Fuzzy Logic Techniques Applied to the Control of a Three-Phase Induction Motor

João L. Afonso Jaime Fonseca Júlio S. Martins Carlos A. Couto

Department of Industrial Electronics
University of Minho
4800 Guimarães, Portugal

Abstract – **Until not long ago, most fuzzy logic based control applications were limited to the management of user interfaces, sensors and actuators, corresponding to a slow software operation. These techniques are specially appropriated whenever the system model is non-linear and/or difficult to obtain. Is it viable to use fuzzy logic in faster "real time" applications? Is it interesting to use these techniques even when the system model is known (including linear cases)? We are trying to answer some of these questions applying fuzzy logic to control electrical machines. In this paper we use a slip control scheme for a three-phase induction motor fed by a voltage-source PWM inverter as an example. The evaluation of the fuzzy logic controller behaviour is made through the comparison with a traditional technique (PI controller with antiwindup mechanism), using computer simulations (performed with Matlab/Simulink) and experimental results. Implementation was made using the *fuzzy*TECH software tool to design the fuzzy logic controller and to produce C-code for an Intel 80C196KC microcontroller. An 1 kW three-phase induction motor fed by a voltage source IGBT power module was used in the experiments.**

I. INTRODUCTION

Fuzzy logic has emerged as a profitable tool for the controlling of complex industrial processes, as well as for household and entertainment electronics, diagnosis systems and other expert systems. It is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth (truth values between "completely true" and "completely false"). Fuzzy logic was introduced by Dr. Lofti Zadeh of UC/Berkeley in 1960's as a mean to model the uncertainty of natural language, but only recently its use has spread over a large range of engineering applications. Fuzzy logic techniques attempt to simulate human thought processes, even in technical environments. In doing so, the fuzzy logic approach allows the designer to handle efficiently very complex closed-loop control problems, reducing engineering time and costs [1, 2].

Fuzzy logic has been mainly used in industrial automation for relatively slow processes. Fuzzy control supports non-linear design techniques that are now being exploited in motor control applications [3-7]. An example includes the ability to distribute gain over a range of inputs in order to not saturate the control capability.

In this paper we try to show, through simulation and experimental results, that the use of fuzzy logic techniques can be advantageous, even in cases where classical control is used and perform well. As an example, fuzzy logic is compared with a PI controller with antiwindup mechanism, in a slip control scheme for an induction motor fed by a voltage-source inverter.

II. SIMULATION AND DESIGN TOOLS

Matlab/Simulink was employed as a simulation tool. Matlab is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language. It is a natural environment for analysis, algorithm prototyping, and application development. Simulink is built on top of Matlab, and is an interactive environment for modelling, analysing and simulating a wide variety of dynamic systems. Together with a graphic interface, this tool provides an extensive block library, several integration algorithms, and allows the user to easily select the simulation parameters.

The Matlab Nonlinear Control Design Blockset was used to optimize the PI controller parameters. This tool provides a time-domain-based optimization approach to control system design. It is designed for use with Simulink block diagrams, and automatically tunes system parameters based on user-defined time-domain performance constraints.

The *fuzzy*TECH MCU-96 Edition was used to design the fuzzy logic controller. It is a full graphical tool that supports all design steps for fuzzy system engineering: structure design, linguistic variables and rules definition, and interactive debugging. Moreover, this tool generates C-code with optimized assembly functions to the Intel MCS-96 microcontrollers family. It also produces M-code, which is usually used for system representation in simulation and mathematical software packages. An M-code file, representing the fuzzy logic controller developed with *fuzzy*TECH, was imported by the Matlab/Simulink to perform the fuzzy logic simulations.

III. SLIP CONTROL

A conventional slip control scheme for an inverter fed induction motor which is used for low performance variable speed drives is presented in Fig. 1 [8, 9]. Traditionally, the speed error (ω_e) is input to a PI or PID controller (*block A*) that sets the motor slip frequency (ω_r). Stator frequency (ω_s) is obtained by adding slip frequency to rotor speed (ω), and stator voltage (U_s) is set accordingly to a pre-defined $U_s/\omega_s \approx$ *constant* law (*block B*), so that motor flux is kept at its nominal value. Voltage and frequency values are then input to the voltage source inverter (*block C*).

Assuming that fast response is not required, a linear approximation of the induction motor steady state model can be used. Slip frequency is almost proportional to torque and must be limited, setting (indirectly) a limit to both peak torque and stator current.

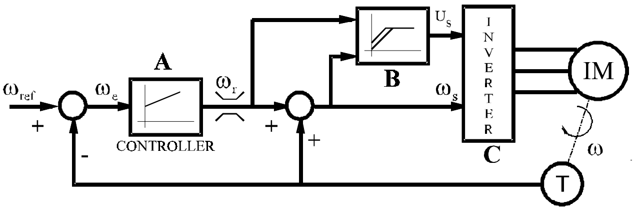


Fig. 1. Slip control for an induction motor voltage-source inverter drive.

IV. CONTROLLERS DESIGN

In order to evaluate the merits of the fuzzy logic techniques compared to a classical approach the induction motor slip controller was implemented, first using PI control, and then fuzzy logic.

A. PI Controller

Note that because of slip limitation, which introduces a non-linearity at the controller output, a PI with antiwindup mechanism must be used [10].

The PI controller was first designed through a classical control approach (root-locus). Then the Nonlinear Control Design Blockset (NCD) was used to optimize its response to a speed reference step and to minimize the speed variation when a torque disturbance is applied.

Fig. 2 presents the system block diagram used to perform the PI controller simulations in the Simulink environment. The *Zero-Order Hold block* is used to set the simulation sampling time equal to 5.1 ms, which is the value used in the practical implementation.

The *PI Controller block*, showed in Fig. 2, is expanded in Fig. 3.

The use of a PID controller was considered. However, simulation results showed that, due to the relatively slow sampling rate used, the improvement achieved was minimal and would not compensate the increase in computational efforts.

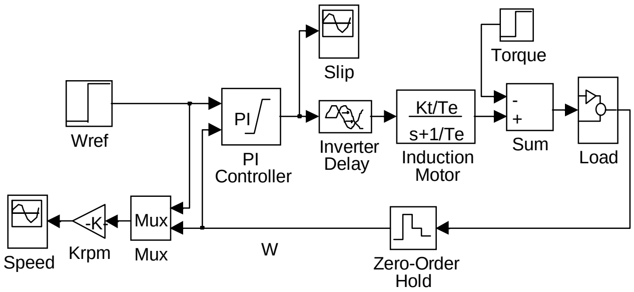


Fig. 2. Slip control system block diagram in the Simulink environment.

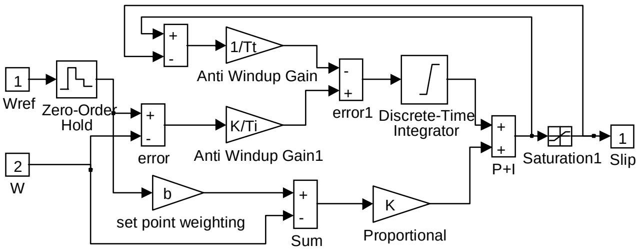


Fig. 3. PI Controller block diagram expanded

B. *Fuzzy Logic Controller*

To simulate the fuzzy logic control the *PI Controller block* in Fig. 2 is replaced by *the Fuzzy Controller block*, which is expanded in Fig. 4. The *M-File block*, seen in this figure, was produced by the *fuzzy*TECH software tool, and represents the same fuzzy logic controller implemented in the real-time environment.

The structure of the fuzzy logic controller is presented in Fig. 5. It has two inputs: the speed error (*SpeedError*) and the speed error variation (*ErrorVar*); and one output: the slip increment (*Slip_Inc*).

Triangular Membership Functions (M.F.) were employed for the inputs, as this is the only function type allowed by *fuzzy*Tech. *SpeedError* uses 3 M.F.: Negative (N), Zero (ZE) and Positive (P). *ErrorVar* is described with 5 M.F.: Negative Large (NL), Negative Small (NS), ZE (Zero), Positive Small (PS) and Positive Large (PL).

The defuzzification method applied was the CoM (Center of Maximum). In control applications, CoM is most commonly used because the output value represents the best compromise of all inferred results, with high computational efficiency. Once CoM considers just the maximum output values, the output was represented by singleton Membership Functions (which can be considered as a special case of triangular M. F.). The output *Slip_Inc* uses 7 M.F.: Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM) and Positive Large (PL).

The Membership Functions for the inputs and the output are showed in Fig. 6. The inputs and the output are related through 11 rules (Table I). Each rule output was determined by "MIN-MAX" inference.

Fig. 5 shows that the inputs *SpeedError* and *ErrorVar* are obtained from ω_{ref} and ω, after saturation and normalization respectively through *block p1* and *block p3*. The *Slip_Inc* value is normalized through *block p4* and then added to its previous value to give, after saturation, the *Slip* value, which is the motor slip control output.

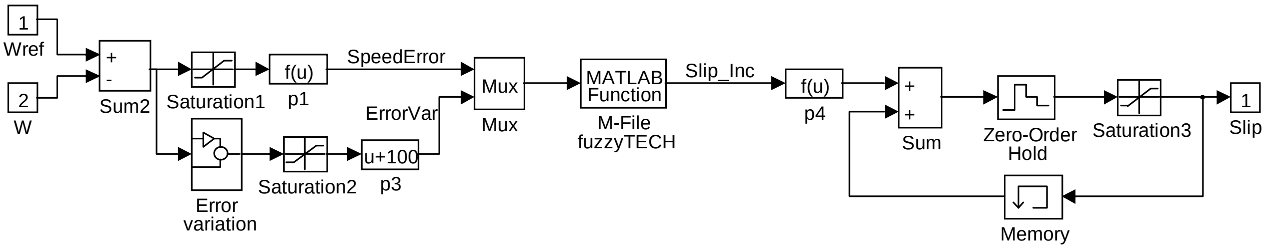


Fig. 4. Fuzzy Controller block diagram expanded.

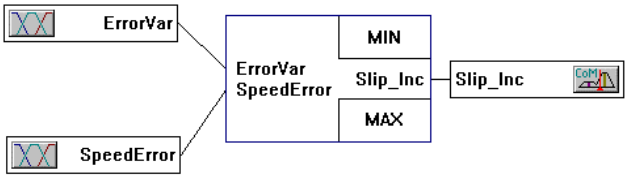


Fig. 5. Fuzzy logic controller structure.

TABLE I
FUZZY LOGIC SLIP CONTROLLER RULES

IF		**THEN**	
ErrorVar	**SpeedError**	**DoS**	**Slip_Inc**
	NL	1.0	NL
N	NS	1.0	NM
N	ZE	1.0	NS
N	PS	1.0	NM
ZE	NS	1.0	NS
ZE	ZE	1.0	ZE
ZE	PS	1.0	PS
P	NS	1.0	PM
P	ZE	1.0	PS
P	PS	1.0	PM
	PL	1.0	PL

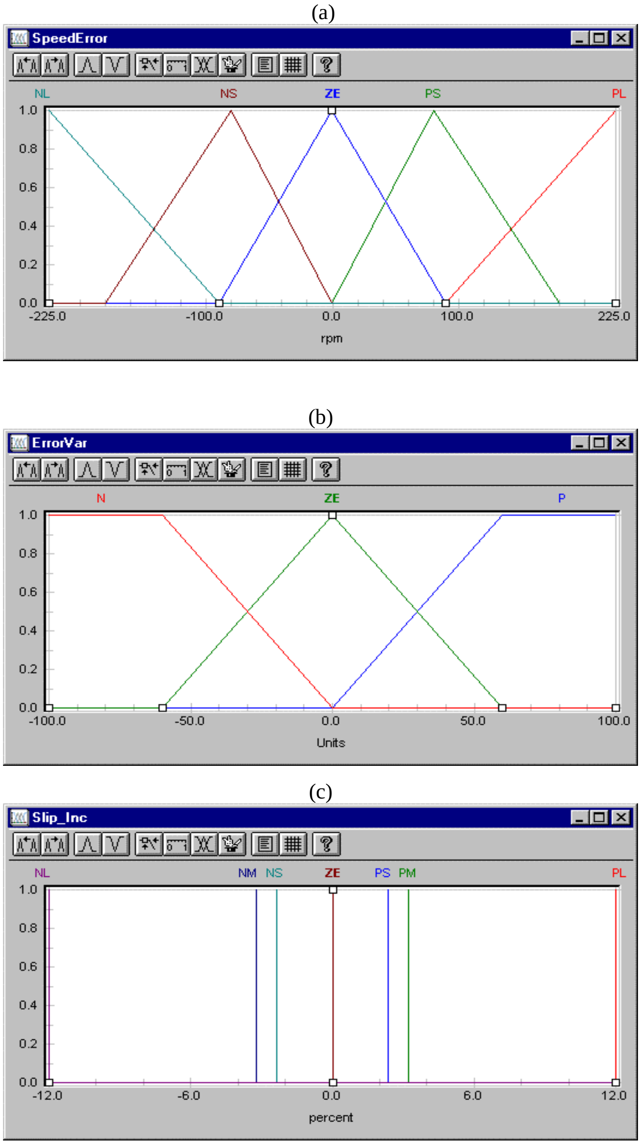


Fig. 6. Fuzzy logic controller Membership Functions: (a) SpeedError; (b) ErrorVar; (c) Slip_Inc.

V. SIMULATION RESULTS

The computational simulations compare the behaviour of PI and fuzzy logic controllers, showing their speed and slip values during motor start-up and then in response to a sudden load change from zero to nominal motor torque value.

First simulations were performed with the motor-load inertia value for which the PI controller parameters were optimized, which is the same value of the experimental implementation. Then the inertia value was modified in order to evaluate the controllers sensitivity to system parameters changes.

Fig. 7 presents the simulations performed for the first inertia value ($J = 2.10^{-2}$ kg.m^2). The response for both controllers is almost identical.

Fig. 8 shows simulations results with $J = 10.10^{-2}$ kg.m^2. In this case the fuzzy logic response is better: the overshoot values are smaller and its response is faster.

The controllers behaviour can be better compared using standard performance indexes. Table II shows the values of IAE (Integral of Absolute Error) and ITSE (Integral of Time Squared Error) for the PI and fuzzy controllers, during start-up and load application conditions. These indexes show that the fuzzy logic controller performs better than the PI controller when the motor-load inertia is changed to $J = 10.10^{-2}$ kg.m^2.

TABLE II
SIMULATION SPEED RESPONSE PERFORMANCE - $J = 10.10^{-2}$ kg.m^2

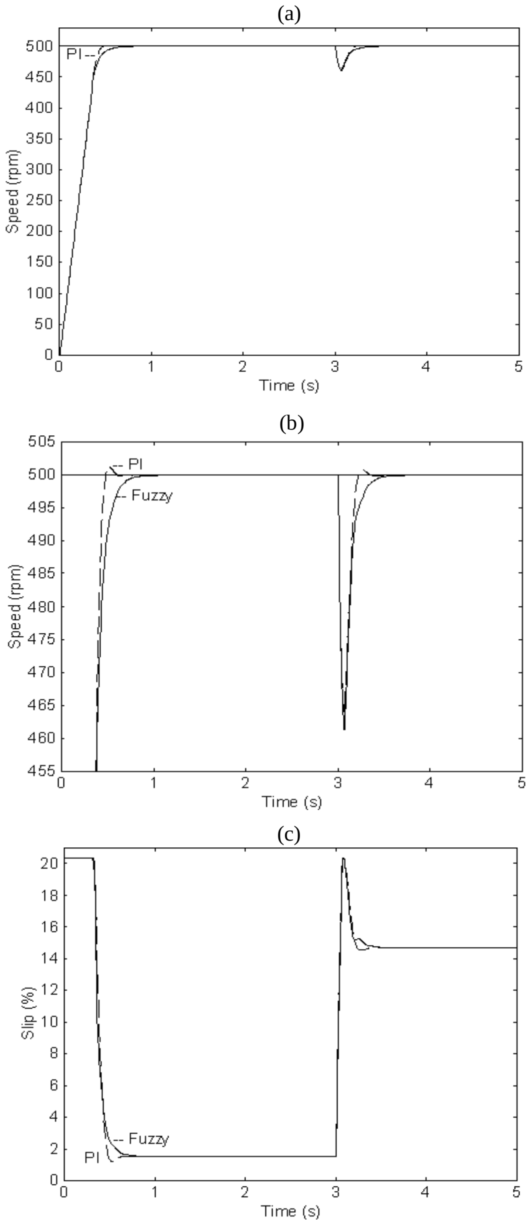
	Start-up		**Load Application**	
	IAE	**ITSE**	**IAE**	**ITSE**
PI	1.0240	2.2532	0.0180	0.0162
Fuzzy	1.0182	2.2565	0.0122	0.0100

VI. HARDWARE IMPLEMENTATION

An 1 kW three-phase induction motor fed by a voltage source IGBT power module was used in the experiments. No fuzzy processors or any other specific hardware was used to implement the fuzzy logic controller. A standard Intel 80C196KC microcontroller performs the control algorithm and generates the PWM waveforms for the IGBT motor drive inverter. The control algorithm runs within a 5.1 ms loop (sampling time). A digital implementation of sinusoidal pulse width modulation (SPWM) with harmonic addition in order to maximize the output voltage and minimize distortion is used [8, 11].

Fig. 9 shows that the additional hardware besides the microcontroller is minimal: just a few components to interface with the IGBT power inverter (*Inverter Interface*) and with the digital tachometer (*Encoder Interface*). A terminal can be connected to the system through the microcontroller serial port to set the speed reference and to monitor some of the control variables.

The *fuzzy*TECH MCU-96 Edition was used to produce the optimized C-code which implements the fuzzy logic controller. The generated code takes about 2.5 ms to run in the 80C196KC microcontroller with a 16 MHz clock.



(a)

(b)

(c)

Fig. 7. PI and fuzzy controllers simulation response - J = 2.10^{-2} kg.m^{2}:
(a) Speed; (b) Speed detail; (c) Slip.

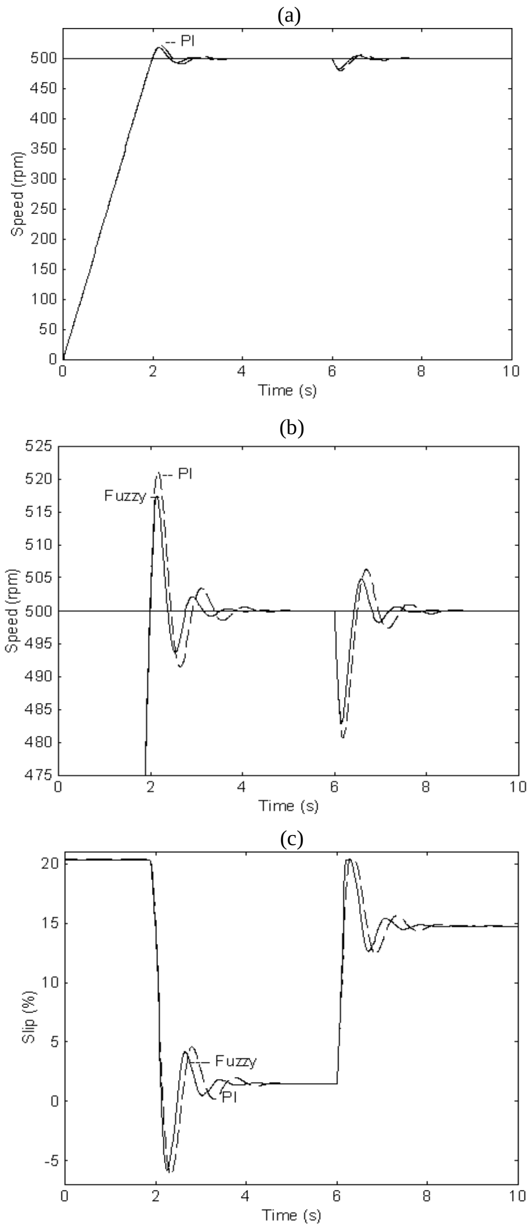


Fig. 8. PI and fuzzy controllers simulation response - J = 10.10^{-2} kg.m^2:
(a) Speed; (b) Speed detail; (c) Slip.

VII. EXPERIMENTAL RESULTS

Experimental results were achieved with the induction motor coupled to an eddy current dynamometer. It was measured a total inertia value J = 2.10^{-2} kg.m^2. In these experiments the motor is started-up unloaded, then when time is nearly 3.0 s a nominal load torque is applied through the dynamometer.

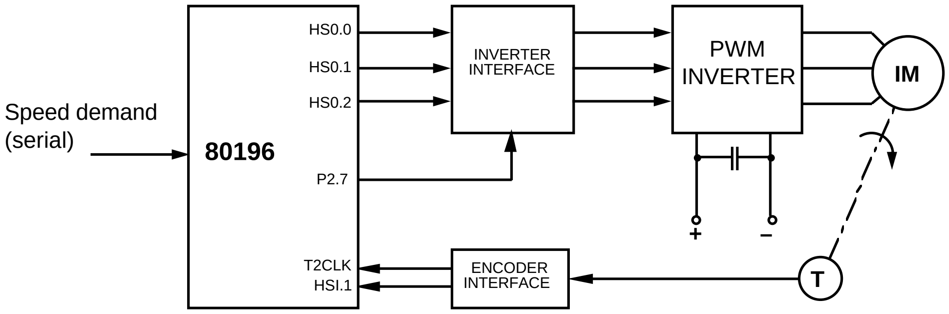


Fig. 9. Simplified diagram of the induction motor slip control scheme.

Fig. 10 and Fig. 11 show the speed and slip responses respectively for the PI controller and for the fuzzy logic controller. These figures confirm that, for this inertia value, the response is almost identical to both controllers. When nominal load is applied motor speed decreases about 4.5% before the controllers manage to re-establish the speed to the reference value, through the increase of the slip.

The main difference between the two responses has to do with the controllers sensitivity to speed noise (which was not considered in the simulations): the fuzzy logic controller behaviour is clearly better, presenting a speed ripple of about 2.7% unloaded and 2.0% loaded, against values of respectively 5.6% and 3.0% for the PI controller.

Table III shows the standard performance values for the measured speed response of the two controllers. The fuzzy logic controller results are better.

TABLE III
MEASURED SPEED RESPONSE PERFORMANCE - J = 2.10^{-2} kg.m^2

	Start-up		Load Application	
	IAE	**ITSE**	**IAE**	**ITSE**
PI	0.2506	0.0182	0.0155	0.0014
Fuzzy	0.2191	0.0143	0.0137	0.0012

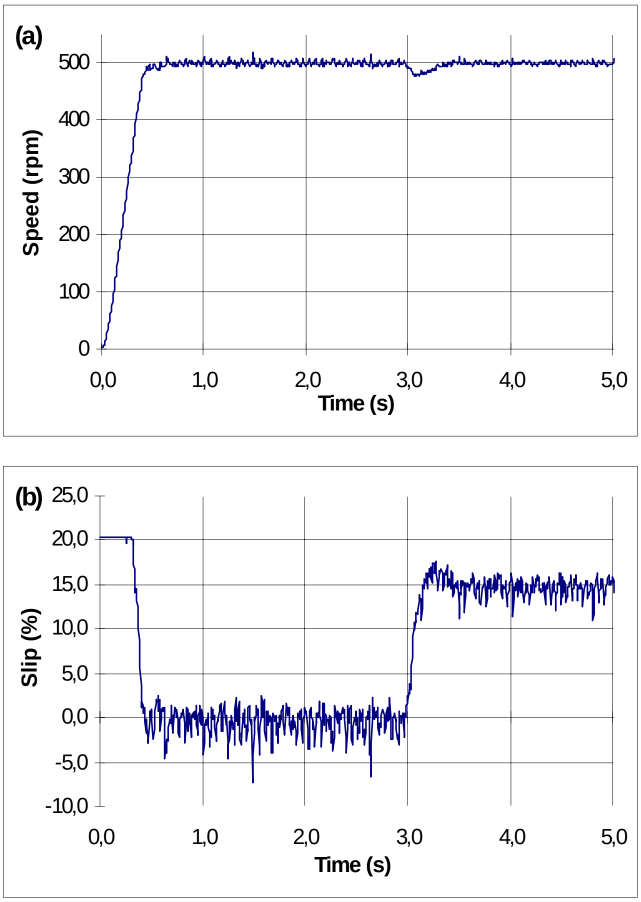


Fig. 10. PI controller experimental results: (a) Speed; (b) Slip.

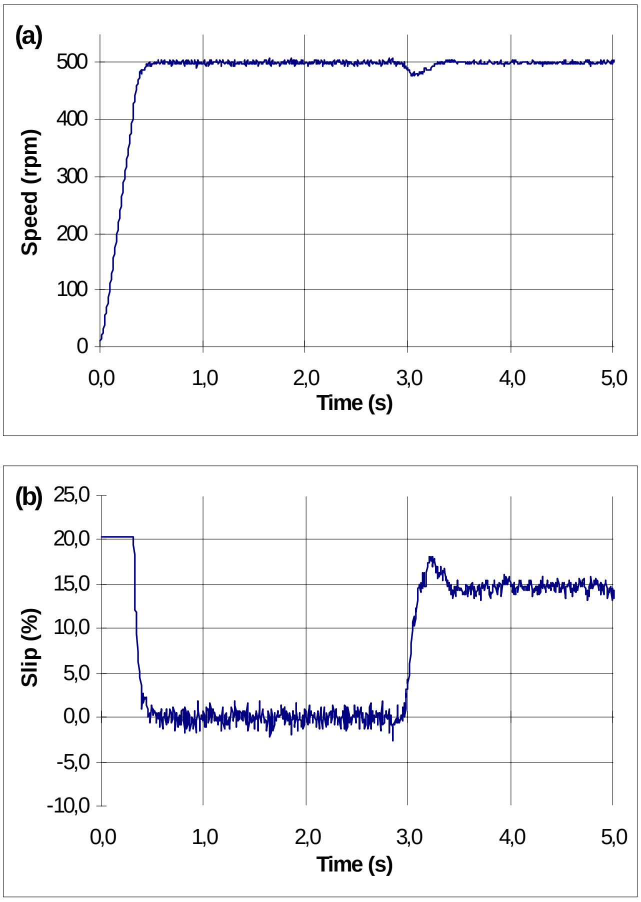


Fig. 11. Fuzzy logic controller experimental results: (a) Speed; (b) Slip.

VIII. CONCLUSIONS

An evaluation of fuzzy logic techniques applied to the control of electrical machines was presented. As an example, a slip control scheme for an induction motor fed by a voltage-source inverter was presented.

Both simulation and experimental results confirmed that the fuzzy logic approach is feasible and can be an interesting alternative to conventional control, even when the system model is known and linear.

The implemented fuzzy logic controller presented a slightly superior dynamic performance when compared with a more conventional scheme (PI controller with antiwindup mechanism), namely in terms of insensitivity to changes in model parameters and to speed noise, which can be an important requirement in speed/position control schemes using electrical machines, namely in robotics.

Some authors claim that fuzzy logic controllers are easier to tune than conventional ones, and therefore the development times are shortened. From our experience we cannot support this statement, at least for this type of application.

Matlab/Simulink and *fuzzy*TECH software tools were used for respectively simulation and controller design.

The hardware used to accomplish the system is minimum. No fuzzy processors or any other specific hardware was used. A standard Intel 80C196KC microcontroller performs the control algorithms and generates the PWM waveforms for the IGBT motor drive inverter.

IX. REFERENCES

[1] René Jager, “Fuzzy Logic in Control”, PhD Thesis, Delft University, Holland, 1995.

[2] Jerry M. Mendel, “Fuzzy Logic Systems for Engineering: A Tutorial”, Proceedings of the IEEE, Vol. 83, no. 3, 1995, pp. 345-377.

[3] G. C. Sousa, B. K. Bose and J. G. Cleland, “Fuzzy Logic Based On-Line Efficiency Optimization Control of an Indirect Vector Controlled Induction Motor Drive”, IEEE Trans. Ind. Electronics, vol. 42, no. 2, April 1995, pp. 192-198.

[4] G. C. Sousa and B. K. Bose, “A Fuzzy Set Theory Based Control of a Phase-Controlled Converter DC Machine”, IEEE Trans. Ind. Applicat., vol. 30, no. 1, Jan./Feb. 1994, pp. 34-44.

[5] S. A. Mir, D. S. Zinger and M. E. Elbuluk, “Fuzzy Controller for Inverter Fed Induction Machines”, IEEE Trans. Ind. Applicat., vol. 30, no. 1, Jan./Feb. 1994, pp. 78-84.

[6] P. Guillermin, “Fuzzy Logic Applied to Motor Control”, IEEE Trans. Ind. Applicat., vol. 32, no. 1, Jan./Feb. 1996, pp. 51-56.

[7] Jaime Fonseca, João L. Afonso, Júlio S. Martins, Carlos Couto, “Evaluation of Neural Networks and Fuzzy Logic Techniques Applied to the Control of Electrical Machines”, in Proceedings of the 5th UK Mechatronics Forum International Conference, Portugal, vol. 2, July 1996, pp. 15-20.

[8] J. S. Martins, “Controlo de Velocidade do Motor de Indução Trifásico”, PhD Thesis, University of Minho, Portugal, 1993.

[9] C. Couto and J. S. Martins, “Control of a Voltage Source Inverter Fed Induction Motor with On-Line Efficiency Optimization”, IEEE ICIT’94, Guangzhou, China, 1994.

[10] Karl J. Astrom, *Computer Controlled Systems*, Prentice-Hall, 1990.

[11] J. A. Houldsworth and D. A. Grant, “The Use of Harmonic Distortion to Increase the Output of a Three-Phase PWM Inverter”, IEEE Trans. Ind. Applicat., IA-20, Set./Out. 1984.