FIREFLY ALGORITHM WITH APPLICATION TO ENGINEERING PROBLEMS

VONG HUI CHON

BACHELOR OF ELECTRICAL ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

FIREFLY ALGORITHM WITH APPLICATION TO ENGINEERING PROBLEMS

VONG HUI CHON

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

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "FIREFLY ALGORITHM WITH APPLICATION TO ENGINEERING PROBLEMS is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : Vong Hui Chon

Date :

APPROVAL

I hereby declare that I have checked this report entitled "Firefly Algorithm With Application To Engineering Problems" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Electrical Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

First at all, i would like to thank to University Teknikal Malaysia Melaka (UTeM) for giving me this wonderful opportunity to futher my study. I am grateful for having an opportunity to meet with professionals who led me through this four years study. Without the study, I won’t have the opportunity to do final year project.

A deep thanks to my supervisor, Dr Mohd Ruzaini Bin Hashim, for sparing his valuable time and precious guidance. He has given me all the freedom to pursue my tasks and ensure the tasks are done perfectly. With his guidance, I have developed new skills that may support my future development. My special thanks to him for his particular time and guidance.

Special thanks to my family which they are the biggest source encouragement and always supporting me. The blessing of my parents, Mr. Vong Chit Fah and Mrs. Jong Mui Fah had encouraged me and helped me to reach on this stage in my life. Without them, I would not have been where and what I am today. Not forget my sibling for their love and care. This is impossible without their love and moral support given to me at all the times.

Lastly, I would like to thanks my friends for their even ready help and support. Throughout the final year project, we are helping each other to complete the task on time. In addition, they are willing to share the experience and teaching each other and glow together.

ABSTRACT

In order to solve difficult optimization problems, there are many bio-inspired optimization techniques can be used. Firefly algorithm is a natural inspired algorithm that can solve the optimization problems. Firefly algorithm consists some parameter can be controlled, which this may influence the performance of the algorithm. This research is focus on the variation of parameter setting to know the performance of the algorithm. In order to verify and test the performance of Firefly algorithm, benchmark functions can be used. The parameter settings that been varying is the number of dimension, number of population and number of iterations. When the amount of fireflies increases, it will move toward the optimal point. The algorithm will perform better at lower dimension compare to higher dimension. The flexible manipulator system (FMS) is used as application to know the performance of the algorithm. The Firefly Algorithm is used to tune the PID controller with using the performance criteria. The parameters of the PID controller will affect the performance of the flexible manipulator system (FMS). The analysis method is used to analyse the performance of the hub angle of FMS. There are 3 conditions been set to analyse the performance of the hub angle for each performance criteria. The most suitable performance criteria as the objective function of Firefly Algorithm to tune the PID controller been selected.

ABSTRAK

Untuk menyelesaikan masalah pengoptimuman yang sukar, terdapat banyak teknik pengoptimuman yang diilhami bio boleh digunakan. Algoritma Firefly adalah algoritma semulajadi terinspirasi yang boleh menyelesaikan masalah pengoptimuman. Algoritma Firefly terdiri beberapa parameter yang boleh dikawal, yang ini boleh mempengaruhi prestasi algoritma. Kajian ini memberi tumpuan kepada variasi tetapan parameter untuk mengetahui prestasi algoritma. Untuk mengesahkan dan menguji prestasi algoritma Firefly, fungsi penanda aras boleh digunakan. Seting parameter yang bervariasi adalah bilangan dimensi, bilangan populasi dan bilangan lelaran. Apabila jumlah fireflies bertambah, ia akan bergerak ke arah titik optimum. Algoritma akan melakukan lebih baik pada dimensi yang lebih rendah berbanding dengan dimensi yang lebih tinggi. Sistem manipulator fleksibel (FMS) digunakan sebagai aplikasi untuk mengetahui prestasi algoritma. Algoritma Firefly digunakan untuk menala pengawal PID dengan menggunakan kriteria prestasi. Parameter pengawal PID akan mempengaruhi prestasi sistem manipulator fleksibel (FMS). Kaedah analisis digunakan untuk menganalisis prestasi sudut hub FMS. Terdapat 3 syarat yang ditetapkan untuk menganalisis prestasi sudut hub bagi setiap kriteria prestasi. Kriteria prestasi yang paling sesuai sebagai fungsi objektif Algoritma Firefly untuk menyesuaikan pengawal PID telah dipilih.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

FA	-	Firefly Algorithm
FMS	-	Flexible Manipulator System

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter discusses briefly the background of firefly algorithm. It is followed by an introduction of motivation, problem statements and objectives of this project. This chapter also present the scopes of this project.

1.2 Research Background

Nowadays, there are many problems which can be formulated to the optimization problem. The main aim of the optimization method is to determine the best possible solution to solve the mathematical function, called objective or fitness functions. Optimization is a process to obtain or find an optimal result to solve the problems such as improve the performance, efficiency and output of the system. The studies about optimization algorithm have been increasing in the field of applied mathematics, computer science and engineering. In the real world, the optimization problems become more complex and difficult, such as multi-objective, discrete, multimodal, nonlinear and many more.

Deterministic and stochastic methods are the major group of optimization methods. Deterministic method is a method that always produces the same set of solution when it is started under the same initial condition. However, the stochastic method is a method that consists one or more components of randomness. For stochastic method, these algorithms may not generate the same optimal solution when it is started with same initial condition and the same problem. For example, the algorithm, making random walks in the search space. More than half of this method is considered as meta-heuristic.

The majority of stochastic methods are considered as meta-heuristic. Meta-heuristic are guiding the iterative generation process to find a heuristic that provided an efficiently near-optimal solutions. These algorithms are usually inspired by phenomena or the behavior of nature and animal. The advantage of the meta-heuristic method are easy to develop, convergence rate to the global optimum is faster, larger range of application, and many more.

Firefly Algorithm is a stochastic and meta-heuristic optimization algorithm that introduced by Xin-She Yang in 2008. The abdomen of the firefly produce a light called bioluminescence, which function to attract and communicated with other fireflies. However, Firefly algorithm is inspired by the flashing pattern of the firefly which known as nature inspired algorithm. Firefly Algorithm consist random parameter, which it will make a random movement in the search space. By using random search movement method, the global best values will be easier to achieve, so Firefly algorithm is more efficiency and more accuracy.

1.3 Motivation

Firefly Algorithm is stochastic and meta-heuristic algorithm, which have the random properties. This means the algorithm will not get the same set of optimal solution although the algorithm is starting with the same initial condition. So Firefly Algorithm is very interesting when the selectivity of the optimal solution and know how the algorithm work in order to get the optimal solution.

Firefly Algorithm can used the algorithm to tuning the parameters of the proportional-integral-derivative (PID) controller. By using firefly algorithm, the tuning process will converge faster and speed up the tuning of these parameters. This means the algorithm will get the optimal solution with less number of iteration.

The efficiency to find the optimal solution and others advantage of the Firefly Algorithm, which is the motivation to do this project. This algorithm is bio-inspired algorithm which behaviour will be easier to study.

1.4 Problem Statement

In the era of industry 4.0, there are many problems that required to be solve. In order to solve the problem efficiently, the efficient methods in finding the best possible solution need to be established. There are several methods to find the optimal solution. Some of the method is difficult to use, for example in the tuning PID controller. When the problems become complex and getting bigger (large scale such as national system grid), its more difficult to find the best optimal point. Some of the methods required long processing time. So use optimization to find the best possible solution is more effective.

The process of manual tuning is too hard and it can use an optimization algorithm to tune the PID control. It is much better because algorithm can generate thousands of solution and find the best possible solution automatically.

1.5 Objective

In this project, there are 3 objectives going to be achieved:

- To investigate the performance of firefly algorithm with different parameter setting using different numerical benchmark functions.
- To investigate the performance of firefly algorithm with the control of flexible manipulated system (FMS).
- To analyse the performance criteria (error) that suitable for the flexible manipulated system (FMS).

1.6 Scope

The scope of the project is using Matlab 2016a to simulate this project. The specification of the computer is Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz with 4.00GB RAM. After select the suitable algorithm, the pseudo code of the Firefly algorithm can be find. Editing the pseudo code is required to let the algorithm run automatically though 30 independent runs.

The initial parameter setting must be set before simulate the algorithm. Firefly Algorithm is started to simulate with different parameters setting. The parameter is set to be varying are the number of iteration, number of dimension and number of population. After the result been simulated, the data must be analysed and simulated the convergence plot.

The simulation for the Flexible Manipulated System is used to know the performance of the algorithms. The Firefly Algorithm will tune the PID controller by using the performance criteria in order to get good response of hug angle of the flexible arm. The analysis method been introduce to analyse the performance of the Flexible Manipulator System (FMS).

1.7 Summary

This chapter is to introduce the project about the background, the problem statement, the scope of the project, the motivation and objective of the project. It is the detail and the work that been done in this project.

The next chapter will discuss the understanding of the project which are the basic concept of the algorithm, the related work that been done about the project and others.

CHAPTER 2

LITERATURE REVIEW

2.1 Optimization

Optimization is a process to determine the best possible solution to solve the problems. As an example of optimization, travelling salesman problem is a problem to find the shortest possible way that going to each town and come back to the origin town. According to N.F. Johari et al. [1] , optimization problem is a computational problem which to find the best solution among all the possible solution. In order to solve the problem, more iteration have to carry out. Throughout every iteration, the solution will move toward the optimal point. One of the major inspiring problem in the field of research is optimization [2].

The optimization problems are classified into few categories, which are continuous, combinatorial, constrained, unconstrained, single and multi-objective problems. The combinatorial optimization problem is a discrete optimization problem where the optimal solution can be determine from a finite set of solutions. However, the continuous optimization problem where the optimal solution is any value within the range of values and normally is a real numbers. The constrained optimization problems are the more constraint on the variables but unconstrained optimization problems does not have limitation on the values of the parameters. Multi-objective optimization problem is finding the optimal solution to achieve more than one objective function simultaneously but single objective optimization problem is concern on finding the solution for an objective function.

The optimization techniques can divide into two types, which are exact and heuristic. The exact optimization technique surely will found the optimal solution and it able doing great for many problems. However, exact optimization techniques needed very high computational costs when solving the complex problems or problems with a very large number of parameters.

There are many real-world problems are complex problem. In order to solve this complex problem, the heuristic optimization techniques can be used. Heuristic optimization techniques uncertain that can found the optimal solution, but it will get very close to the best solution. For high complexity problems, heuristic techniques it will used.

The heuristic techniques that uncertain will get the optimal solution is because the solution is stuck at the local minimum point which unable to find the desired global minimum. M. Jamil stated that the function will stuck at the local minimum point which the algorithm not search effectively because the poor designed of the exploration process of an algorithm [3].

There are optimization algorithm that can be used to solve the problems such as Firefly Algorithm, Particle Swarm Optimization, Artificial Ant Colony Optimization, Artificial Bee Colony Optimization and many more. According to the No Free Lunch Theorem [4] and [5], for certain classes of problems, some of the algorithm is perform excellent but it will outperformed for others classes of problems. This means the algorithm cannot solve every problem, but suitable for certain problem.

The optimization algorithms can be categories into bio-inspired algorithm and natural inspired algorithm. The natural inspired algorithm is the algorithm that inspired by the natural phenomena or natural material, however bio-inspired algorithm is the algorithm that more related to the fields of biology, mathematics and computer science.

2.2 Firefly Algorithm

R. Francisco, M. Costa and A. Rocha stated that the firefly algorithm has developed by Yang in 2008 which consider as one of the swarm intelligence algorithm stated [6]. Firefly Algorithm is a bio-inspired optimization algorithm, which inspired by the flashing pattern of a group of fireflies in nature. Fireflies are nocturnal which will only active at night. The firefly emits light from their abdomen, which called bioluminescence. Adult firefly is able to emit high and discrete flashes by controlling their bioluminescence. Each firefly has their own flashing pattern where their main purpose of flashing is act as a signal to attach other fireflies and to warn the potential predators. The firefly uses flashing pattern to communicate with other fireflies.

Firefly algorithm is able to solve the optimization problems because it is a stochastic, nature-inspired and meta-heuristic algorithm. Stochastic method means it to have one or more components of randomness when searching for a set of solution. This means Firefly Algorithm may not produce the same solutions each time it's run from the same problems.

Majority of stochastic method considers as meta-heuristic. Heuristic means a proceeding to discover or find a solution by trial and error. Meta-heuristic means high level searching process that may generate a sufficiently good solution for the optimization problem where influenced by satisfaction between randomization and local search. This means that the fireflies can make random movement in the search area. The light intensity of the firefly is variation in the search region is related to the encoded objective function.

In order to get the optimum solution, the firefly algorithm basically has three main principles that have to consider:

1. All fireflies are assumed to be unisex and they will attach by each other regardless their sex.
2. The attractiveness of the firefly is determined by their light intensity, which their relation is proportional. The firefly with higher light intensity will attach the firefly with low light intensity. The firefly will move randomly if no other brighter than a particular firefly. When the distance between the fireflies increase, the light intensity will decrease.
3. The light intensity of firefly is depending on the value of the objective function of the optimization problem.

For Firefly Algorithm, the key ideas are the brightness and the attractiveness of the fireflies. The brightness is assumed that it can be define by the objective function. When the light intensity is varying and the change of the attractiveness will be formulated.

L. Zhang, L. Liu, X. Yang et al mention that the advantage of Firefly Algorithm is it can provide better convergence rate and strong exploration ability [7]. This means this algorithm have more faster convergence the problems to get the optimal solution. However Firefly Algorithm is better in local search but some of the solution may stuck in the local optimal point which do not search globally well. Besides that, Firefly Algorithm is very useful to tune these parameters automatically by the control randomization which can solve the problems effectively in real-world application. Firefly Algorithm is able to solve nonlinear, multimodal and global optimization problems where the complexity or difficulty of the problem does not affect the performance of the algorithm. According to [8], the researcher stated that the execution time for each iteration is higher than other algorithm.

2.3 Basic Understanding of Firefly Algorithm

The fireflies are placed in random position when the algorithm started. The location is determined by the values of parameter for the objective function. By random walk attraction between the fireflies will generated a new solution. The location of the fireflies must be considers after they compare their brightness among each other. When the firefly move toward the brighter firefly encountered, the distance between the fireflies, r must be calculated. The distance between fireflies i and j, r_{ij} can be calculated as Cartesian distance which given as

$$r_{ij} = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{2-1}$$

where D is number of dimension, x_i, x_j are the solution position for fireflies i and j and r_{ij} is the distance between fireflies.

After knowing the distance between the fireflies, the light intensity can be calculated. According to the inverse square law, the light intensity is decreasing when the distance between the fireflies i and j, r_{ij} increased. The light intensity $I(r)$ is calculated as

$$I(r) = \frac{Io}{{r_{ij}}^2} \tag{2-2}$$

where I_o is the initial light intensity of the firefly.

The light intensity is the attractiveness between the fireflies. The light intensity coefficient, γ can control the light intensity. Since the brightness is directly related to attractiveness, it can be assume that I_o=β_o, where β_o is the initial attractiveness. The attractiveness β(r) between the firefly i and the firefly j is given as

$$\beta(r) = \beta_o e^{-\gamma r_{ij}^2} \tag{2-3}$$

The attractiveness β(r) and the randomization parameter, αϵ_i can determined the movement of the firefly toward the firefly. The random coefficient, α is a parameter that can control the amount of randomness which normally in range of [0,1]. In order to create a random distance, fireflies can move with a uniform distribution, which the movement can be either forward or backward. The movement from the firefly i at position x_i, to the another more brighter (attractive) firefly j at position x_j which given as

$$x_i = \beta_o e^{-\gamma r_{ij}^2}\left(x_i - x_j\right) + \alpha\varepsilon_i \tag{2-4}$$

If the β_o=0, it means that the fireflies can be simple random movement, and α=0 corresponds to no randomness.

After a firefly move to a new position, the light intensity and attractiveness are updated by evaluating the objective function in the new position. The new evaluation of brightness is compared with the best found. The firefly will move to the new position if the new position produces higher brightness and attractiveness, which the position will become the new best. If the firefly passes through the location and it not better than any other found, firefly will remain in the current location and the best location is still recorded.

2.4 Studies on Firefly Algorithm

Firefly Algorithm is powerful local search with the objective function but it may get a local optimums which does not search globally well [8]. This is because some of the objective function has many local minimum and a global minimum. Generally, local search gets the high precision solution and global search helps the algorithm converge to the area quickly in the practical optimization problem, which stated by C. Liu, F. Gao, and N. Jin [9].

Firefly Algorithm is a powerful method to solve complicated optimization problem and non-deterministic polynomial-hard (NP-hard) problems [1]. According to [10], Firefly Algorithm able to use in such as continuous optimization, constraint optimization, combinatorial optimization, and multi-objective optimization problems. Firefly Algorithm is suitable to solve the continuous mathematical function because the behaviour of the algorithm is simple.

C. Liu, F. Gao, and N. Jin stated that the random coefficient, α and light absorption coefficient, γ are two important parameters when the location updating [9]. With higher algorithm convergence speed, the smaller the light absorption coefficient, the attraction between fireflies is larger. With lower algorithm convergence speed, the larger the random coefficient, the random motion range of fireflies is larger.

H. Kasdirin stated that by modifying the parameters of the algorithm can improve the search capability and improve the convergence rate of of the Firefly Algorithm [11]. In the journal of [8] stated that two importance point in the Firefly Algorithm which are formulation of the attractiveness and the variation in the brightness.

S. Agarwal, A. P. Singh and N. Anand conclude that Firefly Algorithm tends to be better compare to Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO), especially the function having multi-peaks [8]. The researchers stated that there is no effect to FA when solving the complexity or difficult level of the functions.

The research of “Simplified firefly algorithm for 2D key-point search” is in simplified firefly algorithm (SFA) for 2D-image key-point search is done by [12]. The researchers concluded that the application of simplified firefly algorithm is easily and reliably to get the key-areas in examined 2D images. The simplified firefly algorithm is efficient when it apply for searching the area in the 2D image such as human face appearance (hair or eyes), human posture nature element or building (dark constructions, trees, or nature phenomena like shades). The researcher stated that the calculations performed by SFA are simple because of the comparing of research result and performance.

The widely application of Firefly Algorithm for optimization problems is focused on [1]. Based on the analysis, there are two major areas that widely used Firefly Algorithm to find the optimum solution for the problem, which are Engineering and Computer Science. Besides that, the number of single Firefly Algorithm used is twice the hybrid Firefly Algorithm to solve the optimization problem. The researchers stated that the application of hybrid Firefly Algorithm is better than single Firefly Algorithm because hybrid algorithm performed better result and other techniques improve the performance and processing time of Firefly Algorithm.

According to the X.S. Yang and X.He [13], stated that Firefly Algorithm has two major advantages compare with other algorithm which is the ability of dealing with multimodality and automatically subdivision. The attractiveness of Firefly Algorithm decreases when distance increases, which will lead the whole population automatically subdivide into a few groups. Each group will find around the local optimum or each mode. Throughout this searching of local optimum, it can find the best optimum solution. If the population size is greater than the number of modes, the subdivision lets the fireflies to find all optimal solution simultaneously. The whole population subdivide into few groups which given average distance. If γ=0, there are no subdivide in the whole population. The automatic subdivide of FA make it suitable for solving highly non-linear, multimodal optimization problems.

One part of the research of “Firefly algorithm: recent advances and applications” [13] is the parameter setting which stated the recommended range for the parameter value. For the population size is 15 to 100 for the most application and the best range of population size is 25 to 40. The researcher stated that δ normally is using 0.95 to 0.97. For most application, the parameter of initial attractiveness, β_o is suggested to be equal to 1, where β is used to control the attractiveness.

2.5 Benchmark Functions

In order to test the efficiency, reliability and the validation of an algorithm, a standard benchmark or test function can be used stated by M.Jamil [3]. Most of the researcher is using benchmark function to validate and compare the performance of the algorithms. According to E.Alba [14] stated that, when the test involve function optimization, most of the researches compare different algorithm on a large test set. The benchmark functions are using to evaluate the algorithm in form of the convergence rate, the general performance, the precision and the robustness.

Example of real-world problems are come from the field of engineering, physics, mathematic and chemistry and others. However, these problems may consist complicated algebraic or other expressions which cause the problems are hard to manipulate. For the benchmark function, it can be used to investigate the behaviour of the algorithm.

There have different structure of the optimization problem, some of the problem is unimodal and some of it is multimodal. The unimodal benchmark function is only have a global minimum and do not have local minimum. The multimodal problem is the objective functions that consists many local minimum point. The global minimum is the true optimal solution for the optimization problems, which is the lowest point of the objective function. Figure 2-1 clearly show the position of the local and minimum point.

The functions with multimodal problem are used to test the ability of an algorithm to escape from the local minimum. For many algorithms, those multimodal problems are the most difficult class of problem to find the global minimum. This is because some of the optimal solution will stuck at the local minimum point of the multimodal function.

According to M.Jamil [3], for all the optimization algorithms, the dimensionality may significant limit for the highly non-linear problems. This means the complexity of the problems is depend on the number of dimension. When the number of dimension increases, the size of the search space will be increases.

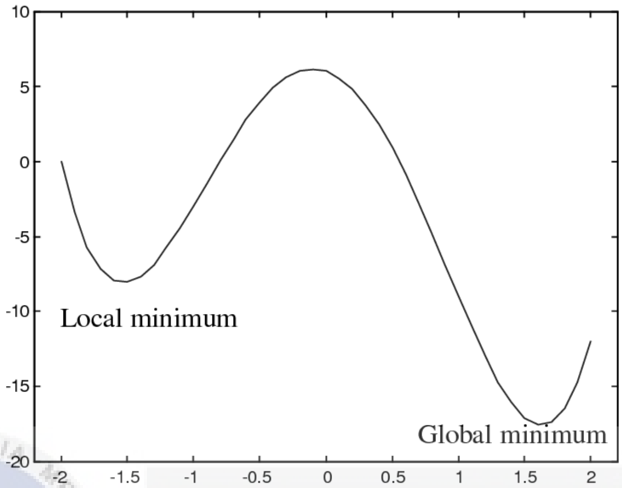


Figure 2-1: The diagram of showing the location of the global and local minimum point

2.6 Flexible Manipulator System (FMS)

In the era of fourth Industrial Revolution (IR 4.0), there are increasing of the number of applications in flexible manipulator system. The flexible manipulator system such as robotic arm, are mostly apply in the field of automation and manufacturing industries.

A flexible object is difficult to manipulate and control, which become a challenging problem because it is hard to predict and the dynamic are complex and highly non-linear. Besides that, the flexible object can stretch and compress cause it more complex and contact issues. This is just like most of the flexible object able to move in all direction and bend in 360 degree.

The problems on the vibration due to flexibility, the precision of the position, the difficulty in obtaining accuracy of the flexible manipulator system are increased. There have increase in number of the research in the field flexible system to solve the problems and increase the performance of the system.

P.Zhang and Y.C.Li doing the research to approximate the vibration of flexible payload, the finite element method is used and the adaptive sliding mode control law is designed to prevent the disturbance of the vibration [15]. According to H.Li, Z.Yang and T.Huang [16] stated that the finite element analysis and the optimization are available to parallel the robot with the flexible links.

In the Figure 2-2, the schematic diagram of the flexible manipulator system show that the FMS is consists of motor, flexible arm, encoder, tachometer and accelerometer. The motor of the FMS is to make the flexible arm can be move and the M_p is a payload. This system consist measuring devices which is encoder, tachnometer and accelerometer in order to measure the performance of the system. The shaft encoder is a sensor to measure the hub angle, the tachometer is to measure velocity and accelerometer is measure acceleration.

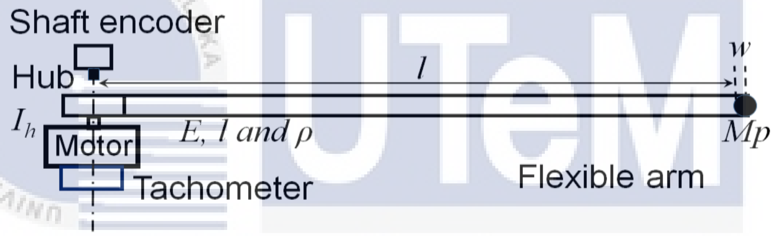


Figure 2-2: The schematic representation diagram of FMS.

According to the research of “Dynamic Modelling and Control of a Flexible Manipulator” [17], the researcher say that the advantages of the flexible robot manipulators are it required less material, lower energy consumption, lighter in weight, required smaller actuators, higher manipulation speed, less overall cost, higher payload to robot weight ratio, more manoeuvrable and transportable. The advantages are the factor in increasing various of applications of flexible manipulators including space exploration and hazardous environments.

In this project, this FMS been used as the application of the Firefly Algorithm. The hub angle of the FMS is analysed and controlled in this project. The controller is used to control the hub angle of FMS which to stabilize the system.

2.7 Controller

There have few types of controller but it mainly about P, PD, PI and PID controllers. For P controller, it is mostly apply for first order system which to stabilize the unstable system. The P controller mostly is used to reduce the steady state error of the system. The PI controller also used to decrease the steady state error but the speed of the system is not considered. The PI controller that without derivative action is less responsible to the real and slower to response compare to PID controller. However,

For this FMS application, the PID controller is been used and the PID controller clearly discussed. Figure 2-3 show the control system block diagram of FMS. The block diagram clearly show that the system is using Firefly Algorithm to tune the PID controller. This is because this application is required to improve the stability and reduce the steady state error in order to get the better performance of flexible manipulated system.

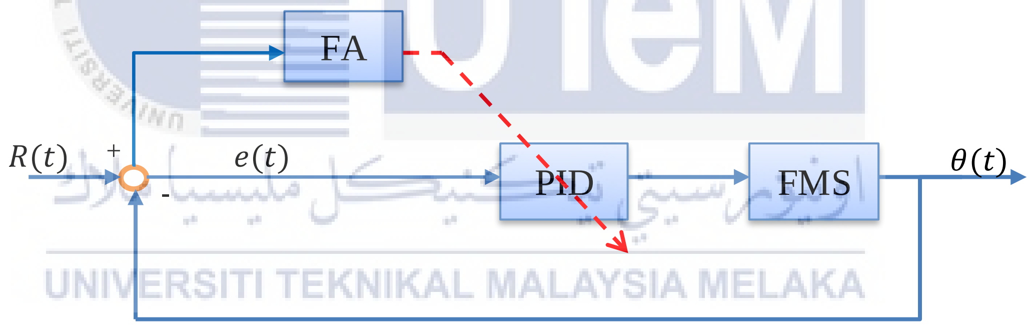


Figure 2-3: The control system block diagram of flexible manipulator system (FMS).

Proportional-integral-derivative PID controller is close loop feedback control system that usually used in various of applications that required continuously modulated control. By using PID controller, the effectiveness and the efficiency of the application can be improved. A PID controller is using to minimize the error from the feedback system. The PID controller consists of 3 parameters, which are proportional gain, Kp, integral gain, Ki, and derivative gain, Kd.

When tuning the PID controller, some effect increasing the parameter of the PID controller should be understand. The proportional gain, Kp is increase, it will reduce the rise time, but increase the overshoot and the steady state error, which is the transient response of the system. D. Corrigan stated that by introducing the integral gain, Ki can reduce the steady state error, but it will make the transient response become worse [18]. In order to improve the transient response, the system must consider the derivative gain, Kd where it not affect the steady state response. The derivative gain will affect the stability because it sensitive to the noise.

G. Lin and G. Liu state that the gains of the PID controller should be identify in order to prosecute the PID controller [19]. The gains of the PID controller can be determined by tuning the PID controller. Tuning the PID controller is to optimize the control function to the system. There are many method can be tuning the PID controller such as Ziegler-Nichols method, Tyreus-Luyben method, Damped oscillation method, C-H-R method, Cohen and Coon method, Fertik method, Ciancone-Marline method, IMC method and minimum error criteria stated by M. Shahrokhi and A. Zomorrodi. [20]

According to A. Eldin A. Awouda and R. Mamat, finding the optimal PID controller parameters can fulfill most of the system qualification such as ensure high system response, maintain robustness, achieve good load disturbances rejection and lower the overshoot. [21] This mean that by tuning the PID controller can find the optimal gains of PID controller that can improve the performance of the control system.

2.8 Performance Measurement

Accuracy means how close the measurement value to the true value. The performance assessment is to measure the effectiveness and the efficiency of the performance of the algorithm. A performance measurement should use to test the accuracy before applied to a decision-making scenario. Measuring with using of performance measurement is a process to validate the performance of an algorithm. The performance measurement that been used for the benchmark function such as standard deviation, average maximum and minimum. The standard deviation is to test the robustness of the algorithm.

For the application, tuning the PID controller is required to improve the performance of the application. Some of the articles call this performance measurement as performance criteria. D. Maiti, A. Acharya and A. Konar stated that their PID controller design method is based on minimizing ITAE criterion [22]. B. Doicin, M. Popescu and C. Patrascioiu stated that the integral criteria is used to analyze the dynamic performance of the automatic control system [23].

In this project, the performance measurement that be used for the FMS application are mean square error (MSE), root mean square error (RMSE),integral time-weighted absolute error (ITAE), integral square error (ISE) and integral absolute error (IAE). This can measure the rise time, settling time, steady state error, and overshoot of the hub angle, which is the performance of the system.

For the analysis of the performance criteria that generated the better result in hub angle of FMS, the performance is measure based on the transient response and the steady state condition. This can be show that Firefly Algorithm more suitable to be used which types of performance criteria to tune the PID controller for the FMS application.

CHAPTER 3

METHODOLOGY

3.1 Overview

In the chapter is briefly discussing about the method to undergo the experiment. In order to be shows it clearly and easier to understand, the flow chart are used. Figure 3-1 is the overall flow chart of the project.

The overall flow chart is divide into 3 part which to fulfill the 3 objectives of this project. Part 1, part 2 and part 3 are to achieve the objective 1, objective 2 and objective 3 respectively.

The first part of the flow chart is focusing on the verification of the algorithm by varying the parameters, the second part is more focus on the application of the algorithm. For the third part of the flow chart is to analyse the most suitable performance criteria as the objective when Firefly Algorithm tuned the PID controller in order to get the best performance of FMS.

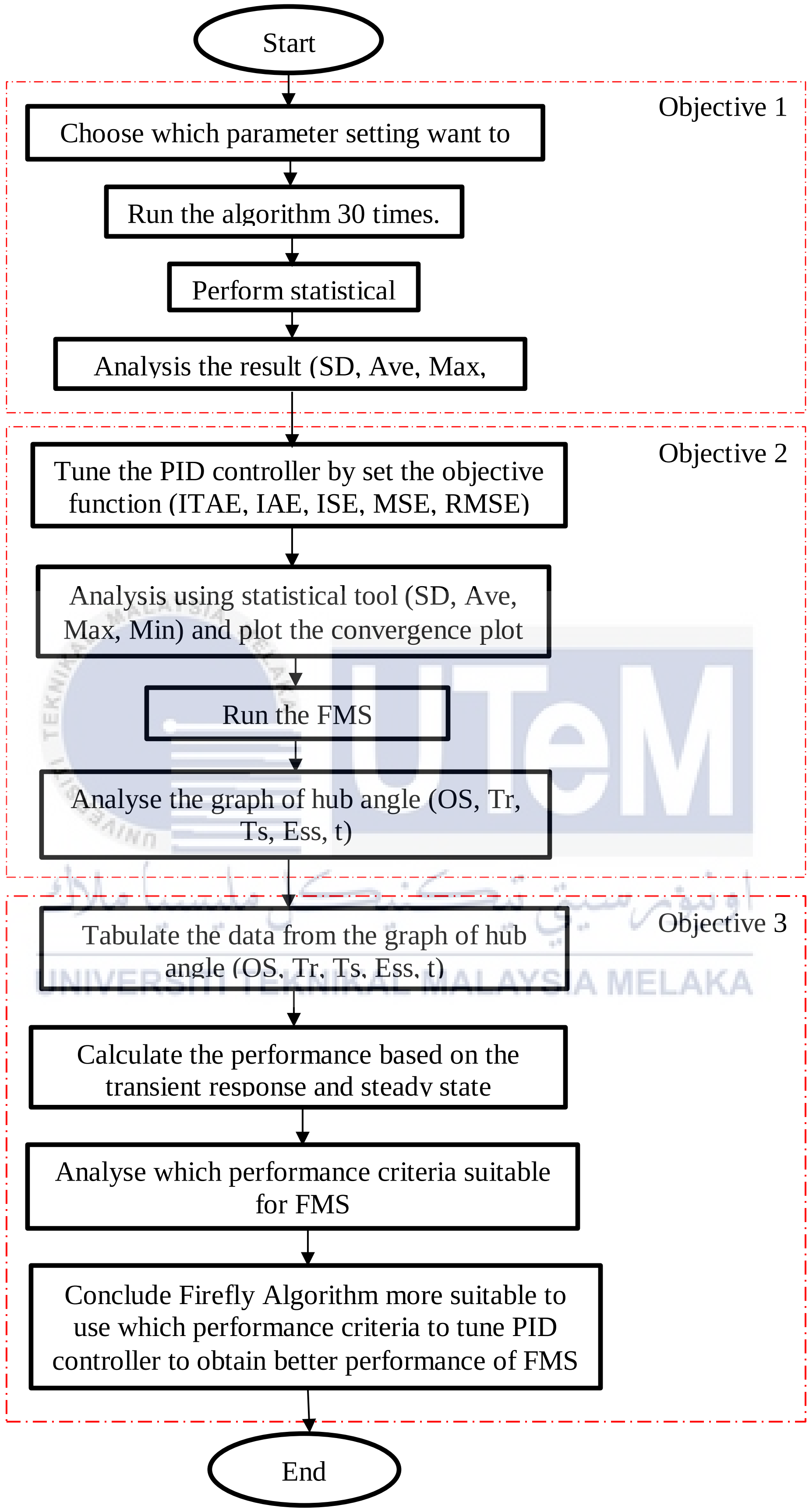


Figure 3-1: The overall flow chart to achieve 3 objective in this project.

3.2 Method to Achieve Objective 1

Objective 1: To investigate the performance of firefly algorithm with different parameter setting using different numerical benchmark functions.

Figure 3-1 part 1 is the flow chart to achieve the objective 1, which is varying the parameter setting of the Firefly Algorithm. This experiment is using Matlab to simulate the Firefly Algorithm.

The experiment is starting by choosing the parameter setting that want to vary. The parameter setting that been chosen are number of population, number of dimension and number of iteration.

The Figure 3-2 is the flow chart of the experiment carried out. Before simulated the algorithm, the parameters should be initialize first. The algorithm runs with 14 benchmark functions. The experiment been run through by 3 different parameter setting with 30 independent runs. The algorithm will repeat the run in order to reach 30 times or independent runs. This is to get the more accurate solution, and this can be determine the robustness of the algorithm.

After run the algorithm, the result is recorded and the data will be analyzed. The result is analyzing by using statistical tool such as standard deviation (SD), average, maximum and minimum of the last iteration. The convergence plot is been plotted and analyzed in order to show the performance of the algorithm clearly.

The benchmark functions also known as the objective function of the experiment. The benchmark functions are to verify the performance of the algorithm. By varying the parameter, it can be clearly show the performance of the algorithm. This is very useful when determine the parameter setting for a specific problem based on the requirement of the problem.

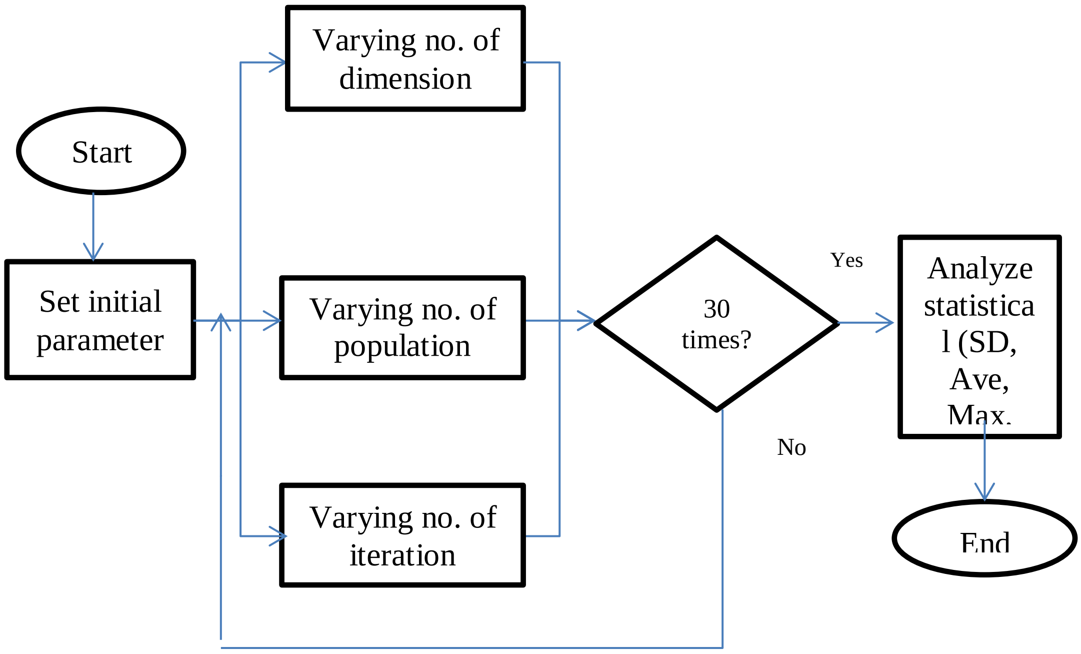


Figure 3-2: The flow chart when carry out the experiment of variation parameter setting.

3.3 Method to Achieve Objective 2

Objective 2: To investigate the performance of firefly algorithm with the control of flexible manipulated system (FMS).

In order to achieve objective 2, the flow chart in Figure 3-1 part 2 is developed. The objective 2 is focus on the application that been applied to this algorithm which is flexible manipulated system (FMS). The Matlab 2016a is used to carry out the experiment to achieve objective 2. The Figure 3-3 is the flow chart that show the detail to achieve the objective 2.

This experiment is starting with set the objective function which are the formula of Integral Time-weighted Absolute Error (ITAE), Integral Absolute Error (IAE), Integral Square Error (ISE), Mean Square Error (MSE) and Root Mean Square Error (RMSE). Table 3-1 show the formula of the performance criteria that also can know as objective function.

After set the objective function, the Firefly Algorithm is used to tune the PID controller. The simulated data will be analyze by using statistical tools such as convergence plot, standard deviation, average, minimum and maximum of the last iteration.

This is to get the best possible of the PID parameters. PID controller have 3 parameter which are Kp, Ki and Kd. Those parameter are important and influenced the performance of the flexible manipulator system.

After the parameter is tuned, the parameter is used to simulate the flexible manipulator system (FMS). In this experiment, the hub angle of the FMS is been controlled. The encoder of the FMS is work as a sensor to detect the hub angle of the flexible arm.

The graph of the hub angle of FMS will display and the step response characteristic of the graph been analyzed. The characteristic of the graph been analyzed is such as overshoot, steady state error and the rise time. The characteristic of the graph is showing the performance of the flexible arm. After the system is finished with a performance criterion, the system is repeated with another performance criteria.

Throughout the steps in the flow chart, the result is analyzed and the performance of the application can be determined. This is to learn how the algorithm will affect the performance of the application.

Table 3-1: The formula of the performance criteria as the objective function for Firefly Algorithm to tune the PID controller.

Performance Criteria	Formula
Integral Time-weighted Absolute Error	$ITAE = \int t\|e\| \ dt$
Integral Absolute Error	$IAE = \int \|e\| \ dt$
Integral Square Error	$ISE = \int e^2 \ dt$
Mean Square Error	$MSE = \frac{1}{n}\sum e^2$
Root Mean Square Error	$RMSE = \sqrt{\frac{1}{n}\sum e^2}$

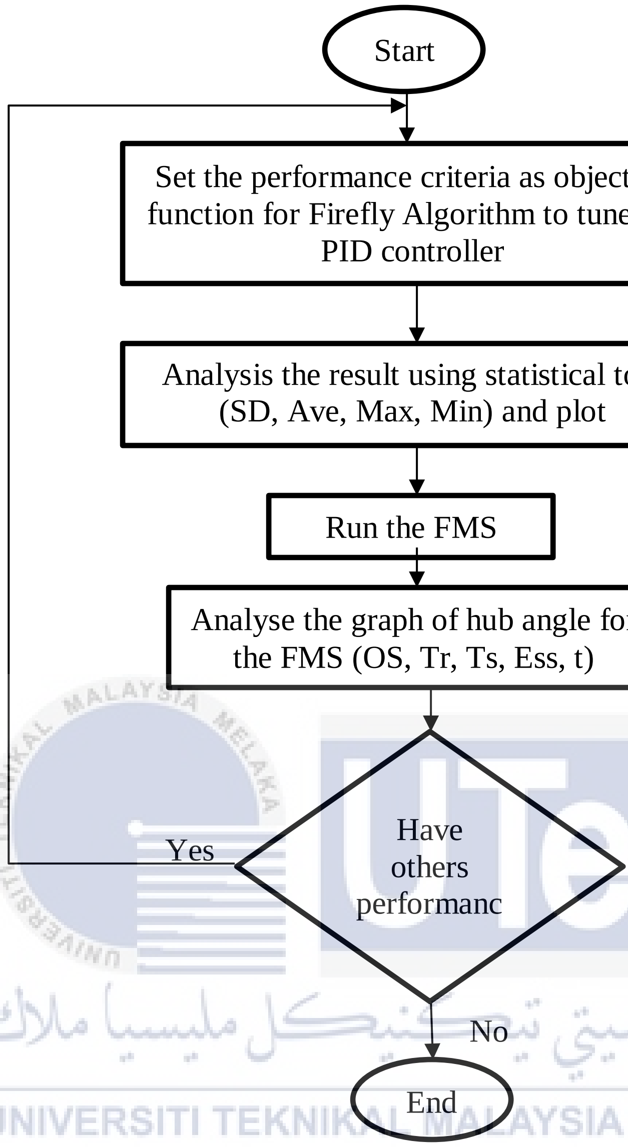


Figure 3-3: The flow chart to achieve objective 2.

3.4 Method to Achieve Objective 3

Objective 3: To analyse the performance criteria that suitable for the flexible manipulated system (FMS).

In order to achieve objective 3, the flow chart in Figure 3-1 part 3 is developed. The objective 3 is focus on how to analysis the suitable performance criteria as objective function when tuning the PID controller by Firefly Algorithm that have the best performance of FMS which show in Figure 3-4.

This experiment is starting with the tabulated the data from the analysis of the performance of hub angle. The data been analysed is based on steady state condition (overshoot and steady state error) and transient response (rise time and settling time).

Next, the method of analysis is introduced. There have 3 analysis condition which are, measure of the performance with equal importance of transient response and steady state condition, measure of the performance with more importance of transient response condition and measure of the performance with more importance of steady state condition. The analysis test condition calculated based on the equation 4.2. Each condition have different value of the coefficient. The performance of the hub angle can be calculated based on the equation and the coefficent.

Those 3 analysis test condition been tabulated need to analyse the most suitable performance criteria. The analyse of the performance is based on the calculation of FMS, where the lower the number of total calculated, the better the performance of the FMS. After the analysis is done, it have to make a conclusion that find out the most suitable performance criteria for FMS.

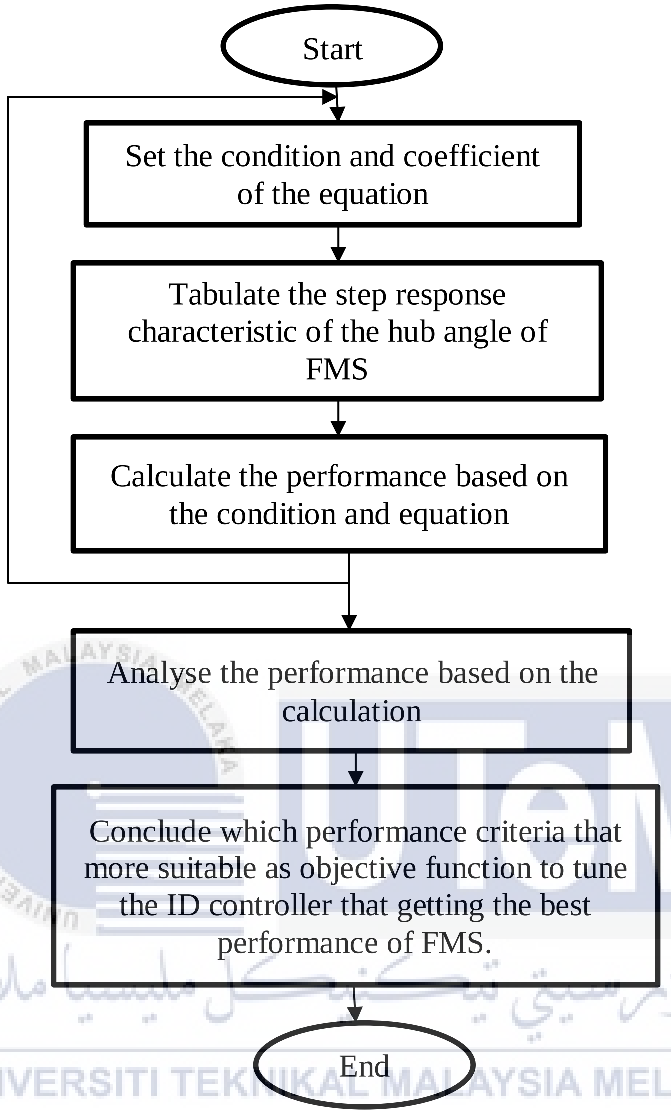


Figure 3-4: The flow chart to achieve objective 3.

3.5 Summary

This chapter has discussed about the method to carry out this project. The methods are developed to achieve the objective and solve the problem of this project. Those methods will be carry out when doing this project.

The next chapter will present the result and the discussion that been simulated by using Matlab.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Overview

This chapter is discussing about the result of the project. The result is using Matlab to run the experiment. The result is represented by using tables and figures. The results are to investigate the performance of Firefly Algorithm with benchmark functions and applications.

This result is simulated by using Matlab to test the performance of the Firefly Algorithm. Throughout the experiment, the result will show clear and it will be analyses. All the experiment will be run with the parameter that been set at the Table 4-1. In order to verify the performance of Firefly Algorithm, different structure of benchmark functions is used. Some of the benchmark functions is unimodal which consist a global minimum point and no local minimum point. Other than that, the multimodal benchmark function is used and the multimodal consists many local minimum, where the solution may stuck at the local minimum. The Table 4-2 shows the 14 benchmark functions been used in this experiment and its range of search space and Table 4-3 shows the properties of the benchmark functions.

The result is analysed based on the standard deviation (SD), average (Ave), the best (Max) and the worst (Min) of the last iteration. The reason analyze the last iteration is because the last iteration is the best solution that the Firefly Algorithm get. The standard deviation is used to show the robustness of the last iteration for each independent run. The robustness means how often to get the same value. If the standard deviation is equal to zero which mean throughout the run, the algorithm gets the same optimal point. The convergence plot is a plot based on the average result of 30 independent runs. This is because Firefly Algorithm is a meta-heuristic algorithm which will generate different solution even the experiment started with the same parameters. The average of the last iteration also considers as the accuracy of the optimal solution.

Table 4-1: The fixed parameter when run all the experiment.

Parameter	Value
Light Intensity Coefficient, γ	1
Random Coefficient, α	0.2
Attraction Coefficient, β	0.2
Mutation Coefficient, δ	0.98

Table 4-2: The benchmark functions used and their range of the search space.

Function	$\boldsymbol{f(x)}$	**Range**
Ackley, f_1(x)	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)) + 20 + \exp(1)$	[-32.768,32.768]
Dixon-price, f_2(x)	$f(x) = (x_i - 1)^2 + \sum_{i=1}^{d} i(2x_i^2 - x_{i-1})^2$	[-10,10]
Griewank, f_3(x)	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]
Levy, f_4(x)	$f(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + sin^2(2\pi w_d)]$ where $w_i = 1 + \frac{x_i - 1}{4}$ for all $i = 1, \dots, d$	[-10,10]
Rastrigin, f_5(x)	$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	[-5.12,5.12]
Rosenbrock, f_6(x)	$f(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-5,10]
Sphere, f_7(x)	$f(x) = \sum_{i=1}^{d} x_i^2$	[-5.12,5.12]
Sum-squares, f_8(x)	$f(x) = \sum_{i=1}^{d} ix_i^2$	[-10,10]
Rotated hyper-ellipsoid, f_9(x)	$f(x) = \sum_{i=1}^{d}\sum_{j=1}^{d} x_i^2$	[-65.536,65.536]
Schwefel, f_{10}(x)	$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]
Alpine, f_{11}	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	[0,10]
Powell, f_{12}	$f(x) = \sum_{i=1}^{d/4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	[-4,5]

Function	$f(x)$	Range
Sum of Different Power, f_{13}	$f(x) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	[-1,1]
Xin-She Yang, f_{14}	$f(x) = \sum_{i=1}^{n} \varepsilon_i \lvert x_i \rvert^{i}$	[-5,5]

Table 4-3: The benchmark functions and its properties.

No	Benchmark Function	Type	Properties
f_1	Ackley	Multimodal	Nearly flat outer region and large hole at the centre
f_2	Dixon-price	Unimodal	No local minimum, continuous and convex
f_3	Griewank	Multimodal	Many widespread local minima
f_4	Levy	Multimodal	Complex with many local minima
f_5	Rastrigin	Multimodal	Location of the minimum are regularly distributed
f_6	Rosenbrock	Unimodal	Convergence to minimum is difficult
f_7	Sphere	Unimodal	No local minimum, continuous and convex
f_8	Sum-squares	Unimodal	No local minimum, continuous and convex
f_9	Rotated hyper-ellipsoid	Unimodal	No local minimum, continuous and convex
f_{10}	Schwefel	Multimodal	Complex with many local minima
f_{11}	Alpine	Multimodal	Not convex, differentiable and non-separable
f_{12}	Powell	Unimodal	No local minimum, continuous and convex
f_{13}	Sum of Different Power	Unimodal	No local minimum, continuous and convex
f_{14}	Xin-She Yang	Multimodal	Not convex. Non-differentiable and separable.

4.2 Investigate the Variation of Parameter Setting

In order to know the effect of changing parameter of Firefly Algorithm, there are some benchmark functions can be used. The parameter setting has to choose and determine properly so that can get the optimum results.

The benchmark functions used are Sphere, Rosenbrock, Levy and Sum-square. From Figure 4-1, the 3D model of benchmark functions of Sphere and Sum-squares are showing that there are an unimodal which do not have local minimum and only consists a global minimum. However, the benchmark function of Levy and Rosenbrock are more complex which consists of many local minimum, where can be seen at the 3D model in Figure 4-2. There have possible of the optimal solution stuck at the local minimum.

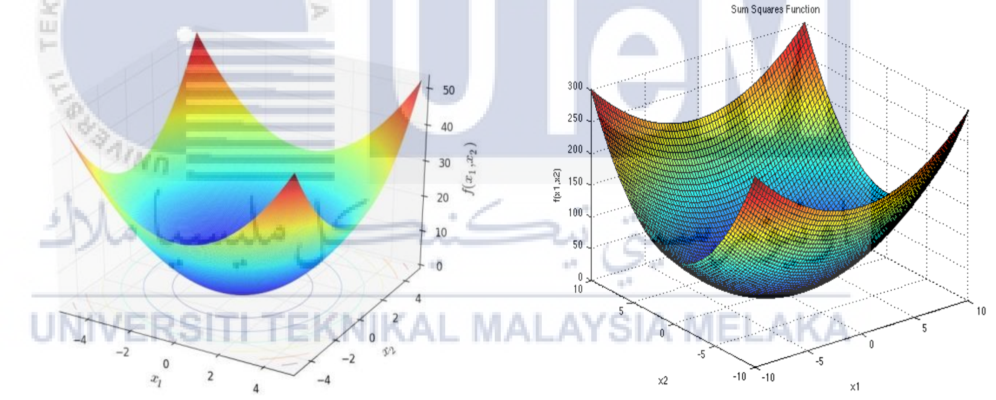


Figure 4-1: The 3D model of the benchmark function of Sphere and Sum-squares.

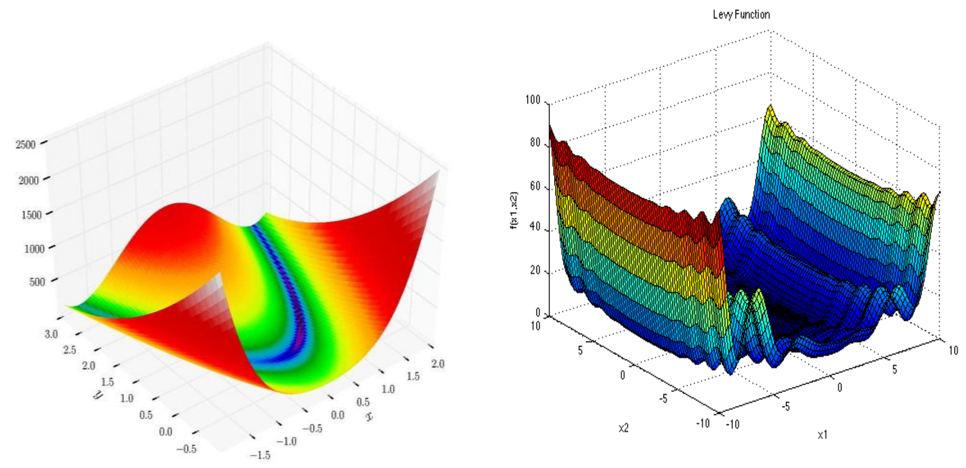


Figure 4-2: The 3D model of the benchmark functions of Rosenbrock and Levy.

For the variation of dimension is using 10 benchmark functions. It is using benchmark functions to verify the performance of the algorithm is because there is a fair comparison where most of the researcher is using benchmark functions and most of the parameters are set to be same.

There are 3 parameters that have been changed which are the number of population, the number of dimensions and the number of iteration. The algorithm runs with 1000 iterations for each independently run. After running by using Matlab, the result must be analyzed.

4.2.1 Varying the Number of Population

The number of population also considers as the number of fireflies. The number of fireflies is there are put how many fireflies in the searching area. In this experiment, the numbers of population being used are 5, 10, 25, 50, 100 and 150 in order to do comparison. As expected, when the number of fireflies increase, the more evaluation, it will move toward the best global minimum point.

From Figure 4-3, it can be seen that for benchmark functions Levy and Rosenbrock, the number of population, p=5 and p=10 are straight line. The straight line and not going down to the global minimum because the solution is stuck at local minimum However, from Figure 4-3, the benchmark functions Sphere and Sum-squares do not have this problem because there are unimodal which does not have any local minimum point.

By comparing all the line in the convergence plot, it can be seen that the higher the number of population, the lower the objective value will get. From this situation, it can be concluded that the higher the number of fireflies in the search space, the higher possible the fireflies will lead on the optimal point. It is because the more fireflies in the search space, the more the evaluation, higher possible can find the optimal point.

For the experiment by using benchmark functions, some of the parameter is fixed to know the performance of Firefly Algorithm which shown in Table 4-4. Those parameters are fixed to see the performance of the algorithm when only a parameter is varied. The Table 4-5 shows the data of the different number of population.

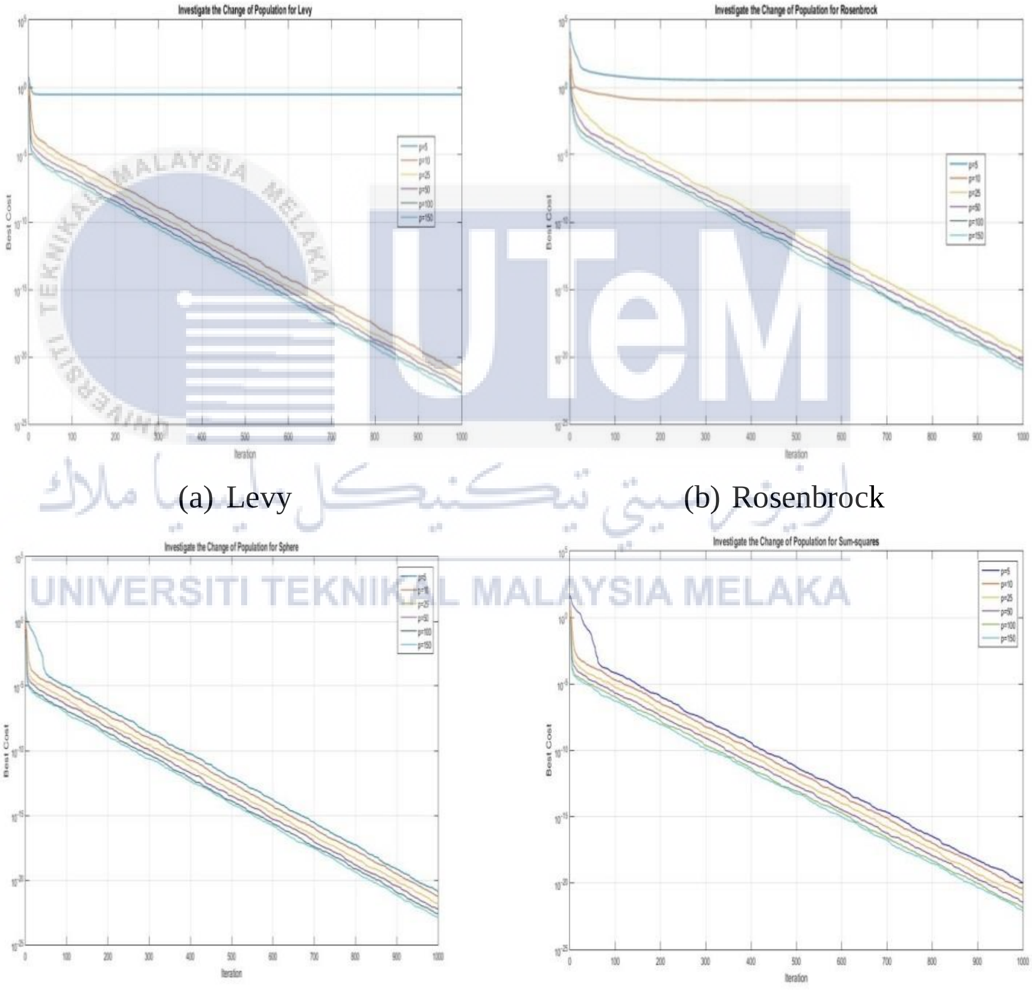
Table 4-4: The fixed parameter when run the experiment of varying number of population.

Parameter	Value
Number of dimension	3
Number of independent run	30
Number of iteration	1000

Table 4-5: The data of the varying number of population with different benchmark functions.

Function	No. of population	SD	Ave	Max(Worst)	Min(Best)
Levy	P=5	1.71E-23	2.43E-23	8.45E-23	2.26E-24
	P=10	9.65E-24	2.00E-23	3.86E-23	3.01E-24
	P=25	6.67E-01	2.93E-01	2.49E-00	1.39E-22
	P=50	3.57E-22	5.94E-22	1.35E-21	5.21E-24
	P=100	5.16E-23	8.37E-23	2.45E-22	1.70E-23
	P=150	2.46E-23	3.85E-23	9.11E-23	5.26E-24
Rosenbrock	P=5	1.58E-21	2.45E-21	7.34E-21	1.72E-22
	P=10	6.99E-22	1.04E-21	3.18E-21	8.34E-24
	P=25	7.44E-00	3.54E-00	39.62E-00	1.35E-05
	P=50	0.56E-00	1.09E-01	3.11E-00	4.11E-20
	P=100	3.40E-21	5.01E-21	1.67E-20	1.53E-22
	P=150	1.81E-21	2.92E-21	7.06E-21	1.68E-22
Sphere	P=5	1.33E-23	2.47E-23	4.85E-23	4.94E-24
	P=10	7.46E-24	1.18E-23	3.63E-23	1.25E-24

Function	No. of population	SD	Ave	Max(Worst)	Min(Best)
	P=25	8.63E-22	1.23E-21	3.67E-21	9.21E-23
	P=50	3.11E-22	5.62E-22	1.24E-21	8.76E-23
	P=100	4.32E-23	5.63E-23	1.86E-22	1.04E-23
	P=150	1.99E-23	3.31E-23	7.56E-23	3.43E-24
Sum-squares	P=5	1.01E-22	1.37E-22	3.90E-22	8.01E-24
	P=10	5.98E-23	8.57E-23	2.54E-22	8.36E-24
	P=25	6.53E-21	1.07E-20	2.59E-20	5.33E-22
	P=50	2.35E-21	4.18E-21	1.18E-20	8.60E-22
	P=100	2.99E-22	3.85E-22	1.13E-21	1.71E-23
	P=150	1.56E-22	2.17E-22	6.01E-22	1.28E-23



(a) Levy (b) Rosenbrock

(c) Sphere (d) Sum-squares

Figure 4-3: The convergence plot of varying number of population.

4.2.2 Varying the Number of Iteration

In order to show the comparison clearly, the number of iterations been chosen are 100, 200, 500, 800 and 1000. When the numbers of iteration increase, the evaluations increase, the final solution will lead on the best global minimum point.

Throughout the experiment, its the line is going down toward to achieve the optimal point when the number of iteration is 100 and 1000. This means the less the number of iteration, the final solution is greater, and the result is less accurate. The number of evaluations is result of number of iterations multiply with number of fireflies. In this experiment, the number of fireflies is fixed, so the number of evaluation depends on the number of iteration. When the number of evaluation increase, the possibility of achieving the global minimum point is increased.

For the experiment by using benchmark functions, some of the parameter is fixed to know the performance of Firefly Algorithm which shown in Table 4-6. The results of the varying of number of iteration are shown in the Figure 4-4 and the data of varying number of iteration is shown in Table 4-7.

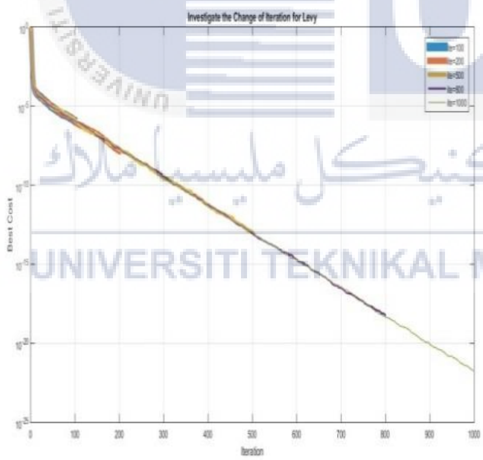
Table 4-6: The fixed parameter when run the experiment of varying number of iterations.

Parameter	Value
Number of dimension	3
Number of independent run	30
Number of firefly	25

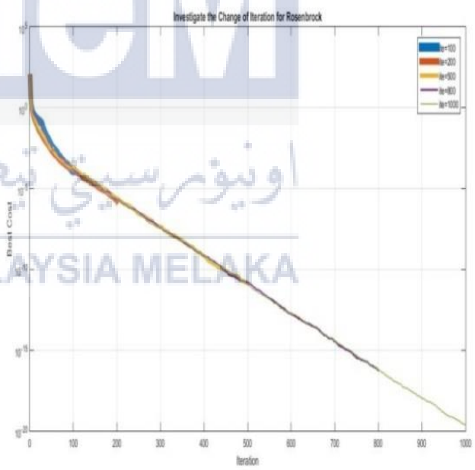
Table 4-7: The data of varying number of iteration with the benchmark functions.

Function	No. of iteration	SD	Ave	Max(Worst)	Min(Best)
Levy	I=100	7.35E-07	1.12E-06	2.33E-06	5.27E-08
	I=200	1.29E-08	1.43E-08	7.52E-08	4.26E-09
	I=500	5.85E-14	1.05E-13	2.39E-13	1.56E-14
	I=800	4.36E-19	6.04E-19	2.09E-18	3.06E-20
	I=1000	1.03E-22	1.73E-22	4.28E-22	3.69E-23

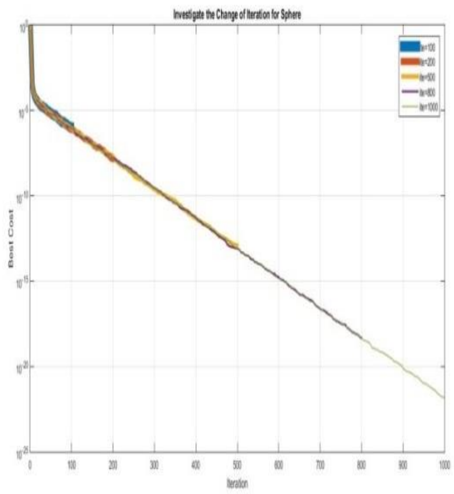
Function	No. of iteration	SD	Ave	Max(Worst)	Min(Best)
Rosenbrock	I=100	8.64E-05	1.26E-04	3.20E-04	1.62E-05
	I=200	1.50E-06	1.69E-06	6.26E-06	2.36E-08
	I=500	1.21E-11	1.39E-11	5.19E-11	2.35E-12
	I=800	4.20E-17	5.68E-17	1.58E-16	1.64E-18
	I=1000	1.64E-20	2.39E-20	6.85E-20	1.61E-21
Sphere	I=100	6.08E-07	9.97E-07	2.61E-06	1.92E-07
	I=200	1.11E-08	1.52E-08	4.95E-08	2.02E-09
	I=500	5.73E-14	1.19E-13	3.02E-13	2.18E-14
	I=800	3.04E-19	4.69E-19	1.15E-18	7.85E-22
	I=1000	1.03E-22	1.41E-22	4.27E-22	2.25E-24
Sum-squares	I=100	4.47E-06	6.62E-06	1.77E-05	1.81E-07
	I=200	1.02E-07	1.31E-07	4.03E-07	1.52E-08
	I=500	3.25E-13	6.28E-13	1.30E-12	1.07E-13
	I=800	2.18E-18	3.67E-18	9.04E-18	5.07E-20
	I=1000	7.74E-22	1.16E-21	2.86E-21	1.26E-22



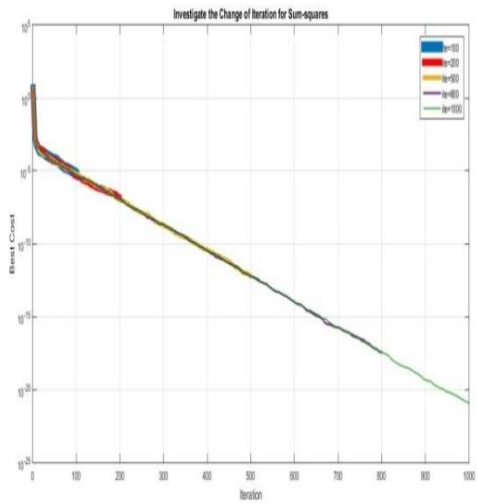
(a) Levy



(b) Rosenbrock



(c) Sphere



(d) Sum-squares

Figure 4-4: The convergence plot of varying number of iterations.

4.2.3 Varying the Number of Dimention

In this experiment, there are fourteen single objective optimization benchmark functions to test and validate the algorithm. The number of dimension been used to do the comparison is 2, 3, 10, 20, and 50. The number of dimension also considers as the complexity of the algorithm. As expected, when the numbers of dimension increase, the complexity of the algorithm increase, the harder to find the global optimum point. The complexity means how complex the algorithm to solve the problem.

From Figure 4-5, some of the benchmark functions with higher dimension are getting a straight line which do not move toward the global minimum point and stuck at the local minimum point. The result showed that 2 dimension have better solution compare to others dimension. It shows that the higher dimension problem is more complex to solve.

For the experiment by using benchmark functions, some of the parameter is fixed to know the performance of Firefly Algorithm which shown in Table 4-8. When simulated the algorithm, the range of each benchmark function must be set. The Table 4-9 shows the simulated data when varying the number of dimension with 10 different benchmark functions.

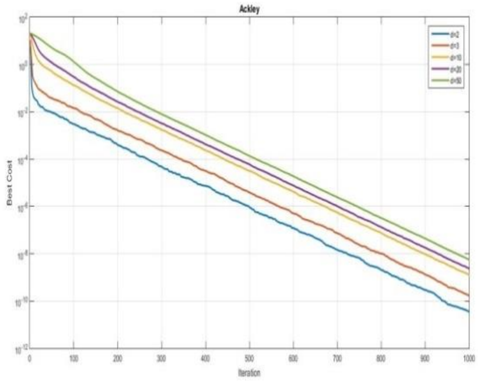
Table 4-8: The fixed parameter when run the experiment of varying number of dimension.

Parameter	Value
Number of iteration	1000
Number of independent run	30
Number of firefly	25

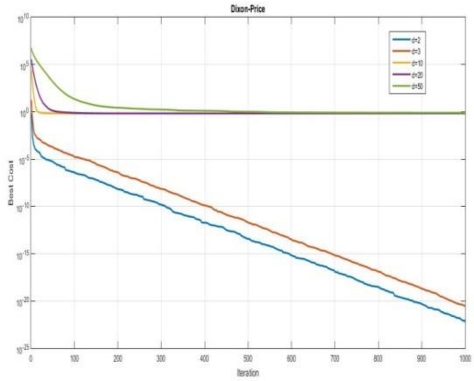
Table 4-9: The simulated data of arying number of dimension with different benchmark functions.

Function	**No. of dimension**	**SD**	**Ave**	**Max(Worst)**	**Min(Best)**
Ackley, $f_1(x)$	D=2	1.89E-11	3.56E-11	7.86E-11	6.70E-12
	D=3	6.19E-11	1.67E-10	3.13E-10	5.10E-11
	D=10	1.69E-10	1.28E-09	1.53E-09	9.84E-10
	D=20	2.18E-10	2.38E-09	2.72E-09	1.74E-09
	D=50	4.44E-10	5.66E-09	6.42E-09	4.64E-09
Dixon-price, $f_2(x)$	D=2	4.66E-23	6.43E-23	1.94E-22	2.70E-24
	D=3	2.61E-21	3.03E-21	1.27E-20	3.25E-22
	D=10	0	6.67E-01	6.67E-01	6.67E-01
	D=20	0	6.67E-01	6.67E-01	6.67E-01
	D=50	5.25E-01	7.64E-01	3.59E-00	6.67E-01
Griewank, $f_3(x)$	D=2	0	0	0	0
	D=3	4.72E-03	4.01E-03	1.23E-02	0
	D=10	2.94 E-02	6.43 E-02	1.55E-01	1.97E-02
	D=20	1.00 E-02	1.12 E-02	3.45 E-02	0
	D=50	6.28 E-03	3.78 E-03	2.21 E-02	5.55E-16
Levy, $f_4(x)$	D=2	5.21E-24	5.13E-24	2.20E-23	1.06E-25
	D=3	1.03E-22	1.73E-22	4.28E-22	3.69E-23
	D=10	1.07E-20	4.36E-20	6.23E-20	2.34E-20
	D=20	7.14 E-01	3.64 E-01	3.27 E-00	2.35E-19
	D=50	7.52 E-00	14.44 E-00	42.52 E-00	3.09 E-00
Rastrigin, $f_5(x)$	D=2	0	0	0	0
	D=3	5.23 E-01	2.98 E-01	1.99 E-00	0
	D=10	3.37 E-00	10.35 E-00	10.35 E-00	2.98 E-00
	D=20	9.98 E-00	28.92 E-00	49.75 E-00	12.93 E-00
	D=50	40.64 E-00	155.78 E-00	233.81 E-00	59.70 E-00
Rosenbrock, $f_6(x)$	D=2	8.31E-23	1.02E-22	3.25E-22	6.50E-24
	D=3	1.64E-20	2.39E-20	6.85E-20	1.61E-21
	D=10	1.31 E-00	0.74 E-00	4.16 E-00	2.06E-05
	D=20	4.01 E-00	8.00 E-00	13.89 E-00	7.36 E-02
	D=50	36.86 E-00	70.31 E-00	148.37 E-00	1.53 E-02
Sphere, $f_7(x)$	D=2	4.85E-24	4.94E-24	2.20E-23	2.18E-25
	D=3	1.03E-22	1.41E-22	4.27E-22	2.25E-24
	D=10	6.82E-21	2.39E-20	3.79E-20	1.16E-20
	D=20	3.27E-20	1.69E-19	2.27E-19	1.09E-19
	D=50	3.43E-19	2.42E-18	3.09E-18	1.57E-18
	D=2	1.42E-23	1.72E-23	5.00E-23	5.05E-25
	D=3	7.74E-22	1.16E-21	2.86E-21	1.26E-22

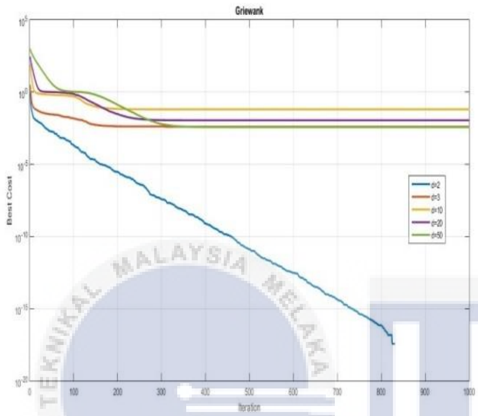
Function	No. of dimension	SD	Ave	Max(Worst)	Min(Best)
Sum-squares, $f_8(x)$	D=10	1.25E-19	4.53E-19	6.93E-19	1.69E-19
	D=20	1.02E-18	6.31E-18	8.73E-18	4.85E-18
	D=50	3.33E-17	2.54E-16	3.12E-16	1.75E-16
Rotated hyper-ellipsoid, $f_9(x)$	D=2	1.33E-21	1.37E-21	6.25E-21	2.96E-23
	D=3	1.94E-20	3.68E-20	7.83E-20	4.40E-21
	D=10	5.78E-18	2.09E-17	3.17E-17	1.01E-17
	D=20	4.22E-17	2.80E-16	3.61E-16	2.08E-16
	D=50	1.30E-15	1.01E-14	1.22E-14	7.66E-15
Schwefel, $f_{10}(x)$	D=2	35.53 E-00	-826.12 E-00	-719.53 E-00	-837.97 E-00
	D=3	66.18 E-00	-1205.63 E-00	-1020.07 E-00	-1256.95 E-00
	D=10	276.84 E-00	-3405.44 E-00	-2882.42 E-00	-3834.51 E-00
	D=20	449.00 E-00	-6020.08 E-00	-4899.14 E-00	-6855.11 E-00
	D=50	813.52 E-00	-13630.05 E-00	-12296.73 E-00	-15218.39 E-00
Alpine, $f_{11}(x)$	D=2	0	0	0	0
	D=3	2.65E-02	7.36E-15	1.66E-13	0
	D=10	7.17E-11	5.68E-11	3.12E-10	0
	D=20	5.63E-10	9.65E-10	2.45E-09	1.66E-10
	D=50	5.51E-09	1.27E-08	2.86E-08	5.45E-09
Powell, $f_{12}(x)$	D=2	0	0	0	0
	D=3	0	0	0	0
	D=10	5.40E-08	4.92E-08	2.11E-07	8.32E-10
	D=20	1.85E-06	4.13E-08	7.76E-06	6.02E-07
	D=50	1.35E-04	5.56E-04	7.72E-04	2.93E-04
Sum of Different Power, $f_{13}(x)$	D=2	2.42E-28	1.56E-28	1.11E-27	4.71E-30
	D=3	3.90E-27	1.85E-27	2.01E-26	3.39E-30
	D=10	4.40E-25	3.03E-25	1.37E-24	5.48E-27
	D=20	3.42E-23	1.37E-23	1.73E-22	1.19E-24
	D=50	1.04E-19	1.97E-20	5.69E-19	5.13E-23
Xin-She Yang, $f_{14}(x)$	D=2	2.43E-15	4.08E-15	1.05E-14	3.39E-16
	D=3	7.73E-15	8.48E-15	3.09E-14	8.92E-16
	D=10	2.33E-12	6.39E-13	1.28E-11	1.80E-14
	D=20	3.99E-03	8.20E-04	2.18E-02	1.17E-13
	D=50	5.73E06	1.05E6	3.14E07	5.67E-06



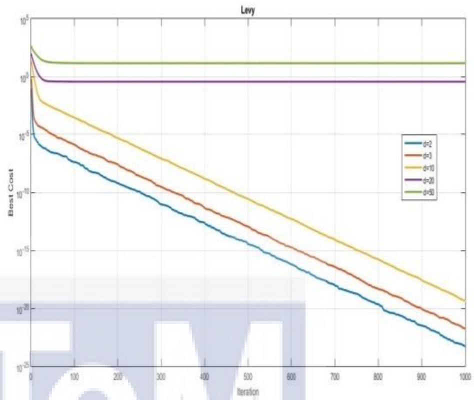
(a) Ackley, $f_1(x)$



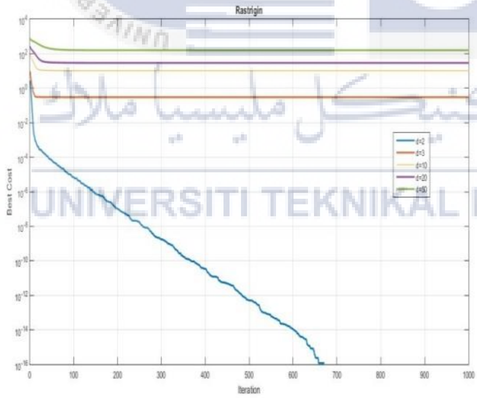
(b) Dixon-price, $f_2(x)$



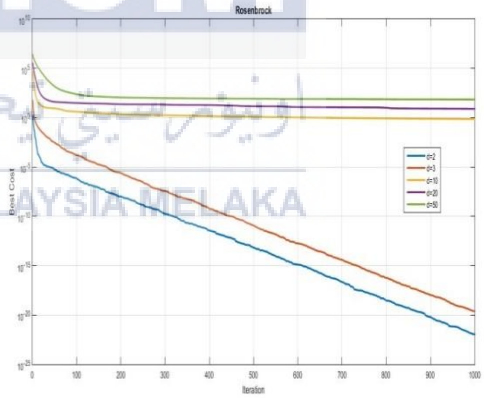
(c) Griewank, $f_3(x)$



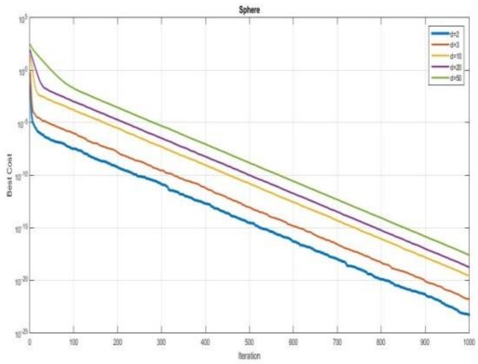
(d) Levy, $f_4(x)$



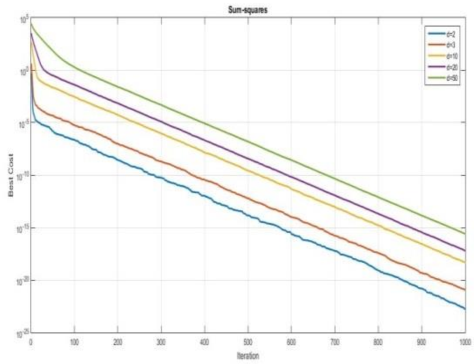
(e) Rastrigin, $f_5(x)$



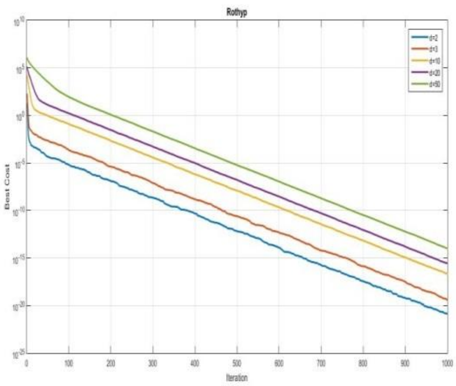
(f) Rosenbrock, $f_6(x)$



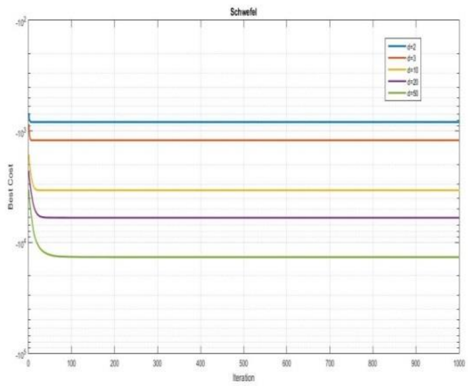
(g) Sphere, $f_7(x)$



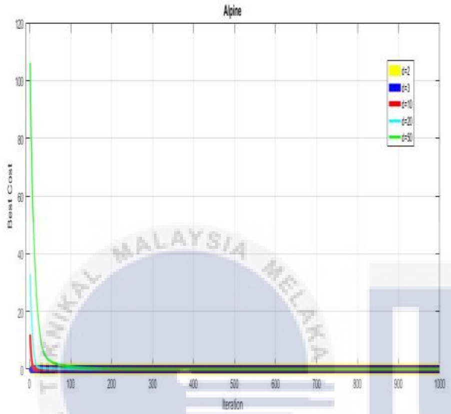
(h) Sum-squares, $f_8(x)$



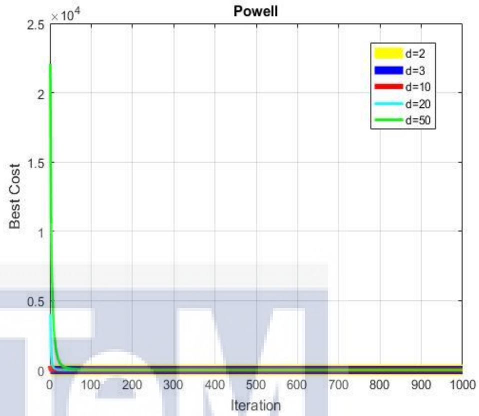
(i) Rotated hyper-ellipsoid, $f_9(x)$



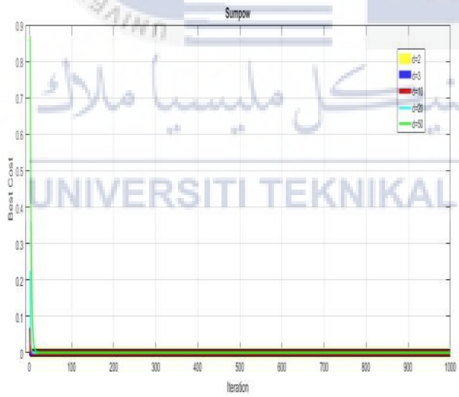
(j) Schwefel, $f_{10}(x)$



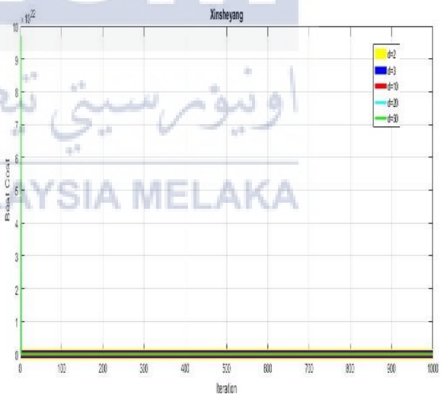
(k) Alpine, $f_{11}(x)$



(l) Powell, $f_{12}(x)$



(m) Sum of Different Power, $f_{13}(x)$



(n) Xin-She Yang, $f_{14}(x)$

Figure 4-5: The convergence plot for 14 benchmark functions.

4.3 Flexible Manipulator System (FMS)

This part of the experiment is to know the performance of Firefly Algorithm by applying to a flexible manipulator system (FMS). The Firefly algorithm will tune the parameter of the proportional-integral-derivative (PID) controller, which will affect the performance of the FMS.

In order to reduce the vibration motion at the end of the beam of the flexible manipulated system during operation, a PID controller is required. Figure 4.6 is shown the Firefly Algorithm is tuned the PID controller to get the optimal PID parameter. The PID controller is 3 dimensions which consists 3 static error constant which are proportional gain, Kp, integral gain, Ki and derivative gain, Kd. The Firefly Algorithm is using the performance criteria to tune the PID controller. Before the tuning process, the initial parameter setting of Firefly Algorithm should be set as the Table 4-10.

Figure 2-2 shows the schematic diagram of FMS which showing the structure of the system. The motor makes the flexible arm move and the shaft encoder is as a sensor to detect the hub angle of the movement of the flexible arm. Throughout this experiment is to control the hub angle of flexible arm. Tuning the PID controller is to find the optimal PID parameter that can improve the performance for the hub angle of flexible arm.

The Firefly Algorithm tunes the parameter of the PID controller for 30 runs with 100 iterations and finds the optimal solution. When the PID parameter is applying to the FMS, the hub angle is controlled. The number of firefly or number of population set as 2, 10 and 20. Different number of firefly will give different performance. By using different number of firefly, the performance for the hub angle of FMS can be compared.

Firefly Algorithm is using 5 performance criteria to tune the PID controller, which are Integral Time-weighted of Absolute Error (ITAE), Integral Absolute Error (IAE), Integral Squared Error (ISE), Mean Square Error (MSE) and Root Mean Square Error (RMSE). The performance criteria used to verify the performance of the Firefly Algorithm on the Flexible Manipulated System (FMS).

Table 4-10: The initial parameter setting when simulated FMS.

No. of independent run	30
No. of iteration	100
No. of dimension	3
Search space size	[0 2]

Table 4-11 shows the parameters of the PID controller for each performance criteria. This is the result of Firefly Algorithm tune the PID controller of the FMS with using the performance criteria. The parameters are substitute in the PID controller to test the performance of the FMS. Different value of the PID controller parameter will get the different performance of FMS. This result is the value that simulated for 30 runs and each run have 100 iterations. There have enough number of evaluation for the algorithm to converge to the optimal solution.

The performance of the FMS will be compare by using different performance criteria. The performance of FMS been measure is the transient response of the hub angle. This result can be seen that for the ITAE, IAE and ISE, the number of the population is 10 and 15 get the same parameters which mean have the same performance. This means that the algorithm is converging to the optimal solution so there will get the same result.

Table 4-11: The PID controller parameter for each performance criteria.

Parameter		Proportional gain, Kp	Integral gain, Ki	Derivative gain, Kd
ITAE	P=2	1.4490	0.9955	0.2999
	P=10	2.0000	0.0000	0.5539
	P=15	2.0000	0.0000	0.5538
IAE	P=2	1.3990	0.5306	1.2896
	P=10	2.0000	0.0000	0.5488
	P=15	2.0000	0.0000	0.5488
ISE	P=2	0.8683	0.4249	0.3074
	P=10	2.0000	0.0000	0.4814
	P=15	2.0000	0.0000	0.4815
MSE	P=2	1.5966	1.4974	0.3930
	P=10	1.2707	0.0000	0.4357
	P=15	1.4513	0.0000	0.5889
RMSE	P=2	1.0338	0.6603	0.9298
	P=10	1.7930	2.0000	0.3877
	P=15	1.3606	1.4662	0.3323

Table 4-12 shows the performance of the FMS based on the performance criteria. The performance being consider is the transient response for the hub angle of FMS.

Rise time is the time taken for a system to reach from 10% to 90% of the final value. For the rise time of the system, IAE with number of population equal to 2 took slightly longer time compare with others performance criteria. This means that IAE with smaller number of population is more suitable for the application that required slower starting time. However, RMSE with 10 fireflies has the shortest rise time, which is 0.286s. This means for the application that required slower starting time can use RMSE with 10 number of fireflies.

Settling time means the time taken for the system to rise and stay within 2.5% of the steady state. The faster the settling time, the more stable the system. The result show that less number of population will cause it take longer time to steady state. IAE with more number of population show the better result, which have the shortest settling time (1.042s) compare to others criteria. However, RMSE have the longer settling time, which not suitable for the application because it take long time to steady state. So, for the application that required high stability, the IAE with more number of population can be used.

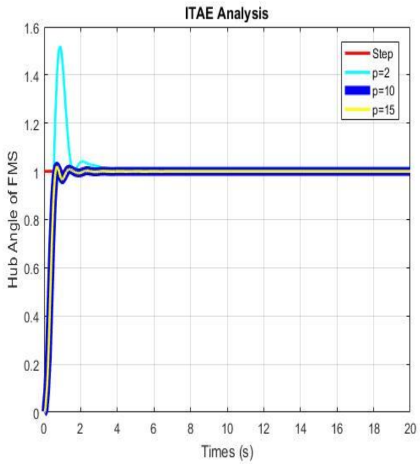
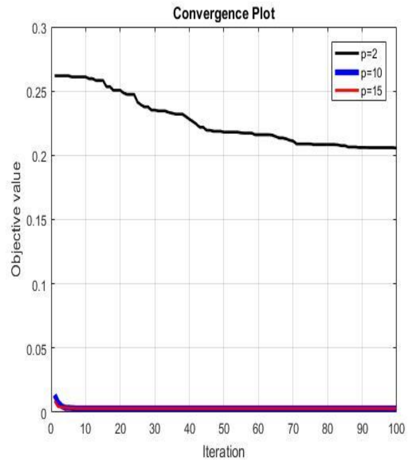
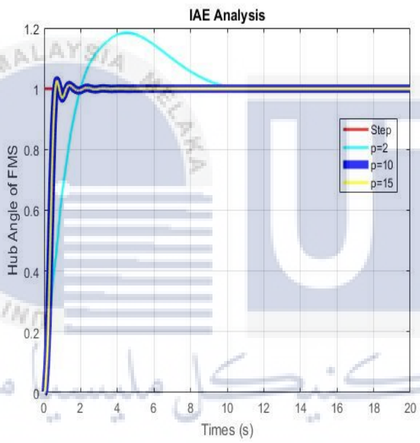
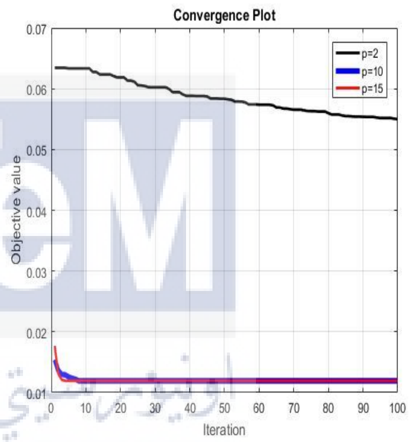
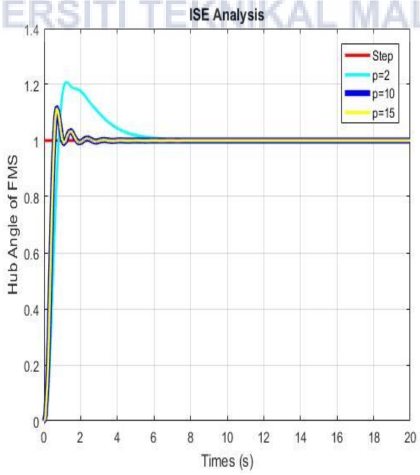
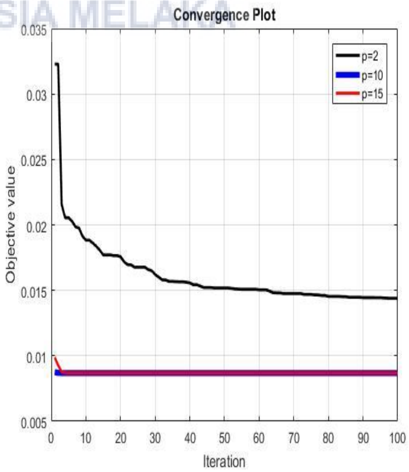
The steady state error is the difference between the input and the output of the system. There is better result when the steady state error is zero, which means there are not difference between input and output. All the result getting show zero steady state error except for the RMSE when number of population is 2. This is because of only 2 fireflies is more difficult to obtain the optimal solution. FMS have PID controller, the error is reduced, so the steady state error is zero.

Overshoot means a system that exceeding the target or the final (steady state) value. For the overshoot, most of the applications prefer to have lower overshoot. MSE with higher number of population have lower overshoot compare to others performance criteria. However, RMSE have the highest overshoot, which more than 50%.

Table 4-12: The step response characteristic of the hub angle of the FMS for each performance criteria.

Parameter		Rise time, T_r (s)	Settling time, T_s (s)	Steady state error, e_{ss} (%)	Overshoot. OS (%)
ITAE	P=2	0.317	2.706	0.000	53.077
	P=10	0.370	1.090	0.000	2.577
	P=15	0.370	1.090	0.000	2.577
IAE	P=2	1.408	9.013	0.000	18.452
	P=10	0.367	**1.042**	0.000	2.577
	P=15	0.367	**1.042**	0.000	2.577
ISE	P=2	0.492	4.639	0.000	21.341
	P=10	0.338	1.569	0.000	11.798
	P=15	0.338	1.569	0.000	11.798
MSE	P=2	0.314	2.252	0.000	46.324
	P=10	0.540	1.504	0.000	**0.298**
	P=15	1.150	2.460	0.000	0.500
RMSE	P=2	1.119	9.779	0.020	25.949
	P=10	**0.286**	1.807	0.000	57.937
	P=15	0.323	1.767	0.000	55.469

The graphs of hub angle and the convergence plot of FMS with difference number of population for each performance criteria is shown in Figure 4-6. The convergence plots show that when more number of fireflies, the objective value is faster to converge to the optimal point. The convergent plots of ITAE, IAE and ISE show when 2 number of fireflies, the objective value not converge to the optimal point. The convergent rate of RMSE with 2 number of fireflies is slower compare with others number of fireflies.

Performance Criteria	Hub Angle of FMS	Convergence Plot
ITAE		
IAE		
ISE		

Performance Criteria	Hub Angle of FMS	Convergence Plot
MSE		
RMSE		

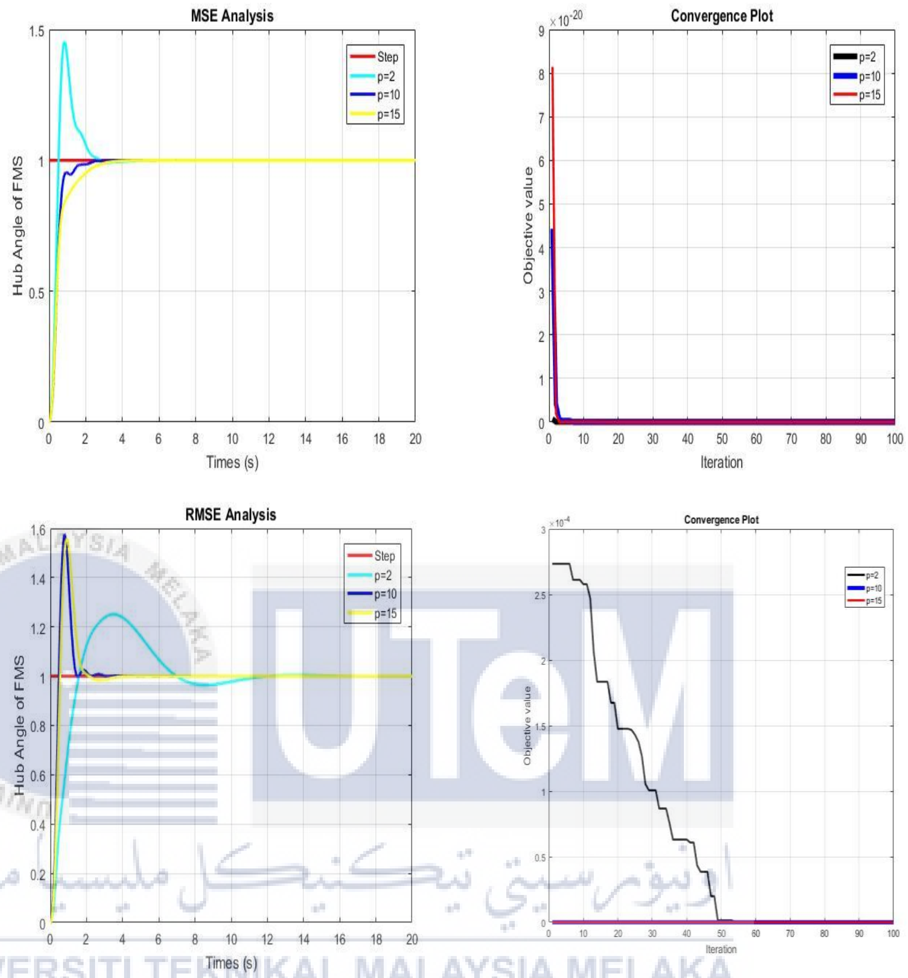
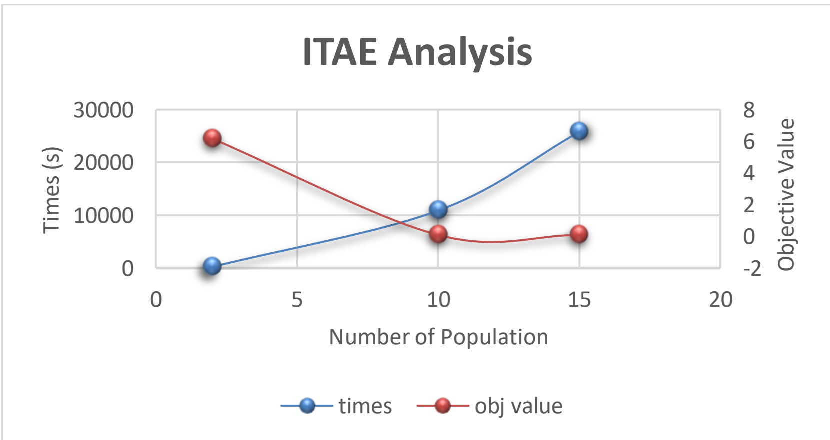


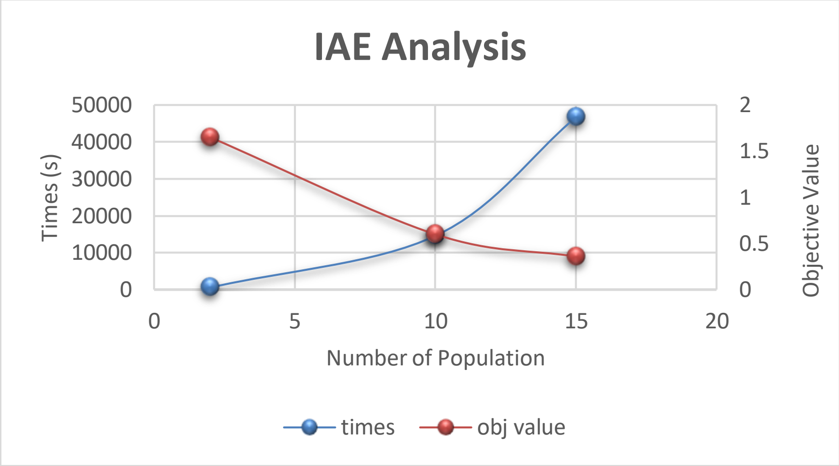
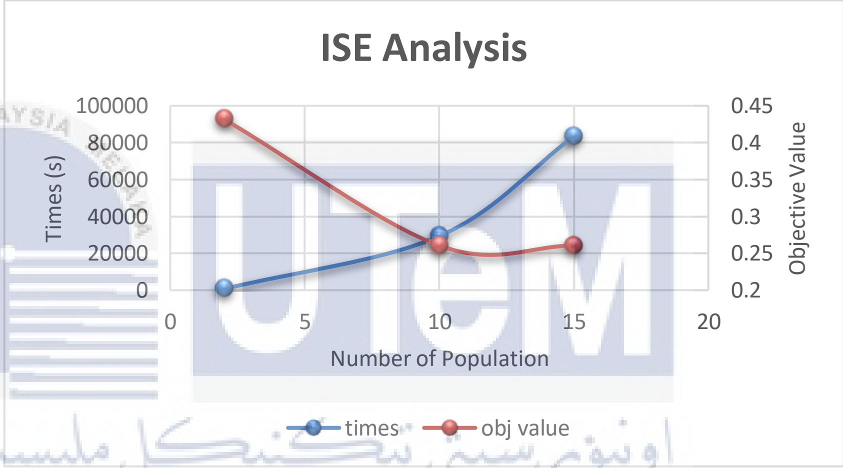
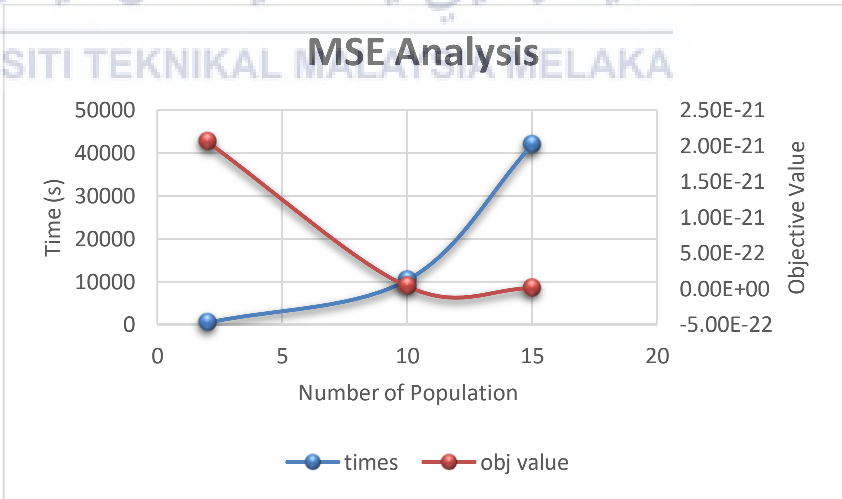
Figure 4-6: The graphs of hub angle of FMS and the convergence plot with difference number of population for each performance criteria.

The processing time and the objective value for each performance criteria is tabulated in Table 4-13. This result show that when number of firefly increase, the processing time increase, and the objective value decrease. This is because the more number of firefly, it required more time to find the optimal value, and it will generate better objective value. The objective value that closer to be zero is much better. Figure 4-7 clearly show that the effect of number of population to the processing time and the objective value.

Table 4-13: The objective value (error) and the processing time for each performance criteria.

Parameter		Processing time, t (s)	Objective value (Error)
ITAE	P=2	215.6726	6.1662
	P=10	10878.0000	0.0900
	P=15	25772.0000	0.0900
IAE	P=2	553.5973	1.6496
	P=10	14552.0000	0.5992
	P=15	46705.0000	0.3580
ISE	P=2	640.6712	0.4318
	P=10	29300.0000	0.2605
	P=15	83460.0000	0.2605
MSE	P=2	367.7314	2.0543E-21
	P=10	10303.0000	3.0990E-23
	P=15	41785.0000	2.3656E-24
RMSE	P=2	480.0306	2.6892E-06
	P=10	23066.0000	9.4301E-14
	P=15	67916.0000	3.8556E-14

Performance Criteria	**Processing time and objective value vs number of population**
ITAE	

Performance Criteria	Processing time and objective value vs number of population
IAE	
ISE	
MSE	

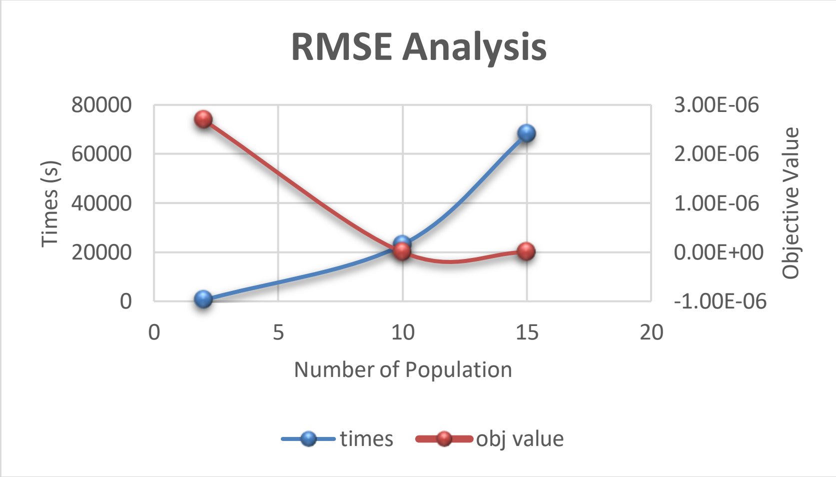
Performance Criteria	Processing time and objective value vs number of population
RMSE	

Figure 4-7: The graph of processing time and objective value vs number of iteration for each performance criteria.

4.4 Analyse the Performance Criteria That Suitable for FMS

The performance criteria is used by the Firefly Algorithm to tune the PID controller to control the performance of the hub angle of FMS. This section is to analysis which performance criteria as objective function that more suitable to tune the PID controller where the FMS have the better performance. In order to analysis the performance, the equation (4-1) is used.

$$\theta_{per} = \beta_1 T_r + \beta_2 T_s + \beta_3 e_{ss} + \beta_4 OS \qquad (4\text{-}1)$$

where θ_{per} is the performance of hub angle, β_1 is the coefficient of rise time, T_r is rise time, $\beta_2$is the coefficient of settling time, T_s is settling time, $\beta_3$is the coefficient of steady state error, e_{ss} is steady state error, β_4 is the coefficient of overshoot, and OS is overshoot.

In order to analysis which performance criteria is more suitable for the FMS, the calculation been carry out. The Condition 1 is to set the coefficient for the transient response and steady state condition is equally important (β_1=β_2=β_3=β_4=0.25). The Condition 2 is set the coefficient with more important of rise time condition which are β_1=0.5, β_2=0.2, β_3=0.1 and β_4=0.2. However, for the Condition 3 is set the coefficient with more important of overshoot condition which are β_1=0.2, β_2=0.2, β_3=0.1 and β_4=0.5. The β_3 is set to be 0.1 because the system have the PID controller, where PID controller will have smaller the steady state error. Table 4-14 clearly show that the equation of each condition.

Table 4-14: The evaluation of each condition.

	More important on	Equation
Condition 1	Equally important	β_1=β_2=β_3=β_4=0.25 θ_{per} =0.25T_r+0.25T_s+0.25e_{ss}+0.25OS
Condition 2	Rise time condition	β_1=0.5, β_2=0.2, β_3=0.1, β_4=0.2 θ_{per} =0.5T_r+0.2T_s+0.1e_{ss}+0.2OS
Condition 3	Overshoot condition	β_1=0.2, β_2=0.2, β_3=0.1, β_4=0.5 θ_{per} =0.2T_r+0.2T_s+0.1e_{ss}+0.5OS

This analysis method is to know the most suitable performance criteria for the Firefly Algorithm to tune the PID controller in order to obtain the best performance for FMS. The performance of FMS is depend on the total based on the calculation of the analysis condition.

By theoretical, the lower the θ_{per}, the better the performance of the hub angle. This is because the rise time and settling time prefer faster (smaller value) to get the better performance. However, for the overshoot and steady state error prefer to be smaller. The result of the Table 4-15 is calculated based on Condition 1,

Table 4-16 is calculated based on the Condition 2, and Table 4-17 is based on Condition 3.

By comparing overall of the result, it shows that for all of the condition, the lowest θ_{per} is MSE with 10 number of fireflies. This means that MSE with 10 number of fireflies is more suitable for the application of FMS. This maybe because the overshoot for MSE with 10 number of population is the lowest compare with others. Throughout the calculation, those 3 conditions show that RMSE with 10 number of population have the worst result. This is because of it have the highest overshoot compare with others.

Figure 4-8 show the combine graph for those 5 performance criteria with 10 number of population. The graph is plot for 10 second to show the shape for each performance criteria. The graph clearly show that the RMSE have the highest overshoot and MSE does not have overshoot. All the performance criteria do not have steady state error, where the graph show that it in the steady state at 10 second.

For the suggestion, most of the system is not suitable to use the equal important of transient response and steady state condition. This is because some of the application that required accuracy which more importance on overshoot condition. For example for the cutting machine need high accuracy to avoid injuries even the system is slow. For this condition, the overshoot condition is more important than the rise time condition. For some of the system that required fast starting time, the rise time condition is more important. The analysis condition can be set the coefficient of transient response and steady state condition depend on the requirement of the system.

Table 4-15: Condition 1: to measure the performance with equal importance of transient response and steady state condition.

Parameter		**0.25 T_r**	**0.25 T_s**	**0.25 e_{ss}**	**0.25 OS**	$\boldsymbol{\theta_{per}}$
ITAE	P=2	1.268	10.824	0.000	212.308	224.400
	P=10	4.180	4.360	0.000	10.308	16.148
	P=15	4.180	4.360	0.000	10.308	16.148
IAE	P=2	5.632	36.052	0.000	73.808	115.492
	P=10	1.468	4.168	0.000	10.308	15.944
	P=15	1.468	4.168	0.000	10.308	15.944
ISE	P=2	1.968	18.556	0.000	85.364	105.888
	P=10	1.352	6.276	0.000	47.192	54.82
	P=15	1.352	6.276	0.000	47.192	54.82
MSE	P=2	1.256	9.008	0.000	185.296	195.56
	P=10	2.160	6.016	0.000	1.192	**9.368**
	P=15	4.600	9.840	0.000	2.000	16.440
RMSE	P=2	4.476	39.116	0.080	103.796	147.468

Parameter		**0.25 T_r**	**0.25 T_s**	**0.25 e_{ss}**	**0.25 OS**	θ_{per}
	P=10	1.144	7.228	0.000	231.748	**240.120**
	P=15	1.292	7.068	0.000	221.876	230.236

Table 4-16: Condition 2: To measure the performance with more importance of transient response condition.

Parameter		**0.5 T_r**	**0.2 T_s**	**0.1 e_{ss}**	**0.2 OS**	θ_{per}
ITAE	P=2	0.634	13.530	0.000	265.385	279.549
	P=10	0.740	5.450	0.000	12.885	19.075
	P=15	0.740	5.450	0.000	12.885	19.075
IAE	P=2	2.816	45.065	0.000	92.260	140.141
	P=10	0.734	5.210	0.000	12.885	18.829
	P=15	0.734	5.210	0.000	12.85	18.289
ISE	P=2	0.984	23.195	0.000	106.705	130.884
	P=10	0.676	7.845	0.000	58.990	67.511
	P=15	0.676	7.845	0.000	58.990	67.511
MSE	P=2	0.628	11.260	0.000	231.620	243.508
	P=10	1.080	7.520	0.000	1.490	**10.090**
	P=15	2.300	12.300	0.000	2.500	17.100
RMSE	P=2	2.238	48.895	0.200	129.745	181.078
	P=10	0.572	9.035	0.000	289.685	**299.292**
	P=15	0.646	8.835	0.000	277.345	286.826

Table 4-17: Condition 3: To measure the performance with more importance of steady state condition.

Parameter		**0.2 T_r**	**0.2 T_s**	**0.1 e_{ss}**	**0.5 OS**	θ_{per}
ITAE	P=2	1.585	13.530	0.000	106.154	121.269
	P=10	1.850	5.450	0.000	5.154	12.454
	P=15	1.850	5.450	0.000	5.154	12.454
IAE	P=2	7.040	45.065	0.000	36.904	89.009
	P=10	1.835	5.210	0.000	5.154	12.199
	P=15	1.835	5.210	0.000	5.154	12.199
ISE	P=2	2.460	23.195	0.000	42.682	68.337

Parameter		0.2 T_r	0.2 T_s	0.1 e_{ss}	0.5 OS	θ_{per}
	P=10	1.690	7.845	0.000	23.596	33.131
	P=15	1.690	7.845	0.000	23.596	33.131
MSE	P=2	1.570	11.260	0.000	92.648	105.478
	P=10	2.700	7.520	0.000	0.596	**10.816**
	P=15	5.750	12.300	0.000	1.000	19.050
RMSE	P=2	5.595	48.895	0.200	51.898	106.588
	P=10	1.430	9.035	0.000	115.874	**126.339**
	P=15	1.615	8.835	0.000	110.938	121.338

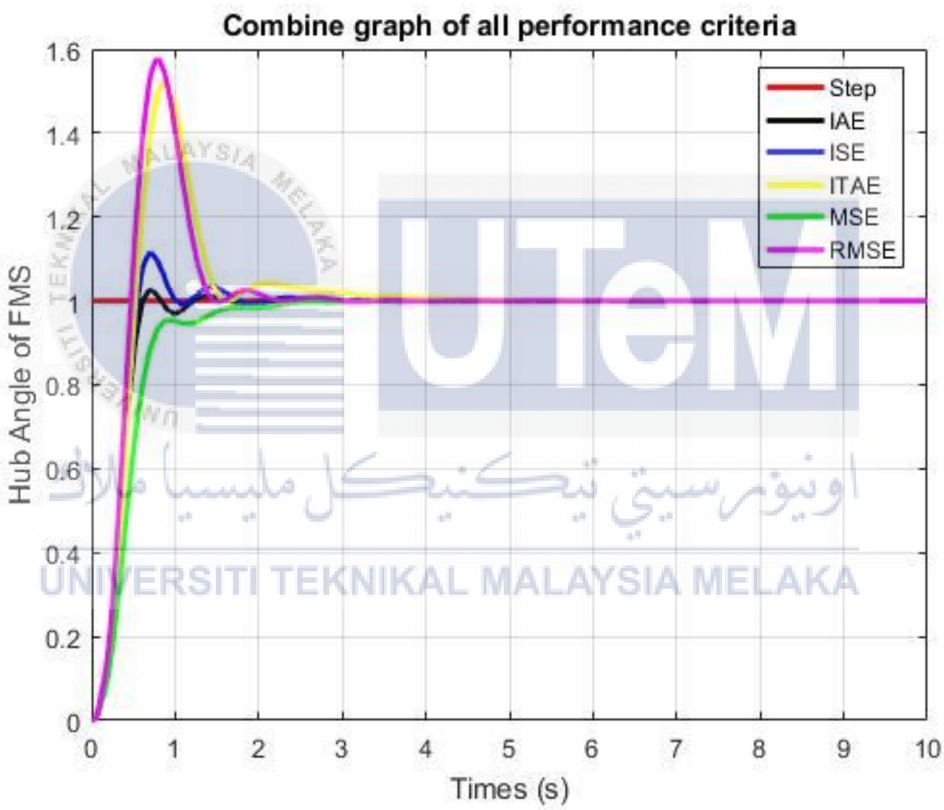


Figure 4-8: The combine graph for all performance criteria with 10 number of population.

4.5 Summary

The result being done for the final year project 2 is shown in the figures and tables. All the result is to measure the performance of the Firefly Algorithm. When the number of population increases, it will move toward the global minimum point. The numbers of iteration increase, the number of evaluations of the solution increase, so the solution will lead at the best possible optimal point. The algorithm achieves better result with lower number of dimension, which the level of difficulty of the problems is reduced. For the FMS, the performance criteria (error) is used by the Firefly Algorithm to tune the PID controller that can control the performance of hub angle. The performance of hub angle is analyse based on the equation (4-1) which is $\theta_{per} = \beta_1 T_r + \beta_2 T_s + \beta_3 e_{ss} + \beta_4 OS$. The analysis is conduct based on 3 conditions. Those 3 conditions get the same result which is MSE with 10 number of population get the best performance of hub angle for FMS.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In the conclusion, the optimization means the process to determine the best possible solution. The Firefly Algorithm is bio-inspired by the flashing pattern of the firefly in the nature which developed by Yang in 2008. Firefly Algorithm is a stochastic and meta-heuristic optimization problem which consists of random parameter.

The Firefly Algorithm is run for the experiment of varying the parameter setting and the flexible manipulator system by using Matlab 2016a. The parameter setting that been varying are number of dimension, number of iteration and number of population. The experiment of varying parameter is using benchmark functions as the objective function. The benchmark function is used to test and verify the algorithm.

The results shown that when number of dimension increase, the complexity of the problem increase, the global optimal point is more difficult to find. When the number of iteration increases, the higher possibility to find the global minimum point, this is because the number of evaluation increases. When the number of population increases, it means more fireflies in the search space, easier to find the optimal point.

The Firefly Algorithm is able to tune the PID controller which to find the most suitable solution for the flexible manipulator system. The PID parameters that tuned by Firefly Algorithm is used to simulated the system. Firefly Algorithm use performance criteria as the objective function to tune the PID controller. There are few performance criteria been used to do a comparison on the performance of FMS. performance criteria been used are Integral Time-weighted Absolute Error (ITAE), Integral Absolute Error (IAE), Integral Square Error (ISE), Mean Square Error (MSE) and Root Mean Square Error (RMSE).

The result of the system is the hub angle of the flexible arm. The method of analysis is based on the transient response and the steady state condition. The equation (4-1) which is $\theta_{per} = \beta_1 T_r + \beta_2 T_s + \beta_3 e_{ss} + \beta_4 OS$ where it is used to analyse the performance of the FMS. The least the θ_{per}, the better the performance. The analysis method is set to be 3 condition which are equal importance of transient response and steady state condition, more importance of rise time condition and more importance of overshoot condition. The results show that the performance criteria of MSE with 10 number of population have the best result for those 3 conditions. This means MSE with 10 number of population is most suitable as the objective function for Firefly Algorithm to tune the PID controller. It shows that it have the best performance on the hub angle of FMS.

5.2 Future Works

For the future word and recommendation, more application can be carry out using the Firefly algorithm and the performance criteria. This can be know the Firefly algorithm is more suitable for which type of application. This also can verify the accuracy and effectiveness of the Firefly Algorithm. Firefly Algorithm can tune the controller by using the performance criteria as the objective function for others application. This FMS also can be modify to use others types of controller to know the effect on the performance of FMS.

REFERENCES

[1] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, “Firefly Algorithm for Optimization Problem,” *Appl. Mech. Mater.*, vol. 421, no. April 2014, pp. 512–517, 2013.

[2] S. K. Sarangi, R. Panda, S. Priyadarshini, and A. Sarangi, “A new modified firefly algorithm for function optimization,” *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 2944–2949, 2016.

[3] M. Jamil, “A literature survey of benchmark functions for global optimisation problems Xin-She Yang,” vol. 4, no. 2, pp. 150–194, 2013.

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

[5] “OPTIMIZATION OF BENCHMARK FUNCTIONS USING CHEMICAL REACTION North Dakota State University,” no. April, 2013.

[6] R. B. Francisco, M. F. P. Costa, and A. M. A. C. Rocha, “Experiments with Firefly Algorithm.”

[7] L. Zhang, L. Liu, X. Yang, and Y. Dai, “A Novel Hybrid Firefly Algorithm for Global Optimization,” pp. 1–17, 2016.

[8] 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 functions,” pp. 1–8, 2013.

[9] C. Liu, F. Gao, and N. Jin, “Design and simulation of a modified firefly algorithm,” *Proc. - 2014 7th Int. Jt. Conf. Comput. Sci. Optim. CSO 2014*, pp. 21–25, 2014.

[10] I. Fister, X. S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm Evol. Comput.*, vol. 13, pp. 34–46, 2013.

[11] H. A. Kasdirin, “Adaptive bio-inspired firefly and invasive weed algorithms for global optimisation with application to engineering problems,” *Thesis*, no.

August, 2016.

[12] C. Napoli, G. Pappalardo, E. Tramontana, Z. Marszalek, D. Polap, and M. Wozniak, "Simplified firefly algorithm for 2D image key-points search," *IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - CIHLI 2014 2014 IEEE Symp. Comput. Intell. Human-Like Intell. Proc.*, 2015.

[13] X. S. Yang and X. He, "Firefly algorithm: recent advances and applications," *Int. J. Swarm Intell.*, vol. 1, no. 1, p. 36, 2013.

[14] E. Alba, "Improving flexibility and efficiency by adding parallelism to genetic algorithms ´," pp. 91–114, 2002.

[15] P. Zhang and Y. C. Li, "Simulations and trajectory tracking of two manipulators manipulating a flexible payload," *2008 IEEE Int. Conf. Robot. Autom. Mechatronics, RAM 2008*, vol. 00, no. 1, pp. 72–77, 2008.

[16] H. Li, Z. Yang, and T. Huang, "Dynamics and elasto-dynamics optimization of a 2-DOF planar parallel pick-and-place robot with flexible links," *Struct. Multidiscip. Optim.*, vol. 38, no. 2, pp. 195–204, 2009.

[17] Z. Mohamed, "Dynamic Modelling and Control of a Flexible Manipulator," no. April, 2003.

[18] D. Corrigan, "Characterising the Response of a Closed Loop System," *Handout, Control Syst. Eng. Electron. Electr.*, pp. 1–25, 2012.

[19] G. Lin and G. Liu, "Tuning PID controller using adaptive genetic algorithms," *ICCSE 2010 - 5th Int. Conf. Comput. Sci. Educ. Final Progr. B. Abstr.*, pp. 519–523, 2010.

[20] M. Shahrokhi and A. Zomorrodi, "Comparison of Tuning Methods of Pid Controller," *Dep. Chem. Pet. Eng. Sharif Univ. Technol.*, pp. 1–2, 2013.

[21] A. Eldin, A. Awouda, and R. Bin Mamat, "New PID Tuning Rule Using ITAE Criteria," *Int. J. Eng.*, no. March, pp. 597–608, 2016.

[22] D. Maiti, A. Acharya, and A. Konar, "Tuning PID and PIλDδ Controllers using the Integral.pdf," pp. 457–462, 2008.

[23] B. Doicin, M. Popescu, and C. Patrascioiu, “PID controller optimal tuning,” *Proc. 8th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2016*, pp. 16–19, 2017.

APPENDICES

APPENDIX A GANTT CHART

Month	Sep	Oct	Nov	Dec	Jan	Feb	Mar	April	May	June
Understand the basic concept and find coding.										
Run with benchmark functions										
Run with application										
Run with more error analysis										
Report writing										

APPENDIX B MATLAB CODE FOR FIREFLY ALGORITHM, FOR OBJECTIVE FUNTION AND MATLAB SIMULATION DIAGRAM OF FMS

Code for Firefly Algorithm

```
%% Reset and Clear
clc;
clear;
close all;

tic;
run=1
run1=30

%% Set how many sets you want to run
for set_num = 1 : 1 : run


%% Initialize CSV file
% fileName = sprintf('set_%d.csv',set_num);
% csvwrite(fileName,set_num);
% column1_title='Iteration';
% column2_title='Best Cost';
% fid = fopen(fileName, 'w');  %'w' sttands for write, fid means file ID
% fprintf(fid, '%s,%s\n', column1_title, column2_title);
% fclose(fid);


```

```
%% Problem Definition

 CostFunction=@(x) tracklsq(x);         % Cost Function

nVar=3;                     % Number of Decision Variables

VarSize=[1 nVar];           % Decision Variables Matrix Size

VarMin= 0;                 % Decision Variables Lower Bound
VarMax= 2;                 % Decision Variables Upper Bound

%% Firefly Algorithm Parameters

MaxIt=100;          % Maximum Number of Iterations

nPop=5;             % Number of Fireflies (Swarm Size)

gamma=1;             % Light Absorption Coefficient

beta0=2;             % Attraction Coefficient Base Value

alpha=0.2;           % Mutation Coefficient

alpha_damp=0.98;     % Mutation Coefficient Damping Ratio

delta=0.05*(VarMax-VarMin);      % Uniform Mutation Range

m=2;


if isscalar(VarMin) && isscalar(VarMax)
    dmax = (VarMax-VarMin)*sqrt(nVar);
else
    dmax = norm(VarMax-VarMin);
end

%% Initialization

% Empty Firefly Structure
firefly.Position=[];
firefly.Cost=[];

% Initialize Population Array
pop=repmat(firefly,nPop,1);

% Initialize Best Solution Ever Found
BestSol.Cost=inf;

% Create Initial Fireflies
for i=1:nPop
   pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
   pop(i).Cost=CostFunction(pop(i).Position);

   if pop(i).Cost<=BestSol.Cost
       BestSol=pop(i);
   end
end

% Array to Hold Best Cost Values
BestCost=zeros(MaxIt,1);
```

```
%% Firefly Algorithm Main Loop

for it=1:MaxIt

    newpop=repmat(firefly,nPop,1);
    for i=1:nPop
        newpop(i).Cost = inf;
        for j=1:nPop
            if pop(j).Cost < pop(i).Cost
                rij=norm(pop(i).Position-pop(j).Position)/dmax;
                beta=beta0*exp(-gamma*rij^m);
                e=delta*unifrnd(-1,+1,VarSize)
                %e=delta*randn(VarSize);

                newsol.Position = pop(i).Position ...
                                + beta*rand(VarSize).*(pop(j).Position-pop(i).Position) ...
                                + alpha*e;

                newsol.Position=max(newsol.Position,VarMin);
                newsol.Position=min(newsol.Position,VarMax);

                newsol.Cost=CostFunction(newsol.Position);

                if newsol.Cost <= newpop(i).Cost
                    newpop(i) = newsol;
                    if newpop(i).Cost<=BestSol.Cost
                        BestSol=newpop(i);
                    end
                end

            end
        end
    end

    % Merge
    pop=[pop
         newpop];   %#ok

    % Sort
    [~, SortOrder]=sort([pop.Cost]);
    pop=pop(SortOrder);

    % Truncate
    pop=pop(1:nPop);

    % Store Best Cost Ever Found
    BestCost(it)=BestSol.Cost;

    % Show Iteration Information
    str_it = num2str(it);
    str_BestCost = num2str(BestCost(it));
    disp(['Iteration ' str_it ': Best Cost = ' str_BestCost]);

    % Damp Mutation Coefficient
    alpha = alpha*alpha_damp;

    % Save data at the end of each iteration
% %     fid = fopen(fileName, 'w');
%     fprintf(fid, '%s,%s\n', str_it, BestCost);
%     fclose(fid);

end
```

```
%% Results
excelno=1

BestCost1(:,run1)=BestCost;

% BestCost1=BestCost
% xlswrite('FA.xlsx',BestCost1,excelno,['C' num2str(run1)]);

xlswrite('rmse_lim0_2_n5_test1.xlsx',BestCost1)
run1=run1+1;
% figure;
% %plot(BestCost,'LineWidth',2);
% semilogy(BestCost,'LineWidth',2);
% xlabel('Iteration');
% ylabel('Best Cost');
% grid on;

t=toc
end
```

Code for Objective Function of FMS

```
function F = tracklsq(pid)

% set_param('FMS2','MaxConsecutiveZCsMsg','none');
        % Track the output of optsim to a signal of 1
       load FMS_parameter
        % Variables a1 and a2 are shared with RUNTRACKLSQ
        Kp = pid(1); %
        Ki = pid(2);%
        Kd = pid(3);%
           pid
        %sprintf('The value of interation Kp= %3.0f, Ki= %3.0f, Kd= %3.0f', pid(1),pid(2),pid(3))
        % Compute function value
        simopt = simset('solver','ode45','SrcWorkspace','Current','DstWorkspace','Current');  % Initialize sim options
        [tout,xout,yout] = sim('FMS11',[0 12],simopt); % Tukar nama file simulink cth FMS1



       F=sum((e)) % you can used ITAE evaluations. % e0 is from simulink FMS11
       F=sum((e1)) % you can used IAE evaluations. % e0 is from simulink FMS11
       F=sum((e2)) % you can used IsE evaluations. % e0 is from simulink FMS11
       F=0.2*(sum((e0))) %you can used MSE evaluations. % e0 is from simulink FMS11
       F=sqrt(0.2*(sum((e0)))) %you can used RMSE evaluations. % e0 is from simulink FMS11

       yout;

   end
```

Simulation Diagram of FMS