Original Paper

Soft Computing 8 (2004) 546–555 © Springer-Verlag 2003
DOI 10.1007/s00500-003-0313-z

Intelligent control of a stepping motor drive using a hybrid neuro-fuzzy approach

P. Melin, O. Castillo

Abstract Stepping motors are widely used in robotics and in the numerical control of machine tools where they have to perform high-precision positioning operations. However, the variations of the mechanical configuration of the drive, which are common to these two applications, can lead to a loss of synchronism for high stepping rates. Moreover, the classical open-loop speed control is weak and a closed-loop control becomes necessary. In this paper, fuzzy logic is applied to control the speed of a stepping motor drive with feedback. A neuro-fuzzy hybrid approach is used to design the fuzzy rule base of the intelligent system for control. In particular, we used the ANFIS methodology to build a Sugeno fuzzy model for controlling the stepping motor drive. An advanced test bed is used in order to evaluate the tracking properties and the robustness capacities of the fuzzy logic controller.

Keywords Intelligent control, Neuro-fuzzy approach, ANFIS method, Stepping motor

1 Introduction

Stepping motors can be viewed as electric motors without commutators [9]. Typically, all windings in the motor are part of the stator, and the rotor is either a permanent magnet or, in the case of variable reluctance motors, a toothed block of some magnetically soft material. All of the commutation must be handled externally by the motor controller, and typically, the motors and controllers are designed so that the motor may be held in any fixed position as well as being rotated one way or the other. Most stepping motors can be stepped at audio frequencies, allowing them to spin quite quickly, and with an appropriate controller, they may be started and stopped "on a dime" at controlled orientations.

For some applications, there is a choice between using servomotors and stepping motors. Both types of motors offer similar opportunities for precise positioning, but they differ in a number of ways. Servomotors require analog feedback control systems of some type. Typically, this involves a potentiometer to provide feedback about the rotor position, and some mix of circuitry to drive a current through the motor inversely proportional to the difference between the desired position and the current position. In making a choice between stepping motors and servomotors, a number of issues must be considered; which of these will matter depends on the application. For example, the repeatability of positioning done with a stepping motor depends on the geometry of the motor rotor, while the repeatability of positioning done with a servomotor generally depends on the stability of the potentiometer and other analog components in the feedback circuit. Stepping motors can be used in simple open-loop control systems; these are generally adequate for systems that operate at low accelerations with static loads, but closed loop control may be essential for high accelerations, particularly if they involve variable loads [1]. If a stepping motor in an open-loop control system is overtorqued, all knowledge of rotor position is lost and the system must be reinitialized; servomotors are not subject to this problem.

In this paper, the application of fuzzy logic is proposed to control the speed of a stepping motor drive. The closed-loop control scheme entails in incorporating engineering knowledge into the automatic control system by using the intuition and experience of the designer. This strategy was proposed by Zadeh [22–25], to describe complicated systems, which are hard to analyze using traditional mathematics. Indeed, Mamdani [11] was the first to report on the application of fuzzy logic to control a small laboratory steam engine. The success of this study led many scientists to attempt to control industrial processes such as chemical reactors, automatic trains, or nuclear reactors using fuzzy algorithms. The results of these experiments, showed that, fuzzy controllers perform better, or at least as well as, classical controllers. Moreover, this technique offers the advantage of requiring only a simple mathematical model to formulate the algorithm, which can easily be implemented by a digital computer. These features are appreciated for nonlinear processes for which there is no reliable model and complex systems where the model is useless due to the large number of equations involved [3].

Published online: 7 October 2003

P. Melin (✉), O. Castillo
Department of Computer Science,
Tijuana Institute of Technology,
P.O. Box 4207, Chula Vista, CA 91909, USA
E-mail: pmelin@tectijuana.mx

We would like to thank the Research Grant Committee of Consejo Nacional del Sistema de Educacion Tecnologica (COSNET) and Consejo Nacional de Ciencia y Tecnologia (CONACYT) for the financial support for performing this research work. We would also like to thank the Department of Computer Science of Tijuana Institute of Technology for the time and resources given to this project.

Additionally, fuzzy logic is used more frequently for the control of electrical machines such as direct current or induction motors [10]. Nevertheless, the main problem with fuzzy logic is that there is no systematic procedure for the design of a fuzzy controller. For this reason, we propose in this paper the use of the adaptive neuro-fuzzy inference system (ANFIS) methodology [7, 8] to adapt the parameters of the fuzzy system for control [2, 4]. We use this neuro-fuzzy approach to train the controller with real data about the problem.

2
Basic concepts of stepping motors

Stepping motors come in two varieties, *permanent magnet* and *variable reluctance* (there are also *hybrid* motors, which are indistinguishable from permanent magnet motors from the controller's point of view). You can distinguish between the two varieties with an ohmmeter. Variable reluctance motors usually have three (sometimes four) windings, with a common return, while permanent magnet motors usually have two independent windings, with or without center taps. Center-tapped windings are used in unipolar permanent magnet motors [9]. Stepping motors come in a wide range of angular resolution. The coarsest motors typically turn 90° per step, while high-resolution permanent magnet motors are commonly able to handle 1.8 or even 0.72° per step. With an appropriate controller, most permanent magnet and hybrid motors can be run in half-steps, and some controllers can handle smaller fractional steps or micro-steps.

A. Variable reluctance motors

If your motor has three windings, typically connected as shown in the schematic diagram in Fig. 1, with one terminal common to all windings, it is most likely a variable reluctance stepping motor. In use, the common wire typically goes to the positive supply and the windings are energized in sequence. The cross section shown in Fig. 1 is of 30° per step variable reluctance motor. The rotor in this motor has 4 teeth and the stator has 6 poles, with each winding wrapped around two opposite poles. With winding number 1 energized, the rotor teeth marked X are attracted to this winding's poles. If the current through winding 1 is turned off and winding 2 is turned on, the rotor will rotate 30° clockwise so that the poles marked Y line up with the poles marked 2.

There are also variable reluctance stepping motors with 4 and 5 windings, requiring 5 or 6 wires. The principle for driving these motors is the same as that for the three winding variety, but it becomes important to work out the correct order to energize the windings to make the motor step nicely. The motor geometry illustrated in Fig. 1, giving 30° per step, uses the fewest number of rotor teeth and stator poles that performs satisfactorily. Using more motor poles and more rotor teeth allows construction of motors with smaller step angle.

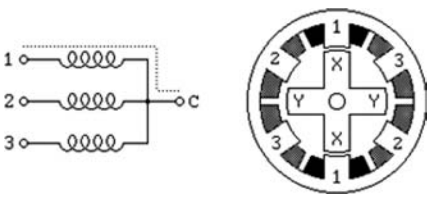

Fig. 1. Example of a variable reluctance motor

B. Unipolar motors

Unipolar stepping motors, both Permanent magnet and hybrid stepping motors with 5 or 6 wires are usually wired as shown in the schematic in Fig. 2, with a center tap on each of two windings. In use, the center taps of the windings are typically wired to the positive supply, and the two ends of each winding are alternately grounded to reverse the direction of the field provided by that winding. The motor cross-section shown in Fig. 2 is of a 30° per step permanent magnet or hybrid motor.

For higher angular resolutions, the rotor must have proportionally more poles. The 30° per step motor in the figure is one of the most common permanent magnet motor designs, although 15 and 7.5° per step motors are widely available.

C. Bipolar motors

Bipolar permanent magnet and hybrid motors are constructed with exactly the same mechanism as is used on unipolar motors, but the two windings are wired more simply, with no center taps. The schematic in Fig. 3 shows how such a motor is wired, while the motor cross section shown here is exactly the same as the cross section shown in Fig. 2.

3
Dynamics of the stepping motor

Each time you step the motor, you electronically move the equilibrium position S radians. This moves the entire curve a distance of S Rad, as shown in Fig. 4.

The first thing to note about the process of taking one step is that the maximum available torque is at a minimum

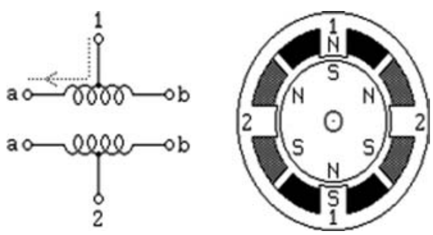

Fig. 2. Example of a unipolar motor

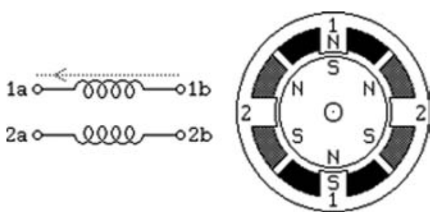

Fig. 3. Example of a bipolar motor

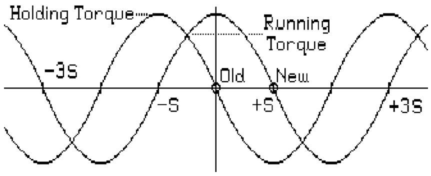

Fig. 4. Dynamics of a stepping motor

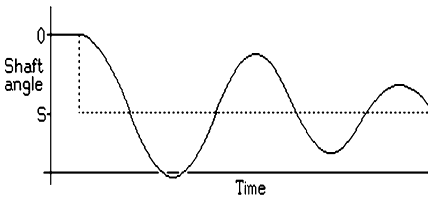

Fig. 5. Trajectory of the motor rotor

when the rotor is halfway from one step to the next. This minimum determines the *running torque*, the maximum torque the motor can drive as it steps slowly forward. For common two-winding permanent magnet motors with ideal sinusoidal torque versus position curves and holding torque h, this will be $h/(2^{0.5})$. If the motor is stepped by powering two windings at a time, the running torque of an ideal two-winding permanent magnet motor will be the same as the single-winding holding torque.

It should be noted that at higher stepping speeds, the running torque is sometimes defined as the *pull-out torque*. The resulting trajectory may resemble the one shown in Fig. 5.

The resonant frequency of the motor rotor depends on the amplitude of the oscillation; but as the amplitude decreases, the resonant frequency rises to a well-defined small-amplitude frequency. Formally, the small-amplitude resonance can be computed as follows:

$$T = \mu A$$

where: T is the torque applied to rotor, μ the moment of inertia of rotor and load, and A the angular acceleration, in Rad/s. We assume that, for small amplitudes, the torque on the rotor can be approximated as a linear function of the displacement from the equilibrium position. Therefore, Hooke's law applies:

$$T = -k\Theta$$

where k is the "spring constant" of the system, in torque units per radian,Θ the angular position of rotor, in radians.

We can equate the two formulas for the torque to get:

$$\mu A = -k\Theta$$

Note that acceleration is the second derivative of position with respect to time:

$$A = d^2\Theta/\mathrm{d}t^2$$

So we can rewrite this the above in differential equation form:

$$\mathrm{d}^2\Theta/\mathrm{d}t^2 = -(k/\mu)\Theta$$

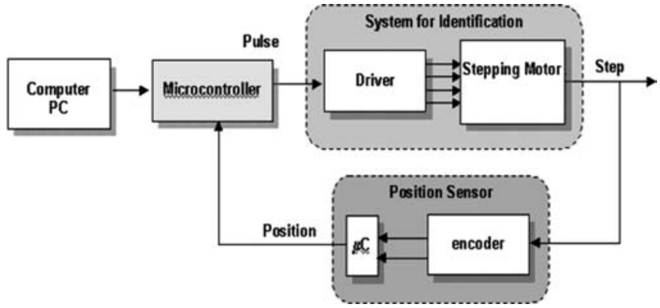

Fig. 6. Block diagram of the experimental system

Fig. 7. Motor Vexta PV266-01E

To solve this, recall that, for: $f(t) = a \sin bt$ the derivatives are:

$$\mathrm{d}f(t)/\mathrm{d}t = ab \cos bt$$

$$d^2f(t)/\mathrm{d}t^2 = -ab^2 \sin bt = -b^2 f(t)$$

Note that, throughout this discussion, we assumed that the rotor is resonating. Therefore, it has an equation of motion something like:

where a is the angular amplitude of resonance, f the resonant frequency.

This is an admissible solution to the above differential equation if we agree that:

$$b = 2\pi f$$

$$b^2 = k/\mu$$

In practice, this oscillation can cause significant problems when the stepping rate is anywhere near a resonant frequency of the system; the result frequently appears as random and uncontrollable motion.

4
Fuzzy logic controller of the stepping motor

The fuzzy logic controller provides an algorithm, which converts the linguistic control, based on expert knowledge into an automatic control strategy [14]. Therefore, the fuzzy logic algorithm is much closer in spirit to human thinking than traditional logical systems [5, 15]. The main problem with fuzzy logic controller generation is related to the choice of the regulator parameters [12]. For this reason, we apply the ANFIS methodology to adapt the parameters of the fuzzy controller according to real data about the problem [13]. The experiments were made on a system described in Fig. 6. A computer program in the PC generates the step input to the system and stores the response. The fuzzy controller is also contained in the PC and acts on the system according to the corresponding responses. We now describe briefly the modules of the system.

1. Motor/Driver: Micro-step motor Vexta PV266-01E with five phases and 500 steps by turn (The motor is shown in Fig. 7). Power driver Vexta DFR1514A with multi-resolution (Minimum: 500 steps by turn; Maximum: 125000 steps by turn).
2. Encoder: Optical encoder Bourns of 40000 steps by turn. This encoder generates two square signals with 90° difference. With these signals the magnitude of motor movement is determined. We use the programmable logic device (PLD) of Altera (EP5032) to determine the movement of the motor.
3. Data Acquisition Card: PCL-818 of Advantech with 8 analog inputs and 2 analog outputs (12 bits), 16 digital inputs, 16 digital outputs. The sampling time used is 0.25 ms.
4. Computer/Software: Pentium III with 733 MHz. We design a small real time kernel in C language for control and data acquisition, and the fuzzy controller was programmed in MATLAB [16].

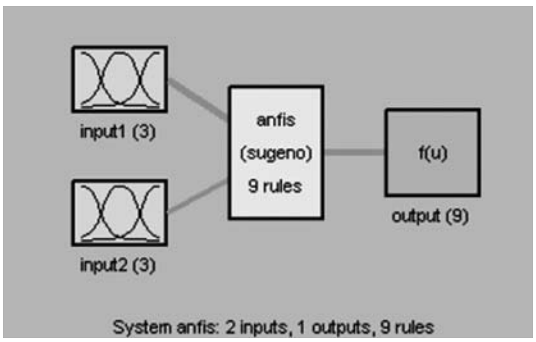

Fig. 8. Architecture of the Sugeno fuzzy system with the ANFIS approach

The linguistic control rules are established considering the dynamic behavior of the stepping motor drive and analyzing the error and its variation. These control rules are expressed as follows:

If Error is LP and Change_Error is LP

$$\text{Then Speed} = p_1{*}\text{Error} + q_1 * \text{Change\_Error} + r_1$$

If Error is LP and Change_Error is MP

$$\text{Then Speed} = p_2{*}\text{Error} + q_2 * \text{Change\_Error} + r_2 \ldots$$

This is a Sugeno fuzzy model [17, 19] for controlling the stepping motor. We used the ANFIS methodology to estimate the parameters of the membership functions and the consequent functions. We used a fuzzy model of 9 rules and 3 membership functions for each linguistic variable. This was the fuzzy controller that gave the best results. We show in Fig. 8 the architecture of the fuzzy system with the ANFIS approach. We also show in Fig. 9 the architecture of the ANFIS methodology, which shows the fuzzy rules in the adaptive network.

The fuzzy rules generated by the ANFIS method are shown in Fig. 10. These rules are generated automatically with the ANFIS method. We also show in Fig. 11 the membership functions generated automatically by ANFIS. In Fig. 12 we show the non-linear surface of the fuzzy model.

Finally, we show in Fig. 13 the fuzzy rule viewer of MATLAB, which shows the use of the fuzzy system for calculating the output of the model for specific input values.

```
1. If (input1 is in1mf1) and (input2 is in2mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf1) and (input2 is in2mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf1) and (input2 is in2mf3) then (output is out1mf3) (1)
4. If (input1 is in1mf2) and (input2 is in2mf1) then (output is out1mf4) (1)
5. If (input1 is in1mf2) and (input2 is in2mf2) then (output is out1mf5) (1)
6. If (input1 is in1mf2) and (input2 is in2mf3) then (output is out1mf6) (1)
7. If (input1 is in1mf3) and (input2 is in2mf1) then (output is out1mf7) (1)
8. If (input1 is in1mf3) and (input2 is in2mf2) then (output is out1mf8) (1)
9. If (input1 is in1mf3) and (input2 is in2mf3) then (output is out1mf9) (1)
```

Fig. 10. Fuzzy rules generated by the ANFIS method

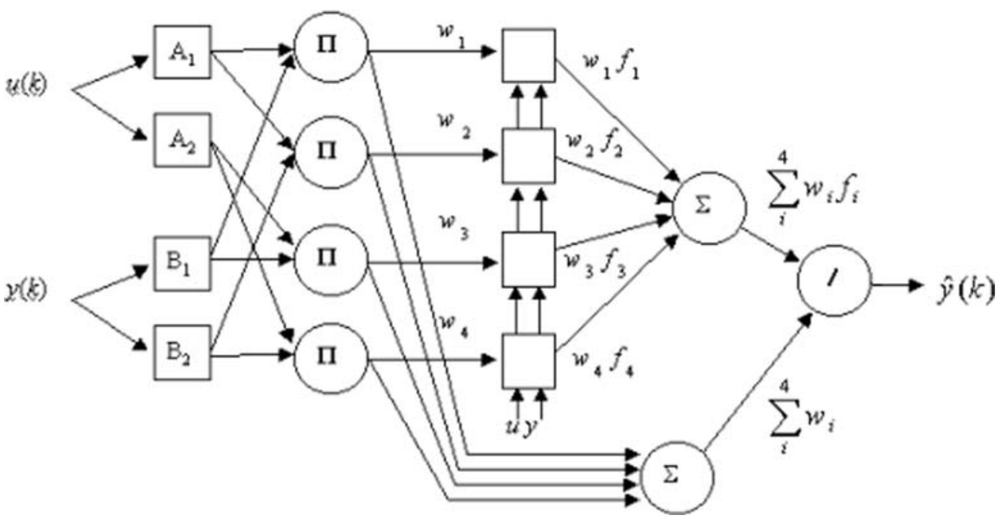

Fig. 9. The ANFIS architecture showing the fuzzy system components

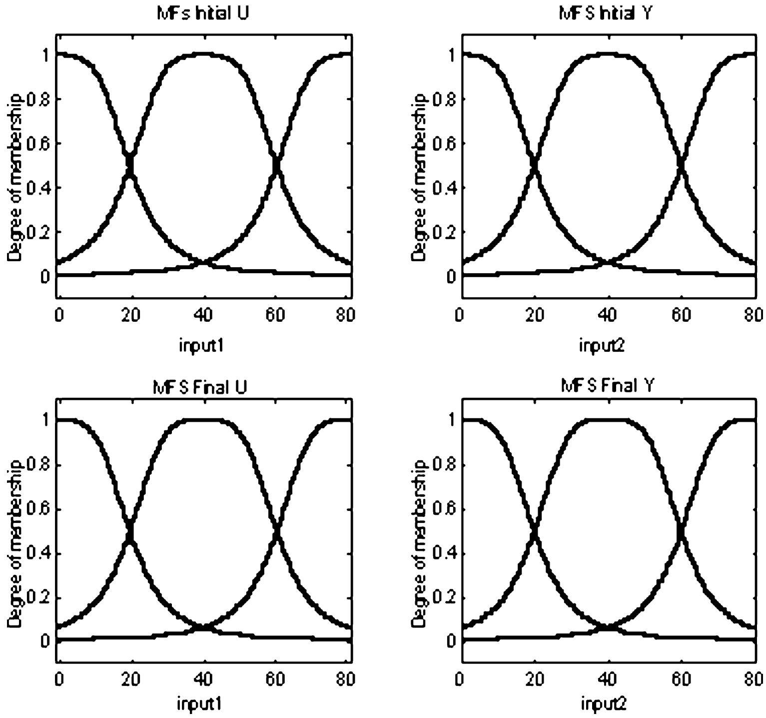

Fig. 11. Membership functions generated by the ANFIS method

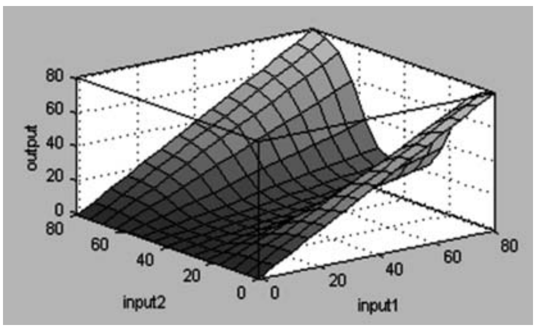

Fig. 12. Non-linear surface of the Sugeno fuzzy model

5 Hardware implementation of ANFIS

In this section, a detailed description of the hardware implementation is given. The ANFIS controller implementation is needed to really achieve the goal of having an intelligent stepping motor. The neuro-fuzzy approach gives to the motor the ability to adapt to changing conditions in the environment. A stepping motor equipped with the intelligent controller will be able to train itself with the new data to update its parameters, and as a consequence be able to change its behavior accordingly.

We have to first mention that the ANFIS implementation required a specific microprocessor to be able to achieve all of the required numerical calculations. The ANFIS methodology requires that the training is performed on an adaptive network, using as data a time series of the relevant variables of the problem. For this reason, the "Jstamp" microprocessor was selected. This specific microprocessor uses JAVA as the native programming language, and has 512 Kb of RAM memory and 512 Kb of FLASH memory. We show in Fig. 14 the Jstamp microprocessor that was used in our research work.

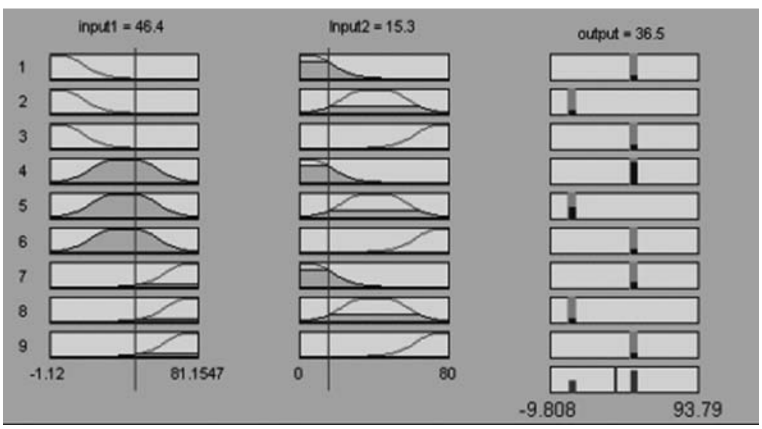

Fig. 13. Fuzzy rule viewer for calculating the output of the fuzzy system for specific values

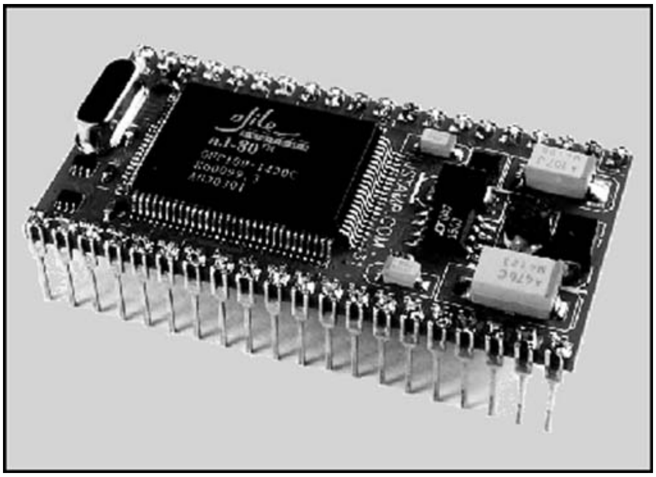

Fig. 14. The Jstamp microprocessor used for the implementation

The ANFIS architecture that was implemented has the general form that is shown in Fig. 15. In this figure, for simplicity we only show the case of 4 fuzzy rules, but the structure is similar for 9 fuzzy rules. We can also appreciate from this Figure the hybrid nature of the ANFIS approach because a least squares method is used in the forward direction and the backpropagation algorithm is used in the backward direction [18, 21]. These methods were implemented in the JAVA language and then they were downloaded to the Jstamp microprocessor to obtain the fuzzy controller.

We also used a micro-controller for detecting the position of the encoder. The specific type of the micro-controller is the SX28 from the Ubicom company. The SX28 is a micro-controller based on Flash memory and with RISC type architecture. We show in Fig. 16 the block diagram of the connections for detecting the position of the encoder.

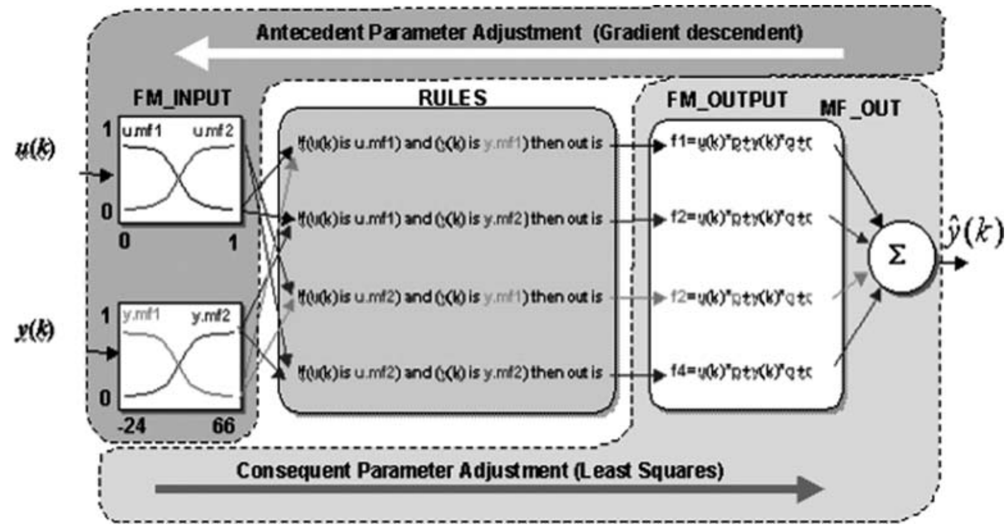

Fig. 15. ANFIS architecture showing the inputs and outputs of the system

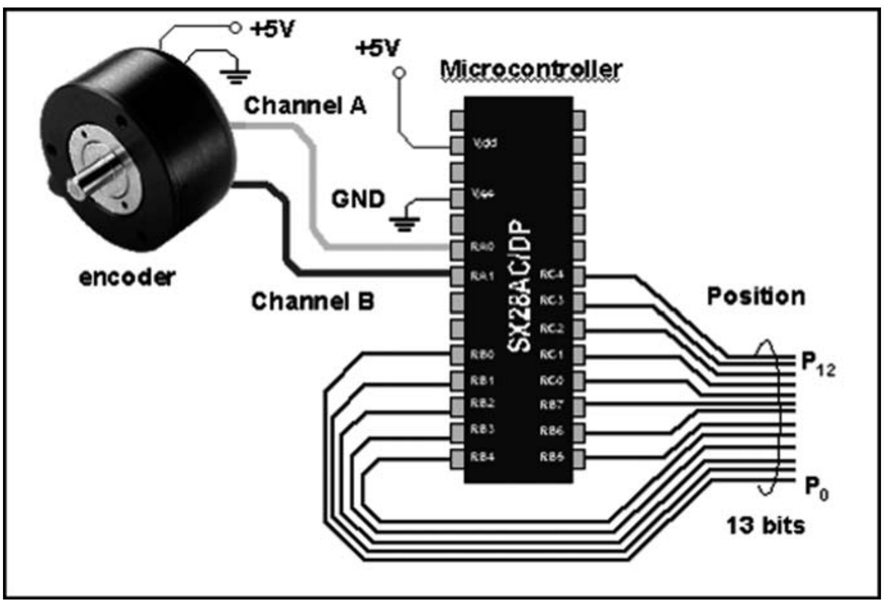

Fig. 16. Block diagram showing the connections of the micro-controller

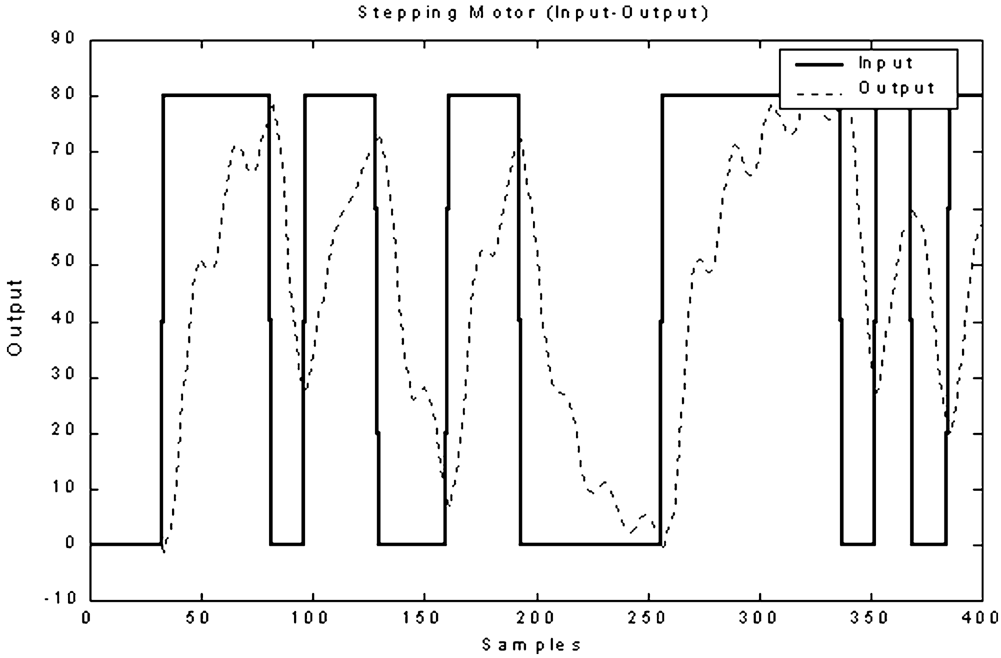

Fig. 17. Response of the stepping motor to a sequence of step input signals

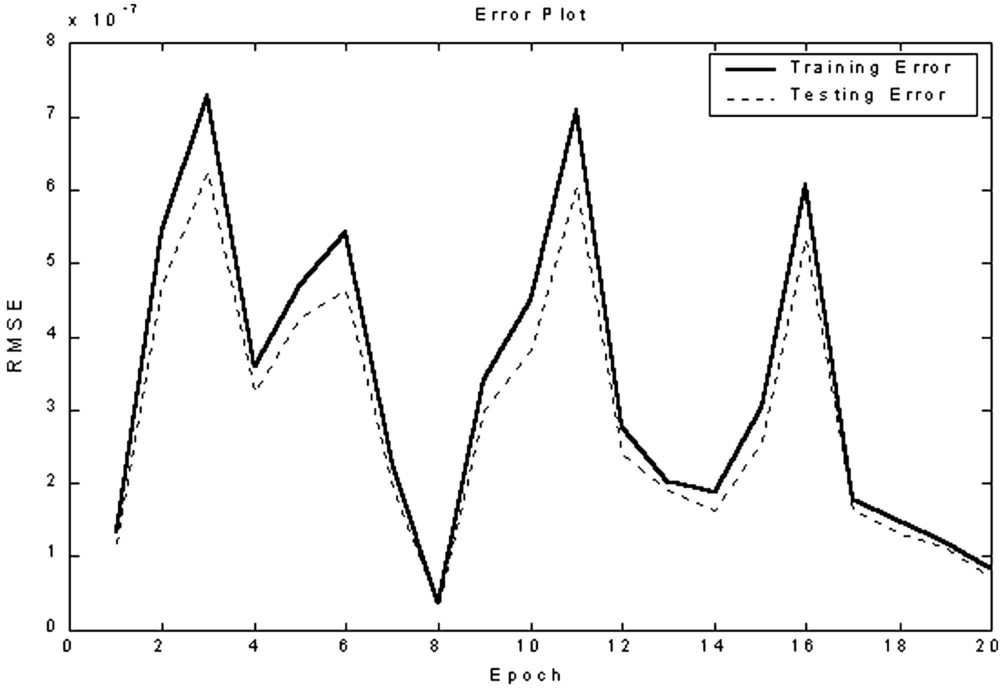

Fig. 18. Results of applying the ANFIS methodology to the training and testing data

6
Experimental results

In this section, the tracking and adaptability features of the fuzzy control applied to the stepping motor are tested using simulation and experimentation. We first show in Fig. 17 the response of the stepping motor to a sequence of step input signals (we use 400 samples). We show in Fig. 18 the results of applying the ANFIS methodology with the training data and with the testing data. We used 20 epochs for training and the final error was of 0.000001, which is very good for this application. In Fig. 19 we plot the predicted values by the fuzzy model and the real values for the system, and the curves are practically indistinguishable. Finally, we show in Fig. 20 a plot of the difference between both the real and the estimated signal by the fuzzy model.

We also compared our results with a classical PID controller and with a fuzzy Mamdani controller, to measure how much the adaptive fuzzy approach could improve the performance. Of course, our fuzzy controller (designed with ANFIS) was better in tracking and adaptability than the other controllers.

Another advantage of this method over classical quantitative controllers is that it does not require a fixed sampling time. Therefore, the proposed design confirms the fact that fuzzy control is relevant to the control fast of

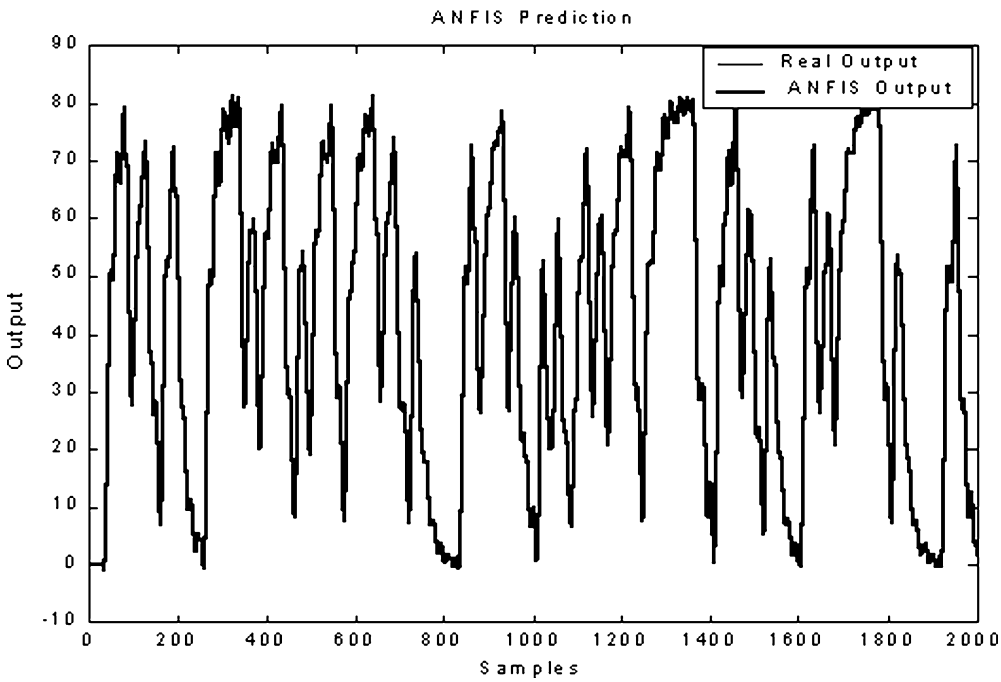

Fig. 19. Predicted values of the fuzzy model compared against the real values

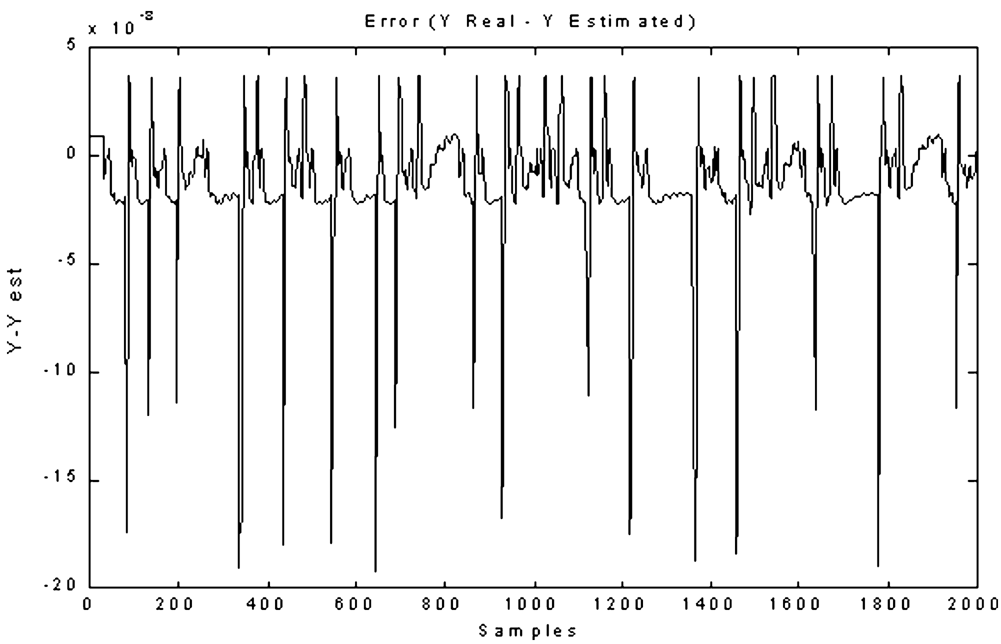

Fig. 20. Difference between the real and the estimated signal

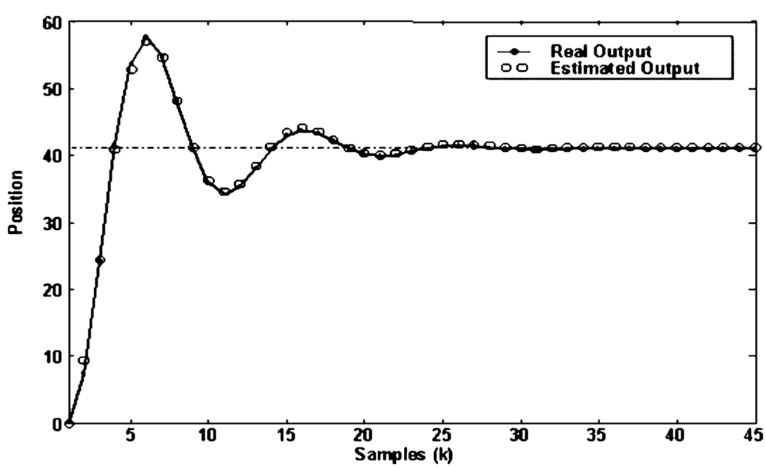

Fig. 21. Comparison between real and estimated outputs

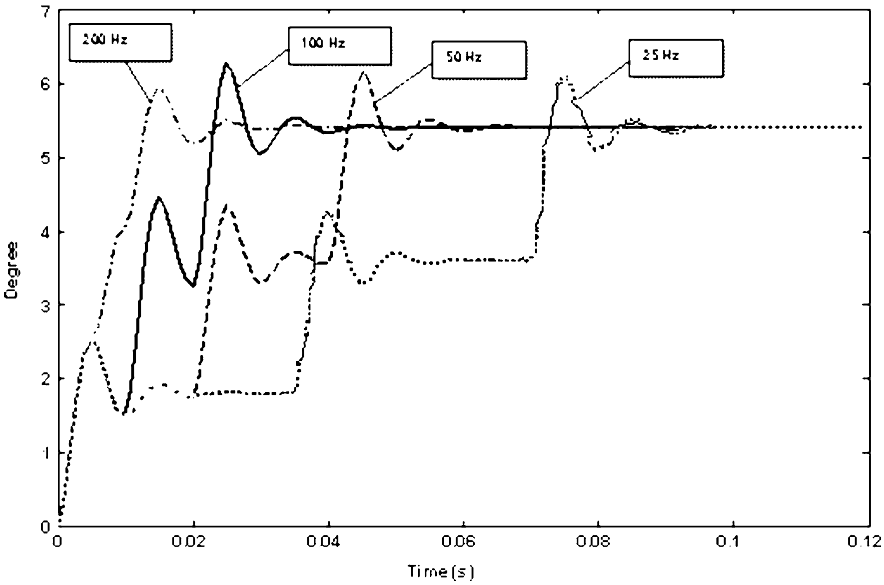

Fig. 22. Case of three consecutive steps at different frequencies

nonlinear processes such as stepping motor drives where quantitative methods are not always appropriate.

We also show in Fig. 21 a comparison between the real and estimated outputs of the stepping motor for one step of the system. It is clear from this figure that the fuzzy model is very close to the real experimental values. Finally, we show in Fig. 22 the case of making three consecutive steps at different frequencies. We also have good results for the case of three steps, which shows the robustness of our intelligent control approach.

7
Conclusions

In this paper, the feasibility of fuzzy control for stepping motor drives has been proved and illustrated by simulation and experimentation. The best parameters for the fuzzy controller were determined by using the ANFIS methodology and also by using simulations of the stepping motor dynamics. An experimental system was used to validate experimentally the tracking ability and the insensibility to plant parameter changes. The fuzzy controller presented very interesting tracking features and was able to respond to different dynamic conditions. Also, the fuzzy control computation is very inexpensive, and this regulator could be used for the control of machine tools and robotics manipulators [6, 20] without significantly increasing the cost of the drive. The only extra cost is for the optical encoder. Another advantage of this method over classical quantitative controllers is that it does not require a fixed sampling time. Therefore, the proposed design confirms the fact that fuzzy control is relevant to the control fast of nonlinear processes such as stepping motor drives where quantitative methods are not always appropriate.

References

1. **Betin F, Pinchon D, Capolino GA** (2000) Fuzzy logic applied to speed control of a motor drive, IEEE Trans. Industrial Electronics **47**: 610–622
2. **Castillo O, Melin P** (1998) A new Fuzzy-Fractal-genetic method for automated mathematical modelling and simulation of robotic dynamic systems, Proc. of FUZZ'98, Anchorage, Alaska, **2**: 1182–1187
3. **Castillo O, Melin P** (1999) Automated mathematical modelling, simulation and behavior identification of robotic dynamic systems using a fuzzy-fractal-genetic approach, J. Robotics and Autonomous Sys, **28**(1): 19–30
4. **Castillo O, Melin P** (2001) Soft Computing for Control of Non-Linear Dynamical Systems. Springer-Verlag, Heidelberg, Germany
5. **Castillo O, Melin P** (2002) A Hybrid fuzzy-fractal approach for time series analysis and plant monitoring, Int. J. Int. Syst **17**(8): 751–765
6. **Fu KS, Gonzalez RC, Lee CSG** (1987) Robotics: Control, Sensing, Vision and Intelligence. McGraw-Hill, New York, NY, USA
7. **Jang JSR, Sun CT, Mizutani E** (1997) Neuro-fuzzy and soft computing, Prentice Hall New York, NY, USA
8. **Jang JSR** (1993) ANFIS: adaptive-network-based fuzzy inference systems IEEE Trans Systems, Man, and Cybernetics **23**: 665–685
9. **Kenjo T, Sugawara A** (1994) Stepping motors. Oxford Science Publications, London, Great Britain
10. **Li YF, Lau CC** (1989) Developments of fuzzy algorithms for servo systems IEEE Control System Magazine **9**: 65–72
11. **Mamdani EH** (1974) Application of fuzzy algorithms for control of a simple dynamic plant, Proc. of IEEE **121**: 1585–1588
12. **Melin P, Castillo O** (2001) A new method for adaptive model-based control of non-linear dynamic plants using a neuro-fuzzy-fractal approach, Soft Computing J **5**(2): 171–177
13. **Melin P, Castillo O** (2001) Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach, IEEE Transactions on Industrial Electronics **48**(5): 951–955
14. **Melin P, Castillo O** (2002) Modelling, simulation and control of non-linear dynamical systems, taylor and francis publishers, London, Great Britain
15. **Melin P, Castillo O** (2002) Adaptive model-based control of non-linear dynamical systems with a neuro-fuzzy-genetic approach, International Journal Smart Engineering System Design **4**(1): 41–47
16. **Nakamura S** (1997) Numerical Analysis and Graphic Visualization with MATLAB, Prentice-Hall, New York, NY, USA
17. **Sugeno M, Kang GT** (1988) Structure identification of fuzzy model, Fuzzy Sets and Systems **28**: 15–33

18. **Sun CT** (1994) Rulebase Structure Identification in an adaptive network based fuzzy inference system, IEEE Transactions on Fuzzy Systems **2**(1): 64–73
19. **Takagi T, Sugeno M** (1985) Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics **15**: 116–132
20. **Yager RR, Filev DP** (1994) Essentials of Fuzzy Modeling and Control, John Wiley and Sons New York, NY, USA
21. **Yen J, Langari R** (1999) Fuzzy Logic: Intelligence, Control, and Information, Prentice-Hall, New York, NY, USA
22. **Zadeh LA** (1965) Fuzzy Sets, Information and Control **8**: 338–353
23. **Zadeh LA** (1971) Quantitative fuzzy semantics, J Information Sci **3**: 159–176
24. **Zadeh LA** (1973) Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Transactions on Systems, Man, and Cybernetics, **3**(1): 28–44
25. **Zadeh LA** (1975) The concept of a linguistic variable and its application to approximate reasoning, J Information Sci **8**: 43–80