

The Efficiency of Particle Swarm Optimization Applied on Fuzzy Logic DC Motor Speed Control

Boumediene Allaoua¹, Abdessalam Abderrahmani²,
Brahim Gasbaoui³, Abdelfatah Nasri⁴

Abstract: This paper presents the application of Fuzzy Logic for DC motor speed control using Particle Swarm Optimization (PSO). Firstly, the controller designed according to Fuzzy Logic rules is such that the systems are fundamentally robust. Secondly, the Fuzzy Logic controller (FLC) used earlier was optimized with PSO so as to obtain optimal adjustment of the membership functions only. Finally, the FLC is completely optimized by Swarm Intelligence Algorithms. Digital simulation results demonstrate that in comparison with the FLC the designed FLC-PSO speed controller obtains better dynamic behavior and superior performance of the DC motor, as well as perfect speed tracking with no overshoot.

Keywords: DC Motor speed control, Fuzzy logic controller, Intelligent fuzzy control, Particle swarm optimization.

1 Introduction

In spite of the development of power electronics resources, the direct current machines are becoming more and more useful insofar as they have found wide application, i.e. automobile industry (electric vehicle), weak power using battery system (motor of toy), the electric traction in the multi-machine systems, etc.

The speed of DC motor can be adjusted to a great extent so as to provide easy control and high performance [1, 2]. There are several conventional and numeric controller types intended for controlling the DC motor speed at its executing various tasks: PID Controller, Fuzzy Logic Controller; or the combination between them: Fuzzy-Swarm, Fuzzy-Neural Networks, Fuzzy-Genetic Algorithm, Fuzzy-Ants Colony.

¹Boumediene ALLAOUA: University Center of Bechar, B.P 417 Bechar (08000) Algeria;
E-mail: elec_allaoua2bf@yahoo.fr

²Abdessalam ABDERRAHMANI: University Center of Bechar, B.P 417 Bechar (08000) Algeria;
E-mail: abderrahmani@yahoo.fr

³Brahim GASBAOUI: University Center of Bechar, B.P 417 Bechar (08000) Algeria;
E-mail: brahim_gasb@yahoo.com

⁴Abdelfatah NASRI: University Center of Bechar, B.P 417 Bechar (08000) Algeria;
E-mail: nasriab1978@yahoo.fr

Fuzzy theory was first proposed and investigated by Prof. Zadeh in 1965. The Mamdani fuzzy inference system was presented to control a steam engine and boiler combination by linguistic rules [3, 4]. Fuzzy logic is expressed by means of if-then rules with the human language. In the design of a fuzzy logic controller, the mathematical model is not necessary. Therefore the fuzzy logic controller is of good robustness. Owing to its easy application, it has been widely used in industry.. However, the rules and the membership functions of a fuzzy logic controller are based on expert experience or knowledge database.

Much work has been done on the analysis of fuzzy control rules and membership function parameters [4]. The PSO (particle swarm optimization) algorithms were used to get the optimal values and parameters of our FLC. The PSO is based on a metaphor of social interaction. It searches a space by adjusting the trajectories of individual vectors, called ‘particles’, as they are conceptualized as moving as points in multidimensional space. The individual particles are drawn stochastically towards the positions of their own previous best performances and the best previous performance of their neighbours. Since its inception, two notable improvements have been introduced on the initial PSO which attempt to strike a balance between two conditions. The first one introduced by Shi and Eberhart [5] uses an extra ‘inertia weight’ term which is used to scale down the velocity of each particle and this term is typically decreased linearly throughout a run. The second version introduced by Clerc and Kennedy [6] involves a ‘constriction factor’ in which the entire right side of the formula is weighted by a coefficient. Their generalized particle swarm model allows an infinite number of ways in which the balance between exploration and convergence can be controlled. The simplest of these is called PSO. The PSO algorithms are applied to choose membership functions and fuzzy rules. However, the expert experiences or knowledge are still necessary for the ranges of membership functions. In this paper, a novel strategy is proposed for designing the optimal fuzzy controller.

PSO algorithms are applied to search globally optimal parameters of fuzzy logic. The best ranges of membership functions, the best shapes of membership functions and the best fuzzy inference rules are dug out at the same time. Furthermore, the performances of three different fuzzy logic controllers are compared. Simulation results are given to show the effectiveness of FLC-Swarm controller.

2 Model of DC motor

DC machines are characterized by their versatility. By means of various combinations of shunt, series, and separately-excited field windings they can be designed to display a wide variety of volt-ampere or speed-torque characteristics for both dynamic and steady-state operation. Because of the ease with which

they can be controlled systems of DC machines have been frequently used in many applications requiring a wide range of motor speeds and a precise output motor control [7, 8].

In this paper, the separated excitation DC motor model is chosen for its good electrical and mechanical performances rather than other DC motor models. The DC motor is driven by applied voltage. Fig. 1 shows the equivalent circuit of DC motor with separate excitation. The characteristic equations of the DC motor are represented as:

$$\frac{d}{dt} i_{ex} = \left(-\frac{R_{ex}}{L_{ex}} \right) i_{ex} + \left(\frac{1}{L_{ex}} \right) V_{ex} \quad (1)$$

$$\frac{d}{dt} i_{ind} = \left(-\frac{R_{ind}}{L_{ind}} \right) i_{ind} + \left(\frac{-L_{index}}{L_{ind}} \right) \omega_r i_{ex} + \left(\frac{1}{L_{ind}} \right) V_{ind} \quad (2)$$

$$\frac{d}{dt} \omega_r = \left(\frac{L_{index}}{J} \right) i_{ex} i_{ind} + \left(\frac{-Cr}{J} \right) + \left(\frac{-fc}{J} \right) \omega_r \quad (3)$$

The equivalent circuit of DC motor with separate excitation illustrated in Fig. 1.

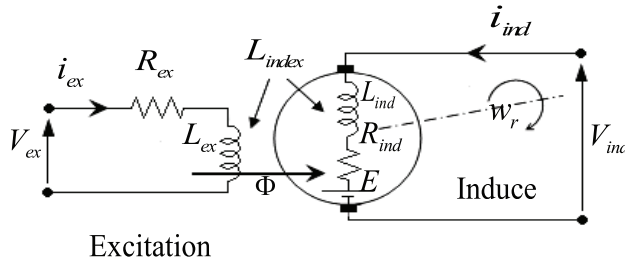


Fig. 1 – Equivalent circuit of DC motor with Separate Excitation.

Table 1
Used symbols.

Symbols	Designations	Units
i_{ex}, i_{ind}	Excitation current and Induced current.	[A]
ω_r	Rotational speed of the DC Motor.	[rad/S]
V_{ex}, V_{ind}	Excitation voltage and Induced voltage	[Volt]
R_{ex}, R_{ind}	Excitation Resistance and Induced Resistance.	[Ω]
$L_{ex}, L_{ind}, L_{index}$	Excitation Inductance, Induced Inductance and Mutual Inductance.	[mH]
J	Moment of Inertia.	[Kgm ²]
Cr	Couple resisting.	[Nm]
fc	Coefficient of Friction.	[Nms/rad]

Mathematical model expressed by the equations (1), (2), (3) can be presented by the MATLAB 7.4 (R2007a) model in Simulink version 6.6. The model of the DC motor in Simulink is shown in Fig. 2. Various parameters of the DC motor are shown in Table 2.

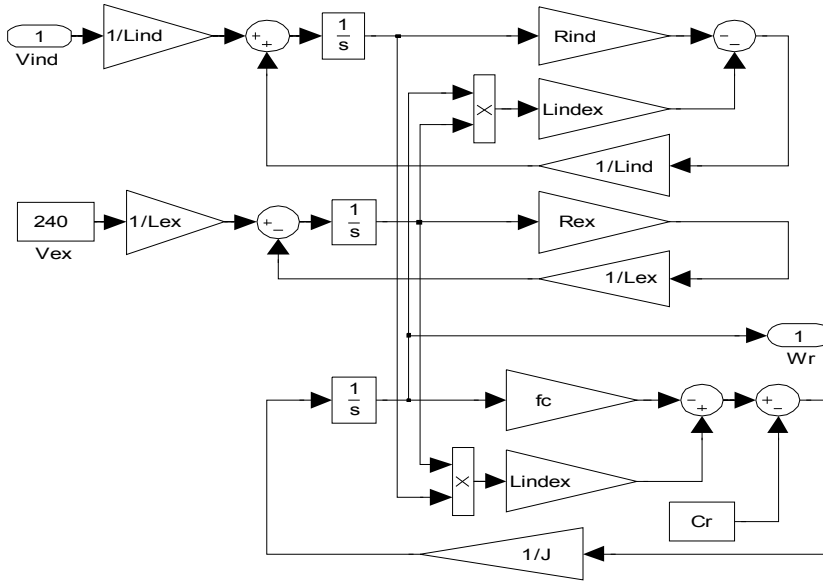


Fig. 2 – Model of the DC Motor in Simulink.

Table 2
Parameters of the DC Motor.

$V_{ex} = 240$ [V]	$L_{ind} = 0.012$ [mH]
$V_{ind} = 240$ [V]	$L_{index} = 1.8$ [mH]
$R_{ex} = 240$ [Ω]	$J = 1$ [Kgm ²]
$R_{ind} = 0.6$ [Ω]	$Cr = 29.2$ [Nm]
$L_{ex} = 120$ [mH]	$fc = 0.0005$ [Nms/rad]

3 Fuzzy Logic Controller

Fuzzy logic is expressed by means of the human language [9]. Based on fuzzy logic, a fuzzy controller converts a linguistic control strategy into an automatic control strategy, and fuzzy rules are constructed by expert experience or knowledge database.

First, set the error $e(t)$ and the error variation $de(t)$ of the angular velocity to be the input variables of the fuzzy logic controller. The control voltage $u(t)$ is the output variable of the fuzzy logic controller.

The linguistic variables are defined as {**NB**, **NS**, **Z**, **PS**, **PB**} meaning negative big, negative small, zero, positive small and positive big respectively. The membership functions of the fuzzy logic controller are shown in Fig. 3. The fuzzy rules are summarized in **Table 3**. The type of fuzzy inference engine is Mamdani.

The fuzzy inference mechanism in this study follows as:

$$\mu_B(u(t)) = \max_{j=1}^m [\mu_{A_1^j}(e(t)), \mu_{A_2^j}(de(t)), \mu_{B^j}(u(t))] \quad (4)$$

where $\mu_{A_1^j}(e(t))$ is the membership function of $e(t)$, $\mu_{A_2^j}(de(t))$ is the membership function of $de(t)$, $\mu_{B^j}(u(t))$ is the membership function of $u(t)$, j is an index of every membership function of fuzzy set, m is the number of rules and is the inference result. Fuzzy output $u(t)$ can be calculated by the center of gravity defuzzification as:

$$u(t) = \frac{\sum_{i=1}^m \mu_B(u_i(t)) u_i}{\sum_{i=1}^m \mu_B(u_i(t))} \quad (5)$$

where i is the output rule after inferring.

Table 3
Fuzzy inference rules.

$u(t)$		$e(t)$				
		NB	NS	Z	PS	PB
$de(t)$	NB	NB	NB	NS	NS	Z
	NS	NB	NS	NS	Z	PS
	Z	NS	NS	Z	PS	PS
	PS	NS	Z	PS	PS	PB
	PB	Z	PS	PS	PB	PB

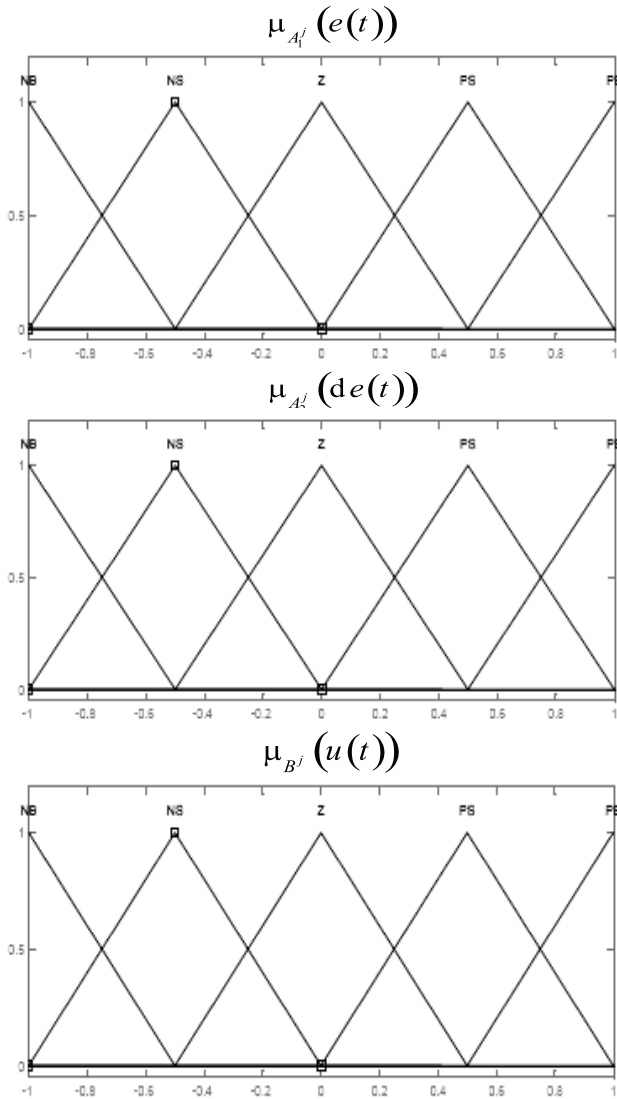


Fig. 3 – Membership function of fuzzy logic controller.

4 Particle Swarm Optimization (PSO)

PSO is a population-based optimization method first proposed by Eberhart and Colleagues [10, 11]. Some of the attractive features of PSO include the ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems. Like evolutionary algorithms, PSO technique conducts search using a population of particles,

corresponding to individuals. Each particle represents a candidate solution to the problem at hand. In a PSO system, particles change their positions by flying around in a multidimensional search space until computational limitations are exceeded. Concept of modification of a searching point by PSO is shown in Fig. 4.

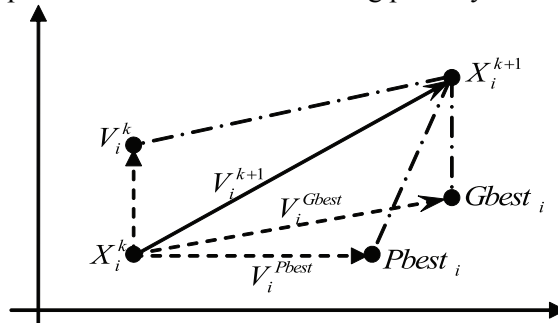


Fig. 4 – Concept of modification of a searching point by PSO.

X^k : current position, X^{k+1} : modified position, V^k : current velocity,

V^{k+1} : modified velocity, V^{Pbest} : velocity based on Pbest,

V^{Gbest} : velocity based on Gbest.

The PSO technique is an evolutionary computation technique, but it differs from other well-known evolutionary computation algorithms such as the genetic algorithms. Although a population is used for searching the search space, there are no operators inspired by the human DNA procedures applied on the population. Instead, in PSO, the population dynamics simulates a ‘bird flock’s’ behavior, where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all the other companions during the search for food. Thus, each companion, called *particle*, in the population, which is called *swarm*, is assumed to ‘fly’ over the search space in order to find promising regions of the landscape. For example, in the minimization case, such regions possess lower function values than other, visited previously. In this context, each particle is treated as a point in a d-dimensional space, which adjusts its own ‘flying’ according to its flying experience as well as the flying experience of other particles (companions). In PSO, a particle is defined as a moving point in hyperspace. For each particle, at the current time step, a record is kept of the position, velocity, and the best position found in the search space so far.

The assumption is a basic concept of PSO [11]. In the PSO algorithm, instead of using evolutionary operators such as mutation and crossover, to manipulate algorithms, for a d-variables optimization problem, a flock of particles are put into the d-dimensional search space with randomly chosen velocities and positions knowing their best values so far (Pbest) and the position

in the d-dimensional space. The velocity of each particle, adjusted according to its own flying experience and the other particle's flying experience. For example, the i -th particle is represented as $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ in the d-dimensional space. The best previous position of the i -th particle is recorded and represented as:

$$Pbest_i = (Pbest_{i,1}, Pbest_{i,2}, \dots, Pbest_{i,d}). \quad (6)$$

The index of best particle among all of the particles in the group is $gbest_d$. The velocity for particle i is represented as $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,d})$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $Pbest_{i,d}$ to $gbest_d$ as shown in the following formulas [12]:

$$v_{i,m}^{(t+1)} = wv_{i,m}^{(t)} + c_1 Rand() (Pbest_{i,m} - x_{i,m}^{(t)}) + c_2 Rand() (gbest_m - x_{i,m}^{(t)}), \quad (7)$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)}, \quad i = 1, 2, \dots, n; \quad m = 1, 2, \dots, d, \quad (8)$$

where

n - Number of particles in the group,

d - dimension,

t - pointer of iterations(generations),

$v_{i,m}^{(t)}$ - velocity of particle i at iteration t , $V_d^{\min} \leq v_{i,d}^{(t)} \leq V_d^{\max}$

w - Inertia weight factor,

c_1, c_2 - Acceleration constant,

$rand()$ - Random number between 0 and 1,

$x_{i,d}^{(t)}$ - Current position of particle i at iterations,

$Pbest_i$ - Best previous position of the i -th particle,

$Gbest$ - Best particle among all the particles in the population.

The evolution procedure of PSO Algorithms is shown in Fig. 5. Producing initial populations is the first step of PSO. The population is composed of the chromosomes that are real codes. The corresponding evaluation of a population is called the "fitness function". It is the performance index of a population. The fitness value is bigger, and the performance is better. The fitness function is defined as follow:

$$PI = MIN_offset - \sum |e| \quad (9)$$

where PI is the fitness value, e is the speed error and "MIN_offset" is a constant.

After the fitness function has been calculated, the fitness value and the number of the generation determine whether or not the evolution procedure is stopped (Maximum iteration number reached?). In the following, calculate the

Pbest of each particle and Gbest of population (the best movement of all particles). The update the velocity, position, gbest and pbest of particles give a new best position (best chromosome in our proposition).

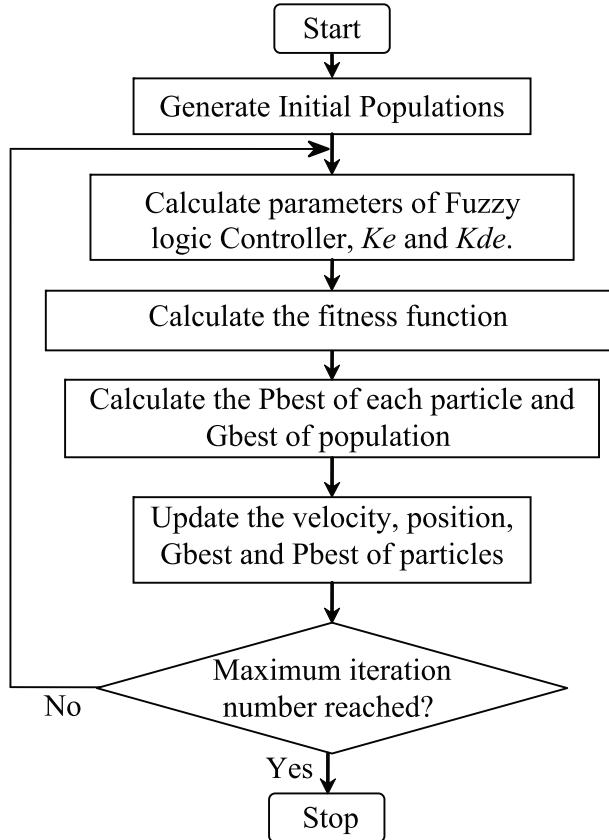


Fig. 5 – The evolution procedure of PSO Algorithms.

5 Optimal Fuzzy Controller Design

In order to design the optimal fuzzy controller, the PSO algorithms are applied to search globally optimal parameters of the fuzzy logic. The structure of the fuzzy logic controller with PSO algorithms is shown in Fig. 6.

In this paper, the chromosomes of the PSO algorithms include three parts: the range of the membership functions (K_e and K_{de}), the shape of the membership functions ($e_1 \sim e_5$, $de_1 \sim de_5$ and $u_1 \sim u_5$) and the fuzzy inference rules ($r_1 \sim r_{25}$). The output voltage is thereby such that the steady-state error of the response is zero. The genes in the chromosomes are defined as:

$$\begin{aligned}
 & [K_e, K_{de}, e_1, e_2, e_3, e_4, e_5, \\
 & de_1, de_2, de_3, de_4, de_5, \\
 & u_1, u_2, u_3, u_4, u_5, \\
 & r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, \\
 & r_{14}, r_{15}, r_{16}, r_{17}, r_{18}, r_{19}, r_{20}, r_{21}, r_{22}, r_{23}, r_{24}, r_{25}]
 \end{aligned} \tag{10}$$

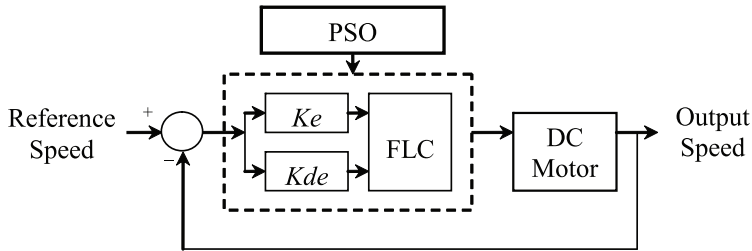


Fig. 6 – Structure of FLC with PSO algorithms.

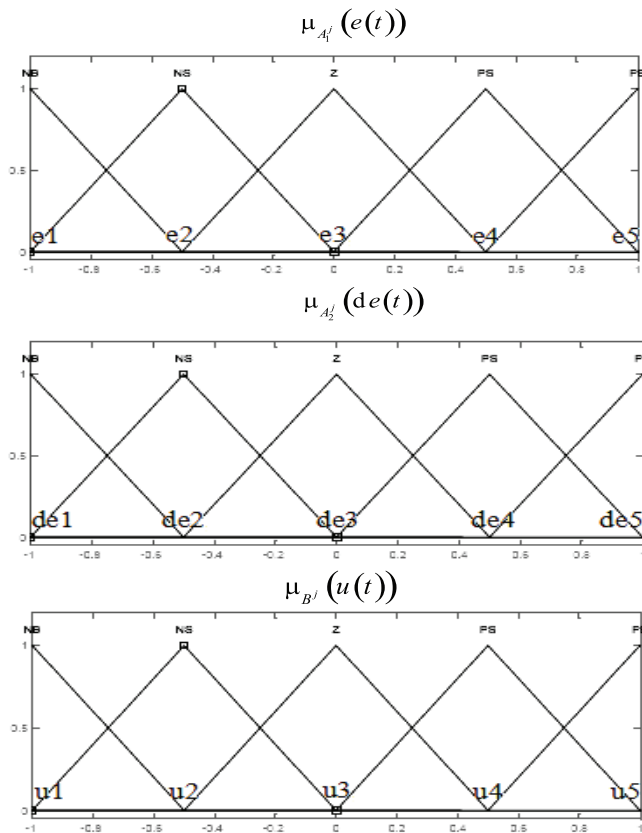


Fig. 7 – Membership function of fuzzy logic controller with PSO algorithms.

Fig. 7 shows the membership functions of the fuzzy logic controller with PSO algorithms. **Table 4** lists the fuzzy inference rules with PSO algorithms. **Table 5** lists the parameters of PSO algorithms used in this paper.

The fuzzy inference rules (r1~r25) are replaced of 1 (NB), 2 (NS), 3 (Z), 4(PS) and 5 (PB).

Table 4
Fuzzy inference rules.

$u(t)$		$e(t)$				
		NB	NS	Z	PS	PB
$d e(t)$	NB	r1	r6	r11	r16	r21
	NS	r2	r7	r12	r17	r22
	Z	r3	r8	r13	r18	r23
	PS	r4	r9	r14	r19	r24
	PB	r5	r10	r15	r20	r25

Table 5
Parameters of PSO algorithms.

Population Size	50	Ke and Kde	[0.001 0.005]
Number of Iterations	100	$e1$, $de1$ and $u1$	[-1 -0.5]
w_{max}	0.6	$e2$, $de2$ and $u2$	[-1 0]
w_{min}	0.1	$e3$, $de3$ and $u3$	[-0.5 +0.5]
$c_1 = c_2$	1.5	$e4$, $de4$ and $u4$	[0 +1]
<i>Min-offset</i>	200	$e5$, $de5$ and $u5$	[+0.5 +1]

6 Computer Simulation

Three different fuzzy logic controllers are designed for the computer simulation. First, fuzzy logic controller is based on the expert experience, as described in section 3. Second, the fuzzy logic controller is based on the PSO algorithms only to find the optimal range of the membership functions (FLC1 with PSO algorithms). Last, the optimal fuzzy controller is based on the PSO algorithms so as to search the optimal range of the membership functions, the optimal shape of the membership functions and the optimal fuzzy inference rules (FLC2 with PSO algorithms). After the evolution process, the optimal values of Ke and Kde in FLC1 with PSO algorithms are calculated as 0.005 and 0.005, respectively. The best chromosomes in FLC2 with PSO algorithms are pursued as:

B. Allaoua, A. Abderrahmani, B. Gasbaoui, A. Nasri

$$\begin{aligned}
 & [0.0047, 0.0039, -0.7860, -0.6640, \\
 & 0.2778, 0.6323, 0.6693, -0.7963, -0.7275, 0.4050, 0.1561, \\
 & 0.8070, -0.6799, -0.1561, -0.0661, 0.1085, 0.5899, \\
 & 1, 1, 1, 2, 4, 5, 2, 3, 3, 5, 5, 3, \\
 & 1, 1, 1, 4, 4, 4, 3, 3, 5, 1, 2, 3, 3]
 \end{aligned} \tag{11}$$

The optimal membership functions are shown in Fig. 8. The optimal fuzzy inference rules are listed in **Table 6**.

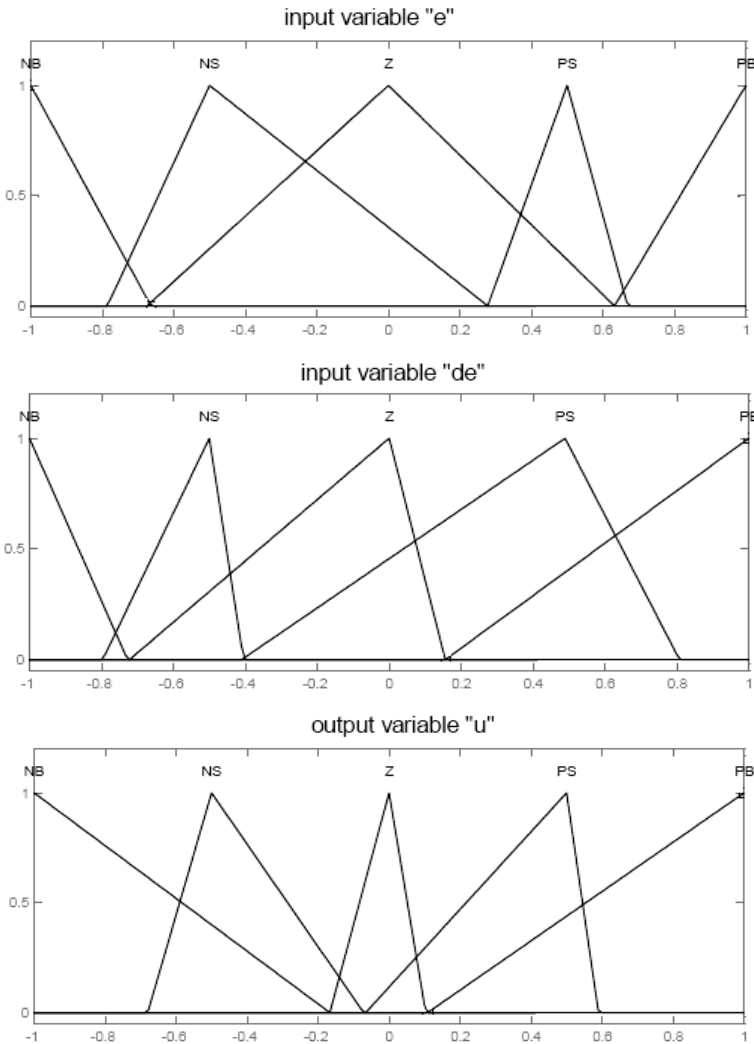


Fig. 8 – The optimal membership functions.

Table 6

The optimal fuzzy rules.

$u(t)$		$e(t)$				
		NB	NS	Z	PS	PB
$de(t)$	NB	NB	PB	PB	PS	PB
	NS	NB	NS	NB	PS	NB
	Z	NB	Z	NB	PS	NS
	PS	NS	Z	NB	Z	Z
	PB	PS	PB	NB	Z	Z

Let the command signal be a step for the speed of the DC motor at 127.93 Rad/Sec. The simulation results are obtained for 0.1 second range time.

The speed response of FLC (Fuzzy Logic Controller) without PSO algorithms is shown in Fig 9. The speed response of FLC1 with PSO algorithms is shown in Fig 10. The speed response of the optimal fuzzy controller is shown in Fig 11. The performances of three different fuzzy logic controllers are listed in **Table 7**.

Table 7

Performances of three fuzzy logic controllers.

Results	FLC without PSO algorithms	FLC1 with PSO algorithms	FLC2 with PSO algorithms
Rising time [Sec]	0.0209	0.0112	0.0087
Overtaking [%]	0	0	0
Steady state error [%]	0.45	0	0

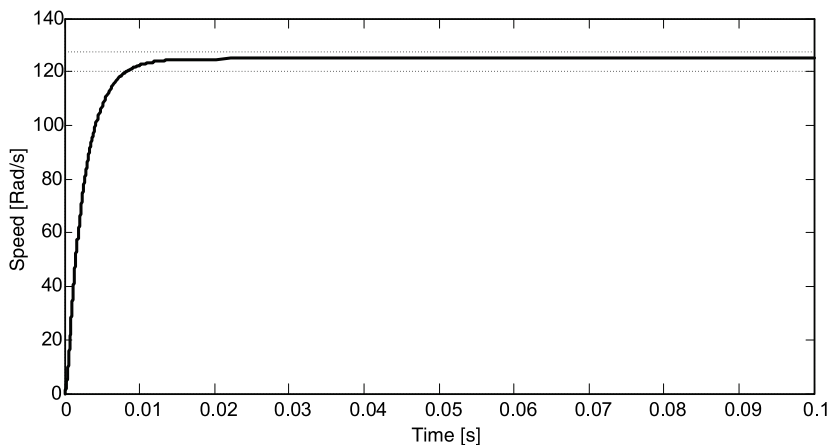


Fig. 9 – *The speed response of FLC without PSO algorithms.*

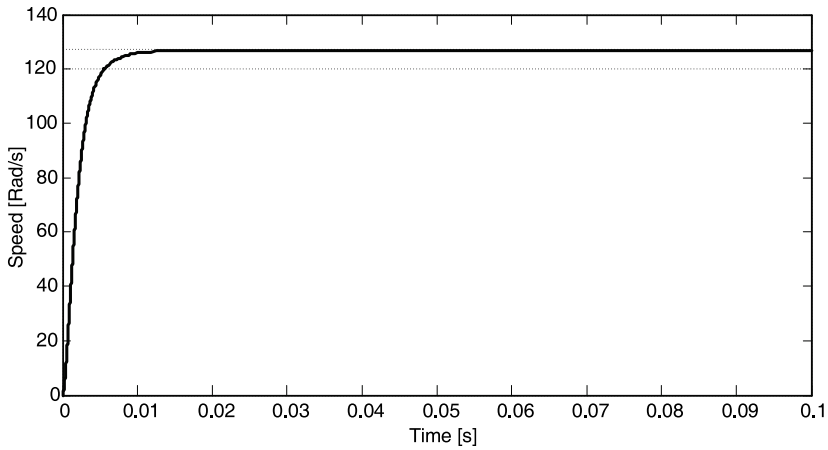


Fig. 10 – The speed response of FLC1 with PSO algorithms.

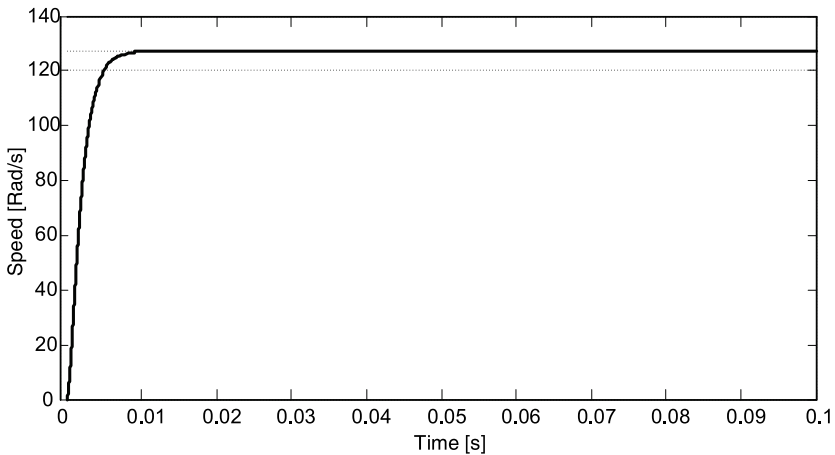


Fig. 11 – The speed response of FLC2 with PSO algorithms.

According to our MATLAB model simulation, we illustrate that the steady state error equals zero in two cases: FLC1 with PSO algorithms and FLC2 with PSO algorithms (Figure 10 and 11); the overtaking value is zero in three cases meaning that the FLC used is robust. The rising time of the DC motor speed step is less important in FLC1 with PSO algorithms compared with FLC alone and it is the minimal value in the FLC2 with PSO algorithms.

In the present work, the intelligent controller based on Fuzzy Logic-PSO Algorithms is in agreement with the step reference speed. In the fuzzy logic DC motor control, the optimization of membership functions and rules was required, its significance being shown in the minimal rising time of speed response. Therefore the membership functions are adjusted in optimal values so as to give

a steady state error speed value equal zero. The computer MATLAB simulation demonstrate that the fuzzy controller associated to the PSO algorithms approach became very strong, giving very good results and possessing good robustness.

7 Conclusion

In this paper, the speed of a DC Motor drive is controlled by means of three different fuzzy controllers. The optimal fuzzy logic is designed using PSO algorithms. According to the results of the computer simulation, the FLC1 with PSO algorithms is better than the traditional FLC without PSO algorithms. The FLC2 with PSO algorithms is the best controller which presented satisfactory performances and good robustness (no overshoot, minimal rise time, steady state error is 0). The major drawback of the fuzzy controller is insufficient analytical technique design (the selection of the rules, the membership functions and the scaling factors). We chose the one with the use of the PSO algorithm for the optimization of this controller in order to control DC motor speed. Finally, the proposed controller provides drive robustness improvement and gives very good results and possesses good robustness.

8 References

- [1] H. Hénao, G.A. Capolino: Méthodologie et application du diagnostic pour les systèmes électriques, Article invité dans Revue de l'Electricité et de l'Electronique (REE), No. 6, (in French), June 2002, pp. 79 – 86.
- [2] S. Raghavan: Digital Control for Speed and Position of a DC Motor, MS Thesis, Texas A&M University, Kingsville, August 2005.
- [3] J. Yen, R. Langari: Fuzzy Logic: Intelligence, Control, and Information, Prentice-Hall, 1999.
- [4] Y.S. Zhou, L.Y. Lai: Optimal Design for Fuzzy Controllers by Genetic Algorithms, IEEE Trans. On Industry Application, Vol. 36, No. 1, January/February 2000, pp. 93 – 97.
- [5] Y. Shi, R. Eberhart: A Modified Particle Swarm Optimizer, Proc. 1998 Int. Conf. on Evolutionary Computation – The IEEE World Congress on Computational Intelligence, Anchorage, May 1998, pp. 69 – 73.
- [6] M. Clerc, J. Kennedy: The Particle Swarm-explosion, Stability, and Convergence in a Multidimensional Complex Space, IEEE Trans. Evolutionary Computation, Vol. 6, No. 1, Feb. 2002, pp. 58 – 73.
- [7] A.Halila: Étude des machines à courant continu, MS Thesis, University of LAVAL, (in French), May 2001.
- [8] G.A. Capolino, G. Cirrincione, M. Cirrincione, H. Henao, R. Grisel: Digital Signal Processing for Electrical Machines, Invited paper, Proceedings of ACEMP'01 (Aegan International Conference on Electrical Machines and Power Electronics), Kusadasi, Turkey, June 2001, pp. 211 – 219.
- [9] J. Foran: Optimisation of a Fuzzy Logic Controller Using Genetic Algorithms, Doctoral Thesis, Texas University of America, USA, 2002.

B. Allaoua, A. Abderrahmani, B. Gasbaoui, A. Nasri

- [10] J. Kennedy, R. Eberhart: Particle Swarm Optimization, Proc. IEEE Int. Conf. on Neural Network, Vol. 4, 1995, pp. 1942 – 1948.
- [11] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, Y. Nakanishi: A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment, IEEE Trans. on Power Systems, Vol. 15, No. 4, Nov. 2000, pp. 1232 – 1239.
- [12] Z.L. Gaing: A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System, IEEE Trans. Energy Conversion, Vol. 19, No. 2, June 2004, pp. 384 – 391.