# Comparison of Fuzzy Control and PID Control in Embedded Systems

A. Nileshwar[1], K. Mhatre[2], N. Kuchroo[3], O. Mansata[4] and Y. S. Rao[5]

*Abstract*---**The primary objective of this work is to discuss the Fuzzy control and PID control approaches followed by a comparative study of the two methods. The first section of this paper explains the intricacies of the Fuzzy set theory and how it is applied for Fuzzy control systems. Further, it explains the PID approach and the issues involved in developing and maintaining a PID control system. The paper outlines the simulation results of the Fuzzy based temperature control system and the PID based temperature control system. The conclusions and the comparisons based on the simulations are presented. Finally, the issues related to developing a real time embedded control system are discussed.**

## I. FUZZY LOGIC

THE term “fuzzy set” was first coined in 1965 when professor Lotfi A. Zadeh from the University of Berkeley, USA, published a paper entitled “Fuzzy sets”. His achievements in this field have been quickly coupled by breakthroughs of numerous research workers developing theoretical works.

### A. Initial Applications

In 1975, professor Mamdani from London developed a process control technique for the control of a steam motor. In 1978, the Danish company, F.L. Smidth, achieved the control of a cement kiln. This was the first authentic industrial application of fuzzy logic. [1]

### B. Acceptance

Fuzzy logic gained momentum in Japan where research was not only theoretical but also highly application oriented. At the end of the eighties, fuzzy logic had taken off in a big way, and consumer products such as washing machines, cameras and camcorders with the mention “fuzzy logic” were too numerous to be counted. Industrial applications such as treatment of water, harbour container cranes, undergrounds and ventilation/air conditioning systems began to use fuzzy logic too. Finally, applications developed in such other fields such as finance and medical diagnosis. From 1990 onwards, many applications began to emerge in large numbers in Germany, as well as, to a lesser extent, in the USA [2].

### C. Points in favour of fuzzy logic:

Fuzzy logic stems from several observations, namely:

1) Fuzzy logic uses linguistic approach thereby making it possible to analyze in an approximate manner those humanistic as well as mechanistic systems which are too complex for the application of classical techniques [3].
2) It is a model-free approach and can incorporate the process knowledge as simple IF-THEN rules.
3) It is non-linear by nature and can be designed to handle simple SISO to complex MIMO systems.
4) It is highly robust and pro-active and severely penalizes the offsets in a stronger way [4].

### D. Fuzzy Sets and Fuzzy Operators:

As per L.A.Zadeh [5], a fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership function which assigns to each object, a grade of membership ranging between zero and one. Consider the two membership functions $f_A(x)$ and $f_B(x)$ of the fuzzy sets A and B respectively.

Complement: The complement of the fuzzy set A is denoted by A’ and is defined by,

$$F_{A'} = 1 - F_A \qquad (1)$$

Union: The union of the two fuzzy sets is a fuzzy set C which is written as A U B and its membership function is related to those of A and B by,

$$F_C(x) = \mathrm{Max}\,[F_A(x), F_B(x)], \qquad x \in X \qquad (2)$$

Intersection: The intersection of the two fuzzy sets is a fuzzy set D which is written as $A \cap B$ and its membership function is related to those of A and B by,

$$F_C(x) = \mathrm{Min}\,[F_A(x), F_B(x)], \qquad x \in X \qquad (3)$$

### E. Steps to implement Fuzzy Logic:

1) *Fuzzify the Inputs:*
   The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the matter in consideration and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation.
2) *Apply the Fuzzy Operators:*
   On the fuzzification of the input variables, we have to apply the fuzzy operators. If there are two or more input variables after fuzzification we have to apply fuzzy operators to get a single output. The output is a

---

[1] Anand Nileshwar is a final year student of SP College of Engg. Mumbai, India. (e-mail: anandniles@yahoo.com)

[2] Kalpesh Mhatre is a final year student of SP College of Engg. Mumbai, India. (e-mail: kalpesh_genius@yahoo.co.in)

[3] Nikhil Kuchroo is a final year student of SP College of Engg. Mumbai, India. (e-mail: nik_ku86@yahoo.co.in)

[4] Omkar Mansata is a final year student of SP College of Engg. Mumbai, India. (e-mail: om_eagle@yahoo.com)

[5] Y. S. Rao is a Asst. Prof. in SP College of Engg. Mumbai, India. (e-mail: ysrao@ee.iitb.ac.in)

single truth value.

3) *Apply Implication Method:*
Before applying the implication method, you must determine the rule's weight. Every rule has a *weight* (a number between 0 and 1), which is applied to the number that is given by the antecedent. After proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.

4) *Aggregate All the Outputs:*
Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. As long as the aggregation method is commutative (which it always should be), then the order in which the rules are executed is unimportant.

5) *Defuzzify:*
The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. The aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

## II. PID CONTROLLER

PID controller is a one of the earliest industrial controllers [6]. PID controller is simple and linear; it can give a good performance for stable linear processes. The error E(t) is the difference between the set point and the current value of the system parameter which is to be brought under control. PID controller consists of three terms:

### A. *Proportional (P) Controllers:*

In the proportional controller, the output is proportional to the error or a change in measurement.

$$P(t) = E(t)*(\text{Proportional Gain}) \qquad (4)$$

### B. *Derivative (D) Controllers:*

In the derivative controller, the output is proportional to the rate of change of the measurement or error. Derivative action speeds up the response. The output is calculated by the rate of change of the error with time.

$$D(t) = \text{Derivative Time}*(\text{Derivative of } E(t)) \qquad (5)$$

### C. *Integral (I) Controllers:*

In an integral controller, the output is proportional to the amount of time the error is present. Integral action eliminates the steady state error.

$$I(t) = (1/\text{Integral Time})*(\text{Integral of } E(t)) \qquad (6)$$

The total controller output of the PID controller U(t) is given by,

$$U(t) = P(t) + I(t) + D(t) \qquad (7)$$

### D. *Why PID?*

The problem with using the proportional control alone for reaching the new desired outputs quickly is that, it is difficult to avoid overshoot and to eliminate the ripple once you get there. Quick response requires a high proportional gain, whereas minimizing the overshoot and the oscillation requires a small proportional gain. Achieving both at the same time may not be possible in all systems.

If the output is changing rapidly, overshoot or undershoot may lie ahead. In that case, the change suggested by the proportional controller must be reduced. The rate of change of a signal is also known as its derivative. The derivative at the current time is simply the change in value from the previous sample to the current one. This proposes the use of PD controller.

The net effect is a slower response time as compared to a proportional controller with far less overshoot and ripple than a proportional controller alone. Another problem with the PD control alone is that it will not always settle exactly at the required output. For avoiding this issue, we need to incorporate an integral term.

An integral is a sum over time, that is, the sum of all past errors in the plant output.

Even though the integral gain factor $K_i$, is small, a persistent error will in due course cause the sum to grow large. In practice, the accumulated error is usually limited at some maximum and minimum values, in order to avoid the integral wind-up [7].

Hence a PI controller is generally used in order to minimize the steady state error, when the reference signal given to the system is a step input. On the other hand if the reference signals imposed to the system are ramps or other kinds of time-functions, it's better to use a PID controller; nevertheless, in practice if PID is not properly tuned, the derivative term could amplify noise. But, this can cause oscillations or could render the system unstable.

### E. *Basic Types of PID:*

#### 1) *Analog PID:*

The equation of the analog PID can be stated as follows:

$$D(s) = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] \qquad (8)$$

This is a continuous PID equation and cannot be implemented in an embedded system.

On the other hand, the digital PID controller can be easily implemented in embedded systems. This is because both are of discrete nature and the delay blocks can be easily realized [8].

### 2) *Digital PID:*

The equation of the digital PID can be stated as follows:

$$D(z) = K_c \left[ 1 + \frac{T}{2T_i}\left(\frac{z+1}{z-1}\right) + \frac{T_d}{T}\left(\frac{z-1}{z}\right) \right] \quad (9)$$

### *F. PID Tuning:*

If the PID controller parameters (the gains of the proportional, integral and derivative terms) are not chosen correctly, the output may become unstable, that is, its output may either become saturated or mechanical breakage of the system may occur. Tuning of a control loop implies the adjustment of its control parameters to optimum values in order to achieve the required control response.

One of the popular methods for tuning the PID controller is the Ziegler-Nichols method [9].

TABLE I
PID TUNING USING ZIEGLER – NICHOLS METHOD

| Control Type | Kp | Ki | Kd |
|---|---|---|---|
| P | 0.5·Kc | - | - |
| PI | 0.45·Kc | 1.2Kp / Tc | - |
| PID | 0.6·Kc | 2Kp / Tc | Kp·Tc / 8 |

## III. SIMULATION AND RESULTS

A temperature control system was simulated using MATLAB$^{©}$ 7.0. The control system consisted of a heater, a cooler, a feedback element, a sensor, a control input and a controller. Two independent control systems were designed. The first system consisted of a Fuzzy controller whereas the second system consisted of a PID controller. The two systems were simulated independently and conclusions were drawn. The block diagram of the system is as shown:

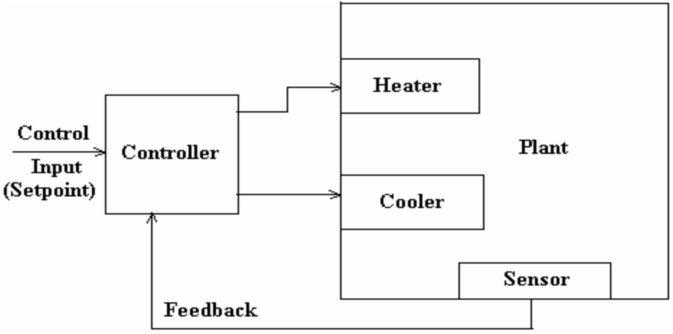


Fig. 1. The Basic Temperature Control System

### *A. Fuzzy Logic Controller:*

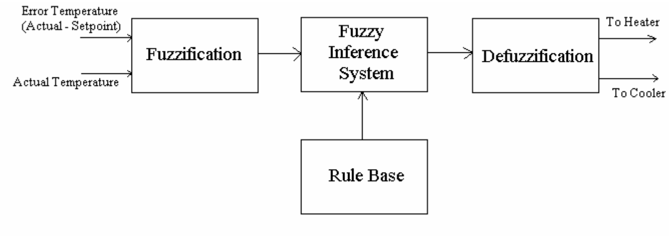


Fig. 2. The Fuzzy Logic Controller

The Fuzzy Logic Controller takes two inputs. The first input is the error temperature which is the difference between the actual temperature and the set temperature. The second input is the actual temperature of the system. These are known as the crisp inputs. These inputs are fuzzified by using their appropriate membership functions as shown in Fig.4 and Fig.5. The input variable "Error" results in three different fuzzy linguistic sets, namely, Vtoocold (negative error values), ok (nearly zero) and Vtoohot (positive error values). The other input variable "Temperature" also results in three different fuzzy linguistic sets, namely, Vcold, normal, hot. All variables used in the simulation for implementing the fuzzy logic controller are 8-bit.

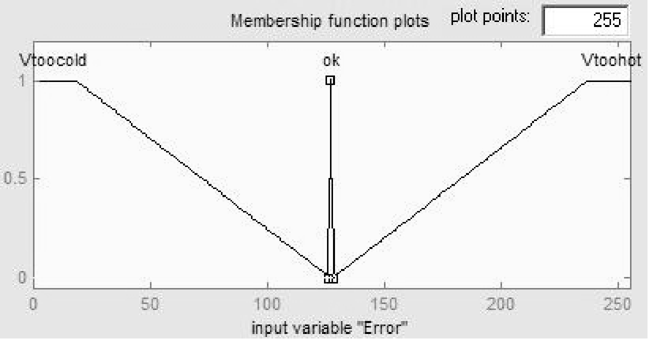


Fig. 3. Membership Function of Error Temperature

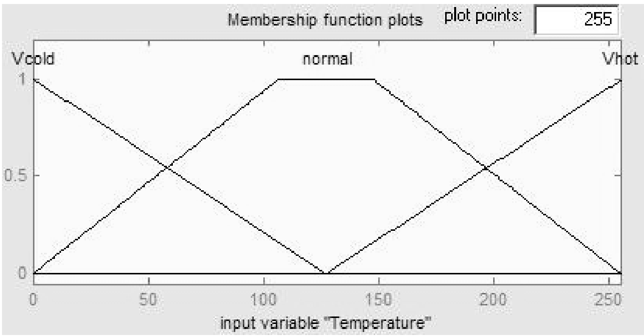


Fig. 4. Membership Function of Actual Temperature

The Fuzzy rules shown in the Table 2 are applied on the fuzzified inputs using AND operator and OR operator. Here the fuzzy inference technique is applied and the one used here is the MIN-MAX method. This results in the fuzzy output sets which represents the status of heater and cooler. The fuzzy output variable "heater" has two different fuzzy linguistic sets, namely, Heater ON (max current) and Heater OFF (no current). Similarly, the fuzzy output variable "cooler" also has two different fuzzy linguistic sets, namely, Cooler ON (max current) and Cooler OFF (no current). The last step is defuzzification that is, obtaining crisp output from fuzzy linguistic sets. Here the "weighted average" method is used for defuzzification which results in two crisp outputs, namely, heater current and cooler current. Hence, the current given to heater and cooler is controlled which finally controls the temperature of the area under consideration.

TABLE II
FUZZY RULES

| Error→<br>Temperature ↓ | Vtoocold | Ok | Vtoohot |
|---|---|---|---|
| Vcold | Heater ON<br>Cooler OFF | Heater OFF<br>Cooler OFF | NOT APPLICABLE |
| Normal | Heater ON<br>Cooler OFF | Heater OFF<br>Cooler OFF | Heater OFF<br>Cooler ON |
| Vhot | NOT APPLICABLE | Heater OFF<br>Cooler OFF | Heater OFF<br>Cooler ON |

The results of the simulation based on the Fuzzy controller used are as shown in Fig. (6)

### B. PID Controller

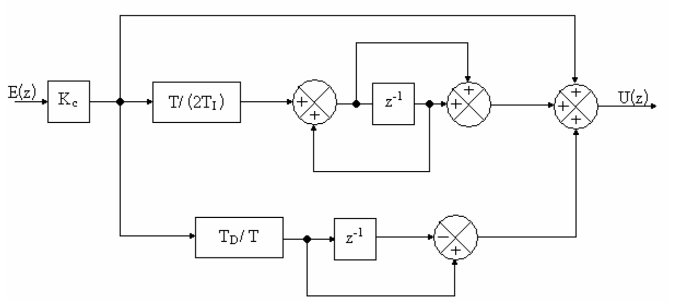


Fig. 5. Block Diagram of Digital PID Controller

The Digital PID controller as shown in the Fig. 1 was simulated. Using Ziegler-Nichols method the PID controller was tuned. Fine tuning was achieved by intuitive approach. If the output of the PID controller is positive, it implies that actual temperature is greater than the setpoint. Hence, the cooler is switched ON and the heater is turned OFF. Whereas, when the output of the PID controller is negative, it implies that the actual temperature is lower than the setpoint and as a result the cooler is switched OFF and the heater is turned ON. All variables used in the simulation for implementing the fuzzy logic controller are 32-bit. The results of the simulation based on the PID controller used are as shown in Fig. (7)

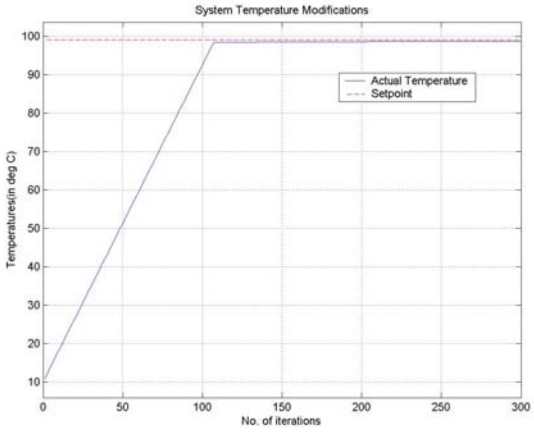


Fig. 6. Output of the Fuzzy Logic Control System. The setpoint was at 99°C while the simulation started from 11°C

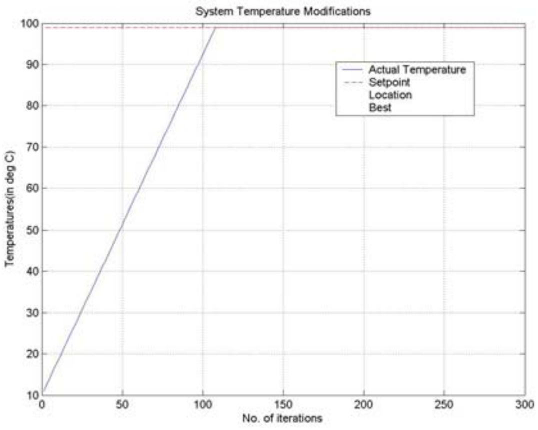


Fig. 7. Output of the PID Control System. The setpoint was at 99°C while the simulation started from 11°C

TABLE III
RESULTS BASED ON THE SIMULATION

| Method→<br>Error↓ | Fuzzy | PID |
|---|---|---|
| | No. of Iterations | |
| 0.5 | 163 | 107 |
| 1.0 | 106 | 106 |
| 5.0 | 101 | 101 |
| 10.0 | 95 | 95 |
| Final Steady State Error | 0.38 | 0 |

## IV. FURTHER DEVELOPMENTS

The advantages of Fuzzy and PID methods can be exploited and the disadvantages can be eliminated by developing a controller that incorporates both the techniques. This type of a controller is a Fuzzy-PID controller. This can be implemented by using Fuzzy Gain Scheduling as demonstrated in [10]**.** Fuzzy logic can be combined with neural networks to form a Neuro-Fuzzy system. Neuro-Fuzzy system adapts itself and learns to do better in changing environments [11].

## V. ISSUES WHILE IMPLEMENTING IN EMBEDDED SYSTEMS

The generality involved in fuzzy logic comes at a price. Since all operations involve sets, rather than numbers, the amount of calculations per inference rises dramatically as the complexity of the fuzzy knowledge base increases. Also, more rules are evoked at any given instance than traditional crisp expert systems. One approach to solve this problem of computational load has been the development of special purpose chips which perform specific variants of fuzzy inference [12].

## VI. CONCLUSIONS

Based on the simulation, we have made certain conclusions. PID systems can be implemented successfully only for single-input single-output (SISO) systems, while fuzzy systems can be implemented as multiple-input multiple-output (MIMO) systems.

While designing a Fuzzy controller, one does not need to know the precise system parameters, whereas for the PID controller we need to know the precise and complete system parameters in order to design a successful control system. In our simulations PID controller variables were 32-bit and Fuzzy controller variables were 8-bit and still the outputs of both the controllers are almost same. From this, we can conclude that an 8-bit Fuzzy controller performs as efficiently as a 32-bit PID controller. If small errors are acceptable, we observe that Fuzzy systems as well as PID systems reach their steady states in the same amount of time. PID works well only if there is an experienced operator at hand to tune it properly. Moreover, PID needs to be retuned in order to adapt itself to the changing system parameters to achieve optimum performance.

## VII. REFERENCES

[1] Jan Jantzen, Foundations of Fuzzy Logic. John Wiley and Sons, Ltd, 2007.

[2] Christopher Hale, CamQuynh Nguyen, “Voice Command Recognition Using Fuzzy Logic,” WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies', 1995.

[3] L.A.Zadeh, “A fuzzy-algorithmic approach to the definition of complex or imprecise concepts,” International Journal Man-Machine Studies, 1976

[4] R. Raji, Y.G. Srinivasa and A. Kistner, “Development of a Neuron chip based Smart Embedded Fuzzy Logic Controller with Intelligent Control strategies for Process Control,” IEEE International Fuzzy Systems Conference, 2001

[5] L. A. Zadeh, “Fuzzy Sets,” Information and Control, no.8, 1965, pp. 338-353.

[6] Jonathan W. Valvano, Embedded Microcomputer Systems: Real Time Interfacing. Brooks/Cole Thomson Learning Pte Ltd. 2001 pp. 732-767

[7] Jonas Ohr, “Anti-Windup and Control of Systems with Multiple Input Saturations: Tools, Solutions and Case Studies”, Dissertation for the Degree of Philosophy, Uppsala University, ISBN 91-506-1691-9, 2003.

[8] M. Gopal, Digital Control and State Variable Methods, 2nd ed. Tata McGraw-Hill Publishing Company, Ltd. 2003, pp. 152-176.

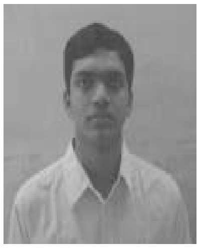
[9] K. Ogata, Modern Control Systems Engineering, 4th ed. Prentice-Hall of India Pvt. Ltd. 2002, pp. 681-705.

[10] Z.Y.Zhao, M.Tomizuka, “Fuzzy gain scheduling of PID controller,” IEEE, pp.698-703, 1992.

[11] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence. Prentice-Hall Inc., 1997, ISBN: 0-13-261066-3

[12] A. Ismael, B. Hussien, R. W. McLaren, “Fuzzy Neural Network Implementation of Self Tuning PID Control Systems,” IEEE International Symposium on 1ntelligent Control, 1994

## VIII. BIOGRAPHIES

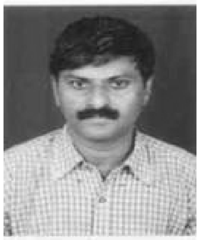


**Anand Nileshwar** is student of final year Electronics Engg., S. P. College of Engg, Mumbai. He is a member of IEEE. His areas of interest include Embedded Systems, Robotics, Digital Signal Processing and Image Processing. (anandniles@yahoo.com)

**Kalpesh Mhatre** is student of final year Electronics Engg., S. P. College of Engg, Mumbai. He is a member of IEEE. His areas of interest include Robotics, Embedded systems, Microprocessor Systems Design and Computer Programming. (kalpesh_genius@yahoo.co.in)

**Nikhil Kuchroo** is student of final year Electronics Engg., S. P. College of Engg, Mumbai. He is a member of IEEE. His areas of interest include Robotics, Embedded Systems and Microprocessor Systems Design. (nik_ku86@yahoo.co.in)

**Omkar Mansata** is student of final year Electronics Engg, S. P. College of Engg, Mumbai. He is a member of IEEE and MENSA. His areas of interest include Embedded Systems, Microprocessor Systems, Computer Systems Design and VLSI. (om_eagle@yahoo.com)



**Y.S. Rao** is a Master of Engineering , research Scholar at Indian Institute of Technology (IIT), Mumbai and currently teaching at Sardar Patel College of Engineering, Mumbai. His areas of interest include Embedded Systems, DSP applications in power electronics, Electrical Power Quality, Microprocessors and Microcontrollers. (ysrao@ee.iitb.ac.in)