Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-021-06026-3

RESEARCH ARTICLE - SPECIAL ISSUE - FRONTIERS IN PARALLEL PROGRAMMING MODELS FOR FOG AND EDGE COMPUTING INFRASTRUCTURES

On Comparing the Performance of Swarm-Based Algorithms with Human-Based Algorithm for Nonlinear Systems

Vishal Srivastava[1] · Smriti Srivastava[2] · Gopal Chaudhary[3] · Xiomarah Guzmán-Guzmán[4] · Vicente García-Díaz[4]

Received: 24 March 2021 / Accepted: 28 July 2021

Abstract

This paper exploits various meta-heuristic optimization techniques to learn PID controller parameters for nonlinear systems. The nonlinear systems considered here are well known ball and beam, inverted pendulum, and robotic arm manipulator. The gain parameters of the controllers are optimized by using two categories of meta-heuristic optimization techniques—swarm-based grasshopper optimization algorithm and particle swarm optimization and human-based, i.e., teacher learning-based optimization. Mean square error has been used to measure the performance of various algorithms. Robustness of these algorithms is studied and compared using parameter perturbation and external disturbance. There are substantial improvements in the performance of these plants using the mentioned algorithms as shown in the simulation results. A detailed comparative analysis of these algorithms has also been done.

Keywords Inverted pendulum · Ball and beam · Robotic arm manipulator · Particle swarm optimization (PSO) · Grasshopper optimization algorithm (GOA), · Teacher learning-based optimization (TLBO)

1 Introduction

Performance analysis of nonlinear systems is a challenging task and plays a vital role in control theory [4, 41, 46]; therefore, design of a suitable controller is the need of the hour [32, 50]. A robust controller should be able to handle the sudden change, uncertainty, and other system nonlinearities [15]. Conventional PID controllers used to be the initial choice for control engineers in industries because of their simplicity [44]. These controllers failed to show an effective operation to complex, coupled nonlinear plants with uncertainties [31]. The analysis of nonlinear systems has been done with different inputs and conditions like time delay, dead zone, etc. [49, 16]. In the literature, many controllers and control techniques with various optimization algorithms have been suggested to tune the controller parameters. The optimized controller not only improves the system response but also adapts the changes quickly. Due to simplicity, PID controller has been used with optimization algorithms and gave the satisfactory results in analysis of nonlinear systems. Ball and beam is a highly nonlinear control problem; hence, conventional PID controller is unable to handle the nonlinearities and disturbances acting on it [7, 45]. Genetic algorithm (GA) with PID controller parameters has been

✉ Vicente García-Díaz
garciavicente@uniovi.es

Vishal Srivastava
vishal.srivastava0708@gmail.com

Smriti Srivastava
smriti@nsut.ac.in

Gopal Chaudhary
gopal.chaudhary88@gmail.com

Xiomarah Guzmán-Guzmán
UO273217@uniovi.es

1 Netaji Subhas Institute of Technology, Sector 3, Dwarka, New Delhi, India

2 Netaji Subhas University of Technology (Erstwhile NSIT), Sector 3, Dwarka, New Delhi, India

3 BharatiVidyapeeth's College of Engineering, Paschim Vihar, Delhi, India

4 Department of Computer Science, University of Oviedo, Oviedo, Spain

Published online: 24 August 2021

used to evaluate the performance of the system in terms of minimizing the error [12 8]. State observer and fractional controller has also been used for dynamical analysis of ball and beam system [25–10]. Ant colony optimization (ACO) with fuzzy logic controller (FLC) was used to improve the performance of ball and beam system [5, 11]. For another example, i.e., robotic arm manipulator, adaptive control and online tuning has been done to improve its performance [23, 9]. Fuzzy logic controller in combination with PD controller has been used for the analysis of wheeled robot system [19, 13]. A comparative study of PID and FLC has been carried out for inverted pendulum also [20]. The performance of the system was also analyzed with algorithm-based LQR controller [48].

1.1 Related Work

In the literature, benchmark problems of control engineering are analyzed with control schemes like linear quadratic regulators (LQR) [33], state-space control [24], sliding mode control (SMC) [36], neural networks (NN) [17, 6], and fuzzy logic control (FLC) [47, 43]. Many nature-inspired and evolutionary techniques have been discussed such as genetic algorithms (GA), ant colony optimization (ACO), and particle swarm optimization (PSO) [3], and they have shown improved performance of the system [9]. Generally speaking, these swarm-based algorithms have some limitations and take more computational effort [42]. Mohammed et al. [29] used genetic algorithm (GA) and moth swarm algorithm (MSA) for finding the optimal location and sizing of distributed generation. Kumar et al. [22] used ant colony optimization (ACO) and *K*-means clustering algorithm to find the shortest path from source to destination for Internet of things (IoT) models. Raj et al. [34] used three-degree-of-freedom (DOF) cabling robot with PID controller to find stable trajectories. Recently few algorithms have been developed, i.e., grasshopper optimization algorithm (GOA) [39] and teacher learning-based optimization algorithm (TLBO) [35] which reduces the computational effort during the optimization process and handles the uncertain behavior in less time [30] as compared to older algorithms. Sahu et.al. [38] used TLBO algorithm to optimize fuzzy PID controller parameters and compared it with various other algorithms for automatic generation control (AGC). The performance was analyzed in terms of time domain parameters. Khooban et.al. [21] used type-2 fuzzy PID controller to track the trajectory of nonholonomic wheeled mobile robots. TLBO algorithm was used to optimize the controller parameters and improve the trajectory response. Abualigah et al. [1], GOA has been widely used since its proposal because of its good exploration ability and less number of controlling parameters. The work related to GOA can be divided as improvements in GOA and its application in different areas. Mafarja et al. [26] used binary GOA (BGOA) in feature selection problem. Feature selection is a complex machine learning problem, and so a binary version of GOA has been used to obtain the optimal subset of informative features. Sigmoid model and V-shaped transfer function have been used to convert GOA into BGOA. BGOA is implemented on several test functions and reflected better performance as compared to other related techniques. Arora et al. [2] proposed chaotic GOA (GOA) to investigate the performance of system. A randomness was introduced in to the system by chaotic maps which tried to balance the search abilities and speed of convergence of the algorithm. The proposed CGOA was implemented on several benchmark functions showing the satisfactory results as compared to others. Lotfipour [25] used discrete TLBO (DTLBO) in reconfiguration disturbed generation problem. It is a complex combinational problem in which objective was to improve the voltage profile while minimizing power loss. The comparison of DTLBO was done with other existing methods and found that DTLBO maintained voltage profile quite well and significantly minimized the power loss. Shabanpour-Haghighi [40] used modified TLBO (MTLBO) to analyze multiobjective power flow problem. The fuel cost and total emission were taken as objective functions. The authors utilized self-adapting wavelet mutations which enhanced the convergence speed of the algorithm. It was verified with other methods and showed better results. Sahu [37] used TLBO algorithm with two degrees of freedom of PID controller for automatic generation control (AGC) and showed better results in terms of sensitivity and load variations.

1.2 Research Gap and Motivation

Several meta-heuristic algorithms have been proposed in the literature to analyze nonlinear dynamical systems. Swarm-based algorithms require algorithm-specific parameters along with common control parameters. For example, PSO uses cognitive and social parameters and inertia weight. Improper tuning of these algorithms may lead to local optimal solutions due to their poor exploration ability or may need lots of computational efforts. Recently, another swarm-based algorithm, GOA, was proposed inspired by grasshopper behavior. It has shown better results as far as balance of exploration and exploitation is concerned. TLBO is a human-based algorithm, which does not have algorithm-specific control parameters and therefore takes less computational efforts that makes it a better choice for nonlinear system analysis.

In this paper, these optimization algorithms to tune the parameters of PID controllers have been implemented on benchmark problems such as inverted pendulum, ball and beam, and robotic arm manipulator. By optimizing the controller parameters and minimizing the mean squared error, system performances have been improved significantly. The paper is organized as follows:

In Sect. 2, mathematical modeling and equations of nonlinear systems are described. In Sect. 3, the optimization algorithm-based control scheme is shown. The effect of these algorithms and their comparative analysis is carried out in Sect. 4. A brief conclusion is discussed in Sect. 5.

2 Mathematical Modeling of Nonlinear Systems

In this section, the three benchmark problems are discussed in brief.

2.1 Inverted Pendulum

Inverted pendulum system is one of the classical examples in control engineering [20, 48]. The problem here is to design a controller that can balance the pendulum position as the cart moves [45, 14] and also adapts the system uncertainties effectively and efficiently. Figure 1 shows the schematic of the inverted pendulum. When force F is applied on the cart, pendulum angle θ changes. [14] The controller tries to stabilize the pendulum position as quickly as possible. Table 1 shows the specific parameters used to analyze the system.

The system equations are given as-

$$F = (M + m)\frac{d^2x}{dt^2} + b\frac{dx}{dt} + ml\frac{d^2\theta}{dt^2}\cos\theta - ml\left(\frac{d\theta}{dt}\right)^2\sin\theta \tag{1}$$

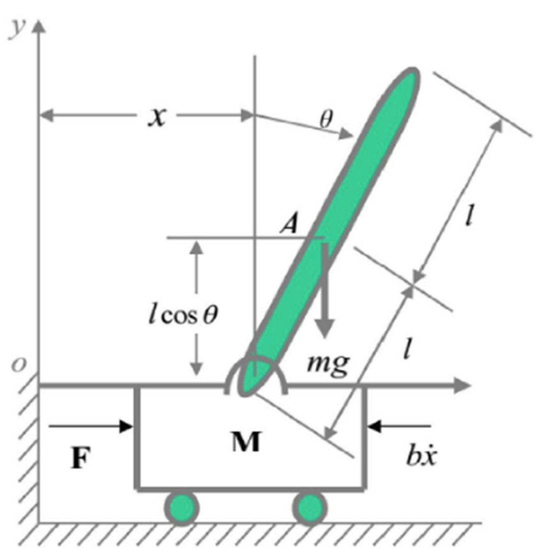


Fig. 1 Inverted pendulum

Table 1 Parameters of inverted pendulum

Parameters name	Values
Cart mass	$M = 0.5$ kg
Pendulum mass	$m = 0.2$ kg
Cart friction	$b = 0.1$ N/m/sec
Pendulum length	$l = 0.3$ m
Pendulum inertia	$I = 0.006$ kg*m^2
Gravitational constant	$g = 9.8$ m/s^2

$$\left(I + ml^2\right)\frac{d^2\theta}{dt^2} + ml\sin\theta = -ml\frac{d^2x}{dt^2}\cos\theta \tag{2}$$

2.2 Ball and Beam

Ball and beam is another benchmark problem in control engineering [11, 28]. In Fig. 2, ball is placed on a beam and due to gravity, it rolls along the beam [17, 18]. As the servo gear angle θ changes, the beam angle α also changes [10, 13]. The objective is to control the ball position as servo angle changes. Table 2 shows the specific parameters used for the analysis of this system.

The equations governing the system behavior are as follows -

$$\left(m + \frac{j_b}{R^2}\right)r\ddot{r} + \frac{j_b}{R\alpha} - m\dot{r}\alpha^2 + mg\sin\alpha = 0 \tag{3}$$

For the small value of α, Eq. (3) can be written as –

$$\left(m + \frac{j_b}{R^2}\right)r = \dot{m}g\alpha \tag{4}$$

Beam angle and gear angle are related by the following equation -

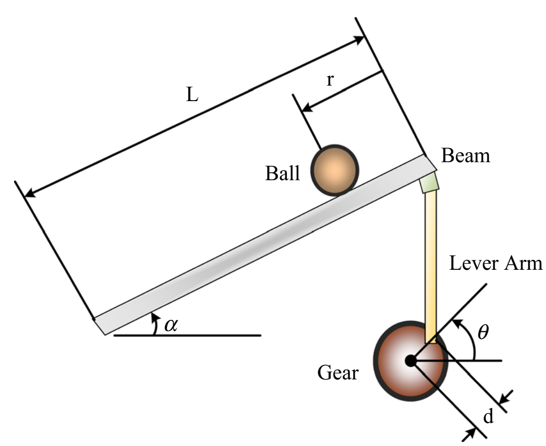


Fig. 2 Ball and beam

Table 2 Parameters of ball and beam

Parameters name	Values
Ball mass	$m = 0.11$ Kg
Ball radius	$R = 0.015$ m
Beam length	$L = 0.015$ m
Lever arm	$d = 0.03$ m
Gravitational constant	$g = 9.8$ m/s^2
Ball inertia	$j_b = 9.99$ e $^{-6}$ kg*m^2
Beam length	$L = 1.8$ m
θ = Servo gear angle	
α = Beam angle	
r = Ball position coordinate	

$$\alpha = \left(\frac{d}{L}\right)\theta \tag{5}$$

2.3 Robotic Arm Manipulator

Robotic arm manipulator is a complex nonlinear system [23]. To control the arm position, various types of controllers like PID, adaptive [19], and robust controllers have been used [7] in the literature. Figure 3 shows robotic arm manipulator, and Table 3 shows the specific parameters used for the analysis of the same. Here in Fig. 3 as the angle θ_1 of link 1 changes, the angle θ_2 of the link 2 also changes simultaneously. The main objective is to control the arm position of link 2 by controlling the angle θ_2.

The displacement of x and y in terms of θ_1 and θ_2 is given as follows-

$$x_1 = l_1 \mathrm{Sin}\,\theta_1 \tag{6}$$

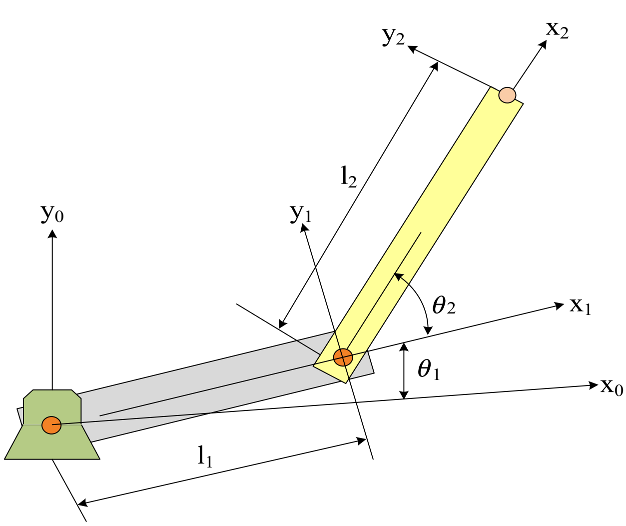


Fig. 3 Robotic arm manipulator

Table 3 Parameters of robotic arm manipulator

Parameters name	Values
Length of link 1	$l_1 = 1$ m
Length of link 2	$l_2 = 1$ m
Mass of link 1	$m_1 = 1$ kg
Mass of link 2	$m_2 = 1$ kg
Distance to half of the link	$l_{c_1} = l_{c_2} = 0.5$ m
Moment of inertia	$j_1 = j_2 = 9.99e - 6$ $kg * m2$
Gravitational acceleration	$g = 9.8$ m/s^2
θ_1, θ_2 are angle of links 1 and 2	
v_1, v_2 are velocity of links 1 and 2	
ω_1, ω_2 are angular position of links 1 and 2	

$$y_2 = l_2 \mathrm{Sin}\,\theta_2 \tag{7}$$

$$x_2 = l_1 \mathrm{Sin}\theta_1 + l_2 \mathrm{Sin}\left(\theta_1 + \theta_2\right) \tag{8}$$

$$y_2 = l_1 \cos\theta_1 + l_2 \mathrm{Cos}\left(\theta_1 + \theta_2\right) \tag{9}$$

The kinetic energy is given as-

$$\mathrm{K.E} = \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 + \frac{1}{2}j_1\omega_1^2 + \frac{1}{2}j_2\omega_2^2 \tag{10}$$

The potential energy is given as-

$$\mathrm{P.E} = m_1 g I_1 \sin\theta_1 + m_2 g (I_1 \sin\theta_1 + \left(I_2 \sin\left(\theta_1 + \theta_2\right)\right) \tag{11}$$

3 Control Scheme

PID controller is implemented with various optimization algorithms that improve the performance as compared to conventional algorithms [27] which not only handles the parametric changes effectively but also tries to settle the sudden changes in the system fast during the operation. These algorithms not only give the optimized value of gains but also minimize the objective function of the system. The control scheme is shown in Fig. 4. In this figure, PID controller along with mentioned optimization algorithms is shown. These algorithms not only optimize the controller parameters K_P, K_1, and K_d but also minimize the mean square error (MSE) Fig. 5, i.e., objective function. The input to the controller is the error, and the output of it is feeding the nonlinear plant. The objective is to bring the actual output to be equal to the set point using PID controller.

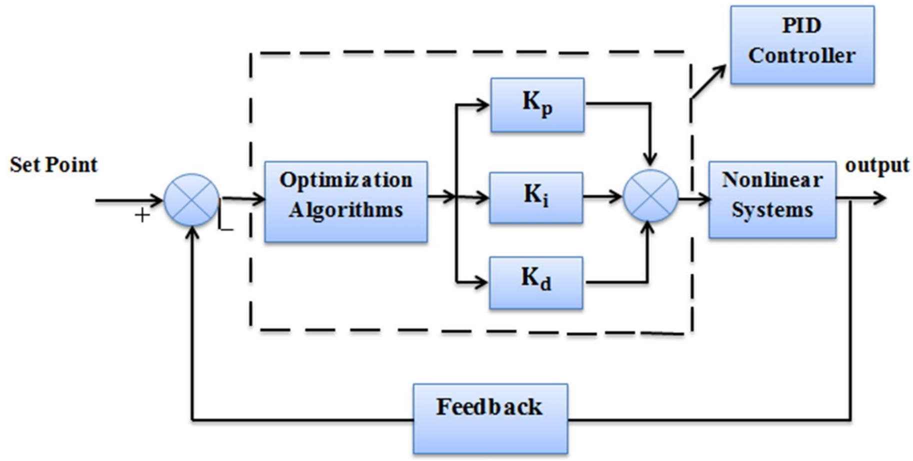


Fig. 4 The control scheme of optimization algorithm-based PID controller

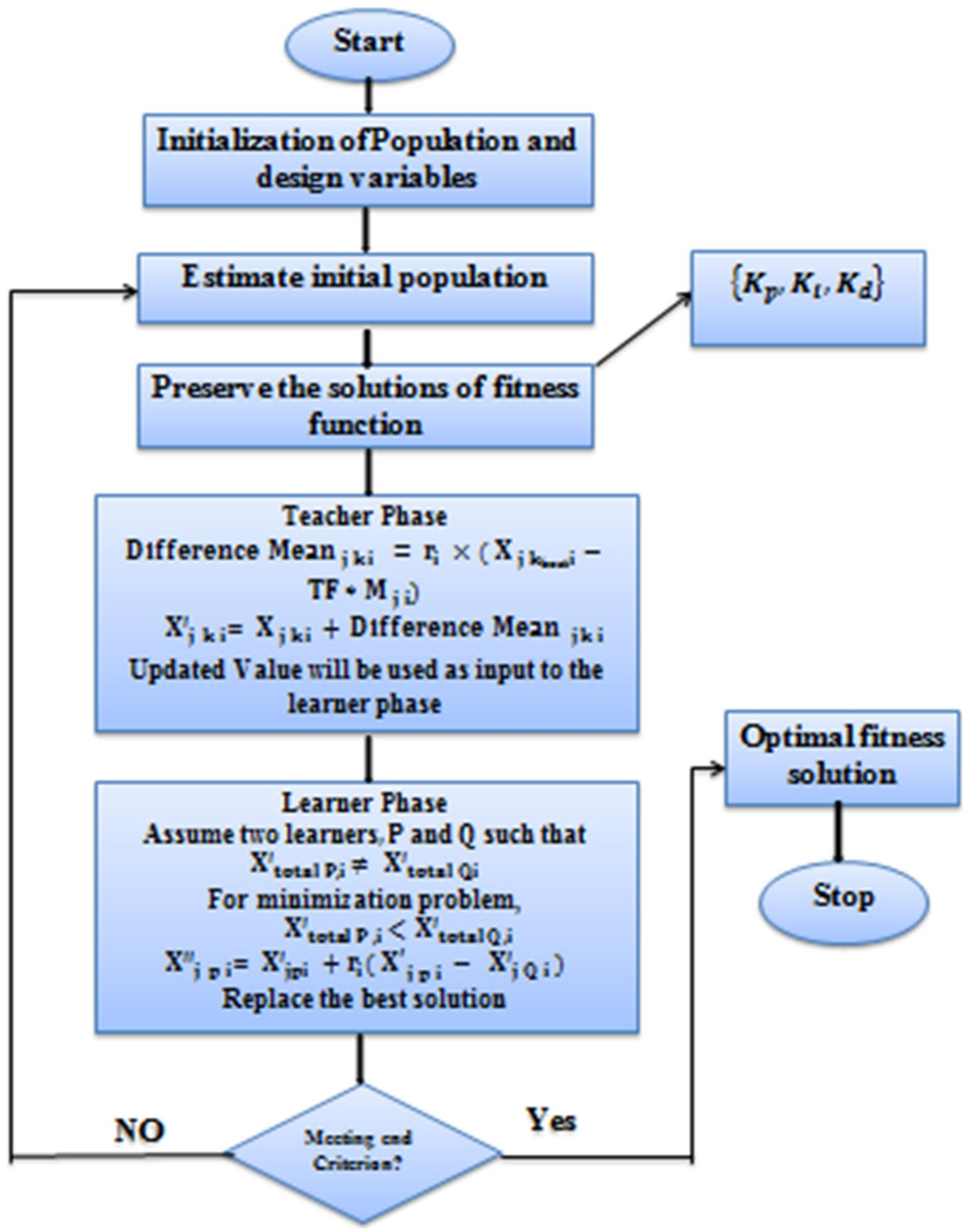


Fig. 5 Flowchart of TLBO algorithm

3.1 Model of PID Controller

PID controller is a conventional controller that is used in industries to control various processes [16]. For nonlinear systems, because of its limitations, PID controller needs lots of efforts to handle the nonlinearities [12] and uncertainties of systems. By proper tuning of PID controller parameters, system performance can be improved to a decent extent [8].

The equation of PID controller is -

$$C(t) = K_P e(t) + K_i \int e(t) + K_d \frac{d}{dt} e(t) \tag{12}$$

where $e(t)$ is the error, K_P is proportional gain, K_i is integral gain, and K_d is derivative gain.

3.2 Particle Swarm Optimization Algorithm (PSO)

PSO is a swarm intelligence algorithm based on the flocking of birds. In PSO, particles move in a search space and follow the optimal path [3]. Fitness values and velocities are obtained by the fitness function [49]. Update equations of velocity and position are given as follows-

$$\begin{aligned} V_i^{k+1} = w * V_i^k + \emptyset_1 \times \mathrm{rand}_1 \times \left(\mathrm{pbest}_i^k - X_i^k\right) \\ \times \mathrm{pbest}_i^k + \emptyset_2 \times \mathrm{rand}_2 \left(\mathrm{gbest} - X_i^k\right) \end{aligned} \tag{13}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{14}$$

where $w$is inertia weight,$\emptyset_1$,$\emptyset_2$ are the learning rates, and $rand_1$, $rand_2$ are random numbers between [0 1].w is the weighting function,V_i^k and X_i^k are velocity and position of current particles, pbest_i^k is the pbest of particle i, and gbest is the best solution. The initial values of algorithm specific parameters are: number of swarms = 1000, $\emptyset_1 = \emptyset_2 = 0.01$ and $w = 2$. Initially, the PID controller parameters, i.e., K_P, K_i, and K_d, are assigned with some random values to calculate the objective function or fitness value. Now with each iteration, the error value changes which further changes the fitness value. Due to convergence property of algorithm, fitness value will be minimum and at the same time the optimal value of the parameters is obtained.

3.3 Grasshopper Optimization Algorithm (GOA)

GOA is a comparatively new optimization algorithm. Swarm in its adulthood has a long-range and random movement in search space [39]. Grasshopper has an effective chief feature for food seeking. Swarm segments the search process in two tendencies, i.e., exploration and exploitation. In exploration, the search agents are fortified to move in whole search space and they move locally during exploitation to find the best solution [3]. Grasshopper naturally performs these two functions in seeking the targets. The initial values of algorithm specific parameters are: number of search agents = 1000, $b = 0.5l = 1.5$. In GOA basically the effect of attraction and repulsion forces is used to calculate the fitness function. Repulsion forces allow to use grasshopper exploration ability while attraction force allows to use exploitation ability. After assigning the initial values, the objective or fitness function value will be calculated for individual grasshopper. Now based on fitness value, not only the worst solution is removed from the population but also position is updated using Eq. (21). Unlike PSO, GOA has only one position vector to improve the existing solutions, which accelerate the convergence mobility of GOA.

Swarming behavior of grasshopper is defined by following mathematical equations-

$$P_x = r_1 A_x + r_2 B_x + r_3 C_x \tag{15}$$

where P_x is position, A_x is social interaction with other grasshoppers, B_x is gravity force, C_x is wind effect of x^{th} grasshopper, and r_1, r_2, and r_3 are random numbers between [0, 1].

$$A_x = \sum_{\substack{y = 1 \\ y \neq x}}^{n} h(D_{xy}) \times \widehat{D_{xy}} \tag{16}$$

whereD_{xy} is the distance between x^{th} and y^{th} grasshopper computed as, $D_{xy} = \left|i_y - i_x\right|$,

$\widehat{D_{xy}} = \frac{i_y - i_x}{D_{xy}}$, is unit vector from x^{th} and y^{th} grasshopper.

The social forces h are computed as follows –

$$h(r) = be^{\frac{-r}{l}} - e^{-r} \tag{17}$$

where b and l are attraction intensity and length and r is the coordinate representing direction of grasshopper, respectively.

Components B and C are calculated as follows-

$$B_x = -g \times \widehat{e_g} \tag{18}$$

$$C_x = -\mu \times \widehat{e_\omega} \tag{19}$$

where g is gravitational constant, $\widehat{e_g}$ and $\widehat{e_\omega}$ are unit vectors, and μ is drift constant.

Now Eq. (15) can be written as-

$$P_x = \sum_{\substack{y = 1 \\ y \neq x}}^{n} h(\left|i_y - i_x\right|) \times \frac{i_y - i_x}{D_{xy}} - g\widehat{e_g} + \mu\widehat{e_\omega} \tag{20}$$

where n is number of grasshoppers. Since the target location is undiscovered, the grasshopper that owns the best fitness is estimated as the near one to the target. To attain the objective, grasshoppers continue their movement in the direction of the target in the social interaction network [39]. To balance between global search and local search, the location of grasshoppers is updated and hence the comfort zone reduces adaptively. After the exploitation and exploration process, grasshoppers ultimately achieved a convergence and find the best solution.

The updated equation of position is given as follows-

$$P_x^d(t+1) = k\left(\sum_{\substack{y=1 \\ y \neq x}}^{G} k(\frac{ub_d - lb_d}{2}\right) \times s\left(P_y^d(t) - P_x^d(t)\right) \times \left(\frac{P_y^d(t) - P_x^d(t)}{D_{xy}}\right) + \hat{T}_d \tag{21}$$

where ub_d and lb_d are upper and lower bound, respectively, and $\hat{T}_d$ is the location of the target in d dimension, k is decreasing coefficient, and $P_x^d(t)$, $P_y^d(t)$ are position of xth and yth grasshopper, respectively.

3.4 Teaching–Learning-Based Optimization (TLBO) Algorithm

TLBO is are recently suggested human-based algorithm inspired from classroom teaching process of teacher and students. This algorithm is efficient and effective due to less computational efforts and few parameters. It has two phases: teacher phase and student phase also known as the learner phase [35]. Teacher phase in this algorithm shows the exploration ability in which learners can enhance their knowledge with teacher experience while the interactions between learners shows the exploitation ability of the algorithm. The teacher is considered as the best solution for the problem. By interacting with learners, teacher tries to increase the mean of the class. In second phase, learners enhance their knowledge by interacting with other learners. Through this interaction process, learner parameters are updated that lead to optimal solution. The initial values of algorithm-specific parameters are: population size = 1000, TW = 1.5, c_1 = 1. Initially in teacher phase based on skill and knowledge, teacher interacts with students and increase their knowledge. Teacher has the responsibility to increase the average result of the class. By using Eq. (23), the fitness value will be calculated. Now with update fitness value, learners interact with each other and increase their knowledge by using Eq. (26) or (28). With each iteration, controller parameters change with fitness function value.

The equation of teacher phase is given by –

Teacher tries to improve difference means, M_{di} of the class and is calculated as follows-

$$\text{Difference Mean}_{dfi} = c_1 \times \left(Y_{df_{\text{best}}i} - \text{TW} * M_{di}\right) \tag{22}$$

where c_1 is a random number between 0 and 1. $Y_{df_{\text{best}}i}$ is the result of the best learner in subject d, TW is the teaching weight whose value is generally taken either 1 or 2, M_{di} is mean result of learners in subject d and f is number of iterations.

Using Eq. (22) solution is updated by the following equation-

$$Y'_{dfi} = Y_{dfi} + \text{Difference Mean}_{dfi} \tag{23}$$

where Y_{dfi} is result of ith learner in subject d on fth iterations.

For learner phase, parameters are updated by the interaction process. If only two learners, U and V, are considered with the assumption that their knowledge is not equal, then –

$$Y'_{\text{total } U,i} \neq Y'_{\text{total } V,i} \tag{24}$$

Assuming that knowledge transfer is taking place from U to V, then

$$\text{If } Y'_{\text{total } U,i} < Y'_{\text{total } V,i} \tag{25}$$

$$Y''_{dUi} = Y'_{dUi} + c_1\left(Y'_{dUi} - Y'_{dVi}\right) \tag{26}$$

Assuming that knowledge transfer is taking place from V to U, then

$$If\ Y'_{\text{total V , i}} < Y'_{\text{total} U,i} \tag{27}$$

$$Y''_{dUi} = Y'_{dUi} + c_1\left(Y'_{dVi} - Y'_{dUi}\right) \tag{28}$$

where Y''_{dUi} is the updated value of learner U in subject d.

4 Results and Discussion

All the three algorithms, i.e., PSO, GOA and TLBO, were implemented to tune the parameters of PID controller so as to improve the performance by minimizing mean square error of the systems, as discussed above.

4.1 Inverted Pendulum

To control the position of pendulum, sequentially all algorithms were implemented on PID controller. The tuned PID controller quickly stabilized the pendulum position.

Figure 6a–d shows that GOA-based PID controller depicts better performance. Table 4 shows that settling time is 0.08 s, rise time is 1.55 s, peak time is 0.011 s, peak value is 1.5 and overshoot is 50%, which is less as compared to PSO- and TLBO-tuned controller. Figure 6b shows that when a sudden step input is applied to the system at $t = 2$ s, GOA-based PID controller responds fast reflecting better performance. The simulation results in Fig. 6c show that MSE for GOA-tuned controller converges faster as compared to others. Table 5 clearly shows, mean square error value significantly reduced to a value of "0.15547" in 03 iterations.

4.1.1 Ascertaining Robustness of Inverted Pendulum

The main objective of this section is to show the robustness of inverted pendulum. The mass of pendulum is perturbed with a value of 0.05 kg. The new mass of pendulum becomes -

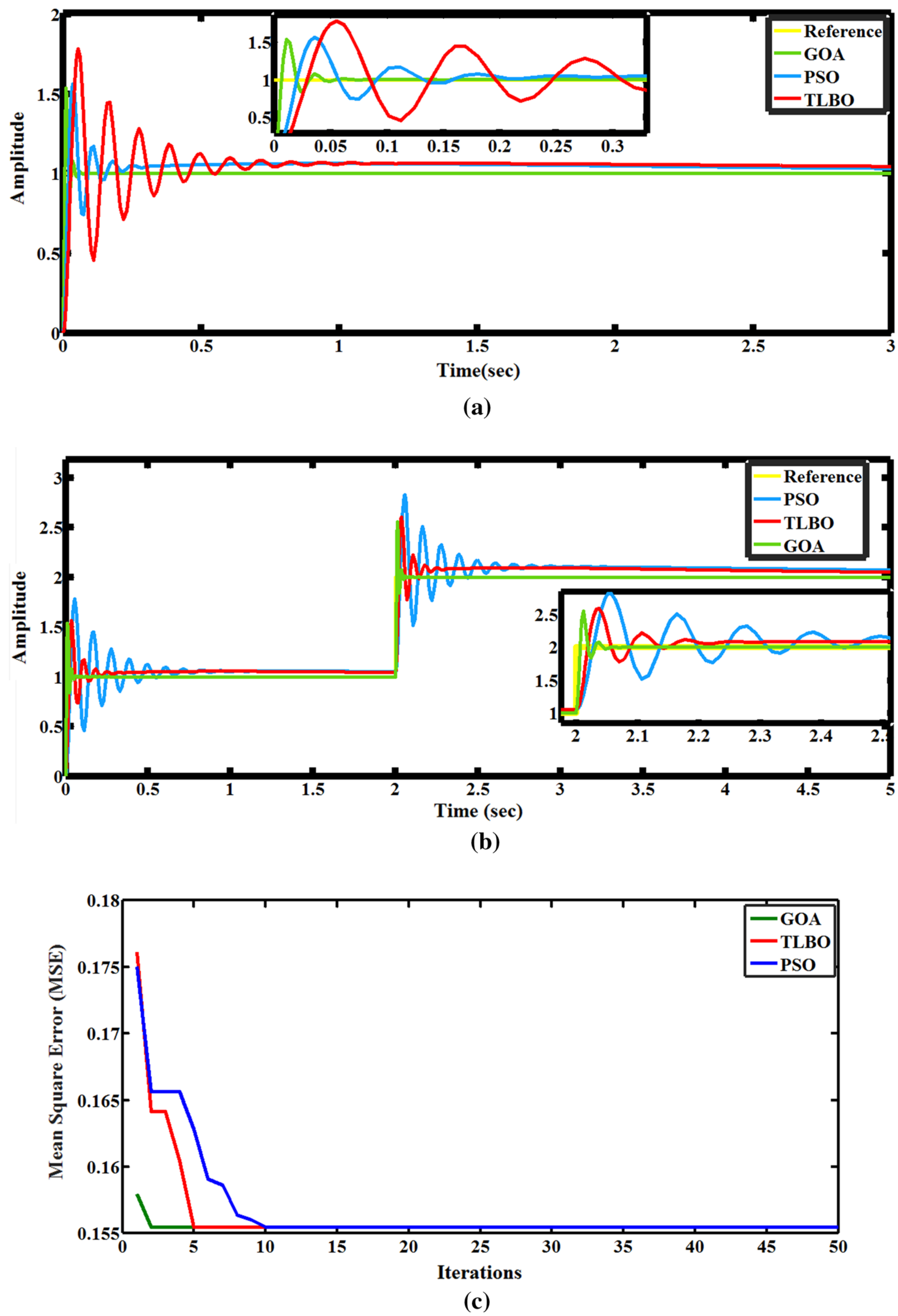


Fig. 6 **a** Pendulum position with different algorithms, **b** pendulum position subjected to a sudden disturbance, and **c** mean squared error using different algorithms

Table 4 Comparison of algorithms for different systems

Algorithms	Rise time (s)	Peak time (s)	Peak value	Settling time (s)	Overshoot (%)
Inverted pendulum					
PSO	1.44	0.032	1.6	0.45	60
TLBO	1.62	0.05	1.8	0.9	80
GOA	1.55	0.011	1.5	0.08	50
Ball and beam					
PSO	1.233	0.31	1.37	1.28	37
TLBO	1.062	0.30	1.18	0.9	18
GOA	1.665	1.75	1.85	-	85
Robotic arm manipulator					
PSO	1.143	0.135	1.27	0.7556	27
TLBO	0.963	0.9	1.07	0.78	7
GOA	1.35	0.0346	1.505	0.2280	50

Table 5 Comparison of mean square error (MSE)

Algorithms	Mean square error (MSE)	Iterations
Inverted pendulum		
PSO	0.15550	10
TLBO	0.11574	5
GOA	0.15547	3
Ball and beam		
PSO	0.00604	42
TLBO	0.00650	35
GOA	0.00604	43
Robotic arm manipulator		
PSO	0.15640	11
TLBO	0.15628	10
GOA	0.15545	5

$$m_1 = m + \Delta m$$

where $m = 0.2$ kg (Pendulum mass),$\Delta m = 0.05$ kg (perturbed value in pendulum mass), and$m_1 =$ new value of pendulum mass.

Figure 7 shows the step response of the plant. Table 6 shows the transient performance, i.e., rise time, settling time, peak time, overshoot, and peak values.

4.2 Ball and Beam

Next, PID controller with three algorithms was simulated on ball and beam. The ball position was stabilized by a tuned PID controller. Figure 8a shows that TLBO-based PID controller shows better performance. Table 4 clearly shows that settling time is 0.9 s, rise time is 1.062 s, peak time is 0.30 s, peak value is 1.18 and overshoot is 18%, which are less as compared to that with PSO and GOA. When a sudden step input is applied at $t = 2$ s, TLBO-tuned controller adapts the changes fast and shows better performance as depicted in Fig. 8c. Table 5 shows, mean square error value significantly minimizes to a value of "0.006054" in 35 iterations.

4.2.1 Ascertaining Robustness of Ball and beam

Robustness of ball and beam is analyzed using the parameter perturbation method. The value of the ball mass is perturbed with a value of 0.02 kg. The new mass of ball becomes -

$$m_1 = m + \Delta m$$

where $m = 0.11$ kg (mass of ball),$\Delta m = 0.02$ kg (perturbed value in ball mass), and $m_1 =$ new value of ball mass.

Figure 9 shows the response of the system with perturbed value of mass. Table 6 shows the system performance in terms of rise time, settling time, peak time, overshoot, and peak values.

4.3 Robotic Arm Manipulator

The third example, i.e., robotic arm manipulator, is also simulated with all the mentioned algorithms. By proper tuning of the controller, robotic arm position was stabilized. Figure 10a shows, GOA-based PID controller gives better performance. Table 4 shows that settling time is 0.2280 s, rise time is 1.35 s, peak time is 0.0346 s, peak value is 1.505 and overshoot is 50%, which are less as compared to PSO- and TLBO-tuned controller. When a sudden step input is applied at $t = 2$ s, GOA-based PID controller adapts the changes fast and shows better performance as depicted in Fig. 10b. Fig. 10c shows that GOA gives better results, in terms of minimizing the objective function, i.e., MSE. Table 5 shows, mean square error value significantly minimizes with a value of "0.16558" in five iterations.

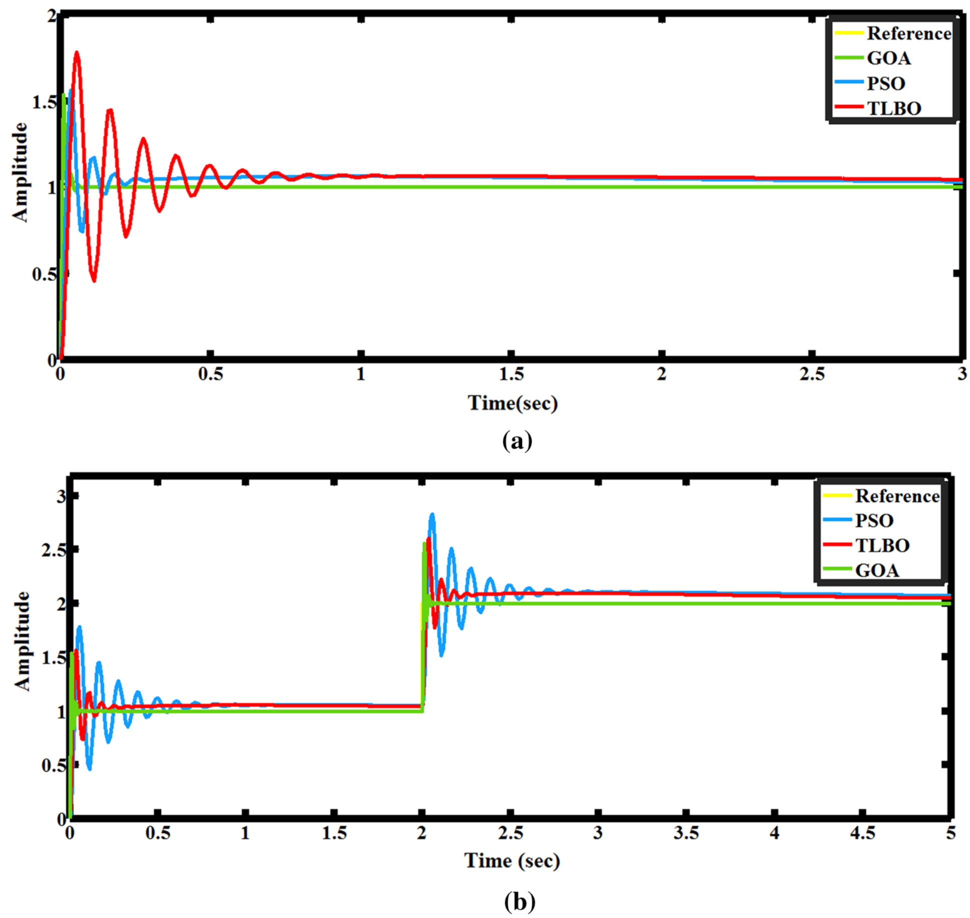


Fig. 7 **a** Response of pendulum position with mass perturbation and **b** sudden disturbance response of pendulum position using perturbed value of mass

Table 6 Comparison of systems with parameter perturbation

Algorithms	Rise time (s)	Peak time (s)	Peak value	Settling time (s)	Overshoot (%)
Inverted pendulum					
PSO	1.411	0.0342	1.5685	0.4824	56
TLBO	1.6074	0.0529	1.7877	1.124	78
GOA	1.3833	0.01015	1.5374	0.112	53
Ball and beam					
PSO	1.2351	0.30	1.3724	1.43	37
TLBO	1.0602	0.302	1.178	1.12	17
GOA	1.683	1.723	1.87	-	87
Robotic arm manipulator					
PSO	1.1403	0.133	1.267	0.85	26
TLBO	0.967	0.0958	1.0751	0.82	7
GOA	1.3527	0.0347	1.503	0.28	50

4.3.1 Ascertaining Robustness of Robotic Arm Manipulator

The objective of this section is to show the robustness of the robotic arm manipulator. In order to test the robustness, the value of the length of robot link 1 is perturbed with a value of 0.02 m, so the new length of robotic link 1 is-

$$l = l_1 + \Delta l$$

where $l_1 = 1$ m (length of robot link 1), $\Delta l = 0.02$ m (perturbed value in robot link 1 length).

l = new value of robot link 1.

Figure 11 shows the step input response with perturbed value of the length of robot link. Table 6 shows the

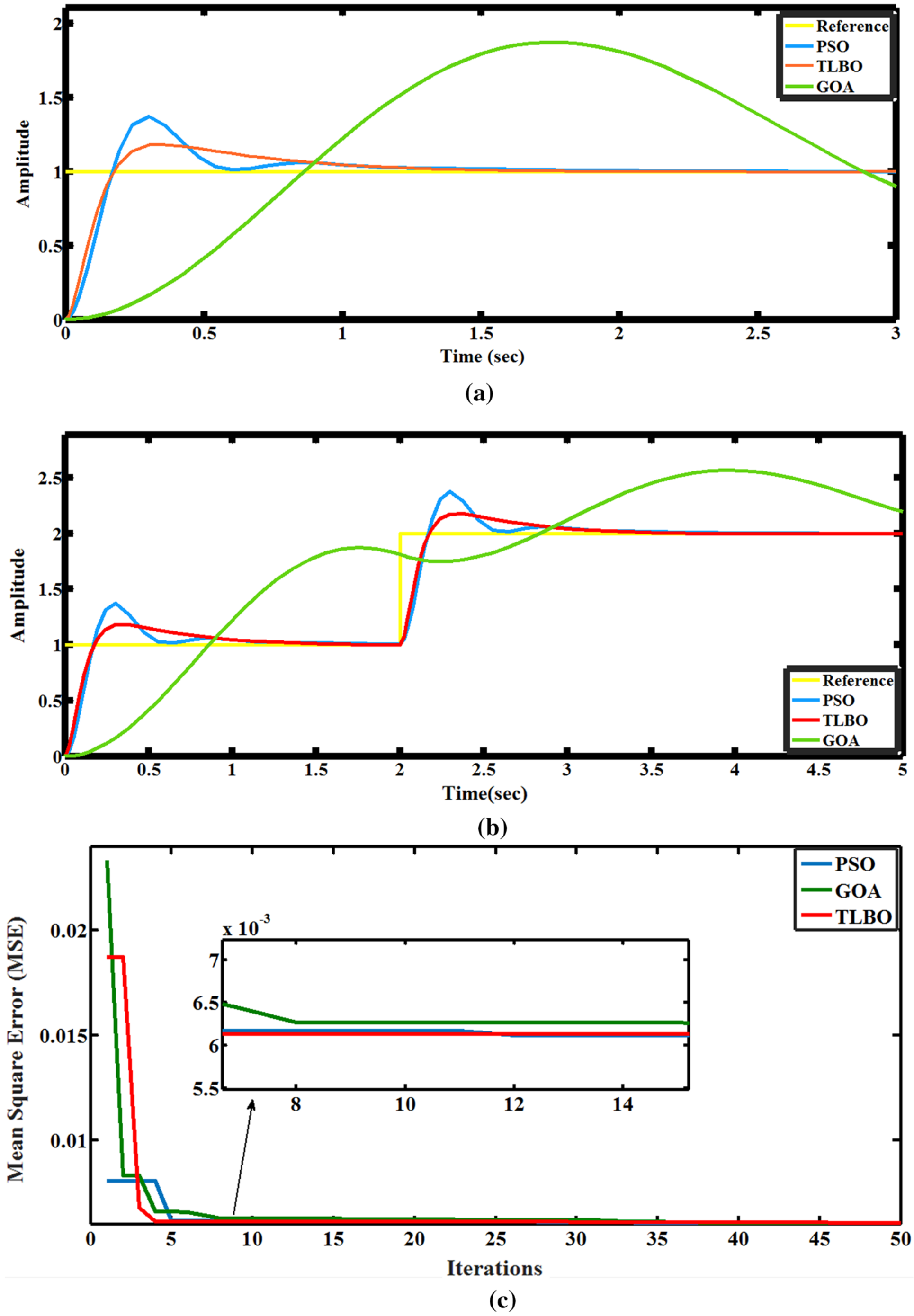


Fig. 8 **a** Ball position with different algorithms **b** Ball position subjected to a sudden disturbance **c** Mean square error of Ball and beam with different algorithms

transient performance, i.e., rise time, settling time, peak time, overshoot, and peak values.

4.4 Comparative Analysis

A comparative study of the three systems is done here. The performance is analyzed with parameters like rise time, settling time, peak value, and mean square error (MSE). Table 4 shows a comparative chart.

For each system, mean square error (MSE) calculated with different algorithms is shown in Table 5.

To evaluate the robustness of the tuning methods, parameter values as discussed in the problems are perturbed. Table 6 shows the comparison in terms of rise time, settling time, peak time, peak value, and overshoot.

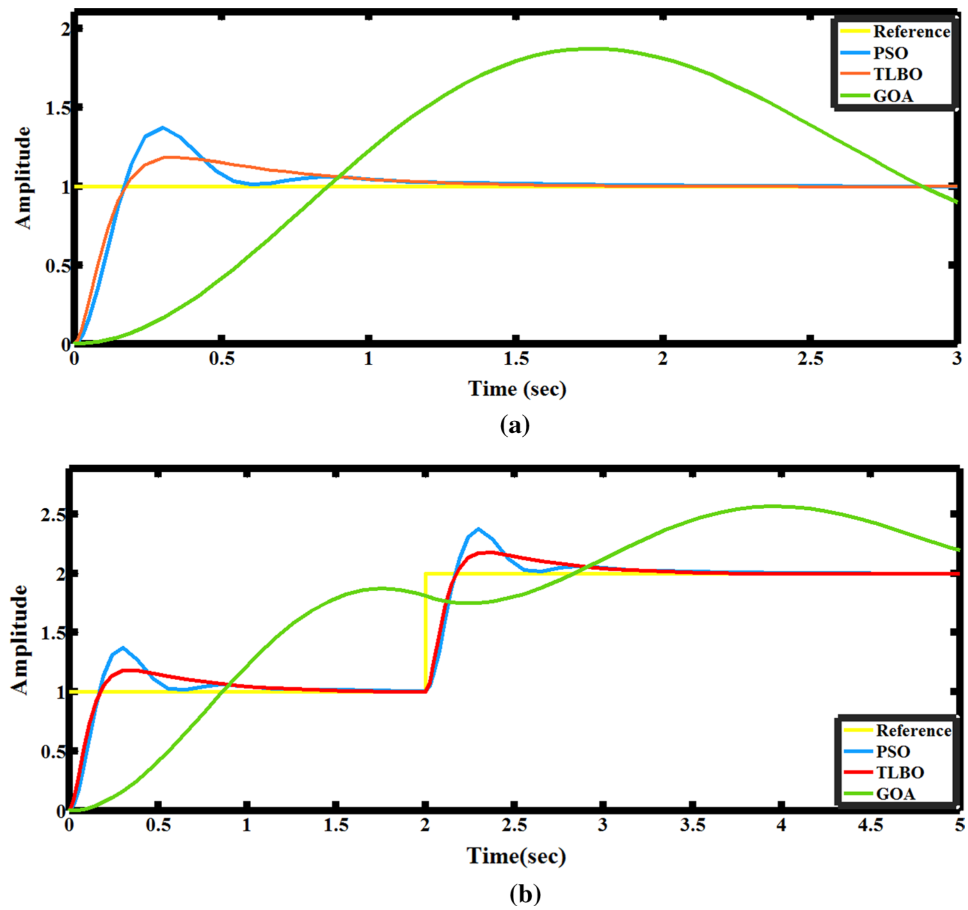


Fig. 9 **a** Response of ball position using perturbed value of mass with different algorithms and **b** sudden disturbance response of ball position using perturbed value of mass with different algorithms

The computational efforts involved in getting the output in one iteration are shown in Table 7. TLBO involves less number of arithmetic operations followed by GOA and PSO, respectively.

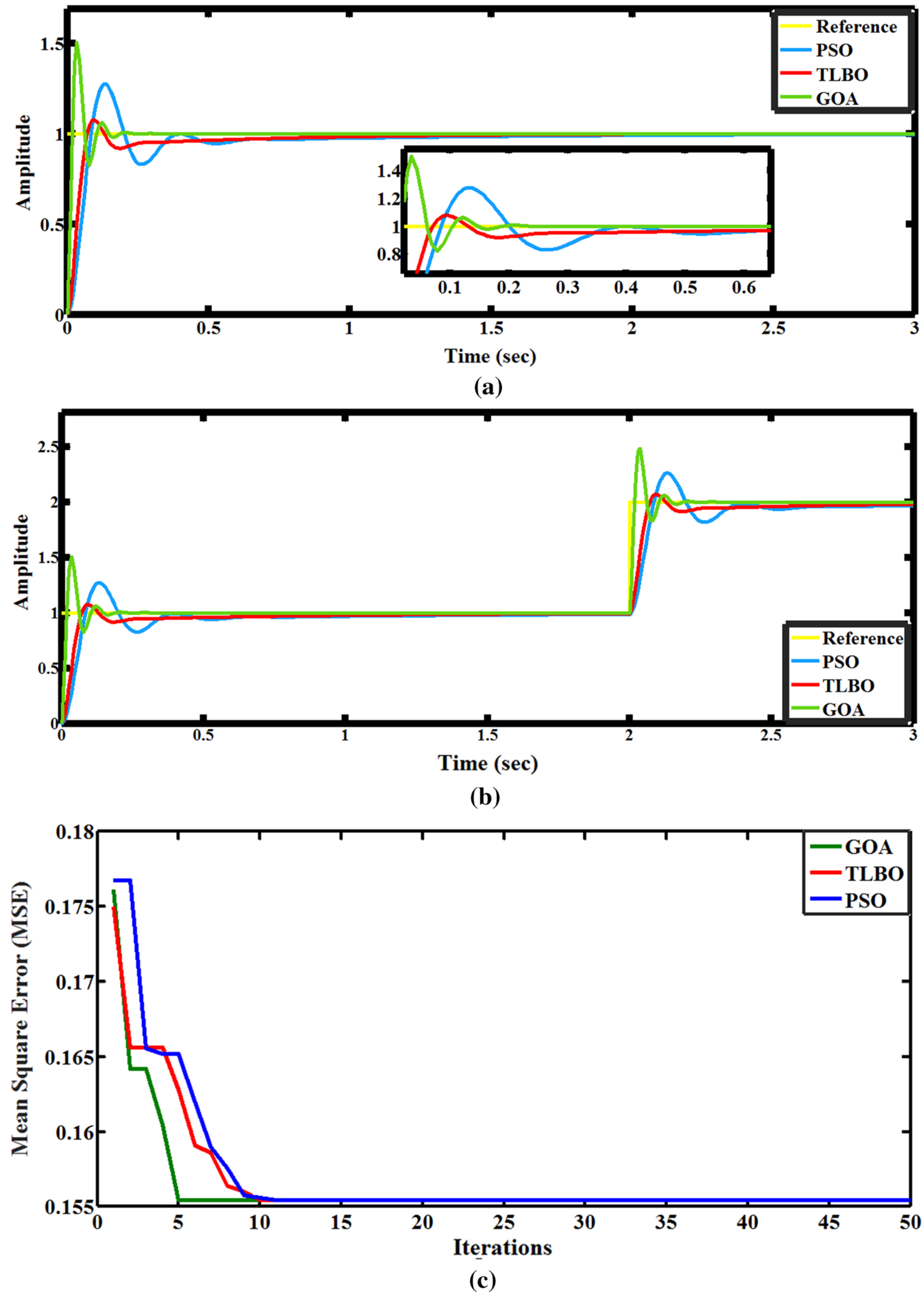
5 Conclusion

In this paper, optimization methods such as PSO-, TLBO-, and GOA-tuned PID controller were implemented on inverted pendulum, ball and beam, and robotic arm manipulator. GOA-based PID controller showed better performance as compared to other techniques. The controller stabilized the pendulum position quickly and adapted the sudden change and uncertainty in the system. Mean square error (MSE) value was reduced quite effectively. TLBO-based PID controller depicted better performance to stabilize the ball position as compared to other methods and also handled the sudden change in the system quite well thus showed an improved system performance. The mean square error (MSE) was minimized significantly which further enhanced the system performance. For robotic arm manipulator, GOA-based PID controller showed better results as compared to other techniques. Mean square error (MSE) was also minimized rapidly with very low value. To test the robustness of the system, parameters of the systems were perturbed from their original values and the results showed that the controller was able to adapt these changes quickly with mentioned algorithms.

In brief, the following points are worth mentioning -

1. From the transient performance of the systems with the mentioned algorithms, it is seen that GOA gives lesser time response parameters as compared to the other two (refer Table 4).
2. Mean square error was calculated with different algorithms for the three systems. For inverted pendulum, GOA is faster as compared to other two. It needed only three iterations to converge with MSE as 0.15547. For robotic arm manipulator, again GOA showed least number of iterations to reduce MSE to minimum. For ball and beam; however, TLBO was better. (Refer Table 5).
3. As seen from Table 6, TLBO is making the two systems (ball and beam and robotic arm manipulator) more

Fig. 10 **a** Robotic arm position with different algorithms, **b** robotic arm position subjected to a sudden disturbance with different algorithms, and **c** mean square error of robotic arm manipulator with different algorithms



robust while GOA depicts better performance for the inverted pendulum.

4. As far as computational time for a single iteration is concerned, TLBO algorithm takes least time for calculating mean square error (MSE) for all the three systems (refer Table 7).
5. To the best of our knowledge, such analysis of TLBO and GOA algorithms for optimization of PID parameters has not been performed on these benchmark problems.

6 Future Scope

Some future directions related to the work are discussed below:

1. Hybrid algorithms are very powerful approach for nonlinear systems analysis. Human-based algorithms and its variants can be used with recent swarm-based and physics-based algorithms and its variants. In this way

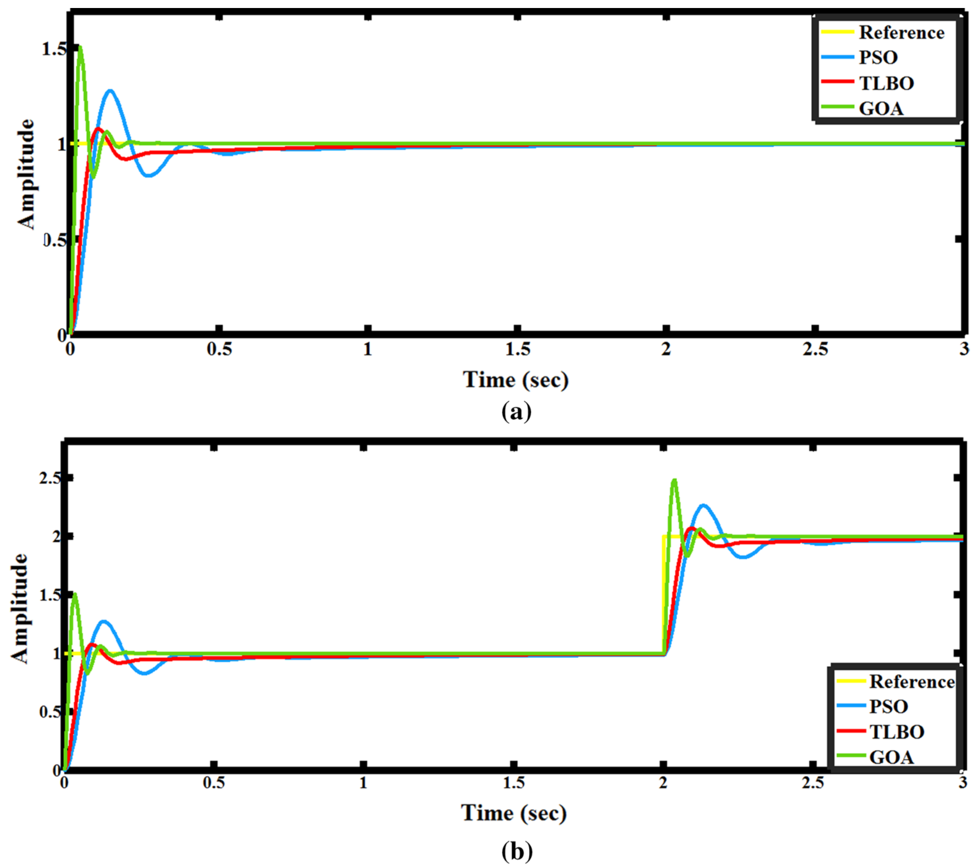


Fig. 11 **a** Response of arm position using perturbed value of the length of robot link with different algorithms and **b** sudden disturbance response of arm position with using perturbed value of the length of robot link with different algorithms

Table 7 Algorithms computing time

Plants	Algorithms	Computational time in Sec required for a single iteration
Inverted pendulum	PSO	17.857668
	TLBO	12.845318
	GOA	14.287136
Ball and beam	PSO	11.078482
	TLBO	7.37569
	GOA	8.250482
Robotic arm manipulator	PSO	20.43442
	TLBO	11.84572
	GOA	13.56488

better utilization of exploration and exploitation phases can be obtained.

2. Different soft computing techniques like fuzzy, artificial neural network (ANN), and adaptive neuro-fuzzy inference systems (ANFIS) can also be used with hybrid algorithms.
3. Lyapunov stability-based learning algorithm can also be used along with hybrid algorithms to study the behavior of nonlinear systems which would ascertain the stability of the systems.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declaration

Conflict of iönterest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Abualigah, L.; Diabat, A.: A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput. Appl. **32**, 15533–15556 (2020). https://doi.org/10.1007/s00521-020-04789-8

2. Arora, S.; Anand, P.: Chaotic grasshopper optimization algorithm for global optimization. Neural Comput. Appl. **31**, 4385–4405 (2019). https://doi.org/10.1007/s00521-018-3343-2
3. Castillo, O.; Lizarraga, E.; Soria, J.; Melin, P.; Valdez, F.: New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system. Int. J. Inf. Sci. **294**, 203–215 (2015)
4. Chang, Y.; Chan, W.; Chang, C.: T-S fuzzy model-based adaptive dynamic surface control for ball and beam system. IEEE Trans. Ind. Electron. **60**(6), 2251–2263 (2013)
5. Chang, Y.-H.; Chang, C.-W.; Tao, C.-W.; Lin, H.-W.; Taur, J.-S.: Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization. Int. J. Expert Syst. Appl. **39**, 3624–3633 (2012)
6. Chen, W.S.; Jiao, L.C.; Du, Z.B.: Output-feedback adaptive dynamic surface control of stochastic non-linear systems using neural network. IET Trans. Control Theory Appl. **4**(12), 3012–3021 (2010)
7. Chen, X.; Rongrong, Yu.; Huang, K.; Zhen, S.; Sun, H.; Shao, Ke.: Linear motor driven double inverted pendulum: a novel mechanical design as a testbed for control algorithms. Int. J. Simul. Model. Pract. Theory **81**, 31–50 (2018)
8. Dabiri, A.; Moghaddam, B.P.; Tenreiro Machado, J.A.: Optimal variable-order fractional PID controllers for dynamical systems. J. Comput. Appl. Math. **339**, 40–48 (2018)
9. Dass, A.; Srivastava, S.: Identification and control of dynamical systems using different architectures of recurrent fuzzy system. ISA Trans. **85**, 107–118 (2018)
10. Demeio, L.; Lenci, S.: Dynamic analysis of a ball bouncing on a flexible beam. Int. J. Sound Vib. **441**, 152–164 (2019)
11. Eaton, P.H.; Prokhorov, D.V.; Wunsch, D.C.: Neurocontroller alternatives for “fuzzy” ball-and-beam systems with nonuniform nonlinear friction. IEEE Trans. Neural Netw. **11**(2), 423–435 (2000)
12. Feng, H.; Yin, C.; Weng, W.; Ma, W.; Zhou, J.; Jia, W.; Zhang, Z.: Robotic excavator trajectory control using an improved GA based PID controller. Int. J. Mech. Syst. Signal Process. **105**, 153–168 (2018)
13. Filho, P.P.R.; Pires, S.; da Silva, P.; Praxedes, V.N.; Hemanth, J.; de Albuquerque, V.H.C.: Control of singularity trajectory tracking for robotic manipulator by genetic algorithms. Int. J. Computat. Sci. **30**, 55–64 (2019)
14. Giacomelli, M.; Padula, F.; Simoni, L.; Visioli, A.: Simplified input-output inversion control of a double pendulum overhead crane for residual oscillations reduction. Int. J. Mechatron. **56**, 37–47 (2018)
15. Giacomelli, M.; Padula, F.; Simoni, L.; Visioli, A.: Simplified input-output in version control of a double pendulum overhead crane for residual oscillations reduction. Int. J. Mechatron. **56**, 37–47 (2018)
16. Hashemi, M.; Askari, J.; Ghaisari, J.: Adaptive decentralised dynamic surface control for non-linear large-scale systems against actuator failures. IET Control Theory Appl. **10**(1), 44–57 (2016)
17. Hirschorn, R.M.: Incremental sliding mode control of the ball and beam. IEEE Trans. Autom. Control **47**(10), 1696–1700 (2002)
18. Jo, N.H.; Seo, J.H.: A state observer for nonlinear systems and its application to ball and beam system. IEEE Trans. Autom. Control **45**(5), 968–973 (2000)
19. Junfeng, Wu.; Zhang, W.; Wang, S.: A two-wheeled self-balancing robot with the fuzzy PD control method. Int. J. Math. Probl. Eng. **469491**, 1–13 (2012)
20. Kharola, A.; Patil, P.; Raiwani, S.; Rajput, D.: A comparison study for control and stabilization of inverted pendulum on inclined surface (IPIS) using PID and fuzzy controllers. Int. J. Perspect. Sci. **8**, 187–190 (2016)
21. Khooban, M.; Alfi, A.; Abadi, D.: Teaching–learning-based optimal interval type-2 fuzzy PID controller design: a nonholonomic wheeled mobile robots. Robotica **31**(7), 1059–1071 (2013). https://doi.org/10.1017/S0263574713000283
22. Kumar, S.; Solanki, V.K.; Choudhary, S.K.; Selamat, A.; Crespo, R.G.: Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT). Int. J. Interact. Multimed. Artif. Intell. **6**(1), 107–116 (2020)
23. Kumar, R.; Srivastava, S.; Gupta, J.R.P.: Online modeling and adaptive control of robotic manipulators using Gaussian radial basis function networks. Neural Comput. Appl. **30**(1), 223–239 (2016)
24. Kumar, R., Srivastava, S., Gupta, J.R.: Modeling and control of one-link robotic manipulator using neural network based PID controller. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2016)
25. Lotfipour, A.; Afrakhte, H.: A discrete teaching_learning-based Optimization algorithm to solve distribution system reconfiguration in presence of distributed generation. Int. J. Electr. Power Energy Syst. **82**, 264–273 (2016)
26. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Al-Zoubi, A.M.; Mirjalili, S.: Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst. Appl. **117**, 267–286 (2019). https://doi.org/10.1016/j.eswa.2018.09.015
27. Mehedi, I.M.; Al-Saggaf, U.M.; Mansouri, R.; Bettayeb, M.: Two degrees of freedom fractional controller design: Application to the ball and beam system. Int. J. Meas. **135**, 13–22 (2019)
28. Moezzi, R.; Trieu Minh, V.; Tamre, M.: Fuzzy logic control for a ball and beam system. Int. J. Innov. Technol. Interdiscip. Sci. **1**(1), 39–48 (2018)
29. Mohammed, E.A.; Mohamed, A.; Mitani, Y.: Genetic-moth swarm algorithm for optimal placement and capacity of renewable DG sources in distribution systems. IJIMAI **5**, 105–117 (2019)
30. Nguyen, H.; Bui, X.-N.: Predicting blast-induced air overpressure: a robust artificial intelligence system based on artificial neural networks and random forest. Int. J. Nat. Resour. Res. **28**(3), 893–907 (2019)
31. Niemann, H.; Poulsen, J.K.: Design and analysis of controllers for a double inverted pendulum. ISA Trans. **44**, 145–163 (2005)
32. Oh, S.K.; Jang, H.J.; Pedrycz, W.: The design of a fuzzy cascade controller for ball and beam system: A study in optimization with the use of parallel genetic algorithms. Int. J. Eng. Appl. Artific. Intell. **22**, 261–271 (2009)
33. Prasad, L.B.; Tyagi, B.; Gupta, H.O.: Optimal control of nonlinear inverted pendulum system using PID controller and LQR: performance analysis without and with disturbance input. Int. J. Autom. Comput. **11**(6), 661–670 (2014)
34. Raj, M.; Singh Jalal, A.; BhaskarSemwal, V.: Optimized fuzzy controller for cable robot. Int. J. Interact. Multimed Artif. Intell. **6**(1), 51–61 (2020)
35. Rao, R.V.; Savsani, V.J.; Vakharia, D.P.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Int. J. Comput.-Aided Design **43**(3), 303–315 (2011)
36. Roose, A.I.; Yahya, S.; Al-Rizzo, H.: Fuzzy-logic control of an inverted pendulum on a cart. Int. J. Comput. Electric. Eng. **61**, 31–47 (2017)
37. Sahu, R.K.; Panda, S.; Rout, U.K.; Sahoo, D.K.: Teaching learning based optimization algorithm for automatic generation control of power system using 2-DOF PID controller. Int. J. Elect. Power Energy Syst. **77**(1), 287–301 (2016)
38. Sahu, BK; Pati, Swagat; Mohanty, Pradeep Kumar; Panda, Sidhartha (2015). Teaching–learning based optimization algorithm based fuzzy-PID controller for automatic generation control of

multi-area power system. Applied Soft Computing, 27: 240–249. https://doi.org/10.1016/j.asoc.2014.11.027
39. Saremi, S.; Mirjalili, S.; Lewis, A.: Grasshopper optimisation algorithm: theory and application. Int. J.Adv. Eng. Softw. **105**, 30–47 (2017)
40. Shabanpour-Haghighi, A.; Seifi, R.; Niknam, T.: A modified teaching-learning based optimization for multi-objective optimal power flow problem. Energy Convers. Manage. **77**(1), 597–607 (2014)
41. Singh, M.; Srivastava, S.; Gupta, J.R.P.; Handmandlu, M.: Identification and control of a nonlinear system using neural networks by extracting the system dynamics. IETE J Res **53**(1), 43–50 (2007)
42. Srivastava, S.; Singh, M.; Hanmandlu, M.; Jha, A.N.: New fuzzy wavelet neural networks for system identification and control. Appl. Soft. Comput. **6**(1), 1–17 (2005)
43. Srivastava, S.; Singh, M.; Hanmandlu, M.; Jha, A.: Modeling of non-linear systems by FWNNS and their intelligent control. Int. J. Adapt. Control Signal Process **19**(7), 505–530 (2005)
44. Sun, Y.; Qiang, H.; Mei, X.; Teng, Y.: Modified repetitive learning control with unidirectional control input for uncertain nonlinear systems. Neural Comput. Appl. **30**(6), 2003–2012 (2018)
45. Sun, N.; Yiming, Wu.; Che, He.; Fang, Y.: An energy-optimal solution for transportation control of cranes with double pendulum dynamics: design and experiments. Int. J. Mech. Syst. Signal Process. **102**, 87–101 (2018)
46. Wang, C.; Lin, Y.: Decentralised adaptive dynamic surface control for a class of interconnected non-linear systems. IET Control Theory Appl. **6**(9), 1172–1181 (2012)
47. Wang, H.; Liu, P.X.; Li, S.; Wang, D.: Adaptive neural output-feedback control for a class of non lower triangular nonlinear systems with un modeled dynamics. IEEE Trans. Neural Netw. Learn. Syst. **29**(8), 3658–3668 (2018)
48. Wang, L.; Ni, H.; Zhou, W.; Pardalos, P.M.; Fang, J.; Fei, M.: MBPOA-based LQR controller and its application to the double-parallel inverted pendulum system. Int. J. Eng. Appl. Artif. Intell. **36**, 262–268 (2014)
49. Yoo, S.J.; Park, J.B.; Choi, Y.H.: Decentralized adaptive stabilization of interconnected nonlinear systems with unknown non-symmetric dead-zone inputs. An Int J Autom **45**, 436–443 (2009)
50. Zhang, H.; Li, : Indirect adaptive fuzzy control of SISO nonlinear systems with input-output nonlinear relationship. IEEE Trans. Fuzzy Syst. **26**, 2699–2708 (2018)