

# IMPLEMENTATION OF A NOVEL FORCE COMPUTATION METHOD IN THE FEMM SOFTWARE

VASIL SPASOV, IVAN KOSTOV, IVAN HADZHIEV

**Abstract:** A novel force computation method is implemented in the Finite Element Method Magnetics software - the nodal force method. For this purpose supplementary code is developed in C++ and added to the FEMM source code. In addition to force computation, the extended version of FEMM enables also force visualization that was not possible until now. To demonstrate the capabilities of the extended version of FEMM, the forces of three models are computed and visualized – two current-carrying copper busbars, an AlNiCo permanent magnet with steel core and a benchmark non-linear dc electromagnet. A comparison is made between the newly implemented nodal force method and the available Maxwell's stress tensor method from the viewpoint of accuracy and visualization capabilities.

**Key words:** Finite element method, electromagnetic force, nodal force method, FEMM

## 1. Introduction

Finite Element Method Magnetics (FEMM) is a finite element software for solving 2D problems in low frequency magnetics and electrostatics [1]. The program addresses linear and nonlinear magnetostatic problems, time harmonic magnetic problems and others. FEMM has been extensively used in science, engineering, industry and for teaching electromagnetics in higher education [2]. It is a free, open source, accurate and low computational cost product. There is no limit on the problem size – the maximum number of finite elements and nodes is limited only by the amount of available memory. This enables to solve problems resulting in more than a million elements on a personal computer.

The aim of this paper is to extend the capabilities of the FEMM software when computing and visualizing electromagnetic forces of electrical devices. For that purpose the mathematical model of a novel electromagnetic force computation method - the nodal force method, is developed [3, 4]. This mathematical model is implemented in the FEMM postprocessor by developing a C++ code.

To verify the extended version of FEMM, the forces of three models are computed and visualized. The models are two current-carrying copper buses, an AlNiCo permanent magnet with steel core and a benchmark non-linear dc electromagnet.

The present paper is organized as follows. The derivation and analysis of the nodal force method are presented in Section 2. The implementation of the nodal force method and the force visualization enhancements to FEMM are discussed in Section 3. The accuracy of the extended version of FEMM is validated numerically in Section 4. Finally, conclusions are drawn in Section 5.

## 2. Derivation of the nodal force method

The nodal force method (NFM) is derived for the two-dimensional case using first-order nodal triangular finite elements. In NFM the work  $\delta W$  performed by electromagnetic force for displacement  $\delta u$  is [5]:

$$\delta W = - \iint \tau_{ik} \frac{\partial(\delta u_i)}{\partial k} d\Omega; \quad (i, k = x, y). \quad (1)$$

Here  $\tau_{ik}$  are the Maxwell stress tensor components and  $\Omega$  is the analyzed region.

The Maxwell stress tensor components are defined as:

$$[T] = \frac{1}{\mu_0} \begin{bmatrix} B_x^2 - 0.5B^2 & B_x B_y \\ B_y B_x & B_y^2 - 0.5B^2 \end{bmatrix} = \begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \end{bmatrix}, \quad (2)$$

where  $\mu_0$  is the permeability of air,  $B$  is the magnetic flux density magnitude and  $B_j$  ( $j = x, y$ ) is the magnetic flux density component along the two axes.

The displacement is interpolated by the well-known continuous and piecewise-differentiable shape functions  $N_i$  of nodal triangular elements [6]:

$$\delta u_i = \sum_n N_i \delta u_{ni}, \quad (3)$$

where  $n$  is the number of element nodes and  $i$  is the shape function number.

Replacing (3) in (1) yields:

$$\delta W = \sum_n \left( - \iint \tau_{ik} \frac{\partial N_i}{\partial k} d\Omega \right) \delta u_{ni}; \quad (i, k = x, y). \quad (4)$$

On the other hand, the completed work is equal to:

$$\delta W = \sum_n f_{ni} \delta u_{ni}, \quad (5)$$

where  $f_{ni}$  is the  $i^{\text{th}}$  component ( $i = x, y$ ) of the nodal force acting on node  $n$ .

After equating (4) to (5) it is obtained for the nodal force:

$$f_{ni} = - \iint \tau_{ik} \frac{\partial N_i}{\partial k} d\Omega. \quad (6)$$

Based on (6), the  $x$  component of the force of node  $n$  of one triangular finite element can be computed by the 2D finite element method as follows:

$$f_{nx} = -(\tau_{xx} b_k + \tau_{xy} c_k) \cdot S_e. \quad (7)$$

Here  $b_k$  and  $c_k$  are the shape functions coefficients of the nodal first-order triangle and  $S_e$  is its area.

The integration in (6) for a node is performed for all elements to which the node belongs. The total force acting on a part is obtained by summing up the nodal forces of all nodes included in the part.

Next the computer implementation of NFM will be analyzed. Formula (7) shows that the nodal force method uses only quantities that have already been computed during the finite element analysis. In other words, due to the absence of additional arithmetic operations, the NFM needs less CPU time as compared to the Maxwell's stress tensor method (MSTM).

To perform (7), no integration contour should be defined, as required by the MSTM in FEMM. Thus two more advantages are to be expected: the NFM can be implemented fully automatically in models of arbitrary shape and its accuracy is not affected by the choice of the integration contour needed by the MSTM [1].

Another important advantage of the NFM is that it directly computes the local electromagnetic force, i.e. the force acting on every finite element node. Therefore, to create vector plots of force, no additional post-processing is needed. In contrast, the Maxwell's stress tensor method calculates only

global force. Local force is often needed for the analysis and design of electrical devices and for post-processing purposes, as shown below in Section 4.

The above-mentioned advantages make the NFM very attractive to use. Due to these advantages, the NFM has been the method of choice to extend the capabilities of the FEMM software.

### 3. Implementation of the nodal force method in the FEMM software

The nodal force method is implemented in the Finite Element Method Magnetics software using object-oriented programming in C++ [7]. To visualize the computed force, supplementary code is developed and added to the FEMM source code.

First a base ForceAlgorithm class is created which defines the common attributes and member functions of all descendent algorithms [7]. The NodalForce class inherits the ForceAlgorithm characteristics and specifies additional methods related to the implementation of the nodal force algorithm. This enables to easily add other classes such as the VirtualWorkForce in the future.

The ForceAlgorithm class defines a virtual function solve(). This means that all derived algorithm classes from this class must provide their own implementation of the function, thus defining a common pattern of usage. As a child of ForceAlgorithm, the NodalForce class contains the code needed to actually perform the calculation in solve(). The implementation details of the above algorithm are hidden by abstraction in the ForceAlgorithm class.

The class ForceAlgorithm stands at the base of the class hierarchy. It contains methods and attributes which are common to the different algorithms. They include parameters of one finite element such as the magnetic flux density components and magnetic permeabilities along the two axes. A MeshNode array is created containing the element node coordinates. Memory is allocated dynamically in the constructor of ForceAlgorithm for the result returned from solve(). The destructor of ForceAlgorithm cares for memory deallocation.

The NodalForce class is a subclass of ForceAlgorithm and inherits all attributes from the parent class. The forces acting on the three nodes of one finite element are calculated in solve(). For this purpose the Maxwell stress tensor components in (2), the shape function coefficients and the force components in (7) are computed and assigned to variables. Then the method solve() returns the force vectors in the three nodes of the finite element being analyzed.

The invocation of the nodal force algorithm is done in the Block Integrals section of the code.

The Block Integral is applied over the selected integration areas from the model. After the integration areas are selected, invocation of the nodal force algorithm can be done.

The method `add_vector_result` performs the integration in (6). This integration is carried out by summing up the force components along the two axes on the nodes of the selected finite elements. The reference passing avoids duplication of objects in memory while calculating the nodal force.

Due to the extension of FEMM by the nodal force method, the original Block Integrals dialog for invoking the Nodal Force algorithm is modified by adding one more entry (Nodal Force) to the drop-down list. The modified dialog of the extended version of FEMM is shown in Fig. 1.

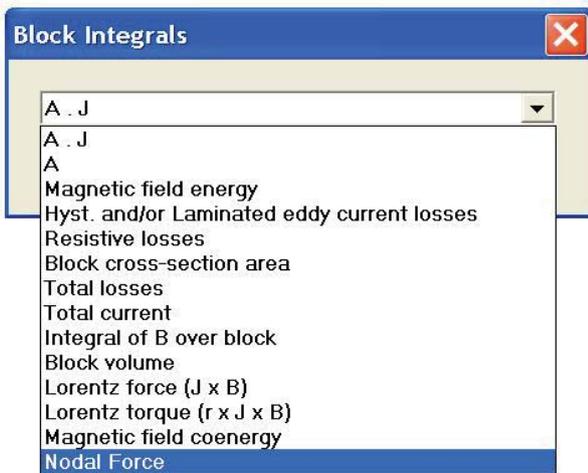


Fig. 1. Modified dialog for the Nodal Force algorithm

As shown in Section 2, an important advantage of the nodal force method is that it computes local force. To utilize this advantage, the visualization capabilities of FEMM are enhanced by developing supplementary code in C++ and adding it to the source code. The original View context menu is modified by adding one more entry (Force vectors) to the drop-down list in Fig. 2.

Fig. 3 shows the entirely new dialog designed for the purposes of force visualization.

The local forces acting on the finite elements nodes are displayed as vectors whose direction coincides with the direction of force. The length of these vectors is obtained by scaling the magnitude of forces using the Choose scale slider at the top of the dialog in Fig. 3. The value in the Maximum length edit box shows the longest vector length when the slider is set to the rightmost position.

The vectors of computed forces are drawn as arrows. As seen in Fig. 3, several options for the head angle, length and colour of arrows are

provided. Applications of the developed visualization enhancements are given in the next section.

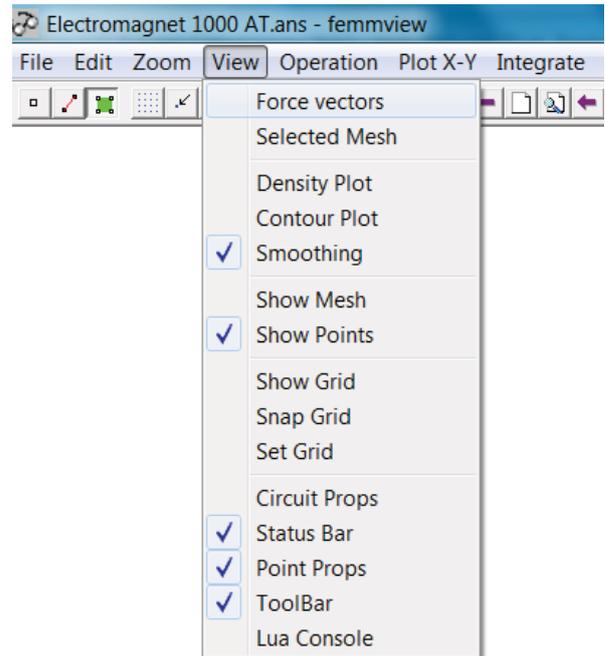


Fig. 2. Modified context menu for the Nodal Force visualization

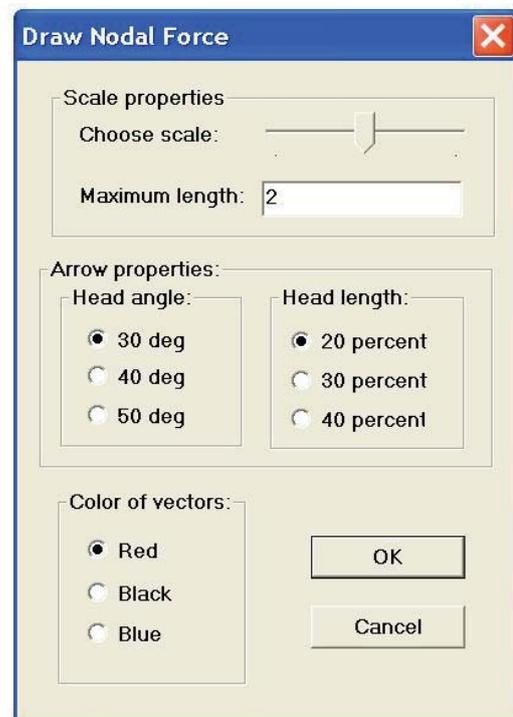


Fig. 3. New dialog for the Nodal Force visualization

The above described extensions to the Finite Element Method Magnetics source code are made according to the Document/View Architecture (MVC) [8].

#### 4. Accuracy validation and visualization by the extended version of FEMM

In this section the accuracy of the extended version of FEMM is validated by comparing the implemented nodal force method with the Maxwell's stress tensor method, available in the conventional FEMM. For that purpose three models are analyzed and their electromagnetic forces are visualized by the new FEMM capabilities.

The first model consists of two copper busbars carrying currents of 100 kA in the same direction [9]. The geometry is shown in Fig. 4. The dimensions are in centimeters. The finite element mesh has 25427 nodes and 50295 triangles.

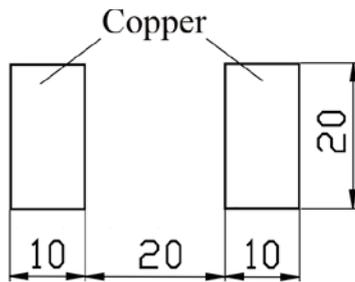


Fig. 4. Geometry of the copper busbars

Table 1 shows the total x-axis electromagnetic forces between the busbars when the currents flow in the same direction. The forces are computed analytically, by the Maxwell stress tensor method and by the nodal force method.

Table 1. Forces between the busbars

Analytical [N]	MSTM [N]	NFM [N]
6333	6328	6336

The vector plot of the local forces acting on the finite element nodes is shown in Fig. 5. The plot is generated using the visualization enhancements to FEMM described in the previous section.

The plot in Fig. 5 confirms the theory that conductors carrying currents of the same directions attract each other. As expected, the nodal forces act only in the current-carrying regions. Due to the nature of the nodal force method, the local force vectors originate from the finite elements nodes.

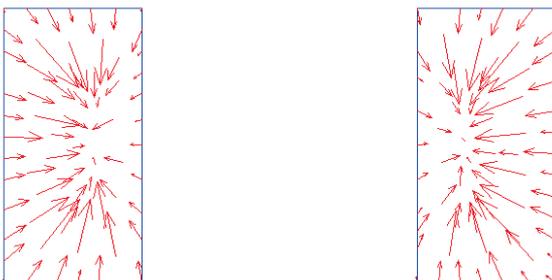


Fig. 5. Vector plot of the forces on the busbars

The second model is a non-linear AlNiCo permanent magnet with steel core [9]. The dimensions in centimeters are shown in Fig. 6. The finite element mesh has 71394 nodes and 141915 triangles.

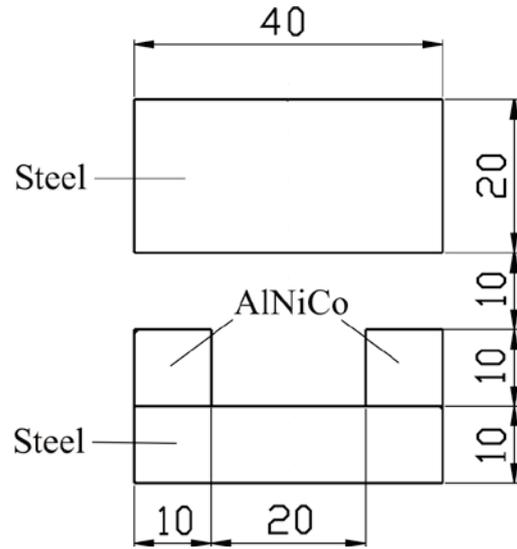


Fig. 6. Geometry of the AlNiCo magnet with steel core

Table 2 shows the total y-axis force acting on the steel armature, computed by the MSTM and NFM. The computed forces by the two methods are very close which confirms the accuracy of the built-in nodal force method.

Table 2. Forces on the steel armature

MSTM [N]	NFM [N]
1290	1293 N

The vector plot of the forces acting on the steel and on the permanent magnet is shown in Fig. 7. The plot is generated using the new visualization capabilities of FEMM.

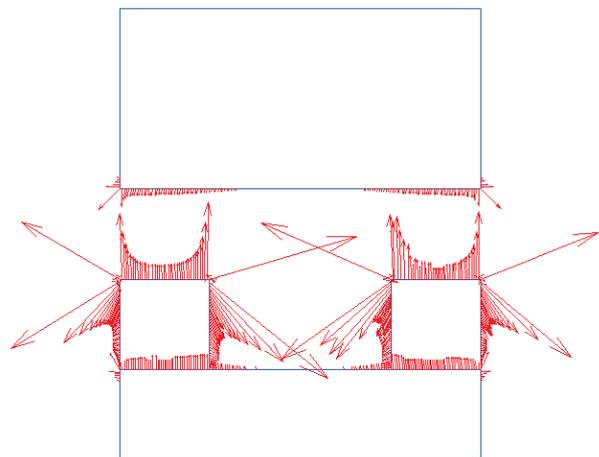
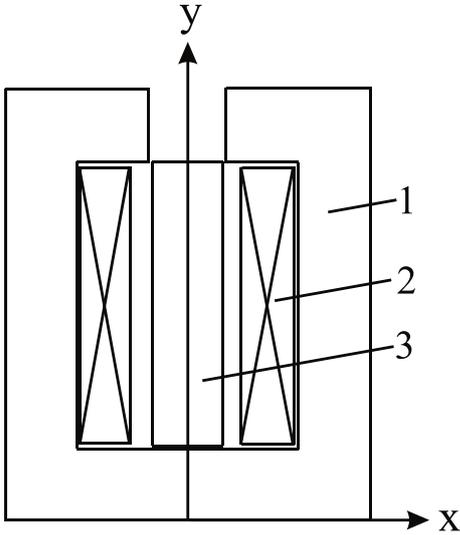


Fig. 7. Vector plot of the forces on the magnet

The third model is a non-linear dc electromagnet. This is a benchmark model used for evaluating the accuracy of the methods for force computation as well as for validation of computer programs [10].

The electromagnet has complex geometry, very small air gaps and uneven magnetic flux distribution. To saturate the steel, the excitation current is varied within a wide range.

The geometry of the model is shown in Fig. 8. The electromagnet is comprised of a steel yoke 1, coil 2 and a central pole 3. The yoke and the central pole are made of steel. The coil has 381 turns and is fed by dc current. The reluctivity curve of steel is given in [10]. To analyze the steel saturation effect, the total current in the coil has values 1000, 2000, 3000, 4000 and 5000 ampere turns. The finite element mesh has 81225 nodes and 162145 first order triangles.



**Fig. 8.** Geometry of the benchmark electromagnet

Table 3 shows the y-axis forces acting on the central pole. They are computed by the newly implemented NFM and the standard MSTM in FEMM. The relative error in force by the NFM and the MSTM is determined by the formula:

$$\varepsilon_r = (F_{\text{NFM}} - F_{\text{MSTM}}) / F_{\text{MSTM}}, \quad (8)$$

where  $F_{\text{NFM}}$  and  $F_{\text{MSTM}}$  are the forces by the NFM and MSTM, respectively.

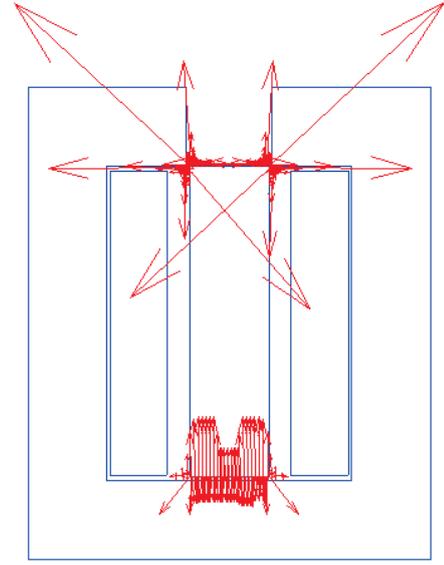
The force by the MSTM is used as reference in (8) instead of the measured values in [10], since the benchmark model analyzed in this paper is two-dimensional.

The absolute values of the relative errors in force are given in Table. 3. They show that the NFM, implemented in FEMM, has excellent accuracy, the maximum relative error in force being less than 1%.

**Table 3.** Y-axis forces acting on the central pole and relative error

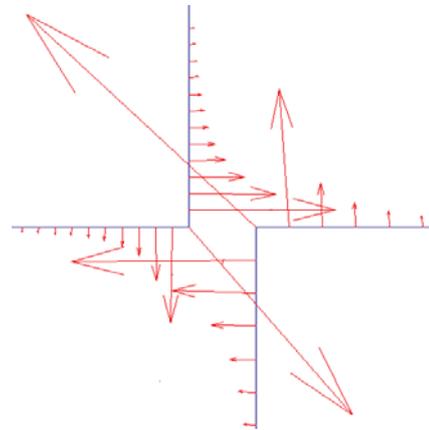
current [A]	NFM [N]	MSTM [N]	$\varepsilon_r$ [%]
1000	324.5	326.9	0.73
2000	1363.6	1369.5	0.43
3000	3170.4	3186.3	0.50
4000	5727.8	5741.5	0.24
5000	8693.8	8721.7	0.32

The vector plot of the nodal forces on the pole and yoke at current 1000 A is shown in Fig. 9. The plot is generated using the visualization enhancements to FEMM.



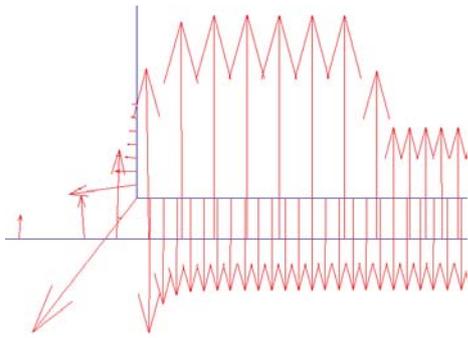
**Fig. 9.** Forces on the pole and yoke

A zoomed-in view of the force distribution in the upper left corner of the pole is given in Fig. 10.



**Fig. 10.** Forces on the upper left corner of pole

Fig. 11 shows the forces on the lower left half of the pole and yoke.



**Fig. 11.** Forces on the lower left half of the pole and yoke

The results from Figures 9, 10 and 11 show that the force has complex distribution and attracts the pole to the yoke. As expected, in the upper part of the pole the force is directed both along the x and y axes, while in the lower part it is mostly along the y axis. Due to the different mesh sizes in the yoke and central pole, the magnitudes of the force vectors on both sides of the air gap in Fig. 11 are different.

The results from Tables 1, 2 and 3 confirm the accuracy of the extended version of the Finite Element Method Magnetics software. There is a very good agreement between the forces computed by the NFM and the MSTM.

## 5. Conclusion

The capabilities of the FEMM software are enhanced by adding a novel method for force computation and visualization - the nodal force method. The NFM is implemented using object-oriented programming in C++.

The extended version of FEMM is validated by comparing the added-on nodal force method with the Maxwell's stress tensor method. The forces of three models are computed and visualized. The results show that the NFM has excellent accuracy.

The visualization capabilities of FEMM are also enhanced by developing supplementary code in C++ and adding it to the source code. Vector plots of the electromagnetic forces of the three models are created. The plots yield reasonable results.

Based on these plots it can be concluded, that nodal forces are localized only on the surface nodes of the steel and on all nodes of the current-carrying coil. This coincides well with the real physical situation where force acts only on the surface of magnetic materials, while the force in electric conductors manifests itself as a volumetric force (Lorentz force).

The extended version of the Finite Element Method Magnetics software can be used for research and design purposes, as well as for teaching numerical methods in electromagnetism and CAD systems at higher schools.

## REFERENCES

1. Meeker, D. (2006). *FEMM 4.2 Magnetostatic tutorial*.
2. Baltzis, K. (2008). The FEMM package: a simple, fast and accurate open source electromagnetic tool in science and engineering. *Journal of Engineering Science and Technology Review*, Vol. 1, pp. 83-89.
3. Spasov, V., Noguchi, S., and Yamashita, H. (2001). Comparison of the methods for electromagnetic force computation by edge elements. *Proceedings of the Third Asian Symposium on Applied Electromagnetics, Hangzhou, China, May 28-30*, pp. 111-114.
4. Spasov, V., Noguchi, S., and Yamashita, H. (2001). Comparative analysis of the force computation methods in the 3D FEM with edge and nodal elements. *Electrical Engineering Research Conference, Kita Kyushu, Japan, SA-01-22, RM-01-90*, pp. 9-13.
5. Spasov, V. (2005). Computation of electromagnetic force by the nodal force method. *XIV-th International Symposium on Electrical Apparatus and Technologies SIELA 2005, Plovdiv, Vol. II*, pp. 139-144.
6. Salon, S. (1995). *Finite element analysis of electrical machines*, 247 p. Kluwer.
7. Preiss, B. (1998). *Data structures and algorithms with object-oriented design patterns in C++*, 688 p. Wiley.
8. Prosiše, J. (2003). *Programming Windows with MFC*, 1376 p. Microsoft Press.
9. Brandisky, K., and Yatcheva, I. (2002). *CAD systems in electromagnetism*, CIELA, 244 p. Sofia.
10. Takahashi, N., Nakata, T., et al. (1994). Investigation of a model to verify software for 3-D static force calculation. *IEEE Transactions on Magnetics*, Vol. 30, No. 5, pp. 3483-3487.

Assoc. Prof. Vasil Spasov, Ph.D.  
Department of Electrical Engineering  
E-mail: [vasilspasov@yahoo.com](mailto:vasilspasov@yahoo.com)

Assoc. Prof. Ivan Kostov, Ph.D.  
Control Systems Department  
E-mail: [ijk@tu-plovdiv.bg](mailto:ijk@tu-plovdiv.bg)

Assistant Prof. Ivan Hadzhiev, Ph.D.  
Department of Electrical Engineering  
E-mail: [hadzhiev\\_tu@abv.bg](mailto:hadzhiev_tu@abv.bg)

Technical University - Sofia  
Branch Plovdiv  
25 Tsanko Dyustabanov Str.