

Smoothing the High Level Canonical Piecewise-Linear Model by an Exponential Approximation of its Basis-Function

Victor M. Jimenez-Fernandez¹, Maribel Jimenez-Fernandez²,
Hector Vazquez-Leal¹, Uriel A. Filobello-Nino¹, Francisco J. Castro-Gonzalez¹

¹ Universidad Veracruzana, Facultad de Instrumentación Electrónica, Xalapa, Mexico

² Universidad Veracruzana, Instituto de Ciencias Básicas, Xalapa, Mexico

vicjimenez@uv.mx, maribjimenez@uv.mx, hvazquez@uv.mx

Abstract. Piecewise-linear models constitute an attractive alternative to construct a function whose graph fits a finite set of discrete points. These models are preferably selected over other approximation strategies like polynomials or splines. Although there are several piecewise-linear models reported in literature, the so-called High Level Canonical has the remarkable advantage of emerging from a well-structured algorithmic methodology to efficiently determine the parameters of any given piecewise-linear function. However, as it happens in all other piecewise-linear models, it also has the problem of lack of differentiability at the breakpoints. In order to solve this problem, an approach based on an exponential approximation of the basis-function is proposed as a strategy to transform the High Level Canonical piecewise-linear model into a smooth-piecewise one. This mathematical transformation ensures the existence and continuity of the n th-order derivatives of the resulting smooth model. Besides of this, it is observed that by applying the piecewise-linear to smooth transformation, the number of terms of the resulting smooth representation can significantly be reduced due to a great number of them can be approximated by a line equation. In order to verify the effectiveness of this proposal, numerical simulations performed on one-dimensional and two-dimensional functions are reported.

Keywords. High-Level-Canonical, piecewise-linear, smoothing, basis-function, approximation.

1 Introduction

As part of an experimental process, it is common to face up to situations where only

a limited amount of data is available and the estimation of values between consecutive data points is needed, either for predicting results or inferring conclusions from data. Although this problem is traditionally approached by function approximation methods based on polynomial interpolation, a preferred alternative can be found in the approximation techniques based on piecewise-linear models [7, 16, 17, 5, 4] which consists in connecting consecutive data points by straight line segments through a continuous function. However, piecewise-linear models have the shortcomings of having zero curvature between data points, exhibiting abrupt changes at the breakpoints, and having undefined derivatives. It is precisely the lack of differentiability what limits their application in such cases where, as a result of derivative computations, the function $\frac{d^n f(x)}{dx^n}$ (where n indicates a derivative order and $f(x)$ is an approximate function) becomes undefined or zero. With the purpose of incorporating the derivation capability into piecewise-linear functions, the strategy of smoothing the basis-function in a piecewise-linear model is adopted. In accordance with numerical simulations performed in a later section, it can be demonstrated the efficacy of this proposal. Although there are many relevant piecewise-linear models, in this paper the High Level Canonical model developed by Pedro-Julian et al. [12] is chosen as a study case due to the fact that it can be constructed by a well-known algorithmic methodology, and by such methodology, its basis-function can easily be identified.

The paper is organized as follows. In Section 2, the absolute-value function that serves as kernel for the High Level Canonical piecewise-linear model is presented. Section 3 describes the methodology for constructing smooth-piecewise functions. Section 4 shows the application of such methodology by illustrative examples (for one- and two-dimensional domains). In Section 5 a comparative analysis and discussion about the curve fitting accuracy that can be achieved through the proposed strategy is exposed. The comparison is done among polynomial, splines, smooth-piecewise, and standard piecewise-linear approximation techniques. Finally, Section 6 presents the concluding remarks of this work.

2 Absolute-value Basis-function

A relationship between piecewise-linear models and the absolute-value function is established and reported in many references. Mathematical expressions such as the canonical form of Chua-Kang [3, 15, 1, 2], the High Level Canonical representation of Pedro-Julian et al. [11, 8], the Guzeliis [6] model, and the Kahlert [13, 14] description are emblematic examples where this function appears implicitly. Specifically, in the High Level Canonical model the basis-function (also expressed in terms of absolute-value functions) is outlined and easily identified in a detailed methodology by which the approximation function can be constructed. Moreover, this construction methodology is widely reported in literature [10, 9]. For these reasons, the smoothing technique which is proposed in this paper takes as reference the basis-function $\gamma(f_i, f_j)$ used in the Pedro-Julian et al. [8] model.

$$\gamma(f_i, f_j) = \frac{1}{4} (||-f_i| + f_j| - |-f_i + |f_j||) + \frac{1}{4} (|-f_i| + |f_j| - |-f_i + f_j|), \quad (1)$$

where f_i and f_j are linear equations used to sketch the linear partitions of the function domain.

3 Methodology for the Construction of Smooth-Piecewise Functions

Based on the methodology of Pedro-Julian et al. [8], our proposal for the construction of

smooth-piecewise functions can be summarized as follows.

1. Consider as input the set D of N -data that contains the breakpoints coordinates of an arbitrary piecewise-linear function. For the one-dimensional case $D = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_i, Y_i)\}$, while for the two-dimensional case $D = \{(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), \dots, (X_i, Y_i, Z_i)\}$ with $i = 1, 2, \dots, N$.
2. Define the basis-function $\gamma(f_i, f_j)$. In this step, a suitable approximation for the absolute-value function is used. In accordance with reference [18] a smooth approximation to absolute-value function is given by

$$|x| = (x)_+ + (-x)_+, \quad (2)$$

where the plus function $(x)_+$ can be approximated by

$$(x)_+ \approx f_a(x, \alpha) = x + \frac{1}{\alpha} \log(1 + e^{(-\alpha x)}). \quad (3)$$

After combining equation (3) with equation (2), the following smooth approximation is obtained:

$$|x| = (x)_+ + (-x)_+ \approx f_a(x, \alpha) + f_a(-x, \alpha) = \frac{1}{\alpha} \left[\log(1 + e^{(-\alpha x)}) + \log(1 + e^{(\alpha x)}) \right]. \quad (4)$$

After numerical simulations on equation (4), a slight deviation from the absolute-value function can be observed. In order to obtain more accurate fitting, a constant β is included as

$$|x| = \frac{\beta}{\alpha} \left[\log(1 + e^{(-\alpha x)}) + \log(1 + e^{(\alpha x)}) \right]. \quad (5)$$

However, taking into account the logarithmic change-of-base formula $\log(x) = \frac{\ln(x)}{\ln(10)}$, equation (5) can be recast as follows:

$$|x| = \frac{\beta}{\alpha \ln(10)} \left[\ln(1 + e^{(-\alpha x)}) \right] + \frac{\beta}{\alpha \ln(10)} \left[\ln(1 + e^{(\alpha x)}) \right],$$

being $\alpha = 10^6$ and $\beta = 2.3$ appropriate values to achieve small tolerance results. An absolute-value function approximation for $\alpha = 4, 6, 8, \dots, 20$ and $\beta = 2.3$ can be observed in Figure 1. In this figure the thin traces (dashed lines) correspond to these approximations while the thickest one corresponds to the exact absolute-value function $y(x) = |x|$. The thin uppermost curve represents the plot for $(\alpha = 4, \beta = 2.3)$ and the lowest curve for $(\alpha = 20, \beta = 2.3)$.

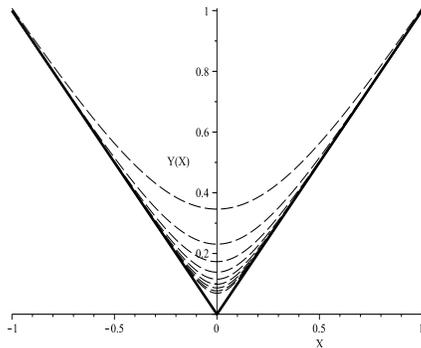


Fig. 1. Approximations for $\beta = 2.3$ and α -values, $\alpha = 4, 6, 8, \dots, 20$

3. Divide the function domain into an equally sized grid. For one-dimensional functions this partition is done by the set of vertical line equations $p_n(x) = x = i$, with $i = 1, 2, \dots, (N - 1)$. For two-dimensional functions the line equations $p_n(x, y)$ are constructed by following a simplicial subdivision over the plane XY which is composed of three types of traces: vertical ($p_n(x, y) = x = i$), horizontal ($p_n(x, y) = y = i$), and crosswise ($p_n(x, y) = x - y = i$). Particularly, for each of these traces the sweep of i must be in accordance with the boundaries existing along the X and Y axes. For example, if the restrictions $X = s_1$ and $Y = s_2$ are defined in the XY plane, the sweeps $i = \{1, 2, \dots, (s_1 - 1)\}$, $i = \{1, 2, \dots, (s_2 - 1)\}$, and $i = \{(-s_1 + 1), (-s_1 + 2), \dots, 0, 1, \dots, (s_2 - 1)\}$ will correspond to the horizontal, vertical, and crosswise traces, respectively.

4. Order the vertexes appropriately. This means that the vertexes must be ordered according to their class. For a two-dimensional domain $s_1 \times s_2$, the class-zero vertex is the point located at the origin of a coordinate system XY , the class-one vertexes are the coordinates $(X_i, 0)$ and $(0, Y_j)$ (for $i = 1, 2, \dots, s_1$ and $j = 1, 2, \dots, s_2$, respectively) and the set of class-two vertexes is composed by the coordinates (X_i, Y_j) , with the j -th index running as $j = 1, 2, \dots, s_2$ for each i -th value from 1 to s_1 . Likewise, for a one-dimensional domain X , bounded by a closed interval $[s_0, s_1]$, only the vertexes class-zero (the coordinate $(s_0, 0)$) and class-one (the coordinates $(i, 0)$ for $i = 1, 2, \dots, s_1$) exist.

5. Generate the symbolic basis-functions $\Lambda^k = \gamma^k(p_n)$, with the index k closely dependent on the vertical and horizontal partition line equations. For example, in a two-dimensional domain $s_1 \times s_2$ will be a set P_H of $(s_1 - 1)$ horizontal equations, and a set P_V of $(s_2 - 1)$ equations. If a cartesian product between these sets is formed, then each ordered pair whose first component is an equation member of P_H and whose second component is an equation member of P_V will be evaluated in the basis-function Λ^k with $k = 0, 1, \dots, (s_1 - 1) \times (s_2 - 1)$. The set of these functions is arranged in vectorial form as

$$\Lambda = [\Lambda^0, \Lambda^1, \dots, \Lambda^k]. \quad (6)$$

For a one-dimensional domain only vertical line partition equations will be presented and $k = 0, 1, 2, \dots, (s_1 - 1)$.

6. Evaluate the vector of basis-functions Λ at each ordered vertex. It is important to observe the similarity that exists between the methodology used to order the vertexes and which is used to construct the matrix Λ_V constituted by the rows of evaluated vectors Λ :

$$\Lambda_V = \begin{bmatrix} \Lambda^0(V^0) & \Lambda^1(V^0) & \dots & \Lambda^k(V^0) \\ \Lambda^0(V^1) & \Lambda^1(V^1) & \dots & \Lambda^k(V^1) \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda^0(V^n) & \Lambda^1(V^n) & \dots & \Lambda^k(V^n) \end{bmatrix}, \quad (7)$$

where n indicates the number of vertexes. For the one-dimensional case, the above procedure is practically the same with the only difference of evaluating two classes of vertexes (zero- and one-, type).

7. Provide the values of function B_i at the break-point coordinates by following a sorted order associated with the vertices V^0, V^1, \dots, V^n . These values are written as follows

$$\mathbf{B} = [B_0, B_1, B_2, \dots, B_n]^T. \quad (8)$$

This step is indistinctly implemented in the construction of one-dimensional and two-dimensional functions.

8. Obtain the interpolated function by $f(\cdot) = \Lambda_V^{-1}\mathbf{B}$. This step can be irrespectively applied to both cases: one-dimensional and two-dimensional one.

4 Simulation Results

In this section, three examples to illustrate the effectiveness of the proposed smoothing strategy are presented. We use the software Maple Release 15 to build the mathematical models and plot the approximate functions.

Example 1: Let the sequence of data points $D = \{(0, 0), (1, 2), (2, 1), (3, 3), (4, -1), (5, 5)\}$ be used to determine a smooth-piecewise function $y_s(x)$ in the range $[0, 5]$ of x . In order to have a comparative reference, the methodology of Pedro-Julian et al.[8] is used to obtain a piecewise-linear function that satisfies the condition of constructing new data points within the range of $[0, 5]$. As a result, the function $y(x)$ is obtained.

$$y(x) = \begin{bmatrix} 1 \\ -\frac{3}{2} \\ \frac{3}{2} \\ -3 \\ 5 \end{bmatrix}^T \begin{bmatrix} |x| \\ |x-1| \\ |x-2| \\ |x-3| \\ |x-4| \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ -\frac{3}{4} \\ \frac{3}{4} \\ -\frac{3}{2} \\ \frac{5}{2} \end{bmatrix}^T \begin{bmatrix} ||x| + x| \\ ||x-1| + x-1| \\ ||x-2| + x-2| \\ ||x-3| + x-3| \\ ||x-4| + x-4| \end{bmatrix}$$

$$+ \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{4} \\ -\frac{3}{4} \\ \frac{3}{2} \\ -\frac{5}{2} \end{bmatrix}^T \begin{bmatrix} |x - |x|| \\ |x-1 - |x-1|| \\ |x-2 - |x-2|| \\ |x-3 - |x-3|| \\ |x-4 - |x-4|| \end{bmatrix}. \quad (9)$$

In accordance with the construction methodology presented in the previous section, a smooth function $y_s(x)$ can be determined by replacing the γ basis-function by its logarithmic approximation reported in equation (6). With aid of Maple software, the resulting smooth expression can be simplified in a more compact algebraic form as

$$y_s(x) = \sum_{i=1}^{30} A_i \ln(1 + e^{B_i x + C_i + E_i P}) + K \quad (10)$$

with $P = \ln(1 + e^{G_i x + H_i}) + \ln(1 + e^{L_i x + M_i})$, where the function parameters for a constant $K = -8.54611$ are summarized in Table 1.

Table 1. Function parameters for the smooth-piecewise function $y_s(x)$

i	A_i	B_i	C_i	E_i	G_i	H_i	L_i	M_i
1, 2	± 0.10687	-10	+10	± 0.43429	-10	+10	+10	-10
3, 4	± 0.10687	+10	-10	∓ 0.43429	-10	+10	+10	-10
5, 6	± 0.13428	+10	0	± 0.43429	-10	0	+10	0
7, 8	± 0.13428	-10	0	∓ 0.43429	-10	0	+10	0
9, 10	± 0.11231	+10	-20	± 0.43429	-10	+20	+10	-20
11, 12	∓ 0.11231	-10	+20	± 0.43429	-10	+20	+10	-20
13, 14	± 0.20782	-10	+30	± 0.43429	-10	+30	+10	-30
15, 16	∓ 0.20782	+10	-30	± 0.43429	-10	+30	+10	-30
17, 18	± 0.30705	+10	-40	± 0.43429	-10	+40	+10	-40
19, 20	± 0.30705	-10	+40	∓ 0.43429	-10	+40	+10	-40
21, 22	+0.26858	∓ 10	0	0	0	0	0	0
23, 24	-0.41564	∓ 10	± 30	0	0	0	0	0
25, 26	+0.22462	∓ 10	± 20	0	0	0	0	0
27, 28	-0.21375	∓ 10	± 10	0	0	0	0	0
29, 30	+0.61411	∓ 10	± 40	0	0	0	0	0

In Figure 2 the dashed line shows the curve of the function $y(x)$ while the solid line depicts the curve for the smooth-piecewise function $y_s(x)$.

An error between the piecewise-linear function and its smooth version can be estimated by the percent error formulation $\epsilon = 100 \times \left| 1 - \frac{y_s(x)}{y(x)} \right|$. Figure 3 shows a curve of the percent error $\epsilon(x)$.

From this figure two important results must be underlined: the error is distributed along each linear segment and a precise fitting is achieved at the breakpoints locations. Moreover, an interesting property can be observed if the differentiability of

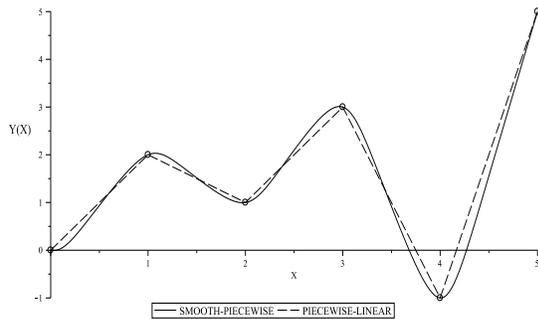


Fig. 2. Function approximation: piecewise-linear (equation (9), dashed line) and smooth-piecewise (equation (10), solid line)

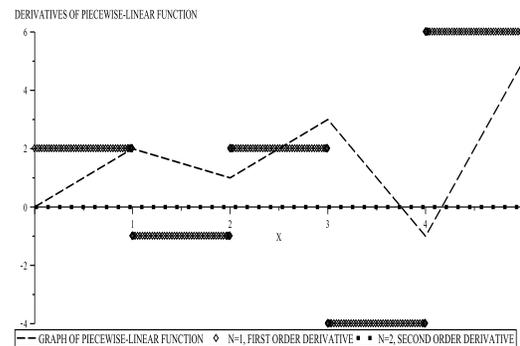


Fig. 4. First and second derivatives of $y(x)$

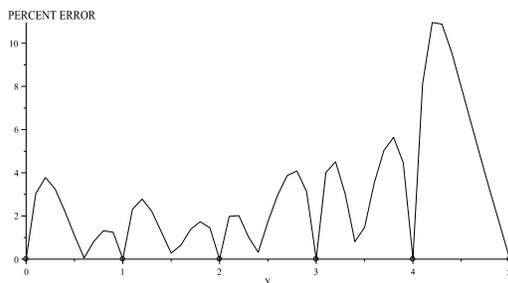


Fig. 3. Percent error between $y_s(x)$ and $y(x)$

functions $y(x)$ and $y_s(x)$ is put under testing. As can be noted in Figure 4, for the function $y(x)$ only the first order derivative is defined (except in the breakpoints) while the n -order derivative (for all $n \geq 1$) is zero or not defined.

In contrast, it must be noted that the existence of the higher order derivatives is guaranteed in the smooth function due to the fact that its second derivative is continuous everywhere.

As an example, Figure 5 shows curves for the first and second order derivatives of $y_s(x)$.

An important observation regarding to equation (10) is that it can be decomposed into two component functions which can be derived from the parameters reported in Table 1: one for $i = 1, \dots, 20$, denoted as $y_l(x)$, and another for $i = 21, \dots, 30$, expressed as $y_{nl}(x)$. This means that the smooth piecewise function can be expressed as $y_s(x) = y_l(x) + y_{nl}(x)$, with a nonlinear behavior of the characteristic curve of

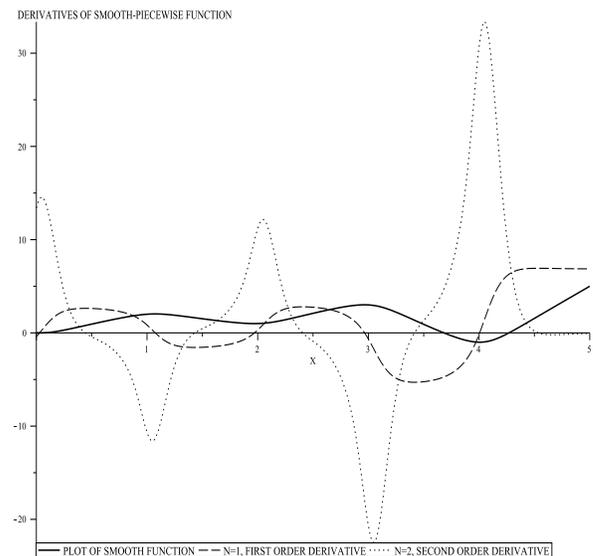


Fig. 5. First and second derivatives of $y_s(x)$

$y_s(x)$ mainly contained in $y_{nl}(x)$, and a quasilinear behavior strongly related to $y_l(x)$. This allows to simplify the smooth-piecewise function in a more compact expression by approximating $y_l(x)$ with a line equation $y_{le}(x)$ (for this example, $y_{le}(x) = 2.0666667x + 14.84611591$). As expected, a good approximation of $y_s(x)$ is obtained by $y_{nl}(x) + y_{le}(x)$. Considering this result, equation (10) can be expressed (also in reference to Table 1) in a

more compact form:

$$y_s(x) \approx \sum_{i=1}^{20} A_i \ln(1 + e^{B_i x + C_i}) + K + 2.0666667x + 14.84611591. \quad (11)$$

In Figure 6 graphs of $y_s(x)$, $y_l(x)$, $y_{nl}(x)$, and $y_{le}(x)$ functions are shown.

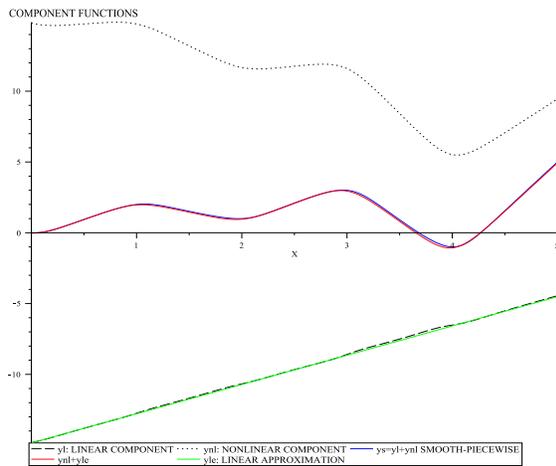


Fig. 6. Components of the smooth-piecewise function $y_s(x)$

Example 2: In this example it is assumed that the only input data that is available to construct an interpolation function $z(x, y)$ is the set of points D .

$$D = \{(0,0,1), (1,0,0), (2,0,2), (3,0,0), (0,1,0), (0,2,1), (3,3,0), (1,1,0), (1,2,0), (1,3,2), (2,1,1), (2,2,1), (2,3,0), (3,1,0), (3,2,0), (3,1,3)\}.$$

By following the traditional construction methodology of Pedro-Julian et al.[8], a two dimensional piecewise-linear function $z = f_{H_S[S]}(x, y)$ can be obtained. Such function is defined over a compact domain S equally partitioned by a simplicial boundary configuration H_S . In accordance with the methodology presented in the previous section, a smooth version of $f_{H_S[S]}(x, y)$ can be obtained if the same construction methodology is applied and the absolute-value basis-function is replaced by the exponential approximation of equation (6). The resulting smooth function has the property of removing the abrupt variation at the breakpoints

while preserving the waveform pattern. This fact ensures the differentiability of the function. In Figure 7 and Figure 8, surfaces for the piecewise-linear function $z(x, y) = f_{H_S[S]}(x, y)$ and its smooth version are shown.

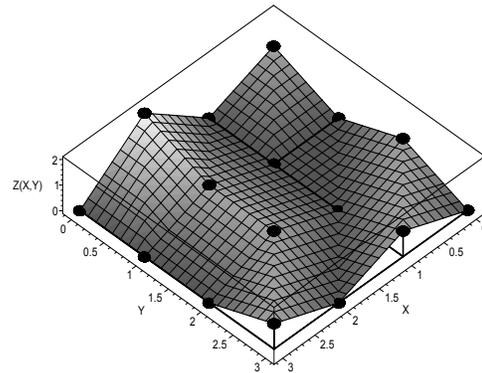


Fig. 7. Surface for the two-dimensional simplicial piecewise-linear function obtained from the input data D of Example 2

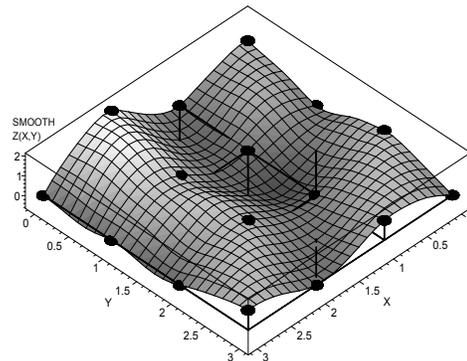


Fig. 8. Surface for the two-dimensional smooth-piecewise function obtained from the input data D of Example 2

Similarly to what was done in the one-dimensional example, a precise curve fitting was achieved at the breakpoint coordinates while the maximum deviation between the curves which are depicted in Figure 7 and Figure 8 is observed inside of each simplex in the XY plane. In Figure 9 this effect can be graphically appreciated. Additionally, it must be observed that in contrast with the original piecewise-linear function where the second and higher order derivatives are null,

in the smooth function this problem is overcome. In order to illustrate this numerical advantage, surfaces for the first and second derivative of $z(x, y) = f_{HS[S]}(x, y)$ are shown in Figure 10 and Figure 11, respectively.

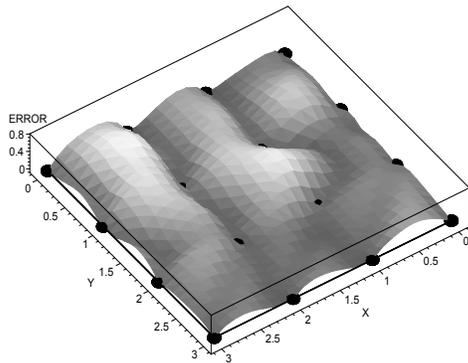


Fig. 9. Deviation between the piecewise-linear and the smooth curve obtained from the input data D of Example 2

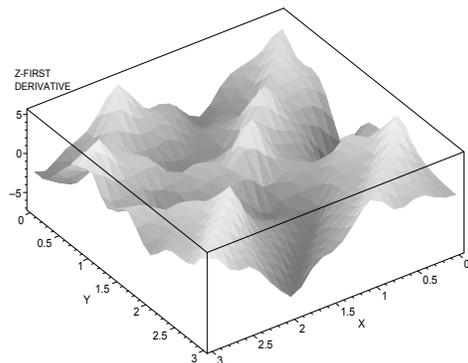


Fig. 10. Derivatives of function $z(x, y) = f_{HS[S]}(x, y)$. (a) First order derivative. (b) Second order derivative

Example 3: This example illustrates a potential application of the smoothing proposal in a typical problem of control systems engineering. The black curve of Figure 12 represents the transient response of a second-order system for a unit-impulse excitation. Consider that we are interested in determining the time t_p when the maximum peak of this response occurs. Let us consider that the analytical transfer function is not accessible and only a set of equally spaced breakpoints (discrete measurements) is the input data available

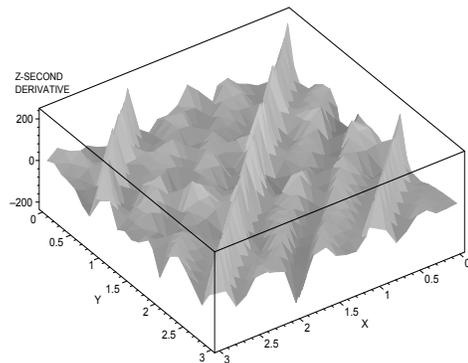


Fig. 11. Derivatives of function $z(x, y) = f_{HS[S]}(x, y)$: (a) First order derivative, (b) Second order derivative

to construct an approximate function. By following the construction strategy exposed in section 3, the two approximations depicted in Figure 12 can be obtained: piecewise-linear $c_{pwl}(t)$ (red) and smooth-piecewise $c_s(t)$ (blue).

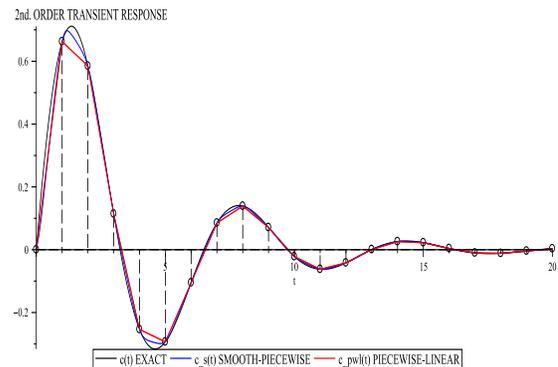


Fig. 12. Approximate functions for the transient response $c(t)$: smooth-piecewise $c_s(t)$, piecewise-linear $c_{pwl}(t)$

In Figure 12, the corresponding smooth and piecewise-linear functions are expressed in the form

$$c_{pwl}(t) = \sum_i^{\sigma} \hat{A}_i |t - \alpha_i| + \hat{B}_i |t - \beta_i| - |t - \beta_i| + \hat{G}_i ||t - \gamma_i| + t - \gamma_i| \quad (12)$$

and

$$c_s(t) = \sum_{i=1}^{\sigma} N_i \ln \left(1 + e^{((-1)^i 10t + (-1)^{(i+1)} 10k_i)} \right) + (2.4 \times 10^{-3})t - 0.104 \quad (13)$$

with the parameters summarized in Table 2 and Table 3, respectively.

Table 2. Function parameters for the piecewise-linear function $c_{pwl}(t)$

i	\hat{A}_i	$\hat{B}_i = -\hat{C}_i$	$\alpha_i = \beta_i = \gamma_i$
0	+0.33134	-0.16567	0
1	-0.37020	+0.18510	1
2	-0.19642	+0.98210 × 10 ⁻¹	2
3	+0.51145 × 10 ⁻¹	-0.25572 × 10 ⁻¹	3
4	+0.16428	-0.82140 × 10 ⁻¹	4
5	+0.11400	-0.57000 × 10 ⁻¹	5
6	+0.9960 × 10 ⁻³	-0.49800 × 10 ⁻³	6
7	-0.68265 × 10 ⁻¹	+0.34132 × 10 ⁻¹	7
8	-0.60870 × 10 ⁻¹	+0.30435 × 10 ⁻¹	8
9	-0.12327 × 10 ⁻¹	+0.61635 × 10 ⁻²	9
10	+0.26036 × 10 ⁻¹	-0.13018 × 10 ⁻¹	10
11	+0.30461 × 10 ⁻¹	-0.15230 × 10 ⁻¹	11
12	+0.11096 × 10 ⁻¹	-0.55482 × 10 ⁻²	12
13	-0.86790 × 10 ⁻²	+0.43395 × 10 ⁻²	13
14	-0.14392 × 10 ⁻¹	+0.71962 × 10 ⁻²	14
15	-0.74405 × 10 ⁻²	+0.37202 × 10 ⁻²	15
16	+0.21605 × 10 ⁻²	-0.10802 × 10 ⁻²	16
17	+0.64210 × 10 ⁻²	-0.32105 × 10 ⁻²	17
18	+0.43571 × 10 ⁻²	-0.21786 × 10 ⁻²	18
19	-0.47850 × 10 ⁻⁴	+0.23925 × 10 ⁻⁴	19

Table 3. Function parameters for the smoothing piecewise-linear function $c_s(t)$

i	N_i	k_i
1, 2	+0.039087	0, 0
3, 4	-0.044564	1, 1
5, 6	-0.019685	2, 2
7, 8	+0.0061166	3, 3
9, 10	+0.017621	4, 4
11, 12	+0.011828	5, 5
13, 14	-0.0002595	6, 6
15, 16	-0.0073524	7, 7
17, 18	-0.0064319	8, 8
19, 20	-0.0010948	9, 9
21, 22	+0.0028448	10, 10
23, 24	+0.0031831	11, 11
25, 26	+0.0011504	12, 12
27, 28	-0.00098792	13, 13
29, 30	-0.0015378	14, 14
31, 32	-0.00077512	15, 15
33, 34	+0.00027019	16, 16
35, 36	+0.00072258	17, 17
37, 38	+0.00039166	18, 18
39, 40	+0.000014116	19, 19

From this set of data, we may determine the peak time (t_p) by differentiating the approximate

function ($c_{pwl}(t)$ or $c_s(t)$) with respect to time (t) and setting this derivative equal to zero

$$\frac{d(c_{pwl}(t))}{dt} = 0 \quad (14)$$

or

$$\frac{d(c_s(t))}{dt} = 0. \quad (15)$$

However, for the piecewise-linear function a numerical solution cannot be reached, hence the smooth function $c_s(t)$ appears as the better approximation alternative. After solving equation (15) with Maple software we obtain $t_p = 1.193s$, which clearly corresponds to the peak time. The piecewise-linear $c_{pwl}(t)$ and the smooth-piecewise $c_s(t)$ approximations of the impulse transient response, as well as their derivatives with respect to time, are shown in Figure 13.

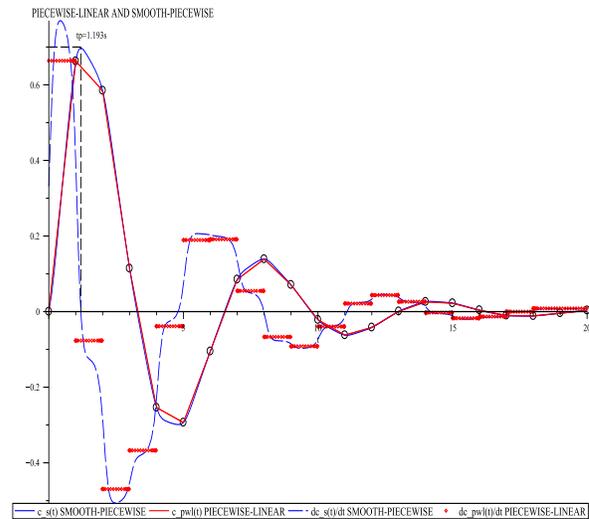


Fig. 13. Approximate functions and derivatives for the transient response: piecewise-linear and smooth-piecewise

In this figure, it is important to observe the notorious difference that exists, from the point of view of function continuity, between the derivatives of these two approximate functions: while $\frac{dc_s(t)}{dt}$ is completely connected at all points, $\frac{dc_{pwl}(t)}{dt}$ has discontinuities at the breakpoints (plotted as the red points).

5 Comparative Discussion

In this section, a comparative discussion about the curve fitting performance of the proposed method against other strategies such as polynomial and cubic spline techniques is presented. To start this analysis, it is important to mention that a better curve fitting can be achieved by the smooth-piecewise method, this is because it always imposes on the function the condition of passing through the input data coordinates. Similarly to the piecewise-linear functions, the continuity is also preserved in all the function domain but unlike the piecewise-linear reference, the smoothing proposal adds the function differentiation capability. In contrast, although in polynomial interpolation the function continuity is guaranteed, curve fitting is not very precise what results in a remarkable curve deviation that increases the derivative growing very quickly. An alternative approach to minimizing the curve fitting error is the spline interpolation which consists in restricting the approximation to low degree polynomials (typically, like in our comparative analysis, polynomials of third order degree are preferred) over partitioned sections along the function domain. However, in spite of showing a better curve fitting in comparison to the polynomial counterpart, the best performance is clearly obtained by the smooth-piecewise proposal. To emphasize this characteristic, the smooth-piecewise, polynomial, and spline approximate functions for $D_C = \{(0, 5), (1, 3), (2, 10), (3, 0), (4, 3), (5, 1), (6, 10)\}$ are shown in Figure 14.

The error in these approximations can be estimated by the deviation ϵ among the smooth-piecewise, polynomial, and splines curves with respect to the piecewise-linear reference. Such deviations are depicted in Figure 15.

6 Conclusion

In this paper, the proof-of-concept related to obtaining smooth-piecewise functions by replacing the basis-function used in the construction methodology of the High Level Canonical piecewise-linear model was demonstrated by numerical simulations. In accordance with such

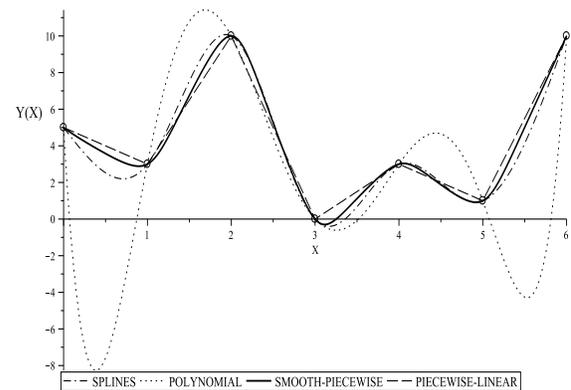


Fig. 14. Approximate functions for the input data D_C : piecewise-linear, smooth-piecewise, polynomial, and splines

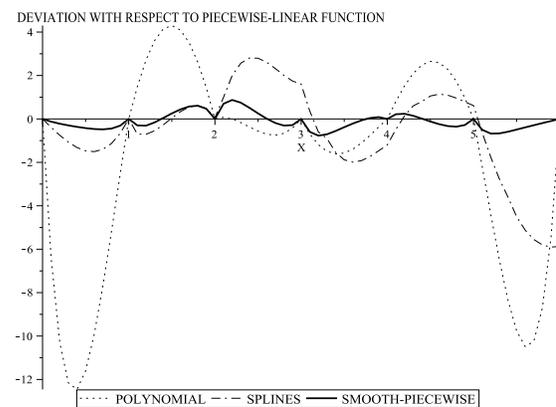


Fig. 15. Curve deviation with respect to the piecewise-linear reference: smooth-piecewise, polynomial, and splines

simulations, smooth-piecewise functions not only preserve curve fitting accuracy but also incorporate derivation capability. By illustrative examples, it was observed that the error between the original piecewise-linear curve and its smoothing version is uniformly distributed along each linear partition, being more pronounced approximately before and after its middle location and extremely reduced around the breakpoints. Moreover, two very important observations must be highlighted: the smooth function can be decomposed into two representations (one nonlinear and other

quasilinear), and the number of terms of the resulting smooth function can significantly be reduced due to the fact that a great number of them can be approximated by a line equation. Although its potential application to practical engineering problems was illustrated in Example 3, now our ongoing work centers on exploring alternative applications for this smoothing strategy.

References

1. **Chua, L. & Deng, A. (1986).** Canonical piecewise-linear modeling. *IEEE Transactions on Circuits and Systems*, Vol. 33, No. 5, pp. 511–525.
2. **Chua, L. & Deng, A. (1988).** Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 1, pp. 101–111.
3. **Chua, L. & Kang, S. (1977).** Section-wise piecewise-linear functions: Canonical representation, properties and applications. *Proceedings of the IEEE*, Vol. 65, No. 6, pp. 915–929.
4. **DAmbrosio, C., Lodi, A., & Martello, S. (2010).** Piecewise linear approximation of functions of two variables in milp models. *Operations Research Letters*, Vol. 38, No. 1, pp. 39–46.
5. **Eriksson, K., Estep, D., & Johnson, C. (2004).** Piecewise linear approximation. *Book chapter in: Applied Mathematics: Body and Soul*, Vol. 2, No. 1, pp. 741–753.
6. **Guzelis, G. & Goknar, I. (1991).** A canonical representation for piecewise-affine maps and its applications to circuit analysis. *IEEE Transactions on Circuits and Systems*, Vol. 38, No. 11, pp. 1342–1354.
7. **Hertling, J. (1970).** Approximation of piecewise continuous functions by a modification of piecewise hermite interpolation. *Numerische Mathematik*, Vol. 15, No. 5, pp. 404–414.
8. **Julian, P. (1998).** *A High Level Canonical Piecewise Representation: Theory and Applications*. Ph.D. thesis, Universidad Nacional del Sur, Bahía Blanca, Argentina.
9. **Julian, P. (2003).** The complete canonical piecewise-linear representation: Functional form for minimal degenerate intersections. *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, Vol. 50, No. 3, pp. 387–396.
10. **Julian, P., Damico, B., & Desages, A. (2000).** A model reduction procedure for high level canonical pwl functions. *Proceedings of the 2000 IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 705–708.
11. **Julian, P., Desages, A., & Agamennoni, O. (1998).** On the high level canonical representation of piecewise linear functions. *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, 1998. ISCAS '98.*, Vol. 6, pp. 314–317.
12. **Julian, P., Desages, A., & Agamennoni, O. (1999).** High-level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 46, No. 4, pp. 463–480.
13. **Kahlert, C. & Chua, L. (1990).** A generalized canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 3, pp. 373–383.
14. **Kahlert, C. & Chua, L. (1992).** The complete canonical piecewise-linear representation.i: The geometry of the domain space. *IEEE Transactions on Circuits and Systems*, Vol. 39, No. 3, pp. 222–236.
15. **Kang, S. & Chua, L. (1978).** A global representation of multidimensional piecewise-linear functions with linear partitions. *IEEE Transactions on Circuits and Systems*, Vol. 11, No. 25, pp. 938–940.
16. **Leenaerts, D. & Van Bokhoven, W. (1998).** *Piecewise linear modelling and analysis*. Springer US, Dordrecht, The Netherlands, 1 edition.
17. **Pottmann, H., Krasauskas, R., Hamann, B., Joy, K., & Seibold, W. (2000).** On piecewise linear approximation of quadratic functions. *Journal for Geometry and Graphics*, Vol. 4, No. 1, pp. 31–53.
18. **Schmidt, M., Fung, G., & Rosales, R. (2007).** Fast optimization methods for l1 regularization: A comparative study and two new approaches. *Book chapter in: Machine Learning*, Vol. 4701, pp. 286–297.

Victor Manuel Jimenez Fernandez received the B.Sc. degree in Electronic Engineering from the Technological Institute of Veracruz in 1998, the M.Sc. degree from the University of the Americas-Puebla (UDLA) in 2000, and the Ph.D. degree from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Puebla, Mexico, in 2006, both in Electronic Sciences. Since 2009 he has been Professor at the School of Electronic Instrumentation of the University of Veracruz (UV),

Mexico. His main research interest is application of nonlinear models in engineering.

Maribel Jimenez Fernandez received the B.Sc. degree in Pharmaceutical Chemistry and Biology from the University of Veracruz (UV) in 1997, the M.Sc. in Food Sciences from the University of the Americas-Puebla (UDLA) in 2000, and the Ph.D. degree in Food Biochemistry from the Technological Institute of Veracruz in 2006. She is currently with the Basic Sciences Institute (ICB) of the University of Veracruz. Her research interests include microencapsulation and microbiology in food technology.

Hector Vazquez-Leal received the B.Sc. degree in Electronic Instrumentation Engineering from the University of Veracruz (UV) in 1999, the M.Sc. and Ph.D. degrees in Electronic Sciences from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, in 2001 and 2005, respectively. His current research mainly covers analytical-numerical solutions and symbolic analysis of nonlinear problems. Since 2010 he has been Professor at the School of Electronic Instrumentation of the University of Veracruz (UV), Mexico.

Uriel Antonio Filobello Niño received the B.Sc. degree in Physics from the University of Veracruz (UV) in 1991, the M.Sc. and Ph.D. degrees in Physics from the National Autonomous University of Mexico (UNAM), Mexico, in 2002 and 2007, respectively. He is currently Professor at the School of Electronic Instrumentation of the University of Veracruz, Mexico. His main research interest covers analytical-numerical methods for ordinary differential equations.

Francisco Javier Castro-Gonzalez received his B.Sc. in Communication and Electronic Engineering from the University of Xalapa, Veracruz, Mexico, in 2004, and the M.Sc. and Ph.D. from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Puebla, Mexico, in 2009 and 2015, respectively. He currently works as a Postdoctoral Researcher at the University of Veracruz (UV). His research interests include device modeling and simulation of hybrid circuits (MOSFET-SETs) and computer-aided design and nonlinear circuit analysis.

Article received on 02/10/2015; accepted on 21/04/2016.
Corresponding author is Victor M. Jimenez-Fernandez.