A design approach for analog neuro/fuzzy systems in CMOS digital technologies

Fernando Vidal-Verdú\textsuperscript{b}, Manuel Delgado-Restituto\textsuperscript{a}, Rafael Navas\textsuperscript{b},
Angel Rodríguez-Vázquez\textsuperscript{a,*}

\textsuperscript{a}Department of Analog and Mixed-Signal Circuit Design, Centro Nacional de Microelettronica-Universidad de Sevilla, Edificio CICA, C/Tarfia s/n, 41012-Sevilla, Spain

\textsuperscript{b}Dto. de Electrónica, Universidad de Málaga, Pza. El Ejido s/n, Málaga, Spain

Received 1 January 1996; received in revised form 1 November 1996; accepted 1 May 1997

Abstract

This paper presents adaptive circuit blocks and related learning algorithms to design neuro/fuzzy inference systems using analog integrated circuits in CMOS, standard VLSI technologies. The proposed circuit building blocks are arranged in an architecture composed of five layers: fuzzification, T-norm, normalization, consequent, and output. Inference is performed using Takagi and Sugeno's if-then rules, particularly where the rule's output only contains a constant term — a singleton. The proposed learning scheme uses weight perturbation for the fuzzification layer and outstar for the output layer. A three-input, four-rule controller has been designed for demonstration purposes in 1.6 \(\mu\)m CMOS single-poly, double-metal technology, and obtains operation speed in the range of 5 MFlips with around 1\% systematic errors. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Fuzzy systems; Neural networks; Analog integrated circuit design

1. Introduction

Fuzzy inference uses inexact rules enunciated in natural language to emulate human abilities to handle imprecise or ill-defined systems [1,2]. Since the late 1970s, fuzzy controllers have been proven for many practical applications areas [3]. However, most of these approaches have been
implemented as *software* in conventional computers and consequently obtain an operation speed of about 1 K flips (fuzzy logic inferences per second), which is insufficient for many high-speed control problems, e.g., those related to automotive engines. These applications require dedicated *hardware*. Consequently, diverse authors have been motivated to focus on the development of such hardware using integrated circuit technology [4–10]. Two basic approaches use either *digital* [4,5] or *analog* [6–10] circuit design techniques. Analog realizations enable a simpler interface between physical sensors and transducers, and require no additional A/D or D/A circuitry. They also occupy less silicon area and have a larger speed/power ratio than digital techniques [11,12] — a consequence of their versatile exploitation of small analog devices (formed by a few transistors) for a wide variety of low-level linear and nonlinear signal processing tasks required for fuzzy inference. Additionally, the larger area efficiency of analog implementations favors the incorporation of massive parallelism and thus further increases the throughput. The disadvantages of analog circuits are that they are more difficult to design and test, have poorer programmability, and are less accurate. Consequently, digital techniques are better suited for general-purpose systems and/or whenever power consumption, system dimensions, or operation speed are not the dominant issues, and high accuracy in the operations is a major demand. Otherwise, analog solutions are a convenient alternative. Their programmability can be enhanced through the use of proper circuit strategies [13]. As to accuracy, it is not a major problem for most practical applications, where requirements range from 10 to 1% [6] — within reach of even the cheapest VLSI technologies [14,15].

Previous approaches to the design of analog fuzzy processor chips used either *discrete-time* [10] or *continuous-time* circuits [7–9]. Discrete-time realizations operate in synchronous mode under the control of a clock signal and cannot fully exploit the speed potential of analog circuits. Also, since the technique presented in [10] requires linear capacitors, it is inappropriate for single-poly CMOS digital VLSI technologies — the standard and cheapest one. Speed is no problem for the techniques presented in [9]. However, it does not actually focus on the *monolithic* design of fuzzy processors, but rather on the realization of fuzzy boards using chips fabricated in different technologies: a rule chip in BiCMOS, and a defuzzifier chip in high-voltage bipolar technology. Other contributions found in literature present innovative circuits for some fuzzy building blocks, but not a complete design methodology for analog fuzzy processors [16,17].

A common drawback encountered in previous approaches to analog fuzzy inference systems is their lack of *adaptability*. These approaches are appropriate for applications which involve fixed rule sets, but inappropriate for the most typical practical case where a complete rule set is not available a priori, or the fuzzy labels are not exactly defined, or both change under environmental modifications [18]. These applications require a new generation of hardware to combine the inference capabilities of fuzzy systems and the learning capabilities of neural systems.

This paper presents an analog hardware solution to design *neuro-fuzzy* systems based on a VLSI-friendly inference mechanism [19]. Different processing nodes are identified and realized through circuits operating mainly in current-domain for reduced complexity and increased speed/power. Section 2 introduces some basic concepts and terminology of fuzzy inference, and outlines the inference algorithm chosen — a simplification of Takagi and Sugeno’s singleton
algorithm [20]. The associated architecture has multiple layers containing fixed function nodes, and others whose parameters must be adapted for training purposes. Circuits for the fixed function nodes are presented in Section 3. Section 4 presents the circuitry used for the adaptive nodes. Regarding these nodes, a hardware-compatible learning algorithm is proposed to train parameters, presented in Section 5. Section 6 presents some practical implementation considerations. Finally, Section 7 presents measurements from silicon prototypes in a single-poly n-well CMOS technology to illustrate performance of the proposed building blocks, a controller prototype, and the learning algorithms.

2. Concept of singleton fuzzy inference and chip architecture

Fuzzy inference is a powerful tool to model multidimensional nonlinear systems. For instance, a fuzzy washing machine sets the water level as a function of the clothes' mass, the water impurity, and the time differential of impurity [18]. This is equivalent to capturing the system behavior as a surface response,

\[ y = f(x) \]  \hspace{1cm} (1)

where \( y \) is the output\(^1\) and the vector \( x = \{x_1, x_2, \ldots, x_M\}^T \), the input. The following features distinguish fuzzy inference as a tool to model nonlinear systems:

- The surface response, which is a *global* model predicting the system behavior under any input condition, is obtained as a composition of several functions which capture the *local* features of this behavior.
- These local features represent *insights* about the system operation, and are described through inference rules of the type,

\[
\text{IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ AND } \ldots \times M \text{ is } A_{iM} \text{ THEN Consequent Action}
\]

where \( A_{ij} \) are called *fuzzy labels*, and the consequent action assigns values to \( y \) depending on the outcome of the combination of statements involved in the antecedent clause.
- The matching between input variables and fuzzy labels in the statements ‘if \( x_j \) is \( A_{ij} \)’ is continuously graded from 0 (no matching) to 1 (maximum matching).

Since the statements involved in the fuzzy inference rules are in natural language, for instance ‘if the temperature is low’, this modeling technique is very well suited to simulate human expertise. On the other hand, the continuous matching feature guarantees generalization of the local pieces of knowledge and, hence, smooth surface responses. Finally, since the pieces of knowledge used to build the surface response are local, it enables simpler model updating for environmental changes that affect only limited regions of the input space.

Key points for fuzzy modeling are the calculation of matching degrees among input and fuzzy labels (*fuzzification*) and the composition rules used to first combine the input statements.

---

\(^1\) We will assume without loss of generality that the output is represented by a single variable, \( y \). Each component of a multiple output can be handled as a single variable.
in each rule and then obtain the output from the rule’s consequent actions (defuzzification). Different approaches are reported in literature [1,2,19]. Here, we will consider a particularization of Takagi–Sugeno’s inference where the consequent of each rule is a constant value — a singleton [20]. This choice is advantageous for hardware implementation and programming [6] and obtains the surface response as a weighted linear combination of fuzzy basis functions,

\[ y = f(x) = \sum_{i=1}^{N} y_i^* w_i^*(x) \]  \hspace{1cm} (2)

where each function \( w_i^*(x) \) corresponds to a fuzzy rule and its weight \( y_i^* \) is its associated singleton. These singletons are real parameters, while the basis functions are calculated from the input as,

\[ w_i^*(x) = \frac{\min\{s_{i1}(x_1), s_{i2}(x_2), \ldots, s_{iM}(x_M)\}}{\sum_{i=1}^{N} \min\{s_{i1}(x_1), s_{i2}(x_2), \ldots, s_{iM}(x_M)\}} \]  \hspace{1cm} (3)

where \( \min\{\cdot\} \) is the multidimensional minimum\(^2\) and \( s_j(x_j) \) are nonlinear functions (called membership functions) which codify the degrees of matching between input and fuzzy labels.

Fig. 1 illustrates the procedure of singleton fuzzy reasoning for a system with three input,

\(^2\) In the most general case this must be a T-norm operator, such as multiplication.
three rules, and three fuzzy labels (correspondingly, membership functions) per input. The more general case of \( M \) input and \( N \) rules is mapped onto the five-layer architecture of Fig. 2. It resembles the layered neural network used for nonlinear mapping, i.e., multilayer perceptrons and radial basis functions [21,22,29]. The figure inset shows a typical membership function shape, described by three parameters: slope, center, and width. This shape is bell-like with continuous derivative. However, piecewise-linear (PWL) shapes are also valid for practical purposes [6].

Each layer of Fig. 2 contains a type of processing node. Nodes in the first layer perform nonlinear transformation to implement the \( N \times M \) membership functions. Each of them should allow independent adjustment of their parameters \( E_{ij}, S_{ij} \) and \( \Delta_{ij} \) to enable learning or programming. The second layer realizes the minimum operation \( w_i = \min(s_{ij}) \), while the third layer performs the normalization. Neither of these two layers needs adjustable parameters. The last layer is responsible for the singleton weighting operation and summation. Once again all singleton parameters \( y^*_i \) must be adjustable to enable learning or programmability. In sum, the learning parameters of the proposed architecture are the vector of singletons \( y^* = \{y^*_1, y^*_2, \ldots, y^*_N\}^T \) and the vectors of centers \( E_i = \{E_{i1}, E_{i2}, \ldots, E_{iM}\}^T \), widths \( \Delta_i = \{\Delta_{i1}, \Delta_{i2}, \ldots, \Delta_{iM}\}^T \), and slopes \( S_i = \{S_{i1}, S_{i2}, \ldots, S_{iM}\}^T \) of the membership functions.

In our proposed circuit realization, all internal variables and the output of Fig. 2 \((s_{ij}, w_i, w^*_i, y)\) are realized through currents while input \((x_i)\) are voltages. All circuit discussions in this paper assume that the MOS transistors operate in strong inversion and use the standard square-law approximation to the MOS current-to-voltage characteristics [15].

---

Footnote 3: For completeness, Fig. 2 contains \( N \times M \) membership functions. However, in many practical applications the membership functions associated to a given input may be identical for some rules, or even for the whole rule set.
3. CMOS current-mode fixed-function nodes

3.1. Multidimensional minimum (layer 2)

The circuits used to evaluate membership functions obtain the matching degrees of the fuzzy labels as currents. Each node in this layer evaluates the minimum among $M$ of these currents ($s_{ij}, 1 \leq j \leq M$) to obtain another current $w_i$ which represents the degree of matching of the multidimensional input and the $i$th rule antecedent. This is functionally equivalent to obtaining the complement of the maximum among the complements of these currents,

$$w_i = \min(s_{i1}, s_{i2}, \ldots, s_{iM}) = \max(\bar{s}_{i1}, \bar{s}_{i2}, \ldots, \bar{s}_{iM})$$  (4)

where the upper bar denotes the complement, evaluated using KCL (Kirchhoff Current Law) from the original current as follows,

$$\bar{s}_{ij} = I_U - s_{ij}$$  (5)

where $I_U$ is a unitary current which corresponds to 1 in the vertical axis of the inset in Fig. 2; i.e., the currents $s_{ij}$ and $\bar{s}_{ij}$ are positive and comprised in the interval $[0, I_U]$. After $\bar{w}_i$ is calculated using a multidimensional maximum, $w_i$ is obtained through KCL as $w_i = I_U - \bar{w}_i$.

Fig. 3(a) shows a conceptual CMOS current-mode maximum circuit of $O(N)$ complexity based on the winner-take-all of Lazzaro et al. [23]. For convenience, this figure and the associated explanation directly use the currents $\bar{s}_{ij}$ as direct input. However, in practical circuits all input currents are shifted by a constant current $I_B$ to preclude the transistors to enter in subthreshold for low values of $\bar{s}_{ij}$ and thus decrease the operation speed. Fig. 3(a) exploits the characteristics of MOS transistors operating in ohmic region; in particular, the possibility to reduce its current density by driving it with small $V_{DS}$ values — shown in the shaded inset of Fig. 3. Note that all the transistors at the bottom of Fig. 3(a) have the same gate-to-source
voltage $V_{GS}$, which is also shared by the output transistor $M_O$. The steady-state value of this voltage is set by the largest input current $\bar{s}_{\text{max}}$ to

$$V_{GS} = V_T + \sqrt{\frac{\bar{s}_{\text{max}}}{\beta_b}}$$

where $\beta_b$\(^4\) is the transconductance factor of transistors at the bottom in Fig. 3(a) and $V_T$ is the transistor threshold voltage. All bottom transistors are driven by this common voltage to draw the maximum current $\bar{s}_{\text{max}}$ while the externally applied current may be smaller than $\bar{s}_{\text{max}}$. Thus, the gate of each top transistor becomes an error sensitive node which detects differences between the corresponding external current $\bar{s}_{ij}$ and $\bar{s}_{\text{max}}$. If $\bar{s}_{ij} < \bar{s}_{\text{max}}$, the error current $\bar{s}_{ij} - \bar{s}_{\text{max}}$ is integrated in the gate-to-source capacitor of transistor $M_{ij}$ causing its gate-source voltage to decreases and consequently, the drain-source voltage of the associated bottom transistor $M_{bij}$ decreases until this transistor enters in ohmic region and the error current becomes null.

Fig. 3(a) requires careful design to reduce errors due to channel length modulation if the drains of the output and the input transistors are not equipotential. Although these errors are palliated by inserting cascode transistors (similar to $M_C$) in series to the input branches, we found that this strategy renders poor dynamic response. An alternative with better dynamic response uses adaptive biasing to properly set the gate voltage of $M_C$, $V_{\text{ref}}$. It is based on the calculation of the drain-source voltages of the output transistor $M_O$ and the input transistor which sinks the maximum current,

$$V_{DO} = V_{\text{ref}} - V_T - \sqrt{\frac{\bar{s}_{\text{max}}}{\beta_C}}$$

$$V_{\text{Dmax}} = 2V_T + \sqrt{\frac{\bar{s}_{\text{max}}}{\beta_b}} + \sqrt{\frac{I_D}{\beta_t}}$$

where $\beta_C$ and $\beta_t$ are the transconductance factors of the cascode transistor $M_C$ and the transistors at the top in Fig. 3(a), respectively; $I_D$ is the bias current driving the common gate node. The issue is to adjust $V_{\text{ref}}$ so that $V_{\text{Dmax}}$ equals $V_{DO}$, which can be achieved through Fig. 3(b). It obtains the value of $V_{\text{ref}}$ controlled by the large signal transconductances of transistors $M_{S1}$ and $M_{S2}$. Assuming for simplicity that they are equal, one obtains for Fig. 3(b),

$$V_{\text{ref}} = 3V_T + 2\sqrt{\frac{\bar{s}_{\text{max}}}{\beta_S}} + \sqrt{\frac{\bar{s}_{\text{max}}}{\beta_C}}$$

Thus, matching between $V_{\text{Dmax}}$ and $V_{DO}$ is achieved through an appropriate choice of $\beta_S$, and enables obtaining systematic errors below 0.3% for input currents of up to 20 $\mu$A. On the other hand, in this circuit as in the others, errors caused by mismatching due to random variations of technological parameters are minimized by following analog layout and sizing guidelines, based on the formulation by Pelgrom \[14\].

\(^4\) We define $\beta = kW/L$ where $k$ is a technological parameter and $W$ and $L$ are the channel width and length.
3.2. Normalization circuit (layer 3)

A convenient strategy to evaluate (3) uses feedback to maintain constant the sum of vector \( \mathbf{w} \) components [7]. Fig. 4(a) illustrates the concept, where we assume that the rule’s firing activities are controlled by the error signal, \( e \), at the differential amplifier output. If the open-loop gain is large enough and the loop stable, feedback forces the differential amplifier input to zero, which consequently obtains \( \sum w_i^*(e^*) = 1 \) where \( e^* \) is the steady-state value of the differential amplifier output. Unfortunately, the transient response of this normalization scheme is rather poor — a negative consequence of feedback.

On the other hand, the normalization circuit operation can be summarized as follows

\[ w_i^* = F(w), \quad w_i^* = \kappa w_i \]  

(9)

and

\[ |\mathbf{w}^*| = \sum_{i=1}^{N} w_i^* = I_{SS} \]  

(10)

where \( F(\bullet) \) is an increasing monotonic function of \( w_i \), for \( 1 \leq i \leq N \), \( I_{SS} \) is a reference current, and \( \kappa \) is a scale factor. Fig. 4(b) presents a circuit based on the Gilbert’ BJT normalizer [24], which realizes this function without explicit feedback, and hence yields much better transient response than previous proposals. Like the circuit used for the minimum, offset currents are

Fig. 4. Normalization function: (a) feedback concept; (b) CMOS circuit for open-loop normalization.
also needed here for improved operation — not depicted in the figure for convenience. The core of Fig. 4(b) consists of two source coupled NMOS arrays, enclosed in the shaded area of the figure. Each transistor in the arrays at the bottom implements a nonlinear I/V conversion, and produces a voltage $V_i$ to drive the gate of a corresponding transistor in the top array. The drain currents of these latter transistors are sensed using p-mirrors and routed to the output$^5$. Analysis of this circuit obtains for voltages $V_{GS}$ and $V'_{GS}$ associated with the $i$th bottom and top transistors of the core, respectively,

$$V_{GS_i} = V_T + \sqrt{\frac{w_i}{\beta_b}}$$

$$V'^*_{GS_i} = V_T + \sqrt{\frac{w'_i}{\beta_t}}$$

(11)

where

$$V_i = V_{GS_i} + V_A = V'_{GS_i} + V_B \quad i = 1, 2, \ldots, N$$

(12)

By Eqs. (11) and (12),

$$V_A - V_B = \sqrt{\frac{w'_i}{\beta_t}} - \sqrt{\frac{w_i}{\beta_b}}$$

(13)

which yields,

$$w'_i = \sqrt{\frac{\beta_t}{\beta_b}} \left( 1 + \sqrt{\frac{\beta_b}{w_i}} (V_A - V_B) \right)^2$$

(14)

Summing for all $i$, as in Eq. (10), and after some manipulation, the following expression is obtained for $F(w)$,

$$w'_i = \sqrt{\frac{\beta_t}{\beta_b}} \left( 1 + \frac{\eta(w)}{\sqrt{w_i}} \right)^2$$

(15)

with

$^5$These mirrors are not a fundamental part of the circuit. In the figure they serve to bias the top array transistors in saturation. Besides, they obtain the normalized output currents flowing outwards, which is necessary due to the circuit strategy adopted in the implementation of the last layer of Fig. 2.
To fulfill $w^*_i = \kappa w_i$ it would be convenient if $\eta(w)/\sqrt{w_i}$ in Eq. (15) be equal for all $i$. Note that if the input currents are already normalized ($\sum w_j = (\beta_n/\beta_i)I_{SS}$) then $\eta(w) = 0$. While, according to how $\sum w_j$ differs from $(\beta_n/\beta_i)I_{SS}$ the proportionality constant $\kappa$ becomes more and more $i$-dependent. However, Eq. (10) is always fulfilled. Proper design obtains quasi-linear transformation of $w_i$ into $w^*_i$. However, linearity is not strictly necessary in a Neuro-Fuzzy system, where nonlinearities are tolerated or corrected through adaptation.

We experienced that using Fig. 4(b) improves the dynamic response by a factor of about 4, compared with other strategies based on feedback [7]. Biasing all the input with an offset current $I_{OS}$ helps maintain this advantage by forcing the impedance of input nodes to decrease. Main error sources in the normalization circuit are channel length modulation of the top core transistors and rejection of the common mode. The interface p-mirrors depicted in Fig. 4(b) alleviate the former by keeping the drains of the top array transistors tied to low-impedance nodes. Similarly, common mode rejection errors are reduced by increasing the output impedance of the current mirror used to drive the common source of the top transistors. Fig. 4 shows a cascode structure to obtain this while maintaining a large output voltage range in the operation of this mirror [27].

4. CMOS adaptive function nodes

Our approach to programmability follows a modular strategy based on the use of compound MOS transistors. These compound MOS transistors have the same signal terminals as conventional MOS transistors, but their transconductance is controllable through a parameter $B$ associated to either a digital word or a control voltage. The following presents these compound transistors and how to combine them to realize the nonlinear shapes of membership functions and the scaling required for singleton weighting.

4.1. Compound MOS transistors

A MOST characteristic of primary importance for analog design is its operation as a VCCS — modeled into first-order approximation by a transconductance gain $g_m$. The incorporation of programmability requires that this parameter be electrically controlled. It is already featured by a simple device, as shown in Fig. 5(a) for n-channel, where we assume operation in saturation [15]. The formula enclosed along with the figure shows that $g_m$ is controlled by the bias 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 is to substitute Fig. 5(a) with one of the compound transistors depicted in Fig. 5(b)–(d). Fig. 5(b) has the same $g_m$ expression as the simple transistor, although $\beta$ 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 forming a quadratic relationship between $g_m$ and the decimal number coded in the digital word $B$. The enclosed $g_m - B$ shape illustrates this situation for a 3 bit word.

The compound transistors of Fig. 5(c) and (d) provide continuous control of $g_m$. Fig. 5(c) is a series configuration where the bottom transistor is assumed to operate in ohmic region. Thus, assuming that the top transistor operates in saturation region, one obtains the expression of $g_m$ enclosed in the figure. This function is illustrated through 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 $\beta_1$ and $\beta_2$; as $\beta_1$ and/or $\beta_2$ increase, the change rate of $g_m$ with $B$ increases. Consider now the parallel configuration of Fig. 5(d) with transistors operating in saturation. 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 $B$. This is shown by the solid line in the figure, where the exact shape again depends on $b_1$ and $b_2$. The saturation value for $B<0$ is larger than that for $B > 0$ if $b_1 < b_2$, and smaller otherwise.

There are advantages and disadvantages of each compound transistor when contemplated in light of the following criteria,

- variation range of the adaptive parameter,
- variation range of the control parameter,
- influence on common-mode input range of the controlled circuit, and
- smoothness of the relationship between the control and adaptive parameters; this favors convergence of the learning algorithm.

The series configuration features large control range and good input range since the global cut-in voltage is equal to a simple threshold voltage $V_T$. On the contrary, it displays a low adaptive parameter range — 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 the control parameter $B$. Its control range is also smaller and its nonlinearity larger than for the series configuration. Finally, the input range of the digital configuration is similar to that of the series, and thus greater than the parallel configuration. It is also the most flexible implementation in terms of control and adaptive ranges. However, the linearity in the control of the transconductance is also smaller. Other advantages of the digitally-controlled configuration are an easier interface to conventional equipments, less sensitivity to technological parameters, and simpler design; the disadvantages are larger area and power consumption. The other configurations have less control nodes and thus, easier routing, and consume less silicon area and power; their continuous nature also renders them more suitable for learning purposes. On the other hand, they are more sensitive to technological parameters and more difficult to design.

4.2. CMOS membership function circuit in transconductance mode

Input to the membership function circuits are voltages; and output are currents. One design alternative consists of generating the nonlinear function in current domain, and using a front-end quasi-linear transconductance amplifier [25] as shown in Fig. 6. Alternatives for the realization of the quasi-linear transconductance amplifier use state-of-the-art CMOS schematics, as found in [24]. As to the nonlinear current shaping, Fig. 7 shows a circuit proposed by the authors in [16], which is based on the use of feedback current switch of Fig. 7.

![Fig. 6. Realization of a transconductance membership function by current shaping.](image-url)
[26]. This circuit features very high resolution in the rectification of the input current with low input voltage excursions for large current input ranges [16,26]. The center of the membership function is directly set through the voltage $E_{ij}$. Its slope is controlled by $g_m$ and the gain of the first bottom current mirror in Fig. 7, and can be set using compound MOS transistors in this mirror. The width can be set through the current source $T_{ij}$.

Fig. 7 is well-suited for analog fuzzy controllers whose input are currents. However, when input are voltages it does not yield optimum area occupation and speed, since the transconductor used for voltage-to-current transformation must be linearized. An alternative to save silicon area and increase speed is to exploit the large signal characteristics of the transconductor to build the membership function directly in transconductance domain. For this, consider the differential amplifier of Fig. 8(a) and assume that the rectangle represents a single MOS transistor. Analysis obtains the following expression for the large signal transconductance,

$$
\begin{align*}
I_o & \approx \begin{cases} 
\sqrt{2\beta I_Q v_i} \sqrt{1 - \frac{v_i^2}{2I_Q}} & |v_i| \leq \sqrt{\frac{I_Q}{\beta}} \\
I_Q \text{sgn} v_i & |v_i| \geq \sqrt{\frac{I_Q}{\beta}}
\end{cases}
\end{align*}
$$

(17)

whose shape, depicted along side Fig. 8(a), shows that the large signal transconductance is a sigmoid with saturations at $+I_Q$ and $-I_Q$. Thus, cross-coupling two differential pairs as in Fig. 8(b) obtains the bells depicted on the right-hand side of Fig. 8(c). Of these, the one at the top is obtained by KCL aggregation of the differential output currents of both pairs, and ranges
between 0 and $2I_Q$. On the other hand, aggregating only the positive or the negative output current components of each pair directly obtains the complementary bell-like characteristics shown at the bottom in Fig. 8(c), whose use simplifies the interface to the circuit used to calculate the minimum.

The $i_o$ curve and the $i_{o-}$ curve of Fig. 8(c) have the same **width** and **center** which are separately controlled through voltages $E_1$ and $E_2$,

$$2\Delta = E_2 - E_1 \quad 2E = E_2 + E_1$$

within the common-mode range of the differential pairs, and with a constraint on the minimum width $\Delta_{\min} = (I_Q/\beta)^{1/2}$, imposed by the operation of the differential pairs.

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

$$S = n\sqrt{\frac{I_Q\beta}{2}}$$

where $n$ is 2 for the $i_o$ curve and 1 for $i_{o-}$. 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. Fig. 9(a)–(c) show a family of $I_{o+} - I_Q$ shapes obtained from Fig. 8(b) when the rectangles are substituted with the different compound transistors for different values of $B$.

4.3. Singleton weighting

Singleton weighting implies multiplying each normalized rule activity $w_i^*$ by its associated singleton $y_i^*$. Then, the inferred output is readily obtained by routing these $N$ current components to a common node to obtain,

$$y = \sum_{i=1}^{N} y_i^* w_i^*$$

(20)

Adaptive singleton scaling is achieved using a generalized current mirror built with compound transistors. Fig. 10 shows the concept of a generalized current mirror [27], consisting of one input and one output transconductor. For proper operation, these transconductors must have
the following voltage–current characteristics,

\[ i_2 = P u(v_1, v_2) \]

\[ i_1 \approx 0 \] \hspace{1cm} (21)

here \( u(\bullet) \) is an invertible function, and the characteristic is parameterized by a controllable scale factor \( P \). Note in Fig. 10(a) that the input and output transconductors have different parameter values, \( P_{in} \) and \( P_o \). Input device is feedback, while output device is in open loop. Assuming that transconductors are matched, their output are equipotential, and their input currents are negligible (for the proposed compound transistors the latter holds exactly at DC) obtains,

\[ i_o = \frac{P_o}{P_{in}}w^* \] \hspace{1cm} (22)

where \( P_o/P_{in} \) sets the singleton value. Fig. 11 illustrates the electrical control of the singleton achieved by using compound transistors in the generalized current mirror structure. It depicts the mirror gain as a function of \( B \) for the three compound transistor configurations, and the formulae enclosed along with the figure show the associated large-signal expressions, where the subindex ‘o’ stands for ‘output’, ‘i’ for ‘input’, and the control parameter \( B \) refers only to the output compound transistor. These expressions are further illustrated through the parametric curves families in Fig. 12(a)–(c). The nonlinearities in Fig. 12(a) and (c) arise because the output characteristics of the parallel and the series compound transistors are nonlinear in the

Fig. 11. Adaptive current mirrors: (a) with serial transconductor; (b) with parallel transconductor; (c) with digital transconductor. (subindexes ‘i’ and ‘o’ denote input and output, respectively; transistor numbers follow Fig. 5).
controlling parameter. However, these nonlinearities are not problematic in neuro-fuzzy systems, where the error signals that guide the learning procedure are measured on the chip.

Actual realizations of the transconductors must also incorporate some circuit strategy to guarantee that input and output nodes are equipotential and/or to reduce the influence of voltage excursions at the output node [27]. A convenient technique for the circuits proposed herein uses stacked cascode mirrors — depicted in Fig. 10(b) [16].

4.4. Implementation of the control voltage $B$

Since the DC input impedance of MOS transistors is essentially infinite, the level shifting between both transistor gates in the compound structures of Fig. 5(c) and (d) can be realized using a floating capacitor and a refreshing mechanism to compensate information losses due to leakage current. Fig. 13(a) illustrates the concept for the series compound transistor. Two
capacitors are needed: $C_1$ is always connected to the transistor gates; $C_2$ is switched ON and OFF to the gates under the control of a clock $\Phi$. While $\Phi$ is high, it is OFF and refreshed using an A/D–D/A loop comprising a read phase, $\Phi_1$, and a write phase, $\Phi_2$. Once the write phase is over, $C_2$ is switched ON to obtain,

$$B_{\text{new}} = \frac{C_1 B_{\text{old}} + C_2 B_{\text{ref}}}{C_1 + C_2}$$

(23)

where $B_{\text{old}}$ is the value of $B$ before refreshing, $B_{\text{new}}$ after refreshing, and $B_{\text{ref}}$ is the value written in $C_2$. From Eq. (23), the larger $C_2$ is compared with $C_1$, the closer $B_{\text{new}}$ is to $B_{\text{ref}}$. Floating capacitors are only needed if the transistor gates are AC-driven. On the contrary, if transistors are DC-driven, two grounded capacitors with voltage difference $B$ can be used (Fig. 13(b)).

5. Hardware compatible learning

Fig. 14 shows the concept of supervised learning applied to the management of parameter adaptation in a fuzzy engine. It highlights three major problems:

- How to calculate the updated parameter values from the error signal.
- Which circuit strategies to apply to control the parameters $E_{ij}$ (center), $\Delta_{ij}$ (width), $S_{ij}$ (slope), and $y_i^j$ (singleton) through intermediate voltages or/and currents.
- How to store these intermediate variables during chip operation.

The last two have been covered in the previous section on hardware-compatible learning; here we focus on the algorithmic issue indicated in the first point. To this end we can take advantage of the many similarities between the chip architectures of Fig. 2 and of neural
networks, which are highlighted by recasting Fig. 2 into Fig. 15(a). There, each input neuron in the input layer corresponds to the rule antecedent and has a nonlinear transfer function $w^i(x)$; the activation function of the output neuron is unity.

From a geometrical point of view, $w^i(x)$ is a multidimensional membership function whose one-dimensional projections are bells (Fig. 15(b)). They divide the input universe into clusters — similar to the role of basis functions in the radial basis functions neural networks (RBFNN) [21,29], although radial basis functions are not commonly normalized. This clustering is clearly displayed in Fig. 15(d), showing measurements taken from a silicon prototype of an analog fuzzy controller. The figure depicts a two-dimensional projection of the surface response and clearly highlights the clustering performed by the inference procedure. Four different clusters are identified, one per each inference rule.

The similarity between singleton fuzzy and RBFNNs leads us to explore learning strategies borrowed from RBFNNs: a clustering algorithm to determine membership functions and an error-correction algorithm to determine weights in output layers. This has already been explored at the algorithmic level in Ref. [21], using the backpropagation algorithm for the antecedents (layer 1) and least mean squares (LMS) for the consequents (layer 4). However, backpropagation is difficult to implement in hardware, and leads to the consideration of weight perturbation [28]. It substitutes derivatives with finite differences and calculates the influence of each parameter on the global error, thus avoiding feedback paths. If $\omega$ is the learning parameter and $\zeta(\bullet)$ the global error at output, a change in the value of $\omega$ is given by
\[
\Delta \omega = -\eta \frac{[\zeta(\omega) - \zeta(\omega + \text{pert})]}{\text{pert}} = G(\text{pert})[\zeta(\omega) - \zeta(\omega + \text{pert})]
\]  
(24)

where ‘pert’ is a small perturbation, \( \eta \) is the learning rate, and both are constant. Note that weight update hardware involves evaluation of the error with perturbed and unperturbed weight, followed by multiplication by a constant.

This is the strategy used for the membership functions. For the singletons, it is convenient to exploit the similarities of singleton fuzzy inference with the counterpropagation network. This becomes evident when crisp rather than fuzzy sets are used for the input labels. In this case, the one-dimensional projections of the membership functions are as depicted in Fig. 15(c) — similar to a trained counterpropagation network with Kohonen input nodes and Grossberg output node. On this basis, our learning algorithm uses the outstar rule,

\[
y_{i_{\text{new}}} = y_{i_{\text{old}}} + \mu [T - y(x)]
\]  
(25)

where \( T \) is the target output and \( \mu \) is the learning rate; \( y_{i}^{*} \) is the singleton whose rule antecedent is maximum, that is \( w_{i}(x) = \text{max}\{w_{1}(x), w_{2}(x), \ldots, w_{N}(x)\} \). The use of this learning equation can also be justified in light of the LMS algorithm and taking into account that input to the output layer are normalized. Due to this normalization, the LMS update rule reduces to [29],

\[
y_{i_{\text{new}}} = y_{i_{\text{old}}} + \mu w_{i}^{*}(x) [T - y(x)] \quad \forall i
\]  
(26)

which in its turn reduces to Eq. (25) by substituting the multiplication operation with the maximum operation. It is worth noting the significant advantage of normalization, when compared with the intensive computations needed to learn output layer weights in classical LMS [30]. Also note that the hardware used to obtain level shifting \( B \) in Fig. 13 can be exploited to implement learning. In a learning cycle we obtain,

\[
B_{\text{new}} = B_{\text{old}} + \gamma (B_{\text{teach}} - B_{\text{old}})
\]  
(27)

where \( \gamma = C_1/(C_1 + C_2) \), \( B_{\text{old}} \) is the value of \( B \) before learning, \( B_{\text{teach}} \) is the target value of \( B \), and \( B_{\text{new}} \) is the value of \( B \) after learning.

6. Interfaces between building blocks

6.1. Interface to membership function circuitry

Bear in mind that the output \( i_{o} \) of the membership function circuit of Fig. 8(b) already has the shape of a complemented bell. On the other hand, the use of PMOS instead of NMOS transistors in this circuit produces a current which leaves, rather than enters, the output node. This enables direct interface to a minimum block built with NMOS transistors, as shown in Fig. 16(a), where the devices in the shaded area pertain to the minimum circuitry and the unitary current is chosen such that \( J_{U} = I_{Q} \). This circuit determines the input voltage range of the controller (the universe of discourse). Restricting this voltage to avoid transistors entering
in ohmic region or cut-off give the following equation for the common mode input range,

\[ V_T + \sqrt{\frac{I_D}{\beta_t}} + \sqrt{\frac{I_Q + I_B}{\beta_b}} \leq x \leq V_A - V_T - \sqrt{\frac{I_Q}{2\beta_{M_N}}} \]  

(28)

where \( V_A \) is the output voltage of the current mirror which realizes the bias current \( I_Q \). Transistors \( M_{S1} \) and \( M_{S2} \) in Fig. 16(a) produce a voltage shift to match the DC voltage at the two output nodes of the cross-coupled differential pairs and thus, reduce errors due to channel length modulation. This voltage shift is determined by transistor geometries, and is calculated to make the voltage at both output nodes equal in common mode.

Fig. 16(b) shows an alternative that also uses NMOS transistors in the membership function differential pairs. In this case, common mode input range is

\[ V_A + V_T + \sqrt{\frac{I_Q}{2\beta_{M_N}}} \leq x \leq V_{DD} - \sqrt{\frac{I_Q}{\beta_{M_N}}} \]  

(29)

where \( I_U = 2I_Q \) and we assume that the lower supply rail is 0 V and the upper, \( V_{DD} \). As long as transistors at top p-mirrors are equal, voltage matching in common mode between output nodes of differential pairs is now a direct consequence of the implementation. Although the cost in silicon area is larger, this circuit features almost 30% larger input ranges and, through optimum design, may obtain higher speed due to the larger mobility of electrons as compared to holes. It also enables reducing the area and parasitics penalties associated with obtaining large values of slope at the crossovers. Finally, it allows easy sharing of membership function circuits between different maximum circuits, that is between different rules, by using multi-output current mirrors at membership function circuit output, as shown in Fig. 16(b).

Apart from DC matching, another static error source is finite output resistance of the current mirrors that realize the current sources. Like Fig. 4, this produces poor rejection of common mode, and is improved by using cascode mirrors with large output swing.
Cascode transistors also reduce errors in the interface of Fig. 16(b) with the maximum circuit. Since the input node of the maximum circuit is low-impedance, these cascode transistors are not intended to increase the output impedance of the membership function block. However, if $V_C$ is chosen properly, their use serves to preserve DC matching — the main source of errors in Fig. 16(b).

6.2. Interface between maximum and normalization circuits

Since the output current of the maximum circuit circulates outwards, interface with normalization circuit is direct. However, a current offset $I_{OS}$ must be added to preclude input transistors of the normalization circuit to enter subthreshold and thus degrade dynamic response. This offset results in another offset at the output, which must be eliminated. One possibility is proper sizing of bottom and top transistors in the normalization circuit. Another possibility is to use current shifting, at either the normalization interface or the output node. The latter provides optimum dynamic response; however, since the bias current required in this case depends on the singletons, it has the disadvantage of requiring more involved circuitry and larger area occupation.

6.3. Interface between normalization and singleton weighting circuits

The output current of the normalization of Fig. 4 can enter on a NMOS weighted current mirror directly. Again, cascode transistors at the normalization circuit output provide good DC matching, thus minimizing errors. The output stage of the normalization circuit does not impose severe restrictions on the input range of the singleton weighting circuit. Thus, we must choose cascode configuration for this stage, attending to requirements on controller output.

7. Results

Fig. 17 is a microphotograph of a 3-input 4-rule controller prototype fabricated in 1.6 μm CMOS single-poly technology. The circuit was designed taking into account systematic errors and random fluctuations of the technological parameters [14] to maintain approximately 1%. Fig. 18 shows measurements for different building blocks of this prototype. Fig. 18(a) and (b) illustrate the operation of the compound transistors. Fig. 18(a) shows an experimental current transfer characteristic, measured on a current mirror realized with the series compound transistor. Similarly, Fig. 18(b) shows measurements for a membership function circuit realized with the parallel compound transistors. In both cases, the floating control voltage is generated on chip from an external grounded voltage. Fig. 18(c) illustrates the combined operation of the membership function and the minimum circuits, and Fig. 18(d) illustrates operation of the normalization circuit. Fig. 19(a) shows the transfer characteristics measured when the controller is used to interpolate a sine wave and Fig. 19(b) shows a two-dimensional surface map measured for a particular setting of the singleton values. Operation speed of the prototype is about 5 MFlops with $\pm 2.5$ V and 15 μA tail current.
Fig. 17. Chip microphotograph of the 1.6 μm CMOS prototype.
The proposed learning technique have been validated using behavioral models of the blocks to represent the nonlinearities that degrade in situ chip operation. Its performance is illustrated for the two-dimensional surface,

\[ \Phi(x,y) = 2.5 + \sin(\pi x) \sin(\pi y) \]  

which was taught to the controller inside the interval [0,1] \times [0,1]. The three options presented for membership functions and singletons result in many alternative combinations, of which only five are considered here. One uses digital transistors for membership functions and singleton. The rest cover the four possible combinations between the remaining composite transistors: parallel transistors in membership function and series in singleton (\textit{parser}); series transistor in membership function and parallel in singleton (\textit{serpar}); and parallel or series in
Fig. 19. Measurements from the controller: (a) interpolation of a sine function; (b) bi-dimensional surface response.
both memberships and singletons (parpar and serser, respectively). Several simulations have been performed for different resolutions and learning step sizes for the hybrid and the weight perturbation algorithms. Building Blocks were designed to cover the same range: [1,4] for singleton values; [1,2] for membership function slopes, and [−0.5,1.5] for parameters $E_{kij}$. Fig. 20(a) shows an example of the learned surface.

Learning parameters for weight perturbation algorithm are $\eta_{pert} = 0.03$, $\eta/\eta_{pert} = 0.26$ with 7 bit resolution for all combinations, with the exception of the digital combination, whose parameters were scaled to $\eta_{pert} = 0.12$ and $\eta/\eta_{pert} = 0.5$ with the same resolution. For the hybrid algorithm, parameter $\mu$ was set to 0.01 for ‘analog’ combinations, except for the combination with series transistors in both adaptive layers, where its value was 0.03. Its value for the digital case was 0.1.

Fig. 20(b) and (c) show RMSE curves for the hybrid algorithm, and Fig. 20(d) gives the curves of the weight perturbation algorithm. Note that the hybrid algorithm shows improved performance with respect to the weight perturbation algorithm. This is also observed in Table 1, by two performance parameters [31]:

- the asymptotic RMSE error, and
- the convergence time $T_C$
The hybrid algorithm provides smaller convergence time for most cases. This is probably because this algorithm favors the local feature by reinforcing rules which have maximum influence on the training data. This local feature is also important because it enables defining a correspondence among singleton and function values in the area of the universe of discourse, i.e., in the area ‘occupied’ by the rule. This ‘transparency’ simplifies the incorporation of knowledge into fuzzy systems, and thus fully realizes the features of fuzzy paradigm. Thus, a neurofuzzy system trained with the hybrid algorithm is easier to initialize and has less possibility to be captured by local minima. Finally, note that our learning technique requires only 7 bits — affordable with even the cheapest VLSI technologies.

8. Conclusions

We will conclude by stating that careful modeling of the non-idealities that degrade the precision of analog fuzzy circuits and the use of sound design strategies allow obtaining the accuracy required for practical applications. However, this implies specialized know-how and long design cycles and consequently, analog fuzzy chips are not the best choice for rapid system prototyping. Other drawbacks of analog fuzzy circuits are low flexibility and poor programmability. We are currently extending our design technique to the design of a new generation of mixed-signal chips that combine the area and power advantages of analog with the flexibility of digital. The basic objective is to enable system-level designers to cover a large variety of problems through programming, instead of designing a different chip for each application.

References


Fernando Vidal-Verdú received the Licenciado en Física degree from the University of Seville, Spain. Since 1992 he has been with the Department of Electronics and Computer Architecture of the University of Málaga, Spain, where he is currently a teaching assistant. His research interests include analog and mixed-signal IC design and neural and fuzzy networks. He is currently working towards a Ph.D. degree in the field of hardware implementation of neuro/fuzzy controllers.

Manuel Delgado-Restituto received the Physics degree in Electronics in 1988 from the University of Seville, Spain. In 1988, he joined the Department of Electronics and Electromagnetism at the University of Seville, where he is a research assistant. He is also at the Department of Analog Circuit Design of the Centro Nacional de Microelectrónica. His research interests include design and modeling of mixed-signal integrated circuits and nonlinear networks.

Angel Rodríguez-Vázquez received the Doctor en Ciencias Físicas degree in 1983, from the University of Seville, Spain. Since 1978 he has been with the Department of Electronics and Electromagnetism at the University of Seville, where he is a Professor of Electronics. He is also at the Department of Analog Circuit Design of the Centro Nacional de Microelectrónica. His research interests are in analog VLSI, including neural, fuzzy and chaotic circuits, mixed-signal IC design, and computer-aided design and modeling of analog integrated circuits. He has published about 50 journal papers, about 90 conference papers and several book chapters. One of these papers deserved the 1995 Guillemin-Cauer award of the IEEE Circuits and Systems Society. Since June 1993, Dr. Rodríguez-Vázquez has been also an Associate Editor of IEEE Transactions on Circuits and Systems.

Rafael Navas received the degree of Licenciado en Ciencias Físicas (Branch of Electronics) from the University of Granada, Spain, in 1987. From 1987 to 1993, he worked as a Design Engineer at the microelectronics group of the R&D department of Fujitsu España S.A. During this period, he participated in several ASICs specification and design projects. In 1993, he joined the University of Málaga, Spain, where he is currently teaching and making research towards the Ph.D. degree. His research activity is focused on the specification and design of integrated circuits to implement fuzzy and neuro/fuzzy controllers using analog and mixed-signal techniques and their applications.