IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPROVE THE PERFORMANCE OF DIFFERENT ORDER PLANTS BY USING NEURO FUZZY BASED PID CONTROLLER

[1]Saumya Gautam,[2]Prabhakar Dubey

[1]Research Scholar, [2]Professor

[1]Electronics and Communication Engineering Department,

[1]Rameshwaram Institute of Technology & Management, Lucknow, India

Abstract: Based on traditional PID control and nonlinear components, this research provides an optimum fuzzy proportional-integral-derivative (PID) controller architecture. Because of the similarity between fuzzy logic controllers (FLCs) and traditional PID controllers, a typical PID controller design may be quickly turned into an equivalent FLC by defining the controller's working ranges. The suggested nonlinear factors can be used to fine-tune the nonlinearity of the membership functions (MFs) that are dispersed throughout the operational ranges. In this approach, a fuzzy PID controller with fewer parameters may be built and optimized using the genetic algorithm (GA). Furthermore, the aforementioned corresponding FLC can operate as one member in the initial GA population, considerably improving GA efficiency. The simulation results show that this strategy is feasible. This resulted in an ideal fuzzy PID controller design with only eight parameters and a compact controller layout, as well as a more systematic optimal fuzzy PID controller design.

***Index Terms* - equivalence; optimal; fuzzy PID controller; genetic algorithm**

I. INTRODUCTION

Despite the development of several control theories, the well-known proportional-integral-derivative (PID) controllers are still frequently used in industrial process control. The popularity of a PID controller can be due to its high performance and ease of use. To provide a suitable response in settling time, steady-state error, and overshoot, the three-mode controller incorporates a proportional (P), an integral (I), and a derivative (D) term. An engineer may effectively tune the three gains using expertise or basic methods, such as the Ziegler-Nichols [1] tuning rules. Furthermore, a simpler PI or PD controller is popular for a wide range of practical applications. The foundation of a fuzzy logic controller (FLC) is fuzzy rules and fuzzy inference. Fuzzy rules may be used to regulate more complicated plants that might be linear or nonlinear, and they can represent human experience or knowledge. FLCs, like conventional PI or PD controllers, have PI or PD controllers. A FLC design consists essentially of the kind of FLC, the number and form of membership functions (MFs), and the fuzzy rules [2]. The genetic algorithm (GA) is used to find the best system settings. Because traditional linear PID controller design approaches have developed, it is preferable to employ the GA to optimize fuzzy PID controller design. An analytical design for an ideal fuzzy PID controller has been presented by researchers [3]. It has a basic framework, yet employs sophisticated techniques. Another ideal fuzzy PID controller was presented [4], which combines a fuzzy PI controller with a fuzzy D controller, although this device is basically a normal PID controller with adaptive control capabilities and difficult analytic formulae. Combining a fuzzy PI controller and a fuzzy PD controller in parallelism [5] with appropriate adjustment of scaling factors and MFs yields an ideal fuzzy PID controller. The standard PID controller may also be directly implemented in the ideal design for fuzzy controllers [6], where the PID control is the master controller and the fuzzy control is the slave control to supplement the master one. The fundamental aspect for a fuzzy PID controller design should be the controller structure. In terms of fuzzy control rules, they should, in theory, follow traditional PID control. The challenge of adjusting the MFs to increase system performance must then be tackled [7]. The form of MFs can be determined by chromosomal bits and optimized by the GA [8,9] in order to improve system responses such as control speed and precision [9,10]. Each fuzzy variable MF, on the other hand, is often assigned to a symmetrical form. The conversion of an MF from symmetrical [11] to asymmetrical [12] can increase system performance. Furthermore, some studies utilize scaling factors to normalize operating ranges and fine-tune scaling factors to complete optimization [13]. We chose to tweak the operational ranges in this study since they are critical factors for creating the comparable FLC from a regular PID controller.

Many evolutionary algorithms, such as the particle swarm algorithm (PSO), cuckoo search (CS), and others, have been created in recent decades. According to evolutionary programming (EP), the standard GA will not only reach a premature convergence but may also become imprisoned in the local optima. [14] Describes a fuzzy PID controller architecture based on a unique PSO-EP hybrid algorithm. Furthermore, it is demonstrated that an FLC + EP based PID controller responds faster than an FLC + GA based PID controller [15]. In this work, we will continue to use GA and give each optimized parameter its own crossover point in the GA process to improve the efficiency of the GA.

This work produced an ideal fuzzy PID controller with fewer parameters and a simple controller structure, in contrast to most optimal fuzzy PID controller designs with complicated structures or a high number of tuning parameters. The equivalence of fuzzy PID controllers and conventional PID controllers is demonstrated in [16] based on our earlier work. Nonlinear factors are also proposed to represent the nonlinearity of MF distributions in operational ranges. Each MF will have an uneven form of its own. There will be just eight changed parameters in the suggested optimum fuzzy PID controller. Furthermore, if a traditional PID controller design is available ahead of time, an analogous FLC in the original GA design can be utilized, potentially speeding up the optimization process. Pelosi has investigated developing optimum control systems using GA and neuro-fuzzy approaches [17,18,19], and the findings may be used to compare with the suggested design. Furthermore, the suggested optimum fuzzy PID controller is used to the motor control system [20], and the simulation results show effective speed control with disturbance rejection [21].

2. LITERATURESURVEY

Various types of neural fuzzy systems have been created and extensively investigated in recent years [1]–[3]–[13].A learning process is realized by the employment of numerous efficient learning algorithms, including back propagation and genetic algorithms.

The weight feature of a "feed-forward neural networks" is traditionally optimized via the back propagation technique. A widely used directed random search strategy in optimization issues is the genetic algorithm. These techniques are the most useful tools for improving neural fuzzy system performance.

The mathematical technique of wavelets separates data into multiple frequency components. Then, every element of frequency is looked at with a resolution appropriate to its scale. They offer several advantages over conventional Fourier techniques when analyzing physical situations when the signal contains discontinuities and sudden spikes.

Wavelets have been independently developed in the fields of math, quantum science, electric engineering, and seismic geology.

The modified PID controller was provided as a dynamic system controller by T. Yucelen, O. Kaymakci, and S. Kurtulan at [10] and the essential procedures are explained to show that the presented PID algorithm is more functional than the traditional PID controller method. Here, the Ziegler-Nichols approach is defined as self-tuning. Self-tuning techniques employ an adaptive PI-D controller algorithm. In this PI-D, the adaptive algorithm is controlled by proportional and integral parameters, and the derivative parameter takes a constant found in the Ziegler-Nichols-based self-tuning approach.

It may be advantageous to use fuzzy logic controllers to govern an inverted pendulum system, as suggested by M. I. H. Nour, J. Ooi, and K. Y. Chan [19].We demonstrated the phases of creating a Takagi-Sugeno fuzzy model that has four inputs and a fuzzy logic controller. The main objective of this study is to apply and optimize the use of fuzzy logic control approaches to balance an inverted pendulum and reduce the computation time of the controller. In this work, a system with an inverted pendulum was modelled and constructed using the performance of the suggested fuzzy logic and SIMULINK.

MATLAB simulations are utilized to compare the logic controller to the more widely used PID controller.

3. PROPOSED TECHNIQUE

Traditional PID controllers are frequently used in industrial applications**.** Fig. 1.1 shows the structure of control system plant which is having fuzzy-PID Controller and Neural Network for controlling the plants and identification of plants. In this paper we have used off line method to identify theplant by using Wavelet based Neural Network.

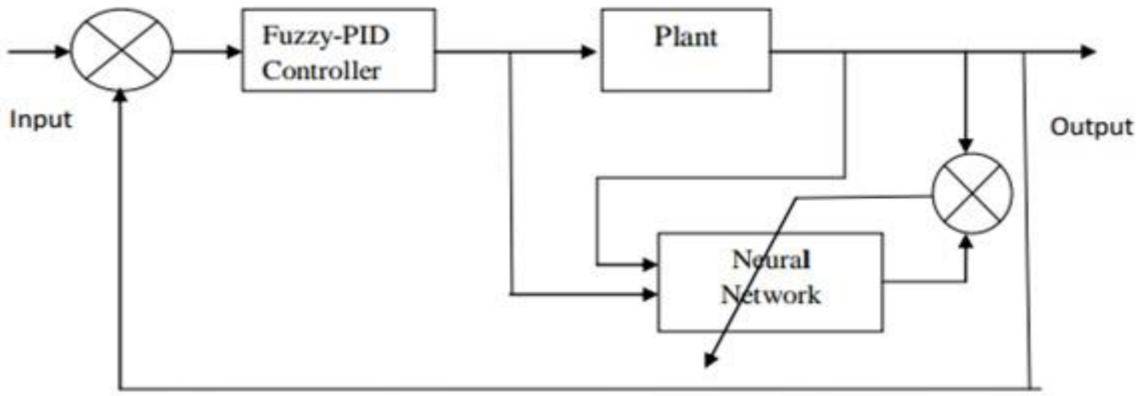

Fig 3.1 Structure of Control System

The following is the common equation:

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}, \quad (3.1)$$

Where the controller provides a proportional term, an integration term, and a derivative term. In the next section, we will describe the proposed optimal fuzzy PID controller design.

3.1.CONTROLLOOPBASICS

The procedure used to regulate the hot and cold spouts valves in order to keep the water coming out of the faucet at the proper temperature is a common illustration of a control loop. Usually, two process streams—the booth and cold water—are mixed together in this manner. To feel or gauge the water's temperature, the individual touches it. Talcing the measurement of a process variable or process parameter (PV) is similar to sensing water temperature. The set point (SP) is the intended temperature. The water valve's position is the process's input, and this is referred to as the variable that has been manipulated (MV). The variation between the temperature reading and.

The controller calculates the inaccuracy after sensing the ambient temperature (PV) and deciding when and how much to change the tap position (MV). When the controller first switches on the valve, the hot valve may only be partially opened if warm water is needed, or it may be entirely released if very hot water is necessary. Here is an example of a simple proportional control in use. The controller may try to speed up the procedure by gradually closing the hot water valves in the event that the hot water is delayed. This is a prime example of an integral control mechanism. If integral and proportionality management techniques are solely employed, it is conceivable that the water level in some systems may rise.

3.2. PIDCONTROLLERMETHODOLOGY

The equivalent and non-interacting variant of the "PID controller" design is described in this section. Please refer to the section titled "Alternative terminology and PID forms" for additional forms. The variable that is manipulated (MV) is the total of the three correcting terms that make the structure of the PIO management scheme (18]. Hence

$$MV(t)=Pout+Iout+Dout \tag{3.2}$$

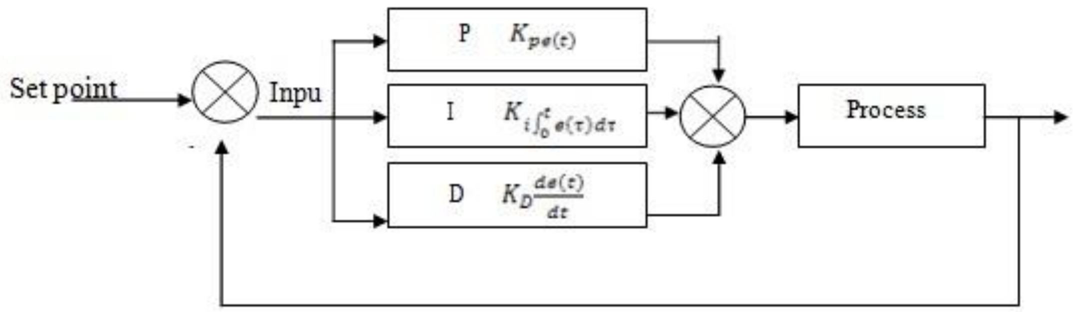

Fig.3.2 A block diagram of a PID controller

3.2.1. Proportional Term

The proportional term, also known as gain, modifies the output in a manner proportional to the current erroneous value. The proportionality response can be altered by raising the error by a certain Kp amount, also known as the proportional improvement (18).Given by is the proportionate term.

$$Pout=Kpe(t) \tag{3.3}$$

Where Pou,: Comparative term of output
Kp: Proportional gain, at using parameter
Error (e) =SP-MV
SP: Set Point
MV: Manipulated value
T: time(the present)

"For an alteration in the error, a high proportional increase causes a significant change in the output. The system may become unstable if the proportionate gain is more than 100. A lower gain, on the other hand, leads to a less receptive (or sensitive) controller and a little output reaction to a significant input mistake. While responding to system disturbances, if the proportional increase is too low, the action of control may be insufficient".

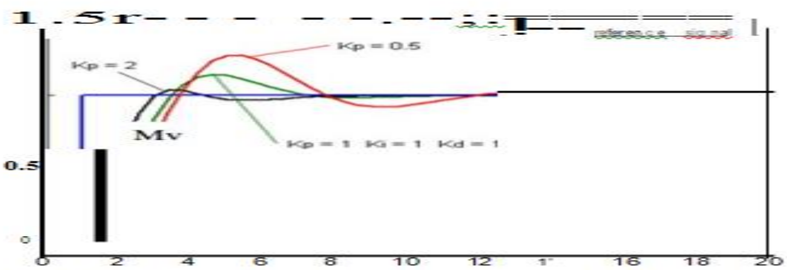

Fig. 3.2.1 plot of MV vs. Time for three values

3.2.2. INTEGRAL TERM

The integral term, sometimes referred to as reset, makes a contribution that is proportional to the size and duration of the error. The immediate inaccuracy is added over time (integrated), yielding the overall offset that needs to be addressed earlier. The output of the controller is then calculated by multiplying the integral gain by the total error. (1l).

The essential durations assumed by:

$$l_0u,=K1 g\, e(r)dt: \tag{3.3}$$

Where
Lout: Integral term of output
K_1: Integral gain, a tuning parameter
e: Error= SP-MV
SP: Set Point
MV: Manipulated value
t: time(the present)
rag dummy integration variable

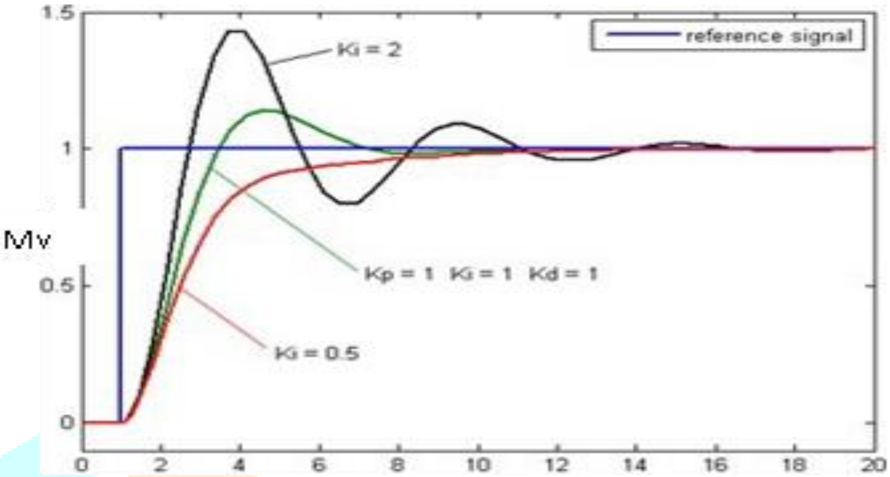

Fig.3.3 Plot of MV vs. time forth values

3.2.3. Derivative Term

The rate for the evolution of the procedure's error is calculated using the sloping curve for the amount of error over the course of time, or one of its initial derivatives with regard to time. The derivative gainK0 is subsequently multiplied by this slope. The total amount of the composite term's (which is additionally known as rate's) contributions to the overall control action is referred to as the "derivative gain K0[1l]". A derivative's value is given by and is:

$$D_{out}=K_d \frac{d}{dt}e\ (t) \tag{3.4}$$

Where
Dou,: Derivative term of output
Kd: Derivative gain, a tuning parameter e: Error= SP-MV,
SP: Set Point
MV: Manipulated value
t:Time

When the control device is in close proximity to its set point, the impact of the derivatives term, which slows the rate of output change, is most noticeable. “Therefore, derivative control is used to improve the stability of the controller-process as a whole and reduce the amount of the overshoot brought on by the integrated component. Signal differentiation, however, enhances noise, making that term in the control system very sensitive to alterations in the error term”.

The total of the integral, derivative, and proportional variables yields the PIO controller's output. The final form of the PIO method is: Using u(t) as the controller output.

u(t)=*MV(t)=Kpe(t)+Kif,'e(t)+Kp!:..e(t)*

(2.5)

3.3. MANUAL TUNING

One tuning option is to first set Kt and Ko quantities to zero if the computer must stay online. For a "quarter amplitude decay" style response, increase Kp until the output of the circuit oscillates, and then adjust Kp to around half that value. Then raise Kt until the process has enough time to fix any offset. Instability will result from K1 if Kt is greater than zero. Then, if necessary, raise K0 until the loop can return to its referent after a load perturbation. Overshoot and excessive reaction are however caused by too much Ko.

Table2.1Effects of independently raising a parameter when tweaking manually

Effects of increasing a parameter independently					
Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
Kp	Decrease	Increase	Small change	Decrease	Degrade
K1	Decrease	Increase	Increase	Decrease significantly	Degrade
Ko	Minor decrease	Minor decrease	Minor decrease	No effect m theoretically	Improve if *Kd* Small

3.3. PIDTUNING SOFTWARE

The manual computation techniques indicated above are no longer commonly used in modern industrial facilities to tune loops. Instead, software for PIO tuning or loop optimization is utilized to guarantee repeatable outcomes. These computer programs will collect the data, create process models, and offer the best tweaking. Some software programs even have the ability to tune themselves by collecting data after comparison modifications.

The system is given an impulse by mathematical PIO loop tuning, which then determines the PIO loop's values depending on the controlled system's frequency response. For loops with reaction durations of a few hours or more, mathematical looping optimization is advocated due to the fact that it may require days of experimentation in order to determine the right amount of loop values. Finding the best values is difficult. Many electronic loop controllers have a self-tuning feature that allows the process to receive incredibly small set point adjustments so that the controller may choose the ideal tuning values on its own.

3.3.1. FUZZY RULE

An expression in the conditional form: is known as a fuzzy rule. If axis A, then Ny is B, and X and Y are linguistic variables. A and B are linguistic values that are derived from fuzzy sets on X and Y, the respective universes of discourse.

“A classical IF-THEN statement uses binary logic, for instance”,

IF man height is > 180cm THEN man weight is>50kg

Boolean logic is used to express conventional rules. Fuzzy rules, on the opposite hand, have several values. For instance:

IF_{man} height is tall

THEN man weight is heavy [2]

3.3.2. FUZZY LOGIC

Fuzzy logic, a subset of multi-valued logical which deals with thinking that is imprecise rather than precise, was created as a result of fuzzy set theory. Instead of being constrained to the two truth values of conventional propositional logic, the corresponding level of validity of a contention can range from 0 to I. This is due to the fuzzy logic variables' ability to have set membership values other than O or I. Unlike "crisp logic," which only applies to binary sets, which also has binary logic. Furthermore, in the presence of linguistic variables, such levels may be regulated by specific functions.

Fuzzy logic was created as a result of LotfiZadeh's fuzzy set theory concept in 1965. Although fuzzy logic has been used to a variety of fields, including artificial intelligence and control theory.

3.3.3. FUZZILYINPUTS:

The initial step is to use membership functions to assess the inputs' degree of membership in each of the relevant fuzzy sets. The output of the Fuzzy Logic Toolbox software is a fuzzy degrees of membership in the qualification linguistic set, which is always in the range between 0 and I. The input is usually a crisp numerical value that limits the universe of conversation of the variable being used (in this case, the interval between O and 10). The input is fuzzy field, which results in either a database lookup or an algorithm evaluation.

The three rules that make up this example each require that the inputs be resolved into various fuzzy linguistic sets. Poor service, nice service, and food?

The following graph illustrates how well the hypothetical restaurant's food (ranked on a scale of O to 10) meets the criteria for the linguistic variable "delicious" (via its membership function). Given your visual definition of delicious, our rating of 8 in this situation equates to a yummy membership function value of 0.7.

Each input is fuzzed in this way over all the qualified membership functions needed to comply with the requirements.

3.3.4. APPLY FUZZY OPERATOR

You can determine the extent to which each component of the antecedent is satisfied for each rule once the inputs have been fuzzified. The fuzzy operator is used to provide a single integer that represents the outcome of the antecedent for that rule when the antecedent of a particular rule has more than one part. The output function is then used with this number. Fuzzified input variables provide two or more membership values as the input to the fuzzy operator. One truth value is produced as the output.

As is described in Logical Operations section, any number of well-defined methods can fill in for the AND operation other OR operation. In the toolbox, two built-in AND methods are supported: min (minimum) and prod (product). Two built-in OR methods are also supported: max (maximum), and the probabilistic OR method prober. The probabilistic OR method (also known as the algebraic sum) is calculated according to the equation

$$\text{probor}(a,b)=a+b-ab$$

In addition to these pre-built ways, you can write any function and designate it as your preferred method to develop your own approaches for AND and OR.

3.4. APPLYIMPLICATIONMETHOD

You must ascertain the weight of the rule prior to using the implication approach. Every rule has a weight (a value in the range of 0 and!), which is applied to the antecedent's number. This weight is typically I (as it is in this example) and has no impact whatsoever on the implication process. You could occasionally want to give one rule more weight than the others by altering the value of its weight from I to something else. The implication approach is used once each rule has received the appropriate weighting. A consequent is a fuzzy set that is represented by a membership function that suitably weights the attributes that are given to it in terms of linguistic properties. Using a function related to the antecedent (a single number), the consequent is transformed. A single number provided by the antecedent serves as the implication process' input, while a fuzzy set is produced as the process' output. Each rule includes implementation of implications. The two built-in techniques supported are min (minimum), which reduces the output fuzzy collection, and prod (product), which grows the output fuzzy set. These two built-in methods are identical to those used by the AND method.

Aggregate All Outputs:

Decisions must be made by combining the rules in some way because they depend on the evaluation of every rule in a FIS. The act of amalgamating the fuzzy sets that represent each rule's outputs into a single fuzzy set is known as aggregation. For each output variable, aggregation only happens once, shortly before the defuzzification phase, the fifth and last step. The list of shortened output functions that each rule's implication process returned is the input to of the aggregation process. Each output variable's fuzzy set is the result of the aggregation procedure.

4. RESULTS AND CONCLUSION

In this chapter, the outcomes of the plant employing various controller types are discussed. Several controllers, including PIO, Fuzzy Logic, and Neuro-Fuzzy, are being used to govern the plant's responses.

In this thesis we have used transfer function as a plant and find out the responseoftheplanti.e.2ndorder,3rdorder and 5thorder respectively by applying the step function as an input. In order to find out the response we have taken following transfer function as a plant.

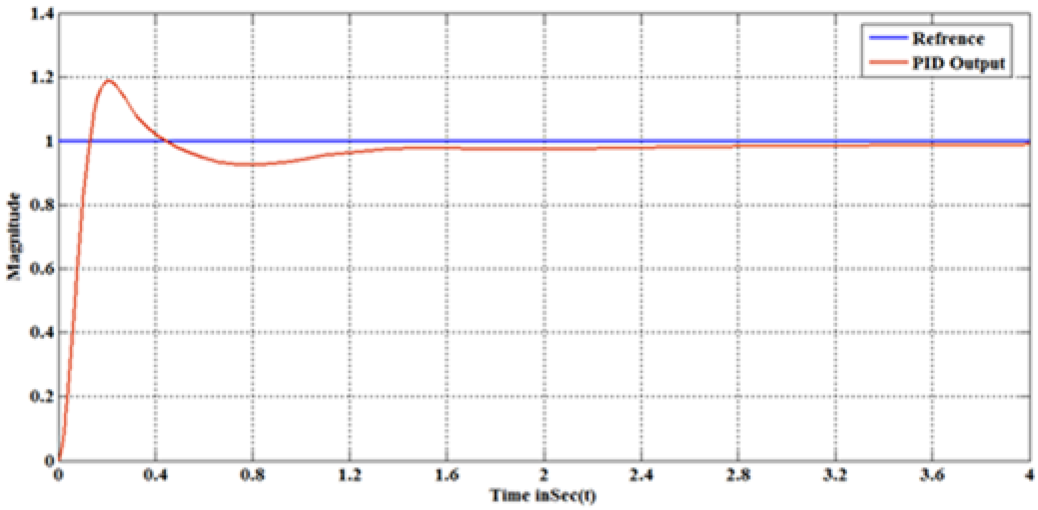

Fig 4.1. PIO Output of 3rdorder system

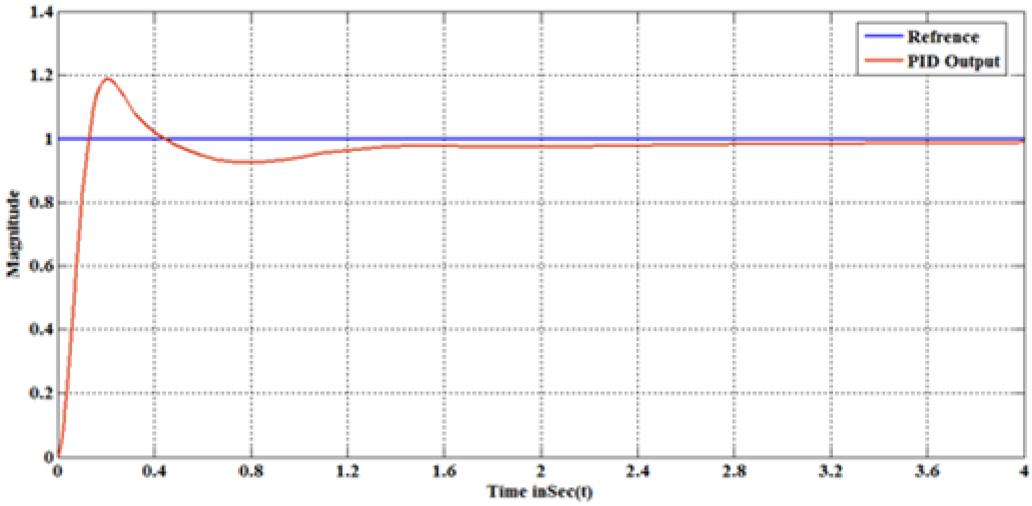

Fig 4.2 PIO Output of 5th orders ys tem

The system can provide the fuzzy logic controller with two inputs: error and its derivative. By altering the gain values for Kp, Ki, and Kd, the PIO Controller was able to modify the fuzzy logic controller's output demonstrate the output response of a second order, third order, and fifth order plant, respectively, while depicts the simulation of a fuzzy logic controller.

Using Fuzzy Logic Controller the value of gain kp,k_1 and kd adjust automatically with changing the transfer function of plant. Fig. shows the simulation output of the Fuzzy PIO controller of 2nd order, 3rdorder, and 5th order plant.

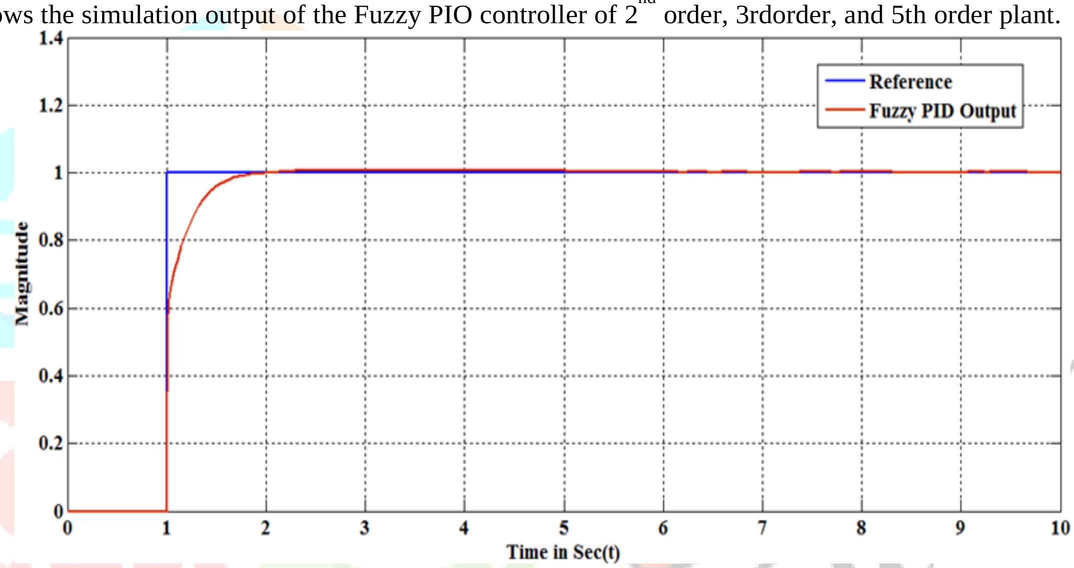

Fig.4.3 Simulation output of 2ndorder plant with Fuzzy PIO

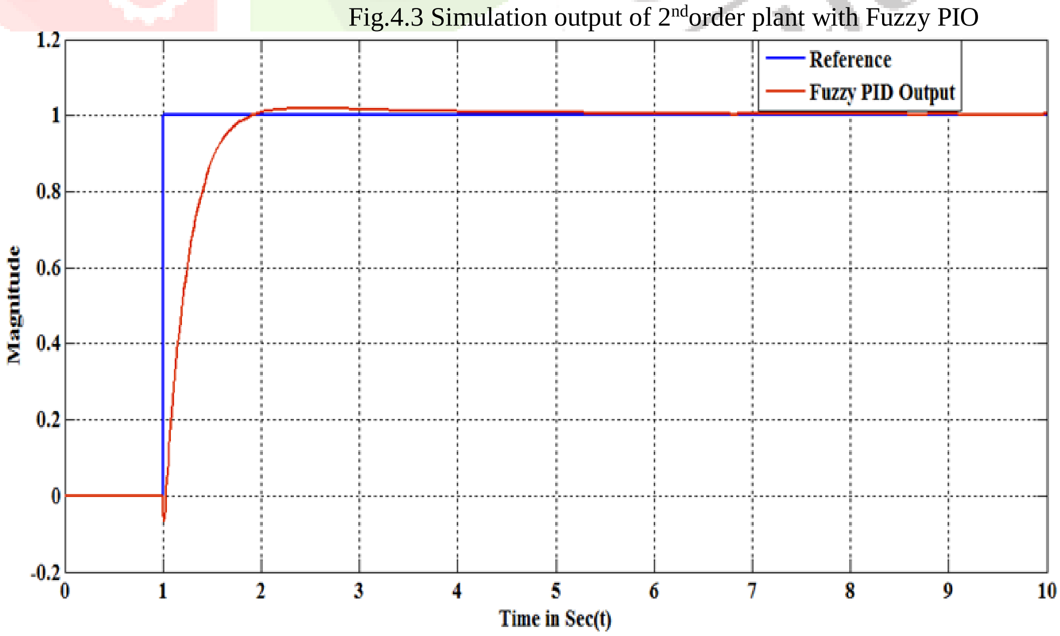

Fig. 4.4 Simulation output of 3'''order plant with Fuzzy PIO

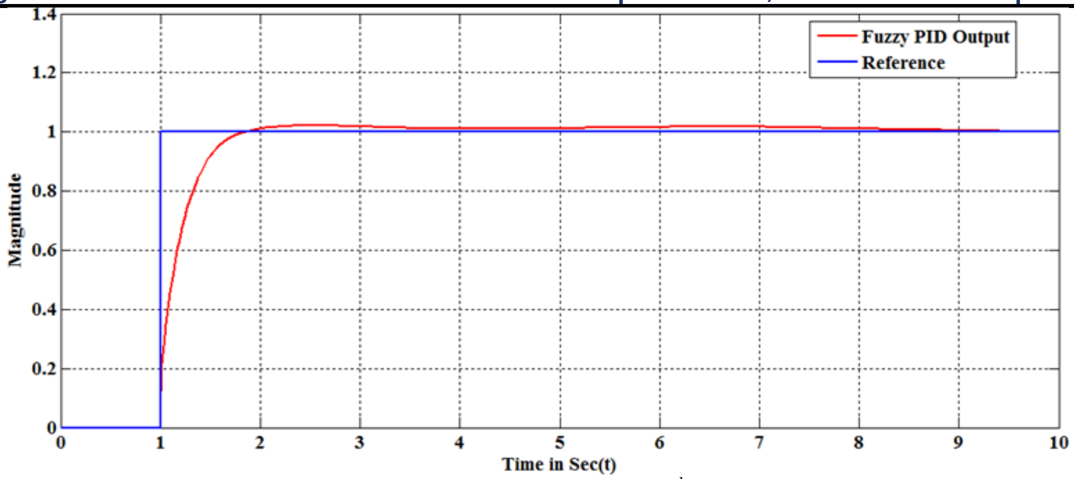

Fig.4.5 Simulation output of 5th order plant with Fuzzy PIO

Fig. 6.9 Show the comparative result of PIO Controller and Adaptive Fuzzy PIO Controller for Sth order plant. A simulation result shows that the response of Adaptive Fuzzy PIO Controller is superior in comparison to conventional PIO Controller

Finally we can see that adaptive Fuzzy PIO Controller gives better performance in comparison to conventional PIO Controller for any order of plants.

4.1. CONCLUSION

The PID controller is suited for processes with almost monotonous step responses provided that the requirements are not too strict. Simple controllers like the PI and PIO controller are obviously not suitable for all processes. When the criteria are extremely strict or the process is highly oscillatory, a conventional PIO controller is not appropriate.

The PIO controller's Kp, K1, and Kd values are adjusted using a self-tuning fuzzy controller. The performance of plants is improved in testing employing step input signals, and the outcomes are satisfactory when compared to standard PIO controllers, according to higher order plants' responses.

5. References

1. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–765. [**Google Scholar**] [**CrossRef**]
2. Yan, K.; Mo, H. Application of fuzzy control under time varying universe in unmanned vehicles. In Proceedings of the 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanjing, China, 18–20 May 2018; pp. 439–444. [**Google Scholar**]
3. Hu, B.G.; Mann, G.K.I.; Gosine, R.G. New methodology for analytical and optimal design of fuzzy PID controllers. *IEEE Trans. Fuzzy Syst.* **1999**, *7*, 521–539. [**Google Scholar**]
4. Tang, K.S.; Man, K.F.; Chen, G.; Kwong, S. An optimal fuzzy PID controller. *IEEE Trans. Ind. Electron.* **2001**, *48*, 757–785. [**Google Scholar**] [**CrossRef**]
5. Sebastiao, A.; Lucena, C.; Palm, L.; Cardoso, A.; Gil, P. Optimal tuning of scaling factors and membership functions for mamdani type PID Fuzzy controllers. In Proceedings of the IEEE International Conference on Control, Automation and Robotics, Singapore, 20–22 May 2015. [**Google Scholar**] [**CrossRef**]
6. Zhou, Y.S.; Lai, L.Y. Optimal design for fuzzy controllers by genetic algorithms. *IEEE Trans. Ind. Appl.* **2000**, *36*, 93–97. [**Google Scholar**] [**CrossRef**]
7. Salleh, Z.; Sulaiman, M.; Omar, R. Tuning fuzzy membership functions to improve performance of vector control induction drives. *J. Telecommun. Electron. Comput. Eng.* **2016**, *8*, 1–4. [**Google Scholar**]
8. Kapetanovic, N.; Osmic, N.; Konjicija, S. Optimization of membership functions of Sugeno_Takagi fuzzy logic controllers with two inputs and one input using genetic algorithms. In Proceedings of the IEEE x International Symposium on Telecommunications (BIHTL), Sarajevo, Bosnia and Herzegovina, 27–29 October 2014. [**Google Scholar**]
9. Pelusi, D.; Vazquez, L.; Diaz, D.; Mascella, R. Fuzzy algorithm control effectiveness on drum boiler simulated dynamics. In Proceedings of the 2013 36th International Conference on Telecommunications and Signal Processing (TSP), Rome, Italy, 2–4 July 2013; pp. 272–276. [**Google Scholar**]
10. Ranjani, M.; Murugesan, P. Modeling of genetic algorithm based fuzzy logic controller for quasi-resonant converter fed drive. *Int. J. Concept. Electr. Electron. Eng.* **2013**, *1*, 25–29. [**Google Scholar**]
11. Chen, S.M.; Chang, Y.C. A new weighted fuzzy rule interpolation method based on GA-based weights-learning techniques. In Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; pp. 11–14. [**Google Scholar**]
12. Al-Gizi, A.; Craciunescu, A.; Al-Chlaihawi, S. Improving the performance of PV system using genetically-tuned FLC based MPPT. In Proceedings of the 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Brasov, Romania, 25–27 May 2017; pp. 642–647. [**Google Scholar**]
13. Gritli, W.; Gharsallaoui, H.; Benrejeb, M. PID-type fuzzy scaling factors tuning using genetic algorithm and simulink design optimization for electronic throttle valve. In Proceedings of the IEEE International Conference on Control, Design and Information Technologies (CoDIT), St. Julian's, Malta, 6–8 April 2016; pp. 216–221. [**Google Scholar**]
14. Chiou, J.S.; Liu, M.T. Numerical simulation for Fuzzy-PID controllers and helping EP reproduction with PSO hybrid algorithm. *Simul. Model. Pract. Theory* **2009**, *17*, 1555–1565. [**Google Scholar**] [**CrossRef**]
15. Yau, H.T.; Yu, P.H.; Su, Y.H. Design and Implementation of optimal Fuzzy PID controller for DC servo motor. *Appl. Math. Inf. Sci.* **2014**, *8*, 231–237. [**Google Scholar**] [**CrossRef**]

16. Chao, C.T.; Sutarna, N.; Chiou, J.S.; Wang, C.J. Equivalence between Fuzzy PID controllers and conventional PID controllers. *Appl. Sci.* **2017**, *7*, 513. [**Google Scholar**] [**CrossRef**]
17. Pelusi, D. Optimization of fuzzy logic controller using genetic algorithms. In Proceedings of the IEEE Third International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 26–27 August 2011; Volume 2, pp. 143–146. [**Google Scholar**] [**CrossRef**]
18. Pelusi, D. Designing neural networks to improve timing performances of intelligent controllers. *J. Discret. Math. Sci. Cryptogr.* **2013**, *16*, 187–193. [**Google Scholar**] [**CrossRef**]
19. Pelusi, D.; Mascella, R. Optimal control algorithms for second order systems. *J. Comput. Sci.* **2013**, *9*, 183–197. [**Google Scholar**] [**CrossRef**]