PID TUNING OF DC MOTOR USING SWARM ITELLIGENCE ALGORITHM

By

ARHIMNY HASDI AIMON

FINAL PROJECT REPORT

Submitted to the Department of Electrical & Electronic Engineering
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PID TUNING OF DC MOTOR USING SWARM INTELLIGENCE ALGORITHM

by

Arhimny Hasdi Aimon

A project dissertation submitted to the
Department of Electrical & Electronic Engineering
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Approved:

____

AP. Dr. Irraivan Elamvazuthi
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

September 2012

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____

Arhimny Hasdi Aimon

ABSTRACT

In this project, Particle Swarm Optimization (PSO) as one of Swarm Intelligence Algorithm based has proposed to be integrated with PID (Proportional, Integral, Derivative) Controller in order to achieve optimal tuning method. PSO-PID Controller is employed to improve the control performance of DC Motor. PSO-PID Controller has successfully eliminated the overshoot at the same time had faster response time. Inertia Weight PSO (IWPSO) and Constriction Factor PSO (CFPSO) are also employed in this project. Both of them verify to be able to improve the convergence of swarm by showing smaller value of MSE. CFPSO-PID Controller has regarded as the best tuning method since it results the best control performance and smaller value of MSE. The simulation is performed both in C++ and MATLAB. PSO-PID Controller algorithm is performed in C++. The results will be the optimal value of controller parameters. In order to evaluate the control performance, the three control parameters will be used to tune DC Motor simulated in MATLAB. PSO-PID Controller Graphical User Interface (GUI) has developed that integrates both C++ and MATLAB program for more practical use.

ACKNOWLEDGEMENT

First and foremost, the author would like to praise Allah the Almighty for His guidance and blessing that gives the author the strength to complete her Final Year Project.

Author also would like to thank AP. Dr. Irraivan Elamvazuthi for all supervision and support that he has given in terms of guidance, advice, commitment and assistance. The co-operation is much indeed appreciated.

Grateful thanks and appreciation goes to author’s parents and friends for their endlessly encouragement, support and suggestion in completion of the project.

Last but not least, endless thanks to those who helped author directly or indirectly for their guidance, sharing and support throughout the entire two semesters. Their help are highly appreciated.

TABLE OF CONTENT

List of Figures

List of Tables

List of Appendices

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Control techniques have been widely proposed to be applied in electromechanical design control. Therefore several control techniques have been researched and developed to enhance the system performance. Control techniques have kept progressing like adaptive control, fuzzy control and neural control. However PID (Proportional Integral Derivative) controller has been extensively applied in industry and more preferable due to simple algorithm, easy implementation, robust performance, and its convenience in adjusting the controller parameter [1]. However these advantages can be achieved if only the controller parameter has been properly tuned. PID tuning means finding the best gain for controller parameter, while properly tuning system will have the better dynamic performance, and guaranteed security.

Therefore, PID tuning method has developed in order to properly tune PID controller parameters. The conventional PID (CPID) tuning like Ziegler-Nichols, Cohen-coon and many other methods has employed to tune controller parameters. Ziegler-Nichols tuning method has been preferred by most of the engineers, because it is quite effective in determining the controller parameter when the system is linear. However it is not applicable when the system is non linear, and most of plants and motors have higher order, time delays and non-linearity [1], [2].

This inadequate performance of CPID has triggered the attempt to integrate Artificial Intelligence (AI) with PID Controller. Fuzzy Logic and BP Algorithm is the pioneer of AI-PID development. However they still has some shortcomings like long and unpredicted training process, low convergence speed and no general systematic procedure [3], [4]. AI-PID has continued its development by introducing GA (Genetic Algorithm) as one of evolutionary computation. Furthermore GA has been successfully implemented in PID controller for non-linear plant. However recent studies have identified some shortcomings in GA performance. The inefficiencies identified in GA emerges in application where the parameters being optimized are highly correlated, and the issue with premature convergence [1], [5], [6].

Table 1 is the summary of the development of existing tuning methods and their limitations:

Table 1: Literature review of PID Tuning Methods

Year	Author	Tuning Method	Limitations
2004	Astrom, K.J.,& Hagglund, T. [7]	Ziegler Nichols.	- The method relies on trial and error - Difficult to properly tune the non-linear system
2011	Luoren, L. & Jinling L [3].	BP Algorithm, Neural Network.	- A long and unpredictable training process - Low convergence speed
2001	Kim, Y. H. et al [4]	Fuzzy Logic Control.	- Restrictive in general application - Expert knowledge needed in input-output relation - No general systematic for the design.
2004	Gaing [1]	Genetic Algorithm.	- Enormous computational efforts
2009	Fang, H., & Chen, L. [5]	Genetic Algorithm	-Needs a complete set of object information, pre-requisite knowledge - Premature convergence
2007	Nasri, M. et al [6]	Genetic Algorithm.	- Inefficiency in application with a highly correlated parameter.

1.2 Problem Statement

The development of PID tuning method have not yet resulted the best technique, therefore the needs of increasing the capability of PID controller by adding new features is indisputable [8]. Conventional PID (CPID) controller has been employed for such a long time in motor control and relied mostly in trial and error and intuitive tuning. Trial and error is time and cost consuming, and also unguaranteed security. While intuitive tuning can only be performed by experienced engineers which has involved in PID tuning for many years, and knows the plant so well. In practical, Ziegler Nichols method relies on trial and error and difficult to properly tune for the non-linear system [1], [2], [7]. However CPID is the most preferred due to the reluctance to learn new techniques.

Over the years, several heuristic methods have been proposed to improve PID tuning method. However there have been some PID tuning methods emerged, but still none of them has been regarded as the best tuning method. Therefore Artificial Intelligence (AI) has introduced to be integrated in PID controller. AI has initiated the growth of Evolutionary Algorithm (EA) and Swarm Intelligence Algorithm (SI). GA as one of evolutionary algorithm has been successfully implemented in PID controller for higher order and non-linear system [6]. However these methods have not been able to improve the control performance into satisfactory level. Therefore Swarm Intelligence (SI) emerges as the features integrated with PID Controller. Particle Swarm Optimization (PSO) as one of swarm intelligence is expected to be able to overcome limitations found in GA.

1.3 Objective

The main objectives of this project are:

- To investigate algorithms that can improve the performance of PID tuning.
- To examine the effectiveness of the control performance of loops tuned with PSO method and its variance
- To evaluate which one of PSO variants that results the best performance of DC Motor
- To prove that PSO method can give the best control performance compare to CPID.

1.4 Scope of Study

The focus of this project is to research about the development of algorithm that can improve the performance of PID tuning starting from the CPID, AI-PID until SI-PID controller. Investigating each of algorithms and finally come up with better understanding about PSO and propose PSO-PID controller design. There will be comparative studies between the available algorithms. After recognizing the advantages and disadvantages of each of algorithms and finally finding out that the most of research papers propose PSO method as the best algorithm among them.

PSO-PID controller program will be simulated in C++ due to faster computation time compare to MATLAB. However after getting the value of the controller parameter from PSO-PID Controller program in C++, this value needs to be simulated in MATLAB to observe the control performance. The control performance set as the standard to determine that PSO has successfully improved the performance of PID controller compare to CPID.

CHAPTER 2
LITERATURE REVIEW

2.1 PID Controller

PID controller consists of three parameters (k_p, k_i, and k_d). Proportional gain, k_p functions to reduce steady state error but never to eliminate it, and also to reduce rise time. Integral gain, k_i eliminates steady state error but poor transient error, while derivative gain k_d can reduce overshoot and improve transient response [2]. The following equation is the PID controller in time domain [9]:

$$u(t) = k_p e(t) + k_i \int_0^t e(t) + k_d \frac{de(t)}{dt} \qquad (1)$$

PID controller is implemented as speed control for DC Motor. First PID Controller will track the actual speed of DC Motor, the difference of actual speed and desired speed will be the error signal, e(t). Error signal will be fed into PID controller results as control signal u(t). Control signal will compensate the error in order to achieve the desired speed.

Performance criteria is to effectively minimize the errors in time domain including overshoot (M_p), rise time (t_r), settling time (t_s) and steady state error (E_{ss}). The objective functions are to minimize overshoot, to reduce rise time and settling time, and to eliminate steady state error. Therefore these time domain criteria are the objective function of the proposed PSO-PID controller which will use to evaluate PSO-PID Controller as the fitness function.

2.2 PSO

PSO is one of optimization and evolutionary computation techniques (Kennedy and Eberhart, 1995). This optimization method is inspired from bird flocking and fish schooling. Furthermore swarm defines as population which consists of particle or individual. Particles monitor its position and velocity based on its own and others flying experience in the search space. The main purpose is to find the best position of particle, as the particle is a candidate solution to the problem. The following section points the basic parameters of PSO [1], [2], [10]:

- **Population or swarm size, n**

  It defines as the number of particles in the swarm. The optimal solution can be obtained by PSO with small swarm size of 10 to 30 particles.

- **Particle**

  Each particle which keeps adjusting its velocity and position in the search space is represented as the potential solution to the problem. The particle i is defined as $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ in d-dimensional space, while the velocity for particle i is represented as $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$.

- **Personal best solution**

  As the particles moves in the search space adjusting its position based on its own flying experience, the best position that particle has been experiencing so far is defined as personal best solution, $pbest$.

- **Global best solution**

  The best position of the particle based on its own and others flying experience in the entire population achieved so far is identified as global best solution, $gbest$.

- **Acceleration Coefficient**

  c1 represents cognitive acceleration changing the velocity of particle towards pbest while c2 represents social acceleration changing the velocity of particle toward gbest. This parameters need to be adjusted in order to get optimal solution.

2.2.1 PSO Variants

In order to improve the performance of PSO in terms of convergence swarm, PSO variants is introduced. The following type of PSO variants is going to employ in this project

- **Inertia Weight Approach**

Inertia weight is another mechanism to control the convergence of swarm proposed by Eberhart and Shi (1998). The inertia weight needs to have large value in initial stage for maximize the global exploitation, and smaller value in final stage to concentrate in local exploration [10], [11]. Random inertia weight is one of type of inertia weight proposed, as shown in the following equation

$$w = 0.5 + \frac{Rand()}{2} \tag{2}$$

Where w_{max} is the initial inertia weight, w_{min} is the final inertia weight, $iter_{max}$ is the maximum number of iterations and $iter$ is the current iteration. Mathematically, the updated velocity and position is calculated using the basic parameters defined above, as shown in the equation below [10]:

$$v_{ij}(t+1) = w.v_{ij}(t) + c_1 r_{1j}(t)\left[pbest_{ij} - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[gbest_j - x_{ij}(t)\right] \tag{3}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{4}$$

- **Constriction Factor Approach**

Furthermore constriction factor is introduced by Clerc (1999) to accomplish a better convergence compare to inertia weight. To focus on previous best position the particle fluctuation is reduced [8]. Constriction factor can be expressed in the following equations [2], [12]:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{5}$$

Where φ is defined as:

$$\varphi = c1 + c2, \quad \varphi > 4 \tag{6}$$

The following equation shows how constriction factor utilized in updating the velocity of particle:

$$v_{ij}(t+1) = \chi.\left[v_{ij}(t) + c_1 r_{1j}(t)\left[pbest_{ij} - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[gbest_j - x_{ij}(t)\right]\right] \quad (7)$$

$$i = 1, 2, \dots, n \qquad j = 1, 2, \dots, d$$

n	number of particles in a population
d	number of members in a particle
t	pointer of iteration
w	inertia weight
χ	constriction factor
$v_{ij}(t)$	velocity of particle i at iteration t $V_j min < v_{ij}(t) < V_j max$
c_1	acceleration coefficient, component of cognitive velocity
c_2	acceleration coefficient, component of social velocity
$r_{1j}(t)$,$r_{2j}(t)$	random number between o and 1
$pbest_{ij}$	pbest of particle i
$gbest_j$	gbest of population

2.3 Related Works

According to Table 2, it has shown that some heuristic algorithms have been integrated with control techniques in order to improve the performance of DC motor or AC motor. The control techniques that have been used are LQR, GA-PID, GA-BPNN-PID, PSO-PID, PSO-FLC, PSO-SMC, and PDPSO. Linear Quadratic Regulator (LQR) has compared with GA and PSO. PSO has the best control performance to improve in overshoot, steady state error and transient response. While for GA it only improves in overshoot and steady state error but it has poor transient response compare to LQR [6]. It proves that GA is no longer satisfactory in improving the control performance because of its transient poor performance. PSO is not only integrated with PID but also with other control techniques like Fuzzy Logic Controller (FLC) and Slide Mode Controller (SMC) [13], [14]. PSO-FLC is able to improve the deficiency of FLC, which is the insufficient analytical technique. However when it compares with PSO-PID, PSO-PID has robust performance than FLC-PSO [13]. Besides conventional PSO, PDPSO (Performance Dependent PSO) has also been applied in DC Motor to improve conventional PSO by implementing adaptive inertia weight. PDPSO proves to have more speed than conventional PSO [15].

PSO has been implemented not only in AC and DC motor, but also in other plants which has more complexity. Implemented in exhaust temperature control of gas turbine system, PSO has shown that the conventional performance criteria has resulted high overshoot. MPPC is introduced as performance criteria to overcome this limitation [2]. In Shell and Tube heat exchanger application, ARPSO (Attractive-Repulsive PSO) has also been introduced to improve PSO because of its convergence issue [16]. Another modification of PSO has applied in pantograph system, which is Global-Oriented PSO to improve computation efficiency of conventional PSO using a Cauchy Operator for escaping local minima [17]. Furthermore, intelligent ship autopilot which has both PID and FLC auto pilots has applied APSO (Anti-predatory PSO)and proves APSO converge faster than PSO [18].In slider-crank mechanism system integrating intelligent fuzzy rule with PSO-PID has transformed into self-tuning PID controller [19]. While for decoupling control system in Ball Mill, the forward NN-PID controller based on chaos PPSO-BP hybrid optimization has been implemented [20].

In conclusion for plants with more complicated system than DC or AC Motor, PSO-PID is no longer has robust performance, the improvement of PSO like GPSO, ARPSO, APSO, self-tuning PSO, and hybrid PSO has been introduced and it proves to has better performance than PSO.

Table 2: Related works

No	Author	Year	Publication	Title of Journal	Technique used	Application	Advantages
1.	MaerzoughiA., Selamat H., Rahmat M. F., & Rahim H.A. [2].	2012	International Journal of Physics Sciences, Vol. 7(5), pp.720-729	Optimized PID controller for the exhaust temperature control of a gas turbine system using particle swarm optimization	- PSO-PID (using Multi-purpose performance criteria (MPPC)) - CPID	Exhaust temperature control of a gas turbine system	-PSO-PID using MPPC results optimal transient response -MPPC is more reliable, consistent, and flexible.
2.	Rajasekaran S., &Kannadasan T. [16].	2012	International Journal of Computer Application 38(4):6-11	Swarm Optimization based Controller for Temperature Control of a heat Exchanger	- Attractive-Repulsive PSO (ARPSO) -PSO -GA	Shell and tube heat exchanger	- ARPSO has the best control performance compare to PSO and GA
3.	Rana M. A., Usman Z., & Shareef Z. [21].	2011	12th IEE CINTI	Automatic Control of Ball and Beam System using Particle Swarm Optimization	-PSO-PID - PD-ITAE equation - PID- FLC	The classic Ball and Beam System	-PSO is the best control performance, negligible transient - ITAE no requirement for chart like root locus and bode plot
4.	Verma H.K., & Jain C. [15].	2011	International Conference on Electrical Energy System	A Performance-Dependent PSO (PDPSO) based Optimization of PID Controller for DC	- PDPSO	Linear brushless DC Motor	- PSO highly depends on its parameters. -PDPSO has faster speed and better performance compare to PSO

				Motor			
5.	Fang H., & Chen L. [5].	2009	Chinese Control and Decision Conference	Application of an enhanced PSO algorithm to optimal tuning of PID gains	- Enhanced PSO-PID (EPSO)	Hydro power Plant	EPSO has stable convergence characteristic and good computational ability
6.	Allaoua B., Gasbaoui B., & Mebarki B. [13].	2009	Leonardo Electronic Journal of Practices and Tech	Setting up PID DC motor speed control alteration Parameter using PSO	- PSO-PID - PSO-Fuzzy	DC motor	-Fuzzy-PSO improve the dynamic performance -PID-PSO satisfactory performance and good robustness
7.	Payakkawan, et al [22].	2009	ISCIT	Dual-line PID Controller based on PSO for speed control of DC motor	- PSO-PID	DC motor	-Adaptive PSO-PID has more robust stability and efficiency compare to PSO-PID and ZN.
8.	Qiming C., Yinman C., Ruiqing G., & Yong Z. [20].	2009	International Conference on Artificial Intelligence	The forward NN- PID Controllers based on Chaos PSO-BP Hybrid Optimization Algorithms for Decoupling Control System of Ball Mill	- NN-PID controller - PSO-BP Hybrid n	Ball Mill	NN-PID control based on PSO-BP hybrid optimization can decouple the coupling object, enhance system stability and improve the dynamics performance

9.	Gaing Z. L., & Chang R. F. [17].	2009	TENCON	Optimal PID Controller for High-Speed Rail Pantograph System with Notch Filter	- Global Oriented PSO (GPSO)	Pantograph or catenary system	Improve computation efficiency of PSO using Cauchy Operator. GPSO has robust and quick search of PID parameter
10.	Jalilvand, et al [23].	2008	12th IEE International Multitopic Conference	Advance (APSO) Based PID Controller Parameters Tuning	- PSO- PID -APSO-PID -GA	Non-linear plant	-APSO reduces the rate of overshoot compared to GA -APSO running time less than PSO and GA
11.	Samanta B. [18].	2008	IEE Swarm Intelligence Symposium	Design of Intelligent Ship Autopilots using Particle Swarm Optimization	-Anti predatory PSO (APSO) - PSO - PID - FLC	Intelligent Ship Auto Pilots	- APSO converge faster than PSO-FLC - PID- PSO faster response, PID-APSO improve the overshoot - FLC, PSO & APSO has same rise & settling time.
12.	Jain T., & Nigam M. J. [24].	2008	Journal of Theoretical and Applied Information Technology	Optimization of PD-PI controller using Swarm Intelligence	-GA -BG (Bacterial Foreaging) - PSO -PD-PI	Inverted Pendulum	- PSO is more effective than GA and BG - The time computation of PSO is less than GA and BG - BG gives better performance than GA

13.	Nasri M., Nezambadi-pour H.,&Maghfoori M. [6].	2007	World Academy Sci. Eng. & Tech 26	A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor	- PSO - GA - Linear Quadratic Regulator (LQR)	Brushless DC-Motor	-PSO has the best dynamic performance (Mp, Tr, Ts, and Ess) compare to GA and LQR -GA has better performance compare to LQR
14.	Xu-zhou L., Fei Y., & You-bo W. [25].	2007	International Conference Computational Intelligence and Security	PSO Algorithm based Online Self-Tuning of PID Controller	PSO	Online Self-Tuning	-Choosing the suitable fitness function, initial range of each particle closed to optimum solution
15.	Kao C. C., Chuang C.W., & Fung R.F. [19].	2006	Mechatronics 16:513-522	The self tuning PID control in a slider – crank mechanism system by applying PSO approach	- Fuzzy self tuning PSO-PID controller -RGA (real-coded GA)	Slider crank mechanism system	- Fuzzy role computation time is shorter than RGA - Fuzzy self tuning PSO-PID controller more robust stability and quickly solve the searching and tuning problem than RGA.
16.	Yu K.W., & Hu S. C. [14].	2006	IEE conference on Systems, Man, and Cybernetic	An Application of AC servo motor by using PSO based sliding mode controller (SMC)	PSO based SMC	AC servo motor	- PSO based SMC The trajectory of the motor and desired input are almost identical

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Project Activities

This project work flow basically consists of literature review, simulation, result, validation and final result which will be specifically explained by Figure1:

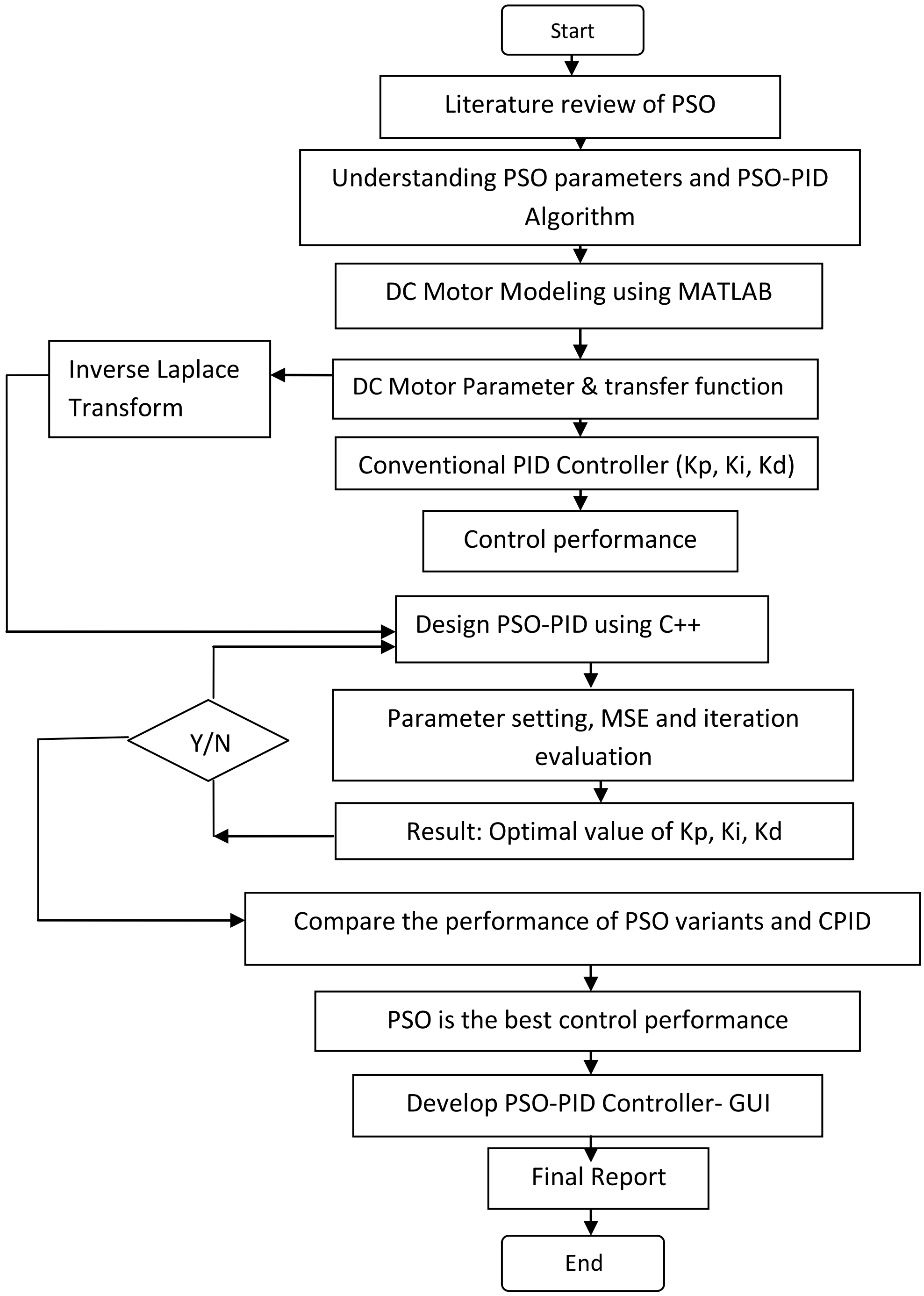


Figure 1: Flowchart of Project Process Flow

The project work starts with literature review comparing the related work regarding algorithm that have been integrated in PID controller. This work continues with investigating each of algorithms like Fuzzy system, ANN, GA, and PSO and how it helps to improve the performance of application like DC motor or other plants. Since PSO algorithm has chosen, based on its advantages from comparative study, now the project work continues with understanding the basic parameters of PSO algorithm. The most challenging step in literature review is to understand how to integrate PSO with PID controller. The gantt chart of this project is also attached in APPENDIX VI and VII as reference for the schedule for FYP1 and FYPII.

The next phase in the project work is to do the simulation of PSO. PSO-PID Controller is simulated in C++. The work starts with modeling DC Motor by deriving its transfer function and evaluate the control performance. The transfer function is converted into time domain using Inverse Laplace Transform. The code is developing in C++ compiler and input the time domain equation of Y_Model into coding. Compile and Run the program. After getting the value of Kp, Ki and Kd then simulate it using MATLAB in order to determine the control performance.

3.2 PSO-PID Controller Algorithm

To utilize the PSO in PID tuning, the population and the particles are required to be clearly defined. The population is set of n particles, and particle is the potential solution in d-dimensional search space. Furthermore the potential solution in PID tuning is the optimal value of controller parameters which consists of three members(k_p, k_i, k_d). Figure 2 shows the block diagram of the implemented PSO-PID Controller:

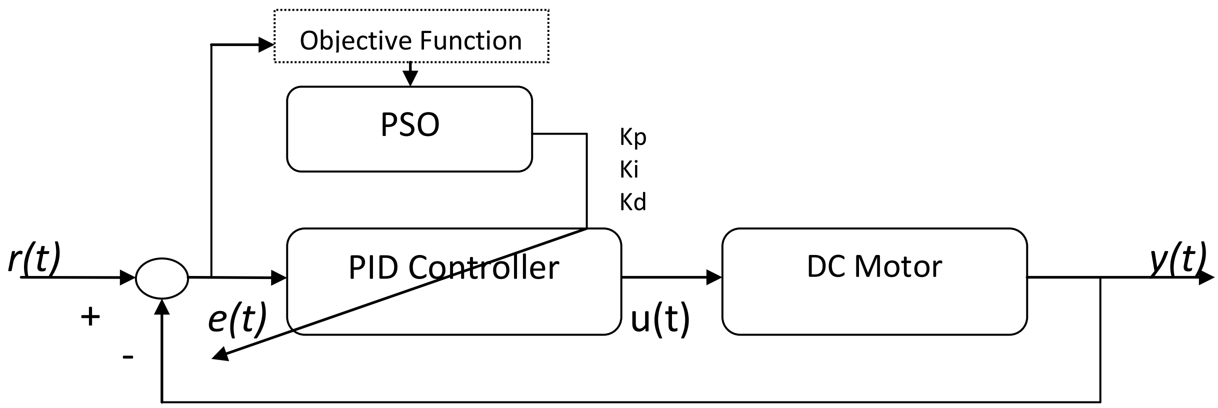


Figure 2: Block Diagram of PSO-PID Controller [22].

The following is the procedures for utilizing the proposed PSO-PID controller in DC Motor:

Step 1: Model DC Motor from the datasheet of chosen motor, get the physical parameter and convert it into SI units. The physical parameter is used to derive open-loop transfer function.

Step 2: After deriving transfer function then simulate in MATLAB both for open loop and closed loop response. After observing the control performance, PID Controller is required to be implemented in the system.

Step 3: Transfer function is converted into time domain using Inverse Laplace Transform. Since the program is developed in C++ compiler (Dev C++), edit Y_Model according to equation that has been calculated in time domain

Step 4: Compile the program and run it from the executable file. Vary the parameter: no of particle, acceleration coefficient (c1and c2), and random number (r1 and r2).

Step 5: The executable file will perform some iteration and get the best value of Kp, Ki, and Kd. Copy and paste the final value of Kp, Ki, and Kd and also Y_model, Y_control, MSE and iteration from the data display in executable file.

Step 6: To copy the data, right click in executable file>> select>>drag the data>> Enter. Next paste the data in Microsoft Excel for data representation and analysis. After pasting the data into excel, the data can be displayed in the trend of Y_Model and Y_Control over the time and see the effect of varying PSO parameters.

Step 7: To determine other control performance parameters like rise time, settling time, overshoot and steady state error, three controller parameters is simulated in MATLAB.

Figure 3 is the flowchart from the PSO-PID algorithm which has been developed in C++.

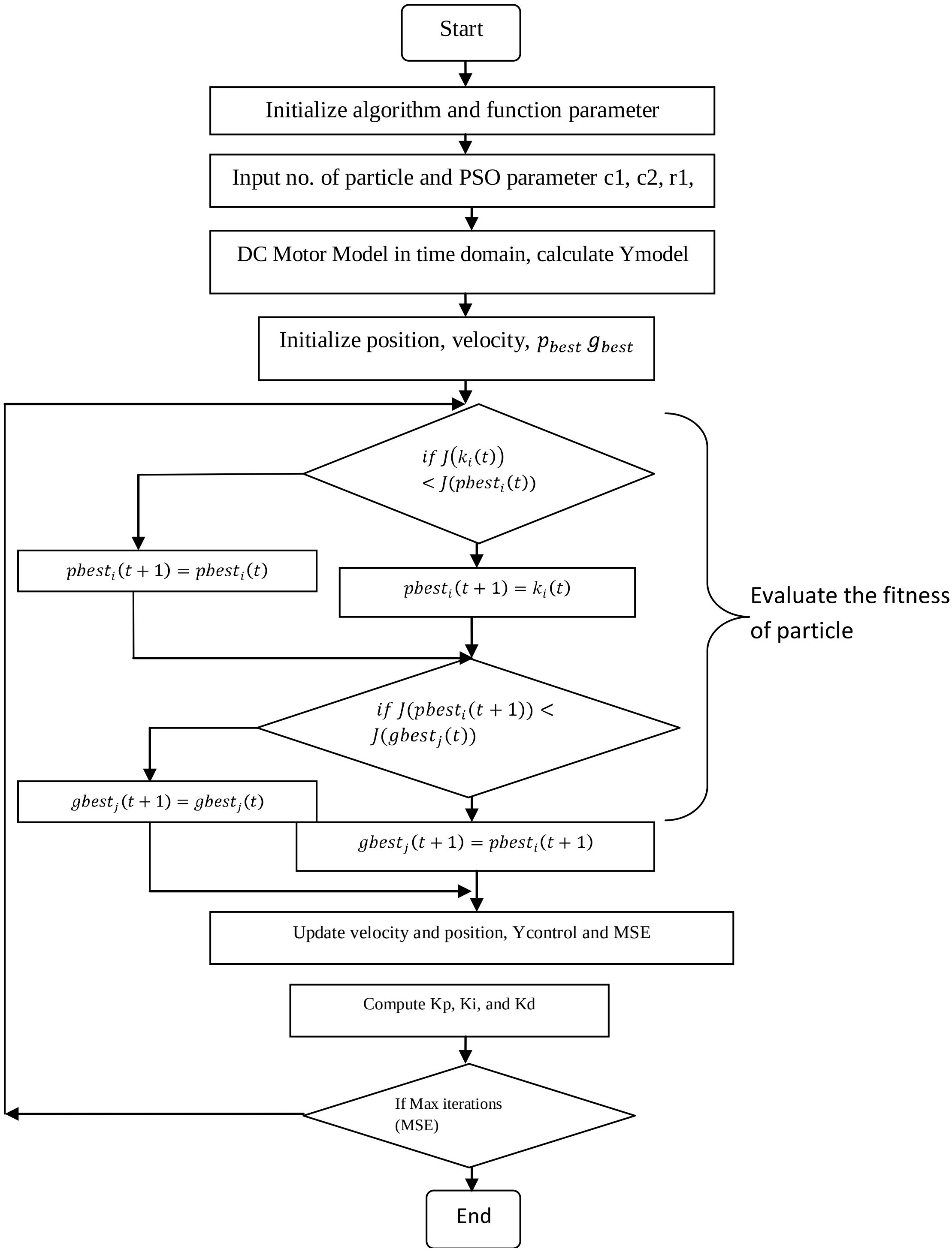


Figure 3: Flowchart of PSO-PID Algorithm

3.3 PSO Parameters Initialization

PSO algorithm is highly dependence on its parameters, therefore parameters value need to be initialized. Table 3 shows PSO parameters that is initialized based on literature review has been conducted. For random number which varies from 0 to 1, the simulation will be conducted by keeping r1 constant and varying r2 to observe its effect to performance of PSO.

Table 3: PSO Parameters Initialization

PSO Parameters	Value
No of Particles	20
Cognitive Acceleration (c1)	1.0
Social Acceleration (c2)	1.2

3.4 Statistical Evaluation

Mean Squared Error (MSE) is utilized to evaluate the dynamical behavior and convergence characteristic [1], [8]. Moreover MSE indicates the convergence of swarm. The smallest value of MSE will be desirable and indicating the control of convergence of swarm. Therefore MSE is significant in this project in order to find the optimal solution. The formula for MSE is shown in the following equation [1], [8]:

$$mean(\bar{x}) = \frac{\sum_{i=1}^{p} w_i}{p} \quad (8)$$

$$MSE = \frac{1}{p}\sum_{i=1}^{p}(w_i - \bar{x})^2 \quad (9)$$

Where w_i is particle position, p population size, MSE is mean squared error, and $\bar{x}$ is mean value.

3.5 PSO-PID Controller-Graphical User Interface (GUI)

PSO-PID Controller GUI is developed in this project. The GUI is built in GUIDE MATLAB. It is integrated both C++ and MATLAB program for practical use. PSO-PID Algorithm will be performed in C++, in order to get the system response the controller parameter is simulated in DC Motor model developed in MATLAB. Figure 4-8 shows the details of PSO-PID Controller GUI simulation procedures:

- Click Plot in DC Motor Closed Loop Response, It will show the closed-loop response of DC Motor

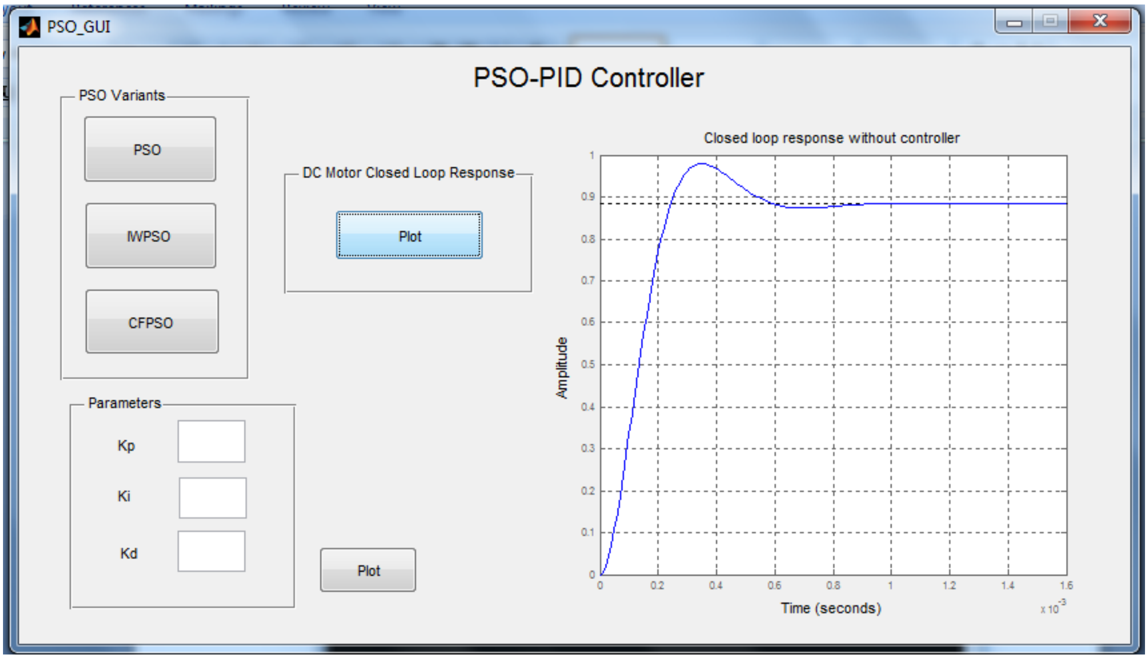


Figure 4: PSO-PID Controller GUI

- Click PSO, then PSO.exe file will appear.

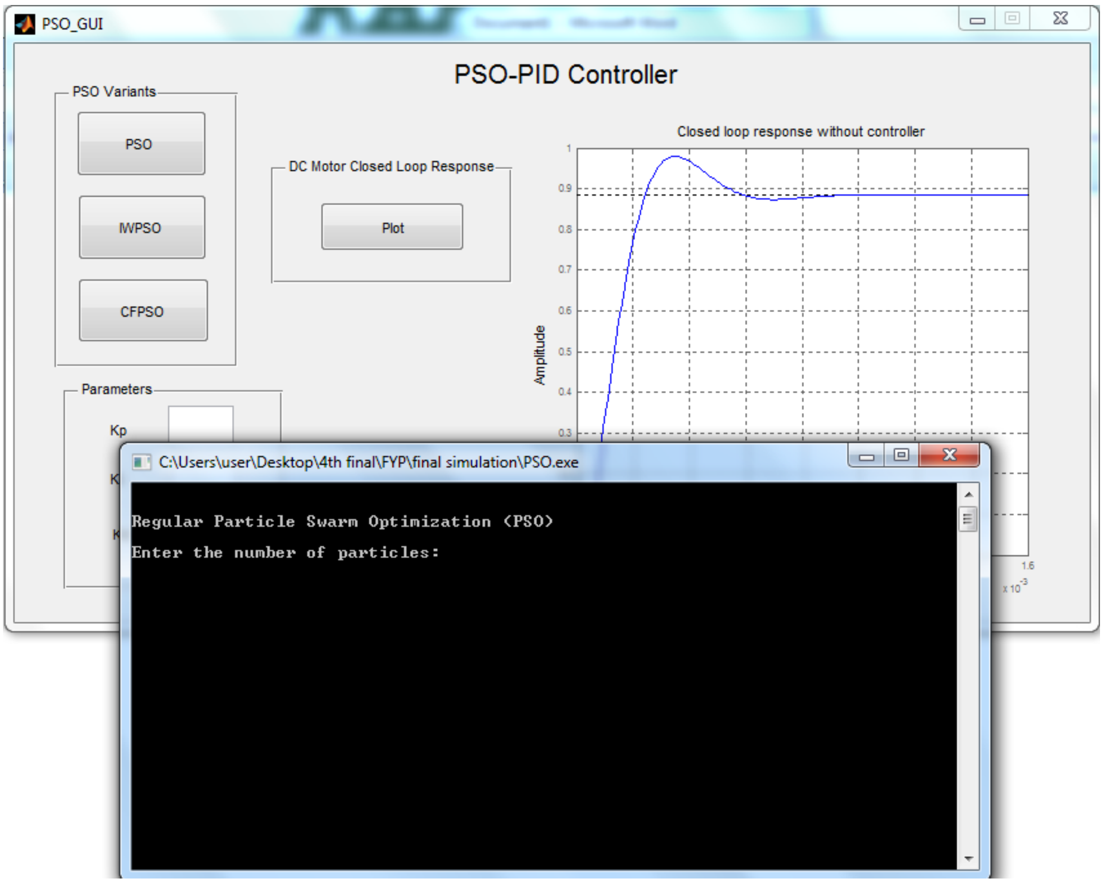


Figure 5: PSO-PID Controller GUI

- Key-in PSO parameters and Press Enter.

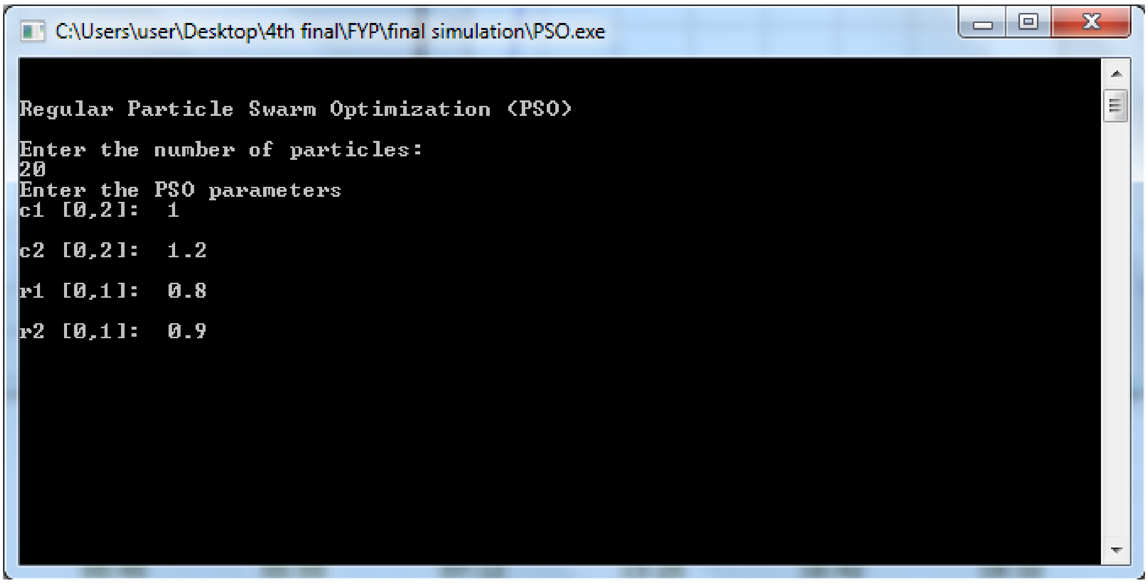


Figure 6: PSO-PID Controller GUI

- The iteration will start and stop until MSE<1.

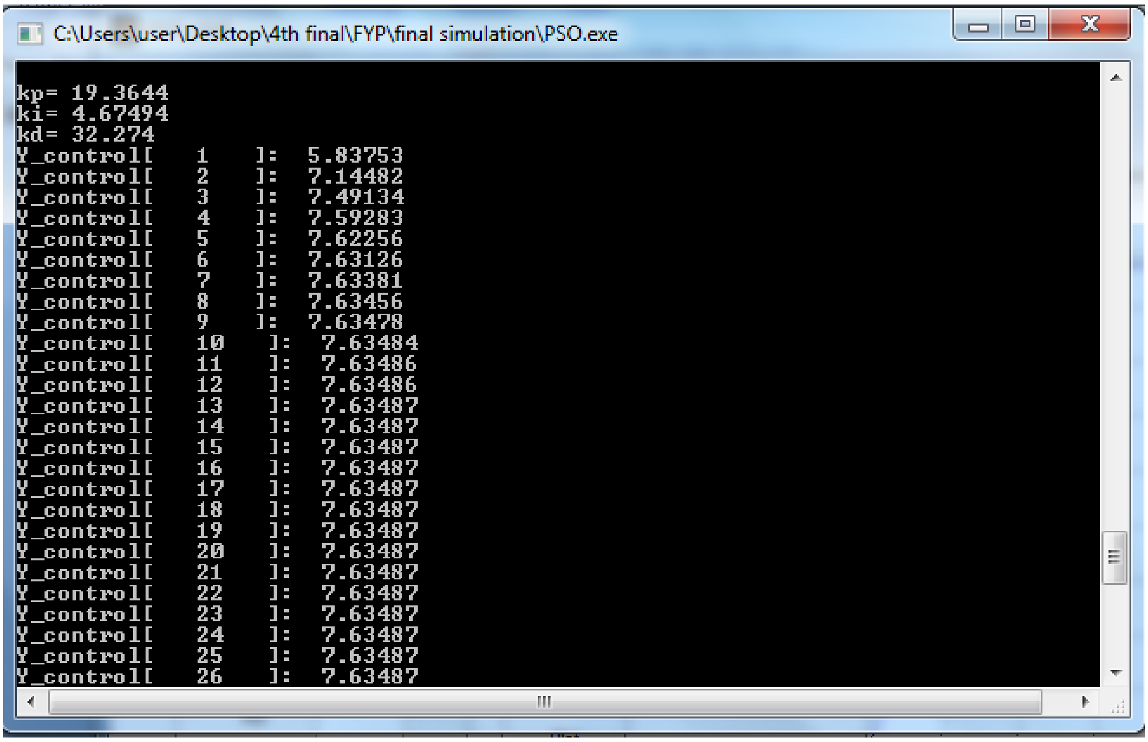


Figure 7: PSO-PID Controller GUI

- Key in the value of Kp, Ki, and Kd from value shown in PSO.exe. Then Click Plot

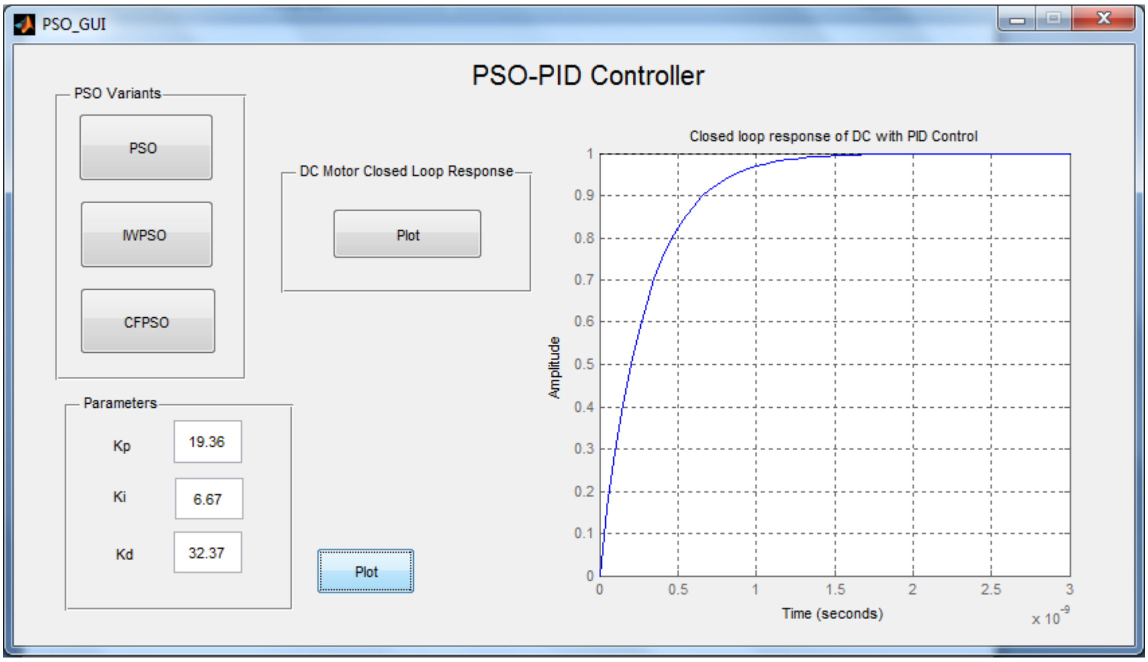


Figure 8: PSO-PID Controller GUI

CHAPTER 4
RESULTS & DISCUSSION

4.1 DC Motor Modeling

Table 4 shows the physical parameter of DC Motor (EC-4pole 22 Ø 22mm 332279) which is already converted into SI unit. DC Motor datasheet is attached in APPENDIX I.

Table 4: DC Motor Physical Parameter

Parameters	Values	Unit
Armature Circuit Resistance (R_a)	13.5	Ω (Ohm)
Armature Circuit Inductance (L_a)	$1.11x10^{-3}$	H (Henry)
Moment of Inertia (J)	$5.54x10^{-7}$	Kg m²
Coefficient of friction (f_v)	$3.215x10^{-4}$	N m s/rad
Torque constant (K_t)	0.066	N m/A
Back-Emf constant (K_b)	0.066	Vs/rad

Transfer function can be constructed using the equation which open loop response DC Motor:

$$G(s) = \frac{W(s)}{V_a(s)} = \frac{k_t}{JL_a s^2 + (JR_a + L_a f)s + R_a f_v + k_t k_b} \tag{10}$$

$$G(s) = \frac{W(s)}{V_a(s)} = \frac{0.066}{6.149x10^{-10}s^2 + 7.836x10^{-6}s + 0.008969} \tag{11}$$

$$Y(s) = \frac{107{,}334{,}525.9\, V_a(s)}{s^2 + 12{,}743.536\, s + 14{,}142{,}136.93} \tag{12}$$

Since $V_a(s)$ is unit step respose therefore $V_a(s) = \frac{1}{s}$, then substitute into transfer function equation:

$$Y(s) = \frac{107{,}334{,}525.9}{s(s^2 + 12{,}743.536\, s + 14{,}142{,}136.93)} \tag{13}$$

From the transfer function, the open loop response of DC Motor is obtained as shown in the following figure:

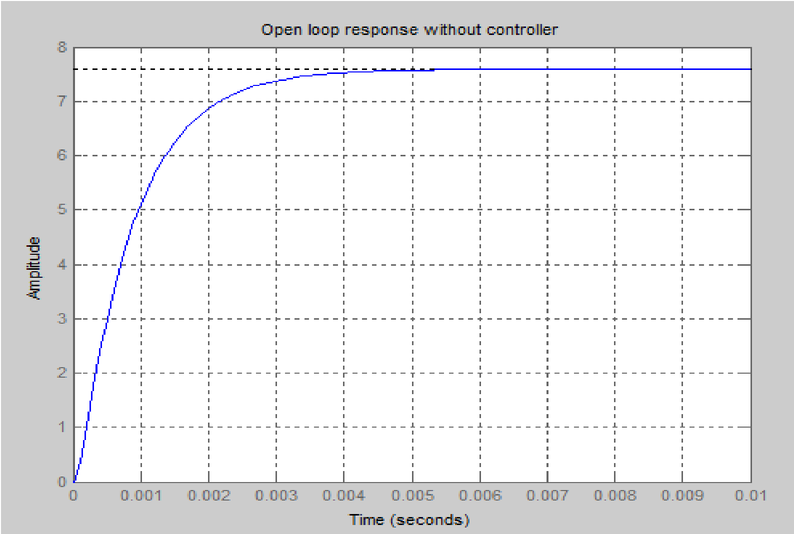


Figure 9: Open loop response of DC Motor

In order to be implemented in C++, transfer function requires to be converted into time domain using inverse Laplace transform. First find the squared root of transfer function, perform partial-fraction and convert it into time domain using Inverse Laplace Transform.

$$x_{1,2} = \frac{-12{,}743.536 \pm \sqrt{12{,}743.536^2 + \ 4x1x14{,}142{,}136.93}}{2x1} \tag{14}$$

$$x_1 = -1{,}228.12 \qquad x_2 = -11{,}515.45$$

$$Y(s) = \frac{107{,}334{,}525.9}{s\ (s + 1{,}228.12)(s + 11{,}515.45)}$$

$$= \frac{A}{s} + \frac{B}{s + 1{,}228.12} + \frac{C}{s + 11{,}515.45} \tag{15}$$

After obtaining squared roots of transfer function, now partial fraction is performed:

$$A = \frac{107{,}334{,}525.9}{(s + 1{,}228.12)(s + 11{,}515.45)}\Big|_{s\to 0} = 7.59 \tag{16}$$

$$B = \frac{107{,}334{,}525.9}{s(s + 11{,}515.45)}\Big|_{s\to 1228.12} = -8.49$$

$$C = \frac{107{,}334{,}525.9}{s(s + 1{,}228.12)}\Big|_{s\to -11{,}515.45} = 0.906$$

Hence,

$$Y(s) = \frac{7.59}{s} - \frac{8.49}{s + 1{,}228.12} + \frac{0.906}{s + 11{,}515.45} \quad (17)$$

Therefore $y(t)$ is the sum of the inverse Laplace transforms of each term.

$$y(t) = 7.59 - 8.49e^{-1{,}228.12t} + 0.906e^{-11{,}515.45t} \quad (18)$$

4.2 Conventional PID Tuning Method (CPID)

PID Controller is utilized when the system has a poor control performance of closed loop system. Figure 10 shows the closed loop system performance which has unacceptable control performance.

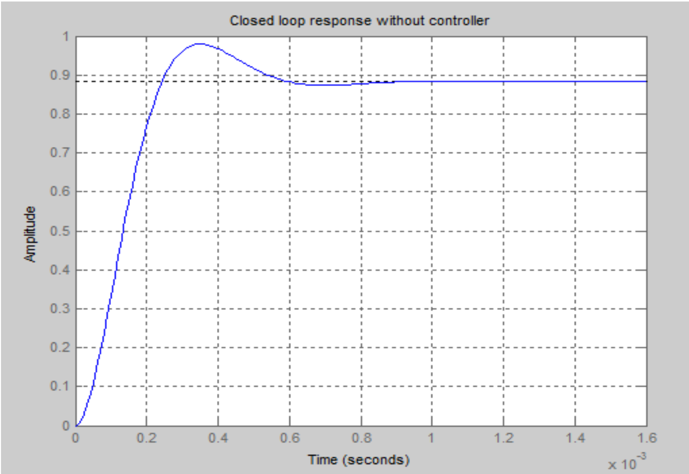


Figure 10: Closed loop response without controller

There are several conventional PID tuning methods available, open loop ZN tuning method is one of the convenient methods to employ for this system. Table 5 is the calculation to get the control parameter Kp, Ki, and Kd using Open Loop ZN Tuning Method.

Table 5: Control Parameter for Open Loop ZN Method

K=1.15	Kp	Ki	Kd
$\propto = 0.01\ sec$ $\tau = 0.01092\ sec$ K= 7.59	$Kp = 1.2\frac{\tau}{\alpha K}$ $= 0.1726$	$Ti = 2\propto = 0.02$ $Ki = \frac{Kp}{Ti} = 8.63$	$Td = 0.5\propto = 0.005$ $Ki = KpTd$ $= 8.63x10^{-4}$

Figure 11 shows the comparison of control performance between ZN PID Tuning methods and Auto Tuning.

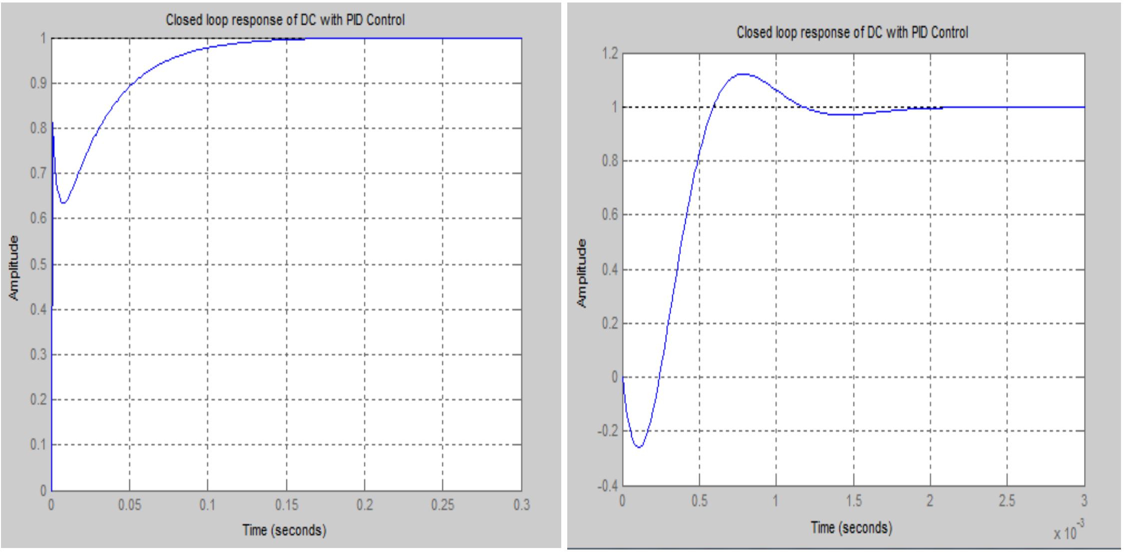


Figure 11: a). Closed-loop response using Open Loop ZN Tuning Method b) Closed-loop response after auto tuning.

4.3 PSO-PID Controller

PSO-PID controller simulation is performed in C++. This simulation is performed to see the effect of varying PSO parameters such as no. of particle, random number (r1 and r2) and acceleration coefficient (c1 and c2).

In order to see the effect of random number (r1 and r2), r1 needs to be kept constant and by varying the value of r2. Table 6 and Figure 12 are the simulation result of PSO-PID Controller referred to APPENDIX II:

Table 6: PSO r1 constant varying r2

Parameters: Particles:20, r1=0.5, C1= 1.0 & C2=1.2	kp= 18.6699	kp= 19.4341	kp= 19.8181	kp= 20.5575	**kp= 21.2447**
	ki= 4.54712	ki= 4.6755	ki= 4.73232	ki= 4.86711	**ki= 5.00271**
	kd= 31.1164	kd= 32.3902	kd= 33.0302	kd= 34.2624	**kd= 35.4079**
	r1=0.5 r2=0.5	r1=0.5 r2=0.6	r1=0.5 r2=0.7	r1=0.5 r2=0.8	**r1=0.5 r2=0.9**
Steady State Value	7.50128	7.50798	7.50873	7.5137	**7.60414**
MSE	MSE: 0.843102	MSE: 0.825143	MSE: 0.839097	MSE: 0.841533	**MSE: 0.709952**
Iteration	iter: 94	iter: 92	iter: 90	iter: 88	**iter: 86**
Parameters: Particles:20,	kp= 17.903	kp= 18.5893	**kp= 19.3663**	kp= 19.5863	kp= 20.4463
	ki= 4.42472	ki= 4.53114	**ki= 4.67036**	ki= 4.69	ki= 4.85839

r1=0.6, C1= 1.0 & C2=1.2	kd= 29.8384	kd= 30.9822	**kd= 32.2772**	kd= 32.6438	kd= 34.0772
	r1=0.6 r2=0.5	r1=0.6 r2=0.6	**r1=0.6 r2=0.7**	r1=0.6 r2=0.8	r1=0.6 r2=0.9
Steady State Value	7.62539	7.54426	**7.58662**	7.51391	7.59858
MSE	MSE: 0.603098	MSE: 0.568158	**MSE: 0.555211**	MSE: 0.787859	MSE: 0.642699
Iteration	iter: 99	iter: 95	**iter: 93**	iter: 91	iter: 89
Parameters: Particles:20, r1=0.7, C1= 1.0 & C2=1.2	kp= 17.0737	kp= 17.8323	**kp= 18.7131**	kp= 19.0333	kp= 19.9586
	ki= 4.282	ki= 4.40319	**ki= 4.55643**	ki= 4.60383	ki= 4.77803
	kd= 28.4562	kd= 29.7204	**kd= 31.1886**	kd= 31.7222	kd= 33.2644
	r1=0.7 r2=0.5	r1=0.7 r2=0.6	**r1=0.7 r2=0.7**	r1=0.7 r2=0.8	r1=0.7 r2=0.9
Steady State Value	7.5655	7.4879	**7.57143**	7.53765	7.62676
MSE	MSE: 0.435939	MSE: 0.932614	**MSE: 0.514451**	MSE: 0.618922	MSE: 0.733253
Iteration	iter: 101	iter: 97	**iter: 95**	iter: 93	iter: 91
Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2	kp= 16.6333	**kp= 17.5261**	kp= 18.1706	kp= 18.6892	**kp= 19.2556**
	ki= 4.22423	**ki= 4.36644**	ki= 4.46928	ki= 4.55633	**ki= 4.65022**
	kd= 27.7222	**kd= 29.2102**	kd= 30.2844	kd= 31.1486	**kd= 32.0926**
	r1=0.8 r2=0.5	**r1=0.8 r2=0.6**	r1=0.8 r2=0.7	r1=0.8 r2=0.8	**r1=0.8 r2=0.9**
Steady State Value	7.54849	**7.58953**	7.5958	7.57506	**7.59557**
MSE	MSE: 0.456883	**MSE: 0.457884**	MSE: 0.503451	MSE: 0.511611	**MSE: 0.564082**
Iteration	iter: 103	**iter: 100**	iter: 97	iter: 95	**iter: 93**
Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2	kp= 16.1508	kp= 17.0977	kp= 17.8242	kp= 18.3953	**kp= 18.7231**
	ki= 4.14349	ki= 4.2973	ki= 4.42063	ki= 4.51689	**ki= 4.55639**
	kd= 26.918	kd= 28.4961	kd= 29.7069	kd= 30.6589	**kd= 31.2052**
	r1=0.9 r2=0.5	r1=0.9 r2=0.6	r1=0.9 r2=0.7	r1=0.9 r2=0.8	**r1=0.9 r2=0.9**
Steady State Value	7.51129	7.53429	7.59336	7.57036	**7.48719**
MSE	MSE: 0.646069	MSE: 0.539189	MSE: 0.479829	MSE: 0.4981	**MSE: 0.981144**
Iteration	iter: 105	iter: 101	iter: 98	iter: 96	**iter: 94**

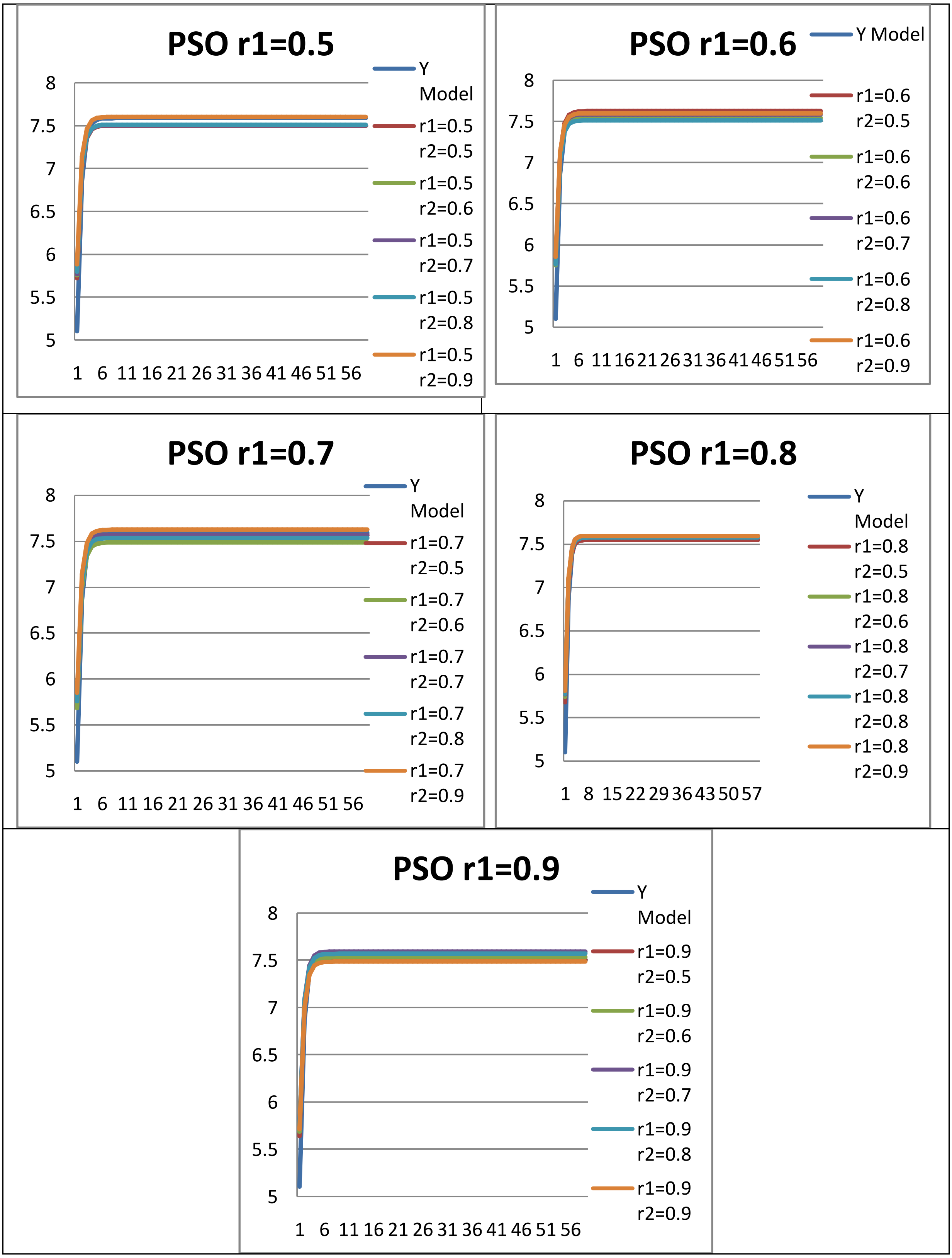


Figure 12: a) PSO-PID r1=0.5 b) PSO-PID r1=0.6 c) PSO-PID r1=0.7 d) PSO-PID r1=0.8 e) PSO-PID r1=0.9

4.3.1 Effect of random numbers (r1&r2) in PSO

According to the results shown in Table 6, it can be observed that increasing r2 will reduce the number of iteration. It will update the particle position to a better solution. While increasing the value of r1 will increase the number of iteration. Increasing r1 will result the smaller value of Kp, Ki, and Kd, however increasing r2 will result the higher value of Kp, Ki, and Kd. Increasing the value of r2 has more influence in determining the better solution than varying r1. In conclusion, r2 requires to be set higher than r1 in order to get optimal solution.

Form the result shown in Table 6, the variation of r1 and r2 which gives smallest MSE is tabulated in Table 7:

Table 7: PSO-PID with the smallest MSE

r1=0.5 r2=0.9	r1=0.6 r2=0.9	r1=0.7 r2=0.7	r1=0.8 r2=0.6	**r1=0.8 r2=0.9**	r1=0.9 r2=0.7
kp= 21.2447	kp= 20.4463	kp= 18.7131	kp= 17.5261	**kp= 19.2556**	kp= 17.8242
ki= 5.00271	ki= 4.85839	ki= 4.55643	ki= 4.36644	**ki= 4.65022**	ki= 4.42063
kd= 35.4079	kd= 34.0772	kd= 31.1886	kd= 29.2102	**kd= 32.0926**	kd= 29.7069
7.60414	7.59858	7.57143	7.58953	**7.59557**	7.59336
MSE: 0.709952	MSE: 0.642699	MSE: 0.514451	MSE: 0.457884	**MSE: 0.564082**	MSE: 0.479829
iter: 86	iter: 89	iter: 95	iter: 100	**iter: 93**	iter: 98

Furthermore the trend in Figure 12 shows that when r1=0.8 the result is the most converging by showing the lowest MSE which is 0.457884 is located when r1=0.8 and r2=0.6. It means that the best solution of particle exists within r1=0.8.

In order to find the best control performance, the value of Kp, Ki, and Kd from each solution of particle requires to be simulated in MATLAB.

Figure 13 is the comparison of system response of PSO-PID when r1=0.8 r2= 0.6 and r2=0.9 simulated in MATLAB:

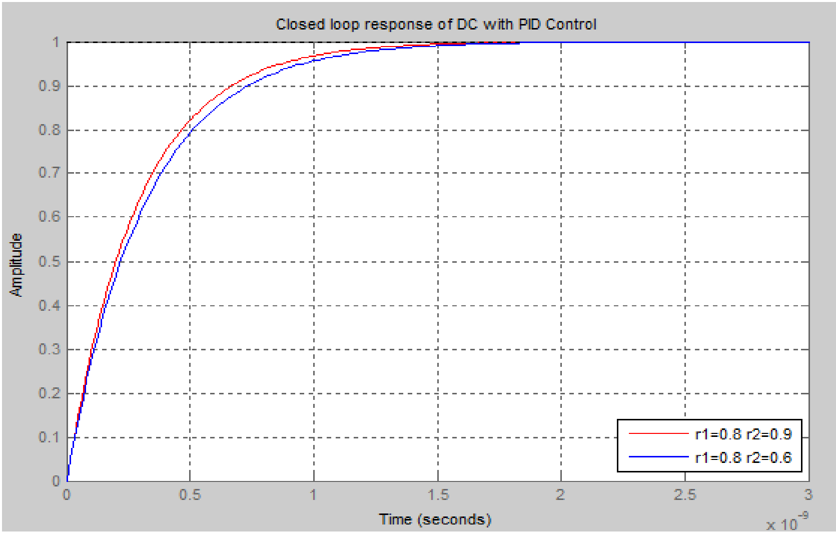


Figure 13: Comparison between PSO-PID r1=0.8 r2=0.9 and r1=0.8 r2=0.6.

According to Figure 13, it can be concluded that the best control performance is located when r1=0.8 r2=0.9, since it give the faster response with shorter rise time and settling time compare to r1=0.8 r2=0.6.

4.4 Inertia Weight PSO-PID Controller (IWPSO-PID)

IWPSO is one of variants from PSO which is used to control the convergence of swarms. The same procedure as PSO-PID needs to be carried out to simulate IWPSO-PSO by keeping r1 constant and varying r2. Table 8 and Figure 14 are the simulation result of IWPSO-PID referred to APPENDIX III:

Table 8: IWPSO r1 constant varying r2

Parameters: Particles:20, r1=0.5, C1= 1.0 & C2=1.2	kp= 14.8245	kp= 15.2105	**kp= 15.8771**	kp= 15.046	kp= 15.2102
	ki= 4.13863	ki= 4.18902	**ki= 4.31114**	ki= 4.09031	ki= 4.10644
	kd= 24.7074	kd= 25.3509	**kd= 26.4619**	kd= 25.0767	kd= 25.3504
	r1=0.5 r2=0.5	r1=0.5 r2=0.6	**r1=0.5 r2=0.7**	r1=0.5 r2=0.8	r1=0.5 r2=0.9
Steady State Value	7.53663	7.56616	**7.58434**	7.52697	7.5602
MSE	MSE: 0.429182	MSE: 0.348836	**MSE: 0.370847**	MSE: 0.491678	MSE: 0.360028

Iteration	iter: 106	iter: 105	**iter: 102**	iter: 107	iter: 107
Parameters: Particles:20, r1=0.6, C1= 1.0 & C2=1.2	kp= 14.334	kp= 14.9154	kp= 16.643	**kp= 17.2648**	kp= 16.496
	ki= 4.06826	ki= 4.1574	ki= 4.52408	**ki= 4.64127**	ki= 4.44165
	kd= 23.8899	kd= 24.8591	kd= 27.7383	**kd= 28.7746**	kd= 27.4933
	r1=0.6 r2=0.5	r1=0.6 r2=0.6	r1=0.6 r2=0.7	**r1=0.6 r2=0.8**	r1=0.6 r2=0.9
Steady State Value	7.54386	7.51379	7.60148	**7.57712**	7.6108
MSE	MSE: 0.376577	MSE: 0.57722	MSE: 0.436764	**MSE: 0.436466**	MSE: 0.459436
Iteration	iter: 108	iter: 105	iter: 97	**iter: 94**	iter: 99
Parameters: Particles:20, r1=0.7, C1= 1.0 & C2=1.2	kp= 14.1495	kp= 15.9144	kp= 14.1422	kp= 14.5629	**kp= 14.9561**
	ki= 4.05974	ki= 4.4174	ki= 3.98416	ki= 4.04944	**ki= 4.11674**
	kd= 23.5825	kd= 26.524	kd= 23.5703	kd= 24.2714	**kd= 24.9269**
	r1=0.7 r2=0.5	r1=0.7 r2=0.6	r1=0.7 r2=0.7	r1=0.7 r2=0.8	**r1=0.7 r2=0.9**
Steady State Value	7.53352	7.56547	7.52857	7.52607	**7.58925**
MSE	MSE: 0.41923	MSE: 0.381223	MSE: 0.447195	MSE: 0.478399	**MSE: 0.33304**
Iteration	iter: 108	iter: 99	iter: 110	iter: 108	**iter: 107**
Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2	kp= 13.4004	kp= 14.5066	kp= 14.9156	kp= 15.2658	**kp= 16.0156**
	ki= 3.92482	ki= 4.12778	ki= 4.18909	ki= 4.24113	**ki= 4.38631**
	kd= 22.334	kd= 24.1777	kd= 24.8594	kd= 25.443	**kd= 26.6927**
	r1=0.8 r2=0.5	r1=0.8 r2=0.6	r1=0.8 r2=0.7	r1=0.8 r2=0.8	**r1=0.8 r2=0.9**
Steady State Value	7.53846	7.53755	7.5863	7.61682	**7.59798**
MSE	MSE: 0.366252	MSE: 0.412106	MSE: 0.328412	MSE: 0.423209	**MSE: 0.397061**
Iteration	iter: 112	iter: 106	iter: 105	iter: 104	**iter: 100**
Parameters: Particles:20, r1=0.9, C1= 1.0 & C2=1.2	**kp= 13.1278**	**kp= 13.78**	**kp= 13.8082**	**kp= 14.2004**	**kp= 14.6556**
	ki= 3.89635	**ki= 3.99401**	**ki= 3.96755**	**ki= 4.02798**	**ki= 4.10473**
	kd= 21.8797	**kd= 22.9666**	**kd= 23.0137**	**kd= 23.6673**	**kd= 24.4261**
	r1=0.9 r2=0.5	**r1=0.9 r2=0.6**	**r1=0.9 r2=0.7**	**r1=0.9 r2=0.8**	**r1=0.9 r2=0.9**
Steady State Value	7.61174	7.55197	7.56727	7.62208	7.63981
MSE	MSE: 0.308087	MSE: 0.324831	MSE: 0.289067	MSE: 0.398137	MSE: 0.524638
Iteration	iter: 114	iter: 110	iter: 111	iter: 110	iter: 108

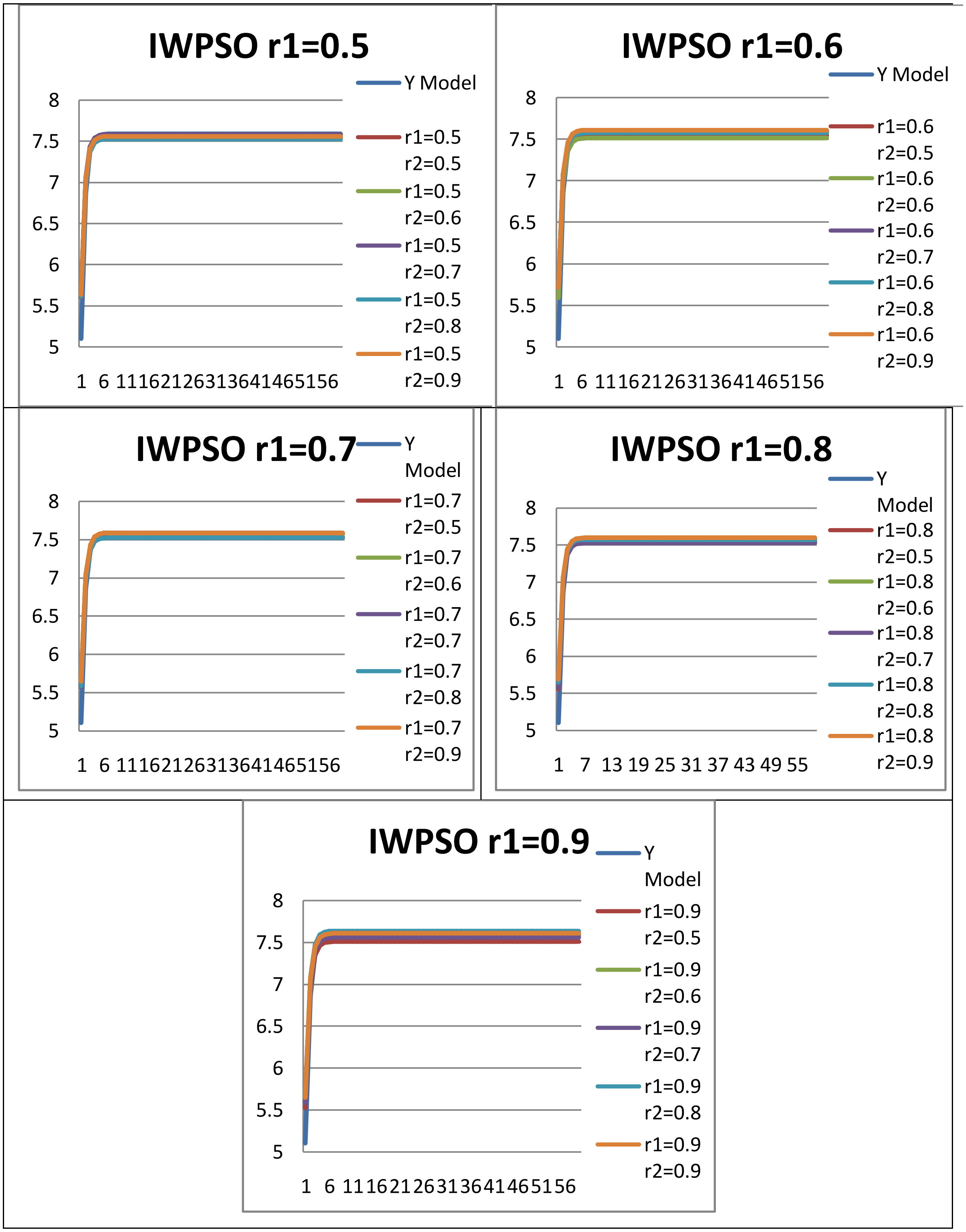


Figure 14: a) IWPSO-PID r1=0.5 b) IWPSO-PID r1=0.6 c) IWPSO-PID r1=0.7 d) IWPSO-PID r1=0.8 e) IWPSO-PID r1=0.9

4.4.1 Effect of r1 constant and varying r2 in IWPSO

According to Table 8, it proves that IWPSO has smaller error compare to PSO. Based on the trend in Figure 14, it shows that the results from r1=0.5 until r1=0.8 are converged. It is also verified by smaller value of MSE, since MSE indicates the dynamic behavior of particles. It proves that IWPSO is used to control the convergence of swarms. Furthermore the value of r1 and r2 does not influence the no. of iteration like PSO. Table 9 shows the IWPSO result with the smallest MSE value.

Table 9: IWPSO-PID with the smallest MSE

kp= 15.8771	kp= 17.2648	kp= 14.9561	kp= 16.0156	kp= 14.6556
ki= 4.31114	ki= 4.64127	ki= 4.11674	ki= 4.38631	ki= 4.10473
kd= 26.4619	kd= 28.7746	kd= 24.9269	kd= 26.6927	kd= 24.4261
r1=0.5 r2=0.7	r1=0.6 r2=0.8	r1=0.7 r2=0.9	r1=0.8 r2=0.9	r1=0.9 r2=0.9
7.58434	7.57712	7.58925	7.59798	7.63981
MSE: 0.370847	MSE: 0.436466	MSE: 0.33304	MSE: 0.397061	MSE: 0.524638
iter: 102	iter: 94	iter: 107	iter: 100	iter: 108

According to Table 9, it indicates that r2 has to be higher than r1 in order to give best solution. Each value of Kp, Ki and Kd from Table 9 requires to be simulated in MATLAB. Figure 15 is the comparison of system response of IWPSO-PID between the best solutions simulated in MATLAB:

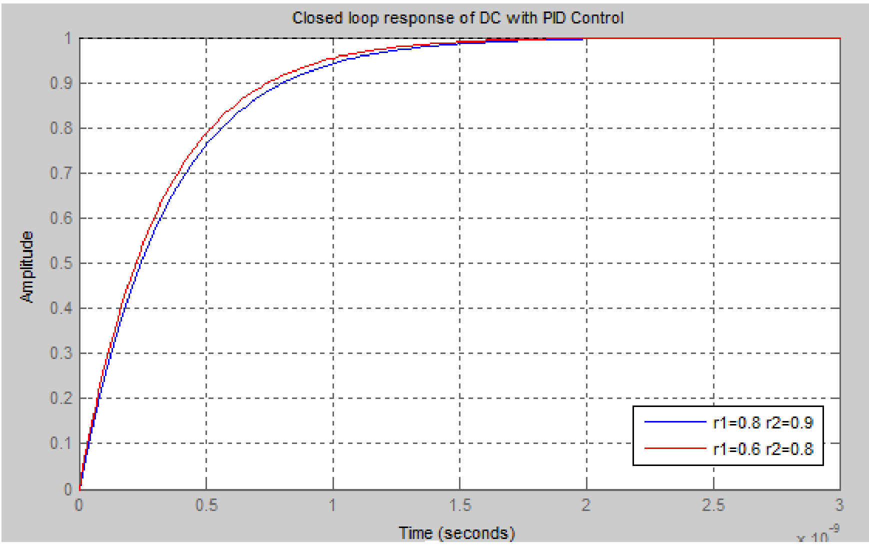


Figure 15: Comparison between IWPSO-PID r1=0.6 r2=0.8 and r1=0.8 r2=0.9

Based on system response from Figure 15, it can be concluded that the best control performance is when r1=0.6 r2=0.8, since it gives the faster response with shorter rise time and settling time compared to r1=0.8 r2=0.9.

4.5 Constriction Factor PSO-PID Controller (CFPSO-PID)

CFPSO is one of variants from PSO which is used to control the convergence of swarms. CFPSO is expected to have the best swarm convergence and control performance compared to other two PSO variants. CFPSO is not required to vary random number (r1 & r2) like in PSO and IWPSO but the variation is performed in acceleration coefficient.

4.5.1 Effect of acceleration coefficients (c1 & c2).

CFPSO requires the investigation of acceleration coefficient. From the literature review that has been conducted the best value of c1 and c2 is when c1+c2=4.1 [1], [2], [12]. It verifies by the simulation result shown in APPENDIX IV. Table 10 is the simulation result of CFPSO-PID:

Table 10: CFPSO varying c1 & c2

Parameters: Particles:20	kp= 21.1186	**kp= 21.8579**	kp= 11.8071	**kp= 22.1208**	kp= 25.3137	**kp= 21.2232**	kp= 15.4387
	ki= 4.92184	**ki= 4.97943**	ki= 2.52951	**ki= 5.00311**	ki= 5.25684	**ki= 4.58339**	ki= 4.32187
	kd= 35.1977	**kd= 36.4299**	kd= 19.6785	**kd= 36.8679**	kd= 42.1894	**kd= 35.372**	kd= 25.7311
	c1=2 c2=2.05	**c1=2 c2=2.1**	c1=2.05 c2=2	**c1=c2=2.05**	c1=2.05 c2=2.1	**c1=2.1 c2=2**	c1=2.1 c2=2.05
SSV	7.55081	**7.58917**	7.51028	**7.59374**	7.60055	**7.53501**	7.5069
MSE	MSE: 0.267662	**MSE: 0.231519**	MSE: 0.393713	**MSE: 0.242612**	MSE: 0.330518	**MSE: 0.340579**	MSE: 0.457989
Iteration	iter: 173	**iter: 157**	iter: 479	**iter: 158**	iter: 150	**iter: 229**	iter: 231

According to Table 10, the result with the lowest MSE is shown when c1+c2=4.1, it verifies that the best solution exists within this range. Table 11 is the result of variation of CFPSO with smallest MSE value:

Table 11: CFPSO with smallest MSE

Particle :20, c1+c2=4.1	c1=2 c2=2.1	c1=c2=2.05	c1=2.1 c2=2
	kp= 21.8579	kp= 22.1208	kp= 21.2232
	ki= 4.97943	ki= 5.00311	ki= 4.58339
	kd= 36.4299	kd= 36.8679	kd= 35.372
Steady State Value	7.58917	7.59374	7.53501
MSE	MSE: 0.231519	MSE: 0.242612	MSE: 0.340579
Iteration	iter: 157	iter: 158	iter: 229

From Table 11, when c1= 2 and c2=2.1 it will give the lowest value of MSE. Figure 16 shows the system response of the lowest MSE value simulated in MATLAB:

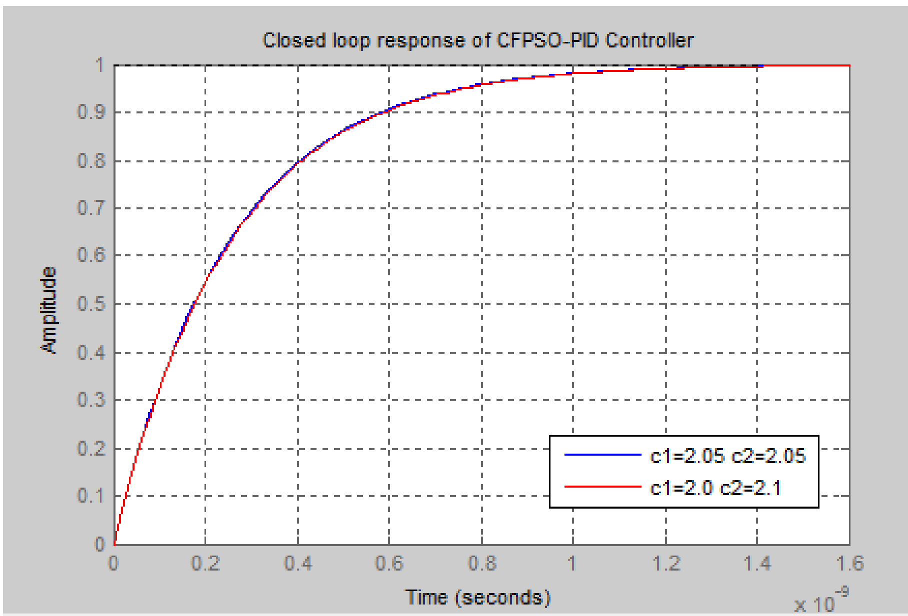


Figure 16: Comparison between CFPSO-PID c1=2.05.6 r2=2.05 and c1=2.0 r2=2.1

According to system response shown in Figure 16, it can be observe that both of them give the same control performance. Since c1=2 and c2=2.1 has the lowest MSE it is chosen as the best performance. In conclusion in order to get the best solution c2 requires to be higher than c1 and c1+c2=4.1.

4.6 Comparison between CPID, PSO-PID, IWPSO-PID and CFPSO-PID

After simulating in C++ and getting the best value of Kp, Ki, and Kd from PSO IWPSO and CFPSO, now it can be simulated in MATLAB in order to determine control performance like rise time, settling time, overshoot and steady state error. Figure 17 shows the comparison of simulated result in MATLAB between PSO and IWPSO:

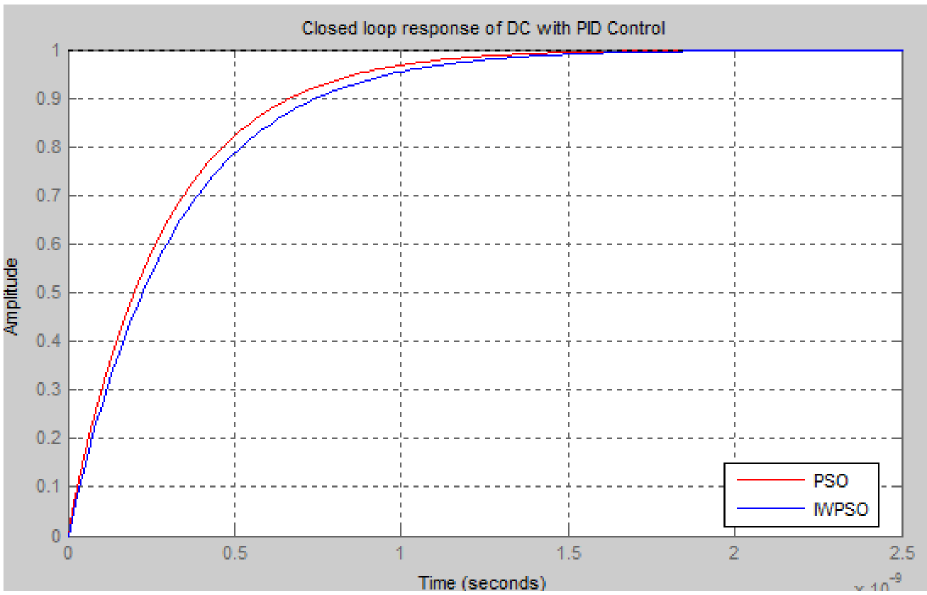


Figure 17: Comparison between PSO-PID and IWPSO-PID

From the system response shown in Figure 17, it can be concluded that PSO has the faster response time compare to IWPSO. Therefore it concludes that PSO has better performance even though it has higher MSE than IWPSO. Next PSO and CFPSO are required to be compared in order to determine the best tuning method. Figure 18 is the comparison between PSO and CFPSO simulated in MATLAB:

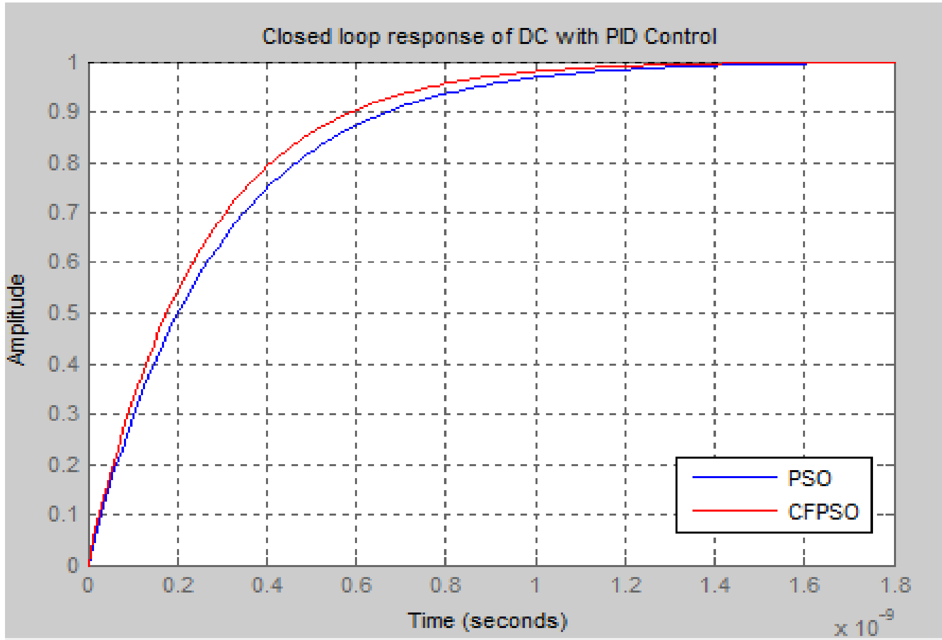


Figure 18: Comparison between PSO-PID and CFPSO-PID

From the system response shown in Figure 18, it can be concluded that CFPSO is regarded as the best tuning method. Table 12 shows the comparison of the control performance as well as MSE for all tuning method utilize in this project.

Table 12: The Comparison between conventional tuning methods and PSO, IWPSO & CFPSO-PID

Performance Criteria	Before Tuning	ZN Tuning	Auto Tuning	PSO-PID	IWPSO-PID	CFPSO-PID
Kp	-	0.1726	0.22497	19.2556	17.2648	21.8579
Ki	-	8.63	317.63	4.65022	4.64127	4.97943
Kd	-	$8.63x10^{-3}$	$-5.02x10^{-5}$	32.0926	28.7746	36.4299
Overshoot, Mp	10.747%	0%	8.71%	0%	0%	0%
Rise Time, Tr (sec)	$0.21x10^{-3}$	0.053	$6.581x10^{-4}$	$4.67x10^{-10}$	$5.2x10^{-10}$	$4.15x10^{-10}$
Settling Time, Ts(sec)	$0.69x10^{-3}$	0.105	$2.02x10^{-3}$	$8.74x10^{-10}$	$1x10^{-9}$	$7.62x10^{-10}$
Steady State Error, Ess	0.116	0	0	0	0	0
MSE	-	-	-	0.564082	0.436466	0.231519
Iteration	-	-	-	iter: 93	iter: 94	iter: 157

According to Table 12, PSO-PID and its variants are able to yield 0% overshoot while conventional PID tuning method has overshoot around 8.271%. PSO is able to eliminate the overshoot as well as has the faster rise time and settling time compared to CPID. PSO-PID has faster response compared to IWPSO, indicated by rise time and settling time value shown in Table 12. However IWPSO has smaller MSE and the solution is converged compare to PSO. Comparing between PSO and CFPSO, according to Table 12 CFPSO has faster time response at same tine smaller MSE value. Therefore CFPSO is regarded as the best tuning method compare to other tuning methods.

CHAPTER 5
CONCLUSION & RECOMMENDATION

5.1 Conclusion

Swarm Intelligence (SI) Algorithm has emerged as one of features integrated with PID-Controller to improve the control performance of non-linear system. Particle Swarm Optimization (PSO) Algorithm has proposed by Kennedy and Eberhart in 1995. PSO Algorithm is originated from research on swarm such as fish schooling and bird flocking. According to literature review, it shows that PSO in combination with other techniques has shown capabilities in solving control limitation in various applications. Furthermore it is anticipated for further research would yield a further result.

Simulation has been done for PSO-PID controller and its variants. From the simulation it proves that PSO-PID has better control performance compare to CPID. PSO successfully eliminates the overshoot at the same time has faster response. Furthermore PSO variants verify to be able to control the convergence by indicating smaller value of MSE. Finally CFPSO has regarded as the best tuning method in this project because it has the best control performance as well as the lowest value of MSE.

5.2 Recommendation

Further study in this project is really recommended. There are still some improvements need to be done in this project. PSO-PID Controller-GUI developed by the author still needs some improvements. The GUI is possible to develop so it can show the control performance of system response instead of doing it manually. Auto tuning is encouraged to develop so this system can model every non-liner plant. Furthermore PSO-PID Controller-GUI as successfully achieved the objectives of this project, and it is expected to contribute to the improvement of control technique.

REFERENCES

[1] Gaing, Z. L. (2004). *A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System.*IEE Trans. Energy. Convers., 19, 384-391

[2] Marghouzi, A., Selamat, H., & Rahmat, M. F. (2012). *Optimized proportional Integral derivative (PID) controller for the exhaust temperature control of a gas turbine system using particle swarm optimization.* Int. J. Phys. Sci., 7(5), 720-729.

[3] Luoren, L., Jinling, L. (2011).*Research of PID Control Algorithm Based on Neural Network.* Energy Precedia, 13:6989-6993

[4] Kim, Y. H., Ahn, S. C., & Kwon, W. H. (2000). *Computational Complexity of general fuzzy logic control and its simplification of a loop controller.* Fuzzy Sets and System. 11:215-224

[5] Fang, H., Chen, L., Wang, W. (2008). *Comparison of PSO and its variants to optimal parameters of PID controller*. Chinese Control and Decision Conference. 3100-3104.

[6] Nasri, M., Nezamabadi-pour, H., & Maghfoori, M. (2007). *A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor.*World Academy of Sci., Eng & Tech., 26, 211-215.

[7] Astrom, K..J., & Hagglund, T. (2004). *Revisiting the Ziegler-Nichols step response for PID control.* Journal of Process Control. 14:635-650.

[8] Pillay, N. (2008). *A Particle Swarm Optimization Approach for Tuning of SISO PID Control loops*.

[9] Nise, N. S. (2008). *Control System Engineering* (5^{th}ed). Wiley & Sons

[10] Engelbrecht, A. P. (2005). *Fundamental of Computational Swarm Intelligence*. Wet Sussex: Wiley & Sons.

[11] Bansal, J.C., et al. (2011). *Inertia Weight Strategies in Particle Swarm Optimization.*

[12] Pillay, N. (2008). *A Particle Swarm Optimization Approach for Tuning of SISO PID Control loops*.

[13] Allaoua, B., Gasbaoui, B., & Mebarki, B. (2009). *Setting up PID DC motor speed control alteration Parameter using PSO.* Leonardo Electronic Journal of Practices and Tech

[14] Yu, K.W., & Hu, S. C. (2006). *An Application of AC servo motor by using PSO based sliding mode controller (SMC).*IEE conference on Systems, Man, and Cybernetic.

[15] Verma, H.K., & Jain C. (2011). *A Performance-Dependent PSO based Optimization of PID Controller for DC Motor.* International Conference on Electrical Energy System.

[16] Rajasekaran, S., &Kannadasan, T. (2012). *Swarm Optimization based Controller for Temperature Control of a heat Exchanger.* International Journal of Computer Application 38(4):6-11.

[17] Gaing, Z. L., & Chang R. F. (2009). *Optimal PID Controller for High-Speed Rail Pantograph System with Notch Filter.* TESCON.

[18] Samanta, B. (2008). *Design of Intelligent Ship Autopilots using Particle Swarm Optimization*. IEE Swarm Intelligence Symposium

[19] Kao, C. C., Chuang, C.W., & Fung R.F. (2006). *The self tuning PID control in a slider –crank mechanism system by applying PSO approach.* Mechatronics 16:513-522.

[20] Qiming, C., Yinman, C., Ruiqing, G., & Yong, Z. (2009). *The forward NN- PID Controllers based on Chaos PSO-BP Hibrid Optimization Algorithms for Decoupling Control System of Ball Mill*.International Conference Artificial Intelligence and Computational Intelligence

[21] Rana, M. A., Usman, Z., &Shareef, Z. (2011*). Automatic Control of Ball and Beam System using Particle Swarm Optimization*.12th IEE CINTI.

[22] Payakkawan, P., Klomkarn, K., &Sooraksa, P. (2009). *Dual-line PID Controller based on PSO for Speed Control of DC Motor*. ISCIT

[23] Jalilvand, A., Kimiyaghalam A., Ashouri A., & Mahdavi M. (2008).*Advance Particle Swarm Optimization (APSO) Based PID Controller Parameters Tuning*.12th IEE International Multitopic Conference

[24] Jain, T., & Nigam, M. J. (2008).*Optimization of PD-PI controller using Swarm Intelligence*. Journal of Theoretical and Applied Information Technology.

[25] Xu-zhou, L., Fei, Y., & You-bo, W. (2007). *PSO Algorithm based Online Self Tuning of PID controller*. Int. Conf. on Computational Intelligence Security,128-132.

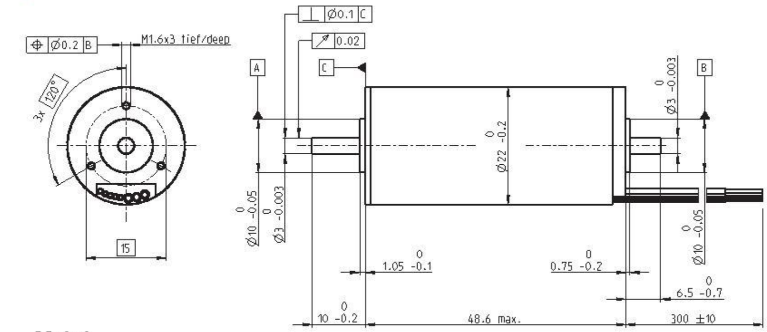
[26] El-Gammmal, A. A., & El-Samahy, A. A. (2009). *A Modified Design of PID Controller Using Particle Swarm Optimization PSO*. POWERENG.

APPENDIX I: DC Motor Datasheet

EC-4pole 22 ∅22 mm, brushless, 90 Watt

High Power

maxon *EC*-4pole



M 1:1

- Stock program
- Standard program
- Special program (on request)

Motor Data (provisional)		Order Number 323217	323218	323219	323220	327739
Values at nominal voltage						
1 Nominal voltage	V	18.0	24.0	36.0	48.0	48.0
2 No load speed	rpm	16200	16200	16200	16200	6900
3 No load current	mA	275	206	137	103	21.5
4 Nominal speed	rpm	14700	14800	14700	14700	5330
5 Nominal torque (max. continuous torque)	mNm	49.4	51.6	50.5	49.4	51.2
6 Nominal current (max. continuous current)	A	4.89	3.82	2.50	1.83	0.786
7 Stall torque	mNm	588	639	612	586	234
8 Starting current	A	55.8	45.5	29.1	20.9	3.55
9 Max. efficiency	%	87	87	87	87	85
Characteristics						
10 Terminal resistance phase to phase	Ω	0.323	0.527	1.24	2.3	13.5
11 Terminal inductance phase to phase	mH	0.0283	0.0503	0.113	0.201	1.11
12 Torque constant	mNm / A	10.5	14.0	21.1	28.1	66.0
13 Speed constant	rpm / V	907	680	453	340	145
14 Speed / torque gradient	rpm / mNm	27.8	25.5	26.7	27.9	29.7
15 Mechanical time constant	ms	1.61	1.48	1.55	1.62	1.72
16 Rotor inertia	gcm^2	5.54	5.54	5.54	5.54	5.54

Specifications

Thermal data

17 Thermal resistance housing-ambient	9.08 K / W
18 Thermal resistance winding-housing	0.904 K / W
19 Thermal time constant winding	3.98 s
20 Thermal time constant motor	358 s
21 Ambient temperature	-20 ... +100°C
22 Max. permissible winding temperature	+155°C

Mechanical data (preloaded ball bearings)

23 Max. permissible speed	25000 rpm
24 Axial play at axial load < 5.0 N	0 mm
> 5.0 N	0.14 mm
25 Radial play	preloaded
26 Max. axial load (dynamic)	4.5 N
27 Max. force for press fits (static)	53 N
(static, shaft supported)	1000 N
28 Max. radial loading, 5 mm from flange	16 N

Other specifications

29 Number of pole pairs	2
30 Number of phases	3
31 Weight of motor	125 g

Values listed in the table are nominal.

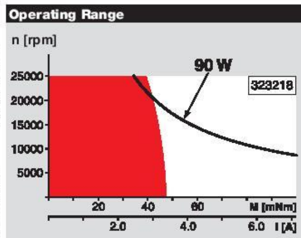
Connection Motor (Cable AWG 20)

red	Motor winding 1
white	Motor winding 3
black	Motor winding 2

Connection sensors (Cable AWG 26)

red/grey	Hall sensor 1
black/grey	Hall sensor 2
white/grey	Hall sensor 3
green	V_{Hall} 4.5 … 24 VDC
blue	GND

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

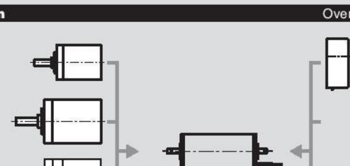
maxon Modular System

Overview on page 16 - 21

Planetary Gearhead
∅22 mm
2.0 - 3.4 Nm
Page 225

Planetary Gearhead
∅32 mm
1.0 - 6.0 Nm
Page 234

Spindle Drive
∅32 mm
Page 249 / 250 / 251



Encoder HEDL 5540
500 Imp.,
3 channels
Page 270

Recommended Electronics:

DECS 50/5	Page 289
DEC 50/5	291
DEC Module 50/5	**291**
DECV 50/5	297
DEC 70/10	297
DES 50/5, DES 70/10	298
EPOS2 24/5	305
EPOS2 50/5	305
EPOS 70/10	305
EPOS P 24/5	**308**
Notes	**20**

May 2010 edition / subject to change

maxon EC motor 173

APPENDIX II: PSO-PID Simulation Result

Parameters: Particles:20, r1=0.5, C1= 1.0 & C2=1.2		kp= 18.6699 ki= 4.54712 kd= 31.1164	kp= 19.4341 ki= 4.6755 kd= 32.3902	kp= 19.8181 ki= 4.73232 kd= 33.0302	kp= 20.5575 ki= 4.86711 kd= 34.2624	kp= 21.2447 ki= 5.00271 kd= 35.4079
Time	Y Model	r1=0.5 r2=0.5	r1=0.5 r2=0.6	r1=0.5 r2=0.7	r1=0.5 r2=0.8	r1=0.5 r2=0.9
1	5.10348	5.72246	5.75474	5.76921	5.79943	5.88522
2	6.86175	7.01535	7.03097	7.03647	7.05022	7.14059
3	7.37671	7.35896	7.36827	7.37042	7.37796	7.46838
4	7.52753	7.4596	7.46706	7.46823	7.47395	7.56438
5	7.5717	7.48907	7.49599	7.49687	7.50206	7.5925
6	7.58464	7.4977	7.50446	7.50526	7.51029	7.60073
7	7.58843	7.50023	7.50695	7.50772	7.5127	7.60314
8	7.58954	7.50097	7.50767	7.50844	7.51341	7.60385
9	7.58987	7.50119	7.50789	7.50865	7.51362	7.60406
10	7.58996	7.50125	7.50795	7.50871	7.5136	7.60412
11	7.58999	7.50127	7.50796	7.50873	7.5136	7.60414
12	7.59	7.50128	7.50797	7.50873	7.5137	7.60414
13	7.59	7.50128	7.50797	7.50873	7.5137	7.60414
14	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
15	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
16	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
17	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
18	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
19	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
20	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
21	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
22	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
23	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
24	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
25	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
26	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
27	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
28	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
29	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
30	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
31	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
32	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
33	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
34	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
35	7.59	7.50128	7.50798	7.50873	7.5137	7.60414

36	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
37	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
38	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
39	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
40	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
41	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
42	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
43	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
44	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
45	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
46	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
47	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
48	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
49	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
50	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
51	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
52	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
53	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
54	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
55	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
56	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
57	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
58	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
59	7.59	7.50128	7.50798	7.50873	7.5137	7.60414
		MSE: 0.843102	MSE: 0.825143	MSE: 0.839097	MSE: 0.841533	MSE: 0.709952
		iter: 94	iter: 92	iter: 90	iter: 88	iter: 86

Parameters: Particles:20, r1=0.6, C1= 1.0 & C2=1.2		kp= 17.903	kp= 18.5893	kp= 19.3663	kp= 19.5863	kp= 20.4463
		ki= 4.42472	ki= 4.53114	ki= 4.67036	ki= 4.69	ki= 4.85839
		kd= 29.8384	kd= 30.9822	kd= 32.2772	kd= 32.6438	kd= 34.0772
Time	Y Model	r1=0.6 r2=0.5	r1=0.6 r2=0.6	r1=0.6 r2=0.7	r1=0.6 r2=0.8	r1=0.6 r2=0.9
1	5.10348	5.77804	5.74844	5.80516	5.76427	5.85246
2	6.86175	7.11795	7.0532	7.10122	7.03825	7.12556
3	7.37671	7.47677	7.40044	7.44446	7.3746	7.46004
4	7.52753	7.58186	7.50214	7.54498	7.47311	7.55801
5	7.5717	7.61264	7.53193	7.57442	7.50196	7.5867
6	7.58464	7.62165	7.54065	7.58305	7.51041	7.5951
7	7.58843	7.62429	7.54321	7.58557	7.51289	7.59756
8	7.58954	7.62507	7.54395	7.58631	7.51361	7.59828
9	7.58987	7.62529	7.54417	7.58653	7.51382	7.59849
10	7.58996	7.62536	7.54424	7.58659	7.51389	7.59855
11	7.58999	7.62538	7.54426	7.58661	7.5139	7.59857

12	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
13	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
14	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
15	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
16	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
17	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
18	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
19	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
20	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
21	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
22	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
23	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
24	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
25	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
26	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
27	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
28	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
29	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
30	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
31	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
32	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
33	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
34	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
35	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
36	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
37	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
38	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
39	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
40	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
41	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
42	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
43	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
44	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
45	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
46	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
47	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
48	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
49	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
50	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
51	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
52	7.59	7.62539	7.54426	7.58662	7.51391	7.59858

53	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
54	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
55	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
56	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
57	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
58	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
59	7.59	7.62539	7.54426	7.58662	7.51391	7.59858
		MSE: 0.603098	MSE: 0.568158	MSE: 0.555211	MSE: 0.787859	MSE: 0.642699
		iter: 99	iter: 95	iter: 93	iter: 91	iter: 89

Parameters: Particles:20, r1=0.7, C1= 1.0 & C2=1.2		kp= 17.0737	kp= 17.8323	kp= 18.7131	kp= 19.0333	kp= 19.9586
		ki= 4.282	ki= 4.40319	ki= 4.55643	ki= 4.60383	ki= 4.77803
		kd= 28.4562	kd= 29.7204	kd= 31.1886	kd= 31.7222	kd= 33.2644
Time	Y Model	r1=0.7 r2=0.5	r1=0.7 r2=0.6	r1=0.7 r2=0.7	r1=0.7 r2=0.8	r1=0.7 r2=0.9
1	5.10348	5.70763	5.68302	5.7712	5.76013	5.85368
2	6.86175	7.05342	6.99276	7.07931	7.05278	7.14494
3	7.37671	7.41553	7.34289	7.4273	7.39564	7.48565
4	7.52753	7.52158	7.44543	7.52921	7.49605	7.58544
5	7.5717	7.55264	7.47546	7.55906	7.52547	7.61466
6	7.58464	7.56174	7.48426	7.56781	7.53408	7.62322
7	7.58843	7.5644	7.48683	7.57037	7.5366	7.62573
8	7.58954	7.56518	7.48759	7.57112	7.53734	7.62646
9	7.58987	7.56541	7.48781	7.57133	7.53756	7.62668
10	7.58996	7.56548	7.48787	7.5714	7.53762	7.62674
11	7.58999	7.56549	7.48789	7.57142	7.53764	7.62676
12	7.59	7.5655	7.4879	7.57143	7.53764	7.62676
13	7.59	7.5655	7.4879	7.57143	7.53764	7.62676
14	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
15	7.59	7.5655	7.4879	7.57143	7.53765	7.62677
16	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
17	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
18	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
19	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
20	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
21	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
22	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
23	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
24	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
25	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
26	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
27	7.59	7.5655	7.4879	7.57143	7.53765	7.62676

28	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
29	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
30	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
31	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
32	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
33	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
34	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
35	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
36	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
37	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
38	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
39	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
40	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
41	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
42	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
43	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
44	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
45	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
46	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
47	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
48	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
49	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
50	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
51	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
52	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
53	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
54	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
55	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
56	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
57	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
58	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
59	7.59	7.5655	7.4879	7.57143	7.53765	7.62676
		MSE: 0.435939	MSE: 0.932614	MSE: 0.514451	MSE: 0.618922	MSE: 0.733253
		iter: 101	iter: 97	iter: 95	iter: 93	iter: 91

Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2		kp= 16.6333	kp= 17.5261	kp= 18.1706	kp= 18.6892	kp= 19.2556
		ki= 4.22423	ki= 4.36644	ki= 4.46928	ki= 4.55633	ki= 4.65022
		kd= 27.7222	kd= 29.2102	kd= 30.2844	kd= 31.1486	kd= 32.0926
Time	Y Model	r1=0.8 r2=0.5	r1=0.8 r2=0.6	r1=0.8 r2=0.7	r1=0.8 r2=0.8	r1=0.8 r2=0.9
1	5.10348	5.68018	5.74023	5.76787	5.77278	5.80716
2	6.86175	7.03253	7.08081	7.09455	7.0823	7.10793

3	7.37671	7.39738	7.44054	7.44899	7.43074	7.45275
4	7.52753	7.50424	7.54589	7.5528	7.53279	7.55374
5	7.5717	7.53553	7.57675	7.5832	7.56268	7.58332
6	7.58464	7.5447	7.58578	7.59211	7.57144	7.59199
7	7.58843	7.54738	7.58843	7.59472	7.574	7.59452
8	7.58954	7.54817	7.58921	7.59548	7.57475	7.59527
9	7.58987	7.5484	7.58943	7.5957	7.57497	7.59548
10	7.58996	7.54847	7.5895	7.59577	7.57504	7.59555
11	7.58999	7.54848	7.58952	7.59579	7.57506	7.59557
12	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
13	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
14	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
15	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
16	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
17	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
18	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
19	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
20	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
21	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
22	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
23	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
24	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
25	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
26	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
27	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
28	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
29	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
30	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
31	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
32	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
33	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
34	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
35	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
36	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
37	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
38	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
39	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
40	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
41	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
42	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
43	7.59	7.54849	7.58953	7.5958	7.57506	7.59557

44	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
45	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
46	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
47	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
48	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
49	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
50	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
51	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
52	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
53	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
54	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
55	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
56	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
57	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
58	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
59	7.59	7.54849	7.58953	7.5958	7.57506	7.59557
		MSE: 0.456883	MSE: 0.457884	MSE: 0.503451	MSE: 0.511611	MSE: 0.564082
		iter: 103	iter: 100	iter: 97	iter: 95	iter: 93

Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2		kp= 16.1508 ki= 4.14349 kd= 26.918	kp= 17.0977 ki= 4.2973 kd= 28.4961	kp= 17.8242 ki= 4.42063 kd= 29.7069	kp= 18.3953 ki= 4.51689 kd= 30.6589	kp= 18.7231 ki= 4.55639 kd= 31.2052
Time	Y Model	r1=0.9 r2=0.5	r1=0.9 r2=0.6	r1=0.9 r2=0.7	r1=0.9 r2=0.8	r1=0.9 r2=0.9
1	5.10348	5.63763	5.68751	5.75364	5.75893	5.71493
2	6.86175	6.99286	7.02551	7.08801	7.07436	7.00328
3	7.37671	7.35945	7.38528	7.44536	7.42509	7.34547
4	7.52753	7.46682	7.49065	7.55002	7.52781	7.44568
5	7.5717	7.49827	7.52151	7.58067	7.5579	7.47504
6	7.58464	7.50748	7.53055	7.58964	7.56671	7.48363
7	7.58843	7.51018	7.5332	7.59228	7.56929	7.48615
8	7.58954	7.51096	7.53397	7.59304	7.57004	7.48689
9	7.58987	7.5112	7.5342	7.59327	7.57027	7.4871
10	7.58996	7.51126	7.53427	7.59334	7.57033	7.48717
11	7.58999	7.51128	7.53429	7.59336	7.57035	7.48718
12	7.59	7.51129	7.53429	7.59336	7.57035	7.48719
13	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
14	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
15	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
16	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
17	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
18	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
19	7.59	7.51129	7.53429	7.59336	7.57036	7.48719

20	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
21	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
22	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
23	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
24	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
25	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
26	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
27	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
28	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
29	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
30	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
31	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
32	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
33	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
34	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
35	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
36	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
37	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
38	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
39	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
40	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
41	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
42	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
43	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
44	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
45	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
46	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
47	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
48	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
49	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
50	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
51	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
52	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
53	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
54	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
55	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
56	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
57	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
58	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
59	7.59	7.51129	7.53429	7.59336	7.57036	7.48719
		MSE: 0.646069	MSE: 0.539189	MSE: 0.479829	MSE: 0.4981	MSE: 0.981144
		iter: 105	iter: 101	iter: 98	iter: 96	iter: 94

APPENDIX III: IWPSO PID Simulation Result

Parameters: Particles:20, r1=0.5, C1= 1.0 & C2=1.2		kp= 14.8245	kp= 15.2105	kp= 15.8771	kp= 15.046	kp= 15.2102
		ki= 4.13863	ki= 4.18902	ki= 4.31114	ki= 4.09031	ki= 4.10644
		kd= 24.7074	kd= 25.3509	kd= 26.4619	kd= 25.0767	kd= 25.3504
Time	Y Model	r1=0.5 r2=0.5	r1=0.5 r2=0.6	r1=0.5 r2=0.7	r1=0.5 r2=0.8	r1=0.5 r2=0.9
1	5.10348	5.60646	5.64035	5.6768	5.60801	5.63632
2	6.86175	6.99916	7.03069	7.05547	6.99319	7.02529
3	7.37671	7.37922	7.40934	7.42945	7.37064	7.40354
4	7.52753	7.49053	7.52023	7.53898	7.48118	7.51432
5	7.5717	7.52313	7.55271	7.57106	7.51356	7.54676
6	7.58464	7.53268	7.56222	7.58045	7.52304	7.55626
7	7.58843	7.53548	7.56501	7.58321	7.52582	7.55905
8	7.58954	7.53629	7.56583	7.58401	7.52663	7.55986
9	7.58987	7.53653	7.56606	7.58424	7.52687	7.5601
10	7.58996	7.53661	7.56613	7.58431	7.52694	7.5601
11	7.58999	7.53663	7.56615	7.58433	7.52696	7.5601
12	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
13	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
14	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
15	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
16	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
17	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
18	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
19	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
20	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
21	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
22	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
23	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
24	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
25	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
26	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
27	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
28	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
29	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
30	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
31	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
32	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
33	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
34	7.59	7.53663	7.56616	7.58434	7.52697	7.5602

35	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
36	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
37	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
38	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
39	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
40	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
41	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
42	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
43	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
44	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
45	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
46	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
47	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
48	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
49	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
50	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
51	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
52	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
53	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
54	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
55	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
56	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
57	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
58	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
59	7.59	7.53663	7.56616	7.58434	7.52697	7.5602
		MSE: 0.429182	MSE: 0.348836	MSE: 0.370847	MSE: 0.491678	MSE: 0.360028
		iter: 106	iter: 105	iter: 102	iter: 107	iter: 107

Parameters: Particles:20, r1=0.6, C1= 1.0 & C2=1.2		kp= 14.334	kp= 14.9154	kp= 16.643	kp= 17.2648	kp= 16.496
		ki= 4.06826	ki= 4.1574	ki= 4.52408	ki= 4.64127	ki= 4.44165
		kd= 23.8899	kd= 24.8591	kd= 27.7383	kd= 28.7746	kd= 27.4933
Time	Y Model	r1=0.6 r2=0.5	r1=0.6 r2=0.6	r1=0.6 r2=0.7	r1=0.6 r2=0.8	r1=0.6 r2=0.9
1	5.10348	5.59349	5.5944	5.71616	5.72238	5.71709
2	6.86175	6.99955	6.97965	7.08055	7.06632	7.08714
3	7.37671	7.38445	7.35735	7.44891	7.42752	7.45744
4	7.52753	7.49717	7.46798	7.55679	7.53331	7.56589
5	7.5717	7.53019	7.50037	7.58839	7.56429	7.59765
6	7.58464	7.53986	7.50986	7.59764	7.57336	7.60695
7	7.58843	7.54269	7.51264	7.60035	7.57602	7.60968
8	7.58954	7.54352	7.51346	7.60115	7.5768	7.61047
9	7.58987	7.54376	7.51369	7.60138	7.57703	7.61071
10	7.58996	7.54384	7.51376	7.60145	7.57709	7.61078

11	7.58999	7.54386	7.51378	7.60147	7.57711	7.6108
12	7.59	7.54386	7.51379	7.60147	7.57712	7.6108
13	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
14	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
15	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
16	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
17	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
18	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
19	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
20	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
21	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
22	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
23	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
24	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
25	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
26	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
27	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
28	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
29	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
30	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
31	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
32	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
33	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
34	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
35	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
36	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
37	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
38	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
39	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
40	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
41	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
42	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
43	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
44	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
45	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
46	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
47	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
48	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
49	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
50	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
51	7.59	7.54386	7.51379	7.60148	7.57712	7.6108

52	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
53	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
54	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
55	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
56	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
57	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
58	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
59	7.59	7.54386	7.51379	7.60148	7.57712	7.6108
		MSE: 0.376577	MSE: 0.57722	MSE: 0.436764	MSE: 0.436466	MSE: 0.459436
		iter: 108	iter: 105	iter: 97	iter: 94	iter: 99

Parameters: Particles:20, r1=0.7, C1= 1.0 & C2=1.2		kp= 14.1495	kp= 15.9144	kp= 14.1422	kp= 14.5629	kp= 14.9561
		ki= 4.05974	ki= 4.4174	ki= 3.98416	ki= 4.04944	ki= 4.11674
		kd= 23.5825	kd= 26.524	kd= 23.5703	kd= 24.2714	kd= 24.9269
Time	Y Model	r1=0.7 r2=0.5	r1=0.7 r2=0.6	r1=0.7 r2=0.7	r1=0.7 r2=0.8	r1=0.7 r2=0.9
1	5.10348	5.57983	5.66546	5.57623	5.58984	5.64662
2	6.86175	6.98789	7.03887	6.98332	6.98633	7.04837
3	7.37671	7.37372	7.41124	7.36888	7.36799	7.43084
4	7.52753	7.48672	7.52029	7.4818	7.47977	7.54285
5	7.5717	7.51981	7.55224	7.51487	7.51251	7.57566
6	7.58464	7.52951	7.56159	7.52456	7.5221	7.58527
7	7.58843	7.53235	7.56433	7.5274	7.5249	7.58808
8	7.58954	7.53318	7.56513	7.52823	7.52573	7.5889
9	7.58987	7.53342	7.56537	7.52847	7.52597	7.58915
10	7.58996	7.53349	7.56544	7.52854	7.52604	7.58922
11	7.58999	7.53351	7.56545	7.52856	7.52606	7.58924
12	7.59	7.53352	7.56546	7.52857	7.52607	7.58924
13	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
14	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
15	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
16	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
17	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
18	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
19	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
20	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
21	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
22	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
23	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
24	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
25	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
26	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
27	7.59	7.53352	7.56547	7.52857	7.52607	7.58925

28	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
29	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
30	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
31	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
32	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
33	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
34	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
35	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
36	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
37	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
38	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
39	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
40	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
41	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
42	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
43	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
44	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
45	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
46	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
47	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
48	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
49	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
50	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
51	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
52	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
53	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
54	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
55	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
56	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
57	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
58	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
59	7.59	7.53352	7.56547	7.52857	7.52607	7.58925
		MSE: 0.41923	MSE: 0.381223	MSE: 0.447195	MSE: 0.478399	MSE: 0.33304
		iter: 108	iter: 99	iter: 110	iter: 108	iter: 107

Parameters: Particles:20, r1=0.8, C1= 1.0 & C2=1.2		kp= 13.4004	kp= 14.5066	kp= 14.9156	kp= 15.2658	kp= 16.0156
		ki= 3.92482	ki= 4.12778	ki= 4.18909	ki= 4.24113	ki= 4.38631
		kd= 22.334	kd= 24.1777	kd= 24.8594	kd= 25.443	kd= 26.6927
Time	Y Model	r1=0.8 r2=0.5	r1=0.8 r2=0.6	r1=0.8 r2=0.7	r1=0.8 r2=0.8	r1=0.8 r2=0.9
1	5.10348	5.55592	5.59552	5.64317	5.67642	5.691
2	6.86175	6.98297	6.996	7.0452	7.07718	7.06953

3	7.37671	7.37577	7.37894	7.42783	7.45877	7.44321
4	7.52753	7.49081	7.4911	7.53989	7.57053	7.55265
5	7.5717	7.5245	7.52394	7.57271	7.60327	7.58471
6	7.58464	7.53437	7.53356	7.58232	7.61285	7.59409
7	7.58843	7.53726	7.53638	7.58514	7.61566	7.59684
8	7.58954	7.53811	7.53721	7.58596	7.61648	7.59765
9	7.58987	7.53835	7.53745	7.5862	7.61672	7.59788
10	7.58996	7.53843	7.53752	7.5862	7.61679	7.59795
11	7.58999	7.53845	7.53754	7.5863	7.61681	7.59797
12	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
13	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
14	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
15	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
16	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
17	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
18	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
19	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
20	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
21	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
22	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
23	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
24	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
25	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
26	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
27	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
28	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
29	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
30	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
31	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
32	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
33	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
34	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
35	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
36	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
37	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
38	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
39	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
40	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
41	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
42	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
43	7.59	7.53846	7.53755	7.5863	7.61682	7.59798

44	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
45	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
46	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
47	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
48	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
49	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
50	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
51	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
52	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
53	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
54	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
55	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
56	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
57	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
58	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
59	7.59	7.53846	7.53755	7.5863	7.61682	7.59798
		MSE: 0.366252	MSE: 0.412106	MSE: 0.328412	MSE: 0.423209	MSE: 0.397061
		iter: 112	iter: 106	iter: 105	iter: 104	iter: 100

Parameters: Particles:20, r1=098, C1= 1.0 & C2=1.2		kp= 13.1278 ki= 3.89635 kd= 21.8797	kp= 13.78 ki= 3.99401 kd= 22.9666	kp= 13.8082 ki= 3.96755 kd= 23.0137	kp= 14.2004 ki= 4.02798 kd= 23.6673	kp= 14.6556 ki= 4.10473 kd= 24.4261
Time	Y Model	r1=0.9 r2=0.5	r1=0.9 r2=0.6	r1=0.9 r2=0.7	r1=0.9 r2=0.8	r1=0.9 r2=0.9
1	5.10348	5.59529	5.5788	5.59011	5.64123	5.66969
2	6.86175	7.0458	6.99994	7.01412	7.06859	7.09032
3	7.37671	7.44599	7.39029	7.40526	7.45998	7.47888
4	7.52753	7.5632	7.50462	7.51982	7.57461	7.59268
5	7.5717	7.59752	7.5381	7.55337	7.60818	7.62601
6	7.58464	7.60758	7.54791	7.5632	7.61801	7.63577
7	7.58843	7.61052	7.55078	7.56608	7.62089	7.63863
8	7.58954	7.61138	7.55162	7.56692	7.62173	7.63946
9	7.58987	7.61164	7.55187	7.56717	7.62198	7.63971
10	7.58996	7.61171	7.55194	7.56724	7.62205	7.63978
11	7.58999	7.61173	7.55196	7.56726	7.62208	7.6398
12	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
13	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
14	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
15	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
16	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
17	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
18	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
19	7.59	7.61174	7.55197	7.56727	7.62208	7.63981

20	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
21	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
22	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
23	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
24	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
25	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
26	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
27	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
28	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
29	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
30	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
31	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
32	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
33	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
34	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
35	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
36	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
37	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
38	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
39	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
40	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
41	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
42	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
43	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
44	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
45	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
46	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
47	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
48	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
49	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
50	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
51	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
52	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
53	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
54	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
55	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
56	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
57	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
58	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
59	7.59	7.61174	7.55197	7.56727	7.62208	7.63981
		MSE: 0.308087	MSE: 0.324831	MSE: 0.289067	MSE: 0.398137	MSE: 0.524638
		iter: 114	iter: 110	iter: 111	iter: 110	iter: 108

APPENDIX IV: CFPSO-PID Simulation Result

Parameters: Particles:20		kp= 21.1186	kp= 21.8579	kp= 11.8071	kp= 22.1208	kp= 25.3137	kp= 21.2232	kp= 15.4387
		ki= 4.92184	ki= 4.97943	ki= 2.52951	ki= 5.00311	ki= 5.25684	ki= 4.58339	ki= 4.32187
		kd= 35.1977	kd= 36.4299	kd= 19.6785	kd= 36.8679	kd= 42.1894	kd= 35.372	kd= 25.7311
Time	Y Model	c1=2 c2=2.05	c1=2 c2=2.1	c1=2.05 c2=2	c1=c2=2.05	c1=2.05 c2=2.1	c1=2.1 c2=2	c1=2.1 c2=2.05
1	5.10348	5.51525	5.55638	5.29482	5.5649	5.63572	5.5068	5.36788
2	6.86175	6.97727	7.01723	6.87407	7.02323	7.05222	6.96373	6.89697
3	7.37671	7.38283	7.42166	7.32395	7.42665	7.43996	7.3677	7.32826
4	7.52753	7.50162	7.54011	7.45571	7.5448	7.55352	7.48601	7.45458
5	7.5717	7.5364	7.5748	7.4943	7.57941	7.58677	7.52066	7.49157
6	7.58464	7.54659	7.58496	7.5056	7.58954	7.59652	7.53081	7.50241
7	7.58843	7.54958	7.58794	7.50891	7.59251	7.59937	7.53378	7.50558
8	7.58954	7.55045	7.58881	7.50988	7.59338	7.6002	7.53465	7.50651
9	7.58987	7.55071	7.58906	7.51016	7.59363	7.60045	7.53491	7.50678
10	7.58996	7.55078	7.58914	7.51025	7.59371	7.60052	7.53498	7.5068
11	7.58999	7.5508	7.58916	7.51027	7.59373	7.60054	7.535	7.5068
12	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5068
13	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5068
14	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
15	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
16	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
17	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
18	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
19	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
20	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
21	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
22	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
23	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
24	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
25	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
26	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
27	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
28	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
29	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
30	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
31	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
32	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
33	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
34	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069

35	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
36	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
37	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
38	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
39	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
40	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
41	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
42	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
43	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
44	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
45	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
46	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
47	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
48	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
49	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
50	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
51	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
52	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
53	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
54	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
55	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
56	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
57	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
58	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
59	7.59	7.55081	7.58917	7.51028	7.59374	7.60055	7.53501	7.5069
		MSE: 0.267662	MSE: 0.231519	MSE: 0.393713	MSE: 0.242612	MSE: 0.330518	MSE: 0.340579	MSE: 0.457989
		iter: 173	iter: 157	iter: 479	iter: 158	iter: 150	iter: 229	iter: 231

APPENDIX V: SEDEX 30th Certificate: Bronze Medal

UNIVERSITI TEKNOLOGI PETRONAS

SCIENCE&ENGINEERING DESIGN EXHIBITION

Certificate of Participation

This certificate is awarded to

ARHIMNY HASDI AIMON
W202801

in recognition of her contribution as

Participant of
Final Year Project

"Bronze"

in the 30th edition of

SCIENCE AND ENGINEERING DESIGN EXHIBITION
UNIVERSITI TEKNOLOGI PETRONAS

held on
8 - 9 August 2012

Dr. Dennis Ling Chuan Ching
Chairman
30th Science and Engineering Design Exhibition

Prof. Ir. Dr. Ahmad Fadzil B. Mohamad Hani
Deputy Vice Chancellor, Academic
Universiti Teknologi PETRONAS

SCIENCE AND ENGINEERING DESIGN EXHIBITION
SEDEX
PORTAL TO THE FUTURE
UNIVERSITI TEKNOLOGI PETRONAS

APPENDIX VI:

Activities/Gantt Chart and Milestone of FYP1

No	Detail/Week	1	2	3	4	5	6	Mid Semester Break (7/02-11/02/2012)	7	8	9	10	11	12	13	14
1	Selection Project Topic: PID Tuning of DC Motor Using Swarm Intelligent Algorithm															
2	Literature Review															
3	Submission of Extended Proposal															
4	DC Motor Modelling															
5	Proposal Defense															
6	Reviewing PSO PID Toolbox															
7	PID Tuning of DC Motor Modelled															
8	Draft Interim															
9	Submission of Final Interim Report															

APPENDIX VII:

Activities/Gantt Chart and Milestone of FYP 2

No	Detail/Week	1	2	3	4	5	6	Mid Semester Break (04/07-08/07/2012)	7	8	9	10	11	12	13
1	Converting DC Motor Transfer Function into time domain using Inverse Laplace Transform	■	■												
2	Simulate PSO-PID program in C++		■	■											
3	Variation of parameters in PSO-PID Controller				■	■									
4	Variation of parameters in IWPSO-PID Controller						■		■						
5	Variation of parameters in PSO-PID Controller								■	■					
6	Progress Report									■					
7	Analyzing the simulation result										■	■			
8	Develop PSO-PID GUI											■	■		
9	Pre-SEDEX												■		
10	SEDEX													■	
11	Draft Final Report													■	
12	Submission of Final Report (Soft Cover)														■