Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPUTATIONAL AND INTELLIGENT OPTIMIZATION TUNING METHOD FOR PID CONTROLLER

ROSALIA H. SUBRATA[1], FERRIANTO GOZALI[2], ENDANG DJUANA[3]
[1,2,3]Electrical Engineering Department, Faculty of Industrial Technology, Universitas Trisakti, Jakarta, Indonesia
E-mail: [1]rosalia@trisakti.ac.id, [2]ferrianto@trisakti.ac.id, [3]edjuana@trisakti.ac.id

ABSTRACT

Proportional Integral Derivative (PID) control systems are quite common in the commercial, institutional, industrial plants and many real-life applications due to their advantages over other kind of controllers. Proper application and tuning of PID parameters can provide many benefits and increase process efficiency. However, improper implementation, due to lack of understanding and poor tuning processes, are often be the main cause behind many problems. Conventional PID tuning methods can't meet the demands as the system becomes more and more complex due to trial and error approach, therefore a more systematic and automated approach is needed. This paper discusses the background and theory behind PID controller and bring a clear understanding how to perform tuning process for PID controllers and presents computational and intelligent optimization techniques such as Genetic Algorithms, Particle Swarm Optimization and Differential Evolution used to make the control deviation of step response close to zero, have faster response and smaller or even without overshoot. Through literature survey from existing publications, several issues related to the implementation of the PID controller to enhance the performance of such as model plant development, parameter setting, control characteristics expected and tuning processes are discussed and summarized. The conclusion is that computational and intelligent optimization tuning method produce less overshoot, shorter rise and settling time and less errors in system responses, therefore producing more optimal tuning and at the end more optimal system performance.

Keywords: *PID Parameter, Computational Method, Tuning Optimization.*

1. INTRODUCTION

After the discovery of PID control by Elmer Sperry in 1910 [1], and tuning method of PID proposed by J. G. Ziegler – N. B. Nichols (Z-N) in 1942 [2], the PID control is growing rapidly. Now, it is common to find many PID controllers or modified PID controllers used as industrial controls. Many tuning methods have been suggested in theory and implemented on-site manually or automatically. However, most of industrial controllers are still implemented based on and around classical PID algorithms [1].

In the field of process control systems, plant's mathematical models are very essential for analytical design methods. But when the analytical design methods cannot be used, PID control proves useful. The basic PID controller or a modified PID controller proved useful in providing satisfactory control, but it is not possible to provide optimal control for all conditions [3].

We can say that process control deals with the process of maintaining the output of a specific process in industrial plant within a desired range to maintain the quality of an end product. In most of industrial process, it is very important to maintain the control of the performance in order to keep the operations running under certain predefined condition and to set more precise limits to maximize the quality of the product and safety.

However, usually it is difficult to achieve desired performance for time-variant system. Therefore, it is needed that control parameters of the process are adjusted automatically by the controller itself and become adaptive to the behavior of the process. In steady state, control parameters are not needed to be adjusted in every step of the process but only when control parameters become out of the range defined and the performance is insufficient [4]. The adaptivity of the controllers depend on how the controllers utilize the performance evaluation of the process and adjust the control design system.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

As PID controllers work based on combination of suitable P, I and D parameters for each system in order to produce optimal system performance, these parameters (P, I and D) need to be tuned, a process known as PID tuning. PID tuning means that the control loop has a specific goal which is achieved by using the right P, I, and D parameters. PID tuning is necessary in order to have the desired closed-loop control. PID tuning can result in many benefits such as the process can operate in a stable way, reduce oscillations, reducing energy consumptions and stability of the system can be achieved with a minimum amount of operator interventions.

Nevertheless, conventional PID parameters tuning methods have limitation in that they may produce suboptimal performance, for which further adjustment of PID parameters is needed. With the advancement of computational methods, many optimization techniques in engineering are used in PID tuning to improve the capabilities of conventional PID parameter tuning methods. Various meta-heuristic methods used in Artificial Intelligence such as Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO) have been used for determining the optimal PID tuning. These methods can produce accurate PID parameters therefore producing optimal system performance.

In this paper a number of methods used for PID tuning are reviewed. We divided the tuning methods into two categories: Conventional Tuning Methods and Computational and Intelligent Optimization Methods. A brief explanation of each method is described then followed by examples of a case study of the tunings that have been carried out previously, and presents the simulation results of the implementation of PID tunings on the control system process using these methods. Several examples of case studies carry out comparative analysis, and explain the comparison results.

2. PID CONTROLLER

Proportional Integral and Derivative or abbreviated PID, and it is one kind of approach used to control different process variables in industrial applications. The PID controller block diagram of a plant is shown in Figure 1. These control elements can perform individually or collectively, for example, a P-controller, a PI controller, or a PID controller. Each control parameter can be adjusted and controlled individually and each control element can be used for specific purposes depending on the application. The transfer function of a standard PID controller usually written in "parallel form" or in "ideal form" can be seen in eq. (1) and eq. (2) [1]:

$$G(s) = K_p + K_i \frac{1}{s} + K_d s \quad (1)$$

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (2)$$

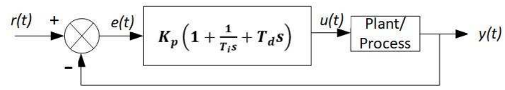

Figure 1: PID Controller of a Plant

Output of the PID controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (3)$$

with K_p , K_i , K_d positive.

$K_i = K_p \frac{1}{T_i}$ and $K_d = K_p T_d$.

Eq (3) can be written as follows [3]:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \quad (4)$$

where $u(t)$ = output controller; $e(t)$ = error signal; K_p = proportional gain; K_i = integral gain; K_d = derivative gain; T_i = integral time; T_d = derivative time.

The Proportional also called P-controller will produce an output proportional to the error signal $e(t)$. This controller always produces a system steady-state errors but it can provide stable operation. By improving the proportional constant K_p, the speed of the system response will improve. The P-controller needs a manual reset if used independently because it never gets to a system steady state. However, if the proportional constant K_p is too large, the system will take a long time to reach steady-state condition. The Proportional controller equation is formulated as [3]:

$$u(t) = K_p e(t) \quad (5)$$

The Integral also called I-controller is needed to eliminate steady-state error and this controller can integrate the error over a period of time so that the error value becomes zero. The integral controller equation is formulated as [3]:

$$u(t) = K_i \int_0^t e(t)dt \quad (6)$$

If a negative error occurs, the controller output will decrease, meaning the response speed is reduced and the system stability is disturbed. To increase the response speed, the integral gain, K_i is reduced.

The Derivatives also called D-controller will give a kick start at the output so as to improve system response. The output produced by this controller depends on the rate of change of the error with respect to time. The D-controller can anticipate future error behavior unlike the I-controller. To increase the speed of system response by increasing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the derivative gain K_d . The derivative control equation is formulated as follows [3]:

$$u(t) = K_d \frac{de(t)}{dt} \quad (7)$$

The effect of each controller parameter P, I, D independently on closed-loop performance is described in Table 1 while for open-loop performance this table is useful as a first reference for stability. However, K_p, K_i (or T_i) and K_d (or T_d) are mutually dependent in tuning for achieving optimum performance [1].

Table 1: Effects of parameter P, I, and D independently

Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-state Error	Stability
Increasing K_p	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_i	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_d	Small Decrease	Decrease	Decrease	Minor Change	Improve

So finally, by combining these three parameters, the desired response for the system can be achieved.

2.1 PID Controller Design

What needs to be done in designing a PID controller in a plant is first to derive the plant's mathematical model. In reality, it is not easy to derive the mathematical model of a complicated plant. In this situation, an experimental approach is performed to improve the controller design. Then select and apply various design techniques so that the system achieves the desired criteria. To produce the desired transient and steady state characteristics, needs to determine the appropriate control parameters. The process of setting the controller parameters in order to produce the desired system criteria is called controller tuning.

PID controllers are most often used by practitioners in designing control systems because the PID technique is quite mature and well understood. PID parameter tuning is very important to obtain the desired system characteristics. By tuning the PID parameters correctly, it can improve slow system response, reduce large oscillations, reduce steady state errors and improve system stability. However, there is no single tuning strategy that applies to all kinds of control systems, so no single tuning method which able to tune all kinds of loops optimally.

Ziegler and Nichols found a rule to adjust the PID controller when the mathematical model cannot be known. The Ziegler-Nichols rule consists of a set of values that will give stable system characteristics. If the system provides conditions that are not as desired, such as too large an overshoot in the step response, a fine tuning must be carried out until the desired condition is reached [2]. In general, the Ziegler Nichols tuning rule produces the value of the integral time, proportional gain and derivative time parameters based on the transient response characteristics of the plant as a starting point for fine tuning [3].

3. PID TUNING METHOD

As stated before, PID tuning relates to processes of determining PID parameters based on the method used to produce the engineering specifications and desired closed-loop performance characteristics. The best plant optimization can be accomplished if the closed-loop control able to control the process stability, reduce plant oscillations, efficiency and minimize energy consumption.

A number of conventional tuning methods are done manually, difficult and take a lot of time. In order to get the ideal PID parameters efficiently, it is necessary to tune the optimal parameters. There are many methods of PID tuning. In this paper, PID tuning methods are divided into two categories: 1). Conventional Tuning Methods and 2). Computational and Intelligent Optimization Methods.

3.1 Conventional Tuning Methods

The steps to determine the PID controller parameters with conventional methods, first make assumptions about the desired plant output, conduct an analytical or graphical analysis, then determine some features of the process. Tuning the controller parameters by this method in most cases does not give perfect results and further tuning is required because it is based on the assumptions made. Conventional PID tuning methods are briefly discussed in the following sections.

3.1.1 Ziegler-Nichols Method

The Ziegler–Nichols tuning formulas were developed empirically based on simulations of many existing cases. The Ziegler–Nichols tuning method developed by John G. Ziegler and Nathaniel B. Nichols in 1942 is a heuristic tuning of a PID controller. To discover the parameters of the PID controller, the following classical tuning methods are used: step response method and frequency response method. There are two parameters in these methods process dynamics, namely the parameter which represent the process gain and the parameter which represent how long the process is. These

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters are expressed by simple formulas. The Ziegler-Nichols tuning method is well implemented in systems with a small gain ratio and a smaller dead-time than the time constant. [5]. These methods are still widely used by controller manufacturers and process industry, both conventional methods and modified conventional methods. For some purposes, the result is acceptable but not optimal for all applications.

In the step response method, parameters are determined from the unit step response of the process. To approximate the characteristics of the system, an open-loop test was carried out. First, determine the point at which the slope of the step response reaches its maximum, and then draw a tangent to this point. The intersection of the tangent and the coordinate axes gives the parameters a and L (measured on the step response of the process), as shown in Figure 2. The step response method can only be used for stable processes. This method provides PID parameters directly as function of a and L and the estimation of the period T_p of the closed-loop systems, as shown in Table 2 [5].

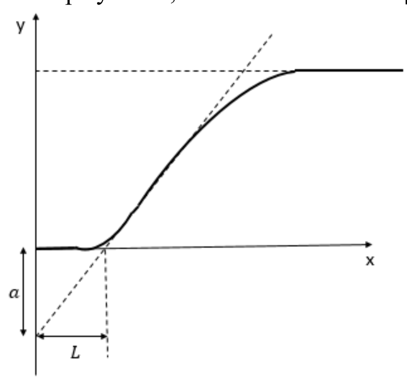

Figure 2: Ziegler-Nichols step response method Characterization

Table 2: PID parameters for the Ziegler-Nichols step response method

Controller	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9/a$	$3L$		$5.7L$
PID	$1.2/a$	$2L$	$0.5L$	$3.4L$

The parameters in the frequency response method named the ultimate gain (K_u) and the ultimate period (T_u). To determine the PID controller parameters, first adjust the parameters until the control action in proportional $T_i = \infty$ dan $T_d = 0$, then improve the gain so the system starts to oscillate. The gain obtained at this stage named K_u and the period of the oscillation is T_u. Table 3 shows the simple Ziegler-Nichols formulas for the PID controller parameters in terms of the K_u and the T_u [5]. T_p = period of the dominant dynamics of the closed-loop system. This method is suitable only for closed loop PID controller systems. The use of this method aims for the system to achieve stability quickly. The advantage of this Ziegler-Nichols method such as K_u and T_u parameters are simpler to find accurately than the parameters a and L in the step response method.

Table 3: The Ziegler-Nichols frequency response method PID controller parameters.

Controller	K	T_i	T_d	T_p
P	$0.5K_u$			T_u
PI	$0.4K_u$	$0.8T_u$		$1.4T_u$
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$0.85T_u$

The advantages of Ziegler-Nichols tuning method are its simplicity, intuitive, easy to use than other methods and requires only a few trials it is quite simple to generate the appropriate controller parameters while the disadvantages of this method are the tuning result is sensitive to non-linearities and approximations for the tuning parameters values might not be entirely accurate for different systems.

3.1.2 Cohen Coon Method

The Cohen-Coon tuning method is derived from the Ziegler-Nichols method, but in the process, it uses more information from the system so as to make the control performance much better. Cohen-Coon tuning method is classified as an offline tuning method. There are three parameters defined in this method, parameter a is steady state gain, L is time delay, and T is time constant.

Cohen-Coon tuning formulas as found in Table 4 were obtained based on analytical and numerical computational. The parameters $a = K_p L/T$ and $\tau = L/(L+T)$ are used in the table, T = apparent time constant, and τ = relative dead time [5]. The initial control parameters are measured based on the response of the system with the time constant and time delay as the output of a step change at steady state. The main design criterion in this method is rejection of load disturbances thus minimizes the steady state error [5].

The advantages of Cohen-Coon tuning method are reducing errors in approximating the steady state gain and the time delay by using the gain, good enough for systems with time delay, and response to closed loop is faster than Ziegler-Nichols

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method. The disadvantage of Cohen-Coon tuning method is that it cannot be used online.

Table 4: Cohen-Coon Method Controller Parameters.

Controller	K	T_i	T_d
P	$\frac{1}{a}\left(1+\frac{0.35\tau}{1-\tau}\right)$		
PI	$\frac{0.9}{a}\left(1+\frac{0.92\tau}{1-\tau}\right)$	$\frac{3.3-3.0\tau}{1+1.2\tau}L$	
PD	$\frac{1.24}{a}\left(1+\frac{0.13\tau}{1-\tau}\right)$		$\frac{0.27-0.36\tau}{1-0.87\tau}L$
PID	$\frac{1.35}{a}\left(1+\frac{0.18\tau}{1-\tau}\right)$	$\frac{2.5-2.0\tau}{1-0.39\tau}L$	$\frac{0.37-0.37\tau}{1-0.81\tau}L$

3.1.3 Chien, Hrones and Reswick (CHR) Method

The Ziegler-Nichols step response method was modified by Chien, Hrones and Reswick to make better damped closed-loop systems. The Chien-Hrones-Reswick (CHR) autotuning method focuses on setpoint response and disturbance response, the controller parameters formulas are given in Table 5 and Table 6 [5]. This method provides overshoot formulas for 0% and 20%. Determination of parameters a and L can be seen in sub chapter 3.1.1 Figure 2 of this article.

Table 5: Controller parameters of the CHR load disturbance response method.

Overshoot	0%			20%		
Controller	K	T_i	T_d	K	T_i	T_d
P	$0.3/a$			$0.7/a$		
PI	$0.6/a$	$4L$		$0.7/a$	$2.3L$	
PID	$0.95/a$	$2.4L$	$0.42L$	$1.2/a$	$2L$	$0.42L$

Parameters tuning formula for the 20% overshoot in Table 5 are similar to the Ziegler-Nichols step response method as found in Table 2. However, for the 0% overshoot, the gain and the derivative time are smaller and the integral time is larger. It can be said that the performance of proportional, integral and derivative, are smaller [5].

Table 6: Controller parameters of the CHR setpoint response method.

Overshoot	0%			20%		
Controller	K	T_i	T_d	K	T_i	T_d
P	$0.3/a$			$0.7/a$		
PI	$0.35/a$	$1.2T$		$0.6/a$	T	
PID	$0.6/a$	T	$0.5L$	$0.95/a$	$1.4T$	$0.47L$

3.1.4 Kappa-tau (KT) Method

Kappa-tau (KT) tuning method was designed to fill the gap of the Ziegler-Nichols method for systems with high proportional gains and the rules providing unsatisfactory performance for systems with long normalized dead time. Kappa-tau tuning method is based on the parameters K and τ and designed for load disturbance response. The method offers opportunity to distinguish between setpoint and disturbance response. The advantages of Kappa-tau tuning method are less oscillatory response and optimal for disturbance rejection with no overshoot while the disadvantages are it cannot achieve closed-loop performance characteristics in consequence of 0% overshoot.

3.1.5 Lambda Method

The Lambda tuning method was developed for systems with a long dead time L and includes an analytical tuning method of PID parameter. The time constant of the closed-loop response is expressed by the λ parameter. The response is assumed to follow a first order with L and λ as a tuning parameter. Lambda tuning is applicable only for PI controller, parameters is given in Table 7 [5].

Table 7: PI controller parameters of Lambda Tuning Method.

λ	K_c	T_i
PI	$\frac{T}{K_p(\lambda+L)}$	T

The advantages of Lambda tuning method are the possibility of choosing how fast the response of the controller, with a wide time delay and response with no overshoot while the disadvantages are minimal for disturbance rejection.

3.2 Computational and Intelligent Optimization Methods

With the advancement of computational methods, many optimization techniques in engineering are used in PID tuning to improve the capabilities of conventional PID parameter tuning methods. The main purpose is to make the control deviation of step response close to zero, have faster response and smaller or even without overshoot. Various meta-heuristic methods used in artificial Intelligence such as Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO) have been used for determining the optimal PID tuning of various systems. GA/DE/PSO is used to obtain PID parameters and an output signal in the context of system requirements [6]. A general schematic of the system can be seen in Figure 3.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

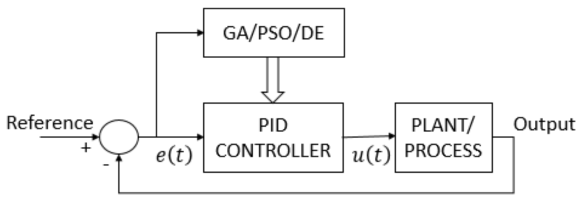

Figure 3: General schematic of the system

3.2.1 Genetic Algorithm (GA)

The Genetic Algorithm's concept was first found by J.H. Holland in 1970 and relies on natural evolution principles [7]. A genetic algorithm (GA) is a stochastic method for optimization problems based on a natural selection process that mimics biological evolution. Three basic operations or genetic operation in generic algorithm are 1) Selection and Reproduction, 2) Crossover and 3) Mutation. The application of these operations allows the creation of new chromosomes which represents a solution of the problem and may be better than their parents.

The GA procedure starts by selecting the right number of chromosomes from the initial population where each chromosome represents an adequate solution and is tested with a fitness function [7], [8]. The current population then undergoes an evolutionary process to produce a new chromosome called a new generation. Chromosomes in this new generation are better than the previous generation. This algorithm is repeated until it produces an optimal solution [9], [10]. The process flowchart of GA is presented in Figure 4 [9], [11].

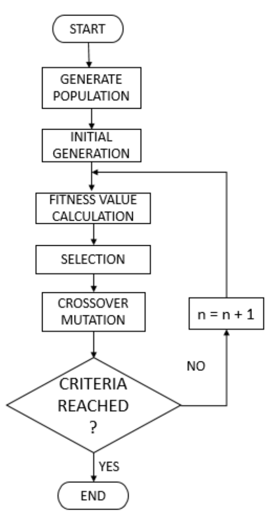

Figure 4: GA process flowchart.

PID parameter tuning process steps using GA:

1. Generate Population: Generate the initial population of a few chromosomes at random. Each chromosome comprises of three PID gains (K_p, K_d, K_i) with value bounds varied depend on the objective functions used and written as binary strings. The first and the most crucial step is to encoding the problem into suitable GA chromosomes. The more the chromosomes number, the better the chance to get the optimal results.

2. Initialization: Create generation of the initial population and fix in an explicit array using the Ziegler-Nichols method [9] to ensure the system stability and the convergence.

3. Fitness Value Calculation

There are two issues related to this step, the objective function and the fitness function. The objective function is useful for ensure the validity of the fitness function in each chromosome. The function is set according to the system requirements to get the optimal value. Several objective functions can be seen below:

a. Integral Absolute Magnitude of the Error:

$$IAE = \int_0^\tau |e(t)|\, dt \quad (8)$$

b. Integral of the Squared Error:

$$ISE = \int_0^\tau e(t)^2\, dt \quad (9)$$

c. Integral of Time multiplied by Absolute Error:

$$ITAE = \int_0^\tau t|e(t)|\, dt \quad (10)$$

d. Integral of Time multiplied by the Squared Error:

$$ITSE = \int_0^\tau te(t)^2\, dt \quad (11)$$

Fitness function: The implemented objective function is evaluated to reduce errors based on system requirements. The result of this evaluation is defined as a fitness function.

4. Selection: Selection is used to obtain chromosomes which will be selected for crossbreeding and mutation. The higher the fitness value of the chromosome, the greater the chance of being selected. When the process is complete all the chromosomes are established.

5. Crossover and Mutation: Crossover is the GA operator that collects two-bit strings as parent values to form a new chromosome of the highest quality. This operation is not performed on all existing chromosomes but chromosomes were randomized to perform crosses. If the crossover is not performed, then the parent value will be passed down to the offspring. A considerable population implicates a smaller crossover rate. Mutation is the next GA operator of that modifies the string codification (changes a few strings at random) of the selected chromosome and replaces the genes lost from the population due to the selection process that allows

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the reappearance of genes that did not appear in the initial population.

3.2.2 Particle Swarm Optimization (PSO)

The concept of Particle Swarm Optimization (PSO) technique was introduced by James Kennedy and Russell Eberhart in 1995 [12]. The PSO algorithm mimics the behavior of a herd insects or birds to take a chance and gain advantage while moving in certain space in finding food or avoiding predators. Each particle with its own two characteristics, namely its position and velocity, able to remember the best position ever traversed in the space. The interaction of particles in the PSO method will produce the best particles in order to obtain the optimal PID parameter specifications [13],[14].

There are 3 stages in the PSO algorithm: The first stage is to generate particle's positions and velocities of the initial population randomly and they move through the space at random speeds. The second stage is to update speeds of the particles for the next iteration, particles are able to remember every interaction, following their previous best position and equivalent fitness. Velocity update is based on the value of particle fitness which is not only determine which particle has the best global value in current herd, but also determine the best position of each particle over time. And finally, the third stage is position update. A particle with the best fitness and unique for the entire space solution, is called best global value of the swarm [12], [14], [15]. The cost function applied to achieve the optimal PID coefficient is ITAE as found in equation (10), because this function is useful for reducing errors (positive and negative), and giving small overshoot percentages [12].

The process flowchart of Particle Swarm Optimization is presented in Figure 5.

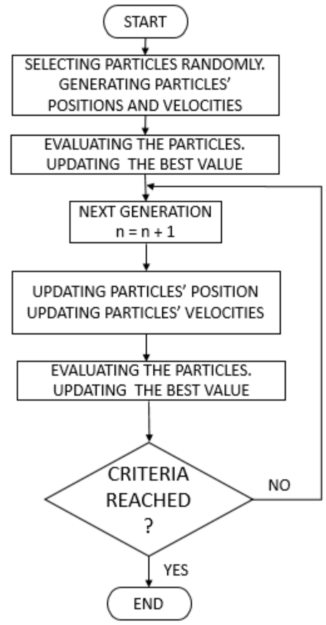

Figure 5: General Particle Swarm Optimization process flowchart

3.2.3 Differential Evolution (DE)

Differential Evolution (DE), proposed by Storn and Price [16] is a variant of computational algorithms used in optimization problems [17]. This algorithm is similar to GA and PSO, but Differential Evolution has some advantages such as easy and simple to use, the ability to locate the accurate global optimum irrespective of the initial parameter values, it has rapid convergence and utilizes few control parameters.

Differential Evolution is a population-based heuristic search algorithm in which the population of D-dimensional individuals is used to optimize a problem. Each individual in the population is a candidate solution to the problem and is coded as a vector. The parameters of the algorithms are the population in the generation G, the dimension of the problem D and the population size NP. DE is composed of four steps, which are initialization, mutation, crossover and selection as seen in Figure 6.

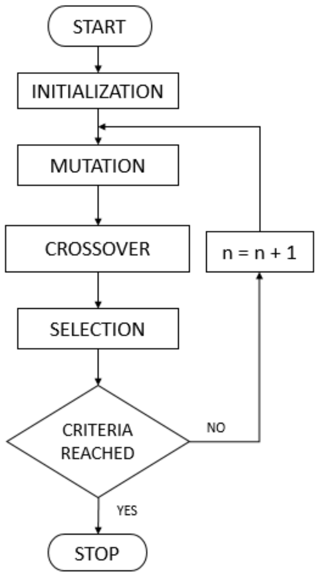

Figure 6: Flow graph of Differential Evolution algorithm.

Initialization: Three main parameters of the algorithm, population size (NP), mutation factor (F) and crossover rate (CR) are defined in the initialization step [18]. The setting of these parameters influences the algorithm's performance, and the optimal setting depends on the problem to be faced [19]. In general, these parameters are set after the objective function is designed [20]. The DE optimization starts with a randomly generated population of NP individuals characterized by D parameters according to a uniform distribution within the search space. Each individual represents a candidate solution to the optimization problem. After initialization, the DE algorithm performs mutation, crossover and finally selection operations iteratively until the user-defined criteria are reached. In this iterative process, new individuals are generated and evolved over generations. In this work, all generated individuals for the search space are feasible solutions since the problems are non-constrained.

Mutation: In each G-generation, the selected mutation strategy is used to create a mutation vector. The popular mutation strategy is DE/rand/1. In this strategy, two vectors (individuals) are randomly chosen, the sum of the weighted difference of two individuals is multiplied by a mutation factor F, and the result is added to a third random vector. The difference vector automatically adapts to the scale of the optimized function, and that is the key success factor of the DE algorithm [21]. There are also some popular mutation strategies such as DE/rand/2, DE/best/1, DE/best/2, DE/current-to-best/1, DE/current-to-pbest/1 and each strategy affects population diversity differently which provides a different rate of search convergency [22]. The mutation factor F defines the search step of the optimization. According to Storn and Price, F range is in [0, 2] [16]. Generally, smaller values of F are the better when the population is closer to the global best value. In contrast, larger values of F are preferred when the population is far from the global best value [23].

Crossover: The Crossover Operation is performed to increase the diversity of mutation vectors from previous operation. In this operation, the crossover procedure generates a trial vector between the interaction of the mutation vector and the target vector. There are some option popular crossover strategies, such as classic binary crossover strategy, exponential crossover strategy [24,25], eigenvector-based crossover strategy [26]. According to Storn and Price, the crossover rate CR range in [0,1] reflects the probability in which the trial vector inherits mutation vector's values [27] and has a direct influence on the diversity of the population [23].

Selection: The last process of the iteration is the selection process. After the crossover operation, the fitness value of the trial vector is calculated. Afterward, the trial vector and the target vector compete, and if trial vector has a better fitness value than target vector, then trial vector will replace target vector in the next generation.

4. RESULTS AND DISCUSSION

The implementation of many optimization techniques to improve the capabilities of conventional PID parameter tuning methods such as GA, PSO and DE can be found in various applications. Through literature survey from existing publications, several issues related to the implementation of the controller is described as follows.

GA based PID controller tuning has been applied to Continuous Stirred Tank Reactor process with a combination of ISE, IAE, and ITAE as the objective function. The simulation results have concluded that the optimal PID parameters obtained using GA have achieved satisfactory set point tracking and disturbance rejection in the process [28].

GA based PID Controller tuning is implemented and compared with the classical Ziegler-Nichols tuning method on a position control system of DC motor

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

order 3. The response produced by the PID controller with the GA method has a rise time and settling time much faster than produced by the classical method and the error produced by the GA is much smaller than the error produced by the classical method [29].

PSO-based Tuning of a PID Controller is implemented for a High-Performance Drilling Machine and is compared with the classical Ziegler-Nichols tuning method. The simulation results conclude that tuning PID parameters based on the PSO algorithm provides less overshoot, faster to reach settling time, rise time; peak time are faster; and reduced errors [30].

PID controller tuning using PSO method was also implemented on four different systems, 1) Linear hydraulic system to control the position and velocity of fluid flow in hydraulic cylinders driven by hydraulic pumps. 2) Linear electrical control with disturbance (D.C motor). 3) Linear thermal control system with delay. 4) Non-Linear Permanent Magnet Synchronous Motor. The simulation results showed that the PID controller tuned by PSO method, provides accurately the desired closed loop dynamics (overshoot, rise time, settling time, and steady state error) in a very short time for a small number of iterations. Overall, the PSO method could be considered as an effective and reliable auto tuning process for PID controllers [31].

A comparative study of the performance of PID controllers tuned with Genetic Algorithm, Particle Swarm Optimization and Hybrid Genetic Algorithm Particle Swarm Optimization applied to the Automatic Voltage Regulator system. The result showed that Genetic Algorithm has a lower fitness cost than the Particle Swarm Optimization method, Particle Swarm Optimization method is faster than the Genetic Algorithm, while the Hybrid Genetic Algorithm Particle Swarm Optimization method is more stable and effective [32].

A comparative study of the performance of PID controllers tuned by GA and PSO for DC motor position controllers concluded that GA method provides a slow response for the system while PSO method provides a better system response for the system with a lower percentage of overshoot. PSO based PID controller provides an ideal performance for position control of DC motor related to the system specifications [33].

A study compares DE algorithm and PSO algorithm for optimizing PID parameter tuning for Electrical Discharge Machining Systems. With IAE fitness function, DE algorithm gives a lower fitness function value than the PSO algorithm. While PSO provides a more fluctuating fitness value than DE. Both of these methods succeeded in finding the optimal gain parameter. Overall, the ability and effectiveness of the DE algorithm to achieve the best PID gain configuration in maintaining the electrode position is better than the PSO algorithm [34].

PID Controller using DE Method is implemented to determine the rotational speed of a DC motor with a high stable output. The simulation results display the rise time, settling time, and overshoot values of the DC motor and can be concluded that the PID tuning using the DE method produces an optimal and effective system [35].

5. CONCLUSION

1. Even though conventional PID tuning have been widely used in industrial fields because of their simple principles, easy implementation and wide applications, most of the system exhibits a large percent of overshoot.
2. Computational and Intelligent PID tuning such as Genetic Algorithm, Particle Swarm Optimization and Differential Evolution which are based on Artificial Intelligence-based computational methods, can produce optimal tuning of PID parameters, therefore producing optimal system performance.
3. These computational and intelligent PID tuning methods exhibits less percent of overshoot, faster rise and settling time and less errors than conventional methods.
4. For real-valued function optimization, due to the encoding of floating-point parameters into bit strings, Genetic Algorithms are less suitable but on the other hand, evolution strategies of Differential Evolution are well suited for real-parameter optimization.

REFERENCES

[1] Kiam Heong Ang, Gregory Chong, and Yun Li , "PID Control System Analysis, Design, and Technology", *IEEE Transactions On Control Systems Technology*, Vol. 13, No. 4, July 2005, pp. 559-576.

[2] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, Vol. 64, 1942, pp. 759–768.

[3] Katsuhiko Ogata, "Modern Control Engineering", *Prentice Hall*, 5th Edition, New Jersey 2010.

[4] Takuya Kinoshita, Toru Yamamoto, "Design of a Smart Adaptive Control System", *9th International Symposium on Advanced Control of Chemical Processes, IFAC Papers Online*, Vol. 48, No. 8, 2015, pp. 1309-1314.

[5] Karl Johan Astrom, Tore Hagglund, "PID Controllers: Theory, Design and Tuning", *Instrument Society of America (ISA)*, 2nd Edition, 1995.

[6] Kachitvichyanukul, "Comparison of Three Evolutionary Algorithms: GA, PSO and DE, *Industrial Engineering & Management System,* Vol. 11, No. 3, September 2012, pp. 215–223.

[7] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", MIT press, 1992.

[8] E. Purwanto, E. Wahjono, I. Ferdiansyah, D. S. Yanaratri, L. P. S. Raharja, R. P. Eviningsih, and G. Basuki, "Implementation of genetic algorithm for induction motor speed control based on vector control method", *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE Xplore, 2020, pp. 244–247.

[9] E. Flores-Moran, W. Yanez-Pazmino, and J. Barzola-Monteses, "Genetic Algorithm and fuzzy self-tuning pid for dc motor position controllers," *19th International Carpathian Control Conference (ICCC)*, IEEE, 2018, pp. 162–168.

[10] D. K. Meena and S. Chahar, "Speed control of dc servo motor using genetic algorithm", *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, IEEE, 2017, pp. 1–7.

[11] D. A. R. Wati and R. Hidayat, "Genetic algorithm-based pid parameters optimization for air heater temperature control," *International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, IEEE Xplore, 2013, pp. 30–34.

[12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", *Proceedings of the 1995 Sixth International Symposium on Micro Machine and Human Science*, IEEE Xplore, 2002, pp. 39–43.

[13] M. E. F. Moran and N. A. P. Viera, "Comparative study for dc motor position controllers", *IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, IEEE, 2017, pp. 1–6.

[14] James Kennedy and Russell Eberhart, "Particle swarm optimization," *Proceedings of International Conference on Neural Networks (ICNN) 1995*, Australia, Vol. 4, IEEE Xplore, 2002, pp.1942-1948.

[15] M. M. Reza Alavi Milani, T. Cavdar, and V. F. Aghjehkand, "Particle swarm optimization - based determination of ziegler-nichols parameters for pid controller of brushless dc motors," *2012 International Symposium on Innovations in Intelligent Systems and Applications*, IEEE Xplore, 2012, pp. 1–5.

[16] Kenneth V. Price, Rainer M. Storn, Jouni A. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", *Springer-Verlag Berlin Heidelberg*, Germany, 2005.

[17] Ruijun Dong, "Differential evolution versus particle swarm optimization for PID controller design", *Proceedings Fifth International Conference on Natural Computation, ICNC'09*, IEEE Xplore, 2009, pp. 236–240.

[18] Lyuyang Tong, Minggang Dong, ChaoJing, "An improved multi-population ensemble differential evolution", *Neurocomputing*, 2018, Elsevier, pp. 130–147.

[19] A.W. Mohamed, A.K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization", *International Journal of Machine Learning and Cybernetics*, Springer, 2019, pp. 253–277.

[20] Tanabe, R.; Fukunaga, A.S. "Improving the search performance of SHADE using linear population size reduction", *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, July 2014; pp. 1658–1665.

[21] Vladimir Stanovov, Shakhnaz Akhmedova , Eugene Semenkin "Selective Pressure Strategy in differential evolution: Exploitation improvement in solving global optimization problems", *Swarm and Evolutionary Computation 2019*, Vol. 50, Nov. 2019, pp. 1-41.

[22] Janez Brest, Mirjam Sepesy Maučec, Borko Bošković, "Single Objective Real-Parameter Optimization: Algorithm jSO", *2017 IEEE Congress on Evolutionary Computation (CEC),* IEEE Xplore, Spain, 2017, pp. 1311-1318.

[23] Wu Deng, Junjie Xu, Yingjie Song, Huimin Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem", *Applied Soft Computing Journal*, Vol. 100, 2020, Elsevier, pp. 1-16 .

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[24] Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, " Empirical investigations into the exponential crossover of differential evolutions", *Swarm Evolutionary Computation*, Vol. 9, 2013, Elsevier, pp. 27-36.

[25] Xin Qiu, Kay Chen Tan, Jian-Xin Xu, "Multiple Exponential Recombination for Differential Evolution", *IEEE Transactions on Cybernetics*, Vol. 47, No. 4, April 2017, pp. 995–1006.

[26] Shu-Mei Guo, Chin-Chang Yang, "Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator", *IEEE Transactions on Evolutionary Computing,* Vol. 19, 2015, pp. 31–49.

[27] R. Storn, "On the usage of differential evolution for function optimization", *Proceedings of the North American Fuzzy Information Processing*, Berkeley, CA, USA, 19–22 June 1996, pp. 519–523.

[28] A. Jayachitra and R. Vinodha, "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor", *Hindawi Publishing Corporation Advances in Artificial Intelligence*, 2014, pp. 1-8.

[29] Neenu Thomas, P. Poongodi, "Position Control of DC Motor Using Genetic Algorithm Based PID Controller," *Proceedings of the World Congress on Engineering*, London, U.K., Vol. 2 , July 2009, pp. 1618-1622.

[30] S. M. G. Kumar, J. Deepak and R. K. Anoop, "PSO based Tuning of a PID Controller for a High Performance Drilling Machine", *International Journal of Computer Applications*, Vol. 1, No. 19, 2010, pp. 12-18.

[31] M.H.T. Omar, W.M. Ali, M.Z. Mostafa, "Auto Tuning of PID Controller Using Swarm Intelligence", *International Review of Automatic Control*, Vol. 4, No. 3, Praise Worthy Prize S.r.l., 2011, pp. 319-327.

[32] Faouzi Aboura, "Tuning PID Controller Using Hybrid Genetic Algorithm Particle Swarm Optimization Method for AVR System*", 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, Istambul Turkey, IEEE Xplore, 2020, pp. 570 – 574.

[33] Eduardo Flores-Moran, Wendy Yanez-Pazmino, Luis Espın-Pazmino, Ivette Carrera-Manosalvas, and Julio Barzola-Monteses, "Particle Swarm Optimization and Genetic Algorithm PID for DC motor position controllers", *IEEE ANDESCON*, Oct 2020, IEEE Xplore, 6 pages.

[34] Trias Andromeda, Azli Yahya, Syahrullail Samion, Ameruddin Baharom and Nor Liyana Hashim , "Differential Evolution for Optimization of PID Gain in Electrical Discharge Machining Control System", *ICETI 2012, Transactions of the Canadian Society for Mechanical Engineering*, Vol. 37, No. 3, 2013, pp. 293-301.

[35] Siti Fatimah Anggraini, Alfian Ma'arif , Riky Dwi Puriyanto, "PID Controller on DC Motor and Tuning Using the Differential Evolution Method", *TELKA*, Vol.6, No.2, November 2020, pp. 147-159.