∇NABLA

String Synthesizer

Version 1.3

http://www.fullbucket.de/music

Table of Contents

Introduction

Nabla is a software synthesizer plug-in for Microsoft Windows (VST2/VST3) and Apple macOS (VST2/VST3/AU) simulating the KORG® *Delta DL-50 Strings Synthesizer* from 1979. It is written in native C++ code for high performance and low CPU consumption. The main features are:

- Paraphonic *Synthesizer* and *Strings* sections
- Up to 64 voices polyphony
- *Synthesizer* section:
  - Four band-limited frequency divider-driven signal generators
  - Additional noise generator
  - 4-pole zero-delay feedback bandpass/lowpass filter
- *Strings* section:
  - Two band-limited frequency divider-driven signal generators
  - 2-band equalizer
  - Ensemble effect
- Flexible pitch/filter modulation by LFO or noise
- Tweaks (not to be found in the original *Delta*):
  - "GOD Mode" to provide *true* polyphony
  - Panning for *Synthesizer*/*Strings* sections
  - Built-in phaser and delay effects
- Double precision audio processing
- All parameters can be controlled by MIDI controllers
- Plug-in supports Windows and macOS (32 bit and 64 bit)

Nabla is based on the new **iPlug2** framework maintained by **Oli Larkin and the iPlug2 team**. Big thanks, guys!!! Without your work it would not have been possible to create a resizable *Nabla* user interface.

To resize the plug-in you just grab the yellow triangle at the bottom right of the *Nabla* window and drag it. You can save the current window size using the menu entry "Save Window Size" in the *Program and File Menu*.

If you have trouble with the standard version of *Nabla* please grab the (sound-wise identical) "N" version of the plug-in which is based on the original **iPlug** framework.

Turning The *Δ* Upside Down

Once upon a time I promised myself not to create a software simulation of an existing piece of hardware that I do not own by myself. Well, as you can see, I broke my promise: I never owned a *Delta* nor is it very likely that I ever will. But due to the internet – or better: due to gentle folks uploading material there for free – I was able to get the PDF of the original *Delta* service handbook including its' schematics, and tons of videos featuring this neat little instrument. At a first glance I thought "Well, you can use your own *deputy Mark II* to simulate the *Delta*!", but then I found out that this not exactly true.

The *Delta* follows a very clever (some would say "cheap") technical design of a polyphonic frequency divider-driven signal generator that is able to simultaneously create square waves in four footages (16', 8', 4', 2'). These waves – called *Signal Generators* – are nothing else than *choirs* of an electronic organ and can be mixed to

produce other wave shapes. While the mixed generator signal is fed into *one single* Voltage Controlled Filter (VCF) and used for the *Delta's Synthesizer* section, a second, hard-wired mixer section combines the four signals to generate the rough equivalent of a 16' and an 8' sawtooth wave. These two "saws", running through a fixed *Ensemble* effect, are the basis for the *Strings* section.

The rest of the sound processing is *paraphonic*, meaning that there is *only one* VCF, ADSR envelope, and amplifier for the *Synthesizer*, and only one AR envelope and amp for the *Strings* section. Why? Well, just to save money – hardware is expensive.

Then what is so striking about the *Delta* besides the mystic "It's one of these good old analog thingies"? The *Delta* is very *usable*: Obviously, the Korg engineers, driven by the reckless forces of cost reduction, did a great job on selecting those controls that really make sense to lay one's hands on, and made them available for the user. "Limitation is the friend of Creativity", and within it's limits the *Delta* is damned efficient to use.

OK, as of today's standards, the *Delta* is indeed pretty limited, so I asked myself: "What would you almost always add to the *Delta* if you played it?"

> First – a *Phaser*: The *Delta* contains a *String Ensemble*, and Monsieur Jarre somehow established the phaser as the natural companion to it.
>
> Second – a *Delay*: Here I would like to give Klaus Schulze and Tangerine Dream the credit of making delays the standard effects of electronic music.
>
> Third – a "GOD Mode": Hardware is expensive, software is cheap, so why not turn the *Delta* into a true polyphonic instrument? Thanks to the innovative *Generative Object Duplication*© technology you can now relieve this fine instrument from the Curse of Paraphony!

All decisions were made except for one: I needed a name for the plug-in version of the *Delta*. From Physics we know that there is a symbol like the Greek letter Δ (Delta) rotated by 180° (turned upside down), i.e. ∇, and this symbol is called *Nabla*.

"You cannot name a plug-in *Nabla*!" I thought. Yes, I can.

The *Synthesizer* Section

This somehow is the heart of *Nabla*: A polyphonic oscillator followed by a filter and an amplifier. An ADSR envelope to control the filter cutoff and/or the amplitude, plus a LFO for vibrato or cutoff modulation. The classic design of a JUNO 6. Or not?

Signal Generators (SG)

The *Synthesizer* section is equipped with four *Signal Generators* providing square waves in four different octaves (16', 8', 4', 2') plus an additional *Noise* generator. The level of each signal is set by a dedicated fader. Note that the *Signal Generators* are realized by using only one Master oscillator and a "Top-Octave Synthesizer", followed by a cascade of frequency dividers. This means that the four square wave signals are totally correlated! For example the rising edge of the 16' square wave always falls together with the rising edge of the 8' square wave. The same is true for the 4' and 2' waves as well as for *keys* (i.e. *C* to *B*) of *different* octaves.

Until now the *Synthesizer* section is *fully polyphonic* – even more than a JUNO 6! But here comes the...

Voltage Controlled Filter (VCF)

Yes, the *filter*, not the *filters*: The whole polyphonic signal pathway now becomes paraphonic (unless you switch to "real" polyphony, but that's a different story; see section *GOD Mode*). The VCF can be run in two modes: *Lowpass* and *Bandpass*. It features a *Resonance* parameter (like in the *Delta* without self-oscillation), a control for positive or negative modulation by the ADSR envelope, and a *Key Follower* switch, tracking the highest key that is currently pressed. Oh, by the way, you can of course set the *Cutoff Frequency*, too.

Do I have to mention that I am using a *Zero-Delay Feedback* design for the filter?

Envelope Generator (EG) and Voltage Controlled Amplifier (VCA)

The EG has the standard controls for *Attack*, *Decay*, *Sustain*, and *Release*; it can be used to control the VCF and/or the VCA. For the latter, one can instead select a simple "gated" envelope, resulting in a organ-like amplitude contour.

The last control of the EG is the infamous *Trigger Mode* switch. Since there is only one VCF, VCA, and EG, what should happen if you press more than one key? You have two options: The EG is re-triggered for each additional key pressed (*Multiple Trigger*), or it is not re-triggered until all keys have been released and a new key is pressed (*Single Trigger*).

The *Strings* Section

In the 70s, it was not uncommon to put multiple sections, e.g. Strings, Brass, Synthesizer, and Organ, into one box sharing a common keyboard controller; examples are the ARP *Quadra*, the Moog *Opus*, but also the Siel *Trilogy*, or the Yamaha *SK* series. The *Delta* is a modest instrument that only features two section different sections, the *Synthesizer* and the *Strings*.

Sound Generation

Nabla's string sound is based on two sawtooth-like signals (16' and 8'). Note that both "saws" are derived from the same source as the *Signal Generators* of the *Synthesizer* section: A fixed, hard-wired mixer combines the four square waves (16', 8', 4', 2') "under the hood" to form the rough approximations of a 16' and an 8' sawtooth. Using the *Octave Balance* control, one can continuously fade between both "saws", and the result is run through an (again fixed & hard-wired) *Ensemble* effect.

Envelope Generator, Amplifier, Equalizer

The *Strings* section again is paraphonic, i.e. for all voices there is only one amplifier controlled by an even simpler *Attack/Release* envelope. This envelope also features a *Trigger Mode* selector which has a slightly different effect than for the EG of the *Synthesizer*: Re-triggering in *Multiple* mode causes the envelope to reset the output to Zero, and then re-starts it from the very beginning of the Attack phase.

The *Strings'* last processing stage is the *Equalizer* with a low- and a high-shelf band. Nothing special here except that it may have quite an impact on the sound.

Common Sections

This chapter describes the features and controls common to both the *Synthesizer* and the *Strings* sections.

From Joystick To Wheels

The original *Delta* features a *joystick* for modulation purposes: The horizontal axis directly controls pitch and/or VCF cutoff frequency while the vertical axis controls the amount of modulation of the LFO (when pushing the joystick "up") or the *Noise* (when pulling the joystick "down").

However, joysticks are pretty out of fashion theses days (which is a pity) and modern controllers almost all have the classic *Pitch* and *Modulation wheels* instead. Thus, for *Nabla* I decided to map the horizontal joystick axis to the Pitch wheel and the vertical axis to the Modulation wheel. Unfortunately, you now have to select what modulation source (LFO or *Noise*) is controlled by the Modulation wheel; this is what the *Wheel* switch is for.

LFO and *Noise*

Apart from the Pitch wheel there are two (!) other modulation sources available: The global LFO and the *Noise*. The LFO features a triangular wave ranging from 0.001Hz to 25Hz. This signal can be used directly to apply a *Vibrato* effect. Note that the *Vibrato* affects both the *Synthesizer* and the *Strings* section because both share the same generator bank!

Noise provides a *continuous* random signal, unlike the classic Sample & Hold which is a stepwise signal. There is no way to control it except for modulation depth.

Destinations

Pitch and VCF cutoff frequency are the only destinations of modulation. The amount of pitch or *Frequency Modulation* (*FM*) and VCF or *Cutoff Frequency Modulation* (*FcM*) can be set separately. It's also possible to switch *FM* and/or *FcM* on or off.

General Pitch and Mixer

The master tune is controlled by the *Tune* parameter, and the whole instrument can be "transposed" one octave up.

Finally, the volume of the *Synthesizer* and the *Strings* section as well as the overall *Volume* can be set individually.

Tweaks

Up to now *Nabla* recreates all the features known from the *Delta*. Apart from the trivial ability to store and recall the programmed sound patches, I added some new features which I think are quite useful. However, I call them *tweaks* because 1) they are somehow tweaking the original design of the *Delta* and 2) "tweak" sounds cooler than "enhancement" or "improvement".

Panorama Controls

The *Delta* features a general mono "Mix" output and two additional mono outputs for the *Synthesizer* and the *Strings* section. In the *Nabla* plug-in, these outputs are merged into a stereo output pair where both sections have their own *Panorama* control. I think this setup suits more the typical use case of today's music production, while the original output configuration is still available (just pan the two sections to opposite directions and handle the Left/Right outputs accordingly).

GOD Mode

A debatable tweak. I struggled long with myself whether I should include it or not. But what if one could play the *Delta* in full polyphonic glory? So I added a simple switch, applied my amazing *Generative Object Duplication*© technology (which is even capable of modifying the color of LEDs), and turned *Nabla* into a true polyphonic instrument. Huzzah!

Note that in *GOD Mode* the *Trigger Mode* control of the *Synthesizer* section has no effect.

Wheel Source

As explained in the section *From Joystick To Wheels* the modulation source controlled by the Modulation wheel has to be chosen by the *Wheel* switch. This is because there exists no proper mapping between the two segments of the vertical joystick axis and the single dimension of the Modulation wheel.

Phaser

I love phasers, and it looks as if others love them too, at least in the context of string machines. The phaser of *Nabla* is a straight forward four stage zero-delay allpass filter with *Speed*, *Feedback* and *Mix* controls. Of course you can deactivate it.

Delay

I love delays, and it looks as if others love them too, at least in the context of electronic music. The delay of *Nabla* simulates a classic *Bucket Brigade Delay* (BBD) with delay times from 62ms to 500ms.

Technical note: The delay consists of a fixed number of sample memory cells (the *buckets*, not to be confused with the *Full Buckets*) like a pipeline that the signal has to pass before it reoccurs at the output. The "speed" of transferring the signal through these cells determines the delay time, and since there is a "maximum speed" and a fixed number of cells, the minimum delay time is not 0 but 62ms.

Program and File Menu

Not really a tweak but still something that the *Delta* did not have: A way of storing programs or *patches* and even giving them names! To select one of the 64 patches just click on the program number, and edit its' name by clicking in the text field.

When clicking on the *File* button, a context menu opens with the following options:

Copy Program	Copy current program to internal clipboard
Paste Program	Paste internal clipboard to current program
Load Program	Load a FXP program file containing a patch to *Nabla's* current program
Save Program	Save *Nabla's* current program to a FXP program file
Load Bank	Load a FXB bank file containing 64 patches into *Nabla*
Save Bank	Save *Nabla's* 64 patches to a FXB bank file
Init Program	Initialize the current program
Reload Configuration	Reload *Nabla's* configuration file (see section *The nabla.ini Configuration File*)
Save Configuration	Save *Nabla's* configuration file (see section *The nabla.ini Configuration File*)
Select Startup Bank	Select the bank file that should always be loaded when *Nabla* is started
Load Startup Bank	Load the Startup bank file; can also be used to check what the current Startup bank is
Unselect Startup Bank	Unselect the current Startup bank
Check Online for Update	When connected to the Internet, this function will check if a newer version of *Nabla* is available at fullbucket.de
Visit fullbucket.de	Open fullbucket.de in your standard browser

The nabla.ini Configuration File

Nabla is able to read some settings from a configuration file (`nabla.ini`). The exact location of this file depends on your operating system and will be displayed when you click on "Reload" or "Save Configuration". After you have edited this INI file in a text editor, you have to reload it using the *Reload Configuration* command from the *File* menu (see section *Program and File Menu*).

Force GOD Mode

This setting configures whether GOD mode is forced to be "always on", regardless of what the *GOD Mode* switch is set to. This might be useful if you are annoyed by *Nabla's* Curse of Paraphony and want to enjoy the full capabilities of today's digital wonders without tweaking the current patch. Just add or modify the following section to/in `nabla.ini`:

```
[Nabla]
ForceGODMode = true
```

If you want to disable *ForceGODMode*, just change `true` to `false`. If it is active then a virtual duct tape will be pasted over the *GOD Mode* switch.

MIDI Control Change Messages

All parameters of *Nabla* can be controlled by MIDI controllers, or more precise: Each MIDI controller (except *Modulation Wheel* and *Sustain Pedal*) can control one of *Nabla's* parameters. The mapping is defined in `nabla.ini` for example like this:

```
[MIDI Control]
# General Purpose controllers
CC16 = 2  # Synth Volume
CC17 = 4  # Strings Volume
CC18 = 44 # Delay Time
CC19 = 45 # Delay Feedback
# trying to follow GM2 definitions here ;-)
CC7  = 1  # Volume
CC8  = 3  # Synth Pan
CC10 = 5  # Strings Pan
...
```

The syntax is straight forward:

```
CC<controller number> = <parameter ID>
```

Given the above example, controller 7 directly controls the overall *Volume* parameter, controller 44 the *Delay Time* etc. As you can see, comments are introduced by the Pound sign (#); they are here just for description purposes and completely optional.

The *parameter ID* of one of *Nabla's* parameters is given in the section *Parameters* below. Note that the *controller number* can run from 0 to 119, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

Parameters

Synthesizer

parameter	ID	description
16'	15	volume of the 16' rectangle wave
8'	16	volume of the 8' rectangle wave
4'	17	volume of the 4' rectangle wave
2'	18	volume of the 2' rectangle wave
Noise	19	volume of the *Noise*
Attack	20	attack time of the *Synthesizer* EG
Decay	21	decay time of the *Synthesizer* EG
Sustain	22	sustain level of the *Synthesizer* EG
Release	23	release time of the *Synthesizer* EG
VCA EG Mode	24	selects whether the amplitude contour is controlled by the EG or by a simple gated envelope
Trigger Mode	25	selects if the EG will be re-triggered only after all keys have been released (*Single*) or for each key pressed (*Multiple*)
Cutoff Freq.	26	cutoff frequency
Resonance	27	resonance level
EG Depth	28	amount of cutoff frequency modulation by the EG
Filter Mode	29	filter mode: *Bandpass* or *Lowpass*
Key Follower	30	selects whether the cutoff frequency increases along with the highest key pressed or not

Strings

parameter	ID	description
Oct Balance	31	mix between 16' and 8' wave
Attack	32	attack time of the *Strings* envelope
Release	33	release time of the *Strings* envelope
Trigger Mode	34	selects whether the *Strings* envelope will be re-triggered only after all keys have been released (*Single*) or for each key pressed (*Multiple*)
Equalizer Low	35	gain of the low equalizer band
Equalizer High	36	gain of the high equalizer band

Joywheels

parameter	ID	description
FM SG	8	enable or disable pitch modulation
FM Depth SG	9	maximum amount of pitch modulation
fcM SG	10	enable or disable VCF modulation
fcM Depth SG	11	maximum amount of VCF modulation

Modulation Generator (MG) and Tune

parameter	ID	description
Vibrato Depth	13	amount of pitch modulation
Speed	14	rate of the MG (0 to 25Hz)
Tune	6	master tune
Octave	7	switches the overall pitch one octave up

Volume

parameter	ID	description
Volume	1	master volume
Synthesizer	2	volume of the *Synthesizer* section
Strings	4	volume of the *Strings* section

Pan, Mode, Wheel

parameter	ID	description
Synthesizer	3	panorama of the *Synthesizer* section
Strings	5	panorama of the *Strings* section
GOD Mode	0	switches the GOD mode (true polyphony) on or off
Wheel Source	12	selects the source for the pitch/VCF modulation (*MG* or *Noise*)

Phaser

parameter	ID	description
Mode	37	switches the phaser mode to *On 1*, *On 2* or *Off*
Speed	38	rate of the phaser frequency modulation (0 to 25Hz)
Feedback	39	amount of feedback
Mix	40	mix between dry and wet signal

Delay

parameter	ID	description
Mode	43	switches the delay mode to *On*, *Ping-Pong* or *Off*
Time	44	delay time (62 to 500ms)
Feedback	45	amount of feedback
Mix	46	mix between dry and wet signal

Hidden Parameters

parameter	ID	description
Phaser Frequency	41	center frequency of the phaser
Phaser Modulation	42	modulation amount of the phaser
Ensemble Delay 1	47	delay time of delay line 1 of the *Ensemble* effect
Ensemble Delay 2	48	delay time of delay line 2 of the *Ensemble* effect
Ensemble LFO 1 Speed	49	speed of the LFO 1 of the *Ensemble* effect
Ensemble LFO 2 Speed	50	speed of the LFO 2 of the *Ensemble* effect
Ensemble LFO 1 to Delay 1	51	modulation amount of LFO 1 to delay line 1 of the *Ensemble effect*
Ensemble LFO 1 to Delay 2	52	modulation amount of LFO 1 to delay line 2 of the *Ensemble effect*
Ensemble LFO 2 to Delay 1	53	modulation amount of LFO 2 to delay line 1 of the *Ensemble effect*
Ensemble LFO 2 to Delay 2	54	modulation amount of LFO 2 to delay line 2 of the *Ensemble effect*

Frequently Asked Questions

How do I install Nabla *(Windows VST2 32 bit version)?*

Just copy the files `nabla.dll` and `nabla.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *Nabla* VST2 plug-in the next time you start it.

How do I install Nabla *(Windows VST2 64 bit version)?*

Just copy the file `nabla64.dll` and `nabla.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *Nabla* VST2 plug-in the next time you start it.

Note: You may have to remove any existing (32 bit) `nabla.dll` from your VST2 plug-in folder or else your DAW may screw the versions up...

How do I install Nabla *(Windows VST3 64 bit version)?*

Just copy the files `nabla.vst3` from the ZIP archive you have downloaded to your system's or favorite DAW's VST3 plug-in folder. Your DAW should automatically register the *Nabla* VST3 plug-in the next time you start it.

How do I install Nabla *(Mac VST2/VST3/AU 64 bit)?*

Locate the downloaded PKG package file `nabla_1_3_0_mac.pkg` in Finder (!) and do a right- or control-click on it. In the context menu, click on "Open". You will be asked if you really want to install the package because it comes from an "unidentified developer" (me ☺). Click "OK" and follow the installation instructions.

What is the plug-in ID of Nabla*?*

The ID is **`d150`**.

How can I decrease Nabla*'s CPU load?*

Whenever it does not degrade the *sound* you need, try this:

- Switch *GOD Mode* off.
- Deactivate the Phaser or the Delay.
- If you don't need the *Synthesizer* or *Strings* section, set the respective volume slider to zero.

There is no sound when I play a key below C-0?

This is by design. Sorry.