A Modular CMOS Analog Fuzzy Controller

Fernando Vidal-Verdú **, Rafael Navas **

**Dto. de Electrónica
Universidad de Málaga
Complejo Tecnológico, Campus de Teatinos, Málaga,
29071-Málaga, SPAIN
email: vidal@ctima.uma.es

Angel Rodríguez-Vázquez *

*Dept. of Analog and Mixed-Signal Circuit Design
Centro Nacional de Microelectrónica-Universidad de
Sevilla
Edificio CICA, C/Tarfia s/n, 41012-Sevilla, SPAIN
email: angel@cnm.us.es

Abstract

The low/medium precision required for many fuzzy applications makes analog circuits natural candidates to design fuzzy chips with optimum speed/power figures. This paper presents a sixteen rules - two inputs analog fuzzy controller in a CMOS 1μm single-poly technology based on building blocks implementations previously proposed by the authors [1]. However, such building blocks are rearranged here to get a highly modular architecture organized from two high level blocks: the label block and the rule block. In addition, sharing of membership function circuits allows a compact design with low area and power consumption and its highly modular architecture will permit to increase the number of inputs and rules in future chips with hardly design effort. The paper includes measurements from a silicon prototype of the controller.

1. Introduction

There are numerous applications where an input vector $\mathbf{x} = (x_1, x_2, ... x_N)^T$ has to be *non-linearly mapped* onto a scalar variable $y = f(\mathbf{x})$[1]. For instance, this problem is found in associative memories and pattern recognition, in control, or behind the modeling of knowledge in intelligent decision-making systems. In many of these applications the mathematical structure of the function is unknown or ill-defined and the nonlinear mapping must be obtained from a collection of input-output data pairs and/or based on the observation of some local features of the system operation. This is intrinsic to the very operation of *fuzzy logic systems* [2] which employs fuzzy inference to obtain a multidimensional function (called *response map*, or *response surface*) that maps $\mathbf{x} = (x_1, x_2, ... x_N)^T$ into a scalar output y. This response map can be represented as an expansion in terms of *fuzzy basis function*, each corresponding to one of the

local pieces of knowledge from which the global operation of the system is built. This resembles the process of creating a function that fits a set of interpolation data -- a well-known problem in *approximation* theory. Local pieces of knowledge in fuzzy systems are expressed by mean of logic rules in natural language like,

IF x_1 is A_{i1} **AND** x_2 is A_{i2} **AND** ... x_M is A_{iM} **THEN** *Consequent* Action

The particular case of fuzzy inference that uses constant terms (singletons) at the consequent of each rule is very suitable for hardware implementations because of its simplicity [3]. This procedure obtains $y = f(\mathbf{x})$ as,

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

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

where $w_i^*(\mathbf{x})$ are the normalized multidimensional fuzzy basis functions which corresponds to normalized rule antecedent outputs, $s_{ij}(x_j)$ are the membership functions associated to linguistic labels A_{ij}, and y_i^* are the singleton values at the consequent of each rule. A conceptual architecture to perform (1) is depicted in Fig.1 where processing is realized through five layers associated to operators in (1).

When high speed/power ratios are required, the use of special ASICs is recommended [4]. Digital implementations of special purpose ASICs achieve higher resolution than their analog counterparts, and many developing tools make them attractive for fast design and prototyping. Analog implementations are usually more costly in design time, and they achieve lower resolution, up to 9 bits in digital standard technologies [5], but get better speed/power ratios and their input-output delay, which is actually the most important parameter in real time control, is usually smaller. Moreover, the electronic devices versatility when the differ-

---

1. Each component of an output vector $\mathbf{y} = (y_1, y_2, ... y_Q)^T$ is treated in this paper as a different single-output systems, that is, $y_j = f_j(\mathbf{x})$, $1 \le j \le Q$.
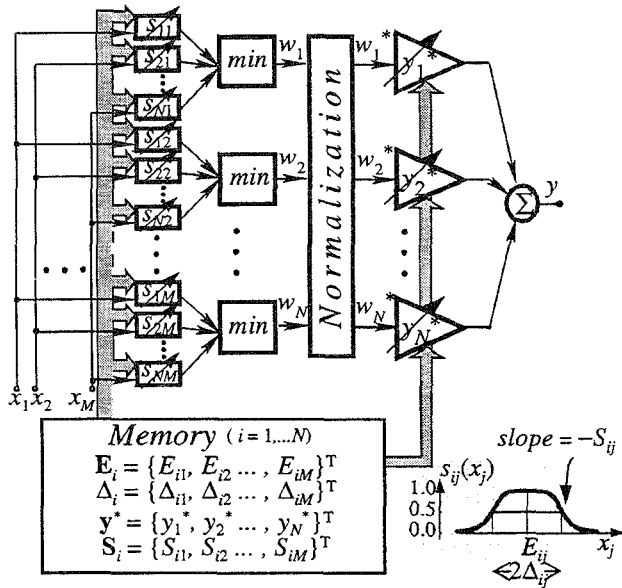
Fig. 1: Neuro-Fuzzy Controller Chip Conceptual Architecture; inset: bell-shaped membership function.

ent operating regions are exploited, allows to get a higher efficiency from them in analog than in digital implementations, thus saving area. The main handicap for the analog implementations, their limited resolution, does not seem a major problem in fuzzy control applications, which often do not need more than 4 bits of resolution [3]. Finally, analog implementations save the A/D and D/A converters used in the interfaces to sensors and actuators, which implies lower delay and cost in area than digital implementations.

An implementation of Fig.1 into silicon was presented by the authors in [1]. Such circuit does not share membership function circuits among rules, thus it allows flexible partitions of the input space. Here, instead of a direct translation of Fig.1 into silicon, circuitry is distributed in order to achieve a high modularity. Thus, minimum and normalization circuits have been split into cells, which are further grouped into two blocks named *label block* and *rule block*. On the other hand, membership functions are shared among different rules, thus there is only one membership function for each label in the rule set, and there is only a set of labels per input that determines its partition. This strategy has two consequences:

- We can generate only grid partitions of the universe of discourse, that is, the partition in each separate dimension or input determine the partition in the overall multidimensional input space. Therefore, scatter and tree partitions [6] are not allowed.
- Membership function circuits can be shared among different rules and do not have to be replicated, thus we save area and power consumption.

In the following, we describe the label and rule high level building blocks as regard to the operators in Fig.1 and their implementation. Then we will see how to interconnect these building blocks to build a highly modular controller and will discuss some performance aspects. Finally, we present some results from a prototype of a sixteen-rules two-inputs controller.

2. Label Block

The tasks realized in the label block are related to the computation of the rule antecedent output from the system input, which corresponds to the operators involved in the numerator of the right part of equation (1), or to the first and second processing layers in the Fig.1. The rule antecedent defines a fuzzy set over the multidimensional input space, thus it provides a membership value to the multidimensional input vector. Such membership value is obtained by selecting the minimum membership value among those associated to each input vector component. The latter membership values are obtained by the membership function circuits, while we need a minimum circuit to compute the minimum of them. Each label block implements one membership function circuit, and the input stages of the minimum circuit for each rule that contains such label.

2.1. Membership function circuitry

The circuit labeled as membership function circuit in Fig.2 obtains a bell shaped curve by adding the outputs from two cross-coupled differential pairs, thus by adding two outputs like,

$$i_k \approx \begin{cases} \sqrt{2\beta I_Q} v_k \sqrt{1 - v_k^2 \beta / (2I_Q)} & |v_k| \le \sqrt{I_Q/\beta} \\ I_Q \operatorname{sgn} v_k & |v_k| \ge \sqrt{I_Q/\beta} \end{cases} \quad (2)$$
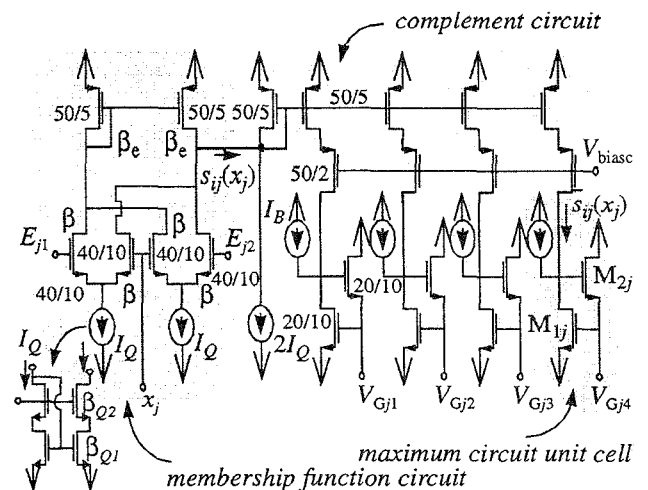

Fig. 2: Label Block.

Then $s_{ij}(x_j)=i_1+i_2$, where $v_1=E_{j1}-x_j$ and $v_2=x_j-E_{j2}$ for $k=1,2$ respectively in (2), and β is the large signal transconductance factor of the transistors in the differential pairs. The parameters related to the shape and location of the membership function (see inset of Fig.1)are the width Δ and the location E,

$$2\Delta = E_2 - E_1 \qquad\qquad 2E = E_2 + E_1 \qquad (3)$$

and the slope at the crossover points S,

$$S = \sqrt{2I_Q\beta} \qquad (4)$$

Note that a differential transconductor, a differential pair, is used as input stage. Differential input transconductors are common in transconductance and voltage mode implementations of membership function circuits because the differential input provides an inherent comparison capability which can be exploited to determine the location and width of the membership function. A third important parameter related to membership function definition, the slope at the crossover points, is given by the transconductance associated to the input transconductor. In addition, membership function values must be constrained between two fixed minimum and maximum values, corresponding to logical values '1' and '0', thus we have to limit the output. Such task is performed in other implementations by rectifying operators, while the implementation in Fig.2 exploits the saturation of the differential pair output curves to make it without any cost. Thus, implementation in Fig.2 exploits at the most the functionality of the transistor to perform all tasks related to location and shaping of the membership functions with just a few transistors.

2.2. Minimum circuit input stage

Minimum operator provides the membership degree of the input vector to a multidimensional fuzzy set. Most implementations in current mode of such operator take advantage from the De Morgan's law [1], which allows to obtain a minimum operator by complementing input and output signals of a maximum operator. Complement operation is easy to realize in current mode, since it is essentially a subtraction of two currents, by exploiting KCL. The circuit labeled complement circuit in Fig.2 performs such task by mean of a current source and a current mirror (complement is denoted with an upper bar), whose multiple outputs provide copies of the complemented membership function output to be used by different rules.

The maximum circuit is split into input and output stages in this implementation. Such circuit is based on a winner-take-all by Lazzaro [7], has a complexity of only $O(2n)$ and a good dynamic response. Each input unit consists of only two transistors, M_{1j} and M_{2j}. The gate of the transistor M_{1j} is shared for all the unit cells of the same maximum circuit,

but only the input unit cell which drives the maximum current imposes the voltage of such common node and is the only that works in saturation, while the transistors M_{1j} in the remaining cells work in ohmic region. Since the output transistor, which is implemented in the rule block, works in saturation and shares the gate with transistors M_{1j}, it replicates the maximum input current. The maximum operator will be completed just by connecting the output nodes of the input unit cells with the input node of the output stage, thus a rule is implemented by joining the proper output label block nodes to a rule block

3. Rule Block

This block performs tasks related to the computation of the rule consequents and the global system output through the center of gravity method, thus the third and fourth layers in Fig.1. However, its input stage corresponds to the output stage of the circuit that implements the minimum operator in the second layer. Such output stage provides the rule antecedent output current to be normalized and weighted.

3.1. Minimum circuit output stage

The minimum circuit output stage, which is depicted at the left of Fig.3 translates the common gate voltage of the minimum circuit into an output current. Since the input transistor in this stage, thus the output transistor in the minimum circuit, works in saturation region, it drives the same current than that transistor in the input minimum cells working also in saturation, which is also that driving the maximum current. The cascode transistor M_{Ci} and the bias voltage V_{biasm} provide a good dc input-output matching of input and output nodes of the maximum circuit, while the complement at output is made by adding the current source I_C. Finally, the current I_b is not necessarily part of the minimum circuit output stage, but is implemented here because it is shared by all input unit cells.
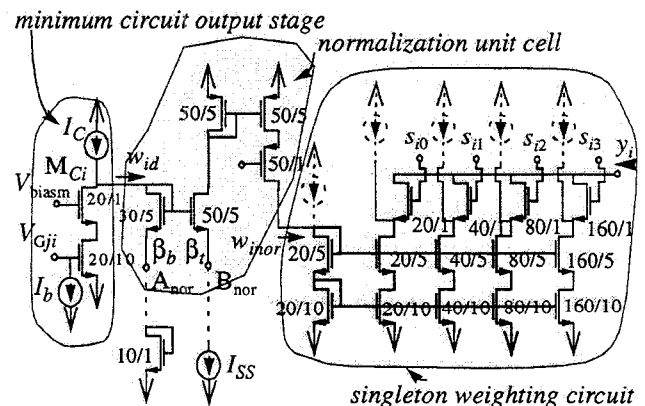

Fig. 3: Rule Block.

3.2. Normalization circuit unit cell

An open chain CMOS normalizer circuit based on a bipolar circuit by Gilbert [8] was proposed by the authors in [1] to perform the normalization at the third layer of Fig.1. Such circuit can be split into unit cells, one per rule, and a common part which consists only of a diode connected transistor and a current source. The unit cell is depicted in Fig.3 as well as this common part, which is connected in the figure with a dashed line to indicate that it is not part of the rule block, but it is implemented in a global bias box. Since normalizing operation of Gilbert's proposal lies on the translinear principle, and the circuit in [1] works above threshold, it actually realizes a non-linear transformation,

$$w_{inor} = \frac{\beta_t}{\beta_b} w_{id} \left[1 + \frac{\eta(\mathbf{w}_d)}{\sqrt{w_{id}}} \right]^2 \qquad (5)$$

where

$$\eta(\mathbf{w}_d) = \frac{\sum_j \sqrt{w_{jd}}}{N} \left(\sqrt{1 + \frac{N\left(I_{ss}' - \sum_j w_{jd}\right)}{\left(\sum_j \sqrt{w_{jd}}\right)^2}} - 1 \right) \qquad (6)$$

$$I_{ss}' = \frac{\beta_b}{\beta_t} I_{ss}$$

which ensures that the sum of all output currents is constant. In addition, it is a soft monotonic transformation, thus the higher the current value at an specific input, the higher its corresponding output current is. Since both previous points are the essence of defuzzyfication, whose purpose is preserving the relative strengths of the rule antecedents at output, thus the meaning of these antecedents as multidimensional fuzzy sets, and non-linear transformations are already accepted in the membership function generation, we can use the circuit in Fig.3 to realize the normalization without feedback nor division, thus with better dynamic response.

3.3. Singleton weighting circuit

Once the antecedent outputs are normalized, a scaling circuit is necessary to perform the weighting in (1). Since the normalization circuit provides currents, the simplest way to perform weighting is through asymmetrical current mirrors, that is through current mirrors whose input and output transistors have different sizes. Digital programmability of the singleton values is got by spliting the output transistors into binary weighted ones, and adding current switches, as depicted in Fig.3, where switches are implemented by simple MOS transistors. Then the output current is,

$$y_i = w_{inor} \sum_{k=0,3} 2^k s_{ki} \qquad (7)$$

where w_{inor} is the output current of the normalization circuit unit cell in (5) and s_{ki} are the digital signals at the current switches gates.

4. Global controller performance and design issues

The last layer in Fig.1 performs the aggregation of all rule consequent outputs. Such task can be realized in current mode just by wiring up all the rule outputs, thus the outputs of the rule blocks, by taking advantage from the Kirchhoff Current Law. In addition, a current source must be attached sometimes to this node to remove a current offset that can be introduced in the normalization circuit to improve the dynamic response. Thus, a complete controller can be obtained by proper connection of a set of rule and label blocks, as Fig.4 illustrates for a sixteen-rules two-inputs controller. In the following, some design considerations are made which concern to the main aspects involved in the analog design of such controller: errors, dynamic behavior, power consumption, range, and dependence on the temperature.

The above presented building blocks must be designed to reduce the errors due to systematic and random causes. The first are mainly avoided by introducing cascode transistors, thus current mirrors that replicate the current sources I_Q, I_C and I_{SS} are cascode current mirrors. The high output impedance that these mirrors provide improves the common mode rejection factor in the input differential pair of the membership function circuit, while the same applies for the normalization circuit. In addition to these cascode current mirrors, cascode transistors and bias voltages in the maximum circuit, the complement unit cell and the normalization unit cell are chosen to improve the dc input-output voltage matching. Singleton weighting current mirror in Fig.3 is also a cascode mirror and output branches are built
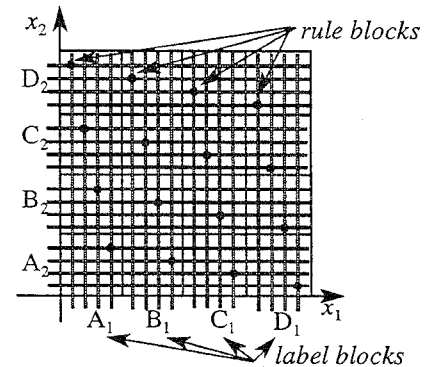

Fig. 4: Conceptual label and rule blocks interconnection.

with unit transistors of the same size than those in the input branch. Finally, in addition to errors due to systematic causes, there are errors due to mismatching between transistors. Large transistors and high current values are required for a good transistor matching, thus trade-offs with speed and power consumption appear.

The common mode input range, closely related to the universe of discourse, of the circuit in Fig.2, thus of the whole controller is,

$$\sqrt{I_Q/\beta_{Q1}} + \sqrt{I_Q/\beta_{Q2}} + V_T + \sqrt{\frac{I_Q}{2\beta}} \leq x_j \leq V_{DD} - \sqrt{\frac{2I_Q + I_B}{\beta_e}} \quad (8)$$

To improve the common mode range by circuit design strategies, we have to implement the source I_Q with high output voltage swing cascode current mirrors, like that depicted in the left bottom corner of Fig.2.

The dynamic behavior of the controller, since there is not feedback between different blocks, is determined by the time constants associated to such blocks, thus their improvement will provide a good global dynamic response. General dynamic considerations apply to Fig.2 and Fig.3, thus better behavior for higher current values and smaller transistors. Bias current I_B is added to improve the dynamic behavior of the maximum circuit. This current is removed at maximum circuit output, which is realized without any cost because we can take advantage from the complement implementation at output. In addition, the current source I_C is used to introduce another offset at the input of the normalization circuit unit cell, which is further removed at system output. Finally, current offsets can also be added in the singleton weighting circuit, as Fig.3 depicts with dashed current sources.

With respect to the controller consumption, let us consider a controller with M inputs, N rules and L fuzzy labels whose maximum singleton value in the associated rule base is $y_{imax}*$, and is loaded with a voltage source V_{load}. The maximum static power consumption is featured by the previous items as,

$$P = (M \times [N \times (I_U + I_B) + L \times (2I_U + I_B)] +$$
$$+ N \times (I_C + I_b) + 2I_{ss} \} V_{DD} + V_{load} I_{ss} y_{imax}* \quad (9)$$

where $I_U = 2I_Q$ and V_{DD} is the voltage supply source.

Finally, as regard to the dependence on the temperature, all the building blocks except the membership function circuit have temperature independent transfer functions. The membership function circuit has a temperature dependent transfer function because of the dependence on the temperature that the differential pair has, which is caused by the large signal transconductance β in (2). However, due to the differential pair symmetry, the location and width defined in (3) are not affected by temperature changes. Electrical val-

ues associated to the logical zero and one are neither affected, because such values are given by the conditions of cut-off or full I_Q current driving in the transistors of the differential pairs, and such references do not depend on the temperature. Thus, the only parameter which is affected by temperature changes is the membership function slope, which depends on β (see eq.(4)). From a global point of view, this means that the slope of the generated function between interpolation points will vary when temperature does, as Fig.5 illustrates for a controller with four rules, where errors below 7.5% are measured in a temperature range of 100 celsius degrees. As a consequence, the interpolation smoothness will vary when the temperature changes, but the interpolation points are not affected if the membership functions are wide enough to saturate in the whole temperature range.
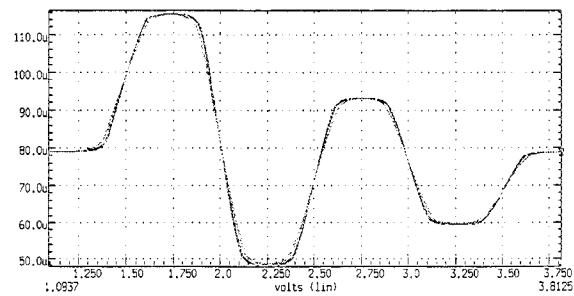

Fig. 5: Illustration of dependence on temperature

5. Chip description and experimental results

Fig.6(a) shows a microphotography of a sixteen-rule two-inputs fuzzy controller based on the previously presented high level building blocks, while Fig.6(b) depicts its physical architecture. The *label blocks* outputs are connected to inputs of *rule blocks* through a *ring bus*. Digital values to program the output current mirror and then the singleton values are stored in a shift register which is the chip internal memory element and is serially programmed through two pads. Apart from digital programmability of singleton values, width and location of membership func-
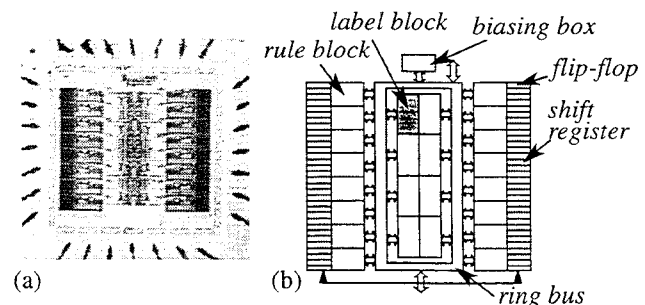

Fig. 6: Chip microphotography (a) and architecture (b).

tions are also analogically programmable by setting two voltages per label, E_{j1} and E_{j2} in Fig.2, associated to membership value 0.5 (*crossover points*) in the membership function circuit related to each label. Finally, bias signals are mainly generated internally and shared by different blocks. They are implemented in a *biasing box* together with the shared parts of the normalization circuit.

The figures Fig.7(a) and Fig.7(b) show two output surfaces generated by the presented chip. The bias signals are V_{DD}=5V, V_{SS}=0V, I_Q=7.5μA, I_B=10μA, I_b=0.5μA, I_C=35μA and I_{ss}=37μA, while the voltages E_{ij} associated to each label were fixed to obtain a uniform lattice partition of the input space. The circuit was loaded with a constant voltage source of 2.5V and a current source to remove the offset introduced in the normalization circuit. Singletons are set to decimal values 1 and 15 in Fig.7(a), which highlights generation with regard to basis functions, while Fig.7(b) illustrates an exemplary surface obtained with different singleton values.

Maximum circuit delay is 471ns (90% of the full scale output current) for a step input, while power consumption is 8.6mW and resolution is around 6.5%. The latter was obtained through Monte Carlo simulations (30 iterations) which take into account parameter mismatching among transistors, with 3σ (or ±1.5σ), where σ is the standard deviation, as error figure. Finally, input voltage range is over 3.25V and the area of the chip without pads is 1.6mm². It is possible to achieve faster designs by introducing bias currents at input and output branches of the current mirror that replicates membership function output, and in the output mirror that implements singleton weighting. It is also possible to achieve a higher precision by inserting the chip in a learning loop with a computer [9].

References

[1] F. Vidal-Verdú and A. Rodríguez-Vázquez: "CMOS Design of Analog neuro-Fuzzy Controllers using Building Blocks" *IEEE Micro*, August 1995.

[2] J.M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial". *Proceedings of the IEEE*, Vol. 83, pp. 345-377, March 1995.

[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] 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.

[5] M.J.M. Pelgrom et al., "Matching Properties of MOS Transistors". *IEEE Journal of Solid-State Circuits*, Vol. 39, pp.
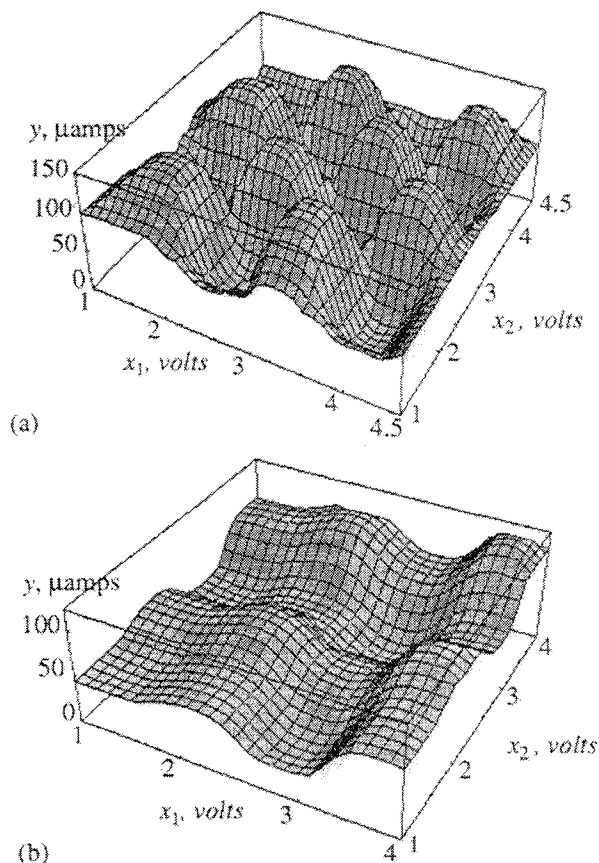
(a)

(b)

Fig. 7: Controller output for two different sets of singleton values.

1433-1440, June 1989.

[6] J.S.R. Jang and C.T. Sun, "Neuro-Fuzzy Modeling and Control". *Proceedings of the IEEE*, Vol. 83, pp. 378-406, March 1995.

[7] J. Lazzaro, R. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of O(n) complexity", *Advances in Neural Information Processing Systems*, Vol. 1, D. S. Touretzky, Ed. Los Altos, CA: Morgan Kaufmann, 1989.

[8] B. Gilbert, "Current-mode circuits from a translinear view point: a tutorial," in *Analogue IC Design: The Current-Mode Approach*, C. Toumazou, F. J. Lidgey, and D. G. Haigh, (Eds.), London: Peter Peregrinus Ltd., 1990.

[9] F. Vidal-Verdú and A. Rodríguez-Vázquez: "Learning under Hardware Restrictions in CMOS Fuzzy Controllers able to Extract Rules from Examples". *Proc of IFSA'95*, Sao Paulo, Brazil, July 1995.