A Comparative Analysis of Optimized PID Controller Tuning using conventional and soft computing Techniques

[1]Eneh J. N and [2*]Abawulor Andrew Uko
Department of Electronic Engineering, University of Nigeria, Nsukka.
[1]nnenna.eneh@unn.edu.ng, [2]andrew.abawulor.pg02935@unn.edu.ng
*Corresponding author

Abstract

PID controllers have been used for industrial processes for long owing to its simple structure, strong robustness, and easy realization. This paper presents a review of the Soft computing methods as well as the conventional techniques used for PID controller tuning. The work analyzed how these various tuning methods affect the rise time, peak overshoot, settling time and overall Stability of the system. A comparison between some of the techniques was done and it was observed that the soft computing methods optimize the performance of the PID controller more than the conventional techniques. This work therefore provides a comprehensive reference source for works in PID controllers and it's tuning in general.

Keywords: *PID Controllers; Tuning; Convention Techniques; soft computing Techniques*

Introduction

The Proportional Integral Derivative (PID) controller is the most widely used and most mature in industrial production. So far, the vast majority of industrial controllers are PID controllers owing to the advantages of simple structure, strong robustness, and easy realization. However, once the controller characteristics, control scheme, interference form, and size are fixed, the quality of the control system depends on the setting of the controller parameters. Therefore, PID parameter tuning and optimizing is one core issue for PID controller design.

The parameter tuning methods for PID controller design fall into two basic categories: conventional parameter tuning methods and soft computing optimization algorithms.

Overview of PID

The general equation of PID controller is

$$U(t) = K_p\, e(t) + K_t \int e(t)dt + K_d \frac{de(t)}{dt}$$

Where,
K_p = proportional gain
K_t = integral time
K_d = derivative time

The variable $e(t)$ represents the tracking error which is the difference between the process variable signal (desired input value) and the set point signal(actual output). This error signal will be sent to the PID controller and the controller computes both the derivative and the integral of this error signal.

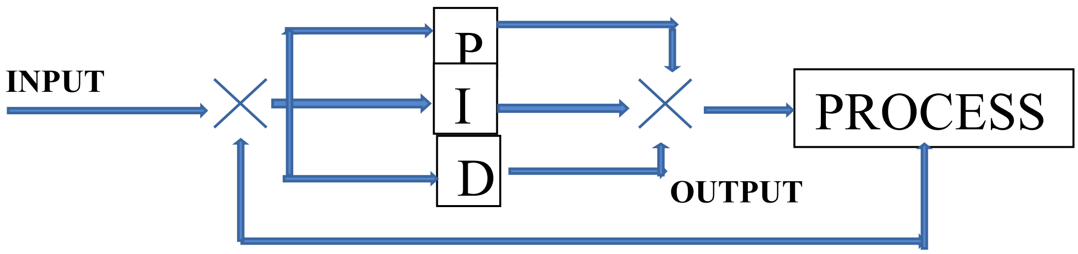

Fig1: Flow chart of a generalized PID controller

PID Controller Working Principles

The PID controller uses the control algorithm as three modes, i.e., proportional + integration + derivative.The proportional term applies appropriate proportional changes for error (which is the difference between the set point and process variable) to the control output. In fact, many control applications work quite well with only proportional control. The integral term examines the process variable over time, offset of set point, and then corrects the output if necessary. Derivative control monitors the rate of change of process variable and accordingly changes the output when there are unusual changes. User to get the desired performance from the process adjusts each parameter of the three control functions. Let us see the individual responses of PID controllers.

Proportional, P Action

It is the most common of all industrial process control action. It calculates the difference between the process variable signal and the set point signal, called as an error.

It is the measure of how far a process variable is deviating from a set point and calculated either as SP – PV or PV – SP depending on whether controller will be direct acting or reverse action.

The process, sensing device and final control element, determines this direction of action. The output of the proportional controller is the multiplication of error signal by a constant (also called as gain). In some cases, bias is also added to the p-control output as shown in figure below.

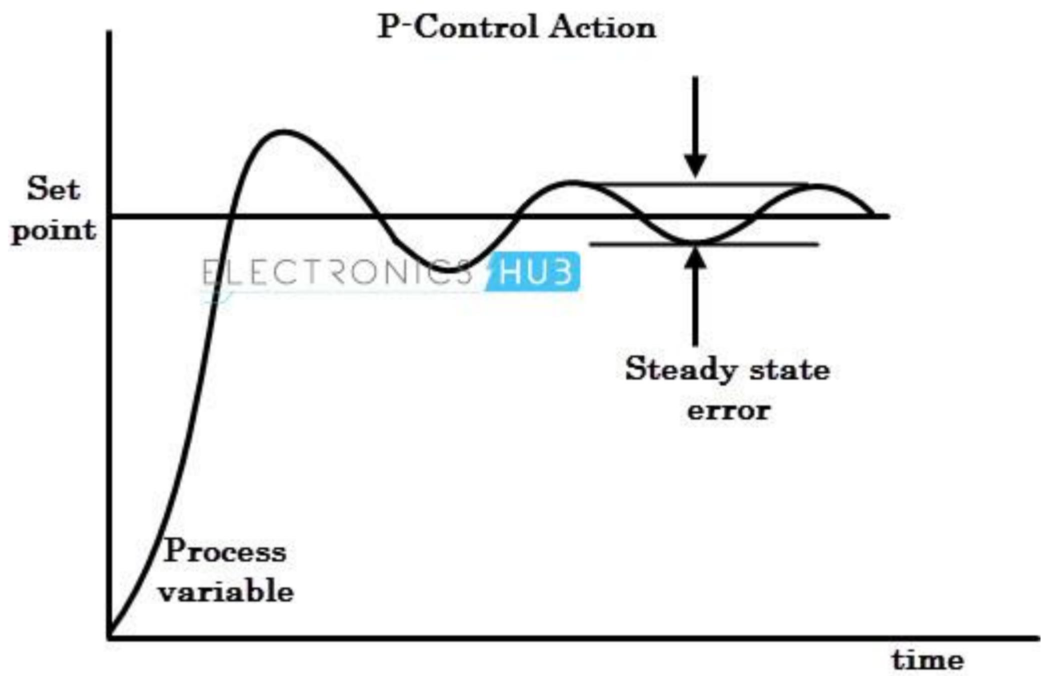

Fig 2: P Control Action Graph

The controller output increases as the error increase and as long as the error remains, it continues to generate corrective effort. As the error is zero, it produces zero output (if no bias is added at output).

If the controller gain is increased, it moves the output so rapidly for any given change in PV or SP.

However, too much control gain can result to an unstable control system. As a result, there exists a steady state error between the set point and the process variable.

Integral, I Action

It is the process of accumulating the process variable value as the time progresses. Integral action decides how fast to move the output. It is mainly used to eliminate steady state error of the system.

The controller output depends on the integral of error signal over time. It's not usually used alone, but mostly it is used along with proportional control.

Here the integration symbol indicates that the controller will accumulate multiple products of error over a small time dt.

When an error signal appears, the controller acts such that the proportional control signal returns the process to the desired control point and it is fast acting and immediate.[10][11]

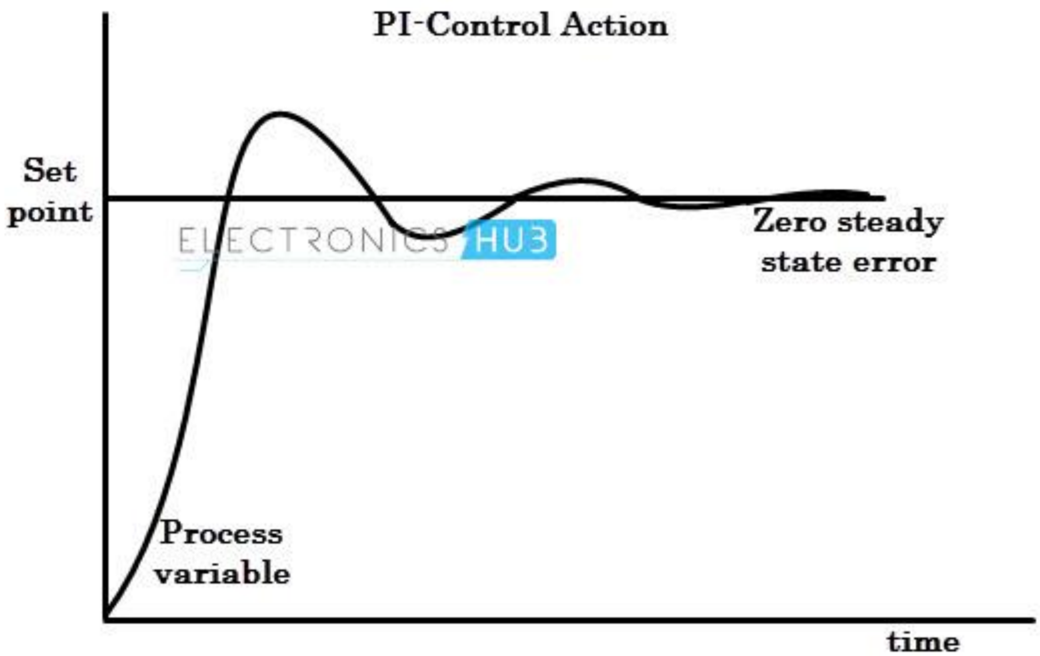

Fig 3: PI Control Action Graph

If there is any deviation between set point and process variable, an additional corrective signal is supplied by the integral control mode function till steady state error becomes zero.

At this situation, controller holds the previous value to maintain final control device such that zero steady state error is achieved. The steady state performance is improved by decreasing the integral gain Ki.

Most of the industrial process controllers are designed to operate proportional plus integral control to achieve an improved steady state response.

Derivative, D Action

This action senses the rate of change of process variable and then applies the corrective action at a proportional rate of change. It looks at how fast the process variable changes per unit time and takes action proportional to its rate of change.

This is also called as anticipatory action because it moves the control valve in such a direction as to counteract the rapid change of process variable.

The output of derivative action is the product of derivative constant and the rate of change of error with respect to time.

It must not be used alone because of rapid start of control output that can result in extremely large rate of change of output, even for a small error change.

Most times, it is used along with proportional plus integral control or with only proportional control. In order to avoid the extreme increase of control output for sudden changes of set points, many PID controllers offer derivative response based on the rate of change process variable only, rather than error (SP-PV or PV-SP).

By combining all three actions describe above, a PID action is obtained. It is the most often used controller in many industrial applications.

This type of controller gets the set point from the user, and gets the process variable from various sensing devices or transmitters.

Depending on the parameter setting in PID equation (i.e., Kp, Ki and Kd), it produces the control output to make the correction promptly and accurately to the set point value. The response curve for the PID controller is shown below.

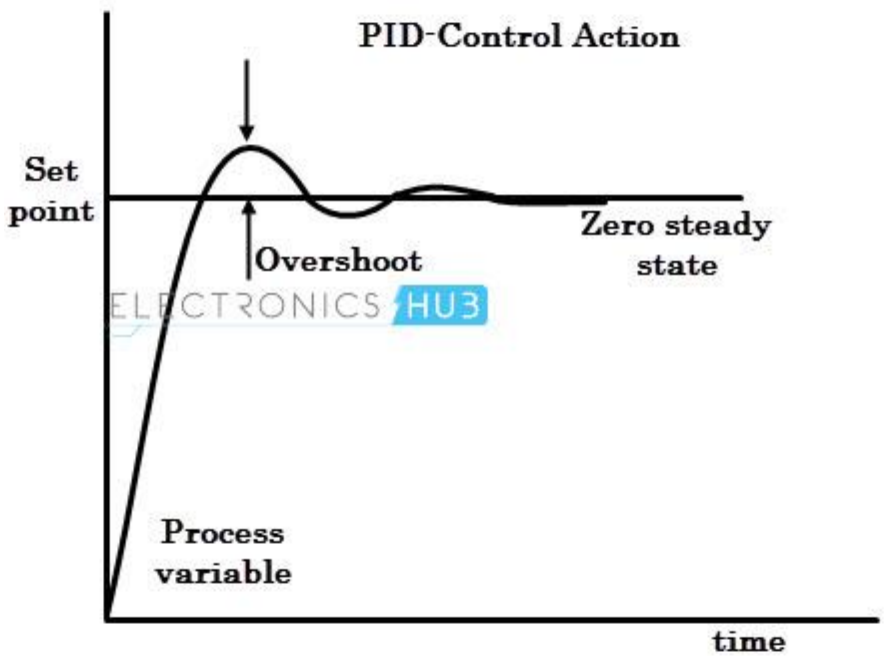

Fig 4: PID control action graph

PID Tuning Method

The determination of corresponding PID parameter values for getting the optimum performance from the process is called tuning. This is obviously a crucial part in case of all closed loop control systems.

A change in the proportionality constants of these terms changes the type of response of the system. That is why PID tuning, which is the variation of the PID proportionality constants, is of utmost importance.

This paper talks about the different types of PID tuning techniques implemented and the comparison between some of them.

There have been various types of techniques applied for PID tuning, one of the earliest being the Ziegler Nichols technique. These techniques can be broadly classified as traditional or classical and soft computational or optimization techniques.

Conventional/ Traditional Techniques

Conventional techniques make certain assumptions about the plant and the desired output and try to obtain analytically, or graphically some feature of the process that is then used to decide the controller settings. These techniques are computationally very fast and simple to implement, and are good as a first iteration. But due to the assumptions made, the controller settings usually do not give the desired results directly and further tuning is required. Some optimal conventional techniques have been reviewed in this paper. Conventional computing/Hard computing requires exact mathematical model and lot of computation time.

A. Ziegler Nichols Method

This is by far the most popular tuning method in use. It was proposed by John Ziegler and Nathaniel Nichols in 1942 and is still a simple, fairly effective PID tuning method.
The Ziegler and Nichols method is the first PID tuning techniques made and they are made based on certain controller assumptions. Hence, there is always a requirement for further tuning; because the controller settings derived are rather aggressive and thus result in excessive overshoot and oscillatory response. Also the parameters are rather difficult to estimate in noisy environment. Furthermore, the system is driven towards instability for determining the parameters, practically this can be quite detrimental to the system.

B. Cohen Coon Method

Cohen and Coon design a method with the PID controller parameters decided based on a FOLPD model. The main design requirement is the rejection of load disturbances. Despite a better model, the results of the Cohen Coon method are not much better than the Ziegler Nichols method.

C. Tyreus – Luyben Method

The Tyreus-Luyben procedure is quite similar to the Ziegler–Nichols method but the final controller settings are different. Also this method only proposes settings for PI and PID controllers. These settings that are based on ultimate gain and like Z-N method this method is time consuming and forces the system to margin if unstable. Some other algorithms have been proposed to solve these problems by obtaining critical data (ultimate gain and frequency) under more acceptable conditions.

D. C-H-R Method

This method was proposed by Chien, Hrones and Reswich is a modification of open loop Ziegler and Nichols method. They proposed to use “quickest response without overshoot” or “quickest response with 20% overshoot” as design criterion.

They also made the important observation that tuning for set point responses and load disturbance responses are different. To tune the controller according to the CH-R method the parameters of first order plus dead time model are determined in the same manner of the Z-N method. The tuning rules based on the 20% overshoot design criterion is quite similar to the Z-N method. However, when the 20% overshoot criteria are used, the gain and the derivative time are smaller and the integral time is larger. This means that the proportional action and the integral action, as well as the derivative action, are smaller.

Soft computing techniques

Soft Computing is an evolving collection of artificial intelligence methodologies aiming to exploit the tolerance for imprecision and uncertainty that is inherent in human thinking and in real life problems, to deliver robust, efficient and optimal solutions and to further explore and capture the available design knowledge. Soft computing utilizes computation, reasoning and inference to reduce computational cost by exploiting tolerance for imprecision, uncertainty, partial truth and approximation. Numerous Soft Computing-based methods and applications have been reported in the literature in a variety of scientific domains. Real world problems have to deal with systems which are non-linear, time-varying in nature with uncertainty and high complexity. The computing of such systems is study of algorithmic processes which describe and transform information:their theory, analysis, design, efficiency, implementation, and application. Conventional computing/ traditional computing requires exact mathematical model and lot of computation time, for such problems, methods which are computationally intelligent, possess human like expertise and can adapt to the changing environment, can be used effectively and efficiently. Soft computing techniques for a PID controller considerably reduced the overshoot and rise time as compared to any other PID controller tuning algorithms. We will also try to discuss some of these tuning methods here too.

A. Genetic Algorithm

Genetic algorithm (GA) is a search algorithm that explores the search space in a manner analogous to evolution in nature. It uses probabilistic rules to search for and changes the potential solutions in the search space, using a cost function to analyze the fitness of solutions. GA requires the solution to be represented in a way that is analogous to genes so that the processes that bring about a change in the genes (like mutation) can be used. Usually this is done by representing the solutions in a binary format. The standard genetic algorithm is given below.

• Initialization, firstly initial solutions are randomly selected from the search space.

• Selection, during each iteration, a proportion of solutions is selected, based on the fitness function (fitter solutions are more likely to get selected), for breeding the next generation of solutions. The selection is done in a probabilistic manner.

• Reproduction, selected solutions are paired up and crossover and mutation operation are performed to get the next generation of solutions.

• Termination, the iterations are terminated when the termination condition (time or accuracy) is reached.

Advantages of GA-PID controller

i) The algorithm is simple and can easily be understood and implemented.

ii) Its robust.
iii) GA is a non-linear process that could be applied to most industrial processes with good results.
iv) GA searches a population of points instead of a single solution.
v) GA does not need information about the system except for the fitness function.

Shortcomings of GA-PID controller

For a GA-PID controller, it cannot be guaranteed that the result obtained through the process is the most optimized values although it's near optimum. GA can have different result for each new search for the same system under same conditions. In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. Therefore, the result may not be the perfectly optimized one.

B. Particle Swarm Optimization (PSO)

PSO is one of the optimization techniques and a kind of evolutionary computation technique. The technique is derived from research on swarm such as bird flocking and fish schooling. In the PSO algorithm, instead of using evolutionary operators such as mutation and crossover to manipulate algorithms, for a d-variable optimization Problem, a flock of particles is put into the d-dimensional Search space with randomly chosen velocities and positions knowing their best values.

So far (p best) and the position in the d- dimensional space3. The velocity of each particle is adjusted accordingly to its own flying experience and the other particles flying experiences.

C. Evolutionary Programming

Generally, the EP algorithm for global optimization contains four parts, initialization, mutation, competition, and reproduction. Mutation is based on the current values and a Gaussian random variable. Furthermore, a quasi-random sequence (QRS) is used to generate an initial population for EP to avoid causing clustering around an arbitrary local optimum. Evolutionary programming was used for PID tuning using IAE and compared with results of which were fuzzy logic based and the results were superior for evolutionary programming and the same as results with a genetic algorithm.

Mathematical Modelling of a Dc Servo System

A DC servomotor has two main components, electrical component which consists of resistance, inductance, input voltage and the back electromotive force. The second component is the mechanical part that consists of motor's shaft, inertia of the motor and load inertia and damping. It can be considered as a linear SISO system having third order transfer function. The theoretical or White - box model which consists of differential and algebraic equations and can be written as a transfer function for the Single - Input -Single - Output (SISO) linear case is used in this work. The successful development of a theoretical model relies on the availability of good process information.

$$\frac{\theta(s)}{V_a(s)} = \frac{K_t}{[L_a.J_m.s^3 + (R_a.J_m + L_aB_m).s^2 + (R_a.B_m + K_b.K_t).s]}$$

Specification of field controlled dc servomotor

A 3.70 kW, 240v, 1750-rpm DC motor with the below parameters was used:
$R_a = 11.2\Omega$
$L_a = 0.1215$ H
$B_m = 0.002953$ Nms/rad
$J_m = 0.02215$ Kgm^2
$K_i = 1.08$ Nm/A
$K_b = 1.08$ Vs/rad

Result comparison of various pid controller tuning algorithms

Below is a combined step input response of ZN, PSO,EP and GA algorithms of PID tuned controller and the plant response.

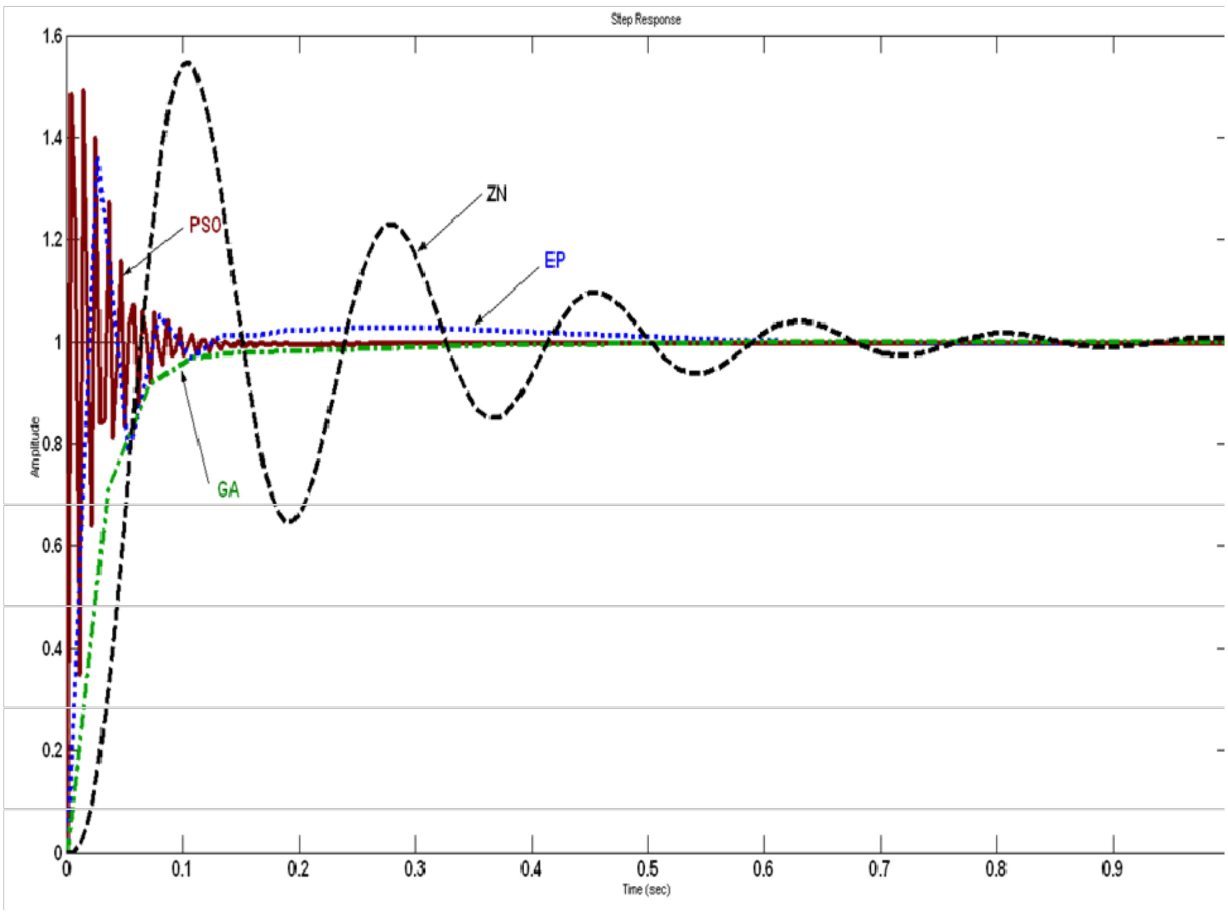

Fig 5: Graph showing comparison of various pid controller tuning algorithms

Conclusions

All the tuning methods have their merits and demerits. However, for a more stable plant with little or small overshoot needs, the soft computing methods were better than the conventional method. In this work, the Ziegler Nichols tuning method, which is the commonly used conventional method, was deployed for the comparison with different soft computing methods such as PSO, EP and GA. However, the main metrics of importance here is the outputting features such as the rise time, settling time and the peak overshoot of the servo system in use. As much as the soft computing methods provided a better response to the conventional methods, the best among the soft computing method was not be immediately established due to different plant behaviors.

References

1. Carriere, A., & Perriere, J.(2013).A Two-Step automatic Tuning Procedure for PID controllers. *IEEE Conference on control applications, 134*(3*),*13-16.
2. Karimi, A., Garcia, D, & Longchamp, R.(2003). PID controller tuning using Bode's integrals. *IEEE Trans. Control Syst. Technol., 11*(2)*,* 812–821.
3. Nagraj, B., Subha, S., & Rampriya, B.(2014).Tuning Algorithms for PID Controller using Soft Computing Techniques", *IJCSNS, 2,*278-281.
4. Bhandare, D. S. & Kulkarni, N. R.(2015). Performances evaluation and comparison of PID controller and fuzzy logic controller for process liquid level control. *International Conference on Control,Automation and Systems (ICCAS), 15*(1)*,*1347–1352.
5. Strom, K. J. A. (2013). PID controllers: Theory, design and tuning. *Instrument society of America.187*(5)*,* 13-16.
6. Barros, R. C., Santos, et al.(2016).Case study: Level system control performed with LPC2148 Microcontroller via PC platform. *International ConferenceOn System Theory, Control and Computing (ICSTCC)*. 56(5), 257–262.
7. Zhang, Y.,Zhang, L., & Dong Z. (2019). An MEA-Tuning Method for Design of the PID Controller. *Math. Probl. Eng. 1*(1)*,*1–11.
8. Rathore, N. S. & Singh, V. P.(2016).Tuning of PID Controller for Position Control of DC Servo Motor using Luus-Jaakola Optimization. *IEEE Int. Conf. Comput. Commun. Control 1*(8)*,* 1–6.
9. Reis, V. C. , Santos, M. F., Carmo, M. J. & Ferreira, F. C.(2016).Controlof level systems by arduino via PC platform. *InternationalConference on System Theory, Control and Computing (ICSTCC)*,*24*(9),251–256.
10. Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automaticControllers. *Trans. ASME, 64*(1)*,* 759–768.
11. Chien, K. L.,Hrones, J. A., & Reswick, J. B. (1952). On the automaticcontrol of generalized passive systems, *Trans. ASME, 74*(1)*,* 175–185.
12. Vaishnav, S. R. & Khan, Z. J. (2010). Performance of tuned PID controller and a new hybrid fuzzy PD controller.*World J. Model. Simul., 6, 141–149.*
13. Kaur, M.(2017).Literature Review of PID Controller based on Various Soft Computing Techniques," *Int. J. Recent Innov. Trends Comput. Commun.*, 5(2), 5–8.
14. Sen, R., Pati, C., Dutta, S., & Sen, R.(2015) Comparison Between Three Tuning Methods of PID Control for High Precision Positioning Stage. *MAPAN-Journal Metrol. Soc. India APAN-Journal Metrol. Soc. India*, 30, 65–70.

15. Urrea-Quintero, J.-H.(2018).Optimum PI / PID PI / PID Controllers Tuning via an Evolutionary Algorithm," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, 2, 20–30.

16. Kushwah, M, & Ashis Patra. (2014).Tuning PID controller for Speed Control of Dc motor using Soft Computing techniques-a review." *Advance in Electronic and Electric Engineering* 4.2, 141-148.