DELIA: A NOISE-BASED VIRTUAL SYNTHESIZER

Cecilia Rossi
Laboratory of Music Informatics (LIM)
Department of Computer Science
University of Milan
cecilia.rossi4@studenti.unimi.it

Giorgio Presti, Federico Avanzini
Laboratory of Music Informatics (LIM)
Department of Computer Science
University of Milan
{name.surname}@unimi.it

ABSTRACT

This paper presents the prototype of a new virtual synthesizer characterized by the idea of generating pitched and non-pitched sounds starting from white noise only, and manipulating it in different ways, such as with subtractive and waveshaping techniques. The possible outcomes cover a palette of intentionally "low quality" sounds and textures, that are meant to be used in sound design scenarios, acusmatic compositions, or simply as original elements in pop songs. The synthesizer is implemented as a VST instrument plugin using the JUCE C++ framework.

1. INTRODUCTION

Luigi Russolo, a member of the Futurist movement of the XX century, in his manifesto "L'arte dei rumori" [1] wrote:

> *Noise is not only loud and unpleasant, you can count a number of noises that are delicate, and provide pleasant acoustic sensations to the ear.*

and further:

> *We need the musicians to get rid of easy and traditional rhythms, they must find in noise a way to grow, to renew, since every noise suggests a blend of the most diverse rhythms.*

It is clear that Russolo believes that noise, in all its facets, possesses its own musical identity and it is surely evident how noise can be either a component or the leading element of Futurist music, which you can interpret as music of the future. In his manifesto he expresses his will to create an orchestra composed of six "noise families" among which it is important to recall the following:

- rumbles, blasts;
- whistles and hisses;
- mumbles and gurglings;
- screechings, creakings, rustlings, buzzes, crackles, creases;

Among his experiments, whose objective was to create an actual instrument that could be part of the orchestra defined above [2], it is important to mention the "Scoppiatore" (the Burster), Russolo's first "Intonarumori" (noise intoner), which emulated a typical bursting noise, and it could change tone in a range of 2 octaves.

Related with Russolo's Intonarumori avant-garde idea, stands the Delia Noise Synth, whose goal is the same: to tune noise. In other words the aim of the project is to make noise "playable" using conventional music practices (i.e., using a keyboard and contemporary music harmony), as well as to produce the aforementioned families of noises.

The name "Delia" was chosen after Delia Derbyshire [3], one of the first electronic music female composers, who was born in Coventry in 1937 and was operational in the BBC Radiophonic Workshop between 1962 and 1973. [1] There she composed, besides the famous "Doctor Who" theme, other extremely creative and experimental works at that time such as "Inventions for Radio" (1964) with playwright Barry Bermange. However the most significant work of hers, which inspired the Delia Noise Synth, is the album "An Electric Storm" (in particular the song "Love Without Sound") realised in 1969 by the White Noise band, founded by Derbyshire, David Vorhaus and Brian Hodgson in 1968. This is a remarkable avant-garde research and accurate experimentation on noise as a full-fledged musical entity.

Among modern days artists, it is important to mention Trent Reznor, the frontman of the American band Nine Inch Nails, who famously said in an interview [4]:

> *I think there's something strangely musical about noise,*

thus reaffirming Russolo's view that noise has impressive musical potential so that it can be seen as a "manifestation of truth" [5, Benjamin Thigpen, p.3].

The purpose of Delia is to make noise more accessible for composers, sound designers or artists in general, in order to let them use it in their compositions, and be able to express all the nuances of their artistic inclinations. Starting from white noise, commonly known as an equal amplitude signal at all frequencies, the user is free to manipulate it: i) through different stages of filtering; ii) through resampling and requantization; iii) through distortion by adding odd harmonics or both even and odd harmonics; not to mention the ability to use noise as it is, in its "natural" shape. All of these possibilities will be discussed in the paper.

[1] https://www.bbc.com/historyofthebbc/100-voices/pioneering-women/women-of-the-workshop/delia-derbyshire, accessed: Feb 08, 2022

2. NOISE AND SYNTHESIZERS

The use of noise was central to the development of experimental music and avant-garde music throughout the 20th century [6], starting with the innovations brought by Edgar Varése in his compositional work and the already cited experiments by Luigi Russolo. In particular, noise has been used as a basic sound source in electronic music since the 1950's, beginning with the explorations carried out by the laboratories in Cologne, Paris, and Milan. Two main scenarios were considered: the manipulation of noises coming from real world recordings, and the electronic generation of white noise to be processed, typically with graphic equalizers [7].

Both scenarios made use of noise in compliance with what has been described in Sec. 1, but due to the technology available at the time the composers could not rely on one comprehensive tool, and instead used a combination of tools to achieve specific goals (i.e. noise sources, equalizers, tape recorders etc.).

When modern normalized synthesizers were introduced to the market, a noise source was almost invariably included, as a companion of one or more periodic oscillators: in most cases the oscillators were meant to produce the main tone, while the noise source was meant to add a stochastic component, aimed at producing a natural sounding output.

This configuration discouraged the use of noise as the main and only source of sound, but at the same time, artists (such as Delia, but also many others such as the krautrock scene or the emerging industrial scene) were looking for tools to introduce noise in pop music [5]. They had to rely on modular tools that, even if designed with this idea in mind, were based on manipulating pitched signals in order to create barely pitched noises (e.g. Buchla, [2] Makenoise, [3] or the "circuit bending" community [8, 9]).

Unlike the above, Delia aims at flipping this perspective: let the user start from noise to design sounds, as in the pioneering times. This approach follows a recent trend of re-discovering through digital technologies the use of noise in 20th century avant-garde music, such as in the digital reconstruction of the Intonarumori family by Serafin and De Götzen [10].

3. DELIA

3.1 Architecture

Delia Noise Synth is a Virtual Studio Technology (VST) [4] plugin developed in C++, using the JUCE [5] framework.

Delia Noise Synth is a polyphonic synthesizer where, as shown in Fig. 1, every voice reflects the traditional structure of a subtractive synthesizer. It is composed of two oscillators, a mixer, a main filter, and an amplification stage. The frequencies of the oscillators and the cutoff frequency of the main filter are controlled by three instances of a *Frequency Control* class, aimed at summing logarithmic frequencies values coming from the keyboard and possible modulation sources, according to corresponding "mod amount" parameters. The mixer, in addition to summing oscillators outputs, can also add to the output the ring modulation between the two oscillators, and additionally high-passes the output in order to remove the DC offset that may be introduced by the ring modulation or the oscillators themselves.

Control signal generators include two Low Frequency Oscillators (*LFOs*), linked to the frequencies of both oscillators, and to the cutoff frequency of the filter, respectively. They also include two envelope generators (*EGs*), linked to the frequencies of the oscillators and filter, and to the amplifier stage, respectively. After all the voices are summed, the signal is passed through a last saturation stage.

Within this general structure, the main distinguishing features of Delia are found in the implementation of the oscillators. The Oscillator class defines the internal structure of the two oscillators and is composed of 4 modules (see Fig. 2), described next.

3.2 Generator

This module deals with source selection and a first filtering stage. You can choose between White Noise or External. The latter refers to the synthesizer external input signal for the first oscillator, while it refers to the first oscillator signal for the second one. In this way the first oscillator can be used to process external signals, while the second one can be used to process the output of of the first one.

An Resistor-Capacitor (RC) filter, characterized by a very moderate slope of 6 dB per octave, is then applied, with a given cutoff frequency determined by the note played and a *tuned* parameter:

$$f_{cut} = 20000 \cdot (1 - tuned) + f \cdot (tuned) \;, \qquad (1)$$

where the *tuned* parameter could vary between 0 and 1 and consequently f_{cut} varies linearly between 20 kHz (for $tuned = 0$) to the frequency f coming from the Frequency Control (for $tuned = 1$).

3.3 Degradation

This module is responsible for resampling the signal from the previous module and re-quantizing it. Signal resampling can be performed in two different ways: by choosing a fixed time in milliseconds, or by using the frequency from the Frequency Control as resampling frequency. By using the latter mode, in conjunction with the previous filtering, a behaviour similar to a comb filter can be obtained, in tune with the note played on the keyboard, as can be heard in the audio samples described in Section 4. Both resampling modes start their clock on a note-on event.

Quantization lets the user choose the number of bits used to represent the signal, allowing to limit the number of possible represented values, introducing sonic possibilities discussed in Sec. 4.

[2] `https://buchla.com/`
[3] `https://www.makenoisemusic.com/`
[4] `https://developer.steinberg.help/display/VST/VST+Home`
[5] `https://juce.com`

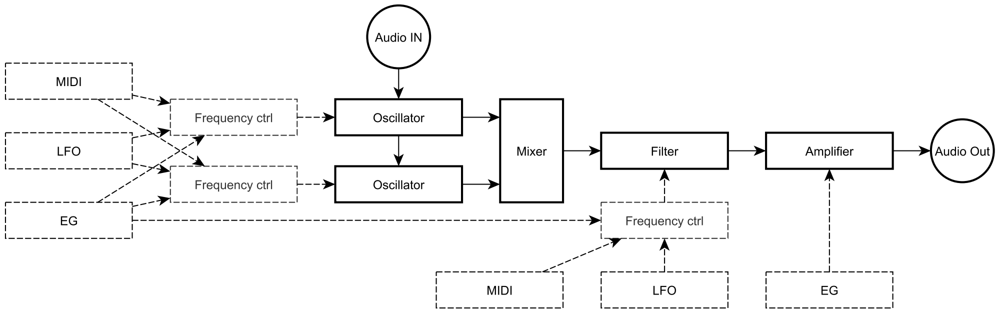

Figure 1. Schematic overview of a single voice of the Delia Noise Synth; Solid elements denotes audio processors and signals, while dashed elements refer to control signal modules and connections.

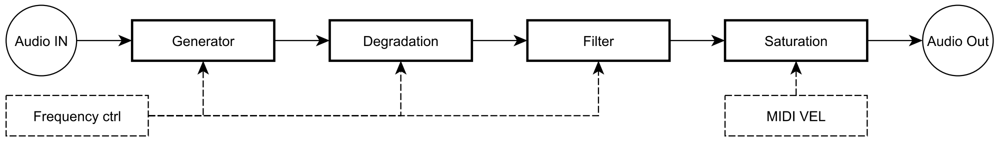

Figure 2. Schematic overview of a single oscillator section; Solid elements denote audio processors and signals, while dashed elements refer to control signal modules and connections.

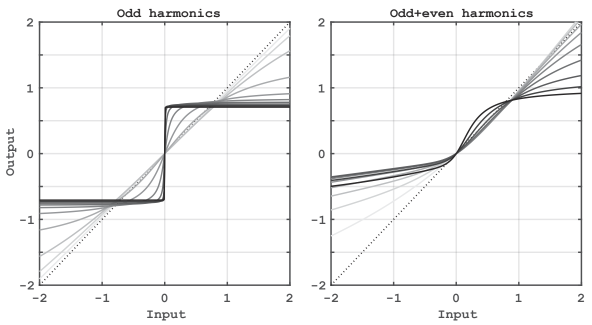

Figure 3. Static response of the saturators used. Different line shades denotes different saturation amount, i.e. the c parameter varying between 0 and 1.

3.4 Filter

This module applies a second filtering stage, which is far sharper than the first one. The filter used is a Topology Preserving Transform filter [11], which which can be considered as a generalization of the bilinear transform for digital filter design [12]. A slope of 24 dB per octave is used, and the user can choose the filtering type (lowpass, band pass, high pass, or none), as well as its resonance, whereas the frequency is controlled by the Frequency Control. This filter is responsible for the oscillator tuning in a conventional sense. The same class is also used in the main Filter section (see Fig. 1), but in that case the cutoff frequency and keyboard tracking are adjustable by the user.

3.5 Saturation

The saturation module applies a distortion to the signal, and it realizes a waveshaping component to create richer waveforms compared to the almost sinusoidal signal that may come out of the filter. Two different types of distortion are implemented.

Odd Harmonics Addition

The distortion function $f_{odd}(x,c)$ used to introduce odd harmonics (typical of square or triangular waves) consists of a simple arc tangent, and it is defined as:

$$f_{odd}(x,c) := \arctan(g_{in}(c) \cdot x) \cdot g_{out}(c) \,, \qquad (2)$$

where x is the input signal, c is the saturation parameter, varying between 0 and 1 along a pseudo-logarithmic scale, $g_{in}(c)$ and $g_{out}(c)$ are a distortion factor and a gain compensation factor, respectively.

The resulting distortion function is visible in Fig. 3 (left).

Odd and Even Harmonics Addition

A distortion function $f_{even}(x,c)$ adds odd harmonics as well as even harmonics in the signal (typical of sawtooth waves). This is performed through the soft++ function [13] $s_{++}(x,c)$, followed by an arc tangent function identical to the one used for odd harmonics. The soft++ function is a simplified version of the one described in [13] and is defined as:

$$s_{++}(x,c) = \ln(1+e^{x}) + (x/q(c)) - \ln(2) \,, \qquad (3)$$

where x is the input signal, c is the saturation parameter, $q(c) := 8c+2$ is the distortion amount, and $\ln(2)$ is an offset value which ensures that the distortion function passes through zero, thus avoiding an offset when no sound is produced.

The $f_{even}(x,c)$ distortion is defined as:

$$f_{even}(x,c) := f_{odd}(\ s_{++}(g(c)x,c)/g(c),\ \ 0.5c) \,, \qquad (4)$$

where $g(c) := 10c + \epsilon$ is a gain compensation factor used to balance the effect of the two saturation functions.

The resulting distortion function is visible in Fig. 3 (right).

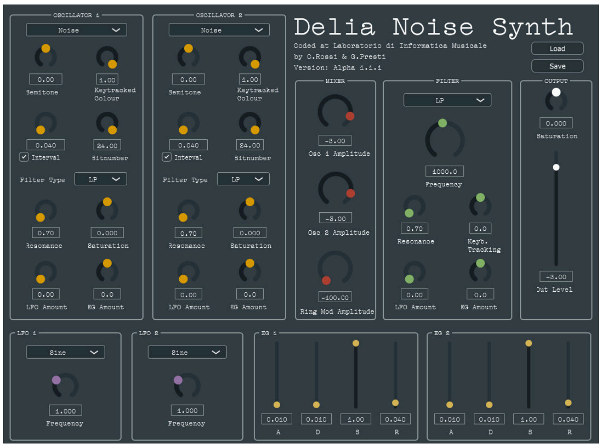

Figure 4. The graphical user interface of Delia Noise Synth.

4. DISCUSSION

The VST3 build of Delia Noise Synth can be downloaded from a dedicated webpage and may be tested with any VST host. [6] A picture of the graphical user interface, exposing all the parameters discussed in the previous section, is visible in Fig. 4.

Sound examples of the output obtainable with Delia are also available at the same dedicated webpage. Except for the final music sample, all the examples discussed in the remainder of this section were produced using the oscillators only (since the effects of the rest of the sound processing chain are consistent with standard synthesizer use practices), in order to demonstrate the sonic possibilities of the synthesizer.

4.1 One Oscillator

Here the effects of progressively including the oscillator components of Fig. 2 in an incremental fashion are discussed. The first examples are a demonstration of how Delia oscillators can behave like traditional oscillators (with the difference of being "imperfect", which is the point of the project). The subsequent examples instead demonstrate some of the peculiar features of Delia.

Generator

By filtering the source noise with a narrow band-pass filter, it is possible to generate an almost-sinusoidal sound. The randomness of the source noise provides to the resulting signal with noticeable imperfections and "movement". Further stages of processing (via second oscillator or the main filter) may result in a more precise signal.

Waveshaping

Adding the odd-harmonics waveshaper to the processing chain results in an almost-square wave, inheriting the same imperfections of the sinusoidal signal. Note that extreme saturation values will result in a naive-square wave, aliasing could have been avoided by oversampling the signal, but we left this choice to the user just in case aliasing is a desired feature.

Similar considerations apply when the $f_{even}(x, c)$ distortion function is used.

Quantization

As an effect of quantizing a noise signal, a rich non-pitched texture with an interesting low-end is produced.

Downsampling

As an effect of downsampling noise, with the sampling frequency being controlled by the played note, a peculiar intonation effect is obtained (the online example demonstrate this through a sequence of three notes).

Downsampling and Filtering

In this case the noise is downsampled to a very low rate (in the order of some tenth of a second) so to produce a rhythmical sequence of steps, a strategy used in Low Frequency Oscillators (LFO) usually referred as Sample & Hold (S&H). After the downsampling stage, a severe quantization reduces the possible step sizes, merging steps with similar amplitude values, resulting in an irregular pattern of steps. Finally, the oscillator filter introduces a very high resonance on the played note, turning each remaining step into a tuned impulse. In the audio sample a major chord is played, but due to the oscillator setup, it sounds like a random arpeggio.

4.2 Two Oscillators

Whereas the examples discussed above use a single oscillator, the following focus on the combination (in series or in parallel) of the two oscillators.

Series Connection With Quantization

Quantization noise introduced by the second oscillator can be used to enrich the base waveform generated by the first one, creating a variety of textures.

Series Connection With Downsampling

The aliasing introduced by downsampling the second oscillator can also be used as a way to enrich the signal produced by the first one, resulting in different pad sounds.

Parallel Connection

Using the two oscillators in a parallel fashion allows to exploit each of them for different musical purposes. As an example (see the related online sample in the dedicated webpage), the first one may be used to produce rhythmic patterns, while the second one produces continuous pad sounds.

Ring Modulation

The ring modulation capabilities of the mixer stage can be used creatively to further enhance the palette of sonic outputs. As an example (see the related online sample in the dedicated webpage), one oscillator can be used as a S&H LFO, which multiplies a continuous pad sound produced by the second oscillator.

[6] https://www.lim.di.unimi.it/demo/delia.php.

A Musical Example

As a final demonstration of Delia, a music sample was produced by exploiting the aforementioned techniques together with other Delia features (i.e., the EGs and the main filter, see Fig. 1). The output from Delia was only processed through a reverb effect only, and no additional equalization nor compression were applied.

5. CONCLUSIONS

A new virtual synthesizer called Delia Noise Synth was presented in this paper, and its capabilities were exemplified through a number of examples. The main distinguishing features of Delia are found in the definition of its oscillators, aimed at letting the user start from noise to produce a variety of sounds, in contrast with traditional oscillators which generate quasi-periodic, spectrally-rich waveforms as raw material for subtractive synthesis. It is our hope that the change of perspective implied by these features can help the creative process followed by the user to reach a desired effect.

6. REFERENCES

[1] L. Russolo, *L'arte dei rumori*. Edizioni futuriste di "poesia", Corso Venezia, 61 - Milano, 1916.

[2] B. Brown, "The noise instruments of Luigi Russolo," *Perspectives of New Music*, vol. 20, no. 1-2, pp. 31–48, 1981.

[3] T. Winter, "Delia Derbyshire: Sound and Music for the BBC Radiophonic Workshop, 1962-1973," Ph.D. dissertation, University of York, 2015.

[4] G. Rule, "Trent Reznor," *Keyboard Magazine*, March 1994, accessed: Feb 08, 2022. [Online]. Available: https://www.theninhotline.com/archives/articles/manager/display_article.php?id=548

[5] A. Cassidy and A. Einbond, Eds., *Noise in and as Music*. University of Huddersfield Press, 2013.

[6] P. Hegarty, *Noise/Music: A History*. New York: Continuum, 2007.

[7] M. M. Novati and J. Dack, *The Studio Di Fonologia: a musical journey 1954-1983, update 2008-2012*. Ricordi, 2012.

[8] R. Ghazala, *Circuit-Bending: Build your own alien instruments*. John Wiley and Sons, 2005, vol. 15.

[9] G. Hertz and J. Parikka, "Zombie media: Circuit bending media archaeology into an art method," *Leonardo*, vol. 45, no. 5, pp. 424–430, 2012.

[10] S. Serafin and A. De Götzen, "An enactive approach to the preservation of musical instruments – Reconstructing Russolo's Intonarumori," *J. of New Music Research*, vol. 38, no. 3, pp. 231–239, 2009.

[11] V. Zavalishin, "The art of va filter design," *Native Instruments, Berlin, Germany*, 2012.

[12] U. Zölzer, *Digital audio signal processing*, 2nd ed. John Wiley & Sons, 2008.

[13] A. Ciuparu, A. Nagy-Dăbâcan, and R. C. Mureşan, "Soft++, a multi-parametric non-saturating non-linearity that improves convergence in deep neural architectures," *Neurocomputing*, vol. 384, pp. 376–388, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219317163