

## Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm

Sheroz Khan, Salami Femi Abdulazeez, Lawal Wahab Adetunji, AHM Zahirul Alam, Momoh Jimoh E. Salami, Shihab Ahmed Hameed, Aisha Hasan Abdalla and Mohd Rafiqul Islam  
Department of Electrical and Computer Engineering, Faculty of Engineering,  
International Islamic University Malaysia, P.O. Box 10, Kuala Lumpur, 50728, Malaysia

---

**Abstract:** All control systems suffer from problems related to undesirable overshoot, longer settling times and vibrations while going from one state to another state. Most of relevant techniques had been in the form of suggesting modification and improvement in the instrumentation or interfacing part of the control system and the results reported, remain suffering from shortcomings related to hardware parameter dependence and maintenance and operational complexities. Present study was based on a software approach which was focusing on an algorithmic approach for programming a PIC16F877A microcontroller, for eliminating altogether the parametric dependence issues while adding the benefits of easier modification to suit a given control system to changing operational conditions. Said approach was first simulated using MATLAB/SIMULINK using the techniques of Proportional Derivative Fuzzy Logic Controller (PD-FLC) whose membership function, fuzzy logic rules and scaling gains were optimized by the genetic algorithm technique. Simulated results were verified by programming the PIC16F877A microcontroller with the algorithm and using it on a temperature control system where a fan was regulated in response to variations in the ambient system temperature. Resulting tabulated performance indices showed a considerable improvement in rising and settling time besides reducing overshoot and steady state error.

**Key words:** Control, fuzzy logic, genetic algorithm, microcontroller, fuzzy logic control, piece-wise linear analog-to-digital converter

---

### INTRODUCTION

Over the years, control of processes and systems in the industry is customarily done by experts through the conventional PID control techniques. This is as a result of its simplicity, low cost design and robust performance in a wide range of operating conditions<sup>[1]</sup>. Although the PID controllers have gained widespread usage across technological industries, it must also be pointed out that the unnecessary mathematical rigorosity, preciseness and accuracy involved with the design of the controllers have been a major drawback. This has made it difficult if not impossible for designers, engineers and technology experts to design intelligent complex systems, nonlinear systems higher order and time-delayed linear systems that can satisfactorily behave as expected while operating in the human-machine interface<sup>[2]</sup>. However, various techniques and modifications to the conventional PID controllers have been employed in order to overcome

these difficulties, this include the use of autotuning PID controllers, adaptive PID controllers and also the implementation of compensation schemes to the PID controllers<sup>[2,3]</sup>. Another alternative technique is the use of unconventional control techniques such as fuzzy logic, neural networks and genetic algorithm or a combination of two or more of these techniques<sup>[1]</sup>. The combination of the genetic algorithm and the fuzzy logic control technique will be considered in this study.

Genetic algorithms, which are adopted from the principle of biological evolution, are efficient search techniques that manipulate the codings representing a parameter set to reach a near optimal solution<sup>[4,6]</sup>. Hence by strengthening fuzzy logic controllers with genetic algorithms the searching and attainment of optimal fuzzy logic rules and high-performance membership functions will be easier and faster<sup>[4,5]</sup>. Although the benefits of harnessing the capabilities of genetic algorithms are huge, research efforts on optimizing fuzzy logic rules, membership functions and

---

**Corresponding Author:** Sheraz Khan, Department of Electrical and Computer Engineering, Faculty of Engineering, International Islamic University Malaysia, P.O. Box 10, Kuala Lumpur, 50728, Malaysia

other parameters are challenging. In<sup>[4]</sup>, genetic algorithm technique has been used to optimized fuzzy logic rules while in<sup>[7]</sup>, a customized GA technique has been proposed to optimize the search for optimal fuzzy logic rules. Research efforts focused majorly on the optimal tuning of membership functions in<sup>[6]</sup> while in<sup>[5,12]</sup> both fuzzy logic rules, membership functions and other parameters were optimized using genetic algorithm. The combination of genetic algorithm and fuzzy logic controllers is normally shortened as GA-FLC and this intelligent hybrid controller has found application in many scenarios like motor speed control<sup>[9,10]</sup>, temperature control<sup>[11]</sup> robotics<sup>[8]</sup> and in many other control systems.

This study employs the fuzzy logic technique to design a Proportional-Derivative (PD) Controller and optimizes the inference rules, membership functions and scaling gains of this controller by using Genetic Algorithm (GA). The resultant optimal fuzzy logic controller is used in the temperature control unit which comprises of a PIC16F877A microcontroller and other hardware components. The performance of the GA-Optimized Fuzzy Logic controller is compared with that of the conventional PID controller. The MATLAB/SIMULINK software forms part of the modeling and design tools employed in this research.

**MATERIALS AND METHODS**

**Conventional PID controller:** The transfer function of the DC motor is used in the modeling of the PID controller. The overall system model is represented in form of a block diagram in Fig. 1.

The MATLAB Simulink software is used in the analysis of the controller by studying the response generated from the modeling and simulation of the controller.

The Simulink model of the PID controller is shown in Fig. 2. The difference between the desired speed and the output feedback is passed as input into the PID controller subsystem. The PID Controller subsystem (Fig. 3) contains the proportional gain scaling factor ( $K_p$ ), the derivative gain scaling factor ( $K_d$ ) and the integral gain scaling factor ( $K_i$ ).

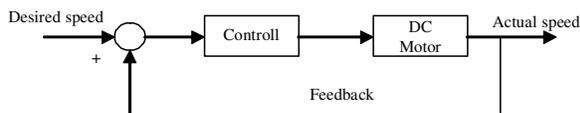


Fig. 1: System model for motor speed control

The derivative gain factor and the integral gain factor are both passed through a derivative block and an integral block respectively before being summed up with the proportional gain factor. The output of the PID controller subsystem serves as an input to the DC motor block. The DC motor block is simply the transfer function of the DC motor. The speed of the motor is also monitored from the scope connected to the output of the DC motor block. The desired speed of the motor is set to a unit constant value to serve as a step response input to the system. The error signal of the system is also monitored using a scope.

The PID controller model is hand-tuned by first increasing the value of the proportional gain,  $K_p$ , until the a desirable response is obtained. The derivative gain  $K_d$  and the integral gain  $K_i$ , are then adjusted to improve and optimize the response of the system. A fairly optimal response is achieved for a proportional gain value of 50, a derivative gain value of 60 and an integral gain of 5. The output response of the controller is shown in Fig. 4.

The performance indicators obtained are shown in Table 1.

**Fuzzy logic controller:** The Simulink model of the fuzzy logic proportional-derivative controller is shown in Fig. 5.

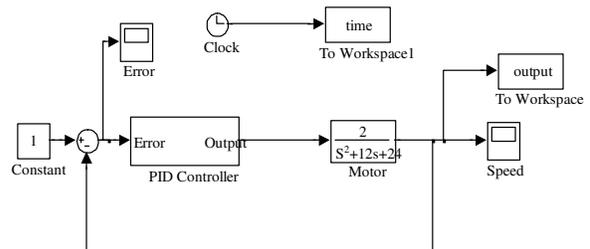


Fig. 2: Simulink model of the conventional PID controller

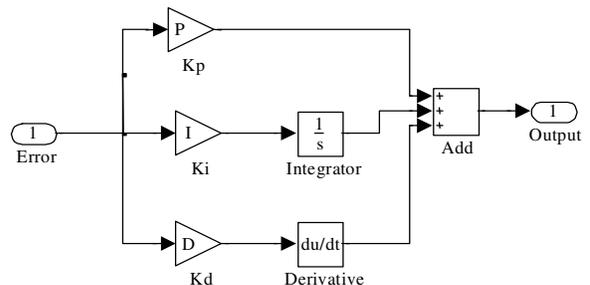


Fig. 3: PID controller subsystem

Metrics	Value
Rise time	0.3741
Settling time	0.6358
Overshoot	0.6748
Peak	1.0067

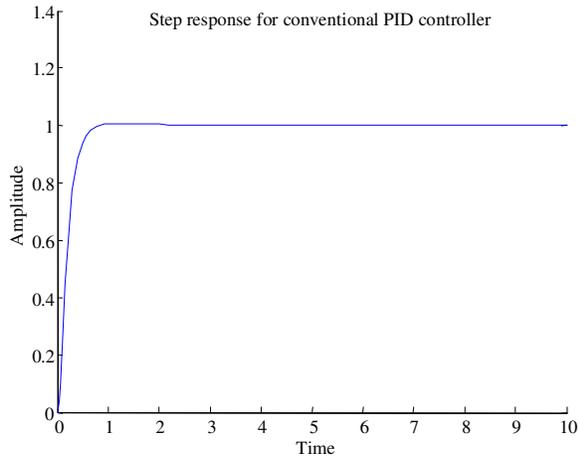


Fig. 4: Output response of the conventional PID controller

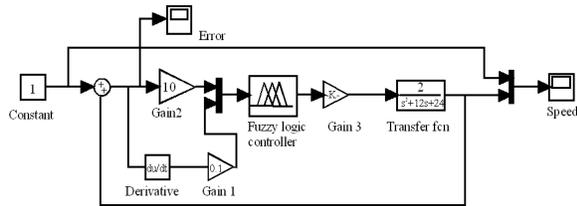


Fig. 5: Simulink model for fuzzy logic PD controller

The model consists mainly of the fuzzy logic controller block and the DC motor block. The fuzzy logic controller block contains a reference to a fuzzy logic inference system. The inference system has three linguistic variables which are the two inputs (error signal and error derivative) and the output (control signal). The fuzzy logic inference system for the fuzzy proportional-derivative controller contains a set of fuzzy logic rules that define the behavior of the system in relation between the error signal, error derivative signal and the control signal of the controller. The first input to the fuzzy logic inference system is the error signal which is the difference between the desired speed and the actual speed of the motor. The error derivative signal is achieved by differentiating the error signal before passing it to the fuzzy logic controller block. Since the fuzzy logic controller block expects two inputs, a multiplexer is used to combine the error signal and the error derivative signals as input into the block.

De	Error		
	N	Z	P
N	RB	RM	NC
Z	RM	NC	IM
P	NC	IM	IB

Metrics	Value
Rise time	0.7094
Settling time	0.8735
Overshoot	0.0000
Peak	0.9973

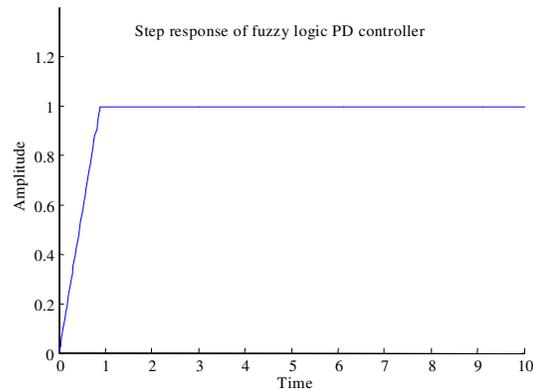


Fig. 6: Output response of for the fuzzy PD controller

The control signal output of the fuzzy logic controller is passed as the input of the DC motor block to determine the speed of the motor. The output of the DC motor block which is the motor speed is monitored using a scope to examine its response. The desired speed of the motor is set to a unit constant value to serve a step response input to the system. The error signal of the system is also monitored using a scope. The tuning of the fuzzy logic controller can be achieved by either adjusting the range of the universe of discourse for the linguistic variables, adjusting the input and output scaling gains of the controller or adjusting the number, type and positions of the membership functions used.

The rules for the fuzzy inference system are shown in Table 2.

Figure 6 shows the system output response after considerable hand-tuning of the fuzzy logic PD-controller.

The performance indicators obtained from the time response plot of the Fuzzy Logic PD Controller are shown in Table 3.

**Optimization of controller:** In essence, the beauty of this study lies predominantly in the use of an effective,

eminent and fast-spreading optimization technique in the optimization of the multivariable PD fuzzy logic controller developed earlier. The Genetic Algorithm (GA) is employed to perform a comprehensive and complete search in finding an optimal set of solution for the fuzzy logic rules, membership functions and scaling gains for the specified fuzzy logic controller.

The proportional-derivative fuzzy logic controller (PD-FLC) for a motor speed is re-implemented but the only difference this time is that is optimized. The fuzzy inference system consists of three linguistic variables (two inputs and one output) each having seven membership function sets. This results in 49-rule fuzzy inference system with inputs as the error and the rate of change in error. The output of the fuzzy logic inference system is the control action of the controller and the universe of discourse of all the variables are set within the range (-1, 1).

**GA optimization of the fuzzy logic controller:** The genetic algorithm technique employed in this study is used to tune the fuzzy logic controller based on the method described in<sup>[13]</sup>. The tuning approach employs the use of MATLAB M-files and functions to manipulate the fuzzy inference system and scaling gains, run the Simulink based simulation, check the resulting performance and continuously modify the fuzzy inference system for a number of times in search for an optimal solution. The integral of absolute error ( $IAE = \int_0^{\infty} |e(t)| dt$ ) is used as a measure of the system performance since it is known to give a better all round performance indicator of a control system response where overshoot, settling time and rise time are the main considerations<sup>[14]</sup>. The GA-optimization algorithm was run for 300 generations with each generation having a population size of 30. The parameters of the fuzzy logic controller to be tuned are encoded into chromosomes of 51 bits. The bit distribution of the chromosomes is shown in Table 4.

MATLAB M-files were utilized for the encoding, testing and decoding of each of the tuned FLC parameters. This includes the fuzzy logic rule base, the membership function definition of the linguistic variables and the scaling gains of the controller. The output scaling gain of the controller was adjusted over a range of 0-250 while the proportional input scaling gain and the derivative input scaling gains were adjusted over a range of 0-50 and 0-1.5 respectively. The suitability of the ranges of the scaling gains was determined from the prior hand tuning of the controller.

Table 4: Chromosome Distribution

Rule base	MF for proportional input	MF for derivative input	MF for output	Proportional scaling gain	Derivative scaling gain	Output scaling gain
9 bits	7 bits	7 bits	7 bits	7 bits	7 bits	7 bits

The modified Simulink model of the system showing the value of the IAE is shown in Fig. 7.

**Optimization results:** The resulting optimized scaling gains are listed:

- Proportional Error input scale: 16.1417
- Error Derivative input scale: 0.31251
- Output Scale: 155.5118

The GA-optimized membership functions for the three linguistic variables are shown in Fig. 8-10.

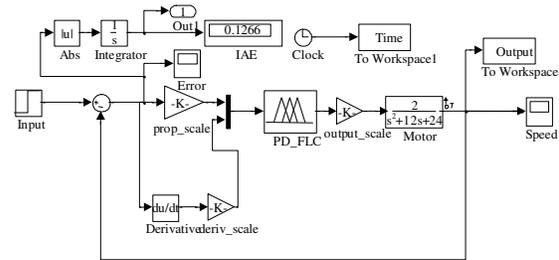


Fig. 7: Modified simulink model for PD-FLC with the inclusion of IAE

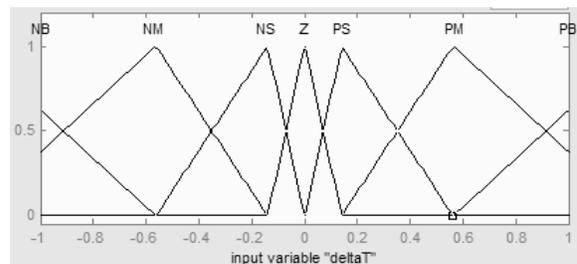


Fig. 8: Modified Membership plot Function for input variable deltaT (error)

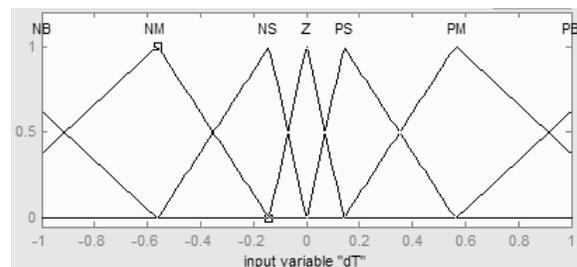


Fig. 9: Modified Membership Function plot for input variable dT (rate of change in error)

The GA-optimized fuzzy logic rules for the fuzzy inference system are shown in Table 5.

The step response of the optimized fuzzy logic PD controller is also shown in Fig. 11 and the performance metrics indicators obtained for the optimized fuzzy logic controller are shown in Table 6.

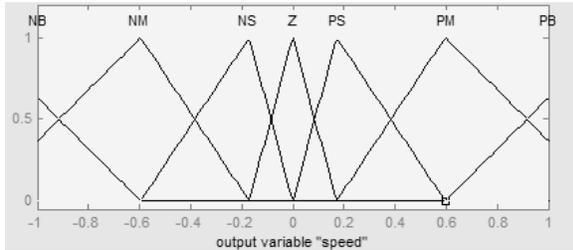


Fig. 10: Modified Membership Function plot for output variable speed (control action)

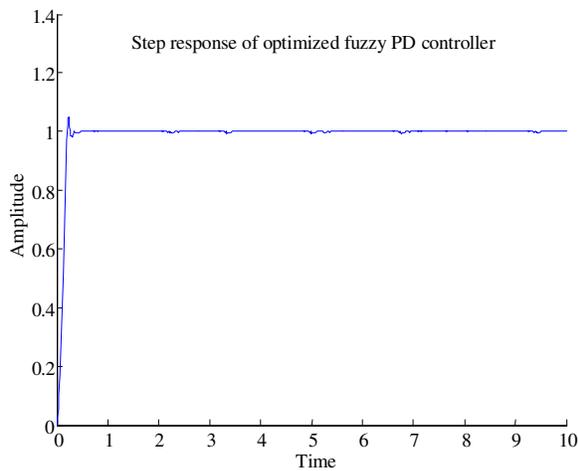


Fig. 11: Step response of optimized fuzzy logic PD controller

Table 5: Optimized fuzzy logic rules

De	Error						
	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NS	Z	PS
NM	NB	NB	NB	NM	Z	PS	PM
NS	NB	NB	NM	NS	Z	PM	PB
Z	NB	NM	NS	Z	PS	PM	PB
PS	NB	NM	Z	PS	PM	PB	PB
PM	NM	NS	Z	PM	PB	PB	PB
PB	NS	Z	PS	PM	PB	PB	PB

Table 6: Performance metrics for optimized fuzzy logic controller

Metric	Rise time	Settling time	Overshoot	Final value
Value	0.1559	0.2526	4.8434	1.00

RESULTS AND DISCUSSION

The performance metrics of the conventional PID controller, fuzzy logic PD controller and GA-optimized fuzzy logic PD controller obtained from the simulation of the motor speed control is shown in Table 7. A comparison of the step response of the controllers is also shown in Fig. 12.

The most desirable performance requires the controllers to have the smallest possible value for the rise time, overshoot and the settling time. It is also required for the final value should be as close as possible to the desired value which is unity.

From the table, it can be seen that the fuzzy logic controller can produce a desirable response performance with the use of only the proportional and Derivative Component (PD). This is contrary to the implementation of the conventional controller which requires the use of the proportional, derivative and integral component before a desirable or satisfactory response can be obtained. When compared to the conventional PID controller, the fuzzy logic PD controller shows a better performance in terms of overshoot while it exhibits a slightly lesser performance in terms of rise time and settling time. However, the results are changed when the fuzzy logic PD controller is further optimized.

Table 7: Performance metrics for conventional and fuzzy logic controller

	Conventional PID controller	Fuzzy logic PD controller	Optimized fuzzy logic PD controller
Rise time	0.3741	0.7094	0.1559
Overshoot	0.6748	0.0000	4.8434
Settling time	0.6358	0.8735	0.2526
Final value	1.0000	0.9958	1.0000

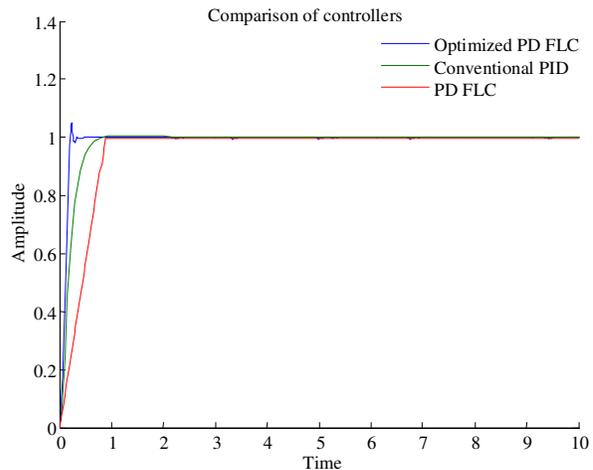


Fig. 12: Output response comparison of controllers

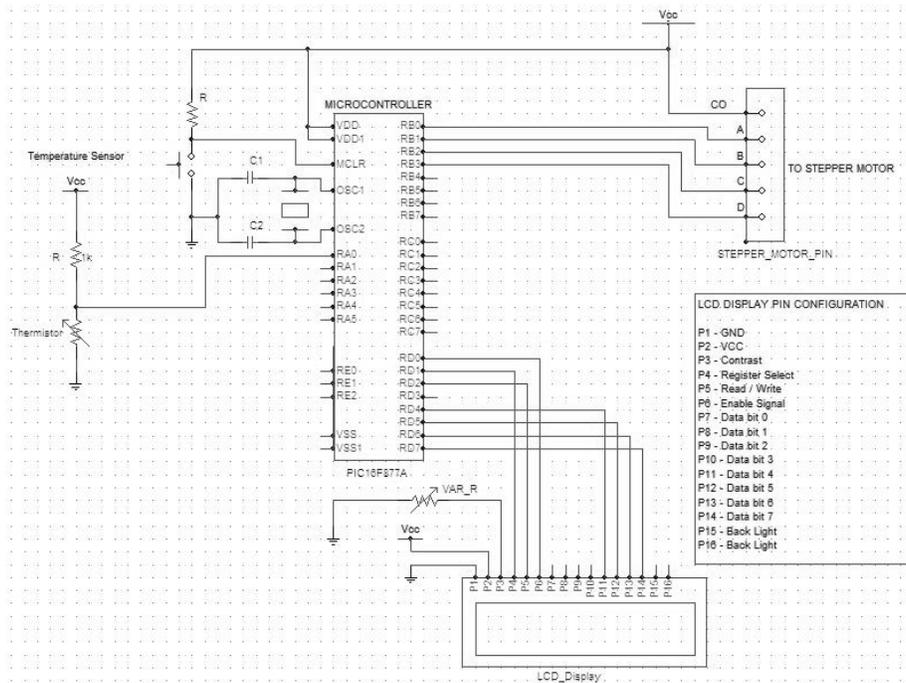


Fig. 13: Schematic design of the temperature control system

The result of the GA-optimized fuzzy logic PD controller shows an outstanding performance in terms of achieving the desired value with very small values for the rise time and settling time. Although the response shows the presence of a little overshoot, this overshoot can be ignored considering the highly satisfactory values for the other performance metrics. By comparing the results of the other performance metrics in Table 7, it can be noted that the GA-optimized fuzzy logic PD controller produces a more desirable performance when compared to the conventional PID controller. At this point, it can be said that with optimal tuning measures, the fuzzy logic PD controller can perform better than the conventional PID controller.

**Hardware design and implementation:** In order to further analyze and verify the performance of the optimal fuzzy logic controller, a temperature control system was designed to utilize the optimal fuzzy logic controller. The aim of the temperature control system is to control the temperature of an environment by regulating the speed of a fan driven by a stepper motor. The PIC16F877A microcontroller is used to operate the temperature control unit based on the appropriate decisions made by the optimal fuzzy logic controller algorithm programmed into the microcontroller with reference to the temperature sensor input. Figure 13 and 14 show the schematic design and pictorial representation of the temperature control system.

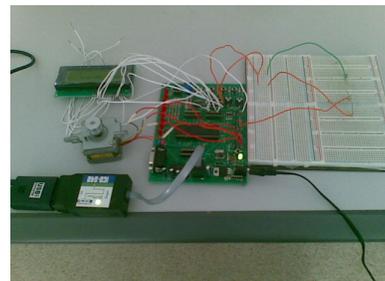


Fig. 14: Pictorial representation of the temperature control system

Since the proper decision making process of the fuzzy logic algorithm relies on the readings from the temperature sensor input, it is important to ensure that the temperature sensor input is accurately measured given the non-linear characteristics of a thermistor temperature sensor. The non-linearity issue of the temperature sensor was taken into consideration by employing the Piecewise Linear Analog-to-Digital Conversion (PLADC) technique to ensure that the temperature readings are measured accurately<sup>[14]</sup>. The results obtained from the linearization of the temperature sensor is shown in Fig. 15. Figure 15 shows that a high degree of accuracy is achieved between the linearized temperature characteristic curve and ideal temperature characteristic curve with a maximum percentage error of about 5%.

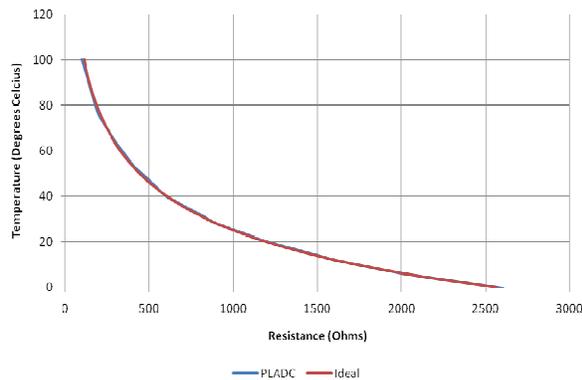


Fig. 15: Comparison of ideal and linearized temperature sensor characteristics

The performance of the temperature control system shows that the optimal fuzzy logic controller algorithm succeeds in smoothly carrying out the temperature control action as well as achieving the required setpoint accurately within a limited time.

### CONCLUSION

This study has succeeded in the design of an optimal Proportional Derivative fuzzy logic controller (PD-FLC) using genetic algorithm technique and implementing it in the temperature control system using PIC16F877A microcontroller and other hardware components. It was also shown through simulation that the optimal fuzzy logic controller is performing better than a conventional PID controller when both controllers are subjected to the same operating conditions. The performance metrics taken into consideration are the overshoot, rise time, settling time and steady state error. MATLAB/SIMULINK is used to simulate and fine-tune the controller models. The simulation results and the performance of the hardware implementation show that the optimal fuzzy logic controller is functioning better than a conventional PID controller in terms of the rise and settling time.

### REFERENCES

1. Kumar, M. and D.P. Garg, 2004. Intelligent learning of fuzzy logic controllers via neural network and genetic algorithm. Proceedings of the Japan-USA Symposium on Flexible Automation, July 19-21, Colorado, pp: 1-8. [http://www.duke.edu/~manish/UL\\_029.pdf](http://www.duke.edu/~manish/UL_029.pdf).
2. Tang, K.S., K.F. Man, G. Chen and S. Kwong, 2001. An optimal fuzzy PID controller. IEEE Trans. Ind. Elect., 48: 757-765. DOI: 10.1109/41.937407.

3. Lu, Y.S. and C.M. Cheng, 2005. Design of a non-overshooting PID controller with an integral sliding perturbation observer for motor positioning systems. JSME Int. J. Series C., 48: 103-110. [http://www.jstage.jst.go.jp/article/jsmec/48/1/48\\_103/\\_article](http://www.jstage.jst.go.jp/article/jsmec/48/1/48_103/_article).
4. Hwang, W.R. and W.E. Thompson, 1994. Design of intelligent fuzzy logic controllers using genetic algorithms. Proceedings of the 3rd IEEE Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, June 26-29, IEEE Computer Society, Washington DC., USA., pp: 1383-1388. DOI: 10.1109/FUZZY.1994.343566.
5. Ng, K.C. and Y. Li, 1994. Design of sophisticated fuzzy logic controllers using genetic algorithms. Proceedings of the 3rd IEEE Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, June 26-29, IEEE Computer Society, Washington DC., USA., pp: 1708-1712. DOI: 10.1109/FUZZY.1994.343598.
6. Mohammadian, M. and R.J. Stonier, 1994. Tuning and optimisation of membership functions of fuzzy logic controllers by genetic algorithms. Proceedings of the 3rd IEEE International Workshop on Robots and Human Communication, July 18-20, IEEE Computer Society, USA., pp: 356-361. DOI: 10.1109/ROMAN.1994.365903.
7. Chan, P.T., A.B. Rad and K.M. Tsang, 1997. An optimised fuzzy logic controller. Proceedings of the 6th IEEE International Conference on Fuzzy Systems, July 1-5, IEEE Computer Society, USA., pp: 975-980. DOI: 10.1109/FUZZY.1997.622841.
8. Rekik, C., M. Djemel, N. Derbel and A. Alimi, 2002. Design of optimal fuzzy logic controller with genetic algorithms. Proceedings of the IEEE International Symposium on Intelligent Control, Oct. 27-30, IEEE Computer Society, Washington DC., USA., pp: 98-103. DOI: 10.1109/ISIC.2002.1157745.
9. Xiu, J., C. Xia and H. Fang, 2006. GA-based adaptive fuzzy logic controller for switched reluctance motor drive. Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21-23, IEEE Computer Society, Washington DC., USA., pp: 8226-8230. DOI: 10.1109/WCICA.2006.1713578.
10. Oh, W.S., Y.T. Kim, C.S. Kim, T.S. Kwon and H.J. Kim, 1999. Speed control of induction motor using genetic algorithm based fuzzy controller. Proceedings of the 25th Annual Conference of the IEEE Industrial Electronics Society, Nov. 29-Dec. 3, IEEE Computer Society, Washington DC., USA., pp: 625-629. DOI: 10.1109/IECON.1999.816464.

11. Hwang, M.W. and J.Y. Choi, 1997. Hybrid feedforward and feedback control of wafer temperature in RTP using genetic algorithm and fuzzy logic. Proceedings of the IEEE International Conference on Intelligent Processing Systems, Oct. 28-31, IEEE Computer Society, Washington DC., USA., pp: 93-97, DOI: 10.1109/ICIPS.1997.672745.
12. Ko, C.N., T.L. Lee, Y.Y. Fu and C.J. Wu, 2006. Simultaneous auto-tuning of membership functions and fuzzy control rules using genetic algorithms. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 8-11, IEEE Computer Society, USA., pp: 1102-1107, DOI: 10.1109/ICSMC.2006.384547.
13. Byrne, J.P., 2003. GA-optimization of a fuzzy logic controller. Masters Thesis, School of Electronic Engineering, Dublin City University.
14. Bucci, G., M. Faccio and C. Landi, 2000. New ADC with piecewise linear characteristic: Case study-implementation of a smart humidity sensor. IEEE Trans. Instrument. Measure., 49: 1154-1166. DOI: 10.1109/19.893250.