Design of synchronous reluctance machines with multi-objective optimization algorithms

Design of synchronous reluctance machines with multi-objective optimization algorithms

Francesco Cupertino
Politecnico di Bari - DEI
Bari, Italy
cupertino@poliba.it

Gianmario Pellegrino
Politecnico di Torino - DENERG
Torino, Italy
gianmario.pellegrino@polito.it

Chris Gerada
University of Nottingham
Nottingham, UK
chris.gerada@nottingham.ac.uk

***Abstract*—The design optimization of synchronous reluctance (SyR) machines is considered in this paper by means of a Finite Element Analysis-based multi-objective optimization algorithm (MOOA). The research focuses on the design of the rotor geometry which is the key aspect of SyR machines design. In particular, this digest analyzes the performance of several popular MOOAs and the impact of their settings on the quality of the final design. A procedure to minimize the computational burden of the optimization process is introduced and applied for the first time to a five layer rotor. A rotor prototype has been realized to demonstrate the feasibility of the design procedure.**

I. INTRODUCTION

Synchronous Reluctance (SyR) motors are a viable alternative to inverter driven induction motors because they allow a size reduction or an improvement in efficiency [1,2]. The rotor geometry is characterized by multiple flux barriers and many configurations are possible, in terms of number of the barriers and their shape and dimensions. The literature of SyR motor design is vast but the design guidelines vary with the authors [3-4]. Optimization algorithms have been applied for the last two decades to the design of such machines [5] but, again, a standard procedure has not emerged yet, especially for machines having three or more layers. This is not a coincidence and it is related to two main causes: the many degrees of freedom involved in the rotor design and the need for a comprehensive evaluation of steel saturation.

Linear magnetic models are too simplistic for SyR machines, because the rotor iron saturates locally even at very little load, e.g. in the structural bridges, and then saturates as a whole progressively with the load. For this reason, all design approaches make use of Finite Element Analysis (FEA), at least for the final verification of the machine characteristics [6-7].

Apart from [5], FEA-based optimization has not been utilized as often in the literature of SyR machines, and it is generally dedicated to simpler rotor geometries with permanent magnets in, such as the single layer rotor motor in [8]. In [9-10] we have presented a multi-objective design optimization of multi-barrier rotors based on FEA, aiming to provide a comprehensive approach to SyR rotors design within a reasonable computation time. The fast FEA evaluation of candidate machines [9] and the two-step application of the multi-objective optimization algorithm (MOOA) [10] where the focuses of past works.

The goals of this paper are to investigate the impact of the MOOA settings on the efficiency of the optimization process and to extend the design procedure of [10] to a five layer rotor. Different optimization algorithms are applied comparatively and general conclusions are drawn. A procedure for obtaining the robust convergence of the optimization within a given number of iterations is formalized. The comparison between the different MOOAs refers to the case of a three-layer SyR rotor. The five layers rotor is finally used as an example of how the MOOA procedure is robust towards the increase of the variables to be optimized.

Two machine prototypes, one with three layers and one with five, are designed for optimal torque and torque ripple and experimentally evaluated. Iron losses are also compared and commented, FEA calculated.

The paper describes how to set-up the optimization procedure so to minimize convergence time and obtain statistically repeatable results. The results are confirmed on two problems of different size, namely the optimization of three and five layer rotors that converge in very similar times. Last, the five layer design found by MOOA is unconventional with respect to the related literature and allows a little improvement of motor efficiency at the expenses of rotor mechanical strength.

II. PROBLEM STATEMENT

The two objectives to be optimized are the average torque and the torque ripple. The stator geometry is defined, and it is the one of a PM-assisted machine for compressor application. The original ratings of the PM-assisted machine are 2.5 Nm at 5400 rpm, liquid cooled at 90°C. The ratings of the SyR motor designed here are 4 Nm at 5000 rpm, with forced ventilation. The continuous peak current is 16.8 A and the rated voltage, at the dc-link level, is 270 V.

The windings are distributed, not chorded, with two slots per pole per phase. The airgap thickness is 0.5 mm, and the rotor design is considered here. The peak current density is set during the optimization, to a value that can correspond to

the continuous operation current i_0, or to a multiple, corresponding to transient overload. The optimization at overload current eases the optimization, at will be shown.

Apart from the selected two objectives, other objectives to be optimized could have been the power factor and the core losses. However, once the airgap and the current level are fixed, a nearly optimized power factor will follow from torque optimization, that is also machine saliency optimization, in a way [11]. Core loss are important and sometimes in competition with torque ripple minimization. They have not been included in the optimization not to weight the computation burden excessively. The conclusion of the paper is that, as for all machines in the literature, the geometries found by the MOOAs have a grade of core loss that depends on the number of layers in the rotor and the number of stator slots [12-14].

A. Parametrization of Rotor Geometry

The rotor geometry is defined in Fig. 1 for an example motor with two pole-pairs and 5 layers. The shapes of the barriers in the literature are so varied and so full of parameters that a strong simplification is necessary, to reduce the number of iterations needed to complete the optimization process. Each variable should have a reasonable impact on at least one of the performance indexes of the optimization. It was demonstrated in [10] that rotor layers with circular shape, described with only two variables per layer, can reach performances similar to those of more complex geometries despite the very reduced set of parameters. For circular layers such parameters are:

- the layer angular positions at the airgap $\Delta\alpha j$;
- the layer heights hc_j;

Last, the current phase angle γ is also included in the optimization variables, for it is unknown a priori.

If n_{lay} is the number of rotor layers, the number of variables to be optimized is then 2 n_{lay} +1. This makes a 7-dimensional space of the inputs, for the three layers machine and 11 dimensions of the five layers one.

The set of parameters and the search bounds are reported in Table I. Where p.u. is indicated, it means that the value is in per-unit of the total space or angle available for the layers. The first angular input $\Delta\alpha_1$ is the most critical one, and it decides how much angular space is left for the other layers tips. The p.u. angles from 2 to n_{lay} then distribute the layers tips over this space. After the barriers tips are placed, their thickness hc_i are evaluated from the p.u. MOOA inputs such that if they are all 1 p.u. the layers are all at maximum thickness and all thick the same. For maximum width it is intended that a minimum thickness of 1.0 mm is guaranteed to all the flux guides between barrier and barrier. This way, incompatible combinations of the inputs, leading to layers overlap are avoided.

B. Fast FEA evaluation of the candidate machines

The performance indexes to be optimized, torque and torque ripple, are evaluated in a single current condition (amplitude and phase angle γ in synchronous coordinates), so to reduce computation.

TABLE I
LIMITS OF THE SEARCH SPACE FOR THE GLOBAL SEARCH (GS)

Parameter	Min value	Max value	Units
hc_i (i=1,..,n_{lay})	0.2	1	p.u.
$\Delta\alpha_1$	15	27	degrees
$\Delta\alpha_j$ (j=2,..,n_{lay})	0.33	0.67	p.u.
γ	20	80	degrees

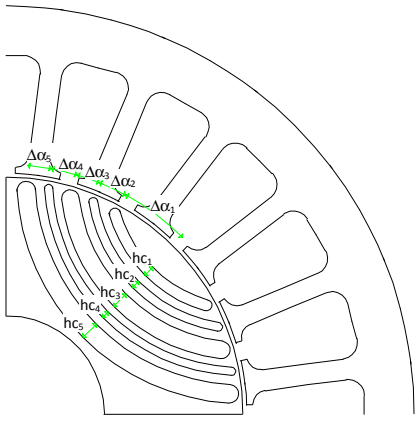


Figure 1. Rotor geometry with 5 layers: the $\Delta\alpha j$ angles define the layer angular positions, hci are the layer heights.

The current amplitude selected in the examples is twice the machine rated current (2 i_0), as a trade-off between continuous torque and maximum overload conditions (3 i_0). This because preliminary investigations revealed that machines with a good torque/torque ripple compromise in overload conditions perform well also at lower current levels, but not vice-versa, because the machines optimized at lower currents have a tendency to perform badly at overload.

Dealing with the current phase angle, the correct choice is to use the maximum torque per ampere (MTPA) angle condition γ_{MTPA}. However, this angle is not known a priori when a new machine is evaluated and its identification normally requires that different trial values of γ are tested, with time consumption. To reduce computational time, the current phase angle is included into the MOOA inputs, and each motor is evaluated for a single current angle that is selected by the MOOA. All the tested MOOAs showed to be capable of finding to the correct current phase condition for all optimized machines.

The torque of each candidate machine is FEA calculated in n equally spaced rotor positions, covering one stator slot pitch (τ_{st}). The average value and the standard deviation of the torque waveform are the two cost functions of the optimization. The stator slot pitch has been chosen for it is representative of the most significant torque ripple harmonic component. The number of simulations required to avoid the aliasing of significant torque harmonics is discussed in [10], where a random offset has been introduced. This permits to minimize the aliasing of the main ripple harmonics when

very few positions are simulated. In this case, five rotor positions are simulated per each candidate evaluation.

In turn, each machine is evaluated by the MOOAs via the FEA calculation of its torque in a single current amplitude and phase condition, and for five rotor positions with the first position decided randomly. All the final Pareto fronts evaluated so far showed that:

1. the γ_{MTPA} angle is correctly estimated by the MOOA.
2. The torque ripple of all output machines is minimized.
3. Specifically, torque ripple is minimum along the MTPA trajectory, also at current amplitudes that are different from the simulated one.

C. Optimization procedure and MOOA settings

The proposed MOOA-based design procedure consists of a first stage called global search (GS) and a successive local search refinement stage (LS). During the GS, the bounds of the search space are kept quite large, meaning that all the feasible rotors are considered as potential solutions. Table I reports the bounds values used for the GS optimization. At the end of the GS stage, one of the optimized machines is selected for further refinement. The LS stage is a successive MOOA run, with the bounds of the inputs restricted around the set of inputs that define the GS-selected machine.

III. Meta-Heuristic Algorithms

There are many meta-heuristic algorithms for single- and multi-objective optimization problems. They have in common the use of a set of candidate solutions (population) that are iteratively modified according to some probabilistic rules and that converge to the global minimum of the objective function with a certain degree of probability. It is well known that no best algorithm exists, being valid for any class of problems (no free-lunch theorem [15]) and that no algorithm can avoid the risk of premature convergence to sub-optimal solutions.

A. Genetic Algorithms

Genetic algorithms (GAs) are a particular class of meta-heuristic algorithms that use techniques inspired by evolutionary biology such as mutation, selection, and crossover. Genetic algorithms are implemented as a computer program in which a population of candidate solutions (called individuals) to an optimization problem evolves toward better solutions. Usually individuals are represented by a vector of N_g real or binary numbers (called genes). An initial sampling of N_p individuals is performed randomly with a uniform distribution function within the decision space D. All the individuals $\boldsymbol{x}_k$, $k=1,\ldots, N_p$ are evaluated and a fitness value (performance index) $f(\boldsymbol{x}_k)$ is associated to each of them. A couple of individuals (called parents) are selected from the population. Individuals with better fitness values are more likely to be selected according to well-known criteria (roulette wheel, rank selection). Two provisional offspring are generated applying a crossover operator to the parents. According to the uniform crossover, each gene of the first parent is randomly assigned to the first or second child. The empty genes are filled with those of the second parent. Finally, a mutation operator is applied to the provisional offspring. A few randomly selected genes of the offspring are replaced with random values.

After several iterations (parent selection, crossover and mutation) N_p offspring are generated and their fitness values calculated. The algorithm is referred to as elitist when the worst part of the offspring is replaced by the best parents. The grade of elitism can be strict (only offspring better than the parents survive) or mild (parents are excluded after being in the population for a maximum number of generations, independently of their performance). Strong elitism can accelerate convergence in the initial steps but it could also lead to premature convergence (the explorative effect of crossover vanishes).

B. Differential Evolution

According to its original definition (see [16]), the differential evolution (DE) algorithm consists of the following steps. An initial population is generated as already described for genetic algorithms. At each generation, for each individual $\boldsymbol{x}_k$ of the N_p, other three individuals ($\boldsymbol{x}_r$, $\boldsymbol{x}_s$, and $\boldsymbol{x}_t$) are randomly extracted from the population. According to the differential evolution logic, a provisional offspring $\boldsymbol{x}'_{off}$ is generated by mutation as

$$\boldsymbol{x}'_{off} = \boldsymbol{x}_t + F(\boldsymbol{x}_r - \boldsymbol{x}_s) \tag{1}$$

where $F \in [0, 2]$ is a scale factor which controls the length of the exploration vector $(\boldsymbol{x}_r - \boldsymbol{x}_s)$ and thus determines how far from point $\boldsymbol{x}_t$ the offspring should be generated. Other variants of the mutation rule have been subsequently proposed in literature. When the provisional offspring $\boldsymbol{x}'_{off}$ has been generated, each of his genes is exchanged or not with the corresponding gene of $\boldsymbol{x}_k$ with a random decision ruled by the crossover rate parameter C_r. The final offspring $\boldsymbol{x}_{off}$ is generated according to the crossover procedure (2):

$$\boldsymbol{x}_{off}[i] = \begin{cases} \boldsymbol{x}'_{off}[i] \; if \; rand \leq C_r \\ \boldsymbol{x}_k[i] \; otherwise \end{cases} \tag{2}$$

where *rand* is a random number between 0 and 1; i is the index of the gene under examination; Cr is a the crossover rate. This crossover strategy is known as binomial crossover. The resulting offspring $\boldsymbol{x}_{off}$ is evaluated and, according to a one-to-one spawning strategy, it replaces $\boldsymbol{x}_k$ if and only if the fitness values has been improved, otherwise no replacement occurs.

C. Simulated annealing

Also the simulated annealing (SA) algorithm shares the same initialization procedure with the aforementioned algorithms. SA is inspired to annealing in metallurgy and is based on a random perturbation of each individual in the population according to the following rule

$$\boldsymbol{x}_{off} = \boldsymbol{x}_k + \sqrt{T}\, rand(N_g) \tag{3}$$

where $rand(N_g)$ is a vector of N_g random numbers, and T is the current "equivalent temperature". The resulting offspring is evaluated and it replaces $\boldsymbol{x}_k$ if the fitness value has been improved. The offspring also replaces $\boldsymbol{x}_k$ when the fitness is not improved but

$$rand < e^{\frac{|f(x_k)-f(x_{off})|}{T}} \quad (4)$$

where $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{off})$ is the difference in fitness value before and after the perturbation. At the end of each iteration the temperature is decreased with a linear or logarithmic rule. At the beginning of the search perturbations are large and the algorithm tends to explore the search domain, at the end it exploits the information with small changes of the potential solutions.

D. Multi-objective Algotithms

All the mentioned algorithms have been firstly introduced as single-objective but can adapt to multi-objective problems by introduction of the concept of dominance. In multi-objective problems a solution is non-dominated when there is no other solution that has better fitness values of this one for all the objective functions. All non-dominated solutions form the Pareto front of the optimization and are all ranked as current best solutions. Among the remaining solutions it is possible to define a second Pareto front. All the solutions that belong to this front are ranked "two" and so on. The initial population is randomly selected and after fitness evaluation all the solutions are ranked according to the front to which they belong. Solutions are modified using the specific operators of the selected algorithm that have been previously introduced. Generally speaking, offspring substitute the parents when they belong to a better ranked Pareto front. To avoid premature convergence, the number of solutions belonging to the first Pareto front is usually limited by discarding solutions too close each other. When the stopping criterion is reached (usually a maximum number of fitness evaluation is fixed) the solutions of the first Pareto front represent the possible compromise among the chosen objectives.

IV. Simulation Results

In this section the just described three MOOAs are applied to the optimization of a five layer SyR rotors, with the machines specifications reported in Table II. The MOOAs acronyms are:

- MOGA: multi-objective genetic algorithm;
- MODE: multi-objective differential evolution;
- MOSA: multi-objective simulated annealing.

Other algorithms could have been considered but, to the authors' knowledge, the ones used here represent a good compromise between effectiveness and simplicity. They are the state-of-the-art of multi-objective optimization algorithms and can be easily implemented thanks to the wide references available in the literature and they are available open source [17]. A special mention is deserved to GODLIKE [19] that is a multi-algorithm approach. It consists of running multiple algorithms in parallel and randomly exchange solutions among them so to counteract the premature convergence. In such a way it is possible to take the most of different algorithms but the advantages become evident only with very long optimization runs, that is not the case here.

TABLE II – SyR machines prototypes ratings

Continuous torque	4.5	Nm
Rated speed	5000	rpm
Rated voltage	270	V (dc-link)
Continuous current	16.8	A (pk)
Stack outer diameter	101	mm
Rotor diameter	58.6	mm
Airgap	0.5	mm
Stack length	65	mm
Steel grade	M470-50 M250-35	(Stator) (rotor)

A. Description of the simulation set-up

As said, each function evaluation (i.e. the evaluation of one candidate) consists of five FEA simulations in five rotor positions. Static-magnetic simulations have been used [20]. The evaluation of one candidate requires 2.6 seconds on a Intel Xeon E5-1620 workstation (4 cores, 3.60 GHz, 16 GB ram), thanks to the 5-core parallel calculation (note that the fifth core is emulated by the Xeon processor).

It is worth to underline that, beside the stochastic nature of the candidates' choice by the algorithm, also the evaluation of the fitness function is stochastic due to the random selection of the simulated rotor positions offset. On the one hand the random offset approach speeds the evaluation time. On the other hand, however, it implies that the machine performances can be under- or over-estimated. If a strong over-estimate occurs, a non-optimal machine can remain in the Pareto front till the end of the optimization due to the algorithms' elitism.

The defect of being too elitist is noticed here. All the machines of the final population of each run are re-evaluated more accurately using 15 time-stepped simulations instead of 5. The re-evaluated front can differ from the one given by the MOOA, as it happens that the solutions that were over-estimated during the optimization are not non-dominated in the re-estimated front. Such solutions can be disregarded after the post-processing. Although the approach proposed here gives satisfactory results, some modification to algorithms are currently under investigation so to avoid the final re-evaluation of the Pareto front.

The two main parameters to be selected for all the algorithms are the ***size of the population*** and the ***total number of function calls*** (or number of evaluations). In theory, larger numbers lead to more accurate solutions. In practice, due to the stochastic nature of the algorithms, a very large single run does not guarantee to find the actual Pareto front and repeated runs are anyway necessary to tackle the premature convergence of single runs.

Having in mind that the aim of this work is to formulate a software tool capable of supporting industrial motor designers, the quicker is the response of the optimization

procedure, the most helpful will be the automatic design proposed here. In this paper we investigate different values of population size and number of evaluations so to find the best compromise between computational burden and quality of the final design.

All the algorithms are tested in different conditions and, for sake of brevity, some of the most representative results are showed here, the ones using a population of solutions equal to 60, a number of function calls limited to 1200 (see fig. 2) and to 3000 (see fig. 3). The search bounds are the ones referred to as global search (GS), reported in Table I. The shorter runs (1200 calls) require about one hour while the longer ones (3000 runs) less than 2.5 hours on the selected processor and including the re-evaluation of the final population. To keep low the population size with no loss of performance, a penalty function has been applied: all the solutions with a torque ripple greater than 6% or an average torque lower than 4 Nm are penalized. In this way the final solutions are concentrated in the portion of interest of the whole Pareto front, and the small population size is not a problem.

B. Analysis of the simulation results

The performance of the algorithms has been investigated through ***ten GS runs for each MOOA***, a first time with 1200 evaluations (Fig. 2) and then again with 3000 evaluations (Fig. 3). The figures report all the obtained Pareto fronts in the torque vs. the torque ripple domain. Negative torque values are used so to obtain a more conventional shape of the Pareto estimates. Figures 2 and 3 give a quick comparative overview of the performances of the algorithms. The best algorithm is the one giving machines with high torque and low torque ripple but, even more important, it is the one giving repeatable Pareto fronts from one run to another. This latter quality is needed here, to allow for the reduction of the number of GS runs needed to complete the GS stage of the optimization. The Pareto front estimates are more concentrated when the number of function calls increases, for all the algorithms. MODE gives the best results in terms of torque/torque ripple values and repeatability of the estimates, and its Pareto fronts are already stable with just 1200 evaluations. MOSA has a great improvement from Fig. 2 to Fig. 3. MOGA has more sparse fronts and does not improve with 3000 evaluations.

The MODE gives the best results, and its Pareto fronts are stabilized already after 1200 evaluations. One of the GS-1200-MODE fronts (not the best one, purposely) is chosen, and one machine is selected out of the front (hereinafter GS-selected machine), to be the benchmark for the comparison with the other MOOAs. The GS-selected machine is worth an average 7.93 Nm, with 3.97% ripple. Table III reports the rate of success of each algorithm calculated as the percentage of GS runs that give at least a machine on the estimated Pareto front with a minimum torque of 7.8 Nm and 4% as maximum ripple. Again, 9 out of 10 MODE runs are over the benchmark already at 1200 function calls, while the other MOOAs are less performing.

TABLE III - PERCENTAGE OF SATISFACTORY GS RUNS (TORQUE>7.8, TORQUE RIPPLE<4%)

Algorithm	1200 function calls	3000 function calls
MODE	90%	100%
MOGA	40%	80%
MOSA	60%	100%

The single LS run is performed starting from the GS-selected solution, using the MODE algorithm stopped at 1200 function calls and with the search bounds restricted to ±15% with respect to GS-selected machine inputs. Figure 4 shows the Pareto front of the LS run (re-evaluated over 15 rotor positions) compared with the MODE-GS one the GS-selected machine belongs to. On this final Pareto front a machine was selected (hereinafter LS-machine) that improves by 1% the average torque and by 35% the torque ripple with respect to GS-machine.

It is important to put in evidence that none of the GS runs had any solution better than the final LS-machine. The whole procedure would require about 5 hours to be completed. When a very low ripple is not specifically required by the application, the LS-stage can be avoided with no practical loss of average torque.

V. EXPERIMENTAL RESULTS

Two SyR rotor prototypes have been built for validating the proposed design procedure. The one with 5 layers is the LS-machine of Fig.4. A three-layer rotor is also tested, designed with the same 4GS+LS procedure [10]. The two machines are indicated with 5C and 3C, respectively.

A. Discussion of the layers geometries

Fig. 5 reports a comparative section of the laminations. The 3C rotor has a regular pitch, consistently with the literature of minimum ripple machines [2,4]. The rotor pitch in the area where this is regular would correspond to rotor with 32 equivalent slots, that is one of the good combinations suggested in [4]. Dealing with the 5C rotor, the five layers thicknesses are neither progressive [4] or all equal [3]. The MOOA has designed a machine with three main layers (1, 3, 5), plus two very thin layers in between. This layer distribution has no equal in the literature, and it seems that the MOOA tried to group the layers together to form a three-layer like distribution. The positions at airgap of the main layers (1, 3, 5) are almost coincident with those expected from a rotor with 32 regular slots. The thin layers in between are positioned in the middle of the two contiguous larger layers. Such two extra layer, however, mitigate the harmonic content produced by the rotor reaction to the q-axis current, as demonstrated by the fact that the MOOAs converges very easily to low ripple solutions, despite of the many degrees of freedom. The more complicate problem (5C) with 10 geometric inputs has a convergence time that is very similar to the one of the simpler problem (3C – 6 inputs). The total computation time needed for running the 4GS+LS procedure is practically the same for both the 3C and the 5C rotors.

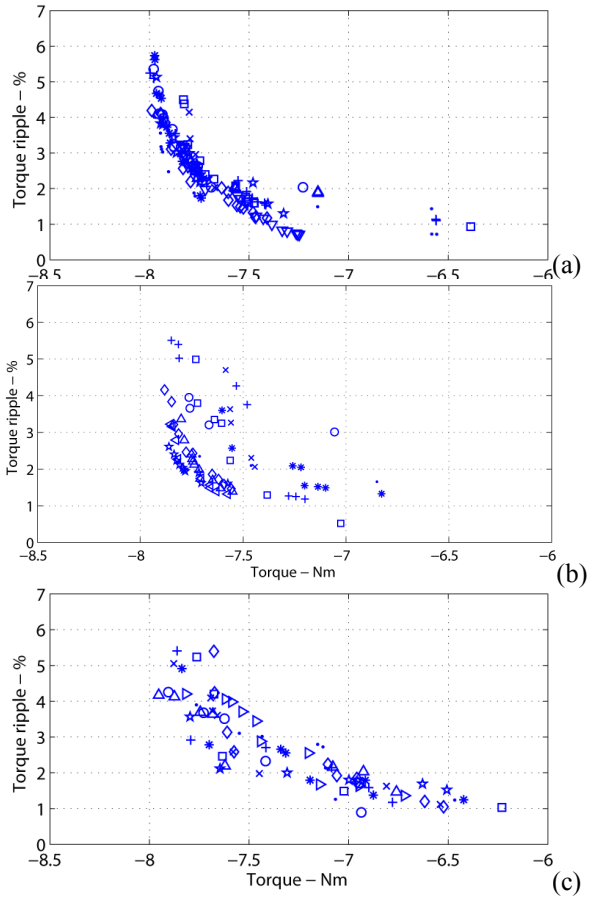


Figure 2. Pareto fronts obtained using (a) MODE, (b) MOGA, (c) MOSA: results obtained with ten runs, each one stopped at 1200 function calls.

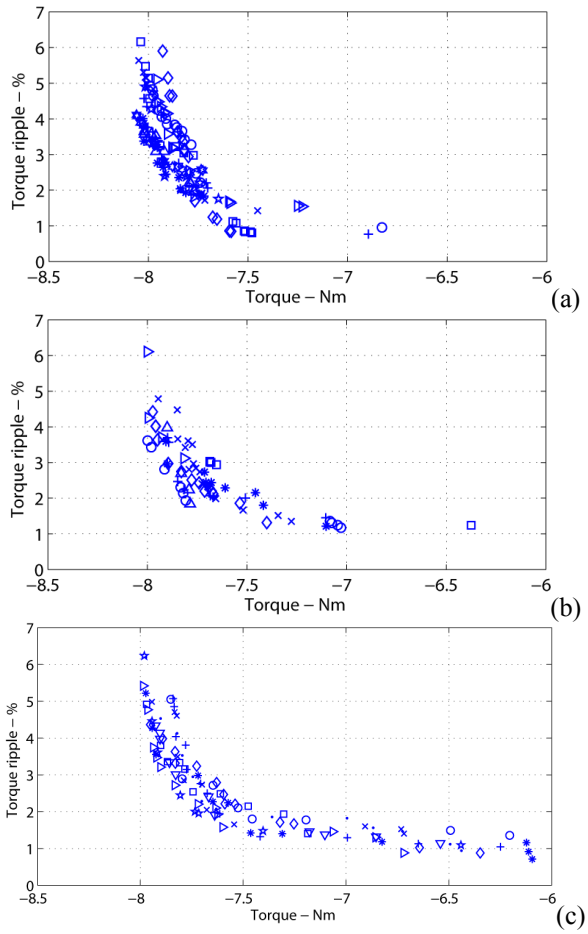


Figure 3. Pareto fronts obtained using (a) MODE, (b) MOGA, (c) MOSA: results obtained with ten runs, each one stopped at 3000 function calls

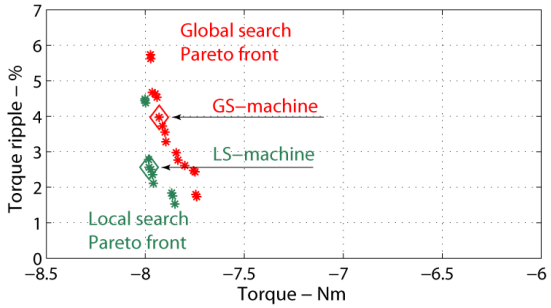
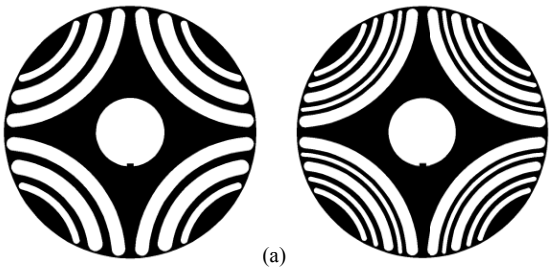


Figure 4. Selected GS run and LS run Pareto fronts obtained using MODE and 1200 function calls. GS- and LS-machines are evidenced with red and green diamonds.

B. Measured and simulated torque

The torque ripple maps versus the i_d, i_q current components have been measured using a dedicated test bench (see Fig. 6). The rotor speed is imposed to 10 rpm by a geared DC machine during the test. The motor under test is vector current controlled, using a dSPACE 1104 board. The current set-points and the torque log along one motor revolution are managed automatically by means of a Matlab script. The rating of the torque-meter has imposed not to exceed the 20 A per 30 A current area. The measured torque and torque ripple values are reported in Figs. 7 to 10 and compared with the magneto-static FE results. The average torque matches very well the simulation results. The torque ripple evaluated experimentally is higher than the simulated one, but both in FEA and experiments present a similar minimum-ripple trajectory in the i_d, i_q domain (Figs. 8 and 9). Even if the laminations have been realized using a high precision wire cut Electric Discharge Machining (EDM), it is reasonable that the increased torque ripple is justified by the prototypes' non-idealities due to the not exact knowledge of the materials characteristic in saturation, the non-ideal steel behavior in the areas close to laminations cuts, tolerances of manufacturing and assembling. The actual thickness of the inter layer bridges, for example, is very critical for the ripple waveform. Finally, the 5-layer rotor is fragile mechanical-wise, and withstood some partial damages in the steel area on top of the layers, where some of the laminations have disconnected bridges. All these considerations justify the differences between the ripple waveforms, when FEA and experimentally evaluated, in particular for prototype 5C (Fig. 10). However, one important results of this analysis is that ***the minimum ripple trajectory*** in the i_d, i_q plane is very ***close to the MTPA***, and this also in experiments (Figs. 8 and 9). The efficiency maps of the two prototypes are reported in

Fig. 11, in the torque versus speed plane. Core losses have been evaluated via FEA with Magnet, by Infolytica [21], in post processing. The detail of stator and rotor core loss is reported in Fig. 12 at the rated speed of 5000 rpm, for rated and partial load. The total core loss is under control, being lower than the Joule loss, for both machines. The most of core losses is on the stator, and they are lower in the 5C machine. This stands for a lower content of harmonics loss. The rotor loss is quite negligible, but they are lower in the 3C machine. This is consistent with the literature: a rotor with a higher number of "rotor slots" has higher rotor harmonic losses [14]. All considered, there is no valid reason for using a 5-layer rotor with this stator configuration (2 slots per pole per phase), as also confirmed in the literature [4]. The 5-layer rotor is more complicate to be cut and fragile mechanical wise, and it gives the same performance of the 3C rotor.



(a) (b)

Figure 5. a) cross section of the 3C rotor; b) cross section of the 5C rotor.

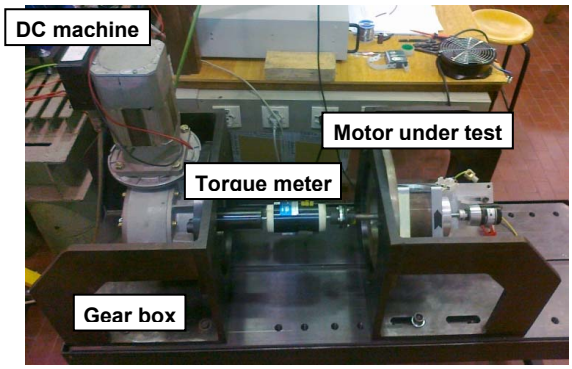
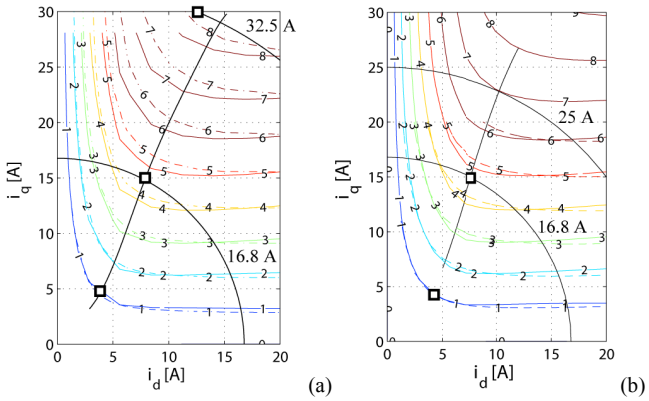
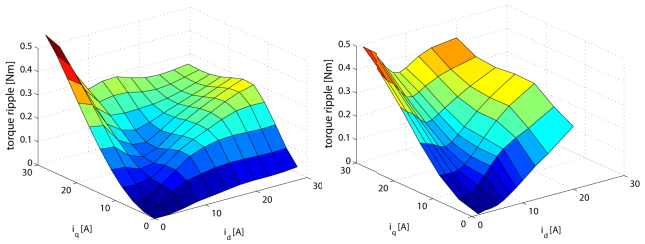


Figure 6. Test bench used to measure the torque ripple maps



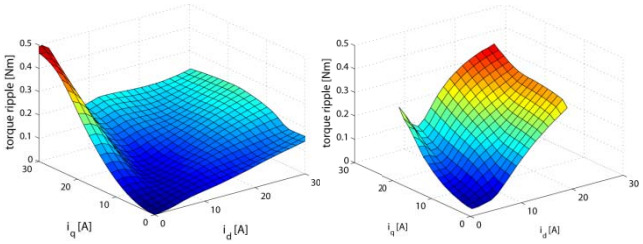
(a) (b)

Figure 7. Torque contours evaluated with FEA (continuous line) and with measures (dashed lines). The MTPA line is in evidence. a) Prototype 3C; b) Prototype 5C.



(a) (b)

Figure 8. Prototype 3C: torque ripple surface over the id, iq plane, according to FEA (a) and measurements (b).



(a) (b)

Figure 9. Prototype 5C: torque ripple surface over the id, iq plane, according to FEA (a) and measurements (b).

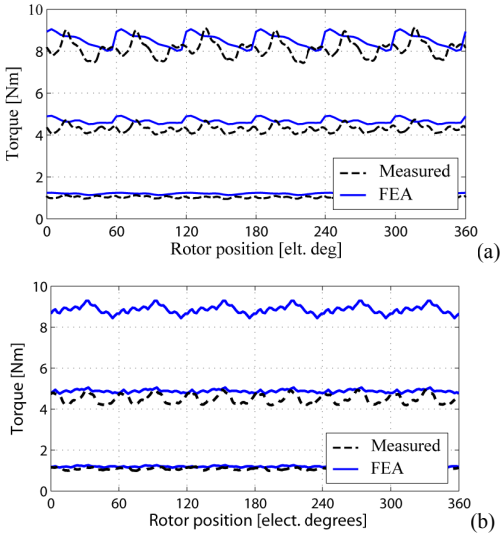


Figure 10. Torque waveforms in the i_d, i_q combinations indicated with black squares in Fig. 7a. a) Prototype 3C; b) prototype 5C.

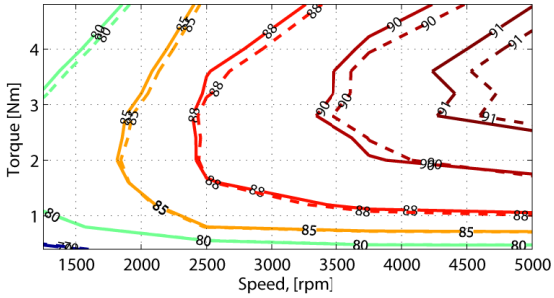


Figure 11. Efficiency map in the torque-speed plane for the prototyped (dotted lines) 3C and (solid lines) 5C machines

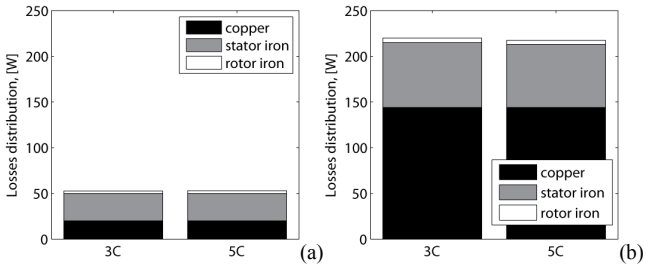


Figure 12. Losses distribution at 5000 rpm: a) 1 Nm; b) 4.5 Nm

VI. CONCLUSION

The paper presented a procedure for the automatic design of multi-layer synchronous reluctance rotors based on FEA and MOOAs of different nature. Three popular MOOAs selected from the literature have been compared, and Differential Evolution gave the best results in terms of time to converge and repeatability of the results. A comprehensive rotor design procedure has been formulated, obtaining a robust optimization of the performance within short time. This procedure is based on the fast FEA evaluation of candidate machines and on the unconventional use of the MOOA, with repeated Global Search runs and the refinement of the best solution via a further Local Search MOOA run. The two-stage 4GS+LS procedure has been validated on two example rotors, producing machines with comparable performance within similar computation times. The experimental results show that the optimized machines have a minimum ripple area along the MTPA control trajectory, and this is a consequence of the proposed MOOA+FEA approach. The conclusions on the 3- and 5-layer machines are consistent with the literature, showing that when the stator has two slots per pole per phase, there is no particular performance improvement when passing to the more complicated five layers geometry. However, the MOOA-FEA approach has produced a competitive five layer machine, using a non-standard distribution of the five layers widths and positions. As a last confirmation of the literature: the three layer rotor is the best match for this number of stator slots, and in fact the geometry found by the MOOA is similar to the ones in the literature. Five layers are non-optimal, according to the literature, but yet the MOOA has produced a competitive machine using a non-standard distribution of the layers.

ACKNOWLEDGMENT

This work was supported in part by project PON MALET – code PON01_01693. The authors would like to thank Maurizio Perta and Marco Palmieri for their support in the realization of MAGNET simulations.

REFERENCES

[1] Vagati, A.; Fratta, A.; Franceschini, G.; Rosso, P., "AC motors for high-performance drives: a design-based comparison," Industry Applications, IEEE Transactions on , vol.32, no.5, pp.1211,1219, Sep/Oct 1996

[2] Moghaddam, R.R.; Magnussen, F.; Sadarangani, C.; , "Theoretical and Experimental Reevaluation of Synchronous Reluctance Machine," Industrial Electronics, IEEE Transactions on , vol.57, no.1, pp.6-13, Jan. 2010

[3] T. A. Lipo, T. J. E. Miller, A. Vagati, I. Boldea, L. Malesani, and T. Fukao, "Synchronous reluctance drives" tutorial presented at IEEE IAS Annual Meeting, Denver, CO, Oct. 1994.

[4] Vagati, A.; Pastorelli, M.; Francheschini, G.; Petrache, S.C., "Design of low-torque-ripple synchronous reluctance motors," Industry Applications, IEEE Transactions on , vol.34, no.4, pp.758-765, Jul/Aug 1998.

[5] Kamper, M.J.; van der Merwe, F.S.; Williamson, S.,"Direct finite element design optimisation of the cageless reluctance synchronous machine", IEEE Transactions on Energy Conversion, Vol. 11, n. 3, September 1996.

[6] Vagati, A.; Canova, A.; Chiampi, M.; Pastorelli, M.; Repetto, M., "Design refinement of synchronous reluctance motors through finite-element analysis," Industry Applications, IEEE Transactions on , vol.36, no.4, pp.1094,1102, Jul/Aug 2000

[7] Lovelace, E.C.; Jahns, T.M.; Lang, J.H.; , "A saturating lumped-parameter model for an interior PM synchronous machine," Industry Applications, IEEE Transactions on , vol.38, no.3, pp.645-650, May/Jun 2002

[8] Sizov, G.Y.; Ionel, D.M.; Demerdash, N.A.O.; , "Multi-objective optimization of PM AC machines using computationally efficient - FEA and differential evolution," Electric Machines & Drives Conference (IEMDC), 2011 IEEE International , vol., no., pp.1528-1533, 15-18 May 2011

[9] Pellegrino, G.; Cupertino, F., "FEA-based multi-objective optimization of IPM motor design including rotor losses," IEEE Energy Conversion Congress and Exposition (ECCE), 2010, vol., no., pp.3659-3666, 12-16 Sept. 2010

[10] F. Cupertino, G.M. Pellegrino, E. Armando, C. Gerada, "A SyR and IPM machine design methodology assisted by optimization algorithms" IEEE Energy Conversion Congress and Exposition (ECCE), 2012, vol., no., pp.3686-3691, 15-21 Sept. 2012.

[11] Matsuo, T.; Lipo, T.A., "Rotor design optimization of synchronous reluctance machine," Energy Conversion, IEEE Transactions on , vol.9, no.2, pp.359,365, Jun 1994

[12] Seok-Hee Han; Jahns, T.M.; Zhu, Z.Q., "Analysis of Rotor Core Eddy-Current Losses in Interior Permanent-Magnet Synchronous Machines," Industry Applications, IEEE Transactions on , vol.46, no.1, pp.196,205, Jan.-feb. 2010

[13] Seok-Hee Han; Jahns, T.M.; Zhu, Z.Q., "Design Tradeoffs Between Stator Core Loss and Torque Ripple in IPM Machines," Industry Applications, IEEE Transactions on , vol.46, no.1, pp.187,195, Jan.-feb. 2010

[14] Pellegrino, G.; Guglielmi, P.; Vagati, A.; Villata, F., "Core Losses and Torque Ripple in IPM Machines: Dedicated Modeling and Design Tradeoff," Industry Applications, IEEE Transactions on , vol.46, no.6, pp.2381,2391, Nov.-Dec. 2010

[15] David H. Wolpert and William G. Macready "No Free Lunch Theorems for Optimization", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, APRIL 1997

[16] F. Neri and V. Tirronen, "Recent advances in differential evolution: A review and experimental analysis," Artif. Intell. Rev., vol. 33, nos. 1–2, pp. 61–106, 2010.

[17] http://www.mathworks.it/matlabcentral/fileexchange/

[18] Xola B. Bomela, and Maarten J. Kamper, "Effect of Stator Chording and Rotor Skewing on Performance of Reluctance Synchronous Machine", IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, VOL. 38, NO. 1, JANUARY/FEBRUARY 2002.

[19] Rody Oldenhuis, "GODLIKE - A robust single-& multi-objective optimizer", 24 Jul 2009, [Online], available: http://www.mathworks.it/matlabcentral/fileexchange/24838-godlike-a-robust-single-multi-objective-optimizer

[20] David Meeker, "Finite Element Method Magnetics", Ver. 4.2 User's Manual, February 5, 2009, [Online] available: http://www.femm.info/Archives/doc/manual.pdf

[21] Infolytica MagNet: Design and analysis software for electromagnetics, [Online] available: www.infolytica.com