Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2022) 46:785–804
https://doi.org/10.1007/s40998-022-00506-4

RESEARCH PAPER

Hybrid State of Matter Search Algorithm and its Application to PID Controller Design for Position Control of Ball Balancer System

Neha Khanduja[1] · Bharat Bhushan[1]

Received: 14 June 2021 / Accepted: 17 January 2022 / Published online: 1 June 2022

Abstract

To optimize the proportional integral derivative controller (PID) controller, this study designs a self-balancing controller utilizing an opposition-based chaotic State of Matter search (SMS)Algorithm. The feedback in a loop-based underactuated system which is multivariate, electromechanical, and nonlinear is represented by ball balancer systems. Initially, a two-degree-of-freedom (2DoF) ball balancer system is used to get the mathematical model of the ball balancer system, which is operated by a PID controller. The original SMS algorithm works by simulating the states of matter concept. The hybridized Chaotic state of matter search with Elite opposition-based learning (CSMSEOBL) is an improved version of SMS in which the concept of Chaotic maps and opposition-based learning is incorporated. Any metaheuristic algorithm that uses a chaotic map improves its randomness and efficacy. The SMS algorithm's diversification capability is improved when the Elite opposition-based learning (EOBL) principle is used. It is used to tune parameters of the PID controller to improve the transient response of a ball balancer system by minimizing the objective function Integral Time Absolute Error(ITAE). As per the analysis, hybridized CSMSEOBL-based PID controller outperformed the traditional PID, PSO-PID, SFS-PID, and SMS-PID Controllers.

Keywords Ball balancer system · Chaotic maps · Elite opposition—based learning(EOBL) · State of matter search (SMS) algorithm

1 Introduction

Unstable electromechanical and nonlinear ball beam balancer is an underactuated system. An underactuated system is a problem that happens in a number of control scenarios and may be remedied using various strategies, such as the employment of intelligent controllers or autonomous decision development (Singh and Bhushan 2020). There are many process control systems that have these features by default, which makes it difficult to design a feasible control strategy for them (Lawrence 2020). Experts in the field have started to investigate the behavior of different linear, nonlinear, model-free, and intelligent controllers because of the wide range of frameworks. Self-adjusting and consistent control of various processes, such as ship control, horizontal stabilization of an airplane during turbulent airflow and landing, the twin-rotor multi-input multiple output system, inverted pendulum, hovercraft, Furuta pendulum, ball beam system and ball plate system, are the goals of these controllers (Aranda et al. 2006; Nowopolski 2013).

For ball and beam systems, there are a few possible hardware setups. These devices all have the same dynamics and control concept: (Oh et al. 2011; Keshmiri et al. 2012; Muskinja et al. 1997). On a beam, the ball may go either left or right, or in the opposite direction of the beam's axis. To regulate the spin of the beam, a voltage-driven servomotor is coupled to it, which in turn affects the ball's speed and placement. (Muškinja and Rižnar 2015). The ball and beam experiment serves as a benchmark for various controller designs, which may be divided into two groups: model-based and non-model-based control systems, in terms of controlled device modeling and recognition criteria (Jimenez and Yu 2007). It is necessary to include in model-based control systems regulated system states that cannot be estimated clearly. Non-model-based control

✉ Neha Khanduja
nehakhanduja.dce@gmail.com

[1] Department of Electrical Engineering, DTU, Delhi, India

systems are reliant on real-time sensor output data for their operation. Model-based control architecture has the drawback of being heavily dependent on the state observer's model fidelity to function properly. Nonlinear dynamics are impossible to represent in a system model; therefore, model-based control loses effectiveness over time owing to changes in the physical structure of the system or wear and tear. Some non-model-based control designs, such as the ball and beam arrangement, have a practical specification that differs from controller designs based on models. In addition, nonlinear processes are linearized, which has a significant impact on device response time. This prompted the development of a variety of nonlinear control strategies to resolve the issues in underactuated systems. Many nonlinear controllers have been suggested, including the Lagrangian, lambda process, and backstepping controller (Choukchou-Braham et al. 2014; Rudra et al. 2017).

Bolívar-Vincenty and Beauchamp-Báez 2014) uses Newtonian mechanics to develop the mechanism model. (Maalini et al. 2016) derives the system's model using Lagrange's technique and proposes two degrees of freedom. The nonlinear model is initially linearized around the equilibrium to get a state-space model (Dušek et al. 2017). It is represented using Lagrange's second form equation (Burghardt and Giergiel 2011). Furthermore, it was stabilized utilizing traditional and modern control methods (Li and Yu 2010). (Yu 2009) controls the nonlinear model using a PD controller. (Dušeket al. 2017) regulates the machine utilizing SIMC-based PID and H infinity. (Lawrence 2020) used a PID controller, a neural network, and LQR regulation to control the gadget. Nonlinear control has been created using adaptive iterative learning approaches such as the Kalman particle filter (Li 2020). The particle filter approach has a severe flaw in that it relies on estimations that grow exponentially with the number of state variables (Sutharsan et al. 2012). Traditional nonlinear approaches employed feedback linearization, as well as partial feedback linearization (Spong 1994) for underactuated mechanical systems. This approach turned a nonlinear system into a linear one by canceling nonlinearities.

The complexity of maintaining underactuated systems necessitates a control approach capable of attaining steady-state activity. To do so, underactuated systems are benchmarked using a ball and plate approach. Due to its significant nonlinearity and instability, balancing control for ball and beam experiments is regarded a difficult control system design and operation challenge. For all ball and beam systems, the underlying device dynamics and control theory is the same: A ball on a beam travels in two directions, left and right, or positive and negative. In turn, the beam rotates, affecting the ball's acceleration and location (Muškinja and Rižnar 2015).

Controlling the ball on the plate structure using a nonlinear proportional integral derivative (PID) controller has been explored classically (Sunand Li 2012). The GKYPL approach improves the PID response and compares it to a standard PID in terms of the relentless state reaction (Shuichi and Ichihara 2013). For best efficiency within reasonable restrictions, several studies have employed the Ziegler Nicholas technique and trial and error (Saad and Khalfallah 2017; Ali et al. 2017). To address the issues, a hybrid fuzzy controller with model-based PID has been developed (Isa et al. 2018; Başçi and Kaan 2017). The fuzzy controller rejects disturbances and has 0% steady-state error. Back-venturing controllers were built based on the Lyapunov dependability hypothesis(Ker et al. 2007). Interference rejection controllers (Pinagapani et al. 2018) and metaheuristic optimization algorithms (Ali et al. 2019) were used to achieve required tracking efficiency. Model predictive controllers were also extensively utilized for ball balancer systems (Bang and Lee 2019). The main disadvantage of these traditional techniques is a protracted settling time and peak overshoot. The self-balancing control, trajectory detection, and ball position monitor of the ball and plate system were also achieved using intelligent and hybrid controllers such as fuzzy (Fan et al. 2004; Dong et al. 2011). Rahmat et al. (2010) employed a PID controller, a neural network, and LQR control. (Márton et al. 2008) compares LQR subspace stability with integrated error metric techniques. Ali et al. (2019) and Rahmat et al. (2010) tunes the controller using SA and CSA heuristics. In (Saad et al. 2012), GA and DE algorithms are employed to adjust the PID controller. The process is controlled by LQR, which is adjusted by GA (Keshmiri et al. 2012). Mahmoodabadi et al. (2016) developed SIRMs dynamically connected fuzzy controllers for inverted pendulum and ball beam structures utilizing convergence and divergence operators. Although several control techniques exist for self-balancing balancer systems, the PID controller is often employed in functional engineering applications. The PID controller has a simple form, great dependability, and stability. The difficulty with traditional PID controllers is parameter adjustment, which is a major drawback. There are several methods for tuning PID parameters that can be found in the literature. Various intelligent methods such as fuzzy (Khodadadi and Ghadiri 2018) and neural network (Rivas-Echeverria and Rios-Bolivar 2001), genetic (Feng et al. 2018), and evolutionary algorithms(Ribeiro et al. 2017) are used to tune PID controller parameters.

Particle Swarm Optimization (PSO) was created by Kennedy and Eberhart (Kennedy and Eberhart 1995). It uses a fundamental technique that replicates swarm behavior in birds and fish to direct the particles to global optimal solutions. It is a swarm-based stochastic optimization technique. It solves issues based on social interaction and does not need a differential optimization problem, as classic optimization algorithms do (Yan et al. 2013). The PSO algorithm has several benefits. It is simple to set up, performs well in global searches, is unaffected by architectural variable scaling, and can be readily parallelized (Gong et al. 2009).

The Random Inertia Weight (Random-w) approach was proposed by Eberhart and Shi to increase PSO convergence (Eberhart and Shi 2002). Using a linearly declining or Time-Varying Inertia weight (TV-w) approach, PSO efficiency and output improved from 0.9 to 0.4 (Xin et al. 2009). The LDIW-PSO method suffers from premature convergence when handling complicated optimization problems (Jayachitra and Vinodha 2015). A PSO-based PID controller with adaptive inertia weight (Adaptive-w) can arrange multiple performance metrics and give an effective approach for comparing convergence, reliability, and robustness. According to Meenakshipriya et al. (2018), PSO-CDM-PID controller delivers consistent and steady performance on unstable ball and beam arrangement. Roy et al. employed PSO to calibrate the PD trajectory controller on the ball and plate system in 2014 (Roy and Kar 2014). The PSO method is known for its early search efficiency but is susceptible to local extremes. To solve its weaknesses, a novel hybrid PSO algorithm (Das and Abraham 2008) was developed (DEPSO). An online training strategy of weighting factors for PID neural network controllers utilizing the DEPSO algorithm is provided in Han et al. (2012). The discrete PSO method (Thangaraj et al. 2011) may solve discrete and combinatorial optimization issues.

The stochastic fractal search (SFS) method (Salimi 2015) uses randomized fractals created by the DLA process. Current metaheuristic weaknesses including premature convergence and poor solution consistency were addressed. The SFS method may discover a solution with the minimum or minimal error relative to the optimal solution in a few iterations, increasing solution efficiency and convergence time. (Çelik 2018). SFS uses the fractal mathematical idea. SFS optimizes random walks using fractal features. Engineering, Physics, Medicine, and Chemistry are only a handful of SFS's successful research topics (Khalilpourazari et al. 2020). Mellal and Zio (Mellal and Zio 2016) developed SFS to increase machine stability. They showed the SFS could develop viable solutions. (Tyagi et al. 2016) and (Dubey et al. 2018) employed SFS to improve multi-objective solar–wind–thermal systems and wind-integrated multi-objective power dispatch. It also only involves adjusting one parameter, the maximum diffusion number (MDN), which enhances the algorithm's performance. (Salimi 2015) compares this algorithm's performance to other state-of-the-art algorithms on various classic benchmark functions and popular computer design functions. Evaluating PID parameters for model identification and scalable beam device vibration control is described in Rahman et al. (2017).

The SMS algorithm simulates matter states. Individuals employ evolutionary methods based on thermal energy motion system physical ideas to simulate SMS molecules. These methods increase population diversity while preventing particle concentrations below a local minimum. The algorithm considers each state of matter at a specific exploration–exploitation ratio. As a consequence, the development process is divided into three phases: gas, liquid, and solid. Molecules (individuals) behave differently in each state. So the technique balances exploration and exploitation better while maintaining the evolutionary approach's powerful search capabilities (Cuevas et al. 2014), the search for states of matter based on drift operators (DSMS). The important technique is to employ the drift operator to keep the sense of location while dropping the concept of velocity. (Al-Attar Ali Mohamed et al. 2016) The SMS technique employs control variables to simplify the ANFIS controller's dynamic setup. The original SMS control parameters were adaptively reduced to obtain a scale-minimum operator optimizer. The SMS technique improves efficiency and solution consistency, proving its capacity to tackle the dynamic problem of smart power distribution for PHEVs (Valdivia-Gonzalez et al. 2017).

Most of the algorithms above have problems throughout their development. These algorithms' success depends heavily on the established control parameters (Yan et al. 2013). In other words, good parameter selection is necessary for evolution to obtain the best solution. To compensate for the difficulty of choosing the optimum monitoring parameters, a PSO algorithm delivers near-optimal solutions (Joorabian and Afzalan 2014). DE generally adopts a stringent selection criteria that sets parent against child. As a consequence, DE is susceptible to early population selection and premature convergence (Das and Suganthan 2010). Because the fractal's formation failed to discover and use the whole search domain, the stochastic movement of each particle could not be ideal (Lagunes et al. 2021). As a consequence, hybridization, or combining optimization techniques, has lately gained popularity (Ali et al. 2016; Awad et al. 2017). The derived method almost always outperforms the original (Farah and Belazi 2018).

Chaos theory is a newer engineering method. Chaos theory studies nonlinear dynamical systems that are very sensitive to their starting circumstances (Majumdar et al. 2000) means Small changes in the starting conditions have a big influence on the system's outcome. Due to its versatility and durability, chaos analysis is a viable technique for hybridization (Farah and Belazi 2018). Previously, chaotic sequences were employed to tweak metaheuristic optimization parameters to prevent local optimums. These include Multiverse (Sayed et al. 2018), Cuckoo Search (Wang and Zhong 2015), Krill herd (Wang et al. 2014), Firefly (Su et al. 2014), and Jaya (Farah and Belazi 2018). Similarly, Opposition-based Learning (OBL) (Tizhoosh 2005) is a newer AI paradigm. It is used to improve the performance of optimization algorithms such as crow search (Shekhawat and Saxena 2020), particle search (Mandal and Si 2015), Gray Wolf (Nasrabadi et al. 2016), and Water Wave Optimization Algorithms (Nasrabadi et al. 2016). OBL's main purpose is to move solutions from one search area to another (Zhao et al. 2017). Also, EOBL beats OBL in terms of convergence time and discovery capabilities (Wang and Huang 2018).

(Chaotic State of Matter search with Elite Opposition-based learning) is an upgraded version of the SMS algorithm (state of matter search) that combines Chaotic Maps with Elite Opposition-based learning (EOBL). Thermal energy mobility is based on the SMS algorithm's basic idea. Every state of matter has its own diversification-intensification ratio, including solid, liquid, and gas. The chaotic definition is utilized by systems that are sensitive to the starting state. It increases chance. This chaotic SMS algorithm approach is utilized to increase SMS convergence. The present algorithm's Diversification abilities are enhanced by combining two crucial OBL qualities, global search and quick convergence. The proposed technique outperforms current algorithms like PSO, SFS SMS, and a hybrid metaheuristic algorithm dubbed CSMSEOBL in MATLAB. Major Contribution of this work is given as follows :

A new hybridized algorithm is used for position and angle control of ball balancer system.

A comparative study has been presented in a well-defined manner with two recent and other metaheuristic algorithms.

Performance analysis has been done by set point response analysis, error convergence analysis and trajectory analysis.

The remaining part of the paper is structured as follows: Section 2 discusses a two-degree-of-freedom Ball balancer system, section 3 elaborates the Controller design with different existing and proposed hybrid metaheuristic algorithm, section 4 explains the PID controller and Section 5 summarizes the findings and analysis, while Section VI contains the conclusion and possible scope.

2 Mathematical Modeling of Ball Balancer System

The 2 DOF Ball Balancer, or 2DBB, as seen in Fig. 1, is made up of a plate on which a ball can be put and moves freely. The plate can be swiveled in any direction by mounting it on a two-degree-of-freedom (2 DOF) gimbal. The ball's location is measured using an overhead USB camera and a vision unit. Quanser Rotary Servo Base Unit (SRV02) systems are the two servos under the surface. Two DOF gimbals are used to attach each of them to a side of the plate. The tilt angle of the plate can be changed by controlling the direction of the servo load gears to balance the ball in an ideal planar position. The overhead digital camera takes photographs of the plate, which are then processed using the provided Quanser image processing blocks to determine the ball's x and y locations. A FireWire attachment is used to easily pass images to the PC. As a result, the 2 DOF Ball Balancer is designed as two decoupled "ball and beam" structures, with the assumption that the angle of the x-axis servo just influences ball movement in the x-direction. The y ball motion is similar. Section 2.1.1 develops the equation of motion for the ball's motion around the x-axis in comparison to the plate angle, while Sect. 2.1.2 integrates the servo angle into the model.

The open-loop structure of the 2D Ball Balancer is illustrated in Fig. 1. The dynamics between the servo input motor voltage and the resulting load angle are represented by the SRV02 transfer function $B_s(S)$. $B_{bb}(S)$ is a transfer function that defines the dynamics between the servo load gear's angle and the position of the ball (s). This is a decoupled framework because the y-axis response is not affected by the x-axis actuator. The dynamics of each axis are the same as all SRV02 systems have the same hardware. Figure 2 depicts the block diagram for a single-axis 2D Ball Balancer, abbreviated as 1DBB.

The complete transfer function of the 1DBB plant is given by

$$B(S) = B_{bb}(S)\,B_s(S) \tag{1}$$

where

$$B_{bb}(S) = \frac{C(S)}{\alpha_1(S)}$$

And

$$\mathrm{B}_s(S) = \frac{\alpha_1(S)}{V_m(S)}$$

The movement of the ball concerning the servo's load angle is defined by the 1DBB transfer function.

$$m_b\ddot{C}(t) = \sum F = F_{x,t} - F_{x,r} \tag{2}$$

where $F_{x,r}$ is the inertial force of the ball and $F_{x,t}$ is the gravitational axial force. The force produced by the ball's momentum must be equal to the force produced by gravity for the ball to be stable at a given moment, i.e., be in equilibrium.

When the incline is positive, the force acting in the positive x-direction is given as follows:

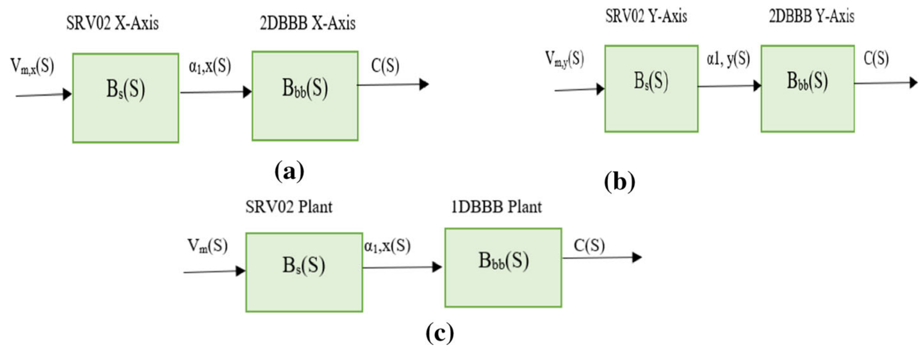


Fig. 1 Ball balancer open-loop block diagram **a** Servo X-axis **b** Servo Y-axis **c** Open-Loop block diagram of 1-D plant

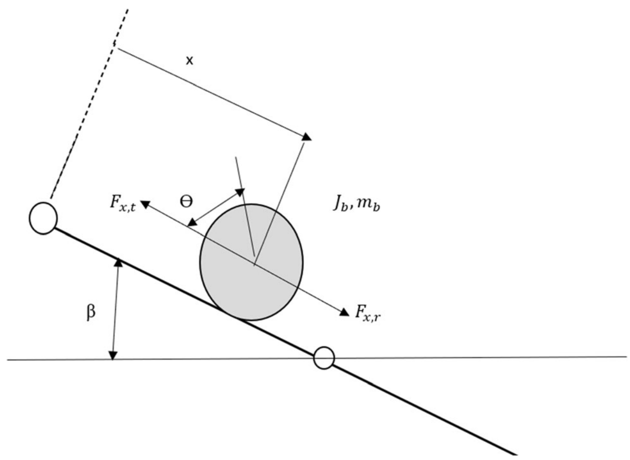


Fig. 2 Free body diagram of Ball balancer

$$F_{x,t} = m_b g \sin \beta(t) \tag{3}$$

The force generated by the ball's rotational spin is given as follows:

$$F_{x,r} = \frac{\tau_b}{r_b} \tag{4}$$

$$\tau_b = J_b \ddot{\theta}_b(t) \tag{5}$$

where r_b – radius of sphere

τ_b – torque

θ_b – ball angle

J_b – ball inertia

The force acting in the x-direction on the ball as a result of its momentum is given as follows:

$$F_{x,r} = \frac{J_b \ddot{x}_b(t)}{r_b^2} \tag{6}$$

From Eqs. (2), (3), and (6)

$$m_b \ddot{C}(t) = m_b g \sin \beta t - \frac{J_b \ddot{C}(t)}{r_b^2} \tag{7}$$

2.1 Calculation of Servo Angle

The action of the ball and plate system in terms of complex variables: the position of the ball around the servo load angle reflects motion and time. The servo angle and the beam have the following relationship:

$$\sin(\beta(t)) = \frac{2 \sin(\gamma_l(t)) r_a}{l_t} \tag{8}$$

where

r_a – distance between the coupled joint and the output gear shaft of the SRV02.

l_t – Table length

The dynamic variables are linearized around $\theta_1 = 0$ to find the dynamic variables for describing the rotation of the ball (equation of motion) that corresponds to the servo angle θ_1. The mathematics for the servo and plate angle relationship is as follows:

$$C(\dot{t}) = \frac{2 m_b g r_a r_b^2 \sin \gamma_l(t)}{l_t \left(m_b r_b^2 + J_b\right)} \tag{9}$$

The sine function $\sin \gamma_l$ is approximated as γ_l to linearize the equation of motion, and the final equation of motion for a 1D ball balancer is given as follows:

$$C(\dot{t}) = \frac{2 m_b g \gamma_l(t) r_a r_b^2}{l_t \left(m_b r_b^2 + J_b\right)} \tag{10}$$

3 Controller Design

The inner loop is stabilized first, so the outer loop is stabilized. The inner loop's job is to keep track of the angle of the motor (Fig. 3). The motor angle should track the reference signal, so the inner controller should be pro-

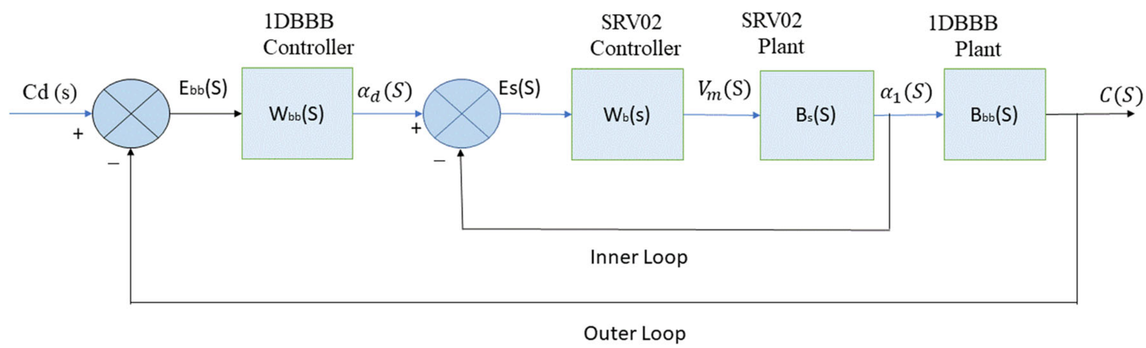


Fig. 3 Block diagram of ball balancer system

grammed accordingly. To control the ball angle, the outer loop uses the inner feedback loop. As a result, the inner loop must come first. The following are some of the control strategies that have been designed and tested for balancing and regulating the ball balancer system.

3.1 State of Matter Search Algorithm

The State of Matter Search Algorithm (SMS) is a nature-inspired evolutionary algorithm that can solve MIMO-style global optimization problems. It works on the principle of thermal energy motion. This algorithm simulates three states of matter: solid, liquid, and gas, each with a different diversification-intensification ratio. SMS (States of Matter Search) is a method of searching for states of matter (Zhou et al. 2015). The entire optimization process is broken down into three stages: 1. gas state; 2. liquid state; and 3. solid-state. The first stage of the SMS system is the gas condition. Molecules undergo extreme displacements and collisions during this stage. The gas state lasts for half of the total number of iterations in the SMS optimization process (Fig. 4). The liquid state is the next step in the SMS optimization process. In contrast to the gas state, the motion and collisions shown by the molecules inside the search space are more restricted at this point. The liquid state lasts for 40% of the SMS evolution process's cumulative iterations. The solid-state represents the third and final stage of the SMS optimization process. In contrast to the previous SMS points, forces between particles are much greater in this stage, preventing particles from moving freely. Just ten percent of the overall SMS iterations are accumulated in the solid-state (Valdivia-Gonzalez et al. 2017).

3.1.1 Initialization

The transition in molecule location as the phase proceeds is represented by the direction vector operator. Each dimensional molecule p_i in the population is given an n-dimensional direction vector d_i. The direction vector, which has a value that varies between [− 1, 1], governs the flow of particles. A new direction vector is assigned to each molecule as follows:

$$d_i^{c+1} = d_i^c \left(1 - \frac{c}{\text{gen}}\right) 0.5 + a_i \tag{11}$$

where attraction unitary vector, a_i is

$$a_i = \frac{(p_{\text{best}} - p_i)}{\|p_{\text{best}} - p_i\|} \tag{12}$$

p_{best}-The best molecule of population P; p_i-the molecule i of population P,k -No. of the current iteration,$gen-$ total number of iterations.

Each molecule's velocity v_i is given by

$$v_i = d_i * v_{st} \tag{13}$$

v_{st} starting velocity is given as

$$v_{st} = \frac{\sum_{j=1}^{n} b_j^h - b_j^l}{n} * \beta \tag{14}$$

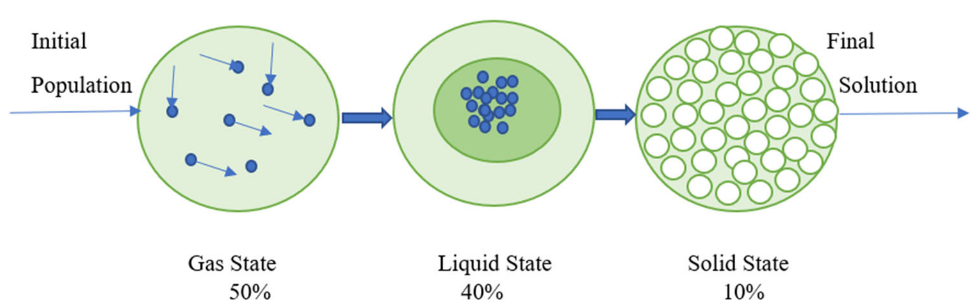


Fig. 4 The evolution process of the State of Matter Search Algorithm

b_j^h—The j parameter's upper limit.

b_j^l—The j parameter's the lower limit.

β—phase constant with a value between [0, 1].

For each molecule, the position update equation is

$$p_{i,j}^{k+1} = p_{i,j}^k + v_{i,j} * \mathrm{rand}(0,1) * \rho * \left(b_j^h - b_j^l\right) \tag{15}$$

where $0.5 \leq \rho \leq 1$.

3.1.2 Collision

As molecules interact with one another and their distance is less than a predetermined value, a collision occurs. i.e.,$\left\|p_i - p_q\right\| < r$

$$r = \frac{\sum_{j=1}^{n} b_j^h - b_j^l)}{n} * \alpha \tag{16}$$

α- phase constant ranging [0, 1].

3.1.3 Random Position

Using a suitable criterion within the search space, the random position operator allows for a shift in molecule position.

$$R = \begin{cases} b_j^l + R(0,1) * \left(b_j^h - b_j^l\right) & \text{with probability } H \\ p_{i,j}^{k+1} & \text{with probability } 1 - H \end{cases} \tag{17}$$

R—a random number varying from [0, 1].

3.1.4 Updating of Elements

If $p_{\mathrm{best}}^{\mathrm{k}}$ (i.e., current best molecule) is greater than p_{best}^{k-1} (i.e., best individual so far), it is revised with $p_{\mathrm{best}}^{\mathrm{k}}$; otherwise, nothing changes.

3.2 Particle Swarm Optimization (PSO)

Kennedy and Eberhart (Kennedy and Eberhart 1995) suggested the particle swarm optimization (PSO) algorithm, which is a stochastic optimization technique focused on the swarm. The PSO algorithm mimics the social behavior of animals such as insects, clusters, birds, and fish. The PSO algorithm's main architecture concept is based on two studies: One is the evolutionary algorithm, which, uses a flock mode to find a wide area in the solution space of the optimized objective function at the same time. Artificial life, on the other hand, is the study of artificial structures with life-like characteristics (Wang et al. 2017). The PSO approach is a population-based search algorithm in which each individual is referred to as a particle and represents a potential solution. Each particle moves through the search space at a variable speed that is dynamically adjusted based on its own and other particles' flying experiences(Nagaraj 2008). Each particle in PSO aims to better itself by imitating good peers' traits. Furthermore, since each particle has a brain, it can recall the best place in the search space it has ever visited. p_{best} denotes the place with the best fitness, while g_{best} denotes the absolute best of all the particles in the population (Lahoty and Parmar 2013).

The position and velocity update equations of PSO are given by

$$v_{i.g}^{(t+1)} = w.v_{i.g}^{(t)} + a_1 * \mathrm{rand}() * \left(\mathrm{pbest}_{i.g} - x_{i.g}^t\right) + a_2 * \mathrm{rand}() * \left(\mathrm{gbest}_g - x_{j.g}^t\right) \tag{18}$$

$$x_{i.g}^{(t+1)} = x_{i.g}^{(t)} + v_{i.g}^{(t+1)} \tag{19}$$

$i = 1,\ 2\ldots\ldots\ n$; $g = 1,\ 2\ldots\ m$. where n—number of particles in a group; m—number of members in a particle; t—pointer of iterations (generations); $v_{i.g}^{(t)}$—velocity of particle i at iteration t; w—inertia weight factor; a_1, a_2—acceleration constant; rand ()—random number between 0 and 1; $x_{i.g}^{(t))}$—current position of particle i at iteration t; pbest$_i$-p_{best} of particle i; $g_{\mathrm{best}} - g_{\mathrm{best}}$ of the group

$$w = \frac{w_{\max} - w_{\min}}{\mathrm{iter}_{\max}} * \mathrm{iter} \tag{20}$$

The inertia weight w is calculated using Eq. (20). A good choice of w strikes a good mix between global and local searches, taking fewer iterations on average to get at a suitably optimum solution.

3.3 Stochastic Fractal Search Algorithm

A trade-off is encountered between accuracy and time consumption with Fractal Search since it is a dynamic algorithm in which the number of agents in the algorithm is changed. As a result, a new version of Fractal Search that addresses the issues of Fractal Search is introduced named Stochastic Fractal Search (Salimi 2015). The aim of stochastic fractal search (SFS) is to find a probabilistic or optimum search pattern that can provide a better solution to an optimization process. To find the search space, SFS uses the diffusion property found in random fractals. Diffusion and redesign are the two primary mechanisms involved (Bhatt et al. 2019).

The SFS algorithm's steps are as follows:

Begin: Each particle's (points) location is initialized randomly based on the problem specifications by defining maximum and minimum bounds as follows:

Begin: Each particle's (points) location is initialized randomly based on the problem specifications by defining maximum and minimum bounds as follows:

$$P = \mathrm{lb} + r(\mathrm{ub} - \mathrm{lb}) \tag{21}$$

r is a random number with a uniform distribution (generated by Gaussian distribution) and a range of [0, 1].

The probability value for each point i in the group is then assigned using the following equation, which follows a simple uniform distribution:

$$\mathrm{Pa_i} = \frac{\mathrm{rank}P_\mathrm{i}}{N} \tag{22}$$

where rank P_i denotes the position of point.

P_i with the other points in the set, and N denotes the total number of points in the group.

Equation (23) is used to update the jth element of P_i; otherwise, it stays consistent.

$$P'_\mathrm{i}(j) = P_\mathrm{r}(j) - r(P_\mathrm{t}(j) - P_\mathrm{i}(j) \tag{23}$$

where P'_i is now in a new updated role.P_r,P_t and P_i are three points in the category that were chosen at random.

If the condition $\mathrm{Pa_i} < r$ r holds for a new point P'_i the current location of P'_i is updated according to Eqs. (24) and (25), otherwise there is no change (Table 1).

$$P''_\mathrm{i} = P'_\mathrm{i}(j) - \hat{r}\left(P'_t - BP\right)\left|r' \le 0.5\right. \tag{24}$$

$$P''_\mathrm{i} = P'_\mathrm{i}(j) + \hat{r}\left(P'_t - P'_r\right)\left|r' > 0.5\right. \tag{25}$$

BP-The best point out of all the points.

3.4 Chaotic Maps

Chaos theory investigates the behavior of systems that is strongly dependent on their initial conditions. Instead of random numbers, they can generate a wide variety of numbers. Since the chaotic system's behavior appears to be unpredictable, evolutionary algorithms may use chaotic systems to provide the necessary randomness (Sayed et al. 2018). Chaos search arises as an important strategy for hybridization due to its diverse characteristics and ergodicity. The use of chaotic sequences created by chaotic maps to replace random numbers falls into one of three categories in the integration of chaos in a metaheuristic algorithm (Table 1). The second uses a chaotic map function to incorporate local search methods, while the third uses a chaotic map function to generate algorithm control parameters (Jordehi 2014). To improve the efficiency of chaos theory in terms of both diversification and intensification, eight chaotic maps are combined with the SMS algorithm (Table 2).

3.5 Elite Opposition-Based Learning

Opposition-based Learning (OBL) is a novel idea in artificial intelligence. It has been demonstrated to be an effective method for improving the performance of several optimization algorithms (Nasrabadi et al. 2016; Zhou et al. 2016; Wang et al. 2013). Tizhoosh proposed OBL in Tizhoosh (2005), which is essentially a machine intelligence method. It takes into account both the present individual and its opposite individual at the same time to provide a better estimate for a current candidate solution. OBL can increase the probability of locating solutions that are closer to the optimum solution. However, OBL may not be appropriate for all optimization issues. When tackling multimodal issues, for example, the converted candidate may deviate from the optimal solution. to avoid this scenario, a new population-based elite selection method is applied (Zhao et al. 2017). Its basic concept is to compute and analyze the opposing solution at the same time to find a feasible solution, and then pick the best one as the individual of the future generation. The elite individual is defined in this study as the individual having the highest fitness value in the population. An example should be used to demonstrate the definition of an Elite opposition-based solution.

Let elite individual of the population is $P_e = \left(p_{e,1}, p_{e,2}, \ldots p_{e,n}\right)$.

For the individual $\left(P_i = p_{1,1}, p_{i,2}, \ldots p_{i,n}\right)$, the elite opposition solution can be defined as $P_i^* = p_{i,1}^*, p_{i,2}^*, \ldots p_{i,n}^*$

$$p_{i,j}^* = h\left(da_j + db_j\right) - p_{e,j}, i = 1, 2, \ldots, N; j = 1, 2, \ldots, n) \tag{26}$$

where N-population size, n is dimension of P, h ε U (0,1) and $\mathrm{da}_{j,}\mathrm{db}_j$ are dynamic bound of the jth decision variable. The following equation may be used to calculate da_j and db_j

$$\mathrm{da_j} = \min\left(p_\mathrm{i,j}\right),\ \mathrm{db_j} = \max\left(p_\mathrm{i,j}\right) \tag{27}$$

When we all know, as the search space shrinks, the algorithm may become trapped at the local minimum. As a result, $\mathrm{da_j}$ and $\mathrm{db_j}$ will be updated every 50 generations in the proposed methodology. It can, therefore, cause $p_\mathrm{i,j}$ to bounce out of $(\mathrm{da_j db_j})$; if this happens, apply the equation below to reset $p_\mathrm{i,j}^*$

$$p_\mathrm{i,j}^* = \mathrm{rand}\left(\mathrm{da_j}, \mathrm{db_j}\right),\ \mathrm{if} p_\mathrm{i,j}^* \left\langle \mathrm{da_j or} p_\mathrm{i,j}^* \right\rangle \mathrm{db_j} \tag{28}$$

This method broadens the algorithm's search space and can increase population diversification in the global search process.

Table 1 Chaotic Maps (Khanduja and Bhushan 2021)

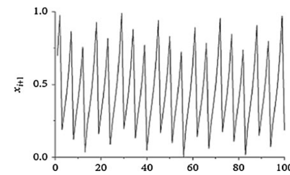
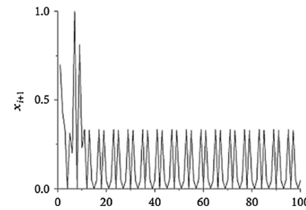
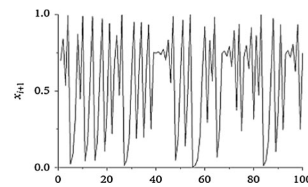
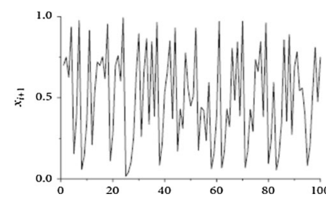
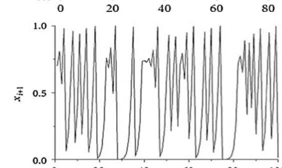
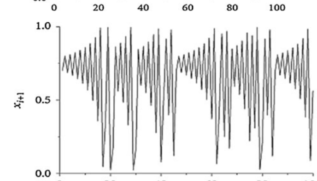
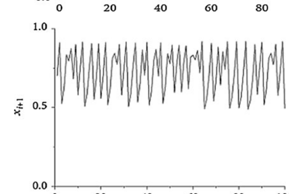
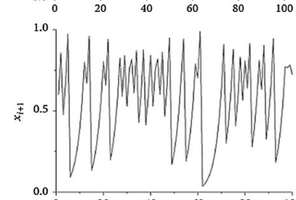
	Chaotic Map	Equation	Range	
C1	Circle	$\left\{x_{i+1} = \mathrm{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right)\sin(2\pi x_k), 1\right)\right\}$ $a = 0.5, b = 0.2$	(0, 1)	
C2	Gauss	$x_{i+1} = \begin{cases} 1, & x_i = 0 \\ \dfrac{1}{\mathrm{mod}(x_i, 1)}, & otherwise \end{cases}$	(0, 1)	
C3	Logistic	$x_{i+1} = ax_i(1 - x_i), a = 4$	(0, 1)	
C4	Piecewise	$\frac{x_k}{P}; 0 \le x_k < P$ $\frac{x_{k-} - P}{0.5 - P}; P \le x_k < 0.5$ $\frac{1 - P - x_{k-}}{0.5 - P}; 0.5 \le x_k < 1 - P$ $\frac{1 - x_{k-}}{P}; 1 - P \le x_k < 1$	(0, 1)	
C5	Sine	$\frac{a}{4}\sin(\pi x_k);\ 1 < a < 4$	(0, 1)	
C6	Singer	$x_{i+1} = \mu\left(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4\right)$ $\mu = 1.07$	(0, 1)	
C7	Sinusoidal	$x_{i+1} = ax_i^2\sin(\pi x_i), a = 2.3$	(0, 1)	
C8	Tent	$x_{i+1} = \begin{cases} \dfrac{x_i}{0.7} x_i < 0.7 \\ \dfrac{10(1 - x_i)}{3} x_i \ge 7 \end{cases}$	(0, 1)	

Table 2 Ziegler–Nichols tuned parameters for PID controller

Tuning method	K_p	I_i	T_d
Z-N closed loop	$0.6K_u$	$P_u/2$	$P_u/8$

3.6 CSMSEOBL Algorithm

A new hybrid algorithm CSMSEOBL (Chaotic State of Matter Search with Elite Opposition-Based learning) is used to tune the parameters of the PID Controller. To achieve the best results, this hybrid algorithm combines three algorithms: SMS, Chaotic Maps, and Elite Oppositional-Based Algorithm (Fig. 5). SMS's random variable "R" is computed using chaotic maps, and the addition of EOBL expands SMS's search field, increasing its global discovery capabilities.

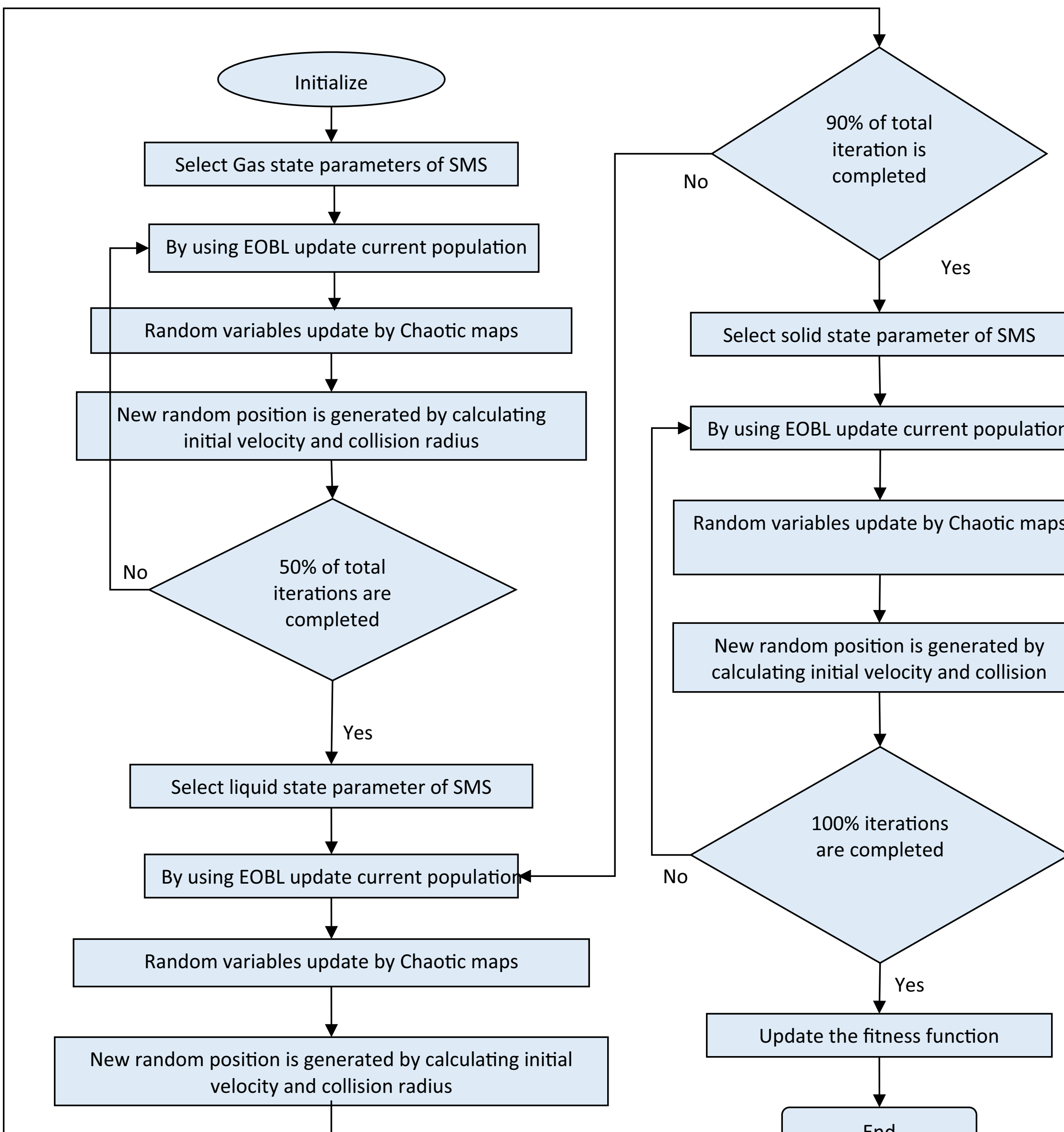


Fig. 5 Flow Chart of CSMSEOBL Algorithm

Pseudocode for CSMSEOBL

Input: Define the fitness function *f(x)*, with, X = ($x_1, x_2 \ldots x_P$)
Output: *x*is the optimal solution*

Step 1. Initialization: Set the SMS algorithm's gas state parameters, namely *α, β, ρ,* and *H*, as well as the complex boundary of the search space.

Step 2. While the stop condition is not met do

Step 3. EOBL methodology equations 26,27 and 28 are used to update the existing population.

Step 4. For every $x \epsilon P$ do

Step 5. Using chaotic maps, update all random variables.

Step 6. Using equations 13 and 16, calculate the initial velocity and collision radius for the gas state.

Step 7. Using the direction vector from equation 11, compute the new molecules.

Step 8. Using the collision operator from equation 15, solve the collision.

Step 9. Using the collision operator from equation 17, generate a new random location.

Step 10. If the total number of iterations done is less than half of the total number of iterations

Step 11. Return to the liquid state and repeat Steps 6, 7, 8, and 9 as necessary.

Step 12. Verify that the number of iterations performed is greater than 90% of the total number of iterations.

Step 13. Return to solid state and repeat steps 6, 7, 8, and 9 once more.

Step 14. If 100% iterations have been completed

Step 15. Update the fitness function $f(x)$

End if

End for

End while

4 PID Controller

Because of its reliability and simplicity of implementation, the PID controller is one of the popular controllers and is used in nearly every industrial control application (Namba and Yamamoto 1997) While there are many classical approaches for designing and tuning PID controller parameters (K_p, K_i, K_d) that are well understood and easy to apply, one of the key drawbacks of these classical techniques is that they require skill and practice to tune PID controllers using these techniques. In these methods, a starting point and fine-tuning of parameters by the hit-and-trial process are needed to achieve the desired efficiency . Due to its dynamic structure, metaheuristic strategies could be a reasonable option (Sabir and Khan 2014).

Inner loop values remain the same for the execution of the PID controller, while PID has been performed on an outer layer to incorporate the position of poles as long as there is a decline in terms of time constant. While manually calculating PID gain values, a large error is produced, particularly when operating under different parametric and external uncertainties. As a result, automatic PID gain tuning is needed, which is accomplished through various metaheuristic algorithms.

4.1 Classical PID Control

The Ziegler–Nichols (Ziegler and Nichols 1942) continuous cycling technique, also known as the ultimate gain method, was introduced in 1942 and is one of the most well-known closed-loop tuning techniques. The PID tuning values were established as a function of ultimate gain K_u and ultimate time P_u, and this tuning approach is often used by controller manufacturers and the process industry (Astrom 1995; Rao et al. 2014). Z-N tuned PID Controller parameters for closed-loop control systems are given in Table 2.

In this paper, parameters of PID Controller are tuned by using the classical method of PID Control, i.e., Ziegler–Nichols, and metaheuristic optimization methods like Particle swarm optimization (PSO), Stochastic Fractal Search (SFS), State of Matter search (SMS), and new hybrid algorithm Chaotic State of Matter Search with Elite Opposition-based Learning (CSMSEOBL).

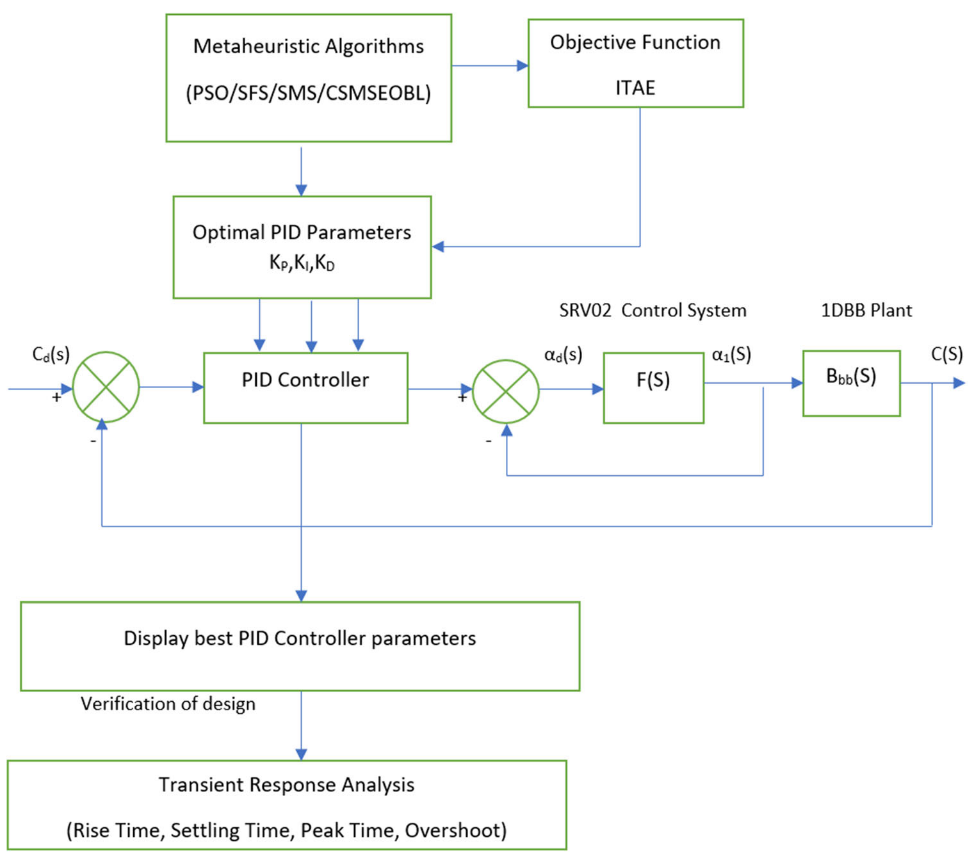
5 Simulation Results and Analysis

MATLAB/Simulink program is used to create the numerical simulation of the 2DoF ball balancer model described in Sect. 2. Since the plate on the two servo units is symmetrical, the action of one servo unit's controller affects the behavior of the other servo unit's controller. For the implementation of the PID controller (Fig. 6), inner loop values remain the same, whereas PID has been executed on an external shell to add the location of poles as far as there is decay in terms of time constant (Fig. 7).

Initially, the PID controller's values are determined using the traditional PID Controller as discussed in 4.1. Furthermore, metaheuristic algorithms such as PSO, SFS, SMS, and finally the hybridized EOBCSMS perform by calculating the difference between the expected and calculated ball position to optimize the PID controller values. For the time response study of a ball balancer system, robustness analysis is used since it investigates the system's performance at the beginning and steady-state. It offers information on the closed-loop system's relative stability and response time. Although the level of maximum overshoot can be linked-to relative stability, the settling and rise times demonstrate the system's reaction speed.

The problem is defined using an objective function or fitness function for any optimization process, such as convergence of a metaheuristic algorithm toward the global optima of PID adjusted parameters. The fitness function is firstly described by defining a controller based on the required requirements and restrictions. The

Fig. 6 Block diagram representing tuning of PID controller parameters using different metaheuristic algorithms for Ball balancer system



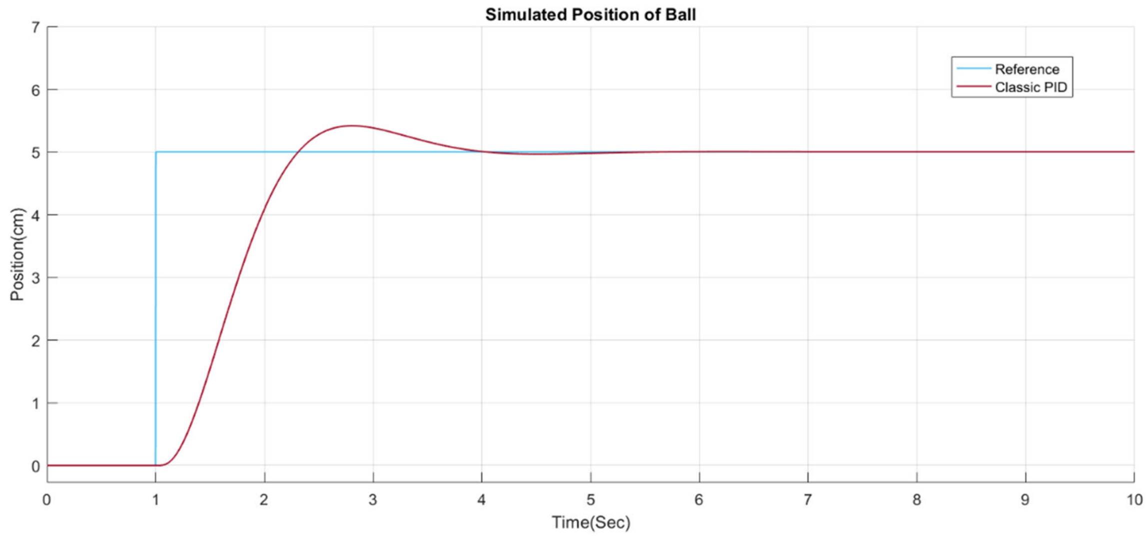


Fig. 7 Simulated position of ball for classical PID controller

Table 3 Parameter setting for different metaheuristic algorithms

Algorithm and parameters	Parameter value	Algorithm and parameters	Parameter value
PSO		SFS	
Population	50	Population	50
Iteration	25	Iteration	25
Weight Function	[0.2,0.9]	S.Diffusion	3
Acceleration constants	2	S.Walk	1
The dimension of search space	5	The dimension of search space	5
SMS		CSMSEOBL	
Vector Adjustment, ρ	1	Vector Adjustment, ρ	1
Beta	[0.8, 0.4, 0.1]	Beta	[0.8, 0.4, 0.1]
Alpha	[0.8, 0.2, 0]	Alpha	[0.8, 0.2, 0]
Threshold Probability, H	[0.9, 0.2, 0]	Threshold Probability, H	[0.9, 0.2, 0]
Phase Percent	[0.5, 0.1, -0.1]	Phase Percent	[0.5, 0.1, -0.1]
Adjustment Parameters	[0.85 0.35 0.05]	Adjustment Parameters	[0.85 0.35 0.05]
Iteration	25	Iteration	25

controller parameter settings are altered by configuring the objective function. Typically, four types of performance requirements are examined in the domain of Controller design procedure. They are the integral of absolute error (IAE), the integral of squared error (ISE), the integral of time multiplied squared error (ITSE), and the integral of time multiplied absolute error (ITAE). Integral time absolute error (ITAE) is used as the fitness or objective function in this study as it is the most stringent controller setting criteria, avoiding peaks and giving controllers a higher load disturbance rejection and lessening system overshoot while maintaining network performance. Parameter setting for different metaheuristic algorithms is shown in Table 3.

$$\text{Objective function} \quad J_{\text{ITAE}} = \int_0^T t|e(t)|dt \tag{29}$$

Figures 8, 9, 10, respectively, show the effects of the ball position, servo angle, and voltage of the ball balancer mechanism for the cation of the hybridized CSMSEOBL-PID, PSO-PID, SFS-PID, SMS-PID, and the classic PID controller. The transient reactions of Classic PID (Table 4), PSO-PID, SFS-PID, SMS-PID, and CSMSEOBL-PID controllers to a reference position are shown in Fig. 8. Table 5 also includes numerical comparisons in terms of maximum overshoot, rising time (10 percent–90 percent), and settling time (2 percent band). All of the performance criteria, such as overshoot, rising time, and settling time,

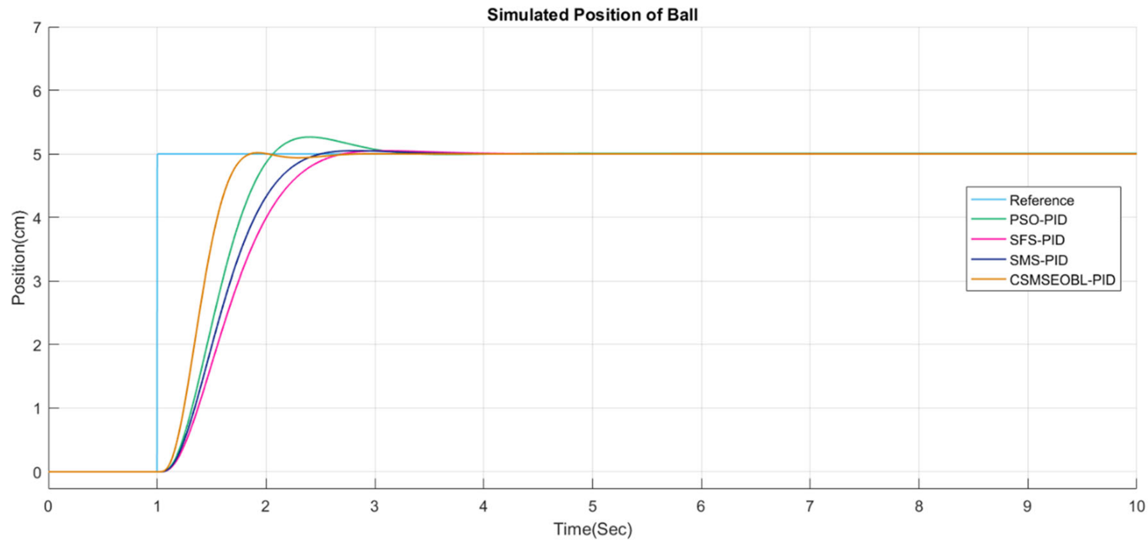


Fig. 8 Simulated position of ball for different metaheuristic algorithms

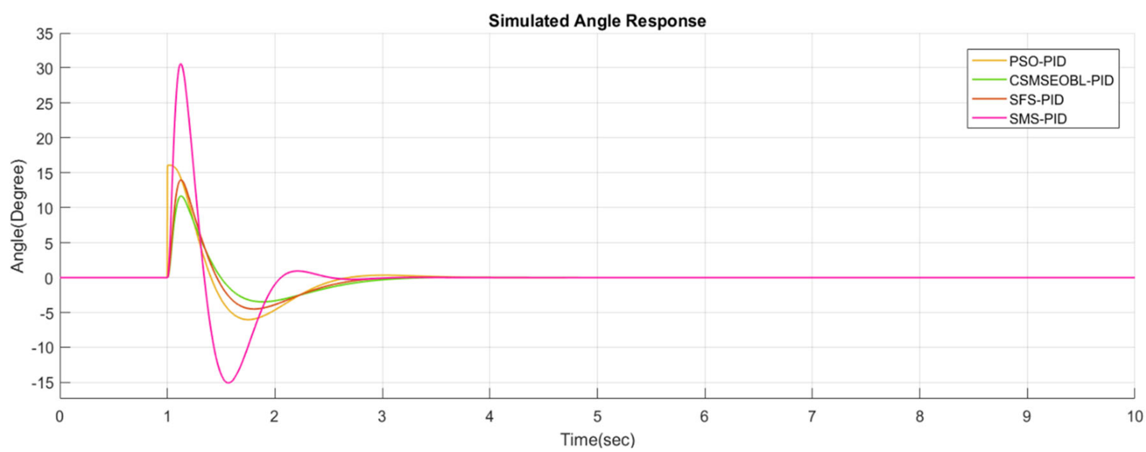


Fig. 9 Simulated Servo angle response of ball for different metaheuristic algorithms

have attained the lowest values with the suggested CSMSEOBL-PID controller for the Ball balancer system, according to the comparative transient response study findings in Fig. 8 and Table 5. These findings support the CSMSEOBL algorithm's significance in terms of strong exploration and exploitation search capabilities, as well as the system's transient reaction. In Fig. 8, the controller's efficacy is shown by the smallest difference between the original and final positions. The hybridized CSMSEOBL-PID in this case has a minimum final position and achieves the optimal benefit in a short amount of time. The simulated position of Ball for different chaotic maps discussed in Sect. 3.4 is shown in Fig. 11, and this shows that the CSMSEOBL algorithm gives the best results for piecewise Chaotic map. By employing hybridized CSMSEOBL-PID, stabilization angle is maintained between − 2.5 and 12 degree whereas SMS-PID have this range varying from − 15 to 30°. Figure 9 demonstrates that the hybridized CSMSEOBL- PID operates smoothly even when external disturbances impair the ball's movement. Additionally, the voltages in Fig. 10 show hybridized CSMSEOBL-PID effects on plate angle stabilization and ball balance on the plate.

Table 5 shows that the classical PID controller (Ziegler-tuned Ball balancer system has maximum overshoot and large Integral Time absolute error (ITAE) along with a large settling time. Tuning of PID with PSO results in lesser settling time, lesser overshoot, and less ITAE as compared to the classical controller. Further parameters of PID are tuned by using SFS, and SMS algorithms. Results are further improved with SFS and SMS algorithms, but the proposed approach, i.e., CSMSEOBL-based tuning of PID

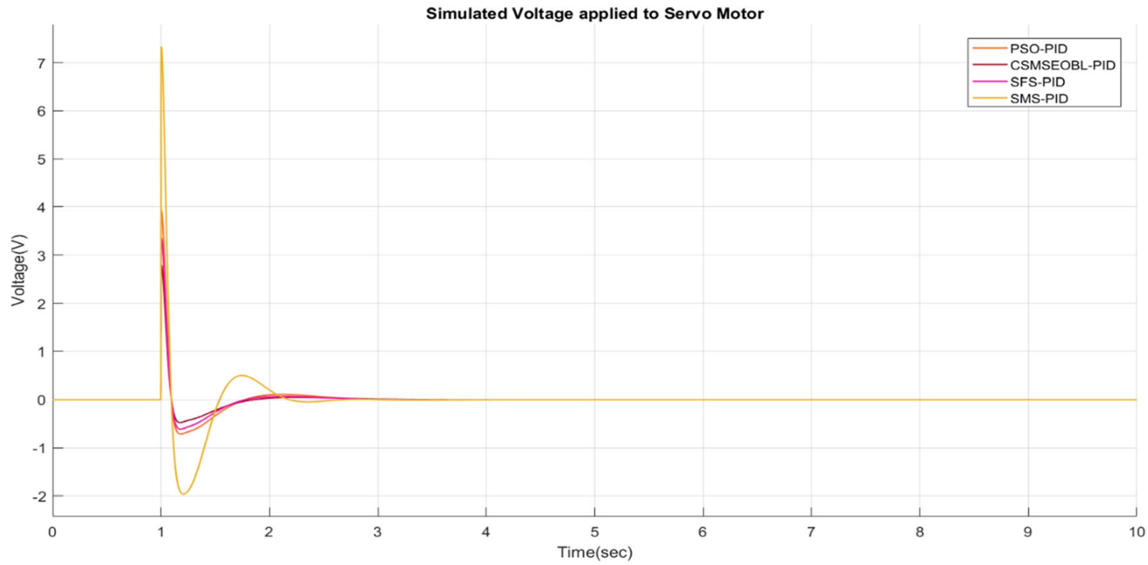


Fig. 10 The simulated voltage applied to servo motor for different metaheuristic algorithms

Table 4 Comparison of step response characteristics for position control of ball balancer system by classical PID Controller

Name of the algorithm	PID controller parameters			Step response characteristics					
	Kp	*Ki*	*Kd*	Rise time	Settling time	Max overshoot	ISE	IAE	ITAE
Classical PID	3.45	0.0012	2.11	0.827	3.65	8.152	0.001291	0.03873	0.06368

Table 5 Comparison of step response characteristics for position control of ball balancer system

Name of the algorithm	PID controller parameters			Step response characteristics			
	Kp	*Ki*	*Kd*	Rise TIME	Settling time	Max overshoot	Fitness function ITAE
Classical PID	3.45	0.0012	2.11	0.827	3.65	8.152	0.06368
PSO-PID	5.61	0.0167	2.86	0.651	2.9	4.737	0.05474
SFS-PID	10.865	4.38E-06	5.587	0.897	2.72	− 0.27	0.03675
SMS-PID	7.9	1.2E-07	3.7431	0.57	**2.1**	0.496	0.03081
CSMSEOBL-PID	**11.032**	**1.02E-06**	**4.386**	**0.447**	2.2	**0**	**0.02538**

controller results in the finest response in terms of least overshoot and least ITAE as shown in Fig. 11 (Figs. 12, 13).

A total of 150 optimization trials are carried out to assess the efficacy of all algorithms' optimization processes. The population size (50) and the maximum number of iterations (25) are maintained constant to provide a fair comparison. The maximum run of the iteration serves as a stopping condition for the optimization process. The maximum number of iterations is 25, while the total number of independent trials and other parameters remain same. Figure 14 illustrates the findings. To provide a fair comparison of the all methods, the simulations are stopped after 25 iterations. A single iteration of the combat approach consists of transition through all states of matter, random variable selection through chaotic maps and opposite population calculation operation (Table 6).

Tracking Response of ball balancer system is represented in Fig. 12, and it reveals that the proposed CSMSEOBL methodology results in better tracking as

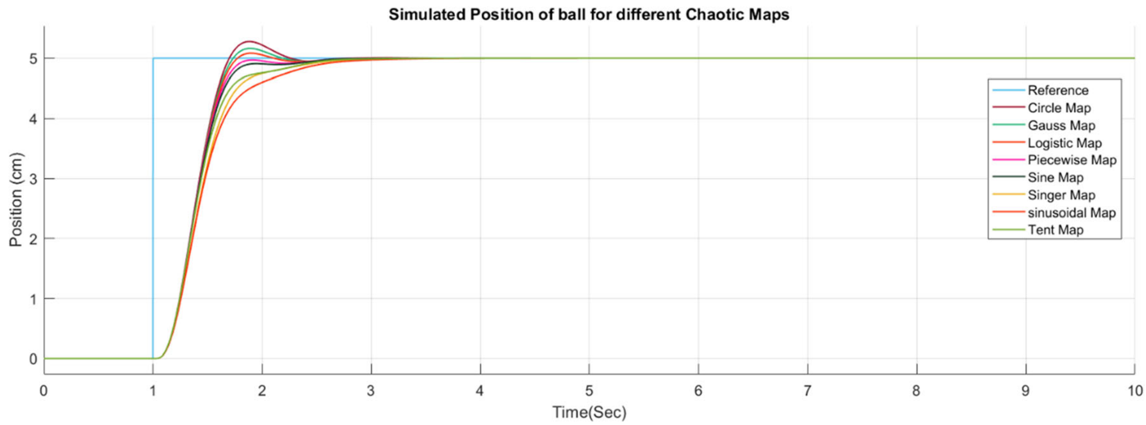


Fig. 11 Simulated position of the ball with CSMSEOBL algorithm for a variety of Chaotic maps

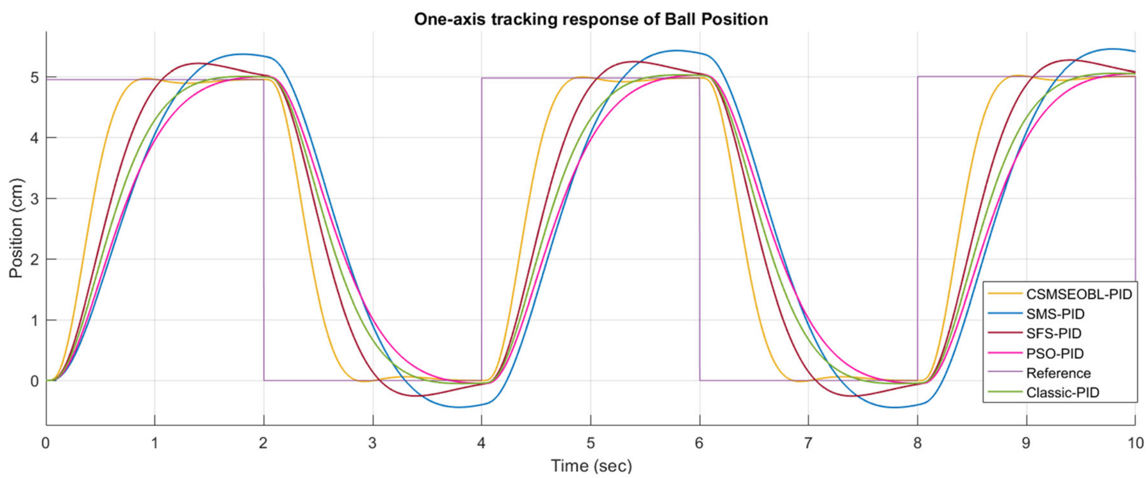


Fig. 12 Tracking response of ball balancer system

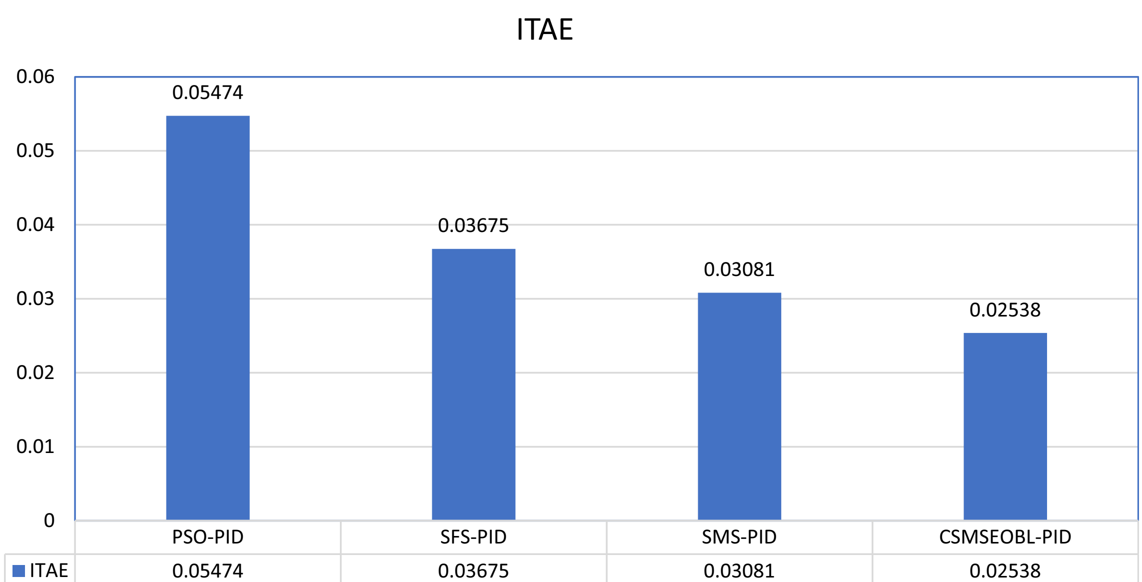


Fig. 13 Variation of ITAE for different Metaheuristic Algorithms

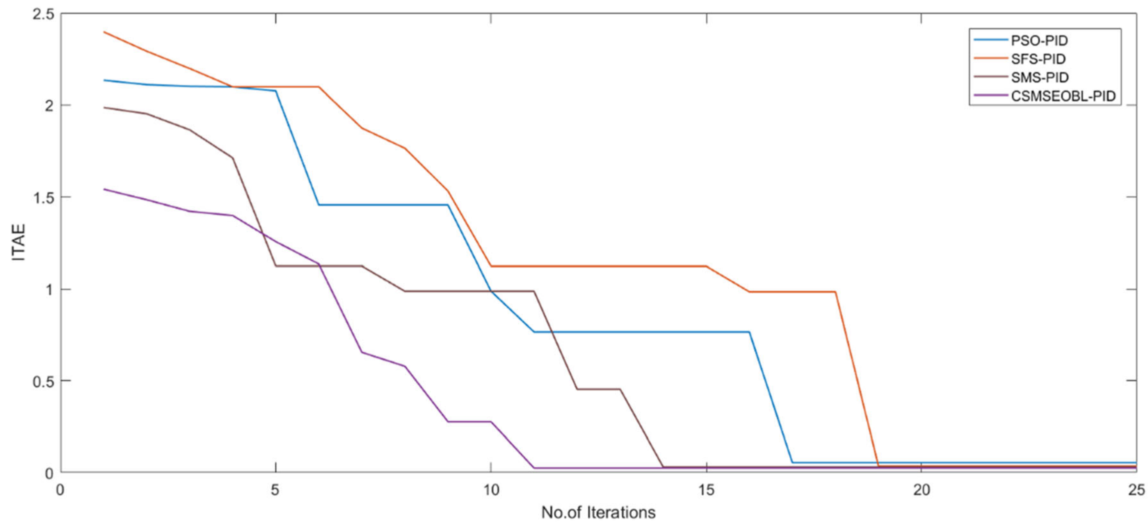


Fig. 14 ITAE variation for different metaheuristic algorithms

Table 6 Statistical analysis for different metaheuristic algorithms

Algorithm	St. deviation in ITAE
PSO-PID	0.01354
SFS-PID-DOF-PID	0.01699
SMS-PID	0.009854
CSMSEOBL-PID	0.00806

compared to other metaheuristic algorithms along with this tracking response ITAE convergence is shown in Fig. 14 and it shows that CSMSEOBL gives best convergence as compared to SMS, SFS and PSO algorithm for same number of iterations. Standard deviation of ITAE is also shown in Fig. 6 to give a good statistical analysis and CSMSEOBL have least standard deviation as compared to other metaheuristic algorithms. The simulation findings show that using the CSMSEOBL-PID scheme with ITAE as the goal function produces no overshoot, and that other characteristics like settling and rising time are comparable to existing methods.

6 Conclusion

For the first time, the CSMSEOBL technique is suggested for the design of a PID controller in a Ball balancer system with ITAE as an objective function. An improved Oppositional-Based Chaotic State of Matter Search Algorithm is used to tune the parameters of Proportional Integral Derivative control to achieve position and self-balancing control of a two-degree-of-freedom balancer device. The findings of simulations show that the evolved approach improves efficiency significantly within the traditional system.

A graphical and numerical comparison of the CSMSEOBL-PID methodology to other existing methods is also given. For ITAE objective function, the CSMSEOBL-PID method seems to have greater outcomes than the current approaches in the literature. The best settling time, rise time, and maximum overshoot are obtained by this methodology.

When validated for simulation, the proposed controller exhibited adaptability and adequate control efficiency on the ball balancer system. The suggested tuning methodology is efficient and outperforms the comparable approaches, and it is a viable method for being applied to other real-world engineering problems like automatic voltage regulator device architecture, DC motor speed control, concentration and temperature control of the reactor, and so on, which may be the subject of future research. Because optimization issues in practical uses are caused by ambiguity, creating a robust version of the CSMSEOBL that could also offer resilient and risk-averse solutions is much important.

Declarations

Conflict of interest The Author(s) Neha Khanduja and Prof. Bharat Bhushan declare(s) that there is no conflict of interest.

References

Al-Attar Mohamed AMH, El-Gaafary AAM, Mohamed YS (2016) Multi-objective states of matter search algorithm for TCSCbased smart controller2106 design. Electr Power Syst Res 140:874–885

Ali RM, Awad MZ, Suganthan NH, Duwairi PN, Reynolds RG (2016) A novel hybrid Cultural Algorithms framework with trajectory-based search for global numerical optimization. Inf Sci 334:219–234

Ali AT, Ahmed AM, Almahdi HA, Taha OA, Naseraldeen A (2017) Design and implementation of ball and beam system using PID controller. Acis-3-1-1 3(1):1–4

Ali HI, Jassim HM, Hasan AF (2019) Optimal Nonlinear Model Reference Controller Design for Ball and Plate System. Arabian J Sci Eng 44(8):6757–6768

Ali ET, Abdullah S, Amir M, Adeel EM (2019) Stability control of ball and beam system using heuristic computation based PI-D and PI-PD controller. Tech J 24(01):21–29

Aranda J, Chaos D, Dormido-Canto S, Muñoz R, Díaz JM (2006) Benchmark control problems for a non-linear underactuated hovercraft: a simulation laboratory for control testing. IFAC Proc 7:463–488

Astrom THKJ (1995) PID controllers:theory, design, and tuning, NC, 1995. Instrum. Soc. Am. Res. Triangle Park.

Awad RG, Ali NH, Suganthan MZ, Reynolds PN (2017) A hybridization of cultural algorithm and differential evolution for numerical optimization. Inf Sci 378:215–241

Bang H, Lee YS (2019) Embedded model predictive control for enhancing tracking performance of a ball-and-plate system. IEEE Access 7:39652–39659

Başçi A, Kaan CA (2017) Position control of a ball & beam experimental setup based on sliding mode controller. Int J Appl Math Electron Comput 1:29–35

Bhatt R, Parmar G, Gupta R, Sikander A (2019) Application of stochastic fractal search in approximation and control of LTI systems. Microsyst Technol 25(1):105–114. https://doi.org/10.1007/s00542-018-3939-6

Bolívar-Vincenty CG, Beauchamp-Báez G (2014) Modelling the ball-and-beam system from newtonian mechanics and from lagrange methods. In: Latin American 22–24, Caribbean conference for engineering and technology, pp. 22–24

Burghardt A, Giergiel J (2011) Modelling and control of a underactuated sphere and beam system. Commun Nonlinear Sci Numer Simul 16(5):2350–2354

Çelik E (2018) Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system. Neural Comput Appl 30(6):1991–2002. https://doi.org/10.1007/s00521-017-3335-7

Choukchou-Braham A, Cherki B, Djemaï M, Busawon K (2014) Analysis and control of underactuated mechanical systems

Cuevas E, Echavarría A, Ramírez-Ortegón MA (2014) An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Appl Intell 40(2):256–272. https://doi.org/10.1007/s10489-013-0458-0

Das AKS, Abraham A (2008) Particle swarm optimization and differential evolution algorithm. Tech. anal. appl. hybrid. perspect. adv. comput. intell. ind. syst., pp 1–38

Das PN, Suganthan S (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31

Dong X, Zhao Y, Xu Y, Zhang Z (2011) Design of PSO fuzzy neural network control for ball and plate system. Int J Innov Comput Inf Control 7(12):7091–7103

Dubey HM, Pandit M, Panigrahi BK (2018) An overview and comparative analysis of recent bio-inspired optimization techniques for wind integrated multi-objective power dispatch. Swarm Evol Comput 38:12–34

Dušek F, Honc D, Sharma KR (2017) Modelling of ball and plate system based on first principle model and optimal control. In: 21st international conference on process control, pp. 216–221

Eberhart RC, Shi Y (2001) Tracking and optimizing dynamic systems with particle swarms. Evolutionary computation, pp 94–100

Fan X, Zhang N, Teng S (2004) Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme. Fuzzy Sets Syst 144(2):297–312

Farah A, Belazi A (2018) A novel chaotic Jaya algorithm for unconstrained numerical optimization. Nonlinear Dyn 93(3):1451–1480. https://doi.org/10.1007/s11071-018-4271-5

Feng H, Yin CB, Weng WW, Ma W, Zhou JJ, Jia WH (2108) Robotic excavator trajectory control using an improved GA based PID controller. Mech Syst Signal Process 105:153–168

Gong D, Lu L, Li M (2009) Robot path planning in uncertain environments based on particle swarm optimization. In: IEEE congress on evolutionary computation, pp 2127–2134

Han K, Tian Y, Kong Y, Li J, Zhang Y (2012) Tracking control of ball and plate system using a improved PSO on-line training PID neural network. In: 2012 IEEE international conference on mechatronics and automation, pp 2297–2302

Howe III JM, O'Brien Jr RT (2008) Experimental investiga_tion of optimal PID controller design using model reduction techniques. In: Proceedings of the 40th southeastern symposium on system theory (SSST '08). New Orleans, La USA, pp 31–36. IEEE

Isa AI, Hamza MF, Zimit AY, Adamu JK (2018) Modelling and fuzzy control of ball and beam system. ICASTECH, pp 1–6

Jayachitra A, Vinodha R (2015) COMPARATIVE STUDY AND IMPLEMENTATION OF MULTI-OBJECTIVE PSO ALGORITHM USING DIFFERENT INERTIA WEIGHT TECHNIQUES FOR OPTIMAL CONTROL OF A CSTR PROCESS. ARPN J Eng Appl Sci 10(22):10395–10404

Jimenez S, Yu W (2007) Table synchronization control for two ball and beam systems. In: Proceedings of the 4th international conference on electrical and electronics engineering, ICEEE, pp 290–293

Joorabian E, Afzalan M (2014) Optimal power flow under The, both normal and contingent operation conditions using Nelder-Mead, hybrid fuzzy particle swarm optimisation and algorithm (HFPSO-NM). Appl Soft Comput 14:623–633

Jordehi AR (2014) A chaotic-based big bang-big crunch algorithm for solving global optimisation problems. Neural Comput Appl 25(6):1329–1335

Kennedy R, Eberhart J (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, pp 1942–1948

Ker CC, Lin CE, Wang RT (2007) Tracking and balance control of ball and plate system. J Chin Inst Eng 30(3):459–470

Keshmiri M, Jahromi AF, Mohebbi A, Hadi Amoozgar M, Xie WF (2012) Modeling and control of ball and beam system using model based and non-model based control approaches. Int J Smart Sens Intell Syst 5(1):14–35

Khalilpourazari S, Naderi B, Khalilpourazary S (2020) Multi-Objective Stochastic Fractal Search: a powerful algorithm for solving complex multi-objective optimization problems. Soft Comput 24(4):3037–3066

Khanduja B, Bhushan N (2021) Optimal design of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm. Sādhanā 46(104):1–12

Khodadadi H, Ghadiri H (2018) Self-tuning PID controller design using fuzzy logic for half car active suspension system. Int J Dyn Control 6(1):224–232. https://doi.org/10.1007/s40435-016-0291-5

Lagunes ML, Castillo O, Valdez F, Soria J, Melin P (2021) A new approach for dynamic stochastic fractal search with fuzzy logic for parameter adaptation. Fractal Fract 5(2):33

Lahoty G, Parmar P (2013) A comparative study of tuning of PID controller using evolutionary algorithms. Int J Emerg Technol Adv Eng 3(1):640–644

Lawrence B (2020) Tuning of a PID controller for optimal performance of ball and beam system. Int J Eng Res V9(04):1–5. https://doi.org/10.17577/ijertv9is040009

Li X (2020) Robot target localization and interactive multi-mode motion trajectory tracking based on adaptive iterative learning. J Ambient Intell Humaniz Comput

Li X, Yu W (2012) Synchronization of ball and beam systems with neural compensation. Int J Control Autom Syst 8(3):491–496

Maalini PM, Prabhakar G, Selvaperumal S (2016) Modelling and control of ball and beam system using PID controller. In: International conference on advanced communication control and computing technologies, pp. 32-–326

Mahmoodabadi MJ, Mottaghi MB, Mahmodinejad A (2016) Optimum design of fuzzy controllers for nonlinear systems using multi-objective particle swarm optimization. J Vib Control 22(3):769–783

Majumdar K, Mitra M, Nishimura T (2000) Optimization and chaos, vol 11. Springer, New York

Mandal B, Si T (2015) Opposition based Particle Swarm Optimization with exploration and exploitation through gbest. In: 2015 int. conf. adv. comput. commun. informatics, ICACCI 2015, pp 245–250. https://doi.org/10.1109/ICACCI.2015.7275616

Márton L, Hodel AS, Lantos B, Hung JY (2008) Underactuated robot control: comparing LQR, subspace stabilization, and combined error metric approaches. IEEE Trans Ind Electron 55(10):3724–3730

Meenakshipriya B, Naveen MC, Kalpana MK (2018) PSO TUNING OF A CDM BASED PID CONTROLLER FOR BALL AND BEAM SYSTEM. Int J Pure Appl Math 120(6):10905–10927

Mellal MA, Zio E (2016) A penalty guided stochastic fractal search approach for system reliability optimization. Reliabil Eng Syst Safety 152:213–227

Mochizuki S, Ichihara H (2013) Generalized Kalman-Yakubovich-Popov lemma based I-pd controller design for ball and plate system. J Appl Math 2013:9

Muskinja N, Tovornik B, Donlagic D (1997) How to design a discrete supervisory controller for real-time fuzzy control systems. IEEE Trans Fuzzy Syst 5(2):161–166

Muškinja N, Rižnar M (2015) Optimized PID position control of a nonlinear system based on correlating the velocity with position error. Math Probl Eng. https://doi.org/10.1155/2015/796057

Nagaraj BRB (2008) Tuning algorithms for PID controller using soft computing techniques. Int J Comput Sci Netw 8(4)

Namba MKR, Yamamoto T (1997) Robust PID controller and its application. In: IEEE international conference on systems, man, and cybernetics, pp 3636–3641

Nasrabadi MS, Sharafi Y, Tayari M (2016) A parallel Grey Wolf Optimizer combined with Opposition based learning. In: Ist conference on swarm intelligence and evolutionary computation (CSIEC), pp 18–23

Nowopolski K (2013) Ball-and-beam laboratory system controlled by SimuLink model through dedicated microcontrolled-Matlab data exchange protocol. Comput Appl Electr Eng 11:310–320

Oh SK, Jang HJ, Pedrycz W (2011) Optimized fuzzy PD cascade controller: a comparative analysis and design. Simul Model Pract Theory 19(1):181–195

Oh SK, Jang HJ, Pedrycz W (2011) Optimized fuzzy PD cas_cade controller: a comparative analysis and design. Simul Model Pract Theory 19(1):181–195

Pinagapani AK, Mani G, Chandran KR, Pandian K (2018) Composite disturbance rejection control for ball balancer system. Procedia Comput Sci 133:124–313

Rahman TA, As'arry A, Jalil NA (2017) Active vibration control of a flexible beam structure using chaotic fractal search algorithm. Procedia Eng 170:299–306

Rahmat MF, Wahid H, Wahab NA (2010) Application of intelligent controller in a ball and beam control system. Int J Smart Sens Intell Syst 3(1):45–60

Rao PVGK, Subramanyam MV, Satyaprasad K (2014) Study on PID controller design and performance based on tuning techniques. In: 2014 int. conf. control. instrumentation, commun. comput. technol. ICCICCT, pp 1411–1417. https://doi.org/10.1109/ICCICCT.2014.6993183

Ribeiro JM, Santos MF, Carmo MJ (2107) Comparison of PID controller tuning methods: analytical/classical techniques versus optimization algorithms. In: 2017 18th international Carpathian control conference (ICCC) 2017, pp. 533–538. IEEE

Rivas-Echeverria F, Rios-Bolivar A (2001) Neural network-based auto-tuning for PID controllers. Neural Netw World 11(3):277–284

Roy IHP, Kar B (2014) Trajectroy control of a ball in a ball and plate system using cascaded PD controllers tuned by PSO. In: Proceedings of fourth international conference on soft computing for problem solving, vol 2, p 53

Rudra S, Barai RK, Maitra M (2017) Block backstepping control of the underactuated mechanical systems

Saad M, Khalallah M (2017) Design and implementation of an embedded Ball-Beam controller using PID algorithm. Univ J Control Autom 5(4):63–70

Saad MS, Jamaluddin H, Darus IZ (2012) Implementation of PID controller tuning using differential evolution and genetic algorithms. Int J Innov Comput Inf Control 8(11):7761–7779

Sabir MM, Khan JA (2014) Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques. Adv Artif Neural Syst 2014:1–8. https://doi.org/10.1155/2014/126317

Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm. Knowl Based Syst 75:1–18

Sayed GI, Darwish A, Hassanien AE (2018) A new chaotic multiverse optimization algorithm for solving engineering optimization problems. J Exp Theor Artif Intell 3079:1–25. https://doi.org/10.1080/0952813X.2018.1430858

Shekhawat S, Saxena A (2020) Development and applications of an intelligent crow search algorithm based on opposition based learning. ISA Trans 99:210–230. https://doi.org/10.1016/j.isatra.2019.09.004

Singh R, Bhushan B (2021) Improved ant colony optimization for achieving self-balancing and position control for balancer systems. J Ambient Intell Humaniz Comput 1–18

Spong MW (1994) Partial feedback linearization of underactuated mechanical systems. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS'94), pp 314–321

Su S, Su Y, Xu M (2014) Comparisons of firefly algorithm with chaotic maps. Comput Model New Technol 18(2):326–332

Sun SQ, Li L (2012) The study of ball and plate system based on nonlinear PID. Appl Mech Mater 187:134–137

Sutharsan S et al (2012) An optimization-based parallel particle filter for multitarget tracking. IEEE Trans Aerosp Electron Syst 48(2):1601–1618

Thangaraj PBR, Pant M, Abraham A (2011) Particle swarm optimization: Hybridization perspectives and experimental illustrations. Appl Math Comput 217:5208–5226

Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computation intelligence on modeling control automation and international conference on intelligent agents, web technologies, pp 695–701

Tyagi T, Dubey HM, Pandit M (2016) Multi-objective optimal dispatch solution of solar-wind-thermal system using improved stochastic fractal search algorithm. Int J Inf Technol Comput Sci (IJITCS) 8(11):61–73

Valdivia-Gonzalez A, Zaldívar D, Fausto F, Camarena O, Cuevas E, Perez-Cisneros M (2017) A states of matter search-based approach for solving the problem of intelligent power allocation in plug-in hybrid electric vehicles. Energies 10(1):92

Wang C, Huang L (2018) A novel Elite Opposition-based Jaya algorithm for parameter estimation of photovoltaic cell models. Optik (stuttg) 155:351–356

Wang L, Zhong Y (2015) Cuckoo search algorithm with chaotic maps. Math Probl Eng 2015:1–14. https://doi.org/10.1155/2015/715635

Wang Z, Rahnamay H, Wu S (2013) Parallel differential evolution with self-adapting control parameters and Parallel, generalized opposition-based learning for solving high-dimensional optimization problems. J Distrib Comput 73:62–73

Wang G, Guo L, Gandomi AH, Hao G, Wang H (2014) Chaotic Krill Herd algorithm. Inf Sci (ny). https://doi.org/10.1016/j.ins.2014.02.123

Wang D, Tan D, Liu L (2017) Particle swarm optimization algorithm: an overview. Soft Comput 22(2):387–408

Xin J, Chen G, Hai Y (2009) A particle swarm optimizer with multistage linearly-decreasing inertia weight. Computational sciences and optimization, pp 505–508

Yan X, Wu Q, Liu H (2013) An improved particle swarm optimization algorithm and its application. Int J Comput Sci 10:316–324

Yu W (2009) Nonlinear PD regulation for ball and beam system. Int J Electr Eng Educ 46(1):59–73

Zhao Y, Luo R, Zhou Q (2017) Elite opposition-based social spider optimization algorithm for global function optimization. Algorithms 10(1):9. https://doi.org/10.3390/a10010009

Zhou Y, Zhou Y, Luo Q, Qiao S (2015) Drift Operator for states of matter search algorithm. Advanced intelligent computing theories and applications, vol 9227

Zhou Q, Wang Y, Luo R (2016) Elite opposition-based flower pollination algorithm. Neurocomputing 188:294–310

Ziegler JG, Nichols NB (1942) Optimum settings for automatic controllers. Trans ASME 64:759–768