WAVE DIGITAL FILTER IMPLEMENTATION OF LINEAR AND NONLINEAR AUDIO CIRCUITS ON EVENTIDE H9000

Di Maria Alberto, Maiolo Valerio

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20122 Milano, Italy
[alberto.dimaria, valerio.maiolo]@mail.polimi.it

ABSTRACT

The Eventide H9000 is a multi-effects processor hardware with an integrated visual software, VSig3, that enables the creation of custom algorithms. This solution combines the efficiency of a hardware with the properties of designing algorithms by the software. VSig3's language allows also the creation of low-level algorithms, although there are not any example in the literature. In this paper, we show how to model linear and nonlinear audio circuits with the Wave Digital Filters (WDFs) method, in the VSig3 platform. In particular, to deal with nonlinearities, the Canonical PieceWise-Linear (CPWL) representation is adopted. Both WDFs and CPWL exhibit highly desirable properties that perfectly match the possibilities offered by VSig3, even considering the language's limitations, which will be presented. We will show the steps of the first successful implementation ever done of a WDFs-based algorithm on H9000 and, at last, a diode clipper circuit as an example of application, in order to verify the validity of the proposed method.

Index Terms— Wave Digital Filters, VSig3, linear and nonlinear audio circuits, Eventide H9000

1. INTRODUCTION

Virtual Analog (VA) refers to a class of digital implementations, which are modeled from linear and nonlinear audio circuits [1]: most of digital audio plugins are implemented through this emulation process. In analog audio circuits, nonlinearities characterize the sound, modifying the harmonic content: diode chains, e.g., are employed in different audio equipment categories, such as compressors or saturators. VA modeling provides two different processes: the *black-box approaches* which, starting from pairs of input/output data, deduce the reference circuit. These approaches can be implemented through neural networks or Volterra series. The second one is the *white-box approaches*, that imitate the behaviour of the reference circuit through the simulation of ordinary differential equations (ODEs), e.g. employing the State-space method, the Port-Hamiltonian method, or Wave Digital Filters (WDFs). In most cases, this last class is more performing than black-box techniques, having as drawback an higher computational cost, which increases in presence of multiple nonlinearities [2]. WDFs are at the top of white-box methods: firstly developed in 70s by A. Fettweis, they are derived from a reference analog circuit [3], which is described through the port-wise method. The reference circuit is so modeled as fundamental input-output blocks, connected through junctions, which are instead modeled through multi-input-multi-output. The free parameters of certain ports can be chosen to make these ports reflection-free [3], arriving at an explicit solution that requires much less computing power respect to other methods. This property makes WDFs a suitable real-time VA method. Due to the computational power of modern hardware, a VA algorithm based on WDFs can be implemented in real-time platforms, such as audio plug-ins to be integrated into Digital Audio Workstations (DAWs) or to be used standalone. Furthermore, the explicit formulation of the WDFs method suggests the possibility of implementing complex physical models with real-time visual programming languages. In this paper, we propose the first implementation ever done of the WDFs in the H9000, a hardware multi-effects processor by Eventide. This hardware is provided with VSig3, a cross-platform visual editor that enables the creation of custom algorithms for the H9000, based on fundamental input/output blocks, through which the WDFs algorithm can be designed. VSig is the main tool Eventide used to create algorithms for the H9000. Only available to owners of H9000 or H9000R, it enables the exploration of the famous Eventide algorithms. The combination of H9000 and VSig3 stands as the perfect compromise between the software and hardware domain: the gear has in fact an high computational power, which can not be provided by any plugins, while VSig3 simplifies the customization process of algorithms. Eventide is known to be the leading company in the production of digital programmable pedals in which these algorithms run. This is immediately clear by looking at the H9, a previous and cheaper version of the H9000. Furthermore, Eventide H9000 also allows to configure up to four independent foot pedals, footswitches as well as expression pedals, to be used during live sessions by guitarists i.e. for controlling their own sound as they would do with any standard pedals. This opens the way to the possibility to create pedals that go beyond the boundaries set by the most famous analog circuits. Even more, it offers the possibility to create very complex circuits in small boxes, without any concern about the space or the cost of the circuitry. Therefore, the possibility of recreating the virtual counterpart of analog circuits becomes extremely attractive on this kind of hardware. Even if this top tier piece of gear is provided with more than 1600 algorithms, none of them is implemented with the WDFs and there is no literature of any attempt, aside from this proposed method.

The manuscript is organized as follows. Section 2 gives an overview on VSig3, emphasizing its limitations. Section 3 provides a background knowledge on WDFs, in particular on the modeling of one-port nonlinear element. Section 4 shows how VSig3's language is suitable to implement WDFs, with a particular focus on the CPWL. Section 5 presents the example of application to show that is possible to implement WDFs into VSig3. Finally, Section 6 concludes this paper.

2. VSIG3 AND ITS LIMITATIONS

VSig3 is the visual editor software provided by Eventide to implement customizable algorithms, made to be run specially on the H9000 [4]. Each VSig3 algorithm is made of fundamental input/output blocks, such as standard mathematical operations, interface elements or basic filters, which can be connected together through the already given input/output ports. The audio source enters through two analog/digital converters (ADCs), which are always the first step of the signal flow, then it is processed sequentially by the various blocks. The end of the signal flow consists in two digital/analog converters (DACs), representing the audio output signal. Both the ADCs and the DACs are automatically provided by VSig3 at the time of the creation of a new file, and they can not be modified by the user.

Nevertheless, implementing a complex algorithm starting from a blank file could be challenging, due to several constraints of VSig3 regarding the limit number of operators and their customizable options. Firstly, there is not the possibility to create a new fundamental block, such as a new filter with a specific transfer functions or, in the interface section, a knob with an alternative design. Even if the set of fundamental block covers includes the most commonly used functions, VSig3 lacks the chance to define a custom block, implemented i.e. with an imported code written for a specific task. Consequently, the second limitation concerns the impossibility to create a new module [5], based on the combination of an arbitrary number of already existing blocks. Adding a new custom module to the already existing collection would be an useful feature in several contexts, e.g. the calculus of matrices: instead, a $N \times N$ matrix needs to be calculated element per element, using the fundamental mathematical operators of sum and multiplication. However, there is an useful and time saving function provided by VSig3: the possibility to select and copy a group of blocks from a file, then paste it in another file. This seems to be the only unofficial way to define a new module, even if it can not be effectively added to the documentation.

These constrains translate into a complexity to implement low level algorithms: basic math operators (*Controlmath module*) are quite exhaustive, unlike the set of functions to work with audio signals (*Math module*). The division, i.e., it is not implemented among the available blocks of the *Math module*. Despite VSig3 provides a brief explanation for each block, concerning its function and input/output ports, it is often not enough to perfectly understand how the block works. This leads to an experimental approach, made by several attempts to learn the functioning of the block, rather than knowing it from the documentation. In addiction, due to the necessity to physically own an H9000 hardware to have a VSig3 license, the community of Vsig3 users is not very active, and so the literature about the software is extremely limited. So, even if the H9000 is a powerful multiprocessor hardware, the integrated software side needs to be largely improved on the already-mentioned aspects to provide a better user experience and, above all, the possibility to implement more easily complex low level algorithms.

3. WDF WITH SINGLE ONE-PORT NONLINEAR ELEMENT

Among the wide variety of white-box methods for digital emulation of reference circuits [6], WDFs are well known for highly desirable properties, such as efficiency and stability [3]. Even more, they preserve many of the properties of the analog reference circuits such as passivity, losslessness, stability, minimal parameter sensitivity [7], [6]. WDFs theory has reached a high level of maturity in dealing with physical models containing linear and non-linear elements as well. In general, one-ports in the WD domain are described by the pair of variables a, *incident wave*, and b, *reflected wave*, defined as a linear combination of the port voltage v and the port current i as

$$a = v + Zi, \qquad b = v - Zi \tag{1}$$

where $Z \neq 0$ is the *port resistance*. They are linked together by the scattering relation: in each block, the reflected wave is in function of the incident wave. The inverse mapping of (1) is defined as

$$v = \frac{(a+b)}{2}, \qquad i = \frac{(a-b)}{2Z}. \tag{2}$$

The free parameters of certain ports can be chosen to make these ports reflection-free, avoiding implicit equations and *delay-free-loops* (DFLs). This procedure, named *adaption*, can be done if the reference circuit implemented in the WD domain has linear elements, or at most one non-linearity. To deal with multiple nonlinearities circuits, iterative WD methods are employed: the literature provides several examples [8], [9].

As regard the treatment of a single one-port nonlinear element, the characteristic of the nonlinear element in the Kirchhoff domain (v, i) is mapped into a $b - a$ curve in the WD domain [10]. Then, this nonlinear characteristic can play its role within the circuit by connecting the bipole to an adapted port of the circuit [7], as long as the nonlinear element cannot be adapted. The WDF structure is efficiently implemented in a systematic fashion [6], scanning the binary connection tree of the topological representation of the reference model. Canonical PieceWise-Linear (CPWL) representation of curves [11] in the WD domain is widely presented in [12], as an explicit and global method for dealing with nonlinearities in the WDF framework. Among the various methods for dealing with nonlinearities, PWL certainly becomes very convenient and even necessary as concerns this particular implementation context. In the next section, we will discuss the implementation details of WDFs in VSig3, retracing the reasons that lead to embrace the workflow in question.

4. WDFS IMPLEMENTATION IN VSIG3

The problem of digitally implementing complex analog circuits brings to light the need for an efficient and stable method. Born in an historical context of low computational power, WDF's method allows to avoid iterative calculations and implicit solutions in favor of an explicit solution for the ordinary differential equations describing the circuit. Even more, the possibility of expressing nonlinearities through an explicit and global method such as PWL opens the way for the implementation of real-time circuits with a high number of nonlinear elements. All of these desirable features encourage the use of WDFs for Virtual Analog implementation and this choice becomes extremely relevant on an hardware of this type. It is worth to highlight they are highly compatible with the possibilities VSig3 currently offers: it is sufficient to focus on the mathematical operators involved.

They can be derived from the scattering relation

$$\mathbf{b} = \mathbf{S}\mathbf{a} \tag{3}$$

which, starting from the scattering matrix $\mathbf{S}$, allows to calculate the relationship between incident a and reflected b waves. The scattering matrix encodes all the information inferred from the reference

circuit components and its topology. As regards nonlinear circuits, the signal flow is deduced by the forward and backward scanning of the binary connection tree. As we can see, the main operators are those involved in a simple linear combination. Specifically, VSig3's language offers many low level control-rate and signal-rate modules and, among them, there are modules dedicated to signal-rate sums and products as well, i.e. the *Math* library. Even if less obvious, there is the possibility to handle single-sample feedback loops. In general, the latter enables the creation of digital filters, and it is fundamental for the implementation of dynamic elements with memory, such as capacitors and inductors. The minimum sensitivity to parameters, the use of few coefficients, the limited number of operations and their simplicity make WDFs a suitable method for VA in VSig3.

These considerations apply to linear circuits, but they are still valid when nonlinearities are treated with the CPWL approach. As exhaustively defined in [11], [12], a scalar nonlinearity can be represented in the wave digital domain as

$$b = \mu_0 + \mu_1 a + \sum_{j=1}^{N} (\eta_j |a - a_j| + \nu_j sgn(a - a_j)) \qquad (4)$$

which segments the curve with $N + 1$ segments with vertices coordinates (a_j, b_j) and

$$\mu_1 = 0.5\ (m_0 + m_N)$$
$$\eta_j = 0.5\ (m_j - m_{j-1}), \qquad j = 1, ..., N$$
$$\nu_j = \begin{cases} 0, & \text{if } f_b(a) \text{ is continuous at } a = a_j \\ 0.5\ (h(a_j^+) - h(a_j^-)), & \text{otherwise} \end{cases}$$
$$\mu_0 = h(0) - \textstyle\sum_{j=1}^{N} (\eta_j |a_j| + \nu_j sgn(a_j))$$

where m_j defines the slope of the j-th segment. As we can see at a glance sums, multiplications, absolute values and sign functions are involved. In many cases, the function is continuous and the nonlinearity is represented by means of sums, multiplications and absolute values. Therefore, the CPWL configures as an explicit and global method, in which nonlinearity is represented through the most basic mathematical operators and without resorting to division. Nonlinear elements are suitable to be easily implemented in VSig3, enriching definitely the range of analog circuits at our disposal in H9000.

5. CASE STUDY: RESULTS

In this Section, we analyze the behavior of a diode clipper circuit, shown in Fig. 1, implemented with the WDFs method firstly in MATLAB and then in VSig3. The two anti-parallel diodes are both assumed to have a Shockley model, characterized by the current function $i(v) = I_s\ (1 - e^{v/(\eta V_t)})$, where I_s represents the saturation current, η the ideality factor and V_t the thermal voltage. For this particular case, $I_s = 2.52$ nA, $V_t = 25.8$ mV and $\eta = 1.752$.

Fig. 2 shows the Wave Digital (WD) structure of the analog reference circuit. The nonlinearity, consisting of the two anti-parallel diodes, is modelled as one-port element using the CPWL method, as shown in detail in the Section 4.

In particular, 20 v-coordinates of vertices, manually chosen, are employed to have a good approximation of the $v-i$ characteristic of the two anti-parallel diodes. This procedure, with a so-small number of vertices, inevitably leads to have a loss of accuracy. Taking in consideration a greater number of vertices will slightly increases the accuracy but, on the other side, increases considerably the complexity of the implementation of the algorithm in VSig3. The obtained results, shown in the following Subsections, demonstrate that the considered number of vertices is an acceptable trade-off.

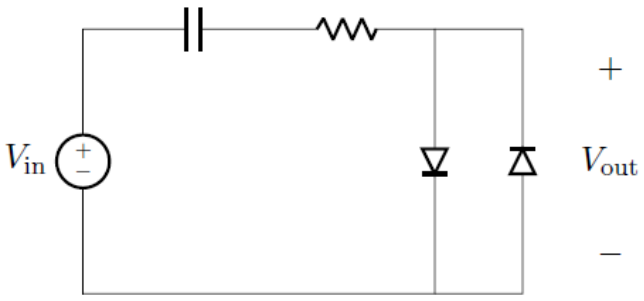

Figure 1: Diode Clipper Circuit. Parameters: $C = 4.7\times\ 10^{-6}$ F, $R = 1000\ \Omega$, $\mathrm{R}_{in} = 10^{-6}\ \Omega$.

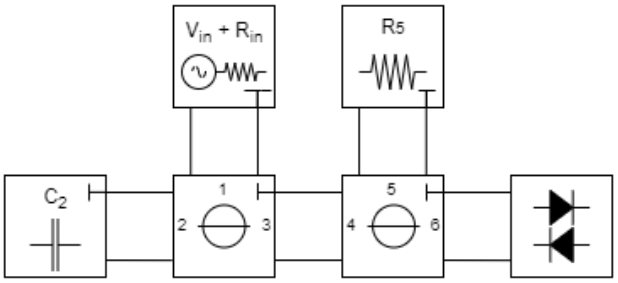

Figure 2: Corresponding WD structure of the Diode Clipper Circuit.

5.1. Comparison between LTSpice and WDFs

The comparison between the output signals of the LTSpice reference signal and WDFs implementation in MATLAB is shown in Fig. 3. We use a LTSpice model of the Shockley diode (1N914 component), whose characteristics match the parameters of the analog reference. The input audio signal consists in a sinusoid with 1 V of amplitude and $f = 500$ Hz, while the LTSpice file for the ground-truth is the output of the circuit of Fig. 1. The LTSpice sampling frequency is set equal to 96 kHz, which is the maximum available working frequency for the H9000.

As we can see from Fig. 3, the results of LTSpice and the WDF implementation match almost perfectly, even though there is a small error of 1% on average. This error, calculated as a difference between the two V_{out} signals, may include also some external factors to the WDFs, such as numerical inaccuracies introduced by LTSpice in the process of re-sampling or by its iterative solvers [13].

5.2. Comparison between VSig3 and WDFs

Here we compare the WDFs output signal and the output of the diode clipper implemented in VSig3. The results are shown in the Fig. 4. The input signal has the same parameters as the one in Section 5.1, as the sampling frequency equal to 96 kHz. The clipping signal of VSig3 follows the reference circuit in most of the points, confirming that the WDFs can be effectively implemented

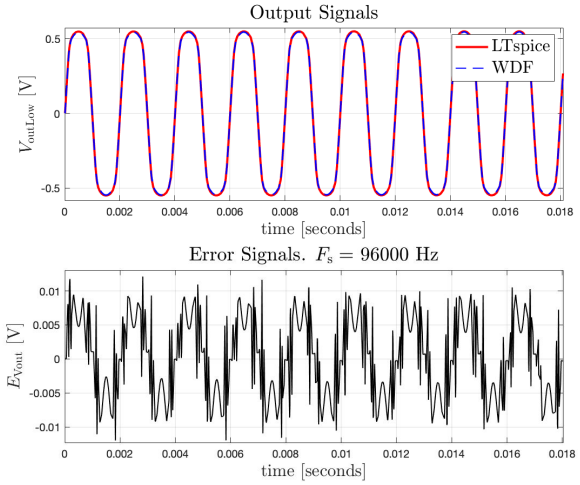

Figure 3: *Top*: V_{out} signal WDF model vs LTSpice. *Bottom*: Error signal WDF model vs LTSpice.

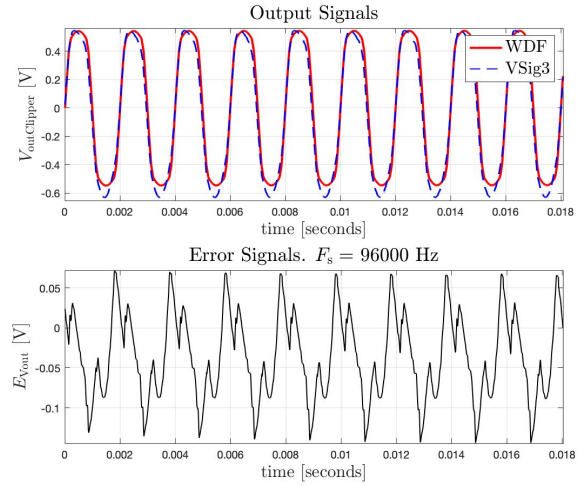

Figure 4: *Top*: V_{out} signal WDF model vs VSig3. *Bottom*: Error signal WDF model vs VSig3.

in the H9000, even though the error is larger than the comparison between LTSpice and WDFs, done in Section 5.1. The bigger entity of the error can be attributed to several factors. Firstly, the huge number of blocks and the complexity of their interconnections in the VSig3 file, make the debugging process difficult to implement. This leads to the possibility of having implementative errors. It is worth recalling that the CPWL has a summation with a large number of elements equal to the number of vertices, emphasizing again the limitation of the visual software regarding low level algorithms. Another external source of error can be related to the process of acquisition of the H9000 output, which is clearly different from a WDFs implementation in a MATLAB environment.

6. CONCLUSIONS AND FUTURE WORKS

In this paper we showed that is possible to successfully implement the WDFs into the multi-effects processor Eventide H9000, through its integrated software VSig3, which allows the possibility to design custom algorithms. We demonstrated the reasons why the H9000 is a particularly suitable choice to run WDFs algorithms, due to the set of mathematical operators involved. We also showed the steps of the WDFs method for a real-time implementation into the H9000, considering both linear and nonlinear circuits. In the last case, nonlinearities are dealt with the Canonical PWL, which is proved to be efficient, even if not easy to be implemented in VSig3, due to the limitations of the software.

This work opens the possibility, in the near future, of focusing on the development of a wide class of audio circuits to be run in the H9000, exploiting the WDFs properties.

7. REFERENCES

[1] G. De Sanctis and A. Sarti, "Virtual analog modeling in the wave-digital domain," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 715–727, 2010.

[2] R. Giampiccolo, M. G. d. Bari, A. Bernardini, and A. Sarti, "Wave digital modeling and implementation of nonlinear audio circuits with nullors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3267–3279, 2021.

[3] A. Fettweis, "Wave digital filters: Theory and practice," *Proceedings of the IEEE*, vol. 74, no. 2, pp. 270–327, 1986.

[4] https://www.eventideaudio.com/software/vsig3/.

[5] https://www.eventideaudio.com/forums/topic/vsig-custom-modules/.

[6] A. Sarti and G. De Sanctis, "Systematic methods for the implementation of nonlinear wave-digital structures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 2, pp. 460–472, 2008.

[7] A. Sarti and G. De Poli, "Toward nonlinear wave digital filters," *IEEE Transactions on Signal Processing*, vol. 47, no. 6, pp. 1654–1668, 1999.

[8] M. J. Olsen, K. J. Werner, and J. O. Smith III, "Resolving grouped nonlinearities in wave digital filters using iterative techniques," in *Proc. 19th Int. Conf. Digital Audio Effects, Brno, Czech Republic*, 2016, pp. 279–286.

[9] A. Bernardini and A. Sarti, "Dynamic adaptation of instantaneous nonlinear bipoles in wave digital networks," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1038–1042.

[10] K. Meerkotter and R. Scholz, "Digital simulation of nonlinear circuits by wave digital filter principles," in *1989 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1989, pp. 720–723.

[11] L. O. Chua and S. M. Kang, "Section-wise piecewise-linear functions: Canonical representation, properties, and applications," *Proceedings of the IEEE*, vol. 65, no. 6, pp. 915–929, 1977.

[12] A. Bernardini and A. Sarti, "Canonical piecewise-linear representation of curves in the wave digital domain," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1125–1129.

[13] M. Verasani, A. Bernardini, and A. Sarti, "Modeling sallen-key audio filters in the wave digital domain," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 431–435.