# PWL Approximation of Non-linear Functions for the Implementation of Neuro-Fuzzy Systems

KOLDO BASTERRETXEA
Departamento de Electrónica y Telecomunicaciones
ESTHER ALONSO, JOSÉ MANUEL TARELA, INÉS DEL CAMPO
Departamento de Electricidad y Electrónica
University of the Basque Country (UPV/EHU)
Plaza La Casilla, no. 3, 48012 Bilbao, Bizkaia
SPAIN

*Abstract:* A piecewise linear (PWL) function approximation scheme is described by a lattice algebra of modified operators that allows for the interpolation of PWL function vertexes. A new recursive method called Centred Recursive Interpolation (CRI) based on such modified operators is analysed for successive function smoothing and more accurate approximation. This approximation method, simple but accurate as few parameters are needed for function definition, turns out to be a natural quadratic approximation. Due to the properties that neuro-fuzzy systems with gaussian-like non-linear functions show, CRI is applied to the approximation of a sample gaussian function with width control. Computational simplicity and parameter programmability have been central objectives. As PWL functions are generated by means of lattice operators in a recursive manner, this approximation method is particularly suitable for hardware implementation of function generation circuits.

*Key-Words:* neuro-fuzzy system, PWL function, lattice operator, recursive linear interpolation, membership function, gaussian function, function generation circuit.

## 1 Introduction

Fuzzy, neural and mixed (fuzzy/neural) systems, which we are going to refer to generically as neuro-fuzzy systems, are being used successfully in both scientific and engineering problem resolution. Traditionally, neuro-fuzzy computation has been made on general purpose processors, but in the last years many applications where high speed, low power consumption or portability must be assured, are demanding for more efficient implementations that require specific hardwired solutions. In the implementation of such structures there is a requirement for non-linear functions, membership functions or activation functions, that must be generated efficiently in terms of computation time and occupied silicon area.

Spline approximation, and more specifically piecewise linear (PWL) function approximation, is one of the most used computational schemes for non-linear function generation circuits [1-5]. PWL functions are adequate for the use of max ($\vee$) and min ($\wedge$) operators that, due to their algebraic characteristics, can be named as lattice operators [6].

In this paper a new recursive method for function approximation based on the use of *modified lattice operators* (MLO) is proposed. Such operators are the lattice operators $\vee$ and $\wedge$ with interpolation capabilities for function smoothing. The aim is to use this general framework in future implementation of membership function circuits (MFC) for fuzzy systems and in artificial neural network (ANN) nodes, specifically in radial basis function networks (RBFN). The application of this method in these fields must lead to the simplification and speeding up of the computation or generation of the mentioned functions when high precision, high speed and low area implementations are needed, overcoming the actual limitations in these areas.

## 2 Modified lattice operators with recursive interpolation

We call modified lattice operators (MLO), *max* and *min* operators with interpolation capability for PWL function vertex smoothing. *Max* modified operator can be defined as follows,

$$y_0 = y_1 \vee y_2 \vee h(y_1, y_2, \Delta) \qquad (1)$$

where *h* is the *interpolation function*. For a linear interpolation, *h* is

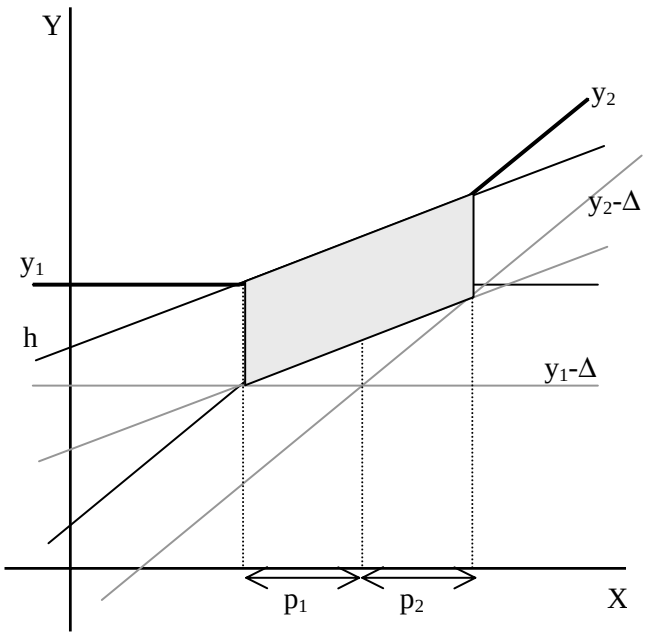



**Fig.1** Max modified lattice operator. Centred Linear Interpolation of the vertex formed by lines $y_1$ and $y_2$.

$$h(y_1, y_2, \Delta) = \alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 \Delta \qquad (2)$$

Although (2) can be used in other circumstances, we will restrict ourselves to the case where $\alpha_1 = \alpha_2 = 1/2$ which will be called Centred Lineal Interpolation (CLI). Then,

$$h(y_1, y_2, \Delta) = 1/2(y_1 + y_2) + \lambda\Delta \qquad (3)$$

where the unique parameter is $\lambda = \alpha_3$. The *interpolation factor* $\lambda$ will be, for simplicity, made equal to 1/2. With this assumption, and in the absence of *interpolation depth* ($\Delta = 0$), we force the interpolation function to go through the vertex generated by the intersection of the lines $y_1$ and $y_2$, point *C* in Fig.1. The most conspicuous characteristics of the lineal interpolation function for *y(x)* affine inputs are:

*i) Centred*: Equation (1) can be written as

$$y_0 = [(y_1 - \Delta) \vee (y_2 - \Delta) \vee (h - \Delta)] + \Delta \qquad (4)$$

As shown in Fig.1, the segments

$$h - \Delta = (1/2)(y_1 + y_2) - (1/2)\Delta \qquad (5)$$

and

$$h = (1/2)(y_1 + y_2 + \Delta) \qquad (6)$$

included in the interpolation region constitute two opposed sides of a parallelogram centred in *C*. Such structure is completed by the segments of height $\Delta$ and therefore $p_1 = p_2 = p$. This property is due to the election of $\alpha_1 = \alpha_2$. If $\alpha_1 \neq \alpha_2$ is chosen, then conditions for asymmetric bevels, which will not be considered here, are obtained.

*ii) Depth and projection of the bevel*: The depth of the bevel is $\lambda\Delta$; $\Delta/2$ if $\lambda=1/2$. The *projection* of the bevel is $p_1 + p_2 = 2p$. This region defines a local or short-range interaction between $y_1$ and $y_2$.

*iii) Associative property verification*: We can define the linear interpolation under study as

$$y_0\big|_{l\text{in interp}(\Delta)} = max_\Delta(y_1, y_2) = \vee_\Delta(y_1, y_2) \qquad (7)$$

being $\vee_\Delta$ the above mentioned *modified max operator*. Let $y_j(x)$, $j=1,2,3$ be three affine non-concurrent functions such that $y_2$ forms vertexes to be interpolated with $y_1$ and $y_3$, and $2p_{12}$, $2p_{23}$ the respective projections of these interpolated vertexes. We say that $\Delta$ is *sufficiently small* if

$$x_{12} + p_{12} \leq x_{23} - p_{23} \qquad (8)$$

where $y_1(x_{12}) = y_2(x_{12})$ and $y_2(x_{23}) = y_3(x_{23})$

In other words, the intersection of the interpolation functions does not take place above the straight lines that generate it. If this restriction is satisfied it turns out that

$$\vee_\Delta [\vee_\Delta (y_1, y_2), y_3] = \vee_\Delta [y_1, \vee_\Delta (y_2, y_3)] \qquad (9)$$

The observance of this property is substantial in lattice operations with various operands. We can define, under the same condition

$$\vee_\Delta (y_1, y_2, y_3) \equiv y_1 \vee y_2 \vee y_3 \vee (1/2)(y_1 + y_2 + \Delta) \vee (1/2)(y_2 + y_3 + \Delta) \qquad (10)$$

that can be generalised for more inputs. Expressions (7) and (8) are functionally equivalent under the restriction of a sufficiently small $\Delta$.

*iv) Affine n-dimensional inputs*: The above mentioned characteristics and properties are still valid for affine $y_j(\boldsymbol{x})$ inputs where $\boldsymbol{x}=(x_1,x_2,...,x_n)^T$. Particularly for each $x_j$ variable, considering $x_1,...,x_{j-1},x_{j+1},...,x_n$ as constants, these properties are verified.

All the above can be applied to a minimum selector with interpolation *or modified minimum operator* ($\wedge_\Delta$)

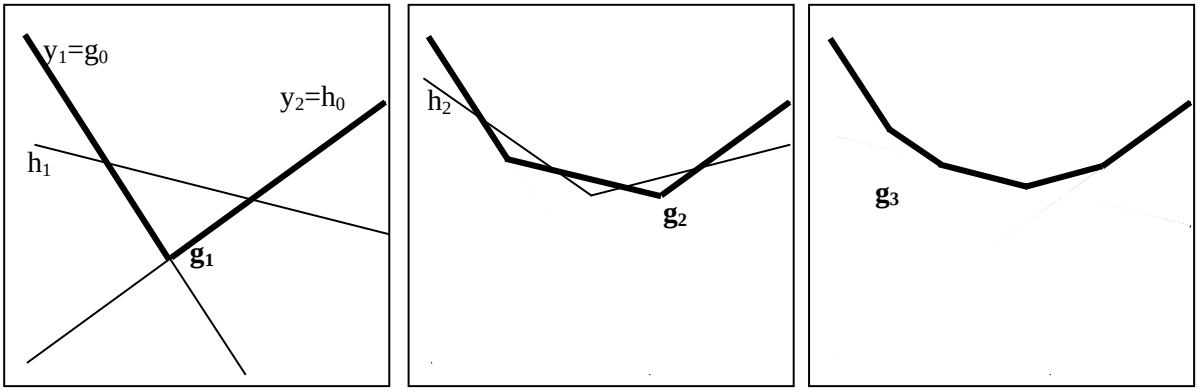



**Fig. 2** Recursive interpolation and dichotomic growth of the number of vertexes. (a) Maximum and first interpolation function $h_1$. Interpolation level: $q$=0; (b) $q$=1; (c) $q$=2.

if $\vee$ by $\wedge$ is substituted in (1) and we force $\alpha_3 = \lambda < 0$ in (2) to obtain the CLI.

## 2.1 Centred Recursive Interpolation (CRI)

The interpolation function described above can be generated recursively, allowing a progressive function smoothing
and enhancing accuracy of approximation through a repetitive computational scheme. Therefore, a complex and consecutively more precise approximations can be achieved by means of a simple computational cell, as we will see immediately.

Being $h_1$ the function $h(y_1,y_2,\Delta)$ for $\lambda=\lambda_0$, expression (1) can be rewritten as

$$max_{\Delta,1}(y_1,y_2) = (y_1 \vee y_2) \vee h_1 = g_1 \vee h_1 = g_2 \quad (11)$$

This interpolation, hereafter level 1 interpolation ($q$=1), is shown in Fig.2 (b). The output after applying a second level interpolation ($q$=2) to the second operand in (11) is shown in Fig.2 (c):

$$max_{\Delta,2}(y_1,y_2) = g_1 \vee h_1 \vee h_2 = g_2 \vee h_2 = g_3 \quad (12)$$

$$h_2(g_1,h_1,\Delta) = 1/2(g_1 + h_1) + \lambda_1\Delta \quad \text{where} \quad (\lambda_1 < \lambda_0) \quad (13)$$

This way, applying consecutive interpolation levels, a recursive interpolation scheme is obtained as shown in Fig.3. This scheme doubles the number of vertexes of the polygonal for each epoch or interpolation level, as depicted in Fig.2:

$$g_{q+1} = g_q \vee h_q \quad (14)$$

$$h_{q+1} = (1/2)(g_q + h_q) + \lambda_q\Delta \quad (15)$$

where $g_0=y_1$, $h_0=y_2$ and $\lambda_0=1/2$.

As if $0 > \lambda_q > (\lambda_{q-1}/2)$ the dichotomic growth of the number of vertexes in the function is impeded, we can choose the mean value for $\lambda_q$,

$$\lambda_0 = 1/2 \;,\; \lambda_q = (1/4)\lambda_{q-1} \Rightarrow \Rightarrow \lambda_q = \frac{1}{2^{2q+1}} \;,\; q = 1,2,... \quad (16)$$

Expression (16) together with (14) and (15) define the *Centred Recursive Interpolation* (CRI). All the information about the interpolation depth has been transferred to $\Delta$. As a result of this, the main parameters for the IRC are the couple $\{\Delta,q\}$. All products in this recursive scheme are powers of two, what remarkably would simplify a digital design.

The extension of this procedure for minimum selectors or intersection with recursive interpolation is straightforward.

## 2.2 CRI Approximation of $y=x^2$

CRI has been applied successfully to the approximation of various sample functions. The approximation of the function $y(x)=x^2$ through RCI is

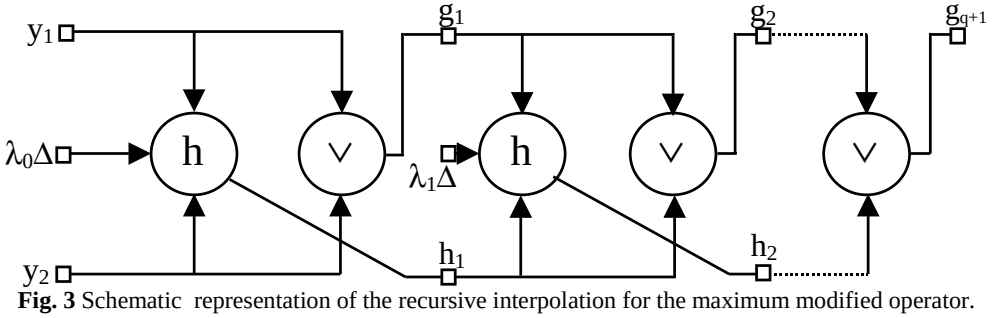



**Fig. 3** Schematic representation of the recursive interpolation for the maximum modified operator.

of particular interest, and is analysed with detail afterwards.

Let the initial affine functions be

$$g_0 = y_1 = 0 \tag{17}$$

$$h_0 = y_2 = 2x - 1 \tag{18}$$

in D=[0,1]. Notice that if $\varepsilon_q(x) \equiv x^2 - g_q$, then $\varepsilon_1(0) = \varepsilon_1(1) = 0$. Applying CRI:

$$g_{q+1} = g_q \vee h_q \tag{19}$$

$$h_{q+1} = \frac{1}{2}\left(g_q + h_q\right) + \frac{\Delta}{2^{2q+1}} = \frac{1}{2}\left(g_q + h_q + \frac{\Delta}{4^q}\right) \tag{20}$$

Let $\delta$ be an increment in $x$. Regard that:

*i)* $\forall x_j, \delta$, the tangents to $y=x^2$ at $x_j-\delta$ and $x_j+\delta$ intersect at $x=x_j$ and at this point $\varepsilon_q(x_j) = x_j^2 - g_q = \delta^2$. Therefore, if $\Delta=1/2$, D results successively divided in $2^0, 2^1, \ldots, 2^q$ sections at whose ends the error is null, as it is shown in Fig.4.

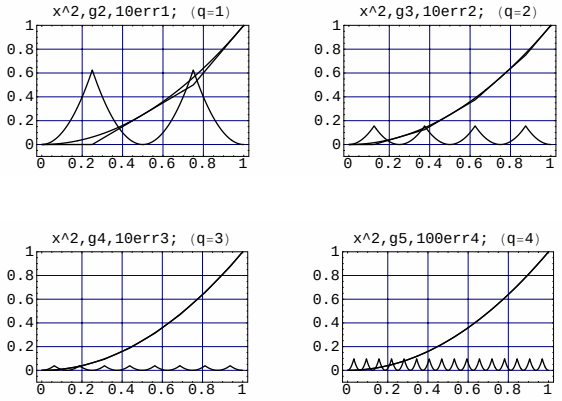



**Fig. 4** CRI approximation of $x^2$ with $\Delta=1/2$ and equidistant cancellation of the error.

*ii)* If $x$ is a finite length binary word $0.b_1b_2...b_n$, n iterations are enough to cancel the error. Moreover, if $q<n$ it is always possible to obtain a optimum value of $\Delta$ ($\Delta_{opt}>1/2$) which minimises the error.

*iii)* If $x$ is real, infinite iterations are needed, but as the interpolation level increases $\Delta_{opt} \to 1/2$ and the error tends to zero.

Fig.5 shows the generated functions at the three first interpolation levels with $\Delta_{opt}$. The optimum value of $\Delta$ has been obtained solving the optimisation problem

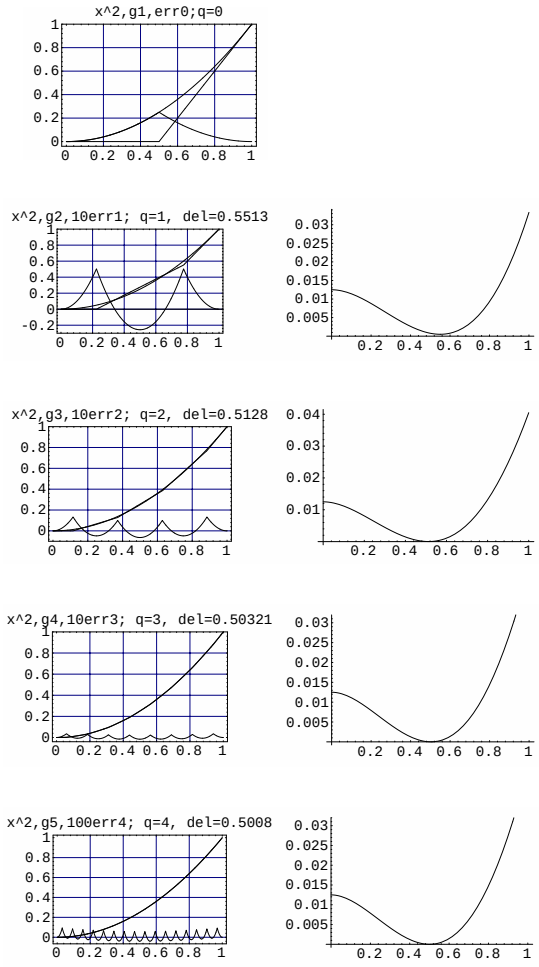
$$\min\{E(\mathbf{P})\} \tag{21}$$

$$E = \int_x \varepsilon^2(x,\Delta)dx \tag{22}$$

where

$$\varepsilon = f(x) - g(x,\Delta,q) \tag{23}$$

and $\mathbf{P}=[\Delta]$ is a one-dimensional parameter vector, as it can be asserted that the approximation improves as $q$ increases. In Fig.5 it can be observed that the error is distributed homogeneously through D. This fact evidences that the RCI method is a *natural quadratic approximation*.




(a) (b)

**Fig. 5** (a) CRI approximation of $x^2$ with the optimum value of the depth parameter (del≡Δ). (b) E(Δ), see (22) and (23), for the first four interpolation levels: q=0, 1, 2, 3, 4.

# 3 Approximation of a Gaussian Function with Width Control

Besides in many other applications, gaussian functions are widely used in the resolution of approximation problems, and specifically in the application of neuro-fuzzy systems as functional approximators. It is well

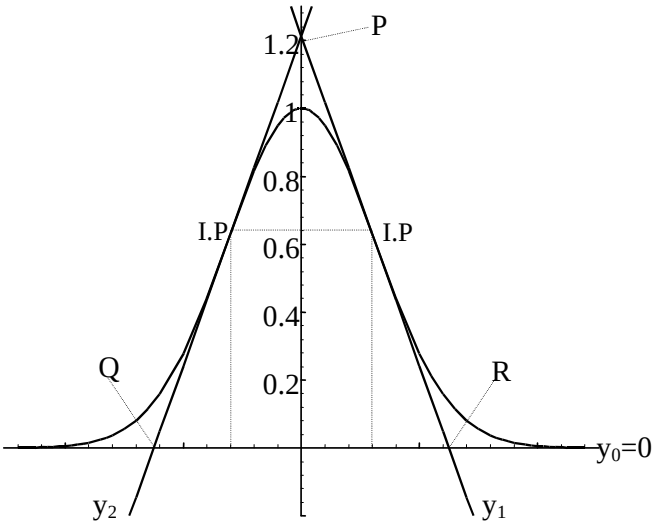



**Fig. 6** Reference gaussian function and initial PWL structure with three vertexes: *P, Q* and *R*

known that ANN's with one hidden layer of gaussian nodes are capable of approximating continuous functions with arbitrary precision [7,8]. RBFN's, which have the property of universal approximation [7-10], typically use gaussian functions as radial basis functions. An additive fuzzy system with gaussian sets reduces to a RBFN with gaussian nodes [11]. Moreover, it has been demonstrated that a fuzzy inference system with gaussian scalable membership functions (MF) and operating with product inference and implication, and with height defuzzyfication is a universal approximator [12]. In [13] similar results have been obtained for an additive system with center of mass defuzzyfication and product implication.

In conclusion, generating gaussian functions with different variances in a simple and fast way, would endow neuro-fuzzy systems with notable versatility and range of applications. This fact motivates the selection of this widely used function shape as a representative example of the applicability of RCI method in neuro-fuzzy computing structures. Originally this study is not biased by any specific design option (software, digital or analog hardware), although an VLSI implementation has been borne in mind as it would be favored by some characteristics of the method.

The gaussian function taken as reference is a function centered in the origin point, given by the expression:

$$f(x) = \frac{1}{\exp\left(x^2/\sigma^2\right)} \tag{24}$$

Next objectives has been defined as imperative:

*Objective 1. Computational simplicity:* The design must be simple in a computational sense, given the cost/speed implications (area/speed in the case of the VLSI). In this sense, the use of a recursive interpolation method (CRI) fundamentally based on a lattice structure, provides a very appropriate framework, especially interesting for physical circuit design.

*Objective 2. Normality:* Normality, as defined in fuzzy set theory, is a requirement of the fuzzy problem if the logical-linguistic nature of the system is to be kept.

*Objective 3. Programmability:* The generation scheme must allow for the modification of the parameters that define the width and the centre of the gaussian function.

We will start from a simple structure such as that shown in Fig.6, formed by the two tangents to the curve in its points of inflection $y_1(x)$ and $y_2(x)$, and the abscissa axis $y=0$, where

$$y_1(x) = -\left(\frac{1}{\sigma}\sqrt{\frac{2}{e}}\right)x + \frac{2}{\sqrt{e}} \Rightarrow y_1(x) = -m\,x + \frac{2}{\sqrt{e}} \tag{25}$$

$$y_2(x) = \left(\frac{1}{\sigma}\sqrt{\frac{2}{e}}\right)x + \frac{2}{\sqrt{e}} \Rightarrow y_1(x) = m\,x + \frac{2}{\sqrt{e}} \tag{26}$$

One sole parameter $m=m(\sigma)$ characterises both tangents, and this is the parameter which controls the width of the function. The centre of the gaussian function, for computational purposes, can be easily established in any point $c_x$ of the input domain by a simple modification of equations (23) and (24):

$$y_1'(x) = y_1(x) + mc_x \tag{27}$$

$$y_2'(x) = y_2(x) - mc_x \tag{28}$$

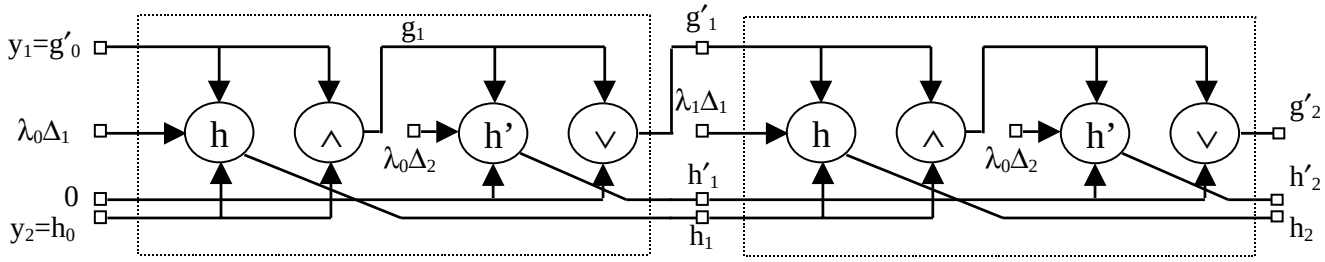



**Fig.7** CRI scheme for the initial structure shown in Fig. 6

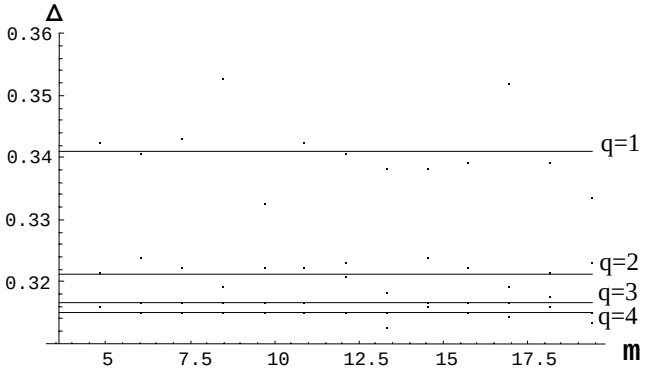



**Fig. 8** Optimum values of the depth parameter ($\Delta_{opt}$) for 13 values of the parameter *m* in the interval given by (30), and null slope linear approximation (mean value) for the first four recursion levels.

What we want to know now is how the CRI parameter Δ must vary to achieve an optimum approximation to the gaussian function for different widths. That is, we want to identify $\Delta_{opt}(m)$. To establish the working range we will take the domain D=[0,1] as our universe of discourse, and suppose a partition of 8 gaussian functions of equal width for the aforementioned universe. The width of each function in its points of inflexion is $(2\sigma/\sqrt{2})$. Therefore, the *mean value* $m_m$ of $m(\sigma)$ obtained for the mentioned partition is:

$$2\frac{\sigma_m}{\sqrt{2}}=\frac{1}{8}\rightarrow\sigma_m=\frac{\sqrt{2}}{16}\rightarrow m_m=\frac{1}{\sigma_m}\sqrt{\frac{2}{e}}=\frac{16}{\sqrt{e}} \quad (29)$$

The working range has been chosen as that comprised between one half of and twice the central value of *m*,

$$\sigma\in\left[\frac{\sqrt{2}}{32},\frac{\sqrt{2}}{8}\right]\rightarrow m\in\left[\frac{8}{\sqrt{e}},\frac{32}{\sqrt{e}}\right] \quad (30)$$

The recursive interpolation scheme is shown in Fig.7, where $g_i$ represents the function interpolated in vertex *P* for *q* = *i*, $g_i'$ is the function interpolated in the three vertexes, *h* is the interpolation function for vertex *P* and *h'* is de interpolation function for vertexes *Q* and *R*.

The optimisation problem defined in (21) and (22) has been solved numerically for the same value of Δ in the three vertexes *P*, *Q* and *R* and for 13 values of *m* in the range defined by (30). The values of $\Delta_{opt}$ are presented in Fig.8 for the first four interpolation levels. This optimum value, although with certain random deviations, is constant (mean value) for all *m*, allowing us to store one sole parameter $\Delta_1$ for all widths.

Nevertheless, $\Delta_{opt}$ does not assure the normality of the function. As CRI method assures that the height of the interpolated function is fixed in the first interpolation level, the solution to the optimisation problem with normality constraint is trivial, resulting in:

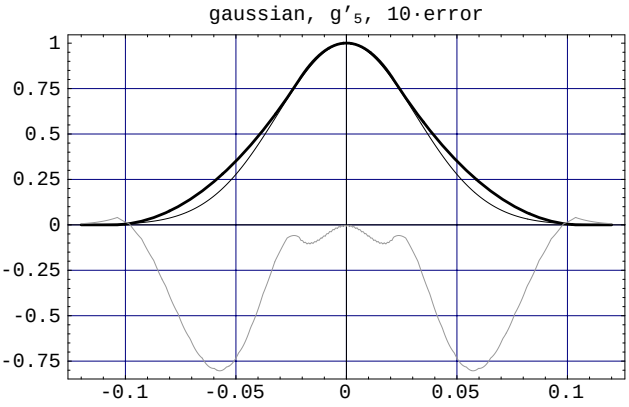



**Fig. 9** Approximation of the gaussian function with one sole depth parameter $\Delta_1 = \Delta_{nor}$ in order to ensure normality.

$$\Delta_{nor}=2\left(\frac{2}{\sqrt{e}}-1\right)\approx 0.42612 \quad (31)$$

This value is not optimum from error minimisation point of view, but it ensures function normality. If $\Delta_{nor}$ is used in the interpolation of the three vertexes, the resulting function for *q*=4 is shown in Fig.9, where it can be observed that error is negligible in the surroundings of the maximum but is drifted towards the "tails" of the function.

Obviously, $\Delta_{opt}(P)\neq\Delta_{opt}(Q)=\Delta_{opt}(R)$. Fig.10 shows the approximation for *q=4* with $\Delta(P)=\Delta_1=\Delta_{nor}$ and $\Delta(Q)=\Delta(R)=\Delta_2=\Delta_{opt}$ for four different widths. The error in the maximum of the function is null and less than 2% in the worst approximated point. This option is more accurate but an extra depth parameter $\Delta_2$ must be stored.

# 4 Conclusion

The function approximation method proposed in this work is based on the lattice structure for PWL function definition given in [6]. It consists of a recursive algorithm that operates with modified lattice operators, that is, maximum and minimum operators with linear interpolation for function smoothing. In the present work we have chosen and developed a centred symmetric interpolation procedure for its simplicity and easiness of computation. Nevertheless, asymmetric options are also possible. CRI shows itself to be a simple and efficient method for approximating functions, accurate even with minimum initial structures and relatively low recursive levels. It must be highlighted that CRI is a natural quadratic approximation scheme.

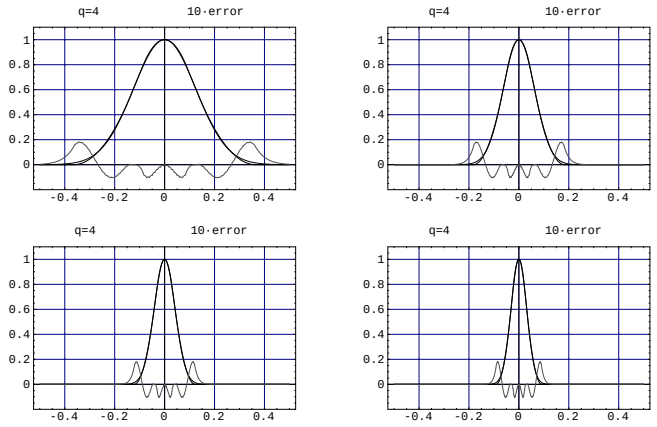



**Fig.10** Gaussian function, CRI approximation and error curve magnified by ten for q=4 using two depth parameters: $\Delta_1=\Delta_{nor}$ and $\Delta_2=\Delta_{opt.}$ The approximation is shown for four different widths: m=8/√e, m=16/√e, m=24/√e and m=32/√e.

The application of the method to membership function generation or activation functions represents an interesting alternative to actual circuit designs for neuro-fuzzy structures to overcome their performance limitations. The use of lattice operators and the recursive nature of the interpolation algorithm simplify remarkably the function computation. In consequence, this is a suitable method for high speed/low area hardware implementation of neuro-fuzzy circuitry.

Due to the properties as universal approximators that certain neuro-fuzzy systems with gaussian-like functions show, CRI has been applied to the generation of this type of functions. The programmability of the function has been a central objective of the design, as it is of a key relevance for neuro-fuzzy systems with learning capabilities or operating in changeable environments. Therefore, this work can be seen as a first step theoretical study previous to a future implementation of neuro-fuzzy electronic circuits based on this approximation technique.

*References:*

[1] D.J. Myers and R.A. Hutchinson, Efficient Implementation of Piecewise Linear Activation Function for Digital VLSI Neural Networks, *Electronics Letters*, Vol.5, No.24, 1989,pp.1662-1663.

[2] M. Zhang, S. Vassiliadis and J.G. Delgado-Frias, Sigmoid Generators for Neural Computing Using Piecewise Approximations, *IEEE Trans. on Computers*, Vol.45, No.9, 1996, pp.1045-1049.

[3] G. Ascia, V. Catania, G. Ficili, S. Palazzo and D. Panno, A VLSI Fuzzy Expert System for Real-time Traffic Control in ATM Networks, *IEEE Trans. on Fuzzy Systems*, Vol.5, No.1, 1997, pp.20-31.

[4] H. Eichfeld, T. Künemund, M. Menke, A 12b General-purpose Fuzzy Logic Controller Chip, *IEEE Trans. on Fuzzy Systems*, Vol.4, No.4, 1996, pp.460-475.

[5] S.K. Halgamuge, T. Hollstein, A. Kirschbaum and M. Glesner, Automatic Generation of Application Specific Fuzzy Controllers for Rapid-Prototyping, *IEEE Int. Conf. On Fuzzy Systems*, Orlando, FL, 1994, pp.1638-1641.

[6] J.M. Tarela, E. Alonso and M.V. Martinez, A Representation Method for PWL Functions Oriented to Parallel Processing, *Mathl. Computer Modelling*, Vol.13, No.10, 1990, pp.5-83.

[7] Girosi and T. Poggio, Networks and the Best Approximation Property, Centre for Biological Information Processing, Romm, 43-787, AI Lab MIT, Cambridge, MA 02139, USA, 1989, pp.257-264.

[8] E.J. Hartman and J.D. Keeler, Layered Neural Networks with Gaussian Hidden Units as Universal Approximation, *Neural Computation*, No.2, 1990, pp.210-215.

[9] J. Park and I.W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, *Neural Computation*, No.3, 1991, pp.246-257.

[10] J. Park and I.W. Sandberg, Approximation and Radial-Basis-Function Networks, *Neural Computation*, No.5, 1993, pp.305-316, 1993.

[11] J.S.R. Jang and C.T. Sun, Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems, *IEEE Trans. on Neural Networks*, Vol.4, No.1, 1993, pp.156-159.

[12] L.X.Wang, Fuzzy Systems are Universal Approximators, *Proc. IEEE Int. Conf. on Fuzzy Systems, San Diego, CA*, 1992, pp.1163-1169.

[13] B. Kosko, Fuzzy Systems as Universal Approximators, *IEEE Trans. on Computers*, Vol.43, No.11, 1994, pp.1329-1333.