IMPLEMENTATION OF SWARM INTELLIGENCE ALGORITHMS IN DESIGNING OPTIMIZED PI CONTROLLER FOR ZETA CONVERTER FOR ENHANCED PERFORMANCE

by

Md. Safwan Zaman (170021086)

Samiur Rahman Sami (170021077)

Zubayer Ahmed (170021076)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering

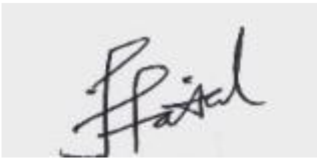
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh

MAY 2022

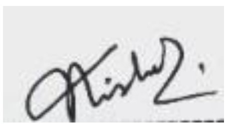
CERTIFICATE OF APPROVAL

The thesis titled " Implementation Of Swarm Intelligence Algorithms In Designing Optimized PI Controller For Zeta Converter For Enhanced Performance '' submitted by Safwan Zaman (170021086), Samiur Rahman Sami (170021077), and Zubayer Ahmed (170021076) has already been considered to be satisfactory and granted as partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 10^{th} May, 2022.

Approved by:



(Signature of the Supervisor)
Mr. Fahim Faisal
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology



(Signature of the Co-Supervisor)
Mr. Mirza Muntasir Nishat
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology

DECLARATION OF CANDIDATES

This thesis, or any portion of it, has not been submitted to any other institution for the accomplishment of a degree or diploma.

Safwan Zaman

(Signature of Candidate)
Md. Safwan Zaman
Student ID: 170021086

Samiurr Rahman Sami

(Signature of Candidate)
Samiur Rahman Sami
Student ID: 170021077

Zubayer

(Signature of Candidate)
Zubayer Ahmed
Student ID: 170021076

ACKNOWLEDGEMENTS

First and foremost, we express our gratitude to Allah for providing us with the strength and goodwill necessary to do proper research and finish our thesis. As well as, we would like to express our heartfelt gratitude to our supervisor and our administration for providing us with the fantastic chance to work on this fine thesis, which also allowed us to conduct a lot of research and learn a lot of new things. We owe them a debt of gratitude.

Mr. Fahim Faisal, Assistant Professor, Department of EEE, IUT, is our esteemed supervisor, and we are grateful for his direction and attention in doing this research. He has always encouraged us to expand our knowledge and study new things in order to keep our minds stimulated. We would not be interested in investigating the power electronics business if it weren't for his motivation. He has urged us to obtain fundamental knowledge on the topic and has also demonstrated how to proceed properly.

We want to thank our co-supervisor, ***Mr. Mirza Muntasir Nishat,*** Assistant Professor, Department of EEE, IUT, for his honest efforts in our thesis research**.** He devoted his significant time to coaching and inspiring us to finish the project. We conducted study and collected numerous useful analyses under his leadership in order to get positive results. He gave us the most efficient technique to have a better understanding of our study.

We'd also want to express our gratitude to ***Prof. Dr. Mohammad Rakibul Islam***, Head, Department of EEE, and the whole EEE Department faculty of IUT for their constant support and inspiration.

Words cannot express how grateful we are to our family and friends. We could not have finished that project without your amazing support! Thanks for sticking with us and providing support.

ABSTRACT

This thesis investigates the consistency and improves the effectiveness of the closed-loop Zeta converter using Swarm Intelligence (SI) methodologies in the building of an optimized PI controller. In the last few years, SI algorithms have become more widely used in the development and optimization of power converters. Swarm intelligence ensures improved data exploitation, concentrating the search technique in the vicinity of optimal solutions while also assisting the procedure to escape from the confines of the local minima, resulting in successful search space exploration. The applicability and compatibility of five swarm intelligence algorithms, including Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant colony Optimization for continuous domain (ACOR), Firefly Algorithm (FA), and Shuffled Frog Leaping Algorithm (SFLA) in optimizing the power converter's control mechanism are investigated. The closed-loop transfer function of the Zeta converter was designed using the State Space Averaging (SSA) approach. The State Space Averaging (SSA) method is employed to evaluate transfer functions and give a viable feedback control method. This study looks into converter stability when it's utilized with both a traditional and a SI-based Proportional Integral (PI) controller, and then compares the results. To test the system's stability, four objective functions (IAE, ITAE, ISE, and ITSE), gain values, and other performance data such as percentage of overshoot, rise time, settling time, and peak amplitude are tabulated. Our system's eigenvalues have been examined to see if it is stable or unstable. The placements of the eigenvalues on the s-plane will indicate the stability of our system. The MATLAB-Simulink platform is used to generate simulation results and analyses. A comparison study is presented to provide a thorough assessment of the performances. The purpose of this study is to investigate the stability and improve the performance of the closed loop Zeta converter using Swarm intelligence methodologies in the building of an improved PI controller. In recent years, swarm intelligence algorithms have become increasingly popular in the development and optimization of power converter. According to our investigation on the performances for the case of the zeta converter, it has been observed that the percentage of overshoot provided by ACOR-PI-ISE is lower than that of other SIA-based PI controllers for each of the objective functions. Aside from that, the ACOR-PI-IAE values are also superior for rise time and settling time. However the peak amplitude is almost same for all investigated five SIA based PI controller with their each objective function. Along with the eigen value observations, the real portions are on the left side of the s plane, while the imaginary parts cancel each other out, implying that each examined SIA-PI Controller is in a stable condition. Furthermore, eigen value graphing shows that the ACOR-PI-ISE has the largest negative real component, indicating that it is more stable than the other four. Hence, In a summary, as per our findings this can be stated that the ACOR-PI controller with the ISE objective function offered better optimization for the zeta converter.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Abbreviated Form	Description
DC	Direct Current
FET	Field-Effect Transistor
BJT	Bipolar Junction Transistor
SCR	Silicon Controlled Rectifier
PWM	Pulse Width Modulation
SSA	State Space Averaging
PI	Proportional Integral
SIA	Swarm Intelligence Algorithm
PSO	Particle Swarm Optimization
FA	Firefly Algorithm
ABC	Artificial Bee Colony
ACOR	Ant Colony Optimization for Continuous Domain
SFLA	Shuffle Frog Leaping Algorithm
SI	Swarm Intelligence
THD	Total Harmonic Distortion
MPPT	Maximum Power Point Tracking
GMPP	Global Maximum Power Point
LMPP	Local Maximum Power Point
MPP	Maximum Power Point
ITSE	Integral Time Square Error
ITAE	Integral Time Absolute Error
IAE	Integral Absolute error
ISE	Integral Sum Error
MCN	Maximum Number of Cycles
DCM	Discontinuous Conduction Mode
CCM	Continuous Conduction Mode
SMPS	Switch Mode Power Supplies
EMI	Electromagnet Interference

CHAPTER 1

INTRODUCTION

Many of us working in the field today are likely unaware of how long power electronics has been around. Several outstanding studies have been published in the last two decades that examine the historical development of power electronics from various perspectives [1]. Power electronics is a new technology that has found its way into a wide range of applications, including renewable energy generation, electric vehicles (EVs), biomedical equipment, and small appliances such as laptop chargers [2]. However, due to the action of the switches, these devices exhibit nonlinear features such as peak overshoot, greater ripples in output voltage, and instability [3]. This thesis study aims to address the aforementioned concerns and give better control over the converter system in order to maximize the effectiveness of their applications. In order to improve the controller's performance and produce optimum system output with stable performance parameters, Swarm Intelligence Algorithms were infused with the PI controller in an investigational method.

1.1 Historical background

The progression of power electronics began with the development of the mercury-arc rectifier around the turn of the century. It was in the 1930s that novel gas tubes, such as ignitrons and phanotrons, were gradually introduced into use. Phase-controlled rectifiers, inverters, and cycloconverters were all developed during this time period. Saturable reactor magnetic amplifiers came into use after World War II. General Electric Company launched the silicon-controlled rectifier, or thyristor, in 1958, starting off a dramatic revolution in power electronics that began in Bell Laboratories in 1956 [4].

The Zeta converter is used in our thesis work to provide a stable output with promising performance parameters that fulfill the requirements. The DC-DC converter we'll look at now was invented by Kazimierczuk and Barbi around the late 1980s, under the names Dual SEPIC and Zeta converter (after the Greek alphabet's sixth letter, to correlate to the "sixth" converter). Zeta is a DC-DC converter with a fourth-order conversion. The output polarity of a zeta converter can vary above or below the input voltage without changing. Zeta converters have the advantage of non-inverted output (the output voltage is the same polarity as the input voltage) in comparison to BUCK – BOOST converters. The ripple voltage and efficiency of the converter can also be influenced by inductors and capacitors. To change from one voltage to another, this converter transfers energy between the inductance and the capacitance. Switching device (MOSFET) regulates the amount of energy transferred [5].

Various mathematical approaches have been described as a result of the continuous active research on DC-DC converters during the last two decades [6]. One of the most well-known strategies is the State-Space Averaging (SSA) technique [7]. The SSA approach is used to dynamically model a Zeta converter in this work. It is widely accepted because it gives a methodical technique to model the converter. There are three stages to SSA modeling: (1) In a switching cycle, for each subinterval, it formulate the converter's state-space equations, (2) then averaging of these equations to obtain a single averaged state-space equation, and (3) lastly to obtain a linear small-signal state-space equation from which multiple transfer functions can be derived, perturbing the averaged equation is performed. [8].

A control technique known as Proportional Integral (PI) was used to update the power converter's steady-state and transient responses. The PI controller has been one of the most basic and widely used controllers for decades. Because they regulate process variables via a control loop feedback mechanism, PI (proportional-integral) controllers are the most precise and stable. Both proportional and integral control actions are used in the PI controller. The rising time and steady-state error are reduced with a proportional controller (Kp), but they are never completely disappeared. Integral feedback (Ki) can be utilized to reduce the amount of forwarding gain necessary and remove steady-state error. Controller tuning is the process of adjusting controller parameters to match a set of performance requirements [9]. A typical PI tuning controller technique is to reduce the performance index in question to a minimum or maximum value [10]. Because manual tuning is difficult to apply in determining optimum Kp and Ki values, multiple swarm intelligence techniques are utilized to overcome this problem.

Swarm Intelligence Algorithms (SIA) such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization for continuous domain (ACOR), Firefly Algorithm (FA) and Shuffled Frog Leaping Algorithm (SFLA) are used to refine the regulation over the PI controller for optimizing the output of the Zeta converter. Swarm intelligence (SI), an important component of artificial intelligence, is progressively gaining traction as more and more high-complexity problems necessitate solutions that are sub-optimal but still feasible in a reasonable amount of time. Swarm intelligence is a subtype of artificial intelligence that is designed to emulate the collective behavior of a group of animals fighting to live. It is primarily inspired by biological systems [11]. A swarm intelligence algorithm is a simulation method for simulating biological group intelligence. The potential parallelism and distributed features of SI algorithms enable them to solve challenging nonlinear problems with improved self-adaptability, resilience, and searchability. [12]. Particle Swarm Optimization (PSO) [13], Ant Colony Optimization (ACO) [14], Artificial Bee Colony (ABC) [15], Firefly Algorithm (FA) [16], Shuffled Frog Leaping Algorithm (SFLA) [17] are only a few of the SI-inspired optimization approaches currently available. These methods have been used to a wide range of issues.

1.2 Literature review

In the realm of power converters, some academics have recommended the use of various algorithms, and there is a large amount of research on the subject. Bayat, Farhad, Maziyar Karimi, and Asghar Taheri [18] investigated the output optimization of the zeta DC-DC converter in the presence of model ambiguity, input voltage changes, and fluctuations in load. The linear matrix inequality tool was used to design and solve this problem, which resulted in a robust state feedback controller that is optimal. Finally, the recommended technique's performance was compared to the commonly used PI controller, and it was determined that the proposed method's nominal performance is comparable to the PI controller.. The ACO-based PID controller for the zeta converter with the model order reduction method is presented by Arun, S., and T. Manigandan [19]. The linear model of the zeta converter is attained using the SSA technique, and the fourth-order transfer function is determined in terms of duty ratio to the output voltage. Thirumeni, Mariammal, and Deepa Thangavelusamy [20] presented the study and design of a DC-DC Cuk converter with a cascaded control approach using a PSO–GSA adjusted PI and SMC controller. Sonmez, Yusuf, and colleagues developed the ABC-PID controller method, which they applied in a buck converter [21]. The results show that ABC-PID outperforms GA-PID in managing the buck converter in terms of settling time, steady-state error, and load shift scenarios. Dhivya P. and K. Ranjith Kumar. used MATLAB/Simulink to generate a simulation of the SEPIC converter using the P&O, PSO, and Firefly algorithms, and found that FA decreases the difficulty of establishing Global Maximum Power, reduces convergence, and enhances the converter's conversion efficiency [22].

1.3 Problem statement

In power systems, the Zeta converter is a popular power converter. But, nonlinear features, ripple currents, fluctuating input voltages, varying loads, and other factors make regulation of such converters difficult. The proportion of overshoot, rising time, settling time, and peak amplitude performance characteristics of an open-loop zeta converter cannot fulfill expectations. As a result, a controller is required to increase performance and maintain a consistent output voltage.

This thesis uses a PI controller to achieve proper voltage regulation of a zeta converter, which has been tuned using different techniques to discover ideal values for the proportional and integral gains, resulting in better converter performance. However, using the manual tuning method to find appropriate values for Kp and Ki in a simple PI tuner for proper control is difficult. To resolve such a dilemma, various types of swarm intelligence algorithms can be recommended. Thus, intelligent search methods that induce learning abilities to analyze the situation and change the controller accordingly are used to obtain the appropriate controller gains. The SIAs used in this thesis work employ population, memory, competitiveness, and cooperative interaction to improve results. When PSO, ABC, ACOR, FA and SFLA are used in conjunction with the PI controller to improve the converter system, the whole system becomes more secure, stable, and capable.

1.4 Thesis objectives

The objectives of this thesis work are as follows:

- Stability investigation of a Zeta power converter by regulating the output voltage to a different level.

- Observing the transient and steady state responses of closed loop Zeta converter Using PI strategy of control.

- Implementation and analyzing of Swarm Intelligence Algorithms(SIA) such as Particle Swarm Optimization(PSO), Ant Colony Optimization for continuous domain(ACOR), Artificial Bee Colony(ABC), Firefly Algorithm(FA) and Shuffled Frog Leaping Algorithm(SFLA) to find their compatibility for the controlling mechanism of power converter.

1.5 Limitation of the study

The Zeta converter was modeled for an ideal situation with a constant load and constant input voltage in this thesis study. MATLAB software was used to run all of the simulations. However, real-world applications will have more complicated and demanding conditions, which will affect the system's output. As a result, the control system programmed for this scenario may not be suitable for more difficult situations. Furthermore, hardware implementations are likely to introduce certain inevitable faults and reduce efficiency. The algorithms must be adjusted to suit the imposed issues in order to retain the optimization level.

CHAPTER 2

DC-DC CONVERTERS

In the 1920s, a DC–DC converter technology was developed. It has been studied for over six decades and is used extensively in power engineering and drive systems. This is widely employed in a variety of industrial applications, computer hardware circuits, and, most notably, the creation of renewable energy power. Traditional basic voltage divider circuits such as rheostat and potential divider are replaced with power electronic-centered DC–DC converters. The output voltage is lower than the input voltage in this approach, and the efficiency is lower [23]. Converters are classified by application, switching type, current mode, and other factors. The terms non-isolated and isolated are commonly used to describe DC–DC converter types. The electrical barrier that separates the input and output of a DC–DC converter is referred to as "isolation" [24].

2.1 Topologies of converter circuits:

Topologies for DC-DC converters are designed to meet the needs of certain DC loads. DC-DC converters such as the buck, boost, buck-boost, Cuk, SEPIC, and Zeta can function as switching mode regulators, regulating unregulated DC voltage with conversion to increase or reduce the value of DC output voltage utilizing power switching devices for PWM switching at a fixed frequency to achieve an appropriate utilization voltage [25]. Buck, boost, buck-boost, Cuk and sepic are briefly discussed and zeta is broadly discussed later.

2.1.1 Boost converter:

It is sometimes required to boost the dc voltage. A boost converter's output voltage is always higher than the input voltage, which is dependent on switching frequency. Only four external components are required for the DC/DC boost converter: an inductor, an electronic switch, a diode, and an output capacitor [26]. Dependent on its energy storing aptitude and the switching time's relative duration, the converter can function in one of two modes. The discontinuous conduction mode (DCM) and the continuous conduction mode (CCM), which correspond to the situations with and without an idle interval, are the two operating modes [27,28]. The switch is on and the diode is reverse biased in the ON state. Energy is stored in the inductor as the current flows from the voltage supply to the switching device. The diode is forward biased and the switch is turned off in the OFF state. The inductor transfers the stored energy to the output load while maintaining a constant current flow and increasing the output voltage.

2.1.2 Buck converter:

Galvanic isolation between the input and output of a buck converter (voltage step-down converter) is not provided because it is not an isolated converter. When a higher input voltage needs to be transformed into a lower output voltage, this device is typically utilized. The converter is made consisting of an active switch controlled by an IC, a rectifier, and filter components.. [29] This ease of usage enables the application to distribute low-cost, high-efficiency power. If the inductor current never falls below zero, the converter is said to be in continuous conduction mode. When the switch is turned on, the input voltage charges the inductor and biases the diode in the opposite direction. When the switch is turned off, the diode is forward biased and the inductor discharges to the load [30].

2.1.3 Buck-Boost converter:

The Buck-Boost circuit is a mixture of a Buck and a Boost circuit with a basic structure. Based on the switch component's switching time, their output voltage may be higher or lower than the input voltage [31]. When the switch is turned on, the diode is turned off, and the inductor is charged. The voltage, the switch, and the inductor will all enhance the current flow. When the switch is turned off, the diode is forward biased, and the stored energy is sent to the capacitor and load resistance by the inductor.

2.1.4 Sepic converter:

Due to the existence of input and output inductors, the DC-DC SEPIC converter has become one of the most popular topologies with continuous input and output currents. When compared to other switching converters, it has a high power density and a fast transient response. In order to develop a controller for a DC-DC SEPIC converter, four state variables must be sensed: input and output inductor currents, coupling capacitor voltage, and output capacitor voltage [32]. The SEPIC (Single-Ended Primary Inductor Converter) converter can do both step-up and step-down operations and has no polarity reversal issues. To adjust the converter's output voltage, feedback control is commonly added into the circuit, usually by Pulse Width Modulation (PWM) [33].

2.1.5 Cuk converter:

DC power supply commonly use Cuk converters that can operate in either step-up or step-down mode. Cuk converters are widely employed in wind energy, solar power systems, electrical vehicles, transmission and reception of radar signals, driver of light emitting diodes, communication systems, compressor and motor controllers, as well as energy absorption from workout bicycles. [34]. [35] shows a typical Cuk converter and its operational modes. It is made up of an input voltage source, a MOSFET switch, an anti-parallel diode, a freewheeling diode, a transferring and storing capacitor, two inductors, and a load resistor.

2.2 Zeta converter:

Zeta is a dc-dc converter topology that can offer output voltages that are higher or lower than the input voltage. Because it balances the same polarity on both the input and output sides, it is a non-inverting regulator. Previously, the Zeta converter paid the least attention to the higher-order converter's sophisticated design and calculations. However, after determining simple strategies in the applications, the researchers focused on the converter. As a spiraled power converter of the buck and boost converters, the zeta converter allows for improved transient response. This converter was dubbed a twin SEPIC converter by the previous researchers. Improved switching approaches attract attention because of their remarkable ability to reduce the size of resonant dc-dc converters while using lighter components. This idea sparked the idea of creating a dual SEPIC converter. The zeta converter's application was limited at the time, but the notion grew with many attempts, and the application spread throughout many industries [36].

2.2.1 Outline of Zeta converter:

The step-up and step-down capabilities of zeta converters are unaffected by polarity reversal, making them a versatile fourth-order DC-DC converter. When the requirement required a low input voltage and a high output voltage, fourth-order converters such as the zeta, CUK, and SEPIC converters were used [37]. The zeta converter, in contrast to the Cuk and SEPIC converters, is a non-linear, non-inverting fourth-order converter that can operate as buck-boost-buck for input and boost-buck-boost for energy output. Reduced switching stress, flexibility, and non-pulsating output current are all advantages of the zeta converter. It is determined by the duty cycle, which can be switched between boost and buck modes [38]. Zeta also surpasses SEPIC in aspects of ongoing maintenance a larger input voltage range, greater load transients, reduced output voltage ripple, and simpler adjustment [39]. Figure 1 depicts the zeta converter's main configuration:

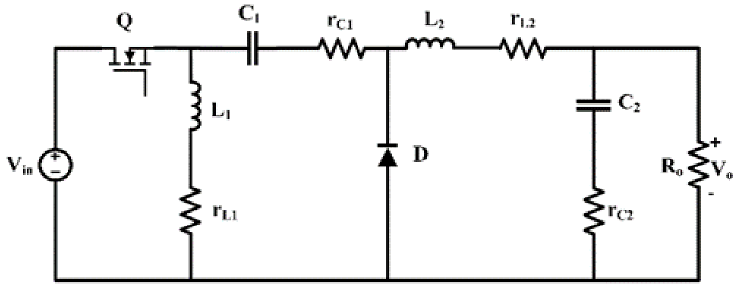


Fig. 2.1 Zeta Converter

2.2.2 Circuit analysis of Zeta converter:

The Zeta converter is a nonlinear fourth-order system made composed of an active power switch, a diode, two inductors, and two capacitors. The converter's output voltage is normally regulated using a PWM feedback control loop. For switching, PWM pulses are applied to the MOSFET. The converter provides either a greater or lower output voltage depending on the duty cycle of the PWM [40].

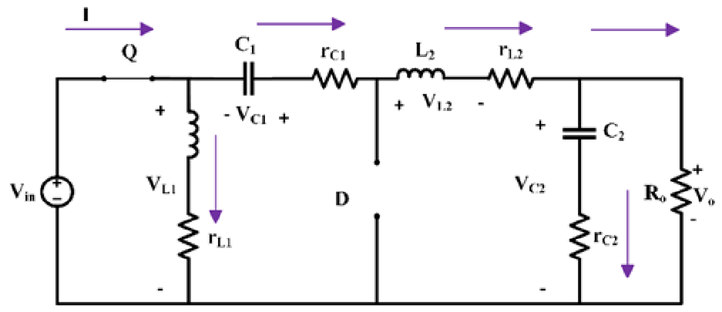


Fig. 2.2 Zeta Converter while Q is shorted

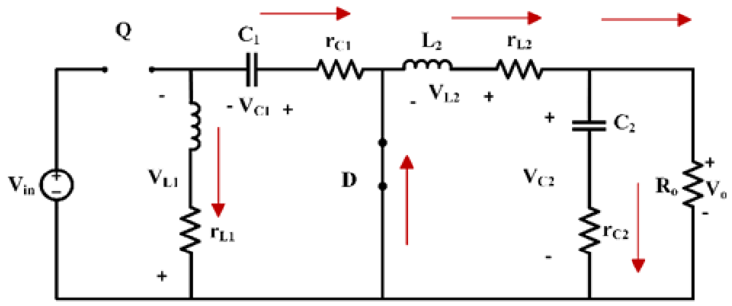


Fig. 2.3 Zeta Converter when Q is open

2.2.3 State-Space average method for zeta converter:

State-space averaging is an acclaimed approach for simulating switching converters (SSA). The equations for the amount of inductor current adjustment as well as the equations for the amount of capacitor voltage adjustment were used to create the state-space averaged model [41]. Switched converters are commonly described as piecewise linear networks, where within a conventional switching cycle, the topology changes at the boundaries between subsequent subintervals. The state of each switching device, such as a transistor or a diode, can be used to determine the relevant state-space equations. The state variables are frequently linked to the energy storage component [42].

There are two functioning modes for the zeta converter. The diode is open while the MOSFET is in the on state. As a result, - (Vin + Vo) is discovered across the diode in this interval. Furthermore, both inductors are in the charging phase, which means their current is increasing linearly.

The capacitor C1 will be discharged, and energy will be wasted at the resistor; as a result, V0 will rise, and L2 will be linked in series. The charging inductor's whole current passes through the MOSFET. The following are the operating mode equations [43,44]:

By KVL for shorted MOSFET in figure 2:

$$\frac{di_{L_1}}{dt} = \frac{1}{L_1}\left[-r_{L_1} i_{L_1} + V_{in}\right]$$

$$\frac{di_{L_1}}{dt} = \frac{1}{L_2}\left[-\left(r_{L_2} + r_{C_1} + \frac{r_{C_2}R_o}{r_{C_2} + R_o}\right)i_{L_2} + V_{C_1} - \frac{R_o}{r_{C_2} + R_o}V_{C_2} + V_{in}\right]$$

$$\frac{dV_{C_1}}{dt} = -\frac{i_{L2}}{C_1}$$

$$\frac{dV_{C_2}}{dt} = \frac{1}{C_2}\left[\frac{R_o}{r_{C_2}+R_o}i_{L_2} - \frac{V_{C_2}}{r_{C_2}+R_o}\right]$$

The MOSFET opens and the diode is shorted in the next phase of operation. Both L1 and L2 inductors will now be discharged. When the polarity of the voltage applied to the inductor is changed, the diode will become forward biased and begin to operate.

The voltage stored in L1 and L2 will be dissipated to capacitor C1 and the output resistor Ro at this point. As a result, both inductor currents fall in a linear fashion. By KVL,

$$\frac{di_{L_1}}{dt} = -\frac{1}{L_1}\left[-\left(r_{C_2} + r_{L_1}\right)i_{L_1} - V_{C_1}\right]$$

$$\frac{di_{L_2}}{dt} = \frac{1}{L_2}\left[-\left(r_{L_2} + \frac{r_2R_o}{r_{C_2} + R_o}\right)i_{L_2} - \frac{R_o}{r_{C_2} + R_o}\ V_{C_2}\right]$$

With the help of KCL, current flows via capacitor C1:

$$\frac{dV_{C_1}}{dt} = \frac{i_{L_1}}{C_1}$$

$$\frac{dV_{C_2}}{dt} = \frac{1}{C_2}\left[\frac{R_o}{r_{C_2}+R_o}i_2 - \frac{1}{r_{C_2}+R_o}V_{C_2}\right]$$

$$V_o = \frac{r_{C_2}R_o}{r_{C_2} + R_o}i_{L_2} + \frac{R_o}{r_{C_2} + R_o}V_{C_2}$$

The zeta converter's input-output voltage connection is as follows:

Applying KVL, for inductor L1 on-state and off-state respectively are VL1=Vin and

$V_{L_1} = V_{in}$

So, $V_{C_1} = \frac{D}{D-1}V_{in}$

Here's the equation for inductor L2 using KVL:

On state: $V_{L_2} = V_{in} - \left(V_{C_1} - V_o\right)$

Off-state: $V_{L_2} = -V_o$

After that, the overall equation will be:

V_{C1}=V_{in} -V_o D

Putting these equations together,

$$\frac{V_o}{V_{in}} = \frac{D}{1-D}$$

These formulas are used to create the SSA equations:

$$x'(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Bu(t)$$

Where, $A = A_1 D + A_2(1 - D)$

And, $B = B_1 D + B_2(1 - D)$

Let's assume the variables,

$x_1 = i_{L_1}$

$x_2 = i_{L_2}$

$x_3 = V_{C_1}$

$x_4 = V_{C_2}$

$u(t) = V_{in}$

State-space equation matrix,

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} -\frac{r_{C_2}(1-D)+r_{L_1}}{L_1} & 0 & \frac{(1-D)}{L} & 0 \\ 0 & \frac{(r_{C_2}+R_o)(Dr_{C_1}+r_{L_2})+r_{C_2}R_o}{L_2(r_{C_2}+R_o)} & \frac{D}{L_2} & -\frac{R_o}{L_2(r_{C_2}+R_o)} \\ \frac{(1-D)}{C_1} & \frac{-D}{C_1} & 0 & 0 \\ 0 & \frac{R_o}{C_2(r_{C_2}+R_o)} & 0 & \frac{-1}{C_2(r_{C_2}+R_o)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \frac{D}{L_1} \\ \frac{D}{L_2} \\ 0 \\ 0 \end{bmatrix} [u(t)]$$

So, $$A = \begin{bmatrix} -\frac{r_{C_2}(1-D)+r_{L_1}}{L_1} & 0 & \frac{(1-D)}{L} & 0 \\ 0 & \frac{(r_{C_2}+R_o)(Dr_{C_1}+r_{L_2})+r_{C_2}R_o}{L_2(r_{C_2}+R_o)} & \frac{D}{L_2} & -\frac{R_o}{L_2(r_{C_2}+R_o)} \\ \frac{(1-D)}{C_1} & \frac{-D}{C_1} & 0 & 0 \\ 0 & \frac{R_o}{C_2(r_{C_2}+R_o)} & 0 & \frac{-1}{C_2(r_{C_2}+R_o)} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{D}{L_1} \\ \frac{D}{L_2} \\ 0 \\ 0 \end{bmatrix}$$

$$C = \left[0 \quad \frac{r_{C_2} R_o}{r_{C_2} + R_o} \quad 0 \quad \frac{R_o}{r_{C_2} + R_o}\right]$$

$$D = [0]$$

A, B, C, and D, respectively, denote the system matrix, the control matrix, the output matrix, and the feed-forward matrix. x, u, and y are the status vector, input vector, and output vector, respectively. For the ON state, A1 and B1 are also stated. The OFF states are denoted by A2 and B2, respectively.

CHAPTER 3

STUDY OF SWARM INTELLIGENCE ALGORITHM (SIA)

Swarm Intelligence is an area of computational intelligence that studies the collective behaviour that emerges in self-organizing communities of agents. These are inspired by the customs and behaviours seen in biological creatures or organisms in their native environment in order to gain efficiency in satisfying natural necessities such as getting food, preying, overcoming danger, mating, and so on. These methods are data-driven issue optimization strategies that use intelligent computing approaches. The collective behaviour of such ecosystems, as well as their artificial counterpart of SI, is not stored in the set of rules that govern the movement of each isolated agent, but develops through the interaction of several individuals. These rules describe each agent's mobility by adjusting its movement in small time intervals in response to environmental factors that it detects. The SI algorithm is used to enhance the flexibility, robustness and resilience in algorithm that seek for the best solution to complicated situations.

3.1 Survey of Particle Swarm Optimization (PSO):

In 1995, Eberhart and Kennedy came up with the very first idea for PSO[45]. Kennedy and Eberhart's first views on swarms of particles were largely focused on the creation of intelligent calculation through leveraging essential social interaction analogues, instead of mere individual computation. As this algorithm follows the social interaction among the creatures in nature, it is a SI algorithm. Swarm intelligence, which employs a multi-agent population to search for and identify answers to optimization problems, is more precisely influenced by PSO.

3.1.1 Identification of PSO

PSO is a tool that uses stochastic, hybrid approaches, and artificial intelligence to solve a wide range of optimization issues. PSO, as a stochastic approach, uses a random probability distribution to generate random locations of possible solutions that may be assessed but not foreseen. Furthermore, since PSO is metaheuristic, it may identify search approaches that give potential answers to important issues even when the data set is inadequate or the processing capacity is insufficient. According to the researchers, particle swarm optimization is based on a very basic concept that may be implemented in a short code. It simply requires elementary numerical operators and is efficient in terms of memory and processing performance.

3.1.2 Objectives of PSO

The PSO method contains various goals that basically allow non-expert individuals to use the optimization technique to its maximum potential in order to accomplish their aim[46]. The objectives are stated below:

- To compile a detailed assessment of the most prevalent PSO variables in order to get long-term utility from it.
- To create an overview of the algorithm's theoretical features and the impact of the PSO parameters on those.
- To come up with new ways to enhance the algorithm's performance.
- To analyse a variety of algorithm applications and outcomes in order to establish the algorithm's functioning and current rules.

3.1.3 Features of PSO

For addressing continuous, non-convex, integer variable type, discrete, non-linear problems, Particle Swarm Optimization is among the most widely utilised search algorithms [47]. It is based on swarm intelligence, which is significantly impacted by colony or swarm rivalry and cooperation. When swarm animals work together to achieve shared community objectives, they exhibit several unique characteristics that have been included into the PSO algorithm to achieve optimal results. There are a number of helpful qualities in this optimization process that include interaction, communication, the exchange of knowledge, and collaborative decision-making. The optimization issue is evaluated using an objective function that is supposed to be minimised in our study. To establish their acceptability and effectiveness, the method's solutions are evaluated in terms of the objective functions.

3.1.4 Methodology of PSO

The method starts with a number of particles, each of which has been programmed to discover suitable outcomes in the search area[48]. The advantage of having a community of particles is that it reduces calculation time because everyone is looking for the same thing at the same time. Furthermore, the particles can travel to each point and corner of the search space to find different information that is useful in reaching conclusions.

In our thesis, the requirement, also known as the optimization problem, is replicated as the cost function in the algorithm that is designed to minimise the cost. The particles are now ready to survey the optimum solution defined by the programmer after defining the objective function. Each particle gains a unique optimal position, indicating the optimum solution.

After each particle's motion has been completed, the information is exchanged throughout the population in order to agree on a global optimal location. In each loop, these actions are repeated, and the values are changed accordingly. The velocity of the particle in the current iteration, the particle's personal best position, and the global best solution of the entire population are three factors that are critical in the algorithm to regulate and decide the movement of the particles. The three vectors mentioned above are used in the statistical model to calculate the particle's displacement and velocity in the following iteration. As a result, the computation took into account both personal and general experiences.

The revised forms of the PSO algorithm get more optimised as the number of iterations increases. To improve the algorithm's efficacy, more parameters are used in the equations for computing particle movement. In the equations, random functions are used to ensure that the parameters are not locked in the same area. The equation also includes acceleration coefficients, allowing the agents to accelerate toward their personal and global optimal values. However, when choosing the value systems of the constants, a balance must be maintained because a higher quantity compels the particles to attain toward such or beyond the desired locations quickly, whereas a lower quantity causes particles to pass the target locations while travelling in the search area without even being called back[49]. To bring balance between both the individual and global optimum outcomes, a parameter called inertia weight is applied with the particle velocity, which varies between 0.9 and 0.4. A higher inertia weight value indicates a preference for a worldwide survey, whereas a lower value indicates a preference for a local survey. Finally, the procedure is ended after the required iterations has been accomplished.

3.1.5 Exploitation and exploration in PSO

Exploitation relates to the algorithm's intensity, while exploration relates to the algorithm's diversification. Both of these characteristics may be seen in this algorithm. The relevant information from the optimization problem is usually utilised for exploitation. To identify the best bargains, our approach focuses on local searches. Random functions, on the other hand, aid in the investigation of the search space.
Although the globally best solution is used for choice in the accelerated particle swarm optimization the implications of the personal optimal solution remain unclear in the PSO algorithm[50]. Due of the lack of crossover, the PSO algorithm exhibits increased mobility while exerting vigorous exploration. However, a balance is tried between exploitation and exploration in order to increase the convergence rate and avoid early convergence with non-profitable solutions.

3.1.6 Mathematical model of PSO

PSO's mathematical model describes following the movement of the particles. The particles' mobility is determined by their displacement and velocity in the search space.

The conventional Particle Swarm Optimization algorithm's equation for determining particle velocity is shown below. [51]

$$v_n^{t+1} = w * v_n^t + c_1 * F_1 * (p_{nb} - d_n^t) + c_2 * F_2(g_b - d_n^t) \tag{3.1}$$

The displacement of the particles is computed using the following equation

$$d_n^{t+1} = d_n^t + v_n^{t+1} \tag{3.2}$$

The terms carry out the following representation:

w= Inertia weight

c_1,c_2= Acceleration Constants

F_1,F_2= Random Function

d_n^t= Movement of the n^{th} particle in the t^{th} iteration

d_n^{t+1}= Movement of the n^{th} particle in the $(t+1)^{th}$ iteration

v_n^t= Speed of the n^{th} particle in the t^{th} iteration

v_n^{t+1}= Speed of the n^{th} particle in the $(t+1)^{th}$ iteration.

p_{nb}= Personal best value of the n^{th} member

g_b= Global best value among the entire community.

3.1.7 Flowchart of PSO

The chronological course of the functions and the algorithm's calculation is given throughout the steps below:

Step 1: Generate a community of particles to begin the process.

Step 2: Define the fitness function that will be used to solve the optimization issue.

Step 3: Using the prior location information, calculate the updated velocity of each particle.

Step 4: Determine the new location.

Step 5: Analyse individual particle's own best response.

Step 6: Calculate the global best value by comparing the individual best solutions of all particles.

Step 7: Execute the method for the number of repetitions for the threshold value.

Step 8: If the entire number of repetitions is not reached, go back to step 3 and repeat the same process.

Step 9: Terminate the process, when an optimal solution is found.

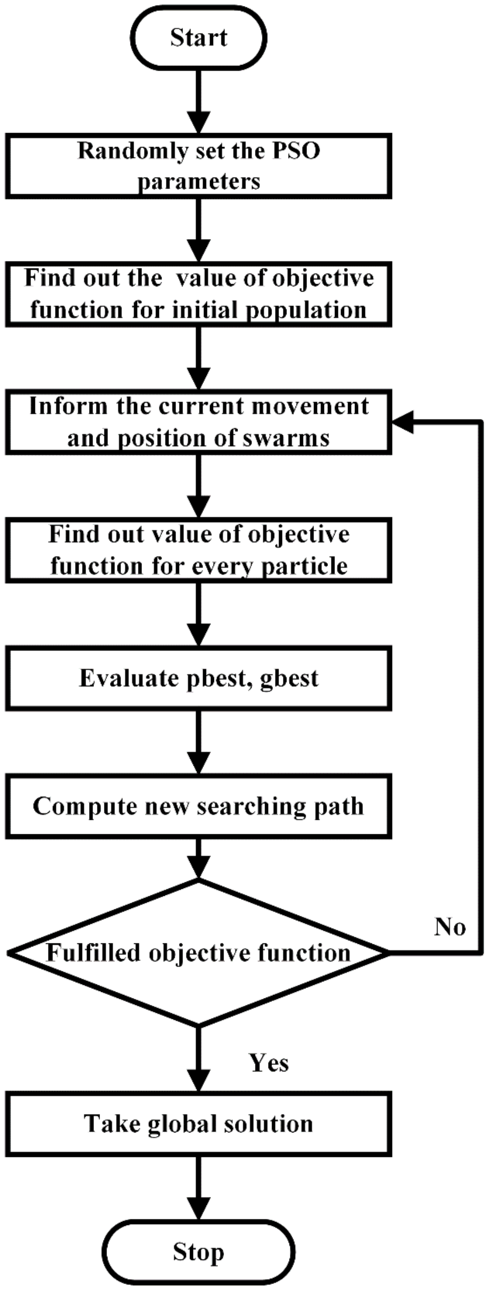


Fig. 3.1 Flow Chart of Particle Swarm Optimization Algorithm[52].

3.2 Survey of Firefly Algorithm (FA):

In 2008, the Firefly Algorithm was proposed as a swarm intelligence (SI) based algorithm[53]. The dynamic colony behaviour of fireflies served as an inspiration for this algorithm's optimisation. Fireflies have glowing abdomens as a consequence of a biological procedure termed as bioluminescence. Their ability to generate light is an important aspect of their social ritual. Fireflies use their flaming body as signals to find food, find refuge, and defend their group from attacks. In this way, the correct use of group knowledge to achieve a common purpose is demonstrated. The aforementioned behaviour is important in the method used to find the best solutions in a search space.

3.2.1 Identification of FA

The Firefly Algorithm is an optimization technique that is based on metaheuristics, stochastics, and artificial intelligence[54]. By its very nature, metaheuristics lead the production of optimum or nearly optimised solutions via an iterative process. In addition, it is non-deterministic and uses an approximation strategy to solve problems[55]. FA may solve poorly specified issues with imprecise data sets since it is also a stochastic process.
Furthermore, the firefly method powered by swarm intelligence (SI) employs a multi-agent body with a decentralised approach to sweep the search area, implying that individuals move independently of any outside coordinator[56]. The SI method also exhibits strong, adaptable, and self-orchestrated qualities that easily apply FA[57]. All of these features together make the FA an ideal match for optimization and promising outcomes.

3.2.2 Features of FA

The Firefly Algorithm is an optimization method that relies on metaheuristics, stochastics, and artificial intelligence. By its very nature, metaheuristics lead the production of optimum or nearly optimised solutions via an iterative process[54]. In addition, it is non-deterministic and uses an approximation strategy to solve problems. FA can solve poorly defined issues with imprecise data sets since it is a random model.
The Firefly Algorithm, which was influenced by colonial firefly' flashing habits, maintains certain basic criteria for carrying out the optimization operation. As such, a short description of the regulations is given below[58]:
i. FA doesn’t care about fireflies’ sex.
ii. The only thing of attraction is the brightness of fireflies.
iii. The brightness of firefly maintains inverse square law with distance.
iv. For fireflies of the same light level, randomized mobility will be guaranteed.
Brighter or more appealing fireflies indicate better solutions in the algorithm[59]. As a consequence, the procedure requires that the less brilliant firefly constantly travel towards the brighter ones in order to get better outcomes. Agents will be able to migrate from locations with no optimal solutions to areas with optimised solutions as a result of this process. To regulate the movement of people towards ideal outcomes, FA employs a host of aspects, including the random sampling factor, the co-efficient of attraction, and the co-efficient of light absorption.

3.2.3 Methodology of FA

In the algorithm, the problem to be solved is transformed into an objective function. The objective function is linked to the brightness of the firefly, which is designed to be as low as possible in order to achieve our goal. As brighter fireflies indicate ideal solutions, this confirms the acquisition of better solutions. The free movement of the firefly, which is governed by the attraction factor of relevant fireflies, distance between them, random functions, and their capacity to absorb light, results in the production of viable solutions[58].
The method starts with a population of fireflies that are ready to explore the search space that the programmer has already defined. To produce a solution, each agent achieves a place in the search region by following the attraction of the other firefly. Both brightness and distance are taken into consideration while pursuing the alluring firefly. Similarly, the light absorption spectrum and the randomization factor play a role in determining the location.

After then, the beautiful firefly's current posture is evaluated in terms of attractiveness. If the new position's attraction is higher than the previous one, the related position is utilised to update the current one. If the attraction does not increase, the firefly will remain in the same place.
While adjusting the locations of the firefly and concurrently seeking better alternatives, the process is halted once the specified number of repetitions are completed.

3.2.4 Exploitation and exploration in FA

The Firefly Algorithm (FA), which is driven by swarm intelligence, exhibits two distinct behaviours that substantially influence the algorithm's output: exploitation and exploration.
Exploitation refers to the characteristics of a local survey in which data from the optimization issue is used to generate new solutions. In FA, attractiveness is linked to the issue to be addressed, hence the co-efficient of attraction is crucial in exploitation.
Exploration, on the other hand, is associated with the notion of searching the full search region for various answers. The investigation is aided by the randomization element and random vectors.
To guarantee the algorithm's efficient performance, a balance between exploitation and exploration must be maintained[60]. The parameters for both processes are modified based on requirement, with neither being reduced nor maximised.

3.2.5 Mathematical model of FA

The following equations may be used to calculate the location of each firefly in the solution space[60].

$$X_n^{t+1} = X_n^t + \rho(X_b^t - X_n^t) + \alpha r \tag{3.3}$$

The equation below is used to calculate the co-efficient of appeal.

$$\rho = \rho_o * \exp(-\eta * h_{bn}^2) \tag{3.4}$$

Here b is the brightness of the n^{th} firefly. It is evaluated by using the following equation.

$$i(h_{bn}) = \frac{i_s}{h_{bn}^2} \tag{3.5}$$

The symbols utilised in the above equations are as follows:

X_n^{t+1}= Position of Firefly n in the current state.

X_n^t= Previous state of Position of Firefly n.

h_{bn} = Firefly b and Firefly n are separated by this relative distance.

ρ_o= C-efficient of attraction of the firefly when relative distance is zero.

α = Randomization factor.

η = Co-efficient of absorption of light.

r = Boundary of sear area specified by a random value vector which is uniformly distributed at time t.

i_s = Very first brightness of the firefly.

3.2.6 Flowchart of FA

A flowchart is used to explain the algorithm's process in a chronological order. The FA's steps are shown in the diagram below.

Step 1: To begin, create an arbitrary population of fireflies to serve as the algorithm's starting point.

Step 2: Get the minimum value of the objective function.

Step 3: Evaluate the co-efficient of attraction.

Step 4: Calculate the state of every firefly.

Step 5: Compare the attractiveness of the firefly with respect its previous state.

Step 6: Update the new position.

Step 7: Position won't be updated if the brightness is better.

Step 8: Iteration is completed based on the stopping criteria or again start from step 3.

Step 9: Terminate the process.

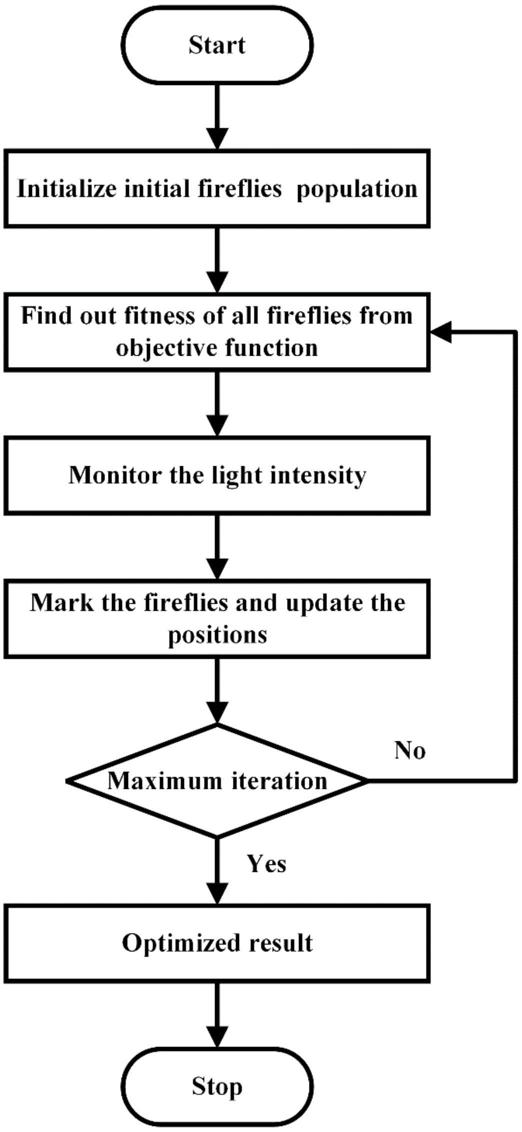


Fig. 3.2 The flowchart of Firefly Algorithm[61].

3.3 Survey of Artificial Bee Colony (ABC)

Karaboga first put the idea on an artificial bee colony algorithm in 2005, which follows the swarm intelligence principle[62]. The ABC algorithm is based on the cooperative and sociable behaviour of honey bees as seen in their everyday activities while looking for nourishment. Honey bees are clever and energetic insects that work together and tirelessly to ensure that the group has enough food. Artificial bees must choose food sources based on the quality of the food, according to the algorithm. The bees' aims and actions are classified, and each category fulfils its duty while also relating information to the other categories in order to achieve maximum efficiency in achieving their shared goal. The ABC algorithm uses a similar technique to identify optimal solutions to a related issue. The algorithm contains a small number of control parameters, making it simple to govern the output by fine-tuning the parameters as needed[63].

3.3.1 Identification of ABC

Artificial Bee Colony is a metaheuristic, stochastic, and artificial intelligence-based algorithm, similar to the previously stated methods[62]. As a result, this algorithm maintains the potential of constantly discovering efficient approaches to problems with little knowledge via an iterative process. The algorithm's optimised solutions are identified by food sources with a large amount of nectar detected by the bee colony. One of the algorithm's beneficial features is that it calculates the likelihood of the created food locations in terms of fitness, allowing for quicker convergence[64]. Furthermore, the greedy selection approach is used to assess both previous and current solutions and save the better of the two[65]. The algorithm's numerous evaluations increase the likelihood of obtaining optimum outcomes. The ABC algorithm incorporates the bee colony's operations, which include looking for solutions, comparing the results, picking the best quality solutions in terms of fitness, eliminating low-quality food sites, and lastly developing improved solutions[64]. The bees' entire communication process is carried out in the form of a ritual dance that takes place in the search area, making the data visible to other bees. Various parameters are tweaked to improve the algorithm's applicability for the situation at hand.

3.3.2 Stratification of bee based on their functions:

The honey bees are divided into 3 groups according to their jobs[66].

i. Employee:

The employed bees are actively involved in the hunt for potential food sources. Each hired bee locates a suitable food source and records relevant data about it, such as the location, direction, and quality of the food supply. Following that, the suitability of the food sources is evaluated in order to provide a likelihood of the solution while also communicating with the community.

ii. Onlooker:

While the wiggle dance takes place in the search area, the spectator bees analyse the produced solutions in terms of fitness and pick the optimal solution. In the next step, these possible solutions will be used to update the food sources. Food sources that do not give viable solutions are deleted, and the employed bees are reclassified as scout bees.

iii. Scout:

Scout bees are created to make up for the inadequate solutions supplied by hired bees in the past. The scout bees scour the search area for food sources that the hired bees have overlooked. Scout bees, on the other hand, seldom come into highly optimised food sources. The scout bees' food sources are being re-evaluated to determine whether they can be turned into commercial solutions.

The ABC algorithm does both local and worldwide investigations by using employed and spectator bees for the local aspect and scout bees for the global element[66]. This enables a comprehensive examination of the search space as well as a high rate of convergence.

3.3.3 Methodology of ABC

The ABC algorithm is completed in three steps by three different types of honey bees. The optimization issue is expressed as an objective function that is proportional to the quantity of nectar in the food sources, indicating that the solutions are more suited. Our work's goal function is set to be as little as possible.

The algorithm begins by creating a colony of employed honey bees, each of which provides a potential food site. They keep track of the location, direction, and food quality of food outlets. After that, the solution's fitness is evaluated in terms of nectar in the food sources, and the probability of the solutions is calculated. The better solutions are saved by comparing previously discovered solutions to freshly discovered ones using the 'greedy selection technique.' Later, the probability data and solutions are exchanged in a ceremonial dance with the observer bees in the search arena.

The quantity of spectator bees equals the employed bees. After considering the likelihood of fitness solutions, the observer bees choose the candidate food sources. The optimum solutions that have been chosen are then utilised to update the new food sources.

Some food sources don't supply viable options, thus they're removed from the search. To compensate for the missing solutions, the hired bees convert into scout bees and explore the search space. They are concerned with locating any source of high-quality food. As a result, the scouts provide low-cost search services and low-quality food sources. Scouts are only sometimes successful in finding valuable food sources. A limiting parameter governs the shift of employed bees to scout bees. The limiting parameter specifies that if the answers do not improve after a certain number of trials, they are eliminated[63]. The method is run using a pre-determined number of iterations, commonly known as the maximum number of cycles (MCN). The algorithm's technique ensures a reliable, simple, and quickly search tool.

3.3.4 Exploitation and exploration in ABC

To maintain a balance between faster convergence and identifying optimum solutions in the search space, the algorithm uses both exploration and exploitation at the same time.

Employed bees and spectator bees both engage in exploitation since they use the data supplied for the goal functions, which ensures a higher quality of solution. Scout bees, on the other hand, carry out the exploration by surveying the total search space for fresh options. In general, scout bee exploration is fairly successful, although convergence may be sluggish since crossover is not present in this process, and so exploitation capacity is restricted[67]. The settings governing the hired bees and scout bees may be tweaked for greater balance.

3.3.5 Mathematical Model of ABC

The equation to calculate the optimal solution[68]

$$X_{nm} = x_{nm} + \varphi_{nm}(x_{nm} - x_{km}) \tag{3.6}$$

Where m ∈ 1,2,3,4.........N (3.7)

And, $k \in 1,2,3 \ldots \ldots \ldots \ldots . . D$ (3.8)

The probability of individual obtained solution with respect to fitness function using the following equation:

$$Probability = \frac{fitness_n}{\sum_{n=1}^{N} fitness_n} \tag{3.9}$$

The updated solution by the scout bees are found by the equation below[68]

$$x_n^{m(new)} = x_{min}^m + rand()(x_{max}^m - x_{min}^m) \tag{3.10}$$

Here, $m \in 1,2,3, \ldots \ldots \ldots . D$ (3.11)

The description of notations used here are given below.

N = Population of Employed bees = Population of onlooker bees.

X_{nm} = New optimal solution

x_{nm} = Initial optimum solution.

φ_{nm}= Random value ranging from -1 to 1.

$Probability_n$ = Probability of individual solution according to fitness function.

$fitness_n$ = Fitness of each solution n.

$x_n^{m(new)}$ = Updated soltution by scout bees.

x_{max}^m = Maximum limit of parameter $j \in 1,2,3 \ldots \ldots D$

x_{min}^m = Lower limit of parameter $j \in 1,2,3, \ldots \ldots . D$

rand() = Array of random numbers which are uniformly distributed between 0 to 1

D = Optimum parameter number of each solution.

3.3.6 Flowchart of ABC:

The approach of the ABC algorithm is described in the following steps.

Step 1: Start with a population of honey bee.
Step 2: Announce the objective function relates to nectar amount of food source.
Step 3: Discover the source of food for every employed bee.
Step 4: Calculate the fitness function.
Step 5: Find out the probability of the solution of fitness function.
Step 6: Using greedy selection process to gather better solution.

Step 7: Using the potential solutions, update the new food sources.
Step 8: Monitor the threshold value. If the nectar is not that much, discard the source of food.
Step 9: Locate a new source of food by scout bee and observe the number of iterations.
Step 10: If the stopping criteria is satisfied, stop the iteration or go from step 3 again.

Now the steps are minutely depicted in following flowchart[69].

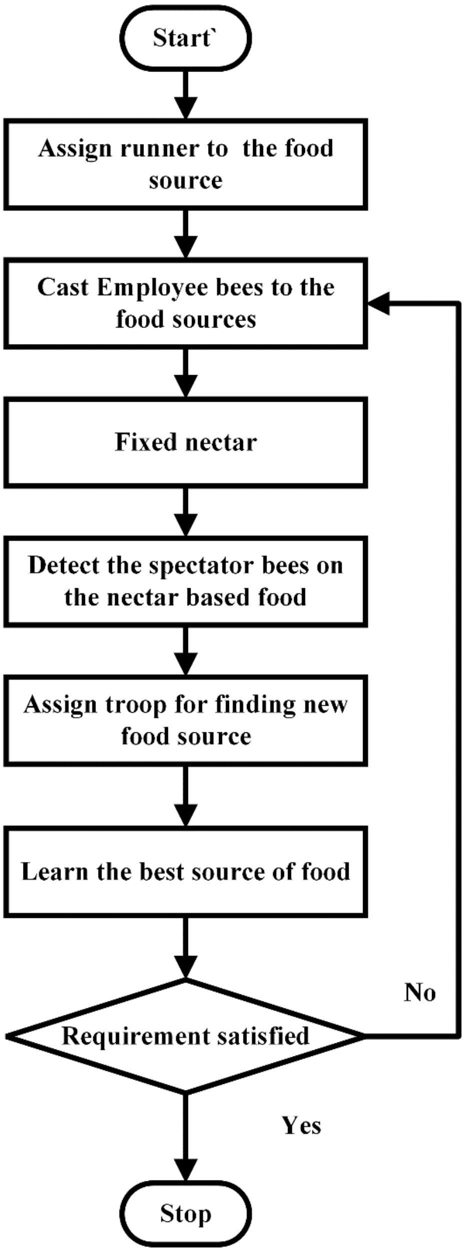


Fig. 3.3 Flowchart of Artificial Bee Colony[69].

3.4 Survey of Ant Colony Optimization for Continuous Domain (ACOR)

Ant Colony Optimisation was first developed in 1992 by Dorigo, and it has since been utilised to tackle a variety of combinational optimization issues involving a collection of discrete choice variables[70]. Socha and Dorigo suggested the notion of using ACO to solve the continuous optimization issue in 2008. It is influenced by the foraging behaviour of ant. Ants are very social. They work in a group on various purpose. While doing any job they communicate among themselves using some kind hormone namely- pheromone. A pheromone is a chemical produced by one animal that causes another animal of the same species to behave differently[70]. It is sometimes referred to as "behavioural-altering agents." Many individuals are unaware that, in addition to sexual activity, pheromones may cause other behaviours in animals of the same species.

3.4.2 Exploitation and Exploration in ACOR:

Often ACO is termed as a pseudo-random proportional method. Ants move from node to node on the basis of starting value and the updated value. The state transition rule shows the path of balance between exploration and exploitation[71].
For this several experiments are executed. Like pheromone intensity reinforcement on the spans of the trip of the ants who identified the best solutions. In this sense, it's possible to say that ants can deposit more pheromone. But this transition rule often converges early. The reason for this is the influence of pheromone intensity which will play a major part in an ant's decision-making process when selecting the next node to visit, whereas heuristic information will have little bearing[71].

3.4.3 Mathematical Model of ACOR

The multi-promising region of search area has been modelled using a Gaussian kernel pdf. In comparison to a mere Gaussian function, a Gaussian kernel pdf provides a more flexible sampling shape. Solutions are kept in ACOR in a file called solution archive. At the beginning of the program, the solution archive is started by arbitrarily creating n entries. Following the collocation of all solutions, k solutions are kept and the remaining are eliminated. In this regard, the stated standard deviation should be calculated as follows:

$$\sigma_l^i = \xi \sum_{e=1}^{n} \frac{|s_e^i - s_l^i|}{n-1}; \quad i = 1,2,3 \dots \dots n$$

(3.12)

Here, ξ remains unchanged in al l dimension and has an impact on evaporation rate of ACOR[72].

3.4.4 Flowchart of ACOR

Step 1: For each timespan, divide the storage volume into classes.

Step 2: Generate pheromone trail and other factors.

Step 3: Initialize every ant at starting point

Step 4: For every ant apply the transition rule.

Step 5: Find out the best the path to follow on the basis of pheromone.

Step 6: For each cycle calculate the local and global value.

Step 7: If the end condition is satisfied, terminate or repeat from step 3.

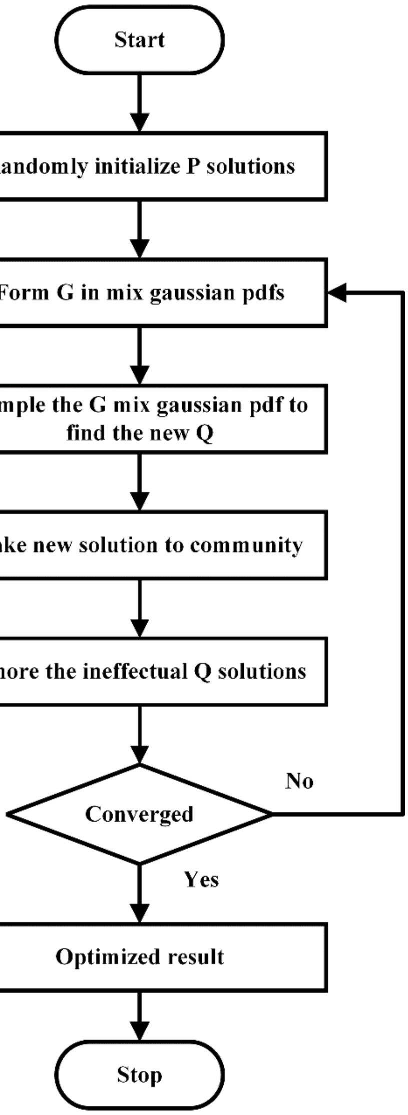


Fig. 3.4 Flowchart of ACOR[71].

3.5 Survey of Shuffle Frog Leaping Algorithm (SFLA)

In terms of taxonomy, the Shuffled Frog Leaping Algorithm (SFLA) is one of the most inventive optimization algorithms inspired by the social behaviour of frogs in nature, and it is listed among behavioural algorithms or Memetic Algorithms. The frog jumping algorithm, also known as the SFLA algorithm, is another term for the frog jump optimization techniques. Eusuff and Lansey developed this technique in 2003[73]. This is a kind of combination of PSO and memetic algorithm[73]. That's why it is simple, fast and efficient global optimizer. Each frog aims to bring more food. This method is done by calculating the fitness value for each frog and sort out the frogs according to their fitness value. The minimum value frog is the best one.

Then local search is performed. After that a new position is found and for this value again we will get new fitness value. Last of all the frogs are sorted out globally.

3.5.1 Identification of SFLA

The SFLA algorithm is a Meta - heuristic optimization memetics-based algorithm. For difficult and substantial optimization issues, the memetic algorithm is a population-based approach. The fundamental concept behind this technique is to increase the overall performance of the search intensification process by using a local search strategy inside the structure of an evolutionary algorithms. The memetic algorithm conceals the total of the original responses, then analyses the utility of each response using a fitness function and creates new solutions using this method. The SFLA algorithm is based on the way frogs find food. This technique searches locally among frog subgroups using the Nomo metric approach. The frog hybrid jump method employs a hybrid technique to enable communication protocol in local search[73]. Particulate cluster optimization and the Nomo metric method's benefits are combined in this algorithm. Words are exchanged not just in local search but also in global search in the frog hybrid jump algorithm. As a result, this algorithm effectively combines local and global searches. The frog hybrid jump method is simple to develop and highly accessible. Many nonlinear, undetected, and multi-state issues can be overcome with the frog composite leap method.

3.5.2 Features of SFLA

SFLA is a system that is based on a swarm of frogs jumping in a swamp, looking for the largest food amount attainable, in which the swamp includes many stone blocks at separate spots that make it simpler for the frogs to find the increasing food volume accessible. The goal is to find a stone that has the most food accessible. As illness spreads among frogs, communication between them might advance their memes. As memes develop, each frog's location will be altered by fine-tuning its jumping stride. The swarm is intended to be a collection of solutions depicted as a swarm of bees[74]. Frogs that are divided into subcategories called memeplexes. Each memeplex represented a different meme. All frog colony has its own objective to complete. Each frog has its own set of thoughts.

The memeplex may be convinced by the ideas of other frogs and advance via memetic evolution process. Concepts are transmitted across memeplexes precisely after numerous memetic processes in a shuffling process.

3.5.3 Methodology of SFLA

Since it integrates the benefits of two evolutionary algorithms (particle swarm optimization and shuffling complex evolution algorithm) concurrently, SFLA is a potential and dependable way for enhancing project planning solutions. The SFLA that has been created might assist construction teams in finding better and more effective planning solutions that are shorter in length, cheaper in cost, and have less resource volatility[74]. To evaluate the updated SFLA model's optimization capabilities, it was used in a real-world situation and compared to a commonly used approach.

At the very first, frogs are moving randomly within a boundary. Then they are sorted out on the basis of their fitness value. They are grouped in m memeplexes, where containing n frogs. All the members of that group is sorted out globally. Acquiring the information, the position is updated time to time.

3.5.4 Mathematical Model of SFLA

The mobility of each frog can be evaluated by the following equations.[75]

Define the memeplexes of m number and frog number in every memeplex is n. Hence total population is m*n.

P_B is the best location of a frog and P_w is the worst location.[74]

For positive feedback step size, S = lowest {int [rand (P_B – P_w)], $S_{highest}$} (3.13)

For negative feedback step size, S = highest {int [rand (P_B – P_w)], $S_{highest}$} (3.12)

For next iteration new position = P_w + S (3.13)

3.5.5 Flowchart of SFLA

Step 1: The population F is produced at random inside the search region. Where F = m*n, m is the number of memeplexes and n is the population in it.

Step 2: Fitness function value is calculated and sorted out.

Step 3: Within the amount of N iterations, all memeplexes execute memetic evolution.

Step 4: Best position frog and worst position frog are located.

Step 5: Find better fitness value for the worst position and replace it.

Step 6: Record the updated position.

Step 7: Check the condition whether it is satisfied or not.

Step 8: If the condition is satisfied then finish iteration otherwise again go from step 3.

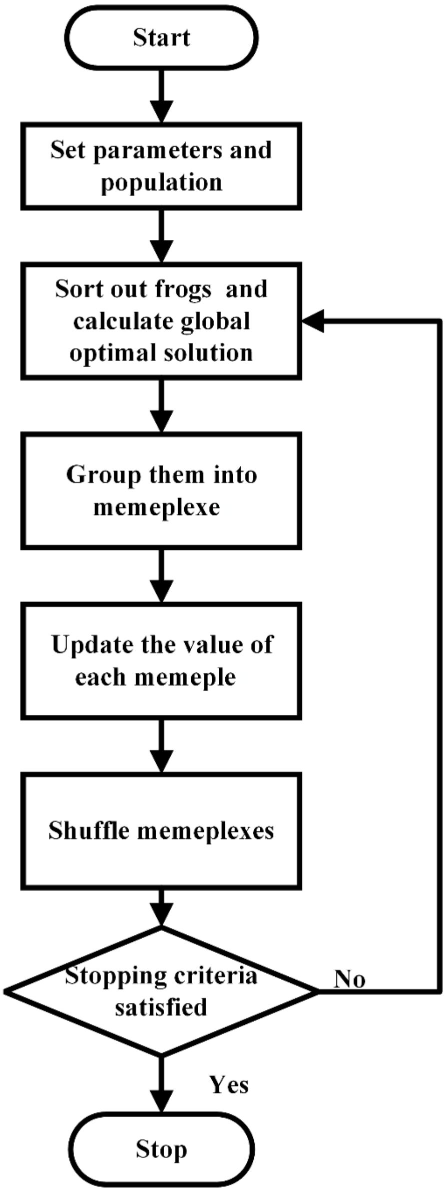


Fig. 3.5 Flowchart of SFLA[75].

CHAPTER 4

IMPLEMENTATION OF SWARM INTELLIGENCE ALGORITHM BASED PI CONTROLLER FOR DC-DC CONVERTER

The Power Electronics has introduced revolutionary change in conversion, regulation and control of power flow of regulator from linear to switching stage. Footprint can be seen in every power management application. LED drivers, laptops and computers, electric cars, hydropower plants, solar systems, and many more applications benefit greatly from DC-DC converters[76-79]. However, owing to the action of the switches, these converters have nonlinear temperaments, displaying larger spike in output voltage and overshoot[80-82]. As a result, several control strategies focused on modulating the output voltage are used in DC-DC converters to achieve improved performance[83]. PI control is one of the most well-known and widely utilised of the numerous control approaches for power converters. However, using the traditional technique, determining the precise value of the PI parameters is difficult and a bit clumsy. As a result, SIA is coupled to the PI controller in order to get best suitable PI parameter values for the closed-loop evaluation of the DC-DC converter's stability.

4.1 PI Controller

A PI controller produce output using proportional and integral mechanism. Feedback control is a control technique that puts the output value in check by sending the mistake back to the controller and directing the input toward a more accurate system. Positive and negative feedback control systems are the most common forms of feedback control systems. The size of the input is raised in positive feedback by joining the output to the input. In the case of negative feedback, the size of the input is reduced by reducing the output from the input. As a result, positive feedback enhances the amplifier's gain whereas negative feedback decreases it[84]. Many sets strict, such as PI, PD, and PID controllers, are built on this feedback mechanism for safe and efficient facilities.

In latest days, the notion of PI has been extensively applied in industrial operation feedback control. The topic of automated drive system saw the first theoretical study and practical implementation[85]. It was later employed in the industrial sector for automated control, where it was extensively applied in hydraulic and subsequently digital controller[86]. The PI principle is now widely typically used in applications that need precise and optimal automated control[87-89].

4.1.1 Outline of PI Controller

A control loop feedback mechanism called a PI controller is made up of two parts: proportional and integral. As a result, it's also known as a two-term controller, since the terms are tweaked using exploratory approaches such as analytical methods and other optimization techniques including PSO, FA, ABC, ACOR and SFLA.

The equation of output for PI controller can be expressed as below[90]:

$$u(t) = k_p\, e(t) + k_i \int e\,(t)dt \qquad (4.1)$$

The transfer function of the PI controller is stated as,

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} \qquad (4.2)$$

In this case u(t) is termed as controller output. Error, e(t) = r(t) -y(t) signifies standard and y(t) denoted as calculated process variable. Here, best value of k_p and k_i will be determined.

4.1.2 Tuning of PI controller

Tuning a PI controller entails adjusting the gain values of proportional and integral terms in order to improve the system's responsiveness by removing steady-state error, decreasing overshoot, and achieving a rapid rising and settling time. As a result, the impact of the adjusting settings is as described in the following,

i. Proportional term(P)

P is based on the current error, while k_p has an impact on both the rising time and the steady state error. While total eradication is impossible, it is feasible to lower the rising time and the steady-state inaccuracy[91]. P's mathematical expression is as follows:

$$p = k_p e(t) \qquad (4.3)$$

ii. Integral term(I)

I counts on a collection of previous mistakes. In return for making the transient response slower, ki may be tuned to reduce steady-state inaccuracy[91]. I is expressed mathematically as

$$I = ki \int e\,(t)dt \qquad (4.4)$$

Table 4.1 Effects of Tuning Parameter of PI Controller on a DC-DC Converter

Type of controller	Rise Time	Overshoot	Settling Time	Steady State Error
Proportional	Decrease	Increase	Small Change	Decrease
Integral	Decrease	Increase	Increase	Eliminate

4.2 Objective Function

The desired function is a legitimate function that is maximised or reduced to improve the performance of a system. As a result, several integral performance functions such as IAE, ITAE, ISE, and ITSE are employed to increase reliability while decreasing steady-state error in this scenario[39].

i. Integral Absolute Error (IAE)

The IAE method is based on the integration of absolute error over duration, with no consideration given to mistakes in a system in response.

$$IAE = \int_0^{\tau} |e(t)| dt \tag{4.5}$$

ii. Integral Time-weighted Absolute Error (ITAE)

ITAE is nothing but a time multiplication of IAE. Here errors are weighted. The mathematical equation for ITAE,

$$ITAE = \int_0^{\tau} t.|e(t)| dt \tag{4.6}$$

iii. Integral Squared Error (ISE)

ISE is the integration of the square of the error during time interval. Here, large errors are targeted to eliminate. The mathematical representation is given below.

$$ISE = \int_0^{\tau} e(t)^2 dt \tag{4.7}$$

iv. Integral Time Squared Error (ITSE)

ITSE is the multiplication of time with ISE. So, squared error will be times with one more time. The mathematical expression is followed by,

$$ITSE = \int_0^{\tau} t.e(t)^2 \, dt \tag{4.8}$$

4.3 Layout of SIA based PI Controller

The SIA method is used to perform closed-loop stability analysis on DC-DC converters, and a PI controller is adjusted by optimal values of K_P and K_I. This procedure's fundamental layout is seen in Fig. The error is continuously regulated, resulting in a compatible output[92]. The performance of the fitness function is evaluated for this purpose utilizing four error formulae (IAE, ITAE, ISE, and ITSE) performed by SIA. As a consequence, the optimal solution achieved via this procedure produces satisfying outcomes.

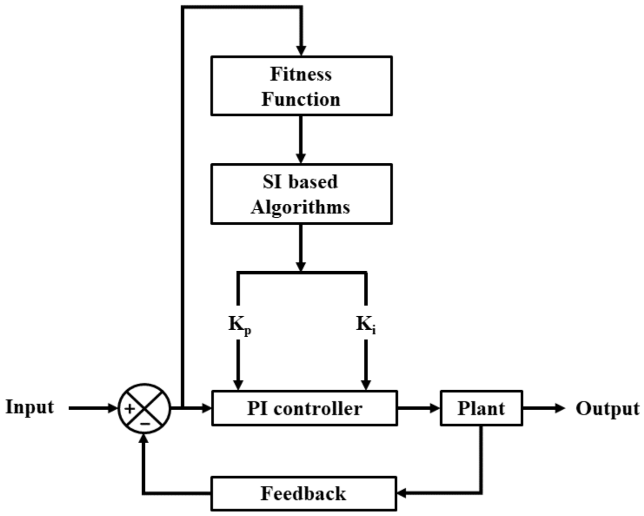


Fig. 4.1 Layout of optimized PI Controller.

CHAPTER 5

SIMULATION RESULTS AND PERFORMANCE COMPARISON

The conducted simulations using MATLAB for the stability study of the zeta converter are based on the foregoing mathematical modeling and structure of the optimized PI controller. The value of the step response's properties such as percentage of overshoot, rising time, settling time, and peak amplitude is used to assess the performance of the optimized PI controller. Initially, the functionality of the conventional PI controller is analyzed by the system's response. Subsequently, utilizing SIA, the gain values of the PI controller are adjusted to provide improved functionality. Lastly, for assuring relatively better stability of DC-DC converter, a comparative study has been performed in order to choose the appropriate PI controller.

5.1 Transfer function:

A transfer function is a mathematical description of the frequency connection between a system's output and input. Based on the categorization of the control system, a transfer function can be divided into two types: open-loop and closed-loop transfer functions.

5.1.1 Open-Loop Transfer Function:

Open loop system can be defined as a system in which the output has no influence over the input's control technique. In our thesis work, The zeta converter's circuit is primarily designed as an open-loop system. The transfer functions of such systems are referred to as open-loop transfer functions. As a result, equation 5.1 represents the transfer function of an open-loop zeta converter.

$$T(s) = \frac{974.7s^3+4.528e07s^2+9.141e10s+2.222e15}{s^4+6804s^3+1.693e08s^2+4.241e11s+2.398e15} \tag{5.1}$$

5.1.2 Closed-Loop Transfer Function:

A closed loop system is one in which the control actions are reliant on the output while the applied input is controlled. Following the open loop analysis, the circuit of zeta converter is incorporated with PI controller to converter it into a closed loop system. The transfer functions of such systems are known as a closed-loop transfer function. consequently, equation 5.2 is used to represent the transfer function of the closed-loop zeta converter.

$$T(s)= \frac{1.864e05s^4+8.663e09s^3+1.759e13s^2+4.252e17s+4.968e18}{s^5+1.932e05s^4+8.833e09s^3+1.801e13s^2+4.276e17s+4.968e18} \quad (5.2)$$

5.2 performance parameters:

The properties of a system in which values are assessed to study the system's response are referred to as performance parameters. The characteristics of the step responses are studied in this case to assess the stability of the DC-DC converter, such as percentage of overshoot, rising time, settling time, steady-state error, and peak amplitude.

i. ***Percentage of overshoot (%OS):***

The percentage of maximum response at peak time with respect to steady state response is known as % OS. Percentage of overshoot is expressed mathematically as,

$$\%OS = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100 \quad (5.3)$$

Here, ζ indicating damping ratio that prevents oscillation.

ii. ***Peak time (Tp):***

Peak time represents the time at which the response gets the maximum peak. The mathematical expression for peak time (Tp) can be written as,

$$Tp = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \quad (5.4)$$

iii. ***Settling Time (Ts):***

Settling time is the amount of time required to maintain a steady-state value within ± 2% of the initial value. Ts has the following mathematical expression:

$$Ts = \frac{4}{\zeta\omega_n} \quad (5.5)$$

iv. ***Rise Time (Tr):***

The rise time is the amount of time it takes to reach from 10% to 90% of the response's ultimate value.

v. ***Steady-state Error:***

The variances of the actual value from the desired value when a system reaches steady-state is termed as steady-state error.

5.3 Simulation in MATLAB for zeta converter

A higher-order transfer function of the zeta converter is produced from the state-space modeling under the foundation of the equations created before as a way of simulating the system's stability. MATLAB is then used to lower the model's order. Therefore, equations 5.1 and 5.2 yield us the open-loop and closed-loop transfer functions. Our zeta converter's parameters are shown in Table 5.1. Moreover, The open-loop and closed-loop zeta converter designs in MATLAB Simulink can be seen in Figures 5.1 and 5.2.

Table 5.1 Parameters of Zeta Converter

Parameter	Symbol	Value
Input Voltage	Vin	20V
Output Voltage	Vo	19.92V
Duty Cycle	D	50%
Switching Frequency	Fs	100KHz
Inductor	L1	100μH
	L2	55μH
Resistance	rL1	0.001Ω
	rL2	0.00055Ω
	rC1	0.19Ω
	rC2	0.095Ω
Capacitor	C1	100μF
	C2	200μF
Load Resistance	R	5Ω

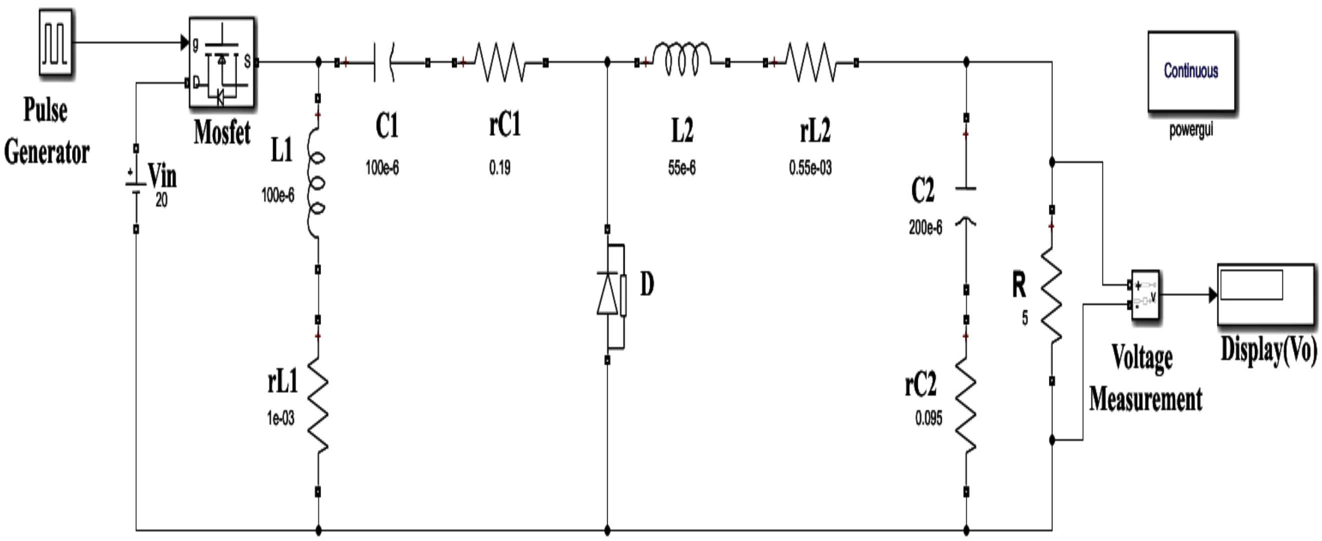


Fig.5.1 MATLAB Simulink model of open-loop zeta Converter

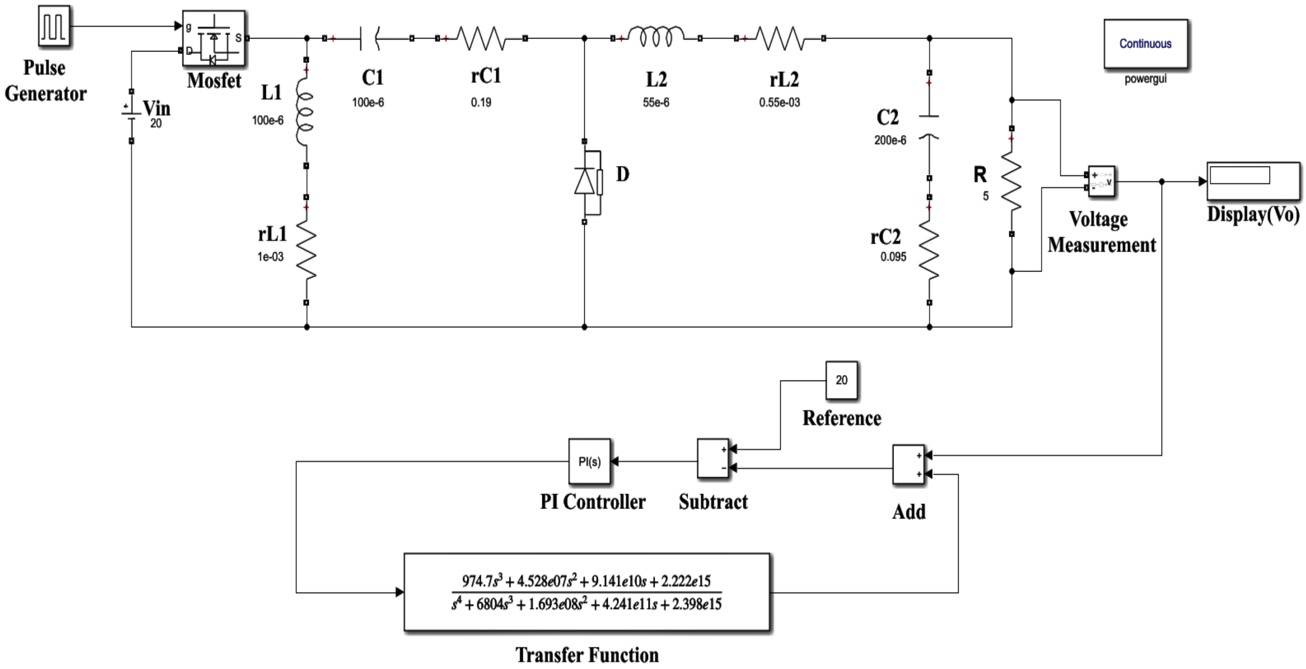


Fig.5.2 MATLAB Simulink model of closed-loop zeta Converter

5.3.1 Conventional PI controller:

The step response of a traditional PI controller is first examined, as shown in Fig. 5.3. Gain values and performance characteristics of a conventional PI controller are presented in Table 5.2.

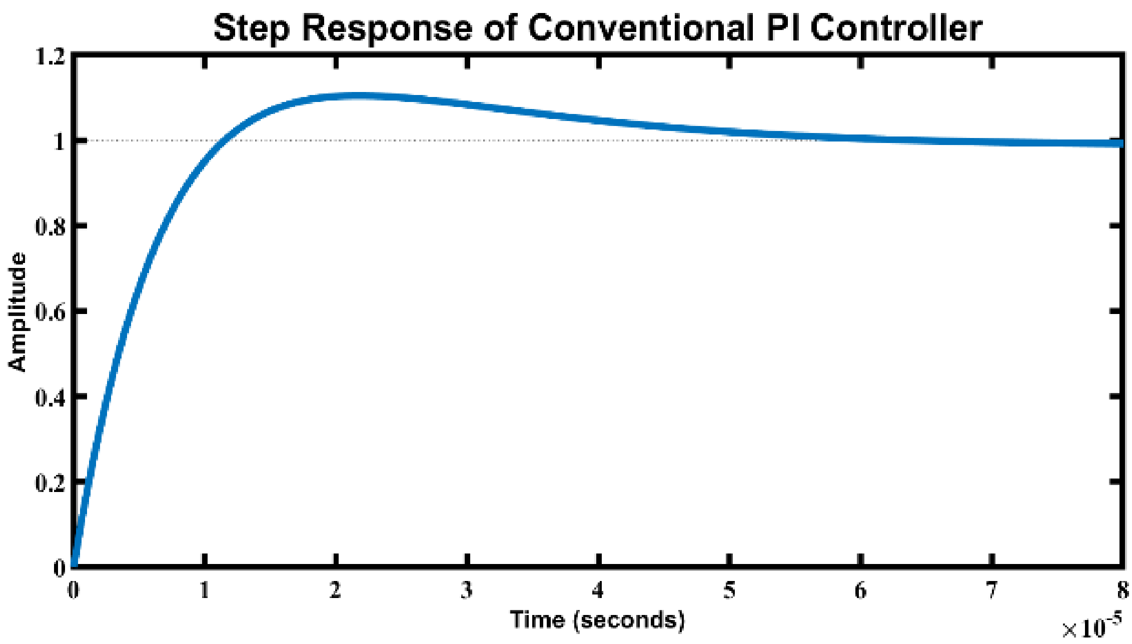


Fig.5.3 Step Response of Conventional PI Controller

Table 5.2 Output of Conventional PI based Zeta Converter

Attributes	Symbols	Values
Gain Values	Kp	191.2786
	Ki	2236.0087
Performance Parameters	%OS	10.4559
	Tr(seconds)	8.2167e-06
	Ts(seconds)	4.9872e-05
	peak Amplitude	1.1046

5.3.2 SIA-PI controller

Swarm Intelligence Algorithm is used to obtain the optimum gain values for the PI controller after applying and determining the performance of the traditional PI controller. ABC,ACOR,PSO,FA and SFLA were among the five algorithms we used. Ultimately, step responses are monitored, and gain values as well as performance parameters for various SIA-PID controllers are recorded.

5.3.2.1 ABC-PI controller

Firstly, ABC is used to find optimized gain values for the PI controller, and the parameters of ABC are listed in Table 5.3. Then for various objective function the step responses are simulated and the comparative analysis of step responses from the four objective function is depicted in the Fig.5.4. Table 5.4 displays the gain and performance characteristics of the ABC-PI controller for our zeta converter, moreover Fig.5.5 and 5.6 present a comparison of performance parameters.

Table 5.3 Parameters of ABC

Acceleration Coefficient Upper Bound	0.02
Abandonment Limit Parameter (Trial Limit)	24
Number of Onlooker Bees	20
Population Size	20
No. Of Iterations	50

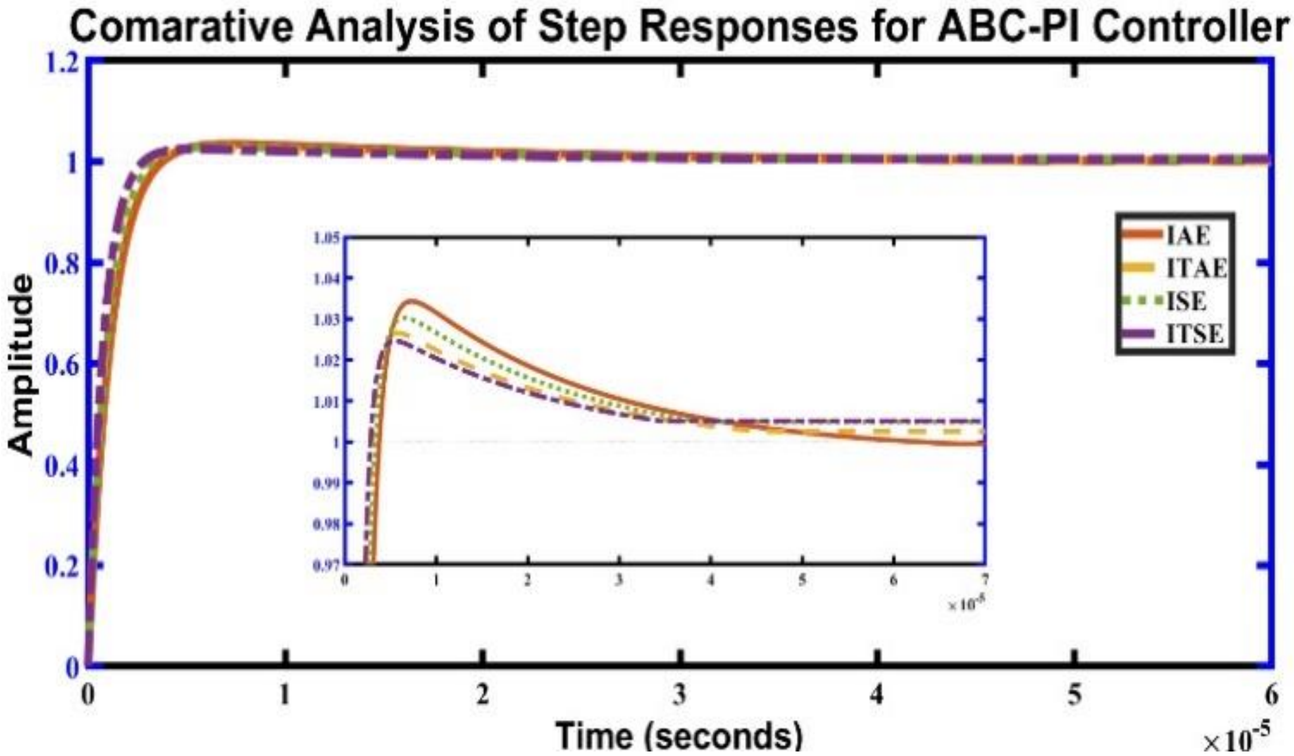


Fig.5.4 Comparative Analysis of Step Responses for ABC-PI controller

Table 5.4 Output of ABC-PI based Zeta Converter

Attributes	**Symbols**	**ABC-PI**			
		IAE	***ITAE***	***ISE***	***ITSE***
Gain Values	**Kp**	909.6358	1236.8	1059.4	1347
	Ki	573.889	883.2359	737.454	1010.5
Performance Parameters	**%OS**	3.4288	2.6524	3.0223	2.4658
	Tr(seconds)	2.2123e-06	1.6707e-06	1.9261e-06	1.5436e-06
	Ts(seconds)	1.8665e-05	1.2165e-05	1.5454e-05	1.0340e-05
	Peak Amplitude	1.0343	1.0265	1.0302	1.0247

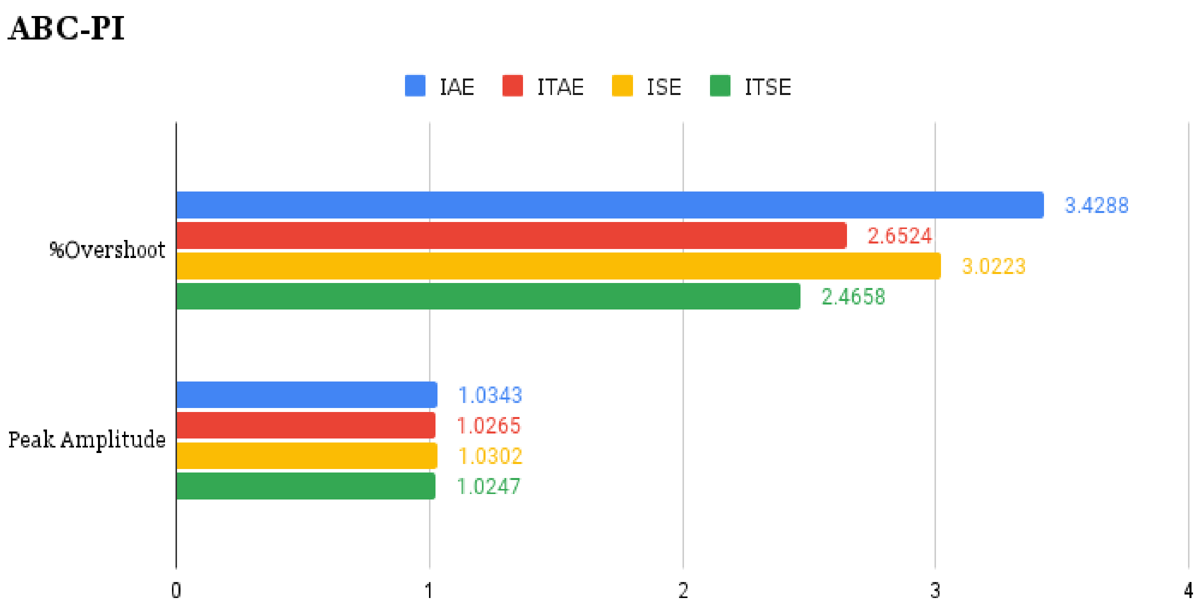


Fig.5.5 Comparative chart in terms of %OS and peak amplitude for ABC-PI controller

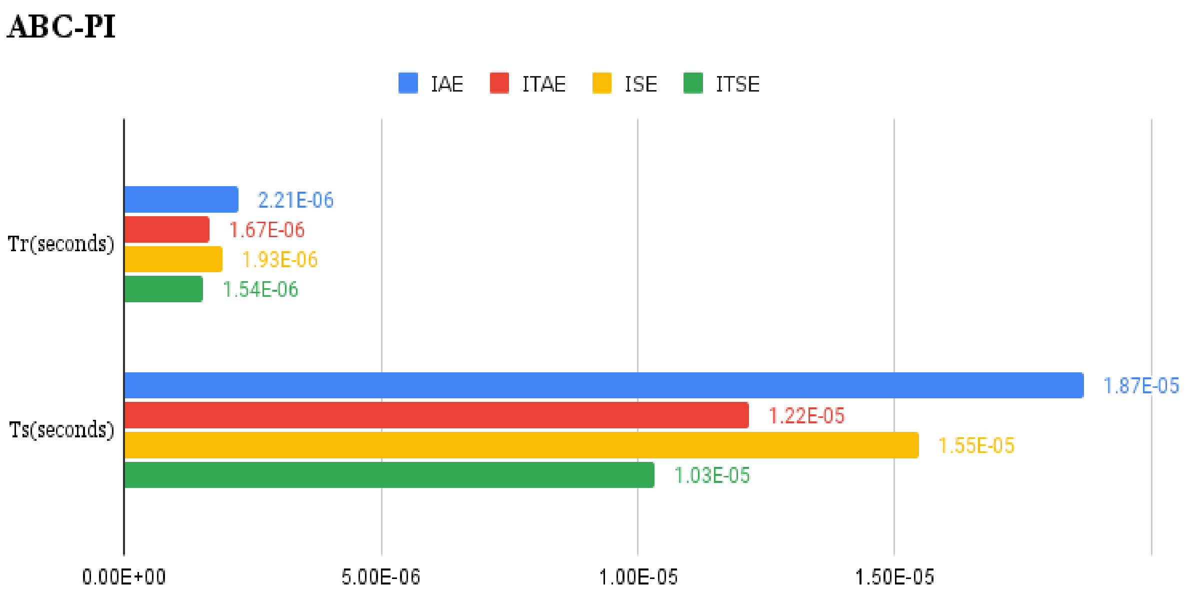


Fig.5.6 Comparative chart in terms of rise time and settling time for ABC-PI controller

5.3.2.2 ACOR-PI controller

Thereafter, ACOR is incorporated to find PI controller optimized gain values, and the parameters of ACOR are provided in Table 5.5. Following that, the step responses for four objective functions are simulated and then all of these step responses are compared in the Fig. 5.7. For a comparison Table 5.6 enumerates the gain values and performance characteristics of step responses for the ACOR-PI controller based zeta converter. In addition, Fig.5.8 and Fig.5.9 both are illustrating comparative charts in terms of performance results.

Table 5.5 Parameters of ACOR

Deviation-Distance Ratio	0.015
Intensification Factor (Selection Pressure)	0.05
Sample Size	10
Population Size	20
No. Of Iterations	50

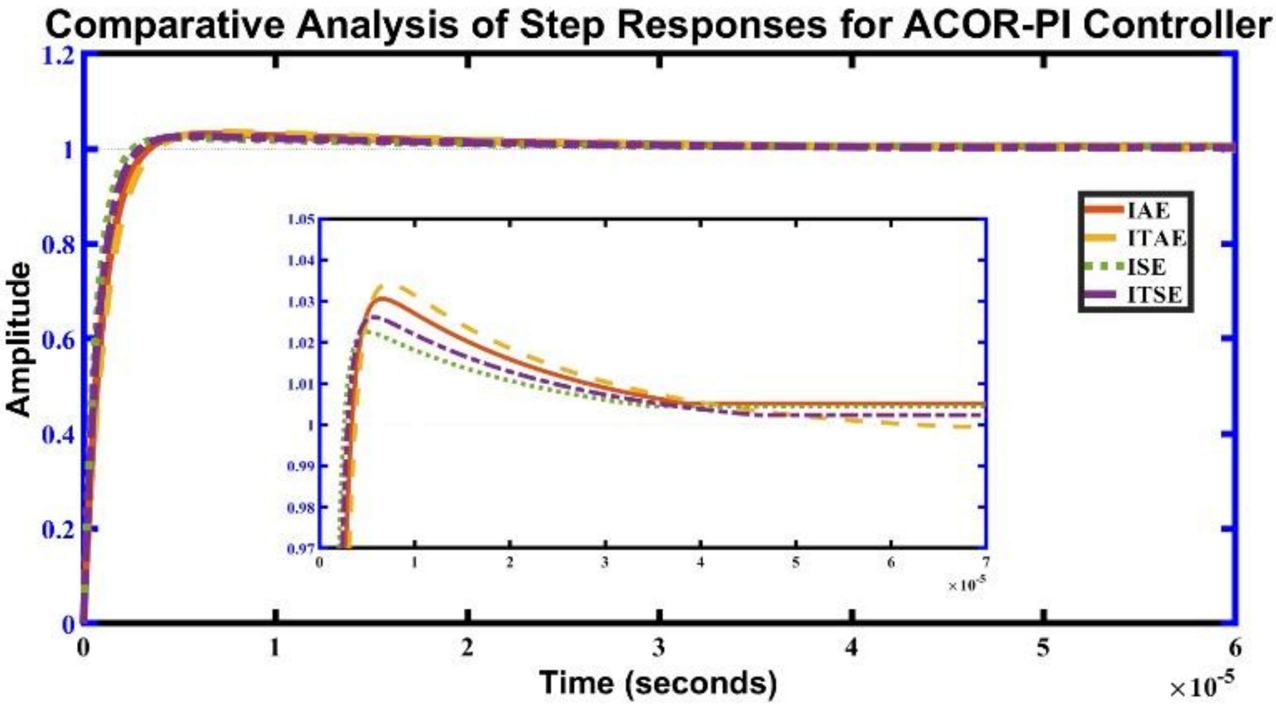


Fig. 5.7 Comparative Analysis of Step Responses for ACOR-PI Controller

Table 5.6 Output of ACOR-PI based Zeta Converter

Attributes	**Symbols**	**ACOR-PI**			
		IAE	***ITAE***	***ISE***	***ITSE***
Gain Values	**Kp**	1045.5	913.1193	1498.3	1261.9
	Ki	1947.8	1989.8	1912	1874.5
Performance Parameters	**%OS**	3.0559	3.4182	2.2495	2.6074
	Tr(seconds)	1.9495e-06	2.2046e-06	1.3979e-06	1.6399e-06

	Ts(seconds)	1.5734e-05	1.8585e-05	8.0472e-06	1.1736e-05
	Peak Amplitude	1.0306	1.0342	1.0225	1.0261

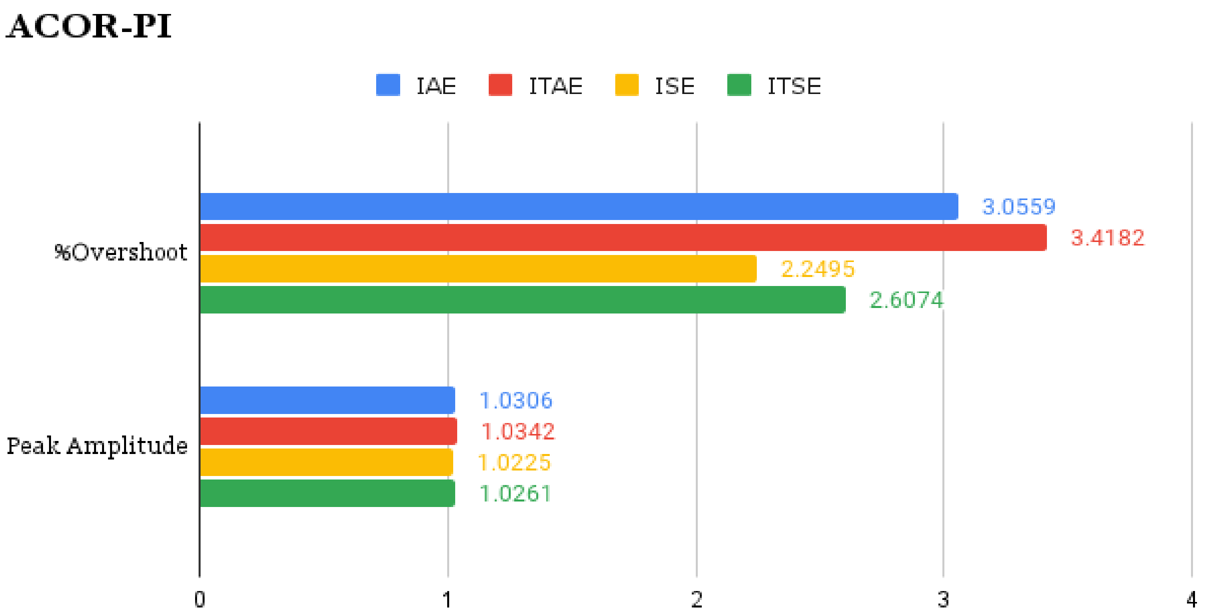


Fig.5.8 Comparative chart in terms of %OS and peak amplitude for ACOR-PI controller

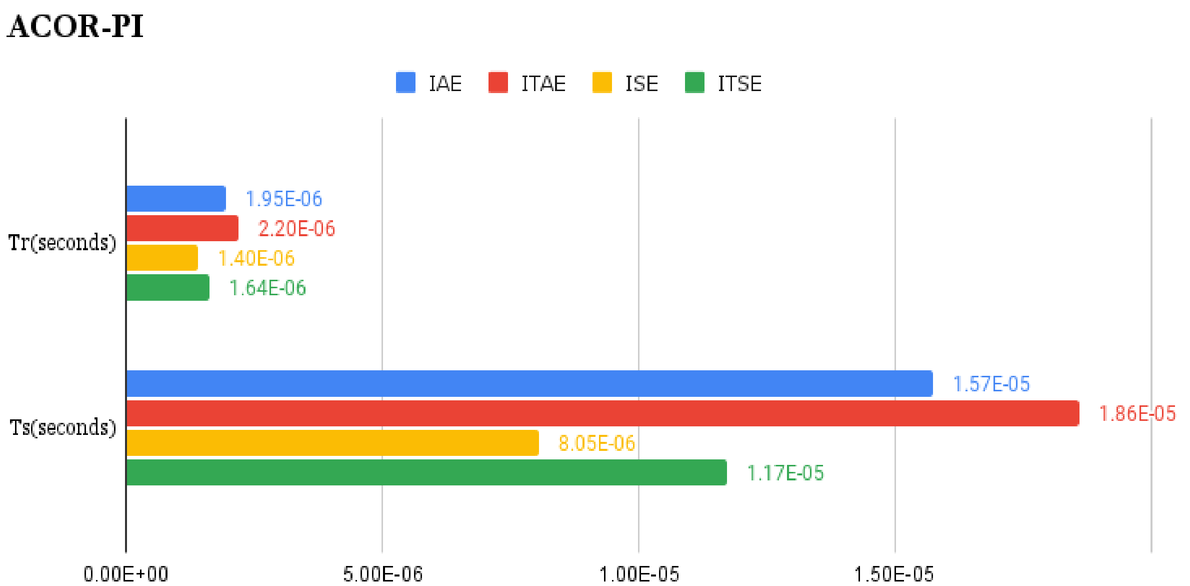


Fig.5.9 Comparative chart in terms of rise time and settling time for ACOR-PI controller

5.3.2.3 PSO-PI controller

Likewise, PSO has been used to adjust the gain parameters of a PI controller and the parameters considered for PSO are listed in Table 5.7. Following that, step responses of every objective function out of four is simulated and then a comparison of all of these step responses are displayed in the Fig. 5.10. Table 5.8 enlisted the gain and performance parameters of the PSO-PI controller for the zeta converter. Subsequently, a comparison based on performance indicators is prepared in the Fig 5.11 and 5.12.

Table 5.7 Parameters of PSO

Personal Acceleration Coefficient	0.002
Social Acceleration Coefficient	0.004
Damping Ratio of Inertia Coefficient	0.099
Inertia Coefficient	1.5
Population Size	20
No. Of Iterations	50

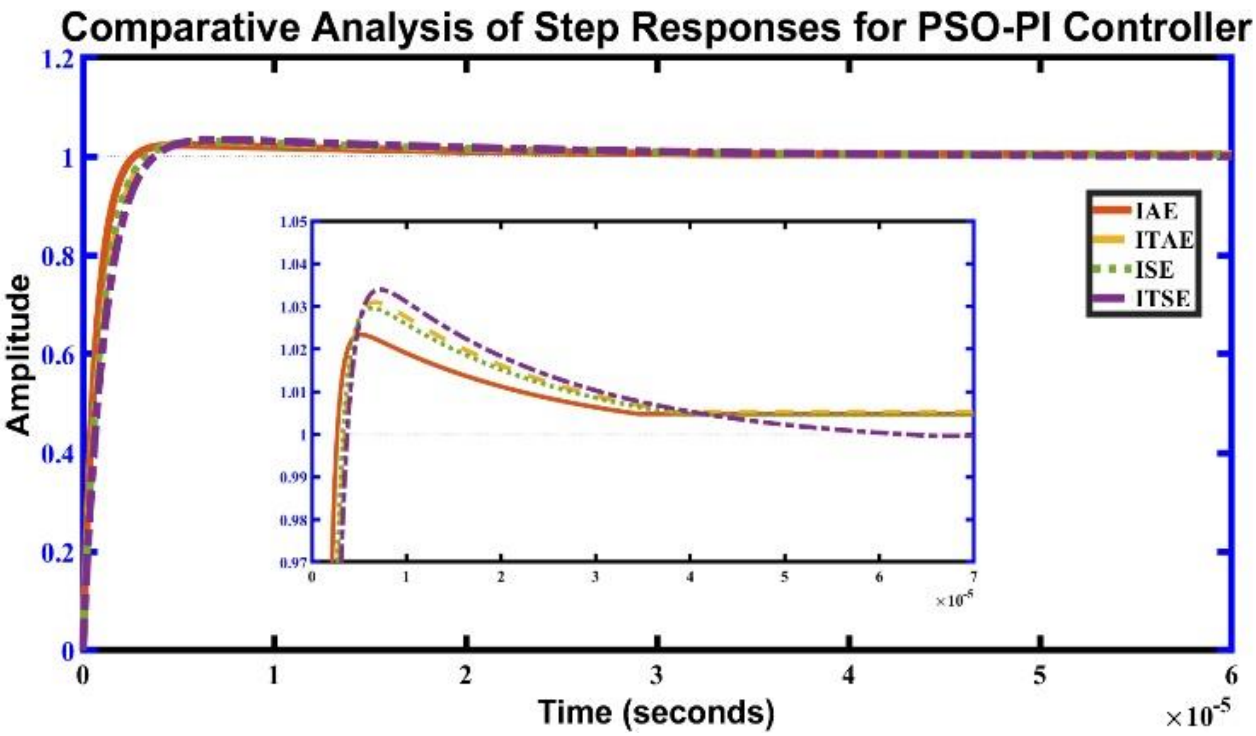


Fig. 5.10 Comparative Analysis of Step Responses for PSO-PI Controller

Table 5.8 Output of PSO-PI based Zeta Converter

Attributes	**Symbols**	**PSO-PI**			
		IAE	***ITAE***	***ISE***	***ITSE***
Gain Values	**Kp**	1433.9	1029.3	1087.6	922.17
	Ki	1479	823.8417	1354.1	1057.6
Performance Parameters	**%OS**	2.3367	3.0959	2.9566	3.3905
	Tr(seconds)	1.4564e-06	1.9775e-06	1.8803e-06	2.1850e-06
	Ts(seconds)	8.9968e-06	1.6064e-05	1.4898e-05	1.8378e-05
	Peak Amplitude	1.0234	1.0310	1.0296	1.0339

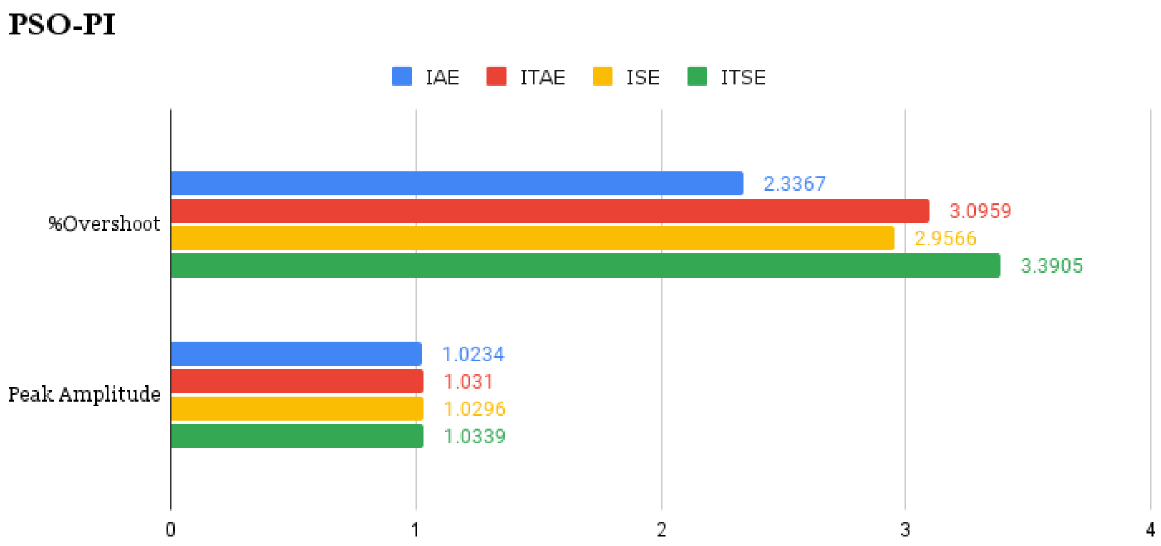


Fig.5.11 Comparative chart in terms of %OS and peak amplitude for PSO-PI controller

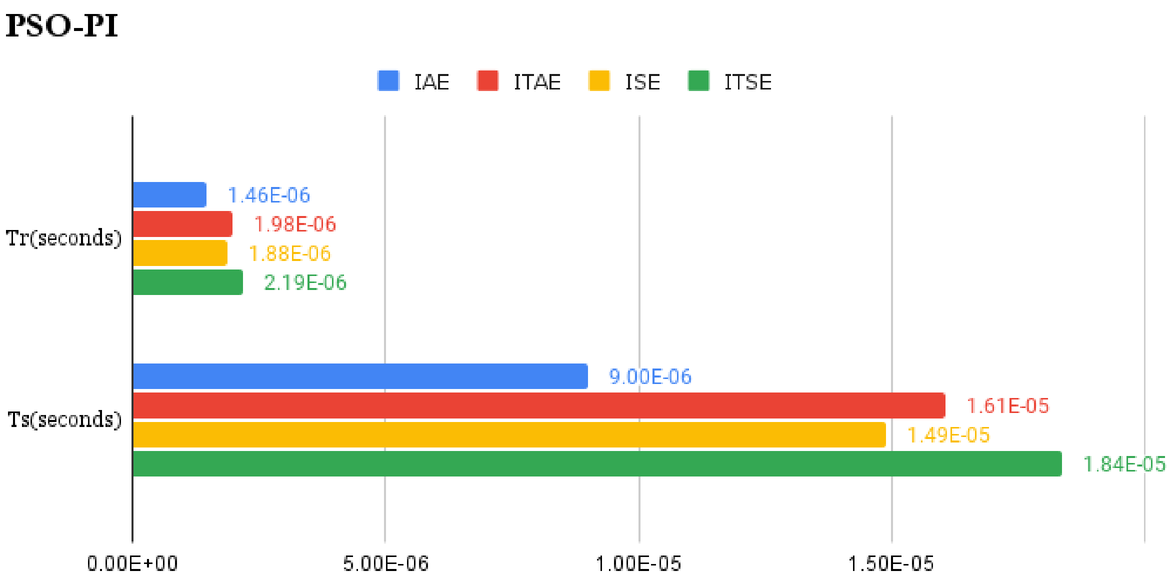


Fig.5.12 Comparative chart in terms of rise time and settling time for PSO-PI controller

5.3.2.4 FA-PI controller

Then Firefly algorithm is implemented with our zeta converter for having optimized gain values. Table 5.9 records the parameters that were used in the algorithm. Step responses are then generated for various objective functions, as before, and a comparative assessment of all of these step responses is shown in Fig. 5.13. Table 5.10 contains the gain and performance attributes of the FA-PI controller for our zeta converter, whereas Fig. 5.14 and 5.15 both are showing a comparative illustration based on the outcomes discovered.

Table 5.9 Parameters of FA

Mutation Coefficient Damping Ratio	1.8
Mutation Coefficient	0.051293
Attraction Coefficient Base Value	0.08293
Light Absorption Coefficient	0.06439
Population Size	20
No. Of Iterations	50

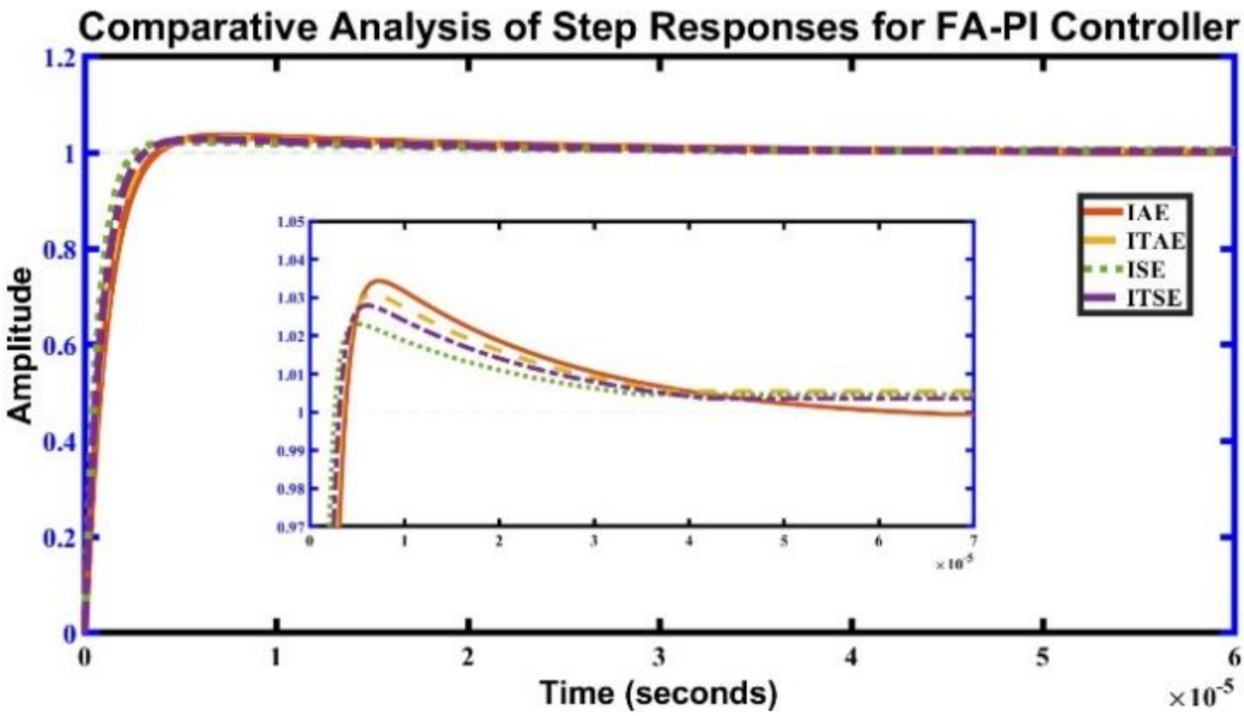


Fig. 5.13 Comparative Analysis of Step Responses for FA-PI Controller

Table 5.10 Output of FA-PI based Zeta Converter

Attributes	**Symbols**	**FA-PI**			
		IAE	***ITAE***	***ISE***	***ITSE***
Gain Values	**Kp**	907.5887	1032.6	1456.3	1163.8
	Ki	515.012	1577.7	1833.6	1728.9
Performance Parameters	**%OS**	3.4351	3.0877	2.3056	2.7928
	Tr(seconds)	2.2168e-06	1.9717e-06	1.4355e-06	1.7669e-06
	Ts(seconds)	1.8712e-05	1.5996e-05	8.6626e-06	1.3461e-05
	Peak Amplitude	1.0344	1.0309	1.0231	1.0279

FA-PI

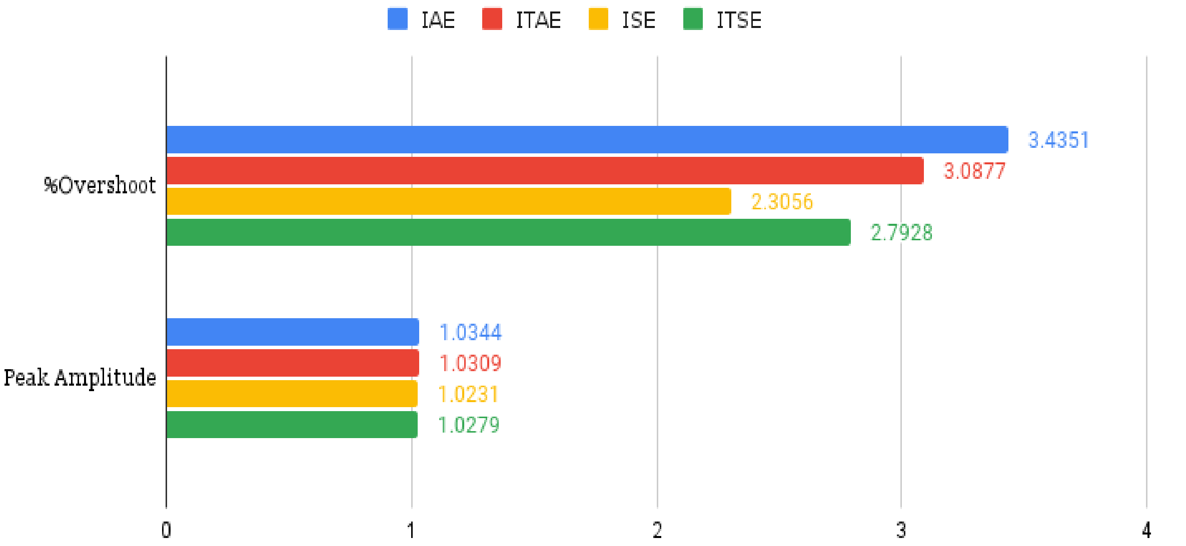


Fig.5.14 Comparative chart in terms of %OS and peak amplitude for FA-PI controller

FA-PI

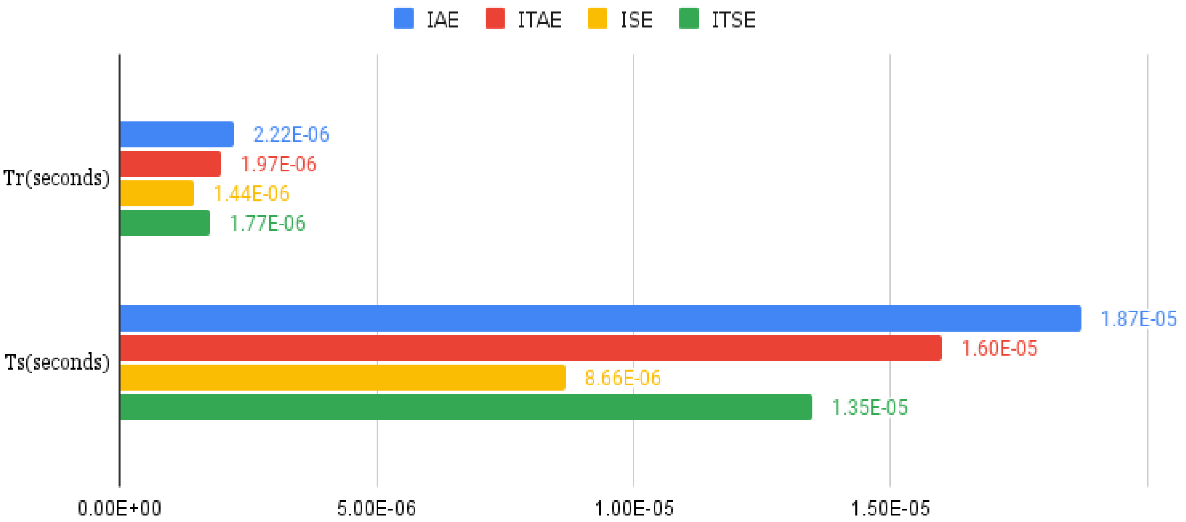


Fig.5.15 Comparative chart in terms of rise time and settling time for FA-PI controller

5.3.2.5 SFLA-PI controller

Finally, SFLA is used to optimize the gain factors of PI controller for the zeta converter with its parameters provided in Table 5.11, as well as step responses for four objective functions are simulated. The four step responses are shown in the Fig 5.16 in a comparable method. Table 5.12 summarizes the gain and performance factors of the SFLA-PI controller. Additionally, Figures 5.17 and 5.18 show a comparison of performance measures.

Table 5.11 Parameters of SFLA

Number of iterations within each memeplex	2
Step Size	1.1
Number of Offspring	4
Number of Memeplexes	2
Memeplex Size	10
No. Of Iterations	50

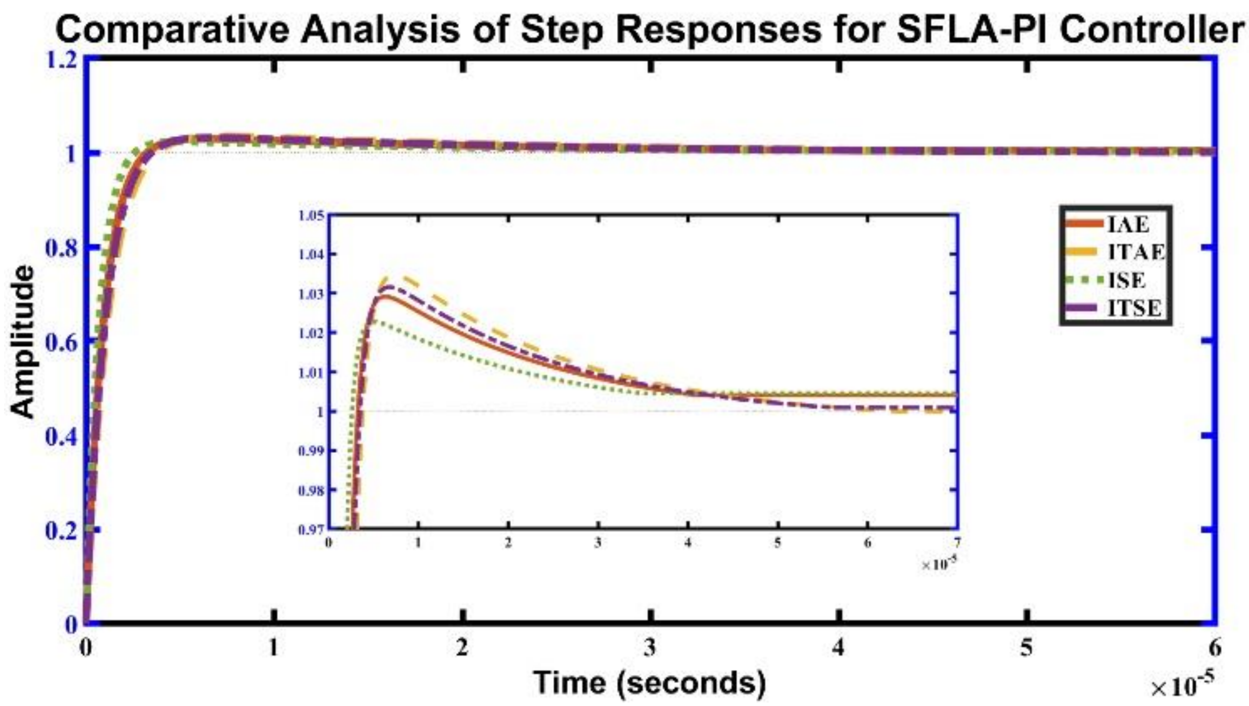


Fig. 5.16 Comparative Analysis of Step Responses for FA-PI Controller

Table 5.12 Output of FA-PI based Zeta Converter

Attributes	Symbols	SFLA-PI			
		IAE	***ITAE***	***ISE***	***ITSE***
Gain Values	**Kp**	1107.1	901.8246	1478.9	1007.1
	Ki	1950.1	1874.2	1715.8	1948.7
Performance Parameters	**%OS**	2.9129	3.4532	2.2750	3.1526
	Tr(seconds)	1.8498e-06	2.2296e-06	1.4150e-06	2.0171e-06
	Ts(seconds)	1.4522e-05	1.8846e-05	8.3296e-06	1.6524e-05
	Peak Amplitude	1.0291	1.0345	1.0228	1.0315

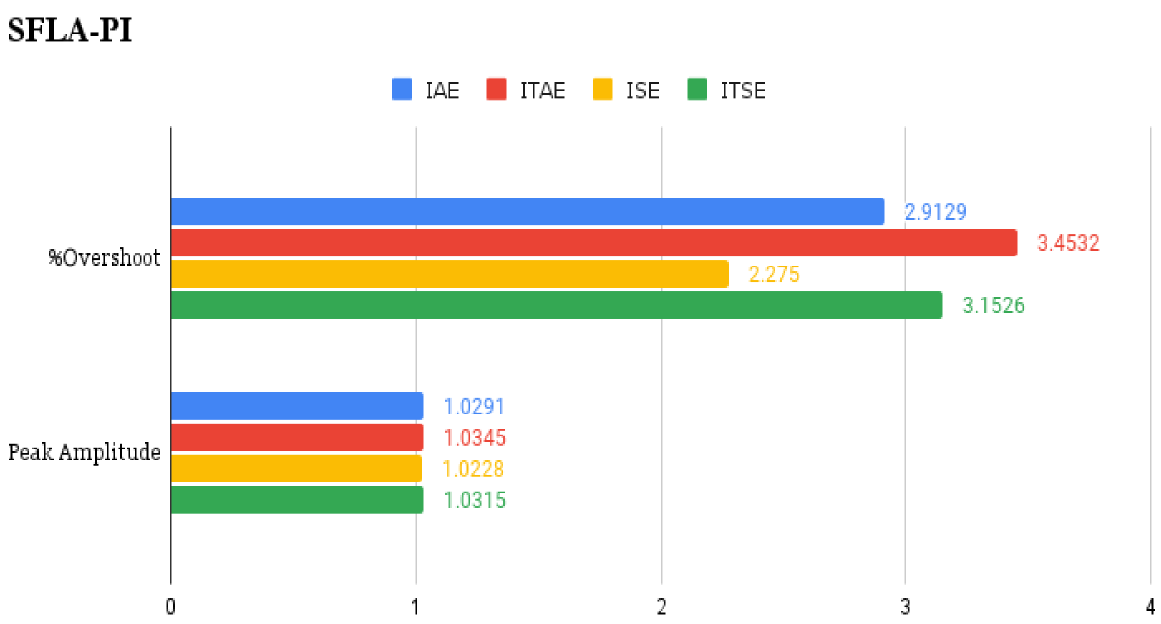


Fig.5.17 Comparative chart in terms of %OS and peak amplitude for SFLA-PI controller

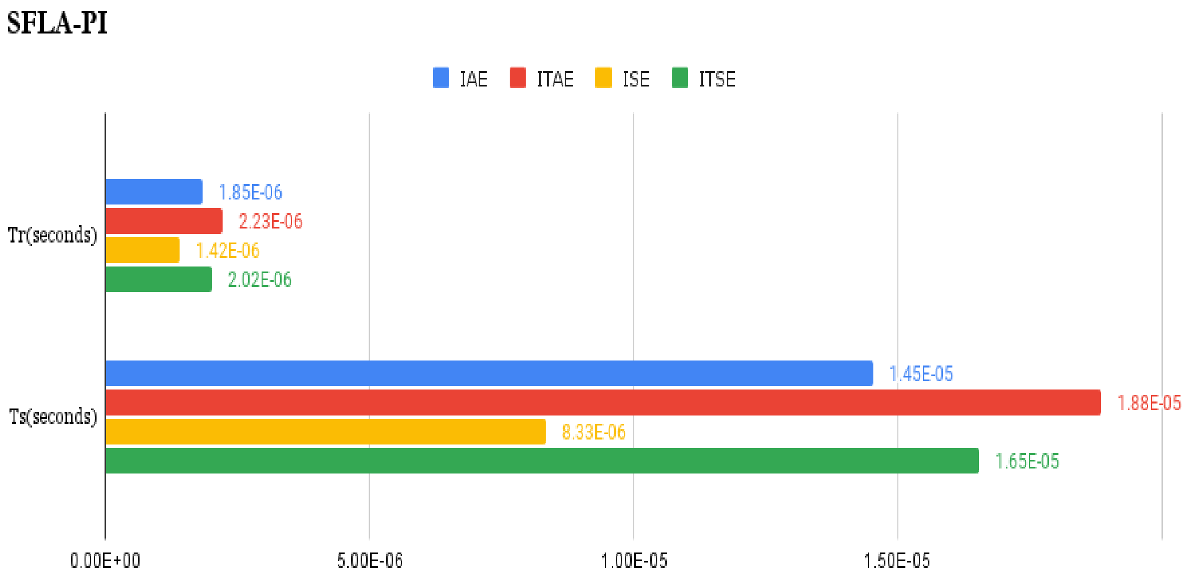


Fig.5.18 Comparative chart in terms of rise time and settling time for SFLA-PI controller

5.3.3 Comparative Analysis

Ultimately, Table 5.13 presents a comparison of performance metrics of the best performing SIA-PI controller depending on the evidence from Tables 5.2, 5.4, 5.6, 5.8, 5.10 and 5.12. Furthermore, an overall investigation on step responses is performed to compare all five SIA based PI controller and conventional PI based on the values of performance attributes, shown in the Fig.5.19. This illustrates the superiority of SIA-based PI controllers in terms of stability over conventional PI controllers. Eventually, in the Fig. 5.20 and 5.21, an aggregate assessment of performance metrics among the SIA-PI controller with their best performing objective function is shown, from which it is apparent that the ACOR-ISE PI controller outperforms the other four SIA-PI controllers in terms of %overshoot, rise time and settling time. However, it is also worth noting that the SFLA-ISE PI controller performs close to the ACOR-ISE PI controller.

Table 5.13 Comparative Analysis on the Output of PI Controllers

Attributes	**PI Controllers**					
	Conventional	***ABC-PI***	***ACOR-PI***	***PSO-PI***	***FA-PI***	***SFLA-PI***
Best Performing Fitness Function	N/A	ITSE	ISE	IAE	ISE	ISE
Percentage of Overshoot(%OS)	10.4559	2.4658	2.2495	2.3367	2.3056	2.2750
Rise Time, Tr(seconds)	8.2167e-06	1.5436e-06	1.3979e-06	1.4564e-06	1.4355e-06	1.4150e-06
Settling Time,Ts(seconds)	4.9872e-05	1.0340e-05	8.0472e-06	8.9968e-06	8.6626e-06	8.3296e-06
Peak Amplitude	1.1046	1.0247	1.0225	1.0234	1.0231	1.0228

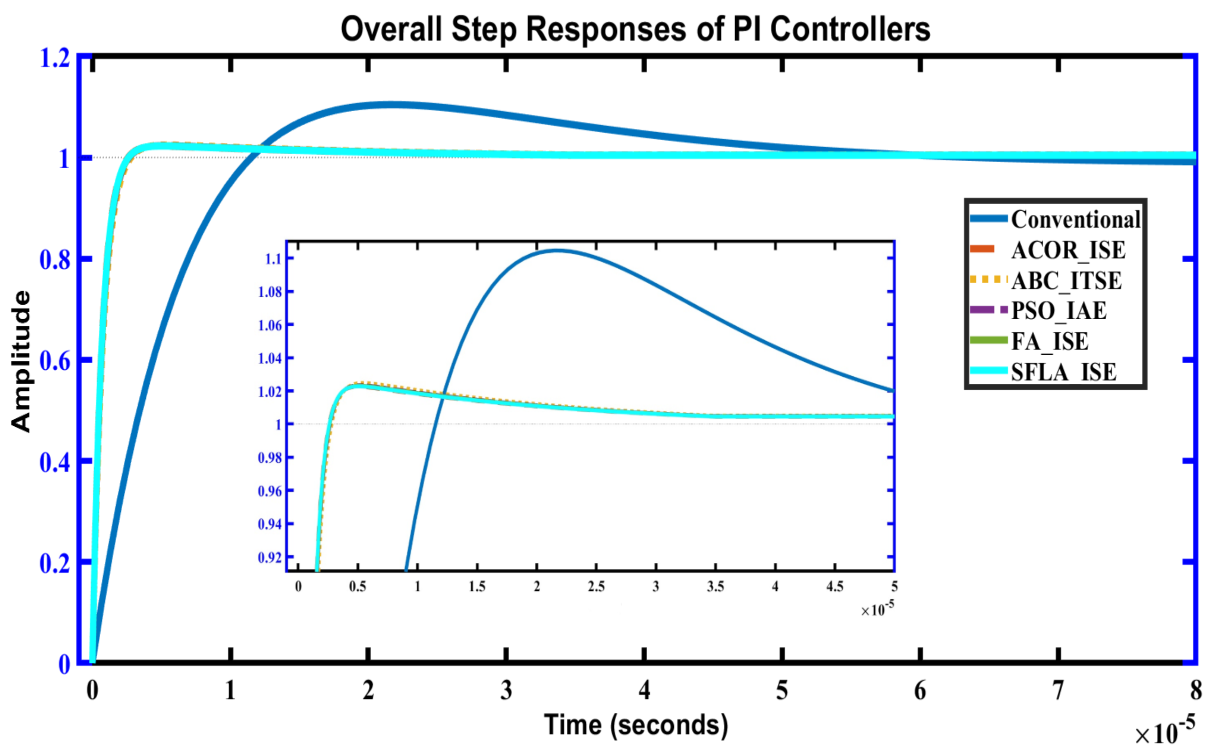


Fig. 5.19 Overall Comparative Step Responses of PI Controllers

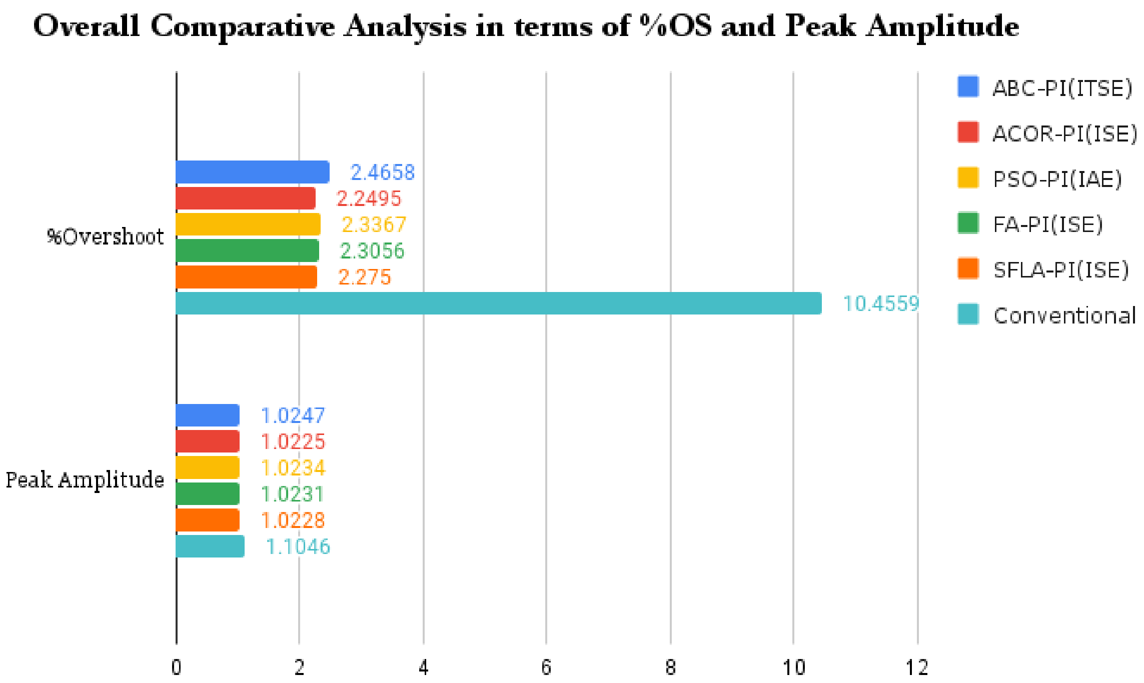


Fig.5.20 Overall Comparative chart in terms of %OS and peak amplitude

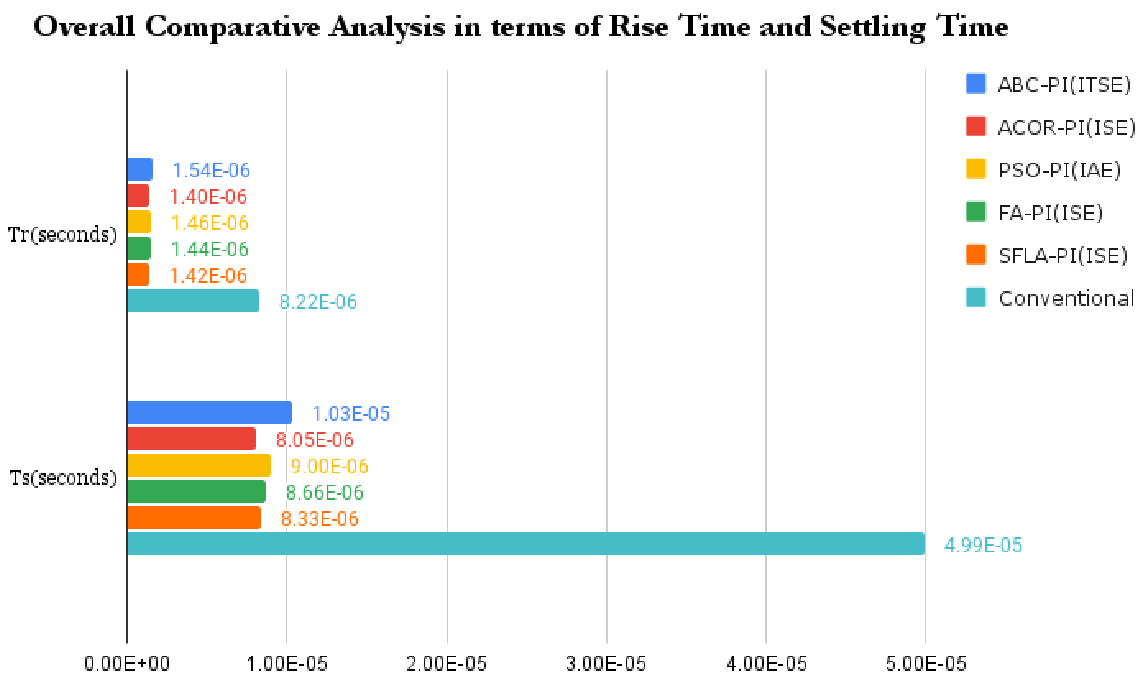


Fig.5.21 Overall Comparative chart in terms of Rise time and Settling time

5.4 Eigen Value Analysis

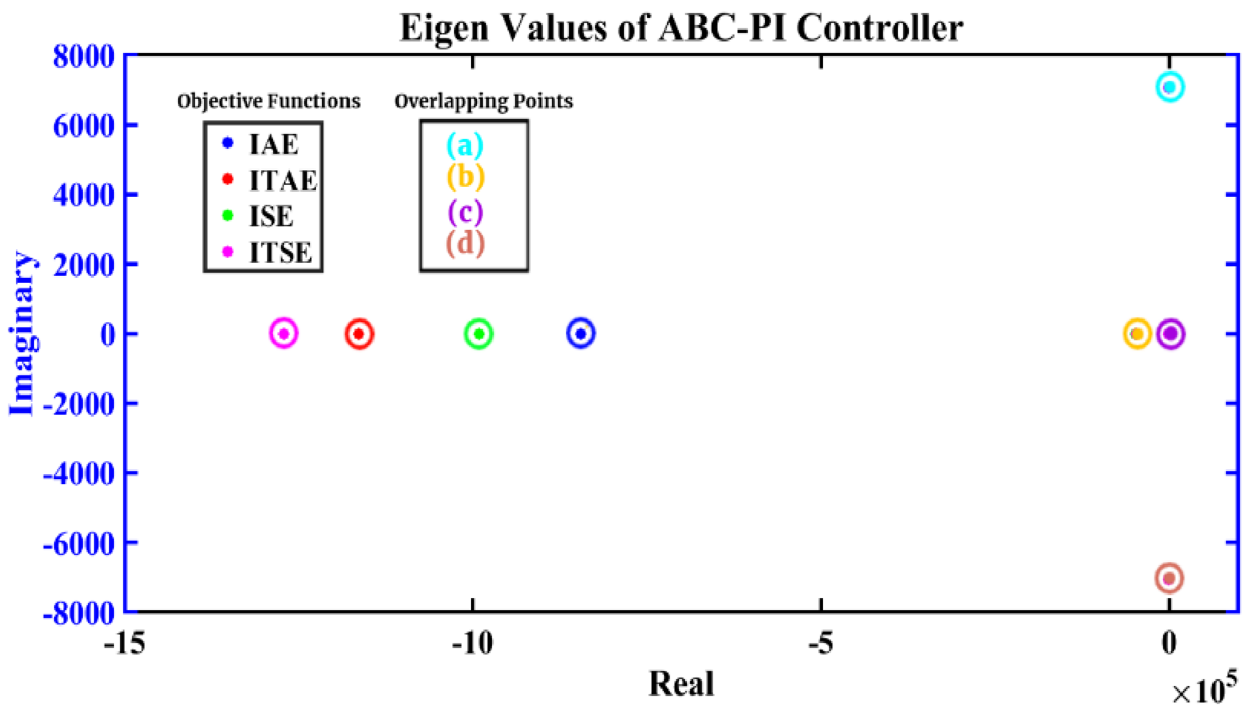
To determine the stability of a swarm-based PI controller incorporated with the zeta converter, an eigenvalue analysis has been performed. Using eigenvalue analysis, the proposed algorithms' ability to attain system stability is determined. The analysis is carried out with the use of four objective functions: IAE, ITAE, ISE, and ITSE. The real part of the eigenvalue holds details on the stability of the system. When the real element of the eigenvalue is on the left side of the s plane, the system is considered to be stable[93]. The eigenvalues for each objective function of five swarm intelligence algorithms are shown in Table 5.14. As shown in the table, Each objective function comprises four eigen values, two of which are of the complex form.

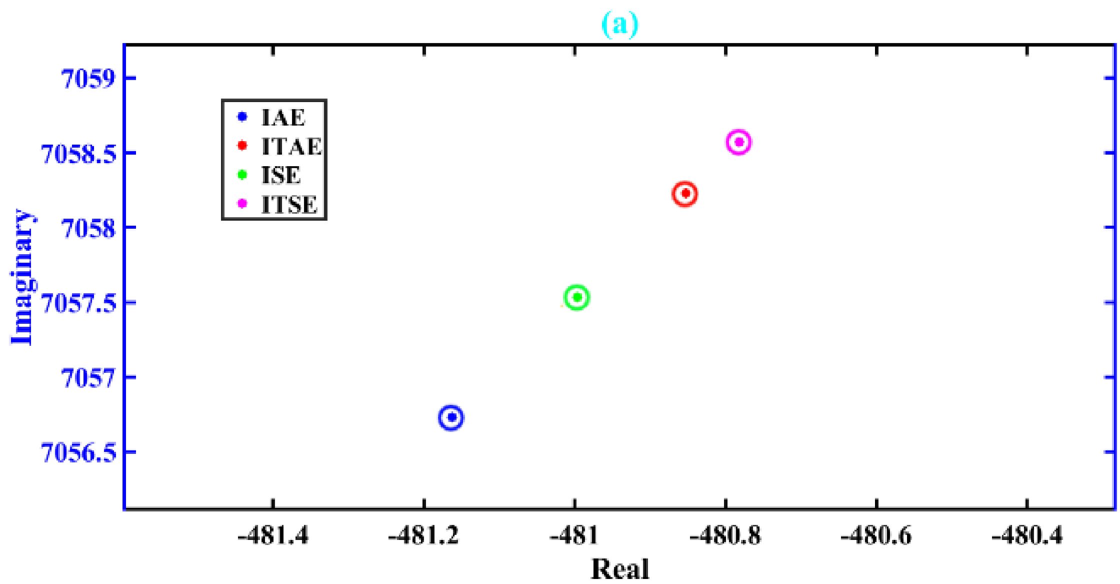
Table 5.14 Eigen Values of SIA-PI controllers

Algorithm	**Objective Function**	**Eigen Value Of The System**			
PSO	IAE	1.0e+06 *(-1.3565 + 0.0000i)	1.0e+06 * (-0.0470 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITAE	1.0e+05 * (-9.6151 + 0.0000i)	1.0e+05 * (-0.4759 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ISE	1.0e+06 * (-1.0185 + 0.0000i)	1.0e+06 * (-0.0475 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITSE	1.0e+05 * (-8.5682 + 0.0000i)	1.0e+05 * (-0.4786 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
ABC	IAE	1.0e+05 *(-8.4457 + 0.0000i)	1.0e+05 *(-0.4789 + 0.0000i)	1.0e+05 *(-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ITAE	1.0e+06 *(-1.1641 + 0.0000i)	1.0e+06 * (-0.0472 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ISE	1.0e+05 *(-9.9092 + 0.0000i)	1.0e+05 * (-0.4752 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ITSE	1.0e+06 *(-1.2717 + 0.0000i)	1.0e+06 * (-0.0471 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
ACOR	IAE	1.0e+05 *(-9.7734 + 0.0000i)	1.0e+05 * (-0.4755 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ITAE	1.0e+05 *(-8.4797 + 0.0000i)	1.0e+05 * (-0.4788 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ISE	1.0e+06 *(-1.4193 + 0.0000i)	1.0e+06 *(-0.0469 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITSE	1.0e+06 *(-1.1886 + 0.0000i)	1.0e+06 * (-0.0472 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
FA	IAE	1.0e+05 *(-8.4257 + 0.0000i)	1.0e+05 * (-0.4790 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ITAE	1.0e+05 *(-9.6474 + 0.0000i)	1.0e+05 * (-0.4758 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ISE	1.0e+06 *(-1.3784 + 0.0000i)	1.0e+06 * (-0.0469 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITSE	1.0e+06 *(-1.0929 + 0.0000i)	1.0e+06 * (-0.0473 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
SFLA	IAE	1.0e+06 *(-1.0375 + 0.0000i)	1.0e+06 * (-0.0474 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITAE	1.0e+05 *(-8.3693 + 0.0000i)	1.0e+05 * (-0.4792 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)
	ISE	1.0e+06 *(-1.4004 + 0.0000i)	1.0e+06 * (-0.0469 + 0.0000i)	1.0e+06 * (-0.0005 + 0.0071i)	1.0e+06 * (-0.0005 - 0.0071i)
	ITSE	1.0e+05 *(-9.3982 + 0.0000i)	1.0e+05 * (-0.4764 + 0.0000i)	1.0e+05 * (-0.0048 + 0.0706i)	1.0e+05 * (-0.0048 - 0.0706i)

Finally, eigen values for each SIA-PI controller are presented on each figure to provide a visual representation of the controller's stability while implementing on our zeta converter. In the figures below, the eigenvalues of the five swarm-based PI controllers are shown (Fig. 5.22 - 26). As observed by the figures, The real component of the eigenvalues is on the left side of the s-plane, indicating that the system is stable. Due to the limited range of eigenvalues in the figures, there is considerable overlapping of points. As a consequence, each algorithm's figure is supplemented by four extra figures that more clearly depict overlapping locations.

5.4.1 ABC-PI controller





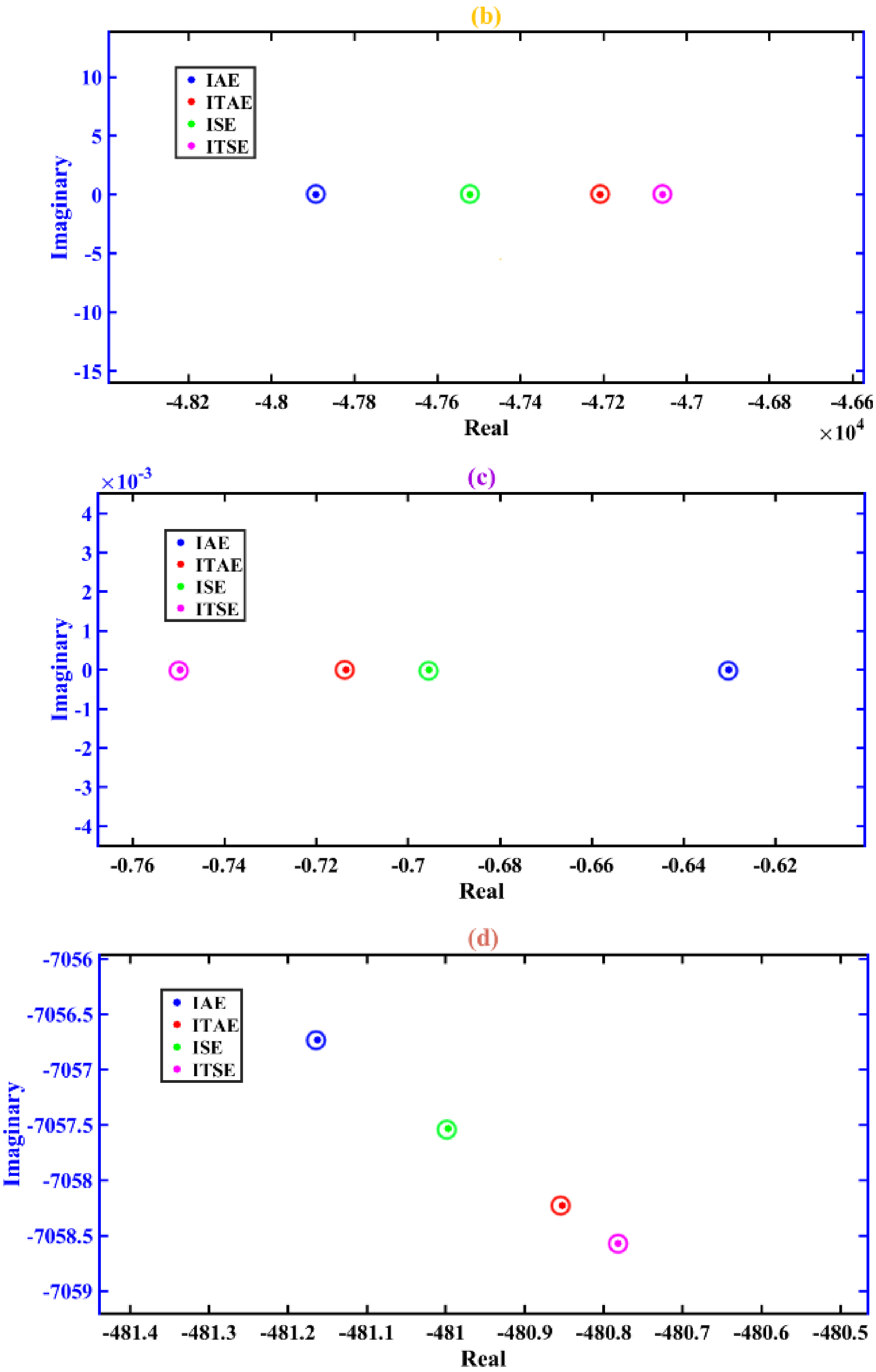
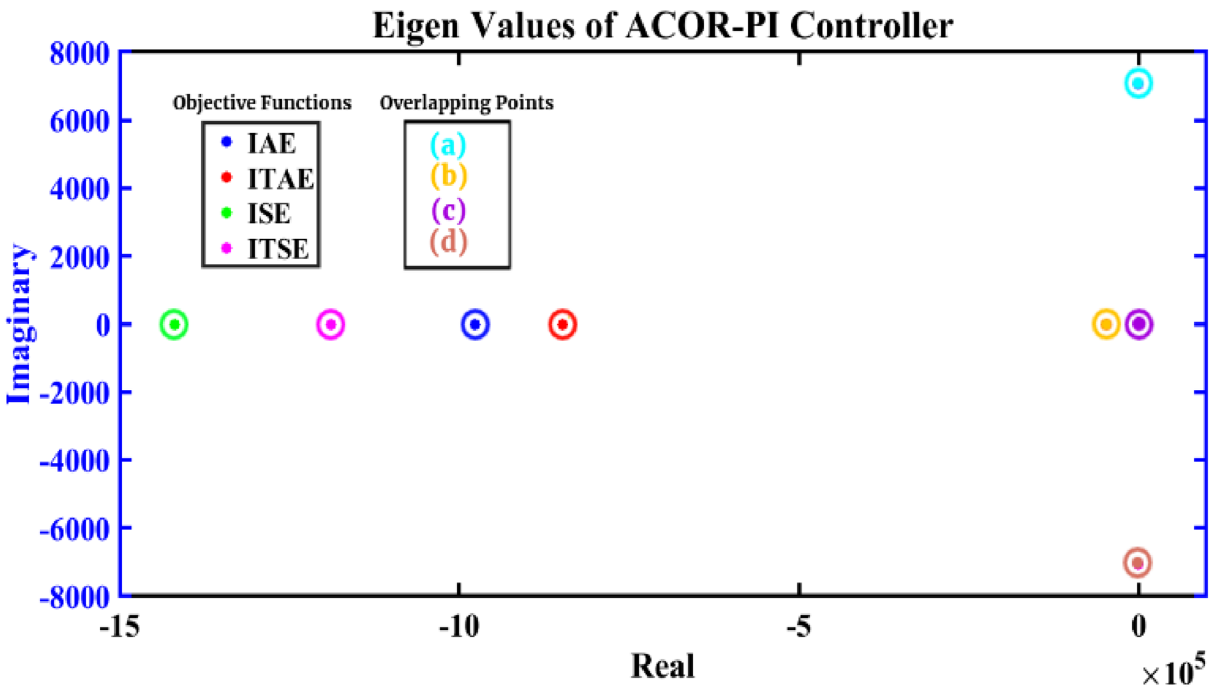
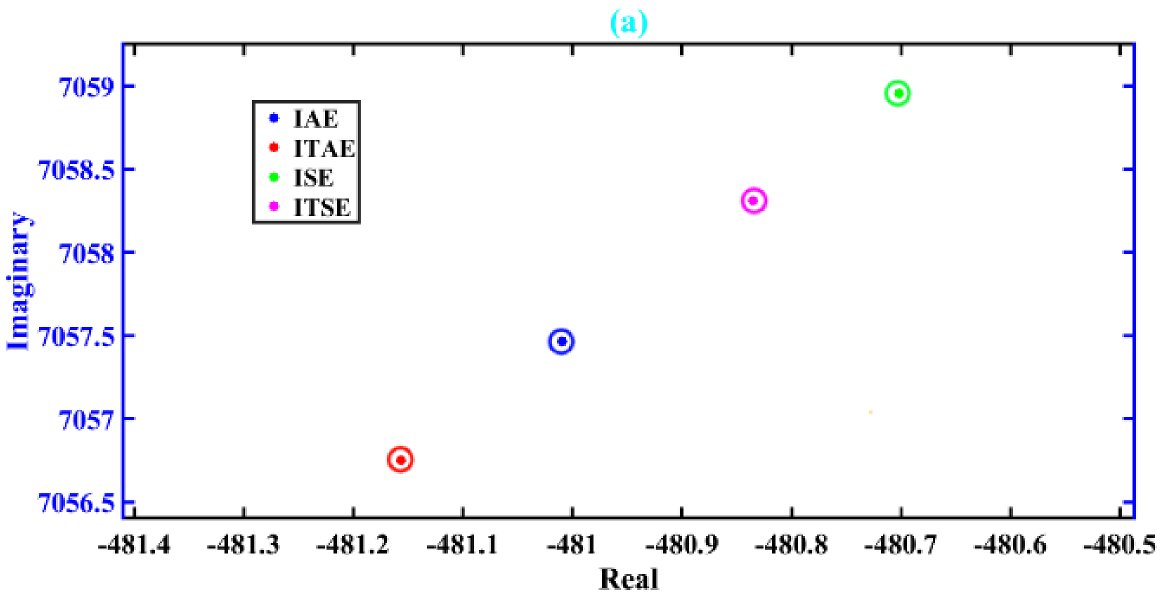


Fig.5.22 Eigen Values of ABC-PI Controller

5.4.2 ACOR-PI controller





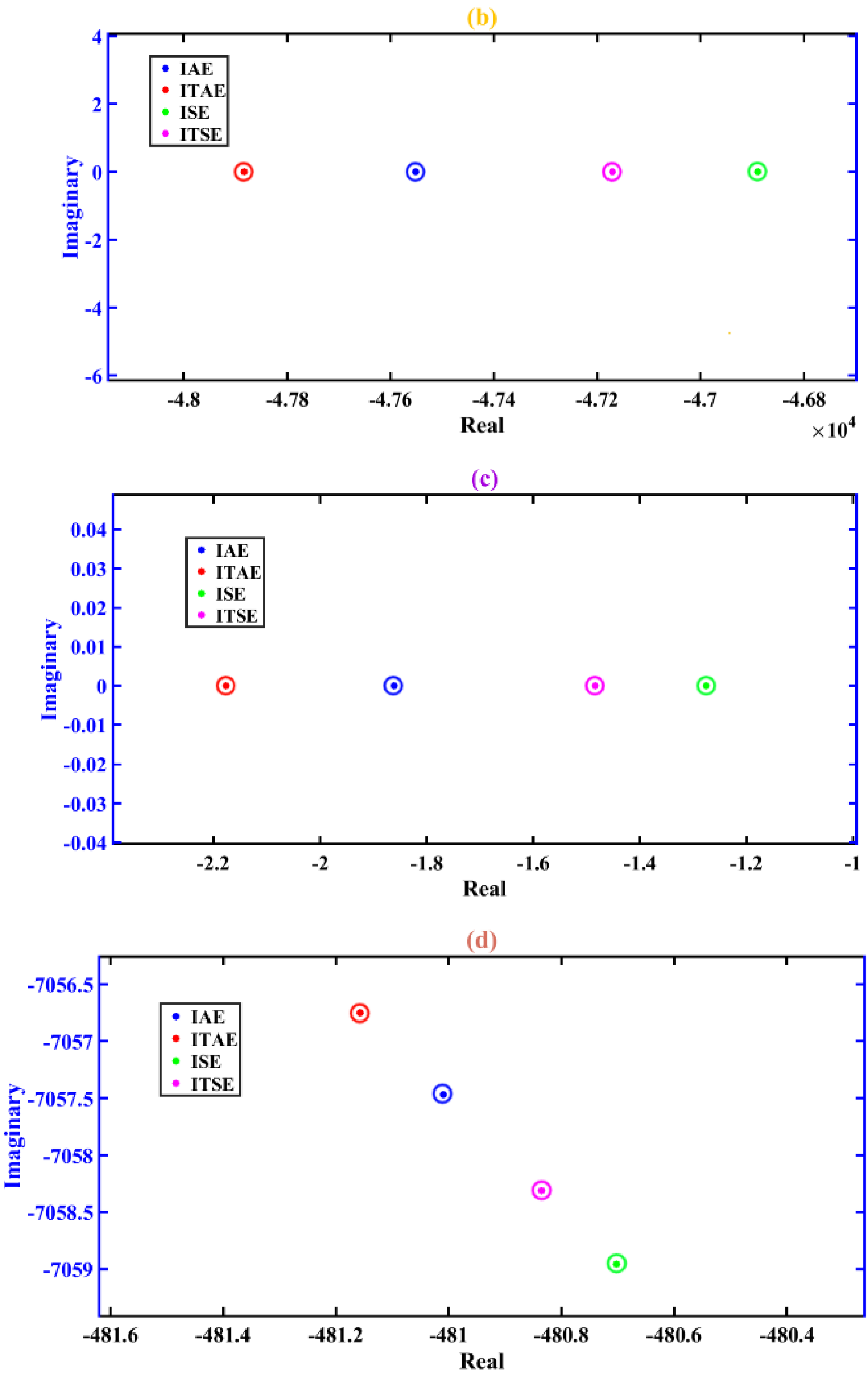
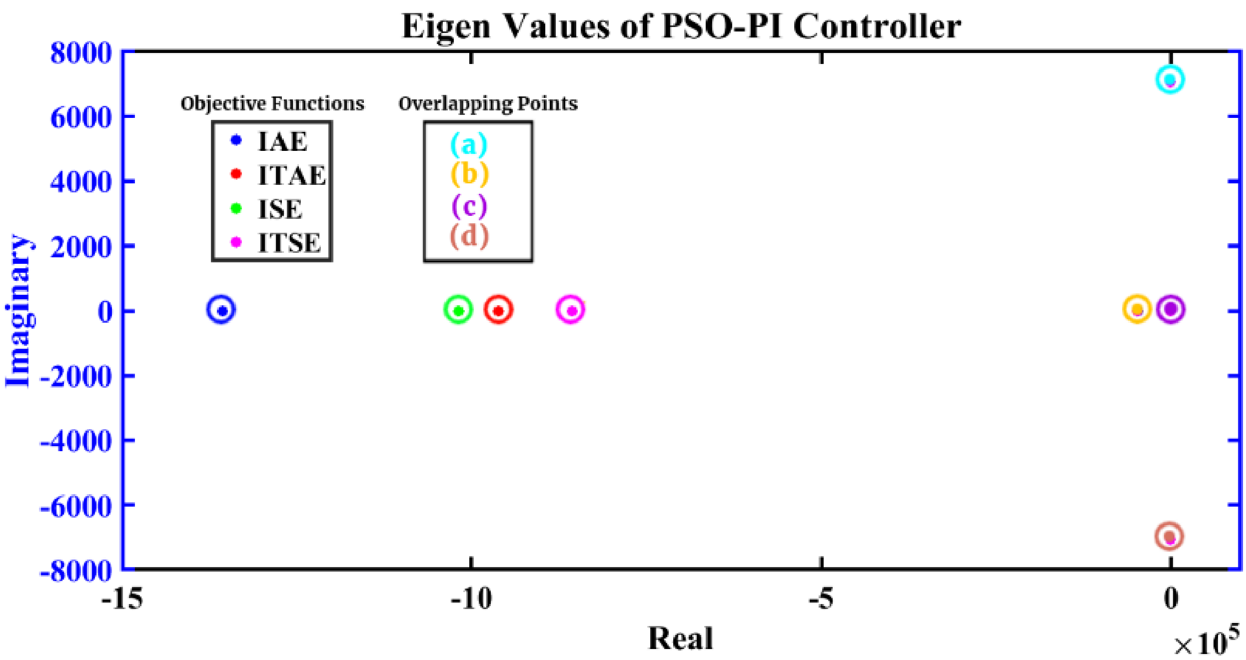
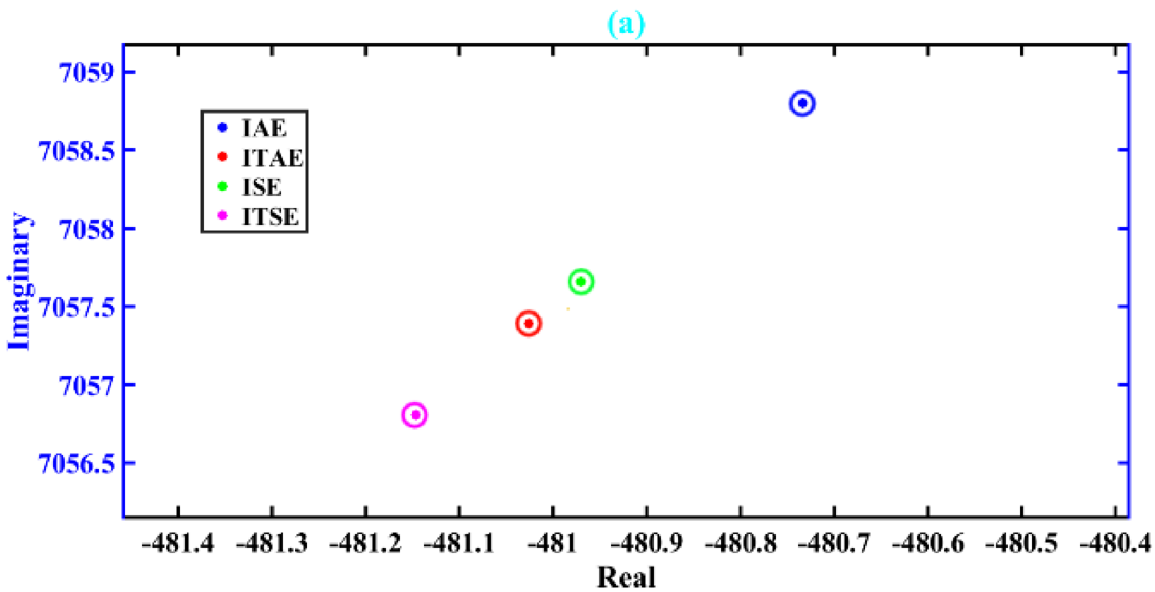


Fig.5.23 Eigen Values of ACOR-PI Controller

5.4.3 PSO-PI controller





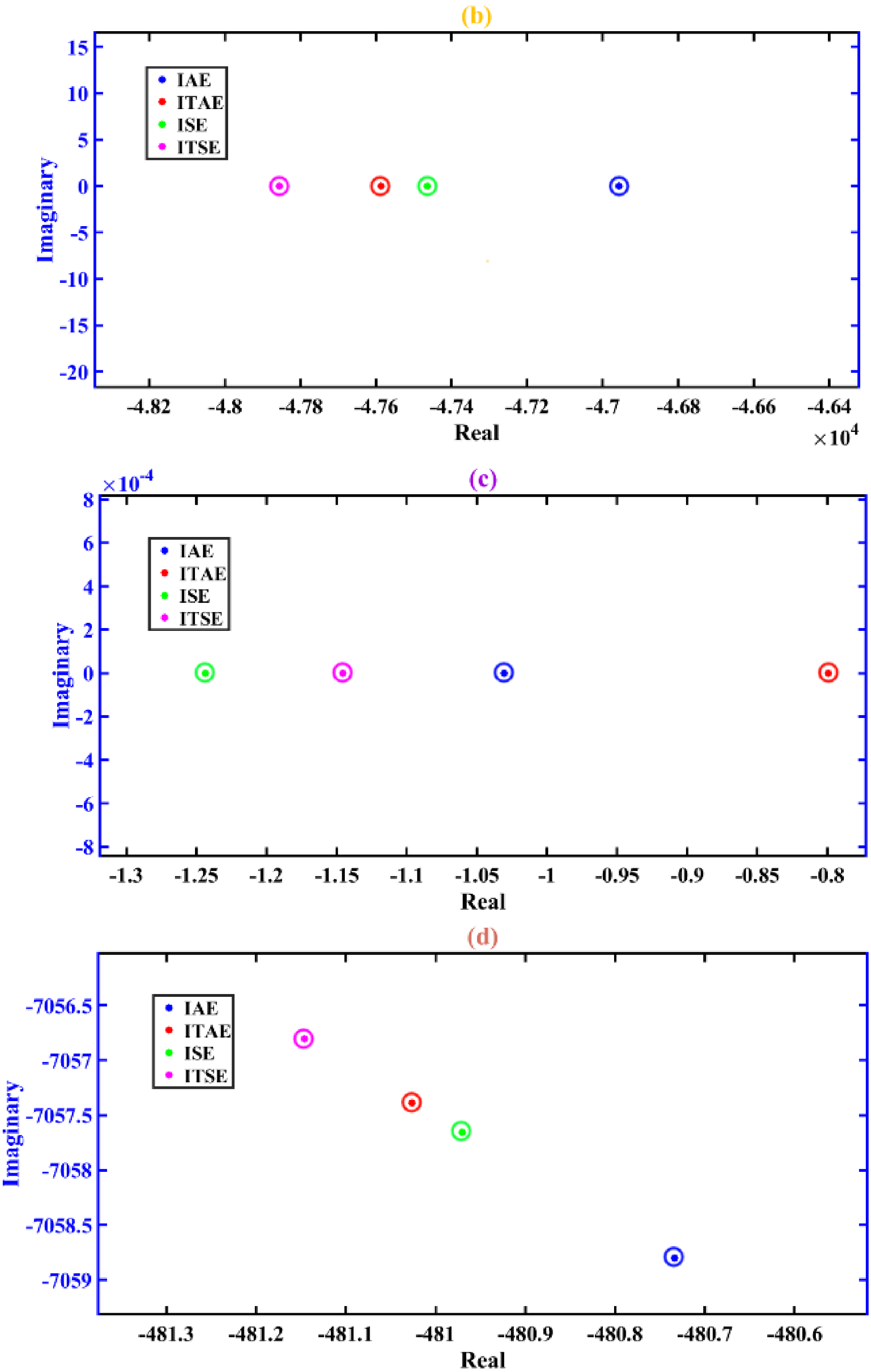
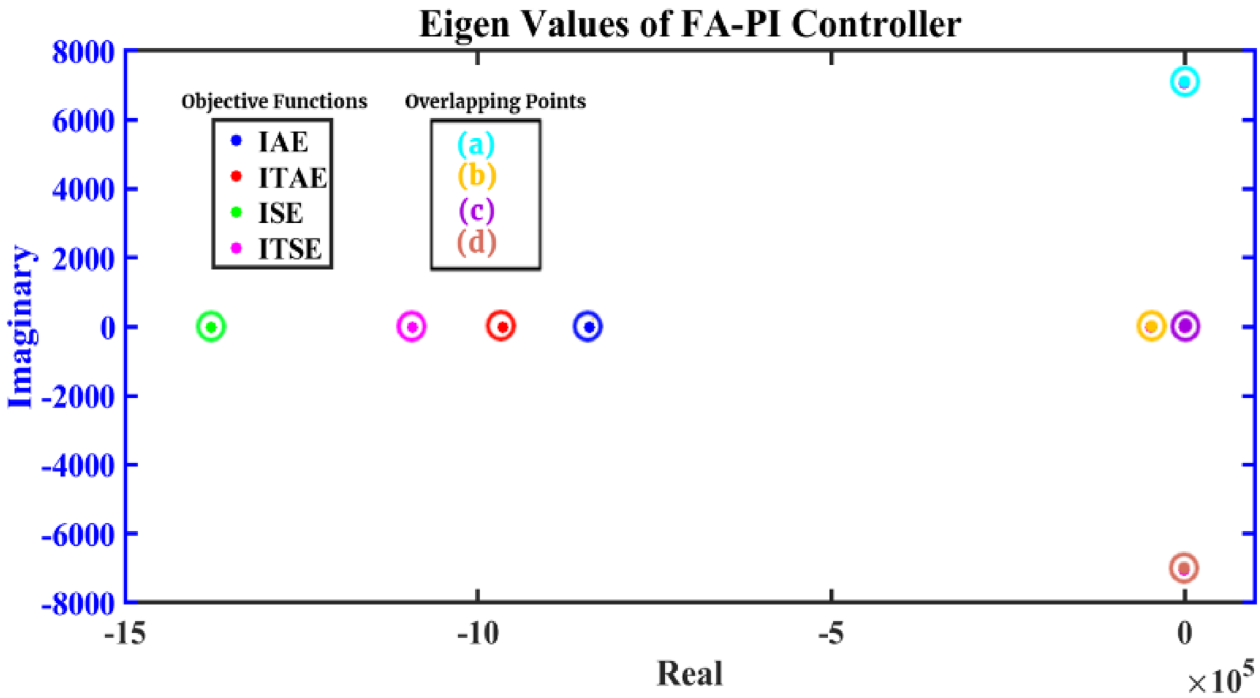
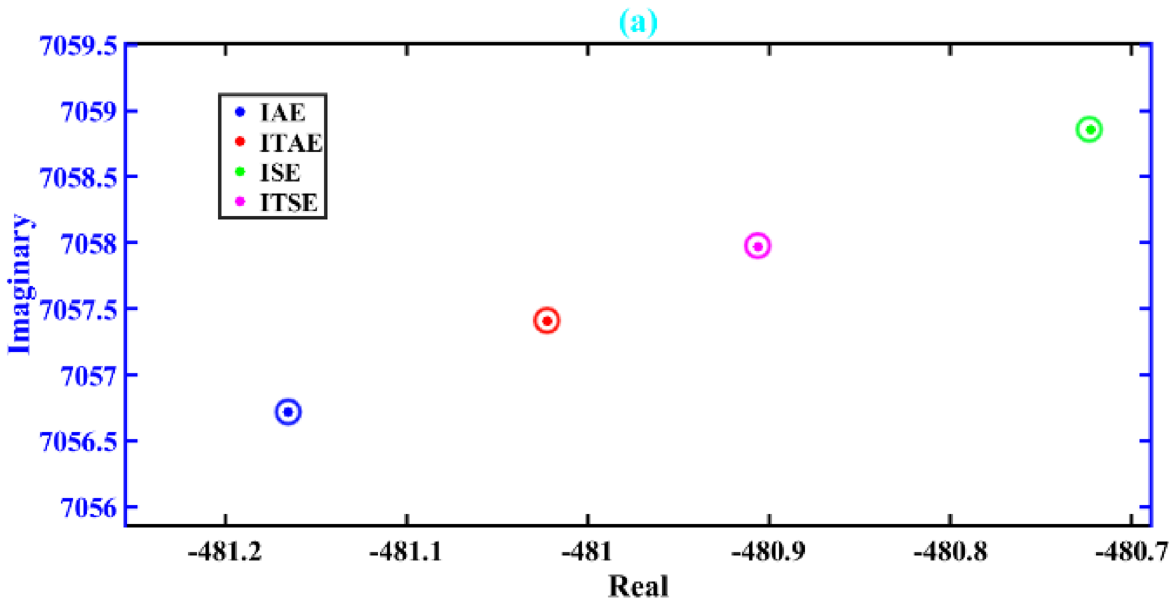


Fig.5.24 Eigen Values of PSO-PI Controller

5.4.4 FA-PI controller





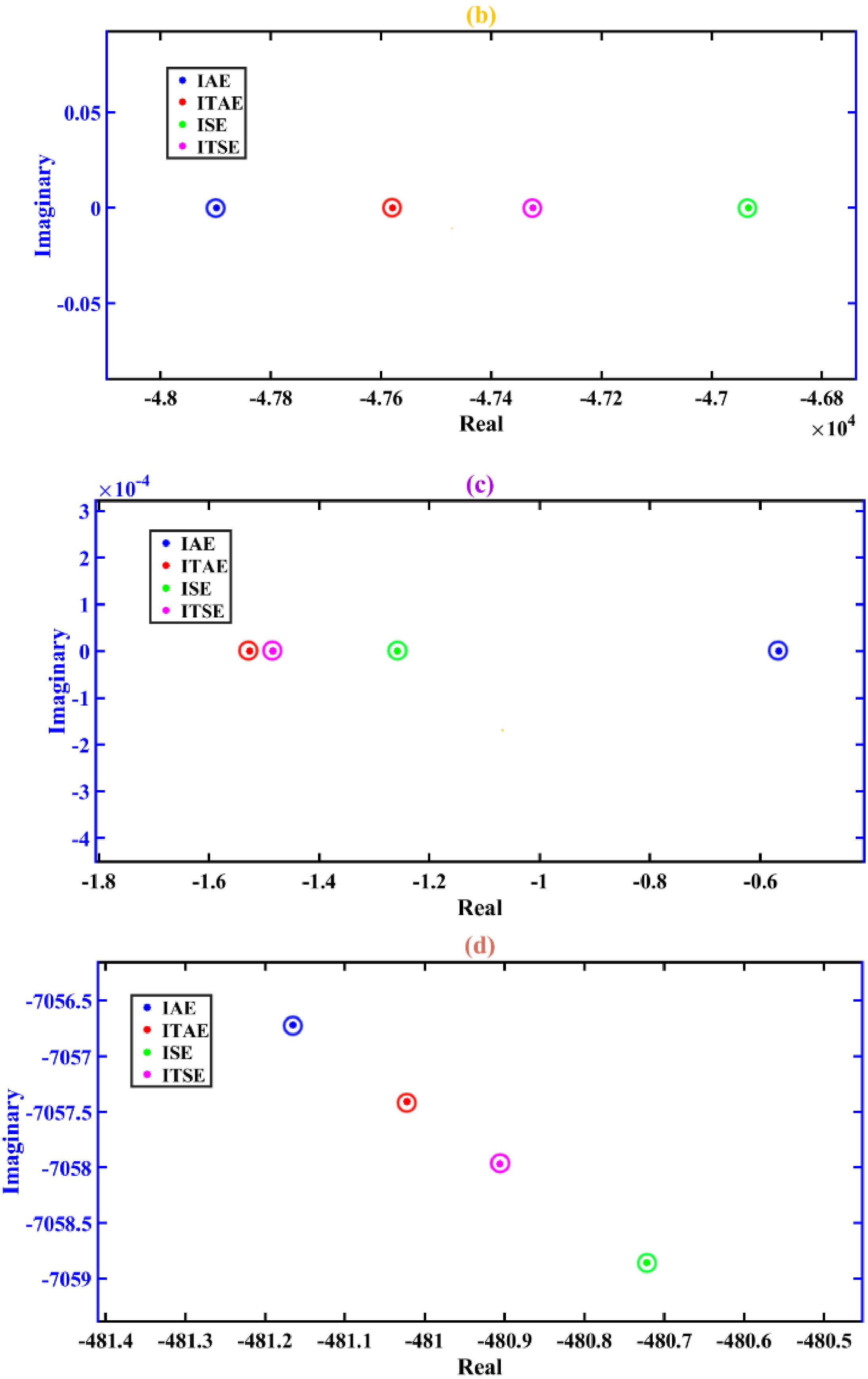
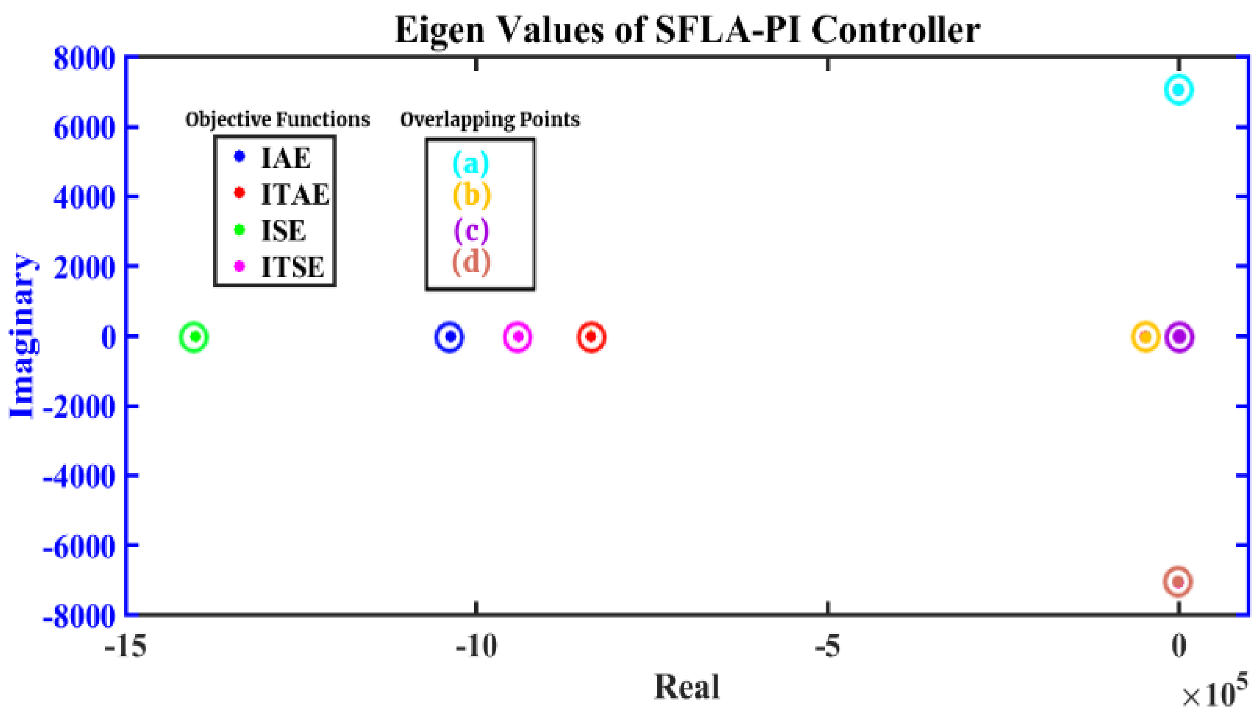
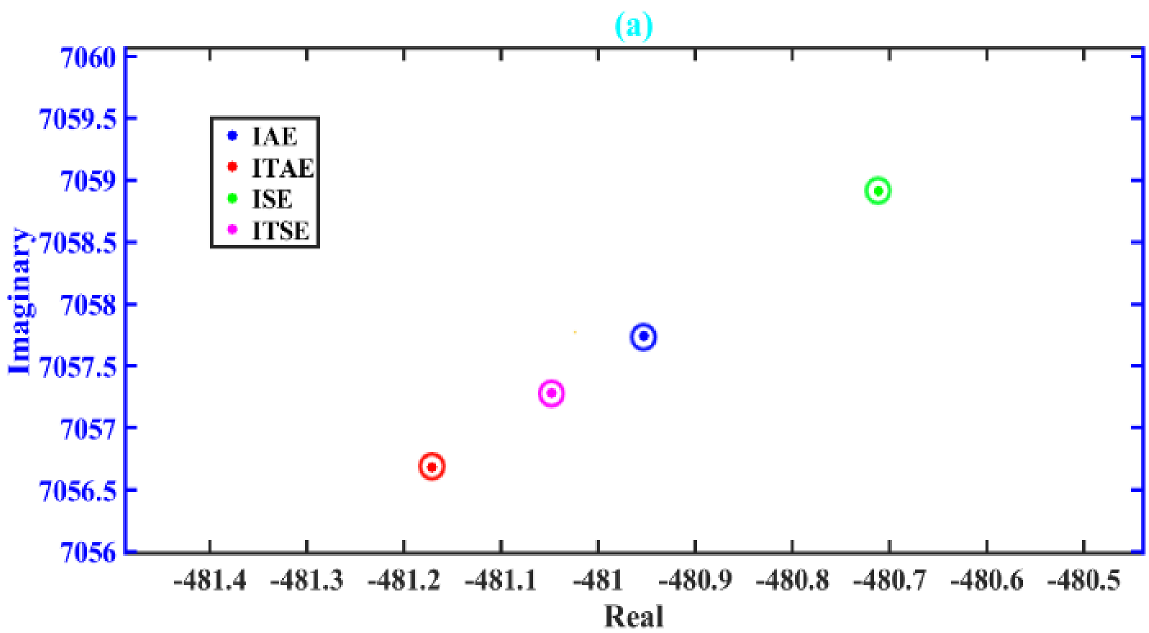


Fig.5.25 Eigen Values of FA-PI Controller

5.4.5 SFLA-PI controller





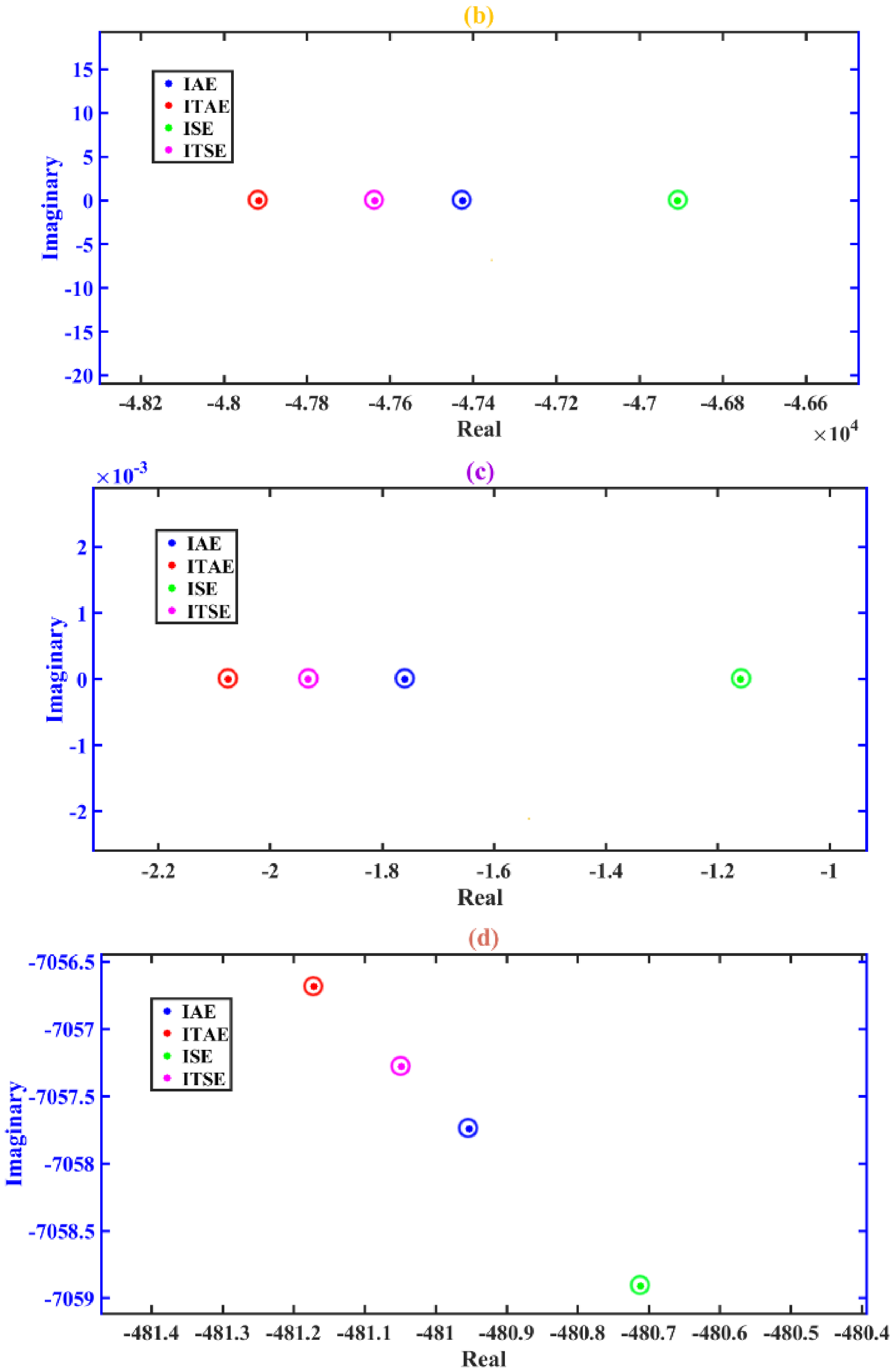


Fig.5.26 Eigen Values of SFLA-PI Controller

As seen in the Fig. (5.22-26), their imaginary components are organized in such a way that they appear to cancel each other out. Furthermore, The ISE objective function for ACOR-PI is more stable than the others, since it is on the leftmost side of the s-plane, with the largest negative real fraction of eigenvalue which is (1.0e+06 *(-1.4193 + 0.0000i)) among all five SIA-PI controller, as seen by the eigenvalue figures. However, the largest negative real eigenvalue of SFLA-PI from the ISE objective function is 1.0e+06 *(-1.4004 + 0.0000i), which is relatively identical to ACOR-PI.

CHAPTER 6

CONCLUSION AND PERSPECTIVE PLANS

6.1 Synopsis

This thesis investigates the idea of enhancing the PI controller by utilizing five different swarm intelligence algorithms with four different Zeta converter objective functions. To assess converter's stability IAE, ITAE, ISE, and ITSE performance measures are utilized as objective functions. In Chapter 2, the operational concept of dc-dc converters is briefly reviewed, followed by a summary of the various topologies. Furthermore, the traditional method to the zeta converter, which is of importance to us, is depicted, as well as modeling using State Space Averaging Technique. Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization for Continuous Domain (ACOR), Firefly Algorithm (FA), and Shuffled Frog Leaping Algorithm(SFLA) were discussed in detail in Chapter 3 of the book. The implementation of the techniques for designing an efficient PI controller was discussed in Chapter 4. The tuning of performance parameters has been explored, as well as the requisite objective functions, in a brief discussion. In Chapter 5, the simulation results of the SIA-based PI controller for Zeta converter were discussed. A graphical depiction of the outcomes was created with the use of five types of optimum SIA-PI controllers, and a comparison analysis was performed combining four objective functions for each controller.

The overall comparative analysis table is created upon observing the graphical representation of all comparative step responses in order to have a better insight of the compatibility of swarm intelligence algorithms in the improvement of the PI controller for the zeta converter. Firstly, the zeta converter's parameters have been specified, and the conventional PI controller's gain values and performance metrics have been recorded. Following the classic PI controller, the PI controller is fine-tuned utilizing swarm intelligence algorithm based approaches in order to reach a better optimized solution. Then, based on our findings, it shows that ACOR-PI-ISE is the best optimized controller in terms of performance characteristics among the five SIA-based PI controllers tested, with the lowest percentage of overshoot(2.2495), rise time(1.3979e -06), and settling time(8.0472e -06). Though it's worth noting that the SFLA-PI-ISE controller performs nearly identically to the ACOR-PI-ISE controller, with a percentage overshoot of 2.2750, a rise time of 1.4150e-06, and a settling time of 8.3296e-06. Lastly eigen value analysis is done to find stability state for each SIA-PI controller, and the eigen value table and figures is also supporting the results found in comparative step response analysis done earlier. Eventually, eigen value analysis is conducted to identify each SIA-PI controller's stability state, and the eigen value table and figures back up the findings of the comparative step response analysis. It is found from the eigen figures that the ACOR-PI-ISE has the real negative component at the leftmost side of s plane with compared to other four SIA-PI controllers. According to the eigen figures, the ACOR-PI-ISE has the real negative component of (1.0e+06 *(-1.4193 + 0.0000i)) at the leftmost side of the s plane when compared to the other four SIA-PI controllers. Furthermore, the eigen values acquired from each SIA-PI controller with each objective function show that the imaginary portions cancel each other out to make system nonoscillatory, and real values are all on the s-left plane's side which ensures the SIA-PI controller is stable. As a result, it's apparent that SIA based optimization approaches for improving the PI controller for our zeta converter produce superior performance.

6.2 Perspective Plans

This study will bring attention to the demand for more research into the design and performance of zeta converter. Different methodologies will be deployed in the future to create an improved PI controller to explore and strengthen the converters' stability. Besides that, the employed five swarm intelligence algorithm based strategies can be also applied to different converters other than zeta to optimize their robustness. Nevertheless, for extensive researches to reassert this workflow, a hardware execution is also plausible. Hence, the extensive comparative analysis can be quite valuable for evaluating stable circumstances in a variety of power electronics applications.

References

[1] Wilson, Thomas G. "The evolution of power electronics." *IEEE Transactions on Power electronics* 15.3 (2000): 439-446.

[2] Blaabjerg, Frede, Tomislav Dragicevic, and Pooya Davari. "Applications of power electronics." (2019): 465.

[3] Gheisarnejad, Meysam, Hamed Farsizadeh, and Mohammad Hassan Khooban. "A novel non-linear deep reinforcement learning controller for DC/DC power buck converters." IEEE Transactions on Industrial Electronics (2020).

[4] Bose, Bimal K. "Power electronics-an emerging technology." *IEEE Transactions on Industrial electronics* 36.3 (1989): 403-412.

[5] Babu, P. Ramesh, S. Ram Prasath, and R. Kiruthika. "Simulation and performance analysis of CCM Zeta converter with PID controller." *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015.

[6] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 2nd ed., Kluwer Academic Publishers, 2001.

[7] Middlebrook, Richard D., and Slobodan Ćuk. "A general unified approach to modelling switching-converter power stages." *International Journal of Electronics Theoretical and Experimental* 42.6 (1977): 521-550.

[8] Vuthchhay, Eng, and Chanin Bunlaksananusorn. "Dynamic modeling of a Zeta converter with State-Space averaging technique." *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Vol. 2. IEEE, 2008.

[9] . Nagarajan, R., et al. "Implementation of PI Controller for Boost Converter in PV System." *International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)* 11 (2016): 6-10.

[10] Selvi, R. Kalai, and R. Suja Mani Malar. "A bridgeless Luo converter-based speed control of switched reluctance motor using Particle Swarm Optimization (Pso) tuned proportional integral (Pi) controller." *Microprocessors and Microsystems* 75 (2020): 103039.

[11] Chakraborty, Amrita, and Arpan Kumar Kar. "Swarm intelligence: A review of algorithms." *Nature-inspired computing and optimization* (2017): 475-494.

[12] Sun, Weifeng, et al. "A survey of using swarm intelligence algorithms in IoT." *Sensors* 20.5 (2020): 1420.

[13] Bansal, Jagdish Chand. "Particle swarm optimization." *Evolutionary and swarm intelligence algorithms*. Springer, Cham, 2019. 11-23.

[14] Dorigo, Marco, and Thomas Stützle. "Ant colony optimization: overview and recent advances." *Handbook of metaheuristics* (2019): 311-351.

[15] Santana Jr, Clodomir J., et al. "A novel binary artificial bee colony algorithm." *Future Generation Computer Systems* 98 (2019): 180-196.

[16] Wang, Chun-Feng, and Wen-Xin Song. "A novel firefly algorithm based on gender difference and its convergence." *Applied Soft Computing* 80 (2019): 107-124.

[17] Pérez-Delgado, María-Luisa. "Color image quantization using the shuffled-frog leaping algorithm." *Engineering Applications of Artificial Intelligence* 79 (2019): 142-158.

[18] Bayat, Farhad, Maziyar Karimi, and Asghar Taheri. "Robust output regulation of Zeta converter with load/input variations: LMI approach." *Control Engineering Practice* 84 (2019): 102-111.

[19] Arun, S., and T. Manigandan. "Design of ACO based PID controller for zeta converter using reduced order methodology." *Microprocessors and Microsystems* 81 (2021): 103629.

[20] Thirumeni, Mariammal, and Deepa Thangavelusamy. "Design and analysis of hybrid PSO–GSA tuned PI and SMC controller for DC–DC Cuk converter." *IET Circuits, Devices & Systems* 13.3 (2019): 374-384.

[21] . Sonmez, Yusuf, et al. "Improvement of buck converter performance using artificial bee colony optimized-pid controller." *Journal of Automation and Control Engineering* 3.4 (2015).

[22] . Dhivya, P., and K. Ranjith Kumar. "MPPT based control of sepic converter using firefly algorithm for solar PV system under partial shaded conditions." *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*. IEEE, 2017.

[23] Sivakumar, S., et al. "An assessment on performance of DC–DC converters for renewable energy applications." *Renewable and Sustainable Energy Reviews* 58 (2016): 1475-1485.

[24] Taghvaee, M. H., et al. "A current and future study on non-isolated DC–DC converters for photovoltaic applications." *Renewable and sustainable energy reviews* 17 (2013): 216-227.

[25] Baharudin, Nor Hanisah, et al. "Topologies of DC-DC converter in solar PV applications." *Indonesian Journal of Electrical Engineering and Computer Science* 8.2 (2017): 368-374.

[26] Dave, Mitulkumar R., and K. C. Dave. "Analysis of boost converter using PI control algorithms." *International Journal of Engineering Trends and Technology* 3.2 (2012): 71-73.
[27] Mohan Ned, Undeland Tore M. and Robbins William P. ,"Power Electronics ,Converters Applications and Design", John Wiley & Sons, Inc., Book, 1995.
[28] C. K. Tse and K. M. Adams, "Qualitative analysis and control of a DC-to-DC Boost Converter Operating in Discontinuous Mode", IEEE Transactions on Power Electronics, Vol. 5, No. 3, July 1999, Pp 323-330.
[29] Ejury, Jens. "Buck converter design." *Infineon Technologies North America (TFNA) Corn Desion Note* 1 (2013).
[30] Redl, Richard, Brian P. Erisman, and Zoltan Zansky. "Optimizing the load transient response of the buck converter." *APEC'98 Thirteenth Annual Applied Power Electronics Conference and Exposition*. Vol. 1. IEEE, 1998.
[31] Duong, Minh Quan, et al. "Design and simulation of PI-type control for the Buck Boost converter." *2017 International Conference on Energy and Environment (CIEM)*. IEEE, 2017.
[32] Durgadevi, Subbiah, and Mallapu Gopinath Umamaheswari. "Analysis and design of single phase power factor correction with DC–DC SEPIC Converter for fast dynamic response using genetic algorithm optimised PI controller." *IET Circuits, Devices & Systems* 12.2 (2018): 164-174.
[33] Eng, Vuthchhay, Unnat Pinsopon, and Chanin Bunlaksananusorn. "Modeling of a SEPIC converter operating in continuous conduction mode." *2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Vol. 1. IEEE, 2009.
[34] Chen, Zengshi. "PI and sliding mode control of a Cuk converter." *IEEE Transactions on Power Electronics* 27.8 (2012): 3695-3703.
[35] H. Sira-Ramirez and R. Silva-Origoza, Control Design Techniques in Power Electronics Devices. New Mexico: Springer, 2006
[36] Kazimierczuk, Marian K., and Jacek J. Jozwik. "Optimal topologies of resonant DC/DC converters." *IEEE transactions on aerospace and electronic systems* 25.3 (1989): 363-372.
[37] Babu, P. Ramesh, S. Ram Prasath, and R. Kiruthika. "Simulation and performance analysis of CCM Zeta converter with PID controller." *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015.
[38] . Oommen, Sujo, et al. "Zeta Converter Simulation for Continuous Current Mode Operation." *International Journal of Advanced Research in Engineering and Technology* 10.1 (2020): 2019.
[39] Nishat, Mirza Muntasir, et al. "An Optimal Design of PID controller for DC-DC Zeta converter using Particle Swarm Optimization." *2020 23rd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2020.
[40] Vuthchhay, Eng, and Chanin Bunlaksananusorn. "Modeling and control of a Zeta converter." *The 2010 International Power Electronics Conference-ECCE ASIA-*. IEEE, 2010.
[41] Babu, P. Ramesh, S. Ram Prasath, and R. Kiruthika. "Simulation and performance analysis of CCM Zeta converter with PID controller." *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015.
[42] Davoudi, Ali, Juri Jatskevich, and Tom De Rybel. "Numerical state-space average-value modeling of PWM DC-DC converters operating in DCM and CCM." *IEEE Transactions on power electronics* 21.4 (2006): 1003-1012.
[43] Ahmad, Ali H., and Nashwan Saleh Sultan. "Design and Implementation of controlled Zeta Converter power supply." *American Journal of Electrical and Electronic Engineering* 2.3 (2014): 121-128.
[44] Babu, P. Ramesh, S. Ram Prasath, and R. Kiruthika. "Simulation and performance analysis of CCM Zeta converter with PID controller." *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015.
[45] Wang, Dongshu, Dapei Tan, and Lei Liu. "Particle swarm optimization algorithm: an overview." *Soft Computing* 22.2 (2018): 387-408.

[46] Sabanci, Kadir, and Selami Balci. "Development of an expression for the output voltage ripple of the DC-DC boost converter circuits by using particle swarm optimization algorithm." *Measurement* 158 (2020): 107694.

[47] Chen, Wei-Neng, et al. "A novel set-based particle swarm optimization method for discrete optimization problems." IEEE Transactions on evolutionary computation 14.2 (2009): 278-300.
[48] Deželak, Klemen, et al. "Proportional-Integral Controllers Performance of a Grid-Connected Solar PV System with Particle Swarm Optimization and Ziegler–Nichols Tuning Method." *Energies* 14.9 (2021): 2516.
[49] Shi, Yuhui. "Particle swarm optimization: developments, applications and resources." Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546). Vol. 1. IEEE, 2001.

[50] Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding of the IEEE international conference on neural networks, Piscataway, NJ, pp 1942–1948
[51] Sundareswaran, K., Kiran Kuruvinashetti, and P. S. Nayak. "Application of Particle Swarm Optimization for output voltage regulation of dual input buck-boost converter." 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). IEEE, 2014
[52] Sundareswaran, K., Kiran Kuruvinashetti, and P. S. Nayak. "Application of Particle Swarm Optimization for output voltage regulation of dual input buck-boost converter." 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). IEEE, 2014
[53] Osman, I.H. and Laporte, G. (1996). Metaheuristics: A bibliography. Annals of Operations Research. Vol. 63, pp 513-623.
[54] Altabeeb, Asma M., et al. "Solving capacitated vehicle routing problem using cooperative firefly algorithm." *Applied Soft Computing* 108 (2021): 107403.
[55] Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. Journal in ACM Computing Surveys. Vol. 35, Issue 3, pp 268-308.
[56] Bonabeau, E. (2003). Swarm Intelligence. O'reilly Emerging Technology Conference.
[57] Bonabeau, E. (2003). Swarm Intelligence. O'reilly Emerging Technology Conference.
[58] Johari, Nur Farahlina, et al. "Firefly algorithm for optimization problem." Applied Mechanics and Materials. Vol. 421. Trans Tech Publications Ltd, 2013.
[59] Yang, Xin-She. "Firefly algorithm, stochastic test functions and design optimisation." *International journal of bio-inspired computation* 2.2 (2010): 78-84.
[60] Yang, Xin-She, and Xingshi He. "Firefly algorithm: recent advances and applications." International journal of swarm intelligence 1.1 (2013): 36-50
[61] Kannan, G., D. Padma Subramanian, and RT Udaya Shankar. "Reactive power optimization using firefly algorithm." Power Electronics and Renewable Energy Systems. Springer, New Delhi, 2015. 83-90
[62] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[63] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." Applied mathematics and computation 214.1 (2009): 108-132
[64] Karaboga, Dervis, and Bahriye Basturk. "On the performance of artificial bee colony (ABC) algorithm." Applied soft computing 8.1 (2008): 687-697.
[65] Liu, Hongzhi, et al. "An improved artificial bee colony algorithm." 2013 25th Chinese control and decision conference (CCDC). IEEE, 2013.
[66] Gozde, Haluk, M. Cengiz Taplamacioglu, and Ilhan Kocaarslan. "Comparative performance analysis of Artificial Bee Colony algorithm in automatic generation control for interconnected reheat thermal power system." International Journal of Electrical Power & Energy Systems 42.1 (2012): 167-178.
[67] Yang, Xin-She. "Swarm intelligence based algorithms: a critical analysis." Evolutionary intelligence 7.1 (2014): 17-28.
[68] Karaboga, Dervis, and Bahriye Basturk. "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems." International fuzzy systems association world congress. Springer, Berlin, Heidelberg, 2007.
[69] Şahin, Arzu Şencan, Bayram Kılıç, and Ulaş Kılıç. "Design and economic optimization of shell and tube heat exchangers using Artificial Bee Colony (ABC) algorithm." Energy Conversion and Management 52.11 (2011): 3356-3362.
[70] . Zhao, Dong, et al. "Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation." *Expert Systems with Applications* 167 (2021): 114122.
[71] Guan, Boxin, Yuhai Zhao, and Yuan Li. "An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems." *Expert Systems with Applications* 164 (2021): 114021.
[72] . Dariane, A. B., and A. M. Moradi. "Reservoir operating by ant colony optimization for continuous domains (ACOR) case study: Dez reservoir." *International Journal of Mathematical, Physical and Engineering Sciences* 3.2 (2009): 125-129.
[73] Xiaoqin, Y. A. N. G., L. E. I. Xiujuan, and Z. H. A. O. Jie. "Essential Protein Prediction Based on Shuffled Frog-Leaping Algorithm." *Chinese Journal of Electronics* 30.4 (2021): 704-711.
[74] . Lei, Deming, Youlian Zheng, and Xiuping Guo. "A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption." *International Journal of Production Research* 55.11 (2017): 3126-3140.

[75] . Huang, Yao, Xiao-Ning Shen, and Xuan You. "A discrete shuffled frog-leaping algorithm based on heuristic information for traveling salesman problem." *Applied Soft Computing* 102 (2021): 107085
[76] Roberts, Steve. "DC/DC book of knowledge." Austria: RECOM Engineering GmbH & Co KG (2015).
[77] Miaja, Pablo F., Miguel Rodriguez, Alberto Rodriguez, and Javier Sebastian. "A linear assisted DC/DC converter for envelope tracking and envelope elimination and restoration applications." In 2010 IEEE Energy Conversion Congress and Exposition, pp. 3825-3832. IEEE, 2010.
[78] Chetty, P. R. K. "Current injected equivalent circuit approach to modeling switching dc-dc converters." IEEE Transactions on Aerospace and Electronic Systems 6 (1981): 802-808.
[79] Liu, Kwang-Hwa, and Fred C. Lee. "Zero-voltage switching technique in DC/DC converters." In 1986 17th Annual IEEE Power Electronics Specialists Conference, pp. 58-70. IEEE, 1986.
[80] Kumar, J. Sai, and Tikeshwar Gajpal. "A Multi Input DC-DC Converter for Renewable Energy Applications." (2016).
[81] Ortiz, G., J. Biela, D. Bortis, and J. W. Kolar. "1 Megawatt, 20 kHz, isolated, bidirectional 12kV to 1.2 kV DC-DC converter for renewable energy applications." InPower Electronics Conference (IPEC), 2010 International, pp. 3212-3219. IEEE, 2010.
[82] Koutroulis, Eftichios, and Kostas Kalaitzakis. "Design of a maximum power tracking system for wind-energy-conversion applications."IEEE transactions on industrial electronics53, no. 2 (2006): 486-494.
[83] A. H. R. Rosa, T. M. de Souza, L. M. F. Morais, and S. I. Seleme. "Adaptive and nonlinear control techniques applied to sepic converter in dc-dc, pfc, ccm and dcm modes using hil simulation." Energies 11, no. 3 (2018): 602
[84] Black, Harold S. "Stabilized feed-back amplifiers." Proceedings of the IEEE 87.2 (1999): 379-385.
[85] Bennett, Stuart. "Development of the PID controller." IEEE Control Systems Magazine 13.6 (1993): 58-62.
[86] Bennett, Stuart. "A brief history of automatic control." IEEE Control Systems Magazine 16.3 (1996): 17-25.
[87] Kaliannan, Jagatheesan, Anand Baskaran, and Nilanjan Dey. "Automatic generation control of Thermal-Thermal-Hydro power systems with PID controller using ant colony optimization." International Journal of Service Science, Management, Engineering, and Technology (IJSSMET) 6.2 (2015): 18-34.
[88] Rajesh, K. S., S. S. Dash, and Ragam Rajagopal. "Hybrid improved firefly-pattern search optimized fuzzy aided PID controller for automatic generation control of power systems with multi-type generations." Swarm and evolutionary computation 44 (2019): 200-211.
[89] Hasanien, Hany M. "Whale optimisation algorithm for automatic generation control of interconnected modern power systems including renewable energy sources." IET Generation, Transmission & Distribution 12.3 (2017): 607-614.
[90] Zhong, Jinghua. "PID controller tuning: A short tutorial." Mechanical Engineering, Purdue University (2006): 1-10.
[91] G. J. Costa, "Tuning a PID controller." Power Transmission Engineering 5.2 (2011): 26-31.
[92] Mehrabian, Ali Reza, and Caro Lucas. "Automatic tuning of decentralized controllers by swarm intelligence." *2006 3rd International IEEE Conference Intelligent Systems*. IEEE, 2006
[93] Shahani, Zulfiqar Ali, Ashfaque Ahmed Hashmani, and Muhammad Mujtaba Shaikh. "Steady state stability analysis and improvement using eigenvalues and PSS." *Engineering, Technology & Applied Science Research* 10.1 (2020): 5301-5306.