

A robust speed sensorless vector control of multilevel inverter fed induction motor using particle swarm optimization

Sanjaya Kumar Sahu¹, D.D. Neema²

Dept. of Electrical Engg., Bhilai Institute of Technology, Durg, C.G., India¹

Dept. of Electrical & Electronics Engg., Chhattisgarh Institute of Technology Rajanandgaon, C.G., India²

Abstract: A novel speed sensor less adaptive robust control method is proposed to improve the trajectory tracking performance of induction motors. The proposed design employs the so called vector control (or field oriented control) theory for the multilevel inverter fed induction motor drives. The inverter design is based on three-level Neutral Point Clamped (NPC) inverter with hysteresis current control technique. Two Mamdani type fuzzy logic controllers are used; one as speed controller and the other is in Luenberger Observer in order to estimate the actual rotor speed. The Particle Swarm Optimization algorithm is used to optimize the parameters such as membership functions, normalizing and de-normalizing parameters of fuzzy logic controller. The performance of proposed scheme is investigated under various load and speed conditions. The simulation results show its stability and robustness for high performance sensor less drive applications.

Keywords: Field Oriented Control (FOC), Multilevel Inverter, Hysteresis Current Control, Particle Swarm Optimization (PSO), Fuzzy Logic Controller.

I. INTRODUCTION

Three phase induction motors are widely used in the industrial applications because they show better performance during heavy loads and cost effective. However the drawbacks associated with induction motor are its non-linear behaviour, controllability and its complexity in developing mathematical model. By vector control or field oriented control (FOC) theory, induction motor can be controlled like a separately excited dc motor. As a result, field and torque of the induction motor can be controlled independently by manipulating the corresponding field oriented quantities [1]–[3]. In this method, the knowledge of the rotor speed is necessary and such necessity requires additional speed sensor. However, a speed sensor cannot be mounted in some cases, such as motor drives in a adverse environment, or high-speed motor drives. Moreover, such sensors lower the system reliability, add to the cost and require special attention to noise. Therefore, sensorless induction motor drives are widely used in industry for their reliability and flexibility, particularly in hostile environments. In order to achieve good performance of sensor less vector control, different speed estimation schemes have been proposed, and a variety of speed estimators exist now days.

The multilevel inverters gained the attention in industrial drive application due to several reasons like better waveform quality, lower size and rating of filter components, lower dv/dt across switches, lower distortion in input current and smaller common mode voltage [4]. The Neutral point clamped (NPC) multilevel inverters are very popular for high voltage and high power applications. Theoretically, NPC topology with any number of levels can be realized. But some of the problems like complexity of switching algorithm, voltage unbalance across

capacitors, voltage clamping requirements, and circuit layouts have limits on the level in practical multilevel inverters [5].

Speed estimation methods using Luenberger observer, Model Reference Adaptive System (MRAS) are the most commonly used as they are easy to design and implement. However, the performance of these methods is deteriorated at low speed because of the increment of non-linear characteristics [6], [7].

In recent years, Fuzzy logic has emerged as an important artificial intelligence tool to characterize and control a system, whose model is not known or ill defined. Fuzzy logic controllers are non-linear elements used in the control of linguistically defined systems, and cannot be modelled accurately. In design of fuzzy logic controllers, there is no well-defined approach. The sophisticated and tedious design process is usually implemented by an expert. In some cases, even a very experienced and skilful expert's extensive efforts may not yield optimal solution for fuzzy logic controller design. The design inherently requires the determination of normalization parameter, de-normalization parameter, membership functions and proper linguistic rules. The conventional trial-and-error based methods make solution very difficult [8], [9].

Recently, there has been a huge interest in the Particle Swarm Optimization (PSO) due to its great potential as an evolutionary algorithm, which is based on the social behaviour of flocks of birds and schools of fish [10]. Since it is population based and self-adaptive, it has gained an increasing popularity as an efficient alternative to the Genetic Algorithm (GA) in solving optimization problem. Similar to other population-based optimization method such as the GA, the PSO algorithm starts with random

initialization of a population of individuals in the search space. Each particle in the search space is adjusted by its own flying experience and the other particles flying experience to find the global best solution at each generation [10].

This paper proposes a robust sensor less vector control of three level inverter fed induction motor. The three-level inverter is an NPC inverter with three level hysteresis current control technique. The design includes a Luenberger observer with a Mamdani type fuzzy logic controller in order to estimate the rotor speed. The conventional PI speed controller is also replaced by a Mamdani type fuzzy logic controller. This paper also involves the development of novel methodology to optimize the performance of Mamdani type fuzzy logic controller, based on a pre-defined objective function. The pre-defined objective function is optimized off-line by optimizing the normalization parameter, de-normalization parameter and the membership functions of the fuzzy logic controller.

II. MODELLING

A. Induction Motor

The three-phase squirrel cage induction motor mathematical equations in synchronous rotating reference frames are as follows [1]–[3]:

$$V_{ds}^e = R_s i_{ds}^e + p\lambda_{ds}^e + \omega_e \lambda_{qs}^e \quad (1)$$

$$V_{qs}^e = R_s i_{qs}^e + p\lambda_{qs}^e + \omega_e \lambda_{ds}^e \quad (2)$$

$$0 = R_r i_{dr}^e + p\lambda_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e \quad (3)$$

$$0 = R_r i_{qr}^e + p\lambda_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e \quad (4)$$

Where,

$$\lambda_{ds}^e = L_s i_{ds}^e + L_m i_{dr}^e \quad (5)$$

$$\lambda_{qs}^e = L_s i_{qs}^e + L_m i_{qr}^e \quad (6)$$

$$\lambda_{dr}^e = L_r i_{dr}^e + L_m i_{ds}^e \quad (7)$$

$$\lambda_{qr}^e = L_r i_{qr}^e + L_m i_{qs}^e \quad (8)$$

And electromagnetic torque

$$T_e = \frac{3}{2} \frac{P}{2} L_m (i_{qs}^e i_{dr}^e - i_{ds}^e i_{qr}^e) \quad (9)$$

$$w_r = \frac{d\theta_r}{dt} \quad (10)$$

$$T_e = j_m \frac{dw_r}{dt} + B_m w_r + T_l \quad (11)$$

B. Three-Level NPC Inverter

The three-level NPC inverter with two DC link capacitors C_1 and C_2 in series and a neutral point O is shown in Figure 1.

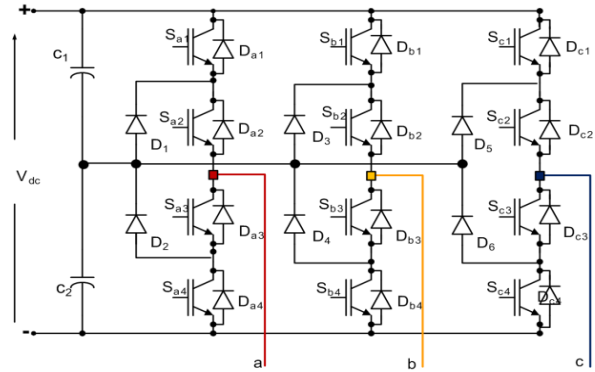


Figure 1: Three Level NPC Inverter

Each phase of the three-level NPC inverter has two pair of switching devices S_{i1} , S_{i2} and S_{i3} , S_{i4} in series, where $i = a, b, c$ phases. The centre of each pair is clamped to the neutral of the DC link capacitors through the clamping diodes D_1, D_2, D_3, D_4, D_5 and D_6 .

Table 1: Switching Levels in a Three-Level NPC Inverter

S_{i1}	S_{i2}	S_{i3}	S_{i4}	i^{th} Pole voltage V_{io}
ON	ON	OFF	OFF	$+V_{dc} / 2$
OFF	ON	ON	OFF	0
OFF	OFF	ON	ON	$-V_{dc} / 2$

Table 1 enumerates the switching states for the semiconductor devices for the i^{th} -phase of this inverter. In Table 1, the switching symbols +, 0 and - respectively denote that the i^{th} -phase terminal is connected to the positive bus, the neutral point and the negative bus.

C. Three-Level Hysteresis Current Controller

An analytical solution of different multilevel PWM techniques for three-level NPC has been presented [11], [12]. Among these techniques, the hysteresis band is used very often because of its simplicity of implementation, fast response current and robust structure [13], [14]. Hysteresis band controller is used to track the line current references. The current errors between the reference and measured currents are used to develop three valid switching states in each inverter leg by the hysteresis band controller.

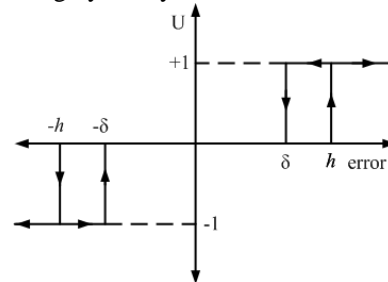


Figure 2: Three-Level Hysteresis Switching Scheme

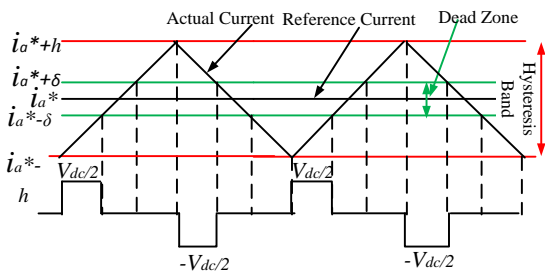


Figure 3: Three-Level Hysteresis Current Control

To develop a switching scheme for the three-level inverter, the zero voltage level should be applied only at appropriate instants. The switching logic must ensure that there is no successive transition between $+V_{dc}/2$ and $-V_{dc}/2$ states, as this will increase the frequency of switching. A dead zone 'δ' is necessary in the hysteresis band 'h', to avoid switching towards two-level scheme, because of finite sampling rate of error. Without the dead zone, when the error becomes zero and is not detected, the opposite polarity of forcing function follows, resulting in a two-level scheme. However, the introduction of dead zone increases the tracking error and has to be chosen to a minimum value, depending on the best sampling speed that can be achieved [15].

If U represents the input state to be applied, e represents error ($i_a^* - i_a$) and ce represents the change in error then from Figure 2 and Figure 3, the switching logic is governed by equation (12)

If $e > 0$ then

$U = 1$ for $e \geq h$

$U = 0$ for $e \leq \delta$

$U = 0$ for $\delta < e < h$ and $ce > 0$

$U = 1$ for $\delta < e < h$ and $ce < 0$

Else if $e < 0$ then

$U = -1$ for $e \leq -h$

$U = 0$ for $e \geq -\delta$

$U = 0$ for $-\delta < e < -h$ and $ce < 0$

$U = -1$ for $-\delta < e < -h$ and $ce > 0$ (12)

The above logic represented in Figure 2 and Figure 3 tracks reference current either in the lower band (through 0 and +1 states) or in the upper band (through 0 and -1 states). Here $U = 1$, means the switch state is $+V_{dc}/2$; $U = 0$ means the switch state is 0; and $U = -1$, means the switch state is $-V_{dc}/2$. Similarly the b -phase and c -phase switching function for the three-phase voltage source inverter can be obtained.

D. Indirect Vector Control

The indirect vector control is a technique that controls the dynamic speed of Induction motor. Unlike direct vector control, in indirect vector control, the unit vectors are generated in an indirect manner. Figure 4 is the phasor diagram that explains the fundamental principle of indirect

vector control. The $d^s - q^s$ axes are fixed on the stator and $d^r - q^r$ axes are fixed on the rotor which rotates at a speed ω_r . Synchronously rotating axes $d^e - q^e$ are rotating ahead of $d^r - q^r$ axes by the positive slip angle θ_{sl} corresponding to slip frequency ω_{sl} . Thus

$$\theta_e = \int \omega_e dt = \int (\omega_r + \omega_{sl}) dt \quad (13)$$

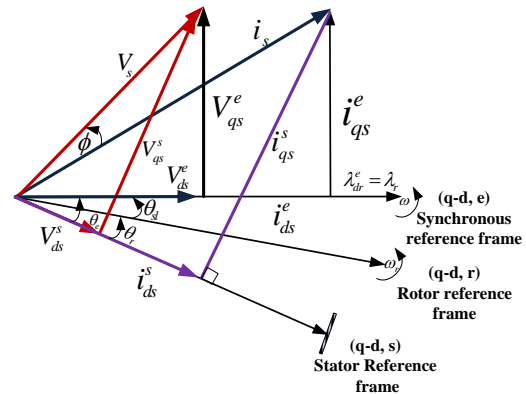


Figure 4: Phasor diagram of Indirect Vector Control principle

For decoupling control $\lambda_{qr} = 0$ or $p\lambda_{qr} = 0$ and

$\lambda_r = \lambda_{dr}$. Substituting the above condition in equations (3), (4), (7) and (8).

$$\omega_{sl} = \frac{R_r L_m i_{qs}^e}{L_r \lambda_r} \quad (14)$$

$$T_e = \frac{3 P L_m}{2 L_r} \lambda_r i_{qs}^e \quad (15)$$

$$i_{qs}^e = \frac{2 L_r T_e}{3 P L_m \lambda_r} \quad (16)$$

$$i_{ds}^e = \frac{1}{L_m} \left[\lambda_r + \frac{L_r}{R_r} p \lambda_r \right] \quad (17)$$

The equations (14-17) are used to produce an adequate field orientation. These equations could be propagated to the set point variables [16].

$$\omega_{sl}^* = \frac{R_r L_m i_{qs}^{e*}}{L_r \lambda_r^*} \quad (18)$$

$$i_{qs}^{e*} = \frac{2 L_r T_e^*}{3 P L_m \lambda_r^*} \quad (19)$$

$$i_{ds}^{e*} = \frac{1}{L_m} \left[\lambda_r^* + \frac{L_r}{R_r} p \lambda_r^* \right] \quad (20)$$

If it is accepted that the rotor flux set point is constant then its derivative is zero and the above equation is simplified as

$$i_{ds}^* = \frac{\lambda^*}{L_m} \quad (21)$$

E. Speed Estimator

The proposed scheme includes a Luenberger Observer for the speed estimation from stator voltage and current. The Luenberger Observer belongs to the group of closed loop observers. It is a deterministic type of observer because it is based on a deterministic model of the system [17]. This observer can reconstruct the state of a system observable from the measurement of inputs and outputs. It is used when all or part of the state vector cannot be measured. It allows the estimation of unknown parameters or variables of a system.

The state equations of an Induction motor in stationary reference frame are given by.

$$\dot{X} = AX + Bv_s \quad (22)$$

Where,

$$X = \begin{bmatrix} i_s \\ \lambda_r \end{bmatrix}, \quad i_s = \begin{bmatrix} i_{ds}^s \\ i_{qs}^s \end{bmatrix}, \quad \lambda_r = \begin{bmatrix} \lambda_{dr}^s \\ \lambda_{qr}^s \end{bmatrix},$$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$A_{11} = -\left(\frac{R_s}{\sigma L_s} + \frac{(1-\sigma)}{\sigma T_r}\right)I = -\gamma I,$$

$$A_{12} = \frac{L_m}{\sigma L_s L_r} \left(\frac{1}{T_r}I - \omega_r J\right) = \delta \left(\frac{1}{T_r}I - \omega_r J\right),$$

$$A_{21} = \left(\frac{L_m}{T_r}\right)I, \quad A_{22} = -\left(\frac{1}{T_r}I - \omega_r J\right),$$

$$B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \quad B_1 = \frac{1}{\sigma L_s}I$$

I=Unit matrix of proper order and $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$,

$$v_s = \begin{bmatrix} v_{ds}^s \\ v_{qs}^s \end{bmatrix}, \quad \sigma = 1 - \frac{L_m^2}{L_s L_r}, \quad T_r = \frac{L_r}{R_r},$$

$$\gamma = \left(\frac{R_s}{\sigma L_s} + \frac{(1-\sigma)}{\sigma T_r}\right), \quad \delta = \frac{L_m}{\sigma L_s L_r}$$

Thus if the speed signal ω_r is known in the matrix A then the fluxes and currents can be solved from the state equations. However, ω_r is not correct, there will be a deviation between the estimated states and actual states. So the estimated currents are compared with the actual machine terminal currents and the errors are injected

through a gain matrix K as shown in Figure 5, so that the error tends to vanish. The state observer which estimates the stator current and rotor flux together is given by [18]

$$\dot{\hat{X}} = \hat{A}\hat{X} + Bv_s + K(\hat{i}_s - i_s) \quad (23)$$

Where, $\hat{i}_s = \begin{bmatrix} \hat{i}_{ds}^s \\ \hat{i}_{qs}^s \end{bmatrix} = C\hat{X}$

$$C = [I \ 0] \quad \text{and} \quad 0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The symbol ^ represents the estimated values and K is the observer gain matrix which is to be decided so that the above equation can be stable.

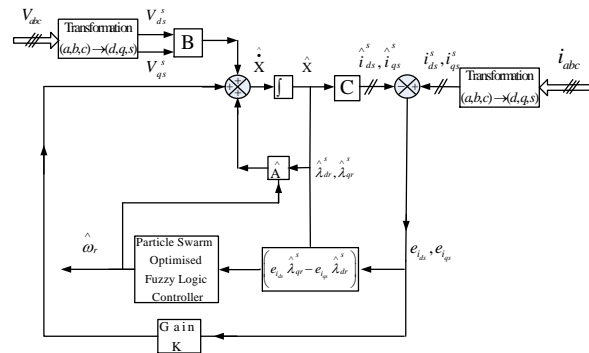


Figure 5: Luenberger Speed Observer

The speed adaptive flux observer permits the estimation of the unknown speed ω_r in matrix A. To derive the speed adaption algorithm Lyapunov's theorem is used. The estimation error of the stator currents and rotor fluxes is described by

$$\dot{e} = (A + KC)e - \Delta A \hat{X} \quad (24)$$

Where, $e = \hat{X} - X$, $\Delta A = \hat{A} - A = \begin{bmatrix} 0 & -\Delta\omega_r \delta J \\ 0 & \Delta\omega_r J \end{bmatrix}$,

$$\Delta\omega_r = \hat{\omega}_r - \omega_r$$

Now let's define the Lyapunov's function candidate:

$$V = e^T e + \frac{(\hat{\omega}_r - \omega_r)^2}{\lambda} \quad (25)$$

Here, λ is a positive constant. This function is zero when the error (e) is zero and when the estimated speed $\hat{\omega}_r$ is equal to the actual speed ω_r . Since a sufficient condition for uniform asymptotic stability is that the derivative

Lyapunov function \dot{V} is negative definite. The time derivative of V becomes

$$\dot{V} = e^T \left[(A + KC)^T + (A + KC) \right] e - \left(\Delta A^T \hat{X}^T e + \Delta A \hat{X} e^T \right) + 2 \frac{\Delta\omega_r}{\lambda} \dot{\omega}_r \quad (26)$$

Substituting the values of $e, e^T, \hat{X}, \hat{X}^T, \Delta A$ and ΔA^T in (26)

$$\dot{V} = e^T \left[(A + KC)^T + (A + KC) \right] e - 2 \frac{L_m}{L_r} \Delta \omega_r \frac{\left(e_{ids} \hat{\lambda}_{qr}^s - e_{iqs} \hat{\lambda}_{dr}^s \right)}{\sigma L_s} + 2 \frac{\Delta \omega_r}{\lambda} \dot{\omega}_r \quad (27)$$

Where, $e_{ids} = i_{ds}^s - \hat{i}_{ds}^s$ and $e_{iqs} = i_{qs}^s - \hat{i}_{qs}^s$

Thus for uniform asymptotic stability, the observer gain matrix K should be chosen such that the first term of the above equation is negative semi definite and the rest terms reduce to zero. So

$$\dot{\omega}_r = \lambda \delta \left(e_{ids} \hat{\lambda}_{qr}^s - e_{iqs} \hat{\lambda}_{dr}^s \right)$$

Or

$$\dot{\omega}_r = \lambda \delta \int \left(e_{ids} \hat{\lambda}_{qr}^s - e_{iqs} \hat{\lambda}_{dr}^s \right) dt \quad (28)$$

To ensure that the estimation error vanishes over time for any value of \hat{X} , the observer gain matrix K should be selected so that (A+KC) is asymptotically stable. Therefore, the observer gain matrix should be chosen so that all Eigen values of (A+KC) have negative real parts. The conventional procedure is to select the observer poles proportional to the motor poles. If the poles of the induction motor are given by P_m , the observer poles P_o are selected as:

$$P_o = k_p P_m \quad (29)$$

Where, $k_p \geq 1$ it can be achieved by defining the observer matrix K in a special form

$$K = \begin{bmatrix} k_1 I + k_2 J \\ k_3 I + k_4 J \end{bmatrix} = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \quad (30)$$

To determine the Eigen values of the matrix A,

$$\left| P_m I - A \right| = \begin{bmatrix} P_m + \gamma I & -\delta \left(\frac{1}{T_r} I - \omega_r J \right) \\ -\left(\frac{L_m}{T_r} \right) I & P_m + \left(\frac{1}{T_r} I - \omega_r J \right) \end{bmatrix} = 0 \quad (31)$$

To simplify the above equation, substitute

$$a = \gamma I, \quad b = \left(\frac{1}{T_r} I - \omega_r J \right) \quad \text{and} \quad c = \left(\frac{L_m}{T_r} \right) I$$

The characteristic equation of the matrix A is then

$$P_o^2 + (a+b)P_o + (a-\delta c)b = 0 \quad (32)$$

To determine the eigenvalues of the matrix (A + KC),

$$\left| P_o I - (A + KC) \right| = \begin{bmatrix} P_o + \gamma I - K_1 & -\delta \left(\frac{1}{T_r} I - \omega_r J \right) \\ -\left(\frac{L_m}{T_r} I + K_2 \right) & P_o + \left(\frac{1}{T_r} I - \omega_r J \right) \end{bmatrix} = 0 \quad (33)$$

So, the characteristic equation is

$$P_o^2 + (a+b-K_1)P_o + \left[(a-\delta c) - (K_1 + \delta K_2) \right] b = 0 \quad (34)$$

Substituting equation (29) in equation (34)

$$k_p^2 P_m^2 + (a+b-K_1)k_p P_m + \left[(a-\delta c) - (K_1 + \delta K_2) \right] b = 0 \quad (35)$$

Comparing (35) with (32) and replacing ω_r by $\hat{\omega}_r$, the observer gain matrix is given by

$$K_1 = \left(1 - k_p \right) (a+b) = \left(1 - k_p \right) \left[\left(\gamma + \frac{1}{T_r} \right) I - J \hat{\omega}_r \right] \quad (36)$$

$$K_2 = \frac{\left(1 - k_p^2 \right)}{\delta} (a - \delta c) - \frac{\left(1 - k_p \right)}{\delta} (a+b) = \left[\frac{\left(1 - k_p^2 \right)}{\delta} \left(\gamma - \frac{\delta L_m}{T_r} \right) - \frac{\left(1 - k_p \right)}{\delta} \left(\gamma + \frac{1}{T_r} \right) \right] I + \frac{\left(1 - k_p \right)}{\delta} \hat{\omega}_r J \quad (37)$$

Comparing (36) and (37) with (30)

$$\left. \begin{aligned} k_1 &= \left(1 - k_p \right) \left(\gamma + \frac{1}{T_r} \right) \\ k_2 &= \left(k_p - 1 \right) \hat{\omega}_r \\ k_3 &= \frac{\left(1 - k_p^2 \right)}{\delta} \left(\gamma - \frac{L_m \delta}{T_r} \right) - \frac{\left(1 - k_p \right)}{\delta} \left(\gamma + \frac{1}{T_r} \right) \\ k_4 &= \frac{\left(1 - k_p \right)}{\delta} \hat{\omega}_r \end{aligned} \right\} \quad (38)$$

The Figure 6 is the block diagram representation of sensor less indirect vector control of induction motor.

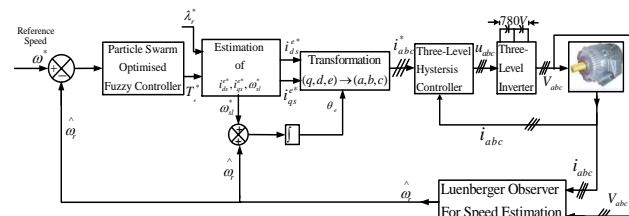


Figure 6: Block Diagram of Sensor less Vector Control of IM

III. FUZZY LOGIC CONTROLLER

The fuzzy logic controllers are proposed to be Mamdani type fuzzy logic controller having five blocks namely normalizer, fuzzifier, inference mechanism, de-fuzzifier, and de-normalizer as shown in Figure 7. [19].

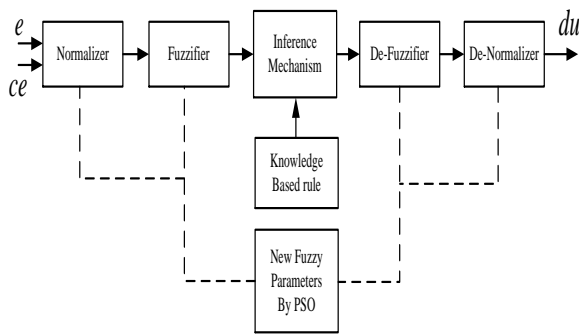


Figure 7: Block Diagram of Fuzzy Logic Controller

A. Normalizer / De-normalizer

In closed loop control system the use of error (e) and the change in error (ce) as controller input is a universal approach. Therefore the fuzzy logic controller has two inputs, error and change in error (e, ce) and one output du as shown in Figure 7. Two normalization parameters (n_1, n_2) for inputs (e, ce) and one de-normalization parameter (n_3) for output du are defined. In normalization process the input variables are scaled in the range of $(-1, +1)$ and in de-normalization process the output values of fuzzy controller are converted to a value depending on the terminal control element. The determination of normalization and de-normalization parameters of fuzzy controller is important for system stability.

B. Fuzzifier / De-fuzzifier

The fuzzifier processes the crisp input values (e, ce) and convert them into fuzzy values. Also the fuzzy values obtained in fuzzy inference mechanism are converted to crisp output du value by a de-fuzzifier. Here a triangular fuzzy membership function is defined for each input and output values by seven clusters. For seven clusters in the membership functions, seven linguistic variables are defined as: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Figure 8 shows the membership functions used to fuzzify two input values (e, ce) and de-fuzzify output du of the realized fuzzy controller. The peak or bottom points of the membership functions to be tuned are a_1 and a_2 for error e , b_1 and b_2 for change in error ce and c_1 and c_2 for output du . Therefore the design of fuzzy controller require the optimization of nine parameters ($n_1, n_2, n_3, a_1, a_2, b_1, b_2, c_1, c_2$).

The centre of gravity or centroid method is used for de-fuzzification. As a result the control increment is obtained by the equation [20]

$$du = \frac{\sum_{i=1}^m d_i A(\mu_i)}{\sum_{i=1}^m A(\mu_i)} \quad (39)$$

Here d_i is the distance between i^{th} fuzzy set and the centre, $A(\mu_i)$ is area value of i^{th} fuzzy set.

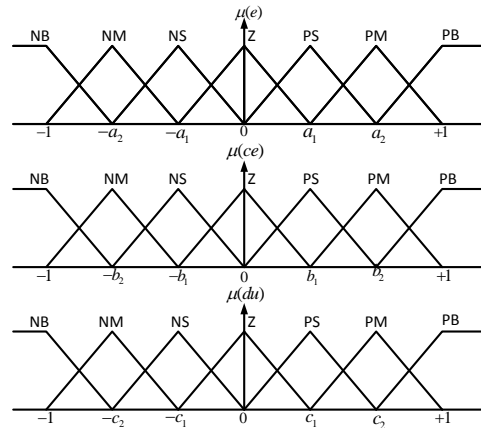


Figure 8: Membership functions of Inputs and output

C. Knowledge base and Inference mechanism

The rule definition is subjective and based on expert's knowledge and experiences. It establishes the relationship between outputs with inputs [21]. For the system with two inputs and seven membership functions in each leads to forty nine combination of these inputs, in which there are forty nine rules. The rules are like:

R1. If $e = NB$ and $ce = NB$ Then du is NB

Or

R2. If $e = NB$ and $ce = NM$ Then du is NB

Or

R49. If $e = PB$ and $ce = PB$ Then du is PB

The rules are represented by a matrix called matrix inference shown in Table 2. A feature of the rule base used is the symmetry across the diagonal.

Table 2: Fuzzy Linguistic Rule Table

		ERROR (e)						
		NB	NM	NS	Z	PS	PM	PB
CHANGE IN ERROR (ce)	NB	NB	NB	NB	NB	NM	NS	Z
	NM	NB	NB	NB	NM	NS	Z	PS
	NS	NB	NB	NM	NS	Z	PS	PM
	Z	NB	NM	NS	Z	PS	PM	PB
	PS	NM	NS	Z	PS	PM	PB	PB
	PM	NS	Z	PS	PM	PB	PB	PB
	PB	Z	PS	PM	PB	PB	PB	PB

This feature occurs in systems where the physical behaviours of the system exhibit symmetry, which is consistent in case of speed control of induction motor. The developed fuzzy logic uses the inference method for each rule given by the relation

$$\mu_i(du) = \min(\mu_i(e), \mu_i(ce)) \quad i = 1, 2, \dots, 49 \quad (40)$$

and the resulting membership function is given by

$$\mu(du) = \max(\mu_1(du), \mu_2(du), \dots, \mu_{49}(du)) \quad (41)$$

IV. FUZZY LOGIC CONTROLLER BASED ON PARTICLE SWARM OPTIMIZATION

A. Particle Swarm Optimization

Particle Swarm Optimization is a population based stochastic optimization technique, inspired by social behaviour of bird flocking or fish schooling. In PSO system the individuals called particles, fly around in a multidimensional search space and change their position with time. During its flight, each particle adjusts its position according to its own experience and according to the experience of neighbouring particle. The position or value corresponding to its own experiences called P_{best} and corresponding to the experience of neighbouring particle is called G_{best} . The search for the optimal position advances as the particles' velocities and positions are updated. In very iteration, the fitness of each particle's position is calculated using a predefined objective (fitness) function and the velocity of each particle is updated using the P_{best} and G_{best} , which were previously defined. The velocity of i^{th} particle can be modified by the following equation.

$$v_i(n+1) = \omega(n) \times v_i(n) + C_1 \times r_1(n) \times (P_{best_i} - x_i(n)) + C_2 \times r_2(n) \times (G_{best} - x_i(n)) \quad (42)$$

where $v_i(n)$ is the velocity of i^{th} particle at iteration n , $v_i(n+1)$ is the velocity of i^{th} particle at iteration $(n+1)$, $r_1(n)$ and $r_2(n)$ are random numbers with uniform distribution in the interval $[0, 1]$, $\omega(n)$ is the momentum or inertial weight constant given by [22]

$$\omega(n) = \omega_{max} - \left\{ \frac{\omega_{max} - \omega_{min}}{n_{max}} \right\} n \quad (43)$$

Here n_{max} is the maximum number of iteration, ω_{max} and ω_{min} are the maximum and minimum weights respectively. Appropriate values of ω_{max} and ω_{min} are 0.9 and 0.4 respectively [23]. The values C_1 and C_2 are two positive constants represent the social and cognitive accelerations for the P_{best} and G_{best} positions, respectively. Varying these parameters has the effect of varying the strength of the pull towards the two bests. Values of $C_1 = C_2 = 0$ mean that both the cognitive and social accelerations are absent, and particles keep moving at their current speed until they hit a boundary of the search space (assuming no inertia) [24]. With $C_1 > 0$ and $C_2 = 0$, each particle searches for the best position in its neighbourhood, and replaces the current best position if the new position is better [24]. However, with $C_2 > 0$ and $C_1 = 0$, the entire swarm is attracted to a single point, G_{best} . Furthermore, having $C_1 \gg C_2$ causes each particle to be attracted to its own personal best position to a very high extent, resulting in excessive wandering. On the other hand, $C_2 \gg C_1$ results in particles being more strongly attracted to the global best position, thus causing particles to rush prematurely towards optima [24]. It is demonstrated that the particle swarm is only stable and guaranteed to converge to a stable equilibrium point if the following conditions are satisfied [25].

$$0 < (C_1 + C_2) < 4 \quad (44)$$

$$\frac{(C_1 + C_2)}{2} - 1 < \omega(n) < 1 \quad (45)$$

However, whether or not this point is actually the global minimum cannot be guaranteed, and its acceptability as a solution should be verified. The position of i^{th} particle at iteration n is $x_i(n)$. The modified position at iteration $(n + 1)$ is given by

$$x_i(n+1) = x_i(n) + v_i(n+1) \quad (46)$$

B. Optimization of Fuzzy Controller

The Particle Swarm Optimization is applied to automate and optimize the fuzzy controller design process. The normalization parameters, de-normalization parameter and the parameters of the membership functions are optimized by optimizing a properly defined objective or fitness function [26], [27]. There are two fuzzy logic controllers used in the proposed system; one as speed controller and other is in speed estimator. The fuzzy logic controller used as speed controller has two inputs, error e_c and change in error ce_c , which are given by

$$e_c(t) = (\omega_r^* - \omega_r) \quad (47)$$

$$ce_c(t) = e_c(t) - e_c(t+1) \quad (48)$$

The set of parameters to be optimized this fuzzy logic controller are $(n_{c1}, n_{c2}, n_{c3}, a_{c1}, a_{c2}, b_{c1}, b_{c2}, c_{c1}, c_{c2})$. In the context of optimization of such fuzzy logic controller, our goal is to have a speed response with a short rise time, small overshoot and near zero steady state error. In this respect a multiple objective function is defined as

$$F_c = \int_0^{t_1} |e_c| dt + \int_{t_1}^t |e_c| dt \quad (49)$$

Where, the first term is the measure of fast dynamic response and the second term is the measure of steady state error. The fuzzy logic controller used in speed estimator has two inputs, error e_e and change in error ce_e which are given by

$$e_e(t) = \left(e_{ids} \hat{\lambda}_{qr}^s - e_{iqs} \hat{\lambda}_{dr}^s \right) \quad (50)$$

$$ce_e(t) = e_e(t) - e_e(t-1) \quad (51)$$

The set of parameters to be optimized this fuzzy logic controller are $(n_{e1}, n_{e2}, n_{e3}, a_{e1}, a_{e2}, b_{e1}, b_{e2}, c_{e1}, c_{e2})$. In the context of optimization of such fuzzy logic controller, our goal is to have estimated speed $\hat{\omega}_r$ exactly equal to actual motor speed ω_r . In this respect an objective function is defined as

$$F_e = \int_0^t (\omega_r - \hat{\omega}_r) dt \quad (52)$$

Thus the purpose of PSO algorithm is to minimize the objective function. The PSO based approach algorithm to find the minimum value of objective function is as shown in Figure 9.

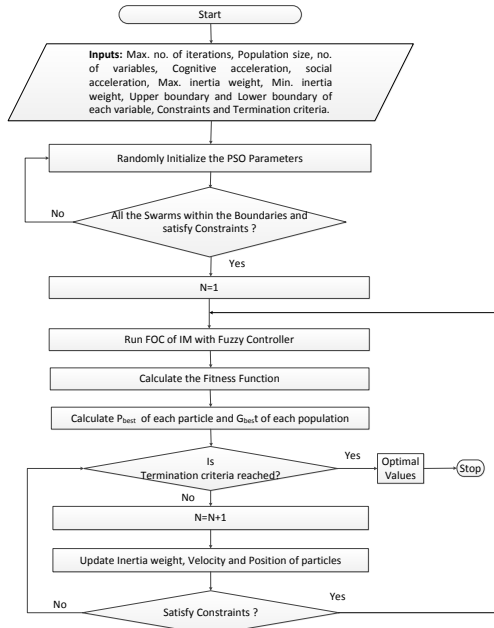


Figure 9: Flowchart of the PSO algorithm

V. SIMULATION RESULTS AND DISCUSSION

A complete simulation model for vector controlled induction motor drive of the proposed scheme is developed using MATLAB/ SIMULINK. The motor parameters are in Table 3. First of all the fuzzy logic controller used as speed controller is optimized by PSO without considering the speed observer i.e. taking actual motor speed ω_r as feedback to the speed controller.

Table 3: Induction Motor Parameters

Parameter	Symbol	Value
Rated Power	P_{rated}	50 Hp
Rated Voltage	V	480 Volt
Rated Frequency	F	50 Hz
Pair of Poles	P	2
Stator Resistance	R_s	0.087 Ω
Rotor Resistance	R_r	0.22 Ω
Stator Inductance	L_s	0.8 mH
Rotor Inductance	L_r	0.8 mH
Mutual Inductance	L_m	34.7 mH
Moment of Inertia	J	1.662 kg.m ²

Figure 10 shows the scores of the fitness function corresponding to different generation in PSO for fuzzy logic controller used as speed controller. The PSO is terminated at 41 generations as the termination criteria reached. Similarly the fuzzy logic controller used in speed estimator is optimized by PSO considering the speed observer taking the estimated speed $\hat{\omega}_r$ as feedback to the speed controller. Figure 11 shows the scores of the fitness function corresponding to different generation in PSO for fuzzy logic controller used in speed estimator. In this case

the PSO is terminated at 49 generations as the termination criteria reached. The termination criteria of the algorithm is either the maximum generations reached or the weighted average change in the fitness function value over Stall generations is less than function tolerance. The optimized values of all the parameters are in Table 4. The particle swarm optimized input and output membership functions for both the fuzzy logic controllers are in Figure 12 and Figure 13.

Table 4: Optimized Values of Parameters

n_{c1}	6.366601440561e-04	n_{e1}	0.204230794614153
n_{c2}	0.538296104367619	n_{e2}	0:372717279683677
n_{c3}	3.474559679239366	n_{e3}	0:958405412315019
a_{c1}	0.035691605541264	a_{e1}	0:479261688178227
a_{c2}	0.700814506194795	a_{e2}	0:904836175738316
b_{c1}	0.053227790847909	b_{e1}	0:142646332443648
b_{c2}	0.793644125939212	b_{e2}	0:768274946372518
c_{c1}	0.050531304972714	c_{e1}	0:460490418982960
c_{c2}	0.695057495825054	c_{e2}	0:533696655366175

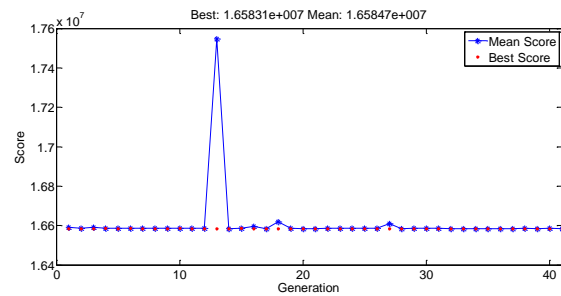


Figure 10: Fitness Score vs Generation for Speed Controller

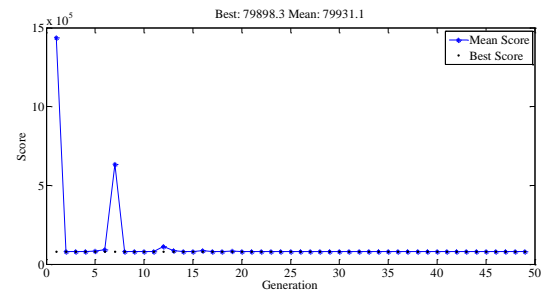


Figure 11: Fitness Score vs Generation for Speed Estimator

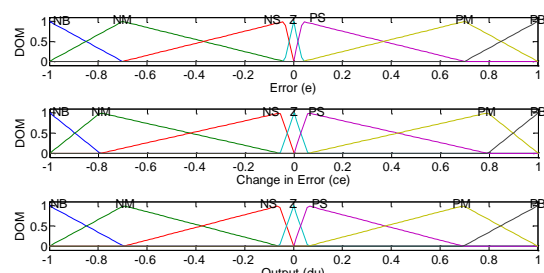


Figure 12: PS Optimized Input Output Membership Functions of Speed Controller

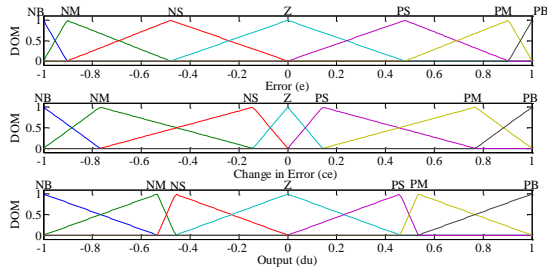


Figure 13: PS Optimized Input Output Membership Functions of Speed Estimator

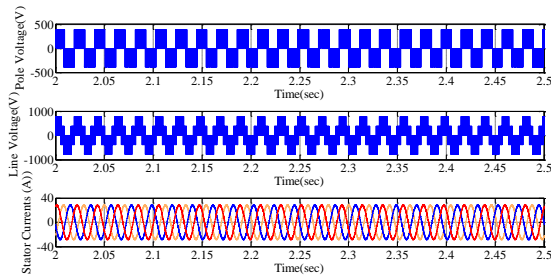


Figure 14: Inverter Voltages and Currents

Figure 14 shows the pole voltage, line voltage and the line currents (stator line currents) of the three-level inverter under steady state condition. The line currents are sinusoidal with almost negligible ripple. Figure 15 shows the speed tracking performance of the motor following a trapezoidal speed reference. The speed tracking experiment is on no load condition. The motor speed almost tracks the reference speed in both the direction.

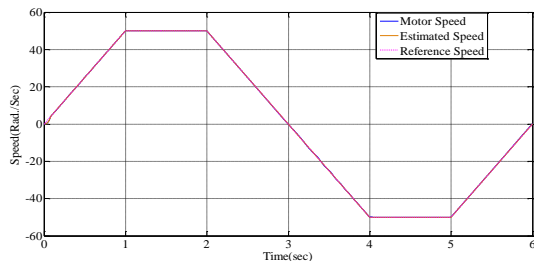


Figure 15: Trapezoidal Speed Tracking

Figure 16 show the performance of motor for the constant reference speed of 120 rad/sec with constant load torque 100 Nm. Figure 17 show the performance of motor when the load torque is suddenly changed from 50 Nm to 150 Nm and then from 150 Nm to 80 Nm at constant reference speed 120 rad/sec. Figure 18 show the performance of the motor when the reference speed is a step speed and the load torque is constant at 20 Nm.

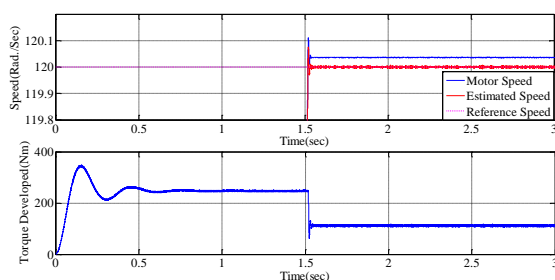


Figure 16: Performance under constant speed and constant torque

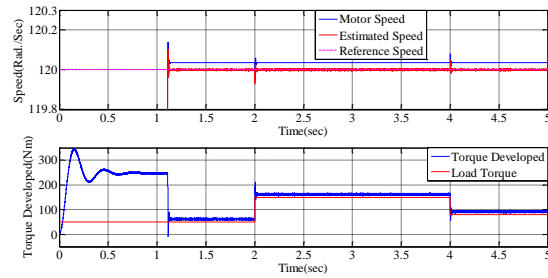


Figure 17: Performance under constant speed and variable torque

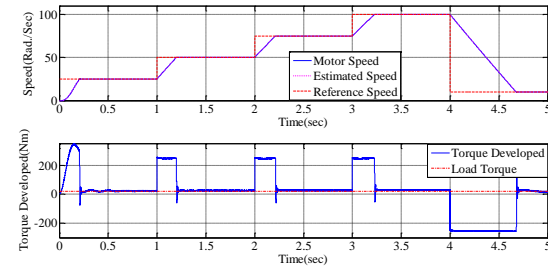


Figure 18: Performance under step speed and constant torque

VI. CONCLUSIONS

The two optimal fuzzy logic controllers have been designed off-line using techniques of Particle Swarm Optimization for sensor less vector control of multilevel inverter fed Induction motor. It achieves good estimation of motor speed, pursuit of reference speed, starting without overshoot and rapid rejection of disturbances with a low drop-out speed. From simulation results, it testifies that this method is not only robust, but also can improve dynamic performance of the system.

REFERENCES

- [1] P. Vas, (1990), "Vector control of AC machines" New York: Clarendon.
- [2] B. K. Bose, (2002), "Modern Power Electronics and AC Drives" Prentice-Hill PTR Companies Inc. Upper Saddle River, NJ 07458.
- [3] R. Krishnan, (2003), "Electric motor drives modelling, analysis and control" New Delhi: PHI Pvt. Ltd.
- [4] J. Rodriguez, J.-S. Lai, and F. Z. Peng, (2002), "Multilevel inverters: A survey of topologies, controls, and applications" *IEEE Transaction on Industrial Electronics*, vol. 49, no. 4, August.
- [5] X. Yuan and I. Barbi, (1999) "A new diode clamping multilevel inverter" *IEEE, conference*, pp. 495-501.
- [6] B. K. Bose, (1993), "Power electronics and motion control-technology status and recent trends" *IEEE Transaction on Industry Applications*, vol. 29, pp. 902-909.
- [7] C. Park and W. Kwon, (2004), "Simple and robust sensor less vector control of induction motor using stator current based MRAC" *Electric Power Systems Research*, vol. 71, pp. 257-266.
- [8] O. Aydogdu and R. Akkaya, (2011), "An effective real coded GA based fuzzy controller for speed control of a BLDC motor without speed sensor" *Turkish Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 413-430.
- [9] M. R. Douiri, M. Cherkaoui, and A. Essadki, (2012), "Genetic algorithms based fuzzy speed controllers for indirect field oriented control of induction motor drive" *International Journal of Circuits, Systems and Signal Processing*, vol. 6, no. 1, pp. 21-28.
- [10] J. Kennedy and R. C. Eberhart, (1995), "Particle swarm optimization" *Proceedings of the IEEE Int. Conf. on Neural Networks*, Perth, Australia, pp. 1942-1948.
- [11] M. Kazmierkowski and L. Malesani, (1998), "Current control techniques for three-phase voltage-source PWM converters: A survey" *IEEE Transaction on Ind. Electronics*, vol. 45, no. 5, pp. 691-703.

- [12] A. Shukla, A. Ghosh and A. Joshi, (2011), "Hysteresis modulation of multilevel inverters" *IEEE Transaction on Power Electronics*, vol. 26, no. 5, pp. 1396–1408.
- [13] J. Zeng, C. Yu., Q. Qi., Z. Yan., Y. Ni., B. Zhang, S. Chen, and F. F. Wu., (2004), "A novel hysteresis current control for active power filter with constant frequency" *Electric Power Systems Research*, vol. 68, pp. 75–82.
- [14] M. Milosevic, (2003), "Hysteresis current control in three-phase voltage source inverter" *Technical Report, Zurich*.
- [15] S. Srikanthan, M. Mishra, and R. Rao, (2009), "Improved hysteresis current control of three-level inverter for distribution static compensator application" *IET Power Electronics*, vol. 2, no. 5, pp. 517–526.
- [16] S. K. Sahu, D. D. Neema, and T. V. Dixit, (2013), "Indirect vector control of induction motor using ANN estimator and ANFIS controller" *International Journal of Computer Applications*, vol. 66, no. 14, March.
- [17] J. Gacho and M. Zalman, (2010), "IM based speed servo drive with Luenberger observer" *Journal of Electrical Engineering*, vol. 61, no. 3, pp. 149–156.
- [18] J. Maes and J. A. Melkebeek, (2000), "Speed-sensorless direct torque control of induction motors using an adaptive flux observer" *IEEE Transaction on Industry Applications*, vol. 36, no. 3, pp. 778–785.
- [19] M. N. Uddin, T. S. Radwan, and M. A. Rahman, (2002), "Performances of fuzzy-logic-based indirect vector control for induction motor drive" *IEEE Transaction on Industry Applications*, vol. 38, no. 5, pp. 1219–1225, Sept./Oct.
- [20] C. C. Lee, (1990), "Fuzzy Logic in Control Systems: Fuzzy Logic controller Part 1 and Part 2" New Delhi: *IEEE*.
- [21] Z. Zhao, M. Tomizuka, and S. Isaka, (1993), "Fuzzy gain scheduling of PID controllers" *IEEE Transactions on Systems, Man and Cybernetics* vol. 23, no. 5, pp. 1392–1398, Sept./Oct.
- [22] Y. Bekakra and D. B. attous, (2011), "Sensorless speed based on MRAS with tuning of ip speed controller in FOC of induction motor drive using PSO" *World Academy of Science, Engineering and Technology*, vol. 60, pp. 1550–1555.
- [23] R. Eberhar, and Y. Shi, (2000) "Comparing inertial weights and constriction factor in particle swarm optimization" *Proceeding of the International Congress on Evaluationing Computation*, pp. 84–88.
- [24] E. AP, (2005), "Fundamentals of Computational Swarm Intelligence" *John Wiley Sons*.
- [25] R. Perez and K. Behdinin, (2007), "Particle swarm approach for structural design optimization" *Computers and Structures*, vol. 85, pp. 1579–88.
- [26] V. Donescu, D. Neacsu, G. Griva, and F. Profumo, (1996) "A systematic design method for fuzzy controller for brushless dc motor drives" *Proc. of the 27th. IEEE Annual Power Electronics Specialists Conference*, pp. 689–694, Baveno, Italy,
- [27] F. D. S. Cardoso, J. F. Martins, and V. F. Pires, (1998) "A comparative study of a pi, neural network and fuzzy genetic approach controllers for an AC drive, IEEE 5th International Workshop on Advanced Motion Control AMC, pp. 375–380, Coimbra.