

## **A Hybrid Dragonfly Algorithm for Efficiency Optimization of Induction Motors**

### Author

Shukla, Niraj Kumar, Srivastava, Rajeev, Mirjalili, Seyedali

### Published

2022

### Journal Title

Sensors

### Version

Version of Record (VoR)

### DOI

[10.3390/s22072594](https://doi.org/10.3390/s22072594)

### Rights statement

© 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

### Downloaded from

<http://hdl.handle.net/10072/415918>

### Griffith Research Online

<https://research-repository.griffith.edu.au>

## Article

# A Hybrid Dragonfly Algorithm for Efficiency Optimization of Induction Motors

Niraj Kumar Shukla <sup>1,\*</sup>, Rajeev Srivastava <sup>2</sup> and Seyedali Mirjalili <sup>3,4</sup>

<sup>1</sup> Department of Electrical Engineering, Shambhunath Institute of Engineering and Technology, Prayagraj 211015, India

<sup>2</sup> Department of Electronics and Communication, J K Institute of Applied Physics and Technology, University of Allahabad, Prayagraj 211002, India; rajeev\_jk@rediffmail.com

<sup>3</sup> Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Brisbane, QLD 4006, Australia; ali.mirjalili@gmail.com

<sup>4</sup> Yonsei Frontier Laboratory, Yonsei University, Seoul 03722, Korea

\* Correspondence: nks.gemini@gmail.com

**Abstract:** Induction motors tend to have better efficiency on rated conditions, but at partial load conditions, when these motors operate on rated flux, they exhibit lower efficiency. In such conditions, when these motors operate for a long duration, a lot of electricity gets consumed by the motors, due to which the computational cost as well as the total running cost of industrial plant increases. Squirrel-cage induction motors are widely used in industries due to their low cost, robustness, easy maintenance, and good power/mass relation all through their life cycle. A significant amount of electrical energy is consumed due to the large count of operational units worldwide; hence, even an enhancement in minute efficiency can direct considerable contributions within revenue saving, global electricity consumption, and other environmental facts. In order to improve the efficiency of induction motors, this research paper presents a novel contribution to maximizing the efficiency of induction motors. As such, a model of induction motor drive is taken, in which the proportional integral (PI) controller is tuned. The optimal tuning of gains of a PI controller such as proportional gain and integral gain is conducted. The tuning procedure in the controller is performed in such a condition that the efficiency of the induction motor should be maximum. Moreover, the optimization concept relies on the development of a new hybrid algorithm, the so-called Scrounger Strikes Levy-based dragonfly algorithm (SL-DA), that hybridizes the concept of dragonfly algorithm (DA) and group search optimization (GSO). The proposed algorithm is compared with particle swarm optimization (PSO) for verification. The analysis of efficiency, speed, torque, energy savings, and output power is validated, which confirms the superior performance of the suggested method over the comparative algorithms employed.

**Keywords:** induction motor; PI controller; speed control; dragonfly algorithm; group search optimizer; particle swarm; optimization; algorithm; meta-heuristic



**Citation:** Shukla, N.K.; Srivastava, R.; Mirjalili, S. A Hybrid Dragonfly Algorithm for Efficiency Optimization of Induction Motors. *Sensors* **2022**, *22*, 2594. <https://doi.org/10.3390/s22072594>

Academic Editor: Alex Alexandridis

Received: 5 February 2022

Accepted: 19 March 2022

Published: 28 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Nowadays, asynchronous motors are massively used in the process of industrial applications with the requirement of a larger performance. From the IM [1,2] discovery period, it is assumed as the actuator dominance with a constant speed tactic. Within several decades, this has been deployed in numerous industries as drivers or actuators for generating mechanical forces and motions. Every year in industries, the operational units of IM are increasing; hence, the enhancement of efficiency is a top priority for industries by minimizing the losses which will increase as a result of global conservation of energy. Some of the challenging characteristics of IM, specifically the cage IM over DC motor, are less cost, small inertia, less maintenance, high efficiency, simpler design, inherently self-starting, and absence of collector brush systems. However, there are some demerits in IM, which

is complex in nature, including a multivariable mathematical model, nonlinearity, and intrinsically incompetent offering of flexible speed operation. The aforesaid issues can be solved by constructing the motor control operators with smart scalar and vector strategies.

The control approach has assumed that the reference magnetic flux [3,4] is stable and nearer to its rated level ( $\lambda_{rm}$ ) on behalf of larger dynamic performance in spite of operating points. The resultant outcome of such procedures has unacceptable energy efficiency, while the IM is assumed as under-loaded. Most of the research explorations have demonstrated that about 45% of IMs drive 40% of their rated load. To tackle this challenge, the auto-adjustment of flux has to be made online regarding the load torque for attaining the reduced power loss. Actually, the flux leakage value with unique optimal on every operating point is tracked by the efficiency optimization algorithm derived [5–7] and pertained it to controlled IM [8,9]. For optimizing the drive system's efficiency, many techniques have been evolved in the literature. They are categorized generally into two major models: MBO and SAO.

The MBO has possessed some challenges, which is its larger sensitivity to parameter variations and the count of arithmetic operations contained within the loss model's solution. The characterization of SAO in [10,11] is made by its parameter insensitivity. However, it poses some promising challenges such as very slow convergence and high torque ripples to the optimal operating point while distinguished over MBO. In such cases, it has a very large search space and thereby outcomes in large time in the quest for optimal operating conditions. In fact, the optimization algorithm in [12,13] may underestimate the optimal flux with no adaptation of real-time parameters' mechanisms, which may as well direct the destabilization of the drive system and impact the problem of "motor stalling", particularly in the existence of an immediate change in load torque. It should be noted that in the full operating region, there is a critical issue in the identification of IM parameters accurately [14–16] and simultaneously, even by using the IM parameter stacking algorithms. Over the last few years, researchers have been working on evolutionary algorithms [17] such as genetic algorithms (GA) and swarm-intelligence-based algorithms [18] such as PSO, GSO, DA, FF, and ACO for optimization purposes under different working environments. These meta-heuristic algorithms are very efficient and determine an accurate estimation of optimal solutions compared to conventional algorithms.

This framework has introduced a novel model to maximize the efficiency of induction motors. For this, the objective model is constructed using the speed-control induction motor drive with the tuning of the PI controller. In this PI controller, the optimal tuning of PI controllers such as proportional gain and integral gain is made by introducing a newly hybrid algorithm named SL-DA, which is the hybridization of DA and GSO. The rest of the paper is organized as follows: Section 2 explains the review on the literature of induction motor and the methodology that is discussed. Section 3 presents the proposed model of AC sensor-less speed control of the induction motor drive. Section 4 presents the original DA, GSO, and PSO along with the proposed SL-DA algorithm. The experimental results are discussed in Section 5. Finally, Section 6 concludes the research work and suggests future directions.

## 2. Literature Review

The challenges and the features of traditional induction motor models are described in this section. Various algorithms are deployed for the efficient working of IM. Some of them are explained in the following. An SVM-based DTC strategy for IM [19] has presented for better optimization of IM energy and has minimized flux ripples and torque and further has constant switching frequency and good waveform of the stator current. A better dynamic and larger robustness has been ensured by this nonlinear approach against external disturbance. However, there are disadvantages of variable switching frequency and increased control difficulty at low-speed regions. A DFOC method [20,21] was proposed that maximizes electric energy saving and has better control of rotor flux

and motor speed. However, it has poor sensitivity and parameter variation, and controller complexity is increased.

Adaptive hybrid LMT by Farhani et al. [22] poses enhanced motor efficiency and has a maximum energy saving, but the power-saving potential is mainly based on robustness and accuracy of flux regulation. A flux search controller [23] has improved the efficiency and accuracy of loss calculation. On the basis of adaptive gradient descent of motor flux value having quick response and easier implementation, the flux search controller was implemented. The implemented model was presented with the combination of a suitable loss method of a six-phase induction motor. However, it needs additional hardware and has slow convergence and torque variations. In 2017, Kong et al. [24] proposed a harmonic current injection method that defines better efficiency and torque density and has increased power density; still, it is complex in nature. The analysis on the impact of harmonic currents on yoke flux density and air-gap flux density has been made altogether. The implemented model has wholly made the utilization of optimal coefficients for equivalent harmonic current, and multiple control freedoms were extracted. The torque density has enhanced the nine-phase induction motor by 60% by sustaining the similar stator current density with the implemented model. The construction of three-phase induction motor was completed for validating the efficiency of multiphase motors. In 2015, Laamari et al. [25] presented a narrative model on the basis of grouping of the extensive Kalman filter with PSO for evaluating the rotor and speed flux of an induction motor drive and suggested that PSO has precise estimation and faster optimization. Further poses better performance in load torque and rotor resistance. However, the scattering problems cannot be exploited and are critical to explain the initial design parameters. The effectiveness of the implemented model has been evaluated using the simulation experiment on rotor and speed flux estimation.

In 2018, Zeb et al. [26] exploited several narrative, robust, and adaptive control mechanisms, called (a) FLC on the basis of LMA, (b) FLC on the basis of SDA, (c) FLC on the basis of NA, and (d) FLC on the basis of GNA for the IVC 3 phase IM. FLC has less power consumption and voltage dips. Further, it has a fast dynamic response and is robust to parameter uncertainties. Despite this, it lacks real-time response, has low speed, and takes a longer time to run. In 2018, Costa et al. [27] suggested that ACO has better performance, and the retuning of controllers is not required. Nevertheless, it lacks secondary information usage and has slow sequential processing. This approach has been taken within the DTC-SVM control loops that include stator flux linkage, computation of the linkage stator flux, electromagnetic torque, and rotor speed. In Table 1, we summarize the main contributions in the field of induction motors for optimizing efficiency.

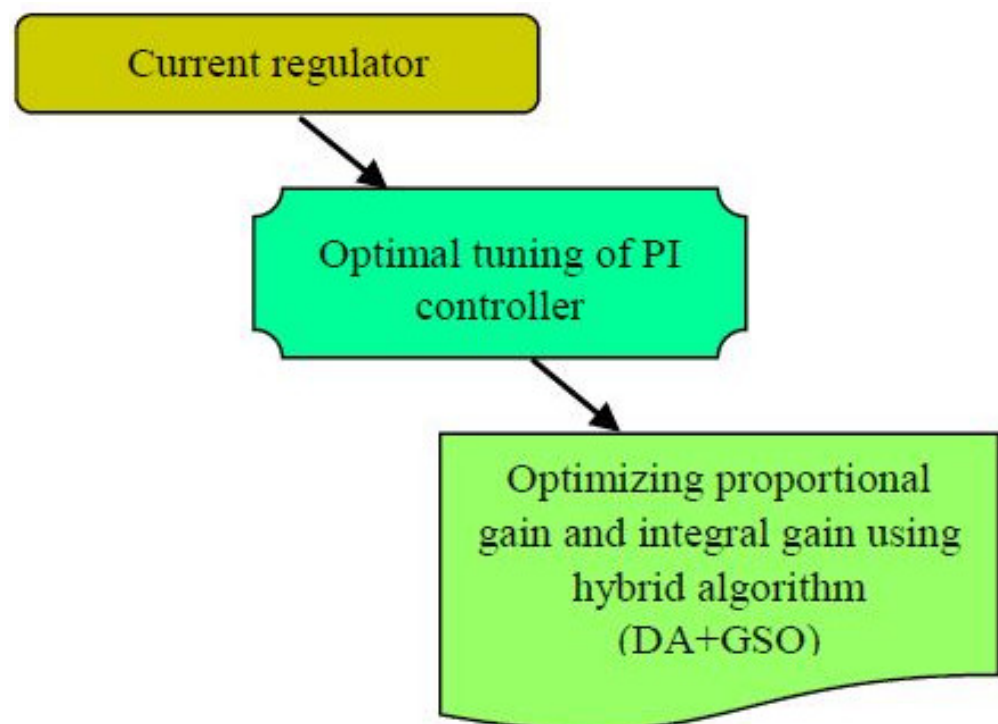
### *Proposed Methodology*

Induction motors are considered the industries' workhorse because of their efficiency, power/mass relation, low cost and quite maintenance-free operation in their life span. Nonetheless, the motors that work on low efficiency waste more energy, which in turns maximizes the operational cost. As for the consequence of a large count of operating units and large energy consumption, even a small rise in improvement of efficiency has had a considerable effect on operational cost and the whole energy consumption. To make this possible, this proposal aims to introduce a newly hybrid algorithm that hybridizes the concept of dragonfly algorithm (DA) and group search optimization (GSO). DA is a newly meta-heuristic optimization technique [28] that solves the single-objective, discrete, and multi-objective problems. GSO is the algorithm [29] that is inspired based on animal searching behavior, which solves the continuous optimization problems. Furthermore, for a large amount of numerical testing, Mirjalili suggested that DA executes in a better way as compared to GA and PSO [30]. Conventionally, in [31,32], the authors used the key attributes of loss model control (LMC) as well as search control (SC) together for assessment and reproduction of optimal flux component of current ( $I_{ds}$ ), for optimal efficiency operation of IM. However, the model struggles with the performance rate when there is variation in the load profile. Hence, to make a solution for this, this proposal introduces a

new contribution that keeps the ( $I_{ds}$ ) value constant and optimizes the current regulator by optimally tuning the proportional integral (PI) controller as depicted in Figure 1, by improving the PI controller parameters such as proportional gain and the integral gain as well. This optimal selection obviously paves way for the optimal efficiency operation of induction motors.

**Table 1.** Methodology used in traditional induction motor models.

Methodology	References
SVM-DTC	Ammar et al., (2017) [20]
DFOC	Farhani et al., (2017) [21]
Adaptive Hybrid LMT	Farhani et al., (2017) [22]
Flux Search Controller	Taheri et al., (2012) [23]
Harmonic Current Injection Method	Kong et al., (2017) [24]
Particle Swarm Optimization	Laamari et al., (2015) [25]
Fuzzy Logic Controller	Zeb et al., (2018) [26]
Ant Colony Optimization	Costa et al., (2018) [27]
Dragonfly Algorithm	Jafari and Chaleshtari (2017) [28]
Group Search Optimizer	He et al., (2009) [29]
Hybrid Dragonfly Algorithm	Wang et al., (2021) [30]
Neural Network	Choudhary et al., (2015) [31]
ANFIS Controller	Shukla et al., (2020) [32]
Proportional Integral Controller	Idi et al., (2015) [33]



**Figure 1.** Proposed efficiency optimization model of induction motors.

### 3. Model of AC Sensor-Less Speed Control of Induction Motor Drive

#### General Block Representation

The system consists of a pure DC supply, which needs to be converted into AC by employing an Insulated Gate Bipolar Transistor (IGBT) inverter. An IGBT-based 3-phase inverter is used in this paper, which consists of twelve IGBT switches, out of which four IGBT switches per phase are arranged in an H-Bridge configuration. The scheme is depicted in Figure 2. A three-phase asynchronous machine is implemented within the asynchronous machine block, which is termed as single squirrel-cage, wound rotor, or double squirrel-cage. Here, the operation is made in motor mode where motor torque and speed are varied under different conditions. For the conventional method, IM uses the vector control technique where the voltage, current, and torque of a 3-phase induction motor are controlled by time-varying components. The speed is used as a feedback loop in a PI controller at different values of load torque. The proposed speed controller (AC) block exemplifies a PI speed regulator technique for AC machines that is utilized within vector-controlled drives. Here, two operating instances are present: the initial one is having both the outputs of torque as well as the flux references and the second has only the output of the torque reference. The speed of the machine is given in rpm as  $N$ ; the speed reference of the machine is in rpm and is denoted as  $N^*$ . In this, the machine's torque reference is termed as input, when the parameter of regulation type is fixed as torque. The flux reference is symbolized as  $Flux^*$  and the unit of this is weber. The induction motor speed is controlled by tuning the PI controller; i.e., the proportional and integral gain of the PI controller is fine-tuned via a proposed SL-DA scheme, which is deeply explained in Section 4.

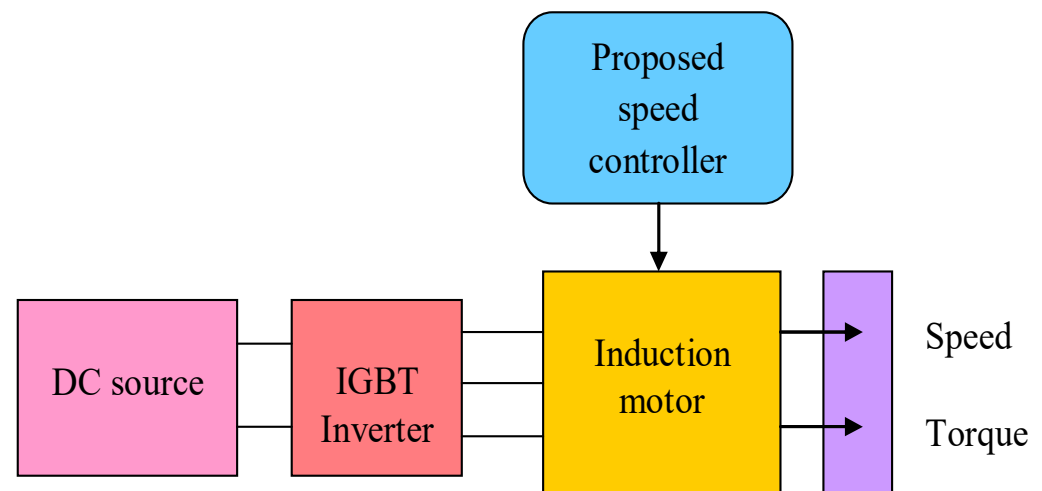


Figure 2. Representation of proposed model.

### 4. Proposed Efficiency Optimization in Induction Motor

#### 4.1. Optimized Speed Control

The speed control concept is depicted in Figure 3. Here, the quadrature current is expressed as  $I_q$ , the direct current is given as  $I_d$ , the reference quadrature current is exemplified as  $I_q^*$ , the reference direct current is specified as  $I_d^*$ , and the reference three-phase current is delineated as  $I_{abc}^*$ . The PI controller [33,34] is considered as a renowned controller which poses the ability to maintain the exact set points. The operational modeling of PI controller is on the basis of grouping of two controller modes viz. proportional and the integral mode. The logical expression for PI controller is gained from  $k_p$  and  $k_i$  parameters. The common form of PI controller is stated as per Equation (1), where  $k_p$  is the proportional gain and  $k_i$  is termed as integral gain.

$$u(t) = k_p e(t) + k_i \int e(t) dt \quad (1)$$

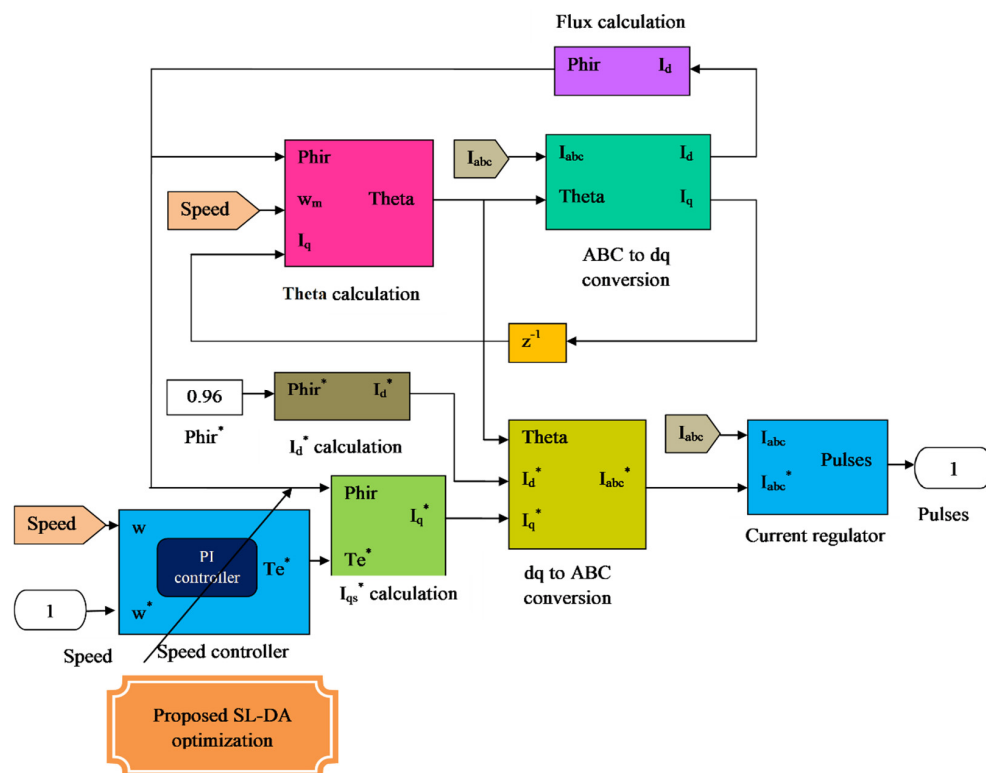


Figure 3. Systematic representation of SL–DA based speed-control block.

Using the Laplace transform, Equation (1) is changed to Equation (2).

$$U(s) = k_p E(s) + k_i \frac{E(s)}{s} \tag{2}$$

PI regarding the time constraints is expressed in Equation (3).

$$U(s) = k_p \left[ 1 + \frac{1}{\tau_i} \right] E(s) \tag{3}$$

where  $k_i = \frac{k_p}{\tau_i}$  and  $k_p = k_d / \tau_d$ .

The proportional and integral terms provide fine control for the error signal and minimize the steady-state errors of the model. For a closed-loop control system, when precise tuning of  $k_p$  and  $k_i$  values are performed, then there is a small decrease in the rise time and improvement in settling time [34]. Many software-based concepts have been realized for the tuning purpose of the controllers. The main contribution or novelty of the work is involved in the speed-control block, where the optimized PI controller is newly implemented to enhance the controlling performance. In Figure 4, the two gains  $k_p$  and  $k_i$  of the PI controller inside the flux PI block are optimized using the proposed optimization algorithm named SL-DA.

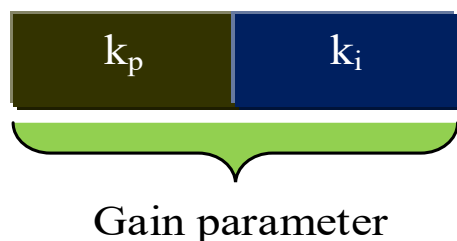


Figure 4. Solution encoding of proposed algorithm.

#### 4.2. Objective Function with Gain Encoding

During the optimization of gains in the PI controller inside the speed-control block, the fixed objective function is to maximize the efficiency of the induction motor, which is as per Equation (4). The computation formula for the efficiency of induction motor is based on Equation (5).

$$obj = \max(\eta) \quad (4)$$

$$\eta = \left[ \frac{Speed/S_p}{Tem/T_m} \right] \quad (5)$$

In Equation (5), *Speed* refers to the output rotor speed,  $S_p$  indicates the reference speed,  $Tem$  denotes the electromagnetic torque, and  $T_m$  refers to the load torque. As mentioned earlier, the above objective function is attained by optimizing or tuning the gains  $k_p$  and  $k_i$  of the PI controller.

#### 4.3. Procedure for Gain Update

For maximizing the efficiency of the induction motor, a procedure is fixed here, which depends on the fine tuning of PI controller gains. Figure 5 represents the flowchart of the proposed methodology. Initially, the load torque  $T_m$  and reference speed  $S_p$  are set. Further, the initialization of gains  $k_p$  and  $k_i$  is performed, and the rotor speed *Speed* and electromagnetic torque  $Tem$  are calculated. With these collected parameters, the induction motor's efficiency is calculated. The efficiency is considered as the objective function, and the solution is updated with  $k_p$  and  $k_i$  using the proposed SL-DA algorithm. The parameters are continuously updated along with the solution update until the termination is reached, which in turn provides the best solution.

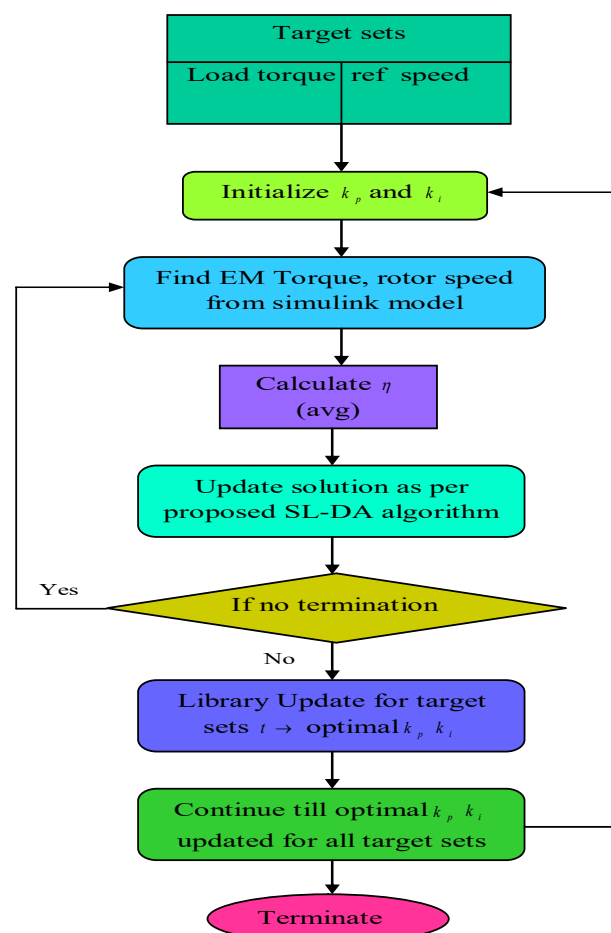


Figure 5. Procedure adopted for gain update in PI controller of speed-control block.



#### 4.4. Conventional DA Algorithm

Across the world, 3000 diverse species of dragonfly are found, and their lifecycle contains two main stages: nymph and adult. Most of their lifecycle is comprised of the nymph stage; after that, the dragonfly becomes adult by metamorphosis. The static as well as dynamic swarming behaviors of dragonflies have been mimicked in DA [28]. Dragonflies work as small clusters in the static swarm (exploration) stage and fly to and from inside a closer area for prey hunting. The core aspects of a static swarm are the abrupt changes and local movements within the flying path. The dynamic swarms (exploitation) have an enormous count of dragonflies for making the swarm migrate over long distances in a particular direction. Moreover, survival is the core objective of any swarm and hence the entirety of the swarm's individuals can be drawn to food sources and divert to the external enemies. Based on the above-mentioned two behaviors, the position update of swarm individuals is of five types, namely: alignment, control cohesion, attraction (over food sources), distraction (over external enemies), and separation. Equation (6) denotes the separation of the  $i^{th}$  dragonfly and is denoted as  $D_i$  to form its neighbor.

$$D_i = - \sum_{j=1}^{NI'} (Z' - Z'_j) \quad (6)$$

In this, the current individual's position is given as  $Z'$ , the  $j^{th}$  neighboring individual's position is denoted as  $Z'_j$ , and the count of neighboring individual is stated as  $NI'$ . Equation (7) estimates the alignment of such dragonflies. Here, the velocity of  $j^{th}$  neighboring individual is delineated as  $v'_j$ . The calculation of cohesion is depicted as per Equation (8). Equation (9) explains the estimation of attraction over the food source. In this, the food source position is specified by  $Fd$ . Equation (10) shows the computation of distraction over the external enemies. In this, the position of the enemy is given by  $Eny$ .

$$Al_i = \frac{\sum_{j=1}^{NI'} v'_j}{NI'} \quad (7)$$

$$Ch_i = \frac{\sum_{j=1}^{NI'} Z'_j}{NI'} - Z' \quad (8)$$

$$Fs_i = Fd - Z' \quad (9)$$

$$EE_i = Eny + Z' \quad (10)$$

The dragonflies' behavior is considered on the basis of these five corrective patterns. In search space, the position of dragonflies' update as well as the movements of their simulation are made using two vectors, namely position  $Z'$  and step  $\Delta Z'$ . The step vector is the same as the PSO's velocity vector, and on the basis of this PSO algorithm, the development of the DA algorithm is made. Equation (11) has demonstrated the step vector which controls the movement of dragonflies.

$$\Delta Z'_{t+1} = (d' D_i + a' Al_i + c' Ch_i + f' Fs_i + e' EE_i) + \delta \cdot \Delta Z'_t \quad (11)$$

where the separation weight is denoted as  $d'$ , the  $i^{th}$  individual separation is given by  $D_i$ , the alignment weight is depicted as  $a'$ , the alignment of  $i^{th}$  individual is delineated as  $Al_i$ , the cohesion weight is  $c'$ , the  $i^{th}$  individual cohesion is  $Ch_i$ , the food factor is given as  $f'$ , the food source of  $i^{th}$  individual is expressed as  $Fs_i$ , the enemy factor is indicated by  $e'$ , the  $i^{th}$

individual enemy position is portrayed as  $EE_i$ , the inertia weight is symbolized with  $\delta$ , and the iteration count is defined as  $t$ . The position vector estimation is as per Equation (12).

$$Z'_{t+1} = Z'_t + \Delta Z'_{t+1} \quad (12)$$

where  $t$  expresses the current iteration. The dragonflies need to roam over a search space in a random walk manner (Levy flight) for improving the stochastic behavior, randomness, and exploration, when having no neighbor solution. At this stage, the updated position of the dragonflies is made using Equation (13).

$$Z'_{t+1} = Z'_t + Levy(h) \times Z'_t \quad (13)$$

$$Levy(h) = 0.01 \times \frac{ran_1 \times \Phi}{|ran_2|^{\frac{1}{\xi}}} \quad (14)$$

$$\Phi = \left( \frac{\Gamma(1 + \xi \times \sin(\frac{\pi\xi}{2}))}{\Gamma(\frac{1+\xi}{2}) \times \xi \times 2^{(\frac{\xi-1}{2})}} \right)^{\frac{1}{\xi}} \quad (15)$$

$$\Gamma(z) = (z - 1)! \quad (16)$$

The dimension of position vectors is expressed by  $h$ , the two arbitrary numbers within the range 0 and 1 are denoted as  $ran_1$  and  $ran_2$ , and  $\xi$  is assigned to be a constant value. The steps as well as the position of each and every dragonfly within every iteration are upgraded by using Equations (11)–(13). The dragonfly identification is made through assessing the Euclidean distance between the total dragonflies and selects  $NI'$  for updating  $Z'$  and  $\Delta Z'$  vector. At this position, the upgradation process is repeated up until the final criteria is met. The pseudo-code representation of the conventional DA model is given in Algorithm 1.

---

**Algorithm 1:** Pseudo code of conventional DA algorithm

---

```

Initiate the dragonfly population using  $Z_i(i = 1, 2, \dots, n)$ 
Start off the step vectors  $\Delta Z_i(i = 1, 2, \dots, n)$ 
While the termination criteria is not achieved
    Compute the objective values of all dragonflies
    Upgrade the enemy as well as food source
    Upgrade  $d, a, c, f$ , and  $e$ 
    Estimate  $D, Al, Ch, Fs$ , and  $EE$  using Equations (6)–(10)
    Upgrade the neighboring region radius
    When the dragonfly constitutes a least-neighbor dragonfly
        The velocity vector is upgraded per Equation (11)
        The position vector is upgraded per Equation (12)
    Else
        The position vector is upgraded per Equation (13)
    End if
    The new locations are validated and corrected according to different boundaries
End while

```

---

#### 4.5. Conventional GSO Algorithm

The GSO algorithm with a population cluster is called a group, and every individual within the population called a member [24]. Over a search space of  $n$ -dimensional, the  $p^{th}$  member at  $l^{th}$  searching iteration (bout) provides a present position  $Z_p^l \in \mathfrak{R}^n$ , a head angle  $\phi_p^l = (\phi_{p_1}^l, \dots, \phi_{p_{(n-1)}}^l) \in \mathfrak{R}^{n-1}$ . The  $p^{th}$  member's search direction is the unit vector

$F_p^l(\phi_p^l) = (f_{p_1}^l, \dots, f_{p_n}^l) \in \mathfrak{R}^n$ , which is computed from  $\phi_p^l$  through a polar to a Cartesian coordinate transformation, which is stated in Equations (17)–(19).

$$f_{p_1}^l = \prod_{k=1}^{n-1} \cos(\phi_{p_k}^l) \quad (17)$$

$$f_{p_q}^l = \sin(\phi_{p_{(q-1)}}^l) \cdot \prod_{k=q}^{n-1} \cos(\phi_{p_k}^l) \quad (q = 2, \dots, n-1) \quad (18)$$

$$f_{p_n}^l = \sin(\phi_{p_{(q-1)}}^l) \quad (19)$$

In the GSO algorithm, there are three members' kinds in a group: scroungers and producers having behaviors on the basis of PS model, and the dispersed one which evaluates the motion of random walk. While taking the optimization issues, the open patches are the ones that are assigned to be unknown optima and that are arbitrarily distributed within a search space. Therefore, the searching of patches is made by moving within the search space by the group members. In accordance with the relevant phenotypic features, the producer and the scroungers are considered to be the same; hence, the two roles can switch with each other.

In iteration, a group of members that is located in the more challenging area and is presented as the premier fittest value is selected as producer. After that, it halts and scrutinizes the whole environment for seeking the optima (resources). Scanning is considered as a core module within the search direction. In GSO algorithm, the behavior of the producer  $Z_g$  at the  $l^{th}$  iteration is given as follows:

Step (1): The scanning will be made by producer at  $0^\circ$  and after that scan laterally by arbitrarily sampling three points within the scanning field: one point within  $0^\circ$ , given as per Equation (20); one point at the right side of hypercube, given in Equation (21); and one point in the left hand of the hypercube, stated in Equation (22).

$$Z_x = Z_g^l + rn_1 v_{\max} F_g^l(\phi^l) \quad (20)$$

$$Z_{rn} = Z_g^l + rn_1 v_{\max} F_g^l(\phi^l + rn_2 \theta_{\max}/2) \quad (21)$$

$$Z_v = Z_g^l + rn_1 v_{\max} F_g^l(\phi^l - rn_2 \theta_{\max}/2) \quad (22)$$

Here, the arbitrarily distributed random number having mean 0 and std. deviation 1 is denoted as  $rn_1$ , and the sequences are distributed uniformly within the interval (0,1) which is stated as  $rn_2$ .

Step (2): The best point will be determined by the producer with the best resources, i.e., fitness value. When the best point is provided with better resources rather than the current position, then the member will move over this point. Otherwise, the members will remain in their present position and twist their heads toward a novel generated angle at random. In Equation (23), the maximum turning angle is denoted as  $\alpha_{\max} \in \mathfrak{R}^1$ .

$$\phi^{l+1} = \phi^l + rn_2 \alpha_{\max} \quad (23)$$

Step (3): The heads will be turned to zero degree if the producer is not able to determine a better area after the  $a$  iteration. In Equation (24),  $a \in \mathfrak{R}^1$  is a constant.

$$\phi^{l+a} = \phi^l \quad (24)$$

At the time of every spell on searching, the group member count is selected as scroungers. The opportunities for joining the resources found by the researchers will be searched by these scroungers. This GSO algorithm only takes the area-copying feature, which is assigned as the basic scrounging behavior. The modeling of the area-copying

behavior of  $p^{th}$  scrounger on the  $l^{th}$  iteration has depicted as a random walk over the producer and is stated in Equation (25).

$$Z_p^{l+1} = Z_p^l + rn_3 \circ (Z_g^l - Z_p^l) \quad (25)$$

Here, the uniform random sequence in the interval (0,1) is depicted by  $rn_2 \in \mathfrak{R}^n$ . The Hadamard product or the Schur product is given as 'o', where the entry-wise product of two vectors is calculated.

The ranging is happened, when the  $p^{th}$  group member gets dispersed, and this is called disperse member rangers. Naturally, the searching concepts are performed by the ranging animals that involve systematic search strategies and random walks for locating the resources effectively. The rangers only deploy the random walks and are given in Equation (26).

$$v_p = a.rn_1 v_{max} \quad (26)$$

and move to a new point that is stated as per Equation (27).

$$Z_p^{l+1} = Z_p^l + v_p F_p^l(\phi^{l+1}) \quad (27)$$

The conventional GSO algorithm pseudo code is explained in Algorithm 2.

---

**Algorithm 2:** Conventional GSO algorithm

---

Assume  $l = 0$

Initiate position randomly  $Z_p$  and head angles  $\phi_p$  of entire group members

Compute the fittest values of initial members  $f(Z_p)$

while (the stopping criteria not met)

  for (every member  $p$  in the group)

    Select  
    producer

    Determine the producer  $Z_p$  of the group members

    Execute  
    producing

    (1) The producer will examine at  $0^\circ$  and then scrutinize laterally by arbitrarily sampling three points in the scanning field using (20) to (22)

    (2) Discover the finest point with the foremost resource (fittest value). If the finest point shows a more preferable resource than its present location, then it will move at this point. If not, it will remain in its present location and move its head toward a new angle using Equation (23)

    (3) When the producer cannot determine a preferable area after an iteration, it will move its head back to  $0^\circ$  using (24)

    Execute  
    scrounging

    To carry out scrounging, select 80% from the remaining members

    Execute  
    dispersion

    For the remaining members, they will be distributed from their present location to execute ranging: (1) produce an arbitrary head angle using (5) and (2) select a random distance  $v_p$  from the Gauss distribution using (8) and proceed towards a new point using (9)

    Calculate  
    fitness

    Compute the fittest value of the present member  $f(Z_p)$

  End for

  assume  $l = l + 1$

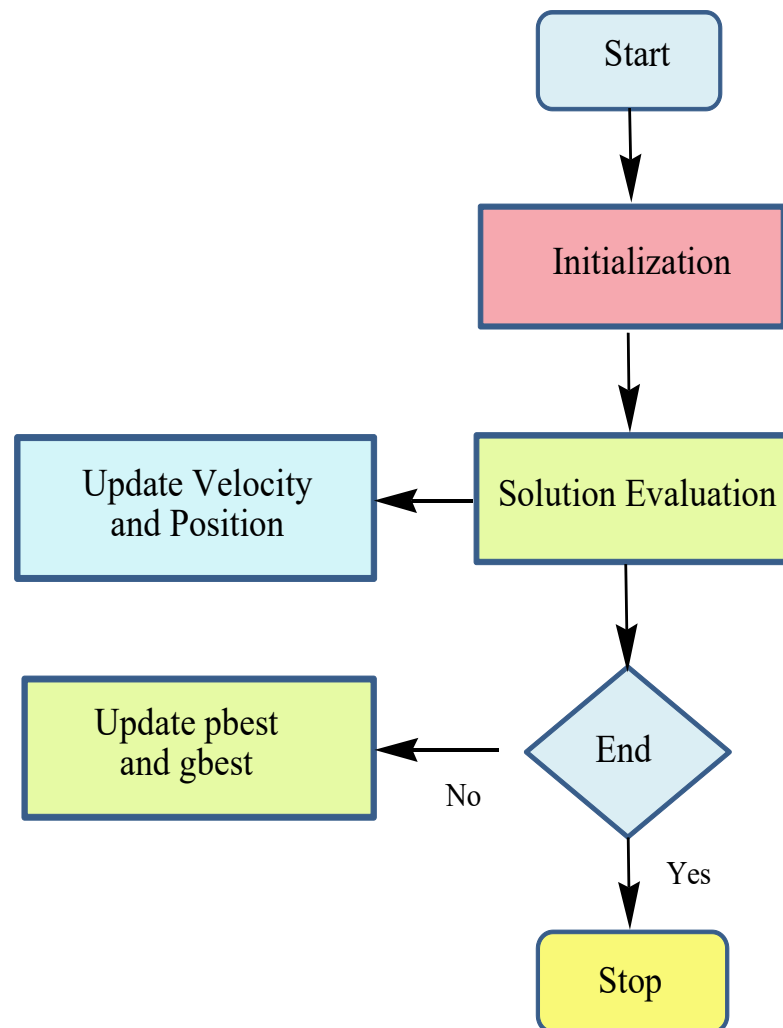
End while

---

#### 4.6. Conventional PSO Algorithm

In 1995, Kennedy and Eberhart [35] introduced a bio-inspired algorithm known as particle swarm optimization (PSO), which was based on the swarm intelligence technique. In this algorithm, a flock of birds moves in a group to navigate and forage. Each particle is a vector of values for the decision variables of the problem that gets updated in each iteration using a velocity vector. The velocity vector considers the current velocity of the particle,

movement toward personal best ( $pbest$ ), and tendency toward the global best ( $gbest$ ). The flowchart of the PSO algorithm is depicted in Figure 6.



**Figure 6.** Flowchart of PSO algorithm.

#### 4.7. Proposed SL-DA Algorithm

DA has achieved much attention among researchers from diverse fields because of its simplicity [36,37]. However, it needs improvement in internal memory as it directs to premature convergence to local optima [38]. Similarly, GSO is the well-known population-based optimization technique that is based on the inspiration of animal group living theory and searching behavior. The main shortcomings in this algorithm are its poor exploration and exploitation ability and slower convergence.

In order to alleviate the above-mentioned drawbacks, this paper proposes a new optimization algorithm called the Scrounger Strikes Levy-based dragonfly algorithm (SL-DA), which incorporates the concept of GSO within the DA algorithm [39]. The proposed framework is detailed as follows: in the conventional DA algorithm, if the condition dragonfly poses, at least one neighbor dragonfly is satisfied; the velocity vector and position vector update are exploited as per Equations (11) and (12), respectively. Otherwise, the update of the position vector is handled using the Levy update in Equation (13). In this proposed model, the same condition is checked and if satisfied, the velocity vector and position vector are made the same as the conventional DA model. In the else condition, the scrounger behavior of the GSO in Equation (25) is carried out instead of the Levy update in the conventional model. The proposed SL-DA algorithm's pseudo-code is explained in Algorithm 3. The proposed SL-DA algorithm's flowchart is illustrated in Figure 7.

**Algorithm 3:** Pseudo code of proposed SL-DA algorithm

```

Initiate the dragonfly population  $Z_i(i = 1, 2, \dots, n)$ 
Start off the step vectors  $\Delta Z_i(i = 1, 2, \dots, n)$ 
While the termination criteria are not met
    Compute the objective values of whole dragonflies
    Upgrade the enemy as well as food source
    Upgrade  $d, a, c, f$ , and  $e$ 
    Estimate  $D, Al, Ch, Fs$ , and  $EE$  using Equations (6)–(10)
    Upgrade the region of neighboring radius
    When the dragonfly constitutes a least-neighbor dragonfly
        Update velocity vector as per Equation (11)
        Update position vector as per Equation (12)
    Else
        Update position vector as per the scrounger behavior of the GSO in Equation (25)
    End if
    The new locations are validated and corrected according to different boundaries
End while
    
```

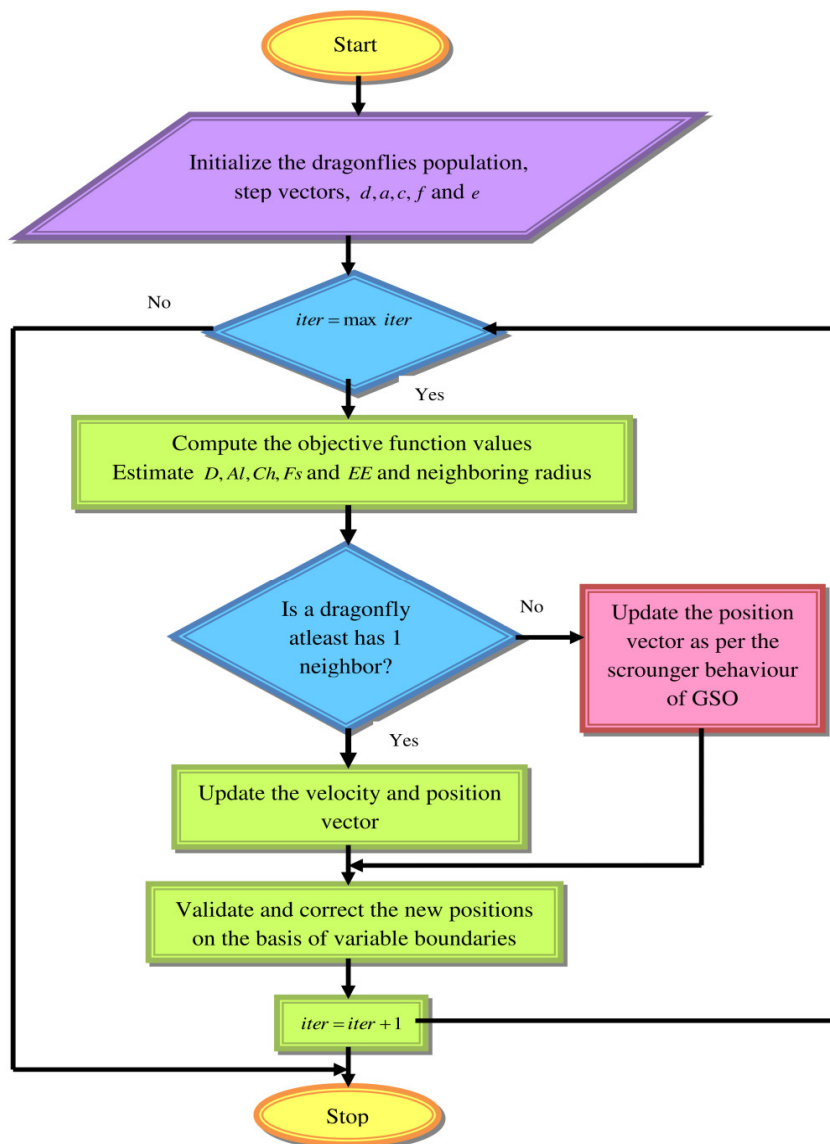


Figure 7. Flowchart of proposed SL-DA algorithm.

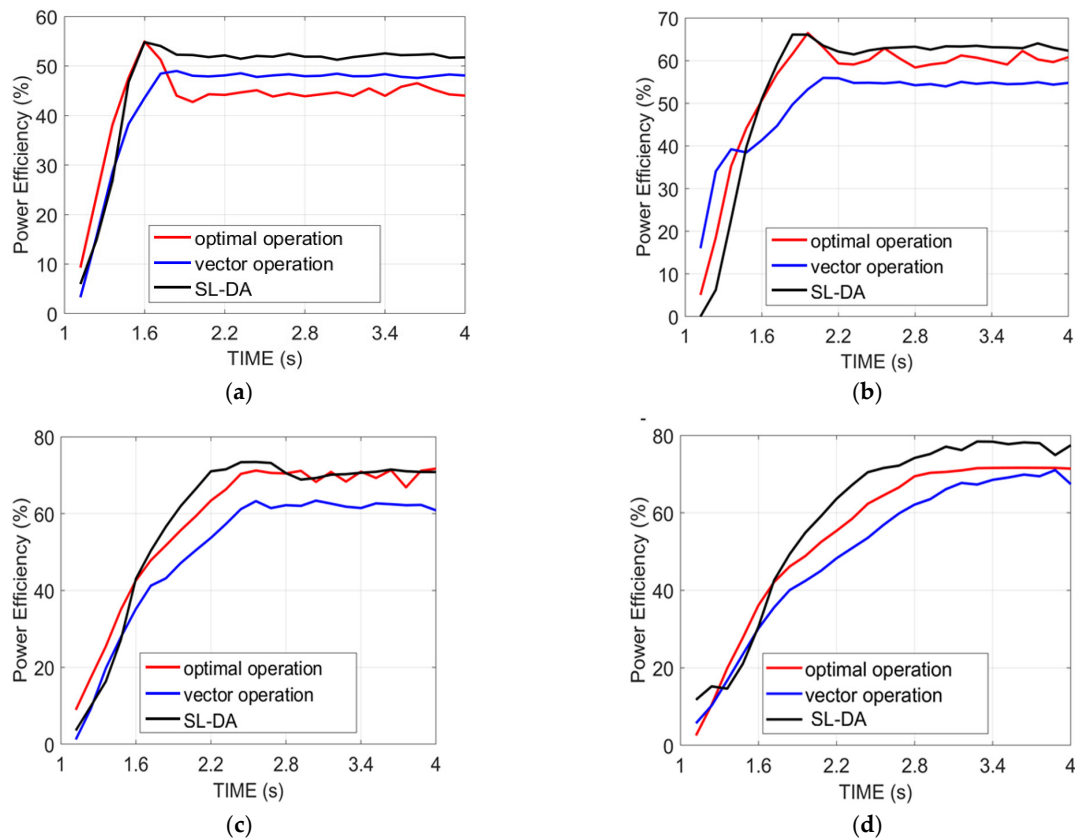
The suggested SL-DA algorithm has the advantage of precise estimation and good accuracy of loss calculation. It supports global and local search capabilities. The convergence rate is better with maximum exploitation capabilities. On the other hand, if we consider sequences of random decisions, then it is not dependent.

## 5. Results and Discussion

In this section, the results of the proposed method are presented, discussed, and analyzed. The simulation work of the proposed model was evaluated using MATLAB 2018a. The performances of the proposed method were examined in terms of power efficiency, speed, and torque under four configurations such as speed 90 and torque 130, speed 120 and torque 150, speed 150 and torque 180, and speed 180 and torque 200. Further, the analysis of the adopted model was exploited over the other conventional models in [31] such as vector operation and optimal operation regarding efficiency, energy-saving, and input power. The analysis between the proposed SL-DA method and PSO algorithm is also presented and an analysis of efficiency, output power, and savings in energy has been presented in different tables.

### 5.1. Analysis on Power Efficiency

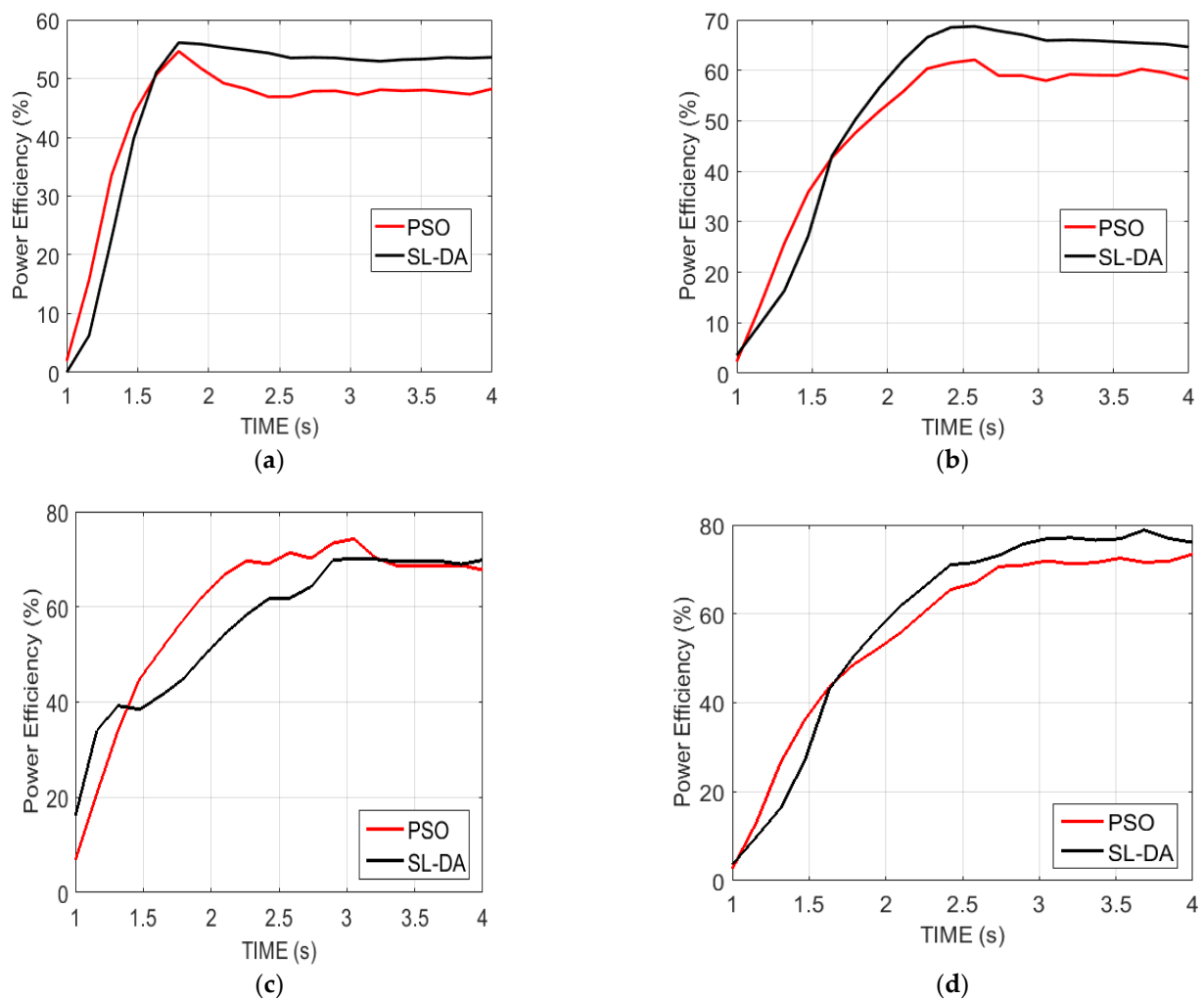
Figure 8 explains the analysis of the power efficiency of the proposed model over conventional models. This analysis is made over four configurations by varying the time. In fact, the power efficiency is on the basis of speed and torque and is assigned to be in maximum for the effective operation of the system. Initially, the efficiency of power is started with a low value and achieves the maximum over the time seconds. On comparing the conventional models in [31] with this proposed work, the results are discussed in the following paragraphs.



**Figure 8.** Analysis on the power efficiency of proposed work over conventional models: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.

Inspecting Figure 8a, the proposed work with speed 90 and torque 130 at time 1.6 has achieved maximum power efficiency, which is 25.77% better than vector operation. Similarly, in Figure 8b, at time 2.2 with speed 120 and torque 150, the implemented model has attained better performance than both optimal operation and vector operation by 9.52% and 4.02%, respectively. Subsequently, the proposed work is compared over other models for other speeds and torques by varying the time, and the resultant outcome has demonstrated the betterment of the proposed work with increased power efficiency.

Another analysis on efficiency has been performed between the proposed SL-DA model with the PSO model to show that the proposed method has high accuracy. The results are depicted in Figure 9, where we can see that exploration of SLDA is high as compared to PSO.



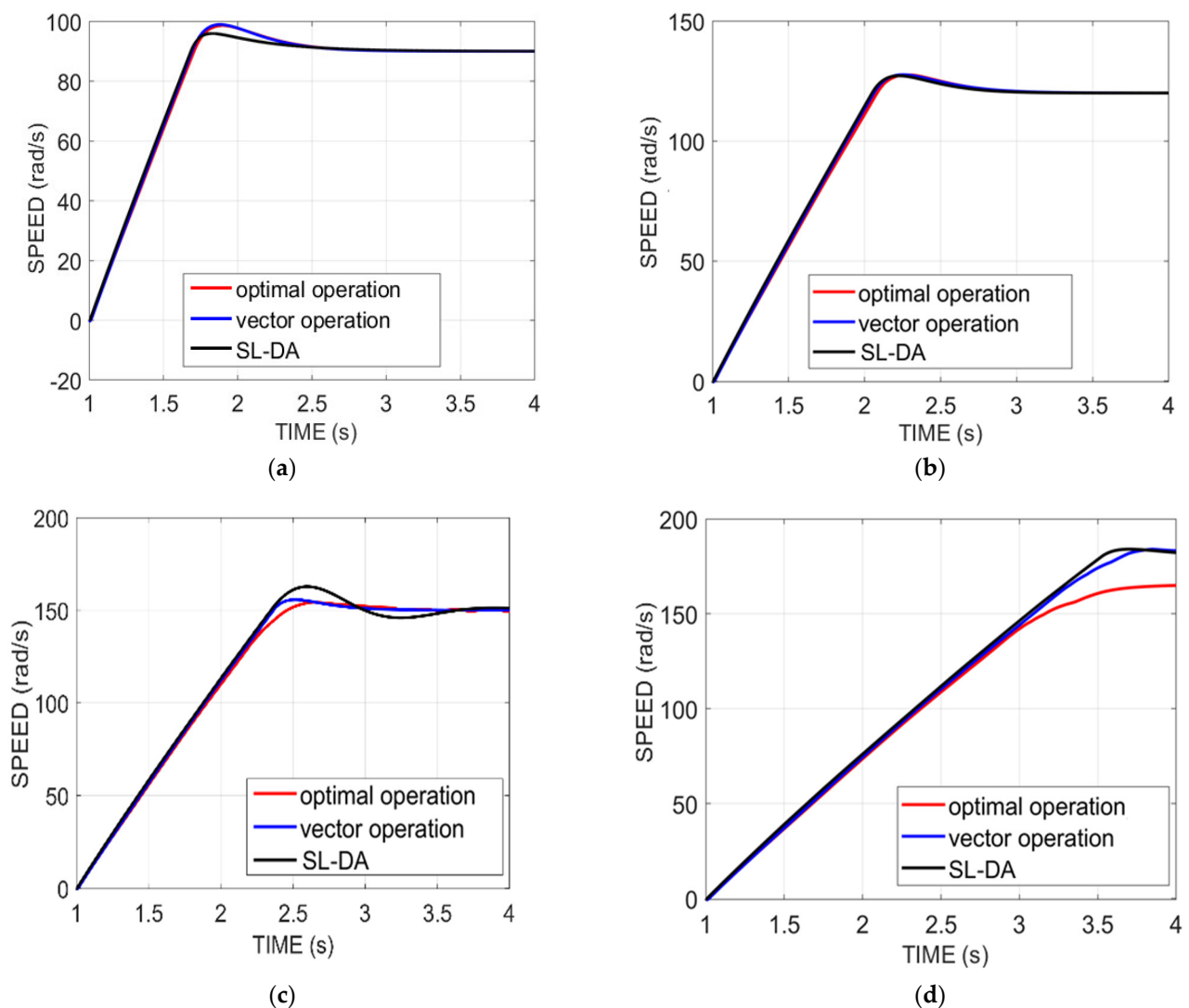
**Figure 9.** Analysis on the power efficiency of the proposed work over PSO: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.

### 5.2. Analysis on Speed

Figure 10 delineates the analysis of the speed of the proposed model against the conventional model under four configurations by varying the time seconds. In this analysis, the speed is set to some value initially. For instance, in Figure 10a, the speed is fixed as 90. The system performance seems to be evaluated efficiently only when the fixed speed is attained on simulation. For instance, on considering the same Figure 10a, the simulation speed attains the fixed speed of 90, by which the system performance is better achieved. Likewise, for the other configurations, the simulation speed of the proposed model is



attained with better value, i.e., fixed value, and thus proves the improved performance of the system with better efficiency.

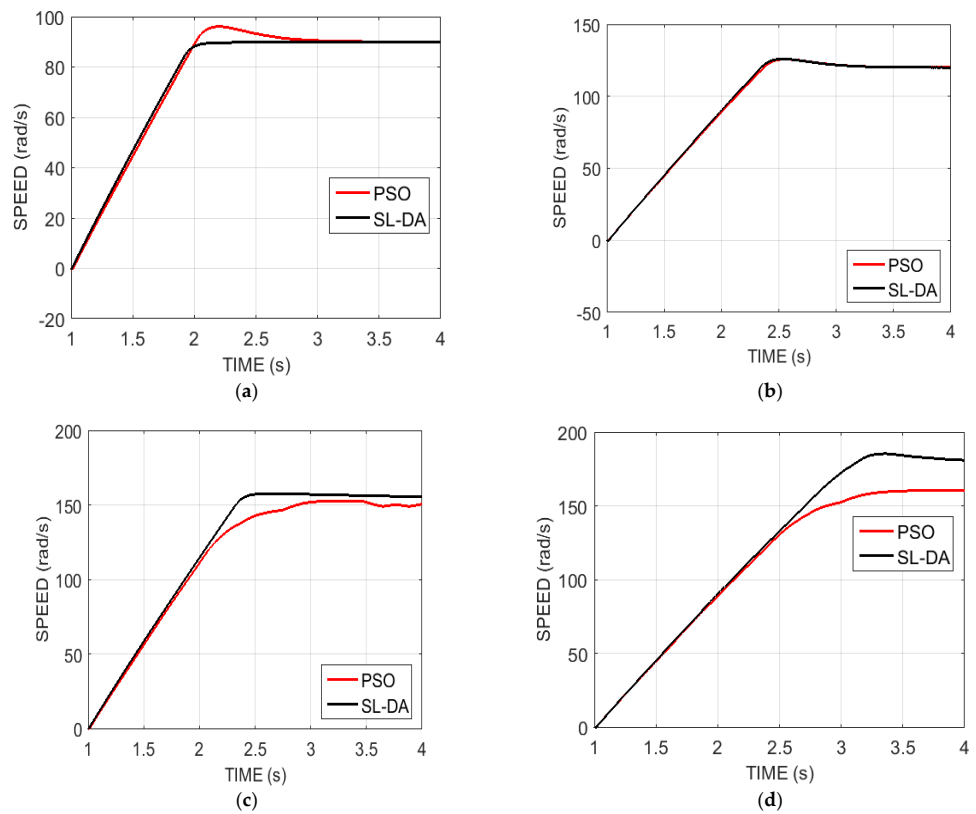


**Figure 10.** Speed analysis of proposed work over conventional models: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.

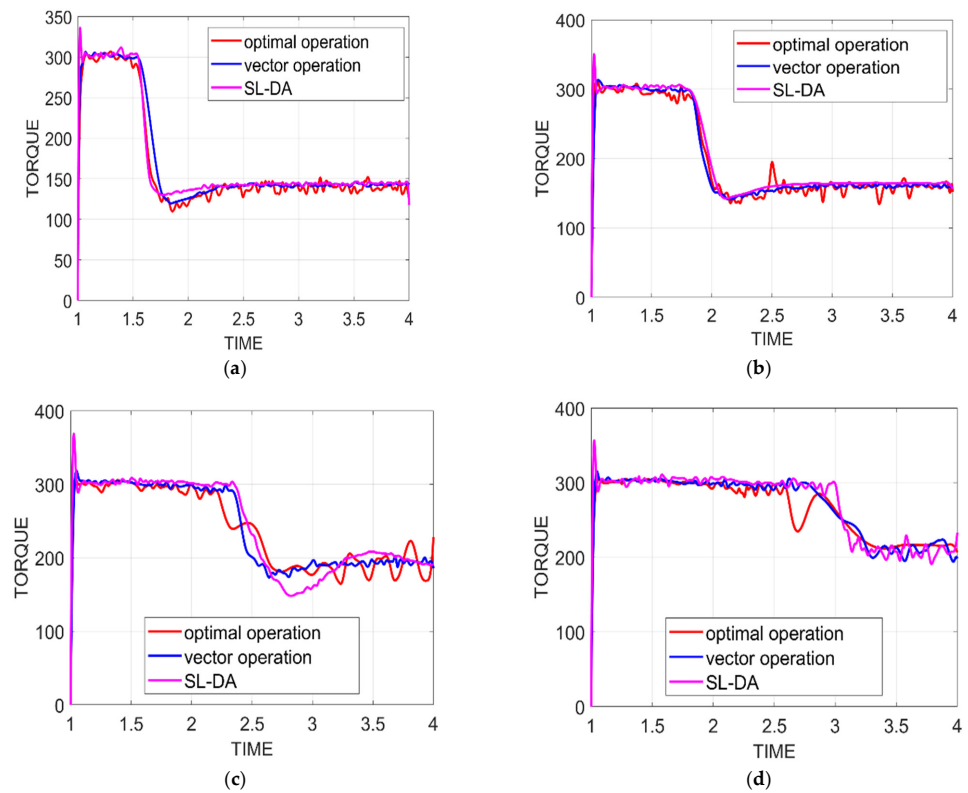
The resulting analysis of the speed graph of the proposed SL-DA algorithm is compared with the PSO and depicted in Figure 11. It can be clearly seen that the SL-DA algorithm reduces the overshoot and attains stability more conveniently as compared to the PSO algorithm.

### 5.3. Analysis on Torque

Figure 12 exhibits the performance analysis of torque of the proposed model against conventional models by varying times under four configurations. In this, the torque is fixed with some value, and if that value is attained on simulation, then the overall system performance is considered to be effective. Initially, the torque value is at maximum and is decreased linearly over time. The results thus obtained in this way are explained as follows.



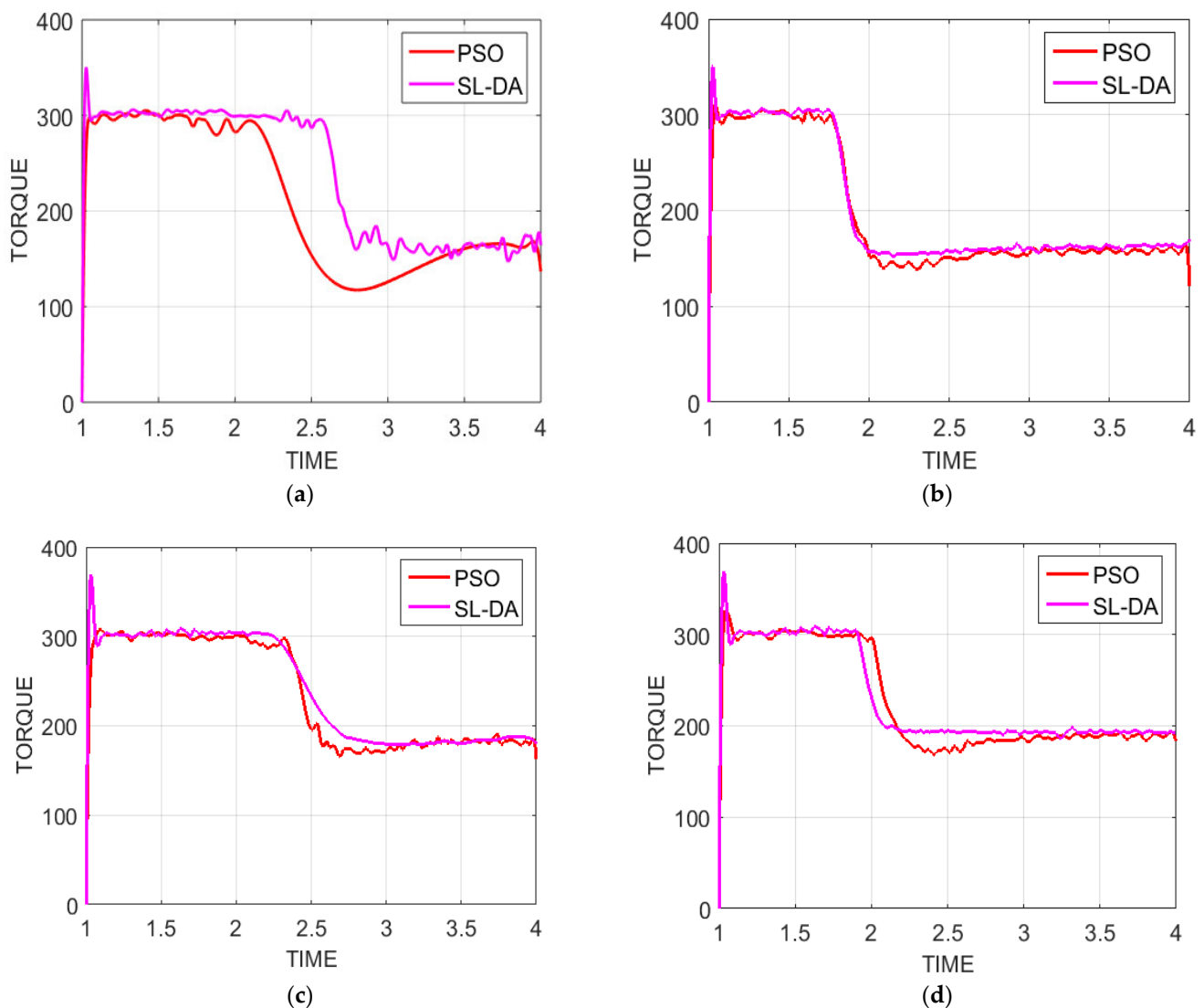
**Figure 11.** Speed analysis of proposed work over conventional models: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.



**Figure 12.** Torque analysis of proposed work over conventional models: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.

As per Figure 12a, the implemented work with speed 90 and torque 130 at time 2.2 has achieved the targeted value and is better than other conventional models' optimal operation and vector operation by 43.82% and 43.97%, respectively. Similarly, from Figure 12d, the proposed model with speed and torque as 150 and 180 at time 3.65 has achieved the same fixed value 150 and that is 18.8% and 20.69%, respectively. The simulation of the implemented model thus validates the improved system performance with better attainment of torque.

Analysis of torque graph under four configurations has been performed between the proposed SL-DA and PSO models. The results can be seen in Figure 13. From the obtained results, we can observe that there is more smoothness in the torque graph, and ripples are slightly minimized. Hence, the proposed SL-DA algorithm shows better results as compared to the PSO algorithm.



**Figure 13.** Torque analysis of proposed work over PSO: (a) speed 90 and torque 130; (b) speed 120 and torque 150; (c) speed 150 and torque 180; and (d) speed 180 and torque 200.

#### 5.4. Analysis on Efficiency

Table 2 describes the efficiency analysis of the proposed work over conventional and PSO models under four configurations. The efficiency of the proposed SL-DA working with speed 90 and torque 130 is achieved with better performance and shows a 13.47%, 8.13%, and 5.29% improvement for optimal operation, vector operation, and PSO model, respectively. Similarly, the implemented method in terms of efficiency with speed 150 and torque

180 has achieved superiority over the optimal operation and vector operation by 0.63%, 13.27%, and 1.34%, respectively. From the overall efficiency analysis, it is demonstrated that the suggested SL-DA model has attained supremacy over the other conventional model as well as PSO model with better efficiency.

**Table 2.** Efficiency analysis of proposed work over state-of-the-art models.

	Efficiency (%)			
	Vector Operation	Optimal Operation	PSO	SL-DA
speed = 90 and torque = 130	45.818	48.08	49.38	51.99
speed = 120 and torque = 150	54.602	60.253	59.208	63.121
speed = 150 and torque = 180	62.174	69.986	69.493	70.426
speed = 180 and torque = 200	67.997	71.273	72.719	77.139

### 5.5. Analysis on Energy Saving

Table 3 explains the energy-saving analysis of the proposed SL-DA model over conventional models and PSO models under four configurations. The energy-saving table is assessed based on the efficiency in Table 2, and the assessment is conducted as per Equation (28). On comparing the overall performance of the model on energy saving, the proposed model seems to attain better performance than the conventional and PSO models. There is an improvement in energy saving of induction motor drive when operated with SL-DA algorithm.

$$\% \text{ energy saving} = \frac{\text{proposed} - \text{conventional}}{\text{conventional}} * 100 \quad (28)$$

**Table 3.** Energy saving analysis of proposed work over state-of-the-art models.

	Comparing Optimal Operation with Proposed SL-DA Work	Comparing Vector Operation with Proposed SL-DA Work	Comparing PSO with Proposed SL-DA Work
speed = 90 and torque = 130	8.1323%	13.471%	5.285%
speed = 120 and torque = 150	4.759%	15.602%	6.608%
speed = 150 and torque = 180	0.6287%	13.273%	1.342%
speed = 180 and torque = 200	8.2303%	13.445%	6.078%

### 5.6. Analysis on Output Power

Table 4 delineates the analysis of the output power of the proposed work over conventional models and PSO under four configurations. The output power obtained by the proposed SL-DA work is increased as compared to the conventional models' optimal operation, vector operation, and PSO algorithm in speed 90 and torque 130 by 13.47%, 8.13%, and 5.29%, respectively. Subsequently, the proposed model with speed 180 and torque 200 has achieved improved performance in output power as compared to the optimal operation, vector operation, and PSO by 13.44%, 8.23%, and 6.07%, respectively. From the simulation, it is observed that the suggested algorithm has more effective performance with increased output power than other conventional models.

**Table 4.** Output power analysis of proposed work over state-of-the-art models.

	Input Power (kw)	Output Power (kw)			
		Vector Operation	Optimal Operation	PSO	SL-DA
speed = 90 and torque = 130	22.273	10.205	10.709	10.998	11.58
speed = 120 and torque = 150	28.292	17.047	15.448	17.713	17.858
speed = 150 and torque = 180	35.149	24.6	21.854	24.426	24.754
speed = 180 and torque = 200	40.118	28.593	27.279	28.956	30.946

### 5.7. Obtained Proportional and Integral Controller Gain Values by Conventional Method and Proposed SL-DA Algorithm

Table 5 defines the values of  $k_p$ ,  $k_i$ , and efficiency for different configurations of speed and torque that were obtained optimally using the conventional vector control, optimal operation, and PSO algorithm along with the proposed SL-DA algorithm by varying the torque and speed. The optimal values obtained by the SL-DA algorithm tune the proportional–integral controller more optimally as compared to other approaches, which gives better performance.

**Table 5.** Obtained proportional and integral values of conventional method and proposed SL-DA algorithm.

	Vector Control			Optimal Operation			PSO Algorithm			SL-DA Algorithm		
	$k_p$	$k_i$	Efficiency	$k_p$	$k_i$	Efficiency	$k_p$	$k_i$	Efficiency	$k_p$	$k_i$	Efficiency
speed = 90 and torque = 130	20.24	32.21	45.818	20.61	33.20	48.08	19.8	34.41	49.38	21.68	34.76	51.990
speed = 120 and torque = 150	12.10	28.09	54.602	13.12	28.71	60.253	19.22	29.38	59.208	13.70	30.60	63.121
speed = 150 and torque = 180	13	26	62.174	13.71	27.33	69.986	14.09	25.94	69.493	14.33	26.21	70.426
speed = 180 and torque = 200	2.06	12.03	67.997	2.93	14.16	71.273	3.15	14.77	72.179	3.37	15.96	77.139

The simulation results of the conventional vector model, optimal operation, PSO algorithm, and proposed SL-DA optimization method are presented in this paper, and a comparative analysis among all the methods has been performed. On comparing the simulated results and the data presented in the tables, it is observed that the SL-DA optimization concept has shown superiority over other conventional methods as well as on the PSO algorithm. The proposed SL-DA concept has shown improvement in the performance of induction motor drive in terms of efficiency and energy-saving with an increase in output power. The proposed SL-DA method provides a fine-tuning of the PI controller, which improves the results of induction motor drive.

## 6. Conclusions

This research work proposed a new contribution for maximizing induction motor efficiency. The objective model was obtained from the speed-control induction motor drive, where the PI controller was tuned. Further, the PI controller's gains such as proportional gain and integral gain were optimally tuned by deploying a new hybridized algorithm named SL-DA, which incorporates the concept of DA with GSO. The proposed SL-DA results are tested and compared with the PSO algorithm for verifications. It is observed that the exploration capability of SL-DA is higher than PSO. As a result, the SL-DA algorithm outperformed PSO and other conventional methods. Finally, the performance of the suggested model was distinguished over other models in terms of controlling the analysis of torque and speed, the efficiency and energy-saving analysis of drive, and analysis on improvement in output power, and thus validates the superiority of the proposed

work. For future work, other meta-heuristic algorithms can be implemented to tune the proportional–integral controller for minimizing loss, enhancing efficiency, and evaluating the performance of induction motor drive.

**Author Contributions:** Conceptualization, N.K.S.; Methodology, N.K.S.; Investigation, R.S.; Project administration, S.M.; Software, R.S.; Supervision, S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors are thankful to the anonymous reviewers for reading the manuscript carefully and for their insightful comments and suggestions.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Description
IM	Induction Motor
DC	Direct Current
MBO	Model Based Optimization
SAO	Search Algorithm based Optimization
SVM	Space Vector Modulation
LMA	Levenberg–Marquardt algorithm
DTC	Direct Torque Control
MIMO	Multiple-Input Multiple-Output
ACO	Ant Colony Optimization
DE	Differential Evolution
PI	Proportional-Integral
FLC	Fuzzy Logic Controller
PSO	Particle Swarm Optimization
NA	Newton Algorithm
GNA	Gauss–Newton Algorithm
SDA	Steepest Descent Algorithm
IVC	Indirect Vector Control
SVPWM	Space Vector Pulse Width Modulation

## References

1. Glowacz, A. Acoustic based fault diagnosis of three-phase induction motor. *Appl. Acoust.* **2018**, *137*, 82–89. [[CrossRef](#)]
2. Werner, U. Vibration control of large induction motors using actuators between motor feet and steel frame foundation. *Mech. Syst. Signal Process.* **2018**, *112*, 319–342. [[CrossRef](#)]
3. Ghosh, E.; Mollaieian, A.; Kim, S.; Tjong, J.; Kar, N.C. DNN-Based Predictive Magnetic Flux Reference for Harmonic Compensation Control in Magnetically Unbalanced Induction Motor. *IEEE Trans. Magn.* **2017**, *53*, 1–7. [[CrossRef](#)]
4. Castaldi, P.; Tilli, A. Parameter estimation of induction motor at standstill with magnetic flux monitoring. *IEEE Trans. Control Syst. Technol.* **2005**, *13*, 386–400. [[CrossRef](#)]
5. Liu, C.; Chau, K.T.; Li, W.; Yu, C. Efficiency Optimization of a Permanent-Magnet Hybrid Brushless Machine Using DC Field Current Control. *IEEE Trans. Magn.* **2009**, *45*, 4652–4655.
6. Meng, L.; Dragicevic, T.; Vasquez, J.C.; Guerrero, J.M. Tertiary and Secondary Control Levels for Efficiency Optimization and System Damping in Droop Controlled DC–DC Converters. *IEEE Trans. Smart Grid* **2015**, *6*, 2615–2626. [[CrossRef](#)]
7. Nguyen-Trong, N.; Fumeaux, C. Tuning Range and Efficiency Optimization of a Frequency-Reconfigurable Patch Antenna. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 150–154. [[CrossRef](#)]
8. Bieler, G.; Marcelo, M.W. A magnetostrictive-fiber Bragg grating sensor for induction motor health monitoring. *Measurement* **2018**, *122*, 117–127. [[CrossRef](#)]

9. Adekitan, A.; Samuel, I.; Amuta, I.; AmutaElizabeth, E. Dataset on the performance of a three phase induction motor under balanced and unbalanced supply voltage conditions. *Data Brief* **2019**, *24*, 103947. [[CrossRef](#)]
10. Li, X.; Li, M. Multiobjective Local Search Algorithm-Based Decomposition for Multiobjective Permutation Flow Shop Scheduling Problem. *IEEE Trans. Eng. Manag.* **2015**, *62*, 544–557. [[CrossRef](#)]
11. Chen, B.; Zeng, W.; Lin, Y.; Zhang, D. A New Local Search-Based Multiobjective Optimization Algorithm. *IEEE Trans. Evol. Comput.* **2015**, *19*, 50–73. [[CrossRef](#)]
12. Bechikh, S.; Chaabani, A.; Ben Said, L. An Efficient Chemical Reaction Optimization Algorithm for Multiobjective Optimization. *IEEE Trans. Cybern.* **2015**, *45*, 2051–2064. [[CrossRef](#)] [[PubMed](#)]
13. Naji, H.R.; Sohrabi, M.; Rashedi, E. A High-Speed, Performance-Optimization Algorithm Based on a Gravitational Approach. *Comput. Sci. Eng.* **2012**, *14*, 56–62. [[CrossRef](#)]
14. Tamrakar, U.; Shrestha, D.; Malla, N.; Ni, Z.; Hansen, T.M.; Tamrakar, I.; Tonkoski, R. Comparative Analysis of Current Control Techniques to Support Virtual Inertia Applications. *Appl. Sci.* **2018**, *8*, 2695. [[CrossRef](#)]
15. Yetgin, A.G. Effects of induction motor end ring faults on motor performance. Experimental results. *Eng. Fail. Anal.* **2019**, *96*, 374–383. [[CrossRef](#)]
16. Bortoni, E.C.; Bernardes, J.V.; da Silva, P.V.V.; Faria, V.A.D.; Vieira, P.A.V. Evaluation of manufacturers strategies to obtain high-efficient induction motors. *Sustain. Energy Technol. Assess.* **2019**, *31*, 221–227. [[CrossRef](#)]
17. Ma, H. Model-Based Multiobjective Evolutionary Algorithm Optimization for HCCI Engines. *IEEE Trans. Veh. Technol.* **2015**, *64*, 4326–4331. [[CrossRef](#)]
18. Bujok, P.; Zamuda, A.; Das, S.; Suganthan, P.; Panigrahi, B. Cooperative Model of Evolutionary Algorithms and Real-World Problems. In *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*; SEMCCO 2019, FANCCO 2019. Communications in Computer and Information Science; Springer: Cham, Switzerland, 2020; Volume 1092. [[CrossRef](#)]
19. Al-Betar, M.A.; Khader, A.T.; Zaman, M. University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm. *IEEE Trans. Syst. Man Cybern. (Appl. Rev.)* **2012**, *42*, 664–681. [[CrossRef](#)]
20. Ammar, A.; Benakcha, A.; Bourek, A. Closed loop torque SVM-DTC based on robust super twisting speed controller for induction motor drive with efficiency optimization. *Int. J. Hydrogen Energy* **2017**, *42*, 17940–17952. [[CrossRef](#)]
21. Farhani, F.; Regaya, C.B.; Zaafour, A.; Chaari, A. Real time PI-backstepping induction machine drive with efficiency optimization. *ISA Trans.* **2017**, *70*, 348–356. [[CrossRef](#)]
22. Farhani, F.; Regaya, C.B.; Zaafour, A.; Chaari, A. Real time induction motor efficiency optimization. *J. Frankl. Inst.* **2017**, *354*, 3289–3304. [[CrossRef](#)]
23. Taheri, A.; Rahmati, A.; Kaboli, S. Efficiency Improvement in DTC of Six-Phase Induction Machine by Adaptive Gradient Descent of Flux. *IEEE Trans. Power Electron.* **2012**, *27*, 1552–1562. [[CrossRef](#)]
24. Kong, W.; Qu, R.; Kang, M.; Huang, J.; Jing, L. Air-Gap and Yoke Flux Density Optimization for Multiphase Induction Motor Based on Novel Harmonic Current Injection Method. *IEEE Trans. Ind. Appl.* **2017**, *53*, 2140–2148. [[CrossRef](#)]
25. Laamari, Y.; Chafaa, K.; Athamena, B. Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive. *Electr. Eng.* **2015**, *97*, 129–138. [[CrossRef](#)]
26. Zeb, K.; Uddin, W.; Haider, A.; Belal, S.; Mehmood, C.A.; Khan, M.A.; Kim, H.J. Robust speed regulation of indirect vector control induction motor using fuzzy logic controllers based on optimization algorithms. *Electr. Eng.* **2018**, *100*, 787–802. [[CrossRef](#)]
27. Costa, B.L.G.; Graciola, C.L.; Angélico, B.A.; Goedtel, A.; Castoldi, M.F. Metaheuristics optimization allied to PI controllers tuning of a DTC-SVM drive for three-phase induction motors. *Appl. Soft Comput.* **2018**, *62*, 776–788. [[CrossRef](#)]
28. Jafari, M.; Chaleshtari, M.H.B. Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out. *Eur. J. Mech. A/Solids* **2017**, *66*, 1–14. [[CrossRef](#)]
29. He, S.; Wu, Q.H.; Saunders, J.R. Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. *IEEE Trans. Evol. Comput.* **2009**, *13*, 973–990. [[CrossRef](#)]
30. Wang, L.; Shi, R.; Dong, J. A Hybridization of Dragonfly Algorithm Optimization and Angle Modulation Mechanism for 0–1 Knapsack Problems. *Entropy* **2021**, *23*, 598. [[CrossRef](#)]
31. Choudhary, P.K.; Dubey, S.P.; Gupta, V.K. Efficiency optimization of induction motor drive at steady-state condition. In Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, India, 18–19 December 2015; pp. 470–475.
32. Shukla, N.K.; Pandey, S.K.; Srivastava, R. Efficiency and Cost Optimization of Three- $\phi$  Induction Motor Using Soft Computing Techniques. In *Computing Algorithms with Applications in Engineering, Algorithms for Intelligent Systems*; Springer: Singapore, 2020; pp. 305–318. [[CrossRef](#)]
33. Idi, M.; Ahmed, M.; Halilu, B.D.; Sadiq, A.A. Proportional Integral (PI) Controller for a Process Plant System. *Int. J. Eng. Res. Technol. (IJERT)* **2017**, *4*, 229–232.
34. Ang, K.H.; Gregory, C.; Li, Y. PID control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **2015**, *1*, 559–576.
35. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
36. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]

37. Rahman, C.M.; Rashid, T.A.; Alsadoon, A. A survey on dragonfly algorithm and its applications in engineering. *Evol. Intell.* **2021**, *8*, 1–21. [[CrossRef](#)]
38. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
39. Emambocus, B.A.S.; Jasser, M.B.; Mustapha, A.; Amphawan, A. Dragonfly Algorithm and Its Hybrids: A Survey on Performance, Objectives and Applications. *Sensors* **2021**, *21*, 7542. [[CrossRef](#)] [[PubMed](#)]