

VI International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2015
B. Schrefler, E. Oñate and M. Papadrakakis (Eds)

MULTIPHYSICS SIMULATION TOOLS FOR DESIGNING MOTORS FOR TRACTION APPLICATIONS IN HYBRID AND ELECTRIC VEHICLES

NIKHIL P. SAWANT^{*}, CHRISTOPHER P. RILEY^{*}, ARTURAS VENSKUS^{*}, DANAIL
H. VASSILEV^{*}, JOHN D. WALE[†], EDDIE WEARING[†], ALEX M. MICHAELIDES[#],
PHILIP J. TOPPING[#], HERMINDER MATHARU[#]

^{*}Cobham Technical Services,
Network House, Langford Locks, Kidlington, OX5 1LH, UK
e-mail: vectorfields.info@cobham.com

[†]Ricardo UK Ltd
Midlands Technical Centre
Southam Road, Radford Semele, Leamington Spa, CV31 1FQ, UK
e-mail: electric.drives@ricardo.com

[#]Jaguar Land Rover Limited
Abbey Road, Whitley, Coventry CV3 4LF, UK
e-mail: hmathar2@jaguarlandrover.com

Key words: Multiphysics, motors, electric vehicle, Opera, python, simulation.

Abstract. Motor manufacturers are facing a difficult challenge in designing traction motors for the latest generation of hybrid and all-electric vehicles. The efficiency with which these motors can perform is critical, as it impacts on the vehicle range and battery life. Many of the issues involved in the motor design have a complex nature which requires multiple fields of physics such as electromagnetics (EM), mechanics and thermal analysis. All these physics are usually interdependent and have to be considered collectively in order to obtain optimal performance for a particular scenario.

This paper presents a multiphysics simulation tool that was implemented to address this situation. The Opera FEA software suite [1] was developed to include a multiphysics analysis that can link several EM, thermal and stress analyses. Opera's Machines Environment (parameterised template software for designing motors and generators) has been extended to allow easy setup of coupled multiphysics analyses such as EM to thermal and EM to stress. In order to further facilitate the coupling of different analyses, a link to the Python programming language was embedded in Opera FEA software. The embedded Python facility offers options to perform certain post-processing operations during the solving stage and hence allow data transfer between different stages of the multiphysics analysis. It also extends Opera's capabilities to interact with other FEA software.

1 INTRODUCTION

With growing concerns over future petroleum supplies and air pollution, sustainable transportation has emerged as a vital mission for countries that seek to expand infrastructure

whilst minimising the associated economic and environmental costs. Hybrid electric vehicles (HEVs) and electric vehicles (EVs) represent one of the most promising routes to better energy security and reduced emissions of greenhouse gases. The research in HEVs/EVs has been widely supported by governments worldwide. The U.S. Department of Energy predicts that by 2030, alternative vehicles will comprise 28% of the total U.S. light-duty cars and trucks—a 20% increase from 2005 [2]. Consequently, there is a strong competition to develop efficient and cost-effective electric powertrains in order to meet these demands by focussing on simulation based development rather than prototype testing.

The motor/generator plays a crucial role in driving the vehicle. Traction motors for EVs are different from other motors because they must work reliably in a severe physical environment. Motors must operate consistently under extreme hot and cold temperatures, hard duty cycles and rough road conditions. All of these variables must be addressed in motor design.

Electrical machine designers have to consider the following fields of physics to obtain optimal behaviour of a particular machine:

- Electromagnetics (EM)
- Mechanics (stress)
- Thermal characteristics

All these physical phenomena usually have an effect on one another and have to be considered collectively. So, machine designers have to consider a combination of two or more physics problems while optimizing the overall design. Also, the focus is on getting very accurate results in less time.

In this paper, various tools implemented in Opera for characterizing machine performance within a multiphysics scenario are discussed. The paper then presents the coupling options implemented in Opera's Machines Environment. All the developments discussed in this paper were developed as part of a collaborative development programme named the 'Rapid SR' project [3] between Cobham Technical Services, Jaguar Land Rover and Ricardo, part supported by Innovate UK (previously known as the Technology Strategy Board).

2 MULTIPHYSICS ANALYSIS

In order to evaluate accurate performance of electric motors for traction application, it is necessary to assess the interaction between the various physics arising from the natural phenomena. For example, material properties are usually temperature dependent. A multiphysics analysis tool to address this situation has been implemented that can chain EM, thermal and/or stress analyses. Results from one analysis are automatically transferred as input to any subsequent analysis in the chain. Table 1 shows the exchange of data between different analyses.

For machines where the air gaps are small, it is important to include the structural changes evaluated by the stress analysis to characterize the machine performance. This can be achieved by chaining magnetostatic, stress and magnetostatic solutions, as shown in figure 1. Element force densities evaluated by the Maxwell stress in the magnetostatic analysis are supplied to the stress analysis, along with the rotational forces on the rotor. The last stage of the chain uses the displaced nodes to re-evaluate the electromagnetic behaviour of the machine. This loop can be continued iteratively until convergence is obtained.

Table 1: Data exchange between different physics [4].

Inputs used	Fields Calculated		
	EM	Thermal	Stress
EM		Temperature	Displacements
Thermal	Heat density from coils, and/or eddy currents		Displacements
Stress	Magnetic flux density, Element force densities	Temperature	

For a multi-case analysis, where the electromagnetic simulation consists of different excitations (for example, different currents or frequencies) then the chain is repeated for all the cases. For example, for a solution at two current excitations chained with stress and a second EM, the multiphysics analysis will run a chain of EM, stress and EM for each excitation (six simulations in total).

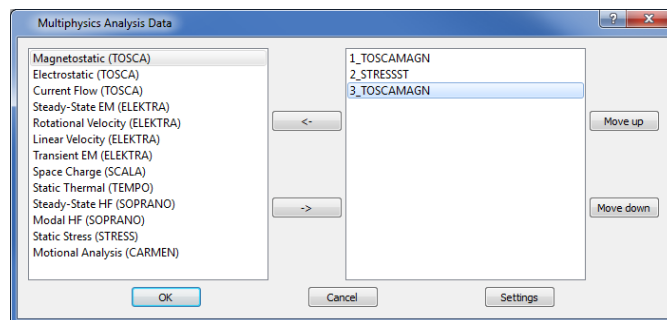


Figure 1: Multiphysics setup showing chained analyses

For a transient analysis, where the electromagnetic simulation is a time-stepping solution, results may be averaged over a cycle and passed to the next analysis in the chain. For example, a transient electromagnetic simulation, which is set up to run for one cycle after reaching steady-state, is chained with a steady state thermal simulation. The program will automatically average the results (electromagnetic losses) from the EM solution over the one cycle and pass that average on to the thermal simulation to evaluate temperature rise.

3 PYTHON

A multiphysics analysis where different analyses are chained involves evaluating certain quantities from a particular solution, which are passed on to the next analysis as input. In some scenarios, this involves accessing the simulation while it is still running. In order to address this, Python (programming language) was embedded in Opera. Python’s functionality such as advanced string manipulation and numerical analysis components were the key features for it being a viable solution for addressing the requirements of the multiphysics analysis. The integration between Python and Opera has also enabled easy interaction of Opera with other 3rd party scientific programs. The following sections give an overview of how the Opera’s functionality was extended by using Python:

3.1 Opera Database Extraction

The Opera Database Extraction package contains functionality to extract a data set of results from a particular simulation in a sequence (for example, one case is a transient simulation), and to subsequently store a subset of this data in an easily referable manner. The primary purpose of this is to allow user computations to be performed during the solving stage, although it may equally be used in a post-processing stage. For example, it may be used to extract x, y and z components of flux density, and store only the subset of this data relevant to ferrous regions (rotor or stator steel) throughout a transient simulation. Such data may then be used once the transient period has completed to evaluate iron loss.

3.2 Opera Steady State Detection

The package Opera Steady State Detection provides functionality to determine steady state in transient simulations (i.e. the time at which the behaviour of the system becomes time periodic). It includes a sequential analysis sub-package, which provides two detection methods that can be applied to global simulation data such as torque, speed, current etc. Sequential analysis solves hypothesis-testing problems in which the sample size is not fixed a priori but depends upon the data that have been already observed. While rudiments of sequential analysis date back to the works of Huyghens, Bernoulli, DeMoivre, Lagrange and Laplace as briefly noted in [7], the theory of sequential analysis was formulated by A. Wald in his celebrated work [8]. Over the recent years sequential analysis has become an important tool for handling real world problems arising across various branches of science and engineering: environment surveillance and monitoring, navigation and radar signal processing, fault detection, quality control, biomedical signal processing, econometrics and financial markets. The common feature of the above problems is the goal to detect one or several abrupt changes in some characteristic properties of the considered object.

In the first method, called a "variance test" following [9] parameter estimation is performed on collected simulation data. Once the required criteria for steady state detection have been satisfied, further sampling is disabled. The stopping criterion is based on the minimization of a cost function that measures the data heteroscedasticity, the tendency of the standard deviation of the observed quantity to vary.

The second sequential analysis detection method, referred to as the "mean test", is a version of CUSUM algorithm [10] specially designed to detect the transient time at which the mean of the data settles down to zero. It may be used, for example, to analyse the DC offset in a transient simulation. A log-likelihood function for the samples is computed and compared to a threshold value to decide whether the parameters of the underlying probability distribution represent steady or transient state. In transient electromagnetic simulations, the data (e.g. current, voltage) is periodic and the assumption is that the samples follow an arcsine distribution, in which the amplitude and the offset are the control parameters. The nominal values of these parameters define the steady state and need to be specified along with the threshold for the log-likelihood function.

3.3 Opera Loss Calculation

The package Opera Loss Calculation provides functionality to calculate losses over a steady state cycle and to subsequently upload such losses to an Opera database. The package

evaluates losses based on current density (J) and material conductivity (σ) and iron losses. The iron losses are based on the Steinmetz formulae and use the following methodology:

- A full harmonic analysis is performed on the flux density waveform on an element-by-element basis, to an order which is user specified. The number of sampling steps is also user-specified, and should be a function of the highest harmonic order chosen by the user, ensuring that the stored flux density variation with angle (time) contains enough information for correct harmonic amplitude extraction.
- Element loss intensity tables are created which are then automatically transferred to the thermal model.

3.3 Coupling to other software

The close integration between Python and Opera has enabled easy extension of Opera’s functionality with 3rd party Python packages and other FEA systems. In order to extend the functionality, any new Python package needs to be installed preferably (but not necessarily) into the Opera Python distribution location and the new functionality can be accessible immediately via a Python “import package_name” statement. In such a manner, a package like *matplotlib* can provide functionality to perform 3D plots where the data can be supplied dynamically, for example during solver solution.

More advanced Python usage can be, for example, chaining Opera computations with other FEA solvers to perform acoustic, fatigue, fracture, dynamic vibration and other required analysis for electrical machines. The open source structural mechanics solver from EDF Code_Aster [6] can be used for this purpose. The following Python script snippet, shown in figure 2, demonstrates how to prepare FE data to Code_Aster using its native data exchange format MED (HDF based) library.

```

# create index class to manage elements of particular geometry
group
geom_label = "STEEL"
multiIndex = odi.opera_multi_index_by_element_class([geom_label])
multiIndex.generateIndices(simu)
index_type = "NODAL"
nodal_indices = multiIndex.getIndex(geom_label, index_type)
# set mesh spatial dimensionality, name and node count
mdim = 3;
mname = "mesh1";
mncount = nodal_indices.size;
# create numpy array to accommodate nodal coordinates
coord = np.empty(nodal_indices.size*mdim)
for i in range(nodal_indices.size):
    loc_index = i*3
    node_index = int(nodal_indices[i])
    node_coord = simu.getCoordForIndex(index_type, node_index)
    coord[loc_index] = node_coord[0] # x
    coord[loc_index+1] = node_coord[1] # y
    coord[loc_index+2] = node_coord[2] # z
# convert to MED float data type format
coo = MEDFLOAT(coord)
# create table name and unit headers
namecoo = "x          y          z          "
unicoo = "m          m          m          "
# create and open MED (HDF) file
fid = MEDfileOpen("plate.med", MED_ACC_CREAT)
# create mesh
MEDmeshCr( fid, mname, mdim, mdim, MED_UNSTRUCTURED_MESH,
           "mesh from Opera", "s", MED_SORT_DTIT,
           MED_CARTESIAN, namecoo, uniCoo)
# write nodal coordinates
MEDmeshNodeCoordinateWr( fid, mname, MED_NO_DT, MED_NO_IT, MED_UNDEF_DT,
                        MED_FULL_INTERLACE, nodal_indices.size, coo)
# close MED file
MEDfileClose(fid)
    
```

Figure 2: Code snippet for transferring mesh data.

The Python script above demonstrates only how to export FE nodes into MED file format. Similarly other FE mesh data (nodal connectivity, faces, and edges) can be exported using Python interfaces from the MED library. The same interface is used to extract field values back from an MED file after a Code_Aster analysis is performed.

4 MACHINE DESIGN ENVIRONMENT

The machines environment is a design package, which capitalizes on the accuracy of Finite Element Analysis of the Opera Suite to offer various options to the machine designer, including standard analysis options required for the design process. The machines environment was extended to exploit the developments in Opera to provide standard multiphysics scenarios for designing motors for traction applications. Two such scenarios will be described in detail using a practical example of a switched reluctance motor, as shown in figure 3. Table 2 shows the specifications of the SRM considered.

Table 2 Specification of SRM

Parameters	Values (cm)
Stator outer radius	3.9
Rotor outer radius	2.165
Air gap	1.735
Stack length	8.16
Number of turns per phase	35
Stator slots	8
Rotor poles	6

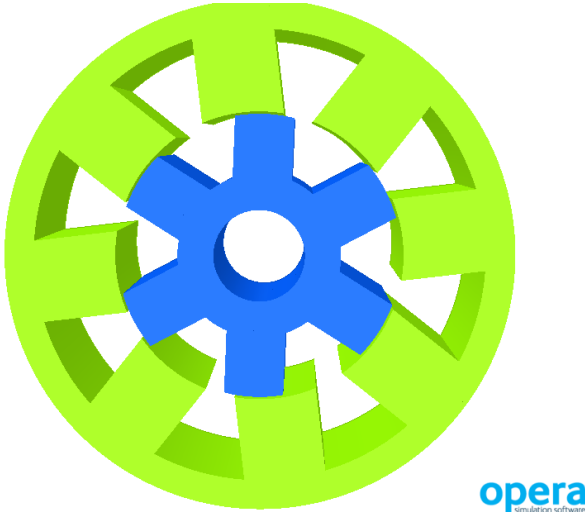


Figure 3: SRM geometry modelled in Opera

4.1 Electromagnetic and Stress Analysis

An analysis required to evaluate how induced stress in various parts of the machine affects its electromagnetic performance was implemented by coupling the electromagnetic and stress analyses. This analysis also includes stress induced by centrifugal loading. The electromagnetic analysis creates a series of static jobs for different rotor positions. The performance of the machine is characterized by evaluating peak torque obtained for the given excitation. For each rotor position, a stress stage was added to evaluate the stress due to electromagnetic and centrifugal loadings.

Convergence criteria with an iteration limit can be set by the user to decide whether further additional EM and stress stages should be added or not. Following are the two criteria which will be observed:

- *Absolute change on the deformation*: For each of the rotor positions, the maximum displacement as a percentage of the air gap radius on the rotor faces is recorded. If this value, for any of the rotor positions, is greater than that specified by the user then the EM and stress stages are repeated up to the iteration limit.
- *Relative change between the iterations*: The change in the percentage displacement between the iterations is also recorded. If this value is greater than that specified by the user then the EM and stress stages are repeated up to the iteration limit.

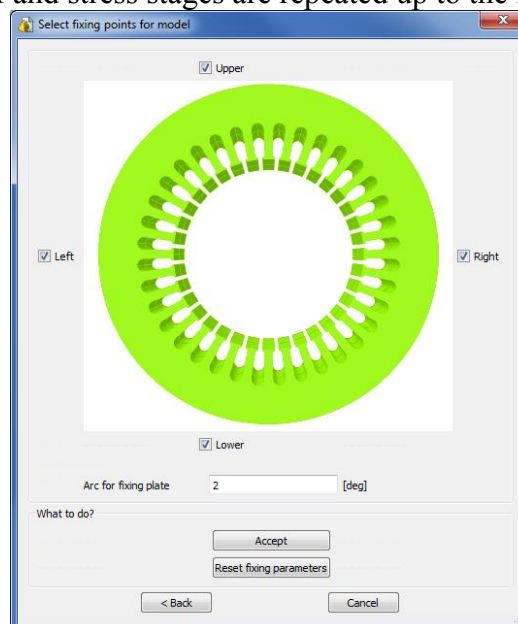


Figure 4: Option to select fixing point for the stator.

In order to accurately evaluate stress in the machine, appropriate boundary conditions need to be specified. Figure 4 shows the standard options that users can choose for the stator. For the rotor, the fixing points will be represented by the extremities of the shaft where a fixed displacement boundary condition will be applied. The length of the shaft can be set independently as compared to the length of the rotor to allow for a more realistic representation of the stresses produced in the device.

Figure 5 shows the electromagnetic characteristics (torque vs. rotor position) of the SRM at the end of simulation after accounting for the stress effects.

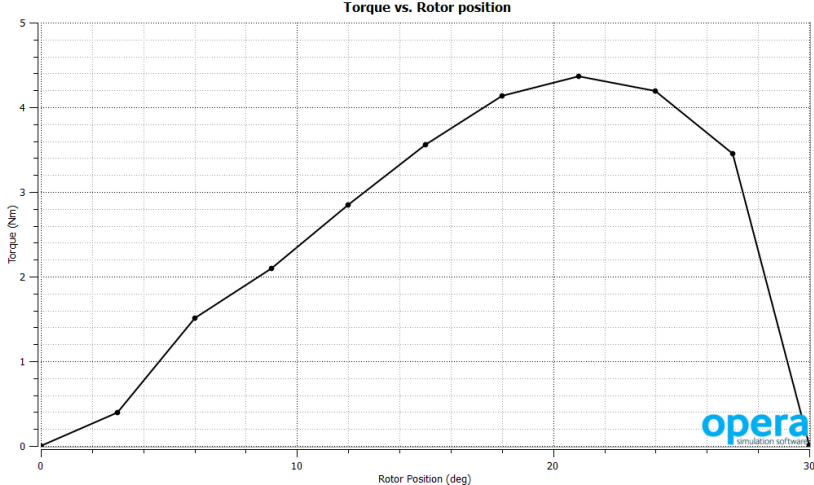


Figure 5: Torque vs rotor position.

Figure 6 shows an exaggerated view of the displacements due to the stress effects, where the winding displayed on the stator tooth is to show which of the phases was excited. The coloured plot shows that the displacement due to centrifugal loading is dominant and hence an even distribution across all the rotor teeth. Figure 7 shows the effect of stress on the air gap of the machine. For both the figures, the mesh shows the actual position of the rotor and the solid view shows the deformed position.

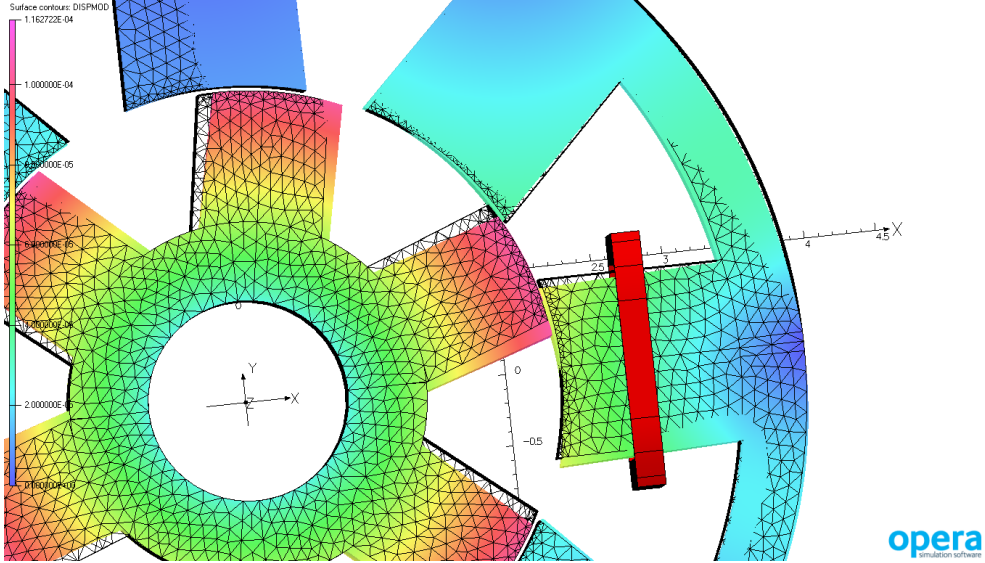


Figure 6: Displacements due to stress effects.

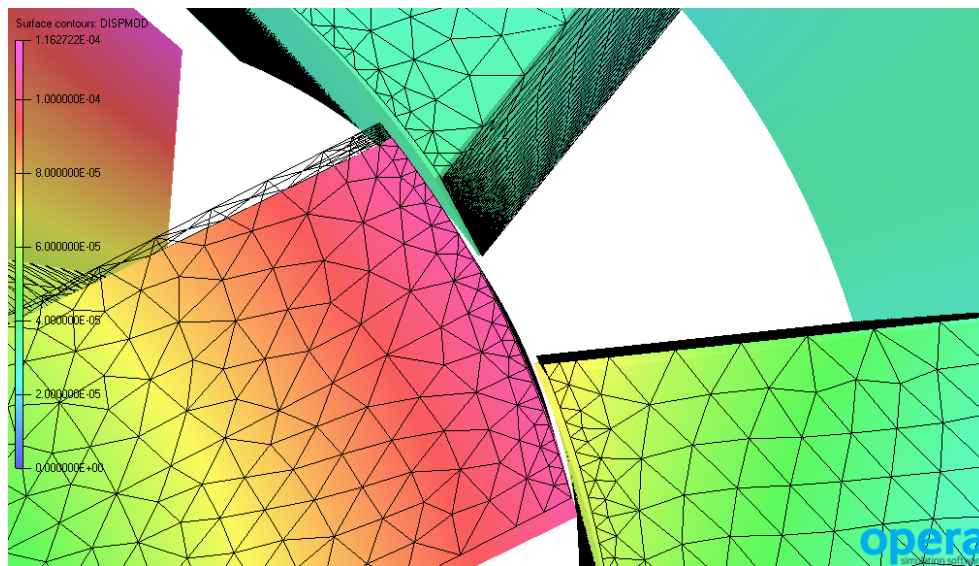


Figure 7: Gap closure due to stress effects

4.2 Electromagnetic and Thermal Analysis

An analysis required to evaluate how temperature rise in a machine affects its electromagnetic performance was implemented by coupling the electromagnetic and thermal analyses. A transient mechanically coupled analysis with rotational motion is run for a machine with a user defined excitation. A default switching strategy is calculated based on the geometric parameters and the number of phases selected, but a custom strategy can also be defined. A hysteretic speed control is also implemented in order to maintain the speed at the defined level. After the transient stage, a static thermal stage is added where the losses calculated over the last cycle of the transient stage are used as input.

This analysis uses the steady state detection package developed using Python in Opera in order to determine the start of the steady state. The analysis is run for one cycle after the steady state is detected. The loss calculation package of Python in Opera is used to perform the calculations over the one steady state cycle and pass the averaged losses over the cycle to the thermal stage.

For the thermal stage, the Stator and Rotor core thermal conductivities can be defined as anisotropic having different coefficients in the in-plane (radial/azimuthal) and axial directions. The conductor thermal conductivity is homogenized as a combination of conductivities for the conducting material and insulation and accounts for the fill factor. The boundary conditions in the form of heat transfer coefficients for the core back and air gap are set along with an ambient temperature.

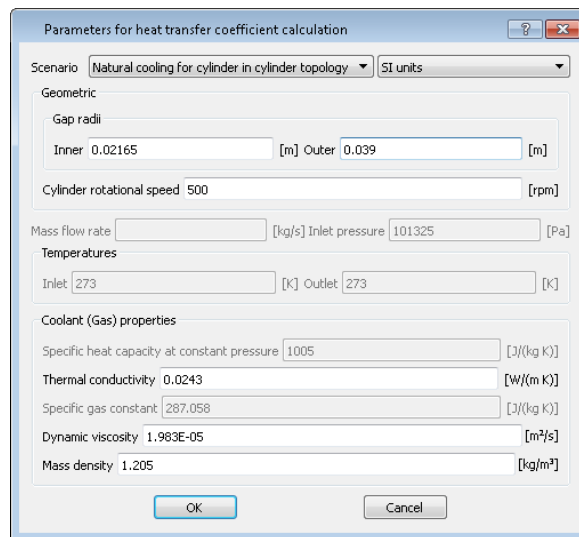


Figure 8: Tool for calculating heat transfer coefficients.

The heat transfer coefficient for the air gap of the machine is evaluated using the Heat Transfer Tool in Opera, developed in collaboration with City University, London [5], the interface to which is shown in figure 8. It calculates an estimate of the heat transfer coefficient for a cylindrical surface of a machine rotor, rotating in a larger cylindrical space, i.e. a rotor within a stator. The formulae used for the calculation have been collated from theory and experiment over different regimes of dimensions, rotation speeds and coolant properties. While this simple theory may only provide limited accuracy, the coefficients given provide good initial estimates and avoid the need for expensive computational fluid dynamics simulation before thermal analysis can proceed.

Figure 9 shows the torque profile of the SRM obtained by running the analysis while maintaining the speed around 500 rpm, as shown in figure 10. Figures 11 and 12 show the temperature rise in the stator and rotor sections respectively. This analysis exploited Opera's functionality of setting a 2d slice [4] of full 3d models for rapid simulation, but, consequently, neglects axial heat flow.

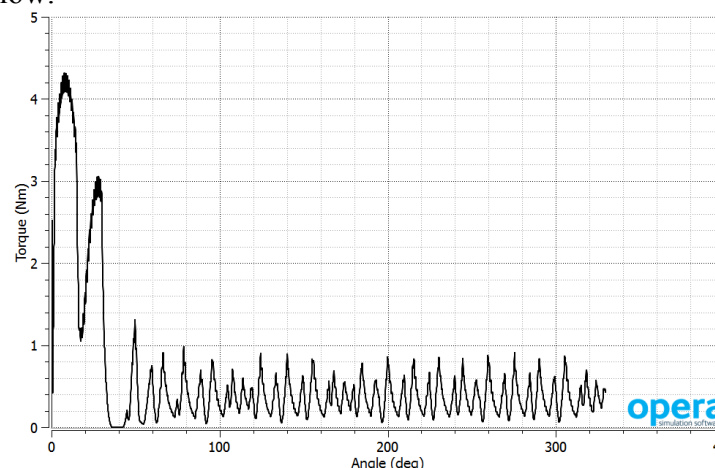


Figure 9: Torque vs. rotor position obtained from CarmenRM analysis

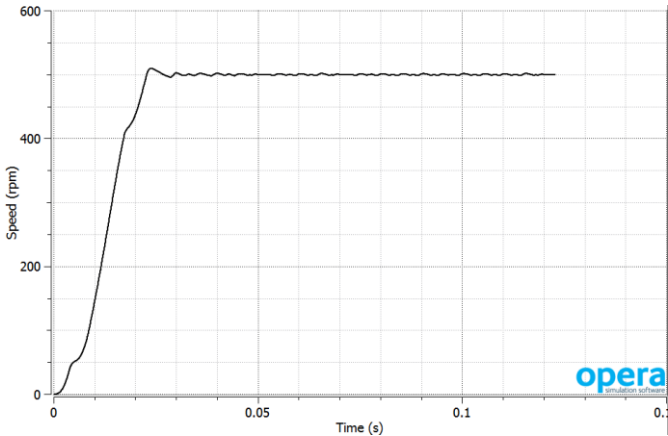


Figure 10: Speed vs time showing the hysteresis control of speed around 500 rpm

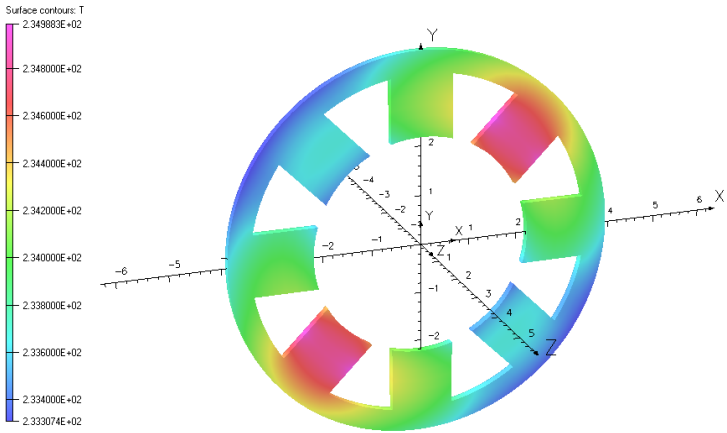


Figure 11: Temperature rise in the stator section

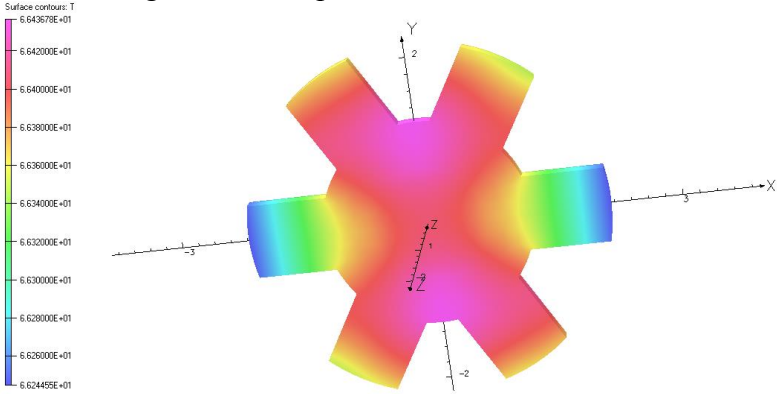


Figure 12: Temperature rise in the rotor section

5 CONCLUSIONS

The impetus for understanding the behavior of electrical machines has increased, following the increased on-going research in HEV/EV for a sustainable future. An in-depth knowledge of the interactions between electromagnetics, thermal and mechanical behavior of machines has become vital in order to understand the material performance, reliability and electromagnetic characteristics in extreme environments and at high speeds. This has driven the rise in interest in multi-physics simulations.

Software tools for characterizing machine performance based on various physics were discussed. Also, tools developed in the machines environment for coupling various fields of physics in order to get the optimized designs were discussed. The tools developed were successfully applied to an 8/6 SRM in order to understand and quantify its advantages. Further work as a part of the Rapid SR project will be carried out in order to verify the simulation results against practical measurements.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of Innovate UK in this work.

REFERENCES

- [1] Cobham Technical Services, “Opera Electromagnetics Finite Element Software Suite”, www.operafea.com
- [2] Annual Energy Outlook 2007 with Projections to 2030, report DOE/EIA.
- [3] Technology Strategy Board Collaborative R&D Project 400233 <http://motor-design-software.com/news/research-programme-targets-nextgeneration-electric-motors-luxury-automobiles>
- [4] C P Riley, et al., “Simulation based design of reluctance motors for traction applications in hybrid and electric vehicles”, PEMD, Manchester 2014
- [5] K.R. Pullen and B. Yoheswaran, Private Communication.
- [6] EDF, “Code_Aster”, <http://www.code-aster.org>
- [7] B.K. Ghosh, P.K. Sen, CRC Press, “Handbook of Sequential Analysis” , 1991
- [8] A. Wald, J. Wiley & sons, inc., 1947, “Sequential analysis”
- [9] Killick, R., Fearnhead, P., Eckley, I.A., “Optimal detection of changepoints with a linear computational cost”, Journal of the American Statistical Association 107(500) 1590-1598, 2012
- [10] P. Granjon, “The CUSUM algorithm - a small review”, <https://hal.archives-ouvertes.fr/hal-00914697> , 2014