Shape Optimization of the Permanent Magnet Pole of the Brushless DC Motor by FEM

Reza Namazi Nanekaran*1, Esmail Fallah Chulabi2

1, 2Faculty of Engineering, University of Guilan, Rasht, Iran

*1namazi.reza67@gmail.com; 2fallah_e@guilan.ac.ir

Abstract-**The use of the Brushless DC Motors (BLDC) for industrial and transportation applications has been developed increasingly. These motors have many advantages such as good efficiency and high torque to weight ratio but the torque pulsation is the most concern of these motors. Pulsating torque can be considered as the result of two components, torque ripple and cogging torque. Torque ripple is the result of the overlap between the feeding time of the two adjacent phases while cogging torque is the result of the variable magnetic reluctance. In this paper, changing two parameters including embrace (pole arc to pole pitch ratio) and offset (displacement of the pole arc center), cogging torque was calculated using Finite Element Method (FEM). An Artificial Neural Network (ANN) was used to interpolate between the limited results of the time-consuming Finite Element Analysis (FEA). Two optimization methods including Particle Swarm Optimization (PSO) and Imperialist Competitive Algorithm (ICA) were used to find the best design parameters in which the torque ripple reaches the minimum. Finally, the optimum designs were validated by FEA.**

***Keywords*- BLDC; FEM; ANN; PSO; ICA**

I. INTRODUCTION

Permanent-magnet brushless DC motors, due to high weight to Torque ratio, high performance and reliability, are widely used in industrial applications such as CNC machines, compressors, micro gas turbine generators and hybrid electric vehicles [1-6]. However, cogging torque is the main obstacle in development of the application of these types of motors. Cogging torque is caused by reluctance variations in the path of the flux produced by permanent-magnet poles (because of stator slots), and results in output torque pulsation and noise generation. Pole shape of the permanent magnet is one of the main factors in generating of this annoying torque.

In recent years, different techniques have been introduced to reduce the torque ripple caused by cogging torque in these classes of motors. In [7], different designs (configuration) are analyzed by finite element method and the best design is chosen in such a way that the phenomenon of slight demagnetization is minimized. In [8], a 6-slot 4-pole PMSM (Permanent Magnet Synchronous Motor) motor which is used in an air compressor is studied, and the pole shape is optimized from the MPE (Magnetic Pole Embrace), MPB (Magnetic Pole Bridge) and MPE (Magnetic Pole Eccentricity) point of view. The effects of these parameters on the average torque and torque ripple are studied using finite element method while the best case is selected from the tables. In [9], considering the number of poles, rotor dimension and the thickness of permanent magnet, magnetic field distribution in the air gap is calculated by ANSOFT. Using FFT tool box in MATLAB, harmonic content of the air gap flux density are analyzed for different cases, and the best design is chosen for sinusoidal flux density.

In this paper, the dynamic behavior of the motor with respect to the offset and embrace of the PM pole was investigated, using FEM software. Due to the time-consuming analysis of the FEM software, the obtained value of maximum torque, average torque and torque ripple are used to train an artificial neural network in such a way that for all possible values of offset and embrace the value of the maximum torque, average torque and torque ripple can be estimated without the FEM software. In Continue, the optimization is performed by using PSO and Imperialist ICA. Finally the optimized results are verified by finite element analysis.

II. GEOMETRY AND SPECIFICATIONS OF THE MOTOR

The geometry of the brushless DC motor is shown in Fig. 1. The number of phases, slots, and poles are 3, 36 and 6 respectively. The B-H curve of steel metal M19-24G used for the stator and rotor laminations is shown in Fig. 2. The B-H curve of the permanent magnet material XG196/96 used in the rotor pole is shown in Fig. 3. Specifications and parameters of the brushless DC motor are given in Table 1.

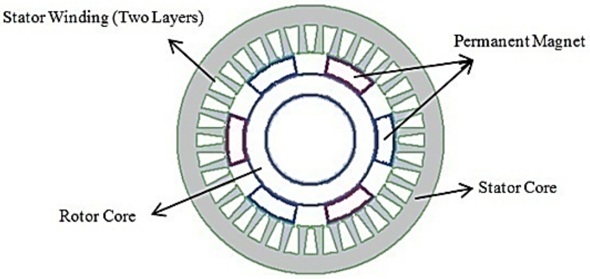

Fig. 1 Brushless DC motor with 6 poles and 36 slots

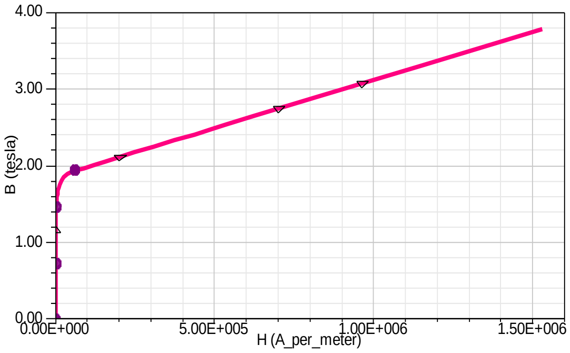

Fig. 2 B-H curve of steel metal M19-24G

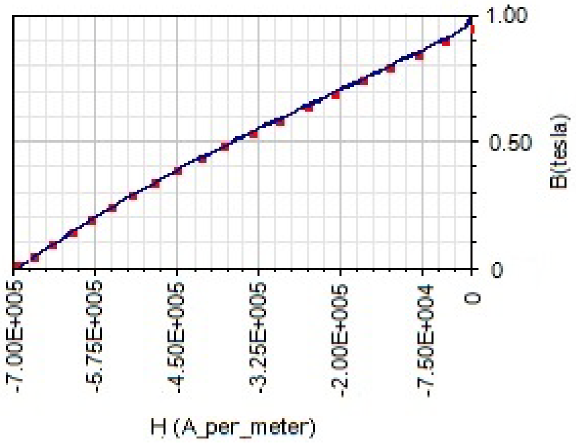

Fig. 3 B-H curve of permanent magnet material XG196/96

TABLE 1 SPECIFICATIONS OF THE BRUSHLESS DC MOTOR [10]

Stator	
96mm	Inner diameter
148mm	Outer diameter
142mm	length
36	No. of slots
0.95	Stack factor
2-layer	Winding type
6	No. of winding parallel branch
1	No. of wires of each conductor
5	Coil pitch
20	No. of conductor of each slot
3	No. of phases
Rotor	
50mm	Inner diameter
94mm	Outer diameter
142mm	length
10mm	Magnet thickness
0.95	Stack factor
6	No. of poles
75°C	Ambient temperature

III. DESIGN FACTORS AND SPECS

The pole arc of the permanent magnet β, and rotor pole pitch γ, are shown in Fig. 4, also the inner radius R_1, and outer radius R_2, of the permanent-magnet pole are depicted in Fig. 5. The ratio β/γ, and difference R_2-R_1 are called embrace and offset, respectively. In this study, the value of the embrace varies from 0.4 to 1, and the value of the offset varies from 0 to 27.

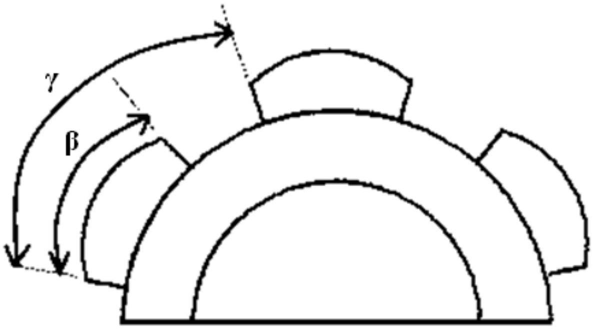

Fig. 4 The schematic of the magnetic pole arc and rotor pole pitch

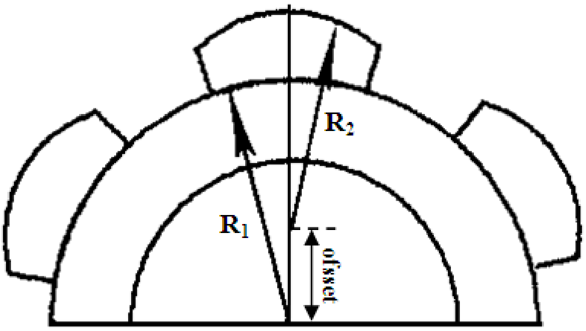

Fig. 5 A view of external and internal centers of the permanent-magnet pole

IV. OPTIMIZATION ALGORITHMS

A. Particle Swarm Optimization (PSO)

PSO is an optimization technique introduced by Eberhart and Kennedy in 1995, based on the social behaviors of birds flocking or fish schooling [10]. PSO algorithm is comprised of a group of components moving in a multidimensional search space with real values of possible solutions. PSO is easy to implement and has low computational complexity [11].

In PSO, each solution named individual or particle, is a bird in the search space. In other words, an individual is a point searching for an optimum position in the multidimensional search space. The initial values of the individuals (particle) are set randomly and the search for optimum solution starts. In each iteration, an individual determines its next position in search space with regards to two parameters; one, the best position (pbest) known for the particle, and the other, the best position known for the individuals of the group (swarm) called gbest. Considering pbest and gbest, Eqs. (1) and (2) are utilized for the next position of each individual:

$$V_{t+1}=W_t.V_t+C_1.\text{rand}.(\text{pbest-present}_t)+ C_2.\text{rand}.(\text{gbest-present}_t) \quad (1)$$

$$\text{present}_{t+1}=\text{present}_t+ V_{t+1} \quad (2)$$

In Eq. (1), C_1 and C_2 are learning parameters and *rand* is a random number between 0 and 1. Also, *$present_t$ and V_t* are the present position and velocity of the individual (particle). W_t is a parameter to keep balance between local and global search ability of the algorithm by controlling the effect of present velocity (V_t) on the next velocity [12].

B. Imperialist Competitive Algorithm (ICA)

Like the other evolutionary algorithm, this algorithm is also initialized with a number of random populations, each of them is called a "county". Some of the best elements of population are selected as imperialists and others are selected as colony. Depending on the imperialists' power, under a specific process to be described below, a number of colonies are controlled by them. The power of each empire depends on its imperialist and colony countries.

Forming initial empires, imperialistic competition begins. The empires which could not win the competition (i.e. could not increase their power or at least save their influence) will be removed from competition. Therefore, survival of an empire depends on its ability to take possession of colonies of other empires and controlling them. In this way, weak empires lose their power gradually and they will finally be eliminated. Moreover, empires to increase their power are supposed to develop their colonies. So the power of colonies gets closer to the empires and a convergence is occurred. This process continues until we have only one empire in the world with colonies which have very close position to the imperialist country. In order to start the algorithm, $N_{country}$ initial countries are created. N_{imp} of the best individuals of this population are selected as imperialists. Others (N_{col} of rest countries) form colonies, each of which belongs to an empire. Based on their powers, each imperialist can take possession of a number of the colonies. To do this, by knowing the costs of all imperialists, the normalized costs are considered as follows:

$$C_n=\max\{C_i\}-c_n \tag{3}$$

In Eq. (3), c_n is the cost of nth imperialist, $\max\{C_i\}$ is the most costly among imperialists, and C_n is the normalized cost of this imperialist. Each imperialist which has a higher cost (i.e. weaker imperialist) has a lower normalized cost too. Having normalized cost, relative normalized power of each imperialist is calculated as follows and colonies are divided among the imperialists based on their powers.

$$\mathrm{p}_n=\left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \tag{4}$$

From a different point of view, the normalized power of an imperialist is proportional to its colonies. Thus the initial number of colonies of an imperialist is given by:

$$N.C._{.n}=\mathrm{round}\{P_n.(N_{col})\} \tag{5}$$

In Eq. (5), $N.C._{.n}$ is the initial number of colonies of an empire, and N_{col} is the total number of the colonies in the population of the initial countries. Also, *round* is a function that rounds a float number to its nearest integer number. Considering N.C. for each empire, N.C. of initial colonies are randomly selected and given to n^{th} imperialist. Having initial states of all empires, imperialist competitive algorithm begins. Evolution process is in a loop, and continues until a stop condition is satisfied [13].

V. FINITE ELEMENT ANALYSIS

In order to obtain the torque ripple of the motor, a finite element based software named MAXWELL 2D, is utilized. First, the geometric and main parameters of the motor are set as given in Table 1, and the motor was modelled by means of the software, then design factors (offset and embrace) were changed by predefined steps in predefined intervals. The allowable intervals for embrace and offset were [0.4,1] and [0,27] respectively. In each case, the software was used to obtain the maximum torque and torque average. For the initial design, variation of the torque with time from starting to steady state, are shown in Fig. 6.

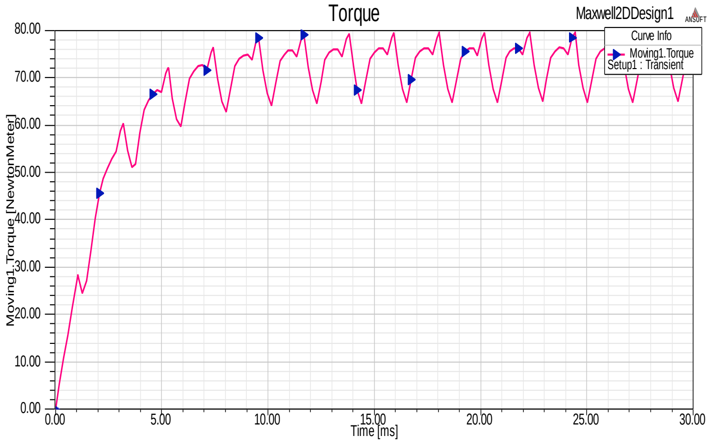

Fig. 6 Output torque for embrace=0.5 and offset=0

The torque ripple can be calculated by the maximum torque (T_{max}) and torque average (T_{ave}) as follow:

$$ripple=\frac{T_{\max}-T_{ave}}{T_{\max}} \tag{6}$$

Since FEM simulation is very time-consuming, we can obtain a limited number of results by FEM. On the other hand, the optimum point may be between these points which are obtained by FEM. So we need an interpolation tool to estimate the value of cost function between theses limited points. In this paper ANN was used for interpolation. Trained ANN can estimate the appropriate values of the torque ripple in terms of the offset and embrace in the allowable interval. The artificial neural network block-diagram, used for analysis, is depicted in Fig. 7. This artificial neural network is comprised of 2, 20 and 1 neurons, respectively, in its input, hidden and output layers. This class of neural network is called Multilayer Perceptron (MLP). The inputs of the neural network are the ratio of permanent magnet pole arc to the rotor pole pitch (embrace), and the difference of inner and outer centers of the permanent magnet (offset). The output of neural network is the torque ripple. Back propagation algorithm was used to train the artificial neural network. As it is seen in Fig. 8, after training the neural network, a maximum error of 1.52% was observed between the output of the ANN and its desired value for the test data.

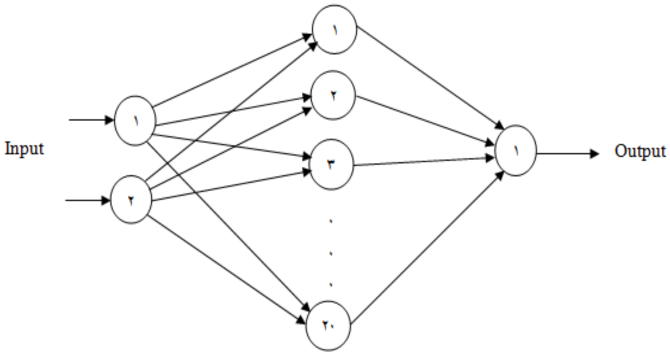

Fig. 7 The neural network with 20 neurons in hidden layer

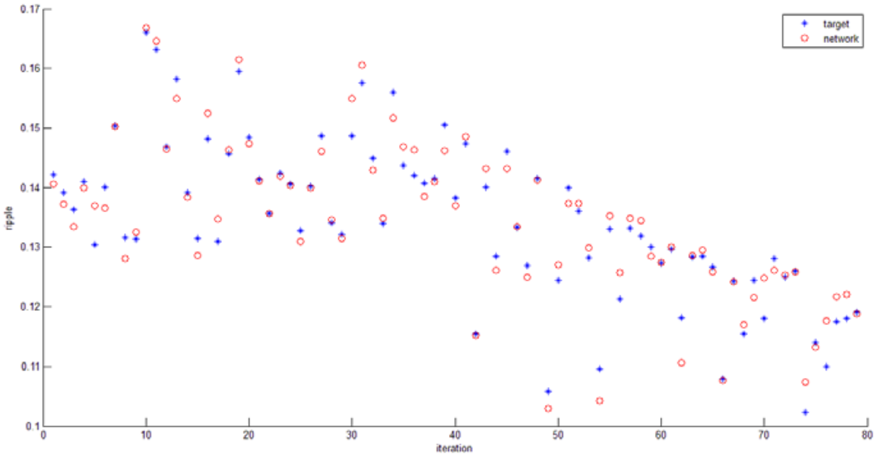

Fig. 8 Comparison of the expected value and the response of the network

VI. OPTIMIZATION RESULTS

After training the neural network, a mathematical function between input and output are presented by the neural network. This function was utilized in Particle Swarm Optimization (PSO) and Imperialist Competitive Algorithms (ICA). For both algorithms the diagram of cost function versus iterations was shown in Fig. 9.

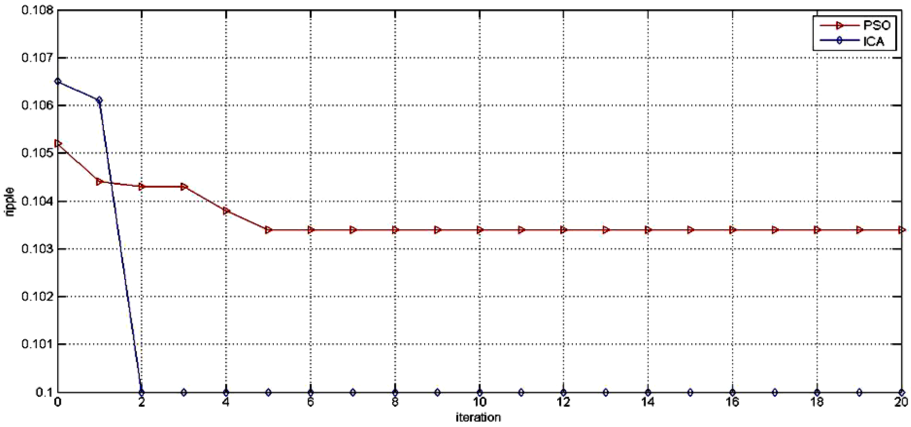

Fig. 9 Decrement of cost function with iterations number

The optimization results are shown in Table 2. In order to verify the obtained solutions, the value of the torque ripple for optimum designs was calculated by the finite element software, and compared with the predicted values from the optimization algorithms. Results are summarized in Table 2. The torque-time diagram of the motor, for the optimum design, is shown in Fig. 10. In this case, the maximum and average torqueses are 84.9N.m and 76.1N.m, respectively.

TABLE 2 COMPARISON OF THE OPTIMIZATION ALGORITHM AND FINITE ELEMENT RESULTS WITH INITIAL VALUES

The optimized **Optimization algorithm**	**embrace**	**Offset**	**torque ripple**
Initial state	0.5	0	19.37%
PSO algorithm	0.6066	26.94	10.34%
ICA algorithm	0.6077	27	10%
finite element results	0.6066	26.94	10.32%

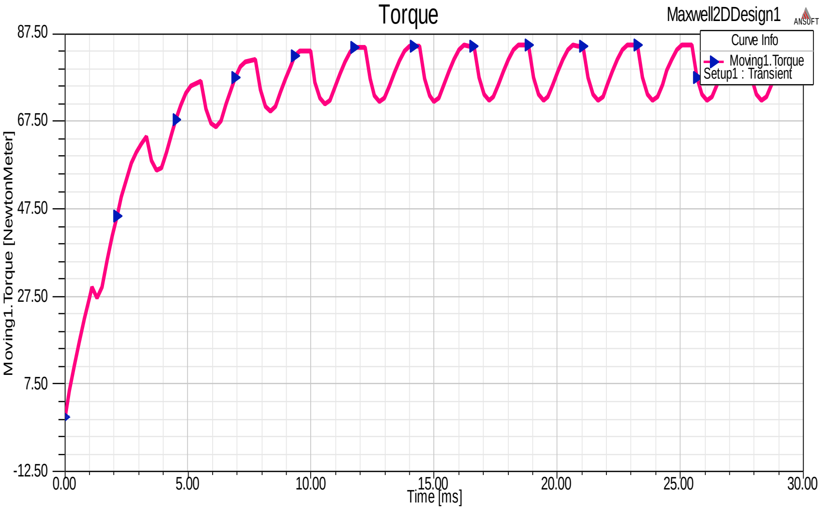

Fig. 10 Torque-time diagram of the optimum design

Mesh diagram of the Finite Element model is depicted in Fig. 11. Also, in this figure the air gap region is zoomed. As it can be seen, the number of the elements in the air gap is more because the field variation in this region is more. The Flux distribution for one pole is shown in Fig. 12. The flux lines of the stator and rotor pass through the air gap. As it can be seen in this figure, the magnetic flux density in teeth is more than the other regions. The distribution of the flux density is shown in Fig. 13. It shows that corner of the teeth goes into the magnetic saturation. The distribution of the flux density in the air gap is depicted in Fig. 14 and Fig. 15. As shown in these figures, after optimization the distribution of air gap flux density becomes more sinusoidal compared to initial design.

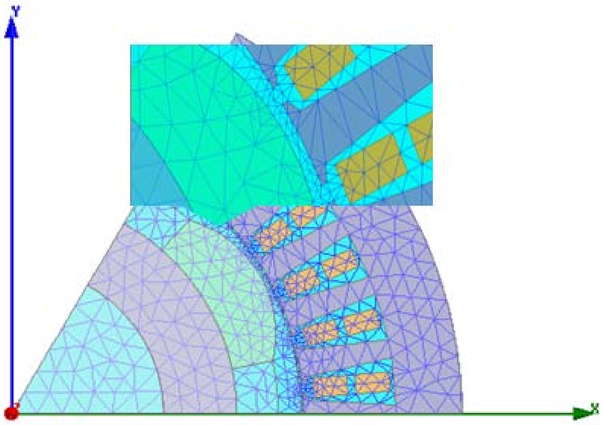

Fig. 11 Mesh diagram of the brushless DC motor

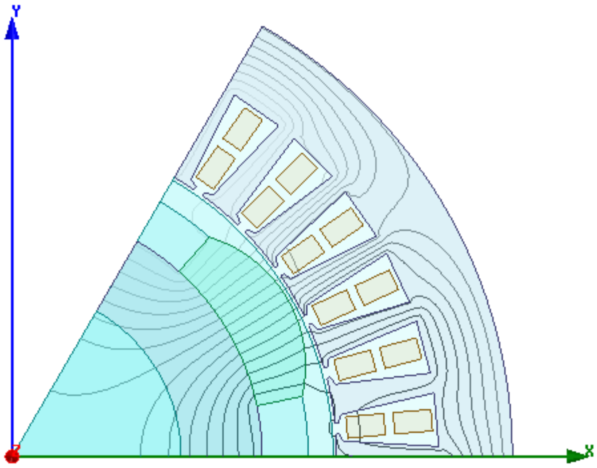

Fig. 12 The distribution flux lines brushless DC motor designed

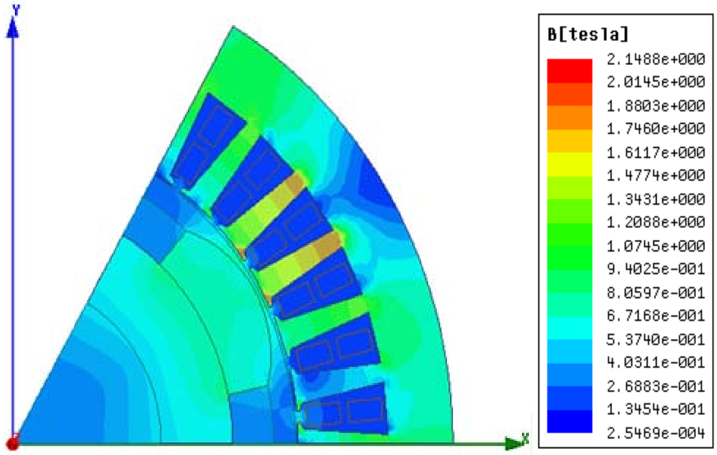

Fig. 13 The distribution flux density brushless DC motor designed

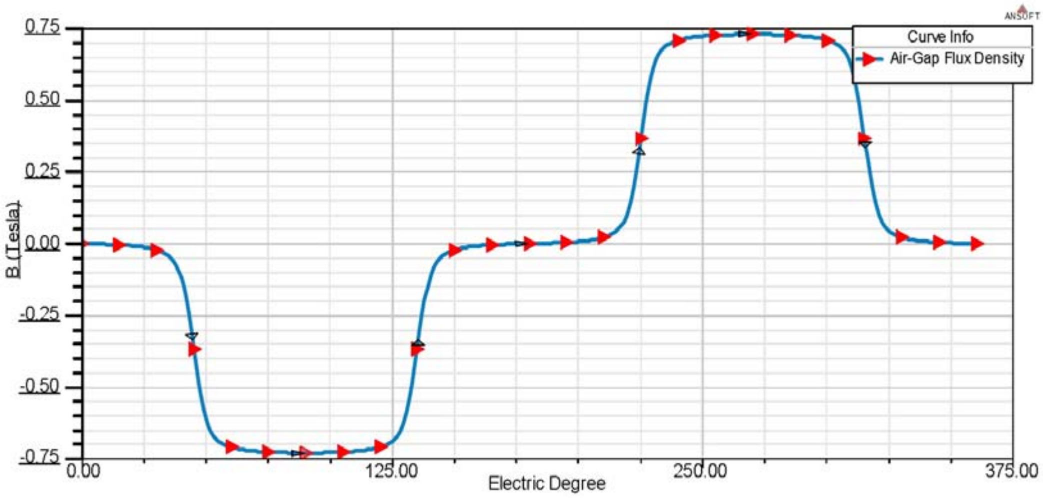

Fig. 14 Distribution of the air gap flux density before optimization

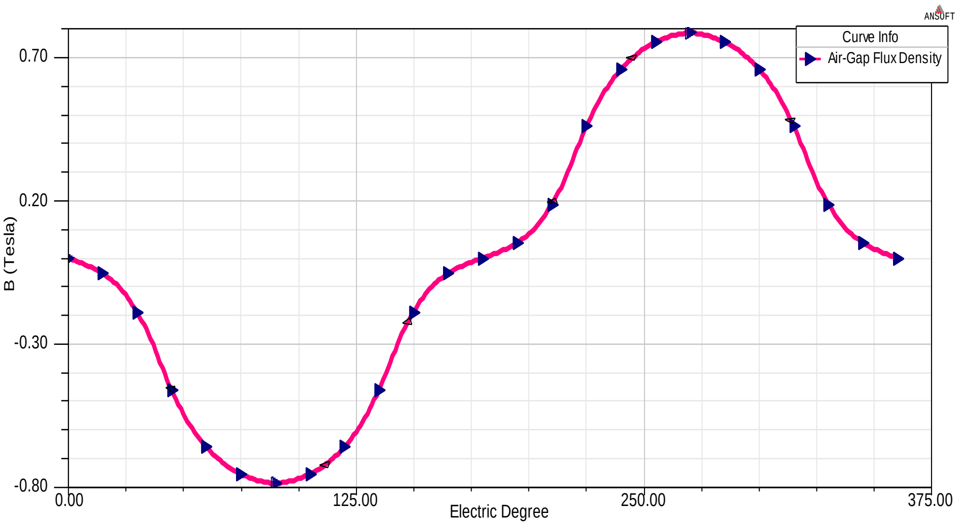

Fig. 15 Distribution of the air gap flux density after optimization

VII. CONCLUSIONS

In this paper, a successful optimization process of the permanent magnet pole of the brushless DC motor was presented. First, using Finite Element Analysis (FEA), Dynamic behavior of the brushless DC motor was investigated. Artificial Neural Network (ANN) was used to generalize the results obtained from FEA. Then Particle Swarm Optimization (PSO) and Imperialist Competitive Algorithm (ICA) was used to obtain the optimum values of the pole embrace and offset to achieve the minimum torque ripple. Results show that by this optimization process torque ripple reduced from 19.37% in initial design to 10.32% in optimum design. The overall approach which was used for design optimization is the contribution of this work; including the selection of torque ripple as cost function, the selection of pole embrace and offset as design parameters, which have a large influence on the cost function, using FEM for determination of cost function for a limited number of points and generalizing it by using the interpolation property of Neural Network and finally using PSO and ICA for cost minimization. As the results show, both methods, PSO and ICA, give good results. The difference is in the convergence time and the simplicity of implementation. ICA converges faster (as shown in Fig. 9) and the implementation of PSO is easier. In fact, the paper shows the capability of employing both of these two evolutionary algorithms.

REFERENCES

[1] Y. K. Chin and J. Soulard, “A permanent magnet synchronous motor for traction applications of electric vehicles,” in IEEE Int. Conf. Electric Machines and Drives, vol. 2, p. 103, June 2003.

[2] B. N. Chaudhari and B. G. Fernandes, “Performance of line start permanent magnet synchronous motor with single-phase supply system,” Proc. Inst. Electr. Eng., Electr. Power Appl., vol. 151, no. 1, pp. 83-90, Jan. 2004.

[3] K. Ogasawara, T. Murata, J. Tamura, and T. Tsuchiya, “High performance control of permanent magnet synchronous motor based on magnetic energy model by sliding mode control,” in Eur. Conf. Power Electronics and Applications, p. 10, Sept. 2005.

[4] B. Stumberger, G. Stumberger, M. Jesenik, V. Gorican, A. Hamler, and M. Trleps, “Power capability and flux-weakening performance of interior permanent magnet synchronous motor with multiple flux barriers,” in Proc. 12th Biennial IEEE Conf. Electromagnetic Field Computation, p. 419, 2006.

[5] G. Ho. Lee, S. I. Kim, J. P. Hong, and J. H. Bahn, “Torque Ripple Reduction of Interior Permanent Magnet Synchronous Motor Using Harmonic Injected Current,” *IEEE Transactions on Magnetics*, vol. 44, no. 6, pp. 1582-1585, June 2008.

[6] M. S. Islam, S. Mir, and T. Sebastian, “Issues in reducing the cogging torque of mass-produced permanent magnet brushless DC motor,” Conference Record of the Industry Applications, 38th IAS Annual Meeting, vol. 1, pp. 393-400, Oct. 2003.

[7] K. C. Kim, S. B. Lim, D. H. Koo, and J. Lee, “The shape design of permanent magnet for permanent magnet synchronous motor considering partial demagnetization,” *IEEE Trans. on Magnetics*, vol. 42, no. 10, Oct. 2006.

[8] P. Zheng, J. Zhao, J. Han, J. Wang, Z. Yao, and R. Liu, “Optimization of the magnetic pole shape of a permanent-magnet synchronous motor,” *IEEE Trans. on Magnetics*, vol. 43, no. 6, pp. 2531-2533, June 2007.

[9] Y. Li, J. Xing, T. Wang, and Y. Lu, “Programmable design of magnet shape for permanent-magnet synchronous motors with sinusoidal back EMF waveforms,” *IEEE Trans. on Magnetics*, vol. 44, no. 9, pp. 2163-2167, Sept. 2008.

[10] B. Singh and D. Goyal, “Computer Aided Design of Permanent Magnet Brushless DC Moto for Hybrid Electric Vehicle Application,” IEEE, 2006.

[11] R. Eberhart and J. Kennedy, “New Optimizer using particle swarm theory,” Micro Machine and Human Science, 1995.

[12] C. Kennedy, “The particle swarm Explosion, stability and convergence in a multideimentional complex space,” IEEE Tran. on Evolutionary Computation, 2002.

[13] E. Atashpaz-Gargari and C. Lucas, “Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic

Competition," IEEE Congress on Evolutionary computation (CEC), Singapore, pp. 4661-4667, 2007.

APPENDIX

The trained Neural-Network has the following representation that indicates the ripple as a non-linear function of embrace and offset. This function represented in MATLAB software language.

```
Function r=ripple(em,off)

n1=tansig([-4.5337 0.0094674;7.6029 -0.021612]*[em;off]+[2.7363;-3.4291]);
n2=tansig([-0.29934 6.0528;
 0.45923 8.8781;
 -6.7262 -8.3564;
 2.537 -4.9159;
 10.3343 -4.6462;
 -5.649 -1.2311;
 -4.8556 -4.4504;
 -7.4791 -1.9291;
 7.0197 2.0466;
 2.4178 -5.5138;
 -4.6168 -6.0106;
 -5.0035 6.5534;
 2.5534 5.598;
 5.3198 -1.8391;
 4.2415 4.6438;
 0.86231 -4.951;
 -5.8555 -1.3803;
 1.0766 6.0893;
 6.4968 -0.89976;
 -5.7616 -1.8305]*n1+[6.4902;
 -3.92;
 6.8105;
 -4.4577;
 -1.474;
 2.9078;
 0.98655;
 3.7408;
 0.523;
 -1.2774;
 0.28514;
 -1.1241;
 2.0247;
 3.6117;
 2.9251;
 6.5458;
 -4.7929;
 4.8622;
 7.2018;
 -6.5174]);
n3=1.3823+[0.54755 0.69906 -0.10665 -0.022054 -0.63462 -1.2411 -0.4111 0.99851 -0.57869 -0.030255 0.17169 -1.3517 -0.089065 0.79343 1.2751 -4.4906
0.9743 -0.20576 0.80486 -1.6096]*n2;
r=n3;
```