METAHEURISTIC ALGORITHMS AND THEIR APPLICATIONS TO NONLINEAR SYSTEMS

A Thesis submitted to the

Delhi Technological University

For the award of the Doctor of Philosophy

In

Electrical Engineering

By

NEHA KHANDUJA

(2K15/Ph.D./EE/02)

Under the Supervision of

PROF. BHARAT BHUSHAN

Professor

Electrical Engineering Department

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI-110042, INDIA

JUNE 2022

Certificate

This is to certify that the thesis entitled." **Metaheuristic algorithms and Their Applications to Nonlinear Systems**" submitted for the award of the degree of Doctor of Philosophy is original to the best of my knowledge. The work was carried out by **Ms. Neha Khanduja** under my guidance and has not been submitted in parts or full to this or any other University for the award of any degree or diploma. All the assistance and help received during the course of study has been duly acknowledged.

Supervisor

26/11/2021

Prof. Bharat Bhushan
Department of Electrical Engineering
Delhi Technological University
Delhi-110042, India

Acknowledgment

Writing this thesis has been fascinating and extremely rewarding. First of all, I am thankful to God, who has blessed me with the strength to acquire knowledge and learning.

This report in the present profile is a result of continuous enthusiasm and efforts put by me, to make it of scholarly level. I would like to take this opportunity to show my gratitude towards various people from the university who had helped in the cause.

First of all, I would like to express my sincere gratitude to my honourable supervisor **Prof. Bharat Bhushan**, for his constant guidance and persistent encouragement throughout the development of this research work. It is a matter of immense pride for me to have worked under him, who has such a great zeal toward research and teaching. It in itself was a source of inspiration for me to complete the research work with great enthusiasm, energy, and determination. I would like to extend my special thanks to him for dedicating his valuable time whenever I needed to discuss the research work. I would also like to thank the reviewers of our papers for their valuable suggestions and comments which motivated me to look into areas which we never thought of venturing into.

I would also like to thank **Prof. Uma Nangia**, Head of the Department, and **Prof. Madhusudan Singh**, Chairman, DRC Electrical Engineering Department, Delhi Technological University, New Delhi for being a visionary and supporting the concept of the research work.

I am also very thankful to all the faculty members of the department for sharing their knowledge, to aid in the concern.

I would also thank to all of them who have been helpful and were associated with me directly and indirectly throughout the work. I would like to thank my friend **Ms. Simmi Sharma** for her technical and moral support throughout the project work.

I would like to extend my thanks to **Mr. Ajit Sharma** for his moral support.

Finally, no words to thank my parents, husband, kids, mother-in-law, and father-in-law for their faith, encouragement, understanding and unfailing support.

Neha Khanduja

Abstract

Researchers working on problems in engineering, computer science, biology, and the physical sciences are developing advanced mathematical methods for control. Technological advances have had a major impact on the use of new analytical methods for dealing with nonlinear problems. One of the most challenging parts of control theory is tuning the parameters of nonlinear systems for an optimum solution. In the past, metaheuristic methods were tried to address this problem. They have proved to be useful when dealing with complex systems. Metaheuristic optimization techniques, unlike deterministic algorithms, excel at addressing problems with uncertain search spaces. Optimization-based control is now favoured over conventional or intelligent control, and because of these aspects, a hybrid CSMSEOBL technique is suggested to accomplish this. Due to their ability to overcome single algorithm limitations without compromising their strength, hybrid techniques outperform stand-alone alternatives. It's a tweaked version of the SMS algorithm (state of matter search) in which Chaotic Maps and Elite opposition-based learning (EOBL) are combined with SMS to improve the system's efficiency and effectiveness. The SMS algorithm's fundamental concept is at the heart of the thermal energy motion system. The method is broken down into three states of matter: solid, liquid, and gas, each with its diversification-intensification ratio. The method begins with a gas state and progresses to a solid-state by changing the diversification-intensification ratio. The proposed approach is compared to other optimization algorithms on unimodal, multimodal, and fixed-dimension multimodal benchmark functions to demonstrate its efficacy.

Proportional-integral-derivative (PID) controllers are the most commonly used controllers in process industries due to their accessibility, efficacy, and durability. The system becomes unstable when process parameters change and disturbances occur. Because of the increasing complexity of plant operations, adjusting the parameters of a PID controller to avoid failures and excellent transient performance has become more difficult in recent years. Optimal adjustment of PID parameters is now a difficult job for control engineers. PID Controller and its variants like FOPID and 2-DOF-PID controllers are used for controlling nonlinear control problems.

To assess the performance of the developed hybrid metaheuristic algorithm simulation studies are carried on fifteen benchmark functions and three nonlinear control systems Continuously stirred tank reactor, Ball balancer, and D.C. motor. A comparative study in terms of setpoint response analysis, convergence analysis, statistical analysis, and trajectory analysis with other recent existing metaheuristic algorithms are also presented to prove the superiority of the proposed algorithm.

Contents

List of Figures

Figure No.	Title of Figure	Page No.

Figure No.	Title of Figure	Page No.

List of Tables

Acronyms

SMS	State of Matter Search
EOBL	Elite Opposition based Learning
DE	Differential Evolution
DOF	Degree of Freedom
MHA	Metaheuristic algorithm
ZN	Ziegler Nichols
GA	Genetic Algorithm
KH	Krill Herd
MBO	Monarch Butterfly Optimization
SA	Simulated Annealing
CS	Cuckoo Search
SFS	Stochastic Fractal Search
WWO	Water wave optimization
PSO	Particle Swarm Optimization
LQR	Linear Quadratic Regulator
PID	Proportional-Integral-Derivative
FOPID	Fractional order Proportional-Integral-Derivative
SIMC	Skogestad Internal Model Control
GKYPL	Generalized Kalman—Yanukovych—Popov lemma
DE	Differential Evolution
ICA	Imperialist Competitive Algorithm
BLDC	Brushless Direct Current
BFA	Bacteria Foraging Algorithm

CSTR	Continuously Stirred Tank Reactor
IMC	Internal Model Control
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control

Chapter 1

Introduction

1.1 Overview

Advanced mathematical techniques for control are being developed by researchers working on issues in engineering, computer science, biology and the physical sciences. The application of novel analytical approaches for tackling nonlinear issues has been significantly influenced by technological advancements [1]. The state may not be entirely quantifiable in most situations involving nonlinear control systems, making complicated control engineering problems difficult to address. The employment of a variety of distinct models and ideas, a lack of parameter standardization, a lack of suitable control approaches, external disruptions and the greater level of nonlinearity of the equations that drive processes are all important challenges in the field of control technology. Another difficulty is the lack of understanding of the critical variables since the system's states might significantly affect the nature of the control design stage, allowing for excellent performance. As a result, enhanced forecasting, control and optimization approaches are required to ensure optimal nonlinear system performance. Understanding the system's control needs necessitates knowledge of the system; nevertheless, nonlinearities are frequently so complicated that control design for system performance is challenging [2]. New control techniques have developed over time to maintain optimal system performance that prevents interruptions, pauses, and design flaws.

Tuning the parameters of nonlinear systems for an optimal solution is one of the most difficult aspects of control theory. Metaheuristic strategies have been used to solve this challenge in the past. When dealing with complicated systems, they have proven to be beneficial. Unlike deterministic algorithms, metaheuristic optimization methods excel at solving problems with uncertain search spaces. These optimization approaches have been utilized in practically every sector of research, technology and engineering to discover the best answer from several feasible solutions [3].

1.2 Background and Existing Challenges

1.2.1 Optimization

An important paradigm that is everywhere along with a wide range of utilizations is optimization. In practically all application areas such as mathematics, computer science, operation research, industrial and engineering designs, we are continually attempting to upgrade something - regardless of whether to limit the expense and vitality utilization or to expand. The benefit yields execution effectiveness. In all actuality, assets, time and money are consistently restricted; thus, optimization is unmistakably progressively significant [4]. How the optimization algorithm works are shown in Figure 1.1.

Optimization is the study of choosing the best choice among a debilitated hover of choices [1] or it tends to be seen as unitary of the major quantifiable mechanism in a system of dynamics in which judgments must be employed to enhance single or more evaluations in some affirmed set of conditions [6]. Each problem of optimization accompanies some decision variables, certain objective (fitness) functions and few constraints [1]. A literature review of optimization algorithms reveals that there is no systematic classification is available. Figure 1.1 shows how an optimization algorithm work. Taxonomy of optimization algorithms is shown in Figure 1.2.

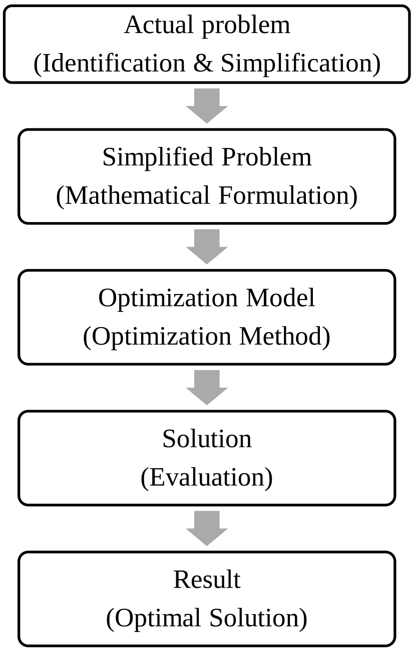


Figure1.1. Flow chart of Optimization

1.2.2 Existing Challenges in Optimization

The effectiveness of an algorithm, the effectiveness and precision of a statistical simulator and assigning the correct methods to the stated problem are the three key challenges in simulation-driven optimization and modelling.

Algorithm's Effectiveness

It's critical to have a good optimizer to get the best results. An optimizer is essentially an optimization technique that has been appropriately built to perform the required search. It may be connected and merged with other modelling elements. No free lunch theorem states that [2], there are several optimization methods in the literature and no one solution is suited for all issues.

Algorithm’s Correctness

The selection of the appropriate optimizer or method for a particular issue is critical from an optimization standpoint. The kind of issue, the structure of the methodology, the desired output quality, the contemporary computing sources, time frame, the method’s implementation availability and the selection experience will all influence the algorithm selected for an optimization job [3][4].

Effectiveness of statistical Solver

Any method for decreasing computing time, whether by limiting the number of assessments or enhancing the simulator's effectiveness, save both time and money. The major approach to reduce the number of objective assessments is to utilize an effective algorithm that requires a minimal number of evaluations [5].

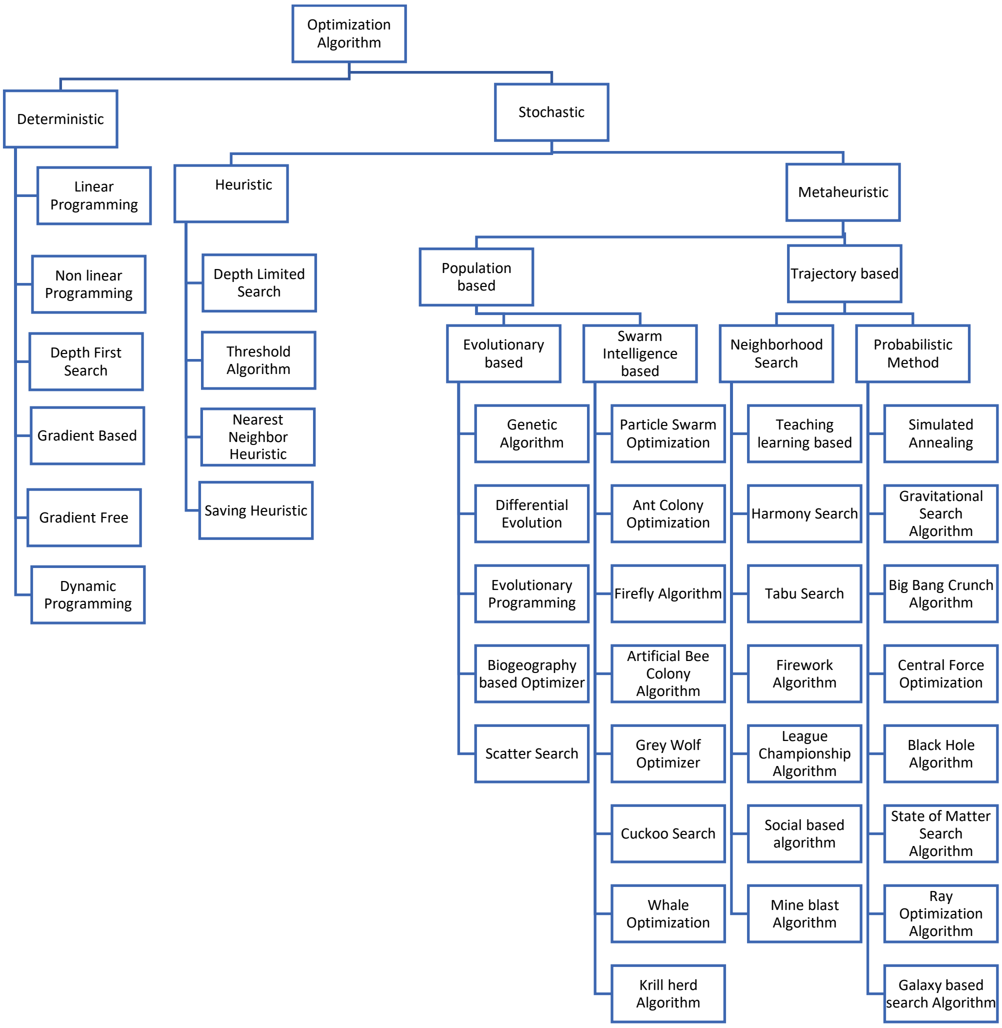


Figure1.2. Taxonomy of optimization algorithm [6].

1.3 Metaheuristic Optimization

In metaheuristic algorithms meta-denotes “beyond” or “higher level”. They outperform basic heuristics. Local search and global exploration are used in some manner by all metaheuristic algorithms. The terms 'heuristics' and ‘metaheuristics' are sometimes used interchangeably by scholars. However, a recent trend has been to label any stochastic algorithms that include randomization and global exploration as metaheuristics. Metaheuristics can be an effective technique to provide acceptable solutions to a complicated problem through trial and error in a reasonable amount of time. There's no certainty that the best solution will be discovered and we have no way of knowing whether an algorithm will work or why it will work [7]. The goal is to create an efficient and practical algorithm that works the majority of the time and produces high-quality results [8]. According to [9] “*Metaheuristic computing is an adaptive and/or autonomous methodology for computing that applies general heuristic rules, algorithms and processes in solving a category of computational problems.”*

Metaheuristic algorithms have two main characteristics: intensification and diversification. The intensification phase, also known as exploitation, searches for and identifies the best candidates or solutions based on the present best approaches. The diversification phase, also known as exploration, guarantees that the algorithm efficiently traverses the search space. A tight balance between these two components has a significant impact on an algorithm's overall efficiency. If the exploration is insufficient and the exploitation is excessive, the system may become stuck in a local optimum. Finding the global optimum would be extremely difficult, if not impossible, in this instance. When there is an excessive amount of exploration but not enough exploitation, the system may be unable to reach convergence. In this particular scenario, the total search performance suffers (Figure 1.3). Balancing these two components is a big optimization challenge in and of itself [10][11].

There are so-called "No free lunch theorems," which can have considerable impacts in the optimization field [12]. According to this, “If algorithm A outperforms algorithm B for particular optimization functions, then B will outperform A for all other functions”. This means if both algorithms A and B are averaged over all potential function space, they will perform well equally. In short, there is no uniformly superior algorithm [13].

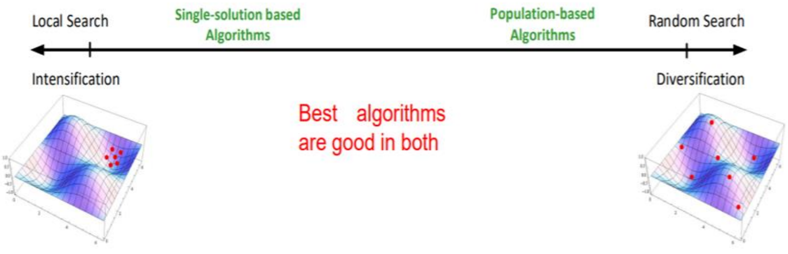


Figure1.3. The balance between intensification and diversification

According to [14], Metaheuristic algorithms share the following traits:

- The algorithms are based on natural events or behaviours and they follow specific rules (e.g., biological evolution, physics, social behaviour).
- Probability distributions and random processes are used in the selection phase, which contains random elements.
- They provide several control parameters to modify the search method since they are intended to be general-purpose solvers
- They don't depend on a priori knowledge, which is information about the process that is accessible before the optimization run begins. Nonetheless, such knowledge may be beneficial to them (e.g., to set up control parameters).

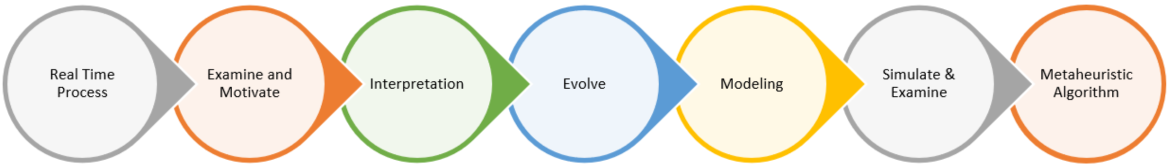


Figure 1.4. Development Procedure of Metaheuristic Algorithms [15]

Necessary steps which must be taken into account for the development of any metaheuristic algorithm are shown in Figure 1.4.

1.3.1 Recent Metaheuristic Optimization Techniques

The metaheuristic algorithms which are used for different nonlinear system analysis in this thesis are discussed below:

A. Particle Swarm Optimization (PSO)

Doctor Kennedy and Eberhart proposed the particle swarm optimization technique in 1995 [16]. It is a heuristic global optimization approach. When looking for food, the birds may scatter or congregate before locating a location where they can obtain food. While birds are moving from one location to another in quest of food, there is usually one bird that can smell the food extremely well, indicating that the bird is aware of the location where the food can be obtained, providing superior food resource information. As they are constantly passing information, especially positive information, while looking for food from one location to another, the birds will inevitably migrate to the location where food can be obtained. In terms of the particle swarm optimization technique, solution swarms are related to bird swarms, with birds traveling from one location to another representing the growth of the solution swarm, excellent information representing the most optimist solution, and food resource representing the most optimist solution throughout the process [17].

B. Stochastic Fractal Search (SFS) Optimization

The stochastic fractal search (SFS) algorithm [17] is a potential technological method that makes use of randomized fractals created using the diffusion-limited aggregation (DLA) technique. The technique was developed to address the shortcomings of prior metaheuristics, including premature convergence and poor solution quality. Within a few rounds, the SFS algorithm is capable of identifying a solution that has the least or most minimal error relative to the ideal solution, resulting in a considerable increase in solution quality and convergence time [18].

The diffusion and update processes are two critical components of the SFS algorithm. To meet the intensification(exploitation) condition, each particle initially diffuses about its present location, a process analogous to Fractal Search. This strategy enhances the probability of getting the global minima while avoiding being trapped in the local minima. The approach duplicates how, in the latter process, a point in a group modifies its location in response to the location of other points in the group. Updates are being made to procedures. To put it another way, the updating process in SFS results in the diversity (exploration) of metaheuristic algorithms [19].

C. Cuckoo Search (CS) Optimization

In addition to their beautiful song, Cuckoos are intriguing birds as they have active breeding strategies. Ani and Guira cuckoos hatch eggs in cooperative nests but may destroy other people's eggs to optimize the chances of their egg hatching. There are a few kinds of birds that are obligated to lay their eggs in other bird’s nests. The three kinds of brood parasitism are cooperative breeding, nest takeover and intraspecies brood parasitism. Certain host birds may come into direct combat with the invading cuckoos. It is possible for a host bird to either discard the eggs or abandon the nest and build a new one elsewhere if it recognizes that the eggs are not it's own. There are a few species of crows that have acquired the ability to mimic the colour and design of a few select host species' eggs to prey on them. A reduction in egg abandonment and an increase in reproduction are the outcomes of this.

Some species can deposit eggs at certain times is astounding. Cuckoo parasites like newly born eggs in the host bird's nest. Cuckoo eggs hatch sooner than host eggs in general. An instinctual response to hatching is for a cuckoo chick's first instinct to expel its host's eggs from its nest by tossing them out of the nest. In [21] research has shown that to get access to more food, a cuckoo chick may mimic the cry of its hosts.

Animals in nature look for food in a semi-random fashion. In general, an animal's foraging route is practically a random walk since the subsequent step is predicated on the present position/state and the likelihood of transitioning to the next place. Which path it picks is based on a probability that can be statistically described. Various investigations, for example, have proven that the flying behaviour of numerous animals and insects has displayed the basic features of Levy flights [18]. In general, Levy flights are random walks whose step length is determined by the Levy distribution, which is commonly expressed in terms of the $L(S) \sim |S|^{1-\beta}$ where $0 < \beta \leq 2$ is an index [18].

D. State of Matter search (SMS) Optimization

The SMS algorithm works by simulating the states of matter phenomena. Individuals in SMS imitate molecules that interact with one another by employing evolutionary processes based on the thermal-energy movement mechanism's fundamental principles. Each state of matter is considered at a distinct intensification-diversification ratio to create the method. The evolutionary process is split into three stages, each of which represents one of the three forms of matter: gas, liquid, or solid. Molecules (individuals) have varied mobility capabilities in each condition. The algorithm changes the intensity of intensification and diversification until the solid-state (pure intensification) is attained, starting with the gas state (pure diversification). As a consequence, the technique may significantly enhance the balance between diversification and intensification while retaining the evolutionary approach's strong search capabilities [19].

E. Opposition based Learning(OBL)

Elite opposition-based Learning is an emerging method in the domain of intelligent computation. Its basic concept is to compute and analyze the opposing response at the same time to find a feasible solution, and then pick the best one for the future generation [20]. Tizhoosh [21] introduced OBL, which is essentially a machine intelligence method. It provides a more accurate approximation of a current candidate solution because it considers both the existing person and its opposing individual simultaneously. A random candidate solution is less likely than an opposing candidate solution to be as near to the global optimal solution as an opposing candidate strategy.

Opposition-based learning can effectively broaden the population's search area and increase the algorithm's capacity to solve problems. It is suitable for use in conjunction with an evolutionary algorithm. Elite individuals in a population are seen to be the best of the best. Elite individuals must have more meaningful knowledge to direct the group toward global optimum convergence. If the algorithm succeeds in achieving global convergence, the elite individuals' search area will converge to the global optimum individuals' search area. As a result, enhancing elite people's spatial neighborhood search will increase the algorithm's convergence rate and improve its global convergence ability [22].

F. Water Wave Optimization (WWO)

The WWO algorithm builds search algorithms for high-dimensional global optimization problems using concepts from wave motions governed by wave-current–bottom interactions [27]. To address the optimization problem, the WWO approach efficiently balances global and local search by simulating wave motion and propagation, refraction, and breaking. The ideal solution for each water wave is related to the wave height and wavelength. The water wave has a bigger ideal solution in shallow water, a higher wave height, and a longer wavelength; in deep water, the water wave has a smaller optimal solution, a lower wave height, and a shorter wavelength [23].

1.3.2 Existing issues with Metaheuristic Optimization

Finding the optimal answer to a problem is the optimization process. As a result, the primary challenge for metaheuristics is figuring out how to cope with this issue. Even though many metaheuristics have been suggested, only a handful of metaheuristics have consistently attained the required success rate. Population-based metaheuristics, in particular, are frequently employed because they can adapt to large-scale optimization issues. Metaheuristics, as previously stated, are problem-specific algorithms. As a result, the issue is, "What is the optimal algorithm parameter specification based on the kind and size of the problem search space?" Furthermore, selecting the proper metaheuristic algorithm is a complex thing. Recent developments seek to liberalize metaheuristic methods to overcome these problems.

1.4 Nonlinear Control

Practical systems are fundamentally nonlinear, at least across a broader range of operating conditions, even though many of them are supposed to 'behave' linearly close to a certain operational point at a slower speed under specific conventions. To represent a broad variety of physical events, nonlinear models are utilized. A few examples are the gravitational and electrostatic attraction, coulomb friction, V-I characteristics of most electrical systems, and drag on a moving vehicle [24].

Nonlinear control is becoming more popular as a result of the upgrading of linear systems, the study of nonlinearities, the requirement to deal with parametric uncertainty and the flexibility of the architecture. Methods that take into consideration dynamic forces such as sensory and Coriolis forces, which fluctuate in speed, improve on basic techniques [30]. As a consequence, the linear control principles limit the speed at which a given accuracy may be achieved. It is possible to adapt to nonlinear forces using a simple nonlinear controller, allowing for high speeds in an extensive range of motion. Real-world systems can't be approximated linearly because of rigid nonlinearities such as hysteresis, dead zones, stiction, Coulomb friction, stiction, saturation and backlash. Once these nonlinearities are predicted, nonlinear approaches can compensate for them in a way that is unparalleled in terms of efficacy. Model parameter uncertainty is common in real-world systems as well, due to sudden or gradual shifts in parameter values. The resilience or flexibility of a nonlinear controller may be able to deal with the consequences of model uncertainty [25].

1.4.1 Nonlinear Control Problems

The nonlinear control problems which are used in this work for efficiency and efficacy analysis of different existing and proposed algorithms are discussed below:

A. Continuously Stirred Tank Reactor

The natural world is complicated having ecological and chemical networks that interact on a global scale. Numerous components of such systems display nonlinear dynamics and continuously stirred tank reactors (CSTR) are a good example of a processing unit that exhibits nonlinear dynamics, posing operational difficulties owing to complicated behaviour such as output operands, oscillations, and even instability [26]. A Continuous Stirred Tank Reactor (CSTR) is a critical unit operation in chemical industries with highly nonlinear behaviour and works on extensively working ranges. Chemical processes in a reactor are either exothermic or endothermic, requiring energy to be withdrawn or given to the reactor to keep a steady temperature.

Their functioning, however, is contaminated by several ambiguities. Some of these result from fluctuating or incompletely known factors, such as reaction rate constants and heat transfer coefficients. In other circumstances, reactor operating points fluctuate, or reactor dynamics are influenced by changes in parameters or even the instability of closed-loop feedback systems [27].

B. Ball Balancer System

An underactuated, multivariate electromechanical, and nonlinear system may be represented by the use of ball balancer systems. Two servo motors operate simultaneously to regulate the position of the ball in a 2-DOF Ball Balancer. This is a nonlinear system. In many applications and approaches, it is one of the most complicated control benchmarking systems. Users may experiment with a variety of control methods to direct a ball towards a certain spot on a table. It's a horizontal plate with slants in both directions, which allows the ball to roll wherever on the plate. Nonlinear kinematics and control theory are shown dynamically in this system. Control algorithms and technologies are often evaluated using this system because of its inherent nonlinearity, instability and under-actuation [28].

C. DC Motor

DC motors are commonly used as actuation elements in engineering applications due to their ease of speed and position control and large adjustability range. They play a critical role in a variety of electrical systems used in residential and industrial applications, including industrial mills, electric vehicles, cranes, robotics, and a variety of household products. This prominence is attributable to their attractive features, which include accuracy, simplicity, and continuous control. A good control strategy is required to run the DC motor at the right speed or torque [29]. As a result, studying DC motor behaviour is a worthwhile endeavour for the analysis and control of a wide plethora of different applications [30].

1.4.2 Existing Challenges

Nature is nonlinear, therefore nonlinear methods are the easiest way to cope with it. Despite this, linear control has been used effectively for years. The issue with linear systems is that they may not be capable of accommodating recent and innovative technologies. It may be challenging to decide whether to use linear or nonlinear control for a certain application. Linear control has been well researched and industry professionals trust it. For linear systems, there are many good analytic methods available, including the root locus, bode plot, Nyquist stability criterion, Laplace transform, Z-transform, Fourier transform and many more. Nonlinear systems, on the other hand, require more sophisticated numerical methods, such as the Lyapunov stability criteria, the Popov criterion and singular perturbation techniques. For nonlinear systems, mathematical modelling may be time-consuming. Limit cycles, chaos and bifurcation may occur in nonlinear systems. The majority of the strategies can only guarantee local stability; global stability cannot be assured.

1.5 Hybrid Metaheuristic Algorithms

The first two decades of metaheuristics study were defined by the use of very typical metaheuristics. However, it has become clear in recent years that focusing on a single metaheuristic is quite limited. When dealing with real-world and large-scale situations, a smart combination of a metaheuristic with other optimization approaches, known as a hybrid metaheuristic, may give more efficient behaviour and more flexibility. This may be accomplished by combining the complementing qualities of metaheuristics on one hand with comprehensive approaches like a branch and bound techniques or mathematical programming on the other [31]. Hybrid algorithms blend two or more algorithms to solve a specified issue simultaneously and effectively. Within a hybrid structure, intrinsic modules of algorithms like crossover and mutation are used to augment other algorithms. Hybrid algorithms may be split into two groups in terms of their scope [32].

Hybrids that serve many functions: To answer the same question directly, all sub-algorithms are applied, and distinct sub-algorithms are used at multiple stages of the search. Algorithms that combine local search with metaheuristics algorithms are an excellent example of this. In contrast, the global exploration expands the solution space, while the local search narrows the areas in which the global optimum may be found

hybrids that have several purposes: The main optimization algorithm is used to solve the issue, whereas the sub-algorithm is used to fine-tune the parameters of the main method. It may be used to discover the best mutation rate in GAs, for example. To put it another way, rather than fixing the issue, PSO is aiding the creation of better solutions by determining which parameters are the most beneficial for increasing performance. Consider hyper-heuristic algorithms as a synthesis of many methods. Parameters are picked in hyper-heuristic techniques (either by a sub-algorithm or by a learning approach) [38].

1.5.1 Existing Challenges

Although hybrid algorithms have the benefit of enhancing population diversification and therefore improving the search capacity of the generated hybrid algorithm, they do have certain limitations, as discussed below

- The addition of another algorithm almost always results in a naming problem. Some academics call their hybrid algorithms by a variety of names.

- In contrast to the two architectures, the collaborative hybrid algorithm seems to generate more complex names. For example, the titles Hybrid GA-PSO and Mutated PSO are similar, even though both hybrids include GA and PSO.

- The hybridization process typically adds additional components to the overall design of the hybrid algorithm. The hybrid algorithm's complexity raises as a result.

- Since hybrid algorithms have a more sophisticated design, overhead is introduced along with their complexity, which is sometimes inevitable. This has an impact on overall performance and, as a result, reduces its robustness.

- The majority of hybrid algorithms will increase their parameter count, making it more difficult to modify their settings. Furthermore, the complexity of a hybrid makes it harder to evaluate, obscuring the origins of such hybrids.

Hybridization of algorithms are also somewhat more difficult to develop and therefore more prone to mistakes. As a consequence, while evaluating the findings of hybrid algorithms, caution is advised [32].

1.6 Motivation

Following are some key insights from the previous research and observations that becomes the motivation for this research:

- "Everything should be made as simple as possible, but not simpler," Einstein famously remarked. Less complicated algorithms are preferable. Algorithms, in the same way, should be as straightforward as feasible. In practice, a robust algorithm is preferred with a simpler architecture for simplicity of implementation while still being efficient enough for real-world applications.

- The installation of new hybrids (integrative hybrids) should be simplified by having a specified structure. To produce long-term better hybrids, each combination should be predicated on clear thought, innovative characteristics, and intelligent methods.

- The researchers developed a stable and pleasing nonlinear system response by either employing a regular PID controller or its more flexible variant namely, FOPID, 2-DOF-PID controller. The purpose of this study is to develop PID controllers with filters, as well as variations such as FOPID and 2-DOF-PID controllers to enhance the nonlinear system response. Additionally, controllers are constructed using not just integer-order filters but also fractional-order filters.

1.7 Objectives

The key objectives of this research titled" Metaheuristic Algorithm and its applications to nonlinear control system "are as follows:

- Study of different recent metaheuristic algorithms.
- To develop a novel hybrid metaheuristic algorithm that may give a more efficient and more flexible behaviour for nonlinear control problems.
- To validate the developed hybrid metaheuristic algorithm on Benchmark Functions and perform Trajectory Analysis, Wilcoxon rank-sum test, mean, median and Standard deviation values.
- To implement the proposed algorithm and other existing metaheuristic algorithms for parameter tuning of the FOPID Controller for concentration and temperature control of continuously stirred tank reactor.
- To implement the proposed algorithm and other existing metaheuristic algorithms for parameter tuning of the PID Controller for angle and position control of the Ball Balancer system.
- To implement the proposed algorithm and other existing metaheuristic algorithms for parameter tuning of the 2-DOF-PID controller for speed control of DC motor.

1.8 Methodology of the Research Work

The methodology opted for this research is shown below in Figure 1.5.

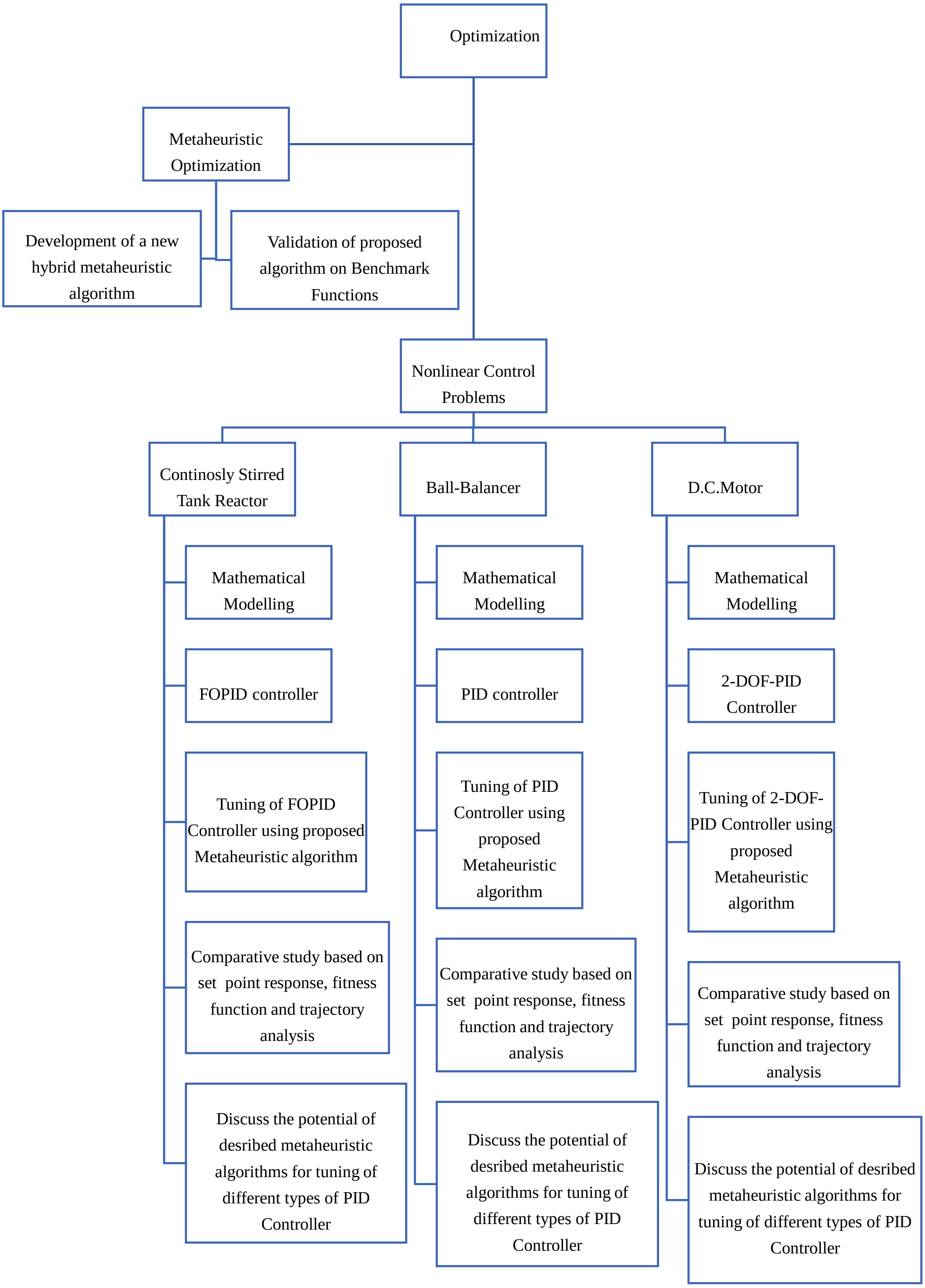


Figure1.5. Research work Methodology

Simulation is carried out using MATLAB 2018, which is powered by an Intel(R) Core (TM) 2 Duo CPU T6400 running at 2.00 GHz and 1.20 GHz with 1.99 GB of RAM. 1.9

1.9 Outline of the Thesis

After an introductory chapter that describes the basic concept of optimization along with a brief on metaheuristic optimization, different types of metaheuristic optimization along with various nonlinear control problems, further chapters of this thesis are organized in the following manner:

Chapter 2 provides a detailed literature review of recent metaheuristic algorithms, hybrid metaheuristic algorithms, metaheuristic algorithms for PID controller tuning, hybrid metaheuristic algorithms for PID controller tuning, optimal tuning of FOPID controller for CSTR, PID controller tuning for Ball balancer and optimal tuning of the 2-DOF-PID controller for CSTR.

CSMSEOBL is a unique hybrid metaheuristic algorithm that is being developed in Chapter 3. A hybridization of State of matter search with Chaotic maps and Elite opposition-based learning has been created, which is referred to as CSMSEOBL. To verify the developed method on 15 benchmark functions, statistical analysis, convergence analysis and the Wilcoxon test are performed.

Chapter 4 deals with mathematical modelling of CSTR and tuning of FOPID Controller parameter by using the proposed hybrid algorithm CSMSEOBL, State of matter search algorithm, Cuckoo search algorithm and Particle swarm optimization algorithm. Error convergence analysis, and setpoint response analysis are done to validate the superiority of the proposed algorithm.

Chapter 5 describes the mathematical modelling of the Ball balancer and tuning of PID Controller parameters by using the proposed hybrid algorithm CSMSEOBL, State of matter search algorithm, Stochastic Fractal search algorithm and Particle swarm optimization algorithm. Error convergence analysis, setpoint response and convergence analysis are done to validate the pre-eminence of the proposed algorithm.

Chapter 6 deals with mathematical modelling of DC motor and tuning of 2-DOF-PID controller parameter by using proposed hybridization CSMSEOBL, State of matter search algorithm, water wave optimization and particle swarm optimization algorithm. Error convergence analysis, setpoint response investigation and convergence study are done to validate the superiority of the proposed hybridization.

Chapter 7 provides the concluding remarks for all the approaches and identifies the state of possible directions for future research.

Chapter 2

Literature Review

2.1 Introduction

The challenge and goals of the current research work are described in Chapter 1. A brief literature review has been carried out on the following issues:

- History of Optimization
- Review of metaheuristic optimization algorithms
- Review of hybrid metaheuristic optimization algorithms
- Hybrid metaheuristic optimization algorithms used to tune PID Controller and its variants
- Optimal control of CSTR system.
- Optimal control of the Ball balancer system.
- Optimal control of DC motor.

As optimization is widespread, it is a significant paradigm having a broad array of applications. We try to optimize something in almost every application in the engineering and manufacturing sector, whether it be to reduce expenses or to increase profit, outcome, productivity and effectiveness. Every part of our lives, seen or unseen, covers a diverse spectrum of optimization difficulties inside its edges. Not only that, but any practical system, or indeed a fraction of one, may be generalized as an optimization system, with one or even more optimization concerns embedded within it. The major goals of optimization algorithms (OAs) are to make anything more and more efficient to the maximum degree feasible by iteratively looking for more precise and flexible solutions to the problem at hand [33][34]. This indicates that the problems under consideration must eventually be reconstructed in the context of optimization, which has two sides: first, creating a "search space" of potential solutions for the specified issue; second and more significantly, evaluating the obtained performance of all accessible solutions based on certain performance standards that should be stated prior [35][34][36].

Optimization is a method for finding solutions with significant nonlinearity, complexity and wide solution space. As a consequence of the increasing complexity, a method is required that can provide a high-quality solution in a reasonable period while using existing resources. The metaheuristic algorithm is an example of a method that has become an essential component of all optimization processes. Finding an optimal answer is just the last stage of an optimization process that includes characterizing the system mathematically, identifying restrictions, defining system attributes, and finding an objective function. In a wider sense, the optimization process may be categorized into two types: (1) exact methods and (2) approximation methods [5][6]. The exact approach ensures the best possible answer, while the approximation method ensures a high-quality solution in a fair period but not optimality. Branch and bound approach and dynamic programming are examples of exact optimization techniques, whereas approximate methods include cut & plane, local search, scatter search, genetic algorithms and others. further Approximation algorithms and heuristic techniques are two types of approximate procedures. The first strategy provides proven arrangement quality and run-time limitations, whereas the second focuses on obtaining a sensible excellent arrangement in a reasonable time. Calculations based on heuristics are very problem-specific. Metaheuristics are a kind of algorithm that demonstrates the basic heuristics in the same way that a governing system does. They don't specify an issue or a region and they may be used for any optimization. Glover [7] [5] coined the phrase "metaheuristics”.” Meta-heuristics are logically improved to get an ideal arrangement that is "acceptable" in a "sufficiently short" figuring time”. Because of its (i) simplicity and ease of implementation; (ii) lack of requirement for slope data; (iii) avoidance of neighborhood optima; and (iv) applicability to a broad range of problems involving different controls, meta-heuristic optimization aids in the resolution of a wide range of ongoing challenges. [8]

2.2 History of Optimization

Table 2.1 A brief review of historical optimization

Year	Name of the Developer	Features
300BC	Euclid(Greek Mathematician)	The greatest area is enclosed by a square among all feasible rectangles having all four sides of equal length.
100BC	Heron	The angle of incidence is equal to the angle of reflection
1613	Johanes Kepler	Find an optimal solution for the secretary problem
1621	W.van Royen	Law of refraction
1637	Rene Descarates	Snell’s results
1657	Pierre de Fermat	Fermat’s Principle (in any medium-light always travel in the shortest time)
1685	Sir Isaac Newton	To minimize the resistance to fluid motion
1696	J.Bernoulli	Development in the area of calculus.

Year	Name of the Developer	Features
1670	Euler and Lagrange	Calculus of variation
1746	P.L.de Maupertuis	Principle of least action
1781	Gapard Monge	Optimal transportation problem and sharing of resources.
1801	Frederich Gauss	Method of least square to anticipate the area of asteroid Cereas
1806	Adrein legendre	Method of least square for curve fitting
1815	D.Riacrdo	Law of diminishing return for the cultivation of land
1847	L.A.Cauchuy	gradient methods, Steepest Descent
1906	J.Jensen	The concept of convexity also introduces the concept of inequality
1917	H.Hancock	The first book on optimization named "Theory of minima and maxima"
1930	Karl Menger	Messenger's problem (search for the shortest route which joins a definite number of points (or cities) and prairies distance is known. Now a day it is known as the travelling salesman problem.
1939	L.Kantorovich	Developed an algorithm for linear programming .it was used in economics for optimal planning of production-related issues. For this research, he was awarded the Nobel prize.
1944	John von Neumann and O.Morgenstern	Develop the solution for sequential decision problems also, develop operational research.
1947	George Dantzig	The simplex method is developed for large-scale linear programming problems. From 1939 to 1947 linear programming was developed three times but each time with a different formation.
1951	Harold Kuhn and A.W.Tucker	Karush-Kuhn-Tucker condition is a necessary condition for a solution to be optimal in the case of nonlinear programming.
1957	Richard Bellman	Develop dynamic programming and the principle of optimality.
1960	**A big explosion occurred in the field of optimization**	

All the above-discussed methods of optimization fall into the category of exact optimization and many of these methods when applied to real-time non-linear problems, they face the problems of trapping into local optima, risk of diversification, constraint handling issues etc. to overcome these limitations of the exact method of optimization. heuristic methods of optimization were proposed in the early ’70s [37]. Although heuristic methods do not guarantee the optimal solution but near-optimal and feasible solutions can be found in a reasonable amount of time. The heuristic method was introduced by G.poyla [38] in 1947 but its actual development begin after 1960. Essentially, a heuristic is intended to give better computational execution most of the time but at the cost of reduced accuracy when contrasted with traditional optimization algorithms. As heuristics are problem-specific so use domain-specific knowledge and are well defined only for basic problems [39]. In more exact terms, heuristics are methodologies utilizing promptly open, however freely accessible, data to control critical thinking in human creatures and machines [6].

The compromise standards for concluding whether to utilize a heuristic for taking care of a given issue incorporate the accompanying

- Optimality is a heuristic, able to give optimal solutions among the several available best solutions.
- Completeness: among multiple existing solutions, will a heuristic method be able to get all? Because heuristics are mostly used to determine a single solution.
- Efficiency and Definiteness's
- Execution time [6]

2.3 Metaheuristic Optimization

This literature review reveals that the heuristic method of optimization is quicker compared to the exact method of optimization which is capable to give an exact solution but at the expanse of high computational time. The limitation of heuristic optimization is problem specific [6]. A heuristic is a method of reasoning in critical thinking that is based on trial and error. As a result, they are prone to falling into local optimum traps and being unable to escape [46]. Whereas A metaheuristic is a nonexclusive or higher-level heuristic that is more inclusive in its approach to problem resolution. Metaheuristic algorithms may be thought of as a kind of flexible processing that use generic heuristic guidelines to solve a class of computational problems [6]. Glover coined the word "metaheuristic" in 1986. Glover defined metaheuristic as “A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms”. Metaheuristic term join two words “Meta” which is a Greek prefix meaning is “high level” with heuristic

Metaheuristics can be an effective method to create satisfactory arrangements by experimentation with an intricate issue in a sensibly acceptable time [40]. Wang[9] defined metaheuristic in several ways as

1. “Metaheuristic computing is an adaptive computing that applies general heuristic rules in solving a category of computational problems.”
2. “A metaheuristic is a generic or higher-level heuristic that is more general in problem-solving.”[1]

These definitions lead to a general mathematical formulation of a metaheuristic algorithm, which is

MHA=$(O, A, R^C, R^I R^O)$

Where

O-set of metaheuristic algorithms

A-group of generic algorithms

$R^C -$ set of inner relations (OXA)

R^I – set of input relations

R^O -set of output relations

Characteristics of Metaheuristic Algorithms:

- Meta-heuristics are techniques that direct the search measure. The objective is to effectively and efficiently investigate the search space to discover feasible or close to the optimal solution.
- Strategies that comprise meta-heuristic calculations run from straightforward neighbourhood search methods to complex learning measures.
- Metaheuristic calculations spread the arrangement space without stalling out in explicit zones (particularly local optima).
- Meta-heuristic calculations are not only approximate and ordinarily non-deterministic but also not peculiar for a problem [41][6].

- Metaheuristics can be classified based on their abstraction level.
- Metaheuristics are relatively simple to implement in parallel.
- Metaheuristics cover a wide range of topics, from simple local searches to complex learning approaches.
- Heuristics are domain-specific knowledge that is dominated by the topmost strategy and may be used by a metaheuristic.
- Advanced metaheuristics rely on guiding memory to keep track of search history [42].

The essential terminology in metaheuristic computing is explored here to present the notion of metaheuristic computing as clearly as possible.

Definition 1. **"***A heuristic is a reasoning methodology in problem-solving that enables a solution to a problem is derived by trial-and-error and/or rule of thumb".*

Definition 2. "*A metaheuristic is a generic or higher-level heuristic that is more general in problem-solving".*

Definition 3. "*Computing in a narrow sense is an application of computers to solve a given problem by imperative instructions; while in a broad sense, it is a process to implement the instructive intelligence by a system that transfers a set of given information or instructions into expected intelligent behaviours".*

The notion of metaheuristic computing may be stated as follows based on Definitions 1 through 3.

Definition 4. "*Metaheuristic computing is an adaptive and/or autonomous methodology for computing that applies general heuristic rules, algorithms and processes in solving a category of computational problems"*[9].

The optimum search strategy is shown in Figure 2.1 and the approach of using meta-heuristic algorithms to select the optimum result is shown in Figure 2.2.

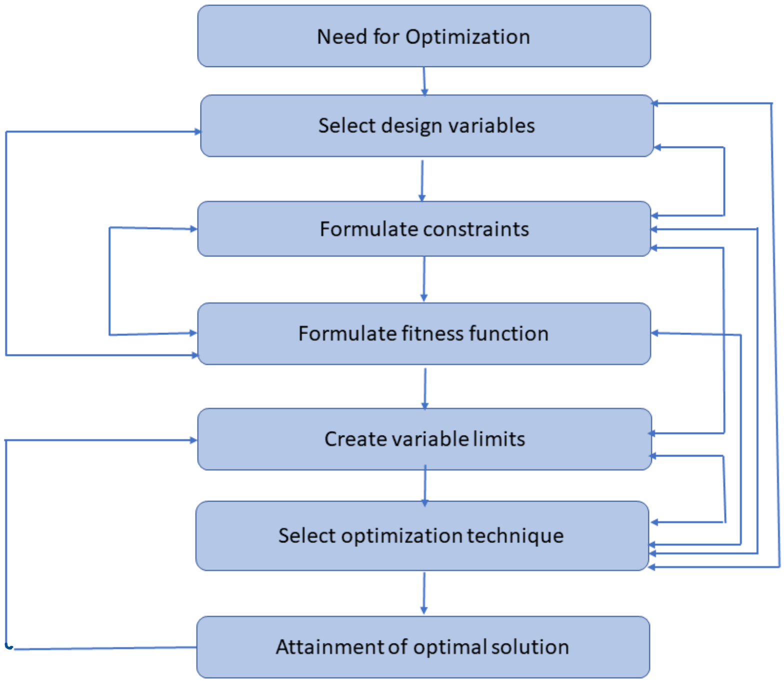


Figure 2.1. Flow chart of the optimum search strategy

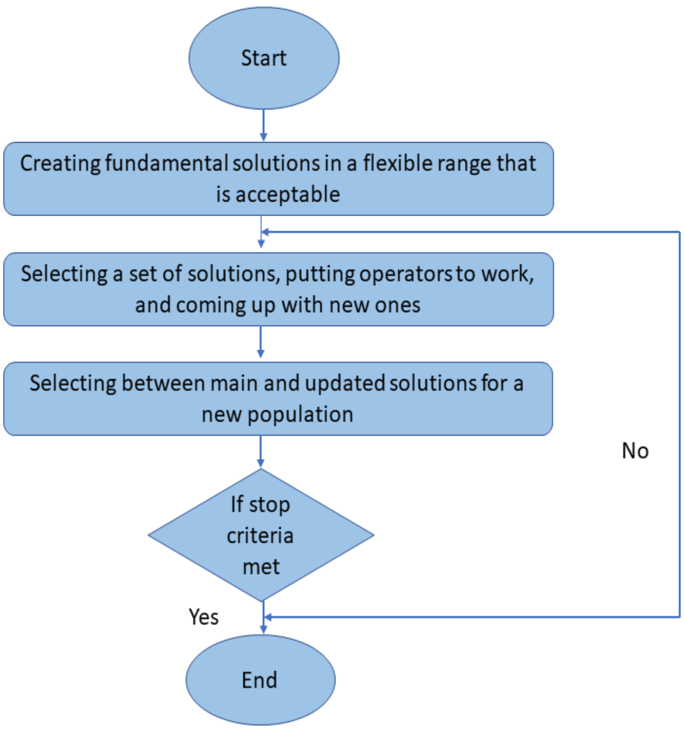
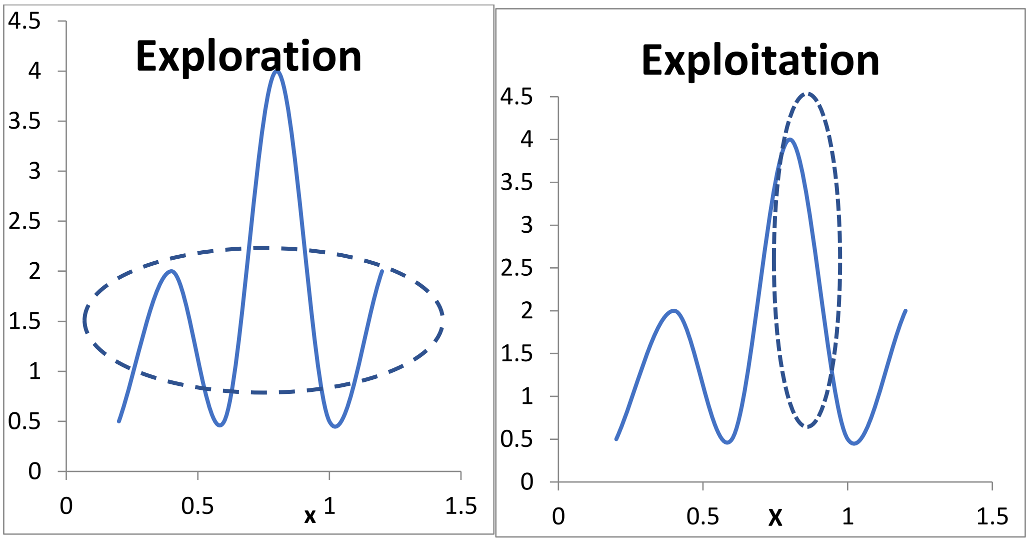


Figure 2.2 The approach of using meta-heuristic algorithms to select the optimum result [43].

2.3.1 Key factors of Metaheuristic Algorithms

The balance between intensification and diversification

These two are elementary concepts for any metaheuristic algorithm (shown in Figure 2.3). Intensification and diversification are also termed exploitation, convergence and exploration, and diversification respectively. In Diversification, search is for the entire search space to ensure prevention of trapping into local optima whereas intensification searches only in the promising area of the search domain ensuring convergence along with optimality [44]. For a metaheuristic algorithm to be successful, it must strike a fair balance between intensification and diversity [41].



(a) (b)

Figure 2.3. (a) Diversification and (b) Intensification of a metaheuristic algorithm [41]

Population-based algorithm and single solution algorithm

The next important key factor of any metaheuristic algorithm is to find whether it is a population-based or single solution-based algorithm. Population-based algorithms are more diversified or explorative and at a time numerous solutions get generated and move in direction of the optimal solution, these multiple solutions can be generated by re-amalgamation of various solutions or by updating each solution [45].

Single solution-based algorithms are also termed trajectory-based algorithms, these algorithms are diversification oriented and get started with a single initial solution to form a trajectory to move towards an optimal solution [1].

Local search and global search metaheuristic

Local search-based metaheuristic methodologies are more intensified in nature whereas global search metaheuristic shows diversified behaviour. Local search metaheuristic includes Tabu Search [46], iterated local search, etc. whereas global search metaheuristic includes genetic algorithm, Particle Swarm Optimization, etc.

2.4 Optimal control of CSTR

In chemical processes, a CSTR is a popular reactor for effective mixing. Industrial uses [47], medicines [48]and wastewater treatment units [49] all utilized it. The requirement to maintain its temperature and output concentration at predetermined levels drives researchers to a computational model. The CSTR model is complicated and nonlinear, exhibiting both stable and oscillatory behaviour, due to its mixture of chemical processes and phase equilibrium. an example of analysing CSTR dynamics using the homotopy perturbation approach has been shown in [50] and an example of analysing the influence of CSTR parameters on its stability is presented in [51].

A partial state feedback controller [58] was created to provide CSTR's global setpoint tracking control. Delbari et al. [59] evaluated and regulated CSTR parameters using adaptive general predictive control (GPC). With the use of an optimum disturbance rejection PID controller, Krohling and Rey [60] explain how to employ evolutionary algorithms to tackle bounded optimization problems in a servo motor system. A modified genetic algorithm was used to determine the optimal PID controller parameters for a variety of process types [61]. The technique for changing the control parameters of the dynamic plant was devised utilizing fuzzy gain scheduling [62]. Nagaraj and Murugananth [63] investigated PID tuning utilizing a controller based on soft computing to improve the performance of the process in terms of time-domain needs, setpoint tracking, and regulations. [64] developed a PID controller for nonlinear and unstable CSTR systems using an artificial bee colony approach, which was inspired by the work of Chang [63]. Using the PSO technique, Singh and Sharma developed a FOPID controller that demonstrates improved servo and regulatory response [65]. Jayachitra and Vinodha devised and applied GA-PID tuning in the CSTR process. Set-point tracking and disturbance rejection are possible with the use of the ISE, IAE, and ITAE cost indices in combination [66].

2.5 Optimal control of Ball Balancer System

Newtonian mechanics are used to modelling the mechanism in [67]. [68] proposes that the system has 2-DOF using Lagrange's technique. An equilibrium model is then linearized around the nonlinear model to generate a state-space model [69]. The equation of Lagrange's second form is used to represent the structure, as shown in reference [70]. In addition, several control approaches have been used to stabilize the ball balancer structure [71]. Conventional PD controllers are used to regulate the nonlinear model in [72]. SIMC-PID and H-∞ approaches are used to control the machine in [15]. In [73], a PID controller, neural network, and LQR regulation were used to regulate the device. As an alternative to the foregoing controls, nonlinear control may be achieved using adaptive iterative learning approaches such as the Kalman particle filter. With each additional state variable, the number of estimates required by the particle filter solution grows exponentially [76]. This is a severe downside. Traditional nonlinear approaches included feedback linearization, as well as partial feedback linearization [77] for underactuated mechanical systems, to address these concerns. Because of this procedure, the nonlinear system became a linear system with the same properties.

There must be a control strategy that can achieve steady-state activity for all the above-described systems, due to the difficulties involved in controlling underactuated systems. An important benchmark system for underactuated systems is the ball and plate technique. In general, control system designers and operators find it difficult to develop and operate systems for ball and beam balancing control because of the system's high degree of nonlinearity and instability The underlying dynamics and control theory for all ball and beam systems [78][79][80] is the same. There are two possible ways for the ball to go concerning the center of the beam: it may travel in either of two directions, left or right. the ball's acceleration and placement are controlled by an electrically powered servomotor that is connected to the beam [52].

For one-to-one management of the ball on the plate framework, the nonlinear PID controller has been explored in classical control [82]. As the PID reaction is organized utilizing the extended Kalman—Yanukovych—Popov lemma (GKYPL) approach, it improves in terms of the relentless state response when compared to a standard PID [83]. Numerous researchers have utilized a mix of trial and error and the Ziegler Nichols technique to optimize the process's efficiency within reasonable restrictions [84][85]. To address these issues, a novel control approach based on fuzzy logic has been developed, combining a hybrid fuzzy controller with model-based PID [86][87]. The fuzzy controller is effective at rejecting disturbances and has no steady-state error. The Lyapunov dependability hypothesis was used to acquire an explicit control rule and swiftly regulate performance [88]. On ball and plate systems, interference rejection controllers [89] and metaheuristic optimization algorithms [90] were proven to achieve the requisite tracking efficiency. MPC i.e. Model Predictive Controller was also often utilized in ball balancers due to benefits for time-varying references [91]. The primary disadvantage of these techniques are that they result in a longer settling time and high peak overshoot. Additionally, various intelligent and hybrid controllers were used to obtain self-balancing, trajectory detection, and position monitoring for the ball-plate system, including fuzzy [92], PSO-based fuzzy-neural controller [93], and fuzzy-neural controller [93]. The system is controlled in [94] using a PID controller, as well as a neural network using LQR control. A comparison of LQR subspace stability and integrated error metric strategies is seen in [53].[54] [19] uses the SA and CSA heuristic tuning approaches to tune the controller. To tune the PID controller, GA and DE algorithms are used and output is evaluated using error criteria in [55]. LQR parameters are tuned using GA for process control[56]. [57] proposed a multi-objective PSO to develop SIRMs coupled fuzzy controllers for ball beam structure and inverted pendulum using convergence and divergence operators. The PID controller is often utilized in practical engineering applications, even though numerous control techniques exist for achieving self-balancing control of ball balancer systems. There are several advantages to using a PID controller, such as its simple construction, high dependability, and stable performance. Classic PID controllers, on the other hand, have a problem with parameter tuning, which is a major disadvantage.

There are several methods for tuning PID parameters that can be found in the literature. Various intelligent methods such as fuzzy [58] and neural network [59], genetic [60] and evolutionary algorithms [61] are used to tune PID controller parameters.

2.7 Optimal control of DC Motor

PID controllers are the most commonly used controllers in process industries because of their simplicity, efficacy and durability. The system becomes unstable when process parameters change and disturbances occur. PID tuning is often done manually in all types of processes and manual tuning is a time-consuming procedure [62]. Over the last few years, Zeigler–Nichols tuning has become increasingly popular; nevertheless, this approach requires a prior understanding of the plant model. System stability necessitates the use of auto-tuning techniques. According to the literature, traditional PID controller tuning strategies including manual tuning, Ziegler Nichols, and Cohen–Coon procedures are unable to optimize complex higher-order processes for optimal performance. [63].

As we go forward into a new era, the intelligent control scheme has shifted from a traditional approach to a new phase based on optimization [64]. Finding the optimum PID controller parameters may be considered an optimization issue. The process of finding the decision variables of a function to reduce or increase its values is known as optimization. Non-linear limitations, huge computational charges, non-convex, complex and the massive number of solution spaces constitute the majority of real-world issues [65]. As a result, addressing problems with a high variety of parameters and constraints is time-consuming and difficult [66][67]. Second, using traditional numerical approaches results in numerous local optimal solutions that do not guarantee the best response [68]. Metaheuristic optimization methods, which are capable of tackling such complicated problems, are proposed to address these issues [69]. There has recently been a surge of attention in evolving metaheuristic optimization methods Because of their adaptability, accessibility, lower mathematical complexity and prevention of local optima. When we talk about adaptability, we're talking about the ability to apply such methods to a wide range of technical issues. For many complicated tasks, such algorithms produce adequate outcomes [70]. It's easy since it's based on natural events such as evolutionary biology to complete a task, physical phenomena and other evolutionary activities. These nature-inspired powerful metaheuristic algorithms are used to address NP-hard issues including engineering optimization [71], economic load dispatch [72], multi-objective and many-objective optimization [73][74]. Metaheuristics are simple to set up, help developers to replicate natural behaviour, modify or develop new metaheuristics and blend metaheuristics from different sources. overall, an algorithm developer needs to know and comprehend how to portray the problem. Furthermore, most metaheuristics contain processes that do not need derivation. Metaheuristics, in contrast to gradient-based optimization methods, optimize the problems at random. Metaheuristics are used to solve issues in a stochastic manner [75]. Particle swarm optimization (PSO) [76], differential evolution (DE) [77], artificial flora [78], krill herd [79], state of matter search [80], bird mating [81], jaguar [82],pufferfish [83],elephant herding [84] and monarch butterfly optimization (MBO)[85], are some of the well-known approaches in this field which have been used for PID controller parameter tuning.

Dineva et al. reviewed soft computing models in the design and control of rotating electrical equipment. The study examined the applicability of several optimization methods to rotating electrical equipment [129]. The use of GA to improve the PID controller settings for brushless DC motor speed regulation is recommended [130]. Using DE and PSO-based PID parameters tweaking, DC motor dynamic stability may be improved [131]. Additionally, an existing method has been compared with BBO's unique migration model for tuning PID parameter settings to govern the DC motor [132]. In [133], a DC motor's speed may be controlled using the SFS algorithm. With the help of the SFS algorithm and the ITAE, the PID controller gains were fine-tuned. The SFS-PID strategy with ITAE cost function outperformed existing approaches in terms of rising time, adaption time, and overshoot when compared. DC motor position control was studied by Afra, Aidin, and Jafar [102] utilizing ICA (Imperialist Competitive Algorithm) [134] and ZN approaches [135]. Firefly Algorithm and PI controller were presented in [136] to regulate the DC motor, while a modified IWO (Invasive Weed Optimization) approach based on chaotic systems was proposed in [137] to tackle the problems of erroneous selection of standard deviation variables. Dual-line PSO-PID controllers have been developed to regulate the speed of DC motors [138]. During their investigation, they used the PSO algorithm twice. Algorithm selection of DC motor specs and PID controller parameter adjustments were the first two steps in the process.

According to the "no-free-lunch" theory [86], no metaheuristic technique is ideal for all situations and there are always improvements to be made. Hybrid methods outperform stand-alone alternatives due to their capacity to overcome individual algorithm constraints without diminishing their strength. PSO method is used to choose the ZN parameters for regulating the PID controller for controlling the BLDC motor speed. The best values of PID controller parameters were determined by combining PSO and ZN [87]. The BFA-PSO algorithms are used to find the optimal PID controller settings for a BLDC motor. To enhance the performance of the PSO algorithm, BFA is utilized [141]. The PSO algorithm was used to transfer social information while the BFA was used to find new solutions via exclusion and dispersion [143]. The previous study has shown that hybridized procedures outperform stand-alone strategies as it chooses the right algorithmic traits to improve on other strategy's flaws. The hybrid technique may also balance diversity and intensity. A combination of diversification and intensification guarantees that the algorithm hits all desirable areas within the search zone. To identify the best answer for a particular problem, these elements must be fine-tuned [88].

2.8 Identified Research gaps

- The benefits of metaheuristics have been frequently described in the literature, despite their lack of theoretical grounding. However, to fully utilize the metaheuristics, a few common concerns must be solved
- It's crucial to understand that the number of algorithm parameters has a direct impact on the algorithm's complexity and the number of parameter relations, which makes analysis more difficult.
- The relevance of tuning metaheuristics is well recognized in scientific literature since the effective application of metaheuristics to actual problems necessitates the discovery of a good starting parameter setting, which is a time-consuming and difficult operation.
- Much theoretical research on the study of landscapes (i.e., the topological structure across which search is carried out) of various optimization problems has revealed that not only different issues but also different instances of the same issue refer to different topologies.
- Another area of study that should be highlighted by researchers is the combination of deterministic and metaheuristic algorithms. Some researchers have looked at combining the two and the findings suggest that it leads to a faster convergence rate.
- Parallel computing is another recent technology that should be highlighted while using metaheuristics [89].

Measures to eliminate the research gap

As evolution develops, some selection mechanism based on the fitness landscape drives solutions to grow more and more similar to one another, resulting in solutions being more similar to one another. Selection is the process through which evolution moves ahead. Good solutions are selected based on their fitness, which is normally judged by objective criteria in most situations. For multi-agent populations to adapt and react to changes in the objective landscape, selection pressure is applied, causing the system to converge towards specific states or solutions. Nonlinear systems, as well as benchmark functions, have been studied using convergence analysis to provide a framework for investigating metaheuristic convergence and efficiency. Algorithms are chosen which has fewer parameters to adjust.

Chapter 3

Chaotic State of Matter Search with Elite Opposition Based Learning (CSMSEOBL): A Hybrid Metaheuristic Algorithm

3.1 Introduction

The origin of metaheuristic optimization algorithms is the emulation of different types of biological, physical, social and other natural phenomena. The process of finding new points in search space is exploration while the process of refining those searched points is exploitation which is required to improve the quality of the solution. Pure exploration enhances the opportunities of finding the new potential solution but decreases the precision whereas pure exploitation refines the existing solution but can be stuck in local optima so the success of any metaheuristic algorithm depends on the balance between exploration and exploitation in search space.

To solve the global optimization issue, the States of Matter Search (SMS) method is applied. When molecules come into contact with one other, they use the principle of thermal-energy mobility, which increases population diversity and prevents particles from being absorbed in local minima. Exploration and exploitation are balanced by SMS algorithm. The process of optimization is divided into three stages that simulate three states of matter: solid, liquid and gaseous phase. Molecule shows different performances for different states of matter. The process starts with the gas state which shows pure exploration i.e., molecules experience rigorous movement and collision after that algorithm alters the strength of exploration and exploitation, and the next state i.e., the liquid state is reached in which movement of molecules decreases and in the last pure exploitation state i.e., solid-state is reached in which the particles experience a strong bond that movement of molecules is completely constrained.

Chaos theory has been widely used for NP-hard problems [90] and many metaheuristic algorithms have been hybridized on chaos concept successfully like PSO [91][92], FA [93], GSA [94], BBO [95], Krill herd [96], Cuckoo Search [97] [98]and PSO-Krill herd [99]. Chaos is a volatile value layout for optimization problems that are highly sensitive to the initial condition. In this paper concept of chaos theory is hybridized with an SMS algorithm to define some random variables to stimulate the convergence of SMS.

Most of the metaheuristic algorithms face the problem of falling into local optima so to increase the exploration capability oppositional-based learning (OBL) was introduced by H. R. Tizhoosh. While associating the concept of OBL with other metaheuristic algorithm exploration capabilities can be enhanced by combining the benefit of global search capability with accelerated convergence rate [100]. In OBL to get a better solution current population and its opposite population are considered at the same point in time. There has been continuous research on this algorithm showing that the opposite population gives better results as compared to random numbers for a global solution and further it is investigated that Elite Oppositional Based Learning gives better results in terms of convergence time and exploration capability [101][102] as compared to OBL. Elite oppositional numbers are defined at the centre point of search space and these numbers are closer to the global optima as compared to the general opposite number.

Problems with optimization include the fact that there are various other, so-called "local" solutions, which may be closer to the achieved target value than the stated one. As these solutions are not seen as global solutions, they may lead to confinement which results in the existence of the local solution. Also, optimization methods face another major challenge: convergence. It is not required that an algorithm capable of avoiding local solutions also delivers superior convergence to global optima. As a result, the major obstacle of an optimization algorithm is to handle real-world issues with these two conflict trade-offs [103]. No optimization method can handle all optimization problems, according to the No Free Lunch theorem [104]. This information will encourage academics from a variety of fields to collaborate on the development of a novel optimization technique [105] [106] [103] or modify the existing algorithms [107] [98] [108] [109] [110] [111].

In the present study, the SMS algorithm is hybridized with Chaotic Maps and EOBL which results in an improved algorithm called “Chaotic State of Matter Search with Elite Opposition Based learning” and has been applied to 14 benchmark functions. The use of chaotic maps results in a stimulated convergence rate of SMS. Different types of one-dimensional chaotic maps are used to define some random variables of SMS. The benefits of using CSMSEOBL can be described as follows:(1) As chaotic maps are self-explanatory so do not enhance the complexity of SMS, (2) Increased performance and, (3) an increase in the exploration capability of CSMS.

3.2 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 is a method of searching for states of matter.

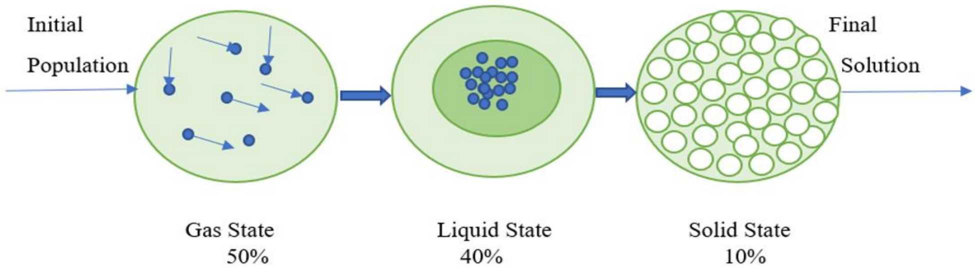


Figure3.1. The evolution process of the State of Matter Search Algorithm [80]

3.2.1 States of matter transition

States are defined by different phases that a matter can take. More broadly there are three states of matter. The gas state is the first stage of the SMS algorithm where the effect of intermolecular force is small because particles have enough kinetic energy so the distance between the molecules is high. The gas state represents 50% of the algorithm's total iterations, and particle movement is denoted by "α".The second state in the SMS algorithm is a liquid state where molecules have more restrictive intermolecular forces in comparison to the gas state.40% of the total no. of iterations are comprised of the liquid state and particle movement is represented by "β".The last stage is Solid-state where molecules are bonded with enough strong force so that molecules can't move freely hence solid has a definite shape and a force is required to change its shape. The remaining 10% of the total no. of iterations is comprised of solid-state and particle movement is represented by "γ". All parameters *"α", "β" and "γ"* are updated during each stage of the SMS algorithm and permit SMS to control the way molecules move in each stage (Figure 3.1).

3.2.2 Definition of molecule movement operators

As the algorithm progress, the position of search agents acting as molecules gets changed in n-dimensional space. The principle of motion of thermal energy is similar to the movement of molecules. Three factors are responsible for a molecule's movement i.e. (1) Attraction force (2) Collision and (3) some random phenomenon. These all factors (or behaviours) have been executed by defining different operators i.e., direction vector, collision and the random position operators, these all represent the operation of actual physics laws [19].

3.2.3 Direction Vector

The direction vector operator represents the change in molecule position as the process progress. An n-dimensional direction vector $\mathbf{d}_i$ is assigned to each n-dimensional molecule $\mathbf{p}_i$ from the population, **P**. This direction vector controls the movement of a particle. The range of direction vector is chosen randomly within [-1,1]. Many attraction forces are experimented with by the molecule with the system evolution. Molecules are moved towards the best position so far to execute the attraction phenomenon. For each molecule new direction vector is given by:

$$\boldsymbol{d}_i^{k+1} = \boldsymbol{d}_i^{k}\left(1 - \frac{k}{gen}\right)0.5 + \boldsymbol{a}_i \tag{3.1}$$

$$\boldsymbol{a}_i = \frac{(\boldsymbol{p}_{best} - \boldsymbol{p}_i)}{\|\boldsymbol{p}_{best} - \boldsymbol{p}_i\|}$$

where,

$\mathbf{a}_i$-Attraction unitary vector

$\mathbf{p}_{best}$ - The best molecule in population P

$\mathbf{p}_i$ – Molecule i of population P

k-Current iteration number

gen-Total number of iterations

As the evolution process progresses the importance of the previous direction decreases and this algorithm gets less susceptible to early convergence also particle search neighborhood thoroughly.

Velocity v_i of each molecule is given by

$$\boldsymbol{v}_i = \boldsymbol{d}_i * v_{st} \tag{3.2}$$

where,

v_{st} - Starting velocity

$$v_{st} = \frac{\sum_{j=1}^{n}(b_j^h - b_j^l)}{n} * \beta \tag{3.3}$$

b_j^h-Upper bound of j parameter

b_j^l- Lower bound of j parameter

$\beta \in [0,1]$

Position update equation for every molecule is given by

$$p_{i,j}^{k+1} = p_{i,j}^{k} + v_{i,j} * rand(0,1) * \rho * (b_j^h - b_j^l) \tag{3.4}$$

where $0.5 \leq \rho \leq 1$

Collision

Whenever molecules interact with each other and the distance between two molecules is smaller than the present value collision occurs. If $\|\mathbf{p}_i - \mathbf{p}_q\| <$ r, a collision occurs between "i" and "q". When a collision occurs then particles are updated according to $\mathbf{d}_i = \mathbf{d}_q$ and vice-versa. The radius of collision is given by

$$r = \frac{\sum_{j=1}^{n}(b_j^h - b_j^l)}{n} * \alpha \tag{3.5}$$

where $\alpha \in [0,1]$

As molecules start to get closer collision operator forces the molecules to change their position and this controls the preterm convergence of the algorithm.

Random Position

The random position operator allows the change in molecule position by using a feasible criterion within the search space.

$$p_{i,j}^{k+1} = \begin{cases} b_j^l + rand * (b_j^h - b_j^l) \;\; with \;\; p(H) \\ p_{i,j}^{k+1} with \;\; p(1 - H\,) \end{cases} \tag{3.6}$$

where "rand" is a random number varying from 0 to 1.

If R is lesser than P a random position of the molecule is generated otherwise there is no alteration in the molecule.

Element updation

If the current best molecule is compared "$\mathbf{p}_{best}^{k}$" with previous best individual "$\mathbf{p}_{best}^{k-1}$" and $\mathbf{p}_{best}^{k}$ is better than $\mathbf{p}_{best}^{k-1}$ than $\mathbf{p}^{best}$is updated with "$\mathbf{p}_{best}^{k}$" otherwise no change. The process repeats until all the iterations are not over and the final result will be evaluated in the last iteration step.

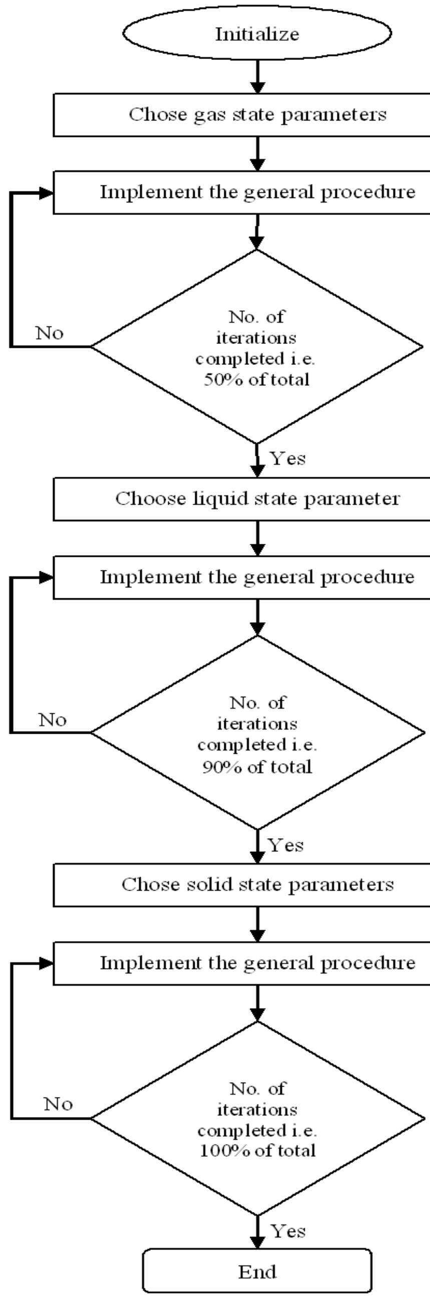


Figure 3.2. Flowchart of SMS Algorithm

Here General procedure has been described in Section 3.2 already (Figure 3.2).

3.3 Chaotic maps

Chaotic maps form a different type of optimization algorithm i.e., chaotic optimization algorithm (COA) based on randomization. It uses chaotic variables instead of random variables (which are generally used in optimization algorithms) main properties associated with chaos concepts are the non-recurrence of parameter value and extent approach to randomness, which results in increased speed of convergence [35]. The dynamic property guarantees that algorithms create a diversity of solutions and explore diverse landscapes in the search space, while ergodicity and non-recurrence speed up the search. Chaotic optimization improves the diversity of movement patterns while simultaneously speeding up the process. Because chaos theory shares the same features as metaheuristic algorithms, that's why it's been integrated with them. A bounded nonlinear system having ergodic and stochastic characteristics is termed chaos [112]. It is quite responsive to the starting state and specifications. Ten different types of chaotic maps have been described in Table 3.1 and the realization of these maps has been shown in Figure 3.3.

Table 3.1 Chaotic Maps

Map No.	**Name**	**Definition Of Chaotic Map**	**Range**
1	Chebyshev	$x_{k+1} = cos(k\, cos^{-1}(x_k))$	(0,1)
2	Circle	$x_{i+1} = mod\left(x_i + b - \left(\frac{a}{2\pi}\right) sin(2\pi x_k)\, , 1\right)$ $a = 0.5, b = 0.2$	(0,1)
3	Gauss	$x_{i+1} = \begin{cases} 1 & x_i = 0 \\ \frac{1}{mod(x_{i,}1)}, & otherwise \end{cases}$	(0,1)
4	Iterative	$x_{k+1} = sin\left(\frac{a\pi}{x_k}\right), a \in (0.1)$	(0,1)
5	Logistic	$x_{i+1} = ax_i(1 - x_i), a = 4$	(0,1)
6	Sine	$\frac{a}{4} sin\,(\pi x_k)\, ; 1 < a < 4$	(0,1)
7	Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 1.07$	(0,1)
8	Sinusoidal	$x_{i+1} = ax_i^2\, sin(\pi x_i)\, , a = 2.3$	(0,1)

Map No.	Name	Definition Of Chaotic Map	Range
9	Piecewise	$\frac{x_k}{P}$; $0 \leq x_k < P$ $\frac{x_{k-} - P}{0.5 - P}$; $P \leq x_k < 0.5$ $\frac{1 - P - x_{k-}}{0.5 - P}$; $0.5 \leq x_k < 1 - P$ $\frac{1 - x_{k-}}{P}$; $1 - P \leq x_k < 1$	(0,1)
10	Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} & x_i < 0.7 \\ \frac{10(1 - x_i)}{3} & x_i \geq 7 \end{cases}$	(0,1)

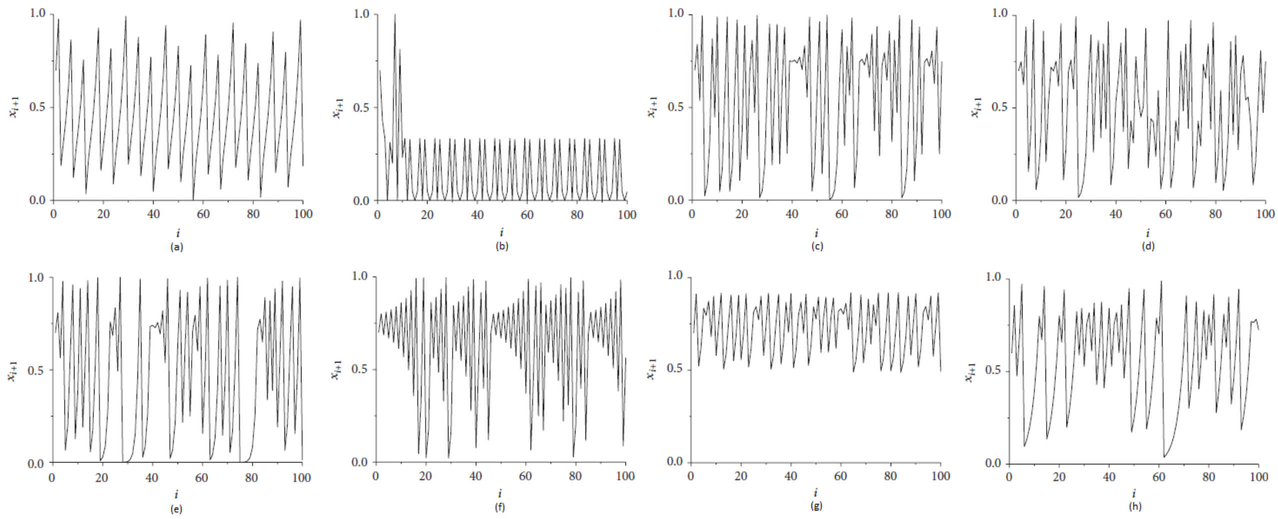


Figure 3.3. Different types of Chaotic maps: a. circle map b. gauss map c. logistic map d. piecewise map e. sine map f. singer map g. sinusoidal map h. tent map

3.4 Elite Oppositional Based Learning Algorithm

In ancient Chinese philosophy, the main opposing notion was initially represented in the Yin-Yang symbol (Figure 3.4) [113]. This sign represents the notion of duality, in which black and white represent Yin (responsive, femininity, gloomy, inactive power) and Yang (productive, manliness, dazzling, dynamic power). In addition, Classical Greek components of nature arrangements (Figure 3.5) depicted opposing notions such as fire (warm and dry) vs. water (cold and moist) and earth (cold and dry) vs.

air (warm and moist). Nature components and their mirror images are represented by the terms cold, hot, wet and dry. The opposing notion appears to be used to convey the concept of numerous things or circumstances in the actual world. Employing the opposing idea simplifies the description of many things. East, west, south and north are paired opposites that cannot be defined independently and can only be described in terms of each other. As a result, the computational opposition idea [21] was influenced by the real-world opposition notion [114].

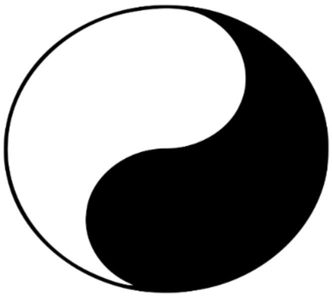


Figure 3.4. The Yin-Yang symbol first stated the notion of opposites [115]

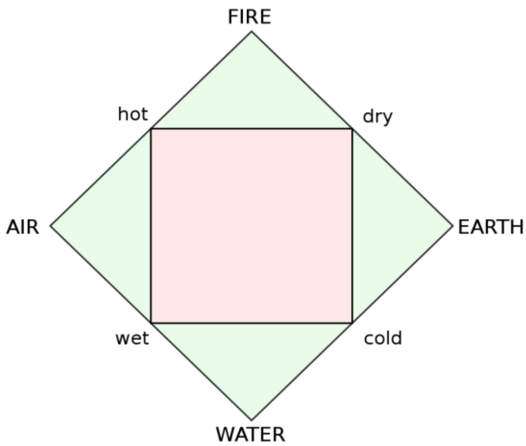


Figure 3.5. The Greek classical components are used to describe natural phenomena [115].

EOBL plays a significant role in searching for global optima because it enhances the exploration ability of the SMS algorithm by introducing a new population. OBL is the pre-requisite for EOBL so first OBL is explained. The fundamental premise of OBL is that it generates a solution that is opposed to the present solution, and then both solutions are assessed at the same time, with the superior solution moving on to the next iteration [102].

Let $x = \{x_1, x_2, \dots \dots .. x_j\}$ is a point in the current population and j is the dimension of search space, $x_j \in [a_j, b_j]$ where, $a_j = \min\{x_{ij}\}$ and $b_j = \max\{x_{ij}\}$. The opposite point is defined as follows:

$$\widetilde{x_j} = a_j + b_j - x_j \quad (3.7)$$

In OBL there are many chances that the generated solution is not better than the current search space to find the global optima so a new strategy is introduced which is EOBL. In this strategy elite individual in the current population is $x_e = \{x_{e1}, x_{e2}, \ldots\ldots x_{ej}\}$, the elite oppositional solution is given by

$$\tilde{x}_{i,j} = \rho * \left(da_j + db_j\right) - x_{e,j} \tag{3.8}$$

where,

$i = 1, 2, \ldots. P;$

P - population size

ρ – Generalized coefficient

da_j, db_j- Dynamic bounds can be calculated as:

$$da_j = \min\,(x_{i,j})$$

$$db_j = \max\,(x_{i,j})$$

In EOBL dynamic bounds are used instead of fixed bounds to secure the search space from shortening. If $\tilde{x}_{i,j}$ crosses its dynamic bound it can be reset by using the following equation:

$$\tilde{x}_{i,j} = rand(da_j, db_j) \tag{3.9}$$

The advantage of EOBL is that it can evaluate the elite population and the current population at the same point in time which results in population diversification and further enhance the global exploration capability of the metaheuristic algorithm.

3.5 CSMSEOBL - Proposed Hybrid Algorithm

In the field of evolutionary algorithms, SMS's algorithm is based on nature and used to solve MIMO-type global optimization problems. The mechanism is based on the movement of heat energy. Three states of matter i.e., solid, liquid, and gas are simulated in this algorithm, and state-by-state, the ratio of exploration to exploitation is variable. The algorithm begins with the gas state which is purely exploration, then after reforming the exploration and exploitation ratio it reaches a liquid state in which a moderate transition takes place between exploration and exploitation and this reforming is continued till the solid state i.e., pure exploitation is reached. This whole process results in the enhancement of population diversity and simultaneously escapes the particles to concentrate within local minima[80].

A hybrid metaheuristic approach is used to enhance the balance between the exploration and exploitation capability of the existing algorithm along with an accelerated convergence rate. The benefits of all three algorithms are combined to form this hybrid algorithm. Chaotic Maps are used to calculate the random variable of the SMS algorithm and increase the exploitation capability. Further, the inclusion of EOBL enhances the exploration capability of the SMS Algorithm. CSMSEOBL algorithm is shown in Figure 3.6.

The complete CSMSEOBL Algorithm can be divided into four stages:

Stage 1: Initialization state & general procedure:

- Find the best element from population P

$$P^{best} \in \{P\} | \, f(P^{best}) = max \, \{f(P_1), f(P_2), \ldots \ldots \ldots f(P_{N_P})\} \tag{3.10}$$

And at the same time opposite population is generated by using elite opposition-based learning.

Calculate initial velocity magnitude

$$v_{st} = \frac{\sum_{j=1}^{n}(b_j^h - b_j^l)}{n} * \beta \tag{3.11}$$

where, $\boldsymbol{b_j^h}$ is the upper bound of the j parameter, $\boldsymbol{b_j^l}$ is lower bound of the j parameter, $\boldsymbol{\beta}$ is a factor ranging [0,1]

Update the direction vector to control the movement of the particle

$$d_i^{k+1} = d_i^k \left(1 - \frac{k}{gen}\right) 0.5 + a_i \tag{3.12}$$

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

where,$'a_{i}'$ attraction unitary vector, $'p_{best}'$ is the best molecule in population 'P', $'p_i'$ is molecule 'i' of population 'P','k' is the current iteration number; 'gen' is the total number of iterations.

Calculate velocity $'v_i'$ of each molecule

$$v_i = d_i * v_{st} \tag{3.13}$$

- Calculate collision radius $'r'$ and $0 \leq \alpha \leq 1$

$$r = \frac{\sum_{j=1}^{n}(b_j^h - b_j^l)}{n} * \alpha \tag{3.14}$$

- Then update the Position of each molecule, which is given by (H is a threshold limit)

$$p_{i,j}^{k+1} = p_{i,j}^{k} + v_{i,j} * rand(0,1) * \rho * (b_j^h - b_j^l); if rand \leq H \quad (3.15)$$

$$p_{i,j}^{k+1} = p_{i,j}^{k} \; ; if \; rand > H$$

Random variable rand is updated by using different chaotic maps and the best chaotic maps are selected for a position update.

Stage 2: Gas state

- Set the parameters for the gas state: $\rho \in [0.8,1], \beta = 0.8, \alpha = 0.8$ & $H = 0.9$.
- Apply the general procedure as described in Stage 1.
- If the no. of iteration=50% of the total no. of iterations then the process shifted to a liquid state otherwise the general procedure is repeated.

Stage 3: Liquid State

- Set the parameters for the liquid state: $\rho \in [0.3,0.6], \; \beta = 0.4, \; \alpha = 0.2$ & $H = 0.2$.
- Apply the general procedure as described in Stage 1.
- If no. of iteration=90% of the total no. of iterations then the process shifted to solid-state otherwise the general procedure is repeated.

Stage 4: Solid State

- Set the parameters for solid-state: $\rho \in [0.0,0.1], \; \beta = 0.1, \alpha = 0$ & $H = 0$.
- Apply the general procedure as described in Stage 1.
- If the total no. of iteration=100% then the process is finished otherwise the general procedure is repeated.

Pseudo Code for CSMSEOBL

Begin: Define fitness function $f(x)$, Population P, $X = \{x_1, x_2, \dots \dots .., x_D\}$

Result: The optimal solution x*

1. Initialization: Initialize the parameter of the SMS algorithm for gas state i.e., *α, β, ρ and* H, and also initialize the dynamic boundary of search space.

2. While the termination criterion is not satisfied do

3. Make use of the EOBL approach to update the current population (Equations (3.5),(3.6),(3.7),(3.8) and (3.9)).

   For each x ϵ P do

4. Update all random variables by using chaotic maps

$$p_{i,j}^{k+1} = p_{i,j}^{k} + v_{i,j} * C(t) * \rho * (b_j^h - b_j^l) \tag{3.16}$$

   Where $C(t)$ is the value of the chaotic map.

5. Calculate initial velocity and collision radius for gas state by using Equation 3.13. and 3.14 respectively.

6. Compute the new molecules by using the direction vector of Equation 3.12

7. Solve collision by using collision operator

8. Generate a new random position by using the collision operator of Equation 3.15

9. Check if the total no. of iterations completed is ≤ 50% of the total number of iterations

10. Go to the liquid state and repeat steps 6, 7, 8 and 9.

    Else

11. Check if the total no. of iterations completed ≤ 90% of the total number of iterations

12. Go to solid-state and repeat steps no.6, 7,8 and 9.

13. If 100% of total iterations completed

14. Update x with x^*.

    End if

    End for

    End while

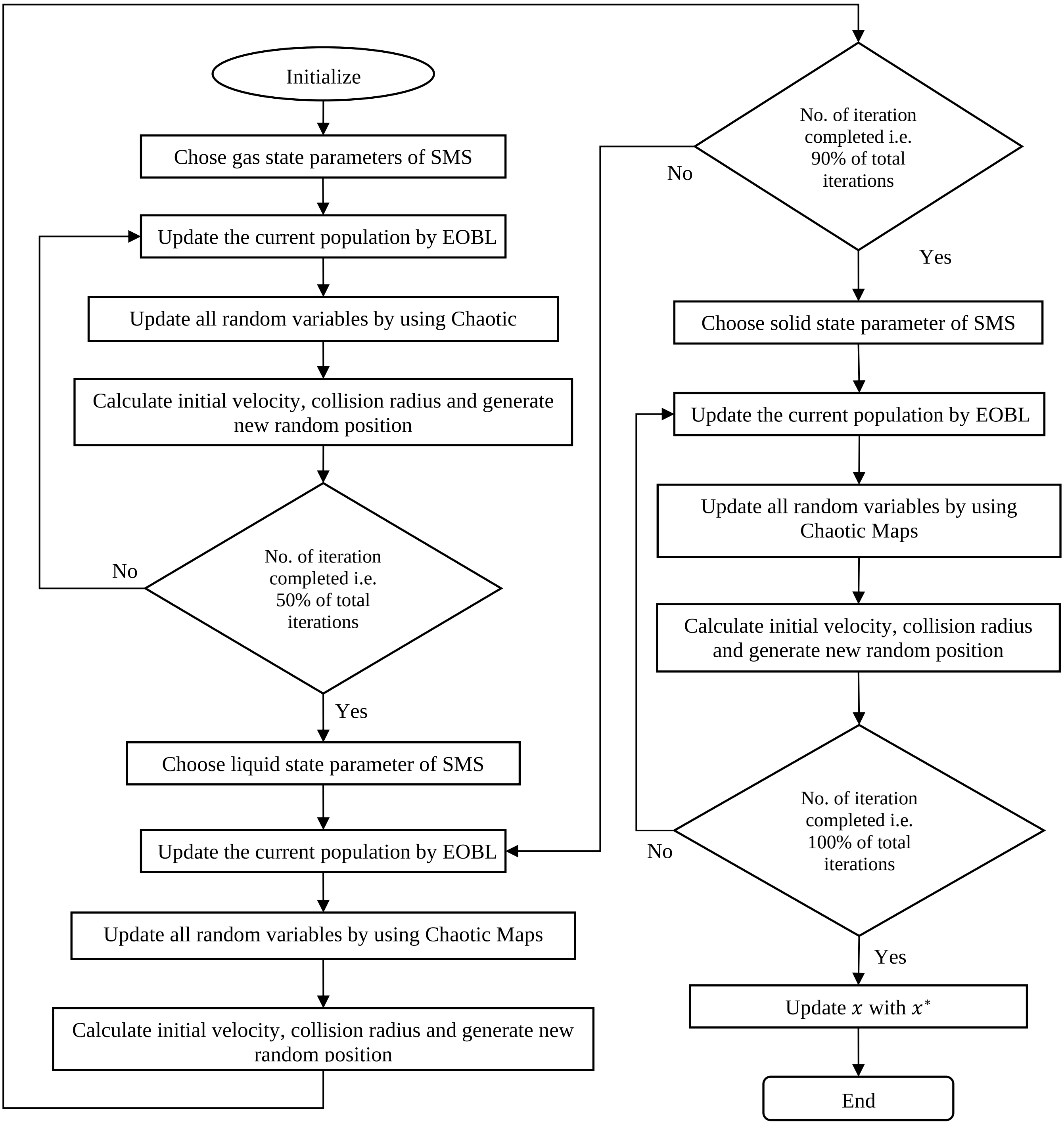


Figure 3.6. Flow Chart of CSMSEOBL Algorithm

3.6 Benchmark Functions

Fundamentally, benchmark test functions are optimization issues given as arithmetic expressions. These functions are optimized using a set of best-fit parameter values that aid in achieving the optimal solution, where D denotes the problem dimensions. The optimum answer is concealed among a huge number of sub-optimal alternatives scattered throughout a problem landscape with a variety of hills and valleys. A variety of test functions are available, including unimodal and multimodal, regular and irregular, separable and non-separable and so on. To add complexity to the optimization environment, these functions are frequently rotated or moved and they are also employed with a larger spectrum of dimensions (from 10D to 1000D). This leaves a large gap for the research community who wishes to test any updated version of an existing metaheuristic algorithm or a completely new method on benchmark test functions. Each optimization method, including metaheuristic algorithms, strives to discover the optimal answer as soon as possible (though this is not always guaranteed). The global searchability and local convergence ability of any optimization algorithm are used to determine its efficiency. Better global searchability algorithms are difficult to trap in sub-optimal regions. Simultaneously, metaheuristics with quick convergence ability make it difficult to overlook any optimal solution in the surrounding [116].

The test functions may be classified into the following groups based on the features (modality, separability and dimensionality) that form the problem's landscape:

3.6.1 Modality

The quantity of spikes in the problem landscape is defined by modality. Local and global minima are formed by these spikes.

a. Unimodal Functions: These functions feature a single valley and a single global minimum at which the optimal solution may be found. These functions are thought to be simple to solve, however moving and rotating them increases the difficulty. These functions can be used to assess metaheuristic methods for evaluating local search efficiency.

b. Multimodal Functions: These functions have many outcomes, but the real global best is only one. There are numerous local minimum sites for such functions, but only one real global minimum. As a result, every metaheuristic algorithm must traverse the whole territory to discover the real global optimum answer. These functions are tough to solve and are useful for assessing an algorithm's global search efficiency.

3.6.2 Separability

Separability relates to the optimization of a function's variables. In both unimodal and multimodal functions, separable and non-separable functions exist.

a. Separable Functions: When a function is separable, each variable may be optimized independently. The purposes of this category are clear.

b. Non-Separable Functions: A non-separable function is one in which all of its variables are tightly coupled and cannot be optimized independently. Solving such functions may be difficult.

3.6.3 Dimensionality

The size of the search space is defined by this parameter. The higher the dimension, the broader the terrain and the more sub-optimal places there are. Small-dimensional functions are simple to solve and most optimization techniques operate well on these functions. Functions must, however, be high dimensional for real performance assessments [117].

Some benchmark functions used in this thesis to analyze the performance of the proposed hybrid metaheuristic algorithm are shown in Table 3.2.

Table 3.2 Benchmark Functions used for validation of the proposed algorithm

S.No.	Function	Definition	Lower bound	Upper bound	Modality	Global Minima	Dimension	Characteristics
1	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	-5.12	5.12	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Separable, Scalable
2	Schwefel 2.21	$f_2(x) = max_i\{\lvert x_i\rvert, 1 \leq i \leq n\}$	-100	100	unimodal	f(0, · · · , 0)=0	n	Continuous, Non-Differentiable, Separable, Scalable
3	Quartic with noise	$f_3(x) = \sum_{i=1}^{n}(i.x_i^4 + rand(0,1))$	-1.28	-1.28	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Separable, Scalable

S.No.	Function	Definition	Lower bound	Upper bound	Modality	Global Minima	Dimension	Characteristics
4	Schwefel 2.26	$f_4(x) = 418.9829 * n - \sum_{i=1}^{n} x_i \sin(\sqrt{x_i})$	-500	500	multi modal	f (±[π(0.5 + k)]^2)=-418.983	n	Continuous, Differentiable, Separable, Scalable
5	Zakharov	$f_5(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$	-5	10	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Non-Separable, Scalable
6	Ackley	$f_6(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e^1$	-32.768	32.768	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Non-separable, Scalable
7	Schwefel 2.22	$f_7(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	-100	100	unimodal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Non-Separable, Scalable,
8	Alpine	$f_8(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	0	10	multi modal	f(0, · · · , 0)=0	n	Separable ,Non-Differentiable,Continuous,
9	Salomon function	$f_9(x) = 1 - \cos(2\pi \sum_{i=1}^{n} x_i^2) + 0.1 \sum_{i=1}^{n} x_i^2$	-100	100	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Non-Separable, Scalable

S.No.	Function	Definition	Lower bound	Upper bound	Modality	Global Minima	Dimension	Characteristics
10	Goldstein and Price	$f_{10}(x,y) = [1 + (x+y+1)^2(19-14x+3x^2-14y+6xy+3y^2)][30+(2x-3y)^2(18-32x+12x^2+4y-36xy+27y^2)]$	-2	2	multi modal	f(0, −1)=3	2	Continuous, Differentiable, Non-separable, Non Scalable
11	Powell Sum Function	$f_{11}(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	-4	5	unimodal	f(0)=0	n	Continuous, Differentiable, Separable Scalable
12	Drop wave function	$f_{12}(x) = -\frac{1+\cos(12\sqrt{x^2+y^2})}{(0.5(x^2+y^2)+2)}$	-5.12	5.12	multi modal	f(0.0)=-1	2	continuous
13	Easom	$f_{13}(x,y) = -\cos(x_1)\cos x_2)\exp(-(x-\pi)^2-(y-\pi)^2)$	-100	100	multi modal	f(π, π)=-1	2	Continuous, Differentiable, Separable, Non-Scalable,
14	Griewank	$f_{14}(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} (\cos\frac{x_i}{\sqrt{i}}) + 1$	-600	600	multi modal	f(0, · · · , 0)=0	n	Continuous, Differentiable, Non-Separable, Scalable,
15	Rastrigin	$f_{15}(x) = 10.n + \sum_{i=1}^{n} x_i^2 - 10\cos 2\pi x_i$	-30	30	multi modal	f(0, · · · , 0)=0	n	Separable, continuous, differentiable

3.7 Simulation Results Analysis

It has been proved that no one search strategy is the best on average for all problems when specified assumptions are made about the problem [118]. The CSMSEOBL technique's optimization potential is tested using a wide range of typical benchmark functions without a predetermined conclusion for specific issues.

The efficiency and the efficacy of the proposed hybrid algorithm have been evaluated on 15 benchmark functions and obtained results are compared with other metaheuristic optimization techniques. To validate the performance of the proposed CSMSEOBL algorithm, SMS algorithm [19], Grey Wolf Optimization [119], Krill Herd [79] and Particle Swarm Optimization[16] are used for comparison. These benchmark functions can be divided into two sections - unimodal functions and multimodal functions [84]. Generally, to evaluate exploitation unimodal functions are used and for evaluation of exploration multimodal functions are used. A detailed description of these benchmark functions is given in [120]. As there are multiple local minima in multimodal benchmark functions that's why they are a challenge for getting good optimization. In the case of the multimodal benchmark function, any metaheuristic algorithm can locate global optima and escape from local optima. The results obtained by this hybrid algorithm have been compared to SMS algorithms. For both algorithms, the population has been set to 50. The maximum number of iterations has been set to 1000 for all benchmark functions. The results obtained by each algorithm have been averaged for 30 runs and the dimension is also set to 30. Also, the proposed hybrid algorithm has been run for different chaotic maps which have been detailed in Table 3.1. The purpose of implementing the algorithm with different chaotic maps is to enhance the quality of the solution.

The parameter setting of the SMS, PSO, GWO[108], KH (Krill Herd Optimization) and CSMSEOBL algorithm is shown in Table 3.3.

The whole experimental procedure and described methods are implemented in Matlab R2016a in Microsoft Windows 10 environment with an i5 processor.

Table 3.3 Parameter setting for different metaheuristic Algorithms

Name of Algorithm	**Parameters Values**	**Name of Algorithm**	**Parameters Values**
SMS	β= [0.8, 0.4, 0.1] α= [0.8, 0.2, 0] H = [0.9, 0.2, 0] Population size P = 50 Dimension, D=30 No. of Iterations=1000	**GWO**	Search agents=80 Control parameter ($\vec{a}$) [2, 0] Population size P = 50 Dimension, D=30 No. of Iterations=1000
Name of Algorithm	**Parameters Values**	**Name of Algorithm**	**Parameters Values**
CSMSEOBL	Total no. of Chaotic Maps=10 β= [0.8, 0.4, 0.1] α= [0.8, 0.2, 0] H = [0.9, 0.2, 0] Population size P = 50 Dimension, D=30 No. of Iterations=1000	**PSO**	Inertia coefficient= 0.75 Cognitive and social coefficient = 1.8,2 Population size P = 50 Dimension, D=30 No. of Iterations=1000
KH	Foraging speed V_f = 0.02 maximum diffusion speed = 0.008 maximum induced speed = 0.02 Population size P = 50 Dimension, D=30 No. of Iterations=1000		

3.7.1 Performance of CSMSEOBL with different chaotic maps

SMS algorithm hybridized with EOBL is run for different 10 Chaotic Maps which have been given in Table 3.1 to improve the quality of the solution.14 benchmark functions are run for 10 different chaotic maps and 1000 iterations are carried out for each. All 10 chaotic maps have been marked as C1, C2…C10.

The benchmark functions are used to evaluate the algorithm's capability to converge quickly, move out of local optima and obtain a high number of local optima while avoiding premature convergence. Table 3.4, Table 3.5 and Table 3.6 shows the median, mean and standard deviation respectively achieved by CSMSEOBL algorithms.

Table 3.4 Median values obtained for 10 different chaotic maps

	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
f1	0.00045	0.00359	0.0039	0.0052	0.0024	0.0072	0.0030	**0.0000**	0.0002	0.0059
f2	0.00021	0.00009	0.00006	0.0001	0.00009	**0.00005**	0.0000	0.0001	0.06905	0.00006
f3	0.00735	0.00263	0.00268	0.0070	**0.00224**	0.00251	0.0026	0.0026	0.02980	0.00281
f4	9519.90	9497.34	9435.18	9509.4	9486.75	9530.79	9484.3	**9385.2**	9426.09	9512.8
f5	0.26087	0.00000	0.00000	0.2673	**0.00000**	0.00015	0.0000	0.0000	5.39434	0.00000
f6	0.20485	0.04905	0.05504	0.1605	0.04052	0.06883	0.0466	0.0262	**0.00557**	0.0654
f7	0.78382	0.32845	0.35851	0.7790	0.28371	0.43178	0.2800	0.1887	**0.18787**	0.3965
f8	0.03644	0.01995	0.02090	0.0385	0.01657	0.02513	0.0193	**0.0152**	0.14549	0.0227
f9	0.19987	0.09987	0.09987	0.1998	0.09987	0.09987	**0.0998**	**0.0998**	0.19987	0.0998
f10	3.00138	3.00000	3.00000	3.0005	3.00000	3.00000	3.0000	3.0000	3.00000	3.0000
f11	0.09987	0.09987	0.09987	0.0998	0.09987	0.09987	0.0998	0.0998	0.09987	0.0998
f12	0.01983	0.00301	0.00708	0.0092	0.00264	0.00738	0.0050	**0.0004**	0.00372	0.0069
f13	0.86100	0.93625	-0.93625	0.9362	0.93625	0.93625	0.9362	**-0.936**	-0.7857	0.93625
f14	0.98688	0.99307	-0.98660	0.9781	0.98956	0.99115	0.9920	**-0.991**	-0.9871	0.98792
f15	56.76709	3.95608	4.485292	60.0936	4.918964	6.719794	3.4315	**3.2194**	120.747	4.50132

Table 3.5 Mean values obtained for 10 different chaotic maps

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
f1	0.0057	0.0033	0.00399	0.007	0.002	0.0067	0.0028	**0.0007**	0.0037	0.0059
f2	0.0004	0.00009	0.00007	0.0005	0.000	**0.0000**	0.00009	0.00011	0.09767	0.0000
f3	0.0077	0.00267	0.00262	0.0074	**0.002**	0.0027	0.00287	0.00268	0.02924	0.0028
f4	9544.6	9457.03	9443.63	9488.7	9458.	9462.2	9484.97	9441.74	**9418.61**	9492.5
f5	0.5651	0.00007	0.00013	0.3388	0.000	0.0004	0.00013	**0.00000**	6.51608	0.0002
f6	0.2024	0.05004	0.05428	0.1575	0.039	0.0682	0.04710	**0.02082**	0.09009	0.0656
f7	0.7622	0.34909	0.35650	0.6488	0.296	0.4276	0.29639	**0.19013**	0.30756	0.3966
f8	0.0367	0.02034	0.02034	0.0389	0.016	0.0251	0.01910	**0.01508**	0.22704	0.0228
f9	0.1798	0.09987	0.09987	0.1732	0.099	0.0998	0.09987	**0.09987**	0.19654	0.0998
f10	3.1751	3.00000	3.00000	3.0287	3.000	3.0000	3.00000	3.00000	5.60030	3.0000
f11	0.0998	0.09987	0.09987	0.0998	0.099	0.09987	0.09987	0.09987	0.14322	0.09987
f12	0.0874	0.00458	0.00589	0.0609	0.0036	0.00786	0.00518	**0.00169**	1.44370	0.00658
f13	0.8610	0.93625	-0.9312	0.8660	0.9362	-0.9362	-0.9362	0.9362	-0.8003	0.93625
f14	-0.8063	0.84276	-0.7466	0.7059	0.8860	-0.8855	**-0.8973**	0.79415	-0.7582	0.79131
f15	61.884	5.18796	4.80082	66.210	5.7212	6.16206	4.12283	3.68039	86.8361	5.02830

Table 3.6 Standard deviation obtained for 10 different chaotic maps

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
f1	0.00710	0.00168	0.00136	0.00826	0.00145	0.00189	0.00160	**0.00089**	0.00893	0.00229
f2	0.00101	0.00002	0.00002	0.00214	0.00002	**0.00001**	0.00003	0.00002	0.10117	0.00002
f3	0.00401	0.00118	**0.00102**	0.00390	0.00121	0.00119	0.00140	0.00125	0.02653	0.00144
f4	341.722	369.312	334.3216	339.278	382.169	301.965	332.306	383.403	**289.564**	363.392
f5	0.58514	0.00022	0.00047	0.31977	0.00012	0.00076	0.00044	**0.00000**	6.43639	0.00069
f6	0.14413	**0.00804**	0.00843	0.15683	0.01115	0.00893	0.01074	0.01371	0.15319	0.01161
f7	0.50859	0.09057	0.07453	0.54768	0.07625	0.07451	0.06901	**0.05079**	0.61548	0.05876
f8	0.00582	0.00300	0.00308	0.00678	0.00275	0.00308	0.00324	**0.00259**	0.36501	0.00343
f9	0.04842	0.00000	0.00000	0.06397	0.00000	0.00000	0.00000	**0.00000**	0.09279	0.00000
f10	0.59523	0.00000	0.00000	0.07609	0.00000	0.00000	0.00000	0.00000	6.02433	0.00000
f11	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.05041	0.00000
f12	0.19630	0.00373	0.00446	0.10701	0.00352	0.00405	0.00474	**0.00259**	2.77190	0.00511
f13	0.07653	0.00000	0.02748	0.07636	0.00000	**0.00000**	0.00000	0.00000	0.14229	0.00000
f14	0.30202	0.33230	0.36480	0.37758	0.28739	**0.27077**	0.27861	0.34848	0.37556	0.37079
f15	22.8060	3.58225	3.162544	32.1523	3.8948	3.07006	2.56390	1.82970	60.7596	2.57856

3.7.2 Performance Comparison

The proposed approach is compared to other optimization algorithms on unimodal, multimodal and fixed dimension multimodal benchmark functions to demonstrate its efficacy. Tables 3.4, 3.5 and 3.6 provide the mean, median and standard deviation of all 15 benchmark functions for 10 different Chaotic maps respectively.

To evaluate the intensification capacity of a metaheuristic algorithm, unimodal test functions f1, f2, f5, and f12 are utilized. The best-normalized value derived from the above-mentioned methods on unimodal test functions is shown in Table 3.7. In terms of best-normalized value, CSMSEOBL is the best optimizer for f1, f2, f5, and f12 test functions, whereas SMS is the second most successful optimizer.

Multimodal test functions can be used to assess an optimization algorithm's diversification. Table 3.7 show that CSMSEOBL can identify the best response for benchmark functions (f1-f3, f5-f12, f14). For f4 krill herd and f15, GWO gives the best-normalized values.

Table 3.7 Best-Normalized Values for different metaheuristic algorithms

	SMS	**GWO**	**KH**	**PSO**	**CSMSEOBL**
f1	0.001613	6.9187	1.1964	6.0651	**6.54E-08**
f2	0.055017	1.7831	0.0061	87.6269	**2.01E-05**
f3	0.015084	0.002	9.7403	56.707	**0.000678**
f4	9112.667	6.1499	**1.2728**	7.6971	8746.403
f5	0.026685	4.8419	6.7275	589.78	**9.81E-08**
f6	0.033842	1.5099	0.987	0.39	**0.000404**
f7	0.164469	1.2559	1.1346	3.7472	**0.069533**
f8	0.012369	1.7676	5.9928	16.62	**0.008682**
f9	0.299873	0.0999	7.1997	23.93	**0.099873**
f10	0.5843	0.954	1	0.974	**0.00764**
f11	0.099184	3.3981	1.678	5.0433	**3.17E-09**

	SMS	GWO	KH	PSO	CSMSEOBL
f12	-0.6195	-0.9362	-0.2297	-0.00189	**-0.93625**
f13	-1	-1	-1	-0.8227	-1
f14	9.43E-05	0.000126	0.989	0.973	**8.15E-10**
f15	4.744741	**0.029014**	0.865	0.998	1.159739

3.7.3 Statistical Analysis

Aside from the fundamental statistical analysis, the Wilcoxon rank test is run at a significance level of 5% [80]. The CSMSEOBL algorithm's superiority is demonstrated by the p-values of less than 0.05. Table 3.8 summarises the Wilcoxon rank-sum test findings. CSMSEOBL outperforms other optimization methods available in the literature, according to the findings.

Wilcoxon rank test which is a nonparametric test has been conducted on results of the proposed hybrid algorithm and SMS algorithm for 14 benchmark functions given in Table 3.2 and results are shown in Table 3.8.

Table 3.8 p-values obtained by Wilcoxon's test that compare SMS vs. CSMSEOBL

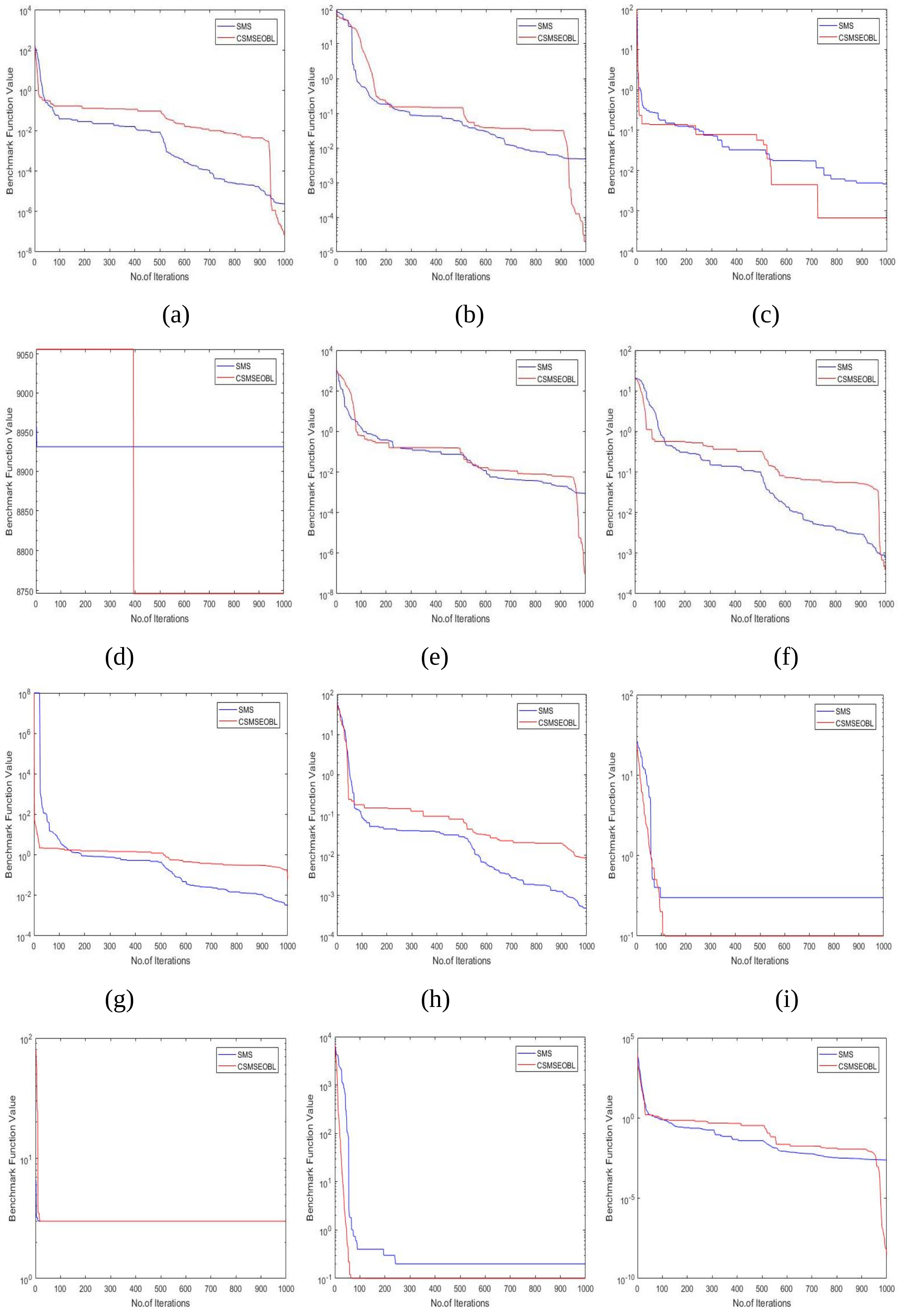
Function	**p-values**	**Function**	**p-values**
f1	0.662734758176107e-4	**f8**	2.98034208171800e-11
f2	3.01985935916215e-11	**f9**	2.98034208171800e-11
f3	3.68972585398101e-11	**f10**	2.99534031311320e-11
f4	6.202653652039733e-04	**f11**	3.01985935916215e-11
f5	3.01985935916215e-11	**f12**	3.52005762606802e-07
f6	9.83289055492186e-08	**f13**	3.01418492280728e-11
f7	3.01985935916215e-11	**f14**	3.55859151949451e-06
f15	1.74366896542643e-10		

3.7.4 Convergence Analysis

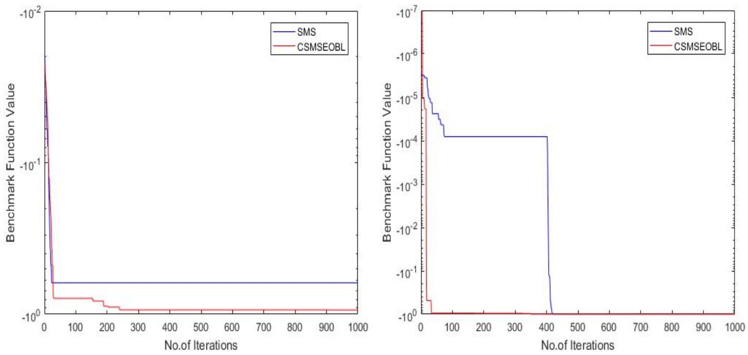
The convergence curve analysis is examined to have a better knowledge of the CSMSEOBL algorithm's behaviour. Both optimization methods i.e., SMS and CSMSEOBL are shown in Figure 3.7. The average optimum of the benchmark function is shown in the above-mentioned figures. Also, the global optima of all benchmark functions are shown in the form of semi-logarithmic convergence plots.

CSMSEOBL has been demonstrated to be highly competitive in benchmark functions. The findings indicate that the CSMSEOBL algorithm finds the global optimum by sustaining a good balance between local and global search.

The proposed algorithm gives better result as compared to SMS for unimodal functions i.e.f1, f2, f5, f7 and f12 and results have been shown in Figures 3.7(a), 3.7 (b), 3.7 (e), 3.7 (g) and 3.7 (l) respectively. Similarly, the proposed algorithm has been proven better compared to SMS for most of the multimodal functions also. These multimodal functions are shown in Figure 3.7.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)



(m) (n)

Figure 3.7. Performance Comparison of SMS and CSMSEOBL Algorithm: (a) Sphere Function (b) Schwefel2.21 Function (c) Quartic with noise function (d) Schwefel 2.26 Function (e) Zakharov Function (f) Ackley function (g) Schwefel 2.22 Function (h)ALPINE Function (i) Salomon Function (j) Goldstein & Price Function (k) Powell Function (l) Drop wave Function (m) Easom Function (n) Griewank Function

3.8 Conclusion

When looking for the global optimum of complicated problems, the key struggle is among ‘reliability,’ ‘accuracy,’ and 'computation time.' If standard optimization approaches fail to provide efficiently and consistently trustworthy outcomes, optimization techniques may be a viable option. To prove the correctness of a new metaheuristic algorithm, a developer must use a set of performance metrics. An algorithm's effectiveness may then be shown by comparing these criteria to other methods. These criteria are then compared to other ways to demonstrate an algorithm's efficacy. The SMS algorithm is hybridized with EOBL and Chaotic concept to give a new and improved metaheuristic algorithm. Multiple chaotic maps are employed to update random variables of the SMS algorithm. The chaos facilitates the control parameter in finding the best solution more rapidly and improving the algorithm's convergence rate. The EOBL concept is used to enhance the exploration capability of the Chaotic SMS algorithm. By concurrently assessing the present population and the opposing one, the algorithm is directed to approximate the space in which the global optimum is encompassed. This technique takes full use of the qualities of elite persons, which include a more beneficial search for information than regular individuals and uses them to drive this algorithm to the global optimum response. The suggested approach, on the other hand, may considerably enhance the population's variation and improve computation accuracy, resulting in excellent optimization performance.

Simulation results show that the proposed hybrid algorithm gives better convergence for many unimodal as well as multimodal functions. The beauty of the proposed metaheuristic algorithm is its simplicity and ease of implementation. Another main benefit is fewer parameters to adjust. Furthermore, the novel strategy may improve solution quality without sacrificing robustness

The computational complexity, Statistical analysis as well as convergence behaviour, have been studied in detail. The statistical measures and convergence analysis are presented to show that the suggested method outperforms other metaheuristics. When compared to other algorithms, the findings show that CSMSEOBL's effectiveness is less sensitive to scalability. It would be fascinating to use the CSMSEOBL method to solve real-world engineering challenges in the future

Chapter 4

Optimal Design of FOPID Controller for the Control of CSTR By Using a Novel Hybrid Metaheuristic Algorithm

4.1 Introduction

Several sophisticated control approaches for regulating linear and non-linear processes have been developed in the process control sector during the last two decades. Even though conventional PID controllers are frequently utilized because of their simple structure and reliable operation, sometimes they do not perform well for non-linear systems [121]. Although a simple PID controller provides the least impenetrable, most productive and effortless tuning of controller parameters for the practical process. But with advantages, PID controllers have limitations also like the less optimal solution for a system loaded with non-linearity, time delay, high order disturbances, noise, etc. These limitations lead to the introduction of new and advanced tuning methods like Fuzzy Logic, Neural Network, Adaptive Control, Internal Model Control, etc. which ameliorate the capability and performance of the traditional PID controller[122] along with enhanced flexibility of conventional PID controller.

FOPID, on the other hand, may be used with five parameters to adjust, when a traditional PID Controller only has three. Even though it increases the complexity of parameter tweaking to some level, it also allows for more fine-tuning [4]. Podlubny [5] suggested FOPID as a more sophisticated version of PID controller, where λ and μ are the non-integer order of integral and differential terms, respectively. According to a survey of the literature, FOPID performs better than traditional PID controllers [123]. In continuation of this, presently many metaheuristic algorithms are in great demand for control tuning parameters of the PID controller [124]. The rising complexities in the research area result in limiting the mathematical methods of finding optimal solutions and this necessity results in the investigation of metaheuristic optimization algorithms.

Major limitations with traditional methods of optimization are time-consuming, tedious, less efficient and less accurate [125]. The imperative feature of the metaheuristic algorithm which makes it prominent among researchers is its adaptability and versatility. It can adapt to the problem and determine the optimal solution of different types of problems, whether it is related to mathematics, engineering, process industry, etc.[126].

Using metaheuristics, which are based on trial and error, a difficult issue may be addressed in a fair amount of time and may provide an acceptable answer. The key goal is to come up with a feasible solution in a fair amount of time. When a metaheuristic is selected for a problem, it never guarantees the best answer and we have no way of knowing whether or not it will provide the greatest solution. The basic goal of choosing an algorithm is to provide acceptable or correct results the majority of the time with the least amount of variation. Exploration (diversification) and exploitation (intensification) are two crucial components of every metaheuristic algorithm, with exploration looking for regions that haven't been discovered yet and exploitation looking for additional interesting locations in the sample space. As a result, an algorithm's success is determined by a proper balance of exploration and exploitation, which ensures convergence to optimality [11].

Most of the chemical processes such as continuous stirred tank reactor (CSTR), biochemical reactor and conical tank systems persist dynamic and highly nonlinear behaviour as they consist of multiple process variables to be manipulated. Many advanced controlling and optimization methods are proposed to control such types of MIMO (Multi-input multi-output) systems. Extensive Literature review shows that evolutionary techniques like PSO (Particle Swarm Optimization)[127][128], IWO(Invasive Weed Optimization) [129], SFS(Stochastic Fractal Search) [130] FFA (Firefly Algorithm)[131],GWO (Grey Wolf Optimizer)[119], CSO(Cat Swarm Optimization)[132], TLBO (Teacher-Learner based Optimization)[133][134], SMS (State of Matter Search)[80], CKH (Chaotic Krill Herd)[135], RDO(Red Deer Optimization Algorithm)[136], SOA(Sailfish Optimization Algorithm) [137] and many more have proved their superiority as compared to traditional controllers like Z-N tuned PID, refined Ziegler-Nichols rule [138], intelligent controllers Fuzzy-PID[139], Neural-PID[140], Model-based controllers MRAC (Model reference adaptive control) [141] and Internal model control (IMC)[142].

The proposed methodology is used for the concentration and temperature control of continuously stirred tank reactors (CSTR). A vast literature is available for controlling methodologies of CSTR but it is highly nonlinear and its complex dynamics properties make it a complex problem. Therefore, it is a tedious task to control CSTR by the conventional controller[143]. Nowadays optimization-based control is preferred over the conventional or intelligent controller and to achieve this a hybrid CSMSEOBL methodology is proposed. It is a modified form of SMS algorithm (state of matter search) in which, Chaotic Maps and Elite opposition-based learning (EOBL) are embedded with SMS to enhance the efficiency and efficacy of the SMS algorithm. The basic principle of the SMS algorithm lies in the heart of the thermal energy motion system. The whole algorithm is divided into three states of matter solid; liquid and gas and each state persist in a different diversification-intensification ratio. The algorithm starts with the gas state and modifies the diversification-intensification ratio and ends at a solid-state [80]. The chaotic concept is used for the systems which have high sensitivity towards the initial condition and also it increases the randomness because the range of random numbers is limited.

The chaotic theory has been used with many evolutionary algorithms like PSO, Krill herd, BFO, etc. [135]. This concept of chaotic SMS algorithms is used to define some random variables to stimulate the convergence of SMS. Further, chaotic SMS is merged with elite oppositional-based learning. The concept of OBL was introduced by H. R. Tizhoosh in 2005 which increases the exploration capability of the existing algorithm by combining two main properties of OBL which are a global search and a good convergence rate[122]. EOBL is the superior form of OBL which gives better global search and a higher convergence rate [144]. A fractional-order PID control of CSTR using a hybrid metaheuristic algorithm CSMSEOBL is implemented on MATLAB and results obtained from this hybrid algorithm prove the excellence of the proposed methodology.

4.2 Continuously Stirred Tank Reactor (CSTR)

Chemical process industries rely on Continuous Stirred Tank Reactors (CSTR). It is very nonlinear and has vast operating ranges. To maintain a constant temperature in a reactor, heat must be evacuated or injected, depending on the chemical reaction [201]. The reactor's jacket has feed and exit streams. Energetic fluxes from the reactor into the jacket are believed to be completely blended and at a lower temperature than the reactor. In Figure 4.1, a single coolant stream cools a constant volume reactor, causing an irreversible and exothermic compound reaction.[145].

Considering the uniform volume, exact blending and uniform values of the parameter, the mass-energy balance condition is given by[89].

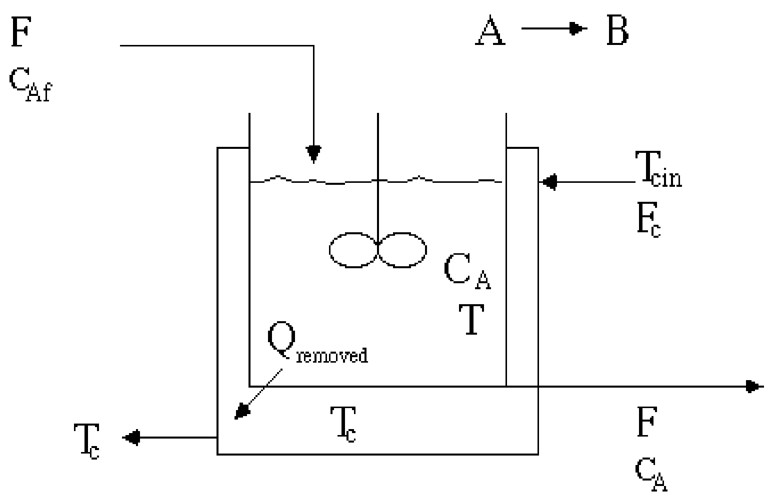


Figure 4.1. Schematic representation of Jacketed CSTR system

4.2.1 Overall material balance[146]

The rate of material accumulation in the reactor is equal to the rate of material inflow minus the rate of material outflow.

$$\frac{dV\rho}{dt} = F_{in}\rho_{in} - F_{out}\rho \tag{4.1}$$

assuming a constant quantity of material in the reactor ($\frac{dV\rho}{dt} = 0$), we find that

$$F_{in}\rho_{in} = F_{out}\rho$$

It is also assumed that the density remains constant, then

$$F_{in} = F_{out} = F \tag{4.2}$$

and $$\frac{dV}{dt} = 0$$

4.2.2 Balance on Component A[146]

The balance on component A is

$$\frac{dVC_A}{dt} = FC_{Af} - FC_A - rV \tag{4.3}$$

where r is the rate of reaction per unit volume.

4.2.3 Energy Balance[146]

The energy balance is

$$\frac{d(V\rho c_p(T-T_{ref})}{dt} = F\rho c_p(T_f - T_{ref}) - F\rho c_p(T - T_{ref}) + (-\Delta H)Vr - UA\left(T - T_j\right) \tag{4.4}$$

where T_{ref} is an arbitrary enthalpy temperature.

4.2.4 State Variable form of Dynamic Equations[146]

We can write (4.1) and (4.2) in the following state variable form (since $\frac{dV}{dt} = 0$)

$$f_1(C_A, T) = \frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - r \tag{4.5}$$

$$f_2(C_A, T) = \frac{dT}{dt} = \frac{F}{V}(T_f - T) + \left(\frac{-\Delta H}{\rho c_p}\right) r - \frac{UA}{V\rho c_p}(T - T_j) \quad (4.6)$$

It is assumed that volume is constant.

$$r = k_o \exp\left(\frac{-\Delta E}{RT}\right) C_A \quad (4.7)$$

where we have assumed that the reaction is first-order.

4.2.5 Steady-State Solution

The steady-state solution is obtained when $\frac{dC_A}{dt} = 0$ and $\frac{dT}{dt} = 0$, that is

$$f_1(C_A, T) = 0 = \frac{F}{V}(C_{Af} - C_A) - k_o \exp\left(\frac{-\Delta E}{RT}\right) C_A \quad (4.8)$$

$$f_2(C_A, T) = 0 = \frac{F}{V}(T_f - T) + \left(\frac{-\Delta H}{\rho c_p}\right) k_o \exp\left(\frac{-\Delta E}{RT}\right) C_A - \frac{UA}{V\rho c_p}(T - T_j) \quad (4.9)$$

Each of the parameters and variables (except for two) must be stated to address these two equations. For the steady-state values of CA and T, the numerical values in Table 4.1 are utilized to solve the equations.

The description of CSTR parameters is given in Table 4.1 [147]. The prime objective is to regulate the temperature and concentration of the reactor by regulating the cooling rate of the reactor.

Table 4.1 CSTR Parameters[141]

Reactor Parameter	**Description**	**Values**
F/V(hr-1)	Flow rate*reactor volume of the tank	1
K_o (hr-1)	Exponential factor	$10e^{15}$
-ΔH (kcal/kmol)	Heat of reaction	6000
E(kcal/kmol)	Activation energy	12189
ρC_P (BTU/ ft^3)	Density*heat capacity	500

Reactor Parameter	Description	Values
Tf(°K)	Feed temperature	315
C_{Af}(lbmol/ft^3)	The concentration of feed stream	1
$\frac{UA}{V}$	Overall heat transfer coefficient/reactor volume	1451
Tj(K)	Coolant Temperature	300

4.3 FOPID Controller

The most pressing problem in control engineering is enhancing system behavior. For that purpose, [148] offered the refinement of conventional Controller parameters to non-integer order of integration and differentiation. Intuitively, this expansion of conventional PIDs provides additional tuning parameters and, as a result, more flexibility in changing the control system's temporal and frequency characteristics. As a result, designs become more enduring. The development of fractional calculus in the past few years has enabled the switch from existing theories and control systems to ones represented by non-integer order ordinary differential equations. As a result, fractional-order estimation techniques and controllers have been developed.

Prof. Oustaloup [148] was the first to introduce fractional-order controllers (FOC). He created the CRONE controller, which stands for "commande Robuste d'ordre nonentier" in French. In [149], the notion of employing FOC for dynamic system control is thoroughly discussed. Podlubny has suggested a generalization of the Classical-PID Controller as a logical consequence to the Fractional order-PID Controller i.e. $PI^{\gamma}D^{\mu}$ controller, incorporating an integrator and a differentiator of order $\gamma \in R^{+}$ and $\mu \in R^{+}$ respectively.

The most common form of PID controller combines three kinds of corrective measures to the error signal, which is the representation of closeness or distance of the desired output from the actual one. In general, these three corrective measures are termed proportional, integral and derivative. The general form of a PID Controller is given by[150]

$$u(t) = k_P e(t) + \frac{1}{k_i}\int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt}) \tag{4.10}$$

Professor Podlubny[151] proposed FOPID Controller in 1999 as an extended form of PID controller which has a comparatively wider range for controlling. The FOPID Controller is shown in Figure 4.2 and represented as

$$u(t) = k_p e(t) + k_i D^{-\gamma} e(t) + k_d D^{\mu} e(t) \tag{4.11}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$ are real numbers with $\boldsymbol{\gamma} > 0, \boldsymbol{\mu} > 0$[152], $\boldsymbol{D}$ is a fractional calculus operator which is defined by Riemann–Liouville as ($\boldsymbol{n}$ is general non-integer order and $\Gamma(\boldsymbol{n})$is Euler's gamma function)

$$D^{-n} f(t) = \frac{1}{\Gamma(n)} \int_0^t f(y)(t-y)^{n-1} dy \tag{4.12}$$

The FOPID controller also takes current error, accumulated error and predicted error into account same as the classical PID controller but fractional operators are non-local in FOPID which gives a modified definition to the integral as well as derivative action[150]. For the analysis purpose, fractional calculus equations must be transferred into algebraic equations. The Laplace transform of the equation for $\boldsymbol{D^{-n} f(t)}$ can be expressed as

$$\int_0^{\infty} e^{-st} f(t) dt = s^{\alpha} F(s) \tag{4.13}$$

Here, it is assumed that all initial conditions are zero [153].

From Figure 4.3 Case I, if $\boldsymbol{\gamma} = 1$ and $\boldsymbol{\mu} = 1$ results in PID controller. Case II, if $\boldsymbol{\gamma} = 1$ and $\boldsymbol{\mu} = 0$, results in PI Controller. Case III, if $\boldsymbol{\gamma} = 0$and $\boldsymbol{\mu} = 1$, resulting in PD controller. Case IV, if $\boldsymbol{\gamma} = 0$ and $\boldsymbol{\mu} = 0$ results in gain controller only. The transfer function of FOPID Control[154] is represented as,

$$G_C(s) = \frac{U(S)}{E(s)} \left(k_p + k_i \frac{1}{S^{\gamma}} + k_d S^{\mu}\right) \tag{4.14}$$

The use of the FOPID Controller results not only in enhanced performance of the control system, and better adaptability but fine control of the dynamical system as well as very fewer variations in parameters of a control system [155].

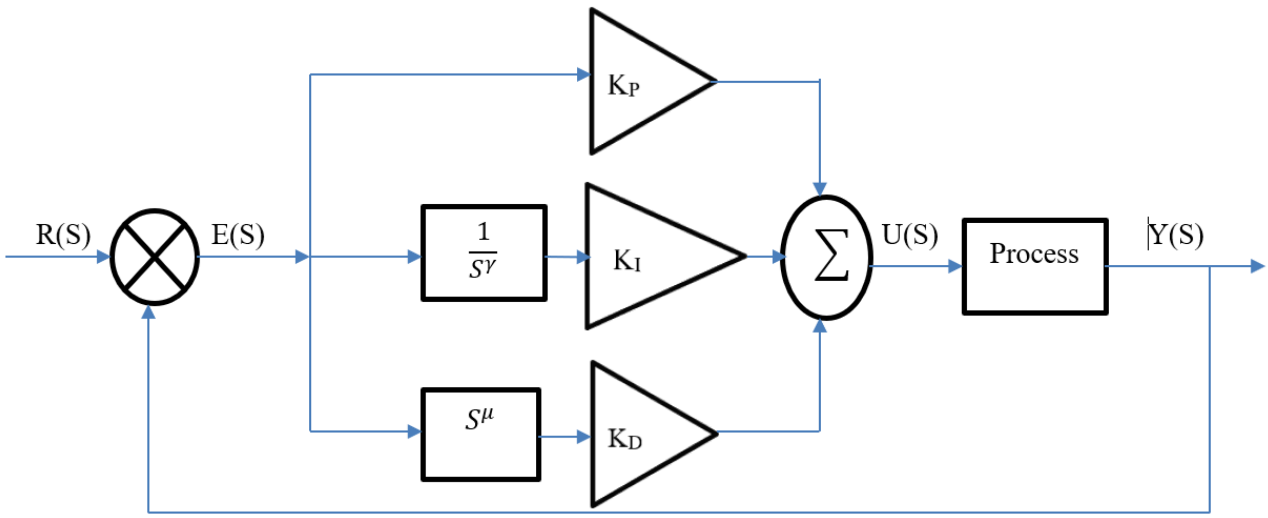


Figure 4.2. FOPID Controller

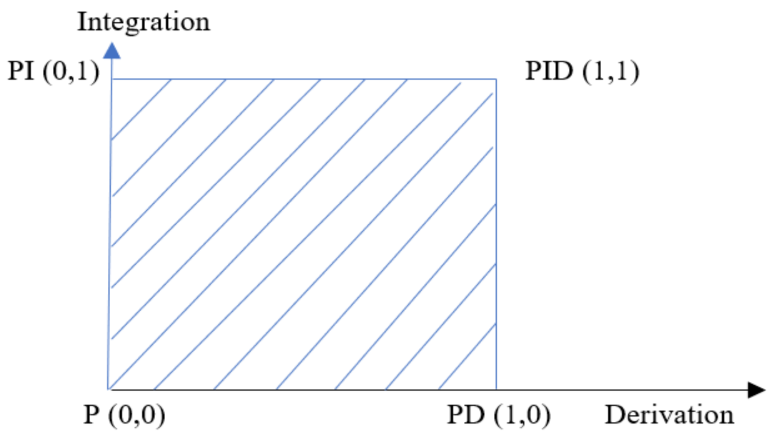


Figure 4.3. Plane of the FOPID controller [156]

The FOPID Controller ensures that the plant model is resistant to gain change, noise and disturbance. As a result, it has a better transient reaction than an integer-order PID controller. As illustrated in Figure 4.3, the fractional-order PID controller extends the integer-order PID controller from point to plane form. When compared to integer-order PID controllers, this expansion gives us greater versatility in controller design and allows us to regulate our real-world processes more precisely.

4.4 Metaheuristic Optimization Algorithms

Figure 4.4 represents the flow of the proposed work with parameters of the FOPID Controller which are optimized by the metaheuristic optimization and controlled parameters are fed into the process.

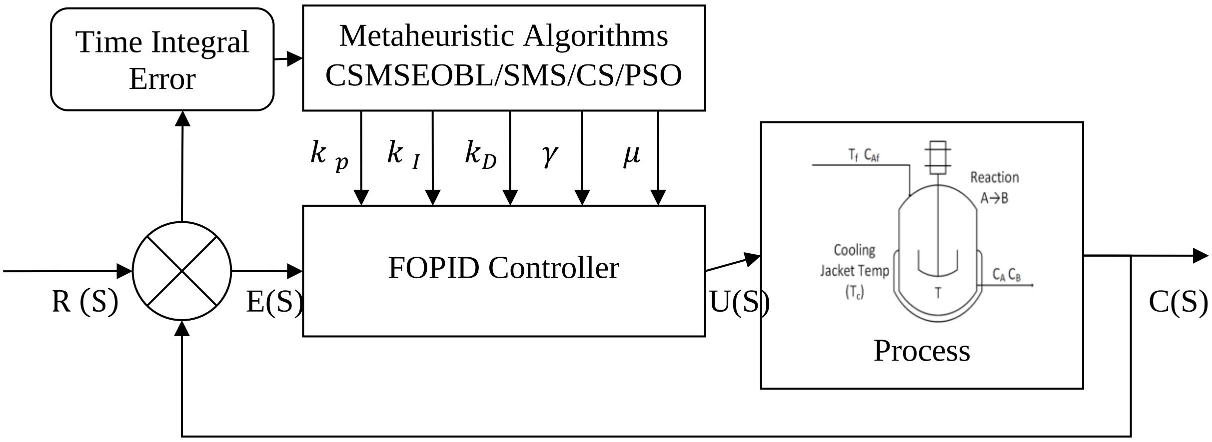


Figure 4.4. Optimized FOPID for Control of CSTR

4.4.1 Particle Swarm Optimization (PSO)

PSO is a swarm intelligence-based optimization algorithm that was proposed by Kennedy and Eberhart in 1995 [16]. It simulates the concept of cooperation, communication and social behaviour in fish and bird schooling. Literature [157] reveals that extensive research has been done on PSO to demonstrate its efficiency in solving real-valued complex, non-linear, non-differentiable optimization problems. However, since the search space dimension can be sufficiently increased, PSO is sensitive to the trend of falling into local optima. To solve this limitation with traditional PSO some improved and hybridized version of PSO has been introduced from time to time to enhance its convergence performance [158]. It is a population-based optimization technique that gives rise to high-quality results within a more concise time and shows stable converge characteristics [147].

There are several steps in the PSO process. The PSO's primary processing loop initially updates each particle's current velocity depending on the particle's current velocity, the particle's local information, and the global swarm information that is available to the PSO [147]. The velocity of each particle is then used to update the location of each particle. Both equations are updated in mathematical terms.:

$$v(t+1) = (w * v(t)) + (c_1 * r_1 * (p(t) - x(t)) + (c_1 * r_2 * (g(t) - x(t)) \qquad (4.15)$$

$$x(t+1) = x(t) + v(t+1) \qquad (4.16)$$

where, r_1 and r_2 are random numbers with a value between [0,1], c_1 and c_2 are two acceleration constants, w is inertia weight, $x(.)$is the position of the particle, $p(t)$ is the personal best position of the particle, $g(t)$ is the global best position of the group. The term $v(t+1)$ gives velocity at the time$(t+1)$. Once $v(t+1)$, has been computed, it is used to compute updated position $x(t+1)$ $of\ particle$[159].

4.4.2 Cuckoo Search Algorithm (CS)

Yang and Deb[160] developed a novel meta-heuristic calculations cuckoo search in 2009. CS is dependent on the brood parasitism of certain cuckoo species for its survival. Moreover, the calculation is upgraded by the purported Lévy flights, as opposed to basic isotropic irregular strolls. Cuckoos are interesting flying creatures, not just as a result of the excellent sounds they can make yet additionally on account of their forceful generation methodology. A few animal types, for example, the ani and guira cuckoos lay their eggs in shared homes, however, they may evacuate others' eggs to expand the bring forth likelihood of their eggs (Figure 4.6). A lot of animal varieties connect with the committed brood parasitism by laying their eggs in the homes of other host winged animals[144]. In cuckoo search calculation cuckoo egg speaks to a potential answer for the structure issue which has an objective function. The calculation utilizes three glorified guidelines (Figure 4.5):

- Rather than laying each egg in turn, each cuckoo just drops them at a random location.
- The finest house, with the best eggs, will be passed on to the next generation of chickens.
- Host flying animals can locate an outsider egg with a probability of $\boldsymbol{P_a} \in [\mathbf{0}, \mathbf{1}]$ when the number of accessible host houses is fixed, and the number of accessible host homes is fixed [161].

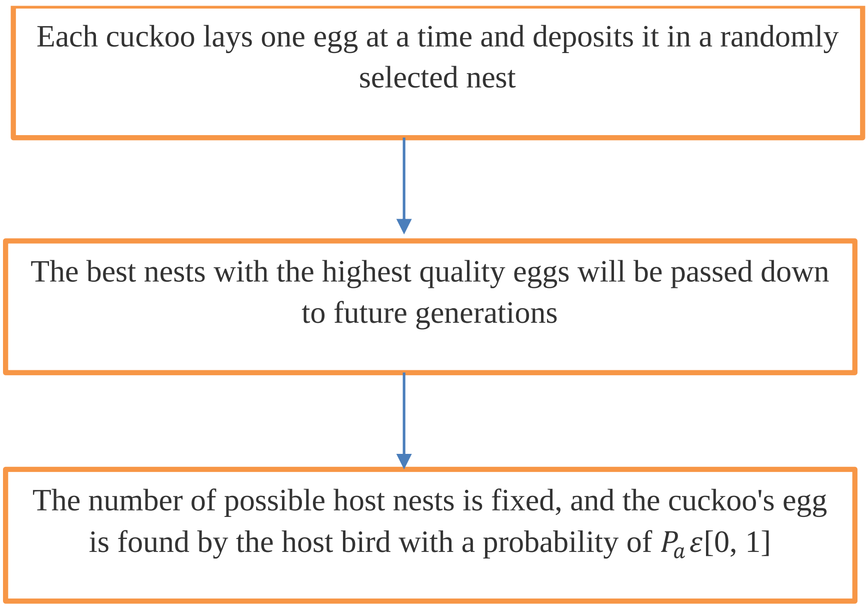


Figure 4.5. How the Cuckoo search algorithm works [144]

Pseudo code of Cuckoo Search [162]

Fitness function f(x), $x = (x_1, x_2, \ldots, x_n)^T$;

Initialize population for 'n' no. of host x_i $(i = 1,2, \ldots, n)$;

while (p <Max no. Of Generation) or (stop criterion);

obtain a cuckoo (say i) randomly by using Levy flights;

$$x_i^{p+1} = x_i^p + \alpha \oplus L'evy(\lambda)$$

Examine its quality f_i ;

Choose one of n (say j) nests at random;

if $(f_i > f_j)$,

Substitute the new solution for j;

end

Abandon a percentage of the worst nests p_a

[and create new ones in other places with Lévy flights];

Sort the options and choose the best one;

end while

Visualization of findings and post-processing;

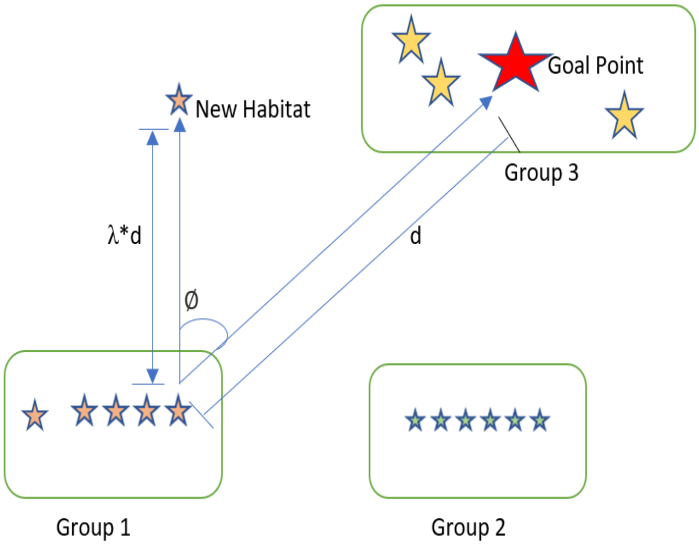


Figure 4.6. A sample cuckoo making its way to its preferred territory [162].

4.5 Simulation Results

To confirm the practicality and viability of the proposed hybrid CSMSEOBL approach, a progression of comparative experiments have been performed on CSTR against the accompanying three states of the art metaheuristic optimization techniques: PSO, CS, SMS and CSMS-EOBL. MATLAB 2018 is used for simulation and Intel(R) Core (TM) 2 Duo CPU T6400@ 2.00 GHz 1.20 GHz, 1.99 GB of RAM. The performance is verified for Control Temperature and Concentration of CSTR by running CSMSEOBL-based FOPID, SMS-based FOPID, CS-based FOPID and PSO-based FOPID controller and results are compared.

For any optimization process convergence of metaheuristic algorithm towards the global optima of the tuned parameters of FOPID, the problem is defined with an objective function or fitness function. To get the finest transient response as well as minimum steady-state error along with the least overshoot, ITAE is utilized as the objective function. Since ITAE is the most aggressive controller setting criteria that avoid peaks and give controllers a greater load disturbance rejection and lessens the overshoot of the system while retaining the robustness of the system. ITAE is defined as

$$J_{ITAE} = \int_0^T t|e(t)|dt$$

4.5.1 Parameter setting for different metaheuristic algorithms

Table 4.2 Parameter Setting of different Metaheuristic Algorithms[89]

Algorithm and Parameters	**Parameter Value**	**Algorithm and Parameters**	**Parameter Value**
PSO		**CS**	
Population	50	Population	50
Iteration	25	Iteration	25
Weight Function	[0.2,0.9]	Pa	0.25
Acceleration constants	2	Beta	1.5
The dimension of search space	5	The dimension of search space	5

Iteration	25	Iteration	25
SMS		**CSMS-EOBL**	
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

The CSMSEOBL is used to optimize the parameters of FOPID for concentration and temperature control of CSTR. To show the comparative study CS, PSO, and SMS algorithms are also implemented on CSTR. MATLAB Simulink environment is utilized for evaluating the results.

Further, different types of chaotic maps are used to enhance the randomness of the SMS algorithm. Figure 4.7 and Figure 4.8 show the variation of concentration and temperature for different types of chaotic maps respectively.

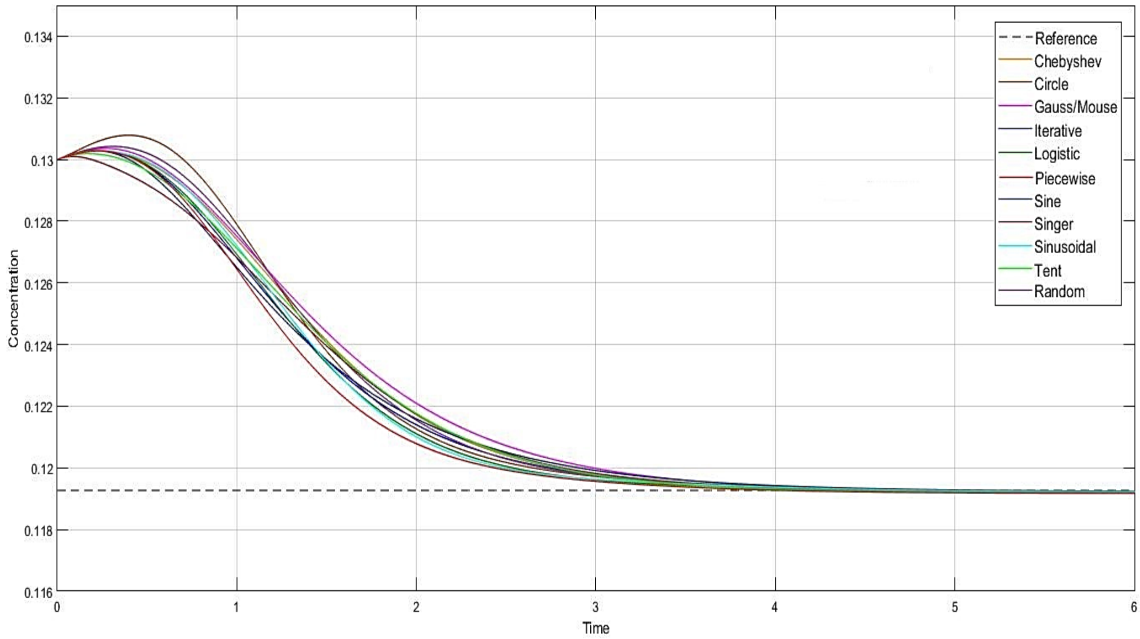


Figure 4.7. Simulation result for concentration control of CSTR with CSMSEOBL algorithm with different types of Chaotic Maps

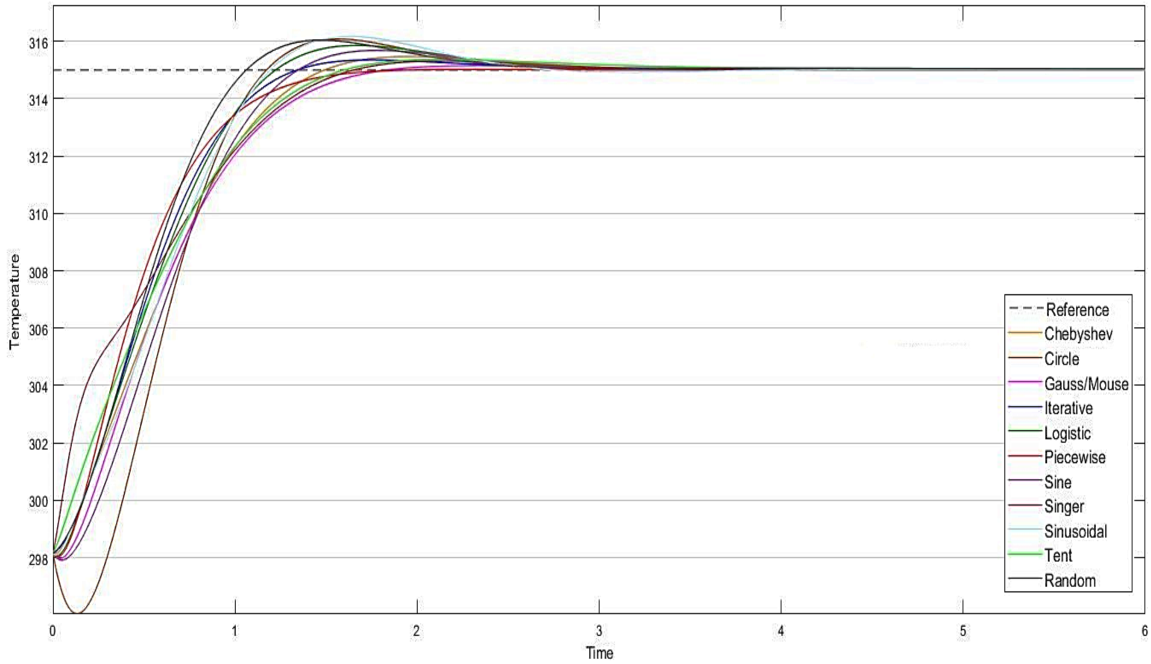


Figure 4.8. Simulation result for temperature control of CSTR with CSMSEOBL algorithm with different types of Chaotic Maps

The proposed algorithm is further demonstrated by a comparative analysis of the best solution obtained from Figures 4.7 and 4.8 with the existing algorithms, which include SMS, CS, and PSO. These results are shown in Figures 4.9 and 4.10, respectively, to demonstrate the superiority of the proposed algorithm over the existing algorithms. The setpoint for Concentration is taken as .119 (lb. mol/ft^3) and the temperature is at 315K.

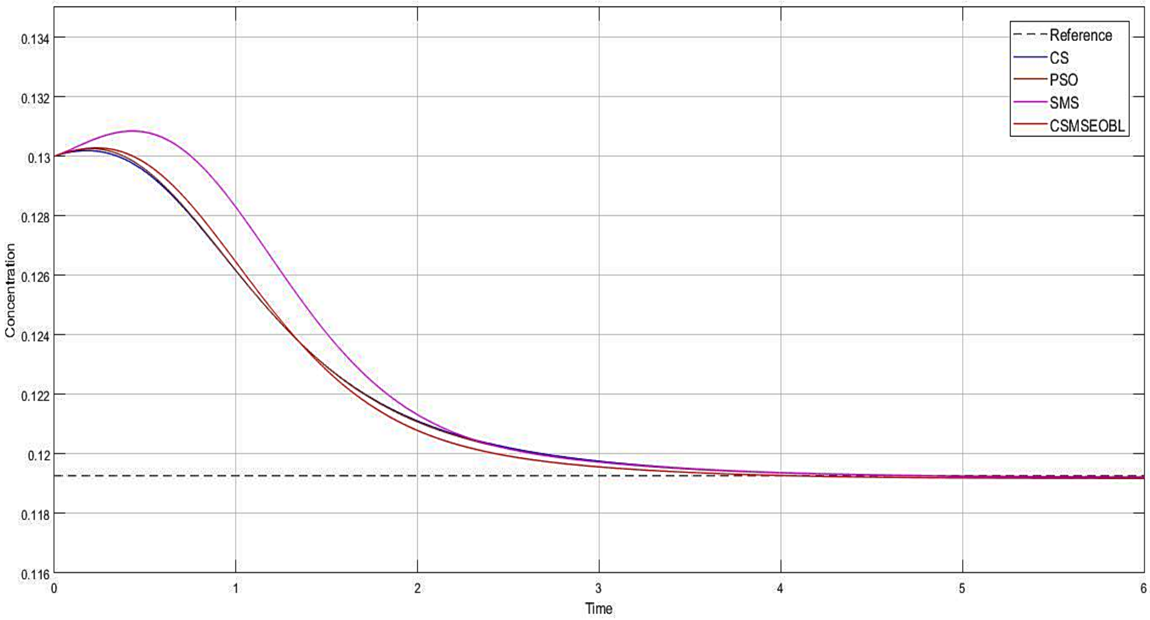


Figure 4.9. Comparison of concentration control of the CSTR system among different metaheuristic algorithms

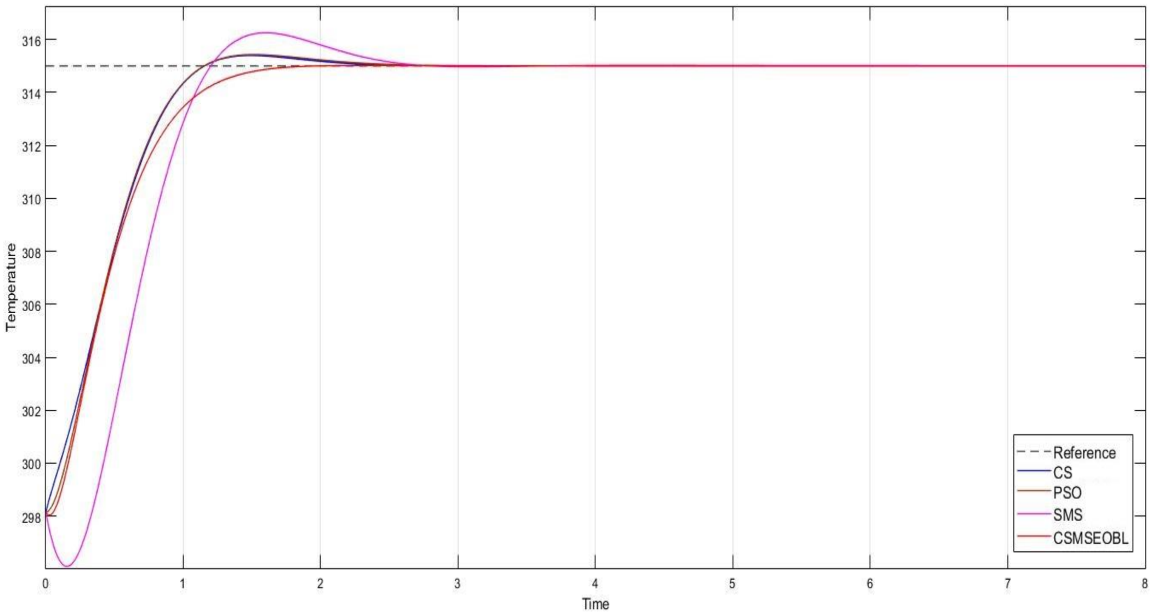


Figure 4.10. Comparison of Temperature control of the CSTR system among different metaheuristic algorithms

4.5.2 Transient response analysis

From Table 4.3 we could conclude that the proposed CSMSEOBL shows a promising approach for concentration and temperature control of CSTR because in process control problems main aim is to obtain the least settling time and minimum overshoot. Even though the rise time and the peak time are large for CSMSEOBL-FOPID as compared to SMS-FOPID, CS-FOPID and PSO-FOPID but the cost function is minimized along with minimum overshoot and least settling time.

Table 4.3 Comparative analysis of controller parameters and time response specifications

	FOPID Controller Parameter					**Rise time**	**Peak time**	**Overshoot**	**Settling time**
	K_p	K_I	K_D	γ	μ				
SMS	12.1	32.5	1	1.006	0.100	1.22	1.53	.38	2.27
CS	21.7	50	0.2	1.002	0.785	1.13	1.36	.12	1.86
CSMSEOBL	15.8	43.3	1.9	.9999	0.138	2.04	1.68	0	1.43
PSO[163]	.2510	.0243	.499	.5968	.0706	3.65	4.76	7	14

4.5.3 Convergence analysis

The CSTR's dynamic performance is enhanced by utilizing a mathematical formulation of the objective function ITAE to improve performance indicators such as settling time, rising time, and overshoot. This is accomplished via the use of a mathematical formulation of the goal function ITAE. ITAE decreases not only the initial extent of error but also decreases the error which develops in later responses[164]. Variation of ITAE for different metaheuristic algorithms has been shown in Figure 4.11. The comparative analysis of considered metaheuristic algorithms in terms of ITAE is shown in Figure 4.12. The proposed CSMSEOBL algorithm outperformed the other metaheuristic algorithms and it has been shown in Table 4.3.

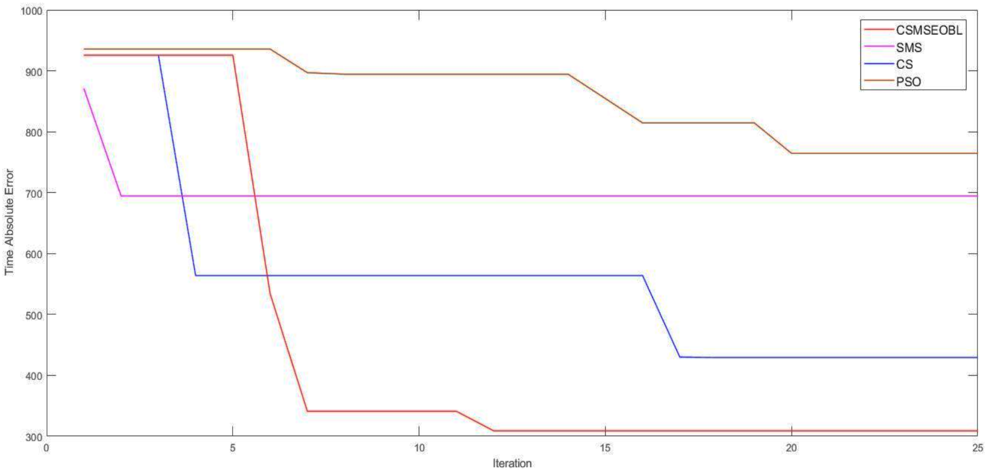


Figure 4.11. Variation of ITAE for different metaheuristic algorithms

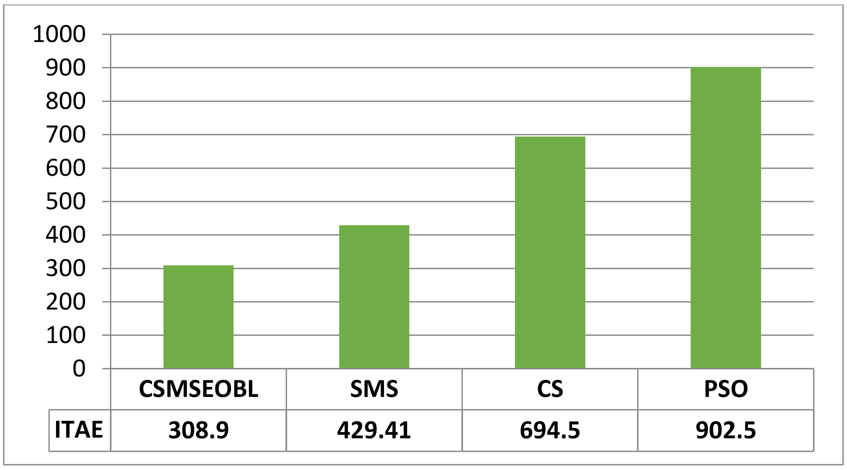


Figure 4.12. Comparison of objective function ITAE for different metaheuristic algorithms

4.6 Conclusion

This paper fixes the limitations of the standard SMS Algorithm by hybridizing it with chaotic maps and Elite Opposition Based Learning. Further, this hybrid algorithm CSMSEOBL is used to find optimal parameters of the FOPID Controller for the temperature and concentration control of a Continuously Stirred Tank Reactor (CSTR). Major findings of the work are as follows: CSMSEOBL gives better exploration and exploitation capability; The use of CSMSEOBL on a non-linear control problem results in faster convergence; CSMSEOBL shows promising results in terms of overshoot, settling time, and ITAE for optimizing the performance; The proposed controller is validated for concentration and temperature control of CSTR.

This study of hybrid metaheuristics can be further extended to reform the transient performance using multiple models, adaptive control strategy, and other latest metaheuristic algorithms.

Chapter 5

CSMSEOBL-PID Controller for Ball Balancer System

5.1 Introduction

The ball balancer is a nonlinear, multivariate, unstable and electromechanical system that is underactuated. The use of intelligent controllers or automatic decision development to approximate such underactuated systems is a challenge that arises in a variety of control situations and may be solved utilizing a variety of techniques[165]. Many process control systems have these characteristics by default, making it difficult to devise a viable control approach for them [166]. Due to the complexity, Researchers in the field have started to investigate different linear, nonlinear, intelligent and model-free controllers. These controllers aim to achieve self-adjusting and consistent control for several processes like the mechanism of ship control, horizontal stabilization of an airplane during turbulent airflow and landing, the twin-rotor multi-input multi-output system, inverted pendulum, hovercraft, Furuta pendulum, the ball beam system and ball and plate system [167] [168].

While there are a few different hardware configurations for ball and plate systems[169]–[171], the basic device dynamics and control concept is the same for all of them: It comprises of a base plate, a ball, an overhead camera, and two servo units. In addition to the servo units and plate, an overhead camera detects the location of the ball for controller input. There are two servos beneath the plate that link to the plate and give 2-DOF gimbal movements for the camera. The plate has two axes of motion. But the system can only be described in one axis since both axes are symmetrical about each other [172]. As a benchmark type engineering challenge for various controller designs, the ball and beam control experiment are divided into two categories: model-based control systems and non-model-based control systems [226]. Model-based control systems must contain regulated system states that cannot be computed clearly because they are needed by the model. Non-model-based systems rely on real-time sensor output data that is available at the moment of control. The primary disadvantage of model-based control architecture is that it is extremely dependent on the correctness of the model employed in the state observer, which may be a source of contention. Because any of the aspects that determine the physical system's nonlinear dynamics cannot be included in the system model, model-based control loses its effectiveness over time as a result of improvements in the physical configuration of the system or wear and tear on the system's mechanical components. In addition, the practical specification of the ball and beam arrangement, which is employed for certain non-model-based control designs, is a factor in determining the differences between various controller designs. [173].

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 [174][175].

Kennedy and Eberhart [176] developed Particle Swarm Optimization (PSO) as an optimization technique in 1995. It guides the particles to find global optimum solutions using a basic process that mimics swarm behaviour in birds flocking and fish schooling. It's a swarm-based, efficient stochastic optimization strategy. It uses the principle of social interaction to solve problems and does not depend on the gradient of the problem to be solved, because it does not necessitate a differential optimization problem, as traditional optimization approaches do [177]. The PSO algorithm has a lot of advantages. It's easy to set up, needs just a few parameters, works well in global searches, is unaffected by architecture variable scaling and can be quickly parallelized for parallel processing [178].

In this chapter, the chaotic SMS algorithm idea is used to describe certain random variables to promote SMS convergence. By integrating the two key properties of OBL, a global search and a fast convergence rate, the diversification capabilities of the current algorithm get improved. PID control of a ball balancer is tested on MATLAB using existing algorithms like PSO, SFS, SMS and a hybrid metaheuristic algorithm called CSMSEOBL and the results show that the suggested approach is superior.

5.2 Mathematical modelling of Ball balancer system

The 2 DOF Ball Balancer, or 2DBB, as seen in Figure 5.1, is made up of a plate on which a ball can be put and move freely. The plate can be swivelled 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-axis ball motion is similar to the x-axis. Section 2.1.1 gives the equation of the ball’s motion at the x-axis in comparison to the plate’s angle, while Section 2.1.2 integrates the servo angle into the model.

The open-loop structure of the 2D Ball Balancer is illustrated in Figure 5.1. 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 5.2 depicts the free body diagram of the 2D Ball Balancer.

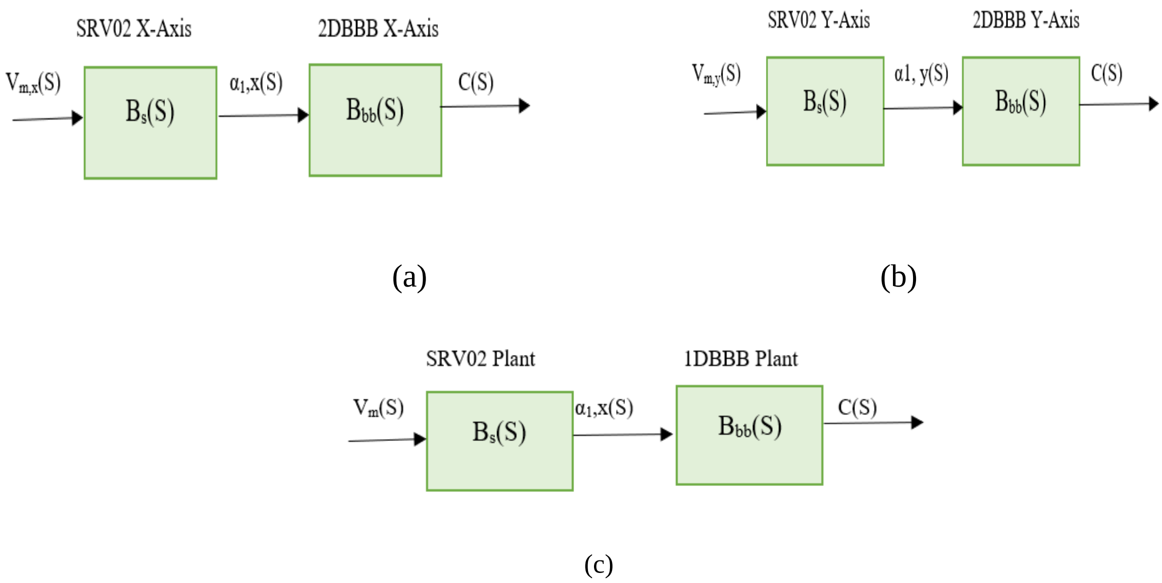


Figure 5.1. Ball balancer open-loop block diagram (a) Servo X-axis (b) Servo Y-axis (c) Open Loop block diagram of 1-D plant

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

$$B(S) = B_{bb}(S)\, B_s(S) \tag{5.1}$$

where

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

and $$Bs(S) = \frac{\alpha_1(S)}{V_m(S)}$$

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

$$m_b \ddot{C(t)} = \sum F = F_{x,t} - F_{x,r} \tag{5.2}$$

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 (Figure 5.2).

When the incline is positive, the force acting in the positive x-direction is

$$F_{x,t} = m_b g \; sin\beta(t) \tag{5.3}$$

The force generated by the ball's rotational spin

$$F_{x,r=\frac{\tau_b}{r_b}} \tag{5.4}$$

$$\tau_b = J_b \ddot{\Theta}_b(\text{t}) \tag{5.5}$$

Where $r_b - radius\ of\ sphere$

$\tau_b - torque$

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

$$F_{x,r=\frac{J_b \ddot{x}_b(\text{t})}{r_b^2}} \tag{5.6}$$

From Equation (5.2), (5.3) and (5.6)

$$m_b C\ddot{(t)} = \; m_b g \; sin\beta t - \frac{J_b \ddot{C}(\text{t})}{r_b^2} \tag{5.7}$$

5.2.1 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:

$$\sin(\beta(t)) = \frac{2sin(\alpha_1(\text{t}))r_a}{l_t} \tag{5.8}$$

where

$$r_a - distance\ between\ the\ coupled\ joint\ and\ the\ output\ gear\ shaft\ of\ the\ SRV02.$$

$$l_t - Table\ length$$

Using linearization around Θ_1 = 0, the dynamic variables for characterizing the rotation of the ball that corresponds to the servo angle Θ_1 are discovered. The mathematics for the servo and plate angle relationship is as follows:

$$C(\ddot{t}) = \frac{2m_b g r_a r_b^2 sin\alpha_1(t)}{l_t(m_b r_b^2 + J_b)} \tag{5.9}$$

The sine function sinγ_l is approximated as γ_l to linearize the equation of motion and the final equation for the 1-D ball balancer is:

$$C(\ddot{t}) = \frac{2m_b g \alpha_1(t) r_a r_b^2}{l_t(m_b r_b^2 + J_b)} \tag{5.10}$$

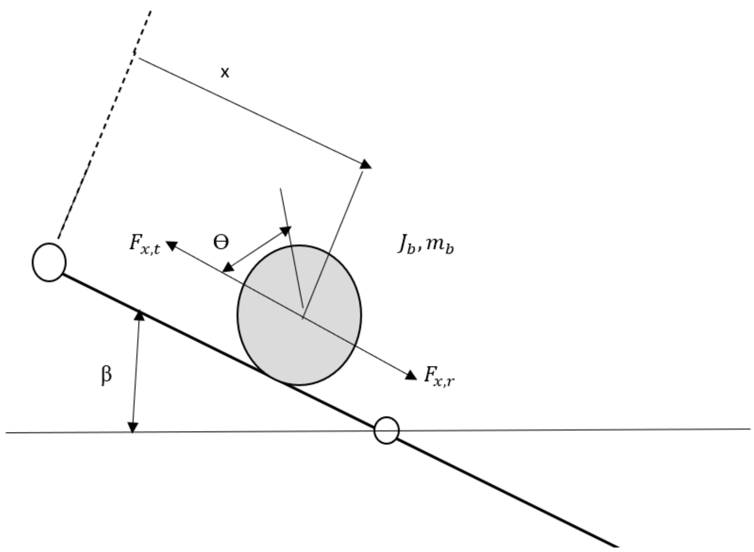


Figure 5.2. Free body diagram of Ball balancer

5.2 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. The motor angle should track the reference signal, so the inner controller should be programmed 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 the ball balancer system.

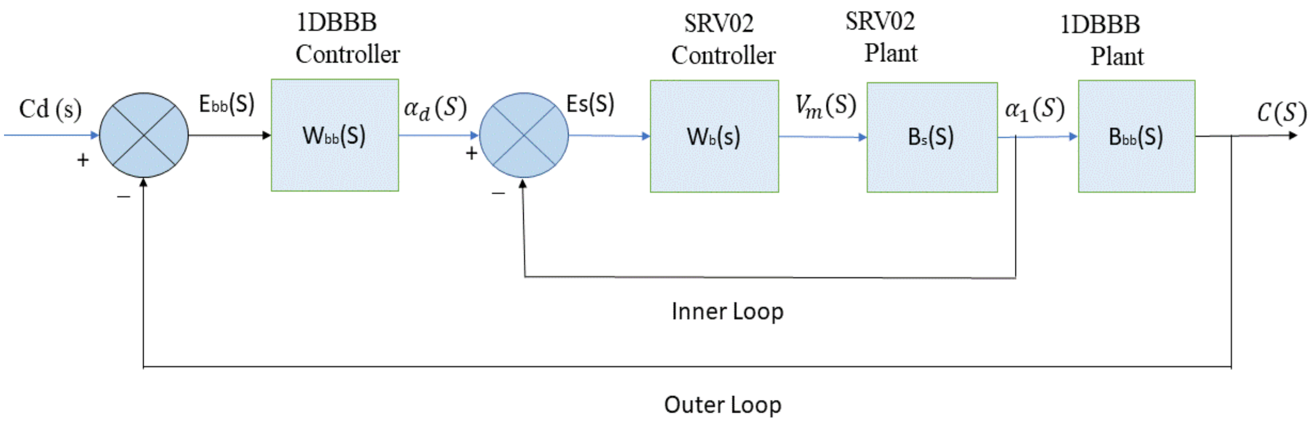


Figure 5.3. Block Diagram of Ball Balancer System

5.2 Particle swarm optimization (PSO)

Kennedy and Eberhart [176] 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 [179]. Each individual in the PSO technique is referred to as a particle, and each particle represents a possible solution in a population-based search process. A variable speed is used by each particle to travel across the search space, which is dynamically changed depending on the flight experiences of the particle and those of other particles[180]. 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 [181].

The position and velocity update equations of PSO are given by

$$v_{i.g}^{(t+1)} = w.v_{i.g}^{(t)} + a_1 * rand() * \left(pbest_{i.g} - x_{i.g}^{t}\right) + a_2 * rand() * \left(gbest_g - x_{j.g}^{t}\right) \qquad (5.11)$$

$$x_{i.g}^{(t+1)} = x_{i.g}^{(t)} + v_{i.g}^{(t+1)} \qquad (5.12)$$

i=1, 2…... n; g=1, 2... 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; $g_{best} - g_{best}$ of the group

$$w = \frac{w_{max} - w_{min}}{iter_{max}} * iter \tag{5.13}$$

The inertia weight $'w'$ is calculated using Equation (5.13). 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.

5.4 Stochastic Fractal Search (SFS) 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 [182]. SFS aims 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 [183]. The flow chart of the SFS algorithm is illustrated in Figure 5.4

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:

$$P = lb + r(ub - lb) \tag{5.14}$$

where '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:

$$Pa_i = \frac{rank\ P_i}{N} \tag{5.15}$$

where $rank\ P_i$ denotes the position of the point P_i with the other points in the set and N denotes the total number of points in the group.

Equation (5.16) is used to update the j[th] element of P_i; otherwise, it stays consistent.

$$P_i'(j) = P_r(j) - r(P_t(j) - P_i(j)) \quad (5.16)$$

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 $Pa_i < r$ r holds for a new point P_i' the current location of P_i' is updated according to Equation nos. (5.17) and (5.18), otherwise, there is no change.

$$P_i'' = \quad P_i'(j) - \hat{r}(P_t' - BP) |\ r' \leq 0.5 \quad (5.17)$$

$$P_i'' = \quad P_i'(j) + \hat{r}(P_t' - P_r') |\ r' > 0.5 \quad (5.18)$$

BP-The best point out of all the points.

Stochastic Fractal Search Algorithm Pseudo Code

```
Initialize N point population;
while G < maximum no. of generation do
    for each Point Pi in the system do
        Call Diffusion (it consists of the following steps):
        q = (maximum number of diffusion).
        for j = 1 to q do
            if the first Gaussian walk is applied then
                generate a new point based on Gaussian Walk
            end
            if second Gaussian Walks is used
                generate a new point based on Gaussian walk after diffusion
            end
            end
        Call Updating Process to consist of following steps
        begin
            First Updating Phase.
            all points are ranked as Pa_i = rank P_i / N
```

for *each Point* P_i *in the system* do

for *each component* j *in* P_i do

if r *[0,1]* > Pa_i then

Update the component as $P_i'(j) = P_r(j) - r(P_t(j) - P_i(j)$

end

end

end

Second Updating Phase:

again, all points are ranked based on $Pa_i = \frac{rank\ P_i}{N}$

for *each Point* $P_i{}'$ *in the system* do

if r *[0,1]* > Pa'_i then

Update the position based on $P_i'' = P_i'(j) - \hat{r}(P_t' - BP) | \, r' \leq 0.5$ and

end $P_i'' = P_i'(j) + \hat{r}(P_t' - P_r') | \, r' > 0.5$

end

end

end

end

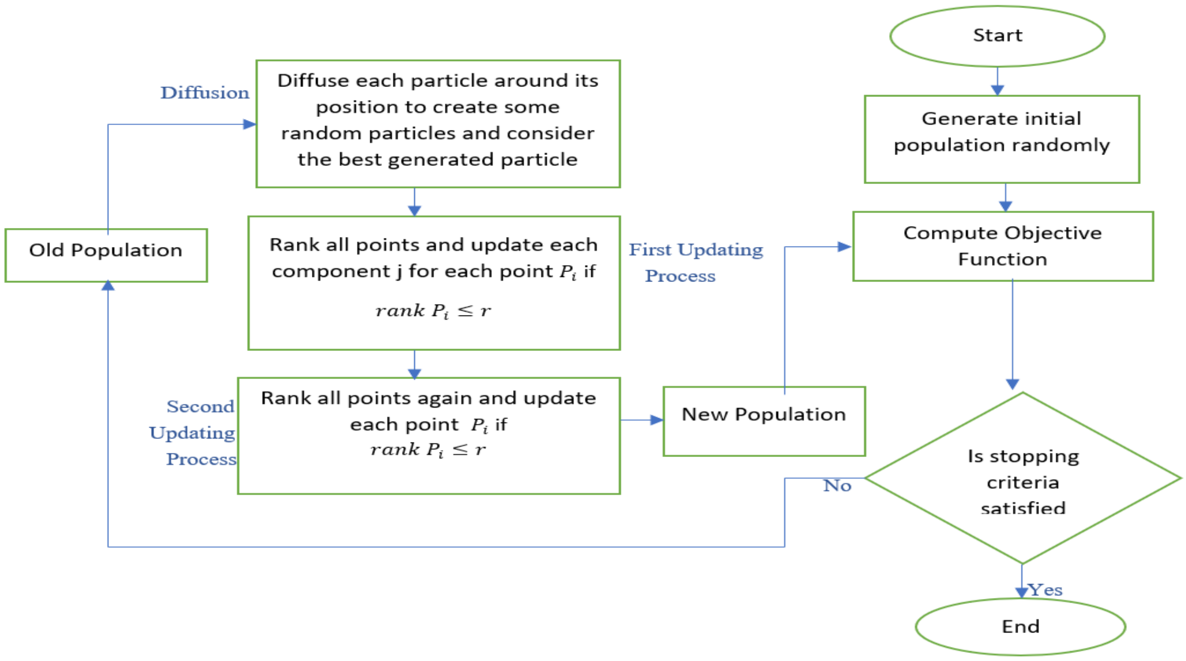


Figure 5.4. Flow Chart of SFS algorithm

5.5 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 [184] While there are many classical approaches for designing and tuning PID controller parameters (K_p, K_i and 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 [185].

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.

5.5.1 Tuning of PID Controller

The Ziegler-Nichols [186] 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 are often used by controller manufacturers and the process industry [187][188]. Z-N tuned PID Controller parameters for closed-loop control systems are given in Table 5.1.

Table 5.1 Z-N tuned parameters for PID Controller

Tuning Method	**K_p**	**T_i**	**T_d**
Z-N Closed Loop	$0.6K_u$	$P_u/2$	$P_u/8$

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).

5.6 Simulation Results and Analysis

The numerical simulation of the 2DOF ball balancer model, which is theoretically presented in Section 5.2, is created using the MATLAB/Simulink tool. Servo unit controllers may influence each other's actions since their plates are symmetrical, allowing them to affect one another. Inner loop values stay unchanged, but PID is applied to an exterior shell to add the position of poles as far as time constant decay is concerned.

Initially, the PID controller's values are determined using the traditional PID Controller. Furthermore, metaheuristic algorithms such as PSO, SFS, SMS and finally the hybridized CSMSEOBL perform by calculating the difference between the expected and calculated ball positions 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 towards 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 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 5.2.

The step response analysis is a study of how a system behaves at the start and finish of its entire lifecycle. It gives information on the closed-loop system's relative stability and responsiveness. While the quantity of maximum overshoot may be linked to relative stability, the settling and rising durations demonstrate the system's speed and quickness. Figure 5.6, Figure 5.7and Figure 5.8 respectively show the effects of the ball position, servo angle and voltage of the ball balancer mechanism for hybridized CSMSEOBL-PID, PSO-PID, SFS-PID, SMS-PID and the classic PID controller. The transient reactions of Classic PID, PSO-PID, SFS-PID, SMS-PID and CSMSEOBL-PID controllers to a reference position are shown in Figure 5.6. Table 5.3 also includes numerical comparisons in terms of maximum overshoot, rising time (10 percent to 90 percent) and settling time (2 percent band). All of the performance criteria, such as overshoot, rising time and settling time, 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 Figure 5.6 and Table 5.3. 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 Figure 5.6, 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 is shown in Figure 5.9 and this shows that the CSMSEOBL algorithm gives the best results for piecewise Chaotic maps.

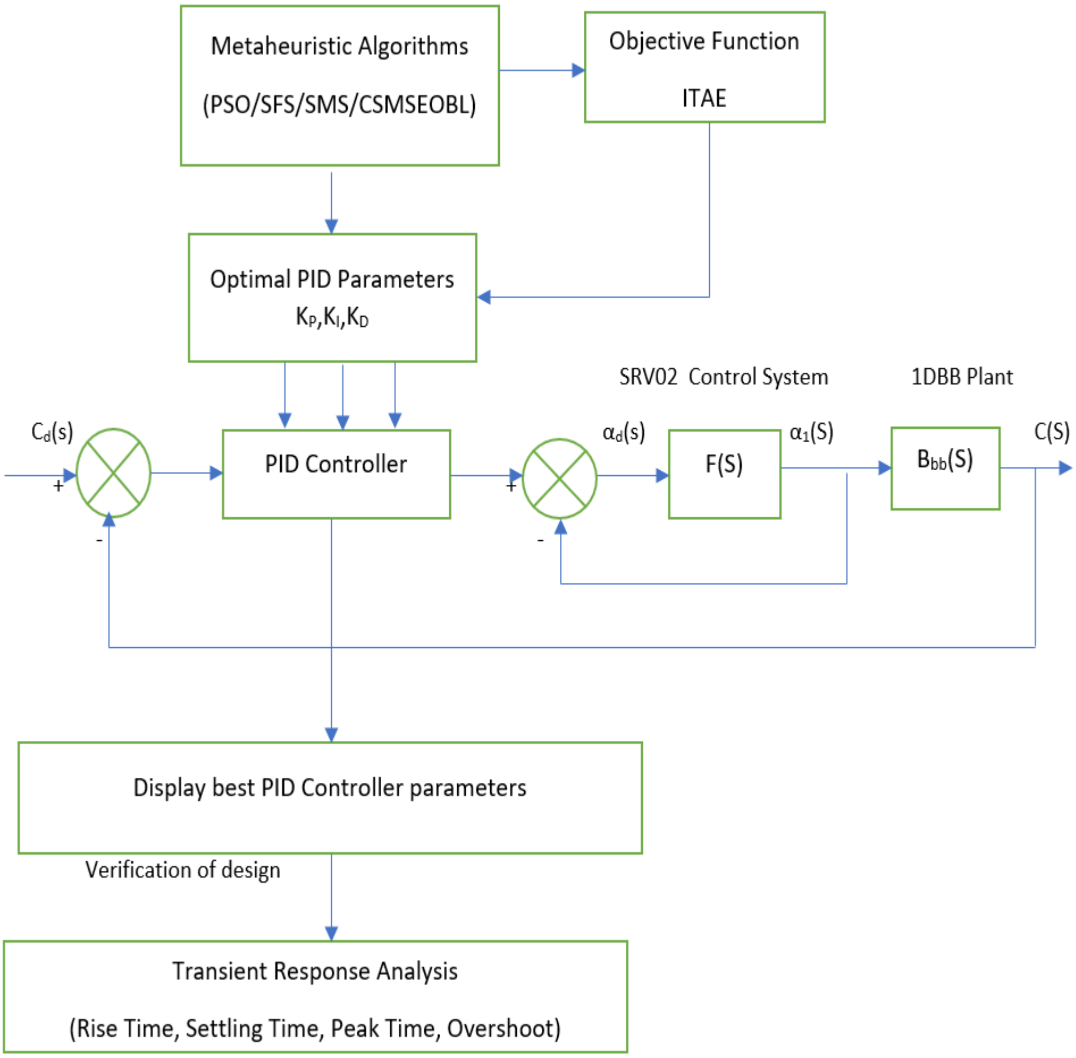


Figure 5.5 Block diagram representing tuning of PID controller parameters using different metaheuristic algorithms for Ball balancer system

Table 5.3 shows that the classical PID controller 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 controller results in the finest response in terms of least overshoot and least ITAE as shown in Figure 5.6.

A total of 150 optimization trials are carried out to assess the efficacy of all algorithm's 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.

5.6.1 Parameter setting for different metaheuristic algorithms

Table 5.2 Parameter Settings 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		**CSMS-EOBL**	
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

5.6.2 Transient response and Error convergence analysis

Table 5.3 Comparison of step response characteristics for Z-N tuned PID controller

	PID Controller Parameters			**Step Response Characteristics**		
Name of the Algorithm	**Kp**	**Ki**	**Kd**	**Rise Time**	**Settling Time**	**Max. Overshoot**
Classical PID	3.45	0.0012	2.11	0.827	3.65	8.152

Table 5.4 Comparison of step response characteristics and Fitness function for Ball balancer system

	PID Controller Parameters			**Step Response Characteristics**			
Name of the Algorithm	**Kp**	**Ki**	**Kd**	**Rise Time**	**Settling Time**	**Max. Overshoot**	**Fitness Function ITAE**
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

The performance of the proposed CSMSEOBL tuned PID controller is compared with already existing metaheuristic optimization algorithms like SMS, PSO and SFS. Table 5.4 shows that CSMSEOBL tuned PID controller results in zero overshoot and the optimal result of fitness function ITAE. Figure 5.6, Figure 5.7 and Figure 5.8 demonstrate the simulated position, servo angle and voltage respectively for the ball balancer system using different metaheuristic algorithms. The simulated position of the ball with the CSMSEOBL algorithm for a variety of Chaotic maps has been presented in Figure 5.9 and the logistic map gives the best step response among all 10 chaotic maps. Figure 5.10 shows the set point tracking response for all four metaheuristic algorithms and CSMSEOBL results in best set point tracking. The convergence of ITAE for different metaheuristic algorithms is presented in Figure 5.11 and Figure 5.12 shows the comparative analysis of fitness function for different metaheuristic algorithms.

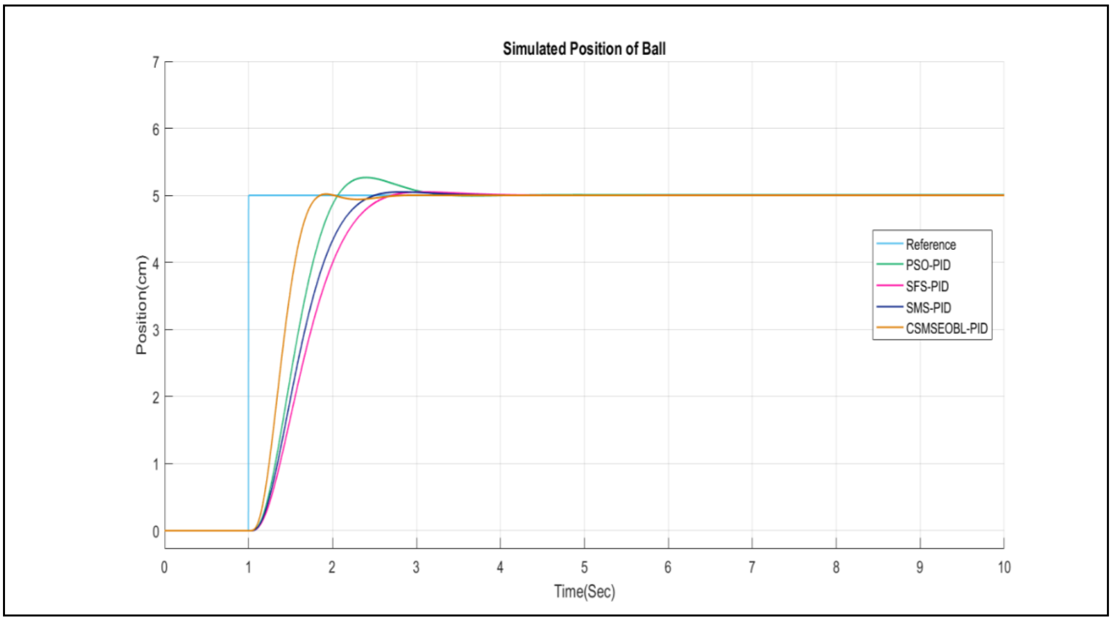


Figure 5.6. Simulated position of ball balancer for different metaheuristic algorithms

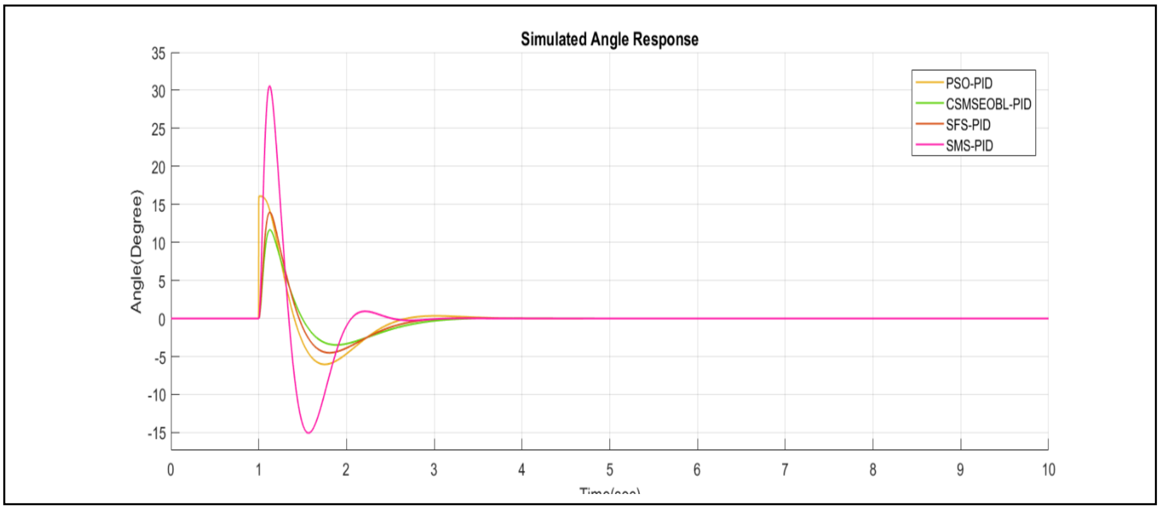


Figure 5.7. Simulated Servo angle response of ball balancer for different metaheuristic algorithms

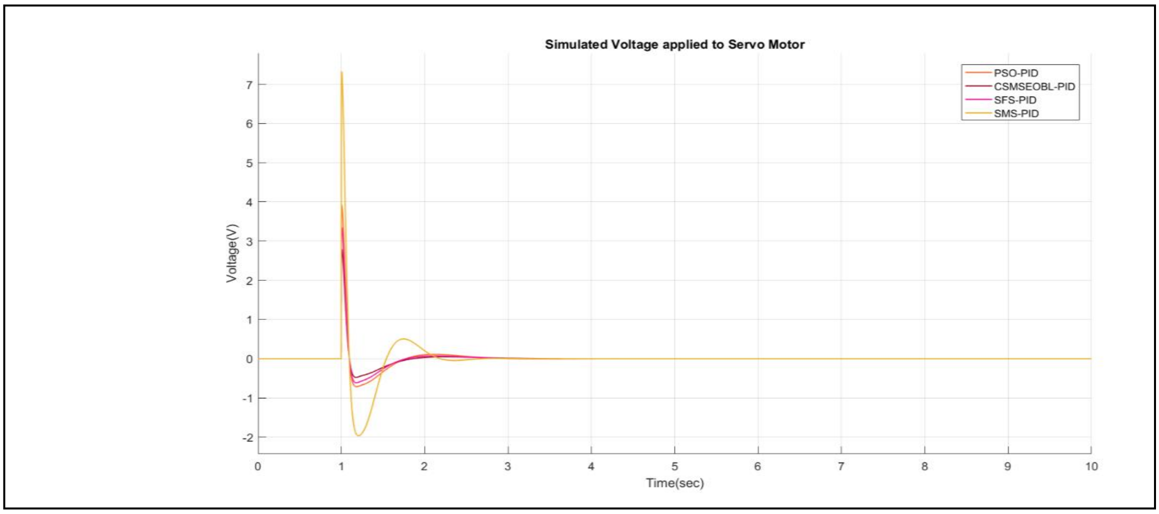


Figure 5.8 The simulated voltage applied to servo motor for different metaheuristic algorithms

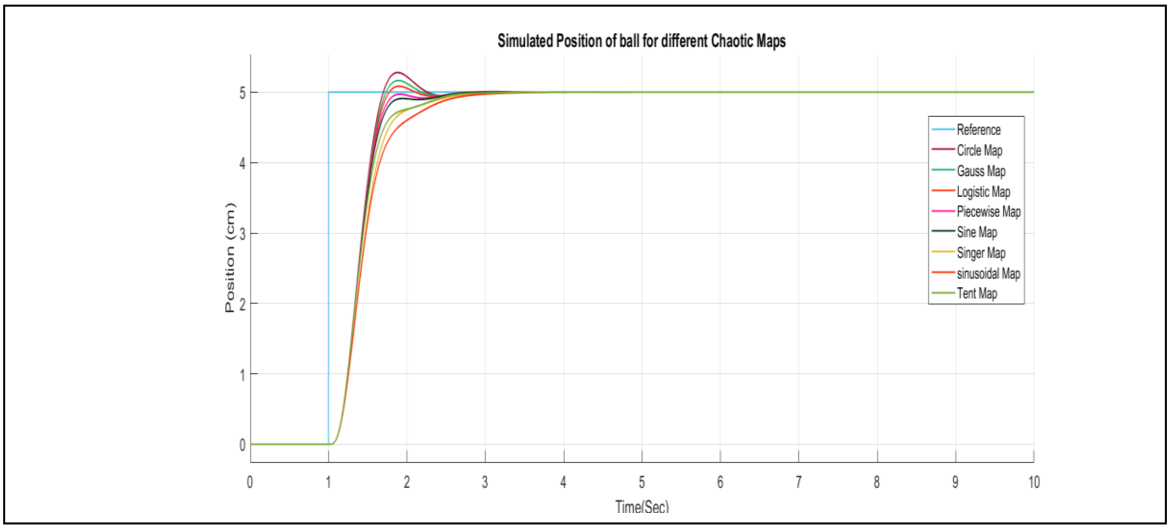


Figure 5.9. Simulated position of the ball with CSMSEOBL algorithm for a variety of Chaotic maps

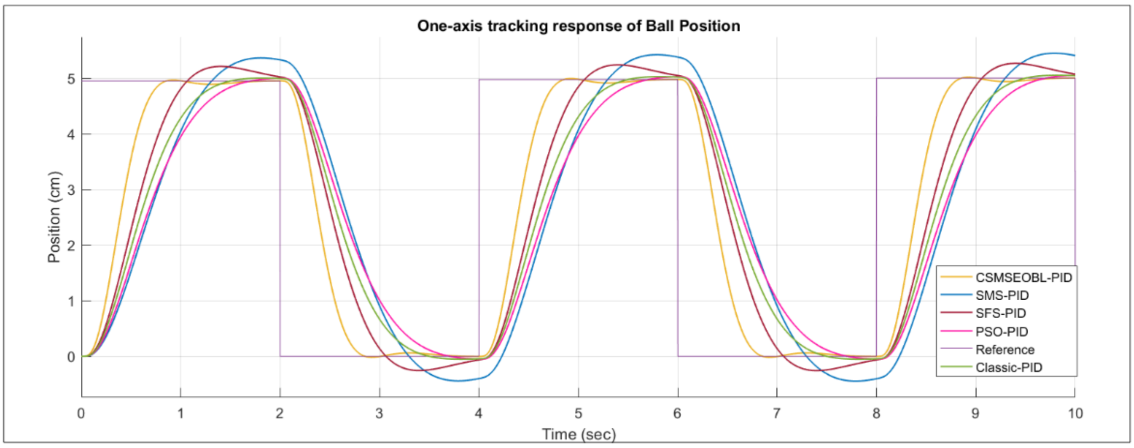


Figure 5.10. Tracking response of ball balancer system

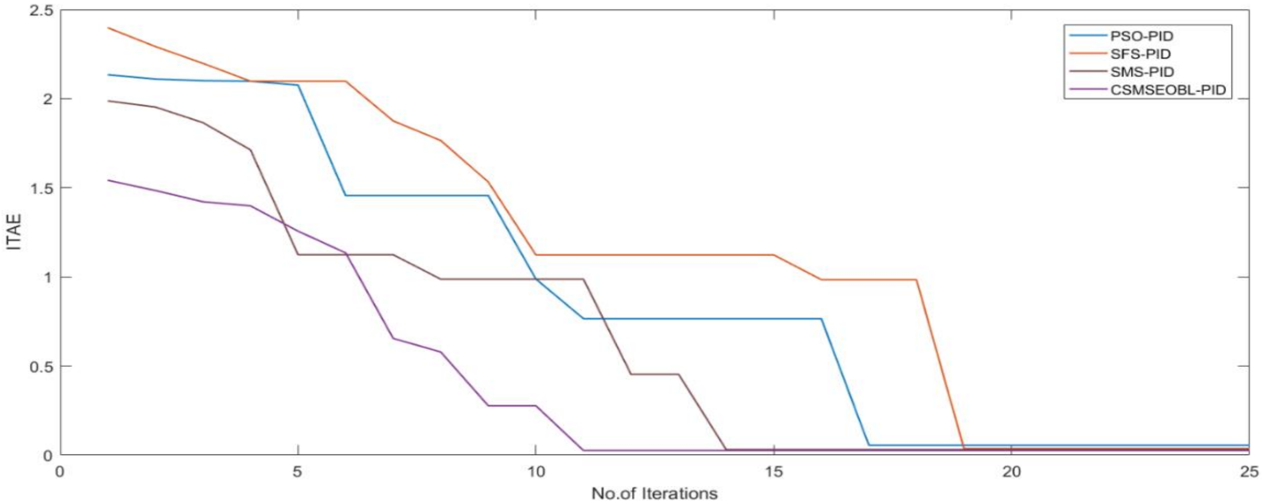


Figure 5.11. ITAE variation for different metaheuristic algorithms

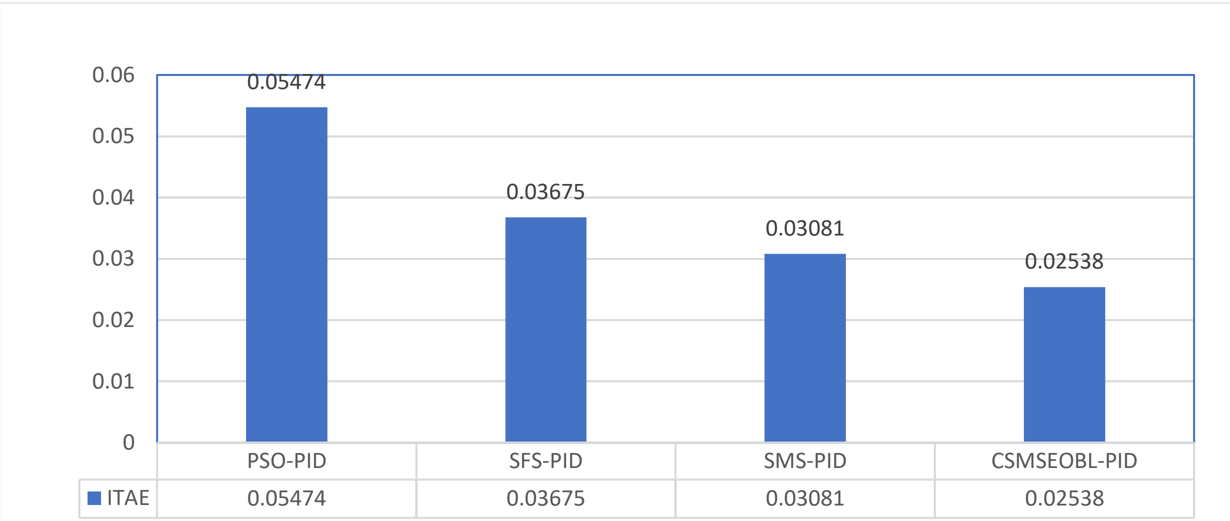


Figure 5.12 Variation of ITAE for different Metaheuristic Algorithms

5.7 Conclusion

CSMSEOBL approach is proposed for the first time for the construction of a PID controller in a Ball balancer system using ITAE as the objective function. This Chapter describes how to tune the parameters of Proportional Integral Derivative control to achieve position and self-balancing control of a two-degree-of-freedom ball balancer system using an improved Elite Oppositional Based Chaotic State of Matter Search Algorithm. Results of simulations demonstrate that the evolved strategy greatly enhances efficiency when used in conjunction with the conventional system.

In addition, a graphical and numerical comparison of the CSMSEOBL-PID approach to other current methodologies is shown in this chapter. When it comes to the ITAE objective function, the

CSMSEOBL-PID technique seems to provide better results than the existing methodologies described in the literature. This approach is used to determine the optimal settling time, rising time, and maximum overshoot possible. The suggested controller demonstrated flexibility and appropriate control efficiency when verified for simulation purposes.

CHAPTER 6

2-DOF-PID Controller tuning for speed control of DC Motor using recent Metaheuristic techniques

6.1 Introduction

In our present industrial revolution, DC motors are used in almost every industrial operation[242]. As a result, it's widely used in a broad variety of industries because of its ease of control, low maintenance costs, and affordable pricing. DC motors are widely used in a variety of industrial settings, including machine tools, paper mills, textile mills, electric traction, and robots. As armature and field windings may be controlled separately, DC motor controller design is much more flexible [243]. It is common for DC motor speed control applications to employ a method where the armature and field winding currents are kept constant or vice versa, resulting in a wide range of desired parameters that can be effectively controlled. Keeping output speed at a predetermined level while monitoring the speed command is the main aim, as well as achieving the necessary speed or position control in the shortest amount of time feasible without excessive overshoots and settle periods [244] [238].

The number of closed-loop transfer functions that may be modified separately defines a control system's degree of freedom [1]. One-degree-of-freedom PID Controller system has the drawback that if an attempt is made to get a disturbance response, it results in the oscillatory response and to get a setpoint response, the disturbance response will get diverged [189]. Because these two requirements contradict and can't be met with a single-degree-of-freedom controller, a two-degree-of-freedom PID controller is used to fulfill them. Set point weighing on the proportional and derivative actions of the 2DOF-PID Controller accomplishes both smooth set-point tracking and excellent disturbance rejection [190].

A hybrid algorithm CSMSEOBL (Chaotic state of matter search with Elite opposition-based learning) is proposed for speed control of dc motor. This hybridization results in the most efficient methodology for locating the global optimal PID Controller parameters concerning the required performance indicators. CSMSEOBL is an enhanced version of the SMS algorithm (state of matter search) that incorporates Chaotic Maps and Elite Opposition-based Learning (EOBL) to increase the algorithm's performance. The underlying principle of the SMS algorithm lies at the heart of the thermal energy motion technique. To enhance SMS convergence, this chaotic SMS algorithm concept is utilized to analyze specific random variables. The SMS algorithm's diversity is increased by combining the two essential features of OBL: a global search and a rapid convergence rate.

The following are the key points of this chapter:

A hybrid CSMSEOBL (Chaotic State of Matter Search with Elite Opposition-based Learning) algorithm is explained.

CSMSEOBL is employed for speed control of dc motor

2-DOF-PID controller parameters are tuned by using the CSMSEOBL algorithm

The performance of the proposed hybrid CSMSEOBL algorithm is compared with other existing metaheuristic algorithms using time response and convergence analysis.

6.2 Mathematical modeling of DC Motor

The DC motor is a popular actuator in process control. It allows for direct rotation and translation when utilized with axles, drums, and wires. The figure below shows the armature's electric circuit and the rotor's free-body diagram (Figure 6.1).

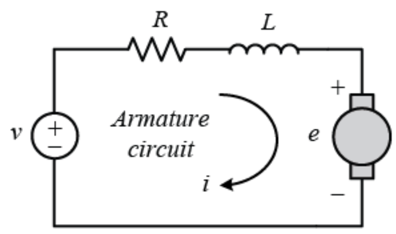


Figure 6.1. Schematic diagram of DC motor [29]

Input to the motor is supply voltage V_S which is applied to the armature of the motor and output is the rotating speed of the shaft. The torque of a DC motor is proportional to armature current and magnetic field intensity. For an armature-controlled motor, it is assumed that the magnetic field is constant thus the torque of the motor (τ_m) is proportional to simply the armature current (I_a) by a constant factor K_a as indicated in the equation below.

$$\tau_m = K_a I_a \tag{6.1}$$

The back emf, E_b has a constant component K_b that is proportional to the shaft's angular velocity ω.

$$E_b = K_b \omega = K_b \frac{d\theta}{dt} \tag{6.2}$$

Based on Newton's 2nd law and Kirchhoff's voltage law, we may obtain the following governing equations from the diagram above.

$$L_a \frac{dI_a}{dt} + R_a I_a + E_b - V_S = 0 \tag{6.3}$$

$$\tau_a = J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = K_a I_a \tag{6.4}$$

From equations (3) and (4), the state space equations need to be formed by considering the 3 output variables (Angular displacement (θ), angular velocity ($\dot{\theta}$), and motor current (I_a)). Each of the variables ($x1 = \theta$, $x2 = \dot{\theta}$, $x3 = I_a$) are substituted from equations (3) and (4).

$$\dot{x}_1 = x_2 \tag{6.5}$$

$$\dot{x}_2 = \frac{-B}{J} x_2 + \frac{K_a}{J} x_3 - \tau_d \tag{6.6}$$

$$\dot{x}_3 = \frac{-K_b}{L} x_2 - \frac{R}{L} x_3 + \frac{V_S}{L} \tag{6.7}$$

Where V_S is the supply voltage, R_a is armature resistance, L_a is armature Inductance, I_a is the armature current, E_b is back emf, τ_a is the armature torque, J is the inertia of the DC motor, B is the viscous damping coefficient [88].

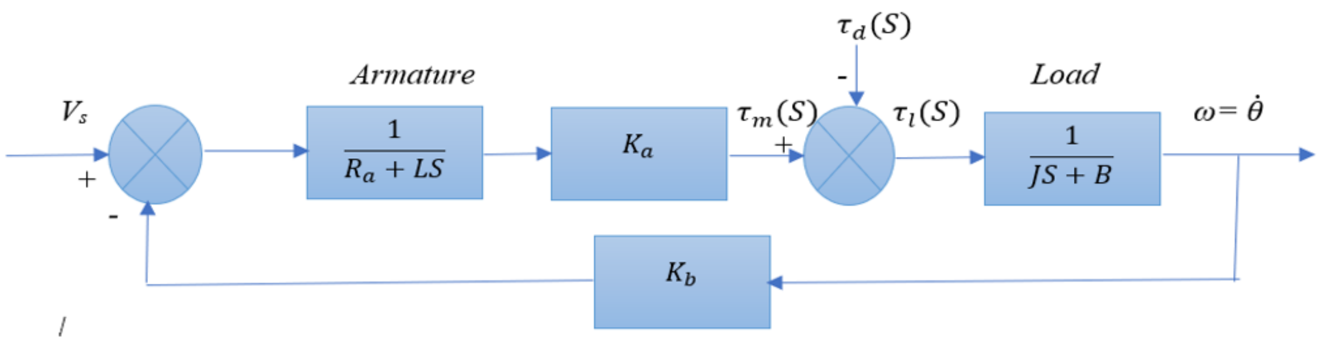


Figure 6.2. Block diagram of DC motor [185]

6.3 PID Controller Tuning

In most situations, the main objective of any controller in the field of control design is to reduce error. PID controllers are preferred by most control engineers for their applications, easy operation and superior performance in the vast majority of instances. The key job for enhanced efficiency is to tune the PID controller [191].

The ZN technique for determining K_p, T_i and T_d is based on the control system's transient reaction. The step response (open loop) approach is used in this investigation [192].In this methodology, parameters are calculated by open-loop step response. Here ‘T’ is the time constant and ‘L’ is the delay time.

Table 6.1 ZN method of PID Controller tuning

Type of Controller	K_p	T_i	T_d
P	T/L		
PI	0.9T/L	L/0/3	
PID	1.2T/L	2L	0.5L

Control system grade of freedom is the number of closed-loop transfer functions that may be altered individually[193]. Either for reference tracking or disturbance rejection, 1DOF PID provides a viable result [194].

6.3.1 Two-Degree of freedom PID Controller

1-DOF Control results in deprived set point tracking for optimization of disturbance response and vice versa. To regulate two control systems at the same time, such as load disturbance and set point, a 2-DOF control system structure is developed (Figure 6.3). Numerous researchers developed various control strategies utilizing a 2-DOF controller e.g. 2-DOF-IMC [195], Control of Uncertain Input-Delay Systems with Input/Output Linearization and 2-DOF[196]and many more.

Multi-objective optimization is the technique of simultaneously optimizing a set of objective functions. Control system design is a multi-objective issue because it includes optimizing several objective functions such as set point responsiveness, different load conditions and model uncertainty tolerance. The quantity of closed-loop transfer functions that may be changed autonomously refers to the degree of freedom in a controller design [197]. When we try to regulate two control system objectives at the same time, such as setpoint value and external disturbances, we get a two-degree-of-freedom control system.

From this block diagram (Figure 6.2)

C(S)- Prime Compensator or serial compensator

$C_f(s)$-Feedforward Compensator

Transfer function from Output Y(s) to set point R(S) is given by

$$G_{Y,R}\,(S) = \frac{P(S)[C(S)+C_f(S)]}{1+P(S)C(S)H(S)} \quad (6.8)$$

The transfer function from Output Y(s) to Disturbance D(s) to is given by

$$G_{Y,D}(S) = \frac{P_d(S)}{1+P(S)C(S)H(S)} \quad (6.9)$$

Assumptions

C(S)- Prime Compensator is a conventional PID Controller

$C_f(s)$-Feedforward Compensator includes only ‘P’ and ‘D’ components of the PID controller for simplicity.

$$C(S) = K_P[1 + \frac{1}{ST_I} + T_D D(S)] \quad (6.10)$$

$$C_f(s) = \; -K_P\{b + cT_D D(s)\} \quad (6.11)$$

where

$K_P, T_I, and\ T_D$ are basic PID Controller parameters

D(S) =Approximate derivative gain= $\frac{S}{1+\tau.S}$

b, c are 2-DOF Parameters.

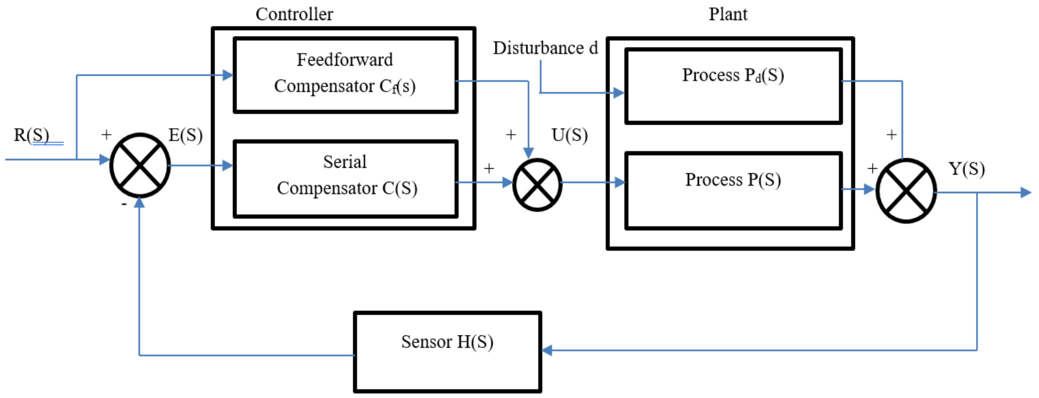


Figure 6.3. Block diagram of 2-DOF-PID Controller[198]

To make the situation easier to understand, we'll use the following two assumptions which apply to a large number of real-world engineering issues with a few exclusions.

Different forms OF 2-DOF Controller

A. Feedforward Controller

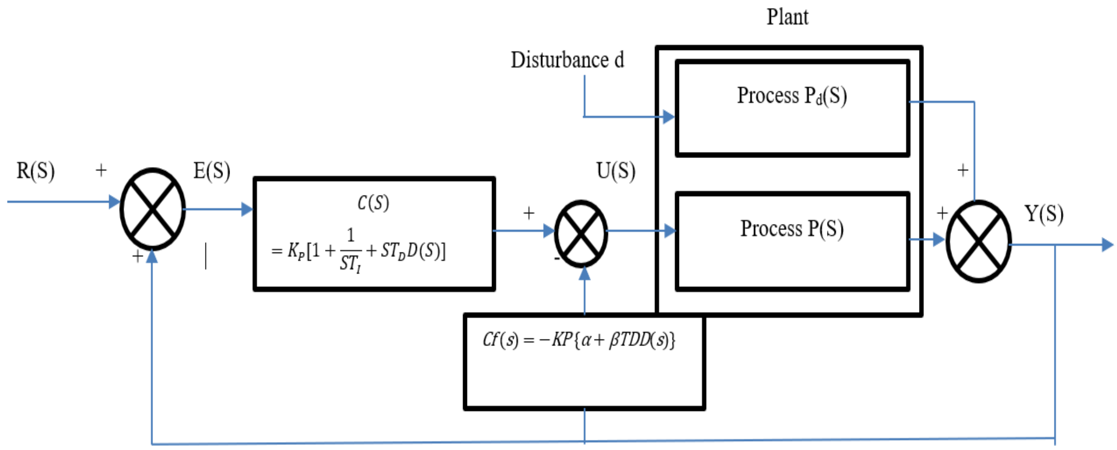


Figure 6.4. Block diagram of 2-DOF-PID Controller in Feedforward mode [197]

Manipulated variable U(s) is given by

$$U(S) = [R(S) - Y(S)]\left[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)\right] - [K_P b + K_P c T_D D(S)] \tag{6.12}$$

$$U(S) = R(S)\left[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)\right] - R(S)[K_P b + K_P c T_D D(S)] - Y(S)[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)] \tag{6.13}$$

$$U(S) = R(S)\left[K_P(1-b) + \frac{K_P}{T_I S} + K_P T_D D(S)(1-c)\right] - Y(S)\,[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)] \tag{6.14}$$

B. Feedback Controller

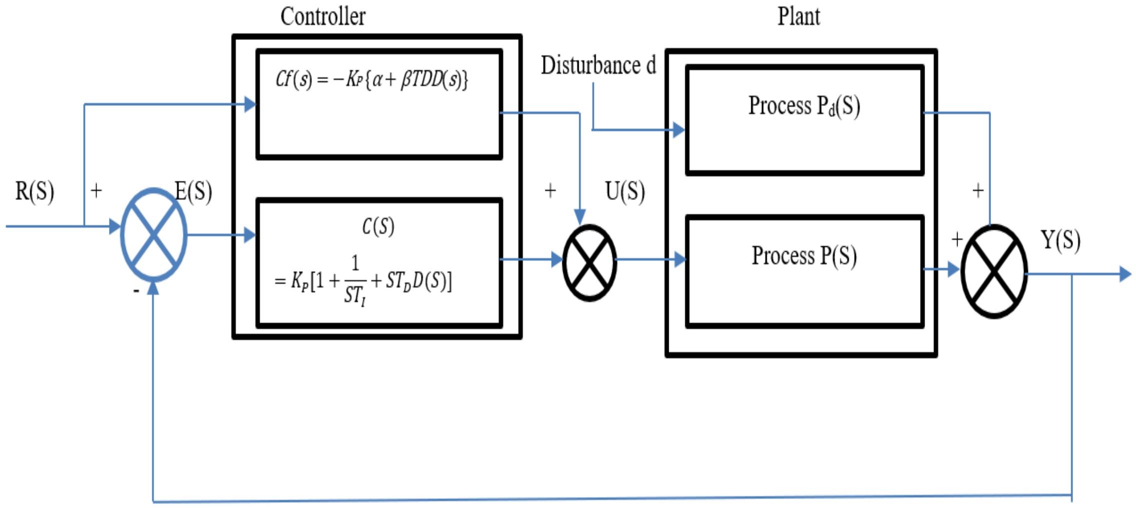


Figure 6.5. Block diagram of 2-DOF-PID Controller in Feedback mode [197]

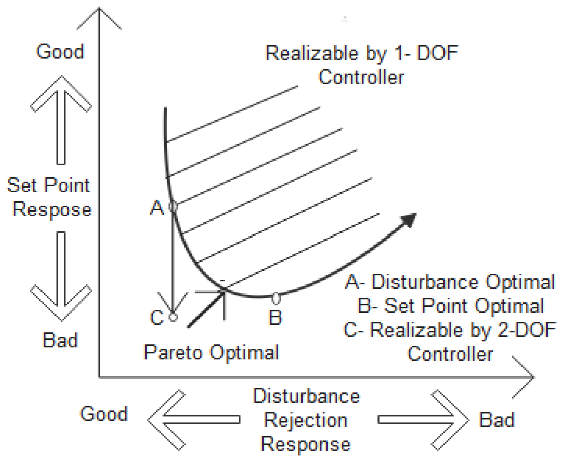
Manipulated variable U(s) for a feedback type 2-DOF controller is given by

$$U(S) = [R(S) - Y(S)]\left[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)(1-c)\right] - Y(S)[K_P b + K_P \beta T_D D(S)] \qquad (6.15)$$

$$U(S) = R(S)[\,K_P(1-b) + \frac{K_P}{T_I S} + K_P T_D D(S)(1-c)] - Y(S)[K_P(1-b) + \frac{K_P}{T_I S} + K_P(1-c)T_D D(S)] - Y(S)[K_P b + K_P c T_D D(S)] \qquad (6.16)$$

$$U(S) = R(S)[\,K_P(1-b) + \frac{K_P}{T_I S} + K_P T_D D(S)(1-c)] - Y(S)[K_P - bK_P + \frac{K_P}{T_I S} + K_P T_D D(S) + K_P b - K_P c T_D D(S) + K_P c T_D D(S)] \qquad (6.17)$$

$$\mathrm{U}(S) = R(S)[\,K_P(1-b) + \frac{K_P}{T_I S} + K_P T_D D(S)(1-c)] - Y(S)[K_P + \frac{K_P}{T_I S} + K_P T_D D(S)] \qquad (6.18)$$



-

Figure 6.6. Effect of 2-DOF Controller Structure [198]

The scenario described above can be theoretically represented in Figure 6.6. The traditional 1-DOF PID controller can only achieve the sketched area. As a result, we can't optimize both the set-point and disturbance responses at the same time. This limitation leads the researchers to select one of the following alternatives:

a. To select one of the Pareto best points from the region AB.
b. To utilize the disturbance optimized values and establish a boundary on the set-point variable's modification.

The second option was good enough in the early days of industrial applications when the set-point value was not altered very often. As a result, several of the best tuning techniques[199] only yielded the "disturbance optimal" values.

However, in recent times, the scenario has altered and industrial control systems are now expected to adjust the set-point variable regularly. In such a circumstance, a 2-DOF-PID controller is a valuable tool. It allows the user to make both the set-point and disturbance responses inside the linear framework effectively optimum at the same time.

6.4 Metaheuristic Algorithms

6.4.1 Water Wave optimization algorithm

The WWO approach handles the optimization problem by modeling wave motion, propagation, refraction, and breaking. A wave's fitness value is related to its height and wavelength. As seen in Figure 6.7, shallow water waves have a higher fitness value, higher wave height, and longer wavelength than deep water waves.[200].

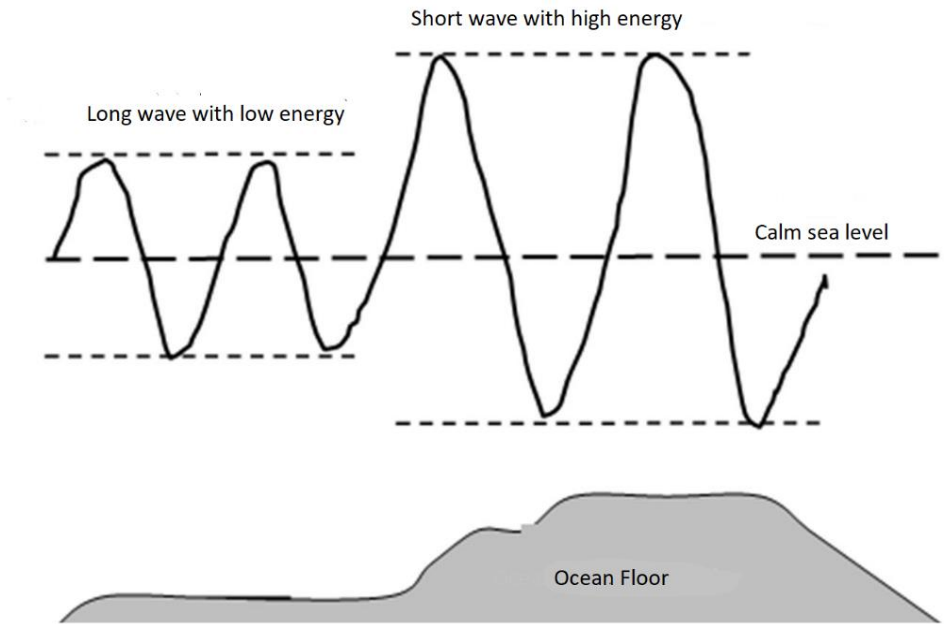


Figure 6.7 Deep and shallow sea waves having different forms

Table 6.2 Problem-population space relationship

Problem space	**Population space**
Practical problem	Wave model for shallow water
The problem's search space	Seabed area
Solve each solution to the problem	A water wave with height *h* and wavelength λ

The value of each solution's evaluation function	The greater the fitness value, the closer the water level is to the bottom. The lower the fitness value, on the other hand, the better

Propagation

Because the seafloor is uneven, wave height and wavelength will vary appropriately as each wave propagates. The initial wave x is propagated to produce the new water x'.

$$x'(d) = x(d) + r(-1,1).\lambda L(d) \tag{6.19}$$

where

$r(-1,1)$-random number distributed uniformly

$L(d)$ −length of d[th] search space

λ −water wavelength of wave x

As shown in Figure 1, as a wave goes from deep water (low fitness region) to shallow water (high fitness region), the wave height rises and the wave length decreases. We compute the fitness of the descendant wave x' post propagation. If f (x') > f (x), the population replaces 'x' with x' and the wave height of x' is adjusted to h_{max}. Otherwise, 'x' remains the same, but its height 'h 'is reduced by one, simulating energy loss owing to inertial resistance, vortex shedding and surface friction [200].

Wavelength λ is updated according to Equation (6.20)

$$\lambda = \lambda \alpha^{-\frac{f(x)-f_{min}+\epsilon}{f_{max}-f_{min}+\epsilon}} \tag{6.20}$$

where

α − attenuation constant of wavelength

f_{min} −minimum fitness value

f_{max} −maximum fitness value

Breaking

As the wave power increases, the crests get steeper and steeper. When the peak speed surpasses the wave propagation speed, the wave breaks into separate waves. The optimum water wave x is accomplished by breaking operations in the WWO algorithm to increase population variety. The formula for updating the location is as follows:

$$x'(d) = x(d) + N(0,1).\beta L(d) \tag{6.21}$$

β −broken wave coefficient

If the fitness value of all solitary waves created by the breaking operation is better than the water wave x *, then x * is retained.

Refraction

During wave propagation, the energy of the water wave continues to diminish until it reaches zero. This improves the model's efficacy and speeds up the WWO algorithm's convergence. The refraction process formula for water wave x is as follows:

$$x'(d) = N\left(\frac{x^*(d)+x(d)}{2}\right), N(\frac{x^*(d)-x(d)}{2}) \tag{6.22}$$

Where

x^* −optimal solution

$N(\mu,\sigma)$ −Gaussian random number having μ mean and σ standard deviation

This time wave height x' is h_{max}

The wavelength is updated as

$$\lambda' = \lambda \frac{f(x)}{f(x')} \tag{6.23}$$

The basic goal of the WWO algorithm is to replicate the dynamic motion of a water wave and find the global best solution in the search space. The process takes place from deep water to shallow seas. Waves with lower fitness values, longer wavelengths and lower wave heights occur in deep water; waves with higher fitness values, shorter wavelengths and greater wave heights occur in shallow water. For the water wave optimization technique, propagation, breaking and refraction give an effective search mechanism.

The depth of the algorithm is increased by local search, while the breadth of the algorithm is increased by global search.

Water waves with a greater fitness value can do a local search in a narrow range, whereas water waves with a lower fitness value may perform a global search in a broad range due to propagation. The breaking converts the ideal water waves into a sequence of isolated waves, which increases the algorithm's calculation accuracy and allows for a more intensive search for the global optimal solution. The refraction may eliminate energy-depleted waves, thus avoiding search stagnation and increasing population variety, creating better waves to replace the bad waves and speeding up the algorithm's convergence speed.

The water wave optimization algorithm's global and local searches are successfully balanced by the combination of three procedures.

6.5 Simulation Results and Discussion

In the MATLAB/Simulink environment, assessments of meta-heuristic algorithms are carried out. The results were obtained after 30 runs of each method on a laptop running 64-bit Windows 10, Intel(R) Core i5 processor, CPU @1.30GHz, 1.5 GHz and 8GB RAM. Tables 6.2 provide the initialization settings for the variables that are held constant after each run of the metaheuristic algorithms' code execution.

6.5.1 Tuning of the 2-DOF-PID controller by using the CSMSEOBL algorithm

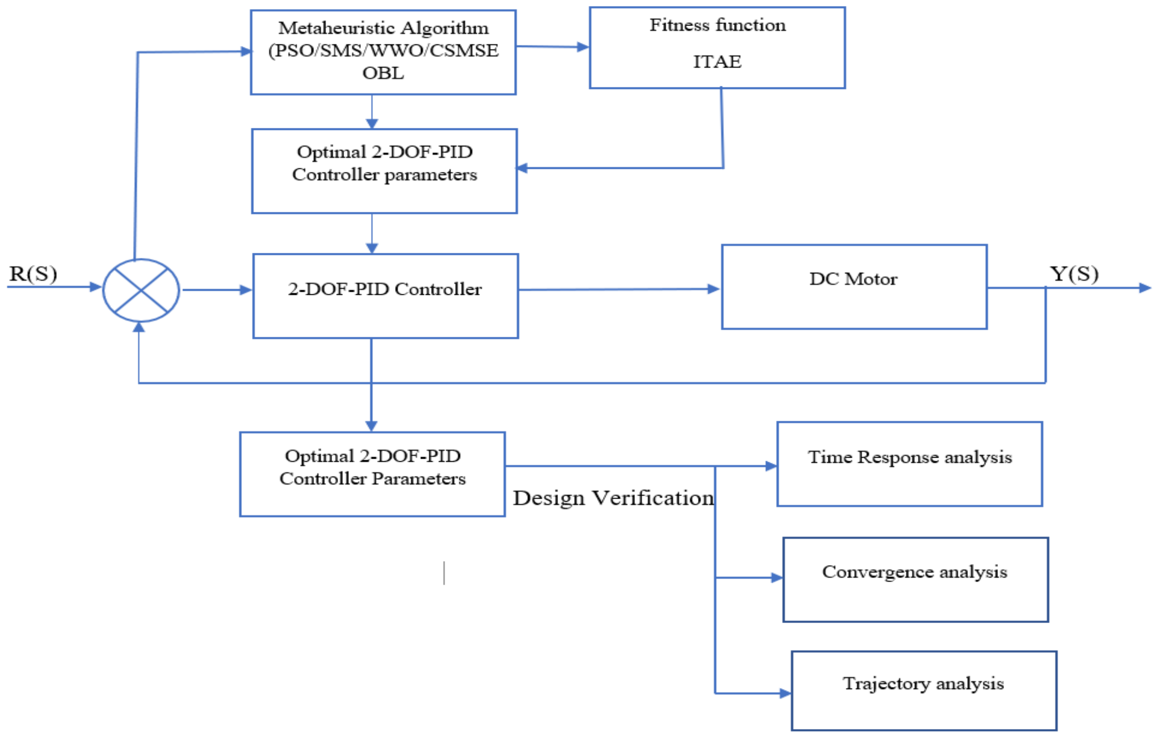


Figure 6.8. Block diagram of 2-DOF-PID controller tuning using CSMSEOBL algorithm

In this simulation, a DC motor with a 2-DOF-PID controller and ITAE as a fitness function is presented which is shown in Figure 6.8. It has a more complicated architecture than a traditional PID controller. The ITAE goal function is minimized in controller synthesis. The goal of this research is to get a DC motor's speed as near to the set point condition as possible.

The conventional PID Controller is used to determine the values of the PID controller at first, as mentioned in Table 6.1. Furthermore, metaheuristic algorithms such as PSO, WWO, SMS and eventually the hybridized CSMSEOBL optimize the 2-DOF-PID controller settings by calculating the difference between the predicted and calculated speed of the DC motor. Parameter settings for all the above-mentioned algorithms are shown in Table 6.3.

Table 6.3 Parameter Setting for different Metaheuristic Algorithms

Algorithm and Parameters	**Parameter Value**	**Algorithm and Parameters**	**Parameter Value**
PSO		**WWO**	
Population	50	Population	50
Iteration	100	Iteration	100
Weight Function	[0.2,0.9]	α = 1.0026	$\beta \in [0.01, 0.25]$
Acceleration constants	2	Wave height h_{max}	12
The dimension of search space	5	Random number $k$max	min(12, D/2)
CSMSEOBL		Wavelength λ	0.5
Vector Adjustment, ρ	1	The dimension of search space	5
Beta	[0.8, 0.4, 0.1]	**SMS**	
Alpha	[0.8, 0.2, 0]	Vector Adjustment, ρ	1
Threshold Probability, H	[0.9, 0.2, 0]	Beta	[0.8, 0.4, 0.1]
Phase Percent	[0.5, 0.1, -0.1]	Alpha	[0.8, 0.2, 0]
Adjustment Parameters	[0.85 0.35 0.05]	Threshold Probability, H	[0.9, 0.2, 0]

a = 0.02	*b* = 20	Phase Percent	[0.5, 0.1, -0.1]
No. of Chaotic maps	10	Adjustment Parameters	[0.85 0.35 0.05]

6.5.2 Time response analysis

Table 6.4 Time response specifications for different metaheuristic algorithms

Name of Algorithm	Kp	Ki	Kd	b	c	T_f	Rise Time	Settling Time	Overshoot	Peak Time
PSO-2-DOF-PID	5.68	16.843	-0.0054	0.0078	0.0078	2390	0.7622	1.3561	1.8399	5.2932
WWO-2-DOF-PID	8.34	26.94	-0.0039	0.0065	0.0065	2360	0.6932	1.2347	1.2327	5.2731
SMS-2-DOF-PID	1.64	9.64	-0.0083	0.0065	0.0065	2165	0.406	5.613	4.4215	5.2206
CSMSEOBL-2-DOF-PID	1.88	52.3	-0.0096	0.0047	0.0047	2290	0.0763	0.139	1	5.0804

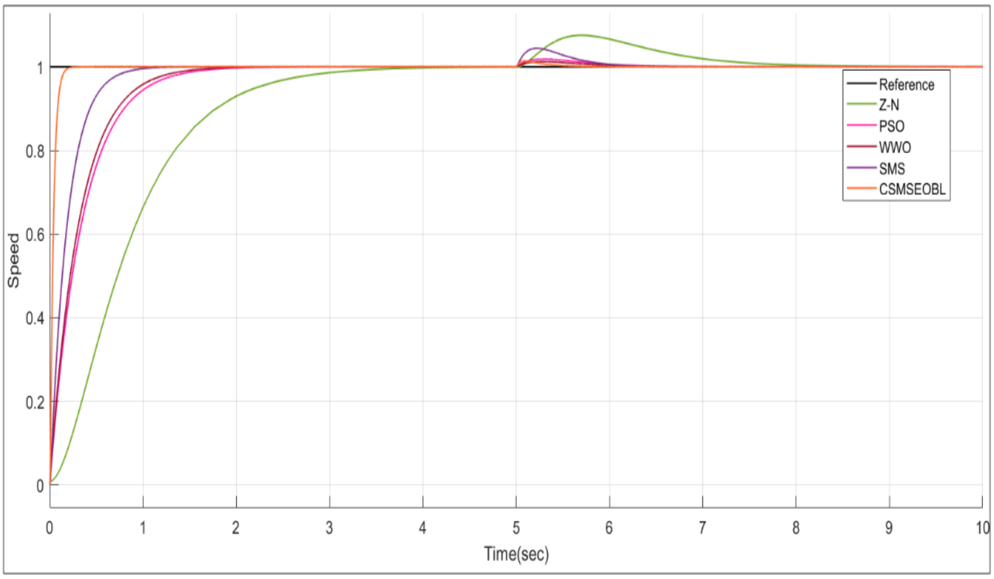


Figure 6.9. Step response analysis for speed control of DC motor using different metaheuristic algorithms

It is clear from Figure 6.9 that CSMSEOBL tuned 2-DOF-PID Controller outperform the other 2-DOF-PID Controller tuning methods in term of settling time and maximum overshoot.

6.5.3 Convergence Analysis

Table 6.5 ITAE and Standard Deviation for different metaheuristic algorithms

Algorithm	**Fitness Function Value** **ITAE**	**Std. deviation in ITAE**
PSO-2-DOF-PID	0.2028	.4521
WWO-DOF-PID	0.1516	.05137
SMS-2-DOF-PID	0.1764	.06908
CSMSEOBL-2-DOF-PID	0.02612	.01211

After 100 iterations, these optimization techniques converge to the lowest value of the fitness function shown in Table 6.5. The convergence of the meta-heuristic method for the optimum simulation run is shown in Figure 6.15. It has been discovered that the proposed techniques require fewer iterations to converge to an optimum PID controller. The CSMSEOBL-2-DOF-PID appears to converge faster than the WWO-2DOF-PID, SMS-2DOF-PID and PSO-2-DOF-PID.

While CSMSEOBL-2-DOF-PID converged to the ITAE of 0.02612 in only 24 iterations, WWO-2-DOF-PID required roughly 56 iterations to attain a similar result. In terms of transient response criterion and optimized Fitness Function value, CSMSEOBL-2-DOF-PID and WWO-2-DOF-PID beat ZN and PSO methods. Furthermore, compared to PSO-PID, the suggested technique requires fewer iterations to get the best ITAE value (Figure 6.14).

Set point tracking responses using different existing and suggested hybrid algorithms are shown in Figure 6.10, Figure 6.11, Figure 6.12 and Figure 6.13 respectively and these Responses clearly show that CSMEOBL tuned 2-DOF-PID controller results in the best method as compared to other.

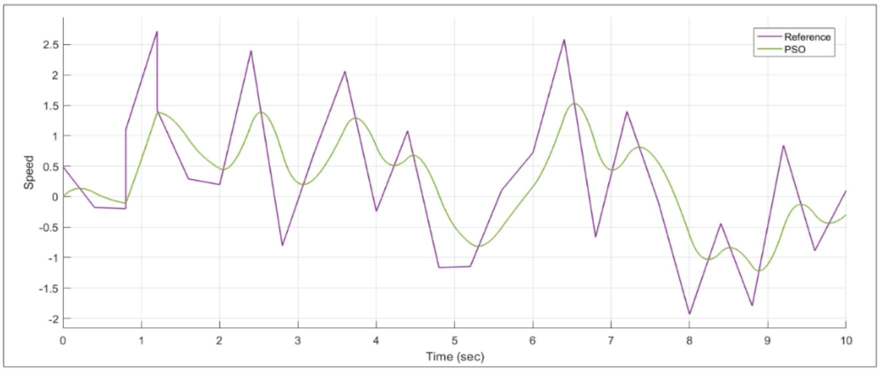


Figure 6.10 Set point tracking response using PSO tuned 2-DOF-PID Controller

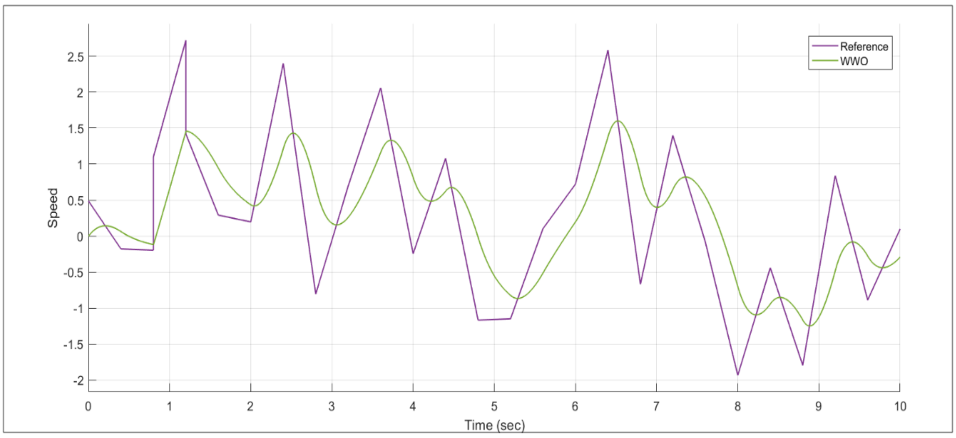


Figure 6.11 Set point tracking response using WWO tuned 2-DOF-PID Controller

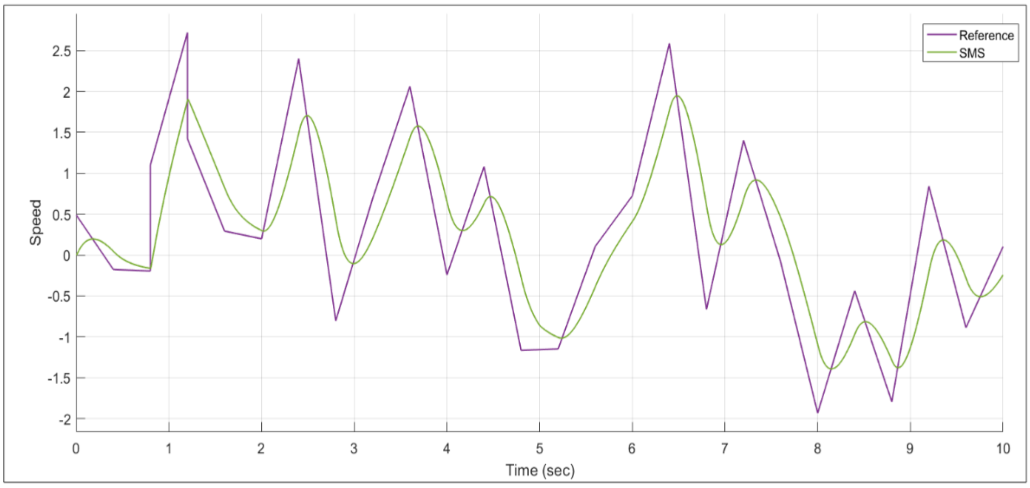


Figure 6.12 Set point tracking response using SMS tuned 2-DOF-PID Controller

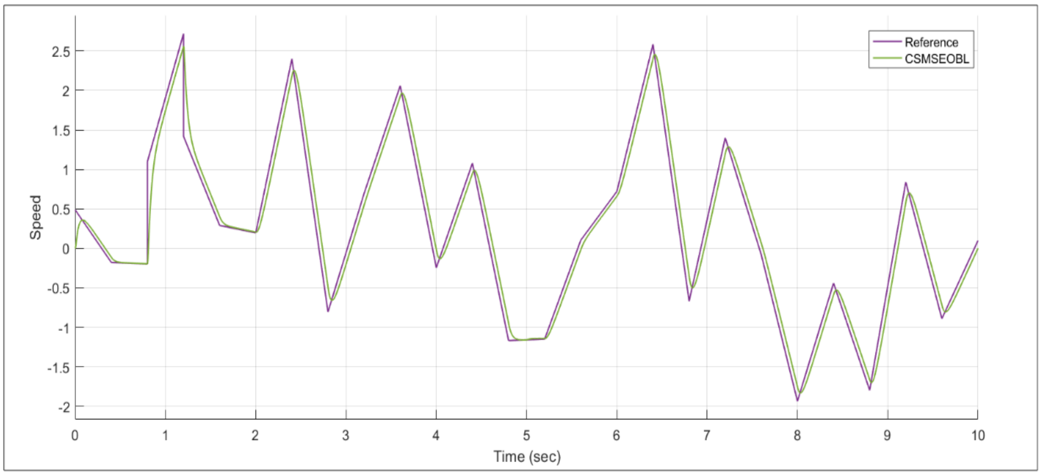


Figure 6.13 Set point tracking response using CSMSEOBL tuned 2-DOF-PID Controller

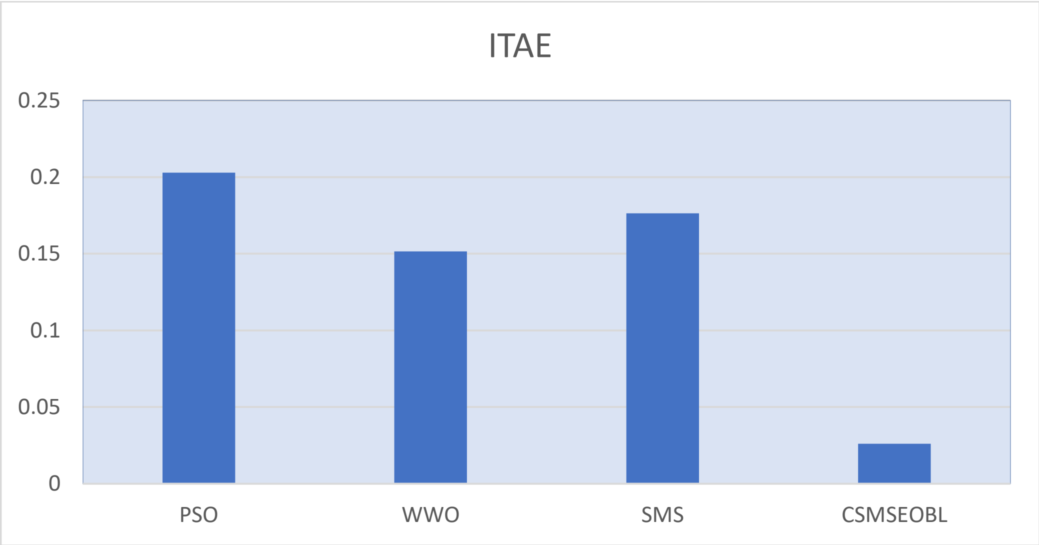


Figure 6.14 ITAE variation for various metaheuristic algorithms

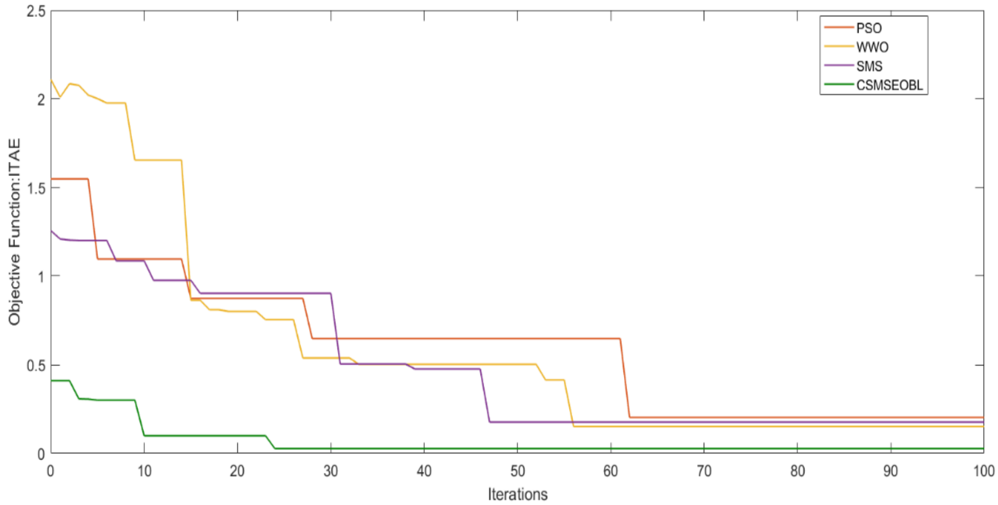


Figure 6.15 Comparative analysis of ITAE convergence using different metaheuristic algorithms

The above analysis leads to the conclusion that the suggested tuning methods offer PID controller settings with a reduced ITAE value which is a desirable characteristic. For 30 individual runs, the CSMSEOBL-2-DOF-PID controller had the lowest ITAE and the lowest standard deviation. As a result, based on the ITAE fitness function the CSMSEOBL-2-DOF-PID controller is the most accurate meta-heuristic algorithm. Similarly, CSMSEOBL-2-DOF-PID shows low variation in the results between runs demonstrating the algorithm's strong repeatability. In terms of fitness function value, WWO-2DOF-PID is ranked second behind CSMSEOBL-2-DOF-PID.

6.6 Conclusion

The optimal gain settings of a PID controller to govern the rotational speed of a DC motor are provided in this paper using a new hybrid technique. Meta-heuristic methods are used to reduce the ITAE fitness function throughout the controller design process. The efficacy of meta-heuristic algorithms is evaluated using the transient response characteristics and set point tracking of a DC motor speed control system. In comparison to other existing methods, the simulation results show that using the CSMSEOBL-based 2-DOF-PID control strategy with ITAE as an objective function leads to decreased settling time and overshoot. As a result, the suggested approach may be used to confirm that 2-DOF-PID controllers work optimally in big electrical systems, the manufacturing industry, and the robotics sector, among other applications.

Chapter 7

Conclusion and further scope of work

7.1 Introduction

As a result of successful implementations and high intensity, research on metaheuristics has been extensively published in the literature. Though little information has been gathered on in-depth research of metaheuristic performance concerns, it remains a "black box" as to why some metaheuristics perform better on particular optimization tasks while others do not.

CSMSEOBL is a novel hybrid metaheuristic algorithm that is developed in this thesis and tested on some benchmark functions as well as nonlinear control problems to demonstrate its efficiency and effectiveness. In addition, a comparative study with other recent metaheuristic algorithms is presented to demonstrate its superiority. This chapter provides a conclusion as well as a discussion of the future potential in this field.

7.2 Contributions of work

The conclusion of the work is being presented in a chapter-wise manner and is given below:

Chapter 1 concludes that optimizing the parameters of nonlinear systems is one of the most challenging elements in control theory. Metaheuristic techniques have been employed to address this issue. They are advantageous when working with complex systems. In comparison to deterministic algorithms, metaheuristic optimization approaches perform very well in tackling nonlinear control problems with unknown search spaces. These optimization techniques have been used in almost every field of science, technology, and engineering to choose the optimum solution from a pool of plausible alternatives.

The literature reviewed in Chapter 2 covers a wide variety of topics, including recent metaheuristic algorithms, the hybridization of two or more metaheuristics, surveys, comparisons, and performance analysis, Variants of controlling techniques as well as a broader range of applications in fields such as process control, electromechanical systems, engineering, etc.

Chapter 3 conclude the mathematical and conceptual description of the proposed metaheuristic optimization technique i.e., CSMSEOBL. Some validity criteria have been considered to examine the efficacy of metaheuristics. The median, mean, normalized value and standard deviation of objective function values acquired over a certain number of runs are among these. In addition to these approaches, statistical analysis techniques have been used, which is the Wilcoxon test p-value. The CSMSEOBL algorithm converged much quicker than the PSO, SMS, WWO, SFS and CS algorithms, according to the findings of unimodal benchmark functions. Similar behaviour was found in the multimodal benchmark functions, demonstrating the proposed algorithm's strong diversification and avoidance of local optima. According to the findings of composite test functions, CSMSEOBL outperformed other algorithms, demonstrating that this algorithm could effectively balance diversification and intensification to find the global optima of complex test functions. The findings of performance measurements revealed that CSMEOBL's search agent needed to modify quickly in the early stages of optimization and progressively in the latter stages. The findings revealed that this approach resulted in comprehensive diversification of the search space and intensification of the most promising parts. The enhancement of the initial random population and the optimal solution produced so far (convergence) by CSMSEOBL is further proved and validated by the average fitness of solutions and convergence curves.

A Continuously Stirred Tank Reactor (CSTR) is described in Chapter 4 as an example of how the suggested hybrid algorithm 'CSMSEOBL' may be used to set the FOPID Controller's parameters for temperature and concentration control. The following are the most significant results of the investigation: It improves the functionality for exploration and exploitation; when implemented to a nonlinear control problem, it results in faster convergence; CSMSEOBL also exhibits promising results in terms of overshoot, settling time, and ITAE for performance optimization.

In Chapter 5, the CSMSEOBL method is proposed for tuning PID controller parameters in a Ball balancer system using ITAE as the fitness function. Using an updated Elite Oppositional Based Chaotic State of Matter Search Algorithm, the parameters of Proportional Integral Derivative control are tuned to accomplish position and self-balancing control of a balancer device with two degrees of freedom. The results of the simulations reveal that the evolved strategy greatly enhances the efficiency of this nonlinear system.

With the new hybrid CSMSEOBL algorithm, the optimal gain settings of a 2-DOF-PID controller for managing the rotational speed of a DC motor are described in Chapter 6. To lower the ITAE fitness function, CSMSEOBL and other current Meta-heuristic approaches are employed. Based on the transient response characteristics and set point tracking of a direct current motor speed control system, the effectiveness of the proposed hybrid meta-heuristic algorithms is determined. When compared to other existing methods, the CSMSEOBL-based 2-DOF-PID control strategy using ITAE as the objective function results in a shorter settling time and less overshoot than the other methods used.

7.3 Suggestions for further work

With various research recommendations for future investigations, this study may be expanded and i mproved:

- Intelligent sampling and surrogate modelling are two more major areas that demand the researcher's attention. Surrogate techniques aid metaheuristics in work evaluations of enormously computationally costly capacities by estimating the actual objective function, while intelligent sampling reduces the limits of issue space for confined looking to best neighborhoods, this region has yet to be thoroughly investigated, thus there is a lot of room for investigation.
- Scalable metaheuristic algorithms that can adapt and fine-tune themselves may be used to solve big optimization problems with enormous variable selection.
- To increase its performance, the CSMSEOBL technique may be combined with other stochastic optimization algorithms. Finally, CSMSEOBL's performance on various real-world engineering optimization tasks may be evaluated.
- Even if algorithms are efficient, adequate implementation and parallelization may improve their practical use.
- The use of other recently developed metaheuristic optimization techniques, such as the Honey Badger Algorithm, the Dingo Optimizer, Artificial Gorilla Troops Optimizer, the Rocks Hyraxes Swarm Optimizer, and many others may be employed for the control of CSTR, Ball Balancers, Direct Current motors, and other nonlinear control problems.

The performance study of CSMSEOBL and other contemporary metaheuristic optimization approaches may benefit from the inclusion of other emerging nonlinear control problems such as inverted pendulum, 2-DOF-Helicopter, Cruise control systems and many more.

References

[1] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.

[2] W. G. Wolpert, D. H. and Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67–82, 1997.

[3] X. S. Koziel, S. and Yang, *Computational Optimization, Methods and Algorithms, Springer, Germany.* 2011.

[4] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley & Sons.* .

[5] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms, First Edition, Luniver Press, UK*. 2008.

[6] S. Desale, A. Rasool, S. Andhale, and P. Rane, “Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey,” *Int. J. Comput. Eng. Res. Trends*, vol. 351, no. 5, pp. 2349–7084, 2015, [Online]. Available: http://www.ijcert.org.

[7] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley and Sons, Hoboken, NJ, USA.* 2010.

[8] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Algorithms in Modeling and Optimization*, no. December. 2013.

[9] Y. Wang, “A Sociopsychological Perspective on Collective Intelligence in Metaheuristic Computing,” *Int. J. Appl. Metaheuristic Comput.*, vol. 1, no. 1, pp. 110–128, 2010, doi: 10.4018/jamc.2010102606.

[10] X. S. Yang, “Metaheuristic optimization: algorithm analysis and open problems,” 2011.

[11] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Algorithms in Modeling and Optimization*, no. May. 2013.

[12] W. G. Wolpert, D.H., Macready, “No Free Lunch Theorems for Optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 67, 1997.

[13] W. G. Wolpert, D.H., Macready, “No Free Lunch Theorems for Optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[14] P. Boussaïd, I., Lepagnot, J., & Siarry, “A survey on optimization metaheuristics.,” *Inf. Sci. (Ny).*, vol. 237, no. 82–117, 2013.

[15] J. Rajpurohit, T. K. Sharma, A. Abraham, and Vaishali, “Glossary of metaheuristic algorithms,” *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 9, pp. 181–205, 2017.

[16] E. R. Kennedy J, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, 1995, pp. 1942–1948.

[17] H. S. Urade and R. Patel, “Study and Analysis of Particle Swarm Optimization : A Review,” *2nd Natl. Conf. Inf. Commun. Technol.*, pp. 1–5, 2011.

[18] X. S. Yang and S. Deb, “Multiobjective cuckoo search for design optimization,” *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, 2013, doi: 10.1016/j.cor.2011.09.026.

[19] M. . Cuevas, E., Echavarria, A., Ramirez-Ortegon, “An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation,” *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, 2014.

[20] R. Zhao, Q. Luo, and Y. Zhou, “Elite opposition-based social spider optimization algorithm for global function optimization,” *Algorithms*, vol. 10, no. 1, 2017, doi: 10.3390/a10010009.

[21] H. R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *IEEE*, 2005, pp. 695–701.

[22] S. Zhang, Q. Luo, and Y. Zhou, “Hybrid Grey Wolf Optimizer Using Elite Opposition-Based Learning Strategy

and Simplex Method," *Int. J. Comput. Intell. Appl.*, vol. 16, no. 2, pp. 1–37, 2017, doi: 10.1142/S1469026817500122.

[23] Y. J. Zheng, "Water wave optimization: A new nature-inspired metaheuristic," *Comput. Oper. Res.*, vol. 55, no. March, pp. 1–11, 2015, doi: 10.1016/j.cor.2014.10.008.

[24] M. A. Ajwad SA, Iqbal J, Ullah MI, "A systematic review of current and emergent manipulator control approaches.," *Front Mech Eng China*, vol. 10, pp. 198–210, 2015.

[25] A. C. Menini L, Zaccarian L, *Current trends in nonlinear systems and control - In Honor of Petar Kokotovic and Turi Nicosia: Springer*. 2006.

[26] L. Suo, J. Ren, Z. Zhao, and C. Zhai, "Study on the nonlinear dynamics of the continuous stirred tank reactors," *Processes*, vol. 8, no. 11, pp. 1–18, 2020, doi: 10.3390/pr8111436.

[27] S. P. and S. V. M. P. G. Glandevadhas, "Optimum Setting of Controller Using Soft Computing Techniques for a Chemical System," *Am. J. Appl. Sci.*, vol. 8, no. 11, pp. 1140–1148, 2011.

[28] A. K. Pinagapani, G. Mani, K. R. Chandran, and K. Pandian, "Composite Disturbance Rejection Control for Ball Balancer System," *Procedia Comput. Sci.*, vol. 133, pp. 124–133, 2018, doi: 10.1016/j.procs.2018.07.016.

[29] B. B. Acharya, S. Dhakal, A. Bhattarai, and N. Bhattarai, "Pid speed control of dc motor using meta-heuristic algorithms," *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 2, pp. 822–831, 2021, doi: 10.11591/ijpeds.v12.i2.pp822-831.

[30] T. Kara and I. Eker, "Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments," *Energy Convers. Manag.*, vol. 45, no. 7–8, pp. 1087–1106, 2004, doi: 10.1016/j.enconman.2003.08.005.

[31] C. Blum and A. Roli, "Hybrid metaheuristics: An introduction," *Stud. Comput. Intell.*, vol. 114, no. 2008, pp. 1–30, 2008, doi: 10.1007/978-3-540-78295-7_1.

[32] T. O. Ting, X. S. Yang, S. Cheng, and K. Huang, "Hybrid metaheuristic algorithms: Past, present, and future," *Stud. Comput. Intell.*, vol. 585, no. December, pp. 71–83, 2015, doi: 10.1007/978-3-319-13826-8_4.

[33] A. S. R. Masadeh, A. Alzaqebah, "Whale Optimization Algorithm for Solving the Maximum Flow Problem," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 8, pp. 2208–2220, 2018.

[34] M. S. K. P. Sindhuja, P. Ramamoorthy, "A Brief Survey on Nature Inspired Algorithms: Clever Algorithms for Optimization," *Asian J. Comput. Sci. Technol.*, vol. 7, no. 1, pp. 27–32, 2018.

[35] P. R. and J. F. Miller, "A Review of Hyper-Heuristic Frameworks," in *Electronic Village Online - Evo20 Workshop, American International School of Bucharest (AISB)*, 2014, pp. 1–7.

[36] N. K. T. El-Omari, "Sea Lion Optimization Algorithm for Solving the Maximum Flow Problem," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 20, no. 8, pp. 30–67, 2020.

[37] M. Gavrilas, "Heuristic and metaheuristic optimization techniques with application to power systems," *Int. Conf. Math. Methods Comput. Tech. Electr. Eng. - Proc.*, pp. 95–103, 2010.

[38] Polya, "Reviewed Work: How to Solve It by G. Pólya," *The Mathematical Gazette*, vol. 30. p. 181, 1945, doi: 10.2307/3609122.

[39] V. N. Drozdov, V. A. Kim, and L. B. Lazebnik, *[Modern approach to the prevention and treatment of NSAID-gastropathy].*, no. 2. 2011.

[40] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Algorithms in Modeling and Optimization*, no. May. 2013.

[41] S. Nesmachnow, "An overview of metaheuristics: accurate and efficient methods for optimisation," *Int. J. Metaheuristics*, vol. 3, no. 4, p. 320, 2014, doi: 10.1504/ijmheur.2014.068914.

[42] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, *Metaheuristic algorithms: A comprehensive review*. Elsevier Inc., 2018.

[43] H. R. Moshtaghi, A. T. Eshlaghy, and M. R. Motadel, "A Comprehensive Review on Meta-Heuristic Algorithms and their Classification with Novel Approach," vol. 8, pp. 63–89, 2021.

[44] K. O. C. Tse Guan Tan, Jason Teo, "Single- versus Multiobjective Optimization for Evolution of Neural Controllers in Ms. Pac-Man," *Int. J. Comput. Games Technol.*, vol. 2013, p. 7, 2013.

[45] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci. (Ny).*, vol. 237, no. February, pp. 82–117, 2013, doi: 10.1016/j.ins.2013.02.041.

[46] F. and M. L. (Glover, "Tabu Search," in *Tabu Search*, 1997, pp. 2–18.

[47] S. D. and J. P. SA. Ravichandran, JJ. Krist, D. Edwards, "Measuring sparingly-soluble, aqueous salt crystallization kinetics using CSTRs-in-series: Methodology development and CaCO3 studies," *Sep. Purif. Technol.*, vol. 211, pp. 408–420, 2019.

[48] J. L. and J. A. TM. Braden, MD. Johnson, ME. Kopach, J. McClary Groh, RD. Spencer, "Development of a commercial flow Barbier process for a pharmaceutical intermediate, Org. Process. 21(3) (2017) 317–326," *Res. Dev.*, vol. 21, no. 3, pp. 317–326, 2017.

[49] A. M. and S. A. D. Gola, P. Chawla, "Development and performance evaluation of native microbial consortium for multi metal removal in lab scale aerobic and anaerobic bioreactor," *Environ. Technol. Innov.*, 2020.

[50] J. A. and J. T. I. Malinen, J. Kangas, "A new homotopy-based strategy for the robust determination of all the feasible solutions for CSTR systems, 60(1) (2016) 8–23.," *Period.Polytech. Chem.*, vol. 60, no. 1, pp. 8–23, 2016.

[51] W. R. and A. P. A. Uppal, "On the dynamic behavior of continuous stirred tank reactors.," *Chem. Eng. Sci.*, vol. 29, no. 4, pp. 967–985, 1974.

[52] N. Muškinja and M. Rižnar, "Optimized PID position control of a nonlinear system based on correlating the velocity with position error," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/796057.

[53] J. Y. Márton, L., Hodel, A. S., Lantos, B., & Hung, "Underactuated robot control: comparing LQR, subspace stabilization, and combined error metric approaches.," *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3724–3730, 2008.

[54] E. M. Ali, E. T., Abdullah, S., Amir, M., & Adeel, "Stability Control of Ball and Beam System Using Heuristic Computation Based PI-D and PI-PD Controller.," *Tech. J.*, vol. 24, no. 1, pp. 21–29, 2019.

[55] I. Z. M. Saad, M. S., Jamaluddin, H., & Darus, "Implementation of PID controller tuning using differential evolution and genetic algorithms.," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 11, pp. 7761–7779, 2012.

[56] & X. Keshmiri, M., Jahromi, A. F., Mohebbi, A., Hadi Amoozgar, M. and W. F., "Modeling and control of ball and beam system using model based and non-model based control approaches.," *Int. J. Smart Sens. Intell. Syst.*, vol. 5, no. 1, 2013.

[57] A. Mahmoodabadi, M. J., Mottaghi, M. B. S., & Mahmodinejad, "Optimum design of fuzzy controllers for nonlinear systems using multi-objective particle swarm optimization.," *J. Vib. Control*, vol. 22, no. 3, pp. 769–783, 2016.

[58] H. Khodadadi, H., Ghadiri, "Self-tuning PID controller design using fuzzy logic for half car active suspension system.," *Int. J. Dynam. Control*, vol. 6, pp. 224–232, 2018, doi: https://doi.org/10.1007/s40435-016-0291-5.

[59] C.-E. J. Rivas-Echeverria F, Rios-Bolivar A, "Neural network-based auto-tuning for PID controllers.," *Neural Netw. World*, vol. 11, no. 3, pp. 277–284, 2001.

[60] Z. Z. Feng H, Yin CB, Weng WW, Ma W, Zhou JJ, Jia WH, "Robotic excavator trajectory control using an improved GA based PID controller. . 2018 May 15;105:153-68.," *Mech. Syst. Signal Process.*, vol. 15, no. 105,

pp. 153–168, 2018.

[61] S. M. Ribeiro JM, Santos MF, Carmo MJ, "Comparison of PID controller tuning methods: analytical/classical techniques versus optimization algorithms. In2017 18th international Carpathian control conference (ICCC) 2017 May 28 (pp. 533-538). IEEE.," in *international Carpathian control conference (ICCC)*, 2017, pp. 533–538.

[62] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsyst. Technol.*, vol. 24, no. 12, pp. 4997–5006, 2018, doi: 10.1007/s00542-018-3920-4.

[63] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control*, no. June, 2020, doi: 10.1007/s40435-020-00665-4.

[64] V. Zakian, *Control systems design. Springer-Verlag London Limited.* 2005.

[65] Z. Yan, J. Fan, and J. Wang, "A Collective Neurodynamic Approach to Constrained Global Optimization," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 5, pp. 1206–1215, 2017, doi: 10.1109/TNNLS.2016.2524619.

[66] R. K. Chandrawat, R. Kumar, B. P. Garg, G. Dhiman, and S. Kumar, "An analysis of modeling and optimization production cost through fuzzy linear programming problem with symmetric and right angle triangular fuzzy number," in *Advances in Intelligent Systems and Computing*, 2017, vol. 546, pp. 197–211, doi: 10.1007/978-981-10-3322-3_18.

[67] P. S. and G. Dhiman, "A fuzzy-LP approach in time series forecasting.," in *International Conference on Pattern Recognition and Machine Intelligence*, 2017, pp. 243–253.

[68] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowledge-Based Syst.*, vol. 165, pp. 169–196, 2019, doi: 10.1016/j.knosys.2018.11.024.

[69] G. D. and V. Kumar, "Spotted hyena optimizer for solving complex and non-linear constrained engineering problems.," *Harmon. Search Nat. Inspired Optim. Algorithms*, pp. 857–867, 2019.

[70] A. K. Bairwa, S. Joshi, and D. Singh, "Dingo Optimizer: A Nature-Inspired Metaheuristic Approach for Engineering Problems," *Math. Probl. Eng.*, vol. 2021, pp. 1–12, 2021, doi: 10.1155/2021/2571863.

[71] J.-L. Li, J.; Li, Y.-X.; Tian, S.-S.; Xia, "An improved cuckoo search algorithm with self-adaptive knowledge learning.," *Neural Comput. Appl*, 2019.

[72] G.-G. Rizk-Allah, R.M.; El-Sehiemy, R.A.;Wang, "A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution.," *Appl. Soft Comput.*, vol. 63, pp. 206–222, 2018.

[73] G.-G. Sun, J.; Miao, Z.; Gong, D.; Zeng, X.-J.; Li, J.; Wang, "Interval multi-objective optimization with memetic algorithms.," *IEEE Trans. Cybern.*, 2019.

[74] G.-G. Gu, Z.-M.; Wang, "Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization.," *Futur. Gener. Comput. Syst.*, vol. 107, pp. 49–69, 2020.

[75] M. Tunay, "A New Design of Metaheuristic Search Called Improved Monkey Algorithm Based on Random Perturbation for Optimization Problems," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/5557259.

[76] J. K. and R. Eberhart, "Particle swarm optimization," in *IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.

[77] K. Di Storn, R.; Price, "Diff erential evolution-a simple and effcient heuristic for global optimization over continuous spaces.," *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997.

[78] X. Wu, S. Wang, Y. Pan, and H. Shao, "A knee point-driven multi-objective artificial flora optimization algorithm," *Wirel. Networks*, vol. 8, 2020, doi: 10.1007/s11276-019-02228-8.

[79] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, 2012, doi: 10.1016/j.cnsns.2012.05.010.

[80] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the States of

Matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, 2014, doi: 10.1007/s10489-013-0458-0.

[81] A. Askarzadeh, "Bird mating optimizer: An optimization algorithm inspired by bird mating strategies," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 4, pp. 1213–1228, 2014, doi: 10.1016/j.cnsns.2013.08.027.

[82] Y. H. Chou, S. Y. Kuo, L. S. Yang, and C. Y. Yang, "Next generation metaheuristic: Jaguar algorithm," *IEEE Access*, vol. 6, pp. 9975–9990, 2018, doi: 10.1109/ACCESS.2018.2797059.

[83] M. C. Catalbas and A. Gulten, "Circular structures of puffer fish: A new metaheuristic optimization algorithm," *3rd Int. Conf. Electr. Biomed. Eng. Clean Energy Green Comput. EBECEGC 2018*, pp. 1–5, 2018, doi: 10.1109/EBECEGC.2018.8357123.

[84] W. Li and G. G. Wang, "Elephant herding optimization using dynamic topology and biogeography-based optimization based on learning for numerical optimization," *Eng. Comput.*, no. 0123456789, 2021, doi: 10.1007/s00366-021-01293-y.

[85] Z. Wang, GG., Deb, S. & Cui, "Monarch butterfly optimization," *Neural Comput Applic*, vol. 31, pp. 1995–2014, 2019.

[86] D. H. W. and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[87] M. M. R. A. Milani, T. Çavdar, and V. F. Aghjehkand, "Particle swarm optimization - Based determination of Ziegler-Nichols parameters for PID controller of brushless DC motors," in *INISTA 2012 - International Symposium on INnovations in Intelligent SysTems and Applications*, 2012, pp. 1–5, doi: 10.1109/INISTA.2012.6246984.

[88] S. Oladipo, Y. Sun, and Z. Wang, "Optimization of pid controller with metaheuristic algorithms for dc motor drives: Review," *Int. Rev. Electr. Eng.*, vol. 15, no. 5, pp. 352–381, 2020, doi: 10.15866/iree.v15i5.18688.

[89] N. Khanduja and B. Bhushan, "Optimal design of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 46, no. 2, 2021, doi: 10.1007/s12046-021-01632-1.

[90] T. L. Pecora, L.M. and Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, p. 821, 1990.

[91] X. G. W. Xu, Z. Geng, Q. Zhu, "A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization," *Inform. Sci.*, vol. 218, pp. 85–102, 2013.

[92] X. S. and T. Gandomi, A.H., Yun, G.J., Yang, "Chaos-enhanced accelerated particle swarm optimization", Commun. 18(2) (2013) 327–340.," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 2, pp. 327–340, 2013.

[93] A. H. Gandomi, A.H., Yang, X.S., Talatahari, S. and Alavi, "Firefly algorithm with chaos.," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, 2013.

[94] X. Han, X. and Chang, "A chaotic digital secure communication based on a modified gravitational search algorithm filter.," *Inf. Sci. (Ny).*, vol. 208, pp. 14–27, 2012.

[95] A. Saremi, S., Mirjalili, S. and Lewis, "Biogeography-based optimisation with chaos," *Neural Comput. Appl.*, vol. 25, no. 5, pp. 1077–1097, 2014.

[96] H. Wang, G.G., Guo, L., Gandomi, A.H., Hao, G.S. and Wang, "Chaotic krill herd algorithm.," *Inf. Sci. (Ny).*, vol. 274, pp. 17–34, 2014.

[97] A. H. Wang, G.G., Deb, S., Gandomi, A.H., Zhang, Z. and Alavi, "Chaotic cuckoo search," *Soft Comput.*, vol. 20, no. 9, pp. 3349–3362, 2016.

[98] A. H. Wang, G.G., Deb, S., Gandomi, A.H., Zhang, Z. and Alavi, "A novel cuckoo search with chaos theory and

elitism scheme.," in *International Conference on Soft Computing and Machine Intelligence*, 2014, pp. 64–69.

[99] A. H. Wang, G.G., Gandomi, A.H. and Alavi, "A chaotic particle-swarm krill herd algorithm for global numerical optimization," *Kybernetes*, 2013.

[100] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence.," in *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and inernet commerce*, 2005, pp. 695–701.

[101] H. Y. ZHOU, X.Y., WU, Z.J., Wang, H., LI, K.S. and ZHANG, "Elite opposition-based particle swarm optimization.," *ACTA ELECTONICA Sin.*, vol. 41, no. 8, p. 1647, 2013.

[102] S. Wang, P.C., Gao, W.C., Qian, X., Gou, H. and Wang, "Hybrid fireworks explosion optimization algorithm using elite opposition-based learning," *J. Comput. Appl.*, vol. 633, no. 2, pp. 283–290, 2014.

[103] V. Dhiman, G. and Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems.," *Knowledge-Based Syst.*, vol. 159, pp. 20–50, 2018.

[104] D. H. W. and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[105] V. Dhiman, G. and Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems.," *Knowledge-Based Syst.*, vol. 165, pp. 169–196, 2019.

[106] M. Dhiman, G., Garg, M., Nagar, A., Kumar, V. and Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–26, 2020.

[107] A. Dhiman, G. and Kaur, "A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization.," *Soft Comput. Probl. solving*, pp. 599–615, 2019.

[108] G. Dhiman, "ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems.," *Eng. Comput.*, pp. 1–31, 2019.

[109] A. H. Wang, G.G., Deb, S., Gandomi, A.H. and Alavi, "Opposition-based krill herd algorithm with Cauchy mutation and position clamping.," *Neurocomputing*, vol. 177, pp. 147–157, 2016.

[110] S. Li, G., Wang, G.G. and Wang, "Two-Population Coevolutionary Algorithm with Dynamic Learning Strategy for Many-Objective Optimization," *Mathematics*, vol. 9, no. 4, p. 420, 2021.

[111] G. Singh, P. and Dhiman, "A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches.," *J. Comput. Sci.*, vol. 27, pp. 370–385, 2018.

[112] S. F. and N. D. Rui Tang, "Metaheuristics and chaos theory.," in *Chaos theory*, 2018, pp. 182–196.

[113] S. Rahnamayan, *Opposition-based differential evolution*. 2007.

[114] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning : A literature review," no. November 2016, 2017, doi: 10.1016/j.swevo.2017.09.010.

[115] H. R. T. and M. M. A. S. S. Rahnamayan, "Opposition-Based Differential Evolution Algorithms," in *IEEE International Conference on Evolutionary Computation*, 2006, pp. 2010–2017.

[116] M. J. and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, pp. 150–194, 2013.

[117] K. Hussain, M. N. M. Salleh, S. Cheng, and R. Naseem, "Common benchmark functions for metaheuristic evaluation: A review," *Int. J. Informatics Vis.*, vol. 1, no. 4–2, pp. 218–223, 2017, doi: 10.30630/joiv.1.4-2.65.

[118] M. W. Wolpert DH, "No free lunch theorems for optimization.," *IEEE Trans Evol Comput*, vol. 1, pp. 67–82, 1997.

[119] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[120] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,” *Eng. Appl. Artif. Intell.*, vol. 90, no. December 2018, p. 103541, 2020, doi: 10.1016/j.engappai.2020.103541.

[121] P. Gupta, V. Kumar, K. P. S. Rana, and P. Mishra, “Comparative study of some optimization techniques applied to Jacketed CSTR control,” *2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015*, 2015, doi: 10.1109/ICRITO.2015.7359368.

[122] M. S. Nasrabadi, Y. Sharafi, and M. Tayari, “A Parallel Grey Wolf Optimizer combined with Opposition based learning,” in *Ist Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, 2016, pp. 18–23.

[123] Nagarajan M and Asokan A, “Design and implementation of fractional order controller for CSTR process.,” *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2014.

[124] P. S. K. and C. C. S. Yadav P, Kumar R, “An intelligent tuned harmony search algorithm for optimization.,” *Inf. Sci.*, vol. 196, pp. 47–72, 2012.

[125] S. H. and G. F. S, “Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems.,” *Appl. Soft Comput.*, vol. 71, pp. 728–746, 2018.

[126] C. M. C. and G. A, “Circular structures of pufferfish: a new metaheuristic optimization algorithm.,” in *Third International Conference on Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC) IEEE*, 2018, pp. 1–5.

[127] A. Banks, J. Vincent, and C. Anyakoha, “A review of particle swarm optimization. Part I: Background and development,” *Nat. Comput.*, vol. 6, no. 4, pp. 467–484, 2007, doi: 10.1007/s11047-007-9049-5.

[128] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, pp. 1942–1948, doi: 10.1109/IEMBS.2011.6091617.

[129] R. Goyal, “A new approach for simplification and control of linear time invariant systems,” *Microsyst. Technol.*, vol. 1, 2018, doi: 10.1007/s00542-018-4004-1.

[130] A. Bhatt, R., Parmar, G., Gupta, R. and Sikander, “Application of stochastic fractal search in approximation and control of LTI systems,” *Microsyst. Technol.*, vol. 25, no. 1, pp. 105–114, 2018, doi: 10.1007/s00542-018-3939-6.

[131] K. Naidu, H. Mokhlis, and A. H. A. Bakar, “Application of firefly algorithm (FA) based optimization in load frequency control for interconnected reheat thermal power system,” *2013 IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. AEECT 2013*, 2013, doi: 10.1109/AEECT.2013.6716461.

[132] A. M. Ahmed, T. A. Rashid, and S. A. M. Saeed, “Cat Swarm Optimization Algorithm : A Survey and Performance Evaluation,” *Comput. Intell. Neurosci.*, vol. 2020, 2020.

[133] R. Venkata Rao, “Review of applications of tlbo algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems,” *Decis. Sci. Lett.*, vol. 5, no. 1, pp. 1–30, 2016, doi: 10.5267/j.dsl.2015.9.003.

[134] B. K. Sahu, S. Pati, P. K. Mohanty, and S. Panda, “Teaching-learning based optimization algorithm based fuzzy-PID controller for automatic generation control of multi-area power system,” *Appl. Soft Comput. J.*, vol. 27, pp. 240–249, 2015, doi: 10.1016/j.asoc.2014.11.027.

[135] G. Wang, L. Guo, A. H. Gandomi, G. Hao, and H. Wang, “Chaotic Krill Herd algorithm,” *Inf. Sci. (Ny).*, 2014, doi: 10.1016/j.ins.2014.02.123.

[136] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, “Red deer algorithm (RDA): a new nature-inspired meta-heuristic,” *Soft Comput.*, vol. 0123456789, 2020, doi: 10.1007/s00500-020-04812-z.

[137] S. Shadravan, H. R. Naji, and V. K. Bardsiri, “The Sailfish Optimizer: A novel nature-inspired metaheuristic

algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, no. July 2018, pp. 20–34, 2019, doi: 10.1016/j.engappai.2019.01.001.

[138] G. P. Liu, S. Daley, and G. R. Duan, "Application of optimal-tuning pid control to industrial hydraulic systems," *IFAC Proc. Vol.*, vol. 35, no. 1, pp. 179–184, 2002, doi: 10.3182/20020721-6-es-1901.01181.

[139] E. Yesil, M. Guzelkaya, and I. Eksin, "Fuzzy PID controllers : An overview," *ResearchGate*, no. October, pp. 1–8, 2014.

[140] M. K. O. Ayomoh and M. T. Ajala, "Neural network modeling of a tuned PID controller," *Eur. J. Sci. Res.*, vol. 71, no. 2, pp. 283–297, 2012.

[141] N. Khanduja and S. Sharma, "Performance Analysis of CSTR using Adaptive Control," *Int. J. Soft Comput. Eng. ISSN 2231-2307, Vol. Issue-2, May 2014*, no. 2, pp. 80–84, 2014.

[142] N. Khanduja and B. Bhushan, "Control System Design and Performance Analysis of PID and IMC controllers for Continuous Stirred Tank Reactor (CSTR)," vol. 10, no. 1, pp. 16–22, 2019.

[143] G. Glan Devadhas and S. Pushpakumar, "An intelligent design of PID controller for a continuous stirred tank reactor," *World Applied Sciences Journal*, vol. 14, no. 5. pp. 698–703, 2011.

[144] K. Huang, Y. Zhou, X. Wu, and Q. Luo, "A cuckoo search algorithm with elite opposition-based strategy," *J. Intell. Syst.*, vol. 2015, no. November, 2015, doi: 10.1515/jisys-2015-0041.

[145] H. Goud and P. Swarnkar, "Investigations on Metaheuristic Algorithm for Designing Adaptive PID Controller for Continuous Stirred Tank Reactor," *Mapan - J. Metrol. Soc. India*, vol. 34, no. 1, pp. 113–119, 2019, doi: 10.1007/s12647-018-00300-w.

[146] N. Khanduja and B. Bhushan, "Intelligent Control of CSTR using IMC-PID and PSO-PID controller," *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst. ICPEICES 2016*, 2017, doi: 10.1109/ICPEICES.2016.7853329.

[147] N. Khanduja, "CSTR Control by Using Model Reference Adaptive Control and PSO," *World Acad. Sci. Eng. Technol. Int. J. Mech. Aerospace, Ind. Mechatron. Manuf. Eng. Vol8, No12, 2014*, vol. 8, no. 12, pp. 2144–2149, 2014.

[148] I. Podlubny, "Fractional-order systems and PIlDm-controllers.," *IEEE Trans. Autom. Control*, vol. 44, pp. 208–214, 1999.

[149] I. Podlubny, "Fractional differential équations," in *San Diego: Academic Press*, 1999.

[150] I. Tejado, B. M. Vinagre, J. E. Traver, J. Prieto-Arranz, and C. Nuevo-Gallardo, "Back to basics: Meaning of the parameters of fractional order PID controllers," *Mathematics*, vol. 7, no. 6, pp. 1–16, 2019, doi: 10.3390/MATH7060530.

[151] I. Podlubny, "Fractional-Order Systems andPI^γ D^μ-Controllers," *IEEE Trans. Automat. Contr.*, vol. 44, no. 1, pp. 208–214, 1999.

[152] M. Nagarajan and A. Asokan, "Design and Implementation of Fractional Order Controller for CSTR Process," *Int. J. Comput. Appl.*, no. 2, pp. 975–8887, 2014.

[153] C. Zhang, T. Peng, C. Li, W. Fu, X. Xia, and X. Xue, "Multiobjective optimization of a fractional-order PID controller for pumped turbine governing system using an improved NSGA-III algorithm under multiworking conditions," *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/5826873.

[154] J. Poovarasan, R. Kayalvizhi, and R. K. Pongiannan, "Design of Fractional Order PID controller for a CSTR process," vol. 3, no. 1, pp. 8–14, 2014.

[155] A. Agarwal, J., Parmar, G., Gupta, R. and Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsyst. Technol.*, vol. 24, no. 12, pp. 4997–5006, 2018, doi:

10.1007/s00542-018-3920-4.

[156] B. Hekimoglu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, no. c, pp. 38100–38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[157] S. Sengupta, S. Basak, and R. Peters, "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 157–191, 2018, doi: 10.3390/make1010010.

[158] L. Sun, X. Song, and T. Chen, "An improved convergence particle swarm optimization algorithm with random sampling of control parameters," *J. Control Sci. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/7478498.

[159] N. Khanduja and B. Bhushan, "CSTR control using IMC-PID, PSO-PID, and hybrid BBO-FF-PID controller," in *Advances in Intelligent Systems and Computing*, vol. 697, Springer Singapore, 2019, pp. 519–526.

[160] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, pp. 330–343, 2010, doi: 10.1504/IJMMNO.2010.035430.

[161] S. R. Kumar and S. Ganapathy, "Cuckoo Search Optimization Algorithm based Load Frequency Control of Interconnected Power Systems with GDB nonlinearity and SMES units," *Int. J. Electr. Power Energy Syst.*, vol. 2, no. 12, pp. 23–28, 2013.

[162] R. Rajabioun, "Cuckoo optimization algorithm," *Appl. Soft Comput. J.*, vol. 11, no. 8, pp. 5508–5518, 2011, doi: 10.1016/j.asoc.2011.05.008.

[163] R. Rohit Gupta, "PSO Based Optimal Design of Fractional Order Controller for Industrial Application," *Int. J. Electr. Comput. Eng. Vol*, vol. 9, no. 6, pp. 665–672, 2015.

[164] A. Singh, "Concentration Control of CSTR Through Fractional Order PID Controller by Using Soft Techniques," 2013.

[165] B. B. Rupam Singh, "Improved ant colony optimization for achieving self-balancing and position control for balancer systems," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–18, 2020.

[166] Bansode Lawrence, "Tuning of A PID Controller for Optimal Performance of Ball and Beam System," *Int. J. Eng. Res.*, vol. V9, no. 04, pp. 1–5, 2020, doi: 10.17577/ijertv9is040009.

[167] D.-C. S. et al Aranda J, Chaos D, "benchmark control problems for a non-lin ear underactuated hov erc raft: a simulation laboratory for control testing," *IFAC Proc.*, vol. 7, pp. 463–468, 2006.

[168] K. Nowopolski, "Ball-and-beam laboratory system controlled by SimuLink model through dedicated microcontrolled-Matlab data exchange protocol," *Comput Appl Electr Eng*, vol. 11, pp. 310–320, 2013.

[169] W. P. S.-K. Oh, H.-J. Jang, "Optimized fuzzy PD cas_cade controller: a comparative analysis and design," *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 181–195, 2011.

[170] M. H. A. M. Keshmiri, A. F. Jahromi, A. Mohebbi and and W.-F. Xie, "Modeling and control of ball and beam system using model based and non-model based control approaches," *Int. J. Smart Sens. Intell. Syst.*, vol. 5, no. 1, pp. 14–35, 2012.

[171] D. D. N. Muˇskinja, B. Tovornik, "discrete supervisory controller for real-time fuzzy control sys_tems," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 161–166, 1997.

[172] H. Gözde, "Comparative Analysis of Evolutionary Computation Based Gain Scheduling Control for Ball and Plate Stabilization System," *Balk. J. Electr. Comput. Eng.*, vol. 7, no. 1, pp. 44–55, 2019, doi: 10.17694/bajece.466306.

[173] J. M. Howe III and R. T. O'Brien Jr., "Experimental investiga_tion of optimal PID controller design using model reduction techniques," in *roceedings of the 40th Southeastern Symposium on System Theory (SSST '08), pp. 31–*

36, IEEE, New Orleans, La USA, 2008, pp. 31–36.

[174] K. (2014). 10. 1007/97.-3-319-02636-7. Choukchou-Braham, Amal & Brahim, Cherki & Djemai, Mohamed & Busawon, *Analysis and Control of Underactuated Mechanical Systems.* 2014.

[175] M. Rudra, Shubhobrata & Barai, Ranjit Kumar & Maitra, *Block Backstepping Control of the Underactuated Mechanical Systems*. 2017.

[176] R. Kennedy, J. and Eberhart, “Particle swarm optimization,” in *In Proceedings of ICNN’95-international conference on neural networks*, 19955, pp. 1942–1948.

[177] H. W. Yan X, Wu Q, Liu H, “An Improved Particle Swarm Optimization Algorithm and Its Application,” *Int. J. Comput. Sci.*, pp. 316–324, 2013.

[178] L. M. I. 2009: 2127–2134. Gong D, Lu L, “Robot Path Planning In Uncertain Environments Based On Particle Swarm Optimization.,” in *IEEE Congress on Evolutionary Computation*, 2009, pp. 2127–2134.

[179] D. W. · D. T. · L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft Comput.*, vol. 22, no. 2, pp. 387–408, 2017.

[180] S. S. and B. R. B. Nagaraj, “Tuning Algorithms for PID Controller Using Soft Computing Techniques,” *Int. J. Comput. Sci. Netw.*, vol. 8, no. 4, 2008.

[181] G. Lahoty, P. and Parmar, “A comparative study of tuning of PID controller using evolutionary algorithms,” *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 1, pp. 640–644, 2013.

[182] H. Salimi, “Stochastic fractal search: a powerful metaheuristic algorithm.,” *Knowledge-Based Syst.*, vol. 75, pp. 1–18, 2015.

[183] R. Bhatt, G. Parmar, R. Gupta, and A. Sikander, “Application of stochastic fractal search in approximation and control of LTI systems,” *Microsyst. Technol.*, vol. 25, no. 1, pp. 105–114, 2019, doi: 10.1007/s00542-018-3939-6.

[184] and M. K. R. Namba, T. Yamamoto, “Robust PID controller and its application,” in *IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 3636–3641.

[185] M. M. Sabir and J. A. Khan, “Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques,” *Adv. Artif. Neural Syst.*, vol. 2014, pp. 1–8, 2014, doi: 10.1155/2014/126317.

[186] J.G. Ziegler and N.B. Nichols., “Opt imum Setting for Aut omatic Controllers,” *Trans. ASME*, vol. 64, pp. 759–768, 1942.

[187] T. H. K. J. Astrom, “PID Controllers:Theory, Design, and Tuning, NC, 1995.,” *Instrum. Soc. Am. Res. Triangle Park.*, 1995.

[188] P. V. G. K. Rao, M. V. Subramanyam, and K. Satyaprasad, “Study on PID controller design and performance based on tuning techniques,” *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, pp. 1411–1417, 2014, doi: 10.1109/ICCICCT.2014.6993183.

[189] M. Araki, “Two-degree-of-freedom control system: part I,” *Syst. Control*, vol. 29, pp. 649–656, 1985.

[190] G. L. Deshmukh and C. B. Kadu, “Design of two degree of freedom PID controller for temperature control system,” *Int. Conf. Autom. Control Dyn. Optim. Tech. ICACDOT 2016*, pp. 586–589, 2017, doi: 10.1109/ICACDOT.2016.7877653.

[191] M. Kumar and P. V. V Patel, “Two Degree of Freedom PID Controller For speed control of DC Motor,” no. Mv, pp. 94–98, 2015.

[192] J. G. Z. and N. B. Nichols, “Optimum settings for automatic controllers,” *Trans. ASME*, vol. 64, no. 11, 1942.

[193] J. barman and J. D. Radh Kushwaha, Tapas Kumar Saha, “Analysis and design of 2DOF control of Push Pull converter used for Standalone PV system,” 2015.

[194] N. H. K. Sundaravadivu, S. Sivakumar, “2DOF PID Controller Design for a Class of FOPTD Models –An Analysis with Heuristic Algorithms,” *ICCC-2015, Elsevier, Procedia Comput. Sci.*, vol. 48, no. 2015, pp. 90–95, 2015.

[195] J. L. Xianzheng LI, Xingcheng WA, Kai Z, “Design of variable pitch-control system based on two-degree-of-freedom internal model control.,” in *International Conference on Information Science, Electronics and Electrical Engineering*, 2014, pp. 1348–1352.

[196] S. P. Panjapornpon C, “Control of Uncertain Input-delay Systems by using Input/output Linearization with A Two-degree-of-freedom Scheme.,” in *1th IEEE International Conference on Control & Automation (ICCA)*, 2014, pp. 473–478.

[197] H. A. Suthar, “Two-Degree-of-Freedom PID Controller, Its Equivalent Forms and Special cases,” *IAES Int. J. Robot. Autom.*, vol. 4, no. 4, p. 269, 2015, doi: 10.11591/ijra.v4i4.pp269-283.

[198] M. Araki and H. Taguchi, “Two-degree-of-freedom PID controllers,” *Int. J. Control. Autom. Syst.*, vol. 1, no. 4, pp. 401–411, 2003, doi: 10.11509/isciesci.42.1_18.

[199] P. W. J. A. Miller, A. M. Lopez, C. L. Smith and Murrill, “A comparison of controller tuning techniques,” *Control Eng.*, vol. 14, pp. 72–75, 1967.

[200] Y. Zhou, J. Zhang, X. Yang, and Y. Ling, “Optimization of PID controller based on water wave optimization for an automatic voltage regulator system,” *Inf. Technol. Control*, vol. 48, no. 1, pp. 160–171, 2019, doi: 10.5755/j01.itc.48.1.20296.

List of Publications

The work conducted in this research has resulted in the following major publications:

Journal Publications

1. Khanduja, N., & Bhushan, B. (2022). Hybrid State of Matter Search Algorithm and its Application to PID Controller Design for Position Control of Ball Balancer System. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 1-20. (doi: 10.1007/s40998-022-00506-4). (Springer, I.F.-1.376). SCIE INDEXED.
2. Khanduja, N., & Bhushan, B. (2021). Chaotic state of matter search with elite opposition-based learning: A new hybrid metaheuristic algorithm. Optimal Control Applications and Methods, (doi: 10.1002/oca.2810 (2021). (Wiley, I.F.-2.53). SCIE INDEXED.
3. Khanduja, N., & Bhushan, B. (2021). Optimal design of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm. Sādhanā, 46(2), 1-12, (doi: https://doi.org/10.1007/s12046-021-01632-1). (Springer, I.F.-1.2). SCIE INDEXED.
4. Khanduja, N., Bhushan, B., & Mishra, S. (2020). Control of CSTR using firefly and hybrid firefly-biogeography based optimization (BBFFO) algorithm. Journal of Information and Optimization Sciences, 41(6),1443- 1452, (doi: https://doi.org/10.1080/02522667.2020.1809098). (Taylor & Francis). ESCI INDEXED.
5. Khanduja, N., Bhushan, B. (2019). Control System Design and Performance Analysis of PID and IMC controllers for Continuous Stirred Tank Reactor (CSTR). *Journal of Control & Instrumentation*,10(1),16–22.

Book Chapters

1. Khanduja, N., & Bhushan, B. (2021). Recent advances and application of metaheuristic algorithms: A survey (2014–2020). Metaheuristic and Evolutionary Computation: Algorithms and Applications (pp 207-228). Springer, Singapore.
2. Khanduja, N., & Bhushan, B. (2019). CSTR Control Using IMC-PID, PSO-PID, and Hybrid BBO-FF-PID Controller. In Applications of artificial Intelligence techniques in Engineering (pp. 519-526). Springer, Singapore.
3. Khanduja, N., & Bhushan, B. (2021). Comparative Control Study of CSTR Using Different Methodologies: MRAC, IMC-PID, PSO-PID, and Hybrid BBO-FF-PID. In AI and Machine Learning Paradigms for Health Monitoring System (pp. 407-417). Springer, Singapore.

Conference Proceedings

1. Khanduja, N., & Bhushan, B. (2016, July). Intelligent control of CSTR using IMC-PID and PSO-PID controller. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1-6). IEEE, (doi: 10.1109/ICPEICES.2016.7853329).
2. Khanduja, N., Jha, N., & Bhushan, B. (2016, July). Swarm and pheromone-based reinforcement learning methods for the robot (s) path search problem. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1-7). IEEE, (doi: 10.1109/ICPEICES.2016.7853733).

NEHA KHANDUJA
Ph.D. Scholar
Delhi Technological University

Career Objective:

To pursue a challenging career and be part of a progressive organization that gives scope to enhance my knowledge, skills and to reach the pinnacle in the computing and research field with sheer determination, dedication and hard work.

Professional Experience: (Approx. 16 Years)

Duration	Company/Organization	Designation
01/08/2009-presently working	Bhagwan Parshuram Institute of Technology (G.G.S.I.P.U), Rohini, Delhi	Assistant professor (EEE)
11/08/08-31/07/2009	Gurgaon Institute of Technology and Management, Gurgaon, Haryana	Lecturer (ECE)
01/07/06-31/03/08	Poornima Group of Colleges	Lecturer (EE)

- **Got best teacher award at Bhagwan Parshuram Institute of Technology (G.G.S.I.P.U), Rohini, Delhi during the session 2014-15.**
- **Got best teacher award at Gurgaon Institute of Technology and Management, Gurugram during the session 2008-09.**
- **Got best teacher award at Poornima College of Engineering, Jaipur during the session 2006-07.**

Subjects Taught:

- Circuits & Systems
- Electrical Science
- Power Station Practice
- Non Conventional Energy Sources
- Power System
- Transmission and Distribution
- Power System Analysis
- Power System Engineering
- Control System Engineering
- Electrical Machines
- Electromechanical Energy Conversion
- Power Quality Management

- Neuro Fuzzy Systems
- Advanced Control Systems

Ph.D Thesis Topic :

Metaheuristic algorithms and its applications to nonlinear control system

Major Projects:

M.Tech Project: "CSTR control using Model Reference Adaptive Control and Bio-Inspired Optimization Technique"

B.Tech Project: "Automatic phase changer & Servo stabilizer"

Professional Skills: -

Programming Languages - Matlab,C, C++

Operating Systems - Windows

Others- Power Point, MS-Word, Excel

Professional Qualification:

M.Tech in Control & Instrumentation from Delhi Technological University, Delhi with CGPA **8.67** in 2013

B.E. in Electrical Engineering from Kautilya Institute of Technology & Engineering, Jaipur Rajasthan University, with **80.05%** in 2006.

***Academic Qualification*:**

Sr. Sec. Education from Vedic girls school, Jaipur (RBSE), with **78.15%** in 2002.

Sec. Education from New modern sec. school, Jaipur (RBSE), with **86.33%** in 2000.

Papers Published:

Journals:

1. Khanduja, N., & Bhushan, B. (2022). Hybrid State of Matter Search Algorithm and its Application to PID Controller Design for Position Control of Ball Balancer System. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 1-20. (doi: 10.1007/s40998-022-00506-4). (Springer, I.F.-1.376). SCIE INDEXED.
2. Khanduja, N., & Bhushan, B. (2021). Chaotic state of matter search with elite opposition-based learning: A new hybrid metaheuristic algorithm. Optimal Control Applications and Methods, (doi: 10.1002/oca.2810 (2021). (Wiley, I.F.-2.53). SCIE INDEXED.
3. Khanduja, N., & Bhushan, B. (2021). Optimal design of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm. Sādhanā, 46(2), 1-12, (doi: https://doi.org/10.1007/s12046-021-01632-1). (Springer, I.F.-1.2). SCIE INDEXED.
4. Khanduja, N., Bhushan, B., & Mishra, S. (2020). Control of CSTR using firefly and hybrid firefly-biogeography based optimization (BBFFO) algorithm. Journal of Information and Optimization Sciences, 41(6),1443- 1452, (doi: https://doi.org/10.1080/02522667.2020.1809098). (Taylor & Francis). ESCI INDEXED.

5. Khanduja,N., Garg,V.,Vats, V., Baliyan, A. & Sharma, A. (2020). Model Based Control of CSTR: A Comparative Study. IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P-ISSN 2349-5138,.7(2): 995-999.
6. Khanduja, N., Bhushan, B. (2019). Control System Design and Performance Analysis of PID and IMC controllers for Continuous Stirred Tank Reactor (CSTR). *Journal of Control & Instrumentation*,10(1),16–22.
7. Khanduja, N. (2015). CSTR control by using model reference adaptive control and PSO. International Journal of Mechanical and Mechatronics Engineering, 8(12), 2144-2149.
8. Khanduja, N., & Sharma, S. (2014). Performance analysis of CSTR using adaptive control. International Journal of Soft Computing and Engineering (IJSCE), 4(2), 80-84.

International Conferences:

1. Khanduja, N., & Bhushan, B. (2016, July). Intelligent control of CSTR using IMC-PID and PSO-PID controller. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1-6). IEEE, (doi: 10.1109/ICPEICES.2016.7853329).
2. Khanduja, N., Jha, N., & Bhushan, B. (2016, July). Swarm and pheromone-based reinforcement learning methods for the robot (s) path search problem. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1-7). IEEE, (doi: 10.1109/ICPEICES.2016.7853733).
3. Khanduja, N., Kumar, N. (2014). Control of CSTR by using Lyapunov's Rule of MRAC and PSO. In India 3rd International Conference on Advance trends in Engineering, Technology and Research (ICATETR-2014), organized by BKIT Kota.
4. Kumar, N., & Khanduja, N. (2012, December). Mathematical modelling and simulation of CSTR using MIT rule. In 2012 IEEE 5th India International Conference on Power Electronics (IICPE) (pp. 1-5). IEEE, (doi: 10.1109/ICPEICES.2016.7853329).

Book Chapters:

1. Khanduja, N., & Bhushan, B. (2021). Recent advances and application of metaheuristic algorithms: A survey (2014–2020). Metaheuristic and Evolutionary Computation: Algorithms and Applications (pp 207-228). Springer, Singapore.
2. Khanduja, N., & Bhushan, B. (2019). CSTR Control Using IMC-PID, PSO-PID, and Hybrid BBO-FF-PID Controller. In Applications of artificial Intelligence techniques in Engineering (pp. 519-526). Springer, Singapore.
3. Khanduja, N., & Bhushan, B. (2021). Comparative Control Study of CSTR Using Different Methodologies: MRAC, IMC-PID, PSO-PID, and Hybrid BBO-FF-PID. In AI and Machine Learning Paradigms for Health Monitoring System (pp. 407-417). Springer, Singapore.

National Conferences:

1. Neha Khanduja, Simmi Sharma, "Study of Structure of Solar Energy for Tomorrow" In National Electrical Energy Conference (NEEC-2011), Organized by Delhi Technological University.

2. Neha Khanduja, Simmi Sharma, "Optimal Control of CSTR" In National Conference on Recent Trends in Electrical, Electronics and Communication Engineering (RTEECE-2019), Organized by BPIT, Delhi.

Seminar/Workshops Attended:

1. AICTE sponsored National Level one-week Teachers Training Programme entitled "NBA Accreditation and Processes" organised by GVM Institute of Technology & Management inMay,2021.
2. One-week online faculty development program entitled "Exploring Trends, Challenges & Perspectives in Engineering Education" organized by Guru Tegh Bahadur Institute of Technology, in April ,2021.
3. One day Faculty Development Programme entitled" Roles of teachers in technology driven higher education" organized by Guru Angad Dev Teaching Learning Centre SGTB Khalsa College, University of Delhi, in April 2020.
4. One day Faculty Development Programme entitled "Challenges and Opportunities before Indian Higher Education due to COVID-19" organized by Guru Angad Dev Teaching Learning Centre SGTB Khalsa College, University of Delhi, in May 2020.
5. One day virtual workshop on" Creation and Use of HTML based Quiz for E-content and Open Educational Resource (OER) development" organized by BSVS Educational Learning Centre, Nagpur held on 18/05/2020.
6. Five Days webinar series on "Beyond the Boundaries: Reinventing the Horizons" Organized by Swami Keshvanand Institute of technology, management and Gramothan, Jaipur from 16-20 May,2020.
7. One week Faculty Development Programme on "Recent Research Directions & Key Concepts in Electrical Engineering" organized by Delhi Technological University, Delhi in December 2019.
8. One day workshop on "Digital Literacy & Online Safety" organized by Delhi Technological University, Delhi in April 2019.
9. One day joint workshop on "Patent Filing Procedure "organized by Rajiv Gandhi National Institute of Intellectual Property Management (RGNIIPM) Nagpur and Delhi Technological University, Delhi in May,2018.
10. One day workshop on "Writing & Publishing Research Paper" organized by Bhagwan Parshuram Institute of Technology, Delhi in January,2018.
11. Two-week ISTE STTP on "Electric Power System "organized by Bhagwan Parshuram Institute of Technology, Delhi in July 2017.
12. One week Faculty Development Programme on "Recent Advances and Challenges in power & Energy for Sustainable Growth" organized by Delhi Technological University, Delhi in June,2015.
13. One day Workshop on "LaTeX" organized by Bhagwan Parshuram Institute of Technology, Delhi in January ,2015.
14. One week Faculty Development Programme on Recent Trends in Switchgear & Protection" organized by Delhi Technological University, Delhi in July,2014.
15. One day Workshop on "Recent Trends and Advances in Power Electronics and Drives "organized by Guru Teg Bahadur Institute of Technology, Delhi in march,2012.

16. Ten days Staff Development Programme on "Soft Computing using Matlab and Latex at a Glance" organized by Delhi Technological University, Delhi in December,2009.
17. Two-day Workshop on "CMOS based Analog Signal Processing & VLSI Design "organized by Bhagwan Parshuram Institute of Technology, Delhi in February ,2014.

***Personal Details*:**

Father	:	Sh. D.R.Malhotra (A.O. in Rajasthan Housing Board)
Husband	:	Vijay Khanduja (Cost Accountant working in an MNC)
Date of Birth	:	Oct.27, 1984
Sex	:	Female
Marital Status	:	Married
Nationality	:	Indian
Language Known	:	English & Hindi
Personal Strength	:	Leadership skills, Self-confidence, Optimism and Hard working.

Present Address

B-43, Sarve Sanjhi Apartment
Plot No.-8, Sector-9, Dwarka,
New Delhi-110077
E-Mail: nehakhanduja.@bpitindia.com
nehakhanduja.dce@gmail.com

***Declaration*:** I hereby declare that all the details furnished above are true to the best of my knowledge and belief.

(NEHA KHANDUJA)