MULTIVARIABLE PID CONTROLLER DESIGN TUNING USING OPTIMIZATION TECHNIQUE FOR ACTIVATED SLUDGE PROCESS

NUR ATIKAH BINTI NOR'AZLAN

A report submitted in partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

JUNE 2017

"Hereby declare that I have read through this report entitle "Multivariable PID Controller Design Tuning using Optimization Technique for Activated Sludge Process" and found that it has been comply the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)".

Signature : ……………………………...................

Supervisor's Name : NUR ASMIZA BINTI SELAMAT

Date : 15 JUNE 2017

I declare that this report entitle "Multivariable PID Controller Design Tuning using Optimization Technique for Activated Sludge Process" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……………………………......................

Name : NUR ATIKAH BINTI NOR'AZLAN

Date : 15 JUNE 2017

ACKNOWLEDGEMENT

First of all, I am grateful to **The Almighty God** for establishing me to complete this final year project.

I place on record, my sincere gratitude to Puan Nur Asmiza Binti Selamat, lecturer of Electrical Faculty and also my kind supervisor. I am extremely grateful and indebted to her for her expert, sincere and valuable guidance and encouragement extended to me during all time until project being finished.

I take this opportunity to express my deep sincere thanks to my parents and members for their cooperation and encouragement while helping me settled this project until became successful.

Lastly, I also place on record, my sense of gratitude to one and all who, directly or indirectly, have lent their helping hand in this venture.

ABSTRACT

The designing of a multivariable PID control for multi input multi output is being concerned in this project by applying four multivariable PID control tuning which are Davison, Penttinen-Koivo, Maciejowski and Proposed method. The determination of this study is to investigate the performance of selected optimization technique to tune the parameter of MPID controller. The selected optimization techniques are Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Bat Algorithm (BA). The best MPID method which is Proposed has been choose to be as a controller tuning method from the all methods of MPID result tuning that have been compared and analyzed. Later, the Proposed has been compared between PSO, GA and BA in order to determine which optimization techniques are better based on the system performances in terms of transient response. The result obtained for the best optimization techniques to be used in Activated Sludge Process (ASP) was the Bat Algorithm with the Proposed control tuning method. This project also was done by simulated the algorithm in a different types of ASP system which are linear and nonlinear system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Activated sludge process (ASP) is biological process that suspended growth secondary treatment process of Wastewater Treatment Plant (WWTP). It is a primarily removes dissolved organic solids while involving in controlling the concentration of microorganisms and sludge particles that are naturally found in unsettled wastewater. Besides, the controller parameters stay remained constant once after the plant been commissioned and this process involves a number of interacting controls. Therefore, a proper tuning of multivariable PID will improves the performances of WWTP while the parameter tuning can be obtained using optimization technique.

1.2 Motivation

Wastewater treatment plant (WWTP) is become very important nowadays due to the increasing of environmental awareness around the world. This situation cause the improvement of WWTP is driving enforcement because of the tightened requirements for the effluents before being released into the river. Activated Sludge Process (ASP) is secondary treatment process of WWTP and quite popular to be known as biological process. The complexity process of ASP make it becomes difficult to be handled. Thus, it can affect the performance of WWTP producing high quality effluent. With the proper tuning of multivariable PID, the performances of WWTP will be improved. Bat Algorithm (BA) technique of optimization was used in this project to gain the parameter tuning of MPID.

1.3 Problem Statement

Lately much of the efforts are giving an attention on new sources of clean energy, transportation and of course, wastewater treatment. An accurate treatment of the wastewater is a common issue in all metropolises before its discharge into the receiving water. Besides, since the ecological awareness is more important for people and politicians nowadays, the quality standards for Wastewater Treatment Plants (WWTP) becoming constricted [7]. WWTP also is categorized as a complex system. Therefore, an effective control methods need to be implemented for economic and environmental reasons, especially its Activated Sludge Process (ASP) because it involve a biological process. Unfortunately, an increasing claim for a more stable effluent water quality [6] makes a scalar PID based control systems are become insufficient anymore due to complexity of the system such as interrelated and highly nonlinear of its biological, physical and chemistry phenomena. Hence, as the system becomes more complex, the process of tuning controllers also becomes more difficult [8].

Then, the multivariable control systems are really needed to overcome that problem by using the suitable tuning method of its controllers and optimization techniques that can helps obtained an optimal solution to get the best values of parameter tuning in order to have good transient response performances of the system.

1.4 Objectives

The aim of this project is to gain parameter tuning based on selected optimization technique. So, the goals of this project are:

1) To implement the Multivariable PID (MPID) control tuning method of Activated Sludge Process.
2) To tune MPID parameter using optimization technique, Bat Algorithm (BA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for MPID control tuning method.
3) To compare the system performances in term of transient response between selected optimization techniques

1.5 Project Scope

This project implements MPID controller design for activated sludge process. Davison, Penttinen-Koivo, Maciejowski and Proposed Combined method are the four types of MPID tuning that will be used in this project. Then, the scalar parameter MPID controllers are being adjusted by the selected optimization techniques which are Bat Algorithm (BA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in the system. All the simulation steps are done by using MATLAB/SIMULINK software and the results of this project are presented based on the performances of linear and non-linear system in terms of transient response and performances index.

1.6 Project Report Summary

This thesis basically is distributed into five chapters and this section delivers a brief overview of the chapters comprised in it.

Chapter 1: Introduction

This section guides readers to the elementary of this project, such as overview of activated sludge process, objectives, problem statement and project scope towards it.

Chapter 2: Literature Review

This part provides a simple explanation on the concept and previous work of related literature studies. The activated sludge process flow, controllers, MPID tuning and optimization techniques are being reviewed.

Chapter 3: Methodology

In this section, the project flow and methodology along to accomplish this project are being presented. Davison, Penttinen-Koivo, Maciejowski and Proposed Combined method are the types of MPID tuning that will be explained. Instead, the process of implementation of optimization technique will be stated in this chapter

Chapter 4: Result and Discussion

This section shows the results of system performance in term of transient response by using MPID tuning method. Its scalar parameter is being obtained via tuning with selected optimization technique. The results are compared and the specific discussions will also include in this section.

Chapter 5: Conclusion and Future woks

This section contains of conclusion based on the whole methodologies and outcomes. Then, some suggested work that can be done for a future is also mention in it.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explained about the definition of related subject or issues for this project. From this review, a several MPID methods and optimization technique will be chosen for additional studies. Hence, this section also reviews the related previous researches that have been done and other topic connected with this project.

2.2 Activated Sludge Process

Wastewater treatment plant (WWTP) is a place where the wastewater treatment process being carried out. The wastewater treatment process is designed in order to achieve enhancement in the quality of the wastewater that including three stages of processes [1] which are primary, secondary and tertiary as shown in Figure 2.1. The primary stage is also known as mechanical process, was aimed to eliminate gross, suspended and floating solid from raw sewage by including screening to trap solid object and sedimentation. Secondary stage is biological process. While, tertiary or advanced stage is a last stage where from this stage, almost all the impurities from the sewage will be removed.

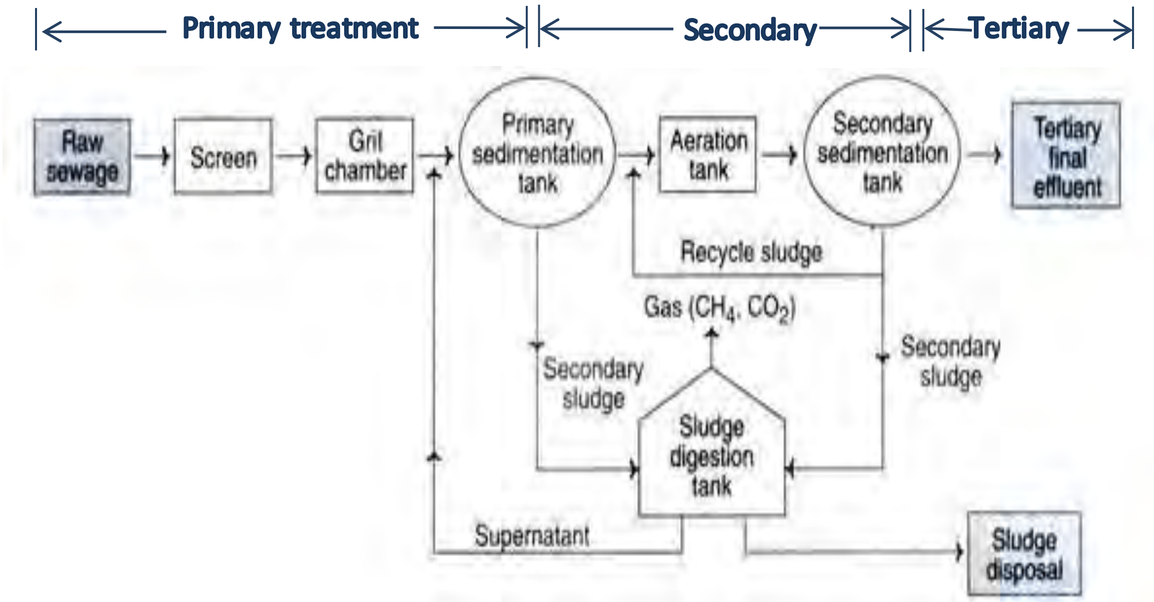


Figure 2.1: The Wastewater Treatment Process

Activated sludge process is the most widely used for biological wastewater treatment plant [2] and a type of secondary treatment where a high level of elimination of biodegradable organic pollutants are given to keep receiving water quality that clarification alone cannot provide. The activated sludge process also can speeds up decomposition by adding an activated sludge into the wastewater where the activated sludge particles hold many living organisms that can feed on the incoming wastewater [3]. This process is divided into two parts; an aerator tank which where the growth of organisms take place and a secondary settling tank, in where the leaving of clear liquid free of organic material happened [4].

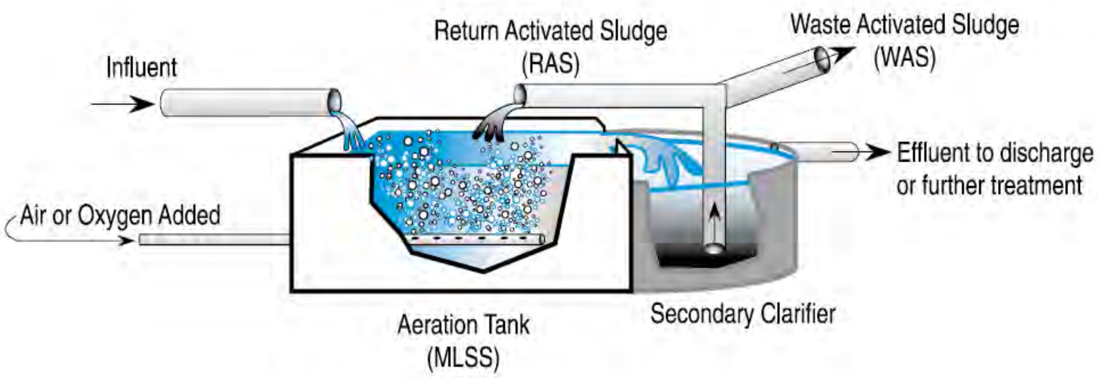


Figure 2.2: Activated Sludge Process

Figure 2.2 illustrated on how the process of activated sludge system takes an action [5]. This process starts with the incoming of influent to the aeration tank. Aeration tank is where the place of biological react occurred to segregate wastes from water and form waste decomposition. During that time, microorganisms are inject in and being contact in the wastewater; they feed and grow by the source of oxygen supplied into the tank. Then, the mixed liquor which is the mixture of wastewater and microorganisms will flow into secondary clarifier and begin to aggregation together. During the bio-flocculation process, the particles start to clump together, called floc and it will be settle to the bottom tank of the clarifier as sludge (separated completely from water). From the secondary clarifier tank, the relatively clear liquid above the sludge will flow on for further treatment while the sludge is driven back to the aeration tank, named Return Activated Sludge process (RAS). Lastly, the sludge that is intentionally eliminated from the ASP is denoted as Waste Activated Sludge (WAS).

2.3 Activated Sludge Process Controller

Nowadays, the health of natural ecosystems has been giving more attention cause of the effect from the human development in very different ways [6] by increased environmental awareness in terms of water pollution anticipation. The constricted laws and requirements toward quality of water are fortunately acting like a driving force for the development of wastewater treatment plants (WWTPs) [7]. The complex, interrelated and highly nonlinear of its biological, physical and chemical phenomena make the WWTP become more difficult to be controlled while optimizing operating and management costs [8].

Meanwhile, the activated sludge process (ASP), a biological processes are usually popular methods used to remove carbon as well as components of nitrogenous from wastewater beforehand it being released [9]. ASP has been widely case study in the automatic control perspective, for example by Yong et al. [10] the concentration of ammonia in the fluent of the wastewater plant is reducing by implementation of cascading PI-like controllers with feed-forward actions. Different with the model predictive control (MPC) by Holenda et al. [11] proposed that MPC method determining and controlling the dissolved oxygen concentration only. A decreasing of more than 25% in power usage and an increasing in plant efficiency are the significant benefits that can be obtained of using the MPC system [12].

A previous study done by Shen et al. [13] where a multiple input approach is implemented, by the recycle flow rates, the oxygen transfer coefficient of three aerated tanks and a complementary carbon source. Fuzzy controller has proved its efficiency to be implemented at WWTPs for improving the denitrification or nitrification process but these methods gives a highly cost and behaves relatively rough toward its control actions [14].

Rojas et al. [15] proposed that in order to be able to actuate based on the measurements of the disturbance, a three degree of freedom controller, tuned with the Virtual Reference Feedback Tuning (VRFT) is offered and applied to the WWTP. The three degree of freedom controller is an extended from a two degree of freedom PI which where control of nitrate concentration by manipulating the internal recycle flow rate, plus disturbance feed-forward action that can control ammonia using ammonia measurements from the influent. This methodology viewing the usability of model-free approach for WWTP control although there are no clear rules to select neither an appropriate close-loop target function nor the correct parameterization of the controllers

2.4 Multivariable PID

PID stands for Proportional Integral Derivative is a type of controller that encouraging to be used with more than 90% of industrial controllers due to its well-known robustness and its straightforwardness where the structure is easy to understand [16]. The structure of PID a combination of "three terms" and the transfer functions are given by (2.4.1) and (2.4.2) where proportional term represented gain factor, integral term by an integral and derivative term by a differentiator. Each of them has their own specific character to make sure better system performance. P-term decreases error but does not remove it. I-term eliminates the error but have a tendency to make the system oscillate while D-term improves the speed of the responses [17].

$$G(s) = K_p + \frac{K_i}{s} + K_d s \tag{2.4.1}$$

$$G(s) = K_p\left(1 + \frac{1}{\tau_i s} + \tau_d s\right) \tag{2.4.2}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$
(2.4.3)

Along the implementation of PID controller due to the popularity of its advantages, this controller has limitation to control multivariable system. Multivariable process is a system with combination more than one variables at the input or output to be controlled in a system and a type of MIMO system which means multi-input, multi-output system. Figure 2.3 shows a multivariable system with PID controller and its transfer function matrix is stated at (2.4.3).

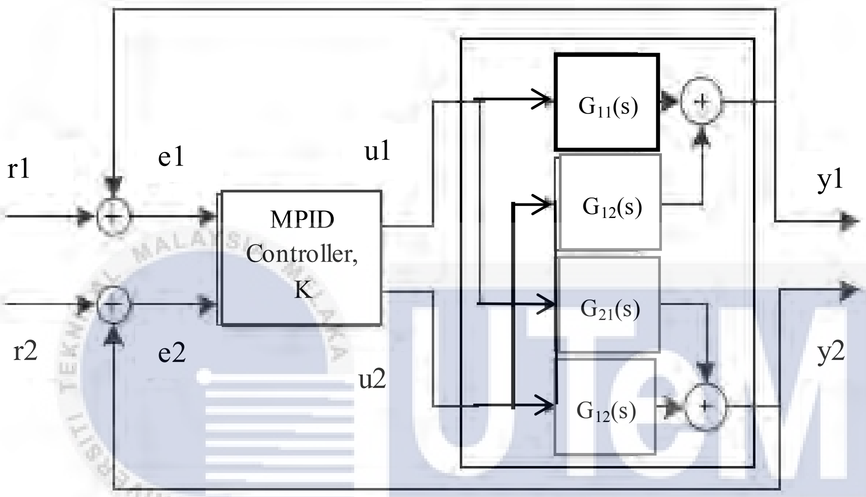


Figure 2.3: Multivariable System with PID Controller

MIMO system can be divided into two, centralized and decentralized controller. Centralized controller only involves one loop while decentralized controller involves several loops. The multivariable system is the system that apply centralized controller since its deal with MIMO by using single controller. Then, the decentralized controller was applied to another system called Multi-loop where each control variables are controlled by different controller [18]. Figure 2.4 shows a centralized controller for multivariable system, while Figure 2.5 shows a decentralized controller for multi-loop system.

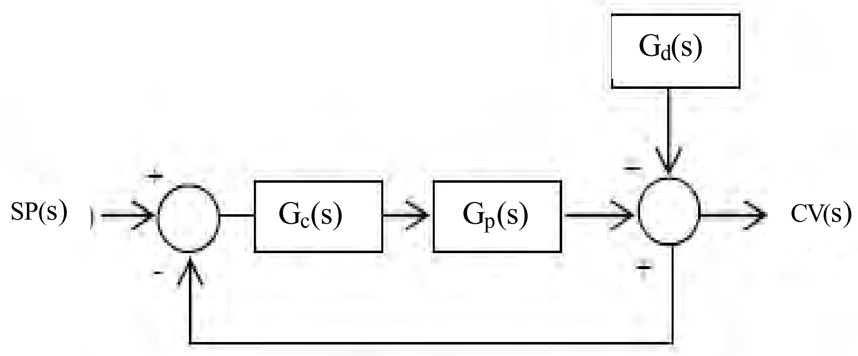


Figure 2.4: Centralized Controller for Multivariable System

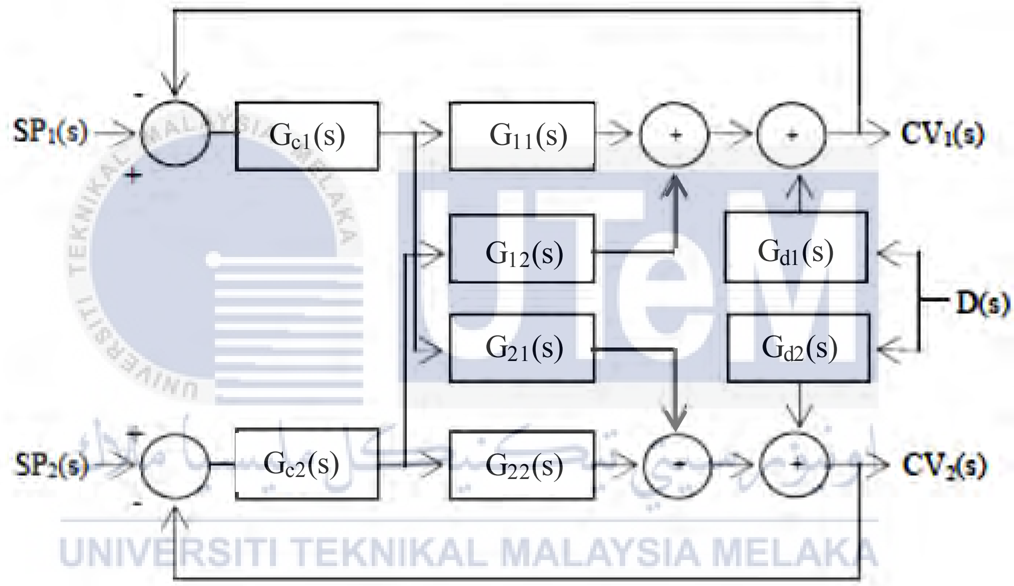


Figure 2.5: Decentralized Controller for Multi-loop System

Unfortunately, sometimes interactions phenomena can be occurred in MIMO system and it happen when loop gain in one loop also depends on other loop gains, particularly multivariable system [19]. Differ to the multi-loop system that does not acknowledge the interaction phenomena due to the structure of loop control which base on a single loop basis [20]. An interaction effect can be shown in (2.4.4) and (2.4.5). Besides, another factor which is quite important in multivariable system is *input/output* pairing problem. A number of quantitative techniques can be used to determine the right pairing of the manipulated and controlled variables [19]. So, the MPID is challenging to build rather than PID that implement in SISO system because it is does not involved an interaction [21].

$$y_1 = u_1 G_{11}(s) + u_2 G_{12}(s) \tag{2.4.4}$$

$$y_2 = u_1 G_{22}(s) + u_2 G_{21}(s) \tag{2.4.5}$$

2.5 Multivariable PID Tuning Method

Multivariable PID tuning method can be classified into two parts; parametric and non-parametric methods. Parametric methods use whichever model or experiment data to determine the controller parameters and are mostly defined as offline tuning methods, through online approaches have also been tested. While, non-parametric methods only partly use models such as critical states and are suitable for online use as well as for application without previous extensive plant studies [22]. The example of parametric tuning methods are Biggest Log modulus Tuning (BLT), gain and phase margin, minimum variance control, internal model control, and robust decentralized method whereas Davison, Penttinen-Koivo and Maciejowski are categorized on non-parametric tuning methods. Figure 2.6 shows the PID turning method classification.

Virtual Reference Feedback Tuning (VFRT) is an example of one-shot technique which means only one set of data required to define the controller. Campi et al. said that VRFT method converts the model reference control problem into an identification problem, where the controller is the transfer function to be identified based on some "virtual signals" figured from a batch of data taken directly from an open loop system [23]. This method has been implemented to a variety of cases [24, 25, 26] including in the field of WWTPs [27] because of its easiness of implementation, flexibility to be used in different kind of control systems and the characteristic of using only data from the system to minimize a simple optimization problem before the parameters of digital controllers could be determined.

Another one-shot method is Correlation based Tuning (CbT) by Karimi et al. [28]. CbT can finds values of a limited order controller by minimizing the relationship between closed-loop error of the system (based on a desired closed-loop behavior) and the reference to the process.

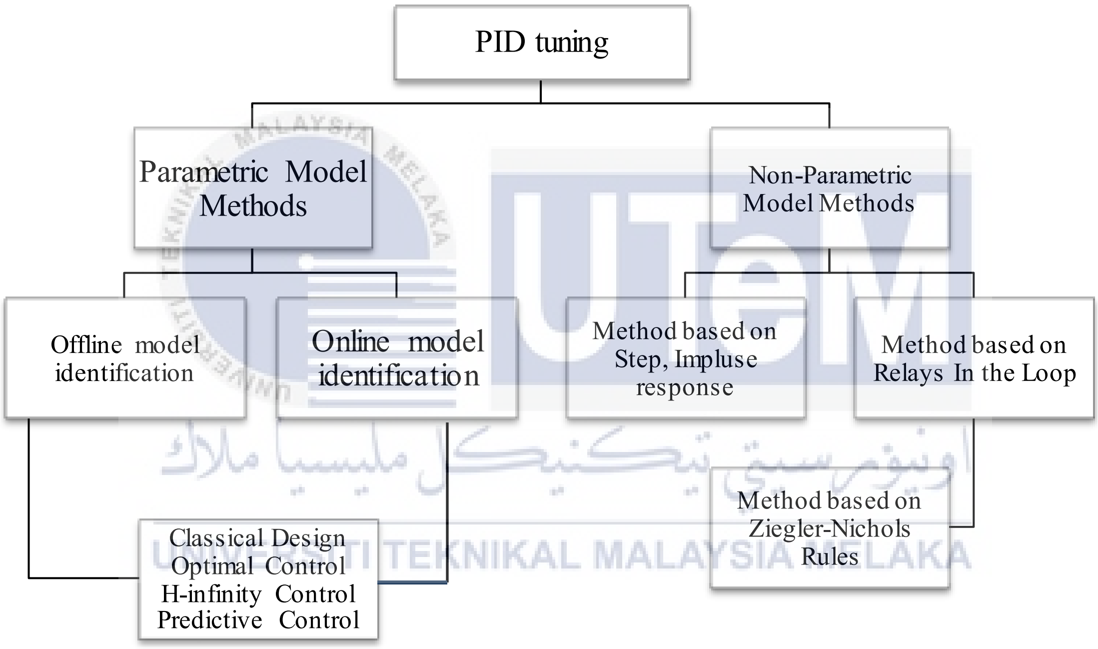


Figure 2.6: PID Turning Method Classification

Hjalmarsson et al. in [29] had study an Iterative Feedback Tuning (IFT) which is an example of iterative methods tuning (several experiments have to be performed in order to express the controller of the system). IFT work out on an unbiased gradient of a performance index to improve iteratively the tuning of the parameters of a reduced order discrete time controller. The multivariable IFT techniques with a multivariable step wise safe switching algorithm have been proposed by Ginestet [30]. The need for identifying the MIMO linear models that describe the plant around trim points can be eliminates using the IFT algorithm for tuning safe switching controllers.

Davison, Penttinen-Koivo and Maciejowski are always been selected due to its capability in dealing with interaction in simple ways and involves only plant step tests or plant frequency responses at a single frequency [31]. The Davison method presents decoupling at low frequency by a constant gain compensator, also guarantees asymptotic stability and asymptotic tracking for a particular form of disturbances. Differs to the Penttinen-Koivo, this method decouples at high frequency. Maciejowski method, the plant is diagonalized at a particular bandwidth frequency to minimize the interaction around the system bandwidth. Instead of three methods mentioned, N. A. Wahab [32] was introduced The Proposed Combined method that uses the principles of Maciejowski in diagonalizing system near the bandwidth frequency to improve the control performance and ease the difficulties in finding a right bandwidth frequency.

2.6 Optimization Technique

Optimization is process of finding an ideal solution to get the best values of parameters for the problem under stated conditions by execute procedure in comparing several solutions till an optimum solution is found using an optimization algorithm. In other words, it is the process of modifying the inputs for a vector to find the maximum and minimum result. Problems in optimization can be classified into four groups; Metaheuristic, Combinatorial optimization, Linear Programming and Integer programming [33, 34].

Optimization problem can be settled by using two main methods; Stochastic and Deterministic algorithms [35]. Stochastic algorithms characterized on probability translation rules where it has randomness embedded in their nature. Even if the initial conditions are kept same, it will produce different solutions at every run. While for the deterministic algorithms, it makes use of certain rules for moving from one solution to other solutions. These algorithms have a tendency to get stuck in local optima because it will generate the same result for every run, if their starting conditions are the same. Therefore, outsider researchers are preferred the stochastic algorithms to avoid from local minima.

Metaheuristic algorithms are solution approaches that conduct an interaction between local improvement procedures and higher level strategies to generate a process capable of escaping from local optima and performing a robust search of solution space. These algorithms are part of stochastic algorithms where it derives their inspiration based on nature world. Besides, the capability to handle complex non-linearity, discontinuities in the objective function, discrete handling and multi-objective optimization are the advantages of metaheuristic algorithms [36]. The easy implementation and accurate result production makes metaheuristic algorithms are proposed to be the most efficient algorithms to solve the optimization problems.

There are certain types of optimization algorithm that fall under metaheuristic technique. Figure 2.7 shows the metaheuristic algorithm for optimization where most of them are inspired by nature or animal.

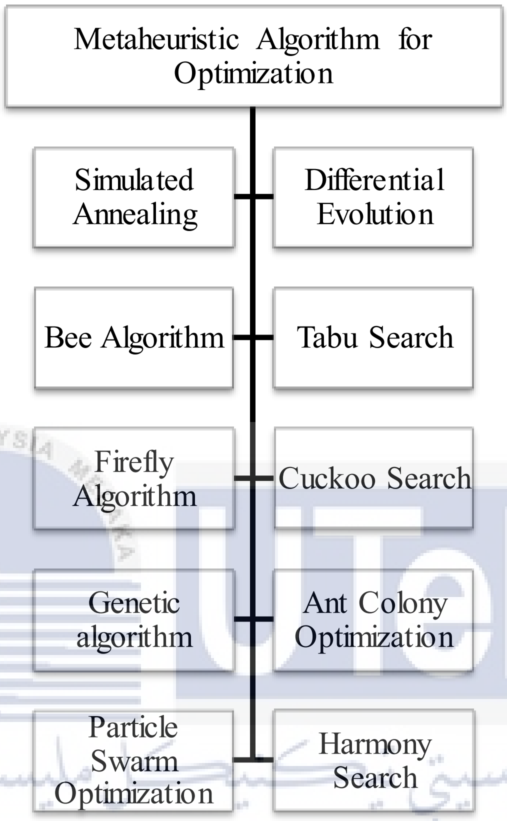


Figure 2.7: Metaheuristic Algorithm for Optimization

In 1983, Kirkpatrick et al. was introduced one type of the metaheuristic technique, Simulated Annealing (SA), which popular with its trajectory-based [37]. Its working principles are based on the technique of nature performs an optimization of the energy of a crystalline solid when it is annealed to eliminate defects in the atomic arrangement. In other words, it based on the annealing process of metals during heat treatment. Fred Glover recommended a new approach, which he called Tabu search in 1986. The main objective of this algorithm is to allow local search (LS) methods to overcome local optima. The advantage of Tabu search is improve efficiency of the exploration process by keeping track of information and decisions used previously during the search [38].

Meanwhile, Harmony search is a music-based metaheuristic optimization algorithm, proposed by Zong Woo Geem et al. in 2001 [39]. It was inspired by the observation that the aim of music is to find for a perfect state of harmony where the perfectly harmony is determined by an audio aesthetic standard. Evolutionary algorithms (EA) are the name for a subcategory of Evolutionary computation model using randomness and genetic operators to a specific degree [34]. Genetic algorithm (GA) and Differential evolution (DE) are categorized under evolutionary algorithm.

Swarm intelligent (SI) is inspired based on the nature collective behavior of likes flocks of birds or social insects. Their characteristics such as co-evolution, self-organization and learning during iteration process makes SI become more popular among metaheuristic algorithms. Groups of metaheuristic technique based on SI are Ant Colony Optimization (ACO), Firefly Algorithm (FA), Cuckoo Search (CS), Bee Algorithm and Particle Swarm Optimization (PSO).

Ant Colony Optimization (ACO) was introduced by Dorigo and further developed by the others developers. The characteristic of behavior of social ants in searching the best and shortest path to the food source was became an inspiration of ACO [40]. In 2007, Xin-She yang was proposed another metaheuristic algorithm which is Firefly Algorithm. It was develop based on idealization of the flashing characteristic of swarming fireflies in the tropical summer [41]. While, the CS algorithm is a population based, on the brooding characteristic of bird types, named cuckoo that lay egg in nest of other birds. This algorithm was introduced by Yang and Deb in 2009. This algorithm also can be used competently for the removal noise from the network [42]. Last but not least, the other class of metaheuristic algorithms is bee algorithms. This bee algorithm was inspired by the foraging behavior of bees [43] and this algorithm can be further expanding into another concept of bee-metaheuristic algorithms.

2.6.1 Bat Algorithm

Bat Algorithm (BA) is another type of metheuristic swarm intelligent optimization algorithms. It was introduced by Yang in 2010 and was inspired by the behavior of micro bats which use echolocation pulses with different emission and sound. On the other hands, it is based on the echolocation capability of micro bats guiding them on their foraging actions [44]. Yang who idealized rules of this algorithm presents that all bats use echolocation to sense distance. Besides, they can differentiate between victim and surrounding barriers in some unknown way because Yang found that the bats can automatically adjust their speed depending on the proximity of their target although they actually are fly randomly with uncertainly speed. In [45] stated that BA also contributed in development of the global numerical optimization where it is inspired by the social manners of bats and the phenomenon of echolocation to sense distance.

2.6.2 Genetic Algorithm

In a period of time between 1960s and 1970s, Genetic algorithms (GA) were presented by John Holland based on the supposition of natural selection and genetics, also known as population-based. Its working principles are used techniques inspired by evolutionary biological like crossover (recombination), mutation and selection for adaptive and artificial systems [46]. GA technique is widely used and quite popular among evolutionary algorithms due to its advantages of GA over traditional optimization algorithms are the ability of dealing with complex problems and parallelism. GA is also a popular approach to optimize non-linear systems with a large number of variables.

2.6.3 Particle Swarm Optimization

James Kennedy and Russel Eberhart was developed an optimization technique for continuous nonlinear functions in 1995, and it called Particle Swarm Optimization (PSO) which it is categorized under population-based metaheuristic algorithms [36]. The working principle of PSO based on social sharing of swarm for example fish schooling or birds flocking. It can be imagined that each particle symbolizes a bird, while the swarm model as particle in space. The swarm of particle communicates through adjustment of velocity and position. PSO concept can be explained in more details by using this example situation, for the bird to hunt food in the field. Each of the birds that fly around will remember their last history, if they found the other location that much better before the past and they will capture that memory to make a selection towards that place with more source of food. The effectiveness of PSO makes it widely used in several of field. Moreover, PSO has been found modest, quite easy to understand and it keeps an eye on the principles of natural selection and search algorithm [47].

.

2.7 Performances of BA, GA and PSO based on Previous Research

Bat algorithms seem quite popular among optimization techniques because there are some enhancement towards this algorithm from the previous researcher to improve the performance of system such as a Binary Bat Algorithm (BBA) [48], Bat Algorithm for Multi-objective Optimization (MOBA) [49] and others [50]. In [51], Jonathan Perez et al. used a fuzzy system to make some enhancement of the BA to improve the performance of the algorithm compared to other metaheuristic optimization.

In previous research by Shubham et al. [52], they came with strong conclusion that GA based optimization on PID controller gives least rise time and settling time among Particle Swarm Optimization (PSO) and Bacteria Foraging Optimization (BFO). GA also has been proven to be more efficient in many fields such as fuzzy logic control design, system identification and others.

A study of performance between conventional gain tuning and modern heuristic approach was done on the model of a DC motor in 2011 by Mahmud et al. [53]. A Ziegler-Nichols method of conventional gain tuning and PSO of modern heuristic approach was being selected in this study. From the results obtained, the came with a strong conclusion that conclude that the designed PID controllers using PSO-based optimization have less overshoot compared to that of the classical method, Ziegler-Nichols although the classical method is good for giving us as the starting point of what are the PID values. Therefore, the advantage of using a modern optimization approach is observed as an effective method to improve performance of the PID system and PSO is the one of the current and effective optimization tools among of them.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This topic comprises the methodological issues that had been used in this project. The main resolution of this topic is to gather information that related to the approaches and techniques that have been used in this chapter development. Each stages of the work are divided into several sections which are activated sludge process, MPID, optimization technique and objective function and simulation.

3.2 Project Flow

A flow chart is used to illustrate the flow of this study and it shows in Figure 3.1 and Gantt chart in Appendix A1. After the selection a proper title from the offered title list was done, a project planning and outcome was determined with the monitor from supervisor, such as the objectives, project scope, methodology and project report summary of this project. Then, a literature review is done based on each subject related in order to understand better about the study and previous work of a researcher about the topic. From a lot of reading during collecting all the information, the ASP modeling from the most reliable previous research had been chosen.

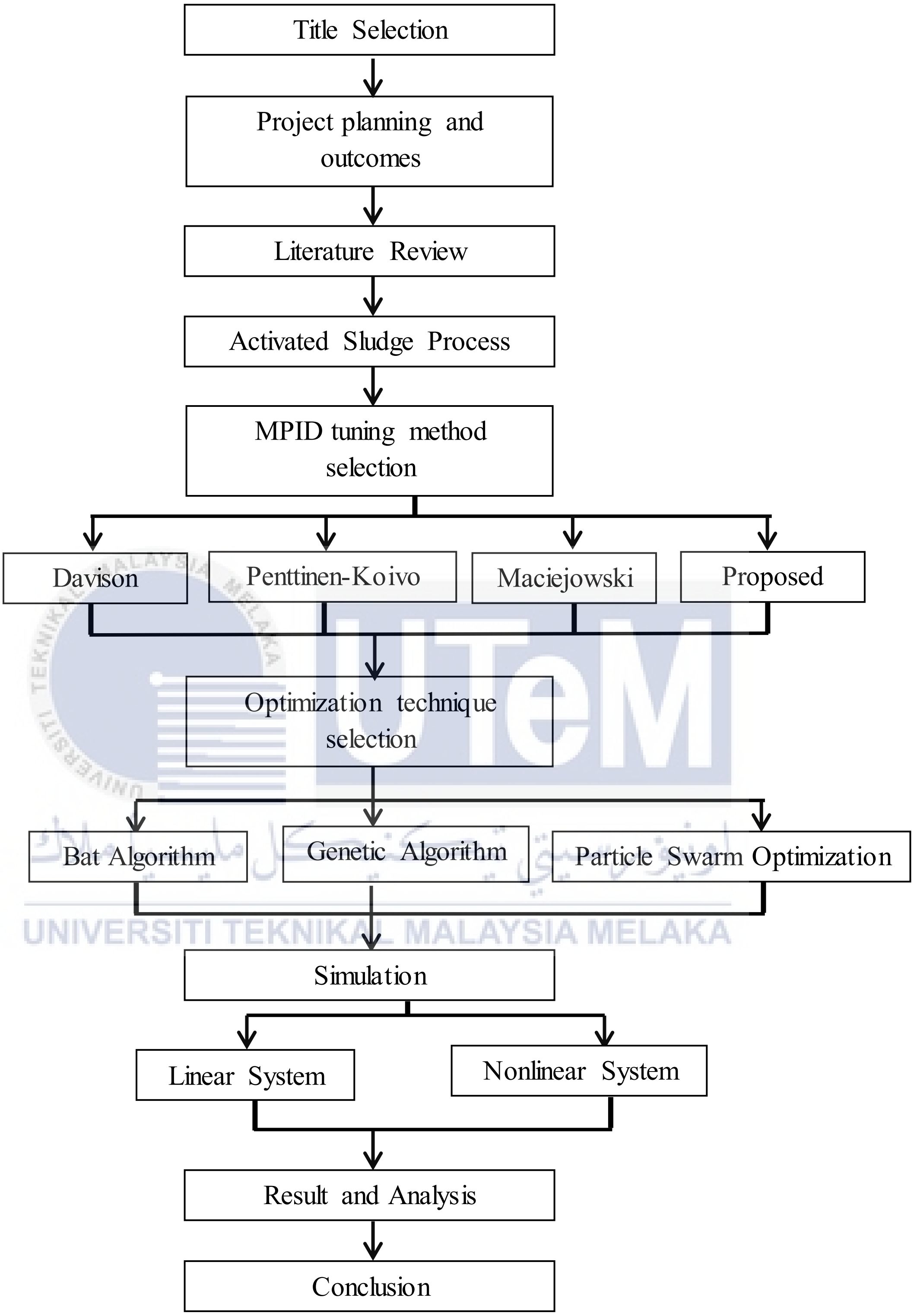


Figure 3.1: Methodology Flowchart

In this project, there are four methods of MPID turning has been selected which are Davison, Penttinen-Koivo, Maciejowski and Proposed Combine method. This selection is based on its ability in dealing with interaction in simple ways and involves only plant step tests or plant frequency responses at a single frequency [31]. From the previous research, optimization technique was already known and used in ASP control modeling for example GA and PSO. Based on the reading, Bat Algorithm (BA) was selected as new optimization technique that will be used for ASP. Therefore, they will also being applied to compare with the selected optimization technique.

After all the related point regarding to this project were determined, the system will be develop and simulate by using MATLAB Simulink software in the differences mode of system which are linear and nonlinear system. Then, the result will be analyzed and discussed before come out with a proper conclusion.

3.3 Activated Sludge Process

The non-linear Activated Sludge Process that has been selected and will be used for this research is taken from [18]. Activated sludge processes are biological that removes pollutant from the wastewater. The system comprise of aerator and settler as shown in Figure 3.2. The bioreactor (aerator) includes secondary clarifier to maintain the biomass in the system while producing high quality effluent. Part of settler output is recycled to allow the right concentration of microorganism the aerated tank.

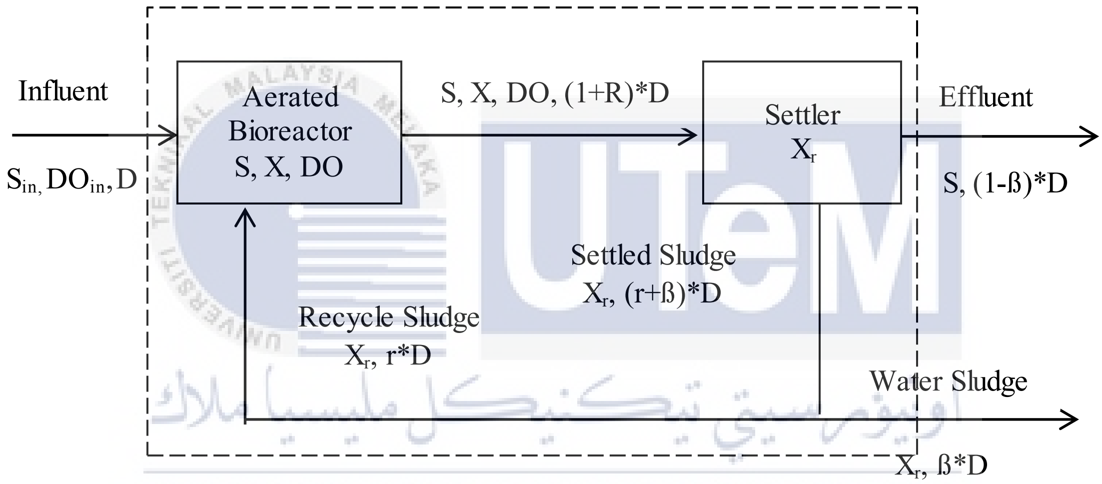


Figure 3.2: Block Diagram of Activated Sludge Process

The model was derived based on component mass balance equation which yields a set of non-linear differential equation given by (3.3.2) - (3.3.5).

$$\textit{Mass balance equation} = \textit{Input} - \textit{Output} \pm \textit{Reaction} \quad (3.3.1)$$

$$\dot{X}(t) = \mu(t)X(t) - D(t)(1+r)X(t) + rD(t)X_r(t) \quad (3.3.2)$$

$$\dot{S}(t) = -\frac{\mu(t)}{Y}X(t) - D(t)(1+r)S(t) + D(t)S_{in} \quad (3.3.3)$$

$$\dot{C}(t) = -\frac{K_0\mu(t)}{Y}X(t) - D(t)(1+r)C(t) + K_{La}(C_s - C(t) + D(t)C_{in} \qquad (3.3.4)$$

$$\dot{X}_r(t) = D(t)(1+r)X(t) + D(t)(\beta + r)X_r(t) \qquad (3.3.5)$$

$$\mu = \mu_{max}\frac{S}{K_s + S} \times \frac{C}{K_c + C} \qquad (3.3.6)$$

Where

$X(t) = Biomass\ (mgl^{-1})$

$S(t) = Substrate\ (mgl^{-1})$

$X_r(t) = Recycled\ biomass\ (mgl^{-1})$

$C(t) = Dissolved\ oxygen\ (mgl^{-1})$

$D = Dillution\ rate\ (h^{-1})$

$W = Air\ flow\ rate\ (m^3/h)$

$X(t)$,$S(t)$, $X_r(t)$ and $C(t)$ are the state variables which represents the concentration, biomass, substrate, dissolved oxygen and recycled biomass respectively. S_{in} and C_{in} represents concentrations substrate and dissolved oxygen of the fluent stream. Ratio of recycled and waste flow to the influent flow rate are given by r and β. Specific growth rate μ produced cell mass$Y$. K_0 is constant, while C_s and K_{La} is maximum dissolve oxygen concentration and oxygen mass transfer coefficient. Relationship between maximum growth rate to substrate and to dissolve oxygen coefficient is given by Monod equation as shown in (3.3.6). μ_{max} is the maximum specific growth rate, K_s is the affinity constant and K_c is the saturation constant.

For the purpose of this study it is assume that only two outputs are desired to be control, the substrate and dissolved oxygen. Another hypothesis is that, there is knowledge of parameter and constant value.

The following step is to linearize the nonlinear model. For simplification the state space in (3.3.7) were represent in (3.3.8). Linearization of the system represented in (3.3.10) was linearization around the operating point described by the data represented in Table 3.1, while Table 3.2 show the other parameters value.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.3.7}$$

$$\dot{Z} = f(Z, U) \qquad ; \ \delta Z = A\delta Z + B\delta U \tag{3.3.8}$$

$$\dot{Z} = \begin{bmatrix} \dot{X} \\ \dot{S} \\ \dot{C} \\ \dot{X}_r \end{bmatrix}, \quad Z = \begin{bmatrix} X \\ S \\ C \\ X_r \end{bmatrix}, \quad U = \begin{bmatrix} D \\ W \end{bmatrix} \tag{3.3.9}$$

$$Z = \begin{bmatrix} \frac{\delta \dot{X}(t)}{\delta X(t)} & \frac{\delta \dot{X}(t)}{\delta S(t)} & \frac{\delta \dot{X}(t)}{\delta C(t)} & \frac{\delta \dot{X}(t)}{\delta X_r(t)} \\ \frac{\delta \dot{S}(t)}{\delta X(t)} & \frac{\delta \dot{S}(t)}{\delta S(t)} & \frac{\delta \dot{S}(t)}{\delta C(t)} & \frac{\delta \dot{S}(t)}{\delta X_r(t)} \\ \frac{\delta \dot{C}(t)}{\delta X(t)} & \frac{\delta \dot{C}(t)}{\delta S(t)} & \frac{\delta \dot{C}(t)}{\delta C(t)} & \frac{\delta \dot{C}(t)}{\delta X_r(t)} \\ \frac{\delta \dot{X}_r(t)}{\delta X(t)} & \frac{\delta \dot{X}_r(t)}{\delta S(t)} & \frac{\delta \dot{X}_r(t)}{\delta C(t)} & \frac{\delta \dot{X}_r(t)}{\delta X_r(t)} \end{bmatrix} \delta Z + \begin{bmatrix} \frac{\delta \dot{X}(t)}{\delta X(t)} & \frac{\delta \dot{X}(t)}{\delta X(t)} \\ \frac{\delta \dot{X}(t)}{\delta X(t)} & \frac{\delta \dot{X}(t)}{\delta X(t)} \\ \frac{\delta \dot{X}(t)}{\delta X(t)} & \frac{\delta \dot{X}(t)}{\delta X(t)} \\ \frac{\delta \dot{X}(t)}{\delta X(t)} & \frac{\delta \dot{X}(t)}{\delta X(t)} \end{bmatrix} \delta U \tag{3.3.10}$$

Table 3.1: Initial condition value

Parameter	Value
$X(o)$	$215(mgl^{-1})$
$S(o)$	$55(mgl^{-1})$
$C(o)$	$6(mgl^{-1})$
$X_r(o)$	$400(mgl^{-1})$
$S_{in}(o)$	$200(mgl^{-1})$
$C_{in}(o)$	$0.5(mgl^{-1})$
D	$0.0825(h^{-1})$
W	$90(m^3/h)$

Table 3.2: Kinetic parameter value

$Y = 0.65$	$\alpha = 0.018$	$\mu_{max} = 0.5h^{-1}$
$r = 0.6$	$K_c = 2(mgl^{-1})$	$K_0 = 0.5$
$\beta = 0.2$	$K_s = 100(mgl^{-1})$	$C_s = 10(mgl^{-1})$

The results of linearization yields linearized matrices A, B, C and D given in (3.3.11). By using (3.3.12) the matrices manipulated into transfer functions matrix form which represent by (3.3.13) - (3.3.16).

$$\dot{Z} = \begin{bmatrix} -0.0990 & 0.1234 & 0.2897 & 0.0495 \\ -0.05077 & -0.3129 & -0.4457 & 0 \\ -0.02538 & -0.0943 & -0.1975 & 0 \\ -0.02538 & 0 & 0 & -0.066 \end{bmatrix} \begin{bmatrix} X \\ S \\ C \\ X_r \end{bmatrix} + \begin{bmatrix} -82.17 & 0 \\ 134 & 0 \\ -9.083 & 0.06994 \\ 8\times10^{-5} & 0 \end{bmatrix} \begin{bmatrix} D \\ W \end{bmatrix}$$

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ S \\ C \\ X_r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} D \\ W \end{bmatrix}$$

$$x(s) = (sI - A)^{-1}Bu(s) \qquad (3.3.11)$$

$$y(s) = [C(sI - A)^{-1}B + D]u(s) \qquad (3.3.12)$$

$$g_{11} = \frac{134(s+2.0049)(s+0.1864)(s+0.0117)}{(s+1.9956)(s+0.2573)(s+0.2014)(s+0.0076)} \qquad (3.3.13)$$

$$g_{12} = \frac{-0.0312(s+0.1863)(s+0.0117)}{(s+1.9956)(s+0.2573)(s+0.2014)(s+0.0076)} \qquad (3.3.14)$$

$$g_{21} = \frac{-9.0830(s+1.4447)(s+0.1942)(s+0.0048)}{(s+1.9956)(s+0.2573)(s+0.2014)(s+0.0076)} \qquad (3.3.15)$$

$$g_{22} = \frac{0.0699(s+2.802)(s+0.1993)(s+0.0074)}{(s+1.9956)(s+0.2573)(s+0.2014)(s+0.00076)} \qquad (3.3.16)$$

$$G(s) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \qquad (3.3.17)$$

In multivariable system, interaction effect are taken into consideration, hence RGA calculation is compulsory. RGA calculations are given by equation below and the calculation result is shown in (3.3.20) – (3.3.22). Since the result near to diagonal matrix, hence it can be conclude that manipulated variable 1 (MV1) controlled by control variable 1 (CV1) and manipulated variable 2 (MV2) controlled by control variable 2 (CV2).

$$\Lambda = \begin{bmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{bmatrix} \quad (3.3.18)$$

$$\lambda_{11} = \frac{1}{1-\frac{K_{21}K_{12}}{K_{22}K_{11}}} \quad (3.3.19)$$

$$K_{dc} = \begin{bmatrix} 742.2014 & -0.0861 \\ -15.4578 & 0.0367 \end{bmatrix} \quad (3.3.20)$$

$$\lambda_{11} = \frac{1}{1-\frac{(0.0861)(-15.4578)}{(742.201)(0.0367)}} = 1.05 \quad (3.3.21)$$

$$\Lambda = \begin{bmatrix} 1.05 & -0.05 \\ -0.05 & 1.05 \end{bmatrix} \quad (3.3.22)$$

3.4 Multivariable PID Tuning

This section will discuss the method used to tune the parameter of MPID controller. MPID controller is used due to consideration of multivariable system. Then, four types of MPID tuning have been selected which are Davison, Penttinen-Koivo, Maciejowski and Proposed Combined. These methods had been chosen due to its ability in dealing with interaction in simpler ways and it also required only step or frequency test. The aims are to obtain the substrate and dissolved oxygen concentrations at desired level.

3.4.1 Davison method

Davison is one of the MPID tuning methods and it is known to diagonalizing plant at low frequency. In Davison, only integral term is taken into consideration and the expression given by (3.4.1).

$$u(s) = K_i \frac{1}{s} e(s)\ , \quad K_i = \varepsilon G^{-1}(0) \tag{3.4.1}$$

where K_i is the integral feedback gain, $G(s)$ is the open loop transfer function matrix and ε as the scalar tuning parameter. Parameter tuning, ε Davison is adjusted in order to obtain a better closed loop system performance.

3.4.2 Penttinen-Koivo method

For the Penttinen-Koivo method, the plant is diagonalized at high frequency. Based on the controller given in (3.4.2), proportional and integral terms are taken into accounts. It also can be observed that the equation is an extension of Davison method.

$$u(s) = \left(K_p + K_i\right)\frac{1}{s}\ e(s) \tag{3.4.2}$$

$$K_i = \varepsilon G^{-1}(0) \tag{3.4.3}$$

$$K_p = (CB)^{-1}\rho \tag{3.4.4}$$

K_i and K_p is given by (34) and (35). C and B value can be obtained from output and input matrices in state space equations. In this method, two scalar parameter can be tune which are ε and ρ.

3.4.3 Maciejowski method

Maciejowski method also use the proportional and integral term of PID, it differ with Penttinen-Koivo method in the way of Maciejowski try to diagonalize the system near to bandwidth. The expression of the controller can be given by (3.4.5).

$$K = \left(K_p + K_i \frac{1}{s}\right) \tag{3.4.5}$$

$$K_p = \rho G^{-1}(j\omega_b) \tag{3.4.6}$$

$$K_i = \varepsilon G^{-1}(j\omega_b) \tag{3.4.7}$$

where K_p is the proportional gain, K_i is an integral gain, ω_b is the bandwidth frequency. The $G^{-1}(j\omega_b)$ results in a complex number, with that a real approximation given in (3.4.8) are necessary. In this method, three parameters can be tune which are ε, ρ and $j\omega_b$. But for the purposed of the study only scalar tuning will be tuned.

$$J(K,\theta) = \left[G(j\omega_b)K - e^{j\theta}\right]^T\left[G(j\omega_b)K - e^{j\theta}\right], \quad \theta = diag(\theta_i, \ldots, \theta_n) \tag{3.4.8}$$

3.4.4 Proposed method

The Proposed Combined method is introduced by [18]. In this method it uses the criteria of Maciejowski in diagonalizing system near bandwidth frequency. Proposed Combined method is being developed to reduce the difficulties in finding the suitable bandwidth frequency. Nevertheless, it is shows similar behavior with Maciejowski. The expression of Proposed Combined controller is shown in (3.4.9).

$$u(s) = e(s)(\rho K + \varepsilon K)\frac{1}{s} \tag{3.4.9}$$

$$K = [\alpha G(0) + (1-\alpha)CB]^{-1} \tag{3.4.10}$$

where K is given in (3.4.10) and α is a constant value between [0, 1]. By that, three scalar tuning will be tuned in this method which are α, ε and ρ.

3.5 Optimization Technique

Optimization technique can be considered as the main part in this study. There are three of optimization techniques that have been selected for the purpose of the studies which are PSO, GA and BA. PSO and GA are the most popular optimization technique available in this moment while BA is the latest optimization technique. All of them are used to tune the scalar parameter tuning of MPID that has been discussed earlier.

3.5.1 Bat Algorithm

The BA is a novel metaheuristic swarm intelligence optimization method developed for global numerical optimization. This algorithm is inspired by the social behavior of bats and the phenomenon of echolocation to sense distance. The idea rules of BA originally presented by Yang et al. in 2010. The approximate rules of BA can be categorized into three. First, all the bats use echolocation to sense distance, and they also know the difference between food or prey and background barrier in some unknown way. Second, bats fly randomly with velocity at position with a fixed frequency, varying wavelength and the loudness to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission depending on the proximity of their target. Lastly, the third rule is although loudness can vary in many ways, we assume that the loudness varies from a large to a minimum constant value.

The overall flow of BA is shown in Figure 3.3. Based on that, it can be seen that the basic or standard BA being used and further explanation will also discuss.

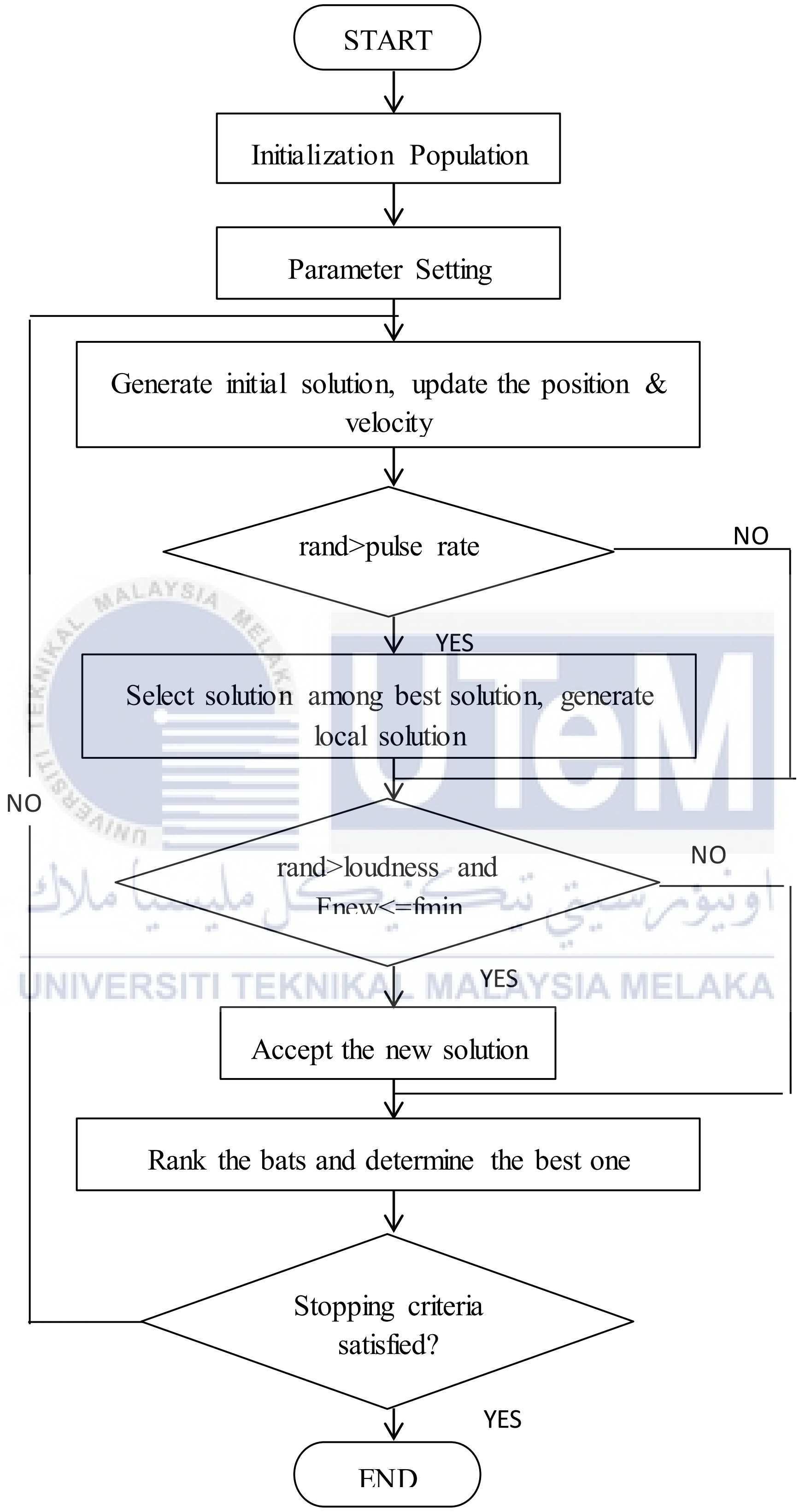


Figure 3.3: BA Flow Chart

1. Initialize the initial population of bats
2. Setting the parameters; pulse frequency, pulse rate, loudness
3. Generate the position and velocity for initial bats
4. If result of rand>pulse rate is yes, select the solution among best solution
5. If result of rand>loudness and Fnew<=fmin is yes, accept new solution
6. Rank the bats and determine the best one
7. Go back to step 3 and repeat all the step until stopping criteria is met

The optimization of BA algorithm is started by initialize the number of population, number of iteration and search range. Then, a certain parameters need to be setting which are pulse frequency, loudness and pulse rate. All the related initializing parameters that will be used in this algorithm are tabulated in Table 3.3. For the process on finding the initial solution, the initialization will be based on the random value with the range search is set in earlier stage by using the equation given in (3.5.1). This process will be repeated depending on the number population size of bats. The fitness of each population also will be determined during this process and the initial best solution is obtained from the minimum fitness of each population. The minimum fitness also will be indicated as the initial minimum frequency, fmin.

$$initialization = range_{min} + (range_{max} - range_{min}) \times random\ no. \quad (3.5.1)$$

Table 3.3: Parameter initialization in BA

Initialization	
Population size = 50	No. of iteration = 100
Search range = 0-10	
BA initialization	
Loudness = 0.95	Max. frequency = 2
Pulse rate = 0.9	Min. frequency = 0

Next is the process of generating the position and velocity for initial solution. In order to determine the position and velocity, the frequency of the bats, Fnew must be evaluate first with the random number in the range of [0, 1] by using the equation given by (3.5.2). The algorithm continued with finding the new velocity and position for each particle in order to improve the search for solution. The new velocity will be calculated by using the equation given in (3.5.3). However, the new particle position is obtained by summing the current position with the new velocity value which express in (3.5.4).

$$Q_{new} = Q_{min} + (Q_{min} - Q_{max}) \times random\ no. \tag{3.5.2}$$

$$V_{i,k+1} = V_{i,k} + \left(P_{i,k} - X_{i,k}\right) \times Q_{new} \tag{3.5.3}$$

$$X_{i,k+1} = P_{i,k} + V_{i,k+1} \tag{3.5.4}$$

where

Q_{new} : New frequency of bats

V_i^k : Velocity of the i^{th} individual at iteration k

X_i^k : Best position of the i^{th} individual at iteration k

$P_{i,k}$: Position of the i^{th} individual at iteration k

The comparing process between random number and pulse rate must be done in order to select the solution of bats among the best solution and local solution also is generate. If the result comparison of random is larger than pulse rate is true, the solution of bats among the best solution is selected. Otherwise, the step will straight go to next comparison process. If the result comparison between random is larger than loudness and Fnew is less than fmin is true, the new solution is selected. Then, the bat will rank and the best one is determined.

Lastly, the algorithm will continue the iteration until the stopping criteria is met. The performance index of BA is determined by the value of fitness function. The better the fitness function, the better the performance index with smaller error (ITSE value).

3.5.2 Genetic Algorithm

Genetic algorithm (GA) introduced by John Holland in 1975 is an intelligent techniques which inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. GA also is a local search technique that used to search approximate optimal solutions for an optimization and search problems. It starts from a population of completely random individuals and occurs in generations where each generation of GA, the fitness of the whole population is evaluated. Then, multiple individuals are stochastically selected from the current population based on their fitness and they are modified to form a new population. For the next iteration of the algorithm, the new population obtained is then used.

There are two popular selection mechanisms in GA that used to help in the reproduction stages, roulette wheel and tournament wheel. For this project, roulette wheel has been selected to be used as selection mechanism. The concept of roulette wheel selection can be expressed by imagining a wheel where each chromosome occupies an area that is related to its fitness value. A fixed marker will determine which chromosomes will be selected to reproduce into the mating pool when a spinning wheel stops. However, this selection mechanism needs more numerical computations.

The overall flow of GA is shown in Figure 3.4. Based on that, it can be seen that the basic or standard GA being used and further explanation will also discuss.

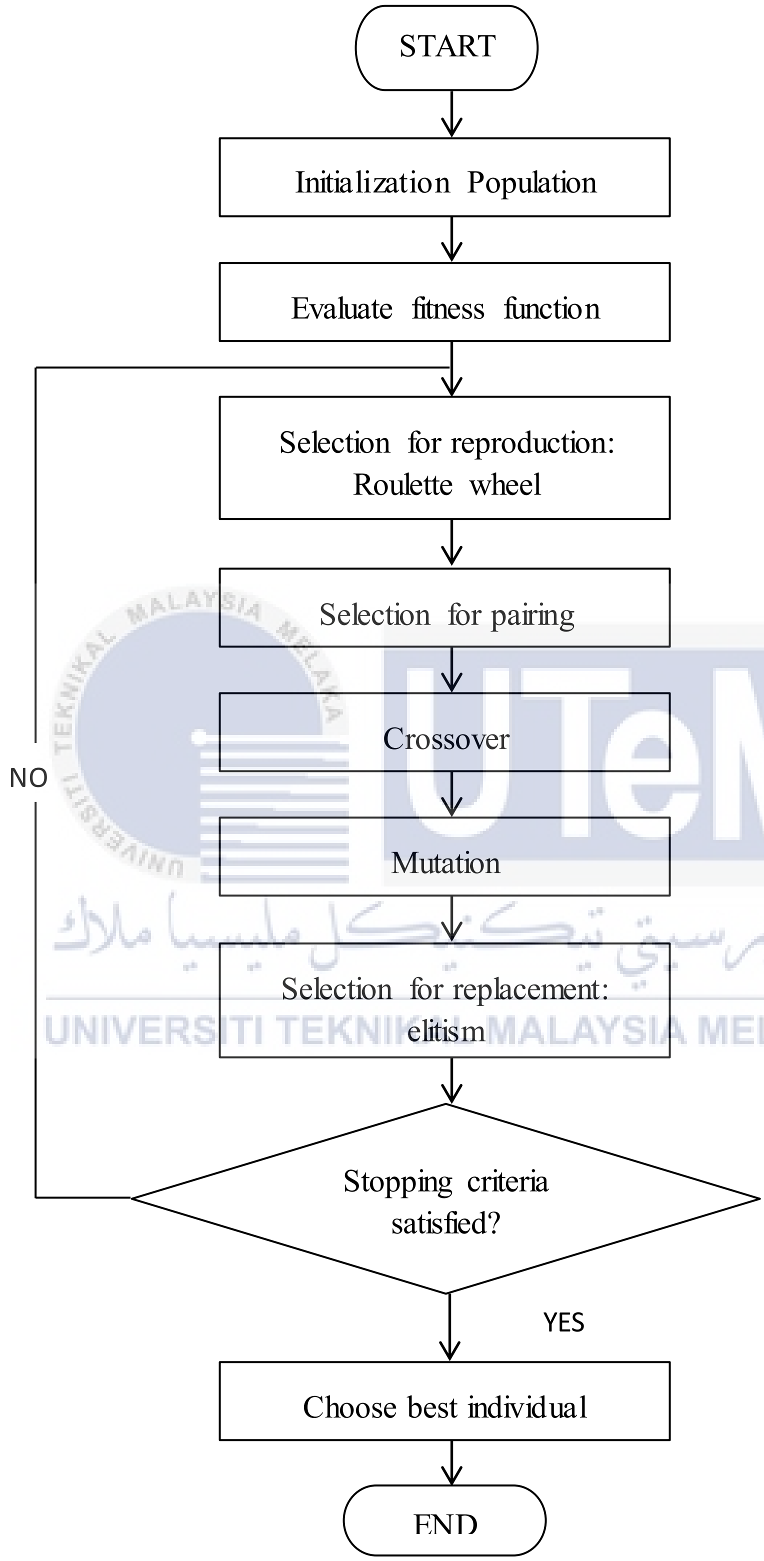


Figure 3.4: GA Flow Chart

1. Initialize a group of random population with set of different chromosomes
2. Obtain the fitness function for each random chromosomes
3. Select half from the total population to be produce
4. Randomly assign each of the chromosomes as parent 1 and parent 2
5. By using one cut point method, cut both parents into random cut-point and exchange between cut-point of parents 1 and parents 2
6. Do mutation by flipped a random bit
7. Evaluate the new population fitness function
8. Combine the new population with the previous population that selected to reproduce and record the best performance index
9. Go back to step 2 and repeat all the step until stopping criteria is met

A binary type of GA algorithm is being used for this study because binary are easier to alter when it comes to crossover and mutation process. Since the parameter tuning will be parameter tuning will be the real number, hence conversion from binary to real and real to binary is compulsory. First of all, it is required to know the bit number that is required based the maximum and minimum range of search which defined by the user. Since, it is strongly belief that the values in four decimal places, hence equation (3.5.5) is use. After the number of bit are known, the chromosomes can start to be initializing by assigning randomly 0 to 1 according to the number of bit calculated earlier. The processes are repeated until one complete population is generated. If there is parameter need to be tune the total bit number will be summation of bit number of gene 1 and gene 2. The other important thing of genetic algorithm is that the number of population must be in an even number, so that the reproduction process will be easier. All data in GA initialization is given in Table 3.4.

Table 3.4: Parameter initialization in GA

Initialization	
No. of particles = 50	No. of counter = 10
Search range = 0-10	No. of iteration = 100
GA initialization	
$P_c = 0.6$	$P_m = 0.1$

After the initialization process is done, it is required to calculate performance index of each initial chromosome. A binary to real conversion in (3.5.6) is required as fitness evaluation in real number. Conversion from real to binary again required after the fitness value is obtained.

$$range = (range_{max} - range_{min}) \times 10^4 \tag{3.5.5}$$

$$r = x_0 + decimal\ (binary\ string) \times \frac{x_1 - x_0}{2^n - 1} \tag{3.5.6}$$

$$decimal = 2^{n-1} \times gene\ value\ (binary\ string) \tag{3.5.7}$$

The selection process will then takes places. Roulette Wheel selection is used as selection technique in this study. Each chromosome in the population will allocate in section of roulette wheel and the area is depending of the performance index. The bigger the performances index value the bigger its section, hence the bigger it's change to be selected. The probability can be calculated using equation in (3.5.8). With the cumulative probability is equal to 1. Roulette wheel will then turn by mean random probability assignments. The process repeated until number of assigned turn complete. The number of assigned turn in this study would be half from the population size.

$$selection\ probability, p = \frac{fitness\ value(i)}{total\ fitness} \tag{3.5.8}$$

$$cummulative\ probability, c = \int_0^t p_i \tag{3.5.9}$$

The process will then continuous with crossover stages. Based on the selection, randomly assigned each of the chromosomes as parents 1 and parents 2. Pair the parent 1 and parent 2 randomly. By using One-point method crossover is performed with Pc=0.6. A new offspring's will generate after process is done with the probabilities of it become completely better compared to parents or just the same. By using mutation rate, Pm=0.1, the mutation process is initialize. After all three main stages are complete, again the fitness functions of the new offspring's need to be done.

The offspring's will then combine with the parents to form a new and better population. In this stages the best performance index will be memorize as the reference. In GA the fitness function is not the same with the performance index. The fitness expression is given in (3.5.10) and can be summarized that, the better the fitness function, the bigger the performance index value with large error.

$$fitness\ value = \frac{1}{performance\ index} \tag{3.5.10}$$

$$performance\ index, ITSE = \int_0^t te(t)^2\, dt \tag{3.5.11}$$

3.5.3 Particle Swarm Optimization (PSO)

The first optimization that will be discussed is PSO. Particle Swarm Optimization is an optimization technique which is its working principle based on social sharing of a swarm such as fish schooling or bird flocking. The information sharing can be happen while searching for food where in PSO each single solution is a "bird" in the search space which referred as a "particle". The swarm is modeled for optimization of nonlinear functions in multi-dimensional space where it follows the principles of natural selection and search algorithm. The particles have the memory of their own best position and knowledge of the global best. The swarms of particle communicate through adjustment of position and velocity. In order to give a better understanding is developing PSO algorithm a flow chart is presented in Figure 3.5. Based on that, it can be seen that the basic or standard PSO being used. Further explanation will also discuss.

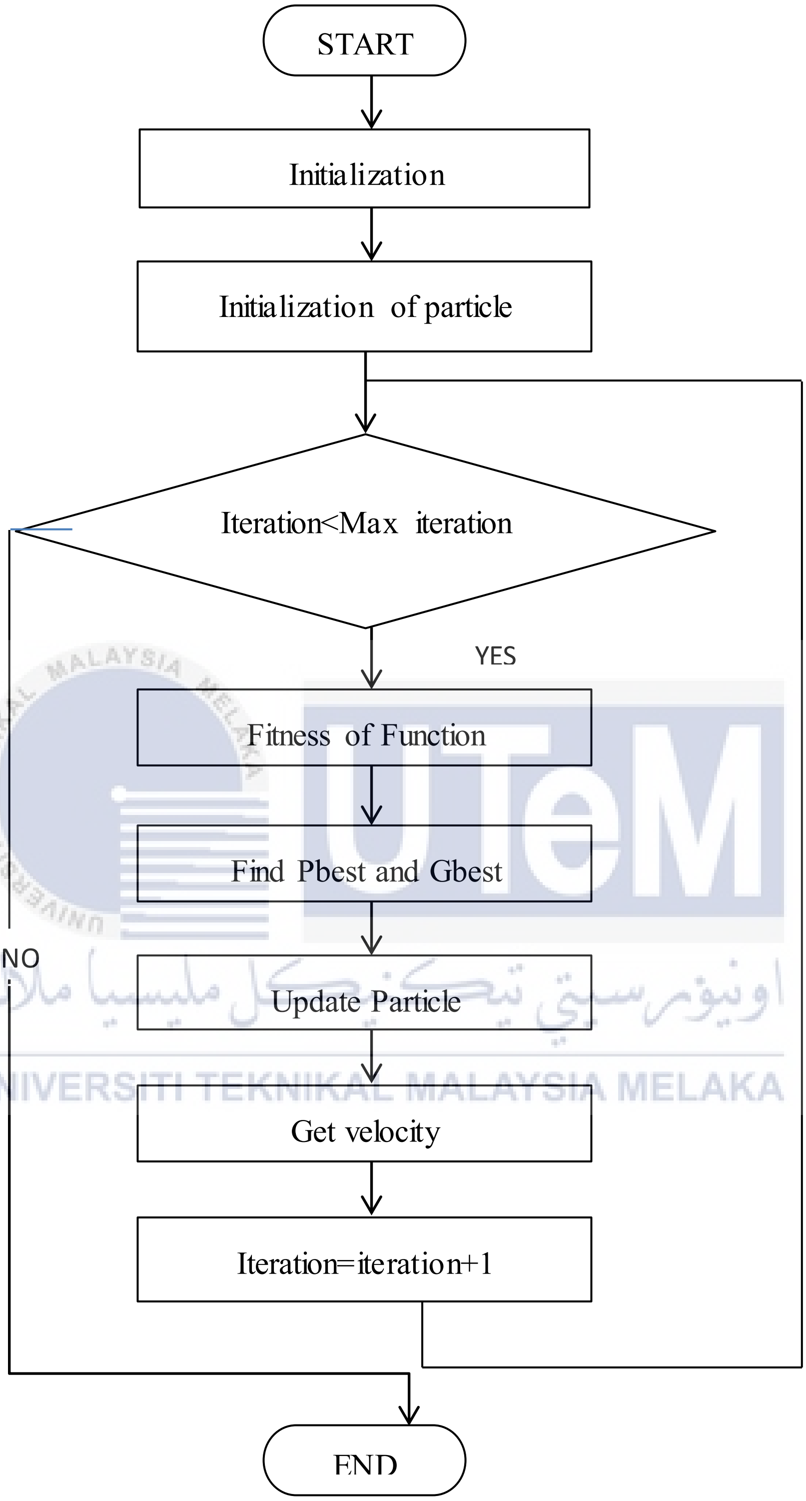


Figure 3.5: PSO Flow Chart

1. Initialize a group of particles including the random positions, velocities and acceleration of particles
2. Evaluate the fitness of each particle
3. Compare the individual fitness of each particle with previous Pbest. If it is better, update as new Pbest
4. Compare the individual fitness of each particle with previous Gbest. If it is better, update as new Gbest
5. Update velocity and position for each particle
6. Go back to step 2 and repeat all the step until stopping criteria is met

The process of PSO to optimize the parameter tuning of MPID will start with the initializing a group of particles including the random positions, velocities and acceleration of particles. A group of particles will be initializing on the number of tuning parameter. The initialization will be based on the random values with the range search is set in earlier stage and the equation are given by (3.5.12), where random number are value from 0 to 1. All the related initializing parameters that will be used in the PSO are given in Table 3.3.

$$initialization = range_{min} + (range_{max} - range_{min}) \times random\ no. \quad (3.5.12)$$

Then, PSO will start its work in finding parameters that will give optimum performance. After checking the number of counter and iteration, PSO algorithm will start its looping to find the particle best (Pbest) in the iteration by evaluating the fitness function. The algorithm will be continued with comparing the individual fitness of each particle with the previous Pbest. If the result is better, the new Pbest will be update which is means only the particle with the best performance index will be selected as the Pbest.

Next is to find the global best (Gbest). In order to determine the Gbest, the Pbest will compare with previous Gbest. The new Gbest will be updated if the result is better, if not the previous Gbest will remain. The algorithm continued with finding the new velocity and position for each particle in order to improve the search for solution. The new velocity will be calculated by using the equation given in (3.5.14), with an inertia weight, ω was added to the velocity equation. However, the new particle position is obtained by summing the current position with the new velocity value which express in (3.5.15). The algorithm will continue the iteration until the stopping criteria is met, which is the maximum number of iteration. Lastly, as a quick conclusion for PSO algorithm, the better the fitness function, the better the performance index with smaller error (ITSE).

Table 3.5: Parameter initialization in PSO

Initialization	
No. of particles = 50	No. of counter = 10
Search range = 0-10	No. of iteration = 100
Velocity initialization	
$c_1, c_2 = 2$	Max. weight = 0.9
Max. velocity $= \pi/100$	Min weight = 0.4

$$\omega_k = \omega_{max} - ((iteration * (\omega_{max} - \omega_{min}) / \max iteration) \quad (3.5.13)$$

$$V_{i,k+1} = \omega_k V_{i,k} + c_1 r_1 \left(P_{pbest,i,k} - X_{i,k}\right) + c_2 r_2 \left(P_{gbest,i,k} - X_{i,k}\right) \quad (3.5.14)$$

$$X_{i,k+1} = X_{i,k} + V_{i,k+1} \quad (3.5.15)$$

where

V_i^k : Velocity of the i^{th} individual at iteration k

ω_k : Inertia weight at iteration k

r_1 and r_2 : uniform random number of [0,1]

c_1 and c_2 : uniform random number of [0,1]

X_i^k : Position of the i^{th} individual at iteration k

$P_{pbest,i,k}$: Best position of the i^{th} individual at iteration k

$P_{gbest,i,k}$: Best position of the group at iteration k

3.6 Objective Function

The objective function is to evaluate the performance index of the system where it can be found in various form such as time-domain specifications, frequency domain specifications and time-integral performance. It is also representing system criteria that desired by the user. In this study, no specific criteria of the system as long as it give the optimum value of the objective function. Thus, the smaller the error or the objectives function, the better the system or performance index.

Time- integral performance or more specifically Integral Time Square Error (ITSE) has been chosen as the objective function. ITSE expression is given in (3.6.1) and from the equation, ITSE only required system error and the answer represents the area of the output and desired output. This selection is due to ITSE behavior that provides a better dynamic performance with good settling time.

$$ITSE = \int_0^t \left(e_i(t)\right)^2 t\, dt \tag{3.6.1}$$

$$e(t) = y(t) - r(t) \tag{3.6.2}$$

Where

$e(t)$ is an error given in (54)

$y(t)$ is output of system

$r(t)$ is desired output

3.7 Simulation

In this study, simulations need to be carried out to achieve the objective of this research. The simulation of using PSO, GA and BA to tune MPID controller for ASP will be simulate by using MATLAB/SIMULINK software and the MPID control methods as previously mentioned will be implemented to ASP transfer function. Furthermore, PSO, GA and BA will be applied to tune the MPID and the results from this simulation will be compared. The simulation for this research is divided into two parts. The first part is about analyzing optimization techniques using linear model. The second part is about using the nonlinear model to analyze the optimization techniques.

There are some criteria have been considered in simulating the algorithm. All selected optimization used in this study are known as stochastic algorithm, where different result obtained for every time the algorithm is executed, even though the same initial point is used. Hence, each optimization technique was executed for 20 times to make fair observation for further analysis. The result was selected based on the execution that gives the best performances index and this process called constant parameter initialization for optimization technique.

The other criteria that must be take care when simulating the algorithm is the stopping criteria. A repeated process will be done in optimization technique to obtain an optimum result. Therefore, in order for the system knows that it has found the best result; the stopping criteria should be introduced. For this study, the stopping criteria used is when the maximum iteration is reached. The repetition process of the algorithm will stopped when the maximum number of iteration is achieved.

This study is using multiple inputs and multiples output system. Therefore, there is four transfer function matrix to be consider for system with two input two output. Within that only the output transfer function, the other two shows a transfer function of an interaction. Hence, for the objective function purpose the ITSE will be summation between two closed loop performances $(ITSE = ITSE_{11} + ITSE_{22})$.

The result of the best performance index will come with the value of the scalar tuning, depending which MPID are use. The parameter tuning will be replaced to the non-linear system (Figure 3.6) in order to observe the system performance. To execute the nonlinear model, an m-file is required to be executed first. The system is an offline tuning.

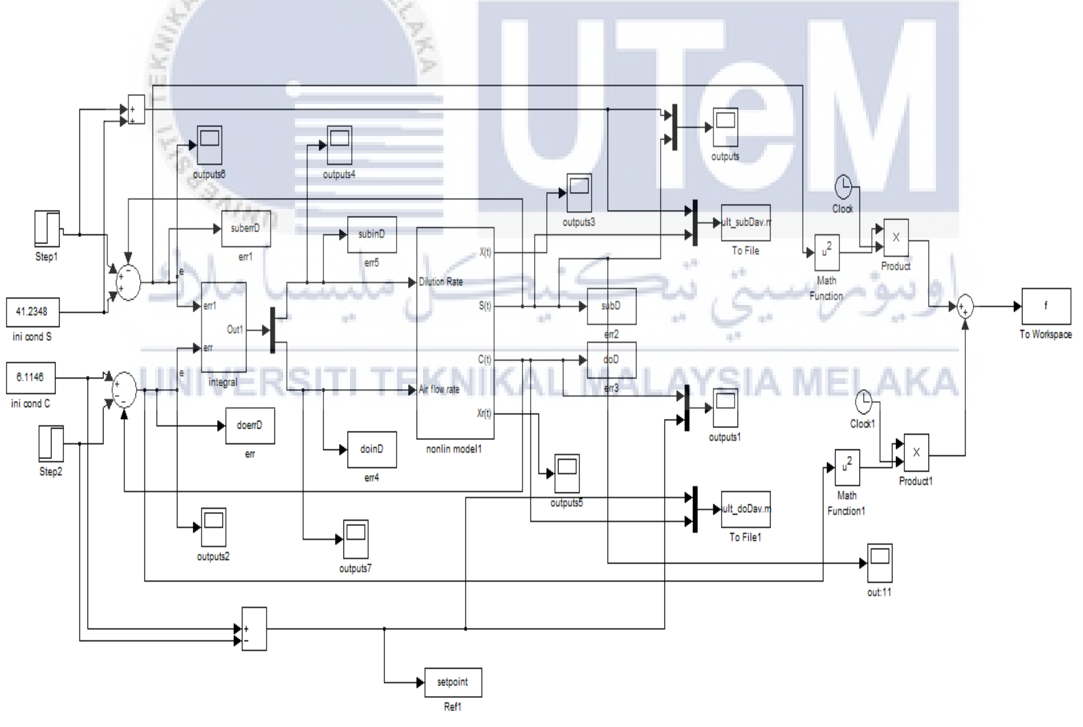


Figure 3.6: Nonlinear Activated Sludge System

CHAPTER 4

RESULT AND DISCUSSIONS

4.1 Introduction

The aim of this study is to show the performance of the MPID in controlling the concentration of substrate and dissolved oxygen of ASP. This chapter briefly discuss on the results and discussions of the simulation including the simulation of open loop response of ASP, initialization parameter to be used for all optimization technique, comparison result between optimization used and the performances of system by using linear system and nonlinear system. The system performances that being evaluated are in term of transient response which are rise time, settling time and percentage overshoot have been determined for each method. Later, the best MPID with the best optimization technique will be selected for activated sludge process.

4.2 Open Loop Response

In this subchapter, it shows the open loop response of the system. Figure 4.1 shows the open loop response with step input, while Figure 4.2 shows open loop system Bode plot. The data performance of open loop system based on transient response is shown in Table 4.1 and Figure 4.2. This result will be used as a reference on how the system can be improves with additional controller and proper tuning.

From Figure 4.2 and Table 4.1, the simulation has been compared with four selected MPID tuning methods for every system in ASP (g_{11}, g_{12}, g_{21}, g_{22}) but the only g_{11} and g_{22} of the system are given attention because they are indicated the desired outputs of plant which are concentration of substrate, S and dissolved oxygen, DO. While, the g_{12} and g_{21} system also can be assumed as neglected parts because of the interaction of them are too small.

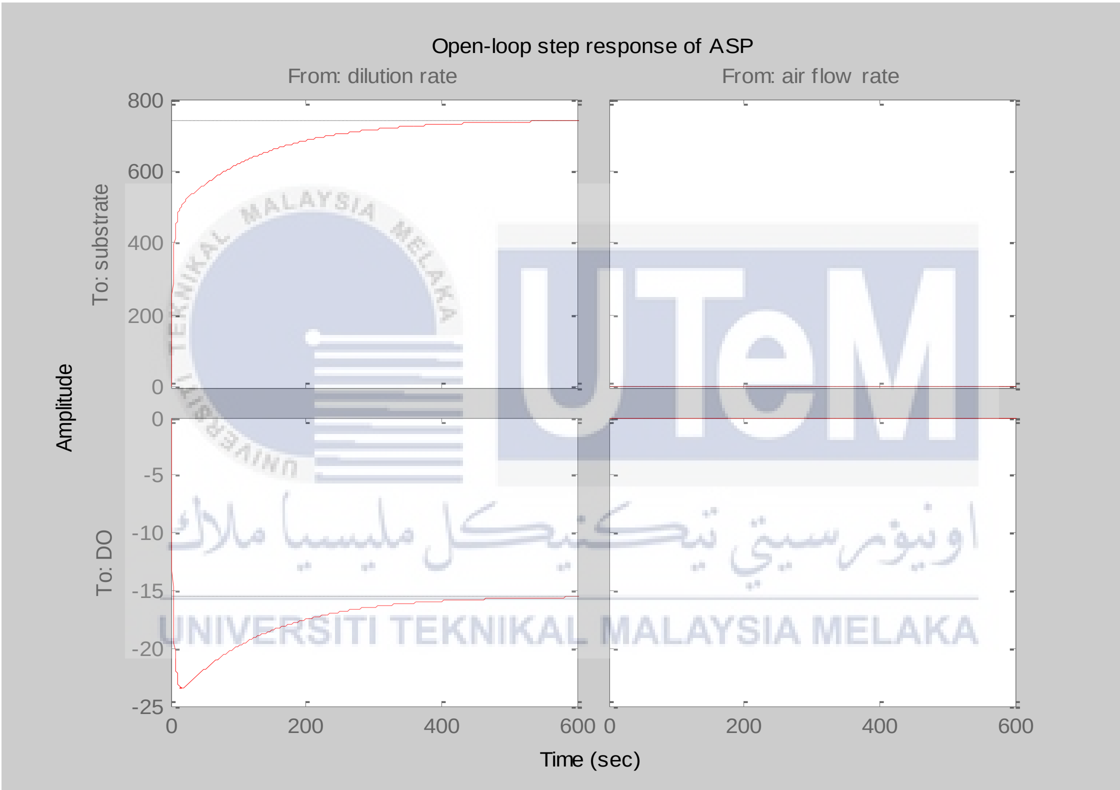


Figure 4.1: Open loop step response of ASP

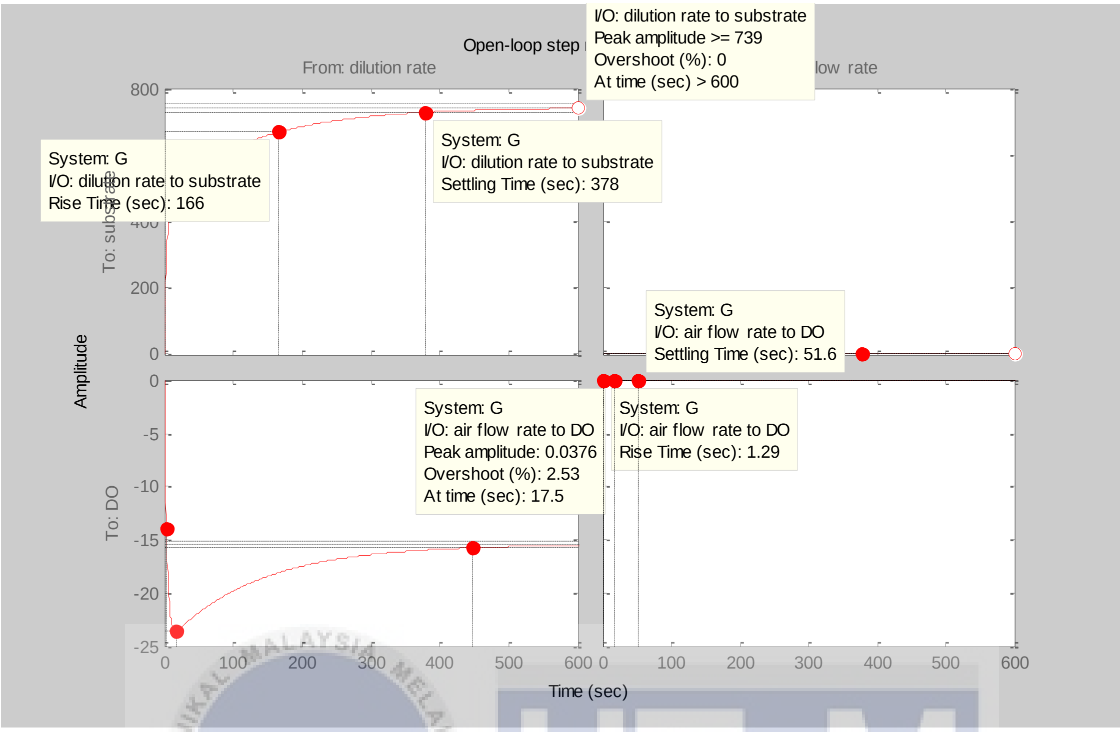


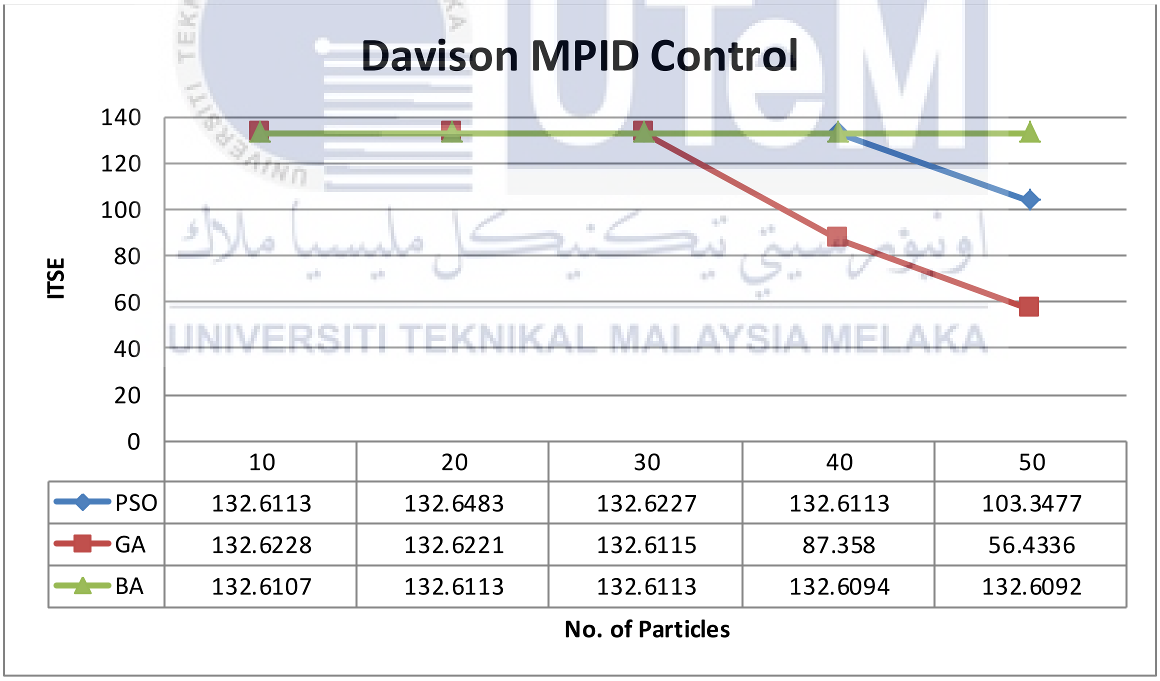
Figure 4.2: Open loop step response of ASP with system performance

Table 4.1: Open loop system performance data

Open Loop System		
Output	Substrate	DO
Settling Time, Ts (s)	378	51.6
Rise Time, Tr (s)	166	1.29
Overshoot, OS%	0	2.53

4.3 MPID Parameters Initialization and Selection

The initialization parameters of MPID must be done before simulation of optimization technique being simulate to get the best result of performance index. The purpose of this stage is in order to get a have a good performance of system by choosing the best initial parameters to be used for all selected optimization techniques. The results from simulating the PSO, GA and BA algorithm can be referred in Appendix B. These simulations were done for all four MPID methods, Davison, Penttinen-Koivo, Maciejowski and Proposed method. The simulations for choosing the final initialize parameters like number of particles, upper boundary and numbers of iteration are shown from Figure 4.3 until Figure 4.5.



	10	20	30	40	50
PSO	132.6113	132.6483	132.6227	132.6113	103.3477
GA	132.6228	132.6221	132.6115	87.358	56.4336
BA	132.6107	132.6113	132.6113	132.6094	132.6092

Figure 4.3: Performance Index vs. No. of Particles

Figure 4.3 shows that the ITSE obtained from varies of the number of particles for Davison method. Then, the chosen value to be the parameter initialization is 50, where that value can give better value of ITSE among others for all optimizations. This value is being selected for all MPID control method to have a consistent parameter value. The consistent parameter value makes sure that the comparison made between all MPID controllers is fair.

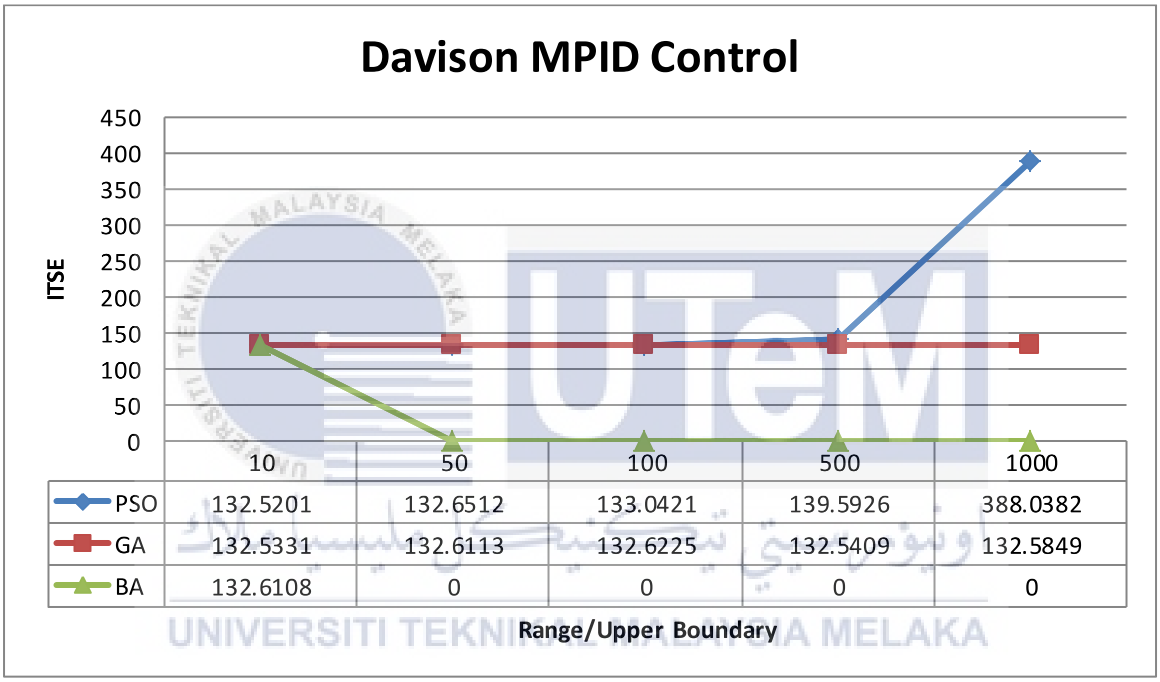


Figure 4.4: Performance Index vs. Range

From the results simulation of Davison method through all optimization technique, the chosen value to be selected as parameter initialization of range is 10. Although the BA optimization gives the best value of ITSE when the number of range is being increased, the results for PSO and GA must be considered before the final value is selected because they will give larger amount of ITSE compared to BA when the range of the system become larger. So, the minimum value of range is picked to compensate for all optimization and secure the system to be obtained the best performance index in term of ITSE.

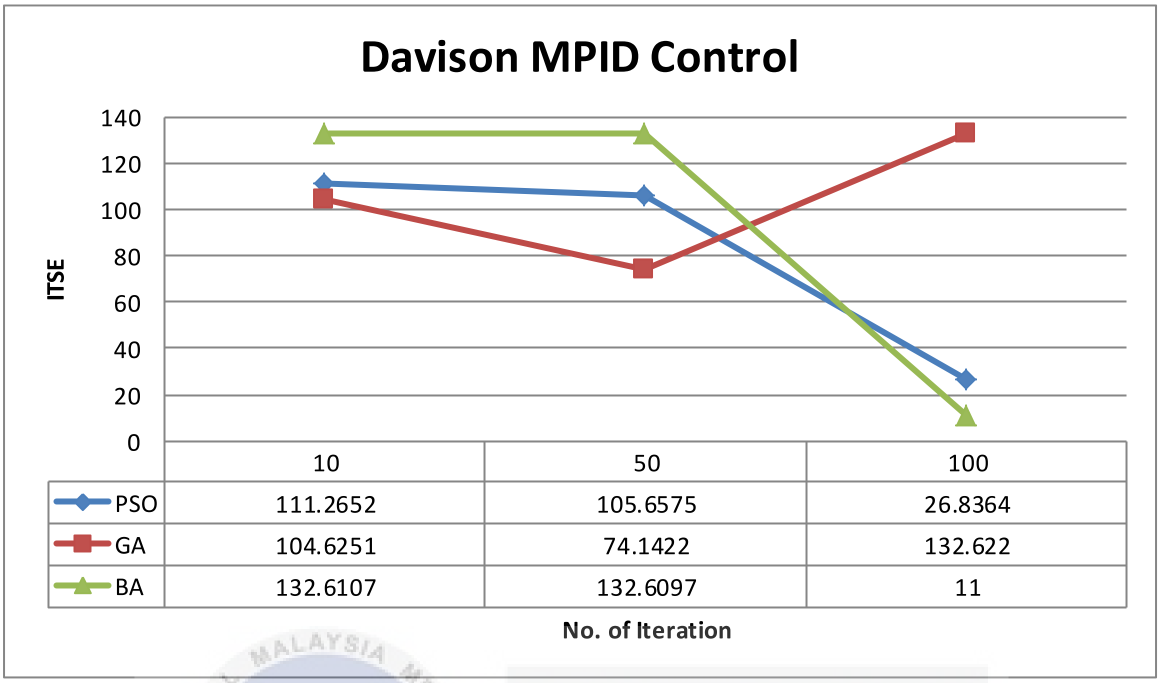


Figure 4.5: Performances index vs. No. of Iteration

Number of iteration is important in order to state the number for the system to continue repeating a process with the aim of approaching a desired goal or result. Therefore, the chosen number of iterations for all MPID method control is 100. The selected is based on the number of iteration that will give the lower result of ITSE. From Figure 4.5, PSO and BA will give the lower result of ITSE when the numbers of iteration become higher. However, the GA algorithm gives the higher value of ITSE. Then, a hundred of iterations are seemed equivalent and suitable to be selected and used for this study. All the final selected parameters initialization are tabulated in a Table 4.2 and they are used for all the MPID method for each optimization.

Table 4.2: Parameter initialization for PSO, GA and BA algorithm

Initialization		
No. of particles = 50	No. of iteration = 100	Search range = 0-10

4.4 Linear System of Activated Sludge Process

4.4.1 Result of MPID Tuning using BA

This subchapter will explain about the system performances of MPID control tuning by using bat algorithm. The results of this simulation are presented in a waveform for each types of control tuning as shown in Figure 4.6. Its data are tabulated in Table 4.3 and Figure 4.7 in a form of bar chart. The simulation was done by using the selected parameter from constant scalar parameter initialization for BA.

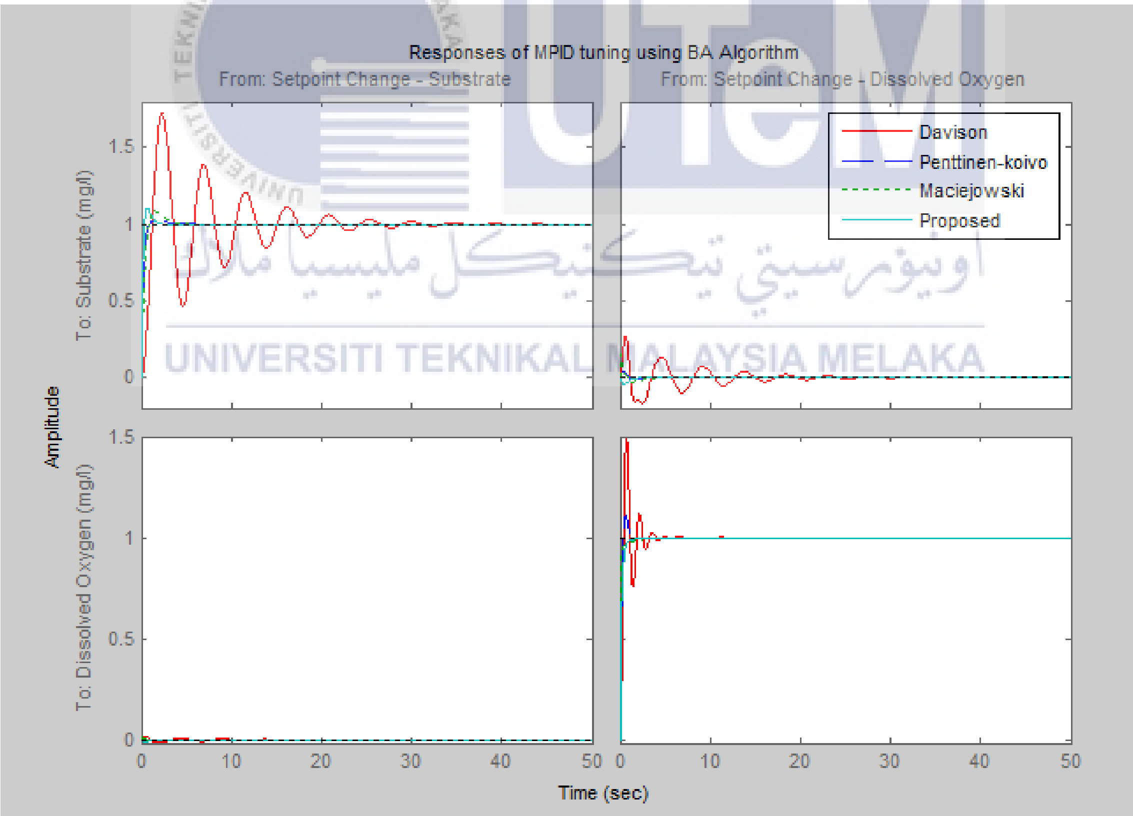


Figure 4.6: System Performances of MPID Control Tuning using BA

From the BA output result, the system performances of bat algorithm are quite similar to the PSO and GA algorithm and it gives the better performances compared to the open loop system. Each MPID control methods are being compared by transient response in terms of rise time, settling time and percentage of overshoot from to make an analysis on which methods will the best performances for the ASP system.

Table 4.3: MPID tuning parameter data using BA

Optimization Type	BA							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	132.6092		0.0075		1.5119		9.8716	
Epsilon, ε	10.2245		10.0027		7.9816		9.9900	
Rho, ρ	-		4.1857		9.9988		6.0562	
Alpha, α	-		-		-		0	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	0.814	0.274	0.476	0.285	0.596	0.144	0.249	0.415
Settling Time, T_S	28.1	3.75	2.10	1.15	3.48	1.40	1.65	1.01
Overshoot, OS%	73.0	48.4	2.4	10.6	8.66	1.36e-6	11.1	6.2e-6

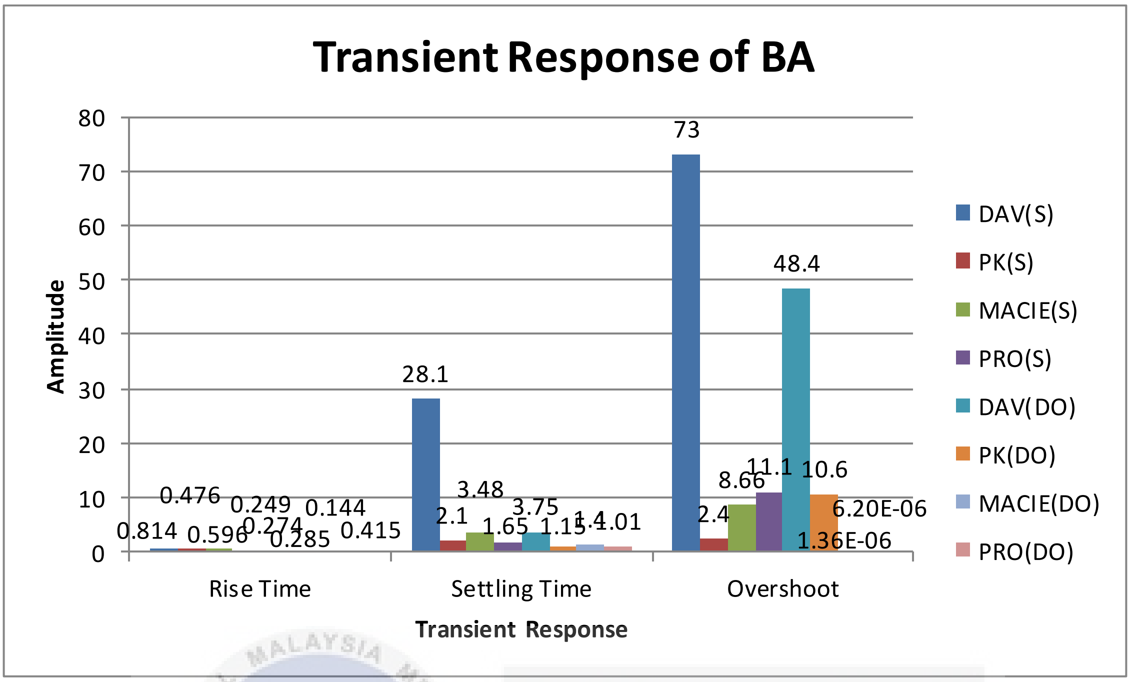


Figure 4.7: Transient Response between MPID Control Method of BA

Based on the time response data, it can be seen clearly that Davison on BA algorithm also give the worst response with the longer settling time and rise time. The reason on it is due to the missing proportional term in its equation where it only used the integral term. The advantages of having proportional term in a system are where it can help to reduce the rise time and settling time. The Davison also give the worst result in terms of overshoot and fitness function compared to the others.

While, the other three methods control tuning of BA algorithm, Penttinen-Koivo, Maciejowski and Proposed method shows the similar results. The similarity of the result is because of the same property of equation that they are used, which is proportional and integral term. However, the Proposed method give the better performance compared to Maciejowski even though they are having similarities in the expression. This is because the complexity in finding suitable bandwidth in Maciejowski method has been reduces by the Proposed method.

4.4.2 Result of MPID Tuning using GA

This subchapter will explain about the system performances of MPID control tuning by using genetic algorithm. The results of this simulation are presented in a waveform for each types of control tuning as shown in Figure 4.8. Its data are tabulated in Table 4.4 and Figure 4.9 in a form of bar chart. The simulation was done by using the selected parameter from constant scalar parameter initialization for GA.

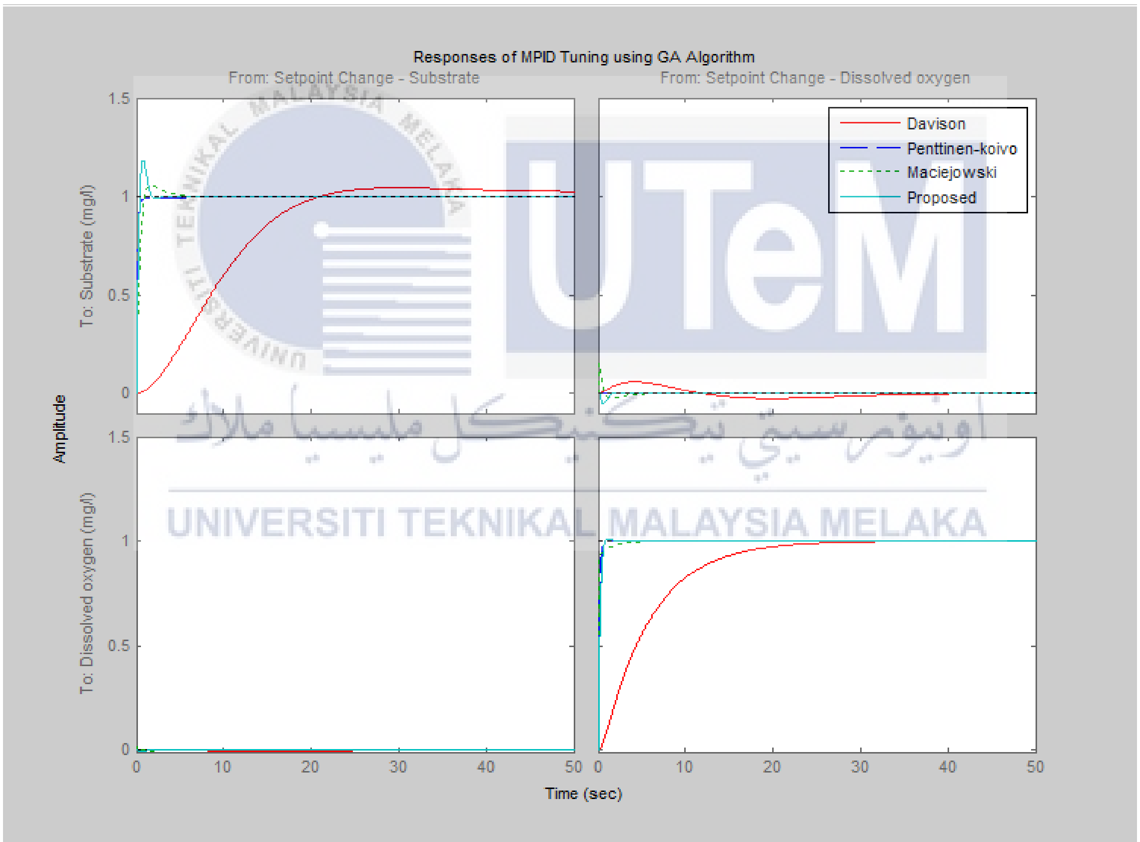


Figure 4.8: System Performances of MPID Control Tuning using GA

From the GA output result, the system performances of genetic algorithm are quite similar to the PSO algorithm and it gives the better performances compared to the open loop system. Each MPID control methods are being compared by transient response in terms of rise time, settling time and percentage of overshoot from to make an analysis on which methods will the best performances for the ASP system.

Table 4.4: MPID tuning parameter data using GA

Optimization Type	GA							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	26.7134		0.0104		17.8869		12.9856	
Epsilon, ε	0.1666		7.5632		6.0449		9.9178	
Rho, ρ	-		7.5609		9.9784		4.2566	
Alpha, α	-		-		-		0.0196	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	13.4	12.0	0.302	0.291	0.655	0.154	0.305	0.446
Settling Time, T_S	69.8	21.7	0.619	0.517	3.77	2.11	1.62	0.684
Overshoot, OS%	4.42	0	0.053	0.29e-3	5.95	1.70e-6	18.5	1.10

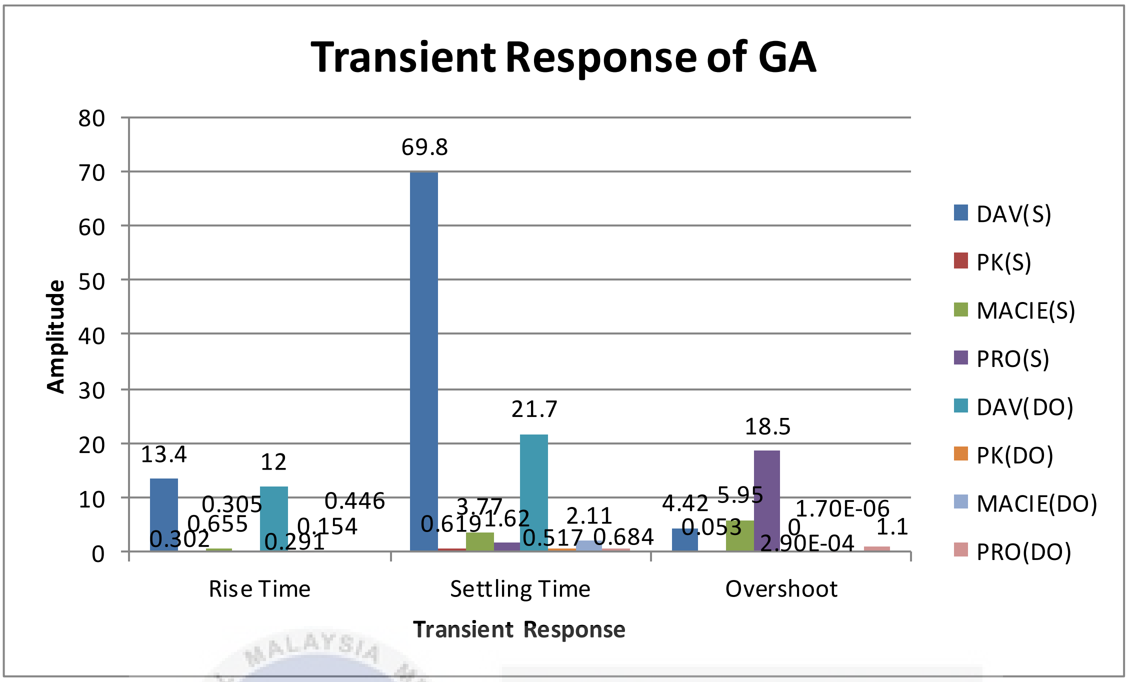


Figure 4.9: Transient Response between MPID Control Method of GA

Based on the time response data, it can be seen clearly that Davison on GA algorithm also give the worst response with the longer settling time and rise time. The reason on it is due to the missing proportional term in its equation where it only used the integral term. The advantages of having proportional term in a system are where it can help to reduce the rise time and settling time. The Davison also give the worst result in terms of overshoot and fitness function compared to the others.

The other three methods control tuning of GA algorithm, Penttinen-Koivo, Maciejowski and Proposed method shows the similar results. The similarity of the result is because of the same property of equation that they are used, which is proportional and integral term. However, the Proposed method give the better performance compared to Maciejowski even though they are having similarities in the expression. This is because the complexity in finding suitable bandwidth in Maciejowski method has been reduces by the Proposed method.

4.4.3 Result MPID Tuning using PSO

This subchapter will explain about the system performances of MPID control tuning by using particle swarm optimization algorithm. The results of system interaction responses are interpret in waveforms for each types of control tuning as shown in Figure 4.10. All the data are tabulated in Table 4.5 and Figure 4.11 in a form of bar chart. The simulation was done by using the selected parameter from constant scalar parameter initialization process for PSO.

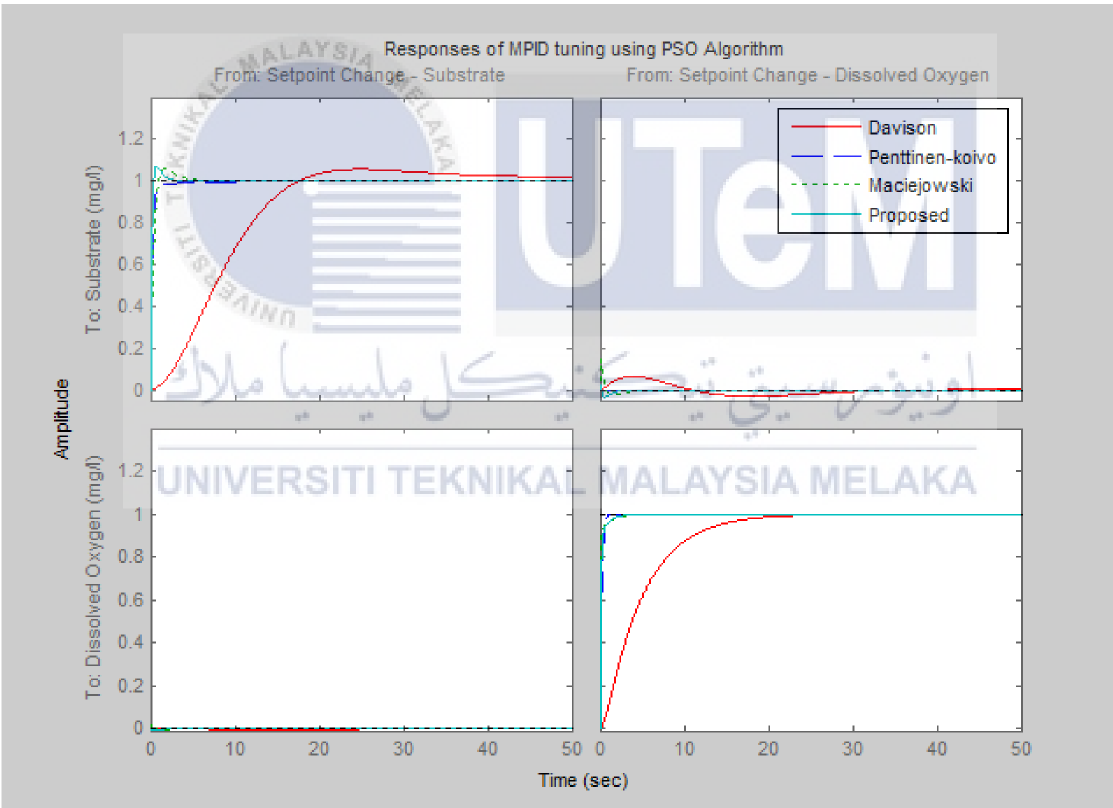


Figure 4.10: System Performance of MPID Control Tuning using PSO

From the PSO output result, there are differences data result between open loop system and activated sludge process with controller tuning system where it give better performances rather than open loop system. Then, the transient response in terms of rise time, settling time and percentage of overshoot from each MPID control methods are being compared to make an analysis on which methods will the best performances for the ASP system.

Table 4.5: MPID Tuning Parameter Data using PSO

Optimization Type	PSO							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	20.7512		0.0173		17.8477		46.5445	
Epsilon, ε	0.1932		4.7946		6.0323		9.3153	
Rho, ρ	-		4.7927		10		8.4862	
Alpha, α	-		-		-		0.0698	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	11.5	10.2	0.488	0.458	0.655	0.154	0.264	0.364
Settling Time, T_S	36.8	18.4	1.14	0.816	3.77	2.12	1.92	1.75
Overshoot, OS%	5.69	0	0.084	9.68e-5	5.90	1.71e-6	7.19	6.30e-6

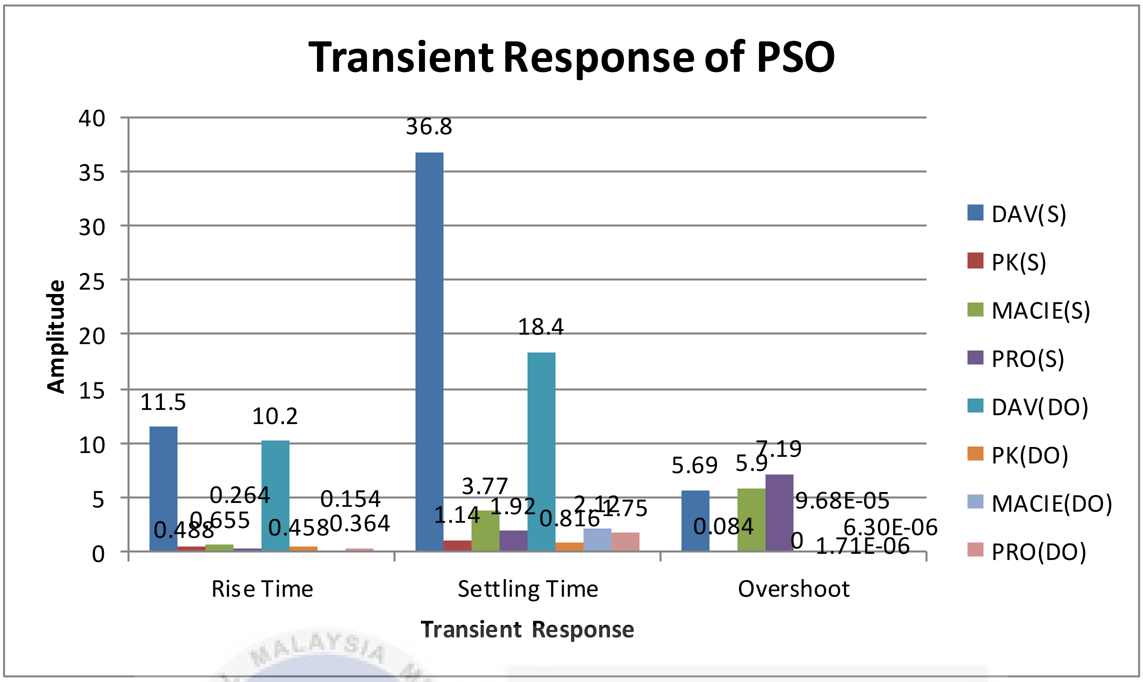


Figure 4.11: Transient Response between MPID Control Method of PSO

Based on the time response data, it can be seen clearly that Davison give the worst response with the longer settling time and rise time. The reason on it is due to the missing proportional term in its equation where it only used the integral term. The benefit of having proportional term in a system is where it can help to reduce the rise time and settling time. While the other three methods shows the similar results. The similarity of the result is because of the same property of equation that they are used, which is proportional and integral term. However, the Proposed method give the better performance compared to the others instead of having an unstable fitness function (ITSE).

4.4.4 Comparisons between BA, GA and PSO

The more details comparison between BA, GA and PSO in terms of transient response and standard deviation based on fitness function will discuss in this subtopic. The comparison is made according to the types of MPID tuning for each optimization techniques. Then, the results are presented in a bar graph from Figure 4.12 until Figure 4.15 for a better viewing and understanding.

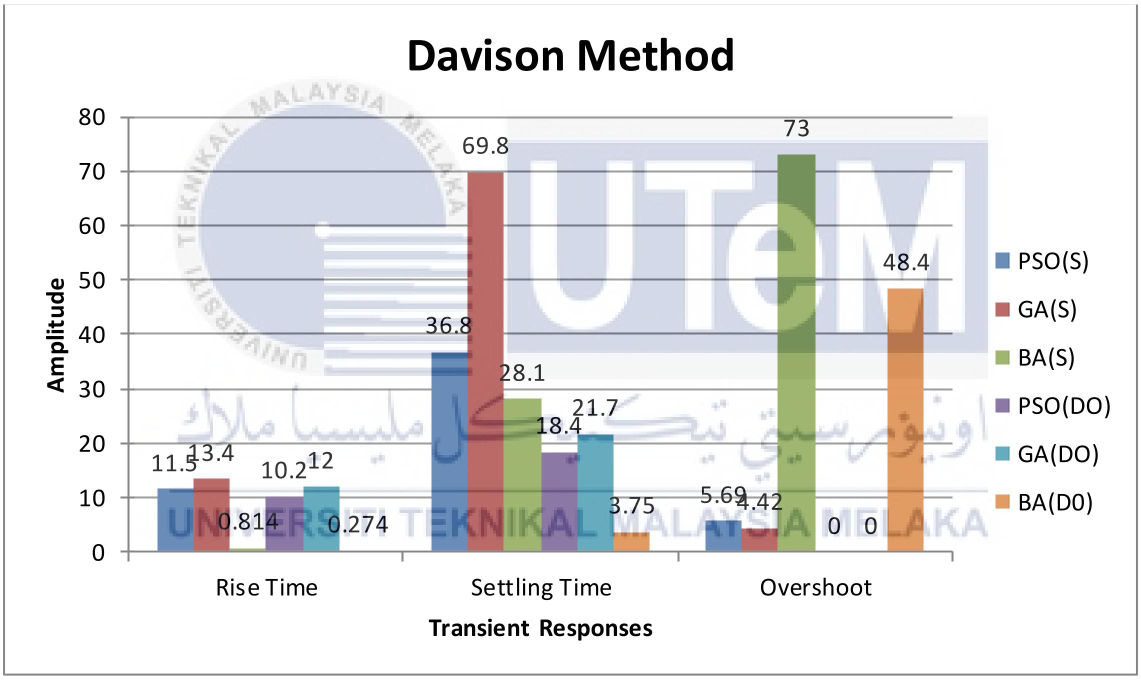


Figure 4.12: Comparison of BA, GA, and PSO for Davison method

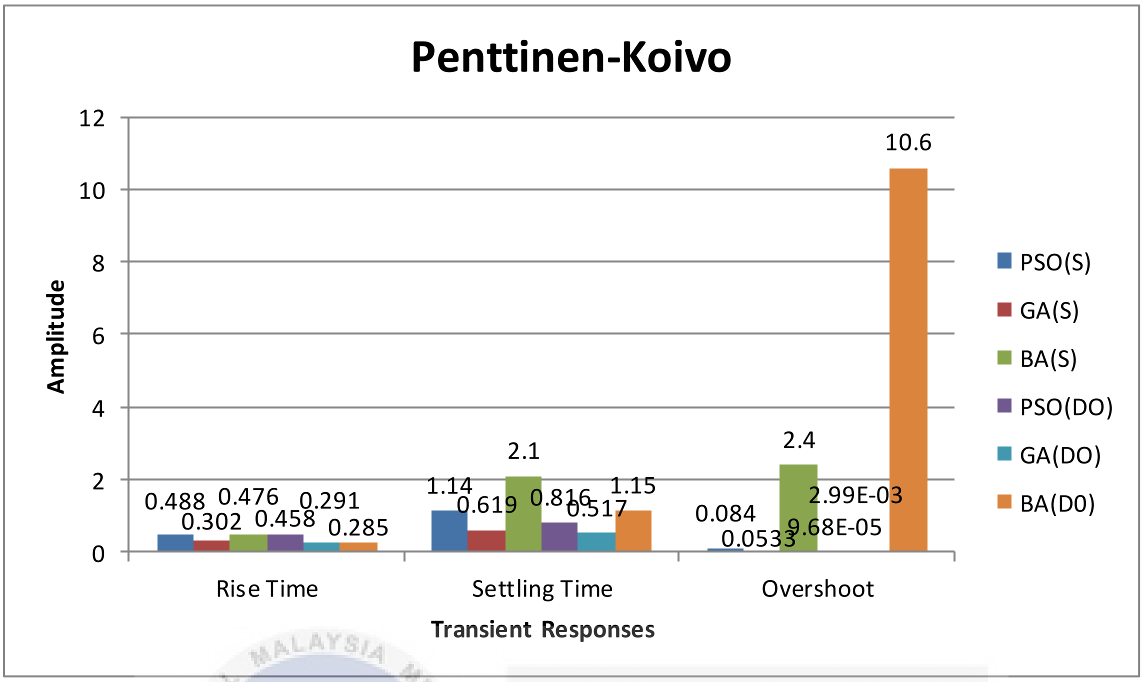


Figure 4.13: Comparison of BA, GA, and PSO for Penttinen-Koivo method

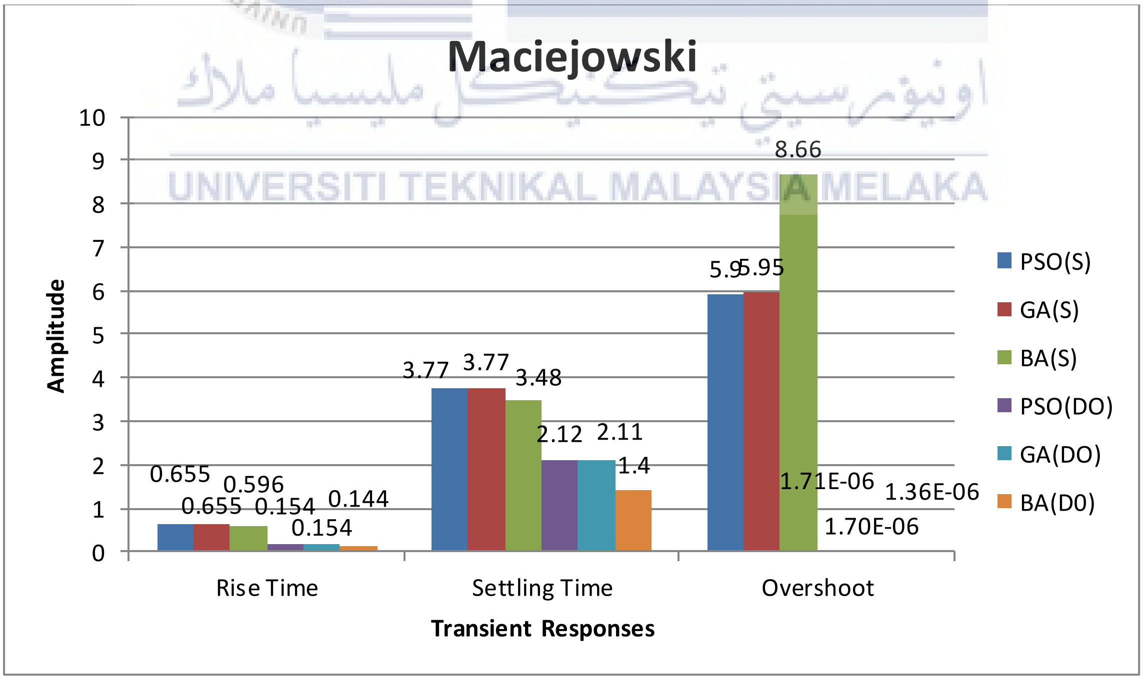


Figure 4.14: Comparison of BA, GA, and PSO for Maciejowski method

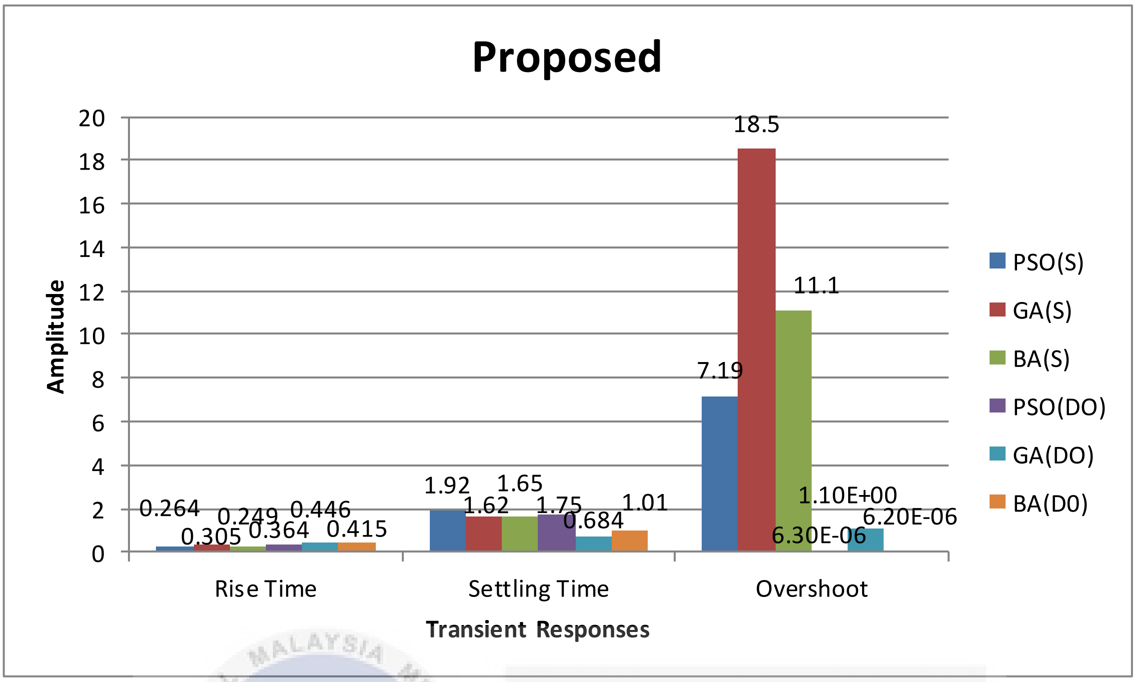


Figure 4.15: Comparison of BA, GA, and PSO for Proposed method

The bar graph shown from Figure 4.12 until Figure 4.15 demonstrate that between the four MPID controllers tuning method, it can be assumed that Proposed method is the best method to be implement in activated sludge process. This is because the result of transient responses for all optimization shows fairly even the results except for some irregularities but if comparison made between the other methods, the Proposed method still give the best transient responses.

Then, the comparison of system performances of using BA, GA and PSO optimization technique are continued by comparing its standard deviation and average mean values of parameter. The data of standard deviation and average mean are tabulated in Table 4.6. While, the more details data can be referred to the Appendix C.

Table 4.6: Comparisons of Standard Deviation and Average of BA, GA and PSO

Control Tuning	Davison			PK		
Optimization Type	BA	GA	PSO	BA	GA	PSO
Standard Deviation	3.6634e-5	38.6806	204.5965	0.3469	0.1463	0.2901
Average	132.6093	109.4487	122.9387	0.5068	0.5847	0.7968
Control Tuning	Maciejowski			Proposed		
Optimization Type	BA	GA	PSO	BA	GA	PSO
Standard Deviation	5.6088	0.0576	0.6954	14.7309	9.7544	58.454
Average	15.3331	17.9626	18.5054	15.0585	23.339	87.377

From Table 4.6, Davison method again gives the worst result but now in terms of standard deviation and average of parameters where Davison created the highest value among the other MPID control methods. This might be caused by the solutions generated by BA, GA and PSO varied greatly. The other three MPID control still give the similar results among them. However, the Proposed method stills the best method for ASP system.

4.5 Comparison of Best MPID Control between BA, GA and PSO

The Proposed method has selected as the best method among the other MPID control methods because it tends to achieve all the requirements of selected criteria by undergoes a certain comparisons. Unfortunately, there is a last comparison need to be done which is comparison of Proposed method between BA, GA and PSO to choose the Proposed method with the best optimization technique.

Table 4.7: Comparisons of Proposed Method between BA, GA and PSO

Optimization Type	BA		GA		PSO	
Fitness Function	9.8716		12.9856		46.5445	
Epsilon, ε	9.9900		9.9178		9.3153	
Rho, ρ	6.0562		4.2566		8.4862	
Alpha, α	0		0.0196		0.0698	
Output	S	DO	S	DO	S	DO
Rise Time, T_R	0.249	0.415	0.305	0.446	0.264	0.364
Settling Time, T_S	1.65	1.01	1.62	0.684	1.92	1.75
Overshoot, OS%	11.1	6.2e-6	18.5	1.10	7.19	6.30e-6
Standard Deviation	14.7309		9.7544		58.4544	
Average	15.0585		23.3395		87.3779	

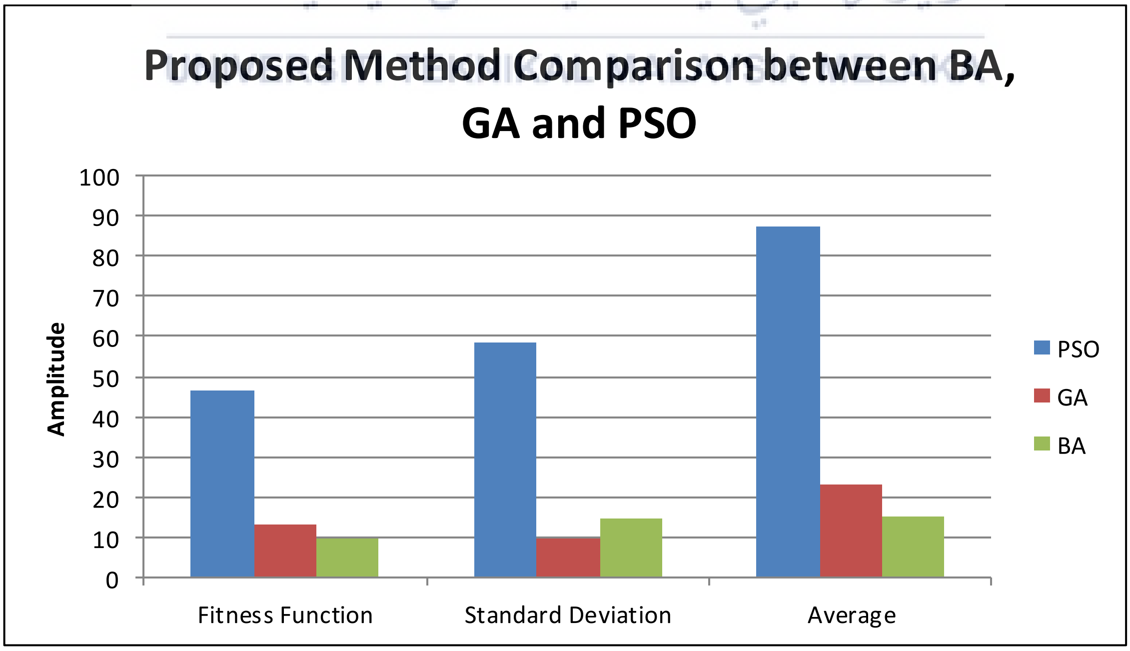


Figure 4.16: Proposed Method Comparison between BA, GA and PSO

Based on Table 4.7 and Figure 4.16, the Proposed method tuning using PSO gives the worst result of fitness function, standard deviation, average and transient response in terms of rise time, settling time and overshoot. It shows clearly that the particle swarm optimization technique is not suitable to be used in the activated sludge process.

Therefore, the comparison is now only between two others optimization techniques which are GA and BA. The GA technique tends to a give system short period of time to settle because of its low value of settling time but it caused increasing value of rise time and overshoot. Genetic algorithm also cannot give a lower fitness function in terms of ITSE because of its low standard deviation and high mean average.

While BA technique has a smaller value overshoot and rise time compared to the GA but a bit late settle which means the system required a longer time to achieve a steady state condition. Bat optimization also has a low value of fitness function in terms of ITSE which means the BA will give a better response compared to GA but the BA cannot give a precise result to the system because of its high level of standard deviation. Both optimizations have their own strengths and weakness but the best optimization that will give a better system performance for ASP system is Bat Algorithm with Proposed MPID controller.

4.6 Nonlinear System of Activated Sludge Process

4.6.1 MPID Tuning Result using BA

This subchapter will explain about the system performances of MPID control tuning by using bat algorithm for the nonlinear system of ASP. The results of this simulation are presented in a waveform for each types of control tuning as shown in Figure 4.17 for substrate and Figure 4.18 for dissolved oxygen, while the data are tabulated in Table 4.8. The simulation was done by using the selected parameter from constant scalar parameter initialization for BA.

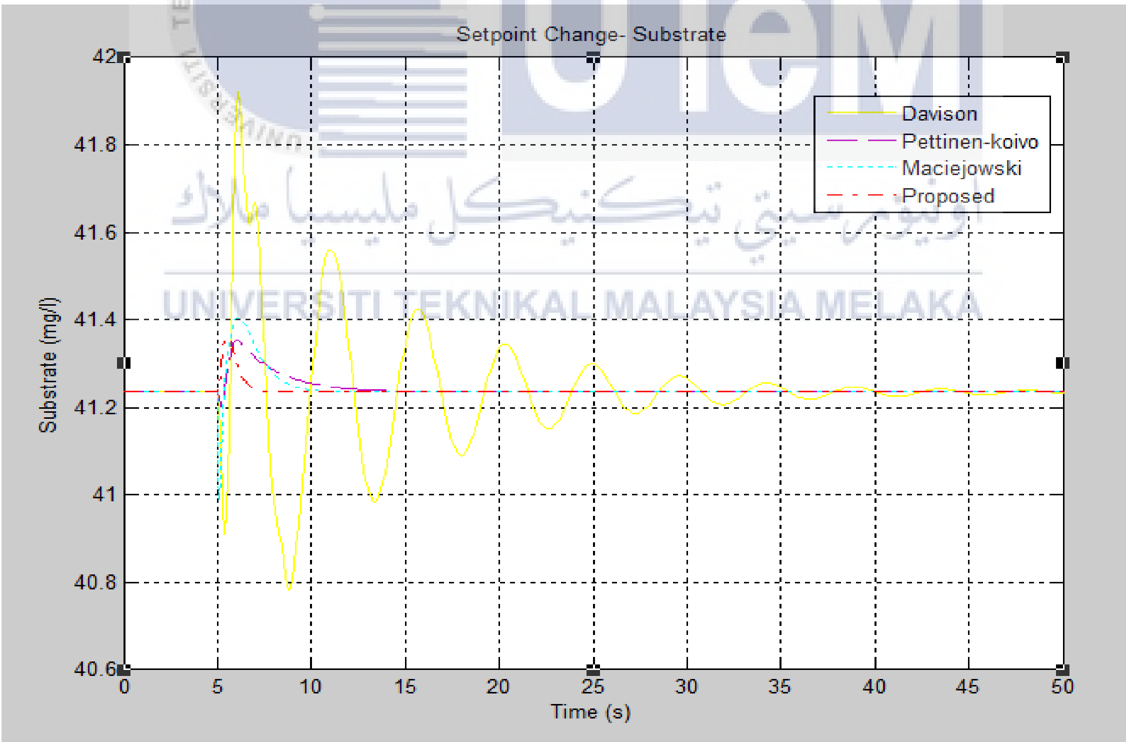


Figure 4.17: Nonlinear System Performances of MPID Control Tuning using BA

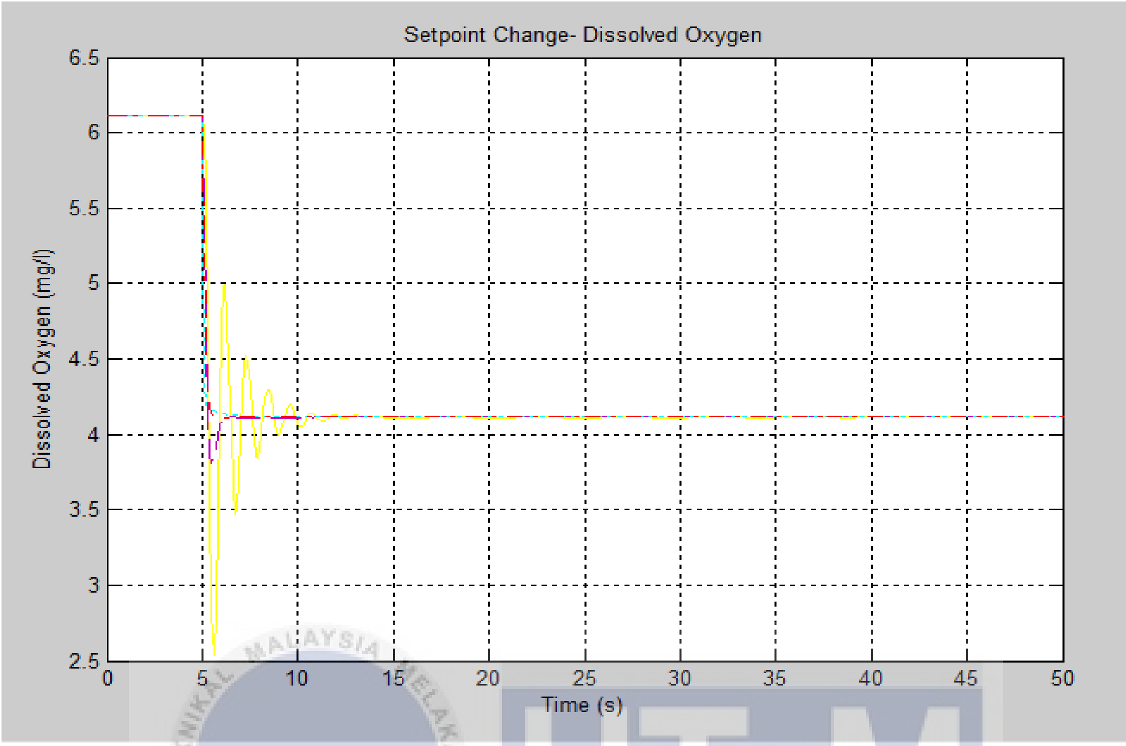


Figure 4.18: Nonlinear System Performances of MPID Control Tuning using BA

Table 4.8: Nonlinear MPID Tuning Parameter Data using BA

Optimization Type	BA							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	5.2768e3		5.0801e3		5.0500e3		5.0721e3	
Epsilon, ε	10.2245		10.0027		7.9816		9.9900	
Rho, ρ	-		4.1857		9.9988		6.0562	
Alpha, α	-		-		-		0	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	0.0146	0.2239	0.0754	0.2245	0.0816	0.0983	0.0816	0.2796
Settling Time, T_S	39.1850	10.3095	14.624	5.9617	9.9422	5.6372	9.9422	5.4777
Overshoot, OS%	1.6692	48.6055	0.2914	48.607	0.4138	48.607	0.4138	48.607

Based on the time response data on Table 4.8 and Figure 4.19, it can be seen clearly that Davison on BA algorithm also give the worst response with the longer settling time and rise time when simulate on a nonlinear ASP system. The reason on it is due to the missing proportional term in its equation where it only used the integral term. The advantages of having proportional term in a system are where it can help to reduce the rise time and settling time. The Davison also give the worst result in terms of overshoot and fitness function compared to the others.

While, the other three methods control tuning of BA algorithm, Penttinen-Koivo, Maciejowski and Proposed method shows the similar results. The similarity of the result is because of the same property of equation that they are used, which is proportional and integral term. However, the Proposed method give the better performance compared to Maciejowski even though they are having similarities in the expression. This is because the complexity in finding suitable bandwidth in Maciejowski method has been reduces by the Proposed method.

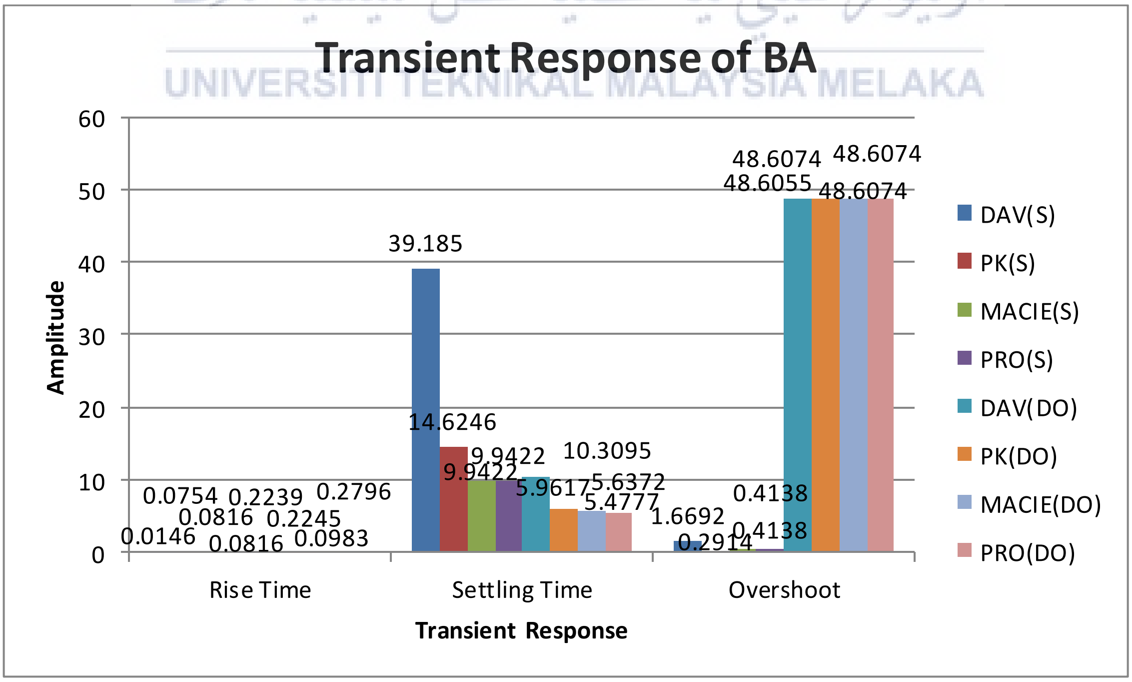


Figure 4.19: Transient Response of BA for Nonlinear System

4.6.2 MPID Tuning Result using GA

This subchapter will explain about the system performances of MPID control tuning by using genetic algorithm for the nonlinear system of ASP. The results of this simulation are presented in a waveform for each types of control tuning as shown in Figure 4.20 for substrate and Figure 4.21 for dissolved oxygen, while the data are tabulated in Table 4.9. The simulation was done by using the selected parameter from constant scalar parameter initialization for GA.

.

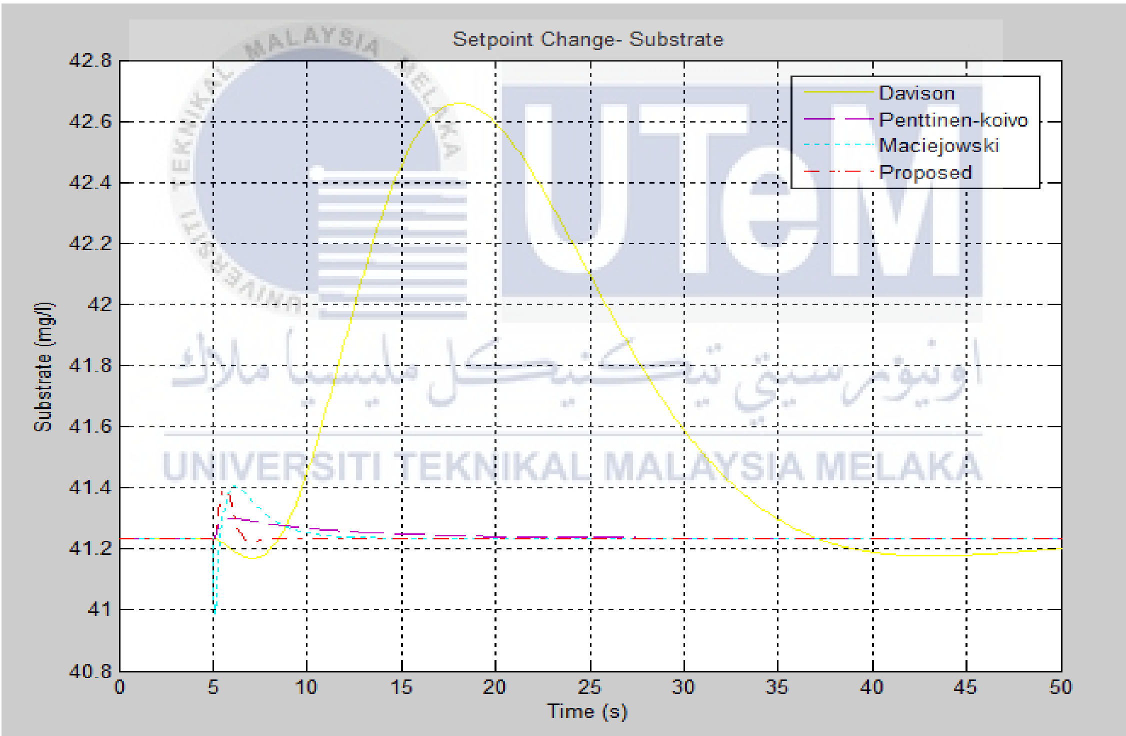


Figure 4.20: Nonlinear System Performances of using GA – Substrate

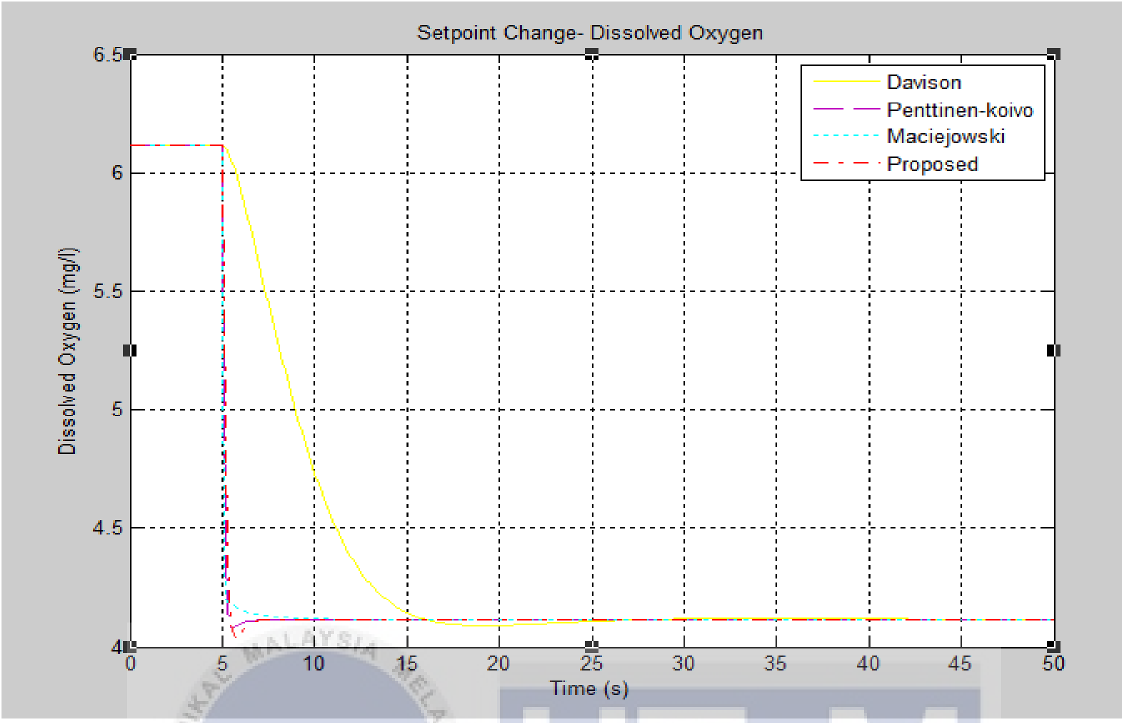


Figure 4.21: Nonlinear System Performances using GA – Dissolved Oxygen

Table 4.9: Nonlinear MPID Tuning Parameter Data using GA

Optimization Type	GA							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	9.9800e3		5.0656e3		5.0512e3		5.0836e3	
Epsilon, ε	0.1666		7.5632		6.0449		9.9178	
Rho, ρ	-		7.5609		9.9784		4.2566	
Alpha, α	-		-		-		0.0196	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	0.6581	6.4767	0.0024	0.2114	0.2540	0.1044	0.0060	0.3256
Settling Time, T_S	37.272	14.728	27.686	5.3199	11.832	6.0587	7.7026	6.1863
Overshoot, OS%	3.5389	48.600	0.1593	48.607	0.4113	48.607	0.3972	48.607

Based on the time response data from Table 4.9 and Figure 4.22, it can be seen clearly that Davison on GA algorithm also give the worst response with the longer settling time and rise time. The other three methods control tuning of GA algorithm, Penttinen-Koivo, Maciejowski and Proposed method shows the similar results. The similarity of the result is because of the same property of equation that they are used, which is proportional and integral term. However, the Proposed method give the better performance compared to Maciejowski even though they are having similarities in the expression. This is because the complexity in finding suitable bandwidth in Maciejowski method has been reduces by the Proposed method.

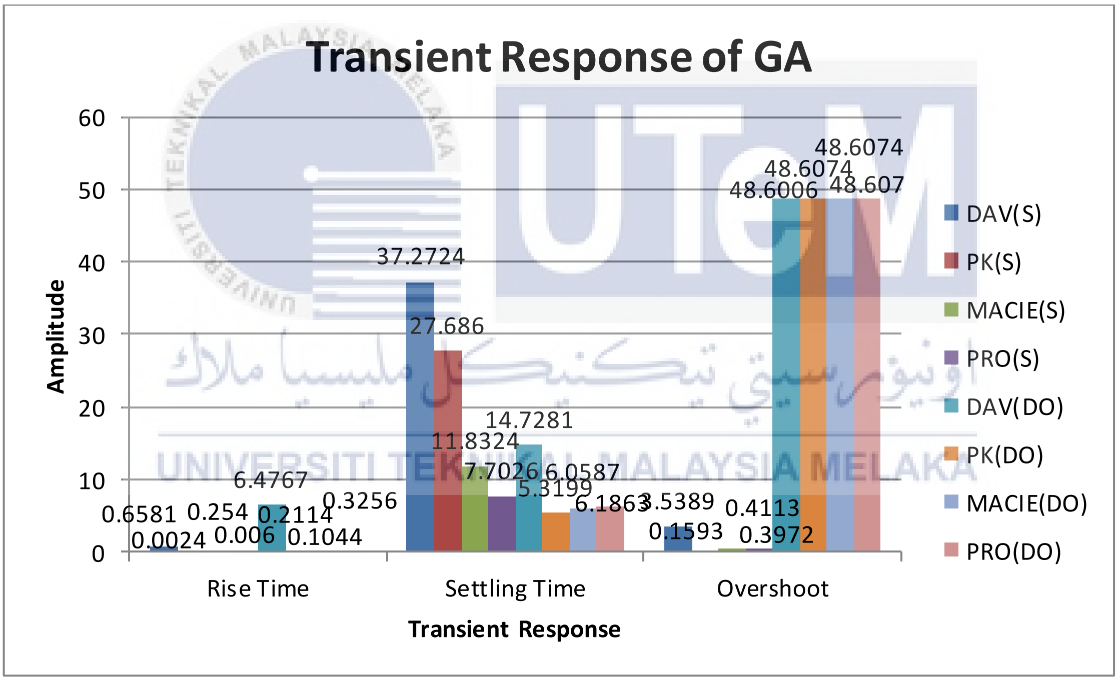


Figure 4.22: Transient Response of GA for Nonlinear System

4.6.3 MPID Tuning Result using PSO

This subchapter will explain about the system performances of MPID control tuning by using particle swarm optimization for the nonlinear system of ASP. The results of this simulation are presented in a waveform for each types of control tuning as shown in Figure 4.23 for substrate and Figure 4.24 for dissolved oxygen, while the data are tabulated in Table 4.10. The simulation was done by using the selected parameter from constant scalar parameter initialization for PSO.

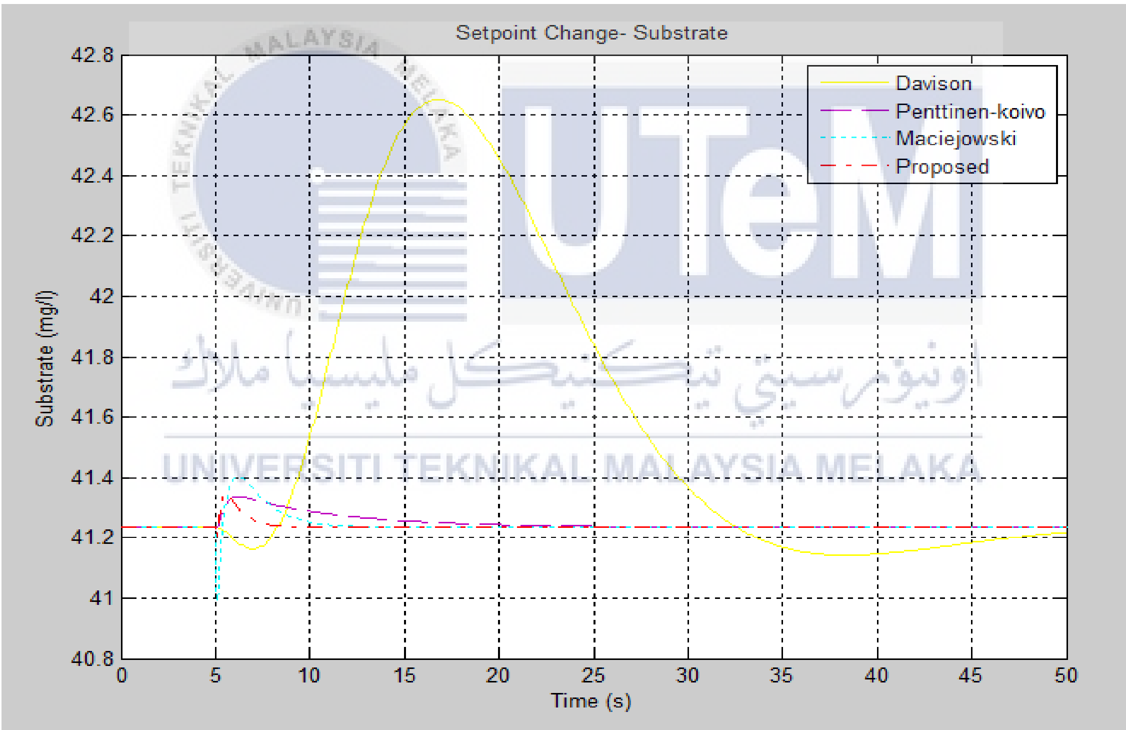


Figure 4.23: Nonlinear System Performances using PSO - Substrate

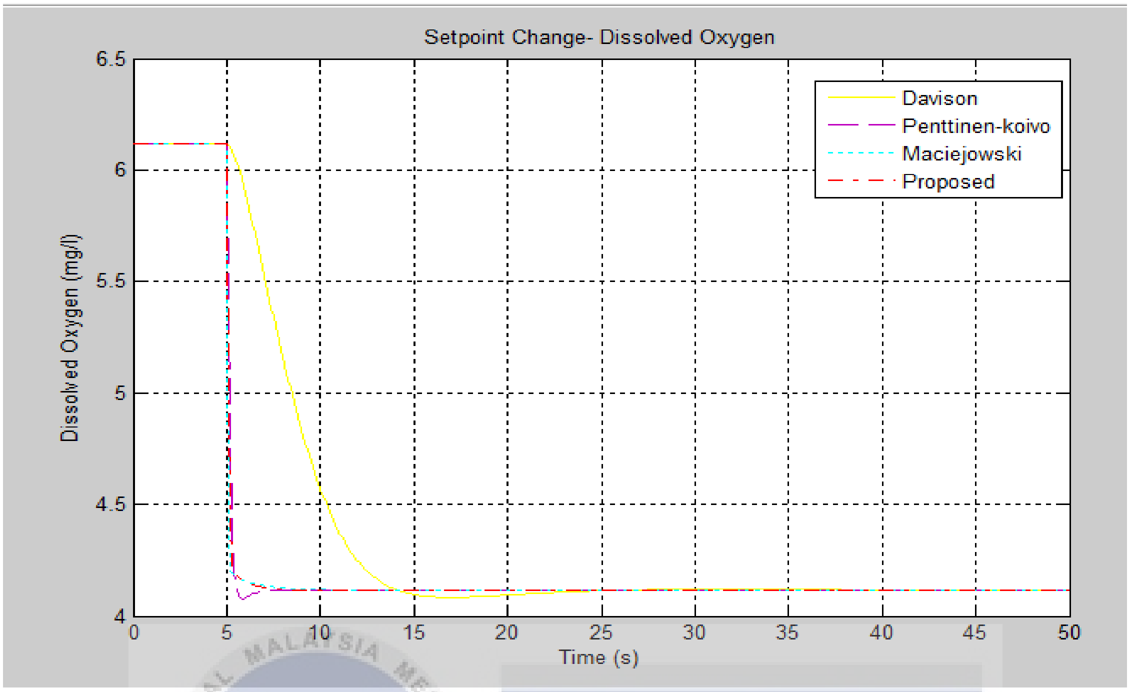


Figure 4.24: Nonlinear System Performances using PSO – Dissolved Oxygen

Table 4.10: Nonlinear MPID Tuning Parameter Data using PSO

Optimization Type	PSO							
Controller Tuning	Davison		Penttinen-Koivo		Maciejowski		Proposed	
Fitness Function	9.1184e3		5.0826e3		5.0512e3		5.0628e3	
Epsilon, ε	0.1932		4.7946		6.0323		9.3153	
Rho, ρ	-		4.7927		10		8.4862	
Alpha, α	-		-		-		0.0698	
Output	S	DO	S	DO	S	DO	S	DO
Rise Time, T_R	0.3500	5.4908	0.0094	0.3248	0.2574	0.1040	0.0105	0.2334
Settling Time, T_S	45.756	13.196	28.011	5.4758	11.862	6.0633	9.0212	5.9621
Overshoot, OS%	3.4745	48.620	0.2470	48.607	0.4106	48.607	0.2713	48.607

Based on the time response data on Table 4.10 and Figure 4.25, it can be seen clearly that Davison give the worst response with the longer settling time and rise time. The reason on it is due to the missing proportional term in its equation where it only used the integral term. The benefit of having proportional term in a system is where it can help to reduce the rise time and settling time. While the other three methods show the similar results. The similarity of the result is because of the same property of equation that they are used, which are proportional and integral term. However, the Proposed method give the better performance compared to the others although the system is nonlinear ASP.

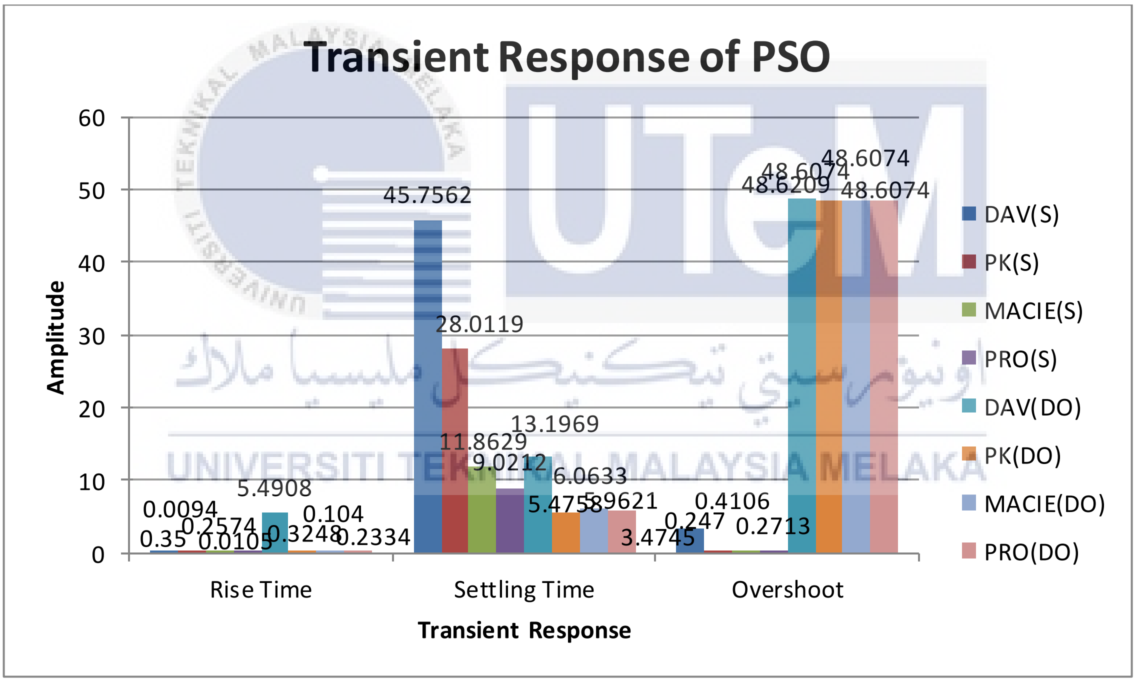


Figure 4.25: Transient Response of PSO for Nonlinear System

4.6.4 Comparisons between BA, GA and PSO

The more details comparison between BA, GA and PSO in terms of transient response and performances index in terms of ITSE will discuss in this subtopic. The comparison is made according to the types of MPID tuning for each optimization techniques. Then, the results are presented in a bar graph from Figure 4.26 until Figure 4.29 for transient response and Figure 4.30 for performances index. The purpose of presenting the result in a bar graph is to give a better viewing and understanding about the comparisons between BA, GA and PSO.

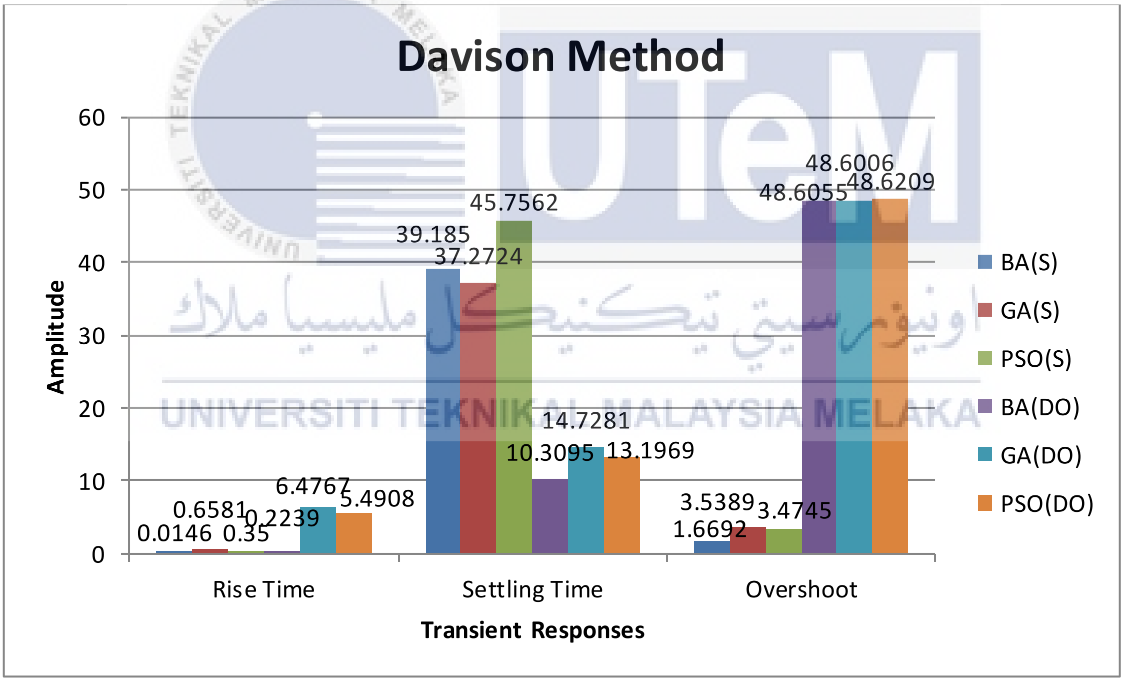


Figure 4.26: Comparison of BA, GA, and PSO for Davison method

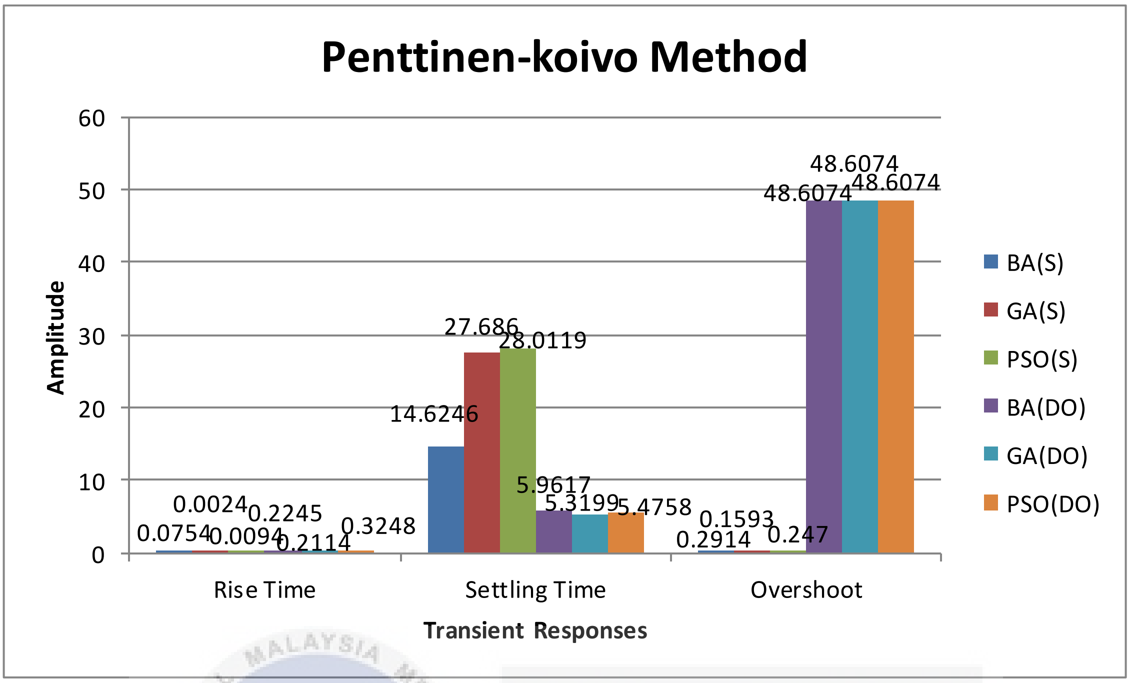


Figure 4.27: Comparison of BA, GA, and PSO for Penttinen-Koivo method

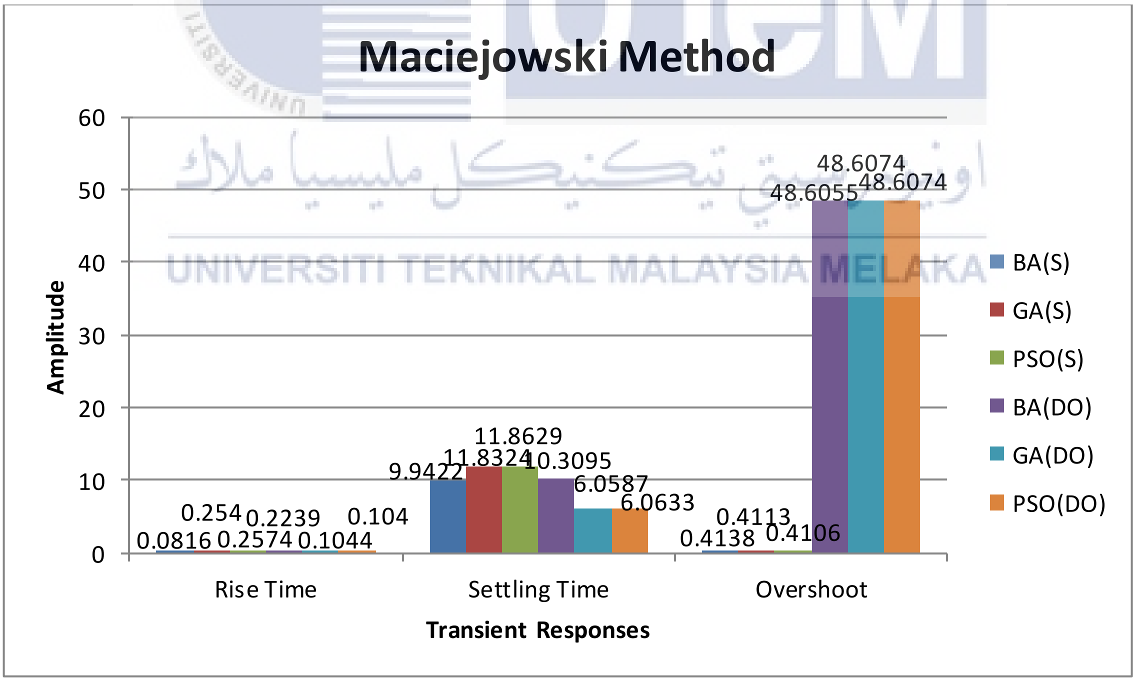


Figure 4.28: Comparison of BA, GA, and PSO for Maciejowski method

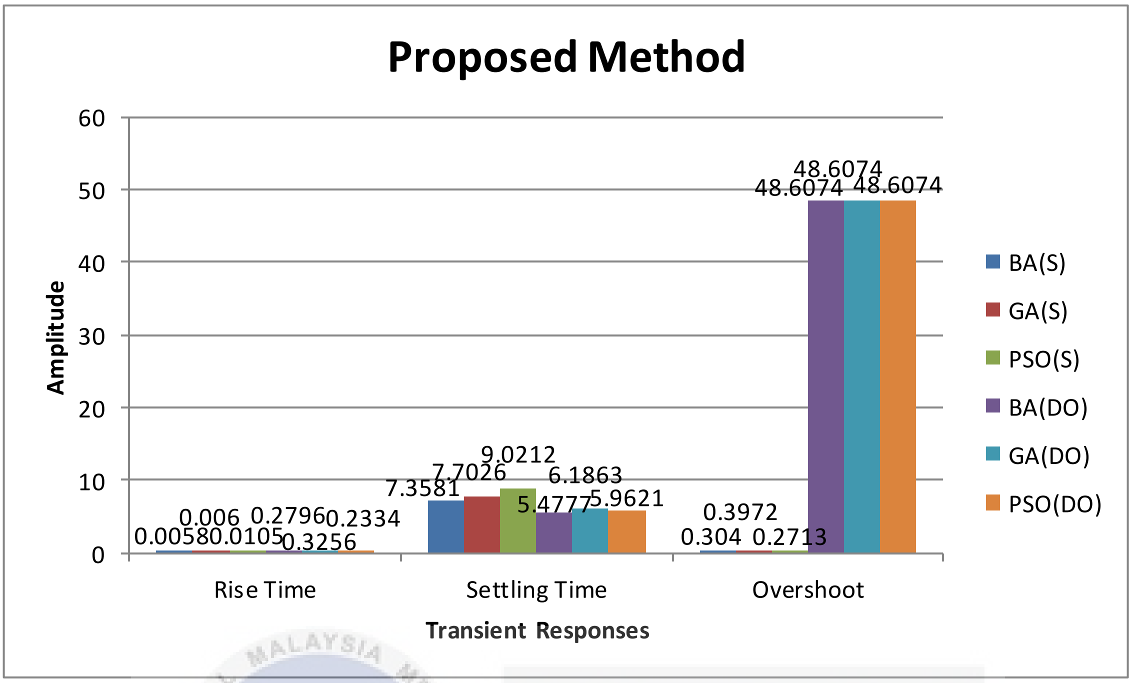


Figure 4.29: Comparison of BA, GA, and PSO for Proposed method

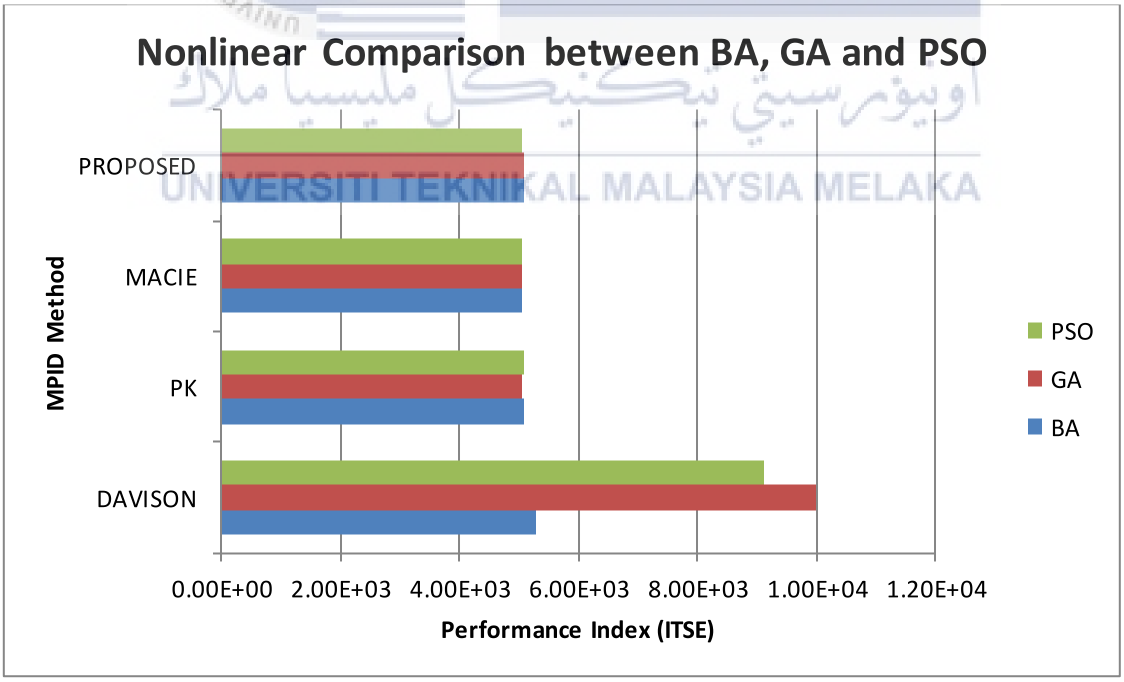


Figure 4.30: Nonlinear Comparison between BA, GA and PSO

Figure 4.30 shows nonlinear comparison between BA, GA and PSO in terms of performance index (ITSE). The Davison method gives the worst result of ITSE for every optimization technique. However, the other methods give a similar results but the Proposed method with BA optimization can be conclude as the best choice to tune the parameters of nonlinear ASP system. This is because the Proposed method come out with the best results of rise time and settling time among others. Then, when the comparison between optimization had done, the BA also was given a better result in terms rise time and settling time although it tends to give a high percentage of overshoot. All the analysis of comparing each optimization for every MPID control tuning method can be referred to the Figure 4.26 until Figure 4.29.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Introduction

This chapter will discuss the conclusion of overall work has been done in this study and future works that can be done based on this study for next generation.

5.2 Conclusion

This aim for this project is to study the performances of MPID control using optimization technique for activated sludge process. Four MPID control tuning methods have been used in this study; Davison, Penttinen-Koivo, Maciejowski and Proposed method. All the results obtained for this study is based on selected criteria which are performances index in terms of fitness function (ITSE), transient responses in terms of rise time, settling time and overshoot, standard deviation and average mean value. Besides, the result from all four MPID tuning methods based on selected optimization techniques; Particle Swarm Optimization, Genetic Algorithm and Bat Algorithm.

After all the analysis being made, it show that Davison method give the worst performance. However, Penttinen-Koivo, Maciejowski and Combined method produce quite similar results. This is because Davison's method only involves with integral term only while the other methods consider the part of proportional and integral terms. The importance of having proportional term in MPID is where the rise time and settle time can be reduced.

This study also proves that by using optimization technique, a system can be tuned to obtain better performance responses. The system response by using the optimization technique produce a better and refined response compared to the open loop response system. The comparison analysis is continued by comparing the MPID tuning method with all optimization technique to select which tuning method will give the best performance to the ASP system. Then, the result concludes that the BA optimization with the Proposed method is the best medium to tuned multivariable PID of activated sludge process. It is because the BA has advantage with its ability to further find a solution with lower fitness function. Thus, the ASP performance can be improved in order to have a great wastewater treatment plant.

This study also has been simulated on the two different mode of ASP system; linear and nonlinear. The purpose of using the linear ASP system is to make a part of linearization of system and part of initialization of parameter become easier. However, the purpose of having nonlinear ASP system in this study is to test whether the MPID controller tuning using optimization can be adapt to the real plant of ASP because the real plant of ASP is design in nonlinear system. From the analysis that had been done, the linear and nonlinear system gives a similar results and a minor conclusion can be made that the MPID controller tuning using BA optimization can be implementing to the real plant of activated sludge process in order to improve the system to have a better quality of water.

5.3 Future Works

Activated sludge process is part of system in wastewater plant. The improvement towards a system in wastewater plant is important nowadays because of the increasing demand on clean environment. Therefore, there are further works can be done to make this study more complete and also the need to seek out the better and efficient way to accomplish this study.

For this study, the BA optimization is only covered the velocity at position solution of the system with fixed frequency. Therefore, the BA optimization in this study can be improved by continue the simulation using the others criteria of bat algorithm which are by varies the loudness, pulse rate and wavelength for the BA to search a better solution.

Other than that, this study also can be better if the system with MPID tuning using PSO, GA and BA optimization techniques are further testing by including a disturbance in the system. This testing process can determine whether the MPID controller can resist with the disturbances or not. It also can prove that whether the system can fix the error caused by the disturbances.

REFERENCES

[1] B. Ahansazan, H. Afrashteh, N. Ahansazan and Z. Ahansazan, *Activated Sludge Process Overview*, International Journal of Environmental Science and Development, vol.5, 2014.

[2] *Predicting Wastewater Sludge Recycle Performance based on Fuzzy Neural Network*

[3] C. Marx, M. Schmidt and J. Flanagan, *Introduction to Activated Sludge Study Guide*, Wisconsin Department of Natural Resources, 2010.

[4] *Explaining the Activated Sludge Process*, Pipeline vol. 14, No.2, 2003.

[5] J. D. Rojas, X. Flores-Alsina, U. Jeppsson and R. Vilanova, *Application of Multivariable Virtual Reference Feedback Tuning for Wastewater Treatment Plant Control*, Control Engineering Practice 20, pp. 499-510, 2012.

[6] M. Mulas, S. Tronci, F. Corona, H. Haimi, P. Lindell and M. Heinonen, *Predictive Control of an Activated Slugde Process: An Application to Wastewater Treatment Plant*, Journal Process Control 35, pp. 85-100, 2015.

[7] G. Olsson, ICA and M. Mulas, *A Subjective Review*, Water Res. 46, pp. 1585-1624, 2012.

[8] M. A. Shannon, P. W. Bohn, M. Elimelech, J. G. Georgiadis, B. J. Marinas, A. M. Mayes, *Science and Technology for Water Purification in The Coming Decades*, Nature 452, pp. 301–310, 2008.

[9] G. Olsson and B. Newell, *Wastewater Treatment Systems*, Modeling, Diagnosis and Control (1st Ed.), 1999.

[10] M. Yong, P. Yongzhen and W. Shuying, *Feedforward-feedback Control of Dissolved Oxygen Concentration in a Predenitrification System*, Bioprocess and Biosystems Engineering 27, pp. 223-228, 2005.

[11] B. Holenda, E. Domokos, A. Redey and J. Fazakas, *Dissolved Oxygen Control of The Activated Sludge Wastewater Treatment Process using Model Predictive Control*, Computer and Chemical Engineering 32, pp. 1270-1278, 2008.

[12] H. Hong-Gui, Q. Hu-Hai and Q. Jun-Fei, *Nonlinear Multiobjective Model-predictive Control Scheme for Wastewater Treatment Process*, Journal of Process Control 24, pp. 47-59, 2014.

[13] W. Shen, X. Chen, M. Ponas and J. Corriou, *Model Predictive Control for Wastewater Treatment Process with Feedforward Compensation*, Chemical Engineering Journal 255, pp. 161-174, 2009.

[14] I. Santin, C. pedret, R. Vilanova and M. Menses, *Advanced Decision Control System for Effluent Violations Removal in Wastewater Treatment Plants*, Control Engineering Practice 49, pp. 60-75, 2016.

[15] J. D. Rojas, J. A. Baeza and R. Vilanova, *Three Degree of Freedom Virtual Reference Feedback Tuning design and its Application to Wastewater Treatment Plant Control*, Proceedings of the 18th World Congress , 2011.

[16] M. W. I. * and S. Baskar, *Evolutionary Algorithms based Design of Multivariable PID Controller*, Expert Systems with Applications, pp. 9, 2009.

[17] N. A. Selamat, N. A. Wahab and S. Sahlan, *Particle Swarm Optimization for Multivariable PID Controller Tuning*, IEEE 9th International Colloquium on Signal Processing and its Applications, 2013.

[18] M. C. Razali, N. A. Wahab, P. Balaguer, M.F. Rahmat and S. I. Samsudin, *Multivariable PID Controllers for Dynamic Process*, IEEE 9th International Colloquium on Signal Processing and its Applications, 2013.

[19] N. Wahab, M. Katebi and J. Balderud, *Multivariable Tuning of Activated Sludge Processes*, In Control 2006, 2006.

[20] “Multivariable Control System Design,” ed.

[21] “Design of Multi-Loop and Multivariable PID Controllers,” ed, p. 39.

[22] M. Carl-Fredrik, J. Nigel and Fitchener-Hocker,

[23] M.C Campi, A. Lechini and S.M. Savaresi, *Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers*, Automatica 38, pp. 1337-1346, 2002.

[24] F. Previdi, F. Fico, D. Belloli, S. Saravesi, I. Presenti and C. Spelta, *Virtual Reference Feedback Tuning of Velocity Controller in Self-balancing Industrial Manual Manipulators*, American Control Conference (ACC), pp. 1956-1961, 2010.

[25] F. Previdi, T. Schauer, S. Savaresi and K. Hunt, *Data-Driven Control Design for Neuroprotheses: A Virtual Reference Feedback Tuning Approach*, IEEE Transactions on Control System Technology 12, pp. 176-182, 2004.

[26] S. Formentin, M. Corno, S, Savaresi and L. Del Re, *Virtual References Feedback Tuning of Internal Model Controller*, IEEE 49th Conference on Decision and Control, pp. 5542-5547, 2010.

[27] J. D. Rojas, R. Vilanova and V.M. Alfaro, *Application of the Virtual Reference Feedback Tuning on Wastewater Treatment Plants: A Simulation Study*, IEE Conference on Emerging Technologies and Factory Automation, pp. 1-8, 2010.

[28] A. Karimi, L. MIskovic and D. Bonvin, *Iterative Correlatiom-based Controller Tuning with application to a Magnetic Suspension System*, Control Engineering Practice 11, pp. 1069-1078, 2003.

[29] H. Hjalmarsson, M. Gevers, S. Gunnarsson and O. Lequin, *Iterative Feedback Tuning: Theory and Applications*, IEEE Control Systems Magazine 18, pp. 26-41, 1998.

[30] S. Ginestet and D. Marchio, *Control Tuning of A Simplified VAV System: Methodology and Impact on Energy Consumption and IAQ*, Energy and Building vol.42, pp. 1205-1214, 2010.

[31] P. Balaguer, N. A. Wahab, M. R. Katebi and R. Vilanova, *Multivariable PID Control Tuning: A Controller Validation Approach*, IEEE Signal Processing and its Applications, 2008.

[32] N. A. Wahab, M. R. Katebi and J. Balderud, *Multivariable PID Control Design for Wastewater Systems*, Proceeding on 15th Mediterranean Conference on Control and Automation, 2007.

[33] M. Dorigo and G. Di Caro, *New Ideas in Optimization*, 1999.

[34] X. S. Yang, *Nature-Inspired Metaheuristic Algorithm*, 2nd Edition, Luniver Press, 2010.

[35] P. M. Pardolas, H. E. Romejin and H. Tuy, *Recent Developments and Trends in Global Optimization*, Journal of Computational and Applied Mathematics 124, pp. 209-228, 2000.

[36] S. Pareek, M. Kishnani and R. Gupta, *Optimal Tuning of PID Controller using Genetic Algorithm and Swarm Techniques*, International Journal of Electronic and Electrical Engineering, pp. 189-194, 2014.

[37] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, *Optimization by Simulated* Annealing, Science 220, pp. 671-680, 1983.

[38] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[39] Z. W. Geem, J. H Kim, G. V. Loganathan, *A new Heuristic Optimization Algorithm: Harmony Search*, Simulation 76, pp. 60-68, 2001.

[40] T. Back, D. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computational*, Oxford University Press, 1997.

[41] M. Dorigo and C Blum, *Ant Colony Optimization Theory: A Survey*, Theory Computer Science 334, pp. 243-278, 2005.

[42] S. Agarwal, A. P. Singh and N. Anand, *Evaluation Performance Study of Firefly Algorithm, Particle Swarm Optimization and Artificial Bee Colony algorithm for Non-linear Mathematical Optimization Function*, ICCCNT, 2013.

[43] X. Yang, *Metaheuristic Optimization*, Scholarpedia, vol. 6, 2011.

[44] G. Wang and L. Guo, *A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization*, Journal of Applied Mathematics Volume 2013, 2013.

[45] R. Nakamura, L. Pereira, K. Costa, D. Rodrigues, J. Papa and BBA, *A Binary Bat Algorithm for Feature Selection*, 2012.

[46] S. Pareek, M. Kishnani and R. Gupta, *Optimal Tuning of PID Controller using Genetic Algorithm and Swarm Techniques*, International Journal of Electronic and Electrical Engineering, pp. 189-194, 2014.

[47] Y. Yan, W. A. Klop, M. Molenaar and P. Nijdam, *Tuning of PID Controller: Particle Swarm Optimization versus Genetic Algorithm*, 2010.

[48] X. S. Yang, *Bat Algorithm for Multiobjective Optimization*, Int. J. Bio-Inspired Computation, pp. 267-274.

[49] G. Komarasamy and A. Wahi, *An Optimized K-means Clustering Technique using Bat Algorithm*, Eurio J Sci Res 84, pp. 263-273, 2012.

[50] S. Yilmaz and U. Kucuksille Ecir, *Improved Bat Algorithm on Continuous Optimization Problem*, Lecture Notes on Software Engineering, pp. 279-283, 2013.

[51] J. Perez, F. Valdez, O. Castillo and O. Roeva, *Bat Algorithm with Parameter Adaptation using Interval Type-2 Fuzzy Logic for Benchmark Mathematical Functions*, IEEE 8th International Conference on Intelligent Systems, 2016.

[52] S. Surender Reddy and P. R. Bijwe, *Efficiency Improvements in Metaheuristic Algorithms to Solve the Optimal Power Flow Problem*, Electrical Power and Energy Systems 82, pp. 288-302, 2016.

[53] X. S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*, 2010.

APPENDIX A

Gantt Chart

Year	2016			
Project Activities	SEPT	0CT	NOV	DEC
First Stage				
Brainstorming idea				
Project planning and find supervisor				
Project Outline				
Second Stage				
Collect data				
Define objective, procedure and simulation				
Third Stage				
Article and journal review				
Books for reference				
Define problems				
Fourth Stage				
MPID selection				
Optimization technique selection				
Software development				
Result analysis				
Fifth stage				
Show studies to the supervisor				
Prepare for presentation				
Viva presentation				
Report writing				
Final proposal				
Submit proposal writing				

Year	2017				
Project Activities	FEB	MAR	APR	MAY	JUNE
First Stage					
Brainstorming idea					
Project Development					
Project Outline					
Second Stage					
Start writing MATLAB command					
Finalizing the MATLAB coding					
Third Stage					
Project executions					
Collect data					
Organized the data					
Fourth Stage					
Result analysis					
Prepare second project proposal					
Fifth stage					
Show studies to the supervisor					
Prepare for presentation					
Viva presentation					
Report writing					
Final report					
Submit final report					

APPENDIX B

NO. OF PARTICLES

(Max iteration=100, Range= 0-10)

DAVISON METHOD						
No. of Particle	PSO		GA		BA	
	ITSE	ε	ITSE	ε	ITSE	ε
10	132.6113	10.0000	132.6228	9.8687	132.6107	10.0338
20	132.6483	9.3914	132.6221	9.9759	132.6113	10.0796
30	132.6227	9.8884	132.6115	9.9870	132.6113	10.1132
40	132.6113	10.000	87.3580	0.0702	132.6094	10.1615
50	103.3477	1.9084	56.4336	0.0877	132.6092	10.2143

PENTTINEN-KOIVO METHOD									
No. of Particle	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	0.6813	8.7641	3.7091	0.0395	9.8927	8.8377	0.0391	10.0405	9.2122
20	0.7584	9.0242	3.5199	0.0400	9.8557	8.2617	0.0455	10.0854	9.2312
30	0.6393	9.2259	3.8526	0.3090	9.9976	9.2174	0.0385	10.1076	9.1301
40	1.1201	9.7506	9.2674	0.0394	9.9307	9.7598	0.0384	10.1604	9.3145
50	1.1170	6.6325	2.7980	0.0394	9.8946	9.3713	0.0383	10.1866	9.3081

MACIEJOWSKI METHOD									
No. of Particle	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	21.1952	8.1282	9.0379	18.1326	6.7833	9.8940	0.9118	10.003	10.0577
20	19.3312	9.8389	9.3226	18.1603	4.8423	9.9861	0.8706	5.8458	10.0741
30	19.2056	9.9617	9.7224	17.9153	6.5732	9.9958	0.8658	5.7378	10.1316
40	18.7801	9.8764	9.6455	18.0315	6.6217	9.9358	0.8637	5.8861	10.1552
50	17.9837	5.5824	9.0379	17.8994	5.9301	9.9711	0.8637	5.7866	10.1571

PROPOSED METHOD									
No. of Particle	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	50.4093	8.2288	0.3427	2.4373	4.2157	2.8021	10.283	10.0038	4.1372
20	42.4762	2.7785	2.6440	3.7614	4.6885	3.9272	8.2111	10.0102	4.2775
30	47.7986	2.4616	5.3517	1.0401	7.6540	3.2257	0.0662	9.9968	4.2240
40	38.0167	2.5755	8.2597	0.6038	9.6432	3.7312	8.2117	9.9834	4.2806
50	47.8182	2.5816	6.4199	0.6705	8.5124	3.3843	8.1920	10.0068	4.3330

RANGE

(Max iteration=100, No. of particles=10)

DAVISON METHOD						
Upper boundary	PSO		GA		BA	
	ITSE	ε	ITSE	ε	ITSE	ε
10	132.5201	9.6942	132.5331	9.9177	132.6108	10.0270
50	132.6512	8.8704	132.6113	10.0000	0	11
100	133.0421	15.9921	132.6225	16.0223	0	11
500	139.5926	21.2563	132.5409	16.1452	0	11
1000	388.0382	154.7523	132.5849	12.2166	0	11

PENTTINEN-KOIVO METHOD									
UB	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	1.2627	9.5092	7.2235	0.7046	9.2950	3.6666	0.0390	10.0233	9.7897
50	0.0042	22.350	29.378	0.0016	34.498	36.929	0.0076	49.9955	50.0063
100	4.34e-8	93.161	78.310	4.28e-4	96.399	86.617	0.0038	99.9971	100.008
500	8.40e-5	489.28	424.29	8.81e-5	486.23	456.86	7.74e-7	500.025	402.542
1000	6.25e-5	763.67	558.82	4.50e-5	913.90	789.04	3.91e-7	1000.00	795.053

MACIEJOWSKI METHOD									
UB	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	18.865	5.7678	9.4403	18.2992	8.2036	9.9783	0.9119	9.9840	10.053
50	0.9985	12.6456	44.1981	0.1949	2.4423	46.439	0.0356	0.2247	47.476
100	0.5226	63.3333	93.0776	0.0064	0.9002	98.241	0.0095	0.7635	99.999
500	0.1123	466.529	301.435	0.0020	25.442	405.40	0.0001	-1.94e-7	5000.0
1000	0.1521	752.334	894.900	0.0450	674.69	463.92	0.0066	1.00e3	1000.0

PROPOSED METHOD									
UB	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
10	102.87	8.9413	9.3399	67.0329	8.9093	8.5326	9.0138	10.0069	4.2389
50	71.567	30.0771	19.3386	17.0656	43.204	43.204	2.2133	27.7800	49.994
100	4.40e2	75.8112	87.1111	16.7145	23.098	48.748	0.4102	100.003	32.812
500	2.69e2	203.363	333.465	0.0123	42.870	496.07	0.0799	113.720	499.99
1000	1.36e2	442.498	393.465	0.0470	77.466	501.49	0.0395	0.0013	999.99

ITERATION

(Upper boundary=10, No. of particles=50)

DAVISON METHOD						
Upper boundary	PSO		GA		BA	
	ITSE	ε	ITSE	ε	ITSE	ε
10	111.2652	1.0355	104.6251	1.6197	132.6107	10.0192
50	105.6575	1.4633	74.1422	0.0762	132.6097	10.1216
100	26.8364	0.1662	132.6220	9.9832	132.6092	10.2015

APPENDIX C

*Davison method

No	PSO		GA		BA	
	ITSE	ε	ITSE	ε	ITSE	ε
1	105.1711	1.5309	132.6113	9.9986	132.6093	10.1766
2	109.5547	1.1226	103.6481	1.8280	132.6093	10.1834
3	113.6636	0.9373	132.6113	10	132.6093	10.1754
4	132.6115	9.9882	15.1039	0.0613	132.6093	10.1879
5	28.6429	0.1601	132.6113	9.9969	132.6093	10.1992
6	27.9559	0.1623	106.3314	1.3763	132.6093	10.1893
7	27.7129	0.1283	132.6114	9.9944	132.6093	10.1844
8	22.0837	0.1862	132.6114	9.9948	132.6093	10.1850
9	20.7512	0.1932	132.6114	9.9952	132.6092	10.2245
10	26.1300	0.1688	104.8347	1.5839	132.6093	10.1940
11	105.3699	1.5021	132.6113	9.9998	132.6093	10.1823
12	112.9200	0.9641	105.6091	1.4697	132.6093	10.1978
13	132.6113	10	132.6113	9.9968	132.6092	10.2070
14	25.1175	0.1727	26.7134	0.1666	132.6092	10.2146
15	24.6849	0.1744	132.6114	9.9953	132.6093	10.1766
16	132.6356	9.6943	104.2763	1.6858	132.6093	10.1898
17	104.2543	1.6903	31.1109	0.1204	132.6093	10.2010
18	119.4712	0.7902	132.6221	9.9764	132.6093	10.1811
19	117.4494	0.8328	132.6113	9.9968	132.6093	10.1922
20	969.9822	1.8367	132.6113	9.9976	132.6093	10.1863
Average	122.9387		109.4487		132.6093	
Standard Deviation	204.5965		38.6806		3.6634e-5	
Variance	41859.74		1496.191		1.3421e-9	

*Penttinen-Koivo method

No	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
1	0.5954	9.5577	4.0055	0.6102	9.8767	3.9690	0.5504	10.0366	4.1882
2	1.3259	9.9859	6.6104	0.5843	9.9702	4.0618	0.5497	10.0442	4.1909
3	0.6774	9.0940	3.7332	0.5784	9.7498	4.0728	1.0105	10.0597	10.0811
4	0.7815	8.1691	3.4351	0.5736	9.9178	4.0982	0.5500	10.0397	4.1897
5	0.6038	9.9260	3.9923	0.5576	9.9802	4.1583	0.0076	10.0043	10.0009
6	0.7098	8.5345	3.6231	0.6840	9.8903	3.7460	1.0127	10.0385	10.0604
7	0.0173	4.7946	4.7927	0.5974	9.7114	4.0065	0.5507	10.0308	4.1868
8	1.1451	9.5421	9.0503	0.5844	9.8286	4.0560	0.0075	10.0056	10.0555
9	0.7306	9.5221	3.6075	0.6408	9.1980	3.8458	1.0124	10.0419	10.0625
10	0.5920	9.7017	4.0247	0.6392	9.3636	3.8572	0.5492	10.0487	4.1934
11	0.6138	9.3962	3.9375	0.8106	8.0383	3.3656	0.0075	10.0003	10.0002
12	1.0846	9.3923	9.4194	0.0104	7.5632	7.5609	0.5512	10.0251	4.1845
13	1.0676	7.8957	2.9066	0.6174	9.9193	3.9471	0.5503	10.0369	4.1887
14	0.8768	7.4588	3.2101	0.5864	9.8700	4.0506	0.5509	10.0283	4.1857
15	1.1595	6.2079	2.7344	0.6246	9.3779	3.9037	0.0075	10.0017	10.0016
16	1.0414	7.9330	2.9464	0.5716	9.8238	4.1002	1.0106	10.0583	10.0799
17	0.6723	9.2637	3.7537	0.5827	9.8861	4.0643	0.5511	10.0583	4.1847
18	0.8092	9.3701	3.4170	0.6063	9.7175	3.9761	0.0075	10.0027	4.1857
19	0.5961	9.9663	4.0201	0.5795	9.8864	4.0758	0.5483	10.0584	4.1970
20	0.8368	7.7823	3.3016	0.6544	9.8800	3.8311	0.5508	10.0294	4.1862
Average	0.7968			0.5847			0.5068		
Standard Deviation	0.2901			0.1463			0.3469		
Variance	0.0842			0.0214			0.1203		

*Maciejowski method

No	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
1	18.9681	7.9221	9.5949	17.9783	5.4315	9.9602	17.5556	6.0607	10.1791
2	18.0579	5.3860	9.9170	17.9057	5.7966	9.9728	1.2394	7.9832	9.9999
3	18.2682	4.7911	9.9390	17.9761	6.7643	9.9779	17.5681	6.1277	10.1710
4	18.0566	6.6444	9.9210	17.8869	6.0449	9.9784	17.5378	5.9307	10.1918
5	18.1897	7.6490	9.9646	17.8919	5.7033	9.9879	17.5885	6.0629	10.1595
6	17.8483	5.9892	10	17.9643	6.0001	9.9322	4.4307	7.9834	10.000
7	17.9126	6.0439	9.9632	17.9317	6.2056	9.9702	18.3287	9.9442	10.1896
8	19.1257	8.6484	9.6070	18.0099	5.3682	9.9509	17.6112	6.0398	10.1455
9	18.7382	7.0432	9.6034	17.8911	6.0437	9.9754	17.6220	6.1414	10.1392
10	20.1291	6.4911	8.8805	17.9269	6.4486	9.9839	17.5678	5.9315	10.1737
11	17.8476	6.0320	10	18.0108	5.3609	9.9513	17.5906	5.9921	10.1598
12	18.3791	6.9713	9.7841	17.9130	6.2670	9.9847	17.6244	6.0741	10.1358
13	18.7389	4.9482	9.5716	17.9742	6.1721	9.9445	17.5404	5.9146	10.1903
14	17.8541	5.8744	10	18.0252	5.6553	9.9109	17.5316	5.9214	10.1956
15	20.0170	6.4038	8.9203	18.0091	5.9362	9.9049	17.5877	6.0547	10.1598
16	17.8477	6.0323	10	17.9415	5.3993	9.9718	17.5274	5.9250	10.1981
17	18.1464	6.8350	9.8919	17.8934	6.2863	9.9969	17.5743	6.0745	10.1683
18	18.8375	4.6841	9.6067	18.0266	6.6898	9.9423	17.5370	5.9351	10.1922
19	19.1490	9.2911	9.3234	18.0940	5.3862	9.8952	1.5119	7.9816	9.9988
20	17.9965	5.5063	9.9404	18.0010	6.1528	9.9284	17.5882	6.1265	10.1598
Average	18.5054			17.9626			15.3331		
Standard Deviation	0.6954			0.0576			5.6088		
Variance	0.4835			0.0033			31.4586		

*Proposed method

No	PSO			GA			BA		
	ITSE	ε	ρ	ITSE	ε	ρ	ITSE	ε	ρ
1	94.5430	6.6966	8.5795	14.1003	8.2506	3.7218	1.8716	9.9900	6.0562
2	57.8459	2.1161	9.1180	23.6414	8.1870	8.1870	8.2205	9.9900	4.2659
3	299.2116	1.1269	0.1950	38.8428	9.4192	2.7353	8.2264	10.0071	4.2659
4	97.8362	4.2608	3.5965	14.8539	9.8289	4.8003	9.8716	10.0000	6.0562
5	52.6630	8.5788	8.3163	21.2018	7.0944	3.7542	8.2205	9.9900	4.2659
6	55.8326	2.8024	3.6815	26.5399	9.5677	5.8656	8.2264	10.0071	4.2566
7	71.6117	2.2380	5.0688	17.9636	8.2562	8.2562	8.2108	10.0025	4.2755
8	58.2865	2.6811	6.0279	14.7777	9.3861	3.9527	8.2261	10.0075	4.2569
9	59.8541	8.9019	7.8029	19.7671	8.7848	4.3940	8.1962	7.9933	4.3053
10	74.4281	8.2464	7.0651	25.1652	2.3831	7.2406	42.0000	0	0
11	82.3573	7.1812	7.6110	22.6377	8.6403	3.0471	8.2155	10.0017	4.2073
12	57.2443	2.0517	1.8710	19.0648	8.5699	4.1416	8.2246	10.0188	4.2584
13	58.3203	1.8017	7.9901	29.1700	9.2113	9.2113	45.1940	10.0024	10.0039
14	173.0715	6.7966	4.9578	38.1746	7.6951	6.0685	8.1645	10.0067	4.3492
15	80.1419	1.8239	5.0216	20.2511	6.6053	2.7119	42.0000	0	0
16	46.5445	9.3153	8.4862	49.9341	7.2133	2.1942	8.1941	10.0033	4.3061
17	48.1098	2.0331	1.4199	14.3352	8.8436	3.4832	8.2253	10.0030	4.2587
18	128.6292	5.9043	9.1810	28.5299	9.3462	4.3503	8.2765	10.0196	4.2092
19	90.1567	2.3840	9.1334	12.9856	9.9178	4.2566	8.2272	9.9988	4.2564
20	60.8714	2.7185	9.8668	14.8539	9.8289	4.8003	45.1804	10.0082	10.0042
Average	87.3779			23.3395			15.0585		
Standard Deviation	58.4544			9.7544			14.7309		
Variance	3416.916			95.1475			217.000		