TECHNICAL AND VOCATIONAL TRAINING INSTITUTE (TVTI)

School of Graduate Studies

FACULTY OF ELECTRICAL AND ELECTRONICS TECHNOLOGY AND INFORMATION AND COMMUNICATION TECHNOLOGY (DEPARTMENT OF ELECTRICAL AND ELECTRONICS TECHNOLOGY)

Tuning of improved form of PID controller for time delayed unstable processes using PSO and PSOGSA optimization algorithm

MSc Thesis for the Partial Fulfillment of
Master of Science in Electrical Automation and Control Technology Management

By,

JEMAL HABIB ID: (MTR/193/12)

Supervisor,

Dr. Petchinathan Govindan

Assistant Professor

AUGUST 2022
Addis Ababa, Ethiopia

Tuning of improved form of PID controller for time delayed unstable processes using PSO and PSOGSA optimization algorithm

A Thesis submitted to
TECHNICAL AND VOCATIONAL TRAINING INSTITUTE (TVTI)
FACULTY OF ELECTRICAL AND ELECTRONICS TECHNOLOGY AND INFORMATION AND COMMUNICATION TECHNOLOGY
(DEPARTMENT OF ELECTRICAL AND ELECTRONICS TECHNOLOGY)
In partial fulfillment for the Degree
MASTER OF SCIENCE *in* ELECTRICAL AUTOMATION AND CONTROL TECHNOLOGY MANAGEMENT

By,

JEMAL HABIB ID: (MTR/193/12)

Supervisor,

Dr. Petchinathan Govindan

Assistant Professor

DECLARATION

I hereby declare that the work which is being presented in this thesis entitled **“**Tuning of improved form of PID controller for time delayed unstable processes using PSO and PSOGSA optimization algorithm” is the original work of my own, has not been presented for a master’s thesis in this or other universities and all sources of materials used for this thesis work have been fully acknowledged.

Name: Jemal Habib   ID (MTR/193/12)

Signature: ________________________

Place: Addis Ababa

Date of Submission: ________________

This thesis proposal has been submitted for examination with my approval as a TVTI advisor.

Dr. Petchinathan Govindan	----------------	--------------
Advisor Name	Signature	Date

TECHNICAL AND VOCATIONAL TRAINING INSTITUTE (TVTI)
FACULTY OF ELECTRICAL AND ELECTRONICS TECHNOLOGY AND INFORMATION AND COMMUNICATION TECHNOLOGY
(DEPARTMENT OF ELECTRICAL AND ELECTRONICS TECHNOLOGY)

Thesis on

Tuning of improved form of PID controller for time delayed unstable processes using PSO and PSOGSA optimization algorithm

By,

Name : Jemal Habib ID:(MTR/193/12)

APPROVED BY THESIS ADVISOR COMMITTEE

Name of Examiner External Signature Date

-- ------------------------- -------------------

Name of Examiner, Internal Signature Date

-- ------------------------- -------------------

Name of Examiner, Internal Signature Date

-- ------------------------- -------------------

Name of the Advisor Signature Date

------------------------------ ------------------------- -------------------

Name of Chairperson Signature Date

-- ------------------------- ----------------

ACKNOWLEDGMENT

I want to express my sincere appreciation to Dr. Petchinathan Govindan, my advisor, who worked diligently with me to organize and perfect my dissertation. He has provided me with so much support during the course of my thesis work, and without him, I would not have been able to finish it. I sincerely appreciate you taking the time to go over my work several times.

ABSTRACT

Time delayed and unstable processes are frequently encountered in industrial and chemical plants, such as heating boilers, liquid storing tanks and batch chemical reactors. Many different approaches have been developed to control such processes. A number of systems utilize proportional–integral (PI) or proportional–integral–derivative (PID) controller methods based on the classical unity feedback closed loop structure for time delayed unstable processes. But those are sometimes not acceptable in cases unstable system. To avoid this problem, the form of PSO (particle swarm optimization) and PSOGSA (particle swarm gravities) Algorithm based I-PD controller is applied for the control of Time delayed unstable processes. In this work the Integral Square Time Error (ISTE) is considered as an objective function for the optimization problem. The performance of PSO and PSOGSA tuned I-PD controller is compared with conventional I-PD controller for the control of three different time delayed unstable processes (model). From the graphical and numerical results, it can be concluded that the PSOGSA tuned I-PD controller is performed well than PSO I-PD and conventional I-PD controller.

Keywords; *PID controller, I-PD controller, time delay unstable process, PSO, PSOGSA, ISTE*

Contents

List of Figures

List of Tables

ABBREVATIONS

PID	proportional- integral-derivative
I-PD	Integral-proportional-derivative
PSO	particle swarm optimization
PSOGSA	particle swarm optimization gravitational search Algorithm
ISTE	integral time square error TDUP time delay unstable process
Z-N	Ziegler- Nichols
KC	proportional gain
TI	integral gain
TD	derivative gain
DS	direct synthesis
IMC	internal model control
UFODT	unstable first order delay time
L	delay time
T	tou
K	proportional constant

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Industrial and chemical processes, including batch processing, liquid storage tanks, and heating boilers integrating and unstable processes are frequent in chemical reactor. Systems that use the common unity feedback closed loop controller technique (PI) or the proportional-integral-derivative method have been implemented using a number of ways (PID). PID controllers are effective because of their straightforward structure, reliable performance, and wide range of control applications designed to manage these processes. to control integrating as well as unstable processes, Therefore, PID controller design is still a topic of research and has a fantastic library of PID controller design techniques. There is a great collection of PID controller design techniques that can be found in [1], [2]. In the literature, several design strategies for fine-tuning PID controllers can be identified, such as phase- and gain-margin-based design techniques.[18] model-based internal design techniques [2],[10] [15], direct synthesis method [14] and The optimization methodology is another popular PID design method. [4], [8]. Direct optimization is one strategy of applying optimization techniques. However, this is of limited utility because the design must be repeated each time the plant transfer function changes. Another technique is to assume a plant transfer function model and optimize it to derive analytical tuning guidelines. Zhuang [5] Tuning criteria for a PID controller were derived by minimizing the time moment weighted integral performance criterion while assuming a steady first order plus dead time plant transfer function. Visioli [14] carried comparable calculations based on integral performance indices in order to develop an ideal PID controller parameter for processes with an integrator and an unstable plant transfer function Kaya found the best PI and PID controller settings for a stable first order dead time delay using the Smith predictor structure [10]. Tuning formulae for PI/PID controllers for pure integrating plus dead time, integrating plus first order plus dead time, and double integrating plus dead time processes were provided by Ali and Majhi [8]. Kaya [1][2]recently proposed the best analytical PI/PID tuning rules for regulating stable and integrating systems with time delay and inverse response. All of the preceding researches use an optimization method to provide analytical tuning rules for traditional PI/PID controllers based on integral performance requirements. However, due to PID controller structure restrictions and Processes with poorly placed poles are obtained [16]. As a result, various controller structures have been developed in order to increase the closed loop performance of the aforementioned processes.

The purpose of this study is to Compare and analyzes the performance of the I-PD controller , PSO and PSOGSA tuned I-PD controller for the control of Time delay Unstable Processes (TDUP).The simulation results have been presented to demonstrate the use of the suggested I-PD controller design technique.

1.2 Problem of statement

The time delay unstable processes are frequently encountered in process industries, such as heating boilers, liquid storing tanks and batch chemical reactors. Many different controllers have been proposed for the control of time delay unstable processes such as proportional–integral (PI) or proportional–integral–derivative (PID) controller methods based on the classical unity feedback closed loop and time weighted square error structure. In that situation, the set point response tends to be accompanied by excessive overshoot and a large settling time. To improved or avoid large settling time and excessive overshoot PSO and PSOGSA tuned I-PD controller plied for the control of Time delay Unstable Processes.

1.3. Objectives

1.3.1 General Objectives

- To Compare and analyzes the performance of the I-PD controller and PSO and PSOGSA tuned I-PD controller for the control of Time delay Unstable Processes (TDUP)

1.3.2 Specific Objectives

- To simulate the mathematical model of three time delayed unstablesystem using MAT LAB/simulink software.
- To apply the I-PD controller for the control of TDU processes.
- To tune the parameters of improved form of I-PD controller using PSO optimization

algorithm.

- To compare and analyze the performance of conventional I-PD and PSO and PSOGSA tuned I-PD controller.

1.4. The scope and limitation of this thesis work

- To learn and getting exposure in design and analysis of improved or I-PD controller for the control of unstable processes using MATLAB simulation
- To learn about application of optimization algorithms for tuning of controller parameters for obtaining better performance of the controller
- The findings of this thesis' are limited to the study of MATLAB simulation. It is not precisely applicable for real-time application because certain assumption and linearization procedures were considered for obtaining a mathematical model TDUP system.
- PSOGSA =PSO (particle swarm optimization) +GSA (gravitational search algorithm).

1.5 Significance of thesis work

The functioning of the process as a whole may be impacted by an unstable process that exceeds the required setpoint because it can cause equipment damage, impact product quality, alter chemical reactions, and damage to process machinery. In order to increase productivity, safety, and product quality, time delay unstable process is a must be improve. The time delay unstable process system is controlled by an I-PD controller based on a PSOGSA optimization algorithm in this thesis study.

CHAPTE TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This process of time delay unstable system, some of the related literature reviews of improved form of PID controller for time delayed unstable processes system at chemical plant are discussed. The work carried out by the various authors related to time delay unstable system, controller tuning methods and implementation procedures have been discussed in detail.

Onat, Cem (2019), Kaya and Atherton proposed a straight forward method for fine-tuning the settings of a PI-PD controller for control of integrating and unstable systems. They employed tuning equations for the ISE (Integral of Squared Error) and ISTE (Integral of Squared Time weighted Error). The proposed method draws stability boundary loci in a parameter plane that are dependent on controller and frequency parameters[1].

Rajinikanth et.al (2012), had presented the fundamental controller design approach, including open-loop unstable, integrating, and stable processes in the main loop and open-loop unstable, integrating, and stable processes in the secondary loop. Using a frequency domain direct synthesis (DS) approach to reject load disturbances and set points tracking. be improved by selecting a suitable range of ‘ζ .but the effect of this paper proportional and derivative kick problems [2].

Kumar, and Munna Prasad, (2020), had suggested an IMC-PID controller for a second-order time delay system that was unstable. ITAE and total variation (TV) data were analyzed and compared to newly release tuning criteria. By using setting a weighting parameter to a fixed value, an overshoot was minimize[3].

T. Liu, et.al (2005),had proposed Set-point weighting method to improve proportional-integral-derivative controller in series with a lead-lag filter is designed for control of the open-loop unstable processes with time delay based on direct synthesis method. Set-point weighting is considered to reduce the undesirable overshoot[4].

Zheng, et al (2013**)**, the researcher study automatic disturbance rejection controller (ADRC) the proposed technique its current limitations with plants that have a significant transportation delay. In this paper a predictive ADRC structure is proposed as a solution to the problem of disturbance rejection in industrial processes with transport delay. The limitation of ADRC is long standing weakness with transport delay. Simulation studies are carried out to show that the proposed structure is simple to use, intuitive to understand, and effective in dealing with delays while maintaining good transient response and robustness. Significant improvements over existing methods are observed[5].

Fu, CaifenTan, Active disturbance rejection control (ADRC) recognizes external and internal uncertainties as general disturbances and employs an extended state observer (ESO) to estimate them in real-time and feed them back into the control loop, achieving good disturbance rejection performance. However, owing to its fundamental structure, ADRC is not well suited for unstable delayed processes[6].

Chan Cheng Huang et. al (2008),have been proposed the set-point weighted PID control system method to control time delayed unstable process. Simulation results show the performance of the proposed set-point weighted PID control system can be effectively enhanced that of many other systems [7].

Ali, Ahmad et.al (2011), have proposed on IMC-based maximum sensitivity and IMC internal model control (UFOPDT) processes. Based on proportional-integral-derivative (PID) controllers .In order to achieve the appropriate degree of resilience in the controller parameters, mathematical tuning methods are devised. The advantages of the suggested analysis have been demonstrated by simulation experiments on a variety of UFOPDT procedures[8].

.Singer, et. al. (2020) had purposed this work is to look at employing a PD-PI controller for disturbance rejection in a third order unstable process. The four parameters of the PD-PI controller are set to allow efficient rejection of a step disturbance input. It is implemented controller tuning based on the use of MATLAB control and optimization toolboxes with five error-based objective functions: ISE, IAE, ITAE and ITSE[9].

Atif Siddiqui et. al .(2021)had proposed Open-loop unstable, integrating and stable processes in the primary loop and open-loop unstable, integrating and stable processes .In the secondary loop of the basic cascade control configuration design technique. Using a direct synthesis (DS) technique in the frequency domain. As a result, order reduction and plant approximation dead time are avoided. Simulations show that the current method gives better closed-loop performance than other methods described in the literature recently[10].

Jin, Q. B., Q. (2014), has proposed A two-degree-of-freedom (2-DOF) PID controller tuning approach employing an upgraded internal model control (IMC) principle is presented. Use Analytical PID tuning guidelines for process integration are discussed. A broad variety of integrating processes can benefit from the presented PID settings. The proposed approach's effectiveness and benefits have been demonstrated through simulations[11].

Raja, et.al (2017),. The suggested PCCS controller settings are determined by equating the first and second derivatives of planned and actual closed-loop transfer functions about the origin of the s- plane, the Routh Hurwitz stability criterion, and the internal model control technique. To assist the user in obtaining the closed-loop time constants, analytical expressions and guidelines are provided. As a result, the suggested strategy outperforms previously published tuning strategies in which the authors advocate a range of values for the closed-loop time constants. The suggested tuning technique improves and stabilizes closed-loop performance for all first order plus time delay secondary process models, integral plus time delay, and second order secondary process models[12]

A new technique for decreasing the destabilizing effects of the time-delay parameter in control loops is introduced and investigated in this paper. The concept of the approach is developed from a knowledge of the dynamic behavior of irrational transfer functions (Ire-TF) in the frequency response domain and. The precondition for the creation of the aforementioned dual phase features may lead one to design a control loop that eliminates the non-minimum phase dynamics of the open-loop transfer function. These capabilities include the ability to regulate processes with an irrational transfer function model as well as integrated processes with a time-delay parameter[13]

Kaya 2018, has been suggested optimal I-PD tuning formulas have been presented in this paper. The suggested optimal I-PD design approach's closed loop performance has been shown through comparisons with existing PID and I-PD design techniques for regulating integrating processes. Controlling integrating processes makes use of controllers. The development of optimum and analytical tuning processes has been done in order to determine the I-PD controller's tuning parameters [14]

Antonio et.al 2004. The usage of clever methods for fine-tuning Proportional-Integral-Derivative (PID) controllers has increased in recent years. Due to their adaptability, evolutionary techniques have achieved a significant position. The automated tuning of systems with stable and unstable system. The advantages of the suggested method were demonstrated by comparison with the Visioli genetic method and the Ziegler-Nichols modified closed loop method. The objective of the suggested methodology is to extend the scope of the intelligent tuning application to more processes (covering systems with oscillatory or unstable modes) [15]
ET. al., G. Sungari (2021), has proposed the metaherustic controllers like GA might be used to tune PID controllers under various Nonlinearities. Such as sinusoidal and saw-toothed noise, using the excellent closed-loop response of a second order system. The suggested strategy is evaluated for its efficacy in comparison to more traditional approaches[16]

Darwish and Noah (2016)) had proposed proportional–integral–derivative (PID) controller design method for stable and integrating time-delay systems with and without non- minimum phase zero (inverse response) using the direct method is proposed. The reference model is chosen to satisfy the desired maximum sensitivity Ms. As a result, three linear algebraic equations in three unknowns are obtained and the solution of them gives the PID controller gains. The proposed method can be applied to low- and high-order systems, and the Pade approximation of the time-delay term e^{-2ls} is not required[17]

Karan, et. al (2021), has proposed the IMC tuning guideline for servo tracking is used to acquire the parameters of the forward route PD controller, while Routh stability analysis is used to realize the feedback path PD controller in order to optimize regulatory responses. the suggested approach is superior to others' reported dead-time compensation strategies[18]

2.2. LITRATURE REVIEW SUMMERY

Various and (CCC) cascade control configuration have been used to design and analyze the performance of controllers for control of Time delayed unstable process. The control of time delayed unstable process is difficult because of excessive over shoot, a large settling time. Because of these reasons, advanced control methods are required for design of controller to achieve good performance in the process. Advanced and improved form of PID controllers techniques, such as PID controller, DS, PI-PD controller, PI controller, IMC-PID, Ant lion optimizer (ALO), proposed Set-point weighting method, ADRC, controller, (2DOF) control mechanism, Ziegler-Nichols controller,(PD)control is required for improving the performance of the overall control system. So in this thesis the I-PD controller is proposed for the control of time delayed unstable process. To tune the parameters of I-PD controller, the PSO and PSOGSA optimization algorithm will be employed after finalizing in this thesis work.

CHAPTER THREE

MATHEMATICAL MODEL OF TIME DELAYED UNSTABLE OF PROCESSES

3.1 Introduction

Time Delayed Unstable Processes (TDUP), which includes bioreactors, polymerization reactors, coupled feed/effluent heat exchangers with adiabatic exothermic reactions, exothermic stirred reactors with back mixing, and pumps with liquid storage tanks, are frequently used in the chemical process industries. For these systems, it is more challenging to fine-tune the controller settings than it is for open loop stable systems. A minimum and maximum value depending on the process time delay limits controller gains .because unstable processes are challenging due to unstable poles. Additionally, this effort hopes to improve unstable systems' closed-loop performance by utilizing the (particle swarm optimization) PSO and (particle swarm gravitational) PSOGSA algorithms, as well as an I-PD controller. In this work, three different FODUP the process models are considered for investigating the performance of proposed (particle swarm optimization) PSO and (particle swarm gravitational) PSOGSA algorithms tuned I-PD controller.

3.2 Mathematical models of Time delayed unstable processes:

The following three unstable First order plus dead time (UFOPDT) models of unstable systems are considered for investigating the performance of proposed PSO and PSOGSA tuned I-PD controller. The following UFOPDT models are obtained from various literature.

Example 1

According to the literature[12] (model of first order time delay unstable system)

$$G_1(s) = \frac{1}{20s-1}e^{-4s} \qquad 3.1$$

The pole is 20s-1=0

S=1/20

Example 2

According to the literature [2](model of first order time delay unstable system)

$$G_2(s) = \frac{4}{4s-1}e^{-2s} \tag{3.2}$$

The pole is 4s-1=0

S=1/4

Example 3

According to the literature[19](model of first order time delay unstable system)

$$G_3(s) = \frac{e^{-0.4s}}{S-1} \tag{3.3}$$

The pole is S-1=0

S=1

CHAPTER 4

4.1 CONTROLLER DESIGN

In this section the theoretical concept of I-PD controllers ,particle swarm optimization (PSO) and the particle swarm gravitational search Optimization algorithm(PSOGSA), as well as several tuning approaches for I-PD controller settings [20].

These simulation examples describe how to use the suggested I-PD controller design method. Comparisons are taken in every instance with the design methods PSO and PSOGSA Optimization algorithm. In this work the controller parameters on tuned based on the ISTE criteria.

4.2 Implementation step for the thesis work

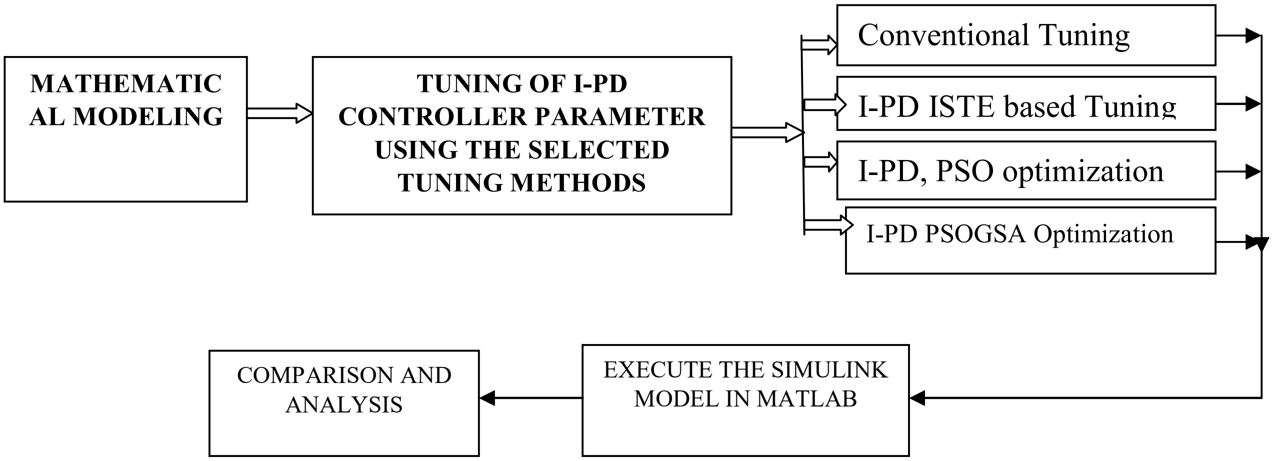

Figure4. 1: Flow chart Representation of this thesis work

4.3. I-PD CONTROLLER DESIGN

Figure 4.2 shows the I-PD controller structure. The transfer function of the unstable process is shown in the picture as G(S). I and PD controller transfer functions, respectively, are represented by $G_{c1}(s)$ and $G_{c2}(s)$ [1].

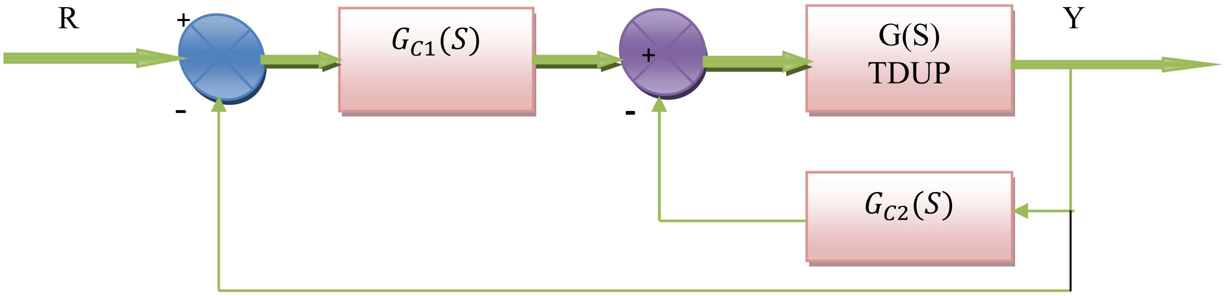

Figure4.2: block diagram of I-PD controller

The transfer function of Integral controller is given by

$$G_{C1}(\mathrm{S}) = \frac{K1e^{-\theta 1s}}{s}$$

4.3

The transfer function of Proportional and derivative controller is given by

$$G_{C2}(\mathrm{S}) = \frac{K2e^{-\theta 2s}}{KC(1+Tds)} \tag{4.4}$$

The error function in figure is given by $\mathrm{E(S)} = \frac{R(S)}{1+G(s)[G_{c1}(s)+G_{c2}(s)]}$

$$\mathrm{G\,(s)} = \frac{Ke^{-\theta s}}{(Ts-1)} \quad . \tag{4.5}$$

The tuning rules for the I-PD controller based on Kaya and Atherton equation is given by

$$\mathrm{KKc} = \frac{212.4-71.55\left(\frac{\theta}{T}\right)+24.76\left(\frac{\theta}{T}\right)^2}{[0.0009459+148.8\left(\frac{\theta}{T}\right)-34.42\left(\frac{\theta}{T}\right)^2+\left(\frac{\theta}{T}\right)^3]} \tag{4.6}$$

$$\mathrm{Ti/T} = \left[\frac{(-1\ .94+685.8\left(\frac{\theta}{T}\right)+837.1\,.\left(\frac{\theta}{T}\right)^2+312.3\left(\frac{\theta}{T}\right)^3-75.49\left(\frac{\theta}{T}\right)^4)}{(262.8+440.3\left(\frac{\theta}{T}\right)-721.7\left(\frac{\theta}{T}\right)^2+220.1\left(\frac{\theta}{T}\right)^3+\left(\frac{\theta}{T}\right)^4}\right] \tag{4.7}$$

$$\mathrm{Td/T} = -0.0002104+0.4259\left(\frac{\theta}{T}\right)+0.05089\left(\frac{\theta}{T}\right)^2-0.01274\left(\frac{\theta}{T}\right)^3] \tag{4.8}$$

From equations. 4.6, 4.7 and 4.8 the parameters of I-PD controller is calculated for Example1, Example 2 and Example 3. The calculated values are recorded in table 4.3.

Table 4.1: conventional I-PD controller parameter

Conventional I-PD controller parameters	Example1	Example2	Example3
kc	7.0118	0.6933	3.4717
Ti	9.8891	6.9501	1.2246
Td	1.7381	0.8955	0.1775

4.4 PSO based tuning method

The evolutionary algorithm-based tuning approaches are often used to get the optimal tuning of controller settings. One sort of evolutionary algorithm is the PSO algorithm. Kennedy and Eberhart developed the PSO algorithm in 1995. Particle swarm optimization technique improves search performance with steady and faster convergence rate when compared to subsequent agent-based stochastic43 optimization systems. Similarly, as compared to other nature-inspired algorithms in the literature, the number of initial algorithm parameters to be specified is limited. The PSO algorithm flow chart is depicted in the Figure below[21].

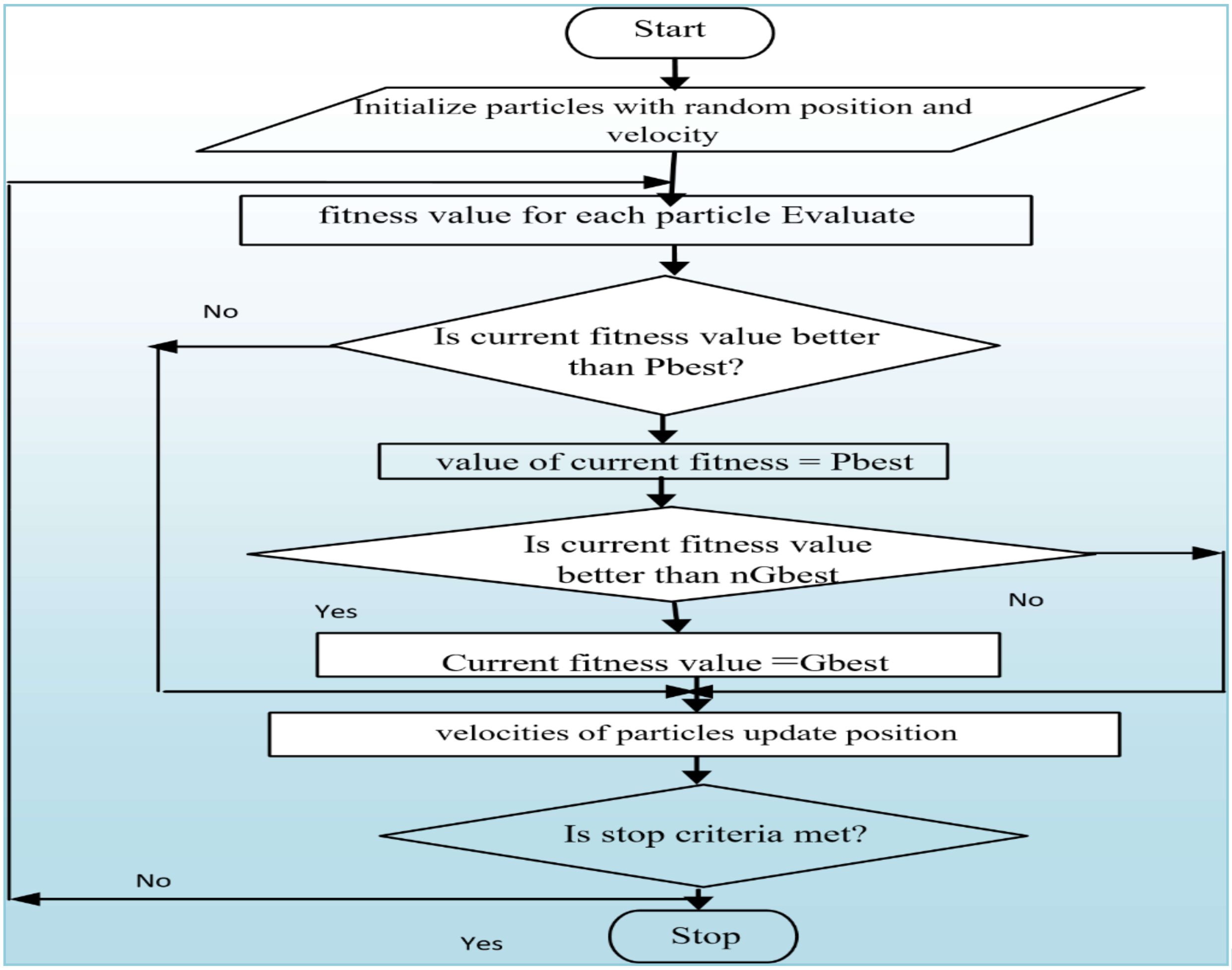

Figure 4.3: Flow chart on evolution of PSO algorithm

4.5. PSOGSA Optimization Algorithm

Particle Swarm Optimization with Gravitational Search Algorithm (PSOGSA) is a revolutionary population-based hybrid approach (GSA). In order to maximize the potential of both algorithms the main concept is to combine the exploitation power of PSO with the exploration capability of GSA. Certain benchmark test functions are employed to compare the hybrid approach to both the conventional PSO and GSA algorithms in order to find optimal answer. The outcomes show hybrid algorithm is quicker at escaping from local optimums than the standard PSO and GSA. In the updating process, PSOGSA takes the solutions' quality (fitness) into account. Agents that are close to effective solutions try to get other agents to look into the search area. Agents move slowly when they are all very near to a good answer. In this case, the gBest helps them use the world's best. The best solution found thus far is stored in a memory (gBest) by PSOGSA, making it always accessible[20].

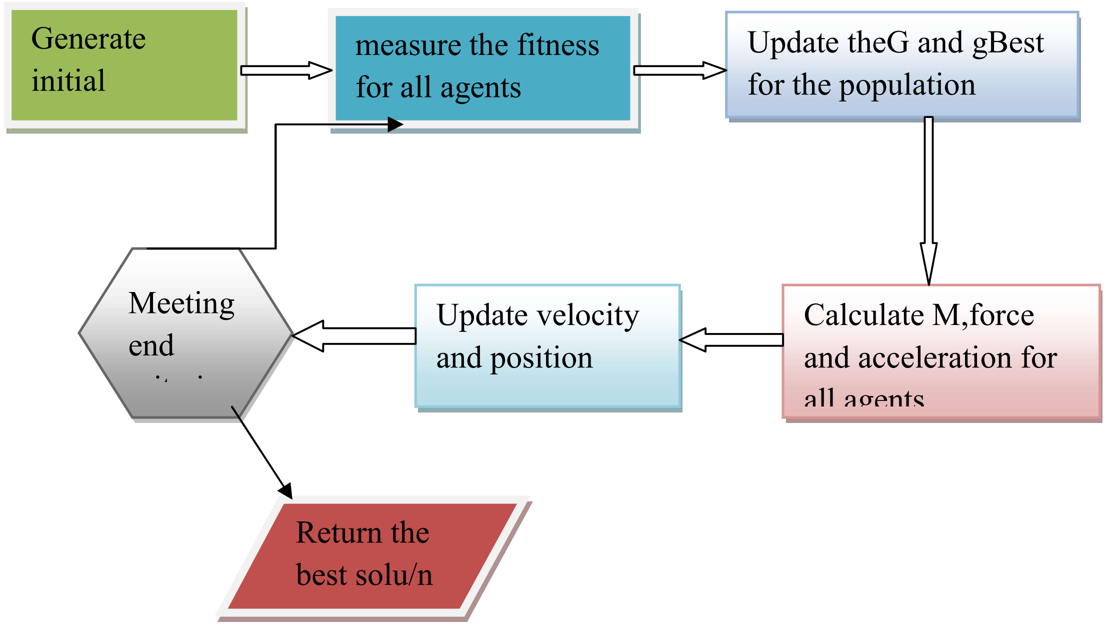

Figure 4.4: Flow chart of PSOGSA Algorithm

4.6 Objective Functions

The formulation of the objective function that directs the algorithms of optimization to determines the optimal values of controller parameters. In this thesis work, the optimization problem is addressed with the goal of minimizing the function of objective. The optimization problem is studied using objective functions like ISTE.

ISTE's equations are provided by

Time Square Integral Error ISTE= $\int t\,[e(t)^2].\,dt$

Table 4.2 Constraints and Boundary of I-PD controller PSO and PSOGSA Algorithm

TDUPs	I-PD controller	kc	Ti	Td
Example 1	Boundary	0-15	0-20	0-5
	Constraints	OS < 10 Ts<31		
Example 2	Boundary	0 to 2	0 to 18	0 to 3
	Constraints	OS < 10 Ts<17		
Example 3	Boundary	0-9	0-5	0-2
	Constraints	OS < 8 Ts<3		

Table: 4.3: PSO and PSOGSA parameters

Parameters	PSO	PSOGSA
Maximum number of iteration	100	100
Population size	50	50
weight of inertia	1	-
Damping Ratio	0.99	-
c1	1.5	-
c2	2.0	-
Dimension	3	3

CHAPTER FIVE

SIMULATION RESULT AND DISCUSSION

In this chapter, the PID and the PSO and PSOGSA based I-PD controllers for the control of the three different TDUPs are constructed, and investigated their performance. The performance of the PID and I-PD controllers are analyzed using graphical and numerical results obtained from the simulation studies. The PSO and PSOGSA optimization techniques are employed in this simulation research to fine-tune the I-PD controller's settings The MATLAB and Simulink software environments are used to carry out the entire simulation investigation.

5.1. Response of Time delay Unstable Processes in Open Loop

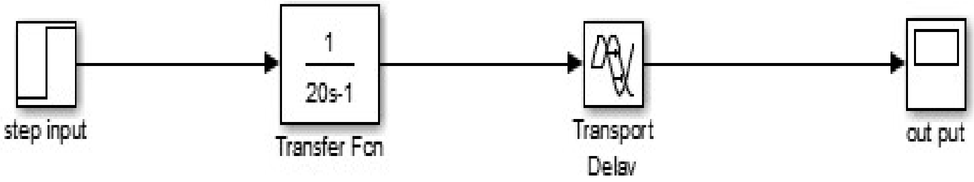

Figure:5.1 TDUP Open-loop Step Response Simulink diagram(Example1)

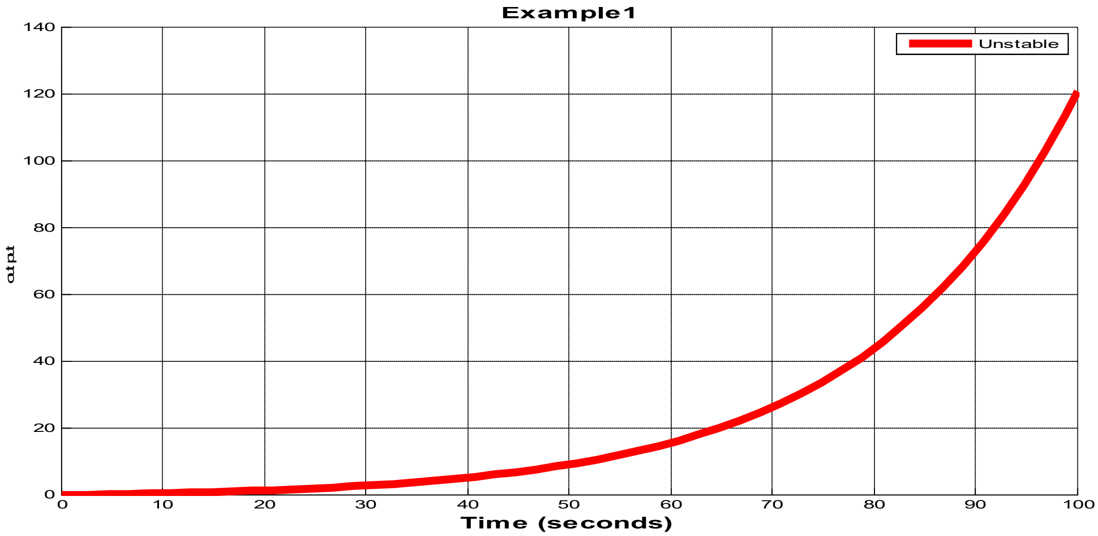

Figure:5. 2 : Step reaction of TDUP in Open-Loop (Example 1)

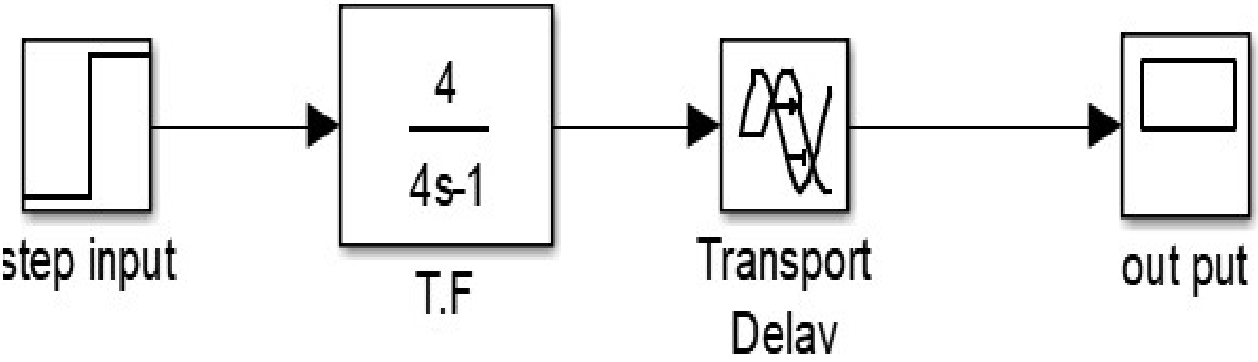

Figure 5. 3 TDUP Open-loop Step Response Simulink diagram (Example2)

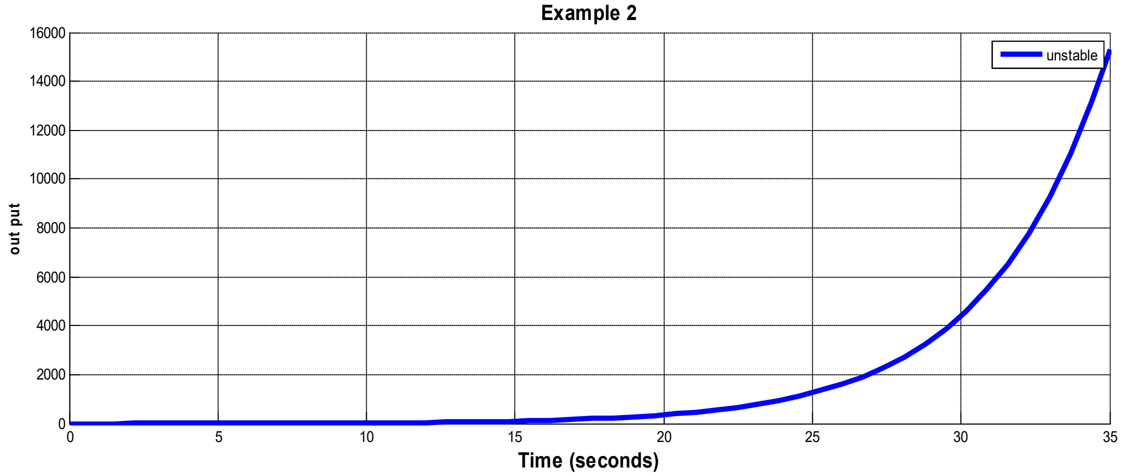

Figure:5. 4: Step reaction of TDUP in Open-Loop (Example2)

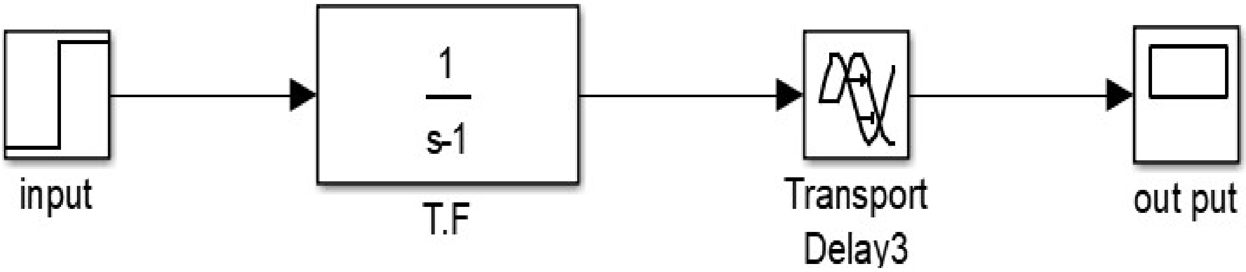

Figure 5. 5:TDUP open-loop step response simulink diagram (Example 3)

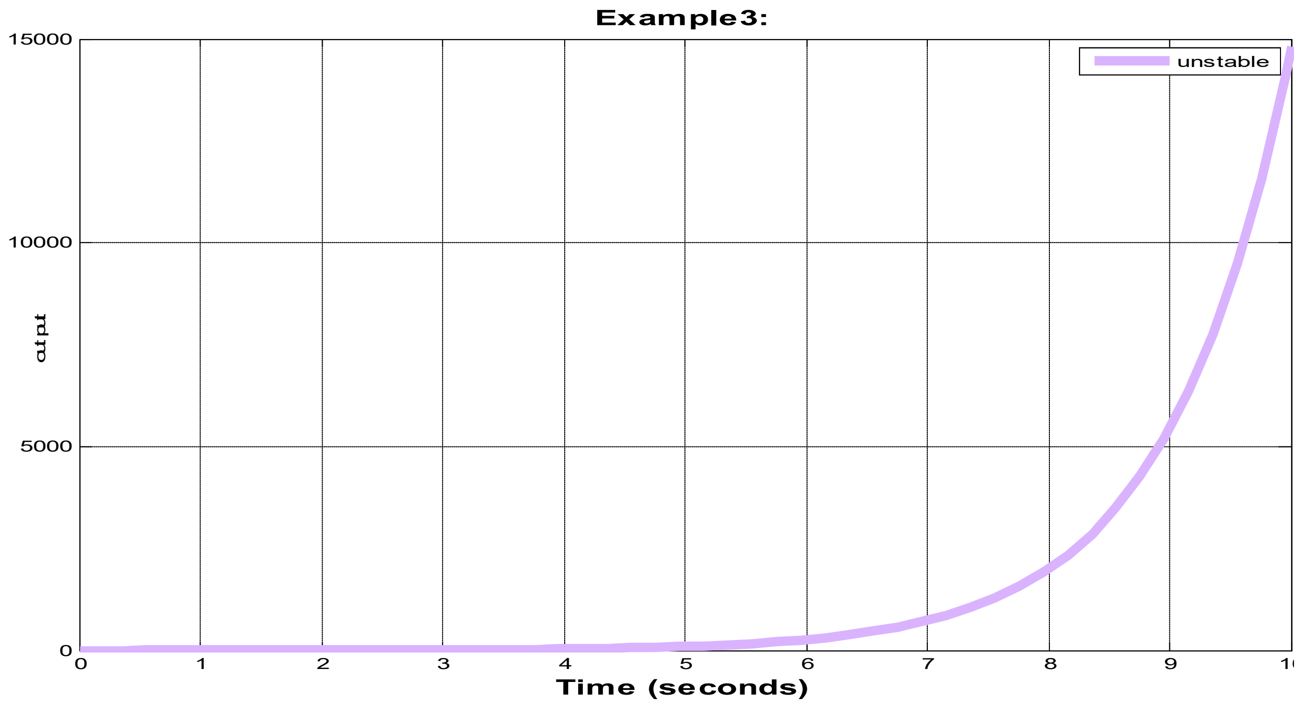

Figure 5. 6: Step reaction of TDUP in Open-Loop (Example 3)

The open loop responses of three distinct unstable first order plus dead time processes (UFOPDT) are examined in this section. The Simulink diagram of open loop UFOPDT processes is shown in figures 5.1, 5.3 and 5.5. The open loop response of UFOPDT processes is shown in figures 5.2, 5.4 and 5.6. Figures 5.2, 5.4 and 5.6 shows the diverged response. From this, it can be observed that the considered systems are unstable. The instability of the unstable systems makes them more challenging to stabilize than open loop stable systems, where the pole is on the right side of the S-plane and controller gains are constrained by a minimum and maximum value dependent on process time. In this work the I-PD controllers employed for improving the performance of closed loop control system.

5.2 Control of Time delay Unstable Processes using conventional I-PD controller

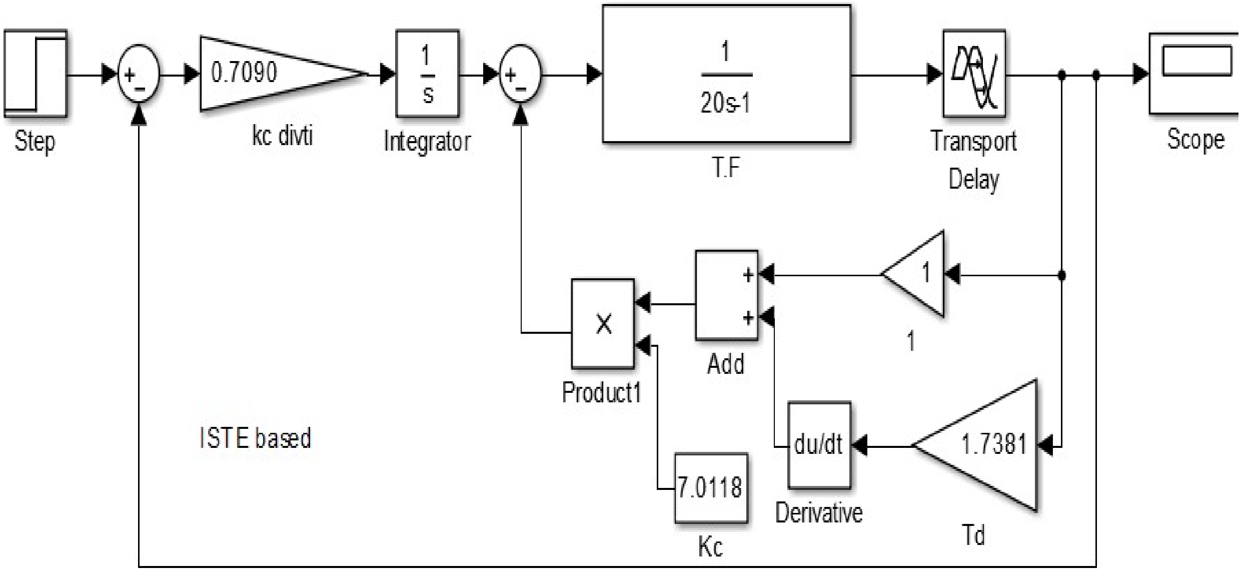

Figure: 5.7: Simulink diagram of control of TDUP (Example 1) using I-PD controller

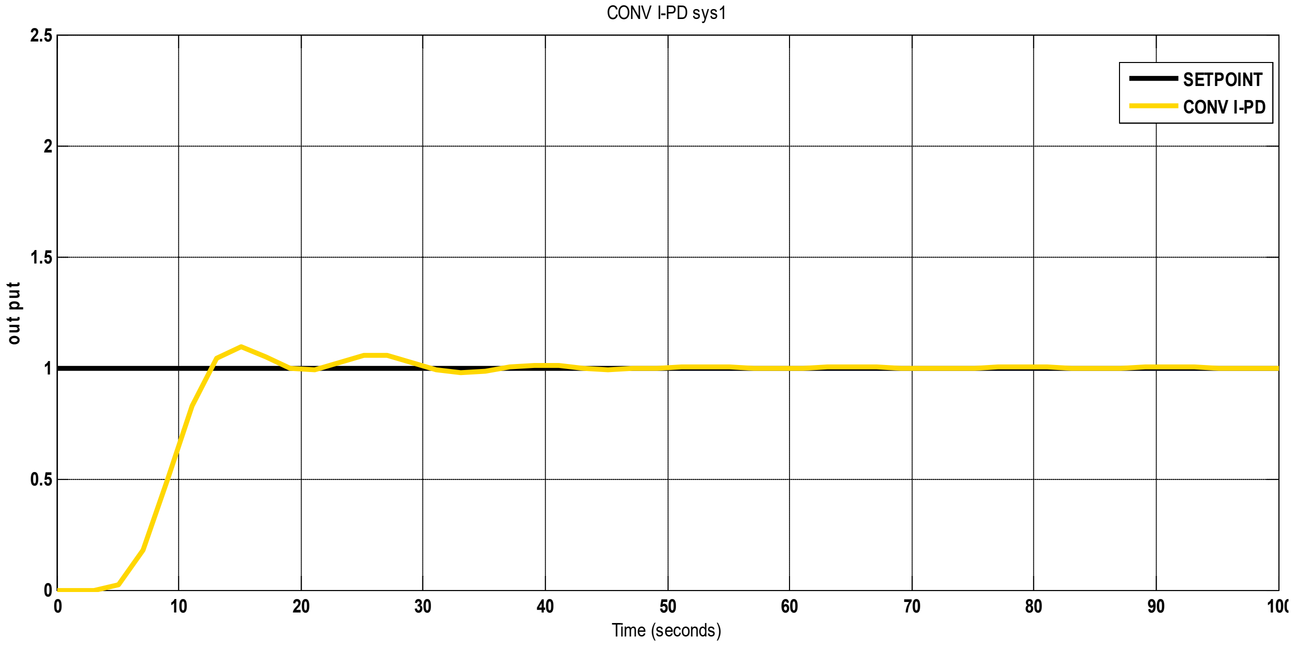

Figure:5.8:Output response of TDUP (Example 1) using I-PD Controller

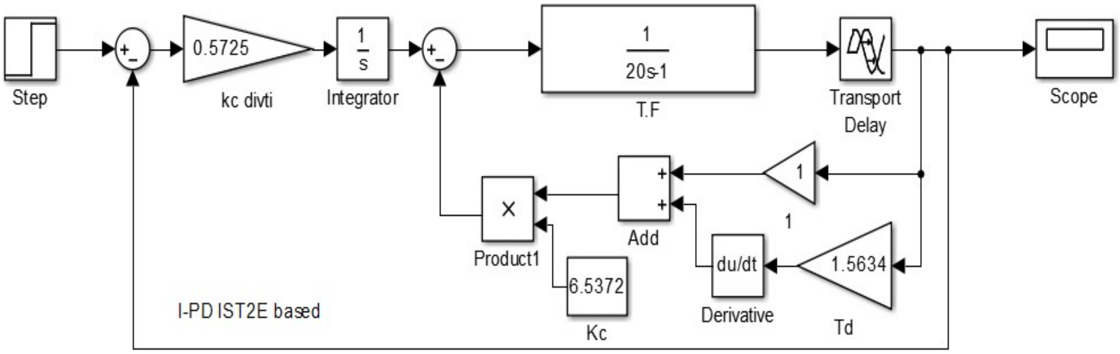

Figure:5.9: Simulink diagram of control of TDUP (Example 2) using I-PD controller

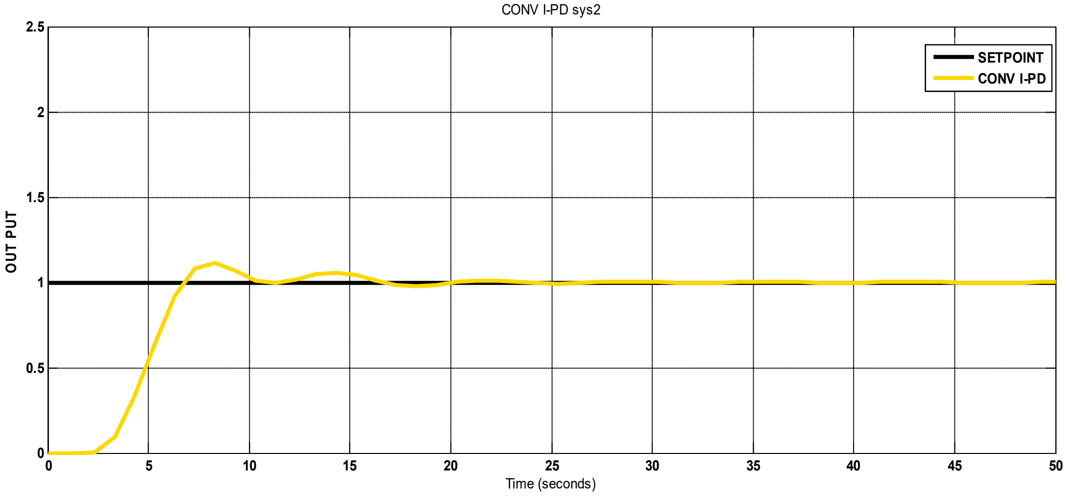

Figure:5.10: Output response of TDUP (Example2) using I-PD Controller

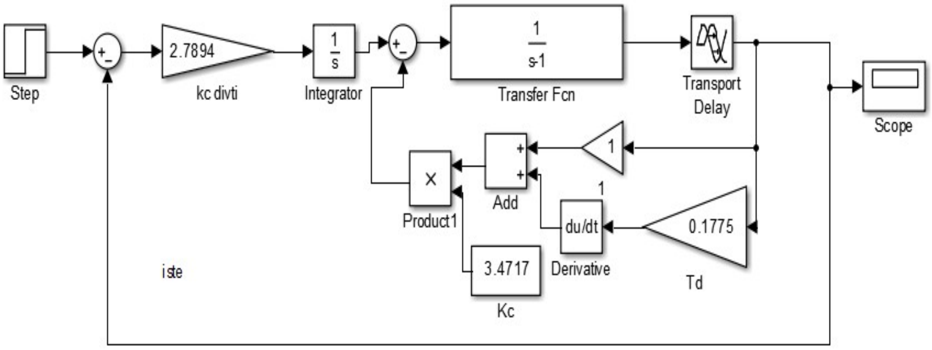

Figure 5. 11: Simulink diagram of control of TDUP (Example 3) using I-PD controller

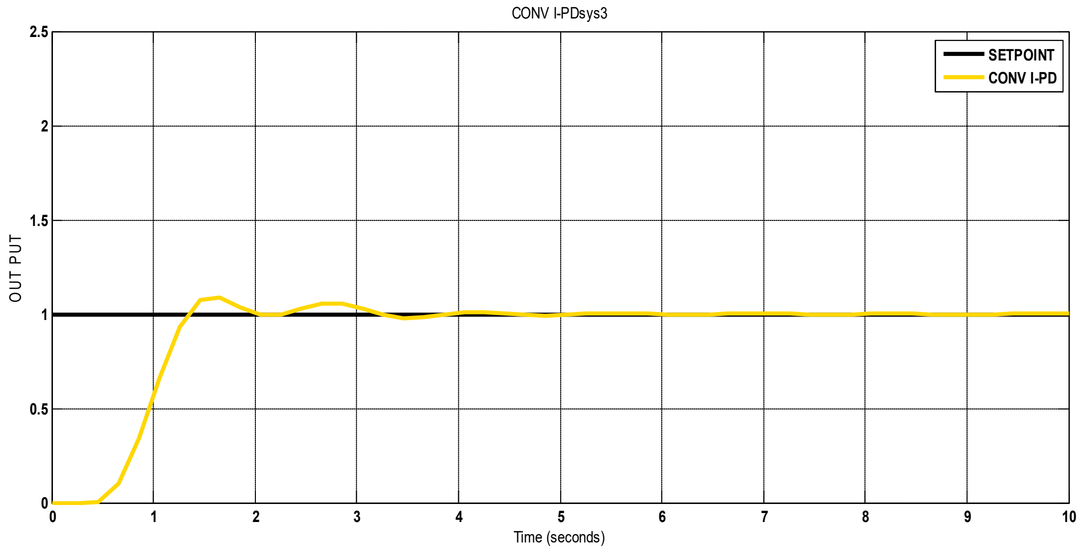

Figure:5. 12: Simulink diagram of control of TDUP (Example 3) using I-PD controller

Table: 5.1 Performance measure time response sp and parameters of the l I-PD controller

Controller Parameters and Performance measure	Conventional I-PD controller		
	Example 1	Example 2	Example 3
K_P	7.0118	0.6933	3.4717
K_i	9.8891	6.9501	1.2246
K_d	1.7381	0.8955	0.1775
Rise time (Tr) in sec	5.6791	2.9252	0.5780
Settling time (Ts) in sec	33.4340	18.6297	3.1103
% Peak overshoot(% Mp)	11.5248	2.3338	8.9279

This section examines the effectiveness of the traditional I-PD controller in controlling time-delayed unstable processes. The simulink diagram for the control of time delayed unstable processes using conventional I-PD Controller is shown in figures 5.7, figures 5.9 and figures 5.11. Kaya and Atherton's traditional technique is used to derive the parameters of the standard I-PD controllers [21,22].The closed loop step response of Time delay Unstable Processes using conventional I-PD Controllers are shown in figures 5.8, 5.10 and 5.12. The performance measure, time response specifications, and controller parameters of the conventional I-PD controller is listed in table 5.1 for three different time delayed unstable system. From table 5.1 and figures 5.8, 5.10 and 5.12, It can be observed that the closed-loop response of the TDUP takes a settling time (33.4340, 18.6297and 3.1103) and large percentage overshoot (9.5248, 2.3338, 8.9279) for Example1, Example 2 and Example3 respectively.There fore I-PD controller is also needs some improvement for getting good performance. Therefore, PSO and PSOGSA optimization algorithms are used for fine tune I-PD controller in this work.

5. 3 Control of Time delay Unstable Processes using PSO tuned I-PD controller

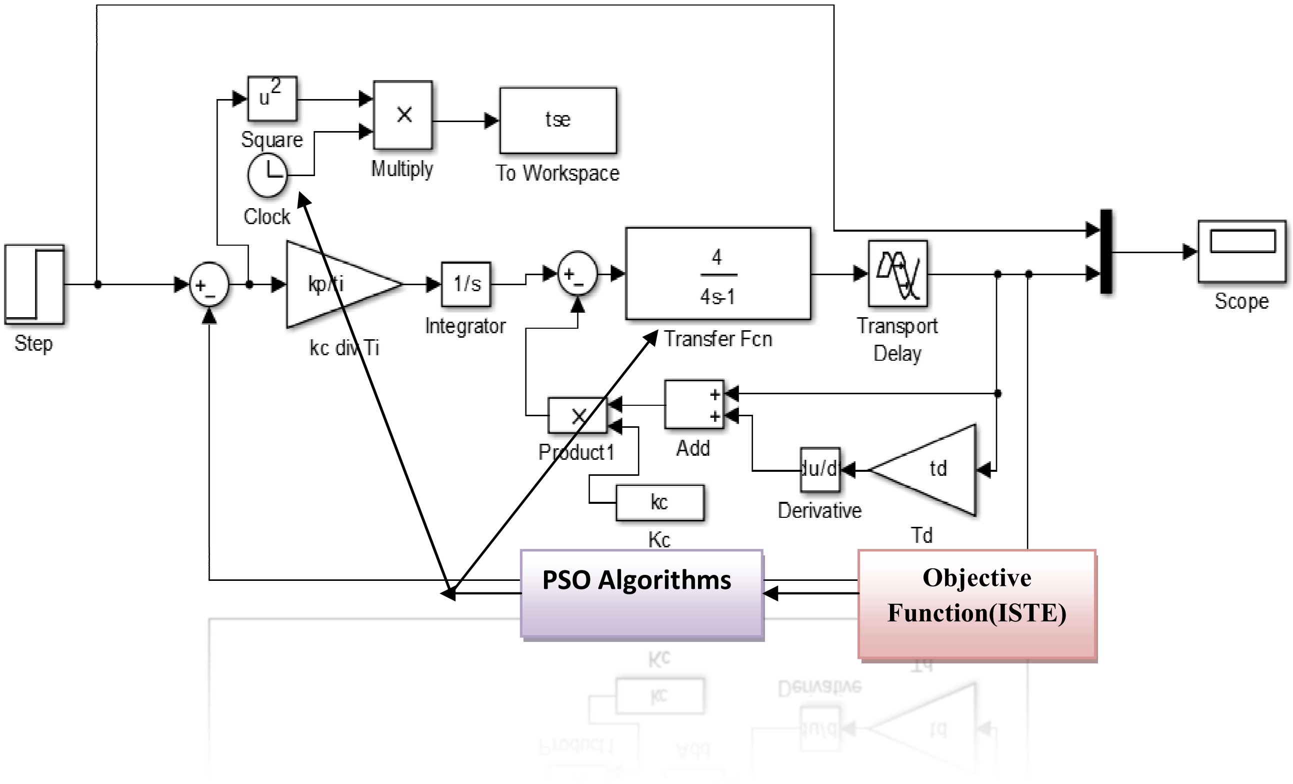

Figure: 5.13:Simulink diagram of control of TDUP using PSO tuned I-PD Controller

Table: 5.2: Performance, time response spe and parameters of the PSO tune for ten runs (Example 1)

Runs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	5.6898	11.6604	1.3258	6.9349	22.1066	9.9956	25.9663	71
Run 2	5.6912	11.6627	1.3243	6.9339	22.0924	10.0000	25.9655	66
Run 3	5.6761	11.6218	1.3448	6.9382	22.2743	10.0000	25.9596	69
Run 4	5.6907	11.6612	1.3250	6.9340	22.0985	9.9997	25.9652	64
Run 5	5.6887	11.6556	1.3278	6.9345	22.1218	9.9998	25.9637	85
Run 6	5.6791	11.6298	1.3408	6.9374	22.2371	9.9991	25.9600	48
Run 7	5.6747	11.6180	1.3467	6.9387	22.2927	10.0000	25.9597	74
Run 8	5.6847	11.6448	1.3333	6.9358	22.1696	9.9984	25.9618	77
Run 9	5.6586	11.5795	1.3668	6.9382	22.2716	9.9995	25.9597	66
Run10	5.6758	11.6210	1.3452	6.9383	22.2783	9.9999	25.9596	72

Table:5. 3: Statistical Analysis for parameters and ISTE for PSO algorithm for ten runs

Parameters→	K_p	K_i	K_d	ISTE
Min	5.6586	11.5795	1.3243	25.9596
Mean	5.75027	11.44287	1.37533	25.96211
Max	5.6912	11.6627	1.3668	25.9663
Median	5.6819	11.6373	1.33705	25.9609
Sta. Deviation	0.21149373	0.629011991	0.128305426	0.002786655

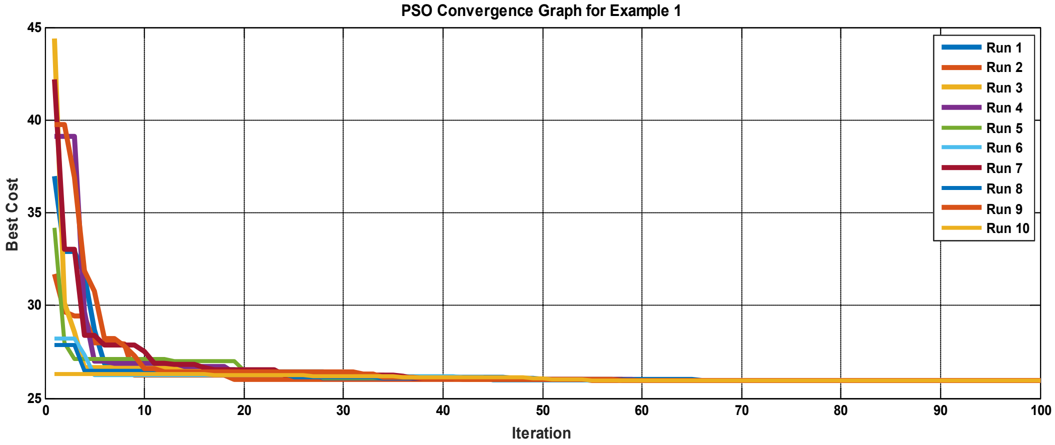

Figure 5. 14: Convergence Graph of PSO algorithm based I-PD controller tuning for ten Runs (Example 1)

Table: 5.4: Performance measure, Time response specifications, and controller parameters of the PSO tuned I-PD controller for ten runs (Example 2)

Runs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	0.6480	7.6028	0.8558	3.2204	16.7044	9.6178	14.9326	67
Run 2	0.6471	7.5740	0.8577	3.2128	16.6886	9.9612	14.9293	62
Run 3	0.6470	7.5718	0.8579	3.2122	16.6876	9.9888	14.9295	58
Run 4	0.6472	7.5783	0.8574	3.2139	16.6911	9.9108	14.9296	72
Run 5	0.6471	7.5741	0.8578	3.2128	16.6885	9.9589	14.9293	55
Run 6	0.6470	7.5712	0.8579	3.2121	16.6870	9.9950	14.9294	57
Run 7	0.6473	7.5808	0.8573	3.2145	16.6924	9.8796	14.9297	67
Run 8	0.6475	7.5870	0.8569	3.2162	16.6960	9.8067	14.9304	71
Run 9	0.6479	7.5994	0.8560	3.2195	16.7024	9.6567	14.9319	56
Run10	0.6474	7.5839	0.8571	3.2154	16.6941	9.8416	14.9300	60

Table: 5.5: Statistical Analysis for Controller parameters and objective function for PSO algorithm for ten runs (Example 2)

Parameters→	K_p	K_i	K_d	ISTE
Min	0.647	7.5712	0.8558	14.929
Mean	0.6474	7.5823	0.8572	14.9301
Max	0.648	7.6028	0.8579	14.9326
Median	0.64725	7.57955	0.85735	14.92965
Sta. Deviation	0.000356682	0.011181041	0.000755425	0.001231801

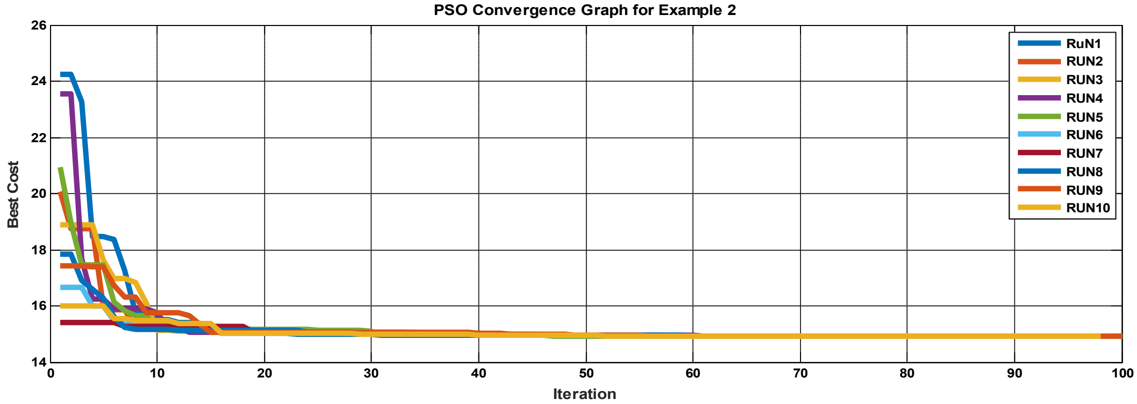

Figure 5.15Convergence Graph of the PSO tuned I-PD controller for ten Runs 10

Table: 5.6: Performance measure, Time response specifications, and controller parameters of the PSO tuned I-PD controller for ten runs (Example 3)

Runs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	3.5530	1.4375	0.1834	0.7429	3.0000	6.0150	2.0421	71
Run 2	3.5564	1.4386	0.1830	0.7426	3.0000	5.9798	2.0425	71
Run 3	3.5530	1.4375	0.1834	0.7429	3.0000	6.0150	2.0421	74
Run 4	3.2317	1.3935	0.1577	0.6660	2.1412	7.9991	2.0494	51
Run 5	3.2409	1.3973	0.1562	0.6651	2.1143	7.9998	2.0486	68
Run 6	3.2328	1.3939	0.1575	0.6658	2.1378	7.9996	2.0493	55
Run 7	3.5514	1.4383	0.1835	0.7440	3.0000	6.0024	2.0434	84
Run 8	3.2427	1.3982	0.1559	0.6650	2.1090	7.9997	2.0486	60
Run 9	3.2434	1.3985	0.1557	0.6650	2.1070	7.9991	2.0486	63
Run10	3.2412	1.3977	0.1561	0.6652	2.1129	7.9988	2.0488	43

Table:5.7: Statistical Analysis for Controller parameters and objective function for PSO algorithm for ten runs (Example 3)

Parameters→	K_p	K_i	K_d	ISTE
Min	3.2317	1.3935	0.1557	2.0421
Mean	3.36465	1.4131	0.16724	2.04634
Max	3.5564	1.4386	0.1835	2.0494
Median	3.24305	1.39835	0.1577	2.0486
Sta. Deviation	0.162543165	0.021475516	0.013859149	0.003314011

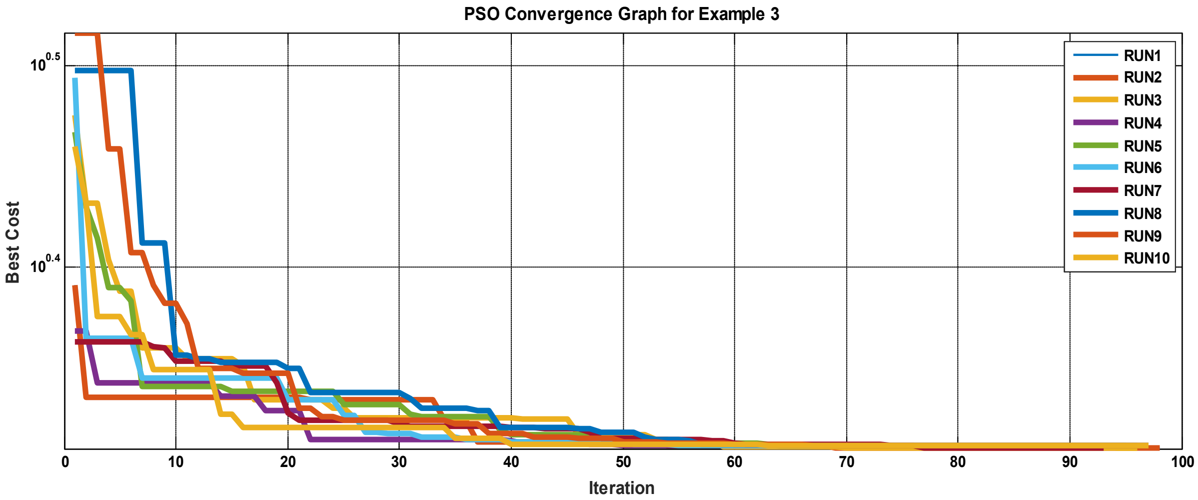

Figure 5.16: Convergence Graph of PSO algorithm-based I-PD controller tuning for ten Runs (Example 3)

Table:5.8: Best values of the Performance measure, Time response specifications, and controller parameters of the PSO tuned I-PD controller for Examples 1,2 and 3

TDUPs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Example1	5.6761	11.6218	1.3448	6.9382	22.2743	10.0000	25.9596	69
Example2	0.6480	7.6028	0.8558	3.2204	16.7044	9.6178	14.9326	67
Example3	3.5530	1.4375	0.1834	0.7429	3.0000	6.0150	2.0421	71

In this section, the parameters of the I-PD controller have been optimized using PSO algorithm by considering ISTE as an objective function. The optimized I-PD controller parameters and time domain specifications are recorded for each run and it listed in Table 5.2, Table 5.4 and Table 5.6 for Example1, Example2 and Example 3 respectively. From the 10 number of run's results, the mean, median, std. deviation, minimum and maximum values of optimal controller parameters and ISTE are tabulated in Table5.3, Table5.5, and Table5.7. The convergence performance of the PSO algorithm for finding I-PD controller parameter is shown in figures 5.14, 5.15 and 5.16 for the three TDUPs. From tables 5.2, 5.4 and 5.6 and figures 5.14, 5.15 and 5.16, it can be observed that the performance of PSO algorithm. The best optimal controller parameters of I-PD controller are selected based on objective function value and recorded in Table: 5.8.

5.4 Control of Time delay Unstable Processes using PSOGSA tuned I-PD controller

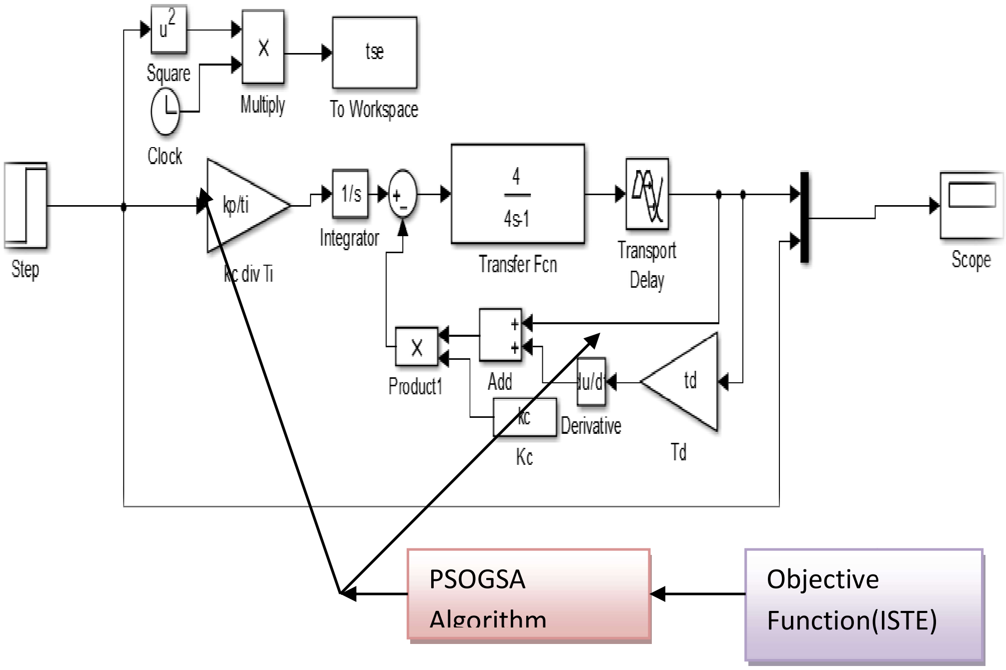

Figure 517: Simulink diagram of control of Time delay Unstable Processes using PSO GSA tuned I-PD Controller

Table: 5.9: Performance measure, Time response specifications, and controller parameters of the PSOGSA tuned I-PD controller for ten runs (Example 1).

Runs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	5.6638	11.5916	1.3604	6.9356	22.1680	10.0000	25.9615	31
Run 2	5.6760	11.6213	1.3450	6.9383	22.2764	10.0000	25.9596	33
Run 3	5.6848	11.6447	1.3332	6.9356	22.1690	10.0000	25.9614	34
Run 4	5.6905	11.6607	1.3332	6.9340	22.1006	10.0000	25.9650	25
Run 5	5.6619	11.5871	1.3628	6.9433	22.4805	10.0000	25.9648	44
Run 6	5.6821	11.6373	1.3369	6.9364	22.2016	10.000	25.9604	32
Run 7	5.6815	11.6357	1.3377	6.9366	22.2091	10.0000	25.9602	33
Run 8	5.6654	11.5953	1.3585	6.9419	22.4111	10.0000	25.9626	32
Run 9	5.6812	11.6350	1.3381	6.9367	22.2125	10.0000	25.9603	33
Run10	5.6836	11.6414	1.3349	6.9411	22.3810	10.0000	25.9609	31

Table: 5.10: Statistical Analysis for Controller parameters and objective function for PSOGSA algorithm for ten runs (Example 1)

Parameters	K_p	K_i	K_d	ISTE
Min	5.6619	11.5871	1.3332	25.9596
Mean	5.67708	11.62501	1.34407	25.96167
Max	5.6905	11.6607	1.3628	25.965
Median	5.6815	11.63535	1.3379	25.96115
Sta. Deviation	0.009934765	0.025265411	0.011900518	0.00189681

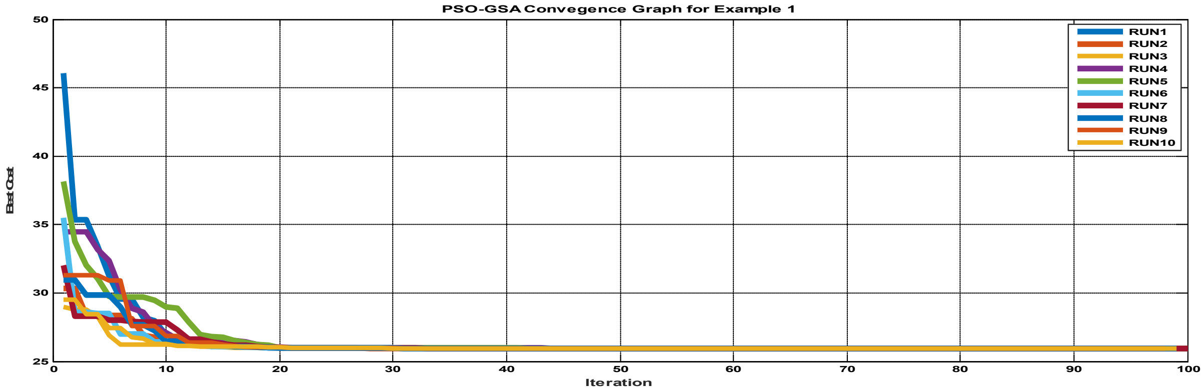

Figure: 5.18: Convergence Graph of PSOGSA algorithm based I-PD controller tuning for ten Runs (Example 1)

Table: 5.11: Performance measure, Time response specifications, and controller parameters of the PSOGSA tuned I-PD controller for ten runs (Example 2)

Runs	K_p	K_i	Kd	Rise time (sec)	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	0.6471	7.5736	0.8578	3.2127	16.6884	9.9655	14.9292	30
Run 2	0.6485	7.6188	0.8548	3.2247	16.7122	9.4275	14.9355	59
Run 3	0.6475	7.5860	0.8569	3.2159	16.6952	9.8166	14.9302	37
Run 4	0.6470	7.5708	0.8580	3.2120	16.6868	9.9994	14.9290	45
Run 5	0.6471	7.5777	0.8577	3.2130	0.9199	9.9496	14.9298	33
Run 6	0.6471	7.5749	0.8577	3.2130	16.6891	9.9496	14.9292	31
Run 7	0.6476	7.5913	0.8566	3.2173	16.6981	9.7523	14.9308	31
Run 8	0.6470	7.5718	0.8579	3.2122	16.6874	9.9877	14.9291	40
Run 9	0.6470	7.5719	0.8579	3.2122	16.6874	9.9867	14.9291	37
Run10	0.6472	7.5788	0.8574	3.2140	16.6913	9.9026	14.9295	33

Table: 5.12: Statistical Analysis for Controller parameters and objective function for PSOGSA algorithm for ten runs (Example 2)

Parameters→	K_p	K_i	K_d	ISTE
Min	0.647	7.5708	0.8548	14.929
Mean	0.64731	7.58156	0.85727	14.93014
Max	0.6485	7.6188	0.858	14.9355
Median	0.6471	7.5763	0.8577	14.92935
Sta. Deviation	0.000467737	0.014666606	0.000982118	0.00196988

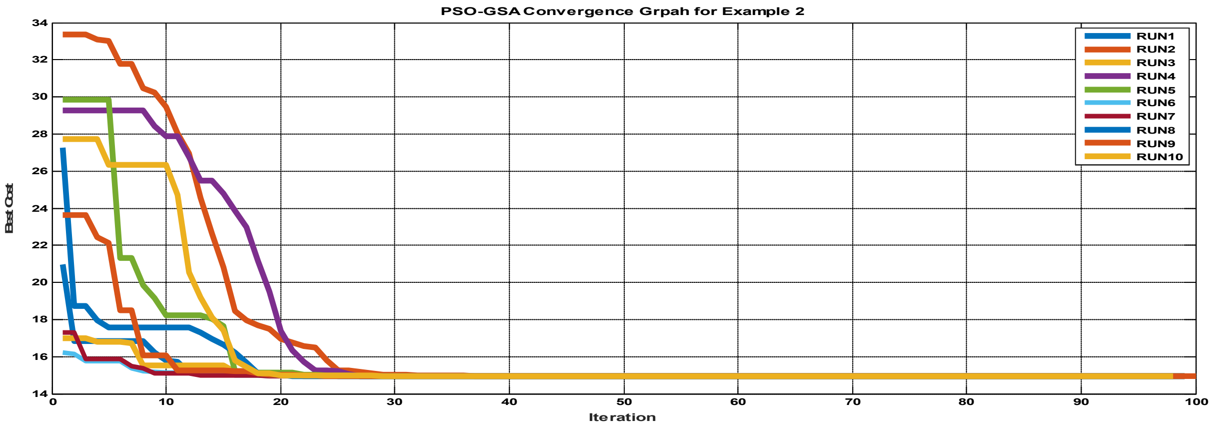

Figure: 5.19: Convergence Graph of PSOGSA algorithm based I-PD controller tuning for ten Runs (Example 2)

Table: 5.13: Performance measure, Time response specifications, and controller parameters of the PSOGSA tuned I-PD controller for ten runs (Example 3)

Runs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Run 1	3.2522	1.3633	0.1566	0.6436	2.1442	8.0000	2.0246	39
Run 2	3.2319	1.3936	0.1577	0.6659	2.1407	8.0000	2.0494	27
Run 3	3.2398	1.3969	0.1564	0.6695	2.1444	8.0000	2.0487	32
Run 4	3.2429	1.3983	0.1558	0.6677	2.1650	8.0000	2.0486	41
Run 5	3.2244	1.3908	0.1588	0.6667	2.1638	8.0000	2.0507	30
Run 6	3.2432	1.3984	0.1558	0.6650	2.1075	8.0000	2.0486	27
Run 7	3.2409	1.3974	0.1562	0.6651	2.1142	8.0000	2.0486	40
Run 8	3.2478	1.4007	0.1549	0.6646	2.4865	8.0000	2.0488	27
Run 9	3.2393	1.3967	0.1565	0.6653	2.1187	8.0000	2.0487	30
Run10	3.2396	1.3968	0.1564	0.6652	2.1177	8.0000	2.0487	28

Table: 5.14: Statistical Analysis for Controller parameters and objective function for PSOGSA algorithm for ten runs (Example 3)

Parameters	K_p	K_i	K_d	ISTE
Min	3.2244	1.3633	0.1549	2.0246
Mean	3.2402	1.39329	0.15651	2.04654
Max	3.2522	1.4007	0.1588	2.0507
Median	3.24035	1.39685	0.1564	2.0487
Sta. Deviation	0.007734483	0.010881631	0.001074399	0.007736522

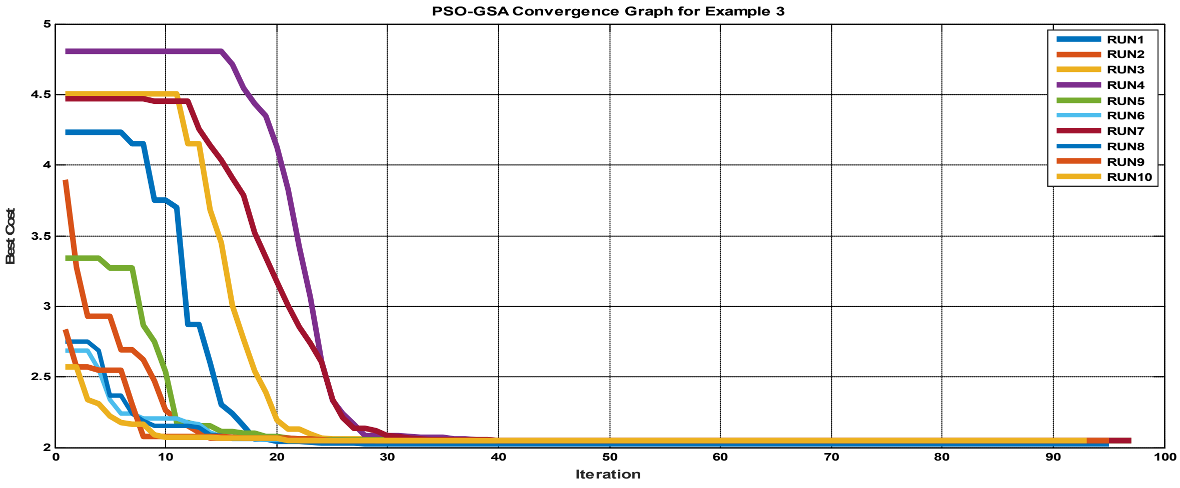

Figure:5.20: Convergence Graph of PSOGSA algorithm-based I-PD controller tuning for ten Runs (Example 3)

Table: 5.1:Best values of the Performance measure, Time response specifications, and controller parameters of the PSOGSA tuned I-PD controller for Examples 1,2 and 3

TDUPs	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Converged at (iteration number)
Example1	5.6815	11.6357	1.3377	6.9366	22.2091	10.0000	25.9596	33
Example2	0.6471	7.5749	0.8577	3.2130	16.6891	9.9496	14.9292	31
Example3	3.2522	1.3633	0.1566	0.6436	2.1442	8.0000	2.0246	39

In this part, the PSOGSA method has been used to optimize the I-PD controller's settings while taking ISTE into account as an objective function. For each run, the optimized I-PD controller settings and time domain requirements are noted and are shown in Tables 5.9, 5.11, and 5.13 for Examples 1, 2, and 3, respectively. From the 10 number of run's results, the mean, median, std. deviation, minimum and maximum values of optimal controller parameters and ISTE are tabulated in Table5.10, Table5.12, and Table5.14. The best optimal controller parameters of I-PD controller is selected based on objective function value from Table 5.9, Table 5.12 and Table 5.13 and it is summarized in table 5.15. The convergence performance of the PSO algorithm for finding I-PD controller parameter is shown in figures 5.18, 5.19 and 5.20 for the three TDUPs. From tables 5.9, 5.11 and 5.13 and figures 5.18, 5.19 and 5.20, it can be observed that the PSOGSA algorithm produced optimal controller parameters with better convergence rate and less % overshoot, rise time and settling time than PSO algorithm.

5.5 Performance comparison, Conventional I-PD, PSO-I-PD and PSOGSA I-PD controller for the control of Time delay Unstable Processes

Table: 5.16: Performance measure, Time response specifications, and controller parameters of various controllers for the TDUPs (Examples 1,2 and 3)

TDUPs	Controller	K_p	K_i	K_d	Rise time (Tr) in sec	Settling time (T_s) in sec	% Peak overshoot (% Mp)	ISTE	Conve at (iterat No)
Example1									
	Conv I-PD	7.0118	9.8891	1.7381	5.6791	33.4340	11.5248	8.07e-11	
	PSO I-PD	5.6761	11.6218	1.3448	6.9382	22.2743	10.0000	25.9596	69
	PSOGSA IPD	5.6815	11.6357	1.3377	6.9366	22.2091	10.0000	25.9596	33
Example2									
	Conv I-PD	0.6933	6.9501	0.8955	2.9252	18.6297	2.3338	9.3996e-07	
	PSO I-PD	0.6480	7.6028	0.8558	3.2204	16.7044	9.6178	14.9292	67
	PSOGSAIPD	0.6471	7.5749	0.8577	3.2130	16.6891	9.9496	14.9292	31
Example3									
	Conv I-PD	3.4717	1.2246	0.1775	0.5780	3.1103	8.9279	1.58e-09	
	PSO I-PD	3.5530	1.4375	0.1834	0.7429	3.0000	6.0150	2.0421	71
	PSOGSA IPD	3.2522	1.3633	0.1566	0.6436	2.1442	8.0000	2.0246	39

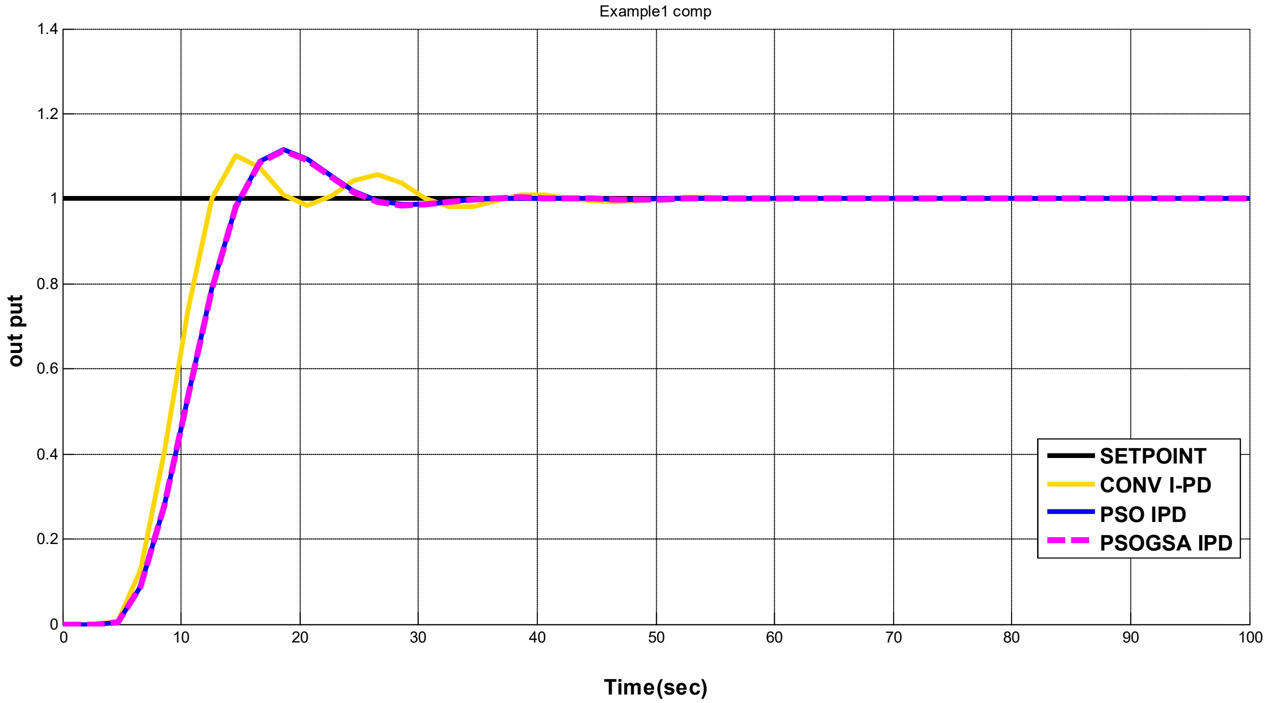

Figure:5.21 Output response of control of TDUP (Example 1) using conventional I-PD Controller, PSO and PSOGSA tuned I-PD Controller (Set point Tracking)

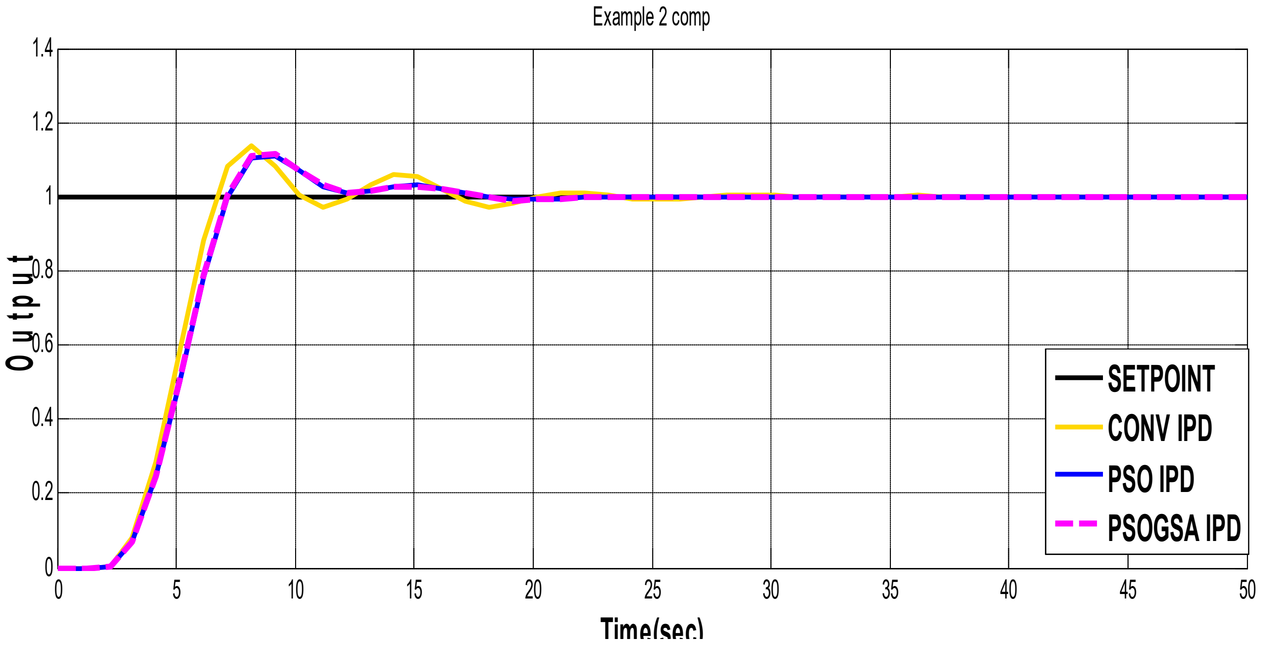

Figure: 5.22: Output response of control of TDUP (Example 2) using conventional I-PD Controller, PSO and PSOGSA tuned I-PD Controller (set point Tracking)

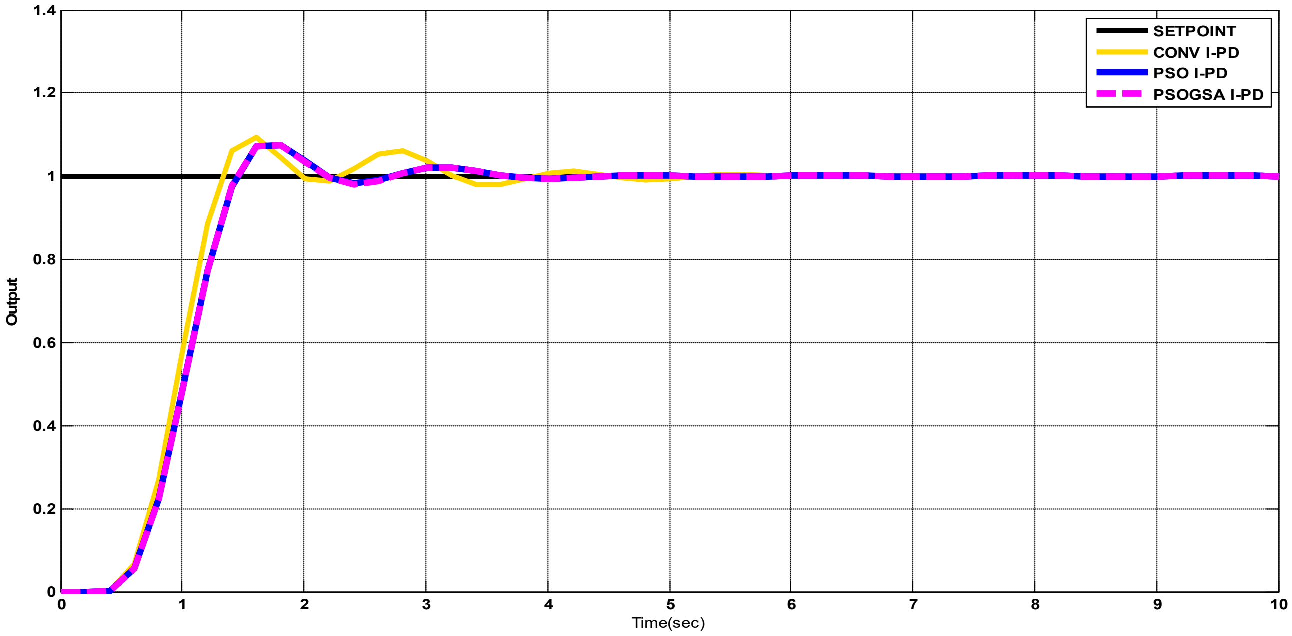

Figure: 5.23: Output response of control of TDUP (Example 3) using conventional I-PD Controller, PSO and PSOGSA tuned I-PD Controller (Tracking of Set point).

This section examines and contrasts the PSO and PSOGSA customized I-PD controllers' performance for the set point tracking problem. Table 5.16 compares the performance of various proposed I-PD controllers based on the response criteria of settling time, convergence rate and objective function. Figures 5.21, Figures 5.22, and Figures 5.23, which demonstrate the performance analysis of the proposed PSO and PSOGSA tuned I-PD controller with set point tracking. From the Figures 5.21, 5.22, and 5.23, it can be observed that the I-PD controller tuned using the PSO and PSOGSA algorithm with objective function ISTE performs well with less settling time, fast convergence rate and best cost. Particularly, the I-PD controller tuned using the PSOGSA algorithm with objective function ISTE outperforms with settling times of (22.2091 sec), convergence rate (33) and objective function ISTE (25.9596) For Example1 (sys1), settling times of (16.6891seconds), convergence rate (31) and objective function ISTE (14.9292) for Example2 (sys2) and less settling time(2.1442 seconds), fast convergence rate (39) and objective function ISTE (2.0246) for Example 3(sys3).Finally, it can be concluded that the PSOGSA set point changes with a short settling period and a quick convergence rate.

5.6: ROBUSTNESS ANALYSIS

Table: 5.17: multi step input various controllers for the TDUPs (Examples 1,2 and 3)

		Tr	Ts	% over shoot
Example1	I-PD	6.7148	32.5038	7.0849
	PSO	7.7949	27.9714	9.6283
	PSOGSA	5.1287	27.8329	9.6283
Example2	I-PD	5.6110	19.5345	5.0186
	PSO	5.6712	18.0469	6.7101
	PSOGSA	5.1287	17.0770	5.4552
Example3	I-PD	5.1107	6.6713	2.2591
	PSO	5.1495	6.4711	1.9589
	PSOGSA	4.2923	6.3986	1.9021

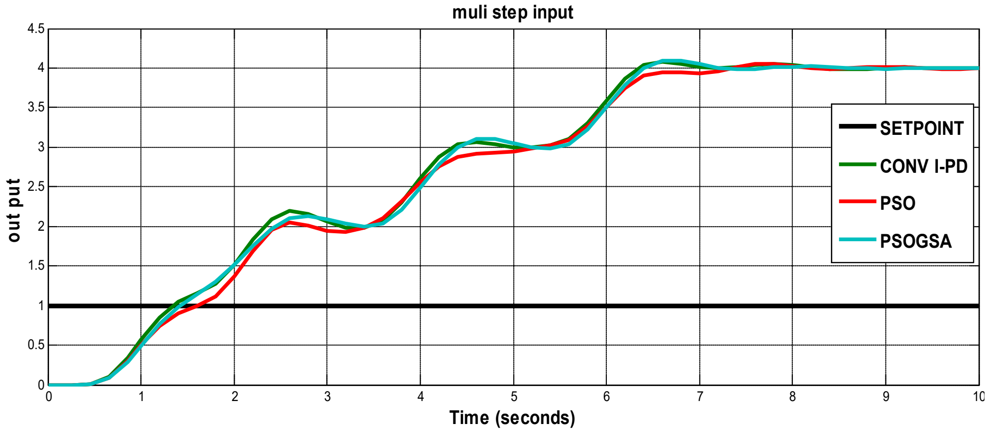

Figure: 5.24: Output response of multi step input for (Examples 1,2 and 3)

This section Robustness analysis of the PSO and PSOGSA customized I-PD controllers' performance (Robustness) for the set point tracking. Table: 5.17 and Figure: 5.24 compares the performance of various proposed I-PD controllers based on the response criteria of settling time and %overshoot observed. Demonstrate the Robustness analysis of the proposed PSO and PSOGSA tuned I-PD controller with set point tracking from the Table:5.17 and Figure:5.24 ,it can be observed that the I-PD controller tuned using the PSO and PSOGSA algorithm with objective function ISTE performs well good nominal and robust control performances are achieved with the proposed method and Significant improvements in the closed-loop performances are obtained.

CHAPTER SIX

CONCLUSION AND FUTURE SCOPE

6.1. CONCLUSION

This thesis study from the simulation result the performance of the conventional I-PD controller, PSO and PSOGSA optimization algorithm are observed. The objective function (ISTE) is considered.PSOGSA algorithms I-PD controller has less settling time, better objective function and fast convergence rate.

6.2. RECOMMENDETION FOR THE FUTURE WORK

Time delay unstable systems can use sophisticated controllers like self-tuning regulators with process parameter estimation and model reference adaptive controllers to achieve satisfying desired performance.

By including Disturbance rejection in the future for controlling time delay unstable system.

REFERENCE

[1] I. Kaya, “Optimum analytical tuning of I-PD controllers for unstable time delay systems,” *2018 6th Int. Conf. Control Eng. Inf. Technol. CEIT 2018*, no. October, pp. 1–6, 2018, doi: 10.1109/CEIT.2018.8751752.

[2] V. Rajinikanth and K. Latha, “I-PD Controller Tuning for Unstable System Using Bacterial Foraging Algorithm: A Study Based on Various Error Criterion,” *Appl. Comput. Intell. Soft Comput.*, vol. 2012, pp. 1–10, 2012, doi: 10.1155/2012/329389.

[3] M. Kumar, D. Prasad, and R. S. Singh, “Performance enhancement of IMC-PID controller design for stable and unstable second-order time delay processes,” no. April, 2020, doi: 10.1007/s11771-020-4280-7.

[4] T. Liu, Y. Z. Cai, D. Y. Gu, and W. D. Zhang, “New modified Smith predictor scheme for integrating and unstable processes with time delay,” *IEE Proc. Control Theory Appl.*, vol. 152, no. 2, pp. 238–246, 2005, doi: 10.1049/ip-cta:20041232.

[5] Q. Zheng and Z. Gao, “Predictive active disturbance rejection control for processes with time delay,” *ISA Trans.*, no. October, pp. 1–9, 2013, doi: 10.1016/j.isatra.2013.09.021.

[6] W. Tan, C. Fu, and W. Tan, “Control of unstable processes with time delays via ADRC Control of unstable processes with time delays via ADRC,” no. June, 2018, doi: 10.1016/j.isatra.2017.09.002.

[7] C. C. Chen, H. P. Huang, and H. J. Liaw, “Set-point weighted PID controller tuning for time-delayed unstable processes,” *Ind. Eng. Chem. Res.*, vol. 47, no. 18, pp. 6983–6990, 2008, doi: 10.1021/ie800001m.

[8] A. Ali and S. Majhi, “Integral criteria for optimal tuning of PI/PID controllers for integrating processes,” *Asian J. Control*, vol. 13, no. 2, pp. 328–337, 2011, doi: 10.1002/asjc.278.

[9] A. F. Singer, G. A. Hassaan, and M. A. Elgamil, “Disturbance Rejection for a Third Order Process Controlled By a Pi-Pd Controller,” *Int. J. Mech. Eng. Technol.*, vol. 11, no. 8, 2020, doi: 10.34218/ijmet.11.8.2020.001.

[10] M. Atif Siddiqui, M. N. Anwar, and S. H. Laskar, "Enhanced control of unstable cascade systems using direct synthesis approach," *Chem. Eng. Sci.*, vol. 232, no. December, p. 116322, 2021, doi: 10.1016/j.ces.2020.116322.

[11] Q. B. Jin and Q. Liu, "Analytical IMC-PID design in terms of performance / robustness tradeoff for integrating processes : From 2-Dof to 1-Dof," *J. Process Control*, vol. 24, no. 3, pp. 22–32, 2014, doi: 10.1016/j.jprocont.2013.12.011.

[12] G. L. Raja and A. Ali, "Smith predictor based parallel cascade control strategy for unstable and integrating processes with large time delay," *J. Process Control*, vol. 52, pp. 57–65, 2017, doi: 10.1016/j.jprocont.2017.01.007.

[13] M. Atif, M. N. Anwar, and S. H. Laskar, "Enhanced control of Unstable Cascade Systems using Direct Synthesis Approach Enhanced control of unstable cascade systems using direct synthesis approach," *Chem. Eng. Sci.*, vol. 232, no. December, p. 116322, 2020, doi: 10.1016/j.ces.2020.116322.

[14] C. Onat, "A New Design Method for PI-PD Control of Unstable Processes with Dead Time Cem Onat," *ISA Trans.*, 2018, doi: 10.1016/j.isatra.2018.08.029.

[15] C. Paper, M. Antonio, and P. Ramos, "PID Controller Tuning for Stable and Unstable PID Controller Tuning for Stable and Unstable," no. June, 2004, doi: 10.1007/978-3-540-24855-2.

[16] G. S. Et. al., "PID Controller optimization using Metahuristic Controller with Different Nonlinearities," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 764–771, 2021, doi: 10.17762/turcomat.v12i2.1082.

[17] N. M. Darwish, "PID controller design in the frequency domain for time-delay systems using direct method," 2016, doi: 10.1177/0142331216675400.

[18] S. Karan, "Modified Smith predictor-based all-proportional-derivative control for second-order delay-dominated integrating processes," no. August, pp. 1–20, 2020, doi: 10.1002/apj.2591.

[19] Y. Lee, J. Lee, and S. Park, "PID controller tuning for integrating and unstable processes with time delay," vol. 55, pp. 3481–3493, 2000.

[20] S. Mirjalili, S. Zaiton, and M. Hashim, “A New Hybrid PSOGSA Algorithm for Function Optimization,” no. 1, pp. 374–377, 2010.

[21] T. Bartz-beielstein, K. Parsopoulos, and M. N. Vrahatis, “Analysis of Particle Swarm Optimization Using Computational Statistics Design and Analysis of Optimization Algorithms Using Compu- tational Statistics,” no. January, 2004, doi: 10.1002/anac.200410007.

APPENDIXES

Tuning of I-PD controller using Particle Swarm Optimization

```
%
% Copyright (c) 2015, Yarpiz (www.yarpiz.com)
% All rights reserved. Please read the "license.txt" for license terms.
%
% Project Code: YPEA102
% Project Title: Implementation of Particle Swarm Optimization in MATLAB
% Publisher: Yarpiz (www.yarpiz.com)
%
% Developer: S. Mostapha Kalami Heris (Member of Yarpiz Team)
%
% Contact Info: sm.kalami@gmail.com, info@yarpiz.com
%
clc;
clear;
close all;
%% Problem Definition
CostFunction=@tracklsq;        % Cost Function
nVar=3;            % Number of Decision Variables
```

```
VarSize=[1 nVar];   % Size of Decision Variables Matrix
VarMin(:,1)=0; % Kp1
VarMin(:,2)=0; % Ti
VarMin(:,3)=0; %Td
VarMax(:,1)=15;        %Kp1
VarMax(:,2)=20;         % Ti
VarMax(:,3)=5;         % Td
% VarMin=-10;         % Lower Bound of Variables
% VarMax= 10;         % Upper Bound of Variables
%
%% PSO Parameters
% MaxIt=1000;      % Maximum Number of Iterations
MaxIt=100;      % Maximum Number of Iterations
nPop=50;        % Population Size (Swarm Size)
% PSO Parameters
w=1;            % Inertia Weight
wdamp=0.99;     % Inertia Weight Damping Ratio
c1=1.5;         % Personal Learning Coefficient
c2=2.0;         % Global Learning Coefficient
% If you would like to use Constriction Coefficients for PSO,
% uncomment the following block and comment the above set of parameters.
```

```
% % Constriction Coefficients

% phi1=2.05;

% phi2=2.05;

% phi=phi1+phi2;

% chi=2/(phi-2+sqrt(phi^2-4*phi));

% w=chi;          % Inertia Weight

% wdamp=1;        % Inertia Weight Damping Ratio

% c1=chi*phi1;    % Personal Learning Coefficient

% c2=chi*phi2;    % Global Learning Coefficient

% Velocity Limits

VelMax=0.1*(VarMax-VarMin);

VelMin=-VelMax;

%% Initialization

empty_particle.Position=[];

empty_particle.Cost=[];

empty_particle.Velocity=[];

empty_particle.Best.Position=[];

empty_particle.Best.Cost=[];

particle=repmat(empty_particle,nPop,1);

GlobalBest.Cost=inf;

for i=1:nPop
```

```
    % Initialize Position

    particle(i).Position=unifrnd(VarMin,VarMax,VarSize);

    % Initialize Velocity

    particle(i).Velocity=zeros(VarSize);

    % Evaluation

    particle(i).Cost=CostFunction(particle(i).Position);

    % Update Personal Best

    particle(i).Best.Position=particle(i).Position;

    particle(i).Best.Cost=particle(i).Cost;

    % Update Global Best

    if particle(i).Best.Cost<GlobalBest.Cost
        GlobalBest=particle(i).Best
BestCost=zeros(MaxIt,1);

%% PSO Main Loop

for it=1:MaxIt

    for i=1:nPop

        % Update Velocity

        particle(i).Velocity = w*particle(i).Velocity ...

            +c1*rand(VarSize).*(particle(i).Best.Position-particle(i).Position) ...

            +c2*rand(VarSize).*(GlobalBest.Position-particle(i).Position);
```

```
    % Apply Velocity Limits
    particle(i).Velocity = max(particle(i).Velocity,VelMin);
    particle(i).Velocity = min(particle(i).Velocity,VelMax);
    % Update Position
    particle(i).Position = particle(i).Position + particle(i).Velocity;
    % Velocity Mirror Effect
    IsOutside=(particle(i).Position<VarMin | particle(i).Position>VarMax);
    particle(i).Velocity(IsOutside)=-particle(i).Velocity(IsOutside);
    % Apply Position Limits
    particle(i).Position = max(particle(i).Position,VarMin);
    particle(i).Position = min(particle(i).Position,VarMax);
    % Evaluation
    particle(i).Cost = CostFunction(particle(i).Position);
    % Update Personal Best
    if particle(i).Cost<particle(i).Best.Cost
        particle(i).Best.Position=particle(i).Position;
        particle(i).Best.Cost=particle(i).Cost;
        % Update Global Best
        if particle(i).Best.Cost<GlobalBest.Cost
            GlobalBest=particl
BestCost(it)=GlobalBest.Cost;
```

```
    disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);

    w=w*wdamp;

end

BestSol = GlobalBest

%% Results

figure;

%plot(BestCost,'LineWidth',2);

semilogy(BestCost,'LineWidth',2);

xlabel('Iteration');

ylabel('Best Cost');

grid on;

Pidbest=BestSol.Position;

tracklsq(Pidbest)

 kp = Pidbest(:,1)

 ti =Pidbest(:,2)

 td =Pidbest(:,3)

% Tunning of I-PD controller using PSOGSA Optimization
%PSOGSA source code v3.0, Generated by SeyedAli Mirjalili, 2011.
%Adopted from: S. Mirjalili, S.Z. Mohd Hashim, “A New Hybrid PSOGSA
%Algorithm for Function Optimization, in IEEE International Conference
%on Computer and Information Application?ICCIA 2010), China, 2010, pp.374-377.
clear all
clc
```

```
N = 50;                        % Size of the swarm " no of objects "
Max_Iteration  = 100;              % Maximum number of "iterations"
% dim=3;
%  low(1,:)=0; % Kp1
%  low(2,:)=0; % Ti
% low(3,:)=0; %Td
%
%
%
% up(1,:)=15;        %Kp1
% up(2,:)=20;         % Ti
% up(3,:)=5;         % Td
Benchmark_Function_ID=1; %Benchmark function ID
% Benchmark_Function_ID=@tracklsq %Benchmark function ID
[gBestScore,gBest,GlobalBestCost]=PSOGSA(Benchmark_Function_ID, N, Max_Iteration)
% [gBestScore,gBest,GlobalBestCost]= PSOGSA(low,up,dim,N,Max_Iteration);
gBest
gBestScore
semilogy(GlobalBestCost,'Color','r');

% title(['\fontsize{12}\bf Benchmark Function: F',num2str(Benchmark_Function_ID)]);

xlabel('\fontsize{12}\bf Iteration');ylabel('\fontsize{12}\bf Fitness(Best-so-far)');

% legend('\fontsize{10}\bf PSOGSA',1);

tracklsq(gBest)
%PSOGSA source code v3.0, Generated by SeyedAli Mirjalili, 2011.
%Adopted from: S. Mirjalili, S.Z. Mohd Hashim, “A New Hybrid PSOGSA
%Algorithm for Function Optimization, in IEEE International Conference
%on Computer and Information Application?ICCIA 2010), China, 2010, pp.374-377.
% This function gives boundaries and dimension of search space for test functions.
function [down,up,dim]=benchmark_functions_details(Benchmark_Function_ID)

%If lower bounds of dimensions are the same, then 'down' is a value.
```

```
%Otherwise, 'down' is a vector that shows the lower bound of each dimension.
%This is also true for upper bounds of dimensions.
%Insert your own boundaries with a new Benchmark_Function_ID.
dim=3;
if Benchmark_Function_ID==1
  down=[0; 0; 0]
  up=[15; 20; 5]
% down(:,1)=0; % Kp1
%  down(:,2)=0; % Ti
%  down(:,3)=0; %Td
% up(:,1)=15;       %Kp1
% up(:,2)=20;        % Ti
% up(:,3)=5;         % Td
%     down=-100;up=100;
end

if Benchmark_Function_ID==2
    down=-10;up=10;                                        end
if Benchmark_Function_ID==3
    down=-100;up=100;
end

if Benchmark_Function_ID==4

    down=-100;up=100;

end

if Benchmark_Function_ID==5

    down=-30;up=30;

end

if Benchmark_Function_ID==6

    down=-100;up=100;

end
```

```
if Benchmark_Function_ID==7
    down=-1.28;up=1.28;
end
if Benchmark_Function_ID==8
    down=-500;up=500;
end


if Benchmark_Function_ID==9
    down=-5.12;up=5.12;
end
if Benchmark_Function_ID==10
    down=-32;up=32;
end
if Benchmark_Function_ID==11
    down=-600;up=600;
end
if Benchmark_Function_ID==12
    down=-50;up=50;
end
if Benchmark_Function_ID==13
    down=-50;up=50;
end
if Benchmark_Function_ID==14
    down=-65.536;up=65.536;dim=2;
end
```

```
if Benchmark_Function_ID==15
    down=-5;up=5;dim=4;
end
if Benchmark_Function_ID==16
    down=-5;up=5;dim=2;
end
if Benchmark_Function_ID==17
    down=[-5;0];up=[10;15];dim=2;
end
if Benchmark_Function_ID==18
    down=-2;up=2;dim=2;
end

if Benchmark_Function_ID==19
    down=0;up=1;dim=3;
end
if Benchmark_Function_ID==20
    down=0;up=1;dim=6;
end
if Benchmark_Function_ID==21
    down=0;up=10;dim=4;
end
if Benchmark_Function_ID==22
    down=0;up=10;dim=4;
end
```

```
if Benchmark_Function_ID==23
    down=0;up=10;dim=4;
end
```

- The MATLAB code used for the calculation of kc, Ti and Td is given below:

Example 1

```
theta1=4;
tou=20;
k=1;
% kc calculation
numkc=212.4-71.55*(theta1/tou)+24.76*(theta1/tou)^2;
denkc= 0.0009459 + 148.8*(theta1/tou)-34.42*(theta1/tou)^2+(theta1/tou)^3;
kc=(numkc/denkc)/k
% Ti calculation
numTi=-12.94+685.8*(theta1/tou)+837.1*(theta1/tou)^2+312.3*(theta1/tou)^3-
75.49*(theta1/tou)^4;
denTi=262.8+440.3*(theta1/tou)-721.7*(theta1/tou)^2+220.1*(theta1/tou)^3+(theta1/tou)^4;
Ti=(numTi/denTi)*tou
%Td calculation
numTd=-0.0002104+0.4259*(theta1/tou)+0.05089*(theta1/tou)^2-0.01274*(theta1/tou)^3;
Td=numTd*tou
kc =7.0118       Ti =9.8891         Td =     1.7381
```

Example 2

```
theta1=2;
tou=4;
k=4;
% kc calculation
numkc=212.4-71.55*(theta1/tou)+24.76*(theta1/tou)^2;
denkc= 0.0009459 + 148.8*(theta1/tou)-34.42*(theta1/tou)^2+(theta1/tou)^3;
kc=(numkc/denkc)/k
```

```
% Ti calculation
numTi=-12.94+685.8*(theta1/tou)+837.1*(theta1/tou)^2+312.3*(theta1/tou)^3-
75.49*(theta1/tou)^4;
denTi=262.8+440.3*(theta1/tou)-
721.7*((theta1/tou)^2)+220.1*((theta1/tou)^3)+((theta1/tou)^4);
Ti=(numTi/denTi)*tou
%Td calculation
numTd=-0.0002104+0.4259*(theta1/tou)+0.05089*(theta1/tou)^2-0.01274*(theta1/tou)^3;
Td=numTd*to
```

```
 kc = 0.6933            Ti =6.9501    Td =  0.8955
```

Example 3

```
theta1=0.4;
tou=1;
k=1;
% kc calculation
numkc=212.4-71.55*(theta1/tou)+24.76*(theta1/tou)^2;
denkc= 0.0009459 + 148.8*(theta1/tou)-34.42*(theta1/tou)^2+(theta1/tou)^3;
kc=(numkc/denkc)/k
% Ti calculation
numTi=-12.94+685.8*(theta1/tou)+837.1*(theta1/tou)^2+312.3*(theta1/tou)^3-
75.49*(theta1/tou)^4;
denTi=262.8+440.3*(theta1/tou)-721.7*(theta1/tou)^2+220.1*(theta1/tou)^3+(theta1/tou)^4;
Ti=(numTi/denTi)*tou
%Td calculation
numTd=-0.0002104+0.4259*(theta1/tou)+0.05089*(theta1/tou)^2-0.01274*(theta1/tou)^3;
Td=numTd*tou
```

```
        kc =3.4717          Ti = 1.2246     Td =0.1775
```