SPIRAL DYNAMIC OPTIMIZATION ALGORITHM FOR ENGINEERING APPLICATION

NUR AIFAA BINTI SHAMSHOL ALI

BACHELOR OF ELECTRICAL ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

SPIRAL DYNAMIC OPTIMIZATION ALGORITHM FOR ENGINEERING APPLICATION

NUR AIFAA BINTI SHAMSHOL ALI

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 “SPIRAL DYNAMIC OPTIMIZATION ALGORITHM FOR ENGINEERING APPLICATION is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

APPROVAL

I hereby declare that I have checked this report entitled “SPIRAL DYNAMIC OPTIMIZATION ALGORITHM FOR ENGINEERING APPLICATION” 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

I am grateful to Allah SWT, my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. Never forget, peace and prayers to the Prophet Muhammad s.a.w. I also dedicate this work to my lovely father, Shamshol Ali Bin Abdul Rahim and appreciated mother, Robaee Binti Md Salleh who has encouraged me all the way and whose encouragement has made sure that I give it all it takes to finish that which I have started. To all my sister; Nur Ayunni, Nur Aimii and Nur Aqilah who have been affected in every way possible by this quest. Thank you. My love for you all can never be quantified. God bless you.

ACKNOWLEDGEMENTS

My deepest gratitude goes to God who has provided all that was needed to complete this final year project for which it was undertaken for. There was never lack or want. Throughout this entire study, He took care of everything that would have stopped me in my tracks and strengthened me even through my most difficult times.

I owe a deep debt of gratitude to our university for giving us an opportunity to complete this final year project.

I am grateful to some people, who worked hard with me from the beginning till the completion of the final year project particularly my supervisor Dr Ruzaini Bin Hashim , who has been always generous during all phases of the research, and I highly appreciate the efforts expended by all the person that involve to accomplish my project.

I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing my final year project. They had given their critical suggestion and comments throughout my research Thanks for the great friendship.

I also would like to express my wholehearted thanks to my family for their generous support they provided me throughout my entire life and particularly through the process of pursuing the degree. Because of their unconditional love and prayers, I have the chance to complete this final year project.

Finally, I would like to thank everybody who was important to this FYP report, as well as expressing my apology that I could not mention personally each one of you.

ABSTRACT

This project presents the optimization technique. Optimization of a design could be simply to minimize the cost of production or to maximize the efficiency of production. While in optimization algorithm is a strategy to find the best or an optimal solution for any real world problem such as engineering problem. This project is focused on Spiral Dynamic Optimization Algorithm (SDA) which is a natured-inspired metaheuristic concept. The algorithm was inspired by spiral phenomena in nature that commonly found in nautilis shells, whirling currents and spiral galaxy. In SDA has two specific setting parameter which are the convergence rate and the rotation rate whose values shows its trajectory. The common centre of trajectory is the best point in all search points. The search points moving toward the common centre with logarithmic spiral trajectories [1]. The algorithm is tested using several benchmark functions and is used to optimize the PI, PD and PID controller of a flexible manipulator system. The results show that the algorithm surpass the SDA and is able to tune the controller parameter to their optimum value.

ABSTRAK

Projek ini membentangkan teknik pengoptimuman. Pengoptimuman reka bentuk boleh jadi semata-mata untuk meminimumkan kos pengeluaran atau untuk memaksimumkan kecekapan pengeluaran. Algoritma pengoptimuman adalah strategi untuk mencari penyelesaian terbaik atau penyelesaian yang optimum untuk masalah dunia sebenar seperti masalah kejuruteraan. Projek ini difokuskan pada Algoritma Pengoptimuman Dynamik Spiral (SDA) yang merupakan konsep metaheuristik. Algoritma ini diilhamkan oleh fenomena lingkaran dalam alam yang biasa dijumpai dalam kerang nautilus, arus berputar dan galaksi spiral. Dalam SDA mempunyai dua parameter penetapan khusus yang merupakan kadar penumpuan dan kadar putaran yang menunjukkan trajektorinya. Pusat trajektori umum adalah titik terbaik di semua titik carian. Titik carian yang bergerak ke arah pusat bersama dengan trajektori logaritmik. Algoritma diuji menggunakan beberapa fungsi penanda aras dan digunakan untuk mengoptimumkan pengawal PI, PD dan PID sistem manipulator yang fleksibel. Keputusan menunjukkan bahawa algoritma SDA dapat diselarikan dengan parameter pengawal ke nilai optimum.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SDA - Spiral Dynamic Algorithm
D - Dimension
Np - Number Of Points
I - Iteration
Kp - Propotional gain
Ki - Integral gain
Kd - Derivative gain
PI - Propotional-integral
PD - Propotional-derivative
PID - Propotional-derivative-integral

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Optimization is a wide, fast and interdisciplinary research area, whose knowledge topics such as mathematics, computer science and art is required. In optimization issues, it is a method which is executed iteratively by comparing various solutions until an optimum or a good solution is found. For example, in the mathematics , it is a way to find a minimum and maximum solution of a certain mathematical function with or without the presence of a constrain such as boundary or the solution. While in computing, it is the process of modifying a system to make some features of it work more efficiently or use a few resources.The optimization can have sense at different levels, from the lowest (development of circuits, writing of machine code designed especially for the architecture) up to the highest levels of making of implementation, use or design of algorithms.

In the context of problem-solving application, optimization is seen as a method to find the best or an optimum solution for a given world problem or whatever, which is naturally more complex and challenging [2]. As always, the solutions found by the optimization algorithm is decisive solution and practically can be used to achieve good result. Furthermore, the number of dimensions is high in problems and incorporation of objective functions also increase the difficulty of finding an optimum solution for the problem. Therefore, there is a need for further research to find a better optimization strategy in order to achieve a good result. In real world optimization, there could be more than one objective that the designer may want to optimize simultaneously. The multiple objective optimization algorithms are complex and computationally expensive. Therefore the most important objective is chosen as the objective function and the other objectives are included as constraints by restricting their values within a certain range.

Metaheuristic optimization algorithms have gained a lot of interest between world researchers. These algorithms are inspired from biological phenomena or natural phenomena. Algorithms that based on natural are such as firefly optimization algorithm , galaxy-based search algorithm and spiral dynamics inspired optimization (SDA)[3] . The SDA is a metaheuristic algorithm inspired from spiral phenomena in nature such as tornado, nautilus shell, low pressure fronts, spiral of waves and galaxy. Spiral dynamic algorithm has a simple structure compared to other algorithms, hence the total computation time to complete the entire search process is relatively short and easy to program[2]. Spiral dynamic algorithm applies diversification in early stage of the search where the aim is to find better solution in a large area during exploration. Spiral dynamic algorithm will perform the search for better possible solution around good solution found during exploration phase. This intensive search toward best possible solution is called intensification. The main component of spiral dynamic algorithm is a spiral model, which can determine the shape, and characteristic of a spiral. It has two specific setting parameters : the convergence rate and the rotation rate whose values characterize its trajectory. The common centre is defined as the best point in all search points. The search points moving toward the common centre with logarithmic spiral trajectories can find better solutions and update the common centre [4].

1.2 Project Motivation

Nowdays, there are many method to solve any problem one of them by using optimization method such as optimization spiral dynamic optimization algorithm. The main aim of the research is to investigate the concept of Spiral Dynamic Algorithm on how it find the best optimal value and to solve engineering problems.The flexible manipulator system is selected as the target application and platform to test the performance of the algorithm. The algorithm is tested using several benchmark functions and is used to optimize the PI, PD and PID controller of a FMS.

1.3 Problem Statement

There a lot of method to be used in solving a problems such as " try and error" method but it is not guarantee to solve it. There are some example that have faced on daily life such as in business optimization need to measure the efficiency, productivity and performance of a business while in mathematical optimization need to find a minimum or maximum solution of a certain mathematical function. So by using SDA it can help to solve that problem. The problem that related to this research which is to find the optimum value that can be practically use to achieve a good outcome.

1.4 Objectives

The objectives of this project are:

1. To investigate the performance of the SDA with various parameter setting by using 10 numerical benchmark functions.
2. To investigate the most suitable controller for flexible manipulator based on error criteria.
3. To investigate the performance of SDA with various parameter setting in tuning propotional-integral-derivative (PID) based on error criteria for application flexible manipulator system

1.5 Scope Of Research

This project is focused on the investigating the performance of the SDA with parameter setting by using 10 numerical benchmark function. Performances of the algorithm that will statistically analyze in terms of convergence speed that will graphically presented. Then, spiral dynamic optimization algorithms will be discussed with applications to various domains. An important aspect of the optimization algorithm is its parameter set, which needs careful analysis for deployment in various domains such as its effectiveness as an optimization technique and applications to engineering problems. Flexible manipulator system is selected as the target application and platform to test the performance of the algorithm. It has been chosen to investigate the most suitable controller with different error criteria. Three common controller been used in this research are PI, PD and PID controller that will be tuned with different error criteria. The performance of SDA with parameter setting in tuning propotional-integral-derivative (PID) with error criteria for application flexible manipulator system also in scope of research. The performance of SDA are graphically presented.

1.6 Report Outline

A brief description of this report is described in this section. Generally, this report contains five chapters in total and all these chapters will deliver the overall information about this report, consequently.

The first chapter of this report will contain the introduction of this project. The overall idea of the project is briefly explained in this chapter.

This second chapter in this report will deliver the information on literature review of this project. This previous work related to the project will be analyzed in detail as guideline to improve the current project so that it will be much better.

The third chapter in this report will explain about the methodology that is being implemented to execute this project. All the formulas and theory used will be explained in this chapter.

The fourth chapter in this report will show the early results of the progress from the methodology used for this project. The data obtained from the results will be analyzed further to verify either the desired outcome of this project is achieved or not and the results gain will be used for the next phase of the project.

Lastly, the fifth chapter in this report will summarize the overall conclusion obtained from this project. The further work will be planned for the next step of this project and all the references source will be cited in this chapter

CHAPTER 2

LITERATURE REVIEW

2.1 Theory of Meta-Heuristic Algorithm

In recent years, meta-heuristic algorithms have attracted more attention in recognition their versatility and concepts since real-world systems seeking optimization have tended to get greater and more complicated. Most of meta-heuristics are constructed on the analogy of physical phenomena or natural phenomena. Meta-heuristic is a heuristic approach strategy for continuous or discontinuous optimization problems[3]. Two main categories of meta-heuristic algorithm are bio-inspired and nature-inspired algorithms. A bio-inspired algorithm is an optimization algorithm where formulation and philosophy inspired by behavior of living organisms while a nature-inspired algorithm is constructed from natural phenomena other than living organism. Some of bio-inspired algorithm are genetic algorithm that imitate the process of natural and genetic evolution, bacterial foraging algorithm inspired by the natural way of Escherichia coli bacteria searching for food throughout their life cycle, ant colony algorithm adopted from the behavior of ants seeking for food sources through the most effective path, particle swarm optimization based on social behavior of bird flocking and bee colony algorithm that is inspired based on the foraging and swarming behavior of honey bee. Examples of nature-inspired algorithms are chemical reaction optimization, simulated annealing and spiral dynamic algorithm that inspired from spiral phenomena in nature.

2.1.1 Theory of SDA

K. Tamura and K. Yasuda recently introduced a new metaheuristics method (in 2010) for continuous optimization problems based on analogy of spiral phenomena in nature which is called Spiral Dyamic Algorithm (SDA). Spiral phenomena commonly found on earth and the universe such as nautilus shells, whirling currents (hurricanes and tornados) , shape of DNA molecule and spiral galaxy[2].

a)Nautilus Shell b)Whirling Currents

c) Spiral Galaxy d) Romanesco Broccoli

Figure 2.1 : Spiral Shapes In Nature [6]

Spiral dynamic algorithm is relatively simple, easy to program, has relatively low computation time for the whole search operation and it has few parameters on the initialisation thus making it practical and easy to use for real world applications. This algorithm is a multipoint search for continous optimization problems (no objective function gradient).

The spiral dynamic algorithm model is composed of plural logarithmic spiral models and their common centre. In this algorithm as search points follow logarithmic spiral trajectories toward the common center defined as the current best point,better solution can be found and the common centre can be updated. Two important features of spiral dynamic algorithm are the diversification and intensification that occur at the early and final phases of the search operation respectively[5].

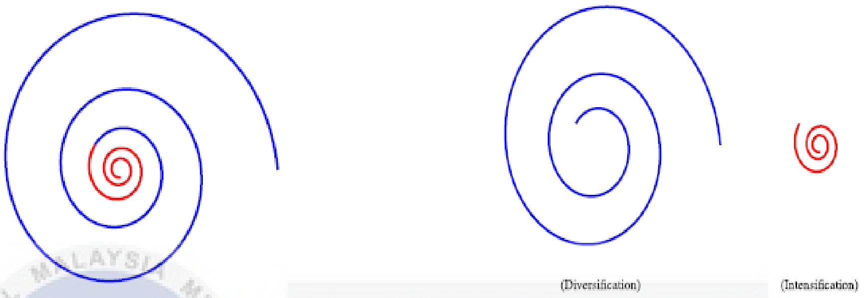


Figure 2.2 : Interpretation Of Logarithmic Spiral [5]

Diversification is the representation of the exploration strategy in which the search point moves from each other to cover the entire search area globally and evaluate the possibility of optimal point locations. Intensification is the strategy searching a better solution by searching around a good solution intensively under a practical concept that better solutions exist around good solutions[7].

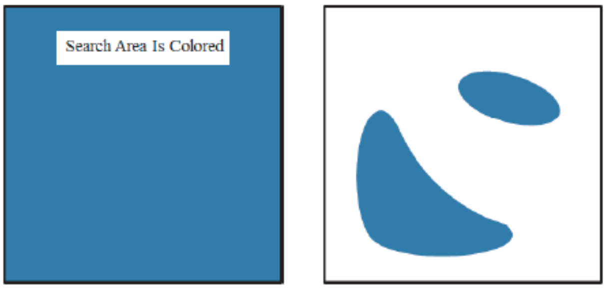
Diversification: Searching wide region,
Intensification: Searching limited region.

In general, strategies based on diversification and intensification is carried out as follows:

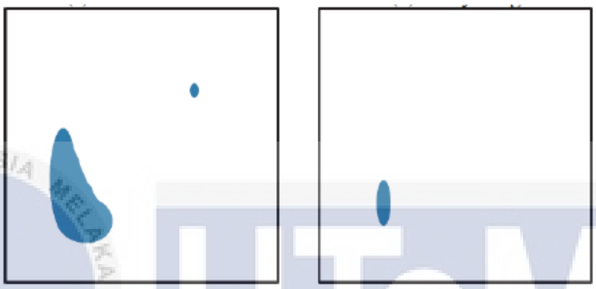
1) In the beginning, search point moves with the purpose of roughly grasping the tendency of distribution of good solutions by searching wider region in the solution space (Figure 2.3 (a), (b)).

2) In the process of searching, to grasp more concrete tendencies, the search region is narrowed down the region where better region in which better solutions may exist.

Then in the final stage, move search point with the purpose of searching intensively the sufficient of regions to find a better solutions (Figure 2.3 (c), (d)).



a)Start of search b)Early Stage



c)Middle Stage d)Final Stage

Figure 2.3: Search Method [7]

A balance of the diversification and the intensification in the search process is important in searching for better solutions in limited time,which can result to a more accurate solution.

In spiral dynamic algorithm, the two features mentioned above are generated from a important spiral model and its contains two sets of parameter mainly on the speed of the convergence and accuracy of the final solution[2]. The spiral model of spiral dynamic algorithm for *n*-dimension is defined as :

$$x(k+1) = Sn(r,\emptyset)x(k) - (Sn(r,\emptyset) - In)x* \qquad 2.1$$

where x^* is a centre point of spiral, *In* is identity matrix, *x* is a coordinate location of a point, k is iteration number and *Sn* $(r,\emptyset) = rR^n\big(\emptyset_{1,2,}\emptyset_{1.3\ldots}\,\emptyset_{n,n-1}\big)$. r is a spiral radius, $\emptyset$ is a rotational angle and $R^{(n)}(\emptyset_{1,2}\,\emptyset_{1,3}\ldots\emptyset_{n,n-1})$is a composition of rotation $n\times n$ matrix. The $R^n\big(\emptyset_{1,2,}\emptyset_{1.3\ldots}\,\emptyset_{n,n-1}\big)$ can be represented as:

$$R^n\big(\emptyset_{1,2,}\emptyset_{1.3\ldots}\,\emptyset_{n,n-1}\big) = \prod_{i=1}^{n-1} 1 \left[\prod_{j=1}^{i} R^{(n)}_{n-i,n+1-j}\,(\emptyset_{n-i,n+1-j}) \right] \qquad 2.2$$

Graphical representations of the spiral model with different values of radius, r and angle, θ are shown in Figure 2.4 where represent spiral forms with *r* = 0.95 , $\emptyset = pi/4$

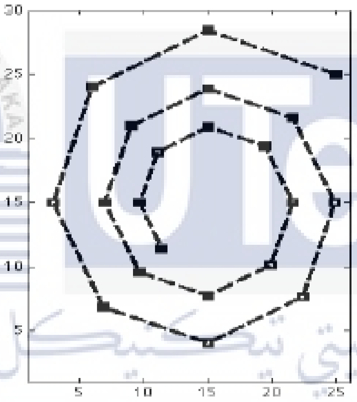


Figure 2.4 : Spiral Forms [5]

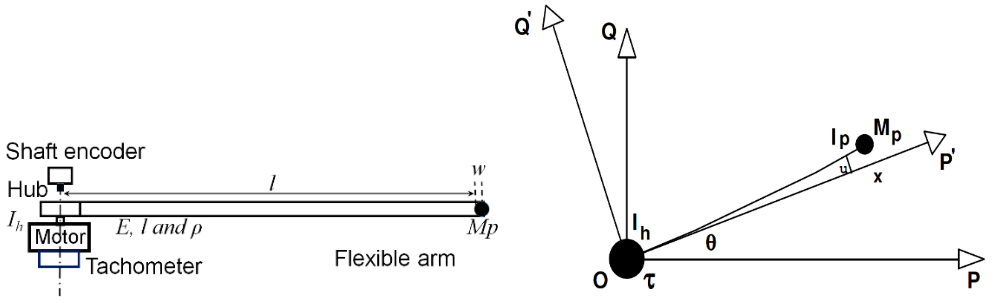
2.2 Theory of Flexible Manipulator System

A schematic diagram of the FMS rig is shown in Figure 2.6 (a) (Azad 1994; Tokhi and Azad, 1997). The system consists of a flexible link made of aluminum beam and attached to an electromechanical motor. The flexible manipulator arm considered in this work is a single-input multi-output system, which is commonly found in the industry [9]. As there are three outputs of interest to be gauged from the system, three different sensor units are incorporated into the system.

An integrated circuit piezoelectric accelerometer is placed at the tip of the beam and used to measure end-point acceleration. The advantages of the sensor are that it is small in size, light in weight, has high voltage sensitivity and low impedance output, which prevent significant amount of signal losses and distortion problems. An encoder with a resolution of 2,048 pulses/revolution and a tachometer are attached to the motor shaft and used to measure hub-angle and hub-velocity respectively. Moreover, a personal computer (PC) embedded with Pentium Celeron 500 MHz processor is connected with PCL818G interfacing unit to the FMS[12].

On the other hand, the schematic representation for the mechanical structure of the FMS is shown in Figure 2.5 (b). The Mp is the payload attached at the tip of the flexible beam, θ is the hub angle or angular displacement , τ is the input torque and u is the elastic deflection of an arbitrary point along the flexible beam. The *POQ* represents the original frame location while the POQ’ represents the new frame location when an input torque, τ is applied to the system to rotate with an angle of θ in the X-Y plane.

The input to the system is an analog voltage (motor torque) while the outputs are hub angle, hub velocity and end-point acceleration [12]. Matlab / Simulink software installed in the PC is used as a tool for controlling and manipulation of the system. The physical parameters and specifications of the flexible manipulator system are shown in Table 2.1.



a) Schematic representation of FMS b) Schematic mechanical model

Figure 2.5 : Flexible Manipulator System [8]

Table 2.1 : Physical Parameters Of Flexible Manipulator System.

Physical parameters	Value
Width (w)	19.008 mm
Length (l)	960 mm
Thickness (h)	3.2004 mm
Hub inertia (Ih)	5.86 x $10^{-4}Kgm^{2}$
Moment of inertia (Ib)	$0.04862Kgm^{2}$
Second moment of inertia (I)	5.1924 x $10^{-11}m^{4}$
Young modulus (E)	71 x $10^{9}Nm^{-2}$
Mass density per unit volume (□)	2.710 Kgm^{-3}

2.3 Types of Controller

2.3.1 PID Controller

PID Controller is having a simple structure, stable and sufficient ability in solving a lot of control problem in the process industry [11].It will give the huge improvements in tuning of PID controllers will have a significant practical impact on its performance. PID controller has three principle of control effects. The proportional process gives a change in the controller output propotional to the control error. While for the integral process gives a change in the controller output propotional to the integral error and its main goal is to eliminate the steady state error.Derivation process is used in some cases to speed up the response or to make sure the stability of the system increase. So that the transient response also improving.Thus, the PID controller will be the main controller being considered in this research.

2.3.2 PI Controller

Directly, PI controller is the most widely used in industrial application due to easy to design, simple structure and low for costing [11]. PI controller will eliminate forced oscillations and steady state error. However, by using integral mode it will give negative effect to speed of the response and overall stability of the system. This controller will not increase the speed of response.

PI controller are very often used in industry and it has been used during:

1. Fast response of the system is not needed.
2. Large disturbance and noise are present during the operation of the process
3. Only one energy storage in the process (capacitive and inductive)
4. There are large transport delay in the system.

2.3.3 PD Controller

Derivative control the controller output is directly propotional to the rate of change of the error. The aim of using PD controller is to increase stability of the system by improving the control. This controller is capable to handle processes with time lag and reduce settling time by improving damping and reducing overshoot [11]. These are disadvantages of PD controller are :

1. Not suited for fast responding systems which are usually lightly damped or initially unstable.
2. Amplifies noise at higher frequencies which result in improper handling of actuators.
3. Does not eliminate steady state error

2.4 Related Previous Work

2.4.1 A Novel Adaptive Spiral Dynamic Algorithm for Global Optimization

Authors in paper [9] is presenting a adaptive spiral dynamic algorithm for global optimization. Spiral dynamic algorithm has a balanced exploration and exploitation through a spiral model. By defining suitable value for the radius and diplacement in the spiral model it may lead the algorithm to converge with high speed. The model allows the algorithm to avoid oscillation around the optimum point and producing the dynamic step size. However for the high dimension the algorithm may easily get trapped into local optima. This is due to the incorporation of a constant radius and displacement in the model. By varying the radius and diplacement of the spiral modelis proposed in this paper. The algorithm is validated with various dimensions of unimodel and multimodal benchmark functions. It also applied to parameter optimization of an autoegressive with exogenous terms dynamic model of a flexible manipulator system. In this paper they also compare with the original algorithm to see the accuracy of the algorithm and it shows the original Spiral Dynamic Algorithm has demonstrated better convergence speed at the initial phase but has settled at local optimum solution. Moreover, the results of time-domain and frequency-domain show that both algorithms have successfully acquired dynamic model for end-point acceleration of the manipulator.The adaptation equation has been formulated as a function of individual search agent fitness costs well as the best global fitness cost at the current iteration.The proposed algorithm shows has successfully avoided local optima. This method has enabled the search points to explore throughly within a search space and exploit the search effectively in a remote location.

2.4.2 Spiral Dynamics Algorithm

For the paper [6] stated that spiral phenomenan can be described mathematically using parametric curves. The logarithmic spiral can realize an effective metaheuristic algorithm. Spiral dynamic-based optimization algorithms have been proposed based on the logarithmic spirals. The algorithm has been applied to well-known benchmarks problems and a number of engineering problems successfully. In the recent years a number of variants of the spiral dynamic algorithm have been reported such as adaptive spiral dynamic algorithm and hybridization with bacterial foraging algorithm, which show improved accuracy, faster convergence and more efficient computations.

2.4.3 A Novel Adaptive Spiral Dynamics Algorithms for Global Optimization

The paper [10] the authors have proposed about four novel adaptive spiral dynamics optimization algorithm. From the proposed novel adaptation strategies based on mathematical and non-mathematical fuzzy logic intelligent methods have been presented without adding extra complexity to the original algorithm structure. The simulation result shows the proposed adaptive algorithm outperforms the original algorithm in terms of speed of convergence based on CPU computation time, fuzzy adaptive approach needed longer time to execute the algorithm compared to other adaptive approaches and spiral dynamic algorithm. Futher simplification of fuzy logic approach is required and computation time in seconds need to be taken into account before fuzzy logic approach can be applied to real world problems. From the paper the results shows that all the proposed adaptive approaches have high potential for real world applications.

2.4.4 Greedy Spiral Dynamic Algorithm with Application to Controller Design

The paper [8] write by M. R. Hashim and M. O. Tokhi focused on greedy spiral dynamic algorithm with application to controller design. Validations using several benchmark functions have been carried out in comparison with original spiral dynamic algorithm. From the paper the results have clearly shown the superior performance of greedy spiral dynamic algorithm in comparison to original algorithm in benchmark function tests. The proposed algorithms has further been validated in controller design of a flexible manipulator system, and it shows that the proposed algorithm has outperformed the spiral dynamic algorithm by resulting in better system performance with response overshoot, rise time and settling time.

2.4.5 The Spiral Optimization Algorithm Convergence Conditions and Settings

Paper [5] stated that in the conditions and settings under which the spiral optimization algorithm converge to stationary point. In this report, it covered the composite rotation matrix the step rate and the initial placement of search points, which characterize each spiral trajectory. Their effectiveness was mathematically proved and numerically verified. The algorithm has been studied as as a nature-inspired metaheuristic that aims to find a better approximated solution within a limited number of iterations specified by the user. In this paper, it showed that this algorithm can also be considered as a strict direct search method to find a stationary point from the proposed settings. The versatility of the spiral optimization algorithm can be determine by changing the conditions of the parameter settings.In the beginning , the spiral phenomena were intuitively considered as nature phenomena appropriate for metaheuristics considering that the behavior have both diversification and intensification. Currently, the spiral phenomena have been considered theoretically as a natural phenomenon appropriate for optimization.

2.4.6 PID Based Control of a Single-Link Flexible Manipulator In Vertical Motion with Genetic Optimization

Paper [10] is stated about PID based control of a FMS in vertical motion with genetic optimisation. It presents about the development of PID controller with PID and ILC feedback control strategy for vibration reduction based on genetic algorithms for FMS in vertical plane motion. The writer consider a combine feedforward and feedback control structure to control the rigid body and flexible motion dynamics. The genetic optimisation action is to design PID controllers in the forward and feedback paths of the control structure. The designation also incorporated with end-point acceleration feedback paths of the control structure. The performance of the developed control approach has been determined in comparison to previously reported PIDPID control and it has been illustrated the good performance is achieved with the proposed approach. The control sheme has shown to perform well in reducing the vibration at the end-point of the manipulation.

2.4.7 P, PD, PI PID CONTROLLER

Paper [11] is presented about P,PD,PI and PID controllers. Nowdays, P-I-D controller is commonly used in the industry. For controlling the first order plants , control engineers usually used PI controller. While to control two or higher order plants P-I-D is being chosen. Most cases for fast transient response and zero steady state error is desired for a closed loop system. These two specification conflicts with each other which makes the design harder. That why P-I-D is preffered it provides both of these feature at the same time.

2.4.8 Swarm Optimization of an Active Vibration Controller for Flexible Manipulator

Paper [12] is stated about swarm optimization (PSO) of an active vibration controller for flexible manipulator. The design of an optimum PID controller using PSO applied on FMS been presented. Initially, FMS has been modelled by PSO modelling techniques in order to obtain the transfer function of the system response. Through input/output mapping,mean square error and correlations test, a number of validation tests were carried out. Afterwards, hybrid PID controller was occupied for control of flexible manipulator. The optimum gains that acquired through global search of PSO techniques has been tested on the control structure. System responses including input tracking and vibration at the end-point has been evaluated. From the data simulaltion, the proposed controller has successfully position the flexible link to the desired position with reduction of vibration at end-point.

2.5 Summary of Literature Review

Based on the literature review, all the formulas and ideas had been used effectively in this project, so that the spiral dynamic optimization algorithm for engineering application can be done effectively.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter will discuss the process of the project in order to achieve all of the objectives in this research project that stated in Chapter 1. The flow process such as analysis and simulation will explain in the methodology. A systematic procedure needs to plan properly to ensure all of the objective can be achieved. This section will briefly the detail procedure that will perform during this research.

3.2 Methodology to Achieve Objective 1

The flow chart of this project is to arrange the process in the systematic way. The Figure 3.1 shows the process flow chart for this project on how to achieve objective 1 which is to investigate the performance of the SDA with differents types of numerical benchmark function. Firstly, a coding of this algorithm is written in Matlab script. Then the coding is being run using 10 different of numerical benchmark functions. In this project three analysis is being done. For the first analyis is testing on four different dimension. The second analysis is testing on different in number of points and third analysis is testing on different number of iteration. About 30 independent runs is being done for every analysis.

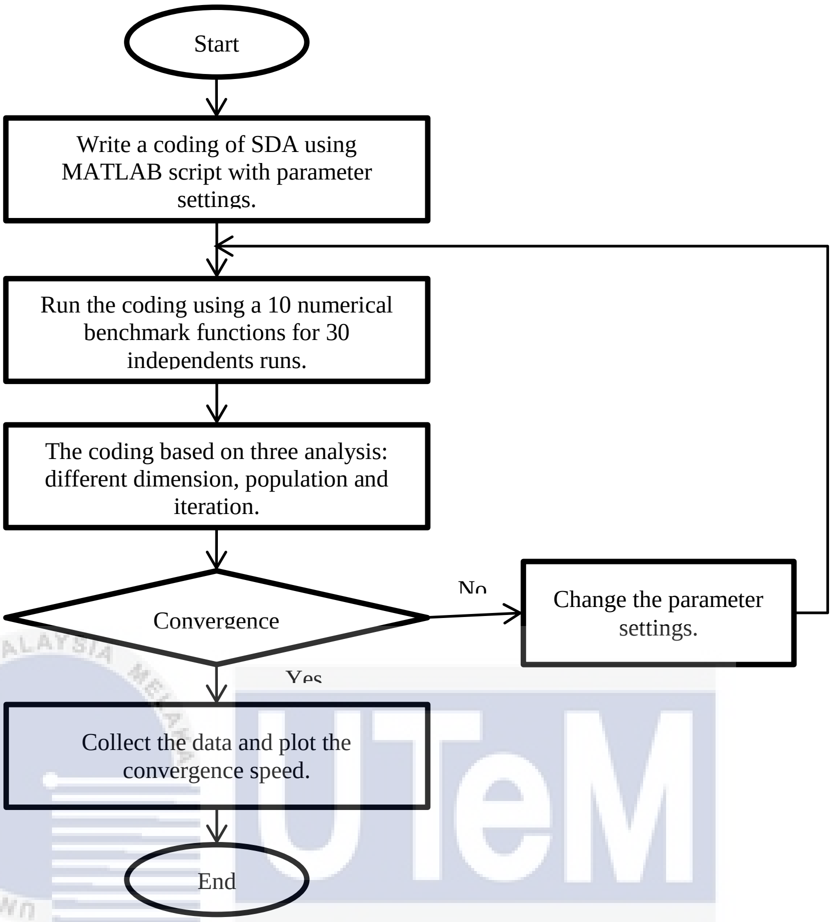


Figure 3.1: Methodology To Achieve Objective One

3.2.1 Benchmark Function Tests

The benchmark functions used in this analysis are adopted from the work presented by Biswas et al. (2007a), Dasgupta et al. (2009b) and Hedar (2004). Optimization benchmarking has many advantages which are can help to select the best algorithm for working with a real-world problem, to evaluate the performance of an optimization algorithm when different option setting are used and it also can compare a new version of optimization software with earlier releases. Ten different benchmark functions with various fitness landscapes comprising uni-modal and multi-modal features are used in the performance tests and these are described below in Table 3.1.

Table 3.1 : List Of Benchmark Functions

Benchmark Function	Formula	Range
1. Shubert Function	$f1\,(x) = \left(\sum_{I=1}^{5} i\cos((i+1)\ x_1 + i\right)\left(\sum_{I=1}^{5} i\cos((i+1)\ x_2 + i\right)$	[-10, 10]
2. Sum Of Different Power	$f2\,(\mathrm{x}) = \sum_{i=1}^{d} (x)_i^{i+1}$	[-1, 1]
3. Matyas Function	$f3\,(\mathrm{x}) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	[-10, 10]
4. Holder Table Function	$f4\,(\mathrm{x}) = -\left\lvert \sin x_1 \cos x_2\ \exp(1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}) \right\rvert$	[-10, 10]
5. Eggholder Function	$f5(\mathrm{x}) = -(x_2+47)\sin(\sqrt{\left\lvert x_2 + \frac{x_1}{2} + 47 \right\rvert}) - x_1 \sin\left(\sqrt{x_1 - (x_2+47)}\right)$	[-512, 512],
6. Cross-In-Tray Function	$f6(\mathrm{x}) = -0.0001(\lvert \sin x_1 \sin x_2\, exp\left(\left\lvert 100 - \frac{\sqrt{x_1^2 + x_1^2}}{\pi} \right\rvert\right) + 1$	[-10, 10]
7. Drop-Wave Function	$f7(\mathrm{x}) = -\frac{1+\cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{0.5(x_1^2+x_2^2)+2}$	[-5.12, 5.12]
8. Three-Hump Camel Function	$f8(\mathrm{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	[-5, 5]

9. Six-Hump Camel Function	$f9(\mathrm{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[-3, 3]
10. Booth Function	$f10(\mathrm{x}) = \left(x_1 + 2_{x_2} - 7\right)^2 + (2x_1 + x_2 - 5)^2$	[-10, 10]

3.3 Methodology to Achieve Objective 2

The Figure 3.2 below shows on how the process flow to achieve objective two which is to investigate the most suitable controller with error criteria for flexible manipulator system. This time the main coding of SDA execute with a function of flexible manipulator system to find the Kp, Ki and Kd. The values is used to tune the PI, PD and PID controller to find hub angle of the flexible manipulator system. About five different error have been used during tuning the controller. The five errors been used are Integral Squared Error (ISE), Integral Absolute Error (IAE) , Integral Time-weighted Absolute Error (ITAE), Mean-Square Error(MSE) and Root Mean Square Error (RMSE).

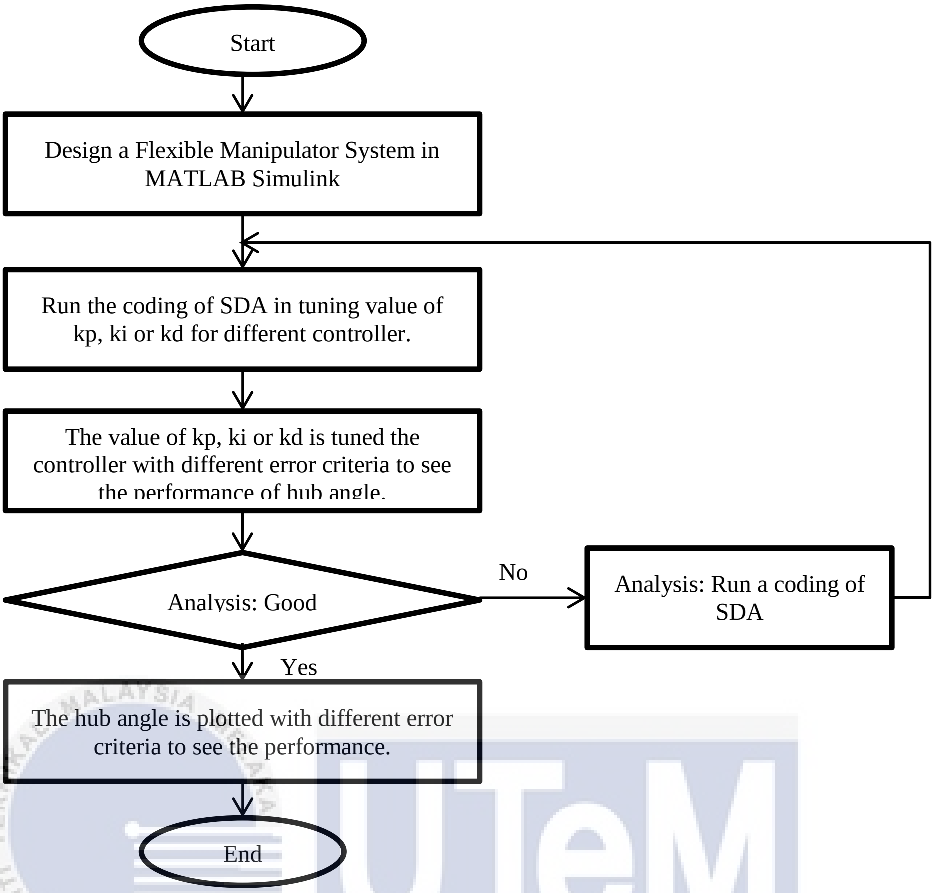


Figure 3.2: Methodology To Achieve Objective Two

3.3.1 Error Criteria

There are five commonly used measures are Integral Squared Error (ISE), Integral Absolute Error (IAE) , Integral Time-weighted Absolute Error (ITAE), Mean-Square Error (MSE) and Root Mean Square Error (RMSE).

3.3.1.1 Intergral Squared Error (ISE)

$$ISE = \int \varepsilon^2 \, dt \qquad 3.1$$

ISE integrates the square of the error over the time. Since the square of a large error will be much bigger, it will penalise large errors more than smaller ones. By minimize ISE will tend to eliminate large errors quickly, but it will tolerate small errors persisting for a long period of time. This leads to fast responses, but with considerable, low amplitude and oscillation.

3.3.1.2 Integral Absolute Error (IAE)

$$IAE = \int \varepsilon \, dt \qquad 3.2$$

IAE integrates the absolute error over time. It doesn't add weight to any of the errors in a systems response. It tends to produce slower response than ISE optimal system. It usually with less sustained oscillation.

3.3.1.3 Integral Time-Weighted Absolute Error (ITAE)

$$ITAE = \int t \, \varepsilon \, dt \qquad 3.3$$

ITAE integrates the absolute error multiplied by the time over time. It is to weight errors which exist after a long time much more heavily than those at the start of the response. The tuning of ITAE will produces systems which settle much more quickly than the other two tuning methods. The disadvantages of this error is the tuning also produces systems with slow initial response (necessary to avoid sustained oscillation)[15].

3.3.1.4 Mean Square Error (MSE)

$$MSE = \int t\,\varepsilon^2\,dt \qquad 3.4$$

MSE is simple to implement, regular to interpret as energy of a signal, which maintain symmetically and differentiability. It is universally used in signal processing.

3.3.1.5 Root-Mean Square Error (RMSE)

$$RMSE = \int \sqrt{t\,\varepsilon^2\,dt} \qquad 3.5$$

RMSE is the square root of the mean of the square of all of the error. It is very common and is considered an great general purpose error metric for numerical prediction.

3.4 Methodology to Achieve Objective 3

The figure 3.3 below shows on how the process flow to achieve objective three which to investigate the performance of SDA with parameter setting in tuning propotional-integral-derivative (PID) with error criteria for application flexible manipulator system.This time the main coding of SDA executed with a function of flexible manipulator system to find the Kp, Ki and Kd. Actually,the SDA parameter setting were being varied. Three different number of points are being used that are 2, 30 and 70. The values of Kp, Ki and Kd of every execution are used to tune PID controller to find hub angle of the flexible manipulator system.

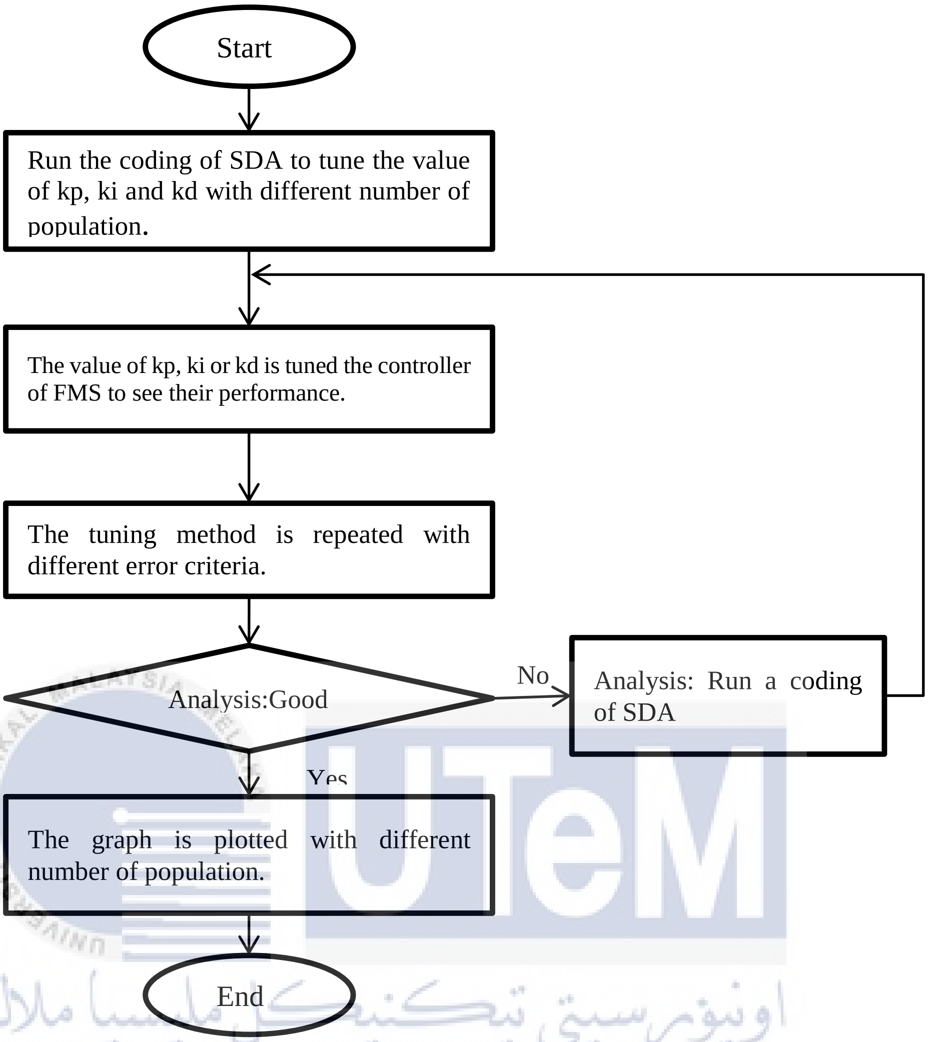


Figure 3.3 Methodology To Achieve Objective Three

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

In this chapter , the result that obtained from the simulation will be discussed.

4.2 Analysis of Different in Dimension

For analysis of different in dimension all of data has been tabulated in Table 4.1 and the convergence plot of the average best fitness against the iteration also shown in Figure 4.1.The graph below is obtained from 30 independent runs for every benchmark function. From the independent runs, we find average, standard deviation, best value, worst value and time processing.

Average : To find the average after doing 30 independent runs

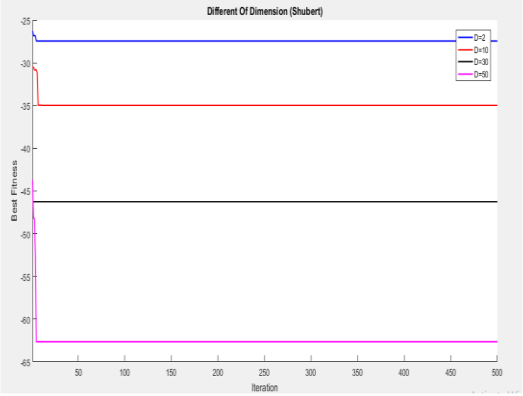
Standard Deviation: A measurement to express how much it deviate from average value (Robust).

From the figure 4.1 shows convergence plot of the average best fitness against the iteration, tested on four different dimension which are 2,10,30 and 50. Every dimension is tested about 30 independent runs. The best mean accuracy among 10 benchmark accuracy among 10 benchmark is highlighted in bold font. From the test conducted Sum of Different Power's benchmark function has achieved the best fitness from four different dimension. Smaller mean value indicated that the solution is closer to the global optimum solution and it is more accurate. The time processing shows increasing pattern when the value of dimension increase. So we can conclude that when the value of dimension is increse the number of complexity becomes getting harder to find the best global minimum point.

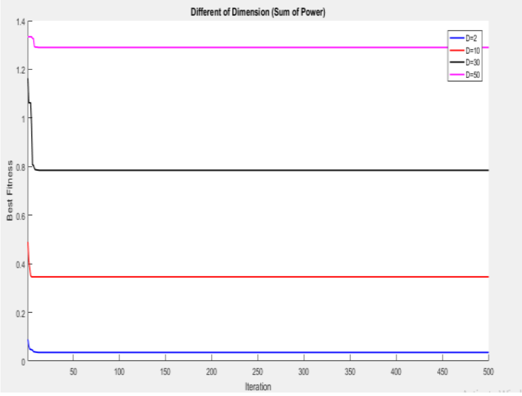
Table 4.1: The Value Of Different In Dimension

Function	Accuracy	Dimension			
		2	10	30	50
F1	Mean	-27.4464	-34.9748	-46.2588	-62.6486
	STD	14.8885	17.2355	26.6862	49.9850
	Best	-11.6264	-1.0466	-16.8615	-10.1735
	Worst	-46.5391	-71.874	-106.7868	-146.7051
	Time Processing	0.2311	0.3254	0.4587	0.5398
F2	Mean	**0.0354**	**0.3463**	**0.7848**	**1.2900**
	STD	0.0574	0.2030	0.5716	0.6750
	Best	0.0003	0.0541	0.0004	1.54E-01
	Worst	0.1893	0.5917	1.9601	2.3486
	Time Processing	0.3028	0.3954	0.4835	0.5214
F3	Mean	0.5547	0.7569	0.8698	1.0713
	STD	0.3109	0.7736	0.8826	1.0037
	Best	0.0514	0.2986	0.0588	0.2843
	Worst	0.9259	2.5499	2.6804	2.8086
	Time Processing	0.2982	0.3912	0.4789	0.5214
F4	Mean	-11.9759	-13.4428	-14.6540	-15.7655
	STD	4.3149	3.8803	2.9022	1.7819
	Best	-6.5235	-6.5066	-9.0326	-16.0803
	Worst	-18.6066	-16.2648	-17.3220	-16.9007
	Time Processing	0.3471	0.4235	0.4887	0.5324
F5	Mean	-561.3680	-579.3340	-653.4790	-698.4665
	STD	169.6668	185.9273	210.3927	178.7555
	Best	-251.5430	-343.6840	-344.7480	-312.3140
	Worst	-763.6145	-866.6630	-931.9550	-950.6751
	Time Processing	0.3175	0.3347	0.4521	0.5784
F6	Mean	-1.8559	-1.9215	-1.9552	-1.9887
	STD	0.0725	0.1047	0.1180	0.0632
	Best	-1.7446	-1.7541	-1.7389	-1.8774
	Worst	-1.9750	-2.0626	-2.0595	-2.0569

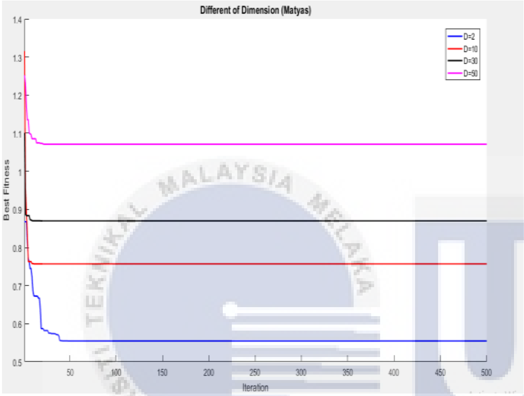
	Time Processing	0.3124	0.3457	0.4571	0.5541
F7	Mean	-0.5677	-0.6541	-0.6879	-0.6948
	STD	0.1709	0.1942	0.1651	0.1050
	Best	-0.2503	-0.3063	-0.4732	-0.6011
	Worst	-0.7782	-.0.9104	-0.9361	-0.9284
	Time Processing	0.2324	0.3255	0.4521	0.5201
F8	Mean	0.7969	1.2008	1.5216	2.1844
	STD	0.6537	1.4015	1.8003	1.9232
	Best	0.0199	0.1200	0.5402	0.1068
	Worst	2.3455	5.0513	6.8553	7.2492
	Time Processing	0.2574	0.3152	0.4251	0.5574
F9	Mean	0.1556	0.4886	1.0962	1.2798
	STD	1.0177	0.3582	4.8270	3.0037
	Best	-0.0930	0.0262	-1.0030	-0.9262
	Worst	2.2871	1.0423	15.5960	9.8760
	Time Processing	0.3021	0.3458	0.4251	0.4897
F10	Mean	4.0937	15.6930	21.3076	25.9565
	STD	5.6448	14.7964	19.5844	29.3217
	Best	0.6242	0.2572	3.0288	0.3812
	Worst	19.1611	38.9075	64.3256	88.7760
	Time Processing	0.3784	0.4021	0.4719	0.5012



a) Shubert

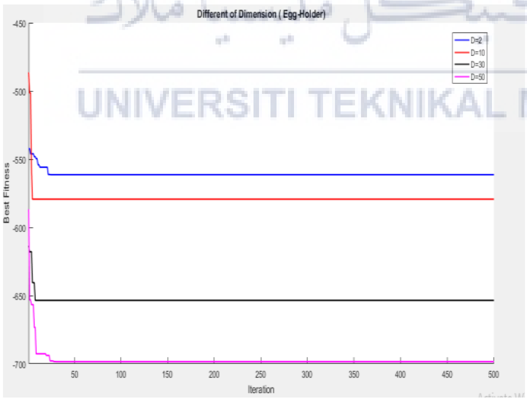


b) Sum Of Power Different

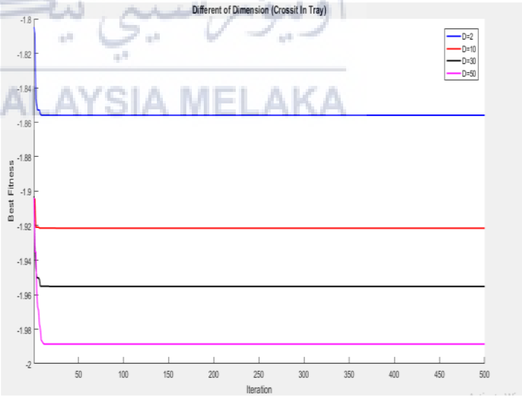


c) Matyas

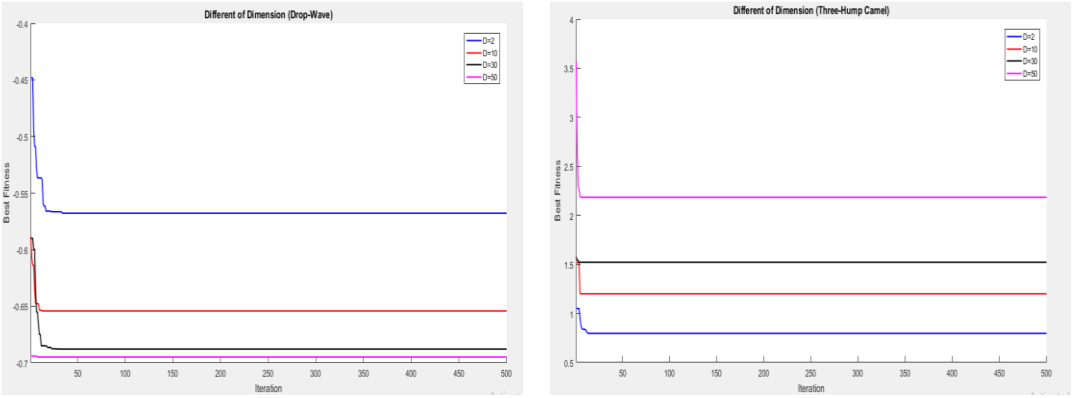
d) Holder Table



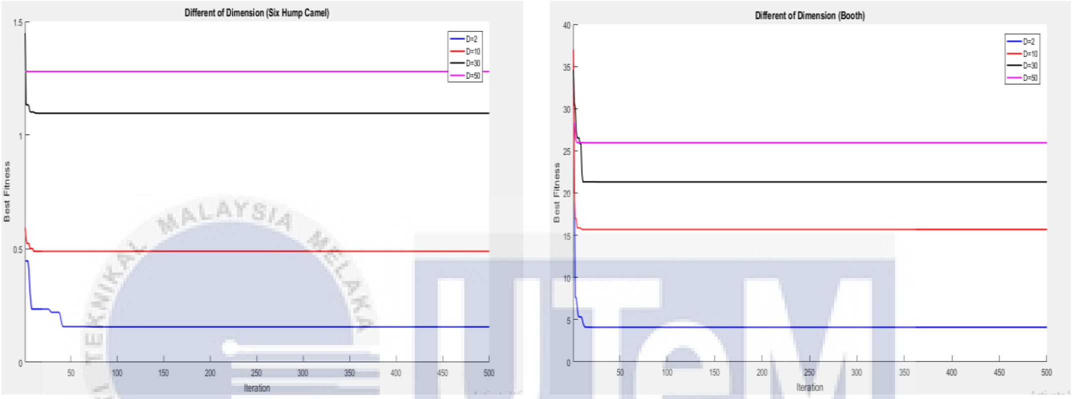
e) Eggholder



f) Cross-In-Tray



g) Drop-Wave h) Three-Hump Camel



i) Six-Hump Camel j) Booth

Figure 4.1 :Different In Dimension

4.3 Analysis of Different Number Of Points

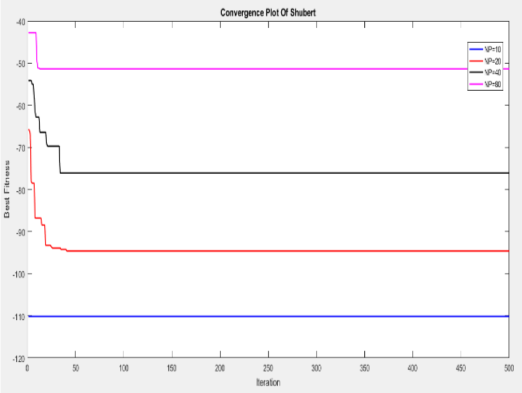
For analysis of different in number points all of data has been tabulated in Table 4.2 and the convergence plot of the average best fitness against the iteration also shown in Figure 4.2 The graph below is obtained from 30 independent runs for every benchmark function. From the independent runs, we find average, standard deviation, best value, worst value and time processing.

From the figure 4.2 shows convergence plot of the average best fitness against the iteration, tested on four different number of points which are 10, 20, 40 and 80. Every number of points is tested about 30 independent runs. Notice that, Sum of Different Power's benchmark has achieved the best fitness for overall number of points. The best mean is being in bold font. The second best performance was achieved by Six-Hump Camel benchmark function. Both of them shows smaller standard deviation which is indicates that the generated solutions tends to be very close to mean value. It shows that the variation or consistency of generated solution from the average value in other words the solution is robust. In conclusion, increasing in number of points , the search point is more towards the best global minimum point.

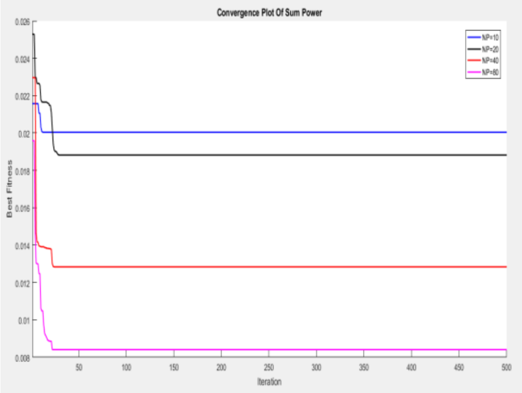
Table 4.2: The Value Different In Number Of Points

Function	Accuracy	Number Of Points			
		10	20	40	80
F1	Mean	-51.3785	-76.1137	-94.6208	-110.1410
	STD	50.2328	49.3878	53.4569	39.4357
	Best	-10.6647	-20.4897	-21.4585	-52.3829
	Worst	-163.9250	-184.6380	-185.244	-16.4350
	Time Processing	0.3122	0.4251	0.5784	0.7452
F2	Mean	**0.0084**	**0.0128**	**0.0188**	**0.0200**
	STD	0.0110	0.0135	0.0141	0.0215
	Best	0.0019	2.87E-05	3.55E-03	0.0019
	Worst	0.3905	0.0192	0.0385	0.0826
	Time Processing	0.2927	0.4257	0.5834	1.0714
F3	Mean	0.1556	0.1711	0.1904	0.2047
	STD	0.1365	0.1735	0.1741	0.0744
	Best	0.0039	0.0023	0.0216	0.1324
	Worst	0.4849	0.5496	0.5419	0.2476
	Time Processing	0.2284	0.3045	0.5914	0.8574
F4	Mean	-12.6889	-14.4313	-15.3913	-16.0322
	STD	5.1748	4.2892	3.2346	1.6650
	Best	-4.4628	-5.5356	-9.1925	-12.9614
	Worst	-19.0322	-18.9517	-18.9958	-18.7116

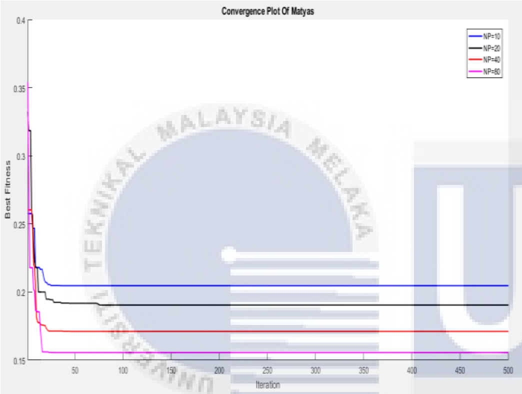
	Time Processing	0.3714	0.4784	0.5514	0.9472
F5	Mean	-562.3230	-634.2380	-693.2580	-701.7440
	STD	155.7266	192.0116	119.9597	128.9241
	Best	-341.6760	-352.8110	-492.4170	-531.0260
	Worst	-787.9140	-934.8690	-934.9970	-887.8110
	Time Processing	0.3171	0.4102	0.7737	0.8408
F6	Mean	-1.8548	-1.9384	-1.9968	-2.0260
	STD	0.0901	0.0951	0.0603	0.0542
	Best	-1.7407	-1.8707	-1.9056	-1.8721
	Worst	-1.9639	-2.0620	-2.0520	-2.0626
	Time Processing	0.3010	0.4457	0.5257	1.0162
F7	Mean	-0.5485	-0.6006	-0.6964	-0.7766
	STD	0.2676	0.1927	0.1673	0.2165
	Best	-0.2289	-0.2887	-0.4592	-0.4887
	Worst	-0.9325	-0.9266	-0.9184	-1.0000
	Time Processing	0.3239	0.3926	0.5471	0.9328
F8	Mean	0.3551	1.0323	1.2560	1.5465
	STD	0.1881	0.8366	1.2697	1.5437
	Best	0.0245	0.0248	0.1477	0.4264
	Worst	0.7257	2.4422	3.4884	3.0704
	Time Processing	0.3041	0.4729	0.5981	0.9301
F9	Mean	-0.0151	-0.1397	-0.5121	-0.7696
	STD	0.6861	0.6375	0.3297	0.2435
	Best	-0.8834	-0.8499	-0.9274	-0.2981
	Worst	1.2415	1.1704	0.6694	-1.0006
	Time Processing	0.2910	0.4491	0.5616	0.8374
F10	Mean	2.6532	3.9800	4.6470	5.4496
	STD	2.3917	3.1878	7.1794	3.3598
	Best	0.0936	0.4371	1.5852	2.7962
	Worst	6.6328	11.0608	26.1243	13.7488
	Time Processing	0.3753	0.4098	0.7231	1.5115



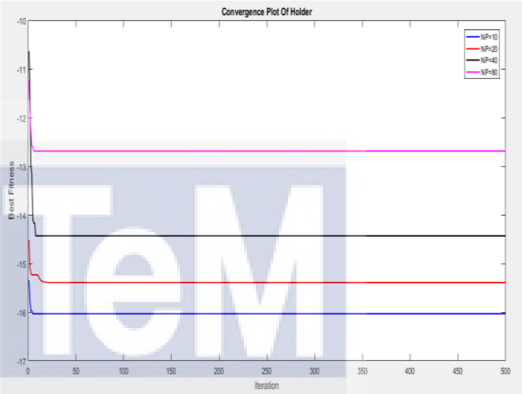
a) Shubert



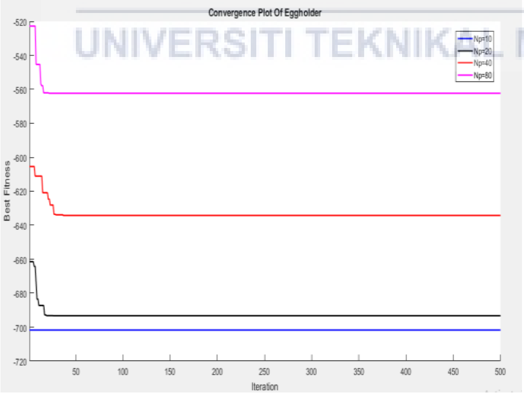
b) Sum Of Power Different



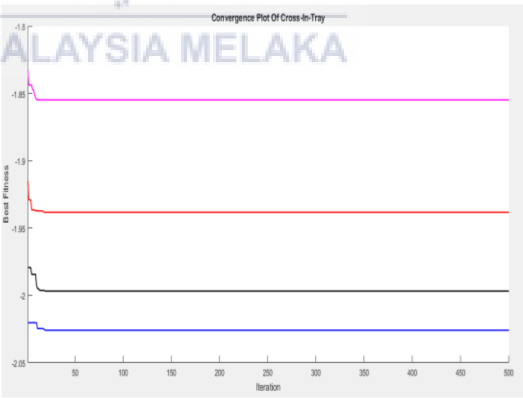
c) Matyas



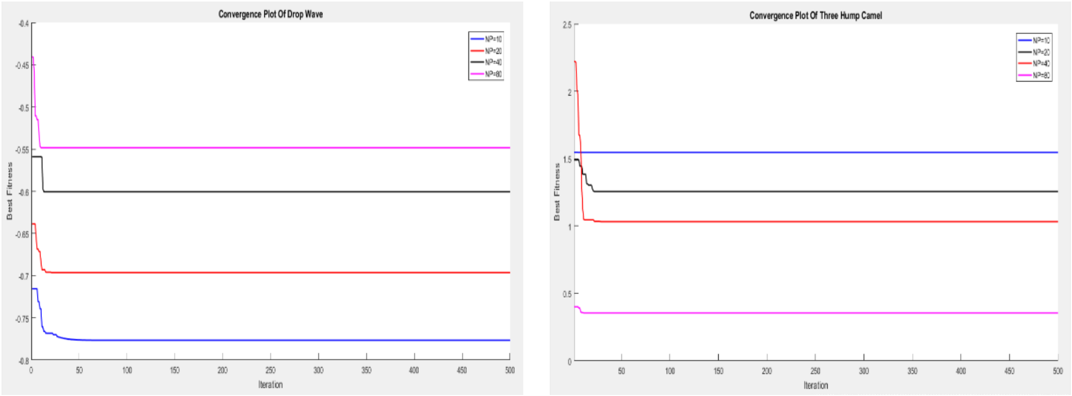
d) Holder Table



e) Eggholder



f) Cross-In-Tray



g) Drop-Wave h) Three-Hump Camel

i) Six-Hump Camel j) Booth

Figure 4.2: Different In Number Of Points

4.4 Analysis of Different Number of Iteration

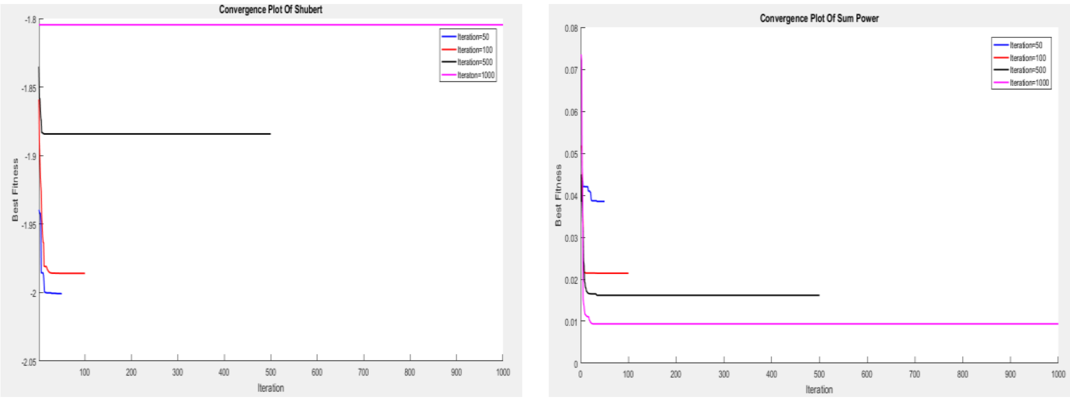
For analysis of different in number of iteration all of data has been tabulated in Table 4.3 and the convergence plot of the average best fitness against the iteration also shown in Figure 4.3.The graph below is obtained from 30 independent runs for every benchmark function. From the independent runs, we find average, standard deviation, best value, worst value and time processing.

From the Figure 4.3 shows convergence plot of the average best fitness against the iteration, tested on four different number of iteration which are 50, 100, 500 and 1000. Every number of points is tested about 30 independent runs. Sum of Different Power’s benchmark function shows the best fitness value among the others. From the convergence plot after a certain number of iteration the graph shows the best fitness is trapped on the local optimum which is in SDA all the search point will settle at a point so adding any further certain number of iteration will not make the algorithm to drastically converge to higher accuracy and better point. The effectiveness of the algorithm depends on the iteration. Notice that, during number of iteration is 1000 it closer to best global minimum point. So the conclusion, increase in iteration thes search point is lie on the best global minimum point.

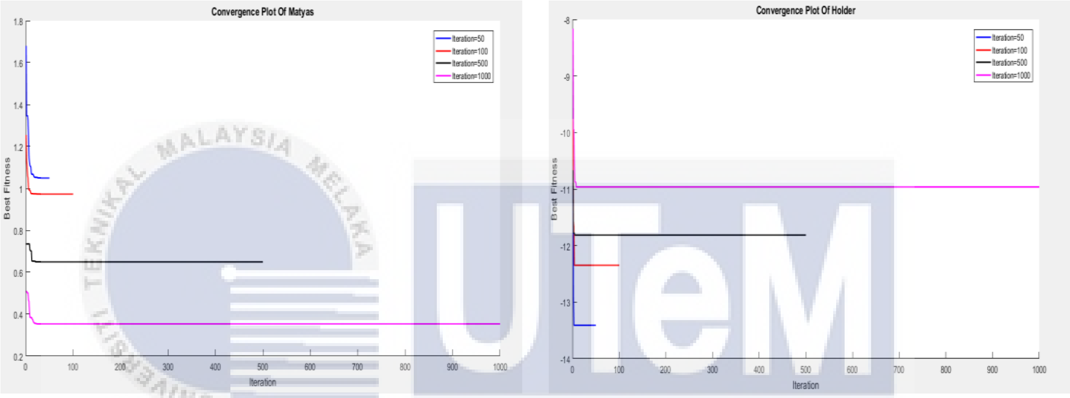
Table 4.3: The Value Different In Number Of Iteration

Function	Accuracy	Number Of Iteration			
		50	100	500	1000
F1	Mean	-62.6351	-53.3245	-35.0006	-30.3115
	STD	50.9178	51.4481	21.1869	14.4997
	Best	-23.1371	-13.1723	-11.8289	-6.8793
	Worst	-186.1645	-170.2630	-69.2794	-47.5948
	Time Processing	0.0714	0.1014	0.2071	0.3214
F2	Mean	**0.0385**	**0.0214**	**0.0162**	**0.0094**
	STD	0.0271	0.0246	0.0192	0.0138
	Best	0.0009	0.0001	0.0003	0.0003
	Worst	0.0777	0.0826	0.0551	0.0496
	Time Processing	0.0874	0.1025	0.1985	0.2415
F3	Mean	1.0492	0.9729	0.6493	0.3535
	STD	0.7051	1.0566	0.7819	0.2644
	Best	0.2371	0.1092	0.0094	0.1317
	Worst	2.4937	2.7776	2.0207	0.9865
	Time Processing	0.0954	0.1450	0.2178	0.3251
	Mean	-10.9625	-11.8135	-12.3483	-13.4091
	STD	5.0073	5.0661	5.5428	4.3557
	Best	-3.4944	-5.0137	-4.0877	-4.2356

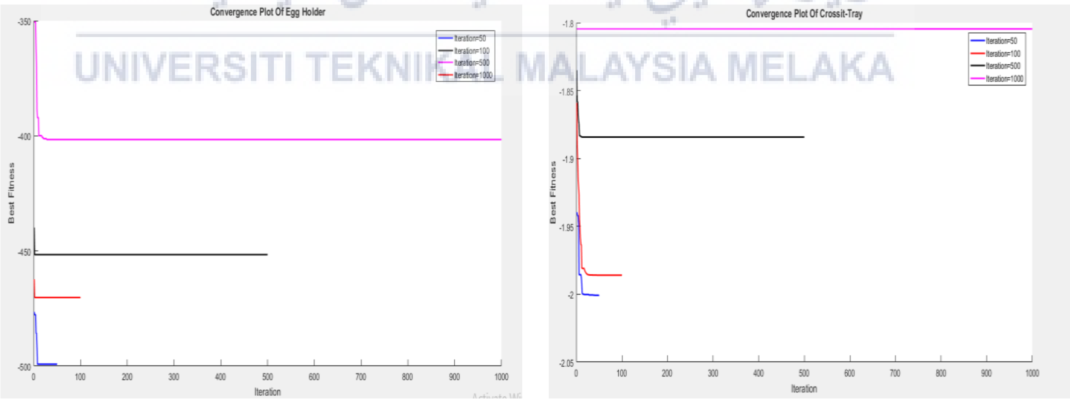
F4	Worst	-15.1402	-17.8583	-18.9294	-18.2159
	Time Processing	0.0812	0.1475	0.2357	0.3587
F5	Mean	-499.1510	-470.2145	-451.6045	-401.6231
	STD	171.5640	79.6414	145.2678	122.2205
	Best	-245.9880	-311.3221	-317.6060	-183.6661
	Worst	-737.4610	558.2780	-796.0361	-564.6438
	Time Processing	0.0984	0.1087	0.2148	0.3201
F6	Mean	-2.0008	-1.9859	-1.8843	-1.8046
	STD	0.0827	0.0748	0.1387	0.0563
	Best	-1.8508	-1.8405	-1.6400	-1.7111
	Worst	2.0625	-2.0620	-2.0387	-1.8772
	Time Processing	0.0927	0.1057	0.1806	0.3015
F7	Mean	-0.6215	-0.5799	-0.5031	-0.4785
	STD	0.2537	0.2314	0.2443	0.1824
	Best	-0.2693	-0.2702	-0.1491	-0.2212
	Worst	-0.9359	-0.9291	-0.8602	-0.7851
	Time Processing	0.0987	0.1058	0.2874	0.3145
F8	Mean	2.4867	2.1721	1.8507	0.9912
	STD	3.1124	3.1720	1.0863	1.0106
	Best	0.0453	0.1710	6.48E-05	1.06E-06
	Worst	9.8638	11.4351	3.8145	3.6049
	Time Processing	0.0982	0.1458	0.2574	0.3514
F9	Mean	0.5072	0.4697	0.2895	0.1645
	STD	0.9044	1.8065	1.0811	0.9338
	Best	-0.2863	-0.0801	-0.0918	-0.1741
	Worst	2.4800	5.1113	2.7806	1.9350
	Time Processing	0.0823	0.1874	0.2541	0.3817
F10	Mean	10.3064	8.4217	5.9131	3.0976
	STD	13.4167	9.0032	6.3055	2.0907
	Best	1.0782	1.6788	0.2899	0.0203
	Worst	46.5531	26.6084	11.3054	6.8210
	Time Processing	0.0974	0.1582	0.2654	0.3251



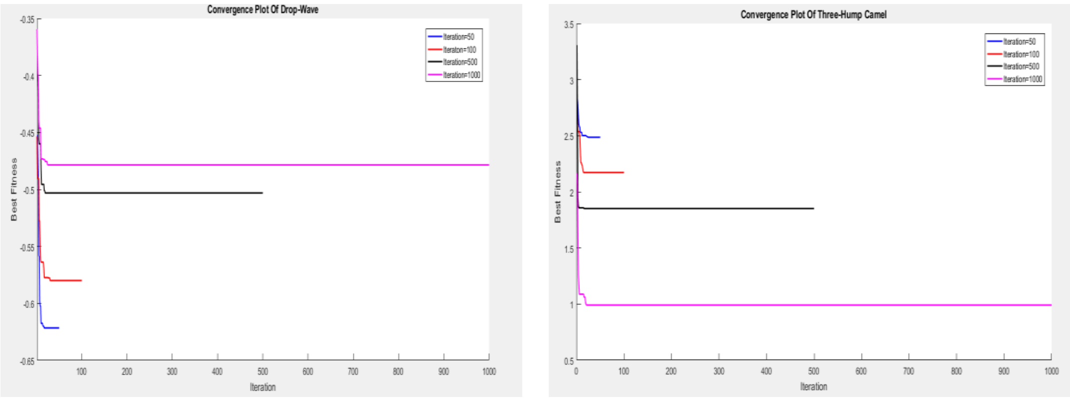
a) Shubert b) Sum Of Power Different



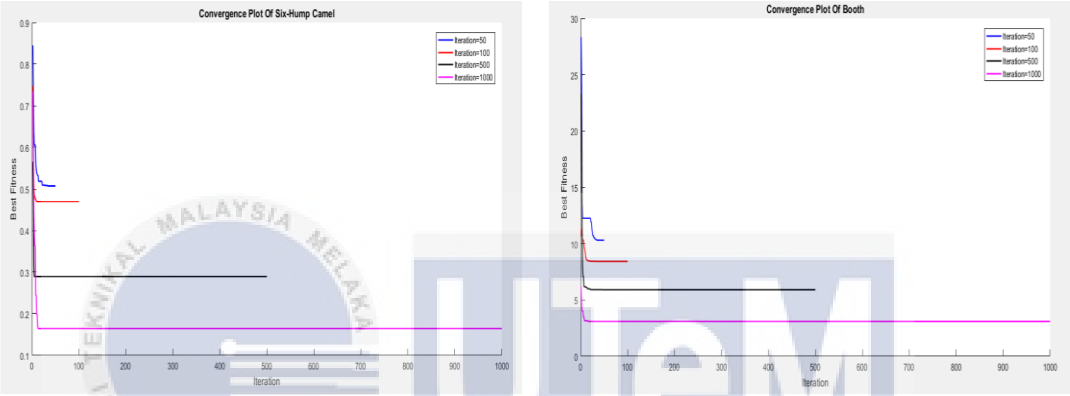
c) Matyas d) Holder Table



e) Eggholder f) Cross-In-Tray



g) Drop-Wave h) Three-Hump Camel



i) Six-Hump Camel j) Booth

Figure 4.3: Different In Number Of Iteration

4.5 Analysis Of The Most Suitable Controller With Error Criteria For Flexible Manipulator System

Figure 4.4 shows the FMS in Simulink/Matlab. The input of FMS is analog voltage (motor torque). The output of FMS are hub-angle, hub-velocity and end acceleration.

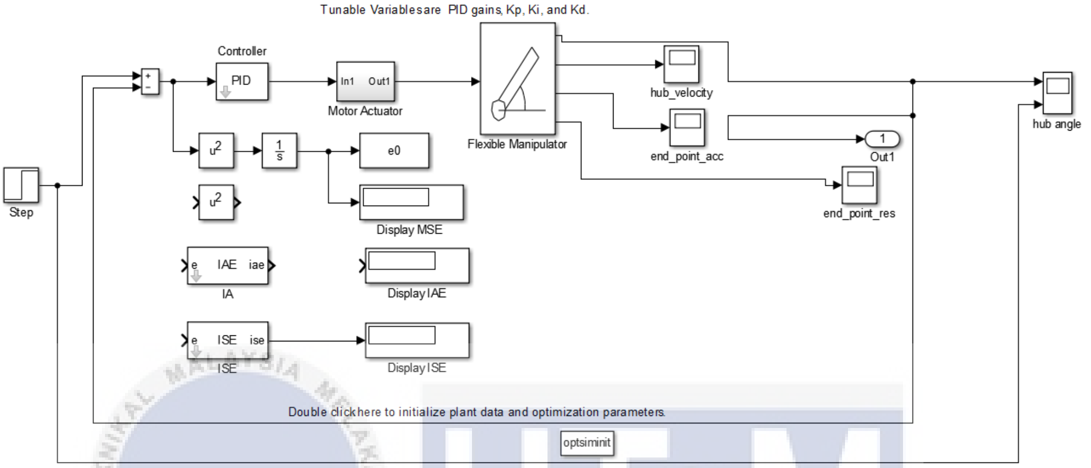


Figure 4.4: Simulink Of FMS

Figure 4.5 indicates on how the SDA tune PI, PD and PID controller to get the output. For this project FMS is selected as the target application and platform to test the performance of the algorithm.

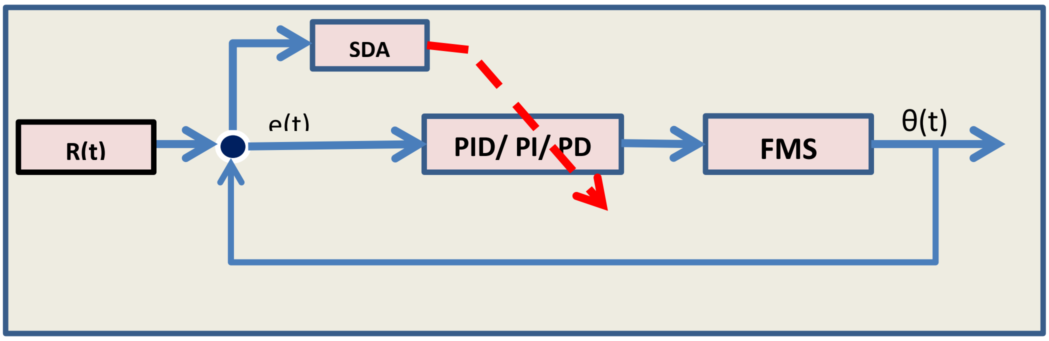


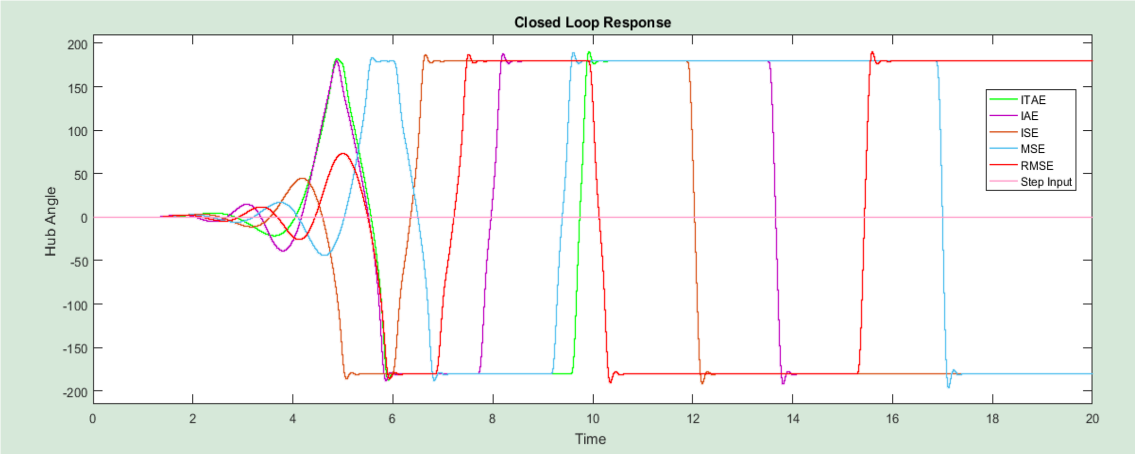
Figure 4.5 : SDA Tune PI, PD and PID Controller

This section presents the result of optimization of PI, PD and PID controller using SDA . SDA is design to tune the values of propotional gain (Kp), derivative gain (Kd) and integral gain (Kd) until it reach the optimum value. There are several objective function that be considered for tuning these parameter such as mean squared error (MSE), root mean suared (RMSE), integral of squared error (ISE), integral absolute error (IAE) and integral time absolute error (ITAE). Overshoot, rise time , settling time have been tabulated to see the performance of the system. The algorithm parameter were set as number of points,Np = 70, rotational angle,$\emptyset = \frac{\pi}{4}$, spiral radius r=0.95 , number of dimension ,D=3 and number of iteration, I=50.

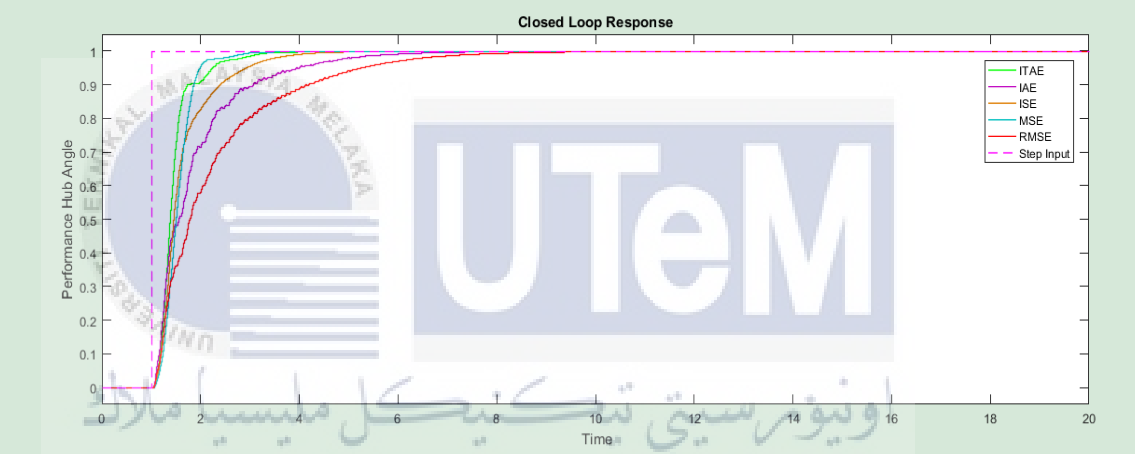
Figure 4.7 shows the convergence of the SDA during tuning PI,PD and PID controllers.The best individual achieved on ITAE with PD controller. Table 4.4 shows numerical values of the performance of the system with three control approches. It is noted that PD and PID gave the best overshoot output 0.4530% and 17.0590% respectively. Even the value of overshoot of PI controller is (2.6040%) lower than PID , the steady state of PID is the lowest than PI controller and PD controller. It shows that the difference between input and the output of the system in the limit goes to infinity is the best for PID controller. PI controller does not have steady state error is because of it does not have derivative gain,which its help to increase the stability of the system, reducing overshoot and improving transient response. The presence of the integral gain can eliminate the steady state error and it make the transient more worse. The rise time of PI controller is 0.5400s while for PID is 0.9680s. From the data, we can conclude that the rise time for PI controller takes for the response to rise from 10% to 90% of the steady state error is much lower. From the Figure 4.6 and Table 4.4 we can conclude short rise time shows a good result but a fast response usually comes at the cost of incresed overshoot and oscillation [11]. In conclusion, the most suitable controller for FMS is PID controller because the presence of proportional gain can improve the rise time, presence of derivative gain can reduce the overshoot and presence of integral gain can reduce the steady-state error. Because of that, moderate peak overshoot, moderate stability and lower steady state error can be achieve.

Table 4.4: Numerical Values Of The Performance Of The System

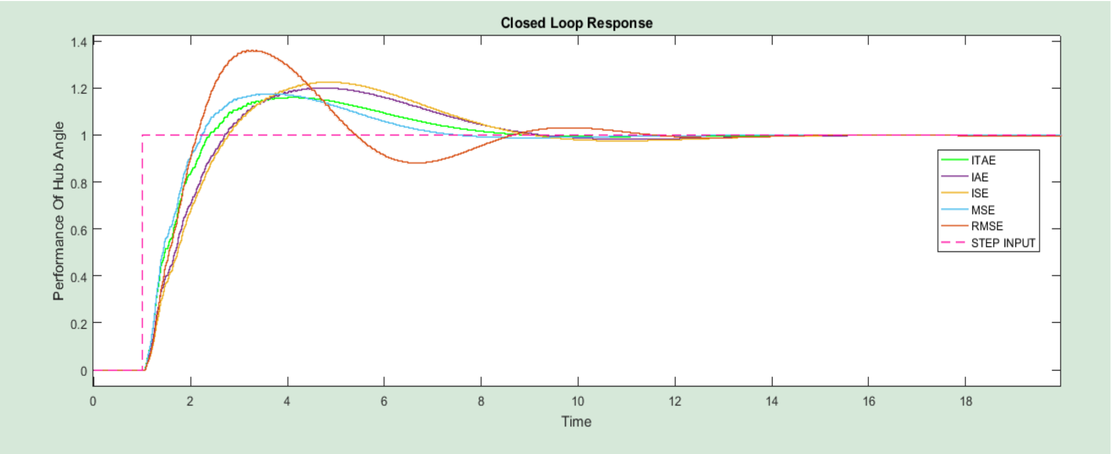
Controller	Parameter	Error Criteria				
		ITAE	IAE	ISE	MSE	RMSE
PI	Kp	0.3023	2.4063	0.7227	1.2220	1.7781
	Ki	2.0401	2.8034	2.2471	1.7801	1.6697
	Overshoot (%)	2.6040	2.6320	2.6320	2.6880	2.1990
	Rise Time (s)	0.2160	0.3390	0.4930	0.2880	0.3300
PD	Kp	1.7611	2.9492	1.3504	0.8819	2.0314
	Kd	0.5934	1.8329	0.5912	0.3145	1.7728
	Overshoot (%)	0.4530	0.5000	0.5020	0.4950	0.5040
	Rise Time (s)	0.5400	1.8220	1.1630	0.6140	2.9090
	Settling Time (s)	3.2000	6.5450	5.1510	3.7190	7.5600
	Steady State Error	0.0030	0.0040	0.0030	0.0010	0.0020
PID	Kp	2.7034	2.0401	1.4951	2.5266	1.4934
	Ki	1.2827	1.0081	0.7899	1.4819	2.0492
	Kd	1.9071	1.7897	1.4123	1.5145	1.3329
	Overshoot (%)	17.0590	19.8800	22.8400	18.4200	36.3010
	Rise Time (s)	0.9680	1.2140	1.2600	0.7940	0.7870
	Settling Time (s)	7.1620	12.0160	11.6800	7.8280	13.6110
	Steady State Error	0.0010	-0.0090	-0.0040	-0.0090	0.0020



Performance Of PI Controller



Performance Of PD Controller



Performance of PID Controller

Figure 4.6: Performance of Controller

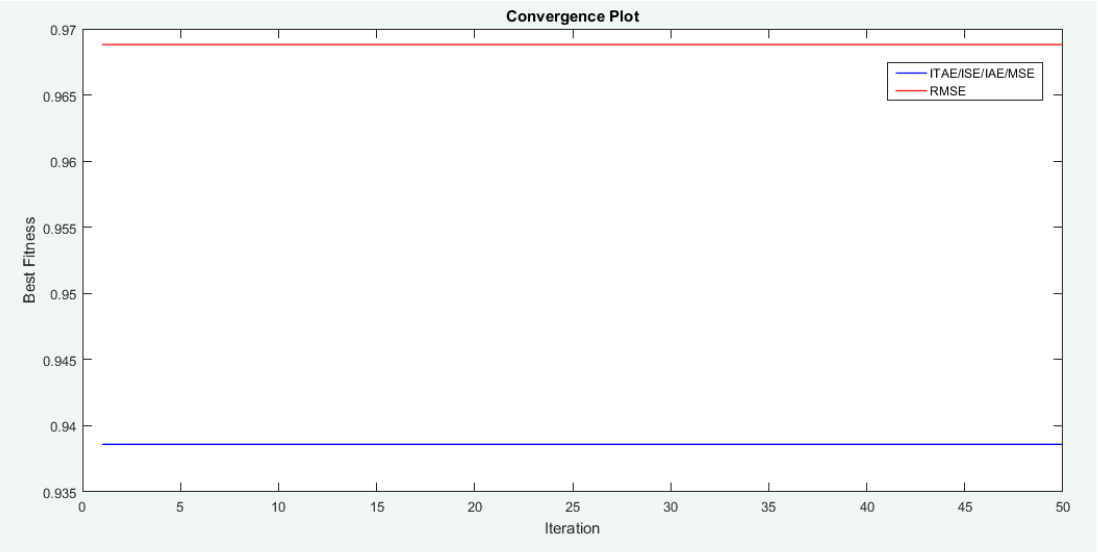
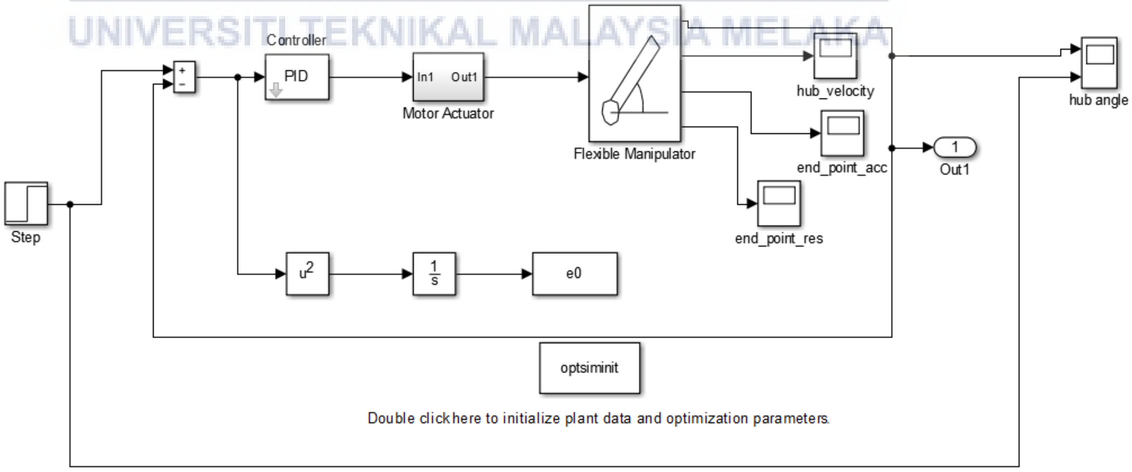


Figure 4.7 Hub-angle Output (Controller)

4.6 Analysis Of The Performance Of SDA With Parameter Setting In Tuning PID Controller With Error Criteria For Application Flexible Manipulator System.

Figure 4.8 shows the FMS in Simulink/Matlab. The input of FMS is analog voltage (motor torque). The output of FMS are hub-angle, hub-velocity and end acceleration.

Tunable Variables are PID gains, Kp, Ki, and Kd.



Double click here to initialize plant data and optimization parameters.

Figure 4.8: Simulink of FMS for PID Controller

Figure 4.9 indicates on how the SDA tune PID controller to get the output. For this project FMS is selected as the target application and platform to test the performance of the algorithm.

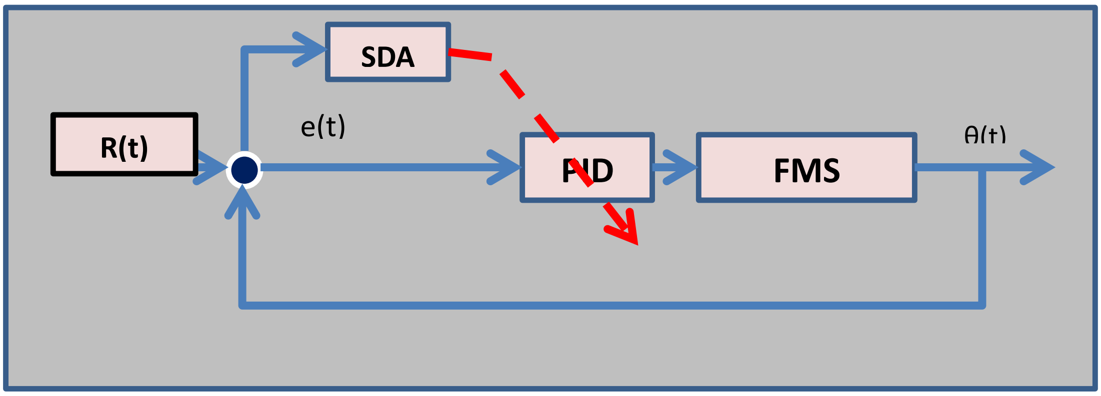


Figure 4.9: SDA Tune PID Controller

SDA is used to optimized PID controller of FMS. A PID controller attempts to correct the error between a measure process variable and desired set point by calculating and then outputting a corrective action that can modify the process [15]. It has optimum control dynamics including zero-steady state error,fast response (short rise time), no oscillation and higher stability. The advantages of the PID controller is that it also deals with important practial issues such as actuator saturation and integrator windup. Input and output data from the simulation are collected and used with SDA to obtain suitable parameter setting based on the performance of the system.

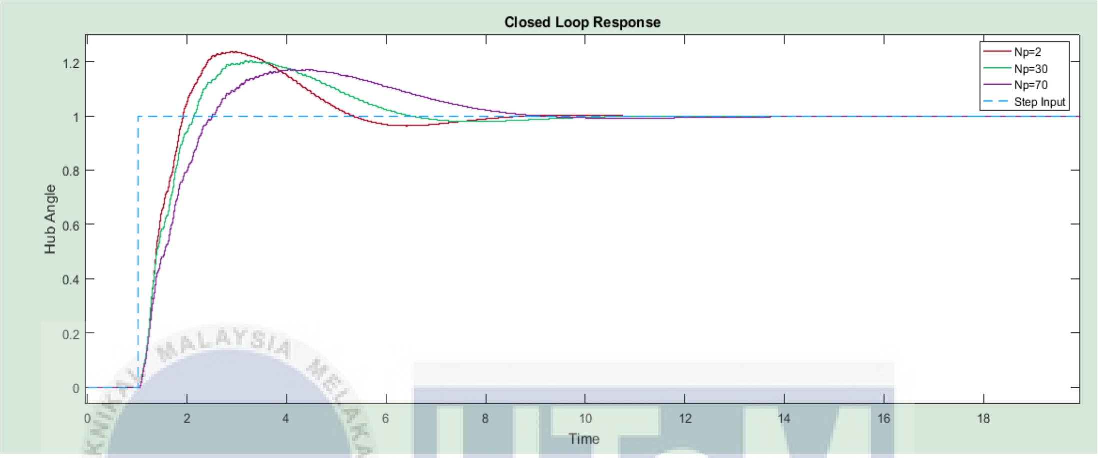
In this section the testing of FMS performance with PID controller with different parameter setting is presented and discussed. The algorithm parameter were varied of three different number of points (Np). The Np being used are 2,30 and 70. While the number of iteration, I=50, rotational angle, $\emptyset = \frac{\pi}{4}$, spiral radius, r=0.95 and dimension,D=3.

Figure 4.11 shows hub-angle response of the FMS with PID controller with different parameter settings. As noted, overall overshoot for every error criteria are decreasing along with increasing in Np. While, the rise time getting slower because the complexity becomes increase. The best individual achieved of different in Np that shown in Figure 4.10 is Np=70. That because of the performance of hub angle gave the best maximum overshoot for Np =70 is about 11.7980% when using ISE error criteria while tuning PID controller. The steady state error shows it getting decreasing along with Np is increasing so we may conclude the difference between input and output of FMS system is faster on reaching steady state error. The settling time for less Np is higher than for higher Np. Therefore, the response of the system was settled faster, resulting the good performance of the FMS [14].

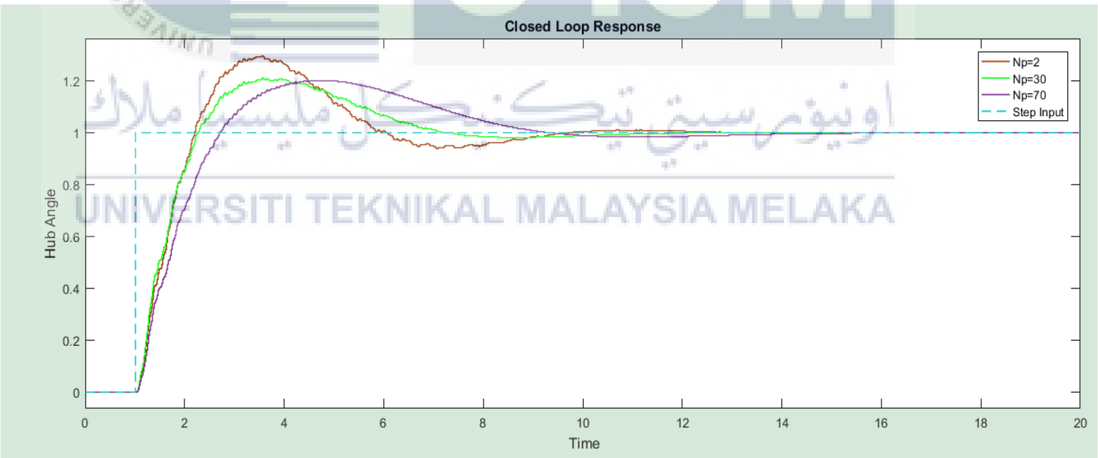
Table 4.5: Numerical Values Of The Performance with Different Parameter Setting

Error Criteria	Parameter	Number Of Points		
		2	30	70
ITAE	Kp	2.4899	2.0140	2.7034
	Ki	2.3671	2.9771	1.2827
	Kd	1.3566	1.6757	1.7507
	Rise Time (s)	0.6450	0.7290	0.9680
	Settling Time (s)	10.2480	10.9410	9.1620
	Overshoot (%)	24.3750	21.3410	17.0590
	Steady State Error	0.0020	0.0100	-0.0010
IAE	Kp	2.7345	2.9751	2.0401
	Ki	2.6982	1.9179	1.0081
	Kd	2.2289	2.0499	1.7897
	Rise Time (s)	0.8850	0.9380	1.2140
	Settling Time (s)	13.0550	12.0160	11.9750
	Overshoot (%)	29.2210	21.3410	19.8800
	Steady State Error	0.0020	0.0100	-0.0010
ISE	Kp	2.4729	2.4097	2.3976
	Ki	1.9110	1.1152	0.7890
	Kd	0.9831	1.6719	1.2599
	Rise Time (s)	0.4780	1.0250	0.8540
	Settling Time (s)	9.1590	9.0400	9.0310
	Overshoot (%)	18.4520	17.0590	11.7980
	Steady State Error	0.0040	0.0020	-0.0010
MSE	Kp	2.0478	2.8309	2.5266
	Ki	1.9086	1.7858	1.4819
	Kd	1.0675	1.9653	1.5145
	Rise Time (s)	0.6340	0.9400	0.7940
	Settling Time (s)	9.1750	9.140	9.120
	Overshoot (%)	22.840	21.3410	18.452
	Steady State Error	0.0000	0.0010	-0.0010
RMSE	Kp	1.3831	1.7611	1.7445

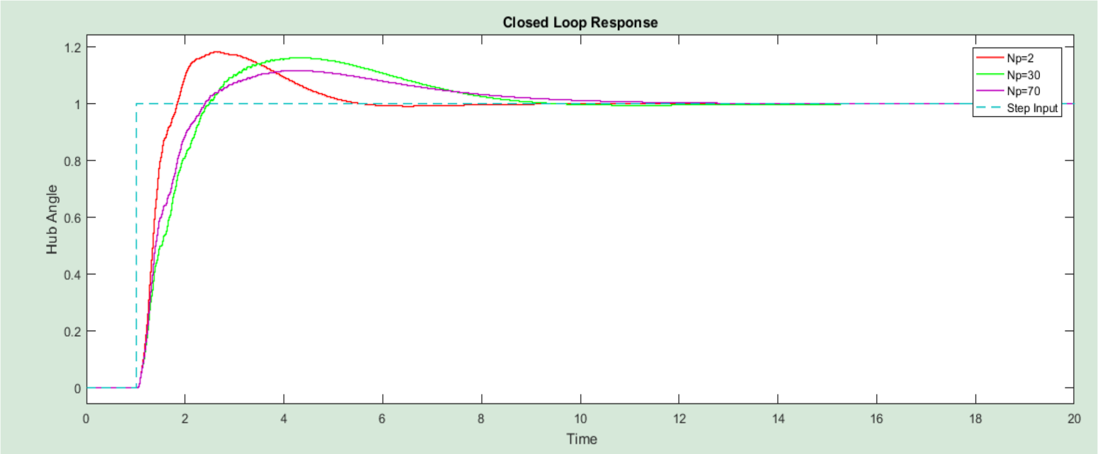
	Ki	2.8021	1.5236	0.5918
	Kd	1.3117	2.1637	2.0602
	Rise Time (s)	0.6800	1.2160	1.7660
	Settling Time (s)	11.7800	11.7190	11.7100
	Overshoot (%)	46.3240	34.4590	21.3410
	Steady State Error	0.0120	0.0040	0.0040



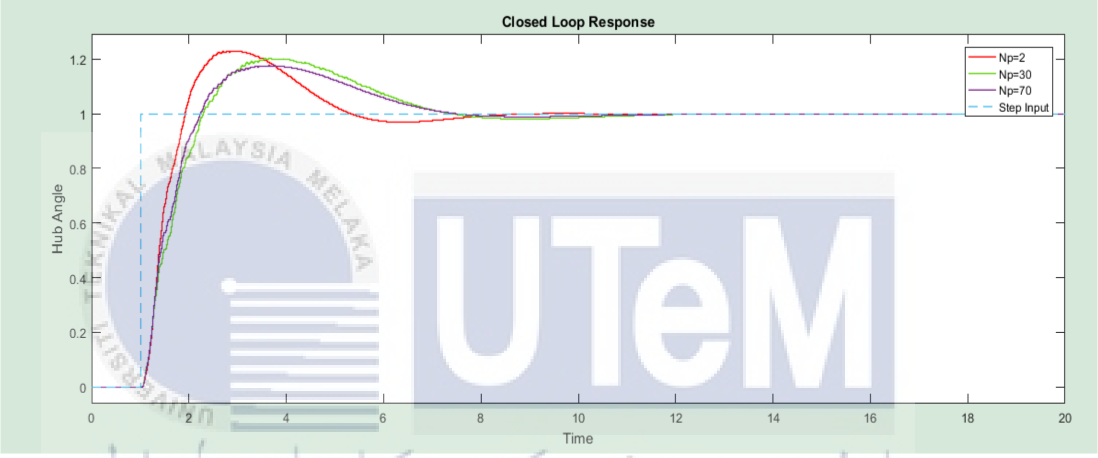
Performance Of Hub-Angle For ITAE



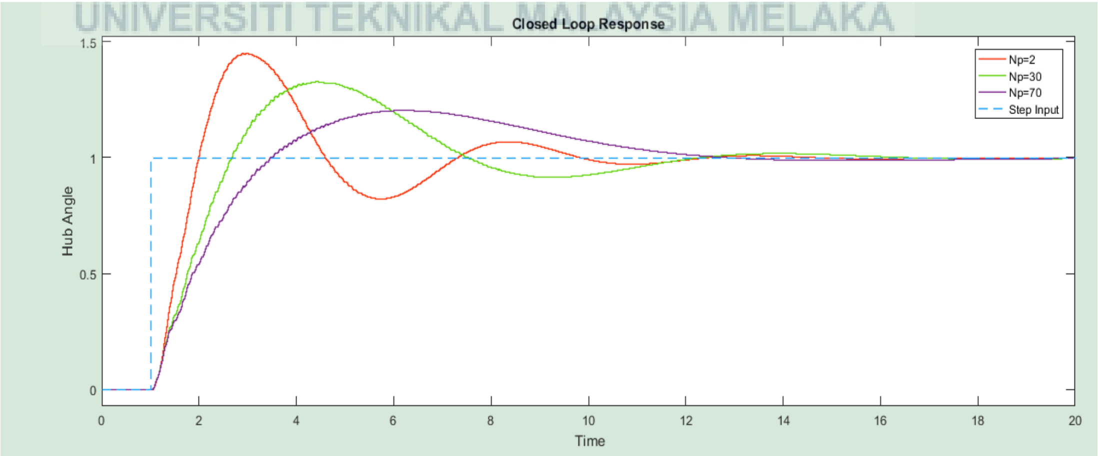
Performance of Hub-Angle for IAE



Performance Of Hub-Angle For ISE



Performance Of Hub-Angle For MSE



Performance Of Hub-Angle For RMSE

Figure 4.10 Performance Of Controller With Different Parameter Setting

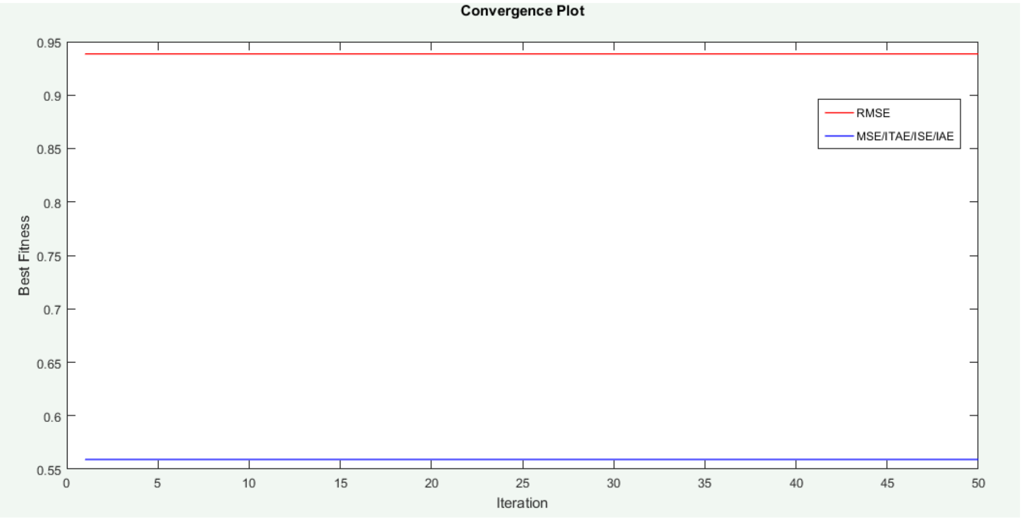


Figure 4.11 Hub-Angle Output (Np)

The convergence plot represents the performance of FMS in state of hub angle. From the graph we can conclude that it was trap at the local optima. Better parameter setting in algorithm will lead to the better output which is nearer to the zero that shown the best fitness obtained.

CONCLUSION

SDA is inspired from spiral phenomena in nature such as tornado and nautilis shell . The radius and rotation angle are two parameters that need to be controlled well. The search strategy of SDA are diversification and intensification. Diversification is to search better solution by searching in wide region. While, intensification is to search better solution by searching around a good solution intensively. The step size of a search point is moving from outer layer to inner layer by following logarithmic pattern. Because of that, everything will settle at a point in a time. For the most suitable controller with error criteria for flexible manipulator system is PID controller. It because of the presence of proportional gain can improve the rise time, presence of derivative gain can reduce the overshoot and presence of integral gain can reduce the steady-state error. So, moderate peak overshoot, moderate stability and lower steady state error be achieved.The investigation of the performance of SDA with parameter setting in tuning propotional-integral-derivative (PID) with error criteria for application flexible manipulator system shows that when the number of points.Np is increasing the performance of the system is getting better. It has capability of getting out of local optima points and fast convergence to close to the global optimum.In conclusion, three objective that mentioned in chapter 1 have been achieved.

REFERENCES

[1] *Spiral Optimization Algorithm Using Periodic Descent Direction*, Kenichi Tamura and Keiichiro Yasuda, SICE Journal of Control, Measurement, and System Integration, May 2016.

[2] *Bacterial Foraging and Spiral Dynamics Based Meta-heuristic Algorithm for Global Optimization with Engineering Application*, Ahmad Nor Kasruddin Nasir, University of Sheffield Mapping Street Sheffield ,United Kingdom, October 2014.

[3] *A New Multipoint Search Method*, 978-1-4799-0652-9/13, IEEE, 2013.

[4] *A Stability Analysis Based Parameter Setting Method for Spiral Optimization*, 978-1-4799-0652-9/13@ IEEE, 2013.

[5] *The Spiral Optimization Algorithm Convergence Conditions and Settings*, IEEE Transaction on System, Man, and Cybernetics: Systems 2013.

[6] *Spiral Dynamics Algorithm International Journal on Artificial Intelligence Tools* Vol.23 No 6World Scientific Publishing Company. (2014)

[7] *Study on Cluster-Structured Spiral Optimization*, IEEE Conference on Systems Man and Cybernatic, 2014.

[8] *Greedy Spiral Dynamic Algorithm with Application to Controller Design.* IEEE Conference on Systems, Process and Control (ICSPC 2016), 16-18 December 2016, Melaka, Malaysia.

[9] *A Novel Adaptive Spiral Dynamic Algorithm for Global Optimization,* IEEE 978-4799-1568, 2014.

[10] *PID-Based Control of a Single-Link Flexible Manipulator in Vertical Motion with Genetic Optimization*, University Tun Hussein Onn Malaysia (UTHM), Batu Pahat, Johor,2009.

[11] *P, PD, PI, PID Controller*, EE402-Discrete Time Control Systems, Middle East Technical University,2008.

[12] *Swarm Optimization of an Active Vibration Controller for Flexible Manipulator,* Department of System, Dynamics & Control University Technology Malaysia.

[13] *Design of PID Controller for Flexible Link Manipulator*, Vol,3 Issue ,International Journal of Engineering Research and Application Department of Electrical and Electronics Engineering, SV University, Tirupati ,A,P India, January-February,2013.

[14] *Comparison of PID Controller Tuning Methods with Genetic Algorithm for FOPTD System,* Journal of Engineering Research and Applications ISSN: 2248-9622,Vol.4, February,2014.

[15] *Tuning PID Controllers using the ITAE Criterion,* Faculty of Engineering ,University of Porto, 867-873,2005.

APPENDICES

Main Coding For SDA

```
tic
clear all
close all
clc
warning off

%/* Control Parameters of SDA algorithm*/

FoodNumber=10; %/*The number of food sources */

maxCycle=100; %/*The number of cycles for foraging {a stopping
criteria}*/
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

D=2;
rz=0.95;
I=eye(D,D);
thetaz=pi/4;
Rz=angle1(thetaz,D);

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%/* Problem specific variables*/
objfun='rosenbrock';

R1=2.048;%upper limit
R2=-2.048;%lower limit

ub=ones(1,D)*R1; %/*upper bounds of the parameters. */
lb=ones(1,D)*(R2);%/*;lower bound of the parameters.*/


runtime=1;%/*Algorithm can be run many times in order to see its
robustness*/

 ObjVal(:,:)=0*ones(1,FoodNumber);

GlobalMins=zeros(1,runtime);

for r=1:runtime

% /*All food sources are initialized */
%/*Variables are initialized in the range [lb,ub]. If each parameter has
different range, use arrays lb[j], ub[j] instead of lb and ub */

Range = repmat((ub-lb),[FoodNumber 1]);
Lower = repmat(lb, [FoodNumber 1]);
Foods = rand(FoodNumber,D) .* Range + Lower;
% ObjValmod=feval(objfun,Foods)
```

```
for i=1:FoodNumber

ObjVal(1,i)=feval(objfun,Foods(i,:));

end

% Fitness=calculateFitness(ObjVal)

%reset trial counters
trial=zeros(1,FoodNumber);

%/*The best food source is memorized*/
BestInd=find(ObjVal==min(ObjVal));
BestInd=BestInd(end);

MinJ=ObjVal(BestInd);
MinJ1=MinJ;


%centerpoint for spiral
xstar=Foods(BestInd,:);
xstarl=xstar;

iter=1   ;
GlobalMinimum(:,iter)=MinJ1;
fprintf('Ýter=%d ObjVal=%g\n',iter,MinJ1);
while ((iter <= maxCycle)),

%%%%%%%%% EMPLOYED BEE PHASE %%%%%%%%%%%%%%%%%%%%%%%%
    for i=1:(FoodNumber);


                        sol=abs(((rz*Foods(i,:)*Rz))- (((xstar*((rz*Rz)-
I)))));


        %/*if generated parameter value is out of boundaries, it is
shifted onto the boundaries*/
        ind=find(sol<lb);
        sol(ind)=lb(ind);
        ind=find(sol>ub);
        sol(ind)=ub(ind);



    Foods(i,:)=sol     ;
    end;


     ObjVal=feval(objfun,Foods);
     BestInd=find(ObjVal==min(ObjVal));
     BestInd=BestInd(end);
     MinJ=ObjVal(BestInd);
```

```
      if MinJ < MinJ1

                 MinJ1=MinJ;
                 xstar=Foods(BestInd,:);
                 xstarl=xstar;

      else

           MinJ1;
           mintest=MinJ1;
           xstarl;
      end


GlobalParameter(:,iter)=xstarl;
GlobalMinimum(:,iter)=MinJ1;

fprintf('Ýter=%d ObjVal=%g\n',iter,MinJ1);

iter=iter+1;


end % End of ABC

end; %end of runs

% xlswrite('MODABC2.xlsx',k,excelno,['D' num2str(count)]);

toc
save all
```