

**MULTI-INPUT MULTI-OUTPUT PROPORTIONAL INTEGRAL  
DERIVATIVE CONTROLLER TUNING BASED ON IMPROVED  
PARTICLE SWARM OPTIMIZATION**

by

**THAMSANQA BONGANI NKWANYANA**

Submitted in accordance with the requirements of the degree of

**MAGISTER TECHNOLOGIAE**

in the subject

**Electrical Engineering**

at the

University of South Africa

Supervisor: Professor Zenghui Wang

January 2021

## DECLARATION

**Name:** Thamsanqa Bongani Nkwanyana

**Student number:** 49435248

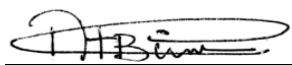
**Degree:** Magister Technologiae: Electrical Engineering

Exact wording of the title of the dissertation as appearing on the copies submitted for examination:

Multi-input multi-output proportional integral derivative controller tuning  
based on improved particle swarm optimization

I declare that the dissertation titled, “Multi-input multi-output proportional integral derivative controller tuning based on improved particle swarm optimization” is my own work. I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution. All the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

**Signature of student:**



---

**Date:**

January 2021

This dissertation is dedicated to all people who believe in education,  
and to my family and friends.

## **ACKNOWLEDGEMENTS**

The work introduced in this dissertation was completed under the supervision of Professor Zenghui Wang. I appreciate and am eternally grateful for his help, significant commitment, and direction throughout the research. His challenging questions, difficult inquiries and creative information profited the work. I will be forever grateful.

A very special thanks to Mr John Saveca for his emotional, technical, and financial assistance throughout my research. An exceptional note of gratitude to my mother for her help and guidance.

I would like to also pass the very special thanks to my family, friends and everyone who supported me directly or indirectly throughout my research. Finally, I owe my gratitude to Sthembile Nxumalo and Olwethu Nkwanyana for their encouragement and tolerance while I spent numerous hours stuck in front of the computer, I am grateful.

## ABSTRACT

The PID controller is regarded as a dependable and reliable controller for process industry systems. Many researchers have devoted time and attention to PID controller tuning and they all agree that PID controllers are very important for control systems. A PID equation is very sensitive; its parameters must always be varied following the specific application to increase performance, such as by increasing the system's responsiveness. PID controllers still have many problems despite their importance for control systems in industries. The problem of big overshoot on the conventional gain tuning is one of the serious problems. Researchers use the PSO algorithm to try and overcome those problems. The tuning of the MIMO PID controller based on the PSO algorithm shows many disadvantages such as high-quality control with a short settle time, steady-state error, and periodical step response. The traditional PSO algorithm is very sensitive and it sometimes affects the quality of good PID controller tuning.

This research has proposed a new equation for improving the PSO algorithm. The proposed algorithm is the combination of linearly decreasing inertia weight and chaotic inertia weight, after which a control factor was introduced as an exponential factor. This was very useful for simulations as it is adjustable. The Matlab simulation results of the experiments show that the new proposed equation converges faster and it gives the best fitness compared to linear inertia weight and oscillating inertia weight and other old equations. The MIMO PID controller system that consists of four plants was tuned based on the new proposed equation for the PSO algorithm (LCPSO). The optimized results show the best rise time, settling time, time delays, and steady-state compared to the systems that are tuned using the old equations. The exploration was directed at considering the impact of using the PSO calculation as an instrument for MIMO PID tuning. The results obtained in the examination reveal that the PSO tuning output improved reactions and can be applied to various system models in the measure control industry. The results for the MIMO PID controller tuned using PSO were assessed using integral square error (ISE), integral absolute error (IAE), and the integral of time expanded by absolute error (ITAE). The five well-known benchmark functions were also used to endorse the feasibility of the improved PSO and excellent results in terms of convergence and best fitness were attained.

**Keywords:** *particle swarm optimization, proportional-integral-derivative, local extreme, globally optimal, convergence, inertia weight, integral square error, integral absolute error.*

## TABLE OF CONTENTS

<u>DECLARATION</u> .....	iii
<u>DEDICATION</u> .....	iv
<u>ACKNOWLEDGEMENT</u> .....	v
<u>ABSTRACT</u> .....	vi
<u>TABLE OF CONTENTS</u> .....	vii
<u>ACRONYMSS AND ABBREVIATIONS</u> .....	viii
<u>LIST OF FIGURES</u> .....	ix
<u>LIST OF TABLES</u> .....	xi
<u>1. CHAPTER 1: INTORDUCTION</u> .....	1
1.2 Study Background .....	1
1.3 Problem Statement .....	2
1.3 Research Question .....	2
1.4 Benefits of the study .....	2
1.5 Objectives .....	3
1.6 Limitations and Hypothesis .....	3
1.7 Methodology and Reseach design .....	4
1.8 Dissertation overview .....	4
<u>2. CHAPTER 2: LITERATURE REVIEW</u> .....	5
2.1 PID control overview .....	5
2.1.1 Proportional Control .....	5
2.1.2 Integral Control .....	6
2.1.3 Derivative Control .....	8
2.1.4 Proportional Integral Derivative Controllers .....	8
2.2 PID Controller Tuning .....	9
2.2.1 Ziegler-Nichols and Cohen-Coon Methods .....	9
2.2.1.1 ZN Closed-Loop Tuning .....	10
2.2.1.2 ZN Open-Loop Tuning .....	10
2.2.2 Cohen-Coon Methods .....	11
2.2.3 Root Locus Methods .....	12
2.2.4 PSO Tuned PID control .....	12
2.3 PSO Overview .....	22

2.4 Conclusion.....	22
3. <u>CHAPTER 3: Particle Swarm Optimization Improvement</u> .....	24
3.1 Improved Particle Swarm Optimization.....	25
3.2 Methodology.....	27
3.3 Simulation .....	27
3.4 Results and Discussion .....	39
3.2 Conclusion.....	39
4. <u>CHAPTER 4: MIMO PID TUNING BASED ON PSO</u> .....	40
4.1 Particle Swarm Optimization .....	40
4.2 Tuning of MIMO PID based on LCPSO .....	42
4.3 Methodology.....	44
4.4 Simulation .....	45
4.5 Results and Discission .....	53
4.6 Conclusion.....	54
5 <u>CHAPTER 5: FUTURE WORK AND CONCLUSION</u> .....	55
5.1 Results and Discission .....	55
5.2 Conclusion.....	55
5.3 Future Work.....	56
Publications.....	56
<u>REFERENCES</u> .....	57
<u>APPENDICS</u> .....	60
<u>APPENDIX A</u> Matlab codes.....	60
<u>APPENDIX B</u> Codes.....	70
<u>APPENDIX C</u> Ethical Clearence.....	74

## LIST OF ABBREVIATIONS

AC	Alternating current
CT	Current transformer
CPSO	Comparative particle swarm optimization
DC	Direct current
DE	Differential evolution
EP	Evolutionary programming
GA	Genetic algorithm
GP	Genetic programming
IAE	Integral absolute error
ISE	Integral square error
ITAE	Integral of time multiplied by absolute error
LIF	Low impedance faults
MIMO	Multi input multi output
MOPSO	Multi-objective particle swarm optimization
NN	Neural network
PD	Proportional-derivative controller
PI	Proportional-integral controller
PID	Proportional-integral-derivative
PSO	Particle swarm optimization
QPSO	Quadratic particle swarm optimization
SI	Swarm intelligence
SIMO	Single-input multiple-output
SISO	Single-input single-output
TFA	Time-frequency analysis
ZN	Ziegler-Nichols



## LIST OF FIGURES

FIGURE 2.1: Closed-loop feedback	6
FIGURE 2.2: Varying control systems	6
FIGURE 2.3: Integral control	7
FIGURE 2.4: Varying integral action	7
FIGURE 2.5: Derivative control	8
FIGURE 2.6: PID controller	9
FIGURE 2.7: Demonstration of SISO and MIMO	15
FIGURE 2.8: Comparison of step response for PID controller	18
FIGURE 2.9: 1-1/2-2 controller pairing	21
FIGURE 2.10: 1-2/2-1 controller pairing	21
FIGURE 2.11: 3x3 controller pairing	22
FIGURE 3.1: Ackley best fitness	34
FIGURE 3.2: Booth best fitness	34
FIGURE 3.3: Cider best fitness	35
FIGURE 3.4: Cider best fitness	35
FIGURE 3.5: Schwefel best fitness	36
FIGURE 3.6: Sphere best fitness	36
FIGURE 3.7: Ackley function in 2 dimensions	37
FIGURE 3.8: Booth function in 2 dimensions	37
FIGURE 3.9: Cider function in 2 dimensions	38
FIGURE 3.10: Restrigin function in 2 dimensions	38
FIGURE 3.11: Schwefel function in 2 dimensions	39
FIGURE 4.1: PSO initialization	40
FIGURE 4.2: MIMO PID control system	42
FIGURE 4.3: Flowchart for PSO-PID system	43

FIGURE 4.4: Simulink model for MIMO system	46
FIGURE 4.5: Simulink model for PID 1	46
FIGURE 4.6: Simulink model for PID 2	47
FIGURE 4.7: System step response result for PID controller	52
FIGURE 4.8: System step response result	52
FIGURE 4.9: Showing MIMO PID tuning in progress	53
FIGURE 4.10: Output step response	53

## LIST OF TABLES

Table 2.1: Ziegler-Nichols closed-loop parameters	10
Table 2.2: Ziegler-Nichols open-loop parameters	11
Table 2.3: Cohen-Coon open-loop parameters	11
Table 2.4: Simulation results	14
Table 2.5: PID parameters	16
Table 2.6: Step response performance	16
Table 3.1: Comparison of simulation results for Ackley problem	28
Table 3.2: Comparison of simulation results for cigar problem	29
Table 3.3: Comparison of simulation results for Booth problem	30
Table 3.4: Comparison of simulation results Rastrigin problem	31
Table 3.5: Comparison of simulation results sphere problem	32
Table 3.6: Comparison of simulation results Schwefel problem	33
Table 4.1: Controller parameters for PID1	55
Table 4.2: Controller parameters for PID2	56
Table 4.3: Simulation results for PID controller and PID-LCPSO for output 1	56
Table 4.4: Simulation results for PID controller and PID-LCPSO for output 1	57
Table 4.5: Simulation results for PID controller and PID-LCPSO for output 2	57
Table 4.6: Simulation results for PID controller and PID-LCPSO for output 2	58

## **CHAPTER 1: INTRODUCTION**

### **1.1 STUDY BACKGROUND**

The improvement of the economy, the increasing of goods production, the increasing of goods distribution, the electricity production and distribution are every country's ideal achievement, therefore industrial good control system that can control multi-input multi-output system is essential in recent years as most industries need more than one system to run at the same time. The PID controller is seen as the most significant control arrangement in many industries and companies. Various modern systems are controlled using PID regulators and most organizations and firms need more than one system simultaneously. In such cases, MIMO PID control tuning is used. The greatest advantage of PID regulators is their extraordinary execution in a wide variety of working conditions, their useful straightforwardness, and their valuable ease. This enables engineers to work with them in a reasonably clear way. It also has a commonality for scientists and professionals working with the systems of control organizations and firms (Pillay and Govender, 2007). Notwithstanding its broad use, one of its weaknesses and essential deficiencies is that there is no viable tuning technique for this sort of regulator. Researchers have suggested a few PID tuning techniques for the tuning of PID controllers (Biswas et al., 2014).

The customary PID tuning procedures include Ziegler-Nichols and Cohen-Coon, of which the Ziegler-Nichols system may be the most notable strategy (Mallick & Khan, 2011). This tuning approach works splendidly. Nevertheless, it does not always give incredible tuning and all-in-all conveys a significant overshoot. Consequently, this methodology normally needs retuning before mechanical cycles. To refresh the limits of ordinary PID limit tuning systems, a couple of fair methods of reasoning have been recommended to improve the PID tuning, for example, those using a generic algorithm (GA), evolutionary programming (EP), and PSO. The PID regulator is seen as the dependable and strong regulator for industry systems (Pungot, 2015). Various algorithms have been applied to PID and various speculative theoretical and application results have been accomplished. To tune PID controllers, various strategies have been proposed, but there are still issues with using the MIMO PID controller since they give high motions and oscillations in step response reaction, particularly for the plants with solid nonlinear elements. Other than the traditional tuning strategies, there are some advanced tuning techniques that use astute methods.

The PID controller was first presented in the early 1900s (O'Dwyer, 2005). The history of controller improvements can be traced back to 1788 to the work of James Watt to supply his steam motor. In 1791, de Prony made improvements to the control system, as did Throop in 1857 and Maxwell in 1868. In 1911, Sperry started using the primary PID controller that was used as a ship control system (O'Dwyer, 2005). It is simple to understand PID control tuning, it gives good performance most of the time, and the characteristics allow engineers to operate in a straightforward manner. Single-input single-output (SISO) PID control is used in practice by many industries. They tune three parameters and achieve logical good performances that are well understood, and PID tuning rules are well known. However, due to the need to control many systems at the same time, MIMO PID systems were introduced. These systems have become the most frequent design for the MIMO PID system. The first approach for a MIMO PID is to use a single-input single-output PID controller for MIMO plants; it has been used for many years. These PID controllers can be tuned one at a time using traditional PID tuning rules and are one of the disadvantages (Dharan, et al., 2017). The multi-loop SISO PID is most frequently used for the plants that are not connected and in these contexts, it works well.

The SISO PID design is less complicated than the MIMO PID design, which is time-consuming. The other approach is to design one MIMO PID controller to control multi-input multi-output systems. With this method, it is possible to specify a simpler structure that uses all sensors. Tuning a MIMO PID controller requires three matrices or more. The number of inputs multiplied by the number of outputs. All this is a challenge and it is almost impossible to tune a MIMO PID controller by hand. Therefore, it is essential to develop a tuning method, hence the improvement of PSO and the tuning of the MIMO PID controller is pursued in this research.

PSO is one of the uncomplicated adaptive optimization algorithms. However, it also has its disadvantages, such as premature convergence, it is difficult to get the globally optimal solution and it easily falls into local extremes. Optimization was first implemented in the 1940s when the British military experienced issues with distributing few and limited assets for fighter airplanes, submarines, and other used hardware equipment (Kennedy et al., 2001). Researchers produced a diverse solutions for the issues over the decades. Various optimization strategies have been implemented for taking care of different issues in recent years. Non-conventional improvement techniques were introduced in recent years. They are also known as modern methods and are ground-breaking and famous for taking care of difficult issues. These

strategies incorporate genetic algorithms, neural systems, particle swarm improvement calculation, fuzzy optimization, and artificial immune systems. The PSO algorithm (PSO) is a population-based random pattern search algorithm. The PSO algorithm was introduced in 1995 by Kennedy and Eberhart and its fundamental conceptual idea was initially inspired by observing the social behavior of animals, for example, bird flocking, fish schooling, and other animals that behave in a similar manner (Shi and Eberhart, 2001). Although PSO has many advantages such as its simplicity and ability to optimize step responses, there are disadvantages, for example, premature convergence, and it easily falls into local extremes.

## **1.2 PROBLEM STATEMENT**

Despite the PID controller having many functions, including error calculation, speed regulation, control temperature, and pressure (Dharan, et al., 2017). PID controller has some drawbacks. In spite of PID controller being encountered in our everyday lives as a society, it still has a serious problem like big overshoot on the conventional gain tuning and oscillation of step response in the system, due to high-quality control with a short settle time, steady-state error, and a periodical step response (Taeib et al., 2013). Despite the use of the PSO algorithm to optimize PID controller tuning, the traditional PSO algorithm is very sensitive and it sometimes affects the quality of good PID controller tuning (Neto et al., 2017). PSO is having the disadvantage problem of falling into local extreme values, the issue of not obtaining a globally optimal solution, and the problem of finding the best fitness (Biswas et al., 2014).

This research is having two main objectives, the first is to improve particle swarm optimization (PSO) to minimize the above-mentioned drawbacks. The second objective is to tune MIMO PID systems using the proposed improved PSO to minimize PID controller mentioned disadvantages.

## **1.3 RESEARCH OBJECTIVES**

- The objective of the research is to improve the PSO algorithm in order to address its disadvantages, such as falling into a local extreme value, the issue of not obtaining a globally optimal solution and the problem of finding the best fitness.
- Another main objective of this research is to tune MIMO PID controller systems using the improved PSO to solve the problem of high oscillations in a step response characteristics.

## **1.4 RESEARCH QUESTIONS**

- How can one improve the particle swarm optimization to solve the problem of easily falling into local extreme values to reach a globally optimal solution and to get the best fitness?
- How can one minimize the difference between the specified required values and tuning outputs obtained values of rise time, settling time, and time delays of step response in a system when tuning a MIMO PID controller?

## **1.5 BENEFITS OF THE STUDY**

The study aims to improve the efficiency of a control system by tuning it well, minimizing the process variability, increasing efficiency, reducing energy costs, and maximizing production rates. The well-tuned MIMO PID controller based on PSO to control the MIMO system will increase production and distribution as many systems are tuned at the same time. Using best practices for controller tuning can help deliver value to the business quickly and accurately without guesswork. The controller will check the equipment, demonstrate the process dynamics, characterize the process needed, pick the right tuning, simulate the output results, and then monitor the results. Once all control systems work effectively, more electricity will be generated to meet the demands, more goods will be produced, and more minerals will be mined and transported, therefore an effective system means fast service delivery for society. Controller tuning refers to the selection of tuning parameters to ensure the best response from the controller so that it gives accurate results.

If we take the example of a car's cruise control, it uses PID to control the cruising system, and if a MIMO PID controller can be used in a car, the car will be able to control many functions at the same time. In South Africa we are faced with serious electricity production problems. ESKOM, the national electricity provider, is failing to produce enough power to supply the country. One of the main issues is a slow control system. An effective MIMO system will help to increase production to the point where we will have enough power. Big industries, nationally and internationally, are faced with the same issues with their control systems, and a well-tuned MIMO system can help to address most companies' issues. Other advantages of well-tuned MIMO systems include that PID is not based on a mathematical model of a system. It is also not heavily equipment dependent and is therefore simple to introduce, it can be well executed on modest equipment. A PID regulator, once planned, does not need a talented workforce to adjust. It is easier to put into practice (only with straightforward conditions), it uses fewer

assets, it is easier to tune by basic experimentation, and it has a better reaction to unmeasured unsettling influences. Model-based regulators recuperate from unmeasured aggravations with just one essential kind of activity, while PID has the corresponding and subsidiary activities that promptly follow up on an obscure unsettling influence.

## **1.6 LIMITATIONS AND HYPOTHESIS**

### **Limitations**

This research will be limited to...

- The study of PSO algorithm systems, background, advantages, and disadvantages.
- The study of PID controller's background, advantages and disadvantages.
- The Ziegler-Nichols method is briefly discussed for comparison
- The improvement of PSO based on the combination of inertia weights
- Comparing improved PSO with traditional PSO
- Tuning MIMO PID using improved PSO.
- The approval of best fitness and convergence by utilizing standard benchmark test function ( Ackley function, Rastrigin function, Schwefel function, Cigar work, sphere function, and the Booth function).
- The MIMO PID controller tuned using PSO assessed using integral square error (ISE), integral absolute error (IAE), and the integral of time expanded by absolute error (ITAE).
- The use of MATLAB programming language.

### **Hypothesis**

The efficiency of a control system, minimization of process variability, reducing energy costs, maximizing production rates and increasing distribution rate will be achieved with a well-tuned MIMO PID controller based on PSO. The unsteadiness of the voltage to the end-users impacts the capacity to the load not to be steady due to oscillations of step response of the system. The direct proportionality between the power and the voltage. Source instability causes harm, breakdown and wasteful activity to the load. In the event that the voltage is balanced, the capacity to the load will be steady and the load won't encounter breakdowns and will work effectively. This can be improved by limiting both voltage droops and oscillation of systems. The MIMO PID controller tuned based on PSO can help to improve speed, effectiveness and reliable convergence to the optimization problem.



## **1.7 METHODOLOGY AND RESEACH DESIGN**

This research consists of theoretical research and a software simulation/experimental research process based on a broad literature review. A quantitative method is used because of its focus on target estimations and the numerical examination of information accumulated through different trials and re-enactments or by using prior information of measurements using computational systems. Therefore, this research additionally includes sorting, investigating and correlating quantifiable information. Mathematical correlations of information are conducted and the outcomes are put together with respect to information examinations. The examination additionally includes hypothesis testing and scientific assessment dependent on information investigation through based on the outcomes of the data gathering procedures. The research problem is of specialized technical nature and accordingly falls into the science and design engineering field. This makes the quantitative examination method a reasonable strategy for this exploration. Within the quantitative method, a deductive methodology is used in light of its concern with creating speculation based on existing hypotheses and planning exploration methodologies to test the theory. This examination includes speculation dependent on actual physical laws and known realities. The simulations are used to affirm the above technique. Test recreations are dependent on objective fact of acquired outcomes in view of the perception of results, and the speculation of the theory is either affirmed or dismissed.

## **1.8 DISSERTATION OVERVIEW**

This dissertation has five chapters that all relate to the research objectives. The chapters give a point-by-point outline of the research problem and a literature review of all related topics that used to improve the methods to solve the problem.

### **Chapter 1: Introduction**

This chapter presents the study background, offers a problem statement, states the research question, objectives, study benefits, study limitations, and hypothesis, and discusses the research methodology of this research.

## **Chapter 2: Literature study**

This chapter offers the literature review to explore a PID controller. It also considers POS. Thereafter, the chapter details the history of PID and PSO, and give an overview of the Ziegler-Nichols method.

## **Chapter 3: Particle Swarm Optimization Improvement.**

Chapter 3 presents a new proposed equation for the improvement of the PSO algorithm. The proposed algorithm is based on the combination of linearly decreasing inertia weight and chaotic inertia weight, and the introduction of the control factor as an exponential. The following benchmark functions were used to endorse the feasibility of the improved PSO: the Ackley function, Rastrigin function, Schwefel function, Cigar work, sphere function, and the Booth function. Matlab is used to obtain the simulation results.

## **Chapter 4: MIMO Proportional-Integral-Derivative tuning based on PSO**

This chapter presents the MIMO PID controller tuned based on a newly proposed PSO algorithm. The MIMO PID controller tuned using PSO was assessed using integral square error (ISE), integral absolute error (IAE), and the integral of time expanded by absolute error (ITAE). The following pairing methods were used for research modeling: 1-1/2-2 controller pairing and 1-2/2-1 controller pairing and MATLAB tool is utilized for simulation.

## **Chapter 5: Conclusion and future work**

The study concludes with recommendations and conclusion. This last chapter gives a summary of study, recommendations for future work, and the conclusion of the research. The references used and the appendices are presented immediately after Chapter 5.

## **CHAPTER 2: LITERATURE REVIEW**

This chapter gives an overview of previous studies on PID controllers, Ziegler-Nichols and Cohen-Coon and PSO. It reviews several hypothetical subtleties related to PID controller tuning and its problems. The chapter further gives theories contributing to the proposed study by reviewing PSO improvement and its challenges. The chapter endeavours to detail each investigation and hypothesis to gain a better understanding of the research.

### **2.1 PID CONTROL OVERVIEW**

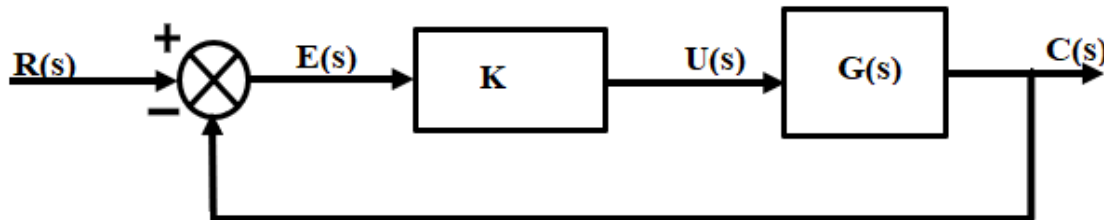
The PID regulator is seen as the most significant control arrangement in many industries and companies. Various modern systems are controlled using PID regulators and most organizations and firms need more than one system simultaneously. In such cases MIMO PID control tuning is used. The greatest advantage of PID regulators is their extraordinary execution in a wide variety of working conditions, their useful straightforwardness, and their valuable ease. This enables engineers to work with them in a reasonably clear way. It also has commonality for scientists and professionals working with the systems of control organizations and firms (Pillay and Govender, 2007). Notwithstanding its broad use, one of its principle weaknesses and essential deficiencies is that there is no viable tuning technique for this sort of regulator. Researchers have suggested a few PID tuning techniques for the tuning of PID controllers. The customary PID tuning procedures include Ziegler-Nichols and Cohen-Coon, of which the Ziegler-Nichols system may be the most notable strategy. This tuning approach works splendidly. Nevertheless, it does not always give incredible tuning and all-in-all conveys a significant overshoot. Consequently, this methodology normally needs retuning before mechanical cycles.

To refresh the limits of ordinary PID limit tuning systems, a couple of fair methods of reasoning have been recommended to improve the PID tuning, for example those using a generic algorithm (GA), evolutionary programming (EP), and PSO. The PID regulator is seen as the dependable and strong regulator for industry systems. Various algorithms have been applied to PID and various speculative theoretical and application results have been accomplished. To tune PID controllers, various strategies have been proposed, but there are still a few issues with using the MIMO PID controller since they give high motions and oscillations in step response reaction, particularly for the plants with solid nonlinear elements. Other than the traditional tuning strategies, there are some advanced tuning techniques that use astute methods. In this exploration the MIMO PID controller is tuned based on PSO. The principal point of the final

project was to improve the PSO streamlining to limit the issue of falling into local extreme values and not acquiring all global optimal solutions around ideal arrangement. In this research the improved PSO was used to tune MIMO PID systems to limit the oscillation of step response and to make a system increasingly proficient. The practical simulation was produced using Matlab and the outcomes are shown as graphs and tables. The following pairing methods were used: 1-1/2-2 and 1-2/2-1 controller pairing.

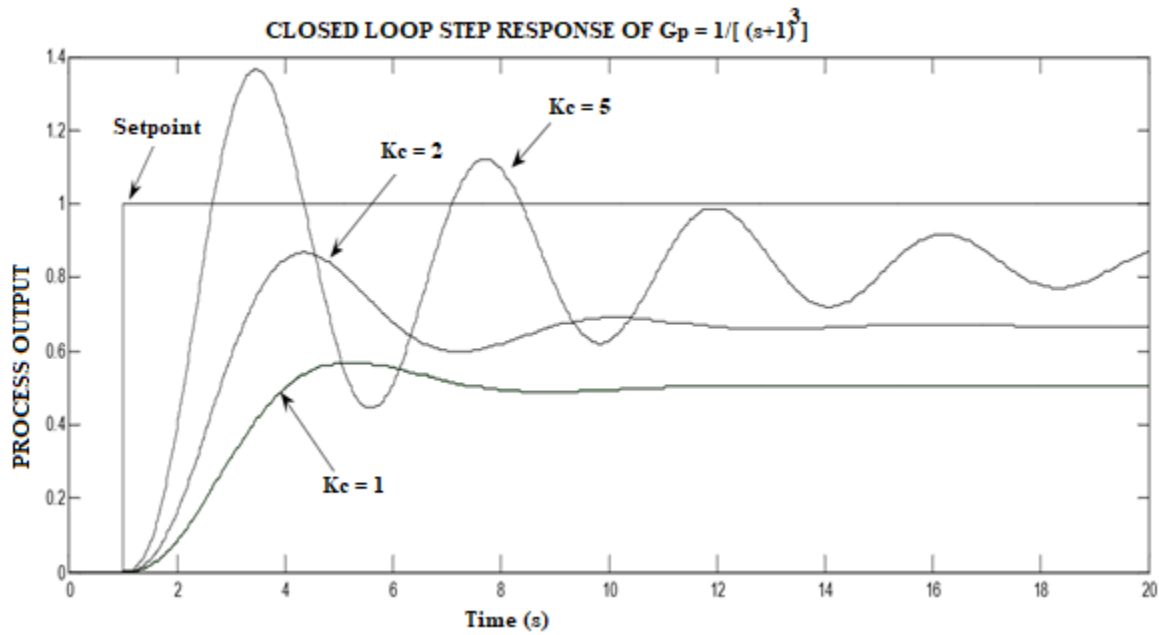
### 2.1.1 PROPORTIONAL CONTROL

The promotional controller or P-regulator is the most fundamental regulator. It is easy to use and simple to tune.



**Figure 2.1:** Closed-loop feedback

The shut circle move capacity of this control system has the additional capacity where  $G(s)$  is the exchange capacity of the procedure,  $R(s)$  and  $C(s)$  speak to the information and yield of the procedure individually and the blunder signal  $E(s)$ . The activity of the corresponding controller for the most part brings about a balance, for example the contrast between the ideal yield and the genuine yield of the system for forms that do not have any characteristic incorporating properties. Under these conditions the consistent state mistake for the control system can be determined using the last worth hypothesis. Relative control is regularly joined with vital control so as to dispense with balance while applying the littler estimations of the increase  $K$ . A run of the mill case of system reaction using just corresponding control is represented in Figure below (Pillay, 2008).



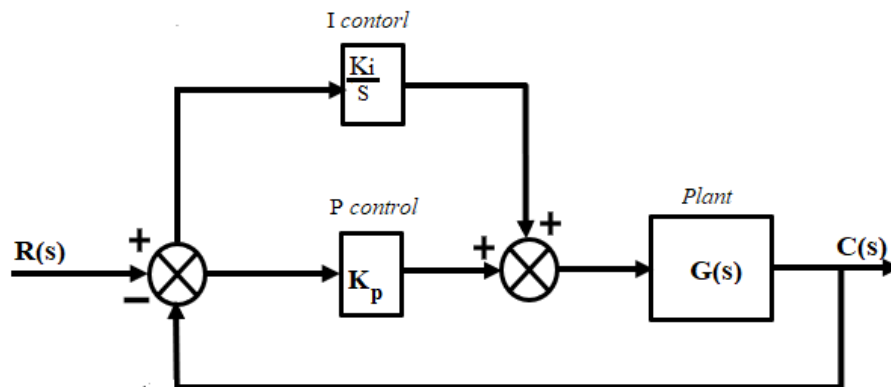
**Figure 2.2:** Varying control systems (Pillay, 2008)

### 2.1.2 INTEGRAL CONTROL

The deficiency of the corresponding regulator for a sort 0 system is that the consistent state blunder is not actually zero. This is remedied by using an ideal vital integral compensator. Since the essential yield will become ever bigger with even the smallest DC blunder, any fundamental increase will kill consistent state blunder. This single bit of leeway is the reason why PI (corresponding in addition to fundamental) control is frequently favoured over P just control. A compensator that uses unadulterated incorporation to improve the consistent state mistake is seen as an ideal basic compensator. The ideal compensator must be developed with dynamic parts, which on account of electric networks require the use of dynamic intensifiers and some of the time extra force sources. An inactive compensator is more affordable to actualize. For this situation the consistent state blunder is not heading for zero as in situations where ideal remuneration is used. Consistent state blunder can be eliminated essentially by adding an unadulterated coordination to the regulator or plant in a fell framework. This will obviously change the system type from a sort 0 to a sort 1. The difficulty that may arise is that adding this unadulterated incorporation will likewise change the transient reaction qualities of the system.

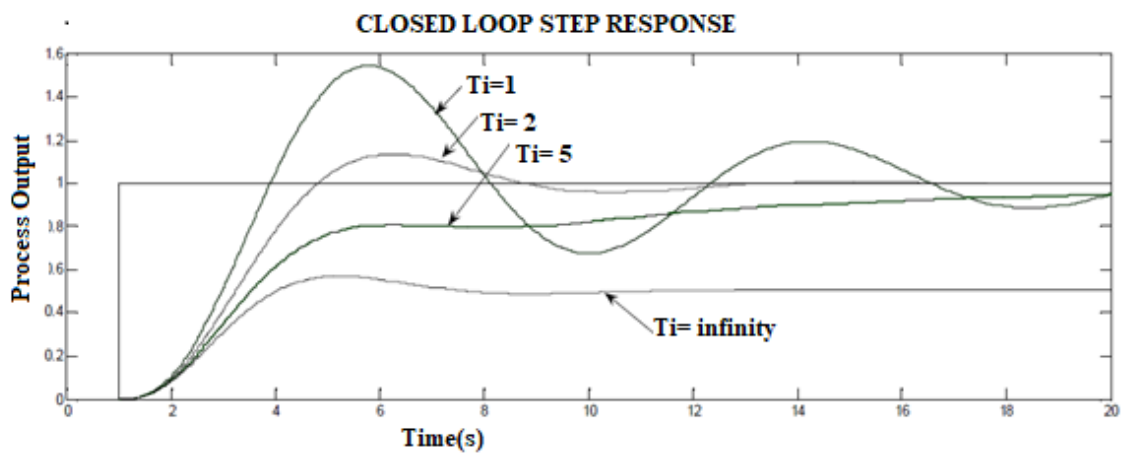
Corresponding activity becomes effective as an error not quite the same as zero occurs. In the event that the relative addition is adequately high, it will drive the error more towards zero.

Essential control achieves a similar control impact as relative control, but with a very high addition. This outcome in the balance disposing of property of indispensable activity, which can be delineated by applying the last worth hypothesis to the control structure of Figure 2.13.



**Figure 2.3:** Integral control

The control effects of the basic activity are shown in Figure 2.14. With respect to Figure 2.14, the corresponding addition is kept steady ( $K_c = 1$ ) and the necessary time is changed in accordance with the impact of the vital time consistency.

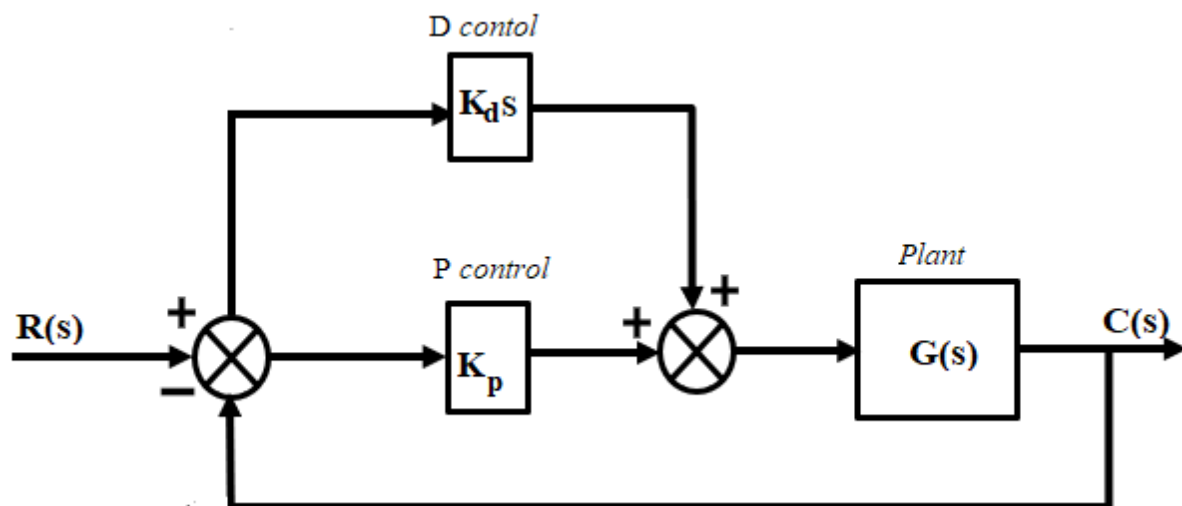


**Figure 2.4:** Varying integral action

The basic time ( $T_i$ ) steady is changed within the range  $T_i = [1, 2, 5, \infty]$ . This is the situation when  $T_i = \infty$  compares to unadulterated corresponding control and is indistinguishable from  $K=1$  in Figure 2.14, where the consistent state error is halved. The consistent state error is expelled when  $T_i$  has limited worth. For large estimations of the mix time steady, the reaction continuously moves towards the setpoint. For little estimations of  $T_i$ , the reaction is quicker, yet oscillatory.

### 2.1.3 DERIVATIVE CONTROL

On the off chance that a system has zero consistent state error, for example type 1 or more prominent, or an adequate level of consistent state error, the originator might have to improve the transient reaction of the framework. The plan objective here is to decrease settling time and accomplish an attractive percentage overshoot. This can be done by using ideal subsidiary remuneration. The term ideal alludes to the way an unadulterated separation is applied to the forward way. The ideal relative besides a subsidiary PD regulator uses dynamic segments in its acknowledgment, and the upsides and downsides of planning and assembling the system are like those of the past dynamic PI organization. The transient reaction of a framework can be picked by choosing the necessary shut circle shaft areas on the s-plane. In the event that these post areas are not effectively on the root locus of the framework, the framework root locus should be reshaped to incorporate these posts. One approach to achieve this is to add a zero to the forward transfer function (Youney, 2007). Figure 2.15 shows the PD controller.

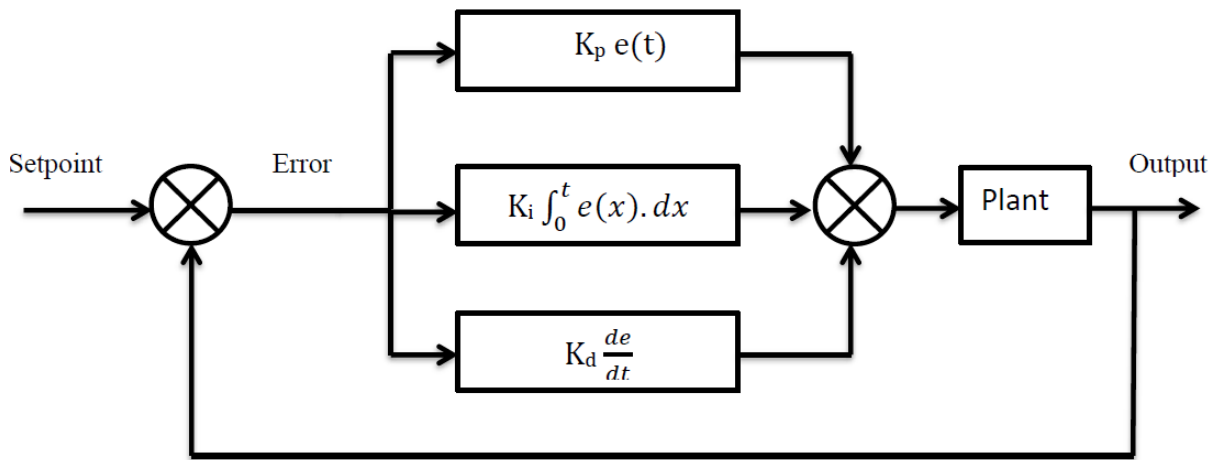


**Figure 2.5:** Derivative control

### 2.1.4 PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROLLERS

PID regulators are the most frequently used type of regulator for mechanical applications. They are basic and show vigorous execution over a wide range of working conditions. In the nonattendance of the total information on the cycle, these sorts of regulators are the most proficient with making decisions. The three primary boundaries included are proportional (P), integral (I) and derivative (D). The relative part is responsible for following the ideal set-point,

while the necessary and subsidiary part represents the amassing of past errors and the pace of progress of errors in the process separately. Figure 2.16 shows the PID controller.



**Figure 2.6:** PID controller(Mallick & Khan, 2011)

## 2.2 PID CONTROLLERS TUNING

Tuning a PID regulator means that its different boundaries and parameters are tuned to accomplish a streamlined estimation of the ideal reaction. The fundamental necessities of the output will be the wanted ascent time, top time and overshoot. Various processes have unique prerequisites for these boundaries, which can be accomplished by tuning the PID parameters. In the event that the system can be disconnected, the tuning technique includes an examination of the step input reaction of the framework to get diverse PID parameters. Yet, in a great number of the modern applications, the framework should be online, and tuning is accomplished physically. This requires experienced personnel and there is a consistent vulnerability because of human error. Another technique for tuning is the Ziegler-Nichols method. While this technique is useful for online counts, it includes some experimentation, which is not attractive (Mallick & Khan, 2011).

### 2.2.1 THE ZIEGLER-NICHOLS AND COHEN-COON METHODS FOR TUNING

One of the first tuning procedures was proposed by Ziegler and Nichols in 1942. They proposed the shut circle (or extreme affectability) strategy and the open-circle (or cycle response bend) technique. The ZN tuning rules have the disadvantage that it uses inadequate cycle data to decide the tuning boundaries (Pillay 2008). This causes system performances that are not robust (Åström & Häggglund, 2004). The Ziegler-Nichols tuning technique depends on the assurance



of cycles' intrinsic qualities, for example, the process gain ( $K_p$ ), measure time consistency ( $T_p$ ) and measure dead time ( $L_p$ ). These attributes are used to decide the regulator tuning parameters. Despite the fact that the Ziegler-Nichols strategies endeavour to yield ideal settings, the just basis expressed is that the reaction has a rot proportion of a quarter (Ziegler and Nichols, 1942).

$$G_c = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2.1)$$

$$K_p = K_p, K_I = \frac{K_p}{T_i}, \text{ and } K_D = \frac{K_p}{T_d} \quad (2.2)$$

### 2.2.1.1 ZN CLOSED-LOOP TUNING

This closed-loop tuning technique was introduced by Ziegler and Nichols around the same time as their open-loop strategy. It is sometimes referred to as the frequency reaction technique. This strategy is also founded on specific qualities of the cycle elements. Their plan of this strategy depends on knowledge of where the Nyquist bend of the process transfer function  $G(s)$  converges with the negative genuine pivot. The method portrays two parameters, Critical Gain  $K_u$  and Critical Period  $P_u$ , in view of this point, which Ziegler and Nichols named extreme increase and extreme period. The technique for deciding these boundaries is as follows: Associate the regulator with the plant, turn off the necessary control, for example set  $T_i = \infty$ , and turn off the subordinate control by setting  $T_d=0$ . Begin raising the addition  $K_c$  until the cycle begins to waver. The addition where this happens is  $K_u$  and the time of the motions will be  $P_u$ . Ziegler and Nichols devised straightforward recipes that relate  $K_u$  and  $P_u$  to  $K_c$ ,  $T_i$ , and  $T_d$  for a P, PI, and PID regulator (Youney, 2007) as shown in Table 2.1 below.

**Table 2.1:** Ziegler-Nichols closed-loop tuning parameter

Controller	$K_c$	$T_i$	$T_d$
P	$0.5 K_u$	$\infty$	0
PI	$0.4 K_u$	$0.8 P_u$	0
PID	$0.6 K_u$	$0.5 P_u$	$0.125 P_u$

### 2.2.1.2 ZN OPEN-LOOP TUNING

In 1942 Ziegler and Nichols inferred their first technique for PID tuning through exact testing. This strategy depends on the plant response to a stage input and is described by two parameters. The technique is regularly alluded to as the open-loop, or step response tuning strategy. The boundaries time constant  $T_p$  and delay time  $L_p$  are controlled by applying a unit step capacity to the cycle. This is a cycle with an integrator and a period delay, where  $b=a/L$ . Alluding to Figure 4.1, the point where the slant of the progression reaction is greatest is first decided, and the digression is drawn at that point. The convergence of this digression and the vertical pivot at  $T_d=0$  and  $T_i=\infty$  gives the boundaries  $T_p$  and  $L_p$ . Ziegler and Nichols inferred PID boundaries as P and PI as elements of  $T_p$  and  $L_p$ . The outcomes are given in Table 2.2 below.

**Table 2.2:** Ziegler-Nichols open-loop tuning parameter

Controller	$K_c$	$T_i$	$T_d$
P	$\frac{T_p}{L_p K_p}$	$\infty$	0
PI	$0.9 \frac{T_p}{L_p K_p}$	$3.33 L_p$	0
PID	$1.2 \frac{T_p}{L_p K_p}$	$2 L_p$	$0.5 L_p$

### 2.2.2 COHEN-COON METHOD FOR TUNING

The ZN strategy was intended for a process that cannot control itself. Offering a self-guideline, Cohen-Coon (CC) presented the self-guideline list or controllability proportion given in Table 2.3 below.

**Table 2.3:** Cohen-Coon tuning open-loop tuning parameter

Controller	$K_c$	$T_i$	$T_d$
P	$\frac{1}{K_p} \left[ 0.35 + \frac{1}{\varepsilon} \right]$	$\infty$	0
PI	$\frac{1}{K_p} \left[ 0.083 + \frac{0.9}{\varepsilon} \right]$	$\left[ \frac{3.3 + 0.31\varepsilon}{1 + 2.2\varepsilon} \right] T_p$	0
PID	$\frac{1}{K_p} \left[ 0.25 + \frac{1.35}{\varepsilon} \right]$	$\left[ \frac{2.5 + 0.46\varepsilon}{1 + 0.61\varepsilon} \right] T_p$	$\left[ \frac{3.7}{1 + 0.19\varepsilon} \right] T_p$

A great advantage of the open-circle technique is that it is faster and only requires a phase change to be applied to choose to separate the pertinent data for choosing the tuning boundaries.

The procedure does have disadvantages, namely:

- The "S-shaped" measure reaction curve and its appearance point are difficult to recognize when the assessment is loud, and
- A great proportion of bumble can be brought into the tuning figure if the reason for sound is not settled exactly (Lipták, 1995).

### 2.2.3 ROOT LOCUS METHODS

The exchange capacity of a PID controller is characterized as follows:

$$GPID(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s} \quad (2.3)$$

The control parameters are resolved by the accompanying strategy:

- Build up a lot of wanted transient and consistent state error particulars dependent on the application and the basic system type (number of open circle posts situated at the starting point).
- From these determinations, decide a conjugate pair of shut circle prevailing shafts and meet these details.
- Compute from the consistent state error detail.
- Assess the joined system move work at the predominant post and set it equivalent to zero to explain for the estimations of  $K_P$  and  $K_D$ .

$$1 + G_{PID}G(s_1) = 1 + \left( K_P + \frac{K_I}{s} + K_D s_1 \right) G(s_1) = 0 \quad (2.4)$$

$$K_P + K_D s_1 = -\frac{1}{G(s_1)} - \frac{K_I}{s_1} \quad (2.5)$$

A case of this structure strategy is introduced in the following segment. Common PID calculations that structure the structure squares of controllers have been addressed. The control activities of relative, fundamental and subsidiary terms and a portion of their unfriendly effects have also been explored. The relative controller gives a remedial activity that corresponds to the size of the blunder and affects the speed of a system's reaction; fundamental control gives restorative activity relative to the time necessary for the error is available for the whole term of the error; the subsidiary controller gives a remedial activity relative to the time subordinate of the blunder sign and reacts to the rate at which the blunder is evolving. The effects of procedure elements on controller tuning are visited in the following section.

#### 2.2.4 PSO TUNED PID CONTROL

Many researchers have devoted time and attention to PID controller tuning and they all agree that PID controllers are very important for control systems. These researchers have suggested many improvements, changes, and conclusions. PID has many functions, including error calculation, speed regulation, control temperature, and pressure. It is always attempting to reduce error over time by adjusting control variables. We encounter it every day in our lives as a society, one of the everyday examples is a car's cruise control. External influences like gradients can affect the car's speed and its velocity, but it restores the actual velocity to the required velocity in the best way without overshoot or delays. PID controllers are used in every place where there are control systems and power systems reasons. A PID equation is very

sensitive; its parameters must always be varied following the specific application to increase performance, such as by increasing the system's responsiveness (Anthony, 2014). PID controllers still have many problems despite its importance for control systems in industry. The problem of big overshoot on the conventional gain tuning is one of the serious problems. Researchers use the PSO algorithm to try and overcome that problem, and their results show the advantages of PID tuning based on PSO optimization. Due to high-quality control with a short settle time, steady-state error, and a periodical step response, the PSO algorithm can minimize the error between the actual output and the desired output. The traditional PSO algorithm is very sensitive and it sometimes affects the quality of good PID controller tuning. The following equation was presented from the first research:

Traditional PSO equation:

$$V_{id}(s + 1) = \omega V_{id}(s) + c_1 r_1 (P_{id}(s) - X_{id}(s)) + c_2 r_2 (P_{id}(s) - X_{id}(s)) \quad (2.6)$$

$$X_{id}(s + 1) = X_{id}(k) + V_{id}(k + 1) \quad (2.7)$$

The following is an improved velocity equation:

$$V_{id}(s + 1) = \mathcal{X}(V_{id}(s) + r_1 c_1 (P_{id}(s) - X_{id}(s)) + r_2 c_2 (P_{gd}(s) - X_{id}(s))) \quad (2.8)$$

where construction coefficient  $\mathcal{X}$  is expressed as:

$$\mathcal{X} = \frac{2}{|2 - l - \sqrt{(l^2 - 4l)}|} \quad (2.9)$$

and the fitness function is:

$$fitness = \sum_{j=1}^n \frac{1}{1 + \theta_j(k)} \quad (2.10)$$

Researchers suggested chaos particle swarm optimization (CPSO) to overcome the problem output (Adel & Abdelkader, 2013). Other research addressed the issue of overshooting when conventional gain tuning was used. These researchers also used PSO for tuning MIMO PID controller systems. Their result was positive and they had better results for PSO-based tuning. Particles update their positions and velocities to obtain great fitness values after any iteration. These equations below were used:

$$V_{pd}^{t+1} = \omega V_{pd}^t + c1r1(pb_{best}^t - X_{pd}^t) + c2r2(g_{best}^t - X_{pd}^t) \quad (2.11)$$

$$X_{pd}^{t+1} = X_{pd}^t + V_{pd}^{t+1} \quad (2.12)$$

where t is number of iterations, pbest is the particle, gbest is the group, r1 and r2 are random numbers with interval 0,1,  $\omega$  is the inertia weight and c1 and c2 are positive constants. The following table presents the simulation results, showing overshoot, rise time and setting time.

**Table 2.4: Simulation results (Taeib, Moez & Chaari, 2013)**

Tuning Method	Overshoot(%)	Rise time	Setting time
Z-N PID	64.8174	0.1352	3.2941
PSO-PID1 (ISE)	40:8062	0.0929	0.7970
PSO-PID2 (IAE)	42:3434	0.0884	0.7775
PSO-PID3 (ITSE)	48:5050	0.0389	0.5265

The Z-N tuning showed high tuning, high rise time and high settling time compared to PSO tuning (Taeib, et al, 2013). The design of PID systems using PSO was proposed, a simulation was conducted, and then conclusions were made that PID controllers tuned based on PSO is more efficient and it offers high level of automation (Biswas et al., 2014). Other researchers redo the PID controller for robotic manipulator design with the aim to improve the systems. They also use PSO for better result. The integral time-weighted absolute error (ITAE) is given by:

$$ITAE_{\psi} = \int_{t_0}^{t_f} t |\psi(t)| dt. \quad (2.13)$$

Djaneye-Boundjou et al. (2016) tackled PID control tuning problems by using a previously developed stable PSO. Other researchers propose an efficient approach for tuning controllers for MIMO systems. The criticism channel gives execution that compromises quick shut circle elements and control signal reach. Suitable tuning of the channel's parameters is essential to accomplish ideal execution. MIMO systems tuning requires multi-target execution. In their

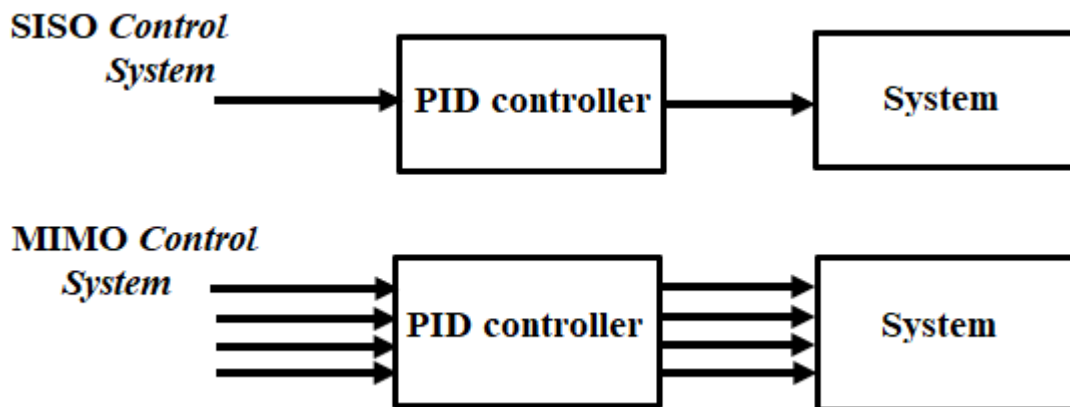
exploration, researchers propose a fluffy-based feedback channel configuration tuned with MOPSO to eliminate these bottlenecks. The MOPSO approach was approved using a twin rotor MIMO system. A multi-objective PSO is a transformative heuristic that duplicates the social conduct of fledgling amassing, fish tutoring, and other gathering creatures. PSO starts populace of particles haphazardly in space with molecules, speaking to a possible arrangement. Every molecule has a bunch of boundaries and moves haphazardly in a multi-dimensional space looking for ideal arrangement. The speed of every molecule in space plays a great part in focusing on the best up-and-comer arrangement. Likewise, speed and position changes for molecules depend on the encounters picked up from their own speed, area, and neighbouring particles. Systems are assessed using a profoundly nonlinear unequalled system with emphatically coupled elements. The power of the regulator was inspected by forcing vulnerabilities in the TRMS boundaries. They are used in cycles whose element models can be portrayed as first or second-request systems. The transfer function  $G_c(s)$  of single loop PID controllers has the following form:

$$G_c(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_I}{s} + K_D s \quad (2.14)$$

TRMS was picked to assess the exhibition of versatile control since it has a place with the class of unequalled systems with exceptionally forceful model nonlinearity and coupled elements (Hashim, et al., 2017). Other researchers propose a new control scheme of PSO for tuning a PID controller. The characteristics of MIMO TRMS are high order non-linearity, significant cross-coupling and inaccessibility for its states and output for measurements. The PSO algorithm was successfully implemented to address this problem. The simulated results of the developed PID controller for a twin rotor system demonstrate its effectiveness. Satisfactory results were anticipated in the experimental as well as in the simulation results, proving that a PID controller based on PSO performs better than the other conventional controllers.

The optimal values obtained from simulations were applied in the experiment with the twin rotor MIMO system. The results of the experiment describe the performance indices of system response, adopting each of the controllers in terms of percentage overshoot, settling time, rise time, and steady state errors, along with each controller's gain values. The MIMO system's problem characteristics are high order non-linearity, significant cross-coupling and inaccessibility of some of its states and output for measurements. The proposed PSO-tuned controller shows better performance criteria compared to the Ziegler-Nichols tuned controller

(El-Sayed, Ahmed & Mohamed, 2009). The Ziegler- Nichols technique (ZN) is the most familiar tuning strategy, introduced by Ziegler and Nichols (ZN) in 1942. They proposed the close-loop strategy and the open-loop strategy. The ZN tuning rules has the weakness of using inadequate procedure data to decide the tuning parameter. This hindrance prompts system exhibitions that have performance with poor robustness. The Ziegler-Nichols tuning technique depends on the assurance of a procedure's inborn attributes, for example, the procedure pick up ( $p K$ ), process time consistent ( $p T$ ) and process dead time ( $p L$ ). These attributes are used to decide the controller tuning parameters. In spite of the fact that the Ziegler-Nichols techniques endeavour to yield ideal settings, the main concern expressed is that the reaction has a rot proportion of quarter. This is seen as a deficiency on the grounds that a controller tuned with this standard may not be at its ideal setting. Other researcher show that the application of the MIMO systems strategy to the heater arrangement of a PID benchmark issue exhibits the viability of the outlined technique, where the plan method is completed with the state space implementations, which is advantageous particularly in managing information dealing with (MIMO) systems (Ochi, & Yokoyama, 2012).



**Figure 2.7:** Demonstration of SISO and MIMO

Yamada and Hagiwarn redesigned and modified the PID controller with the aim to get a close loop system for their PID controller. They show examples to illustrate the effectiveness of the proposed method (Hagiwara & Yamada, 2008). Other researchers aimed to control quad rotor attitude. The PSO algorithm is presented using an enhanced stochastic variation system strategy to upgrade the effectiveness of the fuzzy PID controller. The simulation results suggest that the proposed controller truly gives the best execution in sparing the settling time, dependability, strength, and less wavering is achieved (Chiou et al., 2016).



$$K(s) = K_p + \frac{K_I}{s} + K_D s \quad (2.15)$$

$$K_p = \begin{bmatrix} k_{p11} & \dots & k_{p1n1} \\ k_{pn01} & \dots & k_{pn0ni} \end{bmatrix} \quad (2.16)$$

A PID controller  $K(s)$  is expected to accomplish the the following: 1. The apparent shut circle system is asymptotically consistent, and 2. The energetic constancy execution satisfies the going with dissimilarity  $g1 \leq 1$ . There are various execution or execution measures for creating or outlining controllers, for example, the basic of incomparable bungle (IAE), the essential of squared-botch (ISE), or central of time-weighted-squared-bumble (ITSE). Problems with the IAE and ISE models include that they may achieve a reaction with reasonably little overshoot and have a long settling time since they measure all errors and check all bungles reliably as time goes on. Regardless of the way in which the ITSE execution premise can beat this burden, it cannot guarantee steadfastness. The IAE, ISE and ITSE execution norms are depicted as follows:

$$IAE = \int_0^{\infty} |r(t) - y(t)| dt = \int_0^{\infty} |e(t)| dt \quad (2.17)$$

$$ISE = \int_0^{\infty} e^2(t) dt \quad (2.18)$$

$$ITSE = \int_0^{\infty} t e^2(t) dt \quad (2.19)$$

Zamani and Karimi (2009) propose two cost capacities to outline a MIMO system. The execution of the proposed strategy is greatly improved according to their results (Zamani & Karimi, 2009). They evaluated the performance of a composite control system with PSO and HDE, using an integral square error (ISE), integral absolute error (IAE) and integral of time multiplied by absolute error (ITAE). Their results show that PSO is the best compared with the Ziegler-Nichols methods. An examination of the PSO-based PID (PSO-PID) execution and the ZN-PID was used in their explorative research. The outcomes demonstrate that the PID tuning based on PSO is an improved approach (Solihin, Tack & Kean, 2011). PID is the most favoured controller for many reasons, including its straightforward algorithm, capacity to adjust to an extensive variety of uses where it can guarantee brilliant control exhibitions.

PSO requires crude scientific and mathematical administrators, which makes it worthwhile in terms of accessibility to bigger memory and higher speed. It has effectively been connected to a wide assortment of issues, for example neural systems, auxiliary advancement, shared

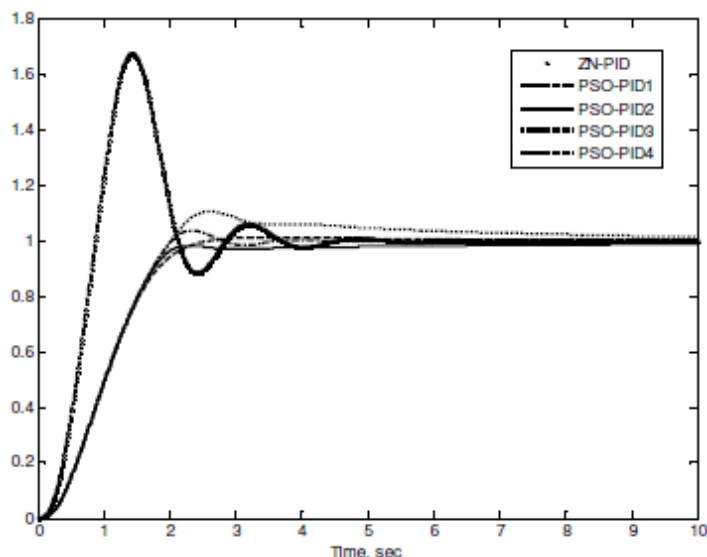
topology improvement, and fluffy systems. In this case the analyst had to enhance the PSO and the expectation was to expand social weight while latency weight diminishes. For their situation, little social weight influenced the universal best position to have a minor effect on the speed refreshing. Toward the finish of the run, the extensive social weight guarantees the best particles data influence. The simulation results are shown in the tables and graph below.

**Table 2.5: PID parameters (Kaya, 2014)**

<b>Tuning Method</b>	$K_p$	$K_i$	$K_d$
Z-N PID	133.2	176.9	18.9
PSO-PID1 (ISE)	97.6	15.6	11.9
PSO-PID2 (IAE)	73.9	11.0	15.5
PSO-PID3 (IAE)	95.0	24.5	12.8
PSO-PID3 (ITSE)	36.2	15.8	11.2

**Table 2.6: Step response performance for PID controller (Kaya, 2014)**

<b>Tuning Method</b>	<b>Overshoot(%)</b>	<b>Settling time</b>	<b>SSE</b>
Z-N PID	68	5.2	82.3
PSO-PID1 (ISE)	4.8	3.6	58.0
PSO-PID2 (IAE)	0	5.9	87.3
PSO-PID3 (ITSE)	11.8	7.6	88.9
PSO-PID4 (ITAE)	1.5	2.7	87.4



**FIGURE 2.8: Comparison of step response for PID controller (Kaya, 2014)**

Simulation results proved that PSO tuning is better than Z-N (Kaya, 2014). PSO has parameters that affect its best performance, some of them less of an effect and some have more of an effect. The parameters include the swarm size/number of particles, velocity components, number of iterations, neighbourhood size, inertia weight, acceleration coefficients, velocity clamping, and velocity constriction. There are parameters in PSO estimation that may affect its execution. For any given improvement issue, some of the parameter's characteristics and choices will have an impact on the capability of the PSO methodology, and distinctive parameters have next to zero effect. The fundamental PSO parameters incorporate the take after swarm size or number of particles, speed segments, number of emphases, neighbourhoods measure, latency weight and increasing speed coefficients. Likewise, PSO is additionally affected by inertia weight, speed cinching, and speed narrowing and these parameters are examined in this exploration. This examination offers the general flowchart for the PSO method where the accompanying advances are checked and adjusted if there is a need to do so. Particles have arbitrary place, speed, and wellness esteem. The momentum seeking point is set to the individual best. The best-assessed estimation of individual best is set to worldwide best and the operator number with the best esteem will be put away. With the calculation and assessment of wellness work, the wellness work is ascertained. In the event that the esteem is superior to the current individual best esteem, the individual best will be supplanted by the present esteem (Pillay, 2008).

Neighbourhood topologies that get a legitimate exchange between investigation and misuse is a critical and productive technique that is broadly used as part of numerous calculations for every particle. Less association happens when the areas in the swarm are few. The smaller the neighbourhood, the slower the joining will be, in spite of the fact that it might enhance the nature of arrangements. For bigger neighbourhoods, the meeting will be quicker, however, there is a chance that occasional joining happens beforehand. The answer for this issue will be that the pursuit procedure will begin with smaller neighbourhood sizes and after that the estimate for smaller neighbourhoods is expanded over the long run. The worldwide best PSO is where the situation of each particle is influenced by the best-fit particle in the swarm. The individual best position has a similarity with the situation in space where the particle had the most diminutive motivator as controlled by the objective work of a minimization issue. In addition, the position yielding the most decreased motivator among all the individual bests is known as the overall best position.

The nearby best particle swarm advancement technique is affected by checking its neighbourhood for every particle. The nearby learning in nature is shown by social data traded inside the area of the particle. For this situation, the speed of the particle will be computed. The quantity of particles in the swarm essentially influences the run-time of the calculation. In this way a harmony between the assortment of more particles and the speed of fewer particles must be found. Another essential factor in the joining rate of the calculation is the most extreme speed parameter ( $\max V$ ). This parameter confines the most extreme bounce that a particle can have in one stage, subsequently an expansive incentive for this parameter will bring about motion. However, a small esteem could cause the particle to be caught inside neighbourhood minima. Issues of swarm size is tended to in this examination (Pillay, 2008). The number of cycles to get a conventional result is furthermore issue subordinate. A too low number of cycles may stop the hunt methodology carelessly, while excessively generous emphases have the consequence of unnecessary included computational multifaceted nature and a need for extra time. A need to adjust various emphases is vital for this exploration.

Speed segments are essential for refreshing the particle's speed. There are three terms for the particle's speed under the given conditions. This segment is an energy that avoids huge alterations in the course of the particles. It is predispositioned towards the present increasing speed coefficients. The increasing speed coefficients together with the arbitrary qualities keep up the stochastic impact of the psychological and social segments of the particle's speed

individually. The study shows how much certainty a particle has in itself, while demonstrating how much certainty a particle has in its neighbourhood. The inactivity weight supplants by modifying the impact of the past speeds. For instance, it controls the energy of the particle by measuring the commitment of past speed, and latency weight will be increased by the speed at the past time step, therefore in the worldwide best PSO the speed condition of the particle with idleness weight will change. >>>Speed clipping encourages particles to remain inside the limit and to make sensibly stride estimate keeping in mind the end goal to go over the inquiry space. Narrowing Coefficient, this development coefficient present another parameter known as the choking factor. The narrowing coefficient was created by Clerc. it is vital in control the investigation and misuse exchange off, to guarantee meeting conduct, and furthermore to reject the idleness weight and the most extreme speed Correlation of enhanced PSO with the customary PSO, after all enhancements then the enhanced PSO will be contrasted with the customary PSO. The Benchmarks capacity will be utilized for Comparison ( Pillay, 2008). The GA is a usually utilized developmental calculation and has been chosen for correlation with the PSO in this examination. Particle swarm enhancement is like the Genetic Algorithm. The Genetic Algorithm has been well known in the enterprises on account of its capacity y to comprehends, simplicity of usage and its capacity to fathom profoundly non-direct, blended whole number enhancement issues that are run of the mill of complex designing systems ( Kachitvichyanukul, 2012). Other researchers in their research focus particle Swarm Optimization (PSO) based calculation is proposed for the advancement optimization of a PID controller. It is discovered that the system having the controller that is planned utilizing PSO method is stronger and shows better unsettling influence dismissal contrasted with the other procedure (Vincent & Nersisson, 2017). In 1900 numerous instrument organizations created programmed controller for the control procedure and for assembling businesses incorporates metals, mash, paper, stumble, control age, synthetic substances, refining and numerous more organizations. In 1907 C.J. Tagliabue Co did the principal establishment of the pneumatic programmed on-off systemtemperature and electrical controller in New York, in 1925-1935, 75000 programmed controllers was sold in the USA. They were just three kinds of programmed controller in early long stretches of improvement in particular: electrical hand-off with solenoid worked valve-on-off, electrical transfer with engine worked valve-drifting (vital) control, Pneumatic hand-off with a stomach valve (P activity 1% - 5% PB). In 1920 Morris E. Leeds got a first patent for a pneumatic PI controller. The primary genuine PID-type controller in Control Engineering was produced by Elmer Sperry in 1911. The principal hypothetical investigation of a PID controller was distributed by Nicolas Minorsky in 1922, and it was

additionally connected to the programmed directing of boats. In 1935 S.D. Miteroff he was first to give the time area conditions of controllers and described them as P, PI, PD and so on in 1934 Albert Callander distributed an inward Imperial Chemical Industries in England report called "Starter notes in programmed control". In 1936 and 1937 paper, Callander and co-creators proposed graphs to permit tuning of PI and PID

controllers for a scope of procedures with a deferral. In 1942 Ziegler and Nichols presented experimentation tuning strategy, this technique is the outstanding and the most generally utilized strategy for tuning of PID controllers is otherwise called consistent cycling or extreme pick-up tuning technique (O'Dwyer, 2005). In 1953 Cohen-Coon presented tuning decides that is second in popularity.it was eleven years after Ziegler and Nichols distributed was distributed when Cohen and Coon distributed their tuning method. Cohen-Coon tuning systems are suited to more broad extent of procedures. The Ziegler-Nichols rules function admirably just on forms where the dead time is not as much as a large portion of the length of the time consistent. In 1959 Bailey Meter presented the main strong state electronic controller. In 1964 Taylor Instruments shows presented first single-circle advanced controller. In 1969 Honeywell presents their Vutronik procedure controller line with subsidiary activity ascertained from the negative of the procedure variable as opposed to specifically from the mistake. The main bundled computerize PI and PID was presented in 1976 by Rochester Instrument Systems presents Media. Throughout the years more strategies have been presented, including Tyreus-Luyben technique, damped wavering strategy, C-H-R strategy, Fertik technique, IMC technique, Minimum mistake criteria Method, AMIGO Method, Lambda Tuning Method and Internal Model Control Method, their methodology are like Ziegler– Nichol's technique, yet the last controller settings are distinctive for a few techniques. At that point in 1992 Ciancone and Marline technique was created, this strategy enabled utilization of charts to fulfill the control objective. Numerous analysts center around various info different yield MIMO PID control systems Because more procedures are multi-input multi-output (MIMO) systems which require multi-input multi-output (MIMO) PID control methods to enhance their execution. As we probably are aware, MIMO PID controller configuration has created over various years. In 1986, Luyben proposed a straightforward tuning technique for PID controllers in MIMO systems. (O'Dwyer, 2005).

The systems can be paired using different way, in the below figures they is a 1-1/2-2 controller pairing and 1-2/2-1 controller pairing

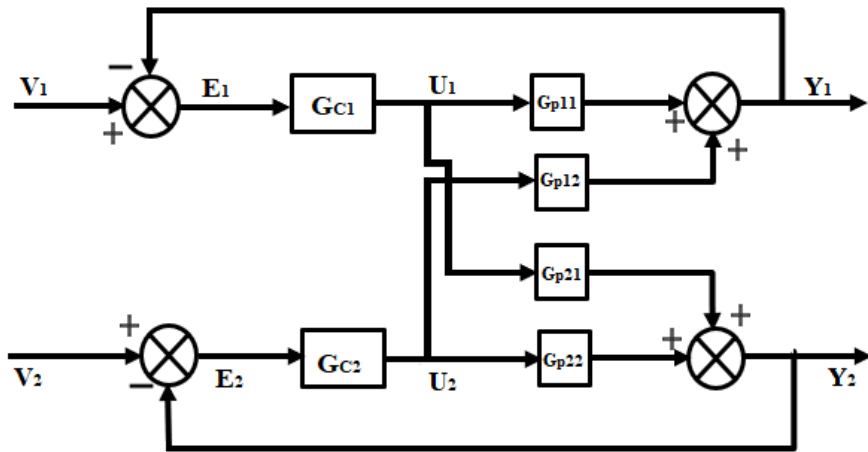


FIGURE 2.9: 1-1/2-2 controller pairing (<https://chemengr.ucb.edu/ch18-1-25-05>)

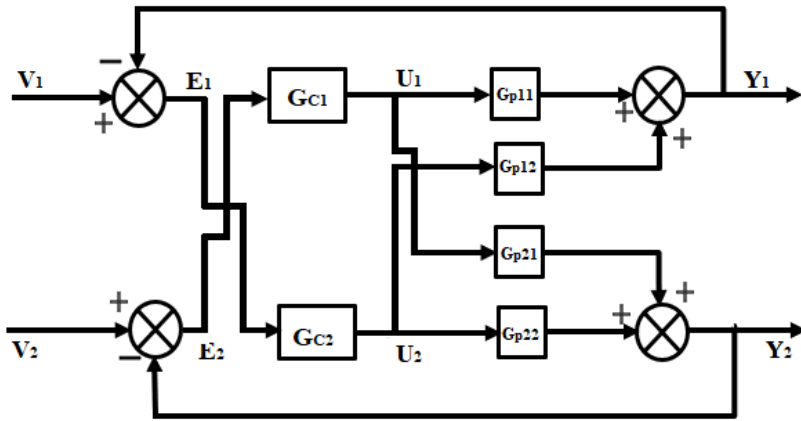


FIGURE 2.10: 1-2/2-1 controller pairing (<https://chemengr.ucb.edu/ch18-1-25-05>)

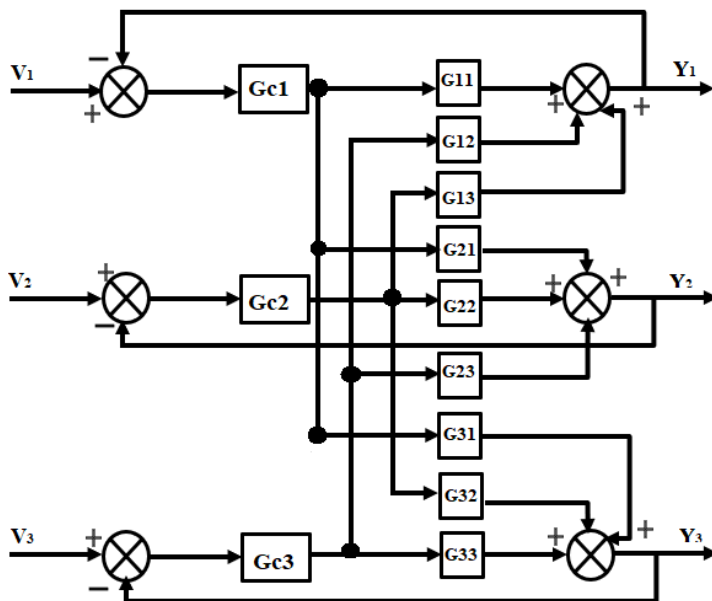


FIGURE 2.11: 3x3 controller pairing (Devikumari & Vijayan, 2015)

### 2.3 PSO OVERVIEW

The PSO algorithm uses the populace-based stochastic improvement calculation that was first discovered by Eberhart and Kennedy (1995). The social conduct of birds rushing for the same destination is what propelled the discovery. Reynolds mimicked the flying creatures for visual PC recreation purposes, seeing that the group gives off an impression of being under focal control. Reynolds continued to demonstrate this using three straightforward guidelines, in particular crash shirking, speed coordinating and herd cantering. Using these standards Reynolds indicated how the conduct of every specialist inside the group can be displayed with basic vectors. This trademark is one of the essential ideas of PSO. Boyd and Recharson (1985) inspected the dynamic procedure of people and built up the idea of individual learning and culture transmission. As per their assessment, individuals use two significant sorts of data in dynamic procedures, in particular their own understanding and other individuals' encounters. The first entails that they have attempted the decisions and realized which state has been exceptional up until now, and they realize how great it was. The second entails that they know about how different operators around them have performed. At the end of the day, they know which decisions their neighbours have experienced as positive up until now and how positive the best example of the decision was. Every specialist's choices depend on his own understanding and others' arrangement. This brand name is another fundamental thought of PSO. Eberhart and Kennedy (1995) joined these musings, which resulted in the improvement of their PSO methodology and their clear speed and position computations that mimic standard multitude direct. In PSO, a great deal of self-created experts insert in the structure space towards the ideal plan over different cycles. Each administrator has a memory of its best position and the multitude's best plan. PSO resembles EC systems. It very well may be said that the two strategies are people-based and each individual is evaluated by a foreordained health work. The huge difference is that PSO is affected by the propagation of social lead rather than normal choice (Shi and Eberhart, 2001). The pseudo-code for confining particle speed is as follows (Kennedy et al., 2001):

$$\text{If } v_{k+1} > V_{max} \quad \text{then } v_{k+1} = V_{max} \quad (2.20)$$

$$\text{Else if } v_{k+1} < -V_{max} \quad \text{then } v_{k+1} = -V_{max} \quad (2.21)$$



The best methodology for the remainder of the flock is to discover the nourishment is following Bird An and looking through its neighbouring region. The speed demonstrates the headings of the considerable number of particles in the following cycle. The nearby most popular position is the best arrangement that has been accomplished by every particle up until this point. The worldwide most popular position is the best arrangement among all the accomplished arrangements. The idleness speed part, nearby most popular position part, and worldwide most popular position and some portion of the speed mirror the participation and rivalry instrument in PSO. The speeding up factors handle the progression sizes of the particles in the following cycle. On the off chance that the quickening factors are excessively few, the particles might not have enough speed to arrive at the objective areas. On the off chance that the speeding up factors are too large, the particles may fly over the ideal worth. The fitting choice of increasing speed elements could abstain from catching into nearby insignificant neighbourhood and lessen the calculation time. The PSO calculation created by Kennedy and Eberhart was motivated by a reproduction of the unpredictable flight examples of a group of birds. Their underlying reproductions developed into a straightforward streamlining calculation that shows complex conduct. The PSO calculation comprises of an assortment of specialists, alluded to as particles, where every particle speaks to an applicant answer for the present enhancement issue. Every particle holds three snippets of data, to be specific its ebb and flow position, ebb and flow speed, and the best position it includes discovered inside the inquiry space.

## **2.4 CONCLUSION**

This chapter has briefly given an overview and definition and explained the difficulties experienced with tuning a PID controller. It also delineated various strategies used to tune a PID. The section additionally laid out research recently done on PSO and the Ziegler-Nichols technique. It additionally featured segments that are critical to obtain PSO improvement and the advancement of PID tuning. The next section apply theories, contributions, speculations and commitment studied to obtain improved PSO.

## CHAPTER 3

### 3. PARTICLE SWARM OPTIMIZATION IMPROVEMENT

In this chapter, a new PSO equation is proposed by combining two types of inertia weights. To find a solution for above-mentioned disadvantages, the linear decreasing inertia weight is combined with the chaotic inertia weight. The control factor is introduced as an exponential function. This research focuses on trying to minimize that negative influence on optimization strategies. An improved solution, namely the linearly chaotic particle swarm optimization (LCPSO) has been proposed to improve the speed of premature convergence. The improved equation is responding positively by getting the globally optimal solutions and fixing the problem of falling into local extremes. Combining the linear decreasing inertia weight and chaotic inertia weight has produced the best results as the two-inertia weight has difference properties have proven to be the best inertia weight strategy.

The newly proposed equation was tested by solving the six well-known benchmark function problems and offering recommended solutions. The PSO optimization strategy is not dependent on any gradient data or angle data of the problem to be solved or the work to be enhance and optimized. It uses simple mathematical calculations and is less expensive, quicker, more efficient and progressively productive compared to other optimization methods. Moreover, there are fewer parameters to modify and adjust in PSO. That is the reason that the PSO algorithm is widely used to solve optimization problems. The particle swarm optimization equations are shown below.

$$V_{id} = V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{id} - X_{id}) \quad (3.1)$$

$$X_{id} = X_{id} + V_{id} \quad (3.2)$$

Inertia weight is one of the very significant parameters for PSO and it plays a vital role during the procedures of providing balanced best fitness. The first PSO algorithm that was discovered by Eberhart and Kennedy in 1995 has no inertia weight. Then in 1998 Shi and Eberhart introduced PSO with inertia weight. Presently there are many different types of inertia weight concepts that have been introduced over the years, including chaotic inertia weight, random inertia weight strategy and linearly decreasing strategy, but the first inertia weight concept to be introduced was constant inertia weight.

The linearly decreasing inertia weight [3] is

$$w_l = w_{max} - \frac{w_{max} - w_{min}}{I_{max}} * t \quad (3.3)$$

The chaotic inertia weight [4] is

$$w_c = (w_{max} - w_{min}) \frac{I_{max} - I}{I_{max}} + w_{min} z \quad (3.4)$$

$$z = 4z(1 - z) \quad (3.5)$$

If the inertia weight is introduced in (1), the equation becomes:

$$V_{id} = wV_{id} + c_1r_1(P_{id} - X_{id}) + c_2r_2(P_{id} - X_{id}) \quad (3.6)$$

### 3.1 IMPROVED PARTICLE SWARM OPTIMIZATION

The improvement of PSO in this research is based on inertia weight. There are many types of inertia weight. In this research two of these inertia weights are combined and the control factor  $\phi$  is introduced as an exponential function. Linearly decreasing inertia weight and chaotic inertia weight were chosen for the proposed new equation to improve PSO. According to past research the above inertia weight strategies are counted as two of the best in terms of less or no errors and best fitness solutions. The linear decreasing inertia weight can improve the convergence speed and chaotic inertia weight can enhance the ability to jump out of the local optima. The experiments, comparisons and conclusions of previous researchers state that chaotic inertia weight is the best strategy for better accuracy and the linear inertia weight strategy is best for better efficiency.

#### Linear Decreasing Inertia Weight

$$w_l = w_{max} - \frac{w_{max} - w_{min}}{I_{max}} * t \quad (3.7)$$

#### Chaotic inertia weight

$$w_c = (w_{max} - w_{min}) \frac{I_{max} - I}{I_{max}} + w_{min} z \quad (3.8)$$

(8) z is logistic mapping

## Improvement of inertia weight

In the basic PSO, control parameters have a great impact on optimizing tuning systems, but if not executed properly the particles cannot converge and they become unstable and as a result the optimal solution of optimization problems cannot be obtained. The control system parameters are mostly chosen based on the researcher's proven theory and practical and experience or experiments from engineers. The linearly chaotic inertia weight strategy is introduced on this research to improve the parameters and improve the ability of PSO to give the best fitness with less errors. Two combined inertia weight are chosen for this research based on the outcomes of past writers. They state that linearly and chaotic are counted as the best inertia weight so far and this proposed equation is not difficult to understand.

The first step is to combine two inertia weights:

$$w = w_c + w_l \quad (3.9)$$

$$w = \left[ w_{max} - \frac{w_{max} - w_{min}}{I_{max}} t \right] + \left[ (w_{max} - w_{min}) \frac{I_{max} - I}{I_{max}} + w_{min} Z \right] \quad (3.10)$$

The next step is finding a common lowest denominator for each equation:

$$w = \left[ \frac{I_{max} * w_{max} - (w_{max} - w_{min}) t}{I_{max}} + \frac{(w_{max} - w_{min}) * I_{max} - I + I_{max} * w_{min} Z}{I_{max}} \right] \quad (3.11)$$

The next step is finding the lowest common denominator for the whole equation:

$$w = \frac{I_{max} * w_{max} - (w_{max} - w_{min}) t + (w_{max} - w_{min}) * I_{max} - I + I_{max} * w_{min} Z}{I_{max}} \quad (3.12)$$

The control factor is  $\phi$

The next step is to introduce a control factor that will help to control the output fitness by varying it into the suitable values. The control factor symbol is  $\phi$  for this research. The control factor is added as the exponential function, therefore:

$w$  becomes  $w^\phi$  and for the purpose of shortening the equation we let  $w_{max}$  be equal to  $w_1$  and  $w_{min}$  be equal to  $w_2$

Then the equation (3.12) becomes the equation (3.13):

$$w^\phi = \left[ \frac{I_{max} * w_1 - (w_1 - w_2)t + (w_1 - w_2) * I_{max} - I + I_{max} * w_2 Z}{I_{max}} \right]^\phi \quad (3.13)$$

When simplifying the equation (3.13), one comes to:

$$w^\phi = \left[ \frac{I_{max}(2w_1 - w_2 Z) - (w_1 - w_2)t - I(w_1 - w_2)}{I_{max}} \right]^\phi \quad (3.14)$$

Then the equation (3.1) becomes the equation (3.15):

$$V_{id} = w^\phi V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{id} - X_{id}) \quad (3.15)$$

The proposed new equation was used in simulations on Matlab to check the fitness, then conclusion and recommendations are made based on the results.

The proposed equation is named as follows: LCPSO

### 3.2 METHODOLOGY

The simulation for this research was taken using the MATLAB tool, they were repeated 30 times for each equation. They are two traditional/ old equation use in this study and one new proposed equation. Results are shown in table 3.1 to table 3.6. The following steps were followed for simulation to test the new proposed equation of modified inertia weight using benchmark functions: Ackley, Rastrigin, Schwefel, Cigar, Rosenbrock, Sphere and Booth.

The Proposed equation and the old traditional equation were tested using the following PSO parameters:  $c_1 = c_2 = 1.9 - 2.2$  (learner factor),  $W_{max} = 1$ ,  $W_{min} = 0$ ,  $I_{max} = 50$ ,  $N=2$  (number of dimensions),  $M=50$  (number of particles). The varying parameters:  $X_{max} = [30; 30]$ ,  $X_{min} = [-30; -30]$  (boundary).

The results of all the above-mentioned benchmark functions during the test for the new equation of inertia weight were then compared with the results that were obtained on the old equations. Analytical data and time intricate were determined.

### 3.3 SIMULATION

The Table 3.1 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.1: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving the Ackley benchmark as a problem**

Problem	Method	Best	Mean	Worst
<i>Ackley Benchmark function</i>	<b>New Proposed PSO (LCPSO)</b>	$2.811e^{-10}$	$1.349e^{-09}$	$6.590e^{-08}$
		$2.248e^{-10}$	$1.959e^{-09}$	$4.384e^{-08}$
		$1.102e^{-10}$	$2.517e^{-09}$	$5.732e^{-08}$
		$4.093e^{-10}$	$4.790e^{-09}$	$1.804e^{-08}$
		$3.404e^{-11}$	$9.552e^{-09}$	$1.039e^{-09}$
		$4.666e^{-11}$	$6.009e^{-09}$	$2.344e^{-09}$
		$1.201e^{-11}$	$1.435e^{-09}$	$6.460e^{-09}$
		$1.477e^{-11}$	$4.228e^{-09}$	$2.234e^{-09}$
		$1.022e^{-11}$	$2.121e^{-09}$	$8.345e^{-09}$
		$2.500e^{-11}$	$9.651e^{-09}$	$2.443e^{-09}$
		$2.444e^{-10}$	$8.987e^{-09}$	$4.898e^{-09}$
		$2.226e^{-11}$	$7.004e^{-09}$	$4.223e^{-09}$
		$1.479e^{-11}$	$7.785e^{-09}$	$1.099e^{-09}$
		$2.641e^{-10}$	$6.002e^{-09}$	$1.343e^{-09}$
		$4.763e^{-10}$	$1.333e^{-09}$	$1.559e^{-09}$
		$9.326e^{-10}$	$0.224e^{-09}$	$3.347e^{-09}$
		$8.443e^{-10}$	$0.094e^{-09}$	$7.560e^{-09}$
		$1.435e^{-10}$	$1.764e^{-09}$	$3.323e^{-08}$
		$4.596e^{-10}$	$1.904e^{-09}$	$6.654e^{-08}$
		$4.783e^{-10}$	$4.731e^{-09}$	$5.089e^{-08}$
$2.308e^{-10}$	$4.550e^{-09}$	$9.003e^{-08}$		
$1.393e^{-10}$	$3.667e^{-09}$	$3.332e^{-08}$		
<i>Ackley Benchmark function</i>	<b>Traditional PSO</b>	0.00064	0.00092	0.0265
		0.00224	0.00735	0.0841

The Table 3.2 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.2: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving Cigar benchmark as a problem.**

Problem	Method	Best	Mean	Worst
<i>Cigar Benchmark function</i>	<b>New Proposed PSO (LCPSO)</b>	$4.307e^{-17}$	$2.702e^{-15}$	$2.004e^{-14}$
		$2.684e^{-17}$	$1.379e^{-15}$	$2.004e^{-14}$
		$2.666e^{-16}$	$2.841e^{-15}$	$4.380e^{-14}$
		$4.378e^{-17}$	$1.168e^{-15}$	$1.333e^{-14}$
		$1.434e^{-17}$	$2.227e^{-15}$	$1.162e^{-14}$
		$3.456e^{-17}$	$2.002e^{-15}$	$3.336e^{-13}$
		$2.235e^{-17}$	$4.890e^{-15}$	$6.400e^{-13}$
		$1.444e^{-16}$	$3.443e^{-15}$	$2.259e^{-14}$
		$0.045e^{-16}$	$0.247e^{-15}$	$9.900e^{-14}$
		$8.566e^{-16}$	$4.909e^{-15}$	$3.433e^{-13}$
		$4.455e^{-17}$	$1.230e^{-15}$	$5.339e^{-13}$
		$3.327e^{-16}$	$1.556e^{-15}$	$5.688e^{-13}$
		$2.407e^{-18}$	$2.568e^{-15}$	$4.988e^{-13}$
		$4.561e^{-17}$	$2.337e^{-15}$	$2.515e^{-13}$
		$6.572e^{-16}$	$3.234e^{-15}$	$4.009e^{-14}$
		$0.566e^{-16}$	$0.707e^{-15}$	$3.557e^{-14}$
		$8.773e^{-17}$	$2.434e^{-15}$	$6.089e^{-14}$
		$2.446e^{-16}$	$1.200e^{-15}$	$0.004e^{-13}$
		$4.006e^{-16}$	$2.340e^{-15}$	$8.400e^{-13}$
		$9.753e^{-17}$	$8.967e^{-15}$	$7.886e^{-14}$
$3.008e^{-20}$	$5.256e^{-15}$	$0.193e^{-13}$		
$1.333e^{-18}$	$9.028e^{-15}$	$3.202e^{-13}$		
<i>Cigar Benchmark function</i>	<b>Traditional PSO</b>	0.00549	0.00823	0.0834
		0.00694	0.00847	0.096

The Table 3.3 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.3: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving Booth benchmark as a problem**

Problem	Method	Best	Mean	Worst
<i>Booth Benchmark function</i>	<b>New Proposed PSO (LCPSO)</b>	$2.533e^{-14}$	$1.126e^{-11}$	$1.500e^{-08}$
		$4.218e^{-14}$	$3.381e^{-11}$	$3.004e^{-08}$
		$4.983e^{-14}$	$3.313e^{-11}$	$5.232e^{-08}$
		$6.723e^{-14}$	$2.941e^{-10}$	$1.890e^{-08}$
		$1.485e^{-14}$	$1.693e^{-11}$	$1.045e^{-09}$
		$7.006e^{-14}$	$0.608e^{-10}$	$2.367e^{-09}$
		$1.221e^{-14}$	$0.565e^{-11}$	$6.422e^{-09}$
		$5.212e^{-14}$	$0.259e^{-11}$	$2.376e^{-09}$
		$2.908e^{-14}$	$5.231e^{-11}$	$8.389e^{-09}$
		$2.340e^{-14}$	$9.571e^{-11}$	$2.465e^{-09}$
		$2.349e^{-14}$	$4.207e^{-11}$	$4.890e^{-09}$
		$9.657e^{-14}$	$4.067e^{-11}$	$4.244e^{-09}$
		$1.479e^{-14}$	$3.115e^{-10}$	$1.007e^{-09}$
		$9.608e^{-14}$	$1.902e^{-11}$	$1.380e^{-09}$
		$4.003e^{-14}$	$1.399e^{-11}$	$1.993e^{-09}$
		$8.378e^{-14}$	$1.203e^{-11}$	$3.395e^{-09}$
		$8.123e^{-14}$	$1.033e^{-11}$	$7.588e^{-09}$
		$8.421e^{-14}$	$5.214e^{-11}$	$3.900e^{-08}$
		$4.533e^{-14}$	$5.004e^{-11}$	$6.602e^{-08}$
		$4.234e^{-14}$	$5.790e^{-11}$	$5.077e^{-08}$
$7.708e^{-14}$	$5.440e^{-10}$	$9.074e^{-08}$		
$9.309e^{-14}$	$4.680e^{-10}$	$3.376e^{-08}$		
<i>Booth Benchmark function</i>	<b>Traditional PSO</b>	0.00059	0.00829	0,0543
		0.00014	0.00243	0.03430



The Table 3.4 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.4: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving Rastrigin benchmark as a problem.**

<b>Problem</b>	<b>Method</b>	<b>Best</b>	<b>Mean</b>	<b>Worst</b>
<i><b>Rastrigin Benchmark function</b></i>	<b>New Proposed PSO (LCPSO)</b>	$3.535e^{-14}$	$3.986e^{-12}$	$2.240e^{-11}$
		$5.319e^{-14}$	$1.466e^{-12}$	$2.365e^{-11}$
		$1.421e^{-14}$	$1.016e^{-13}$	$2.388e^{-11}$
		$7.105e^{-14}$	$5.187e^{-11}$	$2.786e^{-10}$
		$2.487e^{-14}$	$7.673e^{-11}$	$1.214e^{-11}$
		$7.055e^{-14}$	$3.469e^{-11}$	$0.897e^{-10}$
		$1.921e^{-14}$	$3.521e^{-13}$	$0.099e^{-11}$
		$1.913e^{-14}$	$4.334e^{-13}$	$0.200e^{-11}$
		$1.806e^{-14}$	$5.900e^{-13}$	$1.988e^{-11}$
		$4.747e^{-14}$	$6.889e^{-11}$	$1.890e^{-11}$
		$4.147e^{-14}$	$7.254e^{-11}$	$1.266e^{-11}$
		$4.234e^{-14}$	$3.211e^{-11}$	$9.325e^{-11}$
		$9.575e^{-14}$	$3.190e^{-12}$	$9.111e^{-10}$
		$9.056e^{-14}$	$2.578e^{-12}$	$8.657e^{-11}$
		$0.945e^{-14}$	$2.457e^{-11}$	$1.990e^{-11}$
		$8.711e^{-14}$	$1.278e^{-11}$	$8.290e^{-11}$
		$8.212e^{-14}$	$4.907e^{-11}$	$7.077e^{-11}$
		$9.089e^{-14}$	$7.167e^{-12}$	$5.657e^{-11}$
		$9.090e^{-14}$	$8.122e^{-12}$	$5.213e^{-11}$
		$1.177e^{-14}$	$7.435e^{-12}$	$6.435e^{-11}$
$1.432e^{-14}$	$6.564e^{-12}$	$6.466e^{-10}$		
$1.267e^{-14}$	$1.554e^{-12}$	$3.006e^{-10}$		
<i><b>Rastrigin Benchmark function</b></i>	<b>Traditional PSO</b>	0.00349	0.00735	0,0471
		0.00447	0.00794	0.0822

The Table 3.5 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.5: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving Sphere benchmark as a problem.**

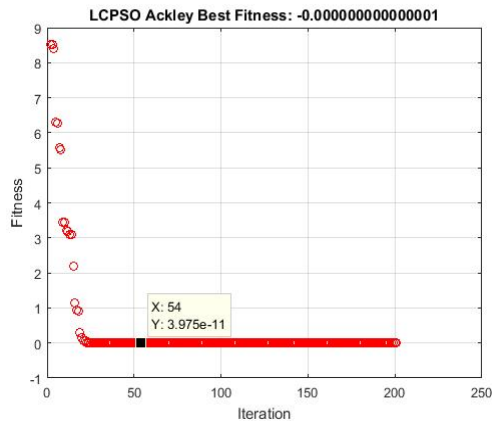
Problem	Method	Best	Mean	Worst
<i>Sphere Benchmark function</i>	<b>New Proposed PSO (LCPSO)</b>	$3.027e^{-20}$	$3.735e^{-17}$	$3.903e^{-15}$
		$4.406e^{-20}$	$2.356e^{-17}$	$1.205e^{-15}$
		$1.317e^{-20}$	$4.573e^{-17}$	$2.003e^{-15}$
		$2.011e^{-20}$	$2.444e^{-17}$	$9.233e^{-15}$
		$3.404e^{-20}$	$5.744e^{-17}$	$3.772e^{-15}$
		$4.666e^{-20}$	$6.554e^{-17}$	$2.366e^{-15}$
		$1.201e^{-19}$	$9.645e^{-17}$	$6.430e^{-15}$
		$1.477e^{-20}$	$8.778e^{-17}$	$2.489e^{-15}$
		$1.022e^{-20}$	$4.321e^{-17}$	$8.980e^{-15}$
		$2.500e^{-20}$	$5.311e^{-17}$	$2.433e^{-16}$
		$2.444e^{-20}$	$5.527e^{-17}$	$4.668e^{-15}$
		$2.226e^{-20}$	$5.574e^{-17}$	$4.683e^{-15}$
		$1.479e^{-20}$	$6.055e^{-17}$	$1.098e^{-15}$
		$2.641e^{-20}$	$2.432e^{-18}$	$1.113e^{-15}$
		$4.763e^{-20}$	$2.273e^{-17}$	$1.019e^{-15}$
		$9.326e^{-20}$	$1.034e^{-17}$	$3.777e^{-15}$
		$8.443e^{-20}$	$3.334e^{-17}$	$7.889e^{-15}$
		$1.435e^{-20}$	$6.734e^{-17}$	$3.804e^{-15}$
		$4.596e^{-20}$	$2.334e^{-17}$	$6.922e^{-15}$
		$4.783e^{-20}$	$5.711e^{-17}$	$5.676e^{-15}$
$2.308e^{-20}$	$3.930e^{-17}$	$9.173e^{-17}$		
$1.393e^{-20}$	$6.637e^{-18}$	$3.562e^{-15}$		
<i>Sphere Benchmark function</i>	<b>Traditional PSO</b>	0.00029	0.0005	0.047
		0.00014	0.03430	0.554

The Table 3.6 shows the Matlab simulation results of best fitness for the new proposed PSO equation and the results of the old traditional PSO. The comparison shows that the new proposed equation is having the best results for best fitness. The ideal fitness is approximate to zero.

**Table 3.6: Comparison between results for the new proposed PSO equation and the results for traditional PSO equation, solving Schwefel benchmark as a problem.**

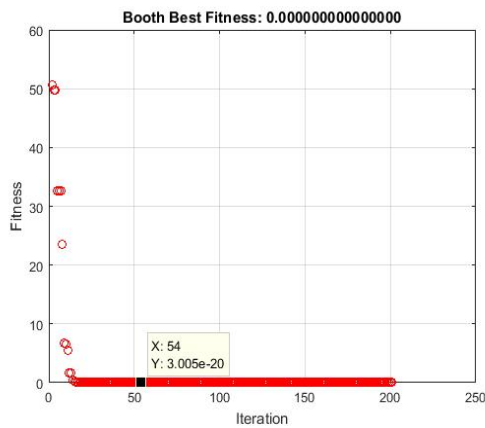
<b>Problem</b>	<b>Method</b>	<b>Best</b>	<b>Mean</b>	<b>Worst</b>
<i>Schweffel Benchmark function</i>	<b>New Proposed PSO (LCPSO)</b>	$4.188e^{-10}$	$2.022e^{-09}$	$0.879e^{-08}$
		$7.223e^{-10}$	$2.976e^{-09}$	$0.097e^{-08}$
		$3.325e^{-10}$	$2.457e^{-09}$	$4.576e^{-08}$
		$2.009e^{-10}$	$4.332e^{-09}$	$4.845e^{-08}$
		$1.877e^{-11}$	$4.369e^{-09}$	$1.009e^{-09}$
		$4.439e^{-11}$	$6.088e^{-09}$	$1.223e^{-09}$
		$2.280e^{-11}$	$6.670e^{-09}$	$6.998e^{-09}$
		$2.477e^{-11}$	$6.007e^{-09}$	$6.290e^{-09}$
		$2.787e^{-11}$	$2.789e^{-09}$	$6.513e^{-09}$
		$2.008e^{-11}$	$0.424e^{-09}$	$0.089e^{-09}$
		$2.213e^{-10}$	$0.900e^{-09}$	$0.501e^{-09}$
		$2.235e^{-11}$	$1.187e^{-09}$	$0.651e^{-09}$
		$4.909e^{-11}$	$7.541e^{-09}$	$8.078e^{-09}$
		$4.320e^{-10}$	$7.087e^{-09}$	$9.457e^{-09}$
		$5.554e^{-10}$	$7.343e^{-09}$	$9.511e^{-09}$
		$9.543e^{-10}$	$0.679e^{-09}$	$5.320e^{-09}$
		$9.898e^{-10}$	$0.127e^{-09}$	$4.871e^{-09}$
		$9.990e^{-10}$	$0.763e^{-09}$	$4.300e^{-08}$
		$0.346e^{-10}$	$0.332e^{-09}$	$6.124e^{-08}$
		$4.345e^{-10}$	$1.788e^{-09}$	$5.0453e^{-08}$
$0.212e^{-10}$	$1.436e^{-09}$	$3.990e^{-08}$		
$5.345e^{-10}$	$1.655e^{-09}$	$3.256e^{-08}$		
<i>Schweffel Benchmark function</i>	<b>Traditional PSO</b>	0.00149	0.00735	0.0339
		0.000294	0.00847	0.5572

The figure 3.1 shows the best fitness for new proposed equation solving the Ackley problem, it shows very good solution.



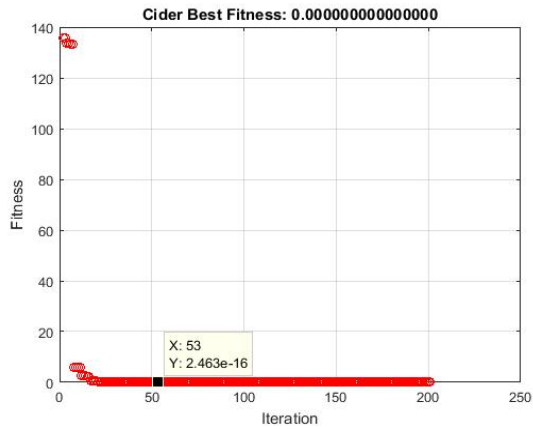
**Figure 3.1: Ackley best fitness**

The figure 3.2 shows the best fitness for new proposed equation solving the Booth problem, it shows very good solution.



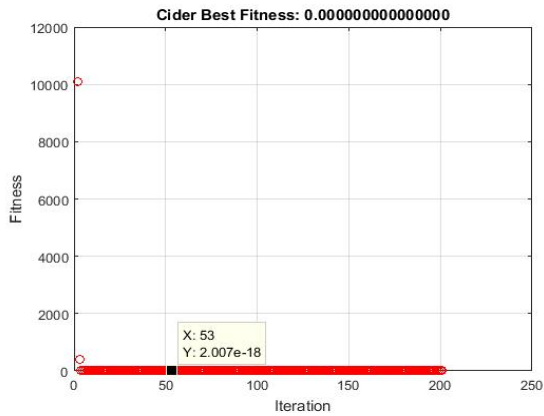
**Figure 3.2: Booth best fitness**

The figure 3.3 shows the best fitness for new proposed equation solving the Cider problem, it shows very good solution.



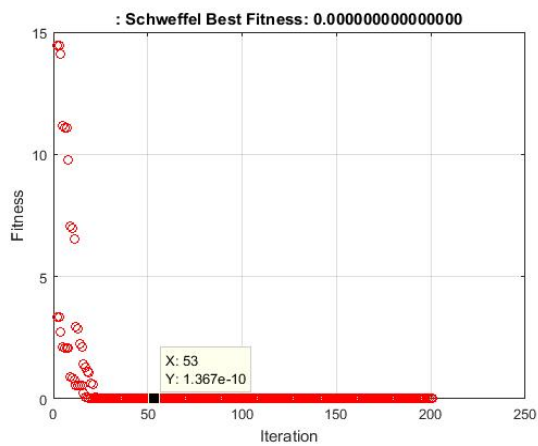
**Figure 3.3: Cider best fitness**

The figure 3.4 shows the best fitness for new proposed equation solving the Cider problem, it shows very good solution.



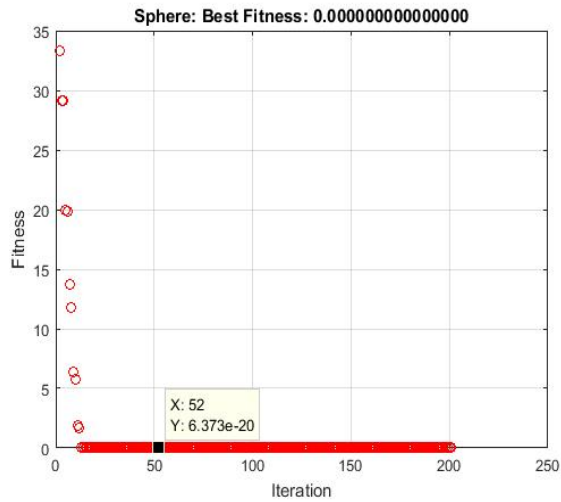
**Figure 3.4: Cider best fitness**

The figure 3.5 shows the best fitness for new proposed equation solving the Schwefell problem, it shows very good solution.



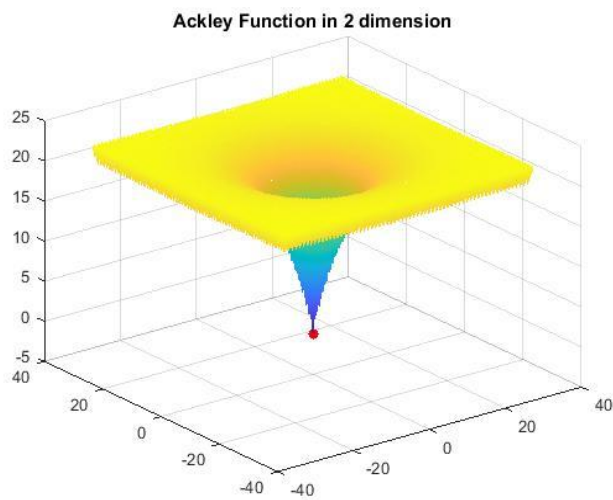
**Figure 3.5: Schwefell best fitness**

The figure 3.6 shows the best fitness for new proposed equation solving the Sphere problem, it shows very good solution.



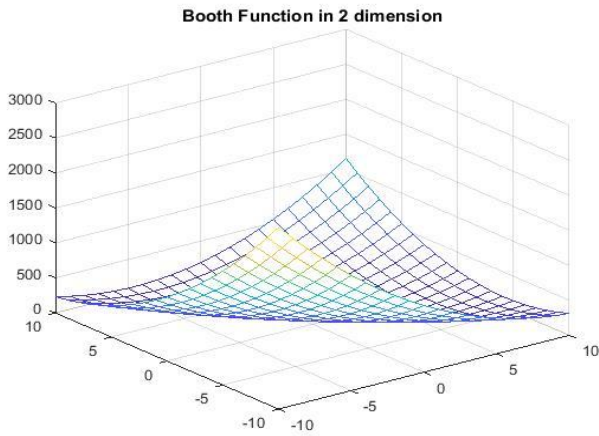
**Figure 3.6: Sphere best fitness**

The figure 3.7 shows the Ackley benchmark function in the form of 2 dimension for new proposed equation, it shows very good Ackley shape.



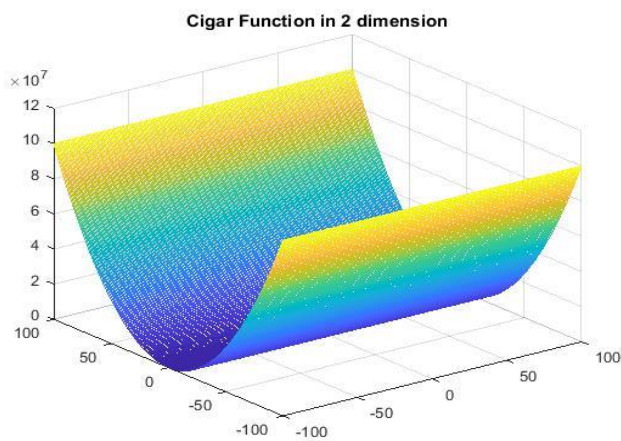
**Figure 3.7: Ackley function in 2 dimensions**

The figure 3.8 shows the Booth benchmark function in the form of 2 dimension for new proposed equation, it shows very good booth shape.



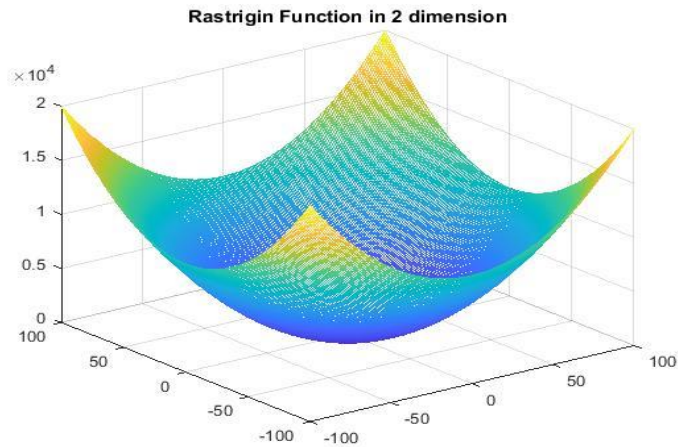
**Figure 3.8: Booth function in 2 dimensions**

The figure 3.9 shows the Ciger benchmark function in the form of 2 dimension for new proposed equation, it shows very good Ciger shape.



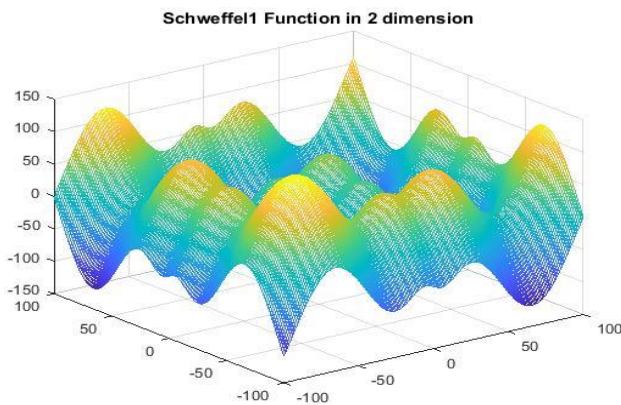
**Figure 3.9: Cigar function in 2 dimensions**

The figure 3.10 shows the Rastrigin benchmark function in the form of 2 dimension for new proposed equation, it shows very good Rastrigin shape.



**Figure 3.10: Rastrigin function in 2 dimensions**

The figure 3.11 shows the Schwefel benchmark function in the form of 2 dimension for new proposed equation, it shows very good Schwefel shape.



**Figure 3.11: Schwefel function in 2 dimensions**

### 3.4 RESULTS AND DISCUSSION

The target of the exploration was to improve the PSO calculation to address its inconveniences, for example, falling into nearby limit esteem, the issue of not getting an around the world ideal arrangement, and the issue of tracking down the best fitness. The new PSO condition was made by joining two inertia weights, specifically linear decreasing inertia weight and chaotic inertia weight. After the blend of the two latency loads, the control factor was consolidated to help control the yield wellness by fluctuating it into reasonable qualities. The control factor symbol is  $\phi$  for this research. The control factor was added as the exponential function capacity. In this exploration numerous computations were done, an examination of the result was done and



numerous re-enactments were finished utilizing the programming language called Matlab. Eight distinctive notable benchmark issues were embraced as a testing issue for this examination to see the impact that the proposed equation for inertia weight has on the PSO execution. The benchmark capacities were the Ackley, the Rastrigin, the Schwefel, the Cigar, the Sphere, and the Booth function, and this benchmark was tried during the reproductions utilizing Matlab. The outcomes are shown in Tables 3.1 to 3.6, and Figure 3.1 to 3.11. The wellness of the new proposed condition PSO calculation meets quicker and it has higher precision. The tables and diagrams likewise show the correlation between the new proposed condition and the old condition. The ideal best fit is zero. It tends to be noticed that the outcomes for the new proposed PSO condition are near zero contrasted with the outcomes for the old traditional PSO condition as shown in the tables and diagrams referenced previously.

### **3.5 CONCLUSION**

This research proposes a new equation for the PSO algorithm. The proposed algorithm is the combination of linearly decreasing inertia weight and chaotic inertia weight, after which a control factor was introduced as an exponential factor. It was very useful for simulations as it is adjustable. The Matlab simulation results of the experiments show that the new proposed equation converges faster and it gives the best fitness compared to linear inertia weight and oscillating inertia weight.

## **CHAPTER 4**

### **4 MIMO PROPORTIONAL-INTEGRAL-DERIVATIVE TUNING BASED ON PSO**

This chapter proposes a new type of MIMO PID controller tuning based on the improved PSO (LCPSO) that was developed using the linearly decreasing inertia weight combined with the chaotic inertia weight. A control factor was introduced as an exponential function. In this research the proposed improved PSO is then used to optimize the PID to minimize the rise time, settling time, time delays, steady state error, and the big overshoot when tuning the MIMO PID controller. The systems performance using experimental trial and error method is not efficient, so in this research the systems are improved to be more efficient. The results for MIMO PID-based improved PSO (LCPSO) tuning is then compared with the results of PID without optimization methods.

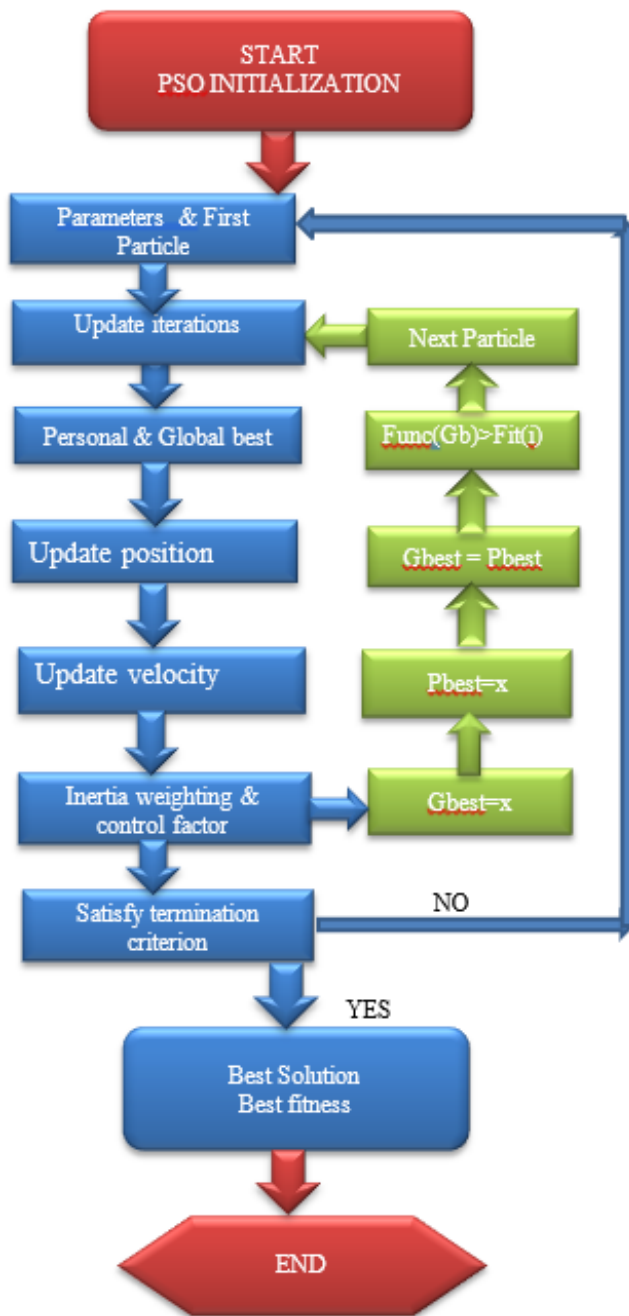
#### **4.1 PARTICLE SWARM OPTIMIZATION**

The PSO optimization technique is not dependent on any information data or angle data of the issue to be addressed or the work to be improved and upgraded. It uses numerical computations. PSO is more affordable or moderate, speedier, increasingly proficient and dynamically beneficial compared to other enhancement techniques. There are very few parameters to change and modify in the PSO. Despite numerous benefits, the PSO has its hindrances, it effectively falls into nearby outrageous value and it cannot reach a universal ideal optimal solution. The PSO equations are demonstrated as follows.

$$V_{id} = V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{id} - X_{id}) \quad (4.1)$$

$$X_{id} = X_{id} + V_{id} \quad (4.2)$$

The PSO parameters used in this examination included: C1- learning factor 1, C2 - global factor 2, Wmax - maximum weight, Wmin - minimum weight, Imax - maximum iteration, N - number of iterations, M - number of particles, Run – number of test time, Xmax – boundary, Z- logistic mapping, Xmin – boundary,  $\phi$  - control factor, and w- inertia weight.



**Figure 4.1: PSO initialization**

Inertia weight is among the boundaries for PSO that influence the imperative basic action during system adjustment. The principal PSO strategy that was founded in 1995 had no inertia weight. In 1998 PSO with the inertia weight was introduced. In recent years there have been many different types of PSOs with different parameters.

The introduction of the control factor  $\phi$  was explained in Chapter 3, as was the detailed step-by-step calculation towards equation 5.

Then the new proposed equation for inertia weight is as follows:

$$w^\phi = \left[ \frac{I_{max} * w_1 - (w_1 - w_2)t + (w_1 - w_2) * I_{max} - I + I_{max} * w_2 Z}{I_{max}} \right]^\phi \quad (4.3)$$

Simplifying the equation (4.3) results in equation (4.4)

$$w^\phi = \left[ \frac{I_{max}(2w_1 - w_2 Z) - (w_1 - w_2)t - I(w_1 - w_2)}{I_{max}} \right]^\phi \quad (4.4)$$

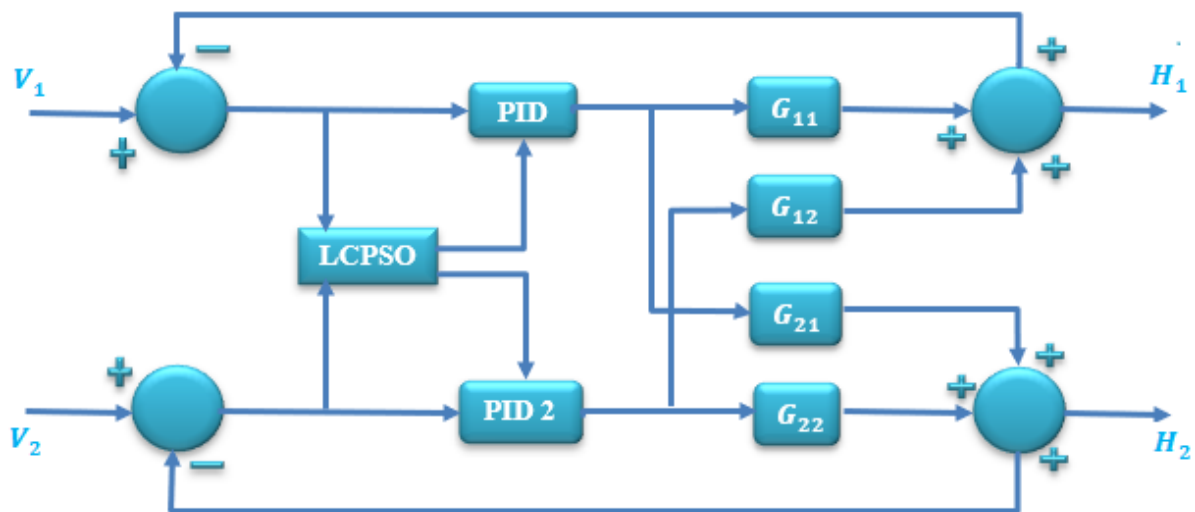
Then equation (4.1) becomes equation (4.5)

$$V_{id} = w^\phi V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{id} - X_{id}) \quad (4.5)$$

The new implemented equation is then used to tune the MIMO PID1 and PID2 parameters using Matlab.

#### 4.2 TUNING OF MIMO PID CONTROLLER SYSTEM BASED ON LCPSO

The transfer function design for the MIMO system is as follows:



**Figure 4.2: MIMO PID control system based on PSO**

In Figure 4.2,  $V_1$  and  $V_2$  are inputs and  $H_1$  and  $H_2$  are outputs,  $G_c(s)$  is the first PID 1 and  $G_{c1}(s)$  is the second PID 2, and the optimization algorithm LCPSO is P. the sub-plant A is  $G_{11}$ , sub-plant B is  $G_{12}$ , sub-plant C is  $G_{21}$ , sub-plant D is  $G_{22}$

The equations from Figure 4.2 are as follows:

$G(s)$  is the transfer function matrix for the plant and is written as follows:

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \quad (4.6)$$

The equations for PID controllers  $G_{c1}(s)$  and  $G_{c2}(s)$  are as follows:

PID 1

$$G_{c1} = K_{p1} \left( 1 + \frac{1}{K_{i1}s} + K_{d1}s \right) \quad (4.7)$$

PID 2

$$G_{c2} = K_{p2} \left( 1 + \frac{1}{K_{i2}s} + K_{d2}s \right) \quad (4.8)$$

The equation for a close loop showing the feedback is

$$H_1(s) = \frac{G_{c1}G_{11}}{1+G_{c1}G_{11}}V_1(s) \quad H_1(s) = \frac{G_{c2}G_{12}}{1+G_{c2}G_{12}}V_2(s) \quad (4.9)$$

$$H_2(s) = \frac{G_{c1}G_{21}}{1+G_{c1}G_{12}}V_1(s) \quad H_2(s) = \frac{G_{c2}G_{22}}{1+G_{c2}G_{22}}V_2(s) \quad (4.10)$$

$G_c(s)$  is the transfer function matrix for  $G_{c1}(s)$  and  $G_{c2}(s)$ .

$$G_c(s) = \begin{bmatrix} G_{c1}(s) & 0 \\ 0 & G_{c2}(s) \end{bmatrix} \quad (4.11)$$

$$\frac{H(s)}{V(s)} = G_c(s)G(s) \quad (4.12)$$

$$\frac{H(s)}{V(s)} = \begin{bmatrix} G_{c1}(s) & 0 \\ 0 & G_{c2}(s) \end{bmatrix} \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \quad (4.13)$$

These tuning methods for the MIMO PID controllers lead to setting the values of the P, I, and D parameters. The tuning of the PID controller is done by adjusting the gains of the Kp, Ki, and Kd parameters. The required performance specifications such as stability margins, dynamic response for both transient and steady state are incurred by the best set of parameters. There are many criteria that can be used to find the best response of the control system.

The integral of absolute error (IAE) is written as:

$$IAE = \int_0^{\infty} |r(t) - y(t)| dt = \int_0^{\infty} |e(t)| dt \quad (4.14)$$

The integral of square of error (ISE) can be written as:

$$ISE = \int_0^{\infty} e^2(t) dt \quad (4.15)$$

The integral of time multiplied by square of error (ITSE) is written as:

$$ITSE = \int_0^{\infty} te^2(t) dt \quad (4.16)$$

Integral of time multiplied by the absolute of error (ITAE) can be given by:

$$ITAE = \int_0^{\infty} tv(t) dt \quad (4.17)$$

The outcomes demonstrate the PID tuning based on PSO to be an improved approach (16). PID is the most favoured controller for many reasons, including its straightforward algorithm and its capacity to adjust to an extensive variety of uses where it can guarantee brilliant control exhibitions the micro processing that has been done has highlighted PID controllers, for example, programmed automatic tuning, pick up planning and nonstop adjustment. PSO requires crude scientific and mathematical administrators, which makes it worthwhile as far as the accessibility of bigger memory and higher speed. It has effectively been connected to a wide assortment of issues, for example neural systems, auxiliary advancement, share topology improvement, and fluffy systems. When enhancing the PSO, the expectation is to make social weight expanded while latency weight diminishes. Little social weight influences the worldwide best position to have a minor effect on the speed refreshing. In this research the MIMO PID controller tuning is based on the LCPSO.

### 4.3 METHODOLOGY

This chapter explains the step by step of tuning the MIMO PID controller. The system with two loops, four plants, and two PIDs was designed in the tool called Simulink as shown in figure 4.4. The PSO was coded on the tool called Matlab, then the Simulink was then called from Matlab as the function using sim (). The process was repeated multiple times and results are presented in the form of tables.

The steps below were followed to tune a MIMO PID controller based on the new proposed equation for PSO. The process of calling the Simulink on Matlab using the sim command was used in this research

**Stage 1:** The PSO parameters were as follows:  $C1 = C2 = 1.9-2.2$ ,  $W_{max} = 1$ ,  $W_{min} = 0$ ,  $I_{max} = 50$ ,  $N = 2$ ,  $M = 50$ . The changing parameters were:  $X_{max} = [30; 30]$ ,  $X_{min} = [-30; -30]$  and  $\phi = -10$

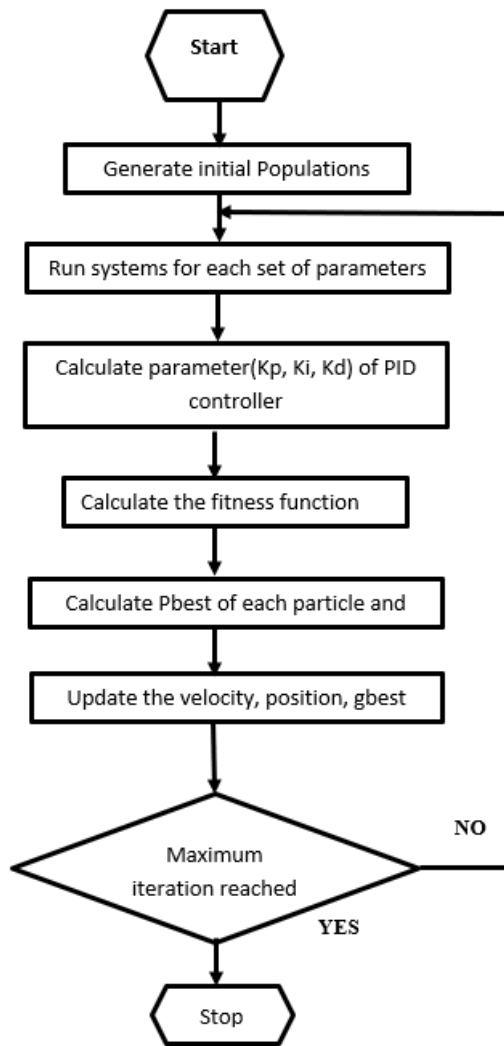
**Stage 2:** Initialize the particle position and velocity.

**Stage 3:** Call the function and select the particles' singular best value for every generation.

**Stage 4:** Select the particles' global best. The particles close to the objective among all the particles, is acquired by contrasting and comparing all the individual best qualities.

**Stage 5:** Update pbest, gbest in the speed per second and acquire the new velocity.

**Stage 6:** Find the best optimal arrangement with a minimum ISE, IAE, ITAE and ITSE from the refreshed new speed and position scientific information and time unpredictable is resolved. Call Simulink model using Sim() function.



**Figure 4.3: Flowchart for PSO-PID system**

#### 4.4 SIMULATION

$G(s)$  is the given matrix for sub-system A, sub-system B, sub-system C and sub-system D, which were used for Matlab simulations.

$$G(s) = \begin{bmatrix} \frac{1}{s^2+20s+15} & \frac{1}{s^2+21s+17} \\ \frac{1}{s^2+25s+19} & \frac{1}{s^2+28s+21} \end{bmatrix} \quad (4.18)$$

In the following equation  $H(s)$  is the output and  $V(s)$  is the input of the system in Figure 4.3.

$$H_1 = \left( K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) + P \right) \left[ \frac{1}{s^2+20s+15} V_1(s) + \left( \frac{1}{s^2+21s+21} V_2(s) \right) \right] \quad (4.19)$$



$$H_2 = \left( K_{p1} \left( 1 + \frac{1}{T_{i1}s} + T_{d1}s \right) + P \right) \left[ \frac{1}{s^2+25s+19} V_1(s) + \left( \frac{1}{s^2+28s+21} V_2(s) \right) \right] \quad (4.20)$$

The system in figure 4.5 were used for Matlab simulation

The model was designed using the Simulink with PID1 and PID2 and four plants. There were two loops for the model, loop 1 and loop 2. The model is called on the Matlab using the sim() function. The below table gives the Matlab simulation results for step response, showing overshoot, rise time, settling time and steady state,

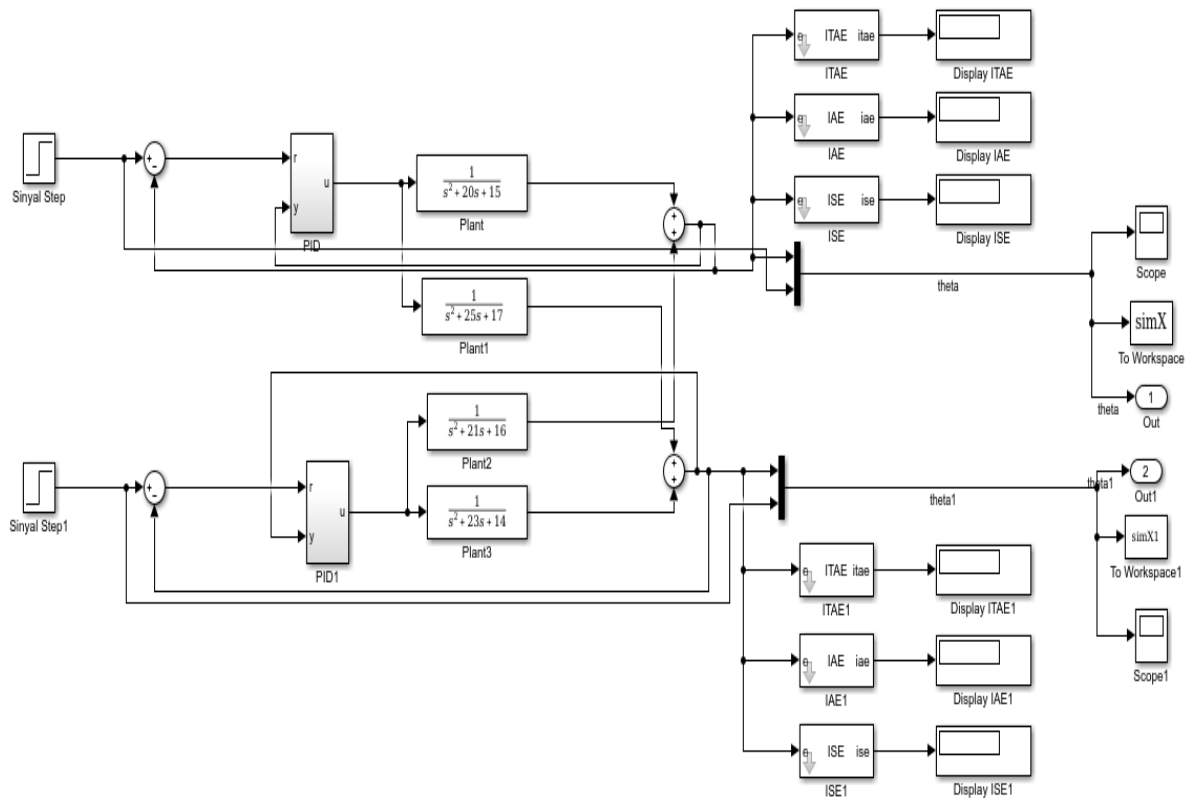


Figure 4.4: Simulink model for MIMO

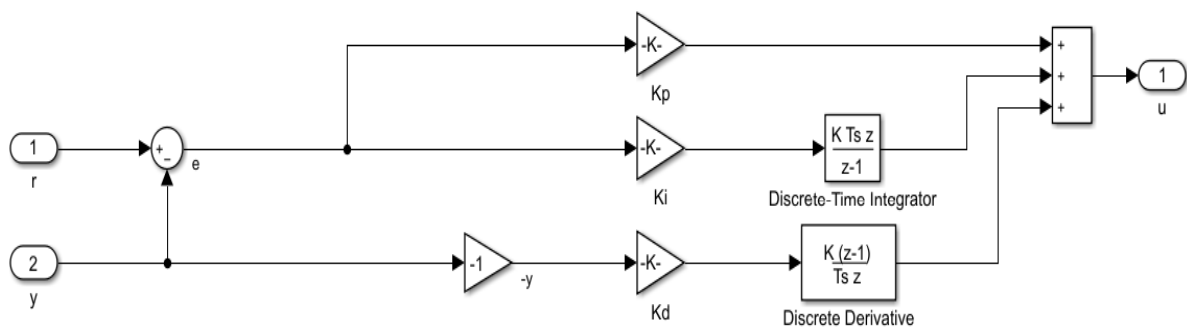
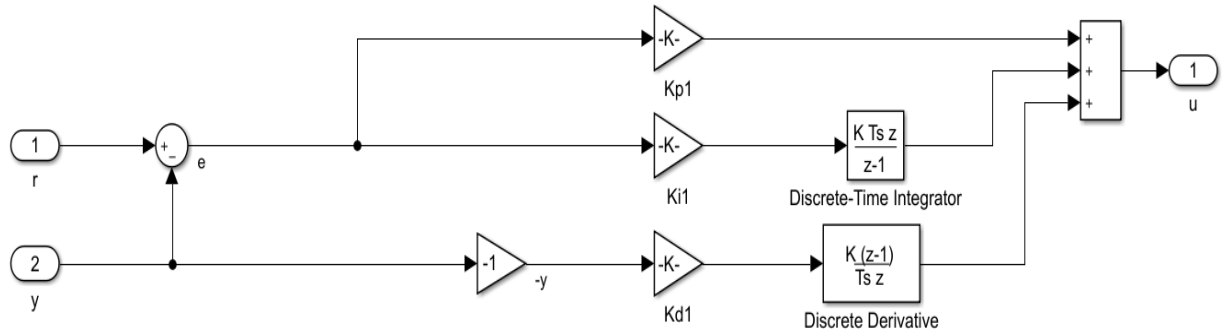


Figure 4.5: Simulink model for PID1



**Figure 4.6: Simulink model for PID2**

Table 4.1 shows the results for tuned PID1 parameters  $K_p$ ,  $K_i$  and  $K_d$ . The model was designed using the Simulink with PID1 and PID2 and four plants. There were two loops for the model, loop 1 and loop 2. The model is called on the Matlab using the `sim()` function. Below table 4.1 gives the Matlab simulation results for the MIMO PID controller based on improved PSO parameters.

**Table 4.1: Controller parameters for PID1 tuning based on the new proposed PSO (LCPSO).**

Parameters	Tuned values (First run)	Tuned values (Second run)	Tuned values (Third run)	Tuned values (Fourth run)	Average values
$K_p$	33.6248	37.7335	49.9390	41.9334	40.8077
$K_i$	22.3180	20.3240	7.3421	5.3474	13.8329
$K_d$	0.0612	0.07120	1.1505	1.0195	0.5756

Table 4.2 shows the results for tuned PID2 parameters  $K_{p1}$ ,  $K_{i1}$  and  $K_{d1}$ . The model was designed using the Simulink with PID1 and PID2 and four plants. There were two loops for

the model, loop 1 and loop 2. The model is called on the Matlab using the sim() function. Below table 4.1 gives the Matlab simulation results for the MIMO PID controller based on improved PSO. parameters.

**Table 4.2: Controller parameters for PID2 tuning based on the new proposed PSO (LCPSO).**

Parameters (PID)	Tuned values (First run)	Tuned values (Second run)	Tuned values (Third run)	Tuned values (Fourth run)	Average values
$K_{p1}$	52.22	45.96	43.60	41.97	45.9375
$K_{i1}$	9.31	8.35	8.21	7.91	8.4452
$K_{d1}$	0.45	1.56	0.5	1.77	1.0744

The Table 4.3 gives the Matlab simulation results for step response, showing overshoot, rise time, settling time and steady-state.

**Table 4.3: Simulation results for PID and the PID-LCPSO for output 1**

Tuning Method	PID	PID-LCPSO
Overshoot (%)	5.02	0.12
Rise Time	2.41	0.15
Settling time	4.33	0.227
steady state	0.91	1

The Table 4.4 gives the Matlab simulation results for step response, showing overshoot, rise time, settling time and steady state.

**Table 4.4: Simulation results for PID and the PID-LCPSO for output 1**

<b>Tuning Method</b>	<b>PID</b>	<b>PID-LCPSO</b>
Overshoot (%)	4.25	0.00
Rise Time	2.38	0.175
Settling time	4.27	0.288
steady state	0.98	1

The Table 4.5 gives the Matlab simulation results for step response, showing overshoot, rise time, settling time and steady state.

**Table 4.5: Simulation results for PID and the PID-LCPSO for output 2**

<b>Tuning Method</b>	<b>PID</b>	<b>PID-LCPSO</b>
Overshoot (%)	2.1440	0.00
Rise Time	0.4509	0.141
Settling time	0.6166	0.364
steady state	0.99	1

The Table 4.6 gives the Matlab simulation results for step response, showing overshoot, rise time, settling time and steady state.

**Table 4.6: Simulation results for PID and the PID-LCPSO for output 2**

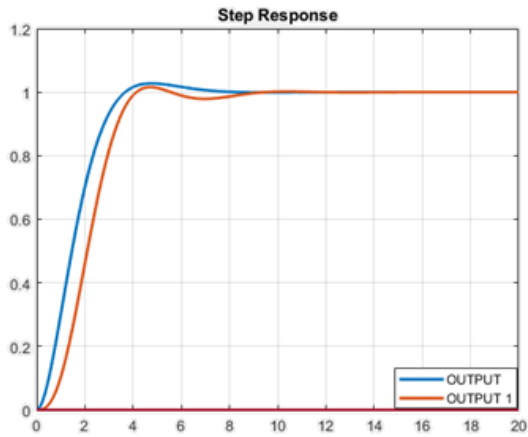
Tuning Method	PID	PID-LCPSO
Overshoot (%)	3.40	0.022
Rise Time	2.59	0.188
Settling time	4.33	0.238
steady state	0.97	1

The figure 4.7 present the results from Matlab, showing the results for systems without a PID. They give the results of PID controller tuning without any optimization method and it also shows PID controller systems tuning based on PSO. It should be noted that the results show a great difference in terms of overshooting, rise time, settling time and steady state error.



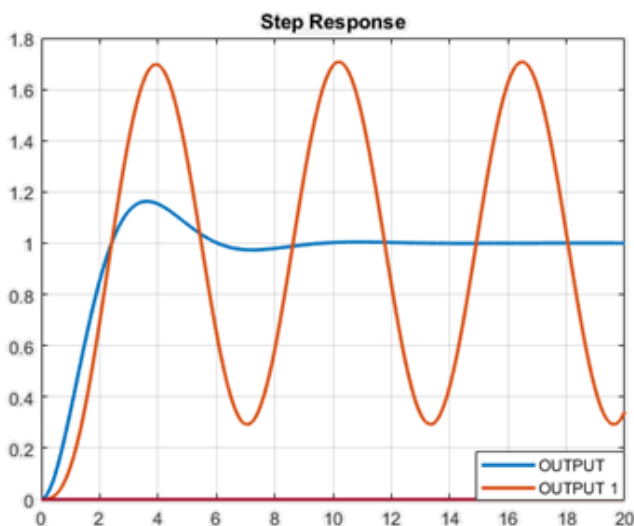
**Figure 4.7: System step response result for PID tuning based on LCPSO optimization**

The figure 4.8 present the results from Matlab, showing the results for systems without a PID. They give the results of PID controller tuning without any optimization method and it also shows PID controller systems tuning based on PSO. It should be noted that the results show a great difference in terms of overshooting, rise time, settling time and steady state error.



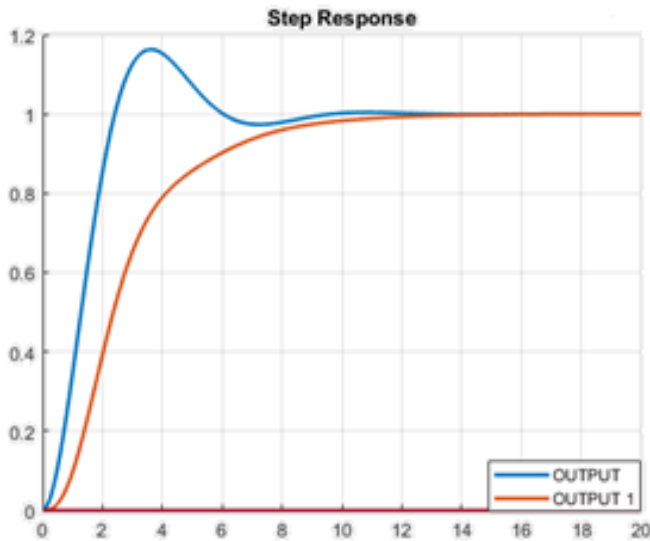
**Figure 4.8: System step response result for PID tuning based on LCPSO optimization**

The figure 4.9 present the results from Matlab, showing the results for systems without a PID. They give the results of PID controller tuning without any optimization method and it also shows PID controller systems tuning based on PSO. It should be noted that the results show a great difference in terms of overshooting, rise time, settling time and steady state error.



**Figure 4.9: Showing MIMO PID tuning in progress. One output is stable and the other is still searching for best results.**

The figure 4.10 present the results from Matlab, showing the results for systems without a PID. They give the results of PID controller tuning without any optimization method and it also shows PID controller systems tuning based on PSO. It should be noted that the results show a great difference in terms of overshooting, rise time, settling time and steady state error.



**Figure 4.10: Output step response results**

#### 4.5 RESULTS AND DISCUSSION

The new proposed equation achieved in chapter 3 was utilized to achieve the next main objective. The objective for this chapter was to tune MIMO PID controller systems using the improved PSO in order to solve the problem of high oscillations in a step response characteristic such as rise time, settling time, time delays, steady-state error, and big overshoot. The combination of the Matlab code and the Simulink model was used to tune the PID controller parameters. Figure 4.4 shows the Simulink model that was created using the Matlab file. The new improved PSO was used to tune the parameters of the PIDs. Tables 4.1 to 4.6 show the results for the PID1 controller and the PID2 controller's tuned parameters. The two PIDs are used to optimize the four different plants that make one system. It is evident that the rise time, settling time, time delays, and steady-state error are minimized when tuning the MIMO PID controller based on the new proposed LCPSO algorithm. The results are also displayed in Figures 4.7 to 4.10, where the curves and graphs show the better outcome curve.

## **4.6 CONCLUSION**

In this research, the MIMO PID controller was tuned based on a newly proposed PSO algorithm. The newly proposed PSO is based on combined inertia weight. The optimized Matlab results less rise time, settling time, time delays and no steady state error. The comparison between PID-LCPSO execution and the PID is presented in the tables to show the advantage of PID tuning using a LCPSO-based improvement approach.



## CHAPTER 5

### 5.1 RESULTS AND DISCUSSION

The objective of the research was to improve the PSO algorithm to address its disadvantages, such as falling into local extreme value, the issue of not obtaining a globally optimal solution and the problem of finding the best fit. The new PSO equation was created by combining two inertia weights, namely linear decreasing inertia weight and chaotic inertia weight. After the combination of the two inertia weights, the control factor was incorporated to help control the output fitness by varying it into the suitable values. The control factor symbol is  $\phi$  for this research. The control factor was added as the exponential function. In this research many calculations were done, a comparison of result was done and many simulations were done using the programming language called Matlab. Eight different well-known benchmark problems were adopted as a testing problem for this research to see the influence that the proposed equation for inertia weight has on the PSO performance. The benchmark functions were the Ackley function, the Rastrigin function, the Schwefel function, the Cigar function, the sphere function, and the Booth function and this benchmark was tested during the simulations using Matlab. The results are shown in Tables 3.1 to 3.6, and Figure 3.1 to 3.11. The fitness of the new proposed equation PSO algorithm converges faster and it has a higher accuracy. The tables and graphs also show the comparison of the new proposed equation and the old equation. The ideal best fit is zero. It can be noted that the results for new proposed PSO equation are very close to zero compared to the results for old PSO equation as displayed in the tables and graphs mentioned above.

Another main objective of this research was to tune MIMO PID controller systems using the improved PSO in order to solve the problem of high oscillations in a step response characteristic such as rise time, settling time, time delays, steady state error and big overshoot. The combination of the Matlab code and the Simulink model was used to tune the PID controller parameters. Figure 4.4 shows the Simulink model that was created using the Matlab file. The new improved PSO was used to tune the parameters of the PIDs. Tables 4.1 to 4.6 show the results for the PID1 controller and the PID2 controller's tuned parameters. The two PIDs are used to optimize the four different plants that make one system. It is evident that the rise time, settling time, time delays and steady state error are minimized when tuning the MIMO PID controller based on new proposed LCPSO algorithm. The results are also displayed in Figure 4.7 to 4.10, where the curves and graphs show the better outcome curve.

## **5.2 CONCLUSION**

This research has proposed a new equation for a PSO algorithm. The proposed algorithm is the combination of linearly decreasing inertia weight and chaotic inertia weight, after which a control factor was introduced as an exponential factor. This was very useful for simulations as it is adjustable. The Matlab simulation results of the experiments show that the new proposed equation converges faster and it gives the best fit compared to linear inertia weight and oscillating inertia weight and other old equations. The MIMO PID controller system that consists of four plants was tuned based on the new proposed equation for the PSO algorithm (LCPSO). The optimized results show less rise time, settling time, time delays and steady state compared to the systems that are tuned using the old equations. The exploration was directed at considering the impact of using the PSO calculation as an instrument for MIMO PID tuning. The results obtained in the examination reveal that the PSO tuning output improved reactions and can be applied to various system models in the measure control industry.

## **5.3 FUTURE WORK**

More study can be done to improve the optimization algorithm, for example, the combination of Particle swarm optimization, the generic algorithm, differential evolution, and other machine learning based on multi-objective evolution algorithm to autotune the parameters of even bigger systems. Moreover, Research can also be done in order to apply new ideas found in other areas of science or technologies to optimize the performance, minimize the cost or reduce the time related to the accomplishment of optimal solutions. It can be advantageous to interchange the different abilities of the PSO, generic algorithm, and differential evolution to improve proficiency and searchability in other to tune more complicated control systems.

## **PUBLICATIONS**

1. T. B. Nkwanyana and Z. Wang, “Improved Particle Swarm Optimization Base on the Combination of Linear Decreasing and Chaotic Inertia Weights”, 12th International Conference on Computational Intelligence and Communication Networks, 2020, pp. 460–465.
2. T. B. Nkwanyana and Z. Wang, “Tuning of multi-input multi-output proportional-integral-derivative controller based on improved linearly and chaotic inertia weight particle swarm optimization”, submitted to 2021 International Conference on Neural Computing for Advanced Applications.

## REFERENCES

- [1] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 1998, pp. 69-73
- [2] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, South Korea, 2001, pp. 94-100
- [3] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multistage linearly-decreasing inertia weight", In International Joint Conference on Computational Sciences and Optimization, 2009, volume 1, pp. 505–508.
- [4] Y. Feng, G.F. Teng, A.X. Wang, and Y.M. Yao, "Chaotic inertia weight in particle swarm optimization", 2007 In Innovative Computing, Information and Control, 2007, ICICIC'07. Second International Conference, pp. 475-457.
- [5] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon and A. Abraham, "Inertia weight strategies in particle swarm optimization," 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, 2011, pp. 633-640
- [6] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, volume 4, pp. 1942–1948.
- [7] Y. Sun and Z. Wang, "Asynchronous and stochastic dimension updating PSO and its application to Parameter Estimation for Frequency Modulated (FM) Sound Waves," 2016.
- [8] S. T. Dharan, K. Kavyarasan and V. Bagyaveereswaran, "Tuning of PID controller using optimization techniques for a MIMO process," IOP Conference Series: Materials Science and Engineering, 2017.
- [9] A. Taeib, S. Moez and A. Chaari, "PID Control for MIMO Nonlinear Processes," World Applied Sciences Journal, vol 25, no. 3, 2013, pp. 487–493.
- [10] M. F. Pungot, "PID control tuning using cuckoo search for multi-input multi-output coupled tank system", 2015.

- [11] A. J. S. Mary, and P. L. Suresh, "Optimization of PID Controller for a Nonlinear MIMO System and Comparative Performance Analysis," *International Journal of Mechanical Engineering and Technology* 8(8), 2017, pp. 777–787.
- [12] E. T. Neto, S. A. Imtiaz and S. Ahmed, "A sequential optimization procedure to tune MIMO PID controllers," *2017 Indian Control Conference (ICC)*, Guwahati, 2017, pp. 415–420.
- [13] K. L. Sowjanya, and L. R. Srinivas, "Tuning of PID controllers using particle swarm optimization," *International Journal of Industrial Electronics and Electrical Engineering*, Volume 3, Issue 2, 2015, pp. 2347–6982.
- [14] P. Biswas, R. Maiti, A. Kolay, K. Das Sharma and G. Sarkar, "PSO based PID controller design for twin rotor MIMO system," *Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, Calcutta, 2014, pp. 56-60.
- [15] A.K. Vincent and R. Nersisson, "Particle swarm optimization based PID controller tuning for level control of two tank system", *IOP Publishing, 14th ICSET-2017. IOP Conf. Series: Materials Science and Engineering*, , 2017, pp. 263-263.
- [16] M. I. Solihin, L. F. Tack, and M. L. Kean, "Tuning of PID Controller Using Particle Swarm Optimization (PSO)," *Proceedings of the International Conference on Advanced Science, Engineering and Information Technology*, vol.1, no. 4, 2011, pp. 458-461.
- [17] S. Talukder, "Mathematical Modelling and Applications of Particle Swarm Optimization," M.S. Thesis School of Eng., Blekinge Institute of Technology, Karlskrona, Sweden, 2011.
- [18] T. Adel, L. Ali and C. Abdelkader, "A PSO approach for optimum design of multivariable PID controller for nonlinear systems," *Research Unit on Control, Monitoring and Safety of Systems (C3S) High School ESSTT*, 2013.
- [19] J Wang, B. T. Brackett and R. G. Harley, "Particle Swarm-assisted state feedback control: From pole selection to state estimation," *2009 American Control Conference*, St. Louis, MO, 2009, pp. 1493-1498.
- [20] S. Boyd, M. Hast and K. J. Åström, "MIMO PID tuning via iterated LMI restriction," *Int. J. Robust Nonlinear Control*, 2016.

- [21] L. Yi, "Study on an Improved PSO Algorithm and its Application for Solving Function Problem," *International Journal of Smart Home*, Vol. 10, No. 3, 2016, pp. 51–62.
- [22] W. Li, "Improving Particle Swarm Optimization Based on Neighborhood and Historical Memory for Training Multi-Layer Perceptron," *Information*, vol. 9, no. 1, pp. 16, 2018.
- [23] Z. Zhang, C. Ma, and R. Zhu, "Self-Tuning Fully-Connected PID Neural Network System for Distributed Temperature Sensing and Control of Instrument with Multi-Modules," *Sensors*, vol. 16, no. 10, pp. 1709, 2016.
- [24] J. Saveca, "Multi-objective power quality optimization of smart grid based on improved differential evolution," M.T. thesis, Dept. Elect. Eng., Univ of South Africa., South Africa, 2018.
- [25] N. Pillay, "A particle swarm optimization approach for tuning of SISO PID control loops," M.S. thesis, Dept. Elect. Eng., Durban Univ of Tech., South Africa, 2008.
- [26] T. B. Nkwanyana and Z. Wang, "Improved Particle Swarm Optimization Base on the Combination of Linear Decreasing and Chaotic Inertia Weights", 12th International Conference on Computational Intelligence and Communication Networks, 2020, pp. 460–465.
- [27] J. Saveca., Z. Wang and Y. Sun, "Improved differential evolution based on mutation strategies". *Lecture Note in Computer Science*, Springer, Cham, Volume 10941, 2018, pp. 233–242.
- [28] M. Zamani, N. Sadati and M.K. Ghartemani, "Design of an  $H_{\infty}$  PID controller using particle swarm optimization. *Int. J. Control Autom. Syst.* **7**, 2009. pages 273–280.
- [29] W. F. W. Tarmizi et al., "A Particle Swarm Optimization-PID controller of a DC Servomotor for Multi-Fingered Robot Hand," 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), Ipoh, 2016, pages 1-6.
- [30] K.R. Harrison, A.P. Engelbrecht and B.M. Ombuki-Berman, "Inertia weight control strategies for particle swarm optimization". *Swarm Intell* 10, 2016. Pages 267–305.
- [31] N. Kumar and S. Kumar Sharma, "Inertia Weight Controlled PSO for Task Scheduling in Cloud Computing," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, Uttar Pradesh, India, 2018, pp. 155-160.

- [32] E.E. Agbugba, "Hybridization of Particle Swarm Optimization with Bat Algorithm for Optimal Reactive Power Dispatch," M.T. thesis. Dept. Elect. Eng., Univ of South Africa, South Africa, 2008.
- [33] A. Mishra, "A Study on Pid Controller Design for Systems with Time Delay," National Institute of Technology Rourkela -769 008, Orissa, India, 2011
- [34] A. H. Devikumari, and V. Vijayan. "Decentralized PID Controller Design for 3x3 Multivariable System using Heuristic Algorithms," Indian journal of science and technology 8 .2015.
- [35] A. Ilakkiya, S. Divya and M. Mannimozhi," Design of PI controllers for MIMO System with decouplers," International Journal of Chemical Sciences, 14 (3), 2016, pages 1598-1612.
- [36] S.K. Mallick and M.A. Khan, "Study of the Design and Tuning Methods of PID Controller Based on Fuzzy Logic and Genetic Algorithm," B Tech. Electronics and Communication Eng., National Institute of Technology, Rourkela, 2011.
- [37] A. O'Dwyer,"PID control: the early years," Control in the IT Sector Seminar, Cork Institute of Technology, Cork, 2005.
- [38] J. Youney, "A comparison and evaluation of Common PID tuning method," M S. thesis Engineering and Computer Science. University of Central Florida, Orlando, Florida . 2007.
- [39] N. Pillay and P. Govender, "A Particle Swarm Optimization Approach for Model Independent Tuning of PID Control Loop", IEEE Africon 2007

## APPENDICES

### APPENDIX A

#### MATLAB CODES

```
PSO CODE

tic
clear all;
clc;

c1=2;           %%%%%%%%% Learning factor
c2=2;           %%%%%%%%% Golbal Learning
factor
W_max=1;        %%%%%%%%% Maximum weight
W_min=0;        %%%%%%%%% Minimum weight
I_max=1;        %%%%%%%%% Maximum iteration
N=6;           %%%%%%%%% Numer of
dimension
M=50;          %%%%%%%%% Number of
Particle
Run=1;         %%%%%%%%% The number of
test time
X_max= [100,30,2,100]; %%%%%%%%% Boundary
X_min= [0,0,0,0,0,0]; %%%%%%%%% Boundary
V_max=1;
Z=0.75;
I=5;
Func=@matlab_sim_project;

for r=1:Run
    % Initialize
    for c=1:M
        for y=1:N
```

```

        x(c,y)=X_min(y)+rand()*(X_max(y)-
X_min(y));
        s(c,y)=rand()*(X_max(y)-X_min(y));
    end
    Fit(c,:)=Func(x(c,:));
    Pb(c,:)=x(c,:);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%global
best

[gb1,ind1]=sort(Fit);
Gb=x(ind1(1,1),:);

for t=1:I_max

    t=t+1;

    for c=1:M

Update the Pb

        if Func(x(c,:))<Fit(c)
            Fit(c)=Func(x(c,:));
            Pb(c,:)=x(c,:);
        end

Update the Gb

        if Func(Gb)>Fit(c)
            Gb=Pb(c,:);

```



```

end

Update the velocity
Calculate the weighting function

n=-10;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%control factor
w=[ [I_max*(2*W_max-W_min*Z)-(W_max-
W_min)*t-I*(W_max-W_min)]/I_max]^n;

s(c,:)=w*s(c,:)+c1*rand*(Pb(c,:)-
x(c,:))+c2*rand*(Gb-x(c,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Check the velocity

for y=1:N
    if s(c,y)>V_max
        s(c,y)=V_max;
    elseif s(c,y)< -V_max
        s(c,y)=-V_max;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Update the position

x(c,:)=x(c,:)+s(c,:);

end

```

```

        Y=Func(Gb);
        % Plot, just for look
        figure(1);
        plot(t,Y);
        xlabel('Iteration');
        ylabel('Fitness');
        title(sprintf('Cider Best Fitness: %.15f',Y));
        grid on;
        hold on;

        dt=0.01;
        tf=20;
        t=(0:dt:tf)';
        D=numel(t);
        y=zeros(D,7);
        for y=1:2
            x=[zeros(y+1,1);1];
            for k=1:D
                x=2*x;
                y(k,y)=x(1);
            end
        end
        plot(t,y,'Linewidth',2);
        grid on
        title('ITAE Step Response');legend('OUTPUT 1','OUTPUT
2');figure;
        hold
        %}

        end
        figure(2);
        PlotC();

```

```
    hold on;  
    scatter3(Gb(1),Gb(2),Y,'fill','ro');  
    hold off;  
end  
toc
```

## **BENCHMARK**

### **Sphere function**

```
function y = Sphere (x)  
  
    n = 2;  
    sp = 0;  
  
    for i = 1:n  
        sp = sp + x(i)^2;  
    end  
  
    y = sp;  
  
function PlotS()  
  
[X,Y] = m(-110:0:10,-110:0:10);  
N = size(X,1);
```

```

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) = Sphere(z);
    end
end

figure(2);
mesh(X,Y,Z);
title('Sphere Function in 2 dimension');
axis([-5.5 5.5 -5.5 5.5 0 9]);

end

[X,Y] = meshgrid(-10:0.03:10,-10:0.03:10);
N = size(X,1);

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) = Sphere(z);
    end
end

figure(2);
mesh(X,Y,Z);
title('Sphere Function in 2 dimension');
axis([-5.5 5.5 -5.5 5.5 0 9]);

```

**Schweffel function**

```

function varargout = schweffel(X)
% Schweffel function
% if no input is given, return dimensions, bounds and
minimum
    if (nargin == 0)
        varargout{1} = 2; % # dims
        varargout{2} = [-500 -500]; % LB
        varargout{3} = [+500 +500]; % UB
        varargout{4} = [4.209687467626741e+002
4.209687464869218e+002]; % solution
        varargout{5} = -8.379657745448676e+002; % function
value at solution

% otherwise, output function value
    else

        % keep all values in the search domain
        X(X < -500) = inf; X(X > 500) = inf;

        % split input vector X into x1, x2
        if size(X, 1) == 2
            x1 = X(1, :); x2 = X(2, :);
        else
            x1 = X(:, 1); x2 = X(:, 2);
        end

        % output function value
        varargout{1} = -x1.*sin(sqrt(abs(x1))) -
x2.*sin(sqrt(abs(x2)));
    end
end

```

```

function y = schwef(x)

%
n = 2;
s = 0;
p = 0;

for i = 1:n
    s = s + abs(x(i));
    p = p + abs(x(i));
end

y = s + p;

end

function Plotschw()

[X,Y] = meshgrid(-100:100,-100:100);
N = size(X,1);

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) =schweffell (z);
    end
end

mesh(X,Y,Z);
title('Schweffell Function in 2 dimension');

end

```

## Rastrigi Function

```
% Rastrigin's Function

function y = Rastrigin (X)

A = 10;
n = 2;
m = 0;

for i = 1:n
    m = m + X(i)^2 - A*cos(2*pi*X(i));
end

y = 10*n + m;

end

function varargout = rastrigin(X)
% Rastrigin function

% if no input is given, return dimensions, bounds and
minimum
if (nargin == 0)
    varargout{1} = 2; % # dims
    varargout{2} = [-5.12, -5.12]; % LB
    varargout{3} = [+5.12, +5.12]; % UB
    varargout{4} = [0, 0]; % solution
    varargout{5} = 0; % function value at solution

% otherwise, output function value
```

```

else

    % keep all values in the search domain
    X(X < -5.12) = inf;    X(X > 5.12) = inf;

    % split input vector X into x1, x2
    if size(X, 1) == 2
        x1 = X(1, :);    x2 = X(2, :);
    else
        x1 = X(:, 1);    x2 = X(:, 2);
    end

    % output function value
    varargout{1} = x1.^2 + x2.^2 - 10*cos(2*pi*x1) -
10*cos(2*pi*x2) + 20;

end

end

function PlotR()

[X,Y] = meshgrid(-100:100,-100:100);
N = size(X,1);

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) =Rastrigin (z);
    end
end

end

mesh(X,Y,Z);
title('Rastrigin Function in 2 dimension');

```



```
end
```

### **Cigar function**

```
function val=cigar(x)
```

```
val=x(1)^2;
```

```
n=size(x,2);
```

```
for i=2:n
```

```
    val=val+(10^4)*x(i)^2;
```

```
end
```

```
%x from [-5 5]
```

```
function PlotC()
```

```
[X,Y] = meshgrid(-100:100,-100:100);
```

```
N = size(X,1);
```

```
for i = 1:N
```

```
    for j = 1:N
```

```
        z = [X(i,j),Y(i,j)];
```

```
        Z(i,j) = Cigar (z);
```

```
    end
```

```
end
```

```
mesh(X,Y,Z);
```

```
title('Cigar Function in 2 dimension');
```

```
end
```

## Booth function

```
function varargout = booth(X)
% Booth function

% if no input is given, return dimensions, bounds and
minimum
if (nargin == 0)
    varargout{1} = 2; % # dims
    varargout{2} = [-10, -10]; % LB
    varargout{3} = [+10, +10]; % UB
    varargout{4} = [1, 3]; % solution
    varargout{5} = 0; % function value at solution

% otherwise, output function value
else

    % keep values in the search interval
    X(X < -10) = inf;    X(X > 10) = inf;

    % split input vector X into x1, x2
    if size(X, 1) == 2
        x1 = X(1, :);    x2 = X(2, :);
    else
        x1 = X(:, 1);    x2 = X(:, 2);
    end

    % output function value
    varargout{1} = (x1 + 2*x2 - 7).^2 + (2*x1 + x2 -
5).^2;

end
```

```

function Plotbooth()

[X,Y] = meshgrid(-100:100,-100:100);
N = size(X,1);

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) = booth (z);
    end
end

mesh(X,Y,Z);
title('Booth Function in 2 dimension');

end

```

### Ackley function

```

function varargout = ackley(X)
% Ackley funcion
%
%
    if (nargin == 0)
        varargout{1} = 2; % # dims
        varargout{2} = [-35, -35]; % LB
        varargout{3} = [+35, +35]; % UB
        varargout{4} = [3, 0.5]; % solution
        varargout{5} = 0; % function value at solution

        % otherwise, output function value
    else

```

```

    % Keep all values in the search domain

    X(X < -35) = inf;    X(X > 35) = inf;

    % split input vector X into x1, x2
    if size(X, 1) == 2
        x1 = X(1, :);    x2 = X(2, :);
    else
        x1 = X(:, 1);    x2 = X(:, 2);
    end

    % output function value
    varargout{1} = 20*(1 - exp(-0.2*sqrt(0.5*(x1.^2 +
x2.^2))))...
        - exp(0.5*(cos(2*pi*x1) + cos(2*pi*x2))) +
exp(1);

    end

end

function y = Ackley(x)
    n=2;
    sum1 = 0;
    sum2 = 0;

    for i = 1:n
        sum1 = sum1 + x(i)^2;
        sum2 = sum2 + cos((2*pi) * x(i));
    end
end

```

```

        y = 20 + exp(1)-20*exp(-0.2*sqrt(1/n*sum1))-
exp(1/n*sum2);

end

function PlotA()

[X,Y] = meshgrid(-32.768:0.03:32.768,-32.768:0.03:32.768);
N = size(X,1);

for i = 1:N
    for j = 1:N
        z = [X(i,j),Y(i,j)];
        Z(i,j) = Ackley(z);
    end
end

mesh(X,Y,Z);
title('Ackley Function in 2 dimension');

end

```

## APPENDIX B

```
PID CODE
function y = matlab_sim_project(K)
% clc
% clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% define model
Ts = 0.01;
J  = 0.01;
b  = 0.1;
Ke = 0.01;
Kt = 0.01;
R  = 1;
L  = 1;
tFinal      = 30;
tStepMax    = 0.05;
theta0      = 3*pi/180;
thethaDot0  = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Kp  = K(1)
Ki  = K(2)
Kd  = K(3)
Kp1 = K(4)
Ki1 = K(5)
Kd1 = K(6)

K = [211 13 1 199 10 1];
assignin('base','K',K);
global Kp
global Ki
global Kd
global Kp1
```

```

global Ki1
global Kd1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Run the Simulink
model using sim command
options = simset('SrcWorkspace','current');
sim('SimModel.slx',[],options)
%sim('SimModel')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% extract the data
generated bythe similink model
t      = simX.Time;
t1     = simX1.Time;
theta  = simX.Data(:,1);
theta1 = simX1.Data(:,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y= 0:pi/100:2*pi;
open_system('SimModel') %optional

[tout,xout,yout]=sim('SIMPID2',[0 10],options);
plot(yout(1:end,1))
Y = simX(length(simX));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot

%figure
plot(t,theta,'LineWidth',2,'MarkerSize',14)
drawnow
hold on
plot(t1,theta1,'LineWidth',2,'MarkerSize',14)
drawnow
xlabel('t (seconds)')
ylabel('x')
grid on

```

```

title('Step response')
legend('output','output1')
end

function E=itaecost(p,dt,tf)
i=numel(p);
A=[zeros(i,1) eye(i);-1 -p];
B=[zeros(i,1);1];
A=expm([A B;zeros(1,i+2)]*dt);
x=[zeros(i+1,1);1];
E=0;
for t=0:dt:tf
    tdt=t*dt;
    x=A*x;
    e=1-x(1);
    E=E+abs(e)*tdt;
end
end

```



**OTHER PID CODE**

```
function [J,J1] = PID_problem(contr)

s = tf('s');

% % G11= 1/(s^2 + 20*s + 15);
% % G12= 1/(s^2 + 25*s +17);
% % G21= 1/(s^2 + 21*s +16);
% % G22= 1/(s^2 + 25*s +21);

%pid controllers
K = contr(1) + contr(2)/s + contr(3)*s/(1+.001*s);
K1 = contr(4) + contr(5)/s + contr(6)*s/(1+.001*s);

G=feedback(G11);
Loop = series(K,G);
ClosedLoop = feedback(Loop,1);
%step(feedback(G*K,1));
%K = Kp + Ki/s + Kd * s;

G1=feedback(G12,G22);
Loop1 = series(K1,G1);
ClosedLoop1 = feedback(Loop1,1);
%step(feedback(G1*K,1));
dt = 0.01;
t = 0:dt:1;
```

```

%step(sys);

t = 0:dt:20;
[y,t] = step(ClosedLoop,t);
CTRLtf = K/(1+K*G);
u = lsim(CTRLtf,1-y,t);

[y1,t] = step(ClosedLoop1,t);
CTRLtf1 = K1/(1+K1*G1);
u1 = lsim(CTRLtf1,1-y1,t);
Q = 1;
R = .001;

%e = 1 - step(feedback(G*K,1),t);
%J=sum(t'.*abs(e)*dt);
J = dt*sum(Q*(1-y(:)).^2+R*u(:).^2)
J1 = dt*sum(Q*(1-y1(:)).^2+R*u1(:).^2)
[y,t] = step(ClosedLoop,t);
[y1,t] = step(ClosedLoop1,t);

plot(t,y,'LineWidth',2,'color','r')
drawnow

Kp=contr(1)
Kd=contr(2)
end

```

**UNISA CSET ETHICS REVIEW COMMITTEE**

15/January/2021

Dear Mr Thamsanqa Bongani Nkwanyana

ERC Reference #: 2021/CSET/SOE/009

Name: Mr Thamsanqa Bongani  
Nkwanyana

Student #: 49435248

Staff #: N/A

**Decision:**  
**Research Ethics Notification  
Letter**

**Researcher(s):** Name: Mr Thamsanqa Bongani Nkwanyana

Email: 49435248@mylife.unisa.ac.za

Telephone: 0786843874

**Supervisor (s):** Name: Prof Zenghui Wang

Email: wangz@unisa.ac.za

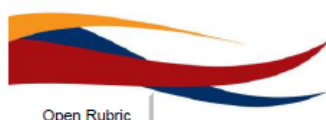
Telephone: +27 715460246

**Working title of research:**

**Multi-Input Multi-Output PID controller Tuning based on Particles Swarm  
Optimization**


**Qualification:** MTECH

This letter serves as notification that the research undertaken in the dissertation entitled: "Multi-Input Multi-Output PID controller tuning based on Particle Swarm Optimization" was conducted in the absence of an approved research ethics certificate. The research study was conducted from 2018 to 2020. The research study did not involve fieldwork where humans were involved, and no secondary data, animals or plants were involved in the study. The research study is regarded as a negligible risk (risk 1) study.



According to policy, the university cannot grant research ethical clearance retrospectively. The researcher submitted his dissertation together with a completed ethics application form to the CSET ERC for review to establish if the researcher acted in accordance with the ethical standards set out in the Policy on Research Ethics. This letter confirms that the dissertation adheres to the minimum research ethics requirements relating to a negligible risk study.

Yours sincerely,




---

Prof E Onyari-Benecha  
Chair of School of Engineering Ethics Review Subcommittee  
College of Science, Engineering and Technology (CSET)  
E-mail: onyarek@unisa.ac.za  
Tel: (011) 471-3379



---

Prof. F Nemavhola  
Director: School of Engineering  
College of Science Engineering and  
Technology (CSET)  
E-mail: masitfj@unisa.ac.za  
Tel: (011) 471 2354



---

Prof. B Mamba  
Executive Dean  
College of Science Engineering and  
Technology (CSET)  
E-mail: mambabb@unisa.ac.za  
Tel: (011) 670 9230



University of South Africa  
Preller Street, Muckleneuk Ridge, City of Tshwane  
PO Box 392 UNISA 0003 South Africa  
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150  
[www.unisa.ac.za](http://www.unisa.ac.za)