# A COMPARATIVE STUDY OF KOLMOGOROV-ARNOLD NETWORKS AND MULTI-LAYER PERCEPTRONS FOR VIRTUAL ANALOG MODELING IN WAVE DIGITAL FILTERS

*Riccardo Giampiccolo* [1], *Enrico Torres* [1], *Mauro G. De Bari* [2], *Samuel Limier* [2], *and Alberto Bernardini* [1]

[1] Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milan, Italy
[2] ARTURIA, Montbonnot-Saint-Martin, France
riccardo.giampiccolo@polimi.it

## ABSTRACT

The design of Virtual Analog (VA) algorithms has traditionally been divided between white-box (physics-based) and black-box (data-driven) approaches. Recent work has shown that hybrid methods, combining physical modeling with neural networks, can effectively leverage the strengths of both paradigms. In particular, Wave Digital Filters (WDFs) can be coupled with Multi-Layer Perceptrons (MLPs) to model circuits with multiple nonlinearities in a fully explicit manner. In this paper, we present a comparative study investigating the use of Kolmogorov-Arnold Networks (KANs) for VA modeling within the WDF framework. Unlike MLPs, KANs shift the learning paradigm by parameterizing activation functions instead of relying exclusively on learned weight matrices, potentially enabling more compact representations. Results show that, for our case study, KANs achieve accuracy comparable to MLPs while requiring approximately 70% fewer parameters at the cost of increased computational complexity. These findings suggest that KANs may represent a promising alternative in scenarios where memory footprint is a primary constraint, such as embedded audio applications, or when target models feature numerous nonlinear elements.

## 1. INTRODUCTION

Virtual Analog (VA) modeling aims to design digital signal processing algorithms able to reproduce the behavior of analog audio circuits while preserving the nonlinear characteristics that contribute to their distinctive sound [1]. Traditionally, VA techniques have been divided into *black-box* and *white-box* approaches. Black-box methods rely on system identification or data-driven techniques (e.g., Volterra series [2] and neural networks [3, 4]) to approximate the global input–output behavior of the target device, whereas white-box methods concern the solution of systems of ordinary differential equations (ODEs) describing the circuit topology and governing equations.

Among white-box techniques, Wave Digital Filters (WDFs) have proven to be a powerful framework for obtaining efficient circuit emulations [5, 6, 7, 8, 9, 10]. Early WDF formulations allowed fully explicit realizations for circuits containing only linear elements or a single nonlinear one-port element (described by means of explicit methods) placed at the root of a connection tree [11]. The introduction of vector waves [12] further extended this class to circuits featuring a single nonlinear multi-port element [9, 13], and more recently, to circuits featuring multiple (even multi-port) nonlinearities [10], in which neural networks are used to approximate the nonlinear mappings directly in the Wave Digital (WD) domain. In this context, neural networks have primarily been employed as parametric regressors of said nonlinear WD mappings. In particular, Multi-Layer Perceptrons (MLPs) [14] have been shown to provide accurate and computationally efficient approximations of static nonlinear scattering relations. The theoretical justification for using MLPs as universal nonlinear approximators stems from the *universal approximation theorem* [14], which guarantees that sufficiently wide networks can approximate any continuous function on a compact domain.

However, the universal approximation theorem is not the only theorem that allows us to obtain equivalent realizations of functions. Alternative representation results suggest that different neural parameterizations may be equally suitable. In particular, the *Kolmogorov–Arnold representation theorem* [15, 16] states that any continuous multivariate function on a compact domain can be expressed as a finite linear combination of continuous univariate functions. Inspired by this result, Kolmogorov–Arnold Networks (KANs) [17] have recently been proposed as an alternative to MLPs, replacing fixed activation functions and learnable linear weights with learnable univariate functions placed on the network edges. In contrast to MLPs, which learn linear combinations followed by fixed nonlinearities, KANs parameterize the nonlinear transformations themselves, leading to more compact representations.

To the best of our knowledge, no prior work has investigated the use of KANs in the context of Virtual Analog (VA) modeling. Given that for the scenario described so far, we are required to learn explicit multivariate nonlinear mappings in the WD domain, a natural question arises: are MLPs the most suitable architecture for this task, or can alternative representations offer structural or efficiency-related advantages?

In this work, we test the use of Kolmogorov-Arnold Networks (KANs) for VA modeling within the WDF framework, with the aim of assessing whether their alternative functional parameterization yields specific advantages. In particular, we compare KANs and MLPs in modeling the nonlinear multi-port scattering relation arising in the low-pass filter stage of the Steiner-Parker topology [18], as implemented in the Arturia MiniBrute synthesizer [19]. Through a controlled case study, we analyze approximation accuracy, parameter efficiency, and computational cost. Our results show that KANs achieve accuracy comparable to MLPs while requiring approximately 70% fewer parameters, at the expense of increased computational complexity. These findings suggest that,

although KANs do not systematically outperform MLPs, they represent a viable alternative whose advantages may become particularly relevant in memory-constrained scenarios or when compact parameterizations are required.

The remainder of this article is organized as follows. Section 2 provides background on Wave Digital Filters. Section 3 discusses the modeling of nonlinearities within WDFs using neural networks, as well as the solution of WD structures. The Kolmogorov-Arnold representation theorem and Kolmogorov-Arnold Networks are presented in Section 4. Section 5 provides the comparative analysis between KANs and MLPs, taking the low-pass filter section of the Arturia MiniBrute Voltage-Controlled Filter as a case study. Finally, conclusions are drawn in Section 6.

## 2. WAVE DIGITAL FILTERS

In this section, we first present the traditional definition of wave variables, and then address vector waves, highlighting their relevance for modeling circuits containing multiple nonlinearities in a fully explicit fashion.

### 2.1. Scalar Waves

Given a port-wise description of a reference analog circuit, WDFs require, for each port $j$, a transformation of Kirchhoff variables into wave variables. Over the years, several scalar definitions have been proposed in the literature [5, 20, 21]; however, in this work, we adopt the most widespread formulation, namely the voltage wave definition, which reads

$$a_j = v_j + Z_j i_j\,, \qquad b_j = v_j - Z_j i_j\,, \tag{1}$$

where $v_j$ and $i_j$ denote the $j$th port voltage and current, while $a_j$ and $b_j$ represent the corresponding incident and reflected waves. The scalar parameter $Z_j$, typically referred to as *reference one-port resistance*, plays a fundamental role in the solution of WD structures [5].

In particular, for linear one-port elements, $Z_j$ can be chosen so as to eliminate the instantaneous dependence of $b_j$ on $a_j$. As an example, the scattering relation of a generic linear one-port element (including dynamic components) in the WD domain can be expressed as [22]

$$b_j[k] = \frac{R_{\mathrm{e},j}[k] - Z_j[k]}{R_{\mathrm{e},j}[k] + Z_j[k]} a_j[k] + \frac{2Z_j[k]}{R_{\mathrm{e},j}[k] + Z_j[k]} V_{\mathrm{e},j}[k]\,, \tag{2}$$

which follows from applying (1) to the discrete-time Thévenin equivalent equation

$$v_j[k] = R_{\mathrm{e},j}[k] i_j + V_{\mathrm{e},j}[k]\,, \tag{3}$$

where $k$ denotes the sample index and $R_{\mathrm{e},j}[k]$, $V_{\mathrm{e},j}[k]$ are the equivalent resistance and voltage terms, respectively. By selecting $Z_j[k] = R_{\mathrm{e},j}[k]$, (2) reduces to $b_j[k] = V_{\mathrm{e},j}[k]$, thus removing the implicit dependence between $b_j$ and $a_j$. Within the WDF framework, this procedure is referred to as *adaptation* [5].

Topological connection networks are modeled as $J$-port WD blocks characterized by the vectors $\mathbf{b}_{\mathrm{J}} = [b_1, \ldots, b_J]^{\mathrm{T}}$ and $\mathbf{a}_{\mathrm{J}} = [a_1, \ldots, a_J]^{\mathrm{T}}$, collecting the waves incident to and reflected from the junction, respectively. The corresponding scattering relation can be written as

$$\mathbf{a}_{\mathrm{J}} = \mathbf{S}\mathbf{b}_{\mathrm{J}}\,, \tag{4}$$

where $\mathbf{S}$ denotes the so-called $N \times N$ scattering matrix. For reciprocal lossless connection networks, $\mathbf{S}$ can be computed as

$$\mathbf{S} = 2\mathbf{Q}^{\mathrm{T}}(\mathbf{Q}{\mathbf{Z}_{\mathrm{J}}}^{-1}\mathbf{Q}^{\mathrm{T}})^{-1}\mathbf{Q}{\mathbf{Z}_{\mathrm{J}}}^{-1} - \mathbf{I}\,, \tag{5}$$

$$\mathbf{S} = \mathbf{I} - 2\mathbf{Z}_{\mathrm{J}}\mathbf{B}^{\mathrm{T}}(\mathbf{B}\mathbf{Z}_{\mathrm{J}}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}\,, \tag{6}$$

where $\mathbf{I}$ is the $J \times J$ identity matrix, while $\mathbf{Q}$ and $\mathbf{B}$ denote the fundamental cut-set and loop matrices, respectively [23, 21]. The matrix $\mathbf{Z}_{\mathrm{J}}$ is diagonal and contains the reference one-port resistances $[Z_1, \ldots, Z_J]$. The reader is referred to [22, 21] for further details on element modeling and WD structure representation.

### 2.2. Modeling Diodes

Diodes are commonly described using the *Extended Shockley* model [24], which exploits series and shunt resistances of the pn-junction, $R_{\mathrm{s}}$ and $R_{\mathrm{p}}$, to mitigate the numerical issues that may arise when dealing with exponential nonlinearities. The resulting current-voltage relation is given by

$$f(v) = I_{\mathrm{s}}\Big(\exp\Big(\frac{v - R_{\mathrm{s}} i}{\eta V_{\mathrm{t}}}\Big) - 1\Big) + \frac{v - R_{\mathrm{s}} i}{R_{\mathrm{p}}}\,, \tag{7}$$

where, for simplicity, the time index $k$ is omitted. In (7), $I_{\mathrm{s}}$ denotes the saturation current, $\eta$ the ideality factor, and $V_{\mathrm{t}}$ the thermal voltage. When (7) is expressed in the WD domain, however, it cannot be adapted. For this reason, several approaches have been proposed in the literature to implement such a nonlinear relation within WDFs. These include iterative techniques such as Newton-Raphson solvers [24], closed-form formulations based on the Lambert $\mathcal{W}$ function [25] or the Wright $\omega$ function [26], as well as data-driven solutions relying on neural networks [7].

In this work, however, diodes are modeled using neural networks operating on vector waves [12].

### 2.3. Vector Waves

As discussed in [12, 9], the scalar definition (1) generally leads to Delay-Free Loops (DFLs) when modeling multi-port WD blocks in the WD domain. To overcome this limitation, vector wave variables have been introduced, allowing us to describe the ports of an $N$-port WD block in a unified fashion. The (voltage) vector wave definition reads

$$\begin{aligned} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} &= \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} + \begin{bmatrix} Z_{11} & \ldots & Z_{1N} \\ \vdots & \ddots & \vdots \\ Z_{N1} & \ldots & Z_{NN} \end{bmatrix} \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix}, \\ \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} &= \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} - \begin{bmatrix} Z_{11} & \ldots & Z_{1N} \\ \vdots & \ddots & \vdots \\ Z_{N1} & \ldots & Z_{NN} \end{bmatrix} \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix}, \end{aligned} \tag{8}$$

where $[v_1, \ldots, v_N]^{\mathrm{T}}$ and $[i_1, \ldots, i_N]^{\mathrm{T}}$ collect the port voltages and currents, while $[a_1, \ldots, a_N]^{\mathrm{T}}$ and $[b_1, \ldots, b_N]^{\mathrm{T}}$ denote the incident and reflected wave vectors. The matrix

$$\mathbf{Z}_{1:N} = \begin{bmatrix} Z_{11} & \ldots & Z_{1N} \\ \vdots & \ddots & \vdots \\ Z_{N1} & \ldots & Z_{NN} \end{bmatrix} \tag{9}$$

is a full-rank $N \times N$ matrix of real free parameters $Z_{ij}$, referred to as *reference multi-port resistance*.

Similar to the scalar case, linear $N$-port elements can be adapted by properly selecting $\mathbf{Z}_{1:N}$ so as to remove instantaneous dependencies between incident and reflected wave vectors [12]. When vector-defined $N$-port elements coexist with scalar one-port elements within the same WD structure, the scattering matrix (4) is still computed using (5) or (6), with $\mathbf{Z}_\mathrm{J}$ now being block-diagonal and containing both scalar and multi-port reference resistances [12].

## 3. NEURAL NETWORK-BASED MODELING OF MULTI-PORT WD BLOCKS

The explicit modeling of circuits with multiple nonlinearities by means of vector waves and neural networks has been thoroughly addressed in [10]. In this section, we briefly summarize the aspects of said framework that are directly adopted in this work.

When a circuit contains multiple one-port and/or multi-port nonlinear elements, they can be grouped into a single $N$-port nonlinear WD block, thereby isolating the nonlinear part of the circuit from the surrounding linear elements. By doing so, we can still resort to the case of a single nonlinear WD block, where, in contrast to the scalar case, we have now a *vector* of waves incident to and reflected from the corresponding junction *vector* port.

Let us assume, without any loss of generality, the $N$-port to be connected to ports 1 to $N$ of the WD topological junction. If the system of equations $\mathbf{S}_{1:N,1:N} = \mathbf{0}$ (where $\mathbf{S}_{1:N,1:N}$ denotes the submatrix of $\mathbf{S}$ associated with ports 1 to $N$) admits a solution for $\mathbf{Z}_{1:N}$, then the junction vector port connected to the $N$-port block can be properly adapted, resulting in a WD structure with no DFLs [10]. However, for the WDF to be fully explicit, the scattering equation of the $N$-port must itself be solved by means of an explicit method. In particular, the generic nonlinear scattering relation of an $N$-port can be expressed as

$$\mathbf{b} = \mathbf{f}(\mathbf{a}) \,, \tag{10}$$

and as shown in [9, 10], the modeling of $\mathbf{f}(\cdot)$ can be formulated as a regression problem directly in the WD domain. The required Kirchhoff-domain data can be measured on the circuit itself or generated through circuit simulation, and then mapped onto the WD domain using (8). It follows that mapping $\mathbf{f}(\cdot)$ can be approximated as

$$\hat{\mathbf{b}} = \tilde{\mathbf{f}}(\mathbf{a}; \boldsymbol{\theta}) \,, \tag{11}$$

where $\tilde{\mathbf{f}}$ is a suitable Neural Network (NN) architecture, whose trainable parameters $\boldsymbol{\theta}$ are optimized to estimate the vector of reflected waves $\hat{\mathbf{b}}$ given the vector of incident waves $\mathbf{a}$.

The resulting WD structure is organized then as a connection tree composed of [10]: (i) a *root*, namely the WD block containing the NN; (ii) a single *node*, corresponding to the topological junction; and (iii) multiple *leaves*, representing the linear elements. At each time sample $k$, the operating point is computed according to the following procedure. First, the waves reflected by the linear elements are evaluated. These waves are then propagated forward through the tree to compute the waves incident to the root via the junction scattering. Next, the vector wave reflected by the root (incident to the junction) is computed through the forward pass of the NN, and propagated back to the leaves through the junction scattering. It is important to note that, since all ports are adapted either element-side (for the linear elements) or junction-side (for the NN block), not only the solution of the WD block but the overall computation remains fully explicit, leading to interesting real-time performance.

In previous works [9, 13, 10], Multi-Layer Perceptrons (MLPs) were employed to approximate static nonlinear scattering relations. In the current work, we retain the same WD framework and training strategy, and focus instead on comparing MLPs with Kolmogorov-Arnold Networks (KANs) for modeling the nonlinear mapping $\mathbf{f}(\cdot)$.

## 4. KANS AS ALTERNATIVE UNIVERSAL FUNCTION APPROXIMATORS

In this section, we summarize the theory behind Kolmogorov-Arnold Networks to provide a solid footing to better appreciate the difference with common approaches.

Recently, it has been questioned whether Multi-Layer Perceptrons (MLPs) [14], i.e., fully-connected feedforward neural networks (see Fig. 1a), represent the most suitable nonlinear regressors for function approximation tasks [17]. Although MLPs enjoy remarkable theoretical properties, most notably those guaranteed by the universal approximation theorem [14], alternative architectures have been proposed that revisit the very way nonlinear mappings are parameterized.

Among these, Kolmogorov-Arnold Networks (KANs) [17], shown in Fig. 1b, have recently attracted attention as promising alternatives to MLPs. Empirical evidence suggests that KANs may offer advantages in terms of interpretability, parameter efficiency, and approximation accuracy, particularly in problems exhibiting compositional structure [17].

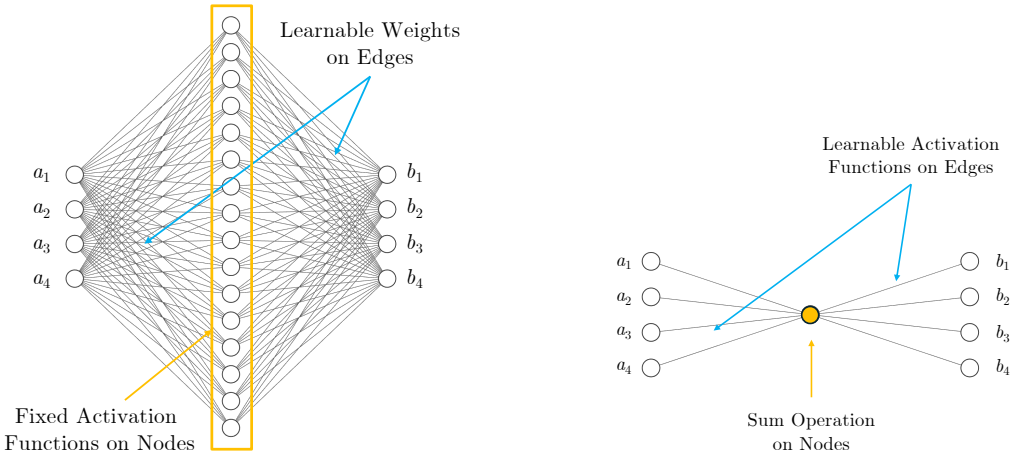
From the architecture standpoint, KANs differ from MLPs in a fundamental way. In MLPs, linear weight matrices are interleaved with fixed nonlinear activation functions applied at the nodes. In contrast, KANs place learnable univariate activation functions on the edges of the network, while nodes are only responsible for summing input signals. In practice, each edge is parameterized as a 1D function in the form of a spline. As a consequence, KANs do not rely on linear weight matrices followed by fixed nonlinearities but they do directly learn nonlinear transformations.

At first glance, replacing scalar weights with parameterized splines may appear computationally demanding. However, KANs often require significantly smaller network widths than MLPs for comparable accuracy, leading to more compact computational graphs in particular application scenarios. Conceptually, KANs can be interpreted as a hybrid between splines and neural networks [17]: splines provide accurate local approximations of univariate functions, while the network structure enables the modeling of high-dimensional compositional relationships, thus alleviating the curse of dimensionality typically associated with pure spline-based models [17].

### 4.1. Kolmogorov-Arnold Theorem

Whereas MLPs are commonly associated with the universal approximation theorem [14], KANs are inspired by the Kolmogorov-Arnold representation theorem [15], originally formulated in 1957. The theorem states that any continuous multivariate function defined on a bounded domain can be expressed as a finite superposition of continuous univariate functions and additions.

Let $g : [0,1]^m \to \mathbb{R}$ be a continuous function. Then, there exist continuous univariate functions $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q :$




**(a)** MLP. **(b)** KAN.

**Figure 1:** Comparison between MLP and KAN architectures. The MLP has shape $[4, 16, 4]$, whereas KAN $[4, 1, 4]$.

$\mathbb{R} \to \mathbb{R}$ such that

$$g(\mathbf{x}) = \sum_{q=1}^{2m+1} \Phi_q \left( \sum_{p=1}^{m} \phi_{q,p}(x_p) \right) . \tag{12}$$

The functions $\phi_{q,p}$ are referred to as *inner* functions and are independent of $g$, whereas $\Phi_q$ are the *outer* functions that do depend on the particular function $g$ under consideration. From a structural viewpoint, the theorem decomposes the modeling process into two stages: first, inner functions (the so-called *inner layer*) map the input vector onto an intermediate vector via univariate transformations, then the outer functions (outer layer) combine this intermediate representation into a scalar through additional univariate mappings.

Despite its elegance, the theorem did not immediately translate into practical machine learning architectures. One reason is that the original representation does not guarantee smoothness of the univariate functions, which may exhibit pathological behavior [17]. Furthermore, the classical formulation corresponds to a shallow structure with fixed width $2m + 1$, and does not directly extend to arbitrary depths and widths.

### 4.2. Kolmogorov-Arnold Networks

Recent work has revisited the Kolmogorov-Arnold theorem from a modern deep learning perspective, ultimately giving birth to the Kolmogorov-Arnold Networks [17]. KANs generalize the theory to deeper and wider architectures and parameterize the univariate functions with smooth basis expansions, such as B-splines [27, 17]. Under suitable smoothness assumptions, this formulation can yield favorable approximation properties and improved neural scaling behavior, particularly when the target function exhibits compositional structure.

Naming $L$ the total number of layers (i.e., the depth of the network), the $l$-th KAN layer is defined as the following matrix of 1D functions

$$\mathbf{\Phi}_l = \{\phi_{l,q,p}\} , \tag{13}$$

with

$$l = 1, \ldots, L , \quad p = 1, \cdots, m_l , \quad q = 1, \ldots, m_{l+1} , \tag{14}$$

where $m_l$ and $m_{l+1}$ are the input and output dimensions of the $l$-th layer, whereas $\phi_{l,q,p}$ are the activation functions that connect neurons $x_{l,q}$ to neurons $x_{l+1,p}$. Moreover, each neuron performs simply the sum of all incoming signals as

$$x_{l+1,q} = \sum_{p=1}^{m_l} \phi_{l,q,p}(x_{l,p}) . \tag{15}$$

According to this formalization, the inner functions of (12) can be thought of as a KAN layer with $m_l = m$ and $m_{l+1} = 2m + 1$, and the outer functions as a KAN layer with $m_l = 2m + 1$ and $m_{l+1} = 1$. Therefore, the original Kolmogorov-Arnold representation entails a 2-layer KAN with shape $[m, 2m + 1, 1]$ [17]. Finally, each edge function $\phi_{l,q,p}$ is parametrized as

$$\phi_{l,q,p}(x) = w_\mathrm{b}\beta(x) + w_\mathrm{s}\,\mathrm{spline}(x) , \tag{16}$$

where $\beta(x) = x/(1+e^{-x})$ is the basis function, whereas $\mathrm{spline}(x) = \sum_i c_i B_{i,o}(x)$ is a linear combination of B-splines of order $o$ characterized by $G + 1$ grid points, with $w_\mathrm{b}$, $w_\mathrm{s}$, and $c_i$ being real trainable parameters.

Being $W$ the width of all layers and $L$ the depth of the neural network, a KAN is characterized by approximately $O(W^2LG)$ parameters. By contrast, an MLP with same width $W$ and depth $L$ is characterized by roughly $O(W^2L)$ parameters. At first glance, this may suggest that MLPs require fewer parameters than KANs. However, empirical evidence [17] shows that KANs typically achieve comparable approximation accuracy with significantly smaller widths $W$ than MLPs, thus mitigating the apparent increase in parameter count.

An additional feature of KANs is that the accuracy of the representation can be improved even after training without modifying the network topology [17]. In particular, the width and depth of the network can be kept fixed while increasing the number of spline grid $G$, effectively refining the representation of the learned univariate functions. This process is performed by fitting a new fine-grained spline to the previously learned coarse-grained spline.

Finally, KANs offer a high degree of interpretability. Since the nonlinear transformations are explicitly represented by spline functions, the learned mappings can be directly visualized and analyzed [17]. This property allows us to visualize the learned edge


**Figure 2:** Arturia MiniBrute synth.

**Table 1:** Parameter values of diode 1N4148.

| Param. | Value | Param. | Value | Param. | Value |
|---|---|---|---|---|---|
| $I_s$ | 2.52 nA | $\eta$ | 1.752 | $V_t$ | 25.8 mV |
| $R_s$ | 1 µΩ | $R_p$ | 1 MΩ | – | – |

functions, highlighting the interpretable functional parameterization of KANs and potentially guiding the design of alternative implementations.

## 5. COMPARATIVE STUDY

In this section, the performance of KANs and MLPs is evaluated using a subcircuit extracted from the Voltage-Controlled Filter (VCF) of the Arturia MiniBrute synthesizer [19] (represented in Fig. 2). Released in 2012, the Arturia MiniBrute is a monophonic analog synthesizer that rapidly gained popularity among musicians and sound designers thanks to its distinctive raw and aggressive sound character. The synth features a fully analog signal path, spanning from the oscillator section, comprising sawtooth, pulse, triangle, and noise generators with waveshaping capabilities, up to the final output stage. The VCF is based on a modified multimode implementation of the well-known Steiner-Parker topology [18]. In particular, in this work, we take into account only the feedforward path as represented in Fig. 3, and we focus only on the low-pass filter section.

The two current sources $I_1$ and $I_2$, together with their parallel resistors $R_{10}$ and $R_{11}$, correspond to the Norton equivalent representation of the original BJT-based current generators, and take constant values $I_1 = 1.6182$ A and $I_2 = 1.2325$ A. Diodes $D_1$ and $D_4$ represent a series connection of three 1N4148 diodes, while $D_2$ and $D_3$ correspond to a series connection of two 1N4148 diodes. Therefore, the Extended Shockley model in (7) used to describe $D_1$ and $D_4$ employs an equivalent ideality factor $3\eta$ and equivalent resistances $3R_s$ and $3R_p$, whereas the model for $D_2$ and $D_3$ uses parameters $2\eta$, $2R_s$, and $2R_p$. Finally, Table 1 reports the 1N4148 diode parameters; the remaining circuit parameters are, instead, reported in Table 2.

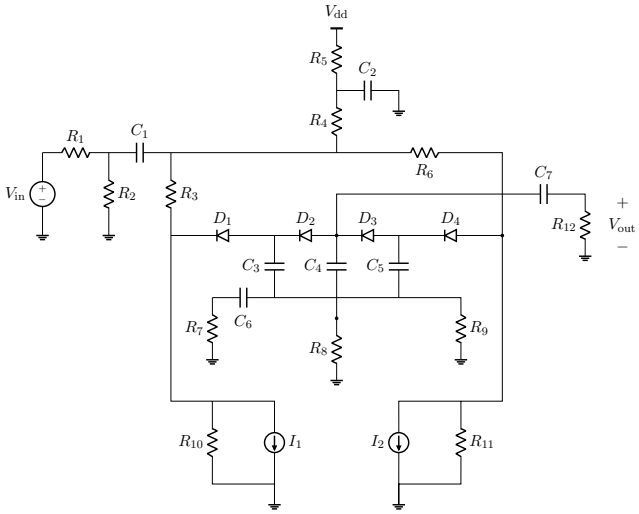



**Figure 3:** Circuit schematic of the Arturia MiniBrute VCF section, where only the feedforward path and low-pass filter section are represented.

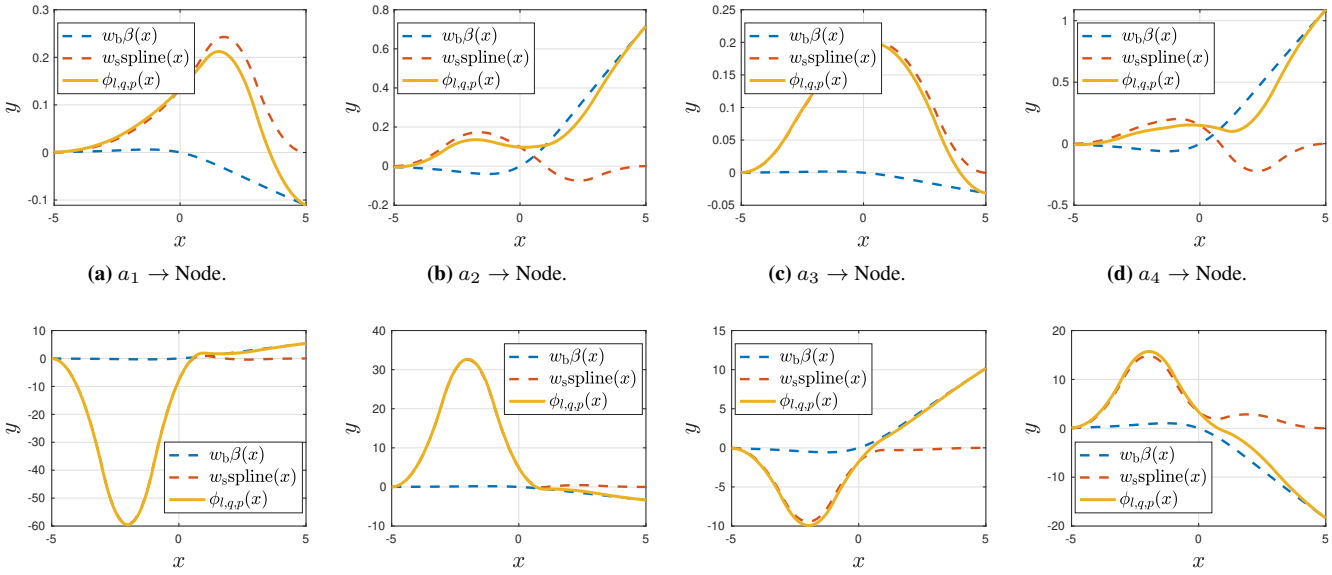
**Table 2:** Parameter values of the Arturia MiniBrute VCF subcircuit shown in Fig. 3.

| Param. | Value | Param. | Value | Param. | Value |
|---|---|---|---|---|---|
| $R_1$ | 10 kΩ | $R_2$ | 1 kΩ | $R_3$ | 2.2 kΩ |
| $R_4$ | 1 kΩ | $R_5$ | 390 Ω | $R_6$ | 2.2 kΩ |
| $R_7$ | 9.47 kΩ | $R_8$ | 909 kΩ | $R_9$ | 1 kΩ |
| $R_{10}$ | 1 TΩ | $R_{11}$ | 1 TΩ | $R_{12}$ | 2.2 MΩ |
| $C_1$ | 2.2 µF | $C_2$ | 47 µF | $C_3$ | 1.5 nF |
| $C_4$ | 1.5 nF | $C_5$ | 1.5 pF | $C_6$ | 2.2 µF |
| $C_7$ | 680 nF | $V_{dd}$ | 12 V | – | – |

### 5.1. Dataset and Training

As discussed in [9, 10], the dataset used to train the neural networks is generated in the Kirchhoff domain. In particular, we construct two separate datasets to characterize both $D_1$/$D_4$ and $D_2$/$D_3$, using the parameters reported in Table 1 and the model in (7). By analyzing the voltage operating ranges within the circuit, the input voltage $v$ is uniformly sampled in the interval $[-0.5, 2.1]$ V for $D_1$/$D_4$, and in $[-0.5, 1.5]$ V for $D_2$/$D_3$, obtaining for both datasets $10^7$ samples. The resulting data are assembled into vectors $\mathbf{v} = [V_{D1}, V_{D2}, V_{D3}, V_{D4}]$ and $\mathbf{i} = [I_{D1}, I_{D2}, I_{D3}, I_{D4}]$, and then mapped into the WD domain by means of (8); the resulting 4-port nonlinear block will be then placed at the root of the connection tree during the simulation.

We retain 20% of the dataset for the test set, while the remaining 80% is used for training. The dataset is organized into input-output pairs of the form $(\mathbf{a}, \mathbf{b})$ with $\mathbf{a} = [a_1, a_2, a_3, a_4]^\mathrm{T}$ and $\mathbf{b} = [b_1, b_2, b_3, b_4]^\mathrm{T}$. For the MLP, the adopted architecture consists of a single hidden layer with 16 neurons and Exponential Linear Unit (ELU) activation functions. The model is implemented in Python using PyTorch and trained with the Adam optimizer (learning rate $1 \times 10^{-4}$ and early stopping after 50 epochs)




**(a)** $a_1 \to$ Node. **(b)** $a_2 \to$ Node. **(c)** $a_3 \to$ Node. **(d)** $a_4 \to$ Node.

**(e)** Node $\to b_1$. **(f)** Node $\to b_2$. **(g)** Node $\to b_3$. **(h)** Node $\to b_4$.

**Figure 4:** KAN's activation functions after training.

by minimizing the Cauchy loss function [28]

$$\mathcal{L} = \frac{1}{D}\sum_{d=1}^{D} \gamma^2 \log\left(1 + \frac{\|\mathbf{b}_d - \hat{\mathbf{b}}_d\|_2^2}{\gamma^2}\right), \qquad (17)$$

where $\gamma = 0.5$, $D$ denotes the number of samples, and $\mathbf{b}_d$ and $\hat{\mathbf{b}}_d$ represent the vectors of true and predicted waves for the $d$-th point pair, respectively. The Cauchy loss has recently gained attention in regression problems due to its robustness, as it reduces the influence of large residuals while behaving similarly to the mean squared error for small residuals. This property improves robustness against outliers and prevents a few large errors from dominating the optimization process [28, 10]. In total, the MLP model comprises 148 parameters, resulting in a Normalized Mean Squared Error (NMSE) on the test set of $2.66 \times 10^{-4}$.

For the KAN architecture, we employ a single hidden layer with one neuron. The remaining KAN hyperparameters are set to $G = 1$ and $o = 2$, resulting in a total of 40 trainable parameters, obtaining thus a reduction of said number of more than 70% when compared against the MLP. Learning rates are selected independently for each architecture through preliminary hyperparameter tuning to ensure stable convergence, rather than enforcing identical optimization settings. The model is trained using Adam (learning rate $1 \times 10^{-3}$ and early stopping after 50 epochs), minimizing the same loss function in (17). This configuration yields an NMSE of $6.71 \times 10^{-4}$ on the test set. Fig. 4 shows the learned edge activation functions, $\phi_{l,q,p}$, after training. In particular, the first row (Figs. 4a–4d) shows the edge functions connecting the input layer to the hidden layer, whereas the second row (Figs. 4e–4h) shows those connecting the hidden layer to the output layer. Such a representation provides insight into how the network models the nonlinear mapping, enabling a direct interpretation of the learned transformations and their contribution to the overall input-output relation.

It is worth noting that both the MLP and the KAN architectures are designed with the aim of minimizing the number of parameters without compromising simulation accuracy. In particular, we select the smallest architectures that achieve an NMSE below a certain threshold in the time-domain simulation (presented in the next subsection), which we set to $1 \times 10^{-2}$. Moreover, we consider the additional constraints of obtaining NMSE values of the same order of magnitude for both models, in order to ensure a fair comparison.

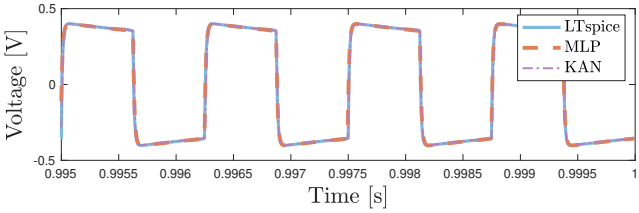



**Figure 5:** Last 5 ms of $V_{\text{out}}$. The comparison among LTspice (blue solid curve), MLP (orange dashed curve), and KAN (purple dot-dashed curve) reveals that both WDF implementations achieve a high degree of accuracy.

### 5.2. Results

This subsection presents the results of our comparative analysis. All tests are conducted on an Apple MacBook Pro with an Apple M4 Processor and are evaluated against a ground-truth simulation performed in LTspice under identical operating conditions.

In particular, to compare the accuracy of the KAN-based WDF against that of the MLP-based implementation, we drive the circuit with a 1-second square wave having fundamental frequency $f_0 = 800$ Hz, amplitude 8 V, and duty cycle 50%. The simulations are performed at a sampling frequency $f_s = 44.1$ kHz with an oversampling factor of four. Both WDF implementations are developed in Python using NumPy, and differ only in the neu-

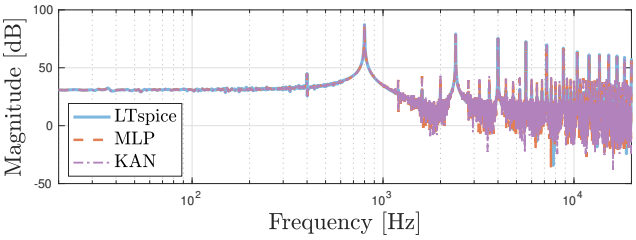



**Figure 6:** Discrete Fourier Transform (DFT) of $V_{\text{out}}$. Curve convention follows what explained for Fig. 5.

ral network used to model the 4-port nonlinear WD block. Using NumPy allows us to avoid the computational overhead associated with PyTorch's heavier data structures, which would otherwise introduce additional runtime costs in the WDF simulation.

Fig. 5 shows the last 5 ms of the output voltage $V_{\text{out}}$. The three curves, namely, the solid blue curve representing LTspice results, the dashed orange curve corresponding to the MLP-based WDF, and the purple dot-dashed curve corresponding to the KAN-based WDF, are overlapped, pointing out the accuracy of the representations. This results in an NMSE with respect to the LTspice reference of $3.8 \times 10^{-3}$ and $6.6 \times 10^{-3}$ for the MLP- and KAN-based implementations, respectively. Fig. 6 shows, instead, a comparison among the Discrete Fourier Transforms (DFTs) of the three signals, from which we evince a coherent spectral content. To further quantify the accuracy of the representations, we compute the Log-Spectral Distance (LSD), taking once again the LTspice results as ground truth. We obtain 1.02 for the WDF based on MLP and 0.82 for the WDF based on KAN, indicating that the two approaches achieve comparable performance, with a slight advantage for the KAN-based implementation in the frequency domain. For the reader's convenience, all the results are summarized in Table 3.

Finally, we compare the performance of the two implementations in terms of real-time execution. To this end, we compute the Real-Time Ratio (RTR) as the average, over $\Gamma = 100$ identical runs, of the ratio between the simulation time $t_{\text{sim}}$ (in seconds) and the total number of processed samples $K$, i.e.,

$$\text{RTR} = \frac{1}{\Gamma} \sum_{j=1}^{\Gamma} \frac{t_{\text{sim},j}\, f_{\text{s}}}{K}\,, \tag{18}$$

obtaining values of 1.31 and 4.74 for the MLP- and KAN-based WDFs, respectively. These results indicate that, although KAN employs a significantly smaller number of parameters, the MLP-based implementation achieves a lower computational cost. We mainly attribute this behavior to the higher number of nonlinear evaluations required by the KAN architecture. In our specific configuration, the KAN model involves 8 B-splines, each expressed as a linear combination of $G + o = 3$ quadratic basis functions (totaling 24 evaluations), whereas the MLP requires only 16 activation function evaluations during the forward pass. This increased computational burden offsets the parameter efficiency of KAN. Moreover, MLP inference on CPUs benefits from highly optimized dense matrix multiplication routines, whereas spline evaluations currently lack an equally mature optimization ecosystem. It is also worth noting that the measured RTR is strictly linked to the implementation-dependent computational cost. With that in mind, we expect that optimized low-level implementations (e.g., in C++ or on dedicated hardware) exploiting dedicated vectorization and/or quantized spline lookup tables could substantially reduce the cost of spline evaluations, narrowing the observed RTR gap.

**Table 3:** Comparison between the MLP- and KAN-based WDF implementations under the simulation scenario described in Sec. 5.

| **Metric** | **MLP** | **KAN** |
|---|---|---|
| Parameters | 148 | 40 |
| NMSE | $3.8 \times 10^{-3}$ | $6.6 \times 10^{-3}$ |
| LSD | 1.02 | 0.82 |
| RTR | 1.31 | 4.74 |

Overall, these findings are consistent with those reported in [17], indicating that KANs can achieve comparable accuracy with a reduced number of parameters. This characteristic is particularly appealing for resource-constrained scenarios, such as embedded and FPGA-based implementations [29]. We would like to point out that KANs are likely to be most advantageous in scenarios where MLPs require a large number of parameters, as the evaluation of a limited set of B-spline functions may become more efficient than computing a large number of activation functions. In the context of VA modeling with WDFs and neural networks, this situation naturally arises when dealing with circuits featuring multiple multiport nonlinearities. In such cases, we believe that KANs may provide a favorable trade-off between accuracy and model complexity, representing a promising direction for future research and practical implementations.

## 6. CONCLUSIONS

In this article, we investigated the use of Kolmogorov-Arnold Networks (KANs) as an alternative to Multi-Layer Perceptrons (MLPs) for Virtual Analog modeling within the Wave Digital Filter (WDF) framework. Through a controlled case study, we showed that KANs achieved accuracy comparable to that of MLPs while requiring a significantly reduced number of parameters. At the same time, we observed that this gain came at the cost of increased computational complexity during inference. These results suggested that KANs represent a viable alternative to MLPs, particularly in scenarios where memory footprint is a primary constraint. Moreover, we argued that their reduced parameter count can be particularly advantageous when modeling circuits with a large number of nonlinear elements, potentially enabling more cost-effective and scalable implementations, and providing the community with a practical alternative for resource-constrained applications.

## 7. REFERENCES

[1] U. Zölzer, *DAFX: Digital Audio Effects*, second edition ed. John Wiley and Sons, Mar. 2011.

[2] T. Hélie, "Volterra Series and State Transformation for Real-Time Simulations of Audio Circuits Including Saturations: Application to the Moog Ladder Filter," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 747–759, May 2010.

[3] A. Wright, E. P. Damskägg, and V. Välimäki, "Real-Time Black-Box Modelling With Recurrent Neural Networks," in

*Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx)*, Birmingham, UK, Sep. 2019, pp. 1–8.

[4] T. Vanhatalo, P. Legrand, M. Desainte-Catherine, P. Hanna, A. Brusco, G. Pille, and Y. Bayle, "A Review of Neural Network-Based Emulation of Guitar Amplifiers," *Applied Sciences*, vol. 12, no. 12, p. 5894, Jun. 2022.

[5] A. Fettweis, "Wave Digital Filters: Theory and Practice," *Proceedings of the IEEE*, vol. 74, no. 2, pp. 270–327, 1986.

[6] K. J. Werner, J. O. Smith, and J. S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx)*, Trondheim, Norway, Norway, Dec. 2015.

[7] J. Chowdhury and C. J. Clarke, "Emulating Diode Circuits with Differentiable Wave Digital Filters," in *Proceedings of the 19th Sound and Music Computing Conference*, Saint-Étienne, France, Jun. 2022.

[8] C. C. Darabundit, D. Roosenburg, and J. O. Smith III, "Neural Net Tube Models for Wave Digital Filters," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx)*, Vienna, Austria, Sep. 2022.

[9] O. Massi, R. Giampiccolo, A. Bernardini, and A. Sarti, "Explicit Vector Wave Digital Filter Modeling of Circuits with a Single Bipolar Junction Transistor," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx)*, Copenhagen, Denmark, Sep. 2023, pp. 172–179.

[10] R. Giampiccolo, S. C. Gafencu, and A. Bernardini, "Explicit Modeling of Audio Circuits with Multiple Nonlinearities for Virtual Analog Applications," *IEEE Open Journal of Signal Processing*, pp. 156–164, 2025.

[11] 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, Feb. 2009.

[12] A. Bernardini, P. Maffezzoni, and A. Sarti, "Vector Wave Digital Filters and Their Application to Circuits With Two-Port Elements," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 1269–1282, Mar. 2021.

[13] O. Massi, S. Yang, R. Giampiccolo, and A. Bernardini, "Explicit Neural Network-Based Modeling of Time-Varying Circuits with a Single BJT in the Wave Digital Domain," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, London, UK, May 2025.

[14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[15] A. N. Kolmogorov, "On the Representation of Continuous Functions of Several Variables as Superpositions of Continuous Functions of a Smaller Number of Variables," *Doklady Akademii Nauk SSSR*, vol. 108, no. 2, pp. 179–182, 1956.

[16] ——, "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition," *Doklady Akademii Nauk SSSR*, vol. 114, no. 5, pp. 953–956, 1957.

[17] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov–Arnold Networks," in *Proceedings of the 13th International Conference on Learning Representations*, Singapore, Singapore, Apr. 2025.

[18] N. Steiner, "Synthacon Filter," *Electronic Design*, vol. 25, Dec. 1974.

[19] Arturia, "Minibrute," accessed: 2026-02-27. [Online]. Available: https://www.arturia.com/products/hardware-synths/minibrute/resources

[20] G. Kubin, "Wave Digital Filters: Voltage, Current, or Power Waves?" in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 10, Tampa, FL, USA, 1985, pp. 69–72.

[21] A. Bernardini, K. J. Werner, J. O. Smith, and A. Sarti, "Generalized Wave Digital Filter Realizations of Arbitrary Reciprocal Connection Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 694–707, Feb. 2019.

[22] A. Bernardini, P. Maffezzoni, and A. Sarti, "Linear Multistep Discretization Methods with Variable Step-Size in Nonlinear Wave Digital Structures for Virtual Analog Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1763–1776, Nov. 2019.

[23] G. O. Martens and K. Meerkötter, "On N-Port adaptors for Wave Digital Filters with Application to a Bridged-Tee Filter," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Munich, Germany, Apr. 1976, pp. 514–517.

[24] A. Bernardini, P. Maffezzoni, L. Daniel, and A. Sarti, "Wave-Based Analysis of Large Nonlinear Photovoltaic Arrays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 4, pp. 1363–1376, Apr. 2018.

[25] S. D'Angelo, L. Gabrielli, and L. Turchet, "Fast Approximation of the Lambert W Function for Virtual Analog Modelling," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx)*, Birmingham, UK, Sep. 2019, pp. 238–244.

[26] R. Giampiccolo, M. G. de 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, Oct. 2021.

[27] C. de Boor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.

[28] T. Mlotshwa, H. v. Deventer, and A. S. Bosman, "Cauchy Loss Function: Robustness Under Gaussian and Cauchy Noise," *arXiv preprint*, 2023, arxiv.2302.07238.

[29] R. Michon, M. Ducceschi, P. Cochard, T. Skare, C. J. o. Webb, and R. Russo, "Evaluating CPU, GPU, and FPGA Performance in the Context of Modal Reverberation: A Comparative Analysis," *Frontiers in Signal Processing*, vol. 5, p. 1522604, Apr. 2025.