Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

# SET POINT WEIGHT TUNING USING PID-FUZZY LOGIC CONTROLLER

Gagan Soni[1], Ashish Singh Thakur[2], Himmat Singh[3]
[1, 2, 3] *Electrical Engineering Department, MITS, Gwalior*

**Abstract -** In this paper, a novel methodology, based on fuzzy logic, for the tuning of proportional-integral derivative (PID) controllers is presented. PID controllers are the most widely used controllers in the industry. Tuning of PID controllers has always been an area of active interest in the process control industry. The conventional PID controller is not very efficient due to the presence of non linearity in the system of the plant and also it has a quite high overshoot and settling time. Many tuning formulae that have been devised such as the Ziegler-Nichols one, assures a good load-disturbance attenuation, but often fail to achieve satisfactory performances, and therefore the operator has to use their experience and might fail to attain the best performances. Though ZN tunes systems very optimally, a better performance is needed for very fine response and this is obtained by using Fuzzy Logic Controller (FLC) methodology which is highly effective. This paper has two main parts. Firstly, a PID controller has been designed by using Ziegler-Nichols frequency response method and its performance has been observed. The Ziegler Nichols tuned controller parameters are fine tuned to get satisfactory closed loop performance. Secondly, for the same systems a PID–FLC has been proposed. Performance comparison between Ziegler Nichols tuned PID controller, and the proposed PID–FLC is presented. The simulation studies are then conducted based on the developed model using MATLAB and Simulation. The behavior of the system is studied in terms of time response (e.g., steady state error, rise-time, settling time, overshoot) and compare PID-FLC adverse Ziegler Nichols tuned PID controller. It found that the performance of the PID-FLC is better than Ziegler Nichols tuned PID controller.

**Keywords** — PID (proportional-integral-derivative), Fuzzy Logic Controller (FLC), Ziegler Nichols Method (ZN), Fuzzy Set Point Weighting Controller (FSPWC), Membership Functions (MF).

## I. INTRODUCTION

It is well-known that, despite many complicated control theories and techniques that have been devised in the last decades, proportional-integral-derivative (PID) controllers are still the most adopted in practical cases. In fact, due to their simple arrangement, PID controllers are relatively simple to tune and their use is well understood by a great majority of industrial practitioners and automatic control designers. In addition, they can provide acceptable performances for a large range of processes, so that other controllers are unlikely to achieve the same cost and benefit ratio.

It is also well-known that PID controllers are mainly adequate for processes whose dynamics can be effectively modeled by a first or second order system. Unfortunately, real systems have, in general, significant characteristics such as high-order, nonlinearities, dead-time, etc. and they can be affected by noise, load disturbances and other environmental conditions that cause parameter variations and sudden modifications of the model structure [1]. PID controllers are designed for linear systems and they provide a preferable advantage. However, the existence of nonlinear property limits their performance. Fuzzy Logic Controllers are effectively applied to non-linear system because of their knowledge based nonlinear structural characteristics [2]. The FLC methodology used in this paper is applied in the form of Fuzzy Set Point Weighting Controller (FSPWC) [3-4]. The idea of multiply the set point value for the proportional action by a constant parameter less than one is effective in reducing the overshoot but has the disadvantage of increasing the rise time. To achieve both the objective of reducing the overshoot and decreasing the rise time, a PID-FLC can be used to modify the weight depending on the current output error and its time derivative. To overcome these difficulties inherent in controlling a system that is both nonlinear and time dependent parameters, a controller based on PID-FLC was implemented [5]. PID-FLC is known for their ability to provide very good control of this type of system.

## II. TUNING METHODS

### A. Ziegler-Nichols Tuning Method

The Ziegler–Nichols tuning method is a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and B. Nichols. It is performed by setting "I" (integral) and "D" (derivative) gains to zero. The "P" (proportional) gain, is then increased (from zero) until it reaches the ultimate gain , at which the output of the control loop oscillates with a constant amplitude and the oscillation period are used to set the P, I, and D Gains depending on the type of controller used. In Ziegler Nichol's (ZN) Method the tuning relations reported by Ziegler and Nichols were determined empirically as shown in table 1 to provide closed-loop responses that have a quarter decay ratio the Z-N controller settings have been widely used as a benchmark for evaluating different tuning methods and control strategies.

*Table 1 Ziegler-Nichols P-I-D Controller Tuning Method Adjusting $k_p$ , $k_i$ & $k_d$*

| Ziegler-Nichols method giving 'K' values | | | |
|---|---|---|---|
| **Control Type** | **$k_p$** | **$k_i$** | **$k_d$** |
| P | $0.50k_c$ | 0 | 0 |
| PI | $0.45k_c$ | $1.2k_p\ dt/p_c$ | 0 |
| PID | $0.60k_c$ | $2k_p\ dt/p_c$ | $k_p p_c / dt$ |

As a result, the whole process depends on two variables and the other control parameters are calculated according to the table 1.

### B. PID Controller

Proportional- integral- derivative controllers have been used for industrial purpose due to their simplicity, easy designing method, low cost and effectiveness. These are the most widely used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. Due to presence of non linearity in the system, conventional PID controller is not very efficient. Proportional (P), integral (I) and derivative (D) are the three main parameters of the PID controller. The values of these three parameters interpreted in terms of time, where P depends on the present error, 'I' on the accumulation of past errors and 'D' is a prediction of future errors, based on current rate of change. By tuning the three parameters in the algorithm of PID controller, the controller can provide control action designed for specific process requirements. The proportional, integral and derivative terms are summed to calculate the output of the PID controller. The final output defined by u (t) and it given by

$$u(\mathrm{t}) = K_P e(t) + K_d \frac{\mathrm{de(t)}}{\mathrm{dt}} + K_i \int_0^t e(x)\,\mathrm{dx} \tag{1}$$

Where $K_P$ - Proportional gain, $K_i$ - Integral gain, $K_d$ - Derivative gain, e - Error present in the controller, t - Time or instantaneous time, x - Variable of integration, taken from time 0 to 1.

 

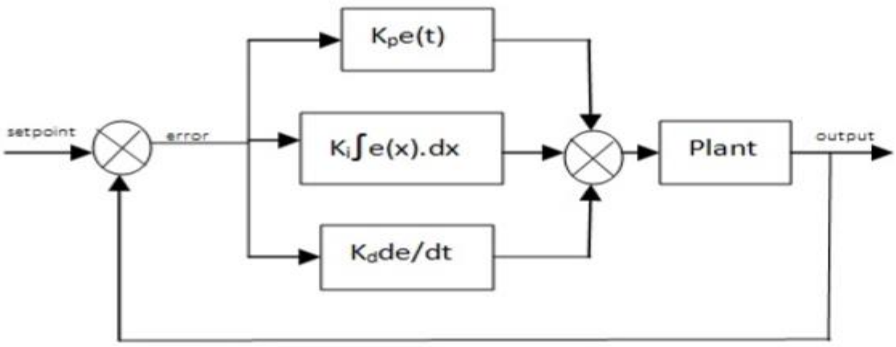


*Figure 1 Block diagram of conventional PID controller*

### C. Fuzzy Logic Controller

Fuzzy logic is a superset of conventional logic that has been extended to handle the concepts of ‘completely true’ and ‘completely false’ values. As its name suggests, it is the logic underlying modes of reasoning which are approximate rather than exact. The importance of Fuzzy logic derives from the fact that most modes of human reasoning and especially common sense reasoning are appropriate in nature.

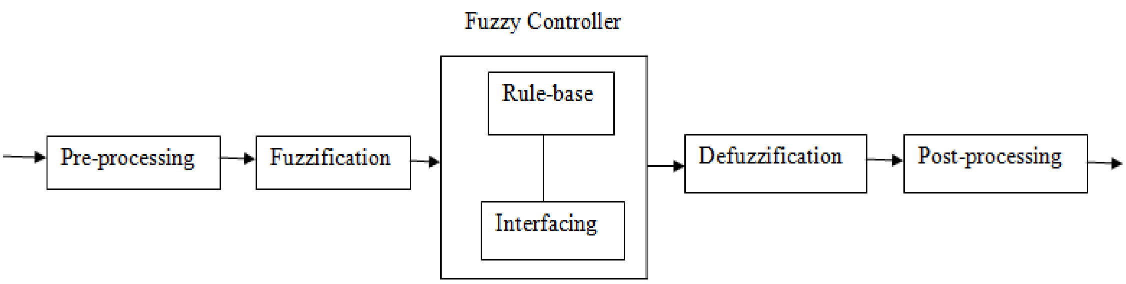


*Figure 2 Structure of fuzzy logic system*

Fuzzy logic is having many values. Here these appropriate values are not fixed time traditional binary sets. So, it is having a truth value that ranges in degree between 0 and 1. Therefore this type of logic system is able to address the values of variables those lie between completely truths and completely false. The variables are called the linguistic variables and each linguistic variable is described by a membership function which gives the probable decision making is an important part of the fuzzy logic. The decision making is mainly the combination of concepts of fuzzy set theory, fuzzy IF-THEN rules and fuzzy reasoning. The fuzzy system makes use of if then statements and with the help of connecters (such as AND gate) necessary rules are constructed. The structure of fuzzy system can be classified according to the different applications. One of the most popular types is the error feedback fuzzy controller, which is called fuzzy logic controller (FLC). In conventional FLC, there are also PD-type FLC, PI-type FLC and PID-type FLC.

## III. DESIGNING OF FUZZY LOGIC CONTROLLER

### A. FIS Editor

In this paper, presents a methodology for rule base fuzzy logic controller applied to a system. The Fuzzy Logic Controller is to be designed, before running the simulation in MATLAB/SIMULINK. It is done using the FIS editor. Using the Fuzzy logic toolbox, FIS file is created. To design of a Fuzzy Logic Controller, it requires the choice of Membership Functions. After the appropriate membership functions are chosen, a rule base is created. In a fuzzy Controller, the set of linguistic rules is the most essential part. The various linguistic variables to design rule base for output of the fuzzy logic controller are enlisted in Table 2. Using in MATLAB/SIMULINK The response of the fuzzy logic controller is obtained. A two input which is Error (e) & Error Compliment (e') and one – output variable change in control, fuzzy controller is created and the membership functions and fuzzy rules are determined.

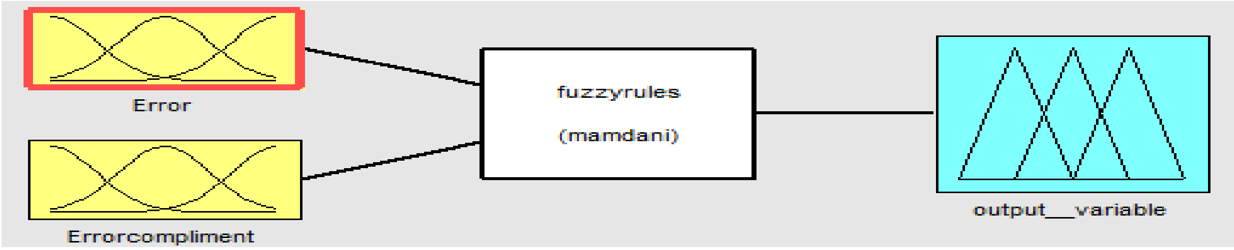


*Figure 3 Mamdani type fuzzy controllers*

**B. Membership Function Editor**

The Membership Function Editor shares some features with the FIS Editor. In fact, all of the five basic graphical user interface tools have similar menu options. The MF Editor is the tool that let you display and edits all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system [6-7].

**a) Fuzzy Set Characterizing Input**

**i. Error [Range (-1 to 1)]**

*Table 2 Crisp range table for "Error"*

| Fuzzy Variable | MF Used | Crisp Input Range |
|---|---|---|
| Negative Big (nb) | Triangular MF | [-1.5 -1 -0.5] |
| Negative Small (ns) | Triangular MF | [-1 -0.5 0] |
| Zero (z) | Triangular MF | [-0.5 0 0.5] |
| Positive Small (ps) | Triangular MF | [0 0.5 1] |
| Positive Big (pb) | Triangular MF | [0.5 1 1.5] |

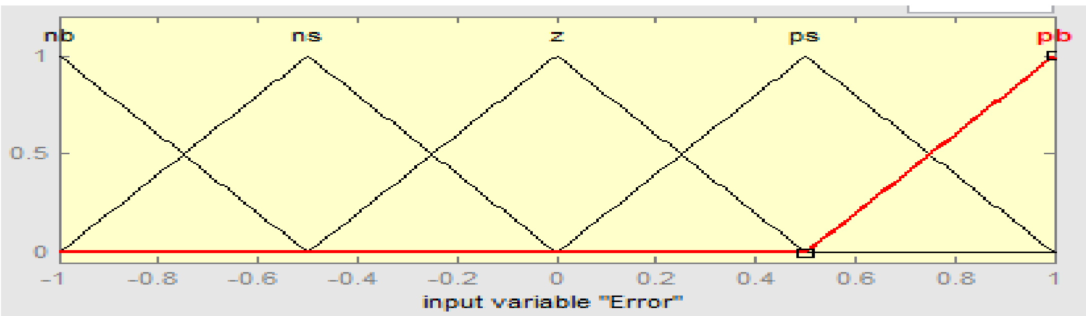


*Figure 4 Membership function fuzzy set characterize input*

**ii. Error Compliment [Range (-1 to 1)]**

*Table 3 Crisp range table for "Error Compliment"*

| Fuzzy Variable | MF Used | Crisp Input Range |
|---|---|---|
| Negative Big (nb) | Triangular MF | [0.5 1 1.5] |
| Negative Small (ns) | Triangular MF | [0 0.5 1] |
| Zero (z) | Triangular MF | [-0.5 0 0.5] |
| Positive Small (ps) | Triangular MF | [-1 -0.5 0] |
| Positive Big (pb) | Triangular MF | [-1.5 -1 -0.5] |

 

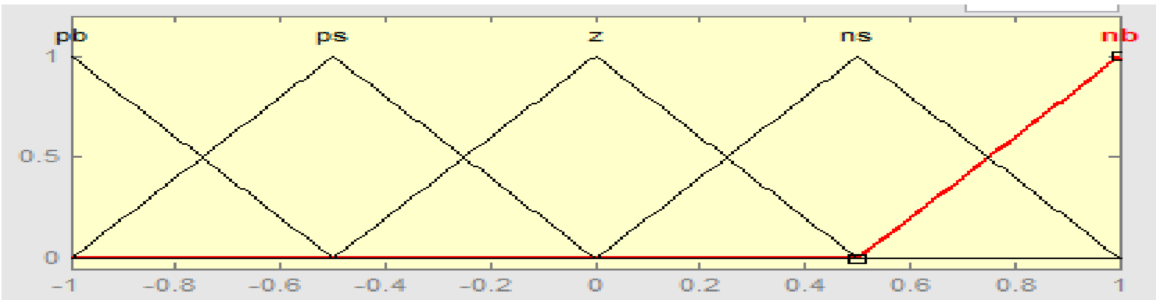


*Figure 5 Membership function fuzzy set characterize output*

## b) Fuzzy Set Characterizing Output

### i. Output Variable [(Range (-1 to 1)]

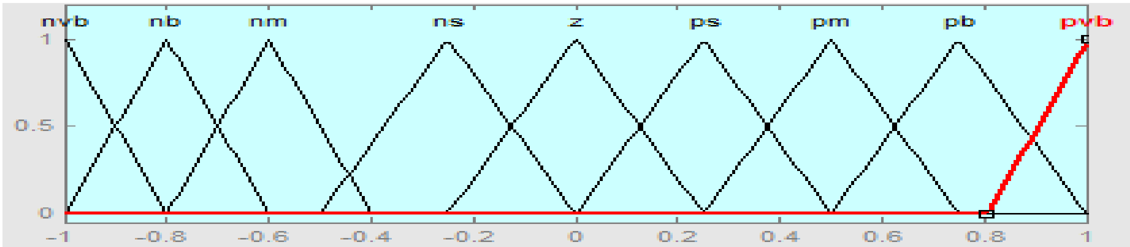


*Figure 6 Triangular membership function output*

*Table 4 Crisp range table for "Output Variable"*

| Fuzzy Variable | MF Used | Crisp Input Range |
|---|---|---|
| Negative Very Big (nvb) | Triangular MF | [-1.2 -1 -0.8] |
| Negative Big (nb) | Triangular MF | [-1 -0.8 -0.6] |
| Negative Medium (nm) | Triangular MF | [-0.8 -0.6 -0.4] |
| Negative Small (ns) | Triangular MF | [-0.5 -0.25 0] |
| Zero (z) | Triangular MF | [-0.25 0 0.25] |
| Positive Small (ps) | Triangular MF | [0 0.25 0.5] |
| Positive Medium (pm) | Triangular MF | [0.25 0.5 0.75] |
| Positive Big (pb) | Triangular MF | [0.5 0.75 1] |
| Positive Very Big (pvb) | Triangular MF | [0.805 1.01 1.21] |

## C. Rule Editor

Constructing rules using the graphical Rule Editor interface is fairly self-evident. Based on the input and output variables defined with the FIS Editor, the Rule Editor allows you to create the rule statements automatically [8].

1. If (Error is nb) and (Errorcompliment is nb) then (output__variable is nvb) (1).
2. If (Error is nb) and (Errorcompliment is ns) then (output__variable is nb) (1).
3. If (Error is nb) and (Errorcompliment is z) then (output__variable is nm) (1).
4. If (Error is nb) and (Errorcompliment is ps) then (output__variable is ns) (1).
5. If (Error is nb) and (Errorcompliment is pb) then (output__variable is z) (1).
6. If (Error is ns) and (Errorcompliment is nb) then (output__variable is nb) (1).
7. If (Error is ns) and (Errorcompliment is ns) then (output__variable is nm) (1).
8. If (Error is ns) and (Errorcompliment is z) then (output__variable is ns) (1).
9. If (Error is ns) and (Errorcompliment is ps) then (output__variable is z) (1).
10. If (Error is ns) and (Errorcompliment is pb) then (output__variable is ps) (1).
11. If (Error is z) and (Errorcompliment is nb) then (output__variable is nm) (1).
12. If (Error is z) and (Errorcompliment is ns) then (output__variable is ns) (1).
13. If (Error is z) and (Errorcompliment is z) then (output__variable is z) (1).

14. If (Error is z) and (Errorcompliment is ps) then (output__variable is ps) (1).
15. If (Error is z) and (Errorcompliment is pb) then (output__variable is pm) (1).
16. If (Error is ps) and (Errorcompliment is nb) then (output__variable is ns) (1).
17. If (Error is ps) and (Errorcompliment is ns) then (output__variable is z) (1).
18. If (Error is ps) and (Errorcompliment is z) then (output__variable is ps) (1).
19. If (Error is ps) and (Errorcompliment is ps) then (output__variable is pm) (1).
20. If (Error is ps) and (Errorcompliment is pb) then (output__variable is pb) (1).
21. If (Error is pb) and (Errorcompliment is nb) then (output__variable is z) (1).
22. If (Error is pb) and (Errorcompliment is ns) then (output__variable is ps) (1).
23. If (Error is pb) and (Errorcompliment is z) then (output__variable is pm) (1).
24. If (Error is pb) and (Errorcompliment is ps) then (output__variable is pb) (1).
25. If (Error is pb) and (Errorcompliment is pb) then (output__variable is pvb) (1).

## IV. SIMULATION BLOCK DIAGRAM

### A. Simulation Model Using Ziegler-Nichols

Simulation model Using Ziegler-Nichols for set point tuning control as shown in figure 7.

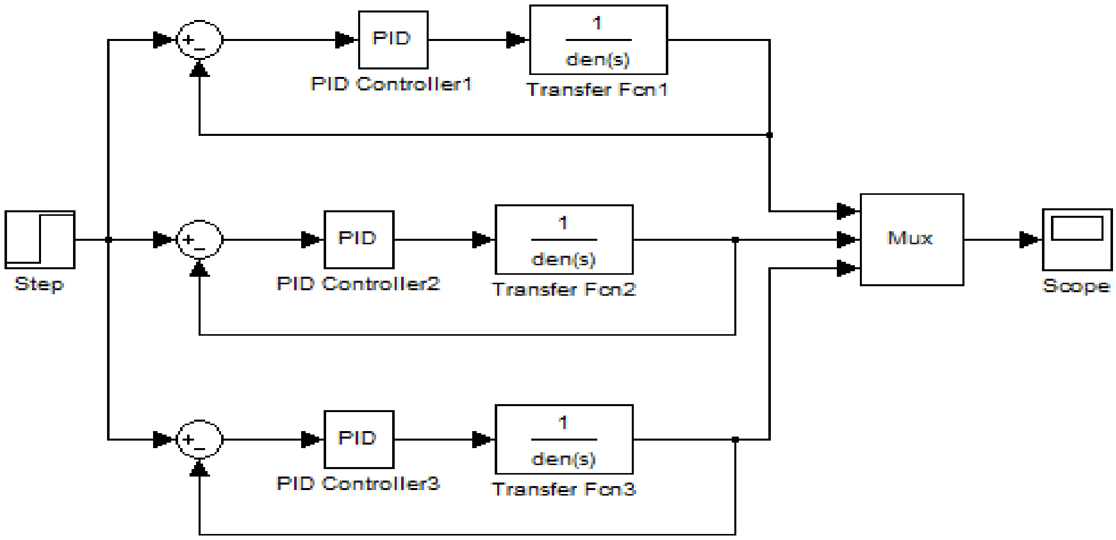


*Figure 7 Simulation model by using Ziegler-Nichols*

### B. Simulation Model Using Fuzzy Logic Controller

A simulation model using Fuzzy Logic Controller for set point tuning control as shown in figure 8.

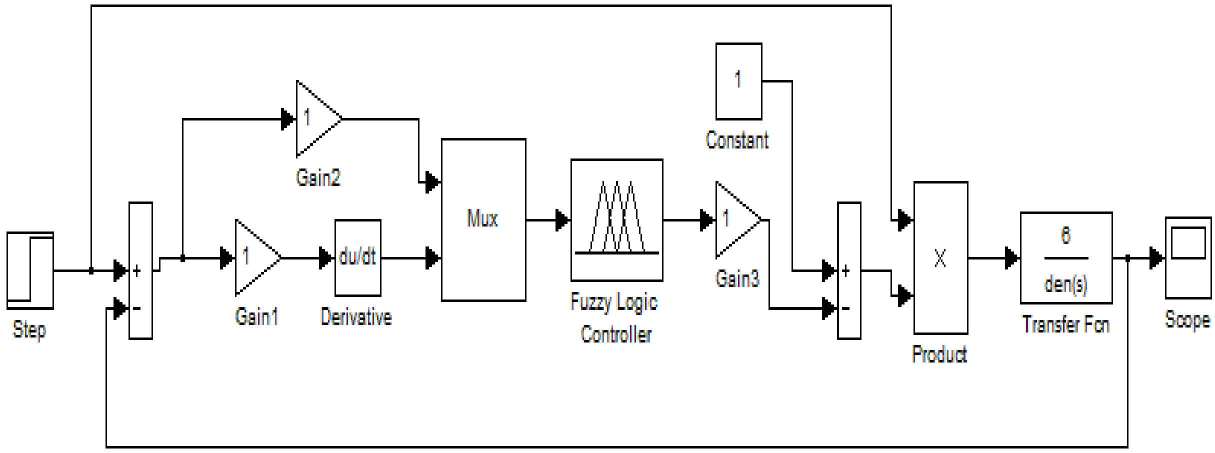


*Figure 8 Simulation model using fuzzy logic controller*

 

### C. Simulation Model Using PID-FLC

A simulation model using PID-FLC for set point tuning control as shown in figure 8.

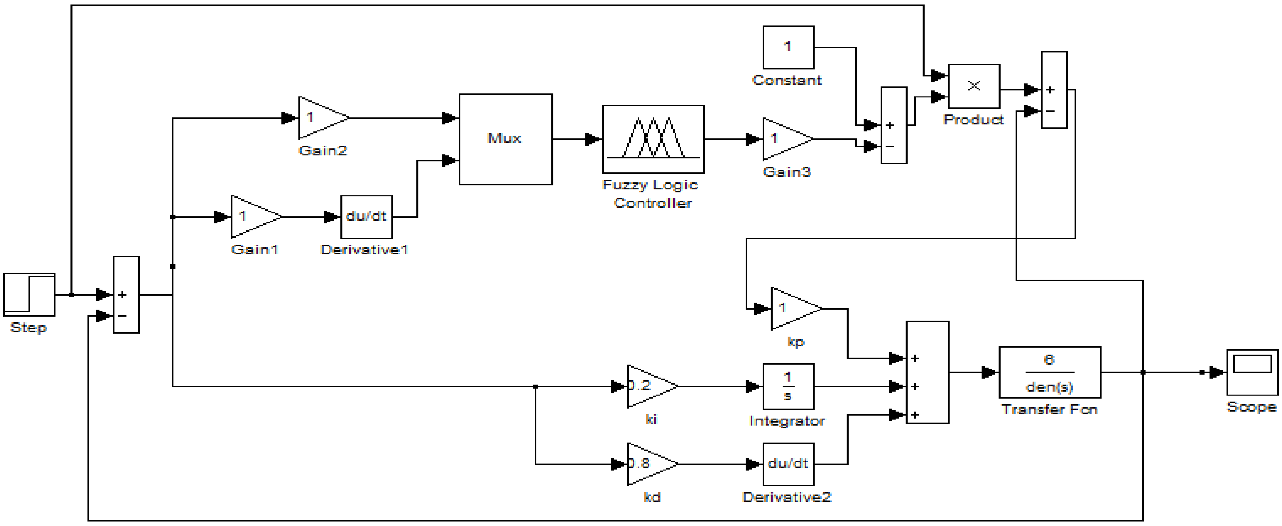


*Figure 9 Simulation model using PID-FLC*

## V. SIMULATION RESULTS & DISCUSSION

The system under consideration is

$$G(s) = \frac{1}{0.008s^3 + 0.04s^2 + 0.5s + 1}$$

To verify the full potentialities of the investigated methodologies, it will be assumed that no saturation levels are present for the control variable. After the tuning phase, accomplished using the various techniques, the unit step responses have been simulated. The resulting values of the time domain specifications and performance criteria are reported.

### A. Simulation Result Using Ziegler-Nichols

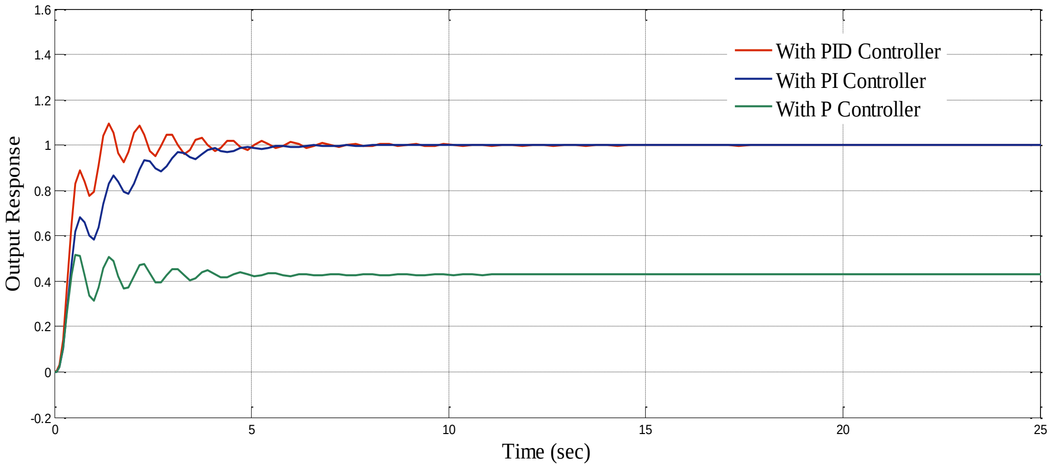


*Figure 10 Simulation result using Ziegler-Nichols*

 

## B. Simulation Result Using Fuzzy Logic Controller

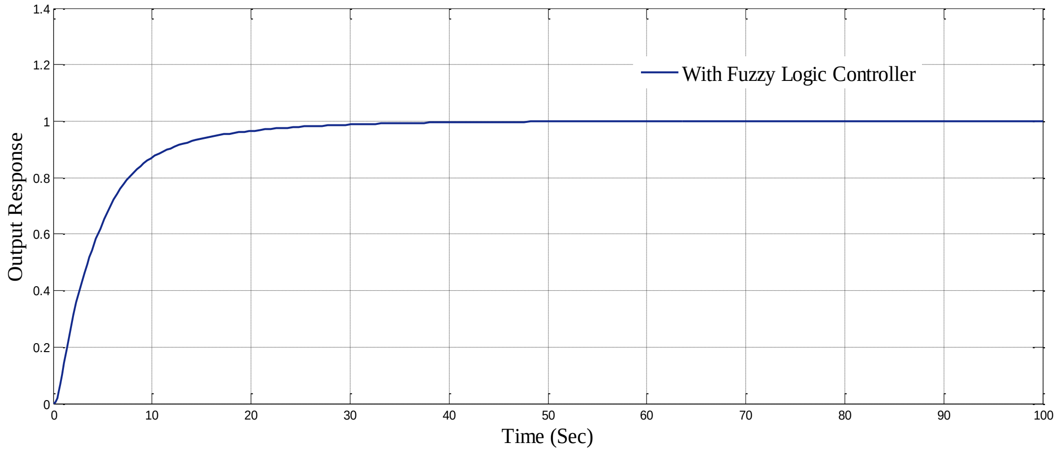


*Figure 11 Simulation result using Fuzzy Logic Controller*

## C. Simulation Result Using PID-FLC

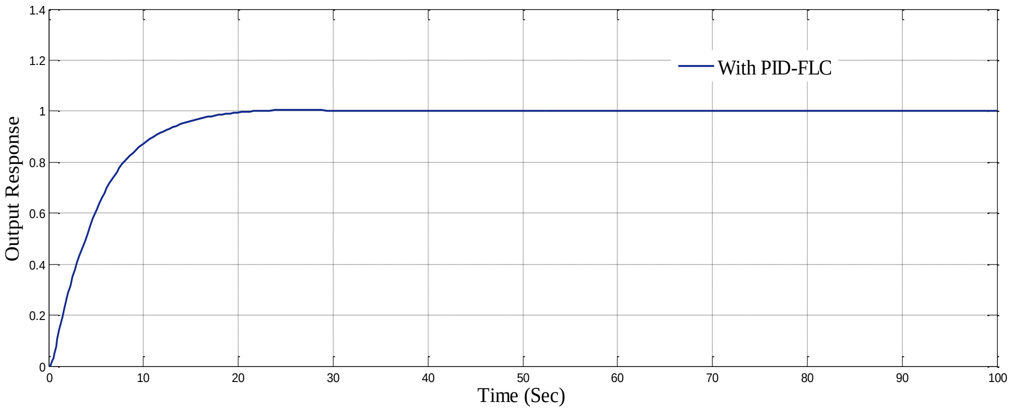


*Figure 12 Simulation result using PID-FLC*

From figure 12 PID-FLC provide good performance in terms of oscillations, settling time and overshoot in the absence of a prediction mechanism.

## D. Discussion

For comparison purposes, simulation plots include a Ziegler-Nichols, and the fuzzy algorithm. It can be seen that the PID-FLC provide good and satisfactory time domain response performance in terms of oscillations and overshoot are quite absence due to prediction mechanism. The PID-FLC algorithm adapts quickly to longer time delays and provides a stable response while the Ziegler-Nichols may drive the system unstable due to mismatch error generated by the inaccurate time delay parameter used in the plant model.

After comparison we find that the PID-FLC significantly reduced overshoot and steady state error. Comparison results of Ziegler-Nichols, FLC & PID-FLC are shown in Table 5.

*Table 5 Comparison table of Ziegler-Nichols, FLC & PID-FLC*

| Parameters | Ziegler-Nichols | FLC | PID-FLC |
|---|---|---|---|
| Overshoot | Present | Not Present | Not Present |
| Settling Time | Less | More | Medium |
| Transient | Present | Not Present | Not Present |
| Rise Time | Less | More | Medium |
| Steady State Error | Present | Not Present | Not Present |

 

*Volume 02, Issue 06, [June – 2015]* **ISSN (Online):2349–9745; ISSN (Print):2393-8161**

## VI. CONCLUSIONS

The proposed approach for designing PID-FLC in this paper gives valid results. The conventional Ziegler-Nichols with fixed gain parameters cannot satisfy this kind of requirements. PID-FLC which can self-tune the values of the gain parameters has been successfully presented in this paper for the set point weight tuning. The simulation results show that the PID-FLC system has a faster response, a lower transient overshooting, and a better dynamic performance than the conventional Ziegler-Nichols and FLC.

The simulation results also conclude that the proposed PID-FLC can also be replaced with Ziegler-Nichols and FLC. PID-FLC is easy to implement than Ziegler-Nichols. In this context, Fuzzy Set Point Weighting Controller appears superior to others, as it guarantees, in general, very good performances in the set point and load disturbance step responses and it requires a modest implementation effort, therefore its practical implementation in industrial environments appears to be very promising.

## REFERENCES

[1] K. Astr¨om and T. H¨agglund, PID Controllers: Theory, Design and Tuning. New York: ISA, 1995.
[2] N.Sianandam, S.Sumathi and S.N.Deepa, “Introduction to Fuzzy Logic using MATLAB(Springer,2007)”.
[3] Visioli.A. (2000) Tuning of PID controllers with Fuzzy Logic
[4] Visioli, A. (2004) Fuzzy logic based setpoint weighting for PID controlers.
[5] Based Tuning of PID Controller Parameters”, NationalConference on New Frontiers for Women in Science and Technology March 20- 21, 2013, JiwajiUniversity, Gwalior (M.P.).
[6] IEEE MELECON 2002, May 7-9, 2002, Cairo, EGYPT “Nonlinear PID controller using Fuzzy logic” Abdel-Azim S.Ibrahi
[7] 2010 International Conference on Computer Design And Applications (ICCDA 2010) “Design and Simulation of a Fuzzy-PID Composite Parameters' Controller with MATLAB” Changhua Lu and Jing Zhang.
[8] Dharamniwas , Aziz Ahmad , Varun Redhu and Umesh Gupta," Loquid Level Control by using Fuzzy Logic Controller", International Journal of Advances in Engineering & Technology, July 2012. Vol. 4, Issue 1,pp.537-549.

 