Characterization of Magnetic Flux Leakage Testing Signals by the Modified Hopfield Neural Network

LI Benliang[1], DU Zhiye[2], LIU Jian[2]
1. Henan Electricity Survey & Designing Institute, Zhengzhou, China
2. School of Electrical Engineering, Wuhan University, Wuhan, China
e-mail: xcivy@sohu.com

Abstract: Magnetic flux leakage technique is used commonly to detect and characterize defects in oil transmission pipeline. A novel approach is presented for training a Hopfield neural network for the 3 three-dimensional characterization of defects from magnetic flux leakage (MFL) signals. To improve the storage capacity, an original Hopfield neural network is modified by adding additional positive self-feedbacks. The robust-2 pseudo-inverse learning rule is employed to training the modified network. The results indicate that significant advantages over original neural network based defect characterization schemes could be obtained, in that the correction percent of the predicted defect profile can be controlled by the parameter of the network. The performances of the proposed method are evaluated by extensive computer simulation and the simulation results confirm the validity of the approach.

Keywords: magnetic flux leakage; modified hopfield network; defect characterization; pseudo-inverse learning rule

1 Introduction

Magnetic flux leakage testing (MFLT) method is one type of electromagnetic nondestructive evaluation (NDE) technique which is widely used in the testing the integrity of ferrous steel plates and pipes. A generic MFLT system consists of oppositely polarized permanent magnet poles or DC excitation coils to magnetize to saturation the object under inspection. Defects and in homogeneities in the test object cause the magnetic flux to "leak out". The leakage flux is measured using an appropriate set of sensors such as Hall element or coils [1–3]. Figure 1 shows a sketch of coils MFLT system typically used in the inspection of steel pipe.

As the MFL system scans the pipe, information from the Hall sensor's is recorded together with the exact location where the sensor reading was taken. The sensors' signals will be processed to determine the type of defect interms of its width, length and depth.

The pattern of the signal's signals is strictly connected to the type of defect presented by the examined structure. It is possible to define a class of curves for each single type of defect. In order to identify and estimate accurately sorts and sizes of defects, various shapes of defects, as for instance notch, grooving and crack have to be tested. The tested data of the defects represent the pattern of the induced voltage versus the time. Figure 2 shows the measured signal curve of notch, which contains 1000 tested data.

The typical curve of crack and grooving are shown in Figure 3 and Figure 4 respectively. It is notable that the signals are corrupted by noise which is caused by

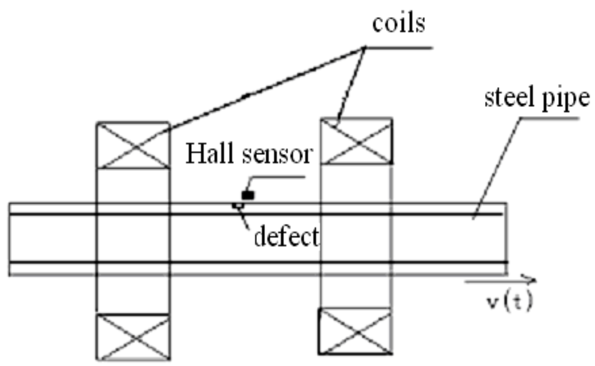

Figure 1. Configuration of the MFL testing equipment

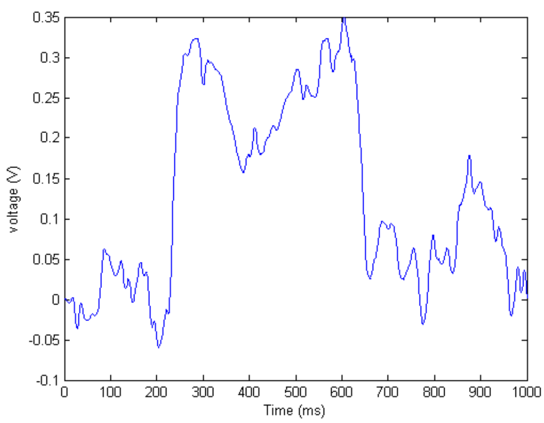

Figure 2. Signal of notch

978-1-935068-04-4 © 2009 SciRes.

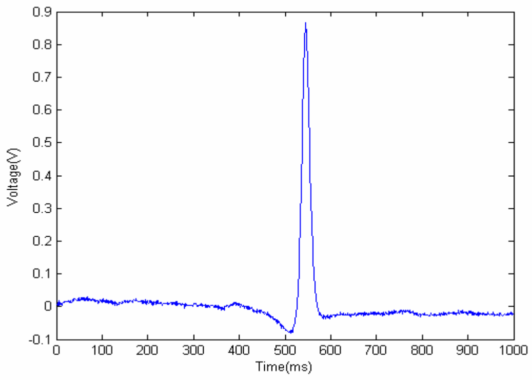

Figure 3. Signal of crack

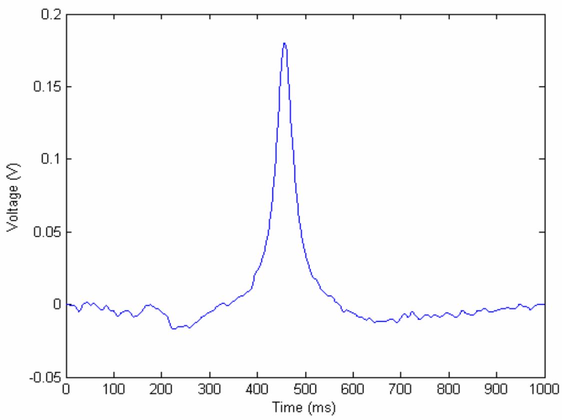

Figure 4. Signal of grooving

disturbance and measure error .

Given any of the measured curves, as for example, a curve got from the test affected seriously by noise or corrupted heavily; we wish to associate it to one of the measured curves corresponding to the different performing automatic defect identification. It is necessary to construct a logical structure that is able to memorize a number of tested data of various classes of defects and to record the stored information. When a curve that is similar to one of the memorized curves is input to the structure, it always outputs the corresponding curve, hence determine out the shape of the defect the input curve represents. Artificial Neural Networks (ANN) are proven particularly to be the most suitable selection for this problem [4,5].

Due to the auto-associative memory feature, the Hopfield ANN can associate an input pattern to one of the stored pattern which is characterized by some proximity properties induced by the used matrix. Considering the performance efficiency and RAM cost [6], a binary Hopfield ANN is used here.

2 Auto-Associative Function of Hopfield Network

2.1 Auto-Associative Memory

Just like human memory, an auto-associative neural network is able to associate an output vector Y to a certain input vector X quickly [11]. If the network is structured suitably, it should associate the right output Y even if the input X is corrupted heavily by noise. The information which is necessary for a right associative process should be stored in the network in advance. In the other words, all the possible output Y the network would to recall has to be contained in the network in term of the CAM manner. After a set of memory patterns are learned by the network, a presentation of a noisy input causes the network to recall a memorized pattern in a successful retrieval.

An associative network has to be able to associate an output Y that is similar to an input X , even the X is not complete or just a little similar to the Y. For a vague input X , the network can also associate a clear output Y. If an associative network has capacity m , that means it can recall m output Y corresponding to different input X ,

$$Y = Y_1, Y_2, \cdots Y_m \, , X = X_1, X_2, \cdots X_m \qquad (1)$$

There is an analogy between an associative memory and a hyper-surface in a n -dimensional space which has m minima corresponding to the output Y. The surface represents the network energy, where the energy is the current network state and the m minima are stable states of energy, called an attractor. Exceeding the definite capacity of the network for the m stored Y will result in undesirable minima for the energy surface or incorrect associative process which gives a wrong output. In fact an associative network is able to recall the desired output $Y = X$ starting from an input:

$$X' = X + \Delta n \qquad (2)$$

Only if Δn is rather small. Otherwise, the network possibly converges to a wrong attractor Y'. According to [6], a binary Hopfield network can be used to construct a auto-associative network. In this case, the similarity between two input X and X' is measured by the Hamming distance:

$$H = \sum_j \left| x_j - x'_j \right|, \quad j = 1, 2, \cdots n \qquad (3)$$

The network is able to perform correct association starting from a noisy input only if the Hamming distance *H* meets a preset criterion. A large value of *H* may cause in a failure associative process that is should be avoided in practice.

2.2 Hopfield Neural Network

The Hopfield ANN was invented by J.J.Hopfield in 1982[7,8].Since Hopfield and Tank's works, This kind of network has been widely used for the content address memory(CAM) implementations and, more interestingly, has attracted a lot of interests duo to its advantage over other approaches for solving optimization problems[9,10].

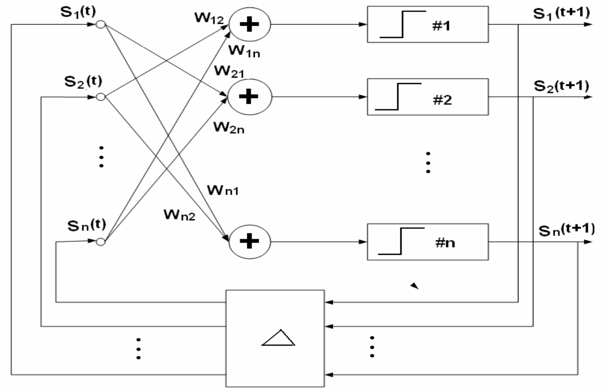

Figure 5. Original Hopfield neural network

As shown in Figure 5, the original Hopfield ANN is composed of highly interconnected nonlinear processing elements, so called "neurons". The output of each neuron is fed back to all the other neurons via weights denoted W_{ij} . The weights (connection strengths) matrix are symmetric with zero diagonal elements (i.e., $W_{ij}=W_{ji}$, $W_{ii}=0$) ,the information flows in both directions and a neuron is not self-connected. Each neuron outputs a nonlinearly transform version of the weighted summation of the neurons that are interconnected to it. The nonlinear transformation is defined by a hard limiting nonlinear function. A binary Hopfield network consists of N neurons that have two states: firing and quiescent, or, $s_i=\pm 1$, where $i=1,2,\cdots N$. Each neuron receives signals from its neighboring neurons, and the signals are transmitted through synaptic weights W_{ij} formed from the bipolar memory vectors $T_a, T_a \in \{-1,1\}^N, a=1,2,\cdots M.$, to be stored in the system. The weights matrix W is computed using the Hebbian rule:

$$W=\frac{1}{N}\left(\sum_{a=1}^{M} T_a T_a^{'} - MI_N\right) \tag{4}$$

where I_N is the $N \times N$ identity matrix. The neuron then either fires if the total input exceeds a threshold, or remains quiescent otherwise. At the time step t, neuron i receives the inputs from other neurons and output:

$$v_i(t)=\sum_{j=1,j\neq i}^{n} W_{ij}s_j(t)-\theta_i \tag{5}$$

where the element W_{ij} is the symmetric interconnection strength from neuron *j* to neuron i, θ_i is the offset of the neuron *i*. The iterative recall of the network is randomly and asynchronously processed by

$$s_i(t+1)=\operatorname{sgn}(v_i(t)) \tag{6}$$

where $sign(x)=1$ for $x>0$
and $sign(x)=-1$ otherwise, until the stable state is reached.

2.3 Modified Hopfield Networks

In order to improve the storage capacity and error correcting capability, we present a modified Hopfield neural network architecture [10]. In this architecture, additional positive self-feedbacks are added to the original Hopfield networks. So, the Equation (5) is modified as follows:

$$v_i(t)=\sum_{j=1}^{n} W_{ij}s_j(t)-\theta_i \tag{7}$$

where all the elements are the same as the mentioned above except for the weights matrix W including positive diagonal element. It is proved that the modified Hopfield neural network architecture converges like the conventional Hopfield neural network according to Yong Li [10].

The weights matrix W can be calculated by means of the robust-2 pseudo-inverse learning rule [12,13], which is with more recall accuracy for noisy vector inputs than the standard pseudo-inverse learning rule. According to [13], we can produce a memory matrix which yields a desired output recollection vector when multiplied by a stored vector, or a noisy version of the vector. Supposing the input pattern vector x_a and output pattern vector y_a, for $a=1,\ldots P$, form an associated/recollection pair. Defining matrices $X(N\times P)$ and $Y(K\times P)$ with P key and recollection vector as their column, we desire weights matrix $W(K\times N)$ satisfying $Y=WX$ .Hence, the W is given by:

978-1-935068-04-4

$$W = Y(X^T X)^{-1} X^T = YX^+ \tag{8}$$

where X^+ is the standard pseudo-inverse of X, which minimizes

$$J_1 = \|Y - MX\|^2 \tag{9}$$

For inputs that are key vectors with additive independent identically distributed noise, an optimal associative processor (AP) had been found by minimizing the criterion function with respect to W

$$J_2 = \|Y - W(X+N)\|^2 \tag{10}$$

where $N(N \times P)$ is an noise matrix. The solution is

$$W = YX^T(XX^T + P\sigma_n^2 I)^{-1} \tag{11}$$

where σ_n^2 is the input noise level specified during synthesis, and the matrix inverse always exists as long as σ_n^2 is not equal to zero. It is proven that this robust-2 pseudo-inverse robust AP improves performance for a wide range of input noise strength (both less than and larger than the σ_n^2 used in synthesis).

After the weights have been calculated, starting from an input curve prototype, the network evolves to a minimum energy stable state corresponding to the prototype associated to that input. When the association process converges to the correct result, the input curve is classified correctly, thus identified successfully.

3 Data Processing and Simulation Results

As mentioned above, the modified Hopfield neural networks contains two-state neurons whose value assume -1(off) or+1(on), thus both training and test signals must be suitable stored using bipolar arrays. The available training set curves should transform into the suitable format which is consistent with the input of identifier. We need to represent the training set data in bipolar vector -1 and +1. We represent each curve as a n-element linear vector. Considering the simulation efficiency, the neurons of the Hopfield network must be limited .It is necessary to reduce the data of training set curves without loss useful information too much. Both the dimensionality reduction and values quantization are taken to keep fairly limited the number of neurons and to obtain a suitable representation of the data. The size of the available curve is reduced by defining a window where they assume interesting values (maxima, minima) and by upper-sampling them within the window, which is with length of nx and width of ny and contains the important data describing the defect on this curve. The nx means a predefined step number of sample, ny is a predefined number of levels. In the same window, we define a matrix whose column number equals the samples number nx and row number equals quantization levels number ny. The entries of each column vector equal -1 except where the curve intercrosses the grid(value=+1).Finally these column vectors are stacked to give only one column vector of dimension $nx*ny$ represent pattern of the curve. As shown in Figure 6

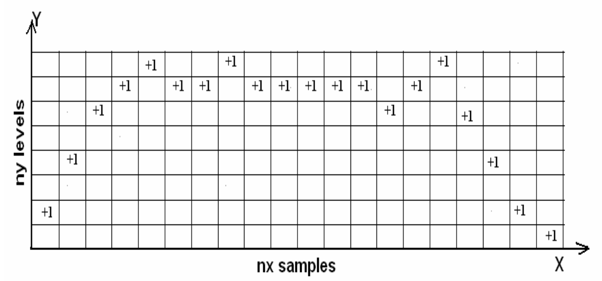

Figure 6. Upper-sampling and quantization of a curve of notch class

the curve of a notch defect is upper-sampled and quantized in the window, which is with $nx = 20$ and $ny = 8$ respectively. It is noted that the value +1 represents the curve crossing with the grid. The other entries of each column vector are equal to -1 that is represented by blank in the Figure 6 for seeing clearly. The curve is then denoted by a column vector with 160 elements containing only bipolar value of -1 and +1.

In practical test, we obtain a “larger” MFLT signal in the case that the width or the depth of defect is very large. As shown in Figure 2, the signal of a notch class is so wide that we have to select a larger value of nx to describe completely the notch than that of the crack in Figure 3. In addition, the amplitude values of signal which are associated with the value of ny change distinctly with depth of defect [1] .For giving attention to the signals of different dimension of defects, two methods are taken and compared .we first make an invariable window with length of 600and width of 1 for all of curves. In this case, we change sample nx from 25 to 100 with skip of 25, and levels ny from 10 to 20 with skip of 5, thus the neurons of network change from 250 to 2000. Second, we make the size of window changing with the curves. For example, we set the length and width of window to be 500 and 0.35 when we deal with the curve shown in Figure 3. For the curve of Figure 2, the values are set to be 80 and 0.1 correspondingly. Then we change the values of nx and ny as the case 1, as shown in Table 1.

Table 1. Results of correct classification percentage

Tes ItD	Window size	Sample nx	Level ny	neurons	Correct association percentage
A1	invariable	25	10	250	28512.8
A2		50	10	500	22.6
A3		75	10	750	30.8
A4		100	10	1000	38.7
A5		25	15	325	15.3
A6		50	15	750	26.4
A7		75	15	1125	38.8
A8		100	15	1500	46.2
A9		25	20	500	25.6
A10		50	20	1000	45.7
A11		75	20	1500	59.7
A12		100	20	2000	76.8
B1	variable	25	10	250	16.9
B2		50	10	500	30.8
B3		75	10	750	45.2
B4		100	10	1000	62.6
B5		25	15	325	22.6
B6		50	15	750	42.8
B7		75	15	1125	65.8
B8		100	15	1500	86.6
B9		25	20	500	26.1
B10		50	20	1000	55.6
B11		75	20	1500	78.8
B12		100	20	2000	85.3

The MFL signals were obtained by means of a 3-D finite element model [1], and include crack, notch, and grooving shapes of defects with varying widths and depths. 300 MFL simulated signals and 300 its randomly noisy version were used to train the neural networks [14]. In our simulation any input test-set pattern evokes an evolution of the internal state on the network towards a minimum Hamming distance state corresponding to one of the stored pattern. Sometimes the result of a test is incorrect due to the coarse approximation provided by the chosen set of size-reduction parameters. Correct identification percentage for the performed experiments based on various size-reduction parameters is shown in Table 1. It is known that the case 2(B8) with $nx = 100$ and $ny = 15$ is the best selection. The number of signals that can be stored in the networks is $0.866 * 600 \approx 520$. Thus, the memory capacity is about $0.3N$, N being the number of neurons in the networks, which is greater than that of original Hopfield networks. In practical projects, we have taken the parameters of the case to obtain well solution.

4 Conclusions

We have presented a modified Hopfield neural network architecture for identifying of MFLT signals, and showed its effectiveness by simulation experiments. The simulation results showed that the proposed method was performing efficiently. The main conclusion emerged from the simulation results can be summarized as follows:

- The modified Hopfield network have higher memory capacity and looks reliable as the convergence to a stable solution usually happens rather rapidly in identification of the MFL signals.

- The correct identification percentage usually increases with the growing of neurons, the number of considered samples is often very important in order to achieve good results.

References

[1] DU Zhiye, RUAN Jiangjun, WANG Xianqin, Research on 3D FE Simulation in MFL testing of Steel Pipe Based on Software Ansoft.NDT Vol.27,No.6, (2005)281–285.

[2] WANG Yousheng, PAN Mengxian, HE Fuyun.:Experimental research on the relation between defect parameters and magnetic leakage signal. Journal of Hefei University of Technology Vol 21 (1998) 28–31.

[3] N.Ida and W.Lord.:3-D finite element predictions of magnetostatic leakage fields. IEEE Trans.Magn, vol. MAG-19, (1983) 2260–2265.

[4] P. BURRASCANO, E. CARDELLI, D. FAZZIOLI, AND S. RESTEGHINI.:Neural network identification of defect in RFEC Technique. Proc. of E'NDE 98, (1998) 32–33.

[5] R. SIKORA, T. CHADY AND J. SIKORA.:Neural Network Approach to Crack Identification. Int. Journal of Applied Electromagnetics and Mechanics, Vol. 9, (1997)391 – 398.

[6] J. J. HOPFIELD E D.W. TANK.:Neural Computation of Decision in Optimization Problems. Biological Cybernetics, Vol. 52, (1985)141–152.

[7] J. J. Hopfield. Neural network and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci. USA, vol. 79, (1982)2554–2558.

[8] J. J. Hopfield.:Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Nat. Acad. Sci. USA,vol. 81, (1984)3088–3092.

[9] X. C. Zeng and T. Martinez.:A new relaxation procedure in the

Hopfield neural networks for solving optimization problems.Neuron ProcessingLett., vol. 10,(1999)211–222.

[10] Yong Li, Zheng Tang, GuangPu Xia, and RongLong Wang.:A Positively Self-Feedbacked Hopfield Neural Network Architecture for Crossbar Switching. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, Vol. 52, NO. 1, (2005).200–206.

[11] S. Barcherini, L. Cipiccia, M. Maggi, S. Fiori, and P. Burrascano.:Non-Destructive Test by the Hopfield Network. IJCNN 2000 Vol.6, (2000) 381–386 .

[12] D. Casasent and B. Telfer.:High capacity pattem recognition associative processors.Neural Networks, vol. 5, no. 4, (1992) 687–698.

[13] Brian A. Telfer and David P. Casasent.:Fast Method for Updating Robust Pseudoinverse and Ho-Kashyap Associative Processors.IEEE TRANSACTIONS ON SYSTEMS，MAN,AND CYBERNETICS, Vol. 24, NO. 9, (2005)1387–1390.