

Article

Nature-Inspired Metaheuristic Optimization for Control Tuning of Complex Systems

Jesús Garicano-Mena ^{1,2,*}  and Matilde Santos ^{3,*} ¹ ETSI Aeronáutica y del Espacio—Universidad Politécnica de Madrid, 28040 Madrid, Spain² Center for Computational Simulation (CCS), 28660 Boadilla del Monte, Spain³ Institute of Knowledge Technology, Complutense University of Madrid, 28040 Madrid, Spain* Correspondence: jesus.garicano.mena@upm.es (J.G.-M.); msantos@ucm.es (M.S.)

† These authors contributed equally to this work.

Abstract: In this contribution, a methodology for the optimal tuning of controllers of complex systems based on meta-heuristic techniques is proposed. Two bio-inspired meta-heuristic optimization algorithms—the Antlion Optimizer (ALO) and the Whale Optimization Algorithm (WOA)—have been applied to two different dynamic systems: the Hoop & Ball electromechanical system, a system where a linearized description is adequate; and to a Wind Turbine–Generator–Rectifier, as an example of a complex non-linear dynamic system. The performance of the ALO and WOA techniques for the tuning of conventional PID controllers is evaluated in relation to the number of agents n_s and the maximum number of iterations $n_{MaxIter}$; given the stochastic nature of both methods, repeatability is also addressed. Finally, the computational effort required for their implementation is considered. By analyzing the obtained metrics, it is observed that both methods provide comparable results for the two systems considered and, therefore, the ALO and WOA techniques can complement each other by exploiting the advantages of each of them in controller tuning.

Keywords: Metaheuristics Algorithm; Whale Optimization Algorithm (WOA); Antlion Optimization Algorithm (ALO); complex dynamics systems; Hoop & Ball electromechanical system; wind energy conversion system



Academic Editor: Heming Jia

Received: 15 November 2024

Revised: 18 December 2024

Accepted: 23 December 2024

Published: 30 December 2024

Citation: Garicano-Mena, J.; Santos, M. Nature-Inspired Metaheuristic Optimization for Control Tuning of Complex Systems. *Biomimetics* **2025**, *10*, 13. <https://doi.org/10.3390/biomimetics10010013>

Copyright: © 2024 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/>).

1. Introduction

Most real systems, especially in the industrial and engineering fields, do not require very sophisticated control strategies to achieve good performance. However, to achieve the required specifications of precision, minimum control effort, energy savings, etc., it is essential that the regulators are well designed and configured [1].

The task of adjusting regulators is a laborious and non-trivial job. Hence, numerous techniques have been applied for the optimal adjustment of their parameters, and it remains an open topic for research since the requirements are increasingly demanding [2].

Furthermore, thanks to technological and computational advances, it has been possible to address the control of increasingly complex, non-linear systems, with coupled variables and disturbances. But this requires a greater effort in the design and adjustment of the controllers.

In this work, this challenge is addressed by applying two meta-heuristic techniques for the optimal adjustment of a conventional controller for two real systems.

The conventional controller is a Proportional–Integral–Derivative (or *PID*), which is possibly the most widespread feedback control strategy in industry [3]. The *PID* control approach offers, in its most simplified setting [4], a control action $u(t)$ determined as the

weighted superposition of an error measure $e(t) \equiv y_{ref}(t) - y(t)$, its temporal integral and its derivative, namely:

$$u(t) = K_p e(t) + T_i \int e(\tau) d\tau + T_d \frac{de(t)}{dt}. \quad (1)$$

This simple formulation is sufficiently flexible to, upon proper setting (i.e., *tuning*) of the parameters (K_p, T_i, T_d) , provide satisfactory results in many situations. Note that, on many occasions, there might be many parameter combinations that lead to solutions of similar merit [5].

There are many strategies to find those parameters (K_p, T_i, T_d) , e.g., the now classical Ziegler–Nichols heuristic method, or the pole placement approach, for instance; these methods work specially well for systems that can be accurately described with low order models, and wherever noise is not a concern [6].

More recently, more sophisticated, *intelligent* methods have been introduced, and have been applied not only to tackle new control challenges, but also to revisit the *PID* tuning problem [7–9]. Among these intelligent techniques, the so-called metaheuristic optimization algorithms have risen as a promising strategy [10]. Genetic Algorithms (*GA*) and Particle Swarm Optimization (*PSO*) algorithms are well known methods of this class, that happen to be inspired by biological processes [11–14]. In this sense, neither *GA* nor *PSO* are a novelty: a plethora of nature-inspired metaheuristic optimization algorithms exists; a taxonomy for those is provided in [8].

Among those methods, in this work we focus on two recent and relatively more unknown nature-inspired metaheuristic optimization algorithms: the Antlion Optimizer (or *ALO* [15]), and the Whale Optimization Algorithm (or *WOA* [16]). The first one draws from the stalking behavior of antlions, whereas the second tries to mimick the cooperative hunting behavior of humpback and Bryde’s whales. Both methods evidence an effective performance on a wide array of non-linear optimization problems.

Specifically, this paper analyzes the performance of these two bio-inspired metaheuristic optimization algorithms for tuning proportional–integral controllers (*PI*, i.e., with $T_d = 0$ in Equation (1)). The objective is to see their applicability in the control of complex non-linear systems and their dependence on the algorithms initial and configuration parameters. To do so, their application to two different dynamic systems is compared: the electromechanical hoop & ball system (*HB*), as a system where a linearized description is adequate; and a wind turbine–generator–rectifier–load (*WTGRL*) system, as an example of a system with a complex and nonlinear dynamics. To our knowledge, these techniques have not been applied in the field of control of these systems. Our objective is to evaluate the applicability of the *ALO* and/or *WOA* techniques to efficiently tune the *PI* controllers using only information of feasible ranges for K_p and T_i gains.

The main contributions of this paper can be summarized as:

- i. a methodology is proposed for the automatic optimal tuning of conventional *PID* controllers;
- ii. this methodology uses two advanced metaheuristic optimization algorithms of biological inspiration, the techniques *ALO* and *WOA*;
- iii. the only information needed is the feasible ranges for K_p and T_i ;
- iv. the performance of both optimization techniques has been compared in two nonlinear dynamic systems of increasing complexity, the hoop and ball system and the wind turbine-generator-rectifier-load system;
- v. they have been compared with the solutions provided by other evolutionary techniques for the same hoop and ball system found in the literature;

$$\begin{aligned} [I_a + m(R - r)^2] \ddot{\theta} + b_m \dot{\theta} - m \frac{(R - r)^2}{R} \ddot{y} &= -mg(R - r) \sin(\psi) + \tau(t), \\ [I_b + m \frac{(R - r)^2}{R}] \ddot{y} + \frac{b_b}{r^2} \dot{y} - m \frac{(R - r)^2}{R} \ddot{\theta} &= mg \frac{R - r}{R} \sin(\psi), \end{aligned} \tag{2}$$

where R and r_b are respectively the hoop and ball radii, whereas $r < r_b$ is the ball rolling radius; and m is the ball mass. The inertia moments of hoop and ball are $I_a = \frac{1}{2} M R^2$ and $I_b = \frac{2}{5} M r_b^2$. A servomotor actions the hoop with a moment $\tau(t)$; this servomotor is affected by a friction b_m , whereas the rolling resistance of the ball is b_b . Finally, the relationship $\psi = \theta - \frac{y}{R}$ closes the system of equations.

The system (2) can be regard as a first order non-linear system by defining $x_1 \equiv \theta(t)$ and $x_3 \equiv y(t)$, so that $\dot{\theta} = \dot{x}_1 \equiv x_2$ and $\dot{y} = \dot{x}_3 \equiv x_4$. In this way, we obtain

$$\begin{aligned} A_\theta x_2 + B_\theta x_2 + C x_4 &= D_\theta \sin\left(\theta - \frac{y}{R}\right) + \tau(t), \\ C x_2 + B_y x_4 + A_y x_4 &= D_y \sin\left(\theta - \frac{y}{R}\right), \end{aligned} \tag{3}$$

Introducing now $\mathbf{x}(t) \equiv [x_1(t), x_2(t), x_3(t), x_4(t)]^T$, it is possible to reformulate Equation (3) as:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & A_\theta & 0 & C \\ 0 & 0 & 1 & 0 \\ 0 & C & 0 & A_y \end{bmatrix}}_{\mathcal{M}_0} \frac{d}{dt} \mathbf{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -B_\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -B_y \end{bmatrix}}_{\mathcal{A}_0} \mathbf{x}(t) + \underbrace{\begin{bmatrix} 0 \\ D_\theta \\ 0 \\ D_y \end{bmatrix}}_{\mathcal{NL}} \sin\left(\theta - \frac{y}{R}\right) + \underbrace{\begin{bmatrix} 0 \\ \tau \\ 0 \\ 0 \end{bmatrix}}_{\mathcal{F}_0}, \tag{4}$$

It can be also expressed in more compact form as:

$$\frac{d}{dt} \mathbf{x}(t) = \mathcal{M}_0^{-1} \mathcal{A}_0 \mathbf{x}(t) + \mathcal{M}_0^{-1} \mathcal{NL} + \mathcal{M}_0^{-1} \mathcal{F}_0. \tag{5}$$

In this expression, it is possible to identify, on the right side, a linear contribution, another non-linear contribution proportional to $\sin\left(\theta - \frac{y}{R}\right)$, and the forcing term.

Under the assumption that $r \ll R$ and $\psi \ll 1$, $\sin\left(\theta - \frac{y}{R}\right) \sim \theta - \frac{y}{R}$. Thus, in the linearized model, the servomotor would act directly the hoop, which in turn excites the ball movement. The specific parameters of the HB model considered in this work are: $g = 9.81 \text{ m/s}^2$, $R = 0.085 \text{ m}$, $M = 0.10 \text{ kg}$, $m = 0.02812 \text{ kg}$, $r = 0.0095 \text{ m}$, $r_b = 0.009525 \text{ m}$, $b_m = 0.1 \text{ kg m}^2/\text{s}$ and $b_b = 3.22 \times 10^{-6} \text{ kg m}^3/\text{s}$.

In order to discretize the HB model, the matlab command ode45 was used. The hoop axis is controlled by a Proportional Integral (PI) controller, where the control signal is generated with feedback of the signal θ :

$$u(t) = K_p \left(\theta(t) - \theta_{ref}(t)\right) + T_i \int \left(\theta(\tau) - \theta_{ref}(\tau)\right) d\tau, \tag{6}$$

where $\theta_{ref}(t)$ is a sequence of step signals defined for $t \in [0, 20]s$ (see e.g., Figure 2c).

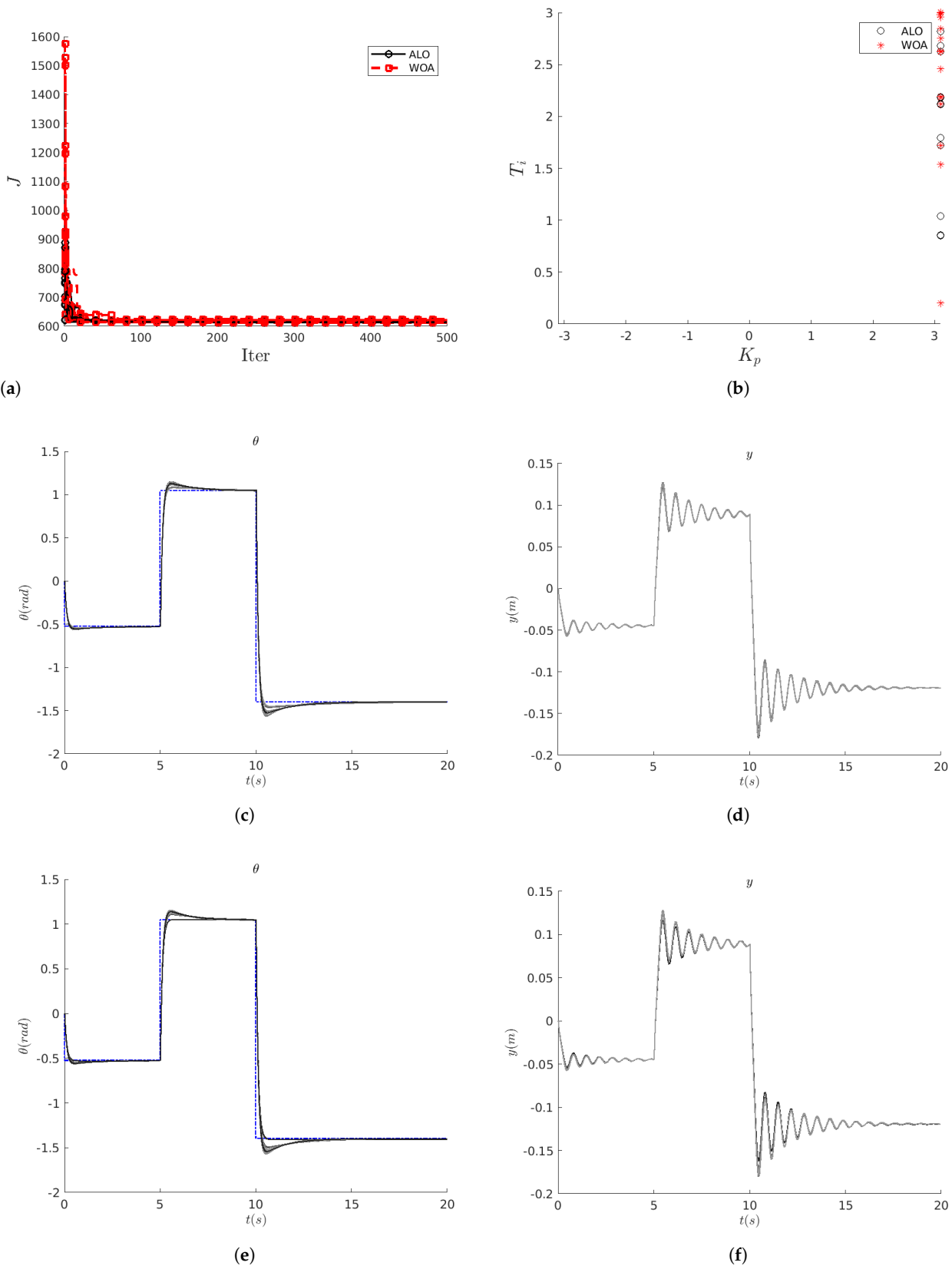


Figure 2. HB system: in (a), evolution of the cost function; in (b), different solutions found by the optimization algorithms ALO (o) and WOA (*); in (c,d), variability of the system solutions identified by ALO; in (e,f) the same for WOA.

The tuning of the K_p and T_i gain is going to be carried out using the ALO and WOA optimization techniques, with the aim of minimizing the Squared Error norm $\|\theta(t) - \theta_{ref}(t)\|_2^2$. The search space is defined for $|K_p| < 3.1$ and $T_i \in [0, 3]$ [20].

2.2. The Wind Turbine–Generator–Rectifier–Load System

Wind energy conversion systems are a typical example of a complex system, involving many mechanical, hydraulic, electrical, and electronic (possibly combined) subsystems that interact to reach a common goal, i.e., extract power from the momentum in a wind stream [21]. Therefore, modeling such systems can easily become complicated [22,23]. Regarding the goals of this contribution—extracting useful knowledge from the comparison of the ALO and WOA algorithms in the context of intelligent control of complex systems—we consider a relatively simple yet sufficiently rich model, inspired in those discussed in [24–27].

In this wind turbine–generator–rectifier–load system, a 12.3 kW three-bladed horizontal axis wind turbine is considered. It drives directly (i.e., no gearbox) a three-phase salient pole permanent magnet synchronous electric generator. This generator excites a rectifier module that enacts the AC/DC conversion; the rectifier, in turn, feeds through a DC-DC boost converter a resistive load of $R_L = 30 \Omega$.

We consider that the turbine is exposed to several wind gusts, Figure 3, acting in the interval $t \in [0, 9]$ s. The rectifier subsystem follows a Maximum Power Point Tracking (MPPT) strategy, whereas in the operating regions of rated wind speed, a Proportional–Integral (PI) controller adjusts the blade angle of attack (β). The angle of attack is constrained to remain in the range $\beta \in [0^\circ, 15^\circ]$, and is also affected by a Rate Limiter.

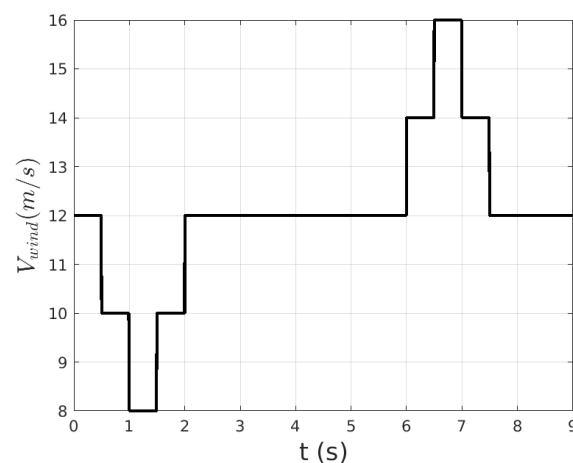


Figure 3. WTGRL system: wind gusts affecting the wind turbine subsystem.

In our model we use powergui: the simulation is performed in discrete mode ($T_s = 20 \times 10^{-6}$ s or $f_s = 50$ kHz), so that the z-transfer function of the PI controller is:

$$C(z) = K_p + T_s T_i \frac{1}{z - 1}. \tag{7}$$

The simulink model for the WTGRL system is shown in Figure 4.

The tuning values of K_p and T_i will be obtained with the two nature-inspired meta-heuristic optimization algorithms (ALO and WOA). The optimization goal, in this case, is to dissipate as much power at the load R_L as possible. This power is given by,

$$P_{DC,0} = V_{DC,0} I_{DC,0}. \tag{8}$$

Thus, the fitness function J is defined as:

$$J = - \sum_{l=1}^{n_{timeSteps}} (P_{DC,0}(t = t_l))^2, \tag{9}$$

which is proportional to the discretized integral, quantifying the power dissipated at the load $\int_0^9 (P_{DC,0}(\tau))^2 d\tau$. We shall explore the search space limited by $|K_p| < 5000$ and $T_i \in [0, 60,000]$ [26,27].

2.3. Metaheuristic Optimization Algorithms

In this work we consider two metaheuristic optimization algorithms, the *Antlion Optimization Algorithm* (ALO) and the *Whale Optimization Algorithm* (WOA), both coined by Prof. S. Mirjalili’s research group. The biological phenomena that inspire these algorithms are illustrated in [28] and [29], respectively.

As metaheuristic algorithms, ALO and WOA present these advantages:

- i they are formulated on simple principles and allow for an easy implementation;
- ii the computing of the objective function gradient is not necessary;
- iii they can avoid eventual local optima; and
- iv they can be applied in many diverse fields.

The implementation of either algorithm is publicly available; and since they share a common interface, so once a script is coded for one of the techniques, it can be easily reused with any other similar algorithm.

2.3.1. The Antlion Optimization Algorithm

We provide here a brief description of the ALO technique, initially presented in [15]; the corresponding pseudo-code is reproduced in Algorithm 1 [30]. As a flock/swarm type algorithm, ALO employs n_S ant agents, but also n_S antlion agents within a d -dimensional space. The ants move through the space following a random walk in all their coordinates. On the other hand, antlions build *sand pit traps*, whose sizes are larger the lower the value of the objective function (i.e., fitness) at a specific location. The movement of the ants must respect the constraints set by the lower \mathbf{b}_l and upper \mathbf{b}_h coordinate vectors. This can be achieved with a random walk given by:

$$A^k = [0, \text{cumsum}(2r(t_1) - 1), \dots, \text{cumsum}(2r(t_{IterMax}) - 1)], \tag{10}$$

if the function $r(t)$ is defined in terms of a uniform random variable $Z \sim U[0, 1]$, so that $r(t) = 1$ if $z \in (0.5, 1]$ and 0 elsewhere. We ensure that all the agents remain in the search space $[\mathbf{b}_l, \mathbf{b}_h]$ by resorting to the normalization:

$$A_i^k = c_i^k + \frac{d_i^k - c_i^k}{b_i - a_i} (A_i^k - a_i), \tag{11}$$

where a_i and b_i are the minimum and maximum of the random walk for variable i , and c_i^k y d_i^k are, respectively, the minimum and the maximum of the i -th variable at the k -th step.

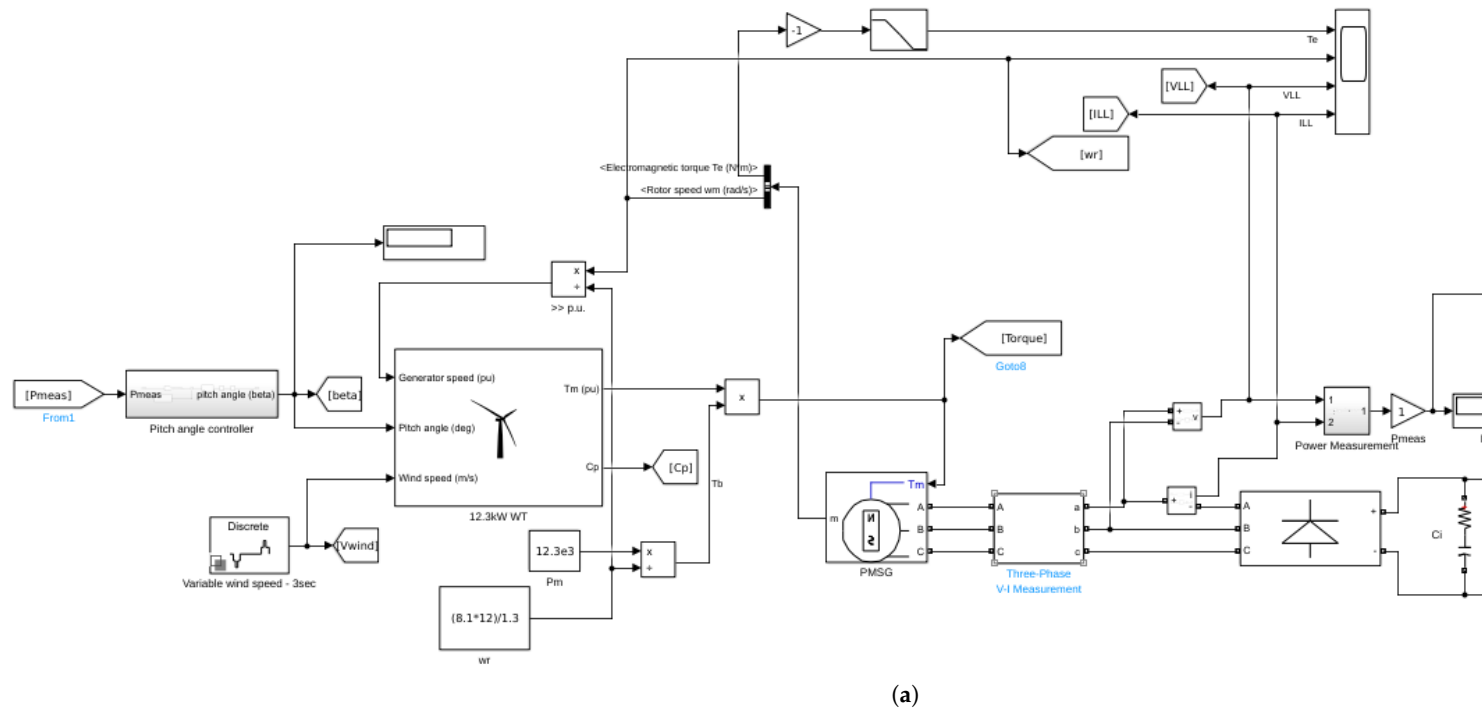


Figure 4. Cont.

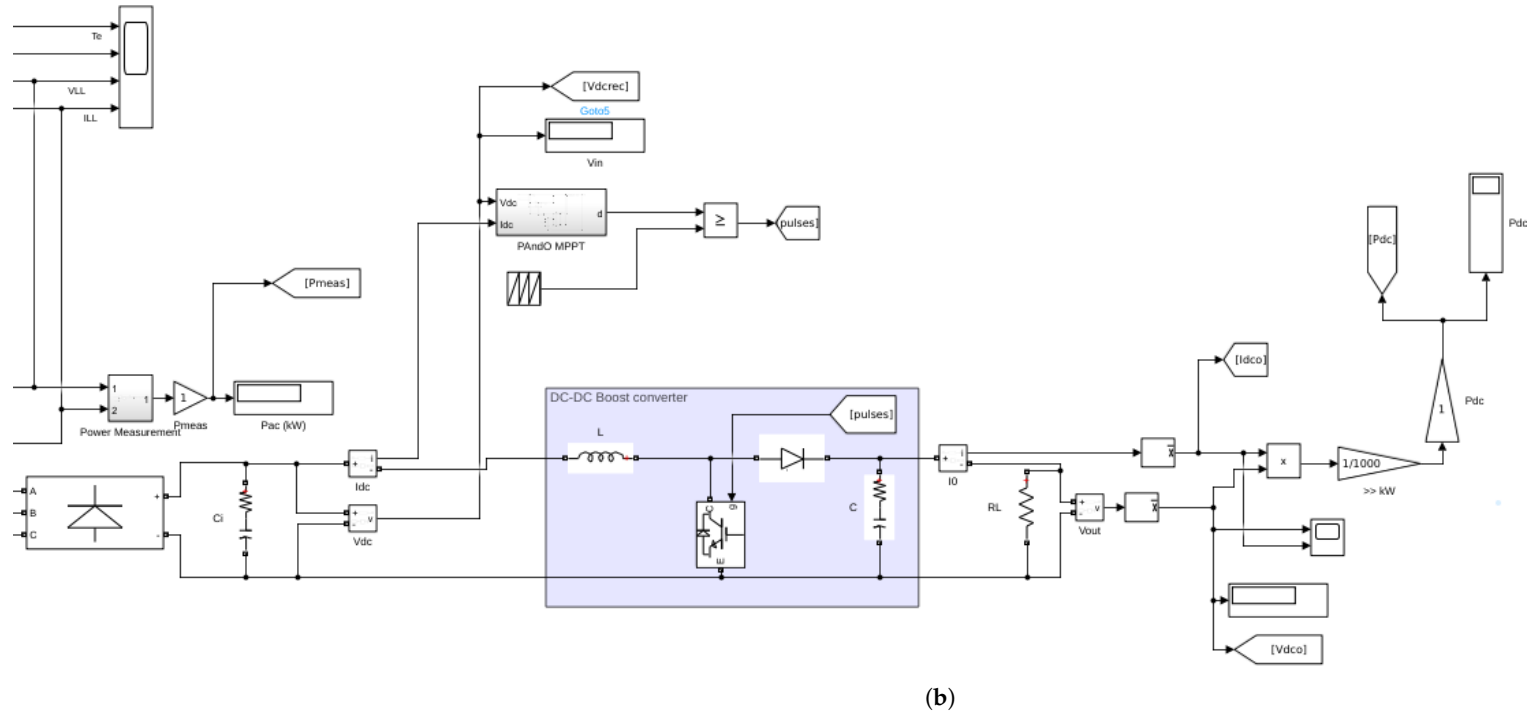


Figure 4. WTGRL system: simulink model of the wind turbine—generator—load with *PI* control for the angle of attack β and MPPT control for the rectifier subsystem. The wind turbine, the generator and the rectifier subsystems are shown in (a); the rectifier and the DC-DC boost converter are shown in (b).

The ALO algorithm starts by randomly distributing the n_s ants and the n_s antlions over the feasible solution space. Next, the best antlion \mathbf{AL}^* is identified, namely, the antlion that $\min_i \mathbf{f}(\mathbf{AL}_i(k=0))$. Then, for the number of iterations chosen a priori, $n_{MaxIter}$, the ants wander throughout the search space, whereas the antlions attempt to hunt them down. This guarantees the exploration of the search space. On the other hand, the exploitation of the areas of interest is guaranteed by the progressive contraction of antlion sand pit traps, which can be modeled as:

$$c^k = \frac{c^k}{I}, \quad y \quad d^k = \frac{d^k}{I}, \quad (12)$$

where I is a compression ratio, and c^k/d^k are the minimum/maximum of all the variables in the k -th iteration.

Note how the combination of the ant random walks with the roulette wheel selection of the antlions contributes to avoid, with high probability, stagnating into local minima.

Even more, the random walk of each ant along every dimension fosters the diversity of the agents involved. And since the sand pit traps relocate to the position of the best ant found during the optimization, the promising areas of the search space are preserved.

Finally, elitism is applied, since the best antlion at every iteration is stored and compared against the best antlion so far (the elite). Algorithmically, ants roam randomly both around the antlion chosen by roulette wheel selection but also around the elite antlion:

$$A_i^k = \frac{R_A^k + R_E^k}{2}, \quad (13)$$

where R_A^k denotes the trajectory around the antlion active in the current iteration, whereas R_E^k is the trajectory around the elite antlion.

The detailed procedure described above can be formalized in the successive application of three operators that can be found in [15].

Algorithm 1 The Antlion Optimization Algorithm [15].

```

1: Algoritmo ALO( $n_s, n_{MaxIter}, \mathbf{b}_l, \mathbf{b}_h, d, \mathbf{f}$ )
2:   Random initialization of ants  $\{\mathbf{A}_i\}_{i=1}^{n_s}$  and antlions  $\{\mathbf{AL}_i\}_{i=1}^{n_s}$ .
3:   Calculate the fitness of ants,  $\mathbf{M}_{O,A}(\mathbf{X}_i)$  and antlions  $\mathbf{M}_{O,AL}(\mathbf{X}_i)$ .
4:   Identify the best antlion  $\mathbf{AL}^*$  (elite).
5:   while  $k \leq n_{MaxIter}$  do
6:     for  $i = 1, \dots, n_s$  do
7:       Choose an antlion using roulette wheel.
8:       Update  $c$  and  $d$  with Equation (12).
9:       Create a random walk (Equation (10)) and normalize it (Equation (11)).
10:      Update the  $i$ -th ant position (Equation (13)).
11:    end for
12:    Recalculate the fitness of every ant.
13:    Replace an antlion with the corresponding ant, provided it becomes fitter.
14:    Update the elite agent if another antlion becomes better.
15:  end while
16:  return elite
17: end Algoritmo

```

2.3.2. The Whale Optimization Algorithm

A brief description of the WOA technique with the corresponding pseudo-code is reproduced here as in Algorithm 2 and [31]. WOA considers a pod (or herd) of n_s whales (or agents or particles), that evolve throughout a d -dimensional space in search of prey, i.e.,

the local and/or global minima of the function f to be optimized. The whales movement remains in the interior of the cartesian product defined by the corresponding components of the lower \mathbf{b}_l and upper \mathbf{b}_h constrain vectors.

The algorithm begins by randomly distributing the n_s whales over the feasible solution space; then, the best solution so far, \mathbf{X}^* , is singled out, namely, that one fulfilling $\min_i f(\mathbf{X}_i(k=0))$.

For a pre-set number of iterations $n_{MaxIter}$, the whale positions are updated, alternating phases of:

- i prey search (exploration or diversification),
- ii prey encircling, and
- iii bubble net attacking (exploitation or intensification).

At every iteration k , and for every whale, a random number $p_i(k) \sim U[0, 1]$ is considered. If $p_i(k) < 0.5$, and depending on the magnitude of a vector \mathbf{A} , the method alternates between the circling phase –if $|\mathbf{A}| < 1$ – or the search of prey phase –when $|\mathbf{A}| \geq 1$.

The circling phase is described by equations:

$$\mathbf{D}_i = |\mathbf{C}_i \odot \mathbf{X}^*(k) - \mathbf{X}_i(k)|, \tag{14}$$

$$\mathbf{X}_i(k+1) = \mathbf{X}^*(k) - \mathbf{A}_i \odot \mathbf{D}_i,$$

where $\mathbf{A} = 2\mathbf{a} \odot \mathbf{r} - \mathbf{a}$ y $\mathbf{C} = 2\mathbf{r}$, with \mathbf{r} is a vector with random components, and \mathbf{a} is a vector whose components shrink linearly with k from 2 to 0. These equations describe the global effect of having the best-positioned whale so far (i.e., with a lower f) to *warn* her mates of how promising her current location is.

The exploratory phase, namely the search of preys, is controlled by equation:

$$\mathbf{X}_i(k+1) = \mathbf{X}_{rand} - \mathbf{A}_i \odot \mathbf{D}_i, \tag{15}$$

but here \mathbf{D}_i is given by:

$$\mathbf{D}_i = |\mathbf{C}_i \odot \mathbf{X}_{rand}(k) - \mathbf{X}_i(k)|. \tag{16}$$

On the other side, whenever $p_i(k) \geq 0.5$, the hunting phase is active, and the agents position is updated according to:

$$\mathbf{X}_i(k+1) = \begin{cases} \mathbf{X}^*(k) - \mathbf{A}_i \odot \mathbf{D}_i, & \text{si } q_i(k) < 0.5 \\ |\mathbf{X}^*(k) - \mathbf{X}_i(k)| e^{bl} \cos(2\pi l) + \mathbf{X}_{rand}, & \text{si } q_i(k) \geq 0.5, \end{cases}$$

where yet another random number $q_i(k) \sim U[0, 1]$ controls whether the whale encircles her prey, or whether she *decides* to approach the prey describing the famous spiral of decreasing radius.

Algorithm 2 Whale Optimization Algorithm [16].

- 1: **Algoritmo** WOA($n_S, n_{MaxIter}, \mathbf{b}_l, \mathbf{b}_h, d, \mathbf{f}$)
 - 2: Random whale positions initialization $\{\mathbf{X}_i\}_{i=1}^{n_S}$.
 - 3: Calculate the fitness of every whale, $f(\mathbf{X}_i)$.
 - 4: Identify the best whale \mathbf{X}^* .
 - 5: **while** $k \leq n_{MaxIter}$ **do**
 - 6: **for** $i = 1, \dots, n_S$ **do**
 - 7: Update $\mathbf{a}, \mathbf{A}, \mathbf{C}, l, p$ y q .
 - 8: **if** $p < 0.5$ **then**
-

Algorithm 2 Cont.

```

9:         if  $|A| < 1$  then
10:            Use Equation (14).
11:         else if  $|A| \geq 1$  then
12:            Choose a whale at random,  $X_{rand}(k)$ , and use Equation (15).
13:         end if
14:         else
15:            Use Equation (17).
16:         end if
17:         end for
18:         Identify whales that wandered beyond region  $[b_l, b_h]$ , and bring them back.
19:         Recalculate the fitness for every whale,  $f(X_i)$ .
20:         Update  $X^*$  if a better solution was found.
21:     end while
22: end Algorithmo

```

The methodology of the application of any of those metaheuristic optimization strategies to adjust the *PI* controller parameters is shown in the flowchart of Figure 5.

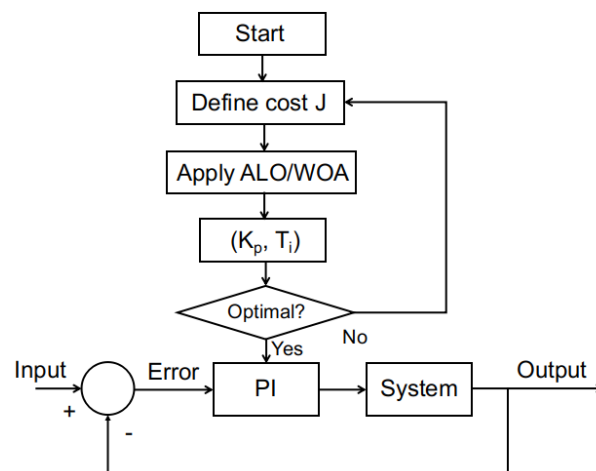


Figure 5. Flowchart of the methodology of PID tuning with ALO/WOA.

3. Discussion of the Results

The application of those two metaheuristic technique to the two selected systems has been carried out using a laptop computer equipped with an 4-core Intel(R) Core(TM) i5-3570K CPU at 3.40 GHz, a cache memory of 6144 kB and 12.0 GB of RAM, using the software Matlab R2023b. The corresponding results are shown and discussed in the next subsections.

3.1. PI Control of the Hoop and Ball System

In order to characterize the performance of both, the ALO and WOA techniques, for the tuning of *PI* controllers, we have carried out a sensibility analysis of the configuration parameters of those techniques, investigating the effects of the number of search agents n_S and the number of iterations $n_{MaxIter}$ along the exploration the search space. For the hoop & ball system, we have considered $n_S \in [3, 6, 12]$ and $n_{MaxIter} \in [125, 250, 500]$. We have also taken into account the random nature of the algorithms; thus, we have carried out $n_R = 20$ simulations for every combination $(n_S, n_{MaxIter})$ for both, ALO and WOA algorithms. Note that any other configuration parameter is set as recommended in [15,16]. The results of these experiments are gathered in Tables 1 and 2 and shown in Figure 6.

According to the data in Tables 1 and 2, it is possible to see how –in general– the more agents exploring for longer time the search space, the smaller (the better) is on average the objective function J . However, there are some exceptions at: $(n_S, n_{MaxIter}) = (12, 250)$

vs. $(n_S, n_{MaxIter}) = (12, 500)$ for ALO; this happens as well for WOA. Also for WOA and for the $n_{MaxIter} = 500$ cases, note how doubling the number of search agents n_S from 6 to 12 does not bring any significant improvement of the average value of the cost function J .

Also from Tables 1 and 2, a practically linear scaling of the average computation time is observed both with the number of search agents n_S and with the number of iterations $n_{MaxIter}$, for the ranges considered here.

The above analysis can be supplemented by inspection of Figure 6. The first observation is that the performance of ALO is indistinguishable from that of WOA on this problem. Moreover, quite often, solutions obtained with either method experience a significant decrease in the fitness function J at the beginning (in the first 20 iterations or so).

However, ALO solutions tend to decrease toward the minimum slightly faster than WOA solutions. In any case, and for this system, we observe how both techniques reach similar values for the cost function J . Note, however, that for some runs of the algorithm WOA, the final value of the cost J is slightly higher than both the other WOA solutions and the ALO solutions.

Table 1. HB system: optimal minimum reached \bar{J} and computational time needed \bar{t}_{Opt} (in seconds) for the ALO technique, averaged over $n_R = 20$ runs.

	$n_{MaxIter} = 125$			$n_{MaxIter} = 250$			$n_{MaxIter} = 500$		
	\bar{J}	\bar{t}_{Opt} [s]	\bar{J}	\bar{t}_{Opt} [s]	\bar{J}	\bar{t}_{Opt} [s]			
$n_S = 3$	617.4049	103.3	614.8049	205.8	614.4621	411.8			
$n_S = 6$	614.9633	206.0	614.4436	411.7	614.0940	825.1			
$n_S = 12$	613.8919	413.5	613.4070	825.0	613.9141	1650.6			

Table 2. HB system: optimal minimum reached \bar{J} and computational time needed \bar{t}_{Opt} for the WOA technique, averaged over $n_R = 20$ runs.

	$n_{MaxIter} = 125$			$n_{MaxIter} = 250$			$n_{MaxIter} = 500$		
	\bar{J}	\bar{t}_{Opt} [s]	\bar{J}	\bar{t}_{Opt} [s]	\bar{J}	\bar{t}_{Opt} [s]			
$n_S = 3$	618.9770	102.6	614.9905	205.1	615.8912	411.1			
$n_S = 6$	615.6602	205.4	614.8341	411.4	614.3943	825.7			
$n_S = 12$	615.2329	413.2	613.8167	826.7	614.2489	1652.4			

In view of the above results, further simulations have been performed using the combination $(n_S, n_{IterMax}) = (6, 500)$. Figure 2 shows a new set of results considering 10 new simulations (i.e., different seeds for the random number generator) for each metaheuristic optimization method.

The evolution of the cost function J (Figure 2a) is comparable to that shown in Figure 6. Figure 2b represents the solutions identified by ALO or WOA in the search space. It is evident that both methods tend to saturate the value of K_p . The values of T_i have more variability, although they are mainly concentrated in the range $T_i \in [2, 3]$. In Figure 2c,d, the outputs of the controlled system $(\theta(t) \text{ and } y(t))$ are shown for the different combinations of (K_p, T_i) identified by the ALO algorithm: in this case, the different solutions are quite close to each other. The same comments apply to the WOA algorithm (Figure 2e,f). However, it should be noted that one of the solutions proposed by WOA does not present overshoot. However, this solution does not correspond to the minimum J .

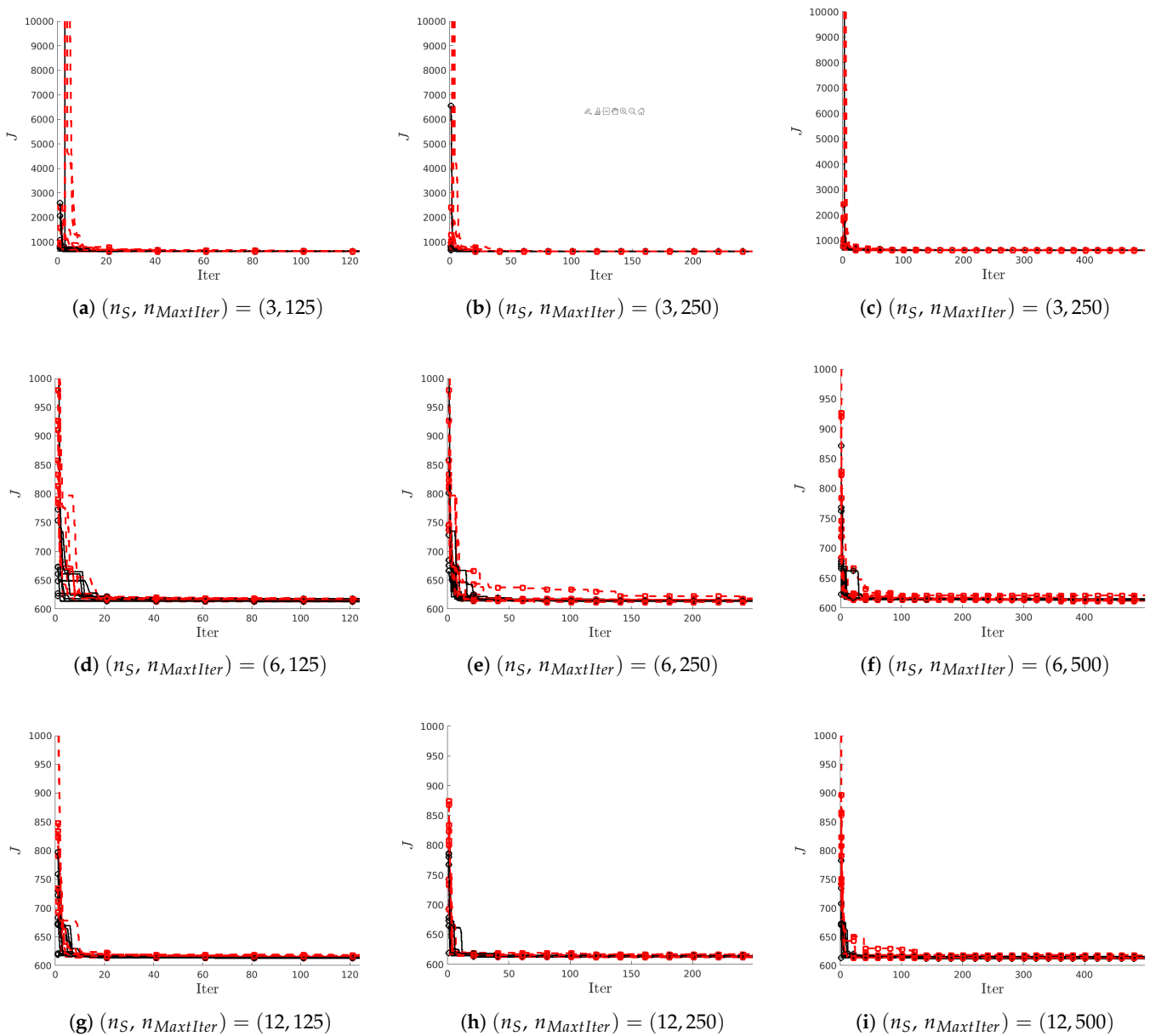


Figure 6. HB system: sensitivity analysis of convergence for different combinations of number of agents n_S and number of iterations $n_{MaxIter}$: $n_R = 20$ runs for AL0 (—○—) and WOA (—□—), respectively.

The best solutions found by the algorithm AL0 –are $(K_p, T_i) = (3.099, 2.122)$ with a cost value of $J = 611.788$ – and for WOA the optimized gains are $(K_p, T_i) = (3.100, 2.121)$, with a cost value of $J = 612.012$. These solutions are shown in Figure 7. Note how these solutions are visually indistinguishable.

The control solutions found by the optimization algorithms AL0/WOA for this HB system can be compared against the unit step response of the same HB system reported in [13], which are found using an Adaptive Hybrid PSO strategy based on the squared error. The solutions presented in this work have faster response (i.e., rise time of $y(t)$ of ≈ 0.23 s and settling time of ≈ 4.6 s vs., respectively, 0.84 s and 9.32 s in [13]) and a lower overshoot of 23.50% vs. 25.89% in [13], see Figure 8. Finally, it should be noted again that the other configuration parameters of the optimization algorithms are set as recommended in [15,16]. Therefore, the PI controller has been de facto tuned automatically, with no input information other than the allowed range in the space (K_p, T_i) .

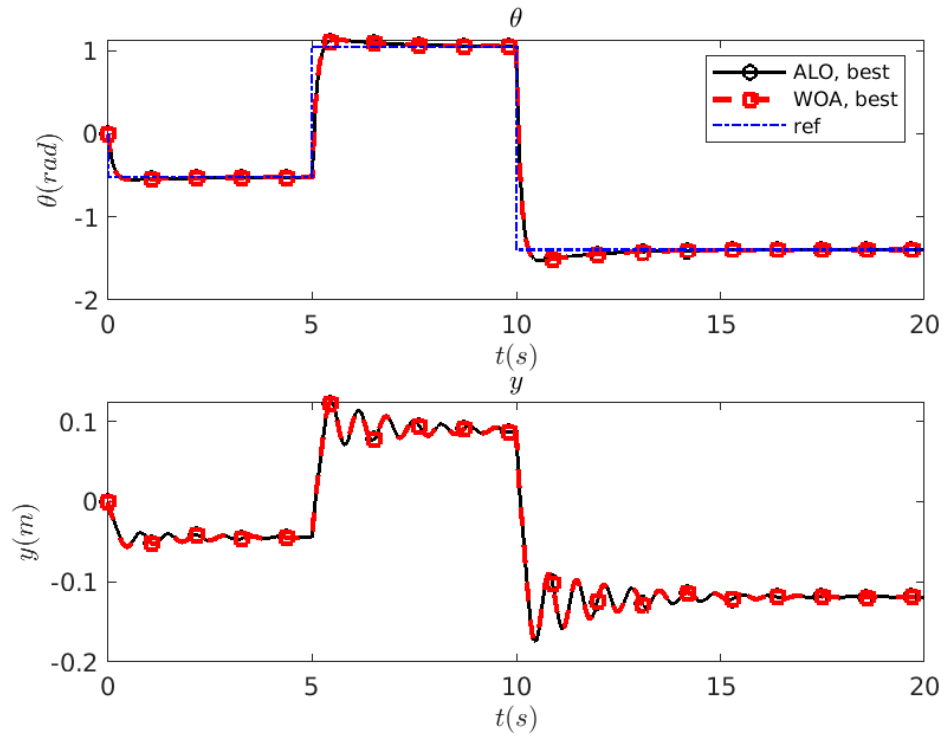


Figure 7. HB system: comparison of the optimal solutions identified by ALO (—○—) and WOA (—□—).

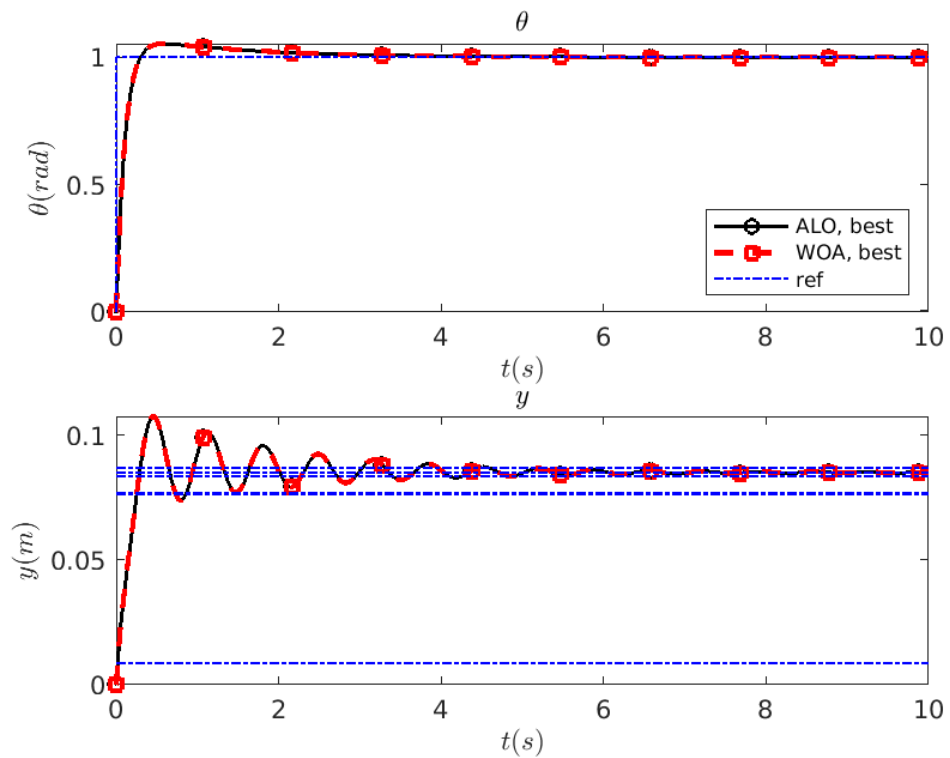


Figure 8. HB system: unit step response of the controlled system found by ALO (—○—) and WOA (—□—).

3.2. PI Control of Wind Turbine—Generator—Load System

In this subsection the results of the optimization of the PI controller for the angle of attack β of the WTGRL system with the two metaheuristic methods are presented and discussed.

The dynamic model of the WTGRL system is more complex than the one of the HB system. The WTGRL system includes several non-linear subsystems that interact together and are coupled. Additionally, the model uses powergui in a discretized setting, with a fairly high sampling frequency ($f_s = 50$ kHz), which is necessary for stability considerations. Therefore, the computational cost of running this model even once is much larger. And many runs are necessary to perform a single optimization/tuning with either ALO or WOA algorithms. This situation prevents an exhaustive exploration of the performance with different parameter combinations ($n_S, n_{MaxIter}$), as shown for the HB system. We have nevertheless carried out a preliminary sensitivity analysis of the main configuration parameters of the optimization algorithms, ($n_S, n_{MaxIter}$), that may have a bigger influence in the behaviour of the optimization. The results are summarized in Table 3. Note how both algorithms, WOA/ALO, tend to saturate the gain K_p . The results also show a linear scaling trend with the number of iterations and the number of search agents.

Table 3. WTGRL system, initial configuration study: computational time \bar{t}_{Opt} (in hours) and solution found (K_p, T_i) for both, the ALO and the WOA techniques, averaged over $n_R = 3$ runs.

	$(n_S, n_{MaxIter})$	$t_{Opt}(h)$	K_p	T_i
ALO	(6, 50)	4.1	−4991.1	9214.9
	(6, 100)	9.2	−4993.5	91.27
	(12, 50)	9.1	−870.3702	54,827
WOA	(6, 50)	4.9	−5000	578.99
	(6, 100)	10.2	−4109.7	75.14
	(12, 50)	9.3	3147.2	57,430

Next, a more exhaustive exploration of the space (K_p, T_i) has been performed, considering $(n_S, n_{MaxIter}) = (6, 500)$ and three simulations for each metaheuristic optimization technique. The results of this study are summarized in Table 4 and shown in Figure 9.

Given the data in Table 4, both ALO and WOA find a similar value for the cost function J on average. It can also be observed, as previously mentioned, the significant increase in the computation time required for an individual simulation, which goes from tens of minutes to more than a day.

Figure 9a shows the evolution of the fitness function J with the number of iterations for the WTGRL system. It is possible to see a significant decrease of J at the beginning. However, in this case, around 150 iterations are necessary to get close to the optimal value, and still appreciable decreases are observed at iteration ≈ 350 . Contrary to what is observed in Figure 2a for the HB system, in these cases the solutions of WOA tend to decrease faster than those of the ALO algorithm. Again, it is observed how both techniques reach similar values of the cost function J .

Figure 9b shows the location of the solutions found by the ALO or WOA algorithms in the search space. In this case, it is possible to see how both optimization algorithms, ALO and WOA, tend to saturate the value of K_p at 2 after 3 iterations. The values of T_i show again more variability.

More specifically, for the WTGRL system, the optimal values of the gains of the controller found are: $J = -1.3086633 \times 10^7$ with tuning parameters $(K_p, T_i) = (4707.6, 85.6)$, with the ALO algorithm; and $J = -1.30866328 \times 10^7$ with values of the controller gains $(K_p, T_i) = (2440.9, 44.2)$, with the WOA algorithm. The value of the cost function is very similar for the two techniques, although the controller parameters are radically different: both

the ratio $K_{p,ALO}/K_{p,WOA}$ and $T_{i,ALO}/T_{i,WOA}$ are ≈ 2 . However, inspection of Figure 9c,d reveals that, regardless of the controller used, the solutions with respect to the evolution of the angle of attack, the dissipated power, the voltage and the current intensity in the load overlap for both methods.

It should be noted that, for greater clarity in the visualization of the performance of the algorithms, only the ALO voltage is shown in Figure 9d since the results can be considered equal for both techniques. The voltage obtained with the parameters found by WOA is superimposed on the ALO result.

The overlapped solutions shown in Figure 9c,d are consistent with the well known fact that equivalent behaviors of a controlled system can be obtained for PI controllers with different values of its parameters.

In this sense, it is possible to affirm the ability of the ALO and WOA techniques to find different combinations of controller tuning parameters that lead to an optimized response of the complex and nonlinear WTGRL system. The rest of the configuration parameters of the ALO/WOA algorithms have been set to the value recommended in [15,16], and, therefore, it has only been necessary to know the desired range in the (K_p, T_i) space to find the optimal PI controller.

Table 4. WTGRL system: optimal minimum reached \bar{J} and computational time needed \bar{t}_{Opt} (in hours) for both the ALO and the WOA techniques, averaged over $n_R = 3$ runs.

ALO		WOA	
\bar{J}	\bar{t}_{Opt} [h]	\bar{J}	\bar{t}_{Opt} [h]
-1.308182×10^7	29.1	-1.308423×10^7	28.76

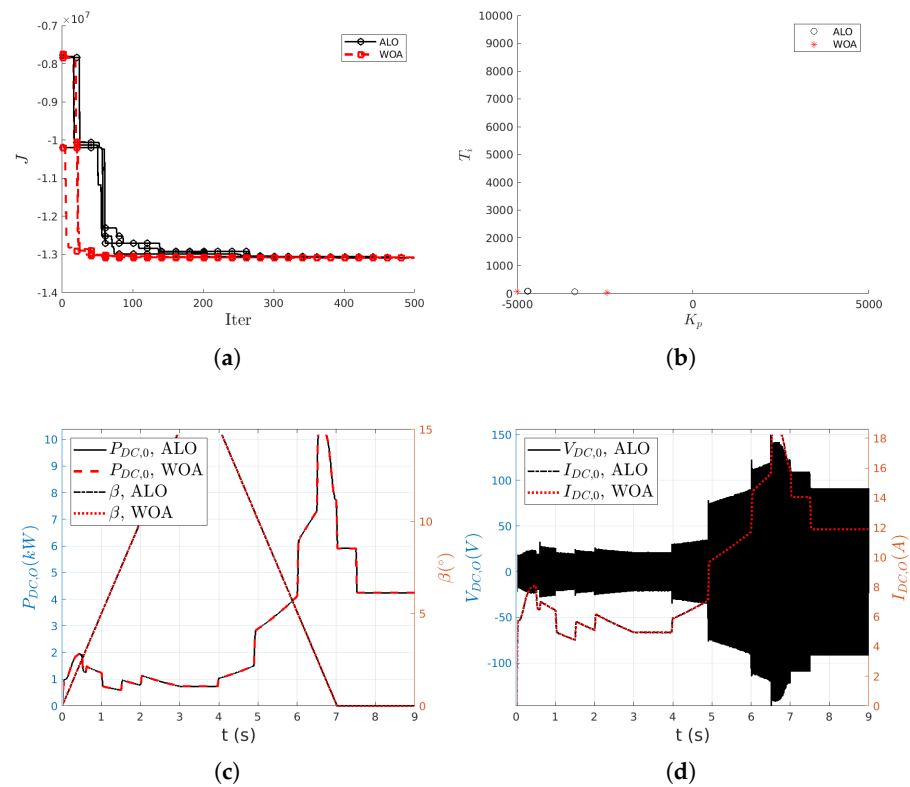


Figure 9. WTGRL system: in (a), evolution of the cost function; in (b), different solutions identified by ALO (o) and WOA (*); in (c,d), comparison of the optimal solutions found by ALO and WOA algorithms.

4. Conclusions and Future Works

In this work, the performance of two bio-inspired metaheuristic optimization algorithms, the Antlion Optimizer (or ALO) and the Whale Optimization Algorithm (or WOA), has been investigated. They have been applied to the adjustment of a conventional Proportional Integral (PI) controller for two systems of different complexity, the electromechanical Hoop and Ball (HB) system, and a wind turbine, generator, rectifier and load (WTGRL) system. Both techniques have been shown to present a robust and effective performance in the problems addressed.

The application to these systems is motivated by comparing both algorithms in two different scenarios. The Hoop and Ball (HB) system, allows a linearized description of it, while the wind turbine (WTGRL) system is an example of a complex non-linear dynamic system, of interest in the field of renewable energies, which cannot be adequately represented by a linear model.

In evaluating the performance of the ALO and WOA techniques for tuning the PI controllers, a sensitivity analysis of the two most relevant parameters has been considered: the number of agents n_S and the maximum number of iterations $n_{MaxIter}$. The repeatability properties (i.e., the dependence on the seed for the random number generator) and the computational effort required have also been studied.

From the results obtained with both optimization techniques, it has been observed that both methods reach similar optimal values for the fitness (cost) function J . In general, the more agents and the more time they are allowed to explore the search space, the better solution, i.e., the lower the value of J , is found.

Regarding the computational time, it is observed that t_{Opt} increases linearly with both the number of agents n_S and the number of iterations $n_{MaxIter}$ with both optimization techniques.

On the other hand, the versatility of these techniques is shown since in some cases the ALO and/or WOA techniques have arrived at very different solutions, finding very different values of the parameters K_p/T_i of the controller that, however, give the same fitness function and very similar responses of the dynamic systems considered.

In general, the effectiveness and robustness demonstrated by both bio-inspired metaheuristic techniques, the Antlion Optimizer and the Whale Optimization Algorithm, shown in their application to various optimization problems, have been confirmed here for the tuning of a conventional controller for non-linear dynamical systems.

Another interesting consideration that can be deduced is that ALO/WOA methods should be used, whenever computationally affordable, by considering a sufficient number of runs, i.e., n_R . This allows for considering the randomness of the initial conditions and the repeatability of the experiments and results obtained.

This last conclusion supports the need to have relatively simple models in terms of their simulation. This is one of the main limitations identified in the application of these techniques. Indeed, the faster a model is executed (the better the computational efficiency), the greater the number of search agents, iterations and realizations that can be performed and, therefore, the more complete the exploration of the solution space will be.

Other future work includes the possibility of applying these techniques ALO/WOA to other control problems in various fields. Specifically, and addressing another limitation of these metaheuristic methods, it would be desirable to have well-defined problems where these or other techniques could be compared. For this reason, in this work an effort has been made to establish the repeatability of the experiments.

Author Contributions: Conceptualization, J.G.-M.; methodology, J.G.-M. and M.S.; software, J.G.-M.; formal analysis, J.G.-M. and M.S.; investigation, J.G.-M. and M.S.; writing—original draft preparation, J.G.-M.; writing—review and editing, J.G.-M. and M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially funded by the Spanish Ministry of Science and Innovation, under the MCI/AEI/FEDER project number PID2021-123543OBC21.

Data Availability Statement: Data generated during the current study are available from the corresponding authors on reasonable request.

Acknowledgments: The authors acknowledge E. Fábregas (Universidad Nacional de Educación a Distancia) for his help with the simulation of the Hoop & Ball system.

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

Abbreviations

The following abbreviations are used in this manuscript:

ALO	Antlion Optimization
WOA	Whale Optimization Algorithm
HB	Hoop & Ball
WTGRL	Wind turbine-generator-rectifier-load
PI	Proportional Integral
PID	Proportional Integral Derivative
MPPT	Maximum Power Point Tracking
GA	Genetic Algorithm
PSO	Particle Swarm Optimization

References

- Borase, R.P.; Maghade, D.; Sondkar, S.; Pawar, S. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control* **2021**, *9*, 818–827. [\[CrossRef\]](#)
- Somefun, O.A.; Akingbade, K.; Dahunsi, F. The dilemma of PID tuning. *Annu. Rev. Control* **2021**, *52*, 65–74. [\[CrossRef\]](#)
- Åström, K.; Hägglund, T. *PID Controllers; Setting the Standard for Automation*; International Society for Measurement and Control: London, UK, 1995.
- Ogata, K. *Modern Control Engineering; Instrumentation and Controls Series*; Prentice Hall: Hoboken, NJ, USA, 2010.
- Sierra-Garcia, J.E.; Santos, M.; Pandit, R. Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104769. [\[CrossRef\]](#)
- Martínez-Luzuriaga, P.N.; Reynoso-Meza, G. Influencia de los hiper-parámetros en algoritmos basados en Evolución Diferencial para el ajuste de controladores del tipo PID en procesos SISO. *Rev. Iberoam. Automática Informática Ind.* **2023**, *20*, 44–55. [\[CrossRef\]](#)
- Blondin, M.J.; Sanchis Sáez, J.; Pardalos, P.M. *Control Engineering from Classical to Intelligent Control Theory—An Overview*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–30. [\[CrossRef\]](#)
- Blondin, M.J.; Pardalos, P.M.; Sanchis Sáez, J. (Eds.) *Computational Intelligence and Optimization Methods for Control Engineering*; Springer International Publishing: Cham, Switzerland, 2019. [\[CrossRef\]](#)
- Muñoz-Palomeque, E.; Sierra-García, J.E.; Santos, M. Enhancing Offshore Wind Turbines Performance with Hybrid Control Strategies Using Neural Networks and Conventional Controllers. *J. Comput. Des. Eng.* **2024**, qwa103. [\[CrossRef\]](#)
- Joseph, S.B.; Dada, E.G.; Abidemi, A.; Oyewola, D.O.; Khammas, B.M. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon* **2022**, *8*. [\[CrossRef\]](#) [\[PubMed\]](#)
- El-Telbany, M. Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study. *ICGST-ACSE J.* **2007**, *7*, 49–54.
- Kanthalakshmi, S.; Maninakandan, V. Genetic Algorithm Based Self Tuning Regulator. *Int. J. Eng. Sci. Technol.* **2010**, *2*.
- Gharghory, S.M.; Kamal, H.A. Optimal Tuning of PID Controller using Adaptive Hybrid Particle Swarm Optimization Algorithm. *Int. J. Comput. Commun. Control* **2012**, *7*, 101–114. [\[CrossRef\]](#)

14. Nagaraj, B.; Muruganath, N. A comparative study of PID controller tuning using GA, EP, PSO and ACO. In Proceedings of the 2010 International Conference on Communication Control and Computing Technologies, Nagercoil, India, 7–9 October 2010 ; pp. 305–313.
15. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
16. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
17. Wellstead, P.E. The ball and hoop system. *Automatica* **1983**, *19*, 401–406. [[CrossRef](#)]
18. Gurtner, M.; Zemánek, J. Ball in double hoop: Demonstration model for numerical optimal control. *Ifac-Papersonline* **2017**, *50*, 2379–2384. [[CrossRef](#)]
19. Readman, M.; Wellstead, P. *Control 1: Ball and Hoop System*; Technical Report; Control Systems Principles; 2000.
20. Fabregas, E.; Duro, N.; Dormido, R.; Dormido-Canto, S.; Vargas, H.; Dormido, S. Virtual and remote experimentation with the Ball and Hoop system. In Proceedings of the 2009 IEEE Conference on Emerging Technologies & Factory Automation, Palma de Mallorca, Spain, 22–25 September 2009; pp. 1–8. [[CrossRef](#)]
21. Zhou, B.; Zhang, Z.; Li, G.; Yang, D.; Santos, M. Review of key technologies for offshore floating wind power generation. *Energies* **2023**, *16*, 710. [[CrossRef](#)]
22. Dhas Bensam, S.; Maruthu Pandi, P. A hybrid MWOAL approach for fast and efficient maximum power point tracking in wind energy conversion systems. *J. Renew. Sustain. Energy* **2019**, *11*, 033302. [[CrossRef](#)]
23. Qais, M.H.; Hasanien, H.M.; Alghuwainem, S. Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators. *Appl. Soft Comput.* **2020**, *86*, 105937. [[CrossRef](#)]
24. Hosseini, S.H.; Farakhor, A.; Haghighian, S.K. Novel algorithm of maximum power point tracking (MPPT) for variable speed PMSG wind generation systems through model predictive control. In Proceedings of the 2013 8th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 28–30 November 2013; pp. 243–247. [[CrossRef](#)]
25. Mrabet, N.; Benzazah, C.; Akkary, A.E.; Ennaciri, Y.; Lahlouh, I. Maximum power optimization of a direct-drive wind turbine connected to PMSG using multi-objective genetic algorithm. *Int. J. Appl. Power Eng. (Ijape)* **2024**, *13*, 453–464. [[CrossRef](#)]
26. HIL Tutorial. MPPT with PMSG Direct Driven Wind Energy Conversion System Considering Variable Wind Speed. Available online: <https://www.youtube.com/watch?v=t4G8Ci-4gyM> (accessed on 12 December 2024).
27. HIL Tutorial. Correction Video for <<Simulation of Pitch Angle Controller and PMSG Based WECS: Part-2>>. Available online: https://www.youtube.com/watch?v=7E_NA14E4ws (accessed on 12 December 2024).
28. BBC Earth. Antlion Cone Death Trap | The Hunt | BBC Earth. Available online: <https://youtu.be/QSYpWafSIRY> (accessed on 12 December 2024).
29. BBC Earth. Whales' Bubble Net Fishing | Nature's Great Events | BBC Earth. Available online: <https://youtu.be/Q8iDcLTD9wQ> (accessed on 12 December 2024).
30. Mirjalili, S. Ant Lion Optimizer (ALO). Available online: <https://nl.mathworks.com/matlabcentral/fileexchange/49920-ant-lion-optimizer-alo> (accessed on 12 December 2024).
31. Mirjalili, S. The Whale Optimization Algorithm—MATLAB Central File Exchange. Available online: <https://nl.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm> (accessed on 12 December 2024).

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.