Response Clustering for Electromagnetic Modeling and Optimization

Mark Dorica and Dennis D. Giannacopoulos

Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada

Developing models from computational data is a major focus in electromagnetic design. This paper introduces ways of creating customized neural models based on a fuzzy clustering of responses. Fuzzy-clustered neural network (FCNN) models are explored, leading to increases in accuracy. The information contained within FCNN models can also be applied to space mapping electromagnetic optimization. This optimization approach strives to combine the accuracy of fine models (such as finite elements) with the low cost of coarse models. These FCNN enhancements are demonstrated through a patch antenna test case.

***Index Terms*—Artificial neural networks (ANNs), electromagnetic modeling, electromagnetic optimization, fuzzy clustering.**

I. Introduction

Selecting the correct model for electromagnetic optimization is a common task but not always straightforward. Fine models, such as finite elements, are impressive, but prohibitively expensive in many cases. Coarse models from circuit simulators or simplified theoretical models are fast but insufficient for the final stages of design.

Empirical models, such as response surfaces (RS), have found success in approximating the finite-element (FE) model [1]. Further increases in efficiency can be obtained by employing design of experiments (DOE) theory. These methods provide significant speed advantages. However, increasingly accurate and localized RS models are sometimes required as the optimization process approaches the final design.

While the building of an RS is a one-time cost, it can nevertheless be a substantial one. For instance, a successful RS model of an electromagnetic shaker required 171 magnetostatic solutions [1]. At 15 min per solution, the RS model would be ready in about 40 h. An alternate approach for achieving a compromise between accuracy and cost is desirable.

One such approach for combining the speed of coarse models with the accuracy of fine models is space mapping (SM) optimization [2]. This technique has been applied to a wide range of problems in the microwave arena [2]. SM strives to find the true fine optimal solution by optimizing in the coarse model space (e.g., coarser FE mesh) and iteratively creating a mapping between coarse and fine parameter spaces. The coarse model is expected to preserve the general character of the physical response, and the dependency on fine model simulations is largely replaced by dependency on much cheaper coarse models.

In the spirit of information reuse, the objective here is to produce coarse models that are customized to the underlying device responses. This strategy goes beyond the standard approach of fitting a mathematical expression to the simulation data. It is expected that customization will allow for an increase in accuracy for a fixed number of data points. Furthermore, the implicit information within these customized models may be used to accelerate the SM optimization process.

To achieve this objective of model customization, we explore a combined fuzzy clustering and artificial neural network (ANN) strategy: fuzzy-clustered neural network (FCNN) models. A patch antenna test case will highlight some of the inherent benefits and challenges of this approach.

II. Artificial Neural Network Models

The FCNN approach is based on ANNs. According to the universal approximation theorem, there always exists a three-layer perceptron ANN that can approximate any arbitrary nonlinear continuous multidimensional function to any desired accuracy [3]. The three-layer perceptron, having an input layer $(L = 1)$, a hidden layer $(L = 2)$, and an output layer $(L = 3)$, is used throughout this study. Each neuron (except for the input layer neurons) forms a weighted sum of its inputs, which is then passed through a nonlinear activation function.

In general, each neuron in layer L (with $L > 1$) produces an output in two steps. The first step is to compute a weighted sum γ_i^L of the outputs z_j^{L-1} of the previous layer $L - 1$

$$\gamma_i^L = \sum_{j=0}^{N_{L-1}} z_j^{L-1} w_{ij}^L \tag{1}$$

where w_{ij}^L is the weight connecting neuron j in layer $L - 1$ to neuron i in layer L. The number of neurons (outputs) in layer $L - 1$ is denoted by N_{L-1}.

The second step in the computation of the overall neuron output varies depending on the location of the neuron. The input neurons $(L = 1)$ simply relay their single input to all neurons in the next layer. Hidden layer neurons $(L = 2)$ produce an overall output by applying to the result of (1) the sigmoid activation function

$$\sigma(\gamma) = \frac{2}{1 + e^{-2\gamma}} - 1. \tag{2}$$

The neurons of the third layer produce the output of (1) (linear activation function). Notice the index of summation in (1) begins at zero. The neuron output $z_0^{L-1} \equiv 1$ is fictitious and serves to provide a bias for the ANN computation.

Digital Object Identifier 10.1109/TMAG.2006.872021

The purpose of training is to optimize the ANN weight vector such that the ANN model best approximates the training data and produces accurate generalizations. Data is normalized by scaling to the range -1 to 1 and is divided into training and validation sets. Training data is used for adjustment of the weight parameters, whereas validation data is used for monitoring the progress of training and determining stopping criteria. Training is accomplished through the Levenberg–Marquardt method [3]. When the validation error increases for a certain number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. This measure is taken to prevent a loss of generalization capability of the ANN. Other stopping criteria include a maximum 500 epochs of training, and a mean-squared error (MSE) goal of 1×10^{-6}. Several hidden layer sizes are attempted, and the three-layer perceptron with the lowest MSE is selected as the final model. The greater the nonlinearity of the function being approximated and the dimensionality of the input and output vectors, the greater the number of hidden layer neurons that are required.

III. Fuzzy c-Means Clustering

Clustering is the organization of items into groups whose members are similar in some way. A multitude of techniques have been proposed for solving this problem, including partitional algorithms [4], self-organizing maps [3], and evolutionary algorithms [4]. In traditional approaches, each data vector belongs to a single cluster, leading to a crisp partition. Conversely, the fuzzy c -means (FCM) algorithm associates each data vector to each cluster through a membership value [4]. FCM iteratively minimizes the cost function

$$J = \sum_{i=1}^{N}\sum_{j=1}^{C}(u_{ij})^m \|x_i - c_j\|^2, \quad 1 \leqslant m < \infty \tag{3}$$

where N is the number of sample vectors, and C is the number of clusters. Data vectors and cluster centers are represented by x_i and c_j, respectively. The membership value u_{ij}, having a range [0, 1], represents the degree of membership of data vector i in cluster j. The tuning parameter m controls the level of cluster fuzziness (chosen to be 2). When m is 1, partitions between clusters are crisp, and the degrees of membership are either 0 or 1. As m increases, membership grades begin taking on values between 0 and 1, leading to cluster fuzziness. The algorithm proceeds as follows.

1) $k = 0$.
2) Initialize the $N \times C$-sized $U^k = [u_{ij}]$ randomly such that $u_{ij} \in [0,1]$ and $\sum_{j=1}^{C} u_{ij} = 1$.
3) $c_j = \frac{\sum_{i=1}^{N}(u_{ij})^m x_i}{\sum_{i=1}^{N}(u_{ij})^m}, \quad j = 1, \ldots, C.$
4) $U^{k+1} = [u_{ij}] = \left[\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{2/(m-1)}\right]^{-1}.$
5) If $|U^{k+1} - U^k\| < \varepsilon$ or $k >$ maxiter STOP, else $k = k + 1$ and go to step 3).

In step 3), the cluster centers represent the mean of all data vectors, weighted by their degree of membership in the cluster. The membership values in step 4) represent the inverse of a measure of the distance between data vector x_i and cluster center c_j.

In other words, FCM is based on minimizing the objective function (3) that represents a sum of products of the data vector-to-cluster center distance and the data vector's membership degree in that particular cluster. Given a desired number of clusters, the algorithm will return the cluster centers along with degrees of membership for each data vector. For example, selecting three clusters will result in three cluster centers and three degrees of membership assigned to each data vector. The stopping criteria are set to $\varepsilon = 1 \times 10^{-5}$ and $\mathrm{maxiter} = 100$.

IV. Fuzzy-Clustered Neural Models

Two methods for clustering response data are explored. In the multiple-cluster-single-output (MCSO) approach, a single-output ANN is assigned to each of the C clusters. The training data for ANN i corresponds to data points having a greater than 0.5 membership degree in cluster i. Thus, a single data point may be used in the training of several ANNs, leading to a smooth and overlapped transition of training data between the C ANNs.

The multiple-cluster-multiple-output (MCMO) approach takes MCSO a step further by providing an ANN for each point in the frequency sweep. In other words, if F frequency points characterize the response of the device being modeled and C clusters are chosen, MCMO will produce $F \times C$ ANN models. Within a given cluster, all F ANN will be trained using the same set of input data, with the "greater than 0.5" membership value criterion intact. In both MCSO and MCMO, each ANN is only responsible for accurate modeling and generalization within a cluster of similar (homogeneous) responses, thus reducing the number of required training vectors.

For purposes of comparison, results for the standard single-cluster-single-output (SCSO) approach are provided. In SCSO, a single (single-output) ANN is trained on the full set of data and no clustering is used.

V. E-Shaped Patch Antenna

The E-shaped patch antenna [5] shown in Fig. 1 will be used to demonstrate the application of FCNNs to modeling and optimization.

Let us assume a starting design (in millimeters)

$$\begin{aligned}(L, W, h) = (70, 45, 10), \quad (X_f, Y_f) = (35, 7)\\ L_s = 35 \quad W_s = 4 \quad P_s = 9.\end{aligned} \tag{4}$$

The width W of the antenna controls the higher resonant frequency, whereas the slots (L_s, W_s, P_s) control the lower resonant frequency. Let us state the design objective of minimizing the maximum reflection coefficient $|S_{11}|$ at 2.21 and 2.58 GHz. The model was solved over a uniform grid of 100 points. The first design variable W is shown as a vertical dashed line in Fig. 1. The width W is sampled at ten equally spaced points between 36 and 62 mm (along the y axis). The second design variable is shown by the two horizontal dashed lines and denoted by H and H'. The H parameter, whose starting design value is 46 mm, is varied through ten sample points between 43 and 69

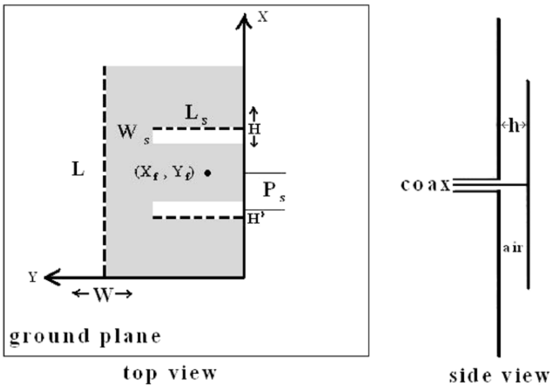

Fig. 1. Geometry of E-shaped patch antenna test case.

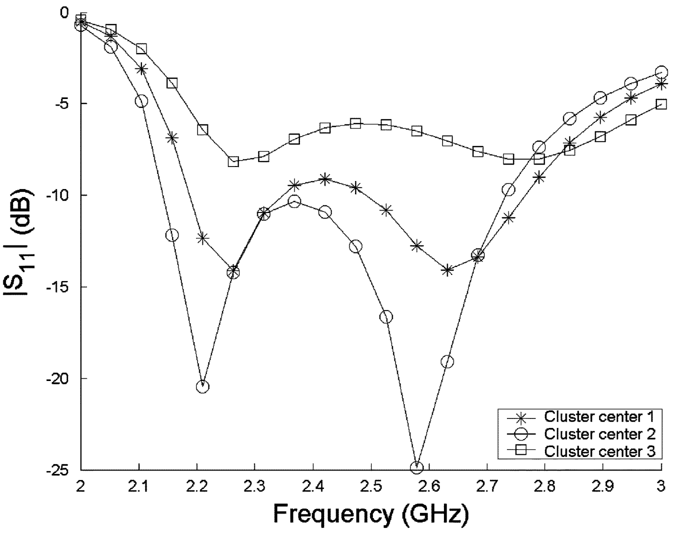

Fig. 2. Cluster centers of antenna responses.

mm (along the x axis). The lower horizontal line H' moves opposite to H (when H moves up, H' moves down, and so on).

The FCM technique was used to compute three cluster centers as shown in Fig. 2. Approximately 1000 test points were selected in the region of interest to test the accuracy of the FCNN techniques. Fig. 3 shows these results relative to the reference SCSO. Each data point in Fig. 3 is placed between two horizontal axis labels and represents the number of samples having an error within that range. MCMO (2.42% mean error) has a clear advantage over both MCSO (14.1% mean error) and SCSO (11.9% mean error). All three required a total ANN training time of about 20 s. Notice that SCSO and MCSO have errors in the 100%–1000% range, while MCMO does not. There are two main factors at play here: the number of ANNs and the amount of training data per ANN. The MCSO technique creates an ANN for each cluster. An increased number of clusters reduces the amount of training data for each ANN. While clustering provides customization to the underlying data, the reduced training volume hampers the end result. Adjusting MCSO so that additional data points are added to low training data zones would remedy this situation. MCMO is less susceptible to this tradeoff because each ANN is responsible for an even more homogeneous response than in MCSO, thereby requiring less training data in each cluster.

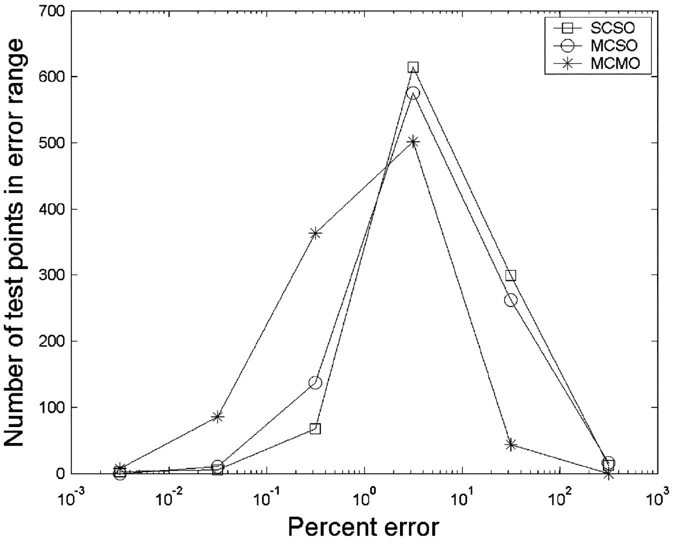

Fig. 3. Percent error of FCNN techniques for selected test points.

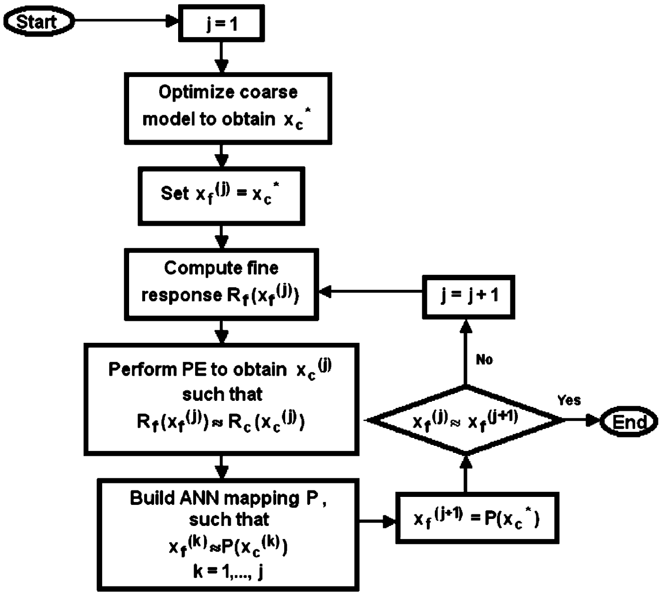
Start
j = 1
Optimize coarse model to obtain x_c^*
Set $x_f^{(j)} = x_c^*$
Compute fine response $R_f(x_f^{(j)})$
j = j + 1
Perform PE to obtain $x_c^{(j)}$ such that $R_f(x_f^{(j)}) \approx R_c(x_c^{(j)})$
No
Yes
$x_f^{(j)} \approx x_f^{(j+1)}$
End
Build ANN mapping P, such that $x_f^{(k)} \approx P(x_c^{(k)})$ k = 1,..., j
$x_f^{(j+1)} = P(x_c^*)$

Fig. 4. Space mapping algorithm based on [7].

VI. Space Mapping Applications

The division of a model into several submodels can also be applied to optimization. Most SM approaches rely on a step called "parameter extraction" (PE). Examples include the "aggressive space mapping" (ASM) [6] and the more recent "neural inverse space mapping" (NISM) [7]. In both cases, the PE stage is responsible for finding the coarse model parameters such that the coarse response is as close as possible to a previously computed fine response. In Fig. 4, the algorithm begins with a coarse model optimization which is inexpensive. Within the loop, a fine model evaluation is carried out at each iteration. For our test case, this involves $|S_{11}|$ at 20 frequency points. Following PE, a neural mapping is built between input spaces (fine parameters and coarse parameters from PE) and progresses until the fine optimum is found.

These SM techniques have been shown to provide extensive time benefits in optimization when applied to coarse models

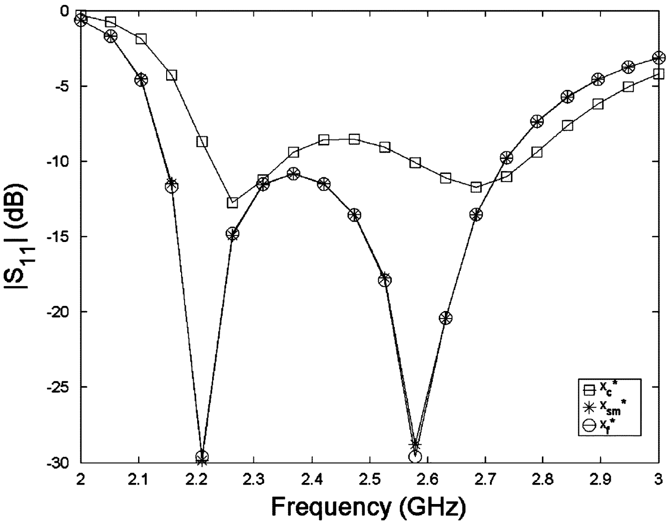

Fig. 5. Optimization results for E-shaped patch antenna.

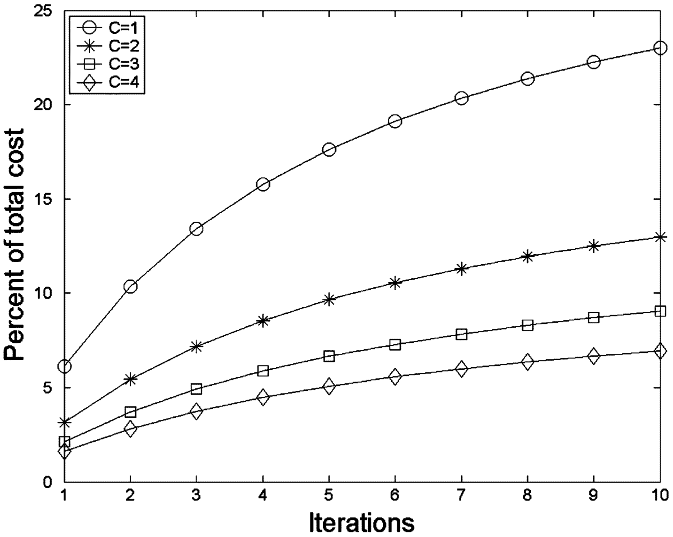

Fig. 6. Effect of number of clusters C on relative cost of PE in SM.

from commercial circuit simulators [2]. Situations arise where the mapping is difficult to determine and multiple SM iterations may be necessary. The use of FCNN under these circumstances may be plausible. Dividing the coarse input space into separate ranges as determined by the clusters of FCNN reduces the search space for the PE step (e.g., genetic algorithm implementation [8]) and may speed up the overall SM process. The target fine response of PE $R_f(x_f^{(j)})$ (obtained by using a finer mesh) is swept through the grid of existing FCNN coarse data to find which data vector (and in turn which cluster) it most closely resembles. The PE step then focuses on this specific range, thereby reducing the size of the search space. The results of SM for the MCSO (14.1% mean error) coarse model over ten iterations are shown in Table I and Fig. 5. The coarse optimum x_c^*, fine optimum x_f^*, and SM optimum $x_{\rm sm}^*$ are provided. MCMO yielded a similar SM result. Fig. 6 shows the effects of the reduced search space on the SM process. An increased number of clusters allows for a reduction in the relative cost of PE over all SM iterations.

TABLE I
DESIGNS FROM OPTIMIZATION TECHNIQUES

Parameter	Coarse (x_c*)	Fine (x_f*)	SM (x_{sm}*)
W (mm)	46.89	48.59	48.62
H (mm)	46.67	45.78	45.75

VII. CONCLUSION

Customizing electromagnetic models based on intrinsic information in the device response is promising for both modeling and optimization. Accuracy enhancements are particularly attractive when a limited number of data points are available. The use of FCNN customization provides guidance for the model building. While the number of ANN models is larger than in standard SCSO approaches, the training data is also more homogeneous for each ANN, allowing for a reduced complexity in the neuronal structure. It was also observed that FCNN can be useful in optimization, where the reduced search space is beneficial.

Selecting the best number of clusters for a given problem is yet to be fully explored. The tradeoff between increased customization (more clusters) and decreased training data in each cluster could be the subject of future work.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and Le Fonds québécois de la recherche sur la nature et les technologies.

REFERENCES

[1] D. N. Dyck and B. S. Murray, “Transient analysis of an electromagnetic shaker using circuit simulation with response surface models,” *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3698–3701, Sep. 2001.

[2] J. W. Bandler *et al.*, “Space mapping: The state of the art,” *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 337–361, Jan. 2004.

[3] P. Burrascano, S. Fiori, and M. Mongiardo, “Review of artificial neural networks applications in microwave computer-aided design,” *Int. J. RF Microw. CAE*, vol. 9, no. 3, pp. 158–174, 1999.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[5] F. Y. Yang, X.-X. Zhang, X. Ye, and Y. Rahmat-Samii, “Wide-band E-shaped patch antennas for wireless communications,” *IEEE Trans. Antennas Propag.*, vol. 49, no. 7, pp. 1094–1100, Jul. 2001.

[6] J. W. Bandler *et al.*, “Space mapping technique for electromagnetic optimization,” *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 12, pp. 2536–2544, Dec. 1994.

[7] J. W. Bandler *et al.*, “Neural inverse space mapping for EM-based microwave design,” *Int. J. RF Microw. CAE*, vol. 13, pp. 136–147, 2003.

[8] G. F. Uler, O. A. Mohammed, and C.-S. Koh, “Utilizing genetic algorithms for the optimal design of electromagnetic devices,” *IEEE Trans. Magn.*, vol. 30, no. 6, pp. 4296–4298, Nov. 1994.

Manuscript received June 20, 2005 (e-mail: dennis.giannacopoulos @mcgill.ca).