COMPUTATIONAL PROBLEMS OF ELECTRICAL ENGINEERING
Vol. 3, No. 2, 2013

TECHNOLOGY OF INVESTIGATION OF ASYNCHRONOUS MOTORS WITH A SQUIRREL-CAGE ROTOR USING A UNIVERSAL PARAMETRIC GENERATOR OF A FINITE-ELEMENT MODEL

Andrey Tikhonov, Yuriy Kazakov, Aleksandr Shmelev
Ivanovo State Power University, Ivanovo, Russia
ait@dsn.ru, schmeloff@yandex.ru, elmash@em.ispu.ru

Abstract: The technology of the numerical research into asynchronous motors with a squirrel-cage rotor with arbitrary-shaped slots using a universal parametric generator of the finite-element magnetic field models has been developed. The mathematical built-in software packages with an integrated, developed by the authors, dynamic-link library EMLib are used. The parametric generator allows a series of models for asynchronous motors used in programmable numerical experiments to be created.

Key words: asynchronous motor, finite-element model, parametric model, numerical experiment.

1. Introduction

In market conditions there is a trend of transition from a large-scale production of electrical products, in particular asynchronous motors (AM), to small-scale manufacturing, and even one-unit-production. This increases the variety of product requirements. In this regard, the role of design and engineering departments of an enterprise grows, and the problem of designing AM reaches a new level of complexity. A phase of development and investigation of the prototypes leaves the product life cycle, it reinforces the role of mathematical modeling using high-tech precision numerical methods that are no longer a tool of research institutions and become an integral part of a design engineer's workplace. In particular, the precise verifying calculation of AM requires the use of techniques based on the calculation of physical fields by the finite-element method.

The main disadvantage of any system of finite-element modeling is the need for a considerable amount of preliminary work on the development of the model. Despite the presence of automation considering the complexity of the design geometry, it may take an experienced engineer a few hours to create the finite element model by means of an interactive graphical editor. There is often a need to generate a lot of similar models that differ by some parametric features, dimensions, number of slots (grooves) in the stator and the rotor, etc. In the problems of this kind, the parametric image editors, in which the model is based on the initially formed algorithm operating with a limited set of variables that uniquely determine the geometry of the computational domain, turn out to be the most comfortable things to work with. Changing the values of variables at a fixed constructs algorithm leads to restructuring the entire model.

The tools that automate creation of parametric models are implemented in most modern graphics programs. However, getting rid of the direct programming is possible only in typical special cases that do not require significant changes in the structure of the model, for example, when you change the size, number of slots, etc.

In some systems, particularly in AutoCAD, SolidWorks, Pro / Engeneer, etc., there is a traditionally radical way, when the parametric drawing is done by the built-in programming language AutoLisp, C + +, Visual Basic, etc. This way has the advantages in terms of flexibility in the formalization of complex constructions. However, from the point of view of an engineer who is not a professional programmer, this way is unacceptable.

At the same time, today's design engineers, in their work, freely handle mathematical packages such as Excel, MatLab, MathCad, Mathematica, etc. In particular, computational CAD subsystems of electric motors are often implemented by means of Excel; to create and analyse complex mathematical models of physical processes in electric machines, as well as to optimize their design, it is convenient to use the mathematical package MatLab; many designers since their student days have accustomed to using the MathCad and Mathematica. All of these packages have their own built-in programming systems adapted to a mathematical language conventional for engineers.

Thus, a relevant issue is the development of the technology which enables a series of finite-element models of AM in a short time to be generated, as well as tested (analysed) using common and familiar for the design engineers mathematical packages.

2. Library of finite-element modeling of two-dimensional quasi-stationary magnetic fields

Traditionally, systems of modeling of physical fields classified as CAE-systems (e.g. ANSYS, Comsol, Cosmos, ElCut, etc.) function as independent autonomous objects (Fig. 1). The internal functions of autonomous systems are separated from the user by the interface window which the interaction between the system and the user takes place through. The system plays a role of the basic environment in which applications created by an engineer address to the intrasystem functions [1]. It binds a designer to a specific CAE-sytem.

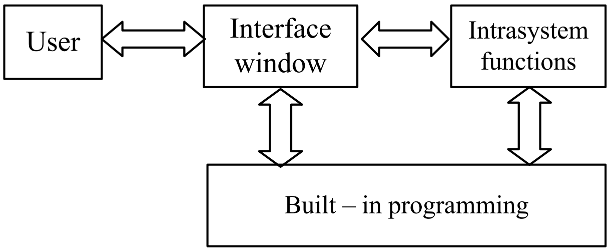

Fig. 1. Scheme of functioning autonomous systems of modeling.

At the same time, there is no system that can satisfy all the needs of the designer. For example, for large-scale numerical experiments it is necessary to combine the capabilities of various software tools. In this sense, using one of the mathematical packages such as Excel as a basic system proves to be the most convenient as it provides the designers with an appropriate for CAD tasks tabular and graphical interface.

A fundamentally different idea can be realized if the CAE-system is a software component that can integrate with any open application that has built-in programming, supplying it with new features, with which you can programmatically generate and explore a finite element model [2]. In addition, the components supply the basic application with an interface window in which all operations with the model are displayed (Fig. 2). This window provides the user with functionality, allowing them to work with the model interactively.

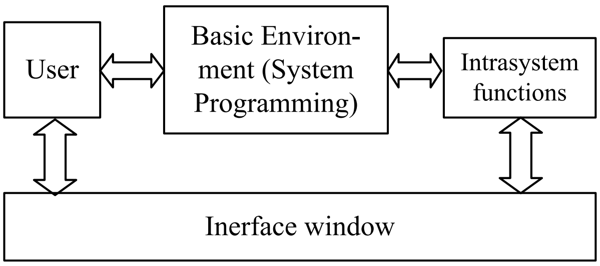

Fig. 2. Scheme of components functioning.

In particular, it has been developed a dynamic-link library EMLib enabling to generate and analyze (test) plane-parallel or rotationally symmetric models of a quasi-stationary magnetic field in the electric motors [3]. The ways of programming a numerical experiment using the library EMLib integrable into any custom application are described in [4].

3. Parametric generation of finite-element asynchronous motor models

The need to generate a series of finite-element models may arise while solving the problem of the optimization the AM design. At the same time the variable parameters can be the values determining not only parametric, but also structural features of the design. Specifically, the finite-element model of asynchronous motors generated at each step of finding the optimal solutions can differ not only in geometrical dimensions, but also in quantity, as well as in shape of the rotor and stator slots, the number of winding layers in the stator slots, the connection diagram of the stator coils etc.

This will allow us to approach the formulation of and solution to the problem of structural optimization. Due to the discreteness of the model, the optimization may be performed, for example, using the genetic algorithms.

In the two-dimensional formulation of the finite-element problem, the computational domain is a plane-parallel or plane meridian cross section of the magnetic system of AM, the geometry of which is given by a set of graphical primitives (lines or arcs)

$$\mathbf{P} = \{P_i \mid P_i = (\grave{O}_{1i}, \grave{O}_{2i}, \grave{O}_{3i})\} \qquad (1)$$

predetermined by reference points (T_{1i}, T_{2i}, T_{3i}), belonging to the set of reference points

$$\grave{\mathbf{O}} = \{T_i \mid T_i = (x_i, y_i, r_i)\} \qquad (2)$$

where x_i and y_i denote the coordinates; r_i stands for the approximation radius of primitives in the vicinity of the reference point.

Nodes of the finite-element mesh dividing it into sections are exposed on the primitives,. Piecewise linear approximation primitives are the boundaries of the sub-regions R_i from the set below

$$\mathbf{R} = \{R_i \mid R_i = (x_i, y_i, n_i, i_i) \wedge R_i \leftarrow M_j \in \mathbf{M}\} \qquad (3)$$

where x_i and y_i represent the coordinates of a point having its guaranteed position inside of the i-th sub-region; n_i is the number of current-carrying conductors in the subdomain; i_i denotes the amount of current in the conductor.

Each i-th subdomain is uniquely associated with an object M_j, that is an element of a set of media

$$M = \left\{ M_i \middle| M_i = \left(p_i, s_i, \mu_i \right) \wedge M_i \leftrightarrow N_j \in N \right\} \qquad (4)$$

where p_i is the type of media (non-magnetic, ferromagnetic, etc.), s_i represents the class of steel (if steel), μ_i stands for the relative permeability (for non-magnetic media). Media M_i can be uniquely related to the object N_j, which is the magnetization curve of the sets

$$\mathbf{N} = \{\mathbf{N}_i \mid \mathbf{N}_i = \{N_{ij} \mid N_{ij} = (B_{ij}, H_{ij})\}\} \qquad (5)$$

where B_{ij} is the magnetic induction in the j-th point of the i-th curve; H_{ij} is the magnetic field.

Each subdomain is triangulated by the exhaustion of the resource (leveling-cut) [4]. In this case, it is filled with triangular elements, forming a set of

$$\mathbf{E} = \{E_i \mid E_i = (U_{1i}, U_{2i}, U_{3i}, \mu_i) \leftrightarrow R_i\} \qquad (6)$$

where μ_i, denotes the permeability of the element; U_{1i}, U_{2i}, and U_{3i} are the nodes of the finite element mesh, which are the vertices of the triangular element and belong to the set

$$\mathbf{U} = \{U_i \mid U_i = (x_i, y_i, p_i)\} \qquad (7)$$

where x_i and y_i are the coordinates of a point; p_i is the node status (current node, the node Dirichlet boundary or boundaries of one of the periodicity).

Elementary objects that comprise winding sections are S_i, forming a set of sections

$$\mathbf{S} = \{S_i \mid S_i = (x_{1i}, y_{1i}, x_{2i}, y_{2i}, w_i, r_i, i_i) \wedge \\ \wedge S_i \leftarrow R_{1i} \wedge S_i \leftarrow R_{2i}\}. \qquad (8)$$

Each section is characterized by coordinates of the active part of the beginning of the section (x_{1i}, y_{1i}), the coordinates of the active side of the end section (x_{2i}, y_{2i}), the number of turns of w_i, electrical resistivity sections r_i. In addition, the section is uniquely related to a one-way communication with the subdomains R_{1i} and R_{2i}, in which its active sides lie.

The series-connected sections form branches, of which the windings are constructed.

A universal finite-element model of AM is based on three objects: a stator, a rotor and a gap. The stator and rotor in turn are constructed from the corresponding tooth zones and backs of the stator and rotor. The backs of the stator and rotor are the usual rings encircling the tooth zone. The basic object of the tooth zone is a slot. Thus, when creating a parametric generator of the finite-element model of AM, work on the algorithmic construction of different forms of slots is considered the most time-consuming.

To unify the procedures for constructing a model of the slot, it is reasonable to break it up into several elements: the spline, the wedge part, the bottom of the slot, central part of the slot. The work has been done on the classification of the data elements used to limit the number of possible options for their execution. By choosing different versions of the data elements and by their combination practically all really existing versions of the stator slots of AM can be built. At present, the work directed to the establishment of a comprehensive library of squirrel-cage rotor slots is being carried on. If there emerge new non-traditional forms of the rotor slots, the library can be expanded by an additional subroutine, which is generated by a standard method.

The presence or absence of the slot insulation is taken as a separate parameter (to simplify the model, the presence of the slot insulation is often neglected).

The Slot Generator provides the following operations:

1) calculation of control points' coordinates on the basis of the list of values characterizing the geometry of the domain under discussion;

2) formation of a description of the graphic primitives based on the generated reference points;

3) formation of a description of the subdomains limited by the generated graphic primitives;

4) triangulation of subdomains.

In particular, Fig. 3 shows some of the options of stator slots created by the generator of the finite-element model of AM.

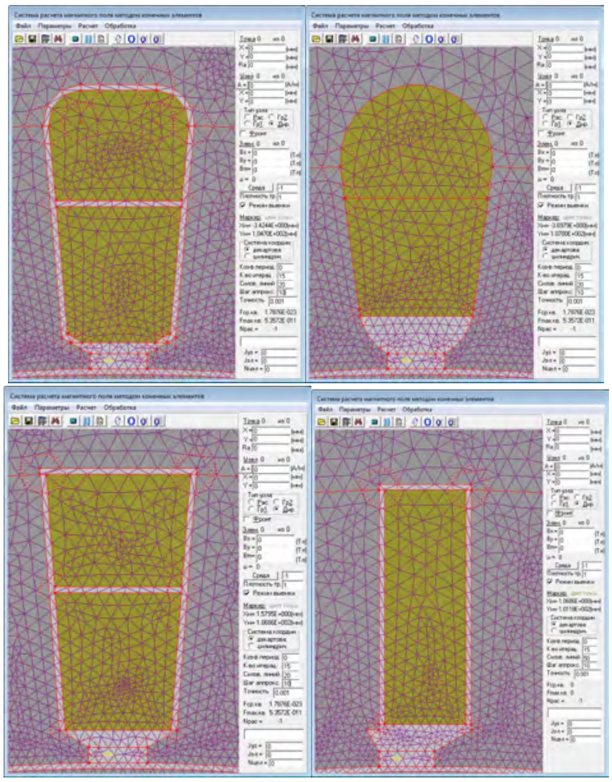

Fig. 3. Variants of stator slots of AM.

The array of the slots is generated by subsequent shifting of one slot by an angle of tooth-shift-distance. Furthermore, the teeth are generated between the slots. The back is formed separately. The gap between the rotor and stator is the last to be triangulated.

Sections of the stator windings are generated after the generation of a finite-element mesh. The stator windings and the contours of the rotor winding are built from these sections.

One of the AM models created in this way is shown in Fig. 4.

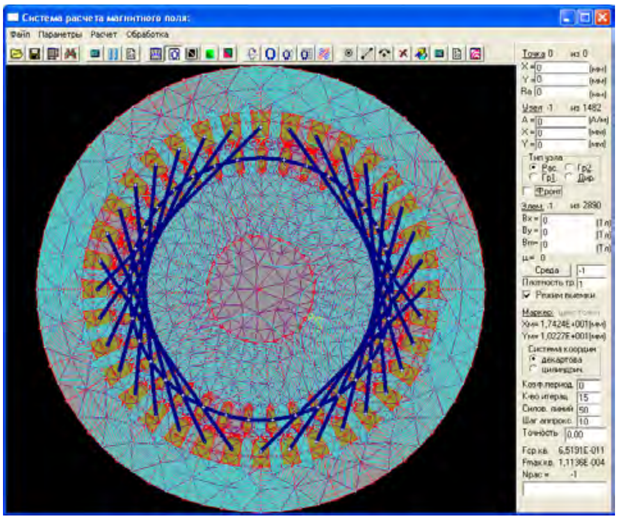

Fig. 4. A generated finite-element AM model.

4. Programming of numerical experiments using finite-element model of AM.

Generation of a finite-element model of AM is not an end in itself. The formed finite-element model can be used in the numerical experiments, built on the principle of imitation of a real experiment.

We can distinguish two modes of numerical experiments:

1) interactive;
2) program.

A traditional mode is an interactive mode, where a researcher works directly in the IDE of a field modeling system, taking the advantage of this system control means and providing feedback to the model by the means of its visualization.

The study comes, as a rule, to analyzing an impact of various deformations of the investigated domain on the picture of the field or to analyzing field patterns for different values of the currents in the windings, etc. In order to carry out a series of calculations, it is necessary to record the results of single experiments.

The software implementation is used if you need to obtain a series of calculations. The method of execution of numerical experiments using the library EMLib is a continuation of the idea of parametric generation of the field model which allows, by varying the parameters, to perform a programmed series of deformations of a magnetic system structure, as well as a series of **calculations of the magnetic field at different currents in** the windings and at different positions of the rotor, etc. to be implemented.

To program the numerical simulation, the library EMLib delivers into a selected mathematical processor several functions enabling:

1) to set currents in the windings and individual sections of AM;

2) to rotate the rotor to a predetermined angle with a subsequent re-triangulation of the gap;

3) to calculate the magnetic field at given input data;

4) to visualize the results of the calculation in the interface library window;

5) to return the value of the magnetic vector potential at the points with given coordinates, the components of the magnetic induction in triangular elements, the flux linkage of sections and windings, the value of the electromagnetic torque acting on the rotor, etc.

Processing of the results of the numerical experiment is carried out by means of mathematical processor according to the algorithm provided by a designer. There are two possible modes of the use of the field calculation results:

1. Training mode. The experiment is divided in time into two stages. The first stage is an exhaustive search for variants with the help of a finite-element model and defining a relationship between the calculation results and varying values, for example by means of a spline approximation. This relationship is used at the second stage of the numerical experiment as a function replacing the finite-element model. The advantage of such a mode is its high-speed performance at the second stage. The disadvantage is the need for a large number of preliminary calculations.

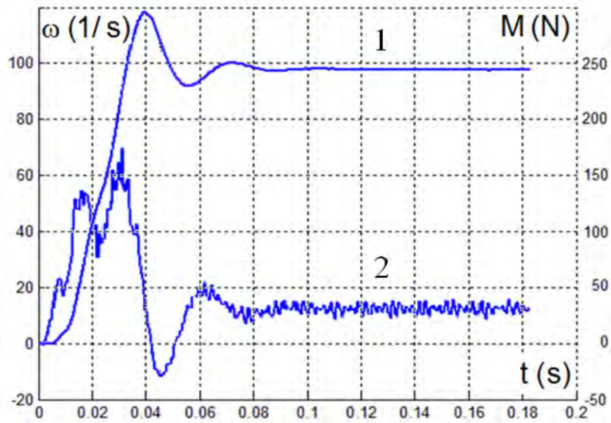

Fig. 5. The results of the numerical experiment: 1 denotes the time dependence of the angular velocity ω; 2 represents the time dependence of the electromagnetic moment M.

2. The mode with a direct application of the finite-element model at each step of the numerical experiment. This causes a significant reduction in the model performance speed. It can "make a noise", that is to produce the outcomes (results) which are distorted by the limitation of bit values. The error increases sharply if the results of calculation are differentiated. This mode is preferred if the number of variable values exceeds 5-6.

In [5,6,7,8], the results of modeling of the dynamic modes of AM in both modes of using the field model. In particular, Fig. 5-6 show the curves of AM acceleration taking into account the displacement of current in the rotor slots, calculated by using the library EMLib. It is planned to optimize AM using the finite-element model.

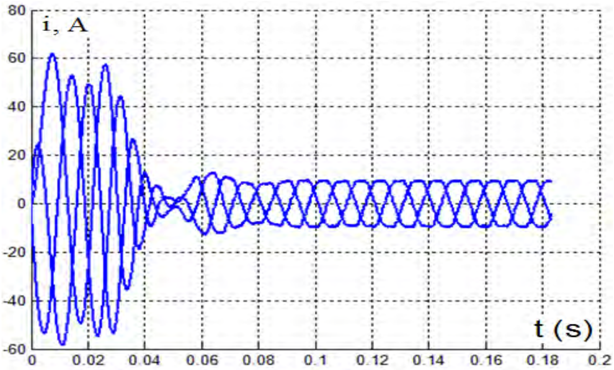

Fig. 6. The results of the numerical experiment: the time dependence of the currents in the stator windings.

5. Conclusions

The developed technology allows us to carry out programmable numerical research of asynchronous machines using a finite element model of magnetic field.

The work is performed within the contract with the Ministry of Education and Science of the RF 02.G25.31.0049.

References

[1] A. Tikhonov, S. Kucherov, I. Lashmanov, and D. Rubtsov, “Technology of numerical investigation of electric machines by using the finite-element library for magnetic field simulation”, *Vestnik IGJeU*, vol. 3, pp. 5 – 8, 2006. (Russian)

[2] A. Tikhonov, *“Techniques of analysis and synthesis of electromechanic devices on the basis of component integration model”, GOU VPO «Ivanovskiy gosudarstvennyi energeticheskiy universitet imeni V.I. Lenina»,* pp. 100, Ivanovo, Russia, 2006. (Russian)

[3] A. Tikhonov, Russian patent № 2006610323. Case № 2005612109, dated of 08/15/2005, *Library of field simuation*.

[4] A. Tikhonov and Y. Kazakov, “Programming of numerical experiments by using a finite-element model for magnetic field simulation in electromechanic devices”, *GOUVPO «Ivanovskiy gosudarstvennyi energeticheskiy universitet imeni V.I. Lenina»*. pp. 80, Ivanovo, Russia, 2008. (Russian)

[5] L.Bulatov and A.Tikhonov, “Dynamic model of an asynchroneous motor with a squirrel-cage rotor”, *Vestnik nauchno-promyshlennogo obshhestva,* Moscow, Russia: «Alev-V», vol. 1, p. 35-40, 2011. (Russian)

[6] L.Bulatov and A. Tikhonov, “Development of a dynamic model of an asynchroneous motor using results of finite-element calculations”, *Vestnik IGJeU*, vol. 4, p. 32-34, 2012. (Russian)

[7] L.Bulatov and A.Tikhonov, “ Dynamic model of an asynchroneous motor with direct application of finite-element model of magnetic field”, *Vestnik IGJeU*, vol. 1, p. 40 – 44, 2013. (Russian)

[8] L.Bulatov and A.Tikhonov, “Techniques for reduction of calculation time of dynamic processes in asynchroneous motor with a squirrel-cage rotor by means of direct application of finite-element model of magnetic field”, *Vestnik IGJeU*, vol. 3, p. 27-32, 2013. (Russian)

ТЕХНОЛОГІЯ РАЗРАХУНКУ АСИНХРОННИХ МАШИН З КОРОТКОЗАМКНЕНИМ РОТОРОМ З ВИКОРИСТАННЯМ УНІВЕРСАЛЬНОГО ПАРАМЕТРИЧНОГО ГЕНЕРАТОРА СКІНЧЕНО-ЕЛЕМЕНТНОЇ МОДЕЛІ МАГНІТНОГО ПОЛЯ

Андрєй Тіхонов, Юрій Казаков, Алєксандр Шмєлєв

Andrey Tikhonov – D.Sc, Professor. Born in 1962. Head of Department of Physics at Ivanovo State Power University, Russia. He is the author of more than 280 scientific works.

Yuriy Kazakov – DSc, Professor. Born in 1954. Head of Department of Electromechanics at Ivanovo State Power University, Russia. He is the author of more than 320 scientific works.

Aleksandr Shmelev – a post-graduate student at Ivanovo State Power University, Russia. Born in 1987. He is the author of more than 10 scientific works.