Design Optimization of an Outer Rotor PMSM for a Drive Cycle using an Improved MODE Algorithm for a Lightweight Racing Vehicle

Muhammed Muhsin Cosdu
Dept. of Electrical & Electronics Engineering
Abdullah Gul University
Kayseri, TURKEY
muhammedmuhsin.cosdu@agu.edu.tr

Ahmet Furkan Hacan
Dept. of Electrical & Electronics Engineering
Abdullah Gul University
Kayseri, TURKEY
ahmetfurkan.hacan@agu.edu.tr

Burak Tekgun
Dept. of Electrical & Electronics Engineering
Abdullah Gul University
Kayseri, TURKEY
burak.tekgun@agu.edu.tr

Abstract— **Hub motors are widely used for light-weight electric drives. The aim of this paper is to design a highly efficient out-runner permanent magnet synchronous motor (PMSM) for a specific drive cycle in order to use it in an electro mobile contest called the "Efficiency Challenge". A multi-objective differential evaluation (MODE) algorithm is used to obtain a variety of different design options. The MODE algorithm is altered to incur less computational cost and yield better-distributed results in a comparison with traditional MODE. The alteration is performed in five different aspects: Pareto Front, Selection Algorithm, Population Size, Scaling Factor, and Rectification. The objectives for differential evaluation optimization are minimizing the motor mass and maximizing efficiency for the target drive cycle. The voltage limit and the torque ripple are defined as constraints. The optimization algorithm is written in MATLAB and the finite element analysis (FEA) is conducted in ANSYS/Maxwell 2D. The modified MODE algorithm is optimized for the PMSM with 100 generations and 3282 candidate designs. A well-distributed Pareto optimal solution set is obtained, and a suitable design is selected to be manufactured.**

Keywords— PMSM, Hub Motor, MODE, PMSM Design Optimization, PMSM Geometric Optimization, PMSM Drive Cycle Optimization.

I. Introduction

The aim of this paper is to design a motor for a lightweight electric vehicle for the contest called "Tubitak Efficiency Challenge". As the name indicates, the contest is intended to produce the most efficient electric car. In this context, a light-weight electric motor that operates with high efficiency over a certain drive cycle will be designed. According to [1], technological advancements have made AC motors more relevant for this purpose due to their features which are higher efficiency, power density, and reliability. It is also stated in [1] that the permanent magnet brushless AC motor is the most suitable type to be used in electric cars due to its characteristics of high torque density and efficiency. Therefore, a 3-phase permanent magnet synchronous motor (PMSM) has been chosen for our design and optimization. Since permanent magnet synchronous motors include numerous geometric and electrical parameters [2], [3], it is important to determine the significant design parameters. In a typical approach, initial sizing and finite element analysis (FEA) is performed to verify and tune the parameters for a limited number of objectives [3]. However, most of the time, designing the motor manually using the trial and error method will not be sufficient to determine the best-optimized results. Thus, various optimization algorithms such as particle swarm optimization (PSO) [4], genetic algorithm (GA), differential evolution (DE), response surface analysis (RSA) [5], [6], and the invasive weed method [7], are used for the optimization of electric motors. However, evolutionary optimization algorithms such as PSO, DE, and GA are the most heavily favored [5], [6]. Among the evolutionary optimization algorithms, the differential evolution (DE) algorithm has become increasingly popular over the previous decades, especially, multi-objective DE (MODE) [8], [9] to obtain a set of optimum designs for electric machines with multiple objectives [3], [4], [9]. In contrast, the traditionally used MODE algorithm is not specified for geometric motor optimization. In order to obtain better results from the optimization, some alterations are required including improved Pareto front approaches and selection algorithms, variable population size and scaling factors, and some kind of rectifications. These alterations are thoroughly described under Section VI. On the other hand, the "Efficiency Challenge" contest is based on the full track efficiency of the vehicle. Efficiency calculated at constant torque and base speed is not representative of track efficiency [10]. Since the vehicle will go over the same track for 30 laps, drive cycle optimization will be highly beneficial for this project. Implementation of drive cycle optimization into MODE brings some challenges like calculation of the drive cycle of the race track and configuring the FEA software to compute drive cycle efficiency, which heavily increases the computational cost [9]. To lower these costs, some solutions such as elimination of ineligible results and pre-optimization that concerns a single operating point to ensure constraints, will be applied, and tested.

This paper is organized as follows: Section II provides general explanations about the DE and MODE, the MODE of the PMSM is introduced in Section III. Section IV gives details of the drive cycle determination and drive cycle efficiency determination in MODE. The required alterations in the traditional MODE algorithm are detailed in Section V. The results and discussion as well as the conclusions are given in Sections VI and VII.

II. Alterations on the Differential Evolution Algorithm

A. General Explanations about DE and MODE

The DE algorithm is generally used for optimizing some parameters to set the cost between certain boundaries. This cost function is also called the objective. In order to accomplish that goal, generally, a cost function consisting of $(X\text{-}1)^{th}$ degree polynomial is defined. (X being the parameter count here.) The algorithm creates a random parent matrix with N (Number of populations for each generation.) lines and X columns. The X values are inserted into the polynomial and according to them, the cost value is calculated. In order to reduce this cost, the algorithm creates another matrix called

978-1-7281-6264-5/20/$31.00 ©2020 IEEE

a mutant, which is created by a combination of the first vector. This combination takes a value for each X value from A^{th} and B^{th} line from the parent matrix and takes the difference, multiplies this difference with a value called the scaling factor, and then adds this value to the C^{th} line from the parent matrix. For each line and column of the mutant vector, this process is repeated. The last step for creating the child vector is crossing over, which determines the values. The determination is done by comparing a random number to a predefined value, and if it is higher than this value, determined line and column values are taken from the parent, otherwise, the value is taken from the mutant matrix. The predetermined value is for setting the probability for the child vector matching the mutant vector. After this step, the cost values for the child vector are calculated and compared with the parent vector line by line. If the cost in the child vector is lower than the parent vector, that line is inserted into the new parent vector which is called the selection. Creating a child matrix, solving this matrix, and selecting a new parent matrix named simply as the generation. The algorithm ends after some predetermined number of generations or whenever the results are lower than the predetermined cost threshold. However, the case of MODE is somewhat different from DE. The general working principle is the same, but since there are two or more cost functions that have a trade-off between them, the optimization seeks to reduce all cost values. Also, for the outcome of the MODE, a variety of results that have a distribution between different cost functions are expected. Yet, obtaining a well-distributed result can be rather challenging thus making some alterations on the algorithm is crucial.

B. Alterations to the Pareto Front

The first insufficiency of the DE Algorithm concerns its Pareto front approach. Especially in MODE, a general approach involves taking the last parent as the Pareto front. However, this approach contrasts with the basic idea of MODE, which involves obtaining a well-distributed result within different objectives. In general, the optimization is dominated by an objective and thus the results of the last parent matrix are inevitably biased in favor of this dominating objective. Moreover, since the comparison between the parent matrix and the child matrix is performed with respect to the same line, superior results can be omitted from the last parent [11]. As a detailed explanation, assuming the first line of the parent matrix has a well- optimized result and the second line of the parent matrix does not yield a desirable result. After the solution of the child matrix, in the first line, there is even more desirable result, however, the second line has a worse result. The DE algorithm will take the first line of the child matrix to the new parent matrix rather than the second line of the child matrix. On the other hand, the well-optimized result in the old parent is lost during the optimization. Hence, a new Pareto front algorithm is developed to obtain nicely distributed results which has the best results covered during the optimization process.

In this algorithm, a new resizable matrix is defined for the Pareto front. Every generation, all lines of the parent vector, is checked whether there is a better result than those already in the Pareto front matrix. Also, since the Pareto front vector is resizable, the distribution of the results is better especially if the optimization is run for many generations. The comparison between this approach and the common MODE algorithm is conducted and the redefined Pareto front approach was found to yield better and broader results than the common MODE.

C. Alterations about Selection

Until this point, only the objectives have been mentioned with respect to the goal of the DE, however, in engineering applications, there are also constraints [11]. Especially in Motor Design, there are constraints about the geometry or Motor Driver. (Voltage, Current, etc.) The main difference between the objectives and constraints is that objectives are sought to be reduced in contrast with constraints which are held within certain boundaries. Adding constraints to the algorithm significantly affects the selection step and there are different options for integration, some can focus on objectives first and constraints second, and some can do vice versa. Since the optimization of the motor in this project focuses on the "*Efficiency Challenge*" contest, which has limitations in the rulebook [12], the selection algorithm is set to check the constraints firstly and then check the objectives. Also, developing a motor outside of these limitations would be pointless, therefore the selection algorithm altered to have a strict rule for clarifying constraints within their prescribed boundaries. This approach is very useful as a means of obtaining only the acceptable results from the optimization.

On the other hand, reviewing the objectives is also challenging due to the fact that MODE has several objectives that have a trade-off between one another. To get a well-distributed result, the parent matrix should not gather around a specific objective. The effortless solution can involve checking the objectives in order. In this approach, if the population in the child matrix has a better result for any of the objectives, it becomes a parent. However, checking the objectives with a specific order also creates a dominance between the first checked objective and the last checked objective. To decrease this effect, the selection algorithm splits the parent matrix into two and performs checking on one side from the first objective to the last objective. For the second side, the algorithm checks the results from the last objective to the first objective. This approach is highly successful if the number of objectives is low.

D. Alterations about Population Size

In the common DE algorithm, population size is fixed in each generation. Although this situation is easy and effective for basic DE, with the addition of constraints and using it on geometric motor design optimization, the situation changes. Since, the proposed approach first tries to meet constraint boundaries, using a large population size in the beginning, is highly ineffective especially in terms of the number of generations ergo time. Also increasing the number of objectives and parameters has increased the computational cost [13]. Therefore, to increase the effectiveness [13] the optimization is modified to have a different population size. Initially, the optimization only attempts to meet constraint boundaries and thus using a low number of populations is more effective. However, as the generations pass, the main goal switches to obtaining more distributed results, so the population size should be increased. Also, to balance the computational work, there should be a limit for the population size. The optimization used in this project starts to increase population size to a certain limit at a predefined generation number. This approach saved a significant amount of time for optimization to converge.

E. Alterations to the Scaling Factor

Assuming a fixed scaling factor value makes it hard for optimization to work in small values, which affects the system greatly. Especially in the last quarter of the generations, obtaining finely tuned values with a high scaling factor is challenging. However, setting a low value for the scaling factor also does not lead to a preferred outcome since it results in a narrow distribution between different objectives in MODE. Thus, a variable scaling factor value is defined, which decreases in each generation. This approach helps the optimization to have a well-distributed parent in the first generations and well-optimized results for the last generations.

F. Alterations about Rectification

Another commonly encountered situation in MODE is that the parent matrix can become stuck in a single desirable place having a narrow distribution or wander from desired states because of some objective and constraints. In order to find a solution to this problem, the altered Pareto front approach was quite helpful. Since the Pareto front matrix only consists of the comparably best results, it can be directly inserted as the new parent. However, to ensure a well-distributed result from this approach, a dominance filter is used on the Pareto front matrix. The dominance filter removes the entries which have absolutely better results in terms of all objectives. Consequently, only the best results are kept in the output which can be used as a new parent to have even better results in the following generations. Nevertheless, using this approach for each generation affects the distribution negatively, so this process is only done in a predefined number of generations. This approach helped optimization to reduce the number of generations to converge. A similar approach for assuring dominant results in the parent matrix is evaluated in [14].

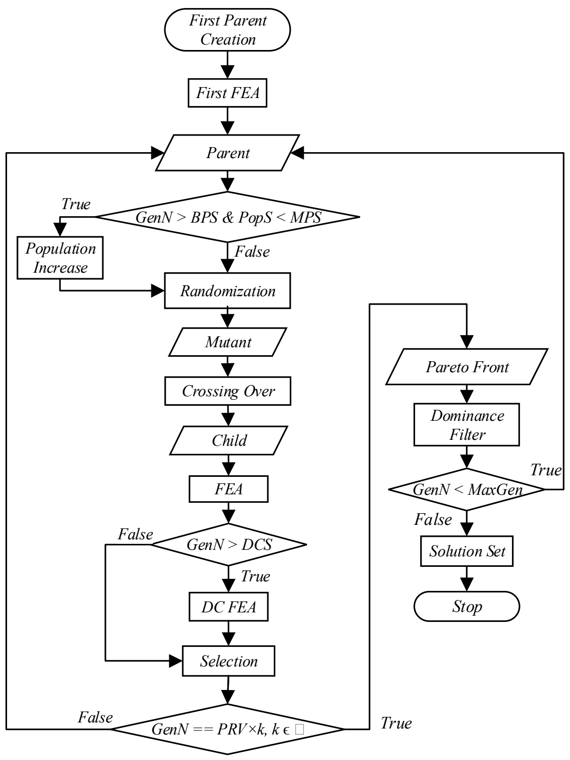

Fig. 1. Flowchart of Altered MODE.

G. General Summary of Optimization

To sum up the algorithm in general, as shown in Fig. 1, this begins with generating a random parent, then solves the parent vector with FEA, followed by generating the Mutant vector. If the generation number (GenN) is higher than a predetermined number (DCS), an elimination algorithm ignores the ineligible results and remaining iterations evaluated with drive cycle FEA (DC FEA). After these steps, the child vector is created by crossing over between mutant and parent vector. The child vector is also solved with FEA and compared with the parent in the selection algorithm. Eligible results are chosen for the new parent and Pareto front. If this is a generation for retraction (GenN == PRV×k, k ϵ □, where PRV is predetermined retraction value), the results of the dominance filter (which evaluates result from Pareto front) update the parent vector. Otherwise, the output of the selection algorithm directly creates the new parent vector. With respect to insertion within the new generation, the generation number is compared with another predetermined number (BPS) which checks whether population size should increase or not. After this step, the new mutant vector is created according to the new population size. If population size reaches its predetermined limit (MPS), the population size (PopS) is kept fixed at maximum.

III. PMSM Geometric Optimization With Improved MODE

Implementing MODE for geometric optimization is a crucial factor for this study. The first requirement for geometric optimization is getting the model ready. Since ANSYS Electromagnetics Suite's Maxwell 2D module is used for FEA, the geometric definitions have numerous parameters which optimization cannot handle in an efficient manner. Therefore, the geometric design is simplified to eight parameters which are highly effective for the motor. Visual representation and explanations can be found in Fig. 2 and Table 1 respectively.

Conversely, sometimes optimization requires some non-geometric parameters. In particular, if the working conditions on the motor are adjustable, the optimization can be set to find the best outcomes around that adjustability limits. In order to do so, current density, maximum current is also parametrized.

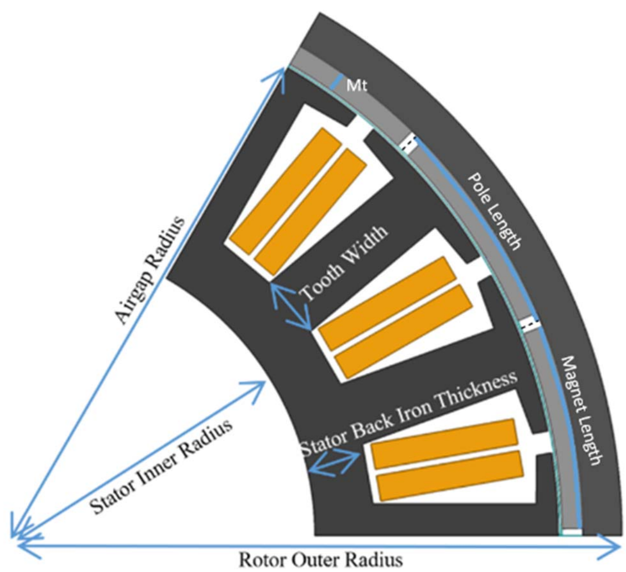

Fig. 2. Visual representation of geometric design parameters.

TABLE I. OPTIMIZATION PARAMETERS

Design Parameter	Definition	Boundaries
Kg	$\frac{\text{Airgap Radius}}{\text{Rotor Outer Radius}}$	0.8 – 0.9
Ks	$\frac{\text{Stator Inner Radius}}{\text{Airgap Radius}}$	0.55 – 0.825
Ksl [mm]	Stator Back Iron Thickness	5 – 20
Me	$\frac{\text{Magnet Length}}{\text{Pole Length}}$	0.8 – 0.975
Mt [mm]	Magnet Thickness	3 – 7
Tw [mm]	Tooth Width	5 – 25
g [mm]	Airgap	0.5 – 1.5
J [A/mm^2]	Current density	3 – 6
Irms [A]	Maximum RMS current	5 – 15

Rotor outer radius is specified as 275 mm since the motor is expected to mount inside the wheel. Two other parameters are not included in the table shown above. One of them is stack length which is not generated by MODE. Since the motor is expected to provide 25 Nm torque as the maximum torque as it operates with 11 A of RMS current, firstly, stack length is taken as 1mm and the torque has been calculated at that point and then the stack length becomes the torque for 1 mm stack divided by 25 Nm which is the required torque. The second parameter that is not listed in the table is the number of turns of the coil. For the sake of getting the highest torque possible from the motor, the number of turns is determined as the highest possible value with respect to current, current density, and slot area. Expression (1) to calculate the number of turns for each variation is shown below. Due to manufacturing capabilities, fill factor is taken as 0.35.

$$No\ of\ Turns = \frac{J \times I_{rms}}{Slot\ Area\ \times Fill\ Factor} \quad (1)$$

As mentioned before in the general explanations about MODE, the algorithm works based on a cost function that has several objectives. The objectives of this project were efficiency and torque density. However, due to the fact that maximum torque value and power are predetermined (25 Nm at a speed of 750 rpm), the second objective is simplified to motor mass. The project also had two constraints, which were line voltage, and torque ripple, because of the limitations of the "*Efficiency Challenge*" rulebook [12]. The limits and calculation methods of these constraints are given in Table 2.

TABLE II. CONSTRAINTS

Design Constraint	Calculation Method	Boundaries
Line Voltage	$\max(Induced\ Voltage) + I_{Rms} * r_{copper}$	175 – 180 (V)
Torque Ripple	$\frac{\text{Peak To Peak Torque}}{\text{Avarage Torque}}\ x\ 100$	0% – 25%

IV. DRIVE CYCLE OPTIMIZATION

A. Drive Cycle Determination

In order to maintain the most efficient possible race at the track, the motor has to be optimized for the duty cycle of the track. A duty cycle should have been created according to the demands of the track in Fig. 3 and the rulebook [12] of the contest. It is mandatory to complete 30 laps in 65 minutes. Therefore, using a similar approach with [15], the speeds of the sectors should have been determined as each lap will finish in 120 seconds.

Fig. 3. Efficiency Challenge race track.

After the speed points have been determined, torque at those points has been calculated. To achieve that, the forces on the car through the circuit with determined speeds have been calculated with (2) (3) [16].

$$\begin{aligned} Total\ Force = {} & Accelaration \times Mass \\ & + Rolling\ Resistance \\ & + Aerodynamic\ Drag\ Force \end{aligned} \quad (2)$$

$$Torque = Total\ Force \times Tyre\ radius \quad (3)$$

As it is shown in the expression above, some parameters are needed to calculate the forces, (such as mass, drag coefficient, etc.). These parameters are taken from the designed vehicle. An algorithm has been designed to get a drive cycle that maintains the least energy consumption in the contest with those numbers.

Mathematical models of rolling resistance force (Fr) (4) (5) and aerodynamic drag force (Fd) (6) as functions of velocity are used in the algorithm [16]. Forces due to the slope of the track are considered as zero since the altitude difference on the track is negligible. Inertial forces are also negligible since there is no transmission in the vehicle.

$$C_r = 0.005 + \frac{1}{p} \times \left(0.01 + 0.0095 \times \left(\frac{V}{100}\right)^2\right) \quad (4)$$

$$F_r = C_r \times Mass \times g \quad (5)$$

$$F_d = C_d \times \rho \times A \frac{V^2}{2} \quad (6)$$

Using the equations above, forces on the car have been calculated through one lap for each speed. With the forces having been calculated on the car, it is straightforward to determine the torque, and thus the power for each point. Furthermore, by integrating power through one lap, the total energy needed to complete the race can be estimated. Determined Torque-Speed points and duration values are shown in Fig. 4.

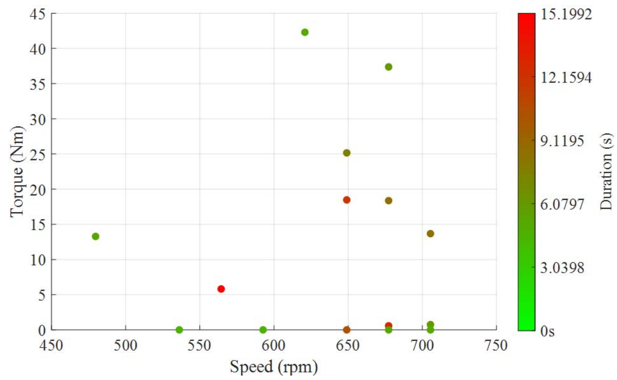

Fig. 4. Torque – Speed and Duration Graph

B. Implementation with MODE

In order to implement a drive cycle efficiency calculation as a cost for the MODE algorithm, some alterations are required in the ANSYS Electromagnetics, since 2D models' losses are only calculated at a specific current and speed. Due to torque and current being proportional and maximum torque being set by the stack length to be fixed at 25 Nm, the required currents are calculated for torque requirements. In order to obtain the drive cycle efficiency, the model is solved for different speeds and current points and added with respect to their time ratio in the drive cycle.

Taking all Torque-Speed points for the drive cycle efficiency calculation is not practical, since the points which have a higher duration, have more impact on the total energy consumption. Also, the points where torque is very close to 0 are ignored as the motor will not waste energy at these points. Considering these facts, some Torque-Speed points in the drive cycle were eliminated from our efficiency calculation. The rational duration of each point is used to calculate the weight for averaging. Since the vehicle will have two identical electrical motors inside the rear wheels, the determined torque is divided by two for it to be used in the drive cycle optimization. Torque - Speed Points used in the calculation of drive cycle efficiency are given in Table 3.

TABLE III. TORQUE – SPEED POINTS

Required Torque (Nm)	**Required Speed (rpm)**	**Weight**	**Required Torque (Nm)**	**Required Speed (rpm)**	**Weight**
6.6	480	0.095	18.7	675	0.095
2.9	565	0.238	9.2	675	0.143
21.1	625	0.079	6.9	705	0.143
9.2	650	0.190	25	750	0.017

Another computing time-saving approach was using a selection algorithm between the solution for constant torque and base speed (25 Nm @ 750 rpm) and the drive cycle solution. Because the drive cycle solutions exert a high computing power, only solutions that fit inside the constraints are solved for the drive cycle.

V. RESULTS & DISCUSSION

In this study, the improved MODE algorithm given in Section V was implemented in MATLAB and all the FEA analyses were performed in Ansys Maxwell. The communication between these software is established through Windows ActiveX commands.

The optimization was operated for 100 generations with a population number that varied between 12 and 48. A total of 3282 candidate designs were simulated. The results of the solved variations can be seen in Fig. 5 and Fig. 6. The optimization took nearly 16 hours with an Intel i5 6600K processor working at 4.1 GHz.

First 25 generations had a population of 12, and after that the population increased by one for each generation until the population reached 48. For the first 50 generations, the optimization was set to work only at a single-point optimization which was determined as 25 Nm at 750 rpm. This approach saved iterations and computing time by obtaining a parent vector that met the predefined constraints. The reason for the drive cycle solutions not having a well-distributed shape in comparison with the single point efficiency graph is that only solutions that were compatible with the constraints were solved for the drive cycle. Since the Line Voltage constraints are strict, results cannot vary.

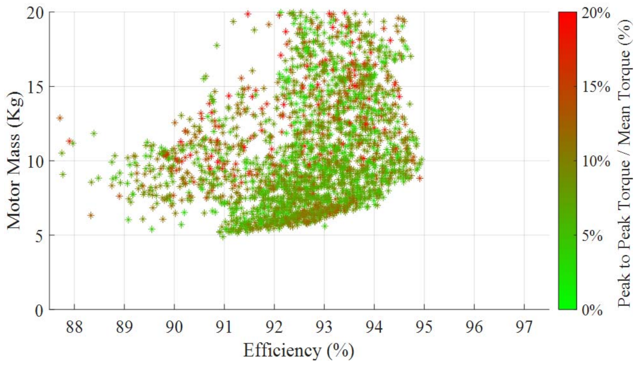

Fig. 5. Solutions for single-point efficiency.

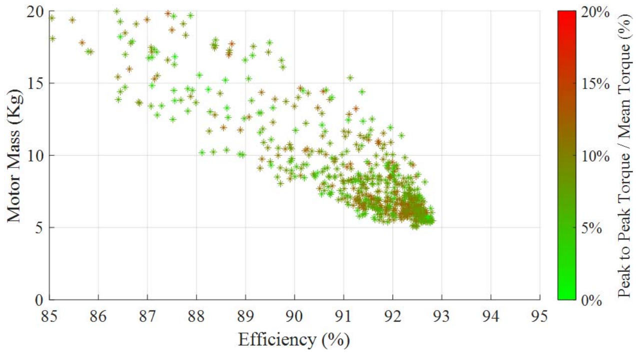

Fig. 6. Solutions for drive cycle efficiency.

However, solutions seemed to cluster around 93% efficiency and 6 kg weight due to the decreased scaling factor approach. This approach helps to find solutions with lower ripple around the best results.

A different alteration on the MODE algorithm is emphasized in Fig. 7 and Fig. 8. The difference between the altered Pareto front and the last generation can clearly be observed since the Pareto front has a large number of different solutions, some of which are superior to the last parent. Surely, the last generation generally has the best results, however the Pareto front also has the best results and some other choices which can be useful for the actual build of the motor in terms of thermal and mechanical specifications.

The candidate that had the highest drive cycle efficiency from the Pareto front was chosen as the final design. The optimal parameters and the resulting performance measures are given in Table 4 and Table 5. Fig. 9 shows the efficiency map of the optimized result and the drive cycle points used in the optimization.

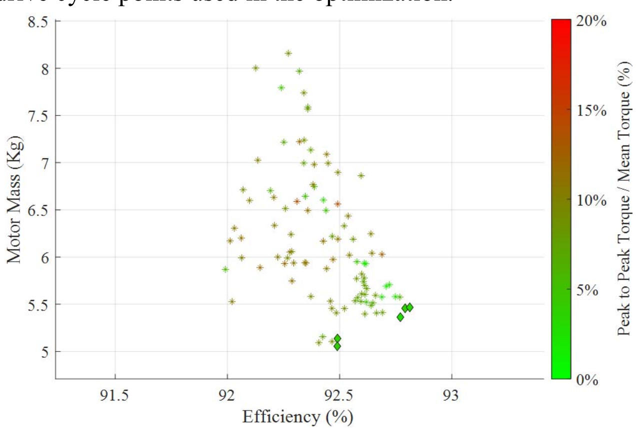

Fig. 7. Pareto Front of the optimization (*Diamond shapes are the results of the dominance filter).*

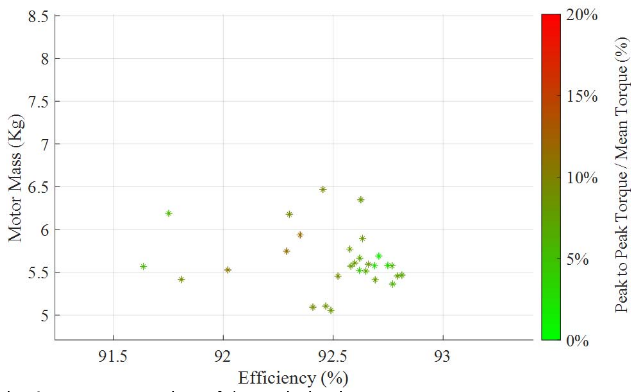

Fig. 8. Last generation of the optimization.

TABLE IV. OPTIMIZED SOLUTION PARAMETERS

Design Parameter	Values	Design Parameter	Values
Kg	0.8725	g [mm]	0.5
Ks	0.5922	J [A/mm^2]	6
Ksl [mm]	10.9161	Irms [A]	10.9493
Me	0.9741	Motor Length [mm]	15.3649
Tw [mm]	9.5924	Number of Turns	77
Mt [mm]	3.8714		

TABLE V. OPTIMIZED MACHINE PERFORMANCE

Result	Values
Maximum Efficiency [%]	94.6519
Drive Cycle Efficiency [%]	92.7920
Mass [kg]	5.4570
Peak Line Voltage [V]	177.70
Torque Ripple [%]	7.6520

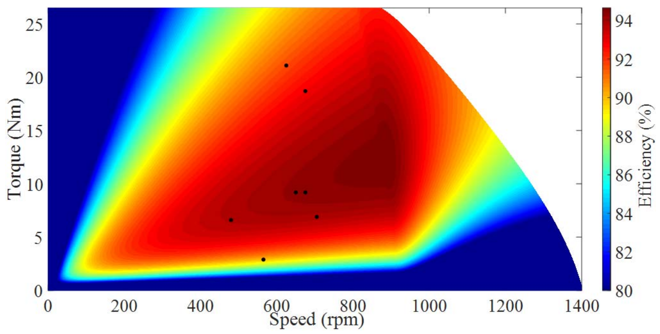

Fig. 9. Efficiency map and drive cycle points.

VI. CONCLUSIONS & FURTHER WORK

In this paper, an improved multi-objective differential evolution optimization algorithm (MODE) was used to optimize an outer rotor PMSM hub motor for the "Efficiency Challenge" contest. As the name of the contest indicates, the race is intended to determine the most efficient motor on the track over 30 laps in 65 minutes. For this purpose, the motor was parameterized to be used in FEA software, the race-track and racing vehicle parameters were used to determine the drive cycle. The MODE algorithm was improved to make a geometric PMSM optimization with respect to the drive cycle. The results of this study clearly showed that the alterations made to the MODE algorithm were successful. Each alteration made to MODE affected the performance in a different manner. Some of these decreased the computational cost thus time, some sought to obtain a better distribution of results among the different objectives, and some of ensured that the results of the optimization were within the project limits. The optimization revealed a set of parameters that resulted in 92.79% drive cycle efficiency with 5.45 kg motor mass. In the context of a light-weight car, this result is quite promising. On the other hand, this motor optimization is intended for manufacturing a vehicle for a contest, and therefore production and testing procedures constitute future work for this study. Thermal and mechanical analysis of the motor will ultimately be performed, followed by the production, and testing of the motor.

ACKNOWLEDGMENT

This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 118E172.

REFERENCES

[1] N. Grilo, D. M. Sousa, and A. Roque, "AC motors for application in a commercial electric vehicle: Designing aspects," *Proc. Mediterr. Electrotech. Conf. - MELECON*, pp. 277–280, 2012.

[2] Y. Duan and D. M. Ionel, "A review of recent developments in electrical machine design optimization methods with a permanent-magnet synchronous motor benchmark study," *IEEE Trans. Ind. Appl.*, vol. 49, no. 3, pp. 1268–1275, 2013.

[3] G. Y. Sizov, D. M. Ionel, and N. A. O. Demerdash, "Multi-objective optimization of PM AC machines using computationally efficient - FEA and differential evolution," *IEEE Int. Electr. Mach. Drives Conf. IEMDC 2011*, pp. 1528–1533, 2011.

[4] J. Zhang, H. Wang, L. Chen, C. Tan, and Y. Wang, "Multi-objective optimal design of bearingless switched reluctance motor based on multi-objective genetic particle swarm optimizer," *IEEE Trans. Magn.*, vol. 54, no. 1, 2018.

[5] R. Benlamine, F. Dubas, S. A. Randi, D. Lhotellier, and C. Espanet, "Design by optimization of an axial-flux permanent-magnet synchronous motor using genetic algorithms," *2013 Int. Conf. Electr. Mach. Syst. ICEMS 2013*, pp. 13–17, 2013.

[6] N. Bianchi and S. Bolognani, "Design optimization of electric motors by genetic algorithms," *IEEE Proc. Electr. Power Appl.*, no. 5, 1998.

[7] E. Sayed, M. H. Bakr, B. Bilgin, and A. Emadi, "Geometric Optimization of Switched Reluctance Motors Using an Invasive Weed Method," *2018 IEEE Electr. Power Energy Conf. EPEC 2018*, no. 1, pp. 2018–2020, 2018.

[8] B. V. Babu and M. M. L. Jehan, "Differential evolution for multi-objective optimization," *Congr. Evol. Comput.*, vol. 3, no. 1, 2003.

[9] A. Fatemi, D. M. Ionel, M. Popescu, Y. C. Chong, and N. A. O. Demerdash, "Design Optimization of a High Torque Density Spoke-Type PM Motor for a Formula e Race Drive Cycle," *IEEE Trans. Ind. Appl.*, vol. 54, no. 5, pp. 4343–4354, 2018.

[10] S. Gunther, S. Ulbrich, and W. Hofmann, "Driving cycle-based design optimization of interior permanent magnet synchronous motor drives for electric vehicle application," *2014 Int. Symp. Power Electron. Electr. Drives, Autom. Motion, SPEEDAM 2014*, pp. 25–30, 2014.

[11] Z. Huang, M. Ma, and C. Wang, "An archived differential evolution algorithm for constrained global optimization," *ICSMA 2008 - Int. Conf. Smart Manuf. Appl.*, vol. 1, no. X, pp. 255–260, 2008.

[12] Tubitak, "TUBITAK Efficiency Challenge 2020 Rulebook," *https://challenge.tubitak.gov.tr/assets/efficiency-challenge_2020_eng.pdf*, 2020.

[13] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 38, no. 5, pp. 1270–1293, 2008.

[14] N. Dong and Y. Wang, "A hybrid multiobjective Differential Evolution algorithm based on improved ε-dominance," *Proc. - 2011 7th Int. Conf. Comput. Intell. Secur. CIS 2011*, pp. 24–28, 2011.

[15] A. Z. Nejad, S. Deilami, M. A. S. Masoum, and N. Haghdadi, "Map-based linear estimation of drive cycle for hybrid electric vehicles," *2015 Australas. Univ. Power Eng. Conf. Challenges Futur. Grids, AUPEC 2015*, 2015.

[16] A. O. Kıyaklı and H. Solmaz, "Modeling of an Electric Vehicle with MATLAB/Simulink," *Int. J. Automot. Sci. Technol.*, vol. 2, no. 4, pp. 9–15, 2019.