

A Mixed-Signal Fuzzy Controller and its Application to Soft Start of DC Motors

Rafael Navas-González

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

Fernando Vidal-Verdú

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

Angel Rodríguez-Vázquez

Dept.of Analog and Mixed-Signal
Circuit Design
C N M - Universidad de Sevilla
Edificio CICA, C/Tarfia s/n,
41012-Sevilla, SPAIN
angel@cnm.us.es

Abstract-This paper presents a mixed-signal fuzzy controller chip and its application to control of DC motors. The controller is based on a multiplexed architecture presented by the authors in [1], where building blocks are also described. We focus here in showing experimental results from an example implementation of this architecture as well as in illustrating its performance in an application that has been proposed and developed. The presented chip implements 64 rules, much more than the reported pure analog monolithic fuzzy controllers [3][4][5], while preserves most of their advantages. Specifically, the measured input-output delay is around 500ns for a power consumption of 16mW, and the chip area (without pads) is 2.65mm². In the presented application, sensed motor speed and current are the controller input, while it determines the proper duty cycle to a PWM control circuit for the DC-DC converter that powers the motor drive. Experimental results of this application are also presented.

I. INTRODUCTION

Analog implementations of fuzzy controllers are plausible because most control tasks do not need more than 10% of precision, and they can accomplish this goal in a more compact, faster and low power fashion than their digital counterparts [6]. However, such good performance is difficult to keep when complexity grows, due to the aggregation of errors as well as parasitics in the global computation nodes. Thus, analog implementation of large circuits usually involves tuning mechanisms [7][8].

In addition, another way to exploit the analog benefits while bounding the error and the dynamic response introduces digital circuitry that programs an analog core which fits the accuracy and speed/power requirements [1]. This approach is followed by the monolithic controller of this paper, whose I/O interface is analog, but internally performs mixed signal processing. The latter is intended to preserve the analog performance, thus the signal path is entirely analog, while the internal control circuitry is mixed.

On the other hand, applications of analog integrated circuits could require just compactness, but those that demand a good speed/power ratio are more suitable to be faced with them. Control of power systems is one of these applications [9]. Specifically, this paper presents a control example to perform a soft start of a DC electrical motor.

The problem of starting this kind of motors lies on preventing damage of the motor winding due to the large spikes that take place in the starting transitory. The latter occur in the range of microseconds [10], thus any control procedure should respond quickly, as analog circuits do.

A common open loop strategy forces a gradual change of some variable along time, thus it follows a ramp instead of a sharp step. This variable can be the supply voltage (positive slope ramp) or the impedance of a resistor in series with the

motor winding (negative slope ramp). However, the use of fast mixed controllers allows closed-loop strategies that perform better than the former. These strategies can take advantage of the sensing of speed and current in the control circuitry of many servo-motors to provide a proper increment of the controlled variable as a function of them.

The above strategy is implemented by the circuitry of this paper. The controlled variable is the supply voltage, which is provided by a DC/DC switching converter whose duty cycle is set by our controller. The so built control system is able to respond to changes in the current in the range of a few microseconds, thus a very efficient control is possible. It can tight the current close to a safety value, thus keeping the torque almost constant and speeding up the motor starting, which could increase the production of machines that perform massive ON/OFF operations, while enlarging their lives because of the current spike prevention.

Fuzzy controllers have already been used to control the speed of DC motors [11]. Such control tasks require around 25 rules. Simulations show that our starting procedure requires a similar or higher number of rules. Consequently, the mixed signal monolithic controller of this paper is a good choice to implement the DC motor soft-start procedure.

II. MIXED-SIGNAL FUZZY CHIP CONTROLLER

The chip presented in this paper is based on the multiplexing strategy reported by the authors in [1], where building blocks implementation is also covered. In the following, we first review the main features of this idea and give a brief description of the resultant architecture, then we describe the chip as it is used for control tasks.

A. Architecture and Functional Description

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

circuitry related to the active rules, then multiplex it dynamically to cover the whole input space.

Fig.2 shows the architecture for a controller that follows the previous approach, with M inputs, L labels per input (thus L^M rules), and S bits per singleton (since the singleton weighting is implemented with a digitally programmable current mirror, the singleton values are digital words). It comprises the following blocks:

- **A/D Converters:** Their function is to encode the interpolation interval C_{ij} associated with the current input. There are M , one per input, with a resolution equal to the next superior integer $\log_2 L$, i.e. $\text{int}_s(\log_2 L)$. Thus, this battery of converters provides a word of $M \lceil \text{int}_s(\log_2 L) \rceil$ bits that drives the *interval selector* block and the *digital memory* block.
- **Interval Selector:** It selects a set of voltage values $E_1, \dots, E_k, \dots, E_M$ to drive the analog core and, thus, makes it implement the active membership functions.
- **Digital Memory:** It selects the active singleton programming values $y_1^*, \dots, y_1^{*'}, \dots, y_{2^M}^*$ that configure the *rule block* of the *analog core* consequents of the active rules. These are digital words of as many bits as needed to encode the required set of singleton values.
- **Analog Core:** It performs the fuzzy computation having a set of programming inputs which are driven by the *interval selector* and the *digital memory* blocks. These inputs set up the analog core to work with the rule set that determines the system output, which means specifying the membership functions associated with the rule antecedents as well as the singleton values related to the consequents.

Note that Fig.2 is valid and even more useful for an increasing number of inputs and labels. The number of outputs can also be higher with little effort because many blocks can be shared by the circuitry dedicated to generate each output. Specifically, each additional output involves one digital memory block, and 2^M rule blocks.

B. Chip Description.

The fuzzy monolithic controller of this paper has the architecture of Fig.2 for $L = 8$, $M = 2$ and $S = 4$ and has

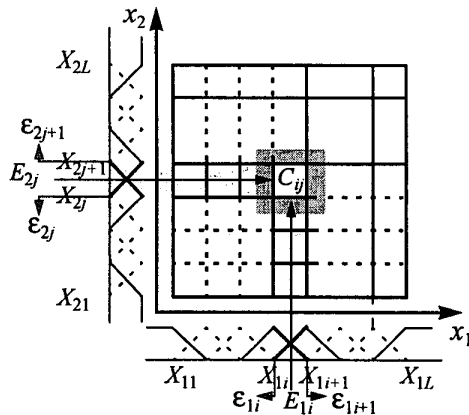


Fig. 1. Illustration of the active rule set

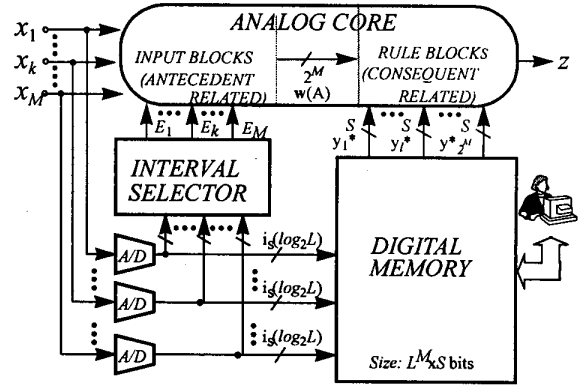


Fig. 2. General architecture of a controller with multiplexed analog core.

been implemented in a CMOS 0.7 μ m technology. Fig.3 shows the digital and analog interfaces with the pin signals grouped in buses. The digital interface corresponds to a typical asynchronous peripheral for microcontroller based systems, with an input address bus(A{0:5}), a bidirectional data bus(D{0:3}) and a control bus (COM, W-R). The controller inputs are x_1 and x_2 , which are driven by voltage mode signals in the range from 2V to 4.8 V, while it provides a current as output in the z_i pin in the range from 0 μ A to 150 μ A. Another output is generated in z_v for testing purposes by converting internally this current into a voltage with a polysilicon resistor, thus the output can be measured by a digital scope. The BIAS bus in Fig.3 groups four external bias currents IPH,IPN,IPG,IPV and a voltage VREF, needed for proper working of the analog circuitry. Although they are generated externally to ease the testing, they could be provided internally in a marketable final version.

Fig.4(a) shows a microphotography of the chip, where the die area is 5mm². However, it is determined by restrictions of the foundry, while the actual chip area consumption is smaller. Specifically, the summation of the building blocks area consumption is below 1mm². Fig.4(b) shows the floor plan of the chip with the different blocks and their sizes. Note that special care has not been taken to optimize the routing because there are not great area constraints.

C. Prototype Chip test results.

Fig.5 shows the block diagram of a prototype board that has

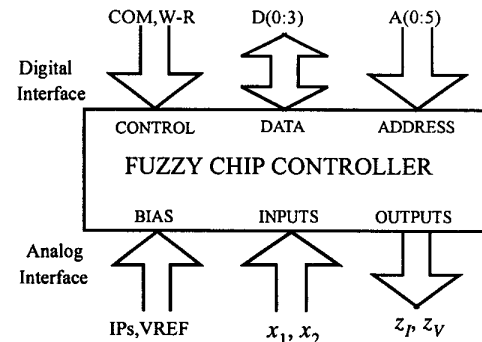


Fig. 3. Chip interface

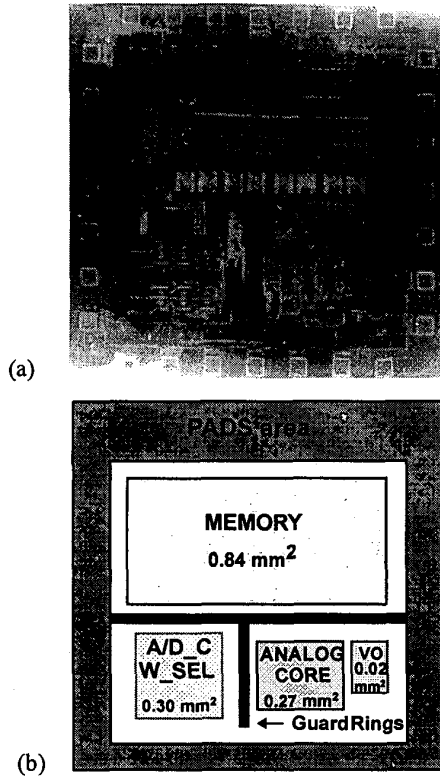


Fig. 4. Microphotography (a) and Floor plan (b) of the chip

been designed to test the chip. It comprises the circuitry needed to build the bias current sources, and that to interface to an 8255/8253 I/O Card for a PC. The PC interface provides read and write access to program the chip memory. It also provides write access to the D/A converters and read access to the A/D converter, thus the board works as a simplified data acquisition one. In addition, a TEKTRONIX TDS 520 digitizing oscilloscope is used to measure the dynamic response as well as to obtain sections of the dc control surface.

Fig.6 shows some experimental results obtained from this test environment. Fig.6(a) and Fig.6(b) depict the section of an example dc control surface and the transient response to a step input respectively, both measured with the oscilloscope. The former illustrates the response (bottom) to a ramp in one input (top) while the other input remains constant, in a kind of mexican hat surface. The latter corresponds to a falling edge in one input that forces the output to change from its maximum to its minimum value, as well as to jump to a different interpolation interval, which means a dynamic programming of the analog core. The measured delay time is around 500ns. Since the oscilloscope is not able to sense currents, previous measurements are voltages in the output z_V . With regard to Fig.6(c) and Fig.6(d), they are built with data from the data acquisition board in Fig.5, where the current output of the chip is externally converted into a digital word and processed. These two examples illustrate the ability of the controller to interpolate nonlinear (Fig.6(c)) as well as linear functions (Fig.6(d)).

On the other hand, the measured chip power consumption

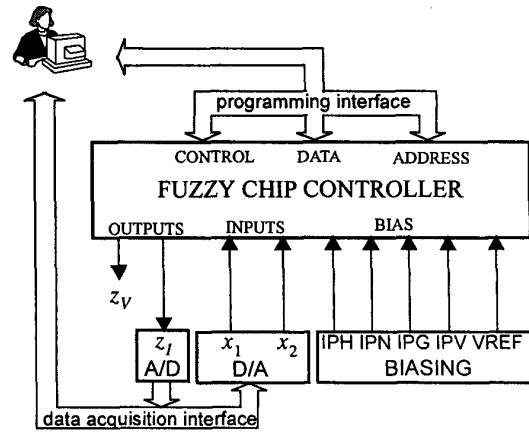


Fig. 5. Testing board block diagram

was around 16mW, which is obtained by sensing the current from the supply voltage source. The controller was designed to take into account mismatching among transistors for a standard deviation (σ) associated to the output that equals an error of 3.3%, while the measured errors were around 10%. The latter corresponds to a worst case and still fit most control requirements [12].

It is illustrative to compare the speed, area and power of this mixed-signal controller to that of a pure analog one. While the circuit in [5] (designed in a $1\mu\text{m}$ technology) comprises 16rules with 470ns delay, 8.6mW power consumption and 1.6mm^2 area, the presented controller implements 64rules with almost the same delay, 16mW power and only 2.6mm^2 area.

III. SOFT-START AND SPEED CONTROL OF DC MOTORS.

DC motors are widely used in modern industry because their torque-speed characteristics can be varied over a wide range while retaining high efficiency, and they provide a high starting torque. As said above, their control is a good field of application for chips like that presented in this paper. In order to illustrate it, a strategy to get a safe and fast start is proposed in this section.

Fig.7 shows a simple electrical model of a DC motor, enclosed in a dashed line [13]. In the steady state, the DC motor variables fulfil

$$V_m = I_m \cdot R_m + E_m \quad (1)$$

where I_m is the motor current, R_m is the motor winding resis-

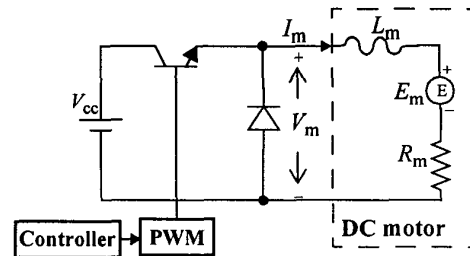


Fig. 7. Speed control of a DC motor by varying the applied voltage using a PWM controlled chopper.

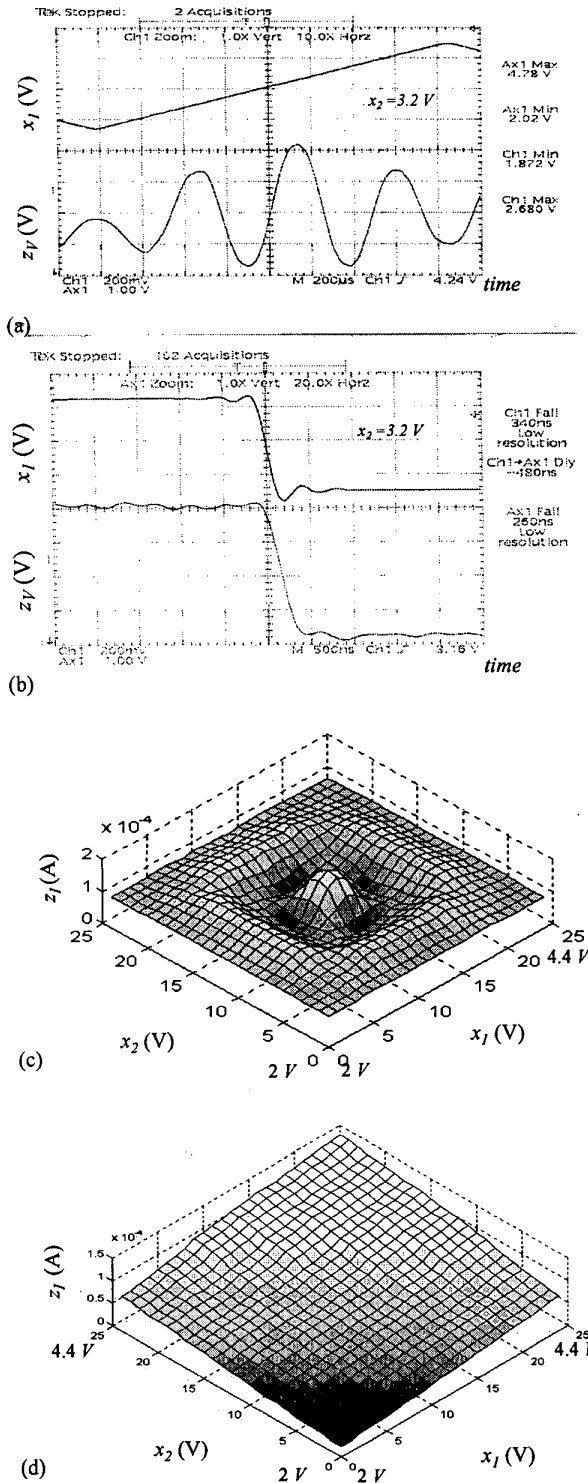


Fig. 6. Measured results: (a) DC nonlinear control surface section, (b) transient response, (c) nonlinear control surface and (d) linear control surface.

tance and E_m the motor electromotive force. The latter is also related to the motor speed ω as

$$E_m = K\phi \cdot \omega \quad (2)$$

where $K\phi$ is a constant associated to the magnetic flux. A DC motor can be started by a direct connection to the supply voltage,

$$V_m = \begin{cases} 0 & \text{for } t < 0 \\ V_{cc} & \text{for } t \geq 0 \end{cases} \quad (3)$$

This is the way to get the greatest starting torque and fastest start. However, at the instant $t = 0$, since the motor is not moving yet $\omega = 0$ in (2), thus $E_m = 0$. Consequently, at $t = 0$

$$I_m = \frac{V_{cc}}{R_m} \quad (4)$$

Thus, if R_m is small, a large current spike takes place at the beginning of the starting. Fig.10 shows this spike as measured in the laboratory when this start procedure is applied to the DC motor also used to illustrate the performance of the fuzzy controller. This current spike can damage the motor and cause a voltage drop in the supply network, thus soft-starting techniques are required. These are specially suitable in applications with frequent starts and stops. Two main strategies limit the current in the motor windings. First, it is possible to increase gradually the voltage applied at the motor terminals, which is equivalent to increase gradually the numerator in (4), hence I_m also grows gradually. Second, a programmable resistor could be connected in series with the motor winding. This is equivalent to increase the value of R_m in (4). Then the value of this resistor can be reduced gradually, hence I_m changes softly.

In the application example presented in this paper we follow the first approach. In order to control V_m we use the circuitry in Fig.7, where the motor is powered by a DC-DC switching converter (chopper). The voltage V_m between the motor terminals is determined by the duty-cycle D of the PWM signal that drives the switch of the converter as

$$V_m = D \cdot V_{cc} \quad (5)$$

Hence a gradual change of the duty cycle allows to increase the voltage V_m gradually.

The winding inductance L_m [13] causes only a current ripple whose amplitude depends on the L_m value and the switching frequency. This ripple can be minimized by using high switching frequencies. In the application example of this paper, a switching frequency as high as 100kHz is used to force a quick control response, thus to illustrate the controller behavior.

A. Control strategy

As said in the previous section, the control consists in providing a proper duty cycle to the switching converter. It is made by means of the closed control loop of Fig.8. If we consider the motor is a time invariant plant, we can write the following equations for Fig.8(a),

$$\begin{aligned} \dot{\omega}(t) &= \Phi_{\omega}(\omega(t), I_m(t), D) \\ I_m(t) &= \Phi_{I_m}(\omega(t), I_m(t), D) \\ D &= g(\omega(t), I_m(t)) \end{aligned} \quad (6)$$

where the first and second equations are related to the dynamic of the motor, and the third equation defines the control function. The latter should be chosen to force the plant state $(\omega(t), I_m(t))$ to follow a given target trajectory $(\omega_d(t), I_{md}(t))$.

In our case, the target trajectory is determined by two conditions: first, the starting must be safe; second, it should be as fast as possible. In order to get it, we propose to keep the motor current close to a maximum safe value,

$$I_{md}(t) = I_{mmax} \quad (7)$$

thus the motor is started with the maximum safe torque. Note that this is not equivalent to limit the current in the winding, where the torque falls at the same time the motor is speeded up. On the other hand, in order to get a soft start, the target speed trajectory is determined by the steady state equations (1),(2) and (5) as

$$\omega_d(t) = \frac{D(t) \cdot V_{cc}}{K\phi} - \frac{I_{mmax} \cdot R_m}{K\phi} \quad (8)$$

Hence the starting trajectory could be figured out as a sequence of steady states.

In order to reach this goal, the controller function $D = g(\omega(t), I_m(t))$ is built with the fuzzy approach as follows

$$\text{IF } I_m(t) \text{ is below } I_{mmax} \text{ THEN } D = D_T \quad (9)$$

$$\text{IF } I_m(t) \text{ is above } I_{mmax} \text{ THEN } D = D_{min} \quad (10)$$

where D_T and D_{min} are respectively the target and the minimum duty cycle. The previous rules force the current $I_m(t)$ to be close to I_{mmax} . Simultaneously, the target speed trajectory is forced by providing the duty cycle that fulfils (8),

$$\text{IF } I_m(t) \text{ is } I_{mmax} \text{ THEN } D = \frac{K\phi}{V_{cc}}\omega + \frac{R_m}{V_{cc}}I_{mmax} \quad (11)$$

Note that, although expressed in the IF-THEN way, the previous sentences are not the final fuzzy rules, which are obtained by interpolating the ramp in (11) as well as forcing (9) and (10) for every value of ω along this ramp. The control finishes once the duty cycle reaches the value it has in the target steady state.

Fig.8(b) depicts the building blocks in the actual control loop, where the motor current and speed are sensed by means of a resistor and a tachogenerator respectively. In addition, some interface circuitry is needed at input and output of the controller to adapt the ranges as well as to translate the output into a PWM signal. Fig.9(a) shows the fuzzy rules for a specific example, expressed as singletons associated to input intervals in a lattice partition. Note that the input variables are not the motor speed and current, but linearly related voltages which are got after sensing and signal conditioning.

B. Results

The Fig.10 shows the starting curves for a direct connection of the motor to the switching converter with $D_T = 71\%$. The peak value corresponds to 1.57 Amp of current in the motor, 4.75 times larger than the nominal motor current after starting, whose value is 0.33 Amp. The measured rise time was 212ms. For the same DC motor and target duty cycle, the curves in Fig.11 are got with the control strategy described above and the monolithic controller presented in the paper. The control

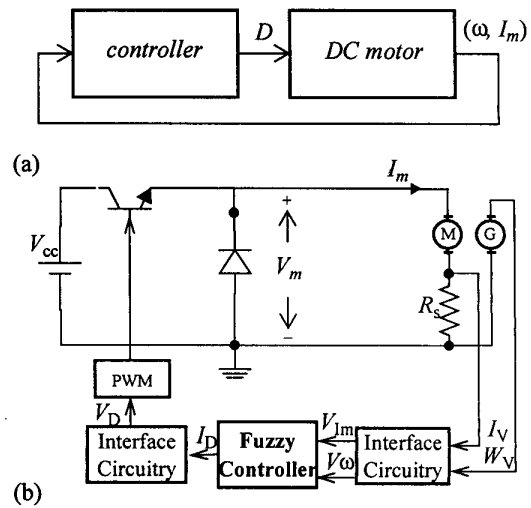


Fig. 8. Closed loop soft-start and speed control of a DC motor by a fuzzy controller: a) Concept; b) implementation.

surface implemented by the controller is shown in Fig.9(b). The maximum motor current is fixed to 0.8 Amp. The control action V_D , and the sensed input V_ω and V_{I_m} (see Fig.8(b)) are shown at the top, middle and bottom of Fig.11 respectively. Note that V_ω and V_{I_m} take their minimum value when the motor is not powered, while V_D takes initially its maximum value. Then, when the motor is powered V_{I_m} rises, although the control action (V_D) forces it to reach a value close to that associated to I_{mmax} . This control action is kept until the motor speed is close to its final value. The measured rise time was 253ms, a little bit larger than that obtained with the direct connection, but now the current peak value is 0.8 Amp, only 2.43 times larger than the final steady-state motor current.

IV. CONCLUSIONS

This paper has presented both a monolithic mixed-signal fuzzy controller as well as an application example. The former performs 500ns delay and 16mW power consumption for a technology of 0.7 μ m and a power voltage supply of 5V. This means a similar behavior that the reported pure analog monolithic controllers, while the complexity is much higher, up to 64 rules and 2 inputs. Many experimental results from the controller are shown. On the other hand, an application example is proposed to illustrate how the controller works. It has been chosen to force high speed in the control action. Specifically, a DC motor is powered with a switching converter which is driven by a PWM signal of 100kHz.

The controller is able to change the duty cycle of this signal every cycle, thus the control is very efficient. Such ability is exploited to carry out a fast and safe starting of the motor. A strategy for this start is proposed and experimental results show its success.

ACKNOWLEDGMENT

This work was supported in part by the Spanish C.I.C.Y.T. under Contract TIC99-0826.

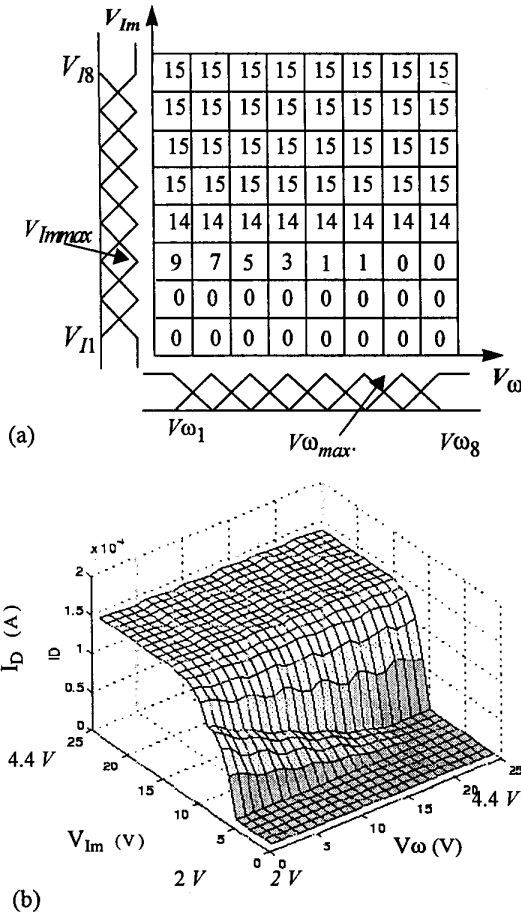


Fig. 9. Fuzzy rule set (a); and Control surface (b) for the application example

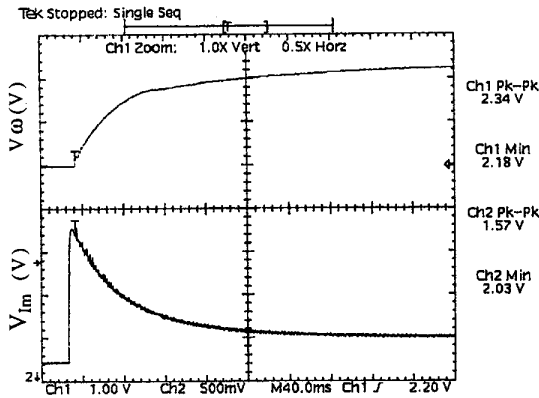


Fig. 10. Direct connection starting of a DC motor: Speed curve (top); Armature Current curve (bottom)

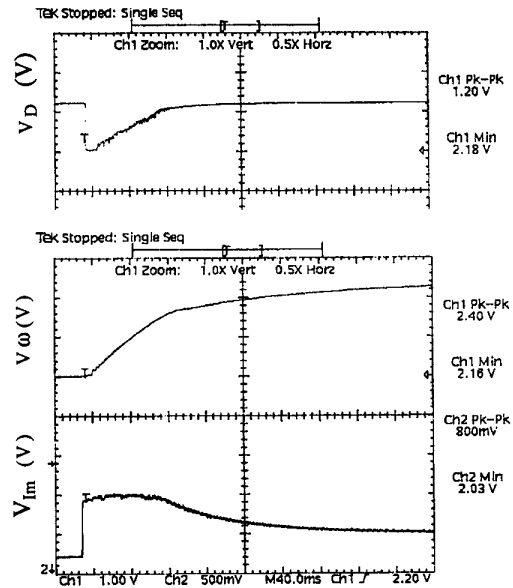


Fig. 11. Results from the soft-start procedure: Control action V_D (top); Speed V_{ω} (middle); Current V_{Im} (bottom)

REFERENCES

- [1] F.Vidal-Verdú, R. Navas-González and A. Rodríguez-Vázquez, "A Multiplexed Mixed-Signal Fuzzy Architecture". *Proceedings of the FUZZ-IEEE '98*, Anchorage, Alaska, May 1998.
- [3] N.Manaresi, R. Rovatti, E. Franchi, R. Guerrieri, and G. Baccarani, "A Silicon Compiler of Analog Fuzzy Controllers: From Behavioral Specifications to Layout". *IEEE Trans. on Fuzzy Systems*, Vol. 4, pp. 418-428, November 1996.
- [4] S. Guo, L. Peters, and H. Surmann, "Design and Application of an Analog Fuzzy Logic Controller". *IEEE Trans. on Fuzzy Systems*, Vol. 4, pp. 429-438, November 1996.
- [5] A. Rodríguez-Vázquez, R. Navas, M. Delgado-Restituto and F. Vidal-Verdú, "A Modular Programmable CMOS Analog Fuzzy Controller Chip", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal processing*, Vol. 46, No.3 pp. 251-265, March 1999.
- [6] K.A. Nishimura, "Optimum Partitioning of Analog Mixed-Signal Circuits for Signal Processing". Memorandum No. ECB/ERL M93/67, University of California at Berkeley. 1993
- [7] J.S.R. Jang and C.T. Sun, "Neuro-Fuzzy Modeling and Control". *Proceedings of the IEEE*, Vol. 83, pp. 378-406, March 1995.
- [8] F.Vidal-Verdú, R. Navas and A. Rodríguez-Vázquez, "Circuits for On-Chip Learning in Neuro-Fuzzy Controllers". *Proceedings of the Micro-Neuro '99*, Granada, Spain, April 1999
- [9] P.P. Bonissone et al., "Industrial Applications of Fuzzy Logic at General Electric", *Proceedings of the IEEE*, Vol. 83, pp. 450-465, March 1995
- [10] J.L. Guardado, K.J. Cornick, V. Venegas, J.L. Naredo, E. Melgoza. "A Three-Phase Model for Surge Distribution Studies in Electrical Machines". *IEEE Trans. on Energy Conversion*, Vol 12, Iss 1, pp. 24-31, 1997
- [11] B. K. Bose, "Expert Systems, Fuzzy Logic, and Neural Network Applications in Power Electronic and Motion Control" *Proceedings of the IEEE*, Vol. 82, pp. 1303-1323, August 1994
- [12] 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.
- [13] M.Rashid, *Power Electronics Circuits, Devices, and Applications*, Prentice Hall International 1993.