
MINIMAL FUZZY MICROCONTROLLER IMPLEMENTATION FOR DIDACTIC APPLICATIONS

F. Lara-Rojo¹, E.N. Sánchez² & D. Zaldívar-Navarro³

¹ ITESO University. A.P. 31-175. 45051 Guadalajara, Jal., México.

Tel. 52-3669.3517, Fax. 52-3669.3511, E-mail flara@iteso.mx

² CINVESTAV. Unidad Guadalajara. A. P. 31-438. 45091 Guadalajara, Jal., México.

Tel. 52-3684.1580, Fax. 52-3684.1708. E-mail sanchez@gdl.cinvestav.mx

³ Universidad de Guadalajara . División de Electrónica y Computación.

44860 Guadalajara, Jal. México. Tel. y Fax 52-3619 8471, E-mail zaldivar@cucs.udg.mx

Received: May 29th 2001 and accepted March 7th 2002

ABSTRACT

Fuzzy techniques have been successfully used in control in several fields, and engineers and researchers are today considering fuzzy logic algorithms in order to implement intelligent functions in embedded systems. We have started to develop a set of teaching tools to support our courses on intelligent control. Low cost implementations of didactic systems are particularly important in developing countries. In this paper we present the implementation of a minimal PD fuzzy four-rule algorithm in a low-cost 8-bit microcontroller, using a fuzzy logic software development system. On this ground we constructed a stand-alone fuzzy controller for a didactic liquid level system. We describe the methodology we followed, and present simulation and real time results of this controller.

RESUMEN

Las técnicas borrosas han sido utilizadas exitosamente en problemas de control en varios campos, y tanto ingenieros como investigadores están dirigiendo su atención a los algoritmos de lógica borrosa con miras a la implementación de funciones inteligentes en sistemas anidados. Nosotros hemos iniciado el desarrollo de un conjunto de herramientas didácticas para apoyar nuestros cursos en control inteligente. La implementación de sistemas didácticos de bajo costo es particularmente importante para los países en desarrollo. En este artículo presentamos la implementación de un algoritmo PD borroso de cuatro reglas mínimo en un microcontrolador de 8 bits de bajo costo, usando un sistema de desarrollo de software de lógica borrosa. En base a este desarrollo construimos un controlador borroso autónomo para un sistema de nivel de líquido didáctico. Describimos la metodología que empleamos, y presentamos los resultados de simulación y tiempo real para este controlador.

KEYWORDS: Fuzzy control, level control, didactic workbench, minimal rule number, real-time implementation.

1. INTRODUCTION

Fuzzy logic is a natural, continuous logic patterned after the approximate reasoning of human beings. It is a computational paradigm originally developed in the early 1960's [1]. Fuzzy logic allows partial truths and multivalued truths. It is therefore especially advantageous for problems that cannot be easily represented by mathematical modeling because data is unavailable, incomplete, or the process is too complex. The real-world language used in fuzzy control enables engineers to incorporate ambiguous, approximate

human logic into computers. Using linguistic modeling, as opposed to mathematical modeling, greatly simplifies system design and modification. It generally leads to quicker development cycles, easy programming, and fairly accurate control. However it is important to underline the fact that fuzzy logic solutions are usually not aimed at achieving the computational precision of traditional techniques, but aimed at finding acceptable solutions in shorter time.

Fuzzy logic enables users to implement intelligent functions in embedded systems. Typical fuzzy logic applications are being used to enhance existing products with intelligent functions that allows better adaptation for user requirements. For instance, we can find fuzzy control algorithms in autofocus, autoexposure and other control functions in image-processing equipment [2], as well as in washing machines and other consumer products [3]. A primary advantage of an embedded fuzzy logic control system is that even complicated functions can be implemented using the limited resources of a low-cost general use microcontroller, instead of an ASIC device or a dedicated fuzzy microcontroller.

Low cost implementations are particularly attractive in the case of didactic systems for universities of developing countries, where resources can be rather scarce. We are building up a set of teaching tools to be developed and employed in our courses on intelligent control in both our bachelors and masters' programs on Electronics Engineering . The fuzzy microcontroller reported in this paper is our first step in this direction.

With the purpose of implementing fuzzy logic control systems in low-cost microcontrollers, we consider the smallest possible fuzzy controller, consisting of two inputs, error and change of error ("rate" for short), one output, and four fuzzy control rules. The relation between PD controllers and this minimal four rules fuzzy controller has been established in [4]. The four rules fuzzy algorithm has been successfully used in real time applications. In [5], for instance, it was applied to swing-up an underactuated two-link mechanism.

There is a wide range of fuzzy logic implementation alternatives [6]. We decided to achieve a software implementation on a low-cost general use microcontroller. To this purpose we chose the PIC16C711 microcontroller [7] . In this endeavor we used the *fuzzyTECH-MP Explorer* [8], which is a fuzzy logic software development system for PIC16/17 microcontroller devices. *fuzzyTECH-MP Explorer* is a limited version of the *fuzzyTECH-MP Edition* that allows for the definition of fuzzy logic rules with a maximum of two input variables and one output variable. It is necessary to develop several assembly code routines to provide for system gains, analog to digital conversion, and some other functions required by the fuzzy control algorithm. We used the MPLAB [9], which is an integrated development environment for the Microchip PIC16/17 microcontrollers, to construct, integrate this code with the *fuzzyTECH* supplied kernel, and simulate it. MPLAB was also used to implant the code into the chip.

We applied the procedure described here (above or below) to develop a stand-alone fuzzy controller around the PIC16/17 microcontroller for an experimental liquid system. This is a relatively simple system which requires a two input - one input controller. The stand-alone fuzzy controller works with a minimal PD fuzzy four rules algorithm embedded in a PIC16C711 microcontroller. We present simulation as well as real time results of the system. The fuzzy microcontroller showed a satisfactory performance.

The paper is organized as follows : In Section 2 we briefly discuss the minimal fuzzy four rules controller. In Section 3 we describe the assembly code construction process for the PIC16C711 using *fuzzyTECH-MP Explorer* and MPLAB . The didactic liquid level system is described in Section 4. Section 5 describes the main points of the process for assembler code integration and construction of a stand-alone fuzzy controller for this plant. We show real-time results in Section 6. Conclusions and future work are presented in Section 7.

2. THE MINIMAL FOUR-RULE FUZZY ALGORITHM

For the sake of completeness, we first briefly review the basic structure and formulation of the MIMO fuzzy PD controller, originally presented in [10].

The structure of this MIMO fuzzy PD controller, which is a continuous-time controller, is based on the two inputs, one output controller shown in Figure 1, from which it can be seen that

$$u(t) = K_u u^*(t) \tag{2.1}$$

where K_u is a fuzzy control gain determined by the designer and $u^*(t)$ is the output of the fuzzy controller.

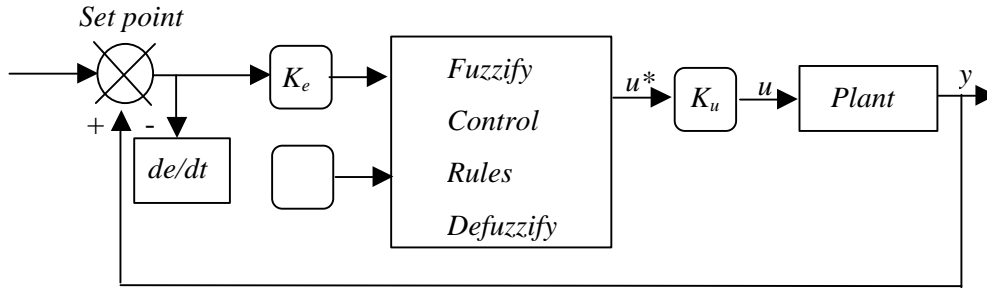


Figure 1. Fuzzy PD Controller.

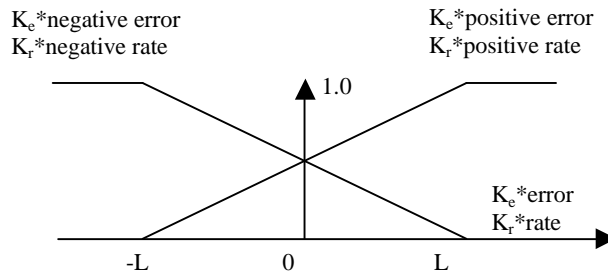


Figure 2. Input membership functions.

The two inputs are: the error signal $e(t)$ and its rate of change $\dot{e}(t)$; the only one output, $u(t)$, is fed to the controlled process. Membership functions are shown in Figures 2 and 3 respectively, where the same membership function is applied for both $e(t)$ and $\dot{e}(t)$ for simplicity. To employ the same membership function on two inputs, two scaling factors, k_p and k_d , are used to magnify $e(t)$ and $\dot{e}(t)$. L and H are two positive constants to be determined.

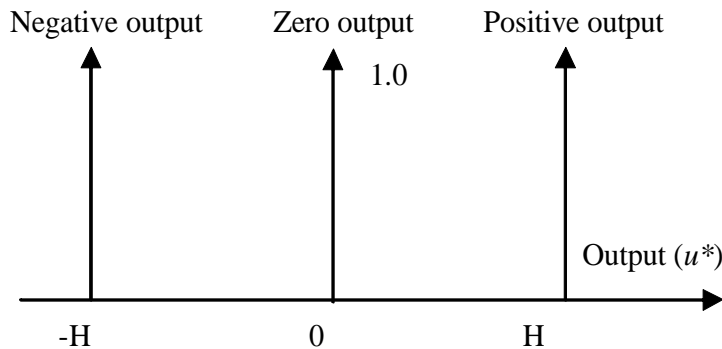


Figure 3. Output membership functions.

Based on the above membership functions, the fuzzy control rules are established as follows:

- (R1). IF e is ep AND e is rp THEN output is op
- (R2). IF e is ep AND e is rn THEN output is oz
- (R3). IF e is en AND e is rp THEN output is oz
- (R4). IF e is en AND e is rn THEN output is on

In these rules, e and e are defined as above, output is the output of the fuzzyfication mechanism, u^* , “ep” means “error positive” and “op” means “output positive,” etc. Moreover, “AND” is Zadeh’s logical “AND”. The commonly used “center of mass” formula is employed to defuzzify the output of the control action of the fuzzy PD controller.

3. IMPLEMENTATION OF THE MINIMAL FUZZY ALGORITHM UNDER FUZZYTECH-MP EDITION

As mentioned above, we chose the PIC16C711, which is a low-cost 8-bit microcontroller, to implement the fuzzy minimal fuzzy algorithm. The PIC16C711 microcontroller has an A/D converter, 1024 bytes ROM program memory, 68 bytes RAM memory and 20 MHz clock frequency.

Figure 4 shows the membership functions and fuzzy rules for the minimal fuzzy algorithm in the *fuzzyTECH* project editor window.

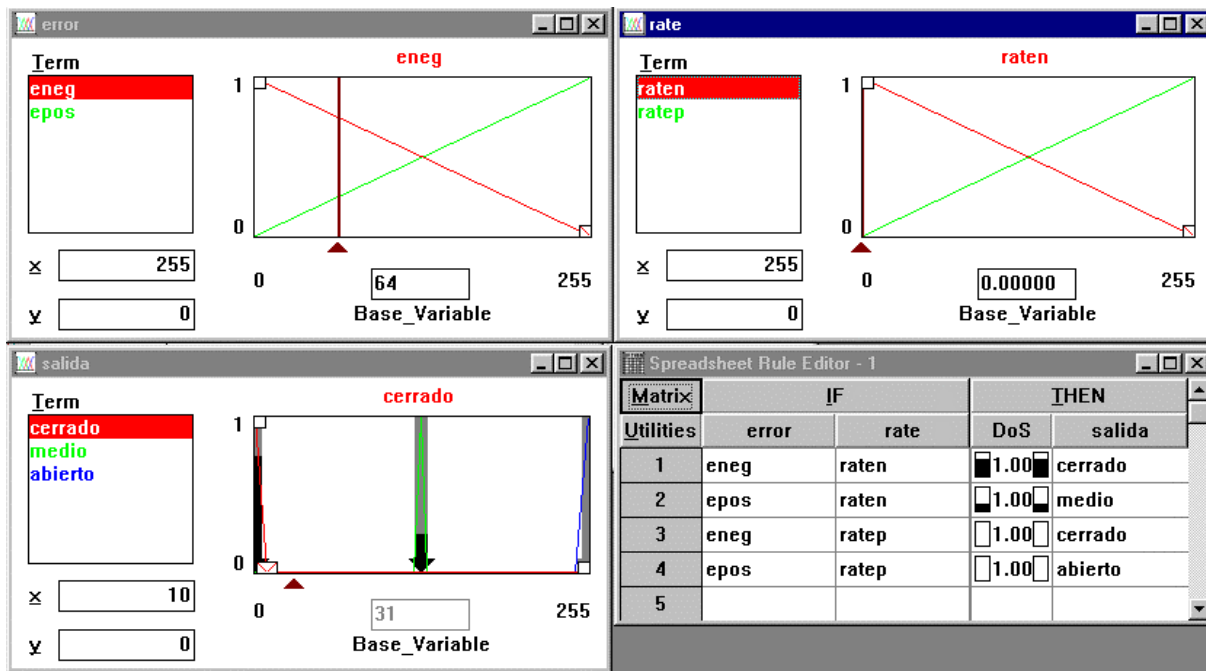


Figure 4. The minimal fuzzy algorithm in the *fuzzyTECH* project editor window.

FuzzyTECH generates a runtime code kernel for the actual fuzzy logic project. It consists of the routines required for fuzzification, inference, defuzzification, management of RAM memory, and some arithmetic functions.

A set of assembler code application routines had to be designed to perform the following functions: minimal fuzzy algorithm gains, error and error rate calculation, A/D conversion, digital filtering, PC set point reading and 128-bit format. The latter one is necessary to be able to operate with both negative and positive 128-bit numbers in the microcontroller.

This set of application routines was integrated with the fuzzyTECH generated code in a main program whose general structure is shown in Figure 5. A flow diagram of the main program is shown in Figure 6

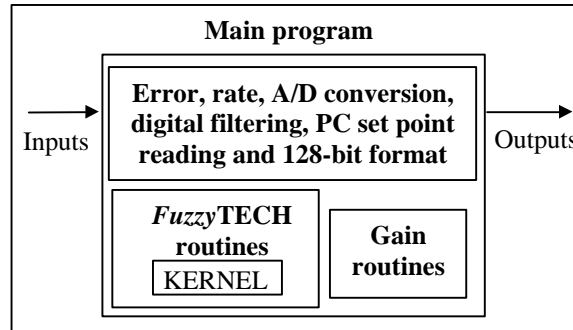


Figure 5. General structure of the main program.

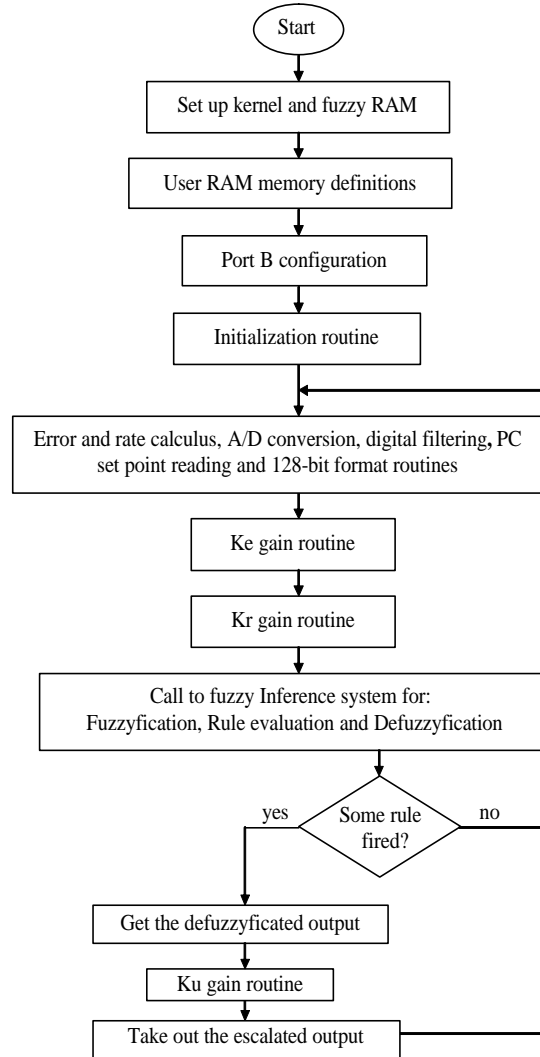


Figure 6. Flow diagram of the main program.

The method just described can be directly applied to develop fuzzy PIC16C711-based microcontrollers for other two-input one output systems. To extend it to systems with more inputs it would be necessary to use the full *fuzzyTECH-MP* Edition development environment.

4. THE LIQUID LEVEL SYSTEM

The liquid level system is shown in Figure 7. It consists of a single cylindrical tank with a fixed outlet at the bottom. Water is fed into the tank through an electric-pneumatic valve. A piezoelectric cell is attached to the bottom of the tank as a pressure sensor. Both the electric-pneumatic valve and the pressure sensor are connected through a signal conditioning interface, so that the input signal to the controller as well as the output control signal are in the 0-10 volts range.

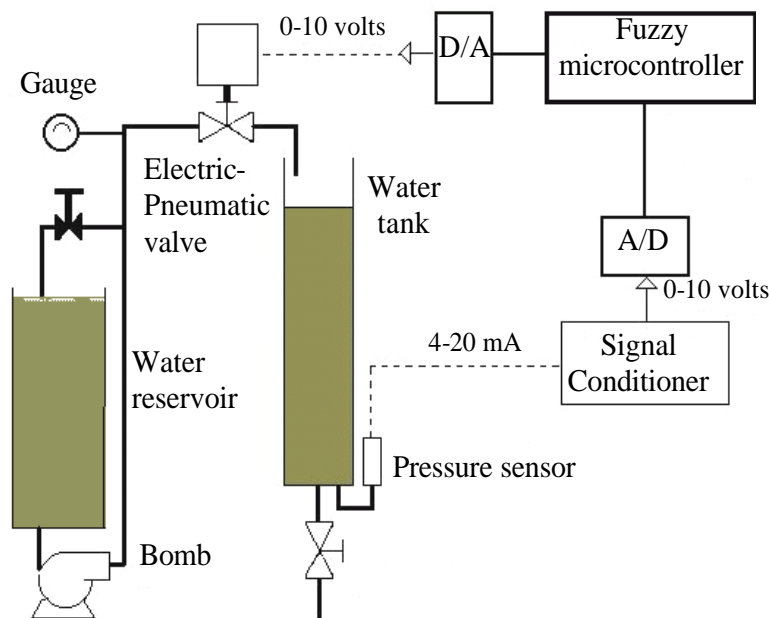


Figure 7. The liquid level system.

A simple model for the tank is given by

$$q_e(t) = A \frac{dh(t)}{dt} + A_o \sqrt{2gh(t)}$$

This model was simulated under MATLAB / SIMULINK in order to determine the input and output gains K_e , K_r , and K_u , as well as the constant L , which have to be integrated in the fuzzy algorithm for this system. We obtained rather good results with the following values for the constants: $K_e = 60$, $K_r = 1$, $K_u = 2$, $L = 127$.

Figure 8 shows the system behavior with a square reference signal that starts from zero (tank empty), and then fluctuates between 64.5 and 32.75 cm. These values were chosen in order to directly compare the results with corresponding experiments calculated on an 8-bit base, which is the word length of the PIC16C711 microcontroller. The same reference signal values were used later for the real time experiments.

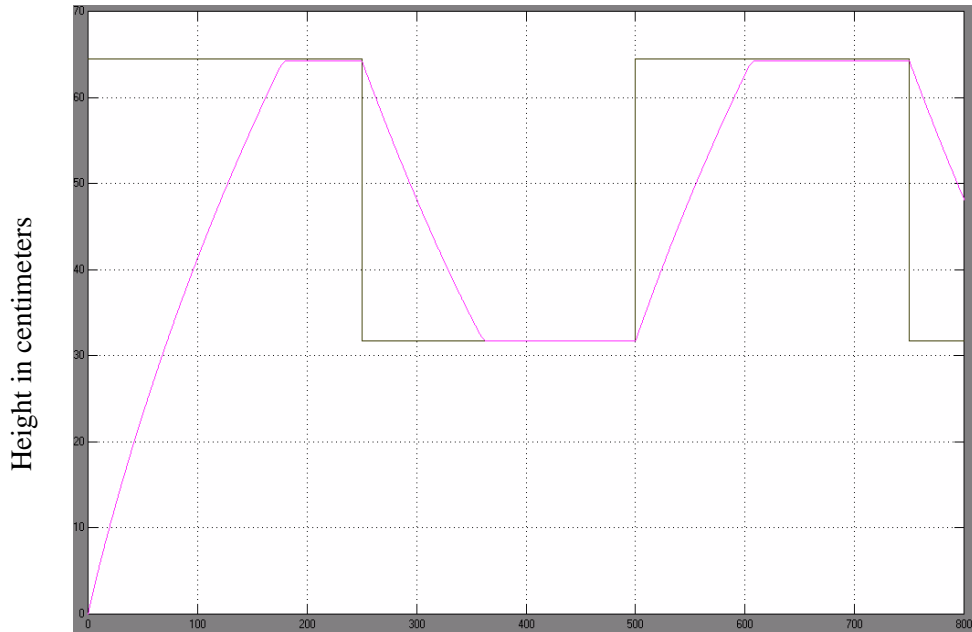


Figure 8. Simulation with a square reference signal of 64.5 and 32.75 cm peak values and 500 seconds period.

5. THE STAND-ALONE FUZZY MICROCONTROLLER

The input and output gains K_e , K_r , y K_u , previously obtained for the liquid level system were incorporated in the design of the fuzzy controller, and then in the assembler code generated, following the procedure described above.

In our implementation, the fuzzyTECH supplied code (basic fuzzy logic routines and kernel) requires 313 bytes. The code for the auxiliary routines is 224 bytes long. Hence, the total length of the fuzzy algorithm code is 537 bytes, which amounts to 53% of the PIC's program memory. There still remain 487 bytes of the PIC16C711 program memory, which are available for other purposes.

The present implementation was conceived for demonstration purposes. For this reason we arbitrarily restricted the possible values of the reference signal to the two values mentioned above (32.75 and 64.5 cm), which are to be commuted with a single bit signal. This restriction can easily be lifted with the help of a small extra routine to capture and use arbitrary reference signal values.

The stand-alone PIC16C711-based fuzzy microcontroller was constructed on a 6.5 cm x 6 cm board. The power supply was constructed on a separate module. The microcontroller board is shown in Figure 9. Both microcontroller board and power supply can be used in other experimental control applications in two input - one output systems. A simple PC-based monitoring interface was also developed. This monitoring interface was used to capture and display the real time results presented in Section 6.

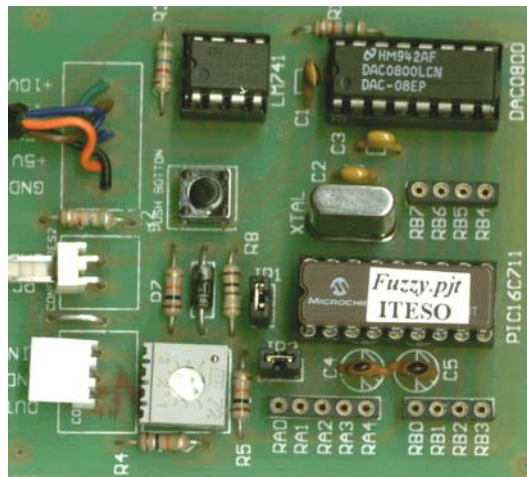


Figure 9. Fuzzy microcontroller board.

6. REAL TIME RESULTS

The experimental layout for the real time test of the stand alone fuzzy microcontroller is shown in Figure 10.

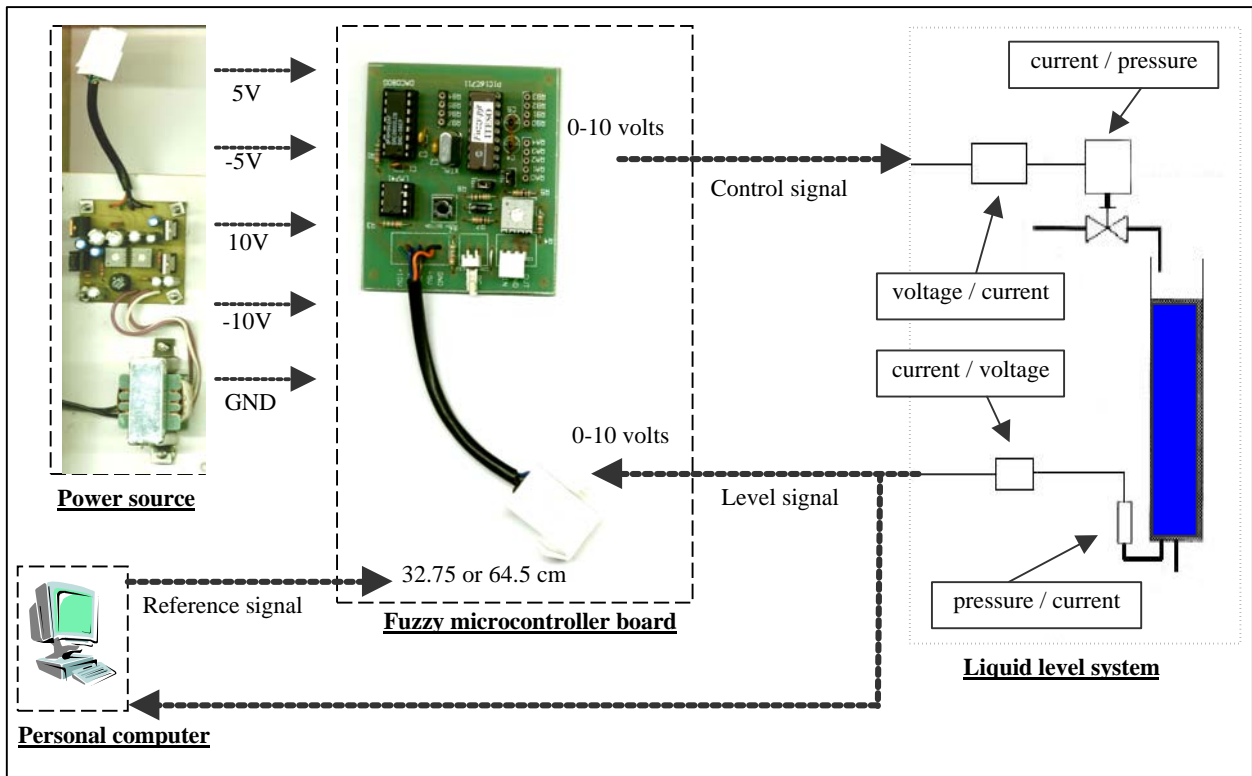


Figure 10. Experimental layout.

Figure 11 shows the real time results with a similar reference signal as the one used in the simulations.

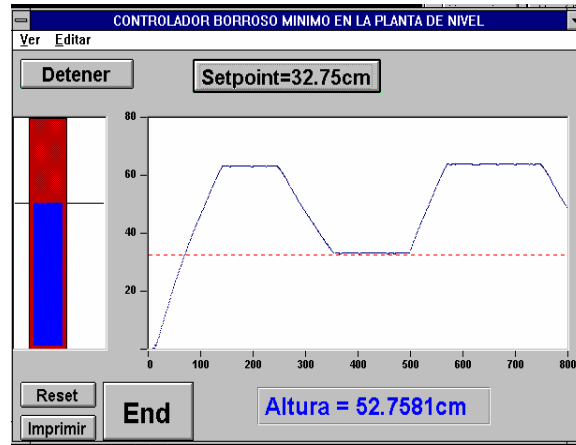


Figure 11. Real time results

One can observe a fairly good approximation to the top and bottom values of the reference signal. Numerical results are presented in the table of Table I. No further adjustment for the gain values was necessary. There are some discrepancies in the rise and fall times. This can be attributed to the roughness of the model used in the simulations, in particular to the fact that it does not take into account a more accurate model for the behavior of the electric-pneumatic valve. The results show nevertheless that the implementation of the fuzzy minimal algorithm in a low-cost microcontroller, can be successfully applied to control tasks in a similar range, which requires sample times not less than 10 microseconds.

Table I. Numeric real time results

Reference (set point)	Real time value
64.5 cm	64.2 cm
32.75 cm	33.0 cm

Based on this experience, we are able to easily reprogram the stand-alone fuzzy controller board produced in this project and are planning to apply this technique to other two input - one output control tasks in our laboratory.

7. CONCLUSIONS

We have started to build a set of didactic tools to support our courses on intelligent control. As a first step we have undertaken the construction of a fuzzy microcontroller. In this approach we have taken into consideration low cost implementations which are particularly attractive for educational purposes in developing countries. We implemented a minimal fuzzy four rules control algorithm in a low-cost microcontroller, and built a stand-alone fuzzy controller for a didactic liquid level system. We selected the PIC16C711 as the target microcontroller. We used the *fuzzyTECH-MP* Explorer software development system to implement the basic fuzzy control algorithm, and MPLAB IDE for the design of auxiliary assembly code routines and integration of the whole system. The code for the liquid level system controller required only 537 bytes of program memory. Sample time must not be less than 15 microseconds. The real time performance of the fuzzy autonomous controller is very satisfactory. Based on this experience, the controller can be easily reprogrammed and adapted to other control tasks. As future work we consider to apply the stand alone controller in other experiments. We also intend to apply a similar technique to plants with more than two inputs.

Remarks :

- FuzzyTECH and MPLAB are registered trademarks of Inform Software Corporation.
- PIC is a registered trademark of Microchip Technology Inc. in the U.S.A.
- MATLAB/SIMULINK is a simulation tool developed and distributed by The Mathworks Inc., 24 Prime Park Way, Natick, Mass. 01760; <http://www.mathworks.com>
- IBM and PC are registered trademarks of International Business Machines Corporation.

8. REFERENCES

- [1] Zadeh, L.A. "Fuzzy sets", Information and Control, vol. 8, 1965, pp. 338-353.
- [2] Takagi, H. "Survey of Fuzzy Logic Applications in Image-Processing Equipment", in Industrial Applications of Fuzzy Logic and Intelligent Systems, Yen J., Langari R., and Zadeh L.A., (ed) , IEEE press, USA,1995, pp. 69-92.
- [3] Takagi, H. "Applications of Neural Networks and Fuzzy Logic to Consumer Products ", in Industrial Applications of Fuzzy Logic and Intelligent Systems, Yen J., Langari R., and Zadeh L.A., (ed) , IEEE press, USA,1995, pp. 93-105.
- [4] Ying H., and Siler W., "Fuzzy Control Theory: A Nonlinear Case", Automatica, Vol.26, No.3, 1990, pp. 513-520.
- [5] Sánchez E., Nuño L.A., Hsu Y.-C., and Chen G., "Real Time Fuzzy Control for an Underactuated Robot", IV Joint Conference on Information Systems, Research Triangle Park, N.C., U.S.A., 1998.
- [6] Costa A., De Gloria A., Faraboschi P., Pagni A., Rizzotto G., "Hardware Solutions for Fuzzy Control", Proc. IEEE, Vol. 83, No. 3, 1995, pp. 422-434.
- [7] FuzzyTECH-MP, User's Guide, Microchip Technology Incorporated, USA, 1994.
- [8] PICmicro Mid-Range MCU Family Reference Manual , Microchip Technology Incorporated, USA, 1997.
- [9] "MPLAB IDE, Simulator, Editor. USER'S GUIDE", Microchip Technology Incorporated, USA,1996.
- [10] Hsu Y., Chen G., Sanchez E.N., "A fuzzy PD analysis for multi-link robot control: Stability analysis", Proc. 1997 IEEE Intl. Conference in Robotics and Automation, Albuquerque, New Mexico, USA, April 1997, pp 1412-1317.

Autor Biography



Fernando Lara-Rojo

Was born in Guadalajara, Mexico, in 1935. He studied Electronics Engineering at Instituto Politécnico Nacional (National Polytechnic Institute), Mexico City, in 1954-1958, and Physics in 1968-1973 at the University of Kiel, Germany, where he obtained a PhD working in Theoretical Solid State Physics.

In 1958-1968, he worked as an Engineer in telecommunication systems. In 1973-1975 he was professor at CINVESTAV, México, and from 1975 to 1980 at INAOE, Tonantzintla, México. Afterwards he has been working on teaching activities at ITESO University in Guadalajara, Mexico. He is mainly interested in Machine Intelligence and Automated Reasoning.

He is member of IEEE and AAAI. He has published 10 technical papers in international journal and conferences. In the last five years his work has focused on applications of Fuzzy Logic.



Edgar N. Sanchez

Was born in 1949, in Sardinata, Colombia. He obtained the BSEE from UI Santander, Colombia in 1971, the MSEE from CINVESTAV-IPN, Mexico, in 1974, and a PhD from Institut Nationale Polytechnique de Grenoble, France in 1980.

In 1971-1976 he worked as an Electrical Engineer in Bogotá, Colombia. From 1981 to 1990 he worked at the Electrical Research Institute, Cuernavaca, Mexico. In 1990-1996 he was a professor at Universidad Autónoma de Nuevo León (Autonomous University Nuevo Leon), Monterrey, Mexico. Since 1997, he has been with CINVESTAV-IPN, Guadalajara, Mexico. His research interest centers in Neural Networks and Fuzzy Logic applied to Automatic Control systems.

In 1985-1987 he was granted an USA NRC Award at NASA Langley Research Center, Hampton, Virginia, USA. He has published more than 50 technical papers in international journal and conferences, and has served as reviewer for different international journals and conferences. He has also been member of international conferences IPCs.



Daniel Zaldivar Navarro

Was born in Chihuahua, Chihuahua, Mexico in 1972. He studied Electronics Engineering at Universidad de Guadalajara in 1990-1995, and pursued postgraduate studies at ITESO Guadalajara, México, in 1995-1997, where he got an MC degree in Industrial Electronics.

He has been teaching at Universidad de Guadalajara in the Department of Electronics Engineering, since 1996 . He has also worked as consultant in Electronics Engineering for local industries. His professional interest centers in electronic design and applications of machine intelligence in control.

He was awarded by the German Service for Academic Interchange (DAAD) to pursue graduate studies in Germany. He began in the Spring 2002 doctoral studies in the field of robotics at the Institute of Informatics of the Free University of Berlin.