

Article

Not peer-reviewed version

Hybrid Particle Swarm and Grey Wolf Optimization for Robust Feedback Control of Nonlinear Systems

[Robert Vrabel](#) *

Posted Date: 14 November 2025

doi: 10.20944/preprints202511.1095.v1

Keywords: metaheuristic optimization; PID tuning; nonlinear control; hybrid algorithms



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Hybrid Particle Swarm and Grey Wolf Optimization for Robust Feedback Control of Nonlinear Systems

Robert Vrabel 

Institute of Applied Informatics, Automation and Mechatronics, Slovak University of Technology in Bratislava, Bottova 25, 917 24 Trnava, Slovakia; robert.vrabel@stuba.sk

Abstract

This study presents a simulation-based framework for PID controller design in strongly nonlinear dynamical systems. The proposed approach avoids system linearization by directly minimizing a performance index using metaheuristic optimization. Three strategies—Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and their hybrid combination (PSO–GWO)—were evaluated on benchmark systems including pendulum-like, Duffing-type, and nonlinear damping dynamics. The chaotic Duffing oscillator was used as a stringent test for robustness and adaptability. Results indicate that all methods successfully stabilize the systems, while the hybrid PSO–GWO achieves the fastest convergence and requires the fewest cost function evaluations, often less than 10% of standalone methods. Faster convergence may induce aggressive transients, which can be moderated by tuning the ISO (Integral of Squared Overshoot) weighting. Overall, swarm-based PID tuning proves effective and computationally efficient for nonlinear control, offering a robust trade-off between convergence speed, control performance, and algorithmic simplicity.

Keywords: metaheuristic optimization; PID tuning; nonlinear control; hybrid algorithms

1. Introduction

The Proportional–Integral–Derivative (PID) controller remains one of the most fundamental and widely adopted strategies in industrial automation, process control, and mechatronic systems. Its enduring popularity is attributed to its conceptual simplicity, robustness, and ease of implementation across diverse application domains. Despite its ubiquity, achieving optimal performance through appropriate tuning of the PID gains (K_p , K_i , K_d) continues to be a challenging and active area of research. This challenge becomes particularly pronounced in nonlinear or time-varying systems, where conventional tuning rules derived from linearized models—such as the Ziegler–Nichols or Cohen–Coon methods—often result in degraded transient behavior, excessive overshoot, or even closed-loop instability. In safety-critical or high-precision industrial processes, such overshoot or oscillatory responses are often unacceptable, motivating the need for more advanced and adaptive tuning methodologies. The monograph by Åström and Hägglund [1] provides a detailed discussion of PID controller design and optimization for various types of systems, including nonlinear ones, with practical insights into tuning and implementation. Similarly, Nise [2] offers a comprehensive foundation for linear and nonlinear control design, while Ogata [3] presents an in-depth treatment of control system analysis and design principles encompassing both linear and nonlinear dynamics. A recent and comprehensive review of modern approaches to PID controller design, including nonlinear extensions, can be found in [4].

In recent decades, metaheuristic and evolutionary optimization techniques have emerged as powerful and flexible alternatives to traditional analytical tuning methods. These algorithms are particularly well-suited for solving highly nonlinear, nonconvex, and multimodal optimization problems without requiring gradient information or model simplification. A comprehensive survey of metaheuristic optimization algorithms, including 47 nature-inspired methods designed for load balancing

in cloud computing, is presented in [5]. In contrast, the present work focuses on applying metaheuristic optimization techniques to parameter tuning in nonlinear dynamical systems, emphasizing their ability to balance convergence efficiency, robustness, and computational cost.

Among metaheuristic approaches, *Particle Swarm Optimization* (PSO) has demonstrated notable success due to its efficient local exploitation of promising regions. Originally developed by Kennedy and Eberhart [6–8], PSO draws inspiration from the collective movement of bird flocks and fish schools, iteratively refining candidate solutions.

The *Grey Wolf Optimizer* (GWO) [9], inspired by grey wolves' social hierarchy and cooperative hunting, excels in global exploration and avoids local minima. Its simplicity and balanced exploration-exploitation behavior make it effective across diverse optimization tasks.

Hybrid strategies combining PSO and GWO leverage the strengths of both methods: PSO refines solutions locally, while GWO ensures thorough global search. Recent applications demonstrate their effectiveness, such as optimal reactive power dispatch [10] and smart grid reconfiguration [11], where the hybrid approach achieved reduced losses, stable voltage, and robust performance in high-dimensional, nonconvex scenarios.

In distributed generation planning, hybrid PSO–GWO strategies have demonstrated superior performance. Alyu [12] employed a hybrid GWO–PSO algorithm to determine the optimal placement and sizing of multiple photovoltaic distributed generators (PV-DGs), achieving reduced active and reactive power losses and improved voltage profiles compared to other metaheuristics such as WOA, SCA, and standalone PSO or GWO. Similarly, Aguila-Lopez [13] integrated a GWO-enhanced PSO into a Maximum Power Point Tracking (MPPT) controller, achieving up to 20% gains in energy extraction efficiency under varying irradiation, highlighting the hybrid algorithm's adaptability and robustness.

The effectiveness of PSO–GWO hybridization extends to broader optimization contexts. For example, Bhandari *et al.* [14] applied it to reliability-redundancy optimization, showing that combining PSO's exploitation with GWO's exploration yields better solutions with fewer iterations. Likewise, Senel [15] confirmed the hybrid method's superiority over standard and hybrid approaches on benchmark functions and real-world industrial problems, including process optimization and material nesting.

It is possible to find several publications that apply metaheuristic algorithms for the optimization of PID control systems. For instance, in [16], a hybrid PSO–GWO optimization algorithm was proposed for tuning a PID controller in an Automatic Voltage Regulator (AVR) system. The main objective of the study was to improve the transient response of the AVR by minimizing key performance indices such as rise time, settling time, peak overshoot, and peak time. Comparative analyses with other heuristic-based tuning methods available in the literature demonstrated that the proposed hybrid PSO–GWO approach provided superior performance and enhanced dynamic behavior. However, the considered system in this case is a linear time-invariant (LTI) process, where the underlying dynamics are relatively simple and predictable, and thus do not capture the complexities encountered in nonlinear systems.

A similar observation can be made in [17], which evaluates the performance and robustness of a metaheuristic-based PID controller for a First-Order with Time Delay (FOPTD) system. Like the AVR case, the FOPTD process represents a linear model analyzed through a transfer function framework, which is a well-established and elegant tool for LTI system analysis. Nevertheless, such methods cannot be directly applied to nonlinear systems, where the system behavior depends on the instantaneous state variables and the dynamic equations exhibit strong nonlinear coupling. Consequently, the use of transfer function-based approaches or linear metaheuristic tuning schemes is inherently limited in such contexts.

Despite these advances across diverse application areas—ranging from power systems and renewable energy to reliability engineering and industrial optimization—none of the aforementioned studies have addressed the problem of **metaheuristic-based PID tuning for nonlinear dynamical systems**. This gap is particularly significant, as strong nonlinearities, such as those present in pendulum-like or

Duffing oscillators, introduce severe challenges for controller design and stability assurance. Consequently, extending hybrid metaheuristic frameworks such as PSO–GWO to the automatic tuning of PID controllers in nonlinear systems represents a novel and impactful research direction, which is the main focus of the present study.

In this work, three metaheuristic-based PID tuning strategies are systematically investigated for nonlinear control systems:

1. **PSO-based PID tuning**, leveraging swarm intelligence for global parameter optimization;
2. **GWO-based PID tuning**, employing hierarchical search to balance exploration and exploitation; and
3. **Hybrid PSO–GWO tuning**, integrating both paradigms to accelerate convergence and enhance robustness under strong nonlinearities.

The proposed optimization framework is evaluated on representative second-order nonlinear benchmark systems, including pendulum-type, Duffing-type, and nonlinear damping models. The optimization objective combines integral-based performance indices—such as the Integral of Time-weighted Absolute Error (ITAE) and the Integral of Squared Overshoot (ISO)—to capture both transient accuracy and overshoot suppression. Particular attention is given to the influence of the ISO weighting coefficient α , which provides an additional degree of freedom to regulate the trade-off between convergence speed and control smoothness.

Simulation results demonstrate that all three optimization methods effectively stabilize the considered nonlinear systems. While the hybrid PSO–GWO consistently achieves the lowest number of cost function evaluations, it may exhibit more aggressive transient dynamics, which can be mitigated by appropriate tuning of α . These findings highlight the potential of metaheuristic-based PID tuning—especially in hybrid form—as a powerful and computationally efficient tool for nonlinear control system design.

The remainder of this paper is organized as follows. Section 2 presents the mathematical formulation of the nonlinear systems and the associated control structure. Section 3 describes the proposed optimization framework for PID tuning, including PSO, GWO, and hybrid PSO–GWO strategies. Section 4 introduces the benchmark nonlinear systems, evaluates the performance of the proposed approaches, and discusses the obtained results. A comparative performance analysis is provided in Section 5. In particular, Subsection 5.2 presents a robustness evaluation in which the hybrid PSO–GWO optimization scheme is applied to the Duffing oscillator under parameter conditions that induce chaotic dynamics. This challenging case highlights the algorithm’s capability to stabilize strongly nonlinear and aperiodic systems, demonstrating its robustness and adaptability in one of the most demanding control scenarios. Finally, Section 6 summarizes the key findings and outlines directions for future research.

2. Problem Formulation

Consider a general n th-order nonlinear dynamic system defined as

$$y^{(n)}(t) + f(t, y(t), y'(t), \dots, y^{(n-1)}(t)) = u(t), \quad (1)$$

over the interval $[0, \infty)$, with zero initial conditions

$$y(0) = y'(0) = \dots = y^{(n-1)}(0) = 0. \quad (2)$$

In the feedback control configuration shown in Figure 1, $u(t)$ represents the control signal generated by the PID controller and applied to the system input, while $y(t)$ denotes the process (or output) variable whose behavior is being regulated. Both $u(t)$ and $y(t)$ are assumed to be scalar functions defined on the interval $[0, \infty)$ and to possess the required degree of continuous differentiability on this interval.

The PID controller is expressed as

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

where $e(t) = r(t) - y(t)$ is the tracking error between the reference input $r(t)$ and the system output $y(t)$.

By analyzing the closed-loop signal flow in Figure 1, the dynamics of the n th-order nonlinear system under PID control can be written as

$$y^{(n)}(t) + f(t, y(t), y'(t), \dots, y^{(n-1)}(t)) = u(t), \quad (4)$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (5)$$

$$e(t) = r(t) - y(t). \quad (6)$$

For numerical implementation and analysis, the system can be equivalently represented in an augmented first-order state-space form using the state vector

$$[Y_0, Y_1, \dots, Y_{n-1}, E_{\text{int}}] = \left[y, y', \dots, y^{(n-1)}, \int_0^t e(\tau) d\tau \right], \quad (7)$$

which evolves according to

$$Y_i'(t) = Y_{i+1}(t), \quad i = 0, 1, \dots, n-2, \quad (8)$$

$$\begin{aligned} Y_{n-1}'(t) &= u(t) - f(t, Y_0(t), Y_1(t), \dots, Y_{n-1}(t)) \\ &= K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \\ &\quad - f(t, Y_0(t), Y_1(t), \dots, Y_{n-1}(t)), \end{aligned} \quad (9)$$

$$E_{\text{int}}'(t) = e(t) = r(t) - Y_0(t). \quad (10)$$

In this study, we focus exclusively on second-order systems ($n = 2$) with a unit-step reference input, defined as

$$r(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0. \end{cases} \quad (11)$$

Second-order systems exhibit more complex transient behavior than first-order systems, including overshoot, oscillations, and slower settling times, providing a richer testbed for evaluating the effectiveness of different PID tuning strategies. The simulation setup examines the system's transient response when the input undergoes a step change from zero to unity, allowing direct comparison of various optimization-based tuning methods under realistic nonlinear dynamics.

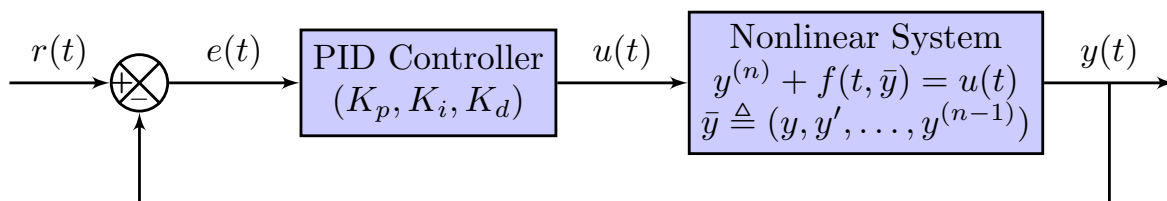


Figure 1. Schematic representation of the feedback control loop with reference input $r(t)$, controlled output $y(t)$, and control signal $u(t)$ generated by a PID controller.

3. Optimization Framework

The objective of PID tuning is to determine the optimal controller gains K_p , K_i , and K_d such that the closed-loop system achieves the desired transient performance while maintaining stability and minimizing overshoot. In essence, the PID controller continuously adjusts its control action to keep the actual process output $y(t)$ as close as possible to the reference signal $r(t)$. In this paper, the reference input is defined as a unit-step function. The unit-step signal is commonly employed in control system analysis and simulation because it effectively excites the system dynamics and provides a clear characterization of transient and steady-state behavior. Its simplicity allows for straightforward evaluation of key performance indicators such as rise time, settling time, and overshoot, which are critical for assessing control quality. Consequently, the step response serves as a standard benchmark for tuning and comparing PID control strategies across both linear and nonlinear systems.

To quantify this objective, we adopt a composite performance index that accounts for both the speed of response and the magnitude of undesirable deviations:

$$J(K_p, K_i, K_d) = \int_0^T \left[t |e(t)| + \alpha (\max(0, -e(t)))^2 \right] dt, \quad (12)$$

where $e(t) = r(t) - y(t)$ is the tracking error between the reference input $r(t)$ and system output $y(t)$. The first term, $\text{ITAE} = \int_0^T t |e(t)| dt$, represents the *Integral of Time-weighted Absolute Error*, which penalizes errors that persist over time, thereby promoting fast settling and reducing long-lasting deviations. The second term, $\text{ISO} = \int_0^T (\max(0, -e(t)))^2 dt$, corresponds to the *Integral of Squared Overshoot*, which selectively penalizes overshoot in the system's response, limiting the peak amplitude above the desired reference during transients.

The weighting factor $\alpha > 0$ allows flexible tuning of the relative importance between minimizing prolonged error versus avoiding overshoot, providing a more nuanced evaluation of closed-loop performance. The finite time horizon T is selected to sufficiently capture both transient dynamics and steady-state behavior, ensuring that the resulting PID parameters yield a balance between rapid response, overshoot control, and long-term stability.

By employing this performance index as the objective function, metaheuristic optimization algorithms such as PSO, GWO, and hybrid PSO–GWO can systematically search the multidimensional space of K_p , K_i , and K_d to identify near-optimal solutions without requiring explicit linearization or analytical design formulas. This approach enables robust PID tuning for nonlinear systems, accommodating complex dynamical behaviors that are challenging to handle using classical methods.

Particle Swarm Optimization (PSO)

In the context of PID controller optimization, the Particle Swarm Optimization (PSO) algorithm treats each particle in the swarm as a potential solution defined by a triplet of controller gains (K_p, K_i, K_d) . During the optimization process, particles iteratively adjust their positions in the search space based on both their individual best experiences and the collective knowledge of the swarm. The objective is to minimize the predefined performance index $J(K_p, K_i, K_d)$, typically expressed as a weighted combination of integral error measures such as the ITAE and ISO criteria defined in Equation (12). Due to its velocity-based update mechanism, PSO exhibits strong local search capability (exploitation), enabling fast convergence toward promising regions of the cost landscape. However, this characteristic also increases the risk of premature convergence in multimodal or highly nonlinear problems, where sufficient exploration of alternative regions is required [9].

Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a nature-inspired metaheuristic based on the cooperative hunting behavior and hierarchical leadership structure observed in grey wolf packs. Within the context of PID controller tuning, each wolf represents a candidate set of controller gains (K_p, K_i, K_d) , while the search process is guided by three leading wolves— α , β , and δ —representing the best solutions found so far. The remaining wolves update their positions by mimicking the encircling and hunting behavior of

the leaders, gradually refining their estimates toward the global optimum. In contrast to PSO, GWO exhibits stronger exploratory behavior, preserving population diversity and systematically scanning the search space through adaptive encircling and shrinking mechanisms [9]. By effectively balancing exploration and exploitation through this hierarchical mechanism, GWO achieves stable convergence and robustness in complex, nonconvex optimization landscapes.

Hybrid Strategy: PSO–GWO

The hybrid PSO–GWO algorithm combines the exploitation strength of PSO with the exploration capability of GWO. In this hybrid framework, the PSO component provides rapid local refinement by updating particle velocities and positions, ensuring efficient convergence in promising regions. Conversely, the GWO mechanism maintains global diversity and exploration by emulating the social hierarchy and cooperative hunting strategy of grey wolves. This complementary interaction allows the hybrid algorithm to dynamically balance global exploration and local exploitation throughout the optimization process [9]. When applied to PID tuning, the PSO–GWO hybrid typically achieves faster convergence, higher solution accuracy, and improved robustness against the multimodal and nonconvex nature of the cost landscape that characterizes strongly nonlinear control problems.

In practical implementation, the population is divided into two cooperative sub-swarms:

- The first sub-swarm updates positions using PSO velocity and position rules, enabling efficient local exploitation.
- The second sub-swarm evolves according to GWO’s hierarchical encircling and hunting behavior, ensuring effective global exploration.
- At each iteration, elite and global best information are exchanged between the sub-swarms to preserve diversity and guide convergence toward the global optimum.

This cooperative structure enhances the robustness and efficiency of the hybrid optimizer, making it particularly suitable for nonlinear and dynamically complex PID controller tuning tasks.

Expected Benefits of the Hybrid PSO–GWO Approach

- **Complementary Search Dynamics:** The PSO sub-swarm exploits rapid convergence in smooth regions, while the GWO sub-swarm ensures broad exploration in complex, multimodal areas.
- **Robustness Against Local Minima:** The hybrid structure mitigates premature convergence by leveraging PSO’s memory-guided exploitation and GWO’s stochastic encircling mechanisms.
- **Enhanced Population Diversity:** Periodic exchange of elite and global best information between sub-swarms maintains diversity and prevents stagnation.
- **Improved Convergence Efficiency:** Hybridization accelerates convergence by allowing PSO to refine promising regions discovered through GWO’s exploratory search.
- **Flexibility in Multi-Objective Optimization:** The framework can be extended to handle multiple conflicting objectives, such as minimizing both ITAE and overshoot.

Overall, the hybrid PSO–GWO algorithm leverages the complementary strengths of both meta-heuristics, achieving faster, more reliable, and globally effective PID tuning for strongly nonlinear control systems.

The following section presents its implementation and performance evaluation on representative nonlinear benchmark models.

3.1. General Procedure

The overall PID tuning framework is implemented in *Python*, leveraging widely-used scientific computing and metaheuristic optimization libraries.

The general procedure for both PSO and GWO consists of the following steps:

1. **Initialization:** Define the search space for PID parameters $K_p, K_i, K_d \in [0, 10]$, generate the initial population (particles or wolves) randomly within these bounds, and initialize historical records for tracking cost evaluations.

2. **Simulation:** For each candidate solution, simulate the second-order nonlinear system using the augmented first-order form of the dynamics, integrating the ODEs with a Runge–Kutta solver (RK45) over the simulation horizon.
3. **Cost Evaluation:** Compute the performance index $J(K_p, K_i, K_d)$ as the sum of ITAE and ISO terms, storing each evaluation for convergence analysis.
4. **Position Update:**
 - In PSO, update particle velocities and positions according to the standard PSO rules, taking into account inertia, cognitive, and social components.
 - In GWO, update wolf positions using the encircling, hunting, and attacking strategies defined by the leader hierarchy.
5. **Global Best Update:** Identify the best-performing solution across the population (global best) and update personal or elite records as needed.
6. **Iteration and Convergence Check:** Repeat simulation, evaluation, and update steps for the specified number of generations or until an early stopping criterion is satisfied (e.g., cost improvement below a threshold $\delta = 10^{-5}$).
7. **Output:** The optimal PID parameters (K_p^*, K_i^*, K_d^*) , the associated cost $J(K_p^*, K_i^*, K_d^*)$, and the total number of cost function evaluations are reported to quantify the performance and computational effort of each optimization strategy.

The hybrid PSO–GWO procedure combines the strengths of both algorithms, exploiting PSO's local refinement capabilities and GWO's global exploration through hierarchical hunting. The general workflow is as follows:

1. **Initialization:** Define the search space for PID parameters $K_p, K_i, K_d \in [0, 10]$. Initialize the PSO swarm and GWO population randomly within these bounds. Set personal bests for PSO particles and the leader hierarchy for GWO (α, β, δ). Initialize historical records for cost evaluations.
2. **Simulation:** For each candidate solution (particle or wolf), simulate the second-order nonlinear system in augmented first-order form. Integrate the ODEs over the simulation horizon using a Runge–Kutta solver (RK45) to obtain the system response $y(t)$.
3. **Cost Evaluation:** Compute the performance index $J(K_p, K_i, K_d)$ as the sum of ITAE and ISO terms. Store each evaluation in a history log for convergence analysis and potential CSV export.
4. **PSO Update:** Update particle velocities and positions using the PSO formula:

$$v \leftarrow wv + c_1 r_1 (p_{\text{best}} - x) + c_2 r_2 (g_{\text{best}} - x), \quad x \leftarrow x + v$$

Clip positions to remain within bounds. Update personal and global bests as necessary.

5. **GWO Iteration:** Execute a single iteration of GWO for the current population. Evaluate costs, update positions based on encircling and hunting mechanisms guided by leaders α, β, δ , and identify the best GWO solution.
6. **Hybrid Global Best Update:** Compare the global best solution from PSO and the best solution from GWO. Update the PSO global best if the GWO solution is superior.
7. **Iteration and Convergence Check:** Repeat PSO and GWO update steps for the specified number of epochs or until the early stopping criterion is satisfied (e.g., improvement below $\delta = 10^{-5}$).
8. **Output:** Report the optimal PID parameters (K_p^*, K_i^*, K_d^*) , the associated cost $J(K_p^*, K_i^*, K_d^*)$, the total number of cost function evaluations, and the convergence history.

In practice, the PSO and GWO procedures share the same simulation and cost evaluation structure, differing only in the respective update rules. This uniformity facilitates hybrid strategies, in which subsets of the population are evolved using PSO rules while others follow GWO dynamics, enabling exchange of global best solutions and complementary search behavior.

The pseudocodes for the PSO, GWO, and hybrid PSO–GWO PID tuning procedures are provided in Appendices A, B and C, respectively.

4. Benchmark Nonlinear Systems and Simulation Results

4.1. Benchmark Nonlinear Systems

To evaluate the effectiveness of the proposed PID tuning strategies, we consider three representative second-order nonlinear systems. Each system is expressed in the standard form:

$$y''(t) + f(t, y, y') = u(t), \quad t \geq 0 \quad (13)$$

where $y(t)$ is the system output, $u(t)$ is the control input, and $f(t, y, y')$ encodes the system-specific nonlinear dynamics. The three benchmark systems considered in this study are as follows:

- **System 1: Pendulum-like nonlinear system**

$$y''(t) + a \sin(y(t)) + by'(t) = u(t), \quad (14)$$

with parameters $a = 2.0$, $b = 0.4$. This system exhibits moderate nonlinearity and resembles the dynamics of a simple pendulum with damping (for $b > 0$). It is suitable for testing the convergence and robustness of PID tuning algorithms.

- **System 2: Duffing oscillator**

$$y''(t) + \tilde{\delta}y'(t) + \tilde{\alpha}y(t) + \tilde{\beta}y^3(t) = u(t), \quad (15)$$

with parameters $\tilde{\alpha} = -1.0$, $\tilde{\beta} = 1.0$, $\tilde{\delta} = 0.2$. This system is commonly used to test control strategies for stiff and highly nonlinear systems due to the cubic stiffness term, which can produce multiple equilibria and nonlinear oscillatory behavior.

- **System 3: Nonlinear damping system**

$$y''(t) + \tilde{c}_1y'(t) + \tilde{c}_2y(t) + \tilde{c}_3y(t)y'(t) = u(t), \quad (16)$$

with parameters $\tilde{c}_1 = 0.5$, $\tilde{c}_2 = 2.0$, $\tilde{c}_3 = 1.0$. This system introduces a velocity-dependent nonlinear damping term $y y'$, creating asymmetric transient responses and testing the adaptability of metaheuristic PID tuning.

These three benchmark systems collectively cover a range of nonlinear behaviors, including oscillatory, stiff, and asymmetric dynamics. They provide a meaningful testbed for comparing the performance of PSO, GWO, and hybrid PSO–GWO-based PID tuning strategies. The selection of additional systems may be performed in subsequent studies to further explore algorithmic robustness and generalizability.

4.2. PSO Parameters

The PSO algorithm is implemented using the `pyswarm` package with a swarm of $N = 30$ particles, each representing a candidate set of PID parameters (K_p, K_i, K_d). Particle velocities are updated using an inertia weight $\omega = 0.7$ and cognitive and social acceleration coefficients $c_1 = c_2 = 1.5$. The maximum number of iterations is set to 50 to ensure sufficient exploration while keeping the computational cost reasonable, as summarized in Table 1.

Table 1. PSO parameters used in simulations.

Parameter	Symbol	Value
Population size	N	30
Inertia weight	ω	0.7
Cognitive coefficient	c_1	1.5
Social coefficient	c_2	1.5
Max iterations	$k_{\max}$	50

4.3. GWO Parameters

For GWO, we use the `OriginalGWO` class from `mealpy-3.0.3` with a population of $N = 30$ wolves representing candidate PID parameters (K_p, K_i, K_d) . The algorithm follows the standard hunting and encircling strategies, keeping the alpha, beta, and delta leadership weights at their default values ($\alpha = \beta = \delta = 1.0$). The maximum number of epochs is set to 50 to ensure convergence while maintaining a computational cost comparable to PSO, as summarized in Table 2.

Table 2. GWO parameters used in simulations.

Parameter	Symbol	Value
Population size	N	30
Max epochs	$E_{\max}$	50
Alpha, Beta, Delta weights	–	1.0 (default)

4.4. Hybrid PSO–GWO Parameters

For the hybrid PSO–GWO strategy, the same population size and algorithmic parameters are used as in the standalone PSO and GWO cases. However, the population is explicitly divided into two equal sub-swarms: 15 particles evolve according to PSO update rules, while the other 15, acting as wolves, follow the GWO dynamics. At each iteration, global best information and elite solutions are exchanged between the sub-swarms to exploit complementary search behaviors. This setup allows the hybrid approach to leverage both the exploration capability of GWO and the fast convergence of PSO, while maintaining the same total computational budget as the standalone algorithms.

For all simulations, unless stated otherwise, the cost function in (12) employs a weighting factor $\alpha = 1.0$, assigning equal importance to the ITAE and ISO components. During each optimization run, we record and visualize the following outputs: the time-domain response of the controlled system $y(t)$, the evolution of the cost function values corresponding to candidate PID parameter sets (K_p, K_i, K_d) , the minimal achieved cost function value, and the total number of cost function evaluations required to reach the optimum. Additionally, an early stopping criterion is enforced: if two consecutive cost function values differ by less than 10^{-5} , the optimization is terminated to prevent unnecessary computations while ensuring convergence.

4.5. Closed-Loop Response under PSO, GWO, and Hybrid PSO–GWO PID Control

In this section, we present the results of the closed-loop simulations for the three benchmark nonlinear systems introduced in Subsection 4.1. These results demonstrate the effectiveness of the PSO, GWO, and hybrid PSO–GWO PID tuning strategies in regulating second-order nonlinear systems. For each system, we provide graphical illustrations of the system output $y(t)$ under the optimized PID parameters, the evolution of the cost function $J(K_p, K_i, K_d)$ during optimization, and the trajectories of the PID gains (K_p, K_i, K_d) as they converge to their optimal values. The achieved minimum of the cost function and the total number of function evaluations required to reach this optimum are reported, along with any early stopping criteria that were triggered. Early stopping is applied when the absolute difference between two consecutive cost function evaluations falls below 10^{-5} . This ensures computational efficiency while maintaining accuracy in identifying the optimal PID parameters.

• **System 1: Pendulum-like nonlinear system**

Figure 2 presents the simulation results obtained for the pendulum-like nonlinear system. The figure is organized into three rows and two columns. The left column illustrates the closed-loop time responses $y(t)$ to a unit-step reference input under the PID gains optimized by the PSO, GWO, and hybrid PSO–GWO algorithms, respectively. All three metaheuristic methods successfully stabilize the nonlinear system, while the hybrid PSO–GWO algorithm demonstrates faster convergence and a reduced overshoot compared to the individual approaches. The right column of Figure 2 illustrates the evolution of the cost function over successive evaluations, providing insight into the convergence behavior of each optimization strategy. For each case,

the achieved minimum cost and the number of evaluations required to reach it are depicted graphically.

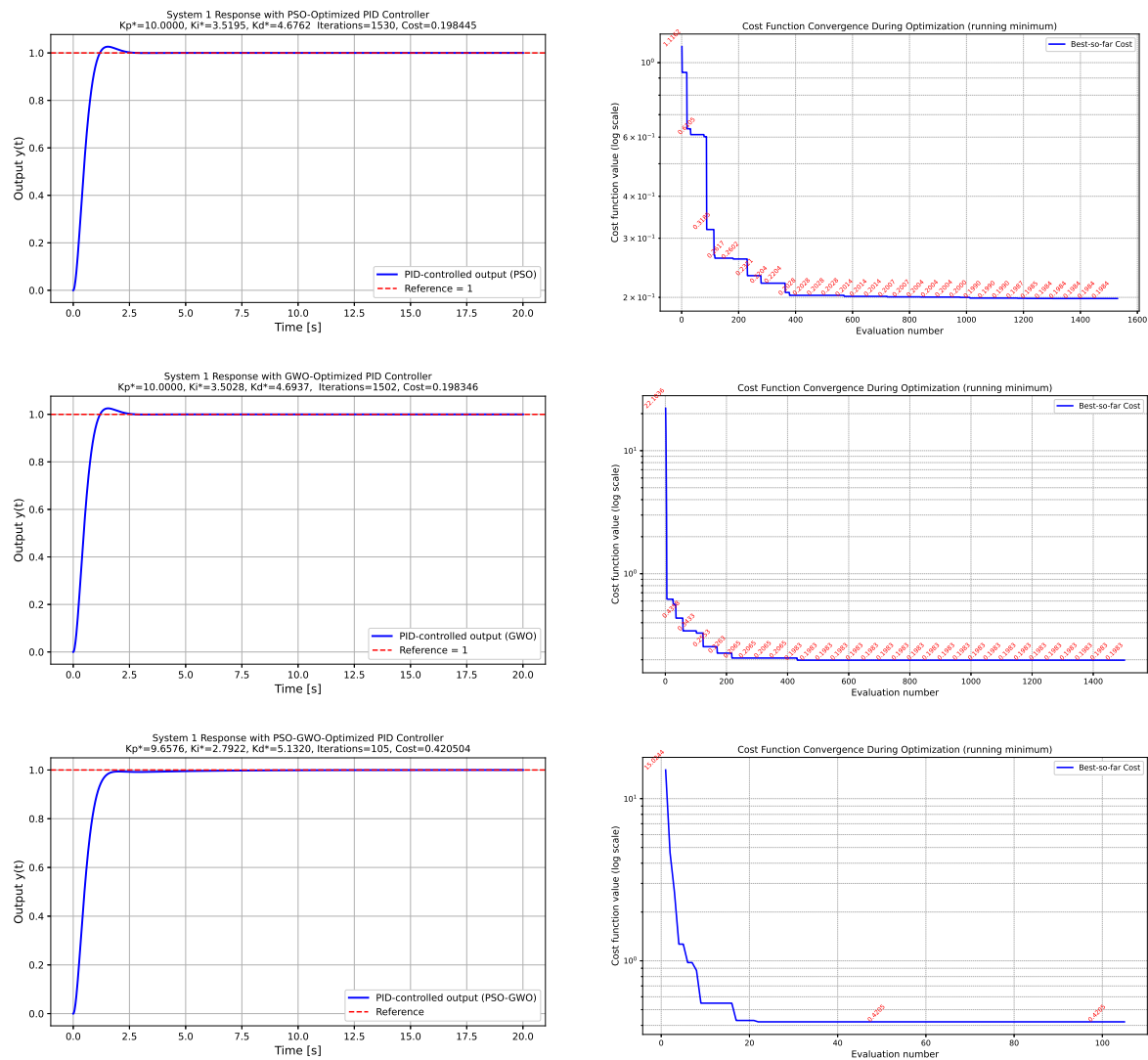


Figure 2. System 1 – Pendulum-like nonlinear system. Comparison of the system responses and convergence profiles for different optimization algorithms. The left column shows the closed-loop responses of the pendulum-like nonlinear system under PSO-, GWO-, and hybrid PSO–GWO-optimized PID gains, while the right column depicts the evolution of the cost function during the optimization process. Minimum cost values and the corresponding number of evaluations are indicated.

• **System 2: Duffing oscillator**

The response of the Duffing oscillator is presented in Figure 3. Due to the presence of the cubic stiffness term, the system exhibits pronounced nonlinear and potentially oscillatory behavior. The left column of Figure 3 displays the time-domain responses of $y(t)$ obtained using PID gains optimized by the PSO, GWO, and hybrid PSO–GWO algorithms. All three controllers are able to regulate the system effectively; however, the hybrid PSO–GWO algorithm produces a noticeably higher overshoot compared to PSO and GWO. This behavior can be mitigated by increasing the weight of the ISO term in the performance function. For instance, when $\alpha = 100$, a significant suppression of overshoot is observed, as illustrated in the first row of Figure 5. The right column of Figure 3 provides insights into the convergence of the cost function across iterations.

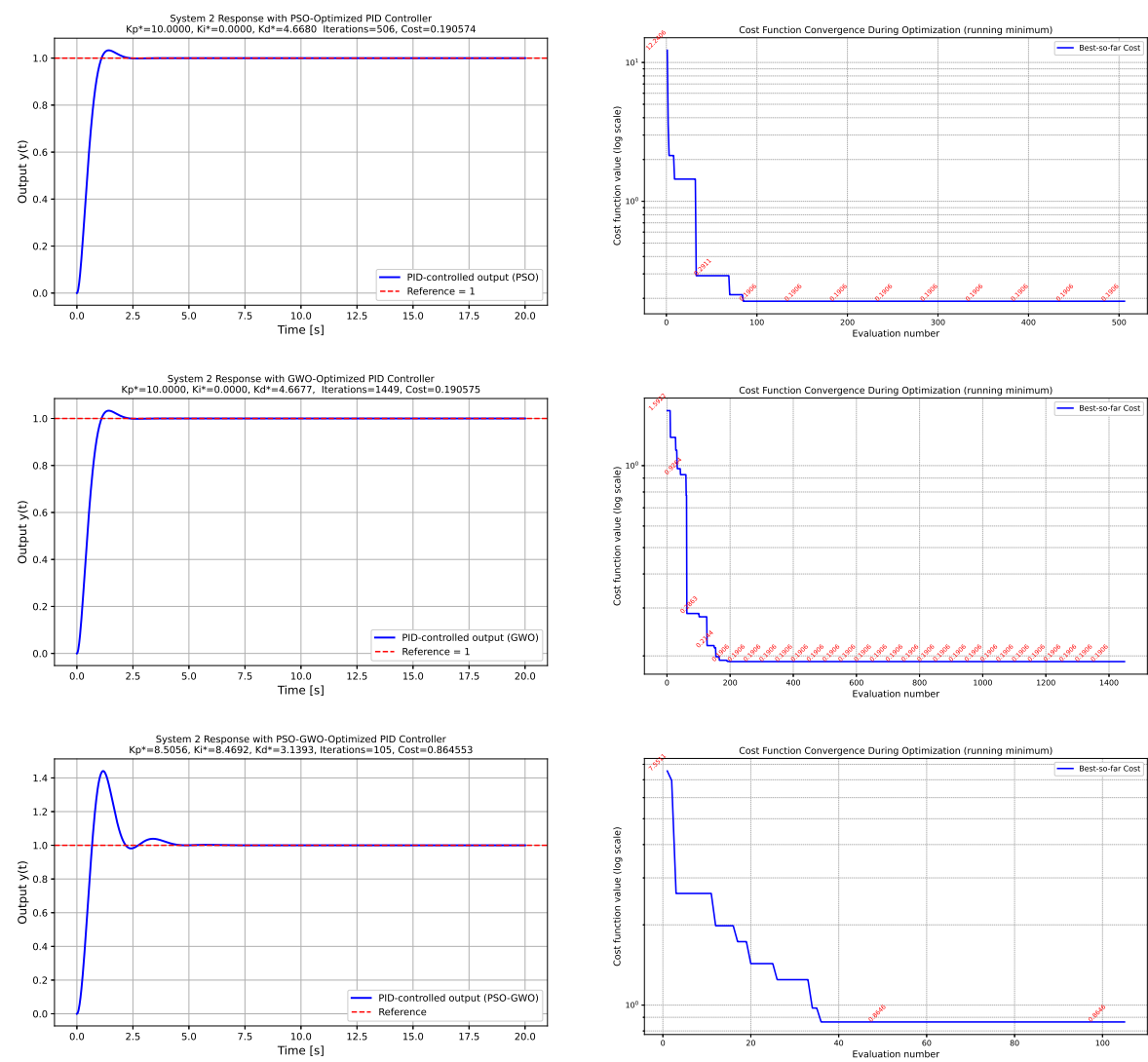


Figure 3. System 2 – Duffing oscillator. Comparison of the Duffing oscillator responses and optimization progress for different metaheuristic strategies. The left column illustrates the closed-loop behavior of the nonlinear oscillator, whereas the right column presents the evolution of the cost function, highlighting convergence differences among PSO, GWO, and the hybrid PSO–GWO approach. Minimum cost and evaluation count are annotated in the plots.

• **System 3: Nonlinear damping system**

Figure 4 illustrates the closed-loop behavior of the nonlinear damping system for the PID controllers optimized by PSO, GWO, and the hybrid PSO–GWO algorithm. The left column shows the system responses $y(t)$ to a unit-step input. The presence of the velocity-dependent damping term yy' introduces asymmetric transient dynamics and nonlinear dissipation effects. All optimized PID controllers are capable of stabilizing the system; however, the hybrid PSO–GWO approach again exhibits a relatively higher overshoot compared to the standalone PSO and GWO methods. Nevertheless, this increased transient excitation occurs at a substantially lower number of cost function evaluations, indicating a more efficient search process. As in the case of the Duffing oscillator, this overshoot can be mitigated by appropriately increasing the weighting coefficient α of the ISO component in the performance index, as can be observed in the second row of Figure 5, thereby emphasizing steady-state accuracy. The right column of Figure 4 depicts the evolution of the cost function during the optimization process, further highlighting the convergence characteristics and trade-offs in exploration and exploitation among the three algorithms.

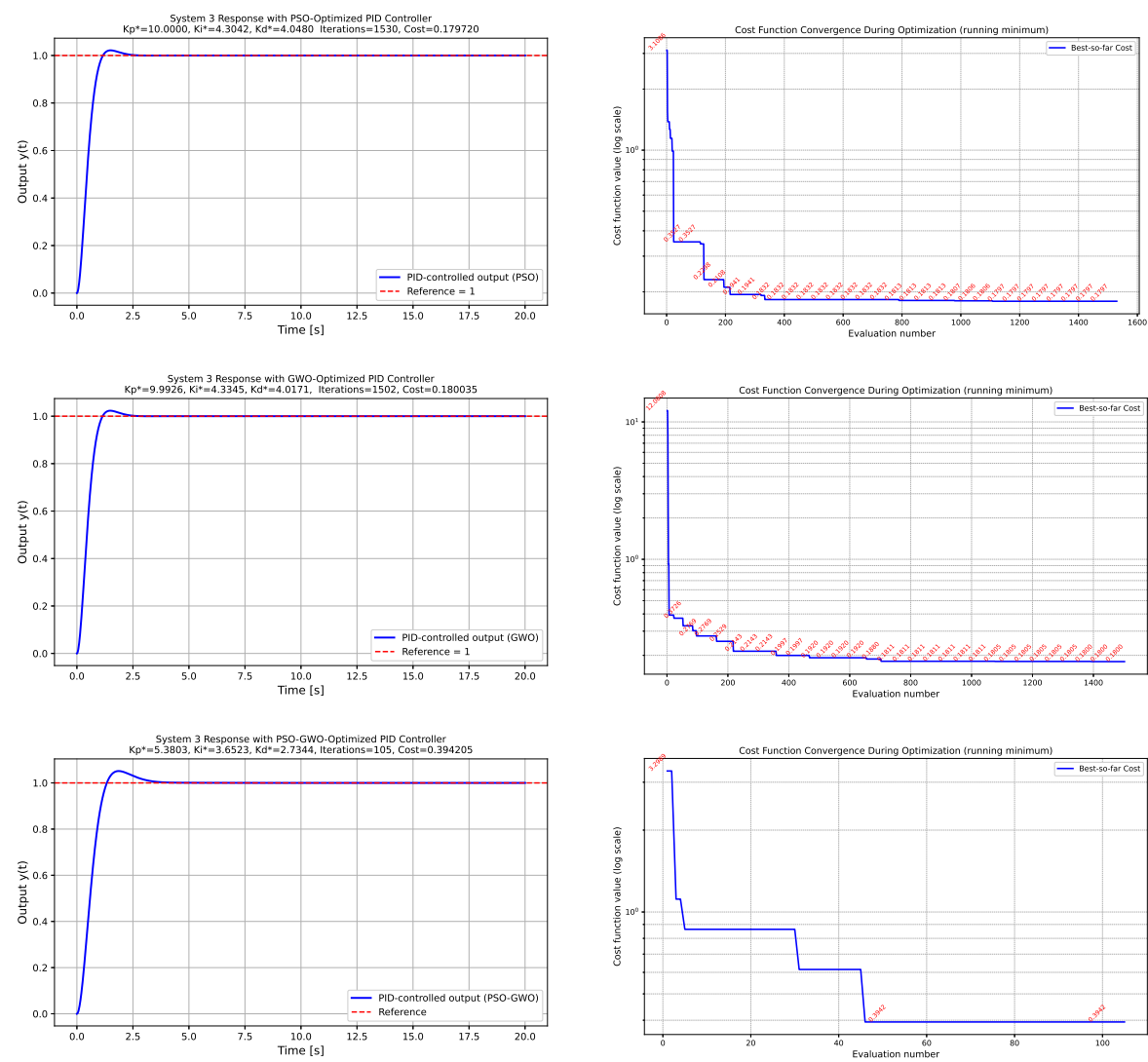


Figure 4. System 3 – Nonlinear damping system. Comparison of the nonlinear damping system responses and cost evolution across optimization algorithms. The left column displays the system output trajectories obtained using PID parameters tuned via PSO, GWO, and hybrid PSO–GWO, while the right column shows the convergence of the cost function.

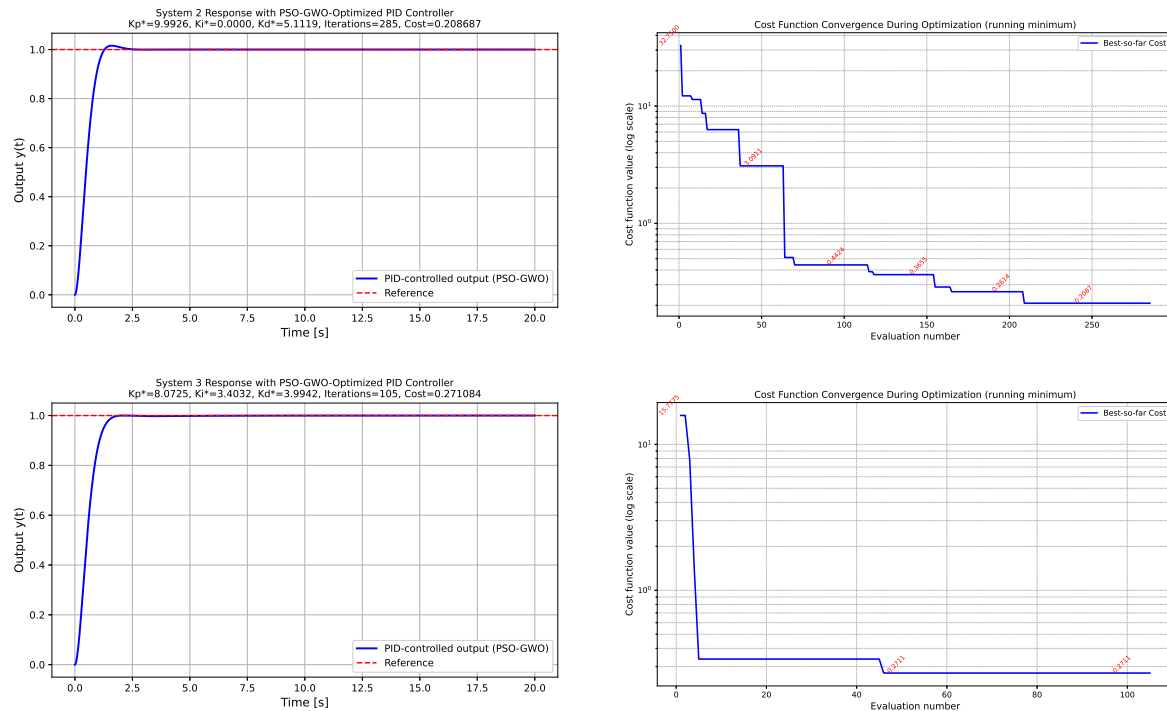


Figure 5. Hybrid PSO–GWO simulations for the Duffing and nonlinear damping systems with $\alpha = 100$. The plots demonstrate the effect of increasing the weighting coefficient α in the ISO term of the performance index. Compared to the previous simulations with $\alpha = 1$, a substantial reduction in overshoot is observed, indicating improved damping and smoother transient behavior. This adjustment effectively mitigates the excessive aggressiveness of the hybrid optimizer observed in earlier cases.

It should be noted that the convergence graph of the cost function $J(K_p, K_i, K_d)$ for the PSO-based PID tuning of System 1 (as well as for some other systems and algorithms) exhibits extreme values during the initial evaluations. These large cost values result from transient numerical instabilities caused by certain initial combinations of PID parameters that produce high initial errors.

To present the tuning process more clearly, only the running minimum of the cost function (the "best-so-far" value) is plotted, which effectively illustrates the progressive improvement of the PID parameters over successive evaluations. Selected cost values are annotated at specific intervals to provide quantitative insight while avoiding visual clutter in the figure.

This approach ensures that the graphical representation accurately reflects the optimization dynamics without being distorted by transiently unstable simulations.

5. Discussion

For a better comparison and quantitative evaluation of the obtained results, the key performance metrics and optimal PID parameters are summarized in Table 3. This table consolidates the outcomes for all three nonlinear benchmark systems and provides insights into the convergence efficiency and stability characteristics of the tested optimization methods.

- **Computational efficiency:** The hybrid PSO–GWO algorithm achieves the target cost values with a substantially smaller number of cost function evaluations—approximately 10% of those required by standalone PSO or GWO. This remarkable reduction can be attributed to the synergy between the exploration capability of GWO and the exploitation behavior of PSO, which enables faster convergence toward promising regions of the search space. The hybrid structure effectively combines global and local search mechanisms, resulting in a rapid decrease in the cost function even in highly nonlinear conditions.
- **System performance and overshoot behavior:** Although the hybrid PSO–GWO demonstrates excellent convergence speed, its time-domain responses for Systems 2 and 3 exhibit pronounced

Table 3. Summary of optimal PID parameters and performance metrics for Systems 1–3.

Algorithm / Metric	System 1	System 2	System 3
PSO			
(K_p^*, K_i^*, K_d^*)	(10.0, 3.5195, 4.6762)	(10.0, 0.0, 4.6680)	(10.0, 4.3042, 4.0480)
$J(K_p^*, K_i^*, K_d^*)$	0.1984	0.1906	0.1797
# cost function evaluations	1530	506	1530
Stop condition	max iter	early stop	max iter
GWO			
(K_p^*, K_i^*, K_d^*)	(10.0, 3.5028, 4.6937)	(10.0, 0.0, 4.6677)	(9.9926, 4.3345, 4.0171)
$J(K_p^*, K_i^*, K_d^*)$	0.1983	0.1906	0.1800
# cost function evaluations	1502	1449	1502
Stop condition	early stop	early stop	early stop
Hybrid PSO–GWO			
(K_p^*, K_i^*, K_d^*)	(9.6576, 2.7922, 5.1320)	(8.5056, 8.4692, 3.1393)	(5.3803, 3.6523, 2.7344)
$J(K_p^*, K_i^*, K_d^*)$	0.4205	0.8646	0.3942
# cost function evaluations	105	105	105
Stop condition	early stop	early stop	early stop

overshoot and oscillations during the transient phase. This behavior suggests that the algorithm tends to generate aggressive control actions due to a strong emphasis on the integral and derivative gains during the optimization process. The observed overshoot can be mitigated by increasing the weighting factor α in the ISO term of the performance function. Numerical experiments show that setting $\alpha \approx 100$ effectively suppresses overshoot without significantly affecting the total number of cost function evaluations, indicating a favorable trade-off between control smoothness and optimization efficiency.

- **Comparison of PSO and GWO:** Both standalone PSO and GWO algorithms achieve stable control performance with low cost values, though at the expense of considerably higher computational effort. GWO, in particular, exhibits consistent but slower convergence, reflecting its exploratory nature. PSO maintains fast convergence and strong exploitation of promising regions, but requires more evaluations to achieve comparable performance to the hybrid approach.
- **Sensitivity to the performance index parameters:** The results further confirm that the design of the performance function, particularly the weighting of its integral and overshoot (ISO) components, plays a critical role in shaping controller behavior. For the hybrid PSO–GWO, an insufficiently penalized overshoot term (small α) leads to aggressive transient responses, whereas larger α values produce smoother trajectories without degrading the convergence rate. Hence, the parameterization of the performance function directly governs the trade-off between control aggressiveness and robustness.

In summary, the hybrid PSO–GWO framework exhibits superior computational efficiency while maintaining satisfactory control accuracy across all considered nonlinear systems. However, the observed overshoot behavior indicates that careful tuning of the ISO weight α is necessary to balance rapid convergence with acceptable transient performance. These findings suggest that hybrid swarm-based metaheuristics are promising for real-time PID tuning, provided that the cost function formulation appropriately reflects the desired control objectives.

5.1. Remark on stochastic variability of optimization results.

During the simulation campaign, repeated runs of the same optimization algorithm—PSO, GWO, or the hybrid PSO–GWO—often converged to slightly different sets of controller parameters, even under identical initialization and stopping criteria (maximum iteration count and tolerance of 10^{-5}). This variability arises from the stochastic nature of metaheuristic search, where random initialization and probabilistic update rules inherently influence the optimization trajectory.

In complex, nonlinear, and multimodal cost landscapes—such as those from chaotic dynamical systems—numerous local minima of comparable depth may exist. As a result, different runs can converge to distinct, yet nearly equivalent, optima that yield similar performance indices while corresponding to different PID parameter combinations. Such differences do not indicate algorithmic instability but rather reflect the multiplicity of acceptable solutions in a rugged landscape. For presentation, representative responses from the most frequently observed convergence patterns were selected.

This phenomenon is common in stochastic metaheuristics. Due to their inherent randomness, independent runs may converge to different local or quasi-global minima even under identical stopping criteria. The stochastic variability of these algorithms is well-documented [18,19], highlighting the importance of multiple independent runs and reporting statistical performance metrics to ensure fair and reproducible comparisons.

In the following subsection, the robustness and effectiveness of the proposed PSO–GWO-based PID tuning framework are rigorously evaluated on one of the most challenging nonlinear dynamic benchmarks—the chaotic Duffing oscillator.

5.2. Robustness Test on the Chaotic Duffing Oscillator

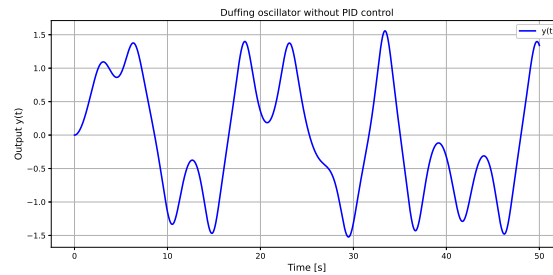
The Duffing oscillator represents one of the most challenging nonlinear dynamical systems to control due to its strong nonlinearity, sensitivity to initial conditions, and the presence of chaotic behavior under certain parameter regimes. Its dynamics can be described by the nonlinear second-order differential equation:

$$y''(t) + \tilde{\delta} y'(t) + \tilde{\alpha} y(t) + \tilde{\beta} y^3(t) = \tilde{\gamma} \cos(\tilde{\omega}t) + u(t), \quad (17)$$

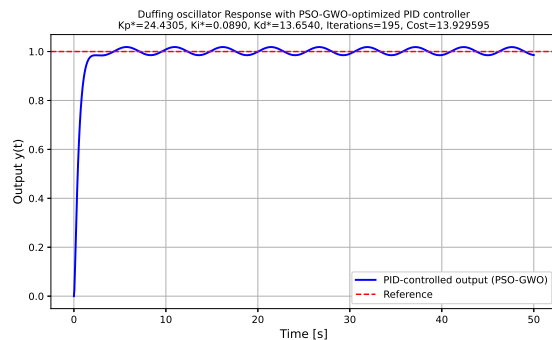
where $\tilde{\delta}$ denotes the damping coefficient, $\tilde{\alpha}$ and $\tilde{\beta}$ are the linear and nonlinear stiffness parameters, $\tilde{\gamma}$ represents the amplitude of the external forcing, $\tilde{\omega}$ is its excitation frequency, and $u(t)$ is the control input generated by the PID controller.

For the selected parameter set $\tilde{\delta} = 0.2$, $\tilde{\alpha} = -1.0$, $\tilde{\beta} = 1.0$, $\tilde{\gamma} = 0.5$, and $\tilde{\omega} = 1.2$, the uncontrolled system ($u(t) = 0$) exhibits aperiodic oscillations and extreme sensitivity to small perturbations—hallmarks of chaotic behavior. Such characteristics render conventional linear control strategies largely ineffective, as even minor variations in the control signal or in the system state may lead to significantly different trajectories. Consequently, the chaotic Duffing oscillator is frequently employed as a benchmark for evaluating the robustness, adaptability, and convergence properties of advanced control and optimization algorithms, particularly in the context of nonlinear and chaotic dynamics.

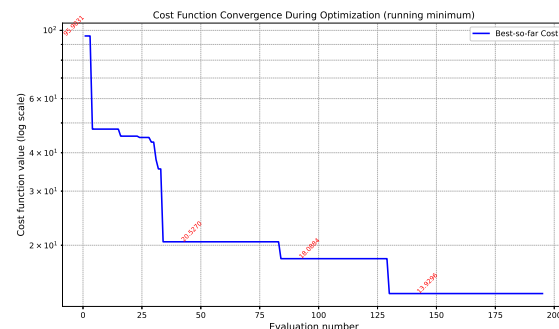
Figure 6 illustrates the contrasting responses of the Duffing oscillator without and with PID control, assuming zero initial conditions for the oscillator, $y(0) = 0$ and $y'(0) = 0$. In the absence of control, the system exhibits irregular, chaotic oscillations that fail to settle to any equilibrium, demonstrating its highly unstable and unpredictable nature. In contrast, when the PID controller is tuned using the proposed hybrid PSO–GWO optimization method, the chaotic oscillations are effectively suppressed, and the system exhibits smooth, near-periodic behavior. Although small oscillations remain due to the periodic excitation term $\cos(\tilde{\omega}t)$, the overall motion is significantly stabilized and follows the desired reference trajectory. This demonstrates the hybrid algorithm's ability to handle strong nonlinearities and nonstationary dynamics, confirming both its robustness and adaptability in one of the most demanding nonlinear control scenarios.



(a) Duffing oscillator without PID control.



(b) Duffing oscillator with PSO-GWO-optimized PID controller.



(c) Convergence of the cost function.

Figure 6. Comparison of the Duffing oscillator time responses: uncontrolled chaotic behavior (a) versus stabilized dynamics achieved by the proposed PSO-GWO-based PID tuning (b). The hybrid controller effectively suppresses chaotic oscillations and enforces smoother convergence toward the reference trajectory. For the PID tuning, the search space for the controller parameters was $[0, 25]^3$, and the weighting coefficient in the cost function (12) was set to $\alpha = 100$.

6. Conclusions and Future Work

This study presented a comprehensive investigation of PID controller tuning for strongly nonlinear dynamic systems using swarm-based optimization techniques. The main objective was to design a direct, simulation-based optimization approach that does not rely on system linearization or analytical model simplification. Three representative nonlinear systems—pendulum-like, Duffing-type, and nonlinear damping models—were employed to evaluate the algorithms under comparable conditions.

The comparative analysis demonstrated that all tested metaheuristics (PSO, GWO, and the hybrid PSO-GWO) were capable of identifying PID parameters that ensure closed-loop stability and satisfactory dynamic performance. Among them, the hybrid PSO-GWO exhibited the fastest convergence and the lowest number of cost function evaluations, typically requiring less than 10% of the computational effort compared to the standalone algorithms. This superior efficiency stems from the hybrid algorithm's ability to exploit the exploitation capabilities of PSO while leveraging the exploration power of GWO.

However, the hybrid method also showed a tendency toward aggressive control behavior, manifesting as higher overshoot and transient oscillations, particularly in the Duffing system. Such behavior can be effectively mitigated by adjusting the weighting coefficient α in the ISO component of the performance index (12), as higher α values penalize overshoot and yield smoother responses without significantly increasing computational cost.

The results further confirm that PID controllers, when properly tuned using modern swarm intelligence techniques, remain highly effective even for strongly nonlinear systems characterized by non-symmetric or amplitude-dependent dynamics. The combination of simplicity, adaptability, and high performance makes swarm-optimized PID control a viable solution for a wide range of nonlinear control problems.

Future research will focus on several directions:

- Incorporating adaptive or self-tuning strategies for real-time adjustment of PID gains based on system operating conditions.
- Extending the hybrid optimization framework to multi-objective formulations, balancing competing criteria such as energy consumption, robustness, and tracking precision.
- Applying the proposed approach to higher-order nonlinear and time-delay systems, and validating it through hardware-in-the-loop or real experimental platforms.
- Due to the inherent stochastic nature of metaheuristic optimization, incorporating statistical assessment across multiple independent runs to ensure robust evaluation of convergence characteristics and to quantify the distribution of attainable control performances in strongly nonlinear systems.

In summary, this study demonstrates that the synergy of swarm intelligence and classical PID control offers a powerful and computationally efficient approach to nonlinear system regulation, bridging the gap between heuristic optimization and practical control implementation.

Acknowledgement

This work was supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic and the Slovak Academy of Sciences (Grant No. 1/0318/25).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author thanks the editors and the anonymous reviewers for their insightful comments which improved the quality of the paper.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. PID tuning of second-order nonlinear systems using PSO

Algorithm A1 PID tuning of second-order nonlinear systems using PSO

- 1: **Input:** Search bounds $K_p, K_i, K_d \in [0, 10]$, swarm size $N = 30$, max iterations $k_{\max} = 50$, tolerance 10^{-5}

2: **Output:** Optimal PID parameters (K_p^*, K_i^*, K_d^*) , cost J^* , simulation output $y(t)$

3: Initialize swarm of N particles randomly within bounds

4: Evaluate initial costs, set personal/global bests

5: **for** iteration = 1 to $k_{\max}$ **do**

6: **for** each particle **do**

7: Update velocity and position according to PSO rules

8: Evaluate cost J

9: Update personal/global best if improved

10: **if** cost change $< 10^{-5}$ **then**

11: **break**

12: **end if**

13: **end for**

14: **end for**

15: $[K_p^*, K_i^*, K_d^*] \leftarrow$ global best

16: $(t, y) \leftarrow$ simulate PID with $[K_p^*, K_i^*, K_d^*]$

17: **Return** Optimal PID parameters, cost, simulation output, convergence history

Appendix B. PID tuning of second-order nonlinear systems using GWO

Algorithm A2 PID tuning of second-order nonlinear systems using GWO

```

1: Input: Bounds  $K_p, K_i, K_d \in [0, 10]$ , population  $N = 30$ , epochs  $E_{\max} = 50$ , tolerance  $10^{-5}$ 
2: Output: Optimal PID parameters  $(K_p^*, K_i^*, K_d^*)$ , cost  $J^*$ , simulation output  $y(t)$ 
3: Initialize population of  $N$  wolves randomly
4: Set  $\alpha, \beta, \delta$  wolves and global best  $g_{\text{best}}$ 
5: for epoch = 1 to  $E_{\max}$  do
6:   for each wolf do
7:     Evaluate cost  $J$  via simulate_pid and cost_pid
8:     Update position using GWO rules (encircle/hunt)
9:     if improvement  $< 10^{-5}$  then
10:       break
11:     end if
12:   end for
13:   Update  $\alpha, \beta, \delta$  wolves
14: end for
15:  $[K_p^*, K_i^*, K_d^*] \leftarrow g_{\text{best}}$ 
16:  $(t, y) \leftarrow \text{simulate PID with } [K_p^*, K_i^*, K_d^*]$ 
17: Return Optimal PID parameters, cost, simulation output, convergence history

```

Appendix C. PID tuning of second-order nonlinear systems using Hybrid PSO–GWO

Algorithm A3 PID tuning of second-order nonlinear systems using Hybrid PSO–GWO

```

1: Input: Bounds  $K_p, K_i, K_d \in [0, 10]$ , PSO swarm  $N_{\text{PSO}} = 15$ , GWO swarm  $N_{\text{GWO}} = 15$ , epochs  $E_{\max}$ , early stopping  $\delta = 10^{-5}$ 
2: Output: Optimal PID parameters  $(K_p^*, K_i^*, K_d^*)$ , cost  $J^*$ , simulation output  $y(t)$ 
3: Initialize PSO swarm, velocities, personal bests, global best
4: Initialize GWO swarm for single-step evaluation
5: for iteration = 1 to  $E_{\max}$  do
6:   PSO update:
7:   for each PSO particle do
8:     Update velocities and positions
9:     Clip positions within bounds
10:    Evaluate cost via simulate_pid and cost_pid
11:    Update personal bests
12:    Update global best if improved
13:  end for
14:  GWO single iteration:
15:  for each GWO wolf do
16:    Run one step of GWO
17:    Evaluate cost for GWO best
18:    if better than PSO global best then
19:      Update PSO global best
20:    end if
21:  end for
22:  Print current iteration and PSO global best parameters/cost
23:  if  $|\text{prev\_best\_cost} - J_{\text{global best}}| < \delta$  then
24:    Early stopping triggered; break
25:  end if
26:  Update prev_best_cost
27: end for
28:  $[K_p^*, K_i^*, K_d^*] \leftarrow \text{PSO global best}$ 
29:  $(t, y) \leftarrow \text{simulate PID with } [K_p^*, K_i^*, K_d^*]$ 
30: Return Optimal PID parameters, cost, simulation output, convergence history

```

Funding: This publication has been published with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic within project VEGA 1/0193/22 "Návrh identifikácie a systému monitorovania parametrov výrobných zariadení pre potreby prediktívnej údržby v súlade s konceptom Industry 4.0 s využitím technológií Industrial IoT".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author thanks the editors and the anonymous reviewers for their insightful comments which improved the quality of the paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. K. J. Åström, T. Hägglund, Advanced PID Control, ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, North Carolina, 2006.
2. N. S. Nise, Control Systems Engineering, 6th Edition, Wiley, Hoboken, NJ, 2011.
3. K. Ogata, Modern Control Engineering, 5th Edition, Pearson, Upper Saddle River, NJ, 2010.
4. D. Çelik, N. Khosravi, M. A. Khan, M. Waseem, H. Ahmed, Advancements in nonlinear PID controllers: A comprehensive review, *Computers & Electrical Engineering*, *In press* (October 2025).
5. F. S. Prity, Nature-inspired optimization algorithms for enhanced load balancing in cloud computing: A comprehensive review with taxonomy, comparative analysis, and future trends, *Swarm and Evolutionary Computation* 97 (2025) 102053.
6. R. C. Eberhart, J. Kennedy, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.
7. J. Kennedy, R. C. Eberhart, A discrete binary version of the particle swarm algorithm, in: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 4104–4108.
8. Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, in: *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.
9. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* 69 (2014) 46–61.
10. M. A. Shaheen, H. M. Hasanien, A. Alkuhayli, A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution, *Ain Shams Engineering Journal* 12 (1) (2021) 621–630.
11. T. L. Nguyen, Q. A. Nguyen, A multi-objective PSO-GWO approach for smart grid reconfiguration with renewable energy and electric vehicles, *Energies* 18 (8) (2025). doi:10.3390/en18082020.
12. A. B. Alyu, A. O. Salau, B. Khan, J. N. Eneh, Hybrid GWO-PSO based optimal placement and sizing of multiple PV-DG units for power loss reduction and voltage profile improvement, *Scientific Reports* 13 (2023) 6903.
13. J. Águila León, C. Vargas-Salgado, D. Díaz-Bello, C. Montagud-Montalvá, Optimizing photovoltaic systems: A meta-optimization approach with GWO-Enhanced PSO algorithm for improving mppt controllers, *Renewable Energy* 230 (2024) 120892.
14. A. S. Bhandari, A. Kumar, M. Ram, Grey wolf optimizer and hybrid PSO-GWO for reliability optimization and redundancy allocation problem, *Quality and Reliability Engineering International* 39 (3) (2023) 905–921.
15. F. A. Şenel, F. Gökçe, A. S. Yüksel, T. Yiğit, A novel hybrid PSO-GWO algorithm for optimization problems, *Engineering with Computers* 35 (2019) 1359–1373.
16. A. Bouaddi, R. Rabeh, M. Ferfra, Optimal control of automatic voltage regulator system using hybrid PSO-GWO algorithm-based pid controller, *Bulletin of Electrical Engineering and Informatics* 13 (5) (2023) 8186.
17. S. Charkoutsis, M. Kara-Mohamed, A particle swarm optimization tuned nonlinear PID controller with improved performance and robustness for first order plus time delay systems, *Results in Control and Optimization* 10 (2023) Article 100289. doi:10.1016/j.rico.2023.100289.

18. W. Dillen, G. Lombaert, M. Schevenels, Performance assessment of metaheuristic algorithms for structural optimization taking into account the influence of algorithmic control parameters, *Frontiers in Built Environment* 7 (2021) 618851. doi:10.3389/fbuilt.2021.618851.
19. A. A. Juan, P. Keenan, R. Martí, S. McGarraghy, J. Panadero, P. Carro, D. Oliva, A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics, *Annals of Operations Research* 320 (2023) 831–861. doi:10.1007/s10479-021-04142-9.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.