TOWARDS THE IC IMPLEMENTATION OF ADAPTIVE FUZZY SYSTEMS

Iluminada Baturone, Santiago Sánchez-Solano, José L. Huertas

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA),
E-41012, Sevilla, Spain

IEICE Trans. on Fundamentals of Electronics. Communications and Computer Sciences, Vol. E81-A, nº 9, pp. 1877-1885, Septiembre. 1998.

Towards the IC Implementation of Adaptive Fuzzy Systems

Summary.

The required building blocks of CMOS fuzzy chips capable of performing as adaptive fuzzy systems are described in this paper. The building blocks are designed with mixed-signal current-mode cells that contain low-resolution A/D and D/A converters based on current mirrors. These cells provide the chip with an analog-digital programming interface. They also perform as computing elements of the fuzzy inference engine that calculate the output signal in either analog or digital formats, thus easing communication of the chip with digital processing environments and analog actuators. Experimental results of a 9-rule prototype integrated in a 2.4-μm CMOS process are included. It has a digital interface to program the antecedents and consequents and a mixed-signal output interface. The proposed design approach enables the CMOS realization of low-cost and high-inference fuzzy systems able to cope with complex processes through adaptation. This is illustrated with simulated results of an application to the on-line identification of a nonlinear dynamical plant.

I.- Introduction.

A typical multi-input single-output fuzzy system contains a set of IF-THEN rules like the following:

$$\text{Rule } r\text{:} \quad \text{IF } x_1 \text{ is } A_1^r \text{ and ... and } x_u \text{ is } A_u^r \text{ THEN } y \text{ is } B^r$$

where x_i (i=1,..., u) are the input and y is the output variables while A_i^r (r=1, ..., R) and B^r are, respectively, the antecedents' and consequent's fuzzy sets that represent linguistic values like "very big", "small", etc. Many times, the information about the output variable is exact (not fuzzy) so that the consequent's fuzzy sets can be simplified to singleton values, c_r, like "1.5", "0.5", etc. These systems, known as zero-order Sugeno's type or singleton fuzzy systems, are widely used because of their simplicity [1-3]. They infer a crisp output by implementing the following weighted average:

$$y = \sum_{r=1}^{R} h_r \cdot c_r / \sum_{r=1}^{R} h_r \tag{1}$$

where the weight h_r is the activation degree of the r-th rule, usually calculated as the minimum or product of the input variables' membership degrees to their corresponding fuzzy sets. This type of inference is shown in the block diagram of Figure 1.

The global structure of a singleton fuzzy system is defined by the set of IF-THEN rules that connect the antecedents' fuzzy sets which cover the input spaces with the consequent's singleton values of the output space. The detailed structure is defined by the membership functions that represent the antecedents' fuzzy sets and by the consequent's singleton values. These structures are

usually obtained from the knowledge of human experts. However, as processes become more complex (highly nonlinear and/or changing over time) this knowledge can be too uncertain or insufficient. This is why a current research area is the development of adaptive fuzzy systems which can automatically find their structure using linguistic information and learning algorithms, most of them taken from the neural network domain [4-6]. The block diagram of an adaptive fuzzy system is shown in Figure 2. The fuzzy inference engine implements equation (1) once the structure of the fuzzy system is defined. The adaptive loop contains a performance monitor that measures the quality of the performance and sends this information to the tuning/learning block. This block implements the learning algorithm that updates the system structure to match the new process state.

Fuzzy systems have been widely implemented with software on standard digital processors. However, when real-time operation and/or low area and power consumption are required the adequate solution is to implement them with dedicated hardware (known as fuzzy chips). The knowledge of the work environment where the fuzzy system is embedded is helpful to design fuzzy chips that can save additional circuitry for signal conditioning between blocks. For the case of adaptive fuzzy systems and according to Figure 2, it is worthwhile that a fuzzy chip accepts analog input signals to directly communicate with the real world, and provides the output in either a digital or analog format to directly interact with a subsequent digital stage (like a computer as a performance monitor) or analog devices (like analog actuators). The parameters that define the structure of the fuzzy inference engine have to be programmable to be tuned by the adaptive loop. Digital programmability is preferred since it offers a simple and safe mechanism for storing the programmable parameters, and it also gives an adequate interface for a manual or automatic digital tuning. On the other side, if the tuning/learning mechanism is implemented with analog circuitry, as happens in analog neural networks, the fuzzy chip should also admit analog programming signals.

This paper describes a design approach of CMOS fuzzy chips capable of performing as adaptive fuzzy systems with all these interface capabilities. The required building blocks are designed as mixed-signal current-mode circuits. They are described in Sections II and III. These circuits have been integrated into a complete 2.4-μm CMOS fuzzy chip that implements 9 rules. Experimental results of this prototype are provided in Section IV. Application of the proposed circuitry to implement adaptive fuzzy systems is discussed and illustrated with an example in Section V. Finally, conclusions are given in Section VI.

II.- CMOS building blocks for each rule processing.

The first operation to perform within each rule processing is to define the antecedents' fuzzy sets by adequate membership functions. The strategy we follow to generate the usually employed triangular/trapezoidal membership functions is to implement the following expression [7]:

$$\mu\,(I_{in}) = I_{\mu} = I_{ref}\ \Theta\ m_b \bullet [\ m_a \bullet (\ I_{aux}\ \Theta\ I_{in}\ ) + (\ I_{in}\ \Theta\ I_{aux}\ )\ \Theta\ I_{sat}\ ] = I_{ref}\ \Theta\ \bar{I}_{\mu} \qquad (2)$$

that can be reduced to:

$$I_{\mu} = I_{ref}\ \Theta\ m_b \bullet (\ |\ I_{in} - I_{aux}\ |\ \Theta\ I_{sat}\ ) \qquad (3)$$

if m_a is equal to 1 (which means a symmetrical membership function).

In the above expressions, I_{in} is the input signal and the parameters $\{I_{aux}, I_{sat}, m_a, m_b\}$ are related, respectively, to the position of the membership function on the universe of discourse, to its saturation, and to its two slopes, as shown in Figure 3a. The signal I_{ref}, which has a constant value in the fuzzy chip, represents the maximum membership degree and Θ is a rectification or bounded

difference operator, defined as:

$$a \,\Theta\, b = \begin{cases} a - b & \text{if } a > b \\ 0 & \text{otherwise} \end{cases}$$

The block diagram of the generic membership function circuit (MFC) that realizes equation (2) is shown in Figure 3b. The D/A blocks represent digitally-programmable current mirrors whose output currents are obtained by multiplying their input currents by a factor that depends on the geometry of the current mirrors and the digital word that controls the switches. They can be seen as low-resolution current-mode D/A converters so that the programmable parameters that define the antecedents' membership functions can be stored in efficient digital memories inside the chip. The D/A blocks not only perform as converters but also as computing blocks. The D/A blocks associated with m_a and m_b implement the scaling operations '$\bullet m_a$' and '$\bullet m_b$'. The block associated with m_b also implements the operation '$\Theta\, I_{sat}$' exploiting the diode-connected transistor at its input stage. Advantages of current-mode analog processing are that addition is carried out by wire connection and the operations '$I_{aux} \,\Theta\, I_{in}$' and '$I_{in} \,\Theta\, I_{aux}$' are simply implemented by the transistors T_1 and T_2 [7].

Once the membership degrees of the input signals to their corresponding fuzzy sets have been obtained, the activation degrees of the rules (represented by the currents I_{hr}) are calculated by connecting them with a T-norm like a minimum operator (Min). This can be done by using the multi-input maximum circuit proposed in [8] and the De Morgan's laws, as shown in Figure 4:

$$I_{hr} = Min(I^r_{\mu i}) = \overline{Max}(\overline{I^r_{\mu i}}) = I_{ref} \Theta Max(\overline{I^r_{\mu i}}) \tag{4}$$

The MFC's employed ease realization of the above equation because they can provide the value of $\overline{I}^r_{\mu i}$ by eliminating the last current mirror that performs the bounded difference operation with I_{ref} (shown with grey lines in Figure 3b). The bounded difference, $I_{ref} \,\Theta\, Max(\overline{I}^r_{\mu i})$, is performed after each maximum circuit, with the block called COMPL in Figure 4.

The next operation is to implement the product $I_{hr} \bullet c_r$ to calculate the partial conclusion of each rule. This is carried out by circuits, which will be called CONS's, that receive the currents representing the activation degrees of the rules, I_{hr}, and weight them by their corresponding singleton values, c_r. They are digitally-programmable current mirrors like the D/A blocks of the MFC's so that the programmable singleton values can be also stored in a digital memory inside the chip.

A fuzzy inference chip implemented with the circuits described above (MFC's, MAX's, COMPL's, and CONS's) accepts digital programming signals to fix the antecedents and consequents of its rule base. An example is the 2.4-μm CMOS prototype whose die photograph is shown in Figure 5a. This chip has two inputs that can be represented by voltages (Vx, Vy) or currents (Ix, Iy) because the MFC's previously described contain the V-I converter proposed in [9]. The input spaces are covered by three fuzzy sets (*N*, *Z*, *P*) whose membership functions (which are symmetrical) are defined by 8-bit digital words: 5 bits, $\{a_0\ a_1\ a_2\ a_3\ a_4\}$, for the position and 3 bits, $\{b_0\ b_1\ b_2\}$, for the slopes. This means that the MFC's contain a 5-bit and a 3-bit D/A blocks. The current I_{aux} related to the position parameter is provided by the 5-bit D/A block as follows:

$$I_{aux} = (a_0 + 2 \cdot a_1 + 4 \cdot a_2 + 8 \cdot a_3 + 16 \cdot a_4) \cdot I_u \tag{5}$$

where I_u is the input current of the D/A block. The 3-bit D/A block associated with the slope parameter, $m=m_b$ ($m_a=1$), provides the following output current according to equation (3) and Figure 3b:

$$\overline{I_{\mu}} = m \cdot (|I_{in} - I_{aux}| \Theta I_{sat}) = (b_0 + 2 \cdot b_1 + 4 \cdot b_2) \cdot (|I_{in} - I_{aux}| \Theta I_{sat}) \quad (6)$$

The output space is covered by three singleton values (*n, z, p*) that are programmed by 4 bits, $\{d_0\ d_1\ d_2\ d_3\}$. This means that the CONS's are 4-bit D/A blocks. They calculate the partial conclusions of the rules as follows:

$$I_{hr} \cdot c_r = I_{hr} \cdot \left(\frac{1}{2} \cdot d_0 + \frac{1}{4} \cdot d_1 + \frac{1}{8} \cdot d_2 + \frac{1}{16} \cdot d_3\right) \quad (7)$$

so that the singleton values, c_r, range from 0 to 0.9375.

The connection of the different blocks in this chip and the rule table implemented are illustrated in Figures 5c and d, respectively. Figure 5b shows the relative area occupation of the different blocks. The active area of this chip is 1 x 1.1 mm^2.

III.- CMOS Blocks for the I/O and programming interface.

The operation of an adaptive fuzzy chip is typically divided into two modes: programming and processing, which are controlled by two non-overlapping signals. During programming mode, the programmable parameters (in our case: the position, saturation and slopes of the antecedents' membership functions and the singleton values of the consequents) are updated if required. To provide the fuzzy chip with an analog programming interface, we propose to add an A/D block to each D/A block of the MFC's and CONS's, as illustrated in Figure 6 (for simplicity, the MFC's shown in this figure generate symmetrical and triangular membership functions). Hence, if a programmable parameter is given in analog format, for instance it is represented by a current, I_{dat}, it is converted to a binary code $\{b_k\}$ by its corresponding A/D block. This binary code verifies in general that:

$$\Sigma_{k=1}^{M} b_k 2^{-k} = Q\left(\frac{I_{dat}}{I_{ad}}\right) \quad \text{or} \quad \Sigma_{k=1}^{M} b_k 2^{k} = Q\left(\frac{I_{dat}}{I_{ad}}\right) \quad (8)$$

depending on whether the A/D block is of dividing (left equation) or multiplying type (right equation). M is the resolution of the A/D block, I_{ad} is its reference current during the programming mode, and Q(.) is a quantization operator. This binary code so obtained is stored as a digital word in the corresponding memory.

During processing mode, the parameters written in the digital memories are employed to implement the rule processing of the fuzzy inference method. The D/A blocks of the MFC's associated with the position, saturation and slope of the antecedents' membership functions as well as the D/A blocks (the CONS's) associated with the singleton values take the corresponding binary codes $\{b_k\}$ stored in the digital memories and provide in general the following output current:

$$I_o = I_{da} \cdot (\Sigma_{k=1}^{M} b_k 2^{-k}) \quad \text{or} \quad I_o = I_{da} \cdot (\Sigma_{k=1}^{M} b_k 2^{k}) \quad (9)$$

where M is the resolution of the D/A block (dividing or multiplying type) and I_{da} is its input current during the processing mode. If the binary code was obtained during the programming mode from an analog parameter, I_{dat}, this means (by combining equations (8) and (9)) that:

$$I_o = I_{da} \cdot Q\left(\frac{I_{dat}}{I_{ad}}\right) \quad (10)$$

In the A/D-D/A blocks that fix the position and saturation of the membership functions, the input current of the D/A part, I_{da}, during the processing mode is equal to the reference current of the A/D part, I_{ad}, which was employed during the programming mode ($I_{da} = I_{ad} = I_u$ in Figure 6). Hence, if the programmable parameter was given as the current I_{dat} ($I_{dat} = I_{aux}$ or I_{sat}), the parameter employed during rule processing is its quantized value: $I_u \cdot Q(I_{dat} / I_u)$, according to equation (10). Regarding the A/D-D/A blocks that fix the slopes in the MFC's and the A/D-D/A blocks that perform as CONS's, the input current of the D/A part is not constant during the rule processing but a signal of the inference process. According to Figure 6 and considering analog programmable parameters, the output currents of these blocks are:

$$I_{o\overline{MFC}} = \left| I_{in} - I_u \cdot Q\left(\frac{I_{aux}}{I_u}\right) \right| \cdot m = \left| I_{in} - I_u \cdot Q\left(\frac{I_{aux}}{I_u}\right) \right| \cdot Q\left(\frac{I_m}{I_v}\right) \tag{11}$$

$$I_{oCONS} = I_{hr} \cdot c_r = I_{hr} \cdot Q\left(\frac{I_{cr}}{I_w}\right) \tag{12}$$

where I_{aux}, I_m and I_{cr} are the analog parameters that program, respectively, the position and the slope of the antecedents' membership functions and the singleton values, while I_u, I_v and I_w are the reference currents of the A/D parts during the programming mode.

An A/D-D/A block is also employed during processing mode to implement a current-mode divider (DIV), as shown in Figure 6. The numerator current (the input current to the A/D part in DIV) has the value $\Sigma_r I_{hr} \cdot c_r$, while the value of the denominator current (the reference current of the A/D part in DIV) is $\Sigma_r I_{hr}$. The analog output of this divider, which is the crisp output of the fuzzy inference singleton method, is given by:

$$I_{out} = I_p \cdot Q\left(\frac{\Sigma_r I_{hr} \cdot c_r}{\Sigma_r I_{hr}}\right) \tag{13}$$

with a quantization error of $\pm \Sigma_r I_{hr} / 2^{(N+1)}$, where N is the resolution of the data converters and I_p is the input current of the D/A part.

The chip that results when employing these A/D-D/A blocks can be combined with digital or analog circuitry that implements the tuning/learning mechanism. Besides, it is able to interact directly with analog sensors and actuators or digital subsequent stages. The chip prototype commented above offers this latter advantage because it contains a 5-bit A/D-D/A divider (the DIV block in Figure 5b).

Current-mode techniques permit efficient implementation of these A/D-D/A blocks. As an example to illustrate the area occupation of these blocks, the 5-bit A/D-D/A divider integrated in the CMOS prototype occupies 528 μm x 145 μm (0.077 mm^2). Among algorithmic, semialgorithmic or binary-weighted current source converters, the first ones greatly simplify the design since they are very modular. They are based on a cascade of identical bit cells [10] or alternating odd and even bit cells [11], as shown in Figure 7a. Taking into account that no high precision is required in fuzzy systems, we have chosen continuous-time algorithmic converters to reduce the control circuitry. Considering the divider block, a dividing A/D conversion technique (Figure 7b) is more convenient than a multiplying type since it enables the reference current, $I_{ad}=\Sigma_r I_{hr}$, to be the full-scale current.

We have analyzed different structures [12] to design the circuitry of the A/D bit cells, following the work in [10-11] [13]. The structure selected, shown in the upper part of Figure 8a, offers a good trade-off between speed and power consumption. It employs the current comparator proposed in [14-15], which is exploited to also carry out the subtraction of $I_{ad}/2^i$ from I_i in the i-th bit cell. The capacitive coupling between the nodes *a* and *b* and the high-speed operation of the current

comparator may cause glitches in the bits generated. They are reduced by inserting the nMOS transistor shown with grey lines in the upper part of Figure 8a. Regarding the design of the D/A part, good dynamic behavior is achieved by choosing all the bit cells identical (to reduce nonsymmetrical switching) and synchronizing the different bits via digital latches. Switching speed is increased if there is always a conducting path for the current. This is the purpose of the pMOS transistor shown with grey lines at the bottom part of Figure 8a.

The schematic illustrated in Figure 8a corresponds to the bit cell of a generic A/D-D/A block that can perform as a DIV, CONS or as the A/D-D/A block associated with the slope parameter in the MFC's. Considering the last two cases, the D/A part is operative during the processing mode, when the switches R are closed and the stored parameters are read from the digital memories. During programming mode, the programmable parameters are written into the digital memories. For this purpose, the switches R are open and either switches D•S (if the programmable parameter is digital) or A•S (if it is analog) are closed. In the last case, the switches W are closed conveniently so that the A/D part is operative.

The schematic shown in Figure 8b corresponds to the bit cell of a more specific A/D-D/A block that can only perform as the block associated with the position or saturation parameters in the MFC's. These blocks are simpler because the D/A is exploited as a part of the A/D (an A/D converter contains in general a D/A one) taking advantage of that the reference currents (I_{ad}, I_{da}) to store and recover the analog parameter are equal ($I_{FS} = I_{ad} = I_{da}$) [16].

IV.- Experimental results.

The behavior of the MFC's, MAX's, COMPL's, CONS's, and the DIV circuit has been verified experimentally with the 2.4-μm CMOS fuzzy chip described above. This is possible because the chip contains many pads to test the signal at internal points.

Figure 9 shows three different coverings of the input spaces with the three fuzzy sets (*N*, *Z*, *P*). In these experiments, the input variables were represented by currents (Ix in Figure 9), the maximum degree of membership that is fixed by the current I_{ref} was 16 μA and the bits of the MFC's were programmed so that the positions and the slopes were as follows:

- Figure 9a: $I_{aux}(N) = 8$ μA, $I_{aux}(Z) = 16$ μA, $I_{aux}(P) = 24$ μA; $m(N) = 2, m(Z) = 2, m(P) = 2$.
- Figure 9b: $I_{aux}(N) = 4$ μA, $I_{aux}(Z) = 16$ μA, $I_{aux}(P) = 28$ μA; $m(N) = 2, m(Z) = 1, m(P) = 2$.
- Figure 9c: $I_{aux}(N) = 12$ μA, $I_{aux}(Z) = 16$ μA, $I_{aux}(P) = 20$ μA; $m(N) = 2, m(Z)=1, m(P)=2$.

Experimental results that illustrate the calculation of the rules' activation degrees are shown in Figure 10. In these experiments, the input variables were represented by voltages, Vx and Vy. The results of Figure 10a were obtained by making equal the two input voltages and sweeping them along their ranges. These results correspond to the minimum operation performed between a given *Z* and differently programmed *P* fuzzy sets, that is, they represent the activation degree of a rule like the following: "If Vx is *P* and Vy is *Z* then ...". The results of Figure 10b also correspond to the minimum operation performed between a given *Z* and a given *P* fuzzy sets. In this case, they were obtained by sweeping Vy and fixing Vx to different constant values (3.45 V, 3.6 V, 3.75 V, and 3.9 V).

Figure 11a shows how the values of the rules' activation degrees are scaled by different singleton values (0, 0.25, 0.5, 0.75, and 0.9375) at the 4-bit CONS blocks. According to equation (7), these singleton values correspond, respectively, to the following programmable bits $\{d_0\ d_1\ d_2\ d_3\}$: {0 0 0 0}, {0 1 0 0}, {1 0 0 0}, {1 1 0 0}, and {1 1 1 1}. Figure 11b illustrates the dynamic behavior of the fuzzy chip before the DIV block. This figure represents the change in the value $I_{hr} \bullet c_r$ of a rule when there is a step input that activates it (a low resistance was connected externally to convert

current to voltage). The response time to obtain the conclusion of a rule was measured less than 250ns.

Figures 12a, b, and c show the analog output current provided by the divider circuit working as the final block of the fuzzy chip when sweeping the two input signals of the chip (represented as currents in the three cases). The antecedents' membership functions were programmed as shown in Figures 9a, b, and c, respectively, while the consequent's singleton values {*n*, *z*, *p*} were {0.9375, 0, 0.9375} for Figure 12a and {0, 0.4375, 0.875} for Figures 12b and c. They represent different control surfaces provided by the fuzzy chip. In these cases, the static power consumption of the chip ranged from less than 9 mW (surfaces in Figures 12 b and c) to 21 mW (Figure 12a) at a 5-V power supply and a 16-μA for I_{ref}. Since two of the test pads of the chip allow us to monitor the currents $\Sigma_r I_{hr} \cdot c_r$ and $\Sigma_r I_{hr}$, we could measure that the error introduced by the DIV block was limited to ± 0.47 LSB, thus verifying its 5-bit resolution. Since the DIV block is of 5 bits while the CONS's are of 4 bits, the maximum output digital code obtained in the situations summarized in Figures 12b and c can represent the values 0.84375, 0.875, or 0.90625, which correspond to the 5-bit quantization of the maximum output value: $(\Sigma_r I_{hr} \cdot c_r / \Sigma_r I_{hr})|_{max} = c_r|_{max} = 0.875$. This was confirmed experimentally: the maximum digital code measured was 0.90625 and, consequently, the output current appeared quantized into 30 levels.

Although the A/D-D/A divider integrated within this fuzzy chip does not contain all the design improvements commented in Section III, the response time of this fuzzy chip was measured less than 2 μs, which translates in more than 0.5 MFLIPs (mega fuzzy logic inferences per second). Hence, the divider is the block that limits the inference speed. Figure 13 shows different discrete analog outputs provided by the fuzzy chip for a step signal at one of the inputs and different constant values at the other. Better design of the divider and taking advantage of modern submicron CMOS processes would increase the total inference speed over the MFLIPs.

Several A/D-D/A quantizer memory blocks have been also integrated alone in a 2.4-μm CMOS process. The experimental results shown in Figure 14 correspond to the conversion and reproduction of an input analog current with 3-, 4-, and 5-bit resolution, which have been the resolutions employed in the CMOS prototype. It was measured that the quantization error never exceeded LSB.

V.- Applications.

Fuzzy systems have drawn a great attention for their capability of translating expert knowledge expressed by linguistic rules into a mathematical framework. This has been particularly exploited in the field of heuristic control where many processes that are too complex to be described with an adequate analytical model are properly controlled by an expert human operator [4] [17]. In addition, fuzzy systems as simple as zero-order Sugeno's type or singleton fuzzy systems have been proved to be universal approximators and, hence, potentially capable of controlling or describing any process [18]. This has opened the way for other types of applications like generation of linearizing or conditioning functions for nonideal sensors or signal predistortion. Adaptive fuzzy systems are more versatile since they can cope with highly nonlinear and dynamical processes. In this sense, they can be applied to many other fields like control and identification of nonlinear dynamical processes, nonlinear channel equalization or adaptive noise cancellation.

To analyze the suitability of the proposed circuitry for implementing adaptive fuzzy systems, we have studied different applications. One of them is the on-line identification of nonlinear dynamical plants. For this application, the flow diagram of the configuration is shown in Figure 15a. The nonlinear plant considered (usually reported in the literature [5]) is governed by the following difference equation:

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g[u(k)] \tag{14}$$

where the unknown function, g(.), has the form:

$$g[u] = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \tag{15}$$

The following series-parallel model is used to identify the plant:

$$y_d(k+1) = 0.3y(k) + 0.6y(k-1) + F[u(k)] \tag{16}$$

where F[u(k)] is the response of an adaptive fuzzy system.

This configuration has been simulated with C language considering that the adaptive fuzzy system is implemented as a one-input one-output fuzzy chip with 18 rules, designed as proposed in this paper. Its MFC's have been considered to provide triangular membership functions that cover the input space uniformly, as shown in Figure 15b. The adaptive chip is tuned by only learning the 18 singleton values of its rules. For this purpose, the performance monitor measures the difference between the plant's and the model's outputs, $y(k) - y_d(k)$, at each time step, k. During training, the input to the plant and to the model was a sinusoid u(k)=sin(2πk/40) and initially, the 18 singleton values were zero. The tuning/learning block receives the measure of the performance monitor and provides the adequate changes of the singleton values as:

$$\Delta c_r = -\zeta \cdot (y - y_d) \cdot h_r \qquad r = 1, ..., 18 \tag{17}$$

which corresponds to a gradient-descent learning algorithm with a ζ learning rate.

The tuning block can be implemented with analog or digital circuitry (for instance with a standard microprocessor) because the CONS circuits of the adaptive fuzzy chip are A/D-D/A blocks. A resolution of 4 bits have been proven enough to obtain efficient results. This is illustrated in Figure 15c where the grey and black lines represent, respectively, the model's and the plant's outputs for the input u(k)=sin(2πk/250) for $0 \le k \le 250$ and $501 \le k \le 700$, and u(k)=0.5sin(2πk/250)+0.5sin(2πk/25) for $251 \le k \le 500$. This proper identification was obtained after 70 time steps of training.

The circuitry proposed in this paper to implement adaptive fuzzy chips is based primarily on current mirrors so that the achieved resolution is limited ultimately by mismatching errors. Resolution below 8 bits are easily obtained with standard digital technologies [19]. Hence, low-resolution (low-cost) realizations of adaptive fuzzy chips are feasible for applications like this here described.

VI.- Conclusions.

Mixed-signal building blocks based primarily on current mirrors have been described and discussed for implementing adaptive fuzzy systems. They provide the chip with both analog and digital outputs, and with a mixed-signal programming interface. These techniques enables low-cost and high-inference CMOS realizations of adaptive fuzzy chips, as proven by measurements on a 9-rule 2.4-μm prototype. This chip accepts two analog inputs represented by currents or voltages and 8- and 4-bit digital words to program, respectively, the antecedents and consequents of its rule base. Its output is provided as either a 5-bit digital word or an analog signal. It occupies an active area of about 1 mm^2, consumes less than about 20 mW for a 5-V power supply, and offers

an inference speed of above 0.5 MFLIPs (although it was not optimized regarding speed). Fuzzy chips so designed can be successfully employed in several applications without requiring high resolution. This has been illustrated with an application example where 4 bits are enough.

VII.- REFERENCES

[1] Jager, R., Verbruggen, H. B. and Bruhx, P. M., "The role of defuzzification methods in the application of fuzzy logic", in *Proc. Symp. on Intelligent Components and Instruments for Control Applications*, (Malaga, Spain), pp. 111-116, 1992.

[2] Yamakawa, T., "A Fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control", *IEEE Trans. Neural Networks*, vol. 4, no. 3, pp. 496-522, May 1993.

[3] M. Sasaki, N. Ishikawa, F. Ueno and T. Inoue, "Current mode analog fuzzy hardware with voltage input interface and normalization locked loop", *IEICE Trans. Fundamentals*, vol. E75-A, no. 6, pp. 650-654, June 1992.

[4] Driankov, D., Hellendoorn, H. and Reinfrank, M., "*An introduction to fuzzy control*", Heidelberg: Springer-Verlag, 1993.

[5] Wang, L. X. and Mendel, J. M., "Back-propagation fuzzy systems as nonlinear system identifiers", in *Proc. Int. Conf. on Fuzzy Systems*, (San Diego, California), pp. 1409-1418, 1992.

[6] Lin, C.-T. and Juang, C.-F., "An adaptive neural fuzzy filter and its applications", *IEEE Trans. Syst., Man, and Cybern.*, vol. 27, no. 4, pp. 635-656, August 1997.

[7] Baturone, I., Sánchez Solano, S., Barriga, A. and Huertas, J. L., "Implementation of CMOS Fuzzy Controllers as Mixed-Signal IC's", *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 1, pp. 1-19, February 1997.

[8] Baturone, I., Huertas, J. L., Barriga, A. and Sánchez Solano, S., "Current-mode multiple-input max circuit", *Electronics Letters*, vol. 30, no. 9, pp. 678-680, April 1994.

[9] Bult, K. and Wallinga, H., "A class of analog CMOS circuits based on the square-law characteristic of an MOS transistor in saturation", *IEEE Jour. of Solid-State Circuits*, vol. 22, no. 3, pp. 357-365, June 1987.

[10] Nairn, D. G. and Salama, C. A. T., "Current-Mode Algorithmic Analog-to-Digital Converters", *IEEE Jour. Solid-State Circuits*, vol. 25, no. 4, pp. 997-1004, August 1990.

[11] Fong, K. L., and Salama, C. A., "Low-power current-mode algorithmic ADC", in *Proc. Int.'l Symp. Circuits Syst.*, (London, England), pp. 473-476, 1994.

[12] Baturone I., Sánchez-Solano, S. and Huertas, J. L., "A CMOS current-mode multiplier/divider circuit", to appear in *Proc. Int.'l Symp. Circuits Syst.*, (Monterey, California), May 31-June 3, 1998.

[13] Deval, P., "High resolution algorithmic A/D converters based on dynamic current memories", chapter 13 in *"Switched currents: an analogue technique for digital technology"*, Peter Peregrinus Ltd. 1993.

[14] Domínguez-Castro, R., Rodríguez-Vázquez, A., Medeiro, F. and Huertas, J. L., "High Resolution CMOS Current Comparators", in *Proc. ESSCIRC'92*, pp. 242-245, 1992.

[15] H. Traff, "Novel approach to high speed CMOS current comparator", *Electronics Letters*, vol. 28, no. 3, pp. 310-312, 1992.

[16] Baturone, I., Sánchez-Solano, S. and Huertas, J. L., "Self-checking current-mode analogue memory", *Electronics Letters*, vol. 33, no. 16, pp. 1349-1350, July 1997.

[17] Munakata, T. and Jani Y., "Fuzzy Systems: An Overview", *Communications of the ACM*, vol. 37, no. 3, March 1994.

[18] Castro, J. L., "Fuzzy logic controllers are universal approximators", *IEEE Trans. Syst., Man, and Cybern.*, vol. 25, no. 4, pp. 629-635, April 1995.

[19] Pelgrom, M. J., Duinmaijer, A. C. J. and Welbers, A. P. G., “Matching properties of MOS transistors”, *IEEE Jour. Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1440, Oct. 1989.

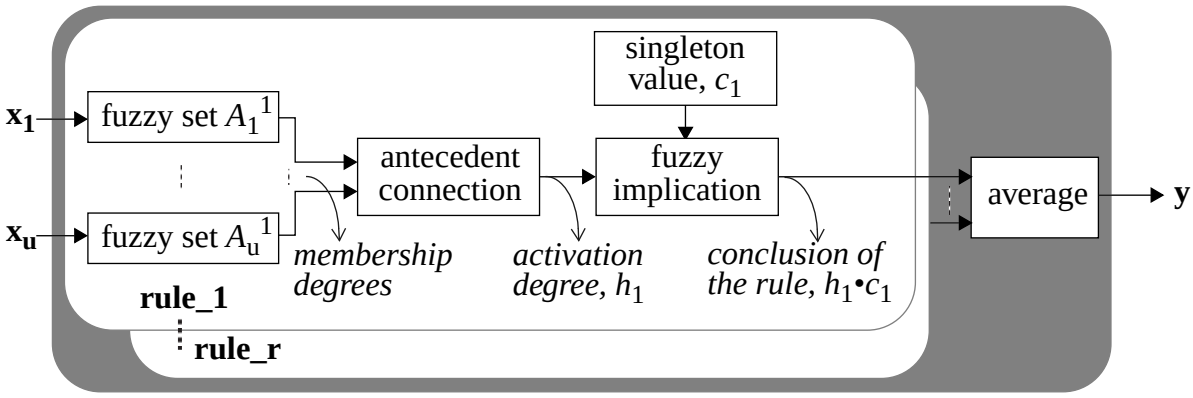


Figure 1: Block diagram of a singleton fuzzy inference engine.

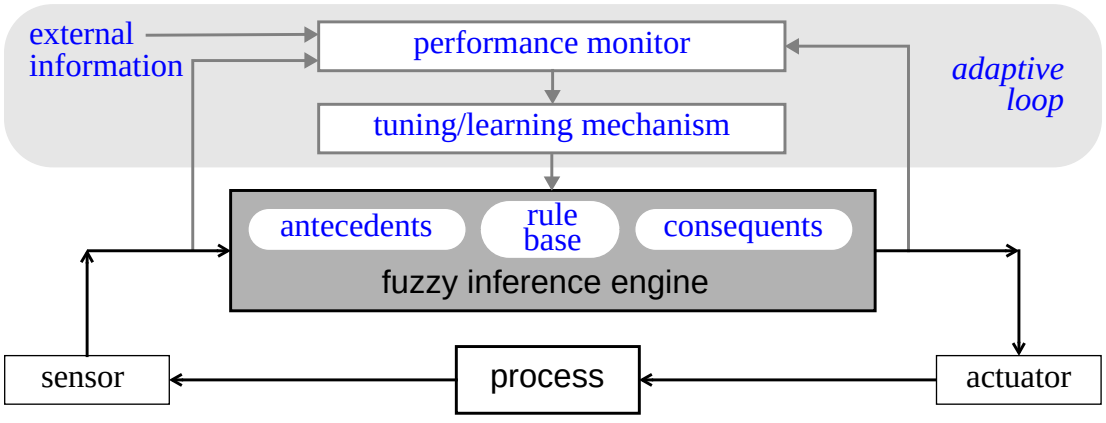


Figure 2: Block diagram of a typical adaptive fuzzy system.

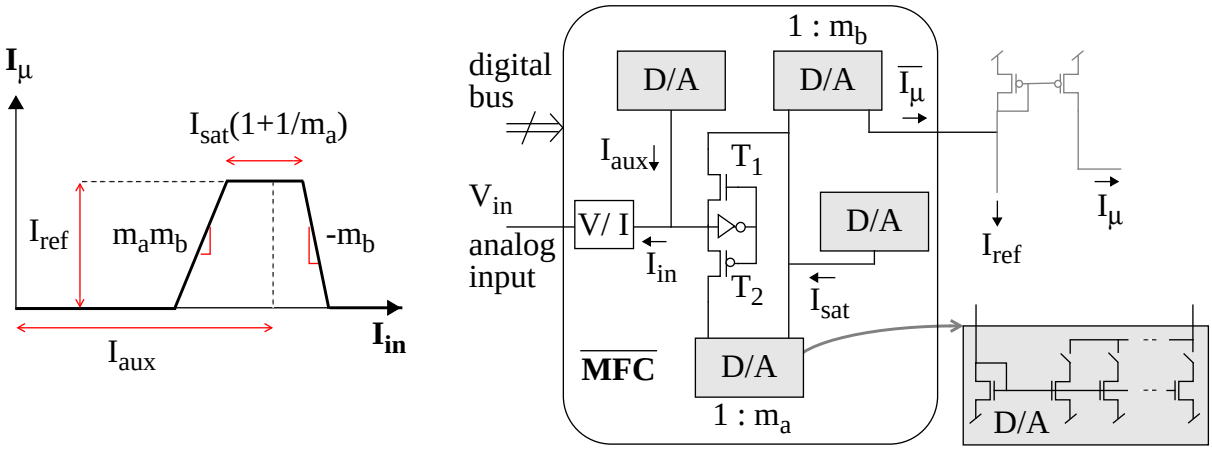


Figure 3: (a) Trapezoidal membership function provided by the circuit in (b).

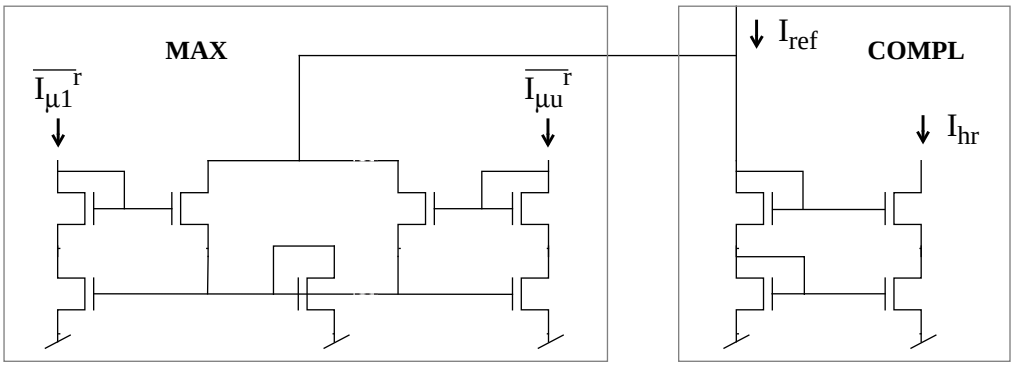
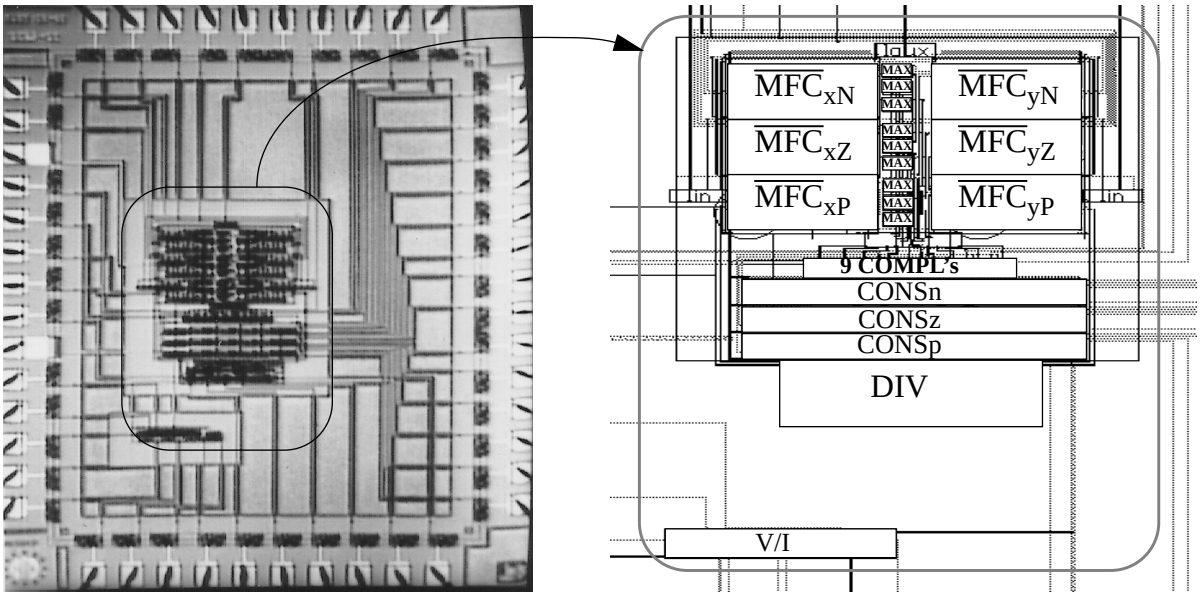
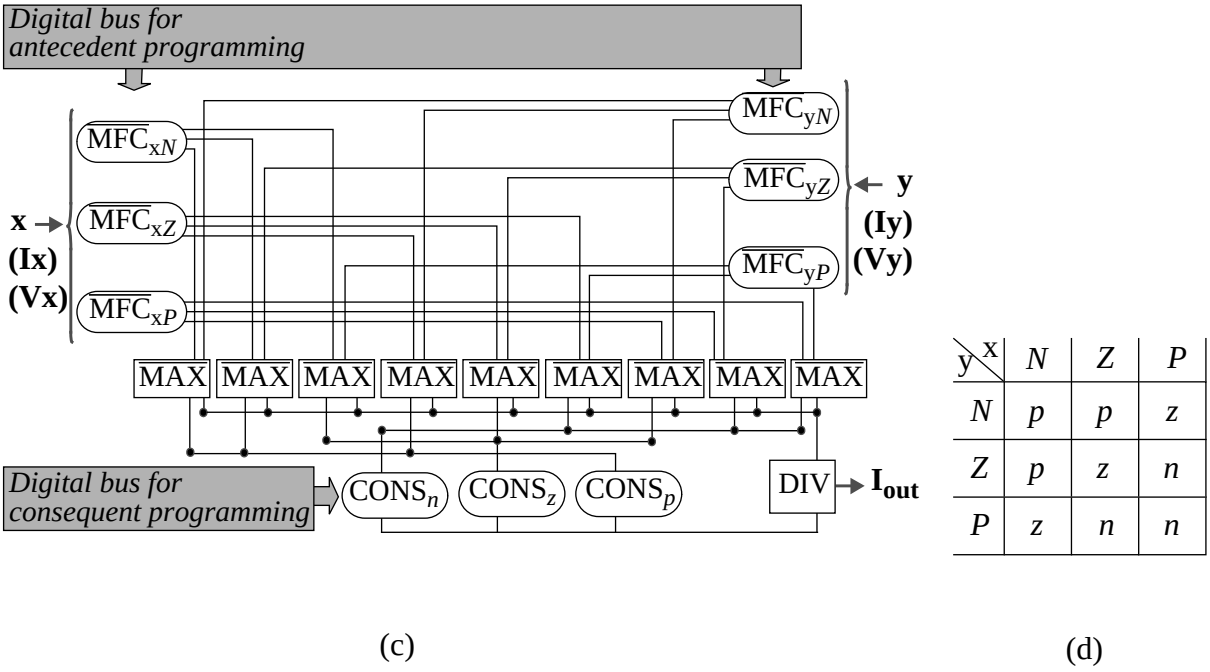


Figure 4: Circuitry to obtain the activation degree of a rule.



(a) (b)



y \ x	*N*	*Z*	*P*
N	*p*	*p*	*z*
Z	*p*	*z*	*n*
P	*z*	*n*	*n*

(c) (d)

Figure 5: (a) Die photograph of a CMOS fuzzy chip prototype. (b) Identification of the building blocks. (c) Architecture. (d) Rule base implemented.

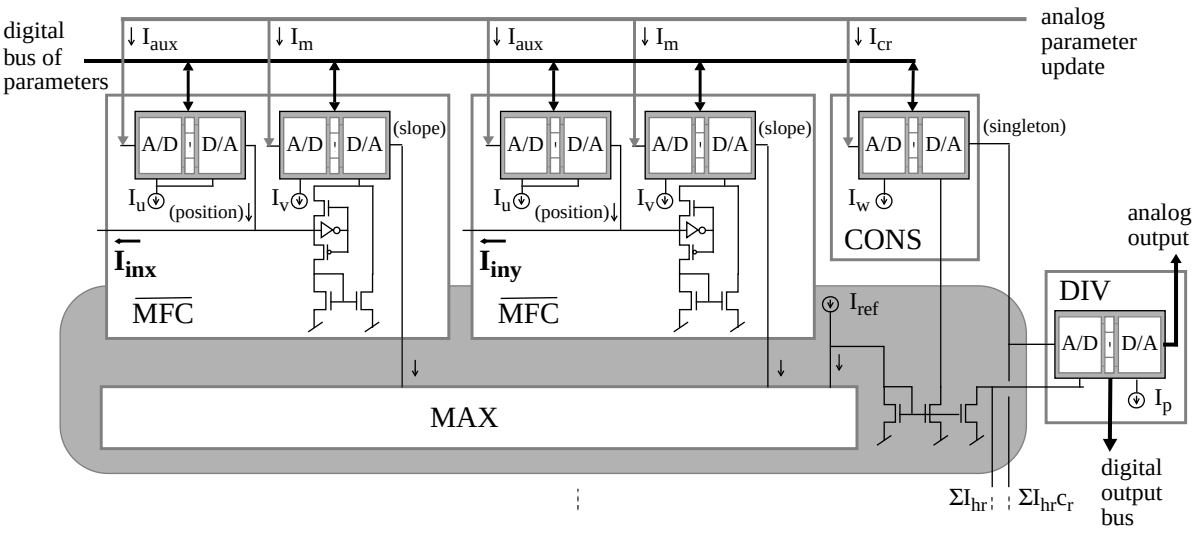
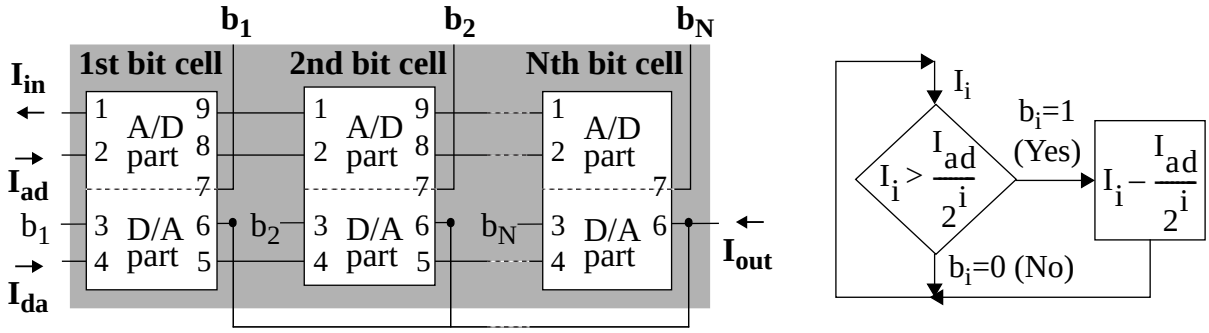
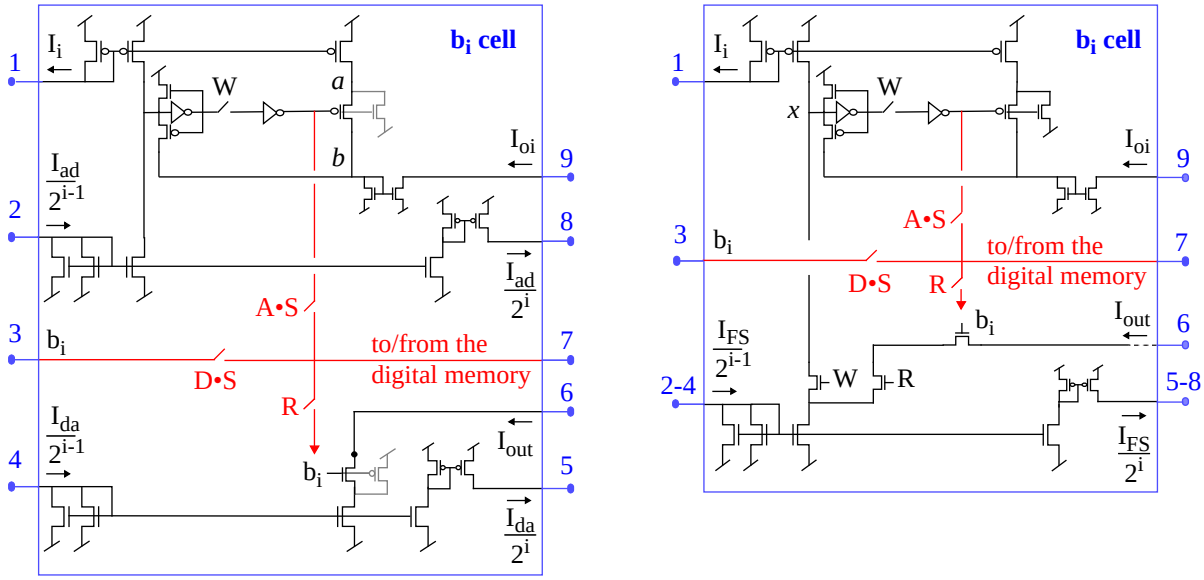


Figure 6: Block diagram of an advanced fuzzy chip.



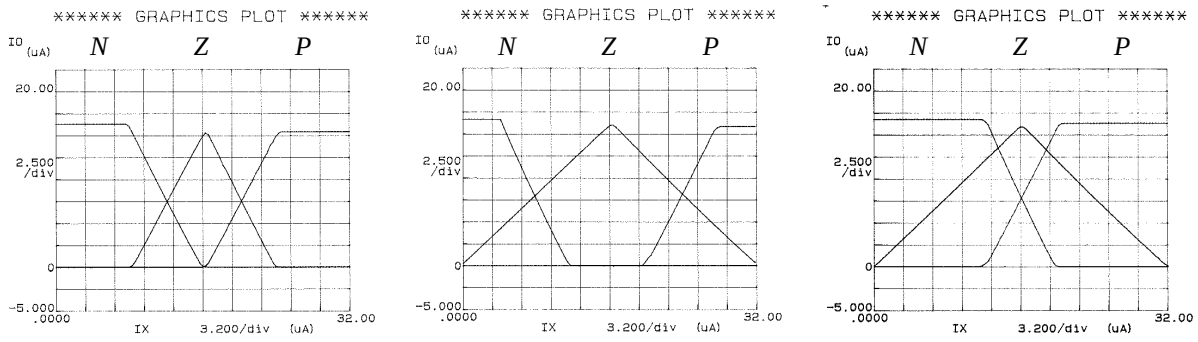
(a) (b)

Figure 7: (a) Modular design of a generic A/D-D/A block. (b) Flow chart of the dividing algorithmic A/D conversion method.



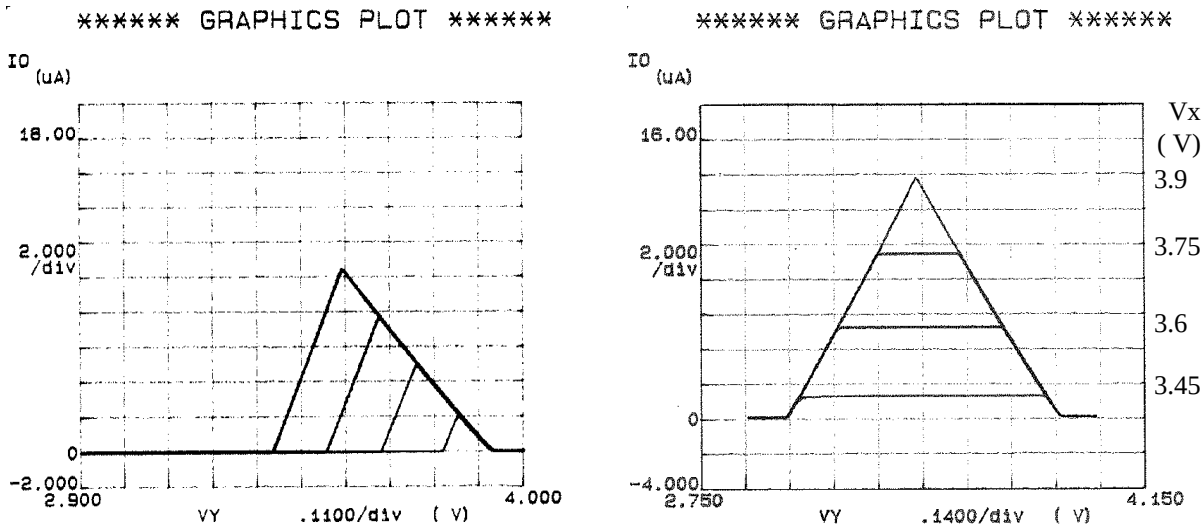
(a) (b)

Figure 8: Selected schematics of (a) a generic and (b) a simplified A/D-D/A bit cells.



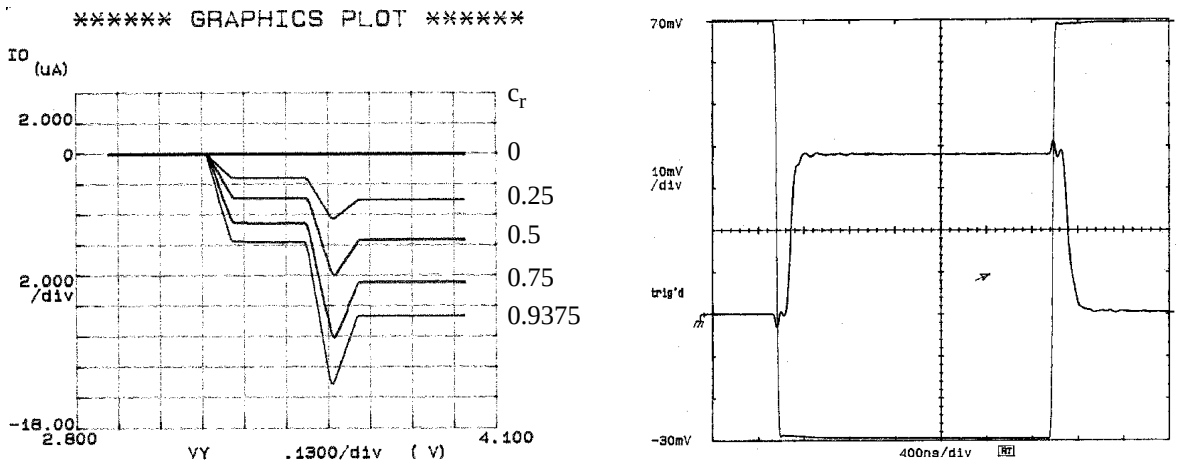
(a) (b) (c)

Figure 9: Experimental results that illustrate the programmability of the antecedents' membership functions.



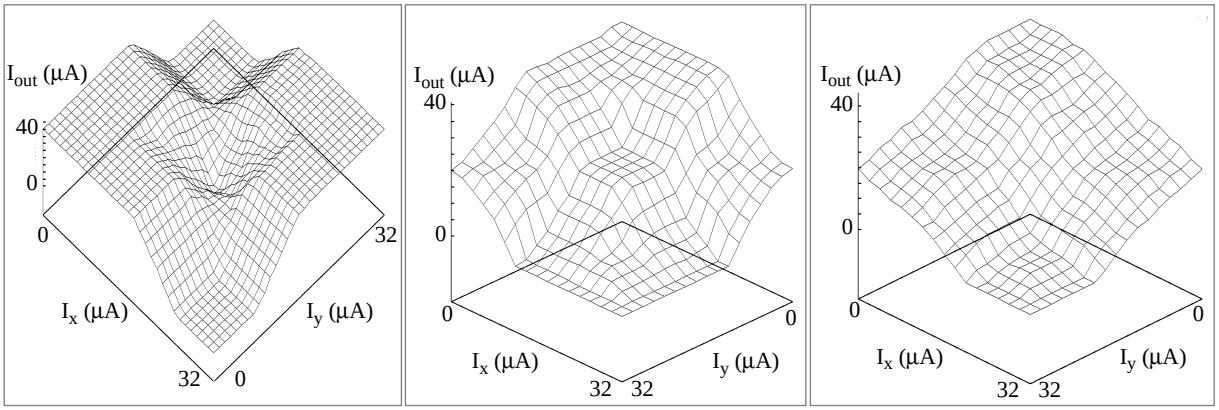
(a) (b)

Figure 10: Experimental results of the minimum operation.



(a) (b)

Figure 11: Experimental results that illustrate: (a) The programmability of the singleton values. (b) The response time to obtain the conclusion of a rule.



(a) (b) (c)

Figure 12: Different control surfaces measured from the fuzzy chip.

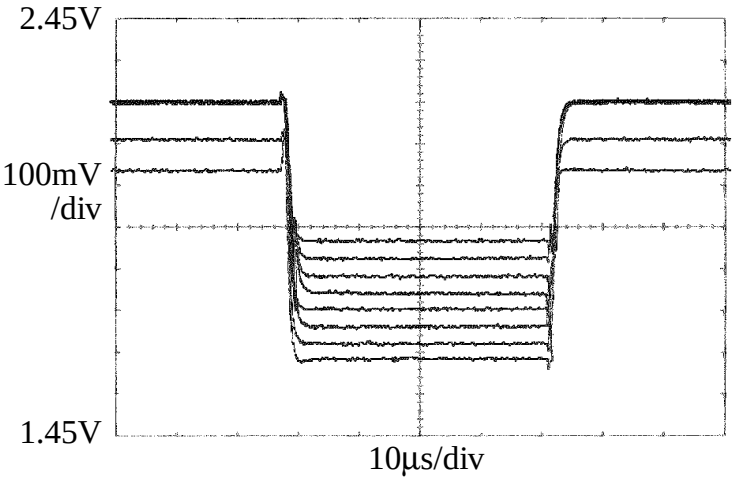
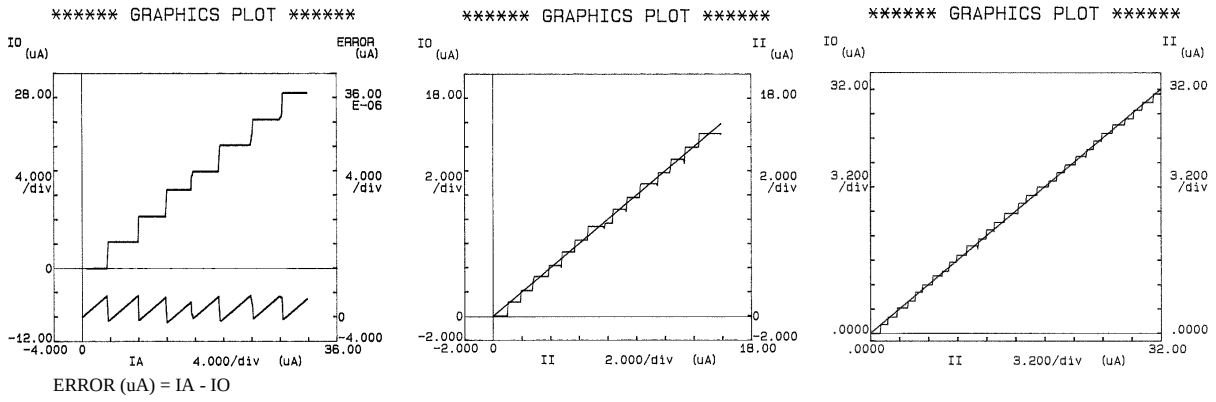


Figure 13: Several step responses of the discrete analog output of the fuzzy chip.



(a) (b) (c)

Figure 14: Experimental results of A/D-D/A blocks storing quantized values of an analog parameter with: (a) 3- (b) 4- and (c) 5-bit resolution.

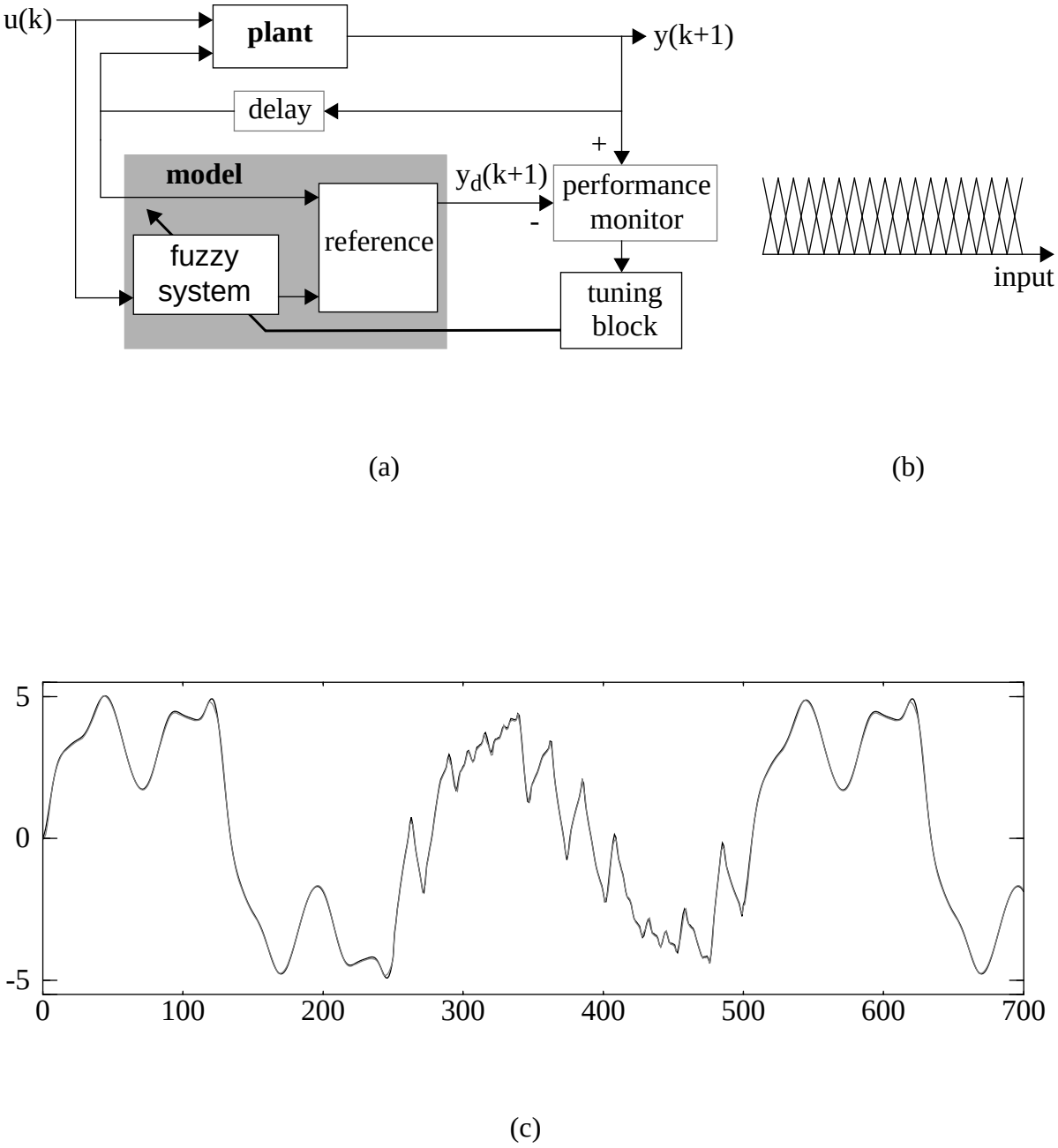


Figure 15: (a) Identification of a nonlinear dynamic plant. (b) Antecedents' fuzzy sets. (c) Output of the model (grey line) and the plant (black line) after the training process.