

Article

# Improvement of Trajectory Tracking by Robot Manipulator Based on a New Co-Operative Optimization Algorithm

Mahmoud Elsisi <sup>1,2</sup> , Hatim G. Zaini <sup>3</sup>, Karar Mahmoud <sup>4,5</sup> , Shima Bergies <sup>6</sup> and Sherif S. M. Ghoneim <sup>7,\*</sup> 

- <sup>1</sup> Industry 4.0 Implementation Center, Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; mahmoud.elsisi@mail.ntust.edu.tw
- <sup>2</sup> Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 1162, Egypt
- <sup>3</sup> Computer Engineering Department, College of Computer and Information Technology, Taif University, Al Huwaya, Taif 26571, Saudi Arabia; h.zaini@tu.edu.sa
- <sup>4</sup> Department of Electrical Engineering and Automation, Aalto University, FI-00076 Espoo, Finland; karar.mostafa@aalto.fi
- <sup>5</sup> Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt
- <sup>6</sup> Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; M10807827@mail.ntust.edu.tw
- <sup>7</sup> Department of Electrical Engineering, College of Engineering, Taif University, Taif 21944, Saudi Arabia
- \* Correspondence: s.ghoneim@tu.edu.sa

**Abstract:** The tracking of a predefined trajectory with less error, system-settling time, system, and overshoot is the main challenge with the robot-manipulator controller. In this regard, this paper introduces a new design for the robot-manipulator controller based on a recently developed algorithm named the butterfly optimization algorithm (BOA). The proposed BOA utilizes the neighboring butterflies' co-operation by sharing their knowledge in order to tackle the issue of trapping at the local optima and enhance the global search. Furthermore, the BOA requires few adjustable parameters via other optimization algorithms for the optimal design of the robot-manipulator controller. The BOA is combined with a developed figure of demerit fitness function in order to improve the trajectory tracking, which is specified by the simultaneous minimization of the response steady-state error, settling time, and overshoot by the robot manipulator. Various test scenarios are created to confirm the performance of the BOA-based robot manipulator to track different trajectories, including linear and nonlinear manners. Besides, the proposed algorithm can provide a maximum overshoot and settling time of less than 1.8101% and 0.1138 s, respectively, for the robot's response compared to other optimization algorithms in the literature. The results emphasize the capability of the BOA-based robot manipulator to provide the best performance compared to the other techniques.

**Keywords:** butterfly optimization algorithm; co-operative optimization; path tracking; robot manipulator



**Citation:** Elsisi, M.; Zaini, H.G.; Mahmoud, K.; Bergies, S.; Ghoneim, S.S.M. Improvement of Trajectory Tracking by Robot Manipulator Based on a New Co-Operative Optimization Algorithm. *Mathematics* **2021**, *9*, 3231. <https://doi.org/10.3390/math9243231>

Academic Editors: Chuangyin Dang, Siyang Gao and Ioannis G. Tsoulos

Received: 18 November 2021  
Accepted: 11 December 2021  
Published: 14 December 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/>).

## 1. Introduction

The robot-manipulator dynamics are driven by a set of high, nonlinear and hardly coupled differential equations. Thus, in order to design the controller based on the traditional tuning methods and to provide accurate motion for the manipulator, a complicated mathematical formulation of the optimization problem is required. In addition, the trajectory variation during the robot's motion represents a significant challenge of robotic manipulators [1]. However, the simple, decentralized proportional integral derivative (PID) controller can be utilized for each robot-manipulator arm instead of the complicated, centralized torque-computation scheme. Furthermore, the decentralized PID control scheme can decrease the complicated online computation that is associated with the inverse dynamics of the robot manipulator. However, the tuning of the controller gains demonstrates the challenging requirements of the robotic manipulator to perform well.

Recently, various control approaches have been devoted to controlling the movement of the robotic manipulator. In [2], a robust PID controller was designed based on Lyapunov's direct theory for cable-driven parallel robots. A wearable robotic system was utilized with a PID controller in order to guarantee the asymptotic stability of the robot in [3]. In [4], A PID controller was applied to a quadrotor based on the analysis of the dynamic characteristics of the robotic system. However, the applied PID controller in these works was tuned based on the trial-and-error method. In [5], adaptive fuzzy logic (FL) was utilized to adjust the robot-controller parameters and motion trajectories. A type-2 fuzzy logic with neural networks was optimized by particle-swarm optimization (PSO) for agricultural robots in [6]. Thus, the FL requires a fine adjusting for the membership functions, which complicates its implementation. In [7], a robotic manipulator based on neural networks (NNs) was reported for different applications. In [8], a distributed NN scheme was utilized for multiple redundant manipulators with co-operative control. An adaptive tracking was performed based on NNs with a robust compensator for robot manipulators [9].

However, the NN requires an adjustable dataset in order to create the predictive model for the training and validation processes. Besides, the computational process requires a high-speed microprocessor during the training and validation processes. Among the previous control techniques, the simple implementation of the PID controller is the main reason that it is the popular controller in most industrial systems [10,11]. Thus, the gains of the PID controller require fine-tuning in order to yield a good performance. Various conventional tuning methods such as Ziegler Nichols (ZN) and graphical systems adjust the PID controller gains in the literature [12,13]. The ZN is structured based on fixed rules by which all systems tune the PID gains that would otherwise fail in most applications in order to yield a good response [14]. On the other hand, the graphical methods require a complicated formulation, especially in large systems, in order to solve the tuning problem.

Furthermore, graphical methods require a long computational time and are non-optimal [15]. Recently, artificial intelligence (AI) techniques effectively solved the optimization issues in various engineering applications [16]. In [17], an optimal tuning of the controller of the robotic manipulator was performed by the genetic algorithm (GA) in order to adjust the output torque of the robot. A developed PSO variant was presented to optimize a fractional PID controller in [18]. In [19], the ant-colony optimization (ACO) was applied to select the proper gains of the fuzzy controller for a mobile robot. The cuckoo search algorithm (CSA) was introduced for the path planning of a mobile robot in [20]. In [21], the CSA was utilized to optimize the parameters of a sliding-mode controller for a robot with multiple degrees of freedom. In [22], a hybrid optimization algorithm was developed based on the GA, PSO, and probabilistic neural network (PNN) for gear-fault diagnosis. In [23–25], ring probabilistic logic neural networks (RPLNN) based on the concept of the PLNN were developed in order to solve the optimization issue [26]. The main problem of these optimization algorithms is issue of trapping at a local optimum.

The contribution of this paper is to suggest a developed optimization algorithm, termed the butterfly optimization algorithm (BOA), for the proper tuning of a robotic manipulator controller. This algorithm requires few adjustable parameters and utilizes the co-operative movement between the butterflies in order to obtain the optimal global solution, as well as to avoid being trapped at a possible local optimum point. The performance of the suggested algorithm is compared to the performances of the GA-based controller [27] and the cuckoo-search-algorithm (CSA)-based controller [28]. Furthermore, the performance of the suggested method is confirmed by carrying out different test scenarios.

The following points are the core contributions of the paper:

- Introducing a new intelligent design for the robotic manipulator controller in order to track linear and nonlinear trajectories.
- The tuning of the controller is performed based on an optimization algorithm with fewer adjustable parameters known as the BOA.

- The proposed BOA is combined with a new figure of demerit objective function in order to handle the minimization of the system overshoot, steady-state error, and settling time in a co-ordinated way.
- The proposed BOA is evaluated with different methods in the literature.
- The results confirm the superiority of the BOA-based robotic manipulator controller to track the linear and nonlinear trajectories with a low steady-state error and system overshoot accepted and a short system-settling time.

The remaining sections of this paper are organized as follows. Section 2 describes an overview of the optimization algorithms. Then, the model of the robotic manipulator is illustrated in Section 3. Next, the results and discussions are presented in Section 4. Finally, Section 5 contains the conclusion of this research.

## 2. Optimization Algorithms-Based Optimality

### 2.1. Overview

Recently, optimization algorithms (OAs) have become widely utilized to tune unknown gains [29,30]. Such OA solutions have been widely applied to engineering applications with promising performances [31–56]. The calculation of the OA is devoted to adjusting the proper values of unknown gains in order to provide an acceptable performance for the controller. According to the target, the optimization problem is defined as a single or multi-objective fitness function [57,58]. The inputs and outputs of the system are combined by mathematical formulation to describe the fitness function [59,60]. The OA investigates the proper values of the controller gains within a predefined limit in regards to the controller and system dynamics. The nonlinear and hardly coupled differential equations of the robotic manipulator make the tuning of the controller more difficult. The OA provides an effective solution for this issue in various control applications for conventional methods [61,62]. The solution process of the OA ensures the determination of the optimal global solutions for complex optimization models [63–69]. The use of statistical tests was investigated in [70] for comparing swarm and evolutionary computing algorithms. The adjustable gains and the trapping at local solutions represent the main issues against the implementation of the OA to tune the controller gains. In [71–73], the PID tuning was performed using the BOA for different engineering problems. This paper proposes the BOA as a recent OA that requires few adjustable parameters. Furthermore, the co-operation between different agents is utilized in the BOA in order to expand the expected exploration behavior that enhances the global search and decreases the trapping possibility at a particular local optimum point. The key structure of the proposed BOA is described in the following subsection.

### 2.2. Butterfly Optimization Algorithm Concepts

The butterfly optimization algorithm is a novel AI technique that imitates the foraging behavior of butterflies [74,75]. The co-operation between butterflies is the inspiration for the global-search behavior of this algorithm. The BOA is divided into three phases named the initial, iterative, and final stages. In the initial stage, the algorithm's parameters and the objective function are defined, the initial population is randomly started, and the initial solutions are determined. After the initial population, the iterative stage starts by calculating the fitness function of all the butterflies. Then, the butterflies generate the fragrance according to stimulus intensity as follows:

$$f = cI^a \quad (1)$$

where

$f$	The value of fragrance
$c$	The sensory modality
$I$	The stimulus intensity
$a$	The absorption indicator

According to the fitness function, each butterfly in the algorithm produces a different fragrance in intensity. The algorithm has two search steps, named global and local search, with switching probability (P). In the global search, the butterfly moves to the best butterfly, or solution, as follows:

$$x_i(t+1) = x_i(t) + (r^2 \times g^* - x_i(t)) \times f_i \quad (2)$$

where

$x_i(t)$	The current solution vector
$t$	The current iteration
$i$	The butterfly index
$g^*$	The current best solution
$f_i$ and $r$	The random number within [0, 1]

In the local search, the butterfly moves randomly to the neighboring butterfly, or solution, as follows:

$$x_i(t+1) = x_i(t) + (r^2 \times x_j(t) - x_k(t)) \times f_i \quad (3)$$

where  $x_j(t)$  and  $x_k(t)$  are the neighboring butterflies of the current solution. This local movement between the butterflies increases the exploration manner of the algorithm and prevents it from being trapped at a local optimum solution. The flowchart shown in Figure 1 gives the detailed steps of the BOA. It is important to note that the key purpose of this paper is to handle the trajectory tracking by the robot manipulator, which any available optimization algorithm can solve. We have nominated the recent butterfly optimization algorithm (BOA) as it has a high performance according to several previous publications (see ref. [62]), which means that it can quickly find the optimal global solution. Specifically, PSO and other optimizers have been compared with the BOA in [62], where the superiority of the BOA was proven after considering different optimization problems.

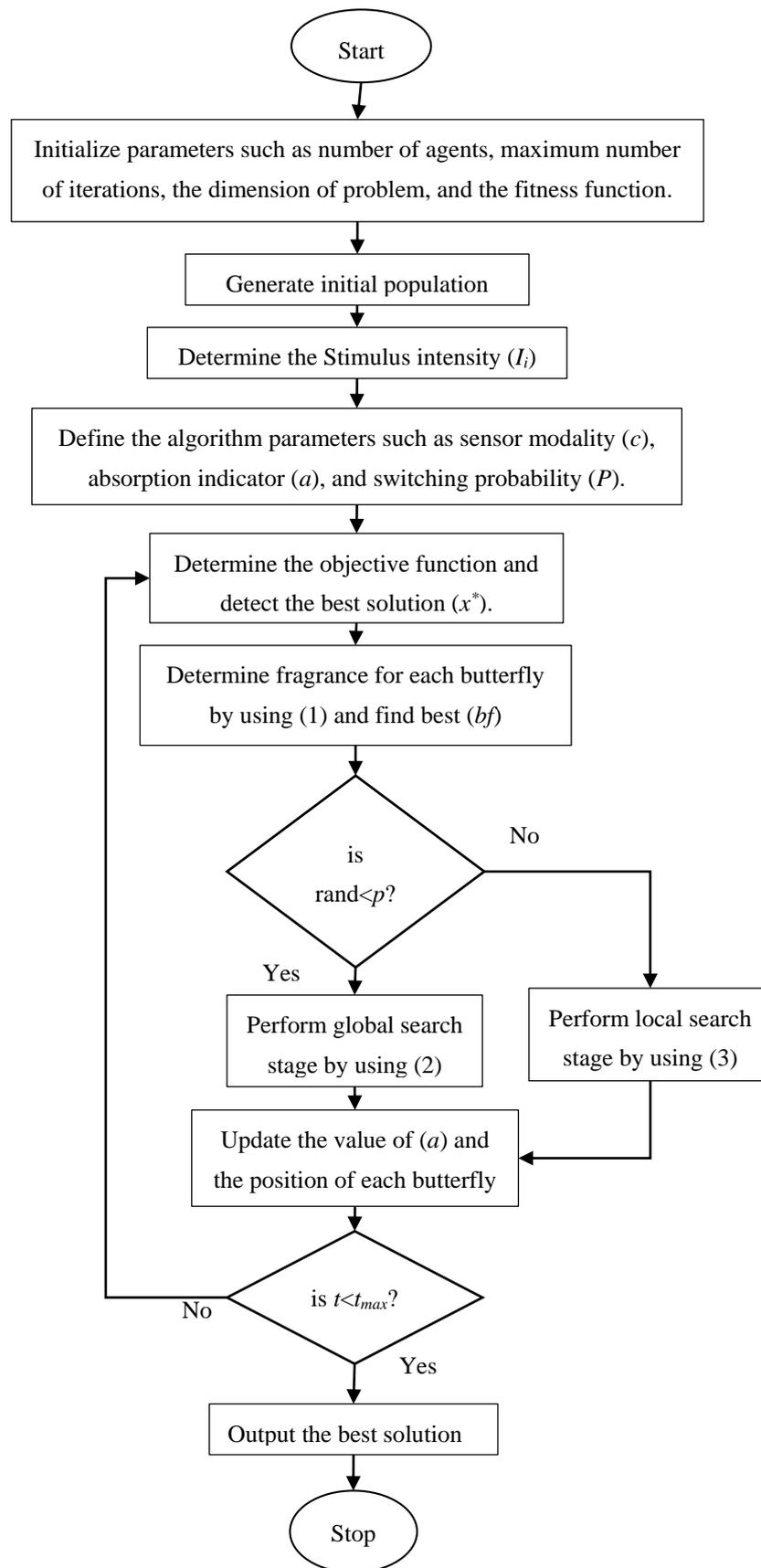


Figure 1. The flowchart of the BOA.

### 3. Robotic Manipulator Modeling

A set of differential equations are utilized to describe the robot dynamics. These equations consist of different terms named inertia, torque, load, and gravity. The movements of the links in a defined trajectory with a certain speed require that an appropriate torque be applied to the actuator of the links. The modeling of the manipulator, which represents the robot dynamics of n-links, is governed by the following nonlinear equations [27]:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \tag{4}$$

where

- $\tau$  Vector of links torques
- $M(\theta)$  Positive matrix
- $C(\theta, \dot{\theta})$  Vector of Coriolis torques
- $G(\theta)$  Vector of gravity torques
- $\theta$  Angular displacement of links
- $\dot{\theta}$  Velocity of links
- $\ddot{\theta}$  Acceleration of links
- $n$  Links number

The robotic manipulator utilized in this paper is a two-degrees-of-freedom robotic manipulator and it has 'n = 2' of links. The dynamics equations that represent these manipulator links, which are shown in Figure 2, are described as follows [75]:

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) \cos(\theta_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_2^2 - 2m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2) + m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos(\theta_1) \tag{5}$$

$$\tau_2 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_2) + m_2 l_1 l_2 \dot{\theta}_1^2 \cos(\theta_2) + m_2 l_1 g \cos(\theta_1 + \theta_2) \tag{6}$$

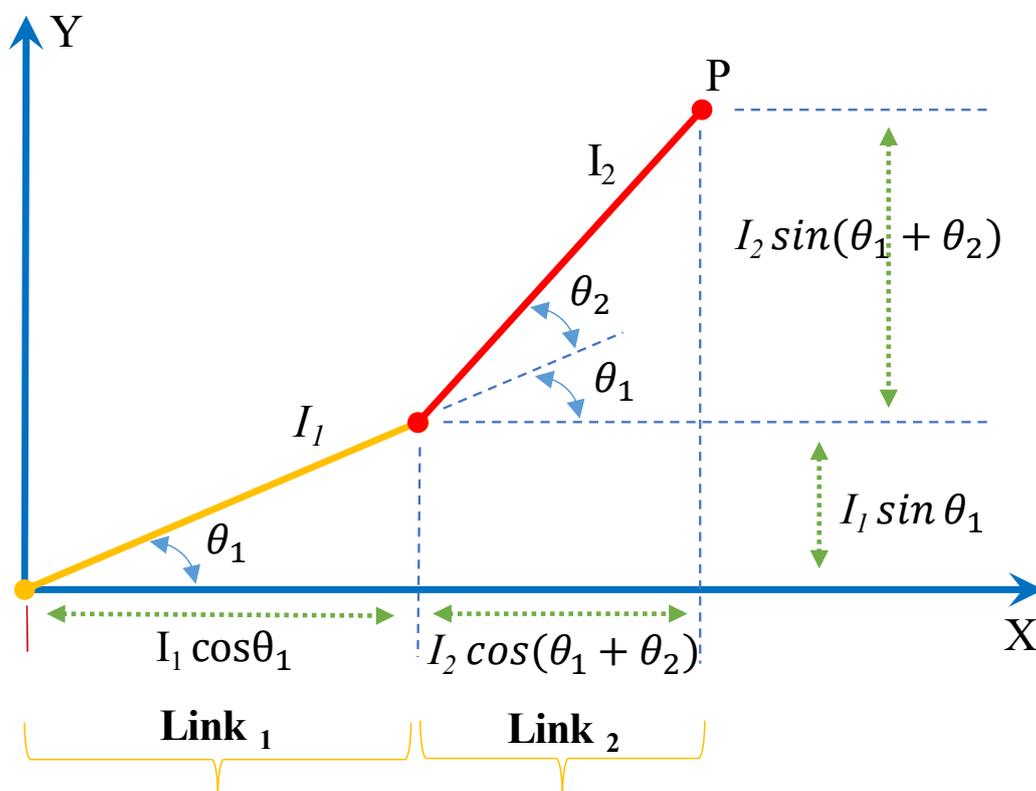


Figure 2. The geometric representation of two degrees of freedom robotic manipulator.

The proper torque for each link is adjusted by the output control signal based on the PID, as described in the following equation:

$$\tau_i = K_{P,i} \times er_i + K_{I,i} \int er_i dt + K_{D,i} \times \frac{der_i}{dt}, \quad i = 1, 2 \quad (7)$$

$$er_i = \theta_{d,i} - \theta_i \quad (8)$$

where  $er_i$  represents the error signal and  $\theta_{d,i}$  characterizes the desired reference trajectory. Note that  $\theta_i$  characterizes the output angular displacement.

#### 4. Results and Discussions

The tracking issue of both linear and nonlinear trajectories is the main target of the robotic manipulator movement. Subsequently, a particular robotic manipulator necessitates the proper torque for its links to track the reference trajectories. The PID controller is proposed as a simple controller to adjust the torque of each link. Thus, the PID gains require fine-tuning in order to provide good performance with less settling time and steady-state error, as well as a low system overshoot. This paper introduces a developed figure of demerit objective function that can handle the minimization of the system overshoot, steady-state error, and settling time in a co-ordinated way. Therefore, the anticipated BOA is tasked with tuning the links' controllers to yield suitable gains in order to track trajectories with less error, settling time, and maximum overshoot based on a reduction of the developed figure of demerit objective function. This function is represented as follows:

$$J = \sum_{i=1}^2 (1 - e^{-\beta})(M_{P,i} + E_{SS,i}) + e^{-\beta}(t_{s,i} - t_{r,i}) \quad (9)$$

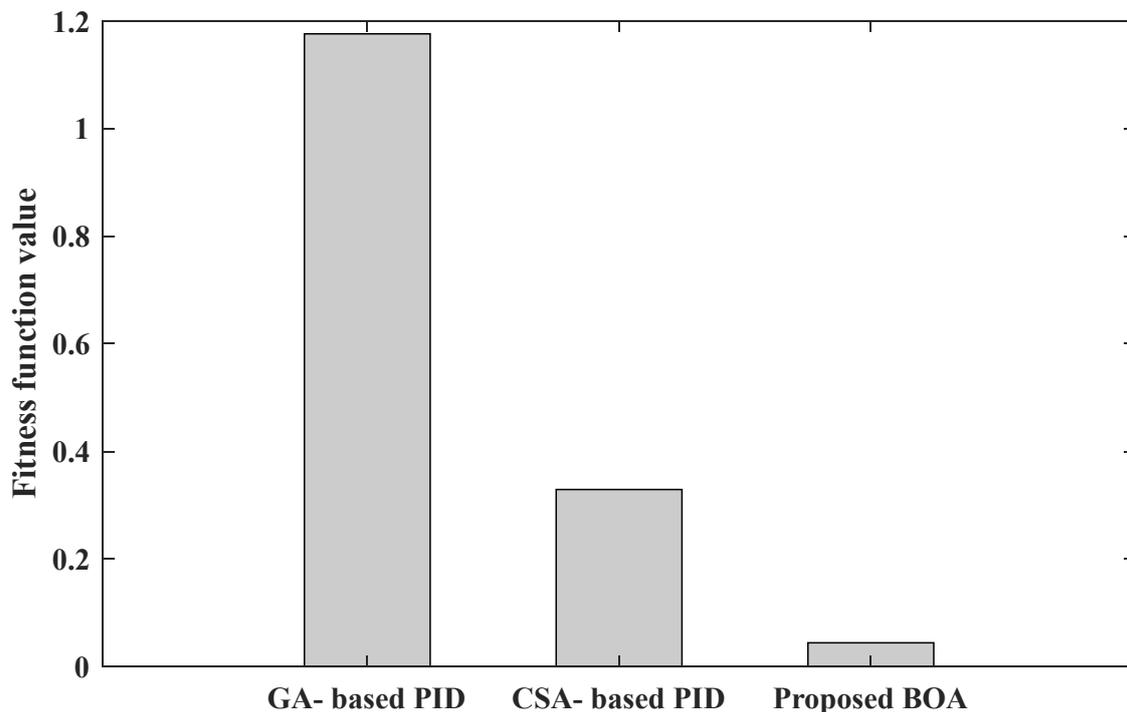
where,  $E_{SS,i}$ ,  $t_{s,i}$ , and  $M_{P,i}$  represent the steady-state error, settling time, and the maximum overshoot of the response due to each link, respectively. Additionally,  $\beta$  is utilized a weighting factor to stabilize the minimization of two parts of the figure of demerit objective function. In this work,  $\beta$  is made equal to 0.7 because, at this point the weighting  $(1 - e^{-\beta}) \approx e^{-\beta}$  and the BOA will equilibrate the minimization of the steady-state error, settling time of the system, and the maximum overshoot of the system. If  $\beta$  is less than 0.7, the BOA will focus on minimizing the settling time. Otherwise, if  $\beta$  is higher than 0.7, the BOA will focus on minimizing the overshoot. The BOA performed the tuning of the controller gains at the nominal parameters of the robotic manipulator, the limits of the controller's gain  $[K_{P,i}; K_{I,i}; K_{D,i}]$  for each link are (lower limit = [0;0;0] and upper limit = [250;1;20]). The BOA parameters of the system are as follows: the extreme number of the possible agents is designated to be 100 while adopting an iteration number of 50. The optimization is performed at a linear unit-step trajectory in order to determine the steady-state error, settling time, and the maximum overshoot of the response due to each link. The results are driven after around 30 runs. The system parameters are: (1) the  $m_1$  and  $m_2$  values are 0.1 kg; (2) the  $l_1$  value is 0.8 m; (3) the  $l_2$  value is 0.4 m; and (4) the  $g$  value is 9.81 m/s<sup>2</sup> [27]. The proposed BOA-based PID controller is compared with the GA-based PID controller from [27] and the CSA-based PID controller from [28]. The controller parameters and the corresponding value of the objective function ( $J$ ) are recorded in Table 1. Figure 3 presents the value of the objective function due to each technique in the bar chart as an illuminated comparison technique. The optimization operation is performed by a MATLAB R2019b intel CORE i7 and 8g RAM computer. Note that the procedure of the BOA to tune the controller gains is concluded in the pseudo-code as follows (Algorithm 1 [62]):

**Algorithm 1.** The pseudo-code of the proposed BOA to adjust the controller gains

- 1: Start BOA
- 2: Create the reference trajectory for each link of the robot manipulator
- 3: Execute the robot-manipulator model by the proposed controller
- 4: Estimate the fitness function
- 5: Update the positions of the agents
- 4: while ( $t < t_{max}$ )
  - 5: Do the steps of BOA as presented in Figure 1
  - 6: Run the robot-manipulator model with the controller
  - 7: Evaluate the fitness function
  - 8: Obtain the best fitness
  - 9: Update the position of the agents
- 10: end while
- 11: Stop

**Table 1.** The controller gains of each technique and the corresponding value of the objective function (J).

Gains	GA-Based PID Controller			CSA-Based PID Controller			Proposed BOA		
	$K_P$	$K_I$	$K_D$	$K_P$	$K_I$	$K_D$	$K_P$	$K_I$	$K_D$
Link <sub>1</sub>	184.76	49.68	8.94	782.417	225.2123	35.1995	249.388	0.4896	11.9204
Link <sub>2</sub>	11.46	16.54	0.2	324.523	119.245	20.1025	192.4835	0.3178	4.3558
J		1.1758			0.3292			0.0443	



**Figure 3.** The value of the objective function due to each technique.

It is clear from Figure 2 and Table 1 that the introduced BOA can minimize the objective function better than the GA from [27] and the CSA from [28]. Table 2 compares the Integral of Absolute Error (IAE) and the Integral of Squared Error (ISE) performance index values based on each algorithm. As noticed, the IAE and ISE values by the proposed BOA are lower than those of the other two controllers. Likewise, the introduced BOA has lower values of tuning factor when compared to the other algorithms (see Table 3), which

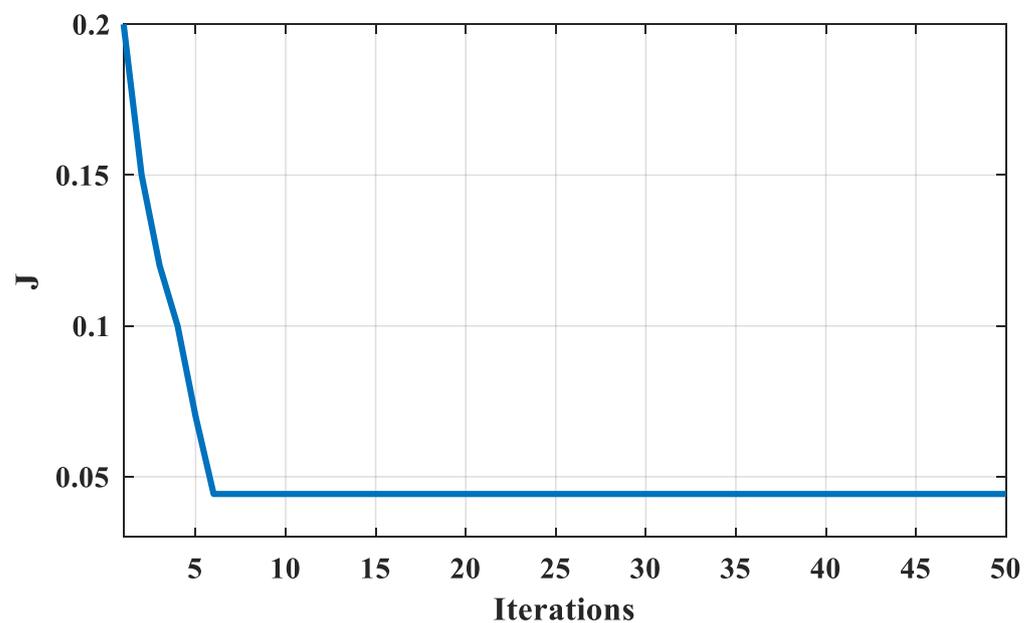
enhance the performance of the robot with a fast convergence rate at 6 iterations, as shown in Figure 4, which was less than the CSA-based PID [28] which reach after approximately 44 iterations. However, the multi-objective GA utilized in [27] has a different convergence curve. Note that the effectiveness of the proposed technique is confirmed by applying the following test scenarios:

**Table 2.** The IAE and ISE performance indexes values based on each algorithm.

		GA-Based PID Controller	CSA-Based PID Controller	Proposed BOA
IAE	Link <sub>1</sub>	0.0692	0.0522	0.0518
	Link <sub>2</sub>	0.2291	0.0731	0.0290
ISE	Link <sub>1</sub>	0.0366	0.0259	0.0353
	Link <sub>2</sub>	0.1666	0.0328	0.0157

**Table 3.** Adjusted settings for various techniques.

Techniques	Values	Tuning Factors
GA-based PID controller [27]	4	Population size, iterations, crossover, mutation
CSA-based PID [28]	3	Nest size, elitism probability, iterations
Proposed BOA-based PID	2	Agent numbers (100), iterations (50)



**Figure 4.** The convergence curve of the proposed BOA.

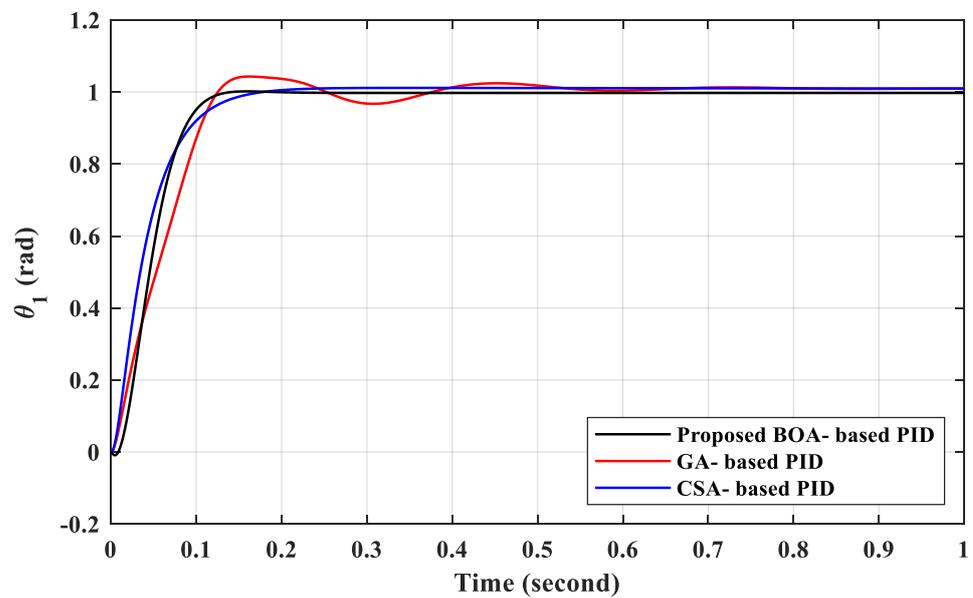
*4.1. Scenario 1: Step Reference Trajectory*

In this Scenario, the effectiveness of the introduced BOA based on the developed figure of demerit objective function is confirmed by applying the unit-step input as a reference trajectory for each link. Figures 5 and 6 show the output responses of the robotic manipulator due to each method. The maximum overshoots and the settling times of the output responses of each link are recorded in Table 4. Figure 7 presents the maximum overshoots and the settling times of the output responses of each link as a bar chart for more clarification. It is clear that the proposed BOA-based PID controller has a high

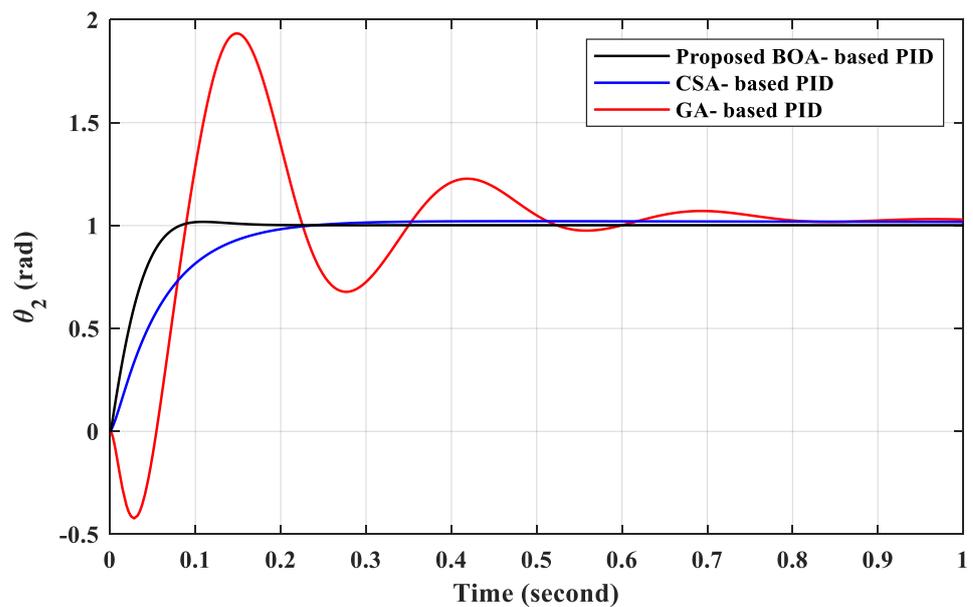
damped performance and less settling time and overshoot when compared to the GA-based controller and the CSA-based controller, as shown in Figures 5–7 and Table 4.

**Table 4.** The supreme overshoot as well as the settling time of the system responses for Scenario 1 due to each technique.

		GA-Based PID Controller	CSA-PID Controller	Proposed BOA
Maximum overshoot	Link <sub>1</sub>	4.301%	1.1421%	0.1791%
	Link <sub>2</sub>	93.3058%	2.1193%	1.8101%
Settling time	Link <sub>1</sub>	0.4899	0.1404	0.1138
	Link <sub>2</sub>	1	0.694	0.0733



**Figure 5.** The output response of link1 in the case of Scenario 1.



**Figure 6.** The output response of link2 in the case of Scenario 1.

4.2. Scenario 2: Nonlinear Reference Trajectory

This scenario is performed by applying a cubic reference trajectory to each robot link. The cubic reference is generated based on the following formulation [75]:

$$\theta_{d,i} = c_{0,i} + c_{1,i} \times t + c_{2,i} \times t^2 + c_{3,i} \times t^3 \tag{10}$$

where  $c_{0,1} = c_{0,2} = 0$ ,  $c_{1,1} = c_{1,2} = 0$ ,  $c_{2,1} = 0.09375$ ,  $c_{2,2} = 0.75$ ,  $c_{3,1} = -0.015625$ , and  $c_{3,2} = -0.125$  at the assumed desired final positions ' $\theta_{f,1} = 0.5$  rad while  $\theta_{f,2} = 4$  rad' for each link at ' $t_f = 4$  s' that represents the final time. The initial position and velocity are equal to zero.

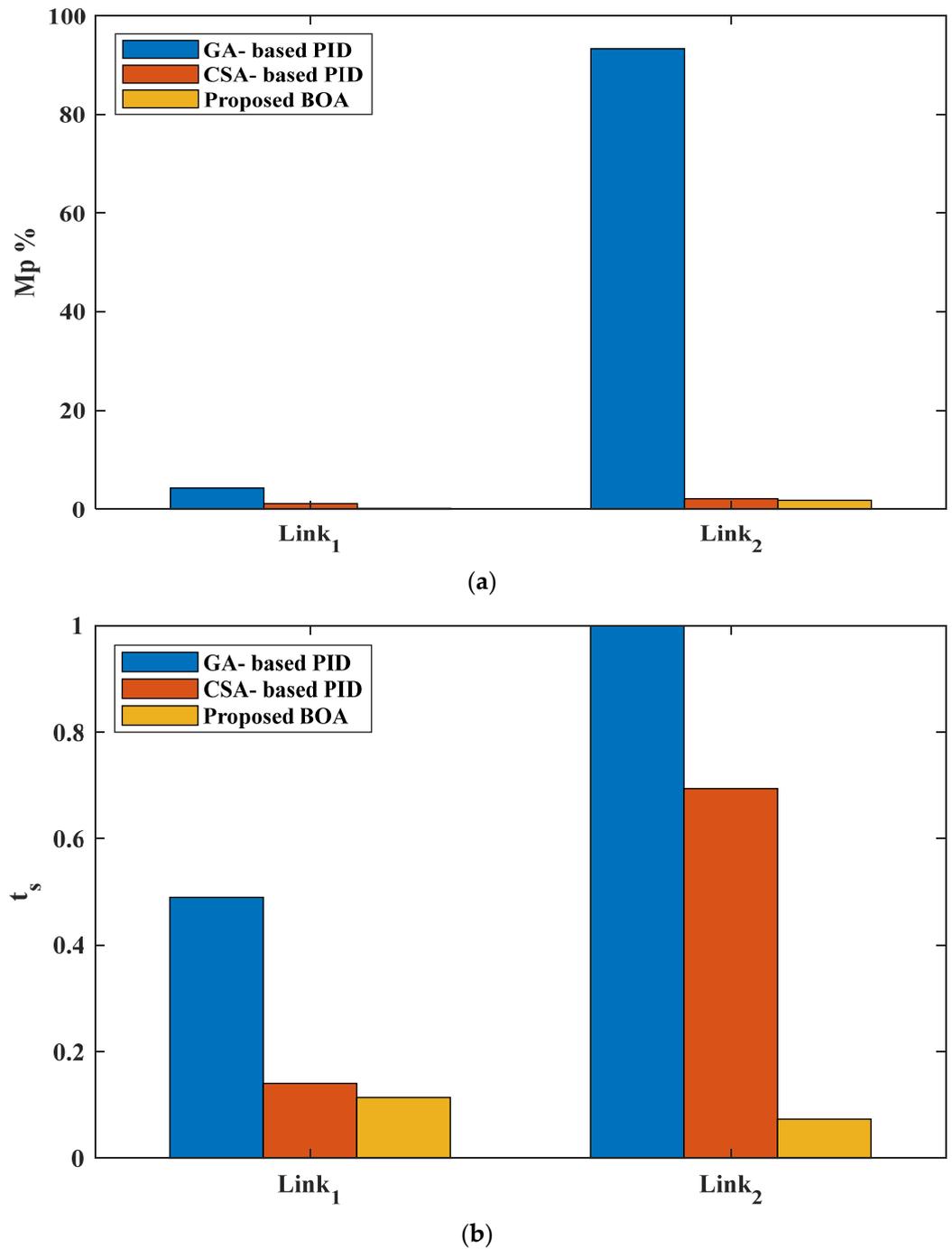


Figure 7. The dynamic characteristics of each technique (a) system extreme overshoot (b) system-settling time.

The generated cubic references at every possible link are presented in Figure 8. Note that the output response of each link due to this scenario is shown in Figures 9 and 10. It is confirmed from these Figures that the introduced BOA-based PID controller successfully tracks the nonlinear cubic reference trajectory with a low error when compared to the GA-based PID controller as well as the CSA-based PID controller.

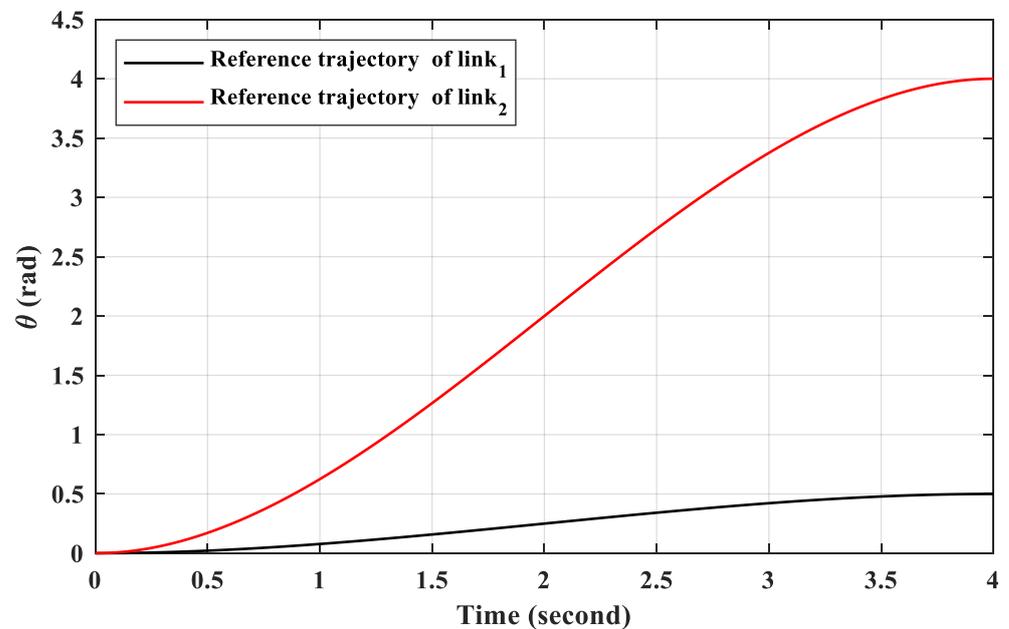


Figure 8. The cubic references for each robot link.

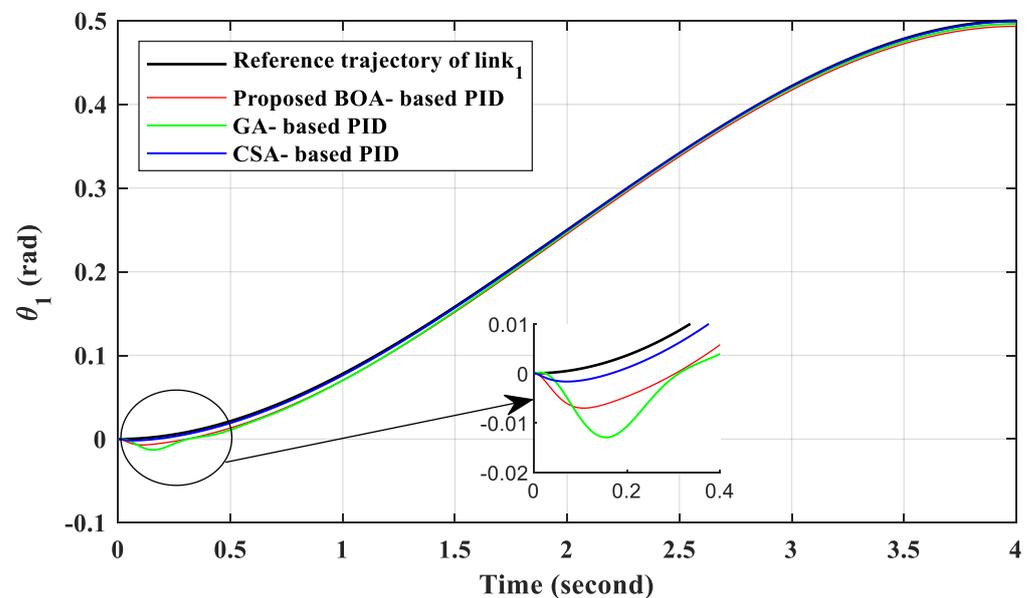


Figure 9. The output position due to link<sub>1</sub> in the case of cubic reference.

4.3. Scenario 3: The Parameters Uncertainty Test for Each Technique

This test is performed to confirm the effectiveness of the PID controller gains based on the proposed BOA against the change of system parameters. The test is carried out by changing the masses as well as the lengths of the robot links by  $\pm 20\%$  considering the rated values. Figures 11 and 12 indicate that the PID controller gains based on the proposed BOA

are effective against the parameters' uncertainty and have a high damped performance when compared to the GA-based PID controller as well as the CSA-based PID controller.

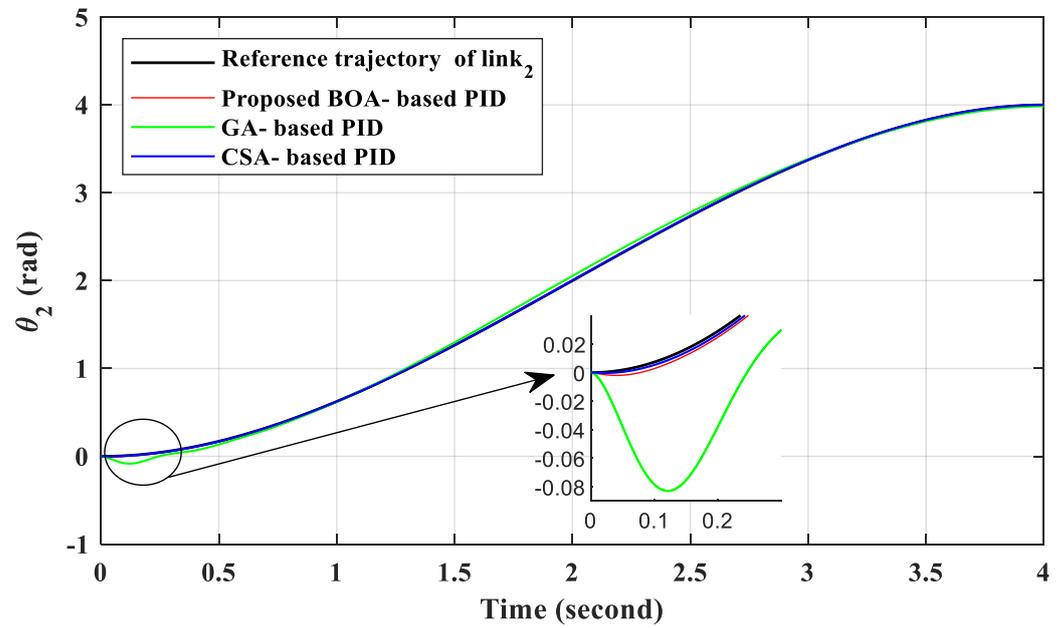
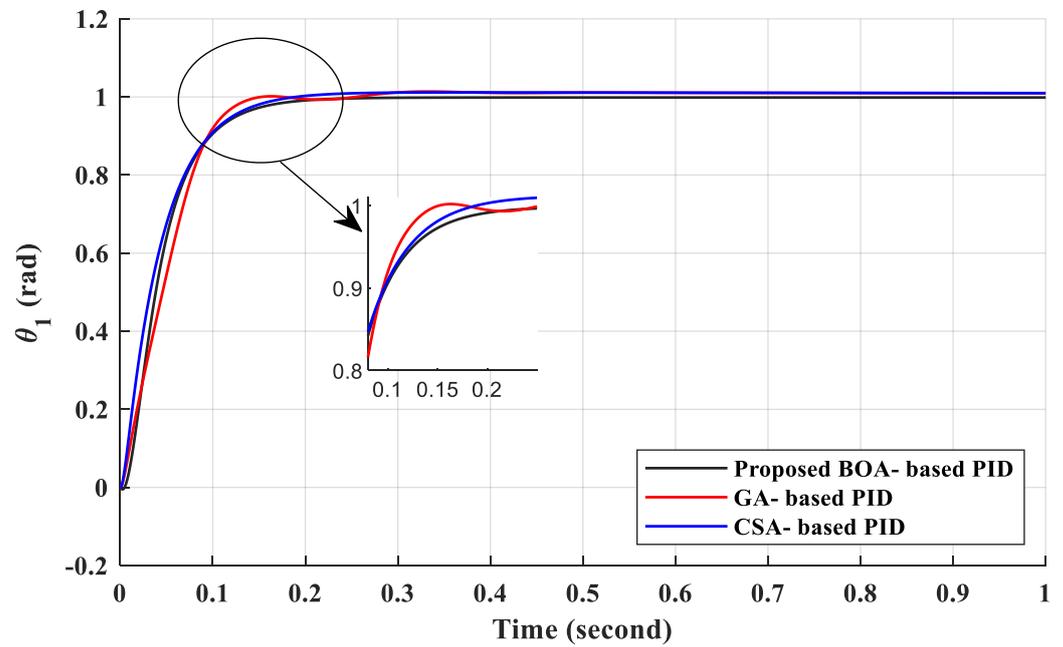
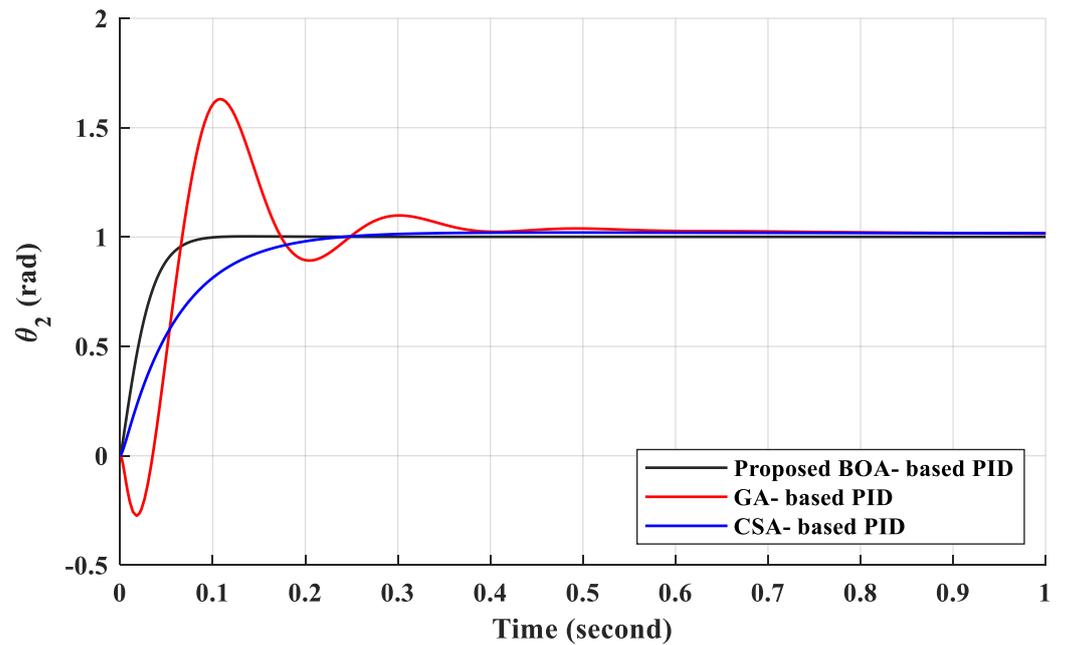


Figure 10. The output position due to link<sub>2</sub> in the case of cubic reference.



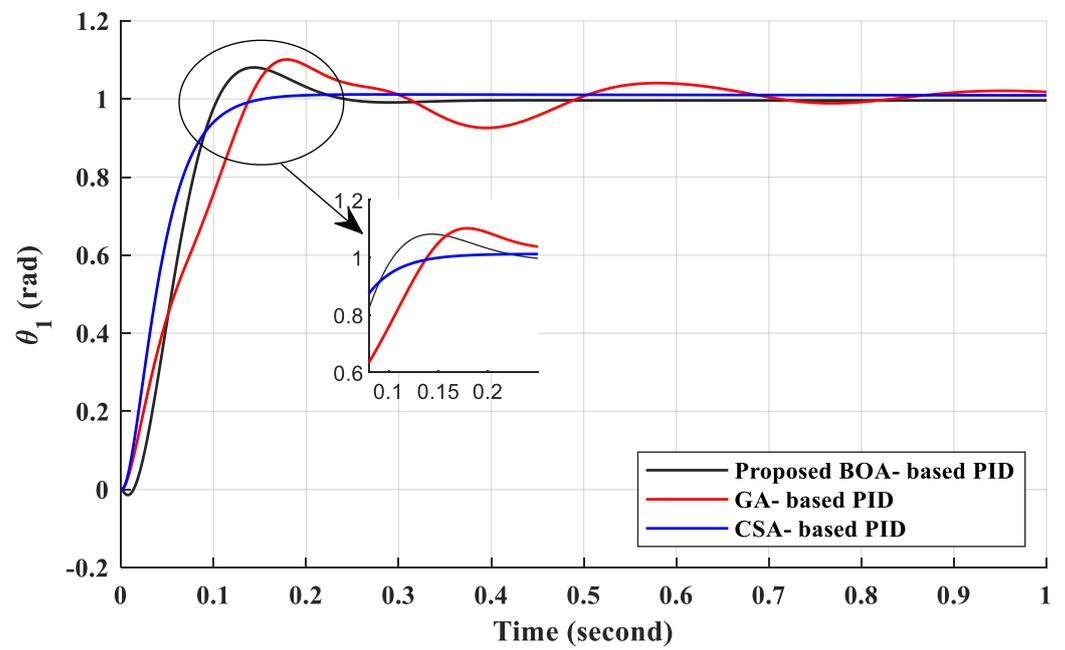
(a)

Figure 11. Cont.



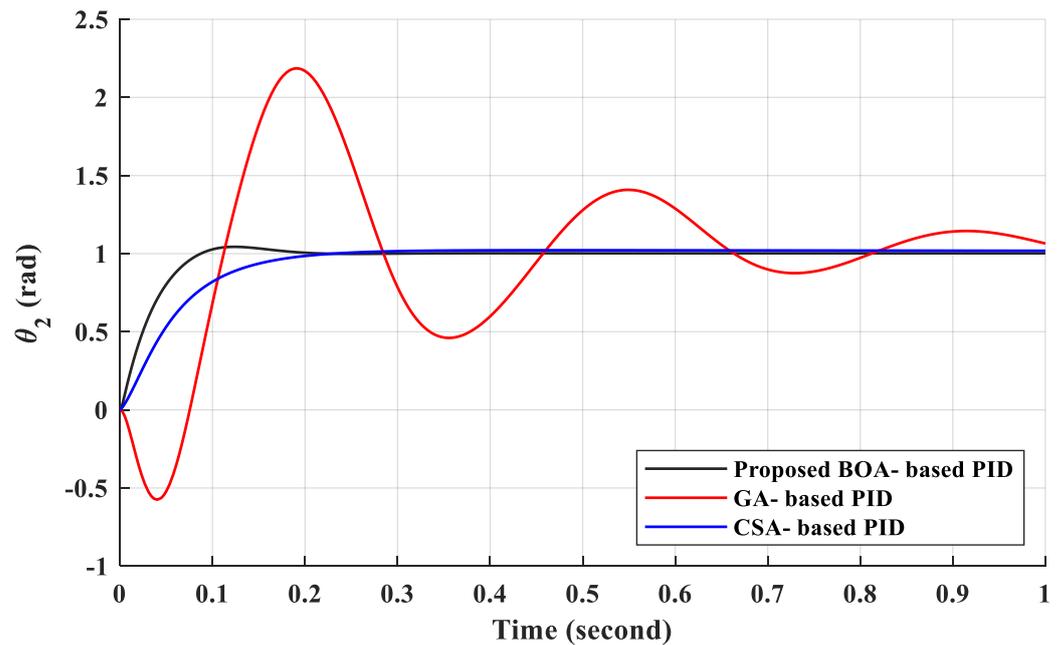
(b)

Figure 11. The output position in case of decreasing the parameters by  $-20\%$  (a) the position of link<sub>1</sub>, (b) the position of link<sub>2</sub>.



(a)

Figure 12. Cont.



(b)

Figure 12. The output position of in case of increasing the parameters by +20% (a) the position of link<sub>1</sub>, (b) the position of link<sub>2</sub>.

4.4. Scenario 4: Random Trajectory

The inspired technique has been tested in this scenario to track several position trajectories. Specifically, the assessment is carried out in two steps. The first step is generated by creating a random step-position reference to each arm, as shown in Figure 13. Figures 14 and 15 depict the system output due to this step. Compared to previous approaches, these data show that the suggested BOA-based controller can track the random step reference with low steady-state error, the shortest settling time, and insignificant overshoots.

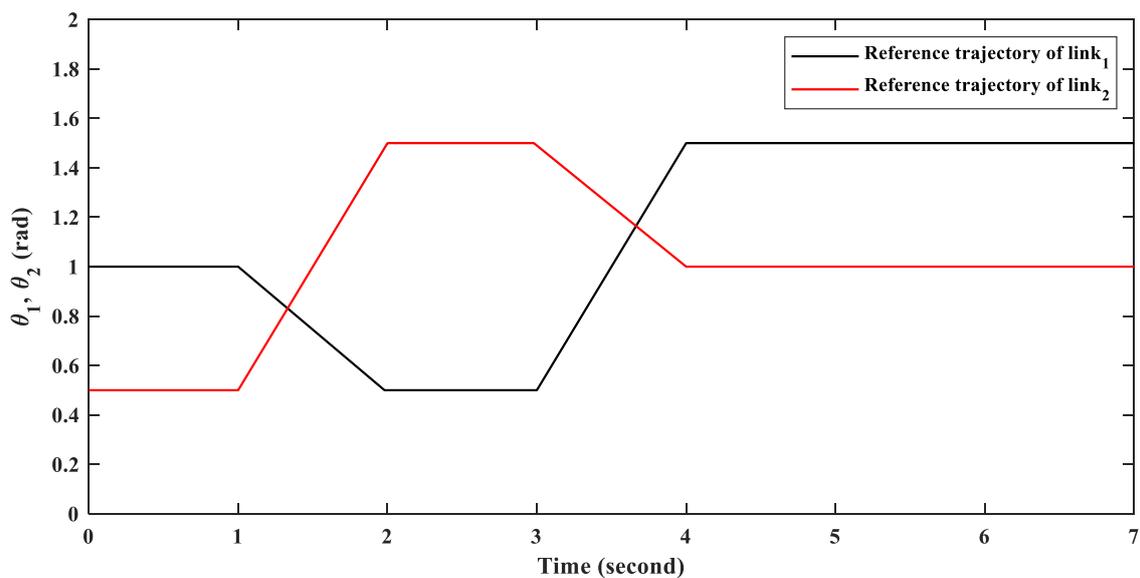
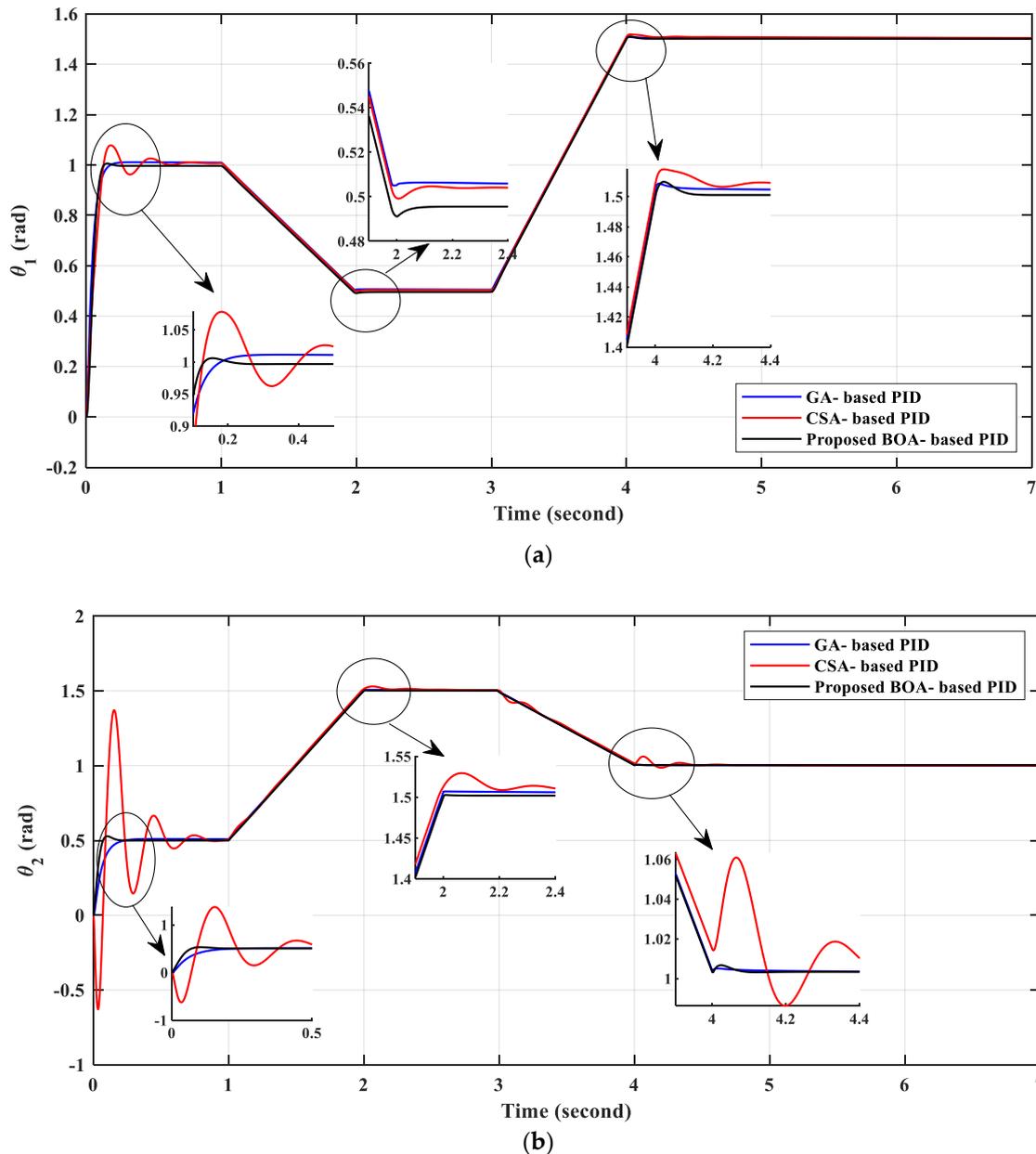


Figure 13. Random step-position trajectories of the two links.



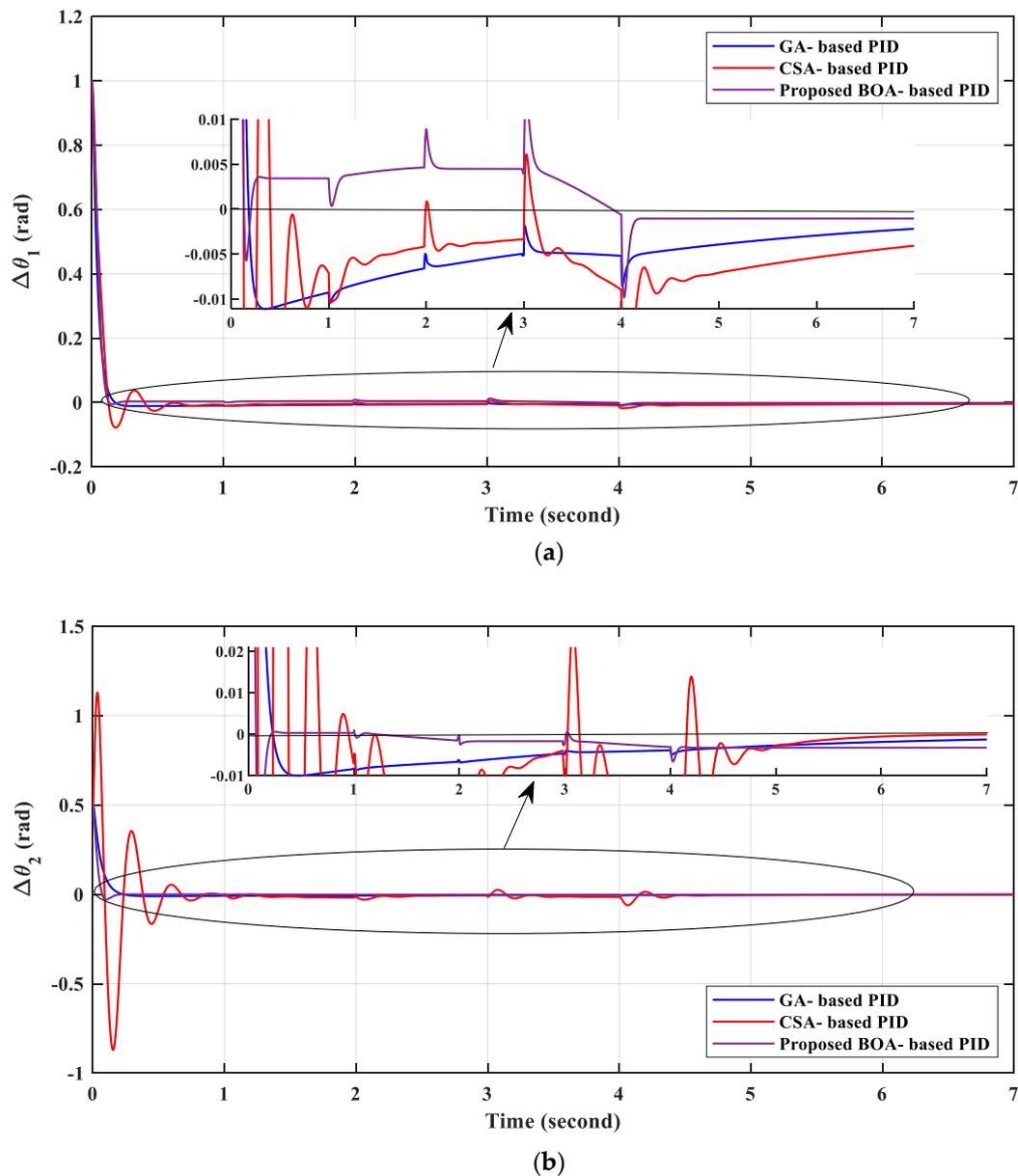
**Figure 14.** The link<sub>1</sub> output wave of as a result of random step-position trajectories; (a) link<sub>1</sub> position (b) link<sub>1</sub> position deviation.

4.5. Summary

The main discussions of the above test scenarios are concluded in the following points:

- The robotic manipulator effectively tracked the linear trajectory based on the introduced BOA-based PID controller, as presented in scenario 1. Furthermore, the output position response of each link had less settling time and overshoot based on the introduced BOA-based PID controller compared to other techniques.
- The introduced BOA-based PID controller successfully tracked the nonlinear trajectory with a low error, as presented in scenario 2. Moreover, the introduced BOA-based PID controller had a better damped response than both the GA-based PID and CSA-based PID control schemes.
- The designed PID controller gains based on the introduced BOA were tested against the robotic manipulator parameter uncertainties, as clarified in scenario 3. The re-

sults of this scenario confirm the superiority of the designed PID controller gains based on the introduced BOA to provide high damped performance against the uncertainty of the parameters in regards to both the GA-based PID and CSA-based PID control schemes.



**Figure 15.** The link<sub>2</sub> output wave of as a result of random step-position trajectories; (a) link<sub>2</sub> position (b) link<sub>2</sub> position deviation.

## 5. Conclusions

This paper introduces a recently developed global optimization algorithm named the BOA for tuning the robotic manipulator controller gains. The proposed BOA is tasked with adjusting the controllers' gains to minimize an innovative developed figure of demerit objective function. Specifically, the developed figure of demerit objective function can handle the problem of the reduction of the system steady-state error, system-settling time, and the system overshoot in a co-ordinated manner. The introduced the BOA-based PID controller is compared with two previous techniques named the GA-PID and the CSA-PID control schemes. The simulation results emphasize that the introduced BOA-based PID controller has a high damped and effective performance with a maximum overshoot and

settling time less than 1.8101% and 0.1138s, respectively, compared with other methods in the tracking of linear and nonlinear trajectories. Furthermore, the proposed BOA based on the introduced figure of demerit objective function can be utilized in different control systems of future works in order to simultaneously enhance the system response with less settling time and fewer oscillations.

**Author Contributions:** Conceptualization, M.E., H.G.Z., K.M., S.B. and S.S.M.G.; Data curation, M.E. and S.B.; Formal analysis, M.E., K.M. and S.B.; Funding acquisition, H.G.Z. and S.S.M.G.; Investigation, H.G.Z.; Methodology, M.E., K.M. and S.B.; Project administration, H.G.Z. and S.S.M.G.; Resources, M.E., H.G.Z. and S.S.M.G.; Software, M.E., K.M. and S.B.; Supervision, M.E. and K.M.; Validation, M.E.; Visualization, H.G.Z., K.M. and S.S.M.G.; Writing—original draft, M.E., K.M. and S.B.; Writing—review & editing, H.G.Z. and S.S.M.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Taif University Researchers Supporting Project under Grant TURSP-2020/345. Taif. Saudi Arabia, and also the Ministry of Science and Technology (MOST) of Taiwan (grant number: MOST 110-2222-E-011-013-) and the Center for Cyber-physical System Innovation from the Featured Areas Research Center Program in the Agenda of the Ministry of Education (MOE), Taiwan.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to acknowledge the financial support received from Taif University Researchers Supporting Project Number (TURSP-2020/345). Taif University, Taif, Saudi Arabia.

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

## References

1. Kröger, T. Literature Survey: Trajectory Generation in and Control of Robotic Systems. In *On-Line Trajectory Generation in Robotic Systems*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 11–31.
2. Khosravi, M.A.; Taghirad, H.D. Robust PID control of fully-constrained cable driven parallel robots. *Mechatronics* **2014**, *24*, 87–97. [[CrossRef](#)]
3. Yu, W.; Rosen, J. A novel linear PID controller for an upper limb exoskeleton. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 3548–3553.
4. Li, J.; Li, Y. Dynamic analysis and PID control for a quadrotor. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August 2011; pp. 573–578.
5. Ardila, L.R.; Coronado, E.; Hendra, H.; Phan, J.; Zainalkefli, Z.; Venture, G. Adaptive Fuzzy and Predictive Con-trollers for Expressive Robot Arm Movement during Human and Environment Interaction. *Int. J. Mech. Eng. Robot. Res.* **2019**, *8*, 207–219. [[CrossRef](#)]
6. Camci, E.; Kripalani, D.R.; Ma, L.; Kayacan, E.; Khanesar, M.A. An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm. *Swarm Evol. Comput.* **2018**, *41*, 1–8. [[CrossRef](#)]
7. Jin, L.; Li, S.; Yu, J.; He, J. Robot manipulator control using neural networks: A survey. *Neurocomputing* **2018**, *285*, 23–34. [[CrossRef](#)]
8. Jin, L.; Li, S.; Luo, X.; Li, Y.; Qin, B. Neural Dynamics for Cooperative Control of Redundant Robot Manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3812–3821. [[CrossRef](#)]
9. Van Truong, L.; Huang, S.D.; Yen, V.T.; Van Cuong, P. Adaptive Trajectory Neural Network Tracking Control for Industrial Robot Manipulators with Deadzone Robust Compensator. *Int. J. Control Autom. Syst.* **2020**, *18*, 2423–2434. [[CrossRef](#)]
10. Dantas, A.D.O.D.S.; Dantas, A.F.O.D.A.; Campos, J.T.L.; de Almeida Neto, D.L.; Dórea, C.E.T. PID Control for Electric Vehicles Subject to Control and Speed Signal Constraints. *J. Control Sci. Eng.* **2018**, *2018*, 1–11. [[CrossRef](#)]
11. Moreno-Valenzuela, J.; Pérez-Alcocer, R.; Guerrero-Medina, M.; Dzul, A. Nonlinear PID-type controller for quad-rotor trajectory tracking. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 2436–2447. [[CrossRef](#)]
12. Åström, K.J.; Hägglund, T. Revisiting the Ziegler–Nichols step response method for PID control. *J. Process Control* **2004**, *14*, 635–650. [[CrossRef](#)]
13. Srivastava, S.; Pandit, V.S. A PI/PID controller for time delay systems with desired closed loop time response and guaranteed gain and phase margins. *J. Process Control* **2016**, *37*, 70–77. [[CrossRef](#)]

14. Tavakoli, S.; Tavakoli, M. Optimal tuning of PID controllers for first order plus time delay models using di-mensional analysis. In Proceedings of the 2003 4th International Conference on Control and Automation Proceedings, Montreal, QC, Canada, 12 June 2003; pp. 942–946.
15. Wu, H.; Su, W.; Liu, Z. PID controllers: Design and tuning methods. In Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2014; pp. 808–813.
16. Patle, B.K.; Pandey, A.; Parhi, D.R.K.; Jagadeesh, A. A review: On path planning strategies for navigation of mobile robot. *Def. Technol.* **2019**, *15*, 582–606. [[CrossRef](#)]
17. Sangdani, M.H.; Tavakolpour-Saleh, A.R.; Lotfavar, A. Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment. *Eng. Appl. Artif. Intell.* **2018**, *67*, 24–38. [[CrossRef](#)]
18. Aghababa, M.P. Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm. *Soft Comput.* **2016**, *20*, 4055–4067. [[CrossRef](#)]
19. Juang, C.-F.; Jeng, T.-L.; Chang, Y.-C. An Interpretable Fuzzy System Learned Through Online Rule Generation and Multiobjective ACO With a Mobile Robot Control Application. *IEEE Trans. Cybern.* **2015**, *46*, 2706–2718. [[CrossRef](#)]
20. Mohanty, P.K.; Parhi, D.R. Optimal path planning for a mobile robot using cuckoo search algorithm. *J. Exp. Theor. Artif. Intell.* **2016**, *28*, 35–52. [[CrossRef](#)]
21. Moezi, S.A.; Rafeeyan, M.; Ebrahimi, S. Sliding mode control of 3-RPR parallel robot on the optimal path using cuckoo optimization algorithm. *Modares Mech. Eng.* **2015**, *15*, 147–158.
22. Ding, J.; Xiao, D.; Li, X. Gear Fault Diagnosis Based on Genetic Mutation Particle Swarm Optimization VMD and Probabilistic Neural Network Algorithm. *IEEE Access* **2020**, *8*, 18456–18474. [[CrossRef](#)]
23. Seyfipour, N.; Menhaj, M.; Nik, R.S. A New Optimization Method by Ring Probabilistic Logic Neural Networks. *AMIRKABIR* **2003**, *14*, 43–57.
24. Azizi, A.; Barenji, A.V.; Hashmipour, M. Optimizing radio frequency identification network planning through ring probabilistic logic neurons. *Adv. Mech. Eng.* **2016**, *8*. [[CrossRef](#)]
25. Menhaj, M.B.; Seifipour, N. Function optimization by RPLNN. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1522–1527.
26. Qu, M.; Tang, J. Probabilistic logic neural networks for reasoning. *arXiv* **2019**, arXiv:1906.08495.
27. Ayala, H.V.H.; Coelho, L.D.S. Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Syst. Appl.* **2012**, *39*, 8968–8974. [[CrossRef](#)]
28. Chhabra, H.; Mohan, V.; Rani, A.; Singh, V. Multi-objective Cuckoo Search Algorithm-Based 2-DOF FOPD Con-troller for Robotic Manipulator. In *Advances in Signal Processing and Communication*; Springer: Singapore, 2019; pp. 345–352.
29. Goli, A.; Tirkolaee, E.B.; Aydin, N.S. Fuzzy integrated cell formation and production scheduling considering automated guided vehicles and human factors. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3686–3695. [[CrossRef](#)]
30. Elsis, M. Optimal design of nonlinear model predictive controller based on new modified multitracker optimization algorithm. *Int. J. Intell. Syst.* **2020**, *35*, 1857–1878. [[CrossRef](#)]
31. Abdullah, J.M.; Ahmed, T. Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. *IEEE Access* **2019**, *7*, 43473–43486. [[CrossRef](#)]
32. Pierezan, J.; Coelho, L.D.S. Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [[CrossRef](#)]
33. Elsis, M. New variable structure control based on different meta-heuristics algorithms for frequency regulation con-sidering nonlinearities effects. *Int. Trans. Electr. Energy Syst.* **2020**, *30*, e12428. [[CrossRef](#)]
34. Zhao, W.; Wang, L.; Zhang, Z. Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization. *IEEE Access* **2019**, *7*, 73182–73206. [[CrossRef](#)]
35. Shabani, A.; Asgarian, B.; Salido, M.; Gharebaghi, S.A. Search and rescue optimization algorithm: A new optimization method for solving constrained engi-neering optimization problems. *Expert Syst. Appl.* **2020**, *161*, 113698. [[CrossRef](#)]
36. Moosavi, S.H.S.; Bardsiri, V.K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **2019**, *86*, 165–181. [[CrossRef](#)]
37. Sulaiman, M.H.; Mustaffa, Z.; Saari, M.M.; Daniyal, H. Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimiza-tion problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103330. [[CrossRef](#)]
38. Elsis, M.; Bazmohammadi, N.; Guerrero, J.M.; Ebrahim, M.A. Energy management of controllable loads in mul-ti-area power systems with wind power penetration based on new supervisor fuzzy nonlinear sliding mode control. *Energy* **2021**, *221*, 119867. [[CrossRef](#)]
39. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [[CrossRef](#)]
40. Kamboj, V.K.; Nandi, A.; Bhadoria, A.; Sehgal, S. An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Appl. Soft Comput.* **2020**, *89*, 106018. [[CrossRef](#)]
41. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]

42. Cao, Y.; Wang, Q.; Wang, Z.; Jermsittiparsert, K.; Shafiee, M. A new optimized configuration for capacity and operation improvement of CCHP system based on developed owl search algorithm. *Energy Rep.* **2020**, *6*, 315–324. [[CrossRef](#)]
43. De Vasconcelos Segundo, E.H.; Mariani, V.C.; dos Santos Coelho, L. Design of heat exchangers using Falcon Optimization Algorithm. *Appl. Therm. Eng.* **2019**, *156*, 119–144. [[CrossRef](#)]
44. Elsis, M. New design of robust PID controller based on meta-heuristic algorithms for wind energy conversion system. *Wind Energy* **2020**, *23*, 391–403. [[CrossRef](#)]
45. Tran, M.-Q.; Liu, M.-K.; Elsis, M. Effective multi-sensor data fusion for chatter detection in milling process. *ISA Trans.* **2021**. [[CrossRef](#)] [[PubMed](#)]
46. Mahmoud, K.; Hussein, M.M.; Abdel-Nasser, M.; Lehtonen, M. Optimal Voltage Control in Distribution Systems With Intermittent PV Using Multiobjective Grey-Wolf-Lévy Optimizer. *IEEE Syst. J.* **2020**, *14*, 760–770. [[CrossRef](#)]
47. Elsis, M. Optimal design of non-fragile PID controller. *Asian J. Control* **2021**, *23*, 729–738. [[CrossRef](#)]
48. Mahmoud, K.; Abdel-Nasser, M.; Lehtonen, M.; Hussein, M.M. Optimal Voltage Regulation Scheme for PV-Rich Distribution Systems Interconnected with D-STATCOM. *Electr. Power Compon. Syst.* **2021**. [[CrossRef](#)]
49. Elsis, M.; Aboelela, M.; Soliman, M.; Mansour, W. Design of optimal model predictive controller for LFC of nonlinear multi-area power system with energy storage devices. *Electr. Power Compon. Syst.* **2018**, *46*, 1300–1311. [[CrossRef](#)]
50. Ghoneim, S.M.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Enhancing Diagnostic Accuracy of Transformer Faults Using Teaching-Learning-Based Optimization. *IEEE Access* **2021**, *9*, 30817–30832. [[CrossRef](#)]
51. Elsis, M.; Ebrahim, M.A. Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics. *Int. J. Intell. Syst.* **2021**, *36*, 6968–6987. [[CrossRef](#)]
52. Ismail, M.M.; Bendary, A.F.; Elsis, M. Optimal design of battery charge management controller for hybrid system PV/wind cell with storage battery. *Int. J. Power Energy Convers.* **2020**, *11*, 412–429. [[CrossRef](#)]
53. Elsis, M.; Aboelela, M.A.S.; Soliman, M.; Mansour, W. Model Predictive Control of Two-Area Load Frequency Control Based Imperialist Competitive Algorithm. *TELKOMNIKA Indones. J. Electr. Eng.* **2015**, *16*, 75. [[CrossRef](#)]
54. Tran, M.-Q.; Elsis, M.; Liu, M.-K. Effective feature selection with fuzzy entropy and similarity classifier for chatter vibration diagnosis. *Measurement* **2021**, *184*, 109962. [[CrossRef](#)]
55. Elsis, M.; Tran, M.-Q.; Hasanien, H.M.; Turky, R.A.; Albalawi, F.; Ghoneim, S.S.M. Robust Model Predictive Control Paradigm for Automatic Voltage Regulators against Uncertainty Based on Optimization Algorithms. *Mathematics* **2021**, *9*, 2885. [[CrossRef](#)]
56. Elsis, M. Future search algorithm for optimization. *Evol. Intell.* **2019**, *12*, 21–31. [[CrossRef](#)]
57. Fakhrzad, M.B.; Goodarzi, F. A new multi-objective mathematical model for a Citrus supply chain network design: Meta-heuristic algorithms. *J. Optim. Ind. Eng.* **2021**, *14*, 127–144.
58. Elsis, M.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M. Effective Nonlinear Model Predictive Control Scheme Tuned by Improved NN for Robotic Manipulators. *IEEE Access* **2021**, *9*, 64278–64290. [[CrossRef](#)]
59. Mokhtarzadeh, M.; Tavakkoli-Moghaddam, R.; Triki, C.; Rahimi, Y. A hybrid of clustering and meta-heuristic algorithms to solve a p-mobile hub location-allocation problem with the depreciation cost of hub facilities. *Eng. Appl. Artif. Intell.* **2021**, *98*, 104121. [[CrossRef](#)]
60. Singh, P.; Choudhary, S.K. Introduction: Optimization and Metaheuristics Algorithms. In *Metaheuristic and Evolutionary Computation: Algorithms and Applications*; Springer: Singapore, 2021; pp. 3–33.
61. Blondin, M.J. Optimization Algorithms in Control Systems. In *Controller Tuning Optimization Methods for Multi-Constraints and Nonlinear Systems*; Springer: Cham, Switzerland, 2021; pp. 1–9.
62. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [[CrossRef](#)]
63. Elsis, M.; Soliman, M.; Aboelela, M.; Mansour, W. Improving the grid frequency by optimal design of model predictive control with energy storage devices. *Optim. Control Appl. Methods* **2018**, *39*, 263–280. [[CrossRef](#)]
64. Kashyap, A.K.; Parhi, D.R. Particle Swarm Optimization aided PID gait controller design for a humanoid robot. *ISA Trans.* **2021**, *114*, 306–330. [[CrossRef](#)]
65. Sumar, R.R.; Coelho, A.A.R.; dos Santos Coelho, L. Computational intelligence approach to PID controller design using the universal model. *Inf. Sci.* **2010**, *180*, 3980–3991. [[CrossRef](#)]
66. Dos Santos Coelho, L.; Mariani, V.C. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Comput. Math. Appl.* **2012**, *64*, 2371–2382. [[CrossRef](#)]
67. Elsis, M.; Soliman, M. Optimal design of robust resilient automatic voltage regulators. *ISA Trans.* **2021**, *108*, 257–268. [[CrossRef](#)] [[PubMed](#)]
68. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [[CrossRef](#)]
69. Mahmoud, K.; Lehtonen, M. Comprehensive Analytical Expressions for Assessing and Maximizing Technical Benefits of Photovoltaics to Distribution Systems. *IEEE Trans. Smart Grid.* **2021**, *12*, 4938–4949. [[CrossRef](#)]
70. Carrasco, J.; García, S.; Rueda, M.M.; Das, S.; Herrera, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm Evol. Comput.* **2020**, *54*, 100665. [[CrossRef](#)]

71. Sambariya, D.K.; Gupta, T. Optimal design of PID controller for an AVR system using monarch butterfly optimization. In Proceedings of the 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), New York, NY, USA, 17–19 August 2017; pp. 1–6.
72. Cui, X.; Chen, Z.; Yin, F. Differential evolution and local search based monarch butterfly optimization algorithm with applications. *Int. J. Comput. Intell. Syst.* **2018**, *12*, 149–163. [[CrossRef](#)]
73. Latif, A.; Das, D.C.; Barik, A.K.; Ranjan, S. Maiden coordinated load frequency control strategy for ST-AWEC-GEC-BDDG-based independent three-area interconnected microgrid system with the combined effect of diverse energy storage and DC link using BOA-optimised PFOID controller. *IET Renew. PowerGener.* **2019**, *13*, 2634–2646. [[CrossRef](#)]
74. Ghanbari, M.; Arian, H. Forecasting Stock Market with Support Vector Regression and Butterfly Optimization Algorithm. *arXiv* **2019**, arXiv:1905.11462.
75. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, 3rd ed.; Pearson Education India: Delhi, India, 2009.