Optimization of Planar Gradient Coil Systems for a Mobile Magnetic Resonance Device by Genetic Algorithms Using Object-Oriented Design Techniques

Hartmut Popella and Gerhard Henneberger
RWTH Aachen, Department of Electrical Machines
Schinkelstraße 4, D-52056 Aachen, Germany
hartmut.popella@iem.rwth-aachen.de

ABSTRACT

This paper presents the application of an object-oriented environment for the computation of a surface gradient coil system for magnetic resonance imaging (MRI). MRI requires spatial information of the examined sample. This information is provided by a variation of the static main magnetic field B_0 using x–, y– and z–gradient coils. A numerical optimization process must be developed in order to fulfill the requirements concerning gradient field strength and linearity. An object-oriented software implementation with the aid of UML notation simplifies the development of such a combined calculation and optimization tool by identification of the system's interacting objects.

Keywords

Application of UML notation, planar surface gradient coil system, MRI, genetic optimization

1. INTRODUCTION

A planar surface gradient coil system consists of several long wire pairs which form a current loop producing the following z-directed gradient field (y–gradient coil pair):

$$B_z^{(i)}(y,z) = \frac{\mu_0 \cdot n_i \cdot I}{2\pi} \left\{ \frac{y+b_i}{(y+b_i)^2+(z-d_i)^2} + \frac{y+b_i}{(y+b_i)^2+(z+d_i)^2} \right\} \quad (1)$$

$$B_z^{Ges} = \sum_{i=1}^{N} B_z^{(i)}(y,z) \quad (2)$$

The linearity of the gradient fields is measured by the constancy of the gradients, defined as:

$$G_x = \frac{\partial B_z}{\partial x}, \quad G_y = \frac{\partial B_z}{\partial y}, \quad G_z = \frac{\partial B_z}{\partial z} \quad (3)$$

A genetic strategy based on the Elitest model is used in order to find the number of coil pairs N, their location b_i/d_i, the number of coil windings n_i and the impressed current I by minimization of the object function [3], [6]:

$$\varepsilon = \iiint \left| G_y^{Grad}(\underline{x}) - G_y^{desired}(\underline{x}) \right| W(\underline{x}) dV \quad (4)$$

The theory of gradient coils and MR imaging is described in [1] and [4]. This paper focuses on the application of the object-oriented approach in order to realize a complex software system which can be divided into a genetic optimization tool and a tool for the calculation of the gradient coil system. The consistent implementation of object encapsulation allows the use of the genetic optimizer for other tasks than gradient coil analysis, e.g. multidimensional function analysis [7].

2. OBJECT-ORIENTED ANALYSIS OF GRADIENT COIL SYSTEMS

A gradient coil system contains one or several gradient coil pairs. The base class `GradientCoil` defines the interface for a gradient coil pair's behaviour which is the calculation of flux density $B^{(i)}_{zGrad_{x,y,z}}$ (1) and field gradient $G^{(i)}_{x,y,z}$ (3). Therefore there are specializations of `GradientCoil` for the x–, y– and z–direction. The class `GradientCoilSystem` implements the computation of the entire field and its derivation according to (2). Now the computated gradient flux density distribution must be compared to the desired flux density and weighted by a weighting function. In order to consider different weighting factors and requirements to the desired field according to its spatial direction the respective classes must be specializations of `WeightingFunction` and `B0_Desired`. For the 3D-integration of the object function ε a set of numerical integrators like Simpson, Rhomberg, Gaussian and adaptive integration as well as a combination of these methods have been implemented.

3. OBJECT-ORIENTED DESIGN OF GRADIENT COIL SYSTEMS

The main requirement of OOD is the linkage between gradient coil systems and the genetic optimizer without loss of generality for other optimization tasks. Fig. 1 and 2 present the static class diagrams for computation of the system's field strength and the required error/object function [2], [5]. The connection between the objects of the gradient coil system and the genetic optimizer is realized by a composition. Genetic optimizers work on chromosomes. Optimization performance is reached by recombination of the genetic operators crossover and mutation which are applied to certain chromosomes and their genes. A `GradientCoil` is a part of one `ChromosomGradientCoil` and aggregates at least one gene. Those genes are the optimization parameters like the number of coil windings and their physical location. One `GradientCoilSystem` associates one or more chromosomes.

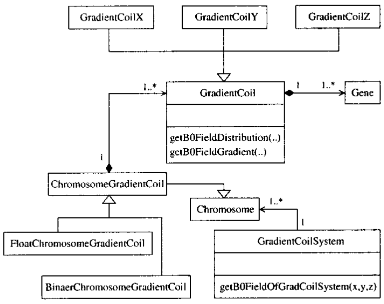

Figure 1: Static class diagram for the computation of the field distribution of the gradient coil system

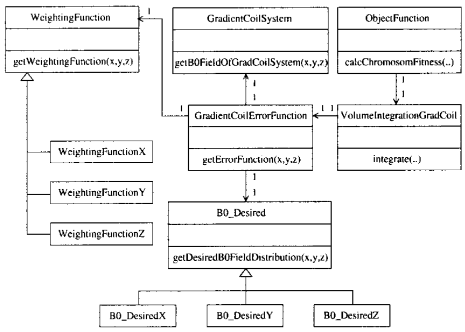

Figure 2: Static class diagram for the computation of the error function

4. COUPLING OF GRADIENT COIL SYSTEM'S COMPUTATION AND GENETIC OPTIMIZATION

The optimization algorithm requires a fitness value for each gradient coil system in order to find out the most suitable one for the next iteration step. So the genetic algorithm uses the method `calcChromosomeFitness(..)` of class `ObjectFunction` which means an integration of the error function for the considered chromosome respectively gradient coil system. In case of multidimensional function analysis this method would deliver the result of the examined function for the set of actual chromosomes and their genes. A selection criterion of the algorithm distinguishes the chromosomes of the actual iteration step according to their fitness. The generalization of class `GeneticAlgo` allows the realization of different selection strategies, Fig. 3. The object-oriented approach provides the simple implementation of other stochastic and non-stochastic optimization techniques in addition to genetic algorithms or even in combination. *C++* has been chosen for the software implementation.

5. OPTIMIZATION RESULTS

The genetic optimizer converges after 2000 iteration steps for a desired field gradient of about $G_y^{desired}(\underline{x}) = 0.5\frac{mT}{cm}$. After 2000 computations and parameter variations the resulting flux density nearly fits the desired one. Finally the optimizer calculated $N = 2$ coil pairs for a y–gradient coil system, their location b_i/d_i as well as the number of coil windings n_i for a current of about $I = 30\ A$.

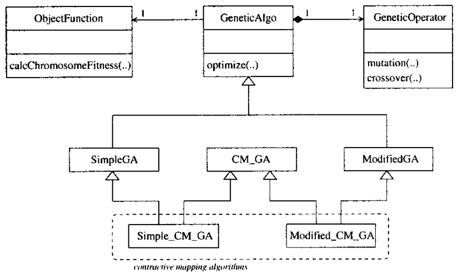

Figure 3: Generalization of the class `GeneticAlgo` for the use of different genetic algorithms

6. CONCLUSION

In this paper an effective application of the object-oriented approach has been depicted in order to construct a planar surface gradient coil system. During the analysis process real objects like the single gradient coil pairs, the entire gradient coil system, chromosomes and genes of the optimizer, their special properties and their typical behaviour have been identified and mapped to objects in the software design process. Attributes and methods reflect the characteristics of the objects. Using UML notation in order to describe the interaction of the objects of such a complex technical system helps to implement stable and less error-prone code and to keep the overwiew at any development stage.

7. REFERENCES

[1] Z. H. Cho, J. H. Yi: Planar Surface Gradient Coil, Concepts in Magnetic Resonance, Vol. 7(2), pp. 95-114, 1995

[2] M. Fowler, K. Scott: UML Distilled, Addison Wesley Longman, Massachusetts, 1999

[3] D. E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Massachusetts, 1989

[4] J. Jin: Electromagnetic Analysis and Design in Magnetic Resonance Imaging, CRC Press, London, 1999

[5] C. Larman: Applying UML and Patterns, Prentice Hall, New Jersey, 1998

[6] Z. Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin, 1999

[7] H. Popella: Object-Oriented Genetic Algorithms for Two-dimensional Design Optimization of the Magnetic Circuit of a Mobile Magnetic Resonance Device, ISEM, Tokyo, 2001