DESIGN AND ANALYSIS OF INTELLIGENT CONTROLLERS FOR NON-LINEAR SYSTEMS

A Thesis Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

DOCTOR OF PHILOSOPHY

Submitted by

SNIGDHA CHATURVEDI

(Enrollment No. 2k17/PhD/EE/07)

Under the Supervision of

Prof. Narendra Kumar II
Delhi Technological University

Dr. Rajesh Kumar
NIT Kurukshetra

To the

Faculty of Electrical Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

January, 2024

CANDIDATE'S DECLARATION

I hereby affirm that the research work presented in the thesis titled "**Design and Analysis of Intelligent Controllers for Non-Linear Systems**," submitted in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in the Department of Electrical Engineering at Delhi Technological University, Delhi, is entirely original and conducted under the guidance of **Prof. Narendra Kumar II** and **Dr. Rajesh Kumar**. This thesis has not been previously submitted for any other academic degree.

Snigdha Chaturvedi
(2k17/PhD/EE/07)

Place: Delhi

Date:__/__/________

CERTIFICATE

I hereby certify, based on the candidate's declaration, that the work presented in this thesis titled **"Design and Analysis of Intelligent Controllers for Non-linear Systems",** submitted to the Department of Electrical Engineering, Delhi Technological University, Delhi, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy**, constitutes an original contribution to existing knowledge. It is a faithful record of the research work conducted by the candidate under my guidance and supervision.

To the best of my knowledge, this work has not been submitted in part or in full for the award of any degree elsewhere.

Prof. Narendra Kumar
Professor
Department of Electrical Engineering
DTU, Delhi-110042, India

Dr. Rajesh Kumar
Assistant Professor [Grade-I],
Department of Electrical Engineering,
National Institute of Technology Kurukshetra,
Kurukshetra-136119, India

The PhD viva voce of **Ms. Snigdha Chaturvedi**, research scholar has been held on __________

Signature of

Supervisors

Signature of

Head,

Dept. of Electrical Engineering

ACKNOWLEDGEMENTS

It has been both fascinating and extremely rewarding to write this thesis. I am grateful to God for giving me an opportunity to learn and acquire knowledge. First, I would like to express my sincere gratitude to my esteemed supervisors, **Prof. Narendra Kumar II** and **Dr. Rajesh Kumar** for their constant guidance and encouragement throughout the research work. I am truly blessed to have them as my mentors. I would like to extend my special thanks to them for dedicating their valuable time whenever I needed to discuss the research work. Working under their guidance has been a transformative experience, and I sincerely appreciate their support. I would also like to thank the reviewers of our papers for their valuable suggestions and comments which motivated me to improve my research work. I would also like to thank **Prof. Rachna Garg**, Head of the Department, and **Prof. Uma Nangia**, Chairman, of the DRC Electrical Engineering Department, Delhi Technological University, New Delhi for being a visionary and supporting the concept of the research work. I would like to use this opportunity to express my gratitude to all the university staff members who contributed to the cause. I extend my heartfelt thanks to my parents, in-laws, husband, and son whose encouragement has been a pillar of strength throughout my research journey.

Place: Delhi

Date: / /

Snigdha Chaturvedi

Research Scholar

2k17/PHDEE/07

ABSTRACT

All the practical systems in the real world are mostly non-linear in nature due to their dynamic behavior. Non-linear systems are extremely complex due to their uncertainties, parameter variations, and other complexities. Soft computing methods are commonly employed to address system dynamics and uncertainties. One of the most effective and successful soft computing methods is optimization. Due to numerous applications, optimization is a significant paradigm. Minimization or maximization are the forms of optimization that occur in nearly all engineering and industrial applications. The necessity to manage complex nonlinear processes with a high degree of uncertainty and satisfy performance requirements are the main drivers of progress in the field of control. Conventional control techniques cannot meet these objectives and have been proven ineffective for complicated nonlinear systems due to their complexity. To enhance the system's performance, an appropriate controller with optimal parameters is needed to be employed. Conventional Proportional-Integral-Derivative (PID) is still utilized in industries and other real-world applications because of its simplicity. This thesis uses PID controllers with various optimization techniques, such as Particle Swarm Optimization (PSO) and Teaching Learning-Based Optimization (TLBO) algorithm, to control non-linear benchmark systems such as artificial respiratory systems and vehicle cruise control systems. Performance indices were used to evaluate the suggested controllers' effectiveness. The robustness of the controllers was also examined. The suggested techniques were also used for the control and tuning of a non-linear ball and beam system by cascade-optimized PID. The controllers' ability to reject disturbances and variations in parameters was examined. Fractional order PID controllers (FOPIDs), which offer more controller flexibility, are frequently employed for the control of non-linear systems. Whale optimization algorithms (WOA), TLBO, and PSO are used in this thesis to tune the FOPID controller and control the inverted cart pendulum system. Several advantages of artificial neural networks have made them particularly attractive for use in modeling and controlling complex non-linear systems. They are capable of adaptation and self-learning. In this thesis, a new PID-like neural network is proposed, whose weights are optimized by the PSO algorithm. The optimization of weights with PSO has many advantages as compared to conventional back propagation learning since it is not based on gradient calculations. The proposed controller was tested for its effectiveness on a highly non-linear Continuous stirred tank reactor. It was used to control the temperature of the CSTR.

The proposed controller was also tested for robustness under disturbance application. It was also compared with the optimization-based PID controllers.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Stands for
PID	Proportional Integral Derivative
FOPID	Fractional Order Proportional Integral Derivative
PSO	Particle Swarm Optimization
ANFIS	Adaptive Network-Based Fuzzy inference system
GA	Genetic Algorithm
CSTR	Continuous Stirred Tank Reactor
TLBO	Teaching Learning Based Optimization
PSO-PID	Particle Swarm Optimization tuned Proportional Integral Derivative
TLBO-PID	Teaching Learning-Based Optimization Tuned PID
ABC	Ant Bee Colony
ZN	Zeigler-Nichols
IMC	Internal Model Control
AVR	Automatic Voltage Regulator
AGC	Automatic Gain Control
LOCA	Commercial Load
BLDC	Brushless Direct Current
WOA	Whale Optimization Algorithm
DE	Differential Evolution
GWO	Grey Wolf Optimizer
GOA	Grasshopper Optimization Algorithm
RVNL	Random Vector Functional Link
NN-PID	Neural Network Based PID
ANN	Artificial Neural Network
CBO	Colliding Bodies Optimization
ECBO	Enhanced Colliding Bodies Optimization
PSO-BP-PID	Particle Swarm Optimization and Backpropagation Tuned PID
BP-PID	Backpropagation Tuned PID

ACCS	Automobile Cruise Control System
ITAE	Integral Time Absolute Error
IAE	Integral Absolute Error
ISE	Integral Square Error
PD+I	Proportional Derivative Plus Integral
ACO	Ant Colony Optimization
WOA-PID	Whale Optimization Algorithm Tuned PID
WOA-FOPID	Whale Optimization Algorithm Tuned Fractional order PID
TLBO-FOPID	Teaching Learning Based Optimization Tuned Fractional Order PID
PSO-FOPID	Particle Swarm Optimization Tuned Fractional Order PID
PSO-NN-PID	Particle Swarm Optimization Tuned Neural Network Based PID
MSE	Mean Square Error
BPNN-PID	Backpropagation Tuned Neural Network Based PID
DGWO-Fuzzy WNN	Deterministic Grey Wolf Optimized Fuzzy Wavelet Neural Network
MIMO	Multiple Input Multiple Output

CHAPTER 1

INTRODUCTION

1.1 Overview

In the domain of control systems, handling nonlinear systems is still a challenging task, due to the unpredictable behavior of these systems. Several complex control problems related to linear systems have been solved efficiently in past years. But mostly all the systems in real life are non-linear like systems employed in aircraft, robotics, and biomedical, all are inherently non-linear [1]. There is a constant need to develop intelligent controllers that can evolve and adapt to these uncertainties and unpredictability offered by non-linear systems. PID controller remains one of the most popular controllers in the industry for decades due to its simplicity and ease of implementation [2]. However, system complexities and non-linearities make tuning the PID controller properly difficult with fixed gains [3]. Earlier PID controllers were mostly manually tuned till the 1940s when Zeigler and Nichols introduced the first empirical formula for tuning PID controllers [4]. Zeigler-Nichol's method and frequency response methods were the first developed formulas for PID tuning. Later on, Cohen-Coon [5] introduced a new method that aimed at disturbance rejection it showed some improvement but was not able to provide much better results as compared to the Zeigler-Nichols method. The Cohen-coon method was developed for first-order plus delay systems, the calculations were based on gain and time-delay of step response. The gain phase method was introduced in 1984 [6], and was an improvement over previously introduced empirical methods, it was able to move the critical point to a specific location in the Nyquist plot. Then the internal model control [7], method is another model-based tuning method that offers better robust control in terms of disturbance rejection and parameter changes. The IMC method was applicable only for systems with constant-time delays, but later, some researchers proposed modifications in this method for varying time delays. However, these empirical methods were based on certain assumptions and were not able to perform optimally, as they did not give the desired results [8]. In the process industry, lambda tuning has been widely used, particularly in the pulp and paper industries [9]. These traditional methods are designed mostly for stable and linear-time invariant systems. To deal with the non-linear systems some modifications were introduced in the PID tuning methods like gain scheduling and non-linear PIDs [10]. In the gain-scheduling method, the PID gains are represented as a function of errors. By scheduling control gains, this technique overcomes non-linear behavior; nevertheless, when learning capabilities are not incorporated into the control system, the system design becomes tedious. With the emergence of various AI methods, they have been used widely for the tuning of PID gains as they can adapt to the non-linearities, uncertainties, and parameter variations efficiently.

1.2 Structures of PID Controllers

PID controllers are used in two structures, series, and parallel form [11]. In a parallel form of PID structure, the three control actions are considered separately in three different terms and the final control action is generated by summing the effect of these three controllers. The control action $u(t)$ of a parallel controller is given as,

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{1.1}$$

where, K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, and $e(t)$ is the error signal. The block diagram of the parallel PID-controlled system is shown in Figure 1.1,

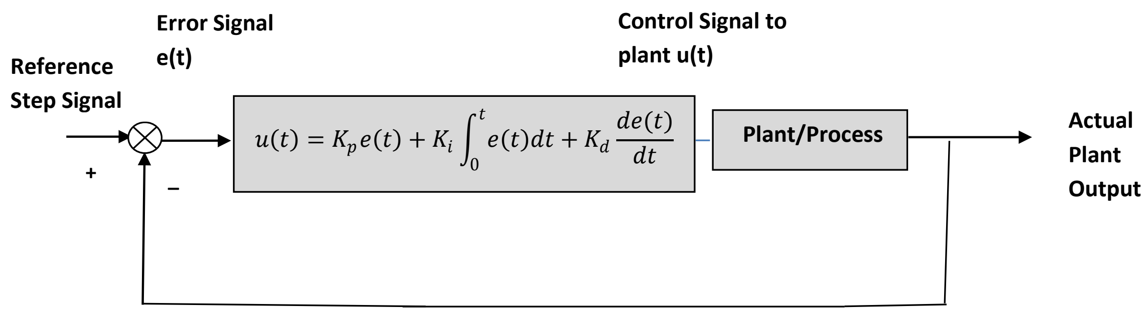

Figure 1.1 Block Diagram of a parallel PID-controlled system

The three gains of the PID controller are defined as [12],

1. K_p, Proportional Gain: It generates a control action proportional to the error signal. It helps in decreasing the large error. It also improves the response of the system.
2. K_i, Integral Gain: It generates a control action proportional to the integral of the error. Therefore, it reduces steady-state error and offset.
3. K_d, Derivative Gain: It generates a control action proportional to the derivative of the error. It improves the transient part of the response.

In earlier times, when pneumatic controllers were used in the industry, transfer functions of the PID controller were represented in series form. The series form of the PID controller is shown in Figure 1.2 [11],

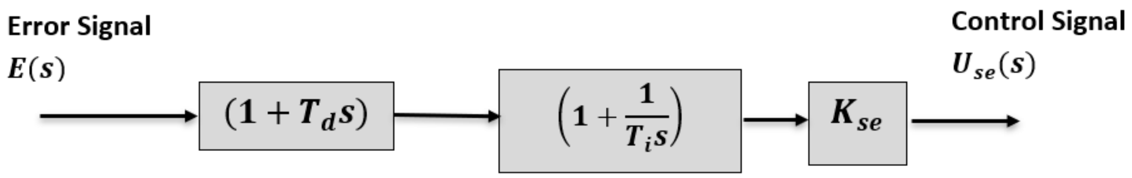

Figure 1.2 Series form of PID Controller

The mathematical representation of the PID controller in series form is represented as,

$$U_{se}(s) = \left[K_{se}\left(1 + \frac{1}{T_i s}\right)(1 + T_d s)\right]E(s) \tag{1.2}$$

where, K_{se} is the series proportional gain, T_i is the integral time constant and T_d is the derivative time constant. To maintain consistency with the analogue PID devices, certain manufacturers choose to keep this series structure. But mostly the PID controllers used today are in digital form and they are applied in parallel form.

1.3 Problems in PID Tuning for Non-linear Systems

Tuning of PID parameters becomes a difficult task when the system to be controlled is nonlinear [13],[14]. The common problems faced in PID tuning of non-linear systems are:

1. **Presence of non-linearities**: Various non-linearities like a dead zone, saturation, hysteresis, and backlash cause difficulty in tuning the PID controller to obtain the desired output. Due to the presence of the above-stated non-linearities, the system behaves differently in different operating regions. Therefore, it becomes difficult to set gains for the entire operating region.
2. **Interdependence of PID gains**: The PID gains are dependent on each other. If we change the value of one gain the performance due to the other gain gets affected. Therefore, it becomes difficult to tune them independently.
3. **Lack of known or accurate system dynamics**: It is very difficult to derive a mathematical model for a non-linear system. Therefore, it becomes challenging to obtain PID parameters for such a system.

To overcome the above-stated problems in PID tuning various AI-based methods can be used for proper tuning of PID gains [15]. These techniques should be capable of overcoming the non-linear effects and adjusting the controller gain in real-time to obtain the desired results. Nature-inspired algorithms, neural networks, and fuzzy logic are some techniques that were capable of handling nonlinearities and complexities present in the system and adjusting the PID parameters accordingly. PID is often used in different structures like feedback, feedforward, and cascade. Tuning of two PID controllers interconnected in cascade is a difficult task, because of the presence of interrelated parameters.

Cascade control is one of the most powerful control structures in which PID is implemented. It was first proposed by R.G. Franks [16] in 1956. A cascade control usually consists of two control loops, the primary and secondary loops. The cascade control structure offers various advantages as compared to a simple control structure [17]. It is efficient in rejecting the disturbances arising in the inner loop quickly and it also increases the speed of the system. However, the implementation of the PID controller in cascade configuration is complex, as two loops are to be tuned sequentially. The standard tuning methods available for PID tuning like Zeigler-Nichols [4] and Cohen-coon [5] are time-consuming and tedious as they are based on hit and trial methods. Some studies in the past have proposed auto-tuning methods for cascade control [18], [19]. But they were restricted to linear time-invariant systems only. Auto-tuning of cascade controllers becomes more difficult when they are applied to non-linear systems due to the presence of disturbance and parameter uncertainties [20]. Due to the challenges in the tuning of cascade PID controllers, the use of soft computing methods is becoming popular as they can reduce the effort of tedious and time-consuming calculations of sequential tuning [21]–[25]. Also, these methods can avoid manual retuning in the presence of disturbance and uncertainty.

In recent years researchers have shown a lot of interest in fractional-order PID controllers. A fractional order PID controller has five degrees of freedom instead of three in conventional PID controller, therefore, they offer more flexibility in design [26]. A FOPID controller offers various advantages as compared to a conventional PID controller. It provides better response for higher-order systems and time-delay systems. It also offers more robustness as compared to a simple PID controller [27]. However, due to the increase in degree of freedom, the tuning of the FOPID controller becomes a difficult task. Some of the methods used for tuning FOPID controllers were frequency domain methods, time domain methods, optimization algorithm-based and fuzzy logic, etc [27]–[29]. In a study [30], authors proposed a modified form of the Zeigler-Nichols method for tuning a FOPID controller. The parameters derived from the method were in quadratic equations. Some other researchers also proposed ZN-based tuning methods with some modifications to the original method [27]. One of the time domain methods used for tuning the FOPID controller is the ITAE method in which integral time absolute error is reduced to tune the parameters [31]. Another time domain method used for tuning of FOPID controller is the modification of Cohen-coon method which uses step response data for tuning the parameters [5]. In a study [32], a modified IMC method was proposed for calculating FOPID parameters. Some researchers have proposed the use of optimization algorithms for tuning FOPID controllers like PSO, genetic algorithms, and differential evolution [33]–[35].

1.4 Need of Intelligent Methods for PID tuning

In almost all the processes in real world we encounter dynamic systems with uncertainty in their structure and parameters [36]. Moreover, it is vital to fulfill the desired performance indices. Due to their inability to regulate nonlinear systems in real time, conventional control techniques are unable to meet these demands. Ku and Lee [36] have identified another drawback of conventional control: the nonlinear control laws are hard to derive. Deterministic models are therefore not a practical option for characterizing these uncertainties, and as a result, traditional control techniques are probably not going to provide the intended performance. To create the controller using these techniques, the plant's mathematical model must be understood in order to develop the controller utilizing these techniques. However, the dynamics of most plants are typically complicated and poorly understood. The performance of the designed system is usually compromised by the mathematical modeling of the plants being inaccurate. For these and other reasons, a variety of soft computing based nonlinear control strategies have been created. They are discovered to be capable of managing the complex non-linear structures [37].

1.4.1 Fuzzy Logic-based Controllers

Fuzzy logic is founded on the idea of fuzzy sets and was initially introduced by L. A. Zadeh in 1965 [38]. About fuzzy logic, he provides broader concepts in [39]–[41]. Furthermore, he introduces the concept of "linguistic variables," which he describes as a fuzzy set in his article. Scientists and researchers are interested in control engineering, one of the most well-known applications of fuzzy set theory.. Scientists and researchers attempted to improve the capabilities of conventional PID controllers, applying intelligent methods in the 1990s by employing innovative techniques like fuzzy logic [42]. To mimic the behavior of a typical PID controller, they attempted to integrate fuzzy logic control technology with a traditional PID controller. Thus, it is thought that a superior control system can be obtained by combining these two strategies. Fuzzy logic can set the PID gains by a rule-based approach, linguistic rules can be designed from different inputs like error, change in error, etc [43], [44]. Many researchers earlier utilized PI or PD controllers proposed by Mamdani [45]. Wang et. al. [46] presented an analysis and synthesis of a controller based on fuzzy and PID. Ketata et. al. [47] proposed a design based on parallel fuzzy controllers. Then in a study for the first time, fuzzy PID controllers were studied. Huang et. al. [48] proposed a practical fuzzy PID controller from conventional PID. Fuzzy logic and optimization algorithms can be hybridized to solve complex problems having uncertainty and optimization. The hybrid method has the advantages of individual methods, they can be integrated in the following different ways:

1. Fuzzy-based optimization: This method defines linguistic variables and rules to direct the optimization process using fuzzy logic. Optimization algorithms can benefit from the dynamic parameter adaptation provided by fuzzy rules, which makes it easier for them to navigate complex solution spaces [11], [49], [50].
2. Selecting fuzzy fitness functions: Fuzzy logic can be used to generate fuzzy fitness functions that account for imprecise or uncertain data, as an alternative to crisp objective functions. This method makes it possible for optimization algorithms to operate on naturally fuzzy real-world data [51].
3. Multi-objective Optimization: By allocating varying degrees of membership to various trade-offs, fuzzy logic can be utilized to manage several competing objectives. The Pareto-optimal solutions that balance these objectives can then be found by using multi-objective optimization algorithms such as MOGA or NSGA-II [52].
4. Adaptive control systems: An optimization algorithm's behavior can be changed in real time by using fuzzy logic to make it more responsive to changing external factors or the dynamics of the problem [53].
5. Decision Support Systems: By utilizing fuzzy logic to manage uncertainty in optimization algorithms' decision-making processes, decision-makers can make more informed decisions.

Fuzzy logic and optimization algorithms can be combined for optimization in any one of the above ways to tune the PID controller. The optimization algorithm can be used to find the best membership function to minimize the membership function. A few examples of such systems are genetic fuzzy systems in which fuzzy rules are obtained by genetic algorithm. In PSO-optimized fuzzy logic, rules of fuzzy logic are optimized by PSO [54]–[56]. In a study [57], authors have combined harmony search and cuckoo search algorithms to tune the parameters of fuzzy PID controllers. In another study [58], authors have used a modified whale optimization algorithm for tuning fuzzy PID parameters. In [59], authors have proposed an adaptive fuzzy PID controller whose output scaling factors are tuned online by a hybrid BFO-PSO algorithm. Therefore, fuzzy logic and optimization algorithms can be combined in various ways to improve the performance of PID. Till date several fuzzy PID structures have been presented, however, the fuzzy-logic controller has some disadvantages like,

1. Implementation of fuzzy based controllers requires expertise in rule designing and variable selection. A lack of expert knowledge can lead to undesirable results [60].
2. Fuzzy systems lack self-learning and accuracy and need to be altered by another entity. Due to the lack of emphasis on precision, this method may produce ambiguous findings.
3. They involve several mathematical calculations for obtaining the output therefore, they have high computational time [46].

1.4.2 Artificial Neural Networks (ANN) Based Controllers

The information processing systems built and implemented to mimic the structure of the human brain is called artificial neural networks. ANN performs a wide range of tasks, including pattern matching, optimization, data clustering, etc. The use of ANN in control systems was a natural step to meet the requirements of handling complex systems, meeting the required design specifications, and dealing with plants whose dynamics are not known [61]. The neural network is a powerful AI tool that utilizes the input and output data to learn the relationship of the system used and identify plant dynamics. An ANN is made up of a collection of highly interconnected processing components, or neurons, such that the output of each processing element is found to be related through connections to itself or to other processing elements. The link between the input and output nodes in a network can be determined by the Transfer function, also known as the activation function. It includes a degree of non-linearity that is ideal for most ANN application implementations. The sigmoid is the most widely used of the four primary categories of activation functions, which are the sine or cosine, linear function, hyperbolic tangent, and sigmoid [62]. The arrangement in which these neurons form different connection patterns is called network architecture. Different architectures in which neural networks are connected are single and multilayer feedforward networks, single and multilayer recurrent networks, and feedback networks [61]. In a multilayer network, there are several layers interconnected between the output and input. If no neuron in the output layer is an input to a node in the same layer or one that comes previously it, then a network is said to be feedforward. Conversely, a feedback network is created when outputs can be sent back as inputs to nodes in the same layer or to a previous layer. If the feedback is towards neurons in the same layer, it is called lateral feedback whereas, the recurrent networks have closed-loop feedback. A variety of training algorithms are employed in the training of artificial neural networks (ANNs). Four commonly used algorithms are the Gaussian covariance kernel, the Levenberg-

Marquardt (LM) algorithm, the Radial basis function (RBF), and the Multi-layer Feed-forward back propagation neural network (MLFFBPNN). The most popular type of neural network is called a multi-layer feed-forward back propagation neural network (MLFFBPNN. This algorithm seeks the optimal set of relation weights to minimize the predicted error between the virtual and the target values. Some researchers have applied different neural network-based methods to tune PID gains for various systems for example, a feedforward neural network was used to tune PID gains [21]. This method takes inputs and current PID gain values to determine output and next PID gain values. In the other type of neural network, a recurrent neural network is used in which the gains can be adjusted by the system parameter changes [25], [63]–[65]. The neural network has the advantage that it can handle system non-linearities, and complexities and can easily adapt to the system changes [62]. But mostly the neural network-based controllers used gradient-based algorithms which can lead to local optimal solutions instead of global optimal solutions[66].

In literature [67], two categories of hybrid methods are defined which combine ANN and fuzzy systems. Neuro-fuzzy systems are the first method whose main function is to process mathematical relationships. Numerous studies [68]–[70] create neuro-fuzzy systems by combining elements of neural and fuzzy techniques. Fuzzy neural systems are utilized in the second way to handle knowledge-based data represented as fuzzy numbers as well as numerical (determination-based) information. Neural networks made from fuzzy neurons are called fuzzy neural networks (FNNs). The primary feature of these networks is the synergistic cooperation of fuzzy theory and neural networks, producing models that combine the neural network's capacity for learning and the fuzzy systems' ability to handle uncertainty and interpretability. Conversely, an intelligent model's algorithm can train a fuzzy system, and this is known as a Neuro-Fuzzy Network (NFN). Considering this parallel, the combination of fuzzy logic and neural networks aims to mitigate the shortcomings of each of these systems, resulting in a system that is more effective, reliable, and simple to comprehend [68], [69], [71].

1.4.3 Introduction to Metaheuristic Algorithms

Metaheuristic algorithms have also gained popularity in recent years due to their efficiency and effectiveness in solving complex problems [72]. Most of the metaheuristic algorithms developed are not based on gradient optimization. They aim to improve fitness functions by finding optimal solutions to a problem. Since these algorithms are not based on gradient descent, they are immune from falling into the trap of local optimal solutions [73]. When a very good solution is originally identified but later becomes trapped in the solution and is unable to escape, it is referred to as a local minima trap [74]. Although most of the metaheuristic algorithms are nature-inspired some of them are not inspired by nature like differential evolution. Nature-inspired algorithms are optimization methods that are inspired by different natural phenomena like evolution, the growth of plants, the swarming of birds, fish schooling, etc. Since, their development these algorithms have found their applications in different areas like machine learning, data analysis, etc.[75]. These algorithms can search a wider search space for different possibilities of solutions. Several researchers have classified meta-heuristic algorithms based on their features. In a study [76], they were classified as evolutionary and non-evolutionary algorithms. In another study [77], they were classified as, evolutionary, swarm-based, physics-based based, and human-based algorithms. The algorithms used in this thesis for PID tuning of nonlinear plants are, PSO, TLBO, and WOA.

1.4.4 Particle Swarm Optimization

PSO is a swarm-based optimization algorithm that was first introduced by Kennedy in 1995[78]. The PSO algorithm mimics the social behavior of various animals, such as insects, fish, birds, and herds. These swarms follow a cooperative approach for food gathering, and everyone in the swarm continuously modifies the search pattern in response to its own and other member's collective learning experiences. In PSO each feasible solution is referred to as a particle, and the group of potential solutions is called a swarm. Two different forms of particle learning impact the search in PSO. Throughout the movement, every particle picks up knowledge from its own experiences as well as from those of other particles. Social learning is the process of learning from others, whereas cognitive learning is the process of learning from one's own experiences. Due to social learning, the particle memorizes the optimal option that every particle in the swarm has visited and this is referred to as the global best solution. The particle saves in its memory the best solution it has found on its own thus far, this is called the local best solution, because of cognitive learning [79]. Any particle's change in direction and magnitude is determined by a factor known as velocity. This represents the rate at which the position is changing over time. In this manner, the velocity could be characterized as the rate at which the position changes with the iteration. Given that the iteration counter rises by one, the velocity and the location remain unchanged. The movement of particles in the particle swarm optimization is shown in Figure 1.3 [78]:

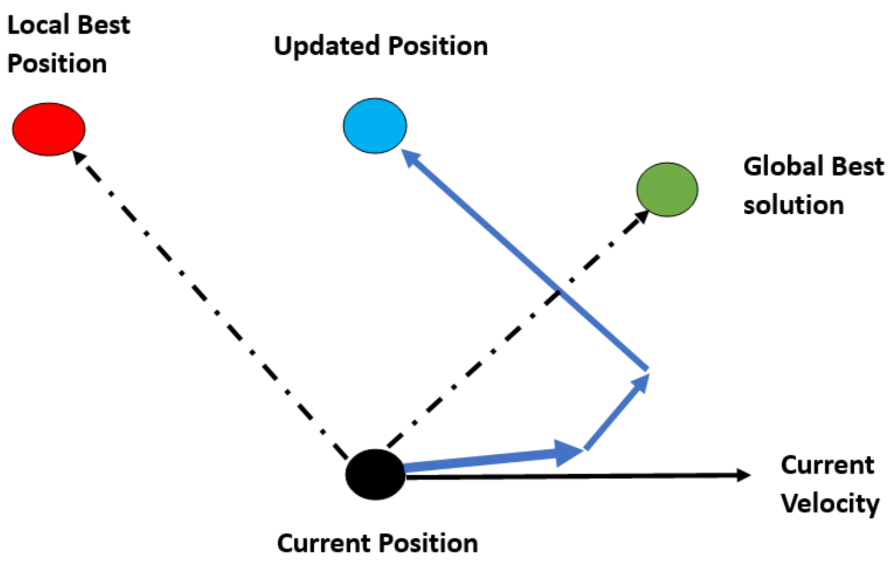

Figure 1.3 Movement of particles in PSO

Any population-based algorithm's capacity to identify an optimal solution and its rate of convergence is significantly impacted by the parameters that are chosen. Since the characteristics of the problem heavily influence the algorithm's parameter selection, it is typically not viable to provide a generic recommendation for it [78].

1.4.5 Teaching Learning-Based Optimization Algorithm

Teaching-Learning-Based Optimization (TLBO) is a recently developed metaheuristic algorithm in 2011 by Rao et. al. [80]. This population-based optimization technique was inspired by the knowledge transfer that occurs in a teaching environment, where students initially gather up knowledge from their teacher (also known as the Teacher Phase) and subsequently from other students (also known as the Learner Phase). The TLBO approach is depending on how a teacher's influence affects the student's performance in a class. In this case, the output is evaluated in terms of outcomes or scores. Most people view teachers as extremely intelligent individuals who impart their knowledge to students. The caliber of a teacher influences the student results, as a competent teacher prepares students to achieve greater outcomes regarding their grades or marks [81]. Like other, population-based techniques like GA, PSO, ABC, and HS, TLBO likewise uses a collection of solutions to arrive at the best one. Proper selection of algorithm parameters that impact algorithm performance is necessary for many optimization techniques. While PSO requires learning factors, weight variation, and maximum velocity, GA requires crossover probability, mutation rate, and selection technique; ABC requires the upper bound; HS needs the number of improvisations, pitch adjustment rate, and harmony memory consideration rate. Since TLBO does not require any algorithm parameters to be adjusted, it can be implemented more easily than other optimization techniques [82]. Like PSO, TLBO modifies the population's current solution by using the iteration's best solution, which raises convergence. TLBO, or Teaching-Learning-Based Optimization, appears to be a promising metaheuristic among several others with comparable results. According to reports, it performs better than a few well-known metaheuristics when it comes to continuous non-linear numerical optimization problems, constrained mechanical design, and benchmark functions [83],[74].

1.4.6 Whale Optimization Algorithm

Whale optimization algorithm metaheuristic algorithm proposed in 2016 [84]. It is a swarm intelligence method suggested for issues involving continuous optimization. It has been demonstrated that this method performs either comparably or better than several other algorithmic strategies. The way humpback whales hunt has served as an inspiration for the WOA. In WOA, each solution is condidered as a whale.. In this solution, the best among the group, a whale, attempts to repopulate a new location in the search space. Whales have two processes that they utilize to find their prey and attack it [85]. The prey are surrounded in the first, and bubble nets are made in the second. In terms of optimization, whales examine the entire area for prey, and they use this discovery during their attacking behavior. WOA is effectively used to support applications in a variety of domains, including engineering optimization issues. Creating a problem that makes sense and establishing variables and appropriate objective functions are the main things that can aid in addressing problems with engineering applications efficiently. The key issues with the optimization algorithm are population illustration, parameter setup, and operator usage [85].

1.5 Optimization Algorithms and Neural Networks

The optimization of weights and biases is a crucial aspect of artificial neural networks. In actuality, the two pillars of structure and learning algorithm set efficient ANNs. Gradient-based techniques have been applied to architecture training in numerous previous research. However, it has been shown that optimization techniques are necessary due to the drawbacks of gradient-based algorithms [86]. To apply BP in gradient-based learning methods, the cost function needs to be derivative. This is another drawback of learning methods that use gradients because the activation function and the cost function are not derivatives mostly. In these algorithms, sigmoid activation functions are frequently employed. Numerous gradient-based techniques, including Levenberg Marquardt (LM) and Back Propagation (BP) methods, have been developed in the literature to develop neural network-based systems. Yet, gradient-based techniques have some significant shortcomings like, being stuck often at the local optima, high computational time, and cost function must be a derivative [87]. Metaheuristic algorithms can easily escape the trap of local optima because of their exploration and exploitation capabilities. They can be used to optimize the weights and biases of the ANN system, to obtain better solutions [86]. In the past different optimization algorithms like PSO, GA, ant colony optimization, GA, etc. have been utilized with ANN to obtain optimal solutions [88], [89]. Neural networks and optimization algorithms can be combined to achieve a better response. We can first set the initial parameters using the conventional Zeigler-Nichols or Cohen-coon method. Then we can use optimization algorithms like PSO, and GA to search for optimal PID parameters based on the objective function. Finally, we can fine-tune the PID parameters based on the real-time input and output data. The fine-tuned parameters can adapt to the changes in the process parameters and set point changes.

1.6 Non-linear Benchmark Problems

The non-linear benchmark problems that are utilized in this study for testing the proposed controllers are given below:

1.6.1 Jacketed CSTR

CSTR (Continuous stirred tank reactor) is an important reactor in the chemical industry. In a CSTR continuous mixing of reactants takes place inside the reactor, and a uniform concentration of product is obtained. In a reactor, a substance is converted to another substance by a chemical process that is exothermic and irreversible. The CSTR exhibits various non-linear behaviors [90]. It is desired to maintain the temperature of a CSTR constant so that the reaction sustains under invariable conditions. In a jacketed CSTR, the temperature of the reactor is maintained by controlling the temperature of the fluid flowing in the jacket [91]. Since the reactions occurring in the reactor are exothermic, there is a sudden temperature change. Also, these reactors are affected by external disturbances and parameter changes. Therefore, this is a highly non-linear problem and control becomes difficult with standard controllers [92].

1.6.2 Ball and Beam Control

The Position control of a ball and beam system is a famous benchmark non-linear problem. The goal is to maintain the ball and beam system's position in balance The nonlinearities present in the ball and beam system are dead zone, saturation, and nonlinearity in the motor and pulley drive [93]. The control of a typical ball and beam system can be seen in various applications like the balance of objects by robots, and control of space vehicles. Many researchers have studied this problem in the past. The classical control theory fails because of the non-linearities present in the system. Researchers have suggested several control strategies for the ball and beam system, including fuzzy controllers, traditional PID controllers, and neural networks [9], [23]–[28].

1.6.3 Automobile Cruise Control System

An automobile cruise control system is a feature built into most automobiles nowadays. In an ACCS driver can set a particular vehicle speed and when ACCS is activated the speed of the vehicle is maintained at the reference speed [94]. ACCS also helps in collision prevention thereby, ensuring safety. A cruise control system also helps avoid collisions, reducing travel time and lowering fuel consumption. To embed these features in the ACCS several sensors are integrated. Due to sensor integration, ACCS control becomes a nonlinear control problem. The driver's reaction time and observation time cause a time delay in the system [95].

1.6.4 Inverted Pendulum-cart System

An inverted pendulum cart system is a popular non-linear benchmark control problem, in which an inverted pendulum is mounted on a moving cart. It is considered the simplest robotic system with a rigid body and a joint with rotational motion. The control problem here is to maintain the position of the inverted pendulum while the cart is moving. Although the structure of the cart system seems to be simple, standard control methods fail to control it [96]. The position control of the inverted cart system is difficult to control for many reasons, such as the fact that the system's geometric features are lost when the pendulum shifts into horizontal positions. There are no equilibrium locations on the output-zeroing manifold, the system's relative degree is not constant, and its controllability distribution lacks a constant rank [97]. Since inverted pendulum cart control is a benchmark problem for testing controllers, several researchers have applied various controllers to it, based on PID [98], fuzzy logic [99], [100], neural network [101], [102] for the control of inverted pendulum cart system.

1.6.5 Artificial Respiratory System

In an artificial respiratory system, a patient is supplied the required air artificially when he is not able to breathe on his own. It is a very important biomedical instrument. Although the first closed-loop artificial respiratory system was not available until the 1950s, still they were utilized to aid with respiration as early as the 18th century [103]. In an artificial respiratory system, the air is cycled into the lungs using mechanical bellows and valves, and a basic proportional (P) or proportional-integral (PI) controller was employed. These controllers were later implemented with microprocessors, and several closed-loop control suggestions have been made since then. The PID controller, which has gained acceptance in the industry for years, was also employed for the control of ARS [104], [105]. PID, however, has certain drawbacks, for example, it does not function effectively when the dynamics of the system are not constant. The connection between pressure and ventilation serves as an illustration of this. To avoid lung damage during ventilation, pressure must be changed by the level of ventilation. Dai et al. [106] employed two different algorithms to enhance controller performance: the PD algorithm is used in the first phase, and the PI method is engaged when the output pressure starts to remain constant. In addition to this, additional methods were employed to enhance the performance of the PID controller for control of an artificial respiratory system. These methods included the application of optimization techniques such as an automatic PID gain tuning process using particle swarm optimization (PSO) [107], fuzzy logic [108], neural network, etc [109], [110].

1.1 Outline of Thesis

Following the introduction, the rest of the thesis is organized into six parts, the brief description of each of these chapters is given below:

Chapter 2: **Literature Review**

This chapter gives a comprehensive review of the research works done in the past related to PID controllers, non-linear systems, conventional and AI-based PID tuning methods, and their implementation in non-linear systems. A detailed comparison of the existing AI methods is done comparing their advantages and disadvantages. The research gaps and contributions of the thesis are listed out in this chapter.

Chapter 3: Design of Optimized PID Controller by Evolutionary Algorithms

In this chapter optimized PID controllers are designed using a Teaching learning-based optimization algorithm and particle swarm optimization algorithm. The optimized PID controller was implemented on the automobile cruise control system and artificial respiratory system. This chapter also involves the mathematical modeling of these nonlinear systems. The proposed controllers are also tested for robustness in the presence of uncertainties in the system.

Chapter 4: Cascade Optimized PID Controller Implementation on the Non-linear Ball and Beam system

This chapter involves the design of a cascade PID structure for position control of a ball and beam system. The two cascaded PID controllers were tuned by the particle swarm optimization algorithm and teaching learning-based optimization algorithm. The controller was also tested for robustness by applying a disturbance signal. A comparative analysis was done with conventional tuning methods.

Chapter 5: **Hybrid PSO-Neural Network-based PID tuning for temperature control of non-linear CSTR**

In this chapter, a hybrid PSO-neural network-based PID tuning method is designed. A very simple PID-like neural network is proposed having only three neurons. The weights of the controller are optimized by Particle swarm optimization. The controller is applied for temperature control of a non-linear jacketed CSTR system. The controller design was compared with a backpropagation-tuned PID controller, PSO-tuned PID controller, TLBO-tuned PID controller, and conventional PID controller. The controller was then checked for robustness by application of a disturbance signal.

Chapter 6: Optimized FOPID controller for position control of an inverted pendulum system

This chapter proposes an optimized fractional-order PID controller for the position control of an inverted pendulum-cart system. The mathematical model of an inverted pendulum-cart system has been developed. The position control of the inverted pendulum-cart system is proposed by metaheuristic algorithms tuned to FOPID controllers. The responses of the proposed controller are compared with the previously proposed PSO-PID and TLBO-PID controllers.

Chapter 7: **Major Conclusions and Future Directions**

This chapter presents the major conclusions of the thesis and the future directions of the study.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a literature review is presented focusing on the problems that are faced while applying a PID controller to a non-linear system. Most real-world systems have dynamic behavior that varies with time, making them more complex. Nonlinear systems are receiving more attention from researchers due to their unpredictability and uncertainty. In the literature, a variety of controllers have been used to handle nonlinearities and improve system performance [111]–[113]. This chapter summarizes the fundamental and most current developments in the field of PID tuning for non-linear systems. A brief survey of the applications of different metaheuristic algorithms is presented. Then the application of different AI methods used for tuning the PID controller is discussed. This chapter also presents the control methods applied for non-linear benchmark systems (automobile cruise control system, artificial respiratory system, ball, and beam system, jacketed CSTR, and inverted pendulum-cart system).

2.2 Review of PID based Control of Non-linear Systems

A PID controller is still the widely used controller in the industry and the reasons for the popularity of PID controllers are that they are simple and easy to implement, even if the system is discrete the PID controller can be implemented with little modifications [114]. Earlier, the PID controllers were tuned manually, till the first tuning method available was proposed by Zeigler-Nichols in 1942 [115]. After that several formulae-based tuning methods were proposed like Cohen-coon [116] and Tyreus-luyben [117] etc. All these methods were mostly designed for linear systems and had some limitations, they were time-consuming and required tedious calculations. After formulae-based methods, some methods were proposed based on gain margin and phase margin [12] but they needed the exact mathematical model of the system, which is not available accurately in the case of non-linear systems. In the past, some significant studies have categorized the PID tuning methods effectively like Moradi et al [118] have classified them according to the availability of the model of the system. To deal with the non-linearities some modifications were made to the PID, like gain scheduling [119]. In gain-scheduling the gains are functions of error and, therefore, can adapt to the system variations.

However, it made the system design complex, and complete knowledge of the system was required to design the controller. Another method proposed to deal with the nonlinearity was model predictive control, which aimed at changing the controller parameters to reduce the error between desired and predictive response [44]. Then many researchers proposed different fuzzy-based PID tuning methods, most of them used error and change in error as the inputs [47].

2.3 Review of Cascade PID tuning methods

Cascade control is a powerful control method, which is used in higher-order systems for disturbance rejection. Cascade control is used as it can effectively reduce the disturbances occurring in the inner loop, because of its two-loop structure [20]. However, the tuning of two PID controllers is difficult due to the presence of interrelated parameters. In the past many researchers have suggested different methods for tuning of cascade PID controllers. In a study [20], authors have suggested an auto-tuning method based on a relay feedback test. The proposed method was tested on two different systems and was found robust and efficient. In another study [120], authors have proposed a PSO-based tuning for cascade PID control of a servo pneumatic system. The controller was compared with self-tuning PID control. J. Zhang et. al. [21], proposed a cascade PID control in which the primary controller was a neural network-based controller. The controller was tested on a superheated process. Recently, in a study [121], authors have proposed cascaded FOPID controllers tuned by sliding mode control to control the output of an artificial respiratory system. In another study [122], recently authors have controlled CSTR efficiently by cascade control. It was found that the cascade control performs better than the feedforward and feedback control.

2.4 Review of Tuning of Fractional-order PID controllers

The FOPID controllers have five degrees of freedom, instead of three in conventional PID controllers. It means in a conventional PID three parameters can be varied independently, whereas, in a FOPID five parameters can be varied. A FOPID controller has two additional parameters integral power and derivative power, due to the additional degrees of freedom the controller has wider flexibility and, therefore, gives more robustness to the system [123]. In the literature, some studies have shown that the FOPID controller performs better as compared to the PID controller [121], [123]–[125]. However, the tuning of an FOPID is more complex than PID because of the presence of five degrees of freedom. In a study [126], authors have compared analytical and heuristic methods of tuning FOPID controllers for first-order plus dead time systems. In another study recently [127], authors have proposed a fuzzy-based fractional order PID controller for a buck converter. It was found that the controller gave a better performance in terms of robustness and efficiency than the PID controller. In another study [124], authors have proposed an efficient method to tune the FOPID controller for the liquid level control.

2.5 Review of Metaheuristic Algorithms

As the name implies, metaheuristic optimization approaches are problem-independent control strategies that have become increasingly popular when applied to challenging engineering problems [128]. This can be understood by their adaptability, simplicity, and high rate of efficiency in solving difficult tasks. Metaheuristic techniques rely heavily on the notion of randomness to find optimal solutions through intensification and diversification. Recent research suggests that optimization approaches based on heuristic algorithms have become a powerful tool for resolving a range of control engineering challenges [129]. Researchers have made considerable use of metaheuristic algorithms because of their quick response times, high optimization capabilities, and straightforward architecture. Metaheuristic algorithms are more effective in solving higher-dimensional optimization issues than classical optimization techniques [77], [130], [131]. They provide the best solution as compared to other AI methods because of several reasons:

1. They explore a large search space effectively and hence provide the best solutions that are often missed by other methods [76].
2. They can be easily integrated with the existing classical controllers [132]
3. It is not based on gradient information [132]

2.5.1 Review of Particle Swarm Optimization

The particle swarm optimization algorithm was proposed by Kennedy [78], in 1995. It is a swarm-based optimization technique based on fish schooling or bird flocking. A swarm represents the population. The optimization process mimics the process of searching for food by the swarm. Each candidate updates its velocity and position by achieving maximum fitness value. Since its development, many complex non-linear problems have been solved by PSO. Meetu Jain et. al. [133] reviewed all the modifications done in the PSO algorithm and its efficacy in solving complex problems to date. Z. Bingul et al. tuned the parameters of integer order and fractional order PID controllers using PSO and ABC algorithms [33]. The results showed that the PSO algorithm gave better results in terms of external and internal disturbances. Khanduja et al. [134] compared the performance of Z-N-tuned PID, IMC-tuned

PID, and PSO-tuned PID on a non-linear CSTR and found that the PSO-tuned controller gave the best results. A. Agalya et al.[135] proposed a PSO-tuned PID controller for concentration control of CSTR and found that the controller performed better than a conventional controller. R. Parouha et al.[136] reviewed a wide range of meta-heuristic algorithms and found that the PSO and DE outperformed all the other algorithms in terms of efficiency. A. Yimchunger et al.[107] efficiently applied a PSO-tuned PID controller for an artificial respiratory system. PSO has proved to be one of the most widely used metaheuristic algorithms due to its simplicity and ease of implementation [133], [137]. Several researchers have proposed modifications, extensions, and hybridization of PSO [133]. In a study, researchers have proposed concurrent PSO which aimed at increasing the convergence speed of PSO [138]. Then Kennedy proposed a binary PSO, the algorithm applied to both continuous and discrete objective functions. In subsequent years other modifications were proposed like fuzzy-PSO [53], guided PSO [79], and self-regulatory PSO [139].

2.5.2 Review of TLBO Algorithm

The teaching-learning-based optimization algorithm was introduced by Rao et al.[80] in 2011. The algorithm has many advantages over various metaheuristic algorithms proposed to date[130]. It has been efficiently applied to complex problems and has given better results [80]. S. Chatterjee et al.[140] applied the TLBO algorithm for PID tuning of the AVR system and found the response better as compared to conventional methods. V. Srivastava et. al.[113] proposed a comparative analysis of different optimization algorithms categorically, it was concluded that the human-based algorithms performed better as compared to others. B. Sahu et. al.[141] proposed a fuzzy-PID controller for automatic generation control (AGC) of a two unequal area interconnected thermal system and they used the TLBO algorithm for optimization. The proposed controller was compared with the simulated algorithm, genetic algorithm, and LOCA algorithm. A. Lins et. al. [142] proposed a TLBO-tuned PID controller for a PV-fed BLDC motor. The response was compared with PSO-tuned and conventional PID and was found better in terms of performance indices and robustness. J. Bhookya et. al. [143] proposed a fractional order PID controller tuned by the TLBO algorithm for a multi-variable system. The proposed controller was effective in eliminating the interaction between loops. A. Tiwari et al. [144] optimized different manufacturing processes by the TLBO algorithm effectively. The review work presented a large-scale application of the proposed algorithm. Despite being a new algorithm TLBO has been successfully applied to various science and engineering fields [81]. Many researchers have proposed modifications to the original TLBO algorithm for different applications [145]–[149].

2.5.3 Review of WOA (Whale optimization algorithm)

The whale optimization algorithm was proposed by S. Mirjalili et al. [84] in 2016. The algorithm was competitive as compared to the algorithms proposed to date. It mimics the hunting behavior of humpback whales. M. Alquaness et al. [89] used six metaheuristic algorithms for predicting crude oil, it was concluded that PSO and WOA performed better as compared to other algorithms. S. Vavilala et al. [150] applied WOA and PSO for the optimization of fractional-order Internal model control and controlled the height of a conical tank system. The proposed controller was able to reject the disturbances more efficiently than other controllers. F. Gharehchopogh et al. [151] gave a detailed comprehensive review of the whale optimization algorithm and its applications in various engineering fields like image processing, clustering, pattern recognition, classification, etc. A. Kumar et al. [152] suggested employing the whale optimization algorithm (WOA) to optimize the Fuzzy-PID + PID hybrid controller for a hybrid power system's frequency control. In a study [153], authors have proposed optimized PID controllers for two interactive surge tank systems. They compared feedforward control, GA, PSO, and bubble-net whale algorithm-tuned PID controllers. It was found that the bubble net whale algorithm tuned PID performed better.

2.5.4 Review of Hybrid Optimization Algorithms

The application of typically prevalent metaheuristics characterized the first two decades of metaheuristics research. But it is evident now that concentrating just on one metaheuristic has its limitations. A hybrid metaheuristic is an algorithm that combines a metaheuristic with additional optimization techniques that may provide better flexibility and efficient behavior in real-world and large-scale scenarios. The complementary properties of metaheuristics on the one hand and all-encompassing strategies like branch and bound techniques or mathematical programming on the other can be combined to achieve this. Hybrid algorithms combine two or more algorithms to efficiently and concurrently tackle a given problem [128]. Researchers have proposed many hybrid algorithms to enhance the optimization of a particular problem. In a study, a novel PSO-GA hybrid algorithm is proposed to utilize variable population size of genetic algorithms [154]. In another study, a PSO and GWO hybrid optimization was proposed which used exploitation of PSO and exploration of GWO to enhance the performance of the hybrid algorithm [155]. Recently, many hybrid optimization algorithms have been proposed to solve complex problems in different fields [156]–[158].

2.6 Review of Existing Artificial Intelligence (AI) Methods to tune PID controller

With the development of fuzzy logic, many researchers applied it to designing fuzzy-based PID controllers [159]. The fuzzy-based controllers generally used error and change in error to obtain PID gain values. In a recent study fuzzy PID controller [55] was introduced for multi-area power systems. It was also tested for robustness in the presence of uncertainties and disturbances. In one more study, a fuzzy PID controller was proposed to maintain the oxygen level for fish survival. The proposed fuzzy controller was improved by particle swarm optimization gated recurrent unit [160]. The fuzzy-based controllers had the advantage of handling the non-linear effects and complexities of the system. However, fuzzy logic requires expert knowledge to determine appropriate membership functions and suffers from computational complexities [8]. With the development of metaheuristic algorithms, like Genetic algorithms and differential evolution, they were used widely for solving various non-linear problems. They searched the entire search space to find the optimal solutions. Therefore, they were able to handle the non-linearities efficiently providing the optimal solution. In a study [161], authors applied several metaheuristic algorithms for human activity recognition efficiently. In a study in the year 2009 [162] authors proposed a self-organizing GA-based tuning method for PID controllers, it was found more effective than conventional PID tuning methods. Some researchers used a Differential evolution algorithm for PID tuning and applied it to different non-linear systems efficiently. However, the convergence of the results depended on the proper selection of algorithm parameters. Then, the development of PSO overcame these difficulties and became popular among researchers [79]. It was utilized for PID tuning by many researchers [134], [142], [163]. Recently, many researchers have used different metaheuristic algorithms like Teaching learning-based optimization [140], [142], [164], ant colony optimization, class topper optimization, gravitational search algorithms, etc. But these algorithms suffered from the disadvantage of slow convergence speed and again optimal solutions were obtained if the parameters of the algorithms were selected properly [75], [165]. To overcome these disadvantages some researchers proposed hybrid algorithms which combined the advantages of two algorithms and performed mostly better than the original algorithms [128]. In a recent study [166], authors have proposed a hybrid particle swarm optimization-cuckoo search optimization algorithm to tune fuzzy PID parameters of a micro gas turbine. In a study, authors have compared different metaheuristic algorithms like PSO, DE, GA, grey wolf optimization GWO, grasshopper optimization GOA, and proposed hybrid GA-GOA algorithms for the optimization of a staggered heliostat field for a PS10, a solar plant. The results showed that the hybrid algorithm performed better than the individual ones. In another study [167] authors proposed a hybrid algorithm based on an enhanced Aquila optimizer that uses the search mechanism of the slime Mold algorithm for the prediction accuracy of CO_2 trapping. The proposed hybrid algorithm performed better than the individual ones.

The neural network was also applied to various non-linear systems efficiently by the researchers [168]–[170]. They have high computational speed and have been used in the past for determining PID parameters efficiently [171]. In [172], authors have proposed an optimized Artificial neural network model called random vector functional Link for wind power prediction efficiently. Hui Liang et. al. [173] proposed a backpropagation-based PID Neural network controller for temperature control of a room heater. The controller performed better as compared to the conventional PID controller, as it was able to adapt easily to the parameter changes. In another study [174] authors have proposed an adaptive population extremal optimization for the initialization of a PID-based Neural network for a multi-variable system. Long Zhang et. al. [170] proposed a backpropagation-based NN-PID controller to suppress combustion instability in a cylindrical rijke tube. Therefore, back-propagation can be combined with the PID controller to obtain self-adaptation in the system. But back-propagation has a lot of disadvantages as compared to metaheuristic algorithms as they depend on gradient decent information because of which there are chances of being stuck in local optima whereas, metaheuristic algorithms search a wider space and provide a global optimal solution [73], [131]. Secondly, metaheuristic algorithms are less sensitive to initial conditions and constraints on the objective function as compared to back-propagation algorithms. Metaheuristic algorithms also provide a faster convergence rate as compared to backpropagation making them more efficient for various applications [168]. In a study [88], authors have trained two ANN networks, a feedforward neural network and a radial basis neural network using four metaheuristic algorithms, PSO, GA, Colliding bodies optimization (CBO), and Enhanced colliding bodies optimization (ECBO) algorithms. The results found were better than the backpropagation algorithm. Recently, in a study [89] authors have proposed a dendritic neuron model for crude oil prediction and forecasting, in the study they utilized six different metaheuristic algorithms to enhance the training process and selection of various parameters. In a recent study [66], authors have combined PSO with Backpropagation to obtain initial conditions for a PSO-BP-PID controller for parallel stabilized platforms in marine operations. The results of the proposed PSO-BP-PID controller were compared with BP-PID and PID controller and they were better as compared to them.

2.7 Review of Application of Various Control Techniques on Non-linear Benchmark Systems

To test the efficiency and efficacy of the proposed controllers in the thesis, the controllers are applied to some benchmark nonlinear systems. The nonlinear systems used in the thesis are as follows:

2.7.1 Control of Automobile Cruise Control System

An automobile cruise control system is nowadays, an added feature in most vehicles. In 1997 a study [94], authors proposed a review of cruise control and compared two cruise control methods, one based on fuzzy logic and the other based on classical control. In a study, authors have proposed GA-based tuning of PID controller for an automobile cruise control system [175]. In another study [176], authors have proposed an ant lion optimizer for PID tuning of an automobile cruise control system. Recently, in a study [177] author proposed a fuzzy PD plus I controller for an ACC and compared the results with classical tuning methods. In [178], authors have proposed a robust adaptive controller for cruise control of high-speed trains based on the Lyapunov method. The controller was robust enough to deal with the uncertainties and disturbances in the train. In [179], authors have presented a controller that is capable of modifying its structure by the disturbance signal present. Recently in a study [180], authors have proposed a novel arithmetic optimization algorithm. In a recent study, authors have proposed a proportional, fractional order integral, the derivative plus double derivative with filter tuned with an elite opposition tuned algorithm. The controller was tested with various operating points and uncertainties [181].

2.7.2 Control of an Artificial Respiratory System

An artificial respiratory system is an automatic ventilation system that helps the patients in breathing. Some studies have suggested the mathematical Modeling of the ventilation system [103], [105], [182], [183]. To control the required airway pressure to the patients some controllers have been suggested in the past, like adaptive pole placement control [184]. In a study [108], authors have proposed a fuzzy PID controller for the respiratory system, which gave a quicker response with less overshoot. In another study [185], authors have proposed a direct adaptive control for an artificial respiratory system. Recently in a study [182], authors have applied a hybrid optimization tuned fractional-order PID controller for the efficient control of airway pressure in an artificial respiratory system. In another study, authors have proposed a fuzzy sliding mode controller for an artificial respiratory system [186]. In [187], authors have applied a fractional order PID controller tuned by BFO and PSO to control the artificial respiratory system.

2.7.3 Control of a Jacketed CSTR

CSTR (Continuous Stirred tank reactor) is an important reactor in which continuous stirring is done. It is used mostly in chemical industries [92]. Temperature and concentration control of a jacketed CSTR becomes a challenging task due to the non-linear dynamics of the system. The control of these non-linear reactors is a big problem for chemical engineers working in the industry. In 1995 B. Wayne Bequette [188], gave a comparison of non-linear control methods used for the control of CSTR. In the study, it was found that non-linear predictive control performed better as compared to other methods. In, [189] the CSTR control is done by a fractional order PID controller tuned by a new hybrid algorithm combining the chaotic maps, SMS, and Elite opposition-based learning algorithm. The results were compared with the other optimization algorithm-tuned FOPID and PID controllers. In another study [190], the authors have proposed a wavelet neural network for CSTR concentration and temperature control. The weights of the neural network were optimized by grey wolf optimizer and the initial parameters were decided by Mamdani fuzzy rules. For control of the continuous stirred tank reactor (CSTR) in the presence of an external disturbance, an asynchronous sliding mode control design approach based on the event-triggered technique is suggested in another study [191]. In another study, authors have proposed a unique approach to controller design for an unstable nonlinear continuously stirred tank reactor (CSTR) chemical system based on artificial bee colony (ABC) algorithms [192]. The results were compared with different controllers proposed in past literature. In the past some other researchers have also applied different controllers to control CSTR temperature based on classical controllers, neural networks, metaheuristic algorithms, [91], [111], [198], [122], [135], [191], [193]–[197], etc.

2.7.4 Control of the Ball and Beam System

Ball and beam are one of the most important laboratory experiments. It can be used to study most of the control methods. It is considered a benchmark problem in non-linear systems. In the past, a lot of researchers have applied different control mechanisms like classical and modern methods to balance the position of a ball in the system. In 1978, P.E. Wellstead et. al. [93] described the experiment and modeled the differential equations of the system. They also proposed the beam transfer function and ball transfer function. Then afterward few researchers tried to apply a fuzzy logic controller to balance the ball position in the experiment [93], [199], [200]. Recurrent neural networks were also applied to the ball and beam system control[201].

Some researchers applied hybrid AI methods to better control ball and beam systems. Yeonghwa-Chang et. al. [202] applied fuzzy sliding mode controller enhanced by parameter optimization using ant colony algorithm. Sung-Kwun Qh et. al. [199] used fuzzy cascade controllers optimized by parallel genetic algorithms. Hybrid learning control algorithms can also be used for controlling ball and beam systems efficiently [203]. In a study author, the author examined the ball and beam system's fractional order control at two degrees of freedom. It uses a model-based approach to create the controller settings for the matching linear model [125]. In another study, authors have proposed the harmony search algorithm's dynamic parameter adaptation employing several fuzzy system types applied to the membership function optimization of a benchmark control issue, the ball and beam controller [204].

2.8 Research Gaps

1. The problem of PID tuning has been identified as a problem in the literature for non-linear systems. Since standard tuning methods fail when they are applied to non-linear systems.
2. The advantages of metaheuristic algorithms have been frequently described in the literature. In many studies, it has been observed that they have solved many complex problems.
3. However, the effectiveness of the efficient application of the optimization algorithm to a particular problem depends on the number of parameters of the algorithm as it directly increases the complexity. TLBO (Teaching Learning optimization) and PSO (Particle swarm optimization) are two algorithms highlighted in the literature that have fewer parameters. They have solved many non-linear problems efficiently.
4. But these algorithms suffered from the disadvantage of slow convergence speed and again optimal solutions were obtained if the parameters of the algorithms were selected properly.
5. The neural network was also applied to various non-linear systems efficiently by the researchers. They have the ability to compute fast and have been effectively employed in the past to determine PID parameters. They can adapt and learn easily; therefore, Neural networks can efficiently overcome setpoint changes and parameter variations.
6. But back-propagation optimization has a lot of disadvantages as compared to metaheuristic algorithms as they depend on gradient decent information because of which there are chances of being stuck in local optima whereas, metaheuristic algorithms search a wider space and provide global optimal solution.

2.9 Motivation

PID is one of the controllers that is currently most commonly utilized in the industry. However, it is shown in past studies that almost 90% of the controllers used in the industries are poorly tuned. Conventional PID tuning methods are based on hit and trial, they are time-consuming and costly. So, there is a need for the development of new methods to improve the efficiency and effectiveness of PID controllers. With the development of AI methods, different problems can be solved efficiently with these methods. AI methods can also improve system response by considering the robustness of the control system. In dynamic systems where there are set point changes and parameter variations conventional tuning methods fail. AI tools such as optimization algorithms, neural networks, and hybrid AI techniques can be applied and tested for PID tuning of systems having non-linearities in them.

2.10 Contributions

The major contributions of this thesis are given below:

1. The development of an optimized PID controller for an adaptive cruise control system and artificial respiratory system is done by applying the TLBO (Teaching-learning-based optimization) Algorithm and PSO (particle swarm optimization) Algorithm.
2. The developed technique was applied to cascade PID controllers for a non-linear ball and beam system and the responses were compared with conventional tuned cascade PID controllers. The proposed controller was also tested under the application of a disturbance signal.
3. A novel neural network-based PID controller structure is proposed in the thesis. The optimization of the proposed neural network like the PID controller is done by PSO. The controller is tested for the temperature control of a non-linear CSTR.
4. The fractional order PID controller was optimized using different optimization algorithms and was applied to control the position of an inverted pendulum cart system.

CHAPTER 3

DESIGN AND IMPLEMENTATION OF EVOLUTIONARY ALGORITHMS TUNED PID CONTROLLER FOR NON-LINEAR SYSTEMS

3.1 Introduction

PID controller tuning is a problem when applied to non-linear systems. The empirical formula-based tuning methods are inefficient as they cannot adapt to the non-linearities and parameter changes in the system. Metaheuristics-based algorithms can be used to tune the PID controller applied to non-linear systems [110], [205]–[209]. In this chapter, we have considered two non-linear problems, automobile cruise control systems and artificial respiratory systems. Particle swarm optimization and teaching-learning-based optimization algorithms are used to tune the PID controller applied to these systems. Cruise control is nowadays an added feature in most automobiles. It reduces driver fatigue while driving on highways and low-traffic areas. It also helps in reducing the probability of collision between vehicles, improves fuel optimization, and reduces traffic congestion [210]. When a vehicle is running on Automobile cruise control mode (ACCS), the speed is controlled and maintained to the reference speed set by the driver without the application of the accelerator. This reduces the collision of vehicles. Therefore, ACCS plays an important role in safety and collision prevention. Recently, automobile vendors have added several features to ACCS. But its main advantages are collision prevention, traffic congestion reduction, and fuel optimization. To embed these features efficiently in ACCS, the sensor information is integrated with the controller which takes the desired control action to the throttle system and brakes of the vehicle. In an ACCS, the nonlinearities present are mainly due to sensors integrations and the time delay in measuring, controller action etc., which affects the system stability. Therefore, an effective and robust control system is desired for automobile cruise control system. But due to presence of non-linearities tuning of PID parameters becomes difficult. In the past various methods have been proposed to control ACCS systems. Y. Prakash et. al. [211] developed a mathematical model for a cruise control system for an inclined road and plane road. Then they applied the PID controller for different conditions. A. Kuyumku et. al. [212] applied a neural network-based controller for the speed control of an automobile cruise control system efficiently. In a study [213], researchers have proposed an optimal robust controller for the control of an automobile cruise control system. Some more researchers have applied conventional PID controllers, fuzzy logic, neural network controllers, and genetic algorithms to control automobile cruise control systems [179], [212], [214]–[217]. In an ACCS system tuning of PID gains becomes a crucial task due to the presence of the sensor and integration of its information to the feedback loop introduces non-linearity in the system. Therefore, standard PID tuning techniques cannot be applied to this system. Earlier some researchers applied fuzzy logic for PID tuning for automobile cruise control systems but fuzzy logic implementation requires expert knowledge for desirable results [177]. In a study [177], researchers have proposed a fuzzy based PD plus I controller for automobile cruise control system. Some have also used genetic algorithms for PID gain optimization effectively. Therefore, metaheuristic algorithms can be used for PID gain optimization of an artificial cruise control system [160], [176], [210].

The other benchmark non-linear system used in this chapter is the artificial respiratory system. In ICUs (Intensive care units) we mostly use mechanical ventilators to support the breathing of the patient. It is an artificial respiratory system that provides appropriate respiratory support to the patient by controlling the flow of gas, pressure, and volume and maintaining the composition of gases [103],[218]. If we look at the history of mechanical ventilation techniques earlier iron lungs were used which had many disadvantages such as size, heavyweight, and difficult control mechanism. At present these techniques are completely replaced by ventilation systems based on positive pressure [103]. The most challenging task while controlling an artificial respiratory system is to set optimal parameter values according to the patient's self-respiration capability [110]. Bram Hunnekens et. al. introduced a variable gain method for the control of respiratory systems [105].

In the past, many researchers have developed various tuning methods for tuning PID gains based on various artificial intelligence techniques. M. J. Mahmoodabadi et. al. used a gravitational search algorithm to obtain optimized PID parameters for a ball and beam system [219]. Alkrwy et. al. used the Crow search algorithm for tuning PID parameters [220]. Yimchunger et. al. [107] developed a PSO-tuned PID controller and compared the results with conventional methods. In [108] authors developed a fuzzy-based PID controller for the controlling airway pressure of an artificial respiratory system. However, the fuzzy-based controller needs expert knowledge for rule formation. Therefore, to control the airway pressure of the artificial ventilator metaheuristic algorithms can be used to tune the PID controller.

3.2 Mathematical Modeling of an Automobile Cruise Control System

The vehicle velocity of a cruise control system is regulated by the reference value of the velocity set by the driver. The block diagram of the cruise control system considered for this work is presented in Figure 3.1.

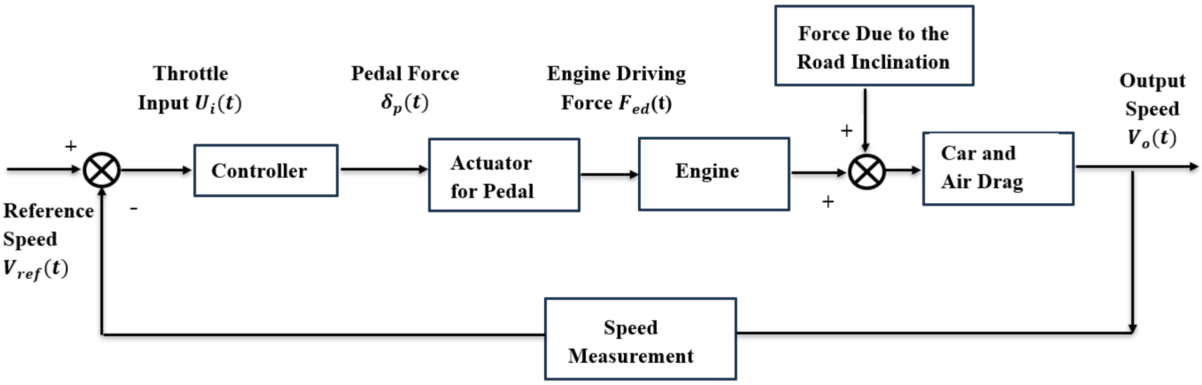

Figure 3.1 Block Diagram of Automobile Cruise Control system

The error signal generated between the desired reference velocity and actual output velocity is given to the controller, which is given to the throttle. The input for the throttle is denoted as $u_j(t)$ which is required to reduce the error between actual speed $V(t)$ and desired reference speed $V_{ref}(t)$. In case the road inclination angle θ increases it also generates a pedal force $\delta_p(t)$. The longitudinal dynamics of a vehicle are derived on the basis of Newton's force equations [221]:

$$F_{ed}(t) = m_v \frac{dV(t)}{dt} + F_{aero}(t) + F_{gr} \tag{3.1}$$

In (3.1), $F_{ed}(t)$ is the engine force that is generated by the injection of fuel in the required quantity. The control of fuel is done by the throttle output. m_v is the vehicle mass in Kg, $m_v \frac{dV(t)}{dt}$ is the force due to inertia which opposes the engine force, $F_{aero}(t)$ is the aerodynamic force which opposes the engine force. Aerodynamic force is the force because of the air density and speed of the vehicle. It is expressed as:

$$F_{aero}(t) = C_{aero}(V(t) - V_w(t))^2 \tag{3.2}$$

Where, C_{aero} is the aerodynamic coefficient $V_w(t)$ is the wind gust speed in Km/hr

$$F_{gr} = m_v g \sin\theta \tag{3.3}$$

Where, F_{gr} is the gravitational force on the vehicle and θ is the inclination angle of the road. Generally, the actuator in a cruise control system is represented as a lag system of 1st order. If T represents the observation time and τ_o is the time of the driver reaction. Then, the engine's driving force now becomes,

$$F_{ed}(s) = \frac{C_O e^{-\tau_o s}}{Ts+1} \tag{3.4}$$

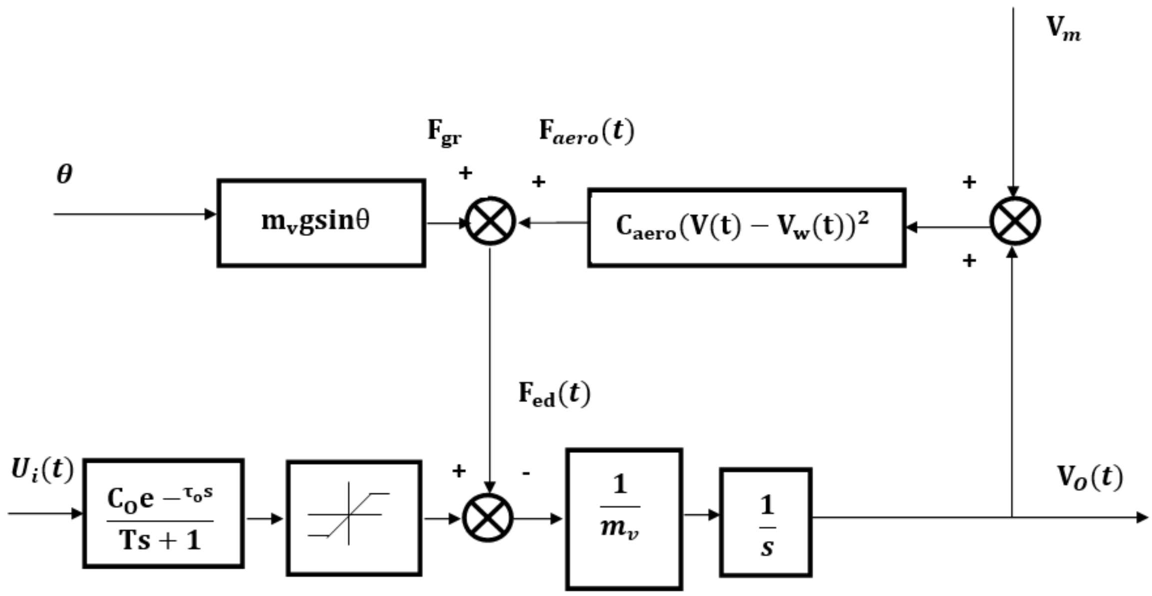

Figure 3.2 Longitudinal Dynamics model of Cruise Control system

Now we must simplify the above model for controller designing, to simplify the model we will assume all initial conditions to be zero and neglect the disturbance parameters like wind gust speed $V_w(t)$ is assumed to be zero. The model is hence now simplified as a unity feedback loop. Here the state variables are chosen as output speed V(t) and engine driving force F_{ed}(t). The state space equations derived from equations (3.1), (3.2), (3.3), and (3.4) are as follows:

$$\dot{v}(t) = \frac{1}{m_v}(F_{ed}(t) - C_{aero}v(t)^2) \tag{3.5}$$

$$\dot{F_{ed}}(t) = \frac{1}{T}(Cu(t-T) - F_{ed}(t)) \tag{3.6}$$

$$y = v(t) \tag{3.7}$$

As we can see from the above equation (3.7) this is a non-linearized state space model due to the presence of the term $v(t)^2$ in the equation. So, we will linearize this model by differentiating the state space equations. After differentiating on both sides, we get:

$$\frac{d\dot{v}(t)}{dt} = \frac{1}{m_v}(-2C_{aero}v(t)\delta v(t) - \delta F_{ed}(t)) \tag{3.8}$$

$$\frac{d\dot{F_{ed}}(t)}{dt} = \frac{1}{T}(C\delta(t-T) - \delta F_{ed}(t)) \tag{3.9}$$

$$y(t) = \delta v(t) \tag{3.10}$$

The equation shows that the output variable δv is a discrete function and δF_{ed} is also a discrete function. $\delta(t-T)$ is the delay time of the engine. After linearizing the model, the transfer function is derived as follows:

$$\frac{\Delta V_1(s)}{\Delta U_1(s)} = \frac{K_1 e^{-\tau s}}{(s+K_2)(s+K_3)} \tag{3.11}$$

Where, $K_1 = C/m_v T$, $K_2 = \frac{2C_{aero}v}{m_v}$ and $K_3 = \frac{1}{T}$

By power series expansion, $e^{-\tau s} \approx 1/(1+\tau_1 s) = \frac{\frac{1}{\tau_1}}{s+\frac{1}{\tau_1}}$

Now the transfer function becomes, $G_p(s) = \frac{\Delta V_1(s)}{\Delta U_1(s)}$

$$G_p(s) = \frac{K_1\frac{1}{\tau_1}}{(s+K_2)(s+K_3)(s+\frac{1}{\tau_1})} \tag{3.12}$$

We have chosen the operating point as $v = 30km/hr$ from the steady state conditions. The values obtained for the state space matrices and transfer function of the system as per the values chosen by Table 3.1 are as follows:

$A_1 = \begin{bmatrix} -0.0476 & 0.00067 \\ 0 & -1 \end{bmatrix}$, $B_1 = \begin{bmatrix} 0 \\ 743 \end{bmatrix}$ and $C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$

$$G_p(s) = \frac{\Delta V_1(s)}{\Delta U_1(s)} = \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \tag{3.13}$$

Table 3.1 Parameters chosen for Automobile[211]

Symbol	Values	Units
C	743	Unitless
C_{aero}	1.19	N/(m/sec^2)
m_v	1500	kg
τ	0.20	sec
T	1.0	sec
F_{dmax}	3500	N
F_{dmin}	-3500	N
G	9.8	m/sec^2

3.3 Mathematical Modelling of an Artificial Respiratory System

The schematic diagram of a typical respiratory system is shown in Figure 3.3. There is a centrifugal blower system that compresses the ambient air which is then used to ventilate the patient. A hollow tube is used to connect the blower to the patient. The output flow Q_{output} runs through the blower to the patient through a hollow tube. The process of exhalation is done in two parts, one part of exhalation is done through the centrifugal blower and the other part through the leak outlet connected to the patient. The provision of leak outlet is provided to fill the hollow tube with fresh air so that the chance of self-exhaled.

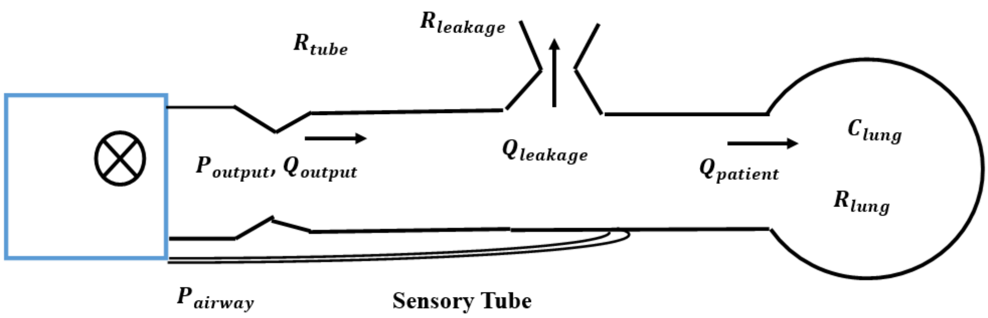

Figure 3.3 Schematic of the artificial respiratory system[105]

The relation between different flows can be expressed as:

$$Q_{patient} = Q_{output} - Q_{leakage} \tag{3.14}$$

The output pressure from the blower is expressed as $P_{output,}$, P_{airway} is the airway pressure at the mouth of the patient and is less than $P_{output,}$ due to the resistance R_{tube} offered by the tube. We can control the airway pressure. The actual pressure developed in the lungs is expressed as P_{lung}, which cannot be measured. The resistances $R_{lung}, R_{leakage}$ and R_{tube} are assumed to be linear. The equations can be expressed as:

$$Q_{output} = \frac{P_{output} - P_{airway}}{R_{tube}}$$

$$Q_{leakage} = \frac{P_{airway}}{R_{leakage}}$$

$$Q_{patient} = \frac{P_{airway} - P_{lung}}{R_{lung}} \tag{3.15}$$

The lung pressure is expressed by the differential equation:

$$\frac{dP_{lung}}{dt} = \frac{1}{C_{lung}} Q_{patient} \tag{3.16}$$

Where, C_{lung} is the compliance offered by the lungs. Now the dynamic equations of the lungs can be written as:

$$\dot{P}_{lung} = \frac{P_{airway} - P_{lung}}{C_{lung} R_{lung}} \tag{3.17}$$

This is an RC network; the lung pressure will increase according to the time constant $C_{lung} R_{lung}$

$$P_{airway} = \frac{\frac{1}{R_{lung}} \dot{P}_{lung} + \frac{1}{R_{tube}} P_{output}}{\frac{1}{R_{lung}} + \frac{1}{R_{tube}} + \frac{1}{R_{leakage}}} \tag{3.18}$$

Now substituting the value of airway pressure in the lung dynamics we get,

$$\dot{P}_{lung} = \frac{-\left(\frac{1}{R_{lung}} + \frac{1}{R_{leakage}}\right) P_{lung} + \frac{1}{R_{tube}} P_{output}}{C_{lung} R_{lung} \left(\frac{1}{R_{lung}} + \frac{1}{R_{tube}} + \frac{1}{R_{leakage}}\right)} \tag{3.19}$$

Converting Equations (3.16), (3.18), and (3.19) into a state space model. The lung pressure P_{lung} is considered as state variable, P_{output} as the input variable and $\begin{bmatrix} P_{airway} \\ Q_{patient} \end{bmatrix}$ as the outputs.

$$\dot{P}_{lung} = AP_{lung} + BP_{output} \tag{3.20}$$

$$\begin{bmatrix} P_{airway} \\ Q_{patient} \end{bmatrix} = CP_{lung} + DP_{output} \tag{3.21}$$

Where,

$$A = -\frac{\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)R_{lung}C_{lung}}$$

$$B = \frac{\frac{1}{R_{tube}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)R_{lung}C_{lung}}$$

$$C = \begin{bmatrix} \frac{\frac{1}{R_{lung}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)} \\ \frac{\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)R_{lung}} \end{bmatrix}$$

$$D = \begin{bmatrix} \frac{\frac{1}{R_{tube}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)} \\ \frac{\frac{1}{R_{tube}}}{\left(\frac{1}{R_{lung}}+\frac{1}{R_{tube}}+\frac{1}{R_{leakage}}\right)R_{lung}} \end{bmatrix} \tag{3.22}$$

Therefore, we can derive the transfer function by the equation,

$$H_t(s) = C[SI - A]^{-1}B + D \tag{3.23}$$

To determine the respiratory system's transfer function, we must model the blower system. The blower system used generally is a DC motor. Therefore, we can assume it is a 2nd order control system. The equation of a general 2nd-order system is expressed as:

$$B_l(s) = \frac{P_{output}(s)}{P_{control}(s)} = \frac{w_n^2}{s^2+2\varepsilon w_n s+w_n^2} \tag{3.24}$$

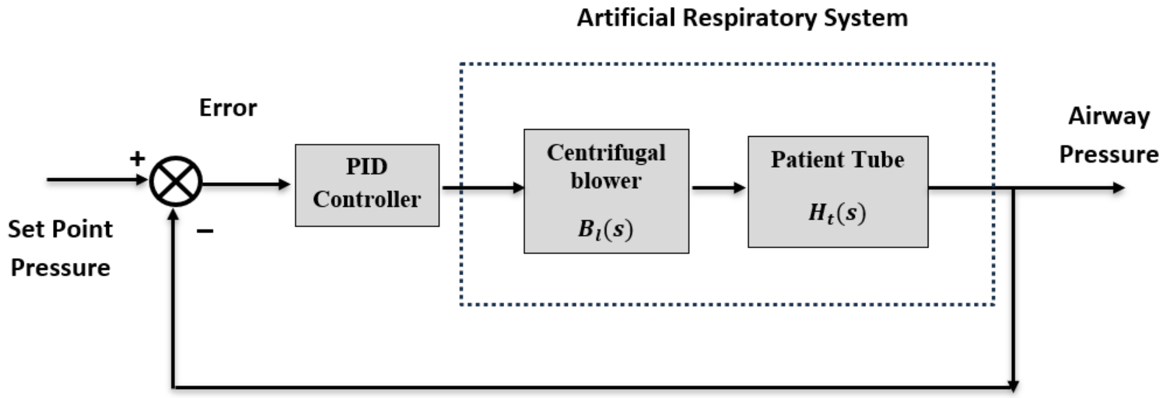

Figure 3.4 Block Diagram of Artificial Respiratory System

For this study the values are chosen by a particular experimental blower system, w_n is assumed to be 60π [105] and the damping ratio is unity. The overall transfer function P(s) is the cascade combination of $H_t(s)$ and $B_l(s)$. The block diagram representation of Artificial respiratory system is shown in Figure 3.5. The state space representation can be expressed as,

$$\dot{x_r}(\mathrm{t}) = \begin{bmatrix} \dot{x}_{bl} \\ \dot{P}_{lung} \end{bmatrix} = \begin{bmatrix} A_{bl} & 0 \\ BC_{bl} & A \end{bmatrix} \begin{bmatrix} X_{bl} \\ P_{lung} \end{bmatrix} + \begin{bmatrix} B_{bl} \\ 0 \end{bmatrix} P_{control}$$

$$y(t) = \begin{bmatrix} P_{airway} \\ Q_{patient} \end{bmatrix} = \begin{bmatrix} DC_{bl} & C \end{bmatrix} \begin{bmatrix} x_{bl} \\ P_{lung} \end{bmatrix} \tag{3.25}$$

3.4 Conventional PID Tuning

In this chapter, the Zeigler-Nichols method is used as a conventional method to determine the K_p, T_i, and T_d based on the open loop step response of the system. In this study parameters are determined from the open loop response of the system. If τ is the time-constant of the system and L is the dead time. The Z-N tuning rules are given in Table 3.2

Table 3.2 ZN Method of PID parameter Determination [115]

Controller	K_p	T_i	T_d
P	τ/L	--	----
PI	$0.9\tau/L$	$L/0.3$	----
PID	$1.2\tau/L$	$2L$	$0.5L$

3.5 Optimization Algorithms Used for Tuning PID Controller

3.5.1 Particle Swarm Optimization

PSO is an evolutionary optimization technique that is inspired by the action of food search by birds flocks or fishes[78] .PSO is one of the most popular algorithms among researchers and is applied in various areas because it is simple and efficient. It has been applied to solve a variety of optimization problems such as PID tuning, energy forecasting identification of parameters, etc. The PSO is comprised of a swarm of moving particles in a D-dimensional search space where a certain fitness measure is being improved. The position of each particle is represented by the position vector and a velocity vector. In the particle swarm optimization, the bird is known as a particle. Each particle is considered a solution candidate for the problem. The particles adjust their positions in the entire search space. In the entire process, each particle changes its location by its past best position called $\mathrm{P_{best}}$ and global best position which is $\mathrm{Gb_{best}}$.

After each iteration, the velocity and location of the particle are modified according to the equations:

$$\mathrm{v_{it}^{k+1} = wv_{it}^{k} + c_1 rand_1 \left(P_{bestit} - x_{it}^{k}\right) + c_2 rand_2 (Gb_{bestit} - x_{it}^{k})} \tag{3.26}$$

$$\mathrm{x_{it}^{k+1} = x_{it}^{k} + v_{it}^{k}} \tag{3.27}$$

Where it is the particle, k is the discrete-time $\mathrm{x_{it}}$ is the position of the ith particle, $\mathrm{v_{it}}$ is the velocity of ith particle, $\mathrm{w_i}$ is the inertia factor, $\mathrm{c_1}$ and $\mathrm{c_2}$ are the acceleration constants, $\mathrm{rand_1}$ and $\mathrm{rand_2}$ is a random number between 0 and 1, $\mathrm{P_{bestit}}$ is the best position found locally by the particle and $\mathrm{Gb_{bestit}}$ is the best position found globally by the particle. The procedure of the PSO algorithm is explained in the flowchart given below in Figure 3.5:

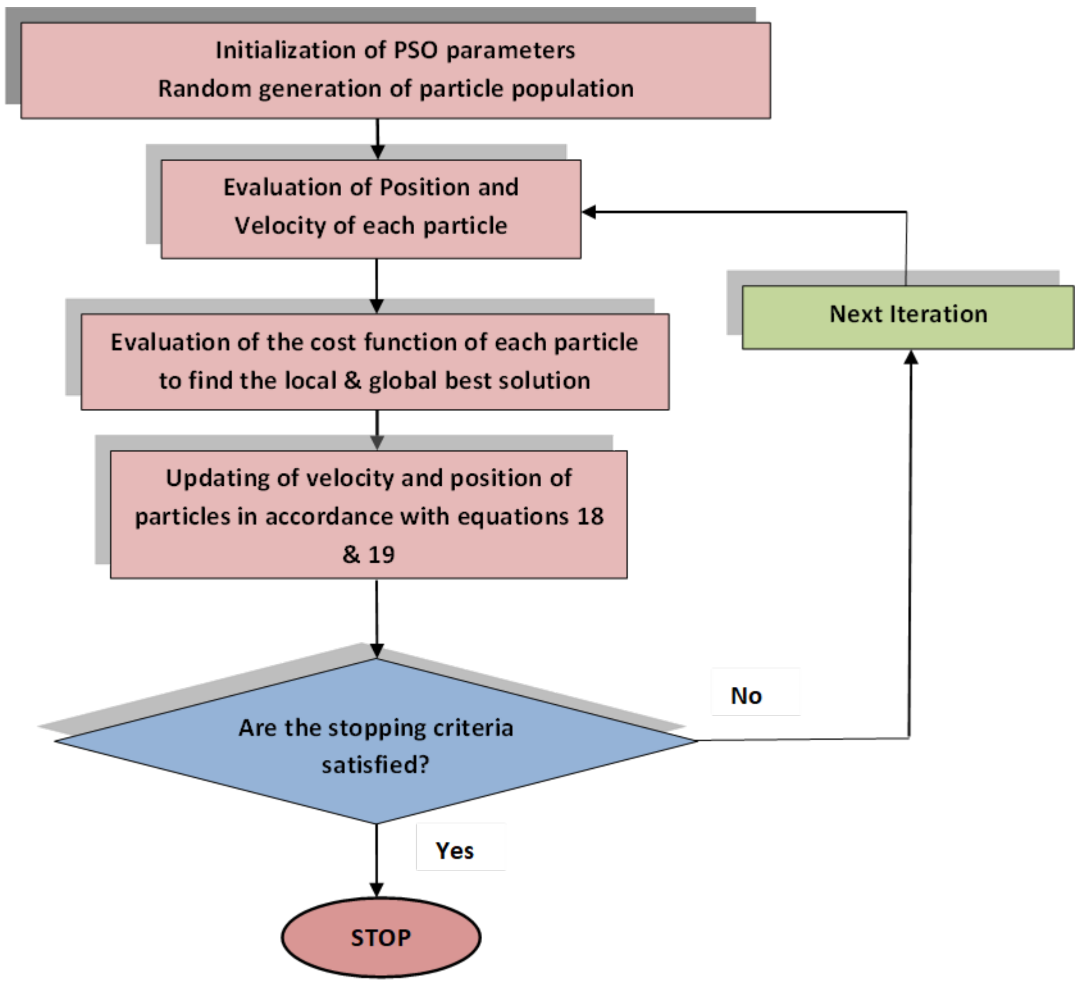

Figure 3.5 Flowchart of PSO

3.5.2 Teaching-Learning-based Optimization

Teaching learning-based optimization is also a nature-inspired algorithm. In this algorithm population is the group learners. This algorithm works on two phases of learning:

i. Learning through the teacher
ii. Learning through other learners

The best solution is a teacher[222]

Phase 1: Learning through teacher

The teacher puts maximum effort into increasing the mean of the subjects by the equation:

$$Mean_{j,i} = R_i(X_{j,kbest,i} - t_f M_{j,i}) \tag{3.28}$$

Where, $X_{j,kbest,i}$ is the best learner in a particular subject j, t_f is the factor according to which the value of mean is changed. R_i is the value between 0 and 1.

Teaching factor t_f is selected as:

$$t_f = round[1 + rand(0,1)[2 - 1]] \quad (3.29)$$

where, $rand$ is the random number in the range (0,1). According to the following equation:

$$X_{j,k,i}(t + 1) = X_{j,k,i}(t) + Mean_{j,i} \quad (3.30)$$

Phase 2: Learning through Other Learners

In this phase, learning is done by interaction in their peer group. The learning in this phase is expressed as:

Two learners are selected randomly like A and B

$$X(t + 1)_{total_A,i} \neq X(t + 1)_{total_{B.i}} \quad (3.31)$$

Where, $X(t + 1)_{total_A,i \;\&\; X(t+1)_{total_B,i}}$ are the updated values at the end of phase one.

For minimization problem,

$$X_{j,A,i}(t + 2) = X_{j,A,i}(t + 1) + r_i(X_{j,A,i}(t + 1) - X_{j,B,i}(t + 1)$$

$$if\ X_{total\_A,i} < X_{total\_B,i} \quad (3.32)$$

$$X_{j,A,i}(t + 2) = X_{j,A,i}(t + 1) + r_i(X_{j,B,i}(t + 1) - X_{j,A,i}(t + 1)$$

$$if\ X_{total\_B,i} < X_{total\_A,i} \quad (3.33)$$

The flowchart of the teaching learning-based optimization algorithm is shown in Figure 3.6

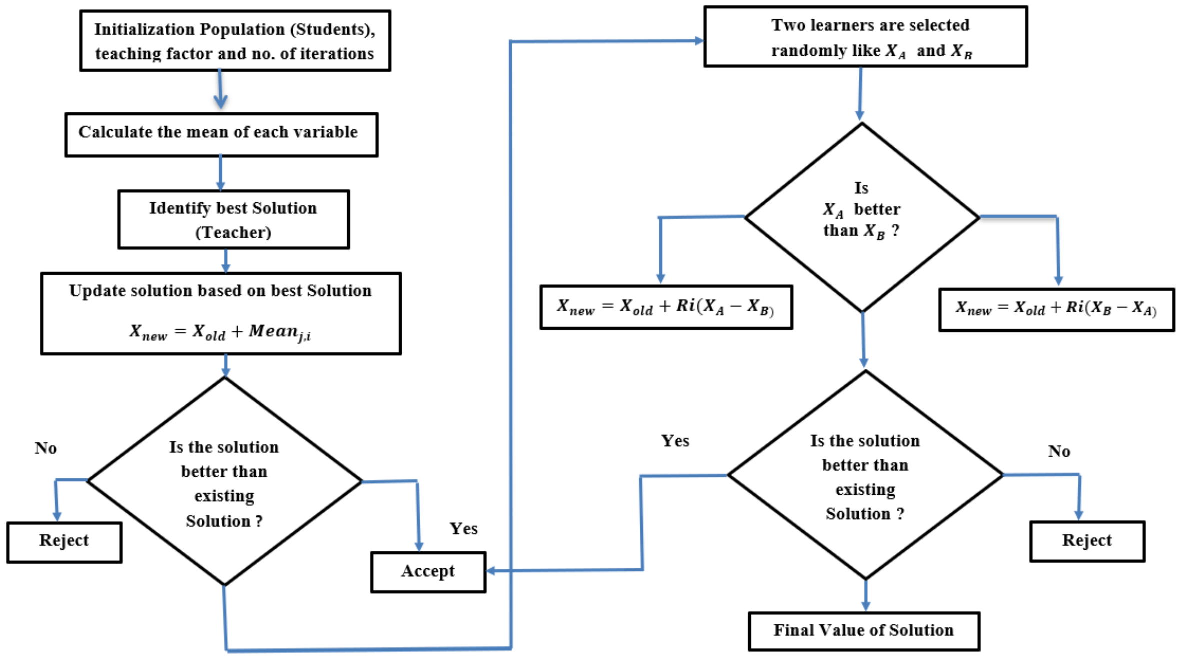

Figure 3.6 Flow Chart of TLBO Algorithm

3.5.3 Objective Functions Selected For PID Tuning

Three objective functions are selected for the tuning of the PID controller and their performances are compared. The three objective functions selected are as follows:

Integral time absolute error (ITAE)[223]:

This error is expressed mathematically as:

$$\text{ITAE} = \int t|e(t)|dt \tag{3.34}$$

Integral absolute error (IAE)[223]:

$$\text{ITE}=\int te(t)dt \tag{3.35}$$

Integral square error (ISE)[223]:

$$\text{ISE}=\int |e(t)|^2 dt \tag{3.36}$$

The three objective functions minimize different types of errors in the system. The integral time error reduces the error occurring in the initial instants of time. It is useful in minimizing the early transient errors. The integral square error reduces both large errors and smaller errors irrespective of the time of error. The ITAE reduces the errors which persists for a longer duration of time.

3.6 Implementation and Analysis

3.6.1 PSO-based Controller Design of Automobile Cruise Control System

The block diagram representation of the Particle Swarm Optimisation tuned Automobile cruise control system is given in Figure 3.7. The K_p, K_d, and K_i controller gains are optimized by the PSO algorithm. It compared with the conventional tuning methods like the Ziegler Nichols method and fuzzy logic. The tuning of PSO parameters is done by hit and trail method. The number of populations was first considered as 20 for which the objective function was not converging. Then, the number of populations was taken as 35 for which the objective function was converging but the value was not satisfactory. For the number of populations 50, the results gave the satisfactory minimum value of the objective function. The inertia factor is selected as 0.9 which promotes more exploration. The value of C_1 and C_2 is usually kept at 1,1.5 or 2. In this case the value of C_1 and C_2 is selected as 2 and 1.5 respectively. The parameters considered for PSO optimization are shown in the Table 3.3 below:

Table 3.2 Parameters Selected for PSO

Parameters	**Values Chosen**
No. of Populations	50
Inertia Factor	0.9
Acceleration Factors	1.5,2

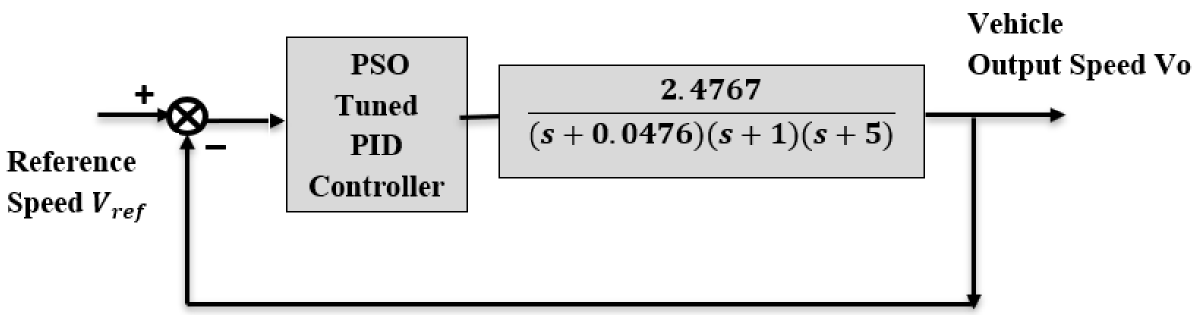

Figure 3.7 Block diagram of PSO-tuned PID Controller for Automobile Cruise Control System

3.6.2 TLBO-based Controller Design of Automobile Cruise Control System

The block diagram representation of the tuned PID controller is shown in Figure 3.8. The K_p, K_d, and K_i parameters of the PID controller are optimized by the TLBO algorithm. To select the TLBO parameters first a small number of populations as 10 was considered since the results were not desirable. The algorithm was run for a higher number of populations 20, 30 ,40 and 50. The maximum number of iterations selected in this problem is 25. For this value, the results are converging and the computational time is also low. To choose the appropriate teaching factor experiments were done selecting the teaching factor as 1, 1.5, and 2. The algorithm gave satisfactory converging results for teaching factor equal to 1. The TLBO parameters chosen are as given in Table 3.4 follows:

Table 3.3 Parameters for TLBO

Parameters	Values chosen
No. of Populations	50
Maximum iterations	25
Teaching factor	1

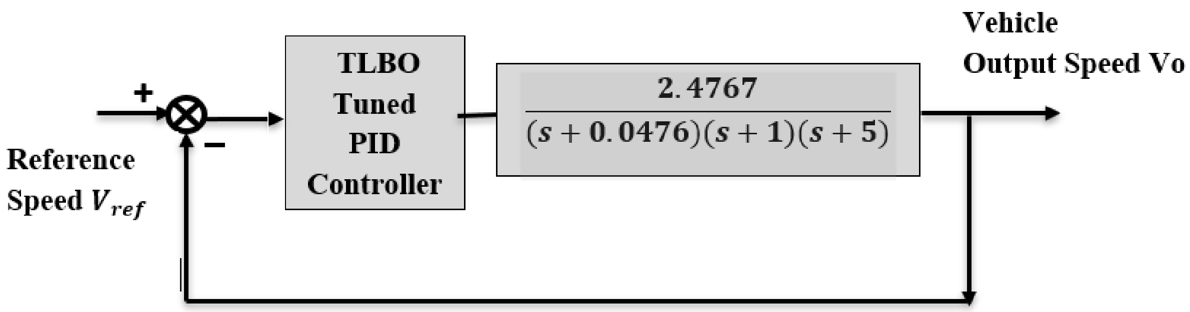

Figure 3.8 Block Diagram of TLBO tuned PID controller for Artificial cruise control system

3.6.3 PSO based Controller Design of Artificial Respiratory System

The optimization of different error functions ITAE, ISE, and ITE is done using the PSO algorithm, and PID parameter values are obtained to get a better response.

The block diagram of the PSO-based PID optimization scheme is shown in Figure 3.9.

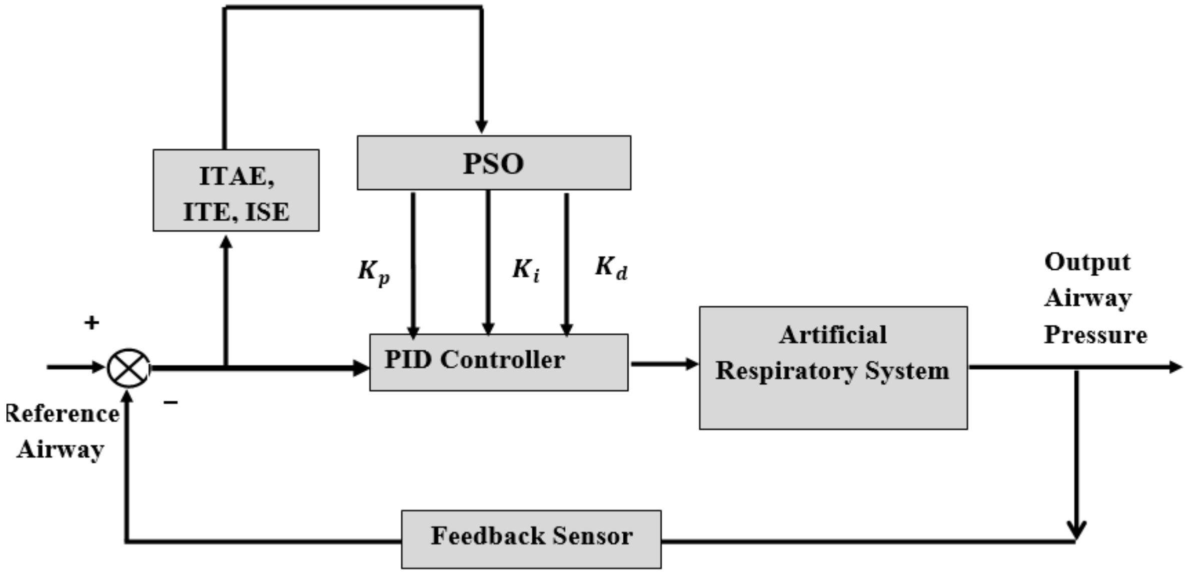

Figure 3.9 Block diagram of PSO tuned PID controller for Artificial respiratory system

The PSO algorithm parameters are selected based on hit and trail. The algorithm was run with varying parameters starting from number of population equal to 10 and gradually increased. The best results were obtained for a number of iterations of 50, the inertia factor of 0.9, and acceleration factors of 1.5 and 2.

3.6.4 TLBO-based Controller Design of Artificial Respiratory System

The optimization of different error functions ITAE, ISE, and ITE is done using the TLBO algorithm, and PID parameter values are obtained to get a better response. The block diagram of the TLBO-based PID optimization scheme is shown in Figure 3.10. The parameters chosen for the TLBO Algorithm are initial populations as 50, the number of iterations as 25, and the teaching factor as 1.

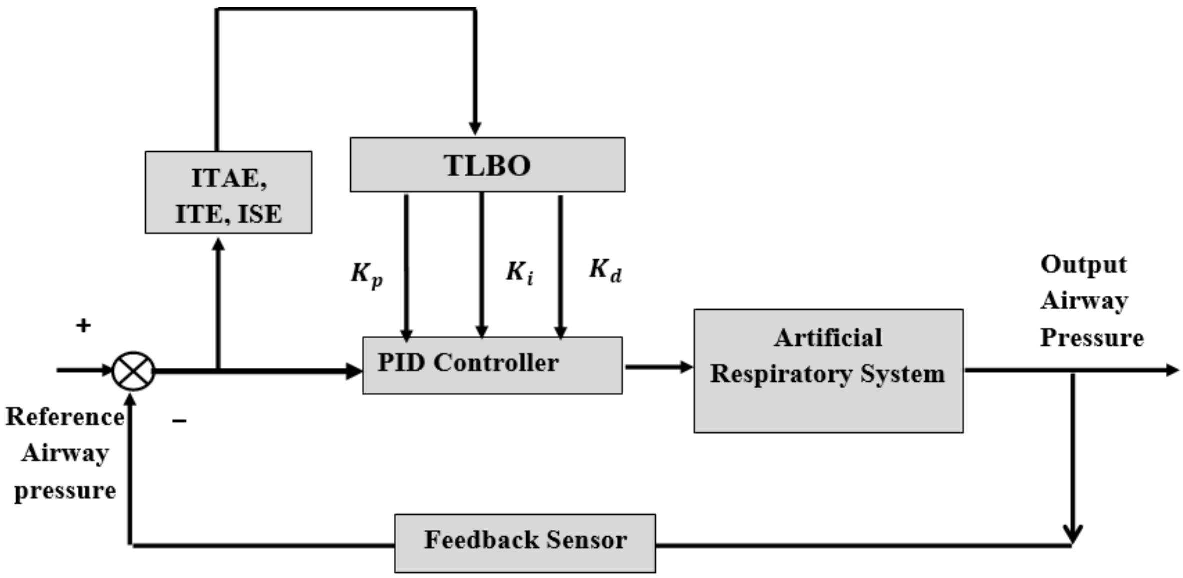

Figure 3.10 Block diagram of TLBO tuned PID controller for Artificial respiratory system

3.7 Results and Discussions

3.7.1 Output Results of Automobile Cruise Control System

In this thesis, the above-developed controllers are applied and evaluated on the artificial cruise control system using MATLAB 2017a software. The objective functions used for optimizing PID gains are ITAE, ITE, and ISE. TLBO and PSO algorithms are used for obtaining optimized PID gains. A comparative analysis is presented by comparing the intelligent controllers with the conventional Zeigler Nichols tuned and Fuzzy PD + I controller. The step responses of different controllers are given in Figures 3.11, 3.12 and 3.13.

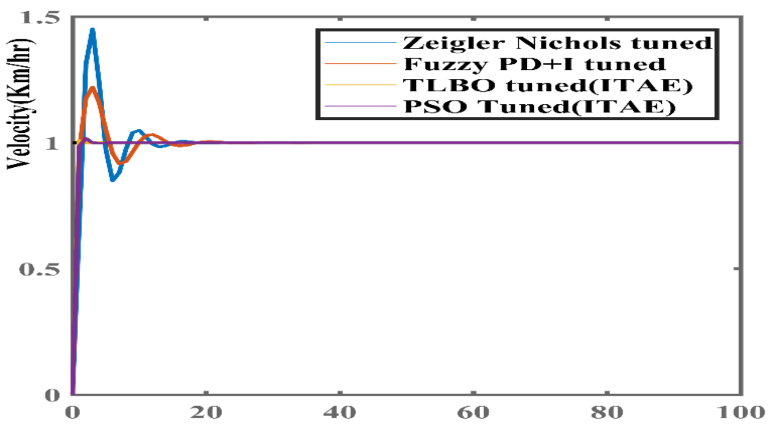

Figure 3.11 Comparative Step responses of Different controllers using ITAE as an Objective Function

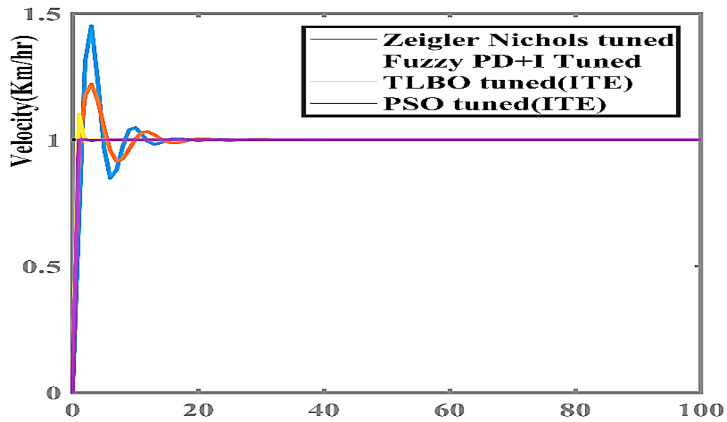

Figure 3.12 Comparative Step response of Different controllers with ITE as an Objective Function

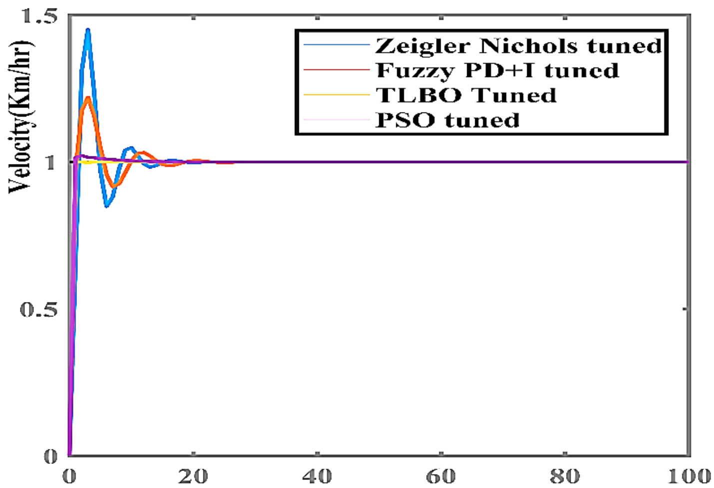

Figure 3.13 Comparative responses of different controllers with ISE as an objective Function

The comparison of different controllers based on various transient response parameters is as follows:

Table 3.5 Comparison of transient response parameters of the proposed intelligent techniques with conventional methods (ITAE as an objective function)

Method	**Rise time t_r (seconds)**	**Settling time t_s (seconds)**	**Max. Overshoot (% M_P)**	**Peak time t_p, (seconds)**
ZN	1.0096	11.07	46.10	2.78
Fuzzy PD+I	0.7130	13.05	22.17	2.65
PSO ITAE	0.6150	3.46	0	1.28
TLBO ITAE	**0.3145**	**1.57**	**3.43**	**1.25**

Table 3.6 Comparison of transient response parameters of the proposed intelligent techniques with conventional methods (ISE as an objective function)

Method	**Rise time t_r (seconds)**	**Settling time t_s (seconds)**	**Max. Overshoot ($\%M_P$)**	**Peak time t_p, (seconds)**
ZN	1.0096	11.07	46.10	2.78
Fuzzy PD+I	0.7130	13.05	22.17	2.65
PSO ISE	0.2712	2.05	11.04	0.61
TLBO ISE	**0.6280**	**1.54**	**2.96**	**1.28**

Table 3.7 Comparison of transient response parameters of the proposed intelligent techniques with conventional methods (ITE as an objective function)

Method	**Rise time t_r (seconds)**	**Settling time t_s (seconds)**	**Max. Overshoot ($\%M_P$)**	**Peak time t_p, (seconds)**
ZN	1.0096	11.07	46.10	2.78
Fuzzy PD+I	0.7130	13.05	22.17	2.65
PSO ITE	0.2818	1.52	17.77	0.62
TLBO ITE	**0.4210**	**1.39**	**10.96**	**0.90**

The comparative analysis of all the tuning methods shows that the TLBO-tuned controller gives a better response as compared to all the above methods used. By comparing various transient response performance indices, the following observations were made:

1. From Table 3.5 we can observe that when we used ITAE as an objective function, the percentage overshoot was very high up to 46% in the ZN method and 22% in the fuzzy PD+I controller. In the PSO tuned controller the overshoot is reduced to 0% but the response time is greater as compared to the tuned controller. In the TLBO-tuned controller, there is a small overshoot of 3.43% but simultaneously the rise time, settling time, and peak time are reduced which makes the response fast.
2. From Table 3.6 we can observe that when we used ISE as an objective function, the overshoot reduced from 46% to 22% in the fuzzy PD +I controller. In PSO tuned controller

the overshoot was reduced to 11% and the response time was also reduced to 0.2712 seconds. In the tuned controller the overshoot was further reduced to 2.96% but the rise time was more than PSO tuned controller at 0.6280 seconds.

3. From Table 3.7 we can observe that when we used ITE as the objective function the overshoot reduced to 17% in PSO tuned controller and 10% in the tuned controller. The rise time was 0.28 seconds in PSO tuned controller and 0.42 seconds in TLBO tuned controller.

From the above comparative analysis, it is evident that controllers tuned with optimization algorithms, PSO, and TLBO gave better responses as compared to the conventional tuning methods.

3.7.2 Robust Analysis

The control system is said to be robust if it can maintain a stable and satisfactory output performance even in the presence of external disturbances, system's parameter variations and uncertainty. Therefore, to test the effectiveness and validity of the proposed controller it is crucial to test the robustness. To test the effectiveness of the proposed controller it is important to test whether the controller is robust in the presence of uncertainties and disturbances. The parameter C_a, air drag coefficient is varied from -50% to +50% nominal range. The step responses in the variation are presented in Figure 3.14.

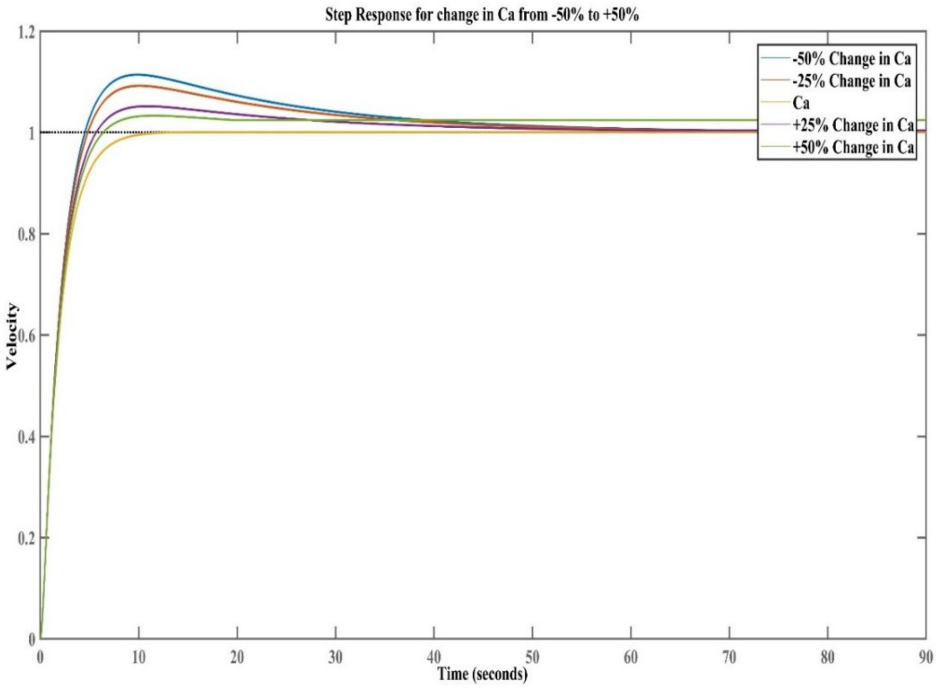

Figure 3.14 Robust Analysis with variation in Ca

The comparison of performance indices for the robust analysis with variation in Ca are given in Table 3.8

Table 3.8 Comparison of Performance indices with variation in Ca

Parameter	**Range of Change %**	**Rise time $t_r(seconds)$**	**Settling Time, $t_s(seconds)$**	**Peak Time, $t_p(seconds)$**
C_a	-50%	1.8447	0.8036	1.7638
	-25%	1.4562	0.8236	1.3256
	+25%	1.6752	0.8136	1.4562
	+50%	1.5789	0.8065	1.4558

3.7.3 Output Results of Artificial Respiratory System

The optimized controller's performance is evaluated by simulating the model of the artificial respiratory system derived in MATLAB software. The performance indices used for simulation are ITAE, ISE, and ITE. The optimized PID gains are obtained by using the nature-inspired optimization techniques PSO and TLBO. For this simulation study, the set point pressure $P_{set} = 1mbar$. The system's parameters selected are given in Table 3.9:

Table 3.9 Values of Parameter chosen

Parameter	Values
R_{lung}	$5 \times 10^{-3}\ mbar/mL/s$
$R_{leakage}$	$6 \times 10^{-2}\ mbar/mL/s$
R_{tube}	$45 \times 10^{-4}\ mbar/mL/s$
C_{lung}	$20\ \ mL/mbar$

The optimized response of the output airway pressure-tuned tuned with the Zeigler-Nichols method is shown in Figure 3.15. The tuned PID gain values are K_p=0.00035, K_i=1.75, and K_d=0. The step response tuned by Zeigler Nichols method has a rise time equal to 8.2015 seconds, and a settling time is approximately 15.4388 seconds, which shows a slow response.

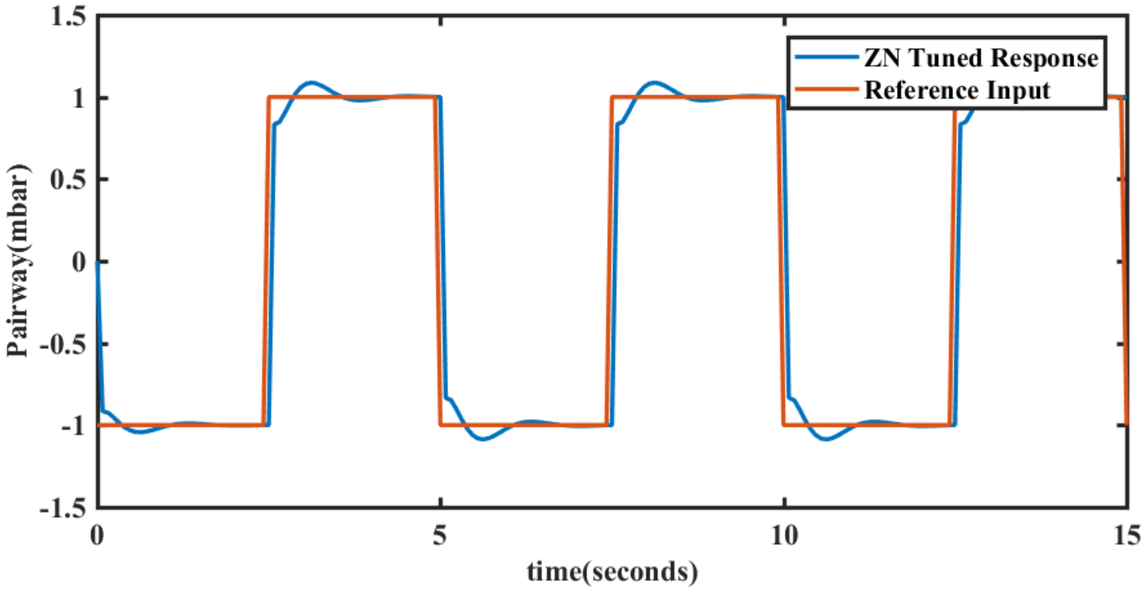

Figure 3.15 ZN Tuned Output Response of Airway Pressure

The comparative step response of different controllers for output airway pressure P_{output} of an artificial ventilation system are given below in Figures 3.16, 3.17, and 3.18. The objective functions chosen are ITAE, ITE, and ISE.

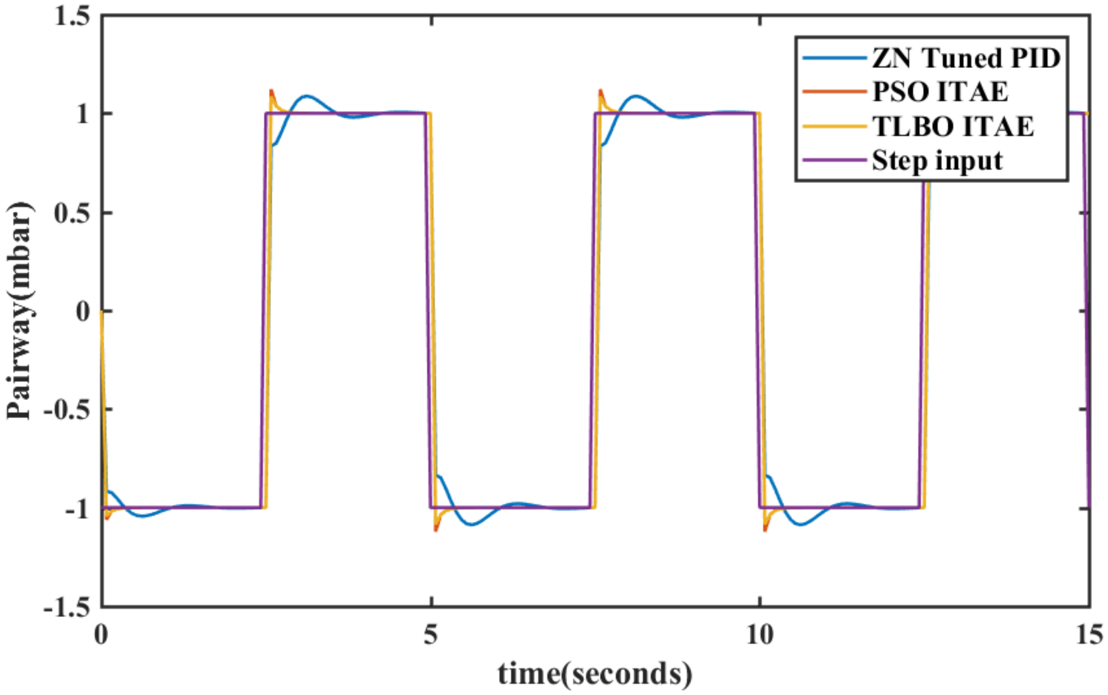

Figure 3.16 Comparative responses of controllers with ITAE as an objective function

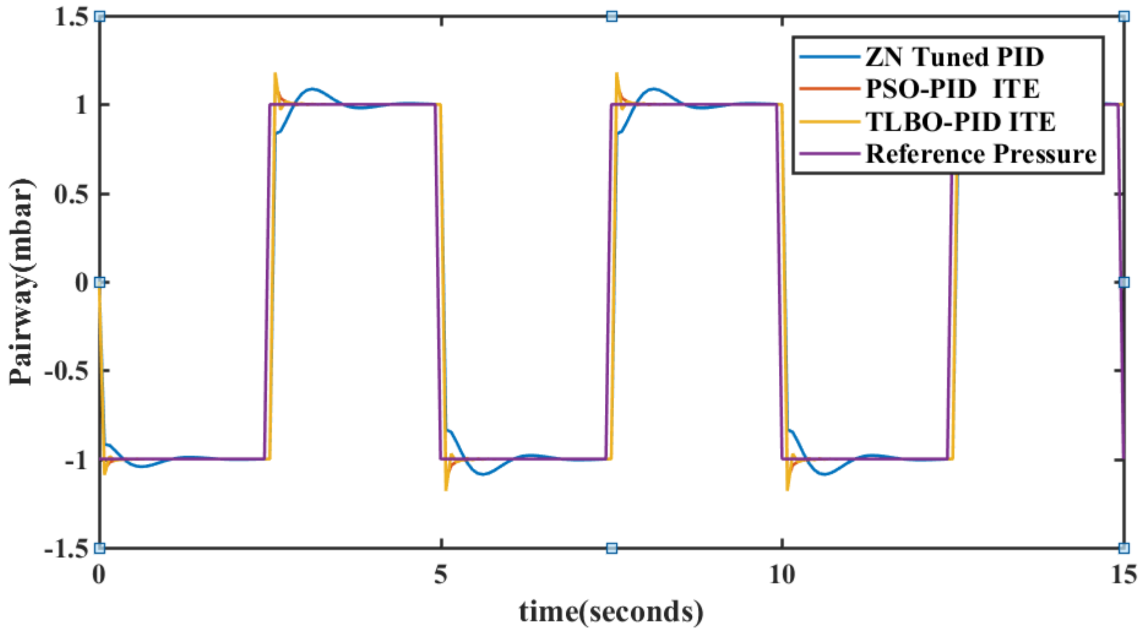

Figure 3.17 Comparative response of different controllers with ITE as an objective function

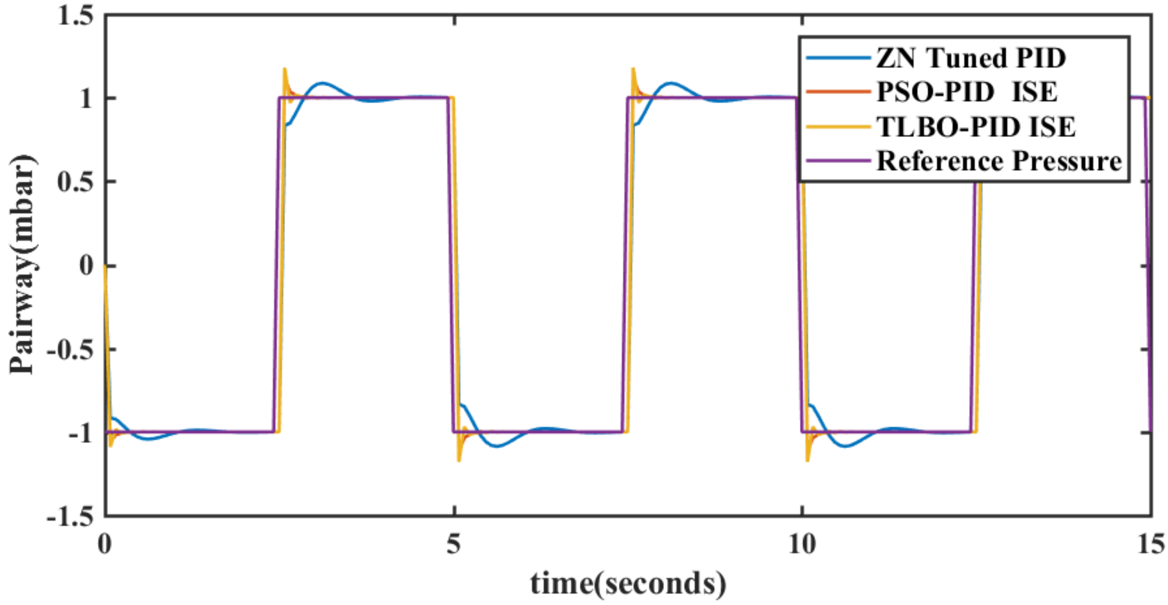

Figure 3.18 Comparative response of different controllers with ISE as an objective function

The comparative analysis of the transient response characteristics of the proposed controllers with conventional controllers is presented in Table 3.8. The performance indices compare the rise time, settling time and overshoot of the output.

Table 3.9 Comparative Analysis of Different Controllers

Sno.	Method used	Objective Function used	Rise time (in secs)	Overshoot % M_P	Peak time (in secs)	Settling time (in secs)
1.	Zeigler Nichols		8.2015	0.373	32.59	15.43
2.	PSO	ITAE	0.2546	0	0.5624	0.7885
		ITE	0.2529	0	0.5164	0.8823
		ISE	0.6950	0.0316	1.3408	2.0303
3.	TLBO	ITAE	0.2835	0	0.8854	0.5588
		ITE	**0.2577**	**0**	**0.8841**	**0.5516**
		ISE	0.1703	0.0316	1.1058	2.1762

The ZN-tuned PID response had a very large rise time equal to 8.2015, the overshoot was 0.373, and the peak time and settling time were 59 seconds and 15.43 seconds. The PSO-tuned response with ITAE and ITE as objective functions has a better response. The rise time was reduced to 0.25 seconds, overshoot reduced to 0 percent, and settling time reduced to 0.88 seconds. The TLBO-tuned response with ISE as the objective function gave rise time equal to 0.1703 seconds, which is less as compared to PSO tuned response. The Figures 3.19, 3.20 ,3.21 represent the convergence curves of the algorithms TLBO and PSO.

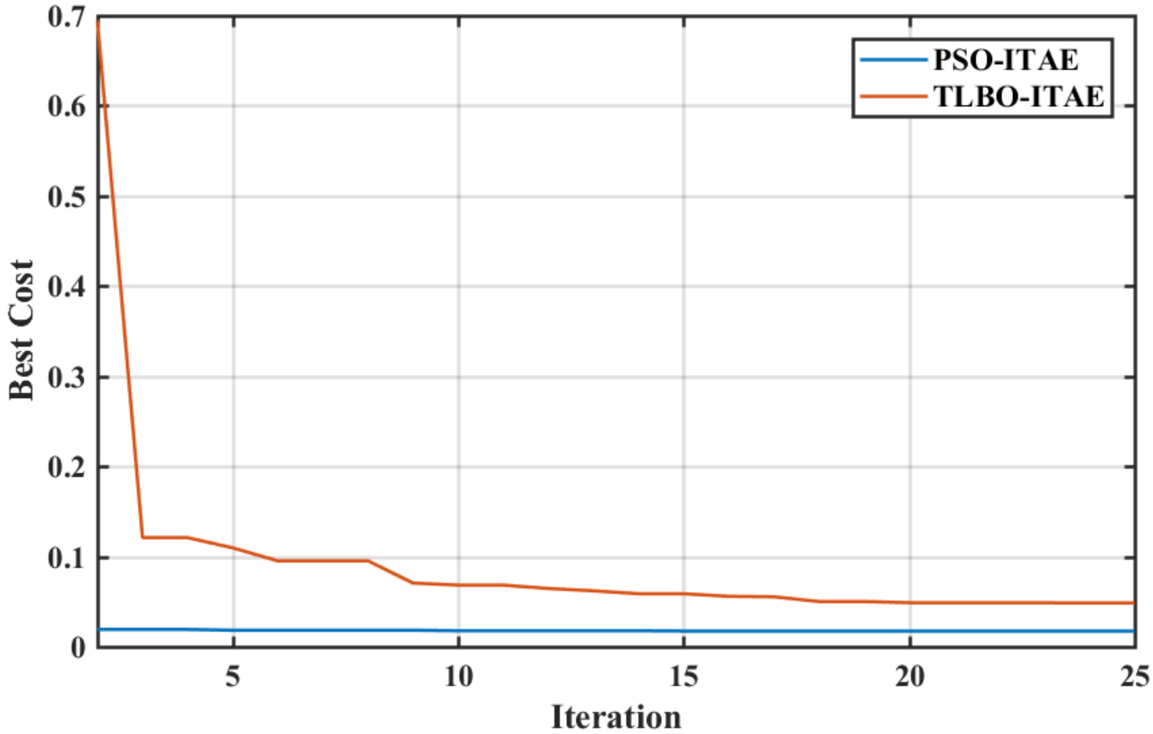

Figure 3.19 Convergence Curve of Algorithms with ITAE as an objective function

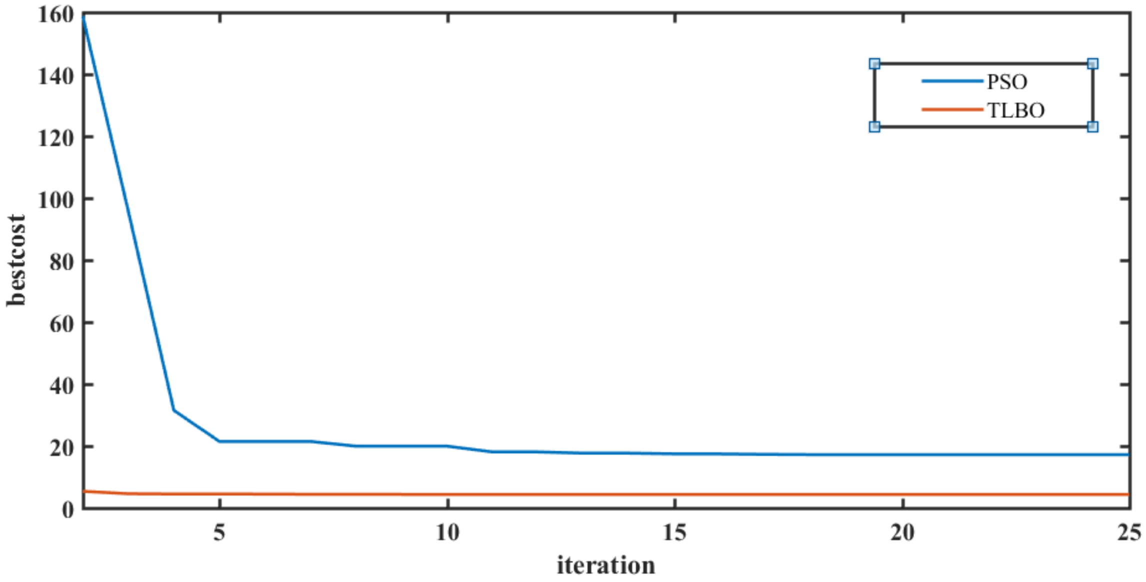

Figure 3.20 Convergence Curves of Algorithms with ITE as an objective function

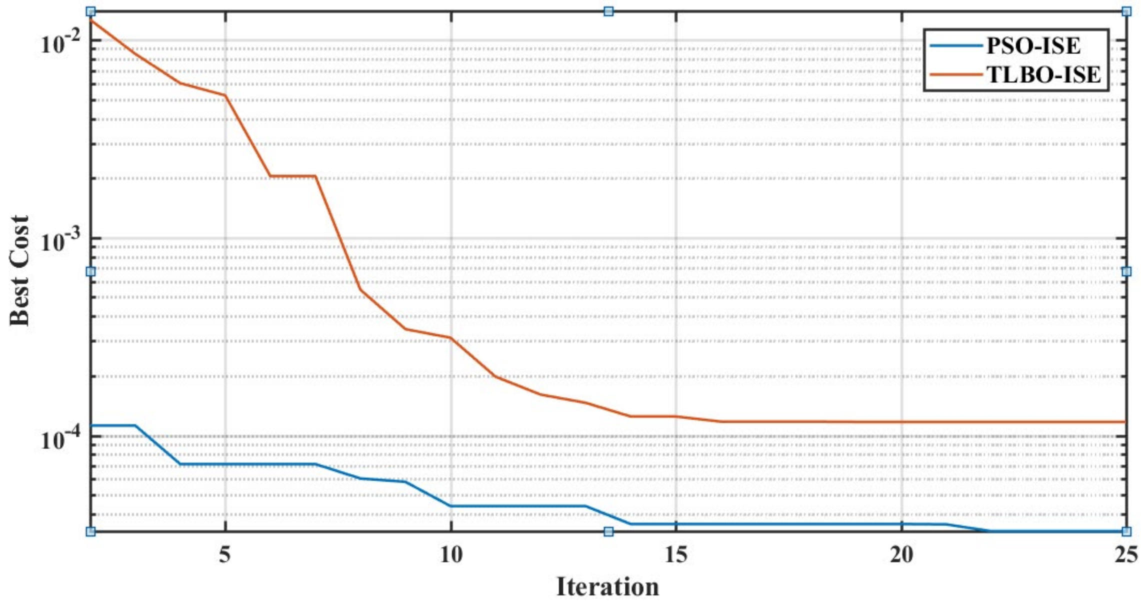

Figure 3.21 Convergence Curves of algorithms with ISE as an objective Function

3.8 Conclusion

In this chapter two benchmark non-linear systems are considered, automobile cruise control system and artificial respiratory system. The dynamics equations of the systems are modelled mathematically and their transfer functions are obtained for a stable operating point. A PID controller tuned by particle swarm optimization algorithm and teaching-learning based optimization algorithm is applied to control both the systems. The different objective functions chosen for the optimization are ITAE, ISE, and IAE. The step response is obtained and it is also compared with the conventionally used PID tuning methods like Zeigler Nichols and Fuzzy logic[177]. It is concluded that TLBO tuned PID controller gives a better response in comparison to all these controllers.

CHAPTER 4

IMPLEMENTATION OF CASCADE OPTIMIZED PID CONTROLLER ON THE NON-LINEAR BALL AND BEAM SYSTEM

4.1 Introduction

Cascade control is a powerful control method. However, the tuning of two controllers used in this technique is a complex problem. In a cascade PID control, there are two control loops, the inner loop, and the outer loop[224]. The cascade control technique is one of the most popular methods in the industry because it has many advantages as compared to other techniques. It is capable of rejecting disturbances in the inner loop. It increases the speed of the system and can overcome the effects of parameter variations in the inner loop. To obtain better control in non-linear systems such as ball and beam systems conventional PID controllers can be replaced by other control techniques like feedforward and cascade methods[223]. However, the conventional cascade PID controller can give unsatisfactory results when subjected to variations in set point and parameter changes. Therefore, there is a need to apply advanced AI methods which can adapt to non-linear changes in the system. Cascade PID controllers have a lot of advantages as compared to conventional PID methods but PID parameter tuning becomes a difficult task due to interrelated parameters. The conventional tuning methods like the Zeigler-Nichols method and, Cohen-coon method are based on trial-and-error methods[13]. Therefore, they become tedious and inefficient. Another method of tuning PID controllers applied to non-linear systems is by gain-scheduling [119]. But its major disadvantage is that the controller design is time-consuming and when the dynamics of the system are unknown it cannot be used[225].

Recently, many complex engineering problems have been solved by applying optimization algorithms efficiently. PID controllers applied to various nonlinear systems, have recently been tuned using a variety of bio-inspired algorithms such as Genetic Algorithm (GA)[88], [153], [162], [226], Particle swarm optimization (PSO)[64], [110], [133], [227], Ant bee colony (ABC)[33], [137], [192], [227], [228], and Ant colony optimization (ACO)[202], [229], [230]. A difficult issue with many variables and constraints can be solved almost perfectly using these bioinspired algorithms. One of the most popular algorithms used to date is GA but it is challenging to estimate population size, crossover rates, and mutation rates. The efficiency of an algorithm changes if the selected parameters are varied. Inertia weight, and social, and cognitive characteristics are also used by PSO. With minimal computational effort and excellent consistency, the Teaching Learning optimization (TLBO) [80], algorithm is used to find global solutions for continuous nonlinear functions. It produces superior outcomes because fewer parameters are needed to use it [81].

A ball and beam system, a benchmark non-linear control problem, is taken into consideration in this thesis. The goal is to maintain the ball and beam system's position in balance. Due to the presence of nonlinearity, such as dead zone, saturation, nonlinear resistance, etc., it is a challenging task[93]. The ball and beam control can be used to solve several practical issues, including robot weight balance, space vehicle control, and aircraft in space control. Researchers have suggested several control strategies for the ball and beam system, including fuzzy controllers, traditional PID controllers, and neural networks[93], [219], [229], [231]–[236]. The TLBO algorithm is used to adjust the cascade PID controllers in this study, and it is applied to a ball and beam system. Both the PID controllers are tuned by PSO and TLBO. The performance of the tuned controller is compared with the PSO-tuned PID controller and conventional Ziegler Nichols-tuned PID controllers. Teaching learning-based Optimization algorithm is applied to tune the parameters of the PID controller to control beam angle and ball position.

In this chapter, we have used the PSO algorithm and TLBO algorithm for tuning cascade PID controllers applied for the position control of the ball and beam system.

4.2 Mathematical Modelling of Ball and Beam System

A ball and beam system comprises of a long beam, a ball, and an electrical system that includes a servo motor. The ball's position is controlled by the servo motor. The force diagram of a ball and beam system is depicted in the Figure 4.1[22] below. By using a servo voltage, the control objective is to move the ball to the desired position. The physical system of the ball and beam system's transfer function is derived below.

The ball is subject to two forces: $F_{tx_0}(t)$, the force resulting from translational motion, and $F_{rx_0}(t)$, the force resulting from rotating motion[237].

$$F_{tx_0}(\mathrm{t}) = m_b g \sin \propto_0 (t) \tag{4.1}$$

$$F_{rx_0}(\mathrm{t}) = \frac{2}{5} m_b \ddot{x}_o(\mathrm{t}) \tag{4.2}$$

Where, m_b is the mass of the ball, g is the acceleration due to gravity & $\ddot{x_o}(t)$ is acceleration of ball

$$m_b\ddot{x}_o(\mathrm{t}) = \sum F = F_{tx_0}(\mathrm{t}) - F_{rx_0}(\mathrm{t}) = m_b g \sin \propto_0 (t) - \frac{2}{5} m_b \ddot{x}_o(\mathrm{t}) \qquad (4.3)$$

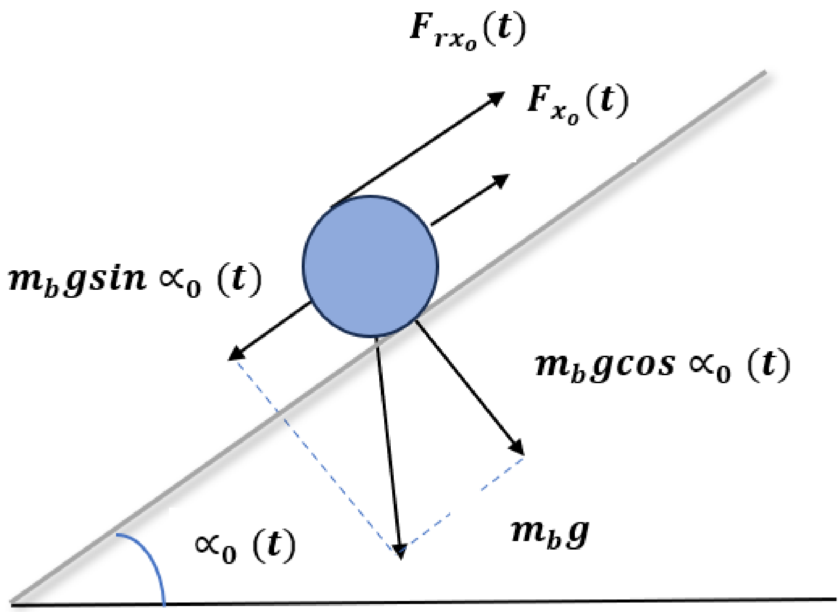

Figure 4.1 Force Diagram of Ball and Beam System

The moment of inertia of a sphere is $2/5m_b$

From (3) we get,

$$\ddot{x}_o(\mathrm{t}) = \frac{5}{7} g \sin \propto_0 (t) \qquad (4.4)$$

For small values of $\propto_0$ we can write $\sin \propto_0 (t) \approx\propto_0 (t)$

Taking Laplace transform,

$$s^2 X_0(s) = \frac{5g}{7s^2} \propto_0 (s) \qquad (4.5)$$

$$\frac{X(s)}{\propto_0(s)} = \frac{5g}{7s^2} \tag{4.6}$$

The transfer function of the servo motor relating V_{mt}, input motor voltage and θ_{mt}, the servo load gear angle is

$$\frac{\theta_{mt}(s)}{V_{mt}(s)} = \frac{\eta_g \eta_{mt} K_{to} K_g}{J_{eq} R_{mt} s^2 + (B_{eq} R_{mt} + \eta_g \eta_{mt} K_{to} K_g{}^2)s} \tag{4.7}$$

The transfer function relating to motor angle θ_{mt}, the servo load gear angle and beam angle $\propto_0$ is

$$\frac{\theta_{mt}}{\propto_0} = \frac{r}{L} \tag{4.8}$$

The overall transfer function relating the ball position $X_0(s)$ and motor input voltage $V_{mt}(s)$ is given as, $\frac{X_0(s)}{V_{mt}(s)} = \frac{\theta_{mt}(s)}{V_{mt}(s)} \times \frac{\propto_0(s)}{\theta_{mt}(s)} \times \frac{X_0(s)}{\propto_0(s)}$ (4.9)

Table no 4.1[238] **Definitions of the parameters and their values selected**

Symbol	**Description**	**Values**
K_t	Torque Constant of the motor	0.00767
K_{mt}	Constant of Back emf	0.00767
K_g	Gear ratio of the servo system	70
R_{mt}	Armature resistance (in ohms)	2.6
J_{eq}	Moment of Inertia(load)	2e-3
B_{eq}	Viscous damping Coefficient	4e-3
r	Lever arm offset(inches)	1
L	Length of beam(inches)	16.75
g	The gravitational constant of Earth (m/s^2)	9.8
η_g	Gearbox efficiency	0.9
η_{mt}	Motor efficiency	0.36

4.3 Cascade Controller Design

The ball and beam system is observed to be a fourth-order system. Consequently, creating a controller is challenging. As a result, a cascade control system is created for the entire system. There are two loops for feedback. A servo control loop controls the position of the motor gears in the inner loop. Ball location is controlled by the outer loop [8]. Figure 4.2 shows a ball and beam system under cascade control. PID 2 controls the ball and beam system, and PID 1 controls the motor.

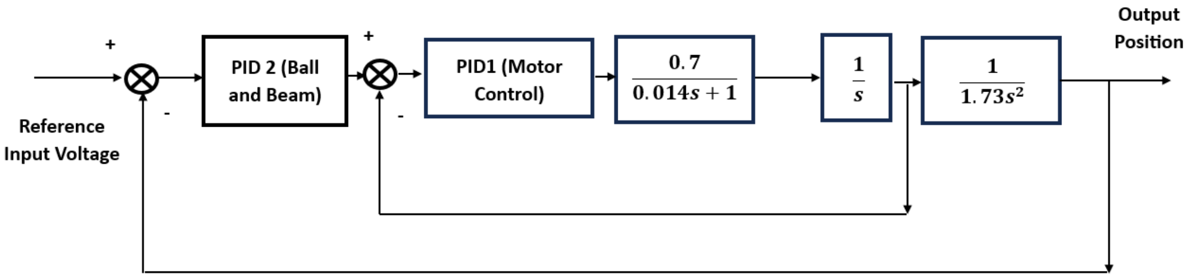

Figure 4.2 Cascade Control of Ball and Beam System

4.4 Proposed TLBO-tuned PID Cascade Design

In this study, a cascade PID controller structure is proposed which is used to control ball position and beam angle. There are two control loops: the outer loop for ball position and the inner loop for motor angle control. The two PID controllers are tuned using the TLBO algorithm and the optimized PID gains are used for controlling the ball position and motor angle. The block diagram of the TLBO-tuned PID control ball and beam system is shown in Figure 4.3. In Figure 4.3, it is shown that the PID controllers are tuned using the TLBO algorithm. The objective functions chosen for tuning the TLBO algorithm are ITAE, ISE, and ITE.

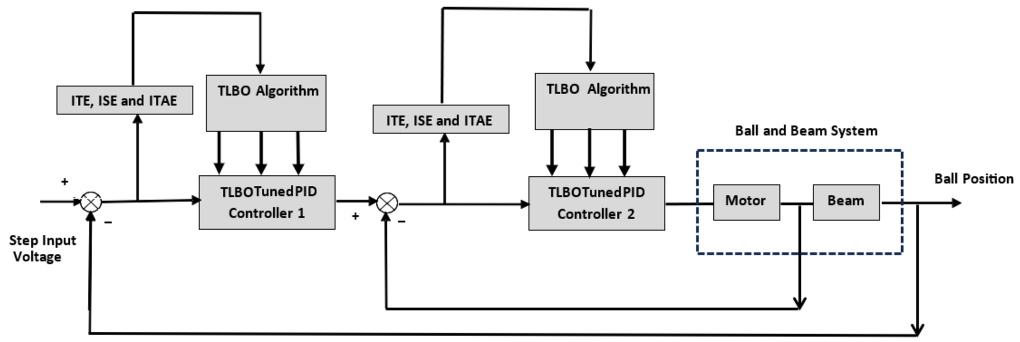

Figure 4.3 Block diagram of TLBO tuned Ball and beam system

The parameters used for Teaching Learning Based Algorithm, are the number of populations is 50, the maximum number of iterations is 25, and the teaching factor is chosen to be 1.

4.5 Proposed PSO-tuned Controller Design

A PSO-tuned cascade PID controller is proposed PID 1 is used to control ball position and PID 2 is used for servo motor control. There are two control loops one for ball position and the other for servo motor control. In the PSO-designed controllers, the two controllers are tuned using the PSO algorithm. The block diagram of PSO tuned cascade control ball and beam system is shown in Figure 4.4. In Figure 4.4 it is shown that the PID controllers are tuned using the PSO Algorithm. The objective functions used are ITAE, ISE, and ITE.

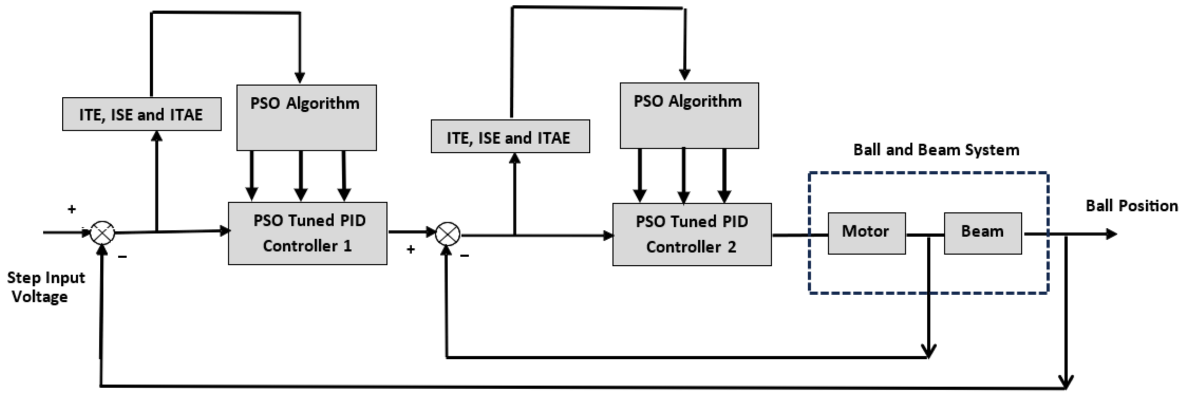

Figure 4.4 Block Diagram of PSO tuned PID control ball and beam system

The parameters chosen for tuning of PID controller using the PSO algorithm are, the number of populations is 50, the number of iterations is 50, acceleration constants are 1.5 & 2 and the inertia factor is 0.9.

4.6 Simulation Results and Discussions

4.6.1 PID tuning of Ball position.

The PID tuning of outer loop controller 1 used for controlling ball position was tuned using the conventional Zeigler Nichols method, PSO Algorithm, and TLBO algorithms. The K_{P1}, K_{i1} , and K_{d1} parameters obtained are shown in Table 4.2.

Table 4.2 Tuning parameters for ball position

S.no.	Method Used	Objective Function	K_{p1}	K_{i1}	K_{d1}
1.	ZN		0.841	1.014	0.254
2.	PSO Algorithm	ITAE	1.03	0.029	5.56
		ISE	15.96	4.60	0.87
		ITE	1.01	0.62	1.01
3.	TLBO Algorithm	ITAE	2.64	10	0.05
		ISE	10	0.000012	0.8016
		ITE	4.86	10	0.57

4.6.2 PID Tuning of Servo Motor Angle

The PID tuning of inner loop controller 2 used for controlling the servo motor was tuned using the conventional Zeigler Nichols method, PSO Algorithm, and TLBO algorithms. The K_{P2}, K_{i2}, and K_{d2} parameters obtained are shown in Table 4.3.

Table 4.3 Tuning parameters for servo motor angle

S.no.	Method Used	Objective Function	K_{p2}	K_{i2}	K_{d2}
1.	Zeigler Nichols		0.015	0.00004	1.119
2.	PSO Algorithm	ITAE	1.03	0.029	5.568
		ISE	0.1819	0.022	15.57
		ITE	1.0880	2.226	6
3.	TLBO Algorithm	ITAE	10	0.0022	10
		ISE	0.841	1.014	0.254
		ITE	10	0.148	10

The comparative step responses of various controllers based on ITAE, ITE, and ISE as objective functions are shown in Figures 4.5, 4.6, and 4.7.

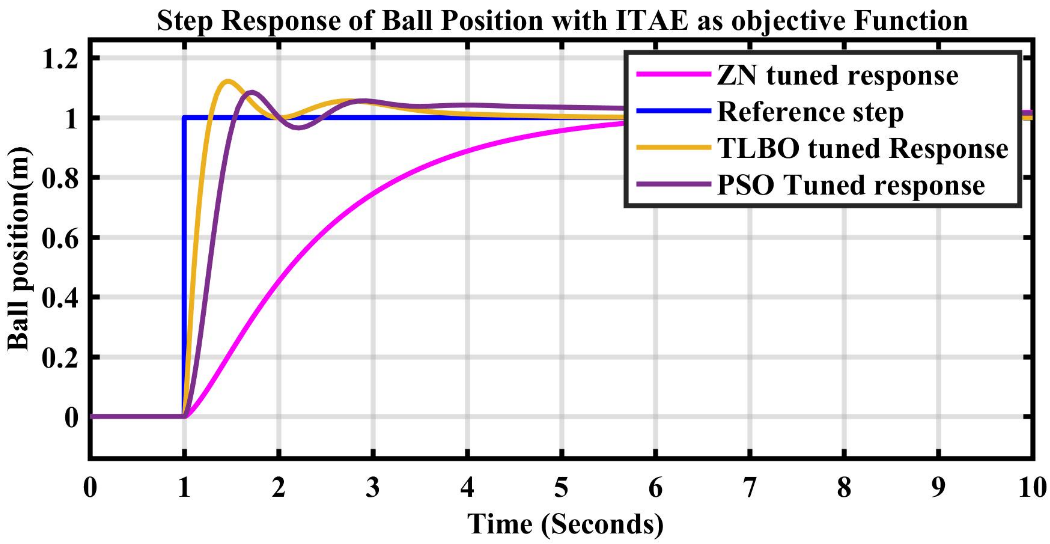

Figure 4.5 Step response of Ball position with ZN tuned, PSO tuned and TLBO tuned PID controller with ITAE as an objective function

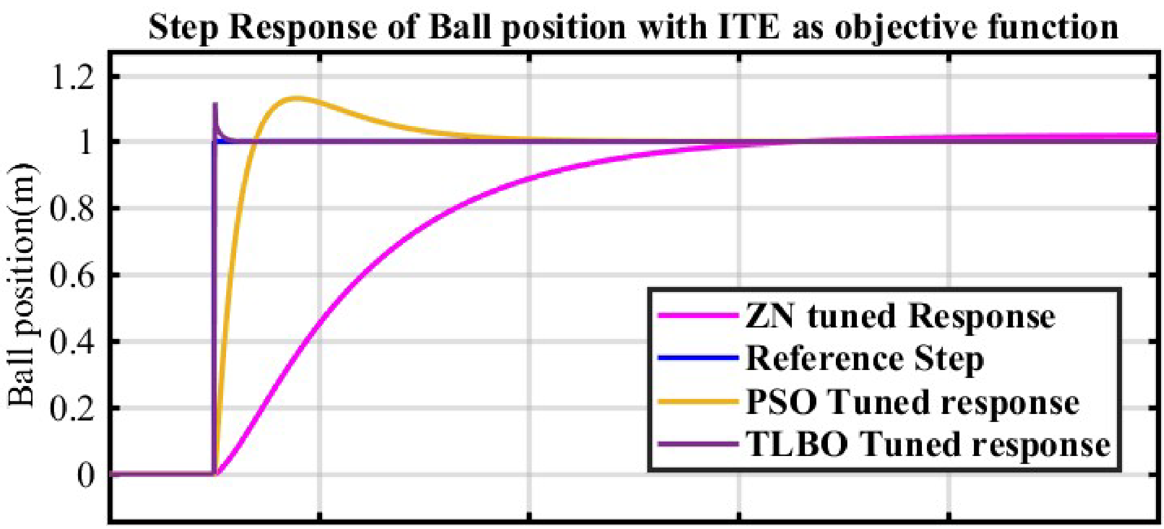

Figure 4.6 Step response of Ball position using ZN-tuned, PSO-tuned, and TLBO-tuned PID controller with ITE as an objective function

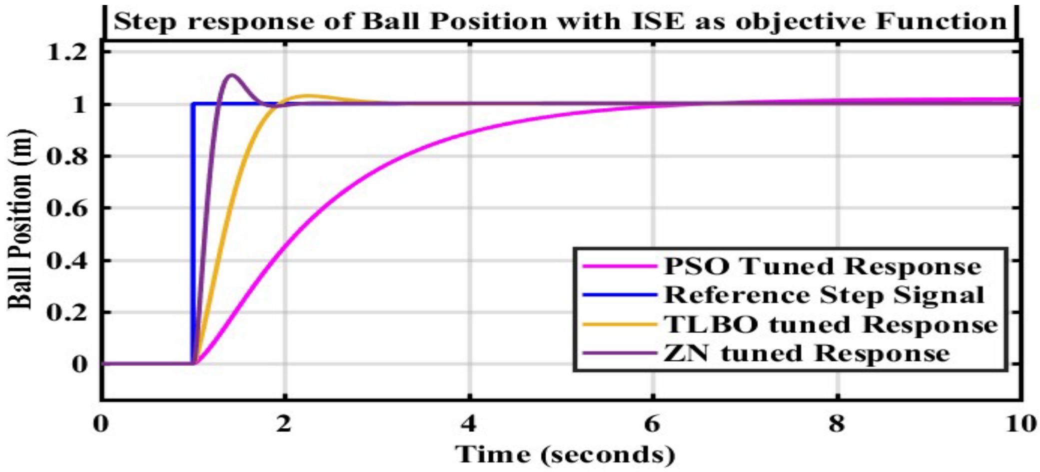

Figure 4.7 Step response of Ball position using ZN-tuned, PSO-tuned, and TLBO-tuned PID controller with ISE as an Objective Function

The performance specifications of various controllers are shown in Table 4.4: Comparative Study of Controllers

Table 4.4 Comparative performance of controllers

Sno.	**Tuning Method**	**Objective Function**	**Rise time (in seconds)**	**Overshoot %**	**Peak time (in seconds)**	**Settling time (in seconds)**
1.	Zeigler Nichols		3.60	0	7.20	8.62
2.	PSO	ITAE	0.3284	10	1.2	3.10
		ITE	0.2174	5.50	1.34	2.60
		ISE	0.2150	9.67	0.4564	0.7722
3.	TLBO	ITAE	0.2150	10.97	0.4794	2.60
		ITE	**0.01**	**4.56**	**0.21**	**0.456**
		ISE	2.86	1.93	3.60	4.55

From the various response specifications shown in Table 4.4, it is observed that the conventional tuned response based on Zeigler-Nichol's method is slow and sluggish. The Zeigler Nichols tuned response has a rise time of 3.6 seconds and a settling time of 8.62 seconds. It can be observed that the PSO-tuned and TLBO-tuned responses based on ITAE as an objective function have shown a much better response as compared to the conventionally tuned responses, the rise time has reduced to 0.3284 seconds in PSO tuned controller and to 0.2150 in TLBO tuned controller. Settling time has also reduced to 3.10 seconds in PSO tuned controller and 2.60 seconds in TLBO tuned controller. From Table 4.4 it is observed that the evolutionary algorithm-tuned response based on ITE as an objective function is better as compared to conventionally tuned controllers. In the case of the PSO-tuned controller, the rise time has reduced to 0.2174 seconds and the settling time 2.60 seconds whereas, in the case of

TLBO tuned response the rise time is 0.01 seconds and the overshoot is 4.56 % also in this case settling time is minimum 0.456 seconds. It can be observed from the comparative analysis that the optimized responses based on ISE objective function gives the best performance in terms of rise time and settling time. The TLBO Tuned controller gives better performance as compared to PSO Tuned controller and ZN tuned controller. To test the controller's performance for different sources, it was also tested with unit ramp input. Figure 4.8 shows the comparative performance of different controllers for a unit ramp input in case of ITE as an objective function.

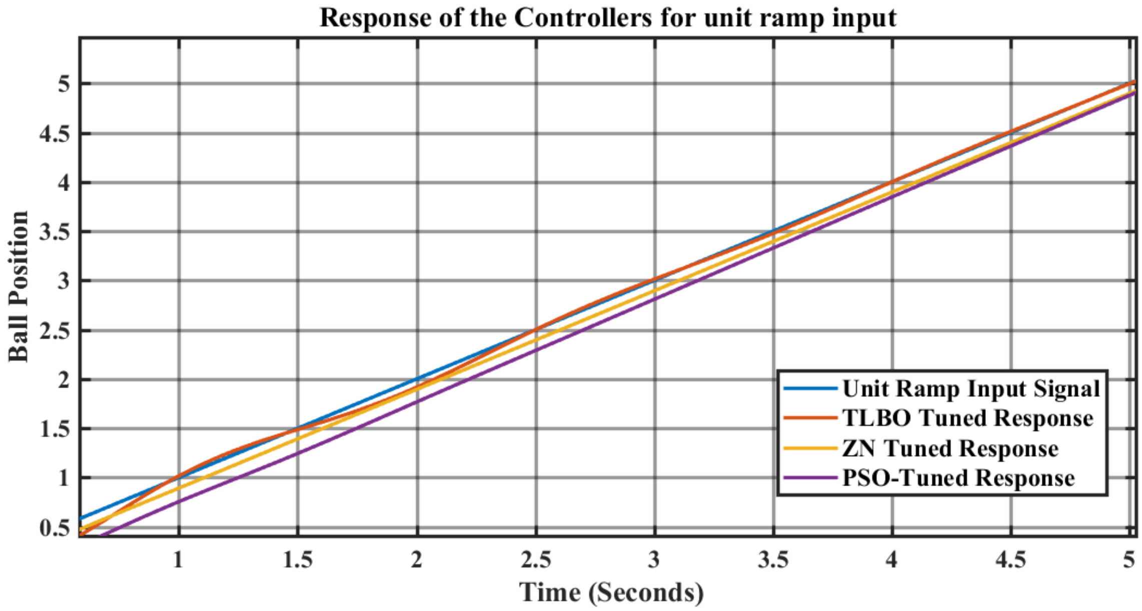

Figure 4.8 Comparative Response of different controllers for ramp input

4.6.3 Disturbance Analysis

The proposed controller was also analyzed under step disturbance conditions, the step response for the disturbance signal added at t=3 seconds is shown in Figures 4.9, 4.10, and 4.11.

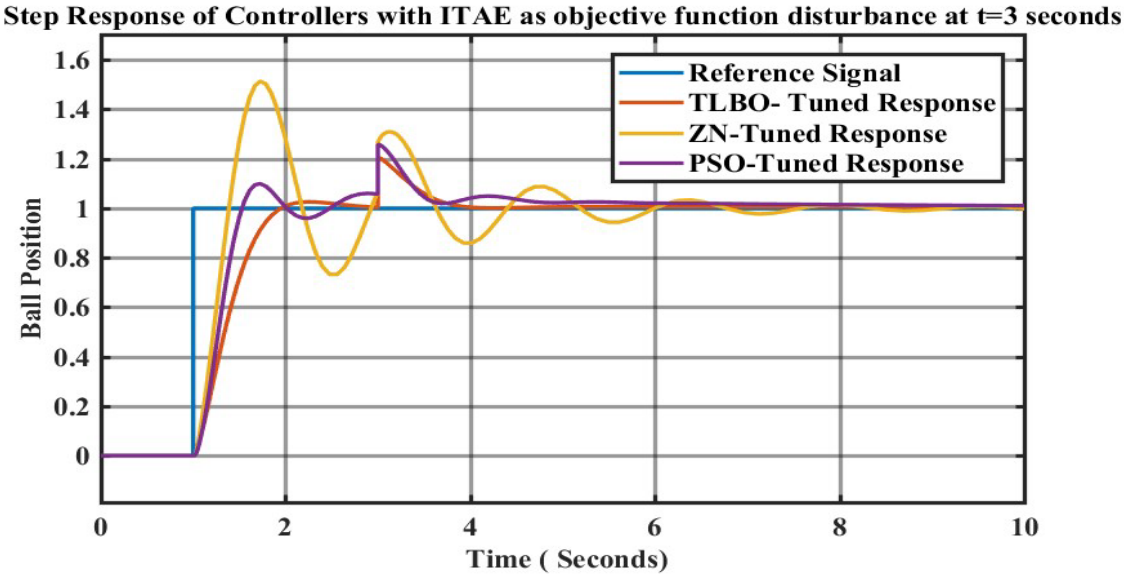

Figure 4.9 Comparative step response of the controllers with disturbance and ITAE as the objective function

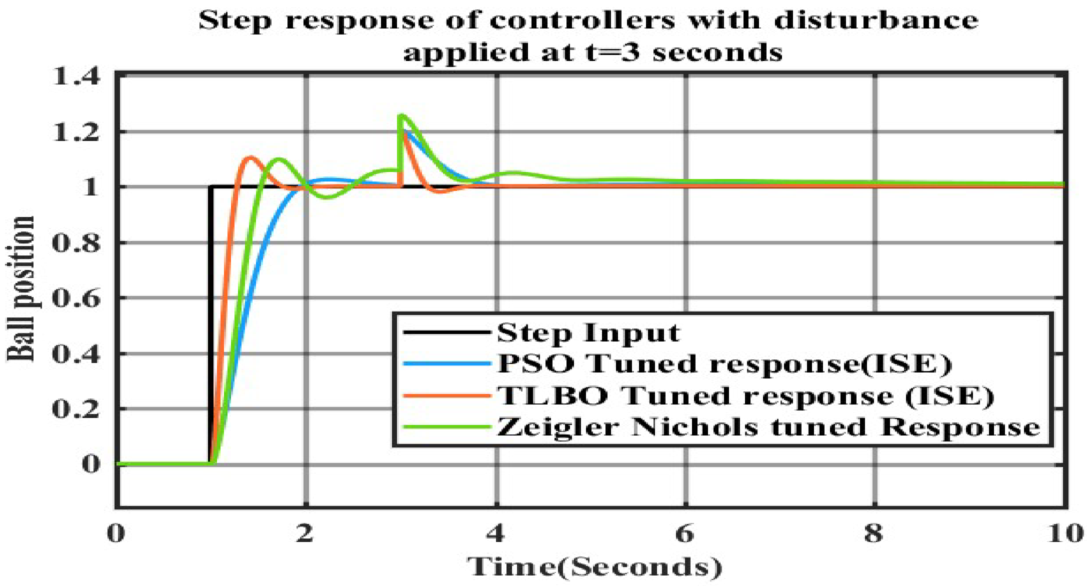

Figure 4.10 Comparative Step responses of the controller with disturbance and ISE objective function

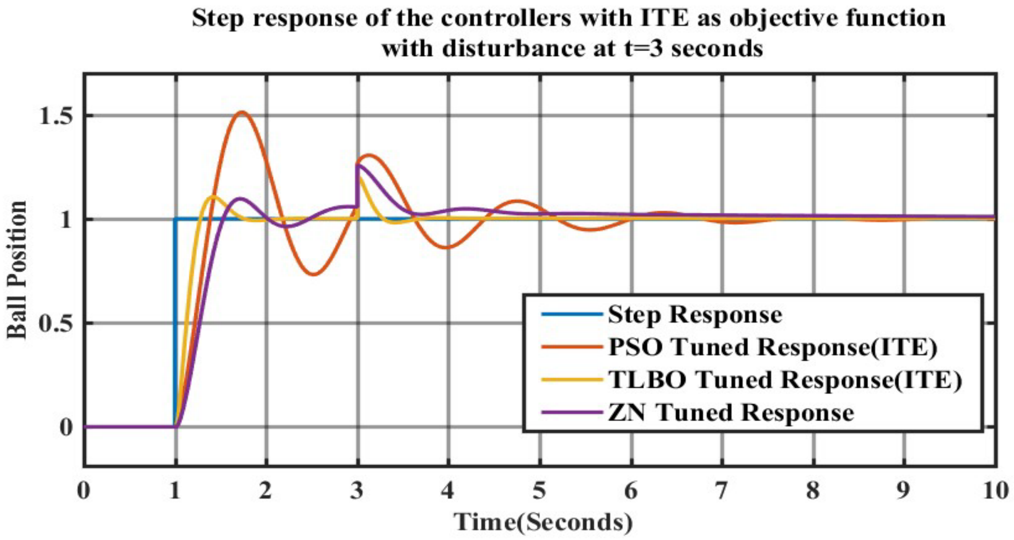

Figure 4.11 Comparative Step responses of controllers with disturbance and ITE as the objective function

It can be observed from the responses that when the disturbance is added at t=3 seconds, the TLBO tuned response performs best as compared to the PSO-tuned and Zeigler Nichols tuned response. The comparative values of the settling time are given in Table 4.5

Table 4.5 Comparative performance of different controllers in case of disturbance signal

Sno.	Tuning Method	Objective Function	Settling Time (Seconds)
1.	Zeigler Nichols		6.86
2.	PSO	ITAE	5.70
		ITE	5.77
		ISE	4.65
3.	TLBO	**ITAE**	**3.86**
		ITE	4.80
		ISE	4.04

From Table 4.5 it can be observed that the TLBO-tuned controller was able to reject the disturbance within 3.86 seconds in the case of ITAE as an objective function. The TLBO-tuned response has a minimum settling time as compared to the PSO-tuned controller and Zeigler-Nichols-tuned controller. The control signals generated by TLBO Tuned, PSO Tuned, and ZN Tuned controllers are shown in Figures 4.12, 4.13, and 4.14 for different objective functions.

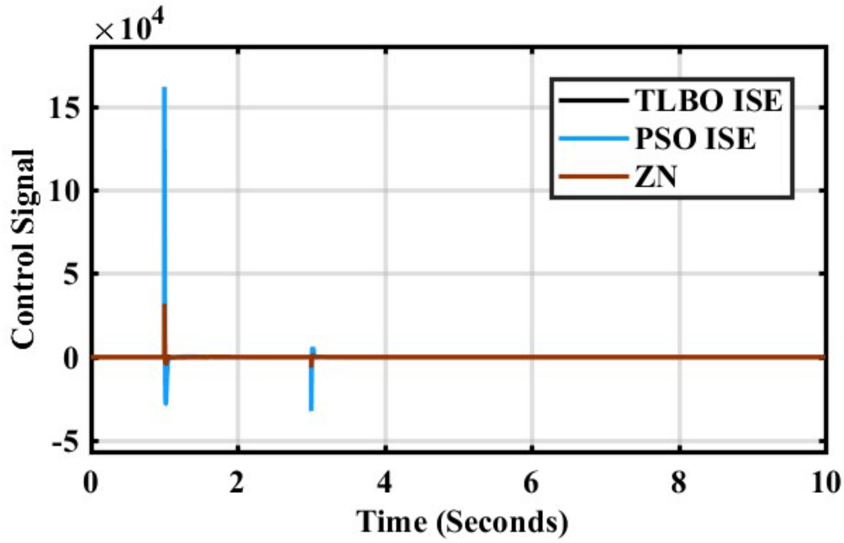

Figure 4.12 Control signals generated by TLBO Tuned, PSO Tuned and ZN Tuned controllers with ISE as objective Function

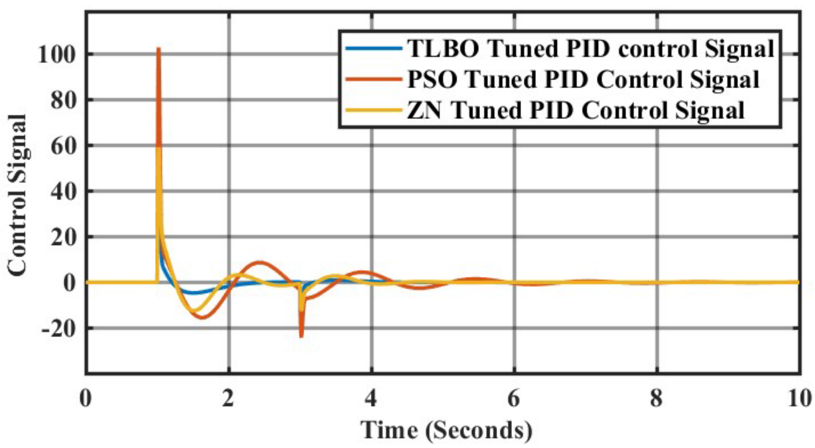

Figure 4.13 Control signals generated by TLBO Tuned, PSO Tuned and ZN Tuned controllers with ITAE as objective Function

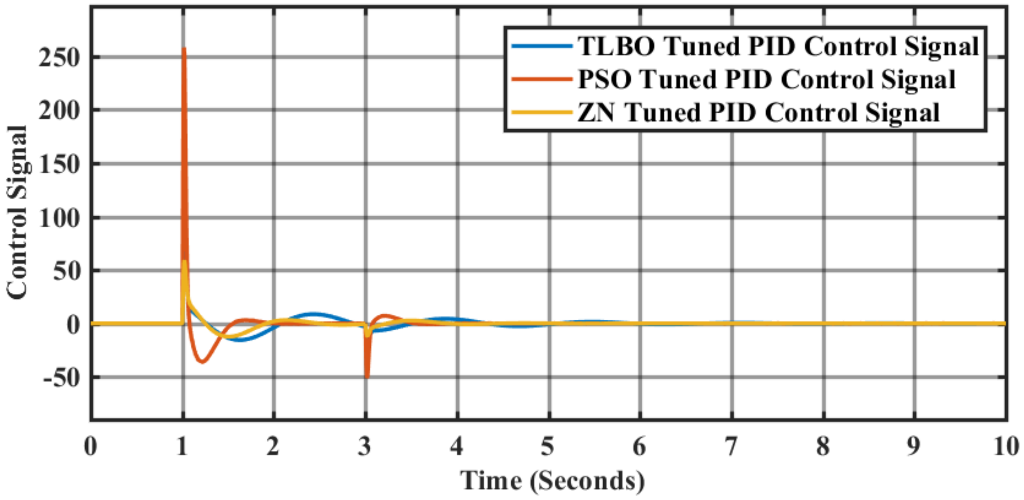

Figure 4.14 Control signals generated by TLBO Tuned, PSO Tuned and ZN Tuned controllers with ITE as objective Function

It can be observed from Figures 4.12,4.13 and 4.14 that the control signal comes into action as soon as the disturbance is applied at t=0 and t=3 seconds and can control the output signal within seconds.

4.6.4 Sensitivity Analysis

The sensitivity analysis of the proposed TLBO-tuned cascade PID controller is carried out for the parameter variations. In the first case, the controller is tested for robustness by varying the system parameter, lever arm offset. It varied from 1 inch to 0.85 inches and 1.25 inches.

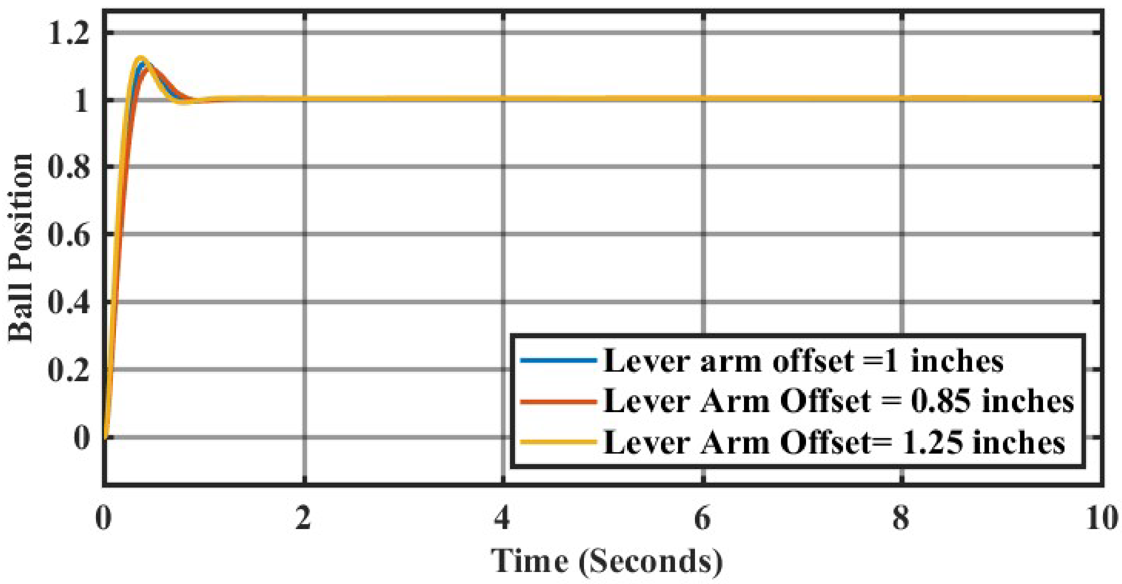

Figure 4.15 Sensitivity Variation with change in Lever arm offset

Figure 4.15 shows the response of the controller for the variations. It can be observed from Figure 4.15 that the controller can cope with the system variations. In the second case, the controller sensitivity was tested for wear and tear by varying the value of gearbox efficiency ηg, from 0.9 to 0.85 and 0.7. Figure 4.16 shows the response of the controller with variation in ηg.

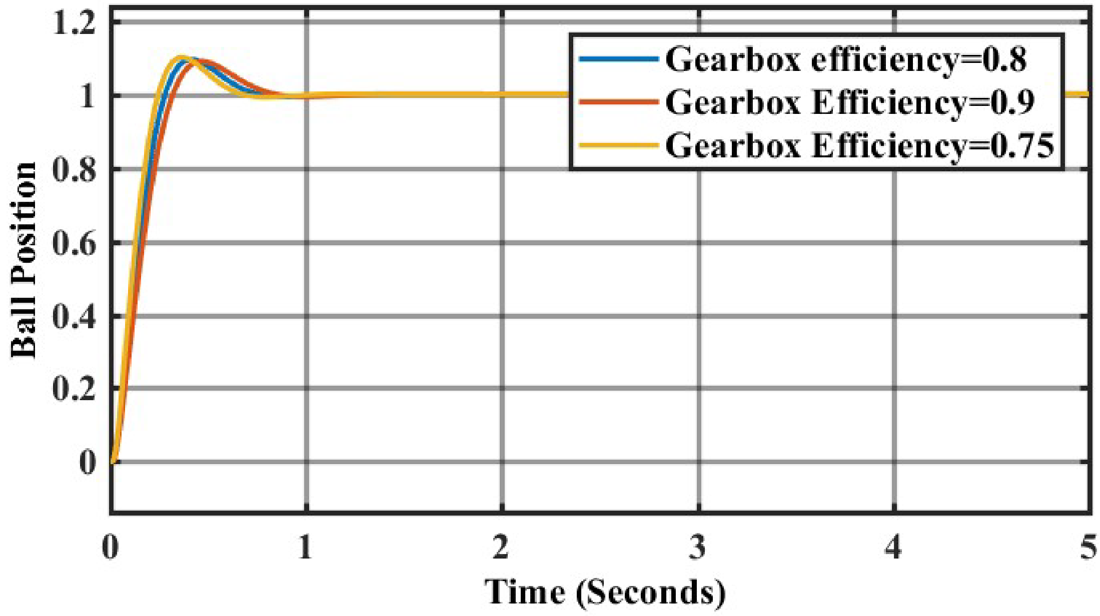

Figure 4.16 Sensitivity Analysis with variation in Gearbox efficiency

It can be observed from Figure 4.16 that the proposed TLBO controller can overcome the variation due to the gearbox efficiency.

4.7 Conclusion

In this chapter, it can be concluded that TLBO and PSO optimization-based algorithms can be used efficiently for tuning PID controllers in a cascade configuration. The proposed controllers are tested on a non-linear ball and beam system benchmark challenge. The mathematical analysis of the ball and beam system is performed. A cascade PID control strategy is used. The traditional Zeigler Nichols method, PSO algorithm, and TLBO algorithm are used to tune the two PID controllers. By comparing the responses of different controllers tested, it was discovered that the TLBO-tuned controller with ITAE and ITE as objective functions produced the best response as compared to the PSO-tuned controller and the conventionally tuned controller. With a settling time of 8.62 seconds, the response of a typically Zeigler Nichols-tuned PID controller is exceedingly slow. The settling time is lowered to 3.20 seconds when the PSO-tuned response is used. However, the TLBO-tuned controller had the best response time of 2.66 seconds. The TLBO-tuned controller is also tested in case of disturbance input. The comparative analysis indicates that the proposed TLBO controller performs well when the disturbance is applied at t=3 seconds.

CHAPTER 5

OPTIMIZATION OF FRACTIONAL ORDER PID CONTROLLER BY METAHEURISTIC ALGORITHMS FOR INVERTED PENDULUM-CART SYSTEM

5.1 Introduction

In the past many researchers have proposed various methods to tune conventional PID controllers for non-linear systems[224]. Since PID is still one of the most widely used controllers in the industry[239]. But its tuning is difficult by standard methods if we are dealing with a system having a lot of uncertainties and parameter changes. Recently, some researchers have proposed a fractional-order PID controller whose performance was more precise as compared to conventional PID controllers[14], [33], [35]. The addition of fractional calculus to the conventional PID controller can enhance the performance, as they have many advantages. A fractional-order operator can add memory to the controller. In a fractional-order PID controller, the degrees of derivative and integration terms are added. The values of the degrees can be an integer or a fractional value. This makes it a controller having five degrees of freedom instead of three in a conventional PID controller. However, this addition increases the system's complexity.

It is evident from the past literature, that a FOPID controller performs better than a conventional PID controller. Several tuning methods have been proposed and implemented by various researchers in the past for tuning PID and FOPID controllers. Out of the various tuning methods, bio-inspired algorithm-based tuning methods are the most popular ones because of their good performance. In [33], Zafer Bingul et. al. applied PSO and ABC for tuning of FOPID and PID controller on two systems with delay. The results proved that the ABC-tuned controller proved better in terms of robustness and external disturbance. In [240], Prashant Kumar et. al. applied a simulated annealing method to tune the parameters of a FOPID controller applied to a field-controlled DC motor. The results found were compared with conventional methods like frequency design-based method and genetic algorithm-based method. In [241] Ruchi V. Jain et al. used PSO to fine-tune the FOPID controller's parameters for controlling the speed of a DC motor. Amlan Basu et al. created numerous ways [35]to derive the parameters of the FOPID controller using traditional PID tuning methods, and they compared the results. The performance of the FOPID controller was shown to be superior to that of the traditional PID and the neural network-tuned PID controller in [242]by Naser Sadati et. al.

One of the most important problems in control theory that has been extensively studied in control literature is the inverted pendulum problem, which we have addressed in this chapter. It is a well-known benchmark challenge that presents numerous difficult control design problems. The system is an underactuated, nonminimum phase, unstable, and nonlinear. Pendulums have maintained their usefulness due to their nonlinear character, and many of the concepts being developed in the field of nonlinear control are currently shown using them. The inverted pendulum systems became a standard tool in control laboratories due to the control issues. An inverted-pendulum cart system is modeled mathematically and controlled by a FOPID controller whose five parameters are tuned by various meta-heuristic algorithms.

5.2 Development of a Mathematical Model of an Inverted-pendulum Cart System

This section models an inverted pendulum that is mounted on a moving cart mathematically. A location on the cart that is fixed to the pendulum is the only place the pendulum is free to move about. The cart glides on a horizontal platform and is propelled by a stepper motor. The control's main objective is to push the cart hard enough to keep the pendulum upright. Figure 5.1 depicts a cart and an inverted pendulum. F is the force expressed in Newtons (N), m_p is the pendulum's mass expressed in kilograms (kg), M_c is the mass of the cart, F_H is the force acting on the cart, F_{sr} is the friction on the cart's surface, x is the displacement of the cart, l is the length of the cart and θ is the angle of the pendulum with vertical axis.

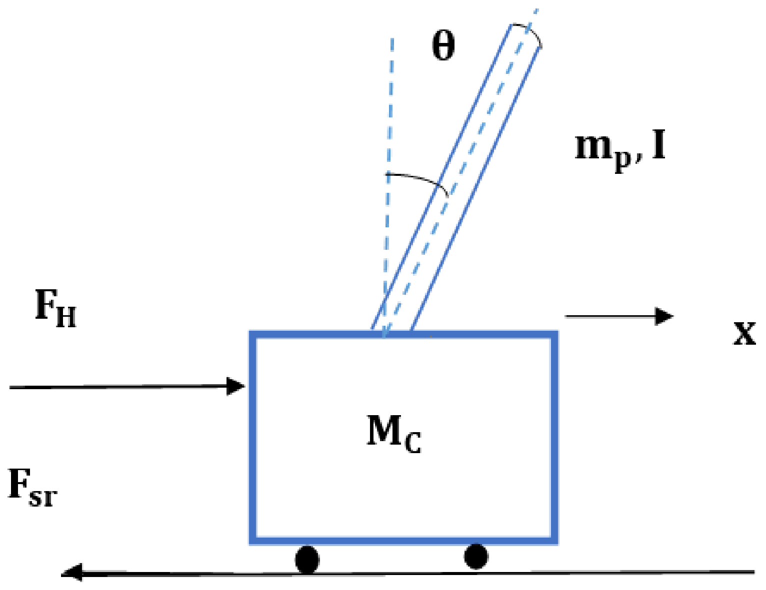

Figure 5.1 Schematic Diagram of an inverted-cart pendulum system

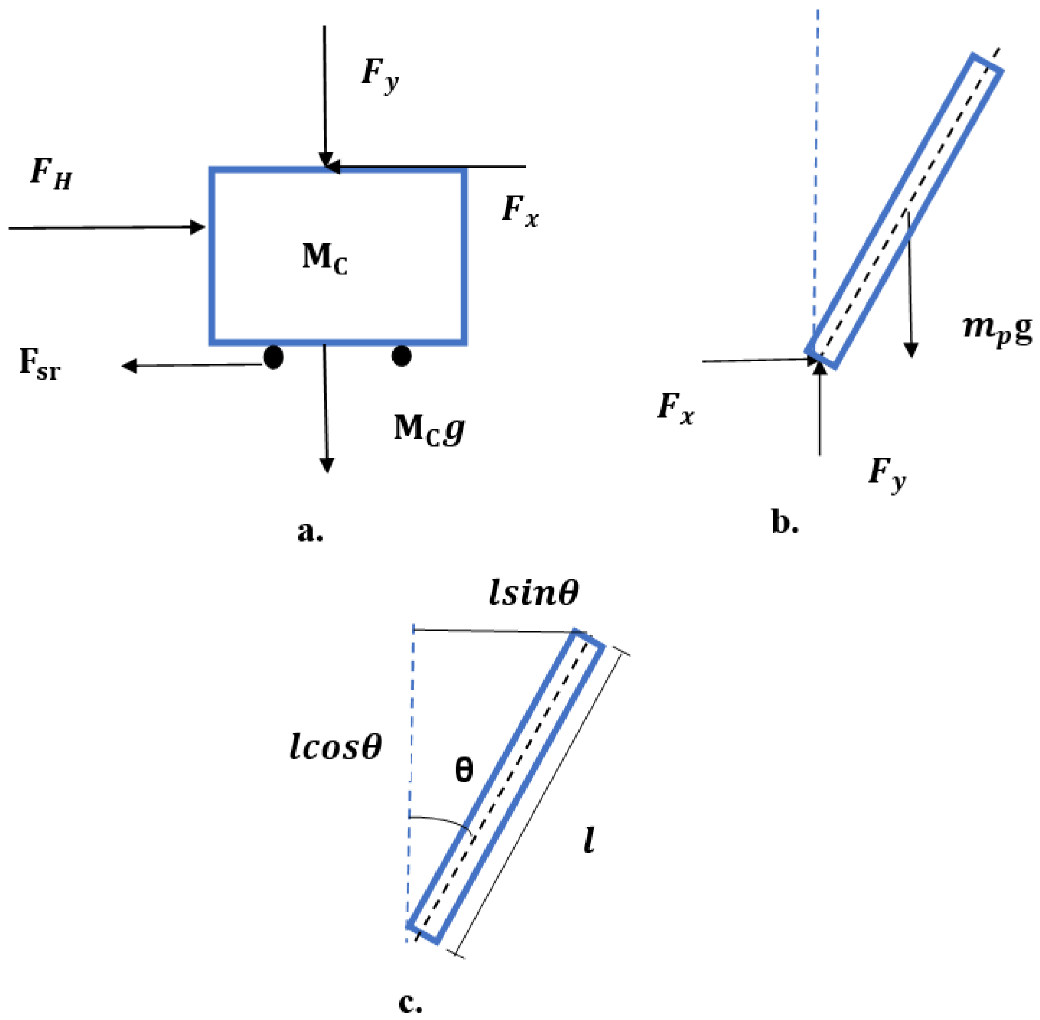

Figure 5.2a. Free body diagram of cart b. Freebody diagram of pendulum c. Diagram depicting the relation between the length and angle of the pendulum

If we calculate the centre of the gravity coordinates of the part pendulum, we get,

$$x_{pg} = x + lsin\theta \tag{5.1}$$

$$y_{pg} = lcos\theta \tag{5.2}$$

The forces in the horizontal direction are given as,

$$M_C \frac{d^2x}{d^2t} = F_H - b_f \frac{dx}{dt} - F_x \tag{5.3}$$

$$F_x = m_p \frac{d^2}{dt^2}(x + lsin\theta\ ) \tag{5.4}$$

Summing all the forces in the horizontal direction,

$$F_X = M_C \frac{d^2x}{d^2t} + m_p lsin\theta \frac{d^2\theta}{d^2t} - m_p lcos\theta \left(\frac{d\theta}{dt}\right)^2 \tag{5.5}$$

Substituting the value of equation (5.2) in equation (5.1)

$$(M_C + m_p)\frac{d^2x}{d^2t} + b_f \frac{dx}{dt} + m_p lsin\theta \frac{d^2\theta}{d^2t} - m_p lcos\theta \left(\frac{d\theta}{dt}\right)^2 = F_H \tag{5.6}$$

In the vertical direction, the forces are,

$$F_y = m_p g - m_p \frac{d^2}{dt^2}(lcos\theta) \tag{5.7}$$

Also,

$$F_y - m_p g = -m_p lsin\theta \frac{d^2\theta}{d^2t} - m_p lcos\theta \left(\frac{d\theta}{dt}\right)^2 \tag{5.8}$$

Eliminating F_x and F_y terms,

$$-F_y lsin\theta - F_x lcos\theta = I\frac{d^2\theta}{d^2t} \tag{5.9}$$

$\theta = \pi + \varphi, cos\varphi = -cos\theta \ and \ sin\varphi = -sin\theta$

$I = \frac{1}{3} m_p l^2$

$$\frac{4}{3} m_p l^2 \theta + m_p glsin\theta = -m_p l \frac{d^2x}{dt^2} cos\theta \tag{5.10}$$

Considering $\theta = \pi$ as the operating point. Linearizing the equations around the operating point, we get

$$\frac{4}{3} l \frac{d^2\varphi}{dt^2} - g\varphi = \frac{d^2x}{dt^2} \tag{5.11}$$

$$(M_C + m_p)\frac{d^2x}{dt^2} + b_f \frac{dx}{dt} - m_p l \frac{d^2\varphi}{dt^2} = u \tag{5.12}$$

To determine the transfer function let us find the Laplace transform of equations,

$$\left(I + m_p l^2\right)\varphi(s)s^2 - m_p gl\varphi(s) = m_p lX(s)s^2 \tag{5.13}$$

$$\left(M_C + m_p\right)X(s)s^2 + b_f X(s)s - m_p lX(s)s^2 = U(s) \tag{5.14}$$

Where X(s) and $\varphi(s)$ are Laplace transform of x(t) and $\varphi(t)$

$$\frac{\varphi(s)}{X(s)} = \frac{4}{3} l - \frac{g}{s^2} \tag{5.15}$$

Substituting the values of X(s) in equation (5.12) we get,

$$\frac{\varphi(s)}{U(s)} = \frac{\frac{m_p l}{q} s^2}{s^4 + \frac{\frac{4}{3} b_f m_p l^2}{q} s^3 - \frac{m_p g l (M_C + m_p)}{q} s^2 - \frac{b_f m_p g l}{q} s} \tag{5.16}$$

where, $q = [(M_C + m_p)(I + m_p l^2) - (m_p l)^2]$

Table 5.1 represents the values of parameters of the inverted pendulum cart system,

Table 5.1 Inverted Pendulum-Cart Parameters

S. No.	Parameters	Values
1.	Length of pendulum(l)	0.25m
2.	Mass of cart system (M_C)	1.096Kg
3.	Mass of pendulum system (m_p)	0.109Kg
4.	Frictional coefficient,(b_f)	$0.1Nm^{-1}s^{-1}$
5.	Inertia(I)	1.0036 Kgm^2

Putting the values from Table 5.1 the transfer function is derived as,

$$\frac{\theta(s)}{U(s)} = \frac{2.35655s}{s^3 + 0.00883s^2 - 27.9169s - 2.30942} \tag{5.17}$$

5.3 Fractional-order PID Controller

The FOPID controller was first made available by Podlubny in 1999, [26]. It was a modified form of the standard PID controller. Through several experimental outcomes, Podlubny shown in [26], that FOPID generates results that are superior to those of a standard PID controller. The FOPID controller has five degrees of freedom compared to the PID controller which has only three degrees of freedom. The five degrees for a FOPID are the proportional gain

K_p,integral gain K_i,derivative gain K_d,integral power, and derivative power term, μ. Figure 5.3 depicts the block diagram of a FOPID Controller. In a fractional order PID controller there is a fractional order integrator having order μ and a fractional order integrator of order λ. These fractional powers help in better closed-loop responses due to the addition of two extra degrees of freedom. Many researchers have suggested that FOPID controllers perform better than classical PID controllers. The main tuning methods available for fractional order PID controllers are analytical formula based, based on frequency domain specifications and optimization based[35]. IMC method is one of the analytical methods applicable to FOPID controllers, but they depend on plant model. The gain margin-phase margin method is also applied by some researchers for tuning FOPID controllers. A study [243] used the frequency domain approximation method for tuning the FOPID controller. But to apply a FOPID controller one should ensure that it is better than a conventional PID controller. Therefore, optimization methods are the best tuning methods for FOPID controllers [244].

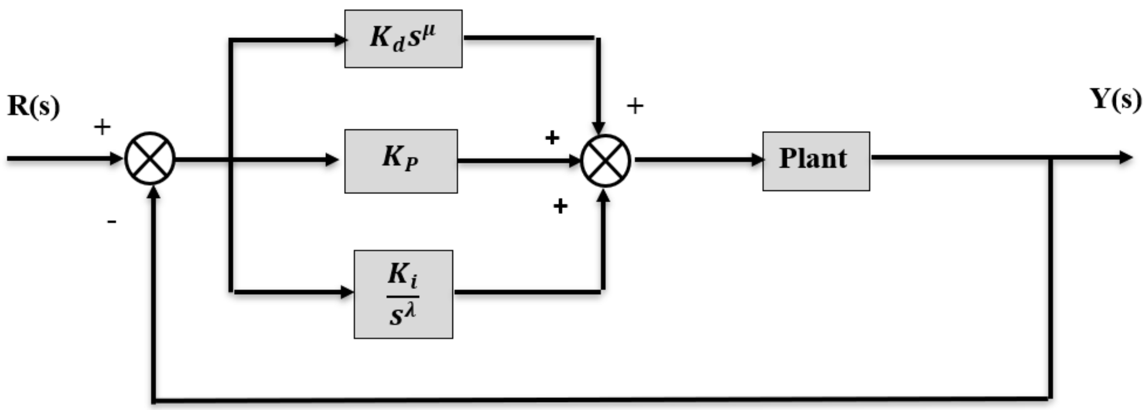

Figure 5.3 Block diagram of Fractional-order PID Controller

The equation of a FOPID system is given as,

$$U(t) = K_p e_{rr}(t) + K_i d^{-\lambda} e_{rr}(t) + K_d d^{\mu} e_{rr}(t) \quad (5.18)$$

Taking Laplace transform,

$$U(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \quad (5.19)$$

To convert a FOPID to a classical PID put $\lambda = 1$ & $\mu = 1$. By putting the integral and derivative powers to 1. The controller will be a simple PID controller.

5.3.1 Fractional Order Calculus

For fractional order calculus, a differential integration operator $D_b^{t\,\lambda}$ is defined. In the operator λ is defined as the order, b & t are the limits.

$$D_b^{t\,\lambda} = \begin{cases} \frac{d^\lambda}{dt^\lambda}, & \lambda > 0 \\ 1 & \lambda = 0 \\ \int_b^t d\tau^{-\lambda}, & \lambda < 0 \end{cases} \tag{5.20}$$

λ can be a real or a complex number

Consider a LTI system with r(t) as input and y(t) as output

$$a_n D^{\lambda_n} y(t) + a_{n-1} D^{\lambda_{n-1}} y(t) + \cdots \ldots \ldots .. + a_o D^{\lambda_o} y(t) = b_m D^{\mu^m} r(t) + b_{m-1} D^{\mu^{m-1}} r(t) + \cdots \ldots \ldots \ldots .. + b_o D^{\mu^o} r(t) \tag{5.21}$$

Where, a_i, λ_i,b_i, μ_i are all real constants for i=0,1,2…..,n).

Taking Laplace transform and considering zero initial conditions,

$$\frac{Y(s)}{R(s)} = \frac{b_m s^{\mu m} + b_{m-1} s^{\mu m-1} + \cdots .. + b_o s^{\mu o}}{a_n s^{\lambda n} + a_{n-1} s^{\lambda n-1} + \cdots \ldots\ldots a_o s^{\lambda o}} \tag{5.22}$$

5.4 Proposed Optimized FOPID Controller

The inverted pendulum position is controlled by an FOPID controller with proportional, integral, derivative, integral order, and derivative order gains optimized via the Whale optimization method. The objective function of the ITAE that has been utilized to achieve the best outcome It can be mathematically represented as,

ITAE (Integral time absolute error)

$$ITAE = \int_0^T t|e_r(t)|dt \tag{5.23}$$

Where, $e_r(t)$ is the error signal between the desired level and actual level

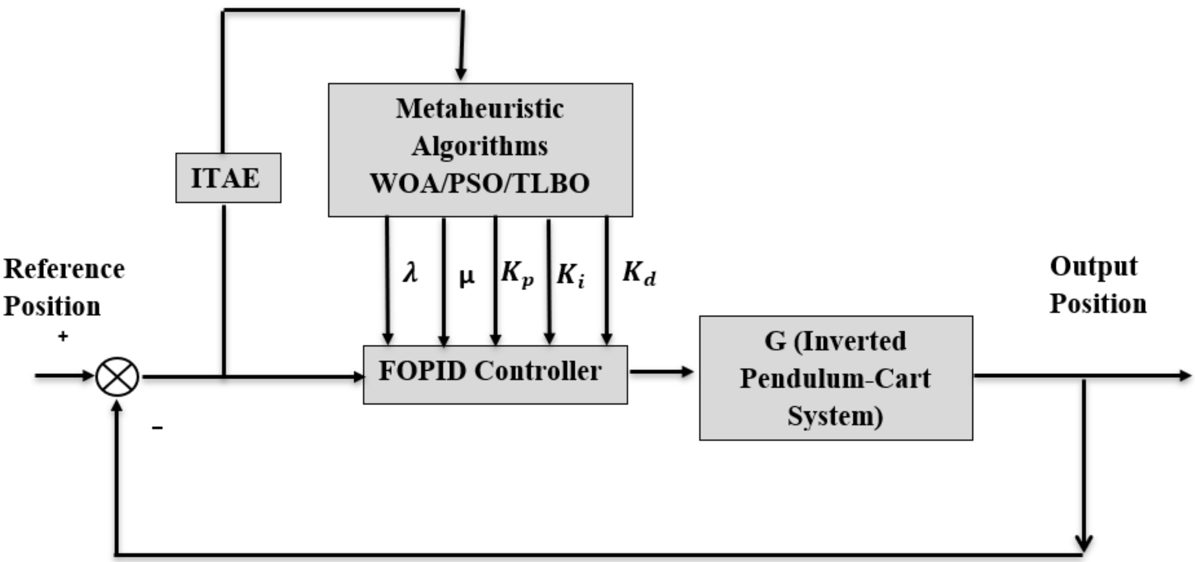

Figure 5.4 Block Diagram of the Proposed Metaheuristic Algorithms tuned FOPID Controller

5.5 Whale Optimization Algorithm

The whale optimization algorithm is a recently developed swarm-based algorithm[84]. In some research, it was found that like humans, whales also have spindle cells. Therefore, they also show emotional and social behaviors. The Whale optimization algorithm has the main advantages of a faster convergence rate and high capability to reach a globally optimal solution. The approach employed for whale optimization is that it mimics the hunting techniques used by humpback whales. A unique form of hunting used by humpback whales is known as the bubble net feeding method. In this behavior, the whale first dives down in the water and then creates spiral-shrinking bubbles around the prey while coming towards the surface. The algorithm of WOA is as follows:

1. In the first step, a random initial population, the maximum number of iterations, and variables are specified
2. In the second step, for each variable fitness function is calculated for every search agent, and the best solution is obtained. The rest of the search agents update the positions accordingly.
3. Updating of position for search agent is done by following equations,

i. If $P_r < 0.5$ where, P_r is a random number between 0 and 1.

a. If $\overrightarrow{|A_w|}$ <1, then position is updated by encircling the prey method. The equation for encircling the prey is defined as,

$$\overrightarrow{X_W(k+1)} = \overrightarrow{X_{BEST}}(k) - \overrightarrow{A_W}\overrightarrow{D_W} \quad (5.24)$$

Where, k denotes current iteration, A_W and C denotes coefficient vectors, $\overrightarrow{X_{BEST}}$ denotes best solution's position and

$$\overrightarrow{A_W} = 2\vec{a}\overrightarrow{r_1} - \vec{a} \quad (5.25)$$

$$\vec{C} = 2\overrightarrow{r_2} \quad (5.26)$$

$\overrightarrow{r_1}\ and\ \overrightarrow{r_2}$ are random numbers between [0,1].

$$\overrightarrow{D_W} = \left|\vec{C}.\overrightarrow{X_{BEST}(k)} - \overrightarrow{X_W(k)}\right| \quad (5.27)$$

b. If $\overrightarrow{|A_w|}$ <1, then the updating of position is done by exploration phase method,

$$\overrightarrow{X_W(k+1)} = \overrightarrow{X_{rand}}(k) - \overrightarrow{A_W}\overrightarrow{D_W} \quad (5.28)$$

$$\overrightarrow{D_W} = \left|\vec{C}.\overrightarrow{X_{rand}(k)} - \overrightarrow{X_W(k)}\right| \quad (5.29)$$

Where, $\overrightarrow{X_{rand}}(k)$ is the random position vector

ii. If $P_r > 0.5$, then position is updated by spiral movement

The equations for spirally updating the position of bubbles are given as,

$$\overrightarrow{D_W} = \left|\vec{C}\vec{X}_{BEST}(t) - \vec{X}(t)\right| \quad (5.30)$$

$$X_W(k+1) = \overrightarrow{D_W}.e^{bl}\cos(2\pi l) + \overrightarrow{X_{BEST}}(t) \quad (5.31)$$

Where, l is a random number between (-1,1)

4. Repetition of the 2nd and 3rd steps is done for updating position and calculation of fitness function till maximum iteration is reached.

The flowchart of the whale optimization algorithm is shown in Figure 5.4,

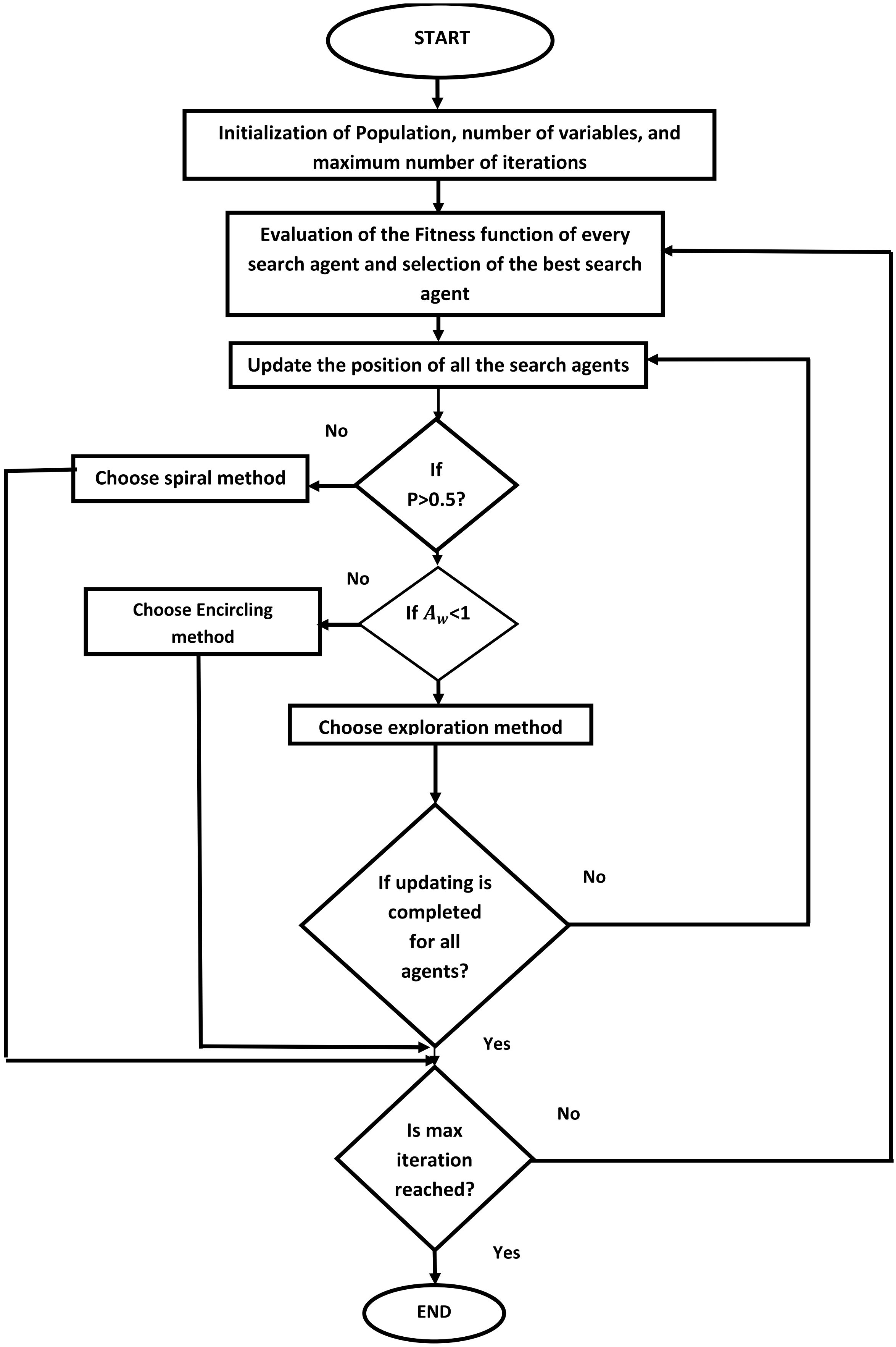

Figure 5.5 Flowchart of Whale Optimization Algorithm

5.6 Simulation Results

The FOPID and PID parameters are tuned by three metaheuristic algorithms PSO, TLBO, and WOA using ITAE as an objective function. The comparative step response of the WOA tuned, PSO Tuned and TLBO Tuned FOPID Controller WOA-tuned, PSO tuned and TLBO-tuned PID controller is shown in Figure 5.6.

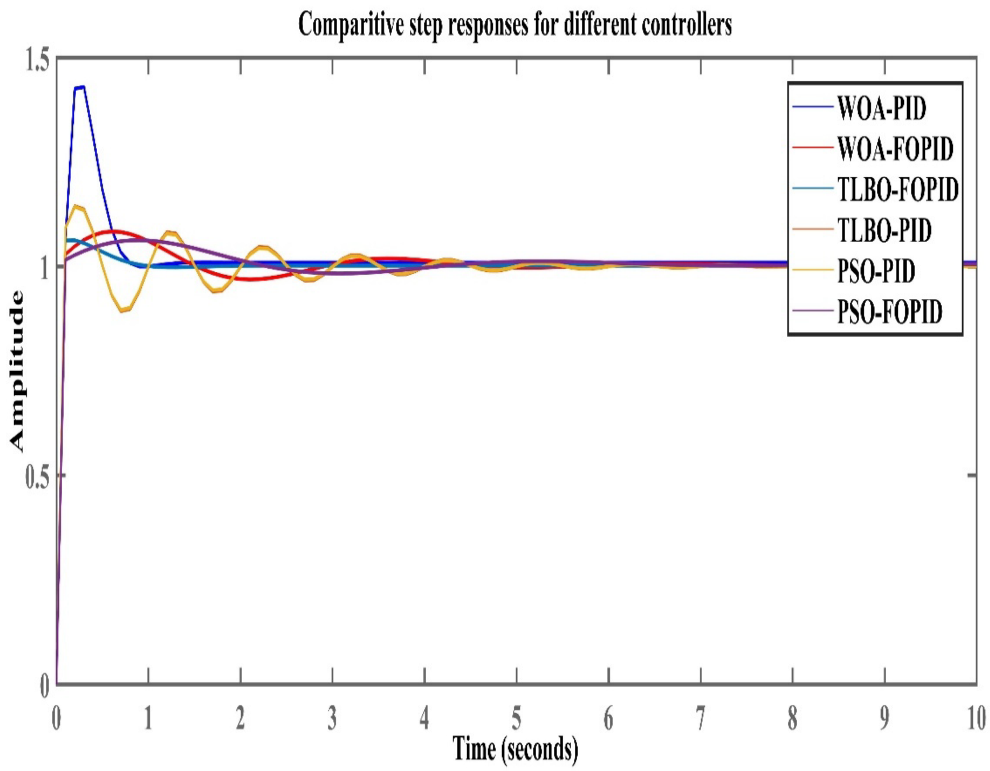

Figure 5.6 Comparison of step responses of WOA-tuned FOPID and PID Controller

The Figure 5.6 compared the optimized PID controllers to the optimized FOPID controllers.

The performance specifications of the proposed controllers are shown in Table 5.2.

Table 5.2 Performance Indices of Various Controllers (With ITAE As Objective Function)

Method	Rise time, tr seconds	Settling time, t_s seconds	Max. Overshoot (% M_p)	Peak time t_p, (seconds)
WOA tuned PID	0.0659	0.8477	35.47	0.2547
TLBO tuned PID	0.0301	3.76	14.72	0.2360
PSO Tuned PID	0.0299	3.34	14.56	0.2346
WOA Tuned FOPID	0.0164	0.6143	7.98	0.1498
TLBO Tuned FOPID	**0.0191**	**0.6322**	**6.18**	**0.1507**
PSO Tuned FOPID	0.0299	0.7173	4.15	0.1324

Table 5.3 Values Of K_p, K_i , K_d, μ and λ parameters

Method	K_p	K_i	K_d	λ	μ
WOA - PID	51.99	100	6.2141	-	-
TLBO-PID	56.45	993.63	25.53	-	-
PSO-PID	292.15	994.65	88.85	-	-
WOA -FOPID	94.55	67.66	98.83	0.766	0.336
TLBO-FOPID	**208.58**	**543.29**	**39.85**	**0.3579**	**0.9038**
PSO-FOPID	70	543.49	51.86	0.4764	0.8908

The comparison of the values of the objective function ITAE is shown in Table 5.4

Table 5.4 Comparison of ITAE values

Controller	ITAE
WOA-PID	0.00625
TLBO-PID	0.00135
PSO-PID	0.00236
WOA-FOPID	0.00025
TLBO-FOPID	**0.00012**
PSO-FOPID	0.00024

The results of the simulation demonstrate that the TLBO-FOPID controller outperforms the other controllers in terms of settling time. The metaheuristic algorithms tuned FOPID controllers were compared with the conventional PID controller. The FOPID controllers gave better results as compared to PID controllers. There is a substantial improvement in the overshoot, settling time, and rise time. The overshoot decreased from 35% in the case of the WOA-tuned PID controller to 7.98% in the case of the WOA-tuned FOPID controller. The value of the TLBO-FOPID controller is minimal as compared to other controllers.

5.7 Conclusion

In this chapter, a FOPID controller whose parameters are tuned using three optimization algorithms, PSO, TLBO, and the Whale optimization algorithm is used to control the position of a nonlinear inverted cart pendulum system. The rise time, settling time, and peak time of the FOPID controllers are all improved when compared to the performance of the conventional PID controllers. If we compare the performance of three proposed metaheuristic algorithms TLBO has performed better as compared to the PSO and TLBO. TLBO-tuned FOPID gives the best results in terms of rise time, settling time, and overshoot.

CHAPTER 6

DESIGN OF PSO-NN PID CONTROLLER FOR CONTROL OF NON-LINEAR JACKETED CSTR SYSTEM

6.1 Introduction

To address the issue of PID tuning for non-linear systems, metaheuristic algorithm-based PID controllers were applied to automobile cruise control systems, artificial respiratory systems, and ball and beam systems in the previous chapters. It was observed from the study that the optimized PIDs performed better as compared to conventional PID controllers. They searched the entire search space to find the optimal solutions[244]. However, these algorithms mainly suffered from the issue of adaptability to variations in the system and external disturbances. Also, there are certain parameters in each algorithm which are needed to be properly tuned to obtain converging results. In the literature, it was observed that many researchers have also used neural network-based controllers for tuning PID gains[169], [171], [245], [246]. Neural network-based controllers provide better adaptability to the system's parameter variation and setpoint changes. However, most of the neural network-based controllers were based on backpropagation. The back propagation is based on a gradient descent algorithm which mostly gives local optimal solution instead of global optimal solution[171]. Therefore, we can use the metaheuristic algorithm to tune the neural network-based controller. From the literature survey, it is evident that metaheuristic algorithm-based controllers have the advantage of having wider space to search solutions and they are less sensitive to initial conditions and objective function constraints. Therefore, instead of using a backpropagation algorithm, it is better to use metaheuristic algorithms.

In this chapter, a simple PID-like neural network-based controller is proposed whose gains are optimized by the PSO algorithm. PSO algorithm has proved itself to be one of the most successful optimization techniques because of the various advantages it offers like having fewer parameters and providing a global optimal solution. The proposed controller is simple and easy to implement as compared to controllers proposed in the literature till now.

6.2 Mathematical Modeling of the CSTR system

Continuous Stirred Tank Reactor (CSTR) is a reactor mostly used in chemical industries[195]. It is generally used in processes where a continuous flow of products is required to obtain a product[92]. Chemical reactors have a lot of effects on the output due to heat, so it is important to control heat in a chemical reactor. Since, in a CSTR continuous stirring takes place therefore, it is assumed that the products are perfectly mixed. A perfect mixing assumption means that there is uniform concentration maintained, there are no dead zones present and rapid mixing is done continuously. Another assumption considered in the study is constant reactor volume is maintained, which means there is continuous flow of reactant and extraction of product. In a jacketed CSTR temperature control can be done by controlling the temperature of the jacket around the reactor. The jacket temperature can be controlled by a coolant flowing in it. It operates with the following assumptions[91]:

1. The reactor volume is kept constant V_R.
2. It is operated under steady-state conditions with perfect mixing

In the CSTR considered we consider a simple reaction $A \xrightarrow{yields} B$. The concentration of the feed flown initially of product A is donated as C_{AR0}, the initial temperature of product A is T_{R0} and it is assumed that there is a constant flow rate q_R. Irreversible reactions take place inside the reactor. The final products generated have a concentration C_{AR} and temperature T_R. The heat produced in the exothermic reaction is controlled by flowing a coolant in the jacketed layer of the reactor. The coolant temperature is T_{CRO} and its flow rate is q_{CR}. The material balance of jacketed CSTR can be given as[91]:

$$\frac{dV_o\rho_o}{dt} = q_{in}\rho_{in} - q_{out}\rho_{out} \tag{6.2}$$

Where, V_o is the reactor volume, ρ_o is the density of the reactor, ρ_{in} is the density of feed flow input, q_{in} and q_{out} are the flow rates of input and output feed. In this case, as per the assumptions $\rho_{in} = \rho_{out} = \rho_o$. The flow rates are also equal.

The balance equation for component A is given as[91],

$$V_o\frac{dC_{AR}}{dt} = q_R(C_{AR0} - C_{AR})V_R - V_o r_A \tag{6.3}$$

Where, C_{AR} is the concentration of component A in the reactor, r_A is the rate of reaction and is given as[91],

$$r_A = K_{R0}\exp\left(\frac{-E_R}{RT_R}\right)C_{AR} \tag{6.4}$$

where, K_{R0} is the frequency factor, E_R is the activation energy and T_R is the reactor temperature. The energy balance equations for the reactor considering above-given assumptions and neglecting kinetic and potential energy are given as[91],

$$V_R\rho_R C_P\frac{dT_R}{dt} = q_R\rho_R C_P(T_R - T) - (-\Delta H)V_R r_A + \rho_{CR}C_{PC}q_{CR}$$

$$\left[1 - \exp\left(\frac{-hA}{\rho_{CR}C_{PC}q_{CR}}\right)\right](T_R - T) \tag{6.5}$$

In Equation 6.3 RHS represents heat accumulation in the reactor, in LHS the first term represents energy in and out due to flow of component A, the second term represents the heat due to reaction in the reactor and the third term represents heat transferred to the jacket. Where, T_R is the reactor temperature and T is the jacket temperature in Kelvin.

The above equations can be presented in the state variable form choosing C_{AR} and T_R as state variables,

$$\frac{dC_{AR}}{dt} = \frac{q_R}{V_R}(C_{AR0} - C_{AR}) - K_{R0}\exp\left(\frac{-E_R}{RT_R}\right)C_{AR} \tag{6.6}$$

$$\frac{dT_R}{dt} = \frac{q_R}{V_R}((T_{RO} - T_R) - \left(\frac{-\Delta H}{\rho_R C_P}\right)K_{R0}\exp\left(\frac{-E_R}{RT_R}\right)C_{AR} + \left(\frac{\rho_{CR}C_{PC}}{\rho_R V_R C_P}\right)q_{CR}$$

$$\left[1 - \exp\left(\frac{-hA}{\rho_{CR}C_{PC}q_{CR}}\right)\right](T_{RO} - T_R) \tag{6.7}$$

The steady-state solutions can be obtained by putting the two equations 5 & 6 to zero. For this study, the parameters of the jacketed CSTR chosen are shown in Table 6.1.

Table 6.1 Parameters of Jacketed CSTR[92]

Parameters	**Values**
C_{AR} Concentration of input feed component A	0.0882 mol/l
Flow rate of the coolant q_{cR}	100 l/min
Reactor Temperature	442 K
Feed flow rate q_R	100 l/min
Input Feed temperature T_O	350 K
Jacket temperature, T	350 K
The volume of reactor V_R	100 l
Coefficient of heat transfer hA	$7\times 10^5\ cal/(minK)$
Reaction rate constant K_R	7.2×10^{10} min-1
Activation Energy $\frac{E_R}{R}$	1×10^4 K
The heat produced in the Reaction (ΔH)	-2× $10^5\ cal/mol$
C_H, C_{HC} Specific heat	1 Cal (g/K)
Densities, $\rho_{R,}\rho_{CR}$	$1\times 10^3 g/l$

6.3 Proposed Structure of NN-Based PID Controller

The structure of the proposed NN-based PID Controller consists of one input layer, one hidden layer, and an output layer as shown in Figure 6.2. There is a single node in the input layer, three nodes in the hidden layer, and one node in the output layer. Therefore, the proposed NN-PID structure is a 1-3-1 structure. The three neurons of the hidden layer $a_1, a_2\ and\ a_3$ correspond to integral, proportional, and derivative control actions of the PID controller. The activation functions, f used are all linear. To produce the integral action the integral node a_1 is taken as feedback to n_1 by using the me delaying effect (z^{-1}). To produce the derivative effect the derivative node n_3 is given negative feedback. The proportional node a_2 is the general node without any feedback.

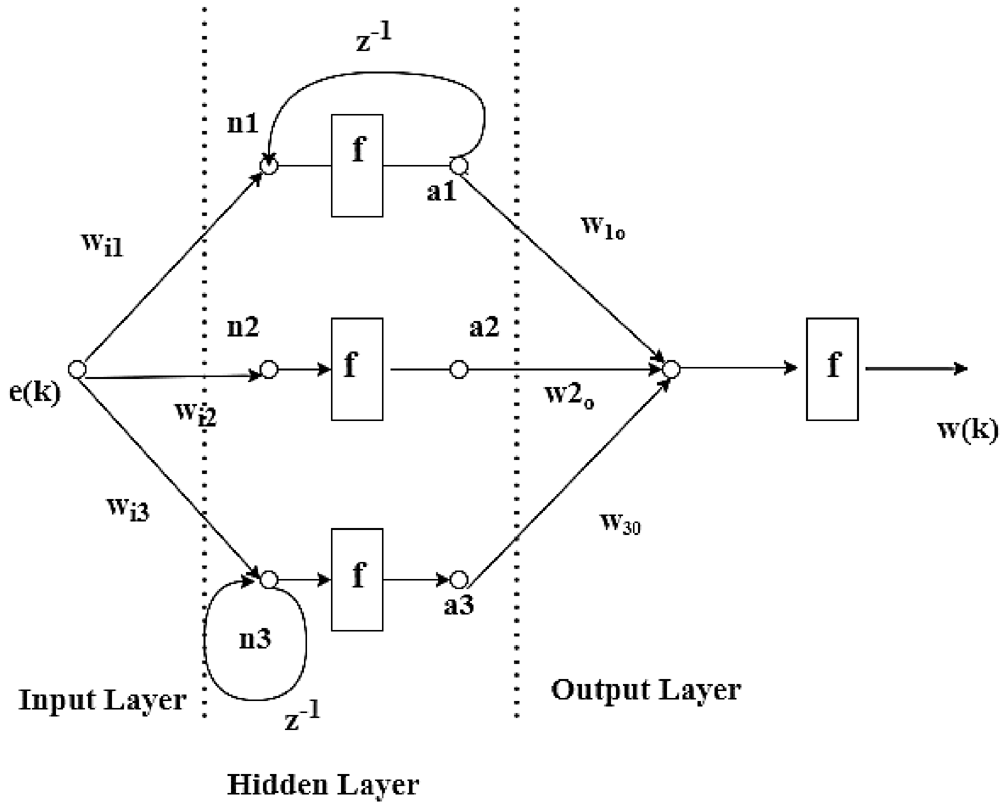

Figure 6.1 Structure of Proposed NN-PID Controller

As seen in the structure shown in Figure 6.2, $e(k)$ is the input data to each node which is the error signal i.e. the difference between the reference step input and the actual output of the system. The outputs of the three hidden layers a_1, a_2 and a_3 at sampling times k may be given as,

$$a_1(k) = e(k)w_{i1}(k) + a_1(k-1) \tag{6.8}$$

$$a_2(k) = e(k)w_{i2}(k) \tag{6.9}$$

$$a_3(k) = e(k)w_{i3}(k) - w_{i3}(k-1)e(k-1) \tag{6.10}$$

Where, $w_{i1}(k)$ represents the weight connecting the input neuron to the first hidden node, $w_{i2}(k)$ represents the weight connecting the input neuron to the second hidden node and $w_{i3}(k)$ represents the weight connecting the input neuron to the third hidden node. The output of the proposed neural network-based controller is,

$$u(k) = \sum_{j=1}^{3} w_{jo}(k)a_j(k) = w_{1o}(k)a_1(k) + w_{2o}(k)a_2(k) + w_{30}(k)a_3(k) \tag{6.11}$$

In the first node z^{-1} is added as delay. So, $a(k-1) = a(k)z^{-1}$

Then, $e(k-1) = z^{-1}e(k)$ (6.12)

The output of the first hidden layer is

$$a_1(k) = \frac{w_{i1}(k)e(k)}{1-z^{-1}} \tag{6.13}$$

An integral relationship is represented by this equation. As a result, this node produces a gain equal to the error's integral. Activation feedback was used by the derivative node to generate the derivative action.

$$a_3(k) = w_{i3}(k)e(k)[1 - z^{-1}] \tag{6.14}$$

A differential mode of operation is represented by this equation. The control output u(k) clearly shows that the neural network-based controller generates output similar to that of a PID controller. The neural network's first node generates output similar to an integral action, the second node generates action similar to a proportional control, and the third node generates output similar to a derivative control. The PID NN that is being proposed here has a simple structure, with its output layers producing the sum or controller output, as opposed to the typical BP-based NN, which is complex and has multiple layers. Thus, we can simply modify the PID parameters to achieve the desired results by training the weights of the PID-based neural network.

6.4 Proposed PID Like Neural Network tuning by PSO

The study proposes an innovative PSO-based PID NN tuning technique. PSO is still one of the most employed algorithms for solving many types of non-linear problems, even after the development of several metaheuristic algorithms. The reason for the selection of the PSO algorithm in this study is that it is simple to implement, as it has fewer parameters as compared to other complex metaheuristic algorithms developed. It has a faster convergence rate. Another advantage of PSO is that there are fewer chances of having a locally optimal solution as compared to other metaheuristic algorithms. The block diagram explaining the function of PSO-NN-PID is shown in Figure 6.4.

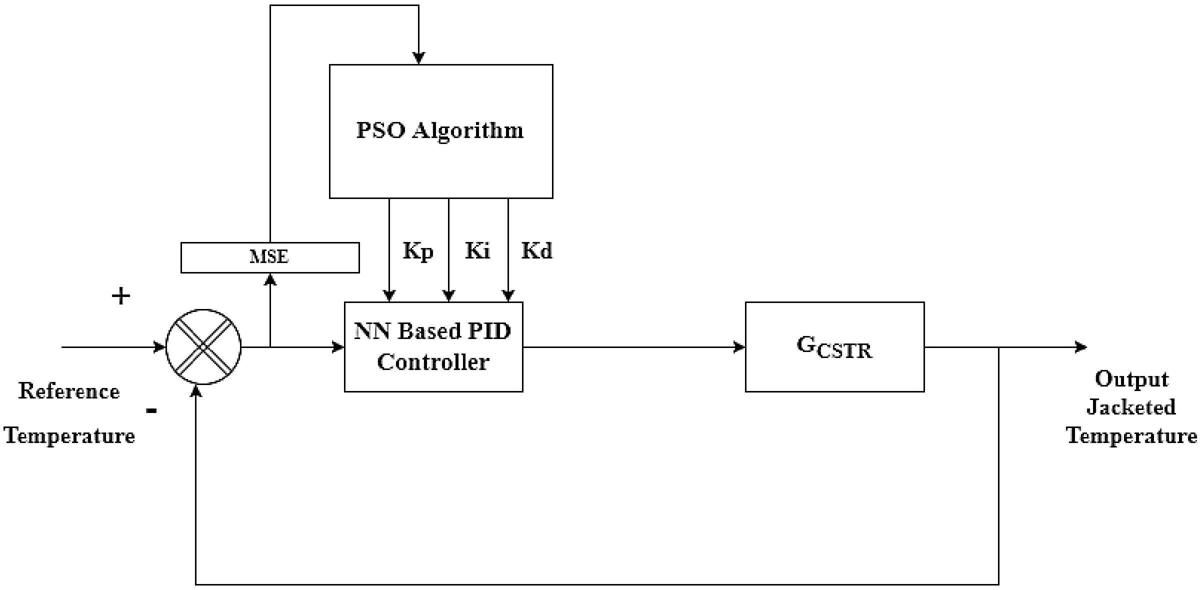

Figure 6.2 Block Diagram of PSO-based novel NN-PID controller

It can be seen in Figure 6.4 that the NN-like PID controller parameters are tuned by PSO. The objective function used is mean square error (MSE). The tuning of weights by the PSO algorithm is done to minimize the objective function MSE. The error $e(k)$ is the difference between temperature output $t(k)$ and the reference step input r(k).

The cost function is calculated after each iteration. The objective of the controller is to minimize the cost function:

$$E_c = \frac{1}{2}\sum_{k=1}^{T}(r(k) - y_{act}(k))^2 \qquad (6.15)$$

Where, $r(k)$ is the desired reference input and $y_{act}(k)$ is the actual output.

The PSO-tuned NN-based PID controller then gives the control signal to the Jacketed CSTR for temperature control. The algorithm of the proposed PSO-tuned NN-PID Controller is described below:

6.4.1 Algorithm of the proposed PSO-tuned NN-PID Controller

1. Define PID as a neural network having one neuron in the input layer, three neurons in the hidden layer, and one neuron in the output layer.
2. Initialize random weights within the range [-1,1].
3. PSO is initialized with dimension size 3, population size 25, and a maximum number of iterations of 50.
4. The fitness function is chosen as the Mean square error.
5. For all particles do

6. Calculation of the control law $u(k)$ by initial weights.
7. Calculation of output $y_i(k)$
8. Evaluate the current position and velocity in the search space
9. Calculate the value of the objective function for the current iteration
10. If the current position gives the best objective function, then,
11. $P_{abesti} = w_i^k$
12. Else If the current objective function is the best overall objective function, then,
13. $G_{abesti} = w_i^k$
14. Endif
15. Move the particles in the search space
16. Update the position and velocity of the particles,
17. Do until the stopping criteria are met
18. End

6.5 Back propagation algorithm for tuning NN-PID controller

The objective of the controller is to minimize the error between the actual output and the desired reference input. The output error can be written mathematically as,

$$error_i(k) = r_i(k) - yout_i(k) \tag{6.16}$$

Where, i = 1,2,….S and $yout_i(k)$ are the output variables measured and $r_i(k)$ is the reference input. The fitness function MSE is given as,

$$MSE(k) = \sum_{i=1}^{s} MSE_i(k) \tag{6.17}$$

Now, we state the rules for updating the weights of the neural network controller

$$\frac{\partial MSE(k)}{\partial w_{ij}(k-1)} = \sum_{i=1}^{S} \left[\frac{\partial MSE_i(k)}{\partial yout_i(k)} . \frac{\partial yout_i(k)}{\partial u(k-1)}\right] \frac{\partial u(k-1)}{\partial w_{ij}(k-1)} \tag{6.18}$$

$$\frac{\partial yout_i(k)}{\partial u(k-1)} \approx \frac{\Delta yout_i(k)}{\Delta u(k-1)} \approx \frac{yout_i(k) - yout_i(k-1)}{u(k-1) - u(k-2)} \tag{6.19}$$

We use the sign function to find the result. The final rule of updating weights is,

$$\Delta w_{ij}(k-1) = -\alpha_{rij} \frac{\partial MSE(k)}{\partial w_{ij}(k-1)} = \alpha_{rij} \sum_{i=1}^{s} error_i(k). sign \frac{\Delta yout_i(k)}{\Delta u(k-1)} . error_i(k-1) \tag{6.20}$$

Where, j=1,2,3

The weights of the output layer are,

$$w_{oi} = 1, i = 1,2,3 \tag{6.21}$$

The function of the back propagation-based NN PID controller is shown in Figure 6.

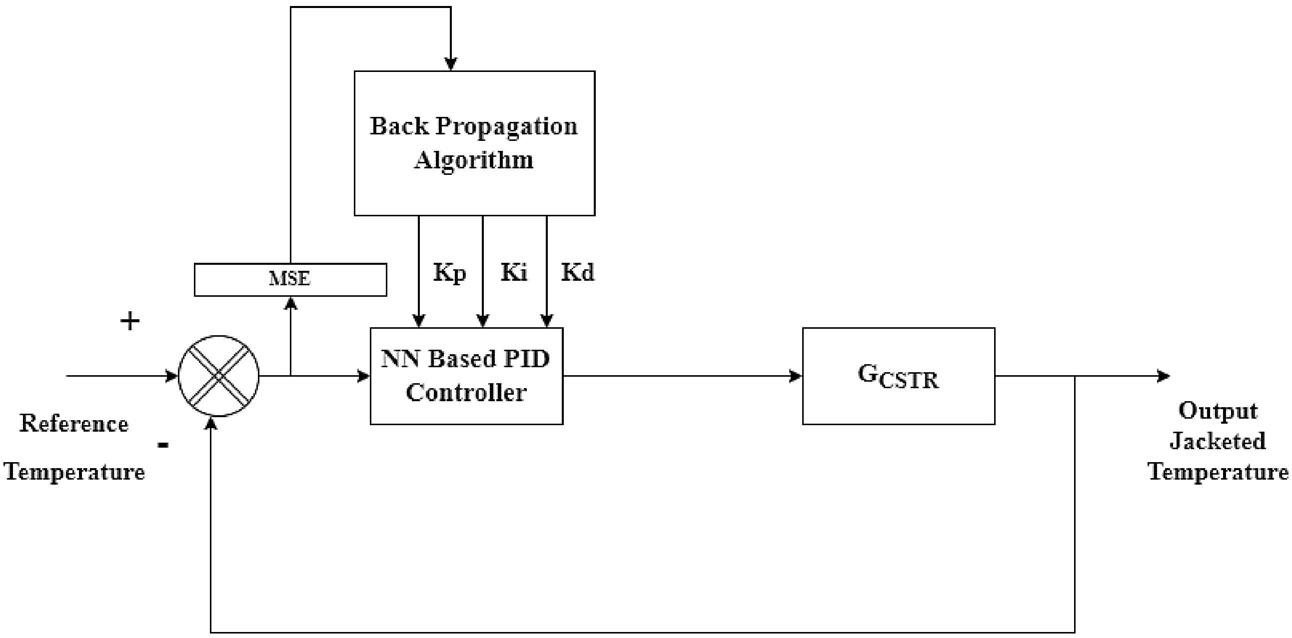

Figure 6.3 Block Diagram of BP-NN-based PID controller

The BP-based NN-like PID controller shown in Figure 6.5 gives the optimized PID parameters. The NN-based PID controller gives the control signal to control the temperature of jacketed CSTR.

6.6 Simulation Results

To check the effectiveness and viability of the proposed PSO-based NN-PID controller a comparative simulation study has been performed with a BP-tuned NN-PID controller and ZN-tuned PID controller to control the temperature of CSTR. The dynamic equations for a non-linear CSTR are developed in equations 5 & 6. The temperature of the CSTR is selected as the controlled variable. From equation 6 difference equation can be calculated and reference input is taken as the step input. The data for the proposed tracking control is obtained CSTR temperature control difference equation and the input data is a step signal. The simulation studies are performed on MATLAB 2018 software. The simulations are performed on a PC with an Intel Core i3 processor with a speed of 2.10GHz and a RAM of 8.00 GB. The parameters chosen while applying the PSO algorithm are presented in Table 6.2.

Table 6.2 Parameters Selected for PSO

S.no.	Parameters	Values
1.	$C_1 \& C_2$ Acceleration Constants	2
2.	w_{max}	0.9
3.	w_{min}	0.4
4.	Maximum no. of Iterations	50

The comparative step responses for various controllers for temperature control of jacketed CSTR are given in Figure 6.6.

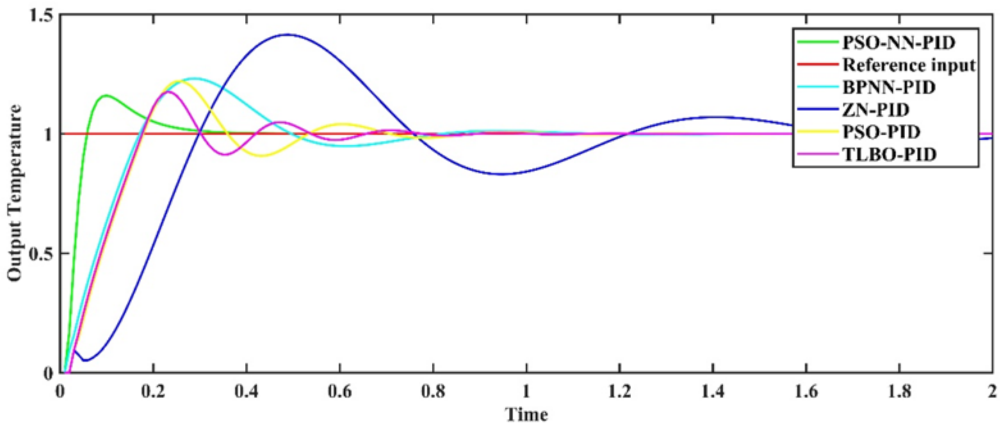

Figure 6.4 Comparative Step Responses of Various Controllers

The performance indices explaining the step responses quantitatively are presented in Table 6.3.

Table 6.3 Comparative Step performance indices of various Controllers

S.no.	**Method**	**Rise time ,t_r seconds**	**Settling time, t_s seconds**	**Max. Overshoot (% M_p)**	**Peak time t_p, (seconds)**
1.	Zeigler Nichols	0.1810	44.08	0.4900	1.6731
2.	BPNN-PID	0.2727	26.33	0.6200	1.8194
3.	PSO-NN-PID	**0.1283**	**23.13**	**0.2900**	**0.7480**
4.	PSO-PID	0.1565	25.64	0.4233	0.9073
5.	TLBO-PID	0.1465	24.63	0.3562	0.8989

It is evident from Figure 6.6 and Table 6.3 that rise time, tr is lowest at 0.1283 in the case of PSO-NN-PID tuned controller as compared to BPNN-PID, PSO-PID and TLBO-PID and ZN tuned PID. The maximum overshoot has also reduced from 44.08% in ZN-tuned PID to 26.33% in BPNN-PID and has further reduced to 23.13% in PSO-NN-PID. The PSO-PID and TLBO-PID also have an overshoot slightly higher than the proposed method. The settling time is also lowest at 0.7480 in the case of the PSO-NN-PID controller. The values of different optimized PID gains K_p, $K_i$and K_d are given in Table 6.4.

Table 6.4 K_p, K_i and K_d parameters of various Controllers

S.no.	Method Used	K_p	K_i	K_d
1.	ZN Tuned	4.0795	0.8703	12.019
2.	PSO-NN-PID	**9.9000**	**2.0272**	**9.4000**
3.	BPNN-PID	4.3300	0.3100	6.1100
4.	PSO-PID	10	8.75	10
5.	TLBO-PID	8.3940	4.7184	2.0562

The proposed controller is also tested for disturbance rejection. The comparative step responses of different controllers in case of disturbance application from time = 1 second to time = 1.75 seconds are presented in Figure 6.7. The performance indices of the responses presented in Figure 6.5 are quantitatively compared in Table 6.5.

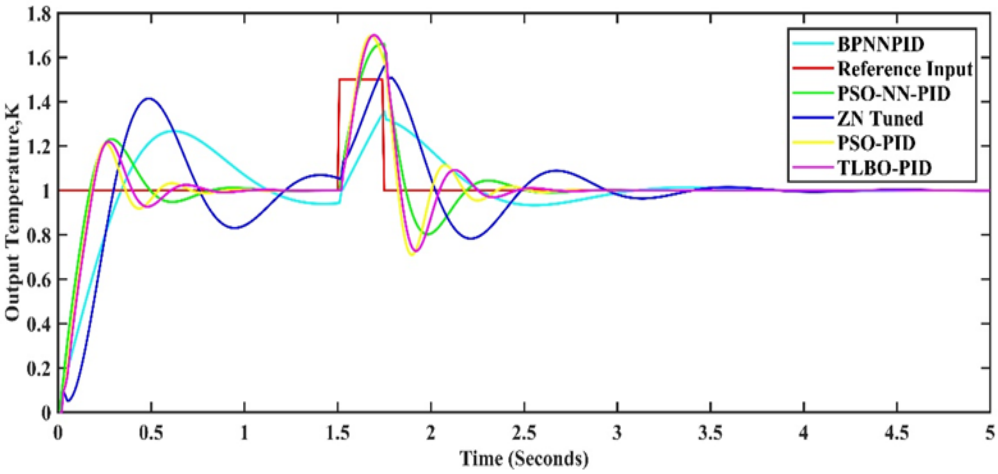

Figure 6.5 Comparative Step responses of various controllers under disturbance application

Table 6.5 Comparative Step performance indices of various controllers under disturbance application

S.no.	Method	Rise time, tr seconds	Settling time, t_s seconds	Max. Overshoot (% M_p)	Peak time t_p, (seconds)
1.	Zeigler Nichols	0.1839	47.03	1.21	2.77
2.	BPNN-PID	0.2713	30.00	1.21	2.36
3.	PSO-NN-PID	**0.1289**	**50.00**	**1.20**	**1.88**
4.	PSO-PID	0.2715	59.94	1.65	2.79
5.	TLBO-PID	0.1873	68.85	1.91	3.79

Table 6.5 shows that the PSO-NN-PID has the best rise time of 0.1289 seconds as compared to 0.2713 seconds in BPNN-PID, 0.1839 seconds in ZN-PID, 0.2715 seconds in PSO-PID and 0.1873 seconds in case of TLBO-PID. There is a slight increase in maximum overshoot in the case of the PSO-NN-PID controller but if we compare the settling time, it shows that the controller efficiently reduces the effect of disturbance quickly. The settling time is lowest at 1.88 seconds in the case of the PSO-NN-PID controller. The values of controller gains are given in Table 6.6.

Table 0.6 K_P, K_i and K_d parameters of various Controllers under disturbance application

S.no.	Method Used	K_P	K_i	K_d
1.	ZN Tuned	4.0795	0.8703	12.019
2.	PSO-NN-PID	9.9000	1.9983	9.4000
3.	BPNN-PID	4.2145	0.2909	6.2350
4.	PSO-PID	8.8566	2.4566	8.8564
5.	TLBO-PID	1.3969	0.3564	6.2345

The comparative analysis of the performance indices is shown in Figure 6.8.

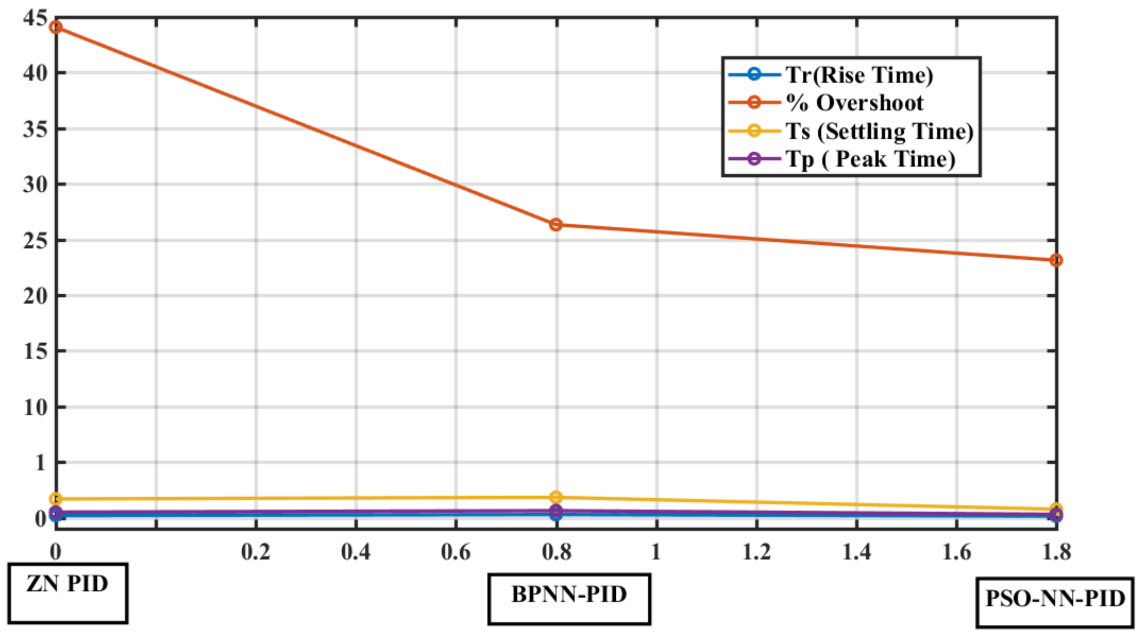

Figure 6.6 Comparative graph of rise time, overshoot, peak time, and settling time for various controllers

The control signal u(t) and error signal e(t) are shown in Figures 6.9 & 6.10.

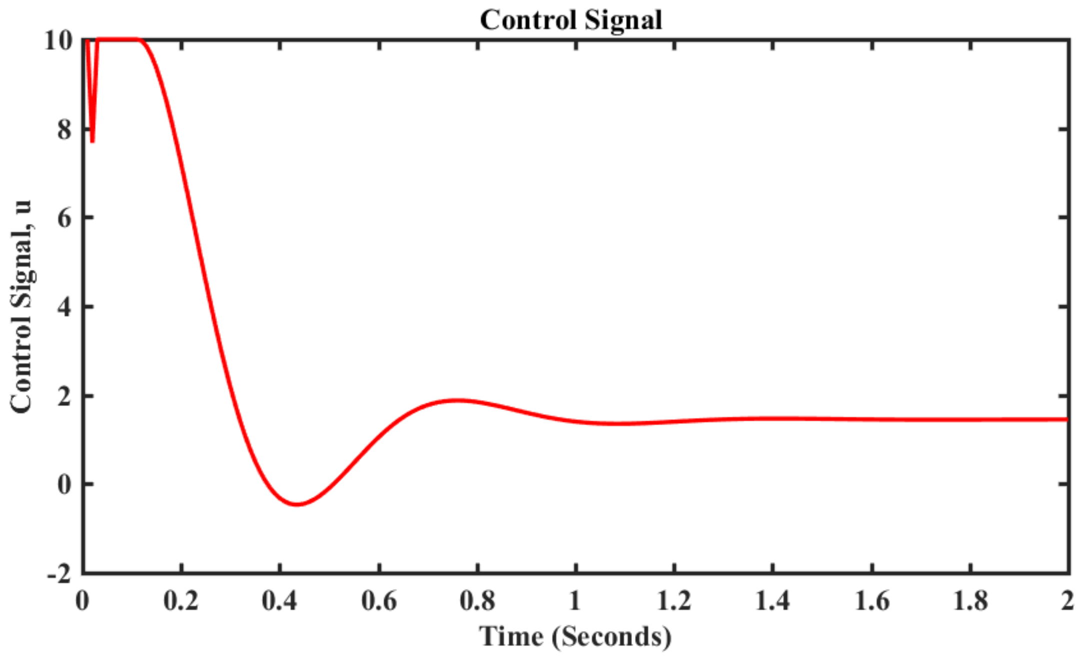

Figure 6.7 Control Signal in case of PSO-NN-PID controller

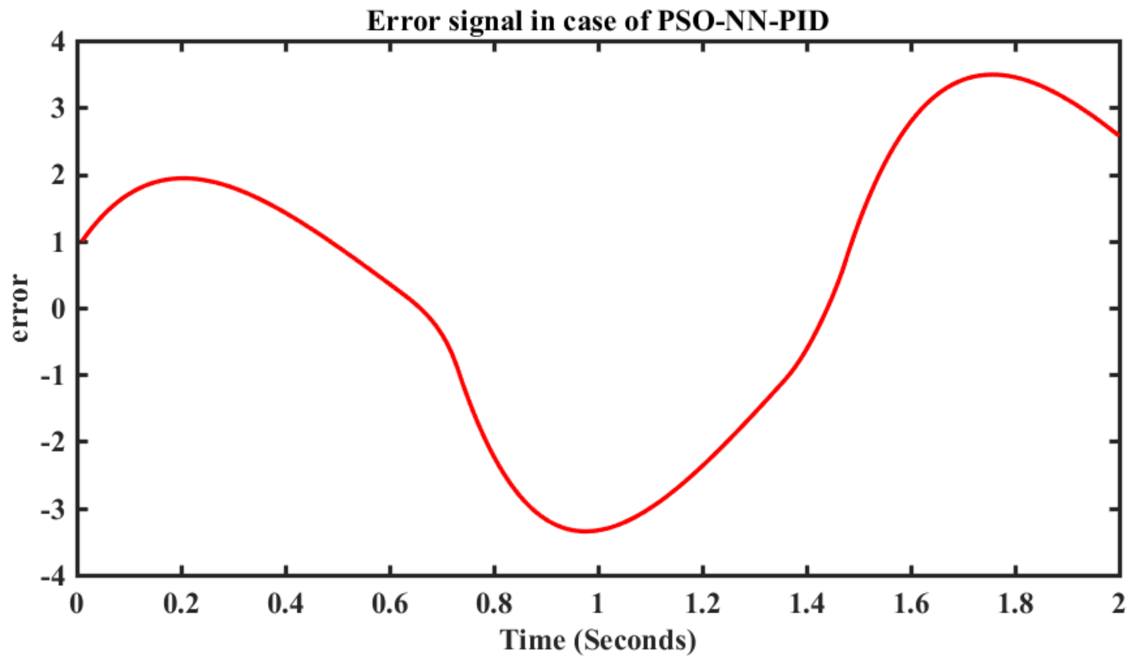

Figure 6.8 Error Signal in case of PSO-NN-PID controller

The graph showing variation of Kp, Ki and Kd with respect to time is shown in Figure 6.11.

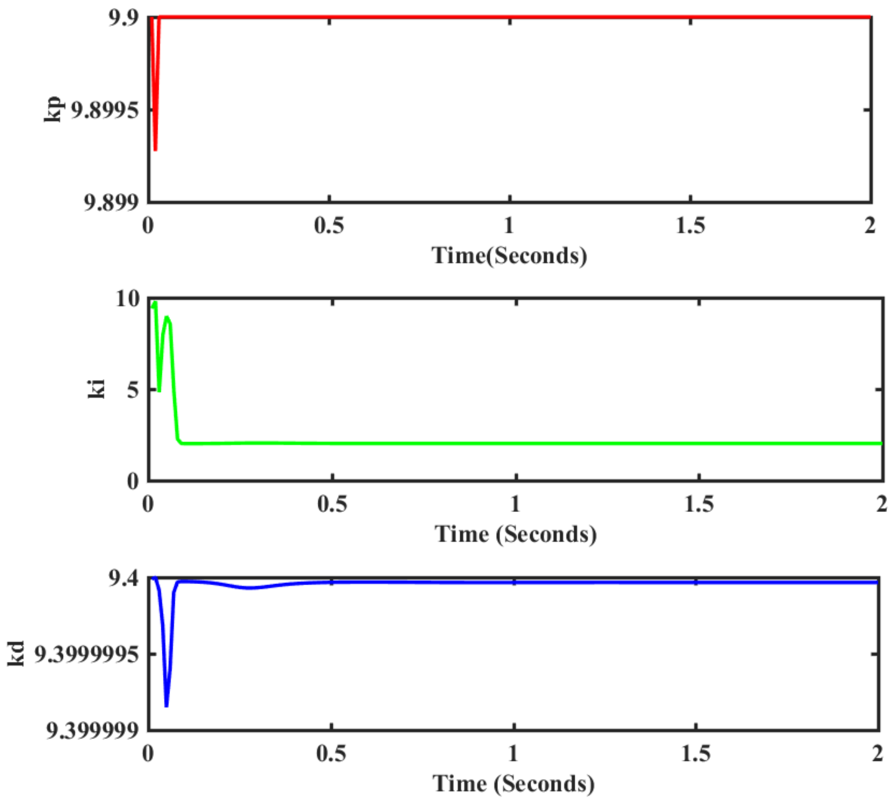

Figure 6.9 Variation of K_P, K_i and K_d values with time

The comparative bar graph of MSE in the case of three controllers, ZN-PID, BPNN-PID, and PSO-NN-PID is shown in Figure 6.12.

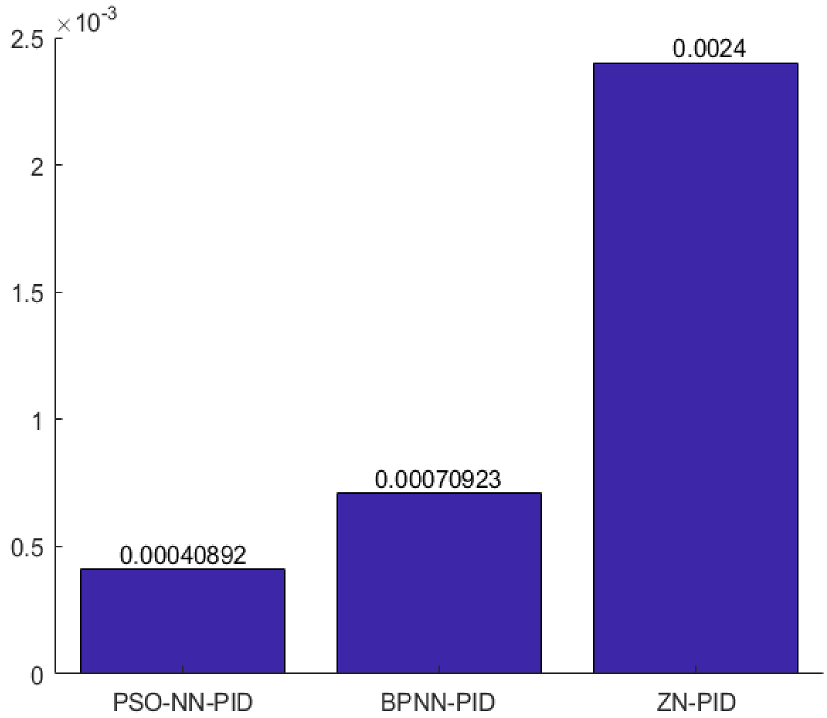

Figure 6.10 Comparative bar graph of MSE values for various controllers

Table 6.7 Mean Square Error for Different Controllers

S.no.	**Controller Name**	**MSE**
1.	BP-NN-PID controller	0.00070923
2.	**PSO-NN-PID controller**	**0.00040892**
3.	ZN-PID controller	0.0024
4.	Deep learning neural network model predictive control [36]	0.0357
5.	DGWO-Fuzzy WNN model [28]	0.0065

Table 6.8 Integral Square Error for Different Controllers

S.no.	Controller Name	ISE
1.	ZN-PID Controller	0.4561
2.	BP-NN-PID controller	11.0025e-5
3.	**PSO-NN-PID Controller**	**9.0012e-5**

6.7 Discussion

A comparison of the simulation results of the three controllers is presented in the above section. ZN-tuned PID, BP-tuned PID-NN, and PSO-tuned PID-NN are compared for temperature control on a non-linear jacketed CSTR plant. Table 6.3 presents the comparative step performance indices of the proposed controller with a BP-tuned NN PID controller and a conventional ZN-tuned PID controller. It is clear from the comparative study that the proposed controller gives a faster response as compared to the BP-NN-PID controller and the ZN-tuned PID controller. The rise time was 0.2727 seconds for the BP-NN-PID controller, and it is about 0.1283 seconds for the proposed PSO-tuned PID controller. The proposed controller is also effective in reducing the overshoot significantly. The overshoot of the response was about 44.08% for a ZN-tuned PID controller, which reduced to 26.33% in a BP-tuned PID controller and further reduced to 23.13% in a PSO-tuned NN-PID controller. Similarly, a significant improvement can be observed in peak time and settling time with the proposed PSO-tuned NN-PID controller. The optimized values of K_p, K_i and K_d in the case of the proposed controller are 9.900, 2.0272, and 9.400, respectively. To test the robustness of the proposed scheme a disturbance signal of amplitude 1.5 was applied for a time period of 1 second. In the case of PSO tuned PID-NN controller there is a slight increase in overshoot but it can be observed that the controller was able to subside the disturbance quickly in 1.88 seconds as compared to ZN-tuned PID taking 2.77 seconds and BP-tuned PID-NN taking 2.36 seconds. Figure 6.9 shows a comparative graph of rise time, peak time, overshoot, peak time, and settling time. It can be seen from the graph that there is a reduction in rise time, overshoot, and settling time in the case of PSO-PID-NN. Figures 6.9 & 6.10 represent the variation of the control signal and error signal with time in the case of PSO-PID-NN. Figure 6.11 represents the variation of $K_p, K_i\ and\ K_d$ in the case of PSO-PID-NN. The values of K_p, K_i and K_d stabilize around 9.900, 1.983, and 9.400 respectively.

Figure 6.12 shows a comparative bar graph of different controllers. The value of cost function MSE is least in the case of PSO-PID-NN and largest in the case of ZN-tuned PID. Table 8 shows a comparison of integral square error for different controllers. It is observed from the comparison that the proposed controller has a minimum integral square error of 9.0012e-5. The computational time of the proposed PSO-tuned NN-PID controller is 0.7546 seconds which is slightly larger than BP BP-based NN-PID controller with a computational time of 0.5689 seconds. However, since the accuracy and disturbance rejection of the proposed PSO-based NN-PID controller is good, a slightly higher computational time can be ignored. Comparing the BP-tuned NN-PID controller and PSO-based NN-PID controller, we can say that the PSO-based NN-PID controller performs better as compared to the BP-tuned NN-PID controller. The reason is that backpropagation performance depends on the correct initial value selection and gradient descent which causes it to fall mostly in local optimal solutions. PSO is a metaheuristic algorithm that has been applied effectively to various complex problems. As compared to backpropagation PSO has the advantage that it searches a large search space for feasible solutions, therefore there are fewer chances of the solution falling in the local optima. The global best solutions found by swarms provide better solutions with MSE as an objective function. Therefore, PSO-based NN-PID controller provides better solutions as compared to BP-based NN-PID controllers in terms of robustness, speed, and adaptability.

6.8 Comparison of the proposed controller with previous studies in Literature

From Table 6.7 it can be concluded that the proposed controller has a Mean square error of 0.00040892 which is the smallest as compared to other controllers proposed recently in the literature [28], [36]. The studies done in the past to control CSTR reactors have different objectives. But they can be compared because they were used to control the reactor. In the past many researchers have applied different controllers like conventionally tuned PID controllers, fuzzy logic-based PID controllers, neural network-based adaptive controllers, and metaheuristic-based PID controllers to control CSTR concentration and temperature [27], [29]–[35], [37]. The limitations of these studies are that the controllers proposed till now had complex structures, more computational time, large steady-state error, higher rise time, and settling time, some of the controllers resulted in falling into local optimal solutions instead of global solutions. Therefore, to reduce the limitations the proposed PSO-tuned NN-PID controller gives a very simple structure having only three neurons in the hidden layer which reduces complexity. The simple structure of NN and optimization of weights by PSO give a faster convergence. The proposed controller gives a smaller overshoot, rise time, and settling time. The hybridization of PSO with NN ensures the attainment of the global optimal solution.

6.9 Conclusion

The study aims to overcome the limitations of backpropagation-tuned PID controllers and conventional ZN-tuned PID controllers when applied to a non-linear system. The novel neural network structure proposed as a PID controller is very simple, therefore it reduces system complexity and gives higher accuracy. The proposed controller is tested by controlling the temperature of a non-linear jacketed CSTR. The major findings of the proposed study are as follows:

1. The proposed PSO-tuned NN-PID controller is proved to be better in terms of rise time, overshoot, peak time, and settling time as compared to the backpropagation-tuned NN-PID controller, metaheuristic algorithms tuned PSO-PID, TLBO-PID, and conventional ZN-tuned PID controller.
2. The proposed controller is simple in structure. Therefore, reduces complexity as compared to structures proposed earlier in the literature. Also, the proposed controller gives better accuracy as compared to BP-NN-PID and ZN-PID.
3. The proposed controller was effective in disturbance reduction efficiently as compared to the backpropagation-tuned NN-PID controller, PSO-PID, TLBO-PID, and ZN-tuned PID controller. It was able to reject the disturbance faster as compared to other controllers.

The comparison of MSE values from different controllers from the literature proves the effectiveness of the proposed PSO-tuned NN-PID controller

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Conclusion

In this thesis, a detailed investigation was carried out about the complexities of PID tuning when applied to a non-linear system. It explored how the performance in complex, dynamic non-linear systems can be improved by integrating the areas of control systems, artificial intelligence, and optimization techniques. After getting an insight into how traditional tuning methods fail for the PID controllers applied to non-linear systems. We shifted our focus towards the integration of control and AI for the specific problem. Some intelligent methods are proposed in this work for PID tuning of different bench-mark non-linear systems. In the thesis, applications of some prominent metaheuristic algorithms have been explored like PSO, TLBO, and WOA algorithms. Each of these algorithms was studied and applied to the specific non-linear system PID tuning. These algorithms brought success and adaptability to the applied non-linear problems. The application of neural networks is also studied and was found that it can provide better adaptability to the system's non-linearities and uncertainties. Therefore, a hybrid approach is proposed in the thesis combining neural network and PSO algorithm to tune PID controller applied to a benchmark non-linear system. A novel PID controller like neural network is proposed in the thesis which is simple in structure and easy to implement. An extensive study is carried out for each of the proposed controllers. Their performance indices and error indices were compared with the traditional tuning methods and some intelligent controllers proposed recently in the literature.

In Chapter 3, we designed an optimized PID controller by using TLBO and PSO algorithms and applied them to automobile cruise control systems and artificial respiratory systems. The controllers were analysed based on performance indices of the output responses and error indices obtained. The results were also compared with the conventional Zeigler-Nichols tuning method and a recently proposed fuzzy PD plus I controller. The proposed controller was also tested for robustness in the presence of uncertainty and the results by changing the system parameter were found within permissible limits.

In Chapter 4, a cascade-optimized PID controller is proposed for a fourth-order ball and beam system. The two cascade PID controllers were optimized by TLBO and PSO algorithm and the results were compared with classical tuning methods. The proposed controllers were also tested for robustness under the presence of a disturbance signal. The TLBO controller performed better as compared to other controllers, as it was able to settle the disturbance within a minimum time.

In Chapter 5, an optimized Fractional-order PID controller is proposed whose parameters are tuned by three metaheuristic algorithms, TLBO, PSO, and a new whale optimization algorithm. The proposed optimized Fractional-order PID controllers were compared with the previously proposed TLBO-tuned PID controller, PSO-tuned PID controller, and whale optimization-tuned PID controller in terms of output response and error indices. The fractional order PID controllers performed better as compared to PID controllers.

In Chapter 6, a novel PID-like neural network was proposed whose weights were optimized by the PSO algorithm. The efficacy of the proposed method was the simple structure of the NN-PID controller proposed. The proposed method was applied for temperature control of a CSTR system. It was also tested for robustness under the presence of a disturbance signal.

7.2 Major Contributions of the Present Thesis

1. Novel intelligent PID optimization methods: In this thesis, some novel PID optimization methods are proposed for non-linear systems by applying the power of optimization algorithms and neural networks.
2. Highlighting the advantages and disadvantages of various intelligence techniques: In this work, we have given a deep insight into the advantages and weaknesses of different intelligent methods used for PID tuning. Like, optimization algorithms, fuzzy logic, and neural networks. Therefore, thesis work can guide engineers in selecting a particular intelligent method for their specific application.
3. Study of different bench-mark non-linear systems: In this work, we have developed the mathematical models of some bench-mark non-linear systems like a continuous stirred tank reactor, ball and beam system, inverted-pendulum system, and artificial respiratory system.
4. Applications to real-world systems: The thesis provides insight into the proposed methods for real-world problems ranging from automobile cruise control systems, artificial respiratory systems in health care, and CSTR systems in the process industry,

to popular bench-mark non-linear systems like ball and beam systems and inverted pendulum-cart system.

7.3 Limitations and Future Directions

While the present work has given various valuable insights into the efficient PID tuning of non-linear systems and has contributed to the field significantly. But the following limitations are enlisted of the present work which could be the future research directions:

1. Study of complex non-linearities: Non-linear systems have a big span of complexity. Therefore, future research can explore more deeply the different non-linear behaviors and understand specialized intelligent techniques for the specific non-linear complexity.
2. Real-time implementation: The implementation of the proposed intelligent PID tuning methods into real-world systems is a challenge. Further research can be done on the constraints during the real-time implementation of the proposed methods.
3. Robustness: The testing of the proposed methods for different uncertainties and non-linearities is another prospect of research.
4. Implementation of the proposed methods for MIMO systems: The proposed intelligent methods can be also tested for MIMO systems and compared with the conventional methods.
5. Extension of the proposed intelligent methods for plant identification: The methods applied in the present work are all based on known plant dynamics. Therefore, future research directions can be toward unknown plant identification and control.

REFERENCES

[1] C. Zhao and L. Guo, "Control of Nonlinear Uncertain Systems by Extended PID," vol. 9286, no. c, pp. 1–8, 2020, doi: 10.1109/TAC.2020.3030876.

[2] S. Bennett, "Development of the PID Controller," *IEEE Control Syst.*, vol. 13, no. 6, pp. 58–62, 1993, doi: 10.1109/37.248006.

[3] D. P. Atherton, M. Benouartes, and O. Nanka-Bruce, "Design of Nonlinear PID Controllers for Nonlinear Plants," *IFAC Proc. Vol.*, vol. 26, no. 2, pp. 125–128, 1993, doi: 10.1016/s1474-6670(17)49090-2.

[4] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 115, no. 2B, pp. 220–222, 1993, doi: 10.1115/1.2899060.

[5] "Cohen and Coon (1953) - Theoretical Consideration of Retarded Control." .

[6] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, 1984, doi: 10.1016/0005-1098(84)90014-1.

[7] R. C. Panda, C. C. Yu, and H. P. Huang, "PID tuning rules for SOPDT systems: Review and some new results," *ISA Trans.*, vol. 43, no. 2, pp. 283–295, 2004, doi: 10.1016/s0019-0578(07)60037-8.

[8] Q. G. Wang, T. H. Lee, H. W. Fung, Q. Bi, and Y. Zhang, "PID tuning for improved performance," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 4, pp. 457–465, 1999, doi: 10.1109/87.772161.

[9] M. Veronesi and A. Visioli, "On the selection of lambda in lambda tuning for PI(D) controllers," in *IFAC-PapersOnLine*, 2020, vol. 53, no. 2, doi: 10.1016/j.ifacol.2020.12.486.

[10] Z. Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers," *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 5, 1993, doi: 10.1109/21.260670.

[11] M. A. Johnson *et al.*, *PID control: New identification and design methods*. 2005.

[12] K. H. Ang, G. Chong, S. Member, and Y. Li, “PID Control System Analysis , Design , and Technology,” vol. 13, no. 4, pp. 559–576, 2005.

[13] K. J. Åström and T. Hägglund, “The future of PID control,” *Control Eng. Pract.*, vol. 9, no. 11, pp. 1163–1175, 2001, doi: 10.1016/S0967-0661(01)00062-4.

[14] O. Ayokunle, K. Akingbade, and F. Dahunsi, “Annual Reviews in Control The dilemma of PID tuning ☆,” *Annu. Rev. Control*, no. May, 2021, doi: 10.1016/j.arcontrol.2021.05.002.

[15] T. George and V. Ganesan, “Optimal tuning of PID controller in time delay system: A review on various optimization techniques,” *Chem. Prod. Process Model.*, vol. 17, no. 1, pp. 1–28, 2022, doi: 10.1515/cppm-2020-2001.

[16] R. G. Franks and C. W. Worley, “Quantitative Analysis of Cascade Control,” *Ind. Eng. Chem.*, vol. 48, no. 6, 1956, doi: 10.1021/ie50558a034.

[17] D. G. Padhan and S. Majhi, “Modified Smith predictor based cascade control of unstable time delay processes,” *ISA Trans.*, vol. 51, no. 1, pp. 95–104, 2012, doi: 10.1016/j.isatra.2011.08.002.

[18] A. Leva, *Fast autotuning of process cascade controls*, vol. 41, no. 2. IFAC, 2008.

[19] S. Song, W. Cai, and Y. G. Wang, “Auto-tuning of cascade control systems,” *ISA Trans.*, vol. 42, no. 1, 2003, doi: 10.1016/s0019-0578(07)60114-1.

[20] S. Song, L. Xie, and W. J. Cai, “Auto-tuning of cascade control systems,” in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2002, vol. 4, doi: 10.1109/wcica.2002.1020152.

[21] J. Zhang, F. Zhang, M. Ren, G. Hou, and F. Fang, “Cascade control of superheated steam temperature with neuro-PID controller,” *ISA Trans.*, vol. 51, no. 6, pp. 778–785, 2012, doi: 10.1016/j.isatra.2012.06.008.

[22] A. Halmous, Y. Oubbati, M. Lahdeb, and S. Arif, “Design a new cascade controller PD-P-PID optimized by marine predators algorithm for load frequency control,” *Soft Comput.*, vol. 27, no. 14, 2023, doi: 10.1007/s00500-023-08089-w.

[23] A. Saleem, B. Taha, T. Tutunji, and A. Al-Qaisia, “Identification and cascade control of servo-pneumatic system using Particle Swarm Optimization,” *Simul. Model. Pract.*

Theory, vol. 52, pp. 164–179, 2015, doi: 10.1016/j.simpat.2015.01.007.

[24] S. K. Oh, H. J. Jang, and W. Pedrycz, "Optimized fuzzy PD cascade controller: A comparative analysis and design," *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 181–195, 2011, doi: 10.1016/j.simpat.2010.06.004.

[25] R. Kumar, S. Srivastava, and J. R. P. Gupta, "Artificial Neural Network based PID controller for online control of dynamical systems," *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst. ICPEICES 2016*, pp. 1–6, 2017, doi: 10.1109/ICPEICES.2016.7853092.

[26] A. Mehta, A. Rawat, and P. Chauhan, *Advances in control systems and its infrastructure*. 2019.

[27] P. Shah and S. Agashe, "Review of fractional PID controller," *Mechatronics*, vol. 38. 2016, doi: 10.1016/j.mechatronics.2016.06.005.

[28] Y. Luo, Y. Q. Chen, C. Y. Wang, and Y. G. Pi, "Tuning fractional order proportional integral controllers for fractional order systems," *J. Process Control*, vol. 20, no. 7, 2010, doi: 10.1016/j.jprocont.2010.04.011.

[29] H. Li, Y. Luo, and Y. Chen, "A fractional order proportional and derivative (FOPD) motion controller: Tuning rule and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, 2010, doi: 10.1109/TCST.2009.2019120.

[30] C. Yeroglu and N. Tan, "Note on fractional-order proportional-integral-differential controller design," *IET Control Theory Appl.*, vol. 5, no. 17, 2011, doi: 10.1049/iet-cta.2010.0746.

[31] A. K. Maurya, M. R. Bongulwar, and B. M. Patre, "Tuning of fractional order PID controller for higher order process based on ITAE minimization," 2016, doi: 10.1109/INDICON.2015.7443503.

[32] C. I. Muresan, E. H. Dulf, and R. Both, "A Novel Tuning Algorithm for Fractional Order IMC Controllers for Time Delay Processes," *Int. J. Mech. Eng. Robot. Res.*, no. November 2017, pp. 4–8, 2015, doi: 10.18178/ijmerr.4.3.218-221.

[33] Z. Bingul and O. Karahan, "Comparison of PID and FOPID controllers tuned by PSO and ABC algorithms for unstable and integrating systems with time delay," *Optim. Control Appl. Methods*, vol. 39, no. 4, pp. 1431–1450, Jul. 2018, doi: 10.1002/oca.2419.

[34] R. Rajesh, "Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system," *SN Appl. Sci.*, vol. 1, no. 7, pp. 1–14, 2019, doi: 10.1007/s42452-019-0754-3.

[35] A. Basu, S. Mohanty, and R. Sharma, "Designing of the PID and FOPID controllers using conventional tuning techniques," *Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016*, vol. 2, no. 1, 2016, doi: 10.1109/INVENTIVE.2016.7824789.

[36] C.-C. Ku and K. Y. Lee, "Diagonal recurrent neural network based control using adaptive learning rates," 2005, doi: 10.1109/cdc.1992.371204.

[37] M. Singh, S. Srivastava, J. R. P. Gupta, and M. Handmandlu, "Identification and control of a nonlinear system using neural networks by extracting the system dynamics," *IETE J. Res.*, vol. 53, no. 1, 2007, doi: 10.1080/03772063.2007.10876120.

[38] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, 1965, doi: 10.1016/S0019-9958(65)90241-X.

[39] L. A. Zadeh, "CALCULUS OF FUZZY RESTRICTIONS," in *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, 1975.

[40] L. A. Zadeh, "Fuzzy Sets and Their Application to Pattern Classification and Clustering Analysis," in *Classification and Clustering*, 1977.

[41] L. A. Zadeh, "Book review," *Inf. Control*, vol. 4, no. 1, 1961, doi: 10.1016/s0019-9958(61)80044-2.

[42] H. Ying, W. Siler, and J. J. Buckley, "Fuzzy control theory: A nonlinear case," *Automatica*, vol. 26, no. 3, 1990, doi: 10.1016/0005-1098(90)90022-A.

[43] K. S. Tang, K. F. Man, G. Chen, and S. Kwong, "An optimal fuzzy PID controller," *IEEE Trans. Ind. Electron.*, vol. 48, no. 4, 2001, doi: 10.1109/41.937407.

[44] M. A. Johnson *et al.*, *PID control: New identification and design methods*. 2005.

[45] R. K. Mudi and N. R. Pal, "A robust self-tuning scheme for PI- and PD-type fuzzy controllers," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 1, 1999, doi: 10.1109/91.746295.

[46] P. Wang and D. P. Kwok, "Analysis and synthesis of an intelligent control system based on fuzzy logic and the PID principle," *Intell. Syst. Eng.*, vol. 1, no. 2, 1992, doi: 10.1049/ise.1992.0014.

[47] R. Ketata, D. De Geest, and A. Titli, “Fuzzy controller: design, evaluation, parallel and hierarchial combination with a PID controller,” *Fuzzy Sets Syst.*, vol. 71, no. 1, 1995, doi: 10.1016/0165-0114(94)00189-E.

[48] Z. Huang and Q. Shen, “Fuzzy interpolation and extrapolation: A practical approach,” *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, 2008, doi: 10.1109/TFUZZ.2007.902038.

[49] S. Elsayed, R. Sarker, and C. A. Coello Coello, “Fuzzy rule-based design of evolutionary algorithm for optimization,” *IEEE Trans. Cybern.*, vol. 49, no. 1, 2019, doi: 10.1109/TCYB.2017.2772849.

[50] C. F. Juang and P. H. Chang, “Designing fuzzy-rule-based systems using continuous ant-colony optimization,” *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, 2010, doi: 10.1109/TFUZZ.2009.2038150.

[51] Z. Pourbahman and A. Hamzeh, “A fuzzy based approach for fitness approximation in multi-objective evolutionary algorithms,” *J. Intell. Fuzzy Syst.*, vol. 29, no. 5, 2015, doi: 10.3233/IFS-151687.

[52] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, 2002, doi: 10.1109/4235.996017.

[53] Y. Shi and R. C. Eberhart, “Fuzzy adaptive particle swarm optimization,” in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 2001, vol. 1.

[54] K. Nisi, B. Nagaraj, and A. Agalya, “Tuning of a PID controller using evolutionary multi objective optimization methodologies and application to the pulp and paper industry,” *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 8, pp. 2015–2025, 2019, doi: 10.1007/s13042-018-0831-8.

[55] S. P. Ghoshal, “Application of GA/GA-SA based fuzzy automatic generation control of a multi-area thermal generating system,” *Electr. Power Syst. Res.*, vol. 70, no. 2, pp. 115–127, 2004, doi: 10.1016/j.epsr.2003.11.013.

[56] T. Niizato and Y. Gunji, “Applying Weak Equivalence of Categories Between Partial Map and Pointed Set 2-Arms Bandit Problem,” *Complexity*, vol. 16, no. 4, pp. 10–21, 2011, doi: 10.1002/cplx.

[57] M. Gheisarnejad, “An effective hybrid harmony search and cuckoo optimization

algorithm based fuzzy PID controller for load frequency control," *Appl. Soft Comput. J.*, vol. 65, 2018, doi: 10.1016/j.asoc.2018.01.007.

[58] R. Sivalingam, S. Chinnamuthu, and S. S. Dash, "A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems," *Automatika*, vol. 58, no. 4, 2017, doi: 10.1080/00051144.2018.1465688.

[59] A. Fereidouni, M. A. S. Masoum, and M. Moghbel, "A new adaptive configuration of PID type fuzzy logic controller," *ISA Trans.*, vol. 56, 2015, doi: 10.1016/j.isatra.2014.11.010.

[60] V. Kumar, B. C. Nakra, and A. P. Mittal, "A review of classical and fuzzy PID controllers," *Int. J. Intell. Control Syst.*, vol. 16, no. 3, pp. 170–181, 2011, [Online]. Available: http://www.ezconf.net/newfiles/IJICS/232/IJICS-2011-03-02-Final-1.pdf.

[61] S. Saksida and F. Bremsak, "Neural networks in control systems," *Melecon*, vol. 10, no. 3, pp. 816–819, 1991.

[62] G. Waterworth and M. Lees, "Artificial Neural Networks in the Modelling and Control of Non-Linear Systems," *IFAC Proc. Vol.*, vol. 33, no. 1, pp. 95–97, 2000, doi: 10.1016/s1474-6670(17)35594-5.

[63] J. Pongfai, X. Su, H. Zhang, and W. Assawinchaichote, "A novel optimal PID controller autotuning design based on the SLP algorithm," *Expert Syst.*, vol. 37, no. 2, pp. 1–15, 2020, doi: 10.1111/exsy.12489.

[64] R. K. Mandava and P. R. Vundavilli, "An adaptive PID control algorithm for the two-legged robot walking on a slope," *Neural Computing and Applications*, vol. 32, no. 8. pp. 3407–3421, 2020, doi: 10.1007/s00521-019-04326-2.

[65] T. Varshney and S. Sheel, "Improved NN-PID control of MIMO systems with PSO-based initialisation of weights," *Int. J. Autom. Control*, vol. 8, no. 2, pp. 158–172, 2014, doi: 10.1504/IJAAC.2014.063362.

[66] G. Tang, J. Lei, H. Du, B. Yao, W. Zhu, and X. Hu, "Proportional-integral-derivative controller optimization by particle swarm optimization and back propagation neural network for a parallel stabilized platform in marine operations," *J. Ocean Eng. Sci.*, no. xxxx, 2022, doi: 10.1016/j.joes.2022.05.015.

[67] R. A. Aliev, B. G. Guirimov, B. Fazlollahi, and R. R. Aliev, "Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: Recurrent fuzzy neural networks," *Fuzzy Sets Syst.*, vol. 160, no. 17, 2009, doi: 10.1016/j.fss.2008.12.018.

[68] J. S. R. Jang and C. T. Sun, "Neuro-Fuzzy Modeling and Control," *Proc. IEEE*, vol. 83, no. 3, 1995, doi: 10.1109/5.364486.

[69] G. Quadrelli, R. Tanscheit, and M. M. Vellasco, "Neuro-fuzzy modelling and control of nonlinear dynamic systems," *Learn. Nonlinear Model.*, vol. 1, no. 1, 2003, doi: 10.21528/lnlm-vol1-no1-artt.

[70] H. Li, C. L. P. Chen, and H.-P. Huang, "Foundation of Neuro-Fuzzy Systems and an Engineering Application," in *Fuzzy Neural Intelligent Systems*, 2019.

[71] B. Pękala, "Introduction to Fuzzy Sets," in *Studies in Fuzziness and Soft Computing*, vol. 367, 2019.

[72] S. Carbas, A. Toktas, and D. Ustun, "Introduction and Overview: Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications," 2021.

[73] N. Khanduja and B. Bhushan, "Recent advances and application of metaheuristic algorithms: A survey (2014–2020)," in *Studies in Computational Intelligence*, vol. 916, 2021.

[74] W. K. Wong and C. I. Ming, "A Review on Metaheuristic Algorithms: Recent Trends, Benchmarking and Applications," *2019 7th Int. Conf. Smart Comput. Commun. ICSCC 2019*, pp. 1–5, 2019, doi: 10.1109/ICSCC.2019.8843624.

[75] A. H. Halim, I. Ismail, and S. Das, *Performance assessment of the metaheuristic optimization algorithms: an exhaustive review*, vol. 54, no. 3. Springer Netherlands, 2021.

[76] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Computing Surveys*, vol. 35, no. 3. 2003, doi: 10.1145/937503.937505.

[77] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," in *Information Sciences*, 2013, vol. 237, doi: 10.1016/j.ins.2013.02.041.

[78] J. Kennedy and R. Eberhart, "Particle swarm optimization PAPER - IGNORE FROM

REFS," pp. 1942–1948, 1995.

[79] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering*, vol. 2015. 2015, doi: 10.1155/2015/931256.

[80] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput. Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/J.CAD.2010.12.015.

[81] M. Črepinšek, S. H. Liu, and L. Mernik, "A note on teaching-learning-based optimization algorithm," *Inf. Sci. (Ny).*, vol. 212, pp. 79–93, 2012, doi: 10.1016/j.ins.2012.05.009.

[82] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *CAD Comput. Aided Des.*, vol. 43, no. 3, pp. 303–315, 2011, doi: 10.1016/j.cad.2010.12.015.

[83] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci. (Ny).*, vol. 183, no. 1, 2012, doi: 10.1016/j.ins.2011.08.006.

[84] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/J.ADVENGSOFT.2016.01.008.

[85] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm Evol. Comput.*, vol. 48, no. March, pp. 1–24, 2019, doi: 10.1016/j.swevo.2019.03.004.

[86] F. Han, J. Jiang, Q. H. Ling, and B. Y. Su, "A survey on metaheuristic optimization for random single-hidden layer feedforward neural network," *Neurocomputing*, vol. 335, 2019, doi: 10.1016/j.neucom.2018.07.080.

[87] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, 2017, doi: 10.1016/j.engappai.2017.01.013.

[88] A. Kaveh and N. Khavaninzadeh, "Efficient training of two ANNs using four meta-heuristic algorithms for predicting the FRP strength," *Structures*, vol. 52, pp. 256–272, Jun. 2023, doi: 10.1016/J.ISTRUC.2023.03.178.

[89] M. A. A. Al-qaness, A. A. Ewees, L. Abualigah, A. M. AlRassas, H. V. Thanh, and M. Abd Elaziz, "Evaluating the Applications of Dendritic Neuron Model with Metaheuristic Optimization Algorithms for Crude-Oil-Production Forecasting," *Entropy*, vol. 24, no. 11, pp. 1–14, 2022, doi: 10.3390/e24111674.

[90] F. Wallam and A. Y. Memon, "A robust control scheme for nonlinear non-isothermal uncertain jacketed continuous stirred tank reactor," *J. Process Control*, vol. 51, pp. 55–67, Mar. 2017, doi: 10.1016/J.JPROCONT.2016.11.001.

[91] B. W. Bequette, "Behavior of a CSTR with a recirculating jacket heat transfer system," *Proc. Am. Control Conf.*, vol. 4, pp. 3275–3280, 2002, doi: 10.1109/acc.2002.1025296.

[92] M. King, *Process Dynamics*. 2016.

[93] P. E. Wellstead, V. Chrimes, P. R. Fletcher, R. Moody, and A. J. Robins, "Ball and Beam Control Experiment.," *Int. J. Electr. Eng. Educ.*, vol. 15, no. 1, pp. 21–39, 1978, doi: 10.1177/002072097801500107.

[94] A. K. Shaout and M. A. Jarrah, "Cruise control technology review," *Comput. Electr. Eng.*, vol. 23, no. 4, pp. 259–271, 1997, doi: 10.1016/S0045-7906(97)00013-X.

[95] P. A. Ioannou and C. C. Chien, "Autonomous Intelligent Cruise Control," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 657–672, 1993, doi: 10.1109/25.260745.

[96] J. J. Wang, "Simulation studies of inverted pendulum based on PID controllers," *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 440–449, 2011, doi: 10.1016/j.simpat.2010.08.003.

[97] C. C. Chung and J. Hauser, "Nonlinear control of a swinging pendulum," *Automatica*, vol. 31, no. 6, pp. 851–862, 1995, doi: 10.1016/0005-1098(94)00148-C.

[98] L. B. Prasad, B. Tyagi, and H. O. Gupta, "Optimal control of nonlinear inverted pendulum system using PID controller and LQR: Performance analysis without and with disturbance input," *Int. J. Autom. Comput.*, vol. 11, no. 6, pp. 661–670, 2014, doi: 10.1007/s11633-014-0818-1.

[99] N. P. Nguyen, H. Oh, Y. Kim, J. Moon, J. Yang, and W. H. Chen, "Fuzzy-Based Super-Twisting Sliding Mode Stabilization Control for Under-Actuated Rotary Inverted Pendulum Systems," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3029095.

[100] J. Yi and N. Yubazaki, "Stabilization fuzzy control of inverted pendulum systems," *Artif. Intell. Eng.*, vol. 14, no. 2, 2000, doi: 10.1016/S0954-1810(00)00007-8.

[101] C. W. Anderson, "Learning to Control an Inverted Pendulum Using Neural Networks," *IEEE Control Syst. Mag.*, vol. 9, no. 3, 1989, doi: 10.1109/37.24809.

[102] S. Jung and S. S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, 2008, doi: 10.1109/TCST.2007.903396.

[103] M. Borrello, "Modeling and control of systems for critical care ventilation," *Proc. Am. Control Conf.*, vol. 3, pp. 2166–2180, 2005, doi: 10.1109/acc.2005.1470291.

[104] A. S. Tran, H. Quang, T. Ngo, V. K. Dong, and A. H. Vo, "Design , Control , Modeling , and Simulation of Mechanical Ventilator for Respiratory Support," vol. 2021, 2021.

[105] B. Hunnekens, S. Kamps, and N. Van De Wouw, "Variable-Gain Control for Respiratory Systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 1, pp. 163–171, 2020, doi: 10.1109/TCST.2018.2871002.

[106] M. Dai, Z. S. Zhang, Z. G. Liu, and D. F. Yin, "Control module design for a continuous positive airway pressure ventilator," in *Applied Mechanics and Materials*, 2013, vol. 321–324, doi: 10.4028/www.scientific.net/AMM.321-324.1657.

[107] A. T. Yimchunger, D. Acharya, and D. K. Das, "Particle Swarm Optimization based PID-Controller Design for Volume Control of Artificial Ventilation System," *2020 IEEE Calcutta Conf. CALCON 2020 - Proc.*, vol. C, pp. 278–282, 2020, doi: 10.1109/CALCON49167.2020.9106480.

[108] I. M. Mehedi, H. S. M. Shah, U. M. Al-Saggaf, R. Mansouri, and M. Bettayeb, "Fuzzy PID Control for Respiratory Systems," *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/7118711.

[109] J. Reinders, B. Hunnekens, F. Heck, T. Oomen, and N. Van De Wouw, "Adaptive Control for Mechanical Ventilation for Improved Pressure Support," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 180–193, 2021, doi: 10.1109/TCST.2020.2969381.

[110] D. Acharya and D. K. Das, "Swarm optimization approach to design PID controller for artificially ventilated human respiratory system," *Comput. Methods Programs Biomed.*,

vol. 198, p. 105776, 2021, doi: 10.1016/j.cmpb.2020.105776.

[111] Srivastava, V., Srivastava, S., Chaudhary, G., & Blanco Valencia, "Performance improvement and Lyapunov stability analysis of nonlinear systems using hybrid optimization techniques," *Expert Syst.*, 2022.

[112] J. Chen and T. C. Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control," *J. Process Control*, vol. 14, no. 2, 2004, doi: 10.1016/S0959-1524(03)00039-8.

[113] V. Srivastava, S. Srivastava, G. Chaudhary, X. Guzmán-Guzmán, and V. García-Díaz, "On Comparing the Performance of Swarm-Based Algorithms with Human-Based Algorithm for Nonlinear Systems," *Arabian Journal for Science and Engineering*. 2021, doi: 10.1007/s13369-021-06026-3.

[114] A. Visioli, "PID Control System Design and Automatic Tuning Using MATLAB/SIMULINK [Bookshelf]," *IEEE Control Syst.*, vol. 41, no. 3, 2022, doi: 10.1109/mcs.2021.3062959.

[115] N. Method, "Ziegler - Nichols Method : Ziegler - Nichols Tuning Chart :," vol. 25, no. 10, pp. 1–2, 2015.

[116] Astrom, Karl J. "PID controllers: theory, design, and tuning." *The International Society of Measurement and Control* (1995).

[117] M. Oglesby, "Essentials of process control," *J. Process Control*, vol. 7, no. 3, 1997, doi: 10.1016/s0959-1524(97)85562-x.

[118] M. H. Moradi, "New techniques for PID controller design," *IEEE Conf. Control Appl. - Proc.*, vol. 2, no. Chien 1988, pp. 903–908, 2003, doi: 10.1109/cca.2003.1223130.

[119] K. J. Åström, T. Hägglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers - a survey," *Control Eng. Pract.*, vol. 1, no. 4, 1993, doi: 10.1016/0967-0661(93)91394-C.

[120] A. Saleem, B. Taha, T. Tutunji, and A. Al-Qaisia, "Identification and cascade control of servo-pneumatic system using Particle Swarm Optimization," *Simul. Model. Pract. Theory*, vol. 52, 2015, doi: 10.1016/j.simpat.2015.01.007.

[121] N. Ullah and A. S. Mohammad, "Cascaded robust control of mechanical ventilator using

fractional order sliding mode control," *Math. Biosci. Eng.*, vol. 19, no. 2, pp. 1332–1354, 2022, doi: 10.3934/mbe.2022061.

[122] M. Ponni Bala, E. B. Priyanka, K. Prabhu, P. Bharathi Priya, T. Bhuvana, and V. Cibi Raja, "Real-time performance analysis of temperature process using continuous stirred tank reactor," *2021 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2021*, pp. 647–650, 2021, doi: 10.1109/ESCI50559.2021.9396877.

[123] X. Li and L. Gao, "Robust Fractional-order PID Tuning Method for a Plant with an Uncertain Parameter," *Int. J. Control. Autom. Syst.*, vol. 19, no. 3, pp. 1302–1310, 2021, doi: 10.1007/s12555-019-0866-y.

[124] N. Divya, S. Manoharan, J. Arulvadivu, and P. Palpandian, "An efficient tuning of fractional order PID controller for an industrial control process," *Mater. Today Proc.*, vol. 57, pp. 1654–1659, Jan. 2022, doi: 10.1016/J.MATPR.2021.12.255.

[125] I. M. Mehedi, U. M. Al-Saggaf, R. Mansouri, and M. Bettayeb, "Two degrees of freedom fractional controller design: Application to the ball and beam system," *Measurement*, vol. 135, pp. 13–22, Mar. 2019, doi: 10.1016/J.MEASUREMENT.2018.11.021.

[126] R. T. Aguiar, B. S. de Lima, and A. A. T. Maia, "Comparative study of tuning techniques for fractional PID controllers for FOPDT systems," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 44, no. 5. 2022, doi: 10.1007/s40430-022-03491-1.

[127] S. M. Ghamari, H. G. Narm, and H. Mollaee, "Fractional-order fuzzy PID controller design on buck converter with antlion optimization algorithm," *IET Control Theory and Applications*, vol. 16, no. 3. pp. 340–352, 2022, doi: 10.1049/cth2.12230.

[128] E. G. Talbi, *Metaheuristics: From Design to Implementation*. 2009.

[129] O. E. Turgut, M. S. Turgut, and E. Kırtepe, "A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems," *Neural Comput. Appl.*, vol. 35, no. 19, 2023, doi: 10.1007/s00521-023-08481-5.

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

[131] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas,

"Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5. p. e09399, 2022, doi: 10.1016/j.heliyon.2022.e09399.

[132] I. El-Henawy and N. Ahmed, "Meta-Heuristics Algorithms: A Survey," *Int. J. Comput. Appl.*, vol. 179, no. 22, 2018, doi: 10.5120/ijca2018916427.

[133] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An Overview of Variants and Advancements of PSO Algorithm," *Appl. Sci.*, vol. 12, no. 17, pp. 1–21, 2022, doi: 10.3390/app12178392.

[134] 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.

[135] A. Agalya and B. Nagaraj, "Certain investigation on concentration control of CSTR - A comparative approach," *Int. J. Adv. Soft Comput. its Appl.*, vol. 5, no. 2, pp. 1–14, 2013.

[136] R. P. Parouha and P. Verma, *State-of-the-Art Reviews of Meta-Heuristic Algorithms with Their Novel Proposal for Unconstrained Optimization and Applications*, vol. 28, no. 5. Springer Netherlands, 2021.

[137] S. Yarat, S. Senan, and Z. Orman, "A Comparative Study on PSO with Other Metaheuristic Methods," in *International Series in Operations Research and Management Science*, vol. 306, 2021.

[138] S. Baskar and P. N. Suganthan, "A novel concurrent particle swarm optimization," in *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, 2004, vol. 1, doi: 10.1109/cec.2004.1330940.

[139] M. R. Tanweer, S. Suresh, and N. Sundararajan, "Self regulating particle swarm optimization algorithm," *Inf. Sci. (Ny).*, vol. 294, 2015, doi: 10.1016/j.ins.2014.09.053.

[140] S. Chatterjee and V. Mukherjee, "PID controller for automatic voltage regulator using teaching-learning based optimization technique," *Int. J. Electr. Power Energy Syst.*, vol. 77, pp. 418–429, 2016, doi: 10.1016/j.ijepes.2015.11.010.

[141] 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, 2015, doi: 10.1016/j.asoc.2014.11.027.

[142] A. W. Lins and R. Krishnakumar, "Tuning of PID controller for a PV-fed BLDC motor using PSO and TLBO algorithm," *Applied Nanoscience (Switzerland)*. 2022, doi: 10.1007/s13204-021-02272-x.

[143] J. Bhookya and R. Kumar Jatoth, "Fractional Order PID Controller Design for Multivariable Systems using TLBO," *Chem. Prod. Process Model.*, vol. 15, no. 2, 2020, doi: 10.1515/cppm-2019-0061.

[144] A. Tiwari and M. K. Pradhan, "Applications of TLBO algorithm on various manufacturing processes: AReview," in *Materials Today: Proceedings*, 2017, vol. 4, no. 2, doi: 10.1016/j.matpr.2017.02.003.

[145] H. Dong, Y. Xu, D. Cao, W. Zhang, Z. Yang, and X. Li, "An improved teaching–learning-based optimization algorithm with a modified learner phase and a new mutation-restarting phase," *Knowledge-Based Syst.*, vol. 258, 2022, doi: 10.1016/j.knosys.2022.109989.

[146] Y. Xu, Y. Peng, X. Su, Z. Yang, C. Ding, and X. Yang, "Improving teaching–learning-based-optimization algorithm by a distance-fitness learning strategy," *Knowledge-Based Syst.*, vol. 257, 2022, doi: 10.1016/j.knosys.2022.108271.

[147] Z. Chen, Y. Liu, Z. Yang, X. Fu, J. Tan, and X. Yang, "An enhanced teaching-learning-based optimization algorithm with self-adaptive and learning operators and its search bias towards origin," *Swarm Evol. Comput.*, vol. 60, 2021, doi: 10.1016/j.swevo.2020.100766.

[148] Y. Xu, Z. Yang, X. Li, H. Kang, and X. Yang, "Dynamic opposite learning enhanced teaching–learning-based optimization," *Knowledge-Based Syst.*, vol. 188, 2020, doi: 10.1016/j.knosys.2019.104966.

[149] S. C. Satapathy, A. Naik, and K. Parvathi, "A teaching learning based optimization based on orthogonal design for solving global optimization problems," *Springerplus*, vol. 2, no. 1, 2013, doi: 10.1186/2193-1801-2-130.

[150] S. K. Vavilala, V. Thirumavalavan, and C. K, "Level control of a conical tank using the fractional order controller," *Comput. Electr. Eng.*, vol. 87, p. 106690, 2020, doi: 10.1016/j.compeleceng.2020.106690.

[151] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale

Optimization Algorithm and its applications," *Swarm Evol. Comput.*, vol. 48, 2019, doi: 10.1016/j.swevo.2019.03.004.

[152] A. Kumar and S. Suhag, "Whale Optimization Algorithm Optimized Fuzzy-PID Plus PID Hybrid Controller for Frequency Regulation in Hybrid Power System," *J. Inst. Eng. Ser. B*, vol. 103, no. 2, pp. 633–648, 2022, doi: 10.1007/s40031-021-00656-9.

[153] K. Amuthambigaiyin Sundari and P. Maruthupandi, "Optimal Design of PID Controller for the analysis of Two TANK System Using Metaheuristic Optimization Algorithm," *Journal of Electrical Engineering and Technology*, vol. 17, no. 1. pp. 627–640, 2022, doi: 10.1007/s42835-021-00891-6.

[154] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Inf. Process. Lett.*, vol. 93, no. 5, 2005, doi: 10.1016/j.ipl.2004.11.003.

[155] F. A. Şenel, F. Gökçe, A. S. Yüksel, and T. Yiğit, "A novel hybrid PSO–GWO algorithm for optimization problems," *Eng. Comput.*, vol. 35, no. 4, 2019, doi: 10.1007/s00366-018-0668-5.

[156] M. H. Spiegel and T. I. Strasser, "Hybrid Optimization Toward Proactive Resilient Microgrid Scheduling," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3110607.

[157] J. Li *et al.*, "Optimal allocation method of multi-energy system based on hybrid optimization algorithm," *Energy Reports*, vol. 9, 2023, doi: 10.1016/j.egyr.2023.04.244.

[158] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2906757.

[159] H. B. Kazemian, "Developments of fuzzy PID controllers," *Expert Syst.*, vol. 22, no. 5, pp. 254–264, 2005, doi: 10.1111/j.1468-0394.2005.00316.x.

[160] D. Mohanty and S. Panda, "Modified Salp Swarm Algorithm - Optimized Fractional - Order Adaptive Fuzzy PID Controller for Frequency Regulation of Hybrid Power System with Electric Vehicle," *J. Control. Autom. Electr. Syst.*, vol. 32, no. 2, pp. 416–438, 2021, doi: 10.1007/s40313-020-00683-9.

[161] M. A. A. Al-qaness, A. M. Helmi, A. Dahou, and M. A. Elaziz, "The Applications of

Metaheuristics for Human Activity Recognition and Fall Detection Using Wearable Sensors: A Comprehensive Analysis,” *Biosensors*, vol. 12, no. 10, pp. 1–21, 2022, doi: 10.3390/bios12100821.

[162] J. Zhang, J. Zhuang, H. Du, and S. Wang, “Self-organizing genetic algorithm based tuning of PID controllers,” *Inf. Sci. (Ny).*, vol. 179, no. 7, pp. 1007–1018, Mar. 2009, doi: 10.1016/J.INS.2008.11.038.

[163] I. O. P. C. Series and M. Science, “Performance Evaluation of Different Objective Function in PID Tuned by PSO in DC-Motor Speed Control Performance Evaluation of Different Objective Function in PID Tuned by PSO in DC-Motor Speed Control,” 2021, doi: 10.1088/1757-899X/1096/1/012061.

[164] V. Srivastava and S. Srivastava, “Control of continuous stirred tank reactor (CSTR) using nature inspired algorithms,” *Journal of Information and Optimization Sciences*, vol. 40, no. 2. pp. 329–338, 2019, doi: 10.1080/02522667.2019.1578092.

[165] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Algorithms in Modeling and Optimization*, no. May 2017. 2013.

[166] S. E. I. Hasseni, L. Abdou, and H. E. Glida, “Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms,” *Evol. Intell.*, vol. 14, no. 1, pp. 61–73, 2021, doi: 10.1007/s12065-019-00312-8.

[167] M. A. A. Al-qaness, A. A. Ewees, H. V. Thanh, A. M. AlRassas, A. Dahou, and M. A. Elaziz, “Predicting CO2 trapping in deep saline aquifers using optimized long short-term memory,” *Environ. Sci. Pollut. Res.*, vol. 30, no. 12, pp. 33780–33794, 2023, doi: 10.1007/s11356-022-24326-5.

[168] J. Kang, W. Meng, A. Abraham, and H. Liu, “An adaptive PID neural network for complex nonlinear system control,” *Neurocomputing*, vol. 135, pp. 79–85, 2014, doi: 10.1016/j.neucom.2013.03.065.

[169] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, “Neural network-based self-tuning PID control for underwater vehicles,” *Sensors (Switzerland)*, vol. 16, no. 9, 2016, doi: 10.3390/s16091429.

[170] L. Zhang, S. Li, Y. Xue, H. Zhou, and Z. Ren, “Neural network PID control for

combustion instability,” *Combust. Theory Model.*, 2022, doi: 10.1080/13647830.2022.2025908.

[171] A. Marino and F. Neri, “PID Tuning with Neural Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11431 LNAI, doi: 10.1007/978-3-030-14799-0_41.

[172] M. A. A. Al-qaness, A. A. Ewees, H. Fan, L. Abualigah, A. H. Elsheikh, and M. Abd Elaziz, “Wind power prediction using random vector functional link network with capuchin search algorithm,” *Ain Shams Eng. J.*, vol. 14, no. 9, p. 102095, 2023, doi: 10.1016/j.asej.2022.102095.

[173] H. Liang, Z. K. Sang, Y. Z. Wu, Y. H. Zhang, and R. Zhao, “High precision temperature control performance of a PID neural network-controlled heater under complex outdoor conditions,” *Appl. Therm. Eng.*, vol. 195, p. 117234, Aug. 2021, doi: 10.1016/J.APPLTHERMALENG.2021.117234.

[174] G. Q. Zeng, X. Q. Xie, M. R. Chen, and J. Weng, “Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems,” *Swarm Evol. Comput.*, vol. 44, pp. 320–334, 2019, doi: 10.1016/j.swevo.2018.04.008.

[175] S. P. Sathiyan, S. S. Kumar, and A. I. Selvakumar, “Optimised fuzzy controller for improved comfort level during transitions in Cruise and Adaptive Cruise Control Vehicles,” 2015, doi: 10.1109/SPACES.2015.7058221.

[176] R. Pradhan, S. K. Majhi, J. K. Pradhan, and B. B. Pati, “Antlion optimizer tuned PID controller based on Bode ideal transfer function for automobile cruise control system,” *J. Ind. Inf. Integr.*, vol. 9, pp. 45–52, 2018, doi: 10.1016/j.jii.2018.01.002.

[177] G. Prabhakar, S. Selvaperumal, and P. Nedumal Pugazhenthi, “Fuzzy PD Plus I Control-based Adaptive Cruise Control System in Simulation and Real-time Environment,” *IETE Journal of Research*, vol. 65, no. 1. pp. 69–79, 2019, doi: 10.1080/03772063.2017.1407269.

[178] M. Faieghi, A. Jalali, and S. K. E. D. M. Mashhadi, “Robust adaptive cruise control of high speed trains,” *ISA Trans.*, vol. 53, no. 2, pp. 533–541, 2014, doi: 10.1016/j.isatra.2013.12.007.

[179] S. Chamraz and R. Balogh, “Two approaches to the adaptive cruise control (ACC) design,” *Proc. 29th Int. Conf. Cybern. Informatics, K I 2018*, vol. 2018-Janua, no. 2, pp. 1–6, 2018, doi: 10.1109/CYBERI.2018.8337542.

[180] D. Izci, S. Ekinci, M. Kayri, and E. Eker, “A novel improved arithmetic optimization algorithm for optimal design of PID controlled and Bode’s ideal transfer function based automobile cruise control system,” *Evol. Syst.*, vol. 13, no. 3, 2022, doi: 10.1007/s12530-021-09402-4.

[181] S. Ekinci, D. Izci, L. Abualigah, A. G. Hussien, C. Le Thanh, and S. Khatir, “Revolutionizing Vehicle Cruise Control: An Elite Opposition-Based Pattern Search Mechanism Augmented INFO Algorithm for Enhanced Controller Design,” *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, 2023, doi: 10.1007/s44196-023-00304-8.

[182] S. Sakthiya Ram, C. Kumar, A. Ramesh Kumar, and T. Rajesh, “Hybrid optimization techniques based automatic artificial respiration system for corona patient,” *Automatika*, vol. 63, no. 2. pp. 226–243, 2022, doi: 10.1080/00051144.2022.2031538.

[183] E. P. Martinoni *et al.*, “Model-based control of mechanical ventilation: Design and clinical validation,” *British Journal of Anaesthesia*, vol. 92, no. 6. pp. 800–807, 2004, doi: 10.1093/bja/aeh145.

[184] K. Takahara and H. Wakamatsu, “Control of artificial respiration by adaptive pole-placement method,” *Syst. Comput. Japan*, vol. 25, no. 8, 1994, doi: 10.1002/scj.4690250807.

[185] R. De Keyser and C. Ionescu, “Adaptive control of a pressure-controlled artificial ventilator: A simulator-based evaluation using real COPD patient data,” *Comput. Methods Programs Biomed.*, vol. 104, no. 3, pp. e178–e188, 2011, doi: 10.1016/j.cmpb.2011.03.001.

[186] I. M. Mehedi, H. S. M. Shah, U. M. Al-Saggaf, R. Mansouri, and M. Bettayeb, “Adaptive Fuzzy Sliding Mode Control of a Pressure-Controlled Artificial Ventilator,” *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/1926711.

[187] I. M. Batiha, R. El-Khazali, O. Y. Ababneh, A. Ouannas, R. M. Batyha, and S. Momani, “Optimal design of PIρDμ-controller for artificial ventilation systems for COVID-19 patients,” *AIMS Math.*, vol. 8, no. 1, 2023, doi: 10.3934/math.2023031.

[188] P. B. Sistu and B. W. Bequette, “CpkLfkD),” vol. 47, no. 9, pp. 2553–2558, 1992.

[189] 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.

[190] S. N. Deepa and N. Y. Jayalakshmi, “Optimized Fuzzy-Based Wavelet Neural Network Controller for a Non-Linear Process Control System,” *IETE J. Res.*, vol. 69, no. 3, pp. 1363–1372, 2023, doi: 10.1080/03772063.2020.1865212.

[191] F. Li, X. Cao, C. Zhou, and C. Yang, “Event-triggered asynchronous sliding mode control of CSTR based on Markov model,” *J. Franklin Inst.*, vol. 358, no. 9, pp. 4687–4704, 2021, doi: 10.1016/j.jfranklin.2021.04.007.

[192] W. Der Chang, “Nonlinear CSTR control system design using an artificial bee colony algorithm,” *Simul. Model. Pract. Theory*, vol. 31, pp. 1–9, Feb. 2013, doi: 10.1016/J.SIMPAT.2012.11.002.

[193] S. Baruah and L. Dewan, “A comparative study of PID based temperature control of CSTR using Genetic Algorithm and Particle Swarm Optimization,” *2017 Int. Conf. Emerg. Trends Comput. Commun. Technol. ICETCCT 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICETCCT.2017.8280312.

[194] P. Kalita and J. K. Barman, “Performance Analysis of Non-linear Jacketed CSTR based on Different Control Strategies,” *ADBU J. Electr. Electron. Eng.*, vol. 1, no. 2, pp. 1–7, 2017.

[195] A. E. Gamboa-Torres and tonio AnFlores-Tlacuahuac, “Effect of process modeling on the nonlinear behaviour of a CSTR reactions A → B → C,” *Chem. Eng. J.*, vol. 77, no. 3, pp. 153–164, 2000, doi: 10.1016/S1385-8947(99)00157-6.

[196] A. Soukkou, A. Khellaf, S. Leulmi, and K. Boudeghdegh, “Optimal control of a CSTR process,” *Brazilian J. Chem. Eng.*, vol. 25, no. 4, pp. 799–812, 2008, doi: 10.1590/S0104-66322008000400017.

[197] S. N. Deepa and I. Baranilingesan, “Optimized deep learning neural network predictive controller for continuous stirred tank reactor,” *Comput. Electr. Eng.*, vol. 71, pp. 782–797, Oct. 2018, doi: 10.1016/j.compeleceng.2017.07.004.

[198] G. Maruthi Prasad and A. Seshagiri Rao, “Multi-model cascade control strategy design

based on gap metric for nonlinear processes," *Indian Chemical Engineer*, vol. 64, no. 2. pp. 183–196, 2022, doi: 10.1080/00194506.2020.1852977.

[199] K. C. Ng, M. M. Trivedi, and C. Vision, "Fuzzy-Neural Controller and Real-Time Implementation of A Ball Balancing Beam," *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2493, pp. 1–17, 1996.

[200] C. L. Chen and Y. M. Chen, "Self-organizing fuzzy logic controller design," *Comput. Ind.*, vol. 22, no. 3, pp. 249–261, 1993, doi: 10.1016/0166-3615(93)90092-F.

[201] Y. Zhang and J. Wang, "Recurrent neural networks for nonlinear output regulation," *Automatica*, vol. 37, no. 8, pp. 1161–1173, 2001, doi: 10.1016/S0005-1098(01)00092-9.

[202] Y. H. Chang, C. W. Chang, C. W. Tao, H. W. Lin, and J. S. Taur, "Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3624–3633, 2012, doi: 10.1016/j.eswa.2011.09.052.

[203] P. Ravichandran, S. Sathiyavathi, S. Sathish Babu, and A. Vimala Starbino, "Hybrid Arrangement of Iterative Learning Control Strategy for Ball and Beam System," *IETE J. Res.*, vol. 69, no. 2, pp. 916–923, 2023, doi: 10.1080/03772063.2020.1844072.

[204] C. Peraza, F. Valdez, J. R. Castro, and O. Castillo, "Fuzzy Dynamic Parameter Adaptation in the Harmony Search Algorithm for the Optimization of the Ball and Beam Controller," *Adv. Oper. Res.*, vol. 2018, 2018, doi: 10.1155/2018/3092872.

[205] L. dos Santos Coelho, "Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach," *Chaos, Solitons and Fractals*, vol. 39, no. 4, pp. 1504–1514, 2009, doi: 10.1016/j.chaos.2007.06.018.

[206] I. O. P. C. Series and M. Science, "Optimal PID Controller based on Convex Optimisation and Particle Swarm Techniques Optimal PID Controller based on Convex Optimisation and Particle Swarm Techniques," 2021, doi: 10.1088/1757-899X/1067/1/012123.

[207] M. A. Sahib and B. S. Ahmed, "A new multiobjective performance criterion used in PID tuning optimization algorithms," *J. Adv. Res.*, vol. 7, no. 1, pp. 125–134, 2016, doi: 10.1016/j.jare.2015.03.004.

[208] W. Der Chang and S. P. Shih, "PID controller design of nonlinear systems using an

improved particle swarm optimization approach," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 11, pp. 3632–3639, Nov. 2010, doi: 10.1016/J.CNSNS.2010.01.005.

[209] Y. Wu, Y. Wei, D. An, and J. Liu, "A Hybrid Control Strategy based on Neural Network and PID for Underwater Robot Hovering," pp. 1843–1848, 2023, doi: 10.1109/cscwd57460.2023.10152857.

[210] R. Pradhan, S. K. Majhi, J. K. Pradhan, and B. B. Pati, "Antlion optimizer tuned PID controller based on Bode ideal transfer function for automobile cruise control system," *J. Ind. Inf. Integr.*, vol. 9, no. January, pp. 45–52, 2018, doi: 10.1016/j.jii.2018.01.002.

[211] Y. S. Prakash and S. A. Kumar, "Mathematical Modelling and control of Cruise Control System on the Inclined Plane," *Int. Conf. Electr. Electron. Eng. ICE3 2020*, pp. 431–436, 2020, doi: 10.1109/ICE348803.2020.9122904.

[212] A. Kuyumcu and N. S. Şengör, "Effect of neural controller on adaptive cruise control," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9887 LNCS, pp. 515–522, 2016, doi: 10.1007/978-3-319-44781-0_61.

[213] F. Diba, A. Arora, and E. Esmailzadeh, "Optimized robust cruise control system for an electric vehicle," *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 175–182, 2014, doi: 10.1080/21642583.2014.891956.

[214] K. Sailan and K. D. Kuhnert, "Modeling and Design of Cruise Control System with Feedforward for all Terrian Vehicles," pp. 339–349, 2013, doi: 10.5121/csit.2013.3828.

[215] Y. Jia, R. Jibrin, Y. Itoh, and D. Görges, "Energy-Optimal Adaptive Cruise Control for Electric Vehicles in Both Time and Space Domain based on Model Predictive Control," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 13–20, 2019, doi: 10.1016/j.ifacol.2019.09.003.

[216] M. Al-Gabalawy, N. S. Hosny, and A. hamid S. Aborisha, "Model predictive control for a basic adaptive cruise control," *Int. J. Dyn. Control*, no. July 2020, 2021, doi: 10.1007/s40435-020-00732-w.

[217] G. Prabhakar, S. Selvaperumal, and P. Nedumal Pugazhenthi, "Fuzzy PD Plus I Control-based Adaptive Cruise Control System in Simulation and Real-time Environment," *IETE J. Res.*, vol. 65, no. 1, pp. 69–79, 2019, doi: 10.1080/03772063.2017.1407269.

[218] M. Walter and S. Leonhardt, "Control applications in artificial ventilation," *2007*

Mediterr. Conf. Control Autom. MED, 2007, doi: 10.1109/MED.2007.4433762.

[219] M. J. Mahmoodabadi and N. Danesh, “Gravitational search algorithm-based fuzzy control for a nonlinear ball and beam system,” *J. Control Decis.*, vol. 5, no. 3, pp. 229–240, 2018, doi: 10.1080/23307706.2017.1403864.

[220] A. Alkrwy, A. A. Hussein, T. H. Atyia, and M. Khamees, “Adaptive Tuning of PID Controller using Crow Search Algorithm for DC motor,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1076, no. 1, p. 012001, 2021, doi: 10.1088/1757-899x/1076/1/012001.

[221] K. Osman, M. F. Rahmat, and M. A. Ahmad, “Modelling and controller design for a cruise control system,” *Proc. 2009 5th Int. Colloq. Signal Process. Its Appl. CSPA 2009*, no. 1, pp. 254–258, 2009, doi: 10.1109/CSPA.2009.5069228.

[222] R. V. Rao and V. Patel, “An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems,” *Sci. Iran.*, vol. 20, no. 3, pp. 710–720, 2013, doi: 10.1016/j.scient.2012.12.005.

[223] “PID Control System Analysis , Design , and Technology,” vol. 13, no. November, pp. 559–576, 2007.

[224] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, “A review of PID control, tuning methods and applications,” *International Journal of Dynamics and Control*. Springer, 2020, doi: 10.1007/s40435-020-00665-4.

[225] O. A. Somefun, K. Akingbade, and F. Dahunsi, “The dilemma of PID tuning,” *Annual Reviews in Control*, vol. 52. Elsevier Ltd, pp. 65–74, Jan. 01, 2021, doi: 10.1016/j.arcontrol.2021.05.002.

[226] A. Y. Jaen-Cuellar, R. D. J. Romero-Troncoso, L. Morales-Velazquez, and R. A. Osornio-Rios, “PID-controller tuning optimization with genetic algorithms in servo systems,” *Int. J. Adv. Robot. Syst.*, vol. 10, 2013, doi: 10.5772/56697.

[227] T. Arrif, S. Hassani, M. Guermoui, A. Sánchez-González, R. A.Taylor, and A. Belaid, “GA-GOA hybrid algorithm and comparative study of different metaheuristic population-based algorithms for solar tower heliostat field design,” *Renew. Energy*, vol. 192, pp. 745–758, Jun. 2022, doi: 10.1016/J.RENENE.2022.04.162.

[228] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” *J. Glob. Optim.*, vol. 39, no. 3,

pp. 459–471, 2007, doi: 10.1007/s10898-007-9149-x.

[229] O. Castillo, E. Lizárraga, J. Soria, P. Melin, and F. Valdez, "New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system," *Inf. Sci. (Ny).*, vol. 294, pp. 203–215, Feb. 2015, doi: 10.1016/J.INS.2014.09.040.

[230] Y. H. Chang, C. W. Chang, C. W. Tao, H. W. Lin, and J. S. Taur, "Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3624–3633, Feb. 2012, doi: 10.1016/J.ESWA.2011.09.052.

[231] M. Ding, B. Liu, and L. Wang, "Position control for ball and beam system based on active disturbance rejection control," *Systems Science and Control Engineering*, vol. 7, no. 1. pp. 97–108, 2019, doi: 10.1080/21642583.2019.1575297.

[232] X. Li and W. Yu, "Synchronization of ball and beam systems with neural compensation," *Int. J. Control. Autom. Syst.*, vol. 8, no. 3, pp. 491–496, 2010, doi: 10.1007/s12555-010-0301-x.

[233] V. Srivastava and S. Srivastava, "Hybrid optimization based PID control of ball and beam system," *J. Intell. Fuzzy Syst.*, vol. 42, no. 2, pp. 919–928, 2022, doi: 10.3233/JIFS-189760.

[234] Y. Wu and Y. Liu, "Fuzzy PID controller design and implement in Ball-Beam system," *Chinese Control Conf. CCC*, vol. 2015-Septe, no. 1, pp. 3613–3616, 2015, doi: 10.1109/ChiCC.2015.7260197.

[235] P. Ravichandran, S. Sathiyavathi, S. Sathish Babu, and A. Vimala Starbino, "Hybrid Arrangement of Iterative Learning Control Strategy for Ball and Beam System," *IETE Journal of Research*. 2020, doi: 10.1080/03772063.2020.1844072.

[236] A. Mohammadi and J. C. Ryu, "Neural network-based PID compensation for nonlinear systems: ball-on-plate example," *Int. J. Dyn. Control*, vol. 8, no. 1, pp. 178–188, 2020, doi: 10.1007/s40435-018-0480-5.

[237] 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.

[238] S. K. Valluru and M. Singh, "Performance investigations of APSO tuned linear and

nonlinear PID controllers for a nonlinear dynamical system," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3. pp. 442–452, 2018, doi: 10.1016/j.jesit.2018.02.001.

[239] S. Bennett, "The past of PID controllers," *Annu. Rev. Control*, vol. 25, 2001, doi: 10.1016/S1367-5788(01)00005-0.

[240] P. Kumar, S. Chatterjee, D. Shah, U. K. Saha, and S. Chatterjee, "On comparison of tuning method of FOPID controller for controlling field controlled DC servo motor," *Cogent Eng.*, vol. 4, no. 1, 2017, doi: 10.1080/23311916.2017.1357875.

[241] İ. H. Köse, Y. Danayiyen and B. Çavdar, "Fractional Order PID Controller Design for Hydroelectric Power Plant Model with PSO, GWO and MA Optimization Algorithms," 2023 Innovations in Intelligent Systems and Applications Conference (ASYU), Sivas, Turkiye, 2023, pp. 1-6, doi: 10.1109/ASYU58738.2023.10296772.

[242] N. Sadati, A. Ghaffarkhah, and S. Ostadabbas, "A new neural network based FOPID controller," *Proc. 2008 IEEE Int. Conf. Networking, Sens. Control. ICNSC*, pp. 762–767, 2008, doi: 10.1109/ICNSC.2008.4525318.

[243] G. L. Grandi and J. O. Trierweiler, "Tuning of fractional order PID controllers based on the frequency response approximation method," in *IFAC-PapersOnLine*, 2019, vol. 52, no. 1, doi: 10.1016/j.ifacol.2019.06.190.

[244] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p. e09399, 2022, doi: 10.1016/j.heliyon.2022.e09399.

[245] P. A. Chertovskikh, A. V. Seredkin, O. A. Gobyzov, A. S. Styuf, M. G. Pashkevich, and M. P. Tokarev, "An adaptive PID controller with an online auto-tuning by a pretrained neural network," in *Journal of Physics: Conference Series*, 2019, vol. 1359, no. 1, doi: 10.1088/1742-6596/1359/1/012090.

[246] D. Park, T. L. Le, N. V. Quynh, N. K. Long, and S. K. Hong, "Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control," *Front. Neurorobot.*, vol. 14, 2021, doi: 10.3389/fnbot.2020.619350.

APPENDIX

Codes of Optimization Algorithms Used

PSO

```
%Initialization%
n = 50;          % Size of the swarm
bird_step  =50;   % Maximum number of "birds steps"
dimension = 3;          % Dimension of the problem
c2 =1.2;          % PSO parameter C1
c1 = 0.12;        % PSO parameter C2
w =0.9;           % momentum or inertia
fitness=0*ones(n,bird_step);
%    initialize the parameter %
R1 = rand(dimension, n);
R2 = rand(dimension, n);
current_fitness =0*ones(n,1);
% Initializing swarm and velocities and position %
current_position = 10*(rand(dimension, n)-.5);
velocity = .3*randn(dimension, n) ;
local_best_position  = current_position ;
%     Evaluate initial population           %
for i = 1:n
current_fitness(i) = pso_pid_cstr1(current_position(:,i));
end
local_best_fitness  = current_fitness ;
[global_best_fitness,g] = min(local_best_fitness) ;
for i=1:n
global_best_position(:,i) = local_best_position(:,g) ;
end
 %  Velocity Update  %
velocity = w*velocity + c1*(R1.*(local_best_position-current_position)) + c2*(R2.*(global_best_position-current_position));
 %   SwarmUpdate %
```

```
current_position = current_position + velocity ;
%  evaluate new swarm   %
% Main Loop%
iter = 0 ;        % Iteration  counter
while  ( iter < bird_setp )
iter = iter + 1;
for i = 1:n,
current_fitness(i) = pso_pid_cstr1(current_position(:,i)) ;
end
for i = 1 : n
if current_fitness(i) < local_best_fitness(i)
local_best_fitness(i)  = current_fitness(i);
local_best_position(:,i) = current_position(:,i)   ;
end
end
[current_global_best_fitness,g] = min(local_best_fitness);
if current_global_best_fitness < global_best_fitness
global_best_fitness = current_global_best_fitness;
for i=1:n
global_best_position(:,i) = local_best_position(:,g);
end
end
velocity = w *velocity + c1*(R1.*(local_best_position-current_position)) + c2*(R2.*(global_best_position-current_position));
current_position = current_position + velocity;
 sprintf('The value of interation iter %3.0f ', iter );
end % end of while loop its mean the end of all step that the birds move it
 xx=fitness(:,50);
 [Y,I] = min(xx);
 current_position(:,I)
 figure;
%plot(BestCosts, 'LineWidth', 2);
```

```
semilogy(global_best_position, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Cost');
grid on;
```

TLBO Algoithm

```
% Objective Function
ObjectiveFunction = @(x) pso_pid_respiratory_new(x);
noVar = 3;          % Number of Unknown Variables
Varvec = [1 noVar]; % Unknown Variables Matrix Size
VarMin = 0;       % Unknown Variables Lower Bound
VarMax =100;       % Unknown Variables Upper Bound
%Define TLBO Parameters%
MaxIter =50;        % Maximum Number of Iterations
noPop = 50;           % Population Size
%Initialization %
empty_individual.Position = [];
empty_individual.Cost = [];
% Initialize Population Array%
pop = repmat(empty_individual, noPop, 1);
% Initialize Best Solution%
BestSol.Cost = inf;
% Initialize Population %
for i=1:noPop
pop(i).Position = unifrnd(VarMin, VarMax, VarSize);
pop(i).Cost = CostFunction(pop(i).Position);
if pop(i).Cost < BestSol.Cost
 BestSol = pop(i);
end
end
% Initialize Best Cost obtained%
BestCosts = zeros(MaxIter,1);
```

```
% TLBO Main Loop start%
for it=1:MaxIter
% Calculate Population Mean%
Mean = 0;
for i=1:noPop
Mean = Mean + pop(i).Position;
end
Mean = Mean/noPop;
% Select Teacher as best solution%
 Teacher = pop(1);
for i=2:noPop
if pop(i).Cost < Teacher.Cost
Teacher = pop(i);
end
end
% Teacher Phase%
for i=1:noPop
% Create Empty Solution%
newsol = empty_individual;
% Select Teaching Factor randomly%
 TF = rand([1 2]);
 % Teaching (moving towards teacher)%
 newsol.Position = pop(i).Position + rand(VarSize).*(Teacher.Position - TF*Mean);
 % Clipping%
newsol.Position = max(newsol.Position, VarMin);
newsol.Position = min(newsol.Position, VarMax);
% Evaluation%
newsol.Cost = CostFunction(newsol.Position);
% Comparision%
if newsol.Cost<pop(i).Cost
pop(i) = newsol;
if pop(i).Cost < BestSol.Cost
```

```
BestSol = pop(i);
end
end
end
% Learner Phase%
for i=1:noPop
A = 1:noPop;
A(i)=[];
j = A(randi(noPop-1));
Step = pop(i).Position - pop(j).Position;
if pop(j).Cost < pop(i).Cost
Step = -Step;
end
% Create Empty Solution%
newsol = empty_individual
% Teaching (moving towards teacher)%
newsol.Position = pop(i).Position + rand(VarSize).*Step;
  % Clipping%
newsol.Position = max(newsol.Position, VarMin);
newsol.Position = min(newsol.Position, VarMax);
 % Evaluation%
newsol.Cost = CostFunction(newsol.Position
% Comparision%
if newsol.Cost<pop(i).Cost
pop(i) = newsol;
if pop(i).Cost < BestSol.Cost
BestSol = pop(i);
End
end
end
% Store Record for Current Iteration%
BestCosts(it) = BestSol.Cost;
```

```
% Show Iteration Information%
disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCosts(it))]);
end
% Results%
figure;
%plot(BestCosts, 'LineWidth', 2);
semilogy(BestCosts, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Cost');
grid on;
```

LIST OF PUBLICATIONS

The list of publications during the research work is as follows:

1. Snigdha Chaturvedi & Narendra Kumar (2021) "Design and Implementation of an Optimized PID Controller for the Adaptive Cruise Control System", IETE Journal of Research, DOI: 10.1080/03772063.2021.2012282**(SCIE), (Published)**

2. Snigdha Chaturvedi, Narendra Kumar & Rajesh Kumar. (2023)." A PSO-optimized novel PID neural network model for temperature control of jacketed CSTR: design, simulation, and a comparative study." Soft Computing. 1-15. DOI: 10.1007/s00500-023-09138-0. **(SCIE**), (**Published**)

3. Snigdha Chaturvedi, Narendra Kumar & Rajesh Kumar "Two feedback PID tuned with teaching-learning based optimization algorithm for ball and beam system", IETE Journal of Research, DOI: 10.1080/03772063.2023.2284955 (**SCIE), (Published)**

4. Snigdha Chaturvedi & Narendra Kumar "Design and analysis of nonlinear controller for performance improvement of an artificial respiratory system", Soft Computing, Springer (**SCIE**), **(Under Review)**

5. Snigdha Chaturvedi & Narendra Kumar "Temperature Control of a Non-Jacketed CSTR through PID tuning: A comparative study of Evolutionary Algorithms and a Novel Multi-objective Function" , Journal of Dynamical and control systems **(SCIE), (Communicated)**

CONFERENCES

1. Snigdha Chaturvedi, Narendra Kumar & Rajesh Kumar "Intelligent Algorithms based Level Control of a Non-linear conical tank system", ICSCA-2023, NIT Kurukshetra, Springer Conference
2. Snigdha Chaturvedi, Narendra Kumar & Rajesh Kumar "Inverted Pendulum Cart control using Optimized FOPID Controller", ICCSAI-2022, Galgotias University, Greater Noida(IEEE)

BIOGRAPHY OF AUTHOR

Snigdha Chaturvedi (2K17/PHD/EE/07) is a PHD student at the Electrical Engineering Department, Delhi Technological University, Delhi. She has received her B. Tech degree in Electrical and Electronics Engineering from Uttar Pradesh Technical University, Lucknow in 2009 and MTech degree in Control and Instrumentation from Delhi Technological University, Delhi in 2014. She has a teaching experience of 9 years in various engineering colleges. Her research interests are optimization algorithms, non-linear control, neural networks, and intelligent control. She has published number of research papers in SCI/SCIE journals and international conferences.