A Parallel Implementation of a Parametric Optimization Environment - Numerical Optimization of an Inductor for Traction Drive Systems -

Uwe Pahner, Kay Hameyer and Ronnie Belmans
Katholieke Universiteit Leuven, Dept. E.E., Div. ELEN, Kardinaal Mercielaan 94, B-3001 Leuven, Belgium

Abstract: Optimum design is defined as a design that is the best possible solution. All design variables are determined simultaneously to satisfy a set of constraints and optimize a set of objectives. Two parametric FE pre-processors and a general purpose optimization environment are presented. Due to its open architecture, finite element as well as analytical models can be implemented. Stochastic algorithms usually require substantially more function evaluations compared to gradient methods, which increases the elapsed computation time. However, the stochastic algorithms feature unmatched simplicity in the setup of an optimization model. A parallel implementation of the Evolution Strategy is presented, which offers one way to reduce the elapsed computation time. An optimization task is discussed to outline the general application range of the developed tools. The optimum design of an inductor used in a traction drive system is described in detail. Special attention is paid to the formulation of the quality function.

Keywords: numerical optimization, stochastic search algorithm, parallelization, finite elements.

I. INTRODUCTION

The development and design of electromagnetic devices reflects a complex process. Originating from an initial idea, the construction runs through different phases. This procedure is terminated when a final concept is selected and considered to be optimized, subject to various targets and constraints. As a whole, the task of the design engineer is to find solutions for technical problems. On the way to the physical and technical product, certain aspects have to be considered. Technological and material depending questions, cost effectiveness and ecological constraints have to be taken into consideration. A cut-set of the mentioned boundary conditions controls the feasibility of the final design. The design process strongly depends on the experience of the engineer and reflects an optimization procedure with often contradicting targets. Therefore, the necessity of a systematic design with engineering tools is obvious. In this paper, solution strategies using modern numerical methods to accelerate the development and to ensure a high standard technical product in an overall design process are discussed.

Simulation and the numerical optimization of electro-magnetic devices is one key to enhance product quality and manufacturing efficiency. Each device has different specifications and thus the goal of the optimization is strongly device dependent. To obtain an optimization tool which is of general applicability, allowing a high number of independent design parameters as well as a simple consideration of constraints, stochastic optimization methods are selected. Stochastic search algorithms as Evolution Strategy, Simulated Annealing and Genetic Algorithms offer all these specifications. The disadvantage of a larger number of quality function evaluations compared to deterministic optimization algorithms is compensated by the simplicity of the implementation of constraints and multiple goal quality functions [1]-[5], as well as the inherent parallelism of these methods. One main objective of the authors is the amplification of the advantages of the selected optimization methods by implementing them into a general purpose tool, which unifies the set-up, test and execution of optimization tasks.

To predict the system behavior of an electromagnetic device and thus to evaluate its quality, field analysis tools are in common use. Analytical, semi analytical [2] and numerical methods can be used. The appropriate choice of an field analysis tool is problem dependent [1]. To simulate electromagnetic fields, the finite element analysis has proven to be a reliable tool for the evaluation of a new design. Combining stochastic optimization algorithms and field simulation techniques into an optimization environment allows the creation of user friendly design tool [1]. Here, a parametric optimization environment is developed to automate the design of electrotechnical devices, allowing the use of analytical as well as semi analytical and numerical methods for the evaluation of the quality function. The optimum design of an inductor used in a traction drive system is described in detail to demonstrate the methodology and practical implementation of the methods used.

II. PARAMETRIC PRE-PROCESSOR

A key requirement for the combination of numerical field analysis tools and optimization algorithms is a pre-processor providing the possibility to parametrize 2D or 3D mathematical models. This includes, apart from the parametrization of the geometry, the parametric definition of material properties, problem defining data and post-processing algorithms. In a first

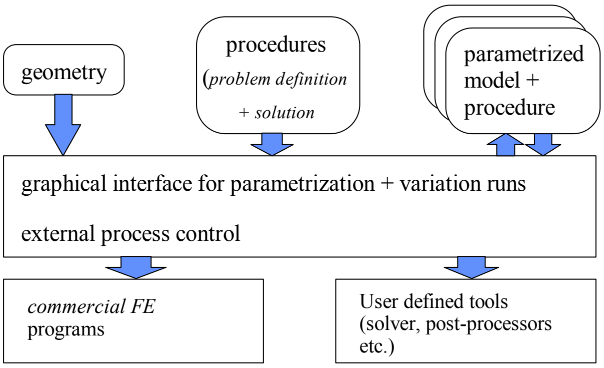

Fig. 1. Structure of the parametric pre-processor implemented in *MATLAB*.

step *MATLAB* has been chosen as the environment to implement an interactive graphical pre-processor (Fig. 1).

This pre-processor is linked to the professional FE-package *MagNet*. Starting from a sketch of the device geometry, the entire analysis procedure for the model is defined. The resulting parametrized sketch file contains all data to describe the single steps of the field analysis, constraints checking and the recommended post-processing algorithms (evaluation of the quality function). Once the analysis procedure of the field model is defined, simple parameter variation can be performed. From within *MATLAB*, the full procedure is controlled by calling external programs, such as the mesh generator, equation system solver and post-processor routines of *MagNet*. The open structure allows the combination of *MagNet* with different analysis tools.

In a second step, a stand-alone, fully interactive and parametrized 2D FE-pre-processor has been developed, which is embedding different solvers and post-processing tools that have been developed in our research unit. However, this pre-processor is linked to the same optimizer as the pre-processor that is implemented in *MATLAB*. This 2D pre-processor includes all features that are expected from a classical FE pre-processor, but additionally provides all tools to set up, test and control optimization tasks, such as:

- the parametric definition of problem types, materials, excitations and boundary conditions,
- the definition and test routines of analysis procedures including different types of solvers and post-processing routines (Several different procedures can be defined with one model.),
- the definition of constraints checks and normalizing factors for the design parameters,
- the setup and execution of parameter variations using beforehand defined analysis procedures,
- the storage of the model and the procedure in a symbolic format,
- the automatic preparation of optimization tasks for a parallel environment using *PVM* (Parallel Virtual Machine, a software that allows the setup of a parallel environment with several computers via a network) [7].

The latter involves the minimization of the necessary data exchange between the different network nodes inside the virtual parallel environment.

III. OPTIMIZATION ENVIRONMENT

The developed optimization environment provides the following features:

- Different optimization algorithms.
- Monitoring of the optimization process at run-time.
- Defined stop and restart procedures in case of problems during execution.
- Handling parametrized procedures provided by the parametric pre-processor.
- Open architecture supporting the optimization of non-FE models.
- Implementation of additional optimization algorithms without changing the whole structure of the environment.

Four stochastic optimization algorithms have been implemented into the optimization environment: Evolution Strategy, Simulated Annealing, a combination of both and a variant of Simulated Annealing. Details can be found in [2]. Typical for stochastic optimization algorithms is the generation of independent sets of parameters within one iteration (Fig. 3).

The optimizer itself is a stand-alone program, which can operate in two modes:

1. Internal optimization: the quality function and the

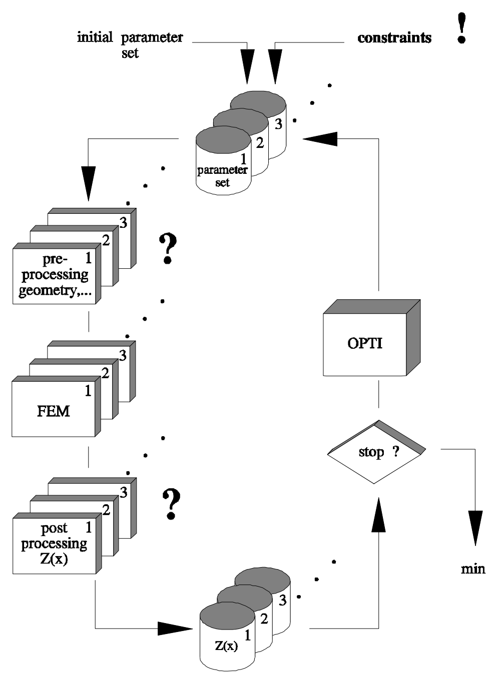

Fig. 3. Simplified flow chart of the optimization process using the external optimizer (OPTI). (The question marks stand for the constraints checking.)

constraints check are compiled and linked into one executable. The function evaluations are sequential, a new set of parameters is created after the previous is analyzed.

2. External optimization (Fig. 3): the optimizer reads the status and quality of the previous function evaluation at startup, and exports a new set of parameters. No information about the type of analysis procedure is given to the optimizer.

This second mode represents the main strength of this optimizer: as no information about the analysis procedure is required for the optimization algorithm itself. This executable can be called from different programs or can bee used even in batch mode. There is no need to re-code the optimizer for a new optimization task. This results in a minimum effort to set up a new, probably totally different optimization problem. This second mode is also used in the parallel implementation of the optimization. The developed optimization environment in *MATLAB* already operates on a higher level (Fig. 4). It calls the optimizer in external mode to generate the new sets of parameters. All the other features allow the fast setup of a new optimization for the different possible analysis procedures:

- *MATLAB* based FE-pre-processor linked with *MagNet*,
- the models defined by the stand-alone pre-processor,
- analytical models (*MATLAB* macro files)
- or semi-analytical analysis (external programs).

Other functions assist the choice of the strategy parameters for the optimization algorithm itself, as well as the definition of the set of start parameters and the stop criteria. In case of an optimization of a finite element model, the parametrized sketch file includes all information required to start the optimization. The environment controls the external process calls for the FE-analysis. Whereas the parametric pre-processor is an interactive graphical tool, the optimization process is entirely automated and can operate as a background process. The optimization can be stopped at any time and restarted from a previous position. This feature has been found very useful in a network environment, when a long lasting optimization should be stopped to allow maintaining the network. The actual progress of the optimization can be monitored graphically. Depending on the optimization algorithm, key data may be visualized together with the variation of all parameters.

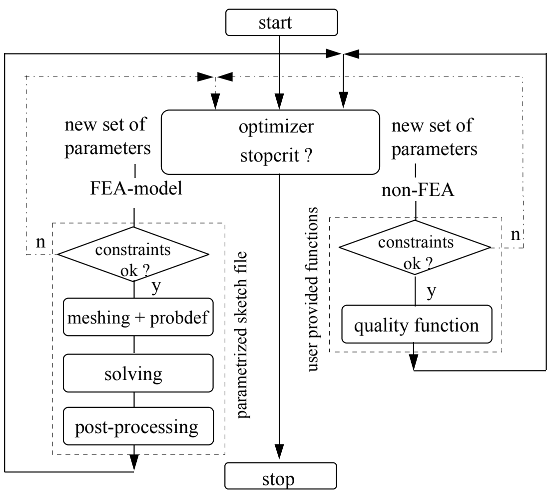

Fig. 4. Structure of the optimization environment.

IV. PARALLEL IMPLEMENTATION OF THE EVOLUTION STRATEGY ALGORITHM

The optimization, as an essential part of the design procedure, must always be evaluated regarding its cost effectiveness. Apart from the costs for appropriate software and hardware, there are two main cost factors:

- Engineering man hours have to be paid for the time of the interactive definition and testing of the optimization problem, and
- the overall computation time that the optimization needs to find the optimum (machine hours).

It is widely discussed, that one of the mayor advantages of stochastic algorithms is their robustness, ease of use and general application range. The setup and testing of new optimization problem (quality function, implementation of constraints) requires much less effort compared to deterministic algorithms. This directly influences the first cost factor. A main argument against the widespread use of stochastic search algorithms is its huge number of necessary function evaluations, which increases the elapsed computation time. This argument, however, will not hold when the cost effectiveness of the optimization is taken into account, as the higher number of function evaluations will only increase the second cost factor, which is negligible against the costs of engineering personal.

The rapid enhancement of computer power, but also the availability of new computer architectures (parallel machines) will help to reduce the total computation time in the future. An implementation using *PVM* is described here (Fig. 5).

PVM provides the software tools and libraries that allow setting up a parallel environment (parallel virtual machine) in an already installed network of heterogeneous computers. From a master process, residing on one of the machines, slave processes can be spawn to the different machines. *PVM* itself provides all the tools for the control and the communication of the single parallel running processes. The actual parallelization of the algorithm, but especially the amount of data exchange between the processes influences the possible down-scaling of the elapsed computation time. Network and machine load due to third party processes play an important role as well.

As shown in Fig. 3, Evolution Strategy is perfectly suited for parallelization, as all generated sets of design parameters in one iteration are independent. Fig. 5 outlines the realized implementation of an Evolution Strategy algorithm. The

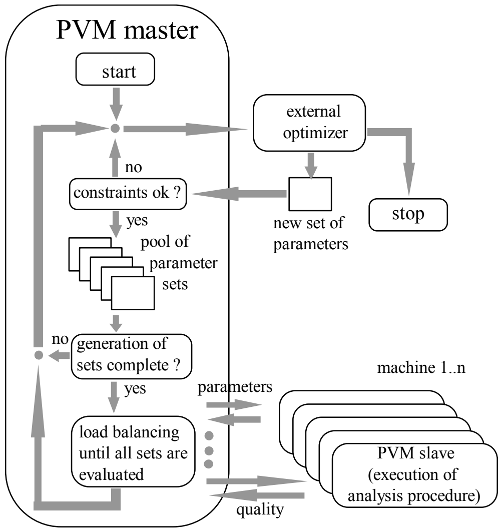

Fig. 5. Parallel implementation of the Evolution Strategy algorithm using *PVM*.

defined procedures for the FE analysis together with the parametric description of the model itself are distributed to the local computers of the parallel environment. Those symbolic descriptions reside local on those machines for the whole optimization process. The execution of the defined procedures for the function evaluation involves no graphics, and can therefore be performed in the background.

During the optimization, the data exchange between the host machine and the slave machines is limited to the updated values of the parameters and the return of the quality and status of the single function evaluation to the master process (the optimization controller). The impact of third party network traffic is therefore reduced to a minimum.

Dynamic load balancing is very important in a network, which consists of heterogeneous machines (in this case several HP 715, 730 and C160). The parameter pool is filled with *n*-sets of parameters, which represent the *n*-design candidates to be evaluated in one iteration of the optimization process. A new set can only be generated when all values of the single qualities of the present iteration are obtained. These processes have to be spawn onto the different machines. Towards the end of each iteration, a situation may occur, were a slower machine could start a function evaluation, which could take longer than the faster machines would need to evaluate the remaining parameter sets in the pool. This would increase the elapsed optimization time unnecessarily. The implementation of dynamic load balancing in the master program allows an effective use of machines with different performance characteristics or with temporarily reduced performance due to third party processes (processes of other users on the local machines). The impact of the latter can not be neglected in a real life network situation.

Two load balancing schemes are combined. As the theoretical performance of the heterogeneous computers in the network is known, the optimization procedure is started with static load balancing. New function evaluations are generally started on the different hosts when the previous process on these hosts have been finished. Before statistically sufficient data about the different computation times per machine are collected, the priority of the machines to receive a new process depends on the their theoretical performance. The computation times of all single function evaluations are recorded during the process of the optimization. The actual performance due to third party network traffic and processes can now be taken into account to ensure the optimal usage of the different machines with regard to a minimized elapsed optimization time.

The achievable down-scaling of the elapsed computation time for a complete optimization depends largely on the performance and the load of the different machines incorporated into the parallel environment. Tests have shown that in a setup with equal machines (HP 9000/715), the down-scaling is almost linearly depending on the number of machines included. This can only be achieved due to the minimized data exchange with the master process. The exchanged data packages are of the sizes of some bytes, whereas the FE data structures can reach several Mbytes during the evaluation of the quality function.

V. OPTIMIZATION OF AN INDUCTOR - AN ANALYTICAL APPROACH

As an example the design optimization of an inductor used in a traction drive system is chosen (Fig. 6). Apart from the desired electric and magnetic characteristics and properties of the inductor, a minimum weight is demanded. Here, the inductor must have an inductance L of 3 *mH* up to a maximum current of 1350 *A*.

The current density J in the copper windings must not exceed 6 A/mm^2. The maximum dimensions for the inductor are given geometrical constraints (Table I).

TABLE I
SET OF DESIGN PARAMETER AND CONSTRAINTS

set of design parameters:		constraints:		
b	yoke width	$d+b_w$	<	500 *mm* (depth)
d	yoke thickness	$2*(b+b_w)$	<	750 *mm* (width)
b_w	window width	$2*b+h_w$	<	660 *mm* (height)
h_w	window height	g_0	<	$h_w/2$
g_0	total air gap	J	<	6 A/mm^2
N	number of turns	B_{max}	<	1.5 *T* (no saturation)

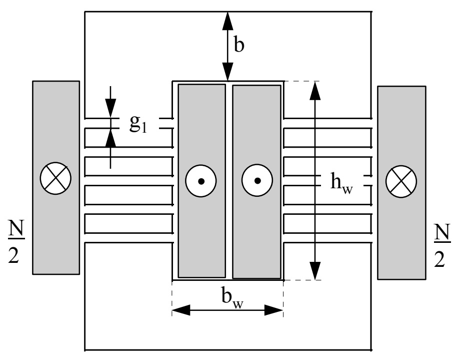

Fig. 6. Geometry and design parameter of the inductor example.

The total air gap is subdivided into multiple gaps with a length less than 1/6 of *b* and *d* respectively, to minimize the leakage flux. 1/6 is empirically chosen, following the fact that a higher number of small air gaps leads to less leakage flux then one large air gap. An (4/4, 20)-Evolution Strategy was chosen to tune the design parameters during the optimization [2]. Particular attention must be paid to the formulation of the quality function *q*. Two different formulations have been tested. Formulation (1) includes a penalty term in case the saturation constraint is violated:

$$q = \frac{m_i}{1000} + \frac{\left|L_{given} - L_i\right|}{L_{given}} + penalty \qquad (1)$$

with

$$penalty = \begin{cases} 0 & : \; if \quad B_i \le B_{\max} \\ \dfrac{\left|\mathrm{B}_{\max} - B_i\right|}{B_{\max}} & : \; if \quad B_i > B_{\max} \end{cases}$$

Here, m_i is the weight of the inductor, L_{given} is the specified inductance of 3 *mH* and B_i is the maximum flux density inside the iron parts. Index *i* indicates the quantities calculated from the actual set of design parameter. The set of parameter is rejected in the case geometrical constraints are violated and new sets of design variables are generated until they meet the constraints [2]. Formulation (2) features no penalty term for the violation of the saturation criterion. The set of parameters is rejected before the function evaluation if geometrical constraints are violated The main difference to formulation (1) follows during the function evaluation. If the flux density values are too high, the parameter set is rejected and a new set is generated and analyzed. The quality function is simplified to:

(2)

To maintain the desired value of the inductance, formulation (2) calculates the number of windings as a function of the given inductance. This second formulation reduces the search space for the optimization algorithm, and consequently inherits the possibility of not finding the global optimum. The optimization process for the inductor problem is started with an initial set of parameters violating the constraints. Fig. 7 and 8 show the change of step length and the rate of convergence during optimization process for the inductor using formulation (1).

One of the first accepted parameter sets describes an inductor with a total weight of 650 *kg* (Fig. 9). Using the Evolution Strategy, the step length of the parameter variation is used as stopping criterion. After the optimization, the inductance is calculated to 3.001 *mH* and the flux density and the current density do not exceed the maximum values. Applying formulation (1), the final weight is 349 *kg* (Fig. 10).

The optimal solution found using formulation (2) (Fig. 11) does not match the results obtained by formulation (1). The significance of an appropriate choice for the quality function is obvious.

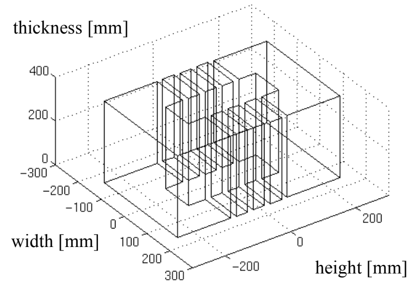

Fig. 11. Optimal inductor design using quality function (2) with a weight of 472 kg.

Fig 7. Step length for the best set of parameters per iteration.

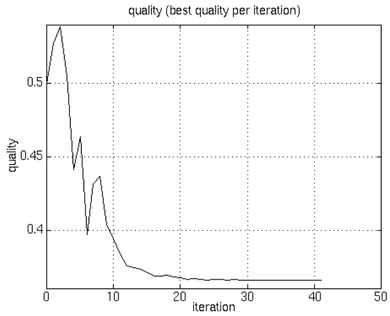

Fig. 8. Quality versus iteration count during optimization

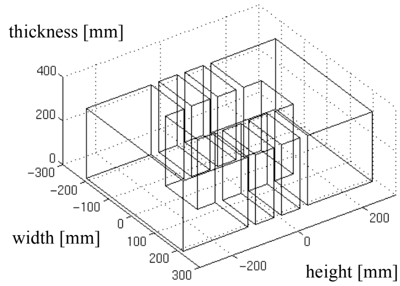

Fig. 9. Initial design of the inductor with a weight of 650 kg

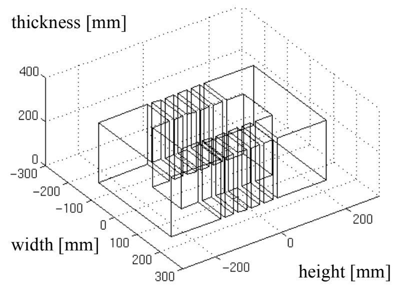

Fig. 10. Optimal design of the inductor using formulation (1) with a weight of 349 kg.

VI. CONCLUSION

A parametrized environment for the optimization of electromagnetic devices has been developed. The emphasis

ne development of a user friendly tool which is both, FE and analytical models. The optimization n inductor demonstrates the open architecture of ent. The optimization environment will be ture, incorporating more optimization algorithms tial Evolution). Other central points in further already started activities are the automatic selection of the strategy parameters defining the optimization algorithm and the further parallelization of the procedures (evolution strategy migration schemes).

VII. ACKNOWLEDGMENT

The authors are grateful to the Belgian Nationaal Fonds voor Wetenschappelijk Onderzoek for its financial support of this work and the Belgian Ministry of Scientific Research for granting the IUAP No. P4/20 on Coupled Problems in Electromagnetic Systems.

VIII. REFERENCES

[1] K. Hameyer and R. Belmans, "Design and optimisation of electromagnetic devices," *Proceedings of the 10th International Conference on Engineering Design*, Praha, Czech Republic, August 22-24, 1995, pp.816-821.

[2] K. Hameyer and R. Hanitsch, "Numerical optimization of the electromagnetic field by stochastic search and MEC-model," *IEEE Transactions on Magnetics*, vol. 30, no. 5, 1994, pp. 3431-3434.

[3] P. Alotto, A.V. Kuntsevitch, Ch. Mangele, G. Molinari, C. Paul, K. Preis, M. Repetto and K. R. Richter, "Multiobjective Optimization in Magnetostatics: A Proposal for a Benchmark Problem," *IEEE Transactions on Magnetics*, vol. 32, no.3, 1996, pp. 1238-1241.

[4] G. Schönwetter, Ch. Mangele, K. Preis, Ch. Paul, W. Renhart and K.R.Richter, "Optimization of SMES Solenoids with regard to their Stray Fields," *IEEE Transactions on Magnetics*, vol.31, no. 3, 1995.

[5] A. Gottvald, K. Preis, Ch. Mangele and A. Savini, "Global optimization methods for computational electromagnetics," *IEEE Transactions on Magnetics*, vol. 28, no. 2, 1992, pp. 1537-1540.

[6] M. Balachandran, *Knowledge-Based Optimum Design*, Computational Mechanics Publications, Topics in Engineering Vol. 10, ed. by C.A. Brebbia & J.J. Connor, Southampton UK and Boston USA, 1996.

[7] A. Geist, A. Beguelin, J. Dongorra, W. Jiang, R. Manchek and V. Sunderam, *PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, Cambridge, Massachusetts, London, England, 1994.

IX. BIOGRAPHY

Uwe Pahner was born in Glauchau/Sachsen (East Germany), on July 5, 1965. He studied at the Technical University Chemnitz (Karl-Marx-Stadt), Germany, from 1987-1992, were he received his M.S. degree in electrical engineering. In 1992 he joined the Division Electrical Energy at the Katholieke Universiteit Leuven, Belgium, were he is currently working on his Ph.D. degree. His research interests include the numerical optimization of electromagnetic energy transducers and its combination with numerical field computations, the design of permanent magnet machines and actuators.

Kay Hameyer (M' 1995) received the M.S. degree in electrical engineering in 1986 from University of Hannover, Germany. He received the Ph.D. degree from University of Technology Berlin, Germany, 1992.

From 1986 to 1988 he worked with the Robert Bosch GmbH in Stuttgart, Germany, as a design engineer for permanent magnet servo motors. In 1988 he became a member of the staff at the University of Technology Berlin, Germany. From November to December 1992 he was a visiting professor at the COPPE Universidade Fderal do Rio de Janeiro, Brazil, teaching electrical machine design. In the frame of a collaboration with the TU Berlin, he was in June 1993 a visiting professor at the Université de Batna, Algeria. Beginning in 1993 he was a scientific consultant working on several industrial projects. From 1993 to March 1994, he held a HCM-CEAM fellowship financed by the European Community at the Katholieke Universiteit Leuven, Belgium. Currently he is professor for numerical field computations with the K.U.Leuven and a senior researcher with the F.W.O.-V. in Belgium, teaching CAE in electrical engineering and electrical machines. His research interests are numerical field computation, the design of electrical machines, in particular permanent magnet excited machines and numerical optimization strategies.

Dr. Hameyer is member of the International Compumag Society.

Ronnie J.M.Belmans (S'77-M'84-SM'89) received the M.S. degree in electrical engineering in 1979 and the Ph.D. in 1984, both from the Katholieke Universiteit Leuven, Belgium, the special Doctorate in 1989 and the Habilitierung in 1993, both from the RWTH Aachen, Germany.

Currently, he is a full professor with the K.U. Leuven, teaching electrical machines and CAD in magnetics. His research interests include electrical machine design (permanent magnet and induction machines), computer aided engineering and vibrations and audible noises in electrical machines. He was the director of the NATO Advanced Research Workshop on Vibrations and Audible Noise in Alternating Current Machines (August 1986). He was with the Laboratory for Electrical Machines of the RWTH Aachen, Germany, as a Von Humboldt Fellow (October 1988-September 1989). From October 1989 to September 1990, he was visiting professor at the McMaster University, Hamilton, ON., Canada. He obtained the chair of the Anglo-Belgian Society at the London University for the year 1995-1996.

Dr. Belmans is a member of the IEE (U.K.), the International Compumag Society and the Koninklijke Vlaamse Ingenieursvereniging (KVIV).