algorithms

MDPI

Article

Intelligent Search of Values for a Controller Using the Artificial Bee Colony Algorithm to Control the Velocity of Displacement of a Robot

José M. Villegas [1], Camilo Caraveo [1,*], David A. Mejía [1], José L. Rodríguez [1], Yuridia Vega [1], Leticia Cervantes [1] and Alejandro Medina-Santiago [2]

[1] Facultad de Ciencias de la Ingeniería y Tecnología, Universidad Autónoma de Baja California, Blvd. Universitario #1000, Unidad Valle de las Palmas, Tijuana 21500, Mexico; villegas_josemanuel@uabc.edu.mx (J.M.V.); damejia@uabc.edu.mx (D.A.M.); jlrodriguez@uabc.edu.mx (J.L.R.); vegay@uabc.edu.mx (Y.V.); leticia.cervantes.huerta@uabc.edu.mx (L.C.)

[2] National Institute for Astrophysics, Optics and Electronics, Puebla 72840, Mexico; amedina@inaoep.mx

* Correspondence: camilo.caraveo@uabc.edu.mx

Abstract: The optimization is essential in the engineering area and, in conjunction with use of meta-heuristics, has had a great impact in recent years; this is because of its great precision in search of optimal parameters for the solution of problems. In this work, the use of the Artificial Bee Colony Algorithm (ABC) is presented to optimize the values for the variables of a proportional integral controller (PI) to observe the behavior of the controller with the optimized T_i and K_p values. It is proposed using a robot built using the MINDSTORMS version EV3 kit. The objective of this work is to demonstrate the improvement and efficiency of the controllers in conjunction with optimization meta-heuristics. In the results section, we observe that the results improve considerably compared to traditional methods. In this work, the main contribution is the implementation of an optimization algorithm (ABC) applied to a controller (PI), and the results are tested to control the movement of a robot. There are many papers where the kit is used in various domains such as education as well as research for science and technology tasks and some real-world problems by engineering scholars, showing the acceptable result.

Keywords: meta-heuristics; integral proportional controller; artificial bee colony; dynamic adaptation

Citation: Villegas, J.M.; Caraveo, C.; Mejía, D.A.; Rodríguez, J.L.; Vega, Y.; Cervantes, L.; Medina-Santiago, A. Intelligent Search of Values for a Controller Using the Artificial Bee Colony Algorithm to Control the Velocity of Displacement of a Robot. *Algorithms* **2021**, *14*, 273. https://doi.org/10.3390/a14090273

Academic Editor: Antonio Della Cioppa

Received: 18 August 2021
Accepted: 15 September 2021
Published: 18 September 2021

1. Introduction

Today is the digitalization era, with robotics, artificial intelligence, IoT (Internet of Things) and automation. Since ancient times, the human being has sought to automate all its activities, avoiding the effort and intervention of the human hand. In recent years, the combination of different intelligent techniques helps different areas of science to improve and optimize time, performance, accuracy, and computational cost when solving any problems [1]. In this paper, an innovative proposal is presented where a combination of meta-heuristics and control are used for the intelligent search of values of the variables K_p and T_i and applied in a LEGO EV3 robot. K_p represents the proportional gain and refers to the constant of proportionality in the action to proportional control. If the value of K_p is small, the proportional action is small, but if K_p is high, the proportional action is high. The variable T_i refers to the time required for the integral action to contribute to the controller output. If T_i is small, the integral action is high, but if T_i is high, the integral action is small.

Today actually is the era of robotics, where, in conjunction with artificial intelligence, humanity has managed to achieve significant advances in some areas, for example: home robots, academics, medicine, agriculture, combat robots, scout robots in other planets (example: Mars exploration rover), and intelligence implemented in other entities such as autonomous land and aerial vehicles and virtual assistants [2], for mention of some areas. It is important to mention the new term that emerged recently: machine learning (ML). ML is the science of getting machines to learn and act like humans, and improve their learning over time autonomously, by feeding them data and information in the form of observations and real-world interactions [3].

Each of the entities mentioned above integrate some of the following: neural networks for learning in conjunction with other sciences for the collection and processing of information, audio and image, and some type of controller to coordinate and activate some actuators of the robot such as: arms, legs, engines, and others. There are several types of controllers that are used for the control of some system such as: proportional integral derivative control (PID) [4], fuzzy logic control (FLC) [5], and its extension, type 2 fuzzy logic control [6].

In this case, a proportional integral controller (PI) is used, where the values for the constants are adjusted in execution time using a bio-inspired algorithm in the behavior of bees in nature in the process of searching and collecting the nectar.

The objective of this work is to demonstrate that bio-inspired algorithms can significantly improve the results in solving optimization problems. It is important to mention that, in the literature, there are many published works with implementations of the ABC algorithm in control problems, for example: PID, type 1 and type 2 fuzzy logical control, and approximation of mathematical functions. However, a few studies have found that this algorithm can be used to optimize the parameters of a PI and observe the behavior. There are many bio-inspired algorithms in the literature, each with different characteristics. The selection of the algorithm for this proposal was determined after analyzing its characteristics and based on [7–9].

2. Related Work

There are multiple publications in the literature of different bio-inspired meta-heuristics in the behavior of the bees in the nature; for example, in [10], the authors present their proposal of a new powerful meta-heuristic for the resolution of complex problems, and years later, authors of [11,12] present a modification of the ABC algorithm for unrestricted searches. Other authors also present some variants similar to this meta-heuristic mentioned above (e.g., [7,13–15]).

During the last several years, multiple variants of the algorithm proposed in this work have been developed. It is important to mention that each variant was developed to solve a specific problem: optimization of neural networks, optimization of fuzzy logic controllers type 1 and type 2, and optimization of mathematical functions, to mention some. The authors of this proposal consider that is important to mention in this paper.

The original version of the ABC algorithm proposed by [10] to dynamically adjust the values of the constants of a controller (PI) is used in this paper. There are also published works by other authors where they use different controllers applied to the LEGO MINDSTORMS EV3 kit, such as: proportional integral derivative controllers (PID) [16,17], fuzzy logic controllers (FLC) [18], fuzzy logic proposed by Zadeh [5], and proportional integral controllers (PI) [19].

It is also necessary to show the reader of this manuscript some works related to our proposal. For example, in [20], “Real-Time PID Control of Wireless Two-Wheeled Balancing LEGO EV3 Robot”, a two-wheeled balancing LEGO EV3 robot is modeled in a state space and controlled by a PID controller, which we find very interesting. In [21], the authors present an application for mobile devices to operate and read the status of motors and controller of the LEGO robot and affirm that this app can operate with different sizes of motors. In [22], the authors use a LEGO robot NXT version with infrared sensor to follow a trajectory with reference to a line and use a control (PID) to control the motors. Furthermore, the use of fuzzy logic is present in the LEGO kits, for example.

In [23,24], a fuzzy logic controller is used as a controller to stabilize the two-wheeled EV3 LEGO robot. In recent years, fuzzy controllers in combination with meta-heuristics have had a great impact in different areas [24].

EV3 MINDSTORMS LEGO characteristics: The brick or block EV3 is the unit of control and electrical feeding of motors and sensors of the robot. It connects to the computer (via USB, Bluetooth, or Wi-Fi) to receive and execute the control programs, and it is the successor of LEGO MINDSTORMS NXT 2.0. The LEGO EV3 kit is the third generation of the LEGO robotics line and recommended for teaching and learning school robotics. It has a 300 MHz ARM9 processor, 16 MB of flash memory, 64 MB of RAM, and a Linux operating system, with an open-source license. The EV3 was implemented to test the optimized PI model with the ABC algorithm and observe the displacement behavior.

However, some researchers have used it for scientific purposes to test their models and observe behavior in different areas and types of problems [25,26].

Proportional integral controller (PI): There are two types of forms in a control algorithm to decide on an action. The open loop system and the parameters of the algorithm are pre-set and do not change. While the system is running, the closed loop system sensors measure the error between the desired state of the system and its real state, and this error is used to decide what action to take. To achieve speed control in a EV3 LEGO MINDSTORM, open circuit control systems were used. Figure 1 shows the general scheme of a loop control system [19,26], where the values of the constants K_p and T_i are calibrated at runtime using the ABC algorithm.

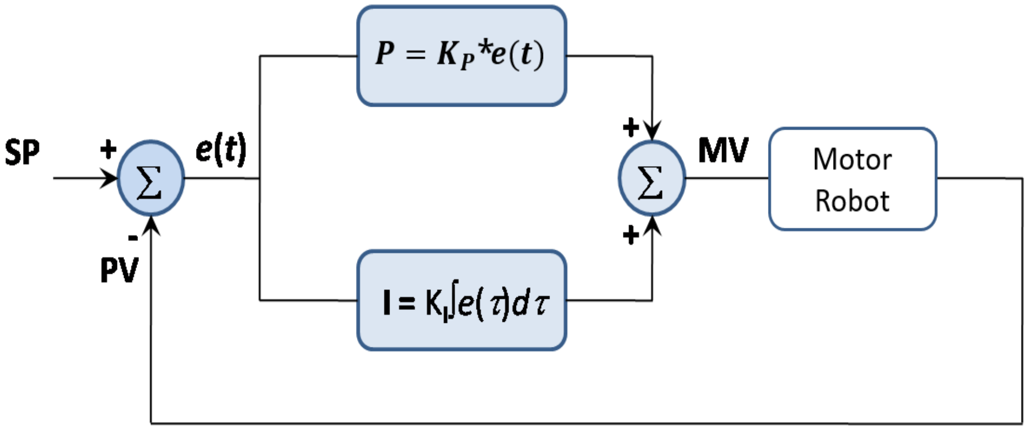

Figure 1. Closed open loop control system.

The parameter SP represents the reference value; the variable transformed into a control value is represented by MV; and the parameter PV represents the output, which is calculated with the equation e = SP − PV.

Artificial Bee Colony Algorithm (ABC): The algorithm of artificial bee colony is a meta-heuristic inspired by the behavior of bees in nature and was created by Dervis Karaboga in 2005 [10]. The algorithm (ABC) uses the principle of collective intelligence where the individuals (bees) collaborate with each other to achieve an objective. After the proposal made by Dervis Karaboga in 2015, many variants and improvements have been developed in this work, the original version is used. This algorithm has very interesting characteristics. Bees are classified into three roles according to their fitness value. For example:

1. Initialization Phase: All the vectors of the population of food sources, (x_m)'s, are initialized ($m = 1, \ldots SN$: population size) by scout bees, and control parameters are set. Since each food source, (x_m), is a solution vector to the optimization problem, each (x_m) vector holds n variables, $(x_{mi}, i = 1 \ldots n)$, which are to be optimized so as to minimize the objective function.

   The following definition might be used for initialization purposes:

$$x_{mi} = l_i + rand(0,1) \times (v_i - l_i)$$

where l_i and u_i represent the lower and upper bound of the parameter x_{mi}, respectively.

2. Employed Bees Phase: New food origins are being searched (v_m); the one that has more nectar in the food neighborhood (x_m) is put in memory. Once the food origin neighbors are found, then rentability (aptitude) is evaluated. For example, a food origin can be determined by using the formula of the following Equation (1) [27].

$$v_{mi} = x_{mi} + \phi_{mi}\,(x_{mi} - x_{ki}) \tag{1}$$

where x_k is a randomly selected food source, i is a randomly chosen parameter index, and ϕ_{mi} is a random number within the range $[-a, a]$. After producing the new food source v_m, its fitness is calculated, and a greedy selection is applied between v_m and $x_{m.}$ The fitness value of the solution, $fit_m(x_m)$, might be calculated for minimization problems using the following Equation (2) [27].

$$fit_m(x_m) = \left\{ \frac{1}{1 + f_m(x_m)\ 1 + abs(f_m(x_m))} if \quad f_m(x_m) \geq 0\ f_m(x_m) \leq 0 \right\} \tag{2}$$

where $f_m(x_m)$ represent the objective function value for solution x_m.

3. Onlooker Bees Phase: In this optimization algorithm, an observer bee chooses a food source based on the calculated fitness value with Equation (2). Data from food sources are provided by the bees employed. The roulette method is used for selection [28]. The probability value P_m with which X_m is chosen by an observer bee can be calculated using the expression given in the Equation (3) [27]:

$$P_m = \frac{fit_m(x_m)}{\sum\limits_{m=1}^{SN} fit_m(x_m)} \tag{3}$$

After choosing a food source x_m for a spectator bee, a neighborhood source v_m is determined using Equation (2), and the fitness value is calculated. As in the phase of bees used, a greedy selection between v_m and x_m is applied.

4. Scout Bees Phase: The bees without jobs who choose their food origins randomly are called scout bees. Then the converted scouts begin to look for new random solutions. For example, if solution x_m has been abandoned, the new solution discovered by the scout who was the used bee x_m can be defined using Equation (4). Therefore, all food origins with a low fitness value are abandoned due to exploitation [27].

The population of food sources is generated by the exploratory bees. x_m (m = 1 ... , SN: population size) and the parameters of the algorithm are configured. In the solution vector for the optimization problem, each vector x_m, represents each food source, x_m, and contains n variables (x_{mi}, i = 1 ... n), which must be optimized to minimize the Objective Equation (4):

$$x_{mi} = l_i + rand(0,1) \times (u_i - l_i) \tag{4}$$

where l_i and u_i represent the lower and upper bound of the parameter x_{mi}, respectively.

In summary, the ABC algorithm:

1. Is inspired by the foraging behavior of honeybees;
2. Is a global optimization algorithm;
3. Has been initially proposed for numerical optimization [10];
4. Can be also used for combinatorial optimization problems [22];
5. Can be used for unconstrained and constrained optimization problems [10,12];
6. Uses only three control parameters (population size, maximum cycle number, and limit) that are to be pre-determined by the user;
7. Is quite simple, flexible, and robust (some of the relevant publications expressing these merits of the ABC algorithm are [10,11]).

3. Proposed Approach

In this work, we present a dynamic adaptation of variables for a proportional integral controller using the algorithm of artificial bee colony modeling in a EV3 LEGO MINDSTORM robot. As mentioned above, the LEGO kit is used in multiple domains such as: research and education for science and technology tasks and some real-world problems by engineering scholars.

The main contribution is the use of a meta-heuristic to adjust the variables of a (PI) control in a runtime and, at the same time, to model it in a LEGO robot. This process is carried out in wireless mode using the language of MATLAB programming in conjunction with Simulink.

In Figure 2, a flow diagram is shown that represents the sequence of steps to follow to achieve an optimal communication between the algorithm, Simulink block diagram, and LEGO EV3 robot.

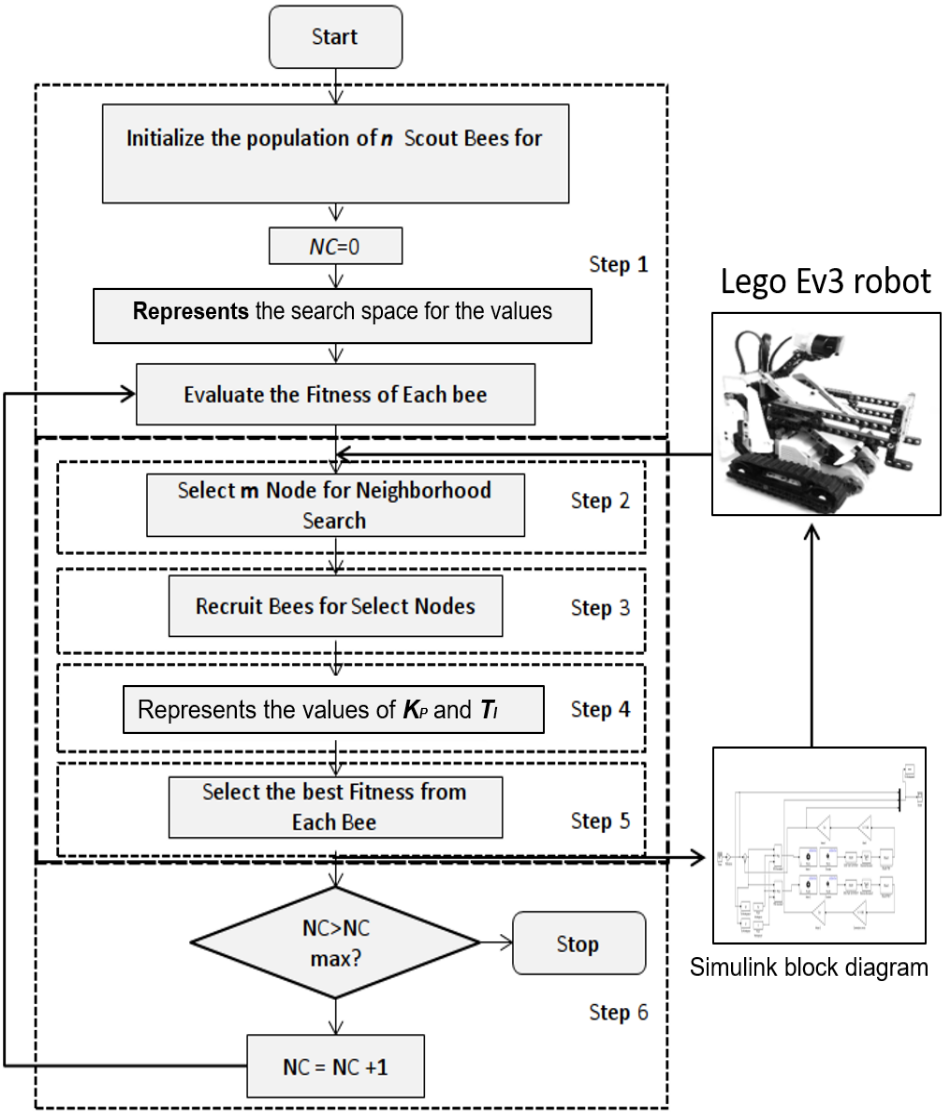
Start
Initialize the population of *n* Scout Bees for
NC=0
Step 1
Represents the search space for the values
Evaluate the Fitness of Each bee
Select **m** Node for Neighborhood Search
Step 2
Recruit Bees for Select Nodes
Step 3
Represents the values of K_P and T_I
Step 4
Select the best Fitness from Each Bee
Step 5
NC>NC max?
Stop
Step 6
NC = NC +1
Lego Ev3 robot
Simulink block diagram

Figure 2. Flowchart of ABC algorithm applied to the Lego Ev3 robot.

The algorithm consists of the following phases:

Step 1: the population of bees is initialized. Considering the search universe, this universe is defined by the problem to be solved; in this case, [−1, 9] is proposed, and the solutions generated so far are evaluated.

Step 2, 3, 4, 5: in these steps, each new solution is evaluated, and the best one found so far goes to the next iteration to be optimized.

The best solution is determined after simulating the path given by ABC in the robot, and its fitness value is calculated using the mean square error Equation (5).

Methodology

In Figure 2, we can see a general illustration of our proposal. For the development of the proposal, as mentioned above, the role of the meta-heuristic bee colony algorithm is to explore the search space in order to find the optimal values for the constants (K_p, T_i) of the PI controller.

Each value found by the meta-heuristic is assigned to the variables of the controller and is modeled in the LEGO robot, and the fitness value is obtained for that model. The connection with the brick LEGO is performed by wireless mode using a block model in Simulink and a Wi-Fi router to obtain the values and calculate the error of the motors (see Figure 2).

Before starting with the dynamic adaptation of variables of the (PI controller), we decided to use a reference method to use it as a guide in this work, there are many open loop tuning methods for self-regulating process (reaches a steady-state value after a step change in the manipulated variable) that use a first-order plus dead time model of the process (FODT) [29–31]. The transfer function of a general first-order plus dead time model is

$$G_{FODT}(s) = \frac{Ke^{-s\theta_D}}{\tau s + 1} \tag{5}$$

where

K = Gain Process
θ_D = Dead time
τ = Time constant

The graphical procedure for determining the model is based on Ziegler–Nichols, according to Kelvin T. Erickson in his book *Programmable Logic Controllers* [31]. The method consists of obtaining the features of the response in a step input of magnitude such as demonstrated by Figure 3. The response to this input is obtained experimentally.

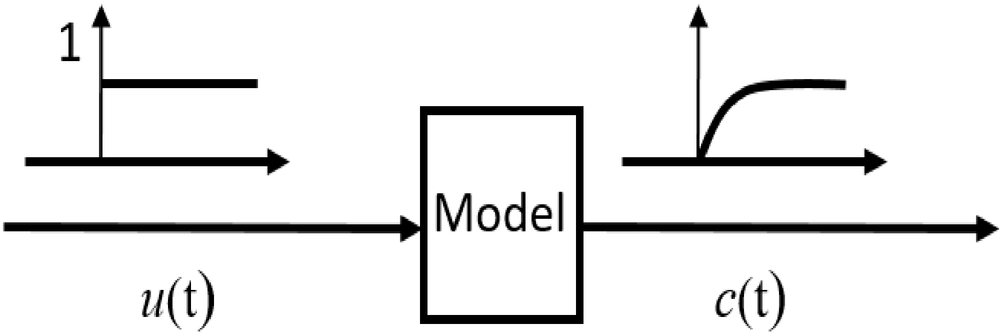

Figure 3. Unit step input and response.

It is important to analyze the response obtained, since it must have an S shape; this type of response has two parameters: the dead time θ_D and the time constant τ. After the process reaches the steady state, the parameters could be determined by drawing a straight-line tangent to the inflection point of the response obtained, as shown in Figure 4.

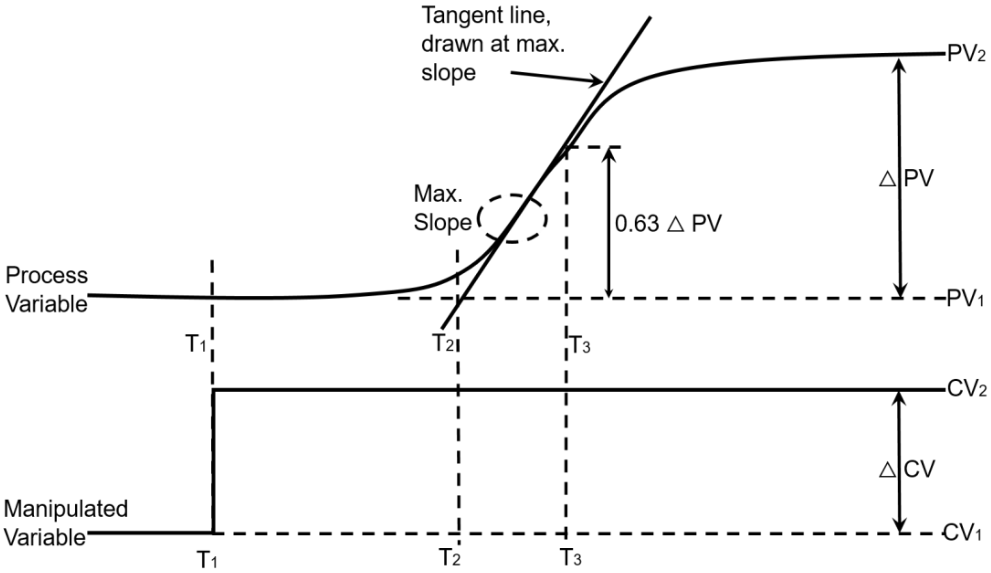

Figure 4. Response to step input.

The term T_1 is the time at which the step change in the manipulated variable was imposed. T_2 is the time at which the tangent drawn in the previous step intersects the time axis. To obtain the dead time, $\theta_D = T_2 - T_1$.

The term T_3 is the time at which the process variable response has made 63% of its total change. To obtain the time constant, $\tau = T_3 - T_2$.

The process gain is the change in the process variable divided by the change in the control variable; see Equation (6).

$$K = \frac{\Delta PV}{\Delta CV} = \frac{PV_2 - PV_1}{CV_2 - CV_1} \tag{6}$$

The values obtained are shown below:

$$\theta_D = T_2 - T_1 = 0.15298$$

$$(1)\tau = T_3 - T_2 = 1.02462$$

$$(2)K = \frac{\Delta PV}{\Delta CV} = \frac{PV_2 - PV_1}{CV_2 - CV_1} = 1$$

The first-order plus dead time model of the process (FODT) is presented in Equation (7).

$$G_{FODT}(s) = \frac{e^{-s0.15298}}{1.02462s + 1} \tag{7}$$

According to the Ziegler and Nichols open loop tuning method, values of K_p, T_i, and T_d are established and calculated from the parameters θ_D, τ, and K. In Table 1, we can see the tuning rule.

Table 1. Ziegler and Nichols' tuning rules.

Controller Type	K_p	T_i	T_d
p	$\frac{\tau}{K\theta_D}$	∞	0
PI	$0.9\frac{\tau}{K\theta_D}$	$\frac{\theta_D}{0.3}$	0
PID	$1.2\frac{\tau}{K\theta_D}$	$2\theta_D$	$0.5\theta_D$

The proposed controller for this test was PI, and the values obtained are shown below:

$$K_p = 0.9\frac{\tau}{K\theta_D} = 0.9 \times \frac{1.02462}{0.15298} = 6.02796$$

$$(3) T_i = \frac{\theta_D}{0.3} = \frac{0.15298}{0.3} = 0.50993$$

The values for the variables of the PI were implemented in the Simulink model.

In Figure 5, the three signals are shown: Yellow signal is the pulse input to the system (reference). The magenta and cyan signals are the responses of motor one and two. The type of input to the system was a pulse signal with an amplitude of 150 square.

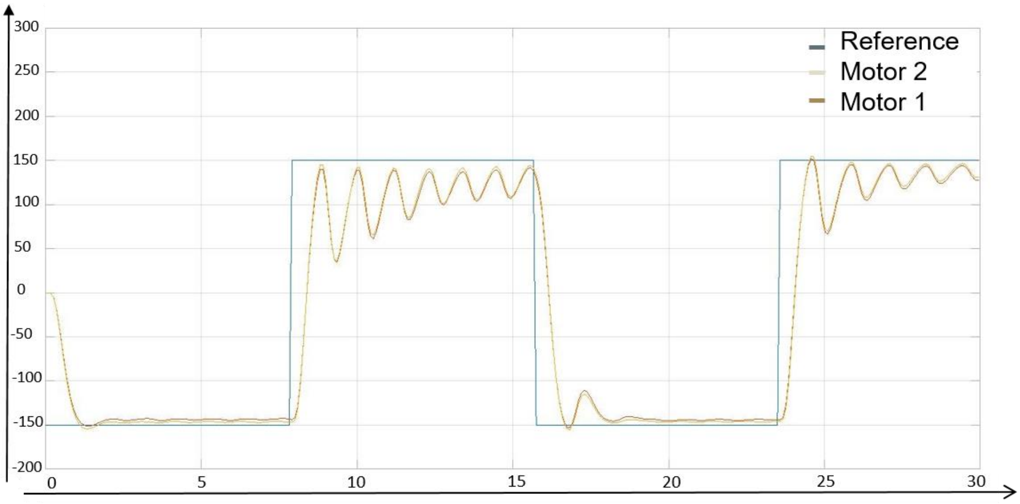

Figure 5. Behavior using PI traditional control.

The objective of this work is the implementation of a meta-heuristic to the automatic adjustment of the value of the variables of the PI. The method of Ziegler and Nichols is used to analyze the behavior of the block model and response of the LEGO robot EV3. Using Equation (5), an error of **16.6874** was obtained for this test.

4. Results Obtained from The Tests Performed

In this section the performance, stability and behavior of the robot is presented using the PI control with intelligent adjustment in the variables (K_p, T_i). Each experiment has 15 tests performed with the following configuration of parameters in the meta-heuristic ABC: population size = 20, iterations = 50, and search range [−1 9], using as a reference the results obtained with the Ziegler and Nichols method. The results obtained are shown in the Table 2.

Table 2. Results of control PI using ABC optimization.

NUM.	T_i	K_p	*BEST*
1	**3.8538**	**8.5949**	**2.0947**
2	1.3049	3.3470	6.5614
3	**5.5574**	**7.0028**	**3.0410**
4	6.5373	7.3083	4.5221
5	7.6869	8.1065	3.1513
6	8.6309	0.6437	4.6822
7	3.8538	8.5949	3.2261
8	6.4815	7.2582	3.7371
9	3.1727	6.7449	3.5709
10	4.2885	8.6309	3.5917

Table 2. *Cont.*

NUM.	T_i	K_p	*BEST*
11	4.1859	7.2531	4.1478
12	4.8299	8.8266	4.1827
13	3.2176	6.4975	3.1795
14	−0.2182	6.1521	3.3044
15	8.1057	8.339	4.1057
16	7.2582	8.6190	4.5323
17	7.002	6.0428	3.8286
18	3.1726	6.7449	4.0684
19	**0.2836**	**7.5682**	**1.2493**
20	1.2345	6.4258	3.1565

In Table 2, we can observe the results obtained from the tests performed on the robot. We consider it important to show the values found for the variables (K_p, T_i) and the error obtained. The error was calculated using the equation of the root of the mean square error (RMSE); see Equation (8):

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_{ref} - y_{data})^2}{N}} \tag{8}$$

where y_{ref} represents the reference to follow, y_{data} represents the data obtained from the motors of the robot, and N is the number of data.

The behavior of the robot motors when following the reference for tests 19, 1, and 3 is shown in Figures 6–8. Observing the fitness value obtained, we determined that they are the best results obtained from the tests carried out with the robot and the meta-heuristics; moreover, the values for the constants of the controller (PI) are shown.

The yellow signal is the pulse of input to the system (reference); the magenta and cyan signals are the responses of motor one and two. The type of input to the system was a pulse signal with an amplitude of 150 square.

For greater ease for the readers of this work, in Table 3, we can observe a comparison of the results obtained with the different approaches. It is important to mention that the error obtained with "Integral proportional control (PI), traditional" used the values defined by [29–31], and the error was 16.6874, which is high. When the optimization is used with values 7.5682 in K_p and 0.2836 in T_i, the error decreases considerably.

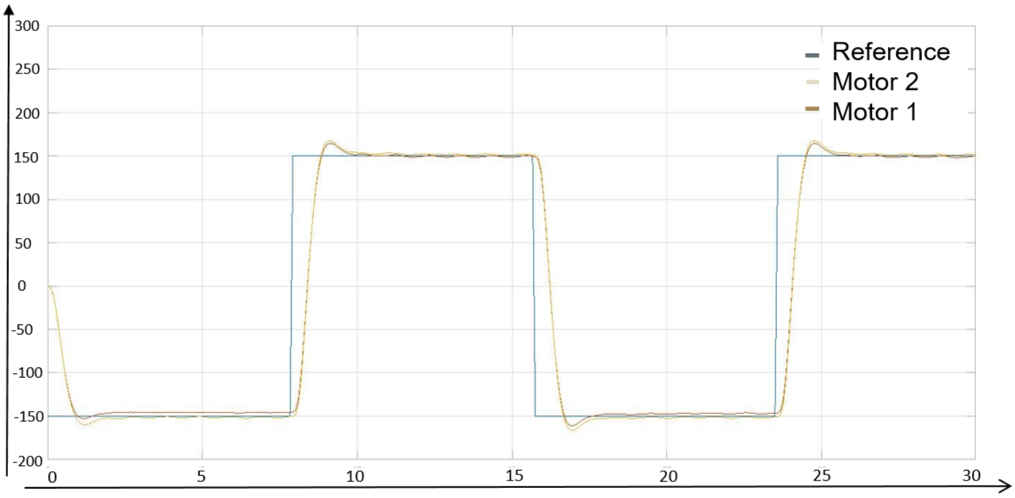

Figure 6. Behavior using PI with optimization values, test 19.

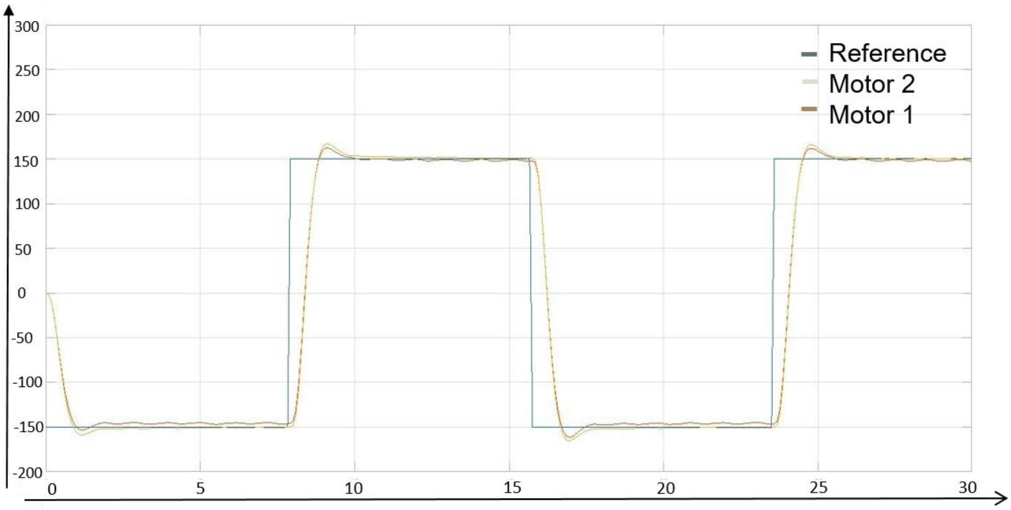

Figure 7. Behavior using PI with optimization values, test 1.

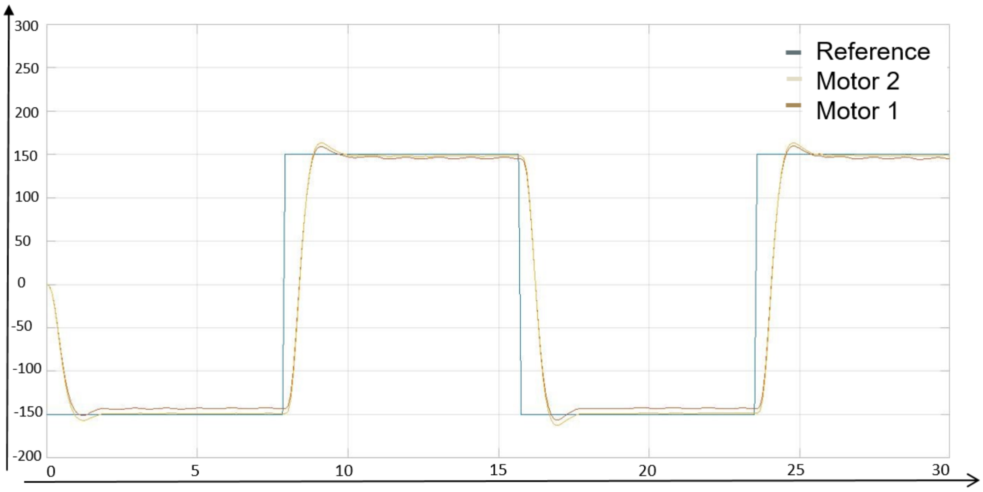

Figure 8. Behavior using PI with optimization values, test 3.

Table 3. Comparison using PI traditional and intelligent PI control.

Control Type	Values K_p	Values T_i	Results Error
Integral proportional control (PI), traditional	6.02796	0.50993	16.6874
Intelligent integral proportional control	7.5682	0.2836	1.2493

The values shown for K_p and T_i with “Intelligent integral proportional control” were found in a dynamic and intelligent way by the proposed algorithm (ABC), as we can see in the comparison: the error difference is very significant compared with that obtained using the traditional PI control. As mentioned in the introductory section, bio-inspired algorithms provide a great variety of solutions to different problems in engineering and other areas with a low computational cost.

5. Discussion

Throughout this work, we have presented a proof of concept of bio-inspired metaheuristics in the behavior of bees in nature applied to the dynamic adaptation of values to control the speed of movement of a robot. In this work, the Artificial Bees Colony (ABC) algorithm is proposed for the intelligent search of values K_p and T_i for Equation (5).

Based on the results obtained, we can conclude the feasibility of implementing bio-inspired optimization algorithms for the intelligent search of values for complex search and optimization problems. In this case, the ABC is used to control the speed of movement of a mobile robot. As we can see in Table 2, the results obtained were better achieved using the values proposed by Ziegler and Nichols [29]; see the results with values proposed by Ziegler and Nichols in Figure 6.

This work can be used as a basis to show readers the implementation of a bio-inspired algorithm to optimize the trajectory in an EV3 MINDSTORMS LEGO robot. It is important to mention that there are many application areas of algorithms such as: electronics, finance, construction, and telecommunications, to mention some. The applications are numerous and very varied due to their great adaptability and generate acceptable results with a low computational cost. As future work on the proposal, machine learning techniques can be implemented, which allows us to improve the models (ABC).

6. Conclusions

In this paper, a dynamic optimization of the parameters in the PI was performed. In the optimization, the search space was based on Ziegler and Nichols (see Table 1). As we mentioned, values were between -1 and 9. One of the objectives is to reduce the error of 16.6874 found with the Ziegler and Nichols method. After the experiments with the implementation in the robot, results show that bee colony optimization can obtain good parameters to the PI and improve it.

In many applications, a simple conventional PI is used to control the application, but we can use the best parameters for the PI, improving the controller even more, and obtain better results than using static values. In Table 2, important results were obtained and were better than Ziegler and Nichols. We were confident in a statistical comparison using Ziegler and Nichols, as the results obtained in this paper show. That is because the values of the K_p and T_i can change. The main objective was obtained, because the results were less than the reference.

Author Contributions: Conceptualization, J.M.V. and C.C.; methodology, J.M.V., C.C. and D.A.M.; software, J.M.V., C.C. and J.L.R.; validation, J.M.V., C.C. and J.L.R.; formal analysis, J.M.V., C.C., D.A.M. and L.C.; investigation, J.M.V. and C.C.; resources, J.M.V., C.C., D.A.M., J.L.R., Y.V., L.C. and A.M.-S.; data curation, J.M.V., C.C., J.L.R. and A.M.-S.; writing—original draft preparation, J.M.V., C.C. and L.C.; writing—review and editing, J.M.V., C.C. and Y.V.; visualization, J.M.V., C.C. and Y.V.; supervision, J.M.V.; project administration, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

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

References

1. Nair, S.; Coronado, E.; Frye, M.; Goldaracena, T.; Arguello, C. Particle Swarm Optimization for the control of a swarm of biological robots. In Proceedings of the 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 17–20 December 2015; pp. 1–4.
2. Ayawli, B.B.K.; Chellali, R.; Appiah, A.Y.; Kyeremeh, F. An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning. *J. Adv. Transp.* **2018**, *2018*, 8269698. [CrossRef]
3. Robert, C. Machine Learning, a Probabilistic Perspective. *Chance* **2014**, *27*, 62–63. [CrossRef]
4. Misir, D.; Malki, H.A.; Chen, G. Design and analysis of a fuzzy proportional-integral-derivative controller. *Fuzzy Sets Syst.* **1996**, *79*, 297–314. [CrossRef]
5. Zadeh, L.A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* **1997**, *90*, 111–127. [CrossRef]
6. Martínez, R.; Castillo, O.; Aguilar, L.T. Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. *Inf. Sci.* **2009**, *179*, 2158–2174. [CrossRef]

7. Amador, L.; Castillo, O. Bee Colony Optimization Algorithm. In *Optimization of Type-2 Fuzzy Controllers Using the Bee Colony Algorithm*; Amador, L., Castillo, O., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 23–32.
8. Cuéllar, M.P.; Pegalajar, M.C. Design and implementation of intelligent systems with LEGO Mindstorms for undergraduate computer engineers. *Comput. Appl. Eng. Educ.* **2014**, *22*, 153–166. [CrossRef]
9. Perez, S.R.; Gold-Veerkamp, C.; Abke, J.; Borgeest, K. A new didactic method for programming in C for freshmen students using LEGO mindstorms EV3. In Proceedings of the 2015 International Conference on Interactive Collaborative Learning (ICL), Florence, Italy, 20–24 September 2015; pp. 911–914.
10. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
11. Karaboga, D.; Akay, B.B. A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. *Appl. Soft Comput.* **2011**, *11*, 3021–3031. [CrossRef]
12. Niknam, T.; Taheri, S.I.; Aghaei, J.; Tabatabaei, S.; Nayeripour, M. A modified honey bee mating optimization algorithm for multiobjective placement of renewable energy resources. *Appl. Energy* **2011**, *88*, 4817–4830. [CrossRef]
13. Chong, C.S.; Sivakumar, A.I.; Low, M.Y.H.; Gay, K.L. A Bee Colony Optimization Algorithm to Job Shop Scheduling. In Proceedings of the 38th Conference on Winter Simulation, Monterey, CA, USA, 3–6 December 2006; pp. 1954–1961.
14. Teodorović, D.; Dell'Orco, M. Bee colony optimization—A cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation*; Publishing House of Poznan Uniwersity of Technology: Poznań, Poland, 2005; pp. 51–60.
15. Teodorović, D.; Lucic, P.; Markovic, G.; Orco, M.D. Bee Colony Optimization: Principles and Applications. In Proceedings of the 2006 8th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, 25–27 September 2006; pp. 151–156.
16. Luyben, W.L. Tuning Proportional−Integral−Derivative Controllers for Integrator/Deadtime Processes. *Ind. Eng. Chem. Res.* **1996**, *35*, 3480–3483. [CrossRef]
17. Saleh, A.L.; Hussain, M.A.; Klim, S.M. Optimal Trajectory Tracking Control for a Wheeled Mobile Robot Using Fractional Order PID Controller. *J. Univ. Babylon Eng. Sci.* **2018**, *26*, 292–306. [CrossRef]
18. Onieva, E.; Milanés, V.; Pérez, J.; De Pedro, T. Genetic Fuzzy-based Steering Wheel Controller using a Mass-Produced Car. *Int. J. Innov. Comput. Inf. Control.* **2012**, *8*, 3477–3494.
19. Nandam, P.K.; Sen, P.C. A comparative study of proportional-integral (P-I) and integral-proportional (I-P) controllers for dc motor drives. *Int. J. Control.* **1986**, *44*, 283–297. [CrossRef]
20. Hao, M.R.; Ghani, N.M.A.; Wahid, M.S.A. Real-Time PID Control of Wireless Two-Wheeled Balancing Lego EV3 Robot. *J. Telecommun. Electron. Comput. Eng.* **2018**, *10*, 6.
21. Bell, M.; Kelly, J.F. Robot Commander Remote Control App. In *LEGO® MINDSTORMS® EV3*; Bell, M., Kelly, J.F., Eds.; Apress: Berkeley, CA, USA, 2017; pp. 319–321.
22. Ding, J.; Li, Z.; Pan, T. Control System Teaching and Experiment Using LEGO MINDSTORMS NXT Robot. *Int. J. Inf. Educ. Technol.* **2017**, *7*, 309–313. [CrossRef]
23. Akmal, M.A.; Jamin, N.F.; Ghani, N.M.A. Fuzzy logic controller for two wheeled EV3 LEGO robot. In Proceedings of the 2017 IEEE Conference on Systems, Process and Control (ICSPC), Malacca, Malaysia, 15–17 December 2017; pp. 134–139.
24. Maharuddin, M.F.; Ghani, N.M.A.; Jamin, N.F. Two-Wheeled LEGO EV3 Robot Stabilisation Control Using Fuzzy Logic Based PSO Algorithm. *J. Telecommun. Electron. Comput. Eng.* **2018**, *10*, 149–153.
25. Bibi, R.; Chowdhry, B.S.; Shah, R.A. PSO based localization of multiple mobile robots employing LEGO EV3. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; pp. 1–5.
26. Kast, C.; Rosenauer, B.; Meissner, H.; Aramphianlert, W.; Krenn, M.; Hofer, C.; Mayr, W. Development of a Modular Bionic Prototype Arm Prosthesis Integrating a Closed-Loop Control System. In Proceedings of the World Congress on Medical Physics and Biomedical Engineering 2018, Prague, Czech Republic, 3–8 June 2018; Lhotska, L., Sukupova, L., Lacković, I., Ibbott, G.S., Eds.; Springer: Singapore, 2018; pp. 751–753.
27. Jadon, S.S.; Sharma, H.; Tiwari, R.; Bansal, J.C. Self-adaptive position update in artificial bee colony. *Int. J. Syst. Assur. Eng. Manag.* **2017**, *9*, 802–810. [CrossRef]
28. Goldberg, D.E.; Holland, J.H. Genetic Algorithms and Machine Learning. *Mach. Learn.* **1988**, *3*, 95–99. [CrossRef]
29. Ogata, K. *Ingeniería de Control Moderna*; Pearson Educación: London, UK, 2003.
30. Meshram, P.M.; Kanojiya, R.G. Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor. In Proceedings of the IEEE—International Conference on Advances in Engineering, Science and Management (ICAESM-2012), Nagapattinam, Tamil Nadu, India, 30–31 March 2012; pp. 117–122.
31. Kelvin, T. Erickson. In *Programmable Logic Controllers*; Dogwood Valley Press: Rolla, MO, USA, 2005.