ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2008 (537-548)
VOLUME : 8
NUMBER : 1

COMPUTATIONAL INVESTIGATION ON THE USE OF FEM AND MLP NEURAL NETWORK IN THE INVERSE PROBLEM OF DEFECTS IDENTIFICATIONS

T. HACIB M. R. MEKIDECHE N. FERKHA

LAMEL Laboratory, Electrical Engineering Department, Jijel University
18000 Algeria, Phone: +213-34 50 11 89, Fax: +213-34 50 26 89
tarik_hacib@mail.univ-jijel.dz

ABSTRACT

In this article an attempt is made to study the applicability of a general purpose, supervised feed forward neural network, namely multilayer perceptron (MLP) neural network and finite element method (FEM) to solve the inverse problem of defect identification. The approach is used to identify unknown defects in metallic walls. The methodology used in this study consists in the simulation of a large number of defects in a metallic wall, using the finite element method. Both variations in with and height of the defects are considered. Then, the obtained results are used to generate a set of vectors for the training of MLP neural network model. Finally, the obtained neural network is used to identify a group of new defects, simulated by the finite element method, but not belonging to the original dataset. Noisy data, added to the probe measurements is used to enhance the robustness of the method. The reached results demonstrate the efficiency of the proposed approach, and encourage future works on this subject.

Keywords: *Inverse problem, MLP neural network, defect identification, FEM.*

1. INTRODUCTION

Inverse problems in electromagnetic are usually formulated and solved as optimization problems, so iterative methods are commonly used approaches to solve this kind of problems [1]. These methods involve solving well behaved forward problem in a feedback loop. The numerical models such as finite element model are used to represent the forward process. However, iterative methods using the numerical based forward models are computationally expensive. Recently, artificial neural networks (ANNs) are introduced to solve the inverse problems in most of the research applications in industrial nondestructive testing, mathematical modeling, medical diagnostics, geophysical prospecting for petroleum and minerals, and detection of earthquakes [2-6].

Received Date: 20.02.2007
Accepted Date: 25.03.2008

Electromagnetic inverse problems can sometimes be stated as simply as the following: if there is an electromagnetic device, it is easy to calculate the magnetic induction in any region of the device. What, about taking some values of magnetic induction to predict defects in a region of the electromagnetic device. Since, the inverse problem is highly nonlinear and without formulations to follow, it is very difficult to construct an effective inversion algorithm. An artificial neural network, however, has the following properties: nonlinearity, input-output mapping, fault tolerance and most important, learning from examples. The need for learning from examples is closely related to the difficulty of formulating explicit rules.

ANNs are based on abstracting from the complex details of human thought and building a simple model using a network of simple processors. ANNs consist of a large number of simple processing elements called neurons or nodes. Each neuron is connected to other neurons by means of directed links, each with an associated weight [7]. The weights represent information being used by the network to solve a problem. The ANN essentially determines the relationship between input and output by looking at examples of many input-output pairs. In learning processes, the actual output of the ANN is compared to the desired output. Changes are made by modifying the connection weights of the network to produce a closer match. The procedure iterates until the error is small enough [8].

In this paper we present a new method for the robust estimation of defect dimensions. The method is based on the use of FEM and ANN scheme. The network is trained by a large number of defects in a metallic wall simulated using the FEM. The obtained results are then used to generate the training vectors for ANN. The trained network is used to identify new defects in the metallic wall, which not belong to the original dataset. The network weights can be embedded in an electronic device, and used to identify defects in real pieces, with similar characteristics to those of the simulated ones.

For the methodology presented here, the measured values are independent of the relative motion between the probe and the piece under test. In other words, the movement is necessary only to change the position of the probes, to acquire the field’s values, which are necessary for the identification of new defects. Furthermore, the use of neural network in conjunction with the FEM permits a very good determination of both, width and height of the defect. To show stability of the proposed method, we add one percent noise to the inputs values of the network.

2. NEURAL NETWORKS ARCHITECTURE

ANNs, also called artificial neural systems. neurocomputers, parallel distributed processors or connectionist models are an attempt to mimic the structure and functions of brains and nervous systems of living creatures. Generally speaking, an ANN is an information processing systems composed of a large number of simple processing elements, called artificial neurons or simply nodes. Neurons are interconnected by direct links called connections with an associated weight, which cooperate to perform parallel distributed processing in order to solve a desired computational task. One of the attractive features of ANNs is their capability to adapt themselves to special environmental conditions by changing their connection strengths or structure. Years of studies have shown that ANNs exhibit a surprising number of the brain's characteristics. For example, they learn from experience, generalize from previous examples, and abstract essential characteristics from inputs containing irrelevant data. In this paper we choose the back-propagation method to demonstrate the potential of ANNs to solve electromagnetic inverse problems of defects identifications [9].

One of the most influential developments in ANN was the invention of the backpropagation algorithm, which is a systematic method for training multilayer ANNs [10]. The standard backpropagation learning algorithm for feedforward networks aims to minimize the mean squared error defined over a set of training data. In feedforward ANNs neurons are arranged in a feedforward manner, so each neuron may receive an input from the external environment or from the neurons in the former layer, but no feedback is formed. The network architecture for a feed forward network consists of layers of processing nodes. The network always has an input layer, an output layer and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there will be one or two. In our case, there is only one hidden layer. Every neuron in each layer of the network is connected to every neuron in the adjacent forward layer. A neuron's activity is modeled as a function of the sum of its weighted inputs, where the function is called the activation function, which is typically nonlinear, thus giving the network nonlinear decision capability. Each layer is fully connected to the succeeding layer. The arrows indicate flow of information (Fig.1) [11, 12].

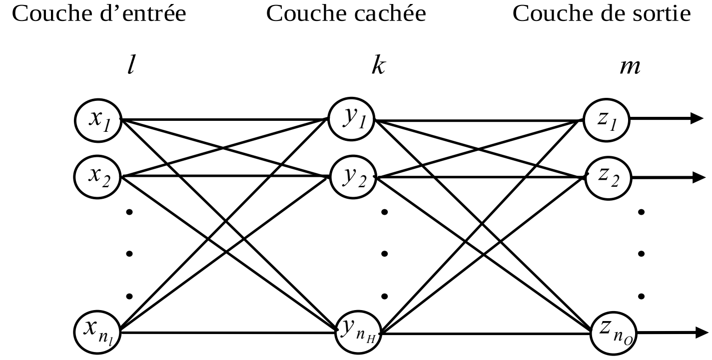

Fig. 1. *Feed forward neural network*

Where n_l is the number of neurons in the input layer, n_H is the number of neurons in the hidden layer, n_O is the number of neurons in the output layer, x_l are the inputs to the input layer where $l = 1, \ldots, n_l$, y_k is the value of the hidden layer where $k = 1, \ldots, n_H$, z_m is the value of the output layer where $m = 1, \ldots, n_O$. $w_{lk}^{[1]}$ is the weight connecting the $l\,th$ neuron in the input layer to $k\,th$ neuron in the hidden layer, and $w_{km}^{[2]}$ is the weight connecting the $k\,th$ neuron in the hidden layer to the $m\,th$ neuron in the output layer. The nodes of the hidden and output layer are:

$$y_k = f\left(\sum_{l=1}^{n_l} w_{lk}\, x_l\right) \quad k = 1, \ldots, n_H \qquad (1)$$

and

$$z_m = f\left(\sum_{k=1}^{n_k} w_{mk}\, y_k\right) \quad \mathrm{m} = 1, \ldots, n_O \qquad (2)$$

where the activation function f is traditionally the Sigmoid function but can be any differentiable function. The Sigmoid function is defined as

$$f(x) = \frac{1}{(1 + e^{-x})} \qquad (3)$$

This activation function is depicted in Fig. 2.

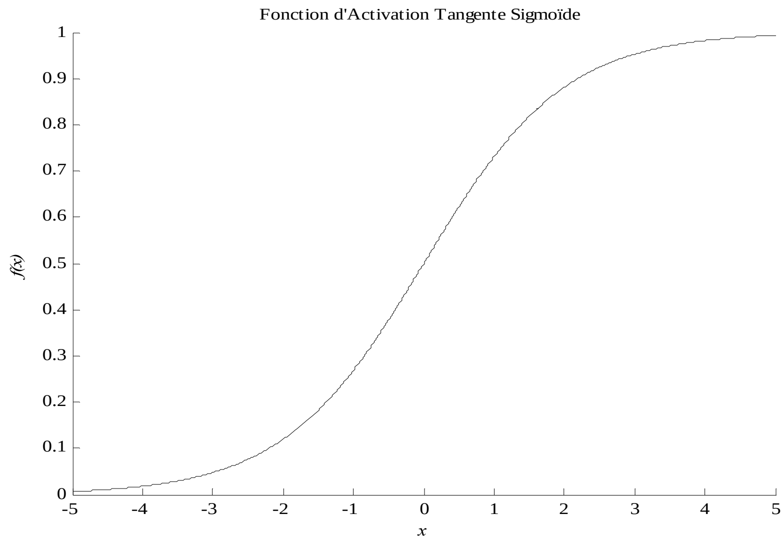

Fig. 2. *Sigmoid activation function*

The backpropagation method is based on finding the outputs at the last (output) layer of the network and calculating the errors or differences between the desired outputs and the current outputs. When the outputs are different from the desired outputs, corrections are made in the weights, in proportion to the error.

$$\Delta w_{km}^{[2]} = y_k f'(z_m)(z_m - d_m) \qquad (4)$$

where d_m represent the desired output, $k = 1,\dots,n_H$, $m = 1,\dots,n_O$ and

$$f'(x) = \frac{\partial f(x)}{\partial x} \qquad (5)$$

If f is the Sigmoid function, and

$$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = f(x)(1 - f(x)) \qquad (6)$$

The update rule for the weights from the hidden layer to the output layer is

$$w_{km(new)}^{[2]} = w_{km(old)}^{[2]} + \eta \Delta w_{km}^{[2]} \qquad (7)$$

where $k = 1,\dots,n_H$, $m = 1,\dots,n_O$ and η is the learning rate. The update rule for the weights from the input layer to the hidden layer is

$$\Delta w_{km}^{[1]} = x_l f'(y_k) \sum_{m=1}^{n_O} w_{km}^{[2]} f'(z_m)(z_m - d_m) \qquad (8)$$

$$w_{lk(new)}^{[1]} = w_{lk(old)}^{[1]} + \eta \Delta w_{lk}^{[1]} \qquad (9)$$

where $l = 1,\dots,n_l$, $k = 1,\dots,n_H$.

3. ELECTROMAGNETIC FIELD COMPUTATION

In this study, the magnetic field is calculated using the FEM. This method is based on the $\boldsymbol{A}$ representation of the magnetic field [13]. The calculations are performed in two steps. First, the magnetic field intensity is calculated by solving the system of equations:

$$\boldsymbol{rot}\,(\boldsymbol{H}) = \boldsymbol{J} \qquad (10)$$

$$div\,(\boldsymbol{B}) = 0 \qquad (11)$$

where $\boldsymbol{H}$ is the magnetic field, $\boldsymbol{B}$ the magnetic induction and $\boldsymbol{J}$ the electric current density. This system of equations is coupled with relations associated to material property, material being assumed to be isotropic,

$$\boldsymbol{B} = \mu(|\boldsymbol{H}|)\boldsymbol{H} \qquad (12)$$

The magneto-static field analysis for a Cartesian electromagnetic system is carried out by the FEM [14]. The equation of the electromagnetic field is expressed by the magnetic vector potential $\boldsymbol{A}$ as,

$$div\left(\frac{1}{\mu}\boldsymbol{grad}\ (\boldsymbol{A})\right)=-\boldsymbol{J} \tag{13}$$

where μ is the magnetic permeability.

Equation (13) is discretized using the Galerkin FEM, which leads to the following algebraic matrix equation:

$$[\boldsymbol{K}]\,[\boldsymbol{A}]=[\boldsymbol{F}] \tag{14}$$

with:

$$\boldsymbol{A}=\sum_{\boldsymbol{j}}\alpha_{\boldsymbol{j}}(x,y)\boldsymbol{A}_{\boldsymbol{j}} \tag{15}$$

$$\boldsymbol{K}_{ij}=\iint_{\Omega}\frac{1}{\mu}\boldsymbol{grad}\,\alpha_i\ \boldsymbol{grad}\,\alpha_j\ dx\,dy \tag{16}$$

$$\boldsymbol{F}_i=\iint_{\Omega}\boldsymbol{J}\ \alpha_i\ dx\,dy \tag{17}$$

where α_i and α_j are the projection function and the interpolation function respectively.

In the second step, the field solution is used to calculate the magnetic induction $\boldsymbol{B}$. More details about the finite element theory can be found in [14].

4. METHODOLOGY FOR DEFECTS IDENTIFICATION

First of all, an electromagnetic device was idealized to be used as an electromagnetic field exciter (Fig. 3). In this paper, we have considered direct current in the coils. So, the material of the metallic wall must be ferromagnetic. To increase the sensitivity of the electromagnetic device a magnetic core with a high permeability is used and the air gap between the core and the metallic wall is reduced to a minimum. Deviations of the magnetic induction at equally stepped points in the region of the device are taken, on the external surface of the metallic wall.

In order to generate the training vectors for the neural networks, a large number of defect shapes must be simulated. In this work, 300 defects have been simulated, each one corresponding to one simulation with the finite element program.

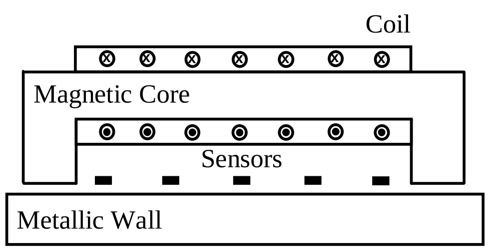

Fig. 3. *Arrangement for the measurements*

Fig. 4 show the steps of the methodology used in this work. Steps 1-4 correspond to the finite element analysis of the defects. In this work we used a 2D finite element program to simulate the defects in a metallic wall.

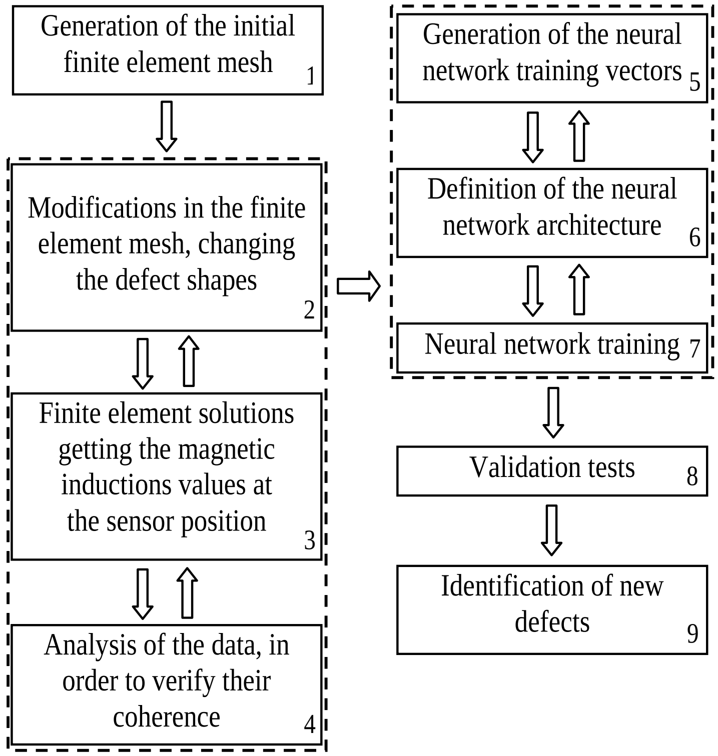

Fig. 4. *Flowchart of the used methodology*

The problem was solved on a PC with P4 2.4G CPU under Matlab® 6.5 workspace using the Partial Differential Equation Toolbox and Neural Network Toolbox for the finite element meshes generation and neural networks architecture definition respectively [15, 16]. For the finite element problem resolution and the inverse problem solution, we use programs developed by us. The simulations were done for a hypothetic metallic wall with 1 mm height and 15 mm length. The material of the wall is 1006 Steel (a magnetic material). The relative permeability of the core is supposed to be 2500 and the permeability of the defects was set to the permeability of the air. The air gap is 0.1 mm. Finite element meshes with 17000 elements and 8000 nodes, approximately, were used in the simulations. Fig. 5 shows a field distribution for one of these simulations.

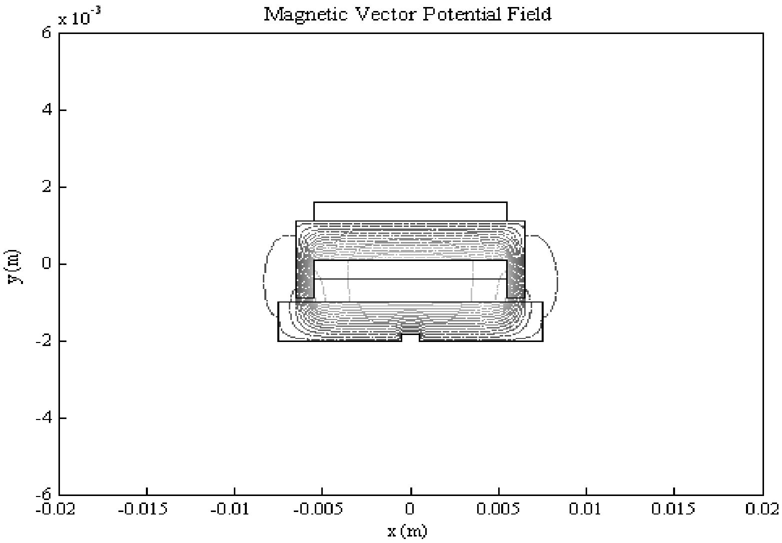

Fig. 5. *Solution in magnetic potential vector* ***A*** *for a defect with 0.15 mm height and 1.18 mm width*

During the phase of finite element simulations, errors can appear, due to it's massively nature. So, the results of the simulations must be carefully analyzed. This can be done, for instance, plotting in the same graphic the magnetic induction deviations for a set of defects. Fig. 6 shows the deviation on the magnetic induction in the region of the device at the sensor position for four defects having the same height (0.15 mm), and width ranging from 0.26 mm to 3.23 mm. A similar graphic, with height equal to 0.3 mm, is shown in Fig. 7. Fig. 8 shows the graphics for a fixed width (1.8 mm), and four different heights ranging from 0.05 mm to 0.43 mm. Fig. 9 shows a similar graphic, for the width equal to 3.6 mm. In this graphics the magnetic inductions deviations are at vertical axes and length are at horizontal axes.

The coherence of the curves in these graphics allows us to infer if there are or not errors in the dataset.

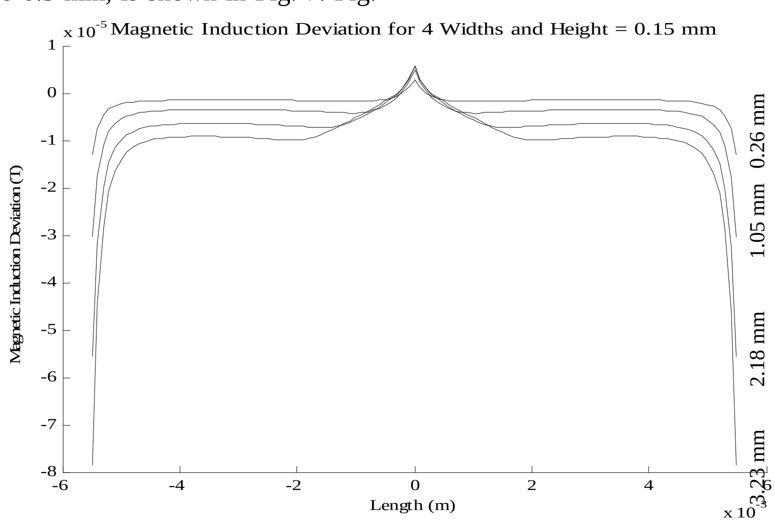

Fig. 6. *Magnetic induction deviation for a set of defects with the same height (0.15 mm)*

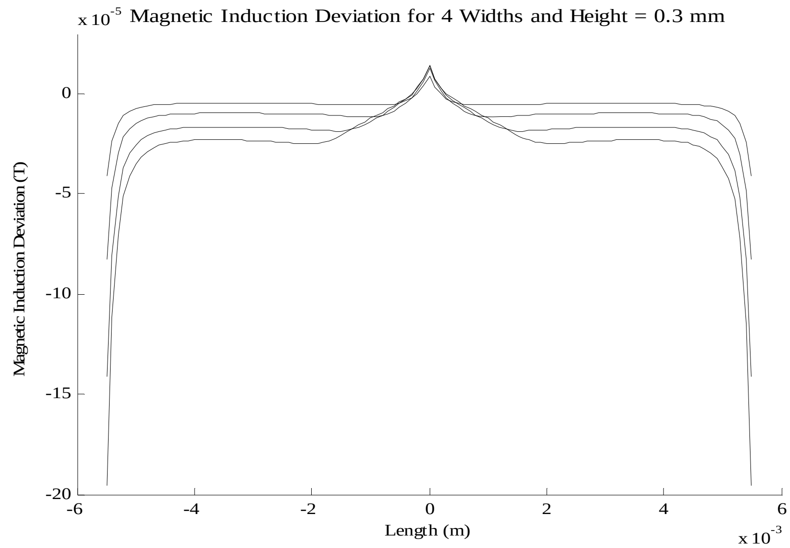

Fig. 7. *Magnetic induction deviation for a set of defects with the same height (0.3 mm)*

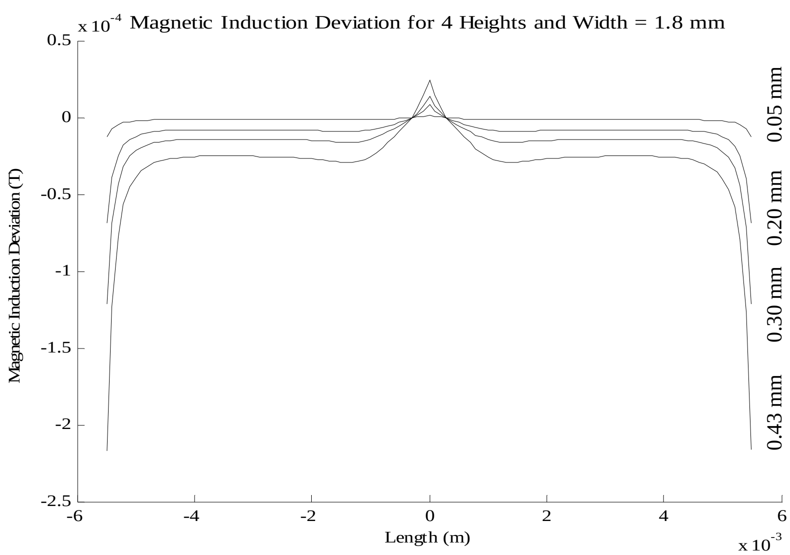

Fig. 8. *Magnetic induction deviation for a set of defects with the same width (1.8 mm)*

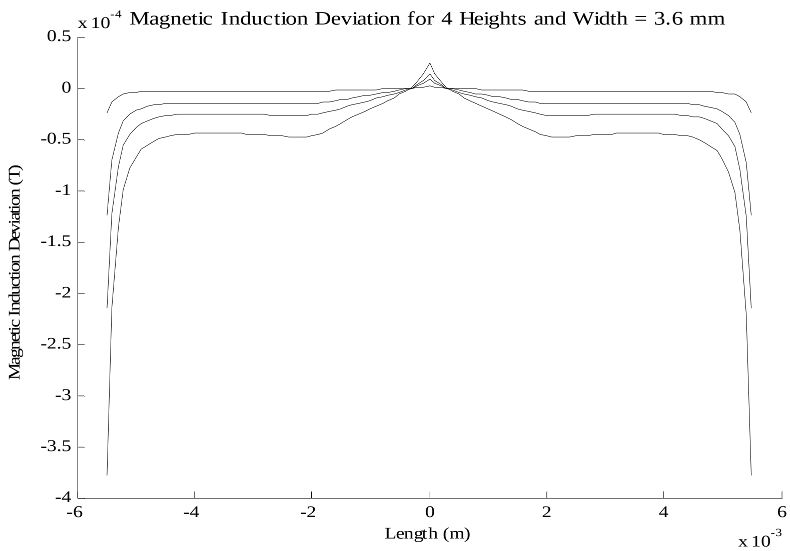

Fig. 9. *Magnetic induction deviation for a set of defects with the same width (3.6 mm)*

5. FORMULATION OF NETWORK MODELS FOR DEFECTS IDENTIFICATION

In the step 5, we generate the training vectors for neural networks. In this work, we generated 300 vectors for neural networks training. Each of the vectors consists of 11 inputs valus, which represent the deviation of magnetic induction, and two output values, which represent the height and width of defect. Of the 300 vectors, a random sample of 225 cases (75 %) was used as training, 75 (25 %) for validation. Training data were used to train the application and the validation data were used to monitor the neural network performance during training.

To show stability of the proposed approach, the measured values, which intrinsically contains errors in the real word, is obtained by adding a random perturbation to the exact inputs values of the network, such that

$$\widetilde{In} = In_{\boldsymbol{exact}} + \sigma\lambda \qquad (18)$$

where σ is the standard deviation of the errors and λ is a random variable taken from a Gaussian distribution, with zero mean and unitary variance.

Twin numerical experiments were performed. In the first one, noiseless data where employed (σ=0). The second numerical experiment was carried out using 1% of noise (σ=0.01).

The MLP neural network architecture considered for this application was a single hidden layer with sigmoid activation function. The learning rate initially is 0.5 but as the root mean squared error gets smaller it decreases to 0.3. This is the experience from the training which also matches the idea of learning rate annealing in [7, 17]. A back propagation algorithm based on Levenberg-Marquardt optimization technique [18, 19] was used to model MLP for the above data.

The Levenberg-Marquardt technique was designed to approach second order training speed without having to compute the Hessian matrix [19]. This matrix approximated with use of the Jacobian matrix which can be computed through a standard back propagation algorithm that is much less complex than computing the Hessian matrix. The performance function will always be reduced on each iteration of the algorithm.

For the MLP neural network, several network configurations were tried, and better results have been obtained by a network constituted by one hidden layers with 18 neurons. The MLP architecture had 11 input variables, one hidden layer and two output nodes. Total number of weights present in the model was 254. The best MLP was obtained at lowest mean square error of 10^{-6}. Percentage correct prediction of the MLP model was 96.7 % and 96.1% for noiseless and noise data respectively.

Fig. 10 shows the performance of the MLP neural network during a training session. Table I show some results for the validation of the network, for this session.

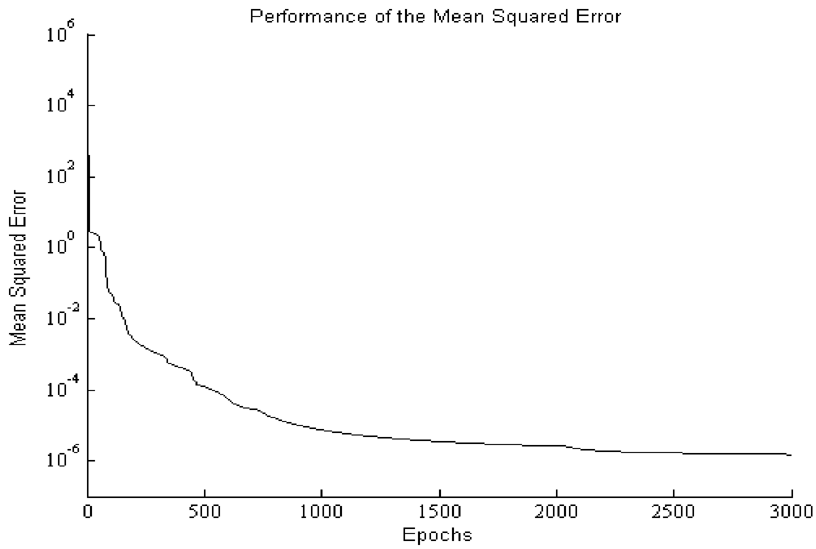

Fig. 10. *Performance of the MLP network during a training session*

Table. I. *Expected and obtained values during a training session*

Height (mm)			Width (mm)		
Expected	Obtained		Expected	Obtained	
	0 % Noise	1 % Noise		0 % Noise	1 % Noise
0.0710	0.0714	0.0703	5.9484	5.9479	5.9472
0.1988	0.1983	0.1974	6.4670	6.4652	6.4587
0.2418	0.2423	0.2409	1.5133	1.5171	1.5189
0.3378	0.3372	0.3367	4.5790	4.5767	4.5708
0.5169	0.5164	0.5155	3.3280	3.3295	3.3351
0.6840	0.6843	0.6836	2.1720	2.1743	2.1827

As we can see, the results obtained in the validation are very close to the expected ones. The worse identification defect was obtained with MLP network, because this network has some drawbacks such as slow convergence and the possibility that the network converges to a local minimum.

6. NEW DEFECT DENTIFICATION

After the neural networks training and respective validations, new defects were simulated by the FEM, for posteriori identification by the networks. Table II shows the dimensions of the defects (height and width), and the obtained dimensions, by the neural networks.

Table. II. *Simulation results for new defects*

Defect	Height (mm)			Width (mm)		
	Expected	Obtained		Expected	Obtained	
		0 % Noise	1 % Noise		0 % Noise	1 % Noise
1	0.0750	0.0751	0.0754	3.910	3.909	3.903
2	0.1250	0.1248	0.1236	2.462	2.471	2.480
3	0.2150	0.2153	0.2110	4.850	4.843	4.837
4	0.5750	0.5755	0.5686	1.080	1.082	1.090

As we can see, the results obtained in the identification of new defects, obtained by the neural networks agree very well with the expected ones, demonstrating that the association of the FEM and ANNs in very powerful in the solution of inverse problems like defects identifications in metallic walls.

7. CONCLUSION

In this paper we presented an investigation on the use of the FEM and MLP neural network for the identification of defects in metallic walls, present in industrial plants. For a given metallic wall characteristics, defects can be simulated by the FEM, and the magnetic fields results are used in the preparation of the training vectors for neural network. The network can be embedded in electronic devices in order to identify defects in real metallic walls. This study indicates the good and stable predictive capabilities of MLP neural network in the presence of noise.

The association of FEM and ANN techniques seems to be a useful alternative for identification of defects trough inverse analysis. Future works are intended to be done in this field, such as the use of more realistic FEM, computer parallel programming, in order to get quickly solutions.

REFERENCES

[1] Ramuhalli. P., *Neural network based iterative algorithms for solving electromagnetic NDE inverse problems*, Ph.D. dissertation, Dept. Elect. Comp. Eng, Iowa Univ, USA, 2002.

[2] Coccorese. E., Martone. R., Morabito. F. C., "A neural network approach for the solution of electric and magnetic inverse problems", *IEEE Trans. Magnetics*, Vol: 30, No: 5, pp. 2829-2839, 1994.

[3] Hoole. S. R. H., "Artificial neural networks in the solution of inverse electromagnetic field problems", *IEEE Trans. Magnetics*, Vol: 29, No: 2, pp. 1931-1934, 1993.

[4] Ramuhalli. P., Udpa. L., Udpa. S. S., "Finite element neural networks for solving differential equations", *IEEE Trans. Neural Networks*, Vol: 16, No: 6, pp. 1381-1392, 2005.

[5] Wong. P. M., Nikravesh. M., "Field applications of intelligent computing techniques", *J. Petrol Geolog*, Vol: 24, No: 4, pp. 381-387, 2001.

[6] Fanni. A., Montisci. A., "A neural inverse problem approach for optimal design", *IEEE Trans. Magnetics*, Vol: 39, No: 3, pp. 1305-1308, 2003.

[7] Haykin. S., *Neural networks: A comprehensive foundation*, Englewood Cliffs, NJ: Prentice-Hall, New York, 1999.

[8] Turchenko. I. V., "Simulation modelling of multi-parameter sensor signal identification using neural networks", *Proc. 2nd IEEE Int Conf. Intelligent Systems, Bulgaria, 2004, pp. 48-53.*

[9] De Alcantara. N.P., Alexandre. J., De Carvalho. M., "Computational investigation on the use of FEM and ANN in the non-destructive analysis of metallic tubes", *Proc. 10th Biennial Conf. Electromagnetic Field Computation*, *Italy, 2002.*

[10] Jain. A. K., Mao. J., Mohiuddin. K. M., *Artificial neural networks: a tutorial*, *Computer*, pp. 31-44, 1996.

[11] Mehrotra. K., Mohan. C. K., Ranka. S., *Elements of artificial neural networks*, MA: MIT Press, Cambridge, 1997.

[12] Cherubini. D., Fanni. A., Montisci. A., Testoni. P., "A fast algorithm for inversion of MLP networks in design problems", *COMPEL. Int. J. Comp and Math in Electric and Electro Eng*, Vol: 24, No: 3, pp. 906-920, 2005.

[13] Chari. M. V. K., Salon. S. J., *Numerical methods in electromagnetism*, CA: Academic, San Diego, 2000.

[14] Silvester. P. P. and Ferrari. R. L., *Finite elements for electrical engineers*, Univ Press, Cambridge, 1996.
[15] Raida. Z., Modeling EM tructures in the neural network toolbox of MATLAB, *IEEE Antenna's and propagation Magazine*, Vol: 44, No: 6, pp. 46-67, 2002.
[16] Partial Differential Equation Toolbox user's guide, for use with MATLAB, The Math Works Inc.
[17] Han. W., Que. P., "2D defect reconstruction from MFL signals based on genetic optimization algorithm", *Proc. 1st IEEE Int Conf. Industrial Technology, China, 2005, pp. 508-513.*
[18] Chady. T., Enokizono. M., Sikora. R., Todaka. T., Tsuchida. Y., "Natural crack recognition using inverse neural model and multi-frequency eddy current method", *IEEE Trans. Magnetics*, Vol: 37, No: 4, pp. 2797-2799, 2001.
[19] Hagan. M. T., Menhaj. M., "Training feed-forward networks with the Levenberg-Marquardt algorithm", *IEEE Trans Neural Networks*, Vol: 5, No: 6, pp. 989-993, 1994.