

FLC IMPLEMENTATION ON A 8-BIT MICROCONTROLLER FOR DC MOTOR SPEED AND POSITION CONTROL

Mohamed A. Darwish, Hossam S. Abbas, Awad I. Saleh, and Mohamed M. M. Hassan

Electrical Engineering Department, Assiut University, Faculty of Engineering, Assiut, Egypt – Postal Code: 71518

E-mail: mohamed_darwish_84@yahoo.com

(Received March 21, 2011 Accepted March 26, 2011)

This paper presents the design and experimental implementation of a Fuzzy logic controller (FLC) for a DC servomotor speed and position control. The motivation to utilize the FLC is its robustness against model's parameters inaccuracy and uncertainty. The implementation of the FLC algorithm is carried out by a low cost 8-bit microcontroller instead of using expensive general purpose microprocessors which are commonly employed in practice. This leads to a reasonable hardware cost for such applications. The experimental results in terms of reference tracking and disturbance rejection show high performance with the FLC approach in comparison with PI and PD controllers designed for the same purposes.

KEYWORDS: *DC motor control, Fuzzy logic controller, Mamdani Type Fuzzy Controller, Microcontroller*

NOMENCLATURE

e_b	Back electromotive force (EMF) [Volt]	R	Armature resistance [Ohm]
i_a	Armature current [Ampere]	T_d	Disturbance torque [N.m]
J	Moment of inertia of the motor [Kg.m ²]	T_m	Motor torque [N.m]
K_b	Back emf constant [Volt/rad/sec]	V_a	Armature voltage [Volt]
K_f	Viscous friction coefficient [N.m.s]	Greek Symbols:	
K_m	Torque constant [N.m/Ampere]	θ	Angular position of the motor shaft [rad]
L	Armature inductance [Henry]	ω	Angular speed of the motor shaft [rad/sec]

1- INTRODUCTION

Modern control techniques such as Variable structure control, H ∞ , e.g. [1-4], usually are designed based on an accurate mathematical representation of the plant to be controlled. Constructing such accurate representation requires considerable efforts,

furthermore, it needs specialists to model complex systems. An alternative way is to utilize experimental modeling techniques, i.e. system identification [5]; however system identification approaches depend on designing rich excitation signals which are capable to excite the dynamics of the underlying process, this is not always possible in practice, especially in the case of systems under restricted input and output variations (for safety demand). This has motivated the researchers to develop control approaches that don't rely on accurate models of the plant. Fuzzy logic controllers (FLCs) are an important example of such approaches. Fuzzy logic control, introduced by Lotfi Zadeh in 1965 [6], has been successfully applied to a large number of control applications such as subway system control, nuclear reactor control, traffic control, domestic appliances, and power systems [7, 8, 9].

Implementation of FLC is accomplished by digital computer; for practical implementation several hardware techniques have been introduced in the literature, such as Digital Signal Processor [10, 11], general purpose microprocessors [12] and field Programmable Gate Array [13]. To provide high speed computation of FLC, dedicated fuzzy chips are employed [14]; however, they are unable to be reprogrammed or upgraded.

In this paper we implement Fuzzy logic controllers on a low cost microcontroller, PIC 16F877A, which possesses several features in comparison with the above mentioned techniques, such as: it is able to easily code and recode any control algorithm, it uses high level languages such as C, it has built-in Analogue to Digital A/D converters, timers for precise adjustment of sampling time, reasonable storage memory and the capability to arbitrary select input and output paths. All these are usually offered in reasonable prices, thus efforts, money and time can be saved [15, 16, 17].

DC motors are widely used in many industrial applications such as electric vehicles, steel rolling mills, electric cranes, robotic manipulators, etc. [18,19,20] due to their high reliability and simplicity to be controlled. The nonlinear characteristics of the DC motor such as saturation and friction make it a suitable platform for applying the FLC techniques. In this work, FLC design and implementation on a low cost 8-bit microcontroller is presented for DC motor speed and position control.

This paper is organized as follows: Section 2 briefly reviews the Generic Mamdani Type Fuzzy Controller utilized here. Section 3 describes the used DC motor model in this work. The design of FLC controller for DC motor speed and position control is introduced in Section 4. In Section 5 a hardware interface circuit and software algorithms for implementing FLC controller are described. Section 6 shows the real time experimental results with load and no load condition. Finally, conclusions and future work are drawn in Section 7.

2- PRELIMINARIES

In this section Mamdani architecture for fuzzy control is reviewed [21]. The basic assumption of fuzzy logic control approach proposed by E. H. Mamdani in 1974 is that with the absence of an explicit plant model and/or clear statement of control design objectives informal knowledge about the operation of a given plant can be codified in terms of "if-then", or "condition-action", rules to form the basis of a linguistic control strategy.

The basic paradigm of fuzzy logic control, which operates in discrete time, is linguistic or rule-based control strategy of the form

If OA1 is – and OA2 is – and ... then CA1 is – and CA2 is –...

This maps the observable attributes (OA1, OA2 ...) of a given system into its controllable attributes (CA1, CA2 ...), as is shown in Fig. 1.

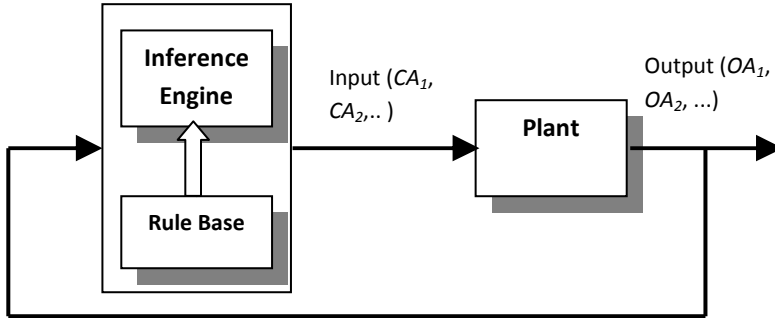


Fig.1: Architecture for Fuzzy Control

Consider the closed loop system shown in Fig.2, where the plant is linear, minimal and stable; assume no explicit information about the plant model. Now it is required to develop a control strategy that maps an error e into a control action u as shown in Fig. 3 to achieve specific control objectives. The core of a generic Mamdani type fuzzy controller is a fuzzy logic control algorithm that maps the normalized values of the error, $e_n(k)$, at instant k and the backward difference of the error, $\delta e_n(k)$, at instant k into a change in the control signal $\delta u(k)$ at instant k , where $e_n(k)$ and $\delta e_n(k)$ are given as

$$\begin{aligned} e_n(k) &= n_e e(k) \\ \delta e_n(k) &= n_{\delta e} (e(k) - e(k-1)) \end{aligned} \tag{1}$$

Where n_e and $n_{\delta e}$ are the corresponding normalization factors, ($n_e = n_{\delta e} = 1/\max|y_d - y|$).

The fuzzy logic control algorithm is based on rules of the form

if $e(k)$ is P and $\delta e(k)$ is N then $\delta u(k)$ is Z ,

where P, N, and Z stand for Positive, Negative, and Zero, respectively. P, N and Z are defined as membership functions of the relevant variables, i.e. $e_n(k)$, $\delta e_n(k)$, $\delta u(k)$, as shown in Fig. 4c, see [21] for more details. Fig. 5 shows an example about how degree of membership for two rules with two inputs is evaluated.

The instantaneous value of the control signal $u(k)$, Fig. 3, is computed in terms of $\delta u(k)$ as:

$$u(k) = u(k-1) + dn_{\delta u} \delta u(k)$$

where $dn_{\delta u}$ is as a denormalization factor given as $dn_{\delta u} = 1/\bar{u}$, where \bar{u} is an upper bound on the magnitude of the input to the plant, which defined by the user.

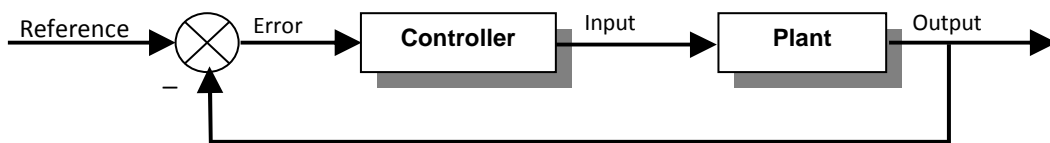


Fig.2: A Feedback Control System

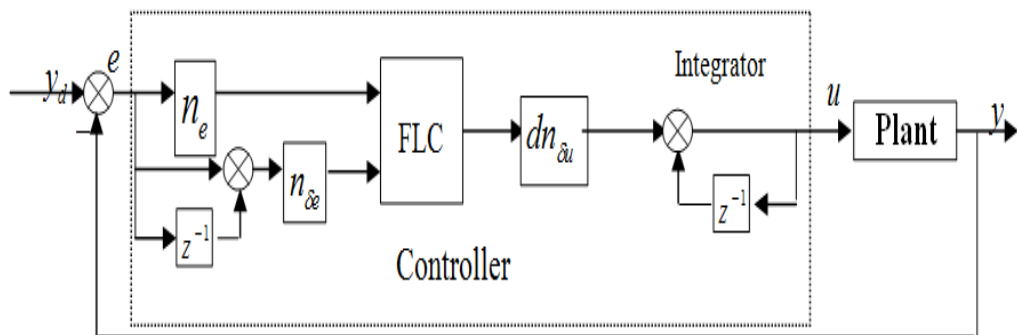


Fig.3: Architecture of the Generic Fuzzy Control System

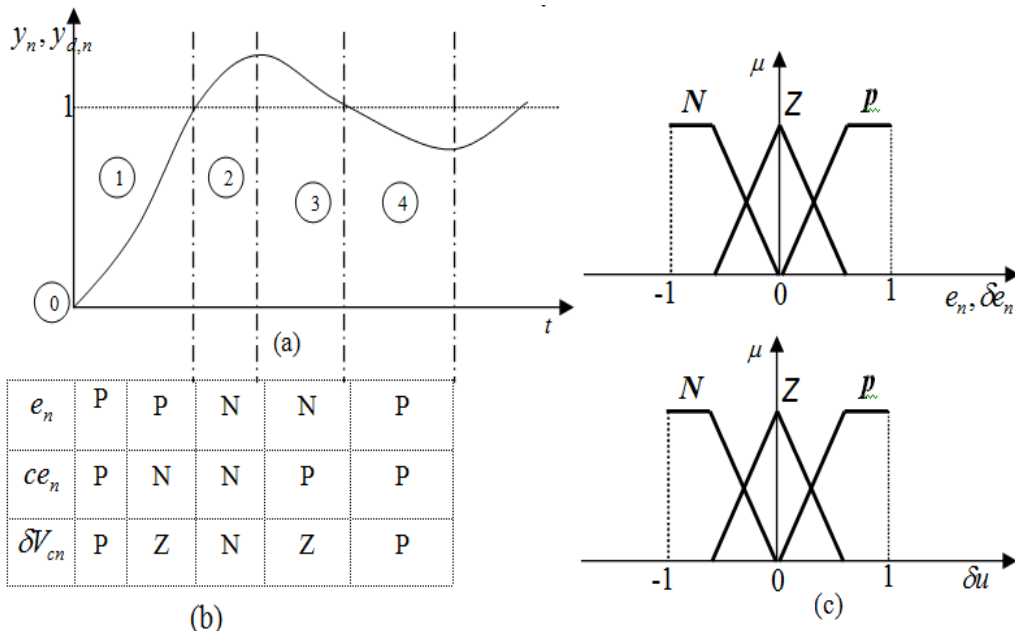


Fig. 4: Control Rules for Simple Generic Fuzzy Controller

The algorithm of the fuzzy rule-based inference consists of the following steps.

1. Fuzzy Matching: Calculate the degree to which the input data match the condition of the fuzzy rule.
2. Inference (implication): Calculate the rule's conclusion based on its matching degree.

3. Combination: Combine the conclusions inferred by all fuzzy rules into a final conclusion.
4. Defuzzification (optional): for applications that need a crisp output (e.g., in control systems), an additional step is used to convert the fuzzy conclusion to a crisp one.

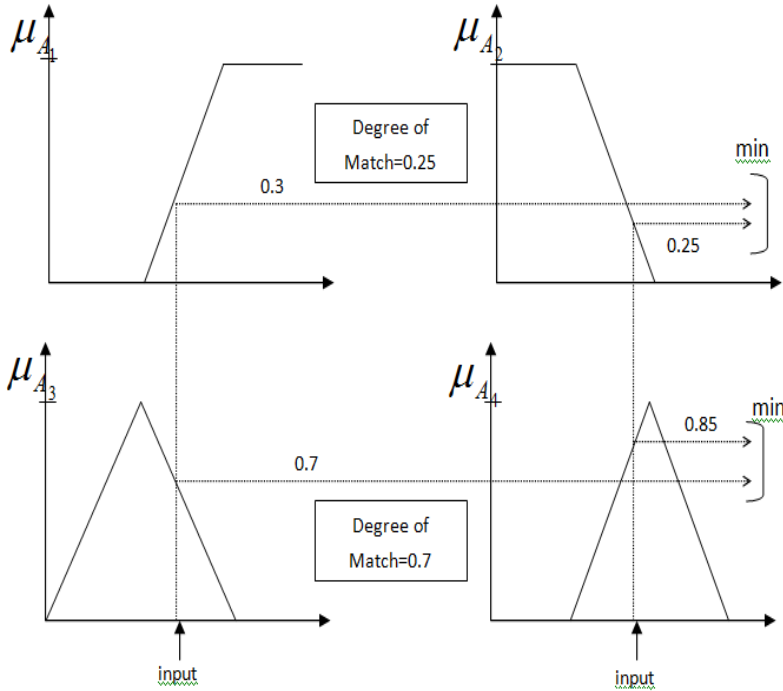


Fig. 5: Fuzzy Matching

Next an example is given to show how the rules are derived based on the step response of a system. The rules effectively reflect a typical scenario of the operation of the system as depicted in Fig. 4a, provided that the sampling time is properly chosen. The step response can be divided into five regions, first the starting state is marked as (0) in Fig 4a, this corresponds to $e_n(k)$ and $\delta e_n(k)$ being positive. It evidently makes sense to suggest that $\delta u(k)$ to be positive as well, therefore the output $y_n(k)$ will rise, and approaches the second region (marked as (1) in Fig. 4a). In region (1), $e_n(k)$ is still positive whereas $\delta e_n(k)$ is negative, one may suggest that $\delta u(k)$ should be zero. Continuing with this approach for the other regions, one can gradually determine by inspection the necessary rules to complete the rule set as shown in Fig. 4b, these rule set is tabulated in Table 1.

3- SYSTEM DESCRIPTION AND MODELING

In this section a brief description of the DC motor used here is given. Then a mathematical model to represent the motor is derived.

Table 1: Rules for the Generic Fuzzy Controller

Observable Attributes	e_n
	δe_n
Controllable Attributes	δu
Condition \longrightarrow Action Rules	
I: Starting up; Change the input in response to the set-point change	<i>if e_n is P and δe_n is P then δu is P</i>
	<i>if e_n is N and δe_n is N then δu is N</i>
II: Plant is not responding; Adjust input	<i>if e_n is P and δe_n is Z then δu is P</i>
	<i>if e_n is N and δe_n is Z then δu is N</i>
III: Plant is responding normally; Keep input the same	<i>if e_n is P and δe_n is N then δu is Z</i>
	<i>if e_n is N and δe_n is P then δu is Z</i>
IV: Reached equilibrium	<i>if e_n is Z and δe_n is Z then δu is Z</i>
V: Error is nil but changing; Take action	<i>if e_n is Z and δe_n is N then δu is N</i>
	<i>if e_n is Z and δe_n is P then δu is P</i>

The DC motor used here is an ironless rotor servo motor manufactured by Mclennan Servo Supplies Ltd [22]. Its rated torque and speed are 1.5 N.cm and 2500 rpm, respectively.

The armature control DC motor shown in Fig. 6 is commonly described for speed control by the following set of differential equations:

$$\frac{di_a}{dt} = \frac{1}{L} v_a - \frac{R}{L} i_a - \frac{e_b}{L} \quad (2)$$

$$T_m = K_m i_a \quad (3)$$

$$e_b = K_b \omega \quad (4)$$

$$J\dot{\omega} = T_m - T_d - K_f \omega \quad (5)$$

$$\omega = \dot{\theta} \quad (6)$$

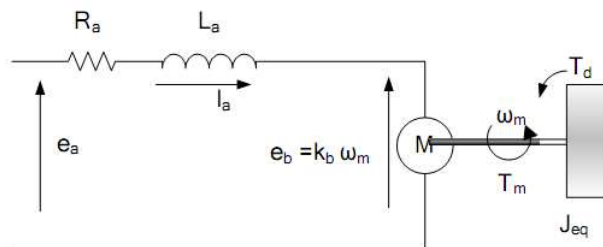


Fig. 6: Armature control DC motor model.

The transfer function representation of (2)-(6) is given as

$$\frac{\omega(s)}{v_a(s)} = \frac{k_m}{JLs^2 + (k_f L + JR)s + k_f R + k_b} \tag{7}$$

The values of the parameters R, Km, Kb, L, J are taken from the manufacturer manual as

$$R= 6.2 \text{ Ohms, } K_m=3.5 \text{ Ncm/A, } K_b=3.6 \text{ v/1000rpm, } L=0.8\text{mH, } J=0.039 \text{ Kgcm}^2.$$

It is not necessary to verify these parameters as a fuzzy control strategy is intended to be used, which not requires an accurate model. By this way a considerable effort to model the system and estimate its parameters is saved. Furthermore, a linear model is enough for the control purposes. For position control the system can be easily derived by integrating the angular speed in Eq. 6 to represent the angular position θ as a controlled output.

4- CONTROL SYNTHESIS

4.1- Speed Control

The fuzzy control strategy considered here is shown in Fig. 8a, with $e(k)$ is the error at instant k given as

$$e(k) = w_d(k) - w(k),$$

where w_d, w are the desired and measured speeds, respectively, and $\delta e(k)$ is the backward difference in the error at instant k, given as

$$\delta e(k) = e(k) - e(k - 1).$$

The fuzzy controller maps $e(k)$ and $\delta e(k)$ into $\delta u(k)$, which is the change in the control signal at instant k. To update the control signal $u(k)$, see Fig. 8a, the value of $\delta u(k)$ is added to a one step backward value of $u(k)$ to form the value of $u(k)$ at instant k as

$$u(k) = u(k - 1) + \delta u(k).$$

The rule by which $e(k)$ and $\delta e(k)$ are mapped into a change in the control signal $u(k)$ can be formulated as

$$\text{if } e(k) \text{ is } P \text{ and } \delta e(k) \text{ is } N \text{ then } \delta u(k) \text{ is } Z$$

where P, N, and Z are membership functions of the fuzzy sets of $e(k)$, $\delta e(k)$, and $u(k)$. These fuzzy sets are defined over the admissible values of $e(k)$, $\delta e(k)$ and $u(k)$ which are chosen according to the implementation constraints. For simplicity, triangular membership functions are considered here, see Fig. 9, a triangular membership function is described as

$$triangular(x : a, b, c) = \begin{cases} 0 & x < a \\ (x - a) / (b - a) & a \leq x \leq b, \\ (c - x) / (c - b) & b \leq x \leq c \\ 0 & x > c \end{cases} \quad (8)$$

Figure 7 illustrates an example of a triangular membership function, where a=10, b=20, c=30.

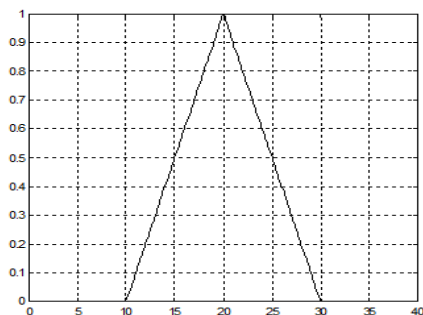


Fig. 7: Example of a membership (triangular function)

The membership functions of the error $e(k)$, the change in the error $\delta e(k)$ and the change in the control signal $\delta u(k)$ can be expressed in terms of Eq. 8, respectively, as:

$$N \equiv \begin{cases} 1 & e \leq -8 \\ -e/8 & -8 \leq e \leq 0 \\ 0 & e \geq 0 \end{cases}, P \equiv \begin{cases} 0 & e \leq 0 \\ e/8 & 0 \leq e \leq 8 \\ 1 & e \geq 8 \end{cases}, Z \equiv \begin{cases} 0 & e < -8 \\ (e/8)+1 & -8 \leq e \leq 0, \\ 1-(e/8) & 0 \leq e \leq 8 \\ 0 & e > 8 \end{cases}$$

$$N \equiv \begin{cases} 1 & \delta e \leq -4 \\ -\delta e/4 & -4 \leq \delta e \leq 0 \\ 0 & \delta e \geq 0 \end{cases}, P \equiv \begin{cases} 0 & \delta e \leq 0 \\ \delta e/4 & 0 \leq \delta e \leq 4 \\ 1 & \delta e \geq 4 \end{cases}, Z \equiv \begin{cases} 0 & \delta e < -4 \\ (\delta e/4)+1 & -4 \leq \delta e \leq 0, \\ 1-(\delta e/4) & 0 \leq \delta e \leq 4 \\ 0 & \delta e > 4 \end{cases}$$

$$N \equiv \begin{cases} 1 & \delta u \leq -8 \\ -\delta u/8 & -8 \leq \delta u \leq 0 \\ 0 & \delta u \geq 0 \end{cases}, P \equiv \begin{cases} 0 & \delta u \leq 0 \\ \delta u/8 & 0 \leq \delta u \leq 8 \\ 1 & \delta u \geq 8 \end{cases}, Z \equiv \begin{cases} 0 & \delta u < -0.5 \\ (\delta u/0.5)+1 & -0.5 \leq \delta u \leq 0. \\ 1-(\delta u/0.5) & 0 \leq \delta u \leq 0.5 \\ 0 & \delta u > 0.5 \end{cases}$$

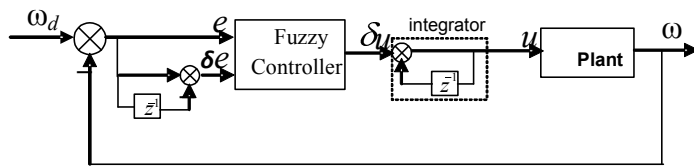
Fuzzy set: Each fuzzy set is chosen to consist of three linguistic terms (Z, N, and P).

Fuzzy conjunction (AND): We choose the minimum operator described by Eq. 9 as a fuzzy conjunction.

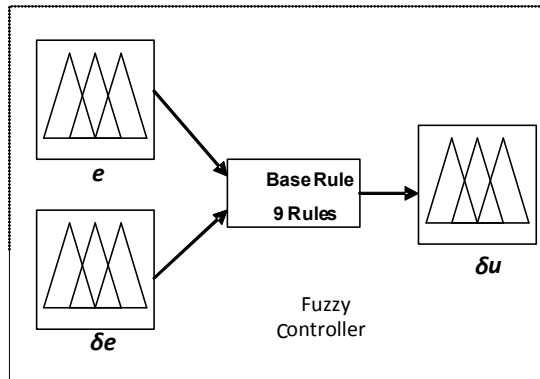
$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (9)$$

Fuzzy implication: We choose the scaling method as an implication method.

Fuzzy aggregation: We choose the scaling method as an aggregation method.



(a)



(b)

Fig. 8: (a) Block Diagram of Speed Control System
(b) Fuzzy Logic Controller

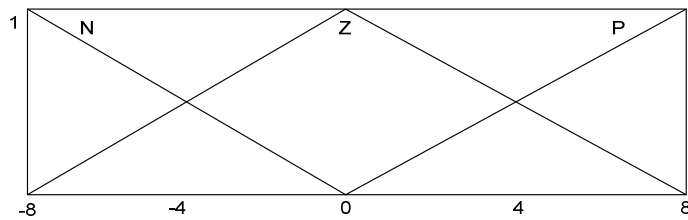


Fig. 9: (a) Fuzzy Set of Error

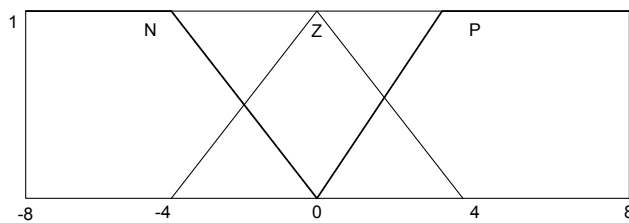


Fig. 9: (b) Fuzzy Set of Change in Error

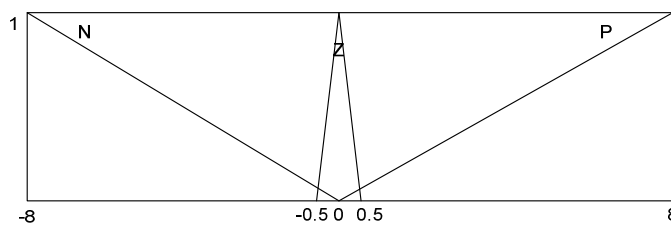


Fig. 9: (c) Fuzzy Set of Change in Control Voltage

Defuzzification: Center of area method (COA).

All the fuzzy sets and inference operation (conjunction, implication, aggregation, and defuzzification) are chosen by trial and error to improve the control action for all regions of operation of the DC motor.

Fuzzy inference rules: The rules of fuzzy controller is derived from the operation of DC motor, the entire rule is summarized in Table 2.

Table 2: Rules of Fuzzy Controller

$\delta e \backslash e$	Z	N	P
Z	Z	N	P
N	N	N	Z
P	P	Z	P

4.2- Position Control

The fuzzy logic control algorithm for the position control maps the values of $e(k)$, and $\delta e(k)$ into the control action $u(k)$, not into $\delta u(k)$ as in the case of speed control, which is equivalent to place the integrator in Fig. 8a after the output ω , therefore the output of the system becomes θ instead of ω . However, the same rules used in the speed control are employed for the position control.

The used fuzzy logic controller has the same features (fuzzy inference rules, fuzzy sets, conjunction, implication, aggregation, and defuzzification operations) as that one used for the speed control.

4.3- PID Controller Synthesis [23]

In order to assess the quality of the fuzzy control based used here, a comparison with PID controllers is demonstrated. A PI controller is tuned and implemented for speed control, whereas a PD controller is used for position control. As inaccurate model is considered for the plant with the fuzzy controller, it is justifiable not to provide an accurate model for the plant to be used to tune the PID controller based on optimal methods, therefore heuristic ways are used to tune the gains K_p , K_i and K_d of the PI and PD controllers. A comparison between the performance with the fuzzy logic controller and the PID one is shown in Sec 6.

5- CONTROLLER IMPLEMENTATION

5.1- Hardware Aspects

The experimental setup used here is the analogue servo trainer CA06 manufactured by LJcreate Instrumentation Ltd [24], see Fig. 10 for the hardware setup, its block diagram is shown in Fig. 11.



Fig. 10: Hardware Setup

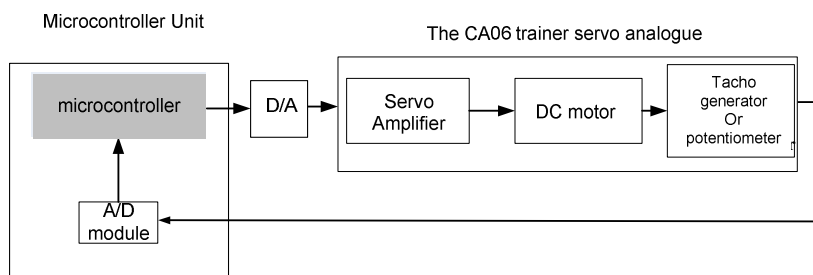


Fig. 11: Block Diagram of the Hardware Setup

The experimental setup consists of the D-C motor, a D-C tacho-generator to measure the angular velocity and a potentiometer fixed on the motor shaft to measure the angular position. The speed of the shaft is reduced by the factor 1/9 via a belt reduction. A magnetic brake is used as a mechanical load unit on the shaft of the motor.

The microcontroller PIC16F877A is used to implement digital controllers. The microcontroller maps the error signal into a control signal. The microcontroller unit, see Fig.11, includes a low cost 8-bit microcontroller with full 8-bit data paths, three independent timers, a 10-bit multi-channel Analog-to-Digital converter module (A/D), selectable oscillator options, FLASH program memory up to 8K x 14 words, Data memory (RAM) up to 368 x 8 bytes, EEPROM data memory up to 256 x 8 bytes and In-Circuit Serial Programming (ICSPTM). The microcontroller runs at an internal clock speed of 5 MHz with 200 ns instruction cycle [25].

The microcontroller receives the measured angular speed and position from the tacho-generator and the potentiometer, respectively, via the built in A/D converter. An external D/A converter has been built to convert the digital output from the microcontroller into analogue voltage. The microcontroller deals only with non negative values of the voltage up to 5 volts at its input and output channels, handling negative values are discussed in the next section.

5.2- Software Aspects

Note that the microcontroller reads zero for a negative input and produces zero voltage for a negative output value. A simple way to enable the microcontroller to read negative values is to receive its input as a subtraction between the measured input and

its negative value as shown in Fig. 12. Similar idea can be used to deal with negative output of the microcontroller, see Fig. 13.

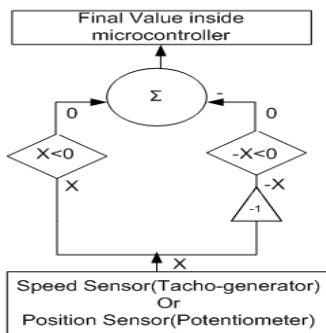


Fig. 12: Dealing with negative inputs into the microcontroller

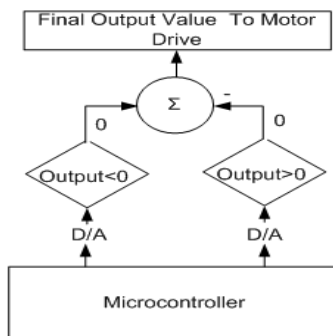


Fig. 13: Dealing with negative outputs of the microcontroller

In digital control implementation (with microcontrollers), the control algorithm is performed every sampling instant. In most microcontrollers it is commonly to use what is called timer interrupt routine to insure that the instant at which the computation of the algorithm starts in a certain sampling interval is the same in all sampling intervals. The timer interrupt approach provides accurate control of the sampling time. Note that the interrupts are generated by the internal timer of the microcontroller. The algorithm, which is written in C++, consists of two parts: the main program and an interrupt service routine as shown in Fig. 14a, Fig. 14b, respectively. The main program sets up the I/O ports (i.e. specify input/output pins, analog/digital pins, etc).

The timer interrupt service routine contains the controller algorithm; the timer is programmed to generate interrupts at regular intervals, equal to the sampling time.

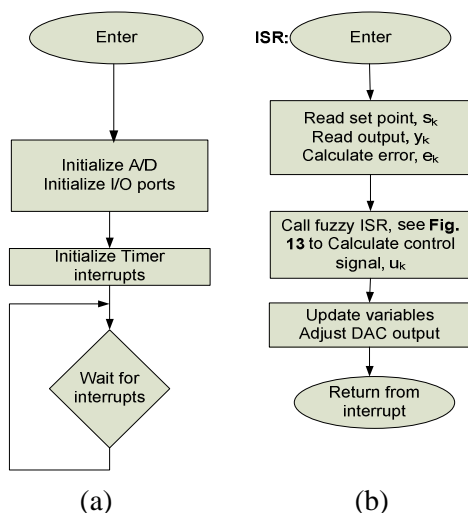


Fig. 14: Controller implementation: (a) main program; (b) interrupt service routine

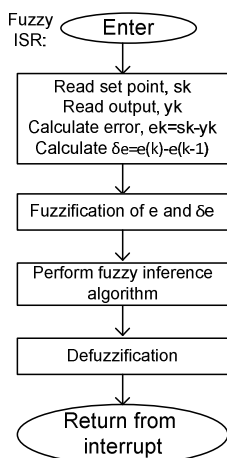


Fig. 15: Flow chart of fuzzy logic controller ISR

6- EXPERIMENTAL RESULTS

The synthesized controllers here are tested for tracking a step reference. To ensure a robust performance, disturbance rejection is considered as well. For speed control the reference speed is 13.966 rad/sec (133.33 RPM) and for position control the reference angle is 132 degree.

6.1 Speed Control

The tracking performance of both the PI and FLC controllers to a step reference is shown in Fig. 16-a. Table 3 summarizes the tracking performance of the two controllers. The FLC shows better performance in terms of transient response and

steady state error with no overshoot. It also gives reasonable control signal, see Fig. 17a. From Fig. 16b it can be seen that the fuzzy controller is able to reject a step disturbance applied to the motor during the interval between 4.5 and 9.5 sec. It is shown that the FLC out performs the PI controller.

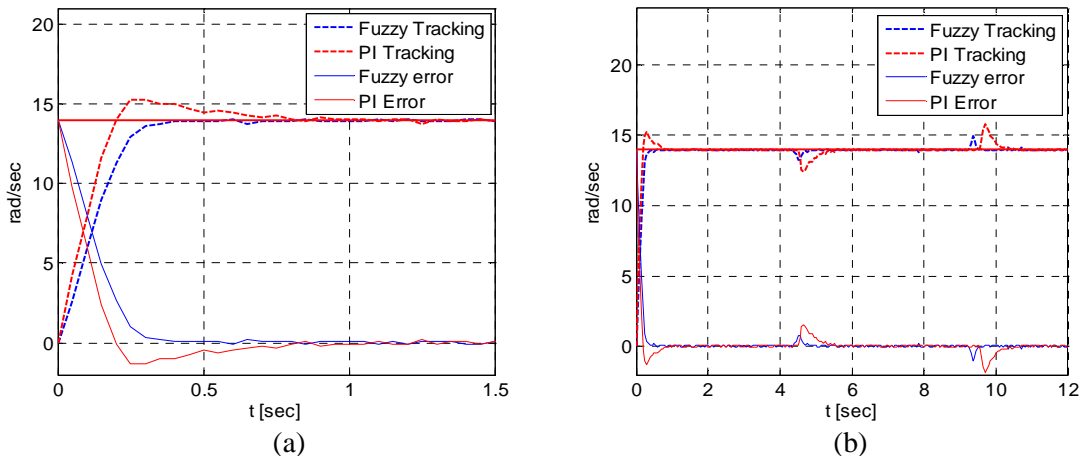


Fig. 16: (a) Motor speed for step reference
(b) Motor speed for step reference with load disturbance

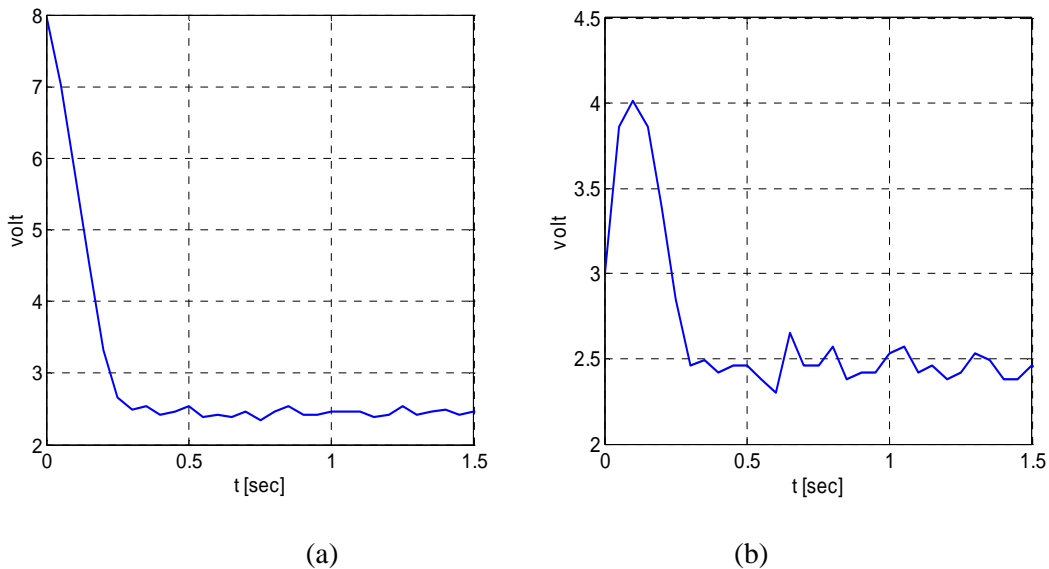


Fig. 17: (a) Controller voltage with PI controller
(b) Controller voltage with Fuzzy controller

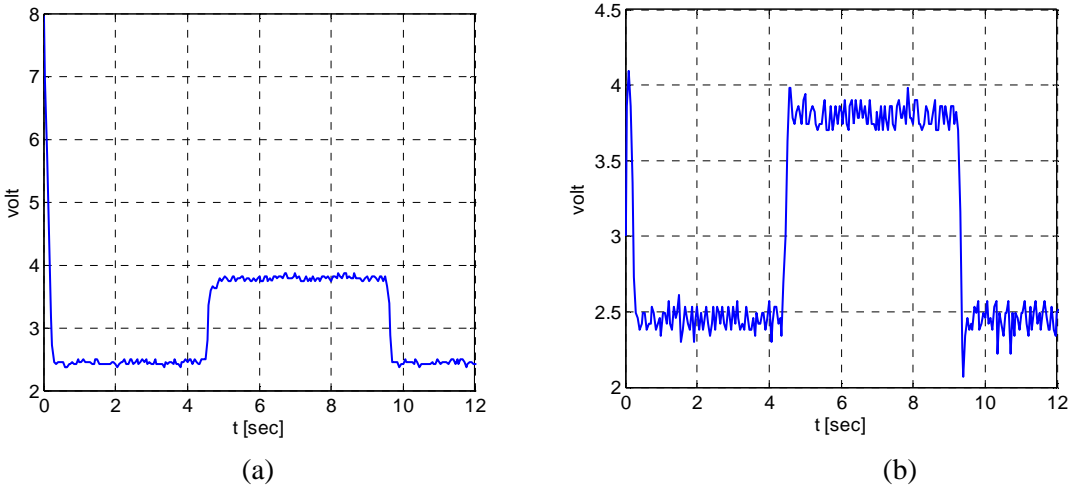


Fig. 18: (a) Controller voltage with PI controller with load disturbance
 (b) Controller voltage with Fuzzy controller with load disturbance

Table 3: Response specification of speed control

Algorithm	Rise Time Sec	Peak overshoot %	Peak Time Sec	Steady state error	Settling Time Sec	Max. control signal
PI	0.2	9.1937	0.3	0.0735	0.75	7.956
Fuzzy	0.3	0	-	0.07	0.4	4.017

6.2 Position Control

The tracking capability of the DC motor in closed loop with both a PD and the FLC controllers to follow a step reference input is shown in Fig. 19. Table 4 summarizes the tracking performance of the two controllers. The FLC shows better performance in terms of overshoot, rise and settling times. With less control effort, see Fig. 20a, b, in comparison with the PD one. In addition, it yields less steady state error than the PD controller.

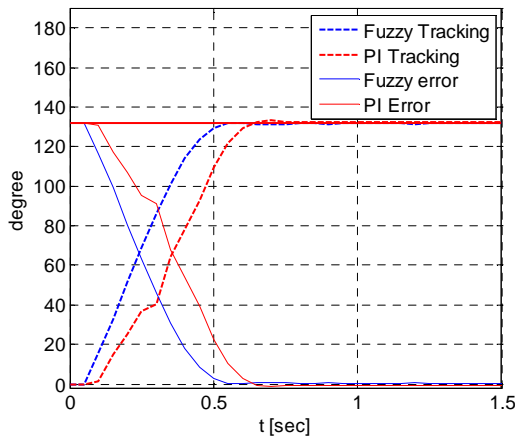


Fig. 19: Motor position for step input

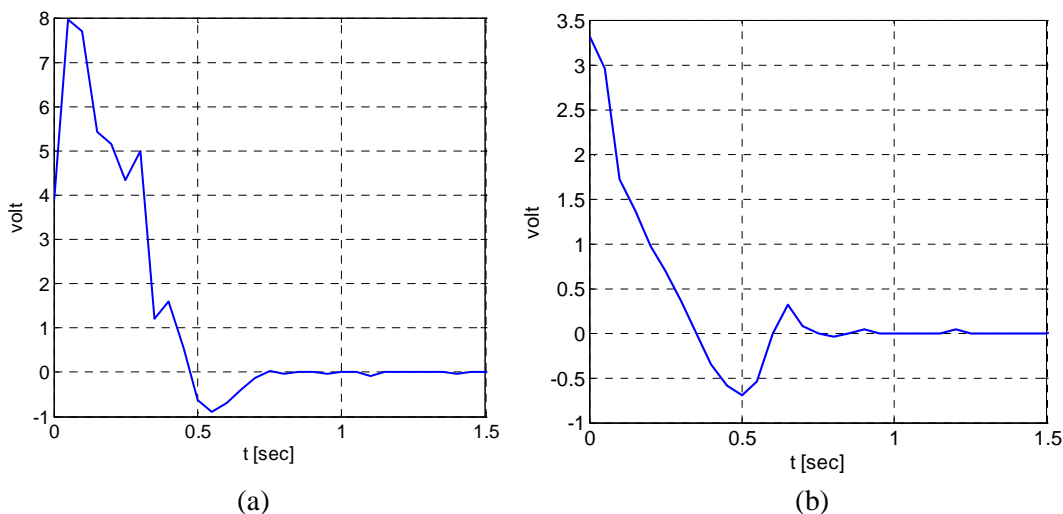


Fig. 20: (a) Controller voltage with PD controller
(b) Controller voltage with Fuzzy controller

Table 4: Response specification of position control

Algorithm	Rise Time Sec	Peak overshoot %	Peak Time Sec	Steady state error	Settling Time Sec	Max. control signal
PD	0.6	0.9	0.7	0.5133	0.75	7.956
Fuzzy	.45	0	-	0.1114	0.5	3.315

6.3 Practical Issues

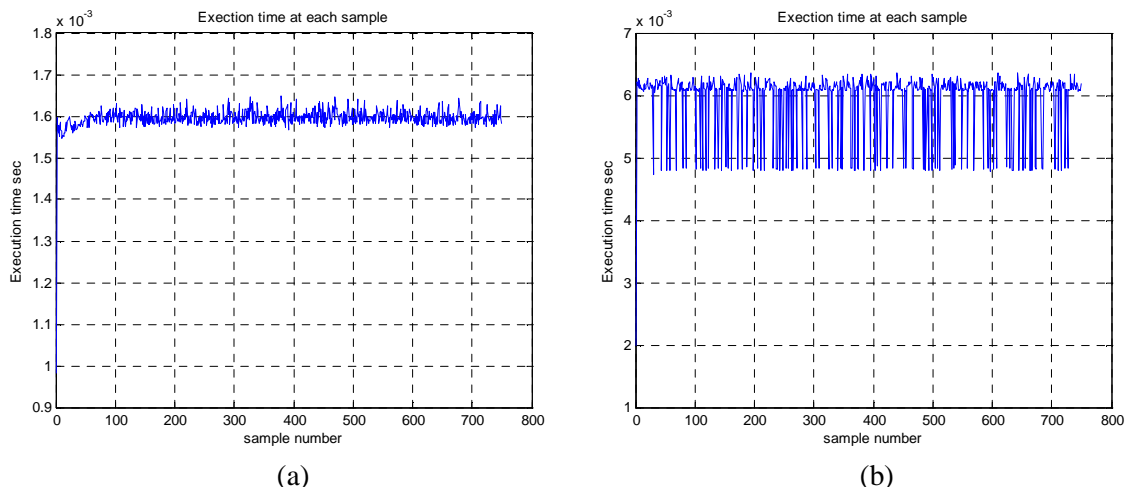
In this section we demonstrate the requirement specifications of the microcontroller unit when a FLC is implemented. This is carried out in terms of execution time of the control algorithm and the required memory.

Table 5 shows, the Ram, Program memory and execution time required to implement the PD/PI and the FLC algorithms in each sampling interval. Furthermore, **Fig. 21a, b** shows the change of the execution time to implement the PD/PI and the FLC algorithms, respectively in each sampling period.

The PD/PI controller demonstrates less microcontroller specifications, however the minimum specifications to implement the FLC controller is not so far from that of the PD/PI one, taken in consideration the advantages discussed above to use the FLC controller.

Table 5: Microcontroller Resources usage

Algorithm	Ram Byte	Program memory Word	Execution time msec
PID	62.56	1.2K	1.6
Fuzzy	69.92	3.36K	6.2



(a) Execution time of PI and PD controller algorithm
 (b) Execution time of FLC algorithm

7- CONCLUSIONS

This paper has demonstrated the implementation of a fuzzy logic controller for a DC motor speed and position control by using a low cost 8-bit microcontroller. In this paper the implementation with a microcontroller gives better results in terms of transient response, steady state error, and disturbance rejection. The controller showed good velocity and position tracking performance with load and no load conditions. The control algorithm is implemented with C language. Also, microcontroller can be used for implementing more complex control algorithms with the ability of easily reprogramming. In future work the microcontroller will be used to implement more efficient control techniques such as H_{∞} and linear matrix inequality (LMI).

REFERENCES

- (1) Nuha Nawash, "H-infinity control of an autonomous mobile robot" MSc IN Electrical engineering at the CLEVELAND STATE UNIVERSITY May, 2005
- (2) M. K. Gupta , A. K. Sharma , D. Patidar, "A robust variable structure position control of DC motor" Journal of Theoretical and Applied Information Technology 2005
- (3) J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: A survey," IEEE Trans. Ind. Electron., vol.40, no. 1. pp. 2-22, 1993.
- (4) H. Butler, G. Honderd, and J. V. Amerongen, "Model reference adaptive control of a direct-drive DC motor," IEEE Mag. Contr. Sys., vol.9, no. 1, pp. 80-84, 1989.
- (5) Lennart Ljung, "System Identification: theory for the user", Prentice-Hall. Inc, 1987.
- (6) L.A. Zadeh, "Fuzzy Sets," Information and Control, 8, 338-352, 1965
- (7) Da Ruan, "Fuzzy Logic in the Nuclear Research World", Fuzzy Sets and Systems Volume 74, Issue 1, Pages 5-13, 25 August 1995.

- (8) Mekhamer, S.F.; Soliman, S.A.; Moustafa, M.A.; El-Hawary, M.E.;" Application of Fuzzy Logic for Reactive Power Compensation of Radial Distribution Feeders" IEEE TRANSACTIONS ON POWER SYSTEMS, VOL.18, NO.1, FEBRUARY 2003
- (9) C.Y. Ngo and Victor O.K. Li, "Freeway traffic control using fuzzy logic controllers", Information Sciences - Applications Volume 1, Issue 2, Pages 59-76, March 1994.
- (10) I. Kalaykov, B. Iliev, R. Tervo, "DSP-BASED fast fuzzy logic controllers"
- (11) Güngör Bal, Erdal Bekiro, evki Demirba and Ihami Çolak, "Fuzzy logic based DSP controlled servo position control for ultrasonic motor", Energy Conversion and Management Volume 45, Issue 20, December 2004
- (12) M D Hanamane, R R Mudholkar, B T Jadhar and S R Sawart, "Implementation of fuzzy temperature control using microprocessor ", Journal of Scientific & Industrial Research 2006
- (13) Sameep Singh and Kuldip S. Rattan, "Implementation of a fuzzy logic controller on A FPGA for a DC motor", IEEE 2003
- (14) Silvio Triebel, J. Kelber and Gerd Scarbata, "Semiautomated implementation of analogous fuzzy systems", Facta UNIVERSITIES, Electronics and Energetics 1998
- (15) D. Ibrahim, "Microcontroller based applied digital control", John Wiley & Sons Ltd, 2006.
- (16) Michael Barr, "Programming Embedded Systems in C and C++", O'Reilly & Associates, 1999.
- (17) Martin P. Bates, "Interfacing PIC Microcontrollers Embedded Design by interactive Simulation", Elsevier, 2006.
- (18) Ravinesh Singh, Godfrey C. Onwubolu, Krishnileshwar Singh and Ritnesh Ram, "DC Motor Control Predictive Models", American Journal of Applied Sciences 3 (11): 2096-2102, 2006
- (19) S. C. Won, D. J. Lim and D. H. Chyung, "D-C Motor driven robotic manipulator control", Proceedings of 24th Conference on Decision and Control Ft. Lauderdale, FL, December 1985
- (20) Moleykutty George, "Speed Control of Separately Excited DC Motor" American Journal of Applied Sciences 5 (3): 227-233, 2008
- (21) Asem M. A. "Intelligent Tracing Control of a D-C motor", M.sc. electrical Eng. Dep. Assuit University, 2002.
- (22) " Low cost 2-12 Watt ironless rotor servo motors ", Mclennan Servo Supplies Ltd.
- (23) Michael A. Johnson and Mohammad H.Moradi, "PID Control: New Identification and Design Methods," Springer-Verlag London Limited 2005.
- (24) " Analog and Digital Motor Control, Curriculum Manual CA06", LJcreate Ltd.
- (25) "PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers", Microchip Technology Incorporated, USA, 2001.

تنفيذ المتحكم المنطقي المبهم باستخدام متحكم دقيق 8 بت للتحكم في سرعة وزاوية طور موتور التيار المستمر

يقدم هذا البحث التصميم والتنفيذ العملي للمتحكم المنطقي المبهم (FLC) للتحكم في سرعة وزاوية طور موتور التيار المستمر، والدافع وراء استخدام المتحكم المنطقي المبهم (FLC) هو قدرته علي التعامل مع عدم الدقة في ثوابت النظام.

تم تنفيذ برنامج المتحكم المنطقي المبهم (FLC) بمتحكم دقيق 8 بت بدلا من استخدام المعالج الدقيق المتعدد الاغراض. ما سبق يؤدي الي تكلفة معقولة لمثل هذه التطبيقات، ومن ناحيه تتبع المرجع وازالة تأثير تغير الحمل اتضح من النتائج العملية الأداء العالي للمتحكم المنطقي المبهم (FLC) بالمقارنة مع المتحكم الخطي التكاملي (PI) والمتحكم الخطي التفاضلي (PD) المصممين لنفس الأغراض.