



RESEARCH ARTICLE - ENGINEERING (MISCELLANEOUS)

Design of Robust Fractional Order PID Controllers for Four Tank Systems Using Dragonfly Algorithm

Zainab Naser Kadhim¹, Hamed Agahi^{1*}

¹Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Islamic Republic of Iran

* Corresponding author E-mail: hamed.agahi@iau.ac.ir

Article Info.	Abstract
<i>Article history:</i> Received 15 June 2025 Accepted 07 September 2025 Publishing 30 September 2025	Fractional Order Proportional Integral Derivative (FOPID) controllers are commonly utilized in reactors, power systems, robotic systems, and various industrial processes. Properly setting the parameters of an FOPID controller is crucial, as well-chosen parameters can significantly enhance performance in dynamic systems. This article introduces a meta-heuristic approach using the dragonfly algorithm, combined with a proposed objective function based on the Root Mean Square Error (RMSE), to optimize the parameters of the FOPID controller for a four-tank system (Quadruple Tank Process, QTP). The method is implemented in MATLAB and compared with traditional techniques. Simulation results demonstrate the effectiveness of the proposed approach, as evidenced by improved performance metrics such as the Integral of Square Error (ISE) and the Integral of Absolute Error (IAE).
This is an open-access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/)	
Publisher: Middle Technical University	
Keywords: Fractional Order Proportional Integral Derivative (FOPID); Quadruple Tank Process (QTP); Four Tank System; Dragonfly Meta-Heuristic Algorithm.	

1. Introduction

Today, Proportional Integral Derivative (PID) controllers are widely used in industrial processes. They are widely used due to their simple structure and reliable performance. Researchers have dedicated considerable attention to the creation and tuning of PID controllers since 1942 [1]. A significant body of research has accumulated since then, encompassing the design, stability analysis, performance evaluation, and diverse applications of PID controllers [2, 3]. The Fractional Order PID (FOPID) controller emerges as a subsequent evolution of the standard PID controller, distinguished by the inclusion of two supplementary parameters representing fractional order integration and differentiation. The FOPID controller finds broad applicability in a range of engineering domains, including intelligent reactors [4], electronic power converters [5], rehabilitation apparatus [6], automatic voltage regulators [7], industrial process simulations [8], robotic mechanisms [9], power grids utilizing synchronous generators [10], and wind energy turbines [11]. It is critical to understand that the specific numerical values of FOPID controller parameters will vary, depending on the unique requirements of each application, yielding an individualized optimal operational profile. There are various methods such as trial-and-error method, curve method, Ziegler-Nichols method, and methods based on meta-heuristic algorithms to adjust FOPID controller coefficients. Recently, methods based on meta-heuristic algorithms have been widely noticed by researchers [12-15]. In [16], The genetic algorithm is presented to tune FOPID parameters. Within this approach, the ISE index is utilized for the goal of the function, and the sequence of bands are in the range of 0 to 100. The outcomes indicate that the proposed FOPID control system remains controllable even under variations in fractional parameters. even when the fractional factor values go beyond the standard limits. In [17], A FOPID controller optimized through a multi-objective genetic algorithm. A magnetically damp semi-active seat suspension system is the subject of this proposal. The core focus of this method is the gain crossover frequency and the phase margin. Findings show the continuous FOPID (cFOPID) provides superior results compared to conventional integer controllers. The evolutionary multi-objective algorithm, dominated sorting genetic algorithm (NSGA-II) is employed in [18] as a tuning strategy for cFOPID within hydraulic turbines; its operation relies on two objective functions: the Integral Squared Error (ISE) and the Integral of the Time-weighted Squared Error (ITSE). The findings confirm that the NSGA-II effectively optimizes cFOPID. In [19], a cloud model-based quantum genetic algorithm (CQGA) is proposed to fine-tune cFOPID parameters for controlling the motion of an autonomous underwater vehicle (AUV). This method combines cloud model theory with quantum genetic algorithms, leveraging principles of quantum computing, with the integral weighted absolute error as the objective function. Results show that cFOPID enhances control over both heading and diving. In [20], The genetic algorithm designed for tuning the cFOPID controller is implemented in a conical tank system, with the Integral of Time Absolute Error (ITAE) serving as the objective function. Reference [21] proposes a genetic algorithm specifically designed for tuning the cFOPID parameters within a boiler turbine system. Utilizing floating-point coding, a selection process predicated on ranking, and a strategy to retain elite solutions, alongside a grouping mechanism, this algorithm is built to boost search effectiveness and prevent early convergence to suboptimal results. Research found in reference [22] investigates a frenzied-based particle swarm optimization algorithm.

Nomenclature & Symbols

FOPID	Fractional Order Proportional Integral Derivative	PSO	Particle Swarm Optimization
QTP	Quadruple Tank Process	ISE	Integral Squared Error
PID	Proportional Integral Derivative	ITSE	Integral of the Time-weighted Squared Error
ITAE	Integral of Time Absolute Error	GA	Genetic Algorithms
ASO	Atom Search Optimization	ICA	Imperialist Competitive Algorithm
HPS	Hybrid Power System	RMSE	Root Mean Square Error
WOA	Whale Optimization Algorithm	DA	Differential Algebra
NSGA- II	Non-Dominated Sorting Genetic Algorithm	COA	Cuckoo Optimization Algorithm
CQGA	Cloud Model-Based Quantum Genetic Algorithm	IAE	Integral Absolute Error
cFOPID	Continuous Fractional Order Proportional Integral Derivative	DA	Dragonfly Algorithm

This study's goal is to identify ideal PID controller parameters suitable for the task of frequency regulation within a multi-source microgrid, featuring the integration of renewable energy. Reference [23] details the implementation of the Atom Search Optimization (ASO) algorithm. The ASO is applied to fine-tune the FOPID controller parameters for managing frequency and load within a linked Hybrid Power System (HPS). The HPS incorporates renewable sources, including wind and solar, alongside plug-in electric vehicles. The presented results clearly illustrate that the FOPID controller, when optimized through the ASO approach, performs favorably when compared to standard controllers such as I, PI, PID, FOI, and FOPI. Finally, reference [24]. The utilization of the Whale Optimization Algorithm (WOA) is explored concerning the ideal configuration of a FOPID controller within a multi-area power grid featuring multiple energy sources. This WOA-integrated FOPID controller shows a desirable dynamic response. Specifically, it excels in aspects of settling time and peak overshoot, while also demonstrating stability amidst fluctuations in system characteristics.

Given the importance of automatic liquid level control in industrial applications, an effective and efficient control method is crucial. The dynamics of a four-tank system closely mirror those of real-world processes, such as boiler operations, distillation columns, and oil refineries in the petrochemical industry. These processes involve intricate interactions within the four-tank system, making control a complex challenge. This complexity drives the need for a robust control method to adjust the parameters of the FOPID controller in the four-tank system.

This study introduces the dragonfly optimization method for fine-tuning the parameters of the FOPID controller in a four-tank system. Dragonfly's performance in reaching optimal solutions swiftly makes it a strong contender, outshining methods like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and the Imperialist Competitive Algorithm (ICA). Furthermore, the adoption of Root Mean Square Error (RMSE) in the cost assessment introduces an innovative approach.

Here's how the subsequent sections are organized in this investigation: Section II serves as an introduction to the four-tank system. Following that, Section III delves into the FOPID controller, along with an explanation of the introduced approach. We then move onto Section IV, which shows the outcomes of the simulation runs. Section V analyzes these simulation results thoroughly, and finally, Section VI brings the study to a close.

2. Four-Tank System

The four-tank system is a nonlinear parameter-varying process utilized in chemical and oil-and-gas processes. The system contains two water tanks and two pumps that are connected; the tanks are filled by two pumps (see Fig. 1) [25]. The power supplied to the pumps serves as the input signal, whereas the water level present in the lower tanks constitutes the output of the system. Each tank's mathematical representation is formulated utilizing Bernoulli's principle and the principle of mass conservation. The goal is to manage the water levels in the two lower tanks, utilizing the control action afforded by the two pumps. Each pump's output is divided into two streams through three-way valves. In this setup, into each tank water is pumped in the top of this tank and out through the bottom of the tank.

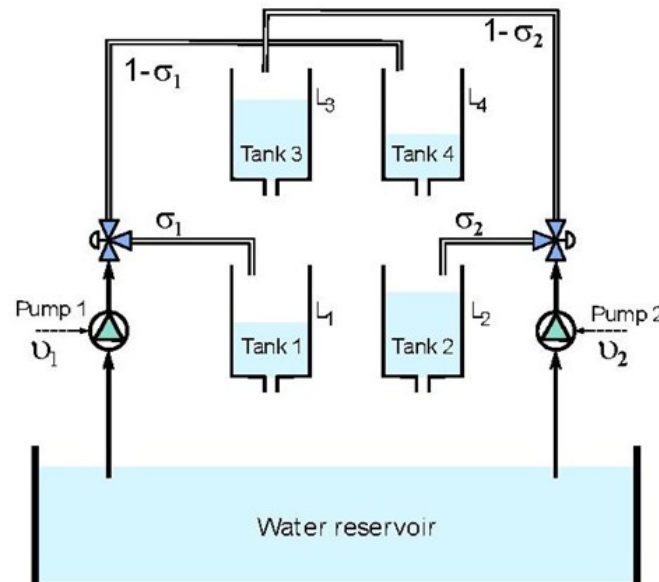


Fig. 1. Four tank system[25]

The mathematical equation of the four-tank system according to Bernoulli's law and conservation of mass is as follows:

$$A_1 \frac{dh_1(t)}{dt} = q_{in_1} + q_{out_3} - q_{out_1} = \gamma_1 k_1 V_1 + a_3 \sqrt{2gh_3(t)} - a_1 \sqrt{2gh_1(t)} \quad (1)$$

The non-linear relations of each tank, similar to Eq. (1) are obtained below.

$$\frac{dh_1(t)}{dt} = \frac{\gamma_1 k_1}{A_1} V_1 + \frac{a_3}{A_1} \sqrt{2gh_3(t)} - \frac{a_1}{A_1} \sqrt{2gh_1(t)} \quad (2)$$

$$\frac{dh_2(t)}{dt} = \frac{\gamma_2 k_2}{A_2} V_2 + \frac{a_2}{A_2} \sqrt{2gh_2(t)} - \frac{a_4}{A_2} \sqrt{2gh_4(t)} \quad (3)$$

$$\frac{dh_3(t)}{dt} = \frac{(1-\gamma_2)}{A_3} k_2 V_2 - \frac{a_3}{A_3} \sqrt{2gh_3(t)} \quad (4)$$

$$\frac{dh_4(t)}{dt} = \frac{(1-\gamma_1)}{A_4} k_1 V_1 - \frac{a_4}{A_4} \sqrt{2gh_4(t)} \quad (5)$$

The above relations are non-linear due to the existence of the root term, which makes the design of the controller challenging. Therefore, the operating region of the system is determined through a first-order Taylor series expansion of the nonlinear dynamics around the equilibrium point after the computation of Jacobian matrix using the Taylor series expansion after transformation of Jacobian matrix.

$$\frac{dx_1}{dt} = f_1(h_1, h_2, \dots, h_n, u_1, u_2, \dots, u_n) \quad (6)$$

$$\frac{dx_n}{dt} = f_n(h_1, h_2, \dots, h_n, u_1, u_2, \dots, u_n) \quad (7)$$

General vector form

$$H_e = h_e + \Delta h \quad (8)$$

$$U_e = u_e + \Delta u \quad (9)$$

Linear approximation with Taylor series

$$\dot{x} = \frac{dx}{dt} f(H_e, U_e) = f(h_e + \Delta h, u_e + \Delta u) \quad (10)$$

$$f(x, u) = f(h_e, u_e) + \frac{df}{dh}(h_e, u_e) + \frac{df}{du}(h_e, u_e) \quad (11)$$

The form of the system space is as follows

$$\dot{x}_1 = -\frac{a_1}{A_1} \sqrt{\frac{g}{2h_{10}}} x_1 + \frac{a_3}{A_1} \sqrt{\frac{g}{2h_{30}}} x_3 + \frac{\gamma_1 k_1}{A_1} u_1 \quad (12)$$

$$\dot{x}_2 = -\frac{a_2}{A_2} \sqrt{\frac{g}{2h_{20}}} x_2 + \frac{a_4}{A_2} \sqrt{\frac{g}{2h_{40}}} x_4 + \frac{\gamma_2 k_2}{A_2} u_2 \quad (13)$$

$$\dot{x}_3 = -\frac{a_3}{A_3} \sqrt{\frac{g}{2h_{30}}} x_3 + \frac{(1-\gamma_2)}{A_3} k_2 u_2 \quad (14)$$

$$\dot{x}_4 = -\frac{a_4}{A_4} \sqrt{\frac{g}{2h_{40}}} x_4 + \frac{(1-\gamma_1)}{A_4} k_1 u_1 \quad (15)$$

Time constants:

$$T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_{i0}}{g}} \quad (16)$$

The transformation function after linearization is according to Eq. (17).

$$G(s) = \begin{bmatrix} \frac{\gamma_1 c_1}{1+T_1} & \frac{(1-\gamma_2)c_1}{(1+T_3s)(1+sT_1)} \\ \frac{(1-\gamma_1)c_2}{(1+sT_4)(1+sT_2)} & \frac{\gamma_2 c_2}{1+sT_2} \end{bmatrix} \quad (17)$$

In Eq. (17), the values of c_1 and c_2 are equal:

$$c_1 = \frac{T_1 k_1 k_c}{A_1} \quad (18)$$

$$c_2 = \frac{T_2 k_2 k_c}{A_2} \quad (19)$$

3. Principles of Operation of the FOPID Controller

In 1999 Podlubny proposed the FOPID controller [26]. An FOPID controller has five parameters. These are proportional gain, integral gain and derivative gain, as well as order of integration and differentiation [27]. The law describing the control is presented in (20).

$$u(t) = k_p e(t) + k_I D_t^{-\lambda} e(t) + k_D D_t^{\delta} \quad (20)$$

The FOPID transfer function is obtained by Laplace transform according to Eq. (21).

$$G_c(s) = k_p + k_I s^{-\lambda} + k_D s^{\delta} \quad (21)$$

The design of FOPID includes the determination of three parameters k_p , k_I , k_D and two orders λ , δ . Different design methods have been presented for this type of controller. In this article, it employs the Dragonfly meta-heuristic to obtain the optimal FOPID settings for a Fractional Order PID (FOPID) controller.

4. Proposed Method

This section introduces a proposed method that employs Differential Algebra (DA) to fine-tune the parameters of the FOPID controller, as shown in Fig. 2. The following discussion will elaborate on this approach.

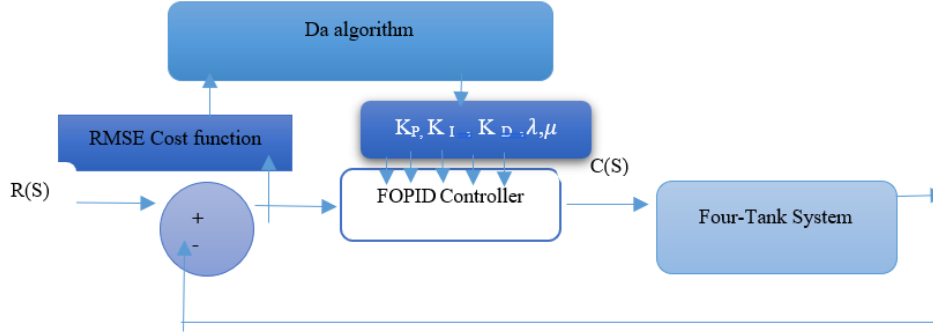


Fig. 2. Flowchart of FOPID controllers design for four-tank system

4.1. Decoupling

In this case, the two-input-two-output system will turn into two single-input-single-output systems, or in other words, the system will be decoupled. In fact, the fluid level within the initial tank is solely dictated by the output rate of the secondary pump, while the second tank's level is determined only by the first pump's output. This characteristic allows for straightforward, separate design of two controllers. Thus, the overall multivariable control strategy simplifies into a collection of isolated control loops. However, a key consideration is that this decoupling approach remains valid only if any direct flow of water into the lower tanks is minimal, effectively rendering it negligible as an external disturbance.

Assumptions and Robustness of the Decoupling Strategy: Let the linearized two-input two-output model around the operating point be written in the standard partitioned form

$$B = \begin{bmatrix} 0 & B_1 \\ B_2 & 0 \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \dot{x} = Ax + Bu, y = cx \quad (22)$$

Where $y^T = [h_1, h_2]$ are the lower-tank levels and $u^T = [V_1, V_2]$ are the pump voltages. The physical cross-coupling arises from (i) the three-way valve splitting ratios (γ_1, γ_2) that route fractions of the pump flows to the "opposite" branch, and (ii) hydraulic interactions that appear as off-diagonal blocks A_{12}, A_{21} in the Jacobian.

It quantifies the "minimal direct-flow" condition by requiring that the off-diagonal influence is small relative to self-dynamics, namely.

$$\varepsilon \geq \frac{\|A_{12}\|}{\|A_{11}\|} \quad \max(\gamma_1, \gamma_2) \geq \frac{\|A_{21}\|}{\|A_{22}\|} \quad (23)$$

With design guidelines $\varepsilon \geq 0.10$ and $\max(\gamma_1, \gamma_2) \geq 0.10$. Under these bounds the closed-loop matrix with two independent SISO controllers is strictly diagonally dominant, which ensures decentralized stability by Gershgorin's theorem; practically, the residual coupling acts as a bounded disturbance that the FOPID loops can reject.

Robustness check under finite coupling. To assess how sensitive the loops are to cross-terms, the above bounds can be relaxed in analysis to $\varepsilon [0.20, 0.05]$ and $(\gamma_1, \gamma_2) [0.20, 0]$. The decentralized design remains valid provided the closed-loop matrix $_{cl}A$ satisfies $|(A_{cl})_{ii}| > |(A_{cl})_{ij}|$ for $i = 1, 2$. This condition can be verified from the linearized model without time-domain simulations and gives a clear, quantitative envelope for which decoupling is justified.

Practical guideline for experiments. If a laboratory setup is available, (γ_1, γ_2) can be estimated by simple flow tests (fixed voltage, measure split), and the Jacobian blocks A_{12}, A_{21} can be obtained from small perturbations around the operating point. Reporting the measured (γ_1, γ_2) and the ratios $\|A_{12}\|/\|A_{11}\|, \|A_{21}\|/\|A_{22}\|$ assumes explicit and reproducible. If (γ_1, γ_2) are larger than the stated bounds, a static pre-compensator $N = (K_{12}k_{21})^{-1}$ can be introduced to partially cancel the steady-state coupling prior to FOPID loops.

4.2. Dragonfly algorithm

The dragonfly algorithm, A meta-heuristic technique was put forth by Mirjalili and collaborators during the year 2015. as documented in reference [28]. The core concept of this algorithm draws inspiration from the way dragonflies behave in the real world. The dragonflies' intelligent conduct is guided by these five key rules: dodging other nearby dragonflies to avoid clashes, modulating their flight velocity in reaction to where their neighbors are, seeking the center of gravity of their surrounding companions, heading toward potential food sources, and, finally, avoiding any detected threats. To simulate these behaviors, the dragonfly algorithm employs five distinct mathematical functions.

Separation function: It happens when dragonflies follow it to avoid collision with their neighbors. The mathematical relationship of this function is according to Eq. (24).

$$S_i = -\sum_{j=1}^N X - X_j \quad (24)$$

Speed function: This function calculates the speed of dragonflies according to the neighboring dragonfly as Eq. (25).

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (25)$$

Cohesion function: This function calculates the cohesion of neighbors according to Eq. (26).

$$C_i = \frac{\sum_{j=1}^N X_i}{N} - X \quad (26)$$

Attraction function: It shows the tendency of dragonflies towards the food source, which is calculated according to Eq. (27).

$$F_i = X^+ - X \quad (27)$$

Distraction function: the natural behavior that every dragonfly does to survive against the enemy's influence. This function can be seen according to Eq. (28).

$$E_i = X^- + X \quad (28)$$

Initially, the dragonfly algorithm initializes the position and step vectors randomly, Limited by the minimum and maximum values. defined for the problem's variables. Subsequently, during each cycle of the iterative process, both the optimal position and step of the dragonflies undergo sequential refinement. To determine the updated position vector for a dragonfly, the step vector, designated as ΔX , is incorporated alongside the current positional vector. This step vector effectively dictates the direction in which the dragonfly will move, and it is computed based on the mathematical relationship provided in Eq. (29).

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (29)$$

In this regard, a, s, c, f, e, w are weight vectors. Also, the position of the dragonflies is updated using Eq. (30).

$$X_{t+1} - X_t = \Delta X_{t+1} \quad (30)$$

In this regard, the parameter t is the number of repetitions.

4.3. Objective functions

Objective functions are typically classified into three categories: classical time-domain objective functions (such as maximum overshoot and time-varying functions), frequency-domain objective functions (like phase limits), and time-domain error it considers time-domain error objectives: ISE, IAE, and RMSE. This research focuses on error objective functions due to their general applicability and widespread acceptance. The RMSE is used according to Eq. (31).

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n e^2(k)} \quad (31)$$

$$e(k) = r_i(k) - h_i(k) \quad (32)$$

In Eq. (32), $e(k)$ is the difference between the input signal $r_i(k)$ and the output signal $h_i(k)$ in the closed loop system. Since each FOPID controller has five parameters, ten parameters must be tuned in total. Therefore, the optimization algorithm searches the controlling parameters in a 10-dimensional space. The vector of FOPID controlling parameters for the i^{th} member of the population is considered according to Eq. (33).

$$\vec{X}_i = [K_{P1}, K_{i1}, K_{d1}, K_{P2}, K_{i2}, K_{d2}, \lambda_1, \mu_1, \lambda_2, \mu_2] \quad (33)$$

where parameters $K_{P1}, K_{i1}, K_{d1}, K_{P2}, K_{i2}, K_{d2}, \lambda_1, \mu_1, \lambda_2, \mu_2$ The elements that make up the initial controller are presented here, followed by those of the subsequent controller.

Objective Function Clarification: it uses the Root Mean Squared Error (RMSE) as the primary scalar objective for tuning because it directly penalizes sustained tracking errors and is scale-consistent across outputs. We do not claim novelty for RMSE itself; our contribution lies in the overall tuning framework and comparative evaluation. For each output $k \in \{1, 2\}$ the discrete-time RMSE is

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n e^2(k)} \quad (34)$$

And the aggregate objective used in optimization is

$$\text{total RMSE} = \sqrt{\frac{RMSE_1^2 + RMSE_2^2}{2}} \quad (35)$$

To facilitate fair benchmarking, we also report ISE and IAE as secondary criteria in the results section

4.4. Sensitivity analysis

To verify robustness against implementation inaccuracies, we performed a simple sensitivity check on the optimized FOPID parameters for the two loops (Loop-1, Loop-2). Each parameter (KP, KI, KD, λ , μ) was perturbed individually by $\pm 10\%$ while the others were kept fixed, it drew a small set of random joint perturbations within $\pm 10\%$ for all parameters to mimic simultaneous errors. The results are summarized in Figs. 3 to 5 and indicate that performance degrades modestly while closed-loop stability is preserved within the tested ranges.

4.5. Practical implementation considerations

Translating the proposed design to a physical quadruple-tank setup requires accounting for sensing, actuation, and computation realities.

- Sensing and filtering. Level measurements are affected by quantization and noise; a first-order low-pass filter (cutoff in the 10–20 Hz range for typical level sensors) and proper sensor-gain calibration improve signal quality. The same filter should be included in validation simulations to avoid optimistic performance estimates.
- Actuation limits and nonlinearities. Pumps exhibit saturation, dead-zones, and rate limits. The controller output must be bounded to the admissible voltage range, augmented with an anti-windup mechanism and a slew-rate limiter. If a dead-zone is present, a small bias or feed-forward can be applied to overcome it while keeping steady-state error negligible.
- Sampling and fractional realization. The sampling period T_s should be chosen relative to the dominant tank time constants (e.g., tens of milliseconds for lab rigs). The FOPID can be implemented via a discrete approximation (e.g., Oustaloup/CRONE) of moderate order to balance accuracy and computational load; the same approximation order should be used in both design and validation.
- Delays and computation. I/O and computation delays should be measured and, if non-negligible, compensated (e.g., by phase margin budgeting or simple Smith-type prediction in software).
- Safety and constraints. Hard bounds on water levels (min/max) must be enforced with software interlocks and emergency shut off. Initial transients during start-up and restart after disturbances should be handled by a bump less transfer to prevent overshoot.

Remark. These considerations can be replicated on standard real-time platforms (e.g., microcontroller/DAQ/PC) without modifying the nominal tuning; they primarily ensure that laboratory behavior matches the validated simulation model.

5. Implementation and Examination of the Results

The proposed method is put to the test through a comprehensive comparison, pitting it against well-established algorithms. Specifically, we examine the performance relative to the classic Genetic Algorithm (GA), as detailed in reference [29], the Cuckoo Optimization Algorithm (COA) from publication [30], Particle Swarm Optimization (PSO), found in reference [31], and finally, the Imperialist Competitive Algorithm (ICA) according to [32]. All experiments were carried out in MATLAB R2023a on a laptop with an Intel Core i7 (2.27 GHz) and 16 GB RAM.

5.1. Evaluation criteria

Evaluation criteria commonly used in control are employed to assess the effectiveness of the proposed approach. For this purpose, in the subject of setting the controller parameters, different algorithms for setting the parameters of FOPID in the four-tank system are measured with various criteria.

- Integral of square value of error (ISE)

The ISE is according to Eq. (36). If the ISE is low, it is more suitable.

$$ISE = \int e^2(t) dt \quad (36)$$

- Integral of absolute value of error (IAE)

The IAE is according to Eq. (37). The IAE is lower, it is more suitable.

$$IAE = \int |e(t)| dt \quad (37)$$

RMSE was selected as the primary tuning objective for its direct interpretability in time-domain tracking; we do not claim novelty for RMSE itself. For fair benchmarking with prior work, ISE and IAE are additionally reported in the results.

5.2. Experimental results

The parameters of the four-tank system are chosen to be constant in accordance with Table 1 in these experiments in order to It is crucial to steer clear of any parameter value choices that might skew the outcomes when analyzing the four-tank system. The initial population size strongly affects convergence; very large populations increase runtime. Compared to the other algorithms being compared, the suggested approach uses a considerably smaller population to demonstrate its capabilities. The initial size of the population for GA, PSO, ICA, and COA algorithms are considered to be 90, and the number of the initial population for the DA is considered to be 40 (Table 2). In this article, the range of FOPID controller parameters is considered based on Table 3. Three sets of experiments are conducted. The suggested method's rate of convergence is studied in the first set. The proposed method's performance in the four-tank system is evaluated in the second set, while the third set compares the proposed method's performance to that of current approaches.

Table 1. Introducing the parameters of the four-tank system

Parameter	Name	Values
a_1, a_3	Cross section of the outlet of the first and third tanks	0.081 cm ²
a_2, a_4	Cross-sectional of the outlet of the second and fourth tanks	0.067 cm ²
k_1, k_2	Constants (relationship between control voltages and the pumps' water flow)	3.33, 3.35
k_c	Fixed measuring device	0.50
h_{max}	Maximum height of tanks	20
A_2, A_4	Cross section of the second and fourth tanks	40
A_1, A_3	Cross section of the first and third tanks	30
g		981
γ_1, γ_2	The parameters of the three-way valve	0

All physical parameters follow the standard quadruple-tank benchmark used in the literature and were kept constant throughout all experiments. Cross-sections A_i come from tank geometry; outlet orifice areas a_i were obtained from steady-drain tests using Torricelli's relation $q_{out} = \sqrt{2gha}$; pump gains k_1, k_2 were measured by a linear fit between flow and voltage; and the sensor gain k_c was obtained by level calibration against a ruler. Units are shown explicitly in the Table 1.

Table 2. Parameter values of different meta-heuristic algorithms

Parameters		Values
Dragonfly Algorithm (DA)	Population size	40
	Number of repetitions	1000
	Random values	$r_1 = r_2 = [0,1]$
	Alignment weights	0.1
	Separation weight	0.1
	Coherence weight	0.7
	food factor	1
	Enemy factor	1
	Inertia factor	0.9-0.2
	constant	1.5
Genetic Algorithm (GA)	Population size	90
	Combination probability	0.8
	Mutation probability	0.03
	Number of repetitions	1000
Particle swarm optimization (PSO)	Population size	90
	Coefficients of social and cognitive parameters	2
Cuckoo Optimization Algorithm (COA)	Inertia weight factor	$[1, 0.99]$
	Population size	90
	Maximum number of cuckoos	130
Imperialist Competitive Algorithm (ICA)	population size	90
	The number of initial empires	12
	revolution rate	0.3

Algorithmic settings were initialized from canonical defaults in the original papers and then refined via a small pilot sweep to balance convergence and runtime under our problem size. Specifically, population sizes were explored in $\{100, 90, 80, 60, 40\}$; we selected 90 for GA/PSO/ICA/COA and 40 for DA based on the fastest convergence without degrading final RMSE. The iteration budget and remaining hyperparameters (e.g., inertia/learning factors or equivalent) follow these defaults unless stated otherwise.

Table 3. Range of FOPID controller parameters

Parameters	Values
k_p	(0, 1000]
k_i	(0, 1000]
k_d	(0, 1000]
λ	[0, 2]
μ	[0, 1.5]

The listed FOPID parameters are the optimization outputs obtained by running the Dragonfly Algorithm with the bounds $K_{p,i,d}[\min, \max]$, $\lambda, \mu[\min, \max]$ for each loop, using the RMSE_{total} objective and a fixed iteration budget. We performed multiple independent runs with different seeds and report the best-achieved parameter set (ties broken by lower IAE). All runs used the same plant model, sampling period, and fractional approximation as in the nominal setup.

5.2.1. Checking the convergence rate of the proposed method

To assess how quickly the proposed dragonfly algorithm converges, an experiment was conducted employing a system comprised of four tanks.

Fig. 3 shows the cost function versus iterations, illustrating convergence is important since the design of many systems takes time. Therefore, it is very important to reach the optimal solution in the minimum repetition. According to Fig. 3, Compared with GA, PSO, COA, and ICA, DA converged faster. Subsequently, it has achieved a minimized value within the Root Mean Square Error (RMSE) cost function.

5.2.2. Investigating the optimal parameter values for FOPID control by the proposed method in the four-tank system

It presents the parameter settings for two fractional-order PID controllers, derived using the method we introduced. You can find the specific values in Tables 4 to 6.

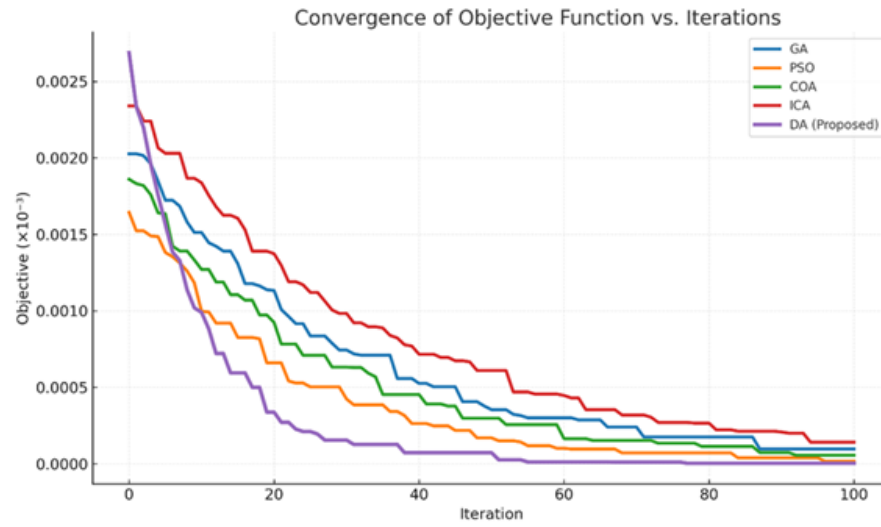


Fig. 3. Convergence rate

Table 4. FOPID parameters of the first controller

	K_{p1}	K_{i1}	K_{d1}	λ_1	μ_1
GA	435.548	1000	527.642	1.82	0.714
COA	445.453	1000	535.89	1.043	0.507
PSO	666.64	886.30	791.81	1.091	0.514
ICA	535.897	290.490	434.02	1.59	0.623
Proposed method	766.13	635.132	754.75	1.090	0.246

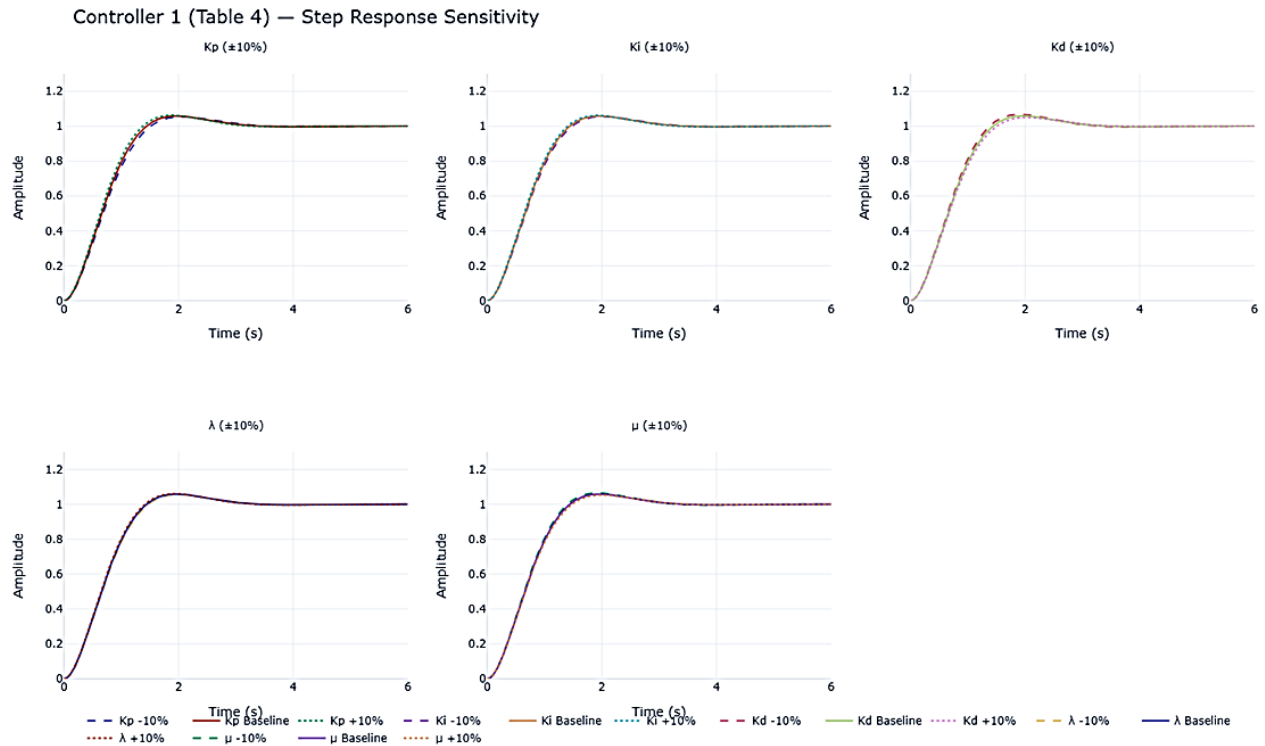


Fig. 4. Step response sensitivity for controller 1

The step response to a change of one parameter (K_p , K_i , K_d , λ , μ), to changes of -10% , baseline, and $+10\%$ is shown in each of the subplots.

- Putting more on K_p speeds the system up but can decrease damping.
- As K_i increases, the steady-state error is eliminated better but very high K_i values lower the stability.

- An increase of K_d will improve the damping and decrease overshoot; small K_d will make the system oscillate.
- Higher λ should be considered more powerful integral action, faster tracking, but can add an overshoot.
- Greater μ gives increased damping and more smooth response.
- All cases remain constant within the test of such percentages as plus/minus 10 percent.

Table 5. FOPID parameters of the second controller

	K_{p2}	K_{i2}	K_{d2}	λ_2	μ_2
GA	732.105	290.810	332.892	1.402	0.776
COA	848.572	978.522	581.073	1.127	0.816
PSO	703.51	280.91	332.781	1.803	0.775
ICA	686.497	669.527	558.426	0.789	0.826
Proposed method	174.31	1000	1000	1.947	0.393

Controller 2 (Table 5) — Step Response Sensitivity

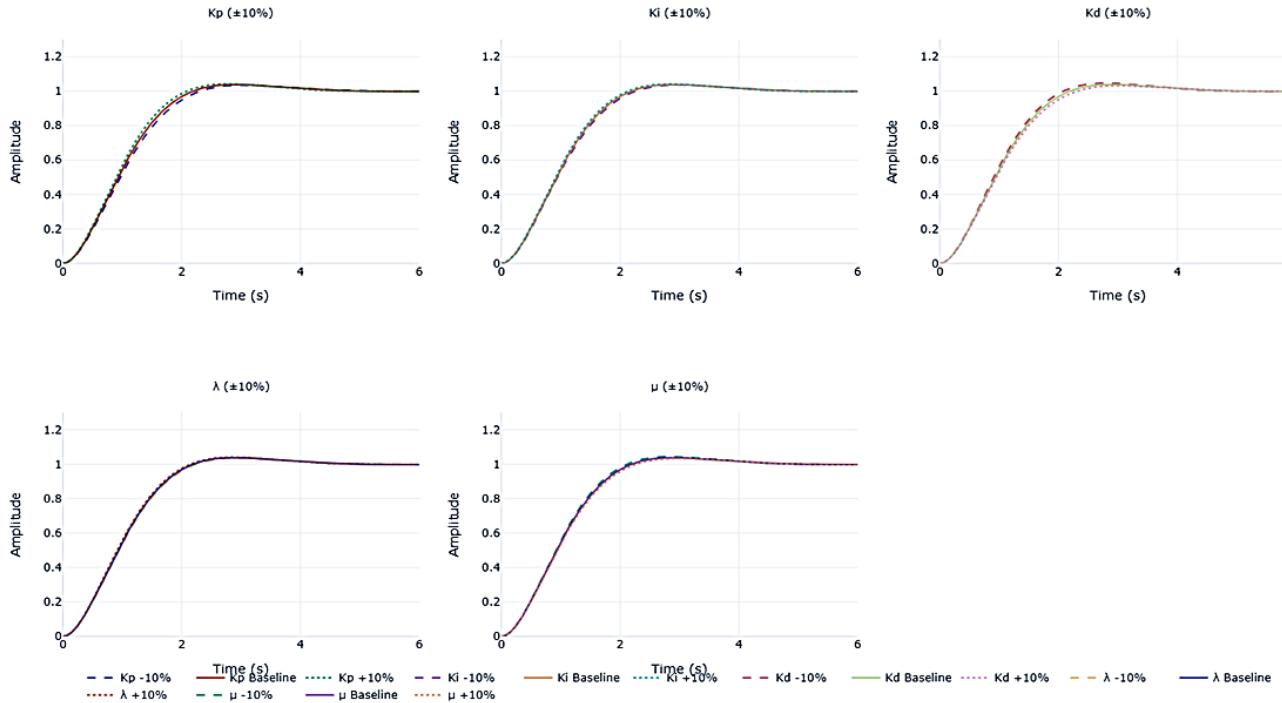


Fig. 5. Step response sensitivity for controller 2

- Repeat experiment with 10 percent change in sensitivity for K_p , K_i , K_d , λ , μ .
- The Rise time and settling are very dependent on K_i and λ hence showing sensitivity in the results.
- K_p has a bearing on response speed; less of K_p slows the system.
- μ and K_d majorly influence damping and overshooting.
- μ , when increased smooths the response by reducing overshoot and when reduced increases oscillations.
- All the tested cases are stable to changes of plus/minus 10 percent.

To show the effectiveness of the proposed method further, the step responses of the water levels h_1 and h_2 under the DA-tuned FOPID controller are compared with baseline methods like hybrids GA, PSO and ICA and COA. They are presented in Fig. 6(a and b).

Table 6 shows that the proposed DA-tuned FOPID achieves lower RMSE. Therefore, the FOPID parameters for both the first and secondary control systems, obtained through the proposed technique, yield outputs that surpass those of the alternative algorithms used for comparison. Considering the total RMSE cost, the FOPID controller derived from the proposed method demonstrates the overall best performance.

Table 6. The value of the cost function obtained by different algorithms

	$RMSE_1$	$RMSE_2$	$RMSE_{total}$
GA	0.302×10^{-3}	0.534×10^{-3}	0.981×10^{-3}
COA	0.195×10^{-3}	0.487×10^{-3}	0.674×10^{-3}
PSO	0.318×10^{-3}	0.609×10^{-3}	0.927×10^{-3}
ICA	0.219×10^{-3}	0.475×10^{-3}	0.627×10^{-3}
Proposed Method	0.182×10^{-3}	0.377×10^{-3}	0.559×10^{-3}

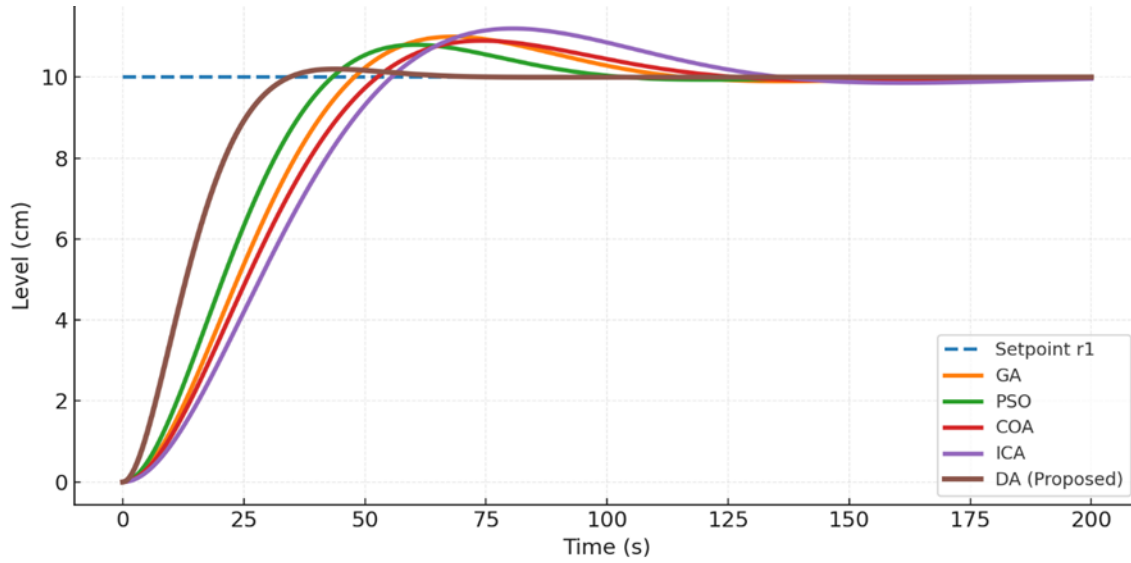


Fig. 6a. Step response of level h1

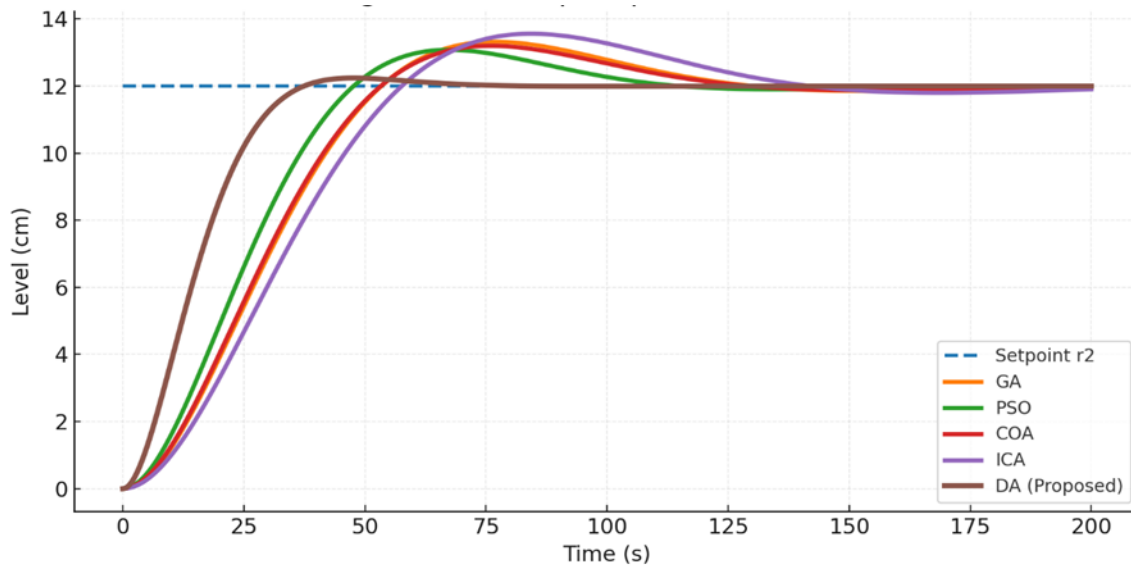


Fig. 6b. Step response of level h2

How RMSE_total was computed: For each output we first compute $RMSE_k = \sqrt{\frac{1}{n} \sum_{k=1}^n e^2(k)}$. The aggregate value reported as RMSE_total is $\sqrt{(RMSE_1^2 + RMSE_2^2)/2}$, evaluated over the same time window and sampling for both outputs.

If your two outputs have very different scales or different importance, you can state a weighted version: $RMSE_{total} = \sqrt{w_1 RMSE_1^2 + w_2 RMSE_2^2}$ with $w_1 + w_2 = 1$. (In Table 6 it used equal weights $w_1 = w_2 = 0.5$.)

5.2.3. Checking the proposed method with recent methods

The proposed method's performance was compared to that of current techniques in this section, and the findings are shown in Table 7. Table 7 indicates that method [33] achieves lower ISE and IAE than method [29], The proposed method further reduces both metrics, achieving the best performance.

Table 7. Checking the performance of the proposed method with other methods

Evaluation criteria	Method [33]	Method[29]	Suggested method
ISE	48.40	49.55	46.02
IAE	107.11	109.61	102.81

6. Conclusion

In this paper will tune a FOPID controller using the Dragonfly Algorithm (DA). The system considered is a four-tank model and the used cost function is the Root Mean Squared Error (RMSE). Is chosen because it does well as compared to other meta-heuristics in terms of convergence. It is better than the GA, PSO, ICA, and cuckoo optimization algorithm in the quadruple-tank system simulations. Future development will focus on tuning other plants and hybrid time/frequency-domain objectives., considering both time-domain properties and frequency-domain properties.

Acknowledgment

To begin, the authors would like to thank the Department of Electrical Engineering, Islamic Azad University (Shiraz Branch), and Middle Technical University for their institutional support. The views expressed are those of the authors alone, for bestowing upon us both this chance and the necessary abilities to achieve. We are truly indebted to the Department of Electrical Engineering at the Shiraz campus of Islamic Azad University; their consistent backing, generosity, and provisions throughout our research have been invaluable. Furthermore, we extend my gratitude to Middle Technical University for assisting in the publishing of this study.

References

- [1] J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, Transactions of the American society of mechanical engineers, vol. 64, no. 8, pp. 759-765, 1942. doi.org/10.1115/1.4019264
- [2] S. Kargar, Compensation of actuator's saturation by using fuzzy logic and imperialist competitive algorithm in a system with PID controller, vol. 3, no. 11, pp. 21-26, 2013.
- [3] K. J. Åström and T. Hägglund, The future of PID control, Control engineering practice, vol. 9, no. 11, pp. 1163-1175, 2001. doi.org/10.1016/S0967-0661(01)00062-4.
- [4] O. Safarzadeh and O. Noori-kalkhoran, A fractional PID controller based on fractional point kinetic model and particle swarm optimization for power regulation of SMART reactor, Nuclear Engineering and Design, vol. 377, p. 111137, 2021. doi.org/10.1016/j.nucengdes.2021.111137.
- [5] L. F. d. S. Pereira, E. Batista, M. A. de Brito, and R. B. Godoy, A robustness analysis of a fuzzy fractional order pid controller based on genetic algorithm for a dc-dc boost converter, Electronics, vol. 11, no. 12, p. 1894, 2022. doi.org/10.3390/electronics11121894.
- [6] 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, no. 1, pp. 310-320, 2022. doi.org/10.1007/s12555-020-0847-1.
- [7] L. H. Abood and B. K. Oleiwi, Design of fractional order PID controller for AVR system using whale optimization algorithm, Indonesian Journal of Electrical Engineering and Computer Science, vol. 23, no. 3, pp. 1410-1418, 2021. doi: 10.11591/ijeecs.v23.i3.pp1410-1418.
- [8] A. Koszewnik, E. Pawluszewicz, and M. Ostaszewski, Experimental studies of the fractional PID and TID controllers for industrial process, International Journal of Control, Automation and Systems, vol. 19, pp. 1847-1862, 2021. doi.org/10.1007/s12555-020-0123-4.
- [9] M. A. Faraj and A. M. Abbood, Fractional order PID controller tuned by bat algorithm for robot trajectory control, Indonesian Journal of Electrical Engineering and Computer Science, vol. 21, no. 1, pp. 74-83, 2021. doi: 10.11591/ijeecs.v21.i1.pp74-83.
- [10] M. Saadatmand, G. B. Gharehpetian, I. Kamwa, P. Siano, J. M. Guerrero, and H. Haes Alhelou, A Survey on FOPID controllers for LFO damping in Power systems using synchronous generators, FACTS devices and inverter-based power plants, Energies, vol. 14, no. 18, p. 5983, 2021. doi.org/10.3390/en14185983.
- [11] I. Paducel, C. O. Saffirescu, and E.-H. Dulf, Fractional Order Controller Design for Wind Turbines, Applied Sciences, vol. 12, no. 17, p. 8400, 2022. doi.org/10.3390/app12178400.
- [12] S. Yaghoobi and H. Mojallali, Tuning of a PID controller using improved chaotic Krill Herd algorithm, Optik, vol. 127, no. 11, pp. 4803-4807, 2016. doi.org/10.1016/j.ijleo.2016.01.055.
- [13] A. Mughees and S. A. Mohsin, Design and control of magnetic levitation system by optimizing fractional order PID controller using ant colony optimization algorithm, IEEE Access, vol. 8, pp. 116704-116723, 2020. doi: 10.1109/ACCESS.2020.3004025.
- [14] B. Hekimoğlu, Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm, IEEE access, vol. 7, pp. 38100-38114, 2019. doi: 10.1109/ACCESS.2019.2905961.
- [15] M. Regad, M. Helaimi, R. Taleb, H. Gabbar, and A. Othman, Optimal frequency control in microgrid system using fractional order PID controller using krill herd algorithm, Электротехника и электромеханика, no. 2 (eng), pp. 68-74, 2020. doi:10.20998/2074-272x.2020.2.11.
- [16] S. Padhee, A. Gautam, Y. Singh, and G. Kaur, A novel evolutionary tuning method for fractional order PID controller, International Journal of Soft Computing and Engineering, vol. 1, no. 3, pp. 1-9, 2011.
- [17] S. Gad, H. Metered, A. Bassuiny, and A. Abdel Ghany, Multi-objective genetic algorithm fractional-order PID controller for semi-active magnetorheologically damped seat suspension, Journal of Vibration and Control, vol. 23, no. 8, pp. 1248-1266, 2017. doi.org/10.1177/1077546315591620.
- [18] Z. Chen, X. Yuan, B. Ji, P. Wang, and H. Tian, Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II, Energy Conversion and Management, vol. 84, pp. 390-404, 2014. doi.org/10.1016/j.enconman.2014.04.052.
- [19] J. Wan, B. He, D. Wang, T. Yan, and Y. Shen, Fractional-order PID motion control for AUV using cloud-model-based quantum genetic algorithm, IEEE Access, vol. 7, pp. 124828-124843, 2019. doi: 10.1109/ACCESS.2019.2937978.
- [20] S. Jaiswal, C. Suresh Kumar, M. M. Seepana, and G. U. B. Babu, Design of fractional order PID controller using genetic algorithm optimization technique for nonlinear system, Chemical Product and Process Modeling, vol. 15, no. 2, p. 20190072, 2020. doi.org/10.1515/cppm-2019-0072.
- [21] R. Dimeo and K. Y. Lee, Boiler-turbine control system design using a genetic algorithm, IEEE transactions on energy conversion, vol. 10, no. 4, pp. 752-759, 1995. doi: 10.1109/60.475849

- [22] R. Alayi, F. Zishan, S. R. Seyednouri, R. Kumar, M. H. Ahmadi, and M. Sharifpur, Optimal load frequency control of island microgrids via a PID controller in the presence of wind turbine and PV, Sustainability, vol. 13, no. 19, p. 10728, 2021. doi.org/10.3390/su131910728.
- [23] A. X. R. Irudayaraj et al., A Matignon's theorem based stability analysis of hybrid power system for automatic load frequency control using atom search optimized FOPID controller, IEEE Access, vol. 8, pp. 168751-168772, 2020. doi: 10.1109/ACCESS.2020.3021212.
- [24] A. Kumar and S. Suhag, Whale optimisation algorithm tuned fractional order PID controller for load frequency control of multi-source power system, International Journal of Bio-Inspired Computation, vol. 13, no. 4, pp. 209-221, 2019. doi.org/10.1504/IJBIC.2019.100153.
- [25] I. Kasiyanto, H. Firdaus, Q. Lailiyah, I. Supono, N. Kusrandar, and I. Supono, Advanced Control for Quadruple Tank Process, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), vol. 10, no. 1, pp. 1-16, 2024. doi: 10.26555/jiteki.v%vi%i.27124.
- [26] I. Podlubny, Fractional-order systems and PI/spl lambda//D/spl mu//-controllers, IEEE Transactions on automatic control, vol. 44, no. 1, pp. 208-214, 1999. doi: 10.1109/9.739144.
- [27] N. Lachhab, F. Svaricek, F. Wobbe, and H. Rabba, Fractional order PID controller (FOPID)-toolbox, in 2013 European control conference (ECC), 2013: IEEE, pp. 3694-3699. doi: 10.23919/ECC.2013.6669365.
- [28] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural computing and applications, vol. 27, pp. 1053-1073, 2016. doi.org/10.1007/s00521-015-1920-1.
- [29] N. A. Al-Awad, Optimal control of quadruple tank system using genetic algorithm, International Journal of Computing and Digital Systems, vol. 8, no. 01, pp. 51-59, 2019. doi.org/10.12785/ijcds/080106.
- [30] R. Rajabioun, Cuckoo optimization algorithm, Applied soft computing, vol. 11, no. 8, pp. 5508-5518, 2011. doi.org/10.1016/j.asoc.2011.05.008.
- [31] M. Zamani, M. Karimi-Ghartemani, N. Sadati, and M. Parniani, Design of a fractional order PID controller for an AVR using particle swarm optimization, Control Engineering Practice, vol. 17, no. 12, pp. 1380-1387, 2009. doi.org/10.1016/j.conengprac.2009.07.005.
- [32] A. Kaveh and A. Kaveh, Imperialist competitive algorithm, Advances in Metaheuristic Algorithms for Optimal Design of Structures, pp. 353-373, 2017. doi.org/10.1007/978-3-319-46173-1_11.
- [33] N. Katal and S. Narayan, Design of robust fractional order PID controllers for coupled tank systems using multi-objective particle swarm optimisation, International Journal of Systems, Control and Communications, vol. 8, no. 3, pp. 250-267, 2017. doi.org/10.1504/IJSCC.2017.085496.