View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Ain Shams Engineering Journal (2014) 5, 399–409

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

ELECTRICAL ENGINEERING

Embedded system based on a real time fuzzy motor speed controller

Ebrahim Abd El-Hamid Mohamed Ramadan *, Mohammad El-Bardini [1], Nabila M. El-Rabaie [1], Mohamed A. Fkirin [1]

Department of Industrial Electronics and Control, Faculty of Electronic Engineering – Menof., Menofia University, 32952 Menof., Egypt

Received 11 May 2013; revised 26 September 2013; accepted 8 October 2013
Available online 15 November 2013

KEYWORDS

Fuzzy logic control;
PI controller;
VHDL;
FPGA;
Spartan-3E starter kit

Abstract This paper describes an implementation of a fuzzy logic control (FLC) system and a/the conventional proportional-integral (PI) controller for speed control of DC motor, based on field programmable gate array (FPGA) circuit. The proposed scheme is aimed to improve the tracking performance and to eliminate the load disturbance in the speed control of DC motors. The proposed fuzzy system has been applied to a permanent magnet DC motor, via a configuration of H-bridge. The fuzzy control algorithm is designed and verified with a nonlinear model, using the MATLAB® tools. Both FLC and conventional PI controller hardware are synthesized, functionally verified and implemented using Xilinx Integrated Software Environment (ISE) Version 11.1i. The real time implementation of these controllers is made on Spartan-3E FPGA starter kit (XC3S500E). The practical results showed that the proposed FLC scheme has better tracking performance than the conventional PI controller for the speed control of DC motors.

© 2013 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.

1. Introduction

Intelligent controllers as new technologies have recently been applied to electrical power control systems in general and motor control systems in particular [1]. Many digital techniques have been used to implement a digital controller. However, the conventional proportional–integral–derivative (PID) controller is still being a key component in industrial control systems, because it is simple and provides useful solutions to many industrial processes [2].

Fuzzy logic control (FLC) provides an alternative to the PID controller since it is a good tool for the control of nonlinear systems that are difficult in modeling [3–5]. PID controller can be transformed into a PID-like structure of fuzzy controllers.

Several fuzzy control applications including the physical systems, require a real time operation to interface high speed constraints [6,7]. Higher density programmable logic devices, such as field programmable gate array (FPGA) can be used

* Corresponding author.
E-mail addresses: Eng_ebrahimramadan@yahoo.com (Ebrahim Abd El-Hamid Mohamed Ramadan), Drelbardini@ieee.org (M. El-Bardini), Nabila2100@gmail.com (N.M. El-Rabaie), mafkirin@yahoo.com (M.A. Fkirin).

[1] Fax: +20 483660716 (work).

Peer review under responsibility of Ain Shams University.

ELSEVIER Production and hosting by Elsevier

2090-4479 © 2013 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.
http://dx.doi.org/10.1016/j.asej.2013.10.001

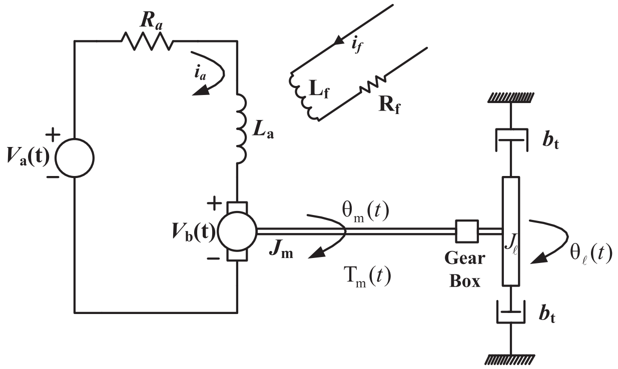

Figure 1 DC motor equivalence circuit.

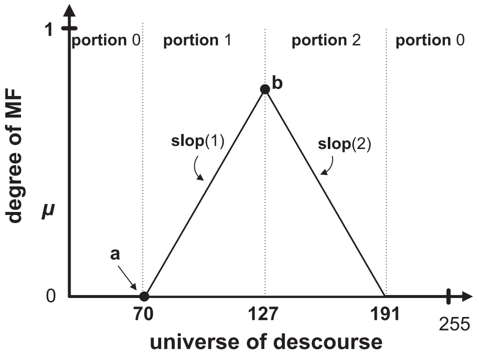

Figure 4 Triangular type membership function.

to integrate large amounts of fuzzy logic in a single integrated circuit (IC) [8,9]. FPGAs are one of the fastest growing parts of the digital integrated circuit market in recent times. Reconfigurable FPGA systems provide additional flexibility and can adapt to various computational tasks through hardware reuse. Therefore, FPGA becomes one of the most successful technologies for developing the systems which require a real time operation [10–12].

Many researchers discussed the design of the hardware fuzzy logic controller. Numbers of these works were specialized in control application [13–15] and were aimed to get better control responses [16]. The proposed design in this paper aims

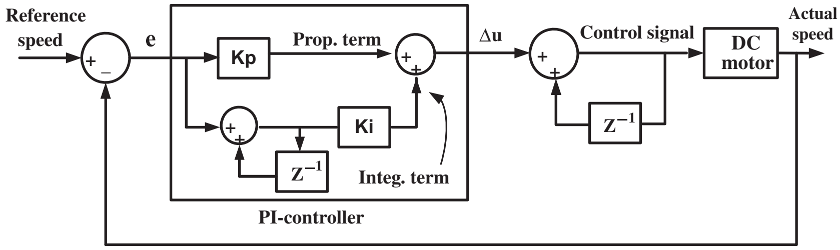

Figure 2 Overall block diagram of PI based control system for DC motor.

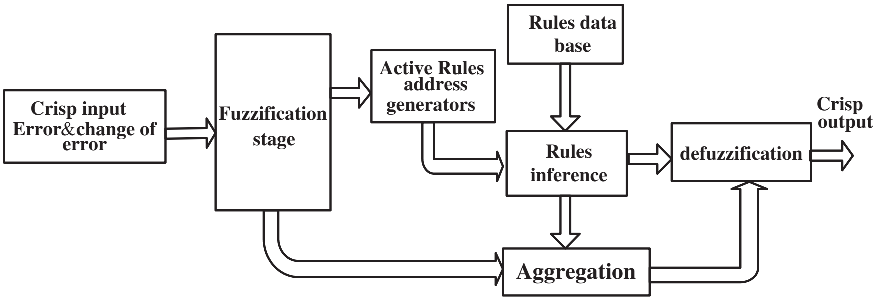

Figure 3 FLC architecture.

Table 1 The proposed fuzzy characteristics.

Fuzzy inference system	Mamdani FIS
Inputs	2
Input resolution	8-bit
Outputs	1
Output resolution	8-bit
Antecedent MF's	7 triangular per fuzzy se
Antecedent MF degree of truth resolution	8-bit
Consequent MF's	7 triangular per fuzzy set
Consequent MF degree of truth resolution	8-bit
Aggregation method	MAX
Implication method	MIN
Defuzzification method	Center of Gravity (COG)

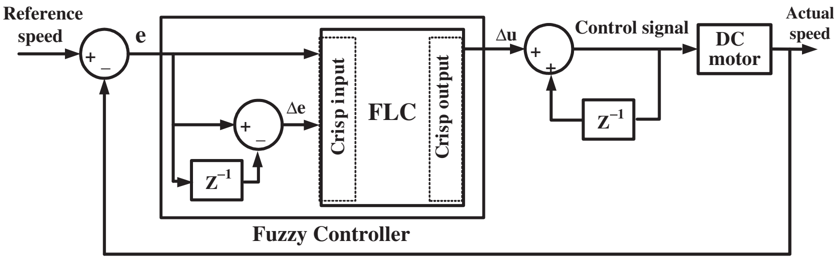

Figure 5 Overall block diagram of FLC based control system for DC motor.

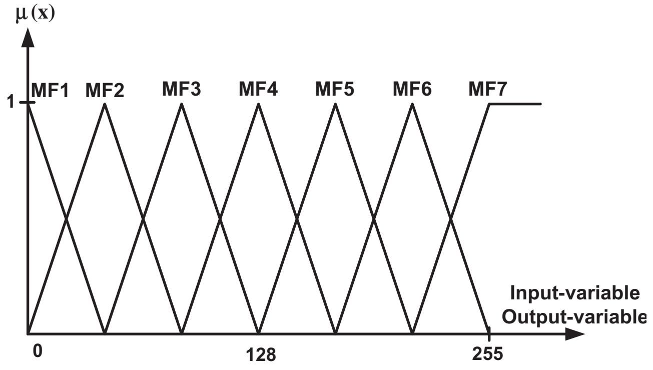

Figure 6 Input and output membership functions.

to employ the new techniques of PI-fuzzy algorithm in estimating and controlling the speed of a separately excited DC motor and compare it to the conventional PI control to serve these applications efficiently. The two controllers are coded in VHDL. Xilinx ISE which is used for synthesis and implementation of controllers on the Spartan-3E FPGA Kit.

2. Modeling of DC motor

A schematic representation of an armature controlled DC motor is given in Fig. 1. According to this schematic representation, the equations based on the Kirchoff's voltage law combined with the Newton's moment law can write as follows [17,18];

Table 3 The synthesis report from ISE WebPack program.

Device utilization summary (estimated values)			
Logic utilization	Used	Available	Utilization (%)
Number of slices	2193	4656	47
Number of slice Flip flops	1900	9312	20
Number of 4 input LUTs	4048	9312	43
Number of bonded IOBs	282	232	121
Number of MULT 18X18SIOs	16	20	80
Number of GCLKs	2	24	8

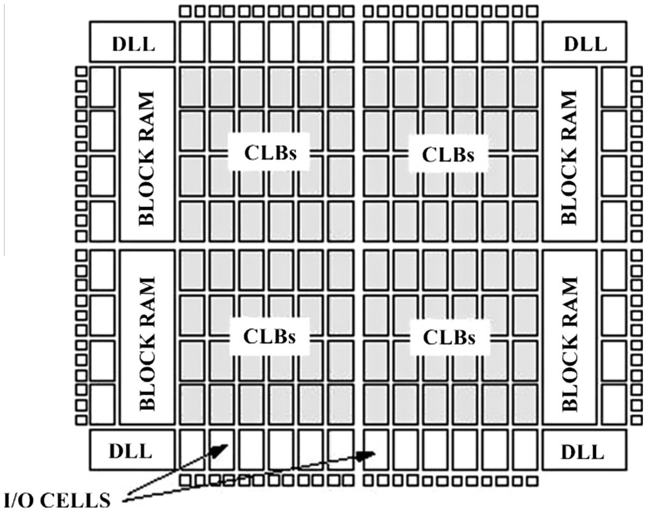

Figure 7 Spartan 3E layout.

Table 2 Fuzzy controller operation.

	e	MF1	MF2	MF3	MF4	MF5	MF6	MF7
Δ*e*	Δ*u*	001	010	011	100	101	110	111
MF1		MF1	MF1	MF1	MF1	MF2	MF3	MF4
001		001	001	001	001	010	011	100
MF2		MF1	MF1	MF1	MF2	MF3	MF4	MF5
010		001	001	001	010	011	100	101
MF3		MF1	MF1	MF2	MF3	MF4	MF5	MF6
011		001	001	010	011	100	101	110
MF4		MF1	MF2	MF3	MF4	MF5	MF6	MF7
100		001	010	011	100	101	110	111
MF5		MF2	MF3	MF4	MF5	MF6	MF7	MF7
101		010	011	100	101	110	111	111
MF6		MF3	MF4	MF5	MF6	MF7	MF7	MF7
110		011	100	101	110	111	111	111
MF7		MF4	MF5	MF6	MF7	MF7	MF7	MF7
111		100	101	110	111	111	111	111

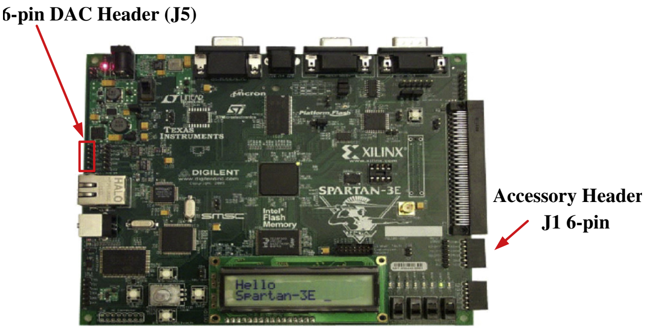

Figure 8 Spartan 3E starter kit.

$$V_a = L_a\frac{di_a}{dt} + R_a i_a + V_b \quad (1)$$

$$T_m = \left(J_m + \frac{1}{K_g^2}J_\ell\right)\frac{d\omega_m}{dt} + \frac{1}{K_g^2}b_t\omega_m \quad (2)$$

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

$$V_b = K_b\omega_m \quad (4)$$

In addition, the transfer function from input V_a to output ω_ℓ is given by the following:

$$\frac{\omega_\ell(s)}{V_a(s)} = \frac{K_gK_T}{L_aJ_{\text{eq}}S^2 + (L_ab_t + R_aJ_{\text{eq}})S + R_ab_t + K_g^2K_TK_b} \quad (5)$$

where $J_{\text{eq}} = (K_g^2J_m + J_\ell)$ *and* $k_g = \frac{\omega_m}{\omega_\ell}$

T_m motor torque,
K_T motor torque constant,
K_b motor constant,
ω_m armature angular velocity,
J_m motor inertia,
J_ℓ load inertia,
ω_ℓ shaft angular velocity,
θ_m armature angular position,
$V_a(t)$ armature voltage,
$i_a(t)$ armature current,
V_b back e.m.f,
R_a armature resistance,
L_a armature inductance,
K_g gear-box ratio,
b_t rotational viscous friction,
θ_ℓ shaft angular position.

3. Modeling of PI controller using VHDL

The standard representation of a PI controller is given by Eq. (6); where, there are two main components that can be identified; they are the proportional and integral terms as shown in Fig. 2. The error signal $e(t)$ is defined by Eq. (7), and it is the difference between the process output and the desired speed. The integral term was obtained using the method of rectangular integration. In this equation, K_p and K_i, are parameters related to the gain of each term.

$$u_{\text{PI}}(t) = k_pe(t) + k_i\int_0^t e(t)dt \quad (6)$$

$$e(t) = \text{Reference speed} - \text{Actual speed} \quad (7)$$

The algorithm given by Eq. (6) is seldom used in practice. In this work, for speed control of a DC motor, a cascaded control structure is usually preferred. This controller is known as parallelism controller which is being used [19]. By differentiating Eqs. (6) and (8) can be obtained, where U_{PI} is the control signal.

$$\frac{du_{\text{PI}}(t)}{dt} = k_p\frac{de(t)}{dt} + k_ie(t) \quad (8)$$

In discrete-time systems, Eq. (8) can be written as follows:

$$u_{\text{PI}}(kT) - u_{\text{PI}}(kT - T) = k_p(e(kT) - e(kT - T)) + k_ie(kT) \quad (9)$$

In simulation, the parameters V_a, ω_ℓ, R_a, L_a, K_g, J_{eq}, b_t, K_b and K_T in Fig. 1 are set as 24 vdc, 300 rpm, 4.67 Ω, 170 mH, 0.16, 42.6e−6 kg m^2, 47.3e−6 N m s/rad, 14.7e−3 V s/rad and 14.7e−6 N m/A, respectively [20]. From the open loop step response of Permanent Magnet DC motor, the values of the controller parameters are tuned using Ziegler-Nichole method [4]. The values found are K_p = 0.19 and K_i = 0.0301.

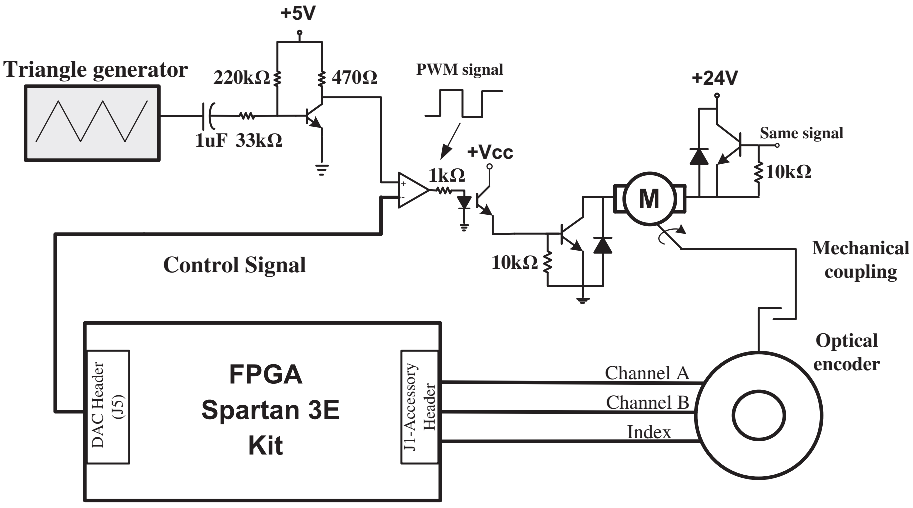

Figure 9 Block diagram of the DC motor speed control using FPGA.

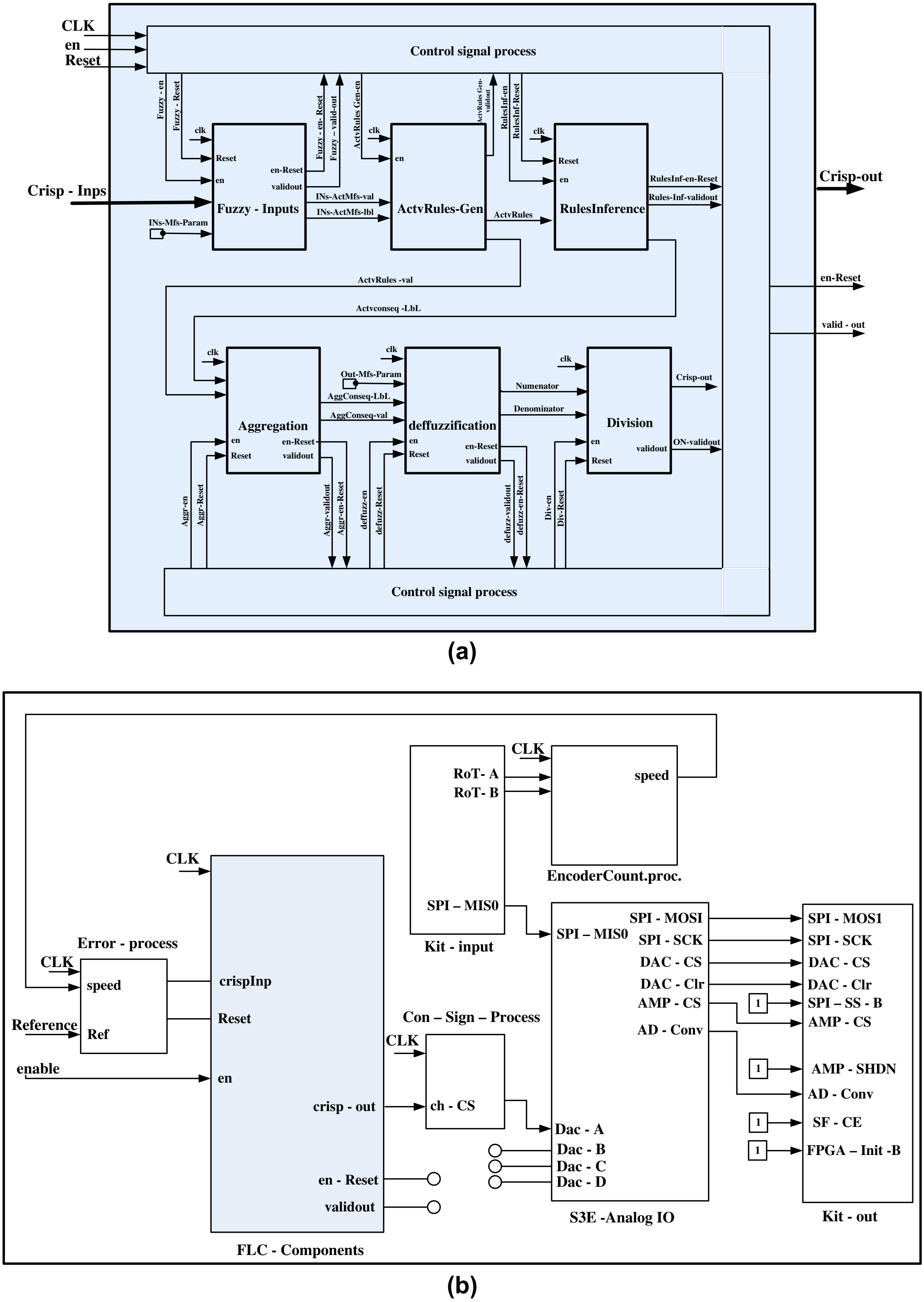

Figure 10 Internal architecture of the FPGA-based speed control for DC motor (a) Internal FLC-components architecture and (b) internal component architecture.

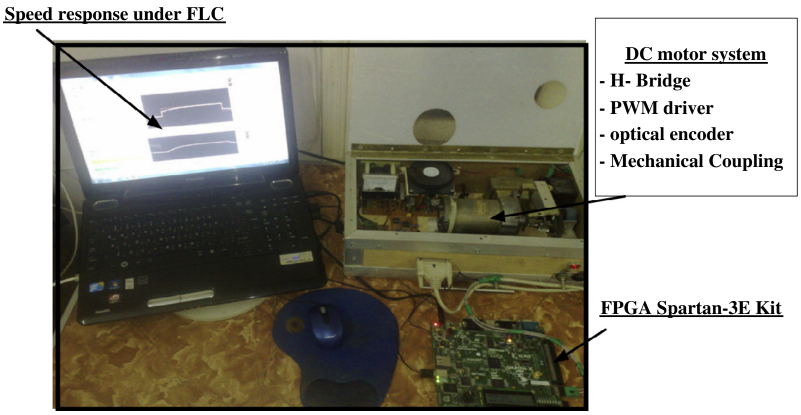

Figure 11 Real system of the DC motor speed control using FPGA.

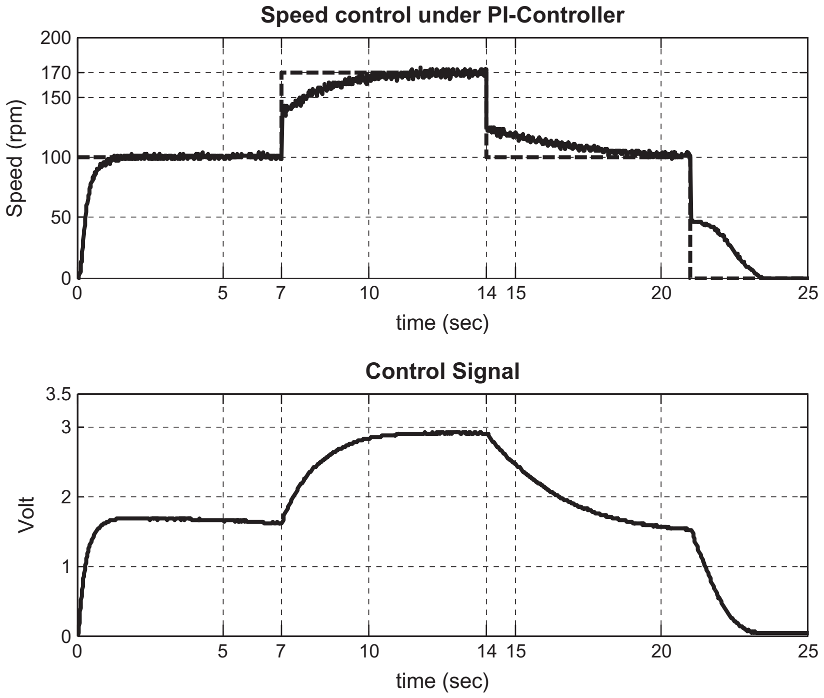

Figure 12 System response of PI controller for changes in the operating point.

4. Fuzzy logic controller structure

A fuzzy controller structure for control consists respectively of fuzzification, rule data base, rules inference and defuzzification [21], as shown in Fig. 3, which has the characteristic parameters as shown in Table 1.

- *Fuzzification stage*: The crisp input of *Error* (e) and *Change of Error* (Δe) are defined by:

$$e(kT) = \omega_{\text{ref}} - \omega_{\ell} \tag{10}$$

$$\Delta e(kT) = e(kT) - e(kT - T) \tag{11}$$

where ω_{ref} is the reference and ω_{ℓ} is the system output. The fuzzification process changes input signal value (e, Δe) from the fields of real numbers to the value of membership function $\mu(x)$ of the fuzzy sets. In the proposed fuzzy system, the fuzzification process is performed by reading out from the system's memory the values of membership function of activated sets and also codes of these sets [22].

Symmetric triangular MFs are used; each membership function can be completely described by only two points (a, b) and two slope values [23]. The entire membership function can be divided into three portions: 0, 1 and 2, as shown in Fig. 4. It shows how triangular input membership functions are formed in the fuzzification stage.

$$\text{The degree of membership}(\mu) = \begin{cases} (1)\ \text{Portion0}, & \mu = 0, \\ (2)\ \text{Portion1}, & \mu = (\text{Input value} - a) * \text{slope1} \\ (3)\ \text{Portion2}, & \mu = 1 - (\text{Input value} - b) * \text{slope2} \end{cases}$$

The two input variables are coded as 8-bit hexadecimal value (00 – FF); each input of crisp input is entered at the same time for the membership function to calculate the μ value. As an example, let us use the input value of 150 to calculate the

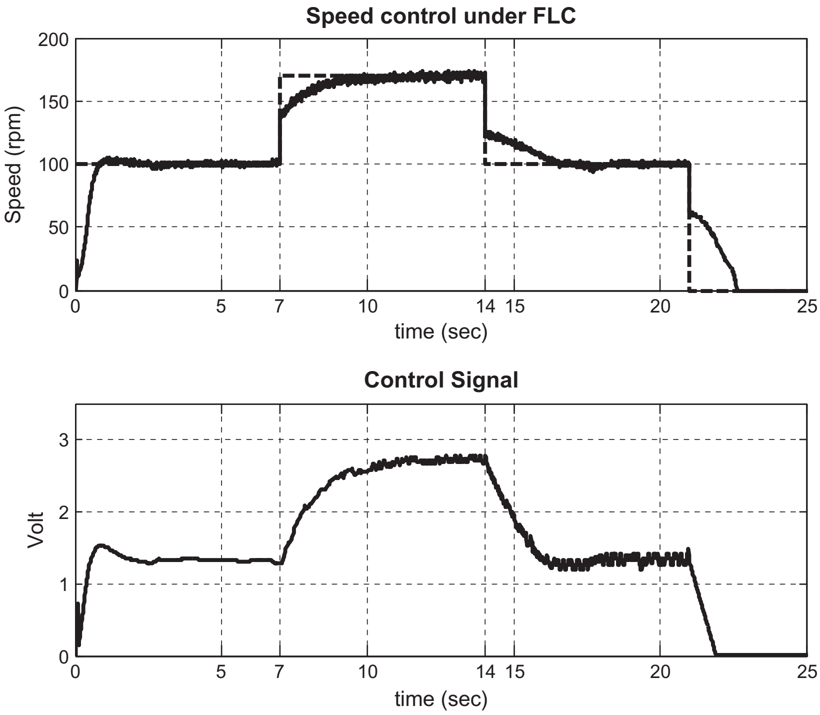

Figure 13 System response of PI-FLC for changes in the operating point.

Table 4 Comparison between two responses.

	PI controller	PI-FLC
Rise time (s)	0.8050	0.7368
Settling time (s)	6.5390	6.886
Over shoot (rpm)	3.6364	1.75
Peak time (s)	4.2800	1.08
RMSE without load	15.3512	11.5399
RMSE with load		
Under −10% load	14.6333	14.3861
Under −20% load	16.8462	15.6657
Under −25% load	20.5089	16.7217

degree of membership function. Using an 8-bit computerized resolution, $\mu = 1$ equals to \$FF or 255 in decimal (the "\$" sign indicates the hexadecimal number representation). The values of (a) and (b) are \$46 and \$7F; respectively, and the two slopes can be calculated as follows:

Slope $(1) = 1/(127 - 70) = \mathrm{FF}/39 = 255/57 = 4 = 04$. And

Slope $(2) = 1/(191 - 127) = \mathrm{FF}/40 = 255/64 = 3 = 03$.

Since the input value of 150 (\$96) is greater than (b) and lies in portion 2, therefore:

$$\mu = \mathrm{FF0} - (\text{Input value} - b) * \text{slope}(2)$$
$$= \mathrm{FF} - (96 - 7F) * 03 = \mathrm{BA}.$$

- *Active rules generator*: each input variable generates the address of memory, which indicates the appropriate discrete sample of the point of the rule activation stored in the memory. Therefore, the output of the fuzzification stage consists of a couple of 3-bit codes (Active MF Labels).

- *Rule evaluation and inference stages*: the input variables (e, Δe) are processed by an inference engine that executes a set of control rules contained in rule bases. The control rules are formulated using the knowledge of the DC motor behavior [24]. Each rule is expressed in the form:

$$\text{IF } e \text{ is } A_i \quad \text{and} \quad \Delta e \text{ is } B_i \text{ THEN } U_f \text{ is } C_i \tag{12}$$

where $i = 1, 2, 3, m$. There are seven fuzzy sets for each linguist value $\{A_i, B_i, C_i\}$ and 49 fuzzy control rules (rule bases) are designed for the two inputs and one output fuzzy system. Different inference algorithms can be used to produce the fuzzy set values for the output fuzzy variable U_f. In this paper, the max–min inference algorithm is used, in which the membership degree is equal to the maximum of the product of (e, Δe) membership degree [3].

- The truth values W_i of the premises are calculated by:

$$W_i = \min\lfloor \mu_{A_i}(e),\ \mu_{B_i}(\Delta e) \rfloor \tag{13}$$

- The membership function $\mu_c(u)$ of the inferred consequence is given by:

$$\mu_c(u) = \max[\min(W_1,\ \mu_{c1}),\ \min(W_2,\ \mu_{c2})] \tag{14}$$

In the proposed fuzzy system, once the degree of membership of the two inputs is tested, the Active MF Labels are used to generate the addresses which needed to select just the fuzzy active rules. The conclusion codes are stored at precisely determined address, which values correspond to the premises of activated rules. During the reading of indicated codes of rule conclusion, the fuzzy AND is computed applying the (MIN) operator to the two 8-bit membership values [25], which relies on the choice of the smallest value

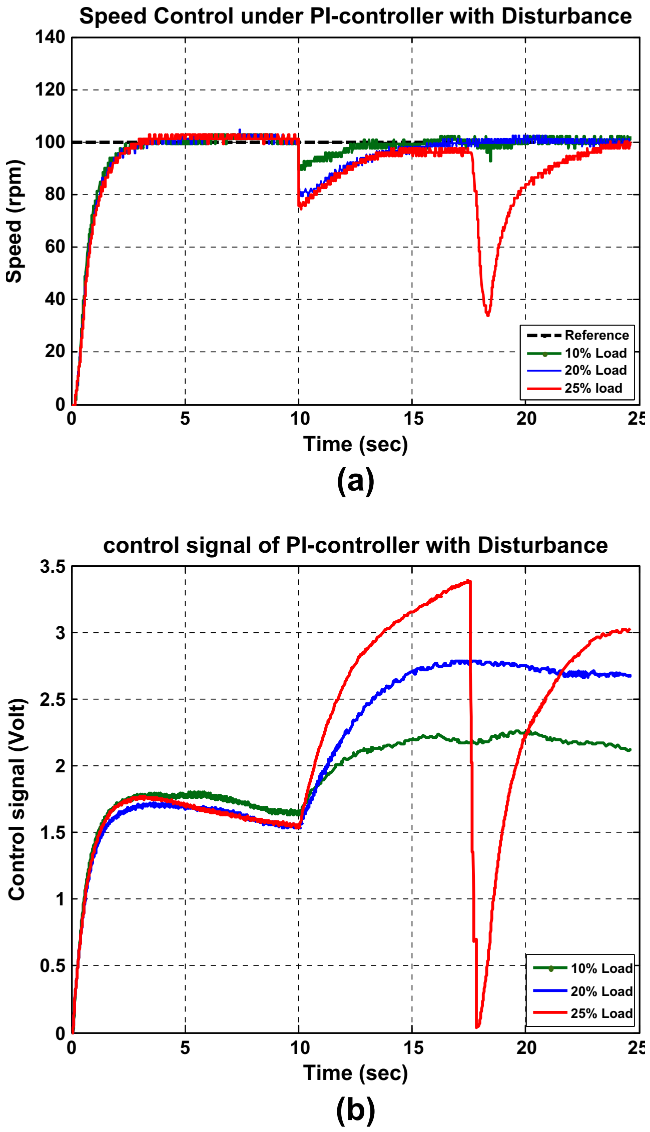

Figure 14 Regulatory responses of PI controller under −10%, −20% and −25% load changes speed control under PI controller with difference loads −10%, −20% and −25%. (b) Control signal of the PI controller.

of the membership values. After this, the operation of aggregation (MAX) begins [16]. Finally, the inference stage decides what the fuzzy control output by using fuzzy reasoning is performed by inference engine strategy. Since rule outputs are expressed by fuzzy numerical values, the numerical results from this calculation are called fuzzy outputs.

- *Defuzzification stage*: Interface converts the fuzzy control action into the inputs to the plant (crisp output). The crisp output (8-bit) is computed using the Center Of Gravity (COG) [26].

$$\text{crisp output} = \frac{\sum_{i=1}^{n}\mu(u_i)\cdot u_i}{\sum_{i=1}^{n}\mu(u_i)} \tag{15}$$

where $\mu(u_i)$ is the values of membership function for output and u_i is the values of locations of output MFs. Next, this result is transferred to DC motors.

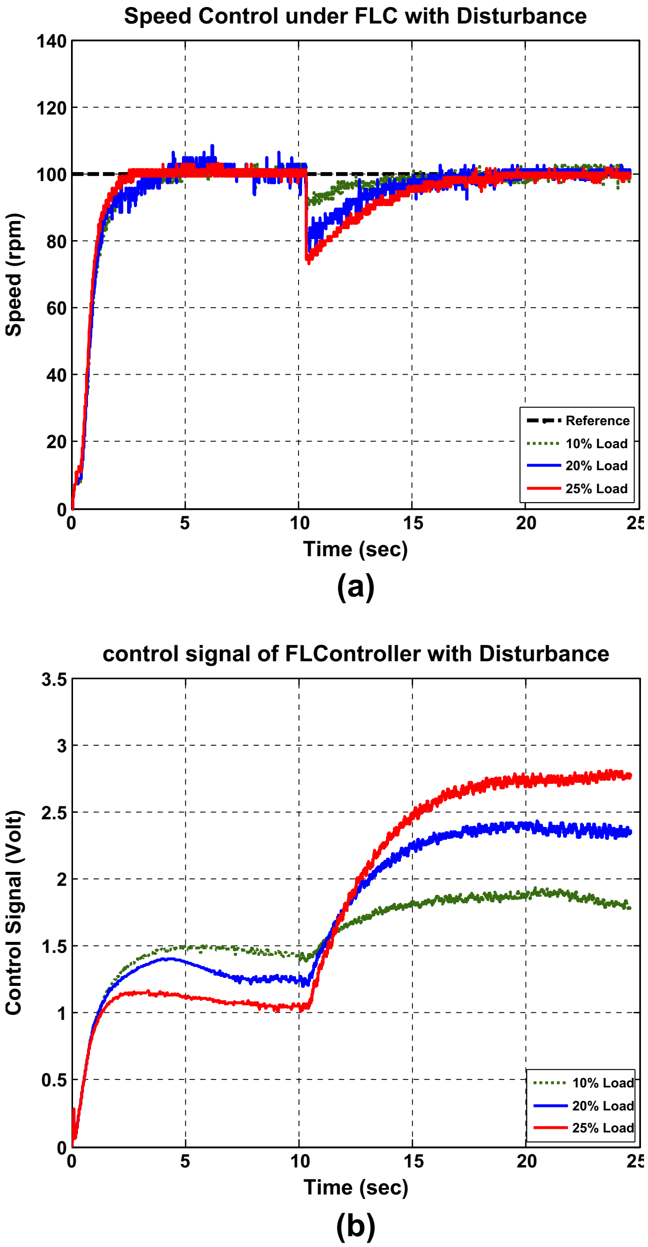

Figure 15 Regulatory responses of PI – FLC under −10%, −20% and −25% load changes (a) speed control under PI-FLC with difference loads −10%, −20% and −25% and (b) control signal of the PI-FLC.

5. Modeling of fuzzy logic controller using VHDL

There are a number of reasons for using fuzzy logic in the speed control of DC motor. The main advantage is the flexibility offered by fuzzy logic [11,27]. Fig. 5 shows a block diagram demonstrating the implementation of the FLC in a speed control of DC motor. In this application, the input interface converts the output of the speed encoder into error and change of error which are used as the two inputs to the FLC.

The fuzzy PI controller was implemented according to Fig. 3. There are seven triangle memberships representing the *Error* (e) and *Change of Error* (Δe) inputs. The normalizing of these membership functions is shown in Fig. 6, where, real range of *Error* (e) and *Change of Error* (Δe) are converted to 8-bits digital scale vectors e_{scale} and Δe_{scale} from 0 to 255 which represent the inputs of FLC. The mathematical equations for this process are as follows:

$$e_{\text{scale}} = 1.25 \times e - 127.5 \tag{16}$$

$$\Delta e_{\text{scale}} = 25.5 \times \Delta e - 127.5 \tag{17}$$

The two scale inputs of *Error* (e) and *Change of Error* (Δe) can have negative physical values. Since the universe of discourse can handle 8-bit vector, the digital representations of (e_{scale}) and (Δe_{scale}) are biased by 128 bits as shown:

$$(e_{\text{scale}} \text{ or } \Delta e_{\text{scale}}) = \begin{cases} > 0 & \text{Then} \quad 128 < \text{value} \leqslant 255; \\ = 0 & \text{Then} \quad \text{value} = 128; \\ < 0 & \text{Then} \quad 0 \leqslant \text{value} < 128; \end{cases}$$

The basis of choice for the universe of discourse range (0–255) is to reduce the computational complexity of the different sets of FLC. Since the memory space in FPGA Spartan®-3E starter kit is limited, and the smallest number of bits can be used for representing the real error and change of error is 8-bit. Hence, when the universe of discourse is increased above 8-bit, the computation size will be increased. The FLC works in a way that relates the controlled system output to the reference input by a set of fuzzy rules. These rules are the most important part of the FLC and they must be obtained correctly so that the relation between input and output of the system is represented properly [28]. Fuzzy rule's operation is based on the control expression of Table 2.

The inputs and outputs, with the gate level details of the subsystem and the synthesis report that is created from ISE WebPack program, are shown in Table 3.

6. Controller design for FPGA implementation

The Spartan-3E FPGA is embedded with the 90 nm technology at the heart of its architecture. This reduces the die size and cost, increases manufacturing efficiency, and addresses a wider range of applications. The Spartan-3E diagram, shown in Fig. 7, allows users to easily migrate to different densities across multiple packages and supports 18 different single-ended and differential I/O standards [29].

The speed of the motor is measured by means of optical encoder and is given as an input to the expansion connectors on board (J1 6-pin Accessory Header) [30]. As shown in Fig. 8.

The reference speed is assigned to motor internal configuration memory of the FPGA. To verify the performance of the controller design on the hardware, the VHDL code (Bit file) is downloaded into the target FPGA device (Spartan-3E family XC3S500) using Xilinx ISE pack 11.1i. The complete experimental system consists of a DC motor, FPGA kit, and the circuits of (PWM driver and H-bridge). The overall block diagram of DC motor speed control is shown in Fig. 9, and the internal architecture of the proposed FPGA-based speed control is shown in Fig. 10. Fig. 11 shows the real system.

Results of measurements confirmed again correct operation of the system. The time of system processing was sample time = 0.05 s based on clock frequency 50 MHz.

7. Results and discussion

The proposed controllers are implemented in FPGA for real time control of speed in the DC motor. The servo responses of the PI controller and PI-FLC are shown in Figs. 12 and 13; respectively. The reference speed is varied from 100 rpm to 170 rpm. For the PI controller (shown in Fig. 12), reference speed tracking performance is characterized by lack of smooth transition between desired speed, as well as the presence of overshoot and higher rise time. From Fig. 13, it can be observed that the PI-FLC has less oscillation, zero overshoot and less rise time. From Figs. 12 and 13 and Table 4, it is clear that PI-FLC performs significantly better than PI controller.

The PI-FLC is used to control the speed of the motor while applying a different loads changing as −10%, −20% and −25% [11]. The motor is also run with a PI controller while applying the same load changes. The variations in speed with time for these load changes for PI controller and PI-FLC are shown in Figs. 14 and 15 respectively. An abrupt change can be seen in the results with the 25% load as it observes in Fig. 14. The control signal gets out to the motor during the Digital-to-Analog Converter (DAC) output channel of FPGA device. It is represented in VHDL code by 12-bit vector. Once it is fully ones and the analog signal reaches the maximum value (3.3v), the next step it will start from initial zeros. The PI-FLC is able to compensate the load changes considerably better than the PI controller as shown in Figs. 14 and 15 and Table 4.

8. Conclusions

The main contribution of this work is to present an approach of real time fuzzy logic controller for DC motor using VHDL description and FPGA technology. FLC based on FPGA has some advantages such as flexible design (the membership functions and rule base can be easily changed), feature accuracy, high reliability, cost improvement, and high speed. The performance of the FLC is compared with the PI controller for reference speed and load changes. The satisfied ability of the system control with FLC is better than PI controller. In a future work it is planned to investigate implementation of adaptive fuzzy controllers on FPGA.

References

[1] Hamed B, Al-Mobaied M. Fuzzy PID controllers using FPGA technique for real time DC motor speed control. Intell Control Autom 2011;2(August):233–40.

[2] Millan I, Montiel O, Sepulveda R, Castillo O. Design and implementation of a hybrid fuzzy controller using VHDL. Soft computing for hybrid Intel. systems. Berlin, Heidelberg: Springer-Verlag; 2008, SCI 154.

[3] Nagib G, Gharieb W, Binder Z. Application of fuzzy control to a non-linear thermal process. In: The 31st conference of the IEEE on decision and control, Tucson, Arizona, December 1992.

[4] Kumar V, Rana KPS, Kumar V. Real time comparative study of the performance of FPGA based PID and fuzzy controllers for a rectilinear plant. In: IEEE conference publications, power electronics (IICPE), 2010 India international conference, 28–30, January 2011.

[5] Kaur A, Kaur A. Development of neuro fuzzy controller algorithm for air conditioning system. (IJEST) Int J Eng Sci Technol 2012;4(04):1667–71, ISSN: 0975-5462.

[6] Sakthivel G, Anandhi TS, Natarajan SP. Design of optimized fuzzy logic controller for area minimisation and its FPGA implementation. (IJCSNS) Int J Comput Sci Network Secur 2010;10(8):187–92.

[7] Precup RE, Hellendoorn H. A survey on industrial applications of fuzzy control", computers in industry, vol. 62. Elsevier; 2011, p. 213–226.

[8] Kung YSh, Fung RF, Tai TY. Realization of a motion control IC for X–Y table based on novel FPGA technology. IEEE Trans Industr Electron 2009;56(1):43–53.

[9] Hamed BM, El-Moghany MS. Fuzzy controller design using FPGA for sun tracking in solar array system. I.J. Intell Syst Appl 2012;1(February):46–52.

[10] Monmasson Eric, Cirstea Marcian N. FPGA design methodology for industrial control systems – a review. IEEE Trans Industr Electron 2007;54(4).

[11] Sakthivel G, Anandhi TS, Natarajan SP. Real time implementation of a fuzzy logic controller on FPGA using VHDL for DC motor speed control, G. Sakthivel et al.. Int J Eng Sci Technol 2010;2(9):4511–9.

[12] Monmasson E, Cirstea MN. FPGA design methodology for industrial control systems–a review. IEEE Trans Industr Electron 2007;54(4).

[13] Karasakal O, Yesil E, Guzelkaya M, Eksin I. Implementation of a new self-tuning fuzzy PID controller on PLC. Turk J Electr Eng 2005;13(2):277–86.

[14] Tipsuwanpornm V, Runglimmawan T, Intajag S, Krongratana V. Fuzzy logic PID controller based on FPGA for process control. In: IEEE international symposium on industrial, electronics, vol. 2, 4–7 May, 2004. P. 1495–1500.

[15] Sulaiman N, Obaid ZA, Marhaban MH, Hamidon MN. FPGA-based fuzzy logic: design and applications – a review. (IACSIT) Int J Eng Technol 2009;1(5):491–503, ISSN: 1793-8236.

[16] Hassan MY, Sharif WF. Design of FPGA based PID-like fuzzy controller for industrial applications. IAENG Int J Comput Sci 2007;34(2), 17 November, IJCS-34-2-05.

[17] Kuo B. Automatic control systems. Englewood Cliffs, NJ: Prentice-Hall; 1995.

[18] Ogata K. Modern control engineering. 3rd ed. Upper Saddle River, NJ: Prentice-Hall; 1997.

[19] Qu L, Huang Y, Ling L. Design of intelligent PID controller based on adaptive genetic algorithm and implementation of FPGA. Berlin, Heidelberg: Springer-Verlag; 2008, Part II, LNCS 5264.

[20] Tipsuwan Y, Chow MY. Fuzzy PID controllers using 8-Bit microcontroller for U-board speed control. In: IECON '99', the 25th annual conference of the IEEE, vol. 3; 1999.

[21] Passino KM, Yurkovich S. Fuzzy control. Addison Wesley Longman, Inc.; 1998, ISBN 0-201-18074-X.

[22] Poplawski M, Bialko M. Implementation of parallel fuzzy logic controller in FPGA circuit for guiding electric wheelchair. Krakow, Poland: IEEE – HIS; 2008. p. 25–7, May.

[23] Vuong PT, Madni AM, Vuong JB. VHDL implementation for a fuzzy logic controller. In: IEEE conference publications, world automation congress, Budapest, Hungary, July 24–26, 2006.

[24] Sousa GCD, Bose BK. A fuzzy set theory based control of a phase- controlled converter Dc machine drive. IEEE Trans Ind Appl 1994;30(1).

[25] Di Stefano A, Giaconia C. An FPGA-based adaptive fuzzy coprocessor. Berlin, Heidelberg: Springer-Verlag; 2005, IWANN 2005, LNCS 3512.

[26] Lizárraga G, Sepúlveda R, Montiel O, Castillo O. Modeling and simulation of the defuzzification stage using xilinx system generator and simulink. Soft computing for hybrid intelligent systems. Berlin, Heidelberg: Springer Verlag; 2008.

[27] Wang HP. Design of fast fuzzy controller and its application on position control of DC motor. IEEE conference publications, (CECNet) consumer electronics communications and network. In: 2011 International conference; 2011. P. 4902–5.

[28] Eminoglu I, Altas IH. A method to form fuzzy logic control rules for a PMDC motor drive system. Electric power systems research, vol. 39. Elsevier; 1996, p. 81–87.

[29] Shrikanth G, Subramanian K. Implementation of FPGA based object tracking algorithm, Postgraduate Thesis, (SVCE, India); 2008.

[30] Spartan-3E FPGA Starter Kit Board User Guide, UG230, vol. 1, No. 1, 20 June, 2008. <http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf> [Last modified in January 2013].

Ebrahim Abd El-Hamid Mohamed Ramadan received the B.Sc. and M.Sc. degrees in automatic control engineering from Faculty of Electronic Engineering, Menofia University, Menof, Egypt. He is currently working toward the Ph.D. degree in automatic control engineering. He is currently Assistant Lecture with the department of Industrial Electronics and Control Engineering, Faculty of Electronic Engineering, Menofia University, Menof Egypt. His current research Embedded Intelligent Control System Based on FPGA.

Mohammad El-Bardini is currently an Associate Professor with the Department of Industrial Electronic and Control Engineering, Faculty of Electronic Engineering; Menoufia University. His research interests include Robotics, Computer Controlled Systems and the Embedded System design for Control Systems. He has coauthored many journal and conference papers and has supervised many Ph.D. and M.Sc. for students working in the field of Intelligent Control Systems, Embedded Control Systems, Advanced Techniques for Systems Modeling, and Vision based Control of Robotic Systems. He is the recipient of the best students project Award in Egyptian Engineering Day.

Nabila M. El-Rabaie is currently a Professor with the Department of Industrial Electronic and Control Engineering, Faculty of Electronic Engineering; Menofia University. Her research interests include modelling, adaptive control and neural control. She has co-authored many journal and conference papers and has supervised many Ph.D. and M.Sc. for students working in the field of Adaptive Control Systems, Intelligent Control Systems, and Advanced Techniques for Systems Modelling.

Mohamed A. Fkirin is currently a Professor with the Department of Industrial Electronic and Control Engineering, Faculty of Electronic Engineering; Menoufia University. He got B.Sc. (1971-1976) and M.Sc. (1977-1980) in automatic control systems identification, in Egypt. After then he went to England and achieved the PhD degree (1982-1986), from Birmingham University. He has authored 55 publications (53 papers and 2 books). He has published 30 publications as single author. He has significant contributions in evolving appropriate strategies of optimal exploitation for online identifying and predicting of dynamic systems. In particular for his pioneering studies on the smoothing forms to identify the SISO and MIMO systems, and for his key role in developing new effective prediction algorithms based on Kalman and lattice filters. These strategies are used for solving prediction problems in developing countries (Egypt and Saudi Arabia), such as econometric, hydrological, industrial and medical applications. Also, his research interests include the development of advanced models of educational and computer-control systems. He was awarded the prize of Young Arab scientists, in 1991. He was selected to be formal referee in Jordanian board for choosing the new winners for the Young Arab scientists prize. He was awarded the First Prize of the Third International Symposium of Date Palm, Saudi Arabia, 1993.