Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, Madrid, Spain, February 15-17, 2006 (pp26-32)

The problem of signal denoising for detecting the presence of spikes

VINCENZO NIOLA
Department of Mechanical Engineering for Energetics
University of Naples "Federico II"
Via Claudio, 21 - 80125 Napoli
ITALY
http://niola.dime.unina.it

ROSARIO OLIVIERO
University of Naples "Federico II"
Via Pansini 5, 80131, Napoli
ITALY

GIUSEPPE QUAREMBA
University of Naples "Federico II"
Via S. Pansini, 5 – 80131 Napoli
ITALY

Abstract: **-** A Self Learning Neural Network was designed to perform the denoising of signals and to detect the anomalies superimposed to signals obtained mathematically from sinusoidal functions. The ability of the network to perform a good denoising process was tested by adding random white noise to the original signals. The performance of the denoising process was evaluated by decomposing orthogonally the spiked and noisy signals by means of the wavelet transform, of which the ability of investigating on such anomalies is well known.

Key-words: - Neural network, signal processing, wavelet analysis, signal denoising, diagnostics.

1 Introduction

Nowadays, one of the most important requirement from a dynamical point of view is the high performance of any mechanical system operating in extreme dynamical conditions [1].

In order to prevent these kind of problems and to face these challenges, several test bed, numerical simulations and mathematical models have been proposed in the last years. They are the basis for the development of new methodologies useful both for understanding the phenomena representing the main limit of mechanical system and for developing new and more reliable predictive test for diagnostics.

The present work proposes a method to detect the existence of spikes that usually are generated in a mechanical system by fatigue crack. The importance of revealing the presence of such anomalies also in presence of noised signals is fundamental, today, in order to prevent mechanical failure and to improve the reliability of dynamical systems. Starting from these considerations some mathematical functions were developed in order to simulate the dynamical response of a mechanical system. Then a neural network was designed in order to perform an *intelligent* and *soft* denoising of signals, to which a white random noise was superimposed for simulating real operating conditions. It will be shortly described. Finally, the neural output was analyzed by means of wavelet transform for detecting the presence of anomalies usually known as spikes.

2 A Mathematical and Fuzziness overview

2.1 Discrete Wavelet Transform

Mother wavelets are special functions, whose first h moments are zero [2]. Note that, if ψ is a wavelet whose all moments are zero, also the function ψ_{jk} is a wavelet, where

$$\psi_{jk}(x) = 2^{-j/2}\psi(2^j x - k) \ . \tag{1}$$

Wavelets, like sinusoidal functions in Fourier analysis, are used for representing signals. In fact, consider a wavelet ψ and a function φ (father wavelet) such that $\{\{\varphi_{j_0k}\}\}$, $\{\psi_{jk}\}$, $k \in$ Z, j = 0, , 2,…} is a complete orthonormal system [3],[4]. By Parseval theorem, for every signal $s \in L^2(\mathbf{R})$, it follows that

$$s(t) = \sum_k a_{j_0 k} \varphi_{j_0 k}(t) + \sum_{j=j_0}^{j_1} \sum_k d_{jk} \psi_{jk}(t). \qquad (2)$$

In particular, the decomposition of signal $s(t)$ performed by means of the Discrete Wavelet Transform (DWT) is represented by the detail function coefficients $d_{jk} = <s, \psi_{ik}>$ and by approximating scaling coefficients $a_{j_0 k} =< s, \varphi_{j_0 k} >$. Observe that d_{jk} can be regarded, for any j, as a function of k. Consequently, it is constant if the signal $s(t)$ is a smooth function, having considered that a wavelet has zero moments. Lemma 5.4 in [5] implies the recursive relations

$$a_{jk} = \sum_{m \in Z} h_{m-2k} a_{j+1,m} \qquad (3)$$

and

$$d_{jk} = \sum_{m \in Z} \lambda_{m-2k} d_{j+1,m}, \quad (4)$$

where $\lambda = (-1)^{k+1} h_{1-k}$; $\{ h_k, k \in Z\}$ are real-valued coefficients such that only a finite number is not zero and they satisfy the relations

$$\sum_{k \in Z} h_{k+2m} \overline{h_k} = \delta_{0m} \qquad (5)$$

$$\frac{1}{\sqrt{2}} \sum_{k \in Z} h_k = 1 \qquad (6)$$

The sequence of spaces $\{V_j, j \in Z\}$, generated by φ is called a multiresolution analysis (MRA) of $L^2(\mathbf{R})$ if it satisfies the following main properties

$$V_j \subset V_{j+1}, j \in Z \text{ and } \bigcup_{j \geq 0} V_j \text{ is dense in } L^2(\mathbf{R}).$$

It follows that if $\{V_j, j \in Z\}$, is a MRA of $L^2(\mathbf{R})$, we say that the function φ generates a MRA of $L^2(\mathbf{R})$, and we call φ the father wavelet.

The relation (2) is also called a multiresolution expansion of s. This means that any $s \in L^2(\mathbf{R})$ can be represented as a series (convergent in $L^2(\mathbf{R})$), where a_k and d_{jk} are some coefficients, and $\{\psi_{jk}, k \in Z\}$, is a basis for W_j, where we define

$$W_j = V_{j+1} - V_j, j \in Z.$$

$\{\psi_{jk}(t)\}$ is a general basis for W_j. The space W_j is called resolution level of multiresolution analysis. In the following, by abuse of notation, we frequently write "resolution level j" or simply "level j". We employ these words mostly to designate not the space W_j itself, but rather the coefficients d_{jk} and the function ψ_{jk} "on the level j".

Furthermore, mathematical functions were used for generating the data to be analyzed both in the case of a new mechanical system and in the case of cracked one. As an example of such functions is reported below.

The function generating two spikes is as follows

$$y(t) = sin(4t) + 0.9|t-10|^{0.2} - 1.2|t-11|^{0.4} \quad (7)$$

where t ∈ [0, 20] (*i.e.*, 2001 values). It simulates the response of a mechanical system when fatigue cracks are present.

The target signal (see next paragraph) is obtained by considering

$s_1(t) = 10|\, t - 7|^{0.2}$

and

$s_2(t) = 1.2|\, t - 11|^{0.4}$

where t ∈ [0, 20] (*i.e.*, 11 values). Therefore we consider the signals $y_1(t)$ and $y_2(t)$, composed by 2001 points, obtained by cubic spline interpolation of s_1 and s_2, respectively.

Finally, we consider the smooth approximation z of y as follows

$z(t) = sin\,(4t) + y_1(t) - y_2(t).$

Several tests were performed with different functions. It is important to note that, in accordance with the signal processing theory, the wavelet analysis, of course, was applied by ignoring the transient part due to the convolution between the signal and filter, which is located at beginning and at the end of convolution process.

2.2 A brief description of Neural Network: The Multilayer Perceptron

The perceptron can be thought like a net composed of elementary processors organized in such a way to recreate the biological neural connections. It is able to learn, to recognize and to classify signal in

independent way. Observe that the nodes of two consecutive levels are connected by one link (or weight) but no connection exists among nodes belonging to the same level. The level where the nodes of input are present is named input layer, while the level which shows the output is said output layer. The layers which lies between the input and the output layers are named hidden layers[6][7].
The output of nodes of one layer is transmitted to the correspondent nodes of the following layers by means of links (weights) which can amplify, attenuate or inhibit such output through weighted factors. With the exception of nodes of the input layer, the total input for each node is the sum of the weighted output of nodes belonging to previous layer. Each node is activated in agreement with the input received from both the other nodes and the activation function. The total input of the *i*-th node of one layer is

$$I_i = \sum_j w_{ij} o_j \qquad (8)$$

where o_j is the output of j-th neuron of the previous layer and w_{ij} is the weighted link between the *i*-th node of one layer and *j*-th node of the previous layer.
The output of the *i*-th node is

$$o_i = f(I_i)$$

where $f(\cdot)$ is the activation function. Generally the activation function is sigmoidal as shown in Fig.1.

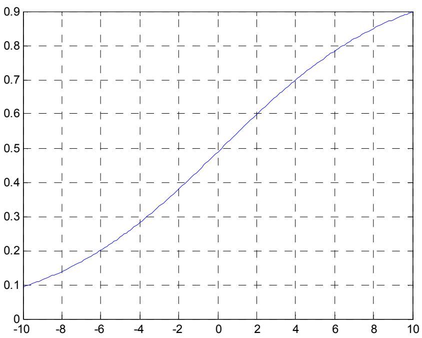

Fig.1 Sigmoidal activation function

The function is symmetrical around ϑ and ϑ_0 controls the degree of steepness of the activation function (*i.e.*, value of threshold / bias). In our application the activation function was set as follows

$$f(\cdot) = \frac{1}{\sqrt{2\pi}} e^{\frac{\overline{X}-x}{5\pi\sigma}} \qquad (9)$$

where $\overline{X}$ and σ were the mean and the standard deviation respectively of the values of the nodes selected by the neighbourhood system.
During the training set, the signal $X=\{x_i\}$ was submitted as input to the net, where x_i is the *i*-th component of vector *X*. In general, the output $\{o_i\}$ is not the same if compared with the target t_i . For a specific target the error can be estimated as

$$E = \frac{1}{2}\sum_i (t_i - o_i)^2 .$$

The procedure, in order to learn the correct set of weights, is to vary them in such a way that *E* is minimized as quickly as possible. From a mathematical point of view this means that gradient of *E* must be negative [8]:

$$\Delta w_{ji} \ \alpha - \frac{\partial E}{\partial w_{ji}} = \eta \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial I_i} \cdot \frac{\partial I_i}{\partial w_{ji}} = \eta \ \delta_j \ \mathrm{o_i}$$

with:

$$\delta_j = -\frac{\partial E}{\partial I_j} = -\frac{\partial E}{\partial o_j} \cdot \frac{\partial o_j}{\partial I_j} = -\frac{\partial E}{\partial o_j} f'\left(I_j\right) .$$

Since *E* can be calculated directly on the output layer, the variation of weights for the links connected to the output layer is

$$\Delta w_{ji} = \eta \left(-\frac{\partial E}{\partial o_j} \right) \cdot f'\left(I_j\right) o_i .$$

In particular, if

$$o_j = \frac{1}{1 + e^{-\left((\sum_i w_{ji} \left(o_i - o_j \right) \right)}}$$

then

$$f'\left(I_j\right) = \frac{\partial o_j}{\partial I_j} = o_j \left(1 - o_j\right) .$$

Finally, we have

$$\Delta w_{ji} = \begin{cases} \eta\left(-\dfrac{\partial E}{\partial o_j}\right) o_j \left(1-o_j\right) o_i \\ \eta\left(\sum\limits_k \delta_k w_{kj}\right) o_j \left(1-o_j\right) o_i \end{cases}$$

for the output layer and other layers, respectively.
The greater value of η, the quicker is the learning but it should procure strong oscillations in the response. For that reason the last relation can be modified as follows [9]

$$\Delta w_{ji} = (t+1) = \eta \cdot \delta_j \cdot o_i + \alpha \Delta w_{ji}(t)$$

where the term $(t+1)$ is used in order to indicate the time $(t+1)^{th}$ and α is a constant of proportionality.
The neighborhood system N_{ij}^d, for a vector of N values and a generic element, is defined as

$$N_{ij}^d = \{(i,j) \in L\}$$

such that

$$(i,j) \notin N_{ij}^d$$

$$(k,l) \in N_{ij}^d \Rightarrow (i,j) \in N_{kl}^d$$

For $d = 6$, then N_{ij}^6 can be obtained by considering the 6 values like in Fig.2.

6	3	1	(i,j)	1	3	6

Fig. 2 Neighborhood system

2.3 The SLNN Architecture

The SLNN was developed by constructing 3 layers. Each layer has N neurons (the length of the signal). Each neuron corresponds to one value. Between the input and output layer there exists one layer indeed. The neurons belonging to the same level do not have any link between them. Each neuron of one layer is connected to the correspondent neuron of the previous layer and to its nearest neurons. Moreover, each neuron of the output layer, is connected to the correspondent neuron of the input layer.
The input to a neuron belonging to the input layer is given from a real number in the range $[0,1]$ proportional to the correspondent value of processed signal. Since we are interested to eliminate the noise and to extract the original spiked signal, all the weights, initially, were put to be equal to 1. The value assigned to ϑ, used in the activation function, was $\vartheta = n_l / 2$, where n_l is the number of neighbor neuron. In the Fig. 3 is depicted the scheme of neural multilayer architecture.

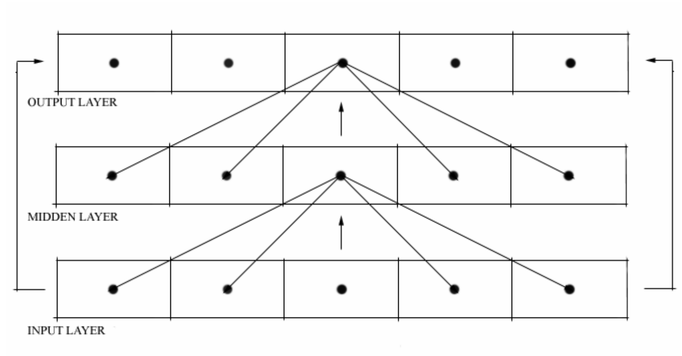

Fig.3 Scheme of neural multilayer architecture

The input value I_i for each neuron belonging to the i-th layer (except the input layer) was calculated through (8).
The goal is to obtain as output, the greatest number of neurons, set from 0 to 1, proportional to the target (*i.e.*, spikeless) signal. Therefore we will say that the state of the output level can be thought like a fuzzy set. The measure of fuzziness of such a set can be considered like the error of instability of the entire system (*i.e.*, neural network). Therefore we can use the fuzziness value as a measure of the error produced by the system and use the back-propagation in order to adjust the weights until to eliminate the error (fuzziness). The measure of E can be adopted as a meaningful function of fuzziness index

$$E = g(I),$$

where I is the measure of fuzziness of a fuzzy set. After a first adjustment of the weights, the output of the neurons belonging to the output layer is used as feedback to the correspondent neurons belonging to the input layer. In the same way the second iteration will proceed. The iteration of weight adjustment shall continue until the net becomes stabilized (*i.e.*, the fuzziness error/index becomes minimum/negligible). When the net shall be stabilized, the state of output of neurons belonging to the output layer shall assume the values from 0 to 1 proportional to the target signal.
The mathematical rules for the weight adjustment are the following (weight correction by fuzziness linear index) [10][11]

$$\Delta w_{ji} = \begin{cases} -\eta f'(I_j) o_i \text{ se } 0 \le \mathrm{o_j} \le 0.5 \\ \eta f'(I_j) o_i \text{ se } 0.5 \le \mathrm{o_j} \le 1 \end{cases}$$

Calculations were made using MATLAB 6.5, The MathWorks, Inc, Natick, Mass, Simulation Toolbox Version 2.1.2.

3 Results

In order to check the effectiveness of the proposed method, computer simulation has been done by generating spiked signals by (7) added with white random noise. The output signal obtained from the simulation was analyzed both by the SLNN and the wavelet transform; a typical signal generated by the proposed simulation with the presence of two spikes due to the fatigue crack of a mechanical system, is shown in Fig.4.

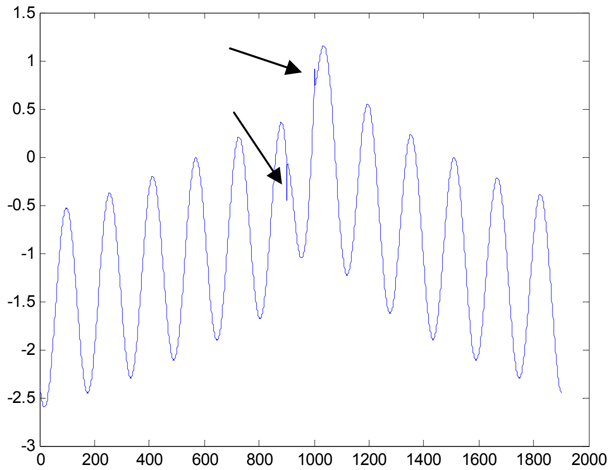

Fig. 4 Simulated spiked signal (two spikes are indicated by arrows)

To realize a more effective and reliable analysis of these kind of signals a random white noise was added to the original ones obtaining the signal depicted in Fig.5.

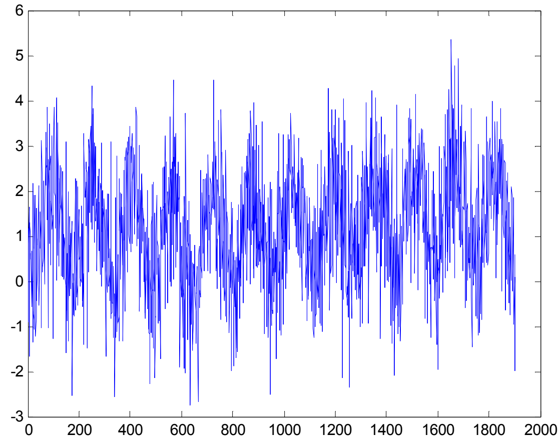

Fig. 5 Simulated output signal with random white noise added

It is impossible to distinguish any dynamics or features showed by the original signal.
The target signal is showed in Fig.6. More detailed description of neural steps is reported below.

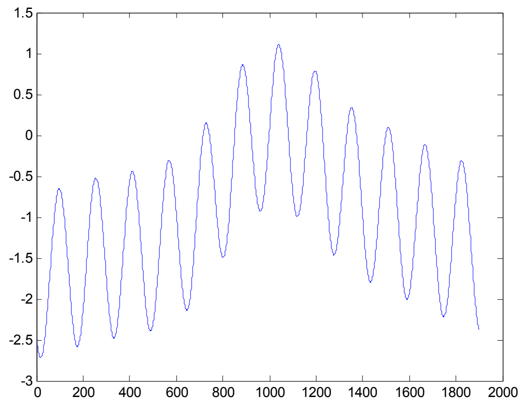

Fig.6 Target signal

The first step was the feeding of the input layers by the signal of Fig.5. This signal was passed through all the three levels forming the net. The first iteration was named “net initializing”: all the weights of the links are equals to 1. The result of the first iteration is reported in Fig.7.

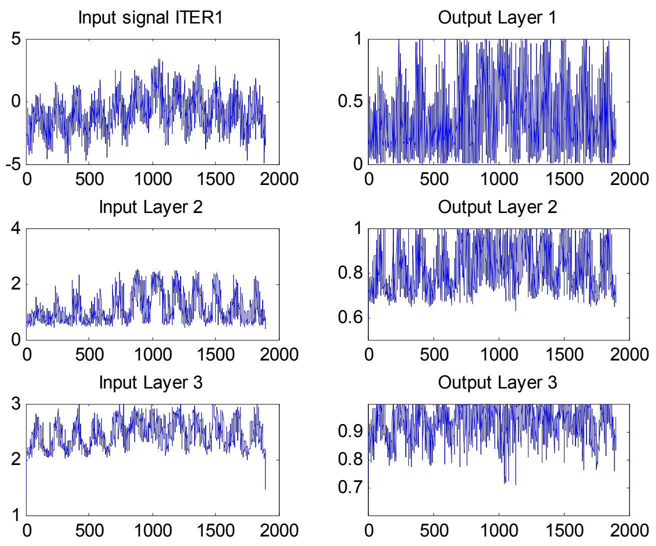

Fig. 7 Neural result for the noisy signal after the 1st iteration

The output was compared with the “target signal” (*i.e.*, the expected signal of Fig.6) to which the net tries to fit the input signal.

The last signal shown is used by the net to adjust the weights of the links for the second iteration . The weights of the links modified as explained before followed a square error minimization index.

The net converged after ten iterations producing the results depicted in Fig.7.

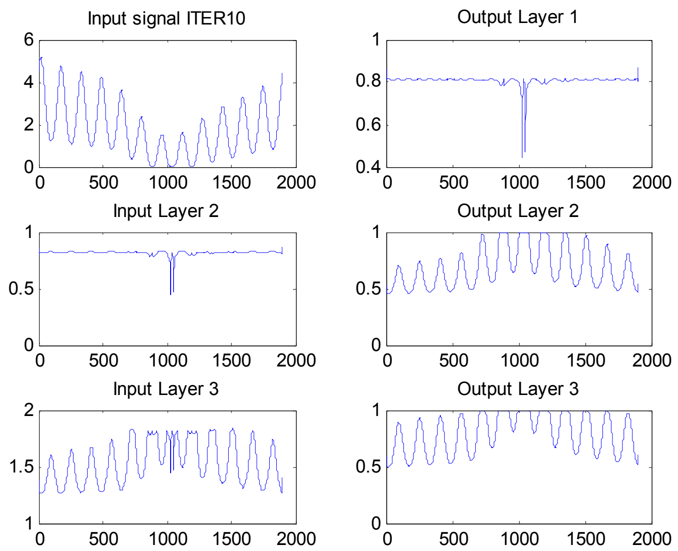

Fig. 7 Neural result obtained after the 10th iteration

If we process the signal depicted in Fig.5, by wavelet transform it is not so easy to distinguish the presence of spikes and their features (see Fig.8). By applying the SLNN, as pre-processing tool, the detection of two anomalies, from the starting noised signal of Fig.5, can be easily performed by wavelet transform [12], see fig.9.

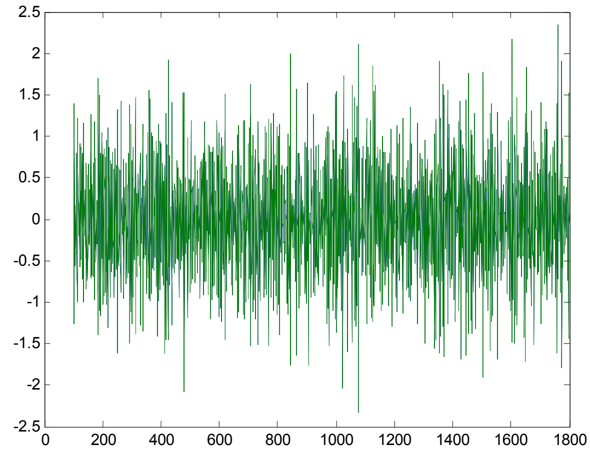

Fig. 8 Result for the 6th-level wavelet decomposition

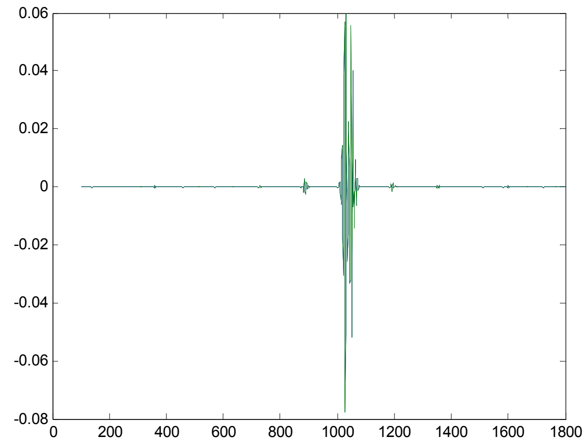

Fig. 9 Wavelet analysis of the SLNN output signal

As expected, note the presence of a consistent spike at t=1000. A minor spike is locate at t=900.

4 Conclusions

A well designed *Self Learning Neural Network* (SLNN) should contribute to determine the presence of anomalies (*e.g.*, spikes), due to fatigue crack in mechanical system, when they are of small entities.

Moreover, the present work constitutes the basis for a next experience based on the improvement of a test bed, in order to compare the results of the present work with the ones provided by a real model by applying the same methodology showed in this paper.

The importance to prevent such anomalies (*i.e.*, fatigue crack) is fundamental for the regularity of functioning and yield of a mechanical system. The methodology should be implemented on that machinery, working, usually, in extreme conditions of lubricating (*e.g.*, high temperature, high speed, instantaneous speed changing, high power transmission, and so on). However the proposed methodology may be extended to the fields where important torsional and vibrational problems (*e.g.*, navigation, automotive, etc.) are involved.

Normally, the signals are affected by noise, so it is fundamental to perform an *intelligent* and *soft* denoising process without the application of any filter. In fact, their application could suppress the presence of spikes (usually showing high frequency and small amplitudes) which reveal the potential mechanical damage. For that reason, in this work, a SLNN was designed in order to realize a reliable real time signal denoising, without the application of any digital filter.

The response performed by means of the application of Wavelet Transform is significative. In particular, it proves both the reliability of SLNN for such an application and, in conjunction with the wavelet analysis, the ability for detecting anomalous signals.

References:

[1] Niola V., Quaremba G., Oliviero R., The location of anomalies spikes on a working rotor: a theoretical – experimental test, *International Journal of Applied Mechanics and Engineering*, Vol.9, No.3, 2004, pp. 547 – 556.

[2] Daubechies I., *Ten Lectures on Wavelets*, SIAM, 1992.

[3] Antoniadis A., Oppenheim G., *Lecture Notes in Statistics – Wavelet and Statistics*, Springer, 1995.

[4] Härdle W., et al, *Lecture Notes in Statistics – Wavelets, Approximation, and Statistical Applications,* Springer, 1998.

[5] Härdle W., Kerkyacharian G., Picard D., Tsybakov A., *Wavelets, Approximations, and Statistical Applications*, Springer, Berlin, 1998.

[6] L.O. Chua, L. Yang. Cellular neural networks: Applications. *IEEE Trans. Circuits Syst.*, Vo. 35, pp. 1273-1290, 1988.

[7] T. Kohonen, *Self-organized formation of topologically feature maps.* Biol. Cybern., Vol. 43, pp. 59-69, 1982.

[8] S. Amari, *Mathematical theory of self-organization in neural nets. In Organization of Neural Networks: Structures and Models.* W.V. Seelen, G. Shaw and U.M. Leinhos. Eds.\quad New York: Academic Press, 1988.

[9] A.Ghosh, N.R. Pal, S. Pal. Self-Organization for Object Extraction Using a Multilayer Neural Network and Fuzziness Measures. *IEEE Trans. On Fuzzy Systems*, Vol. I, No. 1, Feb. 1993., pp.54-68.

[10] N. Babaguchi, K. Yamada, K. Kise and Y. Tezuku. Connectionist model binarization. In Proc. 10th *Int. Conf. Pattern Recon., 1990*, pp. 51-56.

[11] S.K. Pal, D.D.Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: John Wiley (Halsted Press), 1986.

[12] Anthony T., *Computational Signal Processing with Wavelets*, Birkhäuser, Boston, 1998.