Learning in Neuro/Fuzzy Analog Chips

Angel Rodríguez-Vázquez and Fernando Vidal-Verdú

*Dept. of Analog Design, CNM, Edificio CICA, C/ Tarfia sn, 41012-Sevilla, SPAIN

FAX #34 5 4624506, email: angel@cnm.us.es

**Dept. de Arquitectura y Tecnología de Computadores y Electrónica.

Universidad de Málaga, Plaza El Ejido sn, 29013-Málaga, SPAIN

ABSTRACT

This paper focus on the design of adaptive mixed-signal fuzzy chips. These chips have parallel architecture and feature electrically-controlable surface maps. The design methodology is based on the use of composite transistors -- modular and well suited for design automation. This methodology is supported by dedicated, hardware-compatible learning algorithms that combine weight-perturbation and outstar.

INTRODUCTION

During recent years fuzzy inference has been successfully applied mostly to control problems in vehicles, robots, motors, power systems, home appliances, etc., and to decision-making systems and image processing, among others [1]. In many of these systems, fuzzy inference can be realized by *software* on conventional microprocessors, to attain up to 1Kflip inference speed with 8 to 16 bits of resolution. However, those systems requiring high-speed, reduced power consumption, or smaller dimensions have prompted the development of dedicated *hardware* [2], in particular the use of mixed-signal VLSI chips [3].

There are two major classes of analog fuzzy chips: *fixed* function and *adaptive*. The former are better suited for applications where the input-output function is already completely defined at the chip design phase, and does not change with operation. However this is not the situation in most practical cases, where the exact function is unknown a priori or must adapt to specific environmental characteristics [4]. Consequently, the necessity arises to combine the inference capabilities of fuzzy systems with the *learning* capabilities of neural networks, as already discussed by different authors [5]. Based on these developments, this paper presents a neuro/fuzzy analog chip architecture, circuit blocks for its realization in VLSI CMOS technology, and hardware-oriented algorithms to adapt its parameters through learning. Major emphasis is placed on the *modularity* of the circuits used for adaptability, so that the proposed design methodology is applicable to both fixed and adaptive function chips.

CHIP ARCHITECTURE

The proposed neuro/fuzzy chips are implementations of Takagi-Sugeno's *singleton* inference rules [8]. This obtains the output as a weighted linear combination of fuzzy *basis* functions,

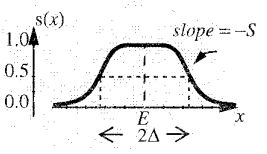
$$y = f(\mathbf{x}) = \sum_{i=1..N} y_i^* w_i^*(\mathbf{x}) \qquad (1)$$

where $\mathbf{x} = \{x_1, x_2 ..., x_M\}^T$ is the input vector, each $w_i^*(\mathbf{x})$ corresponds to a rule, and y_i^* is the singleton associated to it. The-basis functions are calculated from the input as,

$$w_i^*(\mathbf{x}) = \frac{min\{s_{i1}(x_1), s_{i2}(x_2), ..., s_{iM}(x_M)\}}{\sum_{i=1..N} min\{s_{i1}(x_1), s_{i2}(x_2), ..., s_{iM}(x_M)\}} \qquad (2)$$

where $min\{\bullet\}$ is the multidimensional minimum and $s_{ij}(x_j)$ are *membership* functions which codify the degrees of matching between each input and its fuzzy labels.

Fig. 1 shows an architecture for the realization of (1) and CMOS schematics for the different operators involved, in a case where membership functions are bell-like,



The rectangles in Fig. 1 represent composite transistors whose transconductance is controlled through a voltage, as required to incorporate programmability. The inputs to the circuits used for membership functions are voltages, while their outputs are currents. All the remaining circuits operate in current domain.

For a given structure determined by the number of membership functions and rules, the transfer function of Fig.1 is parameterized by the vector of singletons $\mathbf{y}^* = \{y_1^*, y_2^* ..., y_N^*\}^T$ and the vectors of centers $\mathbf{E}_i = \{E_{i1}, E_{i2} ..., E_{iM}\}^T$, widths $\Delta_i = \{\Delta_{i1}, \Delta_{i2} ..., \Delta_{iM}\}^T$, and slopes $\mathbf{S}_i = \{S_{i1}, S_{i2} ..., S_{iM}\}^T$ of the membership functions (see the inset in Fig.1 for the shape). For fixed chip applications, these parameters are calculated off-chip and the circuits sized accordingly. For applications that require adaptability, the circuits used inlayers 1 (Fig. 1(b)) and 4 (Fig. 1(c)) must be programmable and the chips made to learn the required transfer function in situ.

0-7803-2570-2/95 $4.00 ©1995 IEEE

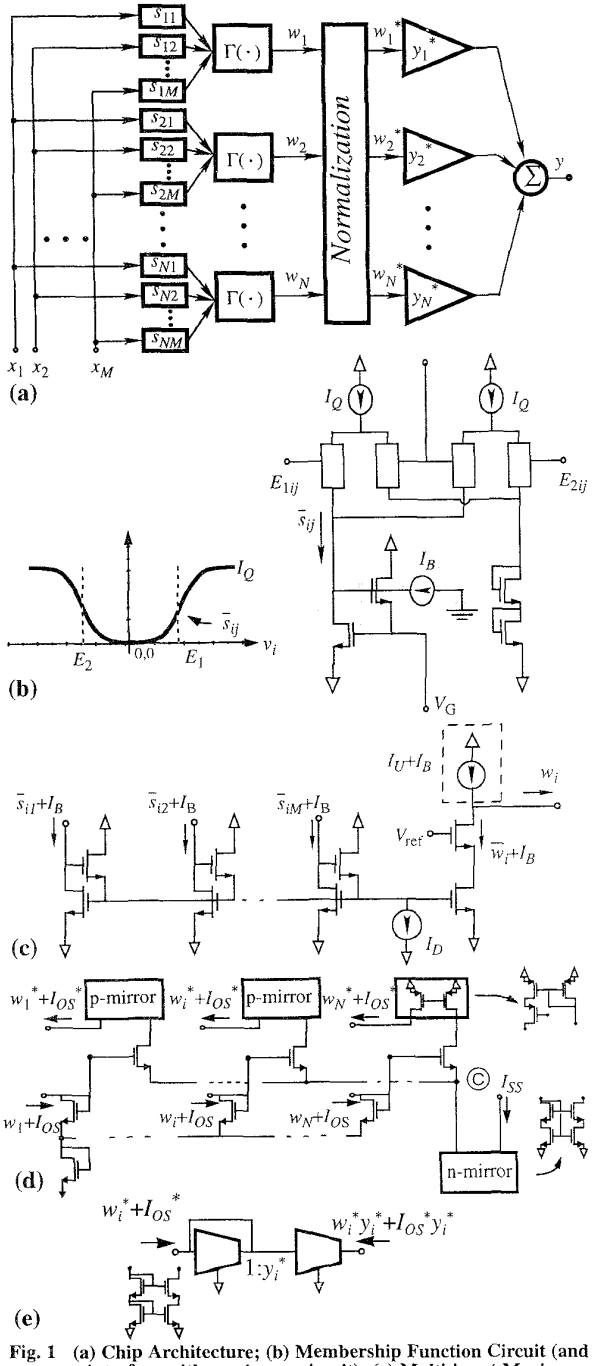


Fig. 1 (a) Chip Architecture; (b) Membership Function Circuit (and interface with maximum circuit); (c) Multi-input Maximum Circuit; (d) Normalization Circuit; (e) Singleton Weighting.

HARDWARE COMPATIBLE LEARNING

The clustering performed by the fuzzy inference procedure is similar to the role played by basis functions in the radial basis functions neural networks (RBFNN) [8], although radial basis functions are not commonly normalized. This leads us to explore learning strategies borrowed from RBFNNs: a clustering algorithm to determine membership functions and an error-correction algorithm for the weights in output layers. This has already been considered at the algorithmic level in [7], using *backpropagation* algorithm for the antecedents (layer 1) and *least mean squares* (LMS) for the consequents (layer 4). However, since backpropagation is hard to implement in hardware we consider *weight perturbation* [9] where derivatives are substituted by finite differences and feedback paths are avoided through the calculation of the influence of each parameter on the global error. If ω is the learning parameter and $\zeta(\bullet)$ the global error at output, a change in the value of ω is given by

$$\Delta\omega = \frac{-\eta\,[\zeta(\omega) - \zeta(\omega + pert)]}{pert} = G(pert)\,[\zeta(\omega) - \zeta(\omega + pert)] \quad (3)$$

where *pert* is a small perturbation, η is the learning rate, and both are constant. Note that weight update hardware involves evaluation of the error with perturbed and unperturbed weight and then multiplication by a constant.

We use this strategy for the membership functions. With regards to the singletons, it is convenient to exploit the similarities of singleton fuzzy inference with the counterpropagation network. This becomes evident when one uses crisp rather than fuzzy sets. In this case the one dimensional projections of the membership functions are as depicted in -- similar to a trained counterpropagation network with Kohonen input nodes and Grossberg output node. Based on this, our learning algorithm uses the outstar rule,

$$y^*_{i_{new}} = y^*_{i_{old}} + \mu\,[T - y(\mathbf{x})] \quad (4)$$

where T is the target output, m is the learning rate, and y_i^* is the singleton whose rule antecedent is maximum, that is $w_i^*(\mathbf{x}) = max\{w_1^*(\mathbf{x}), w_2^*(\mathbf{x}), \ldots w_N^*(\mathbf{x})\}$.

CIRCUIT STRATEGIES FOR ADAPTABILITY

A MOST characteristic of primary importance for analog design is its operation as a VCCS -- modeled by a transconductance gain g_m. Programmability can be achieved by exercising electrical control on g_m. It is already featured by a simple transistor, as Fig. 2(a) illustrates for n-channel, where we assume operation in saturation region within strong inversion. Transconductance g_m can be controlled by the biasing current I_Q. However, this is inconvenient for fuzzy membership function blocks, where any change of the bias current modifies the electrical value of logical "1".

A technique to overcome this problem substitutes Fig. 2(a) with one of the compound transistors depicted in Fig. 2(b),(c),(d). Fig. 2(b) has the same g_m expression as the simple transistor, although β is digitally-controlled. This is achieved by switching elementary devices ON and OFF to the signal path, under the control of a digital word $B = \{b_0, b_1, \ldots, b_P\}$. The sizes of these elementary devices are most typically binary-weighted, thus giving a quadratic

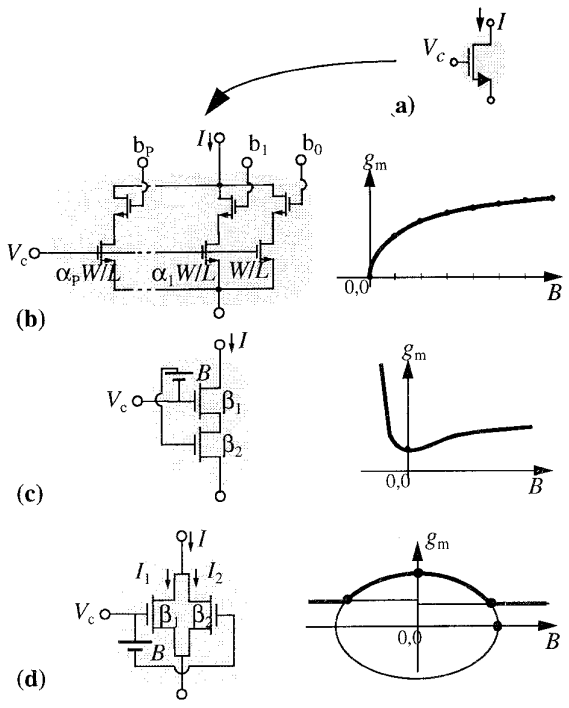


Fig. 2 Transconductors: (a) Simple; (b) Digitally Controlled; (c) Series; (d) Parallel.

relationship between g_m and the decimal number coded in the digital word B. The curve at right of the figure illustrates this situation for a 3bit word. Fig. 2(c) is a *series* configuration where the bottom transistor cannot operate in saturation region due to the biasing voltage B. Thus, assuming that the top transistor operates in saturation region we have the enclosed g_m-B shape which shows a minimum for $B = 0$, and grows monotonically for positive values of B. The exact shape depends on the values of β_1 and β_2; as β_1 and/or β_2 increases the change rate of g_m with B increases as well.

Consider now the *parallel* configuration of Fig. 2(d) with transistors operating in saturation region. The shape of the transconductance expression is an ellipse in the plane g_m - B. Actual devices cover only a portion of this ellipse, which includes the point of maximum transconductance at $B = 0$, and exhibit saturation regions for large negative and positive values of B. The solid line in the figure illustrates this, where the exact shape depends again on β_1 and β_2. The saturation value for $B < 0$ is larger than that for $B > 0$ if $\beta_2 > \beta_1$, and smaller otherwise.

Membership Function Programmability: Fig. 1(b) features membership function characteristic whose *width* and *center* are separately controlled through E_1 and E_2,

$$2\Delta = E_2 - E_1 \qquad 2E = E_2 + E_1 \tag{5}$$

within the common-mode range of the differential pairs, and with a constraint on the minimum width, imposed by the operation of the differential pairs.

The other tunable parameter, the *slope* at the crossover points, is given by,

$$S = \sqrt{\frac{I_Q \beta}{2}} \tag{6}$$

Note that S can be modified on-chip by changing I_Q. However, this forces the inclusion of an additional clamping stage to maintain the level of logical 1 equal for all fuzzy labels, in spite of the actual value of the bias current for each corresponding differential pair. Consequently, the membership function shapes will be less smooth and even more important, the correlation between slope and width increases. For simpler design and easier on-chip tuning, all membership functions should have the same bias current; their slope is then controlled by using compound transistors in the differential pairs.

Singleton Programmability: Similar to the membership function circuits, using compound transistors obtains a current mirror whose input-to-output characteristics are controlled through parameter B. Obtained parametric families for three compound transistor configurations are shown below in the section of Results. The observed nonlinearities are not problematic if the error signals that guide the learning procedure are measured on the chip.

Discussion of Programmability Strategies: The three compound transistors of Fig. 2 have the common feature of controlling g_m without changing the bias current. The pros of digitally-controlled configuration are easier interface to conventional equipments, lower sensitivity to technological parameters, and simpler design; the cons are larger area and power consumption. The other configurations have less control. Apart from these considerations, comparative evaluation of the different strategies for programmability is based on the following criteria:

- variation range of the adaptive parameter,
- variation range of the control parameter,
- influence of the controlled circuit on common-mode input range, and
- smoothness of the relationship between control parameter and slope.

Each compound transistor exhibits pros and cons when contemplated in light of the above cited criteria. Thus, the series configuration features large control range and good input range since the global cut-in voltage equals a simple threshold voltage, V_T. On the contrary, it displays a low range of adaptive parameter -- a negative consequence of the low incremental change of the transconductance with B. On the other hand, the parallel configuration features better gain range, but worse input range since the cut-in voltage of the global transconductor depends on control parameter B. Its control range is also smaller and its non-linearity larger than for the series configuration. Finally, since the switches in the digital configuration must work in saturation, its input range is smaller than the series and the parallel. As a counterpart,

its linearity is also smaller and it is the most flexible implementation in terms of control and adaptive ranges.

RESULTS

In this section some results are given to illustrate viability of proposed learning algorithm as well as circuit strategies for adaptability.

Fig. 3 illustrates the performance of the proposed learning algorithm. The multidimensional function $y = 2 + sin(\pi x)sin(\pi y)$ is taught to a nine-rule controller by showing 36 input-output data pairs in the interval [0, 1]. The system is initialized with membership functions uniformly distributed along the universe of discourse and all singletons equal to 2. Fig. 3 shows the root mean squared error (RMSE) for the proposed learning rule with *pert* = 0.05, η = 0.005 (see (3)), and μ=0.01 (see (4)).

On the other hand, parametric curves in the left side of Fig shows different s_{ij} shapes provided by Fig. 1(b) with different compound transistor configurations and different values of B. Finally, right side of the same figure illustrates singleton tunability.

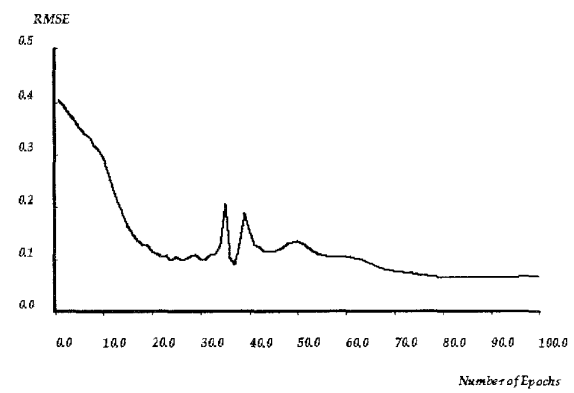


Fig. 3 Illustrating Performance of the Proposed Learning Algorithm.

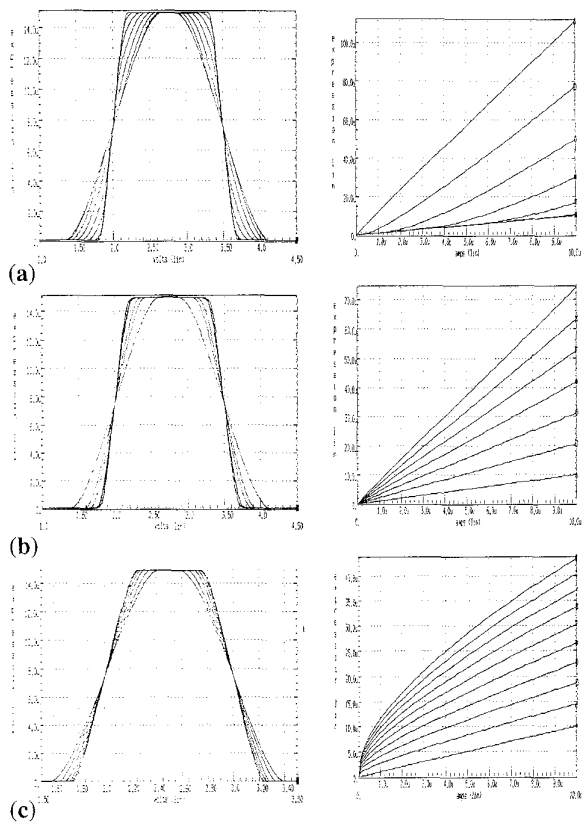


Fig. 4 Parametric families of curves for membership functions (left) and singleton (right) circuits for different values of control parameter B.

REFERENCES

[1] R.J. Marks II (ed.): *"Fuzzy Logic Technologies and Applications"*. New-York: IEEE Press 1994.

[2] K. Namakura et al.: "Fuzzy Inference and Fuzzy Inference Processor". *IEEE Micro*, pp. 37-48, Oct. 1993

[3] T. Yamakawa: "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control". *IEEE Trans. on Neural Networks,* Vol. 4, pp. 496-522, May 1993.

[4] H. Takagi: "Applications of Neural Networks and Fuzzy Logic to Consumer Products". pp. 8-12 in *Fuzzy Logic Technologies and Applications,* New-York: IEEE Press 1994.

[5] J.S.R. Jang and C.T. Sun: "ANFIS: Adaptive-Network-Based Fuzzy Inference System". *IEEE Trans. on Systems, Man and Cybernetics,* Vol. 23, pp. 665-685, May 1992.

[6] T. Takagi and Sugeno: "Derivation of Fuzzy Control Rules from Human Operator's Control Action". *Proc. of the IFAC Symp. on Fuzzy Inf., Knowledge Representation and Decision Analysis*, pp. 55-60, July 1989.

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

[8] D. R. Hush and B. G. Horne. "Progress in Supervised Neural Networks". *IEEE Signal Processing Magazine*, pp. 8-39, January 1993.

[9] M. Jabri and B. Flower. "Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks" *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 154-157, January 1992.

[10] P. A. Jokinen. "On The Relations Between Radial Basis Function Networks and Fuzzy Systems" *International Joint Conference on Neural Networks*, Baltimore, Maryland, June 1992.

[11] J. E. Perkins, I. M. Y. Mareels and J. B. Moore. "Functional Learning in Signal Processing Via Least Squares" *International Journal of Adaptive Control and Signal Processing*, Vol. 6, pp. 481-498, 1992.