

Article

Swarm-Inspired Algorithms to Optimize a Nonlinear Gaussian Adaptive PID Controller

Erickson Diogo Pereira Puchta ^{1,*}, Priscilla Bassetto ^{1,†}, Lucas Henrique Biuk ^{2,†},
Marco Antônio Itaborahy Filho ^{2,†}, Attilio Converti ^{3,†}, Mauricio dos Santos Kaster ^{2,†} and
Hugo Valadares Siqueira ^{1,†}

¹ Graduate Program in Production Engineering, Federal University of Technology-Paraná (UTFPR), Dr. Washington Subtil Chueire St., 330, Jardim Carvalho, Ponta Grossa 84017-22, Brazil; pri_bass@hotmail.com (P.B.); hugosiqueira@utfpr.edu.br (H.V.S.)

² Graduate Program in Electrical Engineering, Federal University of Technology-Paraná (UTFPR), Dr. Washington Subtil Chueire St., 330, Jardim Carvalho, Ponta Grossa 84017-22, Brazil; lucasbiuk1902@gmail.com (L.H.B.); marquin_itaborahy@hotmail.com (M.A.I.F.); mkaster@utfpr.edu.br (M.d.S.K.)

³ Department of Civil, Chemical and Environmental Engineering, University of Genoa, Via Balbi, 5, 16126 Genoa, Italy; converti@unige.it

* Correspondence: ericksonpuchta@gmail.com

† These authors contributed equally to this work.



Citation: Puchta, E.D.P.; Bassetto, P.; Biuk, L.H.; Itaborahy Filho, M.A.; Converti, C.A.; Kaster, M.d.S.; Siqueira, H.V. Swarm-Inspired Algorithms to Optimize a Nonlinear Gaussian Adaptive PID Controller. *Energies* **2021**, *14*, 3385. <https://doi.org/10.3390/en14123385>

Academic Editors: Anastasios Dounis and Edmundas Kazimieras Zavadskas

Received: 08 April 2021

Accepted: 27 May 2021

Published: 8 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: This work deals with metaheuristic optimization algorithms to derive the best parameters for the Gaussian Adaptive PID controller. This controller represents a multimodal problem, where several distinct solutions can achieve similar best performances, and metaheuristics optimization algorithms can behave differently during the optimization process. Finding the correct proportionality between the parameters is an arduous task that often does not have an algebraic solution. The Gaussian functions of each control action have three parameters, resulting in a total of nine parameters to be defined. In this work, we investigate three bio-inspired optimization methods dealing with this problem: Particle Swarm Optimization (PSO), the Artificial Bee Colony (ABC) algorithm, and the Whale Optimization Algorithm (WOA). The computational results considering the Buck converter with a resistive and a nonlinear load as a case study demonstrated that the methods were capable of solving the task. The results are presented and compared, and PSO achieved the best results.

Keywords: GAPID controller; PSO; ABC; WOA; optimization

1. Introduction

In recent years, a significant amount of research has focused on solving optimization problems without any prior knowledge [1,2]. Due to the multiple characteristics of real world problems, such as non-linearity, discontinuity, multimodality, non-differentiability, and so on, traditional mathematical techniques based on derivatives of the gradient may be insufficient for the posed challenges [3].

Inspired by the natural relationships of groups of animals, swarm-based algorithms, such as Particle Swarm Optimization (PSO) [4–6], the Artificial Bee Colony algorithm [7], Bacterial Foraging Optimization (BFO) [8], Cat Swarm Optimization (CSO) [9,10], and Ant Colony Optimization (ACO) [11], among others, provided sufficient evidence of efficiency and effectiveness in finding the optimal solutions to complex optimization problems [12–14].

When dealing with optimization problems, the agents in metaheuristics roam in the search space to obtain good solutions using experimental and local information [15]. The choice of which one performs better depends on the application problem [2,5,9,12,16]. The objective of this work is to analyze the behavior of three bio-inspired optimization algorithms for searching for optimal solutions for a multimodal problem as represented

by the Gaussian Adaptive Proportional, Integral and Derivative (GAPID) control strategy applied to typical power supplies of medical equipment with the Buck converter topology. The objective of this work is to analyze the behavior of three bio-inspired optimization algorithms for searching for optimal solutions for a multimodal problem as represented by the Gaussian Adaptive Proportional, Integral and Derivative (GAPID) control strategy applied to typical power supplies of medical equipment with the Buck converter topology.

The GAPID is a recently proposed technique reported in [17–20], dating back to the years 2016 up to 2019, and studies about the use of bio-inspired optimization algorithms in the design of this controller are being conducted up to the present. This control strategy is difficult to design as it does not have an algebraic solution for the adaptive parameters of the controller. In [18], six variations of genetic algorithms were tested for tuning the GAPID and showed good enhancement of GAPID over traditional PID.

In [19], PSO algorithms with six variations were tested and demonstrated good enhancements of GAPID over PID. In [20], these two algorithms were employed in the same application and compared with both demonstrating good results and little advantage for PSO. The latter also presented faster convergence and lower computational cost; however, on the other hand, the agents positions on the last iterations were very concentrated, which represents a low search capacity, and may converge to a sub-optimal solution instead of the global one.

We highlight that control problems are a major topic in engineering. Despite PID standing as a reference in the control field, currently being the most used control technique and widely employed in industry due to its good performance and robustness, it exhibits certain limitations and, to overcome some of these restrictions, several alternatives have been proposed.

Adaptive techniques can be applied to PID by modifying the gains dynamically to achieve better performances [21–27]. The design of many of adaptive controllers is not an easy task, demanding additional tools to determine the parameters, such as linear or meta-heuristics optimization algorithms. In recent years, the use of bio-inspired optimization, such as PSO, genetic algorithms, or even hybrid multipopulational approaches in control problems stood out due to their capacity for solving complex interdisciplinary problems.

As stated before, the goal of this investigation is to evaluate the performance of three bio-inspired algorithms: the Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO), and the Whale Optimization Algorithm (WOA), to optimize the parameters of the GAPID controlling a DC–DC step-down Buck converter, a widely used converter built into power supplies for most consumer electronic devices. These methods were chosen due to their high optimization power, ease of implementation and understanding, and previous good results reported in the literature. In addition, the PSO and ABC are the most prominent swarm-inspired algorithms [2], while the WOA, has presented good results in the literature, including in control problems [28–30].

While there are other studies about optimization that have referred to this problem, the analysis of the cited optimization strategies has not yet been conducted. Moreover, a nonlinear load was added to the study, which helps to validate the solutions found.

The remainder of this paper is divided as follows: Section 2, with Sections 2.1–2.3 present the three bio-inspired algorithms, PSO, ABC, and WOA, respectively; Section 3 shows the details about the GAPID controller; Section 4 reveals the Buck converter; Section 5 presents the computational results, a critical analysis, and some open questions; and Section 6 summarizes the main conclusions.

2. Optimization Algorithms

2.1. Particle Swarm Optimization (PSO)

In the last two decades, the literature has proven the robustness of Particle Swarm Optimization (PSO) in solving continuous non-linear optimization problems [2,6]. The algorithm emulates the behavior of the social activities of animals, for example, insects, birds, and so on [31]. PSO can produce high quality solutions with small computational effort and can present more stable convergence characteristics compared with others stochastic methods [32].

In PSO, the agents are named particles, and they represent the candidate solutions. They are conceptual entities that constitute a swarm that flies through a multidimensional search space [33]. At any time t , each particle has a different position \mathbf{x}_t and a speed \mathbf{v}_t , which are both influenced by its own best position achieved thus far (**pbest**—personal best) and the best neighbor's position (**gbest**—global best) [34]. Therefore, the new position of particle p is updated according to Equation (1):

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1} \quad (1)$$

In the same way, the velocity of particle p is updated as given by Equation (2):

$$\mathbf{v}_{t+1} = \omega_t \cdot \mathbf{v}_t + r_1 \cdot k_1 \cdot (\mathbf{pbest} - \mathbf{x}_t) + r_2 \cdot k_2 \cdot (\mathbf{gbest} - \mathbf{x}_t) \quad (2)$$

where

- \mathbf{v}_t : the current particle velocity;
- ω_t : the inertia constant, set at 0.9;
- \mathbf{x}_t : the position of all particle from previously looping;
- k_1 : the cognitive coefficient;
- k_2 : the social coefficient;
- r_1 and r_2 : the randomness factors generated from a uniform distribution, in the range between 0 and 1;
- **pbest**: the best position of each particle (personal best); and
- **gbest**: the best global position among all particles (global best).

Algorithm 1 presents the steps involved in the PSO development.

Algorithm 1 Particle Swarm Optimization-Pseudocode

```

1: Randomly initialize a population of particles
2: Calculate the fitness of each particle through the position
3: for  $t = 1$  : maximum number of iterations do
4:   for each particle do
5:     Update  $\mathbf{v}$  using Equation (2)
6:     Update particle position using Equation (1)
7:     Evaluate the fitness
8:     if  $f(\mathbf{x}_{t+1})$  better than  $f(\mathbf{pbest})$  then
9:        $\mathbf{pbest} = \mathbf{x}_{t+1}$ 
10:    end if
11:    if  $f(\mathbf{x}_{t+1})$  better than  $f(\mathbf{gbest})$  then
12:       $\mathbf{gbest} = \mathbf{x}_{t+1}$ 
13:    end if
14:  end for
15: end for
16: return  $\mathbf{gbest}$ 

```

Adapted from [35]

2.2. Artificial Bee Colony (ABC)

The Artificial Bee Colony (ABC) algorithm was proposed by Karaboga in 2005 [36]. Currently, it is one of the most prominent swarm-inspired methods for optimization.

The quality (fitness) of a food source is biologically inspired by the amount and quality of the nectar in a flower. The population of the food source is the agents, representing positions in the search space (candidate solutions). The artificial bees are, indeed, the metaphor for the optimization mechanisms (operators). The bees induce the exploration of different areas of the search space, and a selection process guarantees the exploration of previous experiences [7,37].

To run ABC, it is necessary to perform an initialization stage. After, three kinds of operators are used: employed bees, onlooker bees, and scout bees [38].

2.2.1. Initialization Stage

An agent in ABC (food sources) is an attribute vector x containing the D parameters (dimensions) of the current problem. The initial population is distributed in a uniform and random way, as described in Equation (3):

$$x_{ij} = x_{ij}^{min} + rand(0,1)(x_{ij}^{max} - x_{ij}^{min}) \quad (3)$$

such that min and max are the lower and upper limits of j ($j = 1, 2, \dots, D$) or the bounds of each parameter of the initial solution x_i ($i = 1, 2, \dots, \frac{NS}{2}$), and NS is the total number of food sources.

2.2.2. Employed Bee Stage

The employed bee stage seeks new food sources in the neighborhood of the current solutions. If a better source is found, we apply a greedy selection among the new and the old solutions, considering their quality or fitness value. The new candidate solution v_{ij} provided by an employed bee is generated according to Equation (4):

$$v_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}), \text{ where } k \neq i \quad (4)$$

where v_{ij} is the new candidate to replace x_{ij} , $k \in 1, 2, \dots, \frac{NS}{2}$ and $j \in 1, 2, \dots, D$ are the indexes chosen randomly, with $k \neq i$, and ϕ_{ij} is an aleatory value between $[-1; 1]$.

2.2.3. Onlooker Bee Stage

In nature, employed bees exchange information about the quality and position of a food sources with the onlooker bees through the "waggle dance", sharing information on the aptitude and nectar. This idea is used in ABC in the sense that the onlooker bees choose a food source to explore based on its fitness value. The source is selected considering a roulette wheel scheme, as in Equation (5):

The highest probability of aptitude updates the position of the employed bees; mathematically, this can be calculated by (5):

$$P_i = \frac{fit_i}{\sum_{n=1}^{NS} fit_n} \quad (5)$$

where P_i is the probability of selecting the source x_i , and fit_i is the fitness value of x_i .

Therefore, the onlooker bees tend to select the food sources with high fitness values. Once again, the new solutions provided by this operator are given by Equation (4).

2.2.4. Scout Bee Stage

The scout bee stage begins the search when the current food source in the population runs out. This situation arises when a food source is abandoned and its position is not updated by the predetermined number of cycles defined by the variable *ABANDON_LIMIT*.

The employed bee associated with the abandoned source becomes a scout bee, which generates a new random source in the environment. The new solution is given by (3).

Finally, the pseudocode in Algorithm 2 illustrates the ABC procedure and how its structure is performed [37].

Algorithm 2 Artificial Bee Colony-Pseudocode

- 1: Initialize a bee swarm with N employed and onlooker bees
- 2: Initialize a random clustering for employed bees and compute the cluster center matrix.
- 3: Compute fitness values.
- 4: If the number of scouts exceeds MAX_SCOUTS , make the employed bees with the worst fitness as scouts and the rest as employed.
- 5: Apply random update to employed bee. If fitness has improved, update the cluster centers.
- 6: Set the solution (cluster center matrix) for each onlooker bee by choosing a solution from the employed bees with probability proportional to its fitness.
- 7: Apply random update to each onlooker. If the fitness improves, update cluster centers.
- 8: Make the employed bees scouts for which $ABANDON_LIMIT$ has been reached.
- 9: Remember the best solution.
- 10: Go to 3 and repeat MCN times.

Adapted from [39]

2.3. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA) was proposed by Mirjalili and Lewis in 2016 [40]. This algorithm mimics the unique foraging behavior of humpback whales when capturing prey. This process is called the bubble feeding method [41]. These whales hunt their prey close to the surface by creating bubbles along a path.

According to [42], humpback whales adopt two maneuvering techniques associated with bubbles called *upward spirals* and *double loops*. In the first maneuver, they dive into the water about 12 m deep, and then start to make a wave of bubbles in a spiral shape surrounding the prey. Finally, the whales swim quickly toward the surface to hunt. The second maneuver includes three different stages: the coral loop, lobtail, and capture loop [42].

2.3.1. Encircling Prey

Humpback whales recognize the location of prey and encompass them. The WOA assumes that the position of the whales (agents) is $x_i \forall i = 1, 2, \dots, M$, in which M represents the number of whales that are randomly initialized in the search space. Thus, the algorithm assumes that the best current candidate solution is the target prey or it is close to optimal. After that, the best agent is assigned, and the other agents update their positions considering the best solution x_{best} , as in Equations (6) and (7) [40]:

$$D = |C \cdot x_{best} - x_t| \quad (6)$$

$$x_{t+1} = x_{best} - A \cdot D \quad (7)$$

where A and C are coefficient vectors, calculated by Equations (8) and (9);

$$A = 2a \cdot r - a \quad (8)$$

$$C = 2r \quad (9)$$

where a is linearly reduced from 2 to 0 over the iterations, and r is a random vector in the interval $[0, 1]$.

The vector \mathbf{x}_{best} must be updated every iteration. The feeding behavior using a bubble-net has two steps called the exploration phases [40,43].

2.3.2. Bubble-Net

In the bubble-net phase, humpback whales adopt two mechanisms to chase prey, which can be explained mathematically as follows:

- Shrinking encircling mechanism: A decrease in the value of a in Equation (8) controls the shrinkage mechanism. Then, the whale positions are updated according to Equations (6)–(9);
- Spiral updating position: The following steps perform the simulation for this behavior:
 1. The distance between the current position \mathbf{x} and the best solution \mathbf{x}_{best} is calculated;
 2. The propeller-shaped movement of humpback whales is imitated by creating a spiral equation as in (10) [44]:

$$\mathbf{x}_{t+1} = \mathbf{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{x}_{\text{best}} \quad (10)$$

where $\mathbf{D}' = |\mathbf{x}_{\text{best}} - \mathbf{x}_t|$ indicates the distance of the i -th whale to the prey or best solution obtained so far, b is a constant to define the shape of the logarithmic spiral, l is a random value drawn in the interval $[-1,1]$, and (\cdot) is the element-by-element multiplication.

According to the previous equations, humpback whales move toward the prey with two different types of movements simultaneously:

$$\mathbf{x}_{t+1} = \begin{cases} \mathbf{x}_{\text{best}} - \mathbf{A} \cdot \mathbf{D} & , \text{ if } p < 0.5 \\ \mathbf{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{x}_{\text{best}} & , \text{ if } p > 0.5 \end{cases} \quad (11)$$

where p is a random number in $[0,1]$.

2.3.3. Search for Pray

The humpback whale also searches for prey randomly to realize exploration, updating their positions according to a candidate chosen at random. Mathematically, if $\mathbf{A} > 1$, the candidate agent moves away from the reference point performing a global search, as described in Equations (12) and (13) [44]:

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{x}_{\text{rand}} - \mathbf{x}| \quad (12)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{\text{rand}} - \mathbf{A} \cdot \mathbf{D} \quad (13)$$

where \mathbf{x}_{rand} is a random position vector in the current population.

In WOA, the parameter \mathbf{A} is used to smoothly switch between the exploration and exploitation phases. In addition, the parameter p controls the alternation between the two types of whale movement “spiral or circular movement” [40]. The pseudocode Algorithm 3 presents the steps of the WOA.

Algorithm 3 Whale Optimization Algorithm-Pseudocode

```

1: Initialize the whale population  $x_i$  ( $i = 1, 2, \dots, n$ )
2: Calculate the fitness of each search agent
3:  $x_{best} = \text{the best search agent}$ 
4: for  $t = 1$  : maximum number of iterations do
5:   if ( $p < 0.5$ ) then
6:     if ( $|A| < 1$ ) then
7:       Update the position of the current search agent by Equation (6)
8:     else
9:       if ( $|A| \geq 1$ ) then
10:        Select random search agent ( $x_{rand}$ )
11:        Update the position of the current search agent by Equation (13)
12:      end if
13:    end if
14:  else
15:    if ( $p \geq 0.5$ ) then
16:      Update the position of the current search agent by (10)
17:    end if
18:  end if
19:   $t = t + 1$ 
20: end for
21: Check if any search agent goes beyond the search space and amend it
22: Calculate the Fitness of each search agent
23: Update  $x_{best}$  if there is a better solution
24: return  $x_{best}$ 

```

Adapted from [40]

3. Gaussian PID Controller

Throughout history, several methods for control have been introduced [45,46]. However, none of them have had greater prominence than the traditional Proportional, Integral and Derivative (PID) controller, which was created in the 1940s and is now used by most industries due to its efficiency and robustness and well-known design methodologies [47].

Although there are more efficient techniques than PID, often the complexity of implementation and adjustment is higher. One such technique involves adaptive control and adapts the system parameters according to its state. To improve the performance of the traditional PID controller, Kaster [48] proposed an adaptive controller based on Gaussian functions. In this case, the PID gains are adapted according to Gaussian functions of the input error. It is a smooth function with smooth derivatives, which also provides smooth transitions for the gains. This concept avoids abrupt gain transitions, a characteristic of some types of adaptive controllers, which may cause repetitive chattering in the gains if the error tends to be in the vicinity of the transition threshold leading to stressing the controller and shortening its lifetime.

The Gaussian is an uniform function with two output thresholds: one for large values of the input signal and the other for small values. Its concavity is adjustable and allows to establish how closed or opened a Gaussian curve must be [19]. The function is defined in Equation (14):

$$g(\varepsilon) = \lambda_1 - (\lambda_1 - \lambda_0)e^{-q\varepsilon^2} \quad (14)$$

where λ_0 and λ_1 are the upper and lower limits of the function and q defines the degree of concavity, as seen in Figure 1. The curve is facing up or down depending on who is greater, λ_0 or λ_1 .

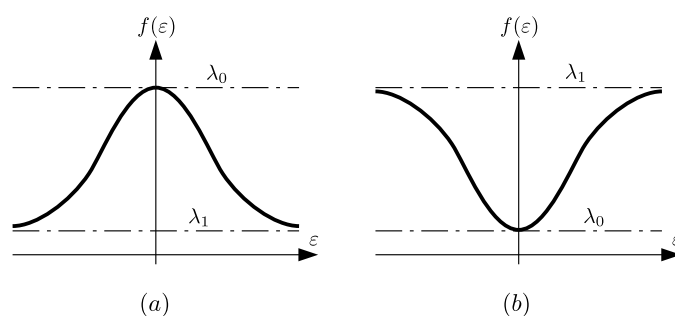


Figure 1. Gaussian function when (a) $\lambda_1 < \lambda_0$ (b) $\lambda_1 > \lambda_0$.

The controller has three Gaussian functions, one for each adaptive PID gain. It is reasonable to consider the lower bound of parameter λ_0 of the derivative gain as zero, which results a null derivative gain when the system operates at steady-state, helping to avoid noise amplification issues in the control action due to the derivative component. Then, the overall problem comprises eight parameters to be defined.

Puchta et al. [17] reported the use of a new set of parameters, referred to as *Linked Parameters* that lower the rank of the parameters by tying the adaptive parameters to the linear PID gains. Then, the parameters have the same design references as a previously designed PID but enhance its performance. This results in a six-parameter problem. Here, the linked rule is modified by putting parameters x , y , and z in log scale, which is better if, during the optimization process the Gaussian function needs to revert its concavity. The linked parameters are defined in Equations (15)–(19).

$$\lambda_{1P} = \exp(x) \times K_P \quad (15)$$

$$\lambda_{0P} = \exp(-x) \times K_P \quad (16)$$

$$\lambda_{1I} = \exp(y) \times K_I \quad (17)$$

$$\lambda_{0I} = \exp(-y) \times K_I \quad (18)$$

$$\lambda_{1D} = \exp(z) \times K_D \quad (19)$$

where K_P , K_I , and K_D are the PID reference gains.

The agent vector turns into $vec = \{x, y, z, q_P, q_I, q_D\}$.

4. Buck Converter

The DC–DC Buck converter is a typical power electronic converter topology employed in power supplies for most electronic equipment, like computers, TVs, mobile phones, and many others [45,49]. The Buck converter is characterized as a controllable step-down converter (Figure 2) that can achieve a regulated output voltage for a wide range of higher input voltages.

The regulation is achieved by modulating the switching of MOSFET Q, that works in conjunction with the diode, feeding a pulsed current to the low pass filter LC , resulting in the regulated output. The converter control design usually considers an average load for its most common application, as shown in the first circuit of Figure 2. This represents a second-order system with state-space Equations (20) and (21).

$$\frac{di_L(t)}{dt} = \frac{1}{L} (-v_o(t) + u(t)V_i) \quad (20)$$

$$\frac{dv_o(t)}{dt} = \frac{1}{C} \left(i_L(t) - \frac{1}{R} v_o(t) \right) \quad (21)$$

In order to verify the effectiveness of the designed control, it is quite common for power converter designers to use a typical nonlinear load attached in place of the original resistive load, as shown in the second circuit of Figure 2. This load is made up of an input inductance, which represents the primary coil of a transformer, a full bridge rectifier, and

an RC load at the very end. The final circuit turns into a fourth-order system represented by Equations (22)–(25).

$$\frac{di_L(t)}{dt} = \frac{1}{L}(-v_o(t) + u(t)V_i) \tag{22}$$

$$\frac{dv_o(t)}{dt} = \frac{1}{C} \left(i_L(t) - \frac{1}{R}v_o(t) \right) \tag{23}$$

$$\frac{di_o(t)}{dt} = \frac{1}{L_2}(v_o(t) - v_{C2}(t) - 2V_D) \tag{24}$$

$$\frac{dv_{C2}(t)}{dt} = \frac{1}{C_2} \left(i_o(t) - \frac{1}{R}v_{C2}(t) \right) \tag{25}$$

where V_D is the diode threshold voltage. In this sense, the optimized GAPID controller is expected to perform well in both situations.

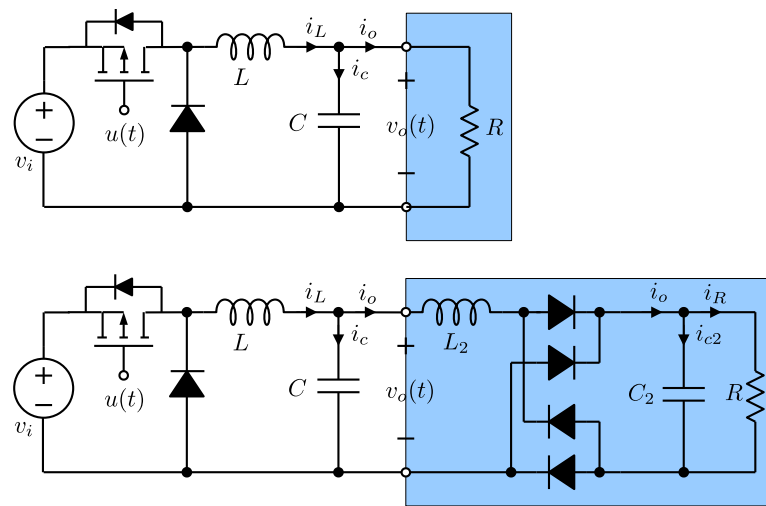


Figure 2. The converter’s schematic and control system with a resistive load (above) and nonlinear load (below).

As stated before, the GAPID employs linked parameters that depend upon a previously designed PID. In this work, the PID was designed using the pole-placement procedure with the transfer function of the second-order system defined in Equation (26).

$$\frac{V_o(s)}{U(s)} = \frac{V_i}{LCs^2 + \frac{L}{R}s + 1} \tag{26}$$

This second-order system has a natural oscillation frequency $\omega_n = \frac{1}{\sqrt{LC}}$ and dumping factor $\zeta = \frac{1}{2R} \sqrt{\frac{L}{C}}$.

The converter must be designed taking into account a high efficiency, a dumping factor preferably no less than 0.7, and the smallest inductor possible, which depends on the switching frequency, in order to generate an output with a low ripple. The inductor can be reduced as the switching frequency increases; however, the high efficiency imposes a limit on the switching frequency due to the switching losses.

In this work, a power supply for specific communication equipment with a regulated 30 V was considered. Taking all the above remarks into account, the converter’s design parameters are shown in Table 1.

Table 1. The Buck converter parameters.

Description	Symbol	Value
Input voltage	V_i	48 V
Output voltage	V_o	30 V
Converter Capacitor	C	10 μ F
Converter Inductor	L	1.2 mH
Load Resistance	R	15 Ω
Load Capacitor	C_2	4.7 μ F
Load Inductor	L_2	0.5 mH

This converter has a PID controller with reasonable performance, but it can be enhanced by using a GAPID controller instead.

5. Computational Results

5.1. Performance Evaluation Metric and Coding

The literature presents many metrics to evaluate the quality of the response of a control system. In this work, we address the Integral Absolute Error (IAE), following the premises defined by previous investigations, which must be minimized [17,20]. The IAE integrates the absolute value of the error over time according to Equation (27):

$$IAE = \int_0^{\infty} |\varepsilon(t)| dt \quad (27)$$

The IAE is used as the fitness function of the three bio-inspired optimizers. In addition, as mentioned in Section 3, the coding of the agents were made using vectors with real values, with the elements $vec = \{x, y, z, q_P, q_I, q_D\}$.

5.2. Buck Converter Response to a Linear PID

The design of the standard PID controller was based on the pole placement method [47]. The gains obtained after the adjustment were $K_p = 6.2 \times 10^{-3}$, $K_i = 35.0$, and $K_d = 0.7752 \times 10^{-6}$. The output responses (current, voltage, control signal, and instant error) are shown in Figure 3.

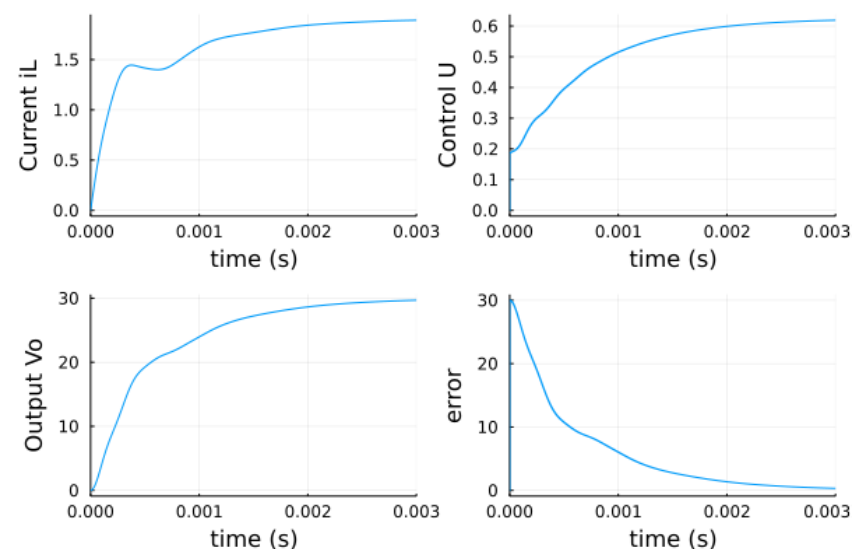


Figure 3. Output responses of the Buck converter controlled by the linear PID: inductor current i_L , output voltage V_o , control signal U , and instant error.

Although the results of the output responses were within the limits established in the design, we desired to investigate how an adaptive controller could improve the system performance. Thus, bio-inspired optimization techniques were used to find the gains of the Gaussian function used by GAPID, as shown next.

5.3. Optimization by Bioinspired Algorithms

The optimization algorithms were designed to search the parameters of the Gaussian PID controller gains, programmed to run 50 iterations per execution, with a population of 80 search agents, and were executed 100 times, each randomly initialized, for comparison. The optimization goal was to find adequate values for the GAPID parameters: λ_{p0} , λ_{p1} , q_p , λ_{i0} , λ_{i1} , q_i , λ_{d1} , and q_d . In this sense, due to the linked parameters approach, each agent encoding for the metaheuristics appliance is a vector containing six elements.

The results obtained on the output of the Buck converter with the GAPID controller and the performance of each bio-inspired algorithm are presented next. We highlight that all parameters of the algorithms were defined empirically.

5.3.1. Parameter Optimization Using PSO

The PSO algorithm was configured with the cognitive coefficient $k_1 = 2.0$, social coefficient $k_2 = 2.0$, and inertia constant $\omega = 0.9$. Thus, the fitness evolution over time for the best round was performed as shown in the blue line in Figure 4. The best fitness value achieved with 100 independent executions was 2.7394×10^{-3} .

The output of the Buck converter is presented in Figures 5 and 6 shows the variation of the adaptive gains of the controller obtained by PSO.

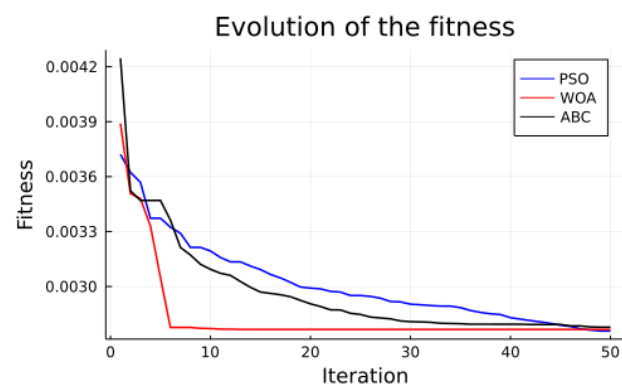


Figure 4. The fitness evolution for the PSO, WOA, and ABC algorithms.

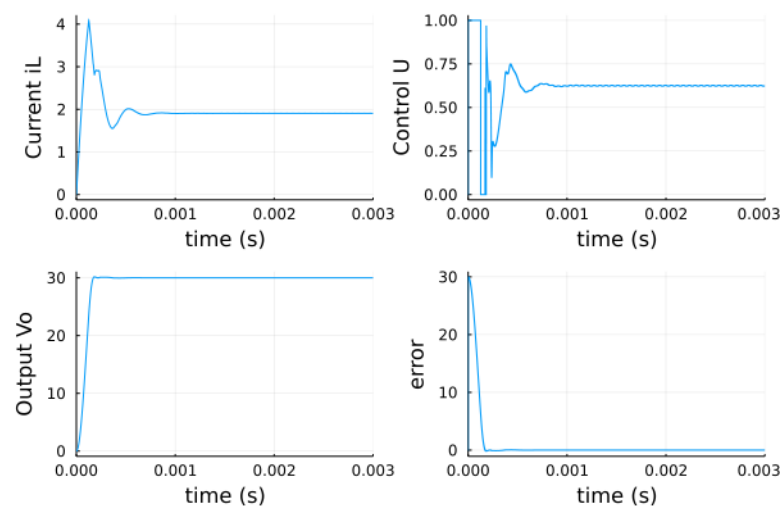


Figure 5. The output response of the Buck converter controlled by GAPID and optimized by PSO: inductor current i_L , output voltage V_o , control signal U , and instant error.

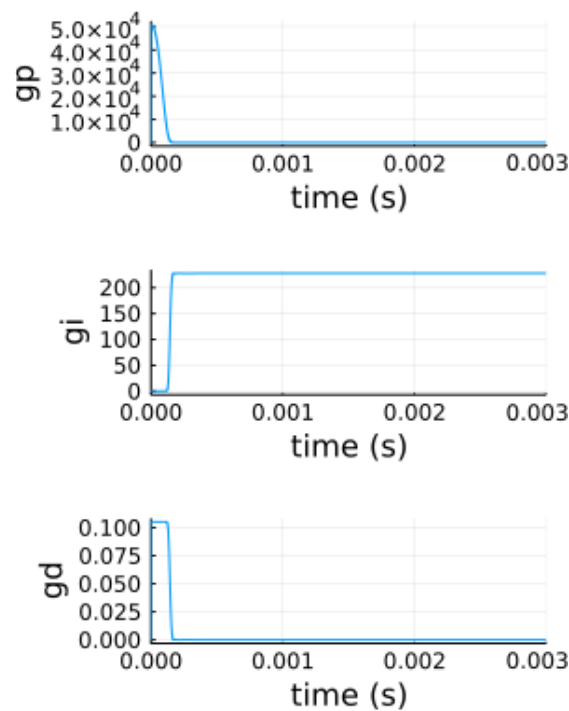


Figure 6. Variation of the adaptive gains of the controller (obtained by PSO optimization) during the initial transition.

5.3.2. Parameters Optimization Using ABC

The fitness evolution over time for the best round was performed as shown in the black line in Figure 4. The best fitness found was 2.7497×10^{-3} .

At the end of the optimization process, the aforementioned parameters were used to perform the parameterization of the GAPID, generating the Buck output waveforms, as shown in Figure 7. Figure 8 shows the variation of the adaptive gains of the controller obtained by ABC.

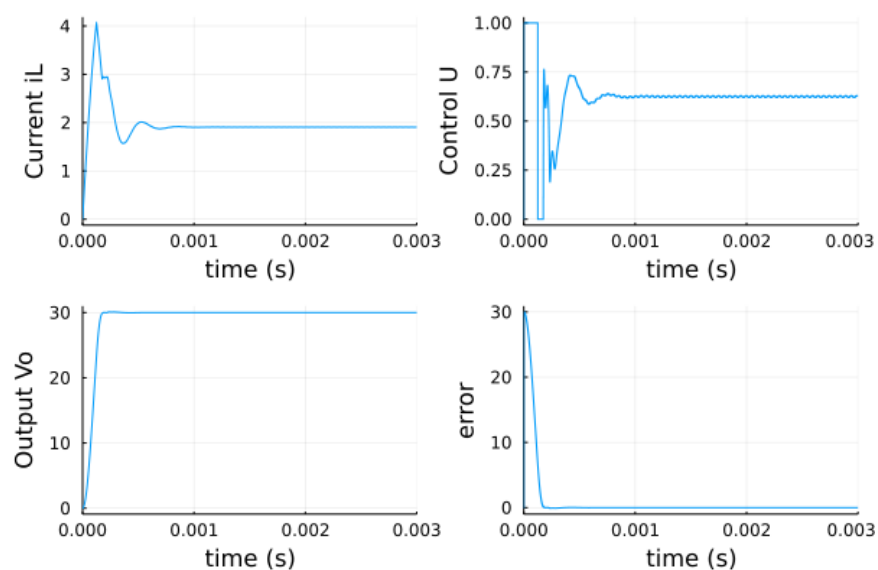


Figure 7. Output response of the Buck converter controlled by GAPID optimized by ABC: inductor current i_L , output voltage V_o , control signal U , and instant error.

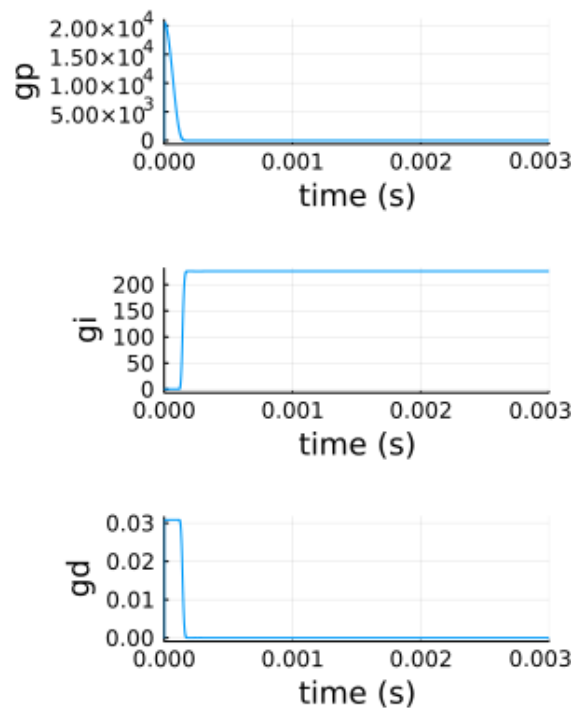


Figure 8. Variation of the adaptive gains of the controller (obtained by ABC optimization) during the initial transition.

5.3.3. Parameter Optimization Using WOA

The WOA algorithm does not use any external parameters; the movement is calculated within the algorithm itself. Thus, the fitness evolution over time for the best round was performed as shown in the red line in Figure 4. The smallest error (fitness value) found was 2.7385×10^{-3} . The Buck's output is shown in Figure 9, while Figure 10 presents the variation of the adaptive gains of the controller obtained by ABC.

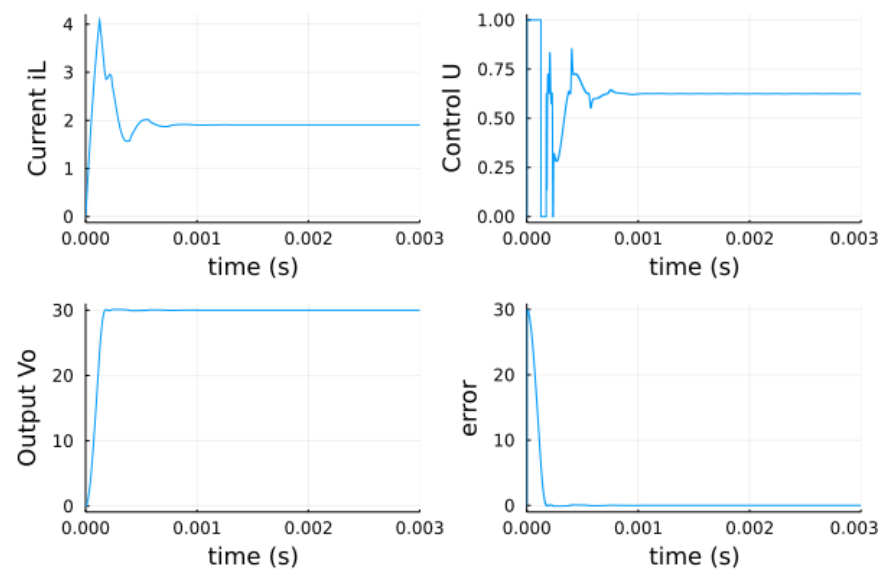


Figure 9. The output response of the Buck converter controlled by GAPID optimized by WOA: inductor current i_L , output voltage V_o , control signal U , and instant error.

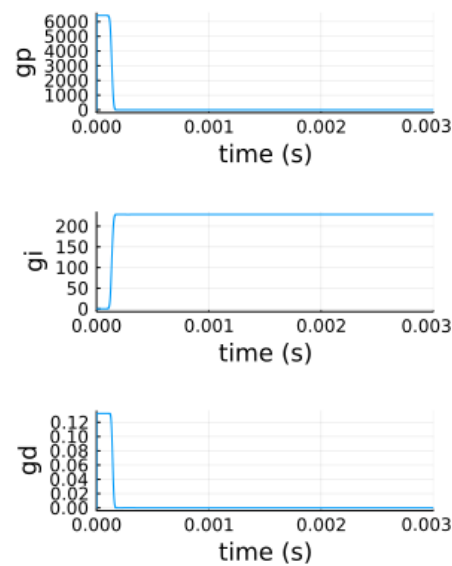


Figure 10. Variation of the adaptive gains of the controller (obtained by WOA optimization) during the initial transition.

5.4. Analysis of the Results

The experimental results obtained during the adjustment of the GAPID controller using metaheuristics allow for some important considerations. Initially, observing Figure 3, the control action U of PID is quite slow. As stated before, the GAPID gains are referred to the PID gains, as linked parameters. As expected, for optimized GAPID, the control action is considerably more aggressive, causing saturation during startup, but does not provoke overshoot in the output signal while obtaining a fast response, mainly because it is a nonlinear controller. This is similar for all optimization strategies (PSO, ABC, and WOA), as shown in Figures 5, 7 and 9. However, in these figures, a high current peak is noticeable, which may be undesirable in certain applications. Fortunately, such high current peaks occur only during the startup.

The objective of optimization relies on obtaining a faster performance of the output voltage with a low overshoot, guaranteed by the fitness evaluation of IAE. Figure 11 summarizes the output voltage waveforms achieved by PID and GAPID tuned by the three different algorithms, where there a small difference in the GAPID responses and only a slight overshoot for PSO, which does not happen for ABC and WOA. As seen in the figure, the superiority of GAPID over PID is evident, showing a settling time about five-times shorter. The graph in Figure 12 proves the adaptability of the GAPID in maintaining better responses for both loads.

The comparison of the final values of the parameters achieved by the three metaheuristics is presented in Table 2.

Table 2. The best GAPID control parameters for each algorithm.

Parameters	PSO	ABC	WOA
λ_{p0}	0.000541	3.93096	0.246775
λ_{p1}	14.077712	1.02262e5	6419.73
q_p	0.1	0.000249032	0.0384613
λ_{i0}	208.560244	225.662	228.545
λ_{i1}	0.170253	0.184214	0.186567
q_i	0.01	0.09	0.0374831
λ_{d1}	0.000570	0.0308419	0.132381
q_d	0.1	0.09	0.0882509

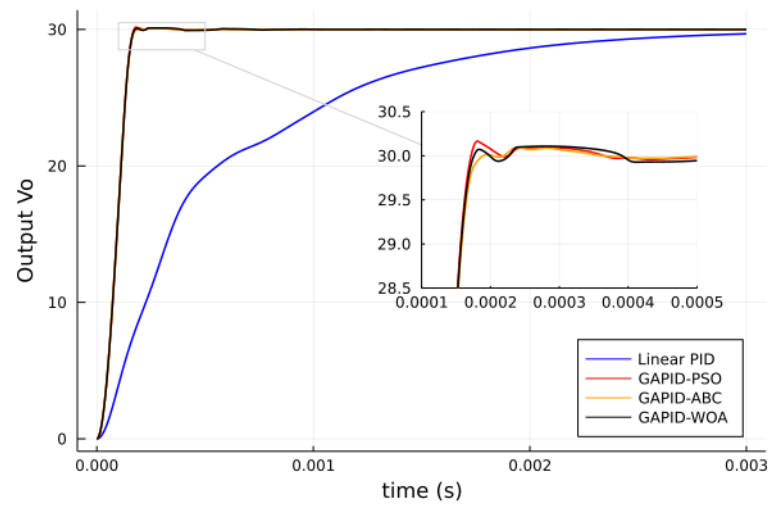


Figure 11. The converter output voltage V_o for each control strategy.

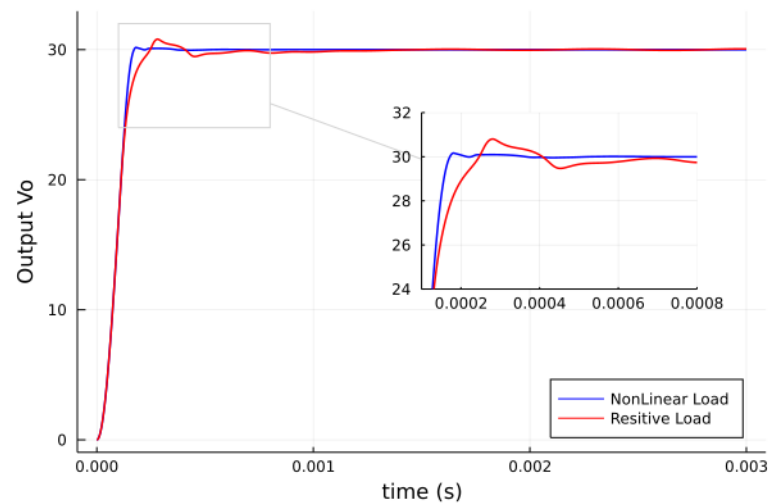


Figure 12. The output responses for the resistive load and for the nonlinear load.

As mentioned, linked parameters were used to determine the best coefficients of the GAPID. In this sense, Table 2 reveals that the best performances were achieved for different parameter values. This is strong evidence that the cost function based on the IAE is multimodal, which makes the search process difficult.

In relation to the performances in terms of the fitness achieved by each optimization method, a statistical description of the results is presented in Table 3 in which the term “Best” means the smallest fitness reached by the three optimizers.

Table 3. Statistical description of the results.

Results	PSO	ABC	WOA
Best	2.7394×10^{-3}	2.7497×10^{-3}	2.7385×10^{-3}
Mean	2.7414×10^{-3}	2.8274×10^{-3}	2.7745×10^{-3}
Standard Deviation	7.1413×10^{-7}	1.0098×10^{-4}	2.2668×10^{-3}

To determine if the results were significantly distinct, two statistical tests were applied considering the IAE of the 100 independent simulations. Friedman’s test [50] indicated a p-value equal to 1.0541×10^{-32} . The pairwise Wilcoxon test [7] revealed the following:

- PSO \times ABC: p -value = 2.5621×10^{-34} ;

- PSO \times WOA: p -value = 1.6116×10^{-30} ;
- ABC \times WOA: p -value = 6.1727×10^{-7} .

In this sense, considering a significance of 95%, one can infer that a change in the optimization algorithm led to different results. Therefore, we can admit that the PSO achieved the best general performance, followed by the WOA in terms of the best performance, smallest average error, and smallest standard deviation of the 100 simulations. Figure 13 presents the boxplot graphic of the final values of the fitness and corroborates this observation. We also highlight that the PSO presented the smallest dispersion.

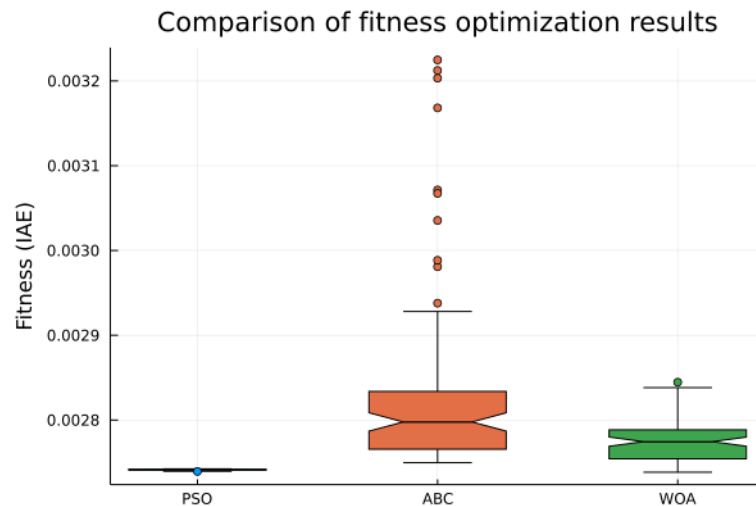


Figure 13. Boxplot of the fitness considering 100 independent experiments.

A boxplot graphic was also elaborated in relation to the number of iterations until the convergence of the algorithms considering 100 independent executions, as can be seen in Figure 14. The best PSO converged in the 21st iteration, while WOA and ABC were close to the 30th. Considering the number of iterations until convergence, PSO used 42 iterations, while ABC used 50, and WOA used 47.

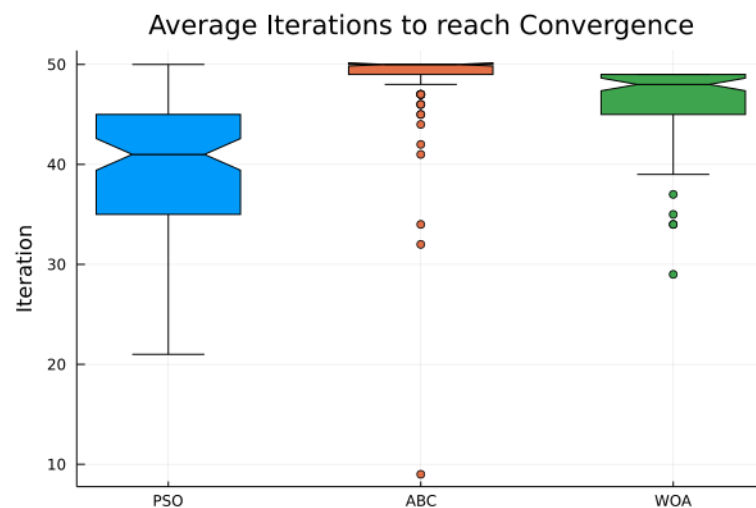


Figure 14. Boxplot of the number of iterations to reach convergence.

The PSO converged faster than the others, and presented the smallest dispersion. This can be particularly important in industrial applications, where online training may be required, since the computational effort necessary to adjust the model is smaller than the others.

The Figure 11 features a zoom-in that represents the converter output, explaining the difference between each algorithm at the moment when the system reaches the steady state. In the zoom area, a small overshoot can be seen on the curves of PSO and WOA, while the ABC has a smoother curve. In Figure 11, it can be observed more clearly that the three optimized GAPID outputs reached the steady state in less time than the linear PID, once again confirming the effectiveness of the optimization.

5.5. Future Research

Beyond the interesting results found in this investigation, future work can be developed using other bio-inspired optimization methods, such as genetic algorithms, differential evolution, and cat swarm optimization, among others. As a multimodal problem, the main challenge is to select one of the near-best solutions that better fits a robust operation of the plant when subjected to parameter variations and disturbances. One can simultaneously address other error metrics, by means of multi-objective approaches. In this sense, the NSGA II and SPEA appear as natural candidates. These approaches can help in the case of this multimodal problem, in attempting to select the best solution by considering additional objectives. Other plants beyond the buck should be controlled to evaluate the general behavior of the GAPID, especially those with nonlinear behavior.

6. Conclusions

This study presented an investigation on the use of three bio-inspired optimization algorithms to determine the gains of an adaptive PID Gaussian controller (GAPID): Particle Swarm Optimization (PSO), the Artificial Bee Colony (ABC) algorithm, and the Whale Optimization Algorithm (WOA). The Gaussian functions are smooth and bounded allowing for tuning the gains without abrupt changes with defined limits and imposing adaptive behavior on the controller. Since there is no algebraic methodology to determine these gains, metaheuristics were proposed to solve the task.

The gains resulting from the optimization were constantly tested and evaluated in the Buck converter, which demonstrated that the adaptive control strategy was robust and reliable. The experimental results revealed the effectiveness and usability of the optimization methods for the proposed problem. All of them were able to exceed the performance of the linear PID. Among them, PSO achieved the best general result with faster convergence.

Author Contributions: Conceptualization, E.D.P.P., M.d.S.K. and H.V.S.; methodology, E.D.P.P., M.d.S.K. and H.V.S.; software, E.D.P.P., L.H.B. and M.A.I.F.; validation, E.D.P.P., L.H.B. and M.A.I.F.; formal analysis, E.D.P.P., M.d.S.K. and H.V.S.; investigation, E.D.P.P., P.B. and A.C.; resources, M.d.S.K. and H.V.S.; data curation, M.d.S.K. and H.V.S.; writing—original draft preparation, E.D.P.P., P.B. and L.H.B.; writing—review and editing, M.A.I.F., A.C., M.d.S.K. and H.V.S.; visualization, E.D.P.P.; supervision, M.d.S.K. and H.V.S.; project administration, A.C., M.d.S.K. and H.V.S.; funding acquisition, A.C. and H.V.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors also thank the Federal University of Technology-Parana for the financial support.

Acknowledgments: The authors thank the Brazilian agencies Coordination for the Improvement of Higher Education Personnel (CAPES)-Financing Code 001, Brazilian National Council for Scientific and Technological Development (CNPq), processes number 40558/2018-5, and 315298/2020-0, and Araucaria Foundation, process number 51497, for their support.

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

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony
PSO	Particle Swarm Optimization
WOA	Whale Optimization Algorithm
PID	Proportional, Integral and Derivative
GAPID	Gaussian Adaptive PID

References

- Gong, Y.J.; Chen, W.N.; Zhan, Z.H.; Zhang, J.; Li, Y.; Zhang, Q.; Li, J.J. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Appl. Soft Comput.* **2015**, *34*, 286–300. [\[CrossRef\]](#)
- Figueiredo, E.; Macedo, M.; Siqueira, H.V.; Santana, C.J., Jr.; Gokhale, A.; Bastos-Filho, C.J. Swarm intelligence for clustering—A systematic review with new perspectives on data mining. *Eng. Appl. Artif. Intell.* **2019**, *82*, 313–329. [\[CrossRef\]](#)
- de Souza Tadano, Y.; Siqueira, H.V.; Alves, T.A. Unorganized machines to predict hospital admissions for respiratory diseases. In Proceedings of the 2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Cartagena, Spain, 2–4 November 2016; pp. 1–6.
- Miao, Z.; Yong, P.; Mei, Y.; Quanjun, Y.; Xu, X. A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. *Future Gener. Comput. Syst.* **2021**, *115*, 497–516. [\[CrossRef\]](#)
- Santos, P.; Macedo, M.; Figueiredo, E.; Santana, C.J.; Soares, F.; Siqueira, H.; Maciel, A.; Gokhale, A.; Bastos-Filho, C.J. Application of PSO-based clustering algorithms on educational databases. In Proceedings of the 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Arequipa, Peru, 8–10 November 2017; pp. 1–6.
- Siqueira, H.; Figueiredo, E.; Macedo, M.; Santana, C.J.; Santos, P.; Bastos-Filho, C.J.; Gokhale, A.A. Double-swarm binary Particle Swarm Optimization. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
- Santana, C.J., Jr.; Macedo, M.; Siqueira, H.; Gokhale, A.; Bastos-Filho, C.J. A novel binary Artificial Bee Colony algorithm. *Future Gener. Comput. Syst.* **2019**, *98*, 180–196. [\[CrossRef\]](#)
- Devi, S.; Geethanjali, M. Application of modified bacterial foraging optimization algorithm for optimal placement and sizing of distributed generation. *Expert Syst. Appl.* **2014**, *41*, 2772–2781. [\[CrossRef\]](#)
- Siqueira, H.; Figueiredo, E.; Macedo, M.; Santana, C.J.; Bastos-Filho, C.J.; Gokhale, A.A. Boolean binary cat swarm optimization algorithm. In Proceedings of the 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Guadalajara, Mexico, 7–9 November 2018; pp. 1–6.
- Siqueira, H.; Santana, C.; Macedo, M.; Figueiredo, E.; Gokhale, A.; Bastos-Filho, C. Simplified binary cat swarm optimization. *Integr. Comput. Aided Eng.* **2020**, *28*, 35–50. [\[CrossRef\]](#)
- Tang, J.; Ma, Y.; Guan, J.; Yan, C. A max–min ant system for the split delivery weighted vehicle routing problem. *Expert Syst. Appl.* **2013**, *40*, 7468–7477. [\[CrossRef\]](#)
- Siqueira, H.; Macedo, M.; Tadano, Y.D.S.; Alves, T.A.; Stevan, S.L.; Oliveira, D.S.; Marinho, M.H.; Neto, P.S.; de Oliveira, J.F.; Luna, I.; et al. Selection of temporal lags for predicting riverflow series from hydroelectric plants using variable selection methods. *Energies* **2020**, *13*, 4236. [\[CrossRef\]](#)
- Santana, C.J.; Bastos-Filho, C.J.; Macedo, M.; Siqueira, H. SBFSS: Simplified binary fish school search. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 2595–2602.
- Purushothama, G.; Jenkins, L. Simulated annealing with local search—a hybrid algorithm for unit commitment. *IEEE Trans. Power Syst.* **2003**, *18*, 273–278. [\[CrossRef\]](#)
- Gholizadeh, S.; Danesh, M.; Gheytratmand, C. A new Newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames. *Comput. Struct.* **2020**, *234*, 106250. [\[CrossRef\]](#)
- Belotti, J.T.; Castanho, D.S.; Araujo, L.N.; da Silva, L.V.; Alves, T.A.; Tadano, Y.S.; Stevan, S.L., Jr.; Corrêa, F.C.; Siqueira, H.V. Air pollution epidemiology: A simplified Generalized Linear Model approach optimized by bio-inspired metaheuristics. *Environ. Res.* **2020**, *191*, 110106. [\[CrossRef\]](#)
- Puchta, E.D.P.; Lucas, R.; Ferreira, F.R.V.; Siqueira, H.V.; Kaster, M.S. Gaussian adaptive PID control optimized via genetic algorithm applied to a step-down DC-DC converter. In Proceedings of the 12th IEEE International Conference on Industry Applications (INDUSCON 2016), Curitiba, Brazil, 20–23 November 2016. [\[CrossRef\]](#)
- Borges, F.G.; Itaborahy, M.A.; Siqueira, H.V.; Correa, F.C.; Kaster, M.S. Comparison of several Genetic Algorithm strategies on a nonlinear GAPID controller optimization applied to a Buck converter. In Proceedings of the Congresso Brasileiro de Automática, Sociedade Brasileira de Automática, João Pessoa, Brazil, 9–12 September 2018.
- Borges, F.G.; Monteiro, L.F.; Martins, M.S.R.; Correa, F.C.; Siqueira, H.V.; Kaster, M.S. Performance Comparison of Particle Swarm optimization Strategies to Adjust a Nonlinear GAPID Controller. In Proceedings of the 2018 13th IEEE International Conference on Industry Applications (INDUSCON), Sao Paulo, Brazil, 12–14 November 2018; pp. 685–691.
- Puchta, E.D.P.; Siqueira, H.V.; dos Santos Kaster, M. Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller. *IEEE Trans. Cybern.* **2019**, *50*, 1185–1194. [\[CrossRef\]](#)

21. Savran, A.; Kahraman, G. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. *ISA Trans.* **2014**, *53*, 280–288. [\[CrossRef\]](#)
22. Anderson, K.; Blankenship, G.; Lebow, L. A rule-based adaptive PID controller. In Proceedings of the 27th IEEE Conference on Decision and Control, Austin, TX, USA, 7–9 December 1988; Volume 1, pp. 564–569. [\[CrossRef\]](#)
23. Benaskeur, A.; Desbiens, A. Backstepping-based adaptive PID control. *IEE Proc. Control. Theory Appl.* **2002**, *149*, 54–59. [\[CrossRef\]](#)
24. Poulin, É.; Pomerleau, A.; Desbiens, A.; Hodouin, D. Development and evaluation of an auto-tuning and adaptive PID controller. *Automatica* **1996**, *32*, 71–82. [\[CrossRef\]](#)
25. Radke, F.; Isermann, R. A parameter-adaptive PID-controller with stepwise parameter optimization. *Automatica* **1987**, *23*, 449–457. [\[CrossRef\]](#)
26. Delavari, H.; Ghaderi, R.; Ranjbar, N.A.; HosseinNia, S.H.; Momani, S. Adaptive Fractional PID Controller for Robot Manipulator. In Proceedings of the FDA'10 the 4th IFAC Workshop Fractional Differentiation and Its Applications, Badajoz, Spain, 18–20 October 2010.
27. Lin, C.M.; Lin, M.H.; Chen, C.W. SoPC-Based Adaptive PID Control System Design for Magnetic Levitation System. *IEEE Syst. J.* **2011**, *5*, 278–287. [\[CrossRef\]](#)
28. Srivastava, V.; Srivastava, S. Whale Optimization algorithm (WOA) based control of nonlinear systems. In Proceedings of the 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 18–19 October 2019; pp. 402–406.
29. Kumar, A.A.; Kumar, S.G. Application of Whale Optimization Algorithm for tuning of a PID controller for a drilling machine. In Proceedings of the ICAARS 2018, Coimbatore, India, 14–15 December 2018.
30. Fan, Q.; Chen, Z.; Li, Z.; Xia, Z.; Yu, J.; Wang, D. A new improved Whale Optimization algorithm with joint search mechanisms for high-dimensional global optimization problems. *Eng. Comput.* **2020**, 1–28. [\[CrossRef\]](#)
31. Niknam, T. A new fuzzy adaptive hybrid Particle Swarm Optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. *Appl. Energy* **2010**, *87*, 327–339. [\[CrossRef\]](#)
32. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
33. Siqueira, H.; Belotti, J.T.; Boccato, L.; Luna, I.; Attux, R.; Lyra, C. Recursive linear models optimized by bioinspired metaheuristics to streamflow time series prediction. *Int. Trans. Oper. Res.* **2021**. [\[CrossRef\]](#)
34. Elloumi, W.; El Abed, H.; Abraham, A.; Alimi, A.M. A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. *Appl. Soft Comput.* **2014**, *25*, 234–241. [\[CrossRef\]](#)
35. Xiao, G.; Juan, Z.; Gao, J. Travel mode detection based on neural networks and particle swarm optimization. *Information* **2015**, *6*, 522–535. [\[CrossRef\]](#)
36. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report; Citeseer: University Park, PA, USA, 2005.
37. Awadallah, M.A.; Al-Betar, M.A.; Bolaji, A.L.; Alsukhni, E.M.; Al-Zoubi, H. Natural selection methods for Artificial Bee Colony with new versions of onlooker bee. *Soft Comput.* **2019**, *23*, 6455–6494. [\[CrossRef\]](#)
38. Karaboga, D.; Akay, B. A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [\[CrossRef\]](#)
39. Hussain, S.F.; Pervez, A.; Hussain, M. Co-clustering optimization using Artificial Bee Colony (ABC) algorithm. *Appl. Soft Comput.* **2020**, *97*, 106725. [\[CrossRef\]](#)
40. Mirjalili, S.; Lewis, A. The Whale Optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
41. Watkins, W.A.; Schevill, W.E. Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis*, *Balaenoptera borealis*, *Megaptera novaeangliae*, and *Balaenoptera physalus*. *J. Mammal.* **1979**, *60*, 155–163. [\[CrossRef\]](#)
42. Goldbogen, J.A.; Friedlaender, A.S.; Calambokidis, J.; Mckenna, M.F.; Simon, M.; Nowacek, D.P. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience* **2013**, *63*, 90–100. [\[CrossRef\]](#)
43. Tharwat, A.; Moemen, Y.S.; Hassanien, A.E. Classification of toxicity effects of biotransformed hepatic drugs using whale optimized support vector machines. *J. Biomed. Inform.* **2017**, *68*, 132–149. [\[CrossRef\]](#)
44. McCann, S.; Lowe, D.G. Local naive bayes nearest neighbor for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3650–3656.
45. Soriano-Sánchez, A.G.; Rodríguez-Licea, M.A.; Pérez-Pinal, F.J.; Vázquez-López, J.A. Fractional-order approximation and synthesis of a PID controller for a buck converter. *Energies* **2020**, *13*, 629. [\[CrossRef\]](#)
46. Ali, N.; Liu, Z.; Armghan, H.; Ahmad, I.; Hou, Y. LCC-S-Based Integral Terminal Sliding Mode Controller for a Hybrid Energy Storage System Using a Wireless Power System. *Energies* **2021**, *14*, 1693. [\[CrossRef\]](#)
47. Ogata, K. *Modern Control Engineering*; Prentice Hall: Englewood Cliffs, NJ, USA, 2010.
48. Kaster, M.; Cunha, G.; Burgardt, I.; Nascimento, C. Análise do uso de controle pid não linear aplicado a conversor buck. In Proceedings of the X Conferência Brasileira de Dinâmica, Controle e Aplicações, Aguas de Londoia, Brazil, 29 August–2 September 2011.

-
49. Yfoulis, C.; Papadopoulou, S.; Voutetakis, S. Robust Linear Control of Boost and Buck-Boost DC-DC Converters in Micro-Grids with Constant Power Loads. *Energies* **2020**, *13*, 4829. [[CrossRef](#)]
 50. Siqueira, H.; Luna, I. Performance comparison of feedforward neural networks applied to streamflow series forecasting. *Math. Eng. Sci. Aerosp. (MESA)* **2019**, *10*, 41–53.