

Speed controller design for three-phase induction motor based on dynamic adjustment grasshopper optimization algorithm

Ammar Falah Algamluoli, Nizar Hadi Abbas

Department of Electrical Engineering, College of Engineering, University of Baghdad, Iraq

Article Info

Article history:

Received Jun 19, 2020

Revised Jul 26, 2020

Accepted Sep 22, 2020

Keywords:

CPIL controller

Dynamic adjustment GOA

Stator direct current

Stator quadrature current

standard GOA

Three-phase induction motor
direct on line

ABSTRACT

Three-phase induction motor (TIM) is widely used in industrial application like paper mills, water treatment and sewage plants in the urban area. In these applications, the speed of TIM is very important that should be not varying with applied load torque. In this study, direct on line (DOL) motor starting without controller is modelled to evaluate the motor response when connected directly to main supply. Conventional PI controller for stator direct current and stator quadrature current of induction motor are designed as an inner loop controller as well as a second conventional PI controller is designed in the outer loop for controlling the TIM speed. Proposed combined PI-lead (CPIL) controllers for inner and outer loops are designed to improve the overall performance of the TIM as compared with the conventional controller. In this paper, dynamic adjustment grasshopper optimization algorithm (DAGO) is proposed for tuning the proposed controller of the system. Numerical results based on well-selected test function demonstrate that DAGO has a better performance in terms of speed of convergence, solution accuracy and reliability than SGOA. The study results revealed that the currents and speed of TIM system using CPIL-DAGO are faster than system using conventional PI and CPIL controllers tuned by SGOA. Moreover, the speed controller of TIM system with CPIL controlling scheme based on DAGO reached the steady state faster than others when applied load torque.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nizar Hadi Abbas

Department of Electrical Engineering

College of Engineering, University of Baghdad

Aljadriya district, Baghdad, Iraq

Email: dr.nizar.hadi@coeng.uobaghdad.edu.iq

1. INTRODUCTION

Some applications like paper mills, water treatment plants and several other industrial factories are used three-phase induction motor (TIM) because it is simple, rugged and low maintenance [1, 2]. Controlling method of induction motor is very difficult due to complexity and nonlinearity of the system [3-5]. Different load torque may apply to the system, and that would effect on speed performance. Therefore, speed of induction motor has to be controlled in order to let the system work with high performance at different load torque [3].

PI controller has been widely used in industrial applications due to its structural simplicity and the ability to solve practical control problems [6]. There are several existing works revealing that the use of PI controller will lead to poor performances as well as causing instability for the controlled system. Several researches have been done on the speed controller of TIM. In [1] particle swarm optimization (PSO) was used for tuning PI speed controller of TIM. Universal developed drive software of a speed controller of single

or TIM was carried out in [2]. Speed controller of TIM using fuzzy logic design and PI controller with a bacterial foraging optimization (BFO) algorithm was studied in [3]. Comparisons between conventional PI and slide mode controllers for speed of three-phase induction motor were investigated in [4].

Speed controller of TIM using PID controller with tuning methods was achieved in [5]. Designing of PID controller using indirect field oriented control (IFOC) for TIM was done in [7]. Speed controller of dual three-phase induction motor using PI conventional controller and IP controller were carried out in [8]. PI compensator technique of three-phase induction motor was used in [9] as well as PI and fuzzy logic controller (FLC) for direct flux and torque ripple of TIM were investigated in [10]. Voltage to frequency control based on using PI controller for TIM was done in [11]. In [12] Modified PID-fuzzy controller was made for speed controller of TIM. Variable frequency drive (VFC) using fuzzy logic controller was carried out for speed control of TIM in [13]. The authors in [14] utilized PID controller for direct torque controller (DTC) of TIM and the results were compared with torque at starting. In [15] genetic algorithm for tuning PI controller was used for speed controlling of three-phase induction motor. Optimal control with neural network algorithm was proposed for speed control of three-phase induction motor in [16].

In this paper, the TIM is modelled and implemented using matlab/simulink environment in α and β coordinate. Conventional PI controller is designed for stator direct current (I_{sd}) and stator quadrature current (I_{sq}) currents of the system. Additionally, PI controller is designed for the speed of the system. In order to enhance the performance of the system, the combined PI-lead (CPIL) controller is proposed. The proposed CPIL controllers for inner and outer loops are tuned using the standard grasshopper optimization algorithm (SGOA) that proved is an efficient algorithm than genetic algorithm (GA) and PSO algorithm for solving complex optimization problems. The SGOA is one of the newest optimization algorithms invented by Saremi *et al.*, in 2017 [17]. This nature-inspired algorithm is developed to maximize the searching capability and the convergence rate, and the generated version is called dynamic adjustment GOA (DAGOA). The standard and developed algorithms are applied for tuning the controllers' parameters of the current and speed of TIM. The impact of disturbance on system performances when load torque is imposed, which not covered well in the literature are studied. The main objectives of this work include:

- a. Design speed controller for TIM using conventional PI controlling method.
- b. Design the system based on the proposed controlling scheme.
- c. Optimize the controllers' parameters using the SGOA and the proposed DAGOA techniques.
- d. Performance comparison between the optimization algorithms to evaluate the effectiveness of the proposed one using a set of test functions.
- e. A broad comparison has been accomplished between optimized conventional PI and CPIL controllers; the performance of each algorithm will be analysed and evaluated according to the objective function.

This paper is arranged as follows: The section 2 provides a TIM system modelling and its implementation in Matlab/Simulink. Section 3 describes the controlling techniques; conventional PI and the proposed controllers that employed in a closed-loop design with the selected performance index. Section 4 presents the theoretical basics for the SGOA algorithm and the details of the proposed algorithm. Section 5 shows the numerical results for evaluating the proposed DAGOA performances as compared with the standard algorithm based on ten test functions as well as presents the simulation results for the controllers tuned by optimization algorithms, and the comparative analysis have been done to verify the capability of the proposed controlling method. Finally, the conclusions are summarized in the last section.

2. SYSTEM DYNAMIC MODEL AND DESCRIPTION

Induction motor consists from two parts constant part, which is stator, and moving part, which is rotor. Matlab/simulink environment is used to implement of three-phase induction motor. The model is built using the stator and rotor equations in alpha and beta axis [18], in order to designing a controlling method for the current and speed of the system in simple way. Three phase induction motor in alpha (α) and beta (β) equations are shown below:

$$V_{s\beta} = I_{s\beta} R_s + \frac{d\psi_{s\beta}}{dt} \quad (1)$$

$$V_{s\alpha} = I_{s\alpha} R_s + \frac{d\psi_{s\alpha}}{dt} \quad (2)$$

$$\psi_{s\beta} = \int (V_{s\beta} - I_{s\beta} R_s) dt \quad (3)$$

$$\psi_{s\alpha} = \int (V_{s\alpha} - I_{s\alpha} R_s) dt \quad (4)$$

$$\psi_{s\beta} = L_s I_{s\beta} + L_o i_{r\beta} \tag{5}$$

$$\psi_{s\alpha} = L_s I_{s\alpha} + L_o i_{r\alpha} \tag{6}$$

$$\psi_{r\beta} = L_o I_{s\beta} + L_r i_{r\beta} \tag{7}$$

$$\psi_{r\alpha} = L_o I_{s\alpha} + L_r i_{r\alpha} \tag{8}$$

$$\psi_{r\beta} = \frac{L_r}{L_o} (\int V_{s\beta} - R_s I_{s\beta}) dt - \sigma L_s I_{s\beta} \tag{9}$$

$$\psi_{r\alpha} = \frac{L_r}{L_o} (\int V_{s\alpha} - R_s I_{s\alpha}) dt - \sigma L_s I_{s\alpha} \tag{10}$$

Where $V_{s\alpha}, V_{s\beta}, I_{s\alpha}$, and $I_{s\beta}$ are stator voltage and current in α - β axis respectively; $\psi_{s\alpha}$ and $\psi_{s\beta}$ are stator flux in α - β axis; R_s is stator resistance; $\psi_{r\alpha}$ and $\psi_{r\beta}$ are rotor flux in α - β axis; L_o, L_s , and L_r are magnetising inductance, stator inductance and rotor inductance respectively; σ is leakage coefficient; $i_{r\alpha}$ and $i_{r\beta}$ are rotor current in α - β axis.

Three-phase induction motor equations from (1) to (10) are used to build the system in Matlab/Simulink as shown in Figure 1. In this paper, the system parameters are illustrated in Table 1. From Table 1, it can be seen that the starting current of induction motor is about 250 A, speed is 1475 rpm and torque is about 160 N.M. The purpose of making speed controller for the TIM is to let the system work with same speed 1500 rpm under various load torques.

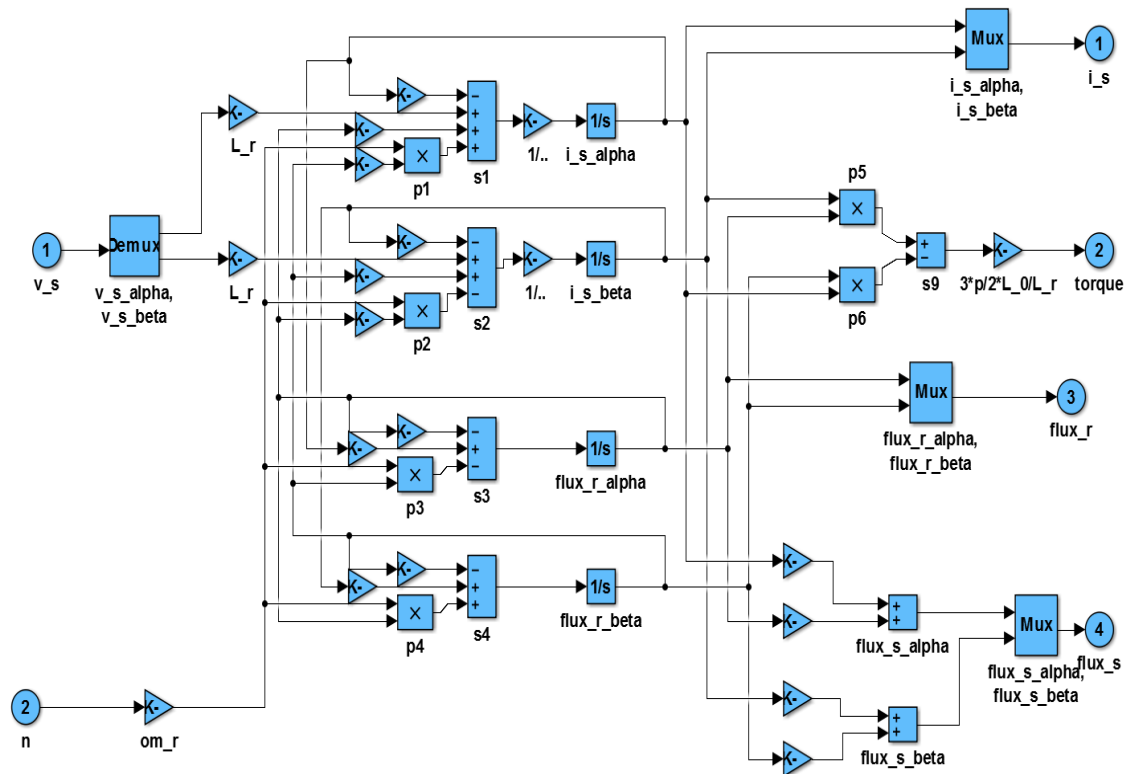


Figure 1. Modelling of three-phase induction motor using Matlab/Simulink in α - β axis

Three-phase induction motor is modelled using Matlab/Simulink block in α - β axis as shown in Figure 1. TIM is connected as direct on line (DOL) to three-phase main supply 415 v to check system behaviour as depicted in Figure 2. In DOL connection the three-phase supply (A, B, C) is converted to alpha-beta axis using Clark transformation as illustrated in Figure 3.

Table 1. Three-phase induction motor parameters

Parameter	Value [Unit]
Power	22 kW
Input voltage	415 V
Frequency	50 Hz
Speed	1475 RPM
Full load current	42 A
Starting current	250 A
Torque	160 N.M
Stator resistance (R_s)	0.6 Ohm
Mutual inductance (L_o)	106.755×10^{-3} H
Rotor inductance (L_r)	109.815×10^{-3} H
Stator inductance (s)	111.368×10^{-3} H
Moment of inertia (J)	0.1 Kg.m ²
Number of poles	4

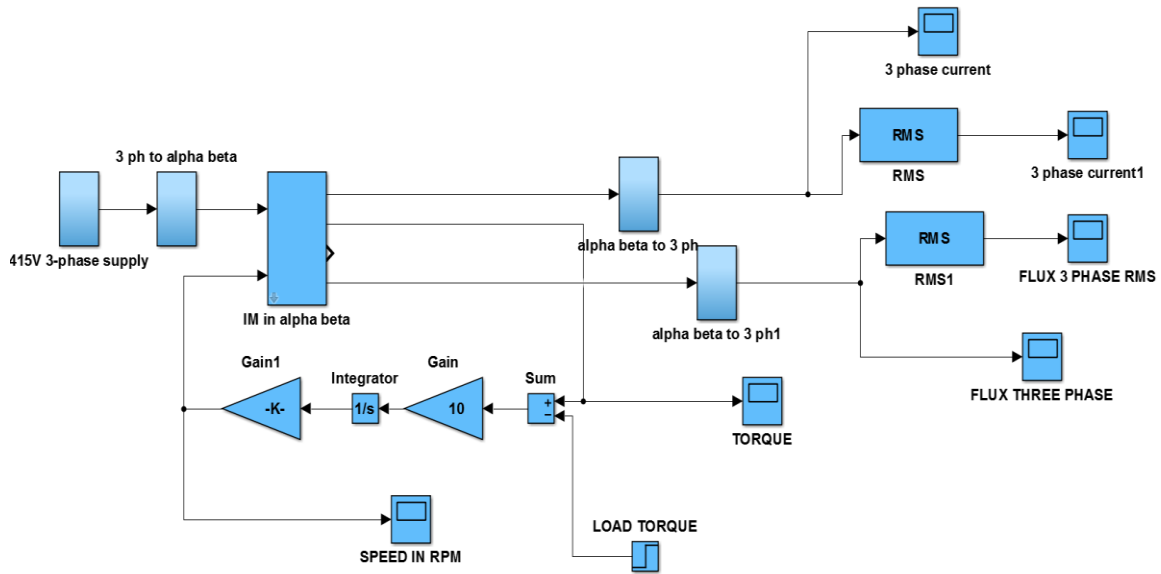


Figure 2. Three phase induction motor modelling in Matlab/Simulink in alpha beta with DOL

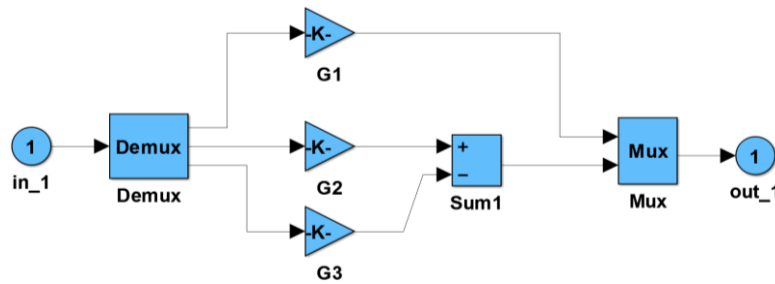


Figure 3. Simulink block of Clark transformation from A, B and C to α and β coordinates

The loop from mechanical torque of the system and add it with load torque to produce speed in RPM using following equation.

$$J \frac{dw}{dt} + T_L = T_{mech} \tag{11}$$

The mechanical (11) of the system is represented in Simulink block as depicted in Figure 4. The stator current, motor torque and motor speed responses for TIM without controller and without load torque are demonstrated in Figure 5. Additionally, the responses for the same variables of TIM but with applying load torque are depicted in Figure 6.

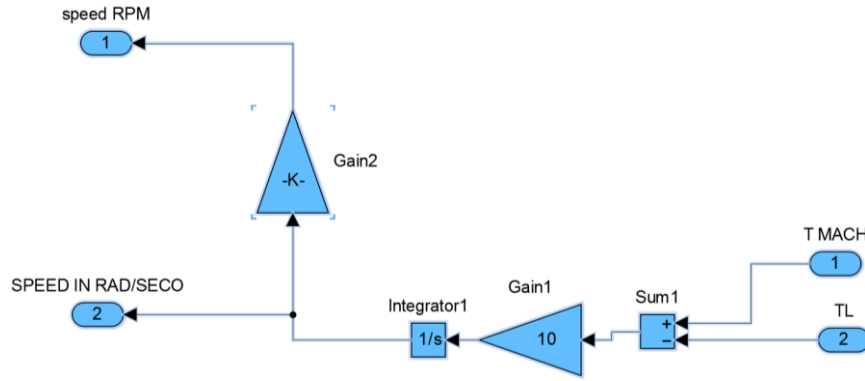


Figure 4. Simulink block for the speed equation of the system

In Figure 5, it shows that starting current is above 200 A at transient with oscillation and motor starting torque is above 500 N.M during transient, while motor speed is 1500 RPM but takes 0.45 s to reach the steady state with high oscillation during transient. Figure 6, it shows that three-phase induction motor speed is reduced to 1452 RPM with oscillation when applied load torque to a system at 0.5 s.

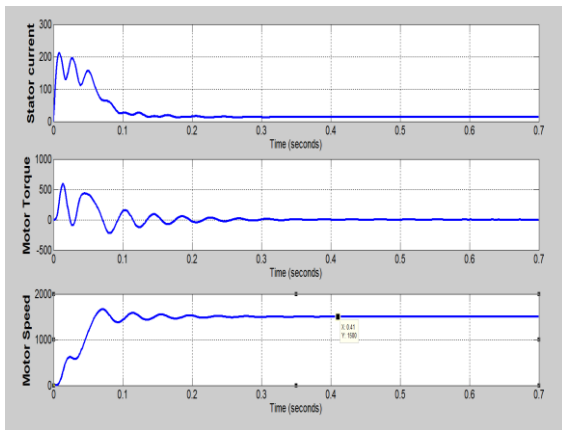


Figure 5. Stator current, motor torque and motor speed without controller without load torque

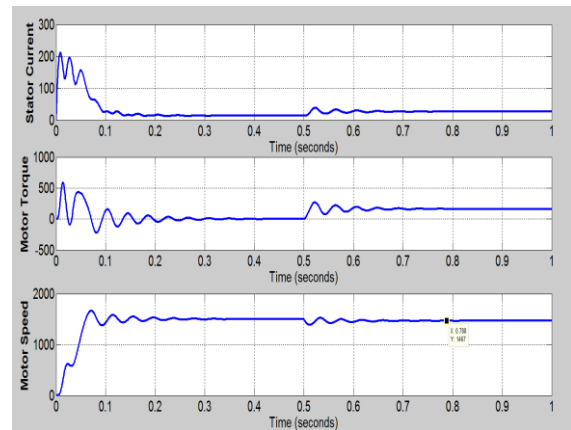


Figure 6. Stator current, motor torque and motor speed without controller with load torque

3. CONTROLLING TECHNIQUES

3.1. Conventional PI controller

The proportional plus integral controllers are the dominating control action for a wide range of industrial processes. The mathematical representation of PI control signal is as follows:

$$G_{PI}(S) = K_P \left(1 + \frac{1}{T_i S} \right) \tag{12}$$

where K_P and T_i are the proportional gain and the integral time constant, respectively. The design objective is to obtain optimal values for K_P and T_i such that an objective function with desired specifications is minimized. For this target, the system is excited by a reference signal, and the system response is obtained in terms of a number of parameters. In this paper, the excitation signal is a unit step with the transient response being characterized by the settling time [19].

Conventional PI controller is designed for I_{sd} and I_{sq} currents. The plant that used to design such controller is derived from (14) and (15) below. Characteristic equation is used to determine the PI controller parameters, which are K_P and T_i . This design depend on natural frequency (w_n) and damping ratio (ζ) of the characteristic equation formulated in (13). Current controller of the system should have very fast response because the system plant have resistance and reactance, which identify the electrical circuit time constant. In addition, the parameters w_n and ζ are chosen to let the system work with a powerful response. Moreover, the

controller depends on resistance and reactance Therefore; the current controller should be faster than speed controller, which depends on moment of inertia (J) and friction. The value of w_n and ζ for I_{sd} and I_{sq} currents controller are chosen 200 HZ and 0.7 respectively.

$$S^2 + 2 \zeta w_n S + w_n^2 = 0 \tag{13}$$

$$V_{sd} = I_{sd} R_s + \sigma L_s \frac{dI_{sd}}{dt} \tag{14}$$

$$V_{sq} = I_{sq} R_s + \sigma L_s \frac{dI_{sq}}{dt} \tag{15}$$

In (14) and (15) are used to find the transfer function of the plant. It can be seen from equations have same resistance R_s , reactance L_s and leakage coefficients σ . Therefore, the system plant are same for both I_{sd} and I_{sq} . Therefore, PI controllers have same parameters for I_{sd} and I_{sq} currents. PI controller parameters are $K_p= 75$ and $T_i =0.0746$. The T_s for the PI controller is time constant which is equal to $T_s = \frac{\sigma L_s}{R_s}$.

Figure 7 illustrate the desigof three-phase induction motor with current controller using PI controller. The value of I_{sd} is 12.3 A and I_{sq} is 20.55 A. It shows that I_{sd}^* and I_{sq}^* references are subtracted from I_{sd} and I_{sq} feedback from system after converting $I_{s\alpha}$ and $I_{s\beta}$ to I_{sd} and I_{sq} using park transformation in block Simulink illustrated in Figure 8.

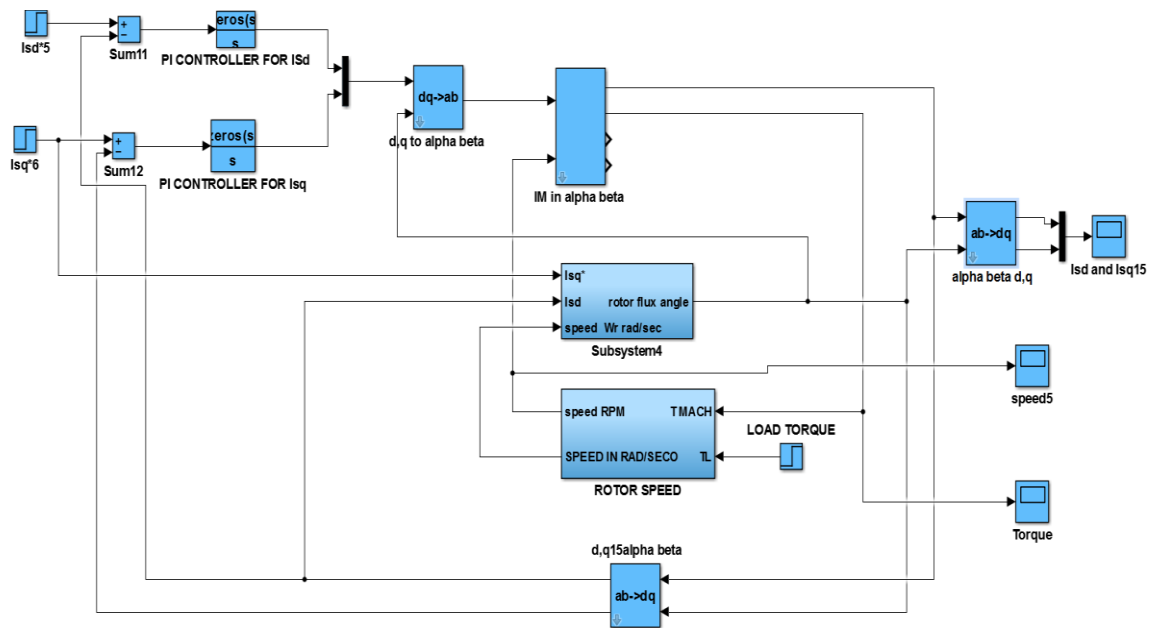


Figure 7. Simulink block for I_{sd} and I_{sq} controlling of TIM with conventional PI controller

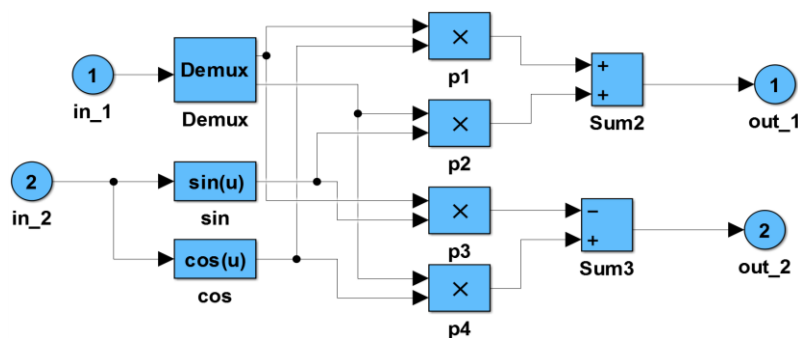


Figure 8. Simulink block to convert α - β coordinates to direct d and quadrature q axis

The response of TIM in terms of I_{sq} currents with PI controller is demonstrated in Figure 9. The stator quadrature current of the system reached steady state at 0.006 s after designing controller using characteristic equation with natural frequency 200 Hz and $\zeta = 0.7$. The speed controller for the system is designed in the same manner followed in the design of the stator direct and quadrature currents controllers. Motor speed depends on the mechanical components, which are motor inertia J and friction B_p . For simplicity, the friction is chosen to be zero in this research work. Moreover, the speed loop frequency is chosen to be much lower the current loop frequency, which is because the motor speed needs more time to change from one speed to another, i.e. it is not practical to change motor speed within milliseconds. Therefore, (13) can be used to find PI controller parameters with natural frequency $\omega_n = 2$ Hz and $\zeta = 0.7$.

From Figure 10, it can be seen that speed of induction motor with conventional PI controller arrived steady state at 0.85 s, this is because of determination PI controller parameters and values of natural frequency and damping ratio. These values has highly effect on current controller response and would effect on speed controller. It can be seen that from Figure 11, speed reference is subtracted from feedback speed of the system and the error is enter to PI controller. The output of controller is I_{sq} current reference to the system. Therefore, it is very interesting to modified I_{sq} current response by using proposed combined PI-lead (CPIL) compensator. In addition, it is very important to achieve I_{sq} current with proposed controller faster than I_{sq} current with PI controller in order to get very fast speed response.

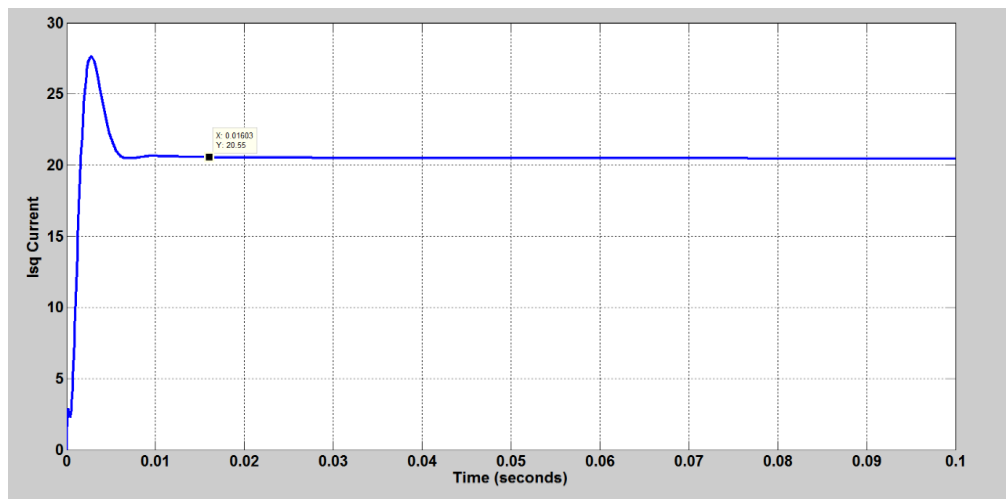


Figure 9. Stator quadrature (I_{sq}) current of the system with PI controller

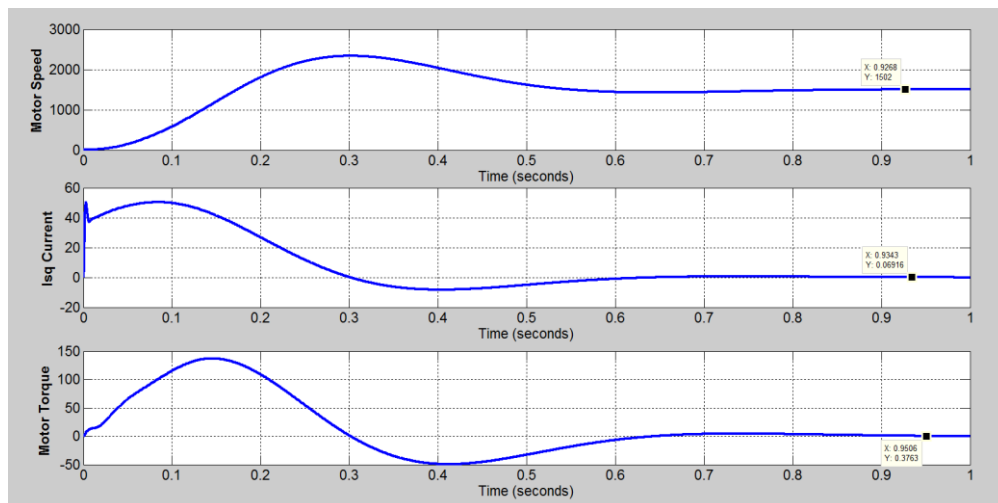


Figure 10. Stator quadrature current, motor torque and speed of the system with PI controller for current and speed

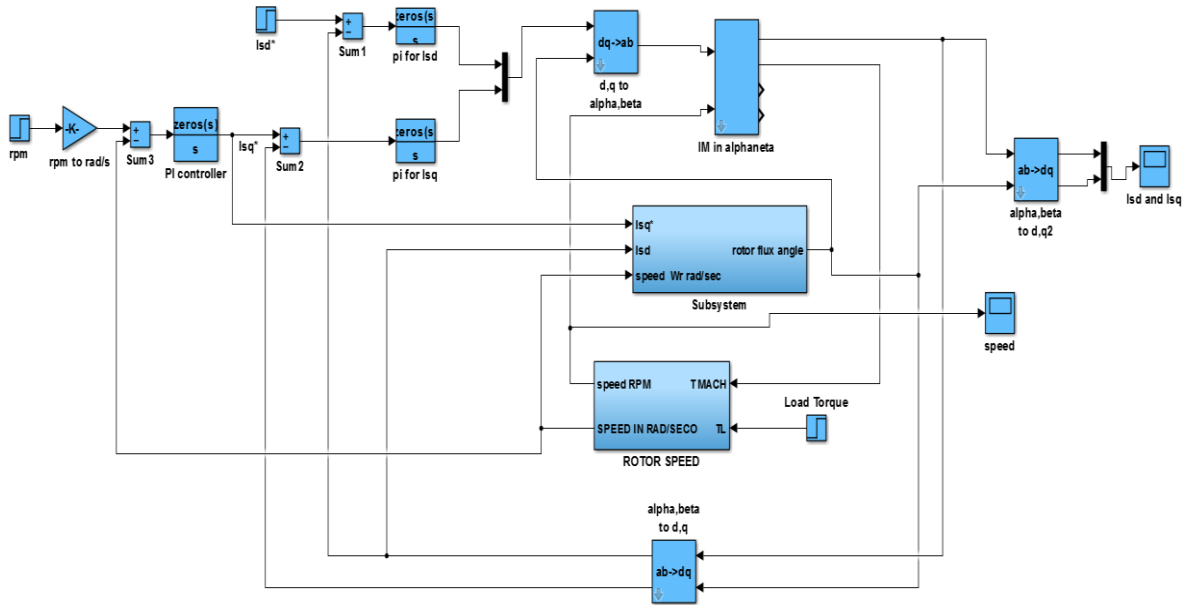


Figure 11. Simulink block of speed controlling of TIM with PI controller

3.2. Proposed controller

The drawbacks of PI controlling scheme are slow response speed and poor robust performance compared with the exogenous disturbances. In this research work, the phase-lead controller is used due to its superiority against conventional PI control action in terms of improve the dynamic performance as well as the transient response of the system. In case of a phase-lead compensator, the zero (Z) of the control law is located nearer to the origin of the S-plane as compared to the compensator's pole (P) [20].

The design of the proposed combined PI-lead (CPIL) compensator is developed to provide a better steady-state tracking, faster recovery from disturbances and perfect stability of the closed loop system. The inner loop controller has impact over the outer one; therefore, the behavior of the inner control law must influence the essential process variables in a predictable way. Finally, the response of the inner loop is quicker than the outer one. Finally, this permits the secondary action adequate time to compensate for inner loop feedback and thereafter they can improve the outer loop performance [21].

The transfer function formulation of the proposed CPIL controller is as follows:

$$G_{CPIL}(S) = K_P \left(\frac{S+Z}{S+P} \right) \left(1 + \frac{1}{T_i S} \right), \text{ where, } P > Z \tag{16}$$

The integral of time-multiplied square error (ITSE) is used as an objective function in this research work to design the proposed controller's parameters, and it's as follows [22]:

$$ITSE = \int_0^{t_{sim}} t [e(t)]^2 dt \tag{17}$$

where, t_{sim} is the simulation time period, the error function for the inner loop represents the I_{sd} and I_{sq} currents deviation and for the outer loop expressed by the speed deviation.

4. OPTIMIZATION ALGORITHMS

4.1. Standard grasshopper optimization algorithm

The SGO algorithm is a swarm intelligence algorithm that tries to mimic the swarm behaviour of the grasshopper insects in nature. In spite of the fact that grasshoppers are usually seen individually in nature, they join in one of the biggest swarms of all creatures. In their route, they eat practically all vegetation. After this reaction, when they become an adult, they establish a swarm in the air to preparing for migrate over large distances. It was confirmed by the research work based on SGOA that it is able of outperforming several well-known nature-inspired algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE), ant colony optimization (ACO), firefly algorithm (FA), bat algorithm (BA) [17].

The main characteristics of the grasshoppers are foraging, target seeking and team behaviour in either larval or adulthood phases. In the larval stage, they reveal slow motion with small steps. In adulthood stage, they do long-range and abrupt movement in the way to find the farming areas. The swarm-intelligence algorithms divide the search routine into two directions: exploration and exploitation. In exploitation, the grasshoppers are tended to move locally, while they encouraged to move suddenly during exploration. Therefore, grasshoppers carry out these two properties, as well as target pursuing naturally [23].

GOA simulates the forces between the grasshoppers, which are the attraction and repulsion. The attraction forces exploit promising space while the repulsion forces enable the grasshoppers to explore the search area. The zone at which the two attraction and repulsion forces are equal is called comfort zone. The position of grasshoppers with the optimum objective function is considered as the closest one to the destination target; accordingly, the rests try to relocation toward that place through swarm interaction during the algorithm procedure; the comfort zone is changed until the finest solution is attained [24]. The position updating for the swarming behaviour of the grasshopper can be modelled mathematically as follows [25]:

$$X_{i,t}^d = c \left(\sum_{j=1, j \neq i}^N c \frac{ub_d - lb_d}{2} S(|X_{j,t}^d - X_{i,t}^d|) \frac{X_{j,t}^d - X_{i,t}^d}{d_{ij}} + \hat{T}_d \right) \quad (18)$$

$$S(r) = f e^{\frac{-r}{l_s}} - e^{-r} \quad (19)$$

where, $X_{i,t}^d$ is the current position in t iteration for d dimension, c is the comfort zone coefficient, N is the swarm size, ub_d and lb_d are the upper and lower limits of the design variables, d_{ij} is the distance between $X_{j,t}$ and $X_{i,t}$ grasshoppers, \hat{T}_d is the target position in the d^{th} dimension that represent the best solution determined so far, $S(r)$ is a function, which calculates either repulsion or attraction forces, f represents the strength of the attraction, l_s is the length of attraction and r is a distance that has been considered in the interval $[0, 15]$.

The comfort zone coefficient is necessary to be decreased linearly *w.r.t.* the number of iterations to enhance the exploration and exploitation properties and is represented as follows:

$$c = c_{\max} - t \frac{(c_{\max} - c_{\min})}{T} \quad (20)$$

where, c_{\max} is the maximum limit, c_{\min} is the minimum limit, and T is the maximum number of iterations.

4.2. Dynamic adjustment grasshopper optimization algorithm

The SGOA has some drawbacks, such as the slow convergence to the optimal solution and the problem of falling in the local optimum [26]. In order to overcome these two deficiencies, a dynamic adjustment GOA scheme is proposed that combines the random initialization-based Gaussian distribution and the nonlinear decreasing comfort zone coefficient to obtain an appropriate exploitation and exploration balancing. Firstly, the enhancement for GOA is based on random candidate solution to increase population diversity, which can be initiated according to the following equation:

$$X_i^d = lb_d + (ub_d - lb_d) g(\alpha) \quad (21)$$

where, $g(\alpha)$ is the Gaussian density function that utilized to take smaller step vector inside the search space bounded by lower and upper limits to precisely explore and achieve a faster convergence. The Gaussian density function is as follows [26]:

$$g(\alpha) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{\alpha^2}{2\sigma^2}} \quad (22)$$

where, σ^2 is the variance for each individual of the population and α is the Gaussian random number between $[0, 1]$.

The second improvement is based on the parameter c_t that responsible for moderating the grasshoppers' movement towards the solution, and it are used twice in the position equation. The outer one acted to balances the exploration and exploitation of the search agents around the target. The inner coefficient minimizes the repulsion or attraction forces between grasshoppers relative to the number of iterations [27]. Therefore, in order to improve the comfort zone, attraction zone and repulsion zone between grasshoppers the following adjustment is proposed that leads to exploit the global optima in the search space.

$$c_t = (c_{max} - c_{min}) \left[e^{-a\frac{t}{T}} - \left(b \tan^{-1} \left(c \frac{t}{T} \right) \right)^3 \right] + c_{min} \quad (23)$$

where, $a = 1.4$, $b = 1.12$ and $c = 0.3$.

The search procedure of the proposed DAGOA is outlined as follows:

- Step 1: Initialize the input parameters of the algorithm; swarm size, no. of iterations, no. of tuned parameters (dimension size), maximum and minimum bound of the tuned parameters, c_{max} and c_{min} .
- Step 2: Assign the position value for the first iteration for all the grasshoppers in the search space.
- Step 3: Determine the system design parameters.
- Step 4: Calculate the objective function value for each individual in the swarm.
- Step 5: Sort the objective function values in the descending order and thereafter nominate the minimum value as \hat{T}_d .
- Step 6: Update the improved nonlinear decreasing comfort zone coefficient.
- Step 7: Normalize the distance between grasshoppers to find out d_{ij} .
- Step 8: Determine the social forces strength.
- Step 9: Update the position $X_{i,t}^d$.
- Step 10: Examine the termination condition; if current iteration number is equal to the maximum no. of iterations, collect the optimal solution for each dimension \hat{T}_d and save the global best design parameters. Otherwise, increase the iteration counter and go to Step 3.

5. SIMULATION RESULTS AND DISCUSSIONS

The implementation of the model, controlling techniques and the proposed algorithm are developed on Matlab/Simulink R2014a environment. The intensive simulations and performance evaluations were performed on a laptop with an Intel Core i5-4210 U, 2.70 GHz CPU, 8 GB RAM, and 64 bits Windows 7 operating system.

The algorithm parameters' setting are; $C_{min} = 0.00001$, $C_{max} = 1$, number of search agents = 15, number of iterations = 400 and $t_{sim} = 1$ minute. The controllers' parameters setting are; the upper and lower limits for inner loop controller are $0 < K_p \leq 500$, $0 < T_i \leq 50$, $0 < Z \leq 1$ & $0 < P \leq 1$ and for outer loop controller are $0 < K_p \leq 20$, $0 < T_i \leq 10$, $0 < Z \leq 1$ & $0 < P \leq 1$.

5.1. Numerical results for performance evaluation of algorithms

The purpose of this subsection is to assess the performance of the proposed optimization algorithm quantitatively. The ordinary procedure used is to utilize chosen benchmark test functions with known optimal values to perform a capabilities' comparison between the SGOA and DHGOA algorithms on the basis of solution accuracy.

The numerical results contain three sets of various test functions; Unimodal set (F1-F5), Multimodal set (F6-F8) and Composite set (F9 & F10) [17, 28-30]. Each test was repeated 30 times in order to obtain reliable statistical results for 10 mathematical benchmark functions with the average best (AB), the median best (MB) and the standard deviation (SD) that are tabulated as in Table 2.

The obtained results of the selected three sets of test functions show that the proposed DAGOA scheme provides competitive results throughout touching the global optimum solution for all the 10 test problems with an average executing time of about 5 seconds. Therefore, the modifications performed on the SGOA enhance the abilities of the proposed algorithms in terms of exploitation and exploration balancing and maximizing the search capability and stability.

5.2. Simulation results of TIM

The system current controller using CPIL is implemented in Matlab/Simulink as illustrated in Figure 12. Where, the CPIL is connected to the system as current controller. SGOA and DAGOA algorithms are used for tuning CPIL controller parameters. From Figure 13, speed controller of TIM using CPIL is implemented in Matlab/Simulink, which is tuned by SGOA and DAGOA algorithms.

From Figure 14, it can be seen that the stator quadrature current arrived steady state at 1.2×10^{-7} s when CPIL-DAGOA is used for tuning the controller parameters. In addition, I_{sq} with CPIL-SGOA is faster than I_{sq} using conventional PI controller. From Figure 15, it can be seen that I_{sq} current arrived steady state at 0.1×10^{-7} s when CPIL-DAGOA is used for tuning controller. Furthermore, I_{sq} with CPIL-DAGOA is faster than I_{sq} using CPIL-SGOA and I_{sq} using conventional PI controller. Therefore, when I_{sq} current settling time reach very fast value that is mean, the system speed response is faster because the output of the speed controller is considered I_{sq}^* reference to the system.

From Figures 16 and 17, it can be seen that I_{sq} starting current is about 200 A which is within limit and with the system requirements. In addition, from Figure 16, speed of the system with CPIL-SGOA for tuning controller parameters arrived to steady state at 0.53 s while speed of the system with CPIL-DAGOA for tuning controller parameters reached to settle at 0.28 s. Hence, that is mean system with proposed controller using DAGOA is faster than the system with CPIL-SGOA by two times.

Table 2. Statistical results based on the selected benchmark test functions

No.	Test function name	Criteria	SGOA	DAGOA
F1	Schwefel absolute	AB	0.0014	5.2E-5
		MB	7.5E-3	1.5E-4
		SD	0.0089	1.24E-4
F2	Schwefel double sum	AB	0.179	0.0026
		MB	1.183	0.0035
		SD	1.193	0.0157
F3	Schwefel 2.21	AB	0.0043	0.0033
		MB	0.037	0.0074
		SD	0.0344	0.0112
F4	Step 2	AB	0.121	0.0813
		MB	0.6	0.496
		SD	0.7	0.58
F5	Quartic	AB	0.04	0.0293
		MB	0.2512	0.1341
		SD	0.152	0.0944
F6	Restrigin	AB	0.18	0.0245
		MB	2.98	1.952
		SD	1.22	0.00247
F7	Penalty #1	AB	0.325	0.079
		MB	1.325	0.949
		SD	2.235	1.232
F8	Penalty #2	AB	0.9	0.14
		MB	1.652	1.002
		SD	1.35	0.75
F9	Composite CF2	AB	1.3256	0.253
		MB	3.289	1.385
		SD	7.236	5.923
F10	Composite CF3	AB	0.852	0.134
		MB	2.965	1.985
		SD	1.125	0.512

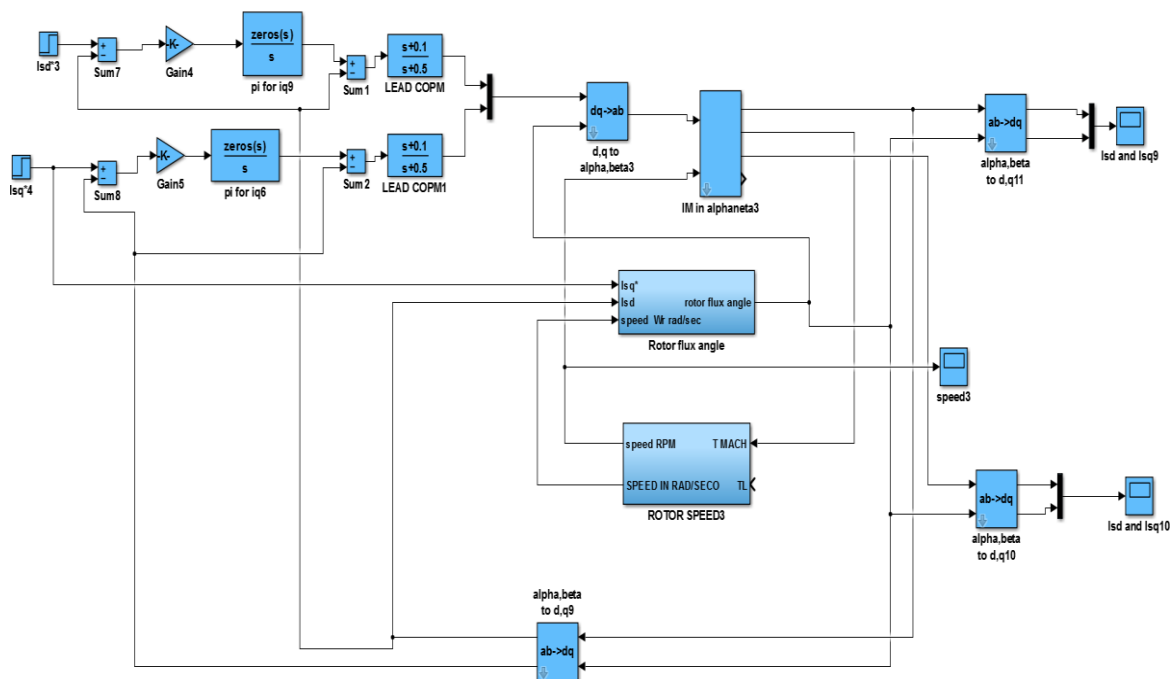


Figure 12. Stator quadrature and direct currents of TIM with proposed CPIL controller

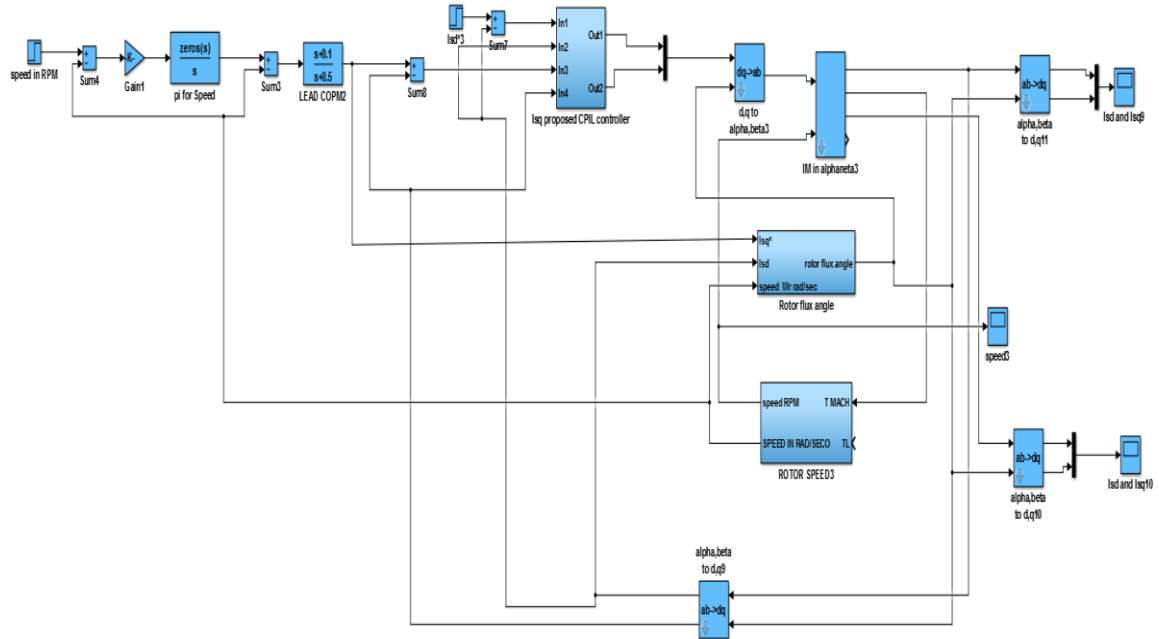


Figure 13. Speed of TIM with proposed CPIL controller

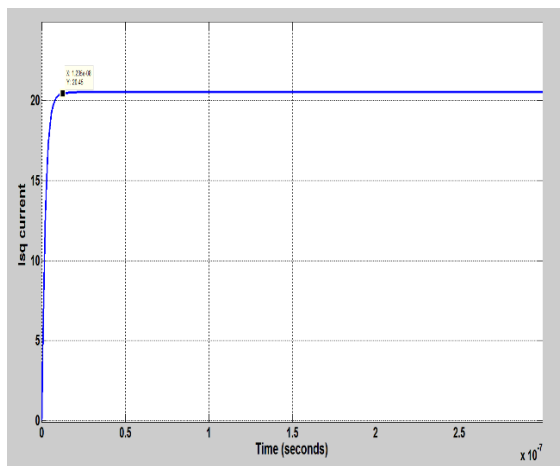


Figure 14. Stator quadrature current with CPIL-SGOA

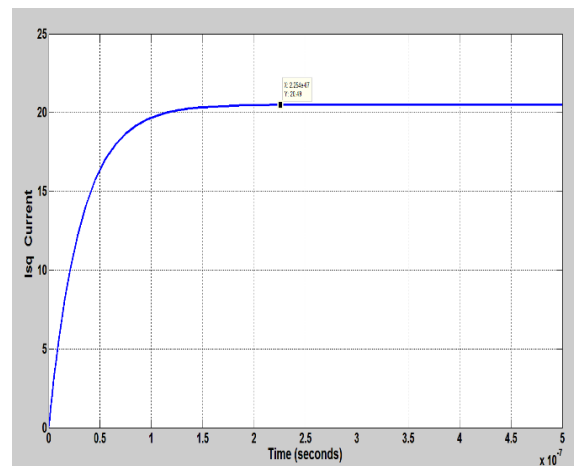


Figure 15. Stator quadrature current with CPIL-DAGOA

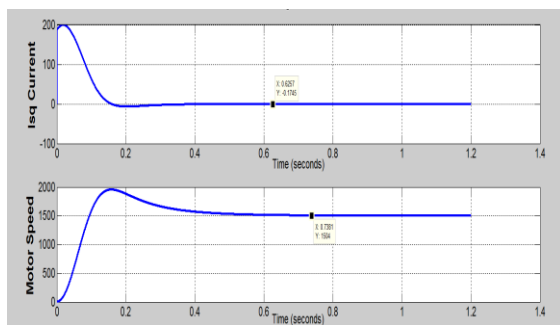


Figure 16. Stator quadrature current and speed of the system using CPIL-SGOA

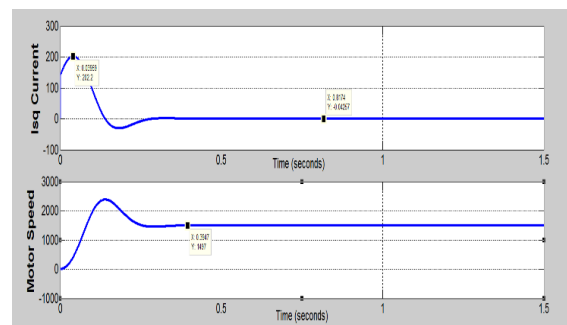


Figure 17. Stator quadrature current and speed of the system using CPIL-DAGOA

5.3. Performance comparison of controlling techniques

In this section, three-phase induction motor current controller using CPIL-DAGOA is compared with current controller using CPIL-SGOA and conventional PI current controller for tuning parameters. In addition, speed controller of the system with CPIL-DAGOA is compared with system speed using CPIL-SGOA for tuning parameters and system speed using PI controller. It can be seen that the system with CPIL-DAGOA scheme for tuning parameters has higher performance and faster than the system using the CPIL-SGOA and system with PI using SGOA as depicted in Figure 18. Figure 18 shows that I_{sq} current of the TIM with CPIL using DAGOA reached settling time of 0.1×10^{-7} s, while I_{sq} current of the system with CPIL-SGOA reached settling time of 1.4×10^{-7} s and system with PI controller arrived steady state at 0.006 s. Finally, that is mean; system with CPIL using DAGOA is faster than system with SGOA and Conventional PI controller as illustrated in Table 3.

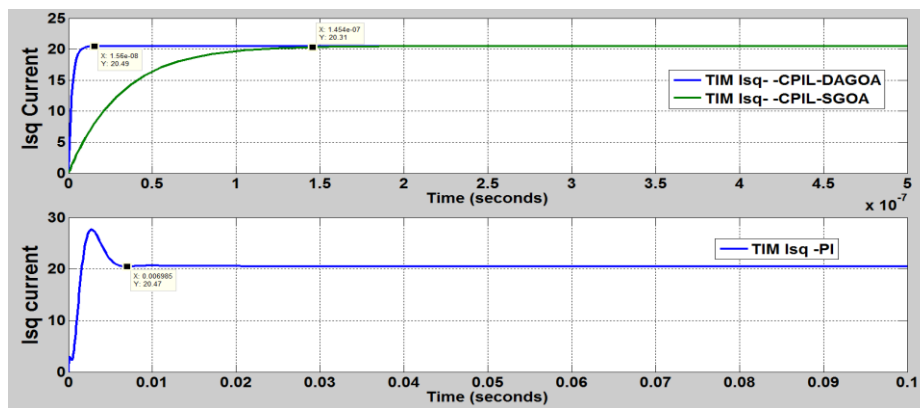


Figure 18. Stator quadrature current of the system using the three controlling methods

Table 3. Performance comparison between the three types of controllers

Controlling techniques	Settling time	ITSE
Stator quadrature current of the system using PI controller	0.006 s	0.3796
Speed of the system using PI controller	0.85 s	3373
Stator quadrature current of the system using CPIL controller tuned by SGOA	1.4×10^{-7} s	6.52×10^{-6}
Speed of the system using CPIL controller tuned by SGOA	0.55 s	1500
Stator quadrature current of the system using CPIL controller tuned by DAGOA	0.1×10^{-7} s	4.42×10^{-9}
Speed of the system using CPIL controller tuned by DAGOA	0.28 s	290

In Figure 19, the speed response of the three-phase induction motor with CPIL-DAGOA scheme reached settling time at 0.3 s, while, system speed with CPIL using SGOA reached settling time at 0.55 s and system with PI controller arrived steady state at 0.85 s. Therefore, that is mean; the system with CPIL using DAGOA is faster than system with CPIL-SGOA and conventional PI controller as demonstrated in Table 3.

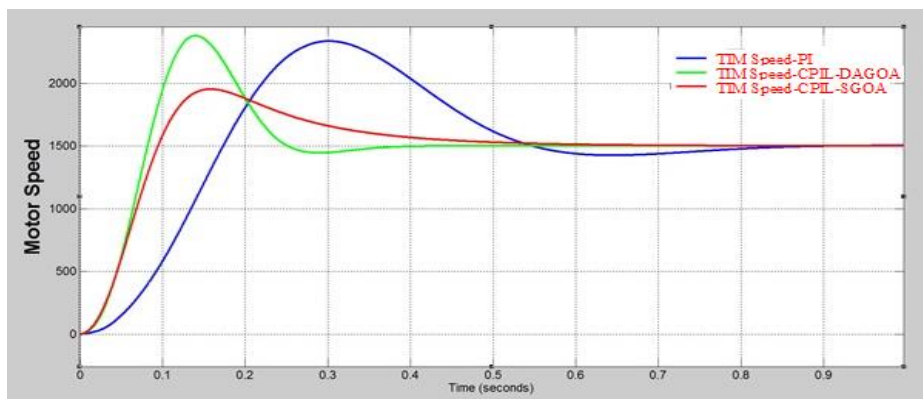


Figure 19. Motor speed of the system using the three controlling methods

In Figure 20, it can be seen that the speed of the TIM with CPIL using DAGOA method reached settling time faster than system with CPIL using SGOA and Conventional PI controller when load torque is applied at 1 s. Additionally, the disturbance due to load torque has less impact on system with CPIL using DAGOA for tuning parameters. While, the disturbance from load torque has high impact on system speed with PI controller, where speed dropped to below 1000 rpm and system has higher under shoot due to load torque when applied at 1 s.

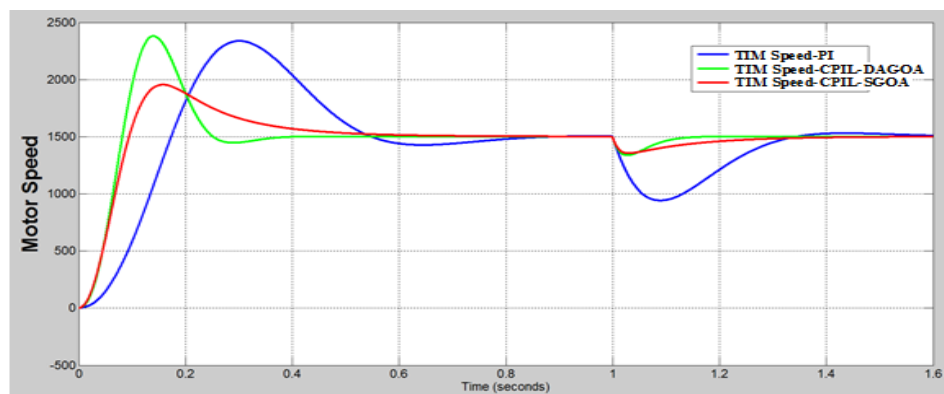


Figure 20. Motor speed of the system using three controlling methods when applied a load torque at 1 s

6. CONCLUSION

The mathematical modelling of the three-phase induction motor has been implemented in Matlab/Simulink in α and β coordinates. Design of conventional PI controller for the current and speed of the system has been performed. Proposed CPIL controller for the current and speed have been connected with the system to improve the performance. The SGOA algorithm has been used for tuning the proposed controller parameters of current and speed of the system. Additionally, the SGOA has been modified and a DAGOA for tuning CPIL controller parameters for currents and speed of TIM is proposed. Furthermore, the effect of applying load torque on TIM after 1 s has been carried out to examine the proposed controller and optimization algorithm robustness. The computational efficiency and convergence rate of the developed DAGOA algorithm has been validated by ten benchmark functions. The simulation results indicate and confirm the effectiveness of the proposed CPIL tuned by DAGOA for the current and speed of the system. Moreover, the current and speed controlling of TIM with CPIL using DAGOA are faster than current and speed of TIM with proposed CPIL using SGOA and PI controller. Furthermore, load torque has less impact on speed of the system with proposed CPIL-DAGOA when load torque is applied at 1 s, whereas load torque has a high impact on system speed with conventional PI controller, where the system speed dropped to below 1000 rpm when load torque is applied. Finally, it is concluded that the proposed compensator and optimization technique is robust and satisfying the design requirements.

REFERENCES

- [1] J. A. Ali, M. A. Hannan and A. Mohamed, "PSO Algorithm for Three Phase Induction Motor Drive with SVPWM Switching and V/f Control," *Proceedings of IEEE International Conference on Power and Energy (PECon)*, Dec. 2014, pp. 250-254.
- [2] P. Talukder, P. K. Soori and B. Arango, "Speed Control of Induction Motor Drive using Universal Controller," *Proceedings of IEEE International Power Engineering and Optimization Conference*, Jun. 2012, pp. 509-514.
- [3] R. K. Antar, N. S. Sultan and A. J. Ali, "Speed Control of Three-Phase Induction Motor Fed by Renewable Energy Source," *Proceedings of 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPE)*, Feb. 2019, pp. 7-12.
- [4] F. Yasien and W. Nasser, "Speed Controller of Three Phase Induction Motor Using Sliding Mode Controller," *Journal of Engineering*, vol. 25, no. 7, pp. 118-133, 2019.
- [5] I. Ferdiansyah, et al., "Speed Control of Three Phase Induction Motor Using Indirect Field Oriented Control Based on Real-Time Control System," *Proceedings of 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 438-442.
- [6] S. K. Yong, K. H. Law, W. P. Q. Ng and M. Dahidah, "Lead Compensator Design for Single-Phase Quasi Z-Source Inverter," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-12, pp. 39-44, 2018.

- [7] I. Ferdiansyah, L. P. S. Raharja, D. S. Yanaratri and E. Purwanto, "Design of PID Controllers for Speed Control of Three Phase Induction Motor Based on Direct-Axis Current (Id) Coordinate Using IFOC," *Proceedings of 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2019, pp. 369-372.
- [8] O. Rabiaa, B. H. Mouna, D. Mehdi and S. Lassaad, "Scalar Speed Control of Dual Three phase Induction Motor using PI and IP Controllers," *Proceedings of International Conference on Green Energy Conversion Systems (GECS)*, Mar. 2017, pp. 1-6.
- [9] G. A. Olarinoye, C. Akinropo, G. J. Atuman and Z. M. Abdullahi, "Speed Control of a Three Phase Induction Motor using a PI Controller," *Proceedings of 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, Oct. 2019, pp. 1-7.
- [10] T. Ramesh and A. K. Panda, "Direct Flux and Torque Control of Three Phase Induction Motor Drive using PI and Fuzzy Logic Controllers for Speed Regulator and Low Torque Ripple," *Proceedings of Students Conference on Engineering and Systems*, Mar. 2012, pp. 1-6.
- [11] R. Bharti, M. Kumar and B. M. Prasad, "V/F Control of Three Phase Induction Motor," *Proceedings of International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Mar. 2019, pp. 1-4.
- [12] A. J. Fattah and I. Abdel-Qader, "Performance and Comparison Analysis of Speed Control of Induction Motors using Improved Hybrid PID-Fuzzy Controller," *Proceedings of IEEE International Conference on Electro/Information Technology (EIT)*, May 2015, pp. 575-580.
- [13] A. Z. A. Firdaus, S. A. Azmi, K. Kamarudin and N. I. Norazli, "Design and Simulation of Fuzzy Logic Controller for Variable Frequency Drive of Three Phase AC Induction Motor," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 12, pp. 834-842, 2019.
- [14] A. N. Abdullah and M. H. Ali, "Direct Torque Control of IM using PID Controller," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 617-625, 2020.
- [15] P. Chiewchitboon, V. Tipsuwanporn, N. Soonthornphisaj and W. Piyarat, "Speed Control of Three-Phase Induction Motor Online Tuning by Genetic Algorithm," *Proceedings of Fifth International Conference on Power Electronics and Drive Systems, Singapore*, vol. 1, pp. 184-188, Nov. 2003.
- [16] T. N. Trong, "The Neural Network-Combined Optimal Control System of Induction Motor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2513-2522, 2019.
- [17] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper Optimization Algorithm: Theory and Application," *Advances in Engineering Software*, vol. 105, pp. 30-47, Mar. 2017.
- [18] H. Abu-Rub, A. Iqbal and J. Guzinski, "High performance control of AC drives with MATLAB/Simulink models," *1st edition, Wiley*, 2012.
- [19] M. Pfeifer, T. Weickert and S. Klose, "Rule-based PI Controller Autotuning for Drive Systems," *Proceedings of European Control Conference (ECC)*, Linz, 2015, pp. 672-677.
- [20] Z. K. Jadoon, et al., "A Comparative Analysis of PID, Lead, Lag, Lead-Lag, and Cascaded Lead Controllers for a Drug Infusion System," *Journal of Healthcare Engineering*, vol. 2017, no. 5, pp. 1-13, 2017.
- [21] R. Prakash, S.A. Anburaja, S. Rishivanth and E. Govinda Kumar, "Non-Overshoot Time Response of Third Order System Using Cascade PID-Lead Compensator Controller," *Proceedings of Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Feb. 2017.
- [22] M. Bahy, A. S. Nada, S. H. Elbanna and M. A. M. Shanab, "Voltage Control of Switched Reluctance Generator using Grasshopper Optimization Algorithm," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 11, no. 1, pp. 75-85, Mar. 2020.
- [23] A. A. Heidari, H. Faris, I. Ajarah and S. Mirjalili, "An Efficient Hybrid Multilayer Perceptron Neural Network with Grasshopper Optimization," *Soft Computing*, vol. 23, pp. 7941-7958, 2019.
- [24] A. Fathy, "Recent Meta-heuristic Grasshopper Optimization Algorithm for Optimal Reconfiguration of Partially Shaded PV Array," *Solar Energy*, vol. 171, pp. 638-651, 2018.
- [25] A. Fathy and O. El-baksawi, "Grasshopper Optimization Algorithm for Extracting Maximum Power from Wind Turbine Installed in Al-Jouf Region," *Journal of Renewable and Sustainable Energy*, vol. 11, pp. 1-14, 2019.
- [26] J. Luo, H. Chen, Q. zhang, Y. Xu, H. Huang, and X. Zhao, "An Improved Grasshopper Optimization Algorithm with Application to Financial Stress Prediction," *Applied Mathematical Modelling*, vol. 64, pp. 654-668, 2018.
- [27] C. Zhou, J. Ma, J. Wu and X. Yuan, "An Adaptive VMD Method based on Improved GOA to Extract Early Fault Feature of Rolling Bearings," *International Journal of Innovative Computing, Information and Control*, vol. 15, no. 4, pp. 1485, 1505, Aug. 2019.
- [28] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150-194, 2013.
- [29] J. Li, L. Guo, Y. Li and C. Liu, "Enhancing Elephant Herding Optimization with Novel Individual Updating Strategies for Large-Scale Optimization Problems," *Mathematics*, vol.7, no. 5, p. 395, 2019.
- [30] S.-S. Guo, J.-S. Wang and M.-W. Guo, "Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1-21, 2020.