The optimisation of electromagnetic devices using a combined finite element/neural network approach with on-line training

J. Seguin, F. Dandurand and D.A. Lowther
Electrical and Computer Engineering Department, McGill University, Montreal, Quebec, Canada, and
J.K. Sykulski
Electrical Engineering Department, Southampton University, Southampton, UK

Keywords *Design, Electromagnetics, Neural networks, Optimization, Sensitivity evaluation*

Abstract *The paper presents a novel method of utilising neural networks for optimisation systems. First, a conventional magnetic circuit model of the device is developed to create a set of sensitivity rules which guide the optimisation. The rules are coded in a knowledge-based neural network. Second, an error network is developed to correct the approximations inherent in the magnetic circuit approach and this combines with the first network to generate realistic outputs. Finally, the error network can be trained on-line with a finite element system. Over time, the network replaces the finite element analysis, thus speeding up the optimisation process.*

Introduction

An optimisation system is fundamental to the design of an electromagnetic device. Such a system explores the design space in order to find a set of parameters which most nearly meet the specifications without violating a set of imposed constraints such as size, power consumption, cost, etc. The relationships between the physical structure and the performance of a device are described by a hypersurface in the design space and the conventional method of exploring this space is based on an algebraic or numerical (often finite element) analysis system which can evaluate the performance given a specific set of input parameter values. However, such an approach is both expensive and slow. Several recent publications (Arkadan and Chen, 1994; Ebner *et al.*, 1998; Ratner *et al.*, 1996) have demonstrated that the performance of a specific device can be modelled by using a neural network. In a sense, the network learns the shape of the hypersurface and provides a fast evaluation of any point in it.

In previous publications, the neural network has been trained in a batch mode, prior to the optimisation process – essentially "off-line". It is the purpose

This work was supported in part by a grant from the Natural Science and Engineering Research Council of Canada.

COMPEL – The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Vol. 18 No. 3, 1999, pp. 266-274.

of this paper to describe a system which can provide "on-line" training, i.e. a network which is capable of learning and modifying its behaviour as it is used. Such a network has major benefits over a static system in that it can handle a large number of variations of a device and track developments in design related to material changes and manufacturing processes.

A diagram of the proposed system is shown in Figure 1. This differs from a conventional system in that the numerical analysis system and the neural network exist in parallel and data can flow either way from the device model to determine the performance parameters. Each time a set of performance parameters is generated, the data are fed back to provide a new training set for the neural network. Initially, as in previously proposed systems, the network is trained off-line on a device typical of the class of problems to be handled. The decision on which approach to take to generate the performance parameters is made within the device model by an intelligent system which contains a description of the current capabilities of the neural network and relates these to the problem being considered.

In fact, the neural network component of the architecture shown in Figure 1 consists of two parts. The first is intended to produce the actual values of the parameters for the specified device in a manner similar to that described in Arkadan and Chen (1994) and Ratner *et al.* (1996), the second part indicates the sensitivity of the device to changes in the inputs. This latter information is then used to guide the optimiser. In addition to the online training aspect of the work, the sensitivity prediction is a novel contribution of this work and it is thus intended to concentrate on this aspect within this paper. The performance prediction and the control and decision-making structure will be left for future papers.

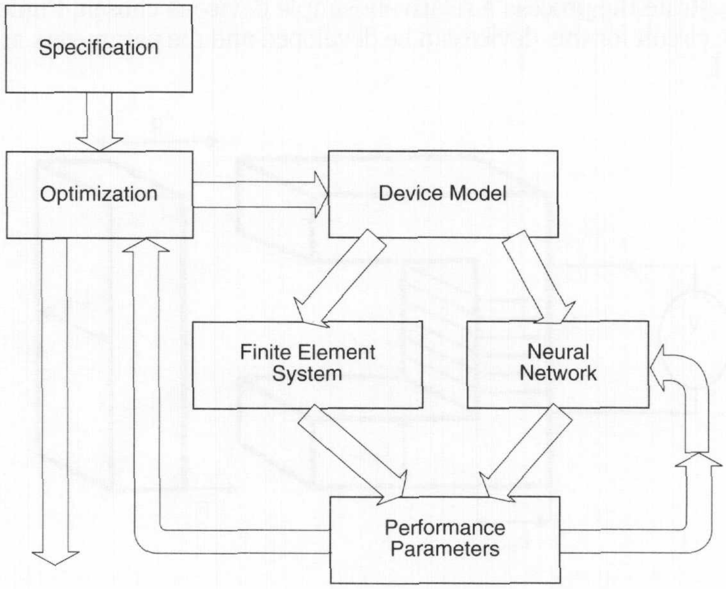

Figure 1. Design process using on-line neural network

The sensitivity prediction is based around a knowledge-based network which implements a set of simple rules derived from a magnetic circuit. This is then corrected by the addition of an error prediction network trained on numerical examples.

A knowledge-based artificial neural network

The performance of an electromagnetic device can be predicted by a range of analytical models, starting with simple magnetic circuits which provide a one-dimensional, lumped parameter model of the device, through specialised algebraic models which can provide closed form solutions for particular aspects of a structure, e.g. the end turn leakage inductance of a machine winding, to full three-dimensional numerical systems. The cost of determining the performance in terms of the time taken increases with the complexity of the model, thus resulting in slower turnaround times for design modifications. Ideally, the designer would like a very fast turnaround time with an accurate modelling tool so that a better feel for the design space can be achieved. The knowledge-based artificial neural network is the first step in this process. Unlike a traditional network, this system is designed to encode simple magnetic circuit models for a particular class of devices. The goal of the network is to provide information which is of use to an optimisation system, i.e. to indicate in which direction to move in order to achieve an optimal design. Thus the network outputs the sensitivity of the design to various input changes. The sensitivities are expressed in the form of rules derived from a simple magnetic circuit representation. The neural network is then designed to implement the rules directly (in effect it bypasses the need to develop an inferencing mechanism for working with the knowledge derived from the model). This system has been described in Dandurand and Lowther (1998).

To illustrate the process a relatively simple device is chosen, Figure 2. The magnetic circuit for this device can be developed and the parameters are shown in Table I.

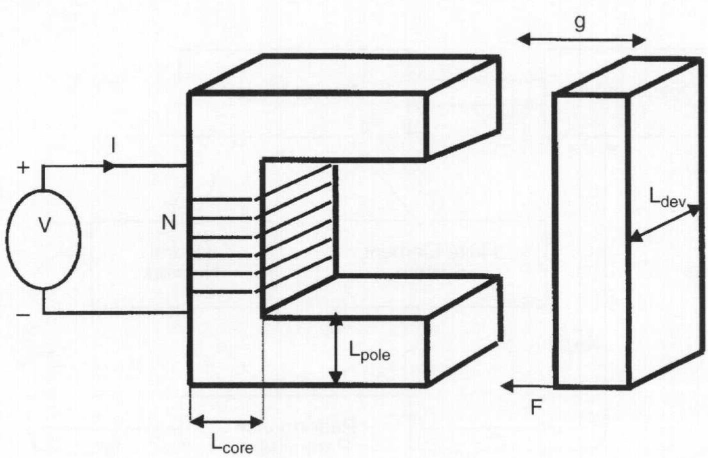

Figure 2.
A simple C-core actuator

Parameter	Description
Structural	
g	Air gap between core and armature
N	Number of turns of the coil
V	Voltage applied to the coil
l_{core}	Length of the core
a_{wire}	Area of the wire used for the coil
l_{pole}	Length of the pole
l_{dev}	Length of the device in the Z plane
Design	
F	Force
I	Current in the coil
R	Resistance of the coil

Table I. The basic set of parameters

Table II shows a basic equation set derived from the magnetic circuit. These equations are next developed to create a set of rules which can be used to control the direction in which the optimization process moves by describing the sensitivity of the device performance to changes in the structural parameters.

For example, the length of the core (L_{core}) affects R according to the following equation:

$$\frac{\partial R}{\partial L_{core}} = \frac{2\rho * N}{a_{wire}} \tag{1}$$

Using this approach, 14 partial differential equations are derived which form the basis of a set of trend rules and, after simplification, a set of 21 symbolic rules acting on 20 input symbols giving 21 output results. The input symbols come from the seven structural parameters plus the force, current and resistance, see Table II, and each parameter has two inputs – "too big" and a "too small". The 21 output symbols consist of three for each structural parameter – increase, decrease and conflict (i.e. increase and decrease simultaneously). Typically, these rules are of the form:

If (R is too small) then (increase L_{core})

If (R is too large) then (decrease L_{core})

Conventionally, these rules would be entered into a rule based system and inferences would be made based on the current state of the system, which would be used to guide the optimisation. However, it is relatively simple to

Equation set of the C-core actuator
$R = 2^* \rho^* N^*(L_{core} + L_{dev})/a_{wire}$
$I = V^* a_{wire}/(2^* \rho^* N^*(L_{core} + L_{dev}))$
$F = \mu_0^* V^{2*} a_{wire}{}^{2*} L_{pole}{}^* L_{dev}/(32^* \rho^{2*} g^{2*}(L_{core} + L_{dev})^2)$

Table II. Equations derived from a magnetic circuit translate the rule structure into a feedforward neural network which produces outputs which directly drive the optimiser.

Once a neural network has been created to represent the magnetic circuit model, it can then be retrained using data provided by a numerical analysis, which removes the assumptions built into the magnetic circuit structure such as linearity of magnetic materials and a lack of fringing and leakage fields. For the training runs, 2,256 sets of training data were generated by running a finite element analysis system for random points in the training space.

However, the retraining process was not as successful as had been hoped (the minimum error achievable was 4.2 per cent) and it was shown in Dandurand and Lowther (1998) that just as effective a result could be achieved by training up a feedforward neural network from outputs of an analysis package alone. The only benefit of the knowledge-based system was that it retained a large percentage of its original, built-in, knowledge, while the feedforward system did not learn the original rules as effectively.

An error correcting network

In general, the effects of non-linearity and leakage are critical in predicting the performance of a magnetic device but are, really, local perturbations on the underlying magnetic circuit structure. Thus, an alternative route to achieving a fast and accurate prediction of the device performance is to measure the error between the magnetic circuit prediction and the numerical analysis. This error can be determined on-line and can be learned by a second neural network operating in concert with the knowledge-based system, Figure 3.

In order to achieve this, the error correcting network needs to have the capability to correct the error "locally" within the design space and a radial basis function network has been considered. In a radial basis function system (Haykin, 1999), each neuron compares the input vector to a weight vector and computes the distance between the two. The closer they are, the higher the output of the neuron and the output takes the form of a Gaussian distribution around the weight vector point. By modifying the width of the function, the

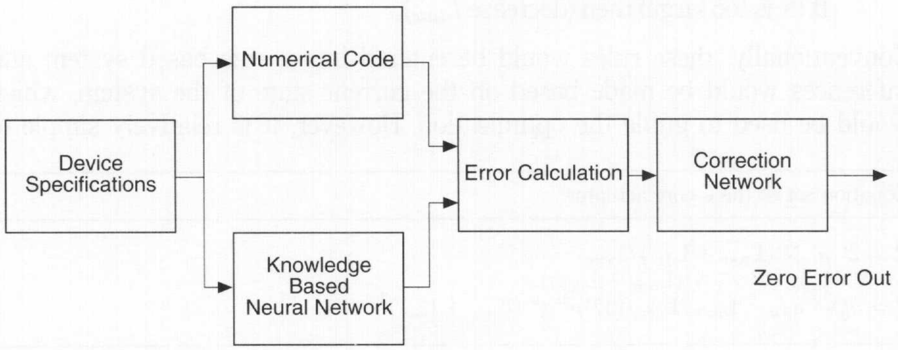

Figure 3.
Training the error correction network

effect of the neuron can be very local, i.e. it would respond only to a very particular input vector structure, or it could be spread out to cover a large part of the search space.

The errors due to non-linearity and fringing being considered above are, in effect, local errors within the search space and thus the radial basis function system is well suited to modelling them.

Radial basis function neurons

As with the standard neural network, a radial basis function network is a fully interconnected system consisting of an input layer, a hidden layer (the radial basis function neurons) and an output layer. The activations of the hidden layer neurons are determined using a radial basis function which has the form given in equation (2).

$$f(x) = e^{-(\sigma x)^2} \tag{2}$$

Figure 4 shows the shape of the activation function and equation (3) describes the output of a neuron with m inputs.

$$O_N = \frac{\sum_{i=1}^{m} (x_i - w_i)^2}{2\sigma^2} \tag{3}$$

Thus the neuron measures the distance between the input vector and the weight vector and the smaller this is, the larger the output of the neuron. The effect of the neuron can be controlled by changing the width of the basis function. The system is trained, i.e. the weights are found, by solving the system of equations derived from the input training set and the desired outputs. In order to avoid the system matrix becoming singular, i.e. two

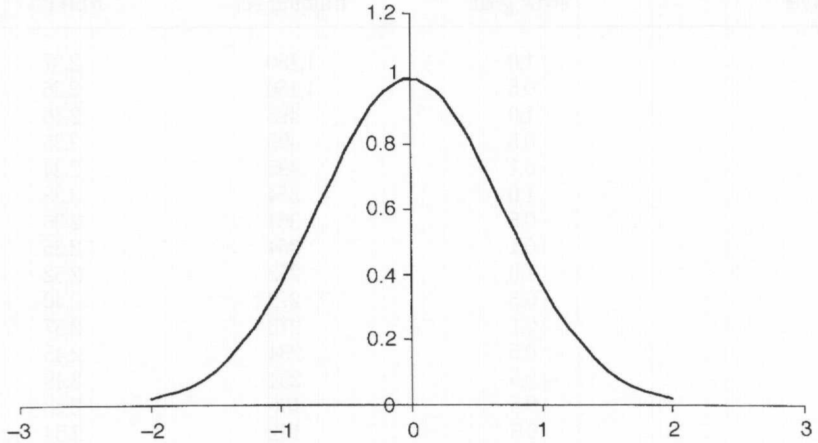

Figure 4. Radial basis neuron activation function

identical inputs, a small amount of white noise is added to the input. This, of course, has some effect on the performance of the network. In the results described below, different amounts of noise were added for each test.

Figure 5 illustrates the use of the two networks in parallel to predict the device structure needed to achieve the desired performance.

Tests and results

The performance in terms of training and accuracy of prediction of a radial basis function network was considered for the simple C-core actuator structure, Figure 2.

A series of tests were performed to examine the performance of the radial basis function neural network. The goal was to minimise the error as the device was driven into saturation and the fringing and non-linearity effects became more important. Table III shows the output error results as a function of the number of radial basis function neurons used, the size of the training set and

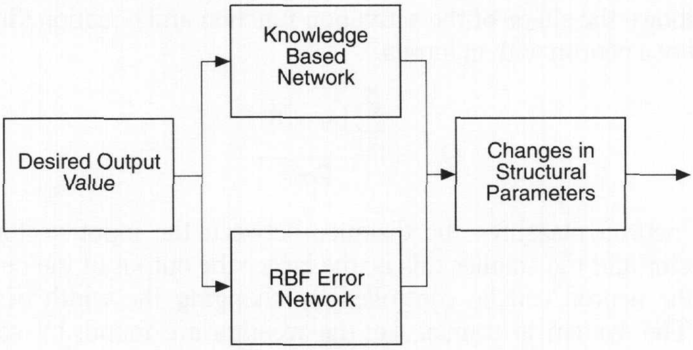

Figure 5.
Architecture of system in use

Table III.
A comparison of network size and taining set with output error

Number of neurons in rbf layer	Sum-square error goal	Size of training set	Error on testing set (rbf) (%)
87	1.0	1,150	2.37
114	0.5	1,150	2.36
61	1.0	495	2.40
79	0.5	495	2.35
145	0.1	495	2.34
56	1.0	364	2.38
74	0.5	364	2.36
121	0.1	364	2.35
50	1.0	273	2.52
66	0.5	273	2.40
96	0.1	273	2.37
64	0.5	234	2.45
63	0.5	204	2.49
59	0.5	166	2.48
55	0.5	132	2.54

the target error goal. As can be seen, the necessary training set size to achieve a particular output error is dependent on the number of neurons being used, as might be expected. It appears that, for this particular problem, the optimum value for each of these was a training set of 364 examples with an rbf layer containing 74 neurons. In this situation, the error obtained was 2.36 per cent. This compares with the 4.2 per cent which was obtained by retraining the knowledge based network on its own. The training set used here was a subset of the set used for the experiments with the knowledge based network on its own. In all the experiments, a set of 600 test examples, drawn randomly from the total set of examples generated from a finite element analysis, was used.

To determine the effect of the white noise which was added to the system to ensure numerical stability when the weights were generated, a series of tests was run using different levels of white noise but keeping the system to 364 training examples. The results are shown in Table IV and it appears that, with a mean white noise level of 0.005, an error of 2.32 per cent is achieved with a radial basis function (rbf) layer consisting of 72 neurons.

A final test was run to investigate the effectiveness of the radial basis function network for this problem by using a conventional feedforward network with sigmoidal neurons. The best performance that could be achieved was about 2.7 per cent, i.e. not as good as the radial basis function system but considerably better than that achieved with the retrained knowledge-based system. However, this was achieved with a training set of 495 examples and required somewhat longer training times.

Conclusions

The paper has described a neural network structure capable of predicting the sensitivity of the performance of an electromagnetic device to changes in its design parameters in the presence of saturation and fringing. In this sense, the neural network system can take over from a full numerical (finite element) analysis once it has been trained, thus providing either a designer or an optimisation system with extremely fast turnaround times on design modifications. The neural network itself consists of two parts: the first is a network structured to represent a simple system of rules derived from a magnetic circuit structure; the second is an error correction system using radial basis function neurons to tune the output of the knowledge-based system. It has been shown that the use of an error removal network consisting of radial basis function neurons operating in parallel with a knowledge based network

Sum-square error goal	Mean = 0.01 (%)		Mean = 0.05 (%)		Mean = 0.001 (%)	
1	2.56	49	2.48	45	2.49	48
0.5	2.42	60	2.42	56	2.37	59
0.1	2.32	78	2.32	72	2.36	72
0.05	2.33	86	2.32	78	2.35	74

Table IV. The effects of input noise can produce better results than either retraining the knowledge-based system or using a conventional feedforward network for the error compensation.

The retraining of the network by using a set of finite element solutions can take some time, although the models described here are all two-dimensional and thus have reasonably fast solution times. However, once the network has been trained, the sensitivity predictions are made extremely quickly by the network and considerably faster than performing a numerical sensitivity analysis (Dyck *et al.*, 1994). Thus the major cost is in the training of the system. In general use, the system should only retrain occasionally when the front end decision system decides that proposed problem is too far from the current experience of the neural network.

However, more work is needed with this architecture to generalise it to a range of electromagnetic devices.

Also, as with all knowledge-based systems, the initial knowledge must be provided to the system by the user in the form of the equivalent circuit model and this may not always be a trivial exercise.

References

Arkadan, A.A. and Chen, Y. (1994), "Artificial neural network for the inverse electromagnetic problem of system identification", *Southeastcon '94*, Miami, FL, April, pp. 162-4.

Dandurand, F. and Lowther, D.A. (1998), "Electromagnetic device performance identification using knowledge based neural networks", *Proceedings of the IEEE Conference on Electromagnetic Field Computation*, Tucson, AZ, June.

Dyck, D.N., Lowther, D.A. and Freeman, E.M. (1994), "A method of computing the sensitivity of electromagnetic quantities to changes in materials and sources", *IEEE Transactions on Magnetics*, Vol. 30 No. 5, pp. 3415-18.

Ebner, T., Magele, C., Brandstätter, B.R. and Richter, K.R. (1998), "Utilizing feed forward neural networks for acceleration of global optimization procedures", *IEEE Transactions on Magnetics*, Vol. 34 No. 5, pp. 2928-31.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ.

Ratner, I., Ali, H.O., Pteriu, E.M. and Eatherley, G. (1996), "Neural network modelling of electromagnetic field problems", *1996 International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Processing*, Venice, Italy, August, pp. 387-91.