Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 3, March 2019, pg.152 – 160

Improve DC Motor System using Fuzzy Logic Control by Particle Swarm Optimization in Use Scale Factors

Noorulden Basil Mohamadwasel[1]; Oguz Bayat[2]
[1,2](*Electrical and Computer Engineering, College of Graduate School, Altınbaş University, Istanbul, Turkey)*
[1] noraldenbasil4@gmail.com, [2] oguz.bayat@altinbas.edu.tr

Abstract: In our system design, we used Particle Swarm Optimization (PSO) Algorithm because we want to design a DC Motor system with Fuzzy logic control to provide high angular speed and low error. The DC Motor system is configured by MATLAB SIMULINK platform R2012a to be find which Method or algorithm will be used with the conventional controller the Proportional Derivative (PD) controller that will be improve this system like Particle Swarm Optimization (PSO) Algorithm has been utilized to improve from The performance of the Designed system. And the use of Particle Swarm Optimization (PSO).Algorithm will improve the Fuzzy logic control when connect with PD controller because PSO it will looking for the best 3 scales factors(gain) among 50 birds and we have set the no. of birds in PSO algorithm code and number of iteration is 10 it will make 50*10=500 process to find the best 3 gains to be used in the scale factors of fuzzy logic control instead of the static values that we have been putted statically it will be received dynamically from PSO algorithm and after that we have token these gains and insert it to fuzzy logic control scale factors. In this paper, three tests are taken to demonstrate between different proposals to studying the-DC Motor system based the Proportional Derivative (PD) controller with various formats by use Proportional Derivative (PD) controller with Fuzzy logic control and also by use this controller with fuzzy logic control by use the algorithm of Particle Swarm Optimization by test the angular speed and also the error to improve which one of them is the best, the designed system to improve it by increasing the angular speed and reduce the error this will improve the efficient of it.

Keywords: Proportional derivative controller, Fuzzy Logic Control, Particle Swarm Optimization, DC Motor System, MATLAB.

I. Introduction

Fuzzy logic is a form of logic used in some expert systems and applications of artificial intelligence. This logic originated in 1965 by the Azerbaijani scientist Lutfi Zada of the University of California[1], where he developed it to be used as a better method of data processing. The logic of ambiguity was used to regulate a steam engine, and its applications evolved until it reached the manufacturing of a fuzzy logic chip that was used in many products such as cameras[2].There are many motives that led scientists to develop the science of fuzzy logic With the development of computer and software arose the desire to invent or programming systems that can deal with inaccurate information like human [3] but this was born a problem since the computer can deal only with accurate and specific data. This trend has resulted in what is known as expert systems or artificial intelligence and fuzzy logic is one of the theories through which such systems can be built[4].The Fuzzy logic In a broad.

sense it is a logical system based on the generalization, of traditional bivalent logic, in order to infer in uncertain circumstances, In narrow sense, it is theories and techniques that use blurry groups that are infinite sets of boundaries. This logic is an easy way to describe and represent human experience, and it provides practical solutions to real problems, solutions that are cost effective and reasonable, compared to other solutions that offer other technologies[5].

II. Related Work

In previous work, In(2015) Noureddine Bouarroudj et al. [6] Discussed the development of a hybrid system, a fuzzy fracture, slip mode control for a class of interconnected nonlinear systems.

In(2009) Boumediène ALLAOUA and Brahim GASBAOUI and Brahim MEBARKI et al. [7] Discussed the DC Intelligent Controller. Use the Particle Optimization (PSO) method to form the best proportional derivative controller (PID). Adjust parameters. The DC controller is designed. environment. Compared with the mysterious logical controller using smart PSO algorithms, the schema. The graph is more efficient in improving the stability of loop response speed, the fixed state error is reduced, the time high. With no overrun.

In(2014) Manoj Kushwah and Prof. Ashis Patra et al. [8] They discussed the Tuning study of the integral derivative (PID). Control for speed control of DC motor. Using soft computing techniques. The AC motor is widely used in industries even if the cost of maintenance is higher than the induction engine. Control of DC motor speed is attracted. Research has evolved and many ways. The PID is a commonly used compensatory control unit used in nonlinear systems. This console is widely used. In many different areas such as space, and process. Control, manufacturing, automation, etc. Setting the PID parameter is very difficult. There are many soft computing techniques that are used to adjust the PID controller to control the DC motor speed. Adjusting PID parameters is important. Because, these parameters has a significant coefficient on the stability and performance of the control system.

III. DC Motor System Design with Proportional Derivative Controller

The idea of dc motor that comes from armature has the resistance and mutual inductance connected with back emf(electro-motive force) that generated inside armature coil [9] and the armature is connected with the load of the system the mechanical load and this connected with electrical Motor so the input voltage to the DC Motor is voltage and the output of DC Motor is angle (θ) or speed (ω) $\theta\,, \omega m = \frac{d\theta}{dt} = \dot{\theta}$ it's a derivative the angle according to the time in additional to that the Motor is dealing with this section so the-Kirchhoff Voltage Law (KVL) will taken for the loop circuit from left to right so the input voltage and resistance voltage and coil voltage and also back voltage V.emf inside armature so the KVL it will calculate as this formula $\sum Vi = 0$ the summation of voltages in one loop = 0 so the Equation it will calculate as this formula $Vi - VR - VL - eb = 0$ the armature current inside each element of circuit is series and back emf deal with Motor Torque speed Constant.

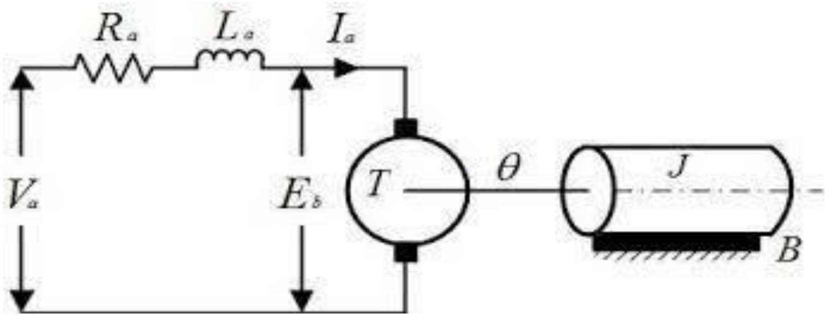

Fig. 1 . DC Motor circuit design.

Back V.emf that dealing with Motor speed ($\dot{\theta}$) and Motor Torque speed Constant is called K so $V.emf = K\omega m$ and ω its $\dot{\theta}$ so its equal to this formula $V.emf = K\dot{\theta} = K\frac{d\theta}{dt}$ and the ωm is the speed of Motor shift[10] and the voltage of coil is the coil factor multiply current derivative of the coil and it's the same armature current because the circuit is series according to the time and $VL = L\frac{dIa}{dt}$ and $VR = Ra\ Ia$ depending on ohm's law and these law's it will be Compensating in Kirchhoff Voltage Law and this will create this formula : $Vi - RaIa - L\frac{dIa}{dt} - k\frac{d\theta}{dt} = 0$ and it should to isolate the electrical part from mechanical part and each part in KVL when theta has inside part of equations its as mechanical part because its dealing with theta so this related with rotation so the new formula it will change because of changing in side of equation and each partition it will transfer to the part that dealing with it and the formula is

$$Vi - k\frac{d\theta}{dt} = Ra\ Ia + La\frac{dLa}{dt} \quad (1)$$

so the left part of equation is shown as mechanical part that dealing with back emf and the right part of equation is shown as electrical part and the equation should to convert to laplace transform to be implement and execute in MATLAB SIMULINK by taking laplace transform for equation (1) [11].In mechanical section any part has ability to rotate its called inertia torque (J) and he has the part that rotate surrounding the motor called Friction coefficient (B).Firstly the electrical torque Te its Torque constant multiply armature current its create a new formula:

$$Te = Kt\ Ia$$

The electrical Torque dealing with Armature Current and Kt Torque Constant

The main identification that depended on derivation for dynamic model of DC Motor is Electrical Torque equal Mechanical Torque[12].Secondly the mechanical torque Tm its inertia torque multiply in double derivation theta over time and also add Friction coefficient multiply by displacement its shown below in formula:

$$Tm = J\frac{d2\ \theta}{dt2} + B\ \frac{d\theta}{dt} \quad (2)$$

So if we take the efficiency=100% now all of electrical torque that converted to Mechanical torque this prove that Tm = Te and this concludes the new equation as a new formula:

$$J\ \frac{d2\theta}{dt2} + B\ \frac{d\theta}{dt} = K.Ia$$

J= polar moment of inertia.

B= damping constant.

K= mechanical torque constant.

TL= load Torque

J,B,K coefficients its changing from motor to another depending on the motor size and current….and so on and the equation that mentioned above dealing with J,B,K we will change it to block diagram to be used in MATLAB SIMULINK and the figure below describe the DC Motor Model[13].

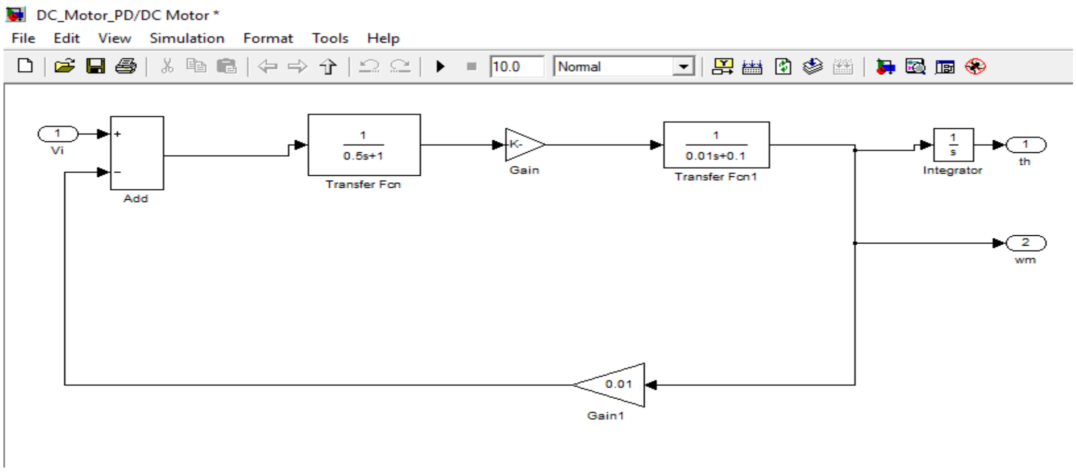

Fig. 2 . DC Motor design in MATLAB SIMULINK.

Firstly we will generate a current so we need voltageVi(S) that input to summation and the feedback will include at last KS $\theta(S)$ so we get this equation Vi(S)-KS $\theta(S)$ it will multiplied by $\frac{1}{Ra+LaS}$ it will multiply by Kt because the current when multiplied by k it will changed to Torque and take laplace transform for equation (4) so we will get this formula:

$$Tm(S) = JS\ \omega m(S) + B\omega m(S)$$

And the mechanical torque equation its second derivation or first derivation for speed so every $\frac{d\theta}{dt} = \omega m$ we can write it either in $\frac{d\theta}{dt}$ or ωm and $\omega m(S)$ it will take as common coefficient so should make control on speed and angle it will improve DC Motor System[14].

$$Tm(S) = \omega m(S)\ [JS + B]$$

From torque we will get the speed so the torque will multiply by $\frac{1}{JS+B}$ so in MATLAB SIMULINK the torque will multiplied by previous above block $\frac{1}{JS+B}$ we get the speed and the speed when take integral for it we get angle.

$$\omega m(S) = Tm(S).\frac{1}{JS + B}$$

$$\omega m = \frac{d\theta}{dt}$$

$$\omega m(S) = S\theta(S)$$

$$\theta(S) = \omega m(S) * \frac{1}{S}$$

First:built Proportional derivative Controller to connect it with DC Motor System. the figure that shown below describe the contents of PD Controller that connected with DC Motor system.

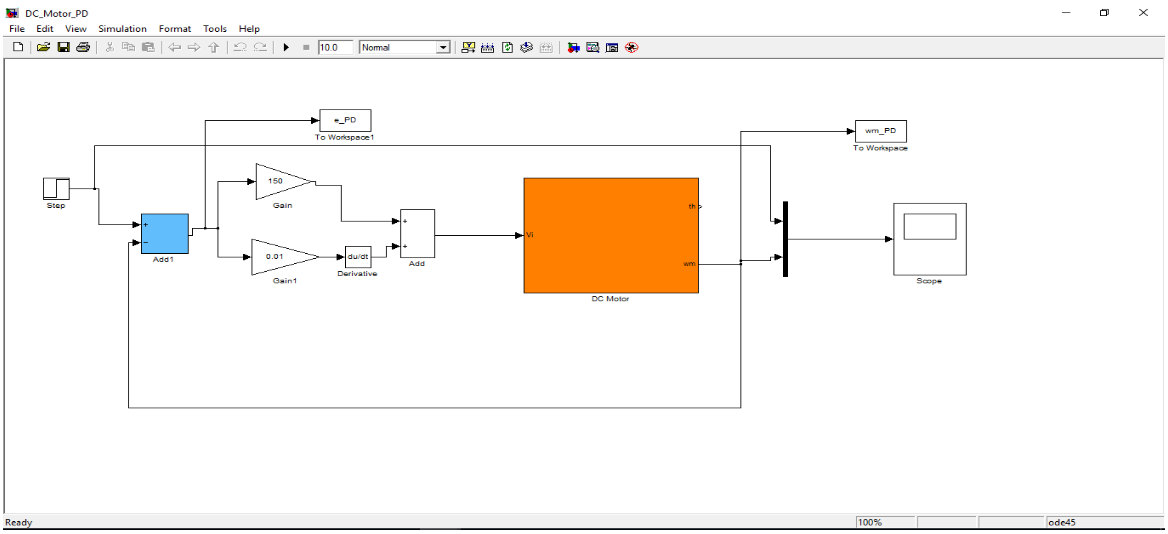

Fig. 3 . The Outside design for PD controller connected with DC Motor system by MATLAB SIMULINK.

The figure above describe proportional derivative it has two partitions (kp)proportional and (Kds)derivative its connected with DC Motor system and the control signal that mentioned below as equation.

$$u(t) = Kpe + Kd\frac{de}{dt} \qquad (5)$$

By take laplace transform for this equation to input it to MATLAB SIMULINK it will introduce to us this formula:

$$U(S) = Kp\ E(S) + Kds\ E(S) \qquad (6)$$

By take common coefficient

$$U(S) = E(S)[Kp + Kds]$$

IV. DC Motor System Design with Proportional derivative controller by Fuzzy logic control

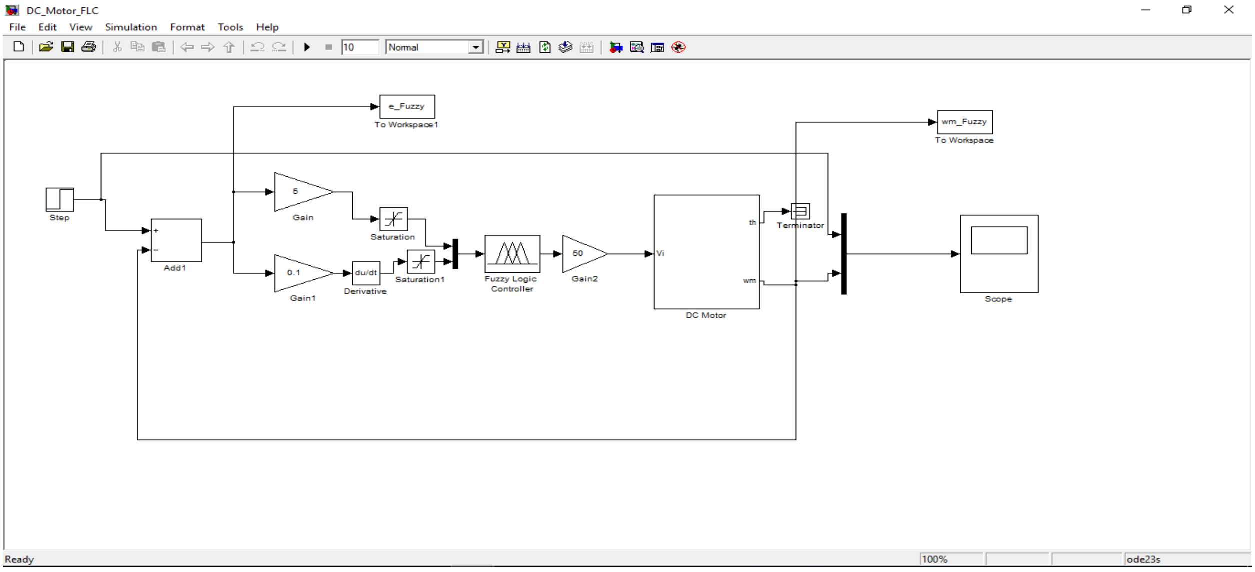

Fig. 4 . The outside design for fuzzy logic control connected with DC Motor system.

The proportional derivative controller has been connected using fuzzy logic control to be improved the DC Motor System and also reduce the steady state error and the maximum no. of overshoot and make the system stable more than in PD controller and the figure above describe the fuzzy logic that connected before DC Motor system and also improve the main coefficient the angular speed and also the error it will be compared with other controllers to see what's these controllers that improved in DC Motor system. The scope used to see the final results for the system and the others that connected by adder the first is ωm to check the angular speed what the thing that improved for this system but the error that connected in gains to check how many error has been occurred during applied the fuzzy logic control and what normalization has been improve.

V. DC Motor System with Proportional Derivative and Fuzzy Logic using PSO Algorithm

The Particle Swarm Optimization (PSO) in every execute operation that give the new values for Fuzzy Logic Control (FLC) and the number of birds is: 50 and iteration is 10 so the process is 500 times that make process on scale factors for best new values in every round.

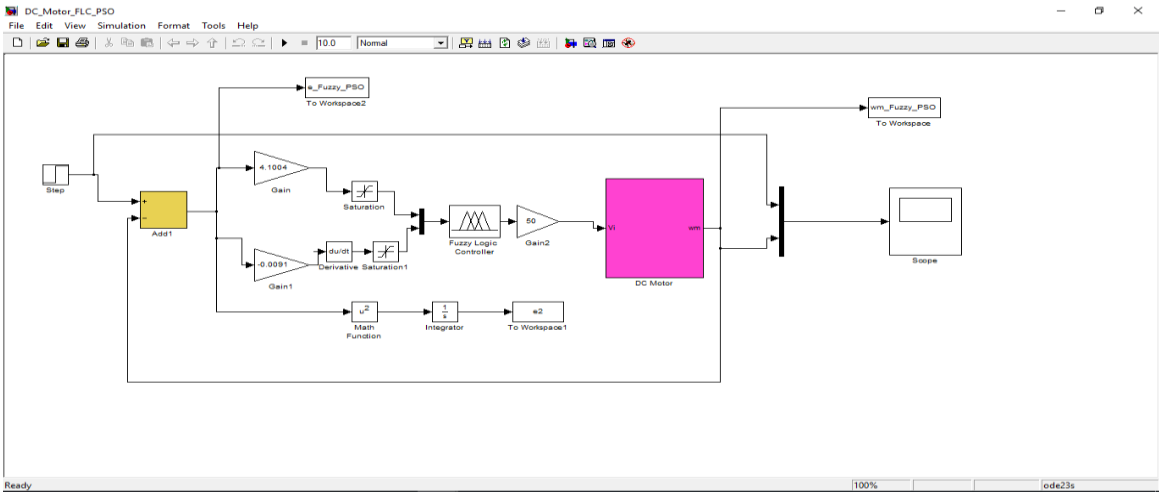

Fig. 5 . The design of PD(proportional derivative) with Fuzzy logic control support PSO Algorithm connected with DC Motor System.

The figure above describe the scale parameters that has been set in (K1,K2,K3) that taken from Particle Swarm Optimization during the tuning process when was the number of birds:50 and iteration number:10 so it will looking for the best 3 gains and give it to the MATLAB and when make sure that the results is the best and then these values that PSO has give it to us and after that take these values has been putted Dynamically from PSO Algorithm and insert it statically to K1,K2,K3 in this step, the PSO Algorithm Its role ends because that benefits of using it is to find the best gains and then we can put these gains in a normal case as well as the values that has been set Previously in Fuzzy Logic Control. the values of (K1,K2,K3) has been putted as the best gains that received from PSO Algorithm its as the Following:K1 = 4.1004,K2 = -0.0091,K3 = 50 The- mechanism of Particle Swarm Optimization (PSO) Algorithm work ,firstly that parameters has been set and after that to can call it, the current_position that has been putted in algorithm:17.5,the current position is the positions of birds currently and it make it as local best position.in evaluate initial population: its calculate the generations so the number of generations from 1:n so this will call the function of tuning that has been marked in MATLAB and the result of it will put it in current fitness, and the tuning the number of birds rotation and the mechanism of tuning it will call MATLAB SIMULINK and calculate the error inside this function this process that make it and generate the gains and also choose the best suitable gains and the current fitness that make it local best fitness and take the lower value in it and then will take it its mean the lower error and make loop to the function is it any one that give lower than global will take it and then will update the speed depending on the values and also update the position, the first step that will take from zero up to bird step about 50 and will call the function as the number of birds and then put these values from tuning to current fitness and then comparing if the current fitness less than local after replace the local by current fitness position values to save the best gains that comes from PSO Algorithm and then take minimum of minimum and make it as current global best fitness, and then if there is the lower values in current global best fitness will take it and if the lower values is in global best fitness then will take it instead of current global best fitness values and modify in the velocity on the best case and this will be on the global and also local and take the best among them and then save the best of global and show it in the gains values.

VI. Simulation Parameters and Results

The figure below describe the final figures, and it’s the comparison in angular speed between Proportional Derivative (PD) controller and when Proportional Derivative (PD) controller used the fuzzy logic control and finally is the best than them is Proportional Derivative (PD) controller used the fuzzy logic control by using particle swarm optimization (PSO) Algorithm.

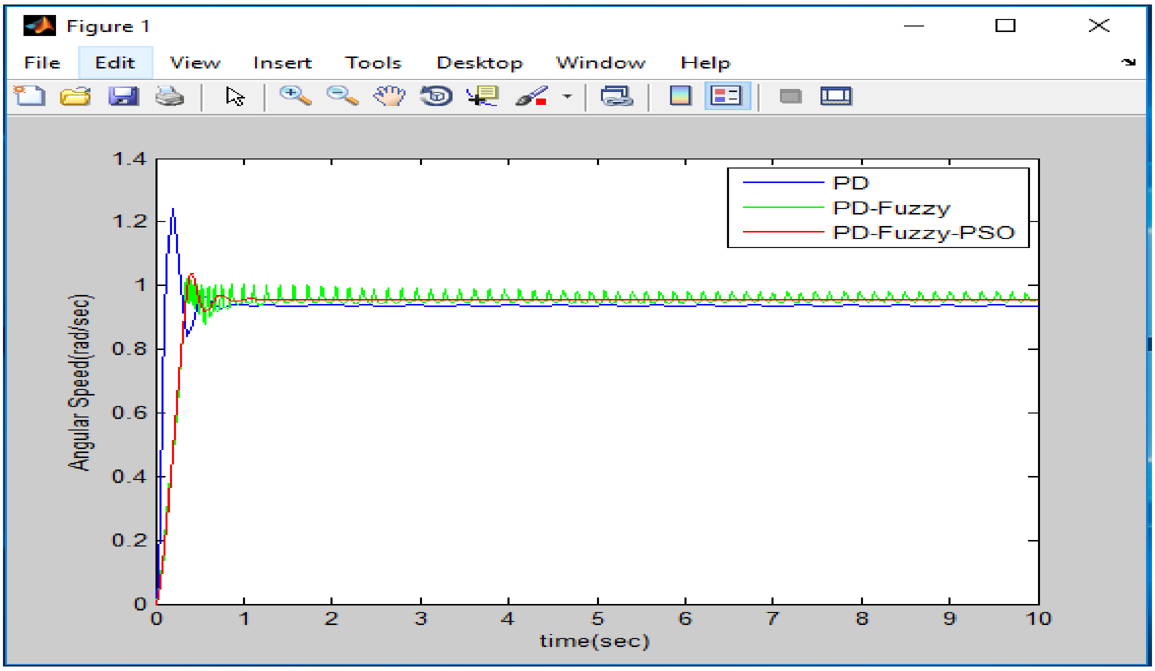

Fig. 6 .the comparison on Angular speed between PD(Proportional Derivative)and PD(Proportional Derivative) with Fuzzy logic control and PD(Proportional Derivative) with Fuzzy logic control using PSO(Particle Swarm Optimization).

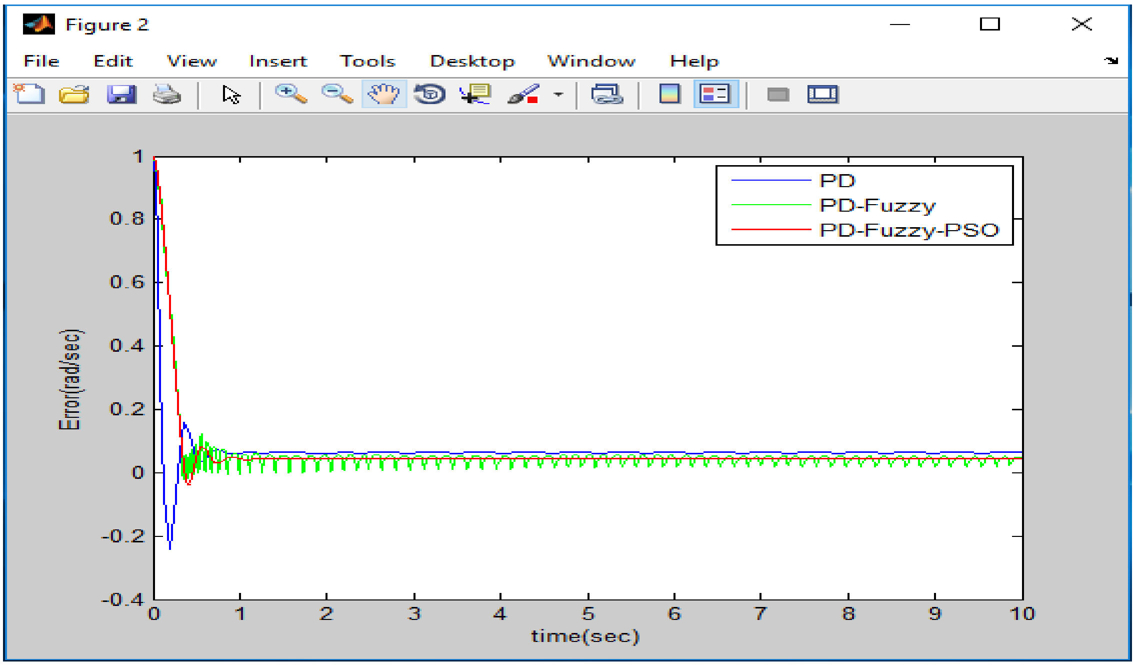

Fig. 7.the comparison on Error between PD(Proportional Derivative)and PD(Proportional Derivative) with Fuzzy logic control and PD(Proportional Derivative) with Fuzzy logic control using PSO(Particle Swarm Optimization).

The figure above describe the comparison between PD and PD-Fuzzy and PD-FUZZY-PSO when we used the particle swarm optimization (PSO) Algorithm this algorithm that improved the system of DC Motor so the Red curve in the curve is the best than them because there is no overshoot and also its stable than the others and when the Proportional derivative used the fuzzy logic control we received the good results also but the curve is not stable because of normalization so every signal that input to the fuzzy its should to input to the rules and the mechanism of rules is normalization so the result its also best than PD results but its not stable like it, the improvement of the system comes with PSO because it will looking for the best 50 birds by 10 iteration to get the best 3 scale factors to be used instead of the values that has been putted in the fuzzy logic control so the

algorithm give us the best gains for each execution operation so in PSO, we get the best high angular speed and the best low error so we improve the system by PSO Algorithm as shown in both curves the time that PSO settled in 1 (sec).
In the table below describe the comparison between PD and PD-FUZZY and PD-FUZZY-PSO Algorithm.

Table. 1. The final Results between PD and PD-FUZZY and PD-FUZZY-PSO

Control Method	**PD**	**PD-FUZZY**	**PD-FUZZY-PSO**
Overshoot %	32.54 %	4.07272727%	NO-Overshoot
Steady State Error	0.0622	0.05	0.0435
Rising Time	0.112 (sec)	1.2641 (sec)	0.326 (sec)
Settling Time	2.5 (sec)	0.72 (sec)	1.05 (sec)

VII. Conclusion

The proposed system that we have create it is DC Motor system and this system has created by using MATLAB SIMULINK and we used the basic tools and also calculate a math model to create it and also the laplace transform we used it to convert all the equations to blocks because MATLAB SIMULINK not accept equations that not deals with laplace transform and the Math model its similar to the model that used in popular papers and we based on Proportional Derivative (PD) controller by scale factor values for (Kp=150 and Kd=0.01)and these values has been checked and putted in the scale factors this depending on previous studying so we get a high overshoot for PD controller (32.54 %) with DC Motor System and high steady state error (0.0622) and this results considered as drawback because of its weak so we have improve it by two methods firstly by use the fuzzy logic control with PD controller and get a low overshoot (4.07272727%) and low steady state error (0.05) but in use PSO with Fuzzy logic control and PD controller so we get (No-Overshoot) in use PSO and a low steady state error (0.0435) and this considered as a good result, and now we have proved that PSO with fuzzy logic control based on Proportional derivative (PD) controller is better than use PD or PD with Fuzzy. In future work, I would recommend any one want to completing on my paper is to use another optimization method and apply it on DC Motor with fuzzy logic control, such as use SSO Algorithm (Sperm Swarm Optimization) or CSO (Cat Swarm Optimization) or GSO (Glowworm Swarm Optimization) and other optimization methods ,I think these Optimization methods it will give the best results than PSO (Particle Swarm Optimization) when Applied in Fuzzy logic control based on PD Controller.

References

[1] F. Salas, M. Llama, and V. Santibanez, “A stable self-organizing fuzzy PD control for robot manipulators,” *Int. J. Innov. Comput. Inf. Control*, vol. 9, no. 5, pp. 2065–5086, 2013.

[2] D. Preeti and N. Beniwal, “Comparison of Conventional and Fuzzy P/PI/PD/PID Controller for Higher Order Non Linear Plant with High Dead Time,” *Int. J. Sci. Res.*, vol. 2, no. 8, p. 217, 2012.

[3] G. Eappen and T. Shankar, “Optimization of two area AGC based power system using PSO tuned Fuzzy PID controller and PSO trained SSSC and TCPS,” *Int. J. Eng. Technol.*, vol. 7, pp. 163–168, 2018.

[4] S. Lin and G. Wang, “Fuzzy PID control algorithm based on PSO and application in BLDC motor,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 69, no. 1, 2017.

[5] H. Kala, D. Deepakraj, P. Gopalakrishnan, P. Vengadesan, and M. Karumbal Iyyar, “Performance Evaluation of Fuzzy Logic and PID Controller for Liquid Level Process,” *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 2, no. 3, pp. 2321–2004, 2014.

[6] N. Bouarroudj, D. Boukhetala, and F. Boudjema, “A hybrid fuzzy fractional order PID sliding-mode controller design using PSO algorithm for interconnected nonlinear systems,” *Control Eng. Appl. Informatics*, vol. 17, no. 1, pp. 41–51, 2015.

[7] R. Rahmani, M. S. Mahmodian, S. Mekhilef, and A. A. Shojaei, “Fuzzy logic controller optimized by particle

swarm optimization for DC motor speed control," *SCOReD 2012 - 2012 IEEE Student Conf. Res. Dev.*, pp. 109–113, 2012.

[8] B. Allaoua, A. Laoufi, B. Gasbaoui, and A. Nasri, "Intelligent Controller Design for DC Motor Speed Control based on Fuzzy Logic-Genetic Algorithms Optimization," *Sci. York*, no. 13, pp. 90–102, 2008.

[9] K. W. Yu and J. H. Hsu, "Fuzzy gain scheduling PID control design based on particle swarm optimization method," *Second Int. Conf. Innov. Comput. Inf. Control. ICICIC 2007*, pp. 2007–2010, 2008.

[10] A. Kumar and S. Suhag, "Multiverse optimized fuzzy-PID controller with a derivative filter for load frequency control of multisource hydrothermal power system," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 25, no. 5, pp. 4187–4199, 2017.

[11] A. Nabi, "Design of Fuzzy Logic PD Controller for a Position Control System," *Int. J. Eng. Manag. Res.*, vol. 3, no. 2, pp. 31–34, 2013.

[12] Y. Al-Dunainawi and M. F. Abbod, "PSO-PD fuzzy control of distillation column," *IntelliSys 2015 - Proc. 2015 SAI Intell. Syst. Conf.*, pp. 554–558, 2015.

[13] I. Pan and S. Das, "Fractional order fuzzy control of hybrid power system with renewable generation using chaotic PSO," *ISA Trans.*, vol. 62, pp. 19–29, 2016.

[14] B. Akbiyik, I. Eksin, M. Guzelkaya, and E. Yesil, "Evaluation of the Performance of Various Fuzzy PID Controller Structures on Benchmark Systems," *4th Int. Conf. Electr. Electron. Eng.*, no. 1, 2005.