Comparison of Multi-objective Optimization Methods Applied to Electrical Machine Design

William R. Jensen(✉), Thang Q. Pham, and Shanelle N. Foster

Michigan State University, East Lansing, MI 48824, USA
{jensenwi,phamtha1,hogansha}@egr.msu.edu

Abstract. Electric machine design optimization is a growing topic of interest. Using a genetic algorithm for optimization, an efficient search of the design possibilities can be performed. Finite element analysis software includes optimization tools for machine designs. The exact operation of these genetic algorithms is unknown to the user, and the operating parameters of the built-in genetic algorithm are not completely configurable. Finite element analysis software in most cases allow for machine designers to link user-defined optimization algorithms. Given this option, the designer has the ability to select and tune an optimization algorithm to achieve diverse solutions that converge close to the true Pareto-optimal front. In this work, the benefit of user-defined optimization algorithms is demonstrated through optimizing design of a linear permanent magnet synchronous machine and evaluating the obtained Pareto-optimal fronts.

Keywords: Linear permanent magnet machine · Optimization · Finite Element Analysis · Genetic Algorithm · NSGA-II

1 Introduction

The use of electric machines is expanding across industries. In 2011, it was estimated that electric motor driven systems account for at least 43% of the global electricity consumption [1]. Several manufacturers offer a wide range of "off-of-the-shelf" electric machine solutions; however, many electric machines selected by end-users for their intended application are oversized, although motors are designed to operate most efficiently at a certain torque and speed. Some applications require customized electric machines to meet strict operating requirements. Performance, cost and reliability are only a few of the considerations that electric motor designers have to meet through optimization. Solving such complex multi-variable, multi-objective and robust optimization problems requires careful selection of the model and search algorithm, as well as strategic formulation of the optimization problem.

Modeling the performance of an electric machine involves a trade-off between accuracy and computation time. Electric machines can be modeled analytically

K. Deb et al. (Eds.): EMO 2019, LNCS 11411, pp. 719–730, 2019.
https://doi.org/10.1007/978-3-030-12598-1_57

or numerically. Finite elements models of electric machines provide accuracy to optimization techniques; however, its evaluation may negatively impact computation time. Some machine topologies require 3D finite elements analysis (FEA) for accuracy, which lengthens computation time. Hybrid modeling techniques, as well as simplified finite element (FE) models that exploit symmetry of the magnetic and electric circuit, are considered computationally efficient [2,3]. Surrogate models, in lieu of finite element models, were used in [4] to reduce computation time; however, development of accurate meta-models require significant training data and knowledge. Analytical machine models have been used with optimization algorithms to reduce total time to reach Pareto-optimal solutions [5,6]. While the machine performance is calculated quickly, accurate calculation of the losses and torque/force harmonics are difficult to include in analytical models. Several recently published techniques use very detailed analytical models for optimization [7–10]. This procedure allows evaluation of many cases without significantly increasing total simulation time. Only the final optimal design is evaluated with FEA to provide confidence; however, it is not certain that the accuracy translates to all designs. Any discrepancies may lead to the omission of non-dominated designs from the Pareto-optimal front.

Machines with complex geometries can have a large number of constraints. Many constraints can lead to discontinuities in the Pareto-optimal front. The selected algorithm must properly penalize constraint violations to avoid infeasible solutions but also reach the Pareto-optimal front. Deterministic and gradient-based optimization algorithms have been used for electric machine design; however, due to the nonlinearities in electric machines, these algorithms have difficulty finding the global optimum [2]. Unlike gradient-based algorithms, evolutionary multi-objective optimization algorithms are able to randomly search a large decision space and arrive at Pareto-optimal solutions quicker. The selection, crossover and mutation involved in evolutionary multi-objective optimization algorithms significantly affects the diversity of the solutions. Therefore, an evolutionary multi-objective optimization algorithm is best suited for optimizing an electric machine.

Over the last decade, optimization tools have been included with many FEA software packages. This allows machine designers to optimize the geometry completely within a single software package. Similarly, some FEA software packages link to other software packages that offer optimization tools. Built-in optimization algorithms are quite convenient but lack documentation detailing the algorithm as well as user control of the algorithmic operating parameters. Although a user-defined optimization algorithm may be more challenging to setup, it is beneficial for electric machine designers to have an understanding of the optimization algorithms. As stated best by the "No Free Lunch" theorem [2], there is no single optimization algorithm that is most efficient at solving every problem. A user-defined algorithm can provide more flexibility to select an efficient optimization algorithm that provides diversity in its solutions, better handle constraints to avoid infeasible solution and define stopping criteria to avoid unnecessary FEA simulations.

In this paper, it is demonstrated that a built-in optimization algorithm can lead to infeasible solutions, low diversity of solutions and solutions that are dominated by a user-defined algorithm. Here, a genetic algorithm (GA) included with a FEA software is compared to a user-defined GA for multi-objective optimization of a linear permanent magnet synchronous machine (LPMSM).

2 Problem Definition

A linear PMSM is similar to a rotating PMSM in its design and operation. PMSMs are advantageous because of their high torque density, and in the case of a linear PMSM, a high force density.

Typically, maximizing the produced force is an objective for electric machine design. One approach to increase average force is to shape the geometry of the stator and rotor [11]. However, modifying certain geometric parameters that increase average force lead to a significant increase in the harmonic content. Higher harmonics create a ripple which increases the noise and vibrations produced and decreases the operating lifetime of the machine.

Obtaining an optimal trade-off between average force and force ripple is a common challenge in machine design. Even small geometric changes produce significant changes in electromagnetic performance, which makes geometric optimization difficult.

The machine selected as the target for optimization is the LPMSM model presented in [12]. This machine was designed to achieve high acceleration by light-weighting the moving mass without significant reduction of the force produced. Although it's not an optimal design, it has been experimentally characterized and compared to its FEA model.

Force ripple reduction can be accomplished by shaping the stator teeth [13]. Instead of open slots, a trapezoidal wedge is added in the slot opening of the stator teeth. The distance from the center of the tooth to the end of the wedge is one geometric variable (d_1 in Fig. 1b) and the other variable is half of the thickness of the tooth (d_2 in Fig. 1b). The positive and negative allowable change from initial values of the variables is provided in Table 1.

Table 1. Allowable percent change from initial values for geometric variables

Variable	Allowable decrease	Allowable increase
d_1	−30%	+90%
d_2	−30%	+30%

In the initial design, a thermal analysis of the motor, including a water-cooled aluminum flange, showed that the current level of 12 A produced a steady-state temperature lower than the permanent magnets and winding insulation temperature limits. A size 18 AWG wire, which has a maximum current limit of

16 A, was used in the original design. Higher current will lead to higher force; however, higher current may also lead to temperatures that are too high for the selected cooling method. Therefore, the peak current is selected to vary between 12 A and 16 A.

The multi-objective optimization problem is represented by:

$$maximize \quad F \tag{1}$$

$$minimize \quad F_{pp} \tag{2}$$

Subject to

$$d_1 \geq d_2 \tag{3}$$

$$\frac{N\hat{I}}{A_{slot}} = J \leq 9.5 \tag{4}$$

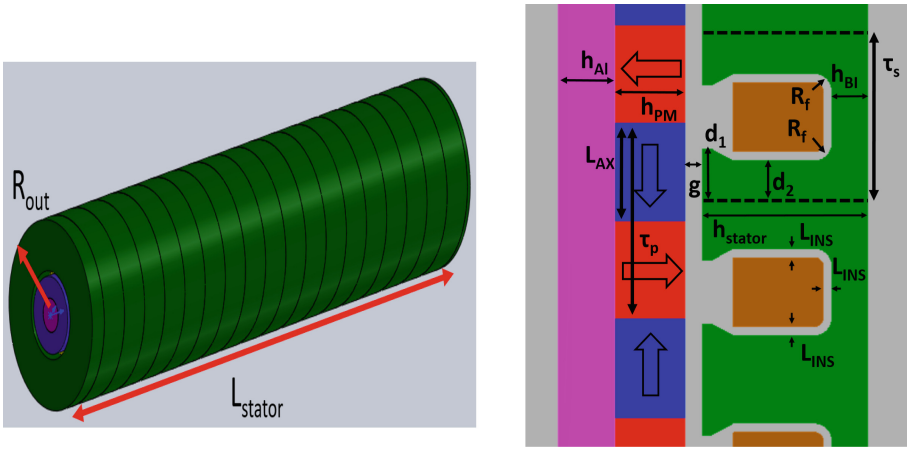
Here F is the average force, F_{pp} is the peak-to-peak force ripple, J is the current density, N is the number of turns, A_{slot} is the stator slot area and $\hat{I}$ is the peak current.

3 LPMSM Model

Evaluating the small geometric changes for optimization requires high-resolution FEA. Tubular LPMSM typically requires 3D FEA simulation; however, a 2D axisymmetric model of the LPMSM was shown to accurately calculate the average force and force ripple of interest for this optimization problem. Using a 2D model reduces the required computation time for the optimization process. A full and cross-sectional view of the LPMSM are shown in Fig. 2. Geometric parameters and performance metrics are provided in Tables 2 and 3 respectively.

Table 2. Geometric parameters of initial design

Parameter	Symbol	Value (mm)	Parameter	Symbol	Value (mm)
Stator length	L_{stator}	354	Stator height	h_{stator}	18.7
Stator outer diameter	R_{out}	35	Insulation length	L_{INS}	0.5
Aluminum height	h_{Al}	6.3	Slot pitch	τ_s	18.88
PM height	h_{PM}	8	Fillet radius	R_f	0
Length of axial PM	L_{AX}	10.625	Back iron height	h_{BI}	4
Pole pitch	τ_p	21.25	Half wedge height	d_1	4.72
Air gap length	g	2	Half tooth width	d_2	4.72



(a) Full view of LPMSM

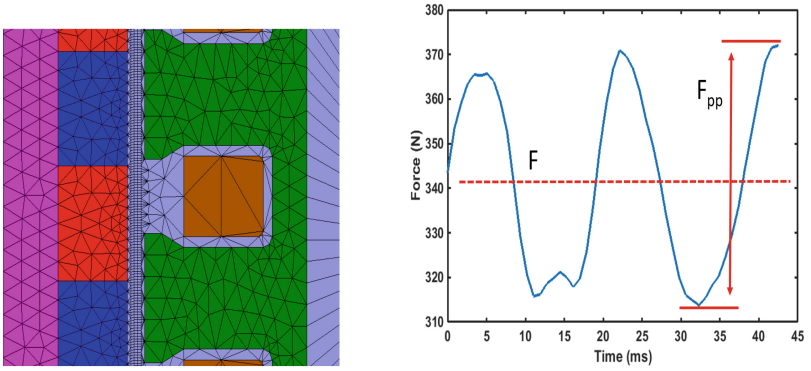
(b) 2D cross-sectional view of LPMSM

Fig. 1. LPMSM geometry with labeled geometric parameters

Table 3. Operating characteristics of initial machine design

Performance measure	Value	Performance measure	Value
Average force	266 N	Rated speed	26.2 m/s
Force ripple	47.88 N	Rated voltage	480 V_{ll}
Number of turns per coil	80	Rated power	5 kW

Figure 2a shows the mesh used in the axisymmetric FEA model of the LPMSM. The mesh in the air gap region is important because that is where the energy transfer takes place. Five layers of elements with a 0.4 mm length are used in the air gap mesh to obtain an accurate calculation of the force ripple. Further away from the air gap region, the mesh size becomes less critical for force and force ripple calculation. To avoid forces due to the end effects that occur in short secondary tubular LPMSM [14], the initial position is two pole-pitches (τ_p) away from the end of the stator and the total travel distance is limited to one electrical cycle. The force waveform from one design solution is shown in Fig. 2b. The maximum and minimum of the force waveform are used to calculate the force ripple F_{pp}.



(a) FEA mesh used in simulation

(b) Example force waveform with F and F_{pp} calculation shown

Fig. 2. Mesh in FEA setup and example force waveform used to evaluate objectives

4 Optimization Algorithm

A GA built into the FEA software and a user-defined GA are used to optimize the LPMSM. The parameters, constraints and objectives used for both algorithms are listed in Table 4.

Table 4. Variable limits and GA parameters used in optimization

Variable	Value
d_1 range	3.3 mm to 8.5 mm
d_2 range	3.3 mm to 6.1 mm
$\hat{I}$ range	12 to 16
Number of generations	10
Population size	200

One of the optimization algorithms used is built into the FEA software package. Details regarding the optimization processes used in the built-in GA are masked from the user. There are few parameters that the user can modify to improve the results of the built-in algorithm, such as population size, number of generations and the weights on each objective. Unfortunately, documentation for the built-in algorithms is limited and does not detail how the algorithm handles diversity, elite preservation, and constraints. Knowing such details is a primary advantage of a user-defined algorithm.

Most FEA software packages allows for the user to custom optimization algorithms. In this work, NSGA-II is selected as the user-defined algorithm. NSGA-II is a well-documented GA in literature [15] and has shown to perform well for two-objective optimization problems. It is well-known that with NSGA-II, the elite

solutions are kept and diversity in the objective space is maintained throughout the generations.

4.1 Procedure

Genetic algorithms are stochastic processes, so multiple executions of the same algorithm for the same problem produces different solutions. Although more runs improve the statistical significance of the results of each GA, due to the use of a high-fidelity model, both algorithms are executed four times. For each GA, the Pareto-optimal front is created from the non-dominated solutions of the last generation of all four runs. In three of the four optimization runs, the population of the initial generation is randomly selected. In the remaining run the population of the initial generation in each algorithm are identical.

The Pareto-optimal front is generated in the process shown in Fig. 3. For electric machine design, strict adherence to the constraints is required. Solutions that violate the constraints are not considered to be on the Pareto-optimal front and are removed.

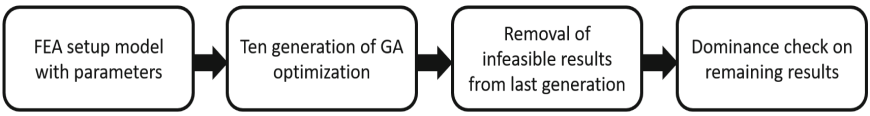


Fig. 3. Flow chart showing how Pareto front solutions were obtained

4.2 Evaluation Metrics

The true Pareto-optimal front is not known for this problem. To compare built-in GA to the user-defined GA, four quantifiable metrics are used; normalized hypervolume, Set Convergence Metric, Spacing, and total number of Pareto-optimal points. The normalized hypervolume and Set Convergence Metric quantify the convergence to the Pareto-optimal front. Spacing is used to quantify the diversity of solutions. The number of Pareto-optimal solutions quantifies the number of design options available for machine designers.

To calculate hypervolume, a reference point and ideal point are used to normalize the area of the Pareto front. The reference point is selected as a combination of the worst objective function values and the ideal point is selected as the combination of best objective function values. The objective value of each solution on the Pareto front is normalized and the hypervolume is calculated as the total area covered by the solutions [15]. A larger hypervolume is a good indicator of convergence to the true Pareto-optimal front. The Set Convergence Metric [15] compares the non-dominated solutions on two Pareto-fronts. A percentage of the non-dominated solutions that are dominated by at least one solution from

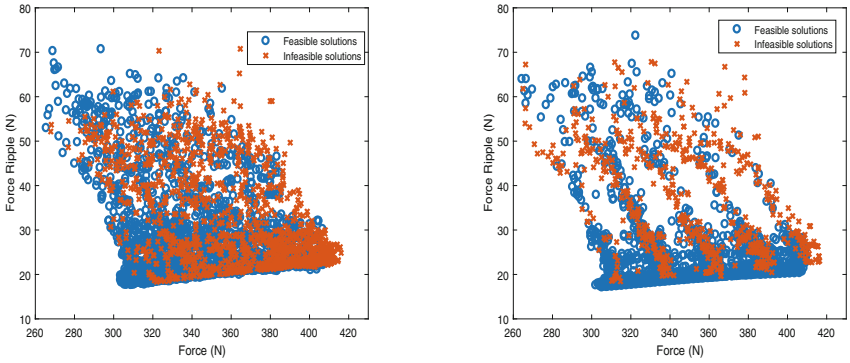
the other Pareto-front is calculated. A lower percentage indicates that fewer solutions found by that algorithm were dominated by a solution found from the other algorithm.

Diversity of solutions on the Pareto-optimal front is compared by calculating the Spacing [15], as given in Eq. (5). Q is the total number of non-dominated solutions, d_i represents the minimum distance between point "i" and all of the other points, and $\bar{d}$ represents the average distance between all points. A smaller number for S is desired, which means there is small variation in the space between the non-dominated solutions.

$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \tag{5}$$

5 Optimization Results

The complete solution space from the built-in GA is provided in Fig. 4a, and that of the user-defined GA is shown in Fig. 4b. As indicated by the different markers, more infeasible solutions were found from the built-in optimization algorithm.



(a) Entire solution space of every generation using the built-in GA

(b) Entire solution space of every generation using the user-defined GA

Fig. 4. All 200 members of 10 generations in the four combined runs of each GA where the infeasible points that violate constraints are indicated

The final Pareto-optimal front from each algorithm is created from the non-dominated, feasible solutions from the solution spaces and is provided in Fig. 5.

The tenth generation of each GA were used to create the Pareto-front. Some of the solutions in the tenth generation of the results from the built-in GA violated the constraint on current density, as shown in Eq. (4). These results

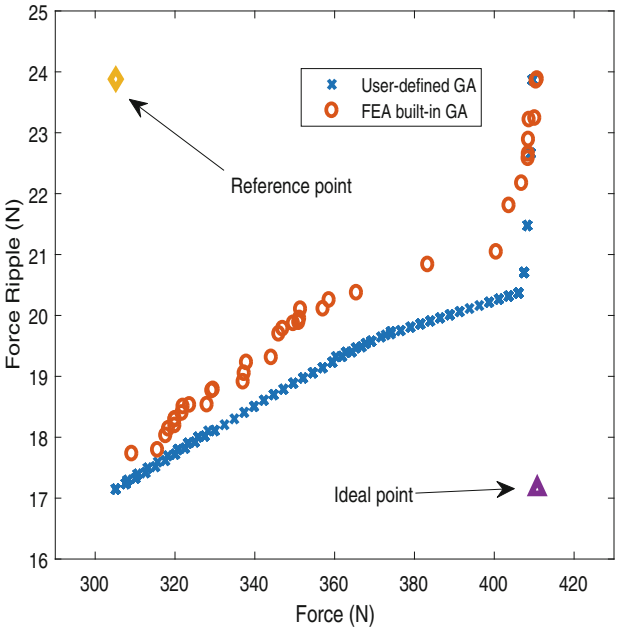


Fig. 5. Pareto-optimal front generated from four total runs of each optimization algorithm

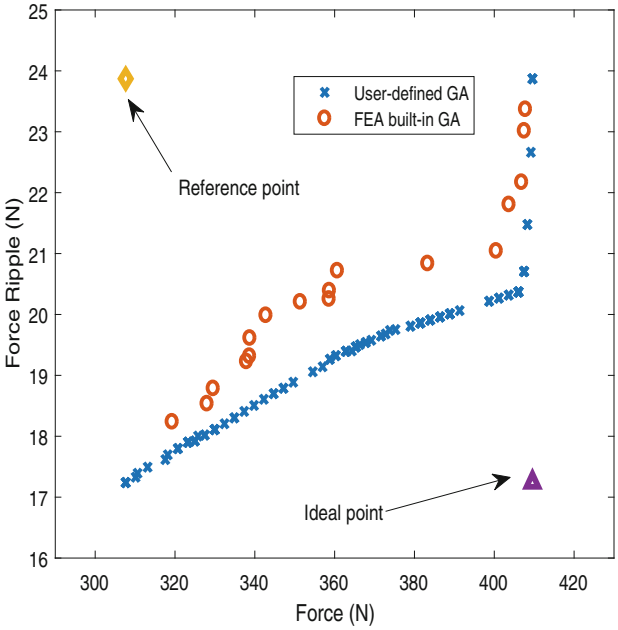


Fig. 6. Pareto-optimal front generated from each algorithm starting with same initial population

were excluded from the dominance check since they are infeasible. The user-defined algorithm did not produce any infeasible solutions from the second to the tenth generation.

The reference points are created from the solutions of the resulting Pareto fronts with the worst values of each objective and the ideal points are created from the solutions with the best values of each objective. The reference and ideal points used for hypervolume calculation of the four combined runs and the runs that began with the same initial population are provided in Table 5 and are displayed in Figs. 5 and 6.

Table 5. Hypervolume reference point and ideal point used to normalize the area covered by the two Pareto-fronts in Figs. 5 and 6

Pareto front	Reference point	Ideal point
Four combined runs	(305.1637 N, 23.88 N)	(410.7616 N, 17.1483 N)
Same initial population	(307.6479 N, 23.8705 N)	(409.5655 N, 17.2399 N)

Table 6 presents the convergence and diversity metrics. The Pareto-optimal front obtained from the user-defined algorithm resulted in 22.1% higher hypervolume and only 1.61% of its solutions are dominated by any solution from the built-in GA, indicating that the convergence is significantly better than that of the built-in GA.

When comparing the diversity of results on the Pareto-optimal front, the user-defined algorithm shows a significant advantage over the built-in GA. Spacing is 71.3% lower, indicating that the solutions are more evenly distributed across the Pareto-optimal front. The number of total points on the Pareto-optimal front is 1.68 times higher in the results of the user-defined algorithm. This means that there is a larger variety of optimal solutions available for decision making.

Table 6. Pareto front analysis from four runs combined for each GA

Result	Built-in FEA GA	User-defined GA linked to FEA
Normalized hypervolume	0.5751	0.7024
Spacing	3.0602	0.8774
Set Convergence Metric	86.49%	1.61%
Solutions on Pareto front	37	62
Infeasible solutions removed	78	0

The Pareto fronts from each algorithm with the same initial population is presented in Fig. 6 and the analysis of the Pareto front is provided in Table 7.

Convergence is significantly better in the Pareto front found with the user-defined GA where the hypervolume is 28.9% higher. Also, the Set Convergence Metric shows all of the Pareto-optimal points of the built-in GA are completely dominated by the Pareto front of the user-defined GA. Diversity is also significantly better where the spacing is 82% lower in the Pareto front from the user-defined GA and there are 2.88 times more optimal points for decision making. Using the same population members in the initial generation did not return a different result in the convergence or diversity of points on the Pareto-optimal front between the user-defined and built-in GAs.

The final row in Tables 6 and 7 show the number of infeasible results that were removed from the final generation from each GA. With the built-in GA the constraint handling method is not known, and as a result, there were a significant number of solutions in the final generation that violated constraints.

Table 7. Pareto front analysis where each GA is set with the same population members in the initial generation

Result	Built-in FEA GA	User-defined GA linked to FEA
Normalized hypervolume	0.5512	0.7105
Spacing	4.2935	0.7701
Set Convergence Metric	100%	0%
Solutions on Pareto front	17	49
Infeasible solutions removed	9	0

6 Conclusion

Results obtained from a built-in optimization tool in FEA software and a user-defined optimization algorithm from a electric machine design problem show several quantitative and qualitative advantages of a user-defined algorithm. It is shown that:

1. The user-defined algorithm results in more Pareto-optimal solutions
2. The user-defined algorithm has more solutions that dominate those resulting from the built-in GA optimization
3. Solutions from the user-defined algorithm are more diverse
4. Infeasible solutions were avoided using the user-defined algorithm

Considering the advantages evaluated in this work, it is important for electric machine designers to be knowledgeable of optimization algorithms rather than relying on undocumented, built-in algorithms. As different machine designs have different constraints and requirements to meet, more customized optimization tools are needed for efficient design of all possible machine topologies.

References

1. Waide, P., Brunner, C.U.: Energy-efficiency policy opportunities for electric motor-driven systems. OECD Publishing, IEA Energy Papers 2011/7 (2011). https://EconPapers.repec.org/RePEc:oec:ieaaaa:2011/7-en
2. Bramerdorfer, G., Zavoianu, A., Silber, S., Lughofer, E., Amrhein, W.: Possibilities for speeding up the FE-based optimization of electrical machines - a case study. IEEE Trans. Ind. Appl. **52**(6), 4668–4677 (2016)
3. Sizov, G.Y., Ionel, D.M., Demerdash, N.A.O.: A review of efficient FE modeling techniques with applications to PM AC machines. In: 2011 IEEE Power and Energy Society General Meeting, pp. 1–6, July 2011
4. Lim, D.K., Woo, D.K., Yeo, H.K., Jung, S.Y., Ro, J.S., Jung, H.K.: A novel surrogate-assisted multi-objective optimization algorithm for an electromagnetic machine design. IEEE Trans. Magn. **51**(3), 1–4 (2015)
5. Sindhya, K., Manninen, A., Miettinen, K., Pippuri, J.: Design of a permanent magnet synchronous generator using interactive multiobjective optimization. IEEE Trans. Ind. Electron. **64**(12), 9776–9783 (2017)
6. Akiki, P., et al.: Multi-physics modeling and optimization of a multi-V-shape IPM with concentrated winding. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), pp. 1–7, May 2017
7. Jing, L., Qu, R., Kong, W., Li, D., Huang, H.: Genetic-algorithm-based analytical method of SMPM motors. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), pp. 1–6, May 2017
8. Islam, M.Z., Choi, S.: Design optimization of rare-earth free PM-assisted synchronous reluctance motor to improve demagnetization prevention capability. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), pp. 1–6, May 2017
9. Takbash, A., Pillay, P.: Design optimization of a new spoke type variable-flux motor using AlNiCo permanent-magnet. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), pp. 1–6, May 2017
10. Baek, J., Rahimian, M.M., Toliyat, H.A.: Optimal design and comparison of stator winding configurations in permanent magnet assisted synchronous reluctance generator. In: 2009 IEEE International Electric Machines and Drives Conference, pp. 732–737, May 2009
11. Lopez, C.A., Strangas, E.G.: Optimization of PMSM performance with torque ripple reduction and loss considerations. In: 2018 XIII International Conference on Electrical Machines (ICEM), September 2018, pp. 899–905 (2018)
12. Jensen, W.R., Pham, T.Q., Foster, S.N.: Linear permanent magnet synchronous machine for high acceleration applications. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), May 2017, pp. 1–8 (2017)
13. Jahns, T.M., Soong, W.L.: Pulsating torque minimization techniques for permanent magnet AC motor drives-a review. IEEE Trans. Ind. Electron. **43**(2), 321–330 (1996)
14. Pham, T.Q., Jensen, W.R., Foster, S.N.: Stator incipient fault identification in short secondary linear permanent magnet synchronous machines. In: 2017 IEEE International Electric Machines and Drives Conference (IEMDC), May 2017, pp. 1–7 (2017)
15. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Hoboken (2003)