PID CONTROL PARAMETER TUNING USING SWARM-BASED METAHEURISTIC ALGORITHM FOR COUPLED TANK SYSTEM APPLICATION

NUR ATHIRAH BINTI RUSLAN

BACHELOR OF ELECTRICAL ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

PID CONTROL PARAMETER TUNING USING SWARM-BASED METAHEURISTIC ALGORITHM FOR COUPLED TANK SYSTEM APPLICATION

NUR ATHIRAH BINTI RUSLAN

A report submitted
in partial fulfilment of the requirements for the degree of
Bachelor of Electrical Engineering with Honours

Faculty of Electrical Technology & Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I declare that this thesis entitled "PID CONTROL PARAMETER TUNING USING SWARM-BASED METAHEURISTIC ALGORITHM FOR COUPLED TANK SYSTEM APPLICATION is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature : athirah

Name : NUR ATHIRAH BINTI RUSLAN

Date : 21 JUNE 2024

APPROVAL

I hereby declare that I have checked this report entitled "PID CONTROL PARAMETER TUNING USING SWARM-BASED METAHEURISTIC ALGORITHM FOR COUPLED TANK SYSTEM APPLICATION", and in my opinion, this thesis fulfils the partial requirement to be awarded the degree of Bachelor of Electrical Engineering with Honours

Signature	:	asmiza
Supervisor Name	:	PUAN NUR ASMIZA BINTI SELAMAT
Date	:	21 JUNE 2024

DEDICATIONS

I would want to express gratitude to my father and mother for their encouragement and support over the process of completing this Final Year Project.

ACKNOWLEDGEMENTS

First of all, I would like to thank God for giving me the means and encouragement I needed to do my senior capstone project. I would like to thank everyone who has supported me in this endeavor from the bottom of my heart. I would like to thank Puan Nur Asmiza Binti Selamat, who is in charge of supervising this project, for all her advice and help throughout this project. Without her help, I could not complete my project. In addition, I would like to express my deepest appreciation to my parents for giving me the motivation I needed to overcome the challenges faced throughout this project. It would be a shame to ignore some of the individuals in my life who helped and inspired me during the challenges I faced in completing this project.

ABSTRACT

Coupled Tank Systems (CTS) are an important technology and are widely used across several sectors. Liquid-level management is important and often problematic in the processing sector for these coupled tank systems. Consequently, in a coupled tank system, the controller becomes the main pillar for controlling the liquid level. Traditional proportional-integral derivative (PID) controllers are widely used due to their simplicity and effectiveness. However, tuning PID parameters for optimal performance remains a challenge, especially for complex or nonlinear systems. Therefore, optimization techniques will be used as the tuning to find the controller parameter. This study investigates the use of swarm-based metaheuristic algorithms, specifically the tunicate swarm algorithm (TSA) and reptile search algorithm (RSA), to obtain the PID controller parameter for CTS applications. The objectives include designing a PID control system using these algorithms, evaluating controller performance, and comparing it with the particle swarm optimization (PSO) method. The entire implementation, such as designing, simulation, and analysis, is performed on the MATLAB R2023b platforms. The transient system performance (overshoot, settling time, peak time, rise time, and steady state error) will be evaluated and compared among these algorithms. A performance index, namely the Integral Time Square Error (ITSE), is used in this study as comparator to compare the performance of different types of PID tuning.

ABSTRAK

Sistem Tangki Berganding (CTS) ialah teknologi penting dan digunakan secara meluas merentasi beberapa sektor. Pengurusan tahap cecair adalah penting dan selalunya bermasalah dalam sektor pemprosesan untuk sistem tangki berganding ini. Akibatnya, dalam sistem tangki berganding, pengawal menjadi tiang utama untuk mengawal paras cecair. *Proportional-integral-derivative (PID)* tradisional digunakan secara meluas kerana kesederhanaan dan keberkesanannya. Walau bagaimanapun, penalaan parameter PID untuk prestasi optimum kekal sebagai cabaran, terutamanya untuk sistem yang kompleks atau tidak linear. Oleh itu, teknik pengoptimuman akan digunakan sebagai penalaan untuk mencari parameter pengawal. Kajian ini menyiasat penggunaan algoritma metaheuristik berasaskan *swarm*, khususnya *Tunicate Swarm Algorithm (TSA*) dan *Reptile Search Algorithm (RSA*), untuk mendapatkan parameter pengawal PID untuk aplikasi CTS. Objektif termasuk mereka bentuk sistem kawalan PID menggunakan algoritma ini, menilai prestasi pengawal, dan membandingkannya dengan kaedah *Particle Swarm Optimization (PSO).* Keseluruhan pelaksanaan, seperti mereka bentuk, simulasi dan analisis, dilakukan pada platform MATLAB R2023b. Prestasi sistem sementara (overshoot, masa penyelesaian, masa puncak, masa naik dan ralat keadaan mantap) akan dinilai dan dibandingkan antara algoritma ini. Indeks prestasi, iaitu *Integral Time Square Error (ITSE)*, digunakan dalam kajian inis sebagai pembanding untuk membandingkan prestasi pelbagai jenis penalaan PID.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

CTS	-	Coupled Tank System
PID	-	Proportional-Integral-Derivative
PSO	-	Particle Swarm Optimization
TSA	-	Tunicate Swarm Algorithm
RSA	-	Reptile Search Algorithm
ITSE	-	Integral Time Square Error
P	-	Propotional Term
I	-	Integral Term
D	-	Derivative Term
Kp	-	Propotional Gain
Ki	-	Intergral Gain
Kd	-	Derivative Gain
Tu	-	Ultimate Period
Ku	-	Ultimate Gain
Z-N	-	Ziegler-Nichols
C-C	-	Cohen-Coon
ACO	-	Ant Colony Optimization
ABC	-	Artificial Bee Colony
Ts	-	Settling Time
Tr	-	Rise Time
Tp	-	Peak Time
OS	-	Overshoot
SSE	-	Steady State Error
ISE	-	Integral Square Error
IAE	-	Integral Absolute Error
ITAE	-	Integral Time Absolute Error
M-ACO	-	Modified – Ant Colony Optimization
BA	-	Bat Algorithm
DE	-	Differential Evolutionary

CHAPTER 1

INTRODUCTION

1.1 Introduction & Motivation

Nowadays, process industries such as the petrochemical industry, paper manufacturing, and water treatment industries require liquids to be pumped, stored in tanks, and then flowed to other tanks. Fluid control in tanks and flow between tanks is a fundamental problem in the process industry. The above-mentioned industries are important industries where liquid level is important. As an example, in chemical engineering systems, generally the liquid will be processed by chemical treatment, and the liquid level in the tank must always be controlled and regulated. To maintain the liquid at a certain height or range, the use of a controller is required. An efficient and effective controller will optimize the operating costs. The selected control method will determine the effectiveness of controlling the height and range of the liquid level.

PID controllers are widely used in engineering due to their simple design, ability to function effectively despite model errors, and ease of use. PID controller performance depends on the suitable combination of proportional, integral and derivative parameter. PID control parameter tuning is the process of determining the ideal settings for the proportional, integral, and derivative gains of the PID controller to achieve the required level of control system performance and stability. PID control settings can be adjusted using various techniques, including metaheuristic algorithms and traditional tuning. Conventional tuning methods are based on some analytical or empirical rules, such as Ziegler-Nichols, Cohen-Coon, and trial and error. This technique is simple and easy to use, but it cannot produce sufficient results for complex or non-linear systems or for systems with uncertainties and disturbances. Metaheuristic algorithms have been widely used for PID control parameter tuning, as they can improve the performance and robustness of PID controllers. To improve its accuracy and efficiency, metaheuristic algorithms can also be used together with traditional tuning techniques.

1.2 Problem Statement

A coupled tank system is formed by connecting two tanks in series. The basic objective of operating a coupled tank system is to maintain a constant liquid level in the tank during liquid inflow or outflow. Maintaining the desired level requires control of the input flow rate[1]. The liquid level is managed using a traditional PID controller. PID controllers are the most used controllers in industrial control due to their ease of implementation. There are many existing tuning methods, such as the Zeigler-Nichols and Cohen-Coon methods[2]. However, when aggressive performance is required, this method cannot produce the desired results due to nonlinear dynamics in the system and variations in system parameters caused by orifice and tank scaling. It has been proposed to use metaheuristics to adjust PID controllers for coupled tank systems [3]. Metaheuristic algorithms are chosen for their ability to produce answers when efficient optimization techniques cannot be used due to the complexity or duration of the problem. Another advantage is that they avoid becoming stuck at local ideal levels while exploring for workable solutions[4].

1.3 Objectives

i. To design the PID control system tuned using tunicate swarm algorithm (TSA) and reptile search algorithm (RSA) for coupled tank system application.

ii. To evaluate the performance of PID controller with the implementation of tunicate swarm algorithm (TSA) and reptile search algorithm (RSA).

iii. To compare system performance of PID controller with the implementation of the particle swarm optimization (PSO), tunicate swarm algorithm (TSA) and reptile search algorithm (RSA).

1.4 Project Scopes

i. Designing and simulating PID controller using MATLAB/Simulink R2023b for coupled tank system application.

ii. Use mathematical model of the coupled tank system from [2].

iii. Apply performance index, Integral Time Square (ITSE) to obtain the error of the system and as the indicator of system performance.

iv. Use tunicate search algorithm (TSA), reptile search algorithm (RSA) and particle swarm optimization (PSO) method to tuned the PID controller for coupled tank system application.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This section identifies and summarizes a variety of research that pertains to the coupled tank system application, the PID controller, PID tuning, and swarm-based metaheuristics algorithms, which are Particle Swarm Optimization (PSO), the Tunicate Swarm Algorithm (TSA) and the Reptile Search Algorithm (RSA). To further understand the research that has been carried out in the past by other researchers, the studies will be critically evaluated and briefly described. In this section, the previous work that was relevant to the project, such as the theory, analysis, synthesis, and evaluation, will be discussed.

2.2 Coupled Tank System

The coupled tank system is a classic system of two tanks connected by a pipe with a valve that can be opened or closed. Liquid level control is crucial for preventing overflows in industries where liquid levels are needed. The coupled tank is one of the most popular technologies in industrial control operations because this system is used to study concepts of fluid flow, pressure, and level control.

A coupled tank system is a hydraulic system with two or more tanks or vessels connected by pipes or ducts. The liquid moves between the two connected tanks that make up the system. Each tank has its own inlet and outflow. The fundamental idea behind this system's regulation is to maintain a steady level of liquid in both tanks during the inflow and outflow of liquid from each tank. The mathematical model of the connected tank system was developed and assessed as a type of linear model in order to regulate the liquid level[1].

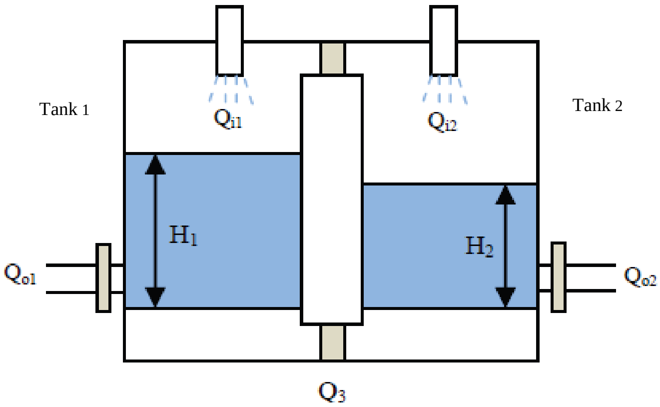

Figure 2.1 Schematic model of coupled tank system[5] .

2.3 PID Controller

PID (Proportional-Integral-Derivative) controllers have been widely used in control applications due to their simplicity and efficiency. It improves the transient response and steady state of the system. Whenever the apparent state of the plant deviates from the reference state, an adjustment estimate is calculated using the words proportional, integral and derivative. The actuation signal from a PID controller that uses the error as input in the Laplace transform is expressed mathematically as [6]:

$$u(s) = K_P + K_I \frac{1}{s} + K_D s$$

where $K_P, K_{D,} K_I$ are the PID parameters. The characteristics of the controller variables P, I, and D are introduced and explained in brief as follows:

Proportional term (P) : The closed-loop time constant decreases with the proportional parameter, which speeds up the response. However, it also guarantees that the order of the system remains unchanged because the output is only proportional to the input. However, proportional parameters do not eliminate offset or steady-state inaccuracies. Checking the amount of error and response of a proportional PID controller is the primary responsibility of the proportional element.[7]

Integral term (I) : This option increases system properties and order by one while removing offsets. The response speed of the system is also increased by this parameter, although at the expense of continuous oscillation. Integral control seeks to reduce the problems caused by proportional control. It tracks inaccuracies over time and gradually increases the size of small errors. The term "rerate" refers to the time component used to indicate integrated control modifications.[7]

Derivative term (D) : In essence, the oscillatory response of the system is decreased with this setting. It has no bearing on the type and order of the system, nor does it affect the offset. It looks into the rate of variation of the error signal. The derivative is what causes the A bigger system's response to alter at a quick rate. With time, the derivative word gets modified. Excessive usage of derivative terms might lead to unpredictable behaviour or overshoot.[7]

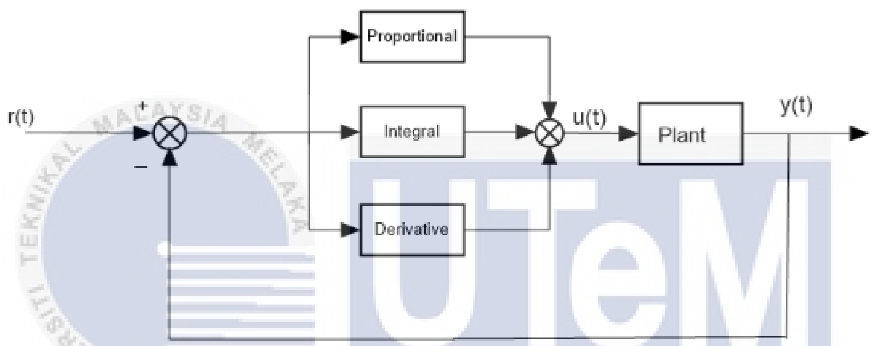

Figure 2.2 Structure of the PID controller.[4]

2.4 PID Control Tuning

There are multiple methods of tuning PID controllers, and they fall into the categories of conventional techniques and optimization techniques.

Conventional techniques include making assumptions about the intended plant and output to determine controller settings. This method aims to determine a specific process through analysis and graphics. This method is quick and easy to use, but due to various assumptions, the set of controller parameters may not always give the desired result; therefore, initial controller parameter setup is required[6]. The most popular conventional tuning techniques for PID controllers are Ziegler and Nichols (Z-N). It invented in the 1940s by John G. Ziegler and Nathaniel B. Nichols. In the method, the K_I and K_D gains are required to set to zero before the tuning procedure. During the period of oscillation, T_U, where K_P is increased to the ultimate gain, K_U, the output of the loop begins to change.

Aside from the Z-N tuning method, the Cohen-Coon tuning method is the second most used and was published in 1953 by Cohen and Coon. Cohen-Coon has an advantage over Z-N method when it involves a larger operating range. Other method to determine the PID parameter values is trial and error. It is one of the easiest methods compared to others because there is no mathematical calculation required. However, an experienced practitioner is required to obtain the ideal parameter value for the controller. In the method, it is necessary to set the K_I and K_D values to zero before increasing the K_D.[2]

Optimization techniques are used to determine the best result in a given situation. The terms "maximize" and "minimize" are used for different aspects of optimization. This method is used for differentiable and continuous functions to determine the best solution in constrained or in maxima or minima. This approach finds the best answer by using differential calculus techniques. This method could be used for single variable and multivariable functions. Some examples of traditional optimization techniques are stochastic programming, dynamic programming, linear programming, and calculus methods. These traditional optimization methods have been further used to propose new algorithms, as listed in [7]. The disadvantages of conventional optimization, due to limited non-differentiable and discontinuous functions, have been addressed by advanced optimization techniques. Proposed annealing, evolutionary algorithms, genetic algorithms, particle swarm optimization (PSO), ant colony optimization (ACO), and artificial bee colony (ABC) are some examples of advanced optimization approaches [6].

2.5 PID Control Tuning for Coupled Tank System

A coupled tank system is a device that has two tanks connected by pipes, and each tank's liquid level is managed by a pump or valve. A PID controller is a type of feedback controller that modifies the input signal according to the difference between the intended and actual output signal. PID is the term for proportional, integral, and derivative, the three terms that comprise the equation for the controller.[8]

PID control tuning is the process of determining ideal PID parameter values (Kp, Ki, and Kd) to reduce errors and improve system performance. PID control tuning can be done in several ways, including conventional techniques such as automatic tuning, Ziegler-Nichols, Cohen-Coon, and optimization techniques such as PSO. Each technique has varying benefits and drawbacks based on system requirements and characteristics.[9]

Table 2.1 and Figure 2.3 compares several previous studies on the performance of various PID tunings using traditional approaches, namely Z-N[3][8], auto-tuning[8], trial and error[8], and for coupled tank systems. Ts, Tr, OS, ISE, and IAE are the performance measures used to measure the system performance of the tuning method. Lower values of these metrics indicate better performance, except for IAE, which is higher for better performance. The Z-N techniques have the fastest and smoothest responses since their Ts and OS values are the lowest. The trial-and-error and auto-tuning approaches have the slowest and most oscillatory responses because their Ts and OS values are the greatest.

Table 2.1 Dynamic Performance Comparison of different conventional tuning method.

Method	**Settling Time, Ts (s)**	**Rise Time, Tr (s)**	**Overshoot, OS (%)**	**ISE**	**IAE**
Z-N[3]	-	46.5	22.7	35.19	69.56
Z-N[8]	32.09	3.29	38.54	-	-
Auto-tuning[8]	53.34	9.14	**1.81**	-	-
Trial and Error[8]	84.4	24.03	6.86	-	-
C-C[8]	**23.59**	**2.81**	33.7	-	-

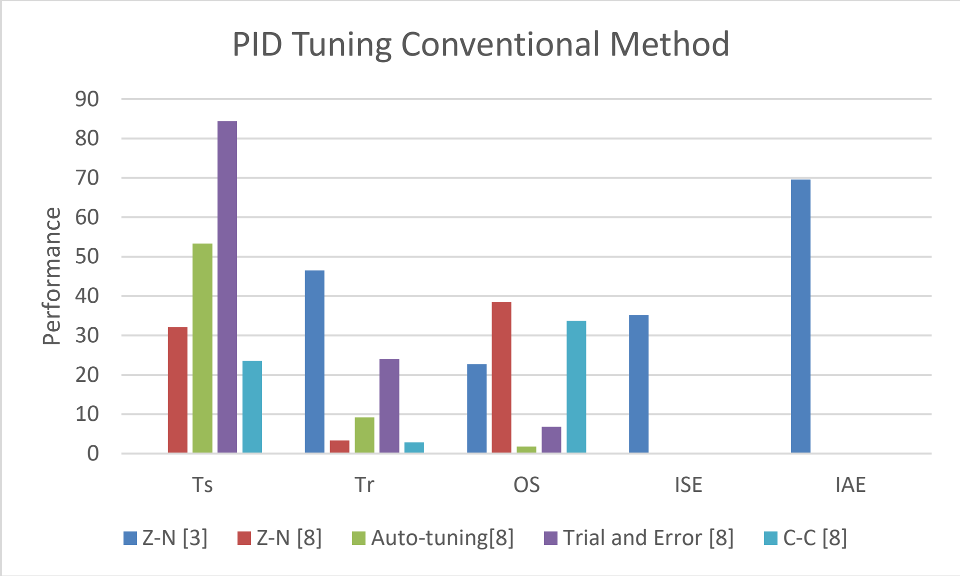

Figure 2.3 Dynamic Performance Comparison of different conventional tuning method in bar graph.

According to Table 2.2 and Figure 2.4, five PID tuning optimization methods, namely PSO[3][8][10][11], M-ACO[3], BA (ITSE)[9], BA (IAE)[9],BA (ISE)[9] and DE[11], from previous studies are compared. The PSO approach exhibits the best performance in Tr and Ts, showing the fastest response time to reach the setpoint. With the least amount of overshoot or variation from the setpoint, the BA (ITSE) approach shows the best performance in OS. With the lowest cumulative error over time, the BA (IAE) technique performed best in both ISE and IAE. The M-ACO and DE approaches performed relatively poorly across the board.

Table 2.2 Dynamic Performance Comparison of different optimization tuning method.

Method	**Settling Time, Ts (s)**	**Rise Time, Tr (s)**	**Overshoot, OS (%)**	**ISE**	**IAE**
PSO[3]	-	12.13	0.8	3.01	6.47
M-ACO[3]	-	10.58	**0**	**2.58**	**5.37**
PSO[8]	17.75	3.27	16.19	-	-
BA (ITSE) [9]	8.73	1.18	34.8	-	-

BA (IAE) [9]	45.35	27.85	0.024	-	-
BA (ISE) [9]	3.68	0.62	4.09	-	-
PSO[10]	**1.46**	**0.22**	11.6	-	-
PSO[11]	10.3	1.52	-	-	-
DE[11]	8.19	2.9	-	-	-

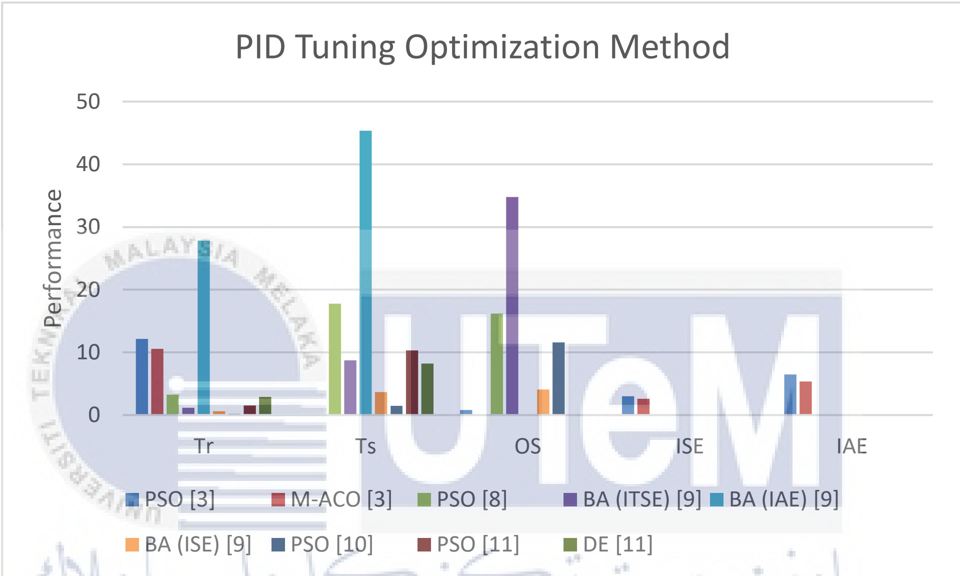

Figure 2.4 Dynamic Performance Comparison of different optimization tuning method in bar graph.

2.6 Metaheuristic Algorithm

Metaheuristic optimization algorithms have gained much popularity and use due to their advantages, including nonlinear search space, easy-to-understand concepts, simple implementation, independence of problem types, and efficiency in non-linear and non-convex environments[12]. Metaheuristics work by iteratively improving the solution to a problem. They start with a first answer and then refine it based on a collection of guidelines or heuristics. Finding a better solution than the existing one is the goal. This procedure is repeated until a workable resolution is reached.

These algorithms are inspired by natural phenomena such as evolution, swarm, and learning. One of the major advantages of metaheuristic algorithms is their ability to provide near-optimal solutions in a reasonable amount of time. Unlike other traditional optimization methods, metaheuristic algorithms do not require any assumptions about the problem's structure, making them suitable for a wide range of applications. They are also highly adaptable can be easily modified to suit specific problem requirements. Additionally, metaheuristic algorithms can handle large and high-dimensional data sets, making them various fields such as engineering, finance, and medicine. These have made metaheuristic algorithms essential tool for solving complex optimization problems.

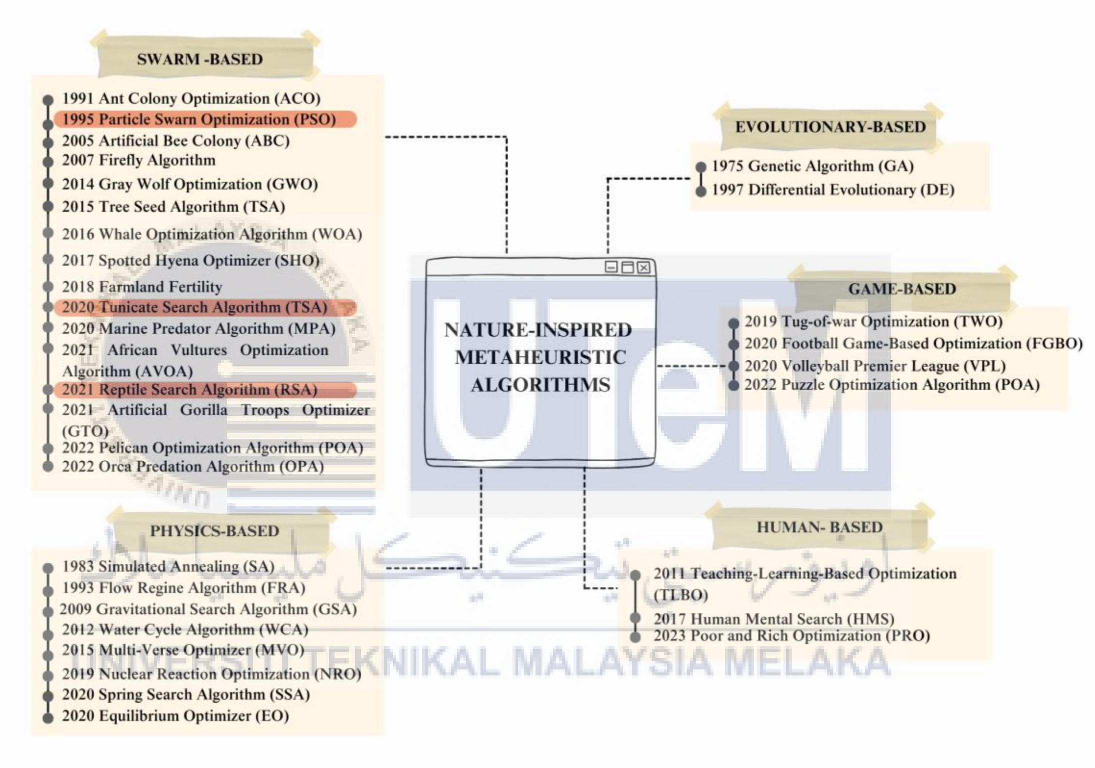

Figure 2.5 Nature-inspired metaheuristic algorithms.

Meta-heuristic algorithms were inspired by natural phenomena such as wildlife, animals, birds, insects, plants, living beings, physical laws, biological sciences, genetics, game rules, human activities, and other natural evolutionary processes. Metaheuristic algorithms may be categorized into five types according to the main source of inspiration for the design: swarm-based, evolutionary-based, physics-based, game-based, and human-based techniques.[12]

2.6.1 Evolutionary-based metaheuristic algorithm

Recently, evolutionary-based metaheuristic algorithms based on genetics, biological science, and random operator simulations have been introduced. Differential Evolutionary (DE) and Genetic Algorithm (GA) are two of the most popular and extensively utilized evolutionary algorithms. The mathematical modelling of the reproduction process, the idea of natural selection, and the use of random operators for crossover, mutation, and selection have all contributed to the development of GA and DE.[12]

2.6.2 Physics-based metaheuristic algorithm

Physics-based metaheuristic algorithms have been developed on the foundation of mathematical modelling of various physical laws and phenomena. Two popular physics-based algorithms are the Gravitational Search Algorithm (GSA) and Simulated Annealing (SA). The physical process of melting and then cooling metals—known as annealing in metallurgy—is the foundation of SA. The primary source of inspiration for the creation of GSAs has been the modelling of Gravitational Forces in a system including objects with varying masses and separations from one another. The Water Cycle Algorithm (WCA) was designed with inspiration from the physical phenomena of the water cycle and its changes in nature. The Multi-Verse Optimizer (MVO) was designed primarily with influence from cosmological principles. Some other physics-based methods are as follows: Flow Regime Algorithm (FRA), Nuclear Reaction Optimization (NRO), Spring Search Algorithm (SSA), and Equilibrium Optimizer (EO).[12]

2.6.3 Game-based metaheuristic algorithm

Game-based metaheuristic algorithms have been created based on simulations of the laws governing various games and the actions of players, coaches, and other persons who have an impact on the games. Football League: The primary concept behind the Football Game-Based Optimization (FGBO) and Volleyball Premier League (VPL) algorithms, respectively, was the creation of modelling contests in the volleyball league. The primary source of inspiration for the Puzzle Optimization

Algorithm (POA) design was the players' tactics and dexterity in piecing together puzzle pieces. The Tug-of-war Optimization (TWO) technique was primarily inspired by the tug-of-war players' efforts.[12]

2.6.4 Human-based metaheuristic algorithm

Human-based metaheuristic algorithms are introduced based on mathematical modelling of diverse human behaviours with an evolution-based approach. The most well-known human-based algorithm, Teaching-Learning-Based Optimization (TLBO), was created by simulating teacher-student interaction and communication in a classroom. The primary concept behind Poor and Rich Optimization (PRO) has been the economic actions of the wealthy and the impoverished in society. Human Mental Search (HMS) is based on the simulation of human behaviour versus online auction marketplaces to achieve success.[12]

2.7 Swarm-based metaheuristic

Swarm intelligence algorithms are metaheuristics inspired by the collective behaviour of species such as birds, fish, bees, and ants. They have simplicity, flexibility, and scalability, which makes them useful in a wide range of optimization issues. These algorithms balance the exploration and exploitation phases to achieve the required convergence during the search. Numerous fields, including global optimization, bioinformatics, power engineering, networking, machine learning, image processing, and environmental applications, can benefit from the use of these metaheuristics.[13]

Popular algorithms include Artificial Bee Colony (ABC), Firefly Algorithm (FA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). The primary concept in PSO design has been the natural behaviour of a swarm of fish or birds searching for food, with their movement impacted by swarming intelligence and individual experiences. The FA design takes advantage of mathematical modelling of the firefly's natural light-flashing characteristic. The main idea behind ABC design is to mimic the intelligence of swarming bee colonies in their search for nourishment. The primary concept behind the ACO's design has been the capacity of an ant colony to choose the quickest route between itself and food sources.

Many metaheuristic algorithms, including the Tunicate Search Algorithm (TSA), Reptile Search Algorithm (RSA), Whale Optimization Algorithm (WOA), Orca Predation Algorithm (OPA), Marine Predator Algorithm (MPA), Pelican Optimization Algorithm (POA), Snow Leopard Optimization Algorithm (SLOA), Gray Wolf Optimization (GWO) algorithm, Artificial Gorilla Troops Optimizer (GTO), African Vultures Optimization Algorithm (AVOA), Farmland Fertility, Spotted Hyena Optimizer (SHO), and Tree Seed Algorithm (TSA) have all been inspired by hunting and attacking prey strategies and the process of finding food sources among living organisms.[12].

Table 2.3 Previous study for swarm-based metaheuristics.

ALGORITHM	REF	FINDING	SYSTEM
Particle Swarm Optimization (PSO)	[14]	The outcomes of the simulation showed that PSO works better than the SA, TSA, and SDP methods.	Wireless system network
	[15]	The PSO approach can more rapidly and simply handle the finding and tuning issues related to PID controller settings compared to the GA technique. It also has more robust stability and efficiency.	PID controller parameter optimization
	[16]	Comparing the BF-PSO optimization algorithm based PID tuning approach to other conventional techniques, it is determined to provide good transient performance.	HVAC system
Tunicate Search Algorithm (TSA)	[17]	TSA has a more efficient optimum solution than PSO.	IEEE 8-bus test system
	[18]	TSA-tuned PID speed control performed better for the specified PMSM with reduced settling time, peak overshoot, and ripple than traditional and PSO approaches.	Permanent Magnet Synchronous Motor (PMSM)
	[19]	When it comes to the damping ratio, the TSA-based approach outperforms the BSA-based approach, and it also requires less time to adjust the controller settings.	Single machine infinite bus (SMIB) network incorporated with a unified power flow controller (UPFC)

Reptile Search Algorithm (RSA)	[20]	The RSA-tuned PID controller offers best transient response metrics compared to BFO, GSA, and PSO	Doubly fed induction generator-based wind turbine energy conversion system
	[21]	The RSA method is far more accurate in identifying the best solutions, and it is thus advised for the adoption of large-scale distribution systems compared to MTLBO, JAYA, GWO, and IRRO approaches,	Radial distribution system
	[22]	An investigation of run times is also conducted, and the results show that PSO is the quickest method, followed by RSA and GA. RSA often has more fitness than PSO and GA, according to performance analysis tools like the DB-index. It is therefore clear that although while PSO requires less computing time, its performance is not as good as that of RSA. As a result, RSA is a promising technique for image segmentation.	Side Scan Sonar (SSS) image object detection
Whale Optimization Algorithm (WOA)	[23]	Proposed WOA-based solutions outperform GA in terms of convergence rate, and optimum solutions are reached on efficient frontiers.	Portfolio selection problem
	[24]	Those benchmark tests show that WOA is more effective. However, there are a few areas that still require improvement	Dendritic neuron model (DNM)
	[25]	In terms of the amount of time needed to prioritize various size sets of needs, the RP-WOA performs around 40% better than the AHP technique.	Requirements prioritizations (RP)
Marine Predator Algorithm (MPA)	[26]	This research demonstrated that the MPA algorithm computes the flatness error more quickly than the GA and PSO approach when comparing the computation results. The computed results exhibit more stability and are capable of efficiently assessing the flatness error.	Flatness error evaluation
	[27]	The MPA provides the best fuel cost values. The MPA algorithm produced a superior result with the help of the given objective function.	Economic Load Dispatch with Valve-Point Loading Effects

	[28]	In terms of error rates, transient response, steady state response, and convergence performance, the suggested MPA:PID controller outperforms the TLBO:PID and SCA:PID controllers.	PID Controller for Frequency Regulation of Standalone Microgrid
Pelican Optimization Algorithm (POA)	[29]	The original gradient dependent line search approach is outperformed by the NSPOA method in terms of global convergence, time efficiency, and iteration speed compared to the GD and SGD methods.	Locally Linear Embedding
	[30]	The faults of POA, such as the tendency to slip into local optimization and the decline in swarm diversity at the conclusion of the iteration, are significantly improved by MSPOA. When comparing the most recent state-of-the-art algorithms with the traditional meta-heuristic algorithms, MSPOA comes out on top.	Mixed strategy-based improved Pelican Optimization Algorithm (MSPOA)
	[31]	The jamming resource allocation optimization problem is solved by the enhanced POA method. In addition to providing a suitable allocation method and locating the global optimal solution inside the solution space, it also introduces a novel notion for this type of WTA discrete optimization issue.	Jamming resource allocation model
Gray Wolf Optimization (GWO)	[32]	When compared to previous methods, GWO may always reduce the overall cost. Therefore, when compared to other inquiries in the article, GWO using the 14 and 30 IEEE frameworks obtains lower values for the objective function and transmission loss.	Optimal power flow (OPF)
	[33]	A high-speed CPU may be used to execute the GWO technique, and the optimized gain values can be updated in the SMC controller. The more search agents and repetitions there are, the more accurate the technique becomes.	Sliding Mode Control
	[34]	(GWO) method performs better in makespan when using the same data separately.	Integrating of process planning and scheduling (IPPS)

Artificial Gorilla Troops Optimizer (GTO)	[35]	GTO beat the AHA algorithm and offers performance that is extremely competitive.	Optimal power flow (OPF) problems considering stochastic wind power
	[36]	The AGTO algorithm outperforms HBA, SMO, ABC, and PSO in terms of outcomes.	Wind farm integrated system
	[37]	The hybrid microgrid system with the optimal sizing is GTO, not ALO or GWO. The convergence characteristics showed that the GTO quickly arrives at the best solution.	Design an optimal sizing of a microgrid system
African Vultures Optimization Algorithm (AVOA)	[38]	In a short number of iterations, AVOA converges quickly and provides the best results for a variety of filter settings, including maximum SBA, less PBR, lower SBR, and TW.	Performance Analysis of Optimal FIR LPF and HPF
	[39]	After distributed generation (DG) access, voltage distribution may be improved by AVOA, which has a quick convergence time and robust optimization properties.	Optimal Planning of Distributed Generation
	[40]	When compared to GWO and PSO, the AVOA algorithm has demonstrated better qualities including accurate solution, consistent convergence characteristic, and high computing efficiency.	Optimal power flow solution including SVC devices
Spotted Hyena Optimizer (SHO)	[41]	Optimal cost, little variance, and less losses are found with the SHO. When compared to other meta heuristic techniques, this strategy achieves an optimal cost.	Dynamic Economic Dispatch (DED)
	[42]	An effective optimizer for analyzing exploration and exploitation is the SHO algorithm. The outcomes of engineering design issues show how the SHO method may be applied with little processing effort in high-dimensional environments.	25-bar truss design and multiple disk clutch brake design
	[43]	SHO demonstrated a faster rate of convergence, improved fuel economy, superior computing capability, and more thoughtful accomplishment. By employing the best fuel selections and optimal generation outputs, the SHO would	Complex Economic Dispatch (ED) problem

		be a workable way to provide a power grid with the best possible scenario to optimize dependability and decrease costs.	
Tree Seed Algorithm (TSA)	[44]	When compared to PSO-based RBFN, the TSA offers more precise numerical function mapping and, hence, superior fitness values.	Radial Basis Function Networks (RBFN)
	[45]	On the five benchmark functions, the impact of boundary conditions mechanisms on TSA performance is examined. When it comes to selecting boundary conditions for optimizing numerical benchmark functions, the TSA is resilient.	Search space limitation techniques
	[46]	When compared to the current techniques, the suggested AVR system with the TSA optimized PID controller performs better in terms of both transient and steady state responses of its voltage tracking performance.	Automatic voltage regulator (AVR) system

2.8 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique introduced in 1995 by Kennedy and Eberhart. PSO is based on the social biological and social-cognitive behaviour of various group living species such as birds and fish, which have been shown to share knowledge to promote group survival to achieve common goals. Each particle adjusts its position based on its own experience and the experience of its neighbours. Working together allows group members to share the greatest knowledge, which can be used to determine optimal hunting locations. Unlike other optimization techniques, this technique requires only an objective function and does not rely on the gradient of the goal or any other form of differentiation.[47]

The PSO algorithm's definition of a group of particles exhibits stochastic behaviour, as each particle in the group is constantly updating its position based on velocity. Velocity is updated according to each particle's memory, which is like autobiographical memory, and according to the group's collective knowledge, which is similar to learning from others. Both purposes are specifically expressed in terms of the social and cognitive dimensions of speed updates. The cognitive parameter determines how confident the particle is in its initial judgment. The social parameter determines the extent to which each particle in the group trusts the other particles. The social behaviour of the swarm, which constantly adapts to its environment to find a better placement over time (e.g., iteration), is what updates the position of particles in the swarm.[47]

2.9 Tunicate Swarm Algorithm (TSA)

One of the latest swarm-based metaheuristic algorithms, the Tunicate Swarm Algorithm (TSA), was originally developed by S. Kaur, L.K. Awasthi, and G. Dhiman [26] for non-linear restricted problems. It mimics the behaviour of tunicates during navigation and hunting for food sources. Using it on 74 benchmark issues involving a wide variety of functions demonstrates its performance. TSA differs from various competing algorithms. When handling high-dimensional and complex situations, TSA is prone to getting stuck at local optima, just like other metaheuristic techniques. As a result, TSA performance can still be improved to deal with complex and real-world issues such as the economic delivery dilemma.[48]

A cylinder-like animal with one of its two ends opens the tunicates move through the water at a speed similar to a jet [33]. Even if they don't know where to hunt for food first, they may still look for it in the waters. TSA's optimization approach is based on swarm intelligence and propulsion such as jet tunicates. The best solution to the TSA optimization puzzle is found in the food supply. To properly demonstrate the motion of the TSA jet thruster, several requirements must be met. Before anything else happens, two things have to happen: Tunicates must, first, avoid conflict and, second, continue their path to run their best search agents. Finally, they must stay close to the agent. Other tunics in the mathematical model use swarm intelligence to adjust their placement with respect to the best solution.[49]

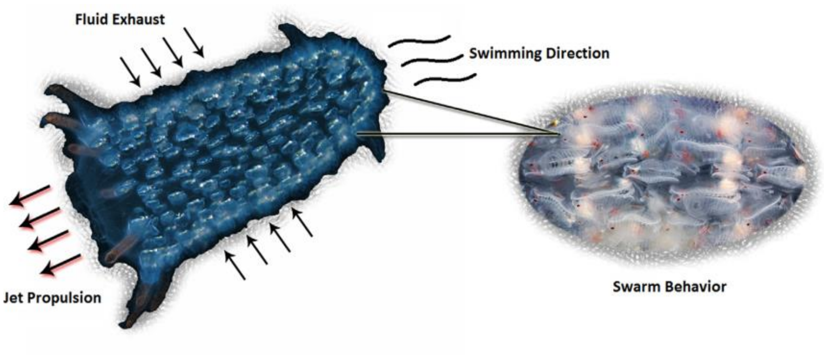

Figure 2.6 Swarm behaviour of tunicate in deep ocean. [50]

2.10 Reptile Search Algorithm (RSA)

The Reptile Search Algorithm (RSA) was created based on the dynamics of the natural surroundings, hunting process and social behaviour of crocodiles. The development of random population groups is the first step in the optimization process. RSA searches every potential place for a near-optimal solution throughout the iterative process. Each solution replaces the location it moves away from the ideal solution according to the proposed RSA approach.[21]

Under some restrictions, RSA's population-based and gradient-free methodology can help with both basic and complicated optimization issues. Crocodiles that belong to cohesive groups are more resilient and actively cooperate with one another. Crocodiles hunt mostly at night and have superb night vision. They take advantage of the deficiencies of their prey, such as poor night vision.[21]

In short, crocodiles are among the most intelligent and skilled hunters—perhaps second only to humans. Mathematical optimization, which selects the best option given constraints, is used to describe crocodile behaviour. Advances in search techniques have sparked the interest of researchers in various sectors. Optimization difficulties appear in a wide range of quantitative disciplines, spanning from engineering, economics, and computer science to operations and industrial research. Surrounding and chasing prey serves as the inspiration for the proposed algorithm (RSA).[21]

2.11 Performance Index

The cost function is extracted and used to determine the optimal controller. The cost function is mostly determined by the controller's response to certain disturbances. We can define the infinitive criterion in reality. It is a quantitative indicator of system performance chosen to highlight critical system requirements. Criterion measurements based on the integral of the control error function and possibly additional variables (such as time) form the performance index. Control loop performance is improved by having a smaller integral criterion value.[51] Several popular performance indicators include: Integral Square Error (ISE), Integral Absolute Error (IAE), Integral Time Square Error (ITSE) and Integral Time Absolute Error (ITAE) and given in equation:

$$ISE = \int_0^t (e(t))^2 \, dt$$

To eliminate negative error components, ISE squares the error. ISE distinguishes between systems that are under- or over-damped; a compromise is made to reduce the ISE.[52]

$$IAE = \int_0^t |e(t)| \, dt$$

To eliminate negative error components, IAE obtains the absolute value of the error. IAE is often beneficial for simulation research.[52]

$$ITAE = \int_0^t |e(t)| t \, dt$$

The ITAE emphasizes error values later in the response rather than early large errors because it weighs errors over time.[52]

$$ITSE = \int_0^t (e(t))^2 t \, dt$$

In contrast to ITAE's performance standards, ITSE places greater emphasis on error values and weighs the squared error over time.[52]

Where e(t) is the error signal in a time domain.

2.12 Chapter Summary

This study focuses on the tuning of PID control parameters in a coupled tank system using three metaheuristic algorithms: Particle Swarm Optimization (PSO), Tunicate Swarm Algorithm (TSA), and Reptile Search Algorithm (RSA). The goal is to optimize proportional, integral and derivative gains for efficient liquid level control in industrial processes, leveraging the strength of these algorithms in dealing with complex optimization problems and potential system uncertainties.

Table 2.4 lists several controllers that have been used in coupled tank systems in past studies. These controllers include sliding mode control[53][54], inverted decoupling controller[55], internal model control (IMC)[55], PID[10] and LQR[10]. In addition, the PID tuning technique is one of the tuning techniques frequently applied to coupled tank systems[56][57][58]. Lastly, the coupled tank system now successfully increases system performance through the implementation of optimization techniques including modified-ACO[3], PSO[8][10], bat algorithms[9] and different evolution[11].

Table 2.4 Previous study for coupled tank system.

TOPIC	REF	FINDING	SYSTEM
Coupled tank system	[53]	It was possible to effectively develop second order sliding mode control for a nonlinear linked tank liquid level system with a variety of input circumstances.	Coupled tank system
	[55]	Two controllers are used to a coupled-tank liquid level system: an inverted decoupling controller and an internal model control decoupling controller. It has been discovered through modelling and experimental research that the inverted decoupling strategy is more reliable than internal model control.	Coupled tank system
	[54]	Second order SMC Sub-optimal and Drift HOSMC is considered to regulate coupled tank systems. Using the drift method, tank two filled more quickly, and the sub-optimal algorithm used less time than the first order SMC.	Coupled tank system

	[10]	System performance is improved by LQR compared to PID. The better overall transient response of LQR than PID supports the conclusion.	Coupled tank system
Coupled tank system with PID controller	[56]	FOPID performs significantly better than IOPID under IAE performance metrics. IOPID performs better than FOPID under the ISE performance metric and both controllers perform almost equally under the ITSE performance metric.	Coupled tank system
	[57]	AMPC shows better performance in reaching the reference value when compared to FOPID and PID controllers. For nonlinear industrial processes, the AMPC approach results in superior undershoot and response time.	Coupled conical tank system
	[58]	The differences between FOPID and traditional PID are compared, and the best performance is determined by analyzing the rising, peak, settling, and peak overshoot timings.	Spherical coupled tank system
Coupled tank system with PID controller tuning using optimization	[3]	The findings show that the performance of m-ACO-tuned PID is better than previous research.	Coupled tank system
	[8]	PID-tuned by PSO performed better than traditional approaches and successfully reduced the values of Ts, Tr, OS, and SSE	Coupled tank system
	[9]	Better results are achieved with the BA-optimized PID controller, with a lower percentage of overshoot, an increase in value and solution time. This is especially true when using ISE with objective functions.	Coupled tank system
	[10]	PSO may be used to adjust both controllers, with LQR providing slightly higher performance than PID.	Coupled tank system
	[11]	The results achieved with DE, whether with TSI or existing, are better than those achieved with other methods. TSI helps in improving the ideal value produced by the current algorithm.	Coupled tank system

Figure 2.7 shows the outcomes of PID tuning using both conventional methods and Particle Swarm Optimization (PSO) for coupled tank system . These results analyse various transient response characteristics, such as rise time (Tr), settling time (Ts), peak time (Tp), overshoot (%OS), steady-state error (ess), and integral time square error (ITSE).

The tuning methods considered are trial and error, Ziegler Nichols, Cohen Coon, autotuning, and PSO. The controller parameters in Table 2.5 for trial and error, Ziegler

Nichols, Cohen Coon, and autotuning are based on [2] while PSO-based PID parameters are referred from [8].

Table 2.5 Parameter of PID Controller

Method	**Parameter**		
	Kp	**Ki**	**Kd**
Trial and error	15.00	1.00	8.00
Z-N	168.00	35.00	201.60
Cohen Coon	235.88	33.92	203.21
Autotuning	53.40	1.54	-2.98
PSO	250.99	4.35	171.64

Table 2.6 and Figure 2.8 show the response characteristics for each tuning method are compared, and PSO shows the best rise time (3.27 s) and settling time (17.75 s). Although PSO has a high overshoot (16.19%), its peak time is 6.40 s, and the ITSE is the smallest (12.84). Consequently, PSO is selected as the benchmark for the TSA and RSA algorithms.

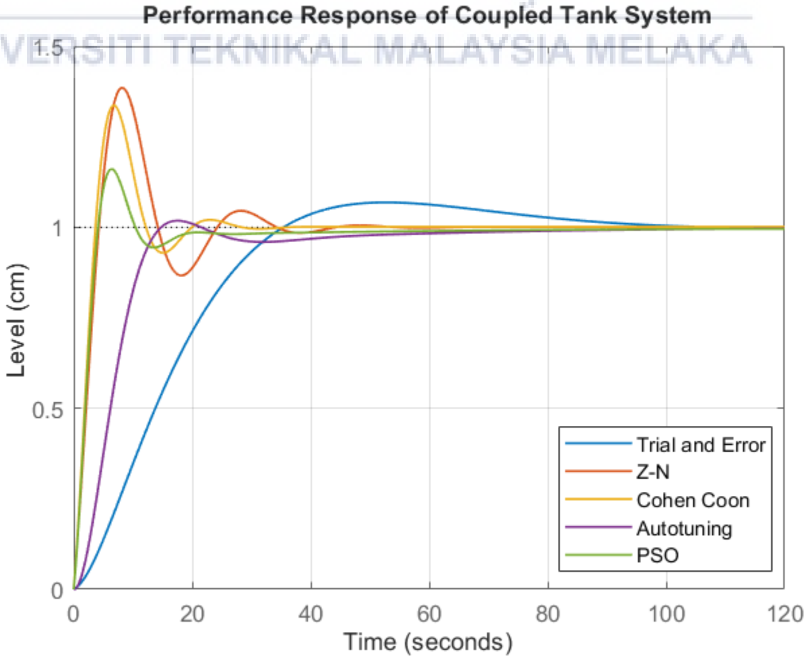

Figure 2.7 Performance response of coupled tank system for all controller.

Table 2.6 Performance of coupled tank system based on conventional method and PSO.

	$T_P(sec)$	T_s (sec)	T_R (sec)	OS (%)	SSE (cm)	ITSE
Trial and error	52.7	84.40	24.00	6.86	0.00	203.10
Z-N	7.9	32.10	3.29	38.50	0.00	22.03
Cohen Coon	6.7	23.59	**2.81**	33.70	0.00	17.36
Auto-tuning	17.70	53.40	9.14	**1.81**	0.00	46.70
PSO	**6.40**	**17.75**	3.27	16.19	0.00	**12.84**

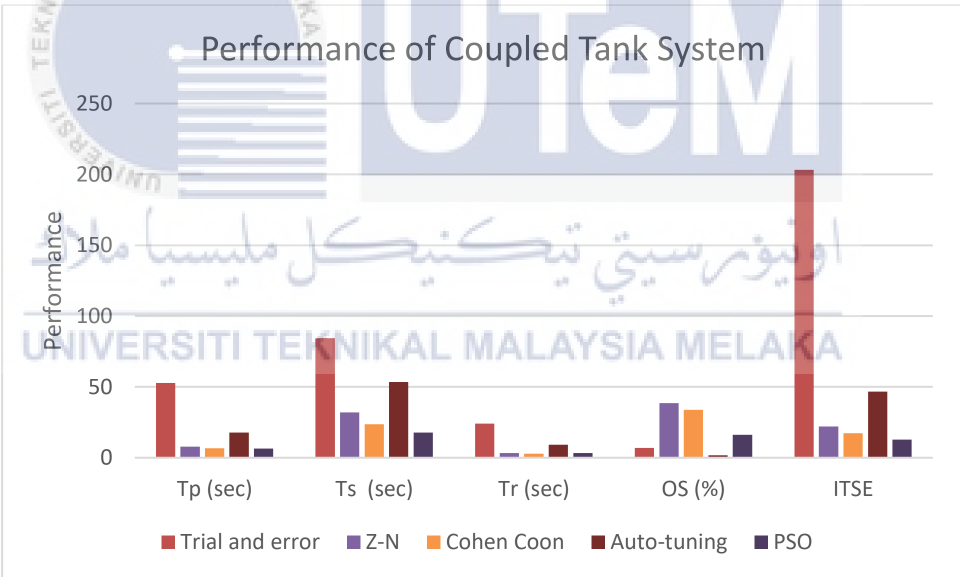

Figure 2.8 Performance of coupled tank system based on conventional method and PSO in terms of bar graph

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses the entire scope of the project, beginning with the mathematical design of the Coupled Tank System (CTS), PID tuning using optimization techniques and performance evaluation of all the techniques.

3.2 Flowchart of Research Activity

A literature review was conducted to improve understanding of the project according to previous studies after the selection of the title. The literature has been completed using the latest research on the subject, which includes optimization techniques, PID tuning methods and other related topics. Figure 3.1 illustrates the flowchart of research activity.

The project’s parameter controller will undergo tuning using optimization techniques, including Tunicate Swarm Algorithm (TSA), Reptile Search Algorithm (RSA) and Particle Swarm Optimization (PSO). TSA and RSA represent the latest algorithms developed by S. Kaur, L.K. Awasthi, and G. Dhiman.[50][59]

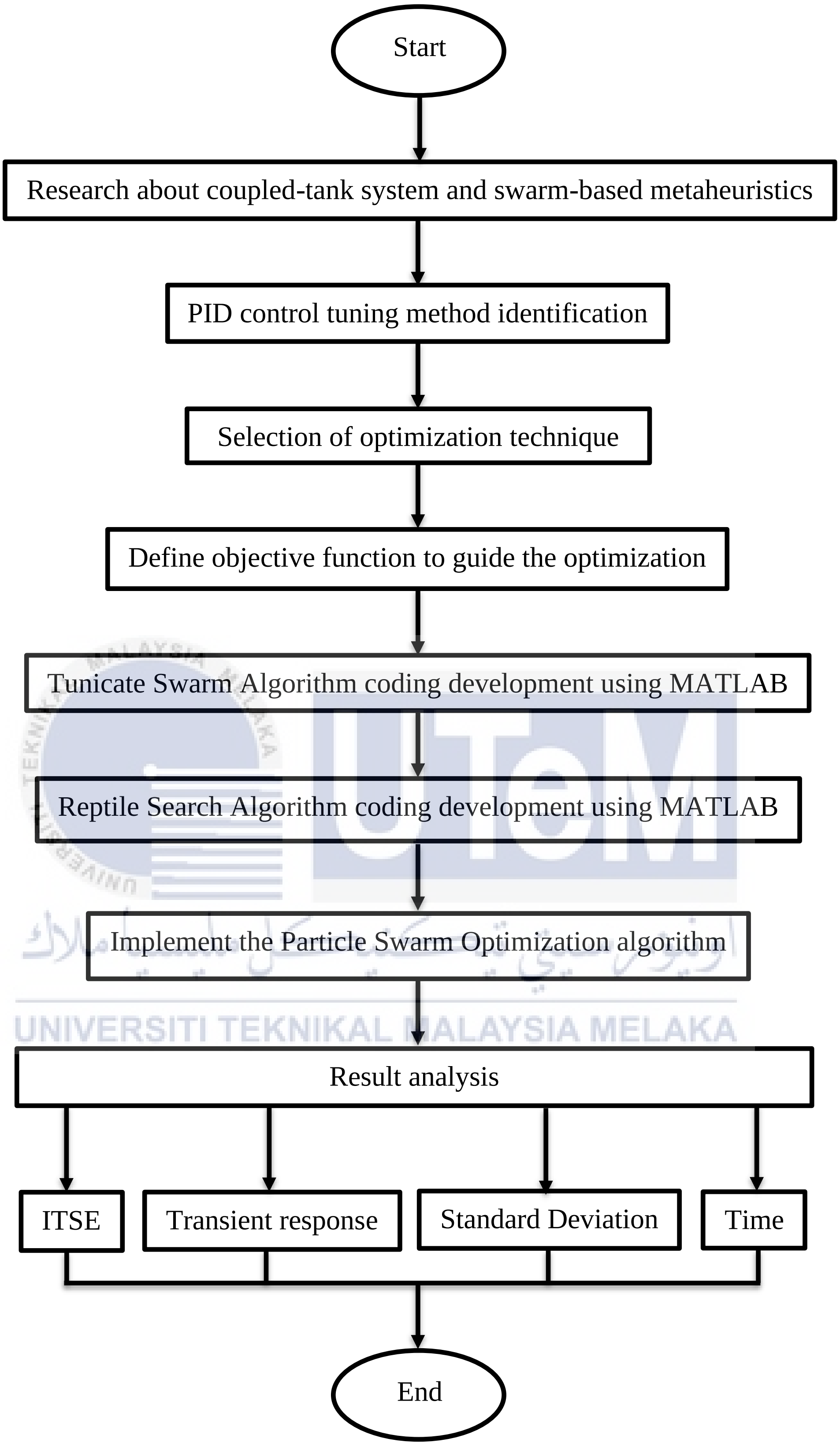

Figure 3.1 Methodology Flowchart.

3.3 Mathematical Model of Coupled Tank System [2]

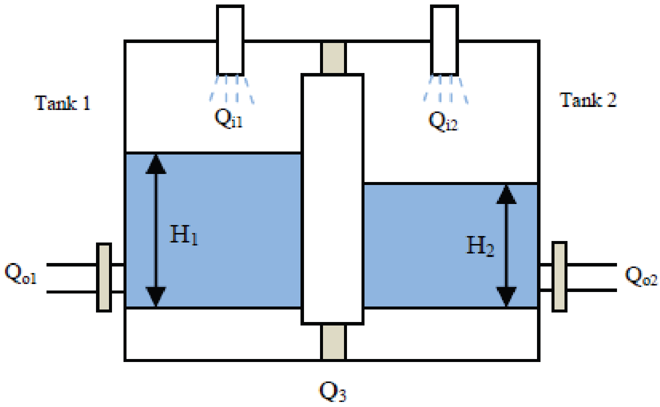

Figure 3.2 Coupled tank liquid level system.[5]

The fluid levels in Tanks 1 and 2 are denoted by H_1 and H_2. It is measured in relation to the corresponding outlet. The net flow of fluid into each tank is equal to the rate of change of volume of fluid in each tank when considering the basic mass balance. So here are the equations for Tanks 1 and 2:

$$A_1 \frac{dH_1}{dt} = Q_{i1} - Q_{o1} - Q_3 \tag{3.1}$$

$$A_2 \frac{dH_2}{dt} = Q_{i2} - Q_{o2} + Q_3 \tag{3.2}$$

Where :

H_1, H_2 = height of fluid in Tank 1 and 2 respectively

A_1, A_2 = cross-sectional area of Tank 1 and 2 respectively

Q_3 = flow rate of fluid between tanks

Qi_1, Qi_2 = pump flow rate into Tank 1 and 2 respectively

Qo_1, Qo_2 = flow rate of fluid out of Tank 1 and 2 respectively

One may represent any outlet drain as a straightforward orifice. The outlet flow in each tank is proportional to the square root of the water head in the tank, according to Bernoulli's equation for steady, non-viscous, incompressible flow. In a similar vein, the square root of the head difference determines the flow between the tanks. As a result:

$$Q_{o1} = \alpha_1 \sqrt{H_1} \tag{3.3}$$

$$Q_{o2} = \alpha_2\sqrt{H_2} \tag{3.4}$$

$$Q_3 = \alpha_3\sqrt{H_1 - H_2} \tag{3.5}$$

where α1, α2, and α3 are the cross-sectional area of each orifice, the gravitational constant, and the coefficients of discharge provide the proportionality constants. The following nonlinear state equations, which represent the system dynamics of the CTS apparatus, may be obtained by substituting (3.3), (3.4), and (3.5) into (3.1) and (3.2):

$$A_1\frac{dH_1}{dt} = Q_{i1} - \alpha_1\sqrt{H_1} - \alpha_3\sqrt{H_1 - H_2} \tag{3.6}$$

$$A_2\frac{dH_2}{dt} = Q_{i2} - \alpha_2\sqrt{H_2} + \alpha_3\sqrt{H_1 - H_2} \tag{3.7}$$

The manipulated variable in the second order configuration is q1, and the process variable is h2. It is assumed that q2 is zero. The second-order system's block diagram may be made simpler, as seen in Figure 3.3.

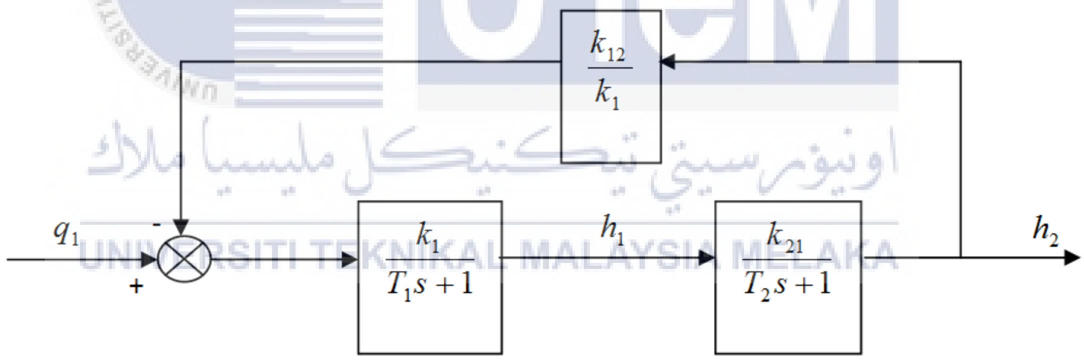

Figure 3.3 Block diagram of second order system

Thus, the nonlinear CTS can be obtained as:

$$\frac{h_2(s)}{q_1(s)} = \frac{k_1 k_2}{(T_1 s+1)(T_2 s+1)-k_{12}k_{21}}$$

$$\frac{h_2(s)}{q_1(s)} = \frac{k_1 k_2}{T_1 T_2 s^2+(T_1+T_2)s+(1-k_{12}k_{21})} \tag{3.8}$$

Where:

$$T_1 = \frac{A_1}{\left(\frac{\alpha_1}{2\sqrt{H_1}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.9}$$

$$T_2 = \frac{A_2}{\left(\frac{\alpha_2}{2\sqrt{H_2}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.10}$$

$$k_1 = \frac{1}{\left(\frac{\alpha_1}{2\sqrt{H_1}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.11}$$

$$k_2 = \frac{1}{\left(\frac{\alpha_1}{2\sqrt{H_2}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.12}$$

$$k_{12} = \frac{\frac{\alpha_3}{2\sqrt{H_1-H_2}}}{\left(\frac{\alpha_1}{2\sqrt{H_1}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.13}$$

$$k_{21} = \frac{\frac{\alpha_3}{2\sqrt{H_1-H_2}}}{\left(\frac{\alpha_2}{2\sqrt{H_2}}\right)+\left(\frac{\alpha_3}{2\sqrt{H_1-H_2}}\right)} \tag{3.14}$$

The values that were supplied from [9] can be substituted to get the plant's transfer function; these parameters are displayed in Table 3.1.

Table 3.1 Parameter of coupled-tank system

Parameters	**Value**	**Unit**
H_1	17	Cm
H_2	15	cm
α_1	10.78	$cm^{3/2}$/sec
α_2	11.03	$cm^{3/2}$/sec
α_3	11.03	$cm^{3/2}$/sec
A_1	32	cm^2
A_2	32	cm^2

Then, all the parameters in Table 3.1 have been inserted into (3.8). Thus, the actual transfer function of the plant with the completed value is:

$$G_p(s) = \frac{h_2(s)}{q_1(s)} = \frac{0.0361}{36.9406s^2+12.1565s+0.4514} \tag{3.15}$$

3.4 Optimization Technique

This study will implement the tunicate swarm algorithm (TSA) and reptile search algorithm (RSA) as optimization techniques. The system's transient response performance of the two algorithm is evaluated by comparing it to the PSO algorithm. The proposed method's advantages and disadvantages can be determine by comparing the three algorithms.

3.4.1 Tunicate Search Algorithm (TSA)

The TSA is a metaheuristic optimization method inspired by the life of tunicates, marine organisms that exhibit interesting collective behaviour. The flowchart of TSA is shown in Figure 3.4 below. The mathematical model system to simulate jet propulsion behaviour as follows with three conditions [49] :

1) First Condition : avoid the conflicts between search agent.

   A vector is used for updating the new position of tunicate, and mathematically as follows :

$$\vec{A} = \vec{G}/\vec{M} \tag{3.16}$$

$$\vec{G} = c_2 + c_3 - \vec{F} \tag{3.17}$$

$$\vec{F} = 2 \times c_1 \tag{3.18}$$

$$\vec{M} = \lfloor P_{min} + c_1 . P_{max} - P_{min} \rfloor \tag{3.19}$$

Where :

$\vec{G}$ is gravity force , $\vec{F}$ is the water flow advection, $\vec{M}$ is the social forces between search agents, P_{min} and P_{max} are the limitations of speeds to make social interaction, and (c_1, c_2, c_3) are random values between (0,1).

2) Second condition : movement towards the position of search agent.

All tunicates try to move towards the best agent position in this condition.

$$\vec{PD} = \left|\vec{FS} - rand.P_p^{\rightarrow}(x)\right| \tag{3.20}$$

Where

$\vec{PD}$ is the distance between search agent and foods, $\vec{FS}$ is the position of food source, $rand$ is a random value between (0,1) , and $P_p^{\rightarrow}(x)$ is the agent position.

3) Third condition : remains close to the best search agent.

The tunicates are move in the direction of the best agent, and mathematically as follows:

$$P_p^{\rightarrow}(x) = \begin{cases} F^{\rightarrow}S + A^{\rightarrow}.P^{\rightarrow}D, if\ rand\ \geq 0.5 \\ F^{\rightarrow}S - A^{\rightarrow}.P^{\rightarrow}D, if\ rand\ < 0.5 \end{cases} \tag{3.21}$$

Finally, the swarm behaviour is simulated by updating the agent position due to the best agent, and mathematically as:

$$P_p(x^{\rightarrow} + 1) = \frac{P_p^{\rightarrow}(x) + P_p(x^{\rightarrow} + 1)}{2 + c_1} \tag{3.22}$$

Table 3.2 Parameter initialization in TSA algorithm

Initialization
c_1, c_2, c_3 = values between (0,1)
P_{min} = 1
P_{max} = 4

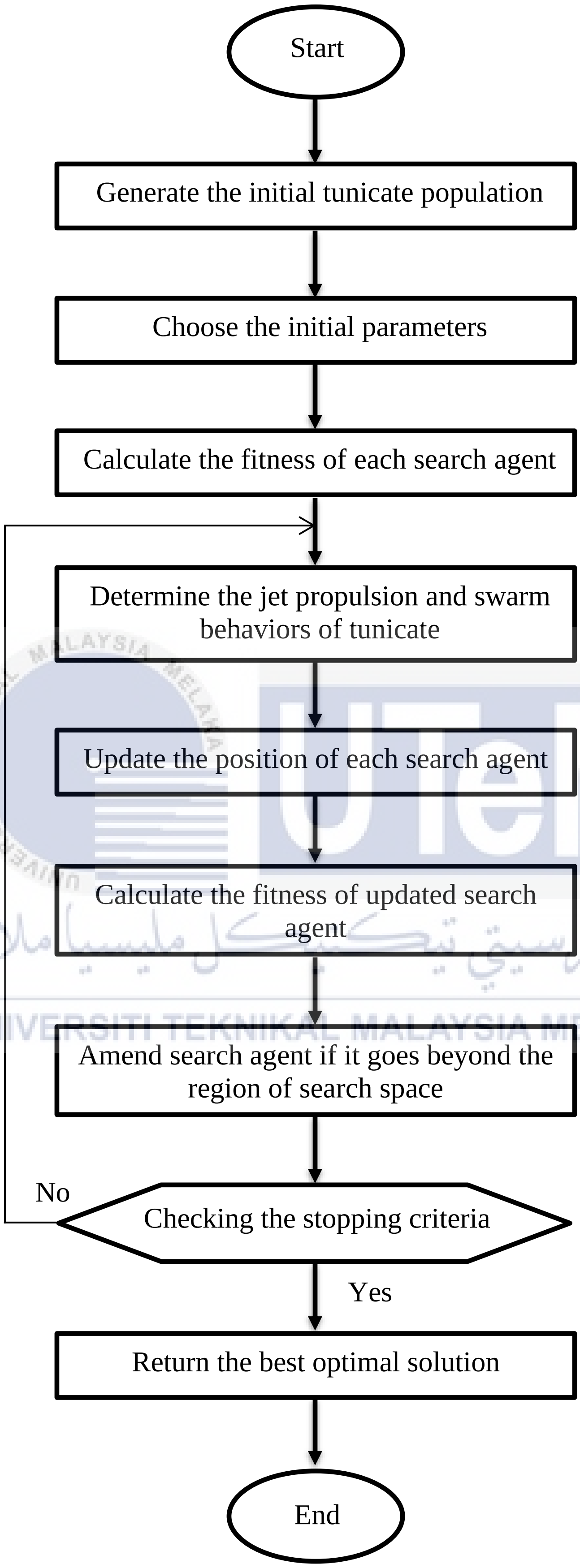

Figure 3.4 TSA Flowchart

1. Setting up the TSA population.
2. Choosing the maximum number of iterations and initial parameters.
3. Assessing each search agent’s fitness value and calculate the fitness.
4. After determining the fitness value, the best search agent in the given search space is scanned.
5. Using the equation (3.22), updating the location of each search agent.
6. In a given search space, adjusting the updated search agent if it exceeds the limit.
7. Working out the updated search agent's fitness value, and updating the optimal solution P_p if a better on exists.
8. Putting an end to the algorithm if the criteria for halting is met, otherwise, go through Steps 5–8 again.
9. Finally, the best optimum result can be obtained.

3.4.2 Reptile Search Algorithm (RSA)

RSA is a meta-heuristic algorithm modeled on the foraging behavior of crocodiles. Despite their typically slow movements, crocodiles are capable of rapid, aggressive attacks. As apex predators, crocodiles often hunt in groups. Their foraging behavior can be categorized into two main phases: the encircling stage, representing exploration, and the hunting stage, representing exploitation.[60]

Flowchart in Figure 3.5 will give a better understanding of RSA algorithm.

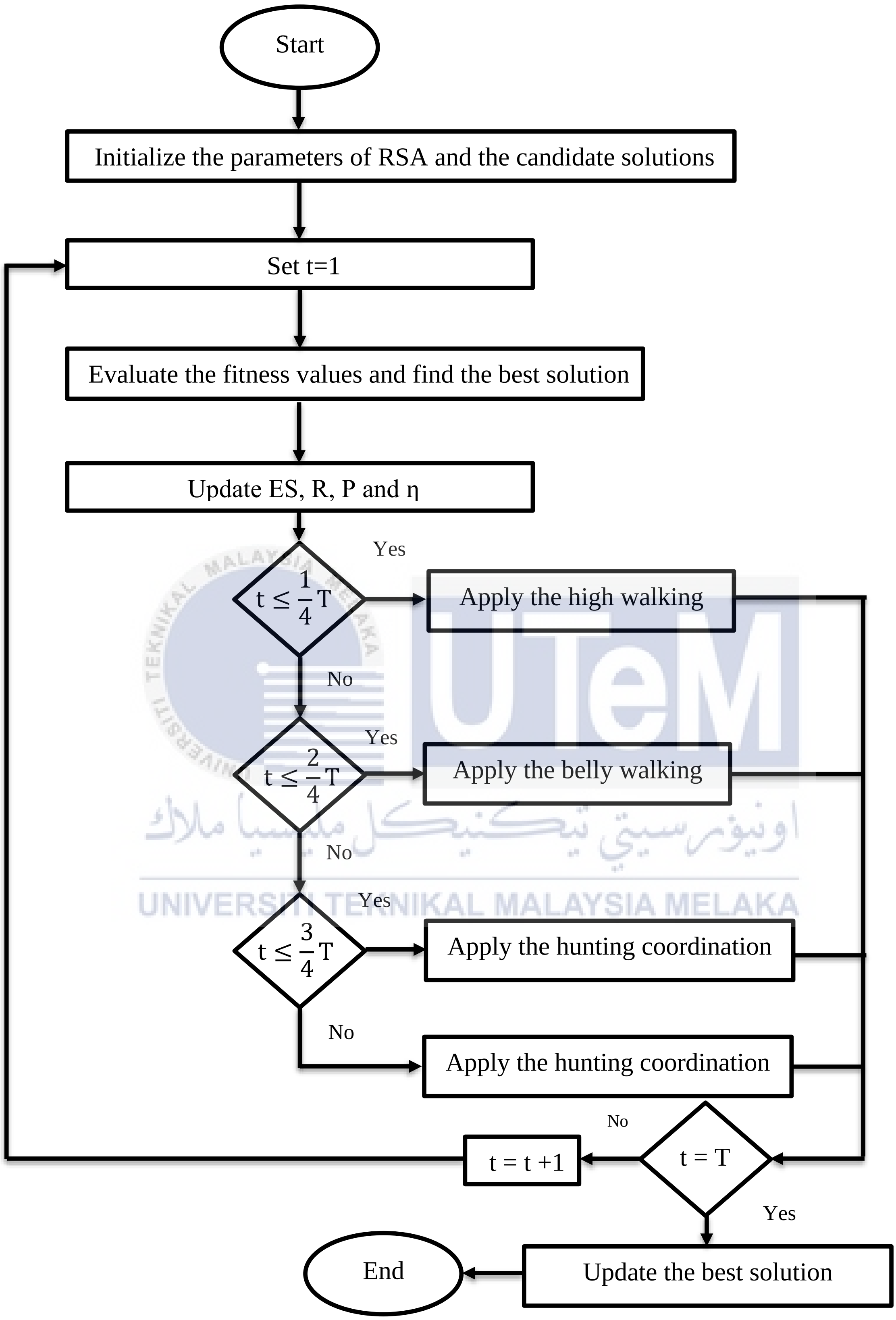

Figure 3.5 RSA flowchart

1. Set up the initial parameters and generate the candidate solutions for RSA.
2. Initialize the iteration counter *t* to 1.
3. Calculate the fitness value of all candidate solutions to determine their quality.
4. Update the parameters of ES, R, P and η.
5. Check if the current iteration $t \leq \frac{1}{4}$T . If yes, use the high walking strategy.
6. If no, check if the current iteration $t \leq \frac{2}{4}$T and apply the belly walking.
7. If the current iteration $t \leq \frac{3}{4}$T , apply the hunting coordination.
8. If the iteration is beyond $t \leq \frac{3}{4}$T, apply the hunting cooperation.
9. Check if the current iteration *t* = T.
10. If not, increment the iteration counter *t* by 1.
11. If the total number of iterations T is reached, update and finalize the best solution found.

In the context of the RSA, RSA will generate *N* candidate solutions, and the dimension size of each solution is *dim*. The *i*th solution is $(X_{(i,1)}, X_{(i,2)}, \ldots, X_{(i,j)}, \ldots, X_{(i,dim)})$. The initialization Formula of the *i*th solution in the *j*th dimension is as follows:

$$X_{ij} = rand \times (UB - LB) + LB, j = 1,2, \ldots, n \tag{3.23}$$

Where LB is the lower bound and UB is the upper bound.

3.4.2.1 Encircling phase (exploration)

Crocodiles will choose two different ways in the process of encircling prey: high walking and belly walking.

1. High walking

The calculation formula for high wallking is expressed in :

$$X_{(i,j)}(t+1) = Best_j(t) \times \eta_{(i,j)}(t+1) \times \beta - R_{(i,j)}(t+1) \times rand \; t \leq \frac{T}{4} \tag{3.24}$$

Where :

$X_{(i,j)}(t+1)$ = the updated position of the *i*th individual in the *j*th dimension.

$Best_j(t)$ = the best-known position in the *j*th dimension up to iteration.

$\eta_{(i,j)}(t+1)$ = the hunting operator for the *i*th individual in the *j*th dimension, determined by Equation (3.25).

β = sensitivity control parameter for search capability, set at 0.005.

$R_{(i,j)}(t+1)$ = adjusts the search area size and is computed using Equation (3.26).

t = the current iteration number, and T is the total number of iterations.

$$\begin{cases} \eta_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t+1) \\ P_{(i,j)}(t+1) = \alpha + \frac{X_{(i,j)}(t) - M(X_{(j)})}{Best_j(t) \times (UB_{(j)} - LB_{(j)} + \varepsilon} \\ M(X_{(i)}) = \frac{1}{dim} \sum_{j=1}^{dim} X_{(i,j)}(t) \end{cases} \tag{3.25}$$

Where :

$P_{(i,j)}(t+1)$ = the percentage difference between the optimal and current individuals in the *j*th dimension.

α = regulates search accuracy and is fixed at 0.1.

$X_{(i,j)}(t)$ = the position of the *i*th individual in the *j*th dimension prior to updating.

$M(X_{(i)})$ = the average value of the *i*th individual's positions across all dimensions.

ε = small constant added to prevent division by zero.

$$R_{(i,j)}(t+1) = \frac{Best_j(t) - X_{(r1,j)}(t)}{Best_j(t) + \varepsilon} \tag{3.26}$$

where $X_{(r1,j)}(t)$ represents the random individual's position.

2. Belly walking.

The formula for calculating belly walking is presented in Equation (3.27):

$$X_{(i,j)}(t+1) = Best_j(t) \times X_{(r2,j)}(t) \times ES \times rand \; t > \frac{T}{4} \;\&\; t \le \frac{T}{2} \tag{3.27}$$

Where :

$X_{(r2,j)}(t)$ = the position of a randomly selected individual.

ES = the evolutionary direction and is a randomly assigned decreasing value ranging between 2 and -2.

The value of ES is determined as follows:

$$ES = 2 \times RAND \times \left(1 - \frac{t}{T}\right), RAND \in [1,1] \tag{3.28}$$

3.4.2.2 Hunting Phase (exploitation)

In line with the hunting behavior of crocodiles, the hunting stage employs two strategies which are hunting coordination and hunting cooperation. Unlike the encircling stage, where the crocodiles position themselves around the prey, in the hunting stage, they remain close to the prey to execute the capture. The formula for hunting coordination is given as follows:

$$X_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t+1) \times rand\ t \leq 3\frac{T}{4}\ \&\ t > \frac{T}{2} \tag{3.29}$$

The formula of hunting cooperation is :

$$X_{(i,j)}(t+1) = Best_j(t) - \eta_{(i,j)}(t+1) \times \varepsilon - R_{(i,j)}(t+1) \times rand\ t > 3\frac{T}{4} \tag{3.30}$$

Table 3.3 Parameter initialization in RSA algorithm

Initialization
ES = randomly decreasing values between 2 and −2
α = 0.1
β = 0.005

3.4.3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) working principle is based on social sharing of a swarm such as fish schooling or bird flocking. During search for food, the infromation sharing is happening between them. In PSO, each single solution is a bird in the search space which referred as a particle. The particles have the memory of their own best position and knowledge of global best. The swarm particles communicate through the best position and velocity. Flowchart in Figure 3.6 will give a better understanding of PSO algorithm.

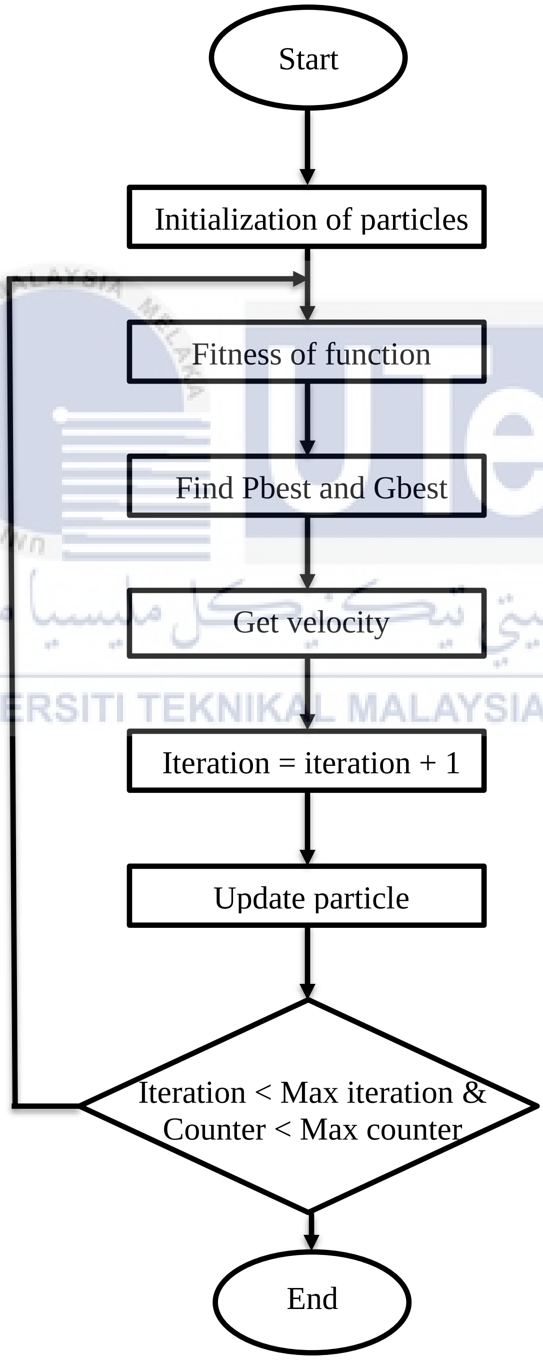

Figure 3.6 PSO Flowchart

1. Initialize particles with random position and velocity vectors. In this step, a swarm particles is created, each representing a potential solution in the search space. The particles are assigned random positions and initial velocities. These initial positions serve as starting points for exploration.
2. For every particle in the swarm, calculate its fitness based on a predefined fitness function. The fitness function measures how well the particle's position performs in solving the optimization problem. This evaluation guides the particles toward better solutions.
3. If fitness (p) is better than the best fitness (Pbest), update Pbest. Compare the current fitness of particle (p) with its personal best fitness (Pbest). If the current fitness is better (i.e., lower or higher, depending on the problem type), update Pbest to the current fitness value. This step ensures that each particle remembers its best position found so far.
4. Update particles' velocity and position: The velocity and position of each particle are adjusted based on its own experience and the collective behaviour of the swarm. The velocity update formula involves inertia weight, cognitive acceleration, and social acceleration. The new velocity determines how the particle moves in the search space. The updated position is calculated by adding the new velocity to the current position.

$$v_i^k = wv_i^k + c_1r_1\left(pbest_i^k - x_i^k\right) + c_2r_2(gbest^k - x_i^k) \tag{3.31}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{3.32}$$

Where

v_i^k: velocity of the ith particle at the kth iteration

x_i^k: current position of the ith particle at the kth iteration

c_1, c_2 : positive constant

r_1, r_2 : random variables with uniform distribution between 0 and 1

w : inertia weight

5. Loop until all particles have exhausted iterations. The PSO process iterates through these steps until a stopping criterion is met (e.g., a maximum number of iterations or a desired fitness level). During each iteration, particles explore the search space, updating their positions and velocities.
6. Set the best of Pbest as Gbest. The global best position (Gbest) represents the overall best solution found by any particle in the entire swarm. It is updated whenever a particle discovers a better solution than the current Gbest. The final Gbest position provides the optimal solution to the optimization problem.

Table 3.4 Parameter initialization in PSO algorithm

Velocity initialization	
c_1, c_2 = 2	Maximum weight = 1
Maximum velocity = 999	Minimum weight = 0.99

3.5 Performance Evaluation

3.5.1 Performance Index

The evaluation of a system's performance relies on a performance index, which can take various forms, such as frequency domain specifications, time-integral specifications, or time-domain specifications. In this study, there is no specific criterion for the performance index as long as it yields an optimal value. Generally, a smaller error of performance index indicates better system performance.

For this particular study, the chosen performance index is the Integral Time Square Error (ITSE). ITSE, also known as time-integral performance, emphasizes both settling time and error value. It quantifies the area between the system's output and the desired output, providing a measure of dynamic performance.

$$ITSE = \int_0^t (e_i(t))^2 t dt \quad (3.33)$$

$$e(t) = y(t) - r(t) \quad (3.34)$$

Where

$e(t)$: error in (3.34)

$y(t)$: output of system

$r(t)$: input of system

3.5.2 Transient Response

Evaluating the transient response of coupled tank systems involves key performance parameters such as rise time, settling time, peak time and overshoot.

Rise time measures how quickly the system reaches a certain percentage (typically 90% or 95%) of the final value after a step input, with a smaller rise time indicating a faster response. The settling time represents the duration required for the output to remain within a specified tolerance band around the final value, where a shorter settling time indicates better performance. Peak time is the time taken for the system output to reach its first peak, or maximum overshoot, during the transient phase. Overshoot refers to the maximum deviation of the system output from the desired set point during this phase and occurs when the system exceeds the desired value before settling. Minimizing overshoot is important to maintain stable and accurate control. Collectively, these metrics provide a comprehensive understanding of the dynamics and stability of system responses.

3.5.3 Time

Measurement of execution time involves recording the duration required for each run to calculate the Integral Time Squared Error (ITSE) value. This measurement represents the time taken by the system or algorithm to perform the necessary calculations. By analyzing the execution time, we gain insight into the algorithm's efficiency and stability, evaluating both consistency and variability in its performance. This analysis helps in understanding how reliably the algorithm can handle calculations in different conditions, providing a comprehensive view of its operational characteristics.

3.6 Parameter determination

To ensure a fair comparison of TSA, RSA and PSO for PID controller tuning, consistent parameters are maintained during code execution. The number of particles, or swarm size is kept the same for all three algorithms to provide comparable exploration and exploitation capabilities. The maximum iteration is also kept constant across TSA, RSA and PSO, allowing a fair assessment of their convergence behaviour. Additionally, parameter bounds (such as PID gain) are the same for all algorithms, preventing bias due to different search spaces. Adhering to these consistent settings allows objective evaluation of each algorithm's performance in optimizing PID controller parameters.

Two key factors when optimizing PID controllers are swarm size and maximum iterations. Using a swarm size of 50 is effective because it balances the need to explore different solutions while keeping computations manageable. If the swarm is too small, the search might end too soon and miss good solutions. If it's too large, it can slow down the process without much benefit. Setting the maximum iterations to 100 gives the algorithms enough time to find good solutions without overdoing it. This limit helps the algorithms converge well without wasting time or overfitting.

To determine the optimal upper bounds for PID controller optimization, 1000 and 10,000 were tested by calculating the standard deviation of performance metrics, such as ITSE, across multiple runs. A smaller standard deviation indicates more consistent results. The upper bound of 10,000 produced greater stability and reliability compared to 1000, making 10,000 the preferred choice. Additionally, a lower bound of 0.1 is enforced for the PID controller parameters to ensure that the proportional, integral, and derivative gains are positive and nonzero, avoiding physically unrealistic PID values.

Table 3.5 Parameter initialization

Initialization	
No of swarm = 50	Maximum iteration = 100
Lower bound = 0.1	Upper bound = 10 000

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The results of the study will be presented in this chapter. It contains the results of PID optimization using Tunicate Swarm Algorithm (TSA), Reptile Search Algorithm (RSA), and Particle Swarm Optimization (PSO) techniques. This chapter will evaluate and discuss system performance using the PID tuning approach.

4.2 PID Tuning using Optimization Technique

Optimization techniques are used to find the ideal PID parameters and to reduce the Integral of Time Square Error (ITSE) value when the process is in a steady state. The three tuning parameters are derivative gain (Kd), integral gain (Ki), and proportional gain (Kp). A performance index that yields a lower integral criteria value will yield better closed-loop performance in terms of settling time, rising time, and steady-state error.

The controller utilizing an optimization technique is implemented in the CTS to achieve the desired output based on the system's step input. Two novel optimization techniques, TSA and RSA, will be evaluated and compared against the well-known Particle Swarm Optimization (PSO) algorithm. The comparison focuses on the transient response of the system and the fitness value. To ensure a fair evaluation, certain parameters are held constant; specifically, the maximum number of iterations and the range of PID parameters are set identically for all three algorithms.

4.2.1 PID with Tunicate Swarm Algorithm (PIDTSA)

This research focuses on simulations to identify the lowest fitness value using three different algorithms which are PIDTSA, PIDRSA and PIDPSO. Lower fitness values are associated with enhanced output, facilitating the determination of the best

PID parameters. The selected minimum fitness value leads to the acquisition of the optimal PID setting. The simulation process was repeated 20 times, each consisting of 100 iterations.

During 20 executions, PIDTSA recorded its lowest ITSE fitness value on the 13th run, which is 0.3511, while the highest was 0.3569 on the 16th run. The ITSE fitness value measures the error in the output signal as the integral of the time-weighted absolute error. As illustrated in Figure 4.1, the ITSE value starts at 0.4566 in the first iteration and decreases to 0.3854 by the second iteration. It further decreased to 0.3522 by the eighth iteration and remained constant until the 25th iteration. The value is then slightly adjusted to 0.352 and maintained until the 33rd iteration. Between iterations 36 and 59, it stabilizes at 0.3513, and from iteration 65 onwards, it remains at 0.3512, indicating that the optimization has reached convergence. Finally, the fitness value decreased slightly to 0.3511 at the 100th iteration. This final fitness value from the 13th run, which is 0.3511, was chosen to determine the optimal value of the PID parameter.

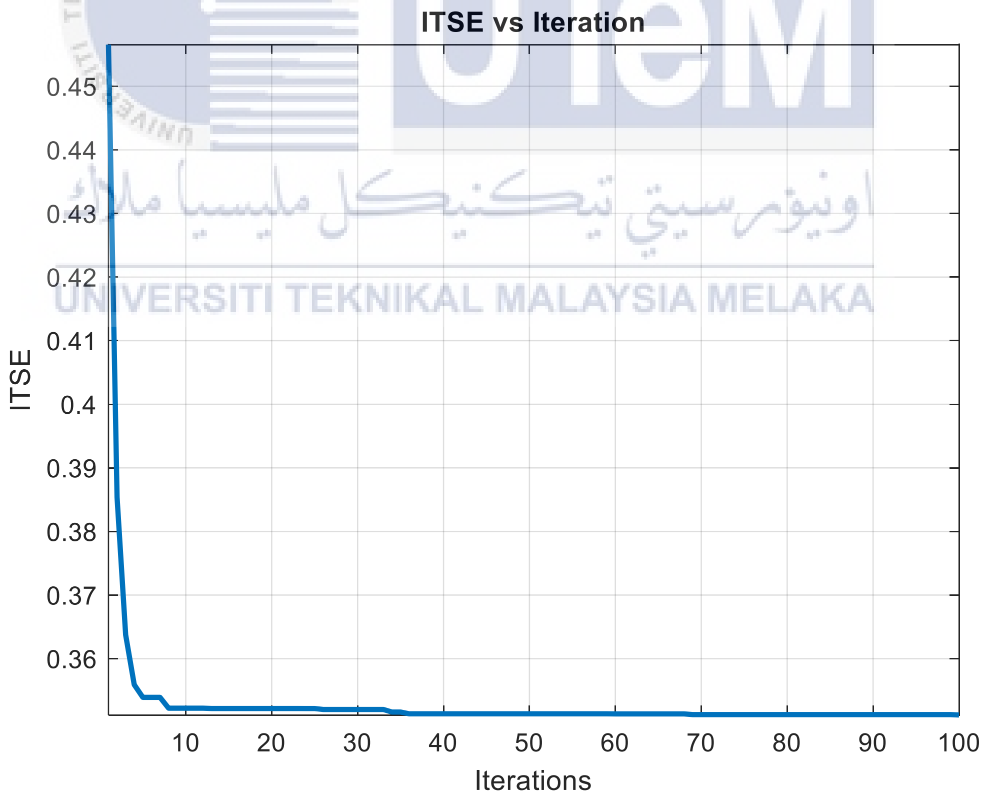

Figure 4.1 Graph fitness function versus iteration for TSA

Therefore, the optimal PID values are proportional gain, Kp of 10000, integral gain, Ki of 0.1091, and derivative gain Kd of 9597.7062. With these PID parameters, the CTS achieves the transient response shown in Figure 4.2, characterized by an overshoot of 4.82% and a steady-state achievement time of 1.45 seconds.

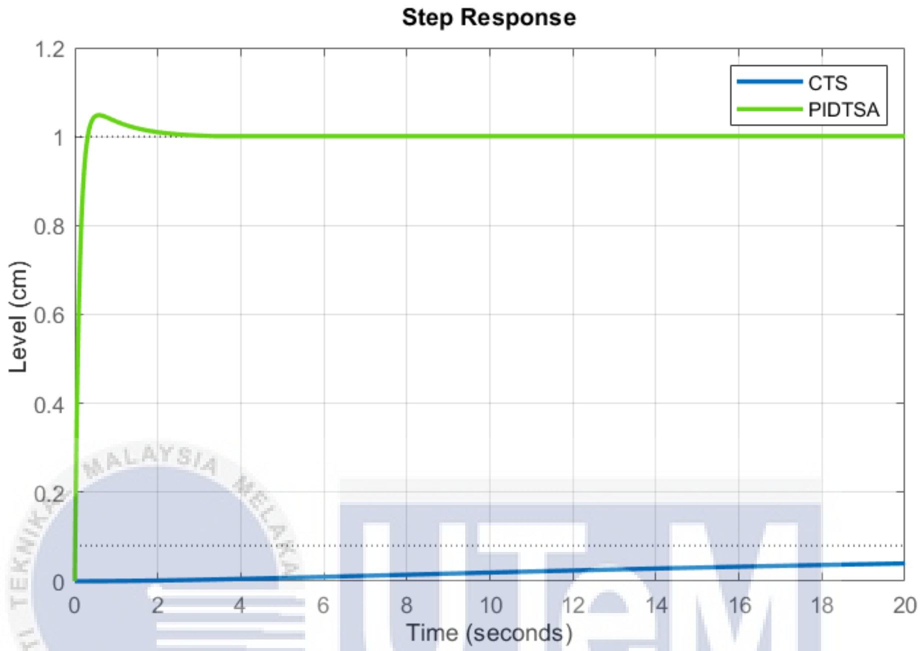

Figure 4.2 Step Response of Closed Loop System using TSA

4.2.2 PID with Reptile Search Algorithm (PIDRSA)

Over 20 executions, the PIDRSA algorithm reached its lowest ITSE fitness value on the 17th run, clocking in at 0.3539, while the highest reached 0.6120 on the 20th run. Figure 4.3 shows that the ITSE starts at 0.6030 in the initial iteration and remains stable for twelve iterations, then settles at 0.5837 for the next twenty-four iterations, before finally falling to 0.3539 from the seventy-fourth iteration onwards. This final fitness value of 0.3539, recorded on the 17th run, was chosen to ensure the optimal PID parameter value.

As a result, the ideal PID parameters were determined as proportional gain, Kp of 10000, integral gain, Ki of 20.4944, and derivative gain Kd of 9551.7129. Using this value, the CTS shows a transient response as illustrated in Figure 4.4, with an overshoot of 4.88% and a steady state reached in 1.46 seconds.

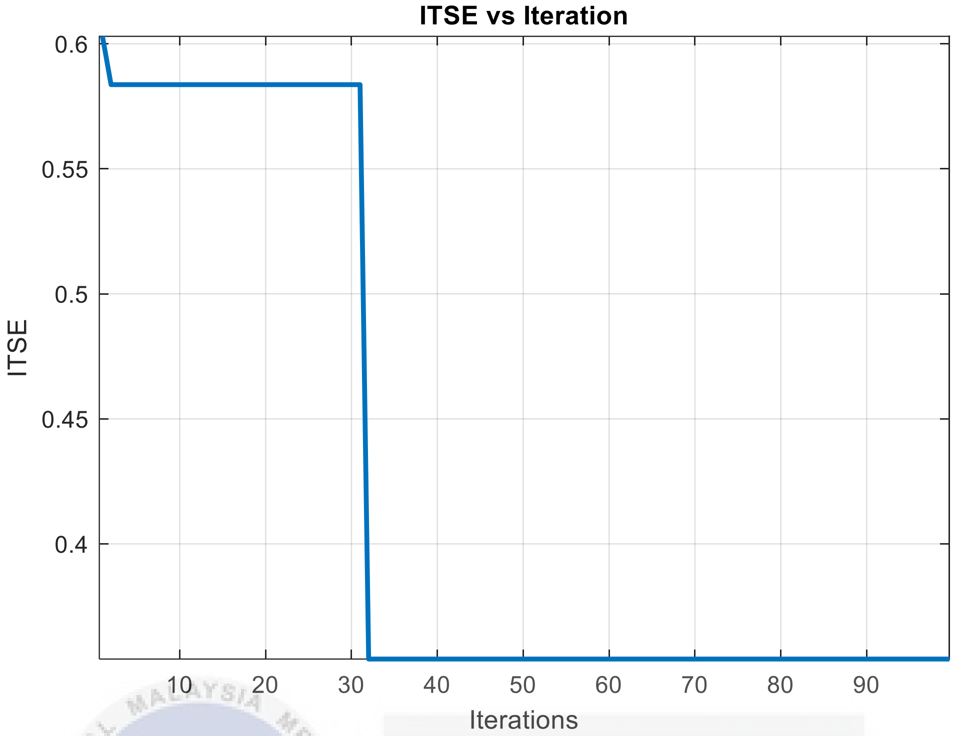

Figure 4.3 Graph fitness function versus iteration for RSA

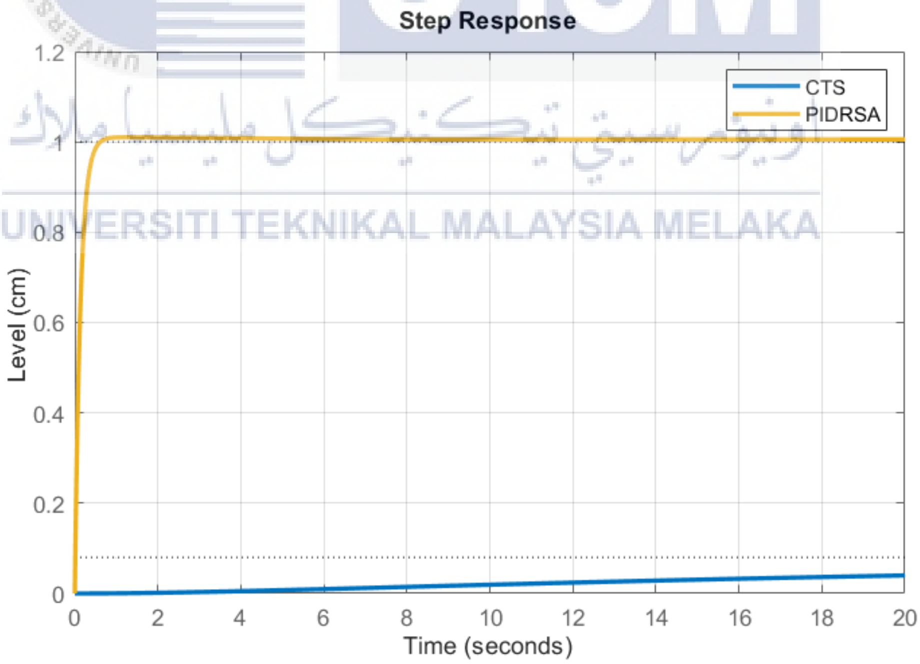

Figure 4.4 Step Response of Closed Loop System using RSA

4.2.3 PID with Particle Swarm Optimization (PIDPSO)

The simulations were run 20 times with 100 iterations each, reflecting the approach taken for PIDTSA and PIDRSA. The ITSE value remained constant at 0.3512 throughout the 20 runs, showing consistent performance in this experiment. As shown in Figure 4.5, the ITSE value drops dramatically from 0.5232 in the initial iteration to about 0.3560 by the ninth iteration. The value then stabilizes around 0.3518 by the tenth iteration and finally settles at 0.3512 from the twenty-ninth iteration onwards, indicating a successful convergence of the optimization process. These fitness values are used to determine the optimal PID parameters, which are proportional gain Kp of 10000, integral gain Ki of 0.1, and derivative gain Kd of 9598.6959. Using these parameters, the CTS displays a transient response as illustrated in Figure 4.6, with an overshoot of 4.82% and a time to reach steady state of 1.45 seconds.

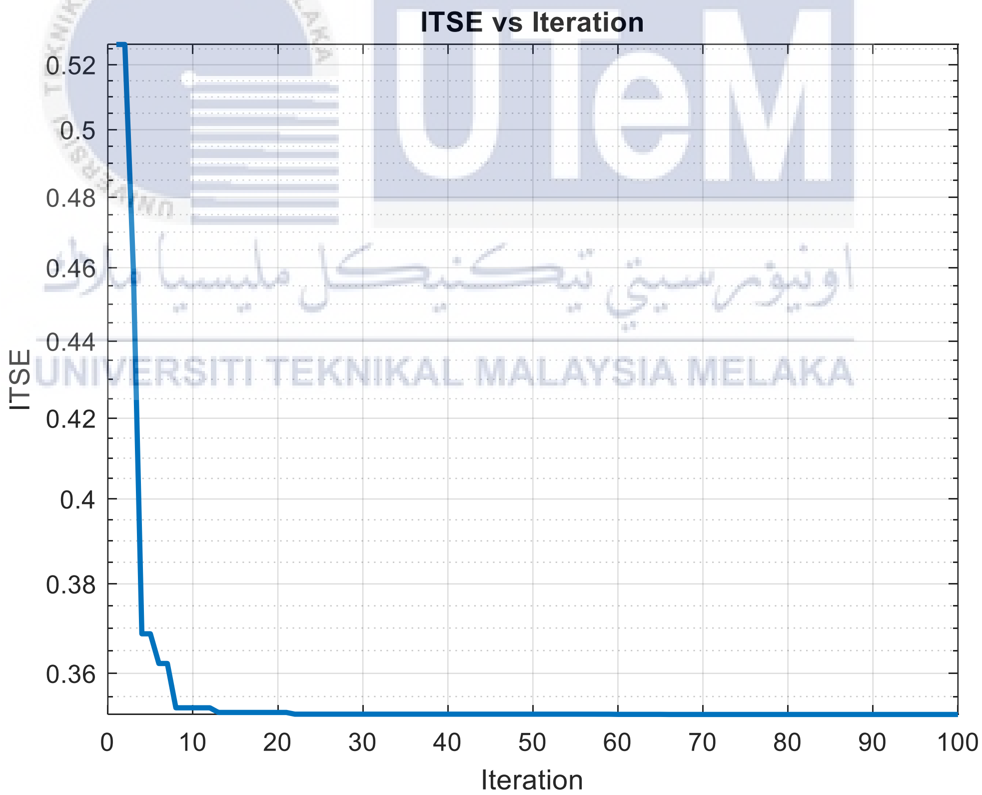

Figure 4.5 Graph fitness function versus iteration for PSO

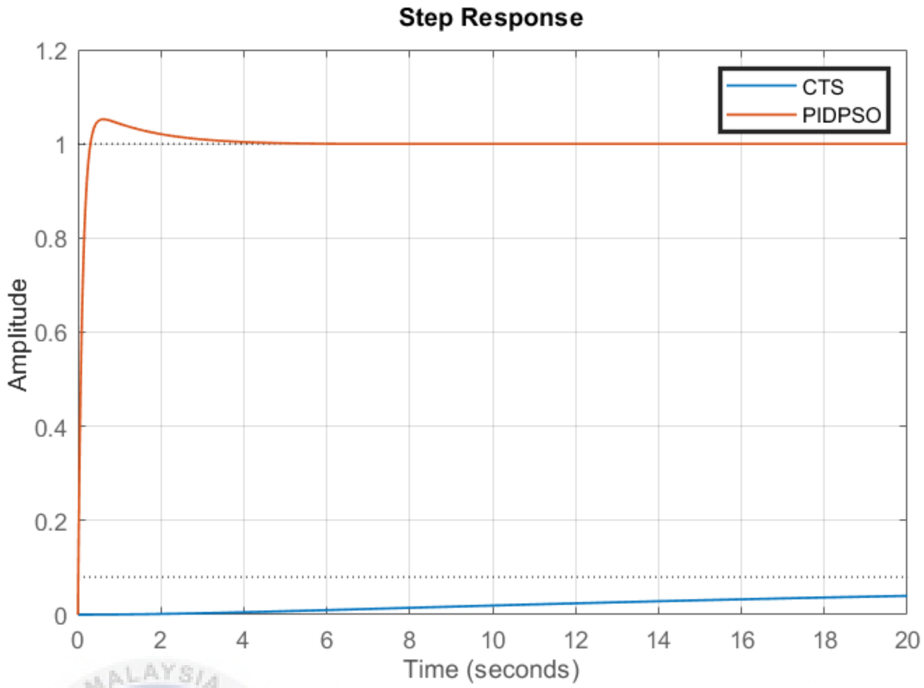

Figure 4.6 Step Response of Closed Loop System using PSO

4.3 Performance Evaluation Between PIDTSA, PIDRSA and PIDPSO.

The PID controller in the CTS is set to the ideal value of the PID parameter in Table 4.1 in order to enhance system performance. Figure 4.7 shows the step response for the three optimization method after tuned using the optimal PID parameter obtained.

Table 4.1 PID parameter value for CTS system

Parameter / Method	PIDTSA	PIDRSA	PIDPSO
K_P	10000	10000	10000
K_I	0.1091	20.4944	0.1
K_D	9597.7062	9551.7129	9598.6959

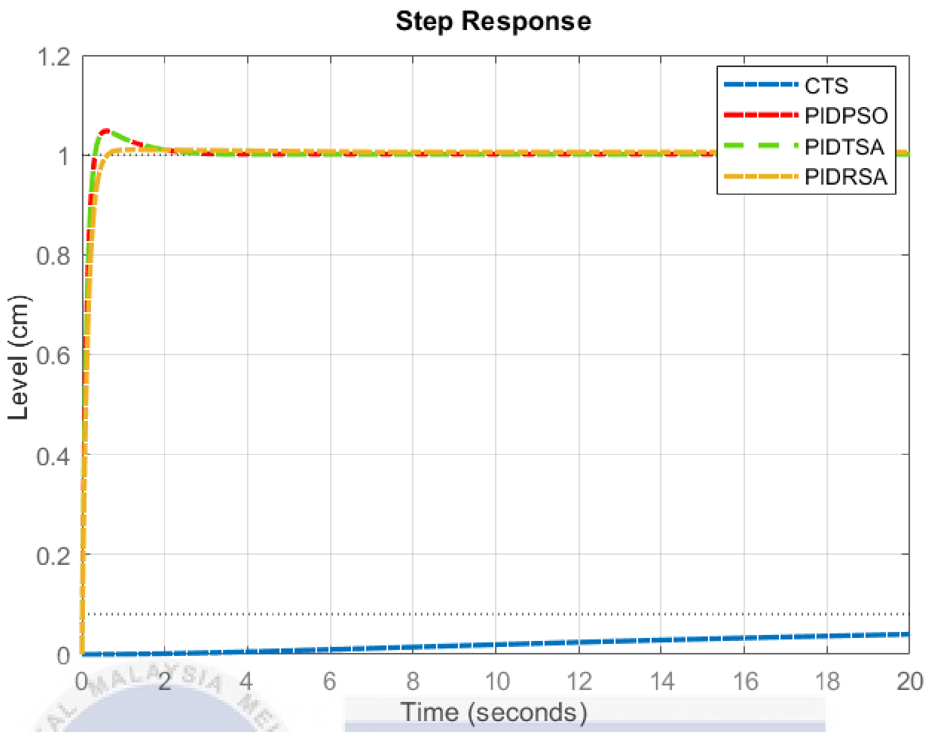

Figure 4.7 Step Response of Closed Loop System using PID optimization (PIDTSA, PIDRSA, PIDPSO)

Table 4.2 Analysis of Performance for PIDTSA, PIDRSA and PIDPSO

		PIDTSA	**PIDRSA**	**PIDPSO**
ITSE		**0.3511**	0.3539	0.3512
Transient Response	**Overshoot, OS**	**4.82**	4.88	**4.82**
	Peak Time, Tp	**0.58**	**0.58**	**0.58**
	Settling Time, Ts	**1.45**	1.46	**1.45**
	Rise Time,Tr	**0.20**	**0.20**	**0.20**
	Steady State Error, ess	0	0	0
Standard Deviation		0.0016	0.0796	**5e-6**
Time		45.26	47.52	**38.44**

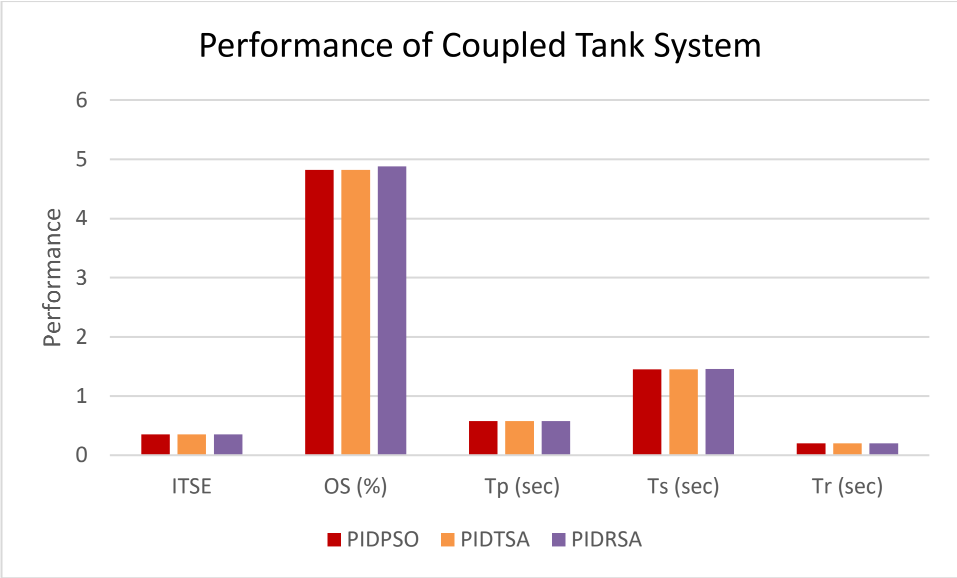

Figure 4.8 Performance transient response of coupled tank system based on PIDTSA, PIDRSA and PIDPSO in terms of bar graph

By referring to Table 4.2, the ITSE values for PIDTSA, PIDRSA, and PIDPSO methods are closely matched, reflecting their effectiveness in minimizing both the magnitude and duration of the error. PIDTSA shows a slight edge with an ITSE value of 0.3511, closely followed by PIDPSO at 0.3512, both indicating efficient performance in system optimization. PIDRSA, with an ITSE value of 0.3539, is marginally less efficient but still within a competitive range, suggesting all three methods are robust options for PID control parameter tuning.

In comparing the transient response of PIDTSA, PIDRSA, and PIDPSO methods in Table 4.2 and Figure 4.8, all exhibit a rapid rise time of 0.20 seconds indicating efficient system response to changes. PIDTSA and PIDPSO show identical overshoots at 4.82%, with PIDRSA slightly higher at 4.88%, while peak times are consistent at 0.58 seconds across all methods. Settling times are nearly identical, with PIDTSA and PIDPSO at 1.45 seconds and PIDRSA marginally longer at 1.46 seconds, demonstrating the close performance of these tuning strategies.

When examining the standard deviation for the ITSE across 20 executions, the PIDPSO method stands out with an exceptionally low standard deviation of just 5e-6, indicating highly consistent performance. The PIDTSA method also shows a low standard deviation at 0.0016, suggesting a reliable and stable tuning process. In contrast, the PIDRSA method has a higher standard deviation of 0.0796, which points to greater variability in its performance outcomes.

In the comparison of performance between PIDTSA, PIDRSA, and PIDPSO methods, it was observed that PIDPSO achieved its ITSE value in the shortest amount of time, taking only 38.44s. On the other hand, PIDRSA took the longest with 47.52s, while PIDTSA required 45.26s to reach its ITSE value. These times reflect the efficiency of each algorithm in minimizing the error over time during the tuning process of the PID controller for a coupled tank system.

4.4 Chapter Summary

From Table 4.3 , PIDTSA offers the best ITSE, with advantages such as low overshoot, low settling time, moderate standard deviation, and a moderate time to achieve the desired ITSE. In comparison, PIDRSA performs less effectively, having the highest overshoot, settling time, standard deviation, and the longest time to achieve the desired ITSE. Conversely, PIDPSO achieves a good ITSE, with low overshoot, low settling time, the smallest standard deviation, and the shortest time to reach the desired ITSE.

Table 4.3 Performance Evaluation of PID Contollers for CTS

<table>
<tr><th colspan="2">Metric</th><th>PIDTSA</th><th>PIDRSA</th><th>PIDPSO</th></tr>
<tr><td colspan="2">ITSE</td><td>Best ITSE</td><td>Less Good ITSE</td><td>Good ITSE</td></tr>
<tr><td rowspan="6">Transient Response</td><td>Overshoot (OS)</td><td>Low (same as PIDPSO)</td><td>Higher than PIDTSA and PIDPSO</td><td>Low (same as PIDTSA)</td></tr>
<tr><td>Peak Time (Tp)</td><td>Same</td><td>Same</td><td>Same</td></tr>
<tr><td>Settling Time (Ts)</td><td>Low (same as PIDPSO)</td><td>Higher than PIDTSA and PIDPSO</td><td>Low (same as PIDTSA)</td></tr>
<tr><td>Rise Time (Tr)</td><td>Same</td><td>Same</td><td>Same</td></tr>
<tr><td>Steady-State Error</td><td>Zero</td><td>Zero</td><td>Zero</td></tr>
<tr><td colspan="2">Standard Deviation</td><td>Moderate</td><td>Highest</td><td>Smallest</td></tr>
<tr><td colspan="2">Time</td><td>Moderate</td><td>Slowest</td><td>Fastest</td></tr>
</table>

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The PIDPSO method appears to be the most effective and consistent for PID control parameter tuning in a coupled tank system. It has an ITSE value comparable to that of PIDTSA, indicating its effectiveness in minimizing error magnitude and duration. Additionally, it boasts the lowest standard deviation, suggesting high consistency across multiple runs. Moreover, it achieves the optimal ITSE value in the shortest time, highlighting its efficiency. While PIDTSA and PIDRSA are competitive and robust options with similar transient response characteristics to PIDPSO, the slightly higher ITSE value and longer execution time for PIDRSA, as well as the higher standard deviation compared to PIDPSO for PIDTSA, make them marginally less optimal than PIDPSO..In conclusion, while all methods are closely matched and robust, PIDPSO stands out as the best method among the three due to its superior consistency and efficiency.

5.2 Recommendation

For the PIDPSO controller in coupled tank systems, it is recommended to study its robustness under varying system conditions, such as parameter uncertainties, disturbances, and noise. Additionally, developing strategies to handle these uncertainties, including robust control techniques and adaptive tuning methods, will be essential to enhance the controller's reliability and effectiveness in diverse and unpredictable environments.

REFERENCES

[1] A. Muntaser and N. Buaossa, “Coupled Tank Non-linear System; Modeling and Level Control using PID and Fuzzy Logic Techniques,” 2021.

[2] H. I. Jaafar, S. Yuslinda, M. Shahrieel, and M. Aras, “Development of PID Controller for Controlling Desired Level of Coupled Tank System,” *Int. J. Innov. Technol. Explor. Eng.*, no. 3, pp. 2278–3075, 2014, [Online]. Available: https://www.researchgate.net/publication/263047616

[3] S. Chauhan, B. Singh, and M. Singh, “Modified ant colony optimization based PID controller design for coupled tank system,” *Eng. Res. Express*, vol. 3, no. 4, 2021, doi: 10.1088/2631-8695/ac2bf3.

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

[5] S. Y. Sim, S. L. Kek, and K. G. Tay, “Optimal control of a coupled tanks system with model-reality differences,” in *AIP Conference Proceedings*, 2017. doi: 10.1063/1.4996669.

[6] H. Om Bansal, H. O. Bansal, R. Sharma, and P. R. Shreeraman, “PID Controller Tuning Techniques: A Review,” *J. Control Eng. Technol.*, vol. 2, no. 1, pp. 481–485, 2017, [Online]. Available: www.vkingpub.com

[7] 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, pp. 1–29, 2022, doi: 10.1016/j.heliyon.2022.e09399.

[8] S. Y. S. Hussien, H. I. Jaafar, N. A. Selamat, F. S. Daud, and A. F. Z. Abidin, “PID Control Tuning VIA Particle Swarm Optimization for Coupled Tank System,” no. 2, pp. 202–206, 2014.

[9] N. Katal, P. Kumar, and S. Narayan, “Optimal PID controller for coupled-tank liquid-level control system using bat algorithm,” in *2014 International Conference on Power, Control and Embedded Systems, ICPCES 2014*, 2014, pp. 2–5. doi: 10.1109/ICPCES.2014.7062818.

[10] N. A. Selamat, F. S. Daud, H. I. Jaafar, and N. H. Shamsudin, “Comparison of LQR and PID controller tuning using PSO for Coupled Tank System,” in

Proceedings - 2015 IEEE 11th International Colloquium on Signal Processing and Its Applications, CSPA 2015, 2015, pp. 46–51. doi: 10.1109/CSPA.2015.7225616.

[11] M. M. Puralachetty and V. K. Pamula, “Differential evolution and particle swarm optimization algorithms with two stage initialization for PID controller tuning in coupled tank liquid level system,” in *ICARM 2016 - 2016 International Conference on Advanced Robotics and Mechatronics*, 2016, pp. 507–511. doi: 10.1109/ICARM.2016.7606972.

[12] M. Dehghani, E. Trojovská, and P. Trojovský, “A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–21, 2022, doi: 10.1038/s41598-022-14225-7.

[13] A. Zeb, F. Din, M. Fayaz, G. Mehmood, and K. Z. Zamli, “A Systematic Literature Review on Robust Swarm Intelligence Algorithms in Search-Based Software Engineering,” *Complexity*, vol. 2023, no. 1, 2023, doi: 10.1155/2023/4577581.

[14] P. H. Namin and M. A. Tinati, “Node localization using Particle Swarm Optimization,” in *Proceedings of the 2011 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2011*, 2011, pp. 288–293. doi: 10.1109/ISSNIP.2011.6146558.

[15] O. Chao and L. Weixing, “Comparison between PSO and GA for parameters optimization of PID controller,” in *2006 IEEE International Conference on Mechatronics and Automation, ICMA 2006*, 2006, pp. 2471–2475. doi: 10.1109/ICMA.2006.257739.

[16] S. Gupta, P. Bansal, R. Gupta, and M. Mewara, “Optimal Tuning of PID Controller Parameters for HVAC System Using BF-PSO Algorithm,” in *2023 IEEE 4th Annual Flagship India Council International Subsections Conference: Computational Intelligence and Learning Systems, INDISCON 2023*, 2023, pp. 16–20. doi: 10.1109/INDISCON58499.2023.10270535.

[17] M. Abdelhamid, S. Kamel, L. Nasrat, H. Shahinzadeh, and H. Nafisi, “Adaptive Coordination of Distance and Direction Overcurrent Relays in Active Distribution Networks Based on the Tunicate Swarm Algorithm,” in *2022 12th Smart Grid Conference, SGC 2022*, 2022, pp. 3–8. doi:

10.1109/SGC58052.2022.9998952.

[18] G. R. Vishal and J. Pradeep, "Improved Performance of PMSM using Tunicate Swarm optimization," in *2022 International Conference on Emerging Smart Computing and Informatics, ESCI 2022*, 2022, pp. 1–5. doi: 10.1109/ESCI53509.2022.9758351.

[19] M. S. Hossen, M. R. Islam, M. Shafiullah, M. Fahim-Ul-Haque, and A. Ali, "Tunicate Swarm Algorithm for Power System Stability Enhancement in a SMIB-UPFC Network," in *Proceedings of the 2nd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2022*, 2022, pp. 1767–1772. doi: 10.1109/ICAIS53314.2022.9743091.

[20] D. Izci, S. Ekinci, C. Budak, and V. Gider, "PID Controller Design for DFIG-based Wind Turbine via Reptile Search Algorithm," in *IEEE Global Energy Conference, GEC 2022*, 2022, pp. 154–158. doi: 10.1109/GEC55014.2022.9986617.

[21] D. K. Rukmani and K. A. Folly, "Performance analysis of multiple distributed generation placement in radial distribution system using Reptile Search Algorithm," in *Proceedings of the 3rd International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2022*, 2022. doi: 10.1109/ICSTCEE56972.2022.10100223.

[22] S. Rajput, R. Chawra, P. S. Wani, and S. J. Nanda, "Noisy Sonar Image Segmentation using Reptile Search Algorithm," in *Proceedings of the 2022 International Conference on Connected Systems and Intelligence, CSI 2022*, 2022. doi: 10.1109/CSI54720.2022.9923950.

[23] F. Hasan, F. Ahmad, M. Shahid, A. Khan, and G. Ahmad, "Solving Portfolio Selection Problem Using Whale Optimization Algorithm," in *Proceedings - 2022 3rd International Conference on Computation, Automation and Knowledge Management, ICCAKM 2022*, 2022, pp. 1–5. doi: 10.1109/ICCAKM54721.2022.9990079.

[24] Z. Han, J. Shi, Y. Todo, and S. Gao, "Training dendritic neuron model with whale optimization algorithm for classification," in *Proceedings of 2020 IEEE International Conference on Progress in Informatics and Computing, PIC 2020*, 2020, pp. 11–15. doi: 10.1109/PIC50277.2020.9350790.

[25] A. Alzaqebah, R. Masadeh, and A. Hudaib, "Whale optimization algorithm for

requirements prioritization," in *2018 9th International Conference on Information and Communication Systems, ICICS 2018*, 2018, pp. 84–89. doi: 10.1109/IACS.2018.8355446.

[26] L. Changying, Y. Yuanyuan, H. Yuguang, and A. Bowen, "Flatness Error Evaluation Based on Marine Predator Algorithm," in *2022 IEEE Conference on Telecommunications, Optics and Computer Science, TOCS 2022*, 2022, pp. 1123–1125. doi: 10.1109/TOCS56154.2022.10015983.

[27] P. Marksan *et al.*, "Marine Predators Algorithm Optimization for Economic Load Dispatch with Valve-Point Loading Effects," in *2023 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2023*, 2023, pp. 2–5. doi: 10.1109/ECTI-CON58255.2023.10153269.

[28] S. Shrivastava and P. K. Pathak, "Design of PID Controller for Frequency Regulation of Standalone Microgrid via Marine Predator Algorithm," in *2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT 2022*, 2022, pp. 1–5. doi: 10.1109/IMPACT55510.2022.10029011.

[29] H. Meng, F. Xu, X. Zhang, Z. Yang, W. Nai, and D. Li, "Locally Linear Embedding Based on Neiderreit Sequence Initialized Pelican Optimization Algorithm," in *ICEIEC 2022 - Proceedings of 2022 IEEE 12th International Conference on Electronics Information and Emergency Communication*, 2022, pp. 244–248. doi: 10.1109/ICEIEC54567.2022.9835085.

[30] Z. Zhang, "Mixed Strategy-Based Improved Pelican Optimization Algorithm," 2023, pp. 428–433.

[31] J. H. W. Wkh *et al.*, "Research on Jamming Resource Allocation Based on Improved Pelican Optimization Algorithm," 2022, pp. 123–127.

[32] R. H. Salem, A. Y. Abdelaziz, and M. Ezzat, "Grey-Wolf Optimization Scenario for Security Constrained Optimal Power Flow Problem," in *2018 20th International Middle East Power Systems Conference, MEPCON 2018 - Proceedings*, 2019, pp. 463–468. doi: 10.1109/MEPCON.2018.8635129.

[33] C. Dimna Denny, R. Kumar P, and S. Abhayadev, "Optimum Gain Selection of Sliding Mode Control using Grey Wolf Optimization Technique," in *2022 2nd International Conference on Advances in Electrical, Computing,*

Communication and Sustainable Technologies, ICAECT 2022, 2022. doi: 10.1109/ICAECT54875.2022.9807875.

[34] M. A. Mahdi and L. M. Dawood, “A Grey Wolf Optimization Algorithm for Integrating Process Planning and Scheduling Problem,” in *HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, 2022, pp. 2–6. doi: 10.1109/HORA55278.2022.9799945.

[35] K. Ramzi and M. Souhil, “Optimal power flow incorporating stochastic wind power using Artificial gorilla troops optimizer,” in *2022 19th IEEE International Multi-Conference on Systems, Signals and Devices, SSD 2022*, 2022, pp. 2041–2047. doi: 10.1109/SSD54932.2022.9955921.

[36] N. K. Singh, S. Gope, C. Koley, S. Dawn, and H. H. Alhelou, “Optimal Bidding Strategy for Social Welfare Maximization in Wind Farm Integrated Deregulated Power System Using Artificial Gorilla Troops Optimizer Algorithm,” *IEEE Access*, vol. 10, no. June, pp. 71450–71461, 2022, doi: 10.1109/ACCESS.2022.3186519.

[37] A. Abdelfatah, S. Kamel, H. A. El-Sattar, H. Shahinzadeh, and E. Kabalci, “Optimal Sizing of an Off-Grid PV/Diesel/Battery Storage System Using Gorilla Troops Optimizer,” in *2022 26th International Electrical Power Distribution Conference, EPDC 2022*, 2022, pp. 90–95. doi: 10.1109/EPDC56235.2022.9817224.

[38] R. Chappa, L. Janjanam, and S. K. Saha, “Performance Analysis of Optimal FIR LPF and HPF using AVOA,” in *2022 6th International Conference on Computing, Communication, Control and Automation, ICCUBEA 2022*, 2022, pp. 1–6. doi: 10.1109/ICCUBEA54992.2022.10011081.

[39] M. Zhao, J. Huan, X. Huang, T. Yu, and Q. Ding, “Optimal Planning of Distributed Generation Based on African Vultures Optimization Algorithm,” in *EI2 2022 - 6th IEEE Conference on Energy Internet and Energy System Integration*, 2022, pp. 611–616. doi: 10.1109/EI256261.2022.10116298.

[40] H. E. Alouache, S. Sayah, and A. Hamouda, “Africa vultures optimization algorithm for optimal power flow solution including SVC devices,” in *2022 19th IEEE International Multi-Conference on Systems, Signals and Devices, SSD 2022*, 2022, pp. 2022–2027. doi: 10.1109/SSD54932.2022.9955917.

[41] J. Paramguru, S. K. Barik, A. K. Sahoo, and A. S. Kumar, “Solution to Dynamic Economic Dispatch Problem in the Current Power System Scenario Using Spotted Hyena Optimizer,” in *IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, and Communication, AESPC 2021 - Proceedings*, 2021. doi: 10.1109/AESPC52704.2021.9708460.

[42] G. Dhiman and A. Kaur, “Spotted Hyena Optimizer for Solving Engineering Design Problems,” in *Proceedings - 2017 International Conference on Machine Learning and Data Science, MLDS 2017*, 2018, pp. 114–119. doi: 10.1109/MLDS.2017.5.

[43] Y. V. Krishna Reddy, A. V. Sudhakara Reddy, M. L. Devi, and P. Vinod Kumar, “Economic dispatch solutions with piecewise quadratic cost functions using Spotted hyena Optimizer,” in *Proceedings of the IEEE Madras Section International Conference 2021, MASCON 2021*, 2021. doi: 10.1109/MASCON51689.2021.9563437.

[44] D. M. P. R. V.Muneeswaran, “Networks Based on Tree Seed Algorithm,” 2016.

[45] A. C. Cinar, “Boundary Conditions in Tree-Seed Algorithm,” 2017, pp. 571–576.

[46] E. Kose, “Optimal Control of AVR System with Tree Seed Algorithm-Based PID Controller,” *IEEE Access*, vol. 8, pp. 89457–89467, 2020, doi: 10.1109/ACCESS.2020.2993628.

[47] V. Galzina, R. Lujić, and T. Šarić, “Adaptive fuzzy particle swarm optimization for flow-shop scheduling problem,” *Teh. Vjesn.*, vol. 19, no. 1, pp. 151–157, 2012.

[48] R. M. Rizk-Allah, O. Saleh, E. A. Hagag, and A. A. A. Mousa, “Enhanced Tunicate Swarm Algorithm for Solving Large-Scale Nonlinear Optimization Problems,” *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, 2021, doi: 10.1007/s44196-021-00039-4.

[49] F. Taher, M. Elhoseny, M. K. Hassan, and I. M. El-Hasnony, “A Novel Tunicate Swarm Algorithm with Hybrid Deep Learning Enabled Attack Detection for Secure IoT Environment,” *IEEE Access*, vol. 10, no. December, pp. 127192–127204, 2022, doi: 10.1109/ACCESS.2022.3226879.

[50] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Engineering

Applications of Artificial Intelligence Tunicate Swarm Algorithm : A new bio-inspired based metaheuristic paradigm for global optimization ☆," *Eng. Appl. Artif. Intell.*, vol. 90, no. December 2018, p. 103541, 2020, doi: 10.1016/j.engappai.2020.103541.

[51] M. Korkmaz, Ö. Aydoğdu, and H. Doğan, "Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller," in *INISTA 2012 - International Symposium on INnovations in Intelligent SysTems and Applications*, 2012. doi: 10.1109/INISTA.2012.6246935.

[52] W. Lv, D. Li, S. Cheng, S. Luo, X. Zhang, and L. Zhang, "Research on PID control parameters tuning based on election-survey optimization algorithm," in *2010 International Conference on Computing, Control and Industrial Engineering, CCIE 2010*, 2010, pp. 323–326. doi: 10.1109/CCIE.2010.198.

[53] A. R. Gopisetti, B. A. Reddy, K. Anusha, and R. P. Kumar, "Performance Evaluation of Second Order Sliding Mode Control Strategies for a Coupled Tank System," in *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, 2018, pp. 1–7. doi: 10.1109/ICCTCT.2018.8551080.

[54] B. A. Reddy and P. V. Krishna, "Traditional and 2-SMC Control Strategies for Coupled Tank System," in *Proceedings of the 3rd International Conference on Inventive Systems and Control, ICISC 2019*, 2019, pp. 682–687. doi: 10.1109/ICISC44355.2019.9036440.

[55] S. R. Mahapatro, B. Subudhi, S. Ghosh, and P. Dworak, "A comparative study of two decoupling control strategies for a coupled tank system," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2017, pp. 3447–3451. doi: 10.1109/TENCON.2016.7848695.

[56] D. Mukherjee and K. Rastogi, "A performance analysis between IOPID and FOPID controller on a coupled tank," in *Proceedings of the International Conference on Inventive Systems and Control, ICISC 2017*, 2017, pp. 1–5. doi: 10.1109/ICISC.2017.8068592.

[57] M. M. Gulzar, M. Munawar, Z. Dewan, M. Salman, and S. Iqbal, "Level Control of Coupled Conical Tank System using Adaptive Model Predictive Controller," in *HONET 2020 - IEEE 17th International Conference on Smart

Communities: Improving Quality of Life using ICT, IoT and AI, 2020, pp. 236–240. doi: 10.1109/HONET50430.2020.9322842.

[58] S. Bhoi and R. Pradhan, "Modeling of Fractional Order PID Controller and study the Application of Couple Spherical Tank Level Control," 2023, pp. 8–11.

[59] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Syst. Appl.*, vol. 191, no. May 2020, p. 116158, 2022, doi: 10.1016/j.eswa.2021.116158.

[60] D. Wu, C. Wen, H. Rao, H. Jia, Q. Liu, and L. Abualigah, "Modified reptile search algorithm with multi-hunting coordination strategy for global optimization problems," *Math. Biosci. Eng.*, vol. 20, no. 6, pp. 10090–10134, 2023, doi: 10.3934/mbe.2023443.

APPENDIX

TABLE 1 : SISO data with 20 times execution using TSA

Run	ITSE	Time	PID Parameter		
			P	I	D
1	0.3531	54.87	10000	31.9580	9590.5630
2	0.3531	46.08	10000	31.9649	9591.1218
3	0.3525	46.91	10000	20.8078	9590.0483
4	0.3540	46.60	10000	44.7178	9583.2867
5	0.3512	50.34	10000	0.1	9595.8067
6	0.3545	45.15	10000	54.6230	9585.4360
7	0.3512	49.94	10000	0.1	9595.4700
8	0.3522	46.68	10000	17.2345	9592.5617
9	0.3545	48.35	10000	53.2353	9583.8012
10	0.3512	48.18	10000	0.2512	9595.8799
11	0.3512	55.19	10000	0.1	9596.7564
12	0.3512	44.66	10000	0.1104	9596.1308
13	0.3511	45.26	10000	0.1091	9597.7062
14	0.3512	45.40	10000	0.8279	9597.3956
15	0.3541	40.89	10000	46.5348	9585.3276
16	0.3569	57.68	10000	83.7633	9562.8725
17	0.3543	41.13	10000	51.5816	9584.7035
18	0.3530	42.28	10000	30.2916	9590.3709
19	0.3545	50.65	10000	55.6992	9584.9866
20	0.3538	43.62	10000	42.2808	9585.0311

TABLE 2 : SISO data with 20 times execution using RSA

Run	ITSE	Time	PID Parameter		
			P	I	D
1	0.3755	74.89	8648.8335	38.4542	8809.5611
2	0.4827	47.76	4658.5824	17.9813	6047.6268
3	0.4692	43.20	4980.9853	25.0539	6315.1764
4	0.5956	84.18	1766.6494	2735.9599	9940.9953
5	0.4495	43.12	5944.7926	0.4880	6777.8762
6	0.3744	44.71	8973.3044	2.8725	8836.6941
7	0.4132	48.12	7731.3673	14.2641	7766.5286
8	0.4969	43.47	7350.5225	57.5514	5550.5237
9	0.4077	51.91	7609.1529	64.4402	7981.4502
10	0.4690	46.28	4807.9634	0.6662	6218.456
11	0.4668	46.45	5409.1009	26.5933	6478.1557
12	0.5409	40.59	3499.5890	10.7358	5087.8383
13	0.6143	52.94	2642.9100	5.6776	4219.9479
14	0.3824	47.61	9020.1608	96.2711	8789.4937
15	0.4512	54.87	5792.9081	3.0113	6737.5477
16	0.5625	64.49	3145.9415	1202.7588	9874.446
17	0.3539	47.52	10000	20.4944	9551.7129
18	0.4695	45.31	4799.8454	11.2264	6233.4176
19	0.3928	47.66	8213.6888	18.4561	8260.6477
20	0.6199	75.34	1999.1907	1900.3626	9438.1073

TABLE 3 : SISO data with 20 times execution using PSO

Running	ITSE	Time	PID Parameter		
			P	I	D
1	0.35117	50.88	10000	0.1	9598.70
2	0.35118	45.95	10000	0.1	9598.6890
3	0.35118	40.20	10000	0.1	9598.6890
4	0.35117	42.00	10000	0.1	9598.6959
5	0.35118	39.19	10000	0.1	9598.6959
6	0.35118	40.67	10000	0.1	9598.6890
7	0.35118	38.76	10000	0.1	9598.6890
8	0.35117	41.04	10000	0.1	9598.6959
9	0.35117	40.56	10000	0.1	9598.6959
10	0.35117	40.43	10000	0.1	9598.6959
11	0.35117	51.60	10000	0.1	9598.6959
12	0.35118	49.30	10000	0.1	9598.6890
13	0.35118	42.97	10000	0.1	9598.6890
14	0.35117	38.44	10000	0.1	9598.6959
15	0.35118	44.88	10000	0.1	9598.6890
16	0.35118	36.38	10000	0.1	9598.6890
17	0.35118	36.70	10000	0.1	9598.6890
18	0.35117	43.22	10000	0.1	9598.6959
19	0.35117	41.15	10000	0.1	9598.6959
20	0.35117	40.30	10000	0.1	9598.6959