# Fuzzy Logic Controller Optimized by Particle Swarm Optimization for DC Motor Speed Control

Rasoul Rahmani*, *Member, IEEE,* M.S. Mahmodian**, Saad Mekhilef**, *Member, IEEE* and A. A. Shojaei*

**Centre for Artificial Intelligence & Robotics, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia*
***Department of Electrical Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia*

***Abstract*—In this paper, we presented an optimized fuzzy logic controller using particle swarm optimization for DC motor speed control. The controller model is simulated using MATLAB software and also experimentally tested on a laboratory DC motor. A comparison of the performance of different controllers such as PID controller, fuzzy logic controller and optimized fuzzy logic controller is presented as well. With reference to the results of digital simulations and experiment, the designed FLC-PSO speed controller obtains much better dynamic behavior compared to PID and the normal FLC designed. Moreover, it can acquire superior performance of the DC motor, and also perfect speed tracking with no overshoot. The optimized membership functions (MFs) are obviously proved to be able to provide a better performance and higher robustness in comparison with a regular fuzzy model, when the MFs were heuristically defined. Besides, experimental results verify the ability of proposed FLC under sudden change of the load torque which leads to speed variances.**

***Index Terms*—fuzzy logic controller, particle swarm optimization, DC motor, optimization**

## I. Introduction

DIRECT current motors (DC) are in wide demand for a variety of applications, due to a number of advantages such as preciseness, fast adaptation, smooth operation, high torque capabilities, electrical efficiency and high power density. DC motor drives are commonly used [1], [2] for industrial products and services, such as food and chemical industries, machine tools, robotics, aeronautics and electric vehicles [3]. A good speed control system makes the DC motor appropriate for the applications in which adjustable speed variation, proper speed regulation, frequent starting, braking and reversing are required. The speed control of DC machines which used to be performed mechanically, has undergone a revolution as a result of advances in the area of power electronics. The operation of adjustable speed drives may be acheived by armature voltage control for speeds below the rated, or by field excitation variation for above rated speeds.

Great advances have been achieved in the process control techniques during the recent decades. The most common industrial type is PID controller due to its simple structure and robust performance in a variety of operating conditions [4]. For the conditions, in which there are changes of the motor or load dynamics, the performance of conventional PID of state feedback controllers is improper. PID controller is able to consider one order differential of the error signal, but not the acceleration and other high-order varying rates, and as a matter of fact it can hardly gain an ideal control effect for high order nonlinear uncertain systems [5], [6], [7]. Moreover, the large actuation, required for PID to obtain a precise control, is not practical. Although the Ziegler-Nichols tuning formula is perhaps the best known tuning method [8], it is quite difficult to tune the PID gains properly, due to the problems encountered in industrial plants such as high order, time delays, and nonlinearities. Artificial Intelligence control techniques; i.e, fuzzy logic control (FLC), artificial neural network (ANN) control and adaptive control models have been extensively proposed and improved to overcome the mentioned problems.

Fuzzy logic system has a pivotal role as an applicable controller and also for system identification, and pattern recognition problems [9], [10]. Undertaking a design optimization process is indispensable for the purpose of maximizing its performance. In such an optimization, maximization of the performance of a pre-defined criterion is obtained by tuning the adjustable fuzzy logic parameters. The fuzzy controllers applied to process control systems have been recently receiving more attention. In the design procedure of a fuzzy logic controller, the mathematical models are not essential. However, when it gets to the ranges and the rules of membership functions, expertise is an inevitable part. There are a wide range of random search methods getting more popular to decrease the complexity of fuzzy logic rules and membership functions. Genetic algorithm (GA) [11], simulated annealing (SA), ant colony optimization (ACO), particle swarm optimization (PSO) [12], and bees algorithm (BA) are some of these methods that are helpful to achievea a high efficiency in searching the global optimum solution [13]. The analysis of the membership function parameters and fuzzy control rules have always been the points of interest among researchers and engineers [14]. The Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart, is one of modern heuristic algorithms [15], [16], [17], developed through simulation of a simplified social system. It has been found to be robust in solving continuous nonlinear optimization problems. Since it is considered as a heuristic and stochastic algorithm, it does not need any mathematical information of the fitness function such as gradient derived or any statistic error function.

Current study proposes an optimized fuzzy logic controller for DC motor speed control and also presents a comparison of performance of controllers such as PID controller, fuzzy logic controller and optimized fuzzy logic controller by Particle Swarm Optimization. A short description of the implemented algorithms and simulation details followed by simulation and experimental results construct the main structure of this paper.

## II. Materials and Methods

To have a fair comparison between the controllers, the control model used is the same for all of them. An unit negative feedback loop system is used which is shown in Fig. 1. For simulation and analysis purpose, MATLAB/Simulink software is used to model the PID and Fuzzy Logic controllers in serial with a DC motor's transfer function. Also, Embedded Matlab Function is used to implement PSO in finding an optimized FLC. In all the simulations, the step response is used as an indicator of the controller performance.

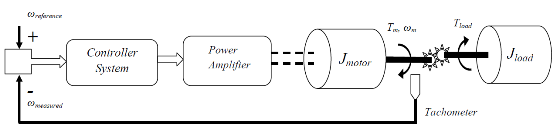



Figure 1. Control model used for motor speed control system

### A. Fuzzy Logic Controller

Fuzzy Logic (FL) is one of the most popular control methods which is known by its multi-rule-based variable's consideration. This method provides faster results compared to other Artificial Intelligent control methods such as Genetic Algorithm and Neural Networks. Being fast and robust are the main reasons of choosing FL for controlling purpose in the current study. Alike any other controller, FL needs some inputs to generate some control signal. The inputs of FLC are chosen based on the variances in speed. Equations (1) and (2) show the chosen inputs for the FLC system.

$$e\left(t\right)=\frac{\omega_{ref}-\omega_{measured}}{\omega_{ref}} \tag{1}$$

$$\widehat{e}\left(t\right)=\frac{\Delta e\left(t\right)}{\Delta t}=e\left(t\right)-e\left(t-1\right) \tag{2}$$

To implement the FL in a problem, different steps of this algorithm must be taken which are as follows:

*1) Fuzzification :* The input defined in equations (1) and (2) need to be fuzzified by some membership functions. For each input value, the respective membership function returns a value of $\mu$. The max-min method was applied to extract the from the triangle type membership function. Figure 5 depicts the membership functions for inputs $e$ and $\Delta e$ and output $\Delta D$ which is the variation needs to be applied to the current $D$ value.

*2) Inference diagram:* A rule base must be applied to the obtained membership function according to Mamdani. The rule table is designed and shown in Table 1. The 3D input-output surface is also obtained by using MATLAB FL Inference (see Figure 6).

Table I
Rules table of Fuzzy Logic Controller

| | $\Delta e$ | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|---|
| $e$ | | | | | | |
| NB | | ZE | ZE | NB | NB | NB |
| NS | | ZE | ZE | NS | NS | NS |
| ZE | | NS | ZE | ZE | ZE | PS |
| PS | | PS | PS | PS | ZE | ZE |
| PB | | PB | PB | PB | ZE | ZE |

*3) Defuzzification:* For the Difuzzification, the centroid method 13 is applied to return a proper value for the duty cycle variation ($\Delta D$). The difuzzified output value of the FLC must be added to a reference value of duty cycle which is considered equal to 0.5 for the current study. The result is the optimum value of D that has to be sent to the buck-boost converter as a control signal.

### B. Particle Swarm Optimization

The PSO uses a vectorized search space where each particle in search space proposes a solution to the problem. It is a swarm intelligence based algorithm which uses location and velocity of the particles to evaluate them using a fitness function or so called objective function. For each particle, the best position visited during its flight in the problem search space referred to as "personal best" ($Pbest$). Personal best position means the one that yields the best fitness value for that particle. For a minimization task such as in this case, the position having the smallest function value is regarded as having the highest fitness. Also, the best position among all Pbest positions, is referred to as global best ($Gbest$). At each iteration, the velocity of each particle is modified using the current velocity and its distance from $Pbest$ and $Gbest$ which is represented by (6). $V_i^{k+1}$ as updated velocity of particle i leads the particle to a new position called $X_i^{k+1}$ (7). $X$ and $V$ are demonstrations of vectors which iteration result gives a new position for the particle. Figure 2 is a simple diagram which shows the movement of a typical particle. Inertia weight in PSO plays an important role because of its control on particle speed. Hence, a suitable selection of it is important. In current study the value of inertia weight decreases from 1.2 to 0.5 during a run time.

$$V_i^{k+1}=W\times V_i^k+c1\times r1\left(Gbest_i^k-X_i^k\right)$$

$$+c2\times r2\left(Pbest_i^k-X_i^k\right) \tag{3}$$

$$X_i^{k+1}=X_i^k+V_i^{k+1} \tag{4}$$

$i$= 1, 2,..., nop (number of particles); $k$=1, 2, ..., kmax (maximum iteration number)

Where: $k$ iteration number; $i$ particle number; $W$ inertia weight factor; $c1$ social constant; $c2$ cognetive constant; $r1$ and $r2$ random values between 0 and 1; $V_i^k$ velocity of particle $i$ at iteration $k$; $X_i^k$ position of particle $i$ at iteration $k$.

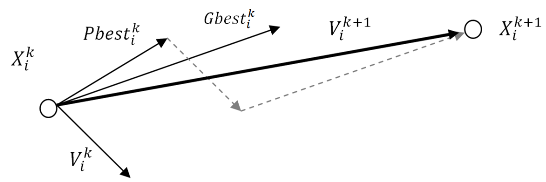



Figure 2. simple diagram for movement of a sample particle in PSO.

## III. OPTIMIZING FLC USING PSO

The PSO technique is employed to tune the FLC parameters as explained in [18]. The approach employs $MATLAB/M-file$ coding scheme in the $Simulink/EmbeddedMATLABFunction$ block. The parameters to be optimized are the fuzzy inference system and scaling gains for the inputs and output signals. At each iteration, fitness values are evaluated for all the particles based on (5) and based on evaluation results, $Pbest$ and $Gbest$ are updated. Afterwards, modifications are applied based on the updated values and next iteration starts. The maximum iteration number is chosen 80 and population size is set to be 20 particles. Table 2 tabulates the details about PSO parameters and Fig. 3 shows a simple flowchart of the algorithm. The only stoppage criteria in the algorithm is reaching the maximum iteration umber. The rule table of the Fuzzy Logic system can be found in Table 2.

$$fitness = t_r + t_s + OS + SSE \tag{5}$$

where: $t_r$is rising time; $t_s$is setteling time; $OS$ stands for Overshoot in percent and $SSE$ indicates the Steady State Error in percent.

Table II
PSO PARAMETERS INITIALIZATION

| Parameter | Value |
|---|---|
| Population size | 20 |
| Maximum itteration | 80 |
| Social rate ($c1$) | 0.9 |
| Cognetive rate ($c2$) | 2 |
| Inertia factor ($W$) | 1.2 ~ 0.5 |

Based on the ptimization process, the scaling gains for inputs $e$, $\Delta e$ and output are obtained as 1.5, 0.015 and 2000 respectively. The modified membership functions of inputs and outpt and also the surface diagram of inputs vs output are shown in Fig. 4.

## IV. RESULTS AND DISCUSSION

The simlation results are based on the paramerters obtained from previous section. Figure 5 depicts the designed model in the $MATLAB/Simulink$ environment. The design is based on the control model presented in Fig. 1. The simulation is run twice for FLC which contains the normal FLC and optimized FLC which is called FLCO here. To compare the FLC based obtained results with PID controller, a PID controller is also

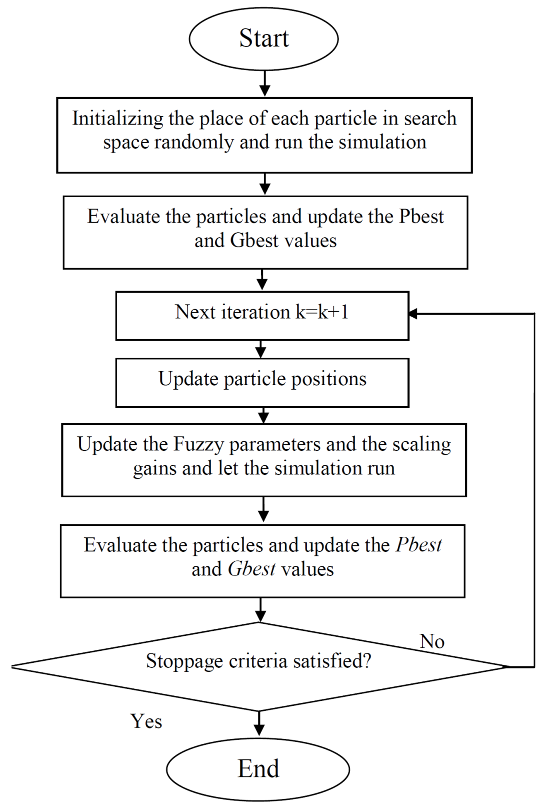



Figure 3. Simple flowchart of the algorithm used.

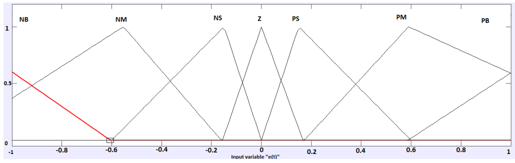
desiged with the same transfer function of DC motor and its parameters are set based on Table 3.

Table III
PARAMERTERS OF THE PID CONTROLLER

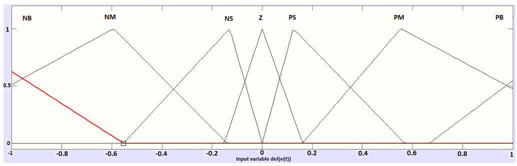
| Parameter | Value |
|---|---|
| Proportional coefficient (P) | 50 |
| Derivative coefficient (D) | 70 |
| Integral coefficient (I) | 6 |

The step response of three different controllers are graphed in Fig. 6. It shows that both FLC desiged have faster response than PID controller which was expected. However, PID controller works better than the normal FLC in terms of steady state error ($SSE$) and having lower overshoot ($OS$). The optimized FLC, so called FLCO, shows better profile in terms of fastness and accuracy compared to both other controllers. The exact values of the response parameters are tabulated in Table 4. There, The best value among all the three controllers are in bold for each parameters. the FLCO obviously shows betters performance for all the parameters.

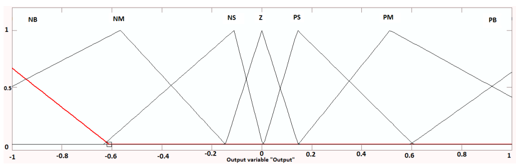
To verify the feasibility of the proposed controller, a laboratory DC motor is used to set up an experimental speed control. Two types of response are taken to account which are response



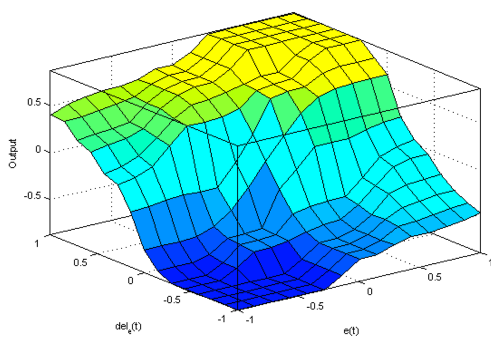

(a) Membership for input e(t)




(b) Membership for input $\Delta e(t)$




(c) Membership for Output




(d) Surface diagram for the inputs vs. output.

Figure 4. The modified membership functions for inputs and output signals.

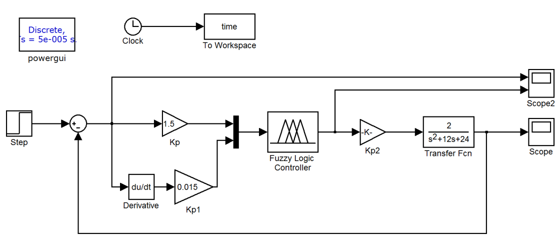



Figure 5. Designed model in MATLAB/Simulink environment.

to sudden load decrement and increment. The specifications of the hardware used are available in Appendix A. Two controllers are applied under the same circumstances to have a fair comparison, PID controller and the optimized FLC (FLCO). Wave forms of Fig. 7 are the results obtained from the experiment performed. The top waveform represents the speed response of the controllers to a sudden load increment

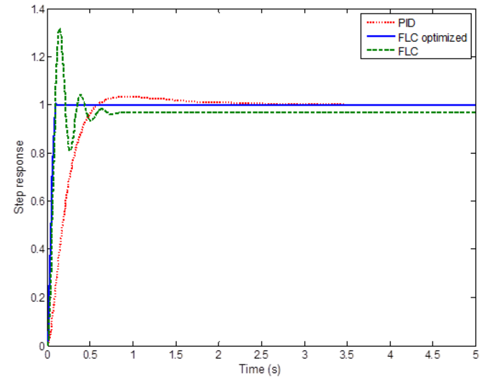



Figure 6. Step response results for different controllers.

Table IV
COMPARATIVE RESULTS.

| Metrics | Values | | |
|---|---|---|---|
| Controller | FLCO | FLC | PID |
| Rise time $t_r$ (s) | **0.065** | **0.065** | 0.354 |
| Settling time $t_s$ (s) | **0.13** | 0.95 | 2.15 |
| Overshoot $OS$ (%) | **0.5** | 31.5 | 3.3 |
| steady state error $SSE$ (%) | **0.05** | 3.46 | 0.06 |

while the bottom waveform is the speed response of the same controllers to a sudden load decrement. There is no overshoot or undershoot in the response of both controllers which is considered a good factor in the stability of the controllers. Both controllers also show low values of $SSE$ in the response. In terms of fastness of the controller, the FLCO can settle the waveform in almost 2 seconds while it takes between 7 to 9 seconds for the PID controller to settle the waveform to the desired value. Also, Fig. 8 shows the waveforms of response to the load increment (left) and decrement (right) for the FLCO on the digital oscilloscope. So, the results obtained from the simulation and experiment clarify that the designed and optimized FLC is capable of responding the DC motor speed changes fast and accurate.

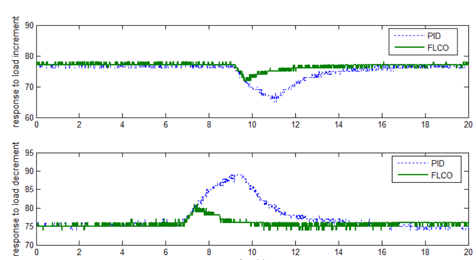



Figure 7. Experimental results for the speed response to the load increment and decrement.

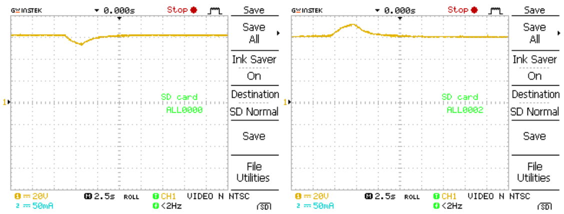



Figure 8. The FLCO response waveforms on the digital oscilloscope, load increment (left) and load decrement (right).

## V. Conclusions

In this paper, we proposed an optimized fuzzy logic controller for DC motor speed control and also presented a comparison of performance of controllers such as PID controller, fuzzy logic controller and optimized fuzzy logic controller by Particle Swarm Optimization. With reference to the results of the digital simulations, the designed FLC-PSO speed controller obtains much better dynamic behavior compared to PID and first FLC design. Moreover, it can acquire superior performance of the DC motor, and also perfect speed tracking with no overshoot. The optimized membership functions (MFs) are obviously proved to be able to provide a better performance and higher rebustness in comparison with a fuzzy model, when the MFs were heuristically defined. Besides, experimental results validate the ability of the proposed FLC under sudden changes of the load torque.

## Appendix A
## DC motor specifications

$P = 0.5hp; In = 3.8A; Te = 1.48N.m; L = 2.4mH; R = 2.5\Omega; p = 2; J = 225 \times 10^{-5} kg.m^2$.

## Acknowledgment

The authors would like to thank Ministry of Higher Education of Malaysia and University of Malaya for providing financial support under the research grant No.UM.C/HIR/MOHE/ENG/16001-00-D000024.

## References

[1] M. Islam, S. Mir, and T. Sebastian, "Issues in reducing the cogging torque of mass-produced permanent-magnet brushless dc motor," *Industry Applications, IEEE Transactions on*, vol. 40, pp. 813 – 820, may-june 2004.

[2] R. Krishnan, *Permanent magnet synchronous and brushless DC motor drives*. CRC Press/Taylor & Francis, 2010.

[3] C. Xia, M. Zhang, Y. Wang, and D. Liu, "The direct torque control for permanent magnet brushless dc motors," in *Zhongguo Dianji Gongcheng Xuebao(Proceedings of the Chinese Society of Electrical Engineering)*, vol. 28, pp. 104–109, Electric Power Research Institute(China), Qinghe Xiaoying Donglu 15, Beijing, 100085, China,, 2008.

[4] B. Ghosh, N. Xi, and T. Tarn, *Control in robotics and automation: sensor-based integration*. Academic Pr, 1999.

[5] J. Spooner, M. Maggiore, R. Ordonez, and K. Passino, *Stable adaptive control and estimation for nonlinear systems*. Wiley Online Library, 2002.

[6] S. Tong, H. Li, and W. Wang, "Observer-based adaptive fuzzy control for siso nonlinear systems," *Fuzzy sets and systems*, vol. 148, no. 3, pp. 355–376, 2004.

[7] F. Dinuzzo and A. Ferrara, "Higher order sliding mode controllers with optimal reaching," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2126 –2136, sept. 2009.

[8] J. Ziegler and N. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942.

[9] L. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.

[10] L. Zadeh, "Fuzzy sets as a basis for theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[11] R. Krohling and J. Rey, "Design of optimal disturbance rejection pid controllers using genetic algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 5, pp. 78 –82, feb 2001.

[12] S. Ghoshal, "Optimizations of pid gains by particle swarm optimizations in fuzzy based automatic generation control," *Electric Power Systems Research*, vol. 72, no. 3, pp. 203 – 212, 2004.

[13] C. Ou and W. Lin, "Comparison between pso and ga for parameters optimization of pid controller," in *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pp. 2471 –2475, june 2006.

[14] J. Yen and R. Langari, *Fuzzy logic: intelligence, control, and information*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942 –1948 vol.4, nov/dec 1995.

[16] R. Rahmani, A. Khairuddin, S. Cherati, and H. Pesaran, "A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (pso)," in *IPEC, 2010 Conference Proceedings*, pp. 134 –139, oct. 2010.

[17] K. Ishaque, Z. Salam, M. Amjad, and S. Mekhilef, "An improved particle swarm optimization (pso) based mppt for pv with reduced steady-state oscillation," *Power Electronics, IEEE Transactions on*, vol. 27, pp. 3627 –3638, aug. 2012.

[18] L. Letting, J. Munda, and A. Hamam, "Particle swarm optimized t-s fuzzy logic controller for maximum power point tracking in a photovoltaic system," in *IPEC, 2010 Conference Proceedings*, pp. 89 –94, oct. 2010.