AN INVESTIGATION INTO THE UTILIZATION OF SWARM INTELLINGENCE FOR THE CONTROL OF THE DOUBLY FED INDUCTION GENERATOR UNDER THE INFLUENCE OF SYMMETRICAL AND ASSYMMETRICAL VOLTAGE DIPS

BY

KUMESHAN REDDY

Submitted in fulfilment of the academic requirements for the degree of Doctor of Philosophy in Electrical Engineering

College of Agriculture, Science and Engineering University of KwaZulu-Natal

DECEMBER 2022

Academic Supervisor: Prof. Akshay Kumar Saha

AN INVESTIGATION INTO THE UTILIZATION OF SWARM INTELLINGENCE FOR THE CONTROL OF THE DOUBLY FED INDUCTION GENERATOR UNDER THE INFLUENCE OF SYMMETRICAL AND ASSYMMETRICAL VOLTAGE DIPS

BY

KUMESHAN REDDY

SUBMITTED IN FULFILMENT OF THE DEGREE OF

Doctor of Philosophy in Electrical Engineering

College of Agriculture, Science and Engineering University of KwaZulu-Natal

SUBMISSION DATE

DECEMBER 2022

SUPERVISED BY

Prof. Akshay Kumar Saha

As the candidate's supervisor, I agree to the submission of this thesis

Signed:

Name: Prof. A.K. Saha

Date: 15/12/2022

COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION 1 - PLAGIARISM

I, Kumeshan Reddy, declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.

2. This thesis has not been submitted for any degree or examination at any other university.

3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
   a. Their words have been re-written but the general information attributed to them has been referenced
   b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.

5. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed

COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION 2 - PUBLICATIONS

DETAILS OF CONTRIBUTION TO PUBLICATIONS that form part and/or include research presented in this thesis (include publications in preparation, submitted, *in press* and published and give details of the contributions of each author to the experimental work and writing of each publication)

Publication 1: K. Reddy and A. K. Saha, "A review of swarm-based metaheuristic optimization techniques and their application to doubly fed induction generator," *Heliyon,* vol. 8, no. 10, 2022. doi : 10.1016/j.heliyon.2022.e10956

Publication 2: K. Reddy and A. K. Saha, "A modified Whale Optimization Algorithm for exploitation capability and stability enhancement," *Heliyon,* vol. 8, no. 10, 2022. doi :

10.1016/j.heliyon.2022.e11027

Publication 3: K. Reddy and A. K. Saha, "An Investigation into the Utilization of Swarm Intelligence for the Design of Dual Vector and Proportional–Resonant Controllers for Regulation of Doubly Fed Induction Generators Subject to Unbalanced Grid Voltages," *MDPI Energies,* vol. 15, no. 20, 2022. doi :

10.3390/en15207476

Publication 4: K. Reddy and A. K. Saha, "A Heuristic Approach to Optimal Crowbar Setting and Low Voltage Ride through of a Doubly Fed Induction Generator," MDPI: Energies, vol. 15, no. 24, 2022. doi :

10.3390/en15249307

Signed:

ACKNOWLEDGEMENTS

"Once you know who is walking with you on your path, you will never be afraid of anything"- source unknown. Firstly, all glory goes to God, for endless guidance throughout this research.

I would also like to express my deepest gratitude to my mentor, Prof Saha, for his constant motivation, endless support and timeous response.

To my loving parents, Raj and Irene, thank you for always supporting me throughout my academic career, and guiding me through any challenge I've faced.

My sweet and wonderful fiancé, Trinelle, for always being a shoulder to lean on, and taking the time to listen to my ideas.

To my sister, Shivani, and brother in law, Annasen, thank you for the constant words of encouragement, and for being my academic role models.

To my beautiful niece, Laiyana, thank you for being a ray of sunshine throughout my studies.

“Kalalō nī nāmasmaraṇa marava cakkani taṇḍri”- Bhadrachala Ramadasu.
Once again, all glory goes to God!

ABSTRACT

The rapid depletion of fossil, fuels, increase in population, and birth of various industries has put a severe strain on conventional electrical power generation systems. It is because of this, that Wind Energy Conversion Systems has recently come under intense investigation. Among all topologies, the Doubly Fed Induction Generator is the preferred choice, owing to its direct grid connection, and variable speed nature. However, this connection has disadvantages. Wind turbines are generally placed in areas where the national grid is weak. In the case of asymmetrical voltage dips, which is a common occurrence near wind farms, the operation of the DFIG is negatively affected. Further, in the case of symmetrical voltage dips, as in the case of a three-phase short circuit, this direct grid connection poses a severe threat to the health and subsequent operation of the machine. Owing to these risks, there has been various approaches which are utilized to mitigate the effect of such occurrences. Considering asymmetrical voltage dips, symmetrical component theory allows for decomposition and subsequent elimination of negative sequence components. The proportional resonant controller, which introduces an infinite gain at synchronous frequency, is another viable option. When approached with the case of symmetrical voltage dips, the crowbar is an established method to expedite the rate of decay of the rotor current and dc link voltage. However, this requires the DFIG to be disconnected from the grid, which is against the rules of recently grid codes. To overcome such, the Linear Quadratic Regulator may be utilized. As evident, there has been various approaches to these issues. However, they all require obtaining of optimized gain values. Whilst these controllers work well, poor optimization of gain quantities may result in sub-optimal performance of the controllers. This work provides an investigation into the utilization of metaheuristic optimization techniques for these purposes. This research focuses on swarm-intelligence, which have proven to provide good results. Various swarm techniques from across the timeline spectrum, beginning from the well-known Particle Swarm Optimization, to the recently proposed African Vultures Optimization Algorithm, have been applied and analysed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 : Introduction

1.1. Motivation and incitement

The world is currently in energy despair. For decades, the production of electrical energy depended on fossil fuels, particularly coal. An abundance of coal meant no limit of the utilization of this fuel to produce electricity. However, in recent times, this fuel source has come under investigation. This occurred for two reasons. The first reason is the scarcity of such fuel. The construction of many industries, urbanization and increasing population has led for a greater demand of electricity. This, in turn, has caused coal to be depleted at an alarming rate Secondly, due to the intense use of coal for the purpose of electricity production, the harmful effects of this fuel on the atmosphere have become more pronounced [1] . These issues have paved the way for the introduction of renewable energy. Due to its cost effectiveness, wind energy conversion systems (WECS) are gaining widespread attention. WECS utilize both asynchronous and synchronous machines [2]. The principle of producing electricity from WECS remains the same as that of a conventional power plant, the difference being the source of mechanical power utilized in driving the prime mover. In WECS, the wind turbine blades are attached to the rotor of the generator. This allows the rotation of the wind turbine blades to be transferred to the rotor of the machine. Generally, there is an interface, usually in the form of a three-stage gearbox, between the blades and the rotor. This allowed the low rotational velocity of the blades to be converted to a higher and more usable velocity. With the addition of various control actions, this rotational speed is converted into electricity. Evidently, WECS comprises of various mechanical and electrical components. The relationship of such can be observed in figure 1.1 [3]:

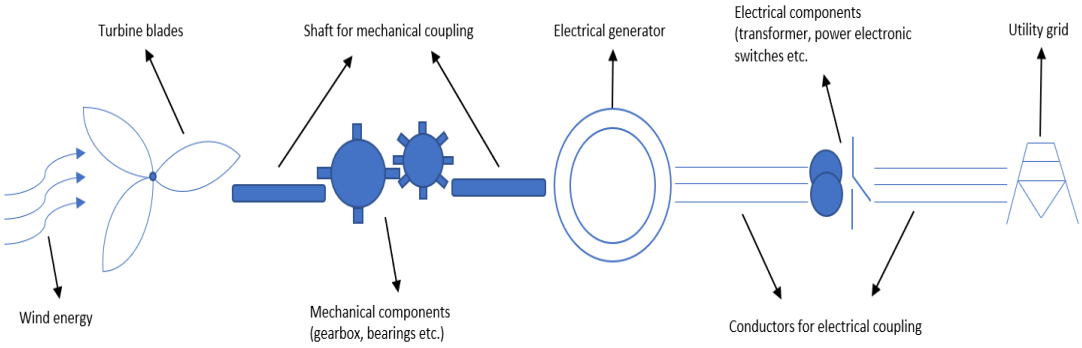


Figure 1.1: Conventional topology of WECS [3]

Due to their robust nature, cheap maintenance costs, and large power generation capabilities, WECS are rapidly becoming the alternative to fossil fuel-based power generation. Moreover, the use of direct grid connected generators ensures that the aspect of grid inertia is present, which is a critical part of power system stability. It can then be observed that research in the control of WECS is of utmost importance and should be carried out extensively.

1.2. Background of global wind energy utilization

The total global installed capacity of WECS has rapidly increased in modern times [4]. From 2013 to 2016, there has been a continual expansion in such systems. This is observed in figure 1.2 [5]. The two leaders in implementation of

such systems are the United States of America and China. With an installed capability of 237 MW, China is ahead by a significant margin. The United States has a capacity of approximately 106 MW, with Germany coming in third with a capacity of 62 MW [6]. Further, it should be noted that various European and Asian nations have recently exhibited a rapid increase in the installation of such systems [6]. This can be seen in figure 1.3, which depicts the magnitude of wind energy contribution from the top ten countries [6]. However, despite China and the USA yielded a larger magnitude of installed capacity, the total contribution of such systems to the total national energy consumption is only a fraction. This is observed in figure 1.4 [7]. Also from figure 1.4, is can be seen that despite European nations yielded a smaller magnitude of energy, this energy accounts for a much larger percentage of the total national energy consumption [5], [8]. The important fact, though, is that there has been a sharp rise in the utilization of wind energy for the production of electricity. This points to a green and sustainable future.

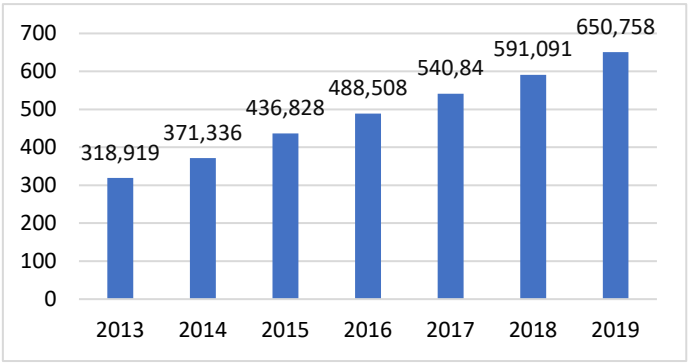


Figure 1.2: Total global capacity of wind energy systems (in GW) from 2013-2019 [5]

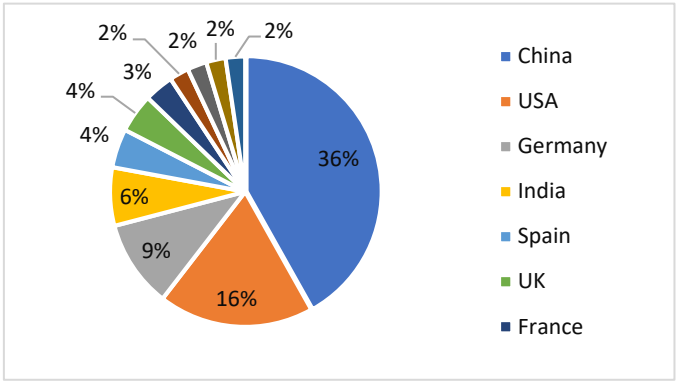


Figure 1.3: Participation of wind energy to global capacity of various nations [6]

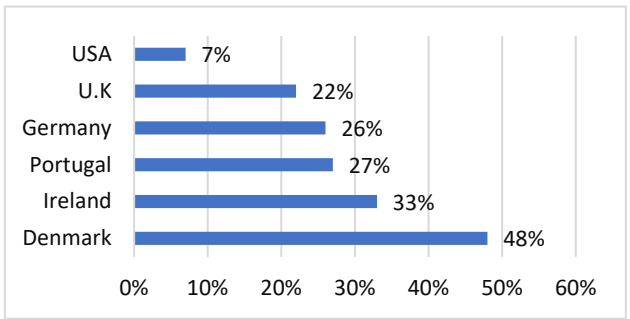


Figure 1.4: Infiltration of wind energy systems of several countries (%) [7]

1.3. Operation and control of current WECS

In wind farms, the generator most commonly utilized are the DFIG and permanent magnet synchronous generator [9].This is due to their capability to produce the maximum possible power, despite fluctuations in the wind velocity. Also, when compared to the squirrel cage induction machine, these machines produce a lower level of stress on the machine components. Despite the advantages of the PMSG, the DFIG proves to be the more efficient generator. Thus, majority of WECS utilize the DFIG [10], [11]. However, recent research has been conducted in the control of the PMSG. One such example can be found in [12], whereby a novel MOT, called Democratic Joint Operations Algorithm, was utilized for the purpose of obtaining PID controller gains. When compared to various other algorithms, the proposed algorithm produced the best overshoot and steady state error of the active power. The authors in [13] propose an adaptive Fractional Order PID controller for Maximum Power Point Tracking (MPPT) which utilizes a linear perturbation observer. The controller is easy to implement, does not require an accurate model, and exhibits a robust control performance. Owing to the rapid increase in the use of the DFIG, control of such needs to be efficient and effective. The most common and established DFIG control method is field oriented control. This control algorithm regulates the DFIG stator active and reactive power via control of the rotor current [14], [15]. This utilizes proportional-integral (PI) controllers. PI controllers are known to produce reliable and robust responses. The issue, however, is that PI controllers are require optimal tuning.

Achieving this via trial and error is a tedious task and may result in sub-optimal performance of the controller. One well-known method of PI controller tuning is the Ziegler-Nichols method, which utilizes either the closed-loop or open-loop response of the plant. Another method is the Cohen-Coon tuning method, which is similar to that of the Ziegler-Nichols method but makes use of different formulae to determine optimal controller performance [16], [17], [18]. These methods prove to be satisfactory, but often cease to meet the stringent levels of control presently required. This becomes an issue with grid code compliance, especially under abnormal conditions such as symmetrical and asymmetrical grid voltages, and fault ride through.

Recently, several alternatives to the PI controller have surfaced. One such method is Sliding Mode Control (SMC). In SMC, a pre-set trajectory is utilized along which the control variable is forced [3]. SMC offers robustness to parameter variations, external disturbances, nonlinear loads, and uncertainties [19], [20]. However, it suffers the demerit of chattering [21], [22]. Hysteresis control makes use of user defined bandwidths. The output of the hysteresis controllers is used to determine which converter switching state will be implemented [23]. This is commonly referred to as a Look-Up Table. This is a simple control method. When applied to the DFIG, it proved to provide efficient dynamic responses. However, the ripple in the output is extremely large, and the output of the stator current is severely distorted [24]. Artificial Neural Network (ANN) is a system which is based on the human central nervous system. ANN simulates a biological neural network [25], [26]. The merit of ANN includes the ability to work with incomplete knowledge and having a strong fault tolerance. However, ANN suffers the demerit of an unexplained behaviour of the network. This compromises the reliability of the network [27]. Further, ANN is known to have a greater than average computational burden [27], [28]. Model Predictive Control has been extensively applied in the process control industry and has recently shown promise in the field of electrical engineering [29]. It offers a simple structure but is built on the knowledge of accurate machine parameters [3]. Practically, machine resistance and inductance values are given in terms of a tolerance, making Model Predictive Control an unreliable control method.

When applied to the DFIG, it is observed that MPC produces a large steady state error, particularly at lower shaft angular velocities [30].

1.4. Research significance

Considering the control of a DFIG, aspects such as frequency and reactive power absorption/generation are required to have an extremely low error tolerance. This is important for the efficient operation and stability of the electrical grid. Poor control of these critical aspects may have catastrophic consequences. In addition to this, national grid code requirements are required to be met. The aim of this research is to thoroughly investigate the effect of utilizing Metaheuristic Optimization Techniques (MOT) in the control of the DFIG. In this research, swarm-based MOT are considered, with five techniques investigated and applied to the control of the DFIG. Each technique is investigated in terms of inception, mathematical modelling, and application procedure. Further, a review of fourteen techniques is provided. This is with regards to merits, demerits, and advancements of such and finally the application of these techniques to the control of the DFIG. This review provides a general review of the techniques and is aimed at researchers interested in the control of DFIG based WECS. The presented literature concerning algorithm advancements, in combination with the presented literature regarding the application of swarm-based MOT to DFIG control, will equip the researchers with sufficient knowledge to utilize a specific technique advancement in the application of DFIG control. This research, therefore, serves as a basis for scientific advancement concerning DFIG based WECS control.

The focus of this study is the optimal control of the DFIG based WECS, particularly under the influence of abnormal grid voltage conditions. Wind farms are generally situated where the grid is weak and may frequently experience asymmetrical voltage sags and swells. These conditions may have a negative effect on the steady-state operation of the DFIG based WECS. Current literature presents various mitigation measures to such circumstances, such as the dual vector control method, and the proportional resonant controller. The issue, however, is fine tuning of these controllers. These techniques yield the potential to produce strong amelioration to these voltage anomalies, but sub-optimal tuning of such may result in poor controller performance. This research, therefore, aims to investigate the effect of applying swarm intelligence to the design of such controllers. The research considers both single-phase, as well as dual-phase voltage dips, as well as controller response to grid frequency perturbation. Results are analyzed in terms of steady-state error, overshoot, and steady-state ripple. The control method utilized in this research is stator-flux oriented control, which implies that stator power control is achieved via rotor current regulation.

Another possible occurrence affecting wind generation systems is the effect of a symmetrical voltage dip. This may be due to the occurrence of a fault, or via start-up of large motors. Sudden, sharp changes in the grid voltage have catastrophic consequence on both the mechanical and electrical components of the DFIG. One such method of mitigation is the use of crowbar protection. In this method, during a symmetrical fault, the rotor terminals of the DFIG are short-circuit, to protection the rotor side converter of the machine. The size of the crowbar resistance should be carefully chosen, as a large size will increase the rotor voltage, but a smaller size will increase the rotor current. In this research, swarm intelligence is applied to the optimal sizing of the crowbar resistance, to ensure a strong trade-off between the rotor current and rotor voltage. Further, despite the crowbar method providing protection to the machine, it requires the DFIG to be disconnected from the grid. Modern grid codes require wind

energy systems to remains connected to the grid in case of fault, as well as to supply reactive power to assist in the restoration of the grid voltage. For this purpose, the linear quadratic regulator is investigated. Once again, swarm intelligence is applied to the system, to optimize the gains of the linear quadratic regulator.

1.5. Research questions

The research undertaken in this thesis aims to answer the following questions

- What are the methods of implementation of various swarm intelligence techniques? What are the factors which affect the suitability of the techniques to engineering optimization problems? What are the drawbacks of the various techniques, and what methods are proposed to mitigate these drawbacks?
- Has swarm intelligence been applied to the control of the DFIG?
- What is the level of effectiveness of applying swarm intelligence to control of the DFIG?
- How effective is swarm intelligence at ensuring a strong trade-off between rotor current and rotor voltage of the DFIG when applied to crowbar sizing? Further, how effective is swarm intelligence at optimizing the gains of a LQR to ensure strong Low Voltage Right Through characteristics of the DFIG?
- How well do the controllers designed using swarm intelligence react to grid frequency perturbations?

1.6. Aims and objectives

The aims and objectives of this research work is as follows:

- Conduct an in-depth review on various swarm intelligence technique. This is in terms of inception, mathematical modelling, method of implementation, merits, demerits, advances, and application of such to the control of the DFIG. Further, implement and analyze the response of various techniques when applied to CEC benchmark functions, as well as to practical engineering problems.
- Implement mathematical techniques to improve to capability of a swarm technique.
- Apply swarm intelligence to optimal crowbar sizing, and optimization of LQR gains and analyze the response in terms of LVRT capability.
- Apply swarm intelligence to optimization of controller gains for control under the influence of asymmetrical voltage dips, and analyse the steady-state response of the DFIG

Throughout the research, numerous state-of-the-art swarm-based metaheuristic optimization techniques have been applied and analysed. It is noted that despite the recent success of such techniques, these algorithms have not been extensively applied to the control of the DFIG. In some cases, the effect of application of these techniques have not been discussed at all. Therefore, by implementing and analysing various techniques, the research may better prove which techniques are better suited for particular applications

1.7. Limitations of research

Whilst this research aims to provide a critical insight into the utilization of Metaheuristic Optimization Techniques to the control of the DFIG during voltage anomalies, the following are the limitations which exist in this research:

- The investigation will only be conducted via simulation. The simulation chosen is MATLAB/Simulink. The software has powerful simulation properties, which allow for almost realistic operation.
- The grid side converter will be implemented but will only be designed for nominal conditions. This means that during the case of a fault condition, only the rotor side converter will be modified and evaluated.
- Since the aim of the study is to ensure acceptable operation of the DFIG in case of faults, the harmonic distortion of the output stator current will not be analyzed.
- Metaheuristic Optimization Techniques comprises of four categories. Swarm techniques, physics-based techniques, evolutionary techniques and human-based based. Due to the large number of techniques within each category, only swarm intelligence will be applied.
- Due to the significant number of swarm techniques, the literature review conducted will focus on fourteen techniques.
- To ensure brevity, only six techniques will be applied and investigated in this research.

This chapter provides an insight to the research work undertaken in this thesis. The aim of this chapter is to clearly highly the requirement of such research, and the subsequent significance of such. The chapter starts off by providing a motivation and incitement into the research work. Afterwards, a background on the recent use of wind energy conversion systems, is provided. Following this, a review of current operation and subsequent control of wind energy conversion systems, is presented. This highlights the need for the research, which is undertaken in this thesis, which is explained in detail. The research questions which this thesis aims to answer is also provided, along with the aims and objectives of the thesis. The limitations of the research work is also provided, followed by the publications arising from this research work. Finally, the structure of the thesis is presented.

1.8. Structure of thesis

The research work presented in this thesis is structured as follows:

Chapter one deals with the introduction to the research work being undertaken. The motivation for the research, background of global wind energy utilization, and operation and subsequent control of current WECS is provided. Thereafter, a critical analysis of the significance of this research work is presented. This clearly defines the lack which exists in current literature, and how this research work aims to contribute to current literature. Following this, the research questions to be answered in this thesis, aims and objectives of the thesis, and limitations of this research work, is presented. Lastly, relevant publications arising from this research work is stated

Chapter two provides a comprehensive analysis of fourteen swarm-based Metaheuristic Optimization Techniques. The chapter starts out by providing insight into the operation of the DFIG. Thereafter, a list of the various sub-categories of MOT, along with their relevant techniques, is provided. Afterwards, fourteen well-known swarm-based MOT are discussed. This is in terms of algorithm motivation, merits, demerits, proposed techniques to overcome demerits, and lastly, application of the relevant technique to the control of the DFIG. Upon completion of this, a table summarizing the captured information is presented, along with a table indicating the most suitable algorithm for specific application to the DFIG.. Lastly, a performance analysis of three common techniques is

carried out. These techniques are applied to well-known benchmark functions, at varying dimension magnitudes. Then results are analyzed in terms of mean value, standard deviation, and rate of convergence. This chapter contributes to current research by allowing other researchers to easily locate the information regarding the performance of the different algorithms, as well as determine the most suitable chose of algorithm for their particular application.

Chapter three provides insight into the research methodology utilized in this research. In this chapter, the relevant equations, and subsequent flowchart of execution of various swarm-based MOT, are presented. These techniques are Particle Swarm Optimization, Bat Algorithm, Gorilla Troops Optimization, African Vulture Optimization Algorithm, and the Whale Optimization Algorithm. Upon completion of such, a modified version of the conventional Whale Optimization Algorithm is presented. The mathematical modifications applied to the algorithm are clearly defined, along with a novel method of implementation. The modified algorithm is then applied to the CEC2019 benchmark functions and compared to various conventional and modified techniques. After such, the algorithm is applied to a practical engineering problem, with the subsequent results analyzed. The proposed algorithm is then critically analyzed in terms of advantages, and limitations.

Chapter 4 provides a heuristic approach to optimal crowbar setting and low voltage ride through of the DFIG. In this chapter, the well-known Particle Swarm Optimization, as well as the recently developed African Vultures Optimization Algorithm, were applied to the control of the DFIG when under the influence of symmetrical voltage dips. The chapter provides an insight into the significance of such works, along with a summary of techniques proposed in recent literature. Thereafter, the performance of the DFIG under the influence of such conditions is analyzed. The control theory pertaining to crowbar protection, as well as Linear Quadratic Regulator control is also provided. Thereafter, the relevant swarm intelligence techniques are applied to optimal crowbar setting, and its results analyzed. This was in terms of rotor current and rotor DC voltage transients. This is succeeded by application of the relevant swarm techniques to design of the Linear Quadratic Regulator, as well as the Demagnetizing Current Injection method, and the results analyzed and compared to the conventional control method. The system was subject to 70%, 80%, and 90% symmetrical voltage sags, with the relevant gain magnitudes provided. The chapter is then concluded, along with a stability performance analysis of the Linear Quadratic Regulator subject to a perturbation in grid frequency

Chapter 5 provides insight into the application of swarm-based Metaheuristic Optimization Techniques to the design of the dual vector control of the DFIG, under the influence of asymmetrical voltage dips. Single-phase voltage dips of 5% and 10%, a well as a dual-phase voltage dip of 7.5%, were applied and analyzed, with the relevant gain magnitudes provided. Three swarm intelligence techniques from across the spectrum were applied to the control system. These are Particle Swarm Optimization, Bat Algorithm, and Gorilla Troops Optimization algorithm. The system was tested in sub synchronous mode, with the control target attempting to be achieved was constant rotor voltage. For this application, four Proportional-Integral controllers were utilized, one for each of the rotor direct and quadrature axis positive and negative sequence currents. A comprehensive set of results are provided, including three-phase stator current, three-phase rotor voltage, and rotor direct and quadrature axis positive and negative sequence currents. Results were analyzed in terms of steady-state error, overshoot, and

steady-state ripple. The chapter starts by providing a brief background into the dual vector control method. Thereafter, the relevant control block diagram to be implemented is provided. This is succeeded by the results of the various experiments. The chapter is then concluded.

Chapter 6 provides an investigation into the utilization of swarm intelligence to the design of the Proportional-Resonant regulator. control of the DFIG, under the influence of asymmetrical voltage dips. Single-phase voltage dips of 5% and 10%, a well as a dual-phase voltage dip of 7.5%, were applied and analyzed, with the relevant gain magnitudes provided. Three swarm intelligence techniques from across the spectrum were applied to the control system. These are Particle Swarm Optimization, Bat Algorithm, and Gorilla Troops Optimization algorithm. The system was tested in sub synchronous mode. A comprehensive set of results are provided, including three-phase stator current, three-phase rotor voltage, and rotor direct and quadrature axis currents. Results were analyzed in terms of steady-state error, overshoot, and steady-state ripple. The chapter starts by providing a background of the current work, as well as the significance of the work undertaken in the chapter. Thereafter, theory pertaining to the Proportional-Resonant controller is provided. This is succeeded by the results of the experiments. Further, a stability analysis of the Proportional Resonant controller is carried out, where the system is subject to a grid frequency perturbation. The work presented in the chapter is then concluded.

Chapter 7 provides a comprehensive summary of the research work undertaken in this thesis. The chapter summarizes work present in current literature, as well as the research methodology proposed in this thesis. Afterwards, the methods applied to, and results obtained from the subsequent chapters, are comprehensively discussed. The chapter then concludes with a scope for future work

Chapter 2 : Literature Review

This chapter aims to provide an insight, as well as an in-depth analysis of the application of swarm-based metaheuristic optimization techniques to the doubly fed induction generator. First, some context of the operation and established control of the DFIG is given. Next, an introduction to metaheuristic optimization techniques, with a subsequent flowchart depicting the various categories of MOT, is provided. Succeeding this, fourteen swarm intelligence techniques are analysed. A brief explanation regarding inception is provided. Thereafter, the associated merits, demerits, and proposed mitigation measures are given. This is followed by a critical analysis of the application of these techniques to the control of the DFIG. A discussion and subsequent summary of the techniques is provided. Lastly, an investigation into the performance of three well-known swarm techniques is carried out, and the results analyzed.

2.1. An overview of the DFIG

The structure of the DFIG based WECS is such that the stator provides a direct grid connection, and the rotor makes use of a back-to-back converter to provide grid coupling. This unique rotor configuration allows the rotor to both absorb and supply electrical power, thus allowed for generator operation at any wind speed [31], [32]. The rotor supplies power at speeds greater than synchronous speed and absorbs power at speeds lower than synchronous speeds [32]. To ensure a constant output frequency, power is absorbed at slip frequency [33]. Figure 2.1 depicts the structure of the DFIG-WECS [3].

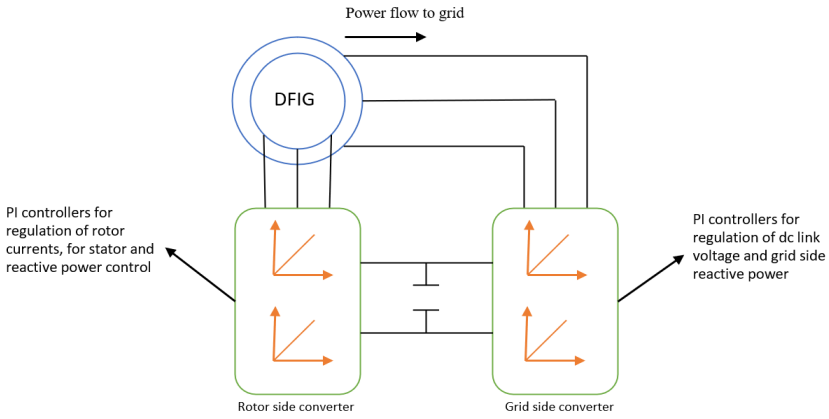


Figure 2.1: General structure of DFIG based WECS [3]

Considering the synchronous (d-q) reference frame, the DFIG voltage equations can be seen in [34], [35]. The DFIG rotor and stator fluxes can be represented as shown in [36]. The DFIG electromagnetic torque, as well as the stator and rotor active and reactive powers are shown in [37] and [38] respectively. When considering DFIG control, various control strategies exist. These are rotor current control, direct power control (DPC) and direct torque control (DTC) [34]. Rotor current control is conventionally achieved via field-oriented control (FOC). FOC utilizes PI controllers to regulate the rotor currents, thus allowing for indirect control of the stator active and reactive powers [35]. This is the most common method of control. The function block diagram of stator voltage FOC is observed in figure 2.2 [39]. In figure 2.2, c1 and c2 denote the coupling terms of the algorithm. The algorithm is explained in detail in [40]. DTC directly controls the generator rotor flux and torque. There are two ways in which conventional

DTC is achieved. In method one a look up table and hysteresis controllers (DTC-ST) are utilized. The second way involves the use of PI controllers [41], [42]. DPC control directly the stator reactive and active powers. DPC is achieved using the same methodology as in DTC-ST [43].

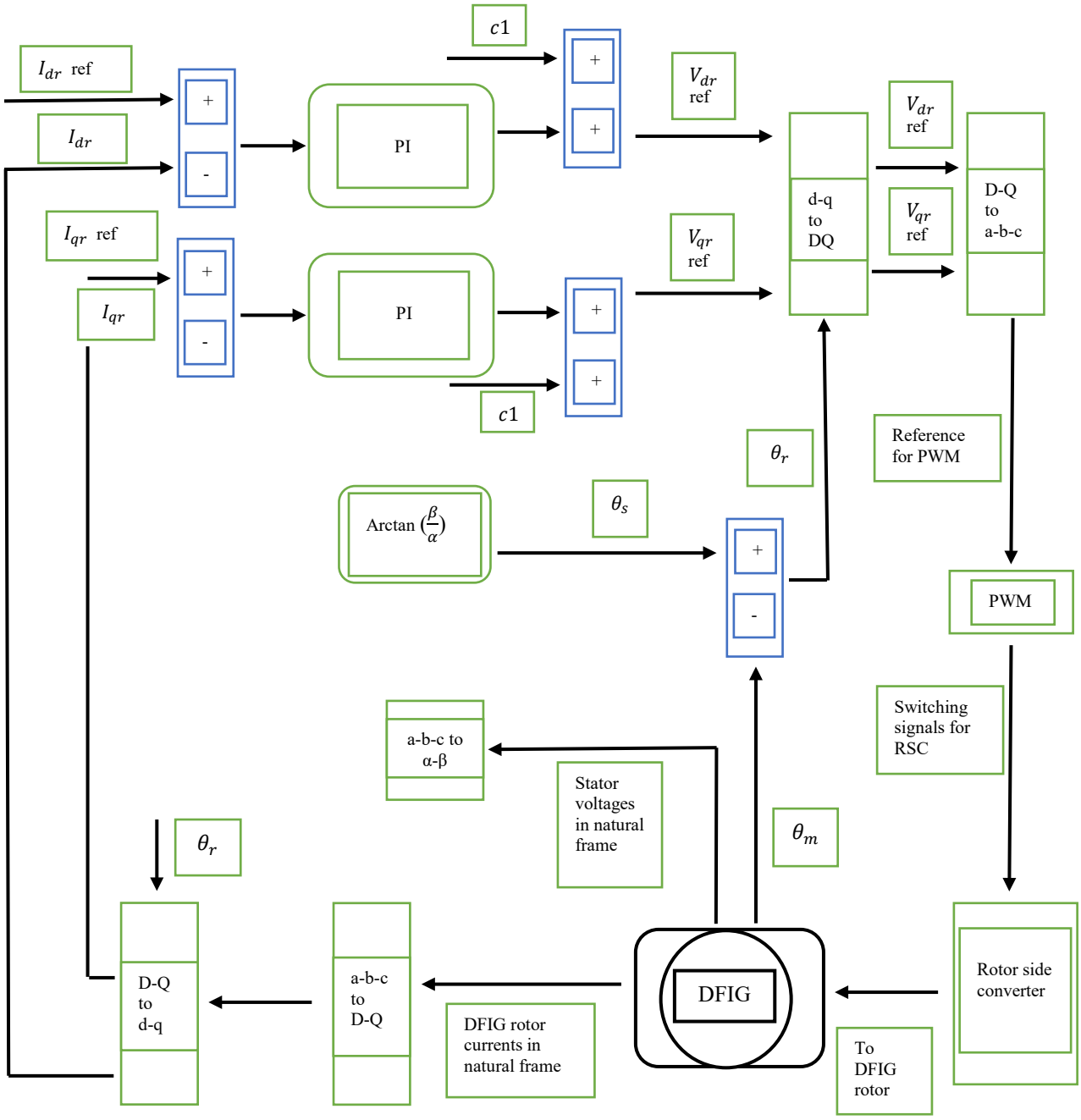


Figure 2.2: Structure of FOC of DFIG [39]

2.2. Metaheuristic Optimization Techniques

Metaheuristic Optimization Techniques, as the name suggests, are problem independent control techniques which has gain rapid popularity in the application of complex engineering problems. This can be attributed to their simplicity, flexibility, and capability to solve complex problems at a high efficiency rate. Metaheuristics techniques are based strongly on the concept of randomness, and search for optimal solutions based on diversification and intensification. Diversification is the scattered search of an entire search space and

intensification is the search in a particular area of a search space [44]. MOT are based on various aspects of everyday life, such as the human body, the laws of physics and the behavior of animals in their natural habitat [45]. Critical evaluation of the working processes of these aspects has allowed for accurate mathematical modelling of various nature-based occurrences. This in turn has been used to solve complex engineering problems successfully and optimally. While there does not exist any definitive way to categorize MOT, it can usually be classified into four categories [45], [46]. This can be seen in figure 8. The classification shown in figure 7 is not an exhaustive list of MOT but does account for most of the currently implemented techniques. The application of MOT has recently been applied to the control of the DFIG but has not been extensively researched. It has mostly been used to optimize the controller gains of the PI controllers used in the control of the DFIG. It is shown later that MOT make use of fitness functions. In terms of proportional-integral (PI) controllers, the various fitness functions (performance indices) are time varying functions of the integral of either the square of absolute value of the error being input into the PI controller [47], [48], [49].

Metaheuristic optimization techniques has been applied extensively to the field of renewable energy systems. In [50], a critical survey on the application of Metaheuristic optimization techniques on proton exchange fuel cell parameter estimation was carried out. The survey considered MOT from all four categories, both in conventional and modified natures, thereby making the survey extensive. Considering application to photovoltaic (PV) systems, the authors in [51] carried out a survey that reports the effects of utilizing MOT for identification of PV cell parameters. As in [50], the experiment considered techniques from all four categories, both in conventional and modified natures. The paper outlined in [52] conducts a comprehensive survey on the effect of MPPT algorithms of PV systems under the effect of partial shading. Seven categories of algorithms are considered, one of which is MOT. Within this, three of the four subcategories of MOT are discussed. Further to discussion of the conventional algorithms, the paper acknowledges the utilization of hybrid algorithms for the said application.

2.3. A review of various swarm-based MOT

The following section provides a review on various swarm-based MOT. The techniques that will be considered are Particle Swarm Optimization, Bacteria Foraging Optimization, Grey Wolf Optimization, Artificial Bee Colony Optimization, Whale Optimization Algorithm, Crow Search Algorithm, Bat Algorithm, Squirrel Search Algorithm, Moth Flame Optimization, Sailfish Optimization, Cuckoo Search Algorithm, Firefly Algorithm, Shuffled Frog Leaping Algorithm and Antlion Optimization. Each technique is discussed in terms of motivation, merits, demerits, advancements and finally their application to the control of the DFIG.

2.3.1. Particle Swarm Optimization

Utilizing the social conduct of the schooling of fish and the flocking of birds as inspiration, particle swarm optimization (PSO) is a MOT which, in was developed by an electrical engineer and social psychologist. PSO is a simple control algorithm which has a light computational burden [53] [54] [55], [56]. Despite its merit of a fast convergence speed [57], [58], the conventional PSO suffers the demerits of poor accuracy and being easily trapped in the local minima [57], [58], [59]. The authors in [57] introduced a mutation factor and a dynamic inertial factor. Large inertial factors enhance the convergence rate of the algorithm while small inertial factors enhance the search accuracy. The proposed dynamic inertial factor is a function of the fitness of all the particles, and lies in the range

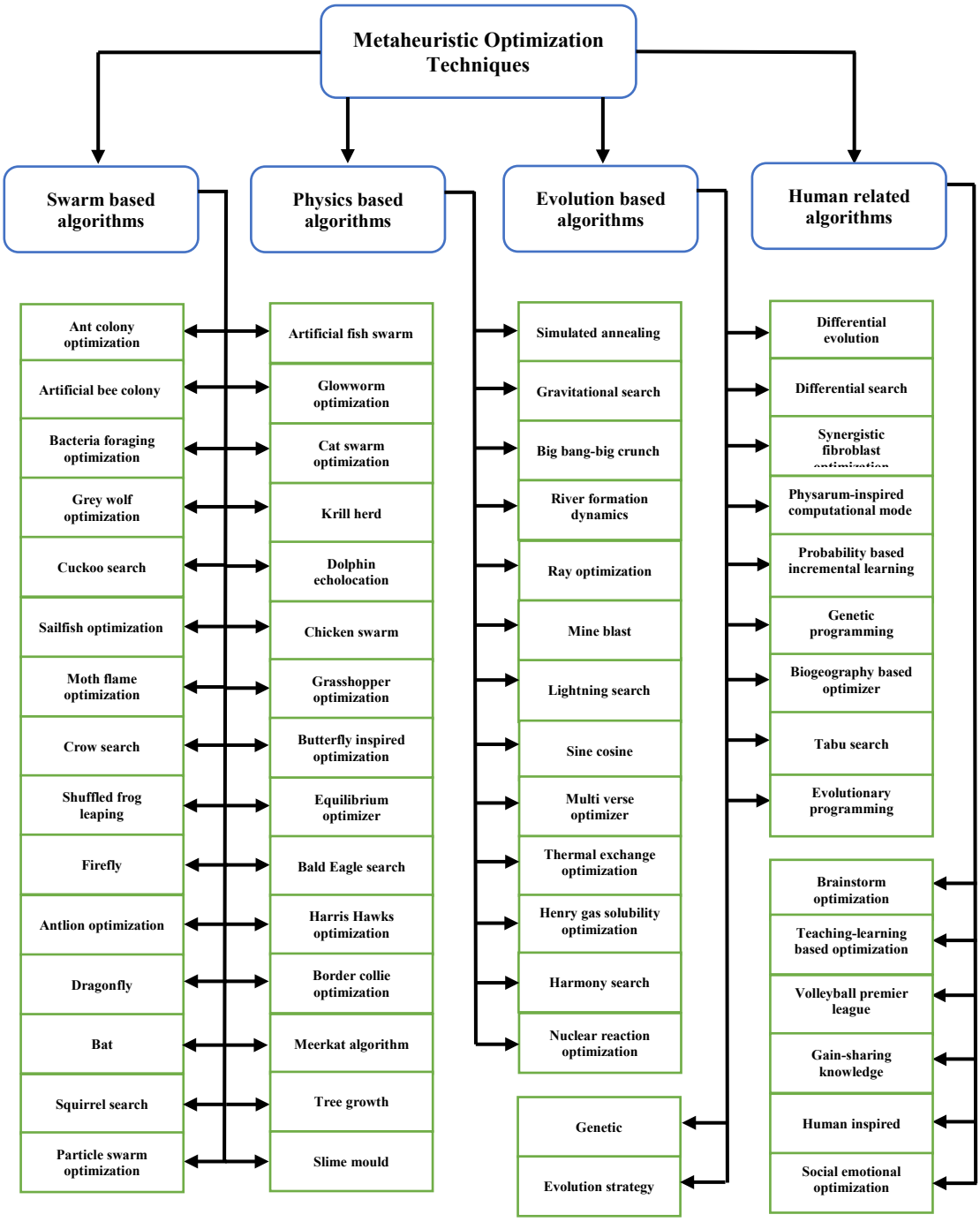


Figure 2.3: Flow chart indicating classification of MOT [44], [45]

[0,1]. Based on the position of all particles except the global best, the mutation factor randomly generates two new particles based on a probability. The proposed algorithm was applied to the 14- BUS system for reactive power optimization and compared to the conventional PSO. It was observed that after 100 iterations, the proposed algorithm produced a 1.28% improvement in the result. For iterations lower than four, MPSO is inferior to PSO. However, in practise, such minute values of iterations are not utilized.

Considering the application of PSO to the DFIG, the authors in [60] applied PSO to the optimization of the parameters in sliding mode control (SMC). Three different control algorithms were presented. The first algorithm was the conventional SMC, in which PSO was used to optimize the positive switching gain. The second algorithm

was the integral SMC, which is an advancement to the conventional SMC in the case of reduction in steady state error. In this algorithm, PSO was used to optimize both the positive switching gain and integral gain. The third algorithm was an intelligent sliding mode controller, which adds a proportional gain to the integral sliding mode controller. In this algorithm, PSO was used to optimize the positive switching gain, integral gain, and proportional gain. The proposed control methods were implemented using the rotor current control method, which means that the control of the DFIG stator active and reactive power was dependent on the control of the rotor direct and quadrature currents. The proposed control methods were tested on a 7.5 kW DFIG. From the results, it is observed that the integral sliding mode controller produced the best dynamic response for both the stator reactive power and active power. This is followed by the intelligent SMC, then the conventional SMC. The superiority was in the order of approximately 100% and 200% to the intelligent SMC and conventional SMC respectively. The results presented do not clearly show a difference in steady state error and steady state ripple among the three control algorithms. Furthermore, the results are not compared to other parameter optimization methods, such as other MOT or the Ziegler Nichols method. This means that the results of the proposed algorithms cannot be verified.

PSO is applied to the DFIG to analyze the small signal stability in [61]. PSO is used to optimize the PI controller gains for both the rotor side converter (RSC) and grid side converter (GSC), and the pitch controller. In total, twelve parameters were optimized. The system was tested on both small and large disturbances. For the small disturbances, the optimized controllers produced smaller overshoots for the dc link voltage, terminal voltage, stator reactive power and stator active power. The optimized controller also damped out the oscillations much quicker. The author claims that the optimized controllers produce a superior dynamic response but due to insufficient evidence, this claim cannot be validated. Considering large disturbances, the optimized controllers produced a better terminal voltage and lower peak dc link voltage. With regards to the stator active power, the optimized controllers continued to inject active power into the grid, whereas the un-optimized controllers failed to produce any active power. Considering the stator reactive power, the optimized controllers absorbed a lower amount of power. An advancement of [40] was carried out in [62], where sensitivity analysis is utilized to identify the unified dominant control parameters. These are the parameters that would be optimized using PSO, so that the algorithm intricacy is lessened. The authors make use of trajectory sensitivity, which measures the degree of change of a system based on a differential change on a specific parameter and eigenvalue sensitivity, which uses eigenvalues to determine the systems sensitivity towards a specific parameter. Using the trajectory sensitivity analysis, the integral gain of the grid voltage regulator and the proportional gains of both the direct and quadrature rotor current regulators were chosen as the dominant control parameters. Using the eigenvalue sensitivity analysis, the integral gain of the stator active power regulator, integral gains of both the direct and quadrature rotor current regulators, proportional gain of the dc link capacitor voltage regulator and the proportional gains of both the direct and quadrature grid current regulators were chosen as the dominant control parameters. In total, there were six parameters to be optimized using PSO. The proposed algorithm (UDCP-PSO) was tested on both a single machine bus system as well as a multi machine bus system for both small and large disturbances. The proposed algorithm is fared against the original ten parameter optimization algorithm (C-PSO) as well as a random parameter optimization algorithm (R-PSO).

Considering the single machine bus system under a small disturbance, the UCDP-PSO produced a smaller overshoot of stator active power when compared to C-PSO and R-PSO. The damping time for UCDP-PSO was

the same as C-PSO, which was superior to R-PSO. The dynamic response of all three are almost identical. For the stator reactive power, the percentage overshoot and damping time of UCDP-PSO and C-PSO are the same and superior to R-PSO. The dynamic response of all three are almost identical. An identical phenomenon was observed with respect to the dc link voltage. Considering the single machine bus system under large disturbance, the UCDP-PSO produced the smallest percentage overshoot. The damping time and dynamic response of all three algorithms appear to be the same, with any variance being negligible. An identical response was seen with regards to the stator reactive power, this time the UCDP-PSO algorithm producing superiority only marginally to C-PSO. Considering the dc link voltage, UCDP-PSO once again produced the smallest percentage overshoot. UCDP-PSO and C-PSO produce an identical damping time, which was superior to that of R-PSO. The dynamic response of all three algorithms was identical. Considering the multi machine bus system under small disturbance, UCDP-PSO and C-PSO produce the same percentage overshoot and damping time, which was superior to that of R-PSO. The dynamic response of all three algorithms was the same. For the stator output voltage, C-PSO produced the lowest overall percentage overshoot, with the damping time of UCDP-PSO and C-PSO being the same and superior to that of R-PSO. Once again, the dynamic response of all three algorithms was the same. A conventional PID control of a DFIG using PSO is implemented in [63]. However, the fitness function used was not a conventional one (such as ITAE), but rather a unique one. This unique fitness function is a function of the steady state error, settling time, rise time and overshoot. The proposed control algorithm was compared to the supervisory PID control method. The DFIG terminal voltage is dropped from 1 per unit to 0.5 per unit, before regaining to 1 per unit. The results showed that the proposed algorithm outperformed the supervisory control method in all aspects i.e., settling time, rise time, peak time, and percentage overshoot. The unbalance in the stator currents during the voltage drop was approximately the same for both the proposed algorithm and the supervisory control method.

A novel control structure for stability enhancement of a DFIG based ocean energy conversion system is proposed in [64]. The structure of the control lies in the basis of a Function Link-based Wilcoxon radial basis function network (FLWRBFN). The learning rates of FLWRBFN were tuned using a hybrid Differential Evolution and PSO technique. The study aimed at analyzing the dynamic and transient performance of wave power generation systems under disturbances and grid fault. The proposed algorithm was compared to the PI controller and radial basis function network (RBFN). Considering the turbine speed, line voltage, dc link voltage and grid side real power, the FLWRBFN achieved a lower overshoot and faster settling time for all aspects for both the dynamic and transient responses. It was also observed that the FLWRBFN with PSO-DE also produced the overall best convergence rate. PSO was applied to a DFIG based dish Stirling system in for maximum power point tracking and regulation of receiver temperature in [65]. A control scheme based on average pressure control and coordinated torque was proposed. This proposed model required only four parameters to be optimized, compared to twenty in existing control schemes. These four parameters were optimized using PSO. The results showed that as irradiance varied, the proposed control scheme was superior in providing reactive power to the grid as well as achieving temperature regulation.

2.3.2. Bacteria Foraging Optimization Algorithm

Bacteria foraging optimization (BFO) is a MOT inspired by the conduct of the E. coli bacterium which is present in the human intestine. The motion of the bacterium is dependent on the motion of the flagella which is attached to the bacterium (as a tail). The bacterium either tumbles (changes direction with minimal displacement) or swims; if the flagella rotates clockwise then the bacterium tumbles and if the flagella rotate anticlockwise then the bacterium swims. There are four steps involved in BFO. These are chemotaxis, swarming, reproduction, and elimination-dispersal [66], [67], [68], [69]. One of the strong merits of the conventional BFO is that it does not easily get trapped in the local minima [70]. To the authors best knowledge, there has been no established demerits of BFO. This does not mean that none exist, but rather points to the lack of application of the algorithm. Considering the application of BFO to the DFIG, the authors in [71] make use of BFO to dampen low frequency oscillations. Both the GSC and RSC were considered. A PI based damping controller was added to the RSC. The control method considered the stator active power, stator voltage magnitude, dc link voltage and GSC reactive power consumption. The entire control algorithm made use of seven PI controllers and used BFO to optimize the gains of each controller. The objective function used was based on the damping ratio of the eigenvalues of the system, meaning that the control method makes use of the differential algebraic equations of the system. The PI controllers were optimized at three different wind speeds (7m/s, 8m/s and 8.5 m/s). For every wind speed, the DFIG is operated at the synchronous speed, as well as speeds above and below this speed. A comparison was made with and without a damping controller. When the controllers were optimized for 8m/s, there exists stability for the synchronous and super synchronous modes, but not sub synchronous. An identical result was seen at 8.5 m/s. However, when the parameters are optimized at 7 m/s, there exists stability across all three operating regions. With regards to efficacy of damping controller, there were scenarios where the controller proved to be effective and scenarios where it failed to produce a superior result. The control method focused more on the effects of using a damping controller and not on the optimization capabilities of the BFO. There was no comparison between optimization using BFO and optimization using another method, like Ziegler Nichols. As a result, the superiority of BFO in this application could not be validated.

PI controllers were optimized using BFO in [49]. The PI controllers were responsible for regulating the rotor currents and dc link voltage of a DFIG. The fitness function used was ITAE. The method was tested under step references and random rotor speed changes and compared to PSO and Genetic Algorithm (GA) based PI controller tuning. Considering the dc link voltage under step reference, the rise time of all three algorithms seem similar. BFO produced the best overshoot, which was 25.33% and 132.8% superior to GA and PSO respectively. The corresponds to a superiority of 17.5% and 253.21% respectively for the settling time. Considering the direct rotor current, BFO yielded the best rise time, marginally beating PSO. However, the PSO optimized controller produced large fluctuations in response. The overshoot superiority was once again exhibited by BFO, this time the result being 54.54% and a huge 834.62% superior to GA and PSO respectively. This corresponds to a superiority of 71.82% and 27.96% respectively for the settling time. For the quadrature rotor current, BFO presented the poorest rise time, but the best overshoot and settling time. The overshoot was superior to GA and PSO by 74.79% and 84.83% respectively and the settling time was superior to GA and PSO by 31.3% and 158.81% respectively. BFOA method also produced a good THD level of the grid current, but this was not compared to the THD produced by the other two control methods.

2.3.3. Grey Wolf Optimization

Grey wolf optimization (GWO) is a MOT which is based on the behaviour of the grey wolf. Proposed by Mirjalili, this algorithm is based on the hunting and democratic conduct of grey wolves [72], [73]. The social hierarchy of a pack of grey wolves is such that the alpha wolf is the highest ranked wolf and serves as the leader of the pack. The beta wolf is responsible for relaying information from the alpha wolf to the other wolves and assists the alpha wolf in decision making. The delta wolf is the third ranked wolf, and their duties include finding paths, killing, and taking care of the other wolves. Finally, all other wolves are classified as omega wolves and obey the rules of the wolves above them [72]. The conventional GWO has the merit of strong local search capabilities [73]. However, its demerits are poor global search capabilities and slow convergence at the latter part of the algorithm [72] - [74] .The authors in [73] proposed the use of the quantum behaved search mechanism to enhance the ability of the GWO algorithm to prevent entrapment in the local optimum. It does so by updating the position of each wolf using a probability density function based on Monte Carlo stochastic simulation. This is based on quantum uncertainty. The proposed algorithm was tested on various benchmark functions, both single and multipeak functions. The proposed algorithm was compared to other improved GWO algorithms, as well as the conventional GWO algorithm. For four of the five benchmark functions, the proposed algorithm produced an equivalent best result to one of the modified algorithms. For the remainder benchmark function, the proposed algorithm exhibited dominancy in accuracy and stability over the next best by incomparable margins (above $5e^5\%$). The convergence rate of the proposed algorithm was shown for each benchmark function. However, no comparison of convergence rate between the proposed algorithm and the conventional algorithm was provided. Hence, there exist the probability that in the process of enhancing the algorithms global search ability, the convergence rate of the algorithm was compromised.

The behaviour of cats is utilized to modify the GWO algorithm in [74]. In an idle state, cats either seek or track prey. The seeking and tracking behaviour is integrated into the social behaviour of the grey wolf. In tracking mode, the position of a particular wolf is updated in a manner similar to that of PSO. The updated position is dependent on the current position and the updated velocity. The updated velocity is calculated using the current velocity, the current position of the respective wolf, the current position of the best wolf, and two random numbers in the range [0,1]. The seeking mode utilizes a stochastic change in the dimension of each wolf in order to improve the randomness of the algorithm. Reference [74] combines both the tracking and seeking modes and applied this to the GWO algorithm. The Tracking Grey Wolf Optimization (TGWO), Seeking Grey Wolf Optimization (SGWO) and Tracking-Seeking Grey Wolf Optimization (TS-GWO) algorithms were applied to various benchmark functions and compared to numerous swarm-based MOT, including the conventional GWO. Considering the 30-Dimensional unimodal and multimodal functions, the TGWO and TSGWO combined produced the best average value and standard deviation for 15 of the 16 functions. Only for one of the functions, the SSA yielded the best result. Considering the 100-Dimensional unimodal and multimodal functions, TGWO and TSGWO combined produced the best average value for all the functions. Considering the standard deviation, there existed a couple of scenarios whereby the WOA proved to be dominant. A similar scenario is observed for the fixed dimension multimodal functions, this time the MFO displaying superiority (in both accuracy and stability) for one of the functions. For another function, despite being inferior to TGWO and TSGWO in terms of accuracy, the ALO exhibited stronger stability. The proposed algorithms were also compared to various GWO

hybrid algorithms and once again tested on 30-Dimensional and 100-Dimensional unimodal and multimodal functions. Considering the 30-Dimensional unimodal and multimodal functions, TGWO and TSGWO combined generated the best average value and standard deviation four 14 of the 17 functions. A near identical result is observed for the 100-Dimensional unimodal and multimodal functions, this time the proposed algorithm also being inferior in stability to another algorithm for one of the functions. The TGWO and TSGWO generated the best average value in all the fixed dimension multimodal functions but failed to display stability dominancy in 50% of the cases.

To enhance the convergence rate, a convergence factor was introduced in [72]. This convergence factor modifies the way in which the coefficient vector A is estimated. This convergence factor is based on the current and maximum iteration numbers. To improve the global search capability of the algorithm and produce a strong balance between exploitation and exploration, the BFGS algorithm as well as the Levy flight technique were used. Local Diversity Measure and Global Diversity measure were used to determine if the wolves perform local or global search. The local search update is modified using the BFGS algorithm, which is based on the position of the best wolf. The proposed algorithm also makes use of a probability criterion which allows some wolves to update their position using the modified equation and some wolves to update their position using the conventional equation. The global search update is based on the use of the Levy flight technique, which is calculated using the current position of the respective wolf as well as the position of a randomly chosen wolf. The proposed algorithm was tested on a range of unimodal, multimodal, and fixed dimensional multimodal benchmark functions. The proposed algorithm was compared to various swarm-based MOT, including the conventional GWO. For the unimodal functions, the proposed algorithm outperformed all the other algorithms in terms of global search capability and convergence rate (after a maximum of 10 iterations). Considering the unimodal functions, the proposed algorithm displayed the same results that were seen for the unimodal functions, except for one function. In this function, the Imperialist Competitive Algorithm based PSO produced the best global search capability, with the Ant Lion Optimization producing the best convergence rate up to 90% of the maximum number of iterations. Nevertheless, the proposed algorithm produced a better global search capability to the conventional GWO, but the former was inferior to the latter in terms on convergence rate up until 99% of the maximum number of iterations. For the fixed dimensional multimodal functions, the proposed algorithm produces competitive results in terms of global search capability but was inferior to the conventional GWO. The proposed algorithm, however, produced the best convergence rate (after a maximum of 75 iterations). However, for the convergence curves, only limited data was available. It was also observed that throughout the duration of the investigation, there existed various scenarios whereby the proposed algorithm was inferior to other algorithms in terms of stability.

Considering the application of GWO to the DFIG, the authors in [75] applied GWO to fractional order PID (FOPID) control. This is due to the improved closed loop performance and enhanced disturbance rejection capabilities of the FOPID controllers. The FOPID controller makes use of two additional parameters, which ensures that the performance does not degrade if the rotor resistance varies. Since the GWO requires initial solutions to be generated, this method initially tunes the three parameters of the fractional order PID controllers using the Ziegler Nichols method and then proceeds to apply GWO to these parameters. The method was

compared to PSO-PID and BFO-PID and was shown to be superior to both these methods with regards to the settling time, as well as the rise time and percentage overshoot. An identical result was observed when the rotor and stator parameters were varied by 25%. The proposed algorithm also produced a better disturbance rejection response than the BFO-PID but was inferior in this aspect to PSO-PID. However, the FOPID controller should have also been optimized using other MOT (such as PSO and BFO), so as to provide an accurate evaluation of the optimization technique and controller combination. A grouped grey wolf optimization strategy is presented in [76] and applied to the optimization of PI controllers for control of the DFIG. The proposed algorithm splits the wolves into two groups. These are the cooperative hunting group, and random scout group. The random scout group searches unknown territory, much like the scout bees in ABC. This is to enhance the exploration capability of the algorithm. In the cooperative hunting group, the number of beta and delta wolves increase to two and three respectively. This is to enhance the exploitation capability of the algorithm. The proposed algorithm was compared to that of GA, PSO, MFO and the conventional GWO. The results showed that optimized of PI controllers via the proposed method yielded a significant reduction in steady state ripple of both active and reactive powers. This result hold true, even when tested under the case of a 30% drop in the grid voltage.

2.3.4. Artificial Bee Colony

The artificial bee colony (ABC) algorithm is a MOT inspired by on the hunting behaviour of honeybees. Created by Karaboga, this algorithm divides the hunting bees into three types; employed bees, onlooker bees and scout bees. An employed bee is a bee which has found an exploitable food source. The onlooker bee awaits the information obtained by the employed bee to decide which food source to visit. The scout bee randomly searches for food on its own [77], [78], [79]. For every food source, there exists one employed bee. When the food source of an employed bee becomes depleted (either by that of another employed bee, or an onlooker bee), it develops into a scout bee [78]. The information regarding the food source is communicated from the employed bees to the onlookers via dancing. The onlookers observe the dance done by the employed bees to choose the best quality food source [79]. The hunting done by the bees is known as foraging and is defined by four characteristics [79]:

Positive feedback: This refers to a proportional increase in onlookers visiting rich food sources

Negative feedback: This refers to the bees eventually ceasing to visit the areas where there exist poor food sources

Fluctuations: This refers to the random search behaviour by the scouts

Multiple interactions: This refers to the exchange of information that exists between the employed bees and onlookers

The conventional ABC algorithm has the merit of a strong global search capability [80] but suffers the demerit of a slow convergence [81], [82], [83]. The authors in [84] proposed a new method to update the position of a bee. The modified position update equation is a function of the global best, the position of a random bee and a random number in the range [-1,1]. The proposed method was applied to various benchmark knapsack problems and compared to the conventional ABC algorithm. Considering the average value, the proposed algorithm produced a superior result for three of the problems. The greatest difference being 3.11% and the smallest difference being 0.02%. For the remainder functions, the proposed technique was equivalent to the conventional algorithm. It was also observed that in numerous cases, the proposed algorithm exhibited a superior stability. Concerning the convergence rate, the proposed algorithm yielded superior results in four cases, the smallest superiority being

61.68%. The authors in [82] propose a new method to update the position of the various bees. This is based on the use of the bee's current position, two cumulative fitness values of all the bees, the position of the best bee and two random numbers in the range [0,1]. The method of position updating does not apply to the worst employed bee. The position of the worst employed bee is updated using the current and previous position of the best bee, an integer which is either 0 or 1 and two random numbers in the range [0,1]. The position of the onlooker bee is updated using the cumulative fitness of all the bees, the total number of bees and the position of any onlooker bee whose position is superior to that of the bee being updated. The proposed method was applied to various benchmark functions, as well as two industrial problems. The proposed method was compared to various swarm-based MOT including the conventional ABC algorithm. However, the results of the experiment were inconclusive hence the proposed method could not be validated.

The authors in [83] made use of two unique equations for updating the position of the employed and onlooker bees. The position update of the employed bee is based on the current position of the bee and the position of a random bee. The position update of the onlooker bee is based on the current position of the bee, the position of the best bee and the position of three bees chosen at random. The proposed algorithm was tested on a synthetic web service selection problem and compared to various swarm-based MOT, including the conventional ABC algorithm. The proposed method proved to have a superior convergence rate to the other algorithms, after 10 iterations. In terms of reliability, the proposed technique exhibits dominancy after completion of 30% of the maximum number of iterations.

In [85], ABC is utilized in the control of the DFIG. The authors in [85] used ABC to optimize the parameters of the field-oriented control PI controllers. Two cases are presented in this paper. In the first case, the ABC algorithm was applied to only the RSC using a fitness function based on the stator active power, stator voltage and current regulation errors. Three weighting factors are used in this fitness function, which were chosen by the ABC algorithm. In the second case, the ABC algorithm was applied to both the RSC and GSC. The fitness function used was based on the stator active power, stator voltage, current regulation, grid current and dc link voltage errors. This time, five weighting factors were used and chosen by the ABC algorithm. In total, 10 PI controller gains were optimized. The control method was applied to a 9 MW DFIG and compared to the GWO method of PI gain optimization, as well as the traditional PI controller optimization method. When only the RSC gains were considered, it is observed that ABC yielded the best overshoot value, being superior to GWO and the advisory method by more than $2e^3$% and $3e^3$% respectively. A similar observation is made for the settling time, this time the superiority being 0.52% and 0.78% respectively. Considering the rise time, the ABC exhibited dominancy to the other techniques by a magnitude greater than 100% When the RSC and GSC gains were considered, it was seen that ABC yielded the best overshoot value, being superior to GWO and the advisory method by 97.7% and 169.73% respectively. A similar observation is made for the settling time, this time the superiority being 0.21% and 0.26% respectively. Considering the rise time, the ABC exhibited dominancy to GWO by 0.32%, and to the supervisory method by more than 100%.

2.3.5. Whale Optimization Algorithm

Whale optimization algorithm (WOA) is inspired by the hunting tactic of the humpback whale. The hunting strategy of the humpback whale is separated into three parts: searching, encircling and bubble-net attacking [86],

[87], [88]. During searching, the humpback whales exchange information about prey to each other. This is to ensure that all the whales stay close to the prey. Although the WOA has the merit of being able to evade the local optima hence obtain the global solution [89], it suffers the demerits of a slow convergence speed and low accuracy [90]. The authors in [91] proposed the use of a new method to update the position of the whales. This is based on the golden sine operator and makes use of the current position of the whale, as well as two random numbers lying in the range [0,1]. The proposed algorithm was tested on a range of unimodal, multimodal, and combined functions and compared to various other MOT, including the conventional WOA. Considering the unimodal functions, the proposed algorithm generated the best average value and standard deviation in 5 of the 7 cases. For the other two cases, the algorithm is inferior to PSO in both accuracy and stability. For the multimodal functions, the proposed algorithm displayed inferiority in both average value and standard deviation to the Firefly Algorithm for two functions. For both functions, for both the average value and standard deviation, this inferiority was approximately 100%. For the fixed dimensional functions, the proposed technique did not do well, being inferior to various algorithms in majority of the cases. This was in terms of both accuracy and stability. In cases where the proposed technique produced the dominant average value, this occurred after a maximum of 22% of the total number of iterations occurred.

The authors in [92] also proposed a new method to update the position of the whales. This is done via a chaotic map and nonlinear inertial weights. The method is complex and involves a significant number of numerical computations. The proposed algorithm was tested on various benchmark functions, at 100, 500 and 1000 dimensions. For all the investigated scenarios, the proposed techniques exhibited superior performance in both accuracy and stability. Considering convergence, for the 100 and 500 dimension sets the proposed algorithm only displayed clear dominancy after completion of 70% and 80% respectively of the maximum number of iterations. For the 1000-dimension set, this value reduces drastically to 30%, indicating the efficacy of the proposed solution when attempting to optimize large scale global problems.

An improved Bernoulli shift map was introduced in [93] to initialize the population of whales so to enhance the algorithm global search ability. A modified Levy flight-based position update equation is also proposed to enhance the global search capabilities of the algorithm. The method also optimizes the factor of convergence (a) to enhance the algorithm rate of convergence. The modified convergence factor is based on the value of the current and maximum iteration numbers, the best and worst fitness of that particular whale thus far and a random number in the range [1,2]. The method proved to improve the algorithms search accuracy and rate of convergence. The proposed algorithm was tested on numerous benchmark functions and was compared to various MOT, including the conventional WOA. When considering search accuracy, it was observed that the proposed algorithm was inferior to the Enhanced advanced guided differential evolution algorithm and SHADE algorithm in only three of the twenty functions. In scenarios where the proposed algorithm produced the best average value, this occurred after a maximum of 40% of the total number of iterations has occurred.

2.3.6. Crow search algorithm

Proposed by Askarzadeh, the idea behind the crow search algorithm (CSA) rests on the hiding of food of crows. Crows are highly intelligent birds, who can remember faces and remember the location of their stored food. The

most important aspect of this algorithm is that when a crow attempts to retrieve its stored food, another crow may follow it and steal the food [45], [94]. The merit of the conventional CSA is that it is a flexible algorithm which requires knowledge of only a few parameters [45]. However, the demerits of this algorithm are a slow rate of convergence and being easily entrapped in the local optima [94]. The authors in [94] proposed various modifications to the conventional CSA. In the first modification, the existing position update equation is multiplied by a weighting factor which is based on the current iteration and maximum iteration numbers. In the second modification, the initial positions are generated via a spiral search. In the third modification, another position update equation based on a Gaussian mutation, is proposed. The first modification is proposed to improve the algorithms rate of convergence, whereas the second and third modification is proposed to enhance the algorithms global search ability and prevent it from being entrapped into local optima. The proposed algorithm was tested on various fixed dimensional, multimodal, and unimodal benchmark functions and compared to various other swarm-based MOT, including the conventional CSA.

The proposed algorithm was evaluated at 30, 50 and 100 dimensions. For the 30 dimensional set unimodal functions, the proposed algorithm yielded the best result (in terms of both accuracy and stability in 6 of the 7 functions). Only for one function was the proposed algorithm inferior to the Harris Hawks Optimization technique. The inferiority to the best result, in terms of average value and standard deviation, was 93.56% and 94.69% respectively. For the 30-dimensional set multimodal functions, a similar result is observed. This time, the proposed technique being inferior to GA, with this inferiority being at 100% for both the average value and standard deviation. A similar trend is displayed for the 50 and 100 dimensional sets, with the proposed algorithm only being inferior in one instance of each case. There is, however, a reduction in inferiority for the 50- and 100-dimensional set multimodal functions. This points to the possibility of the algorithm exhibiting dominancy when optimizing large dimensional problems. Considering the fixed dimensional functions, the proposed algorithm yielded the best result in 8 of the 10 cases, being inferior to the GWO algorithm and the Butterfly Optimization Algorithm. The convergence rate of the proposed algorithm was only displayed for three of the unimodal and multimodal functions and two fixed dimension functions. Despite the superior average value and standard deviation of the proposed algorithm, it was observed that in various instances, the proposed technique required more than 100 iterations to exhibit dominancy. In some cases, more than 1500 iterations were required. This points to a slow convergence and should be modified for the algorithm to be considered competitive.

The CSA was applied to the PI control of a DFIG in [95]. Only the RSC was considered, and three PI controllers were tuned. The fitness function used was ISAE. The control method was tested at both fixed speeds and variable speeds and compared to the conventional method of PI controller tuning, as well as the Genetic Algorithm (GA). Considering the fixed speed operation, it is noticed that the GA produced the best stator active power percentage overshoot and rise time. CSA produced a better percentage overshoot to the conventional method, but an inferior rise time. For the stator reactive power, once again GA produced the best percentage overshoot followed by CSA and then the conventional method. The rise time of the conventional method was the best, followed by CSA and then GA. For the dc link voltage, the conventional method produced the best percentage overshoot followed by GA and then CSA. The conventional method also produced the best rise time, with CSA producing the worst in this aspect. Considering the variable speed operation, it is observed that the GA and CSA jointly produced the best stator active power percentage overshoot, with the former producing the best rise time and the latter producing

the worst. For the stator reactive power, GA produced the best percentage overshoot, followed by CSA. The conventional method produced the best rise time, with the CSA producing the worst result in this aspect. For the dc link voltage, GA produced the best percentage overshoot and CSA the worst. The conventional method produced the best rise time, followed by CSA.

2.3.7. Bat Algorithm

Developed by Xin-She Yang in 2010, the bat algorithm (BA) is inspired by and based on the use of echolocation by microbats. Bats use echolocation to perform various functions, such as locating prey, avoided obstacles, and finding other bats. Echolocation varies greatly, and depends on factors such as frequency, wavelength, loudness, and rate of sonic pulses. The bat algorithm uses a few assumptions of the echolocation used by bats [96], [97]. The first assumption is that every bat utilizes echolocation to determine distance and are able to distinguish between prey and objects. The second assumption is that bats fly randomly at a certain velocity and are also able to vary the wavelength of pulses as well as the pulse rate. The final assumption is that the loudness changes between a specified maximum and minimum.

Although the BA has the merit of a fast convergence rate [98], it suffers the demerits of a poor search accuracy and being easily trapped in local minima [97]. The authors in [97] proposed five unique factors of convergence to enhance the global search capability of the algorithm. These convergence factors are of cosine form, sine form, tangent form, power function form and exponential form. All five convergence factors are based on the current and maximum iteration numbers. To enhance the accuracy of the local search, a Gaussian function is introduced. Furthermore, to improve the local search accuracy of the algorithm, a technique based on the enclosing behaviour of the WOA, and sine cosine algorithm is applied to the bat algorithm. This is based on the current and maximum iteration numbers, a randomized number in the range [0,2], a randomized number which lies in the range [0,2π] and the values of A and D obtainable from the WOA. To assess the global search capability of the proposed algorithms, the five convergence factors were applied to various fixed dimension, multimodal and unimodal benchmark functions and compared to the conventional BA. It was observed that for all the unimodal and multimodal functions, the proposed convergence factors produced an equal performance which was superior to that of the conventional BA. For the fixed dimension functions, the proposed algorithms were again superior to the conventional bat algorithm and although they were highly similar to each other in performance, there existed a minute difference. To assess the accuracy of the proposed algorithms, seven datasets were used. The proposed algorithms were once again compared to the conventional BA. On the iris dataset, the power function form produced the best result. On the wine dataset, the tan form was superior. On the BUPA dataset, the exponential form was placed first. On the seed dataset, the sine form produced the best accuracy.

Considering the heart stat log dataset, the Gaussian function produced the best result. On the WDBC and cancer datasets, the exponential form proved to be the optimal algorithm. The authors in [99] integrate the artificial ABC algorithm into the BA to enhance the local search accuracy of the latter. In this method, the bat's position is updated as usually done. Then, a randomized number is created. If the created randomized number is greater than the value of the pulse emission rate, the position of the bat is further modified again. The proposed method was employed to optimize the path of an automatic ariel vehicle and compared to various other MOT, including the conventional BA. The optimization results obtained were superior to that of the conventional BA, as well as

various other MOT. However, the algorithms rate of convergence is 50% poorer than the conventional BA. The scholars in [100] attempted to enhance the local search capability of the algorithm, as well as prevent premature convergence. The proposed method comprises of the application of various strategies to the conventional bat algorithm. These are the iterative local search, non-dominant, balance and stochastic inertia weight strategies. Stochastic inertia weight strategy applies a weighting factor to the velocity update equation. This is to enhance the algorithms rate of convergence as well as improve accuracy. The iterative local search strategy applies a specific condition in order to maximize the probability of obtaining the global best. The balance strategy attempts to provide a sense of balance between the global and local search. Since it is impossible to optimize various parameters simultaneously, the non-dominant sorting strategy gives precedence to the solution with the best fitness function. The proposed algorithm is employed on the optimal distribution of flexible fault current limiters and applied to the revised IEEE 33-BUS distribution systems with distributed generation and IEEE 30-BUS benchmark system. The proposed method produced optimal configuration of the system and displayed an improved accuracy when compared to a non-dominated sorting genetic algorithm, as well as a Multi Objective Particle Swarm Optimization which are shown in [101] and [102] respectively. However, the algorithm was not compared to the conventional BA.

A hybrid SMC and BA was used to control a DFIG in [103],. The control method made use of rotor current control to provide stator power control. The fitness function used was the mean square error. The rotor speed was held constant throughout the experiment, but the stator active power reference was stepped up. The stator reactive power was constant at 0. The proposed hybrid controller was compared to the conventional sliding mode controller and conventional PI controller. For the stator active power, stator reactive power and dc link voltage, the proposed controller produced a superior steady state ripple, and a competitive dynamic response. With regards to the percentage overshoot for the stator active power, the proposed controller was superior to the conventional PI controller but came second to the conventional sliding mode controller. This inferiority was calculated to be equal to 36%. It was also observed that there existed a minor unbalance in the stator current waveforms. However, the stator current waveforms for the conventional sliding mode controller and conventional PI controller were not provided, hence the superior or inferior quality of the stator current waveforms from the proposed controller could not be validated.

2.3.8. Squirrel Search Algorithm

Proposed by Jain et al in 2018, squirrel search algorithm (SSA) is based on the method of movement and scavenging conduct of flying squirrels. The squirrels usually glide into trees, where they feed and collect nuts. The squirrel search algorithm is based on a few assumptions [104], [105]. The first assumption is that there exists one tree for every squirrel. The second assumption is that there exits one hickory nut tree, a few acorn trees and the rest are normal trees. Hickory nut trees are the best food supply, and the acorn trees are the second-best food supplies. The normal trees are said to contain no food. The number of acorn trees is usually taken as 3. The final assumption is that each squirrel individually attempts to locate food and makes use of the food supplies that are available.

Although the conventional SSA has the merit of a strong stability, it suffers the demerits of a low search accuracy and being easily trapped in the local optima [104]. The authors in [104] attempted to overcome this disadvantage

by incorporating the reproductive behaviour of the invasive weed optimization algorithm into the conventional SSA. The method generates squirrel offspring via the Gaussian distribution and randomly places these offspring across the search space. The number of offspring produced is proportional to the cost function value of each squirrel and varies linearly with such. Secondly, an adaptive step size strategy is implemented to balance the algorithm exploitation and exploration capability. The proposed algorithm was applied to various fixed dimension, multimodal and unimodal functions and compared to other swarm-based MOT, including the conventional SSA. Of the six unimodal functions, eight multimodal functions and eight fixed dimensional functions, the proposed algorithm only to generate the best average value in one of each set of functions. This inferiority to the best value is 9.23%, 49.85% and 4.92% for the unimodal, multimodal and fixed dimension function respectively. However, considering the standard deviation, there existed various scenarios whereby the proposed algorithm did not yield the best value, in some cases producing the worse result. The points to a lack of stability of the proposed technique. Considering the convergence, it was observed that the proposed algorithm required 75% of the maximum number of iterations to exhibit superiority. This amounted to approximately 750 iterations, and therefore indicates a poor convergence.

The scholars in [106] applied the same improved SSA in [104] to the maximum likelihood method for array signal processing-based direction of arrival. The method was compared to various swarm -based and evolutionary based MOT, including the conventional SSA. Compared to these techniques, the improved SSA displayed a faster convergence speed, better search accuracy and reduced computational complexity.

2.3.9. Moth flame optimization

Proposed by Miralji in 2015, the moth flame optimization (MFO) algorithm is based on the technique of navigation used by moths. This method of navigation is known as transverse orientation. To ensure a straight flight path, moths maintain an angle of fixed nature with respect to the moon. They are, however, severely disturbed by artificial light. Moths are seen to spiral towards artificial light and eventually latch onto the light [107], [108] . The MFO has the merit of having a robust selection capability [109], with its demerits being a slow convergence rate and being easily entrapped in the local optima [108], [109], [110]. The authors in [109] attempted to overcome the demerit of being easily trapped in local minima by modifying the update formulas of both the moth and flames. It does so in three ways; the use of a levy flight equation, the use of a weighting factor and via a descending curvilinear strategy. The levy flight equation is applied to the entire of (31) and (32). The weighting factor is applied the last term in these two equations. The updating of the number of flames is done via a curvilinear fashion, using an equation that relates to the total number of flames and current and maximum iteration numbers. The proposed method was applied to the subthreshold image segmentation problem and compared to various swarm-based MOT, including the conventional MFO. The results show that on average, the proposed algorithm was the most superior in terms of global search capability. Of the 96 tests done, the proposed algorithm was only inferior to the conventional MFO in seven tests. In terms of convergence, the proposed algorithm exhibited dominance after a mere 10 iterations. A convergence factor that is reduced linearly from -1 to -2 was introduced in [108]. This enhances the global searching capability of the algorithm while also increasing the convergence rate. The convergence factor is a function of the current and maximum iteration number.

To further enhance the global searching capability, the flame number update equation is modified and is a function of the total number of flames, as well as the current and maximum iteration numbers. The proposed algorithm is employed to locate the optimal placement and sizing of distributed generator units, is applied to the IEEE-69 bus radial distribution test system and compared to various swarm-based MOT which includes the conventional MFO. The proposed algorithm offers a superior performance with regards to the sizing and optimal placement of distributed generators. In terms of the convergence rate, the proposed algorithm required a maximum of 30 iteration before superiority was established, validating a strong speed of convergence. The scholars in [111] proposed a novel swarm moth flame optimization algorithm for the tuning of four PI controllers which are responsible for controlling a DFIG. When compared to other MOT, the proposed method was shown to improve maximum power point tracking and enhanced fault ride through capabilities.

2.3.10. Sailfish Optimization

Based on the group hunting behavior of the sailfish, sailfish optimization (SFO) is a model of the interaction between the sailfish and their prey, the sardine. Being the fastest animal underwater, the sailfish can reach a speed of 100km/h. They hunt sardines by driving them to the surface of the water. Their immense speed makes it difficult for sardines to escape, but sardines do have good manoeuvrability. A sailfish uses its rostrum to slash and injure a sardine, or directly touch it and destabilize it. In the sailfish algorithm, both the sailfish and sardines are critical aspects to consider [112]. The conventional SFO has the merits of a fast convergence rate and being not easily trapped in the local optima [112]. To the authors best knowledge, there has been no established demerits of the SFO algorithm. This does not mean that none exist, but rather points to the lack of application of the algorithm.

2.3.11. Cuckoo Search Algorithm

Proposed by Yang and deb, the cuckoo search algorithm (CuSA) is based on the reproductive behavior of the cuckoo bird. Cuckoos are parasitic birds, which lay their eggs in other bird's nests. To prevent their eggs from being thrown out by the host bird, the female cuckoo lays eggs which imitate the eggs of the host in terms of factors such as shape and colour. Cuckoo eggs usually hatch prior to the eggs of the host, and when this happens the cuckoo chick kicks out the host eggs to increase their share of food [113], [114]. There are three rules that govern the cuckoo search algorithm [113]. The first rule is that each cuckoo bird only lays one egg and places it in a random nest. The second rule states that the best nest (which contains the best quality eggs) has the best chance of being carried over to the next generation. The best nest is that in which the host eggs look very similar to the cuckoo eggs. The third and final rule states that the number of nests is unchangeable.

The conventional CuSA has the merit of requiring knowledge of just a few parameters [115] but suffers the demerits of a stagnant rate of convergence [116] and being easily trapped in the local optima [115], [116]. The authors in [115] incorporated the method of differential evolution in the algorithm. In this proposed modification, the position update of a particular cuckoo is based on the position of three random cuckoos. The proposed algorithm was implemented on monopulse antenna problems in 20-element arrays, 40-element arrays, and a fixed number of subarrays. The proposed algorithm was compared to various MOT, including the conventional CuSA. In the 20-element array, five subarrays were tested. It was noticed that the proposed algorithm produced the best global search capability in all five subarrays. In the 40-element arrays, 10 subarrays were tested. It was observed

that the proposed algorithm produced a superior global search capability in all 10 subarrays. The same is true for the five subarrays tested for the fixed number of subarrays. This clearly confirms the superiority of the proposed algorithm.

The scholars in [116] made use of a coefficient function to change the step size. The proposed algorithm also makes use of a logistic map of each dimension to initialize he location of the host nest and update the position of the host nest beyond the boundary. The proposed algorithm was tested on fifteen benchmark functions and compared to the conventional CuSA. It was observed that the proposed algorithm produced a superior global search capability in all the tested functions. With regards to the convergence rate, the proposed algorithm proved to be superior in all fifteen functions, exhibiting dominancy upon the completion of a maximum of 20% of the total number of iterations. CuSA was applied to both a PI controller and FOPID controller to control a DFIG in [117]. The method applied this control to the pitch controller, RSC and GSC. The method also made use of all the common performance indices (ITAE, IAE, ISE, and ITAE) and combined all of these to form the objective function to be used. The objective function was a sum of these common performance indices, with each index multiplied by a weighting factor. The cumulative sum of the weighting factors is one, and the weighting factors were determined using the (CuSA). For the pitch controller and RSC controllers, the PI controller produced the best results and for the GSC controller, the FOPID controller produced the best results. However, results were only given in terms of errors derived from the use of the different performance indices. Very little graphical results were provided, and no steady state and dynamic response comparisons were provided. Furthermore, there was no comparison with other MOT. This made it difficult to validate the results provided.

2.3.12. Firefly algorithm

Developed in 2007 by Yang, the firefly algorithm (FA) is based on the behavior and patterns of flashing of fireflies [106]. This optimization algorithm uses four rules [118], [119]. The first rule states that the less bright fireflies are attracted to brighter fireflies and this attraction occurs without any regard for gender. The second rule is that the brighter a firefly appears to be, the more attractive is seems. The third rule says that the further away firefly a is from firefly b, the less attractive is appears. In the fourth and final rule, the brightest firefly is the only firefly that moves randomly. The conventional FA suffers the demerits of being easily trapped in the local optima [120], [121] and a slow convergence rate [120], [122]. The authors in [121] added the concept of velocity to the position update equation to both improve the global search capabilities and enhance the algorithm rate of convergence. The velocity factor utilizes the concept of randomness, acceleration coefficients and the position of the i^{th} firefly. The proposed algorithm was employed to the design of a digital infinite impulse response filter and compared to the conventional FA. The proposed algorithm outperformed the conventional algorithm; it produced a lower mean square error and a superior rate of convergence.

Considering a second order system with a second order filter, the proposed algorithm exhibited a minimum mean square error dominance of 6.92%, with a maximum superiority of 26.77%. This corresponded to 0.61% and 1.36% respectively for a second order system with a first order filter. The scholars in [120] proposed various modifications to the conventional FA. The randomization factor, absorption coefficient and initial attractiveness are all constants. This decreases the convergence speed of the algorithm. The first improvement is to change these values from constants to dynamic variables. The modified absorption coefficient is a function of the current

iteration number and a randomized number in the range [0,1]. The randomization factor and initial attractiveness are both functions of the current and maximum iteration numbers. In the second improvement, the influence of the global best is considered in the position update equation. In the third improvement, the search space is updated after each iteration. This is done with respect to the global best firefly and "squeezes" the fireflies to the global optimum. The proposed algorithm was applied to a highly nonlinear and multi model dispatch problem and compared to the conventional FA. Two cost functions were tested. For each cost function, various populations of fireflies were tested. These are 5, 15, 35 and 50 fireflies. For case 1, it was observed that the proposed algorithm produced a superior global search capability for all four firefly populations. For case two, the proposed algorithm is inferior to the conventional FA for the 5-firefly population. In the scenarios whereby the proposed algorithm yielded superior results, this occurred before completion of 5 iterations. Therefore, it can be concluded that the proposed algorithm exhibited a superior rate of convergence

Considering the application of the FA to the control of the DFIG, the authors in [123] made use of a second order lead lag power oscillation damper (POD). The parameters of the POD are optimized by the FA. The aim of the algorithm is to stabilize inter area oscillations in interconnected power systems by means of a POD equipped with a DFIG. The proposed robust DFIG-POD was compared to the conventional DFIG-POD. The proposed method showed a superior damping performance with regards to heavy power flows, fault location, severe faults, and varying patterns of wind. However, no comparison or evaluation of the optimization technique was provided. Reference [124] makes use of a hybrid PSO and FA for the regulation of a multi area power system's frequency. The power system contains DFIG's. Two different controllers were tested: PID controller and a cascaded PD-PI controller. To analyze the dynamic performance of the system, a 1% load disturbance was injected into the system. The fitness function used was the ITAE. The hybrid PSO/FA was applied to both the PID controller and PD-PI controllers. The proposed method modifies the conventional PSO velocity equation by replacing the acceleration constants with FA parameters. The results confirm that the latter controller surpasses the PID controller in all aspects. However, the proposed metaheuristic optimization technique was not compared to other control techniques, hence its superiority could not be validated.

2.3.13. Shuffled frog leaping algorithm

Introduced by Eusuff and Lansely, the shuffled frog leaping algorithm (SFLA) is inspired the hunting strategy of frogs. The SFLA is known to have favorable performances of both GA and PSO. In SFLA, a group of frogs are divided into groups, each group known as a memeplex. Each memeplex performs a local search. Via the process of memetic evolution, each frog evolves based on the ideas of other frogs. After a predetermined number of memetic steps has occurred, information is shuffled among the different memeplexes [125], [126]. The SFLA is divided into 4 steps, namely initialization, partition, updating and shuffling.

The conventional SFLA has the merit of a fast convergence speed [127]. However, it suffers the demerits of premature convergence to the local optima [127], [128] as well as random jumps which leads to blind searches [129]. The authors in [127] proposed a SFLA which introduces the application of a weighting factor based on chaos memory and an absolute balance group strategy. The chaos memory weighting factor aimed at improving the local and global search capabilities of the algorithm. This weighting factor is applied to the position update formula and hence used to update the worst frog's position. The weighting factor is a function of the position of

the best and worst frogs, and a randomized number in the range [0,1]. The absolute balance group strategy aims to avoid premature convergence to local optima by modifying the strategy of sorting the frogs into memeplexes. It does so by randomly sorting the frogs into memeplexes, instead of using the fitness function value of each frog. The proposed algorithm was tested using the K nearest neighbour method and compared to evolutionary and swam-based MOT, including the conventional SFLA. The results showed that the application of both the chaos memory weighting factor and absolute balance group strategy produced superior global optimization performances. The proposed algorithm was applied to nine functions and compared to various MOT, including the original SFLA. It was observed that the proposed algorithm produced the most accurate result in 7 of the 9 cases. Considering stability, the proposed algorithms displayed a 66% success rate. However, it was observed that as the number of subsets (dimensions) increased, the performance of the proposed algorithm declined by a large extent. In many cases, the proposed algorithm was seen to be inferior to the other compared techniques. The converge curve, which is a critical piece of information when evaluating algorithms, was missing. This creates doubt in the confidence of the proposed scheme.

The scholars in [129] present a modified SFLA. In this method, in each memeplex, the worst frog's position is updated using the position of the local best and local worst frogs as well and a randomized number which lies in the range [0,1]. If the fitness value of the updated position is worse than the previous position, the position of the worse frog is again updated, this time using the position of the global best instead of the local best (best in each memeplex). If there is still no improvement, the method makes use of cloning of frogs. Two types of cloning can occur. In the first type, the frogs with the best positions are cloned. In the second method, a frog is cloned at random. The position of the new frog as a result of the cloned frog is a function of the position of the cloned and worst frog and a randomized number in the range [0,1]. If this method also fails to produce a better solution, the frog in question is discarded from the memeplex and replaced with a randomized frog. The proposed method is tested using the Markov chain theory and compared to various MOT, including the conventional SFLA. It was observed that in all the tests performed, the proposed algorithm produced the best global search capability. From the results provided, it is difficult to accurately compare the convergence rate of the different techniques. However, the author claims that the proposed algorithms produce a superior convergence rate.

A new method to update the position of the worst frog is introduced in [128]. The proposed method updates the position of the worst frog using the position of the frog in the centre of each memeplex, as well as a randomized number in the range [0,1]. This simple proposed method was employed to the optimization of the path of a robot under both static and dynamic environments and fared against the conventional SFLA. Considering the static environment, the proposed algorithm yielded a search accuracy superiority of 25.28%. The proposed algorithm also achieved the task 14.18% quicker. The superiority of the proposed algorithm for the dynamic environment was 36.85% and 34.3% respectively for the search accuracy and task completion time.

2.3.14. Antlion optimization

Antlion optimization (ALO) is an algorithm inspired by the foraging behaviour of antlion larvae. Its formation lies on the basis of the relationship between the antlion and their prey, the ant. Antlions dig a hole of conical shape in the ground and hide at the pit to catch prey. Hungrier ant lions dig bigger holes which improves their chances of catching prey. The ants slide down the surface of the hole, at which point the antlion consumes it [130] , [131].

Ants move randomly. Their movement is affected by the holes dug by the ant lions. Ant lions with poorer fitness functions dig bigger holes. In each iteration, every ant can be trapped by an antlion.

The conventional ALO has the demerit of being easily trapped in local optima [132]. The authors in [132] proposed a spiral complex searching pattern to overcome this demerit. In total, eight spiral paths were applied. These are the Rose Logarithmic, Inverse and Archimedes spiral curves, as well as the Epitrochoid, Hypotrochoid, Cycloid, and Spiral based overshoot parameter setting. The proposed algorithms were applied to various unimodal, multimodal, and fixed dimension benchmarks functions and compared to the conventional ALO. Considering global search capability, superior performance for the different functions was seen to be scattered across the various proposed algorithm. It is vital to note though, that for each case, at least one proposed spiral complex searching pattern proved to be superior to the conventional ALO. Majority of the best results were obtained by the Spiral based overshoot parameter setting. It was also noticed that on average, the Spiral based overshoot parameter setting yielded the strongest convergence rate. The method proposed in [133] aims to improve the algorithm global search ability by proposing a modified ant position update equation. This considers the pheromones left behind by other ants. The proposed algorithm was applied to a bearing fault identification which is centered on multi-layer extreme learning machine (MELM). The proposed algorithm was employed to the optimization of the MELM. However, no comparison between the proposed algorithm and other MOT are provided. This means that the effectiveness of the proposed method to the said application cannot be verified.

Considering the application of ALO to the control of the DFIG, the authors in [134] make use of ALO to obtain the fractional order PI (ALO-FOPI) controller gains. The method makes use of stator flux-oriented control and considers both the RSC and GSC control. Apart from the ALO-FOPI control algorithm, two other control algorithms are tested. The first is a PI controller with an ANFIS controlled added. The required gains are optimized using the Cuckoo search algorithm (CuSA-ANFIS). In the second method, a hybrid CuSA and ALO algorithm was utilized to optimize the parameters of a fractional order PI controller (CA-FOPI). The proposed algorithms were tested at two wind speeds, names 15 m/s and 10 m/s. At both 10 m/s and 15 m/s, it was observed that ALO-FOPI and CA-FOPI produces an identical and superior stator reactive power percentage overshoot than Cu-ANFIS. For both wind speeds, all three control algorithms responded identically with regards to the dc link voltage regulation. The results presented for the stator active power were unclear hence could not be analyzed. Furthermore, to clearly compare ALO algorithm and CuSA, it would have been profitable to implement a PI and ANFIS controller which is optimized using ALO.

2.4. Summary of techniques discussed

Table 2.1 summarizes the swarm-based MOT that were discussed in this research. The summary state the merits, demerits, and application of each technique to the control of the DFIG. The development of PSO has been an excellent advancement in science. Since its inception, PSO has been used countless times to solve optimization problems. It has also paved the way for other swarm-based MOT to arise. However, theory upon which PSO stands is basic. Practically, flocks of birds and schools of fish display highly complex and intelligent behaviour, which has not been incorporated into the algorithm. PSO is one of the few swarm-based MOT which displays fast convergence, hence modification and incorporation of intelligent behaviour of fish and birds may result in PSO

once again being a superior optimization technique. A similar observation holds true for GWO. The idea of utilizing the ranking of wolves within a pack was innovative, but the equations of the optimization tool are very basic. Wolves display exceptional survival tactics, and more careful observation of their behaviour could result in alleviation of the algorithms disadvantages. The idea upon which CSA and CuSA stand are interesting, however the equations of these algorithms are not complex enough to simulate the relative behaviour.

ABC, WOA, BA and SSA are complex in structure, and account for many of the characteristics of the respective swarms. However, some key aspects are missing. For example, in a bee colony, the queen bee plays an important role in the colony. The effect of incorporating the behaviour of the queen bee should be investigated. Likewise, it may be beneficial to consider the hunting strategies of other whales so as to broaden and enhance the capabilities of WOA. SSA considers only the foraging behaviour of flying squirrels. This could be broadened to incorporate the behaviours of other types of squirrels. A similar suggestion is given to the BA, which is based on the behaviour of the microbat. BA and SSA have thus far looked promising, but they have yet to been extensively tested. To validate their capabilities, rigorous testing and application is required. The same applies for ALO, SFLA, MFO, SFO and FA. Regarding ALO, the mathematical representation of the holes dug are simple. This type of representation should be investigated thoroughly so as to ensure strong simulation of the effect of these holes on ants. It is observed that of the techniques discussed, BFOA is by far the most complex. It is evident that this technique incorporates most, if not all, of the behavioural traits of the E. Coli bacterium. This can be attributed to the large amount of literature concerning this bacterium. BFOA seems to have a lot of undiscovered potential, which should be researched.

Various comparisons between conventional techniques are presented in literature. A comparison between ABC and PSO is presented in [135]. When tested on various unimodal and multimodal benchmark functions, it is observed that both algorithms display identical characteristics in the case of unimodal functions. For multimodal functions, ABC outperforms PSO. It is also observed that ABC is more sensitive to population and dimension sizes. This opens a wide area of research, as the effects of these parameters were not investigated in current literature. Another comparison is presented in [136], where PSO, FA, ABC, CSA and GWO are fared against each other. Considering unimodal functions, GWO was superior in 6 of 7 functions. GWO also produced good results in fixed dimensional multimodal functions. However, this is not the case for standard multimodal functions. CSA produced the worst average results among all the techniques, showing the strong need for improvement. A comparison between PSO and SFLA is provided in [137]. The results show that SFLA produced the overall best convergence rate and search accuracy. However, unlike PSO, it was found that SFLA is highly sensitive to user defined parameters, in particular the number of frogs and number of memeplexes. BA and BFO are fared against each other in [138]. When applied to a wide range of benchmark function, it was observed that BFO was superior in terms of accuracy. However, BA produced a faster convergence rate.

From Table 2.1, it is observed that many of the discussed swarm-based MOT suffer the demerit of being easily trapped in the local optima. The good news, however, is that various advancements have been made to correct this. It is also seen that the demerits of some MOT have not yet been established. This does not necessarily mean that none exist, but rather points to a lack of investigation into the operating capability of the algorithm. With regards to the application of swarm-based MOT to the control of the DFIG, it is observed that PSO is the most

established in this aspect. Other techniques have been applied once or twice, but not comprehensively. However, the efficacy of PSO as an optimization tool For DFIG control is not well validated. A similar issue is observed with CuSA and FA. With regards to DFIG application, The BA and ALO show promise, but require much more rigorous testing to be validated. The CSA proved to be ineffective when applied to the DFIG, and modifications to this algorithm should be presented before considering reapplication of this algorithm to the DFIG. GWO and ABC have displayed positive results thus far, but it is evident that these techniques have room for enhancement. BFO and MFO show strong capabilities when applied to the DFIG. However, these techniques have not been extensively applied to the DFIG, hence their efficacy is yet to be validated. Lastly, it is seen that some swarm-based MOT are yet to be applied to the control of the DFIG. Examples of these are SFO, SSA and SFLA. Figure 2.4 provides a visual representation of the convergence rate, exploitation and exploration capabilities of each of the algorithms discussed. In figure 2.4, a value of 1 represents a weak capability, 2 represents an average capability, a 3 represents a strong capability. From figure 2.4, it is evident that various swarm-based MOT have a weak capability when it comes to exploration, and majority of the algorithms have an average capability when considering exploitation. Considering an equal weighting of all three factors, it is observed that the SFO provides the best overall response. Therefore, its lack of application to the control of the DFIG is an interesting and possibly promising area of research.

Table 2.2 provides a method for choice of algorithm for specific applications to the DFIG. From this Table, it is observed that should one want to attempt to optimize PI controllers for standard control of the DFIG, various algorithms may be used. In the situation where optimization of FOPID controller gains for standard control is required, the CuSA and GWO seem the best choice. For optimization of parameters of sliding mode controllers, PSO and BA should be the algorithms utilized. It is important to note, though, that the application of swarm-based MOT to the control of the DFIG has not been achieved extensively. This means that, for the objectives defined in Table II, various other swarm-based MOT (in addition to the ones presented) have the potential to yield desired results.

Table 2.1:Summary of reviewed swarm-based Metaheuristic Optimization Techniques

Swarm technique	**Merits**	**Demerits**	**Application to the DFIG**	**Significant findings regarding application to DFIG**
Particle swarm optimization	Fast convergence rate [57], [58]	• Poor accuracy • Easily trapped in local minima [57], [58], [59]	• Optimization of parameters of sliding mode controller [60] • Optimization of PI controller gains to analyze	Regarding PSO optimized sliding mode controller, emphasis was given to the efficacy of the proposed controller and not on the effectiveness

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

			small signal stability [61] • Optimization of PID controller gains for standard control [63] • Optimization of FLWRBFN for stability enhancement of a DFIG based ocean energy conversion system • Optimization of dish Stirling system incorporating DFIG for maximum power point tracking enhancement and receiver temperature regulation	of PSO with regards to parameters optimization • In the case of both small and large disturbances, PSO optimized PI controllers produce strong overshoots and damping rates • Considering standard control, PSO optimized PID controllers produce superior results in all aspects when fared against the supervisory control method • Considering PSO optimized FLWRBFN for stability enhancement, emphasis was given on the efficacy of the proposed controller and not on the effectiveness of PSO with regards to

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

				• parameter optimization • Considering optimization of dish Stirling system incorporating DFIG, emphasis was given on the efficacy of the proposed controller, and not on the effectiveness of PSO with regards to parameter optimization
Bacteria foraging optimization	Not easily trapped in local minima [70]	Not yet established	• PI control gain optimization for standard control [49] • PI controller gain optimization for damping of low frequency oscillations [71]	Considering standard control, BFO optimized PI controllers produce superior results in all aspects when compared to PSO
Grey wolf optimization	Strong local search capability [73]	• Poor global search capability Slow convergence at the latter part of the algorithm [72], [73] [74]	Optimization of FOPID controller gains for standard control [75]	When compared to PSO-PID and BFO-PID the GWO optimized FOPID controller produced a superior rise time, settling time and overshoot.

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

				However, it proved to be inferior to BFO-PID when considering disturbance rejecting capabilities.
Artificial bee colony	Strong global search capability [80]	Slow convergence rate [81], [82], [83]	Optimization of PI controller gains for standard control [85]	ABC optimized PI controllers produce superior overshoots to PSO and GWO optimized PI controllers, but an inferior dynamic response
Whale optimization algorithm	Strong global search capability [89]	• Poor accuracy Slow convergence rate [90]	Not yet established	Not yet established
Crow search algorithm	• Flexible • Requires knowledge of only a few parameters	• Easily trapped in local minima • Slow convergence rate [94]	Optimization of PI controller gains for standard control [95]	CSA optimized PI controllers produced mixed results when fared against Genetic Algorithm and the supervisory control method. CSA proved to be an unsuitable method for PI controller tuning for standard control
Bat algorithm	Fast convergence rate [98]	• Poor accuracy Easily trapped in local minima [97]	Optimization of parameters of sliding mode controller [103]	BA optimized sliding mode controller was superior to the conventional sliding mode

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

				controller and PI controller tuning with respect to rise time and settling time. However, there did exist a minor unbalance in the stator voltage waveforms
Squirrel search algorithm	Strong stability [104]	• Poor accuracy • Easily trapped in local minima [104]	Not yet established	Not yet established
Moth flame optimization	Robust selection capability [109]	• Easily entrapped in local optima • Stagnant convergence rate [108], [109], [110]	Optimization of PI controller gains for standard control [111]	When compared to various other MOT, MFO optimized PI controllers displayed enhanced maximum power point and fault ride through capabilities
Sailfish optimization algorithm	• Fast convergence rate • Not easily trapped in local minima [112]	Not yet established	Not yet established	Not yet established
Cuckoo search algorithm	Requires knowledge of only a few parameters [115]	• Slow convergence rate [116] • Easily trapped in local minima [115], [116]	• Optimization of PI controller gains for standard control Optimization of FOPID controller gains for standard control [117]	Regarding both the PI and FOPID controllers optimized using CuSA, very little analysis was provided. Critical aspects such as rise time and settling

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

				were not considered. Further, there was no comparison to optimization using other MOT
Firefly algorithm	Not yet established	• Slow convergence rate [120], [121] • Easily trapped in local minima [120], [122]	• Optimization of POD controller gains for stabilization of inter area oscillations [123] • Optimization of PI-PD controller gains for frequency regulation • Optimization of PID controller gains for frequency regulation [124]	• Considering FA optimized POD, emphasis was given on the proposed control structure, and not on the effectiveness of the optimization technique • Regarding FA optimized PI-PD and PID controllers, the PI-PD controller was superior in all aspects concerning frequency regulation.
Shuffled frog leaping algorithm	Fast convergence speed [127]	• Premature convergence to the local optima [127], [128] • Random jumps which lead to	Not yet established	Not yet established

Table 2.1: Summary of reviewed swarm-based Metaheuristic Optimization Techniques

		blind searches [129]		
Antlion optimization algorithm	Not yet established	Easily trapped in local minima [132]	Optimization of FOPI controller gains for standard control [134]	ALO optimized FOPI controller seems to be promising in terms of standard DFIG control.

Table 2.2: Choice of algorithms for specific application to DFIG

Objective	**Suggested technique/s**
PI control gain optimization for standard control	PSO, ABC, CuSA, MFO
PI controller gain optimization for damping of low frequency oscillations	PSO
Optimization of FOPI controller gains for standard control	ALO
Optimization of FOPID controller gains for standard control	CuSA, GWO
Optimization of PI controller gains to analyze small signal stability	PSO
Optimization of PID controller gains for standard control	PSO
Optimization of parameters of sliding mode controller	PSO, BA
Optimization of POD controller gains for stabilization of inter area oscillations	FA
Optimization of PID controller gains for frequency regulation	FA

2.5. Simulation-based analysis of common swarm-based MOT

In this section, a simulation-based analysis of the results of various well-known techniques are carried out. These techniques are PSO, ABC, and WOA. These algorithms are applied to three benchmark functions, at three-dimension magnitudes. To allow for a fair comparison, the number of search agents and particles were kept uniform across all three algorithms. Owing

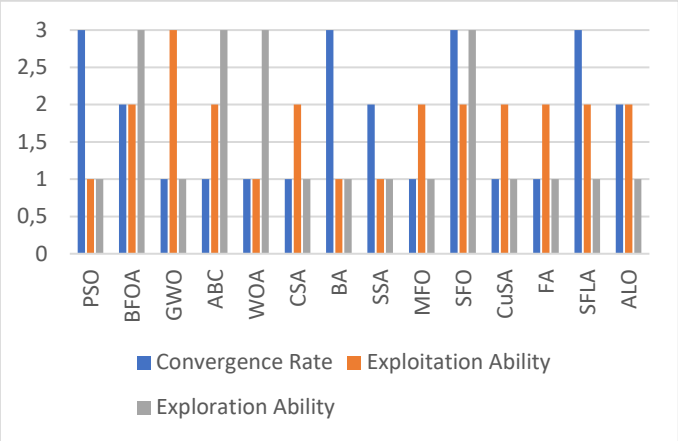


Figure 2.4: Comparison of discussed techniques in terms of convergence rate, exploitation, and exploration capabilities

to the stochastic nature of MOT, each algorithm was run 20 times. The results in table 2.3 are given in terms of average value, and standard deviation. Both, in conjunction with the convergence rate, are critical parameters in the analysis of MOT.

Considering the results of the 5-dimension set, it is observed that for F1, the ABC yields the best average value. This is succeeded by the WOA. For F2 and F3, the WOA produces the best result, with PSO producing the poorest average value. However, for these results, the ABC showed a superior standard deviation to the WOA. This indicates an inferiority of WOA to ABC in terms of performance stability. For the results of the 50-dimension set, the WOA generated the best average value with the ABC having the poorest response. For F1 and F2, PSO showed greater stability to WOA, while for F3 the WOA produced the poorest stability performance. As with the 50-dimension set, the WOA yielded a superior average value for all three functions, with the ABC producing the poorest result. Considering stability, WOA is superior for F1 and F2, but is inferior to both PSO and ABC for F3

Table 2.3: Comparison of PSO, ABC, and WOA for three benchmark functions at 5,50 and 100D

Function	Dimension		PSO	ABC	WOA
1	5	Mean	364	0.08	32.42
		Std	499	0.07	98.69
		Rank	3	1	2
2		Mean	0.54	0.17	0.14
		Std	0.23	0.04	0.14
		Rank	3	2	1
3		Mean	28.23	4.22	1.44
		Std	12.21	1.08	6.47
		Rank	3	2	1

Table 2.3: Comparison of PSO, ABC, and WOA for three benchmark functions at 5,50 and 100D

1	50	Mean	7.76E+05	30.50E+05	3.34E+05
		Std	6.03E+05	10.36E+05	6.68E+05
		Rank	2	3	1
2		Mean	1.15	6.12	0.05
		Std	0.07	0.83	0.16
		Rank	2	3	1
3		Mean	604.39	650.64	181.82
		Std	73.16	43.91	154.84
		Rank	2	3	1
1	100	Mean	2.59E+06	27.12E+06	5.58E+05
		Std	1.97E+06	3.37E+06	1.28E+06
		Rank	2	3	1
2		Mean	1.33	25.46	5.55E-18
		Std	0.18	1.69	2.48E-17
		Rank	2	3	1
3		Mean	1390.33	1644.30	645.32
		Std	120.28	36.98	174.59
		Rank	3	2	1

Figures 2.5-2.7 depict the convergence curves of each algorithm, for each test function and at each of the dimension magnitudes utilized. Considering figure 2.5, for F1, ABC yielded the best convergence rate, with PSO yielding the worst in this aspect. For F2, once again PSO generated the poorest response. ABC is superior to WOA until about 45 iterations, after which WOA outperforms the former. A similar result is seen for F3, with the different being that the superiority of WOA presenting at 75 iterations.

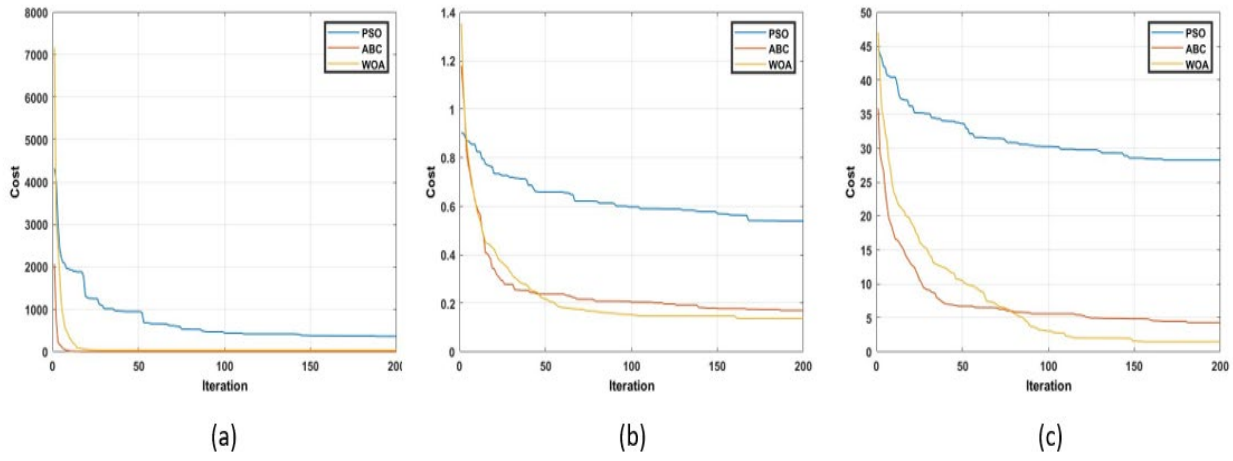


Figure 2.5: Convergence curves of algorithms at 5D for (a) F1, (b)F2 and (c)F3

Considering figure 2.6, for F1, ABC generated the poorest response. PSO proved to be superior to WOA until about 20 iterations, after which WOA exhibits dominancy. A near identical trend is observed for F2. This is not the case for F3, where WOA is superior, and ABC once again yielding the poorest result. Considering figure 2.7, for F1, ABC generated the poorest response. PSO exhibited dominancy to WOA until about 25 iterations, after which WOA proved to be superior. A similar trend is observed for F2, this time the WOA obtaining dominancy at a slightly smaller iteration count. For F3, WOA proved to be superior, and ABC once again yielding the poorest result.

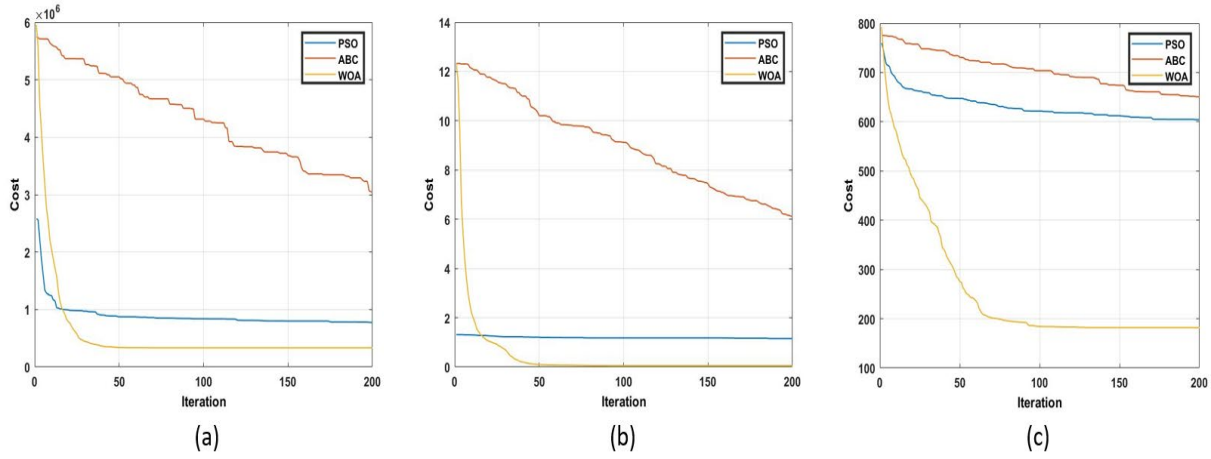


Figure 2.6: Convergence curves of algorithms at 50D for (a) F1, (b)F2 and (c)F3

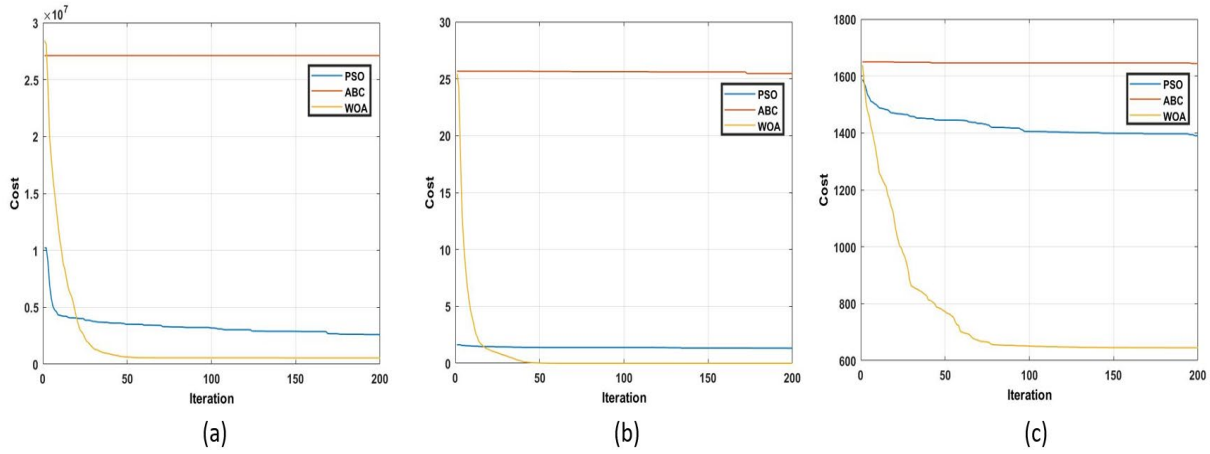


Figure 2.7: Convergence curves of algorithms at 100D for (a) F1, (b)F2 and (c)F3

2.5. Conclusion

This chapter provided a review on swarm-based Metaheuristic Optimization Techniques in terms of algorithm structure, merits, demerits, and application to the control of the DFIG. While there exists numerous swarm-based Metaheuristic Optimization Techniques, only fourteen techniques were covered in this chapter. The swarm-based techniques which featured in this chapter were PSO, BFO algorithm, ABC optimization, GWO, BA, SSA, CuSA, FA, MFO, SFO, ALO, SFLA, CSA and WOA. While the techniques provide strong performances in general, many of the conventional algorithm suffer the demerits of a poor rate of convergence and being easily entrapped in the

local minima. However, the good news is that these problems have been ameliorated in many of the algorithms. The demerits of the SFO algorithm and BFO algorithm have not yet been discovered. Based on the observable demerits of the other algorithms, this can be successfully investigated. However, when considering the application of these techniques to energy generation systems, not many examples exist. Considering application to the DFIG, only particle swarm optimization has been researched and applied somewhat thoroughly. Some algorithms like GWO, ABC optimization, BFO algorithm, CuSA, FA, ALO, MFO, CSA and BA have only been applied once or twice. Other algorithms like SFLA, SSA and SFO are yet to be applied to the DFIG. When applied to the DFIG, swarm-based MOT have produced good results. However, due to the lack of application and rigorous testing of these techniques, extensive testing is required to validate their effectiveness. Hence it would be beneficial to research and apply the algorithms, especially those which are yet to be done so.

Chapter 3 : Research Methodology

This chapter aims to clearly highlight the various metaheuristic optimization techniques utilized in this research. The swarm intelligence techniques discussed are Particle Swarm Optimization, Bat Algorithm, Gorilla Troops Optimization, African Vulture Optimization, and the Whale Optimization Algorithm. The relevant formulas required for application of each technique is provided, along with a flowchart which depicts the method of execution of the relevant algorithms. Further, a modified Whale Optimization Algorithm is proposed, and is tested on both the CEC2019 benchmark functions, as well as on a practical constrained engineering problem. Considering the application of swarm intelligence to the control of the DFIG, it will be seen later that PSO, BA, GTO, and AVOA were applied and analysed. The reason for choosing such algorithms, is to better understand the implications of algorithm advancement to the control of the DFIG. PSO is one of the earliest and most utilized swarm intelligence technique. The BA was developed some years after PSO, and based on current literature, poses some merits of PSO. GTO and AVOA are recently developed metaheuristics, which in their respective literatures, proved dominant compared to older algorithms. However, most tests on these types of algorithms are done on test functions, such as CEC benchmark functions. While these functions give a good indication of the performance of an algorithm, there still exists a need to analyse the true performance of such. This can only be done via application to practical engineering optimization problems. Hence, the aim of this particular choice of algorithms is to determine whether the continuous advancement in heuristic techniques has a positive impact on the control of the DFIG.

3.1. Particle Swarm Optimization

PSO consists of a population of particles which move at a given velocity. The velocity of every particle is updated after each iteration. These updates consider various factors. The aim of the motion of the particles in the population is the move to the most ideal solution of the problem. PSO is a simple control algorithm which has a light computational burden [53] [54] [55], [56]. Considering the real number space, a particle can be defined as a possible solution which moves through the search space of the problem. The position of a particle is a function of the particles previous position and current velocity. This can be expressed as [53], [139], [140]:

$$\bar{x}_i(t+1) = \bar{x}_i(t) + \bar{v}_i(t) \tag{3.1}$$

Where $\bar{x}_i(t+1)$ is the updated position of the particle, $\bar{x}_i(t)$ is the current position of the particle and $\bar{v}_i(t)$ is the current velocity of the particle. The current velocity of the system is defined as [53], [139], [140]:

$$\bar{v}_i(t+1) = \bar{v}_i(t) + \sigma_1 \times rand1 \times \left(\bar{p}_i - \bar{x}_i(t)\right) + \sigma_2 \times rand2 \times \left(\bar{p}_g - \bar{x}_i(t)\right) \tag{3.2}$$

Where $\bar{v}_i(t+1)$ is the updated velocity of the particle, σ_1 and σ_2 are two positive numbers, $rand1$ and $rand2$ are two randomised numbers in the range [0,1], $\bar{p}_i$ is the individual best of each particle and $\bar{p}_g$ is the global best of each particle. As shown, (3.2) comprises of three elements. The first term is based on the inertia of the particle (according to newtons first law, a body in motion tends to continue motion unless disturbed by an external force) [28]. The second term describes the particles propensity to gravitate towards its personal best. It is known as the memory component. The third term describes the particles propensity to gravitate towards to global best i.e., the best of all the particles. It is known as the social component [53], [140]. The individual best and global best are

obtained based on a fitness function which is defined by the user [53]. To ensure convergence of the particles and prevent divergence (going to infinity), selection of appropriate constants and setting limitations is essential. One of the critical limitations that needs to be present in the selection of a maximum velocity. A too large maximum velocity could result in unstable behavior of the particles and a too small velocity limits the search space and could results in the most optimal solution not being discovered. An experiment performed in [141] proved that by dynamically changing the maximum velocity, the performance of the algorithm can be enhanced. Another important limitation is the values of the acceleration constants, σ_1 and σ_2. In a study conducted in [142] and [143], it is shown that if the sum of σ_1 and σ_2 exceed 4, the particle trajectory diverges (goes to infinity). The values of the acceleration constants can be updated dynamically, in which case they are calculated based on a maximum and minimum value, as well as the current and maximum iteration numbers [60].

However, even if the acceleration constants and maximum velocity are selected correctly, there is a possibility that the particles would continue to diverge. To prevent this, there exists two methods which can be applied to (3.2). The first method is applying a constant called the constriction factor. This is applied to the entire of (3.2) and is based on the use of the two acceleration constants [139]. The second method is accomplished by applying either a fixed or dynamic value only to the inertia component of (3.2) [139]. This is termed inertia constant and usually begins at a high value and gradually decreases. Considering a dynamic inertia constant,w, the constant is calculated using an initial weight (usually 0.9), a final weight (usually 0.4), as well as the current and maximum iteration numbers [48], [139], [144]. The suitable selection of the inertia weights results in the requirement of a smaller number of iterations to obtain an acceptable solution [144].

At first, the relevant parameters (number of particles, iteration number, initial acceleration constant and initial inertial weight) are defined. Then, each particle positioned randomly throughout the search space. The next step is the evaluation of the fitness of each particle. The particle which has the lowest fitness function is determined and the position of that particle is taken as the global best. Thereafter, the position of each particle is then updated using (3.2) and the fitness function is evaluated again. This fitness function value of each particle is then compared to the previous fitness function of that particle. If the current fitness function is superior to the previous fitness function, this value replaces the old value, and the current position replaces the previous position as the new individual best. This process continues until the iterations are complete. Once this is so, the particle with the best fitness function is said to be the best solution [53], [140]. The steps to execute the PSO algorithm is depicted in figure 3.1 [145].

3.2. Bat Algorithm

Based on the use of echolocation by microbats, the position of each bat is updated as follows [96], [146]:

$$v_i(t+1) = v_i(t) + f_i(x_i(t) - x_g) \tag{3.3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3.4}$$

Where $v_i(t)$ is the current velocity of the i^{th} bat, $v_i(t+1)$ is the updated velocity of the i^{th} bat, $x_i(t)$ is the current position of the i^{th} bat, $x_i(t+1)$ is the updated position of the i^{th} bat, x_g is the global best position, f_i is the frequency of the i^{th} bat. This frequency is calculated using a specified maximum and minimum frequency and a randomized number in the range [0,1]. A randomized number between 0 and 1 is generated and compared to

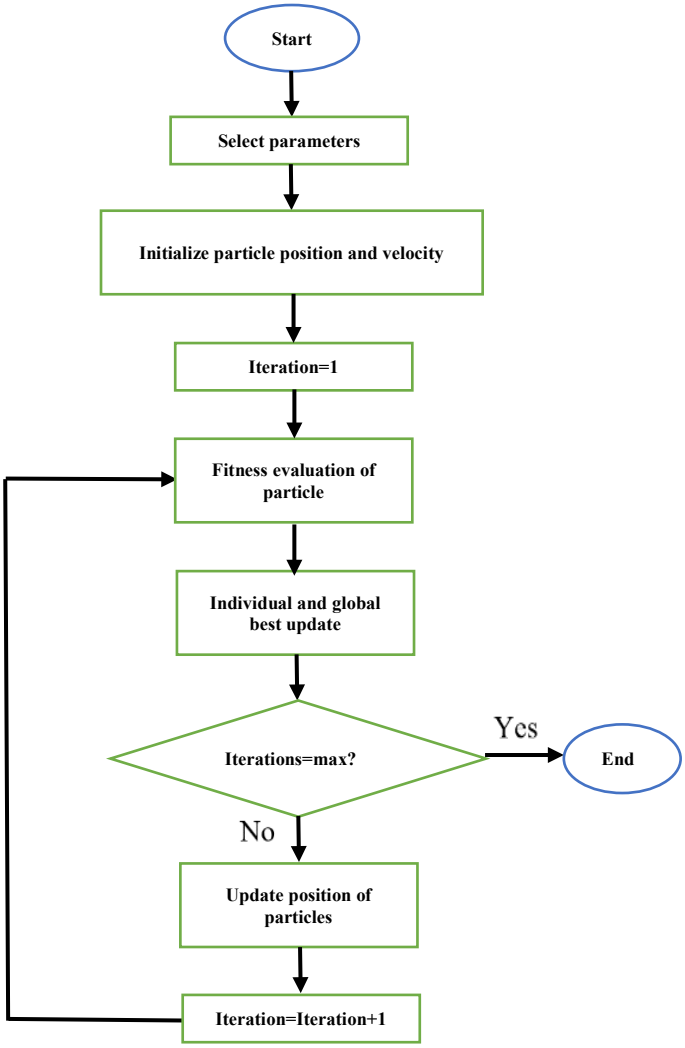


Figure 3.1: Flowchart of implementation of PSO [145]

the pulse emission rate of the i^{th} bat. The pulse emission rate is based on the current iteration number and decreases exponentially from the initial specified pulse emission rate. If the random number is greater, the position of the best bat is updated as follows [96] - [147]:

$$x_{new} = x_{old} + \in A_i{}^t \tag{3.5}$$

Where $\in$ is a randomized number in the range [0,1], $A_i{}^t$ is the current loudness of the i^{th} bat and is based on the current iteration number. Initially, the required parameters are defined. Then, each bat is assigned a random position in the search space. The fitness of each bat is then computed and the position of the bat with the best fitness value is noted. Thereafter, the position of each bat is updated according to (3.4). The fitness of each bat is computed and if the new fitness value of a bat is superior to the previous fitness value of that same bat, that bat takes on the new fitness value (hence position). Afterwards, a randomized number is defined and if this value is greater than the pulse emission rate of a specific bat, then the position of that bat is updated using (3.5). The fitness of each bat is evaluated once more. The randomized number is then compared to the loudness of each bat. If the randomized number is less than the corresponding loudness and the fitness value is superior to the previous fitness value, then the bat takes this new position. Else it remains in its previous position. This continues until all iterations

have been completed. Once this is so, the bat with the best fitness value is said to be the best solution [96]. The steps to execute the BA can be seen in figure 3.2 [148].

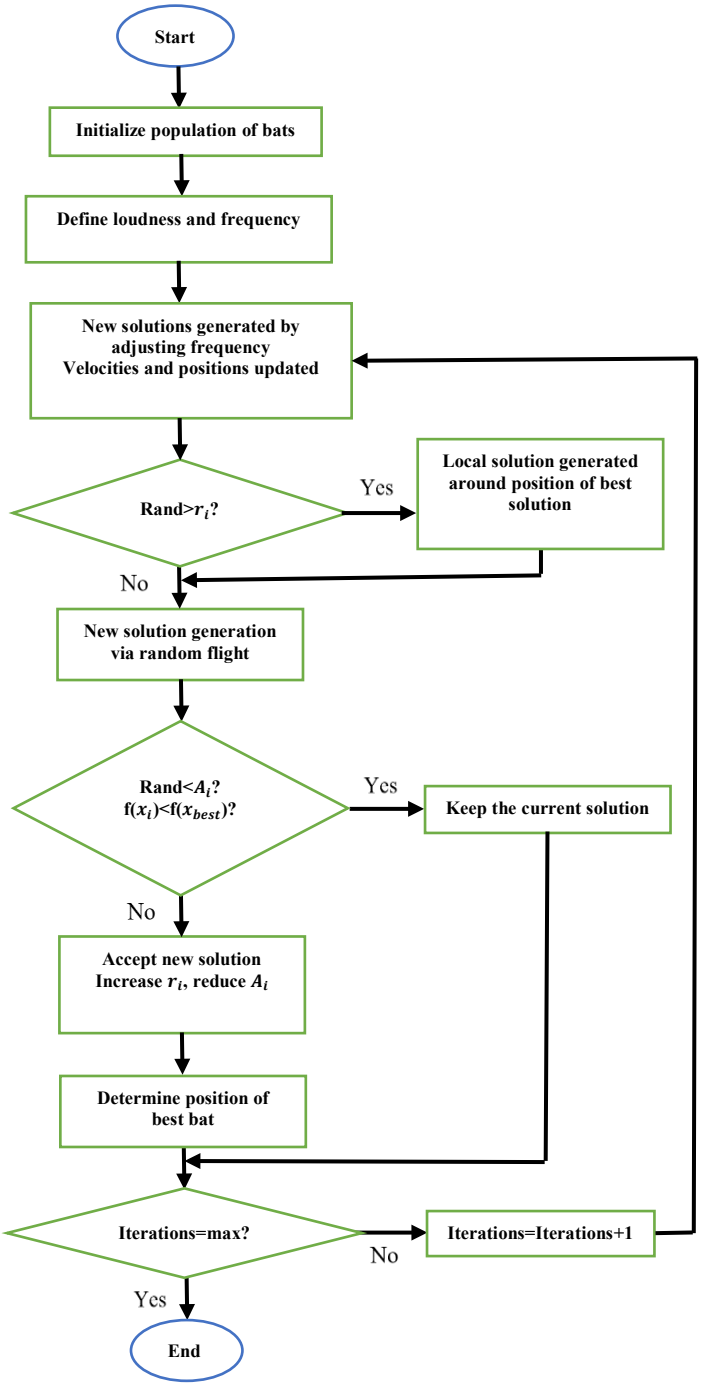


Figure 3.2: Flowchart of implementation of BA [148]

3.3. Gorilla Troops Optimization

Inspired by the social behaviour of the silverback gorilla, Gorilla Troops Optimizer seeks to mathematically represent the various behaviours of the large primate. The silverback gorilla shares various traits with other wild mammals, such as the existence of a social hierarchy, the dominance of the male among the group, and fierce

rivalry for the attraction of mating partners. However, the silverback gorillas are highly intelligent species, who share a 98 % similarity with human genetic makeup. Although the silverback gorilla often lives in troops,

occasionally a few members will move to another location. This could be either to find a new food source, to seek new mating partners, or to join other groups for better protection [149]. To simulate the exploration of a gorilla to an unknown location, the following was developed [149], [150], [151], [152]:.

$$X(t+1) = (ub - lb) \times r_1 + lb \tag{3.6}$$

Where:

- $X_{(t+1)}$ is the updates position of the gorilla
- ub is the upper bound of the search space
- lb is the lower bound of the search space
- r_1 is a random number in the domain 0 to 1

A second random number in the domain 0 to 1 is also defined. Further, a pre-set parameter p is utilized to determine if (3.6) is implemented. If the random number is less than p, then (3.6) is implemented. If not, and the random number generated is greater than or equal to 0.5, the gorillas are said to be migrating to another group. This is expressed as [149], [150], [151], [152]:

$$X(t+1) = (r_2 - C) \times X_r(t) + L \times H \tag{3.7}$$

Where:

- r_2 is a random number in the domain 0 to 1
- $X_r(t)$ is the position of a randomly selected gorilla, which exists in the troop

The parameters C, L and H are calculated as follows [149], [150], [151], [152]:

$$C = F \times \left(1 - \frac{i}{max_i}\right) \tag{3.8}$$

$$F = \cos(2 \times r_3) + 1 \tag{3.9}$$

$$L = C \times A \tag{3.10}$$

$$H = Z \times X(t) \tag{3.11}$$

$$Z = [-C, C] \tag{3.12}$$

Where:

- i is the current iteration number
- max_i is the maximum number of iterations
- r_3 is a random number in the domain 0 to 1
- A is a random number in the interval -1 to 1
- $X(t)$ is the current position of the gorilla in question

If the randomly generated number is less than 0.5, the gorilla is said to have migrated to a known location. This is modelled as [149], [150], [151], [152]:

$$X(t+1) = X(t) - L \times \left(L \times \left(X_t - X_r(t)\right) + r_4 \times \left(X(t) - X_r(t)\right)\right) \tag{3.13}$$

Where r_4 is a random number in the domain 0 to 1. To simulate exploitation, two scenarios are considered. The first is where the other male gorillas within a troop obey the orders of the leader. The second is where these gorillas engage in violent battle to assume power. A pre-set parameter W is defined. If C≥ W, the other male gorillas follow the leader. This is expressed as [149], [150], [151], [152]:

$$X(t+1) = X(t) - L \times M \times \left(X(t) - X_b(t)\right) + X(t) \tag{3.14}$$

Where $X_b(t)$ is the current position of the best placed gorilla. The value of M is obtained as follows [149], [150], [151], [152]:

$$M = \left(\left|\frac{1}{N}\sum_{k=1}^{n} X_i(t)\right|^g\right)^{\frac{1}{g}} \tag{3.15}$$

$$g = 2^L \tag{3.16}$$

Where N is the total number of gorillas within the troop. If the value of C is less than that of W, the other male gorillas engage in battle with the leader. This is modelled as [149], [150], [151], [152]:

$$X(t+1) = X_b(t) - (X_b(t) \times Q - X(t) \times Q) \times B \tag{3.17}$$

The values of Q and B are [149], [150], [151], [152]:

$$Q = 2 \times r_5 - 1 \tag{3.18}$$

$$B = \beta \times E \tag{3.19}$$

Where:

- r_5 is a random number in the domain 0 to 1
- β is a pre-set parameter

The value of E is calculated as follows [149], [150], [151], [152]:

$$E = \begin{cases} K_1, if\ r_6 \geq 0.5 \\ K_2, if\ r_6 < 0.5 \end{cases} \tag{3.20}$$

Where:

- r_6 is a random number in the domain 0 to 1
- K_1 is a random number within the search space
- K_2 is a random number which exists in the interval 0 to 1

The flowchart indicating the method of implementation of GTO is depicted in figure 3.3 [149], [150]. Evident from figure 3.3, the GTO undergoes a definite dual position change in every iteration. This will increase the random-access memory require for algorithm execution, as well as the time taken to execute the algorithm. However, in doing so, the algorithm will exhibit a higher probability of producing a more precise solution.

3.4. African Vulture Optimization Algorithm

African Vultures Optimization Algorithm is a recently developed swarm intelligence optimization technique. Inspired by the various African vulture species, this algorithm aims to mathematically model the scavenging behaviour of these elusive birds. The algorithm is split into five distinct phases. In the first phase, the total number

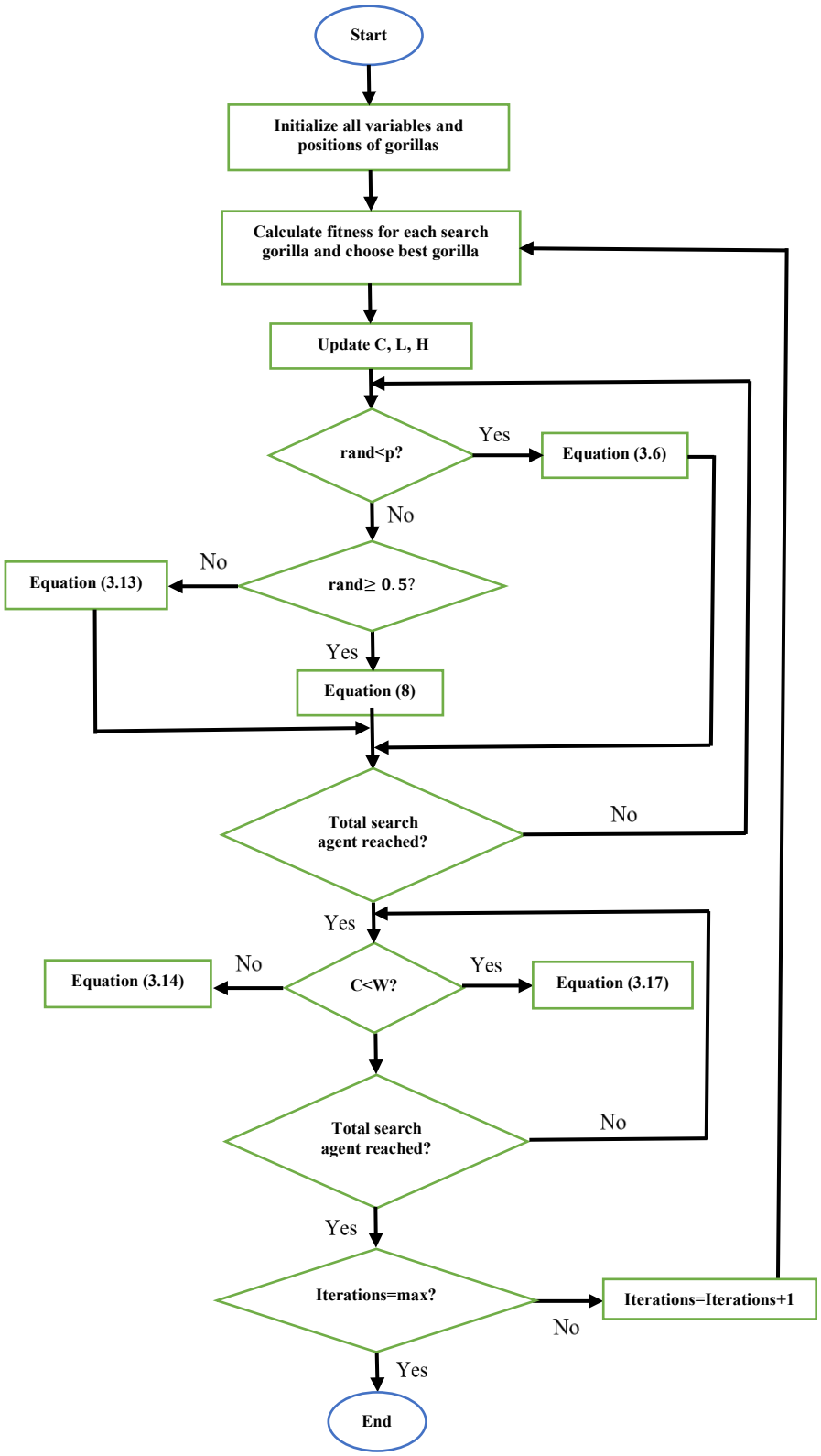


Figure 3.3: Flowchart of implementation of GTO [149], [150]

of vultures are split into two groups. Initially, each vulture is assigned a random position within the search space. Then, the fitness value of each vulture is computed. The vulture with the best fitness is placed in group one,

whereas the vulture with the second-best fitness is assigned group two. Thereafter, each of the remaining vulture are either sorted into group 1 or 2, depending on the following [153], [154], [155], [156]:

$$R_i = \begin{cases} BV_1, if\ p_i = x_1 \\ BV_2, if\ p_i = x_2 \end{cases} \tag{3.21}$$

Where:

- BV_1 and BV_2 denote the groups of the best and second-best vultures, respectively
- x_1 and x_2 are random number in the domain [0,1], and whose total sum equal one
- p_i is the probability based on the fitness function of the vulture being considered, as well as the fitness of the entire group. Its value is calculated based on the roulette wheel selection

Note that there exists a large probability that p_i will not exactly in value to x_1 or x_2. In such instance, the vulture will move to group where p_i is the closes to x. In the second phase, the starvation rate, and the subsequent effect of satiety on the behaviour of the vulture, is considered. The hunger intensity of a particular vulture is expressed as [153], [154], [155], [156]:

$$F_i = (2\times r_1 +)\times z \times (1 - \frac{i}{max_i}) + t \tag{3.22}$$

$$t = h \times \left(sin^w \left(\frac{\pi}{2} \times \frac{i}{max_i} \right) + cos^w \left(\frac{\pi}{2} \times \frac{i}{max_i} \right) - 1 \right) \tag{3.23}$$

Where:

- F_i is the satiation intensity of the i^{th} vulture at the i^{th} iteration
- r_1 is a random number in the domain [0,1]
- z is a random number in the domain [-1,1]
- i is the current iteration number
- max_i is the maximum number of iterations
- h is a random number in the domain [-2,2]
- w is a predefined constant

As evident by equations (3.22) and (3.23), as the iteration magnitude progresses, the value of the satiation intensity decreases. The algorithm proposes that when the magnitude of F_i is greater than 1, the vultures can be assumed to have sufficient energy to explore different areas of food sources. However, when the magnitude of F_i is less than 1, the vultures are assumed to be low on energy, and therefore deploy exploitation in their current vicinity. The nature of the vulture's flight enables a high visibility of the ground below them. However, being cautious creatures, they spend most of their energy ensuring that areas with a potential food source are safe to exploit. The exploration of potential food sources of the African Vulture is achieved via two methods. These are defined as [153], [154], [155], [156]:

$$P_{i+1} = R_i - D_i \times F_i \tag{3.24}$$

$$P_{i+1} = R_i - F_i + r_2 \times ((u - l) \times r_3 + l) \tag{3.25}$$

Where:

- P_{i+1} is the next position of the i^{th} vulture
- R_i is the position of one of the two best vultures
- r_2 and r_3 are randomized numbers in the range [0,1]
- $u\ and\ l$ are the upper and lower bounds of the search space, respectively

D_i is the distance between the vulture in question, and the current optimally situated vulture, and is denoted as [153], [154], [155], [156]:

$$D_i = |X \times R_i - P_i| \tag{3.26}$$

Where:

- X is a random number in the domain [0,2]
- P_i is the current position of the i^{th} vulture

The method of exploration chosen is dependent on a predefined constant, P_1. The value of P_1 lies in the domain [0,1]. Thereafter, a random number, also in the domain [0,1] is generated. If the value of P_1 is greater than that of thew randomly generated number, equation (3.24) is utilized. Else, (3.25) is deployed for exploration. As discussed earlier, if the magnitude of F_i is less than one, the vulture enters the exploitation stage of food search. The exploitation phase comprises of two sub-phases. The first phase is executed as soon as F_i is less than one. In this phase, as with the exploration phase, two possible exploitation methods exist. These are [153], [154], [155], [156]:

$$P_{i+1} = D_i \times (F_i + r_4) - d_i \tag{3.27}$$

$$P_{i+1} = R_i - (K_1 + K_2) \tag{3.28}$$

$$d_i = R_i - P_i \tag{3.29}$$

$$K_1 = R_i \times \left(\frac{r_5 - P_i}{2\pi}\right) \times \cos(P_i) \tag{3.30}$$

$$K_1 = R_i \times \left(\frac{r_6 - P_i}{2\pi}\right) \times \cos(P_i) \tag{3.31}$$

Where r_5 and r_6 are random numbers in the domain [0,1]. As with the exploration phase, the method of exploitation chosen is dependent on a predefined constant, P_2. The value of P_2 lies in the domain [0,1]. Thereafter, a random number, also in the domain [0,1] is generated. If the value of P_2 is greater than that of thew randomly generated number, equation (3.27) is utilized. Else, (3.28) is deployed for exploitation. The second phase of exploitation occurs when the magnitude of F_i is less than 0.5. As with exploration, and phase one of exploitation, two possible scenarios exist. First, the fierce competition between vultures for a food source is mathematically modelled. The second scenario expresses the scavenging behaviour of the vultures. In such instance, the remaining vultures move to the position of the best vulture, to salvage the remaining food. These are expressed in equations (3.32) and (3.35), and are as [153], [154], [155], [156]:

$$P_{i+1} = \frac{T_1 + T_2}{2} \tag{3.32}$$

$$T_1 = BV_{1(i)} - \frac{BV_{1(i)} \times P_i}{BV_{1(i)} - {P_i}^2} \times F_i \tag{3.33}$$

$$T_2 = BV_{2(i)} - \frac{BV_{2(i)} \times P_i}{BV_{2(i)} - P_i^2} \times F_i \quad (3.34)$$

$$P_{i+1} = R_i - |d_i| \times F_i \times Levy(d) \quad (3.35)$$

Where $Levy(d)$ denotes the conventional levy flight mechanism. As with the exploration phase, the method of exploitation chosen is dependent on a predefined constant, P_3. The value of P_3 lies in the domain [0,1]. Thereafter, a random number, also in the domain [0,1] is generated. If the value of P_3 is greater than that of thew randomly generated number, equation (3.32) is utilized. Else, (3.35) is deployed for exploitation. The flowchart depicting the process of execution of the AVOA is shown in figure 3.4 [153], [154], [155], [156].

3.5. Whale Optimization algorithm

The hunting strategy of the humpback whale is separated into three parts: searching, encircling and bubble-net attacking [86], [87], [88]. During searching, the humpback whales exchange information about prey to each other. This is to ensure that all the whales stay close to the prey. Consider the following [157], [158]:

$$X_i(t) = [X_{i,1}(t), X_{i,2}(t) \dots X_{i,D}(t)] \quad (3.36)$$

Where $X_i(t)$ is the current position of the i^{th} whale and D is the number of search space dimensions. The position of the whales at the next sampling instant can be updated using three methods. The first method is via a random search and is shown as [86], [157], [158]:

$$X_i(t+1) = X_r(t) - A|C \times X_r(t) - X_i(t)| \quad (3.37)$$

Where $X_r(t)$ is the position of a whale chosen at random and A and C are coefficients. A is based on the current and maximum iteration numbers, as well as a random number in the range [0,1]. C is based only on a random number in the range [0,1]. It is important to note that the random numbers used in the evaluation of A and C are generated independently. The second method is to

encircle the prey. To encircle the prey, each of the whales update their positions based on the best position found thus far. This update is represented as follows [86]:

$$X_i(t+1) = X_p(t) - A\left|C \times X_p(t) - X_i(t)\right| \quad (3.38)$$

Where $X_p(t)$ is the best position found thus far (at iteration t). The third method is via the use of bubble net attacking. Bubble net attacking is a mathematical model used to imitate the spiral movement of the humpback whale [67], [68]. In bubble net attacking, the whales update their positions as follows [86], [157], [158]:

$$X_i(t+1) = X_p(t) - \left|X_p(t) - X_i(t)\right| e^{bl} \times \cos(2\pi l) \quad (3.39)$$

Where b is a limited constant and l is a random number in the range [-1,1]. The method of position updating to be used is based on a random number q in the range [0,1], as well as the value of A. If q is less than 0.5 and the magnitude of A is greater than one, the whale positions are updated using encircling of the prey. If q is greater than 0.5 and the magnitude of A is greater than or equal to 1, the whale positions are updated randomly. Else, the bubble net attacking method of position updating is used [86].

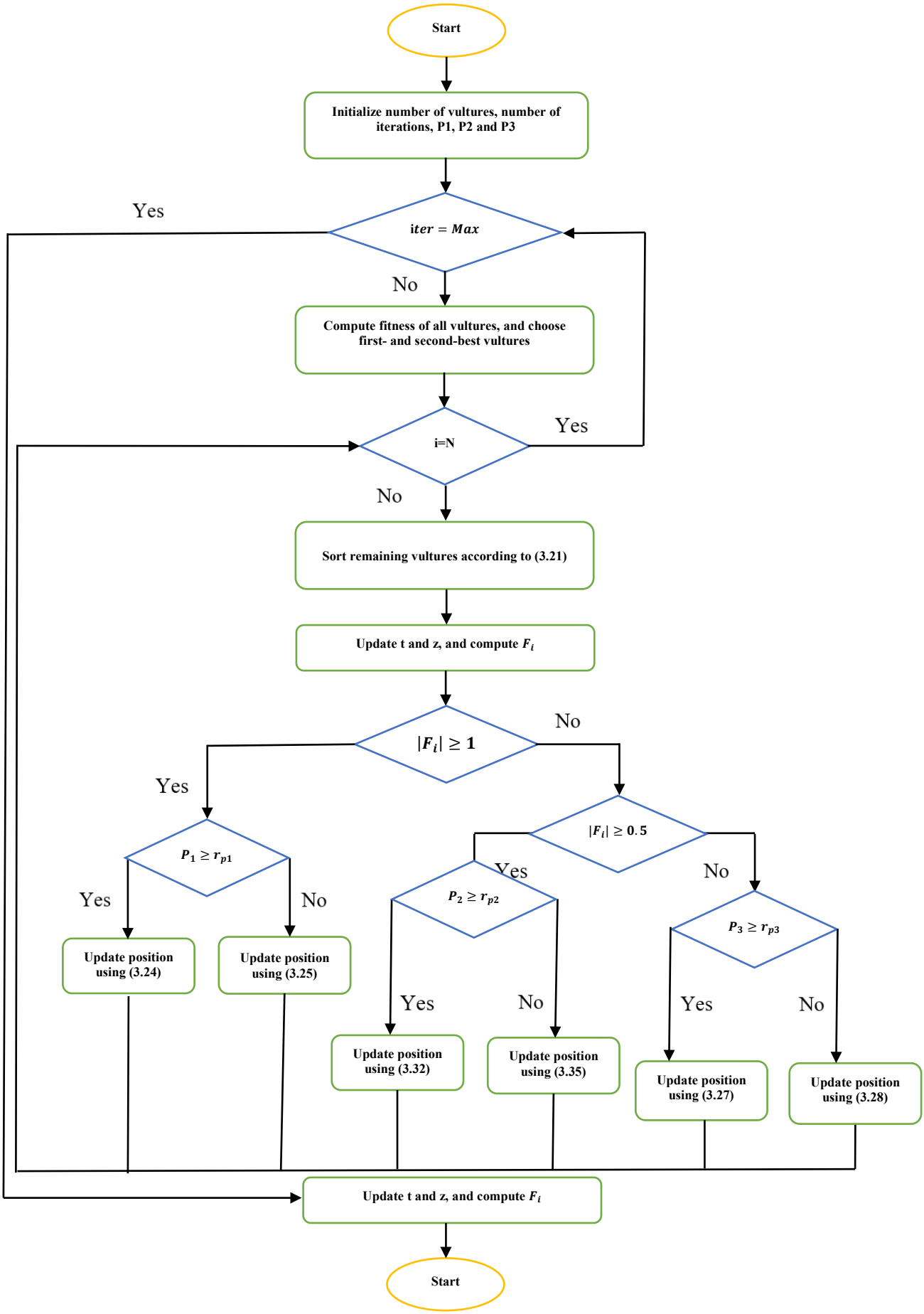


Figure 3.4: Flowchart of implementation of AVOA [153], [154], [155], [156]

Initially, the required parameters are defined. Then, each whale is given a random position. The fitness of each whale is calculated and the whale with the best fitness value is noted. The random numbers P and A are then generated. If the magnitude of A is less than 1 then the position of each whale is updated using (3.39). If P is less than 0.5 and the magnitude of A is greater than one, then the position of each whale is updated using (3.38). Lastly, if q is greater than or equal to 0.5, the position of each whale is updated using (3.37). After the update is completed, the fitness of each whale is calculated and replaces the current best fitness value (of that whale) if its value is superior to that of the current best. This continues until all iterations have been completed. Once this is so, the whale with the best fitness is said to be at the most optimal position [86]. The steps to execute the WOA can be seen in figure 3.5 [159].

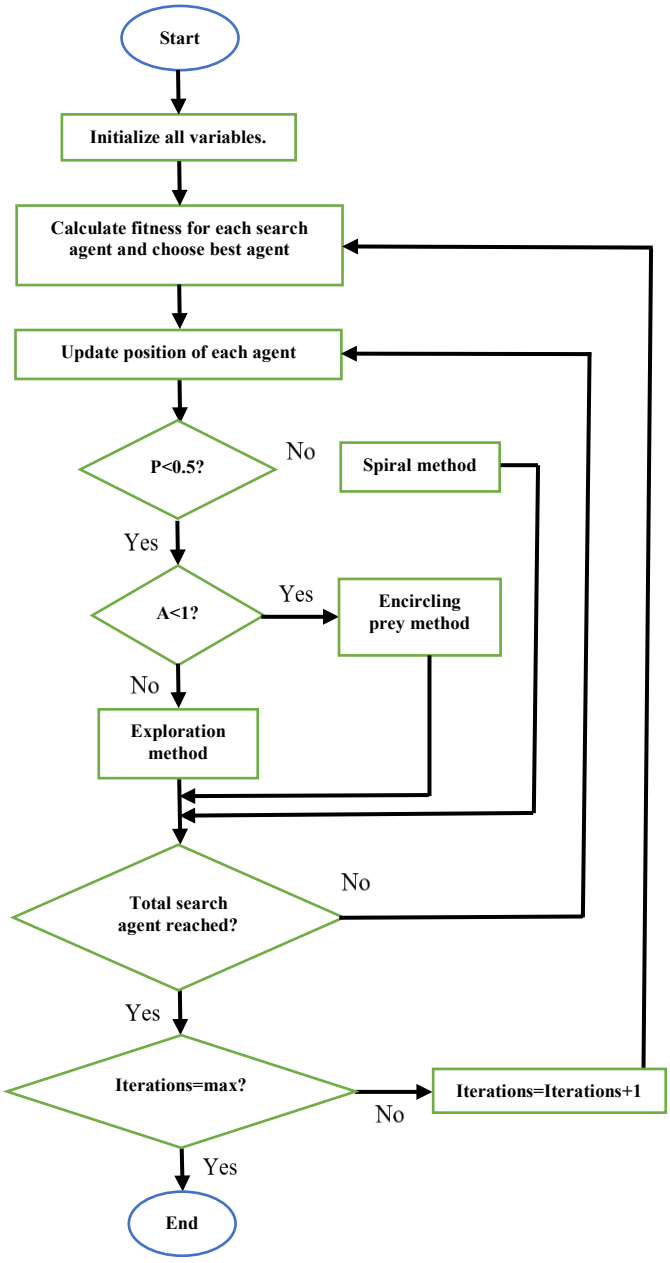


Figure 3.5: Flowchart of implementation of WOA [159]

3.5. A Modified Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) is a MOT which is inspired by the hunting tactic of the humpback whale [160]. Proposed by Mirjalili and Lewis in 2016, this relatively new MOT has shown promise in the optimization of complex engineering problems. This is largely due to the algorithm displaying the merit of a strong global search ability [161]. However, like all other MOT, the WOA suffers two critical demerits. These are

a slow convergence rate and poor accuracy [162]. There has been research done to mitigate such, but this has not been investigated thoroughly. This section aims to propose a modified Whale Optimization Algorithm to enhance the exploitation capability, and stability of the algorithm

3.5.1. Background

Since its inception, the WOA has come under investigation, albeit not extensively. In [163], Gaussian distribution strategies are used to simultaneously enhance the accuracy and convergence rate of the algorithm. However, this work only tests the proposed algorithm in 25 and 30 dimensions and utilizes the same search range across all test functions. The experiment conducted in [164] makes use of quadratic interpolation and a dynamic strategy to enhance the exploitation ability of the algorithm. This was aimed at improving the algorithm when attempting to solve large scale problems. The scholars in [165] proposed a modified technique for COVID-19 X-Ray image segmentation. The co-efficient vector A, as well as the constant value b are dynamically changed to improve both exploration and exploitation. When compared to various other modified versions of the conventional WOA, the proposed technique yielded a superior performance. In [166], a single dimensional swimming based WOA was proposed. This method was tested on a large range of dimensions, but the minimum dimension was 20. In [167], a Levy flight-based mutation, along with a pattern search mechanism are integrated into the conventional WOA. The mutation enhances the exploration and exploitation capability of the algorithm, whereas the patter search improves convergence rate and stability. The proposed technique was utilized for parameter identification of solar cells and photovoltaic modules. Considering the modification proposed in [168], a change to the algorithm structure is presented. This is along with a new position update equation for the encircling prey method. The new equation utilizes the positions of three mutually exclusive whale, which are different to the whale being updated. The algorithm structure change is in terms of a newly randomized number in the domain [0,1]. The positions of the whales are then updated according to the magnitude of this new randomized number. Hybridization of the WOA with other MOT have also been proposed in existing literature. In [169], the Grey Wolf Optimization algorithm, known to exhibit a strong local search ability, is integrated into the WOA (which lacks this aspect). However, the algorithm was tested using only one-dimension magnitude. The WOA was hybridized with a well-known evolutionary MOT, known as Genetic Algorithm, in [170].

In [171], the WOA was hybridized with the sine-cosine algorithm, a physics-inspired MOT. The proposed algorithm enhanced the exploration position update equation of the WOA via utilization of the position update equation of the sine-cosine algorithm, which makes use of four randomly generated numbers in the domain [0,1]. There exist two equations for the exploration search, with utilization determined by the magnitude of one of the random numbers in relation to a critical value. The authors in [171] applied the proposed algorithm to the IEEE 69-bus test system and compared to the conventional WOA. While producing an enhancement in the convergence rate and minor advancement in solution accuracy, the proposed algorithm was not rigorously tested on various

functions and at various dimension magnitudes. The WOA was combined with simulated annealing, another physics-inspired MOT, in [172]. The proposed approach utilizes simulated annealing after the WOA has completed running. Further, the concept of mutation and tournament selection were added to the position update equations of the WOA. The algorithm in [173] introduces four operators into the conventional algorithms. These are differential evolution, density peak clustering strategy, nonlinear parameter design and opposition-based learning method. The proposed technique was tested on various benchmark functions, as well as the seismic inversion problem and compared to various modified WOA technique. The proposed algorithm generated superior results in terms of average value and standard deviation. Considering convergence, the proposed algorithm exhibits superiority after the completion of about 35 iterations. Quadratic interpolation and Levy flight is utilized in [174] to enhance the accuracy of microarray data classification. When compared to another modified WOA, the proposed algorithm generated a superior accuracy. However, no information concerning convergence rate was provided. In [175], the concept of correction factors are applied to the various position update equations of the conventional WOA. When tested on a range of benchmark functions and compared to other conventional algorithms, the proposed technique produced the overall best average result. The proposed algorithm was further applied to an adaptive fuzzy logic PID controller for load frequency control. The result proved to be remarkable, but was not fared against other algorithms, thereby comprising the validity of superiority.

As evident, there has been proposed modifications to the conventional WOA. However, it is observed that there still exists a lack of precision, as well as a sub-par convergence rate. Both exploitation and rate of convergence are critical parameters in the performance of optimization techniques. In numerous applications, even a change of a fraction of a percentage may yield large savings in resources. While the relevant proposed algorithm succeeded in their objectives, there still exists gaps in application. The authors in [164], [166] and [168] do not consider small dimension magnitudes, and only apply the proposed algorithm to classical benchmark functions (as opposed to modern functions). In [167], there was no comparison to any other MOT, even the conventional WOA. The article outlined in [169] improves slightly in this aspect, making comparisons with conventional algorithm. However, no proposed techniques were evaluated. Considering [165], [170] and [172], application was made only to one real-world engineering problem. Further, no application to CEC benchmarks functions, or any other functions, were made. These deficiencies in current literature compromises the validation of the proposed novel ideas for global optimization, particularly in applications requiring high-precision results. An example of such is the gain values of PID controllers, where a low accuracy may result in sub-optimal controller performance. In electrical engineering, such problems are load flow analysis, economic load dispatch, and co-ordination of protection relays [176]. This correlates to optimal weight design of gear systems and machine scheduling in mechanical engineering [177]. In the civil and geotechnical engineering discipline, such examples are pile and rock design, and rock and soil mechanics [178]. There are also various optimization problems in other disciplines, like chemical engineering and computer engineering, where a high solution accuracy is imperative.

In this section, various modifications are applied to the conventional position update equations of the WOA. Further, there is a change in the structure of the algorithm. Firstly, the stochasticity of the exploration search is improved. Then, via the use of parameters already contained within the algorithm, all three position update equations are modified. The proposed algorithm also allows the whales a chance to undergo a dual position update, if a criterion is met. Lastly, a critical fragment of the well-known ABC optimization algorithm is deployed in the

proposed scheme. The aim of these updates is to mitigate the drawbacks of the conventional WOA, particularly in applications requiring high precision results. To demonstrate extensive testing, the proposed algorithm is applied to the CEC 2019 benchmark functions and compared to the conventional WOA, modified versions of such, as well as a recently proposed state-of-the-art technique. Further, to evaluate the true performance of the algorithm, the algorithm is applied to the optimal design of a pressure vessel.

3.5.3. Structure of proposed algorithm

The proposed WOA introduces various modifications to the position update equations of the WOA, as well as a change to the general structure of the algorithm. The aim of such is the mitigate the two common demerits of the WOA: low accuracy and slow convergence, as well as to prevent possible local optima entrapment at higher dimension optimization problems. To enhance the stochasticity of the exploration search method, the equation outlined in (3.37) is modified. This is via introduction of the position of another randomly chosen whale. This whale may be the same as the one already utilized in the current equation, or different. As in the original exploration search, the position of the current whale position is subtracted from the product of coefficient C and the second randomly chosen whale. The product of this value, as well as coefficient A, is added to the original equation. The exploration search then becomes:

$$\mathrm{X_i(t+1) = X_r(t) - A|C \times X_r(t) - X_i(t)| + A|C \times X_{r2}(t) - X_i(t)|} \tag{3.40}$$

Where $X_{r2}(t)$ is the position of the second randomly chosen whale. To further improve the search diversity and hence achieve an enhanced search accuracy, a fragment of the well-known ABC is utilized. In the conventional WOA, upon completion of all fitness values, each value is fared against the current best value. If a value is superior to the current value, it replaces the previous best value, and the corresponding whale positions now become the optimal positions. In the proposed technique, if the fitness value of a whale is not deemed superior to the current best, the position of that particular whale is updated as follows:

$$X_i(t+1)=|(X_i(t) + (\frac{t}{Max_{iter}}) \times (X_p(t) - X_j(t))| \tag{3.41}$$

Where t is the current iteration number, Max_{iter} is the maximum number of iterations and $X_j(t)$ is a randomly chosen whale from the population. The position of $X_j(t)$ must be different to $X_i(t)$. It is observed that the coefficient A utilized in the conventional WOA is a function of a linearly decreasing number. This number is a function of the current iteration number, as well as the maximum number of iterations. However, another component of this coefficient is a stochastic number, which lies in the domain [0,1]. An identical phenomenon is observed with the coefficients C (utilized in the exploration and encircling prey search position update equations and l (utilized in the spiral search position update equation). The stochastic nature of A is shown in figure 3.6. The magnitude of A can be seen to be oscillating between -1.5 and 2. The stochastic nature of coefficient C is shown in figure 3.7. The response of C is nearly identical to A, with two differences. Firstly, the magnitude oscillates between -0.5 and 2 and secondly, the frequency of ripples seems to be more than that of the response of A. Considering the trends of coefficients A and C, it is observed that the effect of a linearly decreasing, or increasing number, on the effect of the WOA is yet to be investigated.

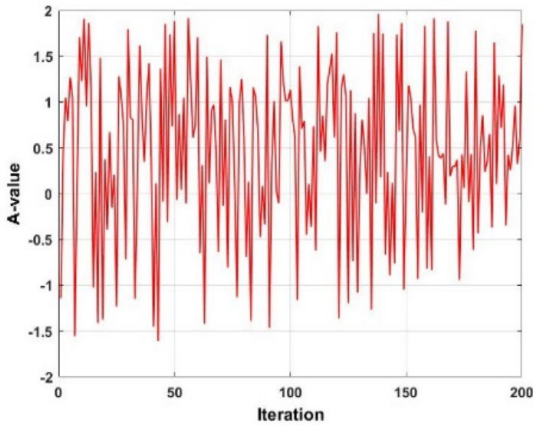


Figure 3.6: Stochastic nature of coefficient A in WOA

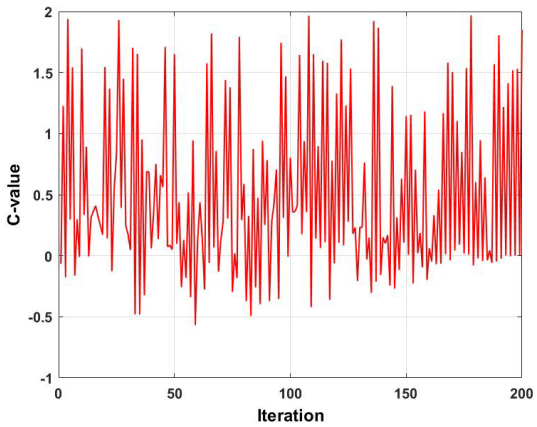


Figure 3.7: Stochastic nature of coefficient C in WOA

When applied to PSO, a linear increasing coefficient outperformed a linear decreasing one in terms of convergence rate and solution accuracy [179]. Therefore, a linearly increasing coefficient, which is a function of the current and maximum iteration numbers, is added to the second terms in (3.37) and (3.38), and the first term in (3.39). In [180], the concept of stochasticity was added to the cognitive and social constants, as well as the dynamic inertial weighting factor which are present in PSO. The results of this experiment produced a superior convergence rate to various other modified PSO algorithms. Utilizing this concept, a random number in the domain of [0,1] is added to the second terms in (3.37) and (3.38), and the first term in (3.39). Note that the three random numbers generated are unique to each other but may be equivalent in magnitude.

To further enhance the exploitation capability of the WOA, the tangent of the two linearly decreasing coefficients A and C are utilized to create a new term in each equation. Figure 3.8 shows the plot of the tangent of coefficient A. It is observed that the value of the coefficient fluctuates around the zero point, both in the positive and negative. Despite the magnitude of the value eventually settling to zero, the initial response will increase the algorithms probability of escaping the local optima. A similar phenomenon is seen in figure 3.9, which displays the plot of the tangent of coefficient C. The difference, however, is that in figure 3.9, there exists spikes of large magnitudes. Taking the product of both responses, figure 3.10 is derived. As evident, figure 3.10 is nearly identical to figure 3.9, but displays exacerbated qualities.

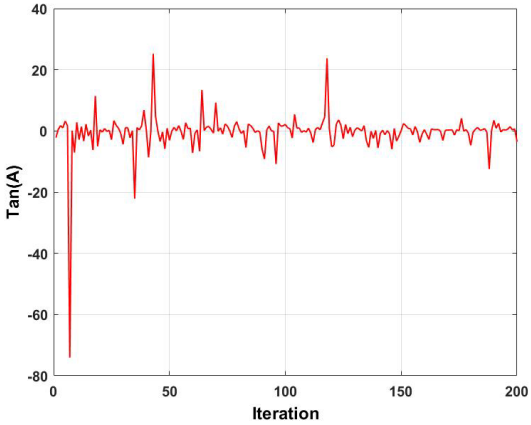


Figure 3.8: Tangent of coefficient A

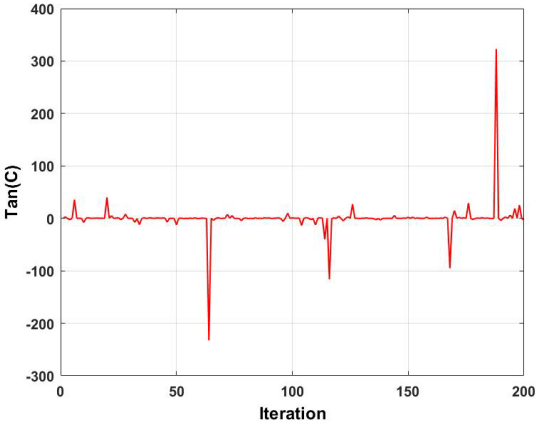


Figure 3.9: Tangent of coefficient C

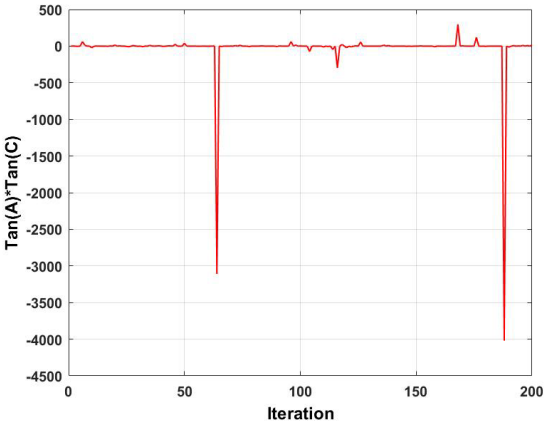


Figure 3.10: Product of the tangent of coefficients A and C

Considering the exploration search position update equation, the product of the tangent of A and tangent of C, along with a randomly chosen whale will exacerbate the randomness of the search. Considering the encircling prey and spiral search position update equations, the product of the tangent of A and tangent C along with the position of the best whale will enhance the exploitation capability of the search equation. This will be achieved

via the minor spikes, as observed in figure 6. The larger spikes which are present will allow the algorithm to increase its capability of jumping out of the local optima, particularly at higher order optimization problems. Considering (3.37), the new term is the product of the position of the random whale, as well as the tangent of coefficients A and C. Considering (3.38) and (3.39), the new term is the product of the position of the best whale, as well as the tangent of coefficients A and C. The effect of dimension magnitude inclusion in position updating equations is not well researched. In several instances, it is observed that algorithms lose either their exploration or exploitation capability when attempting to optimize large scale problems. To mitigate this adverse effect, the inverse of the dimension magnitude is added to the second terms in (3.37) and (3.38), and the first term in (3.39). Considering the various modifications that have been proposed, the new search equations are as follows:

Exploration method:

$$\mathrm{X_i(t+1)} = \mathrm{X_r(t)} - (1/\mathrm{dim}) \times \mathrm{rand1} \times (\mathrm{t}/\mathrm{Max_{iter}}) \times \mathrm{A}|\mathrm{C} \times \mathrm{X_r(t)} - \mathrm{X_i(t)}| + \mathrm{X_r(t)} \times tan(\mathrm{A}) \times tan(\mathrm{C}) \quad (3.42)$$

Encircling prey method:

$$\mathrm{X_i(t+1)} = \mathrm{X_p(t)} - (1/\mathrm{dim}) \times \mathrm{rand2} \times (\mathrm{t}/\mathrm{Max_{iter}}) \times \mathrm{A}\left|\mathrm{C} \times \mathrm{X_p(t)} - \mathrm{X_i(t)}\right| + \mathrm{X_p(t)} \times tan(\mathrm{C}) \quad (3.43)$$

Spiral method:

$$\mathrm{X_i(t+1)} = (1/\mathrm{dim}) \times \mathrm{rand3} \times (\mathrm{t}/\mathrm{Max_{iter}}) \times \mathrm{X_p(t)} - \left|\mathrm{X_p(t)} - \mathrm{X_i(t)}\right| \mathrm{e^{bl}} \times cos(2\pi \mathrm{l}) + \mathrm{X_p(t)} \times tan(\mathrm{C}) \quad (3.44)$$

Where $rand1$, $rand2$, $rand3$ are random numbers in the domain [0,1], dim is the dimension magnitude of theoptimization problem, t is the current iteration number and Max_{iter} is the maximum number of iterations. It is observed that there now exist two possible position search update equations for each of the exploration search, spiral search, and encircling prey. A random number in the domain [0,1] is defined. If this number is less than 0.65, then equations (3.40), (3.38), and (3.39) are implemented. The choice of equations is determined based on the values of A and P, exactly like how is determined in the conventional WOA. Upon completion of this, (3.42), (3.43), and (3.44) are implemented. The choice of equations is determined based on the values of A and P, exactly like how is determined in the conventional WOA. Hence, in such scenario, execution of (3.42) will succeed (3.40), (3.43) will succeed (3.38), and (9) will succeed (3.39). If the randomly generated number is greater than 0.65, then only equations (3.38), (3.39) and (3.40) are utilized. Once again, the choice of equations is determined based on the values of A and P exactly like how is determined in the conventional WOA. The structure of the proposed WOA, known as Enhanced Whale Optimization algorithm (EWOA) is therefore as shown in figure 3.111. In figure 3.11, R denotes a random number in the domain [0,1].

3.5.3. Experimental results

To validate the effectiveness of the proposed algorithm, the EWOA is applied to the well-known CEC2019 benchmark functions. A description of these ten functions can be found in [181]. The EWOA is compared to the conventional WOA, as well as two modified versions of the conventional WOA. To ensure rigorous testing, the EWOA was also fared against the farmland Fertility Algorithm, a newly proposed MOT which has thus far delivered promising results.

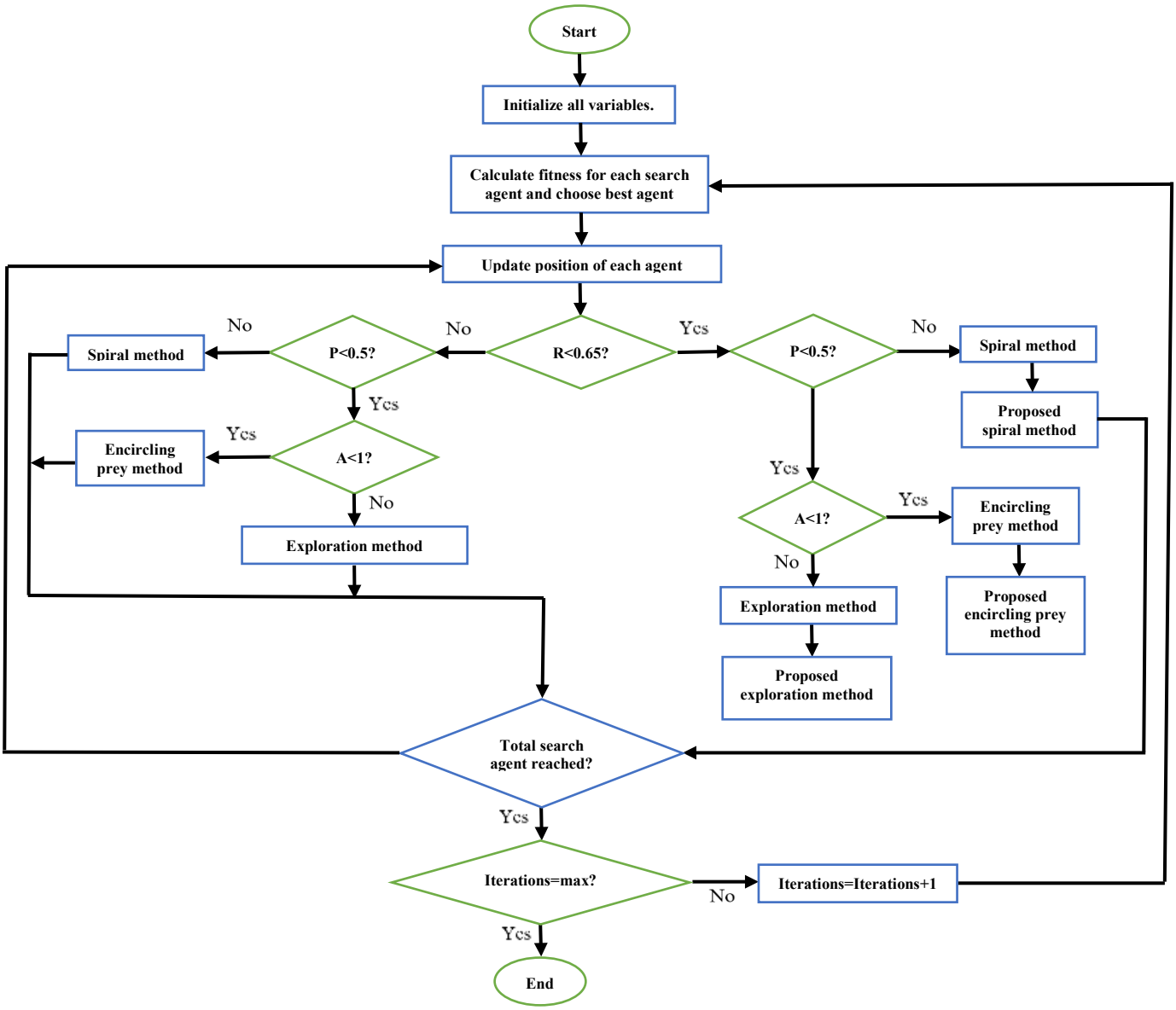


Figure 3.11: Structure of proposed EWOA

This algorithm was compared to various well-known MOT, including ABC, FA, and an improved PSO. The algorithm managed to exhibit superior results in many of the functions tested. Further, it is less complex than other well-known algorithms. Lastly, the algorithm has an extremely fast rate of convergence [183]. Owing to the stochastic nature of MOT, each algorithm was run 20 times. The results are given in terms of average value obtained, standard deviation, and convergence rate. The conventional WOA and MWOA1 were subject to the parameters presented in the original work [86]. This is also the case for MWOA2 [182] and the FFA [183]. The number of whales utilized in the EWOA is 1000, and the algorithm was subject to 100 iterations. Table 3.1 depicts the results obtained from application to the CEC2019 benchmark functions. From table 3.1, it is observed that the EWOA produced the best average value in seven of the 10 benchmark functions. It is also evident that from all five algorithms, the proposed EWOA yielded the best overall rank. Considering only WOA, the proposed algorithm is superior in nine of the ten functions. Considering function 2, it observed that the EWOA exhibits a 0.28% superiority over the next best algorithm (MWOA1). This corresponds to a large superiority of 27.43% over

the next best algorithm (WOA) for function 6. It is noted that FFA generates a fair number of good results but is still inferior to EWOA in terms of average ranking. Another advantage of the EWOA over the other algorithms is the concept of worst ranking. From Table 1, is it evident that the worst ranking attained by EWOA is third. This is superior to the FFA, WOA, and MWOA1 where the worst ranking attained is fourth, and superior to MWOA which attains a value of fifth. This points to an enhanced reliability of the proposed algorithm and verifies the EWOA as an effective algorithm for general optimization purposes.

Table 3.1: Performance analysis of proposed EWOA against other algorithms for the CEC2019 benchmark functions

Function		WOA	MWOA1	MWOA2	FFA	EWOA
1	Mean	1339570,194	51755,52062	**0**	4331170827	67948,78882
	Rank	4	2	**1**	5	3
	Std. dev	482949,83	5093,65	**0**	2484219709	12521,71
2	Mean	19,70131	17,41619	11929,59	17,47991	**17,36779**
	Rank	4	2	5	3	**1**
	Std. dev	0,47089	0,196649	4956,477	0,361856	**0,042862**
3	Mean	**12.7024**	**12.7024**	12.7065	**12.7024**	**12.7024**
	Rank	**2.5**	**2.5**	5	**2.5**	**2.5**
	Std. dev	**0**	**0**	0,000975	**0**	**0**
4	Mean	2767,298	1439,426	19627,67	**30,98509**	155,9416
	Rank	4	3	5	**1**	2
	Std. dev	1805,108	1358,2	5319,816	**14,81722**	64,82155
5	Mean	**8.1325**	**8.1325**	8.1518	**8.1325**	**8.1325**
	Rank	**2.5**	**2.5**	5	**2.5**	**2.5**
	Std. dev	**0**	**0**	0,008139	**0**	**0**
6	Mean	10,09868	10,73033	14,33454	10,25228	**7,92471**
	Rank	2	4	5	3	**1**
	Std. dev	1,16709	0,867645	1,079442	0,559376	**1,286647**
7	Mean	2097,158615	2097,15855	2199,621	**2097.1585**	**2097.1585**
	Rank	4	3	5	**1.5**	**1.5**
	Std. dev	0,000264127	0,000114708	67,75086	**0**	**0**
8	Mean	7,888065	7,89652	8,348015	**7,8796**	**7,8796**
	Rank	3	4	5	**1.5**	**1.5**
	Std. dev	0,026038	0,034719	0,261577	**0**	**0**
9	Mean	**4710,1035**	**4710,1035**	4711,26603	**4710,1035**	**4710,1035**
	Rank	**2.5**	**2.5**	5	**2.5**	**2.5**
	Std. dev	**0**	**0**	1,7626	**0**	**0**
10	Mean	20,90251	20,9092	21,11019	**20,9001**	20,90063
	Rank	**3**	4	5	**1**	2
	Std. dev	0,002697	0,028345	0,108713	**0**	0,001631

Average rank	31.5	29.5	36	23.5	**19.5**
Overall rank	4	3	5	2	**1**

The Wilcoxon Signed Rank Sum Test was carried out for the CEC 2019 benchmark functions. The test statistic was taken as the lower value of the positive and negative vales. For analysis purposes, a 5% confidence interval was utilized. This corresponds to a critical value of 52. Table 3.2 shows the results of the Wilcoxon Signed Rank Sum Test for each of the benchmark functions. It is evident from the table that apart from function 2 against MWOA1 and FFA, the proposed algorithm displays significance in all possible scenarios. This validates the effectiveness of the EWOA to provide a statistically significant superiority to all the compared algorithms.

Table 3.2: Wilcoxon ranked sum test

F	WOA	MWOA1	MWOA2	FFA
1	**0**	NA	NA	**0**
2	**0**	80	**0**	84
3	**0**	NA	**0**	NA
4	**0**	**0**	**0**	NA
5	NA	**2**	NA	NA
6	**0**	**0**	**0**	**2**
7	**0**	**0**	**0**	NA
8	**0**	**0**	**0**	NA
9	NA	NA	**0**	NA
10	**20**	**10**	**0**	NA

The convergence curve for functions 1 and 2 are displayed in figures 3.12 and 3.13. In figure 3.12, it is observed that the proposed algorithm produced a superior result to FFA for the entire duration. In figure 3.13, it is evident that the EWOA exhibits superiority over all the algorithms for the entire duration.

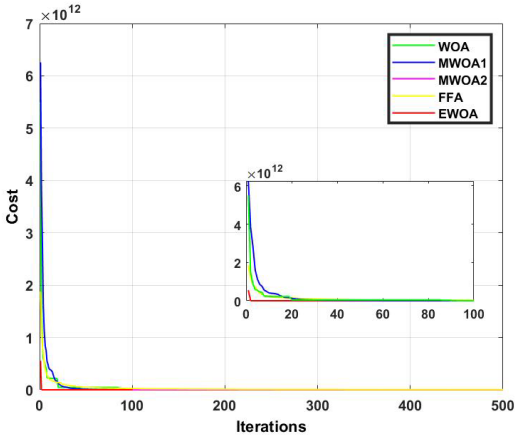


Figure 3.12: Convergence curve for F2

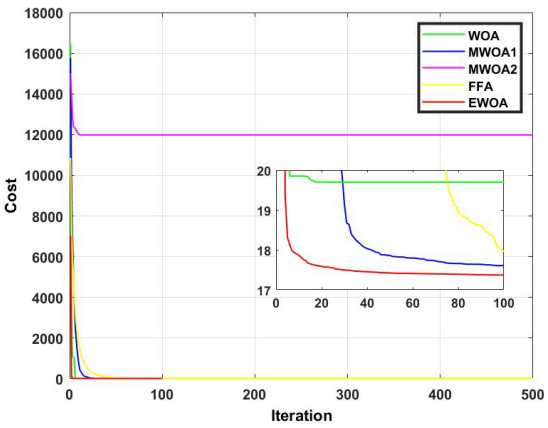


Figure 3.13: Convergence curve for F1

The convergence curve for functions 3 and 4 are displayed in figures 3.14 and 3.15. From figure 3.14, despite most of the algorithms producing an identical result, the EWOA exhibited dominancy in that convergence occurred within less than 20 iterations. In figure 3.15, the proposed algorithm was superior until about 60 iterations, after which the FFA reigned supreme. The convergence curve for functions 5 and 6 are displayed in figures 3.16 and 3.17. In figure 3.16, the EWOA is superior to FFA for about 10 iterations, and to MWOA1 until 60 iterations have complete. In figure 3.17, it is evident that the EWOA exhibits dominancy over all other algorithms for the entire duration.

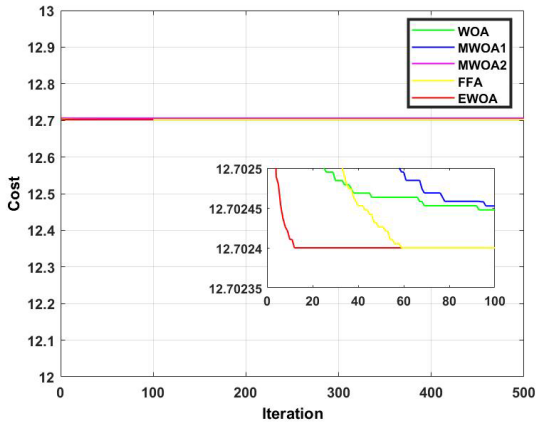


Figure 3.14: Convergence curve for F4

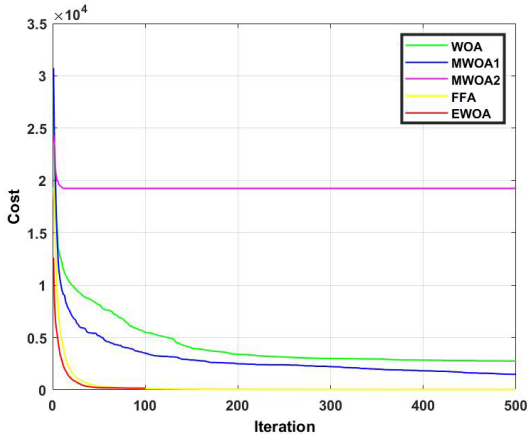


Figure 3.15: Convergence curve for F3

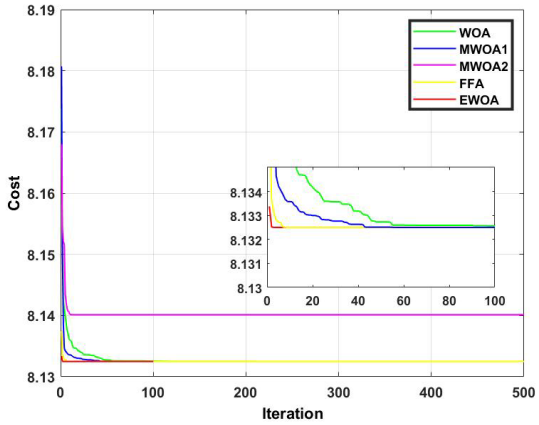


Figure 3.16: Convergence curve for F5

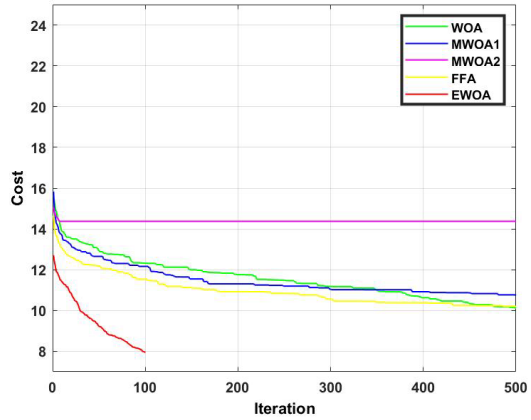


Figure 3.17: Convergence curve for F6

The convergence curve for functions 7 and 8 are displayed in figures 3.18 and 3.19. Figure 3.18 shows us that despite most of the algorithms producing an identical result, the EWOA converges much faster than any other algorithm. In figure 3.19, it is evident that the proposed algorithm exhibited dominancy for the entire duration and converges after a mere 10 iterations. The convergence curve for functions 9 and 10 are displayed in figures 3.20 and 3.21. In figure 3.20, it can be observed that the proposed algorithm produced the best convergence rate. In figure 3.21, despite the EWOA faring second to FFA, the proposed algorithm exhibited superiority until 100 iterations.

An important aspect of algorithm efficacy validation is application to practical engineering optimization problems. The proposed EWOA is applied to the design of a pressure vessel, a well-known constrained optimization problem

used to determine the efficacy of optimization techniques. The structural makeup of a pressure vessel can be observed in figure 3.22, where $T1$ is the head thickness, $T2$ is the thickness of the shell, L is the length of the cylindrical section of the vessel and R denotes the inner radius [184]. The ideal optimization cost of the pressure vessel design is zero.

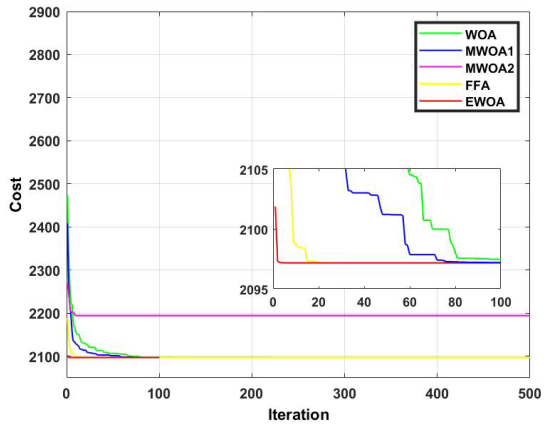


Figure 3.18: Convergence curve for F7

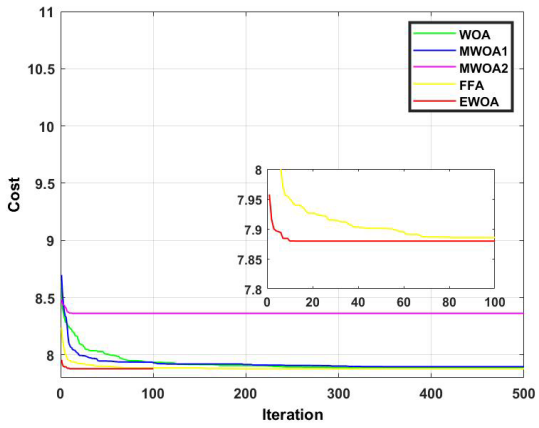


Figure 3.19: Convergence curve for F8

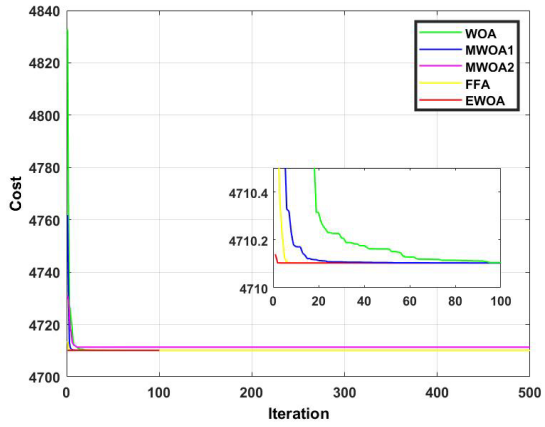


Figure 3.20: Convergence curve for F9

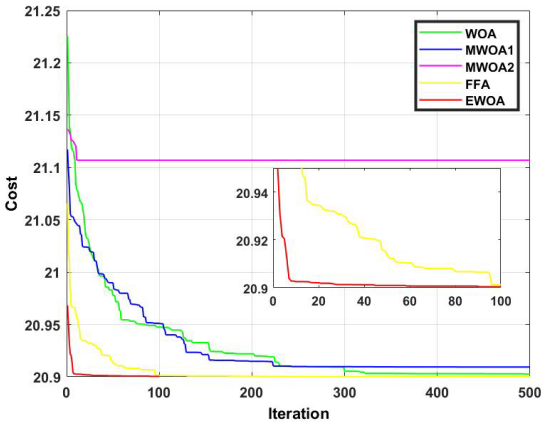


Figure 3.21: Convergence curve for F10

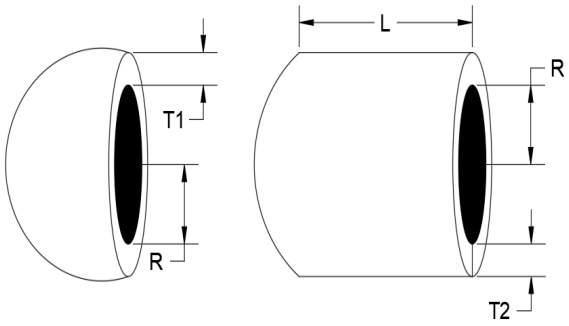


Figure 3.22: Structure of pressure vessel [43]

The cost function of the pressure vessel is expressed as [185]:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2{x_3}^2 + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3 \tag{3.45}$$

The design is subject to the following constraints [185]:

- $-x_1 + 0.0193x_3 \leq 0$
- $-x_2 + 0.00954x_3 \leq 0$
- $-\pi {x_3}^2 x_4 - \frac{4}{3}\pi {x_3}^2 + 1296000 \leq 0$
- $x_4 - 240 \leq 0$

Where x_1, x_2, x_3, x_4 denote $T2$, $T1$, R and L respectively. The design is subject to the following constraints. The range of the design variables are [185]:

- $1 \leq x_1 x_2 \leq 99$
- $10 \leq x_3 x_4 \leq 200$

Upon application of the said problem to the conventional WOA, MWOA and EWOA, the results obtained are as seen in table 3.3.

Table 3.3: Performance analysis for design of pressure vessel

Algorithm	Average	Std. Dev.
FFA	37397.72	0
MWOA1	9047.74	465.14
EWOA	**8810.96**	**17.34**

As evident from Table 3.3, FFA produces a standard deviation of zero. This indicates a strong exploitation capability. However, the algorithm struggles to escape from the local optima, thereby producing a significantly poorer solution to MWOA1 and EWOA. The EWOA produced the best average value, this being 2.62% superior to MWOA. This correlates to a standard deviation superiority of greater than 2500%.

3.6. Conclusion

This section provided an insight into the relevant equations and subsequent method of execution of various metaheuristic optimization techniques. These techniques that were discussed were Particle Swarm Optimization, Bat Algorithm, Gorilla Troops Optimization, African Vulture Optimization, and Whale Optimization Algorithm. Further, this section proposed an enhanced Whale Optimization Algorithm for optimization of complex engineering problems. The aim of such enhancement was to improve the search accuracy of the algorithm, as well as the stability of such. This is imperative for applications whereby precision results are valued, such as optimization of PI controllers, where incorrect controller tuning may result in unacceptable suboptimal performance. The proposed algorithm introduced various components to the position update equations of the WOA, as well as a change to the structure of the algorithm. Further, an aspect of the Artificial Bee Colony optimization algorithm was incorporated into the WOA. The proposed algorithm was applied to the CEC2019 benchmark functions and compared to the conventional WOA, modified versions of such, and the new Farmland Fertility Algorithm. The results show that the proposed algorithm produced the best result in 7 of the 10 functions. Further, the EWOA generated the best overall ranking. In addition, the reliability of the proposed method can be validated via observation that the poorest ranking of the proposed algorithm was third, lower than any other

compared algorithm. When applied to the optimal design of a pressure vessel, the proposed algorithm yielded significantly superior results to both the MWO1 and FFA. However, investigations revealed that the EWOA required a significant number of whales to prove superior. Further, based on the structure of the algorithm outlined in figure 3.12, there could exist instances whereby the whales will undergo a dual position change within one iteration. This may put a strain on the RAM of the PC being used, and as a result may not be able to successfully be executed on PCs with poor random-access memory. Further, these two aspects contribute the time taken to execute the algorithm, which is higher than that of the other compared algorithms. However, this is somewhat compensated for by the requirement and subsequent use of a significant lower number of iterations.

Chapter 4 : A Heuristic Approach to optimal crowbar setting and Low Voltage Ride Through of the Doubly Fed Induction Generator

This chapter aims to critically investigate and analyze the effect of applying swarm intelligence to the control of the Doubly Fed Induction Generator under the influence of symmetrical voltage dip. Swarm intelligence is applied to various aspects of control, such as crowbar protection, demagnetizing current injection, and control via the Linear Quadratic Regulator. Critical performance parameters to be analyzed are rotor current magnitude, as well as rotor DC voltage magnitude transients. Also, for the Fault Ride Through section, results are analyzed in terms of rotor direct axis current response (steady-state error, overshoot, and steady-state ripple). This is to ensure reliable control of the supply of reactive power in the case of three-phase voltage dips. The chapter begins by providing a background on the effects of symmetrical voltage dips, as well as current proposed techniques utilized to mitigate such. Then, the performance of the DFIG under such conditions is explained. Thereafter, control theory pertaining to crowbar protection, and design of the Linear Quadratic Regulator, is provided. Afterwards, the results of applying swarm intelligence to crowbar magnitude optimization is given. This is succeeded by the results of applying swarm intelligence to the optimal gain design of the demagnetizing current injection method, as well as control via the Linear Quadratic Regulator. The chapter is then concluded.

4.1. Background

Despite its many advantages, the direct grid connection of the DFIG has its drawbacks. In the case of a three-phase short-circuit fault, the DFIG reacts sensitively [186], [187]. The abrupt change in grid voltage magnitude creates a large transient stator flux [186]. This has the potential to cause drastic overvoltage and overcurrent in the rotor side of the DFIG [186], [187]. A common approach to preventing these catastrophic consequences would be to short-circuit the rotor via the use of a crowbar [186], [188], [189], [190]. While this works well, the optimal sizing of the crowbar resistance is challenging. A small crowbar resistance ensures a smaller rotor voltage, but a large spike in the rotor current. Similarly, a large crowbar resistance successfully limits the rotor current, but at the expense of an overvoltage. This is observed in figures 4.1 and 4.2. Therefore, optimal sizing of the crowbar circuit is essential.

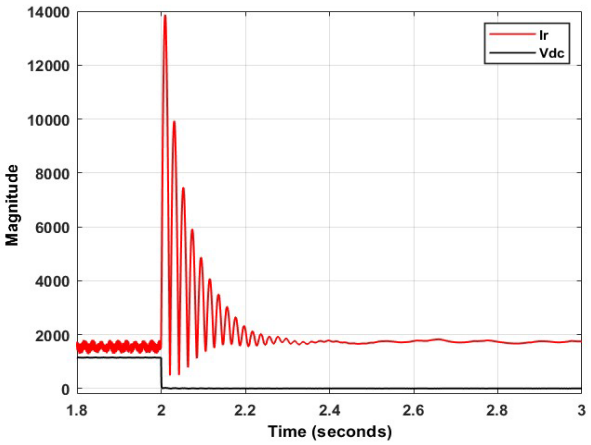


Figure 4.1: Rotor response with a 0.001 Ω crowbar.

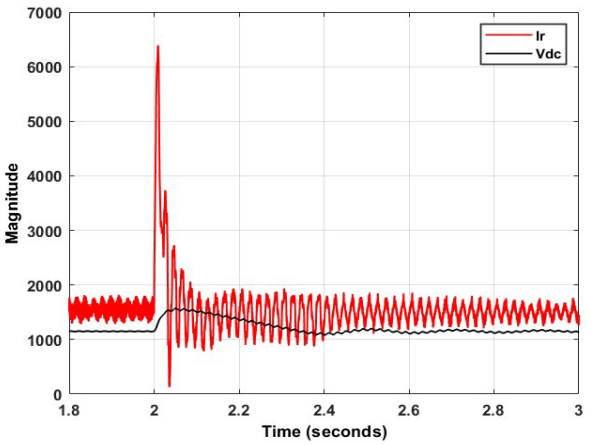


Figure 4.2: Rotor response with a 20 Ω crowbar.

The authors in [186] devised a method to determine the minimum and maximum value of the crowbar resistance. This considered both the maximum rotor voltage and maximum rotor current. Then, various values within this predetermined range were investigated. However, despite the rotor DC voltage being kept below its threshold, the best-case rotor current still exceeded 5 p.u. A similar approach was followed in [188], where the effects of varying the crowbar resistance was clearly depicted. An analytical hierarchy process-based algorithm was utilized in [191] to determine the optimal crowbar resistance. Once again, the rotor current reached unacceptable values of over 6 p.u. The method outlined in [192] makes use of thyristors to control the firing of the crowbar. The study emphasizes the effect of the crowbar resistance on the DC voltage but does not consider the rotor side current. Further, the study was not comprehensive. A study conducted in [193] analyzed the effect of rotor speed of the DFIG during grid faults. However, the study did not consider the rotor side current. The work presented in [194] made use of controllable series crowbar resistors. The resistors were activated via circuit breakers, utilizing the ANFIS system. The value of the crowbar resistors was calculated based on a multiplying factor of the stator resistance. However, as like other research presented, no effect of the fault on the rotor side current was analyzed.

Despite the crowbar approach working well in the reduction of rotor voltage and current transients, it has a major disadvantage; the crowbar resistances consume reactive power [188], [195]. Due to the continuous penetration of wind energy conversion systems to grids throughout the world, numerous nations have revised their grid code. Now, in the case of a three-phase grid voltage sag, WECS are required to remain connected to the grid and provide reactive power to ensure timeous restoration of the grid voltage. A common and established approach is to utilize a method called a demagnetizing current injection [196]. Considering that during the presence of a symmetrical voltage drip, the stator flux comprises both a natural and forced component, the demagnetizing current injection scheme works by estimating the induced natural flux, and then proceeds to eliminate such. The estimated natural flux, coupled with a negative gain, is fed into the respective rotor current reference. Owing to the initial success of the control strategy, various modifications have been proposed. The authors in [197] make use of the demagnetizing current to damp the DC component of the stator flux, as well as a passive switchable circuit to reduce the time constant and subsequent duration of the transient flux. The method presented in [198] makes use of both the crowbar and an optimized demagnetizing co-efficient to comply with grid code requirement. A robust control strategy was explained in [199], which utilizes explicit model predictive control to provide continuous adjustment of the demagnetizing current coefficient. A stator-connected dynamic volage restorer, with a rotor-connected inductor-based fault current limiter, was proposed in [200]. The method efficiently limited the rotor current transient, as well as the DC voltage peak. However, no insight on the reactive power was provided. The authors in [201] utilized fuzzy logic controllers to enhance the reactive power injection of the DFIG. While proving to be successful in achieving reactive power supply, the method did not analyze the effects of the control circuit on the rotor current transient and DC voltage magnitudes.

Apart from the above-mentioned techniques, various other methods are explained in [202]. One such method is the utilization of a DC chopper. In this method, a power electronic switch is used to control the current flowing through a braking resistor. This method successfully limits the DC voltage, but in its conventional form, cannot limit the rotor current transient. Another method is the use of a series dynamic resistor, which is placed between the rotor terminals of the DFIG and the rotor side converter. This method, however, proved to be inferior to the crowbar protection scheme. As of recent times, the linear quadratic regulator has come under investigation for the

purpose of LVRT of the DFIG. The authors in [203] utilized the LQR in place of the conventional PI controller, for stator flux-oriented vector control. The gains of the controller were optimized using a genetic algorithm. The controller produced good results but was only tested at a voltage dip of 50%. The LQR was optimized using PSO in [204], but this was only tested at steady-state response. A study conducted in [205] compared the efficacy of a GA-optimized LQR with the conventional LQR optimization method. However, only a 10% reduction in the grid voltage was tested. A relatively new optimization technique, called the mean variance mapping optimization algorithm, was proposed in [206] to optimize the gains of the LQR controller. The results were compared to that of the PI controller and proved to be superior. However, the machine was operated only in super synchronous mode, and only a voltage dip of 90% was evaluated. Further, evaluation of the post-fault steady-state error was not conducted. Lastly, the effect on transient rotor current was not clearly expressed.

As evident, the effects of crowbar circuits on the DFIG have been well investigated. However, there still exists a lack of optimal crowbar setting to ensure an adequate trade-off between the rotor side current and DC voltage. Further, owing to the continuous penetration of wind farms into the national grid, and subsequent stringent levels of grid codes, there is a dire need to improve fault ride through the capabilities of the DFIG. The promising results via utilization of the LQR makes this an attractive control method to mitigate the effects of symmetrical voltage dips. Thus far, it is evident that metaheuristic optimization techniques have been applied to LQR optimization. PSO, GA, and MVMO has thus far produced promising results. However, there still exists a deficiency of thorough investigation of results. Since their inception, metaheuristic optimization techniques have undergone various enhancements, and many new state-of-the-art optimization techniques were subsequently developed. This chapter provides a novel application of the African vulture's optimization algorithm for control of the DFIG during the case of symmetrical voltage dips. AVOA is a new swarm-based optimization technique, based on the social and feeding behavior of the various species of African vultures. Being a swarm intelligence technique, this algorithm is simple in structure. Further, as validated by the proposed authors, the algorithm exhibits strong exploration, exploitation, and convergence rate capabilities [153]. The algorithm is applied to achieve both optimal crowbar magnitude, as well as optimal gains for a LQR controller and demagnetizing current gains.

4.2. Response of the DFIG under asymmetrical voltage dips

Under stead-state conditions, the operation of the DFIG is governed by the following equations [207], [208]::

$$\overrightarrow{v_s} = R_s\overrightarrow{i_s} + \frac{d\overrightarrow{\emptyset_s}}{dt} \tag{4.1}$$

$$\overrightarrow{v_r} = R_r\overrightarrow{i_r} + \frac{d\overrightarrow{\emptyset_r}}{dt} - jw\overrightarrow{\emptyset_r} \tag{4.2}$$

$$\overrightarrow{\emptyset_s} = L_s\overrightarrow{i_s} + L_m\overrightarrow{i_r} \tag{4.3}$$

$$\overrightarrow{\emptyset_r} = L_r\overrightarrow{i_r} + L_m\overrightarrow{i_s} \tag{4.4}$$

where:

- $\overrightarrow{v_s}$ and $\overrightarrow{v_r}$ are the stator and rotor voltage vectors, respectively.
- $\overrightarrow{\emptyset_s}$ and $\overrightarrow{\emptyset_r}$ are the stator and rotor flux vectors, respectively.
- $\overrightarrow{i_s}$ and $\overrightarrow{i_r}$ are the stator and rotor current vectors, respectively.
- w is the rotor speed.

- R_s and R_r are the stator and rotor resistances, respectively.
- L_s, L_r and L_m are the stator, rotor, and magnetizing flux linkages, respectively.

From (4.3), making the rotor current the subject of the formula, the rotor flux can be expressed as [207], [197]:

$$\overrightarrow{\emptyset_r} = \frac{L_m}{L_s}\overrightarrow{\emptyset_s} - \sigma L_r \overrightarrow{i_r} \tag{4.5}$$

where $\sigma = 1 - \frac{L_s}{L_m L_r}$. Substituting (4.5) into (4.2), the rotor voltage can then be expressed as [207], [197], [209] :

$$\overrightarrow{v_r} = \frac{L_m}{L_s}\left(\frac{d}{dt} - jw\right)\overrightarrow{\emptyset_s} + (R_r + \sigma L_r(\frac{d}{dt} - jw)\overrightarrow{i_r} \tag{4.6}$$

The first term in (4.6) denotes the rotor voltage induced due to the stator flux, when the rotor is open circuit. The second term corresponds to the voltage drop in the rotor resistance and rotor transient inductance. Considering that the stator voltage is a time variant measurement, the stator voltage vector can be represented as [207], [197]:

$$\overrightarrow{v_s} = V_s e^{jw_s t} \tag{4.7}$$

where:

- V_s is the peak magnitude of the stator voltage.
- w_s is the synchronous speed of the revolving stator flux.

The stator flux can then be expressed as [207], [197]:

$$\overrightarrow{\emptyset_s} = \frac{V_s e^{jw_s t}}{jw_s} \tag{4.8}$$

During the steady-state operation, the grid voltage is constant. Therefore, the stator flux is constant. The stator flux rotates at grid frequency and represents the forced response of the system. This is shown as [207], [197]:

$$\overrightarrow{\emptyset_{sf}} = \frac{V_s e^{jw_s t}}{jw_s} \tag{4.9}$$

where $\overrightarrow{\emptyset_{sf}}$ denotes the forced flux. The rotor voltage due to the stator flux can be expressed as [197]:

$$\overrightarrow{v_{r0}} = jw_r \frac{L_m}{L_s}\emptyset_s = \frac{L_m}{L_s}\frac{w_r}{w_s}\frac{V_s e^{jw_s t}}{jw_s} \tag{4.10}$$

where:

- $\overrightarrow{v_{r0}}$ is the rotor voltage due to the stator flux;
- w_r is the slip speed.

Finally, the magnitude of the rotor voltage is [207], [210]:

$$V_{r0} = V_s \frac{L_m}{L_s} s \tag{4.11}$$

With $s = \frac{w_r}{w_s}$. During the case of a partial symmetrical voltage dip, the stator voltage may be expressed as [211], [210]:

$$\overrightarrow{v^s{}_s} = \begin{cases} V_s e^{jw_s t}, & t < 0 \\ V_s e^{jw_s t}(1-k), & t \geq 0 \end{cases} \tag{4.12}$$

Where k represents the depth of the symmetrical fault. The stator flux can then be expressed as [197], [209] :

$$\overrightarrow{\emptyset^s{}_s} = \begin{cases} \dfrac{\overrightarrow{v_s}}{jws} e^{jw_s t}, & t < 0 \\ \dfrac{\overrightarrow{v_s}(1 - k)}{jws} e^{jw_s t}, & t \geq 0 \end{cases} \tag{4.13}$$

Because the stator flux is a state variable, the magnitude of the flux cannot change instantaneously. Instead, there is a gradual decay from one steady-state magnitude to another. The change in stator flux is seen as [207]:

$$\frac{d}{dt}\overrightarrow{\emptyset^s{}_s} = \overrightarrow{v^s{}_s} - \frac{R_s}{L_s}\overrightarrow{\emptyset^s{}_s} \tag{4.14}$$

During a three-phase symmetrical voltage dip, the stator flux comprises two components: the forced flux and the natural flux. These are quantized as [197], [209]:

$$\overrightarrow{\emptyset^s{}_{sf}} = \frac{V_f e^{jw_s t}}{jw_s} \tag{4.15}$$

$$\overrightarrow{\emptyset^s{}_{sn}} = \frac{V_f e^{-t/\tau}}{jw_s} \tag{4.16}$$

where:

- $\overrightarrow{\emptyset^s{}_{sn}}$ represents the natural stator flux.
- V_f represents the post-fault stator voltage.
- τ is a time constant which depends on the machine stator parameters.

During the dip, the forced flux continues to rotate at grid frequency, while the natural flux remains stationary with respect to the stator. In reference [196], the evolution of the stator flux during the case of a symmetrical voltage dip was provided. As evident, the flux continued to rotate in a circular fashion. This is owing to the effect of the forced flux, which was still rotating at synchronous speed. However, the circular trajectory of the flux was not perfectly centered. This is because of the natural flux. The flux continued to move in a decreasing spiral fashion, eventually becoming centered and subsequently reaching a new stead-state magnitude. Both the forced and natural fluxes induced individual rotor voltages [208]:

$$\overrightarrow{e^r{}_r} = \overrightarrow{e^r{}_{rf}} + \overrightarrow{e^r{}_{rn}} \tag{4.17}$$

where:

- $\overrightarrow{e^r{}_r}$ is the total induced rotor emf;
- $\overrightarrow{e^r{}_{rf}}$ is the emf induced due to the forced flux;
- $\overrightarrow{e^r{}_{rn}}$ is the emf induced due to the natural flux.

These quantities can be expressed as [210], [207]:

$$\overrightarrow{e^r{}_{rf}} = \frac{sL_m}{L_s} V_f e^{jw_s t} \tag{4.18}$$

$$\overrightarrow{e^r{}_{rn}} = -\frac{L_m}{L_s} jw_m \emptyset_{n0} e^{-t/\tau} \tag{4.19}$$

$$\overrightarrow{e^r{}_r} = \frac{L_m}{L_s} V_s (s(1 - p)e^{jw_r t} - (1 - s)pe^{jw_m t} e^{-t/\tau} \tag{4.20}$$

The second term in (4.20) is of particular interest. Due to this term being dependent on the slip, depth of fault, and the rotor speed, this term has the potential to induce tremendously large rotor voltages.

4.3. Control theory on crowbar protection and LQR control of DFIG

The aim of this section is to provide insight into the methods undertaken to execute this investigation. The section starts by explaining the concept of the crowbar protection of the DFIG. Then, information regarding the linear quadratic regulator is provided, with a graphical representation of the function block diagram to be implemented. Lastly, the necessary theory and subsequent flowchart of the African vulture's optimization algorithm, which will be utilized in this research, is provided.

4.3.1. Crowbar Protection

In Section 4.2, the effect of a symmetrical voltage dip on the DFIG was examined. However, this was considering that the rotor was open circuit. When the rotor is closed during the presence of a voltage dip, the machine behaves differently. This forms the basis of the crowbar method of protection. Consider the following [196]::

$$\overrightarrow{v^s}_s = R_s\overrightarrow{\imath^s}_s + \frac{d}{dt}\overrightarrow{\emptyset^s}_{sn} = 0 \tag{4.21}$$

Expressing the stator current with respect to the rotor current yields [196]:

$$\overrightarrow{\imath^s}_s = \frac{\overrightarrow{\emptyset^s}_{sn}}{L_s} - \frac{L_m}{L_r}\overrightarrow{\imath^s}_r \tag{4.22}$$

Substituting (4.22) into (4.21), the rate of change of the natural stator flux can be quantized as [196]:

$$\frac{d}{dt}\overrightarrow{\emptyset^s}_{sn} = -\frac{R_s}{L_s}\overrightarrow{\emptyset^s}_{sn} + \frac{L_m}{L_s}R_s\overrightarrow{\imath^s}_r \tag{4.23}$$

The second term in (23) depicts the influence of the rotor current on the evolution of the natural flux. This forms the basis of the crowbar method of control. In the case of a three-phase symmetrical voltage dip, the crowbar creates a short-circuit on the rotor of the machine, causing a large rotor current. This expedites the damping of the natural flux, as shown in Figures 4 and 5. From these figures, it is evident that a short circuit on the rotor terminals drastically reduces the damping time of the natural stator flux. Further, the short-circuit is effective at preventing a transient in the stator flux [196].

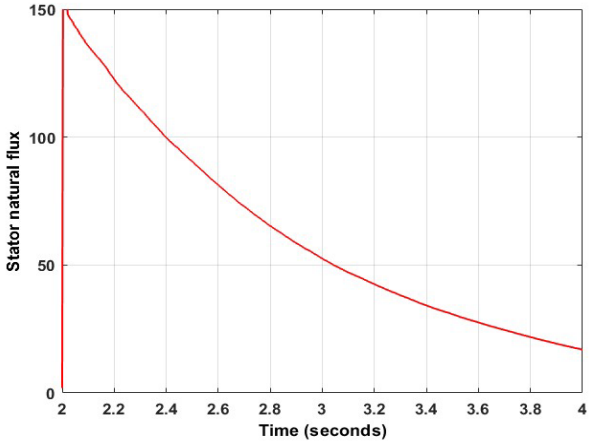


Figure 4.3: Stator natural flux for open-circuit rotor terminals

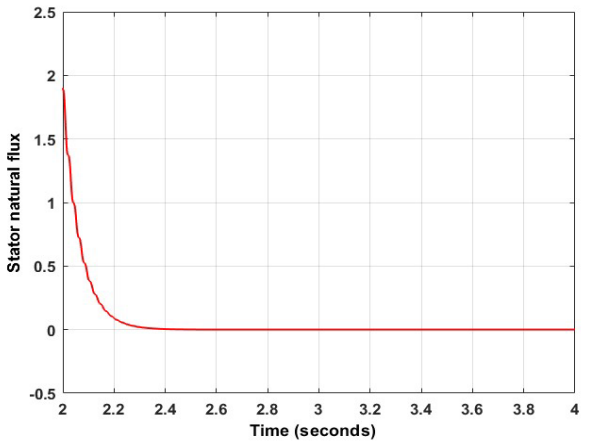


Figure 4.4: Stator natural flux for short-circuit rotor terminals.

Hence, it is concluded that the crowbar is an effective and ultimately essential method of protection of the DFIG. The crowbar resistors are open circuited during normal operation of the DFIG and are activated via power

electronic switches during fault conditions. The topology of the crowbar circuit can be observed in Figure 4.5 [196].

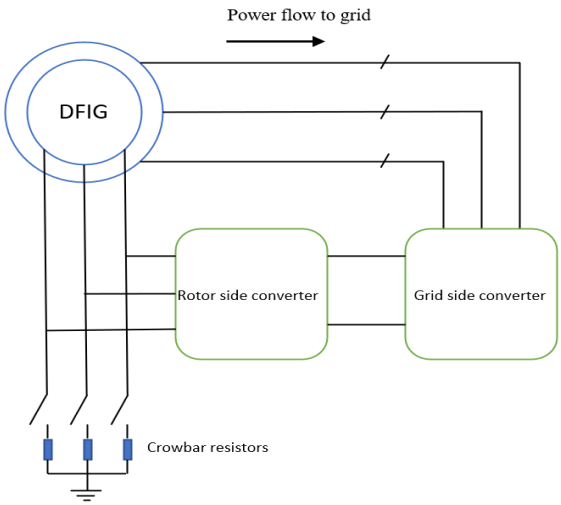


Figure 4.5: Topology of crowbar protection of DFIG [196].

4.3.2. The Linear Quadratic Regulator

The linear quadratic regulator is based on the specific linear time invariant system equation, which is given by [206], [205]:

$$\dot{x} = Ax + Bu \tag{24}$$

$$y = Cx + Du \tag{25}$$

where:

- $\dot{x}$ is the state of the system;
- u is the system input;
- y is the system output.

The purpose of the LQR is to obtain an optimized gain, K, to satisfy the following [206], [205]:

$$u(t) = -Kx(t) \tag{4.26}$$

This optimal value is generally obtained via the use of the following cost function [206], [205], [204]:

$$J(u) = \int_0^\infty [x^{-T}(t)Qx(t) + u^T(t)Ru(t)] \tag{4.27}$$

where Q and R are symmetrical matrices, used to weight the effort of control, and controller accuracy, respectively. Considering the stator-flux oriented vector control of the DFIG, the rotor voltages are expressed as [212], [39]:

$$V_{dr} = R_r i_{dr} + \sigma L_r \frac{di_{dr}}{dt} - w_r \sigma L_r i_{qr} + \frac{L_m}{L_s}\frac{d\theta_{ds}}{dt} \tag{4.28}$$

$$V_{qr} = R_r i_{qr} + \sigma L_r \frac{di_{qr}}{dt} + w_r \sigma L_r i_{dr} + \frac{L_m}{L_s}\emptyset_{ds} \tag{4.29}$$

Therefore, the dynamic closed-loop behavior of the DFIG can be represented as [204], [212]:

$$\dot{x} = Ax + Bu + Tr \tag{4.30}$$

$$\dot{y} = Cx \tag{4.31}$$

where:

- $A, B, C,$ and T are square matrices;
- r is the nonlinear measurable term.

Considering the DFIG as a fifth order dynamic system, rewriting (4.28) and (4.29), making the derivative of the rotor current the output, the following is obtained [204], [212]:

$$\begin{bmatrix} \frac{di_{dr}}{dt} \\ \frac{di_{qr}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{\sigma L_r} & w_r \\ -w_r & -\frac{R_r}{\sigma L_r} \end{bmatrix} \begin{bmatrix} i_{dr} \\ i_{qr} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_r} & 0 \\ 0 & \frac{1}{\sigma L_r} \end{bmatrix} \begin{bmatrix} v_{dr} \\ v_{qr} \end{bmatrix} + \begin{bmatrix} 0 & \frac{w_r L_m}{\sigma L_r L_s} \\ -\frac{w_r L_m}{\sigma L_r L_s} & 0 \end{bmatrix} \begin{bmatrix} \emptyset_s \\ 0 \end{bmatrix} \tag{4.32}$$

Note that in (4.32), the rotor current is taken as the state, the stator flux (derived from the stator voltage) is taken as the disturbance, and the rotor voltage is taken as the control. Generally, the linear quadratic regulator is not well suited for reference tracking applications. This may prove to be problematic, as the operation of the DFIG relies on the decoupled control of the rotor current magnitudes, which follow a reference value. To solve this, an integrator is added to the topology of the regulator. The function block diagram of the linear quadratic regulator to be applied to the DFIG is shown in Figure 4.6 [204].

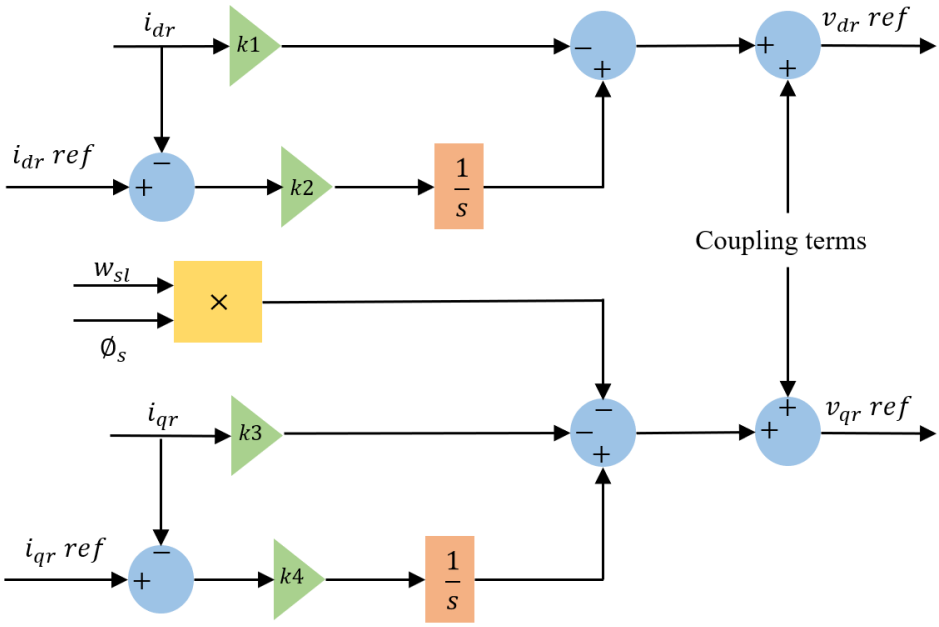


Figure 4.6: Functional block diagram of LQR applied to DFIG [203].

4.4. Investigation into Optimization of Crowbar Resistance via Swarm Intelligence

The aim of this section is the investigate the effectiveness of utilizing the swarm intelligence for determining the optimal setting of the crowbar resistance values. It is known that a high crowbar resistance will limit the fault current, but at the expense of a significantly higher DC voltage. Similarly, a smaller resistance will limit the DC voltage, but will produce an exorbitantly high fault current transient. Both scenarios produce unfavorable

outcomes. The aim of the optimal crowbar setting is to attempt to provide a viable trade-off between the rotor transient current and the DC voltage magnitudes. For this purpose, the well-known particle swarm optimization and the recently proposed African vulture's optimization algorithm will be implemented. The results for both PSO and AVOA for a 70%, 80%, and 90% symmetrical voltage dip are presented. The results are in terms of rotor current and DC voltage transient and settling times. Each of the utilized optimization techniques were subject to 20 iterations with 20 search agents. The lower and upper bounds of the search space were set to 0.00001 and 10, respectively. The objective function was taken as the integral time absolute error between the maximum and actual values of both the rotor current and dc link voltage. This will ensure the smallest possible overshoot, as well as a quick recovery time.

4.4.1. Case 1a: Voltage Dip of 70%

Figures 4.7–4.10 depict the results of applying swarm intelligence to the optimization of crowbar resistance for a 70% dip. Figures 4.7 and4.8 show the rotor current and DC voltage magnitudes, respectively, when applying PSO. Figures 4.9 and 4.10 show the rotor current and DC voltage magnitudes, respectively, when applying AVOA.

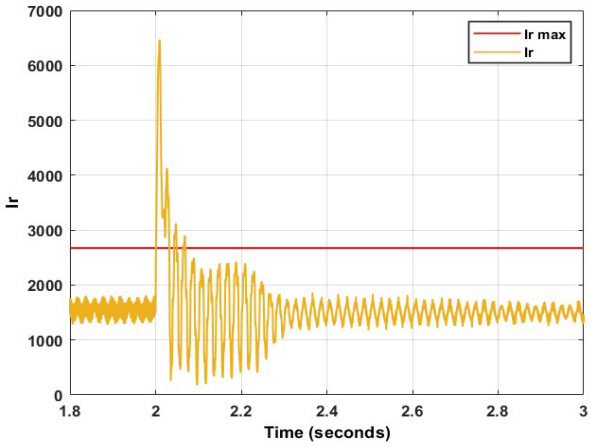


Figure 4.7: Rotor current transient using PSO at 70% dip.

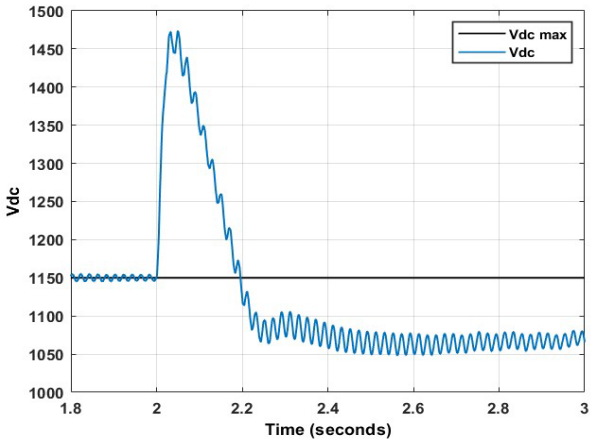


Figure 4.8: DC voltage transient using PSO at 70% dip.

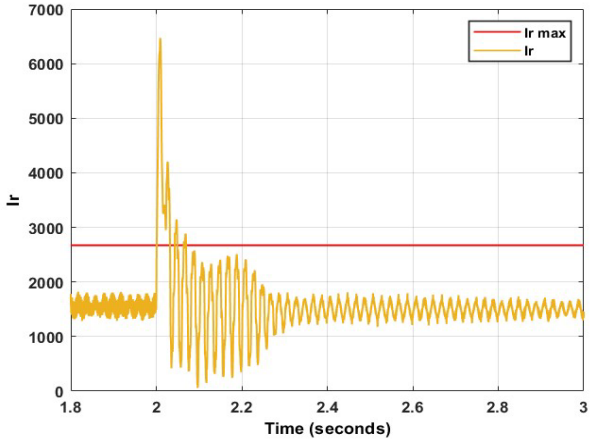


Figure 4.9: Rotor current transient using AVOA at 70% dip.

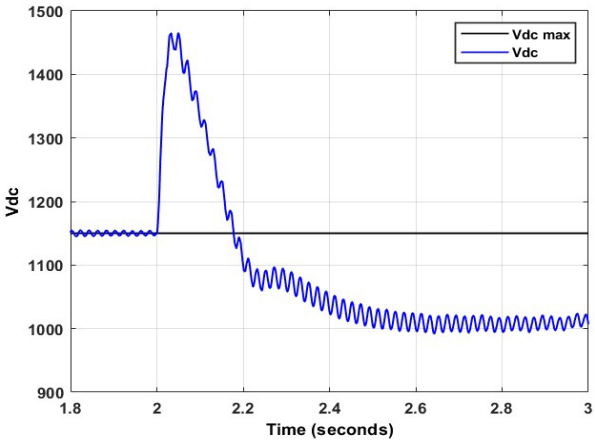


Figure 4.10: DC voltage transient using AVOA at 70% dip

From Figure 4.7, it is observed that the transient rotor current when applying PSO is 6468 A, with a settling time of 70 ms. Considering the DC voltage when utilizing PSO in Figure 4.8, the transient voltage is 1465 A. The settling time of this voltage is 175 ms. From Figure 4.9, it is observed that the transient rotor current when applying

AVOA is 6472 A, with a settling time of 70 ms. Considering the DC voltage when utilizing PSO in Figure 4.10, the transient voltage is 1465 A. The settling time of this voltage is 175 ms. It is therefore noted that for a 70% voltage dip, both the algorithms produced identical transient DC voltages and settling times. However, PSO exhibited superiority over AVOA for the rotor current transient, by a minute factor of 0.062%.

4.4.2. Case 1b: Voltage Dip of 80%

Figures 4.11–4.14 depict the results of applying swarm intelligence to the optimization of crowbar resistance for an 80% dip. Figures 4.11 and 4.12 show the rotor current and DC voltage magnitudes, respectively, when applying PSO. Figures 4.13 and 4.14 show the rotor current and DC voltage magnitudes, respectively, when applying AVOA.

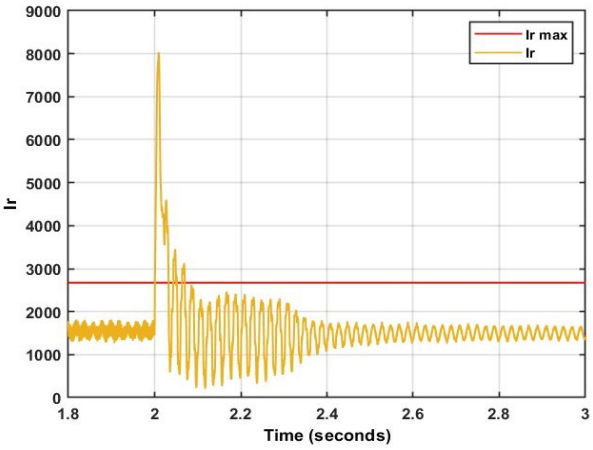


Figure 4.11: Rotor current transient using PSO at 80% dip.

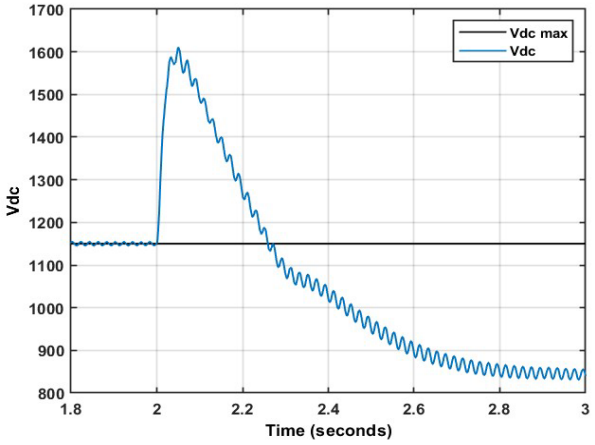


Figure 4.12: DC voltage transient using PSO at 80% dip.

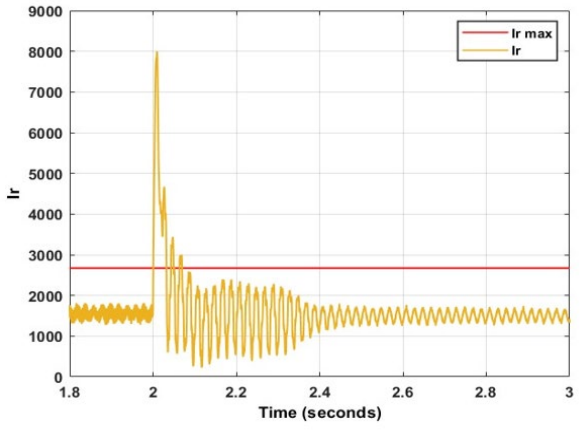


Figure 4.13: Rotor current transient using AVOA at 80% dip.

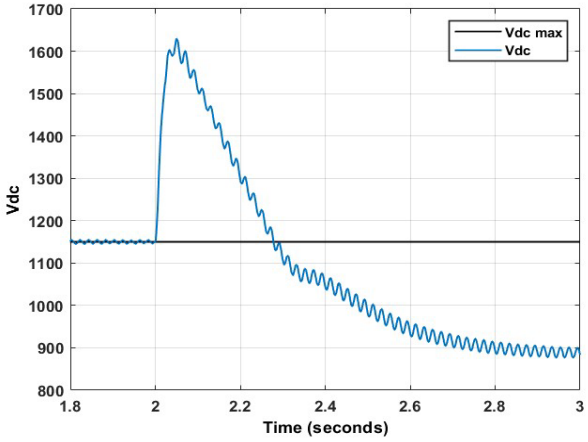


Figure 4.14: DC voltage transient using AVOA at 80% dip.

From Figure 13, it is observed that the transient rotor current when applying PSO is 8035 A, with a settling time of 70 ms. Considering the DC voltage when utilizing PSO in Figure 14, the transient voltage is 1612 A. The settling time of this voltage is 260 ms. From Figure 15, it is observed that the transient rotor current when applying AVOA is 7999 A, with a settling time of 70 ms. Considering the DC voltage when utilizing PSO in Figure 16, the transient voltage is 1630 A. The settling time of this voltage is 280 ms. It is therefore noted that for a 70% voltage dip, both the algorithms produced superior results, but in different aspects. PSO exhibited superiority with regards

to the DC voltage transient and settling time, in the order of 1.12% and 7.69%, respectively. AVOA produced a superior rotor current transient by a magnitude of 0.45%.

4.4.3. Case 1c: Voltage Dip of 90%

Figures 4.15–4.18 depict the results of applying swarm intelligence to the optimization of crowbar resistance for a 90% dip. Figures 4.15 and 4.16 show the rotor current and DC voltage magnitudes, respectively, when applying PSO. Figures 4.17 and 4.18 show the rotor current and DC voltage magnitudes, respectively, when applying AVOA.

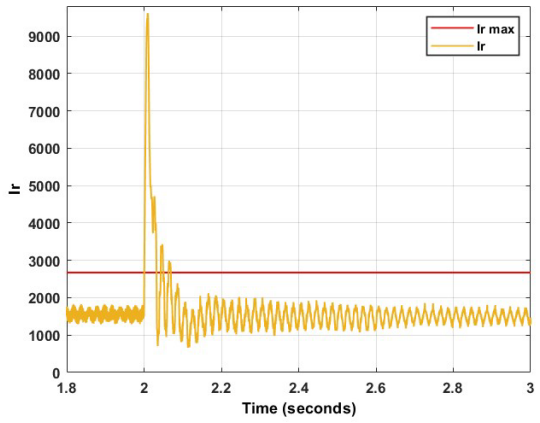


Figure 4.15: Rotor current transient using PSO at 90% dip.

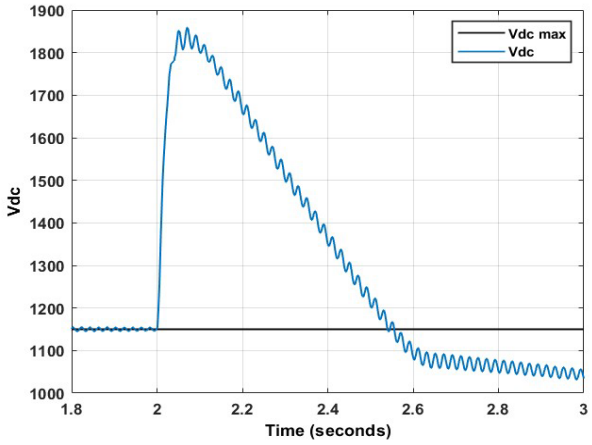


Figure 4.16: DC voltage transient using PSO at 90% dip.

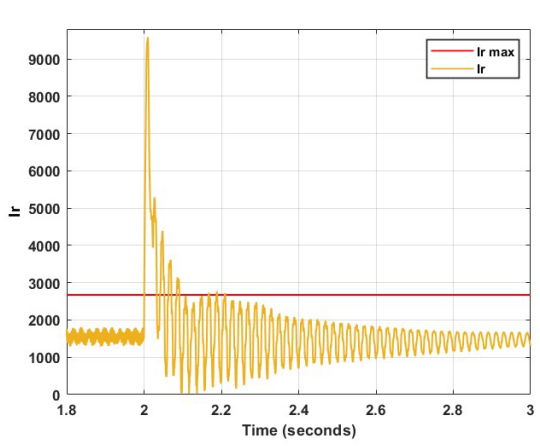


Figure 4.17: Rotor current transient using AVOA at 90% dip.

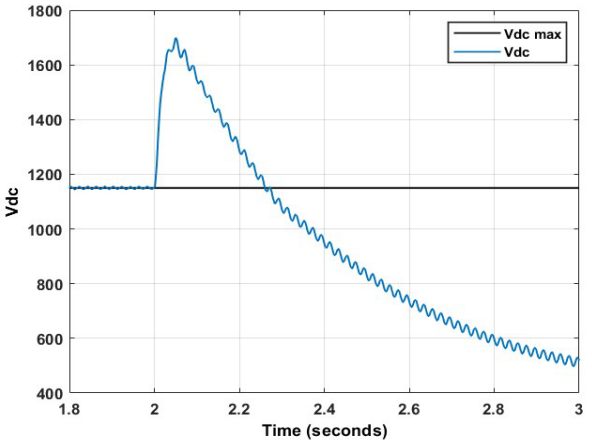


Figure 4.18: DC voltage transient using AVOA at 90% dip.

From Figure 4.15, it is observed that the transient rotor current when applying PSO is 9612 A, with a settling time of 70 ms. Considering the DC voltage when utilizing PSO in Figure 4.16, the transient voltage is 1860 A. The settling time of this voltage is 555 ms. From Figure 4.17, it is observed that the transient rotor current when applying AVOA is 9580 A, with a settling time of 70 ms. Considering the DC voltage when utilizing AVOA in Figure 4.18, the transient voltage is 1700 A. The settling time of this voltage is 275 ms. It is therefore noted that for a 90% voltage dip, the AVOA produced the best results. While the settling time of the rotor current was identical, the AVOA exhibited superiority with regards to the rotor current transient, DC voltage transient, and DC voltage settling time. This is in the order of 0.33%, 8.6%, and a massive 101.8%, respectively. The results of

the experiments conducted are summarized in Table 4.1. From this, it can be concluded that for moderately severe symmetrical voltage dips, PSO is the preferred optimization technique. However, for voltage dips below 80%, the AVOA exhibits far more superior results.

Table 4.1: Summary of results for optimization of crowbar resistance using swarm intelligence.

		$I_{r(transient)}(A)$	$t_{r(settling)}(ms)$	$V_{dc(transient)}$ (V)	$t_{dc(settling)}(ms)$
70%	PSO	**6468**	**70**	**1465**	**175**
	AVOA	6472	**70**	**1465**	**175**
80%	PSO	8035	**70**	**1612**	**260**
	AVOA	**7999**	**70**	1630	280
90%	PSO	9612	**70**	1860	555
	AVOA	**9580**	**70**	**1700**	**275**

4.5. Investigation into African Vultures Optimization Algorithm for the Optimization of the Linear Quadratic Regulator and Demagnetizing Current Injection Gains

As observed in Section 4.4, the crowbar protection scheme successfully provides protection to the DFIG. However, this requires the DFIG to be disconnected from the grid, and therefore is not compliant with modern grid codes. This section aims to analyze the results of the linear quadratic regulator and demagnetizing current injection (DCI) applied to the DFIG under symmetrical grid faults. To enhance the tracking capability of the LQR, an integrator is added to the control mechanism. The LQR and DCI controller gains were optimized using the African vulture's optimization algorithm and compared to the conventional PI controller optimized via the well-known PSO. The cost function utilized was the integral time absolute error. The choice of power converter utilized for switching was the two-level voltage source converter. Three voltages dips were considered: 70%, 80%, and 90%. The results were analyzed in terms of peak rotor current and DC voltage, as well as steady state ripple and steady-state error of the reactive-power controlling rotor current. During the fault, the amount of reactive power injected into the grid is increased, whereas the stator active power reference is set to zero. The steady-state error is evaluated for the last 0.3 s of the simulation. This is to allow the system sufficient time to dampen oscillations. The African vulture's optimization algorithm was subject to 20 iterations of 20 search agents, with an upper and lower bound of 0 and 100, respectively, for the LQR gains, and 0 and 50, respectively, for the DCI controller gains.

4.5.1. Case 2a: Voltage Dip of 70%

Table 4.2 indicates the optimized gains for the PI controller, considering a voltage dip of 70%. As evident, the gain values are high, indicating possible overshoot of the system. This is expected, as there is no way to eliminate the overshoot from the system, only mitigate. Figures 4.19 and 4.20 represent the total rotor current magnitude and DC voltage, respectively, of the PI controller. Figures 4.21 and 4.22 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the PI controller. From Figure 4.19, the peak rotor current magnitude is observed to be at 6256 A, whereas figure 4.20 depicts the peak DC voltage as 1634 V. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.21, is 283 A, with the average steady state value in Figure 4.22 at 1409 A. This correlates to an error percentage of 6.07%

Table 4.2: PI controller gains for 70% dip.

$K_{p(idr)}$	$K_{i(idr)}$	$K_{p(iqr)}$	$K_{i(iqr)}$
53.5253	91.1925	54.9872	48.5211

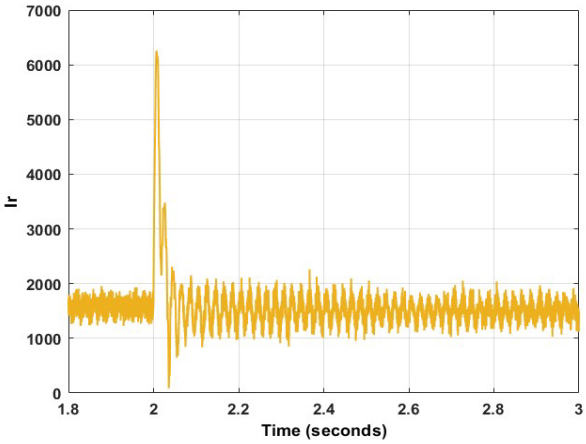


Figure 4.19: Rotor current transient using PI at 70% dip.

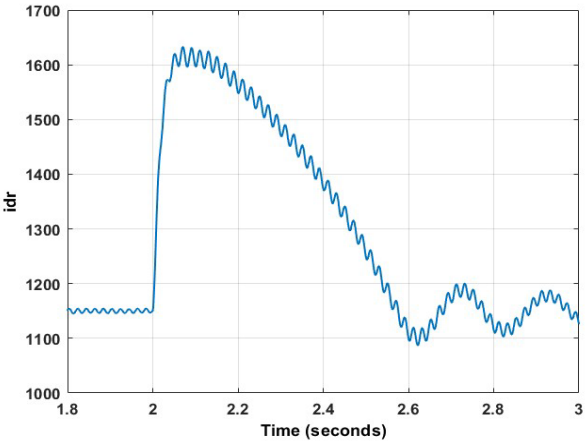


Figure 4.20: DC voltage transient using PI at 70% dip

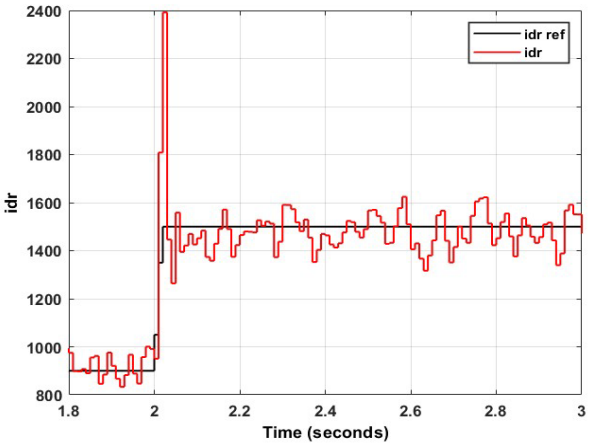


Figure 4.21: Rotor direct axis current ripple using PI at 70% dip

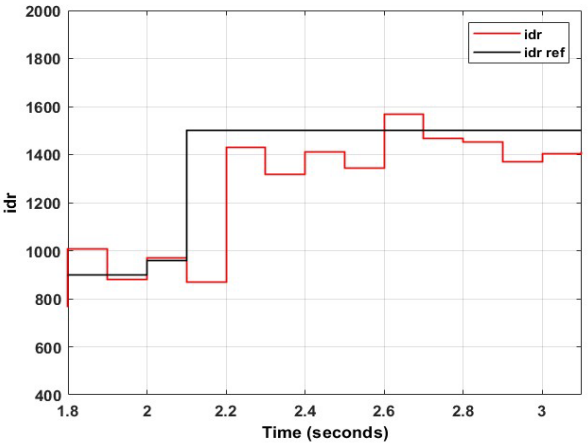


Figure 4.22: Rotor direct axis current average using PI at 70% dip.

Table 4.3 indicates the optimized gains for the demagnetizing current injection (DCI), considering a voltage dip of 70%. Figures 4.23 and 4.24 represent the total rotor current magnitude and DC voltage, respectively, of the DCI. Figures 4.25 and 4.26 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the DCI. From Figure 4.23, the peak rotor current magnitude is observed to be 5557 A, indicating a superiority of 12.58% over the conventional PI controller. Figure 4.24 depicts the peak DC voltage as 1597 V, exhibiting a 2.32% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.25, is 379.5 A, which is 33.92% inferior to the PI controller. The average steady state value in Figure 4.26 is 1355.5 A. This correlates to an error percentage of 6.89%. This error is minorly inferior to the conventional PI controller.

Table 4.3: DCI controller gains at 70% dip.

K_1	K_2
49.2779	48.4913

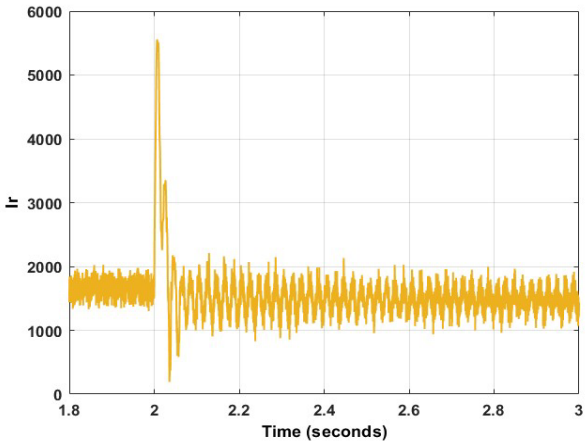


Figure 4.23: Rotor current transient using DCI at 70% dip.

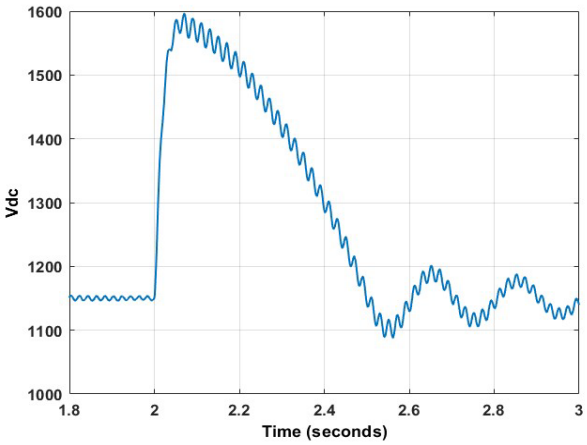


Figure 4.24: DC voltage transient using DCI at 70% dip.

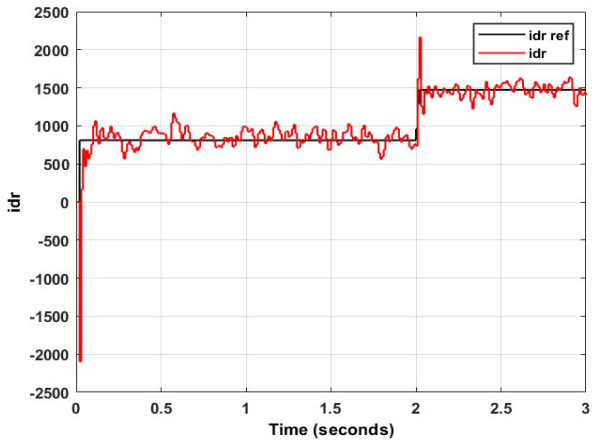


Figure 4.25: Rotor direct axis current ripple using DCI at 70% dip

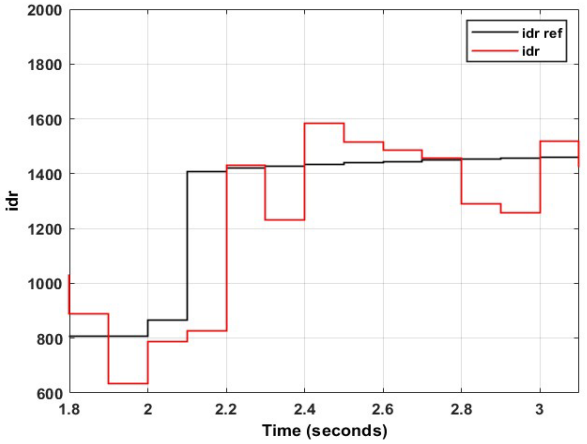


Figure 4.26: Rotor direct axis current average using DCI at 70% dip.

Table 4.4 indicates the optimized gains for the LQR, considering a voltage dip of 70%. Figures 4.27 and 4.28 represent the total rotor current magnitude and DC voltage, respectively, of the LQR. Figures 4.29 and 4.31 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the LQR. From Figure 4.27, the peak rotor current magnitude is observed to be 3735 A, indicating a superiority of 67.5% over the conventional PI controller. Figure 4.28 depicts the peak DC voltage as 1377 V, exhibiting a 18.66% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.29, is 256.5 A, which is 10.33% superior to the PI controller. The average steady state value in Figure 4.30 is 1409 A. This correlates to an error percentage of 10.07%. This error is inferior to the conventional PI controller.

Table 4.4: LQR controller gains at 70% dip.

K_1	K_2	K_3	K_4
100	34.9938	51.29618	7.437115

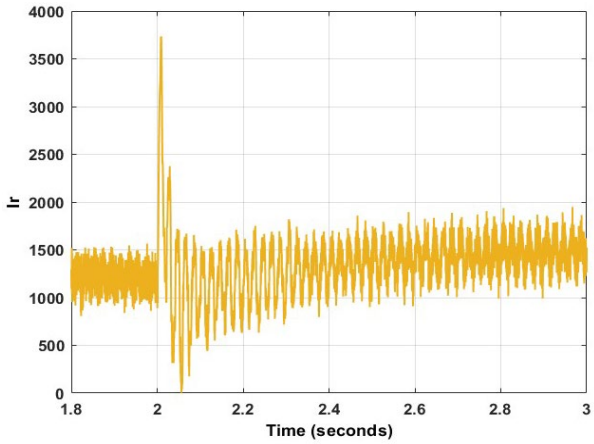


Figure 4.27:Rotor current transient using LQR at 70% dip.

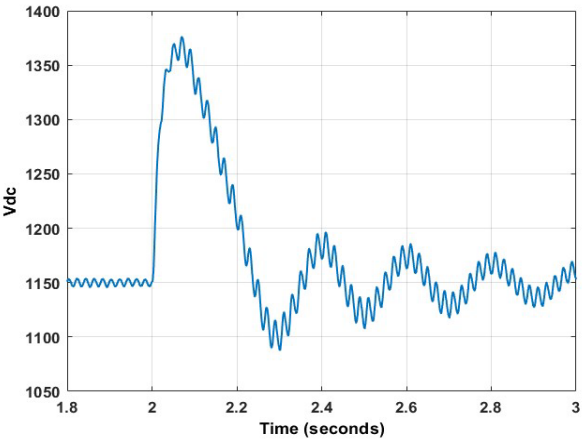


Figure 4.28: DC voltage transient using LQR at 70% dip.

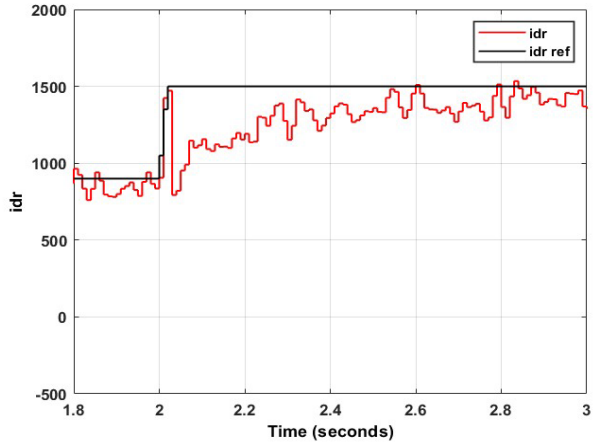


Figure 4.29: Rotor direct axis current ripple using LQR at 70% dip

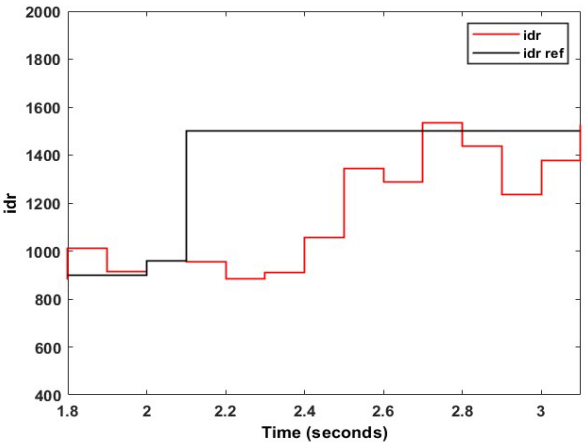


Figure 4.30: Rotor direct axis current average using LQR at 70% dip.

4.5.2. Case 2b: Voltage Dip of 80%

Table 4.5 indicates the optimized gains for the PI controller, considering a voltage dip of 80%. As evident, $K_{i(idr)}$ is zero. This is because the algorithm attempts to search for a solution within the negative domain. However, since the search parameter is set at zero, the algorithm gets trapped at this point. Figures 4.31 and 4.32 represent the total rotor current magnitude and DC voltage, respectively, of the PI controller. Figures 4.33 and 4.34 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the PI controller. From Figure 4.31, the peak rotor current magnitude is observed to be 7550 A, whereas Figure 4.32 depicts the peak DC voltage as 1839 V. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.33, is 456 A, with the average steady state value in Figure 4.34 at 1407 A. This correlates to an error percentage of 6.2%.

Table 4.5: PI controller gains for 80% dip.

$K_{p(idr)}$	$K_{i(idr)}$	$K_{p(iqr)}$	$K_{i(iqr)}$
62.9078	0	65.535	12.8299

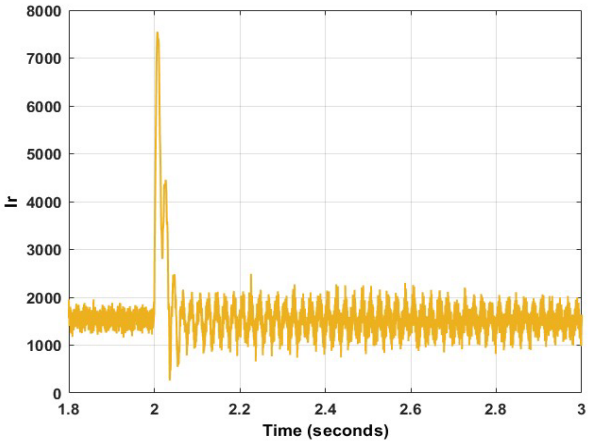


Figure 4.31: Rotor current transient using PI at 80% dip.

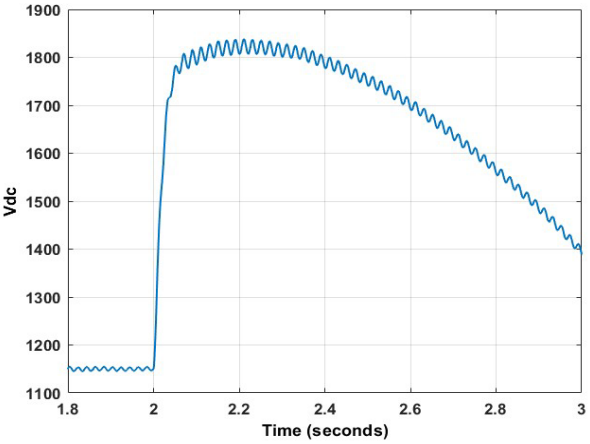


Figure 4.32: DC voltage transient using PI at 80% dip.

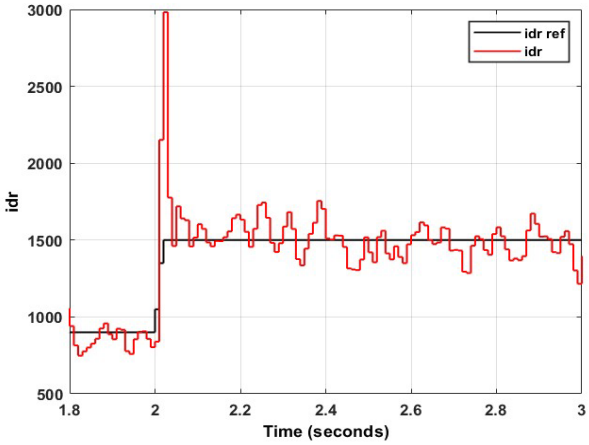


Figure 4.33: Rotor direct axis current ripple using PI at 80% dip

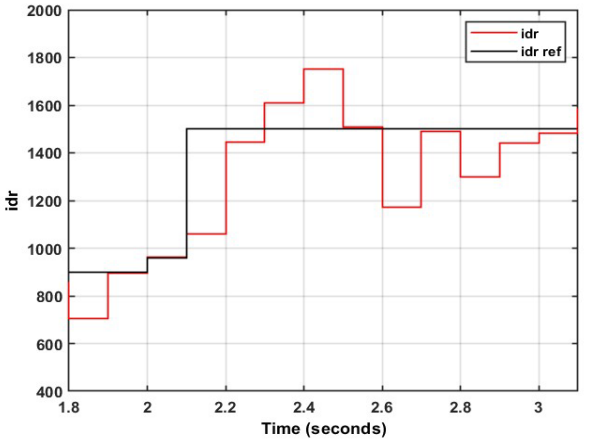


Figure 4.34: Rotor direct axis current average using PI at 80% dip.

Table 4.6 indicates the optimized gains for the DCI, considering a voltage dip of 80%. Figures 4.35 and 4.36 represent the total rotor current magnitude and DC voltage, respectively, of the DCI. Figures 4.37 and 4.38 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the DCI. From Figure 4.35, the peak rotor current magnitude is observed to be 7325 A, indicating a superiority of 52.4% over the conventional PI controller. Figure 4.36 depicts the peak DC voltage as 1787 V, exhibiting a 20.35% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.37, is 357.5 A, which is 27.55% superior to the PI controller. The average steady state value in Figure 4.38 is 1173.67 A. This correlates to an error percentage of 19.74%. This error is superior to the conventional PI controller

Table 4.6: DCI controller gains at 80% dip.

K_1	K_2
49.9272	45.3424

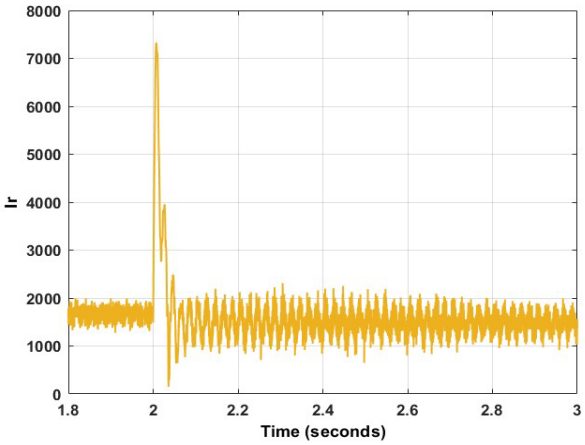


Figure 4.35: Rotor current transient using DCI at 80% dip

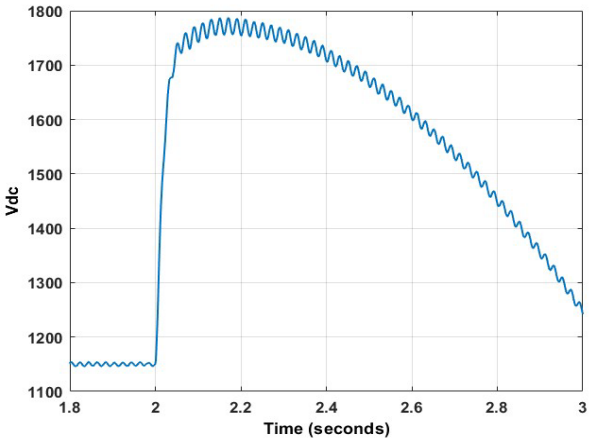


Figure 4.36: DC voltage transient using DCI at 80% dip.

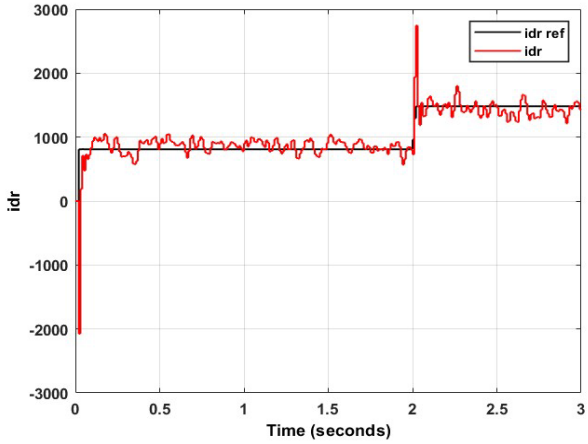
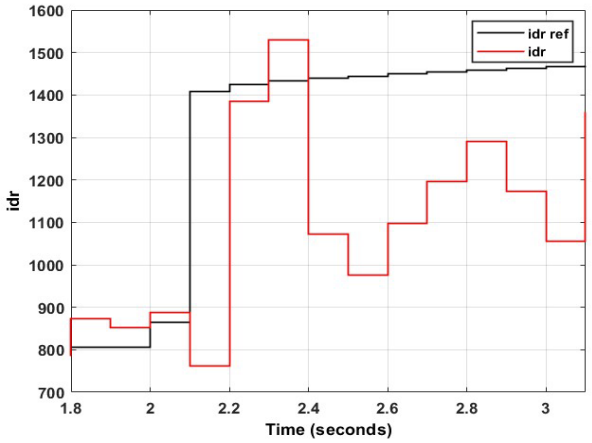


Figure 4.37: Rotor direct axis current ripple using DCI at 80% dip



.Figure 4.38: Rotor direct axis current average using DCI at 80% dip.

Table 4.7 indicates the optimized gains for the LQR, considering a voltage dip of 80%. Figures 4.39 and 4.40 represent the total rotor current magnitude and DC voltage, respectively, of the LQR. Figures 4.41 and 4.42 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the LQR. From Figure 4.39, the peak rotor current magnitude is observed to be 4954 A, indicating a superiority of 52.4%4.42ver the conventional PI controller. Figure 4.40 depicts the peak DC voltage as 1528 V, exhibiting a 20.35% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.41, is 290 A, which is 57.24% superior to the PI controller. The average steady state value in Figure 44 is 1427 A. This correlates to an error percentage of 4.87%. This error is superior to the conventional PI controller, which can be attributed to the zero-gain value in Table 5, which was obtained for the PI controller.

Table 4.7: LQR controller gains at 80% dip.

K_1	K_2	K_3	K_4
99.7437	37.8219	45.7003	6.71573

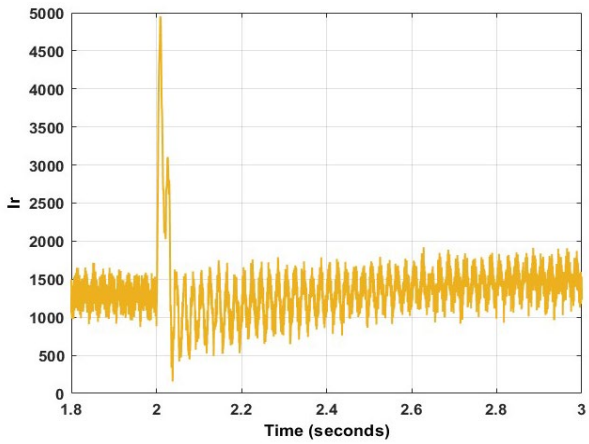


Figure 4.39: Rotor current transient using LQR at 80% dip.

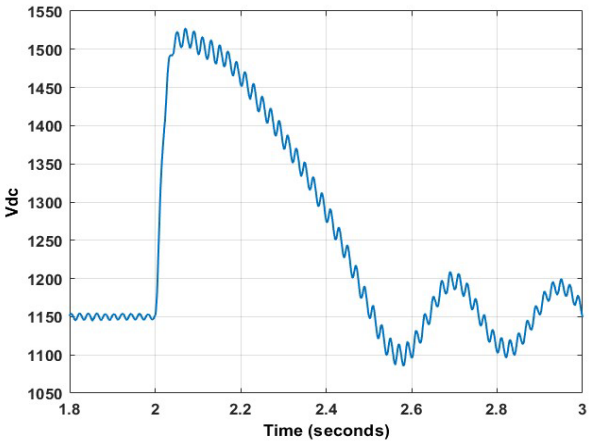


Figure 4.40: DC voltage transient using LQR at 80% dip.

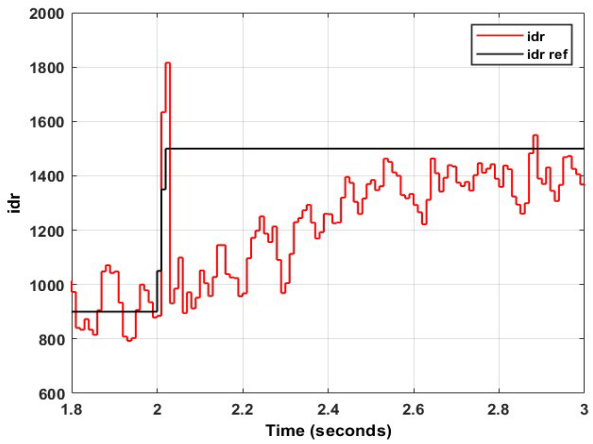


Figure 4.41: Rotor direct axis current ripple using LQR at 80% dip

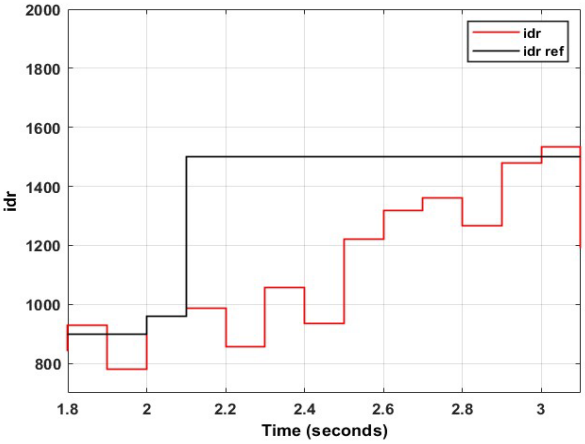


Figure 4.42: Rotor direct axis current average using LQR at 80% dip.

4.5.3. Case 2b: Voltage Dip of 80%

Table 4.8 indicates the optimized gains for the PI controller, considering a voltage dip of 90%. Figures 4.43 and 4.44 represent the total rotor current magnitude and DC voltage, respectively, of the PI controller. Figures 4.45 and 4.46 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the PI controller. From Figure 4.43, the peak rotor current magnitude is observed to be 9110 A, whereas Figure 4.44 depicts the peak DC voltage as 2176 V. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.45, is 416 A, with the average steady state value in Figure 4.46 at 1438 A. This correlates to an error percentage of 4.3%.

Table 4.8: PI controller gains at 90% dip.

$K_{p(idr)}$	$K_{i(idr)}$	$K_{p(iqr)}$	$K_{i(iqr)}$
20.7428	20.4191	19.4448	80.3962

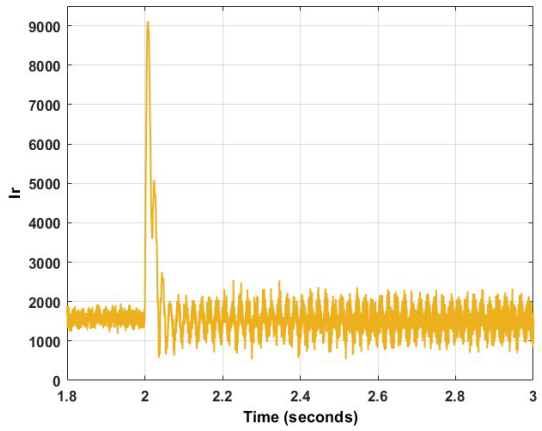


Figure 4.43: Rotor current transient using PI at 90% dip.

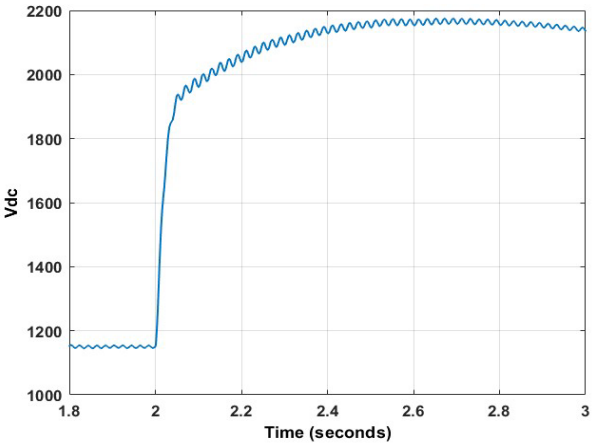


Figure 4.44: DC voltage transient using PI at 90% dip.

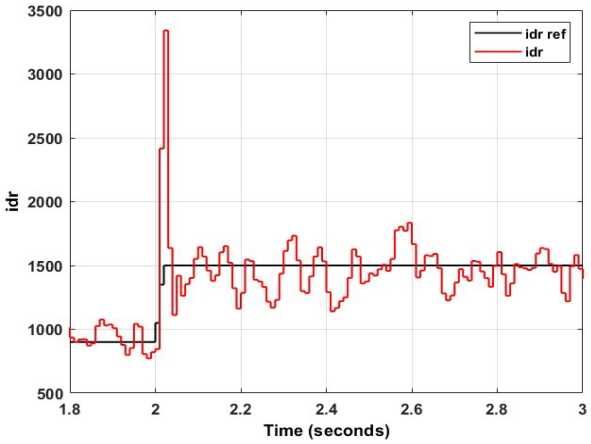


Figure 4.45: Rotor direct axis current ripple using PI at 90% dip.

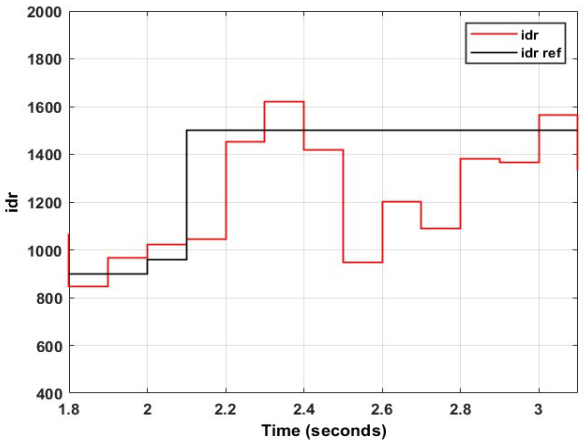


Figure 4.46:Rotor direct axis current average using PI at 90% dip.

Table 4.9 indicates the optimized gains for the DCI, considering a voltage dip of 90%. Figures 4.47 and 4.48 represent the total rotor current magnitude and DC voltage, respectively, of the DCI. Figures 4.49 and 4.50 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the DCI. From Figure 4.47, the peak rotor current magnitude is observed to be 8679 A, indicating a superiority of 25.74% over the conventional PI controller. Figure 4.48 depicts the peak DC voltage as 2137 V, exhibiting a 18.65% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.49, is 667.5 A, which is 60.46% inferior to the PI controller. The average steady state value in Figure 4.50 is 1230.17 A. This correlates to an error percentage of 16.42%. This error is inferior to the conventional PI controller.

Table 4.9: DCI controller gains at 90% dip.

K_1	K_2
100	24.3195

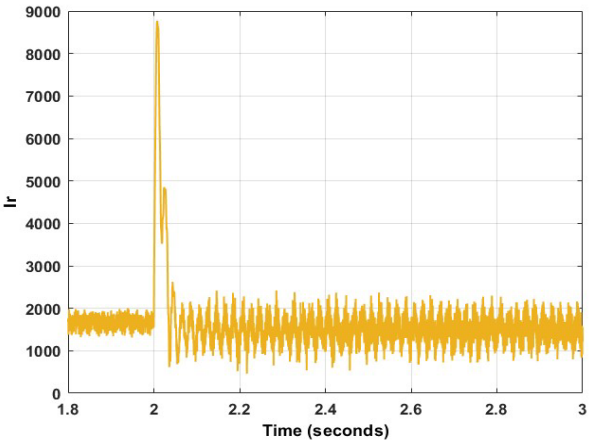


Figure 4.47: Rotor current transient using DCI at 90% dip

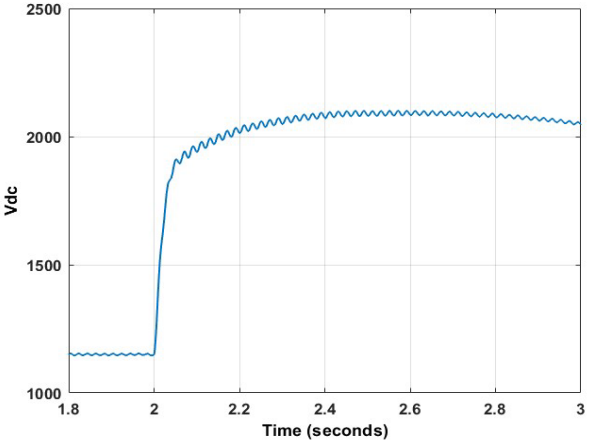


Figure 4.48: DC voltage transient using DCI at 90% dip.

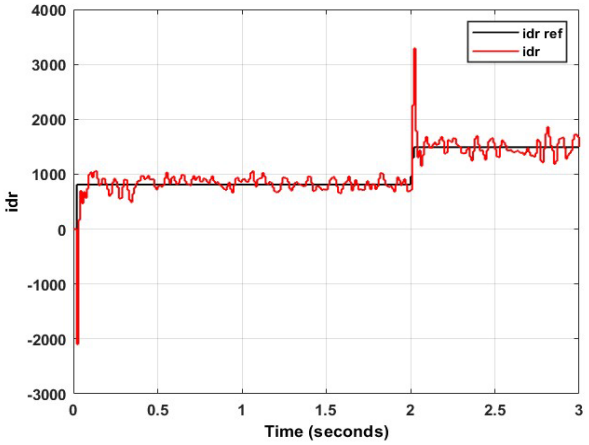


Figure 4.49: Rotor direct axis current ripple using DCI at 90% dip.

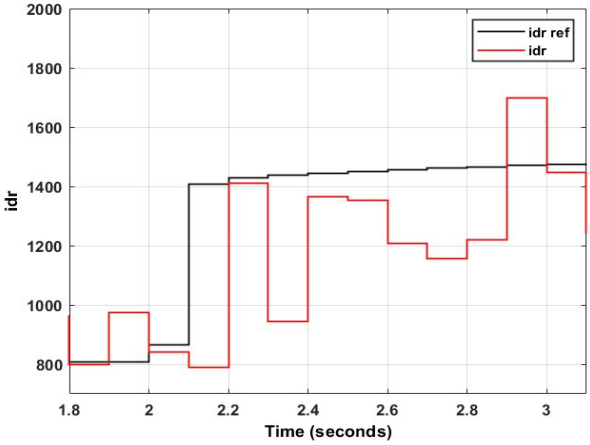


Figure 4.50: Rotor direct axis current average using DCI at 90% dip

Table 4.10 indicates the optimized gains for the LQR, considering a voltage dip of 90%. Figures 4.51 and 4.52 represent the total rotor current magnitude and DC voltage, respectively, of the LQR. Figures 4.53 and 4.54 indicate the rotor direct axis current steady state ripple and steady state error, respectively, of the LQR. From Figure 4.51, the peak rotor current magnitude is observed to be 7245 A, indicating a superiority of 25.74% over the conventional PI controller. Figure 4.52 depicts the peak DC voltage as 1834 V, exhibiting a 18.65% better response than the PI controller. The steady-state ripple of the rotor direct axis current, as shown in Figure 4.53, is 422 A, which is 1.4% inferior to the PI controller. The average steady state value in Figure 4.54 is 1172 A. This correlates to an error percentage of 21.87%. This error is inferior to the conventional PI controller.

Table 4.10: LQR controller gains at 90% dip.

K_1	K_2	K_3	K_4
100	24.3195	44.45006	7.244542

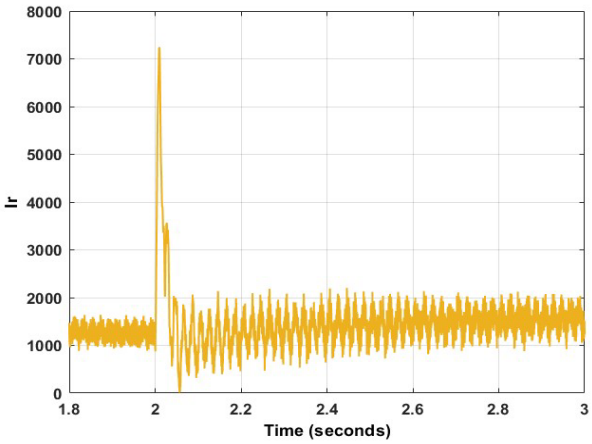


Figure 4.51: Rotor current transient using LQR at 90% dip.

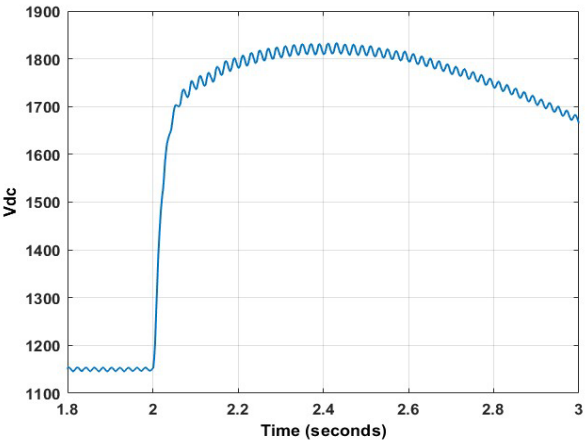


Figure 4.52: DC voltage transient using LQR at 90% dip.

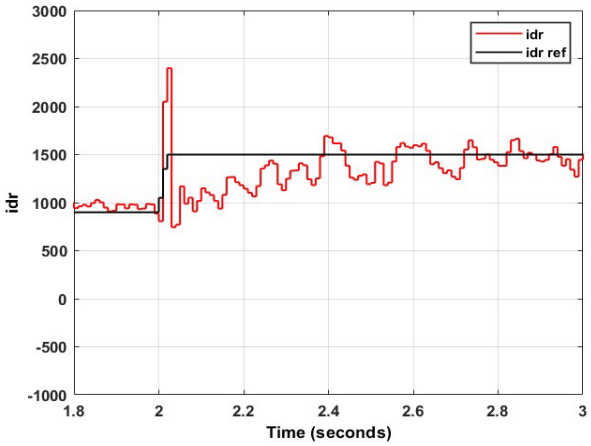


Figure 4.53: Rotor direct axis current ripple using LQR at 90% dip

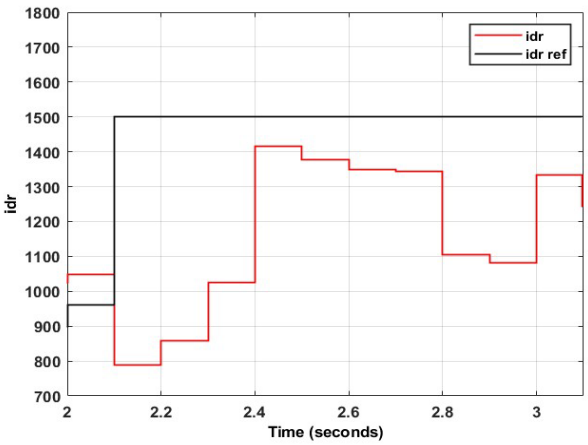


Figure 4.54: Rotor direct axis current average using LQR at 90% dip.

The results of the investigation conducted are summarized in Table 4.11. As evident in Table 4.11, the LQR, coupled with a strong metaheuristic optimization technique, has obtained significantly produced better results than the conventional PI controller on most of the occasions. Superiority in terms of rotor current peak, DC voltage peak, and steady-state ripple were in the order of 67.5%, 20.35, and 57.24%, respectively. In the case of steady-state error, the PI controller seemed to produce better results. This is expected, as the LQR is not well-suited for tracking problems. Comparison between the LQR and DCI methods, it is observed that the latter surpasses the former on only just occasions. These are for the steady-state error at 70% and 90% voltage dips. Once again, since the DCI method utilized PI controllers, the outcome is expected.

Table 4.11: Summary of results for PI, DCI and LQR applied to DFIG under symmetrical voltage dip.

		Rotor Current Peak (A)	DC Voltage Peak (V)	Rotor Direct Axis Current Steady-State Ripple (A)	Rotor Direct Axis Current Steady-State Error (%)
70% dip	PI	6256	1634	283	**6.07**
	DCI	5557	1597	379.5	6.89
	LQR	**3735**	**1377**	**256.5**	10.07
80% dip	PI	7550	1839	456	6.2
	DCI	7325	1787	357.5	19.74
	LQR	**4954**	**1528**	**290**	**4.87**
90% dip	PI	9110	2176	**416**	**4.3**
	DCI	8679	2137	667.5	16.42
	LQR	**7245**	**1834**	422	21.87

4.6. Conclusion

This work presented an investigation into the utilization of metaheuristic optimization techniques for the control of the doubly fed induction generator under the influence of symmetrical voltage dips. Two swarm intelligence techniques were proposed; the well-known particle swarm optimization, and the recently developed African vulture's optimization algorithm. These techniques were first utilized to obtain the optimal crowbar resistance for protection of the DFIG. Next, the AVOA was applied to the linear quadratic regulator for low voltage ride through of the DFIG. These results were compared to the conventional PI controller, optimized via PSO. Considering the crowbar resistance investigation, it was observed that for moderately severe symmetrical voltage dips, PSO exhibited superior results to AVOA in terms of transient magnitude, albeit the settling times were nearly identical. For severe faults (less than 80%), the AVOA exhibited dominance over PSO. This was in terms of transient magnitude, as well as settling time.

For the LVRT experiment, it was observed that the linear quadratic regulator, coupled with the AVOA, proved to be far superior to the PI controller, which was optimized using PSO. This superiority was with respect to transient magnitude and steady state ripple and was seen for almost every fault magnitude tested. However, as expected, despite the addition of the integrator, the LQR was inferior to the PI controller in terms of steady state error. Overall, it was observed that the LQR unanimously produced the best results. To further validate the effectiveness of such a controller, a 3% grid frequency perturbation was applied to the test system. Grid frequency perturbations are common occurrences with symmetrical faults, and it is therefore imperative that control systems do not lose their capabilities in such scenarios. The perturbation was applied for each of the three voltage dip magnitudes, with the results summarized in Table 4.12. As evident from Table 4.12, even when subject to grid frequency perturbations, the LQR still yielded superior rotor current and DC voltage peak magnitudes, when compared to both the PI controller and DCI methods.

Table 4.12: Summary of results for LQR applied subject to 3% grid frequency perturbation.

	Rotor Current Peak (A)	DC Voltage Peak (V)	Rotor Direct Axis Current Steady-State Ripple (A)	Rotor Direct Axis Current Steady-State Error (%)
70% dip	4276	1400	356	12.4
80% dip	5516	1585	316	13.7
90% dip	7706	1873	417	16.8

When comparing the results of the crowbar experiment to the results of the LVRT experiment, it is evident that in all cases, the LQR produced a superior rotor current and DC voltage transient magnitude. However, as observed in the results, the settling time of the crowbar method was unanimously better than that of the LQR. Therefore, it can be concluded that each of these methods offer advantages and disadvantages. When coupled with powerful metaheuristic optimization techniques, these methods can be fine-tuned to produce the best possible results. Future work involves the enhancement of the linear quadratic regulator to achieve a better steady state error. This involves the utilization of the problem-specific transfer function to attempt to achieve a zero steady-state error.

Chapter 5 : Design of dual vector controller for control of the DFIG using swarm intelligence

In chapter four, the effect of applying swarm intelligence to optimal crowbar setting and Low Voltage Ride Through of the DFIG was investigated. The results were positive, indicating the promising capabilities of swarm intelligence for the control of the DFIG under the influence of symmetrical voltage dips. However, most grid voltage anomalies are asymmetrical in nature. The voltage dips (or spikes) are usually not severe in nature but do have the ability to impact the performance of the conventional control system. This chapter serves to analyze the effect of applying swarm intelligence to the control of the DFIG under the influence of asymmetrical voltage dips. Single-phase voltage dips of 5% and 10%, as well as a dual-phase voltage dip of 7.5%, was considered. The control algorithm implemented was the dual vector control method, which is the most widely established method of control for such scenarios. The swarm intelligence technique applied to the system is Particle Swarm Optimization, Bat Algorithm, and Gorilla Troops Optimization Algorithm. The results were measured in terms of rotor direct and quadrature axis current steady state error, overshoot, and steady-state ripple. The chapter starts off by providing a background to the problem being investigated. Thereafter, a control block diagram, which will be implemented, is provided. This is succeeded by a comprehensive analysis of the obtained results. Lastly, the work is concluded.

5.1. Background

Asymmetrical grid conditions are common occurrences in areas where wind farms are situated. This is mostly due to the fact that wind farms are located in areas which are far away from transmission and/or distribution stations. The conventional field-oriented control has demonstrated competency in regulation of stator active and reactive powers. However, the effect of asymmetrical grids deteriorates the performance of such controllers. Under balanced grid conditions, there exists no negative sequence currents. This is not the case in the event of unbalanced grids. Therefore, to obtain normal operating conditions in the case of an unbalanced grid, the negative sequence rotor currents need also be regulated. This can be obtained via the use of PI controllers. In this method, two PI controllers regulate the positive sequence rotor currents (which in turn control the stator active and reactive powers), while another two PI controllers are utilized regulate the negative sequence currents [213]. The positive sequence rotor currents are analyzed in the synchronous rotating reference frame, whereas the negative sequence currents are realized in the inverse rotating reference frame [213]. Owing to the various negative effects of asymmetrical grid voltages, the rotor reference positive sequence currents may vary, depending on the desired target. These targets are:

- Constant stator powers
- Balanced stator currents
- Constant electromechanical torque
- Balanced rotor currents
- Minimum rotor voltage

The most common desired target is balanced rotor currents. Owing to the linear nature between the rotor currents, and stator power, adequate regulator of the rotor direct and quadrature axis currents will ensure acceptable stator power responses. To achieve the target of balanced rotor currents, the negative sequence rotor currents are forced

to zero, whereas the positive sequence rotor currents attempt to regulate the stator active and reactive powers. While achieving the desired result, this method has drawbacks. Firstly, it requires the decomposition of the rotor currents into positive and negative sequence components. This causes considerable time delay and reduces the reliability of the method [213]. Secondly, the use of an additional two PI controllers makes tuning of the controllers a more complex and tedious task. There exists a total of eight parameters to be optimized and attempting this via trial and error may lead to sup-optimal controller performance. The recent success of swarm intelligence in the optimization of control parameters in power systems, makes this a promising option of application to the dual vector control method. The aim of this chapter is to investigate the effect of applying swarm intelligence to the design of the four PI controllers for balanced rotor current control of the DFIG.

5.2. Control structure of dual vector control

As explained in 5.1, the dual vector control method makes use of individual PI controllers, to regulate the rotor direct and quadrature axis currents. The functional block diagram, to be implemented in this chapter, is shown in figure 5.1 [213]. In the figure, it is evident that two reference values are set to zero. This implies the desired target of balanced rotor currents. Note that c1-c4 in the figure refer to coupling terms, which form part of the basics of field-oriented control.

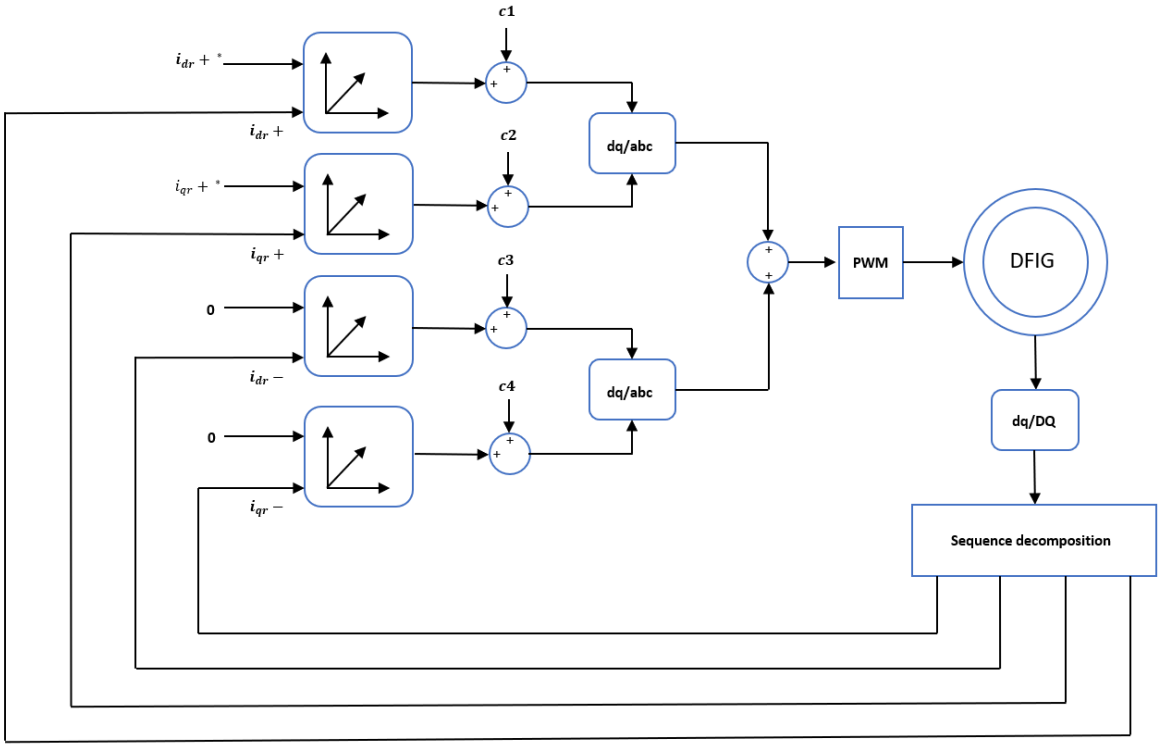


Figure 5.1: Structure of dual-vector control for constant rotor current [213]

5.3. Experimental results and analysis

Three swarm intelligence algorithms were applied to the design of the dual vector control method These are across the timeline of the evolution of such algorithms. The first algorithm to be applied was the well-known particle swarm optimization [55]. Second, the bat algorithm [96], was tested. Lastly, application of the very recent gorilla troops optimizer was evaluated [149]. The control system implemented was stator flux-oriented control. This

means that the stator reactive power was controlled by the rotor direct current, and the stator active power by the rotor quadrature current. The results obtained were evaluated in terms of steady-state error, steady-state ripple, and overshoot. Owing to the direct proportional relationship between the rotor currents and the stator power, the output of the DFIG can be directly analyzed from the waveforms of the rotor currents. These relationships can be observed in [215]. It should be noted that owing to the complexity of the simulation circuit, each iteration takes a while to complete. To ensure that this did not take an unreasonable amount of time to execute, each of the three swarm intelligent algorithms were subject to 20 search agents and 20 iterations. This should suffice, considering that particle swarm optimization, which is the most established and widely used technique, is known to exhibit a fast convergence rate [55]. Rate of convergence is an important aspect when considering metaheuristic optimization techniques; hence, to ensure that this factor was considered, uniformity was applied throughout the experiments. The proportional and integral gains were given an upper limit of 100 and a lower limit of 0. This large controller gain range was chosen arbitrarily and was chosen as such so that the exploitation capability of the metaheuristic optimization technique can be tested. For application of the algorithm, the DFIG was operated at a speed of 150 rad/s. The simulation was carried out on MATLAB/Simulink, and the simulation was run for 2 s. Single-phase and dual-phase voltage dips were investigated. The voltage dips considered were at 5%, 7.5% and 10%. To analyze the steady-state error, the output was obtained in intervals of 0.1 s. To analyze the overshoot and steady-state ripple, the output was obtained in intervals of 0.01 s. To ensure that the system was given sufficient time to reach steady state, the results were analyzed from 1.5 to 2 s. The rotor direct and quadrature current references were kept constant at 1434.6 A and 1281 A, respectively.

5.3.1. Case a: Voltage Unbalance of 5%

Upon application of the three swarm intelligence algorithms to the dual vector controller for a voltage unbalance of 5%, the resultant PI controller gains for the four controllers can be observed in Table 5.1. Figures 5.2–5.11 depict the results of applying PSO to the controller design. Figures 5.2 and 5.3 depict the three-phase stator voltage and current, respectively, with figures 5.4 and 5.5 displaying the three-phase rotor voltage and current, respectively. Figures 5.6 and 5.7 show the rotor direct and quadrature axis negative sequence currents. Figures 5.8 and 5.10 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas Figures 5.9 and 5.11 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

Table 5.1: Controller gains for 5% dip using dual vector control for each swarm technique

	Gain range	PSO	BA	GTO
$\boldsymbol{K_{p1(idr+)}}$	0-100	97.4531	99.0016	0.60363
$\boldsymbol{K_{i1(idr+)}}$	0-100	9.45923	41.4079	0.015496
$\boldsymbol{K_{p2(iqr+)}}$	0-100	3.43091	18.7909	0.16635
$\boldsymbol{K_{i2(iqr+)}}$	0-100	76.3631	21.987	0
$\boldsymbol{K_{p3(idr-)}}$	0-100	72.2922	9.95427	2.1047
$\boldsymbol{K_{i3(idr-)}}$	0-100	98.6869	31.6091	0
$\boldsymbol{K_{p4(iqr-)}}$		66.5793	60.0975	0.98995
$\boldsymbol{K_{i4(iqr-)}}$		18.2864	69.0138	0

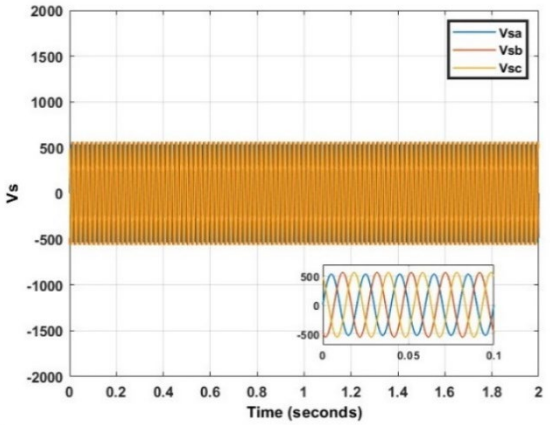


Figure 5.2: Three-phase stator voltage at 5% dip

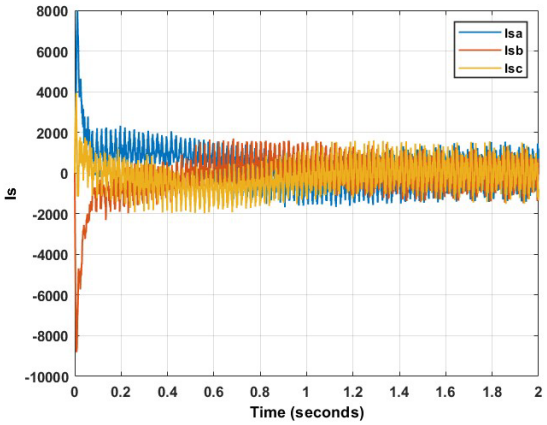


Figure 5.3: Three-phase stator current using PSO at 5% dip

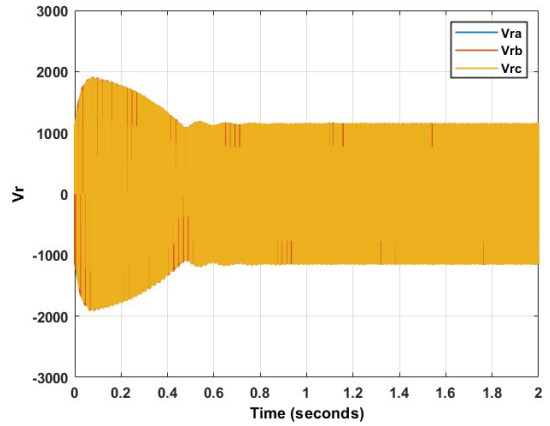


Figure 5.4: Three-phase rotor voltage using PSO at 5% dip

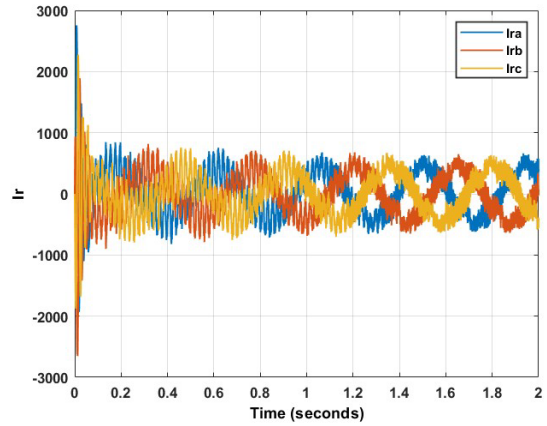


Figure 5.5: Three-phase rotor current using PSO at 5% dip

From Figure 5.8, the average rotor direct steady-state current error percentage is 4.99%, which is strong. The current ripple, as shown in figure 5.9, is large at 315 A. This is somewhat mitigated by the fact that the system is underdamped, thereby providing a 0% overshoot. The rotor quadrature current exhibits a steady-state error of 70.73%, as evident by figure 5.10. Furthermore, the steady-state ripple, as depicted by figure 5.11 is once again large at 401 A. This is somewhat mitigated by the overshoot, which is 41.69%.

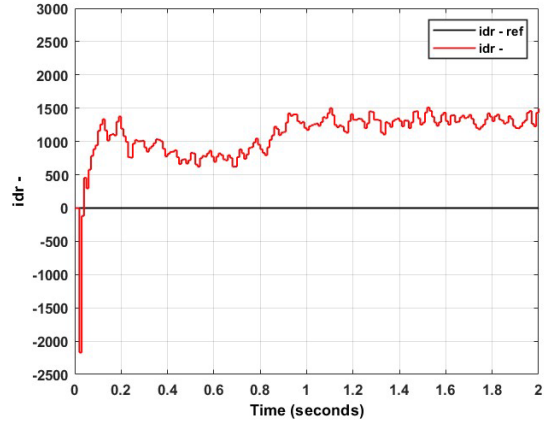


Figure 5.6: Overshoot and ripple of i_{dr} – using PSO at 5% dip

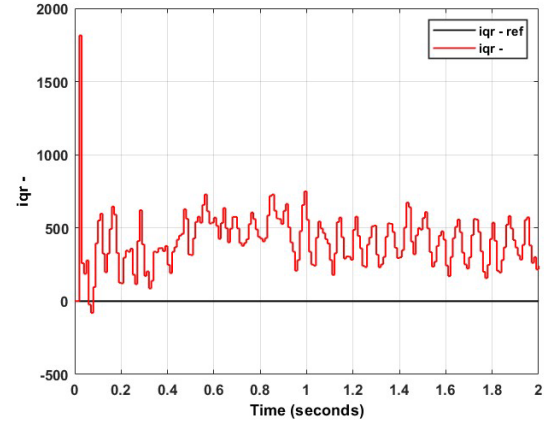


Figure 5.7: Response of i_{qr} – using PSO at 5% dip

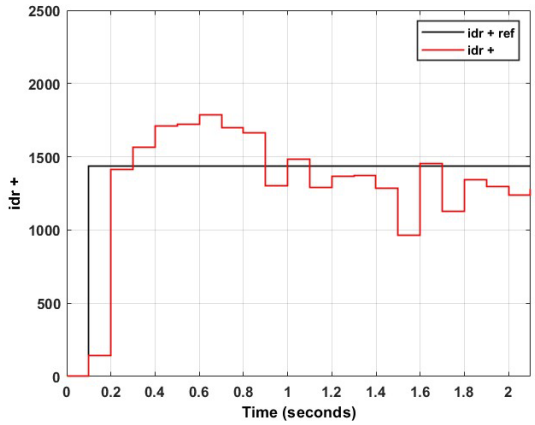


Figure 5.8: Steady-state error of $i_{dr}+$ using PSO at 5% dip

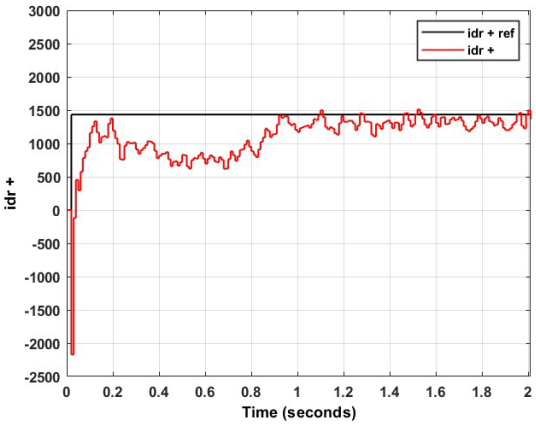


Figure 5.9: Overshoot and ripple of $i_{dr}+$ using PSO at 5%

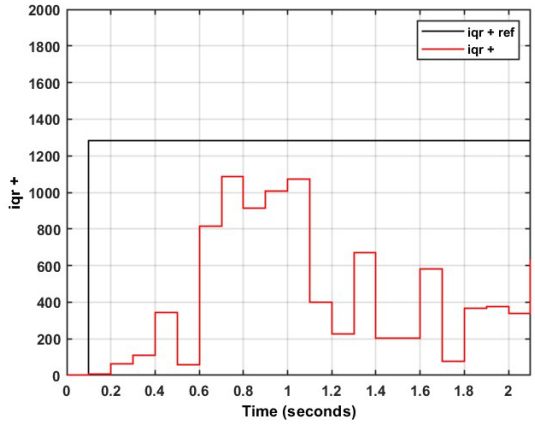


Figure 5.10: Steady-state error of $i_{qr}+$ using PSO at 5% dip

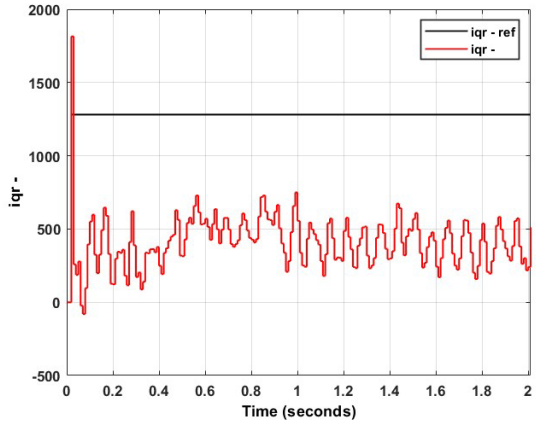


Figure 5.11: Steady-state error of $i_{qr}+$ using PSO at 5% dip

Figures 5.12–5.20 depict the results of applying BA to the controller design. Figures 5.2 and 5.12 depict the three-phase stator voltage and current, respectively, with figures 5.13 and 5.14 displaying the three-phase rotor voltage and current, respectively. Figures 5.15 and 5.16 show the rotor direct and quadrature axis negative sequence currents. Figures 5.17 and 5.19 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.18 and 5.20 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

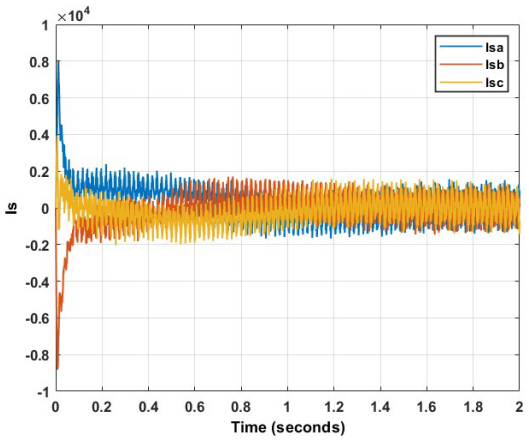


Figure 5.12: Three-phase stator current using BA at 5% dip

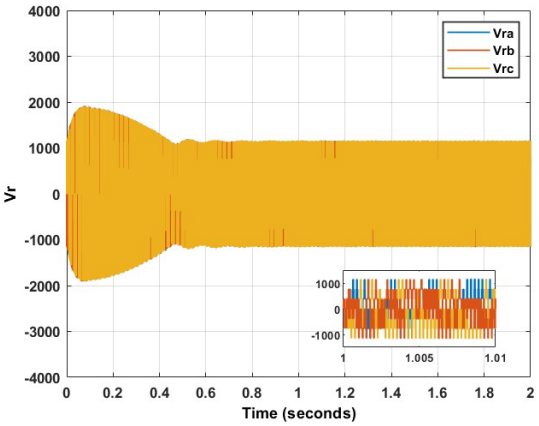


Figure 5.13: Three-phase rotor voltage using BA at 5% dip

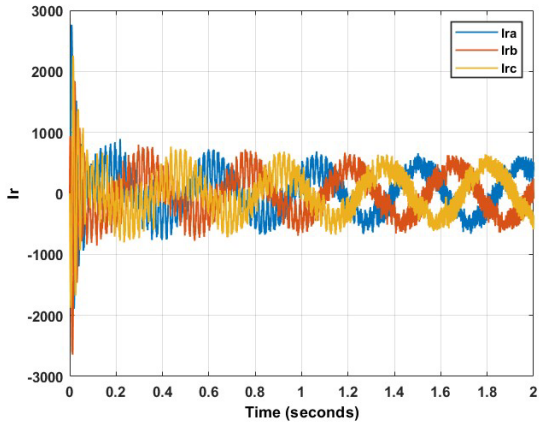


Figure 5.14: Three-phase rotor current using BA at 5% dip

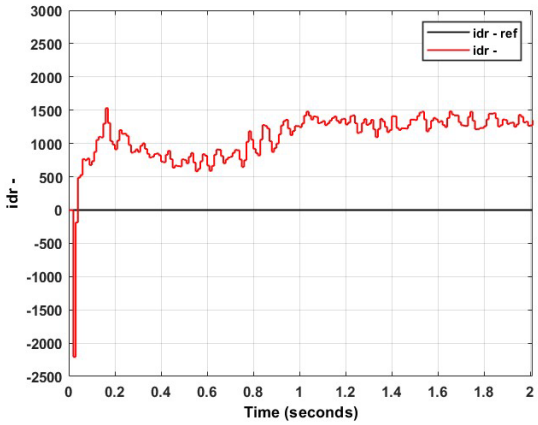


Figure 5.15: Response of $i_{dr}-$ using BA at 5% dip

From Figure 5.17, the average rotor direct steady-state current error percentage is 4.42%, which is strong. The current ripple, as shown in figure 5.18, is large at 333 A. This is somewhat mitigated by the minute overshoot, which stands at a mere 6.72%. The rotor quadrature current exhibits a steady-state error of 76.17%, as evident by figure 5.19. This is unacceptable. Furthermore, the steady-state ripple, as depicted by figure 5.20 is once again large at 539 A. This is somewhat mitigated by the overshoot, which is 43.01%.

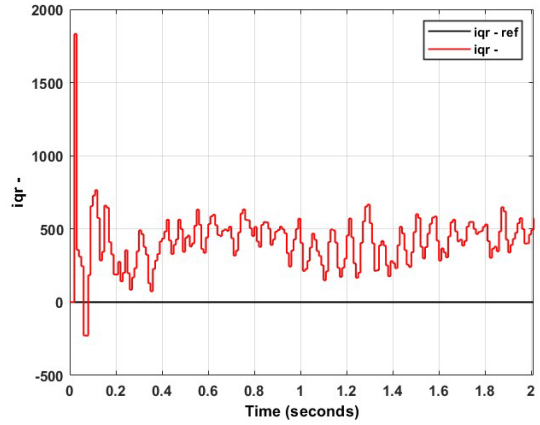


Figure 5.16: Response of $i_{qr}-$ using BA at 5% dip

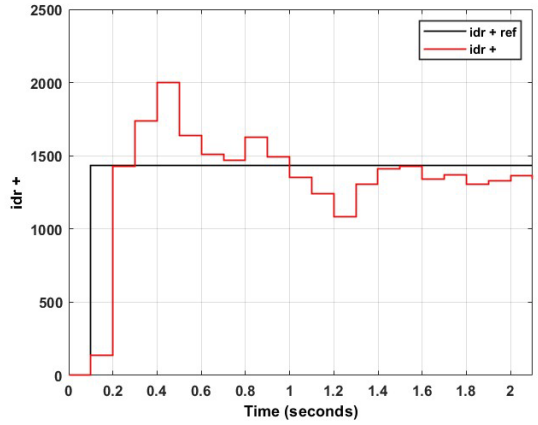


Figure 5.17: Steady-state error of $i_{dr}+$ using BA at 5% dip

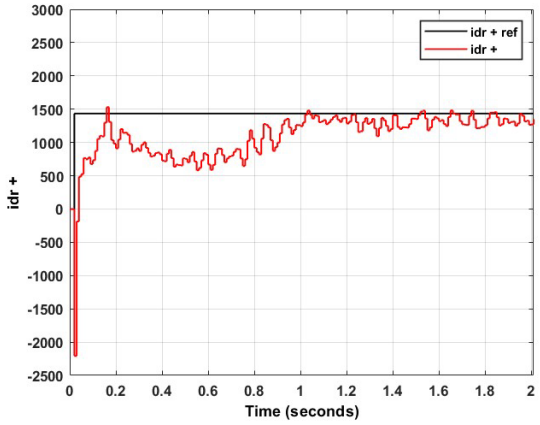


Figure 5.18: Overshoot and ripple of $i_{dr}+$ using BA at 5% dip

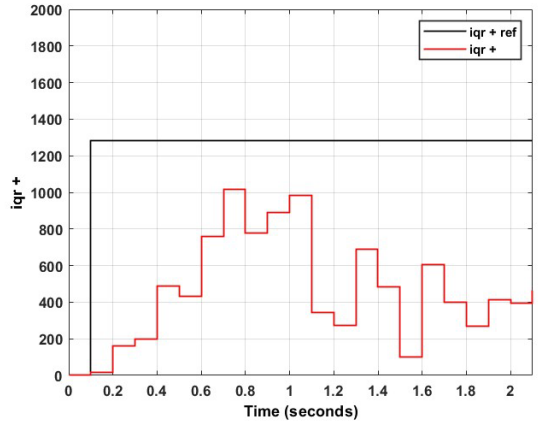


Figure 5.19 : Steady-state error of $i_{qr}+$ using BA at 5% dip

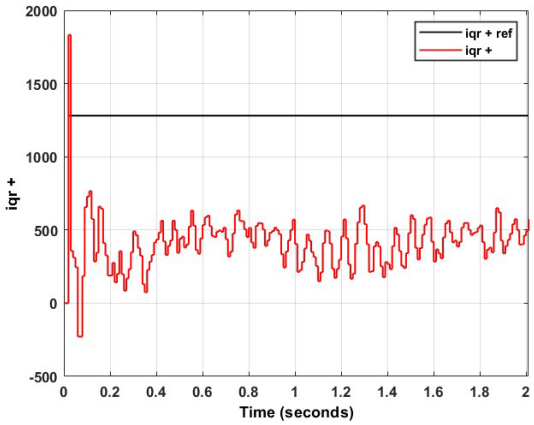


Figure 5.20: Overshoot and ripple of $i_{qr}+$ using BA at 5% dip

Figures 5.21–5.29 depict the results of applying GTO to the controller design. Figures 5.2 and 5.21 depict the three-phase stator voltage and current, respectively, with Figures 5.22 and 5.23 displaying the three-phase rotor voltage and current, respectively. Figures 5.24 and 5.25 show the rotor direct and quadrature axis negative sequence currents. Figures 5.26 and 5.28 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.27 and 5.29 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

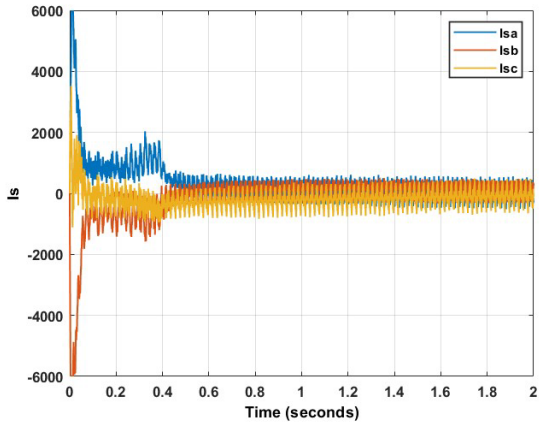


Figure 5.21: Three-phase stator current using GTO at 5% dip

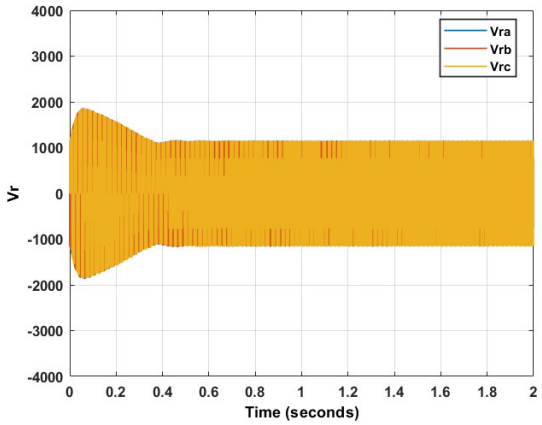


Figure 5.22: Three-phase rotor voltage using GTO at 5% dip

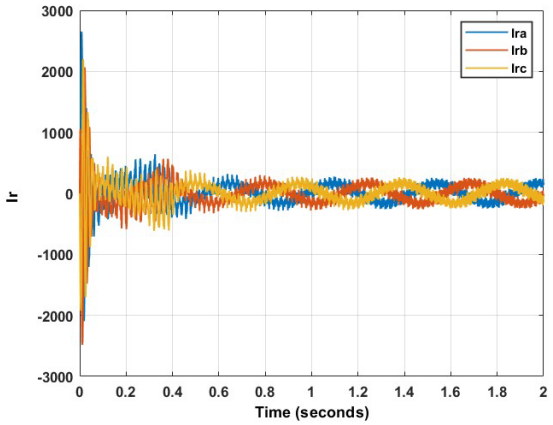


Figure 5.23: Three-phase rotor current using GTO at 5% dip

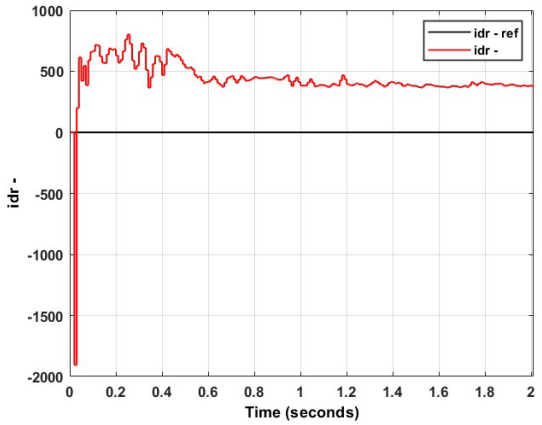


Figure 5.24: Response of $i_{dr}-$ using GTO at 5% dip

From Figure 5.26, the average rotor direct steady-state current error percentage is 65.87%, which is unacceptable. The current ripple, as shown in figure 5.27, is exceptional at 39 A. The overshoot is somewhat acceptable and stands at a 63.6%. The rotor quadrature current exhibits a steady-state error of 86.9%, as evident by figure 5.28. This is once again unacceptable. The steady-state ripple, as depicted by figure 5.29 is once again minute at 58 A. This is complemented by the overshoot, which is strong at 20.22%. The results of the experiments are summarized in table 5.2. As evident, based on the various performance parameters, the best results are split among the various techniques. However, according to rankings, PSO produced the overall best performance, with BA exhibiting the worst results.

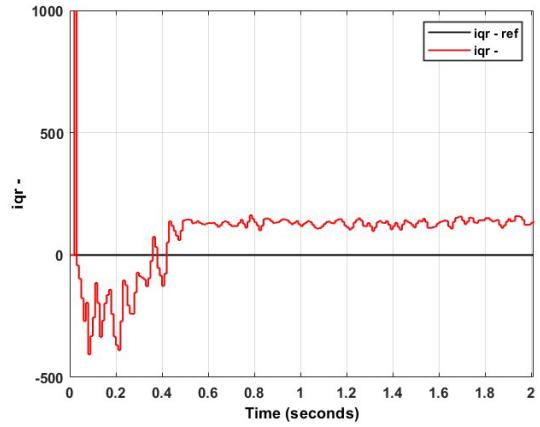


Figure 5.25: Response of $i_{qr}-$ using BA at 5% dip

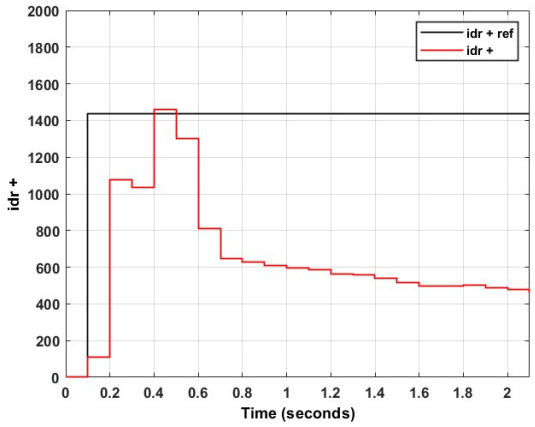


Figure 5.26: Steady-state error of $i_{dr}+$ using GTO at 5% dip

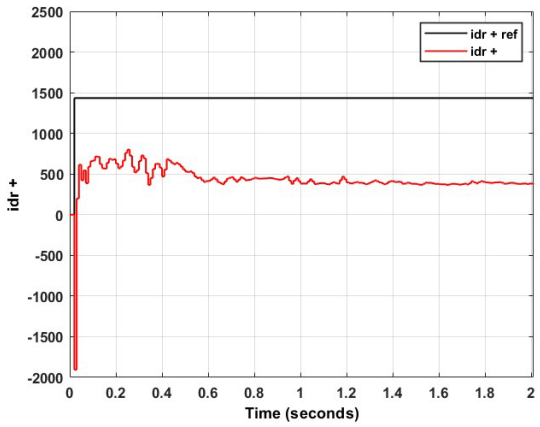


Figure 5.27: Overshoot and ripple of $i_{dr}+$ using GTO at 5% dip

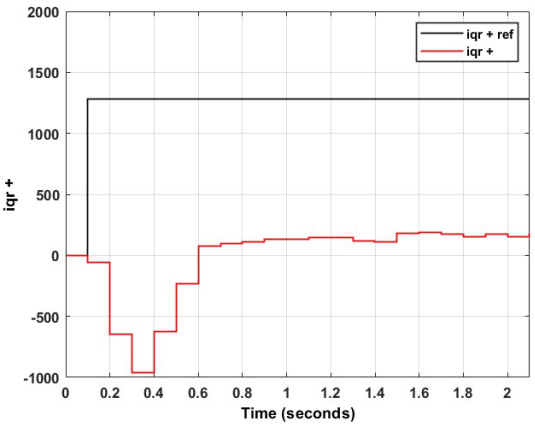


Figure 5.28: Steady-state error of $i_{qr}+$ using GTO at 5% dip

5.3.2. Case b: Voltage Unbalance of 10%

Upon application of the three swarm intelligence algorithms to the dual vector controller for a voltage unbalance of 10%, the resultant PI controller gains for the four controllers can be observed in Table 5.3. Figures 5.30–5.39 depict the results of applying PSO to the controller design. Figures 5.30 and 5.31 depict the three-phase stator voltage and current, respectively, with figures 5.32 and 5.33 displaying the three-phase rotor voltage and current, respectively. Figures 5.34 and 5.35 show the rotor direct and quadrature axis negative sequence currents. Figures 5.36 and 5.38 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas

figures 5.37 and 5.39 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

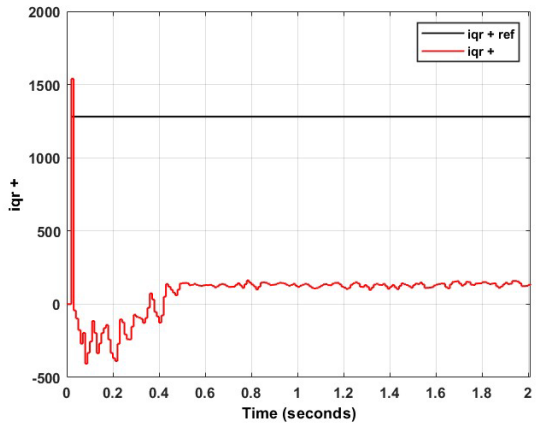


Figure 5.29: Overshoot and ripple of $i_{qr}+$ using GTO at 5% dip

Table 5.2: Summary of results obtained for 5% dip using dual vector control for each swarm technique

	PSO	**BA**	**GTO**
Steady-state error i_{dr+} (%)	4.99	**4.42**	65.87
Overshoot i_{dr+} (%)	**0**	6.72	63.6
Ripple i_{dr+} (A)	315	333	**39**
Steady-state error i_{qr+} (%)	**70.73**	76.17	86.9
Overshoot i_{qr+} (%)	41.49	43.01	**20.22**
Ripple i_{qr+} (A)	401	539	**58**

Table 5.3: Controller gains for 10% dip using dual vector control for each swarm technique

	PSO	**BA**	**GTO**
$K_{p1(idr+)}$	56.68538	1.12463	0.41106
$K_{i1(idr+)}$	70.44795	0	0
$K_{p2(iqr+)}$	48.4827	11.9731	1.0507
$K_{i2(iqr+)}$	62.29809	14.4136	0.066635
$K_{p3(idr-)}$	42.99382	12.5631	0.54513
$K_{i3(idr-)}$	30.89764	26.9153	0
$K_{p4(iqr-)}$	59.44645	31.3774	0.58278
$K_{i4(iqr-)}$	100	14.8617	0.78885

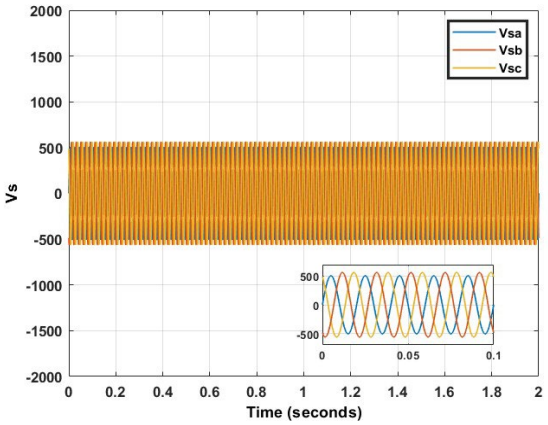


Figure 5.30: Three-phase stator voltage at 10% dip

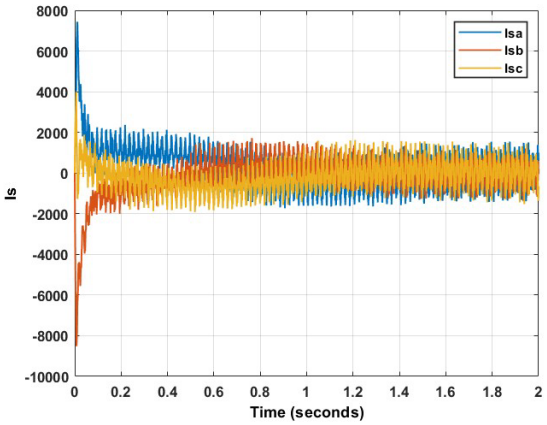


Figure 5.31: Three-phase stator current using PSO at 10% dip

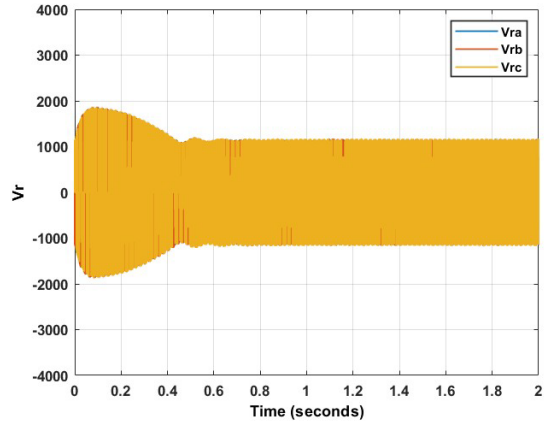


Figure 5.32: Three-phase rotor voltage using PSO at 10% dip

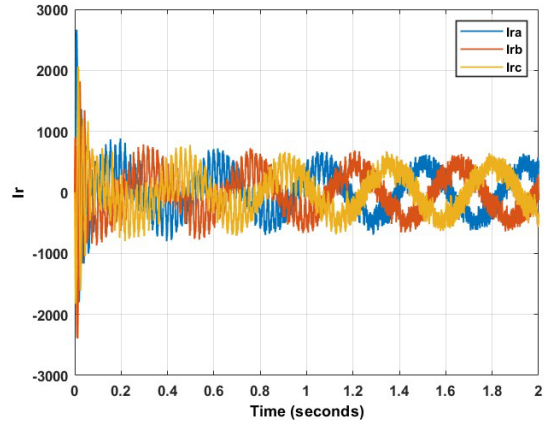


Figure 5.33: Three-phase rotor current using PSO at 10% dip

From figure 5.36, the average rotor direct steady-state current error percentage is 3.03%, which is excellent. The current ripple, as shown in figure 5.37, is large at 375 A. This is somewhat mitigated by the fact that the system is underdamped, thereby providing a 0% overshoot. The rotor quadrature current exhibits a steady-state error of 81%.96, as evident by figure 5.38. Furthermore, the steady-state ripple, as depicted by figure 5.39is once again extremely large at 1652 A. This is somewhat mitigated by the overshoot, which is 0%.

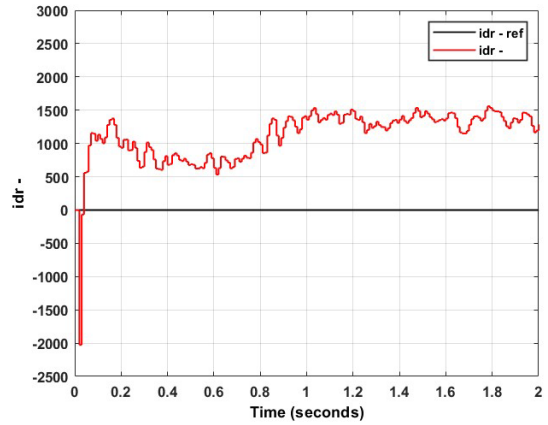


Figure 5.34: Response of i_{dr} − using PSO at 10% dip

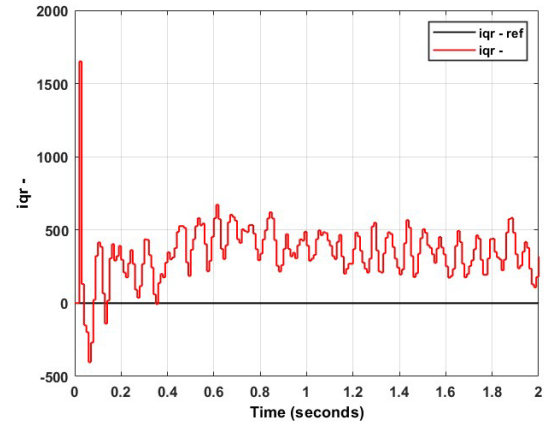


Figure 5.35: Response of i_{qr} − using PSO at 10% dip

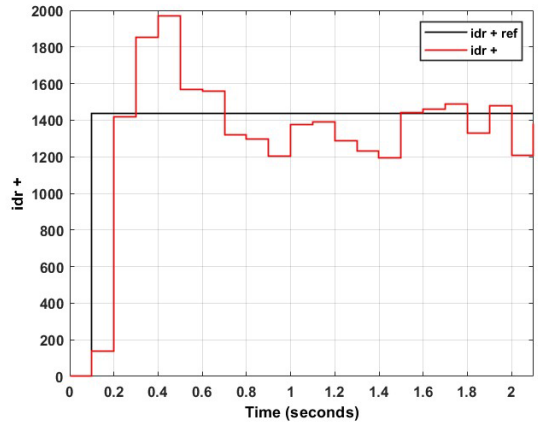


Figure 5.36: Steady-state error of $i_{dr}+$ using PSO at 10% dip

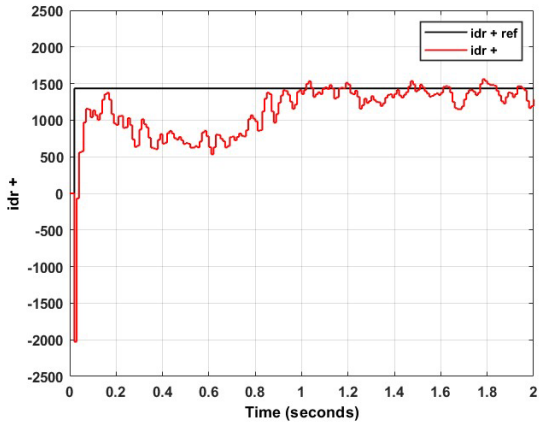


Figure 5.37: Overshoot and ripple of $i_{dr}+$ using PSO at 10% dip

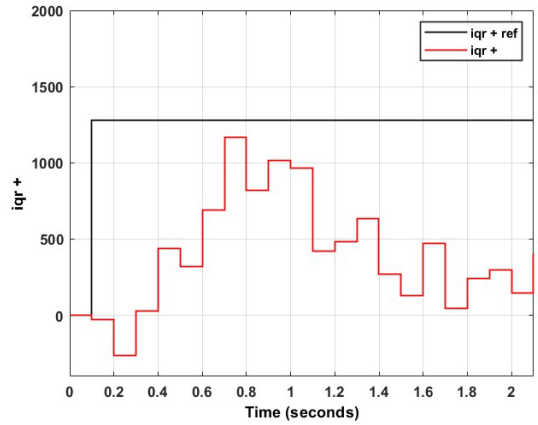


Figure 5.38: Steady-state error of $i_{qr}+$ using PSO at 10% dip

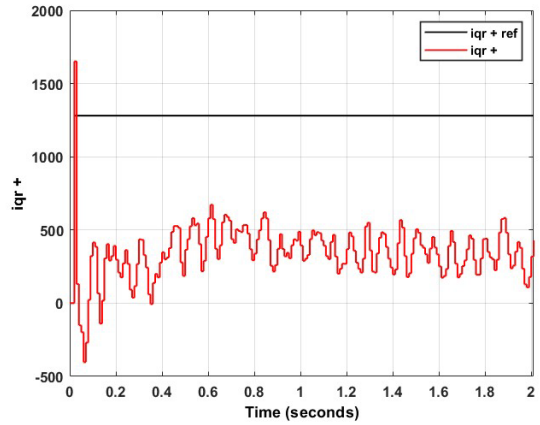


Figure 5.39: Overshoot and ripple of $i_{qr}+$ using PSO at 10% dip

Figures 5.40–5.48 depict the results of applying BA to the controller design. Figures 5.30 and 5.40 depict the three-phase stator voltage and current, respectively, with figures 5.41 and 5.42 displaying the three-phase rotor voltage and current, respectively. Figures 5.43 and 5.44 show the rotor direct and quadrature axis negative sequence currents. Figures 5.45 and 5.47 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.46 and 5.48 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

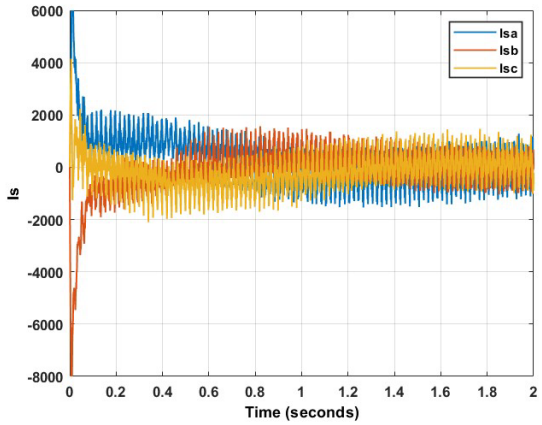


Figure 5.40: Three-phase stator current using BA at 10% dip

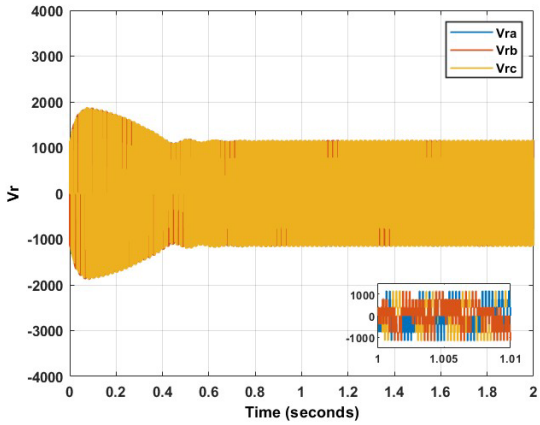


Figure 5.41: Three-phase rotor voltage using BA at 10% dip

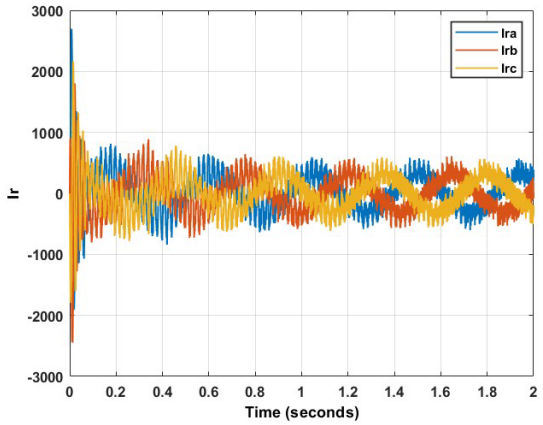


Figure 5.42: Three-phase rotor current using BA at 10% dip

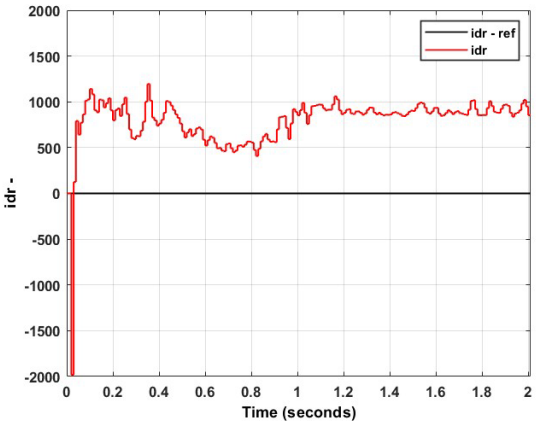


Figure 5.43: Response of $i_{dr}-$ using BA at 10% dip

From figure 5.45, the average rotor direct steady-state current error percentage is 42.6%, which is unacceptable. The current ripple, as shown in figure 5.46, is also large at 253 A. This is mitigated by the 0% overshoot, owing to the overdamped nature of the response. The rotor quadrature current exhibits a steady-state error of 81.62%, as evident by figure 5.47. This is once again unacceptable. The steady-state ripple, as depicted by figure 5.48 is also large, at 296 A. The overshoot is respectable, at 35.68%.

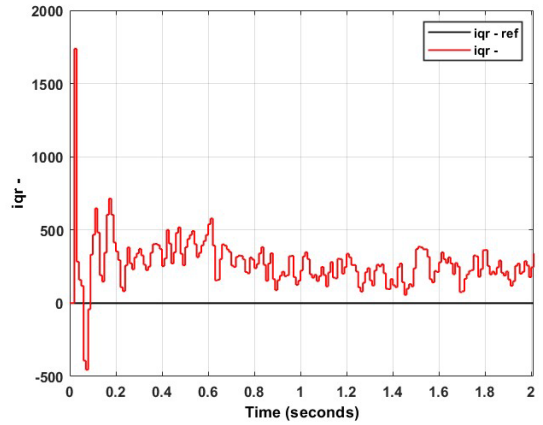


Figure 5.44: Response of $i_{qr}-$ using BA at 10% dip

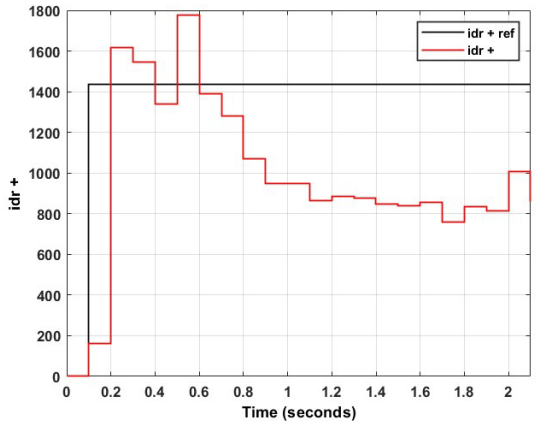


Figure 5.45: Steady-state error of $i_{dr}+$ using BA at 10% dip

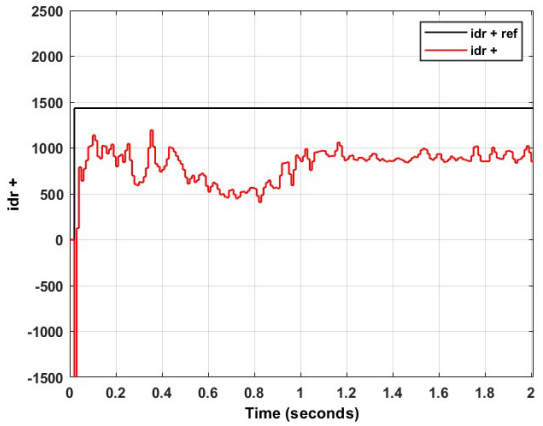


Figure 5.46: Overshoot and ripple of $i_{dr}+$ using BA at 10% dip

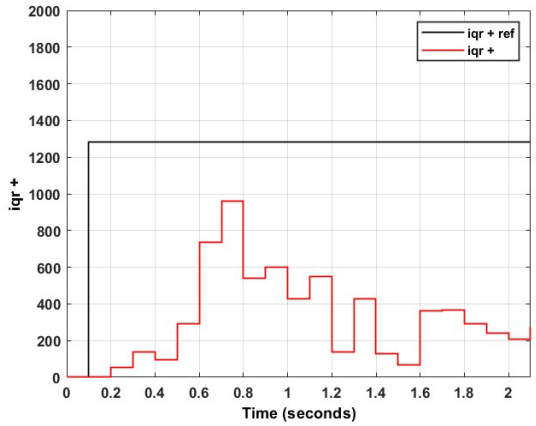


Figure 5.47: Steady-state error of $i_{qr}+$ using BA at 10% dip

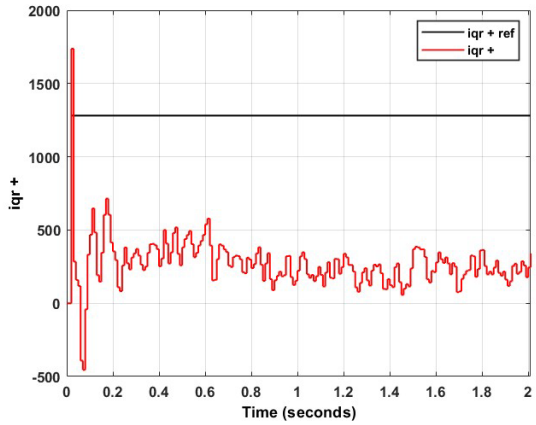


Figure 5.48: Overshoot and ripple of $i_{qr}+$ using BA at 10% dip

Figures 5.49–5.57 depict the results of applying GTO to the controller design. Figures 5.30 and 5.49 depict the three-phase stator voltage and current, respectively, with figures 5.50 and 5.51 displaying the three-phase rotor voltage and current, respectively. Figures 5.52 and 5.53 show the rotor direct and quadrature axis negative sequence currents. Figures 5.54 and 5.56 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.55 and 5.57 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

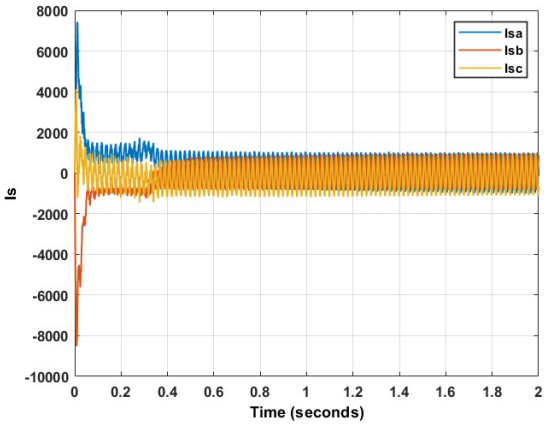


Figure 5.49: Three-phase stator current using GTO at 10% dip

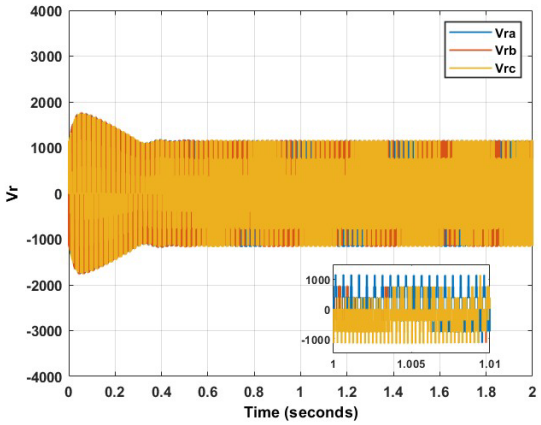


Figure 5.50: Three-phase rotor voltage using GTO at 10% dip

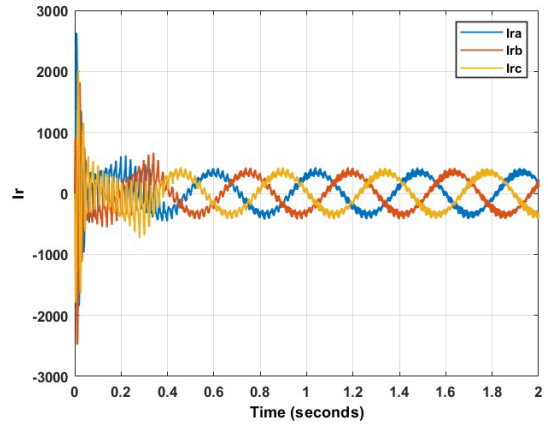


Figure 5.51: Three-phase rotor current using GTO at 10% dip

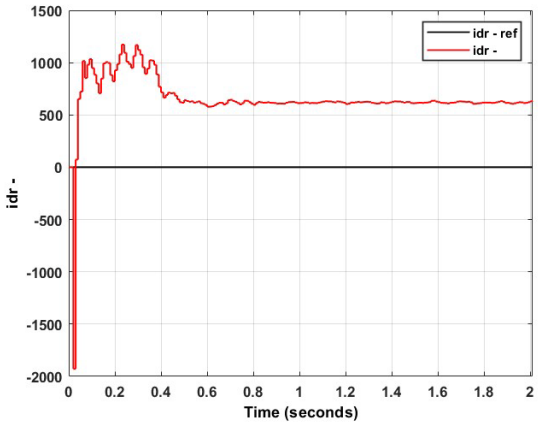


Figure 5.52: Response of $i_{dr}-$ using GTO at 10% dip

From figure 5.54, the average rotor direct steady-state current error percentage is 53.5%, which is unacceptable. The current ripple, as shown in figure 5.55, is exceptional at 29 A. Further, owing to the overdamped nature of the response, the overshoot is excellent at 0%. The rotor quadrature current exhibits a steady-state error of 46.71%, as evident by figure 5.56. This is once again unacceptable. The steady-state ripple, as depicted by figure 5.57 is once again minute at 92 A. This is complemented by the overshoot, which is strong at 27%. The results of the experiment are summarized in table 5.4. As evident, based on the various performance parameters, the best results are split among the various techniques. However, according to rankings, GTO produced the overall best performance, with the BA exhibiting the worst results.

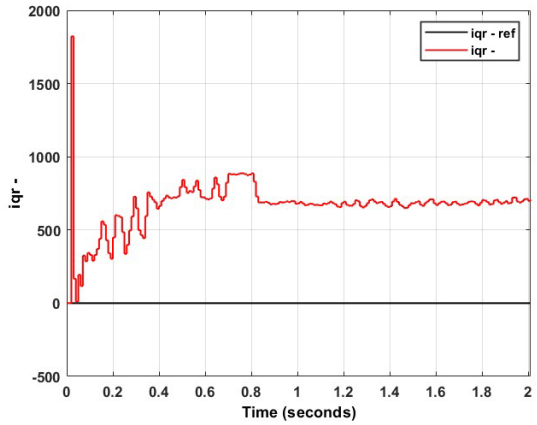


Figure 5.53: Response of $i_{qr}-$ using GTO at 10% dip

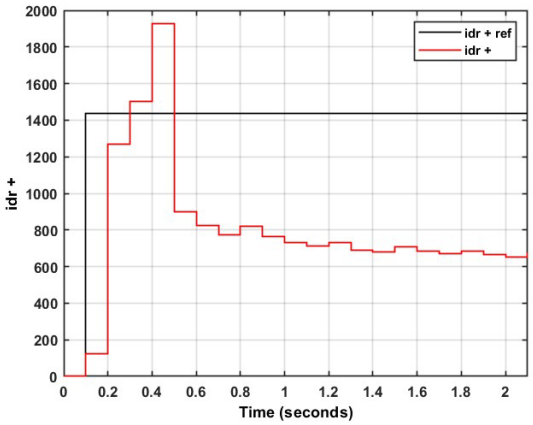


Figure 5.54: Steady-state error of $i_{dr}+$ using GTO at 10% dip

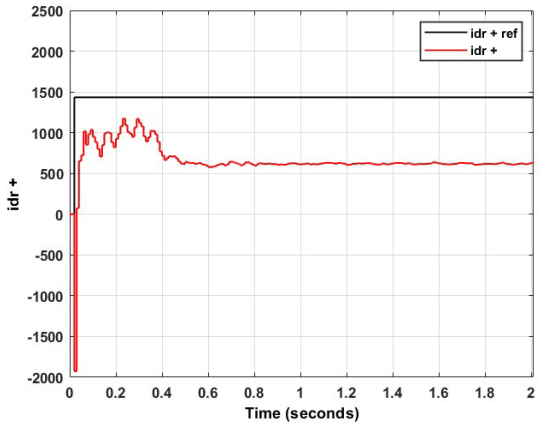


Figure 5.55: Overshoot and ripple of $i_{dr}+$ using GTO at 10% dip

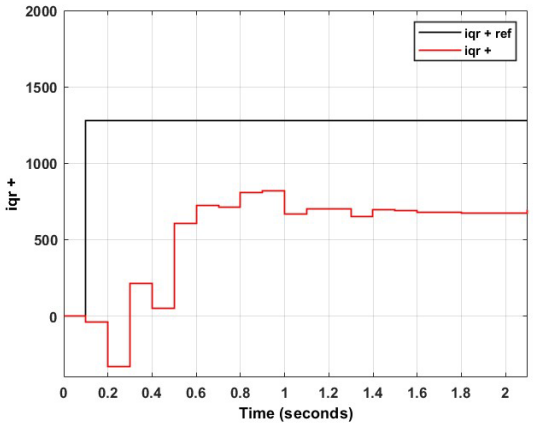


Figure 5.56: Steady-state error of $i_{qr}+$ using GTO at 10% dip

Table 5.4: Summary of results obtained for 10% dip using dual vector control for each swarm technique

	PSO	**BA**	**GTO**
Steady-state error i_{dr+} (%)	**3.03**	42.6	53.5
Overshoot i_{dr+} (%)	**0**	**0**	**0**
Ripple i_{dr+} (A)	375	253	**29**
Steady-state error i_{qr+} (%)	81.96	81.62	**46.71**
Overshoot i_{qr+} (%)	**0**	35.68	27
Ripple i_{qr+} (A)	1652	296	**92**

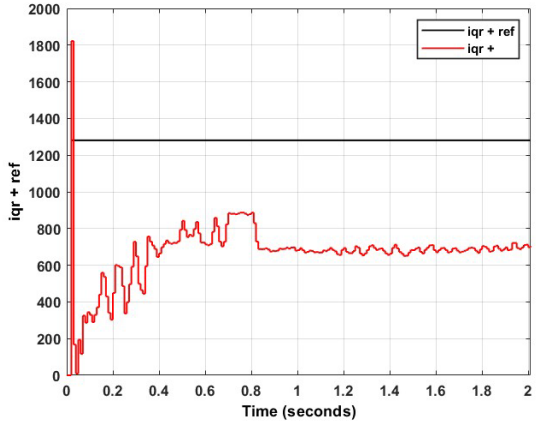


Figure 5.57: Overshoot and ripple of $i_{qr}+$ using GTO at 10% dip

5.3.3. Case c: Voltage Unbalance of 7.5%

Upon application of the three swarm intelligence algorithms to the dual vector controller for a dual-phase voltage unbalance of 7.5%, the resultant PI controller gains for the four controllers can be observed in Table 5.5. Figures 5.58–5.67 depict the results of applying PSO to the controller design. Figures 5.58 and 5.59 depict the three-phase stator voltage and current, respectively, with figures 5.60 and 5.61 displaying the three-phase rotor voltage and current, respectively. Figures 5.62 and 5.63 show the rotor direct and quadrature axis negative sequence currents. Figures 5.64 and 5.66 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.65 and 5.67 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

Table 5.5: Controller gains for 7.5% dip using dual vector control for each swarm technique

	PSO	BA	GTO
$\boldsymbol{K_{p1(idr+)}}$	25.48194	47.0731	100
$\boldsymbol{K_{i1(idr+)}}$	27.21281	0	1.985359
$\boldsymbol{K_{p2(iqr+)}}$	8.821569	73.6876	72.92349
$\boldsymbol{K_{i2(iqr+)}}$	100	96.1084	1.223389
$\boldsymbol{K_{p3(idr-)}}$	3.545885	48.3316	100
$\boldsymbol{K_{i3(idr-)}}$	100	3.81148	1.950241
$\boldsymbol{K_{p4(iqr-)}}$	68.30388	3.65399	1.626019
$\boldsymbol{K_{i4(iqr-)}}$	18.20631	81.3853	0

From Figure 5.64, the average rotor direct steady-state current error percentage is 10.71%, which is not the best, but acceptable. The current ripple, as shown in figure 5.65, is large at 305 A. This is somewhat mitigated by the fact that the system is underdamped, thereby providing a 0% overshoot. The rotor quadrature current exhibits a steady-state error of 69.54%, as evident by figure 5.66. Furthermore, the steady-state ripple, as depicted by figure 5.67 is once again large at 494 A. The overshoot, which is 34.58%, is acceptable.

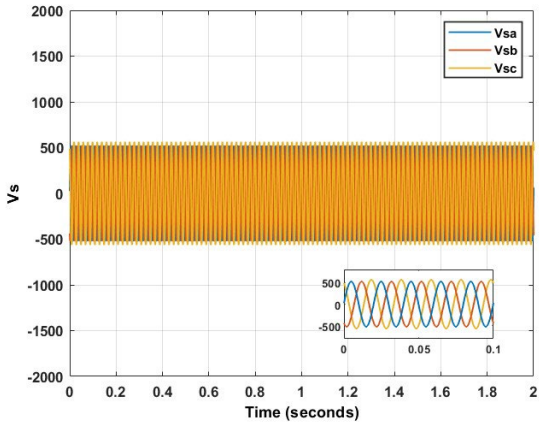


Figure 5.58: Three-phase stator voltage at 7.5% dip

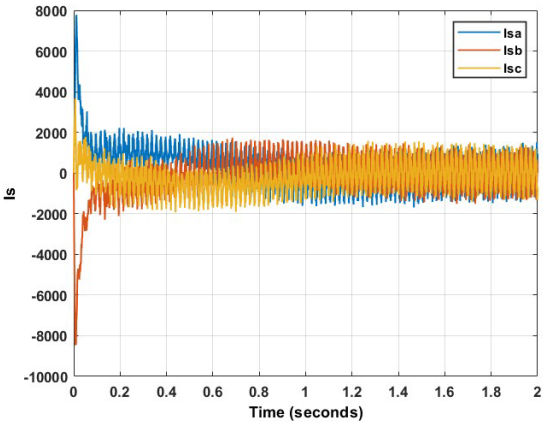


Figure 5.59: Three-phase stator current using PSO at 7.5% dip

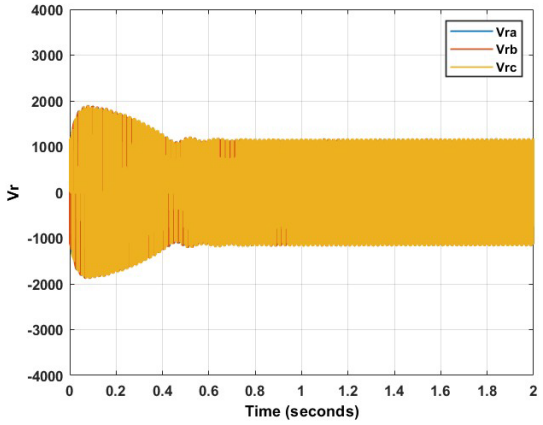


Figure 5.60: Three-phase rotor voltage using PSO at 7.5% dip

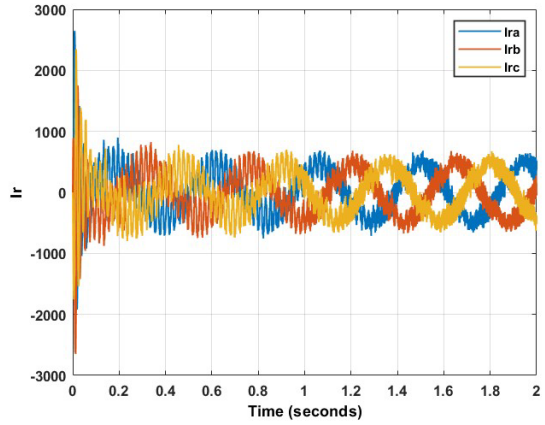


Figure 5.61: Three-phase rotor current using PSO at 7.5% dip

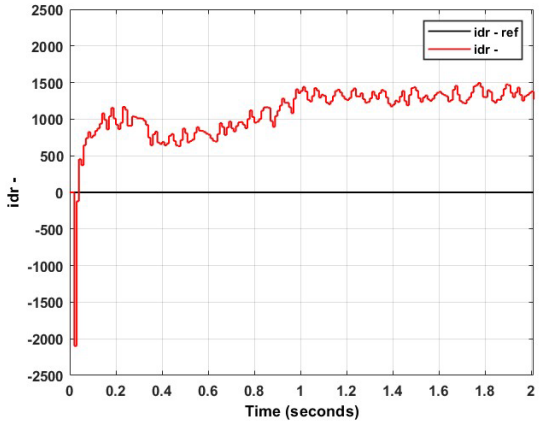


Figure 5.62: Response of i_{dr} − using PSO at 7.5% dip

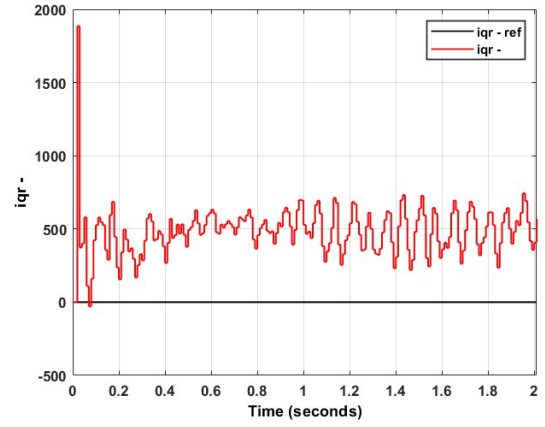


Figure 5.63: Response of i_{qr} − using PSO at 7.5% dip

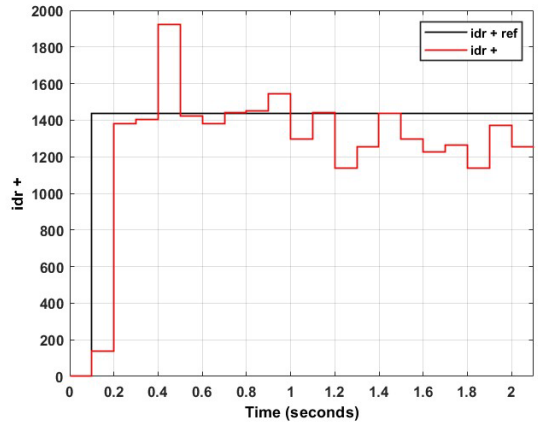


Figure 5.64: Steady-state error of $i_{dr}+$ using PSO at 7.5% dip

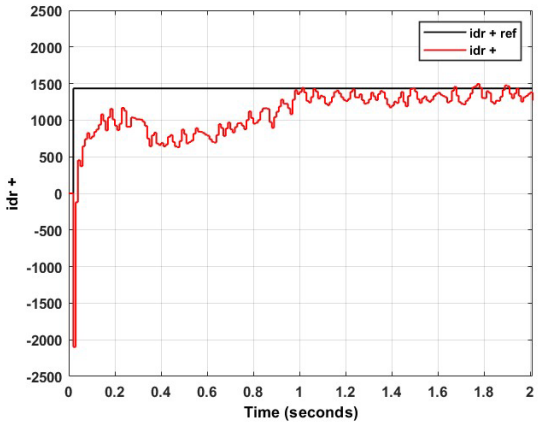


Figure 5.65: Overshoot and ripple of $i_{dr}+$ using PSO at 7.5% dip

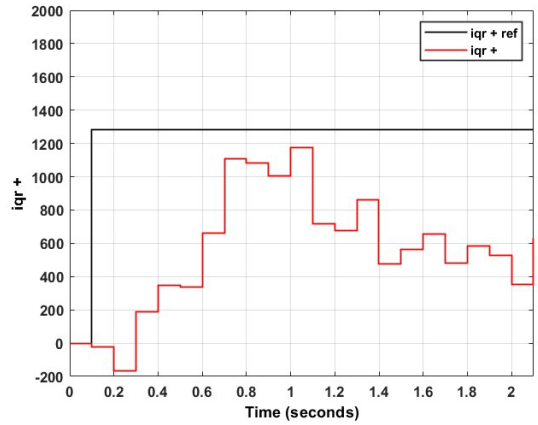


Figure 5.66: Steady-state error of $i_{qr}+$ using PSO at 7.5% dip

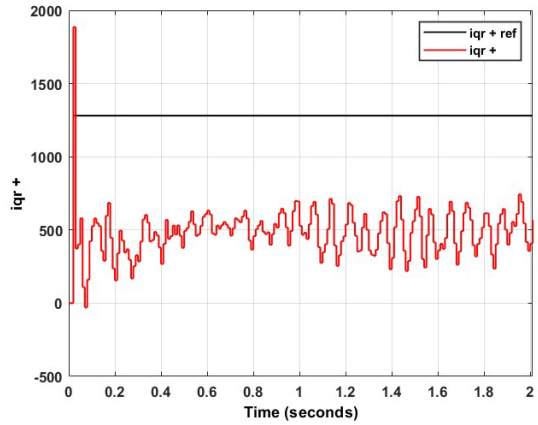


Figure 5.67: Overshoot and ripple of $i_{qr}+$ using PSO at 7.5% dip

Figures 5.68–5.76 depict the results of applying BA to the controller design. Figures 5.58 and 5.68 depict the three-phase stator voltage and current, respectively, with figures 5.69 and 5.70 displaying the three-phase rotor voltage and current, respectively. Figures 5.71 and 5.72 show the rotor direct and quadrature axis negative sequence currents. Figures 5.73 and 5.75 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.74 and 5.76 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

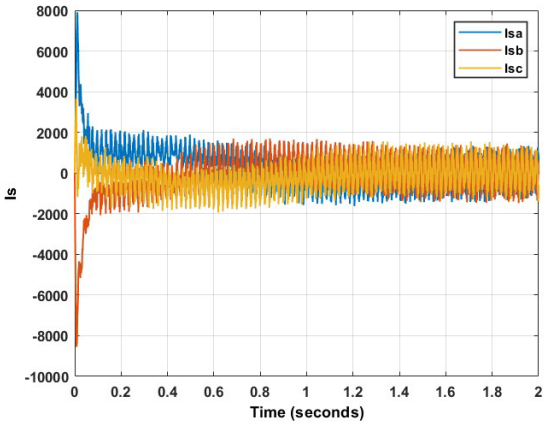


Figure 5.68: Three-phase stator current using BA at 7.5% dip

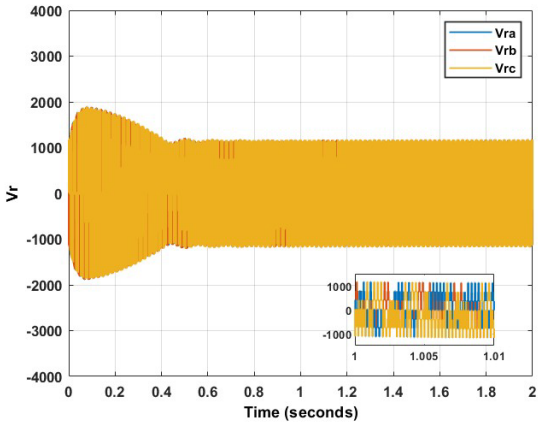


Figure 5.69: Three-phase rotor voltage using BA at 7.5% dip

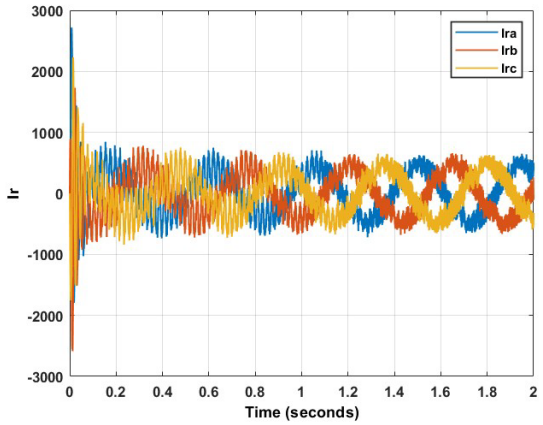


Figure 5.70: Three-phase rotor current using BA at 7.5% dip

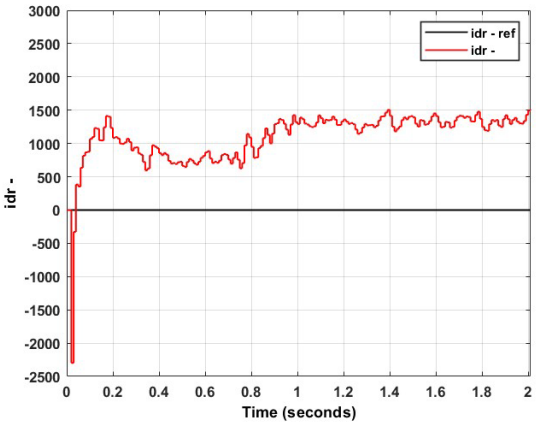


Figure 5.71: Response of $i_{dr}-$ using BA at 7.5% dip

From Figure 5.73, the average rotor direct steady-state current error percentage is 2.19%, which is excellent. The current ripple, however, as shown in figure 5.74, is large at 384 A. This is mitigated by the minute 2.19% overshoot. The rotor quadrature current exhibits a steady-state error of 52.31%, as evident by figure 5.75. This is once again unacceptable. The steady-state ripple, as depicted by figure 5.76 is also large, at 356 A. The overshoot is respectable, at 42.7%.

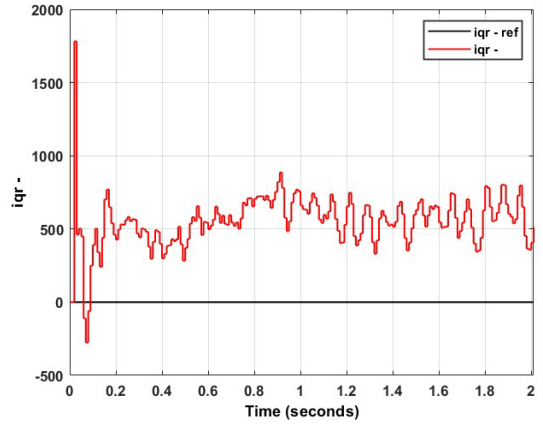


Figure 5.72: Response of $i_{qr}-$ using BA at 7.5% dip

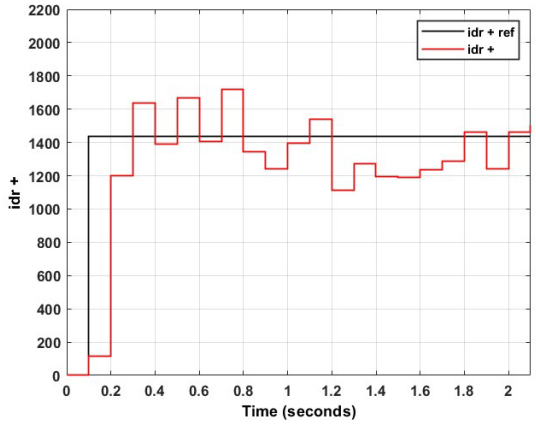


Figure 5.73: Steady-state error of $i_{dr}+$ using BA at 7.5% dip

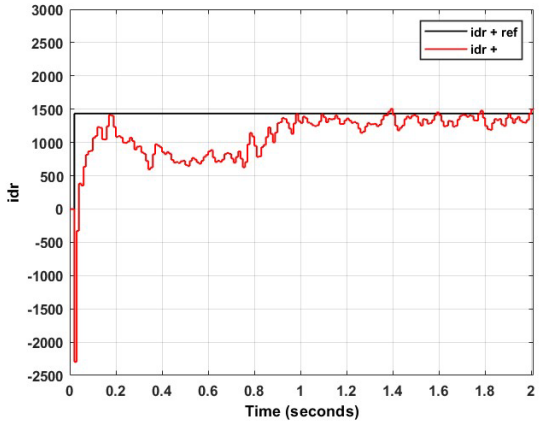


Figure 5.74: Overshoot and ripple of $i_{dr}+$ using BA at 7.5% dip

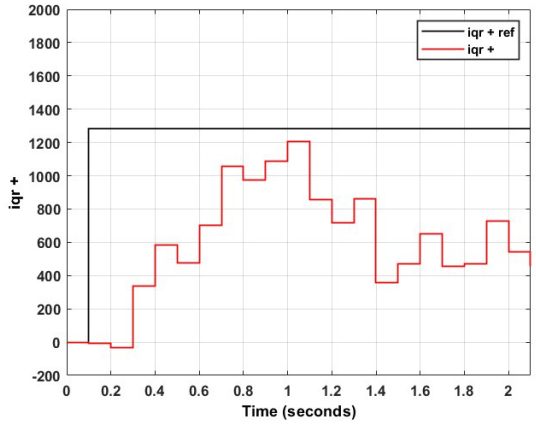


Figure 5.75: Steady-state error of $i_{qr}+$ using BA at 7.5% dip

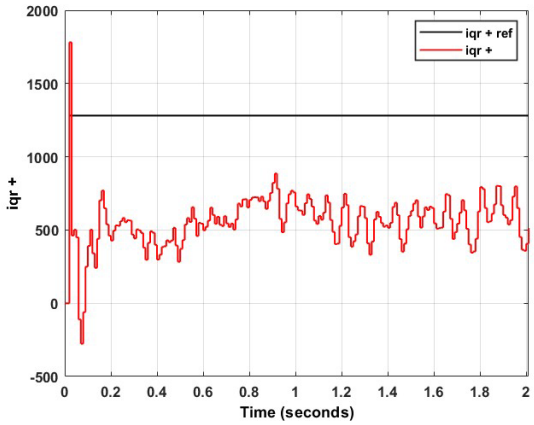


Figure 5.76: Overshoot and ripple of $i_{qr}+$ using BA at 7.5% dip

Figures 5.77–5.85 depict the results of applying GTO to the controller design. Figures 5.58 and 5.77 depict the three-phase stator voltage and current, respectively, with Figures 5.78 and 5.79 displaying the three-phase rotor voltage and current, respectively. Figures 5.80 and 5.81 show the rotor direct and quadrature axis negative sequence currents. Figures 5.82 and 5.84 are utilized for analyzing the steady-state error of the positive sequence rotor currents, whereas figures 5.83 and 5.85 are used to observe the overshoot and steady-state ripple, once again for the positive sequence rotor current.

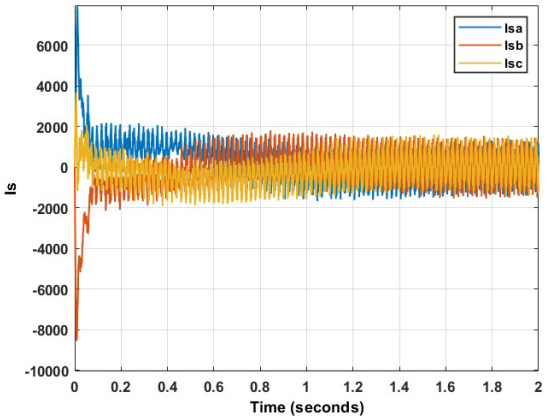


Figure 5.77: Three-phase stator current using GTO at 7.5% dip

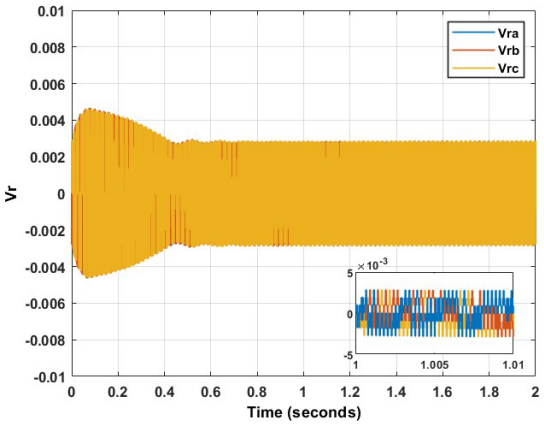


Figure 5.78: Three-phase rotor current using GTO at 7.5% dip

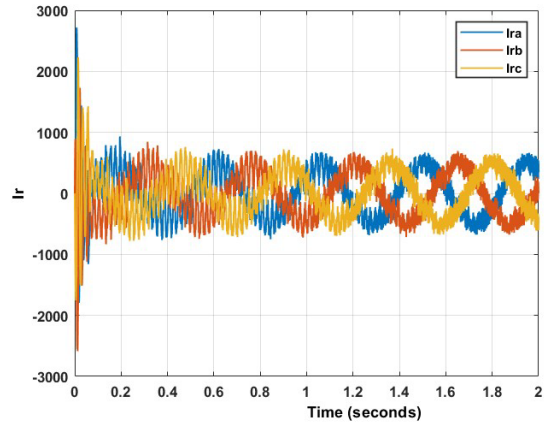


Figure 5.79: Three-phase rotor current using GTO at 7.5% dip

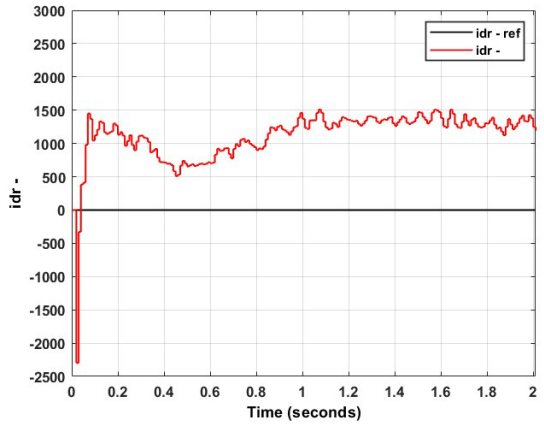


Figure 5.80: Response of $i_{dr}-$ using GTO at 7.5% dip

From figure 5.82, the average rotor direct steady-state current error percentage is 11.8%, which is somewhat acceptable. The current ripple, as shown in figure 5.83, is acceptable at 327 A. Further, owing to the overdamped nature of the response, the overshoot is excellent at 0.6%. The rotor quadrature current exhibits a steady-state error of 49.4%, as evident by figure 5.84. This is once again unacceptable. The steady-state ripple, as depicted by figure 5.85 is once again acceptable at 407 A. This is complemented by the overshoot, which is respectable at 44.4%. The results of the experiment are summarized in table 5.6. As evident, based on the various performance parameters, the best results are split among the various techniques. However, according to rankings, PSO produced the overall best performance, with GTO exhibiting the worst results.

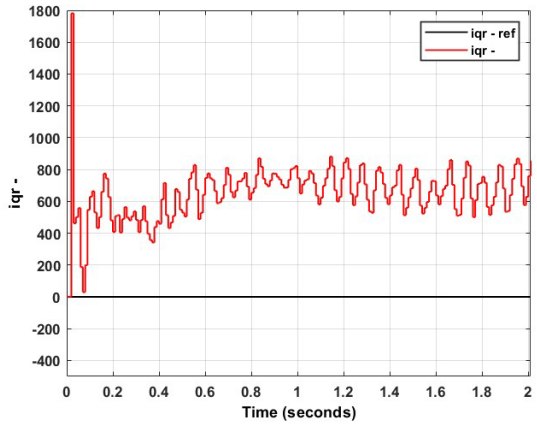


Figure 5.81: Response of $i_{qr}-$ using GTO at 7.5% dip

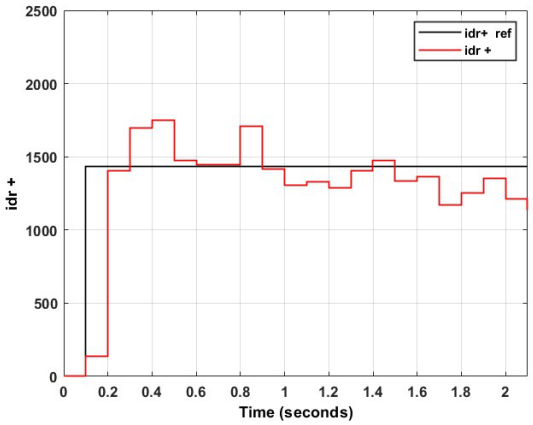


Figure 5.82: Steady-state error of $i_{dr}+$ using GTO at 7.5% dip

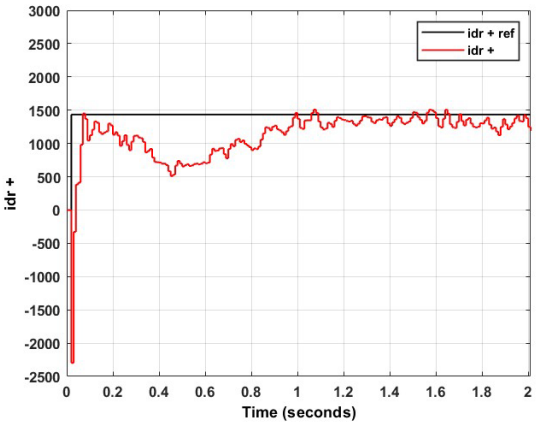


Figure 5.83: Overshoot and ripple of $i_{dr}+$ using GTO at 7.5% dip

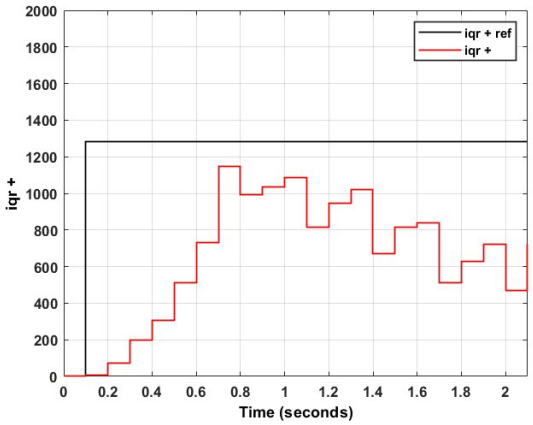


Figure 5.84: Steady-state error of $i_{qr}+$ using GTO at 7.5% dip

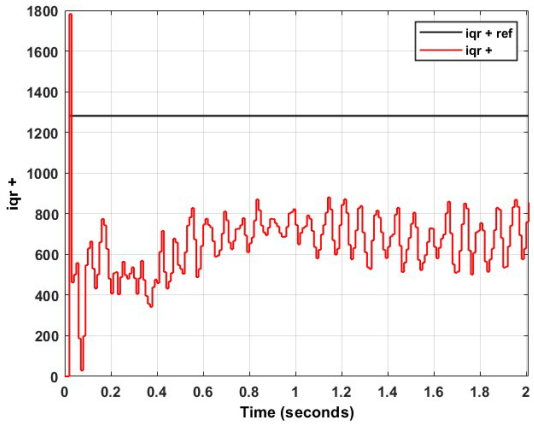


Figure 5.85: Overshoot and ripple of $i_{qr}+$ using GTO at 7.5% dip

Table 5.6: Summary of results obtained for 7.5% dip using dual vector control for each swarm technique

	PSO	**BA**	**GTO**
Steady-state error i_{dr+} (%)	10.71	**2.94**	11.8
Overshoot i_{dr+} (%)	**0**	2.19	0.6
Ripple i_{dr+} (A)	**305**	384	327
Steady-state error i_{qr+} (%)	69.54	52.31	**49.4**
Overshoot i_{qr+} (%)	**34.58**	42.7	44.4
Ripple i_{qr+} (A)	494	**356**	407

5.4. Conclusion

This chapter provided insight into the application of swarm intelligence to the control of the DFIG under the influence of asymmetrical voltage dips. The control technique applied was the dual vector control method, with the control target being balanced rotor currents. Both single and dual-phase voltage dips were applied to the system. The PI controllers responsible for regulating both the positive and negative sequence rotor direct and quadrature axis currents, were optimized via three swarm intelligence techniques. There were chosen across the timeline spectrum, and were Particle Swarm Optimization, Bat Algorithm, and Gorilla Troops Optimization. For the single-phase voltage dips, it was observed that PSO produced the overall best steady-state error and overshoot of the rotor direct axis current. For the same experiments, the GTO exhibited exceptional rotor direct and quadrature axis current ripples. However, despite these adequate results, neither of the three algorithms failed to achieve an acceptable rotor quadrature axis current steady-state error. This also holds true for the dual-phase voltage dip and can be attributed to the algorithms inability to simultaneously optimize eight controller parameters. The computational complexity required in the optimization of eight controller gains significantly decreases the efficacy of the algorithm.

Chapter 6 : An investigation into the utilization of swarm intelligence in the design of Proportional Resonant controller

As evident in chapter 5, it is observed that the use of metaheuristics in the optimization of the dual vector control produced a strong steady-state ripple and decent overshoot. Both parameters are important aspects to consider. However, it is seen that the steady-state error of the rotor quadrature axis currents proved to be extremely poor. This is undesirable, as it would lead to an unstable regulation of the stator active power. This section proposes the use of the Proportional Resonant controller for control of the DFIG subject to asymmetrical voltage dips. This controller possesses various favourable characteristics, such as the lack of sequence decomposition requirement, and infinite gain at the resonant frequency, leading to a zero steady-state error. The proposed controller will be optimized using the same three swarm intelligence techniques which were utilized in chapter 5. Further, a stability analysis of the designed controller is carried out, subject to a 3% perturbation in grid frequency.

6.1. Background

The use of a proportional–resonant controller was introduced in [215]. The controller is used to replace the conventional PI controller. The merits of the controller include the fact that there is no need for the individual control of positive and negative sequence decomposition, thereby eliminating the time delay and instability as seen in the conventional PI controller [216]. The PR controller can obtain an infinite gain at the fundamental frequency, and therefore can achieve a negligible steady-state error [217]. The general structure of DFIG control using PR controller can be observed in Figure 6.1 [218]. The controller outlined in [215] consisted of a proportional gain, a resonant gain, and a bandwidth gain. The bandwidth gain was utilized to negate the effects of grid frequency fluctuations on the operation of the controller. The proposed controller was compared to that of the conventional PI controller, as well as the dual vector control method, and exhibited superior performance. However, optimization of the controller gains was achieved via the tedious method of trial and error. Furthermore, the controller was only tested on a 10% voltage unbalance. The authors in [216] proposed the use of a main proportional–resonant controller, and two auxiliary PR controllers. The resonant frequency of each of the three controllers was different, each making use of the synchronous and/or rotor angular velocities. Each controller consisted of an independent proportional and resonant gain, and the output of each controller was summed. The proposed controller yielded superior results to the conventional control method; however, no mention of the controller gain optimization was made. This would have been a difficult task as there existed six gains to be optimized. The method introduced in [219] proposed a novel method to obtain the rotor reference currents in the case of an unbalanced grid. However, this required sequence decomposition and the use of a notch filter, which contributes to time delay and system instability. An enhancement of [219] is proposed in [220], with the control method this time not requiring sequence decomposition. However, the drawback is the tuning of two additional PI controllers, which significantly adds to the complexity of the control strategy.

In [221], the transfer function of the PR controller was modified to eliminate fifth and seventh order harmonics. The result showed a significant reduction in stator current harmonic distortion. A comparative study between the PI, PID, and PR controllers for fault ride-through enhancement of the DFIG was carried out in [222] and [223]. Once again, in both studies, the PR controller outperformed the PI and PID controllers. A fuzzy-based PR controller was proposed in [224]. The proportional and resonant gains were dynamic and changed according to

the situation. While they exhibit a superior response to the conventional PI control system, fuzzy controllers are known to be unreliable. The PR controller was coupled with a two-stage matrix converter in [225]. However, this method made use of a notch filter and sequence decomposition, thereby reducing the reliability of the control system. The application

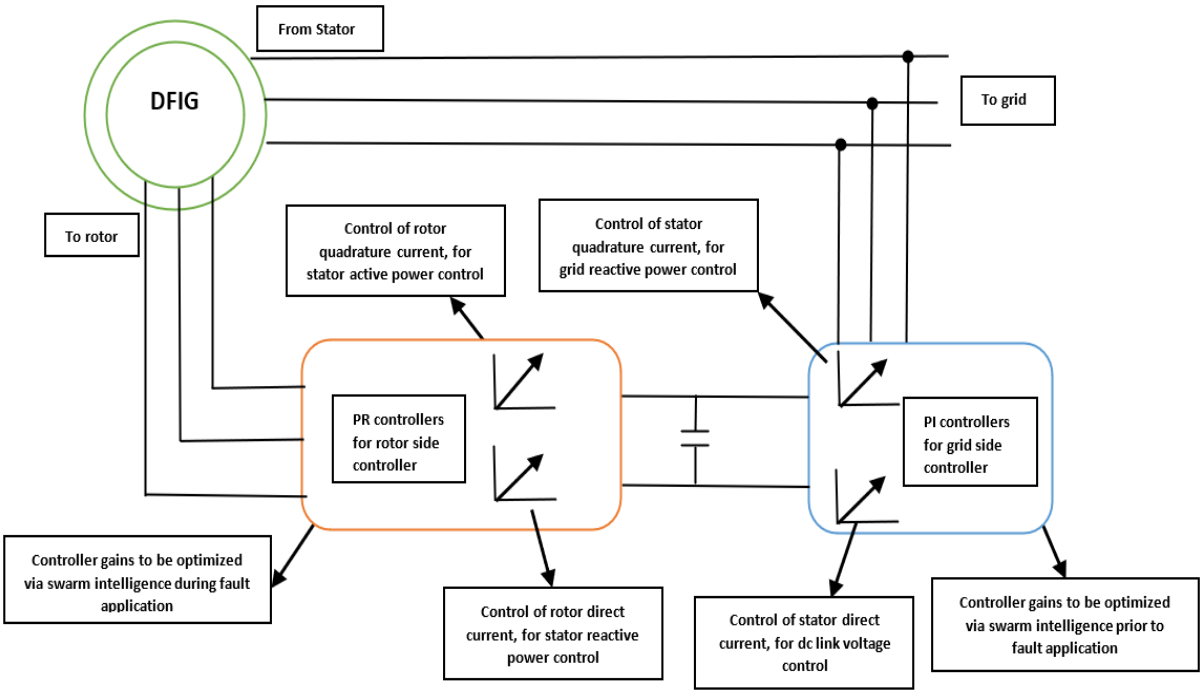


Figure 6.1: General structure of DFIG based PR control [218]

of the PR controller to system stability was investigated in [226]. The control aimed at mitigating sub synchronous stator currents. In this case, the PR controller was applied to both the RSC and GSC and exhibited a strong dynamic response.

The PR controller has been successfully applied to the control of the DFIG. This is not just for the mitigation of the effects of unbalanced grids, but also for aspects such as the reduction in harmonic distortion, low voltage ride through, and stability enhancement. The controller exhibits a superior response to that of the PI controller, particularly during the case of non-ideal occurrences. However, despite the application of the controller being mentioned in the literature, no mention is made regarding the optimization of gain values. Most instances utilize the concept of trial and error, which may lead to suboptimal performance of the controller. Swarm intelligence has recently been applied to the optimization of such controllers. In [227], the well-known PSO was utilized to obtain the gains of a PR controller, for application to a grid-tied PV inverter. Similar applications of PSO are observed in [228] and [229]. A PR-based PWM was applied to a three-phase, three-leg shunt active power filter in [230]. The controllers were optimized using the bat algorithm (BA). Compared to the PI controller, the BA-based PR controller produced a lower harmonic regulation. However, the effectiveness of the BA as a tuning method was not investigated. In [231], the grey wolf optimizer was used to simultaneously optimize the parameters of PR controllers as well as an output filter for a three-phase grid connected inverter. It is evident that there have been instances whereby swarm intelligence was utilized for PR controller gain optimization.

Considering the application of swarm intelligence to the DFIG, numerous scenarios exist. From the literature it is evident that PSO is the technique most widely applied to the DFIG. The authors in [60] utilized PSO optimization of sliding mode controller parameters. A small signal stability analysis was carried out in [61]. Here, PSO was responsible for rotor side controller, grid side controller, and pitch controller gain optimization. Considering the bat algorithm, this technique was also utilized to optimize the parameters of a sliding mode controller in [103]. In [232], the BA was used to optimize the parameters of a proportional–integral–resonant controller. However, this was only analysed in terms of harmonic distortion, and not for the effect of unbalanced grid voltages. There also exists other swarm intelligence techniques, such as artificial bee colony and grey wolf optimization, which have thus far displayed promising results when applied to the control of the DFIG [85], [75]. The gorilla troops optimizer is a recently developed swarm-based metaheuristic optimization technique [149]. This technique has thus far been applied to numerous optimization problems. In [150], the authors made use of GTO to optimize the frequency response of a microgrid. The algorithm was also utilized for the problem of optimal power flow in a power system [151], and for feature selection in cyber security [152]. However, no application of gorilla troops optimization to the control of the DFIG has yet been proposed in the literature.

From the past application of swarm intelligence, it is evident that these algorithms produce excellent results. Therefore, there exists a strong possibility that the utilization of a swarm intelligence technique for the optimization of PR controller gains for DFIG control will exhibit exceptional results. Swarm intelligence has strong application in the field of data science [233]. Furthermore, large power utilities are moving towards the utilization of swarm intelligence for control and process optimization. For example, the concept of high impedance faults is a big concern. The use of swarm intelligence for high impedance fault detection has proved to be valuable [234]. The aim of this chapter is to investigate the effects of utilizing swarm intelligence techniques for the optimization of PI and PR controller gains, for application to the DFIG under the influence of asymmetrical voltage dips. Two well-known swarm techniques, particle swarm optimization and the bat algorithm, as well as the recently proposed gorilla troops optimization algorithm are applied and analysed in this investigation. The algorithms are applied to optimization problems across minor single-phase and phase-to-phase grid faults.

6.2. The PR controller

The structure of the PR controller is like that of the conventional PI controller, with the omission of the integral controller and the addition of a resonant controller. The resonant controller is commonly tuned at synchronous frequency [235], [236]. The general topology of the PR controller is observed in Figure 6.2 [60], [61]. The transfer function of the PR controller can therefore be expressed as [235], [236]:

$$y(s) = e(t)(K_p + \frac{K_r s}{s^2 + w^2}) \tag{6.1}$$

where K_p is the proportional gain, K_r is the resonant gain, w is the frequency at which the resonant controller is tuned, $e(t)$ is the error signal that is input to the PR controller, and $y(s)$ is the output of the PR controller. The proportional gain, K_p, allows for a fast response of the control system. The resonant gain, K_r allows for infinite gain at synchronous frequency

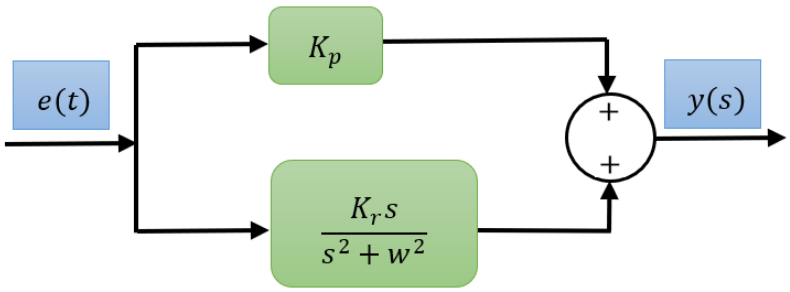


Figure .6.2: Structure of PR controller

It is well known that when connected to an unbalanced grid, the DFIG produces negative sequence rotor and stator voltage and currents. These negative sequence components give rise to double frequency oscillatory terms. Theory pertaining to the behaviour of the DFIG under the influence of an unbalanced grid is thoroughly explained in [216], [219], [237] .The advantage of the PR controller is that it is capable of successfully tracking the ac reference current. This allows the controller to eliminate steady-state errors at the chosen frequency. K_p plays the same role as that in a conventional PI controller. The resonant controller is the one that can reduce the double frequency oscillatory terms. Considering Figure 6.3, which depicts the bode diagram of the PI and PR controllers, it is evident that the PR controller produces a large gain at the resonant frequency, and thus can significantly reduce steady-state error.

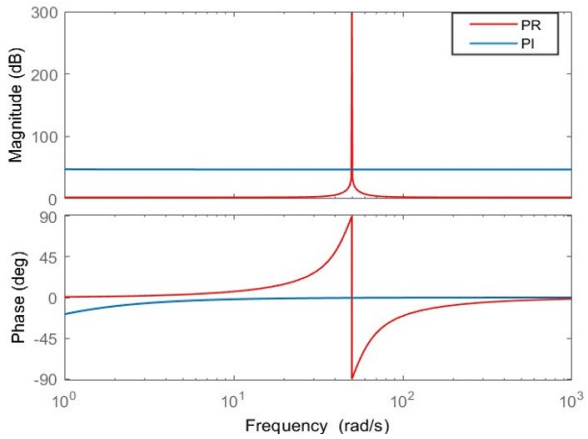


Figure .6.3: Bode plots of PI and PR controllers

The resonant gain, K_r, becomes difficult to optimize. A large gain value will increase the frequency bandwidth but will reduce the reliability of the controller [236]. Similarly, a small gain value provides a reliable response but within a minute frequency range. Considering Figure 6.4, which depicts the bode diagram of a PR controller with different resonant gains, it is evident that larger resonant gains lead to larger bandwidths, a favourable outcome. Hence, it can be concluded that the optimization of the resonant gain is essential to provide both a reliable response and an acceptable bandwidth.

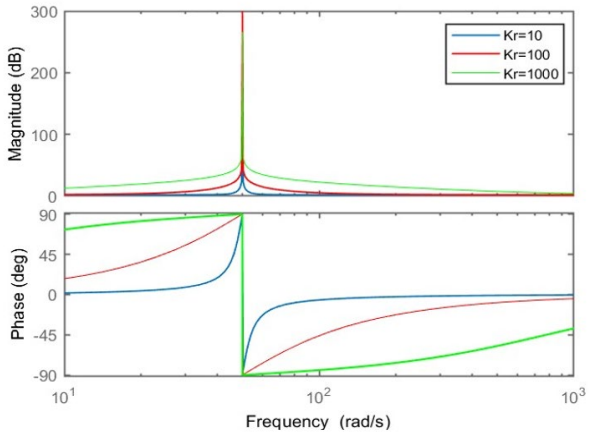


Figure .6.4: Bode plot of PR controller with differing resonant gains

6.3. Experimental results and analysis

Three swarm intelligence algorithms were applied to the design of the proportional–resonant controller. As evident from figure 6.1, the conventional PI controllers were utilized for the grid side converter. Further, these controllers were optimized prior to the fault condition being implemented. The reason for this is to clearly demonstrate the effect of applying the PR controller to the rotor side converter of the DFIG. These are across the timeline of the evolution of such algorithms. The first algorithm to be applied was the well-known particle swarm optimization, which was first described in 1995 [55]. Second, the bat algorithm, which was developed in 2012 [96], was tested. Lastly, application of the very recent gorilla troops optimizer was evaluated [149]. The control system implemented was stator flux-oriented control. This means that the stator reactive power was controlled by the rotor direct current, and the stator active power by the rotor quadrature current. The results obtained were evaluated in terms of steady-state error, steady-state ripple, and overshoot. Owing to the direct proportional relationship between the rotor currents and the stator power, the output of the DFIG can be directly analyzed from the waveforms of the rotor currents. These relationships can be observed in [214]. It should be noted that owing to the complexity of the simulation circuit, each iteration takes a while to complete. To ensure that this did not take an unreasonable amount of time to execute, each of the three swarm intelligent algorithms were subject to 20 search agents and 20 iterations. This should suffice, considering that particle swarm optimization, which is the most established and widely used technique, is known to exhibit a fast convergence rate [55]. Rate of convergence is an important aspect when considering metaheuristic optimization techniques; hence, to ensure that this factor was considered, uniformity was applied throughout the experiments. The proportional and resonant gains were given an upper limit of 100 and a lower limit of 0. For application of the algorithm, the DFIG was operated at a speed of 150 rad/s. The simulation was carried out on MATLAB/Simulink, and the simulation was run for 2 s. Only single-phase voltage dips were investigated. The voltage dips considered were at 5% and 10%. To analyze the steady-state error, the output was obtained in intervals of 0.1 s. To analyze the overshoot and steady-state ripple, the output was obtained in intervals of 0.01 s. To ensure that the system was given sufficient time to reach steady state, the results were analyzed from 1.5 to 2 s. The rotor direct and quadrature current references were kept constant at 1434.6 A and 1281 A, respectively.

6.3.1. Case a: Voltage Unbalance of 5%

Upon application of the three swarm intelligence algorithms to the PR controller for a voltage unbalance of 5%, the resultant proportional and resonant gain for the two controllers can be observed in Table 6.1. Figures 6.5–6.12 depict the results of applying PSO to the controller design. Figures 6.5 and 6.7 depict the three-phase stator voltage and current, respectively, with Figures 6.6 and 6.8 displaying the three-phase rotor voltage and current, respectively. Figures 6.9 and 6.11 are utilized for analyzing the steady-state error, whereas Figures 6.10 and 6.12 are used to observe the overshoot and steady-state ripple. From Figure 6.9, the average rotor direct steady-state current is 1507.9 A. This correlates to an error percentage of 5.46%. In Figure 6.10, the overshoot of the rotor direct current reaches 1767 A, which translates to a percentage of 23.17%

Table 6.1: Controller gains for 5% dip using PR controller for each swarm technique

	PSO	BA	GTO
$K_{p(idr)}$	68.9238	96.2194	1.3393
$K_{r(idr)}$	86.9757	1.96022	6.5823
$K_{p(iqr)}$	63.0181	87.7541	1.3586
$K_{r(iqr)}$	2.07452	15.6185	0.55515

Additionally, from Figure 6.10, the steady-state ripple of the rotor direct current is observed to be 373.5 A. From Figure 6.11, it is seen that the average rotor quadrature steady-state current is 1249.1 A. This correlates to an error percentage of 2.49%. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.12, are 1850 A and 342 A, respectively. This translates to an overshoot percentage of 44.42%. Figures 6.13–6.19 depict the results of applying BA to the controller design. Figures 6.5 and 6.13 depict the three-phase stator voltage and current, respectively, with Figures 6.14 and 6.15 displaying the three-phase rotor voltage and

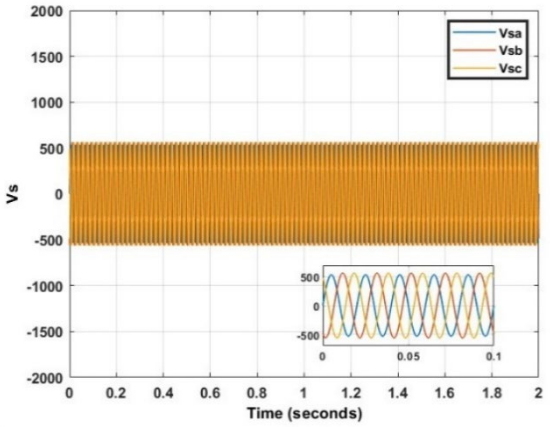


Figure 6.5: Three-phase stator voltage at 5% dip

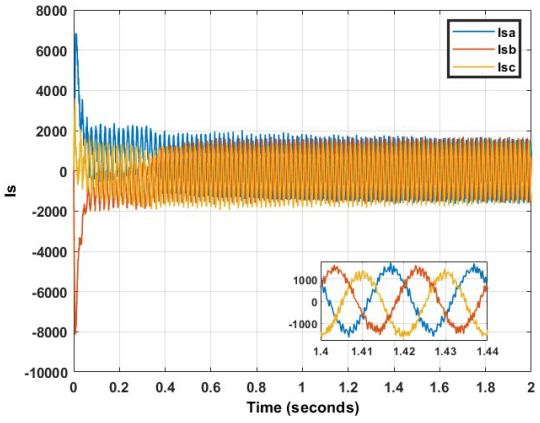


Figure 6.6: Three-phase stator current using PSO at 5% dip

current, respectively. Figures 6.16 and 6.18 are utilized for analyzing the steady-state error, whereas Figures 6.17 and 6.19 are used to observe the overshoot and steady-state ripple. From Figure 6.16, the average rotor direct steady-state current is 1471.3 A. This correlates to an error percentage of 2.56%.

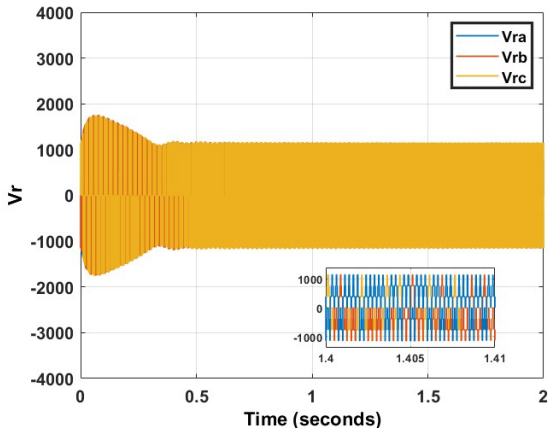


Figure 6.7: Three-phase rotor voltage using PSO at 5% dip

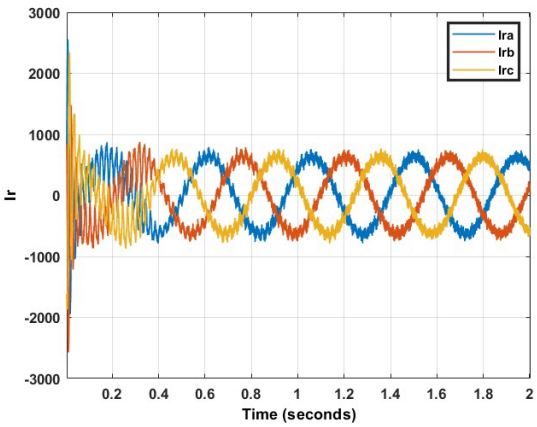


Figure 6.8: Three-phase rotor current using PSO at 5% dip

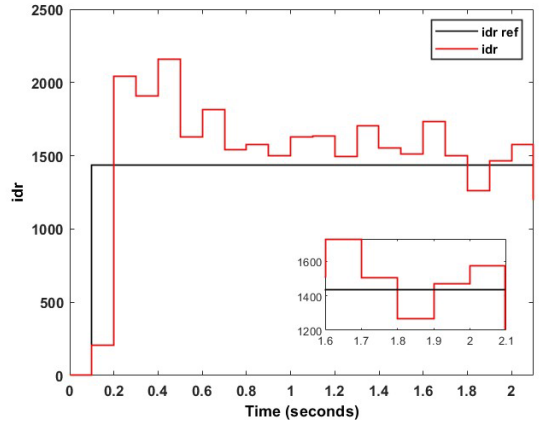


Figure 6.9: Steady-state error of i_{dr} using PSO at 5% dip

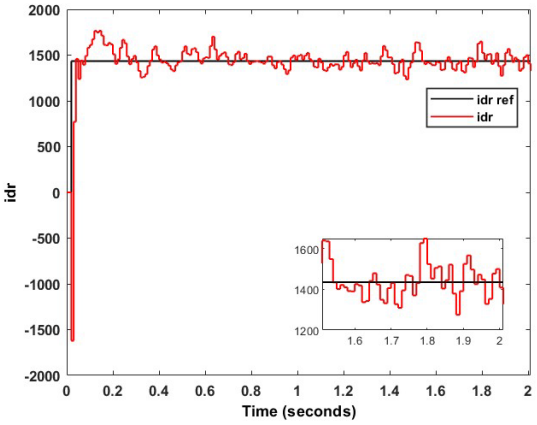


Figure 6.10: Overshoot and ripple of i_{dr} using PSO at 5% dip

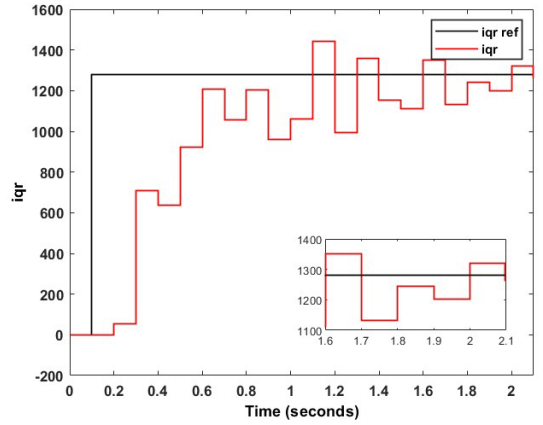


Figure 6.11: Steady-state error of i_{qr} using PSO at 5% dip

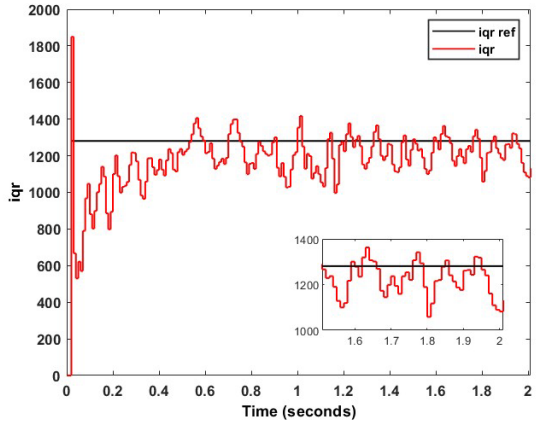


Figure 6.12: Overshoot and ripple of i_{qr} using PSO at 5% dip

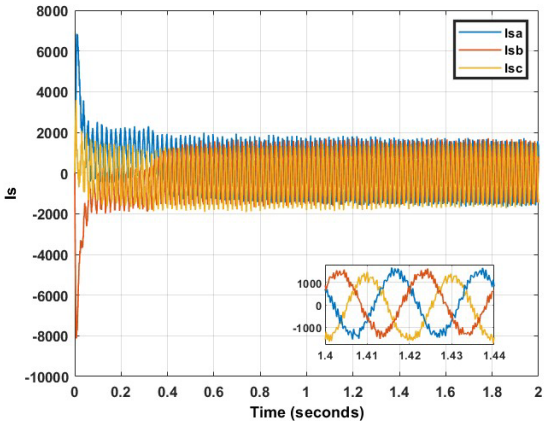


Figure 6.13: Three-phase stator current using BA at 5% dip

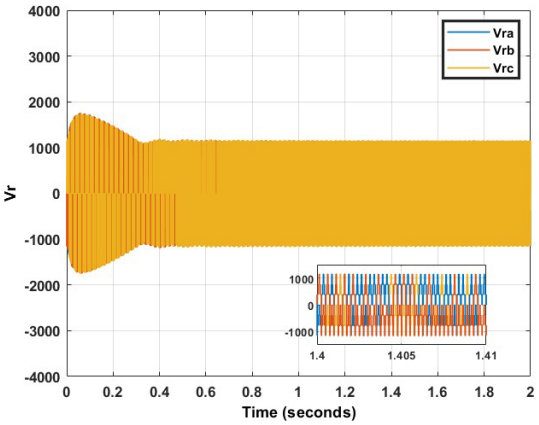


Figure 6.14: Three-phase rotor voltage using BA at 5% dip

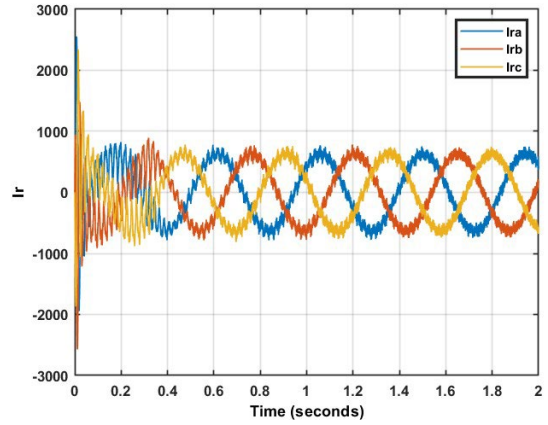


Figure 6.15: Three-phase rotor current using BA at 5% dip

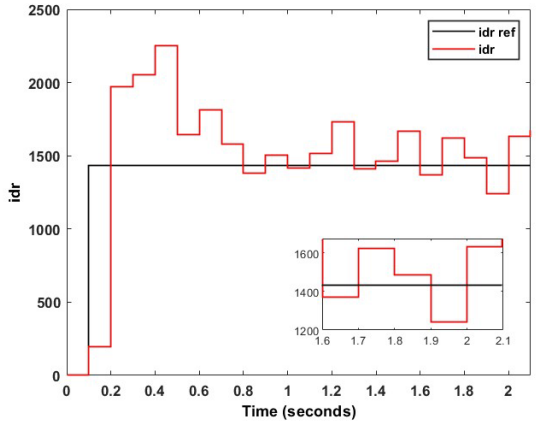


Figure 6.16: Steady-state error of i_{dr} using BA at 5% dip

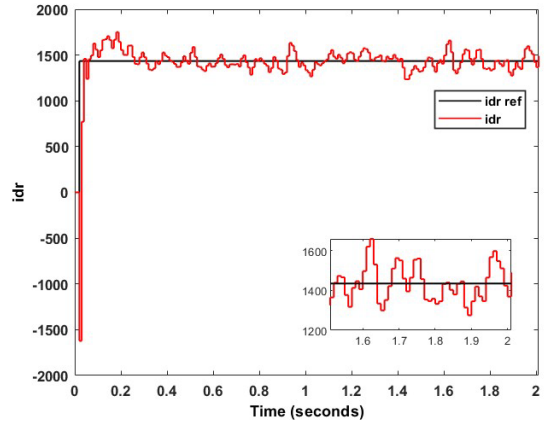


Figure 6.17: Overshoot and ripple of i_{dr} using BA at 5% dip

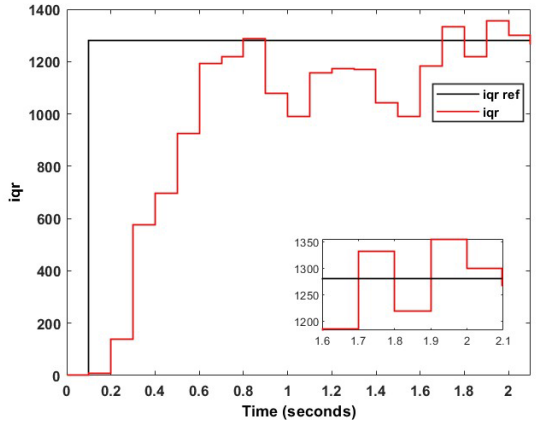


Figure 6.18: Steady-state error of i_{qr} using BA at 5% dip

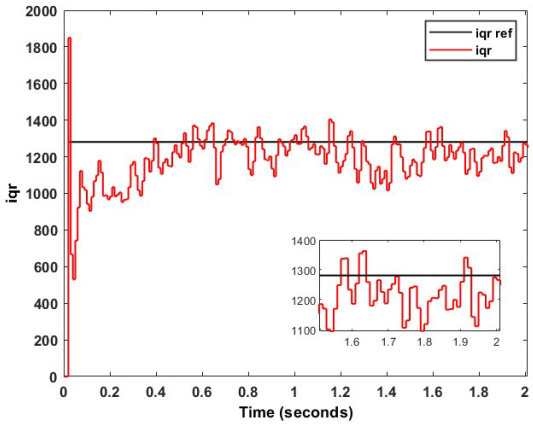


Figure 6.19: Overshoot and ripple of i_{qr} using BA at 5% dip

In figure 6.17, the overshoot of the rotor direct current reaches 1751 A, which translates to a percentage of 22.05%. Additionally, from Figure 6.17, the steady-state ripple of the rotor direct current is observed to be 383 A. From Figure 6.18, it is seen that the average rotor quadrature steady-state current is 1277.6 A. This correlates to an error percentage of 0.27%. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.19, are 1850 A and 267.9 A, respectively. This translates to an overshoot percentage of 44.42%.

Figures 6.20–6.23 depict the results of applying GTO to the controller design. Figures 6.5 and 6.20 depict the three-phase stator voltage and current, respectively, with Figures 6.21 and 6.22 displaying the three-phase rotor voltage and current, respectively. Figures 6.23 and 6.25 are utilized for analyzing the steady-state error, whereas Figures 6.24 and 6.26 are used to observe the overshoot and steady-state ripple. From Figure 6.23, the average rotor direct steady-state current is 1464 A. This correlates to an error percentage of 2.05%. In Figure 6.24, the overshoot of the rotor direct current reaches 1767 A, which translates to a percentage of 22.75%. Additionally, from Figure 6.24, the steady-state ripple of the rotor direct current is observed to be 97.8 A. From Figure 6.25, it is seen that the average rotor quadrature steady-state current is 1300.6 A. This correlates to an error percentage of 1.53%. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.26, are 1813 A and 128 A, respectively. This translates to an overshoot percentage of 41.53%.

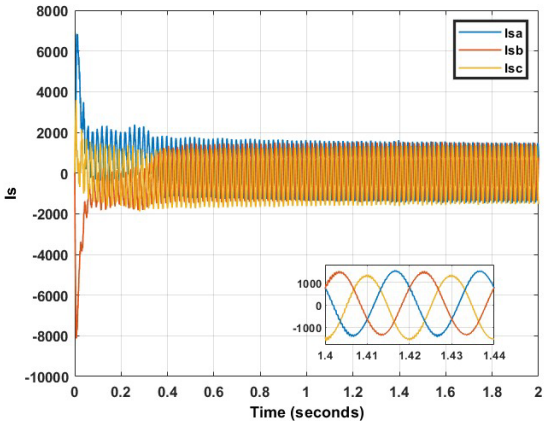


Figure 6.20: Three-phase stator current using GTO at 5% dip

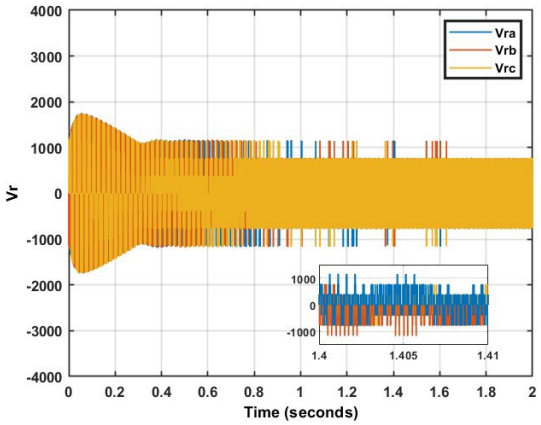


Figure 6.21: Three-phase rotor voltage using GTO at 5% dip

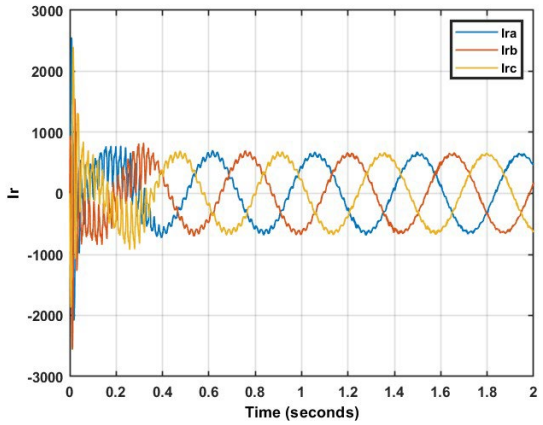


Figure 6.22: Three-phase rotor current using GTO at 5% dip

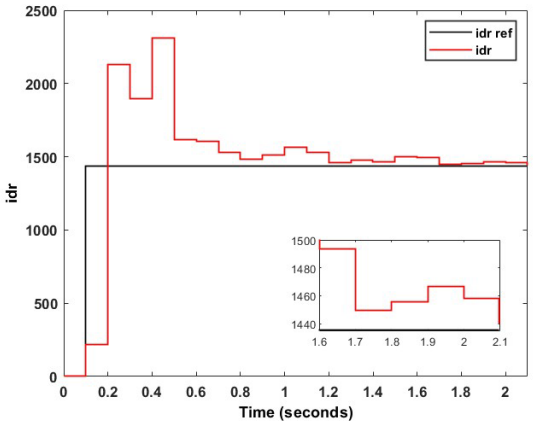


Figure 6.23: Steady-state error of i_{dr} using GTO at 5% dip

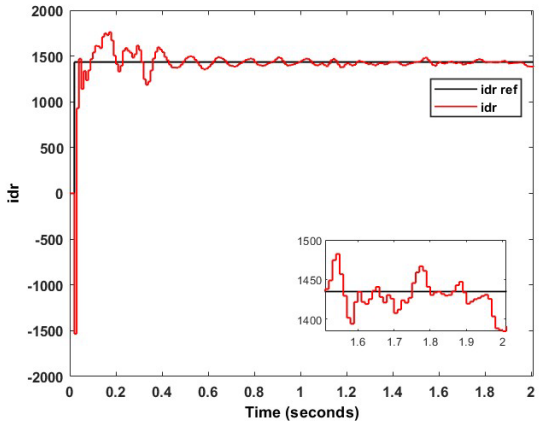


Figure 6.24: Overshoot and ripple of i_{dr} using GTO at 5% dip

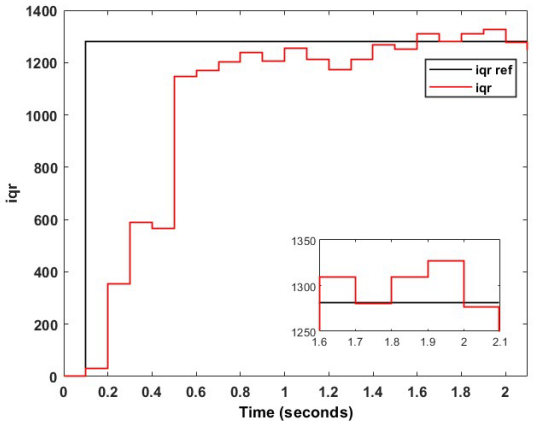


Figure 6.25: Steady-state error of i_{qr} using GTO at 5% dip

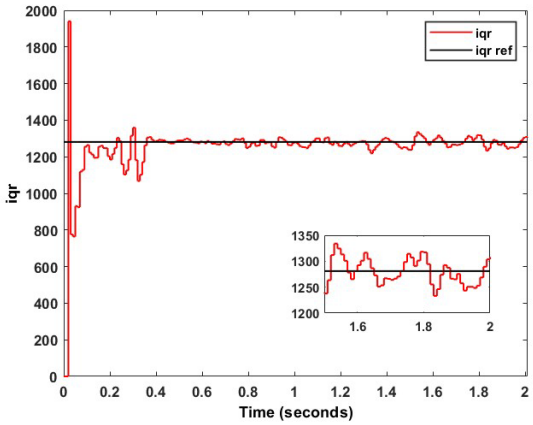


Figure 6.26: Overshoot and ripple of i_{qr} using GTO at 5% dip

The results from experiment *a* are displayed in Table 6.2. From Table 6.2, it is observed that GTO produced the best result on four out of the six occasions. BA proved superior to GTO with respect to the rotor direct current overshoot and rotor quadrature current steady-state error. However, this superiority is minute and negligible as compared to the large steady-state ripple produced by BA as compared to GTO. This holds true for both the direct

and quadrature rotor current. Furthermore, it is observed that GTO is the only algorithm to not be placed last for any of the six measurements. This points to a superior stability of GTO.

Table 6.2: Summary of results obtained for 5% dip using PR control for each swarm technique

	PSO	BA	GTO
Steady-state error i_{dr} (%)	5.46	2.56	**2.05**
Overshoot i_{dr} (%)	23.17	**22.05**	22.75
Ripple i_{dr} (A)	373.5	383	**97.8**
Steady-state error i_{qr} (%)	2.49	**0.27**	1.53
Overshoot i_{qr} (%)	44.2	44.42	**41.53**
Ripple i_{qr} (A)	304.2	267.9	**128**

6.3.2. Case b: Voltage Unbalance of 10%

Upon application of the three swarm intelligence algorithms to a voltage unbalance of 10%, the resultant proportional and resonant gain for the two controllers can be observed in Table 6.3. Once again, not much can be deduced from just analyzing the controller gains; hence, analysis of the waveforms is of paramount importance.

Table 6.3: Controller gains for 10% dip using PR control for each swarm technique

	PSO	BA	GTO
K_{idr}	99.8036	1.29052	1.3436
R_{idr}	25.7434	35.745	0.18306
K_{iqr}	94.0661	2.19815	1.3559
K_{iqr}	0	30.4797	0.0014881

Figures 6.27–6.34 depict the results of applying PSO to the controller design. Figures 6.27 and 6.28 depict the three-phase stator voltage and current, respectively, with Figures 6.29 and 6.30 displaying the three-phase rotor voltage and current, respectively. Figures 6.31 and 6.33 are utilized for analyzing the steady-state error, whereas Figures 6.32 and 6.34 are used to observe the overshoot and steady-state ripple.

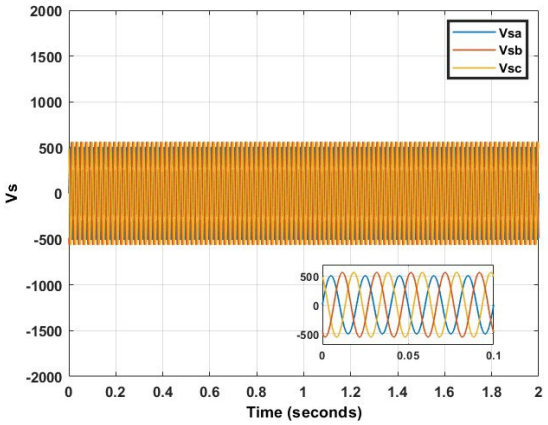


Figure 6.27: Three-phase stator voltage at 10% dip

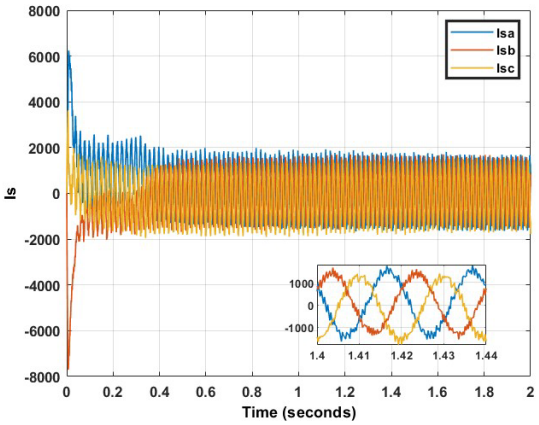


Figure 6.28: Three-phase stator current using PSO at 10% dip

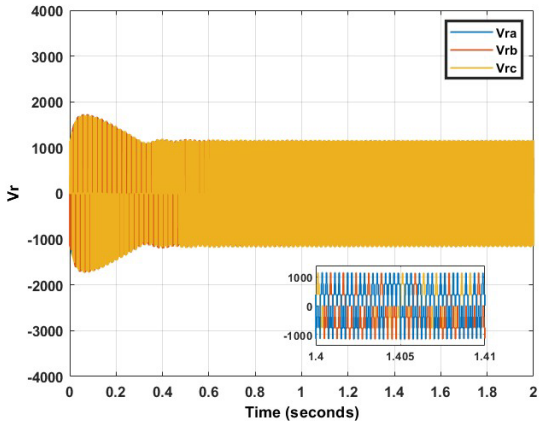


Figure 6.29: Three-phase rotor voltage using GTO at 10% dip

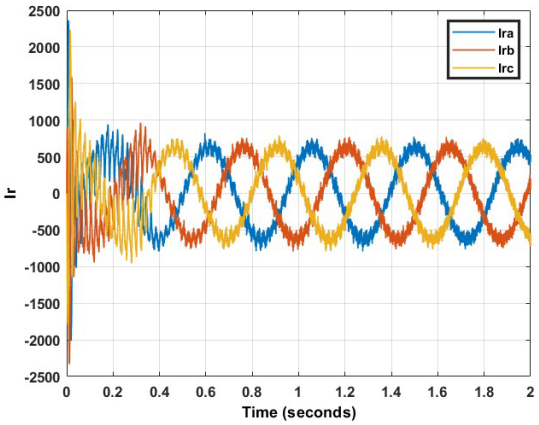


Figure 6.30: Three-phase rotor current using PSO at 10% dip

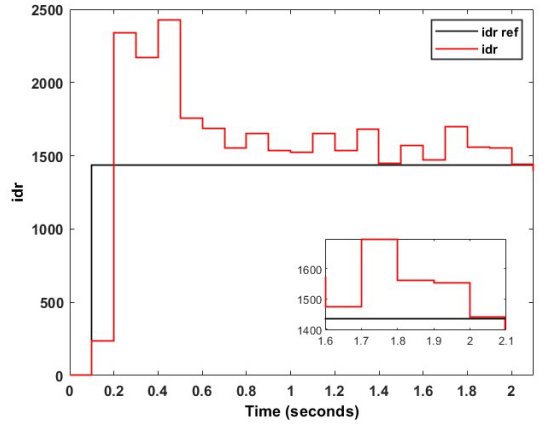


Figure 6.31: Steady-state error of i_{dr} using PSO at 10% dip

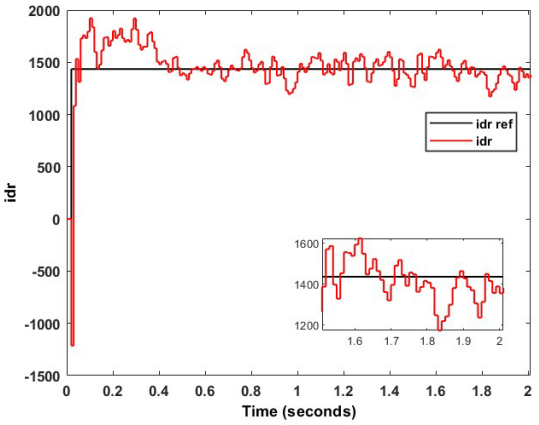


Figure 6.32: Overshoot and ripple of i_{dr} using PSO at 10% dip

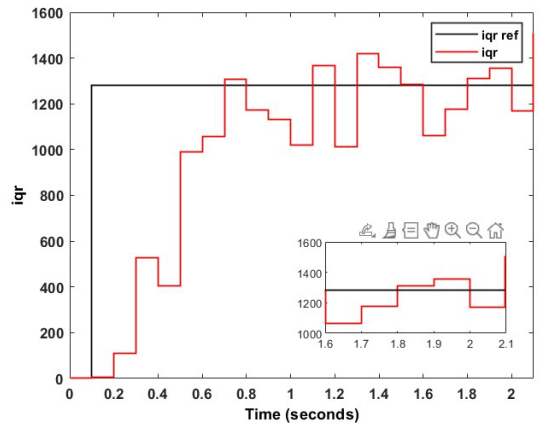


Figure 6.33: Steady-state error of i_{qr} using PSO at 10% dip

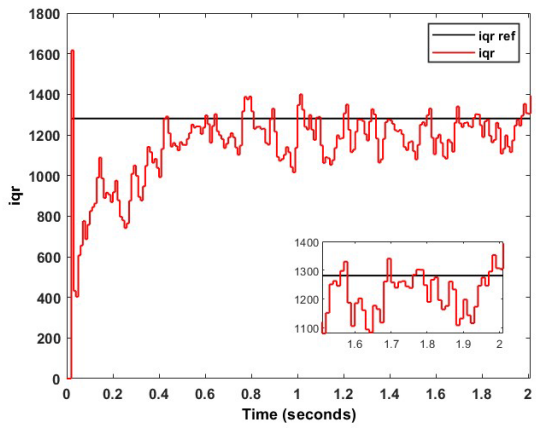


Figure 6.34: Overshoot and ripple of i_{qr} using PSO at 10% dip

From Figure 6.31, the average rotor direct steady-state current is 1504.4 A. This correlates to an error percentage of 4.87%. In Figure 6.32, the overshoot of the rotor direct current reaches 1922 A, which translates to a percentage of 33.97%. Additionally, from Figure 6.32, the steady-state ripple of the rotor direct current is observed to be 449.5 A. From Figure 6.33, it is seen that the average rotor quadrature steady-state current is 1215.4 A. This correlates to an error percentage of 5.12%. The overshoot and steady-state ripple of the rotor quadrature current,

as observed in Figure 6.34, are 1616.5 A and 272.5 A, respectively. This translates to an overshoot percentage of 26.19%.

Figures 6.35–6.41 depict the results of applying BA to the controller design. Figures 6.27 and 6.35 depict the three-phase stator voltage and current, respectively, with Figures 6.36 and 6.37 displaying the three-phase rotor voltage and current, respectively. Figures 6.38 and 6.40 are utilized for analyzing the steady-state error, whereas Figures 6.39 and 6.41 are used to observe the overshoot and steady-state ripple.

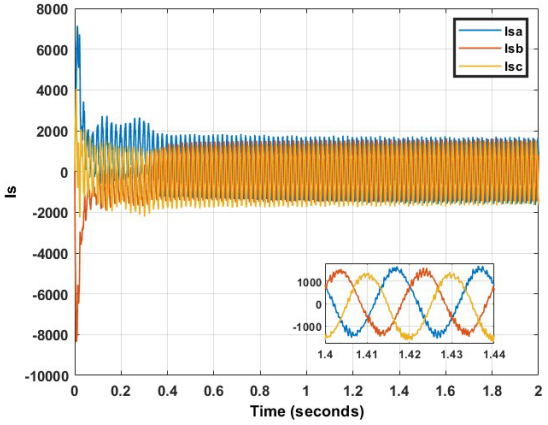


Figure 6.35: Three-phase stator current using BA at 10% dip

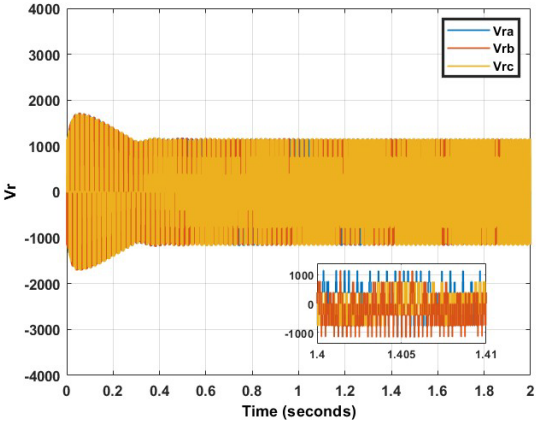


Figure 6.36: Three-phase rotor voltage using BA at 10% dip

From Figure 6.38, the average rotor direct steady-state current is 1474.8 A. This correlates to an error percentage of 2.80%. In Figure 6.39, the overshoot of the rotor direct current reaches 2287 A, which translates to a percentage of 59.42%. Additionally, from Figure 6.39, the steady-state ripple of the rotor direct current is observed to be 130.5 A. From Figure 6.40, it is seen that the average rotor quadrature steady-state current is 1109.4 A. This correlates to an error percentage of 13.4%. The overshoot and steady-state ripple

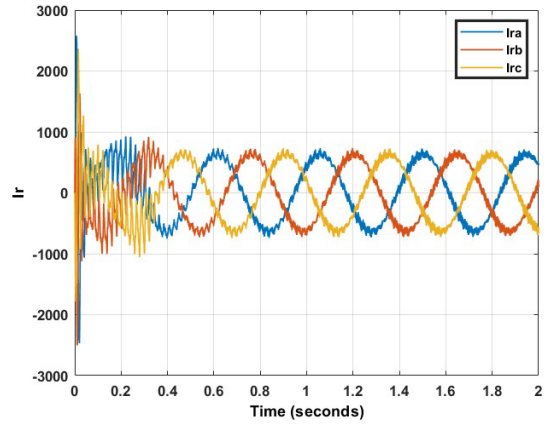


Figure 6.37: Three-phase rotor current using BA at 10% dip

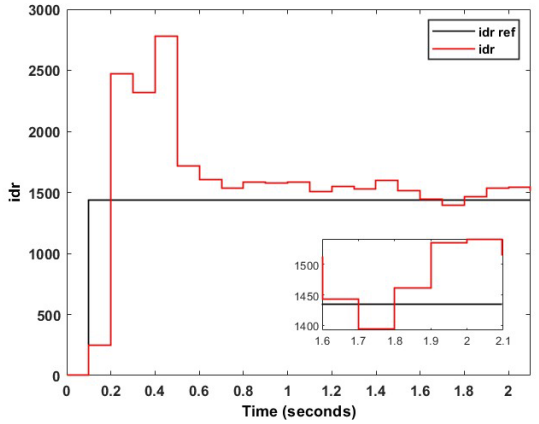


Figure 6.38: Steady-state error of i_{dr} using BA at 10% dip

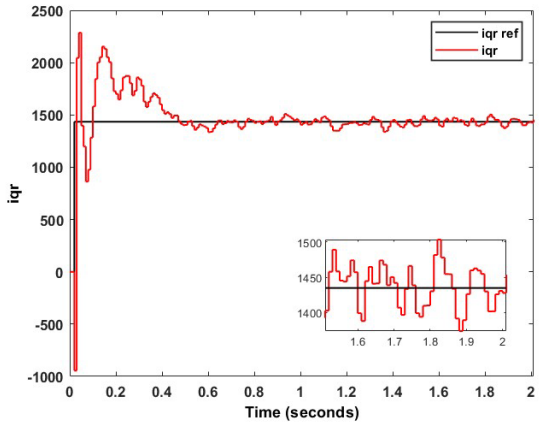


Figure 6.39: Overshoot and ripple of i_{dr} using BA at 10% dip

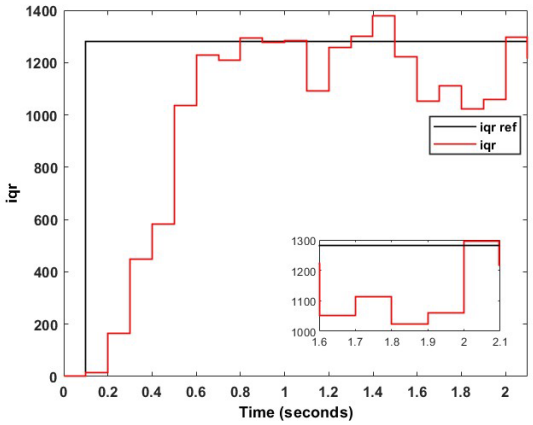


Figure 6.40: Steady-state error of i_{qr} using BA at 10% dip

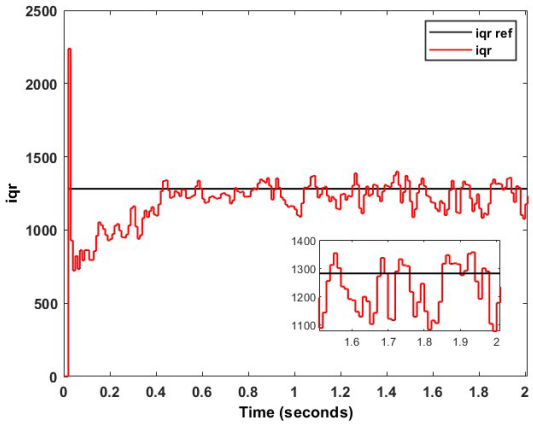


Figure 6.41: Overshoot and ripple of i_{qr} using BA at 10% dip

of the rotor quadrature current, as observed in Figure 6.41, are 2238 A and 280 A, respectively. This translates to an overshoot percentage of 74.71%. Figures 6.42–6.48 depict the results of applying GTO to the controller design. Figures 6.27 and 6.42 depict the three-phase stator voltage and current, respectively, with Figures 6.43 and 6.44 displaying the three-phase rotor voltage and current, respectively. Figures 6.45 and 6.47 are utilized for analyzing the steady-state error, whereas Figures 6.48 and 6.48 are used to observe the overshoot, and steady-state ripple. From Figure 6.45, the average rotor direct steady-state current is 1409.4 A. This correlates to an error percentage of 1.76%. In Figure 6.46, the overshoot of the rotor direct current reaches 1902 A, which translates to a percentage of 38.58%. Additionally, from Figure 6.46, the steady-state ripple of the rotor direct current is observed to be 65 A. Both the overshoot and steady-state ripple of i_{dr} are excellent and are well within acceptable engineering limits. From Figure 6.47, it is seen that the average rotor quadrature steady-state current is 1264.5 A. This correlates to an error percentage of 1.29%. As with the rotor direct current, this error percentage is extremely low, and once again well within acceptable ranges. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.48, are 1757 A and 107 A, respectively. This translates to an overshoot percentage of 37.16%.

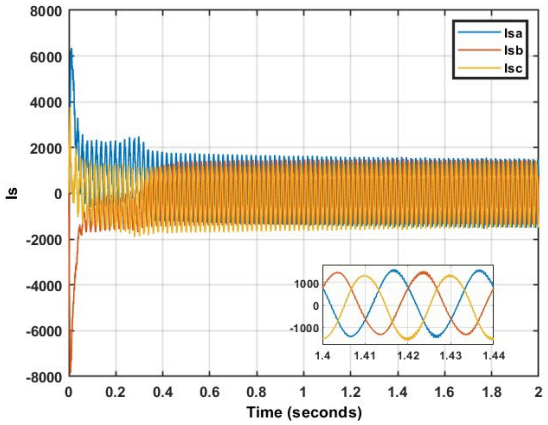


Figure 6.42: Three-phase stator current using GTO at 10% dip

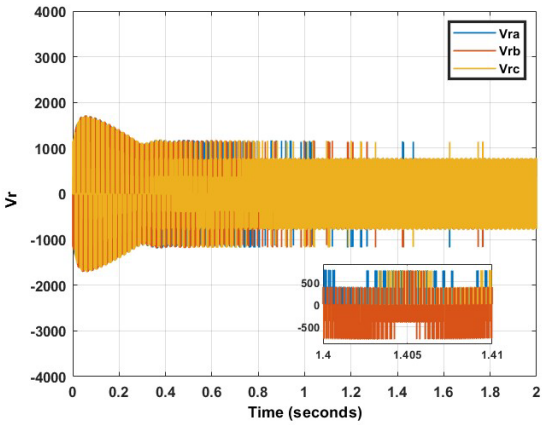


Figure 6.43: Three-phase rotor voltage using GTO at 10% dip

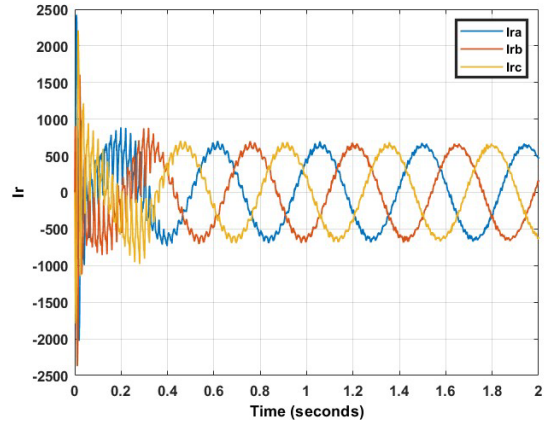


Figure 6.44: Three-phase rotor current using GTO at 10% dip

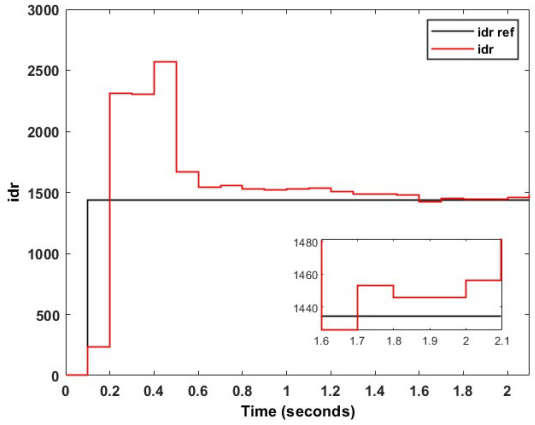


Figure 6.45: Steady-state error of i_{dr} using GTO at 10% dip

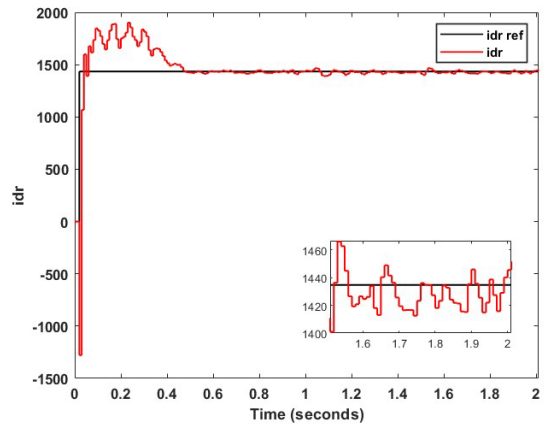


Figure 6.46: Overshoot and ripple of i_{dr} using GTO at 10% dip

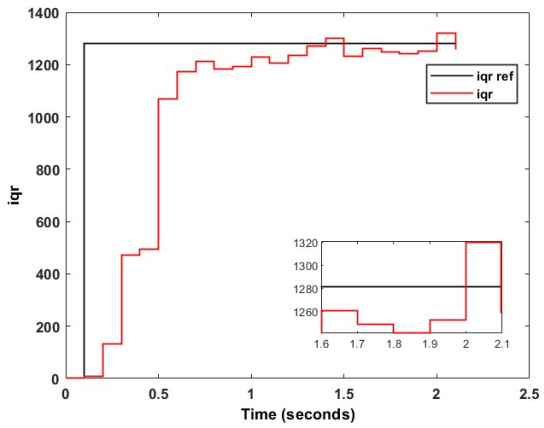


Figure 6.47: Steady-state error of i_{qr} using GTO at 10% dip

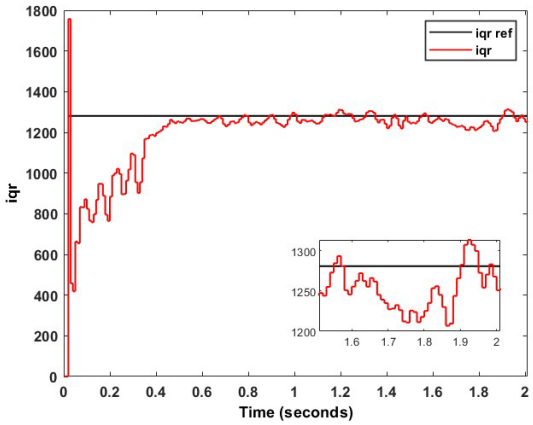


Figure 6.48: Overshoot and ripple of i_{qr} using GTO at 10% dip

The results from experiment b are displayed in Table 6.4. From Table 6.4, it is observed that GTO produced the best result on four out of the six occasions. PSO proved superior to GTO with respect to the rotor direct and quadrature current overshoot. However, as in case *a*, this superiority is minute and negligible as compared to the large steady-state ripple produced by PSO as compared to GTO. This holds true for both the direct and quadrature rotor current. Furthermore, it is observed that, once again, GTO is the only algorithm to not be placed last for any of the six measurements. This points to a superior stability of the GTO.

Table 6.4: Summary of results obtained for 10% dip using PR control for each swarm technique

	PSO	**BA**	**GTO**
Steady-state error i_{dr} (%)	4.87	2.80	**1.76**
Overshoot i_{dr} (%)	**33.97**	59.42	38.58
Ripple i_{dr} (A)	449.5	130.50	**65**
Steady-state error i_{qr} (%)	5.12	13.40	**1.29**
Overshoot i_{qr} (%)	**26.19**	74.71	37.16
Ripple i_{qr} (A)	272.5	280	**107**

6.3.3. Case c: Dual-Phase Voltage Unbalance of 7.5%

Upon application of the three swarm intelligence algorithms to a dual-phase voltage unbalance of 7.5%, the resultant proportional and resonant gain for the two controllers can be observed in Table 6.5. Once again, not much can be deduced from just analyzing the controller gains; hence, analysis of the waveforms is of paramount importance.

Table 6.5: Controller gains for 7.5% dip using PR controller for each swarm technique

	PSO	**BA**	**GTO**
K_{idr}	72.8854	5.86008	1.317
R_{idr}	17.7069	20.5559	0.51571
K_{iqr}	67.971	11.8835	1.2977
K_{iqr}	9.58148	96.8331	1.6227

Figures 6.49–6.56 depict the results of applying PSO to the controller design. Figures 6.49 and 6.50 depict the three-phase stator voltage and current, respectively, with Figures 6.51 and 6.52 displaying the three-phase rotor voltage and current, respectively. Figures 6.53 and 6.55 are utilized for analyzing the steady-state error, whereas Figures 6.54 and 6.56 are used to observe the overshoot and steady-state ripple.

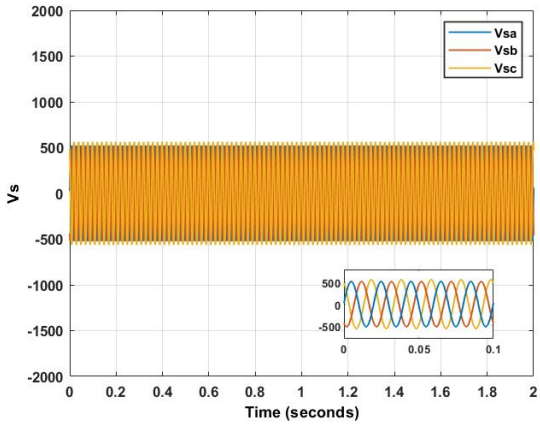


Figure 6.49: Three-phase stator voltage at 7.5% dip

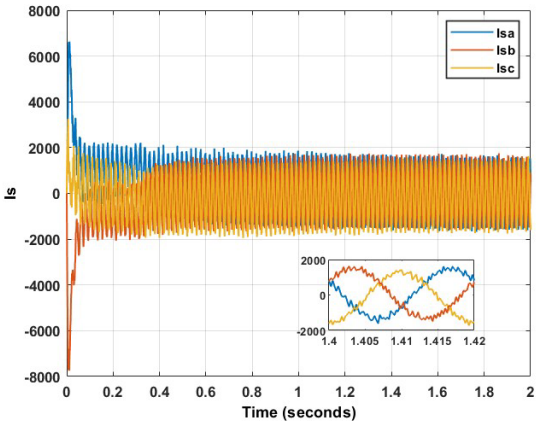


Figure 6.50: Three-phase stator current using PSO at 7.5% dip

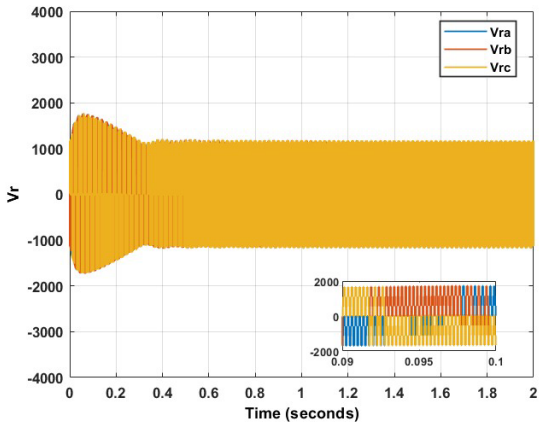


Figure 6.51: Three-phase rotor voltage using PSO at 7.5% dip

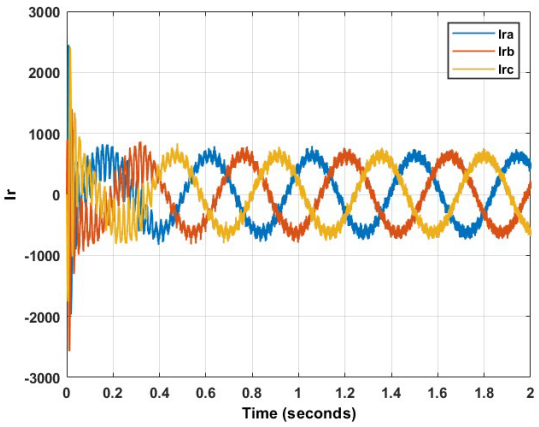


Figure 6.52: Three-phase rotor current using PSO at 7.5% dip

From Figure 6.53, the average rotor direct steady-state current is 1577A. This correlates to an error percentage of 9.92%. In Figure 6.54, the overshoot of the rotor direct current reaches 1502 A, which translates to a percentage of 5.01%. Additionally, from Figure 6.54, the steady-state ripple of the rotor direct current is observed to be 341.5 A. From Figure 6.55, it is seen that the average rotor quadrature steady-state current is 1307.5 A. This correlates to an error percentage of 2.03%. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.56, are 1874.5 A and 273 A, respectively. This translates to an overshoot percentage of 46.33%.

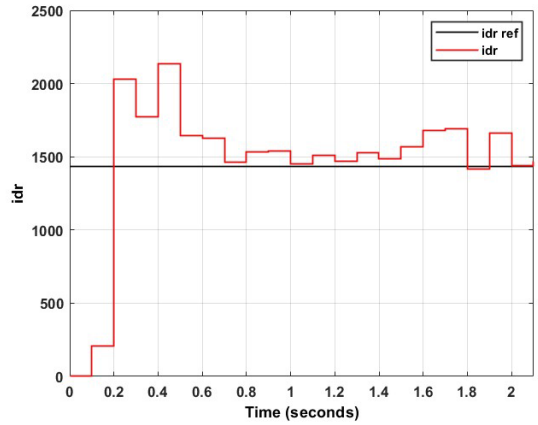


Figure 6.53: Steady-state error of i_{dr} using PSO at 7.5% dip

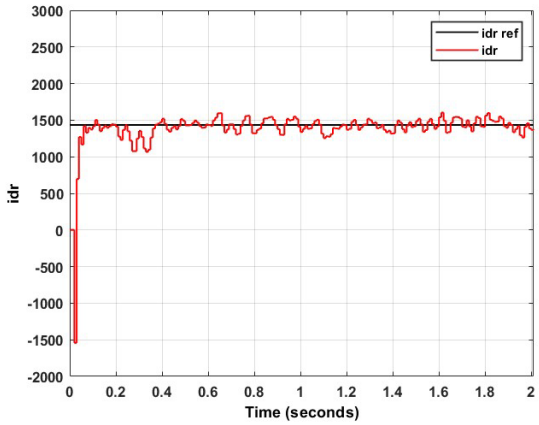


Figure 6.54: Overshoot and ripple of i_{dr} using PSO at 7.5% dip

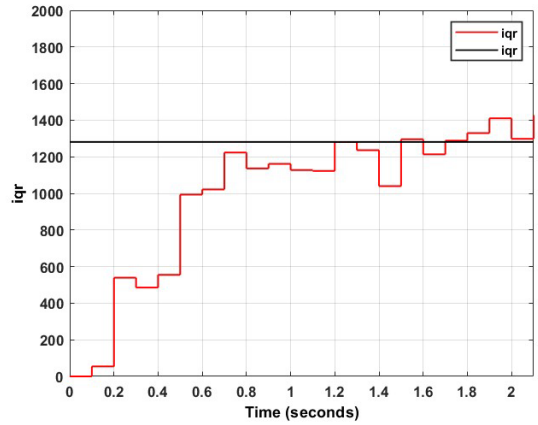


Figure 6.55: Steady-state error of i_{qr} using PSO at 7.5% dip

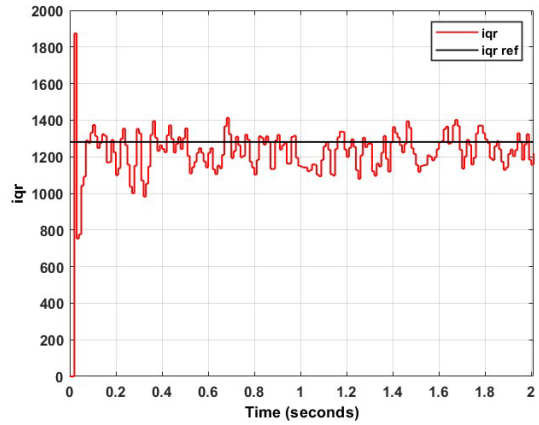


Figure 6.56: Overshoot and ripple of i_{qr} using PSO at 7.5% dip

Figures 6.57–6.63 depict the results of applying BA to the controller design. Figures 6.49 and 6.57 depict the three-phase stator voltage and current, respectively, with Figures 6.58 and 6.59 displaying the three-phase rotor voltage and current, respectively. Figures 6.60 and 6.62 are utilized for analyzing the steady-state error, whereas Figures 6.61 and 6.63 are used to observe the overshoot and steady-state ripple. From Figure 6.60, the average rotor direct steady-state current is 1535 A. This correlates to an error percentage of 7%. In Figure 6.61, the overshoot of the rotor direct current reaches 1651 A, which translates to a percentage of 15.08%. Additionally, from Figure 6.61, the steady-state ripple of the rotor direct current is observed to be 303 A. From Figure 6.62, it is seen that the average rotor quadrature steady-state current is 1270.2 A. This correlates to an error percentage of 0.84%. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.63, are 2366 A and 412 A, respectively. This translates to an overshoot percentage of 84.7%. Figures 6.64–6.70 depict the results of applying GTO to the controller design. Figures 6.49 and 6.64 depict the three-phase stator voltage and current, respectively, with Figures 6.65 and 6.66 displaying the three-phase rotor voltage and current, respectively. Figures 6.67 and 6.69 are utilized for analyzing the steady-state error, whereas Figures 6.68 and 6.70 are used to observe the overshoot, and steady-state ripple. From Figure 6.67, the average rotor direct steady-state current is 1480 A.

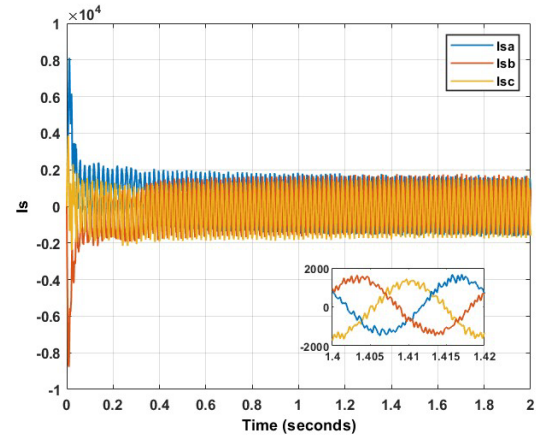


Figure 6.57: Three-phase stator current using BA at 7.5% dip

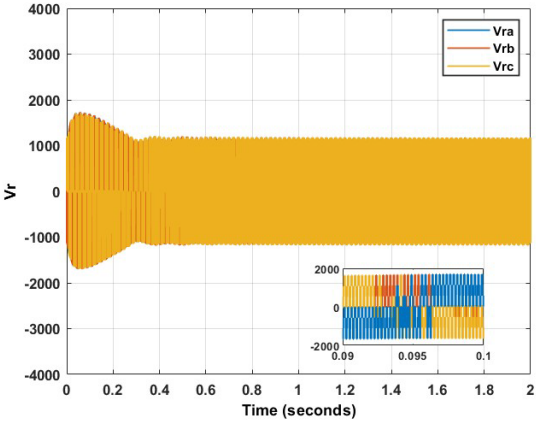


Figure 6.58: Three-phase rotor voltage using BA at 7.5% dip

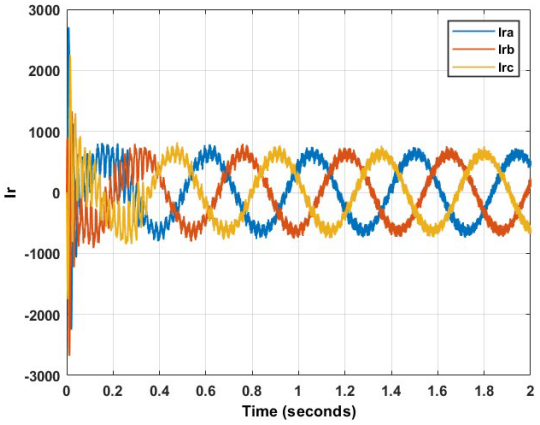


Figure 6.59: Three-phase rotor current using BA at 7.5% dip

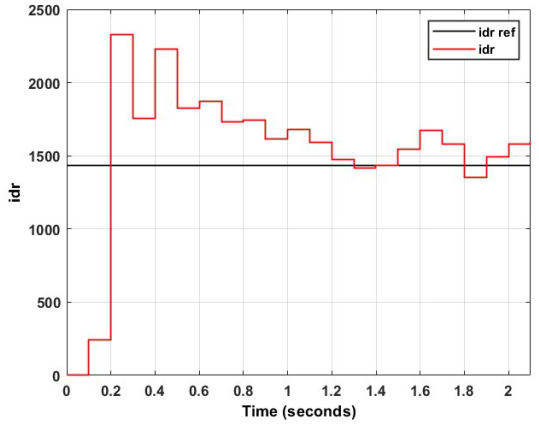


Figure 6.60: Steady-state error of i_{dr} using BA at 7.5% dip

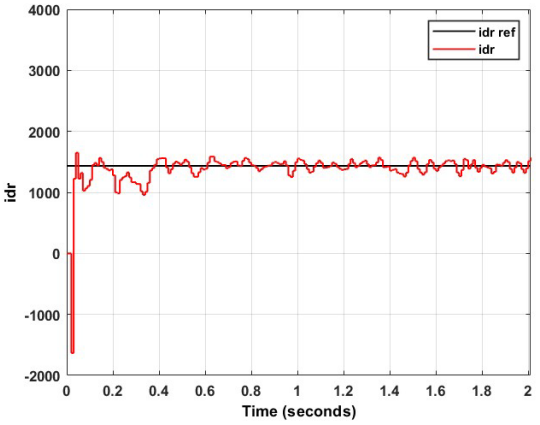


Figure 6.61: Overshoot and ripple of i_{dr} using PSO at 7.5% dip

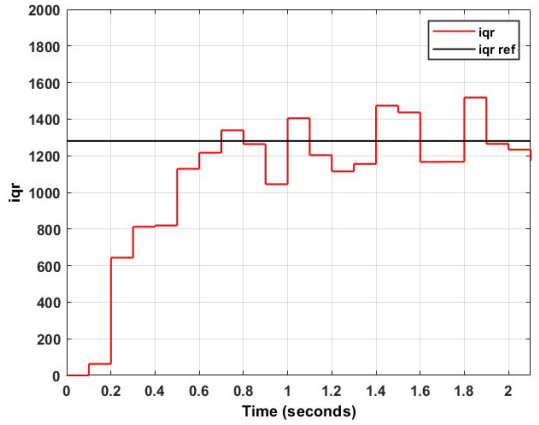


Figure 6.62: Steady-state error of i_{qr} using BA at 7.5% dip

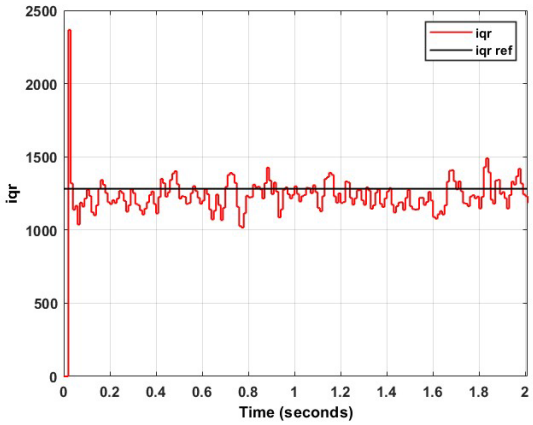


Figure 6.63: Overshoot and ripple of i_{qr} using BA at 7.5% dip

This correlates to an error percentage of 3.16%. In Figure 6.68, the overshoot of the rotor direct current reaches 1472 A, which translates to a percentage of 2.6%. Additionally, from Figure 6.68, the steady-state ripple of the rotor direct current is observed to be 67.5 A. Both the overshoot and steady-state ripple of i_{dr} are excellent and are well within acceptable engineering limits. From Figure 6.69, it is seen that the average rotor quadrature steady-state current is 1287.7 A. This correlates to an error percentage of 0.52%. As with the rotor direct current, this error percentage is extremely low, and once again well within acceptable ranges. The overshoot and steady-state ripple of the rotor quadrature current, as observed in Figure 6.70, are 1938 A and 42.5 A, respectively. This translates to an overshoot percentage of 51.29%.

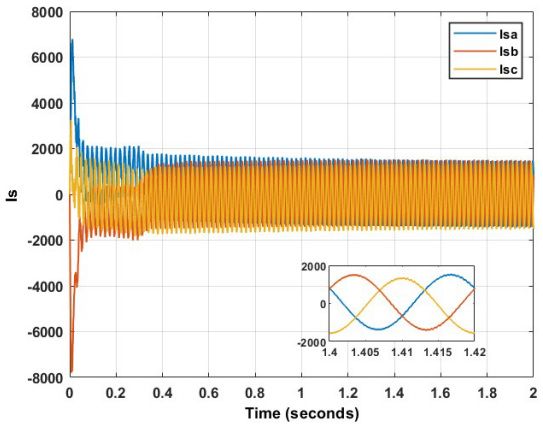


Figure 6.64: Three-phase stator current using GTO at 7.5% dip

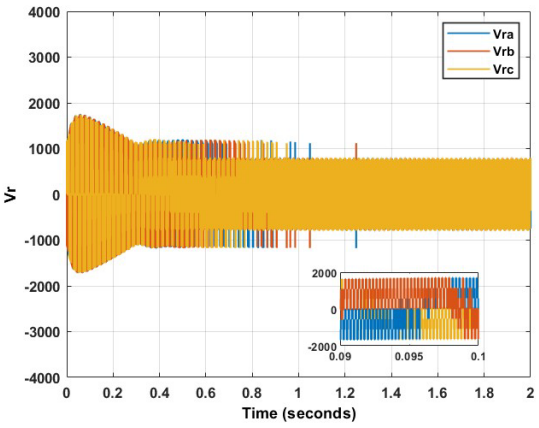


Figure 6.65: Three-phase rotor voltage using GTO at 7.5% dip

The results from experiment 2b are displayed in Table 6.6. From Table 6.6, it is observed that GTO produced the best result on five out of the six occasions. PSO proved superior to GTO with respect to the rotor quadrature current overshoot. However, as in cases *a* and *b*, this superiority is minute and negligible as compared to the large steady-state ripple produced by PSO as compared to GTO. This holds true for both the direct and quadrature rotor current. Furthermore, it is observed that, once again, GTO is the only algorithm to not be placed last for any of the six measurements. This points to a superior stability of the GTO.

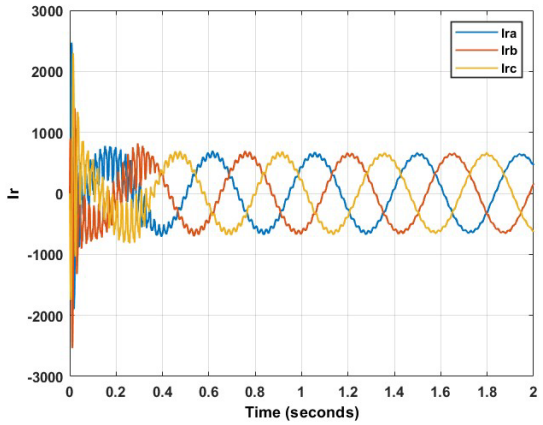


Figure 6.66: Three-phase rotor current using GTO at 7.5% dip

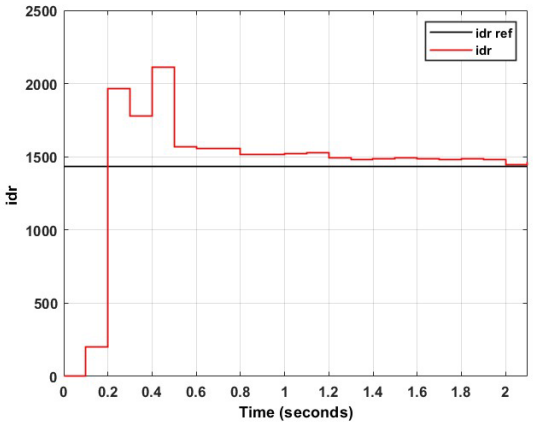


Figure 6.67: Steady-state error of i_{dr} using GTO at 7.5% dip

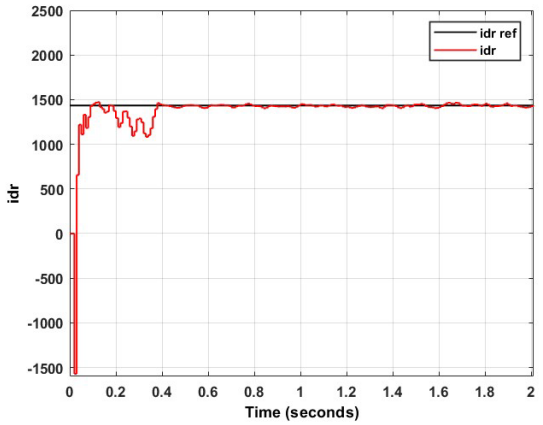


Figure 6.68: Overshoot and ripple of i_{dr} using GTO at 7.5% dip

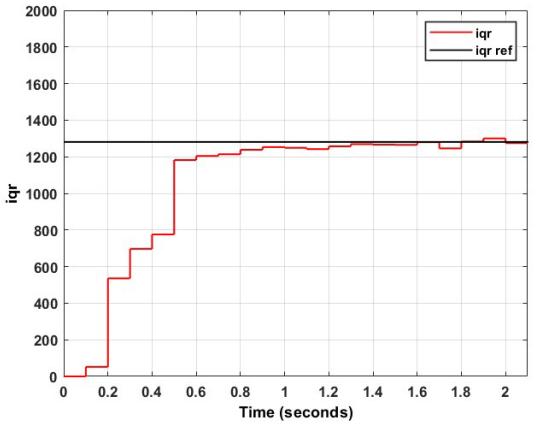


Figure 6.69: Steady-state error of i_{qr} using GTO at 7.5% dip

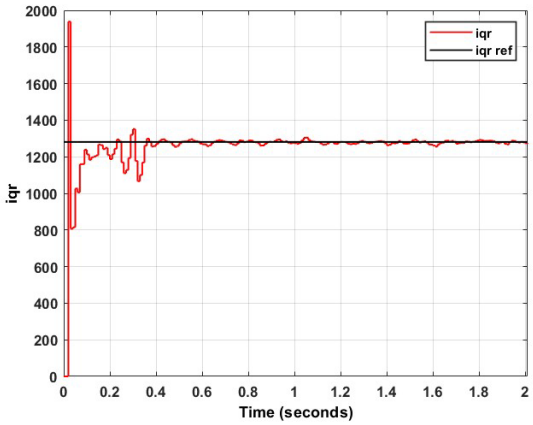


Figure 6.70: Overshoot and ripple of i_{qr} using GTO at 7.5% dip

Table 6.6: Summary of results obtained for dual-phase dip using PR control for each swarm technique

	PSO	BA	GTO
Steady-state error i_{dr} (%)	9.92	7	**3.16**
Overshoot i_{dr} (%)	5.01	15.08	**2.6**
Ripple i_{dr} (A)	341.5	303	**67.5**
Steady-state error i_{qr} (%)	2.03	0.84	**0.52**
Overshoot i_{qr} (%)	**46.33**	84.7	51.29
Ripple i_{qr} (A)	273	412	**42.5**

6.4. Robustness analysis of controller

From the results of the experiments, it is concluded that swarm intelligence performs much better when paired with the PR controller, as compared to with the dual vector controller. This is largely due to the smaller number of variables to optimize, thereby providing precision. When considering the use of computational intelligence, the aspect of non-parametric statistical tests should be considered. Tests such as the sign test, Wilcoxon test, and the Friedman test are important measures of an algorithm's relative performance [47]. In these tests, the algorithm is generally executed multiple times, and is compared to the performance of other algorithms. This may assist in determining the statistical significance of one algorithm relative to others. Owing to the nature of this research, and the time taken to execute one algorithm, these tests were omitted from this research.

Considering the resonant gains generated for case *b*, it is observed that GTO produced significantly lower resonant gains than both PSO and BA. This can be verified by Figures 6.71 and 6.72, which depict the bode plots of the direct and quadrature rotor current controllers when designed using the various techniques. It is evident that in both cases, the bat algorithm produces a superior frequency bandwidth to gorilla troops optimization. This may degrade the results of GTO during grid frequency perturbations. In general, grid frequencies do not drop by more than 3%, which in this case is 1.5 Hz. To analyze the robustness of the designed controllers under the effects of grid frequency perturbations, the system was tested at a frequency of 48.5 Hz, which correlates to a frequency deviation of 3%. The results obtained are displayed in Table 6.7. From Table 6.7, it is clearly evident that GTO produced the best result for all six measurements. Considering both overshoot percentages, PSO came close to GTO but fell short. The same can be deduced for the direct rotor current steady-state error. However, for both current values, GTO produced a significantly better steady-state ripple to PSO. Hence it can be concluded that despite the low frequency bandwidth, the controller designed using GTO is able to produce acceptable results even when subject to a grid frequency fluctuation of 3%.

Table 6.7: Summary of results for 10% dip and 3% perturbation using PR control

	PSO	BA	GTO
Steady-state error i_{dr} (%)	2	2.66	**1.79**
Overshoot i_{dr} (%)	31.12	47.36	**30.35**
Ripple i_{dr} (A)	430.8	16	**105.8**
Steady-state error i_{qr} (%)	7.99	10.46	**0.78**
Overshoot i_{qr} (%)	58.98	91.88	**57.18**
Ripple i_{qr} (A)	408.5	334.5	**123.5**

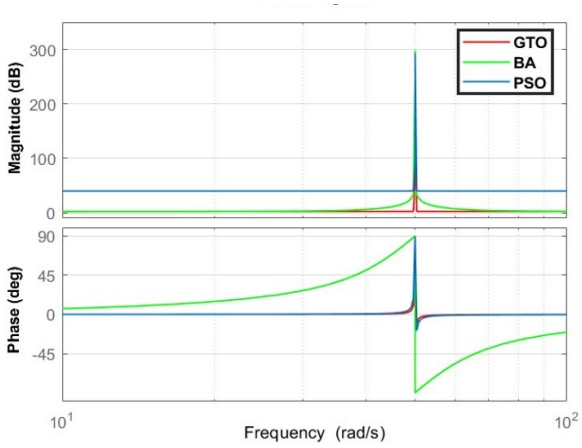


Figure 6.71: Bode plot of i_{dr} using at 10% dip using PR control

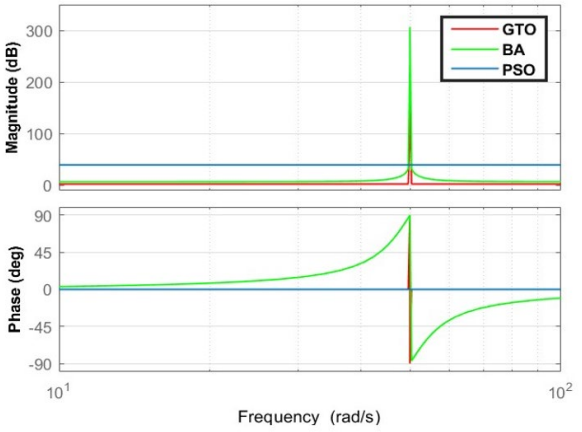


Figure 6.72: Bode plot of i_{qr} using at 10% dip using PR control

6.5. Conclusion

This chapter presented an investigation into the effects of applying swarm intelligence to the design of proportional–resonant controllers for control of a doubly fed induction generator subject to unbalanced grid voltages. Single-phase voltage unbalances of 5% and 10%, and a phase-to-phase voltage dip of 7.5% were investigated. Particle swarm optimization, the bat algorithm, and the gorilla troops optimization algorithm were applied and analyzed. The control method implemented was stator flux field-oriented control. The results were obtained as steady-state error, overshoot, and steady-state ripple. The results showed that due to the significantly reduced number of variables to be optimized, the proportional–resonant controller outperformed the dual vector controller. Considering PSO for the 5% voltage dip, the dual vector method produced superior results for the rotor direct current but was inferior to the PR controller for the rotor quadrature current. For the Bat Algorithm, the PR controller exhibited superior steady-state error and ripple values for both rotor currents. When GTO was applied to the dual vector controller, it was observed that the steady-state ripple was significantly smaller than that of the PR controller, however, the PR controller showed dominancy in all other instances.

Considering PSO for the 10% voltage dip, the dual vector method produced superior results for the rotor direct current but was inferior to the PR controller for the rotor quadrature current steady-state error and ripple. For the

Bat Algorithm, the PR controller exhibited superior steady-state error values for both rotor currents. For the dual vector method, however, a better overshoot was observed. When GTO was applied to the dual vector controller, it was observed that the steady-state ripple and overshoot were significantly smaller than that of the PR controller, however, this was at the expense of a much large steady-state error

Considering PSO for the 7.5% voltage dip, the dual vector method produced superior results with regard to the rotor current overshoot but was inferior to the PR controller in terms of steady-state error and steady-state ripple. It was observed that for the Bat Algorithm, the dual vector controller outperformed the PR controller in four of the six tests, the inferiority occurring in the rotor direct current ripple, and rotor quadrature current steady-state error. When applied to the PR controller, the GTO exhibited dominancy with regards to the rotor direct and quadrature current steady-state error and steady-state ripple.

Moreover, considering PR control, it was observed that GTO exhibited the best overall results. Furthermore, a robustness test of the proportional–resonant controller, in the form of a 3% grid frequency perturbation was carried out. The results showed that the gorilla troops optimization algorithm unanimously produced the best solution, in terms of all aspects. This indicates the growing potential of the application of such algorithms to controller design. It can also be concluded that as swarm intelligence evolves, their application in machine control becomes more promising. Another new swarm intelligence technique that should be considered is African vulture optimization algorithm [48].

Chapter 7 : Conclusion

The research work undertaken in this thesis aimed at thoroughly investigating the application of Metaheuristic Optimization Techniques to the control of the Doubly Fed Induction Generator. In particular, swarm intelligence was considered, and subsequently applied to the control of the DFIG when subject to symmetrical and asymmetrical voltage dips. The first chapter of this thesis dealt with the significance of this research, as well as any limitations and research questions to be answered. This gave the reader a thorough understanding of the remainder of the research work.

7.1. Summary of conclusions

In chapter 2, an in-depth analysis of fourteen swam intelligence techniques were carried out. These techniques were Particle Swarm Optimization, Bacteria Foraging Optimization Algorithm, Artificial Bee Colony, Whale Optimization Algorithm, Grey Wolf Optimizer, Crow Search Algorithm, Cuckoo Search Algorithm, Squirrel Search Algorithm, Shuffled Frog Leaping Algorithm, Sailfish Optimization, Moth Flame Optimization, Firefly Algorithm, Bat Algorithm and Antlion Optimizer. Each of these techniques were discussed in terms of inception, merits, demerits, and techniques proposed in current literature to mitigate any deficiencies. Upon completion of such, it was observed that Particle Swarm Optimization is the most established and researched method. This can be attributed to its fast convergence rate. Considering the demerits of PSO, several scholars have proposed mitigation techniques to ameliorate such. This is contrary to algorithms such as BFOA, where no demerits and subsequent mitigation techniques are recorded in current literature. This does not necessarily mean that the algorithm is void of deficiencies, but rather points to the lack of interrogation of these swarm intelligence techniques. Further, in chapter 2, an analysis of the application of the relevant techniques to the control of the DFIG was analyzed. The results showed that certain algorithms such as PSO and GWO, were applied somewhat thoroughly to the control of the DFIG. However, algorithms such as SSA and SFLA are yet to be applied for such purposes. Lastly, three well-known swarm intelligence techniques, Particle Swarm Optimization, Artificial Bee Colony, and Whale Optimization Algorithm, were applied to three benchmark functions at three dimension magnitudes. The results of the tests correlated with theory and indicated the superiority of Whale Optimization Algorithm over the others, in terms of exploration, exploitation, and convergence rate.

Chapter 3 provides an insight into the research methodology that was utilized in the subsequent chapters of this research. The swarm intelligence techniques considered were Particle Swarm Optimization, Bat Algorithm, Gorilla Troops Optimizer, African Vulture Optimization Algorithm, and Whale Optimization Algorithm. The relevant equations required to execute each of these algorithms were provided, along with a flowchart, depicting the method of implementation. It was observed that PSO is relatively easy to implement, but this is at the cost of a poor exploration and exploitation capability. On the contrary, the GTO is a complex swarm technique, which undergoes a definite dual position change in every iteration. This will increase the random-access memory require for algorithm execution, as well as the time taken to execute the algorithm. However, in doing so, the algorithm will exhibit a higher probability of producing a more precise solution. Lastly in chapter 3, a modified Whale Optimization Algorithm for exploitation capability and stability enhancement is presented. The proposed WOA introduces various modifications to the position update equations of the WOA, as well as a change to the general

structure of the algorithm. The aim of such is the mitigate the two common demerits of the WOA: low accuracy and slow convergence, as well as to prevent possible local optima entrapment at higher dimension optimization problems. The proposed technique utilizes fragments of other conventional and modified techniques which are presented in literature. Additionally, a change in the algorithm structured is presented. The proposed algorithm was applied to the CEC2019 benchmark functions and compared to various conventional and modified techniques. The results show that the proposed algorithm produced the best result in 7 of the 10 functions. Further, the EWOA generated the best overall ranking. In addition, the reliability of the proposed method can be validated via observation that the poorest ranking of the proposed algorithm was third, lower than any other compared algorithm. When applied to the optimal design of a pressure vessel, the proposed algorithm yielded significantly superior results to the other tested techniques. However, investigations revealed that the proposed algorithm required a significant number of whales to prove superior. Further, based on the structure of the algorithm outlined, there could exist instances whereby the whales will undergo a dual position change within one iteration. This may put a strain on the RAM of the PC being used, and as a result may not be able to successfully be executed on PCs with poor random-access memory. Further, these two aspects contribute the time taken to execute the algorithm, which is higher than that of the other compared algorithms. However, this is somewhat compensated for by the requirement and subsequent use of a significant lower number of iterations.

In chapter 4, a heuristic approach to optimal crowbar setting and low voltage ride through is presented. The aim of this section is the investigate the effect of applying swarm intelligence to the optimization of crowbar protection magnitudes, as well as the design of the demagnetizing current injection method and the Linear Quadratic Regulator. For this purpose, two swarm intelligence techniques were considered. These were the well-known Particle Swarm Optimization, and the recently developed African Vultures Optimization Algorithm. The test system was subject to symmetrical voltage dips of 70%, 80%, and 90%. The results of the experiments conducted concluded that for moderately severe symmetrical voltage dips, PSO is the preferred optimization technique. However, for voltage dips below 80%, the AVOA exhibits far more superior results. In particular, it was noted that for the 90% dip, AVOA exhibited a superior DC voltage settling time by a massive 101.8%. Considering the Low Voltage Ride Through experiments, it was observed that the Linear Quadratic Regulator, coupled with a strong heuristic technique, exhibited unanimous superiority in terms of rotor current and DC voltage transient peaks. There also existed dominance with regards to the rotor direct axis steady-state ripple, which gave an indication of the stator reactive power performance of the control system. However, the Linear Quadratic Regulator produced inferior steady-state error values. This was expected, as the LQR is not well-suited for tracking problems. A robustness analysis of the designed LQR was also conducted, whereby the controller was subject to a 3% perturbation in grid frequency. Once again, despite the disturbance, the LQR exhibited superior transient responses to both the PI control method and demagnetizing current injection method. However, despite the LQR providing fault ride through capabilities (whereas the crowbar method does not), it was observed that the crowbar method produced superior DC voltage transient settling times.

Chapter five presented the application of swarm intelligence to the control of the DFIG when subject to asymmetrical grid voltages. The control method implemented was the well-established dual vector control method. The control target attempting to be achieved was constant rotor current. For optimization purposes, three

swarm intelligence techniques were utilized. There were across the evolutionary timeline of swarm intelligence. The techniques considered were the well-known Particle Swarm Optimization, Bat Algorithm, and the recently developed Gorilla Troops Optimization. The test system was subject to single-phase voltage dips of 5%, and 10%, as well as a dual-phase voltage dip of 7.5%. The results presented were comprehensive, and included the rotor and stator voltages and currents, as well as the rotor direct and quadrature axis positive and negative sequence currents. Results were analyzed in terms of steady-state error, overshoot, and steady-state ripple. For the single-phase voltage dips, it was observed that PSO produced the overall best steady-state error and overshoot of the rotor direct axis current. For the same experiments, the GTO exhibited exceptional rotor direct and quadrature axis current ripples. However, despite these adequate results, neither of the three algorithms failed to achieve an acceptable rotor quadrature axis current steady-state error. This also holds true for the dual-phase voltage dip and can be attributed to the algorithms inability to simultaneously optimize eight controller parameters.

Chapter 6 aimed to address to demerits of the dual vector control technique, as presented in chapter 5. For this purpose, the optimal design of the Proportional-Resonant controller was suggested. This was based on the premise that the PR controller can introduce infinite gain at resonant frequency, and hence achieve a zero steady-state error. To ensure equality, for optimization purposes, the same three swarm intelligence techniques applied in chapter 5 were utilized. The test system was subject to single-phase voltage dips of 5%, and 10%, as well as a dual-phase voltage dip of 7.5%. The results presented were comprehensive, and included the rotor and stator voltages and currents, as well as the rotor direct and quadrature axis currents. Results were analyzed in terms of steady-state error, overshoot, and steady-state ripple. For the 5% voltage dip, it is observed that GTO produced the best result on four out of the six occasions. BA proved superior to GTO with respect to the rotor direct current overshoot and rotor quadrature current steady-state error. However, this superiority is minute and negligible as compared to the large steady-state ripple produced by BA as compared to GTO. This holds true for both the direct and quadrature rotor current. For the 10% voltage dip, it is observed that GTO once again produced the best result on four out of the six occasions. PSO proved superior to GTO with respect to the rotor direct current overshoot and rotor quadrature current steady-state error. However, this superiority is also minute and negligible as compared to the large steady-state ripple produced by PSO as compared to GTO. This holds true for both the direct and quadrature rotor current. For the 7.5% voltage dip, the GTO exhibited superiority on five out of six occasions, failing only minutely short to PSO with regards to rotor quadrature current overshoot. Furthermore, it is observed that for the single-phase voltage dips, GTO is the only algorithm to not be placed last for any of the six measurements. This points to a superior stability of GTO. In addition to this, a robustness analysis of the PR controller for the 10% voltage dip was carried out. The controller was subject to a disturbance of a 3% grid frequency perturbation. The results indicated that the controller designed via the use of GTO produced the best results in all six occasions. This indicates the robust nature of the controller, when coupled with a strong heuristic technique.

7.2. Scope of future work

Upon completion of this research work, it was observed that whilst swarm intelligence has been applied to the control of the DFIG, this aspect has not been thoroughly investigated. This holds particularly true for control of the DFIG subject to grid voltage anomalies. Wind farms are generally placed in areas where the national grid is weak, making voltage unbalances a common occurrence. This research has shown that optimization of established techniques via the use of swarm-based Metaheuristic Optimization techniques offer a strong probability for optimal performance of such control systems. This is particularly true for recently proposed swarm intelligence techniques. Upon completion of this research, the following are the future scope of work to be considered:

- An investigation into the demerits of BFO and SFO and measures to overcome possible demerits.
- The application of the SFLA, SSA and SFO to the control of the DFIG.
- The application of the modified swarm-based MOT, which are discussed in this thesis, to the control of the DFIG.
- An investigation into other swarm-based MOT. This is in terms of structure, mathematical modelling, shortcomings, advancements, and application to the control of the DFIG.
- An investigation into physics-based algorithms, evolution-based algorithms, and human related algorithm. This is in terms of structure, mathematical modelling, shortcomings, advancements, and application to the control of the DFIG.
- The combining of MOT and thereby creating hybrid algorithms to be applied to the DFIG. This would be for the intention of combining the merits of two algorithms and thereby eliminating the demerits of such algorithms.
- Investigation of the effect of swarm intelligence on harmonic elimination in the DFIG.
- Application of swarm intelligence to other types of controllers, such as sliding mode control and fuzzy logic control, for control of the DFIG as well as other application.

REFERENCES

[1] A. N. Legesse and A. K. C. R. P. Saha, "Characterisation of wind speed series and power in Durban," *Journal of Energy in Southern Africa,* vol. 28, no. 3, pp. 66-78, 2017.

[2] A. N. Legesse and A. K. C. R. P. Saha, "Damping Local Oscillations of a Direct-drive PMSG Wind Turbine," *International Journal of Engineering and Technology,* vol. 9, no. 1, pp. 158-168, 2017.

[3] V. Yaramasu and B. Wu, Model Predictive Control of Wind Energy Conversion Systems, New Jersey: Wiley, 2017.

[4] "Statistics – World Wind Energy Association," Wwindea.org, [Online]. Available: https://wwindea.org/information-2/information/..

[5] "Global Wind Power Statistics – Wind Energy International Platform," Library.wwindea.org, [Online]. Available: https://library.wwindea.org/global-statistics/.

[6] "World's top 10 countries in wind energy capacity," ET EnergyWorld, [Online]. Available: https://energy.economictimes.indiatimes.com/news/renewable/worlds-top-10-countries-in-wind-energy-capacity/68465090..

[7] "Wind energy penetration: country comparison 2019," Statista, [Online]. Available: https://www.statista.com/statistics/217804/wind-energy-penetration-by-country/.

[8] "China's Electricity Consumption Increased by 4.5% in 2019," Energy Central, [Online]. Available: https://energycentral.com/news/chinas-electricity-consumption-increased-45-2019#:~:text=Overall%2C%20China's%20power%20generation%20rose,.

[9] A. N. Legesse, A. K. Saha and R. P. Carpanen, "Generating Wind Speed Time Series for Time Domain Simulation of Wind Turbines," in *5th Southern African Universities Power Engineering Conference*, Stellenbosch, 2017.

[10] J. Pollefliet, "Applications of Power Electronics," in *Power Electronics: Switches and Converters*, Elsevier, 2017, pp. 15.1-15.44.

[11] H. Liu, H. C. L. Ma, G. Li and T. Z. F. Liu, "H. Liu, H. Ma, L. Chen, G. Li and T. Z. F. Liu, Modeling and control of DFIG based wind turbine/storage system in islanded operation," in *International Conference on Sustainable Power Generation and Supply*, Nanjing, 2009.

[12] B. Yang, Y. Tao, H. Shu, X. Zang, K. Qu and L. Jiang, "Democratic joint operations algorithm for optimal power extraction of PMSG based wind energy conversion system," *Energy Conversion and Management,* vol. 159, 2018.

[13] Y. B, Y. Tao, H. Shu, Y. Han, P. Cao and L. Jiang, "Adaptive fractional-order PID control of PMSG-based wind energy conversion system for MPPT using linear observers," *International Transactions on Electrical Energy Systems,* vol. 29, no. 9, 2018.

[14] M. Bouderbala, B. Bossoufi, H. A. Aroussi, M. Lagrioui, M. Taoussi, Y. Ihedrane and M. E. Ghamrasni, "Modeling and Power Controls of Wind Energy Conversion Systems Based on Doubly Fed Induction Generator," in *6th International Renewable and Sustainable Energy Conference*, Rabat, 2018.

[15] Y. Ihedrane and C. El Bekkali, “Direct and indirect field oriented control of DFIG-generators for wind turbines variable-speed,” in *14th International Multi-Conference on Systems, Signals & Devices (SSD)*, Marrakech, 2017.

[16] “PID Tuning via Classical Methods,” [Online]. Available: https://eng.libretexts.org/@go/page/22413.

[17] P. M. Meshram and R. G. Kanojiya, “Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor,” in *IEEE-International Conference On Advances In Engineering, Science And Management*, Nagapattinam, 2012.

[18] J. Hao and G. Zhang, “Data-Driven Tracking Control for a Class of Unknown Nonlinear Time-Varying Systems Using Improved PID Neural Network and Cohen-Coon Approach,” in *IEEE 10th Data Driven Control and Learning Systems Conference*, Suzhou, 2021.

[19] C. M. Martínez and C. Dongpu, “iHorizon-Enabled Energy Management for Electrified Vehicles, Butterworth-Heinemann,,” *Integrated energy management for electrified vehicles,* pp. 15-75, 2019.

[20] M. N. Aydin and R. Coban, “Sliding mode control design and experimental application to an electromechanical plant,” in *57th International Scientific Conference on Power and Electrical Engineering of Riga Technical Universit*, Riga, 2016.

[21] V. L. H. Utkin, “CHATTERING PROBLEM IN SLIDING MODE CONTROL SYSTEMS,” *IFAC Proceedings Volumes,* vol. 39, no. 5, 2006.

[22] M. D. O’Toole, K. Bouazza-Marouf and D. Kerr, “Chatter Suppression in Sliding Mode Control: Strategies and Tuning Methods,” in *ROMANSY 18 Robot Design, Dynamics and Control*, 2010.

[23] A. Wiama and A. Haddi, “Direct torque control-based power factor control of a DFIG,” *Special Issue on Emerging and Renewable Energy: Generation and Automation,* pp. 296-305, 2019.

[24] K. Reddy and A. K. Saha, “A comparative study on direct torque control algorithms applied to a doubly fed induction generator,” in *Southern African Universities Power Engineering Conference*, Potchefstroom, 2021.

[25] S. Walczak and N. Cerpa, “Artificial Neural Networks,” in *Encyclopedia of Physical Science and Technology*, Academic Press, 2001, pp. 631-645.

[26] S. Walczak and N. Cerpa, “Artificial Neural Networks,” in *Encyclopedia of Physical Science and Technology (Third Edition)*, 2003.

[27] M. M. Mijwil, “What is a Neural Network,” May 2018. [Online]. Available: https://www.linkedin.com/pulse/what-neural-network-maad-m-mijwel-/.

[28] J. V. Tu, “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes,” *Journal of Clinical Epidemiology,* vol. 49, no. 11, pp. 1225-1231, 1996.

[29] Y. R. a. B. Mehta, Industrial Process Automation Systems, Butterworth-Heinemann, Elsevier, 2015, pp. 547-557.

[30] K. Reddy and A. K. Saha, “Model predictive control of a doubly fed induction generator,” in *IEEE PES/IAS PowerAfrica Conference*, Nairobi, 2021.

[31] M. E. Hossain, “Improvement of transient stability of DFIG based wind generator by using of resistive solid state fault current limiter,” *Ain Shams Engineering Journal,* vol. 9, no. 4, pp. 2557-2570, 2018.

[32] J. McCalley, “Doubly-fed electric machines - steady state analysis,” Home.eng.iastate.edu, [Online]. Available: http://home.eng.iastate.edu/~jdm/wind/DFIG-SteadyState.ppt.

[33] T. Tleis, “Modelling of ac rotating machines,” in *Power Systems Modelling and Fault Analysis: Theory and Practice*, Elsevier, pp. 371-468.

[34] K. Bedouda, Ali-rachedic, T. Bahid and R. Lakelb, “Robust Control of Doubly Fed Induction Generator for Wind,” *Energy Procedia,* vol. 24, pp. 886-899, 2015.

[35] G. Byeon, I. K. Park and G. J. Jang, “Modeling and Control of a Doubly-Fed Induction Generator (DFIG),” *Journal of Electrical Engineering & Technology,* vol. 5, no. 1, pp. 61-69, 2010.

[36] L. N. Ntwasa, A. K. Saha and N. Ijumba, “Simulation Study of Induction Generator-based Wind Turbine,” in *International Conference on Power System Technology*, Hangzou, 2016.

[37] C. Soumya and R. rao, “Design of a Harmonic Filter for a Grid Connected,” *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, no. 5, 2020.

[38] V. Q. B. Ngo, P. Rodriguez-Ayerb, S. Olaru and I. Niculescu, “Model Predictive Direct Power Control of Doubly Fed Induction Generator with dead time compensation,” in *International Federation of Automatic Control,*, 2017.

[39] K. Reddy, “Model Predictive Control of a Doubly Fed Induction Generator,” December 2020. [Online]. Available: https://researchspace.ukzn.ac.za/handle/10413/19906.

[40] K. Reddy, “Model Predictive Control of a Doubly Fed Induction Generator,” November 2020. [Online]. Available: https://researchspace.ukzn.ac.za/xmlui/bitstream/handle/10413/19906/Reddy_Kumeshan_2020.pdf?sequence=1&isAllowed=y.

[41] A. Wiama and A. Haddi, “Direct torque control-based power factor control of a DFIG,” *Special Issue on Emerging and Renewable Energy: Generation and Automation,* pp. 296-305, 2019.

[42] R. Gupta and G. Dyanamina, “Matlab Simulation of DTC-SVM of Doubly Fed Induction Generator for Wind Energy System,” in *Innovations in Power and Advanced Computing Technologies (i-PACT*, Vellore, 2019.

[43] M. Valikhani and C. Sourkounis, “A comnparative study on rotor flux position and stator flux position based direct power control method in a DFIG wind turbine system,” in *International Conference on Ecological Vehicles and Renewable Energie*, Monte Carlo, 2014.

[44] K. E. Adetunji, I. W. Hofsajer, A. M. Abu-Mahfouz and L. Cheng, “A Review of Metaheuristic Techniques for Optimal Integration of Electrical Units in Distribution Networks,” *IEEE Access,* vol. 9, pp. 5046-5068, 2020.

[45] A. G. Hussein, M. Amin, M. Wang, G. Liang, A. Alsanad and A. Gumaei, “Crow Search Algorithm: Theory, Recent Advances, and Applications,” *IEEE Access,* vol. 8, pp. 173548 - 173565, 2020.

[46] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo and W. Xie, “An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems,” *IEEE Access,* vol. 8, pp. 173548 - 173565, 2020.

[47] S. Soued, H. S. Ramadan and M. Becherif, “Dynamic Behavior Analysis for Optimally Tuned On-Grid DFIG Systems,” *Energy Procedia,* vol. 162, pp. 339-348, 2019.

[48] Y. Bekakra and D. B. Attous, “Optimal tuning of PI controller using PSO optimization for indirect power,” *International Journal of Systems Assurance Engineering and Management,* vol. 5, no. 3, pp. 219-229, 2013.

[49] H. Bakir, A. Merabet, R. K. Dhar and A. A. Kulaksiz, “Bacteria foraging optimisation algorithm based optimal control for doubly-fed induction generator wind energy system,” *IET Renewable Power Generation,* vol. 14, no. 11, pp. 1850-1859, 2020.

[50] B. Yang, J. Wang, L. Yu, H. Shu, T. Yu, X. Zhang, W. Yao and L. Sun, “A critical survey on proton exchange membrane fuel cell parameter estimation using meta-heuristic algorithms,” *Journal of Cleaner Production,* vol. 265, 2020.

[51] B. Yang, J. Wang, X. Zhang, T. Yu, W. Yao, H. Shu, F. Zeng and L. Sun, “Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification,” *Energy Conversion and Management,* vol. 208, 2020.

[52] B. Yang, T. Zhu, J. Wang, H. Shu, T. Yu, X. Zhang, W. Yao and L. Sun, “Comprehensive overview of maximum power point tracking algorithms of PV systems under partial shading condition,” *Journal of Cleaner Production,* vol. 268, 2020.

[53] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. Hernandez and R. G. Harley, “Particle Swarm Optimization: Basic Concepts,” *IEEE Transactions on Evolutionary Computation,* vol. 12, no. 2, pp. 171-195, 2008.

[54] D. W. Boeringer and D. H. Werner, “Particle swarm optimization versus genetic algorithms for phased array synthesis,” *IEEE Transactions on Antennas and Propagation,* vol. 52, no. 3, pp. 771-779, 2004.

[55] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, 1995.

[56] J. Kennedy and R. Eberhart, “A New Optimizer Using Particle Swarm Theory,” in *Sixth International Symposium on Micro Machine and Human Science*, 1995.

[57] H. Lihong, Y. Nan, W. Jianhua, S. Ying, D. Kingking and X. Ying, “Application of modified PSO in the optimization of reactive power,” in *Chinese Control and Decision Conference*, Guilin, 2009.

[58] Z. Abdmouleh, A. Gastli, L. Ben-Brahim and M. Haouari, “Review of optimization techniques applied for the integration of distributed generation from renewable energy sources,” *Renewable Energy,* vol. 113, pp. 266-280, 2017.

[59] X. Xia, Y. Tang, B. Wei and L. Gui, “Dynamic Multi-Swarm Particle Swarm Optimization Based on Elite Learning,” *IEEE Access,* vol. 7, pp. 184849 - 184865.

[60] R. Laina, F. E. Lamzouri, E. Boufounas, A. E. Amrani and I. Boumhidi, “Intelligent control of a DFIG wind turbine using a PSO evolutionary,” *The First International Conference On Intelligent Computing in Data Sciences,* vol. 127, pp. 471-480, 2018.

[61] F. Wu and X. P. Zhang, “Small signal stability analysis and optimal control of a wind turbine with doubly fed induction generator,” *IET Generation, Transmission and Distribution,* vol. 1, no. 5, pp. 751-760, 2007.

[62] Y. Tang, P. Ju, H. He, C. Qin and F. Wu, “Optimized Control of DFIG-Based Wind Generation Using Sensitivity Analysis and Particle Swarm Optimization,” *IEEE Transactions on Smart Grid,* vol. 4, no. 1, pp. 509-520, 2013.

[63] O. P. Bharti, R. K. Saket and S. K. Nagar, “Controller design for doubly fed induction generator using particle swarm optimization technique,” *Renewable Energy,* vol. 114, no. B, pp. 1394-1406, 2017.

[64] K. Lu, C. Hong, X. Tan and F. Cheng, “Novel Intelligent Control Technology for Enhanced Stability Performance of an Ocean Wave Energy Conversion System,” *MDPI-Energies,* vol. 14, no. 7, 2021.

[65] Y. Li, B. Xiong, Y. Su, J. Tang and Z. Leng, “Particle Swarm Optimization-Based Power and Temperature Control Scheme for Grid-Connected DFIG-Based Dish-Stirling Solar-Thermal System,” *MDPI-Energies,* vol. 12, no. 7, 2019.

[66] A. S. S. Oleynik, “Feature Selection Based on Bacteria Foraging Intelligence,” in *CADSM* , Polyana-Svalyava, 2009.

[67] N. J. Jhankal and D. Adhyaru, “Bacterial foraging optimization algorithm: A derivative free technique,” in *Nirma University International Conference on Engineering*, Ahmedabad, 2011.

[68] S. Das, A. Biswas, S. Dasgupta and A. Abraham, “Bacterial Foraging Optimization Algorithm: Theoretical,” [Online]. Available: http://www.softcomputing.net/bfoa-chapter.pdf.

[69] A. Oleynik and S. Subbotin, “Feature selection based on bacteria foraging intelligence,” in *10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics*, Lviv, 2009.

[70] J. Wang, C. Liu and M. Zhou, “Improved Bacterial Foraging Algorithm for Cell Formation and Product Scheduling Considering Learning and Forgetting Factors in Cellular Manufacturing Systems,” *IEEE Systems Journal,* vol. 14, no. 2, pp. 3047 - 3056, 2020.

[71] Y. Mishra, S. Mishra, F. Li, Z. Y. Dong and R. C. Bansal, “Small-Signal Stability Analysis of a DFIG-Based Wind Power System Under Different Modes of Operation,” *IEEE Transactions on Energy Conversion,* vol. 24, no. 4, pp. 972-982, 2009.

[72] R. Rajakumar, K. Sekaran, C. Hsu and S. Kadry, “Accelerated grey wolf optimization for global optimization problems,” *Technological Forecasting and Social Change,* vol. 169, 2021.

[73] W. GU and B. Zhou, “Improved grey wolf optimization based on the Quantum-behaved mechanism,” in *IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, 2019.

[74] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo and W. Xie, “An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems,” *IEEE Access,* vol. 8, pp. 69861 - 69893, 2020.

[75] S. N. Syed and S. T. Kalyani, “Performance improvement of Doubly Fed Induction Generator using Grey Wolf Optimization,” in *International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, 2021.

[76] B. Yang, X. Zhang, T. Yu, H. Shu and Z. Fang, “Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine,” *Energy Conversion and Management,* vol. 133, 2017.

[77] M. B. Kokare and S. V. Tade, “Application of Artificial Bee Colony Method for Unit Commitment,” in *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, 2018.

[78] V. Neagoe and C. Neghina, “An Artificial Bee Colony Approach for Classification of Remote Sensing Imagery,” in *10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Iasi, 2018.

[79] D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, “A comprehensive survey: Artificial bee colony (ABC) algorithm and applications,” *Springer,* vol. 42, pp. 21-57, 2012.

[80] L. Sun, T. Chen and Q. Zhang, “An Artificial Bee Colony Algorithm with Random Location Updating,” *Data Science and AI-based Optimization in Scientific Programming,* 2018.

[81] L. Cheng, M. Yu, J. Yang and Y. Wang, “An Improved Artificial Bee Colony Algorithm based on Beetle Antennae Search,” in *Chinese Control Conference*, Guangzhou, 2019.

[82] H. Gao, Y. Shi, C. Pun and S. Kwong, “An Improved Artificial Bee Colony Algorithm With its Application,” *IEEE Transactions on Industrial Informatics,* vol. 15, no. 4, pp. 1853 - 1865, 2018.

[83] M. Chandra and R. Niyogi, “Web Service Selection Using Modified Artificial Bee Colony Algorithm,” *IEEE Access,* vol. 7, pp. 88673 - 88684, 2019.

[84] S. Zhang and S. Liu, “A Discrete Improved Artificial Bee Colony Algorithm for 0–1 Knapsack Problem,” *IEE Access,* vol. 7, pp. 104982 - 104991, 2019.

[85] S. Soued, H. S. Ramadan and M. Becherif, “Dynamic Behavior Analysis for Optimally Tuned On-Grid DFIG Systems,” *Energy Procedia,* vol. 162, pp. 339-348, 2019.

[86] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016.

[87] W. Gu, “An improved whale optimization algorithm with cultural mechanism for high-dimensional global optimization problems,” in *IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, Chongqing, 2020.

[88] Q. Zhai, X. Xia, S. Feng and M. Huang, “Optimization Design of LQR Controller Based on Improved Whale Optimization Algorithm,” in *3rd International Conference on Information and Computer Technologies (ICICT)*, San Jose, 2020.

[89] J. Nasiri and F. Khiyabani, “A whale optimization algorithm (WOA) approach for clustering,” *Cogent Mathematics and Statistics,* vol. 5, 2018.

[90] Z. Yan, J. Zhang, J. Zeng and J. Tang, “Nature-inspired approach: An enhanced whale optimization algorithm for global optimization,” *Mathematics and Computers in Simulation,* vol. 185, pp. 17-46, 2021.

[91] J. Zhang and J. Wang, “Improved Whale Optimization Algorithm Based on Nonlinear Adaptive Weight and Golden Sine Operator,” *IEEE Access,* vol. 8, pp. 77013 - 77048, 2020.

[92] W. Jianhao, W. Long, C. Lijie and G. Tian, “Enhanced Whale Optimization Algorithm for Large-Scale Global Optimization Problems,” in *International Conference on Computer Communication and Artificial Intelligence*, Guangzhou, 2021.

[93] X. Chen, “Research on New Adaptive Whale Algorithm,” *IEEE Access,* vol. 8, pp. 90165 - 90201, 2020.

[94] X. Han, Q. Xu, L. Yue, Y. Dong, G. Xie and X. Xu, “An Improved Crow Search Algorithm Based on Spiral Search Mechanism for Solving Numerical and Engineering Optimization Problems,” *IEEE Access,* vol. 8, pp. 92363 - 92382, 2020.

[95] O. H. Zahran, T. A. Boghdady and M. M. Sayed, “Improving the Controller Performance for a Grid Connected Wind Farm,” in *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion* , Dubrovnik, 2018.

[96] J. Senthilnath, S. Kulkarni, J. A. Benediktsson and X. S. Yang, “A Novel Approach for Multispectral Satellite Image Classification Based on the Bat Algorithm,” *IEEE Geoscience and Remote Sensing Letters* , vol. 13, no. 4, pp. 599 - 603, 2016.

[97] L. F. Zhu, J. S. Wang, S. S. Guo, M. W. Guo and W. Xie, “Data Clustering Method Based on Improved Bat Algorithm With Six Convergence Factors and Local Search Operators,” *IEEE Access,* vol. 8, pp. 80536 - 80560, 2020.

[98] “Advantages and disadvantages of Bat Algorithm,” [Online]. Available: https://www.bartleby.com/essay/Advantages-And-Disadvantages-Of-Bat-Algorithm-FJ7QTTVY3G.

[99] X. Zhou, F. Gao, X. Fang and Z. Lan, “Improved Bat Algorithm for UAV Path Planning in Three-Dimensional Space,” *IEEE Access,* vol. 9, pp. 20100 - 20116, 2021.

[100] Z. Shu, Y. Chen, C. Deng, F. Zheng and H. Zhong, “Pareto Optimal Allocation of Flexible Fault Current Limiter Based on Multi-Objective Improved Bat Algorithm,” *IEEE Access,* vol. 9, pp. 12762 - 12778, 2021.

[101] L. Chen, X. Zhang, H. Chen, G. Li, J. Yang and X. Tian, “Pareto optimal allocation of resistive-type fault current limiters in active distribution networks with inverter-interfaced and synchronous distributed generators,” *Energy Sci. Eng. ,* vol. 7, no. 6, pp. 2554-2571, 2019.

[102] A. Mahmoudian, M. R. Islam, A. Z. Kouzani and M. A. P. Mahmud, “Optimal allocation of fault current limiter in distribution network with NSGA-II algorithm,” *IEEE International Conference on Applied Superconductors and Electromagnetic devices,* pp. 1-2, 2020.

[103] O. B. Alzain, X. Liu and A. Ali, “optimization of Sliding Mode Control based on BAT-Algorithm for the DFIG-WT,” in *International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, 2021.

[104] X. Zhang, K. Zhai, L. Wang and Y. Niu, “An Improved Squirrel Search Algorithm With Reproductive Behavior,” *IEEE Access,* vol. 8, pp. 101118 - 101132, 2020.

[105] H. Hu, L. Zhang, Y. Bai, P. Wang and X. Tan, “A Hybrid Algorithm Based on Squirrel Search Algorithm and Invasive Weed Optimization for Optimization,” *IEEE Access,* vol. 7, pp. 105652 - 105668, 2019.

[106] P. Wang, Y. Kong, X. He, M. Zhang and X. Tan, “An Improved Squirrel Search Algorithm for Maximum Likelihood DOA Estimation and Application for MEMS Vector Hydrophone Array,” *IEEE Access ,* vol. 7, pp. 118343 - 118358, 2019.

[107] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowledge Based Systems,* vol. 89, pp. 228-249, 2015.

[108] E. E. Elattar and S. K. ElSayed, “Optimal Location and Sizing of Distributed Generators Based on Renewable Energy Sources Using Modified Moth Flame Optimization Technique,” *IEEE Access,* vol. 8, pp. 109625 - 109638, 2020.

[109] T. Nguyen, H. Wang, T. Dao, J. Pan, T. Ngo and J. Yu, “A Scheme of Color Image Multithreshold Segmentation Based on Improved Moth-Flame Algorithm,” *IEEE Access,* vol. 8, pp. 174142 - 174159, 2020.

[110] L. Hongwei, L. Jianyong, C. Liang, B. Jingbo, S. Yangyang and L. Kai, “Chaos-enhanced moth-flame optimization algorithm for global optimization,” *Journal of Systems Engineering and Electronics ,* vol. 30, no. 6, pp. 1144 - 1159, 2019.

[111] L. Huang, B. Yang and X. Zhng, “Optimal power tracking of doubly fed induction generator-based wind turbine using swarm moth–flame optimizer,” *Transactions of the Institute of Measurement and Controls,* vol. 41, no. 6, p. 1489–1490, 2019.

[112] S. Shadravan, H. R. Naji and V. K. Bardsiri, “The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems☆,” *Engineering Applications of Artificial Intelligence,* vol. 80, pp. 20-34, 2019.

[113] D. A. Nugraha, K. L. Lian and Suwarno, “A Novel MPPT Method Based on Cuckoo Search Algorithm and Golden Section Search Algorithm for Partially Shaded PV System,” *Canadian Journal of Electrical and Computer Engineering ,* vol. 42, no. 3, pp. 173 - 182, 219.

[114] A. S. Joshi, O. Kulkarni, G. M. Kakandikar and V. M. Nandedkar, “Cuckoo Search Optimization- A Review,” *materials today: Proceedings,* vol. 4, no. 8, pp. 7262-7269, 2017.

[115] X. Li, S. Ma and G. Yang, “Synthesis of Difference Patterns for Monopulse Antennas by an Improved Cuckoo Search Algorithm,” *IEEE Antennas and Wireless Propagation Letters ,* vol. 16, pp. 141-144, 2016.

[116] L. Liu, X. Liu, N. Wang and P. Zou, “Modified Cuckoo Search Algorithm with Variational,” *MDPI algorithms,* 2018.

[117] M. A. Al-Gabalawy, N. S. Hosny and N. S. Hussien, "Cuckoo search algorithm based for tunning both PI and FOPID controllers for the DFIG-Wind energy conversion syste," *International Journal of Electrical and Computer Engineering,* vol. 10, no. 6, p. 6319~6329, 2020.

[118] K. Jagatheesan, B. Anand, S. Samantha, N. Dey, A. S. Ashour and V. E. Balas, "Design of a Proportional Integral Derivative controllerfor an automatic generation control of multi-area power thermal system using Firefly Algorithm," *IEEE/CAA journal of automatics sinica,* vol. 6, no. 2, pp. 503-515, 2019.

[119] M. Alb, P. Alotto, C. Magele, W. Renhart, K. Preis and B. Trapp, "Firefly Algorithm for Finding Optimal Shapes of Electromagnetic Devices," *IEEE Transactions on Magnetics ,* vol. 52, no. 3, 2015.

[120] S. Liaquat, M. S. Fakhar, S. A. R. Kashif, A. Rasool and O. Saleem, "Application of Dynamically Search Space Squeezed Modified Firefly Algorithm to a Novel Short Term Economic Dispatch of Multi-Generation Systems," *IEEE Access,* vol. 9, pp. 1918 - 1939, 2020.

[121] S. K. Sarangi, R. Panda and A. Saranga, "Design of adaptive IIR filter with modified firefly algorithm for parameter estimation," in *Technologies for Smart-City Energy Security and Power (ICSESP)*, Bhubaneswar, 2018.

[122] S. K. Sarangi, R. Panda, S. Priyadarshni and A. Sarangi, "A new modified firefly algorithm for function optimization," in *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016.

[123] T. Surinkaew and I. Ngamroo, "Robust power oscillation damper design for DFIG-based wind turbine based on specified structure mixed H2/H∞ control," *Renewable Energy,* vol. 66, pp. 15-24, 2014.

[124] P. Venkatesh, S. U. Rao and T. Suneel, "Frequency Regulation in Multi Area Power System Optimized By Firefly Swarm Hybridization Algorithm," *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, no. 6, pp. 2214-2218, 2020.

[125] K. K. Bhattacharjee and S. Sarmah, "Shuffled frog leaping algorithm and its application to 0/1 knapsack problem," *Applied Soft Computing,* vol. 19, pp. 252-263, 2014.

[126] J. Ni, X. Yin, J. chen and X. Li, "An improved shuffled frog leaping algorithm for robot path planning," in *10th International Conference on Natural Computation (ICNC)*, Xiamen, 2014.

[127] B. Hu, Y. Dai, Y. Su, P. Moore, X. Zhang, C. Mao, J. Chen and L. Xu, "Feature Selection for Optimized High-Dimensional Biomedical Data Using an Improved Shuffled Frog Leaping Algorithm," *IEEE/ACM Transactions on Computational Biology and Bioinformatics ,* vol. 15, no. 6, pp. 1765 - 1773, 2016.

[128] J. Ni, X. Yin, J. Chen and X. Li, "An improved shuffled frog leaping algorithm for robot path planning," in *International Conference on Natural Computation (ICNC)*, Xiamen, 2014.

[129] J. Zhou, E. Dutkiewicz, R. P. Liu, X. Huang, G. Fang and Y. Liu, "A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems," *IEEE Transactions on Broadcasting ,* vol. 61, no. 4, pp. 698 - 709, 2015.

[130] T. Spoljaric, C. Lusetic and V. Simovic, "Optimization of PID controller in AVR system by using ant lion optimizer algorithm," in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, 2018.

[131] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software,* vol. 83, pp. 80-98, 2015.

[132] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo and W. Xie, "Improved Ant Lion Optimizer Based on Spiral Complex Path Searching Patterns," *IEEE Access,* vol. 8, pp. 22094 - 22126, 2020.

[133] L. Zheng, Z. Wang, Z. Zhao, J. Wang and W. Du, "Research of Bearing Fault Diagnosis Method Based on Multi-Layer Extreme Learning Machine Optimized by Novel Ant Lion Algorithm," *IEEE Access,* vol. 7, pp. 89845 - 89856, 2019.

[134] M. V. U. Maheswari, P. V. R. Rao and S. V. J. Kumar, "Active and Reactive Power Control of DFIG Wind Power System by Heuristic Controllers," in *Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, 2019.

[135] V. R. Kulkarni and V. Desai, "ABC and PSO: A Comparative Analysis," in *IEEE International Conference on Computational Intelligence and Computing Research*, Chennai, 2016.

[136] M. S. R. Tanveer, M. J. Islam and A. MAH, "A Comparative Study on Prominent Swarm Intelligence Methods for," *Global Journal of Technology and Optimization,* vol. 7, no. 3, 2016.

[137] M. Eghbal, T. K. Saha and K. N. Hasan, "Transmission expansion planning by meta-heuristic techniques: A comparison of Shuffled Frog Leaping Algorithm, PSO and GA," in *IEEE Power and Energy Society General Meeting*, Detroit, 2011.

[138] Y. A. Alsariera, H. S. Alamri, A. M. Nasser, M. A. Majid and K. Z. Zamli, "Comparative performance analysis of bat algorithm and bacterial foraging optimization algorithm using standard benchmark functions," in *Malaysian Software Engineering Conference*, Langkawi, 2014.

[139] N. Pillay, "A Parsicle Swarm Optimization approach for tuning of SISO PID control loops," 2008. [Online]. Available: https://openscholar.dut.ac.za/bitstream/10321/488/3/Pillay_2008.pdf.

[140] R. Gochhayat, "PSO based PI controller for load frequency control of interconnected power systems," 2014. [Online]. Available: https://core.ac.uk/download/pdf/53190448.pdf.

[141] H. Fan and Y. Shi, "Study on Vmax of particle swarm optimization," in *Proc. Workshop on Particle Swarm Optimization*, Indianapolis, 2001.

[142] R. Eberhart, Y. Shi and J. Kennedy, "Swarm Intelligence," in *Morgan Kauffman*, San Mateo, 2001.

[143] E. Ozcan and C. Mohan, "Particle swarm optimization: Surfing the waves," *Proc. IEEE Congress Evol. Comput,* vol. 9, pp. 1939-1944, 1999.

[144] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Seuol, 2001.

[145] D. Boeringer and D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation,* vol. 52, no. 3, pp. 771-779, 2004.

[146] G. Chen, G. Qian, Z. Zhang and Z. Sun, "Applications of Novel Hybrid Bat Algorithm With Constrained Pareto Fuzzy Dominant Rule on Multi-Objective Optimal Power Flow Problems," *IEEE Access,* vol. 7, pp. 52060 - 52084, 2019.

[147] F. X. Rugema, G. Yan, M. S., Q. Jia, S. Zhang and C. Bananeza, "A Cauchy-Gaussian Quantum-Behaved Bat Algorithm Applied to Solve the Economic Load Dispatch Problem," *IEEE Access,* vol. 9, pp. 3207 - 3228, 2020.

[148] S. Yu, Y. Yu, M. Zhang, E. Wu and L. Wu, "The economic benefit of liner ships," *Advances in Mechanical Engineering,* vol. 9, no. 11, pp. 1-11, 2017.

[149] B. Abdollahzadeh, F. S. Gharehchopogh and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *International Journal of Intelligent Systems,* vol. 36, no. 10, pp. 5887-5958, 2021.

[150] M. Ali, K. Aboras, H. Kotb and N. Abbasy, "Design of Cascaded PI-Fractional Order PID Controller for Improving the Frequency Response of Hybrid Microgrid System Using Gorilla Troops Optimizer," *IEEE Access,* vol. 9, pp. 150715 - 150732, 2021.

[151] A. Shaheen, A. Ginidi, R. El-Sehiemy, A. Elsayed, E. Elattar and H. T. Dorrah, "Developed Gorilla Troops Technique for Optimal Power Flow Problem in Electrical Power Systems," *MDPI Mathematics,* vol. 10, 2022.

[152] I. Ahmed, A. Dahou, S. A. Chelloug, M. A. A. Al-qaness and M. A. Elaziz, "Feature Selection Model Based on Gorilla Troops Optimizer for Intrusion Detection Systems," *Journal of Sensors,* 2022.

[153] B. Abdollahzadeh, F. S. Gharehchopogh and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Computers & Industrial Engineering,* vol. 158, 2021.

[154] G. A. Ghazi, H. M. Hasanien, E. A. Al-Ammar, R. A. Turky, W. Ko, S. Park and H. Choi, "African Vulture Optimization Algorithm-Based PI Controllers for Performance Enhancement of Hybrid Renewable-Energy Systems," *MDPI Sustainability ,* vol. 14, pp. 1-26, 2022.

[155] J. Fan, Y. Li and T. Wang, "An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism," *Plos One,* 2021.

[156] R. Liu, T. Wang, J. Zhou, X. Hao, Y. Xu and J. Qiu, "Improved African Vulture Optimization Algorithm Based on Quasi-Oppositional Differential Evolution Operator," *IEEE Access,* vol. 10, pp. 95197 - 95218, 2022.

[157] R. P. Sahu, P. K. Hota and S. Panda, "Comparison of Grasshopper and Whale Optimization Algorithm for Design of FACTS Controller with Power System Stabilizer," in *5th IEEE International Conference on Parallel, Distributed and Grid Computing*, Solan, 2018.

[158] Z. Wu, K. Huang, J. Chen and C. Yang, "A Memetic Fuzzy Whale Optimization Algorithm for Data Clustering," in *IEEE Congress on Evolutionary Computation (CEC)*, Wellington, 2019.

[159] A. M. Masaad, M. A. attia and A. Y. Abdelaziz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system," *Ain Shams Engineering Journal,* 2019.

[160] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016.

[161] J. K. F. Nasiri, “A whale optimization algorithm (WOA) approach for clustering,” *Cogent Mathematics and Statistics,* vol. 5, 2018.

[162] Z. Yan, J. Zhang, J. Zeng and J. Tang, “Nature-inspired approach: An enhanced whale optimization algorithm for global optimization,” *Mathematics and Computers in Simulation,* vol. 185, pp. 17-46, 2021.

[163] Y. Li, T. Han, H. Zhao and H. Go, “An Adaptive Whale Optimization Algorithm Using Gaussian Distribution Strategies and Its Application in Heterogeneous UCAVs Task Allocation,” *IEEE Access,* vol. 7, pp. 110138 - 110158, 2019.

[164] Y. Sun, X. Wang, Y. Chen and Z. Liu, “A modified whale optimization algorithm for large-scale global optimization problems,” *Expert Systems with Applications,* vol. 114, pp. 563-577, 2018.

[165] S. Chakraborty, A. K. Saha, S. Nama and S. Debnath, “COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction,” *Computers in Biology and Medicine,* 2021.

[166] P. Du, W. Cheng, L. Liu, H. Zhang and J. Lu, “A Modified Whale Optimization Algorithm with Single-Dimensional Swimming for Global Optimization Problems,” *MDPI Symmetry,* vol. 12, no. 11, 2020.

[167] X. Ye, W. Liu, H. Li, M. Wang, C. Chi, G. Liang, H. Chen and H. Huang, “Learning and Adaptation for Optimization and Control of Complex Renewable Energy Systems,” *Hindawi Complexity,* 2021.

[168] K. Lu and Z. Ma, “A modified whale optimization algorithm for parameter estimation of software reliability growth models,” *Journal of Algorithms & Computational Technology,* vol. 15, 2021.

[169] H. Mohammed and T. Rashid, “A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design,” *Neural computing and applications,* vol. 32, p. 14701–14718 , 2020.

[170] M. S. Shabbir, A. F. Siddiqi, L. M. Yapanto and E. E. Tonkov, “Closed-Loop Supply Chain Design and Pricing in Competitive Conditions by Considering the Variable Value of Return Products Using the Whale Optimization Algorithm,” *Sustainability,* vol. 13, pp. 1-17, 2021.

[171] A. Selim, S. Kamel and J. Francisco, “Voltage Profile Improvement in Active Distribution Networks Using Hybrid WOA-SCA Optimization Algorithm,” in *Twentieth International Middle East Power Systems Conference* , Cairo, 2018.

[172] J. Wang, M. Ye, F. Xiong and Y. Qian, “Cross-Scene Hyperspectral Feature Selection via Hybrid Whale Optimization Algorithm With Simulated Annealing,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 14, pp. 2473 - 2483, 2021.

[173] X. Liang, S. Xu, Y. Liu and L. Sun, “Artificial Intelligence-Based Traffic Delivery for Mobile Networks,” Mobile Information Systems, vol. 2022, 2022.,” *Mobile Information Systems,* vol. 2022, 2022.

[174] M. Sathya and S. M. Priya, “A search space enhanced modified whale optimization algorithm for feature selection in large-scale microarray datasets,” *Indian Journal of Science and Technology,* vol. 13, no. 42, 2020.

[175] R. Sivalingam and S. D. S. S. Chinnamuthu, “A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems,” *Journal for Control, Measurement, Electronics, Computing and Communications,* vol. 58, no. 4, pp. 410-421, 2017.

[176] M. Alam, “The Application of Particle Swarm Optimization for solving Electrical Engineering Problems,” in *Focus on Swarm Intelligence Research and Applications*, Nova Science Publishers, Inc., 2017, p. Chapter 4.

[177] N. K. Kulkarni, S. Patekar, T. Bhoshar, O. Kulkarni, G. M. Kakandikar and V. M. Nandedkar, “Particle Swarm Optimization Applications to Mechanical Engineering- A Review,” *Materialstoday:Proceedings,* vol. 2, no. 4, pp. 2631-2639, 2015.

[178] M. Hajihassani, D. J. Armaghani and R. Kalatehjari, “Applications of Particle Swarm Optimization in Geotechnical Engineering: A Comprehensive Review,” *Geotechnical and Geological Engineering,* vol. 36, no. 5, 2018.

[179] D. Tian and Z. Shi, “MPSO: Modified particle swarm optimization and its applications,” *Swarm and Evolutionary Computation,* vol. 41, pp. 49-68, 2018.

[180] S. U. Khan, S. Yang, L. Wang and L. Liu, “S. U. Khan, S. Yang, L. Wang and L. Liu, "A Modified Particle Swarm Optimization Algorithm for Global Optimizations of Inverse Problems,” *IEEE Transactions on Magnetics,* vol. 52, no. 3, 2016.

[181] C. M. Rahman and T. A. Rashid, “A new evolutionary algorithm: Learner performance based behavior algorithm,” *Egyptian Informatics Journal,* 2020.

[182] J. Chen, H. Rong, Z. Zhang and R. Luo, “An Adaptive Evolutionary Whale Optimization Algorithm,” in *33rd Chinese Control and Decision Conference*, Kunming, 2021.

[183] H. Shayanfar and F. S. Gharehchopogh, “Farmland fertility: A new metaheuristic algorithm for solvingcontinuous optimization problems,” *Applied Soft Computing,* vol. 71, pp. 728-746, 2018.

[184] X. Zhao, Y. Fang, L. Liu, J. Li and M. Xu, “An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems,” *Applied Intelligence,* vol. 50, no. 10, 2020.

[185] K. E. Parsopoulos and M. N. Vrahatis, “Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems,” in *Advances in Natural Computation, First International Conference*, Changsha, 2005.

[186] S. Li, Y. Ming and Y. Zhang, “Crowbar Resistance Setting and its Influence on DFIG Low Voltage Based on Characteristics,” *International Journal of Robotics and Automation,* vol. 6, no. 1, pp. 31-38, 2017.

[187] D. Zhou and F. Blaabjerg, “Optimized Demagnetizing Control of DFIG Power Converter for Reduced Thermal Stress During Symmetrical Grid Fault,” *IEEE Transactions on Power Electronics,* vol. 33, no. 12, pp. 10326-10340, 2018.

[188] Y. Yan, M. Wang, Z. Song and C. Xia, “Proportional-Resonant Control of Doubly-Fed Induction Generator Wind Turbines for Low-Voltage Ride-Through Enhancement,” *MDPI Energies,* vol. 5, pp. 4758-4778, 2012.

[189] P. Morsali, P. Morsali and E. Ghadikola, “Analysis and Simulation of Optimal Crowbar Value Selection on Low Voltage Ride-Through Behavior of a DFIG-Based Wind Turbine,” *MDPI Proceedings,* vol. 58, no. 18, pp. 1-9, 2020.

[190] Y. Zhou, P. Bauer, J. Ferreira and J. Pierik, "Operation of Grid-Connected DFIG Under Unbalanced Grid Voltage Condition," *IEEE Transactions on Energy Conversion,* vol. 24, no. 1, pp. 240-246, 2009.

[191] S. R. Kalantarian and H. Heydari, "An analytical method for selecting optimized crowbar for DFIG with AHP algorithm," in *Power Electronics, Drive Systems and Technologies Conference*, Tehran, 2011.

[192] H. L. Zhou, G. Yang and D. Y. Li, "Short circuit current analysis of DFIG wind turbines with crowbar protection," in *International Conference on Electrical Machines and Systems*, Tokyo, 2009.

[193] M. B. C. Salles, K. Hameyer, J. R. Cardoso, A. P. Grilo and C. Rahmann, "Crowbar System in Doubly Fed Induction Wind Generators," *MDPI Energies,* vol. 3, pp. 738-753, 2010.

[194] O. Noureldeen and I. Hamdan, "A novel controllable crowbar based on fault type protection technique for DFIG wind energy conversion system using adaptive neuro-fuzzy inference system," *Protection and Control of Modern Power Systems,* vol. 3, no. 35, 2018.

[195] K. M. Kondapi and R. B. R. Prakash, "Stability Enhancement of Doubly Fed Induction Generator with Virtual Resistance for Grid Disturbances," *Indian Journal of Science and Technology,* vol. 8, no. 17, 2015.

[196] G. L. J. Abad, M. Rodriguez, L. Marroyo and G. Iwanski, Doubly Fed Induction Machine, New Jersey: Wiley, 2011.

[197] M. K. Senapati, C. Pradhan, P. K. Nayak, S. Padmanaban and T. Gjengedal, "Modified demagnetisation control strategy for low-voltage ride-through enhancement in DFIG-based wind systems," *IET Renewable Power Generation,* pp. 3487-3499, 2020.

[198] M. J. Harandi, S. G. Liasi, E. Nikravesh and M. T. Bina, "An Improved Control Strategy for DFIG Low Voltage Ride-Through Using Optimal Demagnetizing method," in *International Power Electronics, Drive Systems and Technologies Conference*, Shiraz, 2019.

[199] J. Luo, H. Zhao, S. Gao and M. Han, "A Low Voltage Ride Through Strategy of DFIG based on Explicit Model Predictive Control," *International Journal of Electrical Power & Energy Systems,* vol. 119, 2020.

[200] D. Zang, H. Xu, L. Qiao and L. Chen, "LVRT capability enhancement of DFIG based wind turbine with coordination control of dynamic voltage restorer and inductive fault current limiter," *Plos One,* vol. 14, no. 8, 2019.

[201] M. R. Hazari, M. A. Mannan, S. M. Muyeen, A. Umemura, R. Takahashi and J. Tamura, "Stability Augmentation of a Grid-Connected Wind Farm by Fuzzy-Logic-Controlled DFIG-Based Wind Turbines," *MDPI Applied Sciences,* vol. 8, no. 20, 2018.

[202] B. Qin, H. Li, X. Zhou, J. Li and W. Liu, "Low-Voltage Ride-Through Techniques in DFIG-BasedWind Turbines: A Review," *MDPI Applied Sciences,* 2020.

[203] A. Khajeh and R. Ghazi, "GA-Based Optimal LQR Controller to Improve LVRT," *Iranian Journal of Electrical & Electronic Engineering,* vol. 9, no. 3, pp. 167-176, 2013.

[204] A. L. L. F. Murari, L. L. Rodriguez, J. A. T. Altuna, A. S. Potts, L. A. L. Almeida and A. J. S. Filho, "A LQRI power control for DFIG tuned by a weighted-PSO," *Control Engineering Practice,* vol. 85, pp. 41-49, 2019.

[205] R. Bhushan, K. Chatterjee and R. Shankar, "Comparison between GA-based LQR and conventional LQR control method of DFIG wind energy system," in *International Conference on Recent Advances in Information Technology*, Dhanbad, 2016.

[206] A. El-Naggar, A. Korai and I. Erlich, "Using MVMO for Optimal Tuning of Linear Quadratic Regulators for DFIG-WT," *IFAC-PapersOnLine,* vol. 48, no. 30, pp. 479-484, 2015.

[207] J. Lopez, P. Sanchis, X. Roboam and L. Marroyo, "Dynamic Behavior of the Doubly Fed Induction Generator During Three-Phase Voltage Dips," *IEEE Transactions on Energy Conversion,* vol. 22, no. 3, pp. 709-717, 2007.

[208] K. Patel and S. P. Das, "Improved Traditional Demagnetization Control of DFIG Under Balanced Grid Faults," in *IEEE International Conference on Power Electronics, Drives and Energy Systems*, Jaipur, 2020.

[209] L. Zhou, J. Liu and S. Zhou, "Improved Demagnetization Control of a Doubly-Fed Induction Generator Under Balanced Grid Fault," *IEEE Transactions on Power Electronics,* vol. 30, no. 12, pp. 6695 - 6705, 2014.

[210] Y. Wu, W. Shu, J. Liao and W. Wu, "Dynamic Behavior of the Doubly Fed Induction Generator During Three-Phase and Single-Phase Voltage Dips," in *IEEE 2nd International Conference on Knowledge Innovation and Invention*, Seoul, 2019.

[211] T. D. V. Nguyen and G. Fujita, "Nonlinear control of DFIG under symmetrical voltage dips with demagnetizing current solution," in *IEEE International Conference on Power System Technology* , Auckland, 2012.

[212] H. El Azri, A. Essadki and T. Nasser, "LQR Controller Design for a Nonlinear, Doubly Fed Induction Generator Model," in *International Renewable and Sustainable Energy Conference* , Rabat, 2018.

[213] G. Abad, J. Lopez, M. A. Rodriguez, L. Marroyo and G. Iwanski, Doubly Fed Induction Machine-Modelling and control for wind energy generation, New Jersey: Wiley, 2011.

[214] K. Reddy, "Model Predictive Control of a Doubly Fed Induction Generator," [Online]. Available: https://researchspace.ukzn.ac.za/xmlui/handle/10413/19906.

[215] H. Wang, W. Zhang, J. Hu and Y. He, "Design and optimization of proportional resonant controller for rotor current of a wind turbine driven DFIG," in *International Conference on Electrical Machines and Systems*, Wuhan, 2008.

[216] Y. Yan, M. Wang, Z. Song and C. Xia, "Proportional-Resonant Control of Doubly-Fed Induction Generator Wind Turbines for Low-Voltage Ride-Through Enhancement," *Energies,* vol. 5, no. 11, pp. 4758-4778, 2012.

[217] H. Cha, T. Vu and J. Kim, "Design and control of Proportional-Resonant controller based Photovoltaic power conditioning system," in *IEEE Energy Conversion Congress and Exposition*, San Jose, 2009.

[218] Y. Djeriri, A. Meroufel, B. Belabbes and A. Massoum, "Three-level NPC voltage source converter based direct power control of the doubly fed induction generator at low constant switching frequency," *Revue des Energies Renouvelables,* vol. 16, no. 1, pp. 91-103, 2013.

[219] V. Phan and H. Lee, “Enhanced Proportional-Resonant Current Controller for UnbalancedStand-alone DFIG-based Wind Turbines,” *Journal of Electrical Engineering & Technology ,* vol. 5, no. 3, pp. 443-450, 2010.

[220] “Improved Control of Dynamic Responses of Wind Driven DFIG under Fault Condition,” in *International Conference on Computing Communication Control and Automation*, Pune, 2015.

[221] T. Shi, Z. Xie and L. Wang, “Harmonie current suppression for DFIG based on PR controller,” in *34th Chinese Control Conference (CCC)*, Hangzou, 2015.

[222] R. Hiremath and T. Moger, “Comparative Analysis of Different Controllers for Enhancing the LVRT of DFIG system,” in *International Renewable and Sustainable Energy Conference (IRSEC)*, Morocco, 2021.

[223] R. Hiremath and T. Moger, “Performance of DFIG-Wind Turbine Generator System for LVRT Enhancement using Proportion-Resonant Controller,” in *International Renewable and Sustainable Energy Conference (IRSEC)*, Morocco, 2021.

[224] S. Omrani and M. R. Azizian, “PR-Fuzzy Control Improvement of Doubly Fed Induction Generator During Grid Faults,” in *Iranian Conference on Renewable Energy & Distributed Generation (ICREDG)*, Tehran, 2019.

[225] J. Wang, Y. Zhong and W. Song, “PR Control for Two-Stage Matrix Converter Excitation Doubly Fed Wind Generation System under Unbalanced Grid Voltage Conditions,” in *Asia-Pacific Power and Energy Engineering Conference*, Shangai, 2012.

[226] L. Dong, J. Kong, J. Feng and Y. Zhang, “Subsynchronous Resonance Mitigation for Series Compensation Transmission System of DFIG Based on PR Control,” in *IEEE 10th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Xi'an, 2019.

[227] D. Adolfo, D. P. Andrea, D. N. L. Pio and M. Santolo, “PSO-PR power flow control of a single-stage grid-connected PV inverter,” in *IEEE 6th International Conference on Renewable Energy Research and Applications*, San Diego, 2017.

[228] Z. Akhtar, J. Zhu and Huse, “Proportional Resonant Controller Tuning in Three-Phase Four-Leg VSI Based on Particle Swarm Optimization,” in *International Power Electronics and Motion Control Conference (PEMC)*, Gliwice, 2021.

[229] M. G. M. Abdolrasol, M. A. Hannan, S. M. S. Hussain and T. S. Ustun, “Optimal PI controller based PSO optimization for PV inverter using SPWM techniques,” in *International Conference on Power and Energy Systems Engineering*, Fukouka, 2021.

[230] B. Parandhaman, S. K. Nataraj and C. B. Baladhandautham, “Optimization of DC-link voltage regulator using Bat algorithm for proportional resonant controller-based control of shunt active power filter in distribution network,” *International Transactions on Electrical Energy Systems,* 2020.

[231] J. Faira, J. Fermeiro, J. Pombo, M. Calada and S. Mariano, “Proportional Resonant Current Control and Output-Filter Design Optimization for Grid-Tied Inverters Using Grey Wolf Optimizer,” *International Congress on Engineering-Engineering for evolution,* vol. 13, no. 8, 2020.

[232] O. B. Alzain, X. Liu, X. Kong, A. B. Alzain, M. A. Abdalla and D. A. Ali, “Improved DFIG-WPGS Efficiency via the tuned PI-Resonant Unit Controller based on BAT Algorithm,” in *International Conference on Computer, Control, Electrical, and Electronics Engineering* , Khartoum, 2017.

[233] J. Yang, L. Qu and Y. Shen, “Swarm Intelligence in Data Science: Applications, Opportunities and Challenges,” *Advances in Swarm Intelligence,* pp. 3-14, 2020.

[234] K. Naidu, M. S. Ali, A. H. A. Bakar, C. K. Tan, H. Arof and H. Mokhlis, “Optimized artificial neural network to improve the accuracy of estimated fault impedances and distances for underground distribution system,” *Plos One,* vol. 15, no. 1, 2020.