A multiplexing architecture for mixed-signal CMOS fuzzy controllers

Limits to precision impose limits to the complexity of analog circuits, hence fuzzy analog controllers are usually oriented to fast low-power systems with low-medium complexity. This paper presents a strategy to preserve most of the advantages of an analog implementation, while allowing a marked increment in system complexity.

Introduction: Because analog fuzzy controllers operate in fully *parallel* mode, they are well suited for applications that require high operation speed. However, the most complex *monolithic* analog controller chips reported to date feature 13-rule at 3-input [1] and 16-rule at 2-input [2] – much smaller than the up to 100-rule and 4-input reported for digital chips [3].

A major obstacle to increasing the complexity of fully-analog controller chips is the existence of *global* computation nodes where the errors caused by the rule antecedent circuits are aggregated; for instance, the centre of gravity is evaluated in such a node. This obstacle is specially pertinent for those architectures based on *lattice* partitions of the universe of discourse because these partitions (otherwise advantageous for control applications [4]) make the rule count increase exponentially with the input count. Thus, the accumulated error increases exponentially as well, and the system architecture may become unfeasible even for low numbers of inputs.

This paper presents a novel *mixed-signal* controller chip architecture which maintains the fully-parallel operation feature of analog chips but obtains the accumulated error independent of the number of rules. It takes advantage of the fact that each fuzzy rule influences the system output only inside a *local*, limited region of the universe of discourse. Consequently, the universe of discourse can be split into sub-regions and the output for each sub-region can be calculated by only evaluating a reduced number of fuzzy rules (called *active* rules [5]). This number is the same for each sub-region, but the corresponding rule parameters differ from one region to another. Our proposal implements this reduced number of rules by a low-dimension *programmable* analog processor, and employs

multiplexing to cover the entire universe of discourse. On chip digital circuits are employed for multiplexing and for programmability.

Architecture and Functional Description: Consider the bi-dimensional lattice partition in Fig.1. It shows the universe of discourse split into *interpolation* intervals, each having a different set of active rules. For instance, any input pair (x_1, x_2) in the light-shaded interval $C_{ij} = [(\varepsilon_{1i}, \varepsilon_{1i+1}) \times (\varepsilon_{2i}, \varepsilon_{2i+1})]$ maps onto an output determined by the rules in the dark shaded interval (active rules) while all remaining rules have no influence on the output. Only the active rule membership functions and their associated singleton values are needed for the interpolation procedure. In addition, the only membership function pieces needed are those which actually contribute to the system output. Thus, in the case illustrated in Fig.1, only the pieces drawn with thick lines and the singletons associated with the four active rule consequents, $y^*_{ij}, y^*_{i(j+1)}, y^*_{(i+1)j}$ and $y^*_{(i+1)(j+1)}$ are needed to generate the output in the interval C_{ij}.

Fig.2 shows the proposed architecture for a controller with M inputs, L labels per input (thus L^M rules), and with S bits per singleton. It comprises the following blocks:

- *A/D Converters*: Their function is to encode the interpolation interval C_{ij} associated with the current input. There are M, one per input, with a resolution equal to the next superior integer $\log_2 L$, i.e. $\text{int}_s(\log_2 L)$. Thus, this battery of converters provides a word of $M[\text{int}_s(\log_2 L)]$ bits that drives the *interval selector* block and the *digital memory* block.

- *Interval Selector*: It selects a set of voltage values $E_1 \ldots, E_k \ldots E_M$ to drive the analog core and, thus, makes it implement the active membership functions.

- *Digital Memory*: It selects the active singleton programming values $y^*_1 \ldots, y^*_l \ldots, y^*_{2^M}$ that configure the *rule block* of the *analog core* consequents of the active rules. These are digital words of as many bits as needed to encode the required set of singleton values.

• *Analog Core*: It performs the fuzzy computation having a set of programming inputs which are driven by the *interval selector* and the *digital memory* blocks. These inputs set up the analog core to work with the rule set that determines the system output, which means specifying the membership functions associated with the rule antecedents as well as the singleton values related to the consequents.The internal architecture of the analog core is the same as for the fuzzy controller in [2], but just 2^M rules are needed here, and only two membership functions per input.

Results and conclusions: An example 64-rule, 2-input, 4-bit singleton controller ($L = 8$, $M = 2$ and $S = 4$) has been designed in a CMOS 0.7μm technology to demonstrate the viability of the proposed architecture.

High-speed flash A/D converters are used for interval encoding. The poly-silicon resistor array used to generate the converter thresholds is also exploited to generate the interval selector voltages. Thus, this latter block includes only analog switches and some logic. The memory block is designed carefully so that the data are put in the singleton programming bus in the proper order – necessary to reduce the memory size by a factor of 4. Finally, the 4-rule 2-input processing core is constructed by using building blocks similar to those presented in [2].

Fig.3 shows the DC surface response of the controller for a case where the singletons are chosen to clearly display all the interpolation points. The DC accuracy was around 1%. Fig.4 illustrates controller transient behavior for different values of x_2 (2.80V, 2.90V, 2.95V and 3V) forcing x_1 to change for each of these values. The input was chosen so that the trajectory goes through different sub-regions. The delay time, including the time required for dynamic reconfiguration, is around 500ns. Chip power consumption is 16mW.

It is illustrative to compare the speed, area and power of this mixed-signal controller to that of a pure analog one with the same number of inputs and rules and digitally-programmable singletons. Assuming that the singletons are encoded with the same number of bits, the size of the digital memory is the same for the two controllers. However,

while the number of rule blocks is L^M for the fully-analog one, it is only 2^M for the new one. The larger the ratio $\alpha = (L/2)^M$ the more advantageous and less error-prone is the new architecture as compared to a fully-analog one. Thus, while the circuit in [2] (designed in a 1μm technology) comprises 16rules with 470ns delay, 8.6mW power consumption and 1.6mm^2 area, the example controller implements 64rules with almost the same delay, 16mW power and only 1mm^2 area.

Fig. 1: Universe of discourse split into *interpolation* intervals

Fig. 2: Proposed architecture

Fig. 3: DC surface response of the example controller

Fig. 4: Controller transient behavior

F.Vidal-Verdú. Dto. de Electrónica. U.Málaga. Complejo Tecnológico, 29081-Málaga, SPAIN. FAX: 34 952132782 Phone: 34 952133326 email: vidal@ctima.uma.es

R. Navas-González. Dto. de Electrónica. U. Málaga. Complejo Tecnológico, Málaga, SPAIN

A.Rodríguez-Vázquez. Instituto de Microelectrónica de Sevilla. Avda. Reina Mercedes s/n, 41012-Sevilla, SPAIN.

Acknowledgement: The works in this papaer has been partially funded by the spanish C.I.C.Y.T. under contract TIC96-1392-C02-02 (SIVA)

References

1 S. Guo, L. Peters, and H. Surmann, “Design and Application of an Analog Fuzzy Logic Controller”. *IEEE Trans. on Fuzzy Systems*, Vol. 4, pp. 429-438, November 1996.

2 F. Vidal-Verdú, R. Navas and A. Rodríguez-Vázquez, "A 16 Rules@2.5Mflips Mixed-Signal Programmable Fuzzy Controller CMOS-1μm Chip", *Proc. of the ESSCIRC*’96, pp. 156-159, Sept. 1996.

3 A. Costa, A. de Gloria, P. Faraboschi, A. Pagni and G. Rizzotto, “Hardware Solutions for Fuzzy Control”. *Proceedings of the IEEE,* Vol. 83, pp. 422-434, March 1995.

4 M. Brown and C. Harris, *Neuro-Fuzzy Adaptive Modeling and Control*. Englewood Cliffs: Prentice Hall 1994.

5 M. Masetti, E. Gandolfi, A. Gabrieli and F. Boschetti, "4 Input VLSI Fuzzy Chip Design able to process an Input Data Set every 320ns", *Proceedings of the JCIS’95*, Wrightsville Beach, North Carolina, USA, October 1995.

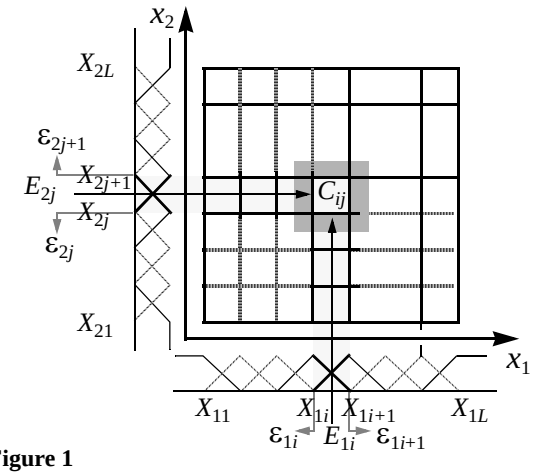

Figure 1

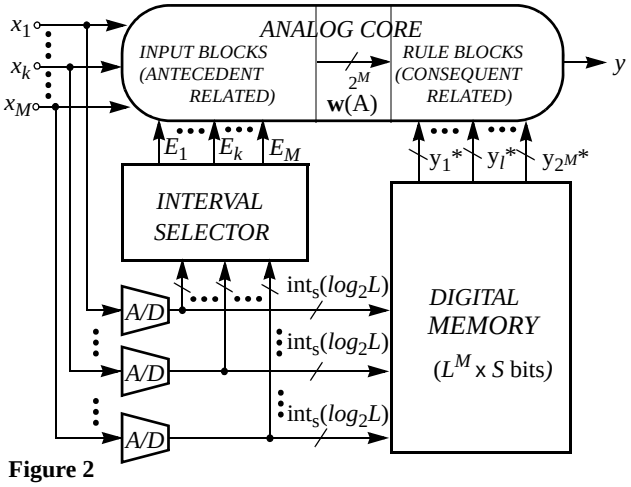

Figure 2

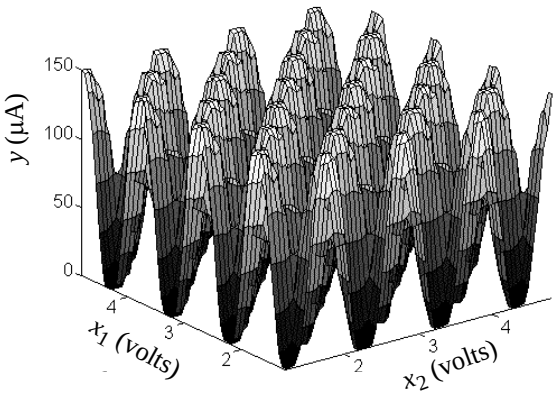

Figure 3

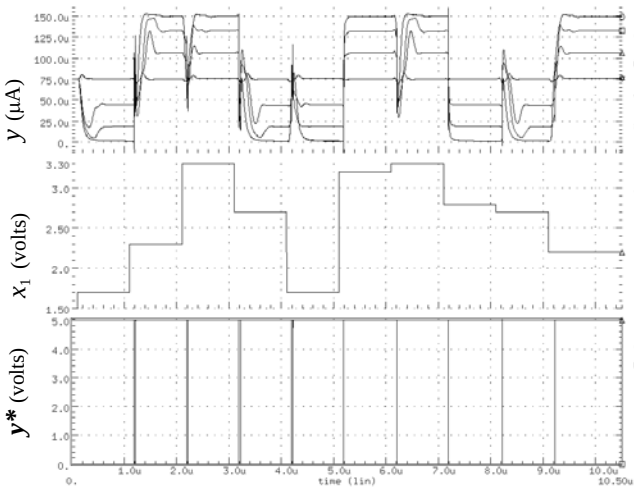

Figure 4

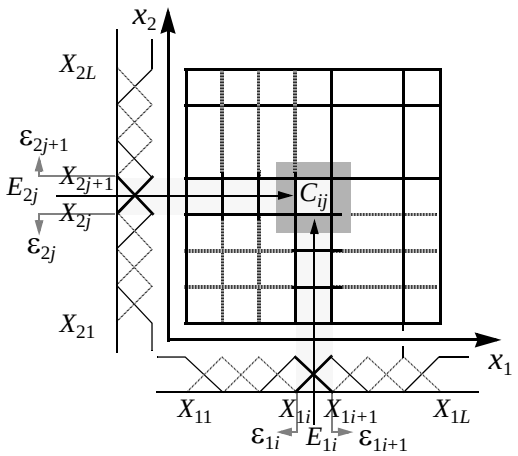

Figure 1

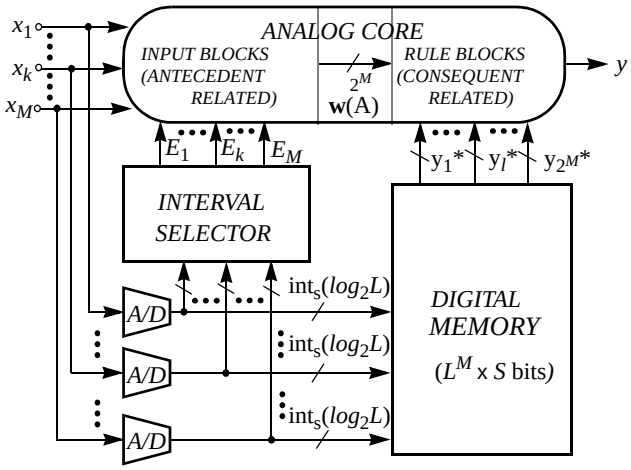

Figure 2

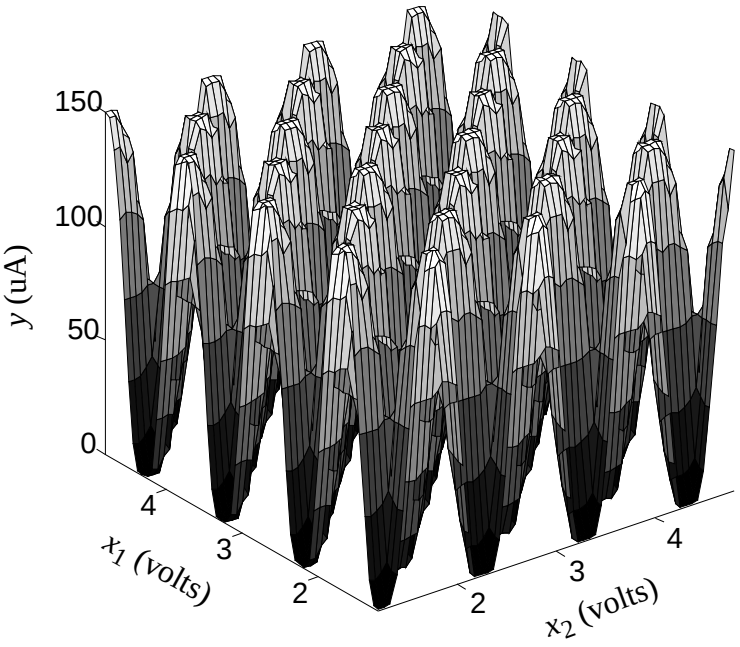

Figure 3

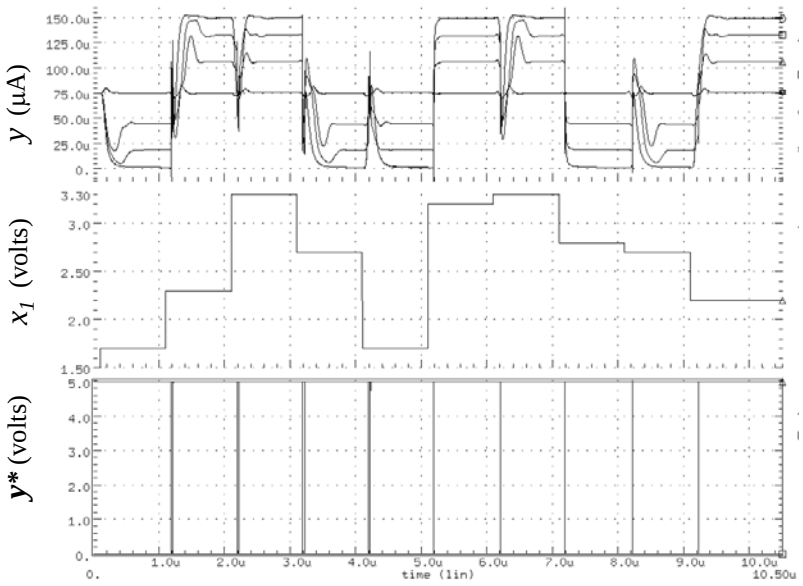
y (μA)
150.0u
125.0u
100.0u
75.0u
50.0u
25.0u
0.
x1 (volts)
3.30
3.0
2.50
2.0
1.50
y* (volts)
5.0
4.0
3.0
2.0
1.0
0.
0.
1.0u
2.0u
3.0u
4.0u
5.0u
6.0u
7.0u
8.0u
9.0u
10.0u
10.50u
time (lin)

Figure 4