Journal of Robotics and Control (JRC)
Volume 5, Issue 2, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i2.20850

Review of Intelligent and Nature-Inspired Algorithms-Based Methods for Tuning PID Controllers in Industrial Applications

Ramakant S Patil [1*], Sharad P. Jadhav [2*], Machhindranath D. Patil [3]
[1,2] Department of Instrumentation Engineering, Ramrao Adik Institute of Technology, D Y Patil Deemed to be University, Nerul, Navi Mumbai, India,
[3] Department of Instrumentation Engineering, V. E. S. Institute of Technology, Chembur, Mumbai, India
Email: [1] ramakant.patil@rait.ac.in, [2] sharad.jadhav@rait.ac.in, [3] mdpatil1610@gmail.com
*Corresponding Author

***Abstract*—PID controllers can regulate and stabilize processes in response to changes and disturbances. This paper provides a comprehensive review of PID controller tuning methods for industrial applications, emphasizing intelligent and nature-inspired algorithms. Techniques such as Fuzzy Logic (FL), Artificial Neural Networks (ANN), and Adaptive Neuro Fuzzy Inference System (ANFIS) are explored. Additionally, nature-inspired algorithms, including evolutionary algorithms like Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO), Simulated Annealing (SA), Artificial Bee Colony (ABC), Firefly Algorithm (FA), Cuckoo Search (CS), Harmony Search (HS), and Grey Wolf Optimization (GWO), are examined. While conventional PID tuning methods are valuable, the evolving landscape of control engineering has led to the exploration of intelligent and nature-inspired algorithms to further enhance PID controller performance in specific applications. The study conducts a thorough analysis of these tuning methods, evaluating their effectiveness in industrial applications through a comprehensive literature review. The primary aim is to offer empirical evidence on the efficacy of various algorithms in PID tuning. This work presents a comparative analysis of algorithmic performance and their real-world applications, contributing to a comprehensive understanding of the discussed tuning methods. Findings aim to uncover the strengths and weaknesses of diverse PID tuning methods in industrial contexts, guiding practitioners and researchers. This paper is a sincere effort to address the lack of specific quantitative comparisons in existing literature, bridging the gap in empirical evidence and serving as a valuable reference for optimizing intelligent and nature-inspired algorithms-based PID controllers in various industrial applications.**

Keywords*— PID controller; Intelligent and Nature-Inspired Algorithms; Fuzzy Logic; Artificial Neural Network; Adaptive Neuro-Fuzzy Inference System; Genetic Algorithm; Particle Swarm Optimization; Differential Evolution; Ant Colony Optimization; Simulated Annealing; Artificial Bee Colony; Firefly Algorithm; Cuckoo Search; Harmony Search; Grey Wolf Optimization.

I. Introduction

The primary objective of this paper is to provide a comprehensive and critical review of state-of-the-art intelligent and nature-inspired algorithms employed for tuning PID controllers in industrial applications. Through an in-depth exploration of various algorithms, including intelligent FLC, ANN, ANFIS, and nature-inspired approaches GA, PSO, DE, ACO, SA, ABC, FA, CS, HS, GWO, etc. This review aims at effectiveness, comparative analysis, trends in the application, challenges and limitations, and future directions of these algorithms. The chemical, petrochemical, oil refineries, distilleries, and manufacturing industries need automation and process control for stable operation and production under safe conditions. In the field of control engineering, Proportional-Integral-Derivative (PID) control is an essential method. It is crucial to control systems and maintain the desired outcomes. PID controller has three basic control actions proportional, integral, and derivative hence the name is PID. These are also used in composite forms like PI, PD, and PID for process control. It has an easy-to-implement structure that makes it suitable for robust control applications [10].

A. Background and importance of PID controllers in control systems

In 1940, PID became the standard controller in industrial process control systems [10]. In process control systems nearly, 95 percent of the control loops use PI controllers. There are two forms of PID parallel and series. In mechatronics, PID is commonly used to regulate a wide range of systems, including motors, robots, drones, and more. It can improve the reliability, accuracy, and functionality of the control system by adjusting the output following the system error. The proportional, integral, and derivative gains all have optimal values that must be determined. Equation (1) represents the PID algorithm and Fig. 1 shows a block diagram of the PID controller. In a resistance network system, the output is directly proportional to the error input signal is a proportional action in that keeping the Kp constant coefficient. In a low-pass filter system output is directly proportional to the integral of the error input signal is an integral action in which Ki is an integral constant. In a high pass filter system output is directly proportional to the derivative of the error input signal is a derivative action in which Kd is the derivative constant. Combining all three actions obtained the PID controller [10].

$$u(t) = Kp.e(t) + Ki \int_0^t e(t)dt + Kd\frac{de(t)}{dt} \tag{1}$$

Where Kp is the proportional gain, Ki = (Kp/Ti) is the integral

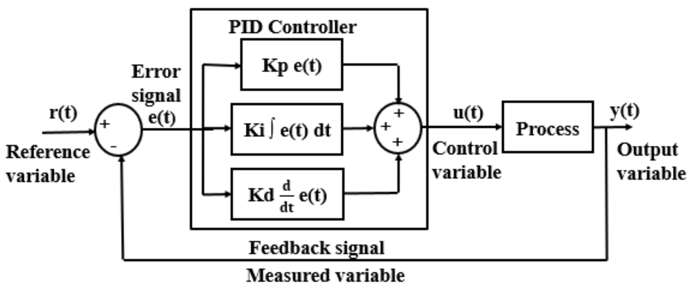

Fig. 1. PID controller.

gain and Kd = (Kp. Td) is the derivative gain. These parameters are to be tuned. The time response parameters of any system such as rise time (Tr), settling time (Ts), peak time (Tp), peak overshoot (Mp), and steady-state error (ess) are affected by system dynamics. PID must be correctly tuned to determine the control system's good response. The Kp is used to minimize 'Tr', the Ki is used to eliminate 'ess', and the Kd is used to reduce the 'Mp' and 'Ts' of the control system. Equation (2) shows the transfer function of the PID controller which is derived from equation (1) simply by taking the Laplace transform of it. The transfer function model is used to define the PID controller in MATLAB.

$$C(s) = Kp + \frac{Ki}{s} + Kd.s \tag{2}$$

The controller whether performing correctly or not makes use of certain performance criteria (performance indexes) such as Integral Square Error (ISE), Integral Absolute Error (IAE), Integral Time Square Error (ITSE), Integral Time Absolute Error (ITAE), etc [11]. These are used for minimizing an integral error criterion in the control system. Adjust the PID parameters to minimize the selected performance index. For ISE and IAE, tuning primarily involves adjusting the integral gain (Ki), and for ITSE and ITAE, tuning may involve adjusting both proportional (Kp) and integral (Ki) gains to balance overall error reduction and faster error reduction over time [10].

$$ISE = \int_0^\infty e^2(t)dt \tag{3}$$

$$IAE = \int_0^\infty |e(t)|dt \tag{4}$$

$$ITSE = \int_0^\infty te^2(t)dt \tag{5}$$

$$ITAE = \int_0^\infty t|e(t)|dt \tag{6}$$

B. Classification of PID controller tuning methods

Ideally, it is expected to obtain fast responses and good stability of any control system. But practically these two things are not possible to achieve simultaneously. Because for achieving a faster response of the system there is obtained worse stability and for achieving better stability obtained slower response of the system. So, any control system can maintain acceptable stability and medium fastness of response. Therefore, accurately tuning PID is an essential task. The process of finding the controller parameters that result in the desired output is known as tuning. The classification of various PID controller tuning algorithms or techniques is shown in Fig. 2. These methods are broadly divided into conventional, intelligent, and nature-inspired [11],[13],[29]. The advantages of conventional methods are that it is simple, systematic, and widely applicable. However, drawbacks are that they may result in aggressive or oscillatory responses, they may not account for the system's non-linearities or uncertainties, and they may require testing the system near instability. Intelligent and nature-inspired PID tuning methods offer improved adaptability, efficiency, and robustness in handling complex and dynamic systems. The capabilities of intelligent and nature-inspired algorithms make them particularly well-suited for applications where traditional methods may fall short. Some algorithms and PID controller tuning methods will be discussed in this review.

The contributions of this paper are as follows:

- The various types of techniques to tune the PID controller, which are broadly classified as conventional, intelligent, and nature-inspired techniques are mentioned in this paper.
- For selected conventional methods challenges are summarized and the importance of intelligent and nature-inspired algorithms is described for PID tuning.
- Special intelligent and nature-inspired algorithms to tune the PID controller with specific systems have been included in the literature survey.
- A comparative analysis of the selected algorithms was carried out across many criteria, including accuracy, stability, efficiency, and parameter sensitivity. The analysis aimed to identify the primary features of each algorithm.
- Highlighted the difficulties with the algorithms and the potential of the entire topic of nature-inspired algorithms, which can serve as a platform for future research.

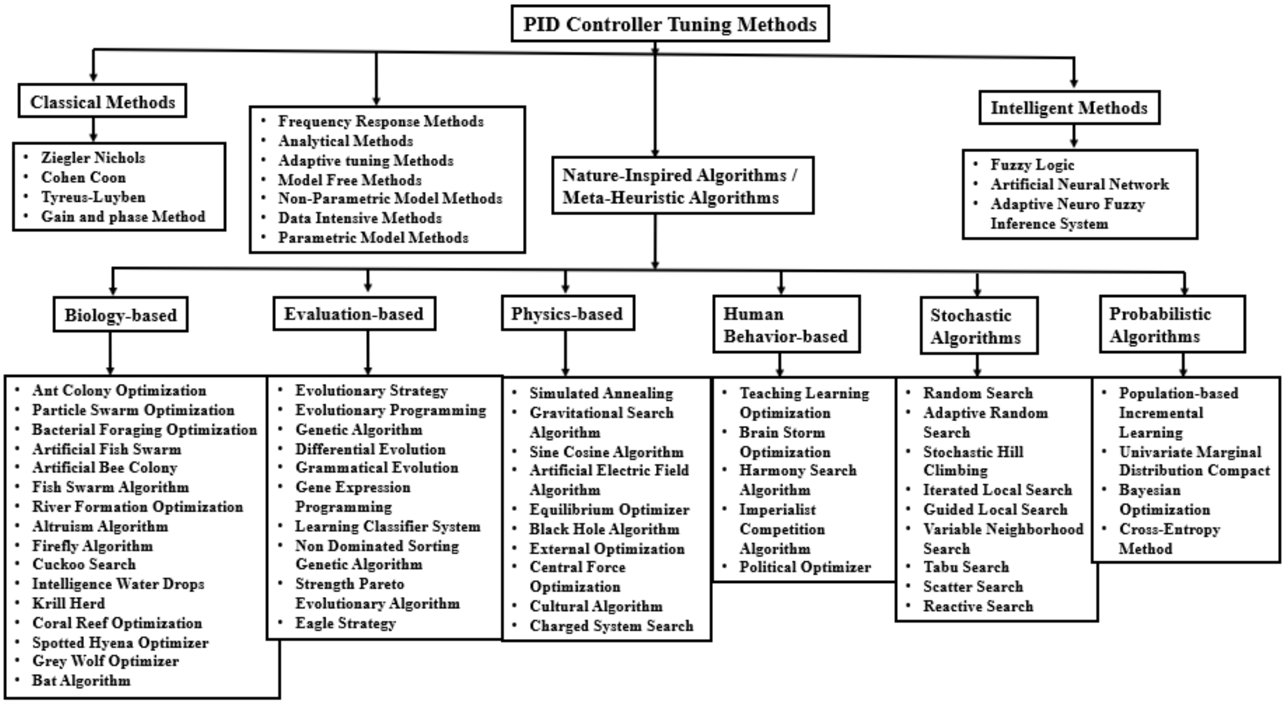

Fig. 2. Classification of PID tuning methods.

Structure of this Paper:
In this paper, after the brief introduction; Section II covers some conventional tuning methods of PID with limitations and advanced control schemes with examples; Section III discusses intelligent techniques for PID tuning considering examples; Section IV provides details on the principles of some PID controller tuning methods using nature-inspired algorithms; Section V focuses on the comparison in terms of complexity, convergence speed, robustness, adaptability, and performance improvement of these algorithms; The few real-world applications of intelligent and nature-inspired algorithms based PID are mentioned in Section VI from various literature survey; Discuss challenges and limitations of algorithms in Section VII; In Section VIII discuss future directions and research opportunities on nature-inspired algorithms and finally conclude this paper in Section IX.

II. CONVENTIONAL PID TUNING METHODS

This section discusses some conventional methods and their limitations. The methods employed for Ziegler-Nichols and Cohen-Coon controller tuning are the most often used conventional methods. When there is not a mathematical model of the system available, such methods can be useful.

A. Time and Frequency Domain Approach

The design of the PID controller is possible in the time domain [14]. In this method consider a closed-loop system (G(sd)) at s = sd (dominant pole) with controller (Gc(sd) = Kp + Ki/sd + sd.Kd) and unity feedback. Tr, Ts, Tp, Mp, and ess time domain specifications consider for PID design. From these specifications find ζ (damping ratio) and ω_n, and dominant pole pairs $sd = -\zeta\omega_n \pm j\omega\sqrt{1-\zeta^2}$

$$Kp = \frac{-sin(\beta+\phi)}{|G(sd))|\, sin\beta} - \frac{2Ki.cos\beta}{|sd|} \tag{7}$$

$$Kd = \frac{sin(180^0-\phi)}{|G(sd)|\,|sd|\, sin\phi} + \frac{ki}{|sd|^2} \tag{8}$$

where $\beta = \angle sd$ and $\phi = \angle G(sd)$. For the PI controller, substitute Kd=0 in equation (8), then find Ki, Kp. For the PD controller, substitute Ki=0 in equation (8), then find Kd, Kp.

The design of the PID controller is possible in the frequency domain [14]. In this method consider a closed-loop system $(G(j\omega gc))$ at $\omega = \omega gc$ (gain crossover frequency) with controller $(Gc(j\omega gc) = Kp + Ki/\omega gc + \omega gc.Kd)$ and unity feedback. Phase Margin (PM), Gain crossover frequency (ωgc), Gain Margin (GM), and Phase crossover frequency (ωpc) specifications are used for PID design. These are found by using methods such as Bode, Polar, Nyquist, etc. For controller design, any two specifications are sufficient such as GM and PM, ωgc and PM, or ωpc and GM. In literature several approaches are given, here use ωgc and PM along with steady-state error specification for PID. PM is determined from the open loop transfer function of the system. PM =

$180^0 + \angle G_{OL}(j\omega gc)$, PM = $180^0 + \angle G(j\omega gc) + \angle Gc(j\omega gc)$, PM = $180^0 + \phi + \theta$.

$$Kp = \frac{cos\theta}{|G(j\omega gc)|} \tag{9}$$

$$Kd = \frac{sin\theta}{|G(j\omega gc)|\,\omega gc} + \frac{Ki}{\omega gc^2} \tag{10}$$

Determine Ki from steady-state error and then find Kp, Kd using equations (9) and (10). For the PI controller, substitute Kd=0 and then find Kp and Ki. For the PD controller, substitute Ki=0 and then find Kp and Kd. The values of gain and phase margins decided the stability of the system. If both GM and PM are positive ($\omega gc < \omega pc$), then the system is stable. If both GM and PM are negative ($\omega gc > \omega pc$), then the system is unstable. If both GM and PM are zero ($\omega gc = \omega pc$), then the system is just stable. However, these methods are time-consuming and tedious.

B. Trial and Error Method

In the trial and error tuning method, first, keep integral and derivative values at a minimum, and the proportional gain is adjusted until a desired output is obtained. For example, in a temperature control system, adjust approximately Kp = 2-10, Ti = 2-10 min, and Td = 0-5 min due to the slow response of temperature sensors to temperature changes.

C. Ziegler-Nichols (ZN)

ZN is the most popular and commonly used tuning technique. It was designed by Ziegler-Nichols in 1942 [10]. This method involves determining the proportional gain at which the output of the system becomes oscillatory to a step input to determine the parameters of the PID controller. The oscillation frequency is known as the ultimate frequency, and this gain is known as the ultimate gain. The ZN method provides two sets of formulas, one for the closed loop response and another for the open loop response, to calculate the PID gains [15],[21]. In the first ZN method, Fig. 3, the system time constant 'τ' and delay 'L' are measured from the S-shaped process reaction curve of the system which is obtained from step response. By using these parameters, find the Kp, Ti, and Td from Table I. In the second ZN method calculate ultimate gain (Ku) and period (Pu) for finding the new Kp, Ti, and Td. In this method maintain the Ti at ∞ and the Td at 0 and adjust the value of Kp till the consistent oscillations are obtained. Then measure Kp=Ku and the distance between two successive oscillations (Pu). Finally, calculate Kp, Ti, and Td from Table II. It is a simple, scientific approach, and a widely use technique. The ZN method gives better results than the Cohen-Coon method.

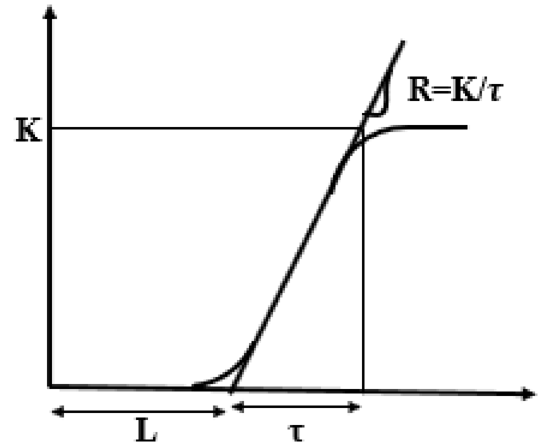

Fig. 3. S-shaped step input response curve.

TABLE I. ZIEGLER-NICHOLS OPEN-LOOP TUNING PARAMETER

Controller Type	Kp	Ti = Kp / Ki	Td = Kd / Kp
P	τ / L	∞	0
PI	0.9 (τ / L)	L / 0.3	0
PID	1.2 (τ / L)	2 L	0.5 L

D. Cohen-Coon (CC)

This method is typically used for open-loop systems [15]. When the processing delay is significant about the open loop time constant, the controller tuning adjusts for the sluggish, steady-state response provided by the ZN technique. The controller does not instantly react to the disturbance, hence the CC approach is only applicable to systems with first-order lag and time delay. The CC technique requires more information from the steady-state response than the ZN method does. This includes the moment at which the input step begins, the response reaches half of its maximum value, the response

TABLE II. ZIEGLER-NICHOLS CLOSED-LOOP TUNING PARAMETER

Controller Type	Kp	Ti = Kp / Ki	Td = Kd / Kp
P	0.5 Ku	-	0
PI	0.45 Ku	Pu / 1.2	0
PID	0.6 Ku	Pu / 2	Pu / 8

reaches 63.2 percent, the time constant, the dead time constant, and the delay measured.

E. Tyreus-Luyben (TL)

The same procedure applies for calculating the critical gain and period as per the ZN method in the TL tuning method [21]. In comparison to the ZN method, the TL method produces greater control loop stability through modifications to the controller parameter formulas. The formulas suggested for the PID controller Kp=0.45Ku, Ti=2.2Ku, and Td=Pu/6.3. The TL method performs better with low values for process dead time because it is more careful than the ZN method. However, if there is a significant amount of dead time, the performance decreases. When tuning the controller, it takes into account the maximum gain Ku and the peak frequency Pu.

F. Internal Model Control (IMC)

Robustness was a consideration in the development of this technique [15]. chemical engineering applications are not suitable for the significant controller gain and short integral time provided by the ZN open loop and CC approaches. The IMC approach is associated with closed-loop control and is devoid of oscillatory or overshooting behavior.

G. Challenges in Conventional PID Tuning and the Need for Automated Methods

Tuning the parameters of PID controllers to get the correct response is one of the most difficult tasks. The system responds properly once the operator adjusts the parameters.

- The frequency response tuning method may be more difficult, time-consuming, and mathematically demanding compared to other methods, and it may not be feasible for all systems that are challenging to excite or measure at various frequencies [14].
- By trial and error tuning method, the PID parameters are adjusted manually. This process can take a long time and requires an operator knowledgeable about system dynamics. The manual tuning method is suitable for systems that are easy to manipulate and that do not have complex dynamics. These methods never give optimal PID tuning parameter values when set point changes due to process disturbances.
- The ZN tuning requires primary knowledge of the process plant model and gives a very poor response if the process is dead-time dominant. Also, it gives a very low robustness, which can lead to loop instability because it tunes the loop for a quarter-amplitude-damping response, which oscillates and overshoots a lot. A tuning rule for quarter-amplitude damping recommends a controller gain of 0.9. The overshoot is challenging in processes like temperature control in the manufacture of plastic gloves. Due to overshoot of temperature, a lot of material will get wastage which cannot be recovered. So, in such applications avoid to use of the ZN method [15].
- The main design requirement for the CC tuning approach is disturbance rejection. However, it is limited to first-order models that involve significant process delays. It is an offline method and approximation values for controller gain, integral, and derivative time may be not accurate for different systems [15].
- The IMC approaches for first-order dead time systems are highly complex.

Table III (referring to several references) summarizes the comparison between conventional PID tuning methods in terms of strengths and weaknesses with their applicability, accuracy, and robustness.

H. Need of Intelligent and Nature-Inspired PID Tuning Methods

The different literature review has shown that the performance of systems was improved by using intelligent and nature-inspired algorithms. According to the various literature surveys, conventional methods are not suitable for tuning the PID controllers for higher order processes for achieving better response of the system. Thus, intelligent techniques along with nature-inspired algorithms are used for tuning a PID controller. Many researchers have proved that intelligent and nature-inspired methods provide better performance of systems as compared with conventional methods. Thus, to continue forward with more in-depth research work, a literature review in this field is required. As per some literature reviews, a few of them are also called intelligent algorithms. In this review, intelligent and nature-inspired algorithms are differentiated as per their structure and working.

For example, Hendril Satrian Purnama et.al [3] proved that the ZN-PID controller is not efficient in the speed control of DC motors. By using the ZN method, the result obtained like the transient characteristics such as rise time and settling time is slow, steady state and maximum overshoot is high affecting the speed control of the motor. On the other hand, by using GA, PSO, and fuzzy logic transient characteristics are improved and the speed control of the motor is also improved. It shows intelligent and nature-inspired techniques were excellent and robust techniques to solve the problem and limitations in the conventional PID controller.

Meenakshi Sharma et.al [8] developed a controller for a process like flow, temperature, pressure, and level utilizing GA, ANN, PSO, and fuzzy logic and found that traditional PID controller tuning is a tedious procedure. Because it is manual, it takes a very long time. Therefore, intelligent and nature-inspired algorithms are applied to increase the PID controller speed.

The control scheme uses intelligent and nature-inspired algorithms for finding the optimized setting of PID shown in Fig.

TABLE III. STRENGTHS AND WEAKNESSES OF CONVENTIONAL PID TUNING METHODS

Criterion	**Frequency Domain**	**Ziegler-Nichols**	**Trial and Error**	**Tyreus-Luyben**	**Internal Model Control**
Strengths	Systematic approach	Quick and straight-forward method	Flexibility for tuning various systems	Suitable for well-understood dynamics processes	Incorporates model-based approach for improved tuning
	Provides insights into stability and performance margins	Suitable for identifying initial tuning parameters	Practical for manual tuning in real-world applications	Balances robustness and performance	Enables better handling of process dynamics
Weaknesses	Requires detailed knowledge of system dynamics	Tends to be conservative, which may result in over-tuning	Time-consuming and may not converge to optimal values	Limited applicability to certain system types	Limited applicability to highly nonlinear systems
	May not be suitable for systems with rapidly changing dynamics	Limited robustness for systems with uncertainties	Highly dependent on operator expertise and intuition	Success depends on the accuracy of the selected model	May require more effort for complex process models
Applicability	Well-suited for systems with known and stable dynamics	Basic and suitable for simple systems	Versatile for various systems and applications	Best suited for processes with relatively well-understood dynamics	Suitable for a wide range of processes with known models
Ease of Implementation	Requires access to the system's transfer function	Simple and quick to implement	Straightforward but time-consuming	Moderate complexity due to model identification	Moderate complexity but may require a good process model
Accuracy in Tuning	Offers accurate tuning when the system model is accurate	Provides a starting point, but fine-tuning may be necessary	May result in suboptimal tuning without iteration	Accurate if the process dynamics match the assumed model	Accurate if the model accurately reflects the process
Suitability for Different Systems	Suitable for systems with well-defined and stable dynamics	Limited to stable and linear systems	Versatile, applicable to various system types	More suitable for processes with known dynamics	Applicable to various systems with known models
Robustness	Robust when the system model accurately represents the dynamics	Limited robustness, may not generalize well	Robust if the tuning process converges to optimal values	Robust to uncertainties if the model accurately reflects the process	Robust to model uncertainties and variations

4 [25]. The first step in this approach is to use performance indices like ISE, IAE, ITSE, and ITAE to minimize the system error signal. Then this minimum error value is considered as the objective function in optimization algorithms. These algorithms find optimized values of the PID controller. Finally, run the system again with optimized values on MATLAB software observe the process or system response, and compare the results obtained from various algorithms.

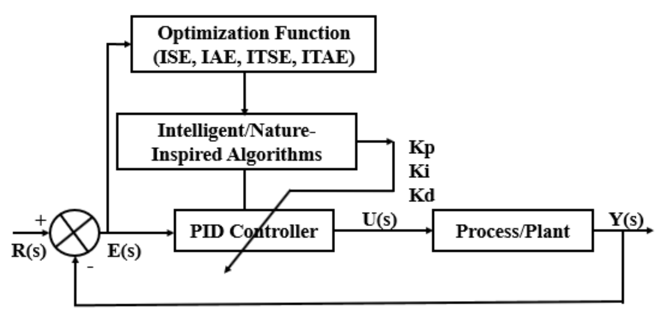

Fig. 4. Intelligent or Nature Inspired PID controller.

III. INTELLIGENT ALGORITHMS FOR PID TUNING

The intelligent technique without modeling the plant takes into account unknown elements, which may be nonlinear and/or time-varying. By using these techniques, the controller design is called an intelligent PID controller. The intelligent techniques are FLC, ANN, ANFIS, etc. The basic operator of intelligent control is a multiscale structure, which is achieved by attentional concentration, generalization, and systematic investigation of states of a problem domain. Some basic aspects of the selected algorithm were provided in this review. The details of the implementation or development of selected algorithms for PID design for a particular system were not presented. An explanation of intelligent techniques considering the example of a continuously stirred tank reactor (CSTR) [16] is given.

A. Fuzzy Logic Control (FLC)

The system performs better with the fuzzy PID controller because it automatically modifies Kp, Ki, and Kd. It gives good performance in set point and load disturbance change, which is not achievable with conventional PID controllers. The fuzzy logic architecture or process is shown in Fig. 5. The rule base is a component used for storing the set of rules given by the experts. The fuzzification module converts the crisp value which is measured by sensors to fuzzy values [31]. All information is processed to the inference engine module. It finds the matching degree between the current fuzzy input and the rules. After that system determines which rule is to be added according to the given input field. When all rules are applied, then they are combined to develop the control actions. The defuzzification module converts the fuzzy value into a crisp value. Various techniques are used to do this, but select the best one for reducing the errors. A fuzzy logic system utilizes expert knowledge to convert a verbal control strategy into an automatic control strategy.

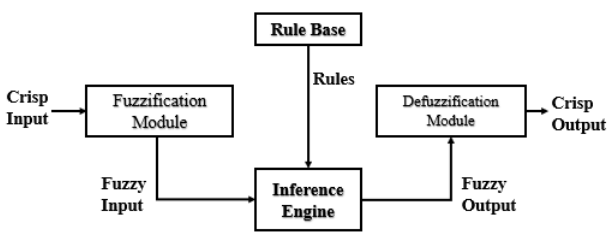

Fig. 5. Process of a Fuzzy Logic.

For example, consider the second-order system of concentration control of CSTR [16]. It is non-minimal (inverse response) with the right half of plane zero shown in equation (11). Fig. 6. shows the block diagram of a fuzzy PID. The structure of a fuzzy-PID controller is two inputs ('e' and 'Δe') and three outputs (Kp, Ki, Kd). The ZN approach was used to calibrate the PID controller to determine the input and output membership functions range. For both inputs, some verbal variable levels are assigned as zero (Z), negative big (NB), negative small (NM), and positive big (PB), positive small (PS).

$$G(s) = \frac{-1.1170s + 3.1472}{s^2 + 4.6429s + 5.3821} \tag{11}$$

The rule base for the fuzzy controller of the CSTR model is shown in Table IV [16]. The output of the fuzzy set from the 'e' and 'Δe' is used to tune the PID parameters with

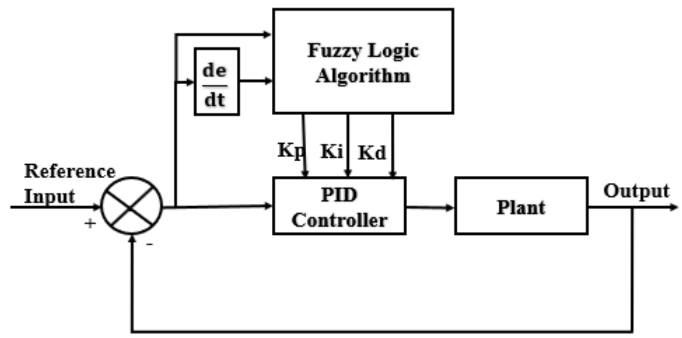

Fig. 6. Block diagram of Fuzzy PID controller.

appropriate values. The output fuzzy sets of this process are obtained according to equations (12),(13),(14). Finally, implement Simulink block set for fuzzy PID controller for CSTR process and obtained step response.

$$Kp' = \frac{Kp - Kpmin}{Kpmax - Kpmin} \tag{12}$$

$$Ki' = \frac{Ki - Kimin}{Kimax - Kimin} \tag{13}$$

$$Kd' = \frac{Kd - Kdmin}{Kdmax - Kdmin} \tag{14}$$

TABLE IV. RULE BASE FOR FUZZY CONTROLLER

Δe \ e	NB	NM	Z	PM	PB
NB	NB	NB	NM	NM	Z
NM	NB	NM	NM	Z	PM
Z	NM	NM	Z	PM	PM
PM	NM	Z	PM	PM	PB
PB	Z	PM	PM	PB	PB

B. Artificial Neural Network (ANN)

ANN model consists of artificial neurons that process data from input to output. Training and testing data through the Input layer, Hidden layer, and Output layer made by various nodes. The Input layer initializes input variables with weight variables. The function of the Hidden layer is a linear transformation. The function of the Output layer is prediction compared with the actual result and the difference is called loss. It minimizes using the back-propagation or supervised delta learning method by using weight manipulation. The PID controller's gain is determined by the neuron weights. The neural network's weights have been designed to achieve the system's desired outcome.

Fig. 7 shows the block diagram of the ANN PID controller. It is a single-neuron structure. x_1, x_2, x_3 nothing but error multiply with w_1, w_2, w_3 and these weights will act as PID gains. The proportional error x_1, the integral error x_2, and the derivative error x_3 are obtained according to equations (15),(16),(17). By using equation (18) neuron output is obtained.

$$x_1 = e(k) - e(k-1) \tag{15}$$

$$x_2 = e(k) \tag{16}$$

$$x_3 = e(k) - 2e(k-1) - e(k-2) \tag{17}$$

$$u(k) = u(k-1) + k\sum_{i=1}^{3} w_i(k)x_i(k) \tag{18}$$

The following are the steps for using ANN to tune the PID controller:
Step 1: To select the weights' random values.
Step 2: To determine the error difference between the output and reference input.
Step 3: Using the error signal, the supervised delta learning technique determines the PID controller's gains.
Step 4: The output of the single neuron is multiplied by gain 'k' to achieve the optimal closed-loop response.
Step 5: The revised weights will act as the Kp, Ki, and Kd.

The weights are updated by using equations (19),(20),(21) as per the supervised delta learning algorithm. Where, η_p, η_I and η_D are the proportional, integral, and derivative learning speeds.

$$w_1(k) = w_1(k-1) + \eta_P(k-1)u(k-1) \tag{19}$$

$$w_2(k) = w_2(k-1) + \eta_I(k-1)u(k-1) \tag{20}$$

$$w_3(k) = w_3(k-1) + \eta_D(k-1)u(k-1) \tag{21}$$

C. Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a new form of neural network that merges fuzzy logic (FL) and neural network (NN) design [16],[30]. In fuzzy logic manually build the rule base but in this method rule base will be modified as per the system using ANN. Using an input-output collection method, it builds a fuzzy inference system (FIS). The input and output data set was obtained via a PID controller that was adjusted through conventional methods. The ANFIS model structure consists of three hidden layers and a feed-forward, two input, and one output structure. The error back-propagation algorithm is used to construct the ANFIS membership functions. The 'e' and 'Δe' are the inputs while the Kp, Ki, and Kd are the outputs of the ANFIS controller. Then the system (CSTR) is implemented by using the ANFIS editor in MATLAB [16].

IV. Nature-Inspired Algorithms for PID Tuning

"Nature-inspired algorithms" refers to a type of algorithms that are inspired by physical and chemical systems, biological systems, and swarm intelligence, among other natural phenomena. GA, PSO, DE, ACO, SA, ABC, FA, CS, HS, GWO, etc. are the nature-inspired algorithms used for PID tuning. For example, K. Anbumani et.al designed GWO-PID and PSO-PID for a heat exchanger process. They compare the performance of GWO and PSO algorithms and results Tr, Ts, Mp, and Tp are compared and found that GWO-PID performs better than PSO-PID for heat exchangers [73].

A. Genetic Algorithms (GA)

It is a search method based on selection and natural genetics [17]. In 1975, John Holland developed it. It is based on natural selection, genetics, and mutation. A number is encoded by GA into a binary string known as chromosomes and encoding includes binary, natural number, real number, matrix, tree, and quantum. To carry out the crossover and mutation processes, the parents are determined from a collection of binary strings based on the value of the evaluation function, also known as the fitness function. It starts with a population of strings and generates a successive population of strings. Simple GA copy and swap strings. The population is in a set of strings. The new chromosomes mean members of the population created after each generation, thus this algorithm is called a genetic algorithm. The individual strings are copied as per the values their objective function is the function of reproduction step. Selection of the fittest candidates from the population is through the reproduction process. After that new chromosomes and genes are for the next generation by using a crossover process. The mutation operator in GA is used to obtain better results and string position. In chromosomes, some genes are altered by the mutation process.

The following steps are used for the development of a GA-PID for the CSTR process [16].
Step 1: Describe the plant model.
Step 2: Determine the objective function that needs to be minimized by initializing the PID parameters Kp, Ki, and Kd.
Step 3: Find the values for particle best (pbest) and global best (gbest).
Step 4: Use the mutation process to compute the new population.
Step 5: Get updated gbest1 and pbest1 values.
Step 6: Compare the values of pbest and pbest1.
Step 7: Compare the values of gbest and gbest1.
Step 8: Continue from Step 2 until the optimal value is reached.
Step 9: Find the step response for the closed-loop system and get the updated values for the PID parameters Kp, Ki, and Kd.

The advantages of the GA technique are efficient and effective, one population solution is used for obtaining a new

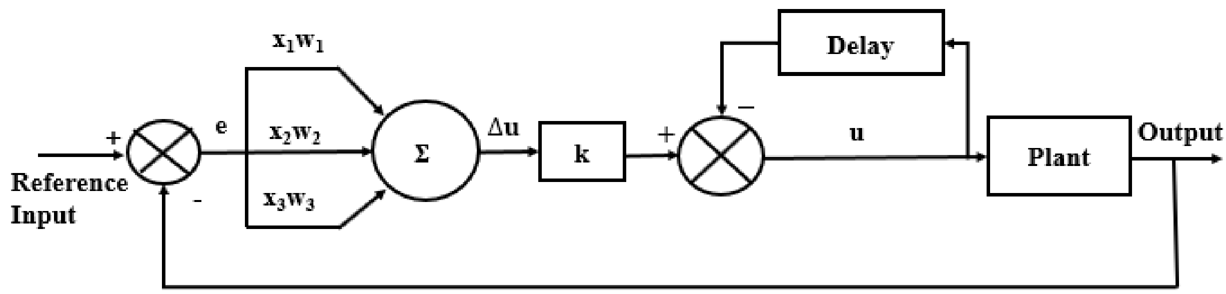

Fig. 7. ANN structure.

population, its concept is very simple to understand, it uses performance index data, not derivatives, it supports multi-objective optimization, and is stochastic, and it is a useful technique in a combined discrete and continuous system. The disadvantages of this technique are its implementation is difficult and is expensive and time-consuming technique. The applications of GA-based PID in industry such that speed control of DC motor [3], level control of tank [8][32], CSTR process control [16], servo processes control [23], Autonomous vehicle control [50] like many more.

B. Particle Swarm Optimization (PSO)

The first PSO was created in 1995 by Kennedy and Eberhart [26],[27]. PSO imitates the movement of particles or a flock of birds in a search space. This algorithm is used for solving problems of nonlinear systems and gives optimized solutions [28]. It searches for optimal values by updating generations. In this method, the swarm is nothing but a population and the particle is a member of the population [1]. It starts with a random initial population and moves randomly in selected directions. Each particle travels under the searching space and remembers the best previous positions of itself and its neighbors. Evaluate fitness values by fitness function and it is used in algorithms for optimization. Optimization means the optimum point where conditions are best and most favorable. It is used to find the best among different possible solutions.

PSO Initialization steps are as follows:
Step 1: A group of randomly distributed particles initializes the PSO algorithm (each particle represents a solution).
Step 2: Each particle updates generation (iteration) in search of the optimum value.
Step 3: Every particle is updated by in each iteration.

- First Best one is the best solution (Fitness)
- Second Best is tracked by particle swarm optimizer.
- After finding the two Best values, the particle updates its velocity and position.
- Position by:

$$\mathrm{x}_i^{t+1} = \mathrm{x}_i^t + \mathrm{v}_i^t \tag{22}$$

- Velocity by:

$$\mathrm{v}_i^{k+1} = w\mathrm{v}_i^k + c_1 r_1 (x\mathrm{Best}_i^t - \mathrm{x}_i^t) + c_2 r_2 (g\mathrm{Best}_i^t - \mathrm{x}_i^t) \tag{23}$$

Where, xBest= best particle position, w= inertia weight, c_1, c_2 = two positive constants=2, r_1, r_2= two random parameters within (0,1), gBest= best group position.

Objective functions are used to maximize or minimize values that are trying to optimize. Each particle has velocities. The main process for creating the PSO algorithm is to randomly initialize each particle's parameters, population, position, and velocity before calculating the fitness value. Set the new value as gBest if the fitness value is above the best fitness value, gBest. Determine the position and velocity of each particle, estimate fitness, and determine the most recent best value gBest. Update time t = t + 1; gBest and the fitness value are the outputs; continue until the condition is met [26],[27]. The best value from each trial is taken into consideration to stabilize the process, and the ideal tuning technique is performed ten times on its own. Therefore, a more optimal selection of PID settings reduces the performance index and improves system response.

The applications of the PSO-PID controller for CSTR temperature control [18]. This PSO has tuned the gains much more optimally, thus improving system performance. The rise time of 0.599 msec and the settling time of 1.12 msec were obtained with no overshoot for CSTR and as compared with GA significant improvement in performance, PSO-PID in seed control of BLDC motor [1], [3], [28], Coupled Tank System control [22], servo process control [23], etc. In their research, N. Divya and A. Nirmalkumar [65] reviewed a few soft computing techniques for fine-tuning PID controller parameters.

C. Differential Evolution (DE)

In 1995, Rainer Storn and Kenneth Price proposed the DE algorithm [76]. It is a stochastic optimization algorithm with a population base. It has three operations like GA such as mutation, crossover, and selection. In GA, two sub-individuals are produced by crossing two parent individuals; on the other hand in DE, new individuals are produced by varying the various vectors of multiple individuals. Additionally, in DE, the individual is updated only when the new individual surpasses the old one, whereas, in GA, the offspring individual replaces the parent individually with a given probability. This algorithm is iterative evolves and improves the population, converging to a set of optimal PID parameters. Through inter-group rivalry and cooperation, DE conducts an optimization search. A population theory-based global search approach is being used. Using real number coding, variation operation based on difference, and a one-to-one competitive survival strategy, reduces the complexity of the genetic operation. It also possesses a considerable degree of memory. It can adapt its search technique in real time to the search environment and exhibits excellent robustness and powerful global search capabilities. The differential evolution algorithm's key benefits include fewer unknown parameters, difficulty falling into local optimization, and quick convergence [103].

During mutation operation each individual x_i(t) can found as per equation (24),

$$v_{i,j}(t) = x_{r_1,j}(t) + F(x_{r_2,j}(t) - x_{r_3,j}(t)) \tag{24}$$

where $x_{r1}(t)$, $x_{r2}(t)$ and $x_{r3}(t)$ are three individuals selected randomly from the whole population, $r_1 \neq r_2 \neq r_3 \neq i$ and $F \in [0, 2]$ is the mutation factor. The crossover individual $u_i(t) = (u_{i,1}(t), u_{i,2}(t), ..., u_{i,D}(t))$ can be generated by the mutation individual $v_i(t)$ and its parent individual $x_i(t)$, as described in Equations (25), (26),

$$u_{i,j} = v_{i,j}(t), \text{if} rand \leq CR, \text{or} j = j_{rand} \tag{25}$$

$$u_{i,j} = x_{i,j}(t), \text{if} rand \geq CR, \text{and} j \neq j_{rand} \tag{26}$$

where rand is a random number in the range [0, 1], CR is the crossover factor and it is a constant in the range [0, 1]; j_rand is an integer selected randomly from the range [1, D]. During the selection procedure, the DE algorithm adopts the "greedy" strategy; the next-generation individual is selected between parent individual $x_i(t)$ and the crossover individual $u_i(t)$, which has the better fitness value, as described in equations (27), (28).

$$x_i = x_i(t), \text{if} f(x_i(t)) \text{is better than} f(u_i(t)) \tag{27}$$

$$x_i = u_i(t), \text{otherwise} \tag{28}$$

DE Initialization steps are,
Step 1: Set DE optimization parameters such as population size, crossover constant, mutation constant, number of generations, and number of variables (for PID=3), Set upper and lower bounds for PID variables.
Step 2: Evaluate the fitness of each vector by randomly initializing all the vector populations within the specified upper and lower boundaries.
Step 3: Until the generation process is complete, optimization will continue. The first individual fitness value from the current population is designated as the objective vector. The target vector is crossed and mutated with the trial vector, which is generated by selecting 3 values at random from the current population. The trial vector's fitness value can be calculated by sending each trial vector to the PID controller independently.
Step 4: Verifying the boundary constraint.
Step 5: By comparing each target vector's fitness value to that of the trial vector, the selection is carried out for each target vector. For the next generation, the vector with the lower fitness value is selected.
Step 6: Repeat steps 3-5 until the new population is completed.
Step 7: Repeat step 6 until the end of generation. The optimization process is completed. The global minimum fitness value is achieved and these are the optimum parameters of the PID controller.

D. Ant Colony Optimization (ACO)

Marco Dorigo developed this technique in 1992 [2][11][51]. Ants can easily communicate with each other using Pheromones. Pheromones are chemical signals used by ants for communication in the environment and ants release pheromones in danger (to alert other ants for help). ACO algorithm is inspired by the social behavior of real ants. This technique is used to find optimal paths. With the help of pheromone signals, ants can easily find the shortest path. ACO algorithm initializes its parameters, and solution construction and positions each and in the starting node, each ant will select the next node by applying the state transition rule, repeat until the ant builds the best solution, then compute the fitness value, update best solution, apply offline pheromone update, display the best result.

The ants are driven by a probability rule to choose their solution to the problem. The probability rule between two nodes i and j, depends on two factors [11]. From equation (29), the factor η_{ij} is the inverse of the cost function. The element τ_{ij}, which is associated with pheromone and has an initial value of τ_0, is updated after each iteration and does not change while the algorithm is being executed. The user can instruct the algorithm to search in favor of the heuristic or the pheromone factor by using the parameters α and β.

$$p_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \epsilon s} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} \tag{29}$$

The change in pheromone quantity in each path is given by equation (30),

$$\Delta^A_{\tau_{ij}} = \begin{cases} \frac{L^{min}}{L^A} & \text{if, } i, j \epsilon T^A \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

L^A is the solution of the ant A and L^{min} is the best solution found so far. The pheromone for the next iteration is decided in equation (31). NA is the number of ants, ρ being the evaporation rate, designed to allow the elimination of bad choices.

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta_{\tau_{ij}}^{A}(t) \tag{31}$$

The limitation of this algorithm is that it uses a greater number of parameters and the advantage is that using fewer iterations provides the best solution [51]. ACO is used for temperature and humidity control of rooms [5]. It is also used for a level control system [32].

The ACO algorithm is as follows:
Step 1: Initialize ACO parameters
Step 2: Ant solution construction
Step 3: Position each and in the starting node. Step 4: Each ant will select the next node by applying the state transition rule.
Step 5: Repeat until the ant builds the best solution, then compute the fitness value.
Step 6: Update the best solution.
Step 7: Apply offline pheromone update.
Step 8: Display the best result.

E. Simulated Annealing (SA)

The idea for the simulated annealing technique came from the annealing process used on metals, which involves heating solid-state metal to a high temperature until it reaches a random state after which it gradually cools to reach thermal equilibrium, causing the atoms to arrange themselves in the ideal crystal's lowest energy state. Another name for this lowest energy state is the ground state. The metal is gradually cooled after being cautiously heated to this ground condition. Here, there is a decrease in the average potential energy per atom, which can be applied as a minimization technique. The purpose of this procedure is to clean the crystal of any defects. The material is allowed to cool more slowly and allowing less-than-ideal solutions expands the search space and increases the depth of the search for the optimal result. A probability distribution controls the search's depth, and an acceptance probability determines whether a poorer point is accepted by the algorithm. It is directly correlated with temperature and is sometimes used to determine a new search area in search of a better minimum. Stochastic/random, unorganized, and nonlinear optimization issues can all be solved with this approach. In 1953, Metropolis presented this algorithm for the first time, and in 1983, Kirkpatrick developed it [108]. It is based on the Boltzmann probability distribution shown in equation (32), where K is the Boltzmann constant, T is the temperature, and E is the energy, and the convergence of the algorithm can be regulated by changing the T [103].

$$P(\Delta E) = e^{(-\Delta E/KT)} \tag{32}$$

The distinguishing feature of SA over other approaches is its ability to avoid getting stuck at local minima. With different probabilities, the technique employs a random search that allows modifications that increase and decrease the objective function. Using the similarity between the annealing process—the cooling and freezing of metal into a minimum energy crystalline structure—and the search for a global optimum of a given function across a wide region. SA is a non-specific probabilistic metaheuristic approach. SA is capable of handling complex constraints, noisy and chaotic data, and highly nonlinear models. Its adaptability and capacity to approach global optimality are its key advantages over other local search techniques. The algorithm is highly versatile because it doesn't rely on any model limitations. This method works under the assumption that annealing will go on until the temperature drops to zero.

The SA algorithm evaluates the objective function after starting with a set of initial Kp, Ki, and Kd. The search space will be defined by the upper and lower bounds for each parameter of the controller, where the SA searches for the best fitness. The objective function is evaluated once more after the PID controller parameters are changed to create a new set of PID parameters.

F. Artificial Bee Colony (ABC)

Three types of bees are identified by ABC algorithm scouts, observers, and employed foragers [19]. Karaboga introduced the ABC algorithm. In most situations, a worker forager bonds with a single food source and shares it with other bees; scouts are in charge of finding new food sources, and by exchanging information with worker foragers, onlookers can locate food sources [103]. Here, L_d and U_d are the lower and upper bounds in the 'd' dimensional space, respectively, and rand (0, 1) is the random integer uniformly distributed in the range (0, 1). The location of the 'i' individual on the 't' iteration is $x_i(t)$, i.e. generated by equation (33). Professional foragers utilize equation (34), in which the random number < i, j $\leq$ M, i $\neq$ j, φ is uniformly distributed in the interval (0, 1). The dietary sources chosen by observers are determined by Equations (35), and (36).

$$x_{i,d}(t) = L_d + rand(0,1) * (U_d - L_d) \tag{33}$$

$$v_{i,d}(t+1) = x_{i,d}(t) + \varphi(x_{i,d}(t) - x_{j,d}(t)) \tag{34}$$

$$p_i = \frac{fit_i}{\sum_{i=1}^{N} fit_i} \tag{35}$$

$$\mathrm{fit_i} = \begin{cases} \frac{1}{1+f(i)}, & f(i) \geq 0 \\ 1 + abs(f(i)), & \text{otherwise} \end{cases} \tag{36}$$

The ABC algorithm eliminates a food source and updates the scout with the associated employed forager if it cannot be updated after limited times searches. The PID controllers Kp, Ki, and Kd are to be optimized parameters as honey sources with a specific performance index as the objective function to develop a new type of ABC-PID optimization method. Finding a set of Kp, Ki, and Kd parameters that will allow a system performance index to be as low as possible, a controlled quantity to quickly reach the desired target, and an algorithm with extremely low overshoot and short adjustment times are the main challenges of the artificial bee colony algorithm.

G. Firefly Algorithm (FA)

The flickering patterns of fireflies are used as an inspiration for FA. Each firefly moves towards brighter fireflies in search of better solutions since their brightness reflects their fitness value. FA assists in leading the search to areas of the parameter space that show potential. PID controller gains are tuned during the firefly algorithm's optimization to ensure the best possible control performance under typical operating conditions [22]. This algorithm was proposed by Yang Xin-She.

The three main processes are the firefly position update [103], the firefly Luminance 'I' update, and the firefly attraction degree β computation. Let us assume that the maximum brightness is I_0, the maximum attraction degree is β_0, the attraction factor is γ, and the step factor is a step. Equations (37) and (38) define luminance, and equation (39) defines the degree of attraction.

$$I = I_0 * exp(-\gamma * r_{i,j}) \tag{37}$$

$$r_{i,j} = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2} \tag{38}$$

$$\beta(r_{i,j}) = \beta_0 * exp(-\gamma * r_{i,j}^2) \tag{39}$$

$$x_i(t+1) = x_i(t) + \beta(r_{i,j}) * (x_j(t) - x_i(t) + step * \epsilon_i \tag{40}$$

The 'i' firefly's position updating as it approaches the 'j' firefly is described by equation (40). At iteration t, the positions of the 'i' and 'j' fireflies are denoted by $x_i(t)$ and $x_j(t)$, respectively. The uniformly distributed random number is ϵ_i.

The FA algorithm is trying to adjust the PID parameters Kp, Ki, and Kd using the simulation and the algorithm. With the objective function as its starting point, the FA algorithm first generates an initial random population, defines parameters, calculates luminous intensity and absorption, initializes the location of fireflies, tunes attractive parameters i towards j moves, calculates the new solutions, and updates luminous intensity.

H. Cuckoo Search (CS)

In 2009, Xin-She Yang and Suash Deb developed the cuckoo search algorithm. It is a population-based metaheuristic optimization approach [103]. In the CS technique, a single egg is laid by each cuckoo, which is dropped into a randomly chosen nest. The host bird would identify the cuckoo-deposited egg with a probability $p_a \in [0, 1]$; that is, the percentage p_a of 'M' nests would be replaced by the new nests, given a fixed number of probable host nests. Equation (41), in which α is the scaling factor equivalent to one of step size, is used by the 'i' individual to update its host nest $x_i(t)$. The $\otimes$ entrywise multiplications are represented by the product, and the Levy flight's derivation from the Levy distribution is indicated by $Levy(\lambda)$.

$$x_i(t+1) = x_i(t) + \alpha \otimes Levy(\lambda) \tag{41}$$

Since real Levy distribution is challenging to achieve, the Levy flight is typically calculated using equation (42), where u and v follow the uniform distribution, $u \sim N(0, \sigma^2)$, $v \sim N(0, 1)$, $\sigma = \left(\frac{\Gamma(1+\beta) sin(\pi\beta/2)}{\beta\Gamma(1+\beta)/2) 2^{(\beta-1)/2}}\right)^{1/\beta}$, $\beta = 1.5$.

$$s = \frac{u}{|v|^{1/\beta}} \tag{42}$$

There is a chance that the host may find some cuckoo eggs and reject them; this chance is p_a. In case of a lost cuckoo egg, it needs to find a new boarding place and update using equation (43), where H is a Heaviside function, ϵ is a random number taken from a uniform distribution, s is the step size, which is defined as a random number within the interval of (0, 1), and α is a scaling factor of step size.

$$x_i(t+1) = x_i(t) + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j(t) - x_k(t)) \tag{43}$$

The following are the design steps to get the best solution:
Step 1: Set the objective function, the maximum number of iterations, and the population initialization.
Step 2: Create a new cuckoo with an arbitrary, and use the Levy flying method to generate it. Then, find its new fitness for the proposed objective function for the parameter tuning problem.
Step 3: From the population that was created at random, select a nest and determine its objective function.
Step 4: If the fitness nest is reached, the new nest obtained with the Levy flight replaces the host nest.
Step 5: Levy flight leads to the creation of new nests at new sites, and some of the worst nests are abandoned.
Step 6: Determine each newly formed nest's objective function values.
Step 7: Update the current iteration's best nest.
Step 8: The best nest obtained in the current iteration.
Step 9: Follow steps 2-8 until the end of the stopping criteria. The ideal answer to the PID parameter is represented by the best nest that was found at the end of the iteration.

I. Harmony Search (HS)

The improvised nature of music is the inspiration for the Harmony Search optimization method. Each member of a band modifies their pitch to create a beautiful harmony. All of the choice variables continuously adjust their values during the global optimization process to help the objective function reach global optimization [93]. The HS method provides several outstanding benefits, including simple implementation, fewer configurable parameters, and speedy convergence. In 2001, Z. W. Geem, J. H. Kim, and G. Loganathan developed the Harmony Search algorithm [94],[95].

The harmony memory (HM) of this algorithm is sized by the harmony memory size (HMS) parameter, and the ideal harmony vectors are kept there during optimization [103]. A new harmony vector such as $\mathrm{x}_i^{'}(t) = (\mathrm{x}_1^{'}(t), \mathrm{x}_2^{'}(t), \mathrm{x}_3^{'}(t), ...\mathrm{x}_D^{'}(t))$ was generated from the HM based on randomization, pitch adjustments, and memory considerations. Using the harmony memory considering rate (HMCR) option, which ranges from 0 to 1, it chooses a new value. According to equation (44) if a randomly generated number r_1 is less than or equal to HMCR, then $\mathrm{x}_i^{'}(t)$, the definition space of the 'i' dimensional variable, would be updated from HM. Next, a value at random would be chosen from $\mathrm{x}_i^{'}(t)$. We analyze each component $\mathrm{x}_i^{'}(t)$ to see if pitch adjustment is necessary. The pitch adjusting rate (PAR), which determines the rate of adjustment for the pitch selected from the HM, is a parameter used in the method.

$$\mathrm{x}_i^{'}(t) = \begin{cases} \mathrm{x}_i^{j}(t) & j \in (1, 2, ...HMS) r_1 \leq HMCR \\ \mathrm{x}_i^{'}(t) \in x_i & else \end{cases} \tag{44}$$

Each component $\mathrm{x}_i^{'}(t)$ from equation (45) is analyzed to discover if pitch adjustment is necessary. One of the factors in the approach is the pitch adjusting rate (PAR), which establishes the rate of adjustment for the pitch chosen from the HM. α = BW * u(-1, 1), where u(-1, 1) is a uniform distribution between -1 and 1, and BW is an arbitrary distance bandwidth for continuous design variable, are represented by the numbers r_2 and α.

$$\mathrm{x}_i^{'}(t) = \begin{cases} \mathrm{x}_i^{'}(t) + \alpha & r_2 \leq PAR \\ \mathrm{x}_i^{'}(t) & otherwise \end{cases} \tag{45}$$

The HS algorithm is developed using the following steps:
Step 1: Set up the HS Memory (HM). A certain number of randomly generated solutions to the optimization problems under consideration make up the initial HM.
Step 2: Get the HM to improvise a novel solution. Based on the HM considering the rate, each component is obtained.
Step 3: Make HM updates. Step 2 revised solution is evaluated. It will take the position of the least fit member in the HM if its results are better. If not, it gets eliminated.
Step 4: Continue from Steps 2-3 until a predetermined termination criterion—such as the maximum number of iterations is reached.

Like swarm intelligence and GA algorithms, the HS technique uses a random search strategy [95]. It doesn't require any primary knowledge, such as how the objective function gradients work. However, it just uses one search memory to evolve, unlike those population-based methods. As a result, the HS approach stands out for its computational simplicity. The PID controller settings are best combined using the harmony search algorithm and its variations, where each harmony is made up of the three gains.

J. Grey Wolf Optimization (GWO)

GWO was proposed by Mirjaliali Mohammad and Lewis in 2014. This algorithm was developed by grey wolf hunting and social hierarchy technique [7]. The grey wolves lived in organized packs. The size of one pack is 5-12 and 4 various ranks of wolves in a pack. The leader of the pack is the alpha wolf it may be a male or a female and other members of the pack follow to alphas. The alpha wolves in the pack make decisions on sleeping places, time to wake up, hunting, etc. The beta wolf is the second level of the Grey wolf hierarchy. The beta wolf helps in the work of the alpha wolf. The beta wolf gives feedback to the alpha wolf. The delta wolf is the third level of the Grey wolf hierarchy. It dominates lower-ranking omega wolves. They provide food to the whole pack. The omega wolf plays a scapegoat role in the pack means a victim who is blamed for the mistakes or faults of others. They are the last wolves allowed to eat. The grey wolf pack faces internal problems and fighting if the omega wolf is not in the pack. Grey wolves hunt in several ways: by tracking, pursuing, and approaching their prey; by encircling and harassing them until they stop moving; and, lastly, by attacking them. The following is the GWO algorithm's mathematical model:

Step 1: Alpha is the best solution. Beta and Delta are the second and third-best solutions, respectively. Omega comes after these three wolves.
Step 2: Encircling behavior is modeled as:

$$\overrightarrow{D} = |\overrightarrow{C}\overrightarrow{Xp} - \overrightarrow{X}(t)| \tag{46}$$

$$\overrightarrow{X}(t+1) = |\overrightarrow{Xp}(t) - \overrightarrow{A}\overrightarrow{D}| \tag{47}$$

Where, t = current iterations, $\overrightarrow{Xp}$ = Position of the Prey, $\overrightarrow{X}$ = Position of Grey wolf, $\overrightarrow{A}$, $\overrightarrow{C}$ = Coefficient vectors, $\overrightarrow{A}$ = $2\overrightarrow{a}\ \overrightarrow{r1}$ - $\overrightarrow{a}$, and $\overrightarrow{C}$= 2 $\overrightarrow{r2}$, $\overrightarrow{r1}$, $\overrightarrow{r2}$ = Random vectors ranges (0,1) Component $\overrightarrow{a}$ linearly decrease from 2 to 0 throughout iterations.

- By changing the values of A and C, several locations around the optimal search agents can be reached to the current position.
- r_1, r_2 allows the wolf to reach any position between 2 particular points.

Step 3: Grey wolf Hunting:

- Grey wolves target the weak or older ones.

- Rather than take the chance of attacking a large animal that is eager to fight, wolves may decide to try alternative prey.
- Alpha directs the Hunting procedure.
- It is thought that Delta, Alpha, and Beta are more knowledgeable about where the prey is (optimal solution).
- As Alpha, Beta, and Delta's positions change, so will the positions of other wolves.

$$\overrightarrow{D\alpha} = |C_1\overrightarrow{X\alpha} - \overrightarrow{X}(t)|, and \overrightarrow{X1} = |\overrightarrow{X\alpha} - A_1\overrightarrow{D\alpha}| \quad (48)$$

$$\overrightarrow{D\beta} = |C_2\overrightarrow{X\beta} - \overrightarrow{X}(t)|, and \overrightarrow{X2} = |\overrightarrow{X\beta} - A_2\overrightarrow{D\beta}| \quad (49)$$

$$\overrightarrow{D\delta} = |C_3\overrightarrow{X\delta} - \overrightarrow{X}(t)|, and \overrightarrow{X3} = |\overrightarrow{X\delta} - A_3\overrightarrow{D\delta}| \quad (50)$$

- The Grey Wolf position is updated as:

$$\overrightarrow{X}(t+1) = \frac{(\overrightarrow{X1} + \overrightarrow{X2} + \overrightarrow{X3})}{3} \quad (51)$$

Step 4: Wolves complete the hunting process by attacking their prey when it stops moving.

- This is represented by a decrease in $\overrightarrow{a}$ from 2 to 0 during the iterations.
- As $\overrightarrow{a}$ decrease, $\overrightarrow{A}$ also decreases.
- $|A| < 1$ Makes the wolf charge in the direction of its prey.
- $|A| > 1$ Find a better prey by diverting from the current one.
- C Random value intervals (0,2) comprise vectors.
- C aids in adding weight to the prey which limits the wolves' ability to locate it.
- If $C > 1$: Emphasize.
- If $C < 2$: De-emphasize or reduce importance.

GWO parameters are the number of search agents or population, dimension, maximum interaction, lower bounds, and upper bounds are required to be set. The objective function is employed to lower error and illustrates how the PID controller improves system performance.

V. Comparative Analysis of Algorithms

In comparison with conventional algorithms, intelligent and nature-inspired algorithms have several benefits. They can effectively handle uncertainties, efficiently search a large search space, and adjust to shifting operational conditions. According to the needs of the control system, they can also be customized to meet particular performance objectives, such as minimizing transient characteristics such as settling time, overshoot, or steady-state error. Successful PID tuning still depends on choosing the best algorithm for a given control problem and making sure it is configured correctly. When implementing these algorithms in real-time control applications, it is crucial to take computational complexity and implementation considerations into account.

For example, the simulation results are shown in Table V [102] of the first order plus dead time model. It shows a comparison of some conventional methods in the time response form and confirms that the implemented Fuzzy-PID with a simple design approach and smaller rule base can provide better performance than ZN-PID, CC-PID, TL-PID, and IMC-PID. It can be observed that the least Tr of 2.32 sec is achieved using the CC tuning formula. However, this method was not recommended as it gave the largest Ts and Mp. Though the reduced Ts of 8.25 sec is reported in IMC but resulted overshoot of 2.24 percent which is larger than fuzzy and TL methods. Almost zero percent overshoot was obtained by fuzzy and TL but huge rise time and settling time by TL as compared to other tuning methods which were not acceptable.

TABLE V. First Order Plus Dead Time Model

Method	Mp	Ts	Tr
ZN	18	33.32	18
CC	28	25.48	2.32
TL	0.0693	193.78	81.32
IMC	2.24	8.25	3.027
Fuzzy PID	0	10.31	5.68

The closed-loop step response of the CSTR model is obtained from ZN-PID, FLC, ANN-PID, ANFIS-PID, and GA-PID and measures rise time, settling time, overshoot, undershoot, and steady-state error[16]. The comparison of controllers is shown in Table VI. It is observed that the poor performance is given by ZN-PID over intelligent PID except less rise time. The linear approximation used in the ZN technique may result in less-than-ideal performance in the CSTR process with nonlinear dynamics, which would mean insufficient adjustments. The performance of ANN-PID is better as compared to others except undershoot and overshoot. Still, there is scope for improvement of this system.

TABLE VI. Comparison of intelligent PID controller for CSTR

Parameters	ZN	Fuzzy	ANN	ANFIS	GA
Rise Time	1.789	1.865	2.98	2.578	4.84
Settling Time	3.745	5.624	6.85	3.425	7.12
Overshoot	20.05	17.95	10.47	1.0149	0
Undershoot	44.46	39.98	2.948	21.2	7.2923
Steady-state error	0	0	0	0	0

TABLE VII. COMPARISON OF INTELLIGENT AND NATURE-INSPIRED ALGORITHMS

Algorithms	**Complexity**	**Convergence Speed**	**Robustness**	**Adaptability**	**Performance Improvement**	**Less effective in**
FL	Moderate	Variable	Robust to uncertainty	Limited adaptation	Improved in some cases	Systems with highly dynamic and rapidly changing dynamics
ANN	High to Moderate	Moderate to Fast	Sensitive to noise	Adaptable	Enhanced in many cases	Systems with limited data, interpretability requirements
ANFIS	Moderate to High	Moderate to Fast	Robust to uncertainties	Adaptable	Enhanced in many cases	Systems with limited data, interpretability requirements
GA	High	Variable	Robust to local optima	Global exploration	Effective in many cases	Real-time applications, highly dynamic environments
PSO	Moderate to High	Fast	Susceptible to local optima	Global exploration	Effective in many cases	Systems with stringent safety constraints, highly nonlinear dynamics
DE	Moderate to High	Fast	Robust to local optima	Global exploration	Effective in many cases	Systems with stringent safety constraints, highly nonlinear dynamics
ACO	High	Moderate to Slow	Robust to local optima	Global exploration	Effective in many cases	Real-time applications, large-scale problems
SA	Moderate to High	Slow to Moderate	Ability to escape local optima	Adaptable	Effective in some cases	Real-time applications, systems requiring rapid adaptation
ABC	Moderate to High	Moderate to Fast	Limited exploration capability	Global exploration	Effective in some cases	Systems with complex and irregular solution spaces
FA	Moderate to High	Fast	Limited exploration capability	Adaptable	Effective in some cases	Highly dynamic environments, systems with rapid changes
CS	Moderate	Fast	Limited exploration capability	Adaptable	Effective in some cases	Systems with strict performance requirements
HS	Moderate to High	Moderate to Slow	Global search capability	Adaptable	Effective in some cases	Rapidly changing solution spaces
GWO	Moderate	Fast	Limited exploration capability	Adaptable	Effective in some cases	Multimodal landscapes with widely separated optima

Table VII provides a comparative overview of various intelligent and nature-inspired algorithms in terms of different criteria relevant to industrial control applications. It's important to note that the data in the table is general and collected by referring to various references. The effectiveness of each algorithm depends on the specific characteristics and requirements of the control system in question.

The computational complexity can vary based on algorithmic parameters, problem-specific characteristics, and the specific implementation details. While these categorizations provide a general perspective, the actual performance may depend on the specific application and problem being addressed.

The convergence speed represents how fast the algorithm converges to a solution. The convergence speed depends on various factors, including algorithm parameters, problem complexity, and the nature of the optimization landscape. Some algorithms may exhibit different convergence speeds for different problem types or characteristics.

The robustness reflects the algorithm's ability to handle uncertainties and disturbances. The robustness of an algorithm is often influenced by the nature of the problem, the quality of the optimization landscape, and the algorithm's adaptability to variations in parameters.

The adaptability shows how well the algorithm adapts to varying operating conditions. Adaptability is context-dependent and can vary based on the specific problem, optimization landscape, and algorithm parameters. High adaptability does not necessarily mean superior performance in all scenarios, as trade-offs with other factors, such as convergence speed and solution quality, may exist.

The performance improvement provides an overall assessment of how much the algorithm improves the PID controller's performance. The level of improvement depends on the specific characteristics of the optimization problem, including its dimensionality, complexity, and the nature of the fitness landscape. Additionally, research and advancements in optimization techniques continue to evolve, leading to further improvements in PID tuning using intelligent and nature-inspired methods [103].

VI. Real-World Applications

High nonlinear constraint optimization issues make up the majority of real-world applications. Some of the earlier created conventional optimization strategies fail to tackle those real-life challenges, and the majority of answers are found by applying intelligent and nature-inspired algorithm methods. Because nature-inspired algorithms are typically simple and adaptable, researchers worldwide have long been attracted by the creation of some advanced intelligent algorithms and their utility in effectively handling a wide variety of challenging engineering issues. Some applications from different literature reviews are mentioned as follows.

The various conventional, intelligent, and nature-inspired tuning methods of PID controllers are presented in a survey paper on a review of PID control, tuning methods, and applications [10]. It shows growth in PID controller year by year. Intelligent tuning techniques facility available with PID controller gives more attention to applications of PID controller for process control systems from the industrial users.

In a comparative analysis of tuning a PID controller using intelligent methods [16], the results obtained by ANFIS achieved good performance of the system in settling time and overshoot because ANN gives a moderate performance of the system. But ANN reduces the overshoot and undershoot in comparison with the ZN method. The choice of control approach depends on the specific characteristics of the CSTR process, including its nonlinearity, uncertainty, and dynamic nature. ANFIS's success in settling time and overshooting can be attributed to its ability to integrate fuzzy logic for uncertainty handling and neural networks for adaptive learning. ANN, while also capable of handling nonlinearity and learning from data, may not capture uncertainty as explicitly as ANFIS. The ZN method, on linear stability analysis, may not be as effective in adapting to the nonlinear and uncertain dynamics of the CSTR process.

Design optimized PID for speed control of brushless DC motor using PSO [1]. The rotor position in the DC motor is determined by changing back emf. The time response specifications of the system such as delay time, rise time, settling time, peak time, peak overshoot, and steady-state error measured from results obtained by PSO and compared with conventional methods. The improved results obtained through PSO in case of sudden change in loads and reference speed. PSO produces better and more robust performance, in motor speed control.

Artificial intelligence techniques such as FLC, GA, and PSO are applied for DC motor speed control [3]. Compared their performance to each other and with conventional PID for motor speed control. The unexpected maximum overshoot and slow speed are reduced by intelligent techniques than conventional techniques. Intelligent techniques are good and fast and solve the problems associated with conventional PID. PSO-PID is a very good technique for DC motor speed control purposes. PSO is often considered a very good technique for DC motor speed control purposes compared to FLC and GA due to its specific strengths in optimization tasks and system tuning.

The PID controller is designed using both intelligent and conventional methods [4]. The sensor data is utilized to determine the system's control objectives. Data from the sensors measured during the controlled process is used to intelligently adjust the control system's parameters as compared with the conventional method for PID design. Using GA, the system parameters are calculated based on a differential equation. For effective process control, use a fuzzy expert system to design a controller. To adjust the PID and higher-order systems that these modules controlled, the ultimate intelligent controller was implemented.

Ant colony optimization (ACO) and symbiotic organism search (SOS) algorithms were developed for PID tuning for temperature and relative humidity control of rooms and obtained a response. SOS is a natural philosophy inspired by the behavior of reactions between organisms living in nature. The SOS algorithm has two control parameters ecosize represents the number of organisms and maximum function evaluation represents the maximum number of iterations. As per time response specifications it shows that in temperature control, the ACO responds quickly to a change in a set point because the ACO algorithm is less affected by poor initial conditions whereas the SOS performs better in relative humidity management. However, the conventional methods are slow. ACO responsiveness and adaptability suit the dynamic nature of temperature control, while SOS's ability to handle multimodality makes it effective for relative humidity management. The limitations of conventional methods in terms of convergence speed and adaptability further emphasize the need for advanced optimization techniques in complex control scenarios.

Developed a new hybrid PSO and GA method controlled by fuzzy logic [6]. This paper focused on the optimization problem of the PSO algorithm and it was solved by developing a hybrid algorithm. In PSO particle is modified from its current position to its best position and it is found throughout the swarm. But PSO has less search efficiency. So improving it with a hybrid algorithm means PSO combines with other algorithms. For example, PSO and GA combined in two ways, GA population was used to start PSO, or PSO swarm was used as the initial population of GA. The impact of hybrid PSO-GA in the searching process is determined by the influence factor which is automatically selected using a fuzzy system. This hybrid algorithm tested for well-known unimodal and multimodal benchmark functions. The results obtained for benchmark functions demonstrate are better than the analyzed algorithm [6].

Design an intelligent controller for the level control of water tanks [8] by using GA and PSO algorithms. The control system performance is improved in terms of tr, ts, steady state, and Mp by using a PSO/GA PID controller as compared with conventional methods due to its intelligent structure. PSO and GA provide advantages in optimizing PID parameters for water tank level control due to their global optimization capabilities, adaptability to system changes, and efficient exploration-exploitation trade-off. These characteristics make them well-suited for addressing the complexities and nonlinearities often present in water tank systems.

To determine the input and output scaling factors for the interval fuzzy PID controller, many algorithms are applied to the servo system, including GA, CS, PSO, DE, Bee colony, and combined PSO and DE algorithms [23]. The servo tracking selects these elements. Reduce the performance index to choose the best scaling factor values. Measurement noise, set point tracking, and load disturbances are used to compare the performances. The hybrid method yields optimal values, which are then used by interval fuzzy PID to achieve increased performance.

Designed fuzzy PID for a standard second-order system and compared the transient characteristics of the system with conventional methods like ZN and relay auto-tune and obtained better results with fuzzy PID than conventional one [31]. Fuzzy logic is an effective tool for integrating intelligence because it can express progressive knowledge in a way that is consistent with human thought processes. Neural networks are powerful intelligent tools. Artificial neural networks are better at handling fast-paced scenarios like real-time communication and autonomous vehicle operation because they can quickly analyze and deconstruct complicated data patterns.

In a survey paper on tuning of PID controllers for industrial processes using soft computing techniques Divya and Nirmal Kumar [65] examined several methods of soft computing for fine-tuning PID controller settings. Five soft computing techniques—Neural Networks, EP, GA, PSO, and ACO methods—were examined in the review. The survey provided flowcharts and basic highlights of a few chosen algorithms. The implementation details of a few chosen algorithms were withheld.

GWO-PID controller is used for controlling the ball hoop system [7]. In this method, consider many intelligent swarm methods and apply them to control the ball hoop system. But in these methods' the leader does not control the entire period. This problem is solved by using the GWO algorithm. The objective functions are minimized with the help of the GWO for obtaining the optimal parameters of PID for controlling the ball hoop system. The results obtained are compared to each other. The comparison shows that the GWO method gives a better solution than other techniques for the ball hoop system. The peak overshoot and settling time are less for all of the objective functions by the GWO algorithm as compared with other techniques due to its less computational complexity, simplicity in nature, and easy programming, etc.

VII. Challenges and Limitations

The limitations of fuzzy systems are completely dependent on human knowledge and expertise. Also, it requires regular updates to the rules of a fuzzy logic control system. The high value of overshoot and undershoot was obtained in the CSTR process step response [16]. When a high number of elements are subjected to mutation, the size of the search space increases exponentially, indicating that genetic algorithms do not scale well with complexity. Because of this, applying the method to issues like constructing PID is quite challenging.

There are some challenges in using ANNs for PID tuning in dynamical systems such as selecting the right ANN design, activation function, learning algorithm, and error function, as well as the requirement for an adequate and representative amount of data to train the ANN [37].

However, because complicated issues include a greater number of decision variables, the main limitation of PSO algorithms is their significant processing time. The number of steps inside also increases the computational complexity throughout the optimization process [6]. The solution is a hybrid controller.

In the optimal tuning of PID for the DC motor via simulated annealing, it is necessary to properly set the cooling schedule in the SA algorithm. A local optimum may be reached by the method if it is too rapid; a significant convergence time may be experienced if it is too slow. To find an appropriate solution, particularly for complicated situations, a lot of iterations may be required [86].

The ABC algorithm for the second order system suffers from low population variety, great equation-searching ability but poor developing capacity, and slow global convergence due to poor solution quality and local optimally [88]. Therefore, the global crossover mechanism is added to the domain search of the improved artificial bee colony algorithm (IABC) to improve its direction and prevent premature convergence of the roulette mechanism. Additionally, the population's diversity is increased and the global search function is enhanced through chaotic avoiding search.

The limitations of the typical FA are high computational time complexity and slow convergence speed, among others for servo processes. The primary cause is that the FA uses a fully attracted model, which causes each firefly to move around while it flies [22]. The slow convergence rate, poor local searching capabilities, and low solving accuracy are some of the limitations of the GWO algorithm that will affect on control of the ball hoop system [7].

Overall, tuning PID controllers using intelligent and nature-inspired algorithms has proven to be a valuable and promising approach, enabling engineers and researchers to enhance control system performance in diverse and challenging applications. It's crucial to remember that the effectiveness of these algorithms can change based on the particular control system and its specifications. To inspire more research and give readers and researchers insight into a variety of nature-inspired algorithms, the issues of stability of nature-inspired algorithms, the theoretical framework of parameter tuning and control, the solutions of large-scale real-life applications, and the advancements made in handling non-deterministic nature problems are addressed.

Also, advances in PID tuning using smart and nature-inspired techniques are made possible by ongoing research and advancements in optimization methodologies. Considering the achievements of modern nature-inspired algorithms, several significant problems remain unsolved. Although the fundamentals of how heuristic algorithms work are widely established, the reasons behind their operation are only partially understood. After some interesting earlier attempts, it is unfortunately challenging to mathematically explain why these algorithms are so successful. These are unsolved research problems.

The optimization method's drawbacks include the possibility of requiring more computing resources, the possibility of being dependent on the objective function and algorithm of choice, and the risk of running into local optima or convergence challenges.

VIII. Future Directions and Research Opportunities

It is yet uncertain how much research has been done on the basic theories, mathematical foundations, and tools for nature-inspired algorithms in the field of technology. For example, to investigate further algorithms in control, GA, PSO, and DE must conduct additional studies. This review paper is an effort to document the development in the field of PID tuning for the novice reader and the future of PID.

Furthermore, as all algorithms work in black-box mode, researchers are always coming up with what they call unique algorithms and claiming that their optimizers find greater value than others. Due to the insufficient and inadequate theoretical exploration of algorithms inspired by nature, it might be challenging to discern the distinct features of many algorithms due to their striking similarity. This motivates researchers to compare and analyze nature-inspired algorithms theoretically and experimentally.

In complicated situations, the algorithms that take inspiration from nature are not as effective in handling continuous optimization challenges. As a result, fitness evaluations could include noise, be tedious and imprecise, and occasionally have unpredictable fitness functions.

Every algorithm has some restrictions, thus combining the current method with another is necessary to increase its performance. The changes must be made to several recently proposed algorithms to meet dynamic optimization problems. It is necessary to adjust the settings of algorithms when addressing optimization problems.

Existing applications, like PID control systems, have focused on designing algorithms using nature-inspired algorithms. Many of these applications either apply algorithms directly or are unable to justify their design in terms of particular issues.

IX. Conclusion

The past three decades have witnessed a significant rise in the utilization of diverse nature-inspired algorithms for PID controller tuning in industrial applications. This paper provides a concise overview and a unified formal definition that encompasses conventional, intelligent, and nature-inspired approaches. Despite the vast literature on tuning PID controllers for industrial applications, our focus is narrowed to specific methods and applications. Through meticulous evaluation of their industrial performance, we identify key methodological elements aimed at enhancing efficacy. Despite the widespread adoption of various PID tuning methods, our analysis reveals a critical knowledge gap, emphasizing the necessity for adaptive algorithmic solutions to address dynamic optimization problems. The growing interest in intelligent PID and nature-inspired algorithms underscores the urgency of overcoming limitations inherent in conventional PID methods. This research serves as a comprehensive guide for practitioners and researchers selecting tailored algorithms, with a specific focus on adaptability and dynamic optimization in the evolving industrial landscape. Looking ahead, the continual evolution of PID control, particularly through nature-inspired algorithms, remains a significant research area. This paper lays the groundwork for further exploration, propelling advancements in PID tuning methodologies. By addressing current gaps and challenges, our findings pave the way for future research, ensuring ongoing relevance and innovation in this ever-evolving field.

References

[1] M. K. Merugumalla, and P. K. Navuri, "Optimized PID controller for BLDC motor using Nature-inspired Algorithms," *International Journal of Applied Engineering Research*, vol. 12, no. 1, 2017.

[2] S. Ghosal, R. Darbar, B. Neogi, A. Das, and D. N. Tibarewala, "Application of Swarm Intelligence Computation Techniques in PID Controller Tuning: A Review," In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications*, vol. 132, pp. 195–208, 2012. doi: 10.1007/978-3-642-27443-5_23.

[3] H. S. Purnama, T. Sutikno, S. Alavandar and A. C. Subrata, "Intelligent Control Strategies for Tuning PID of Speed Control of DC Motor - A Review," *2019 IEEE Conference on Energy Conversion* (CENCON), pp. 24-30, 2019, doi: 10.1109/CENCON47160.2019.8974782.

[4] J. Nowakova, and M. Pokorny, "Intelligent Controller Design by the Artificial Intelligence Methods," *Sensors*, vol. 20, no. 16, pp. 4454, 2020, doi: 10.3390/s20164454.

[5] K. Janprom, W. Permpoonsinsup, S. Wangnipparnto, "Intelligent Tuning of PID Using Metaheuristic Optimization for Temperature and Relative Humidity Control of Comfortable Rooms," *Journal of Control Science and Engineering*, pp. 1–13, 2020, doi: 10.1155/2020/2596549.

[6] P. Dziwinski and L. Bartczuk, "A New Auto Adaptive Fuzzy Hybrid Particle Swarm Optimization and Genetic Algorithm," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 2, pp. 95–111, 2020, doi: 10.2478/jaiscr-2020-0007.

[7] N. Jain, G. Parmar, R. Gupta, and I. Khanam, "Performance evaluation of GWO/PID approach in control of ball hoop system with different objective functions and perturbation," *Cogent Engineering*, vol. 5, no. 1, 2018, doi: 10.1080/23311916.2018.1465328.

[8] M. Sharma, P. Verma and L. Mathew, "Design an intelligent controller for a process control system," *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, pp. 217-223, 2016, doi: 10.1109/ICICCS.2016.7542302.

[9] Ü. Onen, A. Cakan, and I. Ilhan, "Performance comparison of optimization algorithms in LQR controller design for a nonlinear system," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27: no. 3, pp. 1938–1953, 2019, doi: 10.3906/elk-1808-51.

[10] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, pp. 818–827, 2021, doi: 10.1007/s40435-020-00665-4.

[11] H. O. Bansal, and R. Sharma, P. R. Shreeraman, "PID Controller Tuning Techniques: A Review," *Journal of Control Engineering and Technology (JCET)*, vol. 2, pp. 168-176, 2012.

[12] S. K. Pandey, K. Veeranna, B. Kumar and K. U. Deshmukh, "A Robust Auto-tuning Scheme for PID Controllers," *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 47-52, 2020, doi: 10.1109/IECON43393.2020.9254382.

[13] W. V. Jahnavi, Dr. J. N. C. Sekhar, Dr. A. S. Reddy, "A Review on PID Controller Tuning Using Modern Computational Algorithms," *Journal of Emerging Technologies and Innovative Research*, vol. 10, no. 8, 2023.

[14] L. J. D. S. Moreira, G. A. Junior, and P. R. Barros, "Time and Frequency Domain Data-driven PID Iterative Tuning," *IFAC-Papers Online*, vol. 51, no. 15, pp. 1056-1061, 2018, doi: 10.1016/j.ifacol.2018.09.054.

[15] Karl J. Astrom and Tore Hagglund, "*PID Controllers*," Copyright, 1995 Instrument Society of America, 2nd Edition, ISBN 1-5561 7-5 16-7, 1995.

[16] V. Chopra, S. K. Singla, and L. Dewan, "Comparative Analysis of Tuning a PID Controller using Intelligent Methods," *Acta Polytechnica Hungarica*, vol. 11, no. 8, 2014, doi: 10.12700/aph.11.08.2014.08.13.

[17] V. Dubey, H. Goud, and P. C. Sharma, "Role of PID Control Techniques in Process Control System: A Review," In *Data Engineering for Smart Systems*, vol. 238, pp. 659–670, 2022, doi: 10.1007/978-981-16-2641-8_62

[18] S. Sultaniya and Dr. R. Gupta, "Design of PID Controller using PSO Algorithm for CSTR System," *International Journal of Electronic and Electrical Engineering*, vol. 7, no. 9, pp. 971-977, 2014.

[19] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459–471, 2007, doi: 10.1007/s10898-007-9149-x.

[20] C. B. Kadu and C.Y.Patil, "Design and Implementation of Stable PID Controller for Interacting Level Control System," In *7th International Conference on Communication, Computing and Virtualization, Procedia Computer Science*, vol. 79, pp. 737–746, 2016, doi: 10.1016/j.procs.2016.03.097.

[21] B. M. Sarif, D. V. A. Kumar, M. V. G. Rao, "Comparison Study of PID Controller Tuning using Classical/Analytical Methods," *International Journal of Applied Engineering Research*, vol. 13, no. 8, pp. 5618-5625, 2018.

[22] N. A. Selamat, T. O. Ramih, A. R. Abdullah, and M. S. Karis, "Performance of PID Controller Tuning based on Particle Swarm Optimization and Firefly Algorithm," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 3S2, 2019, doi: 10.35940/ijrte.C1042.1083S219.

[23] R. R. D. Maity, R. K. Mudi, and C. Dey, "Nature-inspired and hybrid optimization algorithms on interval Type-2 fuzzy controller for servo processes: a comparative performance study," *Applied Sciences*, vol. 2, no. 7, 2020, doi: 10.1007/s42452-020-3024-5.

[24] M. G. M. Abdolrasol, S. M. S. Hussain, T. S. Ustun, M. R. Sarker, M. A. Hannan, R. Mohamed, J. A. Ali, S. Mekhilef, and A. Milad, "Artificial Neural Networks Based Optimization Techniques: A Review," *Electronics*, vol. 10, no. 21, 2021, doi: 10.3390/electronics10212689.

[25] I. Juniku, and P. Marango, "PID design with bio-inspired intelligent algorithms for high order systems," *International Journal of Mathematics and Computers in Simulation*, vol. 9, 2015.

[26] M. S. Amiri, R. Ramli, M. F. Ibrahim, D. A. Wahab, and N. Aliman, "Adaptive Particle Swarm Optimization of PID Gain Tuning for Lower-Limb Human Exoskeleton in Virtual Environment," *Mathematics*, vol. 8, no. 11, p. 2040, 2020. doi: 10.3390/math8112040.

[27] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Science Direct*, vol. 8, no. 5, 2022. doi: 10.1016/j.heliyon.2022.e09399.

[28] E. H. E. Bayoumi, and Z. A. Salmeen, "Practical swarm intelligent control brushless DC motor drive system using GSM Technology," *WSEAS Transactions on Circuits and Systems*, vol .13, pp. 188–201, 2014.

[29] S. Darvishpoor, A. Darvishpour, M. Escarcega, and M. Hassanalian, "Nature-Inspired Algorithms from Oceans to Space: A Comprehensive Review of Heuristic and Meta-Heuristic Optimization Algorithms and Their Potential Applications in Drones," *Drones*, vol. 7, no. 7, 2023, doi: 10.3390/drones7070427.

[30] X. -K. Wang, X. -H. Yang, G. Liu and H. Qian, "Adaptive Neuro-Fuzzy Inference System PID controller for SG water level of nuclear power plant," *2009 International Conference on Machine Learning and Cybernetics*, pp. 567-572, 2009, doi: 10.1109/ICMLC.2009.5212517.

[31] S. M. Sam and T. S. Angel, "Performance optimization of PID controllers using fuzzy logic," *IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, pp. 438-442, 2017, doi: 10.1109/ICSTM.2017.8089200.

[32] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, "Comparison of PID controller tuning methods: analytical/classical techniques versus optimization algorithms," *2017 18th International Carpathian Control Conference (ICCC)*, pp. 533-538, 2017, doi: 10.1109/CarpathianCC.2017.7970458.

[33] C. -T. Chao, N. Sutarna, J. -S. Chiou, and C. -J. Wang, "An Optimal Fuzzy PID Controller Design Based on Conventional PID Control and Nonlinear Factors," *Applied Sciences*, vol. 9, no. 6, 2019, doi: 10.3390/app9061224.

[34] K. Yan and H. Mo, "Application of fuzzy control under time-varying universe in unmanned vehicles," *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation* (YAC), pp. 439-444, 2018, doi: 10.1109/YAC.2018.8406414.

[35] A. Al-Gizi, A. Craciunescu and S. Al-Chlaihawi, "Improving the performance of PV system using genetically-tuned FLC based MPPT," *2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical*

Machines and Power Electronics (ACEMP), pp. 642-647, 2017, doi: 10.1109/OPTIM.2017.7975041.

[36] S. Bari, S. S. Zehra Hamdani, H. U. Khan, M. u. Rehman and H. Khan, "Artificial Neural Network Based Self-Tuned PID Controller for Flight Control of Quadcopter," *2019 International Conference on Engineering and Emerging Technologies* (ICEET), pp. 1-5, 2019, doi: 10.1109/CEET1.2019.8711864.

[37] R. Kumar, S. Srivastava and J. R. P. Gupta, "Artificial Neural Network based PID controller for online control of dynamical systems," *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1-6, 2016, doi: 10.1109/ICPEICES.2016.7853092.

[38] Subeekrishna M P, Aseem K, "Comparative study of PID and fractional order PID controllers for industrial applications," In *International Journal of Engineering Research and Technology (IJERT)*, vol. 7, no. 1, 2019.

[39] M. Maryam, M. Haghparast, and F. Nasiri, "Air Condition's PID Controller Fine-Tuning Using Artificial Neural Networks and Genetic Algorithms," *Computers*, vol. 7, no. 2, 2018, doi: 10.3390/computers7020032.

[40] R. H. Alvaro, L. G. G. -Valdovinos, T. S. -Jimenez, A. G. -Espinosa, and F. F. -Navarro, "Neural Network-Based Self-Tuning PID Control for Underwater Vehicles," *Sensors*, vol. 16, no. 9, 2016, doi: 10.3390/s16091429.

[41] Zakiah Mohd Yusoff, Zuraida Muhammad, Amar Faiz Zainal Abidin, Mohd Azri Abdul Aziz, Nurlaila Ismail, Mohd Hezri Fazalul Rahiman, "Self-tuning fuzzy PID controller using online method in essential oil extraction process," *Proceedings of the 6th International Conference on Computing and Informatics, ICOCI 2017*, 25-27 April 2017.

[42] C. C. -Tang, N. Sutarna, J. -S. Chiou, and C. -J. Wang, "Equivalence between Fuzzy PID Controllers and Conventional PID Controllers," *Applied Sciences*, vol. 7, no. 6, 2017, https://doi:10.3390/app7060513.

[43] A. Y. Al-Maliki and K. Iqbal, "FLC-based PID controller tuning for sensorless speed control of DC motor," *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 169-174, 2018, doi: 10.1109/ICIT.2018.8352171.

[44] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and A. Abraham, "Neural network and fuzzy system for the tuning of Gravitational Search Algorithm parameters," *Expert Systems with Applications*, vol. 102, pp. 234-244, 2018, doi: 10.1016/j.eswa.2018.02.026.

[45] S. A. Hamoodi, I. I. Sheet and R. A. Mohammed, "A Comparison between PID controller and ANN controller for speed control of DC Motor," *2019 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPCE)*, pp. 221-224, 2019, doi: 10.1109/ICECCPCE46549.2019.203777.

[46] K. Sharma and D. K. Palwalia, "A modified PID control with adaptive fuzzy controller applied to DC motor," *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, pp. 1-6, 2017, doi: 10.1109/ICOMICON.2017.8279151.

[47] D. Babunski, J. Berisha, E. Zaev and X. Bajrami, "Application of Fuzzy Logic and PID Controller for Mobile Robot Navigation," *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-4, 2020, doi: 10.1109/MECO49872.2020.9134317.

[48] M. A. Khan, P. Anand and G. Bhuvaneswari, "Artificial Neural Network based controller design for SMPS," *2019 3rd International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, pp. 253-259, 2019, doi: 10.1109/RDCAPE47089.2019.8979104.

[49] Z. Guan and T. Yamamoto, "Design of a Reinforcement Learning PID controller," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-6, 2020, doi: 10.1109/IJCNN48605.2020.9207641.

[50] V. Vagisha, S. Swati, S. Das, S. K. Mishra and S. S. Sahu, "A Review on Intelligent PID Controllers in Autonomous Vehicle," *Advances in Smart Grid Automation and Industry 4.0*, vol. 693, 2021, doi: 10.1007/978-981-15-7675-1_39.

[51] G. V. Batista, C. T. Scarpin, J. E. Pécora, and A. Ruiz, "A New Ant Colony Optimization Algorithm to Solve the Periodic Capacitated Arc Routing Problem with Continuous Moves," *Mathematical Problems in Engineering*, vol. 2019, pp. 1-12, 2019, doi: 10.1155/2019/3201656.

[52] Z. Shen, J. Zhou, J. Gao and R. Song, "Fuzzy logic based PID control of a 3 DOF lower limb rehabilitation robot," *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 818-821, 2018, doi: 10.1109/CYBER.2018.8688089.

[53] Š. Bucz and A. Kozakova, "Advanced Methods of PID Controller Tuning for Specified Performance," *PID Control for Industrial Processes. InTech*, 2018, doi: 10.5772/intechopen.76069.

[54] R. Sharma, P. Gaur, S. Bhatt, and D. Joshi, "Optimal fuzzy logic-based control strategy for lower limb rehabilitation exoskeleton," *Applied Soft Computing*, vol. 105, p. 107226, 2021, doi: 10.1016/j.asoc.2021.107226.

[55] G. S. Hocaoğlu, N. Çavli, E. Kılıç and Y. Danayiyen, "Nonlinear Convergence Factor Based Grey Wolf Optimization Algorithm and Load Frequency Control," *2023 5th Global Power, Energy and Communication Conference (GPECOM)*, pp. 282-287, 2023, doi: 10.1109/GPECOM58364.2023.10175823.

[56] S. Chatterjee and S. A. Banday, "Comparative Analysis of Different Optimization Technique Based PID Controller for Isolated Microgrid," *2022 International Mobile and Embedded Technology Conference (MECON)*, pp. 325-330, 2022, doi: 10.1109/MECON53876.2022.9751929.

[57] H. Wang, and J. Lu, "Research on Fractional Order Fuzzy PID Control of the Pneumatic-hydraulic Upper Limb Rehabilitation Training System Based on PSO," *International Journal of Control, Automation and Systems*, vol. 20, pp. 310–320, 2022, doi: 10.1007/s12555-020-0847-1.

[58] H. Housny, E. A. Chater and H. E. Fadil, "Fuzzy PID Control Tuning Design Using Particle Swarm Optimization Algorithm for a Quadrotor," *2019 5th International Conference on Optimization and Applications (ICOA)*, pp. 1-6, 2019, doi: 10.1109/ICOA.2019.8727702.

[59] D. H. Kusuma, M. Ali and N. Sutantra, "The comparison of optimization for active steering control on vehicle using PID controller based on artificial intelligence techniques," *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, pp. 18-22, 2016, doi: 10.1109/ISEMANTIC.2016.7873803.

[60] X. Lu, X. Zhang, S. Jia and J. Shan, "Design of Quadrotor Hovering Controller Based on Improved Particle Swarm Optimization," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 414-417, 2017, doi: 10.1109/ISCID.2017.196.

[61] J. Seekuka, R. Rattanawaorahirunkul, S. Sansri, S. Sangsuriyan and A. Prakonsant, "AGC using Particle Swarm Optimization based PID controller design for two area power system," *2016 International Computer Science and Engineering Conference (ICSEC)*, pp. 1-4, 2016, doi: 10.1109/ICSEC.2016.7859951.

[62] D. K. Lal, A. K. Barisal, and M. Tripathy, "Grey Wolf Optimizer Algorithm Based Fuzzy PID Controller for AGC of Multi-area Power System with TCPS," *Procedia Computer Science*, vol. 92, pp. 99-105, 2016, doi: 10.1016/j.procs.2016.07.329.

[63] N. C. Patel, M. K. Debnath, D. P. Bagarty, and P. Das, "GWO tuned multi degree of freedom PID controller for load frequency control," *International Journal of Engineering and Technology*, vol. 7, no. 2, pp. 548-552, 2018, doi: 10.14419/ijet.v7i2.33.14831.

[64] R. k. Kuri, D. Paliwal and D. K. Sambariya, "Grey Wolf Optimization Algorithm based PID controller design for AVR Power system," *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 233-237, 2019, doi: 10.1109/PEEIC47157.2019.8976641.

[65] N. Divya, and A. Nirmalkumar, "A survey on tuning of PID controller for industrial process using soft computing techniques," *International Journal Pure and Applied Mathematics*, vol. 118, no. 11, pp. 663-667, 2018, doi: 10.12732/ijpam.v118i11.85.

[66] A. -S. A. Younis, A. M. Moustafa and M. Moness, "Experimental Benchmarking of PID Empirical and Heuristic Tuning for Networked Control of Double-tank System," *2019 15th International Computer Engineering Conference (ICENCO)*, pp. 162-167, 2019, doi: 10.1109/ICENCO48310.2019.9027382.

[67] S. Yahya, A. R. Al Tahtawi, K. Wijayanto and B. A. Faizah, "Liquid Flow Control Design Based on PID-Fuzzy Controller with anti-Windup Compensator," *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 7-12, 2020, doi: 10.1109/ICITACEE50144.2020.9239237.

[68] M. J. Blondin, and P. M. Pardalos, "A holistic optimization approach for inverted cart-pendulum control tuning," *Soft Computing*, vol. 24, no. 6, pp. 4343–4359, 2020, doi: 10.1007/s00500-019-04198-7.

[69] G. Chen, Z. Li, Z. Zhang and S. Li, "An Improved ACO Algorithm Optimized Fuzzy PID Controller for Load Frequency Control in Multi Area Interconnected Power Systems," in *IEEE Access*, vol. 8, pp. 6429-6447, 2020, doi: 10.1109/ACCESS.2019.2960380.

[70] S. S. Choong, L. P. Wong, and C. P. Lim, "A dynamic fuzzy-based dance mechanism for the bee colony optimization algorithm," *Computing Intelligence*, vol. 34, no. 4, pp. 999–1024, 2018, doi: 10.1111/coin.12159.

[71] E. M. El-Gendy, M. M. Saafan, M. S. Elksas, S. F. Saraya, and F. F. G. Areed, "Applying hybrid genetic–PSO technique for tuning an adaptive PID controller used in a chemical process," *Soft Computing*, vol. 24, no. 5, pp. 3455–3474, 2020, doi: 10.1007/s00500-019-04106-z.

[72] P. Dutta and S. K. Nayak, "Grey wolf optimizer based PID controller for speed control of BLDC motor," *Journal of Electrical Engineering and Technology*, vol. 16, pp. 955–961, 2021, doi: 10.1007/s42835-021-00660-5.

[73] K. Anbumani, R. Ranihemamalini and G. Pechinathan, "GWO based tuning of PID controller for a heat exchanger process," *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, pp. 417-421, 2017, doi: 10.1109/SSPS.2017.8071631.

[74] H. Zhang, W. Assawinchaichote and Y. Shi, "New PID Parameter Autotuning for Nonlinear Systems Based on a Modified Monkey–Multiagent DRL Algorithm," in *IEEE Access*, vol. 9, pp. 78799-78811, 2021, doi: 10.1109/ACCESS.2021.3083705.

[75] H. Wang, Y. Luo, W. An, Q. Sun, J. Xu and L. Zhang, "PID Controller-Based Stochastic Optimization Acceleration for Deep Neural Networks," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5079-5091, 2020, doi: 10.1109/TNNLS.2019.2963066.

[76] X. Chen, F. Li, H. He, and M. Wu, "Optimization of PID parameter tuning for gravity stabilized platform based on improved differential evolutionary algorithm," *Journal of Physics: Conference Series 2029*, vol. 2019, 2021. doi: 10.1088/1742-6596/2029/1/012107.

[77] H. Liu, L. Gao, X. Kong and S. Zheng, "An improved artificial bee colony algorithm," *2013 25th Chinese Control and Decision Conference (CCDC)*, pp. 401-404, 2013, doi: 10.1109/CCDC.2013.6560956.

[78] E. R. Fernandez Cornejo, R. C. Diaz and W. I. Alama, "PID Tuning based on Classical and Meta-heuristic Algorithms: A Performance Comparison," *2020 IEEE Engineering International Research Conference (EIRCON)*, pp. 1-4, 2020, doi: 10.1109/EIRCON51178.2020.9253750.

[79] A. A. Kesarkar, N. Selvaganesan, "Tuning of optimal fractional-order PID controller using an artificial bee colony algorithm," *Journal of Systems Science and Control Engineering*, vol. 3, no. 1, pp. 99-105, 2015, doi: 10.1080/21642583.2014.987480.

[80] W. Liao, Y. Hu and H. Wang, "Optimization of PID control for DC motor based on artificial bee colony algorithm," *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pp. 23-27, 2014, doi: 10.1109/ICAMechS.2014.6911617.

[81] N. Elkhateeb and R. Badr, "A novel variable population size artificial bee colony algorithm with convergence analysis for optimal parameter tuning," *International Journal of Computational Intelligence and Applications*, vol. 16, no. 3, pp. 1750018, Sep. 2017, doi: 10.1142/S1469026817500183.

[82] N. A. Elkhateeb, and R. I. Badr, "Novel PID tracking controller for 2DOF robotic manipulator system based on artificial bee colony algorithm," *Electrical, Control and Communication Engineering*, vol. 13, no. 1, pp. 55–62, 2017, doi: 10.1515/ecce-2017-0008.

[83] Ghassan A. Sultan, Muhammed K. Jarjes, "Optimal PID controller design using artificial bee colony algorithm for robot arm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 84-91, 2021, doi: 10.11591/ijeecs.v21.i1.pp84-91.

[84] K. Rajasekhar, K Raja Naguru Babu, "Firefly Optimization algorithm based PID controller tuning in paper machine," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 9, no. 12, 2021.

[85] Eka Suci Rahayu, Alfian Ma'arif, Abdullah Cakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," *International Journal of Robotics and Control Systems (IJRCS)*, vol. 2, no. 2, pp. 435–447, 2022, doi: 10.31763/ijrcs.v2i2.476.

[86] M. Kishnani, S. Pareek and R. Gupta, "Optimal tuning of DC motor via simulated annealing," *2014 International Conference on Advances in Engineering & Technology Research*, pp. 1-5, 2014, doi: 10.1109/ICAETR.2014.7012928.

[87] J. Liu, W. Pan, R. Qu, and M. Xu, "Research on the Application of PID Control with Neural Network and Parameter Adjustment Method of PID Controller," In *Association for Computing Machinery*, 2018, doi: 10.1145/3297156.3297167.

[88] M. Li and X. Feng, "Research on PID parameter tuning based on improved artificial bee colony algorithm," *Journal of Physics: Conference Series, IOP Publishing*, vol. 1670, 2020, doi: 10.1088/1742-6596/1670/1/012017.

[89] Z. Bingula, and O. Karahan, "A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system," *The Franklin Institute*, vol. 355, no. 13, pp. 5534–5559, 2018, doi: 10.1016/j.jfranklin.2018.05.056.

[90] A. Mamadapur and G. Unde Mahadev, "Speed Control of BLDC Motor Using Neural Network Controller and PID Controller," *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC)*, pp. 146-151, 2019, doi: 10.1109/ICPEDC47771.2019.9036695.

[91] D. Vivek, Dr. P. B. Kumar, "PID controller design with cuckoo search algorithm for stable and unstable SOPDT processes," *IOP Conference Series: Materials Science and Engineering*, 2021. doi: 10.1088/1757-899X/1091/1/012059.

[92] X. L. Zhang and Q. Zhang, "Optimization of PID Parameters Based on Ant Colony Algorithm," in *2021 International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS)*, pp. 850-853, 2021, doi: 10.1109/ICITBS53129.2021.00211.

[93] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *SIMULATION*, vol. 76, no. 2, pp. 60-68, 2001, doi:10.1177/003754970107600201.

[94] S. Pramanik, A. Sengupta and N. Roy, "PID Flow-Level Control Tuned by Genetic Algorithm and Harmony Search Algorithm," *2021 IEEE Second International Conference on Control, Measurement and Instrumentation (CMI)*, pp. 172-177, 2021, doi: 10.1109/CMI50323.2021.9362959.

[95] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony Search Method: Theory and Applications", *Computational Intelligence and Neuroscience*, vol. 2015, pp. 1–10, 2015, doi: 10.1155/2015/258491.

[96] I. A. Abdul Jamil and M. Moghavvemi, "Optimization of PID Controller Tuning method using Evolutionary Algorithms," *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, pp. 1-7, 2021, doi: 10.1109/i-PACT52855.2021.9696875.

[97] A. E. H. Saad, Z. Dong, and M. Karimi, "A Comparative Study on Recently-Introduced Nature-Based Global Optimization Methods in Complex Mechanical System Design," *Algorithms*, vol. 10, no. 4, 2017, doi:10.3390/a10040120.

[98] A. E. Kayabekir, G. Bekdaş, S. M. Nigdeli, and Z. W. Geem, "Optimum design of PID controlled active tuned mass damper via modified harmony search," *Applied Sciences*, vol. 10, no. 8, p. 2976, 2020, doi: 10.3390/app10082976.

[99] M. Gheisarnejad, "An effective hybrid harmony search and cuckoo optimization algorithm based fuzzy PID controller for load frequency control," *Applied Soft Computing*, vol. 65, pp. 121-138, 2018, doi: 10.1016/j.asoc.2018.01.007.

[100] Vijendra Kumar, S. M. Yadav, "A state-of-the-art review of heuristic and metaheuristic optimization techniques for the management of water resources," *Water Supply*, vol. 22, no. 4, pp. 3702–3728, 2022, doi: 10.2166/ws.2022.010.

[101] C. Dumitrescu, P. Ciotirnae, C. Vizitiu, "Fuzzy Logic for Intelligent Control System Using Soft Computing Applications," *Sensors*, vol. 21, no. 8, p. 2617, 2021, doi: 10.3390/s21082617.

[102] R. G. Rakshasmare, G. A. Kamble, R. H. Chile. "Some Tuning Methods of PID Controller For Different Processes," in *International Conference on Information Engineering, Management and Security*, pp. 282-288, 2015.

[103] Z. Wang, C. Qin, B. Wan, and W. W. Song "A Comparative Study of Common Nature-Inspired Algorithms for Continuous Function Optimization," *Entropy*, vol. 23, no. 7, p. 874, 2021, doi: 10.3390/e23070874.

[104] K. Jagatheesan, B. Anand, S. Samanta, N. Dey, A. S. Ashour and V. E. Balas, "Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm," in *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 503-515, 2019, doi: 10.1109/JAS.2017.7510436.

[105] S. Ladjouzi, and S. Grouni, “PID controller parameters adjustment using a single memory neuron,” *Journal of the Franklin Institute*, vol. 357, no. 9, pp. 5143-5172, 2020, doi: 10.1016/j.jfranklin.2020.02.027.

[106] A. Gupta, and P. K. Padhy, “Modified Firefly Algorithm based controller design for integrating and unstable delay processes,” *Engineering Science and Technology, an International Journal*, vol. 19, no. 1, pp. 548-558, 2016, doi: 10.1016/j.jestch.2015.09.015.

[107] M. R. K. Shagor, A. J. Mahmud, M. M. Nishat, F. Faisal, M. H. Mithun and M. A. Khan, “Firefly Algorithm Based Optimized PID Controller for Stability Analysis of DC-DC SEPIC Converter,” *2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 0957-0963, 2021, doi: 10.1109/UEMCON53757.2021.9666555.

[108] R. Bansal, M. Jain and B. Bhushan, ”Designing of Multi-objective Simulated Annealing Algorithm tuned PID controller for a temperature control system,” *2014 6th IEEE Power India International Conference (PIICON)*, pp. 1-6, 2014, doi: 10.1109/POWERI.2014.7117716.

[109] L. Gou, W. Shao, X. Zeng, Y. Shen and Z. Zhou, ”Rapid Simulated Annealing Algorithm for optimization of Aeroengine Control Based on BP Neural Network,” in *2019 Chinese Control Conference (CCC)*, pp. 8848-8852, 2019, doi: 10.23919/ChiCC.2019.8866588.

[110] M. Shatnawi and E. Bayoumi, ”Brushless DC Motor Controller Optimization Using Simulated Annealing,” *2019 International Conference on Electrical Drives & Power Electronics (EDPE)*, pp. 292-297, 2019, doi: 10.1109/EDPE.2019.8883924.

[111] A. Surana and B. Bhushan, ”Design and Comparison of PSO, SA and GA tuned PID Controller for Ball Balancer Arrangement,” *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1-5, 2021, doi: 10.1109/ICECCT52121.2021.9616686.

[112] X. Lv, Z. Zhang, Y. Chen, J. Zhang, Q. Yue and W. Zhang, ”A Fractional Order Control Method of Electromechanical Actuator Based on PSO-SA Optimization,” *2022 34th Chinese Control and Decision Conference (CCDC)*, pp. 103-108, 2022, doi: 10.1109/CCDC55256.2022.10034160.

[113] S. K. Valluru, K. Sehgal and H. Thareja, ”Evaluation of Moth-Flame Optimization, Genetic and Simulated Annealing tuned PID controller for Steering Control of Autonomous Underwater Vehicle,” *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1-6, 2021, doi: 10.1109/IEMTRONICS52119.2021.9422632.

[114] K. S. M. Jagindar Singh, I. Elamvazuthi, K. Z. K. Shaari and F. V. Lima, ”PID tuning control strategy using Cuckoo search algorithm for pressure plant,” *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2016, pp. 1-6, 2016, doi: 10.1109/ICIAS.2016.7824127.

[115] K. S. M. J. Singh, I. Elamvazuthi, K. Z. K. Shaari and F. V. Lima, ”PID tuning control strategy using Cuckoo Search algorithm,” *2015 IEEE Student Conference on Research and Development (SCOReD)*, pp. 129-133, 2015, doi: 10.1109/SCORED.2015.7449309.

[116] T. Adel, S. Hichem and C. Abdelkader, ”Cuckoo Search Algorithm and TS Fuzzy Models for PID Control,” *2019 International Conference on Signal, Control and Communication (SCC)*, pp. 331-336, 2019, doi: 10.1109/SCC47175.2019.9116153.

[117] Holland J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Ann Arbor, MI, USA, 1975.

[118] K. K. Nimisha and R. Senthilkumar, ”A Survey On Optimal Tuning Of PID Controller For Buck-Boost converter Using Cuckoo-Search Algorithm,” *2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT)*, pp. 216-221, 2018. doi: 10.1109/ICCPCCT.2018.8574321.

[119] K. Zawirski, K. Nowopolski and P. Siwek, ”Application of Cuckoo Search Algorithm for Speed Control Optimization in Two-Sided Electrical Drive,” *2018 IEEE 18th International Power Electronics and Motion Control Conference (PEMC)*, pp. 651-656, 2018, doi: 10.1109/EPEPEMC.2018.8522006.

[120] M. Kumar and K. Chaursiya, ”Position control of brushless DC motor using harmony search algorithm optimization technique,” *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, pp. 754-757, 2017, doi: 10.1109/ICECA.2017.8203644.

[121] B. Hekimoğlu, “Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm,” in *IEEE Access*, vol. 7, pp. 38100-38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[122] L. Caiza, D. S. Benítez and O. Camacho, “Non-linear PID Controller Optimization using the Artificial Bee Colony Algorithm Applied to a Small-Scale Pasteurization Plant,” *2022 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pp. 1-6, 2022, doi: 10.1109/ROPEC55836.2022.10018648.

[123] A. Kumar and V. Kumar, ”Artificial bee colony based design of the interval type-2 fuzzy PID controller for robot manipulator,” *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 602-607, 2017, doi: 10.1109/TENCON.2017.8227933.

[124] A. K. Mishra, V. K. Tiwari, R. Kumar and T. Verma, ”Speed control of dc motor using artificial bee colony optimization technique,” *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, pp. 1-6, 2013, doi: 10.1109/CARE.2013.6733772.

[125] M. Li and X. Feng, ”Application of Improved Artificial Bee Colony Algorithm in constant pressure water supply system,” *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pp. 521-525, 2020, doi: 10.1109/CACRE50138.2020.9230243.

[126] F. Majid, M. Mostafa, A. Hassan and E. K. Abdeljalil, ”Differential Evolution Approach for Identification and Control of Stable and Unstable Systems,” *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 218-223, 2022, doi: 10.1109/CoDIT55151.2022.9804077.

[127] P. Sanchez-Sanchez, J. G. Cebada-Reyes, A. Ruiz-Garcia, A. Montiel-Martínez and F. Reyes-Cortés, ”Differential Evolution Algorithms Comparison Used to Tune a Visual Control Law,” in *IEEE Access*, vol. 10, pp. 46028-46042, 2022, doi: 10.1109/ACCESS.2022.3168965.

[128] J. Zhang, P. Wu, X. Wang, X. Yu and S. Duan, ”PID Parameter Tuning of Combined Heat and Power Generation Unit Based on Differential Evolution Algorithm,” *2022 7th International Conference on Power and Renewable Energy (ICPRE)*, pp. 1186-1190, 2022, doi: 10.1109/ICPRE55555.2022.9960306.

[129] S. Prainetr, T. Phurahong, K. Janprom and N. Prainetr, ”Design Tuning PID Controller For Temperature Control Using Ant Colony Optimization,” *2019 IEEE 2nd International Conference on Power and Energy Applications (ICPEA)*, pp. 124-127, 2019, doi: 10.1109/ICPEA.2019.8818517.

[130] M. J. Blondin and P. Sicard, ”Statistical convergence analysis of ACO — NM for PID controller tuning,” *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 487-492, 2015, doi: 10.1109/ICIT.2015.7125146.

[131] M. Sreejeth, R. Kumar, N. Tripathi and P. Garg, ”Tuning A PID Controller using Metaheuristic Algorithms,” *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, pp. 276-282, 2023, doi: 10.1109/ICCES57224.2023.10192687.

[132] D. Sandoval, I. Soto and P. Adasme, ”Control of direct current motor using Ant Colony optimization,” *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pp. 79-82, 2015, doi: 10.1109/Chilecon.2015.7400356.

[133] B. A. Kouassi, Y. Zhang, S. Ouattara and M. J. Mbyamm Kiki, ”PID Tuning Of Chopper Fed Speed Control Of DC Motor Based On Ant Colony Optimization Algorithm,” *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, pp. 407-412, 2019, doi: 10.1109/CIEEC47146.2019.CIEEC-2019179.

[134] R. Singh, A. Kumar and R. Sharma, ”Fractional Order PID Control using Ant Colony Optimization,” *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1-6, 2016, doi: 10.1109/ICPEICES.2016.7853387.

[135] Y. Pei, W. Wang and S. Zhang, ”Basic Ant Colony Optimization,” *2012 International Conference on Computer Science and Electronics Engineering*, pp. 665-667, 2012, doi: 10.1109/ICCSEE.2012.178.

[136] C. S. Rajan and M. Ebenezer, ”Grey Wolf Optimizer Algorithm for Performance Improvement and Cost Optimization in Microgrids,” *2022 6th International Conference on Green Energy and Applications (ICGEA)*, pp. 115-121, 2022, doi: 10.1109/ICGEA54406.2022.9791902.

[137] A. H. Khan, Z. Shao, S. Li, Q. Wang, and N. Guan, "Which is the Best PID Variant for Pneumatic Soft Robots? An Experimental Study," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 451-460, 2020, doi: 10.1109/JAS.2020.1003045.
[138] W. Long, J. Jiao, X. Liang, S. Cai and M. Xu, "A Random Opposition-Based Learning Grey Wolf Optimizer," in *IEEE Access*, vol. 7, pp. 113810-113825, 2019. doi: 10.1109/ACCESS.2019.2934994.
[139] P. Ouyang, and V. Pano, "Comparative Study of DE, PSO and GA for Position Domain PID Controller Tuning," *Algorithms*, vol. 8, no. 3, pp. 697-711, 2015, doi: 10.3390/a8030697.
[140] J. A. Abdulsaheb, and D. J. Kadhim, "Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey," *Robotics*, vol. 12, no. 4, p. 93, 2023, doi: 10.3390/robotics12040093.
[141] P. Warrier, and P. Shah, "Optimal Fractional PID Controller for Buck Converter Using Cohort Intelligent Algorithm," *Applied System Innovation*, vol. 4, no. 3, 2021, doi: 10.3390/asi4030050.
[142] H. Khan, S. Khatoon, P. Gaur, M. Abbas, C. A. Saleel, and S. A. Khan, "Speed Control of Wheeled Mobile Robot by Nature-Inspired Social Spider Algorithm-Based PID Controller," *Processes*, vol. 11, no. 4, pp. 1202, 2023, doi: 10.3390/pr11041202.
[143] A. Ahmed, G. Parmar and R. Gupta, "Application of GWO in Design of Fractional Order PID Controller for Control of DC Motor and Robustness Analysis," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 646-651, 2018, doi: 10.1109/ICACCCN.2018.8748548.
[144] V. Yadav, G. Parmar and R. Bhatt, "Robustness Analysis with Perturbation for Control System with GWO," *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 812-815, 2019, doi: 10.1109/ISCON47742.2019.9036254.
[145] Z. Liang, L. Fu, X. Li, Z. Feng, J. W. Sleigh and H. K. Lam, "Ant Colony Optimization PID Control of Hypnosis With Propofol Using Renyi Permutation Entropy as Controlled Variable," in *IEEE Access*, vol. 7, pp. 97689-97703, 2019, doi: 10.1109/ACCESS.2019.2927321.
[146] S. Gupta et al., "Metaheuristic Optimization Techniques Used in Controlling of an Active Magnetic Bearing System for High-Speed Machining Application," in IEEE Access, vol. 11, pp. 12100-12118, 2023, doi: 10.1109/ACCESS.2023.3241854.
[147] H. Sarma and A. Bardalai, "Tuning of PID Controller using Driving Training-Based Optimization for Speed Control of DC Motor," *2023 4th International Conference on Computing and Communication Systems (I3CS)*, pp. 1-8, 2023, doi: 10.1109/I3CS58314.2023.10127458.
[148] A. K. Bhullar, R. Kaur and S. Sondhi, "Design And Comparative Analysis Of Optimized Fopid Controller Using Neural Network Algorithm," *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pp. 91-96, 2020, doi: 10.1109/ICIIS51140.2020.9342743.
[149] V. P. Meena, U. K. Yadav, A. Gupta and V. P. Singh, "Reduced-Order Model Based Design of PID Control for Zeta Converter Using GWO Algorithm," *2022 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1-5, 2022, doi: 10.1109/PEDES56012.2022.10080587.
[150] S. Yousaf, A. Mughees, M. G. Khan, A. A. Amin and M. Adnan, "A Comparative Analysis of Various Controller Techniques for Optimal Control of Smart Nano-Grid Using GA and PSO Algorithms," In *IEEE Access*, vol. 8, pp. 205696-205711, 2020, doi: 10.1109/ACCESS.2020.3038021.
[151] J. Tang, G. Liu and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," in *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627-1643, 2021, doi: 10.1109/JAS.2021.1004129.
[152] M. S. Ayas and E. Sahin, "Parameter effect analysis of particle swarm optimization algorithm in PID controller design," *An International Journal of Optimization and Control: Theories and Applications*, vol. 9 no. 2, 2019, doi: 10.11121/ijocta.01.2019.00659.
[153] S. Harshitha, S. Shamanth, and A. K. Chari, "A Review of Various Controller Techniques Designed for the Operational Control of DC and Servo Motors," *Journal of Physics*, vol. 2273, 2022, doi: 10.1088/1742-6596/2273/1/012001.
[154] P. Deulkar and S. Hanwate, "Analysis of PSO-PID controller for CSTR temperature control," *2020 IEEE First International Conference on Smart Technologies for Power, Energy, and Control (STPEC)*, pp. 1-6, 2020, doi: 10.1109/STPEC49749.2020.9297750.
[155] S. Xianghan, N. Liu, R. Shen, K. Wang, Z. Zhao, and X. Sheng, "Nonlinear PID Controller Parameters Optimization Using Improved Particle Swarm Optimization Algorithm for the CNC System," *Applied Sciences*, vol. 12, no. 20, pp. 10269, 2022, doi: 10.3390/app122010269.
[156] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Arch Computat Methods Eng,* vol. 29, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.
[157] M. Z. Efendi, F. D. Murdianto and H. N. Baweani, "Robustness Analysis of PID-Cuckoo Search Algorithm to Voltage Control in Three Phase of Synchronous Generator with Dynamic Load Condition," *2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, pp. 133-138, 2018, doi: 10.1109/ELECSYM.2018.8615486.
[158] R. S. Barbosa, I. S. Jesus, "Special Issue on Algorithms for PID Controllers 2021," *Algorithms*, vol. 16, no. 1, 2023, doi: 10.3390/a16010035.
[159] A. Zulu, "Towards explicit PID control tuning using machine learning," *2017 IEEE AFRICON*, pp. 430-433, 2017, doi: 10.1109/AFRCON.2017.8095520.
[160] J. H. Lanker, R. Bhushan and N. Gupta, "Load Frequency Control of Multi Area Power System Using Meta-heuristic/Artificial Intelligence Techniques," *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, pp. 1-6, 2022, doi: 10.1109/ICICCSP53532.2022.9862473.
[161] T. Wang, H. Wang, H. Hu, X. Lu, and S. Zhao, "An adaptive fuzzy PID controller for speed control of brushless direct current motor," *SN Applied Sciences*, vol. 4, no. 71, 2022, doi: 10.1007/s42452-022-04957-6.
[162] Dr. T. R. Rangaswamy, Dr. S. P. Vijayaragavan, "Efficient Drum Level Control for Steam," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 12, pp. 429-434, 2018.
[163] M. Shehab, M. A. A. Hashem, M. K. Y. Shambour, "A Comprehensive Review of Bat Inspired Algorithm: Variants, Applications, and Hybridization," *Archives of Computational Methods in Engineering*, vol. 30, pp. 765–797, 2023, doi: 10.1007/s11831-022-09817-5.