3D Optimisation of Average Torque of Electrostatic Micromotors

TB. Johansson, K. Hameyer, M. Van Dessel, R. Belmans

Abstract The paper presents a technique to optimise electrostatic micromotors. The finite element technique is combined with different types of optimisation strategies, of which one of them, a version of the evolution strategy is discussed. It further involves an automated generation of 3D meshes where the dimensions and rotor positions of the motor models can be chosen arbitrarily. It also involves constructing equivalent circuit models for the electrostatic motors. Both, the automated 3D mesh generation and the construction of the equivalent circuit model are treated in detail in this paper. The possibilities of producing powerful tools in the field of optimisation are described, but also some warnings and limitations concerning the automated optimisation are mentioned.

1 Introduction

The main challenge for electrostatic micromotors among other electrostatic micromechanical devices is still, after their introduction, achieving enough torque to overcome the friction forces due to its small dimensions, described e.g. Trimmer (1989). This is why optimisation of each micromotor design is a necessity to ensure or even enable good operation.

Electrostatic motors, sometimes called variable capacitance motors, consist of a specified number of stator and rotor electrodes (poles). Depending on the arrangement of stator and rotor poles the electric flux of these motors can be directed perpendicular to or along the motor shaft, being a radial flux or an axial flux motor respectively. Both types are investigated in this paper.

Two numerical methods are combined to obtain a self running optimisation procedure. Figure 1 illustrates the simple link between the numerical field computation and the numerical optimisation method. With a valid set of initial parameters and some additional constraints the optimisation problem is set up.

The generation of a three dimensional finite element mesh constitutes the first step. Then the set of differential field equations is solved, the equivalent circuit parameters generated and the circuit analysed for the average torque generated and/or torque-ripple. This is the evaluation of the objective function. The numerical optimisation takes the results from the field calculation and decides whether to stop the optimisation or to continue in generating a new set of objective variables. In this paper this is done with a stochastic search strategy, the evolution strategy.

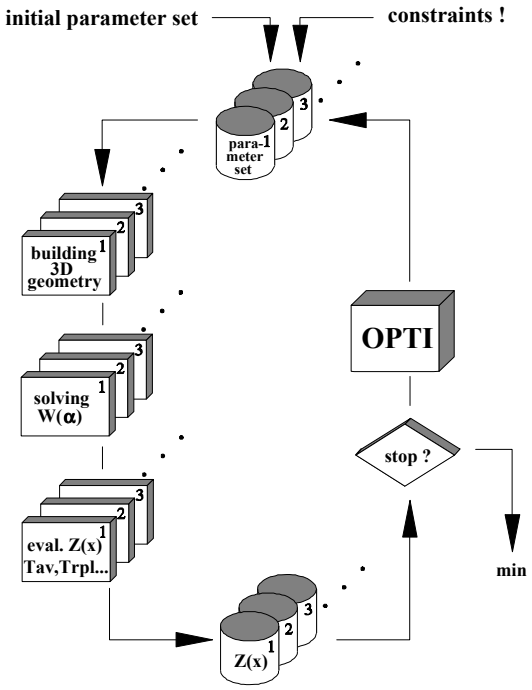

Fig. 1 Process control

2 Numerical Optimisation - Evolution Strategy

Stochastic optimisation techniques such as the evolution strategy and simulated annealing are attracting a great deal of interest due to their

- insensibility to disturbances,
- easy treatment of constraints,
- straightforward implementation,
- independence of derivatives,
- stable solutions.

Their main advantage is the insensibility to stochastic disturbance of the objective function caused by numerical evaluation. This insensibility results out from non-deterministic search and the fact that no derivatives are needed. The second important property is the easy

treatment of constraints. A complicated transformation into an unconstrained problem formulation is not necessary. The generation of the different configurations is independent. This offers the opportunity to run the algorithm on parallel computers. Figure 1 illustrates this process reducing the high computational cost to a large extent.

In this paper only higher evolution strategies are taken into account. More detailed information can be found in Hameyer and Hanitsch (1993) and Schwefel (1981).

Numerical optimisation requires a single function to be minimised.

$$Z(\mathrm{x}) = Z(x_1, x_2, ..., x_n) \quad \rightarrow \quad \min. \tag{1}$$

This function is called the objective- or quality-function. It depends on all design parameters involved. When minimising with respect to more than one goal $Z_1, Z_2, \ldots, Z_n$ simultaneously, the single aim has to be weighted in a linear combination. With Z as the total quality and a_i the weighting factors it can be formulated as:

$$Z = \sum_{i=1}^{n} a_i \, Z_i \tag{2}$$

The evolution strategy copies the natural principles of mutation and selection (*survival of the fittest*) by the biological evolution into the technical optimisation problem. The basic concept of the evolution strategy is founded in the substitution of DARWIN's notation of fitness to the quality of a technical product. The driving force in the optimisation process is the repetition of mutation and selection in successive steps. The mutation of the objective variables of an initial generation (parents) leads to a number of offspring's. The variables of one child may depend on multiple parent variable vectors. The best offspring's are selected to form the next generation.

In the optimisation process the repetition of mutation and selection of the objective variables leads from a temporary to an improved solution. With μ the number of parents, λ the number of offspring's or children and ρ a hereditary factor, at the (μ/ρ , λ)-strategy ρ parent vectors contribute to the creation of a child. $1/\rho$ of the properties of one parent is transmitted to a child.

Mutation of an object is done with random additions to the object variables. For iteration step (k) the new variable $x_{C,i}^{(k)}$ is created from the parent variable $x_{P,i}^{(k)}$

$$x_{C,i}^{(k)} = x_{P,i}^{(k)} + \delta_{C,i}^{(k)} \; p_i^{(k)} \quad \text{with} \quad i = 1(1)n \tag{3}$$

To transmit the step width $\delta_{C,i}^{(k)}$ of the parent generation to the child generation, the step length of the randomly chosen parents is taken to build an average value.

$$\delta_{C,i}^{(k)} = \frac{1}{\rho} \sum_{j=1}^{\rho} \delta_{P,j}^{(k)} \left(z_j^{(k)}(\mu) \right) \text{ with } \begin{cases} i = 1(1)n \\ j = 1(1)\rho \end{cases} \tag{4}$$

$z_j^{(k)}(\mu)$ is an uniformly distributed random integer number out of the interval [1,μ]. To obtain the mutation step width the before mentioned average value is multiplied by a step width factor α.

$$\delta_{C,i}^{(k)} = \begin{cases} \delta_{C,i}^{(k)} \cdot \alpha & : \; i = 1(1)\,\lambda/2 \\ \delta_{C,i}^{(k)} \cdot 1/\alpha & : \; i = \lambda/2 + 1(1)\,\lambda \end{cases} \tag{5}$$

3 Numerical Field Computation

The method to compute the field of the electrostatic micromotors used is the 3D finite element method. Earlier work has been limited to 2D finite elements, by Johansson et al. (1992). For radial flux motors this is possible, since their geometry does not vary in the axial direction, (fig. 2), but as the axial length is short compared with the rotor radius a 3D analysis is to prefer. For the axial flux motors, 3D analysis is required due to the geometry, (fig 3).

3.1 Automated Construction of 3D Meshes

Building 3D meshes is normally a time consuming process. Therefore, performing optimisation in 3D is very tedious since a large number of models, slightly different from one another, must be built and analysed. For micromotor structures the required labour is even vaster since for every model the analysis normally must be carried out for a number of rotor positions, in order to obtain macroscopic parameters as torque and capacitance as function of the rotor position. For most 3D finite element packages a new rotor position requires a new mesh to be generated. In this paper the fast and reliable extrusion-based 3D mesh generator is automated in such a way, that a large range of different motor geometries can be generated, with a fully arbitrarily choice of the rotor position.

3.2 Periodic Geometry

All electric rotating motors inherently possess some kind of periodicity. Typically each motor has a stator and a rotor where the geometric period, in degrees, is defined as the pole pitch τ. Each periodic geometry also has an extension within the pole pitch. This extension is referred to as the polar arc τ'.

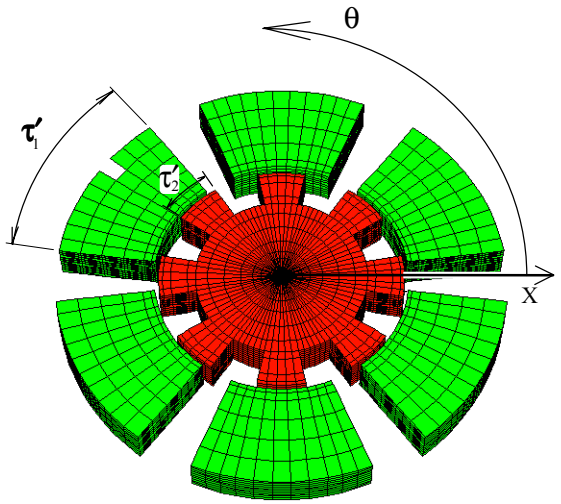

Fig. 2 3D mesh of a 6/8 pole electrostatic radial flux micromotor

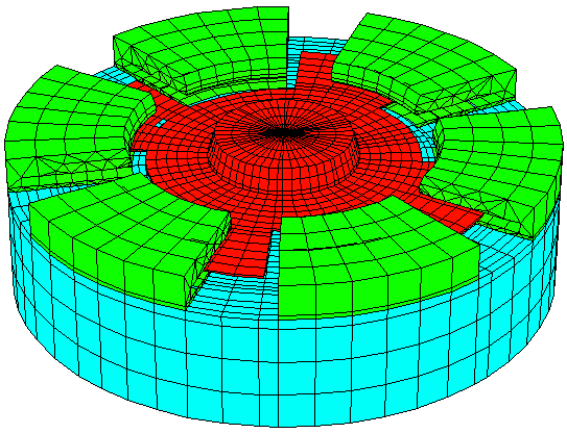

Fig. 3 3D mesh of a 6/4 pole electrostatic axial flux micromotor

With these two parameters, the pole pitch τ and the polar arc τ', it becomes easy to describe where the sides of the stator teeth and rotor teeth, parallel with the rotor axis, are located in the motor. If the motor is scanned in the positive θ direction, (figs. 2 and 3), location of the front-sides and the back-sides of the stator teeth are defined by:

$$\left.\begin{aligned} \Theta^{n}_{1,front} &= \left(\frac{\tau_1 - \tau'_1}{2}\right) + n\cdot\tau_1 \\ \Theta^{n}_{1,back} &= \left(\frac{\tau_1 + \tau'_1}{2}\right) + n\cdot\tau_1 \\ \text{where} \quad n &= 0,1,2,..\left(\frac{360°-\tau_1}{\tau_1}\right) \end{aligned}\right\} \qquad (6)$$

The reference point for θ is chosen in the middle between two stator teeth. By introducing the offset angle α accounting for the rotor position, the same may be done for the rotor.

$$\left.\begin{aligned} \Theta^{n}_{2,front} &= \left(\frac{\tau_2 - \tau'_2}{2}\right) + n\cdot\tau_2 + \alpha_2 \\ \Theta^{n}_{2,back} &= \left(\frac{\tau_2 + \tau'_2}{2}\right) + n\cdot\tau_2 + \alpha_2 \\ \text{where} \quad n &= 0,1,2,...\left(\frac{360°-\tau_2}{\tau_2}\right) \end{aligned}\right\} \qquad (7)$$

The motor geometry can, in the general case, be split in more than 2 periodic geometries. Subscripts 1 and 2 indicate parameters for the stator and rotor respectively.

3.3 Extrusion Technique

The 3D meshes are built using an extrusion technique. With this technique many 2D meshes, placed at different locations in space, are connected in order to generate the 3D mesh. The reference 2D mesh is referred to as the baseplane, (fig. 4). Each plane is applied with extrusion data describing its position in space. The extrusion technique, discussed in this paper, only uses a rotation around the y-axis, thus θ, to define the extrusion data. One vertical side of the base-plane coincides with the global y-axis. Since each copy of the base-plane only gets the θ parameter as extrusion data, one of the vertical sides of all planes also coincide, with the global y-axis, i.e. the centre of the rotor shaft, (fig. 5).

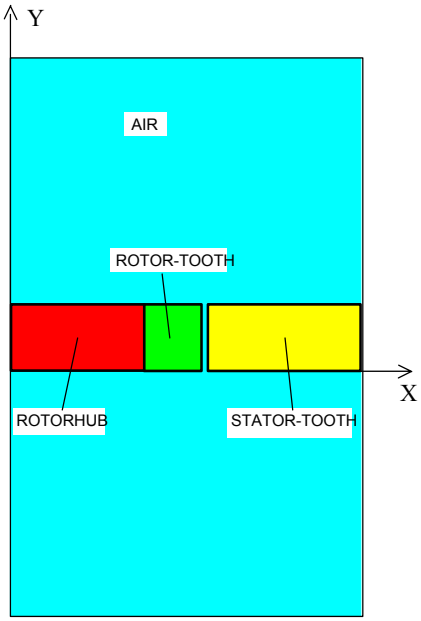

Fig. 4. Base-plane used when generating model of fig. 2

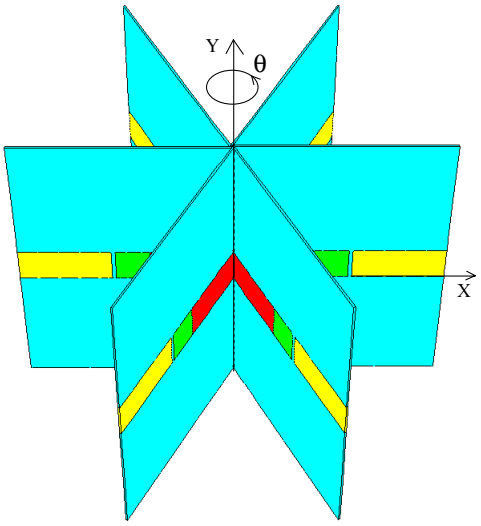

Fig. 5 Copies of the base-plane mesh, placed at different angles θ

From the knowledge of the number of poles, the polar arc τ' of stator and rotor, and the rotor position, and using (6) and (7) it is easy to find where planes are located. For each of the periodic geometries 4 different type of actions (sequence of relabeling and/or constraining) are defined. Types 1 and 2 define a front and a backside respectively, on e.g. a rotor tooth. Types 3 and 4 define a plane inside or outside of e.g. a rotor tooth. These four actions define every plane for each periodic geometry. Thus the maximum number of different plane-types required to construct a model is 4^N, where N is the number of periodic geometries in the model. These planes are generated in advance and are sufficient to define all possible combinations of τ, τ' and α.

Once the planes are generated the remaining task is to find out at which angle what kind of plane must be located. This is performed with repetitive use of (6) and (7) and a sorting algorithm. This sorting algorithm finds what the next plane must be from knowing the previous. A front-plane must be followed by and inside-plane, an inside-plane by a back-plane and so on. The angle between two consecutive planes has to be chosen depending on the size of the elements in the base-plane and the required aspect ratio of the 3D elements, the tetrahedrons. This implies that the same kind of planes sometimes has to be repeated with the desired angle between them in order to improve the aspect ratio of the tetrahedrons. This is automatically controlled.

3.4 Performance

The major advantage of automating the extrusion-based mesh generator, rather than a solid modelling mesh generator, is speed. To generate a motor model with approximately 100,000 tetrahedrons takes, from the instant that the parameters are entered to the model is built ready for the solver, less than 10 minutes on a HP715. Once the different kinds of planes are generated, it takes always less then 5 min. This should be compared with typically more than 45 min for a solid modelling mesh generator.

4 Field Solution

The application of the methods mentioned is demonstrated by optimising two electrostatic micromotors. The objective is to maximise the average torque. The average torque is calculated for the best possible excitation of each motor. To find this excitation an equivalent circuit technique has been used, (fig. 6).

4.1 Equivalent Circuit Model

Although the finite element calculation together with discrete fourier transform, DFT, can provide the stored electric energy as function of the rotor position, $W_e(\alpha)$, in the form of a fourier series, it is only valid for that or those excitations applied during the finite element calculations.

The information gained from two different excitations over the rotor positions however, is sufficient to create an equivalent circuit describing the motor geometry. The equivalent circuit used for a 6/8 pole motor is shown in figure 6 and consists of 12 capacitors, i.e. twice the number of stator electrodes (poles).

Their capacitance varies with the rotor position α. There are only two principally different capacitances. Writing $C_{SR}^{k}(\alpha)$ for the capacitance between stator electrode *k* and the rotor, C1 to C6, and $C_{SS}^{k}(\alpha)$ for the capacitance between stator electrode *k* and the next stator electrode, C7 to C12, it can readily be seen that the difference between capacitances with different index *k* is only a phase shift

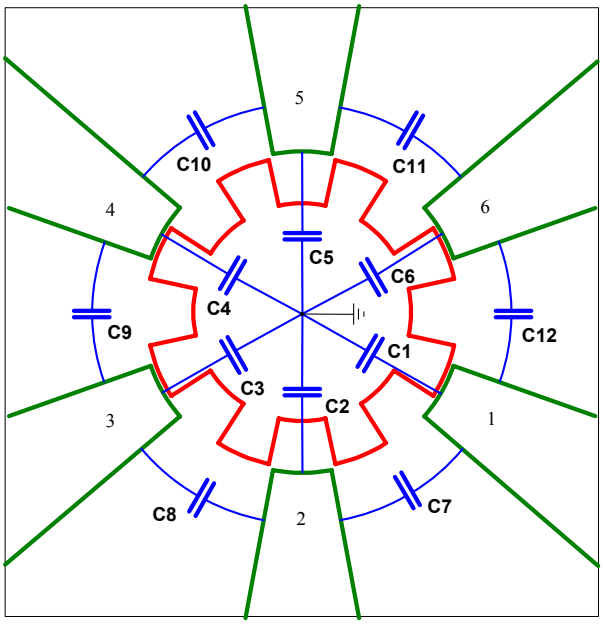

Fig. 6 Equivalent circuit for an electrostatic motor with 6 stator poles

equal to a multiple of the stator pole pitch τ_2 . With two different excitations for n rotor positions, the stored energy is calculated and the system of equations,

$$\begin{aligned} W_e(\alpha_1) &= \tfrac{1}{2}\sum_{k=1}^{n_p}\left[C_{SR}^k(\alpha_1)V_k^2 + C_{SS}^k(\alpha_1)\cdot(V_{k+1}-V_k)^2\right] \\ &\vdots \\ W_e(\alpha_n) &= \tfrac{1}{2}\sum_{k=1}^{n_p}\left[C_{SR}^k(\alpha_n)V_k^2 + C_{SS}^k(\alpha_n)\cdot(V_{k+1}-V_k)^2\right] \end{aligned} \qquad (8)$$

where $V_{n_p+1} = V_1$

can be solved. Once $C_{SR}^k(\alpha)$ and $C_{SS}^k(\alpha)$ are found the torque is calculated using an analytic differentiation

$$T(\alpha) = \frac{\partial W_e(\alpha)}{\partial \alpha} \qquad (9)$$

4.2
Optimum Excitation Sequence

Having the equivalent circuit any kind of excitation wave shape can easily be applied. The block-wave is applied throughout the optimisation, since it is the most powerful one. To avoid or at least minimise radial forces on the rotor shaft, the motors have to be excited symmetrically. Fig. 7 shows the possible symmetric excitations of a motor with 6 stator electrodes, where the black electrodes are excited to 1 V and the white electrodes, and the rotor, are kept at 0 V.

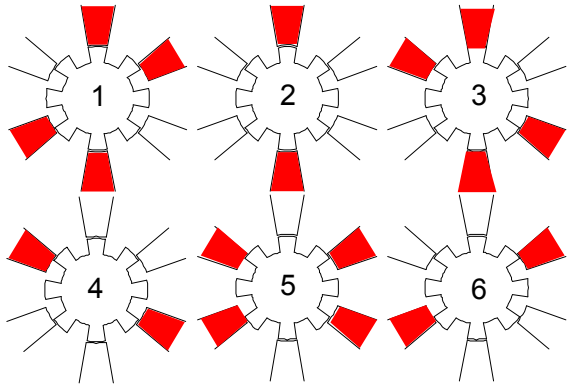

Fig. 7 Possible symmetric excitations of a motor with six electrodes

By applying these excitations to the equivalent circuit the torque characteristics over the rotor positions can be calculated. Fig. 8 shows the torque produced over one electric period for the six different excitations from fig. 7. By switching from one excitation to another, at the crossovers marked by arrows, the optimum excitation sequence is achieved. The average torque T_{av} is then calculated by integration of the curve always following the highest torque. The torque ripple T_{rpl} is given by

$$T_{rpl} = \frac{T_{max} - T_{min}}{T_{av}} \qquad (10)$$

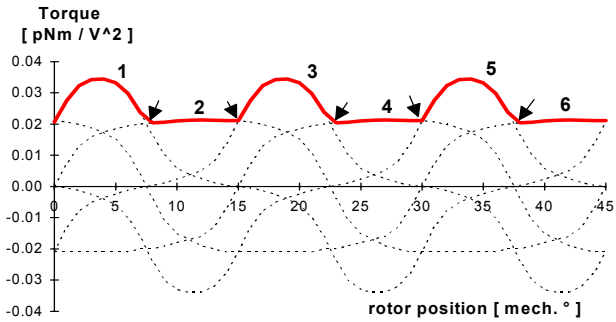

Fig. 8. Produced static torque over one electric period for 6 different excitations on a 6/8 pole motor.

5
Optimisation of a Micromotor

The geometry of the micromotors from figs. 2 and 3 is modelled by 2 design variables. The free parameters are the pole arcs of the rotor and stator teeth. Due to constraints resulting from the manufacturing conditions, the other design variable were set to fixed values. For the motor from fig. 2 the air gap is fixed to a value of 10 µm, the rotor height to 100 µm and the outer diameter of the rotor is 600 µm. For the motor from fig. 3 the air gap is fixed to a value of 3 µm, the rotor thickness to 4 µm and the outer diameter of the rotor is 320 µm.

6
Results

Fig. 9 shows a surface fit through the selections when the objective was to maximise the average torque. Figure 10 shows a surface fit of the torque ripple for the same selections of design parameters. Both figures refers to optimisation of the 6/8 pole motor with radial flux, (fig. 2.) The maximum average toque for this motor is calculated to be 71 pNm for an excitation of 50 V.

Fig. 11 and 12 show for the 6/4 pole axial flux motor from fig. 3 the same properties. The surface fits in fig. 11 and 12 are from 18 selections of design parameters made according to the optimisation strategy called *surface response methodology*. With this technique first 9 selections of design parameters was made. A function is fitted to the resulting 9 torques and in the area of maximum. torque a new set of 9 selections is made. A new function is fitted to all 18 torques. The maximum average torque of the axial flux motor from fig. 3 is calculated to be 149 pNm for an excitation of 50 V.

It can be seen from figs. 9 and 11 that the average torque has one global maximum for two design variables, and the torque ripple one local maximum for almost the same combination of parameters. This was found also from the 2D optimisations, by Johansson et al. (1992).

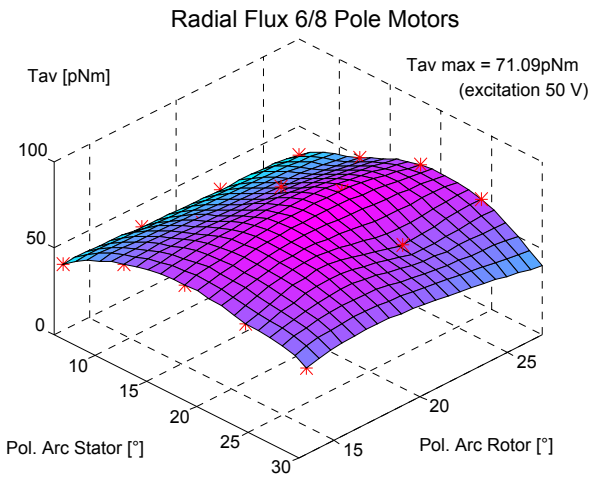

Fig. 9 Average torque as function of combinations of polar arcs in the stator and the rotor, for 6/8 pole radial flux motors represented by a surface fit of the selections during the optimisation

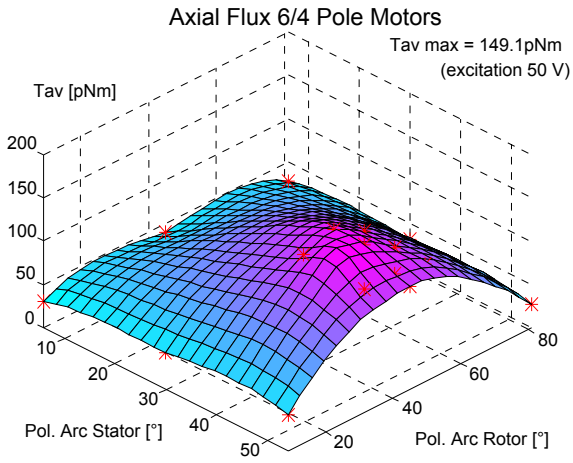

Fig. 11 Average torque as a function of combinations of polar arcs in the stator and the rotor, for 6/4 pole axial flux motors represented by a surface fit of the selections during the optimisation

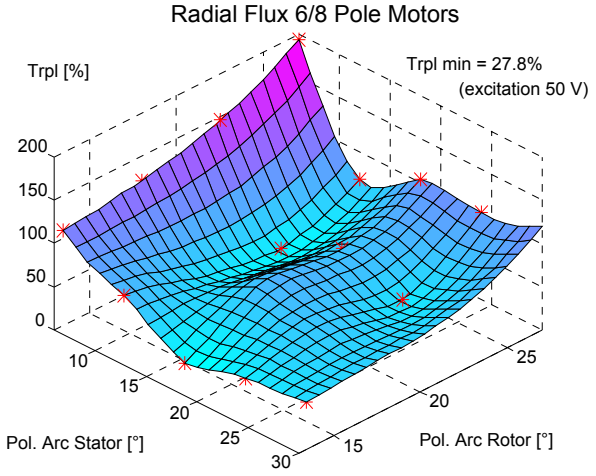

Fig. 10 Torque ripple as a function of combinations of polar arcs in the stator and the rotor, for 6/8 pole radial flux motors represented by a surface fit of the selections during the optimisation

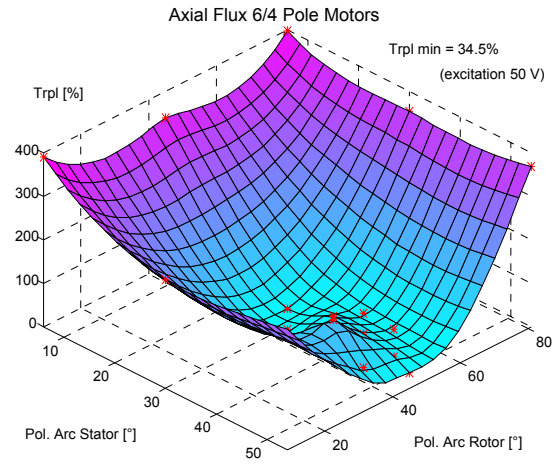

Fig. 12 Torque ripple as function of combinations of polar arcs in the stator and the rotor, for 6/4 pole axial flux motors represented by a surface fit of the selections during the optimisation

Other problems, reported by Johansson et al. (1994), with the optimisations of the radial flux 6/8 pole motor have been re-examined. It was suggested that the use of discrete Fourier transform (DFT) would, for some motors, introduce errors rather than filter errors. This was told to be for motors which energy function W(α) is partially very flat. The DFT here introduces oscillations if the function is sampled with to few points. This problem exist but does not cause as large problems as the first view indicated. The actual error was due faults in the software generating the 3D models. This has been fixed and figs. 9 and 10 in this text shows the results from the improved model generator.
The model generation error did however not cause such large problems for the axial flux motors. The optimisation is repeated also for these motors but, compared to the previous simulation, very little difference is observed.

In the 2D study and in the 3D study, one electric period is discretised with 45 and 18 rotor positions respectively. The reason why the number of rotor positions is reduced for 3D is the computation time. The computational cost for one rotor position was for 2D about 1 min for a first order solution and about 2 min for a second order solution. For 3D this is in the range of 20 min for a first order solution and 45 min for a second order solution.

7 Conclusion

An application of the evolution strategy, a method for minimisation of functions of continuous variables is presented. Numerical field computation to evaluate the objective function is performed using a three dimensional finite element method.

The presented technique for automating the model generation and extracting global quantities as average torque offers a tool well worth exploring further. In this text the analysis has been performed statically. If the equivalent circuit defined here is complemented with e.g. terminal resistance, the inertia of the rotor and the friction torque, as indicated by Sarraute et al. (1994), this would already be sufficient for a dynamic analysis.

The optimisation method described in the paper is very easy to implement since it need no other knowledge of the system than the constraints. However, it does need many function evaluations to find the global optimum. Since for 3D finite elements, the evaluation is rather time-consuming there is a risk that many hours are lost before it can be stated that the accuracy of the function evaluations are poor for some of the selections, but not for all of them. It is thus of great importance that the quality evaluation works well in the whole range of design variables.

The generation of different configurations is independent. This offers the opportunity to run the algorithm on parallel computers reducing the computation time.

References

Hameyer K. ; Hanitsch R. (1993): "Numerical optimization of the electromagnetic field by stochastic search and MEC-model", *Proc. Conf. COMPUMAG*, November 1993, Miami, USA.

Johansson T.B. ; Hameyer K. ; Van Dessel M. ; Belmans R. (1994) : "3D Optimisation of Average Torque of Electrostatic Micromotors", MST '94, MICRO SYSTEM Technologies 94, Berlin, Germany, October 19 - 21 1994, pp. 949-957.

Johansson T.B. ; Van Dessel M.; Belmans R.; Geysen W. ;Hanitsch R. (1992): "An optimisation scheme of electrostatic micromotors based on an Equivalent Circuit - Finite Element approach.", *Proc. Conf. ICEM 1992*, UMIST Manchester, UK, pp. 1157-1161.

Schwefel H.P. (1981): "Numerical optimization of computer models", *Wiley & Sons*, Chichester, 1981.

Sarraute E. ; Lefèvre Y. ; Lajoie-Mazenc M. (1994): "Modelisation and simulation of the dynamics of variable capacitance micro motors", ICEM '94, Paris, France, 5-8 Sept. 1994, pp. 305-310.

Trimmer W. (1989): "Microrobots and Micro-mechanical systems", Sensors and Actuators Vol. 19, 1989, pp 267-287