algorithms

MDPI

Article

Performance Evaluation of Fractional Proportional–Integral–Derivative Controllers Tuned by Heuristic Algorithms for Nonlinear Interconnected Tanks

Raúl Pazmiño [1], Wilson Pavon [2,*], Matthew Armstrong [3] and Silvio Simani [4]

[1] Automation and Control Postgraduate Program, Universidad Politécnica Salesiana, Quito 170525, Ecuador; rpazminop1@est.ups.edu.ec
[2] Electronics Engineering, Universidad Politécnica Salesiana, Quito 170525, Ecuador
[3] School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK; matthew.armstrong@ncl.ac.uk
[4] Dipartimento di Ingegneria, Università degli Studi di Ferrara, 35-44121 Ferrara, Italy; silvio.simani@unife.it
* Correspondence: wpavon@ups.edu.ec

Abstract: This article presents an in-depth analysis of three advanced strategies to tune fractional PID (FOPID) controllers for a nonlinear system of interconnected tanks, simulated using MATLAB. The study focuses on evaluating the performance characteristics of system responses controlled by FOPID controllers tuned through three heuristic algorithms: Ant Colony Optimization (ACO), Grey Wolf Optimizer (GWO), and Flower Pollination Algorithm (FPA). Each algorithm aims to minimize its respective cost function using various performance metrics. The nonlinear model was linearized around an equilibrium point using Taylor Series expansion and Laplace transforms to facilitate control. The FPA algorithm performed better with the lowest Integral Square Error (ISE) criterion value (297.83) and faster convergence in constant values and fractional orders. This comprehensive evaluation underscores the importance of selecting the appropriate tuning strategy and performance index, demonstrating that the FPA provides the most efficient and robust tuning for FOPID controllers in nonlinear systems. The results highlight the efficacy of meta-heuristic algorithms in optimizing complex control systems, providing valuable insights for future research and practical applications, thereby contributing to the advancement of control systems engineering.

Keywords: cost function minimization; fractional order PID controllers; heuristic optimization algorithms; meta-heuristic tuning strategies; nonlinear system modeling; performance indez tuning; system response analysis

Citation: Pazmiño, R.; Pavon, W.; Armstrong, M.; Simani, S. Performance Evaluation of Fractional Proportional–Integral–Derivative Controllers Tuned by Heuristic Algorithms for Nonlinear Interconnected Tanks. *Algorithms* **2024**, *17*, 306. https://doi.org/10.3390/a17070306

Academic Editor: Mircea-Bogdan Radac

Received: 7 June 2024
Revised: 3 July 2024
Accepted: 6 July 2024
Published: 10 July 2024

1. Introduction

The dynamic behavior of interconnected tank systems, characterized by their nonlinear properties, poses significant challenges in control engineering. Fractional PID (Proportional–Integral–Derivative) controllers have emerged as a robust solution to address these challenges due to their ability to provide extra freedom in tuning. This research paper delves into the tuning of fractional PID controllers using heuristic optimization algorithms. It aims to evaluate and compare the effectiveness of various tuning strategies based on different performance indices. The goal is to identify the most efficient algorithmic approach to optimize the control parameters of a nonlinear plant, focusing on the derivation of a transfer function through linearization at an equilibrium point and subsequent application of fractional control.

This paper is structured to facilitate a comprehensive understanding of the methodologies and findings. Section 2 reviews the state of the art, encompassing the modeling of nonlinear systems, the principles underlying fractional PID controllers, heuristic optimization algorithms, and system curve error criteria. Section 3 describes the process of deriving

the plant's mathematical model, followed by its linearization using the Taylor Series to obtain a suitable transfer function for fractional control application. Section 4 presents a detailed analysis of the system's response curves, performance indices, and the iterative performance of the tuning algorithms. Finally, Section 5 synthesizes the conclusions drawn from the comparative analysis of the experimental results, emphasizing the implications and potential applications of the findings in control systems engineering.

2. State of the Art

The paper by [1] demonstrates the effectiveness of heuristic algorithms in tuning Fractional Order LQI (FOLQI) controllers for a highly interacting Multiple Input Multiple Output (MIMO) quadruple tank system. The study highlights the superior performance and robustness of the FOLQI controllers over traditional Integer Order LQI (IOLQI) controllers. This paper introduces a novel DualMode Adaptive Fractional Order PI controller with an Adaptive Feedforward controller for the Quadruple Tank Process, showcasing enhanced performance in reference signal tracking across all output ranges through numerical simulation and experimentation. The paper [2] presents a self-tuning Dual Mode Adaptive Fractional Order PI and Adaptive Feedforward controller for the Quadruple Tank Process, demonstrating superior performance.

The paper by [3] proposes an optimum H-infinity norm-based controller for a Single Capacity (SC) tank system, demonstrating significant performance improvements over P, PI, and PID controllers, as well as a Grey Wolf Optimizer (GWO)-tuned PID controller, in terms of settling time, overshoot, and steady-state error. The paper by [4] uses a simplified genetic algorithm to tune PID controllers for a multipurpose water tank plant in Peru, showing superior performance over traditional methods in optimizing control parameters for flow, pressure, level, and temperature loops.

Table 1 summarizes recent advancements in intelligent self-tuning PID controller methodologies applied to various nonlinear tank systems. The table highlights key research works, including self-tuning fuzzy PID controllers, meta-heuristic algorithms for PID tuning, and hybrid approaches for optimizing control performance. Each study demonstrates the application of different optimization techniques, such as genetic algorithms, particle swarm optimization, and cuckoo search optimization, to improve the efficiency and robustness of PID controllers. The table also indicates whether the studies involved a mathematical tank model and the PID methodology, showcasing a comprehensive overview of the latest trends in advanced control strategies for nonlinear systems.

Table 1. Summary of intelligent self-tuning fuzzy PID controller research.

Main Author and Reference	Methodology	Mathematical Tank Model	PID Methodology
Bhandare, D [5]	Self-tuning fuzzy PID for dynamic model of Coupled Tank System (CTS)	X	X
Amuthambigaiyin Sundari K. [6]	Meta-heuristic PID tuning for two-tank system	X	X
Chauhan, S [7]	Modified ACO for PID tuning in CTS	X	X
Sahin, Ali Kivanc [8]	GWO and battle royale optimization (BRO) for decentralized PI in Quadruple-Tank Process (QTP)	X	X
Nedumal Pugazhenthi P. [9]	Hybrid Artificial Bee Colony (ABC) for nonlinear PID in CSTR	X	X
Kumar, Jitendra [10]	CSA-tuned Nonlinear Proportional Plus Integral Plus Derivative (NPID) for surge tank control	X	X
Jaiswal, S [11]	Genetic Algorithm (GA)-tuned FOPID for nonlinear conical tank	X	X
Febina C. [12]	Particle Swarm Optimization (PSO)-tuned Neural Network (NN) Robustness, Setpoint tracking, Disturbance Rejection, Aggressiveness (RTDA) for nonlinear conical tank	X	X

Table 2 and Figure 1 present a comprehensive overview of significant research advancements in intelligent self-tuning PID controllers applied to various nonlinear tank systems. The table highlights key studies emphasizing methodologies such as self-tuning fuzzy PID, meta-heuristic algorithms, and hybrid approaches, indicating whether they involve mathematical tank models and PID methodologies. For instance, Bhandare, D [5] focuses on a self-tuning fuzzy PID for dynamic CTS, while Amuthambigaiyin Sundari K. [6] explores meta-heuristic PID tuning for two-tank systems. The bar graph visualizes the citation counts of the top 10 cited articles, underscoring these pioneering research efforts' significant impact and scholarly influence.

Table 2. Top 10 papers with methodologies.

Main Author	Reference	Citations	Methodology Summary
Geibel, P.	[13]	214	Risk-sensitive RL for constrained control
López-Ibáñez, M.	[14]	203	ACO for optimal pump control in water networks
Li, J.A.	[15]	147	Optimal allocation of empty containers between ports
Sharafi, M.	[16]	93	Design hybrid renewable energy systems in buildings
Duan, N.	[17]	85	Reliability-based design of pumping systems
Ronen, D.	[18]	84	Shipments planning in marine inventory routing
Braun, M.W.	[19]	80	Model-on-demand identification for nonlinear processes
Ruelens, F.	[20]	67	Demand response using batch RL in water heaters
Risbeck, M.J.	[21]	65	Real-time cost optimization for HVAC equipment
Drees, K.H.	[22]	61	Rule-based control strategy for ice storage systems

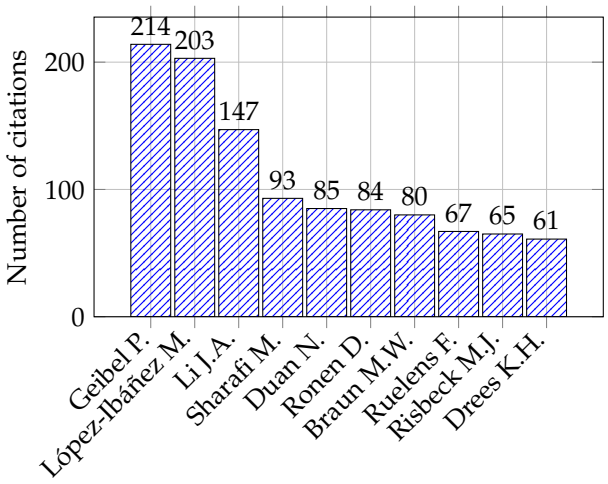

Figure 1. Number of citations of the top 10 most cited articles.

3. Methodology

The proposed system focuses on height control in a system of interconnected tanks. The modeling is grounded in Bernoulli's equations for laminar flow, accounting for flow rate and balance associated with the energy conservation of mass flow at various points. Figure 2 illustrates the plant model.The parameters used in the mathematical model of the interconnected tank system are summarized in Table 3. This table lists the symbols and their corresponding descriptions, including flow rates (U_{in}, U_{12}, U_{out}), tank heights (H_1, H_2), cross-sectional area (A), and coefficients (c_1, c_2). The redefined variables (Z_1, Z_2) simplify the system equations, while K_1 and a_1 represent the gain and valve opening for the inlet flow. These parameters are crucial for understanding the system dynamics and deriving the transfer functions for control purposes.

Table 3. Parameters for the mathematical model of the interconnected tank system.

Symbol	Description
U_{in}	Inlet flow rate from the pump
U_{12}	Intermediate flow between tanks
U_{out}	Outlet flow rate through the orifice
A	Cross-sectional area of the tank
H_1	Height of liquid in tank 1
H_2	Height of liquid in tank 2
v	Velocity of liquid through an orifice
K_1	Gain for the inlet flow rate
a_1	Valve opening for the inlet flow
Z_1	Redefined variable for height of tank 2
Z_2	Redefined variable for height difference between tanks
c_1	Coefficient related to tank 2 outflow
c_2	Coefficient related to intermediate flow

The percentage change in liquid volume in the tank is determined by the inlet flow rate from the pump minus the outlet flow rate through the orifice, as described by Equations (1) and (2). The velocity (v) of liquid passing through an orifice is akin to that of an object falling freely from a height h, represented by Torricelli's formula (Equation (3)). Redefining Torricelli's formula and flow balance equations, the intermediate flow (U_{12}) and outflow (U_{out}) are given by Equations (4) and (5), respectively. The inlet flow (U_{in}) is modeled with gain K_1 and valve opening a_1 as shown in Equation (6). Through variable changes, z_1 is defined as H_2 with a coefficient c_1, in Equation (7), and z_2 as $H_1 - H_2$ with c_2, in Equation (8). These transformations simplify the complex dynamics of the tank system.

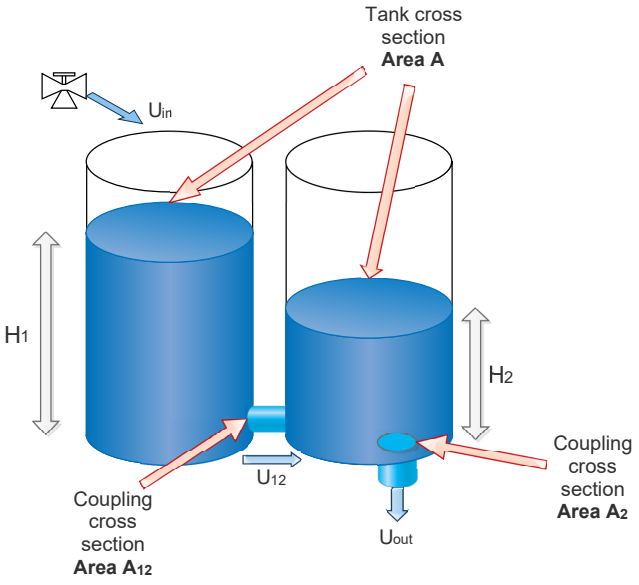

Figure 2. Diagram of the interconnected tank system with flow and height parameters.

$$U_{in} - U_{12} = A\frac{dH_1}{dt} \tag{1}$$

$$U_{12} - U_{out} = A\frac{dH_2}{dt} \tag{2}$$

$$v = \sqrt{2gh} \tag{3}$$

$$U_{12} = A_{12}\sqrt{2g(H_1 - H_2)} \tag{4}$$

$$U_{out} = A_2\sqrt{2gH_2} \tag{5}$$

$$U_{in} = K_1 a_1 \tag{6}$$

$$z_1 = H_2 \qquad c_1 = \frac{A_2\sqrt{2g}}{A} \tag{7}$$

$$z_2 = H_1 - H_2 \qquad c_2 = \frac{A_{12}\sqrt{2g}}{A} \tag{8}$$

The derived system focuses on the dynamic behavior of Z_1 and Z_2. The rate of change of Z_1 is determined by the differential Equations (2), (4), (5) and (7), leading to Equation (9). For Z_2, by subtracting Equation (2) from Equation (1) and substituting Equations (4) through (6), we derive Equation (10). These nonlinear differential equations are subsequently linearized using Taylor Series expansion. The equilibrium positions of state variables Z_1^* and Z_2^* are determined by setting their derivatives to zero, resulting in Equation (11). The first derivative of the Taylor Series is used for linearization, as shown in Equation (12). Analyzing the differential equation for Z_2, the linearized form is derived in Equation (13). Applying the Laplace transform yields Equation (14). Similarly, for Z_1, the linearized form is given in Equation (15), and its Laplace transform is shown in Equation (16). These transformations enable a simplified analysis of the system dynamics.

$$\frac{dZ_1}{dt} = -c_1\sqrt{Z_1} + c_2\sqrt{Z_2} \tag{9}$$

$$\frac{dZ_2}{dt} = c_1\sqrt{Z_1} - 2c_2\sqrt{Z_2} + \frac{K_1 a_1}{A} \tag{10}$$

$$Z_1^* = \left(\frac{c_2\sqrt{Z_2}}{c_1}\right)^2 \quad Z_2^* = \left(\frac{c_1\sqrt{Z_1}}{2c_2} + \frac{K_1 a_1}{2Ac_2}\right)^2 \tag{11}$$

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)\Delta x \tag{12}$$

$$\frac{d\Delta Z_2}{dt} = \frac{K_1}{A}\Delta a_1 + \frac{c_1}{2\sqrt{Z_1^*}}\Delta Z_1 - \frac{c_2}{\sqrt{Z_2^*}}\Delta Z_2 \tag{13}$$

$$\Delta Z_2(s)\left[\frac{s\sqrt{Z_2^*} + c_2}{\sqrt{Z_2^*}}\right] = \frac{2K_1\sqrt{Z_1^*}\Delta a_1(s) + c_1 A\Delta Z_1(s)}{2A\sqrt{Z_1^*}} \tag{14}$$

$$\frac{d\Delta Z_1}{dt} = \frac{-c_1}{2\sqrt{Z_1^*}}\Delta Z_1 + \frac{c_2}{2\sqrt{Z_2^*}}\Delta Z_2 \tag{15}$$

$$\Delta Z_1(s)\left[s + \frac{c_1}{2\sqrt{Z_1^*}}\right] = \frac{c_2}{2\sqrt{Z_2^*}}\Delta Z_2(s) \tag{16}$$

Replacing Equation (17) into Equation (18) results in a transfer function that relates the height of the tank to the flow inlet. This transfer function is expressed as Equation (19), where $\Delta Z_1(s)$ represents the change in the height of tank 1, and $\Delta a_1(s)$ represents the change in inlet flow. Equation (19) provides the resulting transfer function. The transfer function describing the plant for subsequent control design is obtained by utilizing the equilibrium positions derived in Equation (20). This transfer function, incorporating system parameters and equilibrium variables, is shown in Equation (21).

$$\Delta Z_2(s)\left[\frac{s\sqrt{Z_2^*}+c_2}{\sqrt{Z_2^*}}\right]=\frac{2K_1\sqrt{Z_1^*}\Delta a_1(s)+c_1A\Delta Z_1(s)}{2A\sqrt{Z_1^*}} \tag{17}$$

$$\Delta Z_1(s)\left[s+\frac{c_1}{2\sqrt{Z_1^*}}\right]=\frac{c_2}{2\sqrt{Z_2^*}}\Delta Z_2(s) \tag{18}$$

$$\frac{\Delta Z_1(s)}{\Delta a_1(s)}=\frac{\frac{K_1}{2A\sqrt{Z_2^*}}}{s^2+\frac{c_1Z_2^*+2c_2Z_1^*}{2\sqrt{Z_1^*Z_2^*}}s+\frac{c_1c_2}{4\sqrt{Z_1^*Z_2^*}}} \tag{19}$$

$$Z_1^*=\left(\frac{c_2\sqrt{Z_2}}{c_1}\right)^2 \quad Z_2^*=\left(\frac{c_1\sqrt{Z_1}}{2c_2}+\frac{K_1a_1}{2Ac_2}\right)^2 \tag{20}$$

$$\frac{\Delta Z_1(s)}{\Delta a_1(s)}=\frac{0.3461}{s^2+0.01305s+0.008458} \tag{21}$$

The fractional PID controller tuning values are optimized using heuristic algorithms. These algorithms have distinct parameters, and an objective function is assigned to each test. The heuristic optimization process ensures that the fractional PID controller attains optimal performance in regulating the height of the interconnected tank system. The PID controller is formulated as Equation (22). The optimization process involves adjusting K_p, K_i, and K_d to minimize the control error and achieve the desired system performance. The optimization process consists of two stages: the first stage optimizes the proportional, integral, and derivative constants, while the second stage focuses on the integral and derivative orders. The algorithm continues to run until the specified number of iterations is completed. The results, including tuned values and performance indices for each test, are printed for further analysis and interpretation.

The objective function for optimizing the fractional PID controller tuning values is designed to minimize control errors in regulating the height of the interconnected tank system, in Equation (23), and the explanation of the parameters on the Table 4. This objective function, denoted as J, is formulated based on the performance indices Integral Absolute Error (IAE), Integral Square Error (ISE), and Integral Time-weighted Absolute Error (ITAE). The controller's performance is then evaluated using Overshoot, Establishment Time, and Rise Time. The results are compared, and the best method is selected visually based on these criteria.

Table 4. Parameters for the fractional PID controller.

Symbol	Description
$u(t)$	Control signal
$e(t)$	Error signal, representing the difference between the desired and actual height of the tank
K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
λ	Order of the fractional integration
μ	Order of the fractional differentiation
$\Gamma(\lambda)$	Gamma function, generalizes the factorial function to non-integer values

$$u(t)=K_pe(t)+K_i\frac{1}{\Gamma(\lambda)}\int_0^t(t-\tau)^{\lambda-1}e(\tau)\,d\tau+K_d\frac{d^\mu e(t)}{dt^\mu} \tag{22}$$

$$\min J=\mathrm{OF} \tag{23}$$

where the objective function (OF) can be one of the following performance indices:

$$\text{IAE} = \int_0^T |e(t)|\, dt \tag{24}$$

$$\text{ISE} = \int_0^T e^2(t)\, dt \tag{25}$$

$$\text{ITAE} = \int_0^T t|e(t)|\, dt \tag{26}$$

$$\text{ITSE} = \int_0^T te^2(t)\, dt \tag{27}$$

The heuristic optimization process ensures that the fractional PID controller attains optimal performance in regulating the height of the interconnected tank system. The objective functions, based on performance indices such as IAE, ISE, and ITAE, steer the optimization to minimize control errors. The results from the optimized controller parameters are then scrutinized to ascertain the robustness of the control strategy, thereby ensuring the reliable and efficient performance of the nonlinear tank system. By leveraging the transfer function derived from the linearized system equations and applying advanced heuristic optimization techniques, the proposed methodology provides a systematic approach to achieve precise height control in interconnected tank systems, enhancing overall system stability and performance.

3.1. Ant Colony Optimization

Ant Colony Optimization (ACO) algorithms are inspired by the foraging behavior of ants, particularly how they locate food and communicate this information to their colony through pheromone trails. This approach is suitable for combinatorial optimization problems, quadratic assignments, derivative methods, and stochastic problems. Initially, ants wander randomly; when one locates food, it returns to the colony while depositing pheromones, marking the path for others. Ants that follow this path reinforce the pheromone trail, making it more attractive. Over time, pheromone marks fade, giving shorter and more frequently traveled paths an advantage. In ACO algorithms, artificial ants traverse a search space to locate optimal solutions. These artificial ants track their locations and the candidate solutions they encounter, using this information to guide subsequent iterations. This method is often combined with local search techniques activated after identifying a promising area. ACO's combined optimization ability and robustness make it highly effective. It was introduced ACO to the rough set theory, converting minimal-feature subset problems into optimal path combination problems, thus proposing a feature selection method based on ACO [23,24].

The ACO algorithm can be outlined through several key steps. Parameter Initialization involves setting the pheromone intensity $\tau_{i,j}(t)$, and parameters α (pheromone intensity control), β (visibility control), ρ (pheromone evaporation rate), and k (number of ants). The Transmission State Rule determines the probability that an ant at one point will move to another, defined by Equation (28). Here, S_k is the set of objects to be scheduled, $\tau_{i,j}$ is the pheromone amount between points i and j, and $\eta_{i,j} = \frac{1}{t_{i,j}}$ represents the visibility of connection (i, j). The parameters α and β control the relative weights of pheromone and visibility, respectively, both ranging from 0 to 1. Pheromone Update occurs after ants traverse a path, updating pheromones via Equation (29). The variable $\tau_{i,j}$ represents the pheromone intensity at time t, with ρ as the evaporation rate ($0 < \rho < 1$). $\Delta\tau_{i,j}(t)$ is the probability that an ant at point i will choose point j, defined by Equation (30). A Global Pheromone Update is performed after ants complete their paths, marking the best routes based on the highest pheromone concentration. This update uses Equation (31), where $\Delta\tau_{i,j}(t)$ is defined in Equation (32).

$$P_{i,j}^{k}(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{i=1}^{n}[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}(t)]^{\beta}}, & \text{if}(i,j) \in S_k \\ 0, & \text{if}(i,j) \notin S_k \end{cases} \tag{28}$$

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \Delta\tau_{i,j}(t) \tag{29}$$

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{i=1}^{n}[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}(t)]^{\beta}}, & \text{if}(i,j) \text{ is passed by ants} \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \Delta\tau_{i,j}(t) \tag{31}$$

$$\Delta\tau_{i,j}(t) = \begin{cases} \zeta T, & \text{if}(i,j) \text{ is passed by ant-k} \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

The suitability or quality of a path (solution) in ACO is linked to the pheromone concentration on that path. Higher pheromone concentrations indicate better paths, which are more likely to be chosen. The roulette-wheel selection technique selects candidates based on their probability $P_{i,j}$, which is influenced by the heuristic value $\eta_{i,j}$. This method balances probabilistic selection with the exploration of new paths, enhancing the robustness of the ACO algorithm in finding optimal solutions for discrete optimization problems.

The parameters used in the ACO algorithm are summarized in Table 5. This table lists the symbols and their corresponding descriptions, including pheromone intensity ($\tau_{i,j}(t)$), control parameters (α, β, ρ), and the number of ants (k). Additionally, it includes the probability of ant movement ($P_{i,j}^{k}(t)$), visibility value ($\eta_{i,j}$), change in pheromone intensity ($\Delta\tau_{i,j}(t)$), and other related constants (ζ, T). These parameters are crucial for understanding the behavior and optimization process of the ACO algorithm.

Table 5. ACO algorithm parameters.

Symbol	Description
$\tau_{i,j}(t)$	Pheromone intensity at time t between points i and j
α	Parameter controlling the influence of pheromone intensity
β	Parameter controlling the influence of visibility
ρ	Pheromone evaporation rate
k	Number of ants
$P_{i,j}^{k}(t)$	Probability of ant k moving from point i to j
$\eta_{i,j}$	Visibility value between points i and j
$\Delta\tau_{i,j}(t)$	Change in pheromone intensity
S_k	Set of objects to be scheduled
ζ	Constant used in Global Pheromone Update
T	Total unit of the path taken by an ant

The proposed Algorithm 1, titled "FOPID Tuning Ant Colony Optimization" systematically tunes Fractional Order PID controllers using ACO. Initially, the search space and algorithm parameters are declared as outlined in Table 5. Allowable discrete values are initialized with equal pheromone levels (τ). The algorithm iterates through several steps: computing the selection probability of design variables (Equation (28)), setting probability ranges, and generating random positions for each ant based on cumulative probability ranges (roulette-wheel design). The control signal values are assigned and the objective function is evaluated. The algorithm updates the best and worst objective functions, strengthens the pheromone of the best path, and evaporates pheromones from other paths. The tuning values are printed after completing the iterations, ensuring optimal

controller parameters. This structured approach leverages the robustness and efficiency of ACO in optimizing FOPID controllers for complex systems.

Algorithm 1 FOPID tuning Ant Colony Optimization

Step 1: Declare search space and algorithm parameters (Table 5)
Step 2: Initialize allowable discrete values m with equal pheromone τ
for i = 1:T **do**
  for j = 1:ls **do**
    Step 3: Compute selection probability of design variables (Equation (28))
  for j = 1:ls **do**
    Set probability ranges $r\vec{w}s, r\vec{w}e$ with steps of p_j
  for k = 1:N **do**
    Step 4: Compute cumulative probability ranges (roulette-wheel design)
    Step 5: Generate random positions $\vec{r}_k = rand(1)$ for each ant
    for m = 1:ls **do**
      if $rws(m) < r_k < rwe(m)$, assign $\vec{f}_m = \vec{x}_m$
  for k = 1:N **do**
    Step 6: Assign $\vec{f}_m$ to the control signal
    Step 7: Evaluate the objective function
  Step 8: Update best f_{best} and worst f_{worst} objective functions
  for j = 1:ls **do**
    Step 9: Strengthen pheromone of the best path (Equations (29) and (30))
    Step 10: Evaporate pheromone from other paths ($\Delta\tau_{i,j}(t) = 0$)
Step 11: Print tuning values

3.2. Grey Wolf Optimizer

Grey Wolf Optimizer (GWO) emulates the social leadership hierarchy and hunting strategy of grey wolves in the wild. Wolves in a pack have a strict social hierarchy, and GWO leverages this to solve optimization problems. GWO does not require prior initialization of control parameters other than population size, dimension of control variables, and maximum iterations. This method offers advantages such as avoiding local optima, requiring fewer function evaluations, and being free from the need for extensive parameter tuning. In GWO, the population is divided into four categories (alpha (α), beta (β), delta (δ), and omega (ω)), reflecting the hierarchical structure of a wolf pack. Alpha wolves lead the pack, beta wolves assist in decision-making, delta wolves follow alpha, beta wolves lead omega wolves, and omega wolves follow the other wolves. The hunting strategy involves monitoring, pursuing, and attacking the prey, which in this context translates to selecting the best parameters for FOPID tuning [25].

The circling behavior of wolves around prey is modeled using Equations (33) and (34). The vectors $\vec{A}$ and $\vec{C}$ are computed as $\vec{A} = 2 \times \text{rand} \times \vec{\gamma} - \vec{\gamma}$ and $\vec{C} = 2 \times \text{rand}$, where rand is a random vector between 0 and 1, and $\vec{\gamma}$ decreases linearly from 2 to 0 over iterations. The position vector $\vec{X}(t+1)$ is updated based on the prey's position $\vec{X}_p(t)$ and the computed distance $\vec{\delta} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|$. During hunting, alpha, beta, and delta wolves guide the search. Their positions are updated using Equation (35), where $\vec{\delta}_\alpha$, $\vec{\delta}_\beta$, and $\vec{\delta}_\delta$ represent distances from alpha, beta, and delta wolves to the prey, respectively. The new positions $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ are calculated using Equation (36), and the average position $\vec{X}(t+1)$ is obtained from Equation (37). GWO's exploration and exploitation balance is controlled by the values of $\vec{A}$ and $\vec{C}$. When $\vec{A} > 1$, exploration is favored; when $\vec{A} < 1$, exploitation is emphasized. This mechanism ensures that the GWO can effectively search the space and avoid local optima, with $\vec{A}$ and $\vec{C}$ dynamically adjusting to maintain a balance between diversification and intensification of the search process.

$$\vec{A} = (2 \times \text{rand} \times \vec{\gamma}) - \vec{\gamma}$$
$$\vec{C} = 2 \times \text{rand} \tag{33}$$

$$\vec{\delta} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|$$
$$\vec{X}(t+1) = \vec{X}_p(t) - (\vec{A} \times \vec{\delta}) \tag{34}$$

$$\vec{\delta}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right|$$
$$\vec{\delta}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \tag{35}$$
$$\vec{\delta}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right|$$

$$\vec{X}_1 = \vec{X}_\alpha - (\vec{A}_1 \times \vec{\delta}_\alpha)$$
$$\vec{X}_2 = \vec{X}_\beta - (\vec{A}_2 \times \vec{\delta}_\beta) \tag{36}$$
$$\vec{X}_3 = \vec{X}_\delta - (\vec{A}_3 \times \vec{\delta}_\delta)$$

$$\vec{X}(t+1) = \frac{1}{3}(\vec{X}_1 + \vec{X}_2 + \vec{X}_3) \tag{37}$$

The proposed "FOPID Tuning Grey Wolf Optimizer" Algorithm 2 systematically optimizes Fractional Order PID controllers by emulating the social hierarchy and hunting strategy of grey wolves. Initially, the search space and algorithm parameters are declared as shown in Table 8. The population is initialized, with each member's position assigned to the control signal, and the objective function evaluated. The algorithm iterates through several steps: identifying the three best solutions (X_α, X_β, X_δ), establishing coefficients $\vec{A}$ and $\vec{C}$ (Equation (33)), and calculating distances $\vec{\delta}_\alpha$, $\vec{\delta}_\beta$, $\vec{\delta}_\delta$ (Equation (35)). Positions are updated (Equation (36)), and the population position is adjusted (Equation (37)). The best objective function value is continuously updated, ensuring optimal tuning of the FOPID controller parameters. This methodology leverages the efficient search capabilities of GWO to enhance control performance in complex systems.

Algorithm 2 FOPID Tuning Grey Wolf Optimizer

Step 1: Declare search space and algorithm parameters (Table 8)
for p = 1:N **do**
    Step 2: Initialize population: $\vec{x}_p = lb + \text{rand} * (ub - lb)$
    Step 3: Assign x_p to control signal and evaluate objective function
Find fo_{best} among all solutions fo_p
for i = 1:T **do**
    Step 4: Identify three best solutions: X_α, X_β, X_δ
    Step 5: Establish coefficients $\vec{A}$ and $\vec{C}$ (Equation (33))
    for p = 1:N **do**
        Step 6: Calculate $\vec{\delta}_\alpha$, $\vec{\delta}_\beta$, $\vec{\delta}_\delta$ (Equation (35))
        Step 7: Find $\vec{X}_1$, $\vec{X}_2$, $\vec{X}_3$ (Equation (36))
        Step 8: Update population position (Equation (37))
        Step 9: Assign x_p to control signal
        Step 10: Evaluate objective function $\vec{fo}_p$
    Step 11: Update fo_{best} if $fo_p < fo_{best}$
Step 12: Print the optimal solution

3.3. Flower Pollination Algorithm

The fundamental concept of the Flower Pollination Algorithm 3 is inspired by the natural pollination process of flowering plants, encompassing both biotic and abiotic pollination, as well as the floral constancy exhibited by certain flower species and their pollinators. In the FPA, the position of a pollen particle is represented as a solution vector. The algorithm involves two main equations representing different pollination strategies. The global pollination mechanism, which simulates long-distance pollination, is modeled by Equation (38). Here, x_i^{t+1} denotes the new position of pollen particle i at iteration $t+1$, x_i^t is the current position, g_* is the best solution found at iteration t, γ is a scaling parameter, and $L(\lambda)$ is a step size drawn from a Lévy distribution characterized by the exponent λ. Lévy pseudorandom numbers can be generated using Mantegna's algorithm, providing a robust mechanism for global search [26].

The local pollination mechanism, which emulates the local characteristics of pollination and floral constancy, is described by Equation (39). In this equation, x_i^{t+1} represents the new position of pollen particle i at iteration $t+1$, x_i^t is its current position, U is a uniformly distributed random number, and x_j^t and x_k^t are the positions of two different pollen particles. The switching probability p is employed to decide between global and local pollination mechanisms, reflecting the natural process of abiotic and biotic pollination, as well as cross-pollination and self-pollination. Lévy flights enhance the search capability in FPA by allowing for larger steps than those typically generated by Gaussian distribution, thus improving the algorithm's ability to escape local optima. Equation (40) shows the computation of Lévy steps, where u and v are normally distributed random numbers, and σ is set to 0.7 based on an exponent β of 3/2. The parameters in FPA, such as the switching probability p, population size n, and scaling parameter γ, are crucial for its performance. A typical value of $p = 0.5$ is effective for real-world pollination processes, while γ generally ranges from 0.01 to 0.1. For implementation in simulation software, the parameters provided in Table 9 can be applied [27].

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) \tag{38}$$

$$x_i^{t+1} = x_i^t + U(\lambda)(x_j^t - x_k^t) \tag{39}$$

$$\begin{aligned} u &= \text{randn} \times \sigma \\ v &= \text{randn} \\ L &= \gamma \times \frac{u}{|v|^{1/\beta}} \end{aligned} \tag{40}$$

The "FOPID Tuning Flower Pollination Algorithm" systematically optimizes Fractional Order PID controllers by mimicking the pollination process of flowering plants. Initially, the search space and algorithm parameters are declared as shown in Table 9. The flower population is initialized, with each member's position assigned to the control signal and the objective function evaluated. The algorithm iterates by comparing random values to a switching probability p, generating step sizes using either Lévy distribution (Equation (40)) for global pollination or uniform distribution for local pollination. New solutions are generated (Equations (38) and (39)), and the best current solution is continuously updated. This approach enhances the exploration and exploitation capabilities, ensuring optimal tuning of the FOPID controller parameters [28,29].

Algorithm 3 FOPID Tuning Flower Pollination Algorithm

Step 1: Declare search space and algorithm parameters (Table 9)
for i = 1:N **do**
    Step 2: Initialize flower population: $x_i^t = lb + \text{rand} * (ub - lb)$
    Step 3: Assign x_i^t to control signal and evaluate objective function fo_p
Step 4: Identify fo_{best} and corresponding pollen value g_*
for i = 1:T **do**
    for p = 1:N **do**
        if rand $> p$ **then**
            Step 5: Generate step size L by Lévy distribution (Equation (40))
            Step 6: New solution via global pollination (Equation (38))
        else
            Step 5: Generate step size by uniform distribution $\epsilon = \text{rand} * c$
            Step 6: New solution via local pollination (Equation (39))
        Step 7: Assign solution to control signal
        Step 8: Evaluate objective function
        Step 9: Update best current solution g_*
Step 10: Print the optimal solution

4. Experimental Results

The experimental setup for the simulation of the interconnected tank system is detailed in Tables 6–9. Table 6 outlines the system parameters, including the tank area (A), interconnection cross-sectional area (A_{12}), drain pipe area (A_2), gravity (g), inflow rate (K_1), and valve gain/opening (a). These parameters are critical for establishing the equilibrium point of the interconnected tank system. Table 7 specifies the values used in the Ant Colony Optimization (ACO) algorithm, such as the number of ants (N), vaporization factor (ρ), scaling parameter (ζ), number of iterations (T), step size (h), and the search space for the PID controller gains and fractional orders. Table 8 presents the parameters for the Grey Wolf Optimizer (GWO), detailing the population size (N), number of iterations (T), and search space for the controller parameters. Finally, Table 9 lists the parameters for the Flower Pollination Algorithm (FPA), including population size (N), number of iterations (T), coefficient (c), probability of change (p), scale parameter (γ), and search space for the controller gains and fractional orders. These tables collectively provide a comprehensive overview of the experimental conditions and algorithm-specific settings used in the optimization of the FOPID controller.

Table 6. System parameters.

Symbol	Magnitude	Description
A	200 cm^2	tank area
A_{12}	0.6 cm^2	interconnection cross-sectional area
A_2	0.25 cm^2	drain pipe area
g	981 $\frac{cm}{s^2}$	gravity
K_1	100 $\frac{cm^3}{s}$	inflow
a	0.08	valve gain/opening

The computational experiments conducted in Matlab assessed the performance of fractional PID controllers using three different meta-heuristic algorithms: ACO, GWO, and FPA. Each algorithm's effectiveness was evaluated based on the performance indices used as cost functions. The system response characteristics for the ACO algorithm (Table 10) reveal that the ITAE index exhibited the worst overall performance, with an overshoot of 48.5889, an establishment time of 4.0736, and a rise time of 0.29. In contrast, the ISE index produced the best overshoot performance at 9.077, although the establishment and rise times were superior using the ITSE index, with values of 0.8282 and 0.1232, respectively.

The ISE and ITSE indices performed comparably, but due to the lower overshoot, ISE is preferred as the cost function.

Table 7. ACO values.

Parameters	Symbol	ACO
Number of ants	N	10
vaporization factor	ρ	0.6
Scaling parameter	ζ	2
number of iterations	T	10
step size	h	5
search space	Kp, Ki, Kd [lb-ub]	[0–100]
search space	λ, μ [lb-ub]	[0–5]

Table 8. GWO values.

Parameters	Symbol	GWO
Population size	N	60
number of iterations	T	300
search space	Kp, Ki, Kd [lb-ub]	[0–100]
search space	λ, μ [lb-ub]	[0–5]

Table 9. FPA values.

Parameters	Symbol	FPA
Population size	N	200
number of iterations	T	30
coefficient	c	0.5
probability of change	p	0.5
scale parameter	γ	0.1
search space	Kp, Ki, Kd [lb-ub]	[0–100]
search space	λ, μ [lb-ub]	[0–5]

Table 10. ACO algorithm results.

Performance Index	Overshoot	Establishment Time	Rise Time
ITAE	48.5889	4.0736	0.29
ISE	9.077	1.0616	0.128
ITSE	13.1782	0.8282	0.1232
IAE	11.659	1.7917	0.1768

Examining the GWO algorithm results (Table 11), the ITAE index stood out with the best overall performance, achieving an overshoot of 10.0831, an establishment time of 1.2592, and a rise time of 0.1229. Although the ISE index showed a better overshoot of 8.6316, it resulted in the worst establishment time of 5.1937. The ITAE index provided a balanced performance with acceptable overshoot and superior establishment time, making it the optimal choice for the cost function.

Table 11. GWO algorithm results.

Performance Index	Overshoot	Establishment Time	Rise Time
ITAE	10.0831	1.2592	0.1229
ISE	8.6316	5.1937	0.2396
ITSE	14.0986	2.0887	0.1727
IAE	10.0657	4.7645	0.2896

For the FPA algorithm (Table 12), the ISE index again demonstrated the best performance, with an overshoot of 9.7737, an establishment time of 1.0704, and a rise time of 0.1514. The ITAE and IAE indices showed higher overshoot values and longer establishment times, while the ITSE index had a lower establishment time but a higher overshoot. Thus, the ISE index emerged as the most effective cost function for the FPA algorithm, ensuring a lower overshoot and reasonable establishment time.

Table 12. FPA algorithm results.

Performance Index	Overshoot	Establishment Time	Rise Time
ITAE	36.2235	1.5567	0.1534
ISE	9.7737	1.0704	0.1514
ITSE	10.8043	2.0958	0.1653
IAE	26.3363	2.7784	0.213

Overall, the results indicate that the ISE index generally performs better for both ACO and FPA algorithms, while the ITAE index is preferable for the GWO algorithm due to its balanced performance across different metrics. These findings highlight the importance of selecting appropriate performance indices as cost functions to optimize the tuning of fractional PID controllers using meta-heuristic algorithms.

Using the IAE performance index as a cost function proved to be the least effective across all tuning algorithms. Figure 3 depicts the comparative results for the best-performing algorithms. The bar graph illustrates the overshoot, settling time, and rise time for the best results obtained by the ACO, GWO, and FPA algorithms. For the ACO algorithm, the overshoot was 10.6165%, the settling time was 9.07704 s, and the rise time was 1.27984 s. The GWO algorithm demonstrated an overshoot of 10.0831%, a significantly higher settling time of 12.5917 s, and a rise time of 1.22887 s. In comparison, the FPA algorithm achieved an overshoot of 10.7044%, a settling time of 9.77365 s, and a rise time of 1.51424 s. These results highlight the distinct performance characteristics of each algorithm, with the ACO algorithm showing a balance between overshoot and settling time, the GWO algorithm excelling in minimizing overshoot at the expense of a longer settling time, and the FPA algorithm providing a moderate performance across all metrics. This comparative analysis underscores the importance of selecting the appropriate algorithm based on the specific performance criteria required for optimal FOPID tuning.

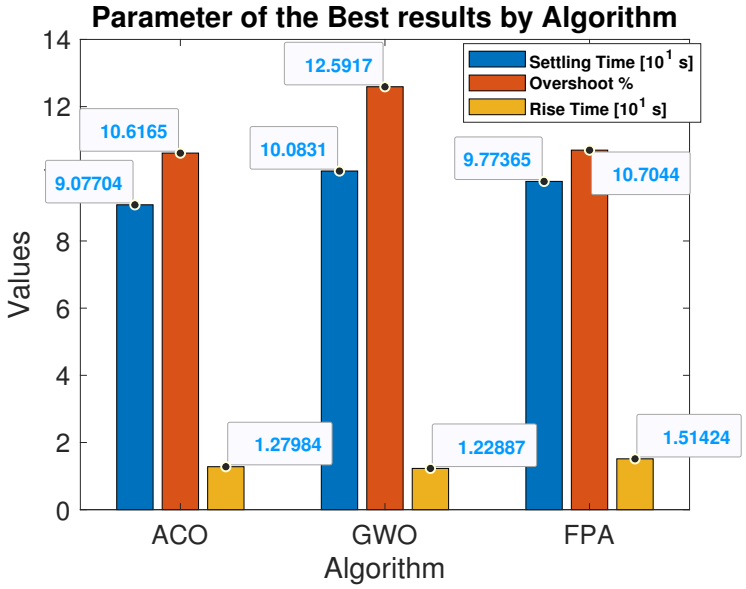

Figure 3. Parameters of the system response of each algorithm with its best index.

The comparative results indicate that the tuning algorithm with the highest performance indices was the GWO algorithm with an ITAE cost function, achieving superior settling time and rise time metrics. In contrast, the ACO and FPA algorithms, which employed an ISE cost function, also delivered commendable results, particularly in overshoot and overall stability. Figure 4 illustrates the step response curves for each algorithm with their respective best cost functions. The graph demonstrates that the GWO algorithm provides a smoother response with minimal overshoot and rapid settling time. The ACO algorithm shows a slightly higher overshoot but maintains a quick rise time, while the FPA algorithm balances overshoot and settling time. These step response curves highlight the effectiveness of each optimization algorithm in tuning the FOPID controller, with the GWO algorithm excelling in minimizing transient response characteristics.

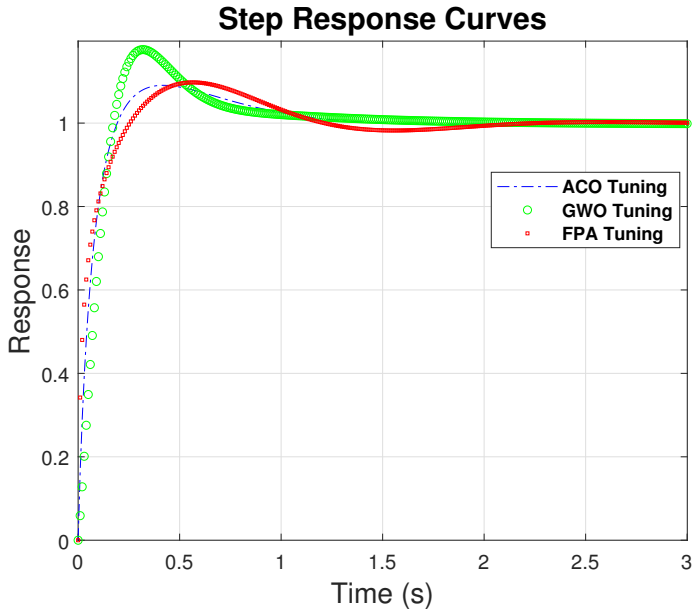

Figure 4. Step response curves of tuned systems with FOPID control.

The response curves indicate that the system exhibits characteristics of an underdamped response without multiple oscillations. Upon closer examination, it is evident that the GWO tuning results in a higher overshoot, whereas the establishment time remains relatively consistent across all algorithms. Table 13 presents the tuning results, detailing the integral (λ) and differential (μ) orders along with the proportional (K_p), integral (K_i), and derivative (K_d) constants for each algorithm. The ACO and FPA algorithms utilized the ISE cost function, while the GWO algorithm employed the ITAE cost function. These results underscore the specific tuning parameters that yielded the optimal performance for each respective algorithm.

Table 13. Results of tuned values.

Algorithm	Cost Function	K_p	K_i	K_d	λ	μ
ACO	ISE	95	20	35	0.3	1.1
GWO	ITAE	50	31.8391	50	0.809	0.7729
FPA	ISE	73.3803	83.971	23.4439	2.4309×10^{-6}	1.4011

The bar graph in Figure 5 demonstrates that the parameters of the system response using the Grey Wolf Optimizer (GWO) algorithm tend to have greater overshoot and a comparatively longer settling time than the ACO and FPA algorithms. Specifically, the GWO algorithm resulted in an overshoot of 33.2651%, a settling time of 10.7198 s, and a rise time of 1.79475 s. In contrast, the ACO algorithm achieved an overshoot of 19.3879%, a

settling time of 20.6258 s, and a rise time of 1.79475 s. The FPA algorithm displayed similar performance to ACO, with an overshoot of 18.7534%, a settling time of 20.7844 s, and a rise time of 1.70798 s. These results indicate that while the GWO algorithm tends to produce higher overshoot, it achieves better settling times than ACO and FPA. The ACO and FPA algorithms show similar values for overshoot, rise time, and settling time, highlighting their comparable performance in tuning the system response.

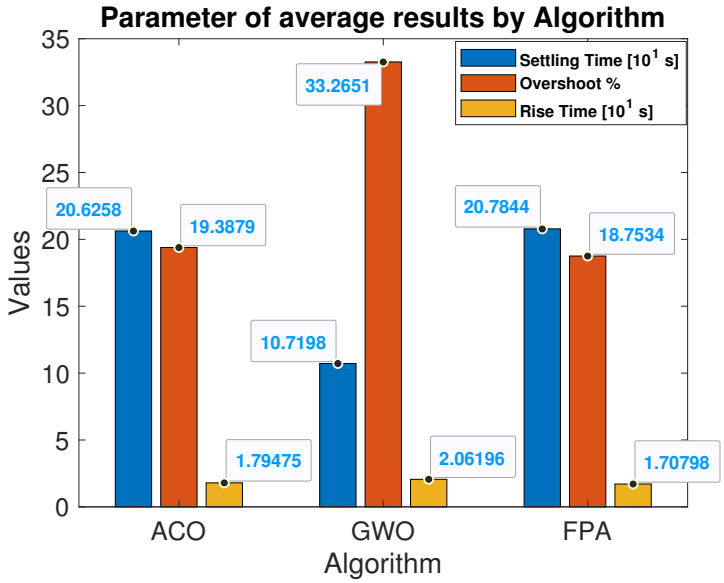

Figure 5. Parameters of the system response with the average result of each algorithm.

Figure 6 presents the values of the ITAE and ISE indices, highlighting the best results achieved by each algorithm. It is evident that the indices for the GWO algorithm are significantly higher compared to those for the ACO and FPA algorithms. Specifically, the ITAE index value for the ACO algorithm is lower than that of the FPA algorithm, indicating a more efficient performance in terms of settling time and rise time. Conversely, the ISE value for the ACO algorithm is higher, reflecting a higher overshoot as shown in Table 10, where the ACO algorithm using the ITAE cost function resulted in an overshoot of 48.5889%. In comparison, the FPA algorithm, as seen in Table 12, achieved a lower overshoot of 36.2235% with the ITAE cost function.

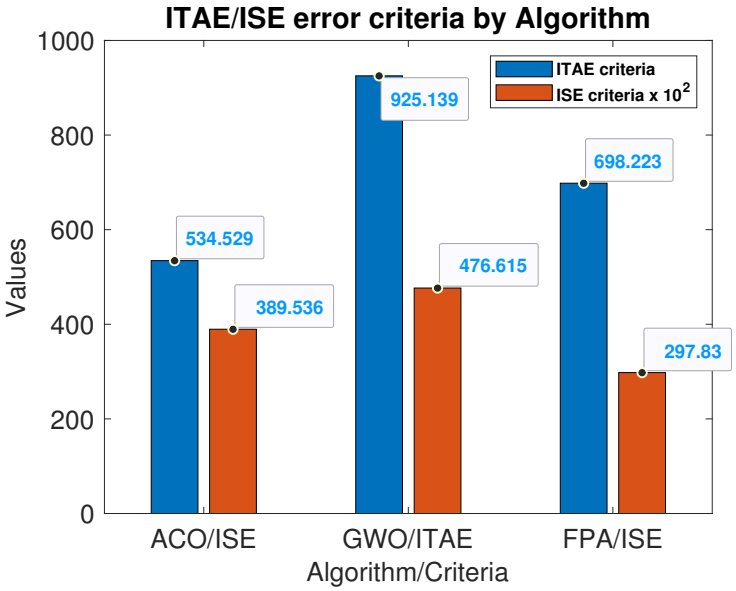

Figure 6. Comparison of ITAE/ISE indices of the best results by meta-heuristic algorithm.

Moreover, examining the ISE values, which are the indices with the best results for both the ACO and FPA algorithms, it can be observed in Figure 6 that the ISE value for the FPA algorithm (297.83) is lower than that of the ACO algorithm (389.536), indicating better overall performance in minimizing the integral of the squared error. These results underscore the effectiveness of the FPA algorithm in achieving lower error indices, thereby enhancing the control system's stability and performance.

Figure 7 illustrates the values of the respective cost functions for each algorithm as they iterate to find the optimal FOPID constants (K_p, K_i, K_d) of the control signal. The graph shows that the ACO algorithm's cost function values are consistently higher than those of the GWO and FPA algorithms. The convergence behavior of the algorithms is also noteworthy. The ACO algorithm takes four iterations to converge to its optimal solution, while the FPA algorithm requires seven iterations, and the GWO algorithm converges the fastest at three iterations. This indicates that while the ACO algorithm has higher cost function values, it converges relatively quickly, similar to the GWO algorithm, which combines lower cost values with the fastest convergence. Though the FPA algorithm has lower cost values, it takes more iterations to reach its optimal solution. These observations highlight the trade-offs between convergence speed and cost function values among the different algorithms.

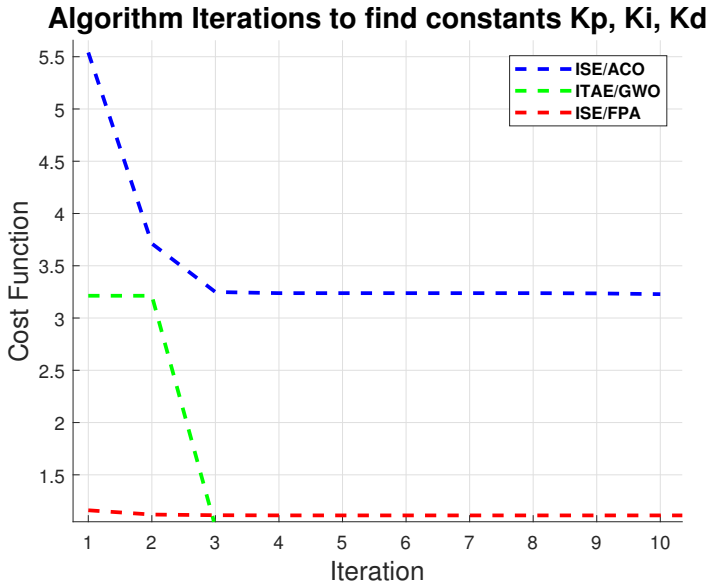

Figure 7. Convergence of cost functions for ACO, GWO, and FPA algorithms.

Figure 8 presents the values of the respective cost functions for each algorithm as they iterate to find the optimal fractional orders (λ, μ) of the FOPID control signal. The graph shows that the cost function values for the ACO algorithm are consistently higher compared to those for the GWO and FPA algorithms. The convergence behavior of the algorithms is also noteworthy. The ACO algorithm requires seven iterations to converge to its optimal solution, while both the FPA and GWO algorithms converge more quickly, each taking three iterations. This indicates that although the ACO algorithm has higher cost function values and requires more iterations to converge, the GWO and FPA algorithms achieve lower cost values and faster convergence. These observations highlight the efficiency of the GWO and FPA algorithms in optimizing the fractional orders of the FOPID controller.

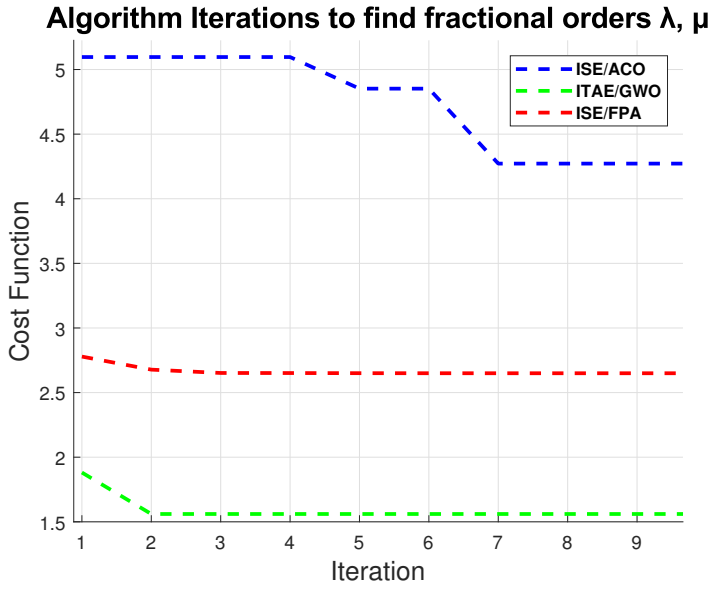

Figure 8. Algorithm iterations to find fractional orders λ, μ.

5. Conclusions and Future Works

Nonlinear equations effectively represented the proposed nonlinear model of the interconnected tank system. To facilitate control using a fractional PID controller, it was necessary to linearize these equations around an equilibrium point. This was accomplished by developing the Taylor Series and applying the Laplace transform, resulting in a transfer function suitable for fractional PID control. The linearization process enabled the accurate modeling and subsequent control of the system's dynamics, ensuring that the complex behavior of the nonlinear system could be managed effectively using advanced control techniques.

The implementation of fractional PID control for the system's transfer function involved tuning the controller constants and fractional orders using meta-heuristic algorithms: Ant Colony Optimization (ACO), Gray Wolf Optimizer (GWO), and Flower Pollination Algorithm (FPA). Each algorithm was applied with different performance error criteria. The system's responses to a step input were simulated on the Matlab platform, demonstrating the feasibility and robustness of these algorithms in tuning the FOPID controllers. This approach allowed for the precise adjustment of the controller parameters to achieve desired performance characteristics in a nonlinear environment.

The system's transient response to step inputs was thoroughly analyzed for each tuning strategy and performance index. Key metrics such as overshoot, settling time, and rise time were extracted from the simulation results. For instance, the GWO algorithm resulted in an overshoot of 10.0831% and a settling time of 5.1937 s, while the ACO and FPA algorithms exhibited lower overshoots and shorter settling times. Additionally, the system's ability to reject disturbances was evaluated, revealing that the ACO and FPA algorithms provided superior disturbance rejection capabilities, maintaining system stability and minimizing deviation from the setpoint more effectively than GWO. These response curves provided a comprehensive understanding of how each tuning method influenced the system's dynamic behavior, highlighting the strengths and weaknesses of each algorithm in achieving optimal control performance and robust disturbance rejection.

A comparative validation using performance metrics and graphical analysis of the output responses established the efficiency of each tuning method. The GWO algorithm showed higher overshoot and settling times than ACO and FPA. Specifically, the FPA algorithm demonstrated superior performance with an ISE criterion value of 297.83, lower than the ACO's 389.536. Furthermore, the FPA algorithm required fewer iterations to converge to the optimal proportional, integral, and derivative constants (three iterations for constants and four iterations for fractional orders) compared to the ACO algorithm

(four iterations for constants and eight iterations for fractional orders). These findings underscore the FPA algorithm's effectiveness in achieving faster convergence and better performance, making it a preferable choice for tuning fractional PID controllers in nonlinear systems.

Author Contributions: Conceptualization: R.P. and S.S.; Methodology: W.P. and M.A.; Software: R.P. and W.P.; Validation: S.S. and M.A.; Formal Analysis: R.P.; Investigation: W.P.; Resources: M.A.; Data Curation: W.P.; Writing—Original Draft Preparation: R.P. and W.P.; Writing—Review and Editing: M.A. and S.S.; Visualization: W.P.; Supervision: R.P. and S.S.; Project Administration: W.P.; Funding Acquisition: W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universidad Politécnica Salesiana (UPS) also supports the Master's program in Electronics and Automation and covers the costs associated with publishing in Algorithms Journals. Additionally, UPS funded my Ph.D. studies at Ferrara University. The original material for the connected tank system was provided by Newcastle University (NCL) during my Master's studies. My Master's studies in the UK were financially supported by the Ecuadorian government.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This paper presents results from a project at Universidad Politecnica Salesiana (UPS). The model for the connected tank was developed during a course on industrial process control by M. Armstrong at Newcastle University as part of the master's program in Automation and Control in 2016. S. Simani, the Ph.D. supervisor at Ferrara University, contributed to heuristic methodologies for system identification. Ing. R. Pazmiño is a master's student in electronics and automation at UPS, with W. Pavon supervising his graduation project. UPS has provided financial support for the APC for Algorithms journal and offers an encouraging research environment. All individuals mentioned have consented to be acknowledged.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohankumar, R.; Selvaganesan, N.; Jayakumar, M.; Sathishkumar, P. Heuristic algorithms based optimal tuning of FOLQI controller for quadruple tank process under disturbance conditions. *Meas. Control* **2024**, *57*, 164–186. [CrossRef]
2. Roy, P.; Roy, B. Dual mode adaptive fractional order PI controller with feedforward controller based on variable parameter model for quadruple tank process. *ISA Trans.* **2016**, *63*, 365–376. [CrossRef]
3. Choudhari, P.; Kulkarni, N.; Bakshi, M. A System Theoretic-Based Optimum Controller for Single-Tank System and Its Performance Comparison with PID Controller. *J. Inst. Eng. India Ser. B* **2023**, *104*, 551–561. [CrossRef]
4. Cama, J.; Alegria, E.; Narvaez, D. Genetic-Algorithm-Based Tuning of PID Controllers for a Multipurpose Water Tank Plant. In Proceedings of the 2023 IEEE XXX International Conference On Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2–4 November 2023; pp. 1–7. [CrossRef]
5. Bhandare, D.; Kulkarni, N.; Bakshi, M. An intelligent self-tuning fuzzy-PID controller to coupled tank liquid level system. *Int. J. Inf. Technol.* **2022**, *14*, 1747–1754. [CrossRef]
6. Amuthambigaiyin Sundari, K.; Maruthupandi, P. Optimal design of PID controller for the analysis of Two TANK system using metaheuristic optimization algorithm. *J. Electr. Eng. Technol.* **2022**, *17*, 627–640. [CrossRef]
7. Chauhan, S.; Singh, B.; Singh, M. Modified ant colony optimization based PID controller design for coupled tank system. *Eng. Res. Express* **2021**, *3*, 045005. [CrossRef]
8. Şahin, A.; Taş, T.; Bertuğ, E.; Ayas, M. Metaheuristic algorithm based PI controller design for Linearized Quadruple-Tank Process. In Proceedings of the 2021 3rd International Congress On Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–6. [CrossRef]
9. Pugazhenthi, P.N.; Selvaperumal, S.; Vijayakumar, K. Others Nonlinear PID controller parameter optimization using modified hybrid artificial bee colony algorithm for continuous stirred tank reactor. *Bull. Pol. Acad. Sci. Tech. Sci.* **2021**, *69*, e137348. [CrossRef]
10. Kumar, J. Design and Analysis of Nonlinear PID Controller for Complex Surge Tank System. In *International Conference on Communication and Artificial Intelligence: ICCAI 2020*; Springer: Singapore, 2021; pp. 189–199. [CrossRef]
11. Jaiswal, S.; Suresh Kumar, C.; Seepana, M.; Babu, G. Design of fractional order PID controller using genetic algorithm optimization technique for nonlinear system. *Chem. Prod. Process Model.* **2020**, *15*, 20190072. [CrossRef]
12. Febina, C.; Vijula, D. Model based controller design using real time neural network model and PSO for conical tank system. *J. Control Eng. Appl. Inform.* **2020**, *22*, 13–24.

13. Geibel, P.; Wysotzki, F. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.* **2005**, *24*, 81–108. [CrossRef]
14. López-Ibáñez, M.; Prasad, T.; Paechter, B. Ant colony optimization for optimal control of pumps in water distribution networks. *J. Water Resour. Plan. Manag.* **2008**, *134*, 337–346. [CrossRef]
15. Li, J.; Leung, S.; Wu, Y.; Liu, K. Allocation of empty containers between multi-ports. *Eur. J. Oper. Res.* **2007**, *182*, 400–412. [CrossRef]
16. Sharafi, M.; ElMekkawy, T.; Bibeau, E. Optimal design of hybrid renewable energy systems in buildings with low to high renewable energy ratio. *Renew. Energy* **2015**, *83*, 1026–1042. [CrossRef]
17. Duan, N.; Mays, L.; Lansey, K. Optimal reliability-based design of pumping and distribution systems. *J. Hydraul. Eng.* **1990**, *116*, 249–268. [CrossRef]
18. Ronen, D. Marine inventory routing: Shipments planning. *J. Oper. Res. Soc.* **2002**, *53*, 108–114. [CrossRef]
19. Braun, M.; Rivera, D.; Stenman, A. A'Model-on-Demand'identification methodology for non-linear process systems. *Int.J. Control* **2001**, *74*, 1708–1717. [CrossRef]
20. Ruelens, F.; Claessens, B.; Vandael, S.; Iacovella, S.; Vingerhoets, P.; Belmans, R. Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. In Proceedings of the 2014 Power Systems Computation Conference, Wroclaw, Poland, 18–22 August 2014; pp. 1–7. [CrossRef]
21. Risbeck, M.; Maravelias, C.; Rawlings, J.; Turney, R. A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment. *Energy Build.* **2017**, *142*, 220–235. [CrossRef]
22. Drees, K.; Braun, J. Development and evaluation of a rule-based control strategy for ice storage systems. *HVAC R Res.* **1996**, *2*, 312–334. [CrossRef]
23. Abusini, S.; Sukmarini, M.; Karim, C. Ant colony optimization with double selections for solving integrated scheduling problem in manufacturer. *JEMIS J. Eng. Manag. Ind. Syst.* **2019**, *7*, 25–34. [CrossRef]
24. Alam, J.; Hu, G.; Babu, H.; Xu, H. *Control Engineering Theory and Applications*; CRC Press: Boca Raton, FL, USA, 2022. [CrossRef]
25. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [CrossRef]
26. Rajesh, R. Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system. *SN Appl. Sci.* **2019**, *1*, 758. [CrossRef]
27. Büchi, R. Parameter tables for PID controllers for time delayed systems optimized with a learning method. In *Langbeiträge ASIM SST 2022, 26. Symposium Simulationstechnik*; TU Wien: Vienna, Austria, 2022; Volume 20, pp. 179–185. [CrossRef]
28. Ong, K.; Ong, P.; Sia, C. A new flower pollination algorithm with improved convergence and its application to engineering optimization. *Decis. Anal. J.* **2022**, *5*, 100144. [CrossRef]
29. Pinzón, S.; Pavón, W. Diseño de sistemas de control basados en el análisis del dominio en frecuencia. *Rev. Técnica Energía* **2019**, *15*, 76–82. [CrossRef]