

POLITECNICO DI TORINO

Master's Degree program in Electrical Engineering



Master's Thesis

Advanced e-machine design using open-access scripting languages with Motor-CAD solving engine

Supervisors:

Prof. Gianmario Pellegrino

Dr. Mircea Popescu

Co-supervisors:

Eng. Simone Ferrari

Eng. Nicolas Rivière

Candidate

Paolo Ragazzo

March 2020

Abstract

Nowadays, Synchronous Reluctance machines represent a competitive replacement of asynchronous motors thanks to higher efficiency, torque density and overload capability. Yet, they are recognised as a machine typology that needs to be further analysed and studied since its design procedures are still not well-established. Indeed, despite the research effort during the last two decades, a standard procedure for their design has not been established yet. The present thesis aims to investigate a possible design process and to provide an original procedure by joint use of two tools: SyR-e and Motor-CAD. Before this thesis work, the two software environments were totally distinct. The initial goal of this research was to make such two software compatible via Matlab scripting, for complementing areas of evaluation not covered in SyR-e, as well as for bench-marking reasons. Thus, the first step was to make the SyR-e motor models exportable to Motor-CAD, run the needed simulations and collect the data back to Matlab. Later on, the new bridge between the two environments has been used to formulate a design process made upon the points of strength of both parties. Finally, the design procedure constructed was tested by re-designing a demanding SyR motor for automotive purpose, which is denominated ReFreeDrive project and realized by the joint effort of Motor Design Limited (MDL) and University of L'Aquila.

Abstract - Italian version

Oggigiorno le macchine sincrone a riluttanza rappresentano un'interessante alternativa dei motori asincroni grazie ad una maggiore efficienza, densità di coppia e capacità di sovraccarico. Tuttavia, sono riconosciuti come una tipologia di macchina tecnologicamente immatura e da analizzare e studiare ulteriormente. Infatti, le sue procedure di progettazione non sono ancora ben definite nonostante gli sforzi di ricerca negli ultimi due decenni. La presente tesi si propone di indagare gli attuali processi di progettazione e di fornirne uno innovativo ed efficace mediante l'uso congiunto di due software: SyR-e e Motor-CAD. Prima di intraprendere il lavoro di tesi, i due strumenti erano totalmente distinti, quindi il primo obiettivo è stato renderli compatibili tramite codici Matlab, al fine di integrare simulazioni non incluse in SyR-e e così come per benchmark. Il primo passo è stato quello di rendere i modelli di SyR-e esportabili in Motor-CAD, per eseguire le simulazioni necessarie e raccogliere i dati su Matlab. Successivamente, l'export creato è stato fondamentale nella formulazione di una nuova procedura di progettazione basata sui punti di forza di entrambi i programmi. Infine, tale procedura di progettazione è stata testata riprogettando un motore SyR ad alte prestazioni per trazione automobilista, realizzato all'interno del progetto Europeo Re-FreeDrive e dallo sforzo congiunto di Motor Design Limited (MDL) e dell'Università dell'Aquila.

Acknowledgement

This master thesis has been carried out within the PEIC (Power Electronics Innovation Center) research group in association with Politecnico di Torino (Italy) and at Motor Design Ltd in Wrexham (United Kingdom), during six months.

Firstly, I would like to express my deepest appreciation to my supervisor Prof. Gianmario Pellegrino for all his assistance and for having conceived the opportunity to have such a valuable thesis experience. Special thanks also to Eng. Simone Ferrari for his essential support.

I am very grateful to Dr. Mircea Popescu and Eng. Nicolas Rivière for introducing me to Motor Design Ltd and for their crucial assistance. I would like to express gratitude to the whole staff of Motor Design Ltd for its constructive comments and for creating a friendly environment.

In addition, I thank my parents, Franca and Guglielmo, and my sister, Alice, for their delightful encouragement, moral support and care.

I am profoundly grateful to my friends and university companions scattered throughout Italy, Norway and United Kingdom, I want to thank them for their support and the amazing experiences we shared. In particular, I have very much appreciated the unforgettable moments had with Luca, Gianluca, Giorgio, Michele, Rosario and Stefano.

Last, I would like to thank Francesca for her lovely patience and for offering me her heartfelt smile whenever was possible.

To all of them is dedicated this thesis.

Contents

Abstract	I
Abstract - Italian version	III
Acknowledgement	V
List of Figures	XI
List of Tables	XV
List of Code Listings	XVII
List of Abbreviations	XIX
1 Introduction	1
1.1 Motivation and content	2
1.2 Introduction of software tools	2
1.3 Motor-CAD Software	4
1.3.1 Motor-CAD sections	5
1.3.2 Motor-CAD project workflow	6
1.3.3 Linked tools	6
1.4 SyR-e Software	7
1.4.1 Software sections	8
1.4.2 Linked tools	9
1.5 Electric motors types overview	10
1.6 Synchronous Reluctance motors	11
1.6.1 Background	11
1.6.2 Principal of operation	11
1.6.3 Control strategy	12
2 Synchronous Reluctance motors design in SyR-e	15
2.1 Main design Rules of Synchronous Reluctance motors	15
2.1.1 Number of flux barrier layers and number of slots combination	15
2.1.2 Flux linkage	16
2.1.3 Electric loading	17
2.1.4 Torque and power factor	18
2.1.5 Design factors x and b	19
2.1.6 dq-axes design	19
2.2 Model refinements	23
2.3 SyR-e design procedure	25

3	Export SyR machines from SyR-e to Motor-CAD	29
3.1	Export of a motor model to Motor-CAD	30
3.2	Simulation of the exported design in Motor-CAD	30
3.2.1	dq axes conventions	31
3.3	New Motor-CAD section in SyR-e interface	32
3.4	Export test	33
3.4.1	Sample motor specification	33
3.4.2	'Export .mot' button	33
3.4.3	'EMag Sim' button	35
4	ReFreeDrive project review	37
4.1	ReFreeDrive project	37
4.1.1	Project overview	37
4.1.2	Requirements	39
4.1.3	Original performance	39
4.2	Proposed design methodology	43
4.3	Preliminary design	45
4.3.1	Fixed parameters	45
4.3.2	Number of pole pairs	45
4.3.3	Number of flux barrier layers and number of slots	46
4.3.4	Hairpin stator winding	46
4.3.5	Possible design solutions	47
4.4	Design in SyR-e	48
4.4.1	Ribs thickness preliminary sizing	50
4.4.2	Design option 1: $q = 4, N_s = 20$	51
4.4.3	Design option 2: $q = 4, N_s = 16$	52
4.4.4	Design option 3: $q = 2, N_s = 18$	54
4.4.5	Design option 4: stranded winding, $q = 4, N_s = 18$	56
4.4.6	SyR-e design outcomes	57
4.5	Design in Motor-CAD	58
4.5.1	Export from SyR-e to Motor-CAD	58
4.5.2	Ribs thickness adjustments	59
4.5.3	Slot area trade-off	62
4.5.4	Electromagnetic module	64
4.5.5	Thermal module	69
4.5.6	Lab module	72
4.6	Comparison	77
4.6.1	700 A_{pk} case	77
4.6.2	800 A_{pk} case	78
5	Conclusions	81

Bibliography	83
Appendix A Code Listening	A-1

List of Figures

Figure 1: Motor-CAD logo	3
Figure 2: SyR-e logo	3
Figure 3: Software logo - (a) FEMM, (b) Matlab, (c) MagNet, (d) AutoCAD	3
Figure 4: Motor-CAD product development life-cycle [5]	6
Figure 5: Rotor geometries included in SyR-e: Circular (a), Seg (b), ISeg (c), Fluid (d), SPM (e) and Vtype (f).	8
Figure 6: Data flow to analyse machine performances [7]	8
Figure 7: GUI - Main data window	9
Figure 8: Synchronous reluctance motors sketch with two (left) and three (right) flux barriers per pole [8]	11
Figure 9: MTPA and MTPV curves (left) and vector diagram [9] (right)	13
Figure 10: Power factor - saliency (iron saturation neglected) [8]	13
Figure 11: Two-pole rotor structure (left), sinusoidal m.m.f applied $f(\xi)$ and magnetic potential distribution along the rotor $r(\xi)$ (right) [10]	16
Figure 12: (a) d-axis flux path - (b) q-axis flux path [13]	17
Figure 13: Sinusoidal stator MMF [9]	21
Figure 14: Stator geometric quantities nomenclature [9]	22
Figure 15: Rotor geometric quantities nomenclature [9]	22
Figure 16: Torque trade-off in the x - b plane - Torque with FEAfix refinement (solid line), Torque before the FEAfix refinement (dashed line)	24
Figure 17: Power factor trade-off in the x - b plane - Power factor with FEAfix refinement (solid line), power factor before the FEAfix refinement (dashed line)	25
Figure 18: Torque and power factor trade-off before FEAfix in the x - b plane	26
Figure 19: Torque and power factor trade-off after FEAfix in the x - b plane	26
Figure 20: SyR machine design procedure in SyR-e: FEA characterization of the final design	27
Figure 21: Export process diagram	29
Figure 22: Axes conventions - in red the Motor-CAD axes, in black the SyR-e axes	31
Figure 23: Motor-CAD tab in SyR-e interface	32
Figure 24: SyR-e exported model	34

Figure 25: Motor-CAD imported model	34
Figure 26: Sample motor exported in Motor-CAD - Winding	34
Figure 27: Sample motor imported in Motor-CAD - FEA regions	35
Figure 28: d -axis flux comparison	36
Figure 29: q -axis flux comparison	36
Figure 30: Torque comparison	36
Figure 31: ReFreeDrive project logo [14]	37
Figure 32: ReFreeDrive project process [14]	38
Figure 33: Optimized rotor shape [16]	40
Figure 34: Electromagnetic torque over the speed range [16]	40
Figure 35: Output power over the speed range [16]	41
Figure 36: q -axis inductance curves [16]	41
Figure 37: q -axis inductance curves [16]	41
Figure 38: Saliency ratio over the speed range [16]	42
Figure 39: Deformation at the airgap [16]	42
Figure 40: Mechanical stress [MPa] [16]	42
Figure 41: Joint procedure	43
Figure 42: Loop for mechanical sizing	44
Figure 43: Case 1 - Motor sketch	46
Figure 44: Case 2 - Motor sketch	46
Figure 45: Design options	47
Figure 46: Low accuracy plane provided by ' <i>syrmdesign</i> '	49
Figure 47: High accuracy plane provided by ' <i>FEAfix4</i> '	49
Figure 48: Option 1 - Power factor and shaft power curves at different peak current	51
Figure 49: Option 1 - Torque	52
Figure 50: Option 1 - Peak phase current	52
Figure 51: Option 2 - Power factor and shaft power curves at different peak current	53
Figure 52: Option 2 - Torque	53
Figure 53: Option 2 - Peak phase current	53
Figure 54: Option 3 - Power factor and shaft power curves at different peak current	55
Figure 55: Option 3 - Torque	55
Figure 56: Option 3 - Peak phase current	55
Figure 57: Option 4 - Power factor and shaft power curves at different peak current	56
Figure 58: Option 4 - Torque	57
Figure 59: Option 4 - Peak phase current	57

Figure 60: Option 3 - Machine model export to Motor-CAD	58
Figure 61: Option 4 - Machine model export to Motor-CAD	59
Figure 62: Ribs preliminary design - Mechanical stress	59
Figure 63: Ribs preliminary design - Mechanical stress (a white region indicates an area exceeding the 340 MPa)	60
Figure 64: Rounding process at the radial ribs by means of AutoCAD	60
Figure 65: Ribs final design - Mechanical stress	61
Figure 66: Ribs final design - Displacement	62
Figure 67: Option 3 - Torque and power factor vs slot length	63
Figure 68: Option 4 - Torque and power factor vs slot length	64
Figure 69: EMag module - Radial and axial view	65
Figure 70: EMag module - 3D view	65
Figure 71: Winding pattern and conductors definition	66
Figure 72: Winding factor harmonics	66
Figure 73: Winding factor harmonics	67
Figure 74: EMag module - Radial and axial view	67
Figure 75: EMag module - 3D view	68
Figure 76: Winding pattern and conductors definition	68
Figure 77: Winding factor harmonics	69
Figure 78: Winding factor harmonics	69
Figure 79: Thermal module - Radial and axial view	70
Figure 80: Thermal module - 3D view	70
Figure 81: Thermal module - Radial and axial view	71
Figure 82: Thermal module - 3D view	71
Figure 83: Option 3 - Peak torque (blue line) and continuous torque (red dots) vs speed	72
Figure 84: Option 3 - Peak power (blue line) and continuous power (red dots) vs speed	73
Figure 85: Option 3 - MTPA angle vs speed, peak operation points (blue line) and continuous operation points (red dots)	73
Figure 86: Option 3 - Power factor vs speed	73
Figure 87: Option 3 - L_d [mH] vs speed	74
Figure 88: Option 3 - L_q [mH] vs speed	74
Figure 89: Option 3 - Efficiency map	74
Figure 90: Option 4 - Peak torque (blue line) and continuous torque (red dots) vs speed	74
Figure 91: Option 4 - Peak power (blue line) and continuous power (red dots) vs speed	75

Figure 92: Option 4 - MTPA angle vs speed, peak operation points (blue line)	
and continuous operation points (red dots)	75
Figure 93: Option 4 - Power factor vs speed	75
Figure 94: Option 4 - L_d [mH] vs speed	76
Figure 95: Option 4 - L_q [mH] vs speed	76
Figure 96: Option 4 - Efficiency map	76

List of Tables

Table 1: RAWP specification	33
Table 2: Export test results in SyR-e and Motor-CAD	35
Table 3: Requirements	39
Table 4: Inputs parameters	45
Table 5: Number of slots and number of flux barrier layers combinations as- essment	46
Table 6: Main data window - inputs parameters	48
Table 7: Other options window parameters	50
Table 8: Windings window parameters	50
Table 9: Materials window parameters	50
Table 10: Preliminary tangential ribs sizing	51
Table 11: Option 1 - Performances	52
Table 12: Option 2 - Performances	53
Table 13: Option 3 - Performances	55
Table 14: Option 4 - Performances	57
Table 15: SyR-e outcomes summary	57
Table 16: Preliminary and final tangential ribs sizing	61
Table 17: Preliminary and final radial ribs sizing	61
Table 18: After and before the ribs adjustments	62
Table 19: Option 3 - Slot length trade-off	63
Table 20: Option 4 - Slot length trade-off	64
Table 21: $700A_{pk}$ - Option 3 and 4 inputs comparison	77
Table 22: $700A_{pk}$ - Option 3 and 4 electromagnetic performance comparison	78
Table 23: $700A_{pk}$ - Option 3 and 4 thermal performance comparison	78
Table 24: $700A_{pk}$ - Option 3 and 4 mechanical performance comparison	78
Table 25: Option 3 $700A_{pk}$ and $800A_{pk}$ - Inputs comparison	79
Table 26: Option 3 $700A_{pk}$ and $800A_{pk}$ - Electromagnetic performance com- parison	79
Table 27: Option 3 $700A_{pk}$ and $800A_{pk}$ - Thermal performance comparison	80
Table 28: Option 3 $700A_{pk}$ and $800A_{pk}$ - Mechanical performance comparison	80

List of Code Listings

Code Listing 1: draw_motor_in_MCAD function	A-1
Code Listing 2: syreToDxfMCAD function	A-5
Code Listing 3: windingSyreToMCAD function	A-6
Code Listing 4: MCADfitness function	A-8
Code Listing 5: simulate_xdegMCAD function	A-9
Code Listing 6: simulate_xdegMCAD function	A-11

List of Abbreviations

EMag Electromagnetic Models

FEA Finite Element Analysis

FEMM Finite Element Method Magnetics

GUI Graphical User Interface

IM Induction Motor

MDL Motor Design Limited

MTPA Maximum Torque Per Ampere

PM Permanent Magnet

PMS Permanent Magnet Synchronous

SyR Synchronous Reluctance

SyR-e Synchronous Reluctance Evolution

VBS Virtual Basic Script

1 Introduction

A rotating electric machine is an electromechanical energy converter, which is capable to convert mechanical power to electricity and vice versa. During such conversion, the machine is subject to mechanical stress and, as any energy transformation, to energy loss. This heat has to be managed and disposed of towards the external environment. For these reasons, the electric machine design is not barely an electromagnetic matter, but it involves mechanical and thermal evaluations. Moreover, the design must contemplate economic aspects, as the construction costs (materials and manufacture) and the operating costs.

In turn, it is clear that the e-machine design is a complex problem which involves multiple facets, usually conflicting with each other. Therefore, the ultimate e-machine design is a compromise between different performance and compliance goals.

The performance of an electric machine depends chiefly on three aspects:

1. e-machine type and geometry (e.g. winding type, rotor type, synchronous or asynchronous operation).
2. materials characteristics (e.g. resistivity, steel grade, maximum operating temperature).
3. geometric parameters (e.g. poles number, dimensional parameters, number of turns).

The goal of the design process is to determine the geometric parameters, given the e-machine type and the material grade to reach the project specifications requested (e.g. efficiency, torque, over-temperature).

To deal with all of these aspects, dedicated software tools are employed. There are numerous software dedicated to the e-machine design, which allow to collect information about magnetic, thermal and mechanical quantities. The thesis deals with two software platforms for electric machines design: the open-source SyR-e (Synchronous Reluctance Evolution) and the commercial Motor-CAD, by Motor Design Ltd. The thesis is part of a joint research project between Politecnico di Torino and Motor Design Ltd.

1.1 Motivation and content

Nowadays, Synchronous Reluctance (SyR) machines represent a competitive replacement of asynchronous motors thanks to higher efficiency, torque density and overload capability. Yet, they are recognised as a machine typology that needs to be further analysed and studied since its design procedures are still not well-established. Indeed, despite the research effort during the last two decades, a standard procedure for their design has not been established yet. The present thesis aims to investigate a possible design process and to provide an original procedure by joint use of two tools: SyR-e and Motor-CAD.

Before this thesis work, the two software environments were totally distinct. The initial goal of this research was to make such two software compatible via Matlab scripting, for complementing areas of evaluation not covered in SyR-e as well as for bench-marking reasons. Thus, the first step was to make the SyR-e motor models exportable to Motor-CAD, run the needed simulations and collect the data back to Matlab. Later on, the new bridge between the two environments has been used to formulate a design process made upon the points of strength of both parties.

Finally, the design procedure constructed was tested by re-designing a demanding SyR motor for automotive purpose, which is denominated ReFreeDrive project and realized by the joint effort of Motor Design Limited (MDL) and University of L'Aquila.

1.2 Introduction of software tools

In this thesis work, mainly two tools were employed to design and analyze electric machines:

- Motor-CAD (paragraph 1.3), a widespread commercial software by Motor Design Limited (MDL).
- SyR-e (paragraph 1.4), an open source software matured by a collaboration between the Politecnico di Torino and the Politecnico di Bari.



Figure 1: Motor-CAD logo



Figure 2: SyR-e logo

In addition to these two design environments, also other tools were utilized for specific designing parts:

- FEMM [1]: Finite Element Method Magnetics (FEMM) is an open source software finite element analysis software package for solving magnetostatic problems.
- Matlab [2]: it is a high-performance language for technical computing, which integrates computation, visualization, and programming.
- MagNet by Mentor [3]: it is adopted for the design of motors, sensors, transformers, actuators, solenoids or any component with permanent magnets or coils. It employs the finite element technique for an accurate and quick solution of Maxwell's equations.
- AutoCAD [4]: it is computer-aided design (CAD) software aimed to create 2D and 3D drawings.

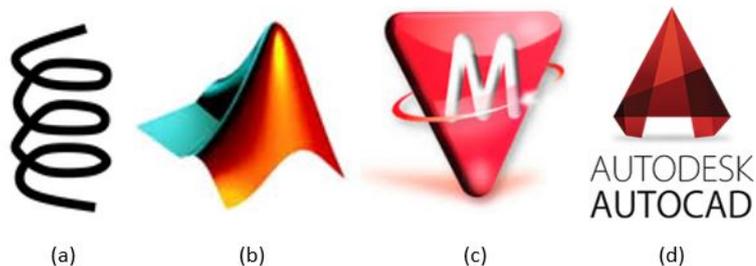


Figure 3: Software logo - (a) FEMM, (b) Matlab, (c) MagNet, (d) AutoCAD

1.3 Motor-CAD Software

Motor-CAD is an advanced software design tools developed by [MDL](#). Initially released in 1998, it is used by major motor manufacturers and universities worldwide. Motor-CAD grants the capability to quickly and easily perform electromagnetic and thermal performance tests on different electric machine designs. It covers several typical types of radial flux rotating electric machines:

- Brushless permanent magnet motor (inner and outer rotor)
- Induction motor
- Synchronous reluctance motor
- Switched reluctance motor
- Synchronous wound field motor
- Permanent magnet DC motor
- Single phase induction motor

Further, several cooling types are implemented as listed below.

- TENV - totally enclosed non-ventilated: natural convection from housing
- TEFC - totally enclosed fan cooled: forced convection from housing
- TV - through ventilation
- TE with internal circulating air (internal air circulating path or water jacket as heat exchanger)
- Open end-shield cooling
- Water jackets (axial or circumferential)
- Submersible cooling
- Wet rotor and wet stator cooling
- Spray cooling (e.g. oil spray cooling of end-windings)
- Direct conductor cooling (e.g. slot ducts with oil)
- Custom cooling configurations addable

1.3.1 Motor-CAD sections

Motor-CAD executes analysis for each of the reported motors through four modules, disclosed below.

- Electromagnetic Models (**EMag**): finite element and analytical methods are adopted to compute the electromagnetic performance of the motor. Such module conducts a range of electromagnetic performance tests on prototype design, including calculation of torque, power, losses, voltages, current, inductances, flux linkages and forces. **EMag** embeds 2D transient or magneto-static Finite Element Analysis (**FEA**) solvers with meshing and boundary conditions automatically set up, allowing various advanced calculations (e.g. eddy current and AC winding losses). Moreover, it is capable to run the simulations with imported custom DXF geometries, a feature that was largely used in this thesis work.
- Thermal model: 3-dimensional lumped-parameter circuit model is exploited to calculate the steady-state and transient thermal characteristics of the motor. The thermal model is solved with all the thermal resistances and capacitances automatically obtained from geometric dimensions and material properties. Motor-CAD features algorithms for forced and natural convection, liquid cooling, radiation and conduction. This module is adopted to optimise the cooling system of a wide variety of motor types and cooling methods. Furthermore, custom cooling configurations might be added to the model.
- Lab module: it combines the electromagnetic and thermal modelling toolboxes to undertake numerous important calculations which facilitate the modelling and optimisation of a motor design over its entire operating range. Due to this module, efficiency maps, loss maps and torque/speed characteristics can be tracked and it is possible to analyse performance over driving cycles. Last, it implements control strategies as maximum torque/amp and maximum efficiency control.
- Mechanical module: it allows centrifugal stress evaluation, based on 2D mechanical FEA. This module enables engineers to consider magnetic and mechanical performance trade-off design optimisation. Template or .dxf import options are available, as well as for the **EMag** module.

Motor-CAD is ActiveX enabled and allows scripting from other applications (e.g. Matlab) to automate the design process. Moreover, a scripting interface allows creating and running Visual Basic Scripts (**VBS**). A script control option allows the script to

be automatically run before or during the solving process. With the "Run before Analysis" option, the model can be set up prior to running calculation; whereas, with the "Run during Analysis" option, the script runs every step of solving process. The latter option grants a lot of flexibility and customization in the use of Motor-CAD, since it enables advanced features as vary input parameters or calculate losses using user-defined equations.

1.3.2 Motor-CAD project workflow

Motor-CAD is mainly used for initial design, topology selection, sizing, analysis and optimisation across the full operating envelope. However, more detailed analysis are available in the latter stages of design due to 3D FEA and CFD tools. A sample product development life-cycle in Motor-CAD is disclosed in fig. 4.

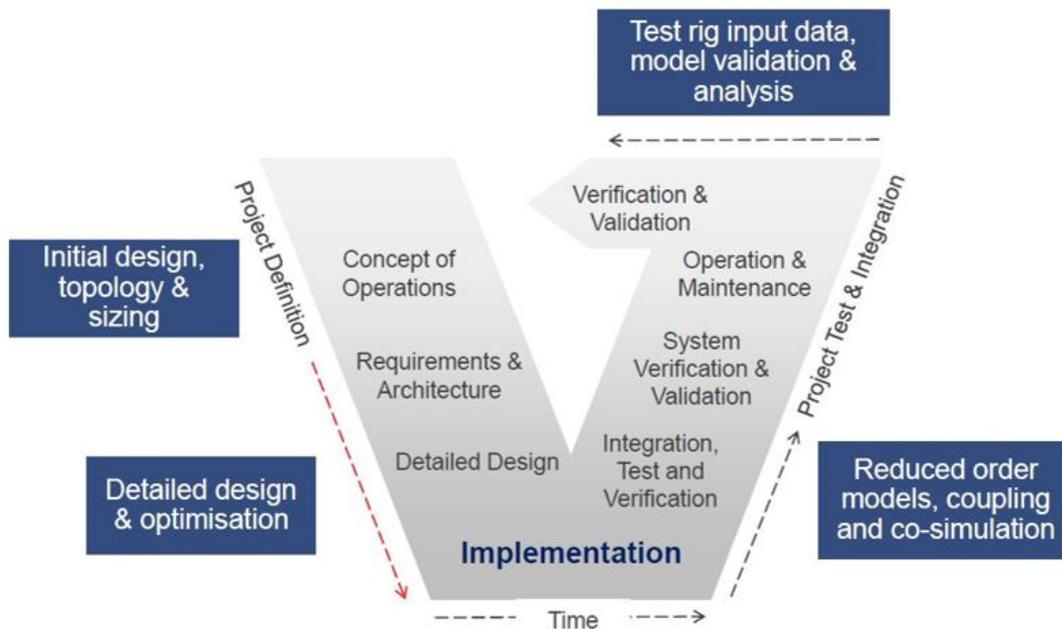


Figure 4: Motor-CAD product development life-cycle [5]

1.3.3 Linked tools

Motor-CAD is linked to several software in order to improve the evaluations performed during the design procedure:

- ANSYS software

- OptiSLang to provide cutting-edge optimisation workflow
- RomaxDESIGNER software for transmission integration and NVH
- GT-Suite for co-simulation and model export
- Matlab and Simulink for electromagnetic and thermal reduced order modelling, further to ActiveX coding.

For further information, it is suggested to refer to Motor-CAD manual [\[6\]](#).

1.4 SyR-e Software

Synchronous Reluctance evolution ([SyR-e](#)) is an open-source code launched in 2014 and developed in Matlab or Octave, aimed to design synchronous reluctance machines automatically by means of finite element analysis and multi-objective optimization algorithms, or using analytical models. The SyR-e project started with two main aims: to investigate SyR motors without prejudices from existing literature and to provide an automatic tool from both expert and non-expert machine designers.

The rotor geometries implemented are listed below and reported in [fig. 5](#).

- Circular flux barrier SyR motors
- Segment flux barrier (Seg and ISeg geometry) SyR motors
- Fluid flux barrier SyR motors
- Surface Permanent Magnet (SPM) motors
- V-type Interior Permanent Magnet (IPM) motors

The data flow to evaluate machine performances is displayed in [Fig. 6](#). A parameterized drawing of the machine is computed by a Matlab script as a .fem file, afterward quickly analysed by FEMM. The main results may be evaluated in Matlab. Such data flow is helpful for automatic design purposes or for the analysis of existing machines. Indeed, hundreds of potential machines may be tested by the multi-objective optimization algorithm or either only the optimized design may be analysed.

FEMM can solve magneto-static problems, whereas motion is emulated with a sequence of static simulations with rotor position and current angles varied accordingly. This process is called static time stepping and it needs the number position and the corresponding rotor angular excursion to be inserted. To optimize the simulation

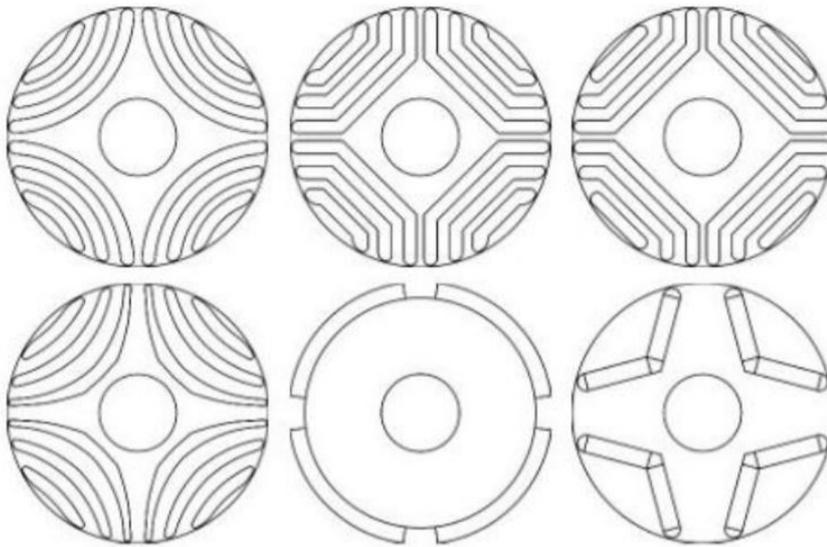


Figure 5: Rotor geometries included in SyR-e: Circular (a), Seg (b), ISeg (c), Fluid (d), SPM (e) and Vtype (f).

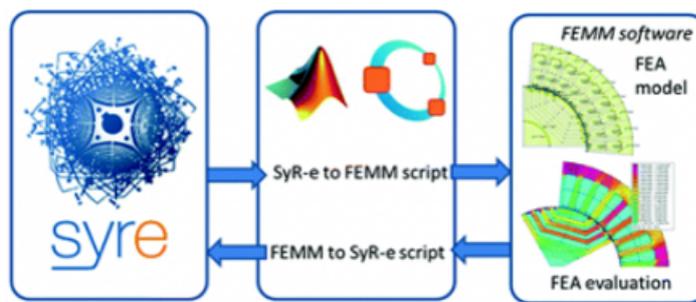


Figure 6: Data flow to analyse machine performances [7]

time, SyR-e adopts the sliding gap boundary of FEMM, i.e. the rotor is not physically rotated but its rotation is provided by the boundary condition, and a series of "fast-FEA" approaches.

1.4.1 Software sections

The parameters in SyR-e are editable via the Graphical User Interface (GUI) (shown in fig. [7]), launched by *GUI_Syre.m*, and they are organized in five sections:

- Main data
- Stator and rotor geometry
- Other options
- Windings

- Materials

Each parameter of the **GUI** is linked to a corresponding Matlab variable with the name reported in round brackets. All of these parameters are stored in the *.mat* file. At the opening of the GUI, the section shown is the one with the main data. The section may be switched, while on the right hand side remains the motor sketch.

To further details regarding the SyR-e interface and features, it is suggested to refer to the user manual **[7]**.

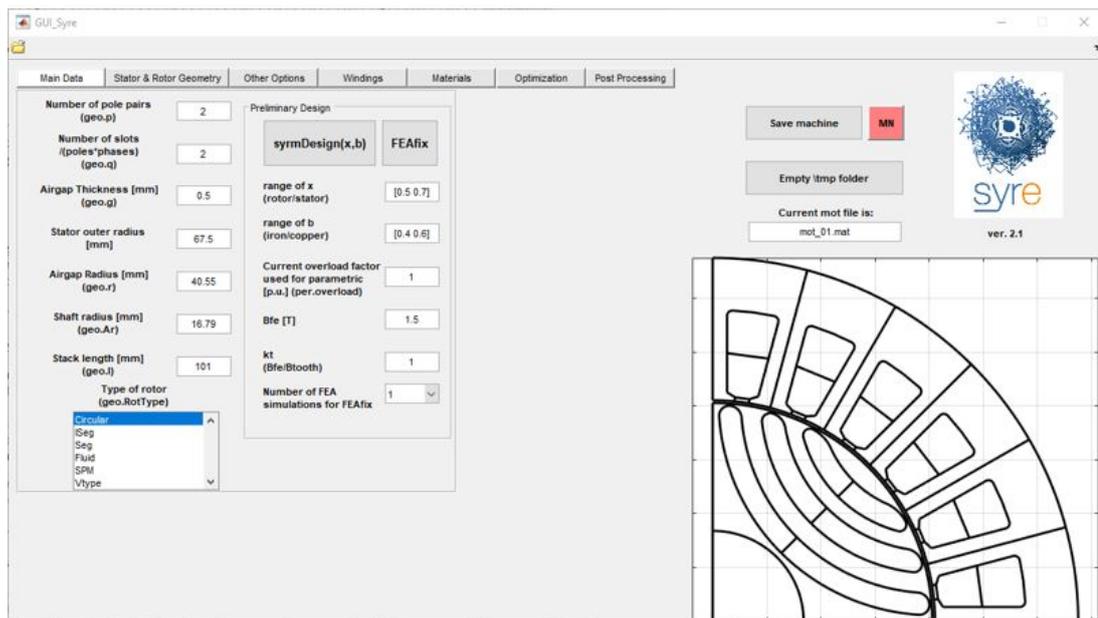


Figure 7: GUI - Main data window

1.4.2 Linked tools

Via Matlab scripts, the following tools are coupled to SyR-e in order to cover different design aspects:

- FEMM
- MagNet
- Motor-CAD (achieved thanks to this thesis as described in Chapter **[3]**)
- AutoCAD (export to *.dxf* format)

1.5 Electric motors types overview

Nowadays, several electric motors typologies are available and largely employed in numerous fields of application; among these, one of growing interest is the automotive sector. Indeed, the diffusion of Electric Vehicles (EV) is still partial, but also growing at increasing pace. This leads the need of a deep electric motors knowledge and of the capability to design electric machines viewed as component of a more complex system. Indeed, with the electric and hybrid powertrain, the electric motors have to satisfy several system level constraints, strictly related to trade-offs in terms of performance, weight, volume and cost. Furthermore, the electric machines design requests evaluations of many different concepts and typologies, which involve different disciplines requiring a multi-physics complex analysis.

Actually, the full electric or hybrid drive trains are equipped with different motors types. For instance, the Chevrolet Volt 2016, the BMW i3 and the Tesla Model 3 employ PM-SyR motors, while the Toyota Prius and the Nissan Leaf use IPM motors, last, the Tesla S and X adopt IMs. Obviously each of these has advantages and disadvantages that have to be taken into account. Concerning the drive control, the field-oriented vector control has been largely studied and used for IMs, which have also the advantages to be naturally de-excited in case of inverter fault, such feature is greatly appreciated for electric traction applications.

While, Permanent Magnet Synchronous (PMS) motors include surface-mounted PM (SPM) and interior PM (IPM) types, guaranteeing best-in-class torque density and efficiency. However, SPMs suffer from eddy-current loss at high speed and require structural sleeves for PM retention. Furthermore, also the motor control has several critical points to be faced, as the flux weakening operation, a variable dc-link voltage related to the batteries physics and a consistent torque response, i.e. independent from rotor temperature. Last, the rare-earth magnet price volatility obstacles the PM motor drives spread, in fact the neodymium and dysprosium price has been unstable and led to an energetic search for alternative machines, as PMS machines with smaller amounts of Permanent Magnet (PM) or with alternative magnets as ferrite. From this point of view, an interesting solution might be the Synchronous Reluctance motors, which provide torque entirely from rotor anisotropy without any magnets.

1.6 Synchronous Reluctance motors

1.6.1 Background

The high efficiency and the low realization cost of the **SyR** machines led them to be view as a valid alternative to induction motors in variable speed industry applications. Nevertheless, their design procedures are not extensively well-known, particularly for the rotor design, whereas the stator is similar to that of Induction Motor (**IM**). Several design procedure have been proposed over the last years, providing a defined guideline in the design algorithm, but not an accurate procedure.

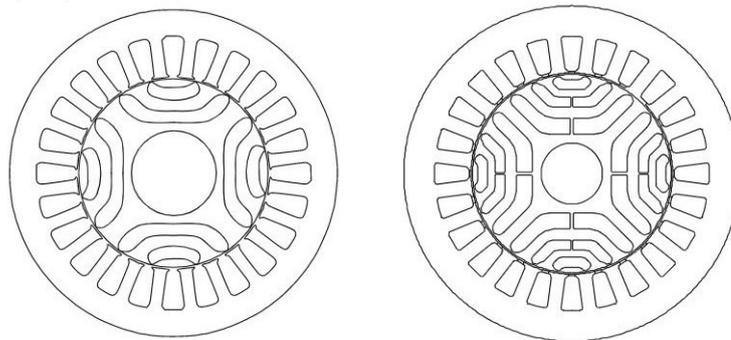


Figure 8: Synchronous reluctance motors sketch with two (left) and three (right) flux barriers per pole [8]

As every machine typology, even **SyR** machines have disparate advantages balanced by some disadvantages. For instance, compared to **PMS**, the **SyR** machines have no **PM**, this reduces the cost of manufacturing, but, on the other side, leads to a lower torque per volume. Whereas, comparing the **SyR** machines to the Induction Machines (**IMs**), the first have greater efficiency and simplified manufacturing, balanced by worse power factor and less well-known design procedures. Moreover, the **SyR** rotors do not present Joule losses, indeed they are denominated cold rotor motors. Such feature allows to meet an higher efficiency merged with an easier management.

1.6.2 Principal of operation

Considering a generic **SyR** machine, it counts two different rotor paths for the flux:

- one is a high permeability path allowing the flux lines to flow in rotor iron paths, parallel to the flux barriers (commonly referred to as d -axis path);

- the second is a low permeability path, in which the flux lines have to cross the rotor flux barriers (commonly referred to as q -axis path).

The rotor design is aimed at finding an optimal trade-off between the design of the high permeance and high reluctance axes, to maximize the reluctance torque and the power factor. However, to mechanically sustain the rotor, the iron bridges have to be included at the ends and sometimes in the middle of each barrier. The bridge thickness has to be chosen as trade-off between good saliency ratio and acceptable maximum mechanical stress.

The general torque expression is shown in eq. [1.1](#), however it can be split in two contributions as in eq. [1.2](#): reluctance torque (needs i_d and i_q) and, if PMs are inserted, also a not null permanent magnet torque.

$$T = \frac{3}{2}p \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \quad (1.1)$$

$$T = \frac{3}{2}p \cdot (\lambda_m i_q + (L_d - L_q) \cdot i_d i_q) \quad (1.2)$$

$$T = \frac{3}{2}p \cdot (L_d - L_q) \cdot i_d i_q \quad (1.3)$$

- Permanent magnet torque - $\lambda_m i_q$: it is zero if the machine is a [SyR](#) without [PM](#).
- Reluctance torque - $(L_d - L_q) \cdot i_d i_q$: with the fixed conventions $L_d > L_q$, i.e. $L_d - L_q > 0$, thus to provide a positive torque (motor mode) the dq currents must have same signs (first and third quadrants).

1.6.3 Control strategy

The motor is commonly current controlled and the optimal control strategies aim to maximize the total torque per power losses (maximum efficiency) or phase current (MTPA). The reluctance torque is maximized with a current angle exactly equals to 45° if saturation is neglected, otherwise the MTPA angle depends on the current amplitude and it overcomes the 45° , as shown in Fig. [1.6.3](#).

Whereas, considering a sample [SyR](#) machine, its vector diagram in fig. [1.6.3](#) shows that even if the q -axis current component is greater than the one in the d -axis, the d -axis flux linkage results to be higher than the q -axis flux linkage. Moreover, from the vector diagram, it is clear a drawback of the [SyR](#) machine, since the voltage

vectors leads the current vector and the power factor angle is quite high, i.e. low power factor. Besides the torque, also the power factor is a function of the saliency ratio as is underlined in fig. 10, where, neglecting the iron saturation, the red curve shows the power factor when the current vector operates along the MTPA trajectory. Referring to a saliency of 10, the power factor is 0.63 in the MTPA. While, the blue curve corresponds to operating conditions that maximize the power factor (tangent of the current angle equals to the square root of the saliency ratio). Here, the power factor, referring again to a saliency of 10, is roughly 0.8.

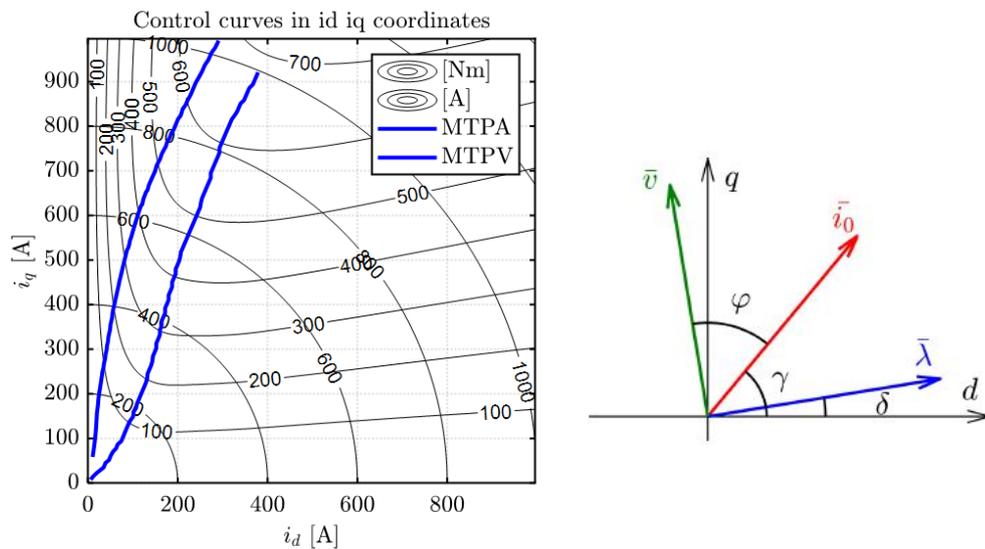


Figure 9: MTPA and MTPV curves (left) and vector diagram [9] (right)

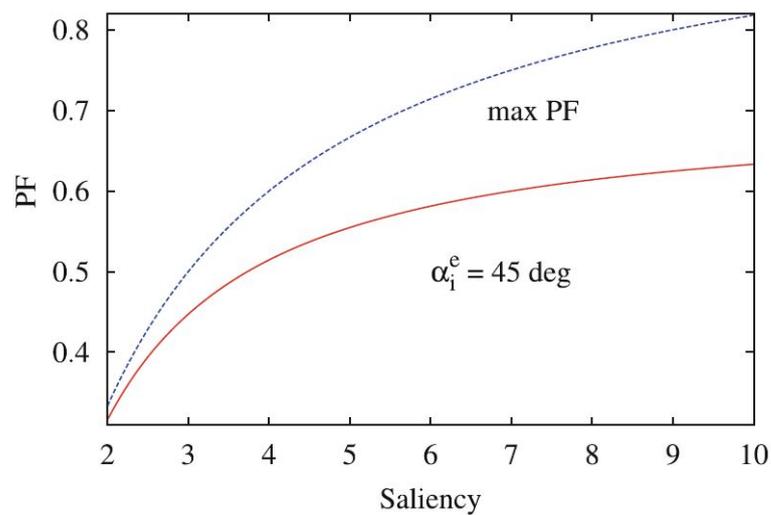


Figure 10: Power factor - saliency (iron saturation neglected) [8]

2 Synchronous Reluctance motors design in SyR-e

As aforementioned, the e-motors design is a complex matter, which involves several aspects and requests multi-physics trade-offs. Furthermore, each SyR machine demands more than one FEA simulation due to magnetic saturation influence and to the mandatory torque ripple minimization. Even the Maximum Torque Per Ampere (MTPA) current phase angle has to be computed by means of an high number of FEA simulations. Overall, the optimization algorithms and the FEA simulations lead to protract the computational times, which can be detained due to the continuous increment in the computation capability of Personal Computers.

2.1 Main design Rules of Synchronous Reluctance motors

This section reports an overview of the fundamental quantities and of the design guideline of SyR motors according to [9] and [10].

2.1.1 Number of flux barrier layers and number of slots combination

Concerning the rotor design, a main point is the choice of the number of the flux barrier layers. A strategy often adopted has been proposed in [10] and it starts from an analytical model. Such guideline bears on the rotor separation points between adjacent segments, i.e. the rotor iron ribs saturated by stator m.m.f allowing different segments to have different magnetic potentials. To clarify the mentioned definition, in fig. [11] the n_r separation points are evidenced by circles or by crosses if virtual. In the reported two-pole structure, there are a total of ten separation points ($n_r = 10$), in which 2 across the q-axis are virtual.

Further, the model is based on the assumption that the rotor flux barrier ends are evenly spaced. Considering Q stator slots, p pole pairs and n_r separation points, it is suggested to fulfill the relationship below.

$$n_r = \frac{Q}{p} \pm 4 \quad (2.1)$$

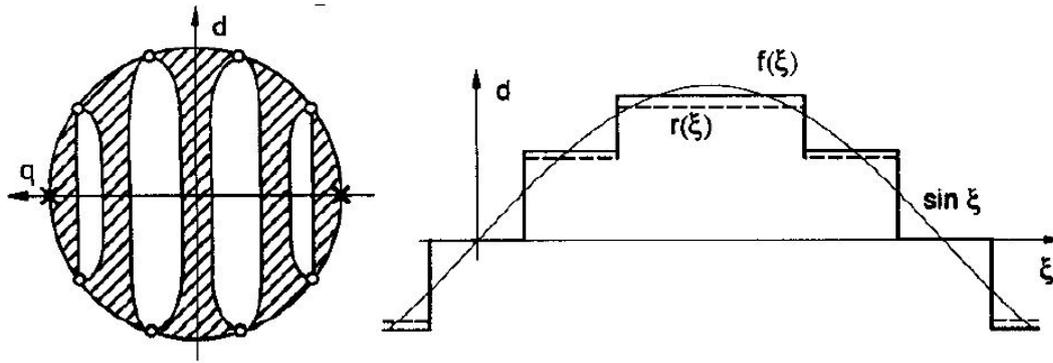


Figure 11: Two-pole rotor structure (left), sinusoidal m.m.f applied $f(\xi)$ and magnetic potential distribution along the rotor $r(\xi)$ [10]

However, in [11], [12] it is demonstrated that the torque ripple can be lowered when the angle of the first smaller flux barrier end is increased with respect to the others. In other words, the flux barriers ends are all evenly spaced but the first one, whose angle can be optimized.

2.1.2 Flux linkage

The fundamental equations describing the dq flux linkages of the SyR machine are disclosed in eq. 2.2

$$\begin{cases} \lambda_d = L_d \cdot i_d = (L_{md} + L_\sigma) \cdot i_d \\ \lambda_q = L_q \cdot i_q = (L_{mq} + L_\sigma) \cdot i_q \end{cases} \quad (2.2)$$

where i_d and i_q are the dq current components, L_{md} and L_{mq} are the magnetizing inductances, whereas the term L_σ corresponds to the leakage inductance, same for both axis. The sum of magnetizing and leakage inductances provides the dq inductances components L_d and L_q . Note that both the inductances depend on geometric inputs and number of turns.

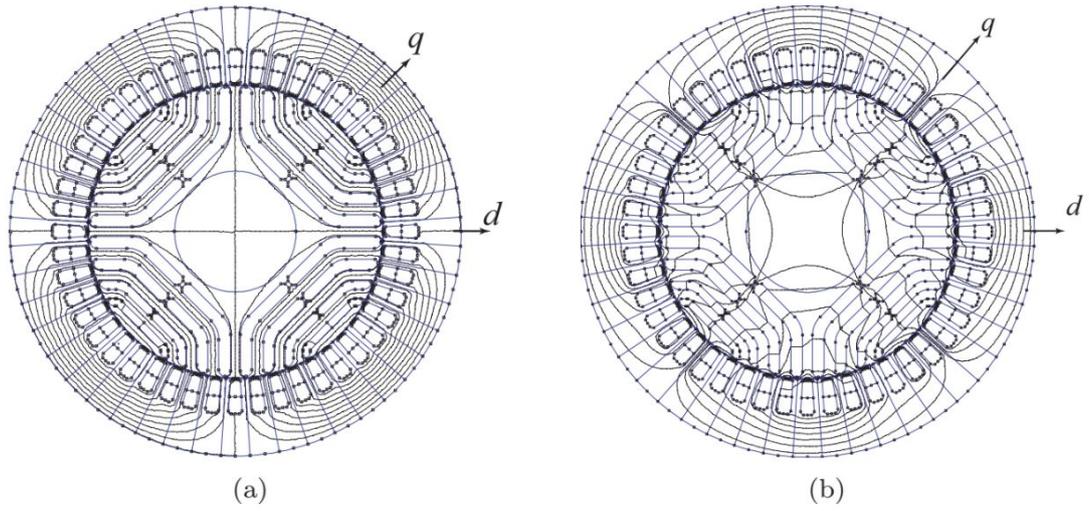


Figure 12: (a) d-axis flux path - (b) q-axis flux path [13]

2.1.3 Electric loading

As first step of the design procedure, the following inputs have to be inserted: the stator outer radius R , the stack length L , the pole-pair p , the slot per pole per phase q and the peak flux density B_{Fe} .

Through the design process, a critical parameter is the thermal loading factor k_j , defining the current loading through eq. 2.4. It can be described as the allowed copper loss by the stator outer surface according to eq. 2.3.

$$k_j = \frac{\frac{3}{2} \cdot R_s \cdot i_0^2}{2\pi RL} \quad (2.3)$$

in which i_0 is the nominal peak current, R_s is the stator resistance, R is the stator outer radius and L the stack length. The k_j parameter depends on the type of the cooling adopted; for example for non-ventilated machines, typical values of k_j are $1.5 \sim 3 \text{ kW/m}^2$, while with forced ventilation it can achieve greater numbers as $5 \sim 12 \text{ kW/m}^2$, however, a water-cooled system may lead to k_j of $12 \sim 20 \text{ kW/m}^2$. In the automotive sector, it is often adopted a water-glycol cooling system, which allows the k_j to overcome the 20 kW/m^2 .

As aforementioned, the rated current i_0 is related to the thermal loading as displayed in eq. 2.4.

$$N_s \cdot i_0 = \sqrt{k_j \cdot \frac{k_{cu}}{\rho} \cdot \frac{L}{L + l_{end}} \cdot \frac{\pi R A_{slot}(x, b)}{9}} \quad (2.4)$$

where N_s is the number of turns in series per phase, k_{cu} is the slot filling factor, ρ the copper resistivity, l_{end} is the end-winding length and A_{slots} is the total cross-section, sum of all the slot sections.

2.1.4 Torque and power factor

Another aspect that has to be pointed out is the torque expression shown in eq. [1.1](#), which can be reassembled as:

$$T = \frac{3}{2}p \cdot (L_{md} - L_{mq})i_0^2 \cdot \frac{\sin(2\gamma)}{2} \quad (2.5)$$

where γ is the phase angle of the current vector with respect to the d axis and the current components in dq axis are $i_d=i_0\cos(\gamma)$ and $i_q=i_0\sin(\gamma)$. Note that the leakage inductance does not influence the obtained torque.

Furthermore, it is also interesting to express the torque as in eq. [2.6](#) to underline the saliency dependence.

$$T = \frac{3}{2}p \cdot i_d i_q \cdot (L_d - L_q) \propto L_d \left(1 - \frac{1}{\xi}\right) \quad (2.6)$$

Thus, the torque requires a high L_d as well as a high saliency ratio ξ defined as $\frac{L_d}{L_q}$. Another fundamental output figure is the power factor, which may be expressed as in eq. [2.7](#)

$$\cos(\phi) = \sin(\gamma - \delta) \quad (2.7)$$

where γ and δ are the angles shown in fig. [1.6.3](#). Like torque, the power factor depends on the saliency ratio, which should be increased as much as possible, as disclosed below.

$$\text{high } \xi \rightarrow \text{high } (\gamma - \delta) \rightarrow \text{low } \phi \rightarrow \text{high } \cos(\phi) \quad (2.8)$$

Furthermore, the power factor can be re-written as a function of inductances and current phase angle γ , according to:

$$\cos(\phi) = \sin(\gamma) \cdot \left(1 - \frac{L_q}{L_d}\right) \cdot \sqrt{\frac{1}{1 + \left(\frac{L_q}{L_d} \cdot \tan(\gamma)\right)^2}} \quad (2.9)$$

With a fixed current amplitude, the power factor is function of L_d , L_q and γ . It is meaningful to notice that, unlike the torque, the power factor depends on the leakage inductance L_σ .

2.1.5 Design factors x and b

Stator and rotor geometries are parametrized by means of few variables, aiming to keep the model simpler as possible. The key design inputs are the rotor/stator split ratio x and the airgap/iron flux density ratio b , defined as follow:

$$x = \frac{r}{R} \quad (2.10) \quad b = \frac{B_g}{B_{Fe}} \quad (2.11)$$

where the r and R are the rotor and stator outer radius, displayed in fig. [14](#) - [15](#). Whereas, B_g is the peak flux density in the airgap and B_{Fe} is the peak flux density in the stator yoke. To obtain torque and power factor as (x, b) functions, it is worth to express all the geometric parameters in terms of x and b .

2.1.6 dq -axes design

Disposed the d and q axes as shown in Fig. [1.6.3](#), the d axis corresponds to the main flux direction, since it represents the maximum inductance direction. The stator and rotor iron cores are designed according to the d -axis equations.

The d -axis airgap flux under one pole can be computed according to eq. [2.12](#).

$$\Phi_d = 2rLB_g \quad (2.12)$$

Rearranging the eq. [2.12](#) to obtain a function of x and b :

$$\Phi_d = 2RLB_{Fe} \cdot xb \quad (2.13)$$

Therefore, the yoke length, imposing the yoke flux equals to half the pole flux, is determinable according to eq. [2.14](#).

$$l_y = \frac{R}{p} \cdot xb \quad (2.14)$$

Whereas, the tooth width is assessed considering the flux in a tooth equal to the airgap flux across one slot pitch ($\frac{2\pi rL}{6pq}$) at peak flux density B_g with a peak tooth flux density of ($\frac{B_{Fe}}{k_t}$). Note that the k_t is usually minor than one to saturate teeth more than the back iron.

Concerning the rotor design, as said for the stator back iron, its flux carriers must bear the airgap flux of half pole. The total size of the rotor carriers must be equal to the yoke size l_t , i.e. the peak flux density in the rotor corresponds to the one in the stator yoke B_{Fe} .

The magnetizing inductance L_{md} can be computed according to:

$$L_{md} = \frac{6}{\pi} \mu_0 \cdot \left(\frac{k_w N_s}{p} \right)^2 \cdot \frac{RL}{k_c g} \cdot x \quad (2.15)$$

where g is the airgap length, the k_c is the Carter factor and k_w the winding factor.

With ideal iron, the Ampere's law has to be fulfilled and the d -axis Magneto-Motive Force (MMF) imposes the airgap flux density value $B_g = bB_{Fe}$:

$$\frac{3}{\pi} \frac{k_w N_s}{p} i_d = \frac{k_c g B_{Fe}}{\mu_0} \cdot b \quad (2.16)$$

The MMF determines the excitation current i_d , fixed B_{Fe} and b :

$$i_d = \frac{\pi k_c g}{3 \mu_0} \frac{p}{k_w N_s} B_{Fe} \cdot b \quad (2.17)$$

The direction with the maximum reluctance corresponds to the q axis of the machine, therefore, the design in this axis intends to curtail the flux flowing in this direction. This task is accomplished through minimization of the $\frac{L_{md}}{L_{mq}}$ ratio by design of the rotor flux barriers. The q -axis magnetizing inductance dwells of two terms:

1. The circulating inductances L_{cq} : it accounts for the stator flux paths crossing the airgap locally without crossing the air barriers.

$$L_{cq} = L_{md} \cdot \left(1 - \frac{4}{\pi} \sum_{k=1}^{n_{lay}} f_k^2 \Delta \alpha_k \right) \quad (2.18)$$

where n_{lay} is the number of flux barriers, $\Delta \alpha_k$ is the k^{th} flux barrier position and f_k is the corresponding component of q -axis MMF. According to [2.18](#), the circulating inductances depends on the flux barrier number n_{lay} and their

positions at the airgap α_k , whereas the barrier widths hc_k do not influence the the circulating inductance.

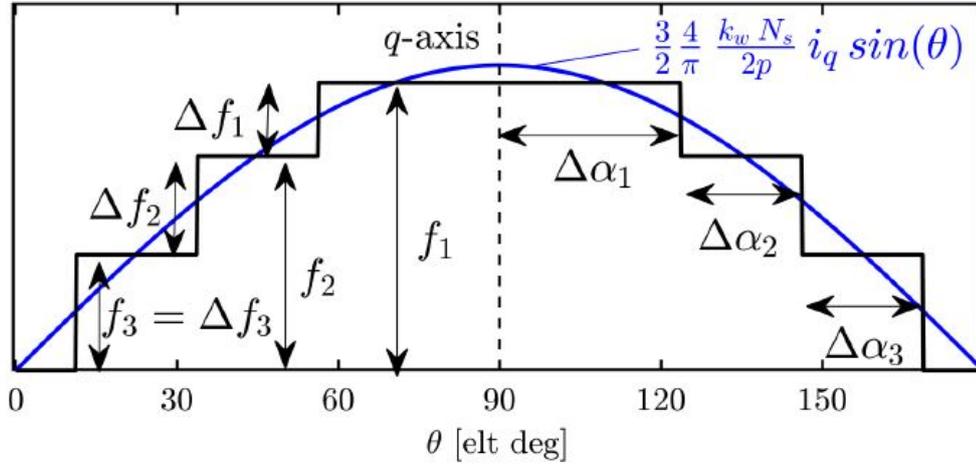


Figure 13: Sinusoidal stator MMF [9]

2. The flow-through inductances L_{fq} : it accounts for the flux paths crossing the rotor barriers from pole to pole.

$$L_{fq} = L_{md} \cdot \left(\frac{4p k_c g}{\pi R x} \sum_{k=1}^{n_{lay}} \Delta f_k^2 \frac{s_k}{hc_k} \right) \quad (2.19)$$

where the geometric parameters hc_k and s_k are displayed in Fig. [15]. The flow-through inductances depends on the sum of the flux barriers hc_{tot} , indeed, thicker barriers lead to shrink the flux flowing through the pole, i.e. lower inductances. Imposing the constant barrier permeance law $hc_k \cdot s_1 = s_k \cdot hc_1$, the [2.19] becomes:

$$L_{fq} = L_{md} \cdot \left(\frac{4p k_c g}{\pi R x} \cdot \frac{\sum_{k=1}^{n_{lay}} s_k}{\sum_{k=1}^{n_{lay}} hc_k} \cdot \sum_{k=1}^{n_{lay}} \Delta f_k \right) \quad (2.20)$$

The sum of the flux barriers lengths (i.e. total insulation) must fulfil the relationship:

$$\sum_{k=1}^{n_{lay}} hc_k = \Delta r - l_y \quad (2.21)$$

Computed the total insulation, the barriers widths are calculated according to the constant ratio principle:

$$hc_k = \sum_{k=1}^{n_{lay}} hc_k \cdot \frac{s_k}{\sum_{k=1}^{n_{lay}} s_k} \quad (2.22)$$

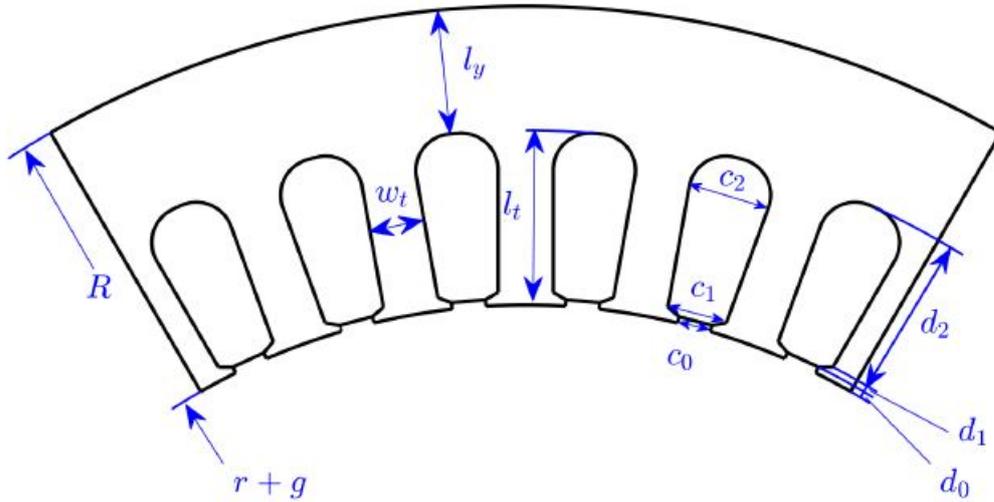


Figure 14: Stator geometric quantities nomenclature [9]

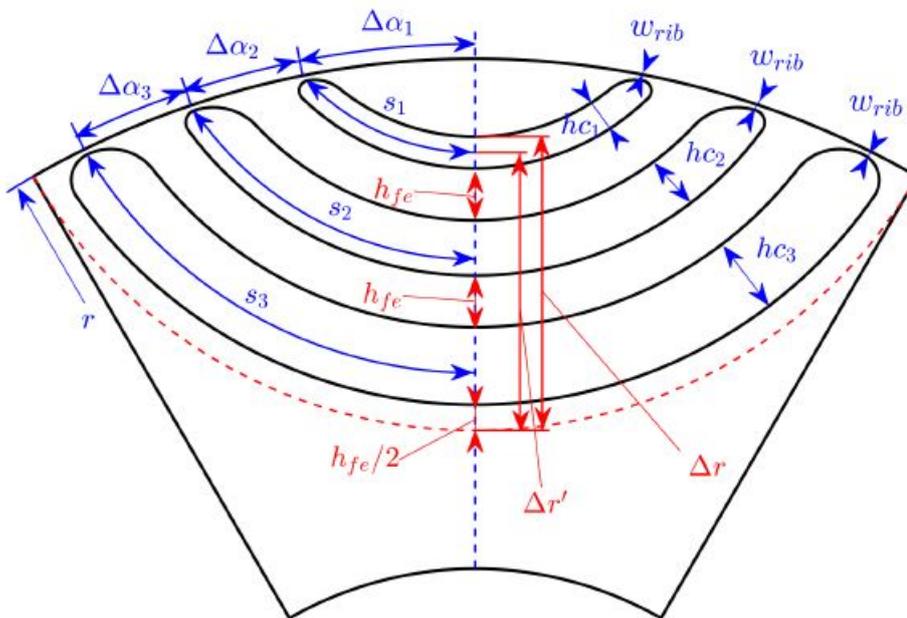


Figure 15: Rotor geometric quantities nomenclature [9]

A further contribution to the q -axis flux linkage is carried by the rotor ribs. This component is undesired and has to be minimized.

$$\lambda_{rib} = \frac{4}{\pi} k_w N_s \omega_{rib} L B_{rib} \quad (2.23)$$

in which w_{rib} is the ribs width, B_{rib} is the flux density in the ribs, corresponding to saturated iron. The impact of such contribution is decreased by curtailing the ribs as possible achieving the steel saturation. The presence of rotor ribs flux linkage is included by adding a q inductance component, $L_{rq} = \lambda_{rib}/i_q$, to L_{mq} .

The slot dimensions are linked to the slot leakage inductance L_σ , according to:

$$L_\sigma = \frac{2\mu_0 N_s^2 L}{p q} p_s \quad (2.24)$$

where p_s is the permeance factor of the stator slot. By introducing the coefficient $\beta = c_1/c_2$, the permeance factor results:

$$p_s = \frac{d_0}{c_0} + \frac{d_1}{c_0} \frac{\ln \frac{c_1}{c_0}}{\frac{c_1}{c_0} - 1} + \frac{d_2}{c_2} \frac{\beta^2 - \frac{\beta^4}{4} - \ln(\beta) - \frac{3}{4}}{(1 - \beta)(1 - \beta^2)^2} \quad (2.25)$$

2.2 Model refinements

The model accuracy may be improved by adding three correction factors to the flux linkage equations [2.2](#).

$$\begin{cases} \lambda_d = k_{cross,d} \cdot \left(\frac{L_{md}}{k_{sat}} + L_\sigma \right) \cdot i'_d \\ \lambda_q = k_{cross,q} \cdot (L_{mq} + L_\sigma) \cdot i'_q \end{cases} \quad (2.26)$$

- k_{sat} takes into account the reduction of the magnetizing inductance related to the direct saturation of the d -axis, computed as in eq. [2.29](#);
- $k_{cross,d}$ refines the cross-saturation effect and other residual errors in d -axis by means of 'FEAfix' Matlab scripts;
- $k_{cross,q}$ refines the cross-saturation effect and other residual errors in q -axis by means of 'FEAfix' Matlab scripts.

The model refinements bears on an imposed peak flux density, therefore the d -axis current component is increased by a factor corresponding to k_{sat} (eq. [2.27](#)), whereas the q -axis component is reduced as shown in eq. [2.28](#)

$$i'_d = k_{sat} \cdot i_d > i_d \quad (2.27)$$

$$i'_q = \sqrt{i_0^2 - i_d'^2} < i_q \quad (2.28)$$

Considering a flux tube given by two stator teeth, one section of the stator yoke and one section of rotor flux carrier and applying the Ampere's law, the k_{sat} expression results in:

$$k_{sat} = 1 + \mu_0 \frac{H_t l_t + H_y \left[\left(R - \frac{l_y}{2} \right) \frac{6}{2pq} + \frac{s_{nlay} + s_{nlay} - 1}{2} \right]}{k_c g B_{Fe} b} \quad (2.29)$$

Note that the saturation factor k_{sat} is inversely proportional to both x and b , then when the latter are higher the analytical model is more accurate since the k_{sat} is closer to one.

The refinement effect is underlined in Fig. [16](#) and [17](#), the first reports the torque variation and the latter the power factor variation. As expected, the motor performances, in terms of power factor and torque, are worse after the model refinements introduction, however, they are more accurate.

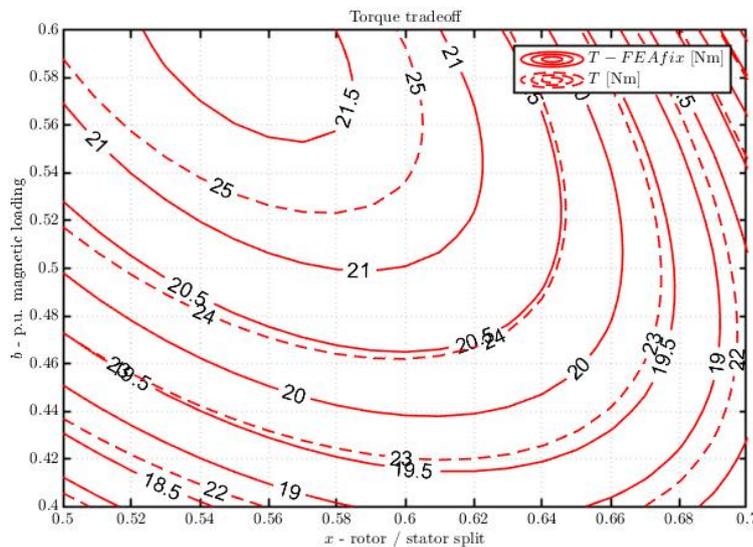


Figure 16: Torque trade-off in the x - b plane - Torque with FEAfix refinement (solid line), Torque before the FEAfix refinement (dashed line)

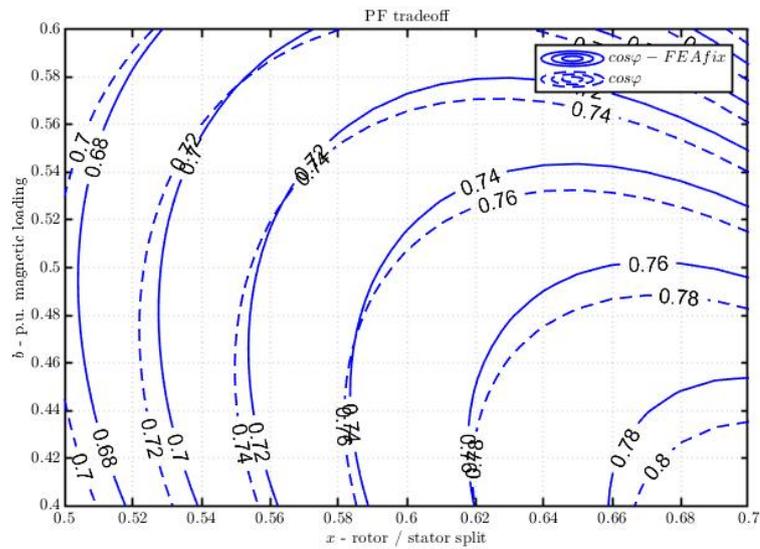


Figure 17: Power factor trade-off in the $x - b$ plane - Power factor with FEAfix refinement (solid line), power factor before the FEAfix refinement (dashed line)

2.3 SyR-e design procedure

The **SyR-e** design procedure, disclosed in fig. [20](#), begins with the definition of the active parts envelope and the feasible Joule loss per outer surface, which is strictly related to the maximum winding temperature.

Then, a trade-off between power factor and torque has to be selected by means of a script called '*syrmdesign*'. The design process in SyR-e is based on two geometric parameters, x and b , reported in the equations [2.10](#) and [2.11](#); the first controls the rotor diameter, the latter the core dimensions. Thus, torque and power factor are expressed as a function of x and b and plotted as shown in Fig. [18](#). The torque contours are displayed in red and the power factor contours in red. From this plot a machine can be picked fixing its x and b values given the torque and power factor selected. Note that the a x reduction increases the slot area, i.e. higher slot current and output torque. Whereas, a b reduction leads to thinner iron path with greater stator slots and rotor air barriers, which means a decrease in the power factor value.

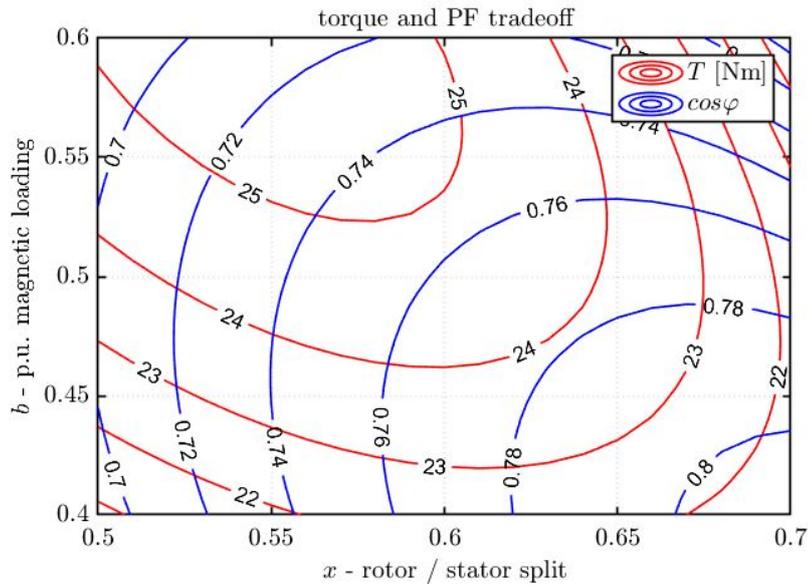


Figure 18: Torque and power factor trade-off before FEAfix in the x - b plane

This first step is executed by running the *syrmdesign* Matlab script and it consists of the design equations and refinements described in the subsection 2.2. Including such refinements, the Torque - PF before and after FEAfix refinement are shown in Fig. 18 and Fig. 19 respectively.

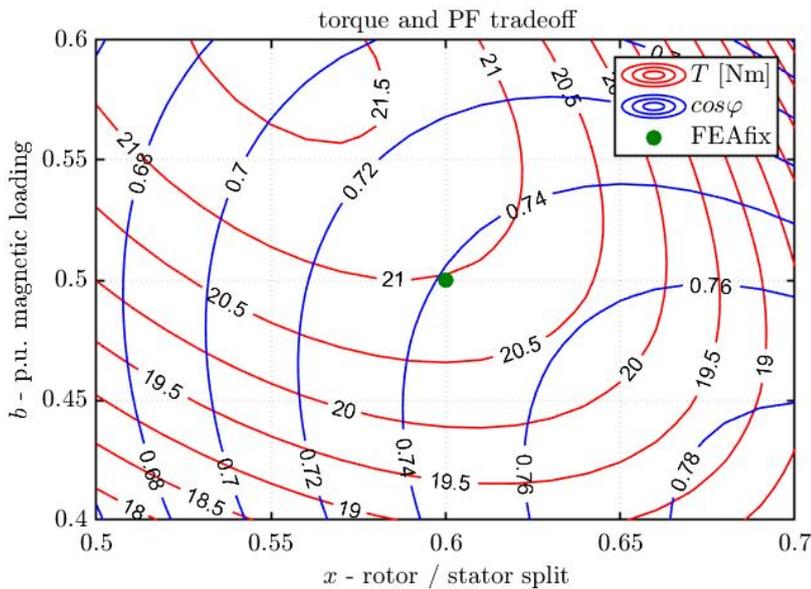


Figure 19: Torque and power factor trade-off after FEAfix in the x - b plane

At this point, if the design constraints are met, the SyR machine design procedure is finished. As mentioned, SyR-e allows PMS machine design and it starts at the end of a SyR machine design by adding permanent magnets. Anyway, once the design

is seized, several evaluations may be executed and the data collected by means of Matlab scripts, as displayed in Fig. 20.

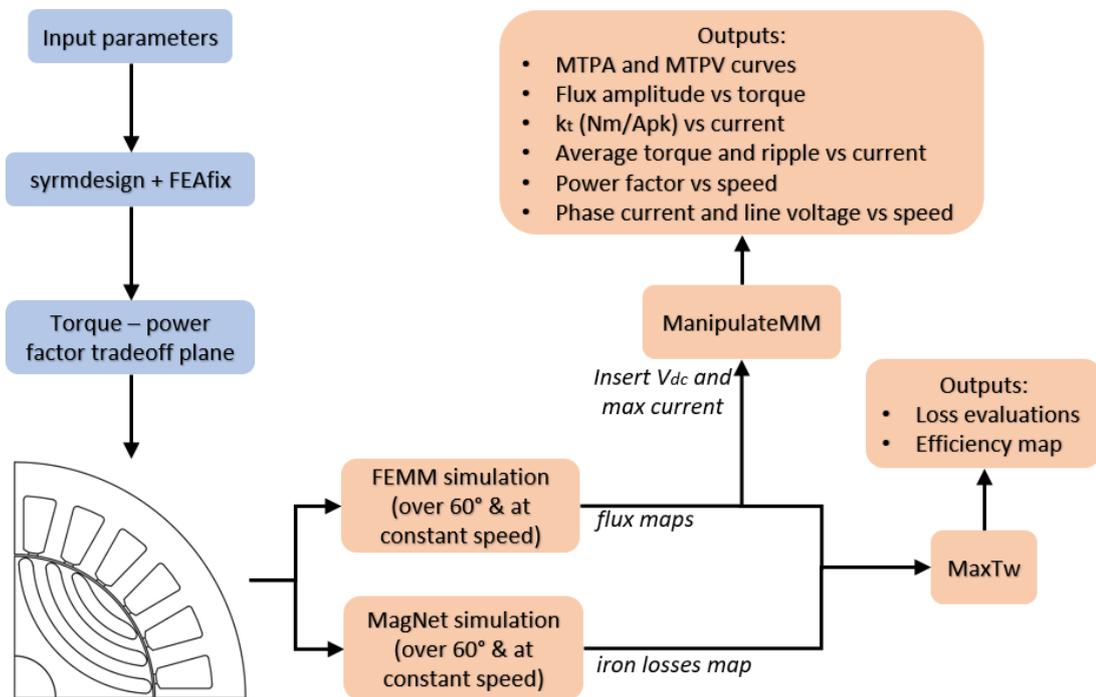


Figure 20: SyR machine design procedure in SyR-e: FEA characterization of the final design

3 Export SyR machines from SyR-e to Motor-CAD

To optimize the synchronous reluctance machines design procedures and their performance analysis, SyR-e and Motor-CAD have been coupled.

The first step was to export a machine model from SyR-e to Motor-CAD, in order to allow analysis and evaluations of the same machine in both software. This task was composed by two main sections: model export and simulation parameters export. The critical part was to match up the parameters of the two software, since they are based on different quantities and conventions. Last, the scripts created are reported at the bottom of the thesis as appendix.

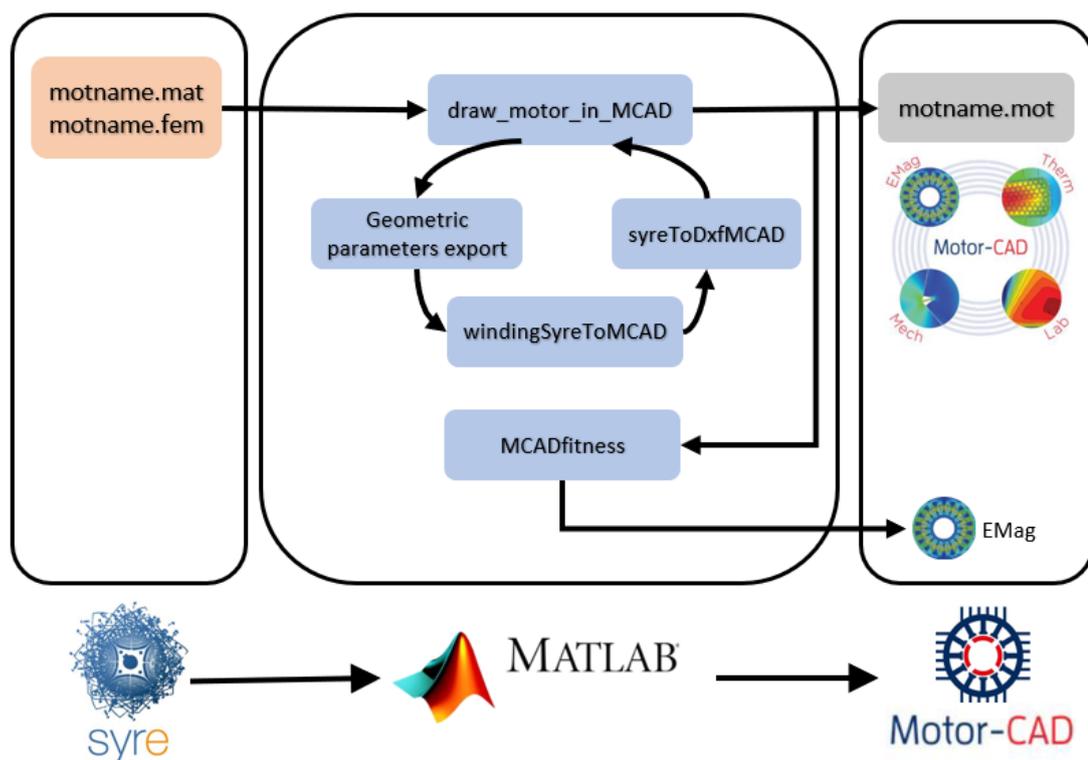


Figure 21: Export process diagram

3.1 Export of a motor model to Motor-CAD

The function *draw_motor_in_MCAD* has as inputs the SyR-e model filename and the pathname, and as result the corresponding model in Motor-CAD. It consists of four parts:

1. geometric variables (lines 7-178 in *draw_motor_in_MCAD*): stator and rotor Motor-CAD parameters are computed by the SyR-e parameters, further even the material names are exported. Note that the names of the material used must be the same in SyR-e and in Motor-CAD. Moreover, it is suggested to check if the material properties (B-H curve, electric and mechanical parameters) match in both software.
2. winding export (line 180 in *draw_motor_in_MCAD* - corresponding to the *windingSyreToMCAD* function): it is possible to export upper/lower and side by side winding.
3. .dxf creation (line 181 in *draw_motor_in_MCAD* - corresponding to the *syreToDxfMCAD* function): a .dxf sketch for the rotor is created and passed to Motor-CAD, whereas the stator considered in the **FEA** is not passed by .dxf but taken directly from the Motor-CAD model constructed.
4. save model in Motor-CAD (line 190 in *draw_motor_in_MCAD*) and print in Matlab the pathname of the created Motor-CAD file.

The model export is valid for whatever circular synchronous reluctance motor, i.e. different pole pairs, number of slots and dimensions.

3.2 Simulation of the exported design in Motor-CAD

The simulation export is executed by means of the Matlab function '*MCADfitness*', once the following requested inputs are provided:

- *.mat* SyR-e motor (*geo*, *per*, *mat* structures);
- *eval_type* indicates the evaluation type which is intended to execute;
- *filenameIn* is the entire filename, including the pathname.

3.2.1 dq axes conventions

In order to be able to export simulation parameters from one software to the other and to fully understand the results obtained, it was fundamental to recognize the respective dq axes conventions. In turns, SyR-e adopts the most common axes conventions for SyR motors: d -axis corresponds to the highest permanence direction and the q -axis to the lowest one. However, Motor-CAD uses a different axes convention, as underlined in fig. [22](#).

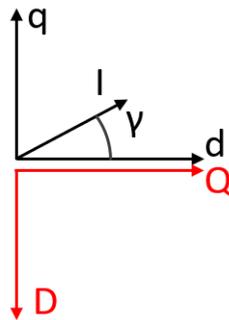


Figure 22: Axes conventions - in red the Motor-CAD axes, in black the SyR-e axes

This was fundamental to be pointed out because it affects the way in which the results have to be read. Indeed, a general quantity x fulfills the following relationship between its dq components in the two software.

$$x_{d,s} = x_{Q,m} \quad (3.1)$$

$$x_{q,s} = -x_{D,m} \quad (3.2)$$

in which $x_{d,s}$ and $x_{q,s}$ are the dq SyR-e components, whereas $x_{D,s}$ and $x_{Q,s}$ are the dq Motor-CAD components. Last, the current phase definition corresponds in both software, as can be seen in fig. [22](#) since SyR-e starts to count the current angle in anti-clockwise direction from the d -axis, while Motor-CAD does it in the same direction but starting from its q -axis.

3.3 New Motor-CAD section in SyR-e interface

One of the main deliverables of this work is a new section in SyR-e entirely dedicated to Motor-CAD. When this thesis was concluded, it was released a new SyR-e version with a Motor-CAD section organized as shown in fig. 23. The available options are listed below, however it is likely that in the future new features will be added.

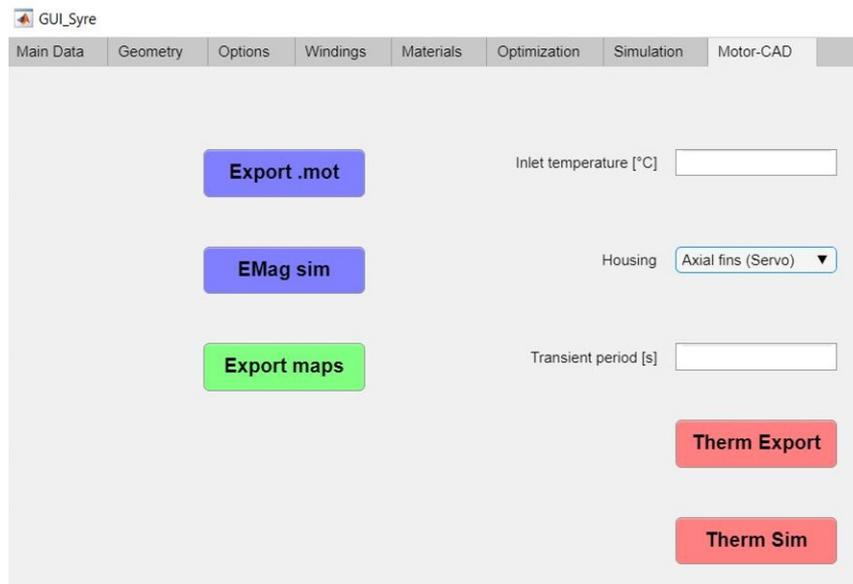


Figure 23: Motor-CAD tab in SyR-e interface

- 'Export .mot' button: it allows to create a .mot Motor-CAD model as a copy of a circular flux barriers synchronous reluctance machine model realized in SyR-e. The stator is perfectly duplicated, while the rotor is modeled by a .dxf sketch.
- 'EMag sim' button: it takes as reference the simulation parameters in SyR-e to run a single point simulation in the Motor-CAD EMagnetic module. The results are also saved and displayed in Matlab.
- 'Export maps' button: it exports flux maps to the Motor-CAD Lab, allowing to perform analysis using custom flux maps without computing them in Motor-CAD.
- 'Therm Export' button: it exports thermal parameters to the Thermal module in Motor-CAD.
- 'Therm Sim' button: it runs a thermal simulation in Motor-CAD and collects the results in Matlab.

3.4 Export test

3.4.1 Sample motor specification

To evaluate the accuracy of the built export procedure, a sample SyR-e model, reported in Fig. 24, was used. Its main specification are displayed in tab.1.

Rated power [kW]	2.2
Rated torque [Nm]	21
Rated ω [rpm]	1000
Rated current [A]	13.2
Rated voltage [V]	157
$\cos(\phi)$	0.7
Peak current [A]	30
Peak torque at low speed [Nm]	44
Pole pairs	3
Slots per poles phases	2
Rotor barriers	3
Airgap thickness [mm]	0.325
Stack length [mm]	110
Shaft radius [mm]	25
Rotor radius [mm]	59.5
Stator outer radius [mm]	87.5
Turns in series per phase	120
Slot filling factor	0.45
Thermal load [W/m ²]	2800
Rated current [A]	15.8
Rated torque [Nm]	21
Torque at 1.8 pu current [Nm]	42
Iron material	M600-50A

Table 1: RAWP specification

3.4.2 'Export .mot' button

By hitting the first button labeled with 'Export .mot', two outputs are provided: an unvaried SyR-e model and a Motor-CAD model constructed by the scripts (respectively shown in fig. 24 and fig. 25). While, Fig. 26 underlines the scripts capability to export also the SyR-e winding setup.

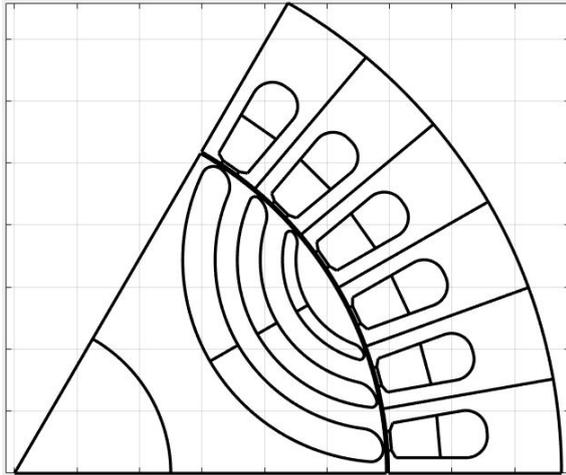


Figure 24: SyR-e exported model

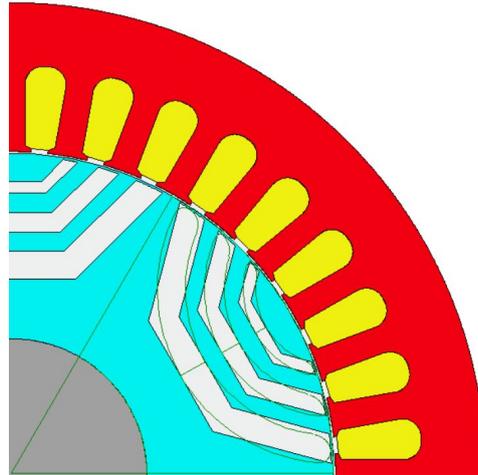


Figure 25: Motor-CAD imported model

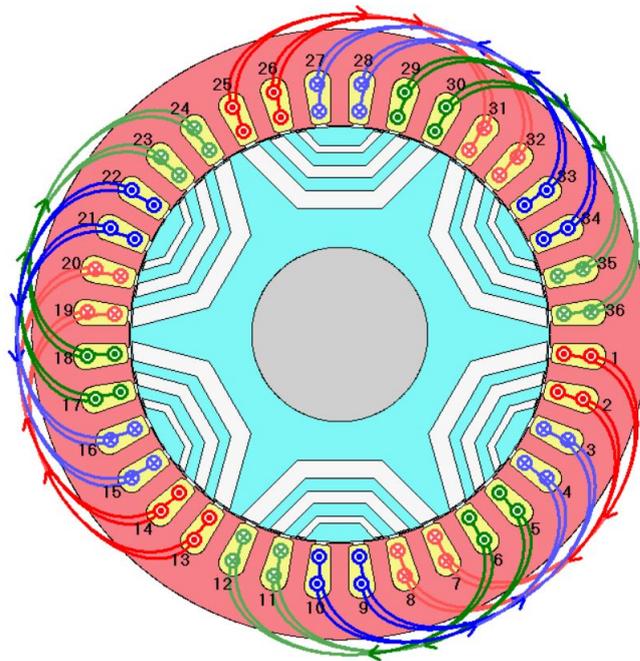


Figure 26: Sample motor exported in Motor-CAD - Winding

Concerning the FEA regions, they are automatically recognised correctly by Motor-CAD, once the .dxf is sufficiently similar to the model. Moreover, it is interesting to point out that an undefined region is labeled as air from the FEA Motor-CAD solver.



Figure 27: Sample motor imported in Motor-CAD - FEA regions

3.4.3 'EMag Sim' button

A single operating point was simulated in Motor-CAD by means of the 'EMag Sim' button and the results collected back to Matlab. A simulation with identical parameters was run in SyR-e with the FEMM support, in order to assess the accuracy of the created process. Then, the results are compared in Tab. [3.4.3](#), underlining a satisfying low percentage error for every furnished output.

In the reported example an operating point along the MTPA was picked, however similar tests were performed for different operating points and errors always resulted below 5%.

The simulations in both software were run with the same FEA and mesh parameters:

- mesh with 1440 airgap internal and surface points;
- 15 points simulated each 60 electric degree.

Variable name	SyR-e value	Motor-CAD value	Percentage error
i_d [A]	9.5427	9.5427	< 0.01%
i_q [A]	12.1528	12.1528	< 0.01%
λ_d [Vs]	0.4265	0.4193	1.7%
λ_q [Vs]	0.0714	0.0720	0.8%
T [Nm]	20.2	19.9	1.4%
dT_{pu}	0.0718	0.0690	3.9%

Table 2: Export test results in SyR-e and Motor-CAD

The simulation time in FEMM resulted to be 71 seconds, while in Motor-CAD 88 seconds. However, with the suggested simulation parameters in Motor-CAD (360 airgap points and 18 simulated points per 60), it takes 23 seconds.

Fig. 28 and 29 show the dq linkage flux components obtained in SyR-e (blue) and in Motor-CAD (orange), while the fig. 30 compares the output torque.

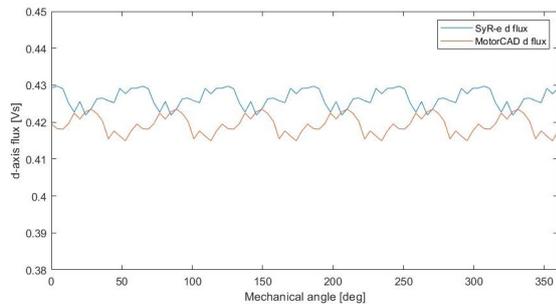
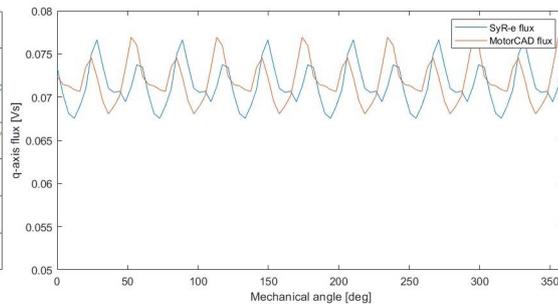
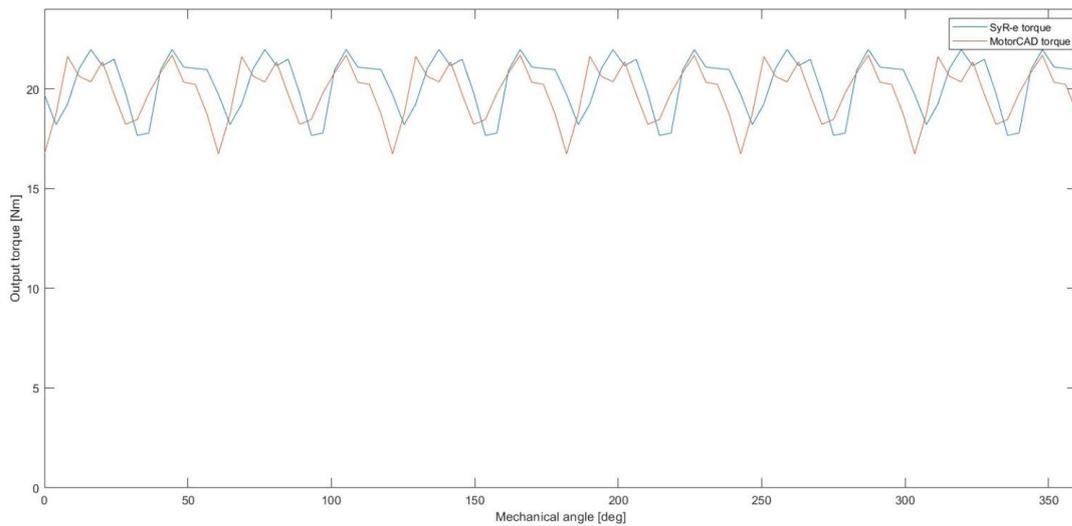
Figure 28: d -axis flux comparisonFigure 29: q -axis flux comparison

Figure 30: Torque comparison

4 ReFreeDrive project review

4.1 ReFreeDrive project

4.1.1 Project overview

The ReFreeDrive project started on October 2017 thanks to the European Union funding and to the effort of 13 partner companies in 6 European countries. The project contributed to develop rare earth free traction technologies beyond their current state of art, with a strong focus on industrial feasibility for mass production while focusing on lower cost with higher specific torque and power.



Figure 31: ReFreeDrive project logo [14]

The main project deliverables are two brushless AC machines: an induction machine with copper die-cast rotor and a synchronous reluctance machine. These machines have common features, as the stator design, that justify their development in just one project. The final products are aimed to be integrated in two powertrains: a medium power range with 75 kW and an high power range of 200 kW . The electric powertrains to be developed covered a broad power range, so as responding to the demand from the current and future market. The list below reports the companies within the ReFreeDrive Consortium.

- European Copper Institute
- Motor Design Ltd
- Jaguar Land Rover
- Aurubis
- IFP Energies Nouvelles
- Università degli studi dell'Aquila

- Centro Sviluppo Materiali
- Tecnomatic
- Mavel
- R13 Technology
- Privè
- MetallGiesserei
- Breuckman
- Fundación Cidaut

The diagram in fig. 32 discloses the project process, underlining which company was involved in each step. For further information regarding the ReFreeDrive project, the reader can refer to [14], [15].

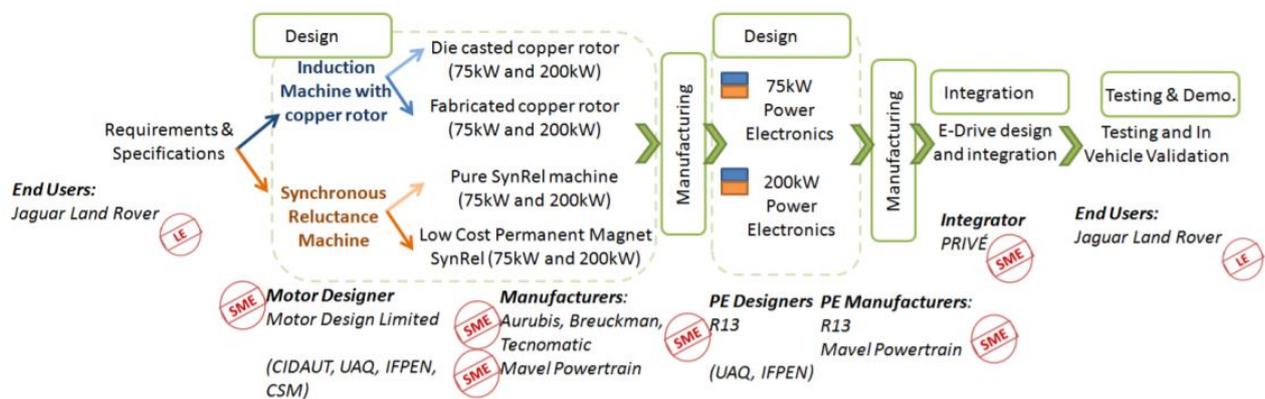


Figure 32: ReFreeDrive project process [14]

In this thesis work, to investigate the SyR machine design procedure and to test the export process constructed, the ReFreeDrive Synchronous Reluctance machine was re-designed.

4.1.2 Requirements

According to [16] [17], the SyR ReFreeDrive machine requirements are reported in tab. 3 and they were taken as reference in the design process.

Requirements	
Stator diameter	< 230 mm
Active length	< 200 mm
Airgap length	0.7 mm
Weight	< 48 kg
Maximum speed	18000 rpm
Maximum peak current	700 A
Maximum torque up to 5 krpm	> 380 Nm
Peak power	> 200 kW
Peak power at maximum speed	> 50 kW
Material adopted	M235-35A
DC voltage	800 Vdc
Ripple at maximum torque	< 10 %
Ripple at maximum speed	< 25 %
Maximum efficiency	> 95 %
Power density	> 4 kW/kg
Torque density	> 7.9 Nm/kg

Table 3: Requirements

4.1.3 Original performance

The design and optimization procedure executed in the ReFreeDrive project led to a rotor layout with multiple ribs and fluid flux barriers. This geometry, displayed in fig. 33, is clearly unusual compared to those typically reported in literature, however it represented the best design solution achieved by means of a topology optimization made in Ansys Maxwell.

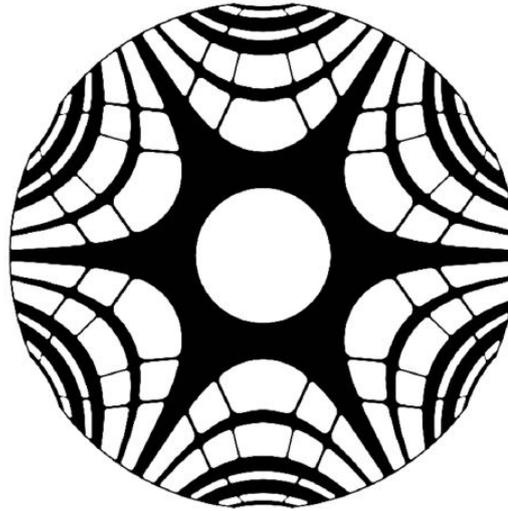


Figure 33: Optimized rotor shape [16]

According to [16], the torque and power graphs are reported below, arranged in 3 lines: red line for the design with no radial ribs, blue line for the optimized ribs and the black one represents the starting design with the usual radial ribs.

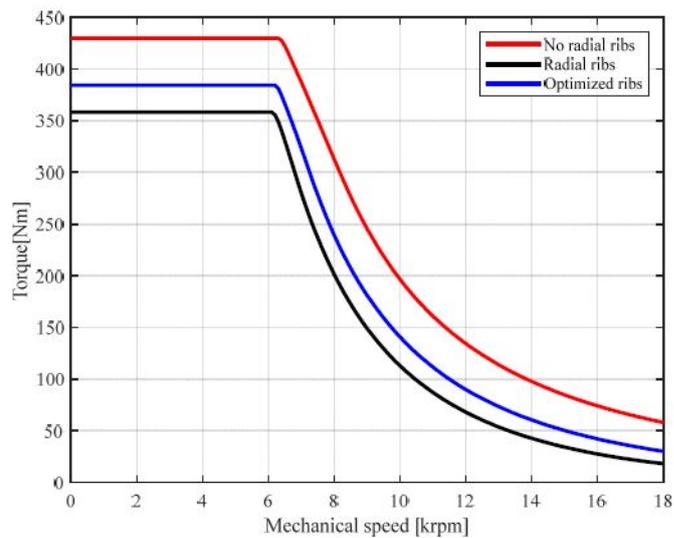


Figure 34: Electromagnetic torque over the speed range [16]

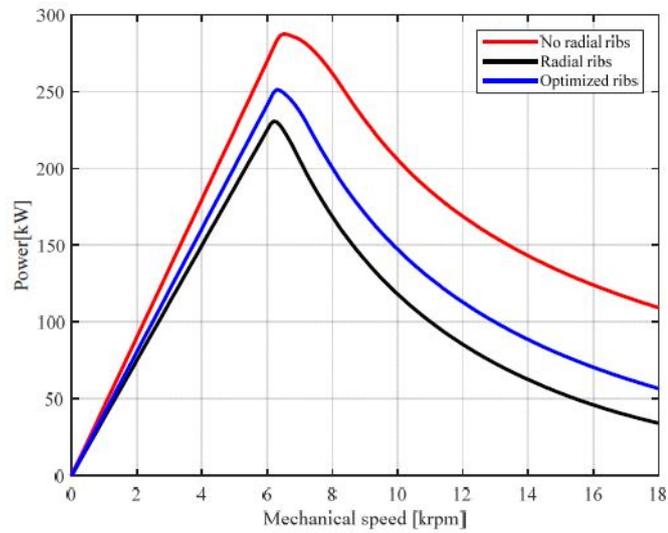


Figure 35: Output power over the speed range [16]

Since the SyR machines bear entirely on the rotor anisotropy, here are reported the inductances and saliency ratio curves over the speed range, according to [16].

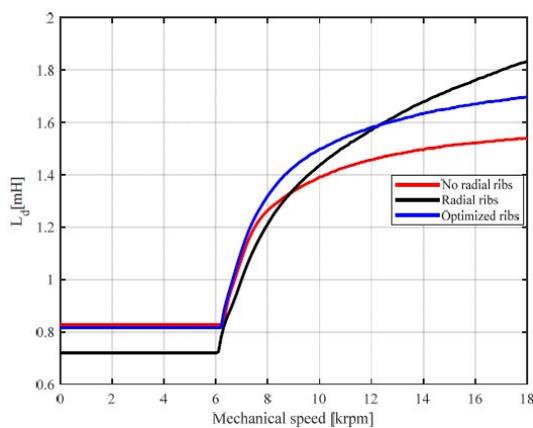


Figure 36: q-axis inductance curves [16]

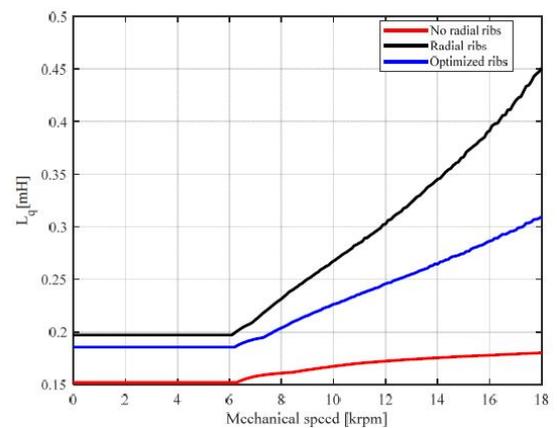


Figure 37: q-axis inductance curves [16]

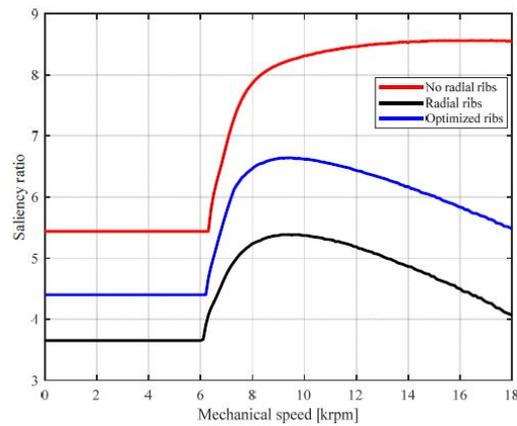


Figure 38: Saliency ratio over the speed range [16]

One of the most demanding design part for the SyR machines is the mechanical validation, due to the importance of the ribs thickness both in electromagnetic and mechanical fields. Hence, [16] reports the mechanical performances displayed below.

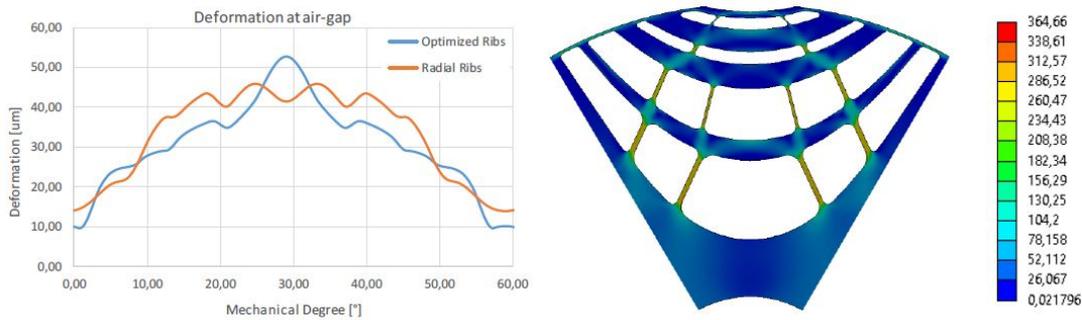


Figure 39: Deformation at the airgap [16] Figure 40: Mechanical stress [MPa] [16]

4.2 Proposed design methodology

Once the new bridge between the two environments has been created, it was used to formulate a design process made upon the points of strength of both parties. The optimal combination of the two tools was deemed to be the one displayed in fig. 41.

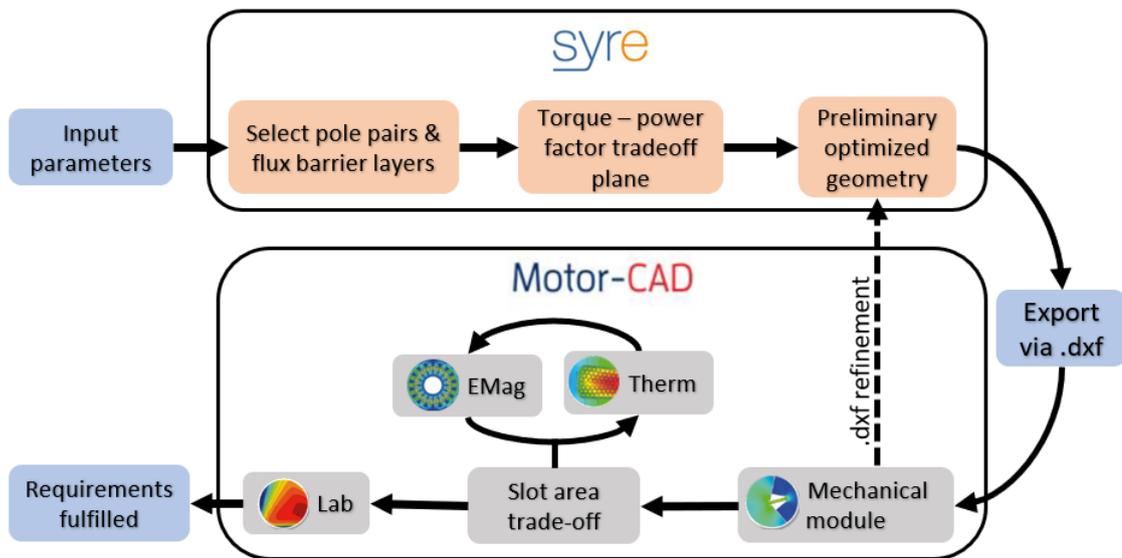


Figure 41: Joint procedure

The first part of the design process is executed in SyR-e. Here, once the input parameters are selected, different solutions can be quickly investigated by varying fundamental parameters, like number of pole pairs, flux barrier layers and slots number. Then, the x and b values are picked by the torque and power factor trade-off plane, in order to get a preliminary optimized geometry. Such procedure executed in SyR-e takes few minutes and it is capable to provide an excellent starting point to continue the design and the analysis in Motor-CAD.

The SyR-e model is easily exported to Motor-CAD by means of the Matlab scripts described in Chapter 3. A *.mot* file is so created, with a stator perfectly duplicated by the SyR-e model and a rotor modeled by a *.dxf* sketch, since, for the time being, Motor-CAD does not include the circular flux barriers geometry. As said, SyR-e lacks of an accurate mechanical validation for high speed motors, thus, in Motor-CAD, one of the first step to be executed is to run the mechanical FEA. This allows to evaluate the mechanical stress and the maximum displacement, in order to be able to properly adjust the geometry, in particular the ribs thickness. However, since the rotor is reported as a *.dxf* file, it is not possible to quickly change its parameters in Motor-CAD, but the *.dxf* sketch has to be modified. This, reported in fig. 42, is the most onerous activity and it is related to the lack of the circular geometry in

Motor-CAD, indeed, without this loop, the joint procedure would be linear, without any feedback and significantly quicker. However, such complexity is expected to be solved within the next versions of Motor-CAD, since **MDL** should include the circular flux barrier geometry.

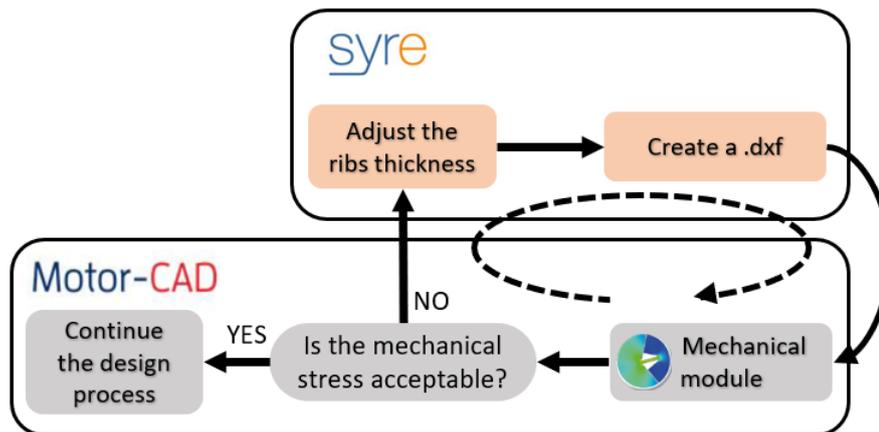


Figure 42: Loop for mechanical sizing

Hence, once the mechanical concerns are satisfied, it is suggested to check the EMagnetic and Thermal modules. In the first one, the winding can be deeply specified by selecting conductor section, copper insulation, impregnation area and number of strands. Further, the winding goodness can be evaluated by checking the MMF and the winding factor harmonics. Whereas, in the thermal module, it is possible to accurately size and define the thermal system. At this stage, the Lab model can be built and the motor simulated in order to collect comprehensive results regarding its performance.

4.3 Preliminary design

4.3.1 Fixed parameters

To get a preliminary design, some parameters have to be fixed as reported in Tab. 4. Initially, the dimensions are imposed at the maximum values within the constraint ranges, afterwards, if possible, they may be reduced.

Parameters	Values
Stator diameter	230 mm
Active length	200 mm
Airgap length	0.7 mm
Material adopted	M235-35A

Table 4: Inputs parameters

As first step, the number of flux barrier layers, number of slots and pole pairs have to be chosen. All of them are strictly linked and different combinations have to be investigated according to the theory in paragraph 2.1.1.

4.3.2 Number of pole pairs

Firstly, it is interesting to point out that a low number of pole pairs can reduce the q-axis inductance (higher saliency) but it also increases the torque ripple. Moreover, low number of poles machines have a larger stator yoke, reducing the torque density. On the other hand, machines with high number of poles can hardly have an high number of flux barriers, this leads to reduce saliency ratio. For these reasons, the numbers of pole pairs pp considered suitable for the given application are:

- $pp = 3$;
- $pp = 4$.

However, with 8 poles the i_d current component grows in p.u. due to a larger p.u. airgap, therefore, fixed to 0.7 mm the airgap, $pp = 4$ is not a competitive option. Thus, the design procedures continued with a new constrained parameter: $pp = 3$ (3 pole pairs).

4.3.3 Number of flux barrier layers and number of slots

Subsequently, the number of slots and number of flux barriers combinations were investigated, as shown in tab. 5. Note that q indicates the number of slots per poles per phases and that 4 flux barriers layers were selected, leading to have $n_r = 18$ (9 separation points over one rotor pole pitch, thus, 18 over one pole pair).

$$n_r \leq n_s - 4 \quad \vee \quad n_r \geq n_s + 4 \quad (4.1)$$

	n_r	pole pairs pp	q	n_s	eq. 4.1 assessment
Case 1	18	3	2	14	YES
Case 2			3	18	NO
Case 3			4	24	YES

Table 5: Number of slots and number of flux barrier layers combinations assessment

The outcomes of such analysis were two design options:

1. 3 pole pairs, 4 flux barrier layers, 2 slots per phases per poles (36 slots) - Fig. 43;
2. 3 pole pairs, 4 flux barrier layers, 4 slots per phases per poles (72 slots) - Fig. 44.

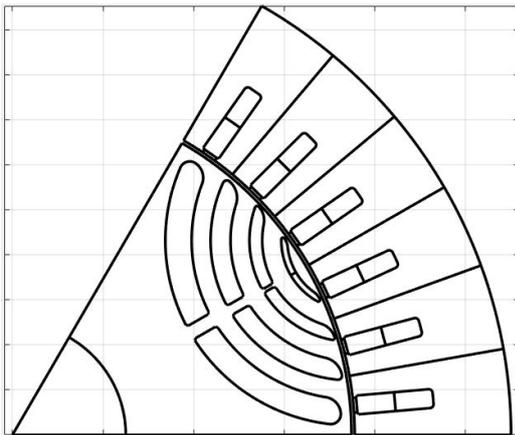


Figure 43: Case 1 - Motor sketch

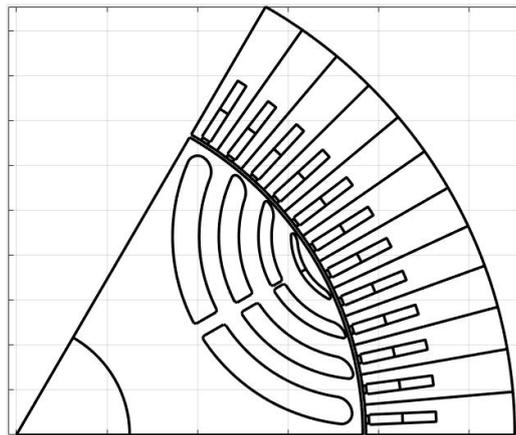


Figure 44: Case 2 - Motor sketch

4.3.4 Hairpin stator winding

The investigated designs take advantage of the great performance of the hairpin winding. Each section is formed of rectangular profile conductors, which suit well for

distributed windings. Compared to the traditional round wire winding, the hairpin one has higher slot fill factor, up to 0.75, and shorter end connections. Further, also the thermal dissipation through the slot is improved, ensuring a thermal path within the slots. On the other hand, the drawback is the constraint number of conductors in one slot, which has to be an even number minor than 8.

In the following evaluation, the hairpin winding was adopted considering a slot filling factor of 0.65 (*geo.Kcu* in SyR-e).

4.3.5 Possible design solutions

Once the main parameters were chosen, several design solutions were evaluated by changing the windings type and the turns in series per phase (N_s), as shown in fig. 45. Last, it was also considered an increase in the peak current from $700 A_{pk}$ to $800 A_{pk}$, analysing its advantages and disadvantages.

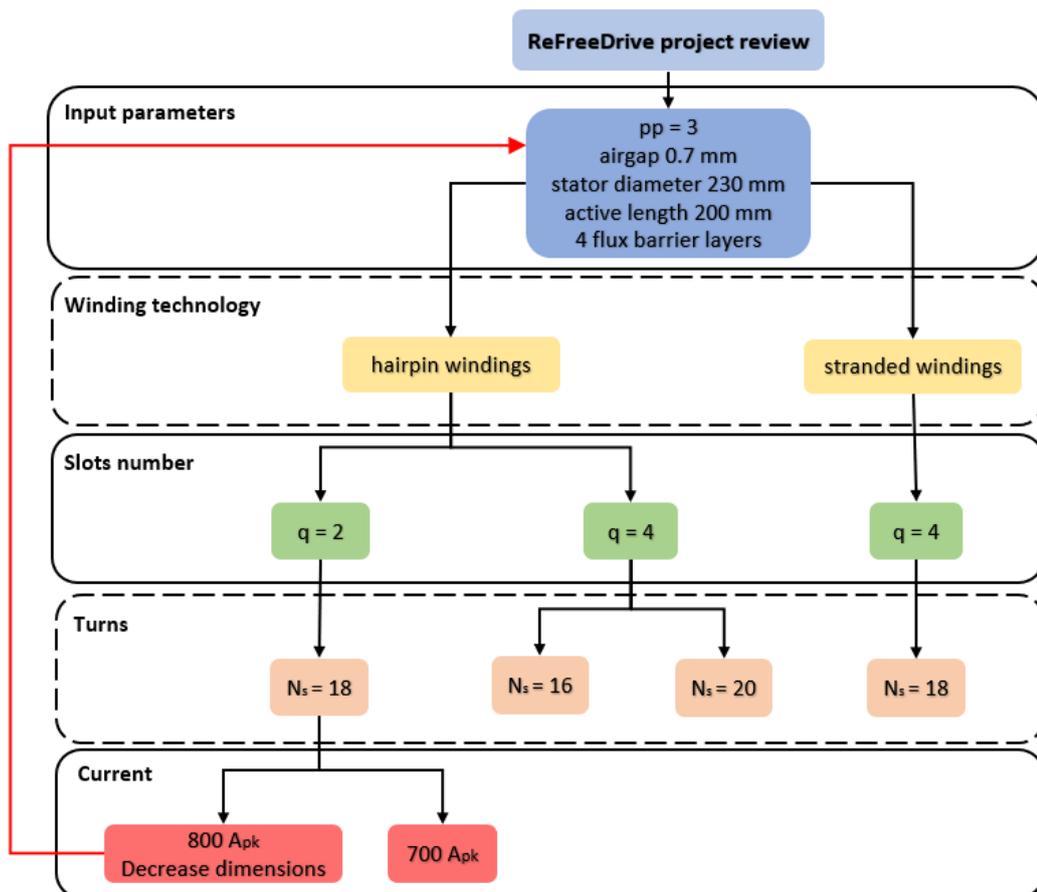


Figure 45: Design options

4.4 Design in SyR-e

The SyR-e main data window was populated by inserting the inputs in tab. 6.

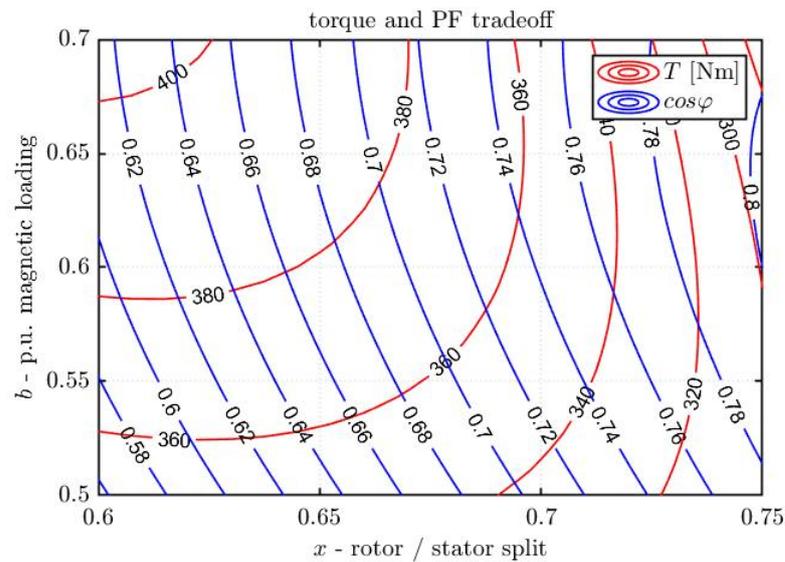
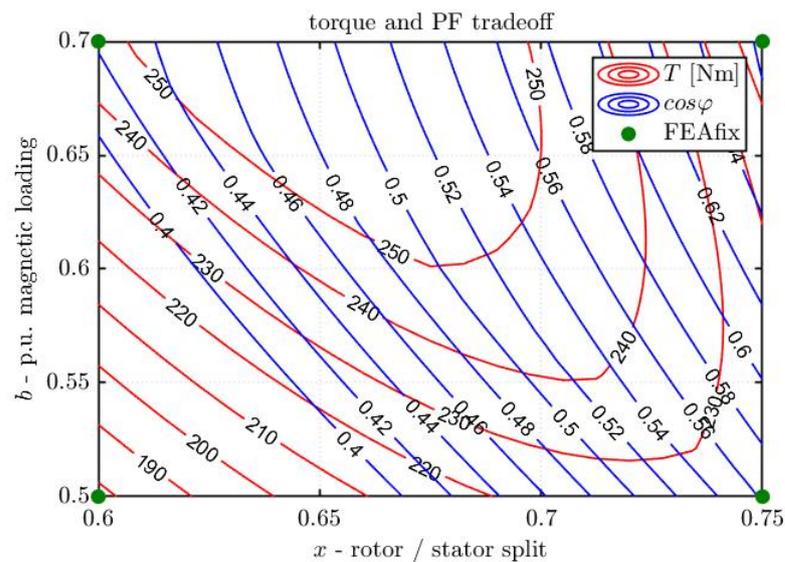
Parameters	Values
Number of pole pair (geo.p)	3
Number of slots per poles phases (geo.q)	2 or 4
Airgap thickness (geo.g)	0.7 mm
Stator outer radius	115 mm
Stack length (geo.l)	200 mm
Type of rotor (geo.RotType)	'Circular'

Table 6: Main data window - inputs parameters

Then, the number of slots per poles phases and the turns in series per phases were varied in order to detect the best design option. The hairpin winding was adopted, except for the design option 4, which employs a stranded winding to make the 18 turns in series per phases feasible even with 72 slots. All the analysis were performed considering a peak current of $700 A_{pk}$, just at the end it was evaluated a design solution with $800 A_{pk}$.

The stator/rotor split ratio x and the magnetic loading b were selected by running '*syrmdesign*' with FEAfix4. Indeed, when using high loading factor (liquid cooled system) and radial bridges (high speed machines), '*syrmdesign*' becomes extremely imprecise, since torque and power factor are greatly overestimated. Thus, it is suggested to directly rely on FEAfix to reach a good level of accuracy. The above appears clear by comparing the result with and without the use of FEAfix (fig. 46 and 47).

With the chosen inputs, the torque and power factor trade-off plane is represented in Fig. 47. Note that all the design options bear on the same x and b values, since it was deemed a good trade-off point. In fact, a high value of x is mandatory once the power factor tends to be low everywhere in the plane and even a high b appears to be beneficial.

Figure 46: Low accuracy plane provided by '*syrmdesign*'Figure 47: High accuracy plane provided by '*FEAfix*'

A satisfying trade-off is at $x = 0.68$ and $b = 0.7$. As aforementioned, the following designs have in common such x and b values.

To have a comprehensive overview of the obtained performance, other inputs are needed:

- in 'Stator Rotor Geometry' window the number of rotor barriers was selected ($geo.nlay = 4$), as well as the slot type, which was set to parallel to better suit the hairpin winding.

- a roughly thermal sizing was executed matching the target copper temperature with the estimated copper temperature by varying the thermal loading k_j .
- a roughly mechanical sizing was performed inserting the requested over-speed in the SyR-e GUI (it is suggested to include a 20% margin).
- winding definition, e.g. slot filling factor and turns in series per phase.
- materials definition for conductors and iron.

Parameters	Values
Thermal loading (per.tempcuest)	between 32000 - 42000 [W/m ²]
Target copper temperature (per.tempcu)	180 °C
Housing temperature (per.temphous)	70 °C
Overspeed (geo.nmax)	21600 rpm

Table 7: Other options window parameters

Parameters	Values
Slot filling factor (geo.Kcu)	0.65
Turns in series per phase (geo.Ns)	16 - 18 - 20
Number of 3-phase sets (geo.n3phase)	1

Table 8: Windings window parameters

Parameters	Values
Slot material (geo.BLKLABELmaterials)	'Copper pure'
Stator material (geo.BLKLABELmaterials)	'M235-35A'
Rotor material (geo.BLKLABELmaterials)	'M235-35A'
Flux barrier material (geo.BLKLABELmaterials)	'Air'
Shaft material (geo.BLKLABELmaterials)	'M235-35A'

Table 9: Materials window parameters

4.4.1 Ribs thickness preliminary sizing

Concerning the SyR machines, the ribs sizing is a critical facet, since it is related to several quantities both in magnetic and mechanical fields. Indeed, the ribs overly influence the output torque, in terms of average and of ripple, as well as the maximum mechanical stress.

Hence, even though an accurate trade-off has to be met, at the first preliminary design the ribs were roughly sized by means of the overspeed input in the 'Other options' SyR-e window. Such input allows to assess the needed radial ribs thickness,

whereas the tangential ribs were manually overestimated and then accurately sized in Motor-CAD. The preliminary tangential ribs design is reported in tab. [10](#)

Tangential ribs thickness	Values
1 st barrier	2.7 mm
2 nd barrier	1.2 mm
3 th barrier	1 mm
4 th barrier	1 mm

Table 10: Preliminary tangential ribs sizing

To fairly compare the selected design options, the same rotor was employed, while the stator changes in terms of slots number and winding settings. In evaluating the outcomes, it has to be taken into account that the ribs thickness is overestimated and even that further adjustments may allow to fit better the constraints (e.g. trade-off between continuous and peak performances related to the slot area). In other words, it is foreseen to be able to improve the first results obtained in this section.

4.4.2 Design option 1: $q = 4$, $N_s = 20$

Selected a stator with 72 slots, a 20 turns in series per phases hairpin winding is feasible with 6 parallel paths and 10 conductor per slot in a double layer setting. Such design led to the results reported below.

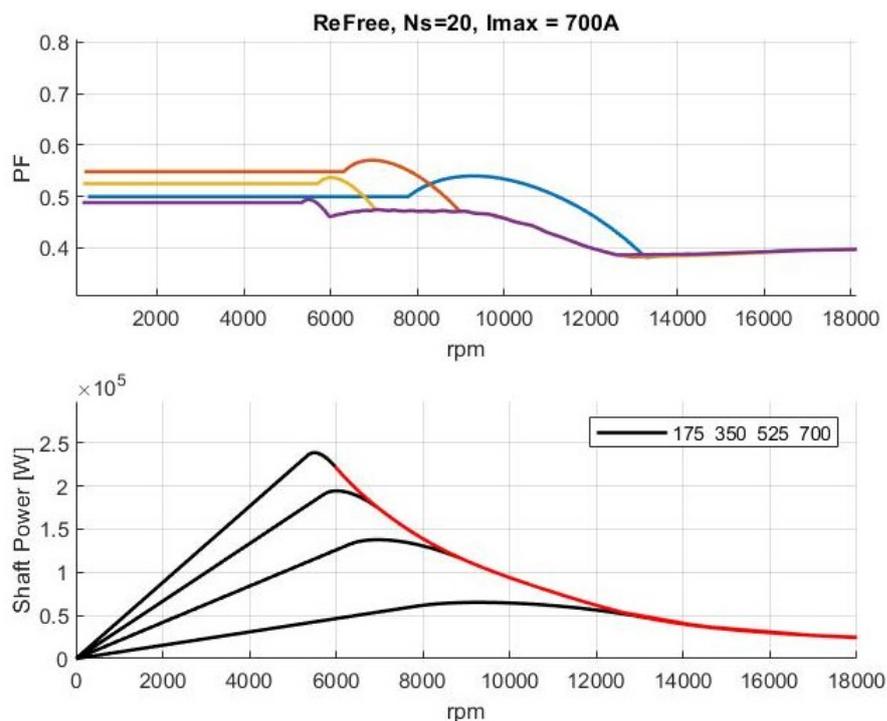


Figure 48: Option 1 - Power factor and shaft power curves at different peak current

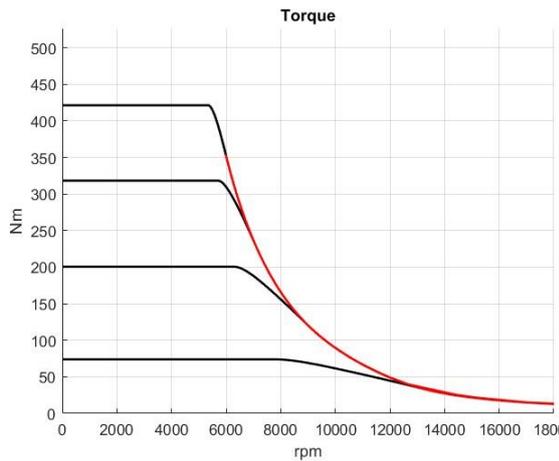


Figure 49: Option 1 - Torque

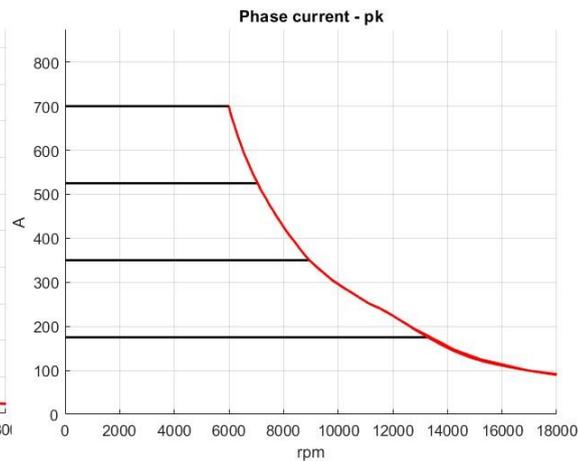


Figure 50: Option 1 - Peak phase current

Parameters	Values
Peak torque at low speed	420 Nm
Peak power at maximum speed	26 kW
Maximum peak power	240 kW
Power factor at low speed	0.56

Table 11: Option 1 - Performances

It has to be considered that the minimum ribs thickness are overestimated and that further optimizations may be applied, therefore a decent increase in the torque-power performances has to be taken in to account. In this case, the maximum torque is already largely satisfying ($420 Nm$), however the power at maximum speed is really depressed ($26 kW$), thus it is unlikely that any adjustments could lead to meet the constraints. For these reasons, the present design was deemed unsuitable.

Instead, it is delightful to reduce the number of turns, since it decreases low speed torque and retards the flux weakening causing a rise on the power at maximum speed.

4.4.3 Design option 2: $q = 4$, $N_s = 16$

A 16 turns in series per phase windings is feasible with 3 ways and 4 conductor per slots organized in a double layer setup. The resulting performance are shown below.

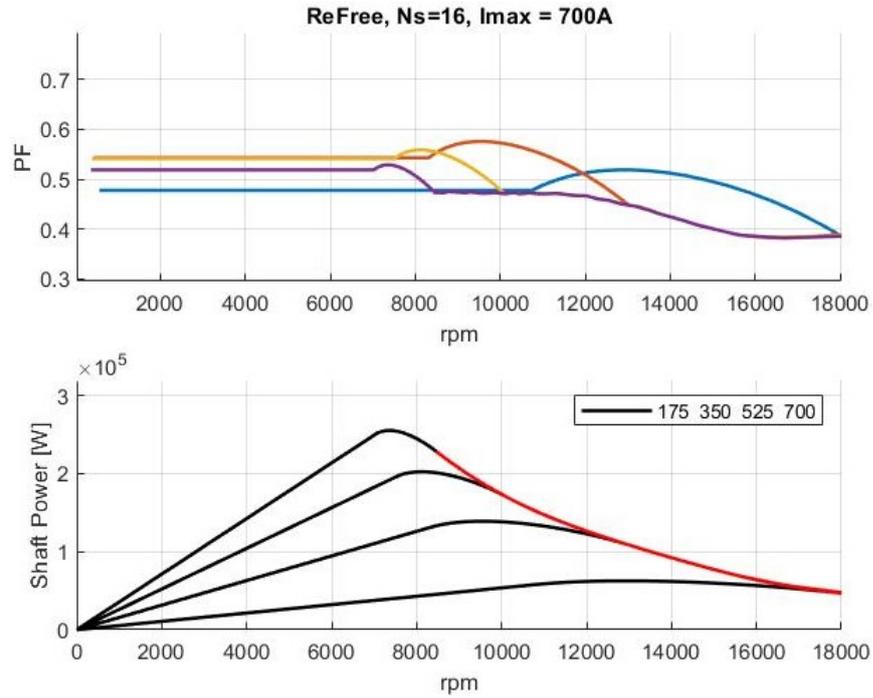


Figure 51: Option 2 - Power factor and shaft power curves at different peak current

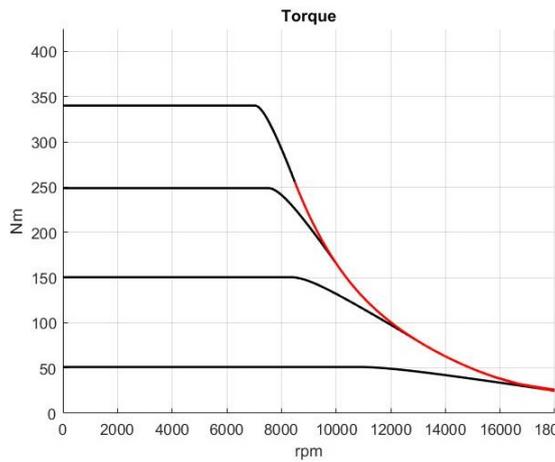


Figure 52: Option 2 - Torque

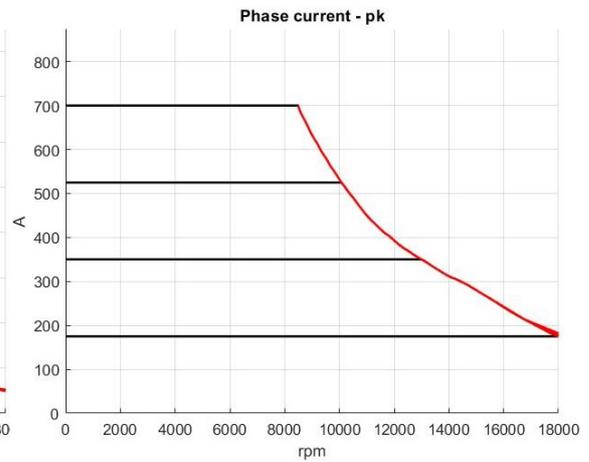


Figure 53: Option 2 - Peak phase current

Parameters	Values
Peak torque at low speed	340 Nm
Peak power at maximum speed	48 kW
Maximum peak power	255 kW
Power factor at low speed	0.53

Table 12: Option 2 - Performances

As already said, it can be foreseen a performance improvement by adjusting the ribs thickness and thanks to trade-off evaluations. Thus, the power at the maximum

speed constraint is likely to be easily fulfilled, however, on the other hand, the torque at low speed is quite deficient and probably it will remain below the limit. For these reasons, the present solution was not chosen.

At this point, the mentioned options can be summarized:

- Option 1: it violates the power at maximum speed constraint, thus it was increased, accepting a drop in the low speed torque, by reducing the turns. With an hairpin winding the next feasible turns in series per phases number is 16.
- Option 2: even though a satisfying high speed power, the low speed torque had a drop which led to violate the 380 Nm constraint.

It is clear that a 18 turns would be the optimal solution, however, with an hairpin winding and 72 slots, it is not feasible because it requests an odd number of conductor in a slot. Thus, two further design paths were investigated:

- Option 3: the number of slots were cut from 72 to 36, by adopting a $q = 2$ solution. In this way, the 18 turns in series per phase are feasible also adopting an hairpin winding.
- Option 4: maintaining the 72 slots, it was chosen to switch to a stranded winding, allowing to have 18 turns in series per phase.

4.4.4 Design option 3: $q = 2$, $N_s = 18$

A 36 slots stator allows to have 18 turns in series per phases by employing 2 parallel paths and 6 conductors in a slot with a double layer setup and with the hairpin technology.

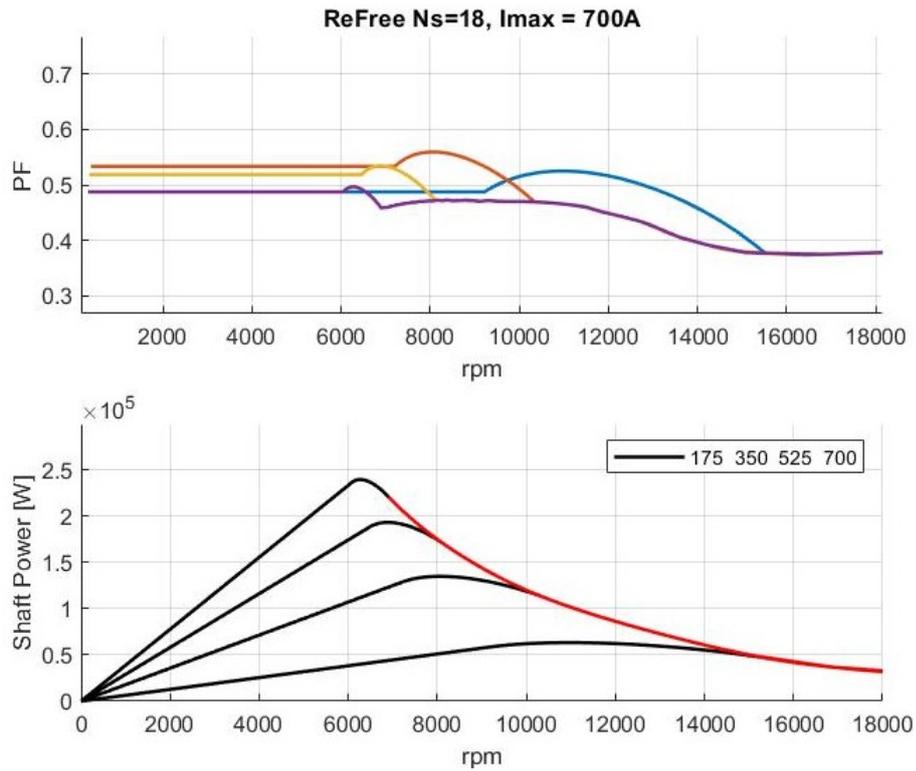


Figure 54: Option 3 - Power factor and shaft power curves at different peak current

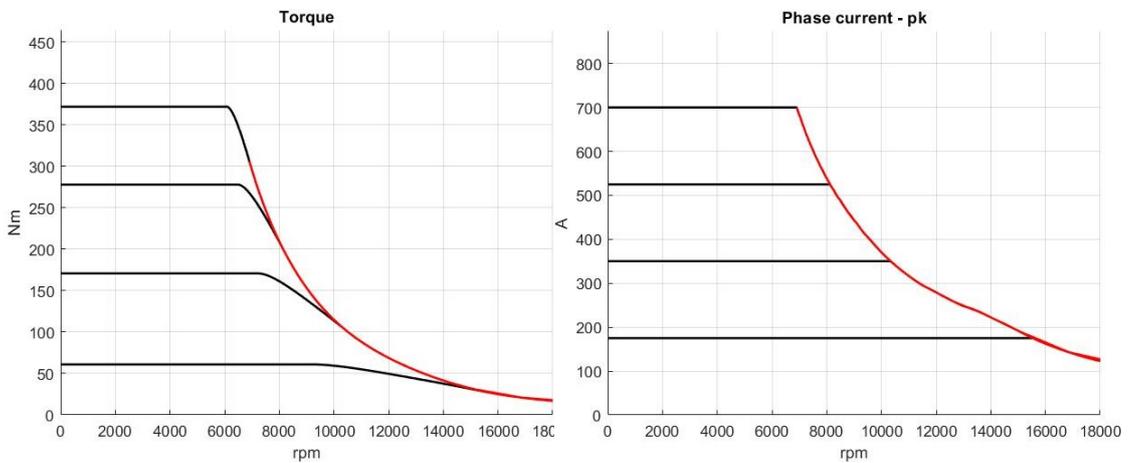


Figure 55: Option 3 - Torque

Figure 56: Option 3 - Peak phase current

Parameters	Values
Peak torque at low speed	370 Nm
Peak power at maximum speed	35 kW
Maximum peak power	240 kW
Power factor at low speed	0.52

Table 13: Option 3 - Performances

In this case, the most demanding performances, i.e. peak torque at low speed and

peak power at maximum speed, are quite close to the requirements. Thus, it is likely that further refinements might be sufficient to meet the constraints. Hence, such design was deemed satisfactory and further analysed in Motor-CAD.

4.4.5 Design option 4: stranded winding, $q = 4$, $N_s = 18$

With 72 slots, the hairpin winding does not allow to have 18 turns in series per phases, because it would request an odd number of conductor per slot. Hence, maintaining 72 slots, the stranded winding was adopted, which is related to a lower slot filling factor usually around 0.4. Further, to better suit the stranded winding, parallel tooth were employed.

Such setup led to the results shown below.

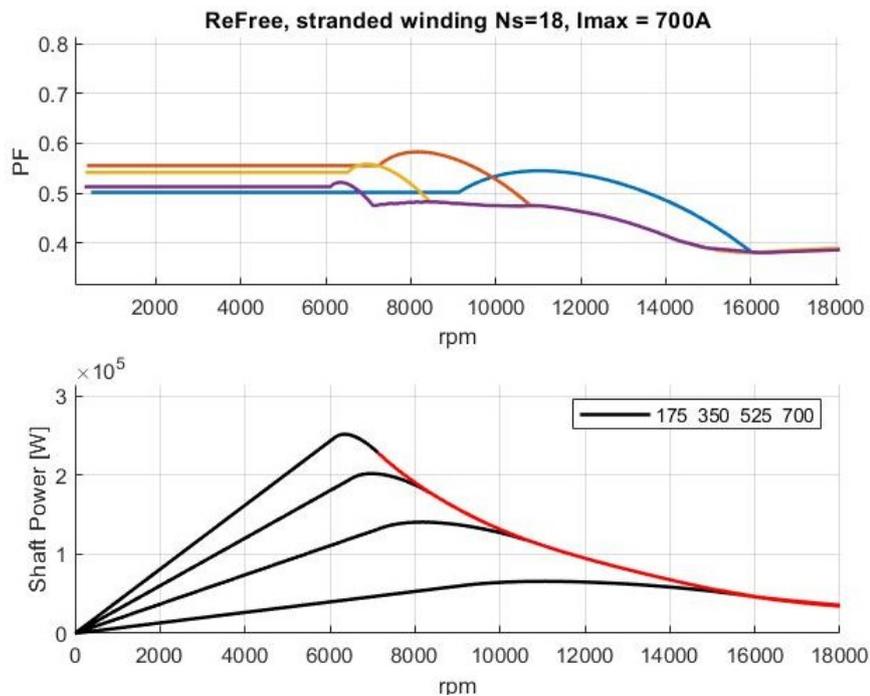


Figure 57: Option 4 - Power factor and shaft power curves at different peak current

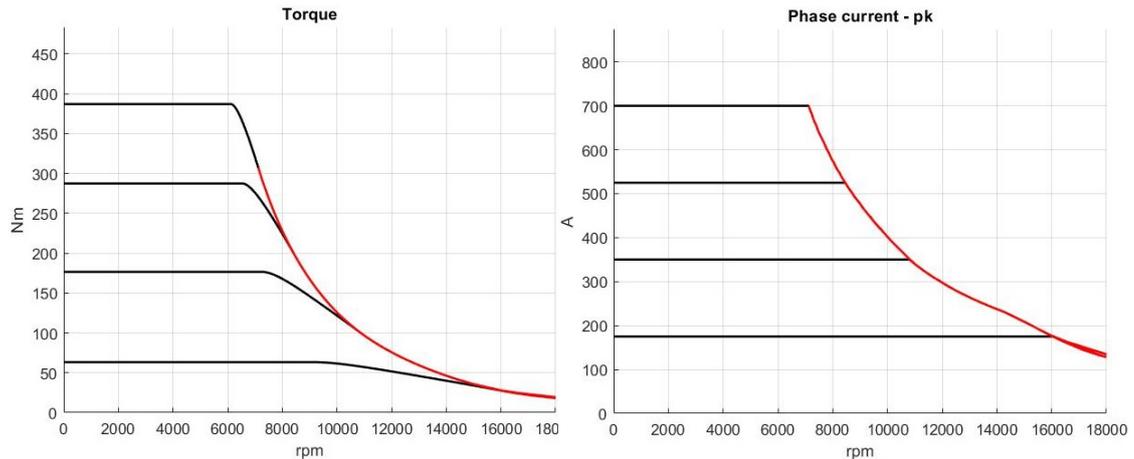


Figure 58: Option 4 - Torque

Figure 59: Option 4 - Peak phase current

Parameters	Values
Peak torque at low speed	380 Nm
Peak power at maximum speed	35 kW
Maximum peak power	250 kW
Power factor at low speed	0.56

Table 14: Option 4 - Performances

The results are quite similar to the option 3 (subsection 4.4.4), however the winding technology adopted is the stranded one, which is usually not preferred respect to the hairpin for the reasons disclosed in paragraph 4.3.4. Anyway, the present design was exported in Motor-CAD and further analysed as well as the option 3.

4.4.6 SyR-e design outcomes

The performance of the examined 4 design options are summarized and compared in tab 15.

	Option 1	Option 2	Option 3	Option 4
Slot number	72	72	36	72
Winding technology	hairpin	hairpin	hairpin	stranded
Turns in series per phases	20	16	18	18
Peak torque at low speed [Nm]	420	340	370	380
Peak power at maximum speed [kW]	26	48	35	35
Maximum peak power [kW]	240	255	240	250
Power factor at low speed	0.56	0.53	0.52	0.55
Assessment	Discarded	Discarded	Exported	Exported

Table 15: SyR-e outcomes summary

The selected designs, underlined in yellow, were further analysed in Motor-CAD thanks to the export script built in chapter 3. This was crucial to exploit the mechanical validation, which is a fundamental matter for SyR machines, as well as to quickly evaluate the optimized slot area and to run specific simulations that are not yet available in SyR-e. Last, it was also possible to accurately design the cooling system.

4.5 Design in Motor-CAD

4.5.1 Export from SyR-e to Motor-CAD

To create a Motor-CAD model identical to the SyR-e one, the export scripts built in chapter 3 were employed. Hence, the two SyncRel ISeg machines were obtained in Motor-CAD merged with two .dxf modelling the rotor, since, for the time being, the circular barriers flux have not been yet included in Motor-CAD. Anyway, it could be pointed out that the rotors of the two design options are identical, whereas the two stators are different in terms of slots number and typology (parallel slot vs parallel tooth).

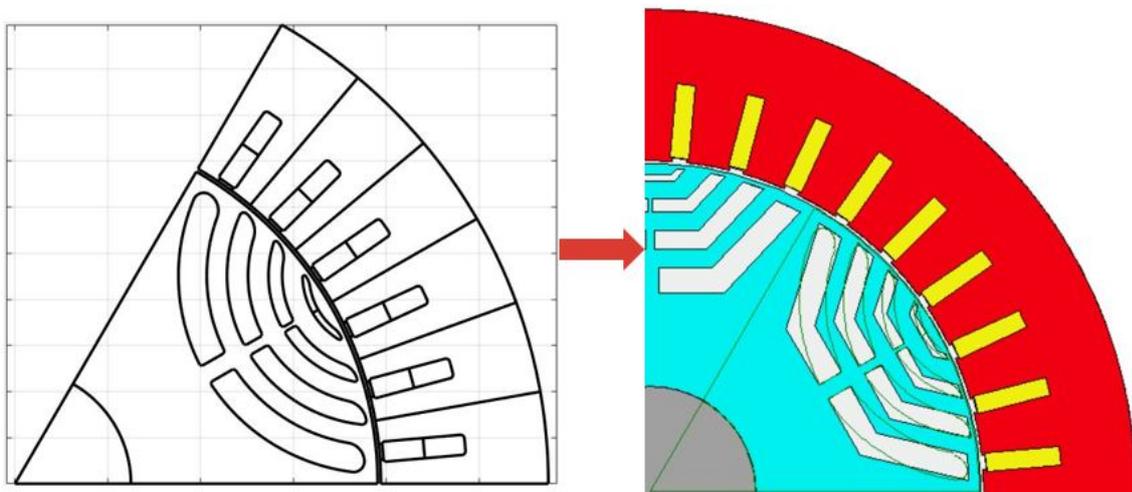


Figure 60: Option 3 - Machine model export to Motor-CAD

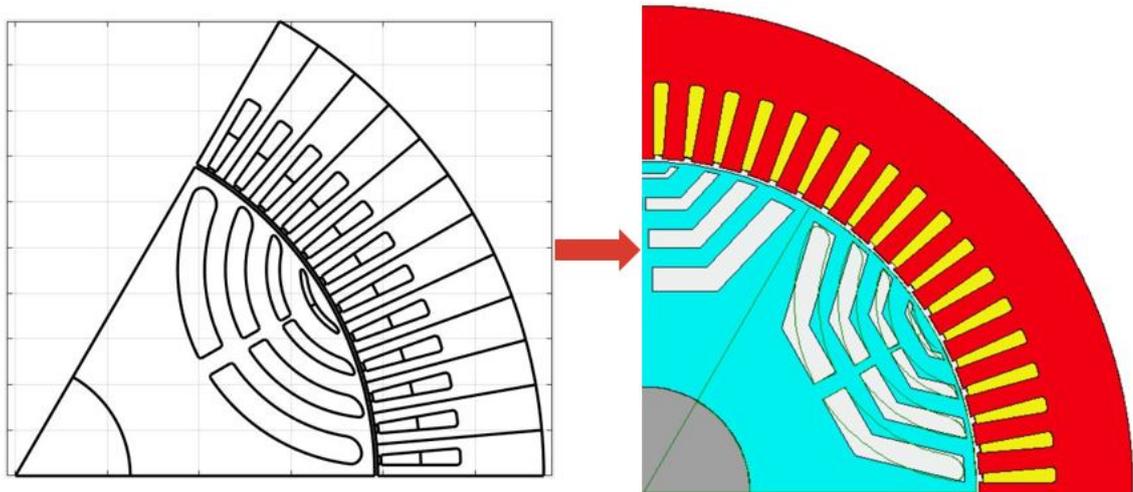


Figure 61: Option 4 - Machine model export to Motor-CAD

4.5.2 Ribs thickness adjustments

Firstly, the ribs roughly sized in SyR-e were assessed by the Motor-CAD mechanical FEA and the outcomes reported below.

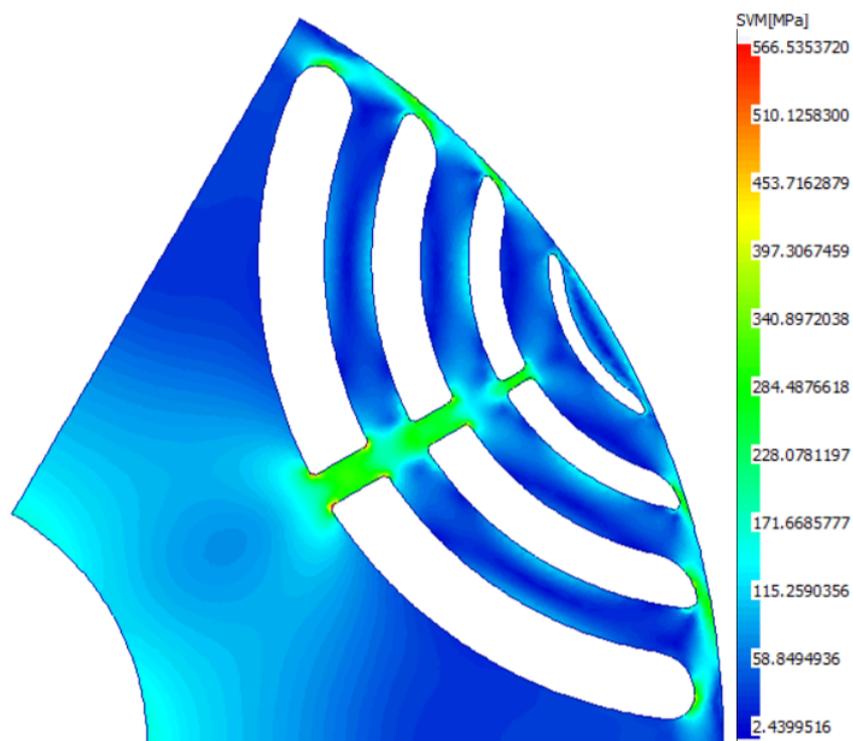


Figure 62: Ribs preliminary design - Mechanical stress

By zooming, it appears clear that the main issues are related to the sharp corners at the internal side of the flux barriers (in fig. [63](#) the regions exceeding the 340 MPa

are underlined in white), while the tangential ribs can be slightly thinned.

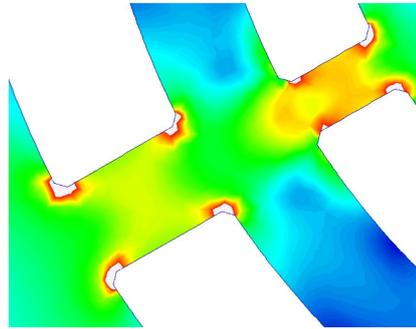


Figure 63: Ribs preliminary design - Mechanical stress (a white region indicates an area exceeding the 340 MPa)

Hence, two actions were executed:

- Round the radial ribs corners: action performed by modifying the .dxf file with AutoCAD.

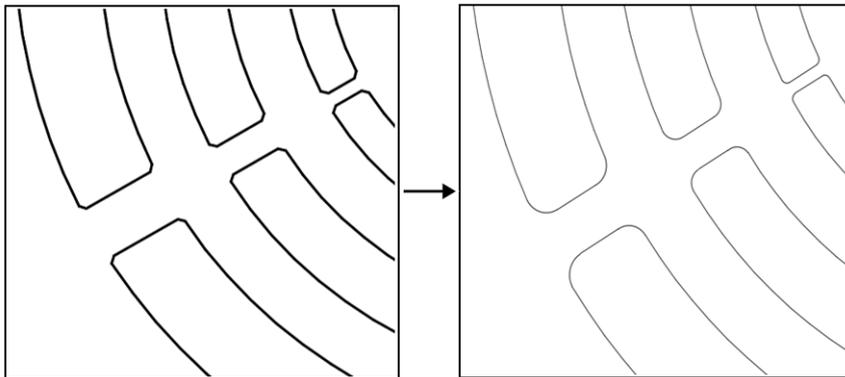


Figure 64: Rounding process at the radial ribs by means of AutoCAD

- Adjust the tangential and radial ribs thickness: as previously said, the preliminary design overestimated the tangential ribs, thus, they were slightly lowered until the mechanical stress limit was met. Such process, already displayed in fig. 42, was made cumbersome by the absence of the circular flux barrier in Motor-CAD, indeed, for every iteration, the tangential ribs had to be sized in SyR-e and the .dxf exported to Motor-CAD in order to compute the mechanical stress through a FEA analysis. The iteration process was repeated until the most suitable ribs design was detected. While, concerning the radial ribs, the estimation in SyR-e was executed by a specific script, however, further adjustments were applied by means of mechanical FEA and iterations to achieve an optimal solution.

Tangential ribs thickness	Preliminary sizing	Final sizing
1 st barrier	2.7 mm	2.5 mm
2 nd barrier	1.2 mm	1 mm
3 th barrier	1 mm	0.8 mm
4 th barrier	1 mm	0.8 mm

Table 16: Preliminary and final tangential ribs sizing

Radial ribs thickness	Preliminary sizing	Final sizing
1 st barrier	4.14 mm	3.2 mm
2 nd barrier	2.43 mm	2.5 mm
3 th barrier	1.09 mm	1 mm
4 th barrier	0 mm	0 mm

Table 17: Preliminary and final radial ribs sizing

The outcome of this section was a sufficient mechanical validation, however it was reached a small margin respect to the material Yield stress (460 MPa).

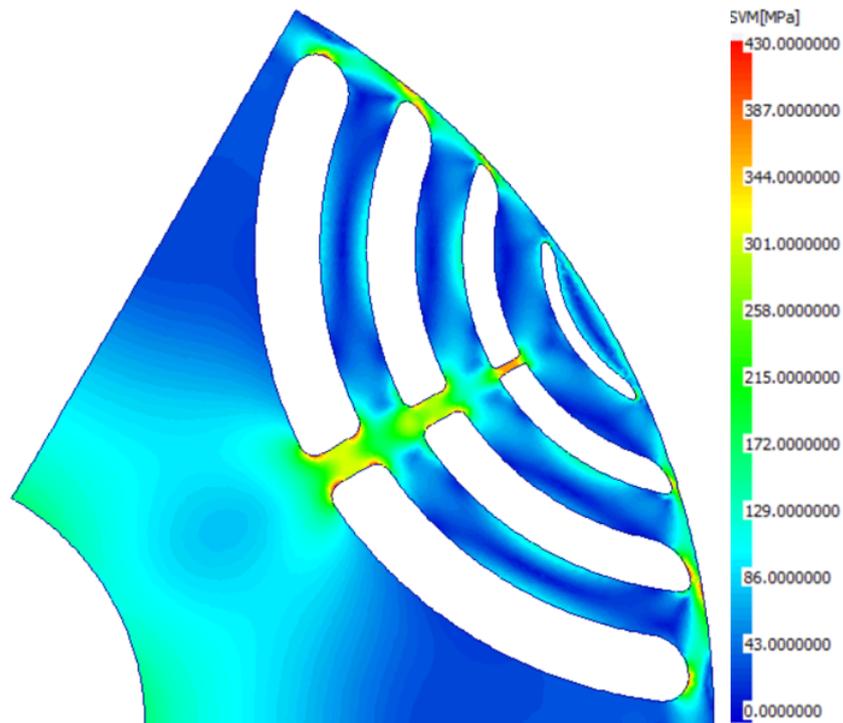


Figure 65: Ribs final design - Mechanical stress

The situation before and after the ribs adjustments are compared in tab [18](#). It can be notified that the maximum displacement was not a demanding issue, since even with the preliminary sizing it was below the 15% of the airgap length.

	Preliminary sizing	Final sizing
Maximum mechanical stress	508 MPa	430 MPa
Maximum displacement	92 μm (13% of the airgap)	84 μm (11% of the airgap)

Table 18: After and before the ribs adjustments

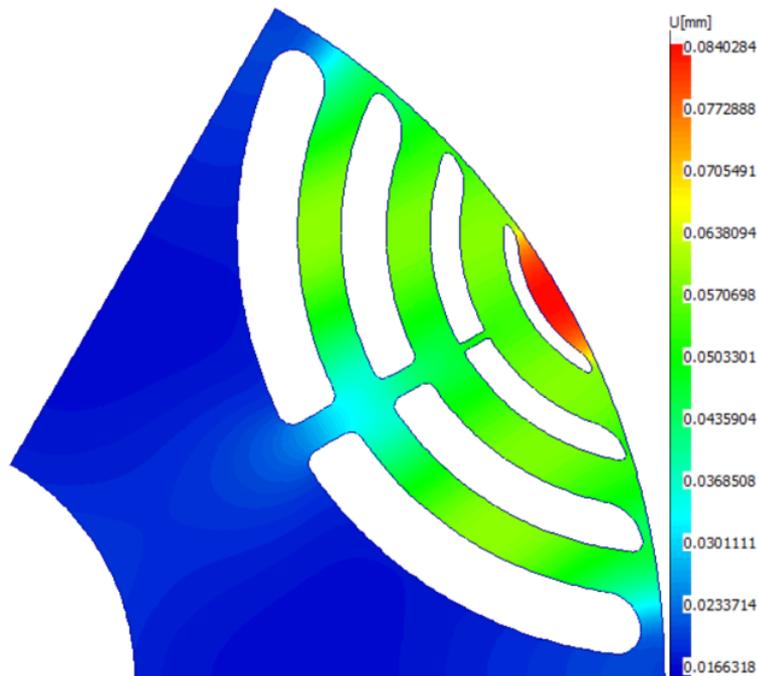


Figure 66: Ribs final design - Displacement

4.5.3 Slot area trade-off

- Option 3: 36 slots - hairpin winding

SyR-e provided a slot length of 18.51 mm, however such parameter was further investigated by means of the Motor-CAD sensitivity section, as shown in Fig. 67. A shorter slot furnishes an higher peak torque and a slightly greater power factor, on the other hand, the current density is increased, leading to a rise in the winding temperature. Hence, a satisfying trade-off point has to be found, keeping the winding temperature below 180°C over the demanding drive-cycle US06 and providing acceptable continuous performance.

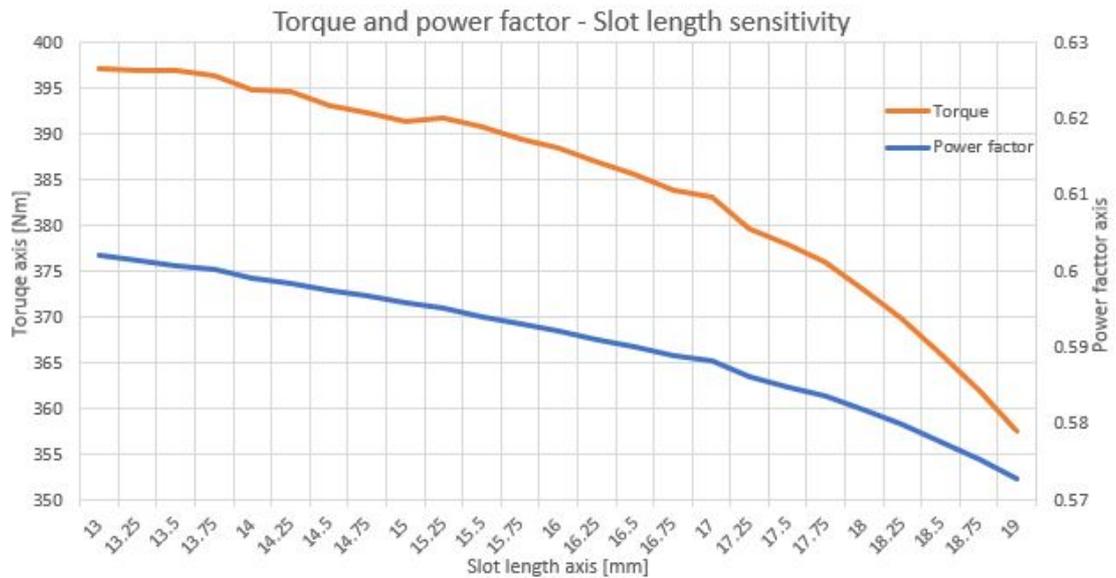


Figure 67: Option 3 - Torque and power factor vs slot length

In tab. 19 some significant points of the sensitivity analysis are reported. Note that the maximum winding temperature was evaluated considering the most demanding drive cycle US06 repeated 7 times, while the continuous torque displayed is referred to 3000 rpm.

Slot length	Peak torque	Max winding temperature	Continuous torque
18.51 mm	366 Nm	108°C	170 Nm
15.25 mm	392 Nm	125°C	140 Nm
13 mm	397 Nm	145°C	115 Nm

Table 19: Option 3 - Slot length trade-off

The solution deemed most suitable is with a 15.25 mm slot length, which provided a valuable peak torque with discrete continuous performances.

- Option 4: 72 slots - stranded winding

The same procedure was repeated aiming to detect the best trade-off point for the design option 4. In this case, SyR-e provided a slot slightly shorter than the previous case, corresponding to 18.01 mm.

The Motor-CAD sensitivity analysis furnished the results in fig. 68.

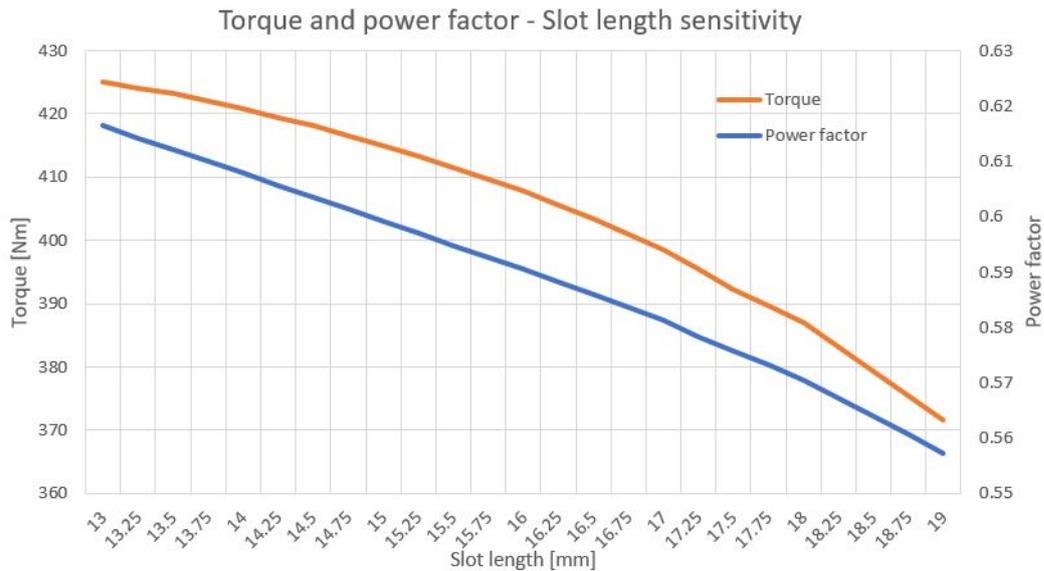


Figure 68: Option 4 - Torque and power factor vs slot length

The main trade-off points evaluated are listed in tab. 20. As well as the previous case, the considered drive-cycle was given by seven US06 in series and the continuous torque is referred to 3000 rpm.

Slot length	Peak torque	Max winding temperature	Continuous torque
18 mm	387 Nm	130°C	130 Nm
17 mm	399 Nm	135°C	126 Nm
16 mm	408 Nm	145°C	110 Nm
13 mm	425 Nm	205°C	75 Nm

Table 20: Option 4 - Slot length trade-off

The analysis led to fix the slot length to 18 mm, obtaining similar results of the previous option, even though with slightly smaller peak and continuous torque.

4.5.4 Electromagnetic module

Once the machine design was seized, the electromagnetic module was checked. In particular the stator winding was more deeply inquired by defining the conductors section, copper insulation, impregnation area and the winding pattern. Further, the winding goodness was evaluated through the MMF and winding factor harmonics.

- Option 3: 36 slots - hairpin winding
In the electromagnetic module the radial, axial and 3D views were displayed as reported below.

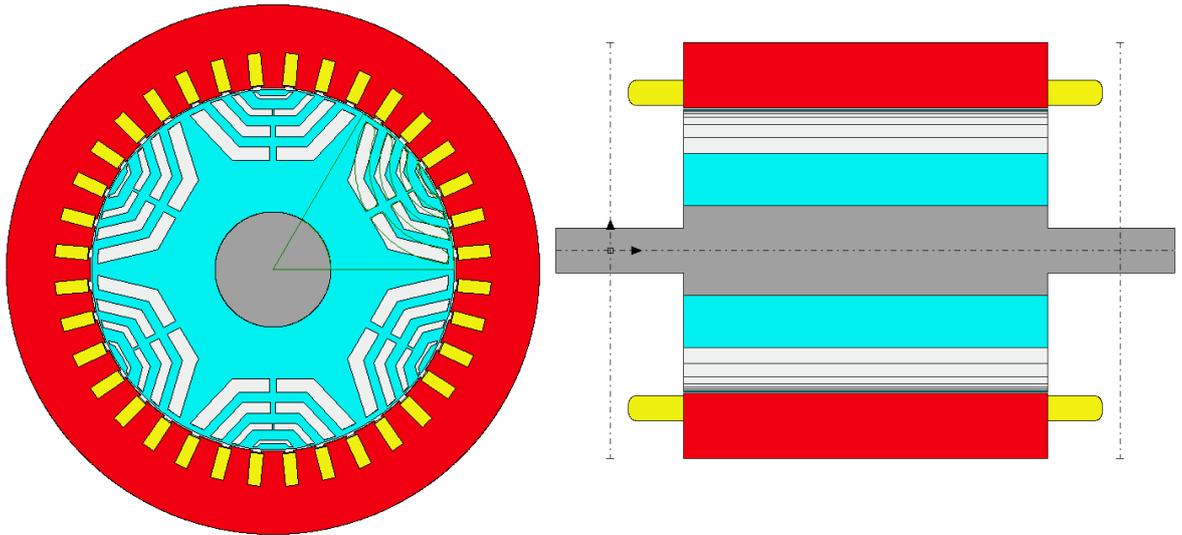


Figure 69: EMag module - Radial and axial view

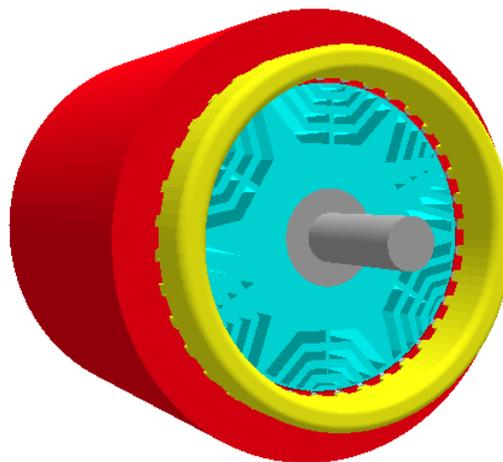


Figure 70: EMag module - 3D view

A double layers 3-phases winding was selected with 2 parallel paths, a throw of 7 and 6 conductor per slot. Such winding pattern is reported on the left of Fig. [4.5.4](#), whereas on the right the conductors definition in a slot is shown. The hairpin technology allowed to increase the slot filling factor up to 0.67. Further, the analysis provided a winding factor with a first harmonic of 0.933, which was deemed an acceptable value; while the undesired 5-th and 7-th harmonics are both roughly equal to 0.067, which are satisfying low values.

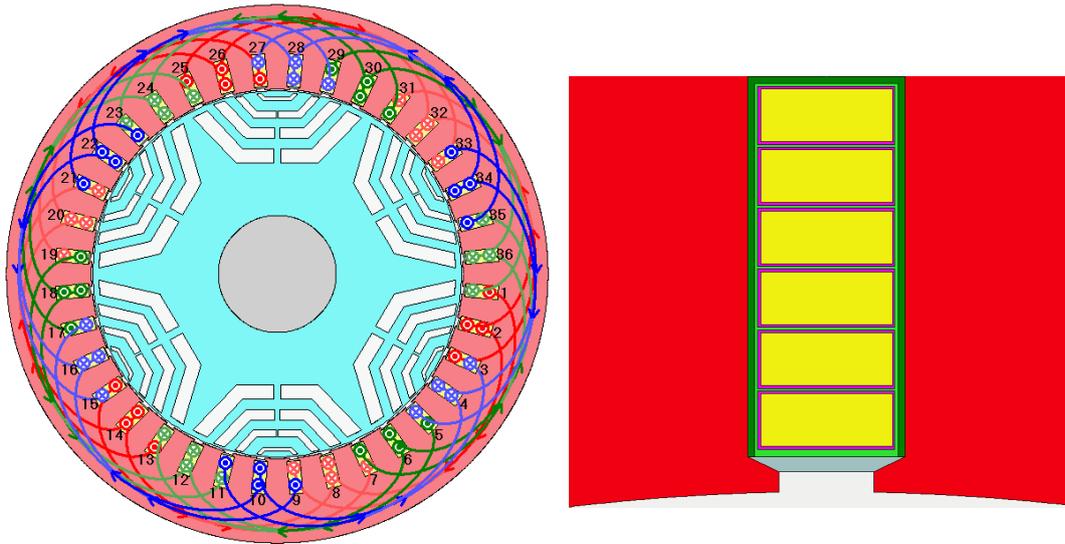


Figure 71: Winding pattern and conductors definition

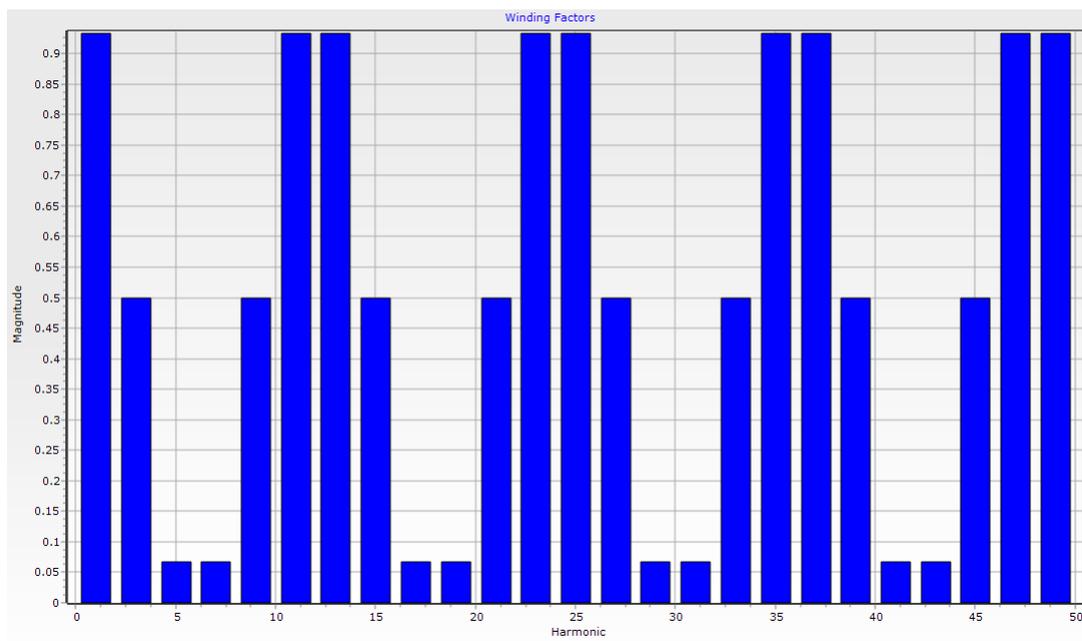


Figure 72: Winding factor harmonics

Harmonic	Magnitude
1	0.933013
3	0.500000
5	0.066987
7	0.066987
9	0.500000
11	0.933013
13	0.933013
15	0.500000
17	0.066987
19	0.066987
21	0.500000

Figure 73: Winding factor harmonics

- Option 4: 72 slots - stranded winding
As well as done for the option 3, also for the option 4 the several EMag views are reported.

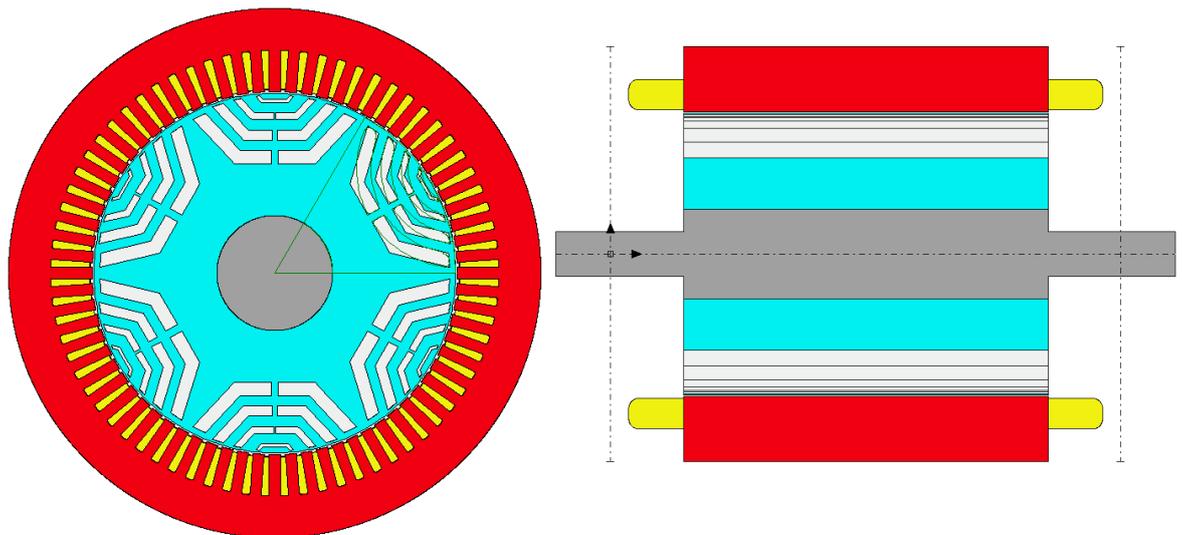


Figure 74: EMag module - Radial and axial view

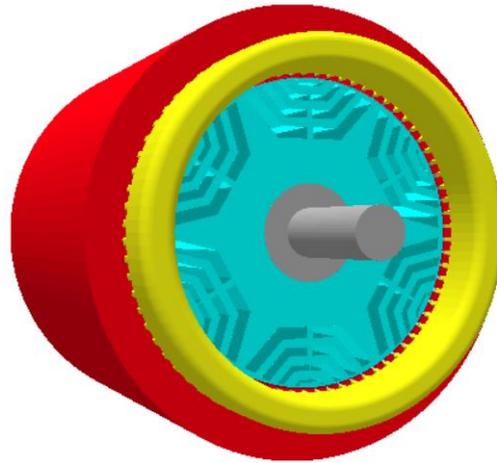


Figure 75: EMag module - 3D view

A single layers 3-phases winding was selected with 6 parallel paths, a throw of 11 and 9 conductors per slot. Such winding pattern is reported at the left of Fig. 4.5.4, whereas at the right the conductors definition in a slot is shown. According to the electric machines literature, the stranded technology bears a low slot filling factor, which is roughly 0.4. The analysis provided a winding factor first harmonic of 0.958, which is a satisfying value; while the undesired 5-th and 7-th harmonics are respectively equal to 0.205 and 0.158, which are decent values.

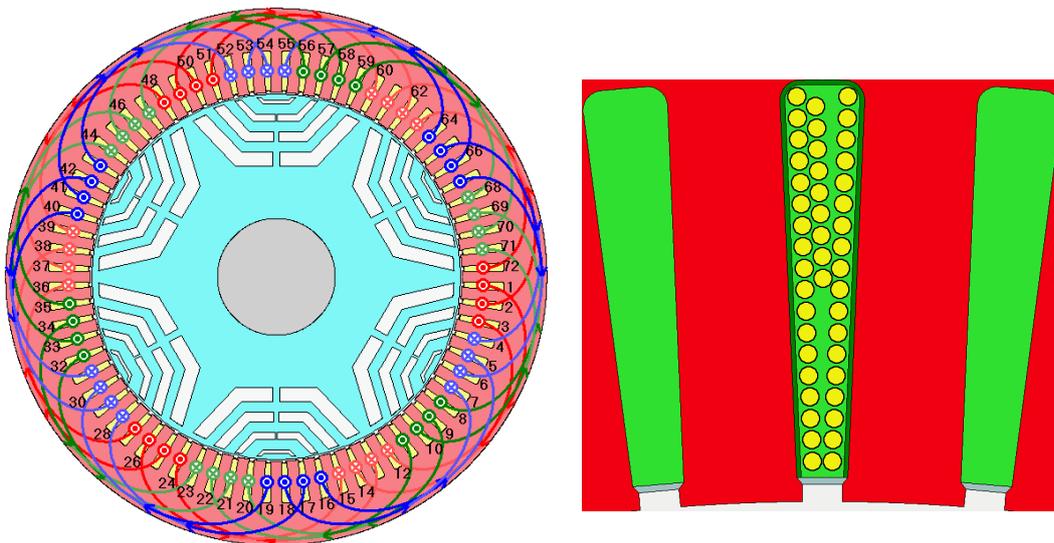


Figure 76: Winding pattern and conductors definition

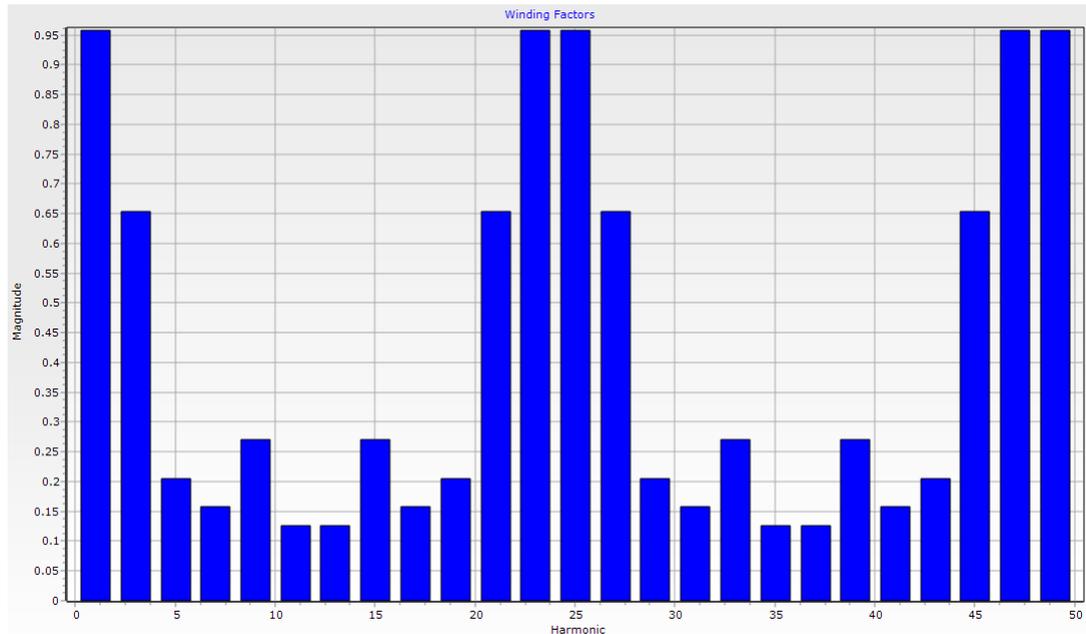


Figure 77: Winding factor harmonics

Harmonic	Magnitude
1	0.957662
3	0.653281
5	0.205335
7	0.157559
9	0.270598
11	0.126079
13	0.126079
15	0.270598
17	0.157559
19	0.205335
21	0.653281

Figure 78: Winding factor harmonics

4.5.5 Thermal module

The same cooling system is adopted for both the design options: a spiral water jacket with 50/50 water glycol, a fluid volume flow rate of 6.5 l/min and 65°C as inlet temperature. Further, to improve the heat management at the end-winding sides, potted cavities were implemented. The thermal system was made less demanding to size thanks to the cold rotor, which is one of the synchronous reluctance machine advantages. The main source of heat is the copper, indeed, it was critical to manage the heat produced by the end-winding, aiming to keep the temperature below 180° (insulation class H).

The radial, axial and 3D views in the thermal module for both the options are reported below.

- Option 3: 36 slots - hairpin winding

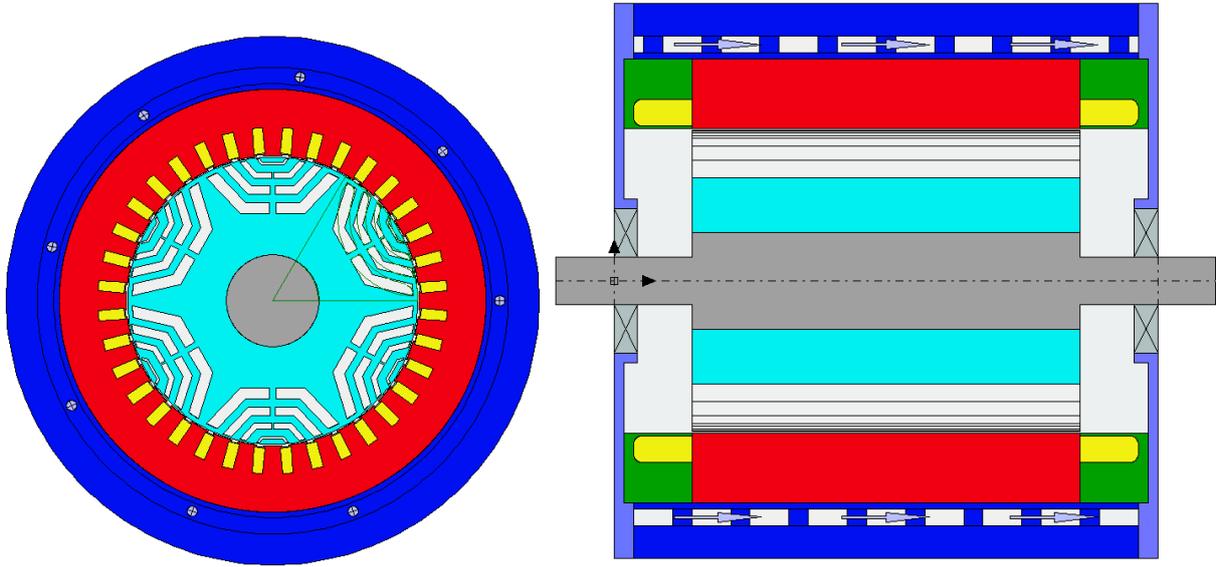


Figure 79: Thermal module - Radial and axial view

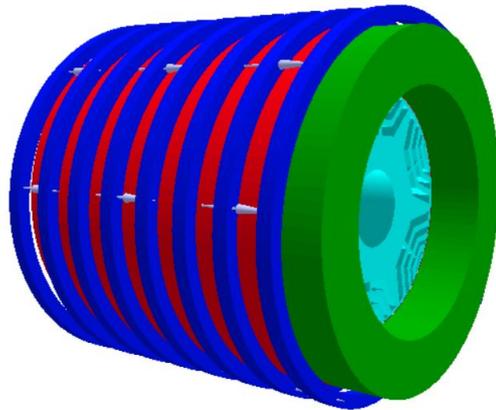


Figure 80: Thermal module - 3D view

- Option 4: 72 slots - stranded winding

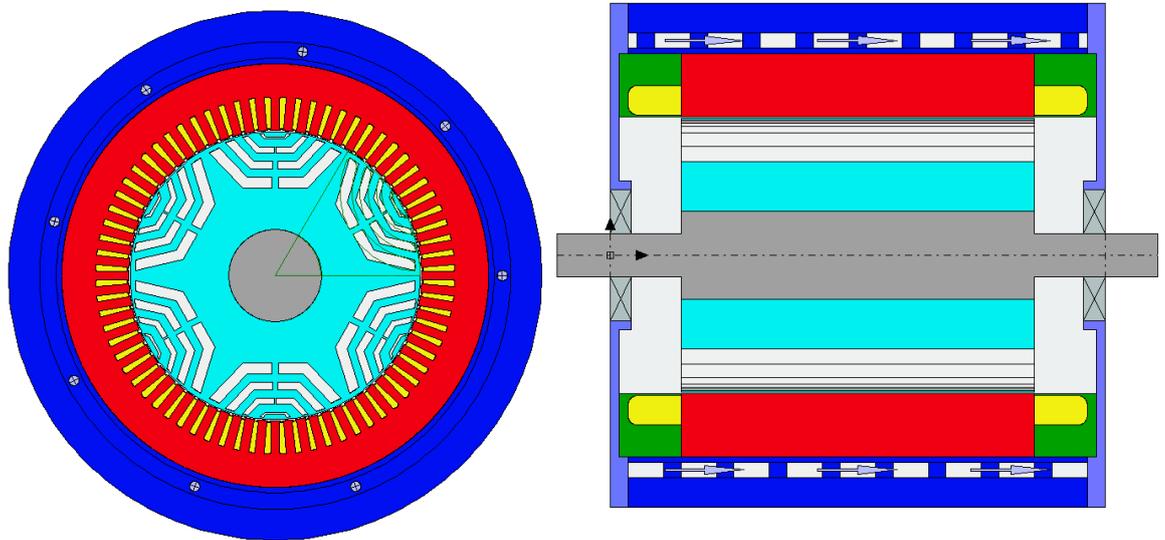


Figure 81: Thermal module - Radial and axial view

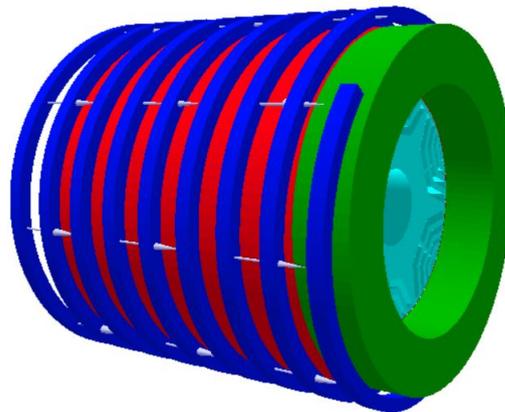


Figure 82: Thermal module - 3D view

4.5.6 Lab module

The last module exploited in Motor-CAD is the Lab, where a wide range of analysis are available. The resulting graphs are displayed below, while the values collected are summarized in tables in the next paragraph [4.6](#). The listed simulations were executed:

- a) Peak and continuous torque vs speed
 - b) Peak and continuous power vs speed
 - c) Phase advance vs speed
 - d) Power factor vs speed
 - e) L_d and L_q vs speed
 - f) Efficiency map
- Option 3: 36 slots - hairpin winding

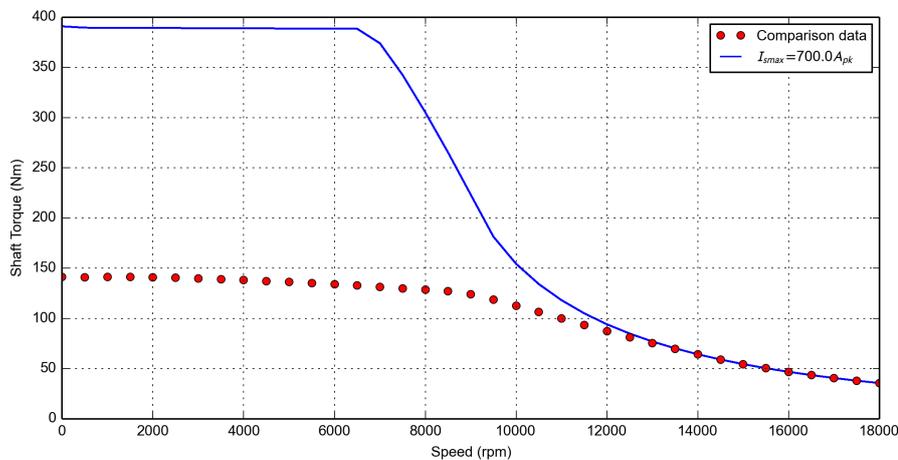


Figure 83: Option 3 - Peak torque (blue line) and continuous torque (red dots) vs speed

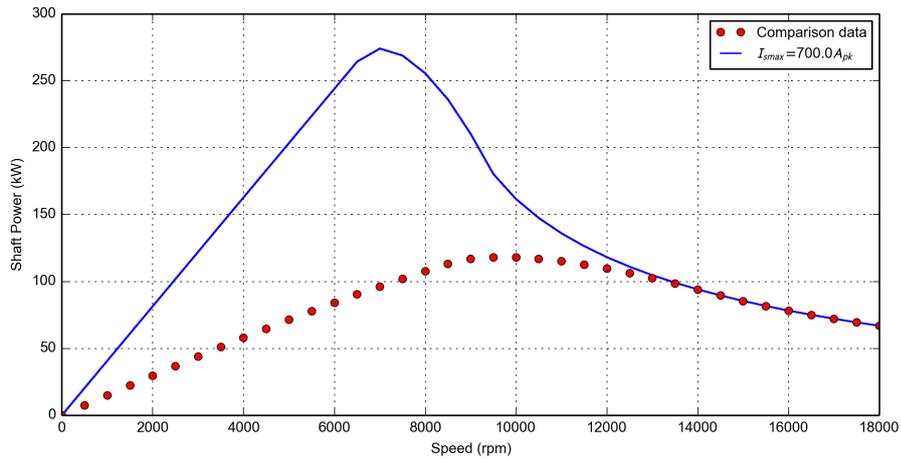


Figure 84: Option 3 - Peak power (blue line) and continuous power (red dots) vs speed

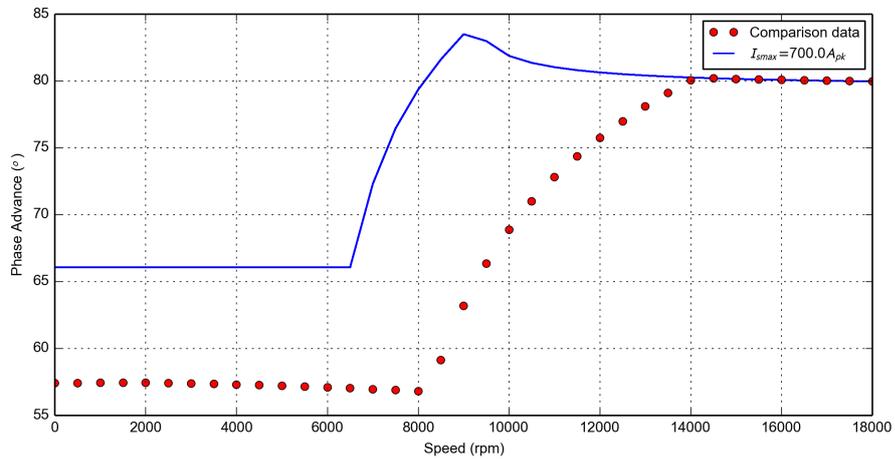


Figure 85: Option 3 - MTPA angle vs speed, peak operation points (blue line) and continuous operation points (red dots)

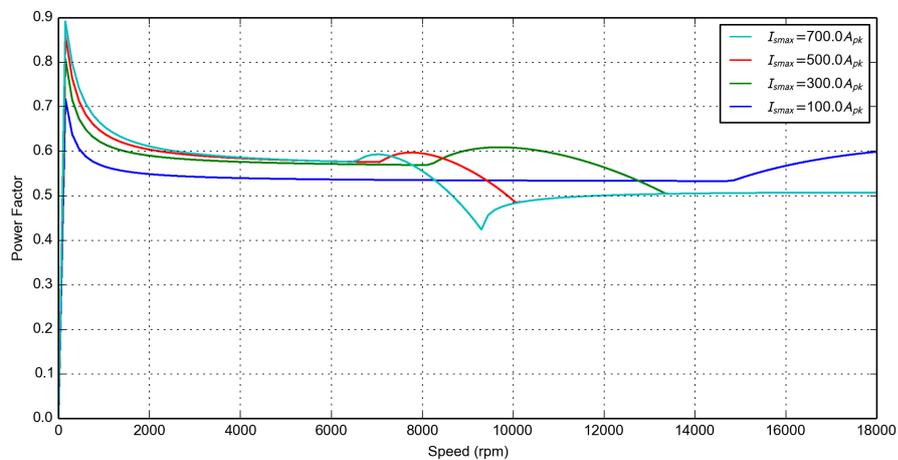


Figure 86: Option 3 - Power factor vs speed

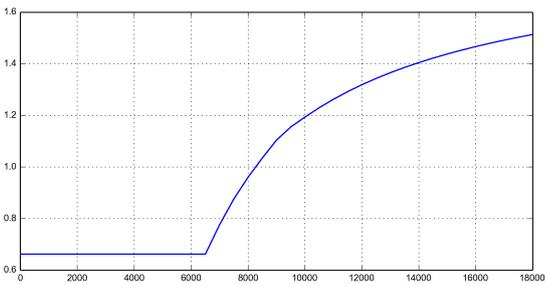
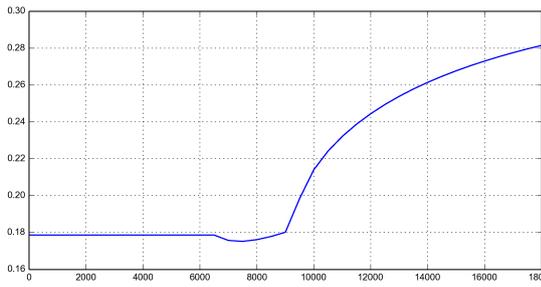


Figure 87: Option 3 - L_d [mH] vs speed Figure 88: Option 3 - L_q [mH] vs speed

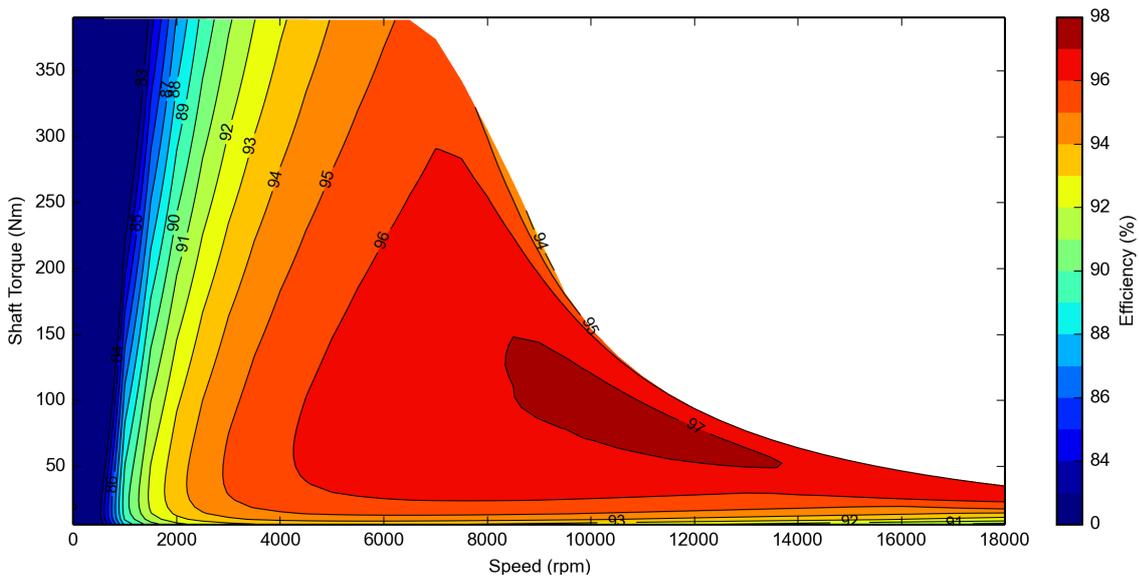


Figure 89: Option 3 - Efficiency map

- Option 4: 72 slots - stranded winding

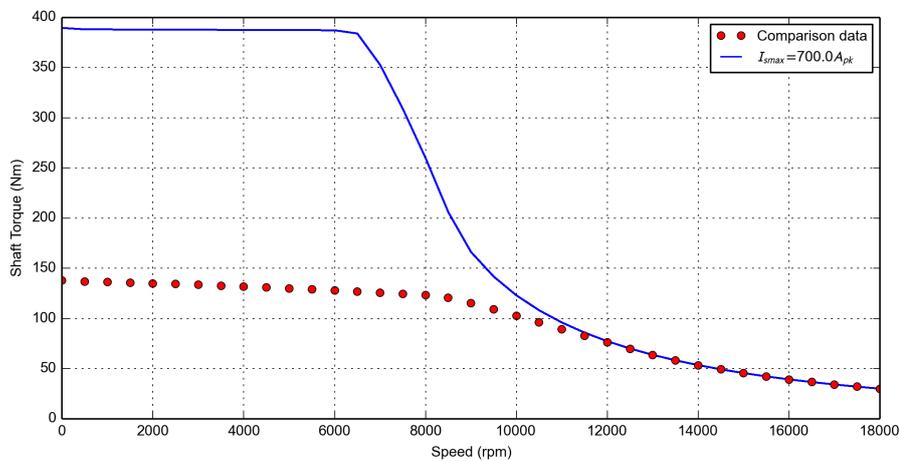


Figure 90: Option 4 - Peak torque (blue line) and continuous torque (red dots) vs speed

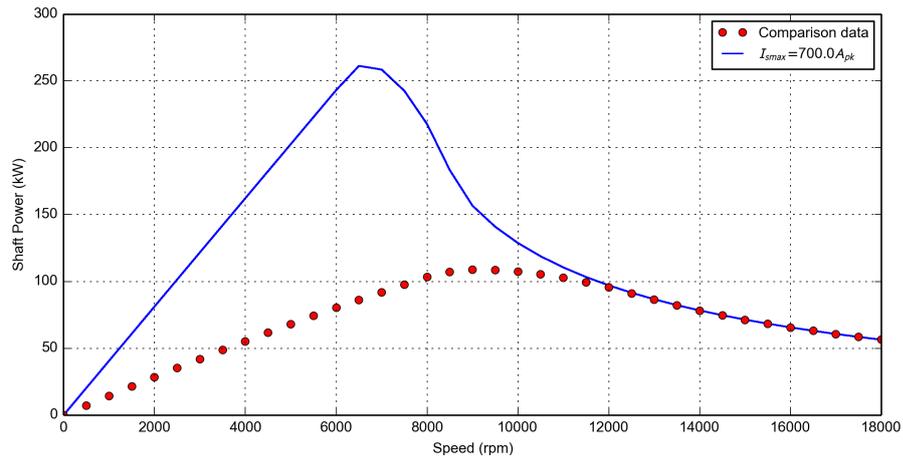


Figure 91: Option 4 - Peak power (blue line) and continuous power (red dots) vs speed

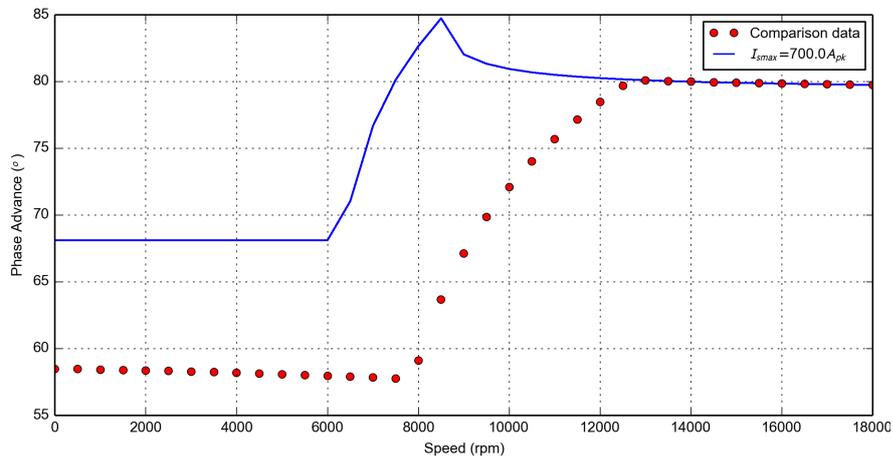


Figure 92: Option 4 - MTPA angle vs speed, peak operation points (blue line) and continuous operation points (red dots)

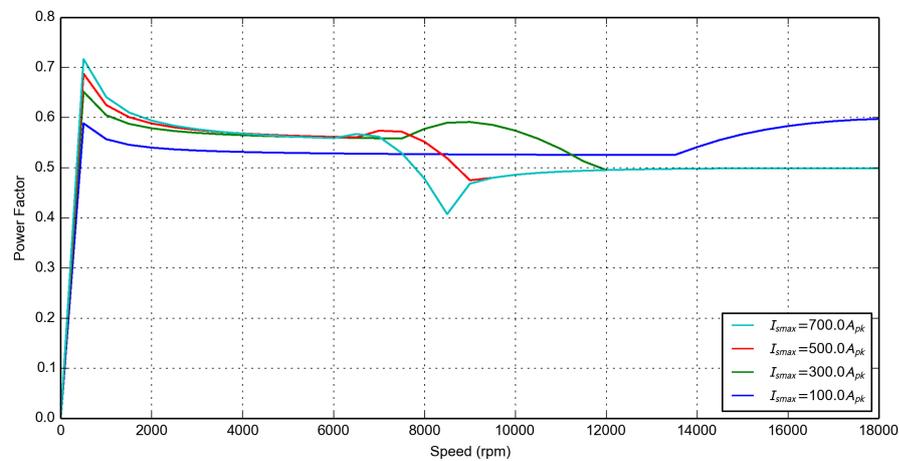


Figure 93: Option 4 - Power factor vs speed

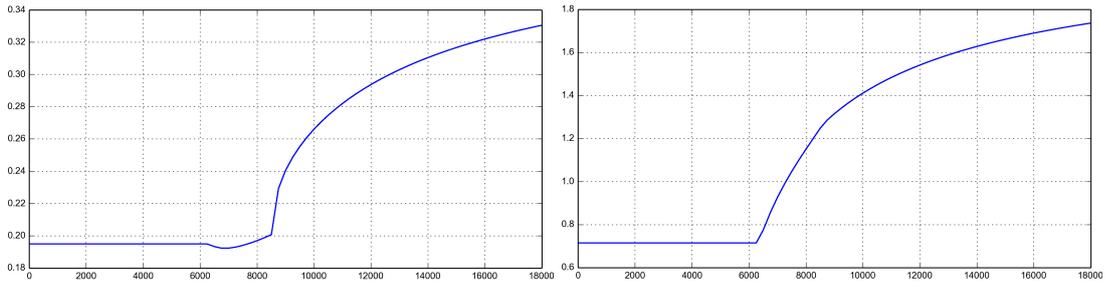


Figure 94: Option 4 - L_d [mH] vs speed Figure 95: Option 4 - L_q [mH] vs speed

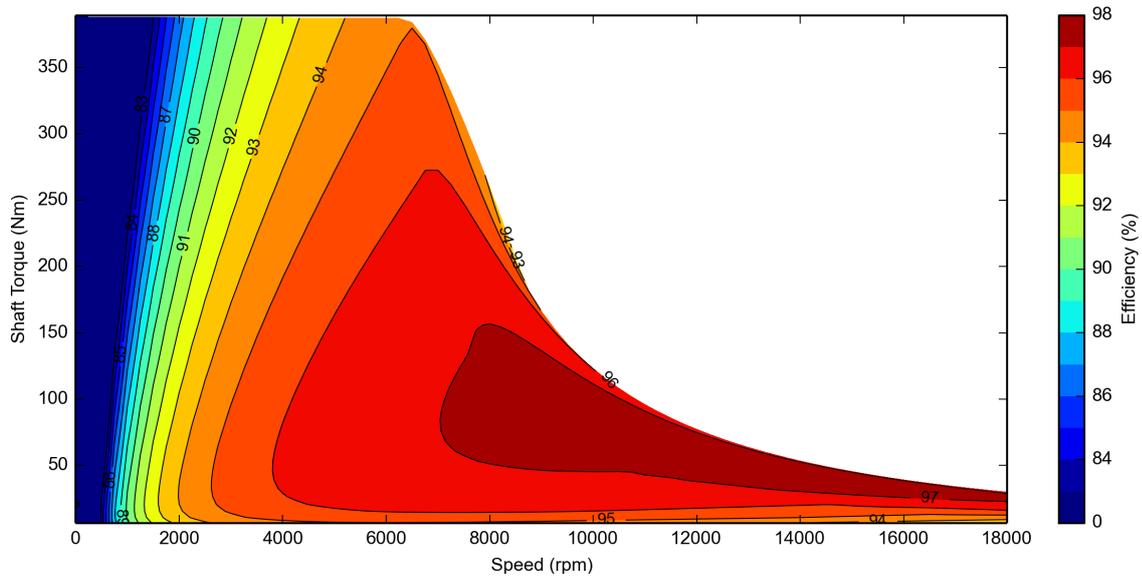


Figure 96: Option 4 - Efficiency map

4.6 Comparison

In this paragraph two comparison were executed:

1. 700 A_{pk} : option 3 (36 slots and hairpin winding) vs option 4 (72 slots and stranded winding);
2. option 3 with 700 A_{pk} vs option 3 with 800 A_{pk} and reviewed dimensions.

4.6.1 700 A_{pk} case

The output performance resulted mainly comparable, however whenever there is a significant difference between the two options, the better one is underlined in red. Comparing the results, the hairpin winding option appeared to be slightly preferable thanks to an higher peak power at maximum speed and better heat management, which allows to sustain the peak power at 5000 *rpm* for more than one minute.

	Option 3	Option 4
Pole pairs	3	3
Slot number	36	72
Number of flux barrier layers	3	3
Airgap thickness [mm]	0.7	0.7
Stator outer radius [mm]	115	115
Rotor radius [mm]	78.2	78.2
Stack length [mm]	200	200
Winding typology	hairpin	stranded
Copper insulation	class H (180°C)	class H (180°C)
Slot typology	parallel slot	parallel tooth
Turns in series per phases	18	18
Tangential ribs thickness [mm]	2.5 - 1 - 0.8 - 0.8	2.5 - 1 - 0.8 - 0.8
Radial ribs thickness [mm]	3.2 - 2.5 - 1 - 0	3.2 - 2.5 - 1 - 0
Steel material	M235-35A	M235-35A

Table 21: 700 A_{pk} - Option 3 and 4 inputs comparison

	Option 3	Option 4
Maximum peak current [A]	700	700
DC voltage [V]	800	800
Peak torque up to 5 krpm [Nm]	392	387
Peak power at maximum speed [kW]	67	56
Maximum peak power [kW]	270	260
Power factor at 5 krpm and max torque	0.59	0.57
Continuous torque at low speed [Nm]	140	135
Continuous power at maximum speed [kW]	67	56
Maximum efficiency	97 %	97.5 %
Points above 95% of efficiency	55 %	54.8 %
Ripple at maximum torque	8.6 %	10.6 %
with stator skewing	1.8 %	4 %
Ripple at maximum speed and torque	10 %	14.7 %
with stator skewing	3 %	5 %
Weight	57	54
Power density [kW/kg]	4.8	4.82
Torque density [Nm/kg]	7	7.2

Table 22: $700A_{pk}$ - Option 3 and 4 electromagnetic performance comparison

	Option 3	Option 4
Maximum time at peak power and at 5 krpm [s]	72	45
Maximum temperature over a drive-cycle [$^{\circ}C$]	125	130
Maximum low speed torque for a max period of 60 s [Nm]	peak performance	350

Table 23: $700A_{pk}$ - Option 3 and 4 thermal performance comparison

	Option 3	Option 4
Maximum mechanical stress at max speed [MPa]	430	430
Maximum displacement at max speed [μm]	83	83

Table 24: $700A_{pk}$ - Option 3 and 4 mechanical performance comparison

4.6.2 $800A_{pk}$ case

Increase the peak current allows to cut both radial and axial machine dimensions; thus, the weight drops while the torque and power densities raise. A decreased radial dimensions lowers the ribs mechanical stress, which was one of the main disadvantages of both the $700A_{pk}$ design options. On the other side, the electromagnetic peak performance are slightly worse, while the continuous ones are significantly decreased. To achieve the new design, the proposed methodology in Fig. 41 was used, retracing the procedure described in the present chapter.

	Option 3 700 Apk	Option 3 800 Apk
Pole pairs	3	3
Slot number	36	36
Number of flux barrier layers	3	3
Airgap thickness [mm]	0.7	0.7
Stator outer radius [mm]	115	110
Rotor radius [mm]	78.2	75.5
Stack length [mm]	200	185
Winding typology	hairpin	hairpin
Copper insulation	class H (180°C)	class H (180°C)
Slot typology	parallel slot	parallel slot
Turns in series per phases	18	18
Tangential ribs thickness [mm]	2.5 - 1 - 0.8 - 0.8	2.2 - 0.9 - 0.5 - 0.6
Radial ribs thickness [mm]	3.2 - 2.5 - 1 - 0	3.2 - 2.3 - 0.8 - 0
Steel material	M235-35A	M235-35A

Table 25: Option 3 700 A_{pk} and 800 A_{pk} - Inputs comparison

	Option 1 700 Apk	Option 2 800 Apk
Maximum peak current [A]	700	800
DC voltage [V]	800	720
Peak torque up to 5 krpm [Nm]	392	390
Peak power at maximum speed [kW]	67	63
Maximum peak power [kW]	270	262
Power factor at 5 krpm and max torque	0.59	0.56
Continuous torque at low speed [Nm]	140	110
Continuous power at maximum speed [kW]	67	63
Maximum efficiency	97 %	96.7 %
Points above 95% of efficiency	55 %	37 %
Ripple at maximum torque	8.6 %	9.3%
with stator skewing	1.8 %	1.9 %
Ripple at maximum speed and torque	10 %	10.8%
with stator skewing	3 %	3.5%
Weight	57	48
Power density [kW/kg]	4.8	5.5
Torque density [Nm/kg]	7	8.12

Table 26: Option 3 700 A_{pk} and 800 A_{pk} - Electromagnetic performance comparison

	Option 3 700 Apk	Option 3 800 Apk
Maximum time at peak power [s]	72	29
Maximum temperature over a drive-cycle [$^{\circ}C$]	125	170
Maximum low speed torque for a max period of 60 s [Nm]	peak performance	270

Table 27: Option 3 $700A_{pk}$ and $800A_{pk}$ - Thermal performance comparison

	Option 1 700 Apk	Option 1 800 Apk
Maximum mechanical stress at max speed [MPa]	430	340
Maximum displacement at max speed [μm]	83	69

Table 28: Option 3 $700A_{pk}$ and $800A_{pk}$ - Mechanical performance comparison

5 Conclusions

This report investigated the SyR motor design, focusing on proposing an innovative comprehensive procedure. The key contributions of the Thesis are:

- The coupling between the two software SyR-e and Motor-CAD via Matlab: this was executed as presented in Chapter 3 and achieving a remarkable low percentage error. Furthermore, this gave the opportunity to benchmark SyR-e with a commercial software as Motor-CAD, comparing the results accuracy and the calculation times.
- A new SyR-e version released with a section entirely dedicated to Motor-CAD: once the coupling was created, a new section in SyR-e was introduced allowing the export to Motor-CAD as well as to run thermal and magnetic simulations in Motor-CAD.
- A proposal of an original design procedure for SyR machines: once the coupling was made, this synergy turns into the most complete design package available on the market, including accessible design equations, magnetic and mechanical FEA validation, trademark thermal analysis and drive operating profiles evaluation.
- Validate the procedure by designing an automotive high performance electric machine: an high speed 200 kW SyR motor was designed in Chapter 4, taking as reference a machine prototype related to the European project Re-FreeDrive. The proposed design procedure furnished notable performance and it has demonstrated to be as straightforward and quick as in accordance with the nowadays challenge to make the design process a multi-physics problem.

Bibliography

- [1] *FEMM Software Description*. <http://www.femm.info/wiki/HomePage>. Accessed: 2020-02.
- [2] *Matlab Software Description*. <https://uk.mathworks.com/products/matlab.html>. Accessed: 2020-02.
- [3] *MagNet Software Description*. <https://www.mentor.com/products/mechanical/magnet/magnet/features>. Accessed: 2020-02.
- [4] *AutoCAD Software Description*. <https://www.autodesk.co.uk/products/autocad/overview>. Accessed: 2020-02.
- [5] Motor Design Ltd. "Intro to MDL and Motor-CAD". In: *Training Course on the Design of BPM and Induction Motors: Oswestry, UK* (2019).
- [6] Motor Design Ltd. "Motor-CAD Manual v12". In: (2019).
- [7] Francesco Cupertino and Gianmario Pellegrino. "SyR-e user manual". In: (2019).
- [8] Nicola Bianchi. "The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors". In: Chapter 2 - 'Synchronous Reluctance and PM Assisted Reluctance Motors' (2016), pp. 27–35.
- [9] Simone Ferrari and Gianmario Pellegrino. "FEAfix: FEA Refinement of Design Equations for Synchronous Reluctance Machines". In: IEEE (2019).
- [10] A. Vagati, M. Pastorelli, G. Franceschini, et al. "Design of low-torque-ripple synchronous reluctance motors". In: *IEEE Trans. on Industry Application IA-34* (1998), pp. 758–765.
- [11] R. Moghaddam, F. Magnussen, and C. Sadarangani. "Novel rotor design optimization of synchronous reluctance machine for high torque density". In: *6th IET International Conference on Power Electronics, Machines and Drives (PEMD 2012)*, (2012), pp. 1–4.
- [12] R.-R. Moghaddam, F. Magnussen, and C. Sadarangani. "Novel rotor design optimization of synchronous reluctance machine for low torque ripple". In: *XXth International Conference on Electrical Machines (ICEM)* (2012), pp. 720–724.
- [13] Matteo Gamba. "Design of non conventional Synchronous Reluctance machine". In: Thesis at Politecnico di Torino Department of Energy (2014).
- [14] ReFreeDrive Consortium. "Project Overview". In: (2017).
- [15] *ReFreeDrive consortium Project Description*. <http://www.refreedrive.eu/>. Accessed: 2020-02.

-
- [16] Andrea Credo, Giuseppe Fabri, Marco Villani, et al. "High speed synchronous reluctance motors for electric vehicles: a focus on rotor mechanical design". In: 2019 IEEE International Electric Machines Drives Conference (IEMDC) (2019).
 - [17] Andrea Credo, Marco Villani, Mircea Popescu, et al. "Synchronous reluctance motors with asymmetric rotor shapes and epoxy resin for electric vehicles". In: 2019 IEEE Energy Conversion Congress and Exposition (ECCE) (2019).

A Code Listening

```
1 function draw_motor_in_MCAD(filename, pathname)
2
3 load([pathname filename])
4 addpath('./functions')
5 mcad=actxserver('MotorCAD.AppAutomation');
6
7 % stator parameters
8 slot=geo.parallel_slot;
9 if slot==0
10     invoke(mcad, 'SetVariable', 'SlotType', 0);    %slot type (ParallelTooth)
11 else
12     invoke(mcad, 'SetVariable', 'SlotType', 2);    %slot type (ParallelSlot)
13 end
14
15 Q=6*geo.p*geo.q;
16 tmp=num2str(Q);
17 tmp(tmp=='.')=',';
18 invoke(mcad, 'SetVariable', 'Slot_number', tmp);
19
20 tmp=2*dataSet.StatorOuterRadius;
21 tmp=num2str(tmp);
22 tmp(tmp=='.')=',';
23 invoke(mcad, 'SetVariable', 'Stator_Lam_Dia', tmp);
24
25 tmp=geo.l;
26 tmp=num2str(tmp);
27 tmp(tmp=='.')=',';
28 invoke(mcad, 'SetVariable', 'Stator_Lam_Length', tmp);
29 invoke(mcad, 'SetVariable', 'Rotor_Lam_Length', tmp);
30 invoke(mcad, 'SetVariable', 'Magnet_Length', tmp);
31
32 tmp=2.5*dataSet.StatorOuterRadius;
33 tmp=num2str(tmp);
34 tmp(tmp=='.')=',';
35 invoke(mcad, 'SetVariable', 'Housing_Dia', tmp);
36
37 tmp=2*(geo.r+geo.g);
38 tmp=num2str(tmp);
39 tmp(tmp=='.')=',';
40 invoke(mcad, 'SetVariable', 'Stator_Bore', tmp);
41
42 tmp=geo.g;
43 tmp=num2str(tmp);
44 tmp(tmp=='.')=',';
45 invoke(mcad, 'SetVariable', 'Airgap', tmp);
46
47
48 if slot==0
49     tmp=num2str(geo.wt);
```

```

50     tmp(tmp=='.'),'',';
51     invoke(mcad, 'SetVariable', 'Tooth_Width', tmp);           %ParallelTooth
52 else
53     tmp=(geo.r+geo.g+geo.lt/15)*sin(pi/geo.p/geo.Qs)-geo.wt;
54     tmp=num2str(tmp);
55     tmp(tmp=='.'),'',';
56     invoke(mcad, 'SetVariable', 'Slot_Width', tmp);           %ParallelSlot
57 end
58
59 tmp=geo.lt;
60 tmp=num2str(tmp);
61 tmp(tmp=='.'),'',';
62 invoke(mcad, 'SetVariable', 'Slot_Depth', tmp);
63 if slot==0
64     tmp=num2str(geo.SFR);
65     tmp(tmp=='.'),'',';
66     invoke(mcad, 'SetVariable', 'Slot_Corner_Radius', tmp); %ParallelTooth
67 end
68
69 tmp=num2str(geo.ttd);
70 tmp(tmp=='.'),'',';
71 invoke(mcad, 'SetVariable', 'Tooth_Tip_Depth', tmp);
72 tmp=(geo.acs)*((geo.r+geo.g)*2*pi/Q);
73 tmp=num2str(tmp);
74 tmp(tmp=='.'),'',';
75 invoke(mcad, 'SetVariable', 'Slot_Opening', tmp);
76
77 tmp=num2str(geo.tta);
78 tmp(tmp=='.'),'',';
79 invoke(mcad, 'SetVariable', 'Tooth_Tip_Angle', tmp);
80 invoke(mcad, 'SetVariable', 'Sleeve_Thickness', '0');
81
82 tmp=num2str(geo.Nbob);
83 tmp(tmp=='.'),'',';
84 invoke(mcad, 'SetVariable', 'MagTurnsConductor', tmp);
85
86 %Wedge
87 if geo.nmax>9999
88     invoke(mcad, 'SetVariable', 'Wedge_Model', '0'); %% with wedge
89 else
90     invoke(mcad, 'SetVariable', 'Wedge_Model', '1'); %% without wedge
91 end
92
93 % rotor parameters
94 tmp=geo.Ar*2;
95 tmp=num2str(tmp);
96 tmp(tmp=='.'),'',';
97 invoke(mcad, 'SetVariable', 'Shaft_Dia', tmp);           %rotor diameter
98 invoke(mcad, 'SetVariable', 'Pole_number', geo.p*2);
99
100 %Stator and Rotor angle
101 invoke(mcad, 'SetVariable', 'StatorRotation', 0);
102 invoke(mcad, 'SetVariable', 'RotorRotation', (90/geo.p));
103 invoke(mcad, 'SetVariable', 'BPM_Rotor', '13')
104
105 %Magnetic layers
106 invoke(mcad, 'SetVariable', 'Magnet_Layers', int2str(geo.nlay));
107
108 %Centre posts

```

```

109 tmp=fliplr(geo.pontR);
110 tmp=mat2str(tmp);
111 tmp=tmp(2:end-1);
112 tmp(tmp==' ')=:'';
113 invoke(mcad,'SetVariable','UShape_CentrePost_Array',tmp);
114
115 nlay=geo.nlay;
116 mg_leng=zeros(1,nlay);
117 mg_leng=mat2str(mg_leng);
118 mg_leng=mg_leng(2:end-1);
119 mg_leng(mg_leng==' ')=:'';
120 invoke(mcad,'SetVariable','UMagnet_Length_Inner_Array',mg_leng);
121 invoke(mcad,'SetVariable','UMagnet_Length_Outer_Array',mg_leng);
122
123 if geo.p==2;
124     tmp=(-geo.delta(2)).*ones(1,nlay);
125 else if geo.p==3;
126     tmp=(-geo.delta(2)*2).*ones(1,nlay);
127     else if geo.p==4;
128         tmp=(-geo.delta(2)*3.8).*ones(1,nlay);
129     else tmp=(-geo.delta(2)).*ones(1,nlay);
130     end
131 end
132 end
133
134 tmp=mat2str(tmp);
135 tmp=tmp(2:end-1);
136 tmp(tmp==' ')=:'';
137 invoke(mcad,'SetVariable','UShape_OuterAngleOffset_Array',tmp);
138
139 tmp=fliplr(geo.hc);
140 tmp=mat2str(tmp);
141 tmp=tmp(2:end-1);
142 tmp(tmp==' ')=:'';
143 invoke(mcad,'SetVariable','UShape_Thickness_Inner_Array',tmp);
144 invoke(mcad,'SetVariable','UShape_Thickness_Outer_Array',tmp);
145
146 hc=geo.hc(1);
147 tmp=0;
148 for i=1:1:nlay
149     tmp(i)=2*geo.B1k(nlay-i+1);
150 end
151 tmp=mat2str(tmp);
152 tmp=tmp(2:end-1);
153 tmp(tmp==' ')=:'';
154 invoke(mcad,'SetVariable','UShape_InnerDiameter_Array',tmp);
155
156 %%%
157 tmp=fliplr(geo.pontT);
158 tmp=mat2str(tmp);
159 tmp=tmp(2:end-1);
160 tmp(tmp==' ')=:'';
161 invoke(mcad,'SetVariable','UShape_BridgeThickness_Array',tmp);
162
163 if geo.p==2
164     tmp=sqrt(2)/2*geo.xxD1k-sqrt(2)/2*geo.yyD1k;
165 else
166     m=tan(pi/(2*geo.p));
167     tmp=(abs(geo.yyD1k-m*geo.xxD1k))/(sqrt(1+m^2));

```

```

168 end
169     tmp=2*fliplr(tmp);
170     tmp=mat2str(tmp);
171     tmp=tmp(2:end-1);
172     tmp(tmp==' ')=:'';
173     invoke(mcad, 'SetVariable', 'UShape_WebThickness_Array', tmp);
174     %%%%%%%%%
175     tmppont=geo.pontT;
176
177 %%%%%%%%%%% Materials
178 tmp=geo.BLKLABELS.materials(4);
179 tmp = convertCharsToStrings(tmp);
180 invoke(mcad, 'SetComponentMaterial', 'Stator Lam (Back Iron)', tmp);
181 invoke(mcad, 'SetComponentMaterial', 'Stator Lam (Tooth)', tmp);           %stator
182 invoke(mcad, 'SetComponentMaterial', 'Rotor Lam (Back Iron)', tmp);       %rotor
183 invoke(mcad, 'SetComponentMaterial', 'Shaft [Active]', tmp);              %shaft
184
185 %%%%%%%%%Lamination stacking factor
186 invoke(mcad, 'SetVariable', 'Stacking_Factor_[Stator]', 1);
187 invoke(mcad, 'SetVariable', 'Stacking_Factor_[Rotor]', 1);
188
189
190 file_mot=strrep(filename, '.mat', '.mot');
191 invoke(mcad, 'SaveToFile', [pathname file_mot]);
192
193 %export winding to MotorCAD
194 windingSyreToMCAD(mcad, pathname, filename, file_mot)
195
196 %create a proper .dxf for MotorCAD with 1 rotor pole
197 syreToDxfMCAD(pathname, filename)
198
199 %dxf MCAD settings
200 invoke(mcad, 'LoadDXFFile', [pathname filename(1:end-4), '.dxf']);
201 invoke(mcad, 'SetVariable', 'UseDXFImportForFEA_Magnetic', true);
202 invoke(mcad, 'SetVariable', 'UseDXFImportForFEA_Mechanical', true);
203 invoke(mcad, 'SetVariable', 'DXFImportType', 1);
204
205 %Save MCAD model
206 invoke(mcad, 'SaveToFile', [pathname file_mot]);
207 invoke(mcad, 'Quit');
208 save([pathname filename])
209
210 disp('Motor-CAD file saved in:')
211 disp([pathname file_mot])
212 disp(' ')
213 disp('Syr-e file saved in:')
214 disp([pathname filename])
215 disp(' ')
216 end

```

Code Listing 1: draw_motor_in_MCAD function

```
1 function syreToDxfMCAD(pathname,filename)
2
3 % syreToDxf.m - exports a fem model created by syre to dxf
4 % input: motorname.mat (created by syre along with motorname.fem)
5 % output: motorname.dxf, into the folder motorname
6
7 load([pathname filename]);
8
9 stator = geo.stator(1:17,:);
10 stator(:,1:6)=0;
11 rotor = geo.rotor;
12
13 % export to dxf
14 pathname_DXF=pathname;
15
16 if not(isfolder(pathname_DXF))
17     mkdir(pathname_DXF);
18 end
19
20 raggi=[];
21 avvolgimento=[];
22 magneti=[];
23
24 DXFconv(raggi,avvolgimento,rotor,stator,magneti,[pathname_DXF filename(1:end-4),'.dxf
    ']);
```

Code Listing 2: syreToDxfMCAD function

```

1 function windingSyreToMCAD(mcad,pathname,filename,file_mot)
2
3 load([pathname filename])
4
5 geo.avvtot=geo.avv;
6 cyclew=1;
7 for k=2:1:(geo.p*2)
8     if cyclew==1;
9         geo.avvtot=[geo.avvtot (-geo.avv)];
10        cyclew=0;
11    else geo.avvtot=[geo.avvtot geo.avv];
12        cyclew=1;
13    end
14 end
15
16 i=1; m=1; e=1; n=1; j=1; o=1; %counters
17 for k=1:1:(geo.Qs*2*geo.p)
18    value=geo.avvtot(1,k);
19
20    if value==1;
21        ph1go(i)=k; i=i+1;
22    end
23
24    if value==-1;
25        ph1ret(m)=k; m=m+1;
26    end
27
28    if value==2;
29        ph2go(n)=k; n=n+1;
30    end
31
32    if value==-2;
33        ph2ret(o)=k; o=o+1;
34    end
35
36    if value==3;
37        ph3go(j)=k; j=j+1;
38    end
39
40    if value==-3;
41        ph3ret(e)=k; e=e+1;
42    end
43
44 end
45
46 nbob=(e-1)*2;
47 geo.NbobInteger=round(geo.Nbob);
48
49 invoke(mcad,'LoadFromFile',[pathname file_mot]);
50
51 invoke(mcad,'SetVariable','MagWindingType', 1);
52 invoke(mcad,'SetVariable','MagPathType', 1);
53 invoke(mcad,'SetVariable','NumberOfCoils', nbob);
54 invoke(mcad,'SetVariable','Coil_Divider_Width',0);
55
56 a=1;
57 for i=0:2:(nbob-1)
58    invoke(mcad,'SetVariable',['Phase_1_Go1', num2str(i), ''], 0);
59    invoke(mcad,'SetVariable',['Phase_1_Go2', num2str(i), ''], ph1go(a));

```

```

60     invoke(mcad, 'SetVariable', ['Phase_1_Return1[' , num2str(i), '']], ph1ret(a));
61     invoke(mcad, 'SetVariable', ['Phase_1_Return2[' , num2str(i), '']], 0);
62     invoke(mcad, 'SetVariable', ['Phase_1_Turns[' , num2str(i), '']], geo.NbobInteger);
63
64
65     i=i+1;
66
67     invoke(mcad, 'SetVariable', ['Phase_1_Go1[' , num2str(i), '']], ph1go(a));
68     invoke(mcad, 'SetVariable', ['Phase_1_Go2[' , num2str(i), '']], 0);
69     invoke(mcad, 'SetVariable', ['Phase_1_Return1[' , num2str(i), '']], 0);
70     invoke(mcad, 'SetVariable', ['Phase_1_Return2[' , num2str(i), '']], ph1ret(a));
71     invoke(mcad, 'SetVariable', ['Phase_1_Turns[' , num2str(i), '']], geo.NbobInteger);
72
73     a=a+1;
74 end
75
76 a=1;
77 for i=0:2:(nbob-1)
78
79     invoke(mcad, 'SetVariable', ['Phase_2_Go1[' , num2str(i), '']], 0);
80     invoke(mcad, 'SetVariable', ['Phase_2_Go2[' , num2str(i), '']], ph2go(a));
81     invoke(mcad, 'SetVariable', ['Phase_2_Return1[' , num2str(i), '']], ph2ret(a));
82     invoke(mcad, 'SetVariable', ['Phase_2_Return2[' , num2str(i), '']], 0);
83     invoke(mcad, 'SetVariable', ['Phase_2_Turns[' , num2str(i), '']], geo.NbobInteger);
84
85     i=i+1;
86
87     invoke(mcad, 'SetVariable', ['Phase_2_Go1[' , num2str(i), '']], ph2go(a));
88     invoke(mcad, 'SetVariable', ['Phase_2_Go2[' , num2str(i), '']], 0);
89     invoke(mcad, 'SetVariable', ['Phase_2_Return1[' , num2str(i), '']], 0);
90     invoke(mcad, 'SetVariable', ['Phase_2_Return2[' , num2str(i), '']], ph2ret(a));
91     invoke(mcad, 'SetVariable', ['Phase_2_Turns[' , num2str(i), '']], geo.NbobInteger);
92
93     a=a+1;
94 end
95
96 a=1;
97 for i=0:2:(nbob-1)
98     invoke(mcad, 'SetVariable', ['Phase_3_Go1[' , num2str(i), '']], 0);
99     invoke(mcad, 'SetVariable', ['Phase_3_Go2[' , num2str(i), '']], ph3go(a));
100    invoke(mcad, 'SetVariable', ['Phase_3_Return1[' , num2str(i), '']], ph3ret(a));
101    invoke(mcad, 'SetVariable', ['Phase_3_Return2[' , num2str(i), '']], 0);
102    invoke(mcad, 'SetVariable', ['Phase_3_Turns[' , num2str(i), '']], geo.NbobInteger);
103
104    i=i+1;
105
106    invoke(mcad, 'SetVariable', ['Phase_3_Go1[' , num2str(i), '']], ph3go(a));
107    invoke(mcad, 'SetVariable', ['Phase_3_Go2[' , num2str(i), '']], 0);
108    invoke(mcad, 'SetVariable', ['Phase_3_Return1[' , num2str(i), '']], 0);
109    invoke(mcad, 'SetVariable', ['Phase_3_Return2[' , num2str(i), '']], ph3ret(a));
110    invoke(mcad, 'SetVariable', ['Phase_3_Turns[' , num2str(i), '']], geo.NbobInteger);
111    a=a+1;
112 end
113
114 end

```

Code Listing 3: windingSyreToMCAD function

```

1 function [cost,geo,mat,out,pathname]=MCADfitness (RQ,geo,per,mat,eval_type,filenameIn
   )
2
3 % [~,filename,ext] = fileparts(filenameIn);
4 [pathname,filename,ext] = fileparts(filenameIn);
5 filename = [filename ext]; % fem file name
6
7 pathname=[pathname '\'];
8
9
10 %load Syr-e and MCAD model
11 mcad=actxserver('MotorCAD.AppAutomation');
12 file_mot=[filename(1:(end-4)) '.mot'];
13 invoke(mcad,'LoadFromFile',[pathname file_mot]);
14
15 %MCAD mesh
16 invoke(mcad,'SetVariable','AirgapMeshPoints_layers',1440);
17 invoke(mcad,'SetVariable','AirgapMeshPoints_mesh',1440);
18
19 [SOL]=simulate_xdegMCAD(geo,per,mat,eval_type,pathname,filename);
20
21 %save outputs
22 out.id = mean(SOL.id); %const
23 out.iq = mean(SOL.iq); %const
24 out.fd = mean(SOL.fd); %waveform
25 out.fq = mean(SOL.fq); %waveform
26 out.T = mean(SOL.T); %waveform
27 out.dT = std(SOL.T);
28 out.dTpu = std(SOL.T)/out.T;
29 out.dTpp = max(SOL.T)-min(SOL.T);
30 out.IPF = SOL.IPF;
31 out.SOL = SOL;
32
33 %check Torque sign
34 if sign(out.T)~=sign(out.fd*out.iq-out.fq*out.id)
35     out.T = -out.T;
36     out.SOL.T = -out.SOL.T;
37 end
38
39 %save losses
40 out.Pfes_h=SOL.Pfes_h;
41 out.Pfes_c=SOL.Pfes_c;
42 out.Pfer_h=SOL.Pfer_h;
43 out.Pfer_c=SOL.Pfer_c;
44 out.Pfe_total=out.Pfes_h+out.Pfes_c+out.Pfer_h+out.Pfer_c;
45
46
47 %unused output
48 cost=0;
49
50 %save MCAD model
51 invoke(mcad,'SaveToFile',[pathname file_mot]);
52
53 %save MCAD results and quit
54 invoke(mcad,'SaveResults','EMagnetic');
55 invoke(mcad,'Quit');
56 end

```

Code Listing 4: MCADfitness function

```

1 function [SOL]=simulate_xdegMCAD(geo,per,mat,eval_type,pathname,filename)
2
3 filename=[filename(1:end-4) '.mat'];
4 load([pathname filename])
5 mcad=actxserver('MotorCAD.AppAutomation');
6 invoke(mcad,'SetVariable','PhaseAdvance',dataSet.GammaPP); % phase advance
7 io=calc_io(geo,per);
8 invoke(mcad,'SetVariable','PeakCurrent',dataSet.CurrLoPP*io); % peak current
9
10 invoke(mcad,'SetVariable','DCBusVoltage',800);
11
12
13 if dataSet.EvalSpeed~=0
14     invoke(mcad,'SetVariable','Shaft_Speed_Ref',dataSet.EvalSpeed);
15 else invoke(mcad,'SetVariable','Shaft_Speed_[RPM]',1000);
16     disp('simulation runs with a default value of 1000 rpm - No input speed from Syr-
17         e')
18 end
19
20 %Simulation settings
21
22 invoke(mcad,'SetVariable','BackEMFCalculation','False');
23 invoke(mcad,'SetVariable','CoggingTorqueCalculation','False');
24 invoke(mcad,'SetVariable','TorqueSpeedCalculation','False');
25 invoke(mcad,'SetVariable','DemagnetizationCalc','False');
26 invoke(mcad,'SetVariable','TorqueCalculation','True');
27 nPoints=dataSet.NumOfRotPosPP*6; %over 360 eltDeg
28 invoke(mcad,'SetVariable','TorquePointsPerCycle',int2str(nPoints));
29 magnetic_solver=1; %multi-static magnetic solver
30 invoke(mcad,'SetVariable','MagneticSolver',magnetic_solver); %multi-static magnetic
31     solver
32 invoke(mcad,'SetVariable','ArmatureEWdgMLT_Multiplier',0); %no end-windings effect
33 invoke(mcad,'SetVariable','MagThreads_Option',1); %multiple threads
34
35 % disp('Magnetic simulation in progress...')
36 success=invoke(mcad,'DoMagneticCalculation');
37 if success==0
38     disp('Magnetic calculation successfully completed')
39 else
40     disp('Magnetic calculation failed')
41 end
42
43 %save losses
44 if magnetic_solver==0
45 [tmp,Pfes_h_BackIron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Hys');
46 [tmp,Pfes_h_Tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Hys');
47 [tmp,Pfes_exc_backiron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Excess');
48 [tmp,Pfes_exc_tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Excess');
49 SOL.Pfes_h=Pfes_h_BackIron+Pfes_h_Tooth+Pfes_exc_backiron+Pfes_exc_tooth;
50
51 [tmp,Pfes_c_BackIron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Eddy');
52 [tmp,Pfes_c_Tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Eddy');
53 SOL.Pfes_c=Pfes_c_BackIron+Pfes_c_Tooth;
54
55 [tmp,Pfer_h_BackIron]=invoke(mcad,'GetVariable','RotorBackIronLoss_Hys');
56 [tmp,Pfer_h_Tooth]=invoke(mcad,'GetVariable','RotorMagnetPoleLoss_Hys');
57 [tmp,Pfer_exc_backiron]=invoke(mcad,'GetVariable','RotorBackIronLoss_Excess');
58 [tmp,Pfer_exc_tooth]=invoke(mcad,'GetVariable','RotorMagnetPoleLoss_Excess');

```

```

58 SOL.Pfer_h=Pfer_h_BackIron+Pfer_h_Tooth+Pfer_exc_tooth+Pfer_exc_backiron;
59
60 [tmp,Pfer_c_BackIron]=invoke(mcad,'GetVariable','RotorBackIronLoss_Eddy');
61 [tmp,Pfer_c_Tooth]=invoke(mcad,'GetVariable','RotorMagnetPoleLoss_Eddy');
62 SOL.Pfer_c=Pfer_c_BackIron+Pfer_c_Tooth;
63 end
64
65 if magnetic_solver==1
66 [tmp,Pfes_h_BackIron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Hys_Static');
67 [tmp,Pfes_h_Tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Hys_Static');
68 [tmp,Pfes_exc_backiron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Exc_Static');
69 [tmp,Pfes_exc_tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Exc_Static');
70 SOL.Pfes_h=Pfes_h_BackIron+Pfes_h_Tooth+Pfes_exc_backiron+Pfes_exc_tooth;
71
72 [tmp,Pfes_c_BackIron]=invoke(mcad,'GetVariable','StatorBackIronLoss_Eddy_Static');
73 [tmp,Pfes_c_Tooth]=invoke(mcad,'GetVariable','StatorToothLoss_Eddy_Static');
74 SOL.Pfes_c=Pfes_c_BackIron+Pfes_c_Tooth;
75 SOL.Pfer_c=0;
76 SOL.Pfer_h=0;
77 end
78
79 %save current dq (dq axis Syr-e)
80 [tmp,SOL.id]=invoke(mcad,'GetVariable','CurrentLoad_Q');
81 [tmp,SOL.iq]=invoke(mcad,'GetVariable','CurrentLoad_D');
82 SOL.iq=-sqrt(2)*SOL.iq;      SOL.id=sqrt(2)*SOL.id;
83
84 %save IPF
85 [tmp,SOL.IPF]=invoke(mcad,'GetVariable','WaveformPowerFactor');
86
87
88 %save Torque
89 RotorPosition = linspace(0,360,nPoints);
90 SLOT.T =zeros(nPoints,1);
91 for loop=1:nPoints
92     [success,x,y]=invoke(mcad,'GetMagneticGraphPoint','TorqueVW',loop);
93     if success == 0
94         RotorPosition(loop)=x;
95         SOL.T(loop)=y;
96     end
97 end
98
99 %save flux dq
100 for loop=1:nPoints
101     [success,x,y]=invoke(mcad,'GetMagneticGraphPoint','FluxLinkageLoadTotalD',loop);
102     if success == 0
103         RotorPosition(loop)=x;
104         SOL.fq(loop)=-y;
105     end
106 end
107 for loop1=1:nPoints
108     [success,x,y]=invoke(mcad,'GetMagneticGraphPoint','FluxLinkageLoadTotalQ',loop1);
109     if success == 0
110         RotorPosition(loop1)=x;
111         SOL.fd(loop1)=y;
112     end
113 end
114 end

```

Code Listing 5: simulate_xdegMCAD function

```
1 % Copyright 2019
2 %
3 % Licensed under the Apache License, Version 2.0 (the "License");
4 % you may not use this file except in compliance with the License.
5 % You may obtain a copy of the License at
6 %
7 % http://www.apache.org/licenses/LICENSE-2.0
8 %
9 % Unless required by applicable law or agreed to in writing, software
10 % distributed under the License is distributed on an "AS IS" BASIS,
11 % WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 % See the License for the specific language governing permissions and
13 % limitations under the License.
14
15 function eval_operatingPointMCAD_FEMM(dataIn)
16
17 % simulates single or multiple (id,iq) conditions
18 % example inputs:
19 % single condition: CurrLoPP = 1, GammaPP = 45
20 % multiple points: CurrLoPP = [1 1.5 2], gamma = [45 45 45]
21
22 % Uses matlabpool (parfor)
23
24 % Key INPUTs: CurrLoPP: current to be simulated
25 %             GammaPP: current phase angle
26 %             BrPP: remanence of all barriers magnets
27 %             NumOfRotPosPP: # simulated positions
28 %             AngularSpanPP: angular span of simulation
29 %=====
30
31 pathname=dataIn.currentpathname;
32 filemot = strep(dataIn.currentfilename, '.mat', '.fem');
33 load([pathname dataIn.currentfilename]);
34
35 CurrLoPP = dataIn.CurrLoPP;
36 GammaPP = dataIn.GammaPP;
37 BrPP = dataIn.BrPP;
38 NumOfRotPosPP = dataIn.NumOfRotPosPP;
39 AngularSpanPP = dataIn.AngularSpanPP;
40 NumGrid = dataIn.NumGrid;
41
42 % Iron Loss Input
43 if dataIn.LossEvaluationCheck == 1
44     per.EvalSpeed = dataIn.EvalSpeed;
45 end
46
47 clc;
48
49 eval_type = dataIn.EvalType;
50
51 per.overload=CurrLoPP;
52 per.BrPP=BrPP;
53
54 geo.nsim_singt = NumOfRotPosPP; % # simulated positions
55 geo.delta_sim_singt = AngularSpanPP; % angular span of simulation
56
57 iAmp = dataIn.SimulatedCurrent;
58
59 % single point or array of points simulation
```

```

60 performance = cell(1,length(CurrLoPP));
61 output = cell(1,length(CurrLoPP));
62 geometry = cell(1,length(CurrLoPP));
63 tempDirName = cell(1,length(CurrLoPP));
64 for ii = 1:length(CurrLoPP)
65     performance{ii} = per;
66     performance{ii}.overload = CurrLoPP(ii);
67     performance{ii}.gamma=GammaPP(ii);
68 end
69 geo.RemoveTMPfile = 'OFF';
70 % check parallel computing
71 ppState=parallelComputingCheck();
72 if (ppState==0 && length(CurrLoPP)>4)
73     parpool();
74     ppState=parallelComputingCheck();
75 end
76
77 fileMotWithPath=[pathname filemot];
78
79 geo0=geo;
80 mat0=mat;
81 % evaluation
82 if dataIn.MCADFEMM==1
83     if ppState<1
84         for ii = 1:length(CurrLoPP)
85             geoTmp = geo0;
86             perTmp = performance{ii};
87             matTmp = mat0;
88             [~,geometry{ii},~,output{ii},tempDirName{ii}] = FEMMfitness([],geoTmp,perTmp,
            matTmp,eval_type,fileMotWithPath);
89         end
90     else
91         parfor ii = 1:length(CurrLoPP) %%
92             geoTmp = geo0;
93             perTmp = performance{ii};
94             matTmp = mat0;
95             [~,geometry{ii},~,output{ii},tempDirName{ii}] = FEMMfitness([],geoTmp,perTmp,
            matTmp,eval_type,fileMotWithPath);
96         end
97     end
98 end
99
100 if dataIn.MCADFEMM==0
101     if ppState<1
102         for ii = 1:length(CurrLoPP)
103             geoTmp = geo0;
104             perTmp = performance{ii};
105             matTmp = mat0;
106             [~,geometry{ii},~,output{ii},tempDirName{ii}] = MCADfitness([],geoTmp,perTmp,
            matTmp,eval_type,fileMotWithPath);
107         end
108     else
109         parfor ii = 1:length(CurrLoPP) %%
110             geoTmp = geo0;
111             perTmp = performance{ii};
112             matTmp = mat0;
113             [~,geometry{ii},~,output{ii},tempDirName{ii}] = MCADfitness([],geoTmp,perTmp,
            matTmp,eval_type,fileMotWithPath);
114         end

```

```

115 end
116 end
117
118 % save output into individual folders
119 for ii = 1:length(CurrLoPP)
120
121     geo = geometry{ii};
122     out = output{ii};
123     per = performance{ii};
124     dirName = tempDirName{ii};
125
126     iStr=num2str(iAmp(ii),3); iStr = strrep(iStr, '.', 'A');
127     gammaStr=num2str(GammaPP(ii),4); gammaStr = strrep(gammaStr, '.', 'd');
128     if ~contains(gammaStr, 'd')
129         gammaStr = [gammaStr 'd'];
130     end
131
132     FILENAME = [filemot(1:end-4) '_T_eval_', iStr, '_', gammaStr];
133     switch eval_type
134         case 'flxdn'
135             FILENAME = [FILENAME '_flxdn'];
136         case 'izero'
137             FILENAME = [FILENAME '_izero'];
138         case 'force'
139             FILENAME = [FILENAME '_force'];
140     end
141     mkdir(pathname, FILENAME);
142     newDir=[pathname, FILENAME, '\'];
143
144     if isoctave() %OCT
145         file_name1= strcat(newDir, FILENAME, '.mat');
146         save('-mat7-binary', file_name1, 'geo', 'per', 'out');
147         dirIn=strcat(dirName, ['\ ' filemot]);
148         dirDest=strcat(newDir, FILENAME, '.fem');
149         movefile(dirIn, dirDest);
150         clear file_name1 dirIn dirDest
151     else
152         save([newDir, FILENAME, '.mat'], 'geo', 'per', 'out');
153     %     copyfile([dirName filemot], [newDir FILENAME '.fem']);
154     end
155
156     % plot and save figs
157     klength = 1; kturns = 1; delta_sim_singt = geo.delta_sim_singt;
158
159     if dataIn.MCADFEMM==1
160         plot_singt(out, klength, kturns, delta_sim_singt, newDir, filemot);
161     end
162
163     if dataIn.MCADFEMM==0
164         plot_singtMCAD(out, klength, kturns, delta_sim_singt, newDir, filemot);
165     end
166
167     switch eval_type
168         case 'flxdn'
169         plot_flxdn_fig(geo, out, newDir, filemot);
170         plot_flxdn_gif(geo, out, newDir, filemot);
171         case 'force'
172         plot_force_fig(geo, out, newDir, filemot);
173         plot_force_gif(geo, out, newDir, filemot);

```

```

174     end
175
176 end
177
178 % extra figs, if input current is array
179 if length(CurrLoPP)>1
180
181     id = zeros(1,length(CurrLoPP));
182     iq = zeros(1,length(CurrLoPP));
183     T = zeros(1,length(CurrLoPP));
184     dTpu = zeros(1,length(CurrLoPP));
185     dTpp = zeros(1,length(CurrLoPP));
186     fd = zeros(1,length(CurrLoPP));
187     fq = zeros(1,length(CurrLoPP));
188
189     for ii = 1:length(CurrLoPP)
190         id(ii) = output{ii}.id;
191         iq(ii) = output{ii}.iq;
192         T(ii) = output{ii}.T;
193         dTpu(ii) = output{ii}.dTpu;
194         dTpp(ii) = output{ii}.dTpp;
195         fd(ii) = output{ii}.fd;
196         fq(ii) = output{ii}.fq;
197     end
198     dirPower=[pathname,filemot(1:end-4),'_singT'];
199     mkdir(dirPower);
200
201     x = 1:length(CurrLoPP);
202     figure();
203     if ~isoctave()
204         figSetting();
205     end
206     subplot(2,1,1)
207     plot(x,T,'-x',x,T+0.5*dTpp,'r',x,T-0.5*dTpp,'r'), grid on, ylabel('$T$ [Nm]')
208     subplot(2,1,2)
209     plot(x,dTpp,'-x'), grid on, ylabel('$\Delta T_{pp}$ [Nm]')
210     xlabel('simulation \#')
211     h=gcf();
212     if isoctave() %OCT
213         fig_name=strcat(dirPower, filemot(1:end-4), '_torque_sens');
214         hgsave(h,[fig_name]);
215     else
216         saveas(gcf,[dirPower,filemot(1:end-4),'_torque_sens.fig'])
217     end
218
219     figure()
220     if ~isoctave()
221         figSetting();
222     end
223     subplot(2,1,1)
224     plot(x,fd,'-x',x,fq,'-x'), grid on, ylabel('[Vs]'), legend('$\lambda_d$', '$\lambda_q$'),
225     subplot(2,1,2)
226     plot(x,abs(sin(atan(iq./id)-atan(fq./fd))),'-x'), grid on, ylabel('$\cos \varphi$')
227     xlabel('simulation \#'),
228     h=gcf();
229     if isoctave() %OCT
230         fig_name=strcat(dirPower, filemot(1:end-4), '_fdq_IPF_sens');

```

```
231     hgsave(h,[fig_name]);
232 else
233     saveas(gcf,[dirPower,filemot(1:end-4),'_fdq_IPF_sens.fig'])
234 end
235
236 figure()
237 if ~isoctave()
238     figSetting();
239 end
240 subplot(2,1,1)
241 plot(x,fd,'-x','DisplayName',' $\lambda_d$ ');
242 plot(x,fq,'-x','DisplayName',' $\lambda_q$ ');
243 ylabel('[Vs]')
244 legend('show');
245 subplot(2,1,2)
246 plot(x,id,'-x','DisplayName',' $i_d$ ');
247 plot(x,iq,'-x','DisplayName',' $i_q$ ');
248 xlabel('simulation \#')
249 ylabel('[A]')
250 legend('show');
251 h=gcf();
252 if isoctave() %OCT
253     fig_name=strcat(dirPower, filemot(1:end-4), '_fdq_idiq_sens');
254     hgsave(h,[fig_name]);
255 else
256     saveas(gcf,[dirPower,filemot(1:end-4),'_fdq_idiq_sens.fig'])
257 end
258 senseOut.id = id;
259 senseOut.iq = iq;
260 senseOut.fd = fd;
261 senseOut.fq = fq;
262 senseOut.T = T;
263 senseOut.dTpp = dTpp;
264 senseOut.PF = abs(sin(atan(iq./id)-atan(fq./fd)));
265 save([dirPower,filemot(1:end-4),'_senseResults.mat'],'senseOut');
266
267 end
```

Code Listing 6: simulate_xdegMCAD function