The MiMi-d synthesizer

Ricard Wanderlöf November 1, 2024

Manual version 1.0.4
Plugin version 2.0.5

Introduction

The MiMi-d is a virtual analog synthesizer in the form of a plugin, designed primarily as an LV2[1] synth engine for the Zynthian platform, although the plugin itself is not tied to any specific hardware. The plugin has no GUI of its own, and thus relies completely on the host for setting parameters, as well as saving and loading patches.

Background

The MiMi-d stems from an analog polyphonic synthesizer named the MiMi-a which I built as a one-off project in 1987-1988. It was loosely based on my desire for a Memorymoog, although in the end the resulting instrument turned out rather different. I had previously built an all-analog monophonic synthesizer, the Delta Seven. As digital technology became available, many times I had been thinking of creating a digital equivalent, but the project did not come together until I discovered the OB-Xd plugin, and upon examination of its source code realized that it would be within reach to use and modify the internal modules to create a digital model of the MiMi-a. The OB-Xd has, above all, excellent oscillator and filter implementations, but I've never been fond of the architecture of the classic Oberheim synthesizers, and the OB-Xd architecture, being modeled on them, was no exception.

So, I present the MiMi-d, which is intended not to be an accurate model of the MiMi-a, but rather an instrument in the same spirit, drawing from my long-time experiences of building and playing synthesizers, while using all the bells and whistles that can be gained from a digital implementation. Some features, such as variable sync, have been modeled on the Delta Seven. It is not intended to be the be-all and end-all of virtual analog synthesizers, rather, it's a nod to synthesizers with compact yet eminently usable architectures such as the Prophet-5. In other words, a go-to plugin for standard subtractive synthesis needs, with various bonus features thrown in along the way.

Architecture

The MiMi-d is a two oscillator subtractive synthesizer, with additional sub oscillator, mixed together and feeding a 24 dB/octave variable-slope resonant ladder type low pass filter, followed by a non-resonant 6 dB/octave high pass filter and finally an output VCA. For modulation/control there are two ADSSR envelope generators, one for the filter and one for the VCA, and two modulation generators which can act as LFOs or supplementary AD envelope generators.

What sets the MiMi-d apart from other instruments are various details and modulation paths:

- Frequency modulation from oscillator 2 to oscillator 1

1 The framework used (Distrho Plugin Framework) additionally has support for the DSSI, VST2, VST3 and CLAP formats, as well as a standalone JACK binary, although primary development focus has been the LV2 format for use in Zynthian.

- Frequency modulation from oscillator 2 to the filter
- Variable sync level between the oscillators
- Not only oscillator frequency, pulse width, and filter frequency can be modulated, but also filter resonance and the amount of oscillator to filter modulation
- VCA overdrive, modeling the overdriven sound of a CA3080 OTA often used as a VCA chip in analog synthesizers such as the MiMi-a
- Bipolar filter envelope – no more running out of the low end of the filter cutoff control when applying envelope modulation
- Faithful reproduction of the CEM3310 and SSM2050/2055/2056 envelope shapes, as well as a completely linear curve mode which models the MiMi-a envelopes (and other synthesizers with trapezoid envelope shapes like the VCS3)
- The envelopes are ADSSR types, with a sustain phase that can be constant as in a traditional ADSR or work like a second decay
- Per-voice LFOs, where each voice can have its own phase. When the LFO waveform is set to sample-and-hold, each voice will has a different different random value for each cycle.

The MiMi-d owes much of its existence to the OB-Xd and its designer Filatov Vadim from which a lot of the DSP code stems, in particular the excellent alias reduction implementation in the oscillators and the filter implementation, and also the use of randomness in the appropriate places to model analog behavior.

Parameters

All parameters have been divided into groups of four to be easily accessible using the Zynthian GUI.

Main

General parameters: **Volume**, **Master Tune** (+/- one semitone), **Portamento Tine** and **Octave**.

Key assignment

The MiMi-a had quite a comprehensive and configurable key assigner, which has been carried over and augmented in the MiMi-d. Although voice allocation can be of lesser importance in an software instrument which has neigh infinite number of voices, it is important when modeling an instrument which has a fairly low voice count.

Key assignment Page #1

Voice count – How many notes can be played at once in Poly mode; in the Mono modes, this controls how many voices are played in unison..

Assign Mode – **Poly** (Polyphonic), **Mono** (Monophonic), **AutoP** (Monophonic with auto portamento, i.e. portamento is only engaged when playing legato)

Envelope Attack – Governs whether the envelopes curves start where they happened to leave off after the voice was used previously (**FreeRun**), or always reset to zero (**KeyReset**).

Env Retrig – When a voice that is playing is reused for a new note, by default (**All**) it will always retrigger the envelopes. Alternatively, it will retrigger for note on events but not for note off events (**Note On**), or neither for note on nor note off events (**None**). This is mostly useful in Mono modes but operates in an extrapolated fashion also in Poly mode.

Key assignment Page #2

Assign Order – The basic order in which voices are allocated. Cyclic mode (**Cyc**) is most common, where the voice chosen is the one that hasn't been pressed for the longest time. Alternatively, Reset-to-zero mode (**RSZ**) means that the lowest numbered non playing voice will be selected at all times. This tends to cut off already playing voices, allowing for long release times in Poly mode yet still having the release cut short when new notes are played like a monophonic synthesizer would behave.

Assign Memory – Governs whether repeatedly playing the same note will attempt to allocate a voice governed by the Assign Order (**Off**) or allocate the same voice used for the note previously (**Mem**). This can be useful for instance when modeling a stringed instrument with a long release, where there is actually only one string (or set of strings) per note.

Voice Rob – When there are no free voices because the number of notes being played is equal to the Voice Count, the Voice Rob parameter governs whether new notes will be ignored (**Off** – but see the Voice Restore parameter below), or the oldest playing voice will be reused (**Rob O**, short for Rob Oldest), or alternatively, the oldest playing voice but avoiding robbing the voice with the lowest pitch (**Rob NL**, short for Rob Next Lowest; NL is also a nod towards the Nord Lead which has a similar voice robbing strategy).

Voice Restore – If more notes are played than there are available voices, the Voice Restore parameter governs whether the excess notes will be forgotten (**Off**), or saved in a list and played once voices become available (**Res**). This is most useful in Mono modes, but can be useful in Poly mode as well, for instance when playing strings and one doesn't want notes played in excess of the voice count to reduce the number of notes playing once they have been released. In Mono mode it models the classic last-note-priority of a monophonic synthesizer, where the pitch jumps back to a previously held note when the last played note is released.

Recommended key assign settings

Having so many key assignment settings can be daunting, as most synths have one or two, but most of the time they can be left at their default settings.

For polyphonic playing (**Assign Mode** = ***Poly***): **Assign Order** = ***Cyc***, **Assign Memory** = ***Mem***, **Voice Rob** = ***Rob NL,*** **Voice Restore** = ***Off*** and **Env Retrig** = ***All*** (these are the defaults). Experiment with setting **Assign Memory** to ***Off*** in order to get repeated notes played to trigger a new voice every time.

When playing monophonically (**Assign Mode** ***Poly*** or ***AutoP***), keep these settings but set **Voice Restore** to ***On*** to get the classic monosynth bounce-back-to-the-held-note behavior when holding one key and playing and releasing a second one. Setting **Env Retrig** to ***None*** will implement the behavior of synthesizers like the Minimoog, or a Pro-One in NORMAL mode, where the envelopes

are only retriggered if all keys are released before pressing a new one, sometimes termed ‘legato mode’. Setting the parameter to ***All*** will cause the envelopes to retrigger whenever a new note is assigned, whether it is because a new key is pressed, or because a key is released when another one is held (when **Restore** is ***On***), causing the pitch revert back to the key that is held. This is sometimes termed ‘retrig mode’ on monophonic synthesizers.

Bend

Bend Range – The bend range can be set in steps of semitones, up to a complete octave.

Dest – Bend destination: **Off**, **Osc 1** or **Osc 1+2**. The Osc1 position is useful when oscillator 1 is hard synced to oscillator 2.

Modulator #1

In the MiMi-a, there were two sources of modulation: an LFO, and a General Envelope Generator which additionally could be set in repeat mode. The LFO was permanently wired to the mod wheel. In the MiMi-d, there are two modulators, which basically are LFOs which additionally can run in one-shot mode, whereby the waveform stops after one full cycle, and it is possible to select what type of controller (mod wheel, aftertouch or velocity (or none)) will control the modulation amount.

Modulator #1 page #1

Rate – Speed in LFO mode, or envelope rate in one shot mode

Wave – There are a number of LFO waveforms. A lot of them like **Tri**, **Saw**, Reverse Saw (**RSaw**), Square (**Squ**) are obvious, but there are a couple of special was forms. The **Rise** waveform is like a (rising) sawtooth, except that the waveform reset is not instantaneous, being set to 10% of the total waveform. The **Fall** waveform is the same, but reversed. Pulse (**Pul**) is a pulse wave with 25% duty cycle. Finally sample-and-hold (**S/H**) is a stepped random waveform, which delivers a new value for every cycle.

Initial Amount – Amount of modulation

Dest – Modulation destination. Most are self-explanatory (**Off**, **Osc 1**, **Osc 1+2**, **Osc 2**, **PW 1**, **PW 1+2**, **PW 2**, **Cutoff**. In addition, the MiMi-d offers modulation of the filter resonance (**Res**) as well as the amount of oscillator-to-filter modulation (**Osc2Filt**).

Modulator #1 page #2

Controller Amount – The amount of modulation is set by the Initial Amount on the previous page, but additionally a controller (modwheel, aftertouch or velocity) can be used to add to the initial modulation amount. This parameter sets the amount of additional modulation provided by the controller

Controller – Selects the controller to affect the modulation amount using the Controller Amount parameter (**Off**, **Modw**, **Aftert** or **Vel**).

Polarity – For many LFO uses, the **Normal** (bipolar) polarity is best, such as Vibrato. The waveform may be inverted in the **Inv** mode. **Uni** is a unipolar mode, where the modulation excursions are always positive, and **InvUni** is an inverted unipolar mode, where the modulation excursion is always negative.

Sync – Synchronization mode. In the **Off** position, the modulator functions as a free running LFO. All voices share the same frequency and phase. The **Clock** position synchronizes the LFO to the currently playing MIDI tempo (see below). The **Key** sync mode restarts the LFO every time a key is pressed. Since there is a separate LFO per voice, this means that the LFOs will potentially be out of phase if the voices are triggered at different times. Finally, **OneShot** mode will allow the modulator to run one full cycle and then stop, in effect turning it into an AD envelope generator.

There are a couple of subtle differences in the waveforms between the LFO and OneShot modes. The triangle waveform in LFO modes starts at the central point in the waveform, whereas in OneShot mode, the triangle waveform starts at its lowest point, modeling an AD envelope. For the Sawtooth waveform, in OneShot mode the waveform does not reset but stays at its maximum value after the cycle is completed, in effect modeling an AR envelope with infinite release time.

Clock sync

When the **Sync** parameter is set to **Clock**, the LFO is hard synchronized to the tempo supplied by the host. In this mode, the **Rate** parameter takes on a different meaning: instead of setting the absolute frequency of the LFO, it sets the ratio of the LFO to the beat rate, in integer steps (e.g. setting the parameter to 4.85 (see the table below) gets rounded to 5, which means that the LFO cycle will be one beat long, or one quarter note in 4/4 or 3/4 time).

Parameter value (range)	Ratio to beat	LFO cycle length for quarter note beats (e.g. 4/4 or 3/4 time signatures)
0 (0-0.5)	8:1	Breve (eight quarter notes)
1 (0.5-1.5)	6:1	Six quarter notes
2 (1.5-2.5)	4:1	Whole note (four quarter notes)
3 (2.5-3.5)	3:1	Three quarter notes
4 (3.5-4.5)	2:1	Half note (two quarter notes)
5 (4.5-5.5)	1:1	Quarter note
6 (5.5-6.5)	2:3	Quarter note triplet
7 (6.5-7.5)	1:2	Eighth note
8 (7.5-8.5)	1:3	Eighth note triplet
9 (8.5-9.5)	1:4	Sixteenth note
10 (9.5-10)	1:8	Thirtysecondth note

Table 1. LFO tempo sync ratios

DC mode

When the modulator rate is set to 0, the modulator output will be static (DC) (unless the modulator is in clock sync mode; see above). Setting the waveform to square (Squ) with Sync in the Key position will cause the modulator to output a DC value which can be controlled by the Initial Amount, Controller and Controller Amount parameters. In this way, controllers such as aftertouch can directly be used to control for instance the filter.

If the waveform is set to S/H with Sync in the Key position, a new random value will be output every time a key is pressed.

Modulator #2

Modulator #2 is identical to Modulator #1 in every respect.

Oscillator 1

In the MiMi-d, as in the MiMi-a and Delta Seven, oscillator 1 is the slave oscillator, which receives FM modulation and sync from oscillator 2. In other respects, it is identical to oscillator 2.

Detune – Detunes the oscillator by up to +/-1 semitone. (In order to detune further, use the assistance of the **Pitch** parameter).

Pitch – Oscillator pitch, from 0 to 72 semitones. At an oscillator pitch setting of 24, a MIDI note number of 60 will result in the oscillator pitch being Middle C (261.63 Hz) (when the global octave setting is 0).

PulseWidth – Sets the width of the Pulse waveform from 50% to 100%. No effect on other waveforms, but it may be future proof to leave the parameter at its minimum value in the event that additional waveshaping is implemented for the other waveforms in the future

Wave – Waveform. **Off**, Sawtooth (**Saw**), Pulse (**Pul**) or Triangle (**Tri**)

Oscillator 2

Apart from its function as a master oscillator, oscillator 2 is identical to oscillator 1 and has the same parameters.

Osc Common

Parameters which govern the interaction between the oscillators.

Xmod – Modulation from oscillator 2 to oscillator 1. The modulation takes place using the ordinary exponential input of the oscillator. Normally this means that there will be appreciable pitch shift as the modulation is advanced. In the MiMi-d this has been compensated to some extent, so that the pitch shift is zero when the modulation amount is set to max. For intermediate values there is a slight remaining pitch shift which can be compensated for using the Detune and Pitch parameters for Oscillator 1.

Osc2filterMod – Audio modulation of the filter cutoff frequency by oscillator 2. This is a slightly unusual modulation path (especially in digital synthesizers) which can lead to a wide variety of sounds.

SyncLevel – Synchronizes oscillator 1 to oscillator 2. The sync level is variable, from no sync at all to full conventional hard sync. In the intermediate settings, the oscillator 1 waveform is only reset if it happens to be lower than sync level value when the reset pulse comes from oscillator 2. One use of this is to create pseudo arpeggio sequences by setting oscillator 2 to a higher frequency than oscillator 1 and modulating oscillator 2 with an LFO.

Waveform Reset – In a conventional analog synthesizer, the oscillators are free-running, and thus when a voice is triggered, they can be at any phase in the waveform. This corresponds to the

FreeRun setting. In the **KeySync** setting, the waveforms of the oscillators resets every time the voice is triggered. This creates a ‘static’ sound when voices are retriggered which can be useful in some cases, for instance to model a stringed instrument where the oscillating movement of the strings start out at the same state ever time they are struck or plucked.

Mixer (and sub oscillator)

This is the audio mixer for the oscillators, and this page also includes the settings for the sub oscillator slaved to oscillator 2.

Osc1Mix – level of oscillator 1

Osc2Mix – level of oscillator 2

Osc2Sub Mix – level of sub oscillator

Osc2Sub Wave – Sub oscillator waveform and ratio to oscillator 2. This is a classic sub oscillator, proving square waves at 1 (**-1 Squ**) and 2 (**-2 Squ**) octaves below the oscillator 2, as well as a 25% pulse wave 2 (**-2 Pul**) octaves below. Alternatively, although technically not a sub oscillator waveform, a **Noise** source can be selected. (The noise waveform is slightly unusual in that rather than being a model of an analog noise source, it models the digital output of a pseudo-random number generator. This is the type of noise generator employed in the MiMi-a, and has the advantage that it has a higher RMS (or in other words, perceived loudness) value for the same peak value than classic analog noise.)

Filter

The filter in the MiMi-d is modeled on a classic ladder type open loop four pole (24 dB/octave) low pass filter, with pre-set gain compensation when the Resonance control is advanced. It is based on the Zero Delay Feedback design pioneered by Vadim Zavalashin in his book “The Art of VA Filter Design”, providing a faithful reproduction of the classic 24 dB/octave filter type used by many manufacturers such as Moog, Sequential and Oberheim. It can handle audio rate modulation, and oscillates nicely when the feedback (Resonance control) is advanced far enough. Additionally, it is possible to continuously vary the filter output from 6 dB/octave to 24 db/octave. (Technically, the DSP code originates from the 4 pole filter in the OB-Xd, with a number of tweaks and modifications).

Cutoff – The Big Red Knob of all subtractive synthesizers, setting the cutoff frequency of the filter. The range is roughly 20 Hz to 20 kHz, with a change in the parameter value of 1.0 corresponding to exactly one octave (e.g. a cutoff frequency setting of 6 is one octave up from a setting of 5).

Resonance – Filter resonance control. At high settings (above 0.95) the filter will self oscillate, producing a sine wave.

Key Track – Keyboard filter tracking, adjustable from 0 (no tracking) to 1 (1:1 tracking). The base note for the key tracking is Eb2 (MIDI note 39), i.e. when this note is played, the Key Track setting has no effect.

EnvAmount – Amount of filter envelope modulation.

Filter Configuration

Sets various ancillary parameters governing the filter and the filter envelope.

Pole Count – Continuously variable pole count from 1 to 4 (6 dB/octave to 24 dB/octave). Technically, there are only four available outputs from the filter, one for each filter stage (pole); setting intermediate values of the parameter combines the outputs from adjacent stages.

Env Invert – Inverts the filter envelope.

Linear Env – Sets a linear envelope shape instead of the standard exponential envelope shape.

VCA

Technically, the VCA is the final signal processing point in a voice, controlled by the Loudness Envelope. On the VCA parameter page, a couple of VCA and VCA-related parameters have been collected.

HPF Freq – This controls a non resonant 6 dB/octave high pass filter which is inserted between the lowpass filter and the VCA. The control range (0 to 10) corresponds to HPF cutoff frequencies from 4 Hz (minimum) to 2500 Hz (maximum). The fact that the HPF is always in the signal path models the coupling capacitors used in analog electronics.

VCA Drive – The MiMi-a used a CA3080 chip for its VCA, and for various reasons, it is driven quite hard, leading to a subtle but eminently audible waveform distortion at high volume levels. The Minimoog VCA behaves in a similar way. In the MiMi-d the amount of ‘drive’ can be controlled from none at all (minimum) to a fairly severe waveform flattening at the maximum setting, although even at this setting, the signal is not in fact clipped. Note that this ‘distortion’ happens before the VCA but after the filter, thus, its effect is not filtered by any subsequent lowpass filter, and rather can add some ‘zing’ to an otherwise heavily lowpass filtered sound.

Env/VCA Mode – Most synthesizers employ a VCA with linear response, coupled with envelope generators employing exponential curves. This is modeled in the MiMi-d when the EnvMode parameter is in the (default) **Exp/Lin** setting. In the **Lin/Lin** setting, the VCA remains linear, but a linear envelope curve is employed. This models the MiMi-a ‘Linear VCA’ mode, and above all results in the decay and release phases seemingly not doing much until suddenly the level drops drastically. Finally the **Lin/Exp** setting models the MiMi-a ‘Exponential VCA’ mode, with employs a linear envelope curve but exponential VCA response. This is an attempt at achieving an exponential loudness response with linear envelopes. Since the exponentiation occurs in the VCA rather than the envelope, the Sustain (level) control has different impact than in Exp mode.

Filter Envelope

Classic ADSR envelope with an additional sustain time parameter (thus, an ADSSR envelope). When the sustain time is set above 9.91, the sustain time becomes infinite, and the envelope reverts to a classic ADSR type.

The exponential mode of the envelope generator is modeled on a classic analog style envelope; the attack asymptote is 1.3 times the maximum value (modeling the CEM3310 and SSM2050/50/56 chips and other, discrete, analog envelope generators), whereas the decay asymptote is the sustain level and the sustain and release asymptotes are 0. This results in a nearly linear attack curve, while

the decay, sustain and release phases are classic exponential decay curves. This tends to be the most natural sounding and useful envelope shape in a synthesizer.

Additionally, a linear mode can be selected on the Filter Config page, which implements a trapezoid envelope shape, with linear segments. This models the envelope generators used in the MiMi-a.

Since the envelope has five parameters, but each parameter page in Zynthian has only four, the **Attack**, **Decay**, **Sustain** and **SustainTime** parameters are on the Filter Envelope page, whereas the Release parameter is, together with the Loudness Envelope Release parameter, on a separate Env Release page.

Loudness Envelope

The Loudness Envelope and Filter Envelope are functionality identical and have the same **Attack**, **Decay**, **Sustain** and **SustainTime** pages with the Release parameter being on a separate Env Release page.

Env Release

This page contains the **Release** parameters for the Filter and Loudness envelope that didn't fit on the respective envelope pages. Actually, it can be quite nice to have these grouped together, as it allows for quick tailoring of the release portion of the sound.

Controller Sens.

The controller sensitivity page contains controls for the velocity and aftertouch response. The MiMi-d currently only responds to channel aftertouch, not polyphonic aftertouch.

Velocity Scale – At its medium (5) setting, the velocity values are passed through in a linear fashion. At low settings, a scaling is applied which makes low values more prominent, in other words, more velocity must be applied to get higher resulting velocity value. At high settings, the opposite is true. Note that all velocity values are still eminently reachable no matter which value the parameter has; the scaling is achieved by bending the transfer curve (rather than adding an offset). Technically, an increasing mathematical power function is employed, with negative powers for higher parameter values and positive powers for lower values.

AfterTouch Scale – Functionally equivalent to the Velocity Scale, but operating on aftertouch.

FilterEnv Velocity – This sets how much the filter envelope amount is controlled by velocity. Note that the control is bipolar: When medium velocity (MIDI value 64) is applied, the envelope amount remains unchanged. At higher velocity levels, the amount is increased, and at lower velocity levels, the amount is decreased. This tends to keep the average modulation amount constant (although the 'medium' value will vary depending on the Velocity Scale parameter), so that it's not necessary to 'chase' the velocity control with the filter cutoff setting.

Amp Velocity - This sets how much the modulation level of the loudness envelope controlling the VCA is affected by velocity. Unlike the filter, this is not a bipolar operation; with increasing Amp Velocity, the average level will drop, being the same only at maximum velocity values.

Spread

The spread parameters contain the amount of random deviation of different facets of the synthesizer behavior, in order to model component tolerances and maladjustment of analog circuitry, causing various variations from voice to voice. Rather than have a global ‘analog slop’ parameter randomly affecting everything, here one has control over the degree of randomness for a number of key parameters.

The actual deviation is calculated when the plugin is initiated and remains constant after that, to model the behavior of component tolerances or maladjusted calibration, rather than having a general random value that either varies all the time or every time a new note is triggered. (Analog electronics tend to be very stable in the short term.) The result is the characteristic cyclic behavior of such deviations as the different voices are cycled through due to key assignment.

Spread #1

OscSpread – How much the individual oscillators deviate from each other in terms of pitch.

FilterSpread – How much the filter tuning varies from voice to voice.

LevelSpread – How much the output level from the individual voices deviate from each other.

PanSpread – How much the panning of the individual voices varies from voice to voice. (This parameter harks back to the OB-Xd which in turn models the OB-Xa individual pan controls for the different voices, but rather than have 8 controls for the different voices, here they are all set randomly, which is the best way to achieve a stereo width anyway.)

Spread #2

EnvelopeSpread – How much the envelopes deviate from each other in terms of timing. A single random value is calculated for every envelope, not for every time parameter, to model analog component tolerances which tend to be most prevalent when it comes to the capacitor used for the timing circuitry.

PortamentoSpread – How much the portamento time deviates from voice to voice.

LfoSpread – How much the modulator rates deviate from voice to voice and modulator to modulator.

DSP Control

Oversample – Enables two times oversampling when enabled (default). This results in twice the DSP load of the plugin compared to when oversampling is disabled, but significantly reduces aliasing artifacts when features such as cross modulation, filter modulation by oscillator at high resonance settings, or VCA overdrive is applied, which all tend to generate a lot of high frequency harmonics. For standard subtractive synthesis with just a swept filter and no VCA overdrive, switching off this parameter will have little or no impact on the sound, and reduces the DSP load by half. A suggestion if you are not sure how to set this parameter is to edit your sound with Oversample turned on (this is the default anyway, and also the setting for all Factory presets), then if you find you’re running out of DSP capacity (clicking or notes being dropped), try turning it off to see if it has any impact on your sound in the range in which you want to use it (digital artifacts generally get worse the higher the pitch is).

Economy Mode – When set (default), the bulk of DSP code for the individual voices is only run when the voice is actually producing sound, thus keeping the CPU load on the system to minimum. When set to off, the DSP code for all voices is run at all times, resulting in a higher but more consistent load on the system. (There will still be some variation as for instance the antialiasing algorithms on the oscillators consume more DSP the higher the oscillator pitch is). There shouldn't be any practical difference in terms of sound, with one exception: When the filter is set to self oscillate, it takes time for the oscillation to build up. This is not noticeable when working on a patch and advancing the Resonance control, but when loading a patch where the filter is set to self oscillate (and it was not set to oscillate in the previous patch), there is a slight delay when each voice is played for the first time after loading the patch before the filter breaks into oscillation, as if the loudness attack had been set to a fraction of a second.

Trademarks

Oberheim, OB-Xa, Moog, Memorymoog, Nord Lead, Sequential, Prophet, Pro-One and VCS3 are trademarks of their respective companies, and are used for reference only. No infringement is intended.

Thanks to

Bob Moog, Dave Smith, Tom Oberheim and all the other pioneers in the synthesizer industry.

José Fernando for creating and maintaining the Zynthian project.

Brian Walton for helpful discussions, proof reading, suggestions and bug finding.

Nils Ericson for longtime support and suggestions.