

Arbitrary-Order Antiderivatives of Canonical Piecewise-Linear Functions

Alberto Bernardini, *Senior Member, IEEE*, Riccardo Giampiccolo, *Member, IEEE*

Abstract—Canonical Piecewise-Linear (CPWL) functions can approximate any continuous single-valued scalar function with arbitrary accuracy through a global and explicit representation. This letter presents a general closed-form expression for computing p -th order antiderivatives of CPWL functions. The proposed analytical formulation is directly parameterized by the CPWL coefficients and avoids recursive integration schemes and symbolic manipulation. Numerical results demonstrate that the proposed method enables efficient and accurate evaluation of p -th order antiderivatives while maintaining low computational cost and memory footprint.

Index Terms—Canonical Piecewise-Linear functions, Antiderivative, data integration.

I. INTRODUCTION

CANONICAL Piecewise-Linear (CPWL) functions [1], [2] can approximate any single-valued nonlinear function with arbitrary accuracy by means of a global and explicit representation that employs a low number of parameters and requires neither lookup tables (LUTs) nor iterative evaluation techniques. Owing to these appealing properties, CPWL functions are used both to explicitly describe implicitly defined scalar single-valued nonlinearities and to provide global representations of unknown functions for which only a finite set of points is available.

Despite these advantages, CPWL functions have so far been less popular than other methodologies in the literature. Indeed, LUT-based approaches are still widely adopted [3], largely because of their simplicity, ease of implementation, and longstanding use in both industrial and academic practice. As a result, LUTs often remain the default choice for representing and constructing piecewise-linear approximations of nonlinear functions or for fitting measured data, although alternative approaches exist [4]. In general, all the aforementioned methods require the selection of breakpoints in the curve to be represented, which can be chosen directly from the reference nonlinear function or determined through more sophisticated optimization procedures [5].

The literature also reports several generalizations of CPWL function representations. For instance, multidimensional extensions are discussed in [1], [6]–[8], while smooth canonical piecewise representations have been proposed to address the discontinuity of first-order derivatives of CPWL functions [9]–[11]. These approaches mainly aim to enforce differentiability in the CPWL function itself while preserving its global structure.

The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Piazza L. Da Vinci 32, 20133 Milano, Italy (e-mails: [alberto.bernardini,riccardo.giampiccolo]@polimi.it).

As far as antiderivatives of piecewise functions are concerned, they are typically obtained through successive segment-wise integration steps, often requiring the recursive computation of lower-order antiderivatives and the explicit enforcement of continuity conditions. Such strategies are commonly adopted in antiderivative antialiasing (ADAA) frameworks, where antiderivatives play a central role in mitigating aliasing artifacts in nonlinear discrete-time systems [12]–[18], especially when the global analytical expressions of antiderivatives are not available or are too computationally demanding. Although other techniques allow for analytical computations of antiderivatives of piecewise polynomial functions [16], to the best of our knowledge, no existing work derives global closed-form expressions for arbitrary-order antiderivatives of CPWL functions directly from their canonical parametrization.

In this letter, building on the CPWL parametrization, we present a compact analytical expression that enables the computation of the p -th order antiderivative of a continuous CPWL function directly from the selected breakpoints, without the need to compute the $(p-1)$ -th order antiderivative. The proposed formulation eliminates the need for search operations, interpolation, and numerical integration typically required when using LUTs, thereby avoiding successive integration steps and symbolic manipulation. We compare the proposed methodology with a recent LUT-based technique to compute antiderivatives from a finite set of points [3] and show that the proposed formulation achieves higher accuracy when the number of breakpoints is kept small, while exhibiting a comparable computational cost.

II. BACKGROUND ON CPWL FUNCTIONS

Real-valued scalar CPWL functions [1] are characterized by a set of segments and a set of points shared by adjacent segments, known as *breakpoints*, with coordinates $(x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)$, where $M \in \mathbb{N}$ and $M < +\infty$. A continuous single-valued CPWL function $F_0(x) : \mathbb{R} \rightarrow \mathbb{R}$ is given by [1]

$$F_0(x) = \mu_0 + \mu_1 x + \sum_{j=1}^M \eta_j |x - x_j|, \quad (1)$$

where $|\cdot|$ indicates the absolute value, $M+1$ is the number of segments (from the leftmost with index $j=0$ to the rightmost with index $j=M$), and M is the number of breakpoints with coordinates (x_j, y_j) (from the leftmost with index $j=1$ to the rightmost with index $j=M$). We also define the parameters

$$\mu_1 = 0.5(m_0 + m_M), \quad (2)$$

$$\eta_j = 0.5(m_j - m_{j-1}) \quad \text{with } j = 1, \dots, M, \quad (3)$$

$$\mu_0 = F_0(0) - \sum_{j=1}^M \eta_j |x_j|, \quad (4)$$

where m_0 is the slope of the leftmost segment, m_M is the slope of the rightmost segment, $F_0(0)$ is the value assumed by the function at $x = 0$, whereas m_j is the slope of the j th segment, with $j = 1, \dots, M-1$, and it is defined as $m_j = (y_{j+1} - y_j)/(x_{j+1} - x_j)$. Let us assume that the slopes of the leftmost and the rightmost segments, corresponding to the intervals $[-\infty, x_1]$ and $[x_M, +\infty]$, are set as $m_0 = m_1$ and $m_M = m_{M-1}$, respectively. Finally, it is worth noting that a CPWL representation in the form given in (1) is obtained if and only if $x_1 < x_2 < \dots < x_M$, where the points x_j are not necessarily uniformly spaced.

III. ANTIDERIVATIVES OF CPWL FUNCTIONS

Let us consider the generic CPWL function $F_0(x)$, also referred to as the antiderivative function of 0-th order. We show that CPWL functions with global formulation (1) allow the analytical derivation of antiderivatives. Indeed, Table I reports the first-, second-, and third-order antiderivatives of $F_0(x)$ computed by applying successive integration operations to the canonical representation in (1). The first order antiderivative $F_1(x)$ of (1) is defined such that $F_0(x) = dF_1(x)/dx$ holds true $\forall x$. Hence $F_1(x)$ is C^1 smooth and continuous. Among the infinite primitive functions of $F_0(x)$, we consider the unique one satisfying $F_1(0) = 0$. Since any antiderivative is defined up to an additive constant, this condition removes the resulting ambiguity and yields a canonical expression that simplifies further analysis. In fact, it can be verified that (5) in Table I satisfies $F_1(0) = 0$. Then, the second-order antiderivative of $F_0(x)$ in (6) is the primitive function of (5) that passes through the origin, and, in turn, the third-order antiderivative of $F_0(x)$ in (7) is the primitive function of (6) that passes through the origin.

In general, the antiderivative of order p of $F_0(x)$, with $p \geq 1$, can be obtained by taking the integral of the antiderivative of $F_0(x)$ of order $p-1$. It follows that, by induction, we can derive a general formula, which expresses the p -th order antiderivative of $F_0(x)$ as

$$F_p(x) = \mu_0 \frac{x^p}{p!} + \mu_1 \frac{x^{p+1}}{(p+1)!} + \sum_{j=1}^M \eta_j \varphi_j(p, x), \quad (8)$$

where

$$\varphi_j(p, x) = \begin{cases} \Phi_j(p, x), & (x - x_j) \geq 0 \wedge x_j < 0 \\ \Psi_j(p, x), & (x - x_j) \geq 0 \wedge x_j \geq 0 \\ -\Phi_j(p, x), & (x - x_j) < 0 \wedge x_j \geq 0 \\ -\Psi_j(p, x), & (x - x_j) < 0 \wedge x_j < 0 \end{cases} \quad (9)$$

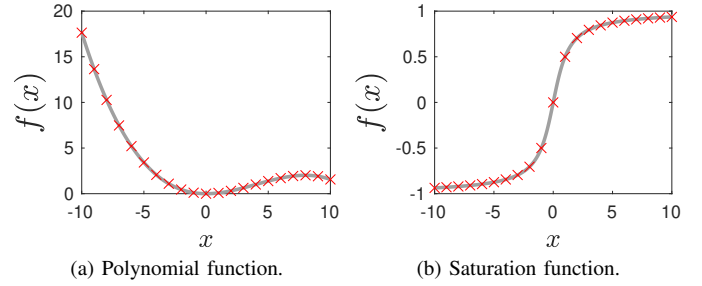


Fig. 1: Functions considered in our examples of application. Markers indicate the 21 points used for tuning the methods.

with

$$\begin{aligned} \Phi_j(p, x) &= \frac{x^p (x - (p+1)x_j)}{(p+1)!}, \\ \Psi_j(p, x) &= \frac{(x - x_j)^{p+1}}{(p+1)!} + \sum_{k=0}^{p-1} \frac{(p-k)(x - x_j)^k x_j^{p+1-k}}{k! (p+1-k)!}. \end{aligned} \quad (10)$$

Equation (8) allows for the direct computation of the p -th order antiderivative of a CPWL function such that $F_p(0) = 0$ without the need of recursively applying integration operations. It can also be verified that $F_p(x)$ is continuous, i.e.,

$$\lim_{x \rightarrow x_c} F_p(x) = F_p(x)|_{x=x_c}, \quad \forall x_c \in \mathbb{R}, \quad (11)$$

and, trivially, that it is C^p smooth.

IV. EXAMPLES OF APPLICATION

In this section, we compare the performance of the proposed methodology against the technique used in [3] to compute antiderivatives of nonlinear functions starting from a set of points. To ease the analysis, we consider two case studies that allow for the analytical computation of antiderivatives: an arbitrary polynomial function and a saturation function that is of interest in audio signal processing applications.

In particular, the method proposed in [3] entails uniformly sampling the function $f(x)$ at a finite number of points M over the interval $[a, b]$, equally spaced by Δx . The authors tested the methodology using 128, 1024, and 8192 breakpoints, considering only antiderivatives of low orders. Said points are used to perform numerical integration via the Riemann sum (or rectangular rule), and the results are stored in an LUT. The LUT corresponding to the antiderivative of 1-st order is subsequently obtained using an interpolation scheme; in our analysis, we select cubic interpolation. Higher-order antiderivatives are obtained by recursively applying the aforementioned processing steps. Finally, it is worth pointing out that all algorithms are implemented and tested in a plain MATLAB script.

A. Polynomial Function

Let us consider the polynomial function $f(x)$ shown in Fig. 1a, defined through the following equation

$$f(x) = \frac{(-4x^3 + 48x^2 - 2x)}{500}, \quad (12)$$

TABLE I: First-, Second-, and Third-Order Antiderivatives of Canonical Piecewise-Linear Functions.

1st-order Antiderivative		2nd-order Antiderivative	
$F_1(x) = \mu_0 x + \mu_1 \frac{x^2}{2} + \sum_{j=1}^M \eta_j \varphi_j(1, x)$	(5)	$F_2(x) = \mu_0 \frac{x^2}{2} + \mu_1 \frac{x^3}{6} + \sum_{j=1}^M \eta_j \varphi_j(2, x)$	(6)
$\varphi_j(1, x) = \begin{cases} \frac{x(x-2x_j)}{2}, & (x-x_j) \geq 0 \wedge x_j < 0 \\ \frac{(x-x_j)^2}{2} + \frac{x_j^2}{2}, & (x-x_j) \geq 0 \wedge x_j \geq 0 \\ -\frac{x(x-2x_j)}{2}, & (x-x_j) < 0 \wedge x_j \geq 0 \\ -\frac{(x-x_j)^2}{2} - \frac{x_j^2}{2}, & (x-x_j) < 0 \wedge x_j < 0 \end{cases}$		$\varphi_j(2, x) = \begin{cases} \frac{x^2(x-3x_j)}{6}, & (x-x_j) \geq 0 \wedge x_j < 0 \\ \frac{(x-x_j)^3 + 3x_j^2(x-x_j) + 2x_j^3}{6}, & (x-x_j) \geq 0 \wedge x_j \geq 0 \\ -\frac{x^2(x-3x_j)}{6}, & (x-x_j) < 0 \wedge x_j \geq 0 \\ -\frac{(x-x_j)^3 + 3x_j^2(x-x_j) + 2x_j^3}{6}, & (x-x_j) < 0 \wedge x_j < 0 \end{cases}$	
3rd-order Antiderivative			
$F_3(x) = \mu_0 \frac{x^3}{6} + \mu_1 \frac{x^4}{24} + \sum_{j=1}^M \eta_j \varphi_j(3, x)$	$\varphi_j(3, x) = \begin{cases} \frac{x^3(x-4x_j)}{24}, & (x-x_j) \geq 0 \wedge x_j < 0 \\ \frac{(x-x_j)^4 + 6x_j^2(x-x_j)^2 + 8x_j^3(x-x_j) + 3x_j^4}{24}, & (x-x_j) \geq 0 \wedge x_j \geq 0 \\ -\frac{x^3(x-4x_j)}{24}, & (x-x_j) < 0 \wedge x_j \geq 0 \\ -\frac{(x-x_j)^4 + 6x_j^2(x-x_j)^2 + 8x_j^3(x-x_j) + 3x_j^4}{24}, & (x-x_j) < 0 \wedge x_j < 0 \end{cases}$		

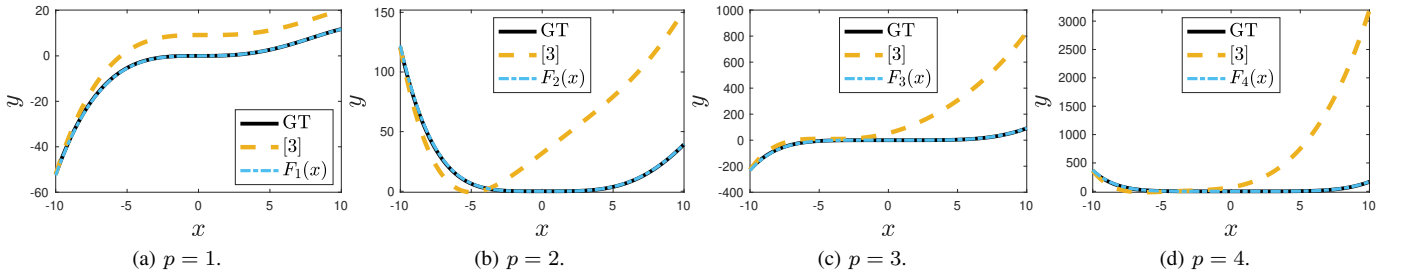


Fig. 2: First four antiderivatives of (12). The proposed method in light blue and the baseline method in orange are both tuned over 21 points. The ground-truth (GT) in black represents the analytical solution.

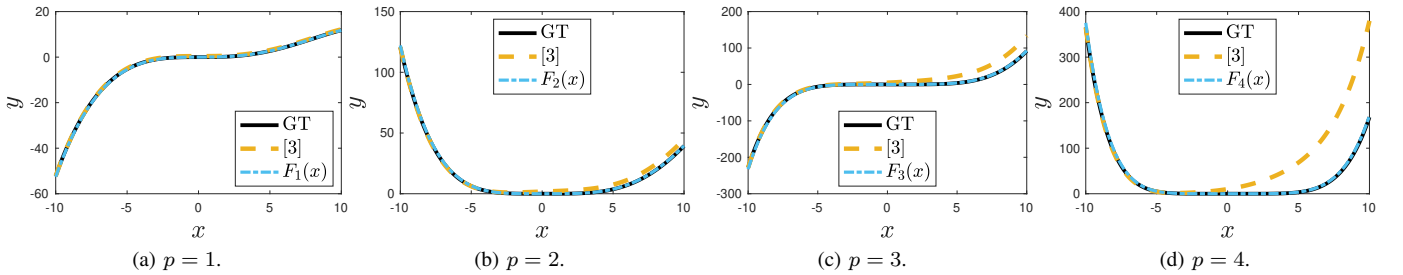


Fig. 3: First four antiderivatives of (12). The proposed method in light blue is tuned over 21 points, while the baseline method in orange over 401 points. The ground-truth (GT) in black represents the analytical solution.

where the red crosses mark the $M = 21$ breakpoints selected for tuning the methods under test and obtained by uniformly sampling the range $[-10, 10]$ with $\Delta x = 1$. The test points, instead, are obtained by uniformly sampling the same range but with $\Delta x = 0.01$ and amount to 2001.

Fig. 2 shows the comparison between the proposed method (in light blue), the baseline method (in orange), and the analytical solutions (ground-truth, in black) for the first- (Fig. 2a), second- (Fig. 2b), third- (Fig. 2c), and fourth- (Fig. 2d) order antiderivatives of (12). As already anticipated in Sec. III,

it is evident that the obtained antiderivatives are continuous although the 0-th order function is indeed piecewise-linear. Then, while the CPWL curves closely match the ground-truth (GT) curve over the entire domain, the baseline curves present noticeable errors. For instance, in Fig. 2a, the baseline curve begins to deviate clearly around $x = -7.5$, owing to the quadrature error introduced by the Riemann sum. To quantify the deviation between the curves, Table II reports the Normalized Mean Squared Error (NMSE), computed between the method under test and the GT, for each antiderivative order

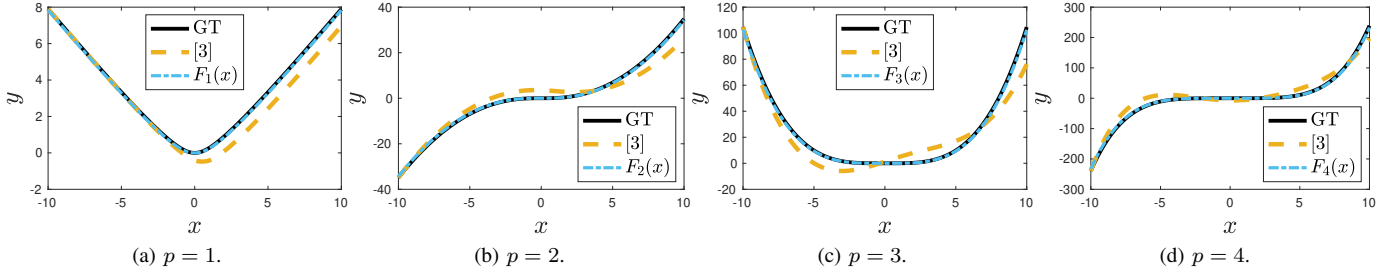


Fig. 4: First four antiderivatives of (13). The proposed method in light blue and the baseline method in orange are both tuned over 21 points. The ground-truth (GT) in black represents the analytical solution.

p . The CPWL-based curve achieves an NMSE, averaged over the four orders, of 4.42×10^{-4} , while the baseline yields an NMSE of 43.9. Moreover, by examining the first section of the table and by comparing the curves in Figg. 2a and 2d, we can notice that the quadrature error increases with the order p , while the antiderivatives computed via (8) maintain comparable accuracy across all orders. In addition, the proposed and baseline methods exhibit comparable computational costs, with the proposed approach performing slightly faster. Specifically, the results (averaged on 100 runs on a Apple Silicon M1 Pro processor) show that the baseline method processes the 2001 test points in approximately 9.2×10^{-3} s, whereas our approach in 3.4×10^{-3} s.

To further assess the proposed methodology, we repeated the experiment by tuning the proposed method still with $M = 21$ breakpoints, but the baseline method with $M = 401$ points (obtained by setting $\Delta x = 0.05$). Fig. 3 shows the results obtained under this configuration. The third row of the first section in Table II reports, instead, the NMSE computed across the four antiderivative orders. Although the quadrature error of the baseline method is substantially reduced (e.g., with the NMSE for order $p = 1$ being now 7.37×10^{-4}), the NMSE averaged over the four orders amounts to 0.23, remaining significantly higher than that achieved by the proposed approach. Finally, it is worth pointing out that, despite the increase in the number of breakpoints, the runtime remains almost unchanged compared to the previous case.

B. Saturation Function

As a second example, let us consider the saturation function $f(x)$ shown in Fig. 1b, defined through the following equation

$$f(x) = \frac{2}{\pi} \operatorname{atan}(x), \quad (13)$$

where the red crosses mark the $M = 21$ breakpoints as in the previous example. We consider the same range, Δx , and test points as in Sec. IV-A.

Fig. 4 shows the comparison between the proposed method (in light blue), the baseline method (in orange), and the analytical solutions (GT in black) for the first four antiderivates of (13), when the methods under test are both tuned over $M = 21$ breakpoints. The second section of Table II shows the NMSE computed under this configuration. The average NMSE achieved by our approach amounts to 4.7×10^{-4} ,

TABLE II: NMSE Between Ground-Truth and the Methods Under Test for Orders p and Numbers of Breakpoints M .

Func.	Method	M	$p = 1$	$p = 2$	$p = 3$	$p = 4$
(12)	Ours	21	8.6×10^{-5}	2.22×10^{-4}	4.92×10^{-4}	9.7×10^{-4}
(12)	[3]	21	0.32	3.60	28.22	143.53
(12)	[3]	401	7.37×10^{-4}	9.35×10^{-3}	0.1	0.8
(13)	Ours	21	1.3×10^{-4}	3.0×10^{-4}	5.6×10^{-4}	8.8×10^{-4}
(13)	[3]	21	2.1×10^{-2}	5.3×10^{-2}	5.4×10^{-2}	4.8×10^{-2}
(13)	[3]	401	5.35×10^{-5}	1.5×10^{-4}	2.1×10^{-4}	2.2×10^{-4}

whereas the baseline yields 4.4×10^{-2} , with the runtimes being similar to those obtained in the previous example. By contrast, the results in the last row of Table II show that tuning the baseline method with $M = 401$ breakpoints yields a slight accuracy improvement over our approach, with an average NMSE of 1.6×10^{-4} . We attribute this result to the lower quadrature error in the Riemann sum caused by the presence of relatively flat regions with respect to the function analyzed in Sec. IV-A. Nonetheless, the proposed approach turns out to be the preferred choice when aiming for higher accuracy while limiting the number of breakpoints on which the curve is tuned (and thus the memory footprint), making it particularly promising in scenarios with scarce availability of data.

As a final remark, preliminary experiments applying the proposed methodology to ADAA [13] yield lower noise than [3] when a limited number of breakpoints is available. Further implications will be investigated in future work.

V. CONCLUSIONS

In this letter, we presented an explicit closed-form expression to compute the p -th order antiderivative of a continuous CPWL function. The analytical formula leads to continuous representations and enables the direct computation of the p -th order antiderivate without computing the previous $p - 1$ antiderivatives, as typically required by other approaches in the literature. Compared to a recent method for computing antiderivatives of functions stored in an LUT [3], the proposed approach turned out to be characterized by higher accuracy while maintaining a comparable computational cost, which makes it promising for applications such as dynamic system identification [19], [20] and antiderivative antialiasing in non-linear audio signal processing [12]–[18].

REFERENCES

- [1] 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, June 1977.
- [2] A. Bernardini and A. Sarti, "Canonical Piecewise-Linear Representation of Curves in the Wave Digital Domain," in *Proc. 25th Eur. Signal Proces. Conf. (EUSIPCO)*, Aug. 2017, pp. 1125–1129.
- [3] L. Gabrielli and S. Squartini, "Simplifying Antiderivative Antialiasing with Lookup Table Integration," in *Proceedings of the 28th International Conference on Digital Audio Effects (DAFx25)*, Ancona, Italy, Sep. 2025, pp. 7–14.
- [4] A. Magnani and S. P. Boyd, "Convex piecewise-linear fitting," *Optimization and Engineering*, vol. 10, p. 17, 2009.
- [5] R. A. DeVore, "Nonlinear approximation," *Acta Numerica*, vol. 7, p. 51–150, 1998.
- [6] V. M. Jimenez-Fernandez, M. Jimenez-Fernandez, H. Vazquez-Leal, U. A. Filobello-Nino, C. H. Castañeda-Roldan, and V. M. Tlapa-Carrera, "A New Methodology to Extend the Canonical Piecewise-Linear Model from One to Two Dimensions," *National Academy Science Letters*, vol. 44, no. 3, pp. 221–224, Jun. 2021.
- [7] C. Kahlert and L. O. Chua, "A generalized canonical piecewise-linear representation," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 373–383, Mar. 1990.
- [8] P. Julian, "The complete canonical piecewise-linear representation: functional form for minimal degenerate intersections," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 3, pp. 387–396, Mar. 2003.
- [9] J. N. Lin and R. Unbehauen, "Canonical representation: from piecewise-linear function to piecewise-smooth functions," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 7, pp. 461–468, July 1993.
- [10] M. Lazaro, I. Santamaria, C. Pantaleon, A. M. Sanchez, A. T. Puente, and T. Fernandez, "Smoothing the canonical piecewise-linear model: an efficient and derivable large-signal model for mesfet/hemt transistors," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 184–192, Feb 2001.
- [11] V. M. Jimenez-Fernandez, M. Jimenez-Fernandez, H. Vazquez-Leal, E. Munoz-Aguirre, H. H. Cerecedo-Nunez, U. A. Filobello-Nino, and F. J. Castro-Gonzalez, "Transforming the canonical piecewise-linear model into a smooth-piecewise representation," *Springerplus*, vol. 5, no. 1, p. 1612, 2016.
- [12] J. D. Parker, V. Zavalishin, and E. Le Bivic, "Reducing the aliasing of nonlinear waveshaping using continuous-time convolution," in *Proc. 19th Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 5 – 9 2016.
- [13] S. Bilbao, F. Esqueda, J. D. Parker, and V. Välimäki, "Antiderivative antialiasing for memoryless nonlinearities," *IEEE Signal Processing Letters*, vol. 24, no. 7, pp. 1049–1053, July 2017.
- [14] M. Holters, "Antiderivative Antialiasing for Stateful Systems," *Applied Sciences*, vol. 10, no. 1, p. 20, 2020.
- [15] D. Albertini, A. Bernardini, and A. Sarti, "Antiderivative Antialiasing Techniques in Nonlinear Wave Digital Structures," *Journal of the Audio Engineering Society*, vol. 69, no. 7/8, pp. 448–464, 2021.
- [16] K. J. Werner and E. Azelborn, "Antialiasing Piecewise Polynomial Waveshapers," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx23)*, Copenhagen, Denmark, Sep. 2023, pp. 180–187.
- [17] C. Bennett and S. Hopman, "Antialiasing for simplified nonlinear Volterra models," *Journal of the Audio Engineering Society*, vol. 70, no. 9, pp. 690–697, 2022.
- [18] L. Gabrielli, S. D'Angelo, P. P. L. Pastina, and S. Squartini, "Antiderivative antialiasing for arbitrary waveform generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2743–2753, 2022.
- [19] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, p. e1602614, 2017.
- [20] K. Kaheman, J. N. Kutz, and S. L. Brunton, "SINDy with integral formulations," in *IFAC-PapersOnLine*, vol. 52, no. 1, 2019, pp. 510–515.