillators.....	7	Morph and bend controls.....	19
Waveforms.....	7	Performance.....	20
Motion.....	7	Pitch Bend and Modulation wheel.....	20
Bend.....	7	Keyboard Mode and Glide.....	20
Partial Clip.....	8	Oscillators.....	20
X-Morph.....	8	Tuning.....	20
Harmonize.....	8	Keyboard Tracking.....	20
Filters.....	8	Filters.....	20
Response Curve.....	8	Modulation.....	20
Frequency Parameter.....	8	Keyboard Tracking.....	21
X > Y Parameter.....	9	Back Panel.....	22
Unison.....	9	CV Inputs.....	22
Octave Copy.....	9	PolyCV support.....	22
Mixer.....	9	CV Outputs.....	23
Stereo control.....	9	Audio Outputs.....	23
Modulation.....	9	Mixed outputs.....	23
LFO.....	10	Separate outputs.....	23
Envelopes.....	10	Gate and Pitch out.....	24
Tempo synchronization.....	10	A brief introduction to Additive Synthesis...25	
Loop.....	10	Harmonic sounds.....	25
Mono trigger.....	11	Inharmonic sounds.....	25
Release Modes.....	11	Tips for sample loading.....26	
Sources.....	11	Playing the sample with loop.....	26
Destinations.....	11	Stereo or multi-channel samples.....	26
Front Panel.....	14	Very short samples.....	26
Display.....	14	Very long samples.....	26
Common controls.....	15	Influence of audio sample sample rate.....	27
		Influence of sample pitch.....	27
		MIDI Implementation Chart.....28	

Remote Map Template.....31



First Contact

The **Oberon** synthesizer has two oscillator sections and two filters. Each oscillator can be fed to one of the two filters or to both, in serial or parallel mode. These filters enable you to alter the timbre of your sound but are not the only mean to do so.

Each oscillator can generate a number of different waves, from simple sawtooth and square to more complex metallic or robotic sounds. Each of these waves is in reality a wave sequence that can be played at different speeds. All this could seem very classic if it wasn't for these following features.

Multi-point loop-able envelopes

Each of Oberon's four envelopes are defined by a user editable curve, with up to 16 points. These envelopes are furthermore loop-able, between user definable start and end points, with a tempo synchronization option.

Editable waves

There are two editable waves, or wave sequences, that the user can customize. For each of these wave sequences, two editable curves let you draw the shape of the start and end waveforms of the sequence.

Editable Filter response curves

In Oberon, the filters don't have predefined modes such as low pass, high pass or comb. Instead you define the filter response yourself, again with an editable curve. Each filter response can morph between two curves for even more dramatic and changing effects.

Sample loading

Since version 2.0, Oberon is capable of loading samples from any format supported by Reason. The sample is analyzed and transformed into a wave sequence.

X-Morph between oscillators

This special feature lets you smoothly morph between two sound sources, potentially of different pitches and timbre. Morph between a pitch downed version of the same oscillator or between oscillator 1 and oscillator 2 for original transitions. The morphing effect can also be masked for certain portions of the spectrum if, for instance, you want to apply the character of one sound source solely to the high frequencies.

Unison and Octave copies

To fatten your sounds, each oscillator can be virtually duplicated up to 8 times with a pitch detune. After that, if this isn't enough, another virtual copy of the resulting sound can be added to your final sound mix at an octave higher. As the term "virtual copy" coins, all this is extremely efficient on your CPU.

Timbre bend

If you are in need of metallic inharmonic sounds, this is the functionality you need. With it you can quickly bend your sound to mimic highly rigid strings, bells or quirky buzzes, in a predictable way.

Separate voice audio and gate outputs

On the back panel you will find 8 separate outputs for both stereo audio and CV gate. Individual voices can be distributed among these outputs.

What's new with version 2.0?

Version 2.0 of Oberon brings some great new updates that either enriches your sound creation or facilitates your patch editing.

Sample loading

Wave 1 and Wave 2 can be fed with audio samples, provided they are of any format supported by Reason. This enables you to manipulate recorded audio inside Oberon or use samples as wave sequences.

Envelope enhancements

Now each envelope that is made loopable can have its loop start point and end point defined on any envelope points. With this, looped envelopes now have a distinct attack, loop and release phase. Combined with sample loading, this lets you control your sample loop sections.

The release sections of all three modulation envelopes now have two new behaviors which let you loop the release section for great echo style effects or completely disable the release section and continue looping through the sustain part when a note is released.

Wave form modifiers

This new feature will completely change the way you edit your user wave forms. Start by drawing your shapes like usual and apply FM or Sync to these shapes. Two modifiers applied in sequence are available for each user waves.

LFO Delay & Rise

Each LFO now has two parameters to control the evolution of its modulation intensity over time. The delay parameter delays the time of activation of the LFO. The rise parameter controls the duration of the attack of the modulation intensity.

PolyCV audio output mode

If you wish to use Oberon inside Blamsoft's polymodular system, this will make it even more transparent. Switch the audio output mode to "PolyCV" on the back panel and Oberon will dispatch the note audio outputs in a polymodular compatible manner.

GUI improvements

The most obvious visual change with version 2.0 is the disappearance of the modulation matrix on the front panel. We have migrated it inside the central top dynamic display with improved usability like grouped modulation destinations inside the popup menus. On all edit sections they are a number of popup menus now with grouped functionalities too.

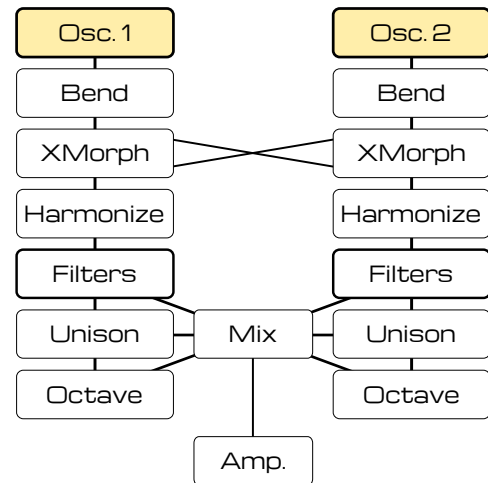
For even faster curve editing, Ctrl / Cmd, Alt or Shift keys can now be used in conjunction to instantaneously add a point, remove a point or move a point with X & Y snap enabled.

The Oberon engine

Under the hood of Oberon lies an efficient additive synthesis engine. If you don't know what additive synthesis is, don't worry as Oberon masks all the complexity of it in a simple paradigm. All you need to know is that with it, Oberon lets you sculpt your sound in a very precise way. If you are of the more curious type, you can take a look at the "A brief introduction to Additive Synthesis" chapter for some general presentation and terminology.

Oscillators

These are the sources of your sounds and Oberon offers two of them. By default, only the first oscillator is enabled. Both have the same functionalities, starting with a choice of basic wave sequences.



Waveforms

These wave sequences are made up of waveforms that are not necessarily of a determined pitch, which is what you find in most synthesizers. Oberon can produce inharmonic wave forms as well:

- **Wave 1 and 2** : these are two editable wave sequences that produce user defined periodic waveforms. With version 2, these can also be replaced by audio samples.
- **Oberon** : as the name suggests, this is Oberon's presentation phrase in a heavily robotic fashion.
- **Metal** : look no further for metallic sounds. When in motion, this wave sequence is finger nails on board metal heaven.
- **FM 1 and 2** : two wave sequences that span two or three operator FM waveforms.
- **Water** : when in a motion, a gurgling noise sound. If not, this is a source of low frequency inharmonic waves.
- **Noise** : a classic white noise, when in motion.

Each oscillator has a wave position and wave motion parameter to control the playback of these wave sequences.

Motion

Wave motion can be synced to the songs tempo. At speed 1/1 the wave sequence duration is one bar. Furthermore, the wave motion can be reset at each note on.

Samples

The editable Wave 1 and Wave 2 waveforms can also be replaced by audio samples. The wave sequences are constructed after analyzing the sample spectrum over time. A complete discussion on samples in Oberon can be found at "Tips for sample loading".

Bend

An oscillator produces a particular sound spectrum that can be "bent" with this specific tool. By default, the **bend** parameter stretches the sound spectrum towards the high frequencies. This is the same effect as a guitar string being tightened more and more, while keeping the same note. As the string tightness increases, it becomes more rigid and gradually becomes inharmonic.

But the bend operator is much more powerful than that. Each oscillator has a user editable bend curve associated with its bend control. From low frequencies at the left, to the high frequencies at the right, the curve determines if the spectrum around the curve point should be pitched up or down depending on the vertical position of the point.

Partial Clip

Each waveform produces a complex spectrum with highly varying levels for each frequency. By applying partial clipping, you will be able to flatten out these levels. Partial clipping has the effect of applying a gain value to each spectrum levels and clipping them to a maximum value. This is a very rapid mean of enhancing high frequencies as this is frequently the part of the spectrum of low level.

X-Morph

Cross morphing is the tool to continuously morph between two spectrums. An oscillator is the morph source. The destination, or target, can either be the other oscillator or itself. Of course, morphing an oscillator to itself doesn't do much, unless you decide to morph an oscillator to a detuned version of itself. The morph target has a detune control, with semitone steps. Morphing with a detuned target yields an interesting "growling" effect completely dissimilar to plain pitch bending.

Furthermore, the cross morphing can be limited to certain parts of the spectrum. Once again, this is achieved through an editable morph curve.

Harmonize

The **harmonize** control lets you bend an inharmonic spectrum towards a harmonic one. On harmonic sounds such as a sawtooth wave, this has no effect. This is mostly useful when you have bent heavily or morphed your sound and still wish to keep it musical.

The harmonize parameter can be limited to only the lower part of the spectrum through the limit parameter. At high values, the entire spectrum is harmonized while at low values only the low part is.

Furthermore, you can choose to which octave the harmonization should apply. The default value is 0, meaning the spectrum is harmonized to the current played note. At values -1 and -2, the spectrum is harmonized one octave or two octaves under the played note, respectively. This is useful if you wish to keep some low frequencies after harmonization.

Filters

Oberon possesses two filters which can be configured in series or parallel. Each oscillator can bypass the filters or be routed to one or both of them.

Response Curve

The filters in Oberon are in fact akin to graphical equalizers. You draw the frequency response with a multi-point curve. By default, this curve is defined around a central frequency and ranges eight octaves below to eight octaves above this central frequency.

The filters can be set in "repeated" mode, in which case the curve defines the response from 0 to a maximum in linear frequency. Above the maximum frequency, the curve is repeated, as a comb filter.

Frequency Parameter

The **Frequency** parameter of a filter controls the central frequency, or the maximum frequency in repeat mode. It is similar to the traditional "frequency" parameter in classic filters of subtractive synthesizers. You can also see this parameter as a frequency scale value on the filter curves.

X > Y Parameter

Each filter has two response curves, X and Y. When the X/Y parameter is 0, the X response curve is used. At its highest value, 100, the Y response curve is used. In between, the response curve is morphed between the two.

By default, the filters are configured with a low pass filter response and the high Y curve has a resonance at the central frequency. Changing the X/Y thus adds some resonance to your low pass filters.

Unison

Unison can be disabled (the default) or set to 2, 4 or 8 voices. This has the effect of adding slightly detuned copies of the oscillator sound after the filtering stage. The start phase of these copies is controlled with the **Phase Lock** and **Phase Random** parameters. Phase Lock resets the phase to the same value at each note trigger to have a consistent sound.

Phase random randomizes the phase of the detuned copies for each partial of the spectrum. This, on periodic sounds, has the effect of giving a harsher metallic character to your sound.

The final mix of these unison copies as well as the detune spread are controlled in the mixer stage.

Octave Copy

Like the Unison effect, the Octave Copy adds a copy of the oscillator sound after the unison stage, but this time detuned a whole octave above the original sound. This is great to fatten and double bass sounds for instance or enrich your pads.

The final mix of the octave copy for each oscillator is controlled in the mixer stage.

Mixer

At this final stage, you can control the overall output gain of each oscillator as well as the gain of its virtual octave higher copy. In all, Oberon can output four sound sources.

The mix between the dry oscillators and the unison signal is controlled with the **Unison Mix** parameters, one for each oscillator. The amount of unison detuning is controlled, again individually for each oscillator, with the **Unison Pitch**.

Finally, the overall volume, modified by the amplitude envelope is applied, along with velocity modulation.

Stereo control

By default, the oscillators are mono but a **Width** parameter enables you to spread their output in stereo, for a wider field. This result can then be panned from left to right, per oscillator.

Modulation

A modulation matrix lets you route modulation sources to modulation destinations. Each line of the matrix offers one source, two destinations and a scaling source. Each destination as well as the scaling source have a strength value to control the amount of modulation.

LFO

There are two LFOs in Oberon. Both of them can be synced to the song tempo and reset at each note on. With version 2, the LFOs now have a delay and rise parameter which controls a simple "Delay / Attack" envelope of the LFO intensity. Each LFO can be configured to produce a certain wave form among the following list:

- **Triangle** : a basic triangle wave with initial rise,
- **Triangle 4, 8 and 16** : same triangle as above but sampled and held at $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$ th of the wave period,
- **Sine** : a basic sine wave with initial rise,
- **Saw** : a classic saw wave with initial high value then fall,
- **Saw 4, 8, and 16** : same saw wave as above but sampled and held at $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$ th of the wave period,
- **Parabolic Saw** : a saw wave but instead of a constant linear fall features a steep starting fall gradually decelerating.
- **Soft Saw** : a saw with a soft filtered attack,
- **Square** : classic square wave starting with a high value,
- **Soft Square** : a square with softened rises and falls,
- **Pulse** : same as a square wave but with a reduced high value duration,
- **Soft Pulse** : same as above but with softened rises and falls,
- **Random** : a smoothly varying random wave,
- **Sample and hold** : a sample and held version of the above, sampled at the LFO's rate,
- **Random Pulses** : randomly occurring high value pulses,
- **Random Soft Pulses** : same as above but with softened rises and falls,
- **Random Signed Pulses** : randomly occurring pulses of random high or low value,
- **Random Signed Soft Pulses** : same as above but with softened rises and falls.

Envelopes

The four envelopes (Amplitude and Modulation 1 to 3) all share the same features. They are each defined by a curve, with up to 16 points. One of these points can be set to the sustain value. Before this sustain point is the attack section. After this sustain point is the release section.

When a note is played, the envelope follows the curve up to the sustain point. When the note is kept pressed, the sustain point value is held if the envelope has reached the sustain point. When the note is released, the envelope proceeds to follow the release section using the current envelope value as the start value of the release. Each segment of the envelope curve has a duration of up to 10 seconds.

Tempo synchronization

Envelopes can be synced to the tempo. In this case, the segment durations are not defined in seconds but in note durations. In this case, the maximum duration of a segment is 5 bars.

Loop

An envelope can be set to "loop" mode in which case, when a note is held and the envelope reaches the loop end point, it will jump back to the loop start point of the curve. Both loop start and end points can be modified.

Mono trigger

When this option is activated, the envelope will trigger on the “note on” event of the first note played. If this note is held and another note is pressed, this second note will use the same envelope value as the first one. The envelope is reset once all the notes using this envelope are released.

Release Modes

Modulation envelopes have an extra “release mode” parameter. By default, the release section of the envelopes is played when a note is released. When set to “Loop”, the release section is played in a loop, indefinitely. Finally, when set to “None”, the release section is ignored and the envelope is either held on to its sustain value, or looped indefinitely.

Sources

Name	Description	Polarity
<i>Velocity</i>	Note on velocity value	Unipolar
<i>Amplitude Envelope</i>	-	Unipolar
<i>Modulation Envelope 1 to 3</i>	-	Unipolar
<i>LFO 1 and 2</i>	-	Bipolar
<i>LFO 1 and 2 note on value</i>	The initial value of the LFO is sampled at note on and held throughout the key press. This is mostly useful when the LFO is not key synced.	Bipolar
<i>Aftertouch</i>	Also called pressure	Unipolar
<i>Modulation Wheel</i>	-	Unipolar
<i>Keyboard Tracking</i>	-	Bipolar
<i>Sustain Pedal</i>	-	Unipolar
<i>Expression</i>	-	Unipolar
<i>Breath Control</i>	-	Unipolar
<i>CV input 1 to 5</i>	Any signal from other devices can be plugged in Oberon and used as modulation sources with these inputs.	Bipolar
<i>Note On random value 1 and 2</i>	At each “note on” event, a random value is generated and held throughout the key press.	Bipolar
<i>Pitch Bend Wheel</i>	-	Bipolar

Destinations

Name	Description
<i>Global Level</i>	Final volume
<i>Oscillator 1 or 2 mixer levels</i>	-

<i>Filter 1 or 2 Frequency value</i>	-
<i>Filter 1 or 2 X > Y value</i>	-
<i>Oscillator 1 or 2 pitch</i>	-
<i>Oscillator 1 and 2 pitch</i>	Both oscillators receive the same amount of modulation.
<i>Oscillator 1 or 2 semitones</i>	This is similar to the oscillator pitch destination but only the semitone parameter is modulated, in discrete values.
<i>Oscillator 1 or 2 pan</i>	-
<i>Oscillator 1 and 2 pan</i>	Both oscillators receive the same amount of modulation
<i>Oscillator 1 or 2 width</i>	Stereo width
<i>Oscillator 1 and 2 width</i>	Both oscillators receive the same amount of modulation
<i>Oscillator 1 or 2 morph position</i>	Controls the amount of morphing
<i>Oscillator 1 or 2 morph target transpose</i>	The pitch transpose value of the morph target.
<i>Oscillator 1 or 2 wave position</i>	-
<i>Oscillator 1 or 2 wave motion</i>	-
<i>LFO 1 or 2 rate</i>	-
<i>Amplitude Envelope attack scale</i>	This will shorten the duration of the amplitude envelope's attack section (see the Envelopes paragraph) if the modulation is negative or lengthen it if it is positive.
<i>Modulation Envelope 1 to 3 attack scale</i>	Same as above but for the modulation envelopes.
<i>Amplitude Envelope release scale</i>	This will shorten the duration of the amplitude envelope's release section (see the Envelopes paragraph) if the modulation is negative or lengthen it if it is positive.
<i>Modulation Envelope 1 to 3 release scale</i>	Same as above but for the modulation envelopes.
<i>Oscillator 1 or 2 harmonicity</i>	-
<i>Oscillator 1 or 2 harmonicity limit</i>	-
<i>Oscillator 1 or 2 bend</i>	-
<i>Oscillator 1 or 2 unison mix</i>	-
<i>Oscillator 1 and 2 unison mix</i>	Both oscillators receive the same amount of modulation
<i>Oscillator 1 or 2 unison pitch</i>	The amount of unison detune
<i>Oscillator 1 and 2 unison pitch</i>	Both oscillators receive the same amount of modulation

Oscillator 1 or 2 octave level

-

Oscillator 1 or 2 partial clip

-

Front Panel



Display

At the top of Oberon's front panel lies a huge display. This is where all the curves, for envelopes, waves, bend, morph and filter responses, are edited. Switching from one curve to another is done by activating the corresponding button underneath the display. From the left to the right lie the buttons for editing:

- Oscillator 1 bend curve,
- Oscillator 1 morph curve,
- Modulation matrix,
- Wave 1 start and end waveform curve or audio sample 1 if a sample was imported,
- Wave 2 start and end waveform curve or audio sample 2 if a sample was imported,
- Amplitude envelope,
- Modulation envelope 1,
- Modulation envelope 2,

- Modulation envelope 3,
- Filter 1 “X” curve,
- Filter 1 “Y” curve,
- Filter 2 “X” curve,
- Filter 2 “Y” curve,
- Oscillator 2 bend curve,
- Oscillator 2 morph curve.

Common controls

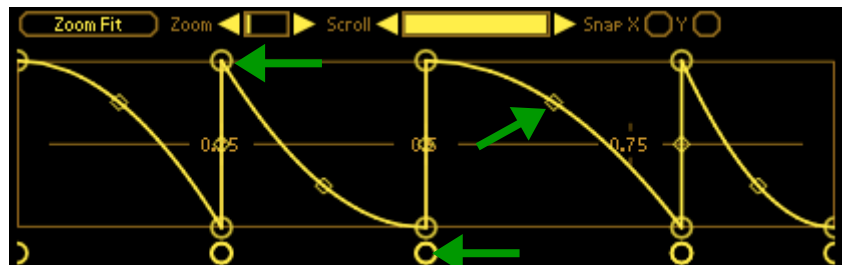
All curves share a common set of controls for edition. Specific controls are listed in curve specific paragraphs.

Curve display

At the center is displayed the currently edited curve.

Circular handles on the curve correspond to the curve points and can be clicked and dragged. Between each point lies a diamond shaped handle that lets you, by clicking and dragging vertically, adjust the shape of the segment from linear to accelerated or decelerated.

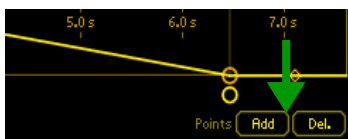
Underneath the curve lie other circular handles which, again, correspond to the curve points but let you move the point only in the horizontal direction.



Above the curve representation are a number of controls to adjust the view of the curve. The first button at the left, labeled “Zoom fit”, when clicked, will adjust the zoom and scroll position of the view so as to have all the curve points visible in the view.

Following to the right is a scrollbar linked to the zoom of the view. You can either click and drag the zoom handle or click the left and right arrows located around the scrollbar. Controlling the scroll position, or center of the view, follows the same method as the zoom, with a slightly larger scrollbar.

Finally, two checkboxes let you, when checked, limit the movement of the curve points to an invisible coarse grid, independently in the horizontal (X) or vertical (Y) direction. The resolution of the grid depends on the edited curve. This is referred to as “Snapping”.



Underneath this panel, at the far right, you will find two buttons to add or delete a curve point. Pressing one of these buttons will switch point edition from move to the respective function. For instance, if “Add” is activated, clicking on the curve display will add a point at the clicked location. If you have exhausted the maximum number of points allowed for that curve, the

“Add Point” button will be disabled. Likewise, if “Del.” is activated, clicking on a point handle will remove that point. If the minimum number of curve points has been reached, this button will be disabled.

Keyboard shortcuts

For quicker edition of your curves, with version 2, you can combine various mouse movements with modifier keys to gain immediate access to certain editing features:

- Clicking on a curve while pressing the Ctrl (PC) or Cmd (Mac) key adds a point at the click position,
- Clicking on an existing point while pressing the Alt key removes that point,
- Dragging a point while pressing the Shift key enables snapping in the X and Y directions.

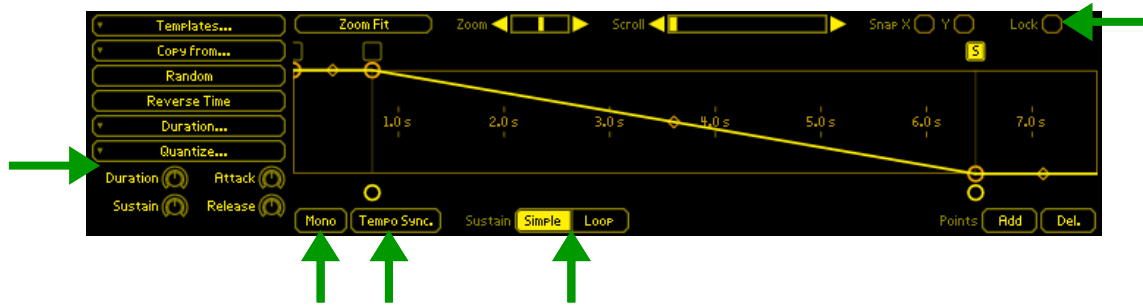
Edit Panel

To the right of the curve display lies a panel of buttons dedicated to the global edition of the curve. The exact buttons are context sensitive and will depend on the edited curve. They are mainly copy buttons to copy one curve to another, template selection or a random creation function.



Envelope controls

Envelope curves have four specific controls. The first one, “Lock” at the right of the snap controls changes the way the curve is adjusted when a point's horizontal position is changed.



Except for envelopes, moving a curve point doesn't have any effect on all the other points. For envelopes, the default behavior when moving a point is to translate horizontally the points located to the right of the moved so as to keep the segment durations constant. In this way, the envelope curve edition is similar to the traditional Attack, Decay and Release controls. When “Lock” is checked, the behavior will revert to the standard method.

Finally, underneath the curve display you will find the buttons to active **Mono Trigger**, sustain **Loop** and **Tempo Synchronization** for this envelope. When tempo synchronization is activated you will notice the horizontal scale labels of the curve change to three numbers separated by points. The first number is the bar number, the second quarter note number and the third sixteenth notes.

Attack

On all envelopes, the attack section is defined as the curve section between the first point and sustain point, or start of loop point in looped sustain (see below).

Sustain

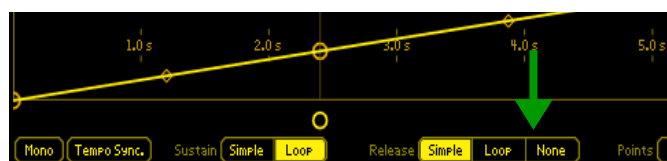
When the sustain loop mode is set to **Simple**, the default, the envelope value will be maintained at the value of the defined sustain point. This point is set by clicking on the square handles at the top of the envelope curve. The current sustain point is identified by a “S” mark.

When the sustain loop mode is set to **Loop**, the envelope is looped between a start and end point, which can also be set at the top of the envelope curve. These two points are symbolized by two arrows pointing left or right. The arrow pointing to the left corresponds to the start point and the arrow the right, to the end point.

Release

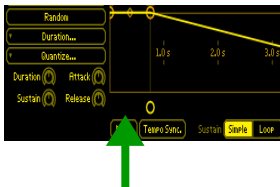
On all envelopes, the release section is defined as the curve section between the sustain point, or end of loop point and the last point of the envelope.

Modulation envelopes have an extra parameter for the release section which lets you configure the envelope playback behavior once a note is released.



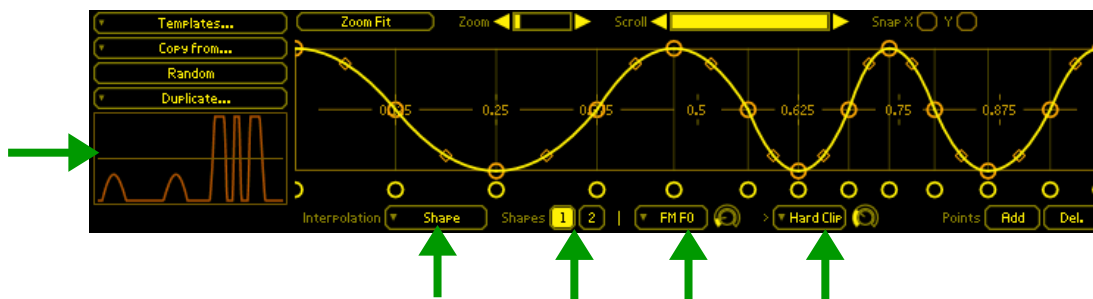
- **Simple:** this is the default mode. The envelope values on a note release correspond to the curve value, played in increasing time until the last point.
- **Loop:** On a note release, the envelope is read as above but once the last point is reached, the envelope “play position” is set back to the start point of the release section.
- **None:** this completely disables playback of the release section. On a note release, the envelope stays in sustain mode, single point or loop mode, whatever the sustain mode.

Time scaling knobs



On the left panel, envelope edit pages display four knobs that enable you quick time scaling of the envelopes. The first “Duration” knob scales the entire duration of the envelope while each following knob controls only a sub-section of it. Please note that when the sustain is in “simple” mode or if the loop start and end points are identical, scaling duration of the sustain has no effect.

Waveform controls



When either Wave 1 or Wave 2 is selected for editing, you will notice two new buttons underneath the curve display. These “Shapes” button lets you pick which waveform of the wave sequence you wish to edit. Shape 1 is the waveform output when an oscillator’s wave position control is 0% and 100% while Shape 2 is the waveform output when this same wave position control is at 50%.

In the edit section you will also see some new actions to create random shapes or duplicate points.

Interpolation Mode

As of version 1.1 of Oberon, interpolation between Shape 1 and Shape 2, can be chosen between three different modes:

- **Shape:** this is the legacy mode from version 1.0 and consists of morphing the wave curve shapes. The sound is frequently very similar to classic Pulse Width Modulation or comb filtering.
- **Spectrum:** This new mode works by morphing the frequency spectrum between the spectrum of shape 1 and the spectrum of shape 2. Spectrum mode is interesting when you wish to keep the formant characteristics of the shapes as the formant peaks will move during interpolation.
- **Crossfade:** This last mode is the most neutral in character as it is just a classic level crossfade between the two shapes.

Modifiers

With version 2 is introduced two modifiers per shape. These modifiers are used to modulate the shape curve and produce the final wave shape. The modifiers are applied in sequence and have an intensity parameter. Here the list of modifiers:

- **Sync:** This reproduces the hard sync effect of analog synthesizers where a master oscillator resets the phase at a certain frequency of another oscillator, which can be tuned to another frequency. The intensity of the modifier controls the frequency of the servant oscillator playing the shape.
- **FM F1-11:** This applies frequency modulation to the shape. The modulator is a sine wave of frequency multiple corresponding to the F index. FM F2, for instance modulates the shape with a sine wave at twice the frequency of the shape. The intensity of the modifier controls the modulation depth.
- **FM Auto:** This also applies frequency modulation but this time the modulator is the shape itself.
- **Soft Clip:** This applies a soft clipping to the shape. The intensity of the modifier controls the drive value.
- **Hard Clip:** This applies a hard clipping to the shape. The intensity of the modifier controls the drive value.
- **Low Pass, LP Res 1 & 2:** This applies a low pass filter to the wave shape, the last two being with increasing level of resonance. The intensity controls the cutoff of the filter, high values giving low cutoff frequencies.
- **Bit Depth:** This applies a bit reduction distortion to the wave shape. High intensity values gives lower bit depth and increased distortion.
- **Phase Distortion 1 & 2 :** This applies two different phase distortion modulation on the wave shape. Logically, high intensity values gives more modulation.

Final Wave View

To have a clearer view of what the modifiers are doing, a wave view on the lower left of the wave display, displays the final wave as it will be synthesized.

Sample controls

Importing a sample

With version 2, you can import your own samples into Oberon and use them like any other built-in wave sequence. To do so, click on the Reason standard sample browse buttons underneath the “Waves” label and drag and drop a sample onto Reason. This will apply the sample to the Wave 1 wave form. To apply a sample to Wave 2, simply display the Wave 2 edit page and drag a sample.

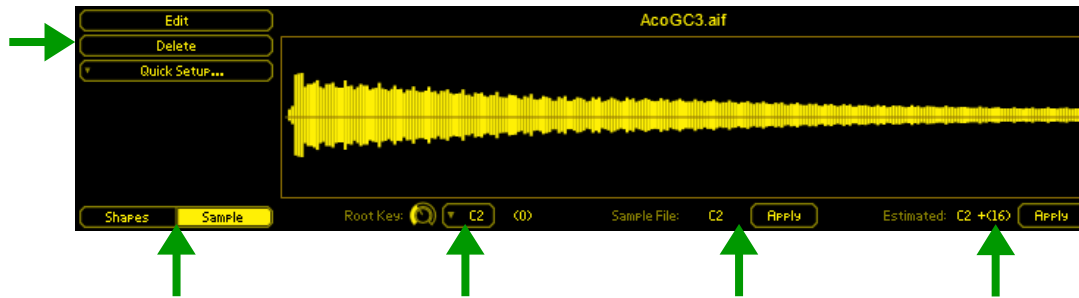


Once a sample is loaded in Wave 1 or Wave 2, the display changes to show the resulting sample's wave envelope. Whichever Wave 1 or Wave 2 has this sample will now play the sample's spectrum based on the oscillator's start position, just like any wave form.

The “Tips for sample loading” chapter at the end of this document is a recommended read to understand how Oberon manages samples and the limitations.

Sample edition

To switch back to the wave shape curves, just select the “Shapes” entry on the radio button at the bottom left of the display. Likewise, to switch back to the sample, select “Sample”.



If you wish to play the entire sample and its loop section, you must modulate the start position of the oscillator playing the wave. You could set the oscillator’s motion to 100% but this will play the sample in cycle, restarting it at each cycle, without using its loop section. To play the sample only once while also playing its loop section, we can use the modulation envelopes. A “Quick Envelope Setup” button on the left can be used to quickly setup a modulation envelope that correctly plays the sample. Click on this button and choose the envelope to use then go to the modulation matrix and link the envelope to the oscillator’s start position with a modulation strength of 100.

To change the loop section of the sample, or select a subset of the sample, you can also open Reason’s standard sample edition window by clicking on the “Edit” button at the top left.

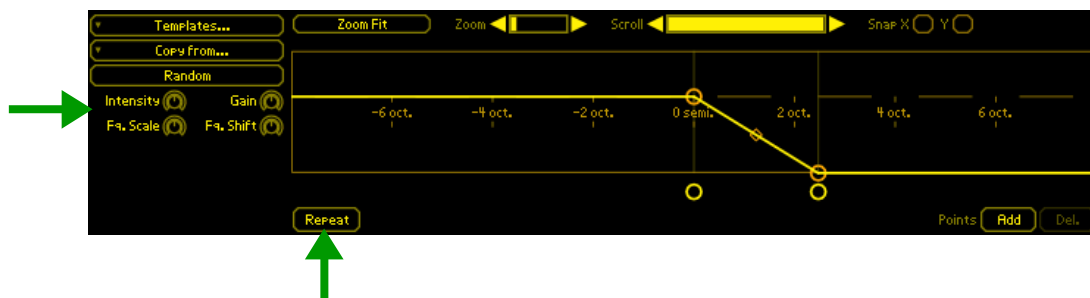
Deleting the sample will clear the sample for that Wave and go back to the normal wave curve mode.

Root Note

To set the sample’s root note, you can either manually set it with the root note knob or the direct node selection menu by clicking on the note button to the right of the knob. Fine tuning of that root note is done by dragging the semi tone display to the left, its value shown in parenthesis. But you can also quickly use the root note defined inside the sample’s audio file or through the Reason sample edition window by clicking on the “Apply” button in the “Sample File” section. Likewise, you can quickly use an estimated root note based on a pitch analysis performed by Oberon by clicking on the “Apply” button in the “Estimated” section.

Filter controls

The filter response curve display adds a number of controls. The first one is a “Repeat” button that lets you toggle your filter response to repeat (see Response Curve paragraph). The repeat control is common for both X and Y filter curves.



In the left panel, four knobs let you quickly scale either the gain values or frequency values of your filter curve. The “Intensity” knob controls the amplitude of the points in the vertical direction. The “Gain” knob will shift the points upwards or downwards, vertically. The “Fq. Scale” knob will squeeze or stretch the frequency of the curve points, around the “0 semi” value. Finally, the “Fq. Shift” knob will let you translate the points left or right along the frequency axis.

Morph and bend controls

Morph and bend curves do not have a shape control handle. All segments are linear.

Performance

Pitch Bend and Modulation wheel

The range of the pitch bend can be adjusted with a maximum range of 2 octaves (24 semitones).

Keyboard Mode and Glide

There are four keyboard modes:

- **Polyphonic – Key Pressed:** the default mode where each new pressed key creates a new voice.
- **Polyphonic – Note :** voices are allocated to notes. If you press a key twice on the same note and the previous voice for that note is still playing, this previous voice will be killed and restarted. This is the case for piano like sounds when repressing a note has the effect of killing the previous sound for that note.
- **Mono Retrigger :** when pressing a key, the synth retriggers the envelopes and changes the pitch if a previous key is pressed.
- **Mono Legato :** this is the same as *mono-retrigger* but the envelopes are not retriggered.

By changing the *Glide* duration, you can add portamento effect that will make the pitch glide from the previous note to the next. This also works in *polyphonic* mode but in this case the portamento will take place only if you press a new key while previous notes are in release mode.

You can increase or decrease the glide duration based on the velocity with the dedicated knob underneath the glide control.

Oscillators

Tuning

There are three knobs to define the basic pitch of the oscillator. The first one at the left is used to adjust the pitch in octaves. The second one in semi tones and the last one lets you finely tune the oscillator in cents or hundredth of semi tones.

Keyboard Tracking

When set to 100 (the default), the pitch of the oscillator follows the note pressed on the keyboards. When set to 0, the pitch is constant across all keys.

Filters

The central knob of each filter section controls the filter's frequency parameter. The smaller knob to the right Labeled "X > Y" is used to morph between the X filter response curve and Y filter response curve.



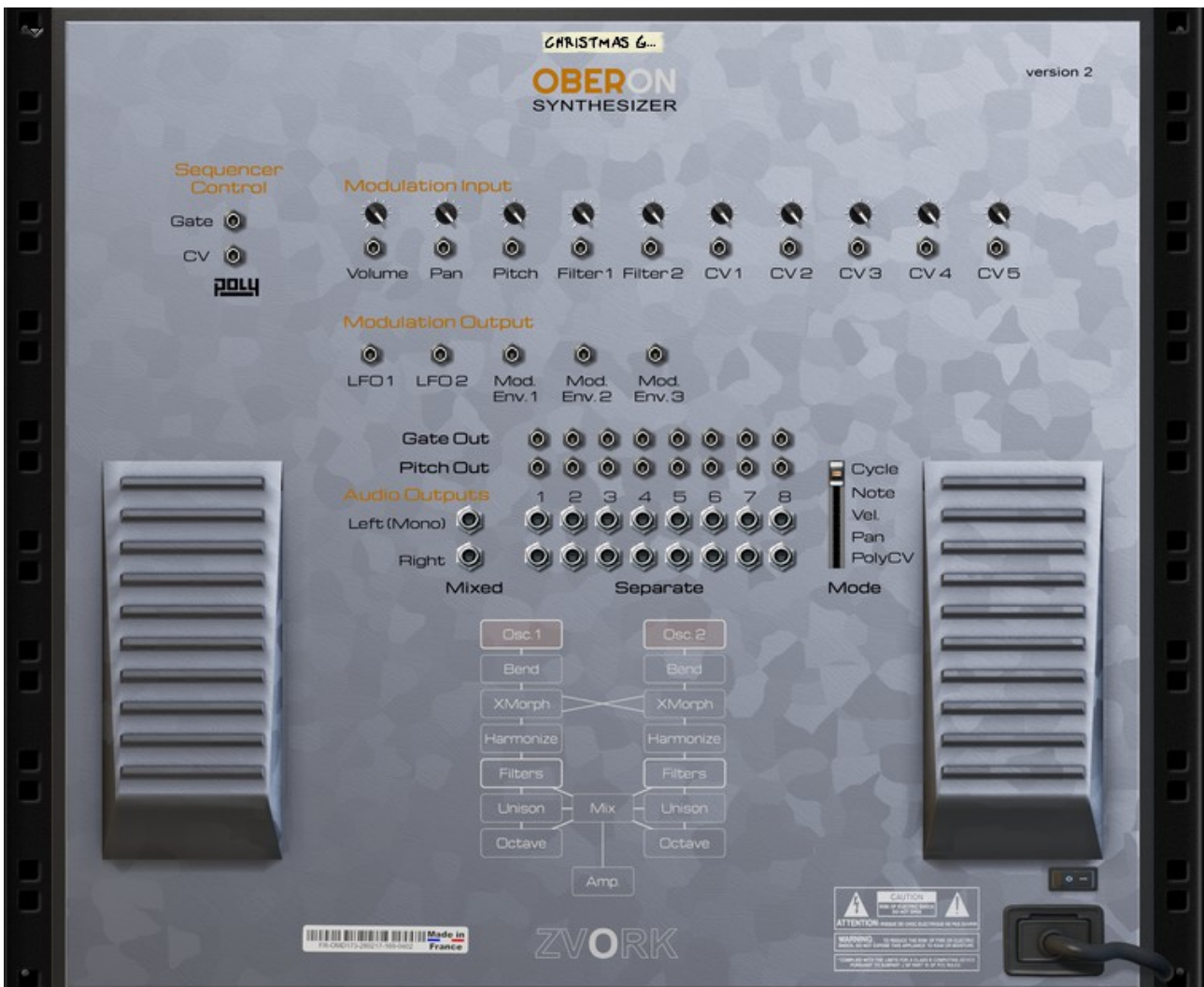
Modulation

Filter 1 and Filter 2 have predefined modulation paths between Modulation Envelope 1 and Modulation Envelope 2, respectively. The amount of this respective modulation can be controlled via a dedicated knob. In the same manner, velocity modulation intensity of the filters' frequency parameter can be controlled with a knob.

Keyboard Tracking

This controls how the filter frequency parameter is adjusted for each note. By default, the value is 0, in which case the filter frequency is identical along all the note range. At 100%, the frequency parameter is increased along with the note. Finally at 200%, the frequency parameter increases twice as fast as the note.

Back Panel



The back panel exposes the input / output CV and audio output sockets.

CV Inputs

Some of Oberon's parameters can have their values modulated by control voltage signals coming from other devices. Trim knobs let you control the amplitude of the modulation. Five special CV inputs (labelled CV 1, 2, 3, 4 and 5) are by default not routed to a specific parameter. It is up to you, through the modulation matrix, to route them to a particular modulation destination.

PolyCV support

As of version 1.1, Oberon fully supports Blamsoft's PolyCV protocol which enables you to control Noxious notes on/off from an external Rack Extension that supports the output of such signals. Check out the PolyCV description for more info: <http://blamsoft.com/polycv/>

CV Outputs

Oberon exposes five internal parameter to CV outputs:

- LFO 1 & 2 : this the value of the LFO if it were not key synced.
- Modulation envelope 1, 2 & 3 : the value of the modulation envelope of the first pressed voice is sent to this output. The value of a modulation envelope falls to zero when no notes are played. This happens as soon as the amplitude envelope of the last note playing ends, even if the modulation envelope has a longer release duration. When multiple notes are played, the combined output sent to the CV socket is the maximum of all voice values for that particular envelope.

There are also 8 special “gate out” CV outputs and 8 “pitch out” that are explained in the Separate outputs paragraph.

Audio Outputs

Mixed outputs

The stereo output is sent to the left and right mixed audio socket. If only the left socket is connected to another device, Oberon sends the mixed mono signal to the left socket.

Separate outputs



Oberon can output individual voice stereo outputs separately from the mixed stereo output. There are 8 stereo pair outputs with corresponding CV note gate and pitch output.

How voices are distributed among these outputs is determined by the voice output Mode. Oberon will automatically detect outputs that are connected to cables and only connected outputs are taken into account in

the voice output Mode. The order of the outputs are from left to right.

Here are the four possible modes:

- **Cycle:** the default mode will send the first voice played to the first connected output (on the far left). Subsequent played voices will be sent chronologically to increasing connected outputs. When all outputs are used, the next voice is sent and mixed to the first connected output. On the image above, separate audio outputs number 2, 4 and 5 are connected. If five notes are activated, then the first voice will be sent to output 2, the second to output 4, the third to output 5, the fourth to output 2 (and mixed with the first voice) and finally, the last voice will be sent to output 4 (and mixed with the second voice).
- **Note:** the distribution of voices among the separate audio outputs is done according to the MIDI note number of each voice. There are 128 possible MIDI notes. If there are three connected outputs (as in the example above), the lower third of MIDI notes (here, notes 0-42) are associated with the first connected output (here output 2), the middle third (here, notes 43-85) to the second connected output (here output 4) and finally the upper third (here, notes 86-127) to the last connected output (here output 5).
- **Velocity:** the distribution of voices is done according to the voice velocities. Lower velocity voices are sent to lower numbered connected separate outputs and, logically, higher velocity voices are sent to higher numbered connected outputs.

- **Pan:** the distribution of voices is done according to the voice stereo pan at the moment the voice is activated. The voice pan is estimated as the mean of each oscillator pan. If the pan changes during the lifetime of a voice, through modulation, this will not change the audio output to which it is sent.
- **PolyCV:** this is the mode you should use if you wish to correctly integrate Oberon with Blamssoft's Polymodular system, in conjunction with a PolyCV gate signal.

Gate and Pitch out

When a voice sent to a separate output is activated (on a Note On) a gate signal will be sent to the associated gate output, above the audio output, as a CV signal with amplitude proportional to the velocity of the last activated voice of the output. Likewise a Note (Pitch) CV signal will be sent to the associated pitch output corresponding to the note number.

If no separated outputs are connected, no gate or pitch outs are sent.

A brief introduction to Additive Synthesis

A majority of synthesizers are based on a particular form of synthesis called “subtractive synthesis”. It consists in producing a spectrally rich wave form, such as sawtooth, square or noise and applying a filter on these wave forms to remove, or subtract, parts of their spectrum. By varying the types of filters and their parameters, they can produce a fairly wide variety of sounds.

Additive synthesis works the other way around. It consists in building this rich spectrum from scratch by adding up individual sinusoidal oscillators, called partials in Oberon. Sine waves have the particularity of producing sound in a single frequency band of the spectrum. By adding up multiple sine waves of different frequency, we can, in theory, reproduce any frequency spectrum. In fact, science demonstrates that any sound of any length can be created with an infinite number of sine waves, properly tuned and equalized. This of course isn't practical and in reality additive synthesis constrains you to a finite number of sinusoidal oscillators but with controllable pitch and level. Typical sounds consists of hundreds of sine waves so controlling each one of them is a bit of a challenge. We will see how Oberon handles the problem later on but in the meantime, let us see how typical sounds are built from these individual partials.

Harmonic sounds

Musical harmonic sounds, such as created by acoustic instruments or analog subtractive synthesizers, are produced by periodic wave forms. The frequency of these wave forms determines the pitch of the note. Typical periodic wave forms in subtractive synthesis include sawtooth, triangle and square waves. They are all made up of sine waves of frequencies multiple of the wave form's note frequency. This note frequency is called **fundamental frequency** in additive synthesis. For example, if you hit the A4 key on your keyboard, which corresponds to a frequency of 440 Hz, and produce a saw tooth wave form, this saw tooth is in fact made up of sine waves of frequency 440 Hz, 880 Hz, 1320 Hz or 1760 Hz (or 1 x 440Hz, 2 x 440Hz, 3 x 440Hz and 4 x 440Hz). The multiples of the fundamental frequency are called **harmonics**. The sine waves also have a particular amplitude but these amplitudes only change the timbre of the wave, not its note frequency or harmonicity. This difference in amplitude is what makes a sawtooth sound different from a square wave.

Waves 1 & 2, in Oberon are naturally harmonic as they are totally periodic.

Inharmonic sounds

The vast majority of sounds are not harmonic or tuned. Percussion or noise sounds for instance don't have a clear pitch and thus don't have partials tuned so simply as harmonic sounds. In fact most inharmonic sounds have partials tuned with no relation between one another. In reality, acoustic instruments are not totally harmonic in the pure sense of the term but are almost harmonic. Their partials are not strictly tuned to multiples of the fundamental frequency and this is what gives their richness as this introduces some slight chorus / beating effects. Typical inharmonic sounds are metallic sounds, such as mistuned bells or rigid strings, and these can be easily built from harmonic sounds by detuning some of their partials.

In Oberon, inharmonic sounds can easily be created with the Bend parameter or by using naturally inharmonic waves like “Metal”, “Noise”, “Water” or “Crowd”.

Tips for sample loading

Oberon comes with a number of built-in complex wave sequences, “Metal”, “FM 1”, “FM 2”, “Noise”, “Water”, “Crowd” and “Oberon”. Importing an audio sample into Wave 1 or Wave 2 is a new way to create new types of these complex wave sequences. But the term “wave sequence” in Oberon is slightly misleading, especially if you are accustomed to more traditional wave table based synthesizers.

In Oberon, when a sample is loaded, it is sliced in a number of shorter audio samples. Each slice is then analyzed to extract the various partials necessary to recreate the slice’s audio spectrum. In the end, just like the built-in “wave sequences” that we’ve talked about, a sample is transformed into a sequence of partial spectrum. The big difference between this method and traditional wave tables is that these slice spectrums don’t necessarily produce harmonic sounds.

The important thing to understand, is that the final spectrum sequence is manipulated inside Oberon just like the other wave sequences available, meaning there isn’t a default wave motion. If you reset your Oberon device and load a sample into Wave 1, playing a note will produce a sound that, unless you are particularly lucky, doesn’t resemble the imported audio. That’s because you are hearing the sound produced by the first partial spectrum in the sequence. To hear the complete original sample, we must play all the spectrums in sequence. The speed at which these spectrums are scanned determines the speed of the sample and we can thus play the sample at a different speed than its original form, while retaining the original pitch.

Playing the sample with loop

If you wish to play the entire sample and its loop section, you must modulate the start position of the oscillator playing the wave. You could set the oscillator’s motion to 100% but this will play the sample in cycle, restarting it at each cycle, without using its loop section. To play the sample only once while also playing its loop section, we can use the modulation envelopes. When a sample is loaded into Wave 1 or 2, a “Quick Envelope Setup” button on the left of the wave edit pages can be used to quickly setup a modulation envelope that correctly plays the sample. Click on this button and choose the envelope to use then go to the modulation matrix and link the envelope to the oscillator’s start position with a modulation strength of 100.

Stereo or multi-channel samples

Before analysis, all channels are mixed down to mono. The partials created after the analysis process are then spread out in the stereo field with Oberon’s special algorithm.

Very short samples

When importing very short samples (less than approximately 1000 samples), two situations can arise. If the sample doesn’t have a defined loop, it is treated “as is” and probably will be considered as having no defined pitched. The resulting partial spectrum will probably be inharmonic.

If the sample has a defined loop, that loop will be used in the analysis stage. This will probably result in a periodic wave form of some sort. If that loop period is short enough, it will appear as a clear pitch in the analysis. If you want to import specific **single cycle wave forms**, this is the way to go.

Very long samples

Oberon accepts samples of any size, provided you have enough memory on your computer, of course. But very long samples will suffer from poor time resolution. This starts to kick in when your samples get longer than approximately 1 minute at a 44100 sampling rate. The analysis will also take some time to complete, depending on your CPU power so expect Reason to freeze for a second or two if your samples are longer than 30 seconds, for instance.

Influence of audio sample rate

The sample analysis process in Oberon has been particularly tuned for samples at a sampling rate of 44100. Samples at a lower sampling rate will lose some time resolution but gain in low frequency resolution. In other hand, higher sample rates will gain in time resolution but lose in low frequency resolution. Thus, If you are importing really low bass sounds and the analysis gives a muddy inharmonic result, try lowering the sample rate of your file.

Influence of sample pitch

The pitch of the sample can play a big role in the quality of the analysis result. If the pitch is too high, your partial spectrums will lack in harmonic density. This will also cap out the higher partials as they do not exist in the signal.

If the pitch is too low, the analysis will have trouble detecting if the sound is pitched or not, probably ending into a non pitched one. Even worse, some low frequency partials will not be perceived by the algorithm or in a wrong way.

As a general rule, try using partials pitched to around C2 to C5.

MIDI Implementation Chart

MIDI Controller #	Parameter
4-5	Oscillator 1-2 Pan
7-8	Oscillator 1-2 Width
10 & 12	Oscillator 1-2 Pitch octaves
13-14	Oscillator 1-2 Pitch semi-tones
15-16	Oscillator 1-2 Pitch cents
17-18	Oscillator 1-2 Pitch keyboard tracking
19-20	Oscillator 1-2 Harmonicity
21-22	Oscillator 1-2 Harmonicity limit
23-24	Oscillator 1-2 Harmonicity octave
25-26	Oscillator 1-2 Bend
27-28	Oscillator 1-2 Unison mix
29-30	Oscillator 1-2 Unison pitch
31 & 33	Oscillator 1-2 Unison voice count
34-35	Oscillator 1-2 Unison phase lock
36-37	Oscillator 1-2 Unison phase random
39-40	Oscillator 1-2 Octave copy level
41-42	Oscillator 1-2 Partial clip
43-44	Oscillator 1-2 Level
45	Oscillator 2 On/Off
46-47	Oscillator 1-2 Morph target
48-49	Oscillator 1-2 Morph position
50-51	Oscillator 1-2 Morph target transpose
52-53	Oscillator 1-2 Wave
54-55	Oscillator 1-2 Wave sequence position
56-57	Oscillator 1-2 Wave sequence motion tempo sync.
58-59	Oscillator 1-2 Wave sequence key position reset

60-61	Oscillator 1-2 Wave sequence motion – free running
62-63	Oscillator 1-2 Wave sequence motion – synced
65	Amplitude Envelope – Point Count
66	Amplitude Envelope – Sustain Point Index
67-82	Amplitude Envelope – Point Levels 1-16
83-95	Amplitude Envelope – Section Durations 1-13
102-103	Amplitude Envelope – Section Durations 14-15
104-105	Filter 1-2 Frequency
106-107	Filter 1-2 X to Y
108-109	Filter 1-2 Frequency keyboard tracking
110-111	Filter 1-2 Frequency envelope modulation strength
112-113	Filter 1-2 Frequency velocity modulation strength
114-115	LFO 1-2 Key reset
116-117	LFO 1-2 Rate tempo sync.
118-119	LFO 1-2 Rate – free running
128-129	LFO 1-2 Rate – synced
130-131	LFO 1-2 Shape
132	Volume
133	Volume velocity modulation strength
134	Voice mode
135	Glide
136	Glide velocity modulation strength
137-142	Modulation strength for destination 1 of modulation matrix lines 1-6
143-148	Modulation strength for destination 2 of modulation matrix lines 1-6
149-154	Scale strength for modulation matrix lines 1-6
155-157	Modulation Envelopes 1-3 Release mode
158	Amplitude Envelope loop start point index
159-161	Modulation Envelopes 1-3 loop start point index

162 & 163	LFO 1 & 2 delay
164 & 165	LFO 1 & 2 rise
166 & 167	Wave 1 & 2 mode (shapes or sample)

Remote Map Template

Scope Zvork fr.zvork.Oberton

```
//Map _control_ Display Mode
//Map _control_ Osc. 1 - Stereo - Pan
//Map _control_ Osc. 2 - Stereo - Pan
//Map _control_ Osc. 1 - Stereo - Width
//Map _control_ Osc. 2 - Stereo - Width
//Map _control_ Osc. 1 - Pitch - Octave
//Map _control_ Osc. 2 - Pitch - Octave
//Map _control_ Osc. 1 - Pitch - Semitones
//Map _control_ Osc. 2 - Pitch - Semitones
//Map _control_ Osc. 1 - Pitch - Cents
//Map _control_ Osc. 2 - Pitch - Cents
//Map _control_ Osc. 1 - Pitch - Kbd. Tracking
//Map _control_ Osc. 2 - Pitch - Kbd. Tracking
//Map _control_ Osc. 1 - Harmonize
//Map _control_ Osc. 2 - Harmonize
//Map _control_ Osc. 1 - Harmonize Limit
//Map _control_ Osc. 2 - Harmonize Limit
//Map _control_ Osc. 1 - Harmonize Octave
//Map _control_ Osc. 2 - Harmonize Octave
//Map _control_ Osc. 1 - Bend
//Map _control_ Osc. 2 - Bend
//Map _control_ Osc. 1 - Unison - Mix
//Map _control_ Osc. 2 - Unison - Mix
//Map _control_ Osc. 1 - Unison - Pitch
//Map _control_ Osc. 2 - Unison - Pitch
//Map _control_ Osc. 1 - Unison - Voice Count
//Map _control_ Osc. 2 - Unison - Voice Count
//Map _control_ Osc. 1 - Unison - Phase Lock
//Map _control_ Osc. 2 - Unison - Phase Lock
//Map _control_ Osc. 1 - Unison - Random Phase
//Map _control_ Osc. 2 - Unison - Random Phase
//Map _control_ Osc. 1 - Octave Copy Level
//Map _control_ Osc. 2 - Octave Copy Level
//Map _control_ Osc. 1 - Partial Saturation
//Map _control_ Osc. 2 - Partial Saturation
//Map _control_ Osc. 1 - Filter Mode
//Map _control_ Osc. 2 - Filter Mode
//Map _control_ Osc. 1 - Level
//Map _control_ Osc. 2 - Level
//Map _control_ Osc. 2 - On/Off
```

//Map	_control_	Osc. 1 - Morphing Destination
//Map	_control_	Osc. 2 - Morphing Destination
//Map	_control_	Osc. 1 - Morphing Position
//Map	_control_	Osc. 2 - Morphing Position
//Map	_control_	Osc. 1 - Morphing Target Transpose
//Map	_control_	Osc. 2 - Morphing Target Transpose
//Map	_control_	Osc. 1 - Wave Type
//Map	_control_	Osc. 2 - Wave Type
//Map	_control_	Osc. 1 - Wave Start
//Map	_control_	Osc. 2 - Wave Start
//Map	_control_	Osc. 1 - Wave Motion Tempo Sync.
//Map	_control_	Osc. 2 - Wave Motion Tempo Sync.
//Map	_control_	Osc. 1 - Wave Motion Key Reset
//Map	_control_	Osc. 2 - Wave Motion Key Reset
//Map	_control_	Osc. 1 - Wave Motion
//Map	_control_	Osc. 2 - Wave Motion
//Map	_control_	Osc. 1 - Wave Motion Tempo Synced
//Map	_control_	Osc. 2 - Wave Motion Tempo Synced
//Map	_control_	Amp. Env. - Monohonic Trigger
//Map	_control_	Amp. Env. - Loop
//Map	_control_	Amp. Env. - Tempo Synced
//Map	_control_	Amp. Env. - Point Count
//Map	_control_	Amp. Env. - Sustain Point
//Map	_control_	Amp. Env. - Level 1
//Map	_control_	Amp. Env. - Level 2
//Map	_control_	Amp. Env. - Level 3
//Map	_control_	Amp. Env. - Level 4
//Map	_control_	Amp. Env. - Level 5
//Map	_control_	Amp. Env. - Level 6
//Map	_control_	Amp. Env. - Level 7
//Map	_control_	Amp. Env. - Level 8
//Map	_control_	Amp. Env. - Level 9
//Map	_control_	Amp. Env. - Level 10
//Map	_control_	Amp. Env. - Level 11
//Map	_control_	Amp. Env. - Level 12
//Map	_control_	Amp. Env. - Level 13
//Map	_control_	Amp. Env. - Level 14
//Map	_control_	Amp. Env. - Level 15
//Map	_control_	Amp. Env. - Level 16
//Map	_control_	Amp. Env. - Duration 1
//Map	_control_	Amp. Env. - Duration 2
//Map	_control_	Amp. Env. - Duration 3
//Map	_control_	Amp. Env. - Duration 4
//Map	_control_	Amp. Env. - Duration 5

//Map	_control_	Amp. Env. - Duration 6
//Map	_control_	Amp. Env. - Duration 7
//Map	_control_	Amp. Env. - Duration 8
//Map	_control_	Amp. Env. - Duration 9
//Map	_control_	Amp. Env. - Duration 10
//Map	_control_	Amp. Env. - Duration 11
//Map	_control_	Amp. Env. - Duration 12
//Map	_control_	Amp. Env. - Duration 13
//Map	_control_	Amp. Env. - Duration 14
//Map	_control_	Amp. Env. - Duration 15
//Map	_control_	Mod. Env. 1 - Monophonic
//Map	_control_	Mod. Env. 2 - Monophonic
//Map	_control_	Mod. Env. 3 - Monophonic
//Map	_control_	Mod. Env. 1 - Loop
//Map	_control_	Mod. Env. 2 - Loop
//Map	_control_	Mod. Env. 3 - Loop
//Map	_control_	Mod. Env. 1 - Tempo Synced
//Map	_control_	Mod. Env. 2 - Tempo Synced
//Map	_control_	Mod. Env. 3 - Tempo Synced
//Map	_control_	Mod. Env. 1 - Point Count
//Map	_control_	Mod. Env. 2 - Point Count
//Map	_control_	Mod. Env. 3 - Point Count
//Map	_control_	Mod. Env. 1 - Sustain Point
//Map	_control_	Mod. Env. 2 - Sustain Point
//Map	_control_	Mod. Env. 3 - Sustain Point
//Map	_control_	Mod. Env. 1 - Level 1
//Map	_control_	Mod. Env. 1 - Level 2
//Map	_control_	Mod. Env. 1 - Level 3
//Map	_control_	Mod. Env. 1 - Level 4
//Map	_control_	Mod. Env. 1 - Level 5
//Map	_control_	Mod. Env. 1 - Level 6
//Map	_control_	Mod. Env. 1 - Level 7
//Map	_control_	Mod. Env. 1 - Level 8
//Map	_control_	Mod. Env. 1 - Level 9
//Map	_control_	Mod. Env. 1 - Level 10
//Map	_control_	Mod. Env. 1 - Level 11
//Map	_control_	Mod. Env. 1 - Level 12
//Map	_control_	Mod. Env. 1 - Level 13
//Map	_control_	Mod. Env. 1 - Level 14
//Map	_control_	Mod. Env. 1 - Level 15
//Map	_control_	Mod. Env. 1 - Level 16
//Map	_control_	Mod. Env. 2 - Level 1
//Map	_control_	Mod. Env. 2 - Level 2
//Map	_control_	Mod. Env. 2 - Level 3

//Map	_control_	Mod. Env. 2 - Level 4
//Map	_control_	Mod. Env. 2 - Level 5
//Map	_control_	Mod. Env. 2 - Level 6
//Map	_control_	Mod. Env. 2 - Level 7
//Map	_control_	Mod. Env. 2 - Level 8
//Map	_control_	Mod. Env. 2 - Level 9
//Map	_control_	Mod. Env. 2 - Level 10
//Map	_control_	Mod. Env. 2 - Level 11
//Map	_control_	Mod. Env. 2 - Level 12
//Map	_control_	Mod. Env. 2 - Level 13
//Map	_control_	Mod. Env. 2 - Level 14
//Map	_control_	Mod. Env. 2 - Level 15
//Map	_control_	Mod. Env. 2 - Level 16
//Map	_control_	Mod. Env. 3 - Level 1
//Map	_control_	Mod. Env. 3 - Level 2
//Map	_control_	Mod. Env. 3 - Level 3
//Map	_control_	Mod. Env. 3 - Level 4
//Map	_control_	Mod. Env. 3 - Level 5
//Map	_control_	Mod. Env. 3 - Level 6
//Map	_control_	Mod. Env. 3 - Level 7
//Map	_control_	Mod. Env. 3 - Level 8
//Map	_control_	Mod. Env. 3 - Level 9
//Map	_control_	Mod. Env. 3 - Level 10
//Map	_control_	Mod. Env. 3 - Level 11
//Map	_control_	Mod. Env. 3 - Level 12
//Map	_control_	Mod. Env. 3 - Level 13
//Map	_control_	Mod. Env. 3 - Level 14
//Map	_control_	Mod. Env. 3 - Level 15
//Map	_control_	Mod. Env. 3 - Level 16
//Map	_control_	Mod. Env. 1 - Time 1
//Map	_control_	Mod. Env. 1 - Time 2
//Map	_control_	Mod. Env. 1 - Time 3
//Map	_control_	Mod. Env. 1 - Time 4
//Map	_control_	Mod. Env. 1 - Time 5
//Map	_control_	Mod. Env. 1 - Time 6
//Map	_control_	Mod. Env. 1 - Time 7
//Map	_control_	Mod. Env. 1 - Time 8
//Map	_control_	Mod. Env. 1 - Time 9
//Map	_control_	Mod. Env. 1 - Time 10
//Map	_control_	Mod. Env. 1 - Time 11
//Map	_control_	Mod. Env. 1 - Time 12
//Map	_control_	Mod. Env. 1 - Time 13
//Map	_control_	Mod. Env. 1 - Time 14
//Map	_control_	Mod. Env. 1 - Time 15

//Map	_control_	Mod. Env. 2 - Time 1
//Map	_control_	Mod. Env. 2 - Time 2
//Map	_control_	Mod. Env. 2 - Time 3
//Map	_control_	Mod. Env. 2 - Time 4
//Map	_control_	Mod. Env. 2 - Time 5
//Map	_control_	Mod. Env. 2 - Time 6
//Map	_control_	Mod. Env. 2 - Time 7
//Map	_control_	Mod. Env. 2 - Time 8
//Map	_control_	Mod. Env. 2 - Time 9
//Map	_control_	Mod. Env. 2 - Time 10
//Map	_control_	Mod. Env. 2 - Time 11
//Map	_control_	Mod. Env. 2 - Time 12
//Map	_control_	Mod. Env. 2 - Time 13
//Map	_control_	Mod. Env. 2 - Time 14
//Map	_control_	Mod. Env. 2 - Time 15
//Map	_control_	Mod. Env. 3 - Time 1
//Map	_control_	Mod. Env. 3 - Time 2
//Map	_control_	Mod. Env. 3 - Time 3
//Map	_control_	Mod. Env. 3 - Time 4
//Map	_control_	Mod. Env. 3 - Time 5
//Map	_control_	Mod. Env. 3 - Time 6
//Map	_control_	Mod. Env. 3 - Time 7
//Map	_control_	Mod. Env. 3 - Time 8
//Map	_control_	Mod. Env. 3 - Time 9
//Map	_control_	Mod. Env. 3 - Time 10
//Map	_control_	Mod. Env. 3 - Time 11
//Map	_control_	Mod. Env. 3 - Time 12
//Map	_control_	Mod. Env. 3 - Time 13
//Map	_control_	Mod. Env. 3 - Time 14
//Map	_control_	Mod. Env. 3 - Time 15
//Map	_control_	Filter 1 Frequency
//Map	_control_	Filter 2 Frequency
//Map	_control_	Filter 1 X>Y
//Map	_control_	Filter 2 X>Y
//Map	_control_	Filter 1 Frequency - Kbd. Tracking
//Map	_control_	Filter 2 Frequency - Kbd. Tracking
//Map	_control_	Filter 1 Frequency - Mod. Env. 1 Strength
//Map	_control_	Filter 2 Frequency - Mod. Env. 2 Strength
//Map	_control_	Filter 1 Frequency - Velocity Modulation
//Map	_control_	Filter 2 Frequency - Velocity Modulation
//Map	_control_	LFO 1 Key Sync.
//Map	_control_	LFO 2 Key Sync.
//Map	_control_	LFO 1 Tempo Sync.
//Map	_control_	LFO 2 Tempo Sync.

//Map	_control_	LFO 1 Frequency
//Map	_control_	LFO 2 Frequency
//Map	_control_	LFO 1 Freq. Synced
//Map	_control_	LFO 2 Freq. Synced
//Map	_control_	LFO 1 Shape
//Map	_control_	LFO 2 Shape
//Map	_control_	Volume
//Map	_control_	Volume Velocity Modulation
//Map	_control_	Voice Mode
//Map	_control_	Glide
//Map	_control_	Glide Velocity Mod.
//Map	_control_	Pitch Bend Range
//Map	_control_	Mod. 1 - Source
//Map	_control_	Mod. 2 - Source
//Map	_control_	Mod. 3 - Source
//Map	_control_	Mod. 4 - Source
//Map	_control_	Mod. 5 - Source
//Map	_control_	Mod. 6 - Source
//Map	_control_	Mod. 1 - Destination 1
//Map	_control_	Mod. 2 - Destination 1
//Map	_control_	Mod. 3 - Destination 1
//Map	_control_	Mod. 4 - Destination 1
//Map	_control_	Mod. 5 - Destination 1
//Map	_control_	Mod. 6 - Destination 1
//Map	_control_	Mod. 1 - Destination 2
//Map	_control_	Mod. 2 - Destination 2
//Map	_control_	Mod. 3 - Destination 2
//Map	_control_	Mod. 4 - Destination 2
//Map	_control_	Mod. 5 - Destination 2
//Map	_control_	Mod. 6 - Destination 2
//Map	_control_	Mod. 1 - Strength 1
//Map	_control_	Mod. 2 - Strength 1
//Map	_control_	Mod. 3 - Strength 1
//Map	_control_	Mod. 4 - Strength 1
//Map	_control_	Mod. 5 - Strength 1
//Map	_control_	Mod. 6 - Strength 1
//Map	_control_	Mod. 1 - Strength 2
//Map	_control_	Mod. 2 - Strength 2
//Map	_control_	Mod. 3 - Strength 2
//Map	_control_	Mod. 4 - Strength 2
//Map	_control_	Mod. 5 - Strength 2
//Map	_control_	Mod. 6 - Strength 2
//Map	_control_	Mod. 1 - Scale Strength
//Map	_control_	Mod. 2 - Scale Strength

//Map	_control_	Mod. 3 - Scale Strength
//Map	_control_	Mod. 4 - Scale Strength
//Map	_control_	Mod. 5 - Scale Strength
//Map	_control_	Mod. 6 - Scale Strength
//Map	_control_	Mod. 1 - Scale Source
//Map	_control_	Mod. 2 - Scale Source
//Map	_control_	Mod. 3 - Scale Source
//Map	_control_	Mod. 4 - Scale Source
//Map	_control_	Mod. 5 - Scale Source
//Map	_control_	Mod. 6 - Scale Source
//Map	_control_	Mod. Env. 1 - Release Mode
//Map	_control_	Mod. Env. 2 - Release Mode
//Map	_control_	Mod. Env. 3 - Release Mode
//Map	_control_	Amp. Env. - Loop Start Point
//Map	_control_	Mod. Env. 1 - Loop Start Point
//Map	_control_	Mod. Env. 2 - Loop Start Point
//Map	_control_	Mod. Env. 3 - Loop Start Point
//Map	_control_	LFO 1 Delay
//Map	_control_	LFO 2 Delay
//Map	_control_	LFO 1 Rise
//Map	_control_	LFO 2 Rise
//Map	_control_	Wave 1 Mode
//Map	_control_	Wave 2 Mode

More information at <http://www.zvork.fr/audio>

© 2015-2017 Olivier Prat