Modelling the Development of Fluid Dispensing for Electronic Packaging: Hybrid Particle Swarm Optimization Based-Wavelet Neural Network Approach

S.H. Ling, H.H.C. Iu, F.H.F Leung, and K.Y. Chan

Abstract — **An hybrid Particle Swarm Optimization PSO-based wavelet neural network for modelling the development of fluid dispensing for electronic packaging is presented in this paper. In modelling the fluid dispensing process, it is important to understand the process behaviour as well as determine optimum operating conditions of the process for a high-yield, low cost and robust operation. Modelling the fluid dispensing process is a complex non-linear problem. This kind of problem is suitable to be solved by neural network. Among different kinds of neural networks, the wavelet neural network is a good choice to solve the problem. In the proposed wavelet neural network, the translation parameters are variables depending on the network inputs. Thanks to the variable translation parameters, the network becomes an adaptive one. Thus, the proposed network provides better performance and increased learning ability than conventional wavelet neural networks. An improved hybrid PSO [1] is applied to train the parameters of the proposed wavelet neural network. A case study of modelling the fluid dispensing process on electronic packaging is employed to demonstrate the effectiveness of the proposed method.**

I. Introduction

Recently, new kinds of neural networks known as the wavelet neural networks (WNNs), which combine feed-forward neural networks with the wavelet theory [2], have been proposed [3-5]. The wavelet theory provides a multi-resolution approximation for discriminate functions. The WNN can thus exhibit better performance in function learning than the conventional feed forward neural networks. Researchers have successfully applied WNNs in function approximation [1], robotics [4], and power systems [5]. Using neural networks to achieve learning [6] usually involves two steps: designing a network structure and deriving an algorithm for the learning process. The structure of the neural network governs the non-linearity of the modelled function. The learning algorithm determines the rules for optimizing the weight values of the network within the training period. A typical wavelet neural network structure offers a fixed set of weights after the learning process. This single set of weights is used to capture the characteristics of all input data. However, a fixed set of weights may not be enough to learn the data set if the data are distributed in a vast domain separately and/or the number of network parameters is too small. This also applies to a typical wavelet neural network structure.

S.H. Ling and H.H.C. Iu are with the School of Electrical, Electronic, and Computer, The University of Western Australia, Australia
F.H.F. Leung is with the Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
K.Y. Chan is with the Dept. of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong

In this paper, a variable translation wavelet neural network (VTWNN) is proposed. Wavelets are used as the transfer functions in the hidden layer of the network. The network parameters, i.e. the translation parameters of the wavelets, are variable depending on the network inputs. Thanks to the variable translation parameters, the proposed VTWNN has the ability to model the input-output function with input-dependent network parameters. It works as if several individual neural networks are handling different sets of input data. Effectively, it becomes an adaptive network capable of handling different input patterns, which exhibits a better performance. Fig. 1 shows the architecture of the proposed VTWNN, which consists of two units, namely the parameter memory (PM) and the data-processing (DP) neural network. The PM stores some parameters ($\boldsymbol{\kappa}$) governing how the DP neural network handles the input data. By using this proposed neural network, some cases that cannot be handled by the traditional neural networks with a limited number of parameters can now be tackled.

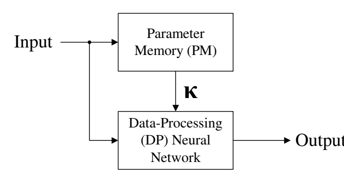


Fig.1. Proposed architecture of the neural network.

One of the important issues on neural networks is their training. The training process aims to find a set of optimal network parameters. One commonly used training method is the gradient method [6] and the back-propagation technique [6], which adjusts the network parameters based on the gradient information of the fitness function in order to reduce the errors over all input patterns. However, gradient methods may only converge to a local minimum, and is sensitive to the values of the initial parameters. The function to be optimized needs to be differentiable and the learning method may only be good to some specific network structure. Particle swarm optimization (PSO) [7] is one of the stochastic search algorithms. The error functions are less likely to be trapped in a local optimum, and need not be differentiable or even continuous. Thus, PSO is more suitable for searching in a large, complex, non-differentiable and multimodal domain. It is a good training algorithm for neural or neural fuzzy networks [8-9]. The same PSO can be used to train many different networks, regardless of whether they are feed-forward, recurrent, wavelet or other structure-type networks. This generally saves a lot of

978-1-4244-1821-3/08/$25.00©2008 IEEE

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

efforts in developing the training algorithms for different types of networks.

Recently, different hybrid PSOs have been proposed to overcome the drawback of trapping in local optima. An improved hybrid PSO with wavelet mutation [1] is proposed which the solution quality and solution stability are improved.

Fluid dispensing is a manufacturing process by which fluid materials are delivered to substrates, boards or work-pieces in a controllable manner. This process is widely used in various packaging processes in the electronics and semiconductor manufacturing industry such as integrated circuit encapsulation, die bonding and surface mount technology. In the competitive market of today, this manufacturing process needs to be controlled at each of the many processing steps in the manufacturing line. The process directly affects the overall quality of the finished product, as well as the throughput of the production line. All the variables controlling the desired outputs in a given process need to be understood and optimized for tight control. To achieve this, it is necessary to develop an accurate model for describing the process.

The advantage of using the neural network approach to process modelling that it can provide learning and generalization abilities for nonlinearities. In this paper, VTWNN trained by the hybrid PSO with wavelet mutation (HPSOWM) [1] is proposed to model the fluid dispensing process for electronic packaging. By employing the proposed method on modelling the fluid dispensing process, smaller modelling errors with smaller computational effort can be achieved comparing with the other tested modelling methods.

This paper is organized as follows: The basic theory of wavelet is discussed in Section II. In Section III, the proposed VTWNN model is presented. Also, the training of the parameters of the proposed VTWNN using hybrid PSO with wavelet mutation is presented. In Section IV, application on modelling the fluid dispensing process is given to show the merits of the proposed methodology. A conclusion is drawn in Section V.

II. Basic Wavelet Theory

Certain seismic signals can be modelled by combining translations and dilations of an oscillatory function with finite duration called a "wavelet". A continuous function $\psi(x)$ is a "mother wavelet" or "wavelet" if it satisfies the following properties:

Property 1:

$$\int_{-\infty}^{+\infty} \psi(x)dx = 0 \tag{1}$$

In other words, the total positive energy of $\psi(x)$ is equal to the total negative energy of $\psi(x)$.

Property 2:

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 dx < \infty \tag{2}$$

where most of the energy of $\psi(x)$ is confined to a finite domain and is bounded.

In order to control the magnitude and the position of $\psi(x)$, $\psi_{a,b}(x)$ is defined as:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{3}$$

where a is the dilation parameter and b is the translation parameter. It should be noted that $\psi_{a,b}(x)$ is scaled down as the dilation parameter a increases, and the location of the centre of the wavelet is controlled by the translation parameter b.

III. Design and Analysis of Variable Translation Wavelet Neural Network Model

In this section, the design and the analysis of the variable translation wavelet neural network (VTWNN) model will be presented. The wavelet neural network (WNN) can be considered as a particular case of feed-forward neural networks. The special point is that the transfer function of the WNN is a multi-scaled wavelet function $\psi_{a,b}(x)$.

In the proposed VTWNN, the translation parameter in the transfer function of the hidden nodes is variable and depends on the network inputs. With the variable translation parameters, the proposed VTWNN performs better and has higher learning ability than the conventional WNN [3] and feed-forward neural network [6].

A. Design of the Network Model

The proposed VTWNN has a three-layer structure with n_{in} nodes in the input layer, n_h nodes in hidden layer, and n_{out} nodes in output layer as shown in Fig. 2. The input of the hidden layer, S_j, is given by,

$$S_j = \sum_{i=1}^{n_{in}} z_i v_{ji}, \quad j = 1, 2, \ldots, n_h \tag{4}$$

where z_i, $i = 1, 2, \ldots, n_{in}$ are the input variables; v_{ji} denotes the weight of the link between the i-th input and the j-th hidden node. In order to control the magnitude and the position of the wavelet, the multi-scaled wavelet function $\psi_{a,b}(x)$ defined in (3) is used as the hidden node transfer function. The dilation parameter a of the first hidden node ($j = 1$) is set as 1, i.e. $\psi_{1,b_1}(x) = \psi(x-b_1)$. For the second hidden node ($j = 2$), the dilation parameter a is set as 2, i.e. $\psi_{2,b_2}(x) = \frac{1}{\sqrt{2}}\psi\left(\frac{x-b_2}{2}\right)$, where the output of the wavelet is scaled down by $1/\sqrt{2}$. Similarly, for the j-th hidden node, the dilation parameter a is set as j. Hence, the output of the hidden layer of the proposed VTWNN is given by,

$$\psi_{j,b_j} = \frac{1}{\sqrt{j}}\psi\left(\frac{S_j - b_j}{j}\right) \tag{5}$$

In this proposed network, the Maxican Hat function as shown in Fig. 3 is selected as the mother wavelet $\psi(x)$, which is defined as:

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

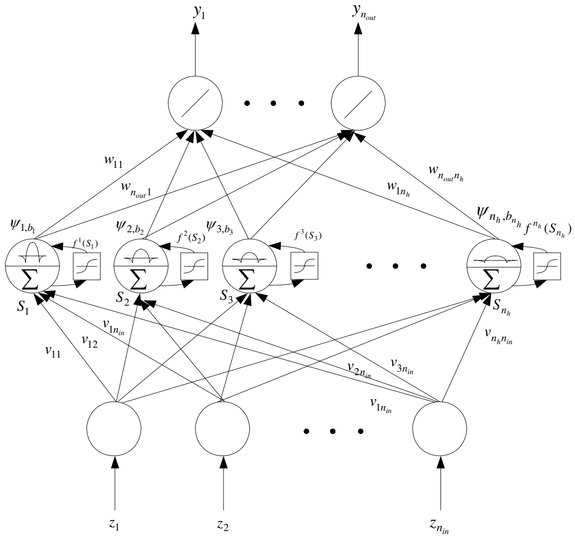


Fig.2. Variable translation wavelet neural network model.

$$\psi(x) = e^{-x^2/2}(1-x^2) \tag{6}$$

$\psi(x)$ meets the *Property 1* in (1) and *Property 2* in (2) of wavelets. Referring to (5) and (6),

$$\psi_{j,b_j} = \frac{1}{\sqrt{j}} e^{\frac{-\left(\frac{S_j - b_j}{j}\right)^2}{2}} \left(1 - \left(\frac{S_j - b_j}{j}\right)^2\right) \tag{7}$$

The translation parameter b_j is variable depending on the input S_j, and is governed by a nonlinear function $f^j(\cdot)$,

$$b_j = f^j(S_j) \tag{8}$$

We set

$$f^j(S_j) = 4 * j\left(\frac{2}{1+e^{-\kappa^j \times S_j}} - 1\right) \tag{9}$$

where κ_j is a tuned parameter which is used to control the shape of the nonlinear function $f^j(\cdot)$. The shape of the $f^j(\cdot)$ with different κ_j is shown in Fig. 4. The parameter κ is used to control the shape of the nonlinear function $f(\cdot)$ and the parameter κ governs the parameter set (Referring to Fig. 1). Figure 4 shows the effect of the tuned parameter κ_j to b_j. In general, the range of κ is tuned within 0.3 to 1.5. We see as $\kappa \to \infty$, the function reduces to a threshold function. Similarly, as $\kappa \to -\infty$, the function will become a constant line.

From (8) and (9), the value of the translation parameters b_j depends on the network inputs and the parameters κ_j. In other words, it operates such that the neural network will handle different input data with different network parameters b_j. Thus, the proposed VTWNN is an adaptive network and the network outputs depend on the network inputs.

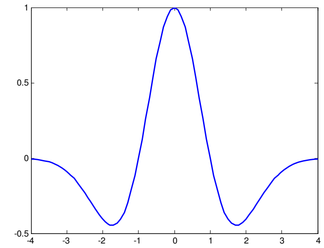


Fig. 3. Maxican Hat mother wavelet.

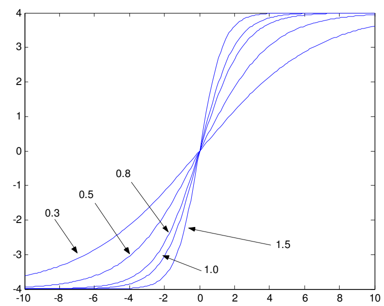


Fig .4. Sample nonlinear functions with different values of parameter κ (κ= 0.3, 0.5, 0.8, 1.0 and 1.5).

The output of the proposed VTWNN is defined as,

$$y_l = \sum_{j=1}^{n_h} \psi_{j,b_j}(S_j) \cdot w_{lj} \tag{10}$$

$$= \sum_{j=1}^{n_h} \psi_{j,b_j}\left(\sum_{i=1}^{n_{in}} z_i v_{ji}\right) \cdot w_{lj} \tag{11}$$

where w_{lj}, j = 1, 2, …, n_h; l = 1, 2, …, n_{out} denotes the weight of the link between *j*-th hidden node and *l*-th output node. The tuned parameters of the VTWNN are v_{ji}, w_{lj}, and κ_j. The number of parameters for v_{ji} is equal to $n_{in} \times n_h$; the number of parameters for w_{lj} is equal to $n_h \times n_{out}$, and the number of parameters for κ_j is equal to n_h. Thus, the total number of parameters of the proposed VTWNN is equal to $n_h(1 + n_{in} + n_{out})$.

B. Interpretation of the Network

Fig. 5 explains the operating principle of the proposed network and why it works well. In this figure, P1, P2 and P3 are three sets of input patterns. $\hat{P}_{b_j}1$, $\hat{P}_{b_j}2$ and $\hat{P}_{b_j}3$ are the input translation parameter with the corresponding input patterns. When the proposed neuron manipulates the input pattern P1, the shape of the wavelet transfer function is characterized by $\hat{P}_{b_j}1$, and the function eventually outputs the pattern P'1. Similarly, when the neuron manipulates the input pattern P2, the shape of the

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

wavelet transfer function is characterized by $\hat{P}_{b_j}2$, and the function eventually outputs the pattern P'2. So, the activation function is variable and dynamically dependent on the input pattern. Hence, the degree of freedom of the modelled function is increased. Comparing with the conventional wavelet and feed-forward neural networks, the VTWNN should be able to offer a better performance.

All the parameters of the neural network can be tuned by an improved hybrid PSO which will be discussed in the next section.

C. Tuning of the Network Parameters with Hybrid PSO

Hybrid PSO is a powerful search algorithm that has been widely applied in various optimization problems. One superior characteristic of hybrid PSO is that the detailed information of the nonlinear system, e.g. the derivative information of the fitness function is not necessarily known. Hence, hybrid PSO is suitable to handle some complex optimization problems. In this paper, the hybrid PSO with the wavelet mutation operations discussed is employed to optimize a fitness function, which is characterized by the parameters of the VTWNN. The fitness function is a mathematical expression quantitatively measures the performance of the hybrid PSO tuning process. A larger fitness value indicates a better tuning performance. By adjusting the values of the network parameters, the fitness value is maximized by using the hybrid PSO. During the tuning process, particle with better fitness values is reproduced. Also, the effect of the proposed mutation operation decreases gradually in the search domain with respect to the iteration number. This helps the convergence of the searching process of the network parameters. After the tuning process, the obtained network parameter values will be used by the proposed neural network. As the proposed neural network is a feed-forward one, the outputs are bounded if its inputs are bounded, which happens for most of the real-life applications. Consequently, no convergence problem is present for the neural network itself.

The proposed VTWNN can be used to learn the input-output relationship of an application using hybrid PSO. The input-output relationship can be described by,

$$\mathbf{y}^d(t) = g\left(\mathbf{z}^d(t)\right),\ t = 1, 2, \ldots, n_d \tag{12}$$

where $\mathbf{z}^d(t) = \begin{bmatrix} z_1^d(t) & z_2^d(t) & \cdots & z_{n_{in}}^d(t) \end{bmatrix}$ and $\mathbf{y}^d(t) = \begin{bmatrix} y_1^d(t) & y_2^d(t) & \cdots & y_{n_{out}}^d(t) \end{bmatrix}$ are the given inputs and the desired outputs of an unknown nonlinear function $g(\cdot)$ respectively; n_d denotes the number of input-output data pairs. The fitness function of the PSO depends on the application. In this paper, the fitness function is defined as,

$$fitness = \frac{\sum_{t=1}^{n_d}\sum_{k=1}^{n_{out}}\left(y_k^d(t) - y_k(t)\right)^2}{n_d n_{out}}. \tag{13}$$

The objective is to minimize the fitness value of (13) using the hybrid PSO [1] by coding the particles of the swarm to be $\begin{bmatrix} v_{ji} & w_{lj} & \kappa_j \end{bmatrix}\ \forall\ i, j, l$.

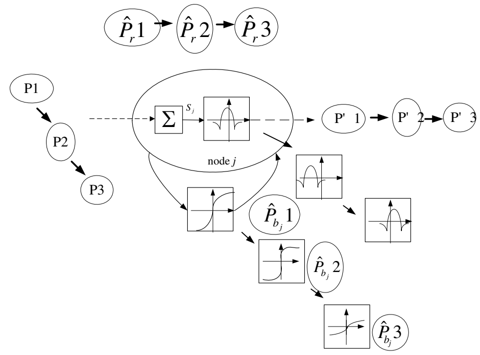


Fig. 5. Operation of the proposed neuron with 3 sets of data patterns.

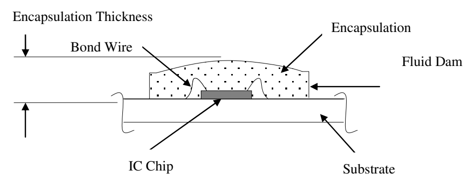


Fig. 6. Encapsulation of microchip.

IV. Modelling the fluid dispensing for electronic packaging (MFD-EP)

Fluid dispensing is an important and popular process for electronics packaging. In this paper, modelling the fluid dispensing for microchip encapsulation is studied. Normally, silicon chips are covered using an X-Y numerically controlled dispensing system that delivers fluid encapsulant through a needle. The material is commonly dispensed in a pattern, working from the centre out. A fluid dam around the die site and second wire bond points can be made to contain the flow material and make a uniform looking part as shown in Fig. 6.

Modelling the fluid dispensing process is critical for understanding the process behaviour and achieving the process optimization. The process is controlled by a number of parameters such as the diameter of the needle, the temperature of the epoxy, compressed air pressure, the viscosity of the epoxy resin, the pump motor speed, the distance between the needle and the substrate, the substrate temperature and the path of dispensing. With the advice from the supporting company, three significant process parameters, compressed air pressure, the distance between the needle and the substrate, and the pump motor speed, were studied in the research. The process parameter 'compressed air pressure' is referred to the pressure of compressed air imposing on the epoxy resin which is in the storage device of a dispensing system. The 'pump motor speed' is to control the amount of epoxy to be dispensed. The 'distance between the needle and the substrate' is controlled by using a stepping motor. Therefore, the distance is specified as number of steps in the parameter setting. The three process parameters and

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

their corresponding normal operating ranges are shown below:

- Compressed air pressure (1 bar to 4 bar), x_1
- Distance between substrate and needle (250 to 2000 steps), x_2
- Pump motor speed (400 rpm to 1000 rpm), x_3

Two quality characteristics (named as outcome variables) were also identified as shown below:

- Encapsulation weight (mg), y
- Encapsulation thickness (mm), z

96 experiments are carried out based on a full factorial design with 4 levels in compressed air pressure (x_1), 6 levels in pump motor speed (x_2) and 4 levels in the height between the substrate and the needle (x_3).

A. Modelling with Wavelet Neural Network

A variable translation wavelet neural network is used to model the fluid dispensing process. Its structure, as shown in Fig. 2, consists of an input layer in which the input vectors (including process parameters x_1, x_2 and x_3) are fed, the output layer which produces the output response (either the quality characteristic *y* or *z*), and one hidden layer in between.

According to (11) the input-output relationship of the proposed three-layer neural networks for encapsulation weight *y* and encapsulation thickness *z* can be written as follows:

$$y = \sum_{j=1}^{n_h} \psi_{j,b_j} \left(\sum_{i=1}^{3} x_i v_{ji} \right) \cdot w_{1j} \quad (14)$$

$$z = \sum_{j=1}^{n'_h} \psi'_{j,b'_j} \left(\sum_{i=1}^{3} x'_i v'_{ji} \right) \cdot w'_{1j} \quad (15)$$

where $n_h \left(or\ n_h^{'} \right)$ denotes the number of the hidden nodes; $w_{1j} \left(or\ w_{1j}^{'} \right)$, *j*=1, 2, …, $n_h \left(or\ n_h^{'} \right)$, denotes the weight of the link between the *j*-th hidden node and the output node; $v_{ij} \left(or\ v_{ij}^{'} \right)$, *i*=1,2,3 and *j*=1, 2, …, $n_h \left(or\ n_h^{'} \right)$, denotes the weight between the *i*-th input node and the *j*-th hidden node;. $\psi_{j,b_j} \left(or \psi'_{j,b'_j} \right)$ denotes the wavelet function, and $x_i \left(or\ x_i^{'} \right)$ denotes the input data for encapsulation weight and encapsulation thickness respectively.

To develop the neural network based model for the fluid dispensing process, values of the neural network parameters (i.e.: v_{ji}, $w_{lj,}$ κ_j with *i*=1, 2, 3 and *j*=1, 2, …, n_h) and the number of hidden-nodes (n_h) used in the hidden layer need to be determined. These two settings are important because they affect the prediction accuracy of the neural network based process model.

To tune the network, we use hybrid PSO to minimize the mean square error (MSE) by setting the swarm particle to be $\left[v_{ji} \quad w_{1j} \quad \kappa_j \right]$ for all *i* and *j*. The MSE for encapsulation weight *y* and for encapsulation thickness *z* are defined as follows:

$$\text{MSE}_y = \frac{\sum_{k=1}^{n_{pat}} \left(d_k^y - y_k \right)^2}{n_{pat}} \text{ and } \text{MSE}_z = \frac{\sum_{k=1}^{n_{pat}} \left(d_k^z - z_k \right)^2}{n_{pat}} \quad (16)$$

where d_k^y and d_k^z denotes the desired value of the encapsulation weight *y* and the encapsulation thickness *z* respectively; n_{pat} denotes the number of patterns. After training, the values of these network parameters will be fixed during the operation.

B. Results and Analysis

To illustrate the performance of the proposed method to this industrial application, 10-fold cross-validation is considered. Cross-validation [10] is the statistical practice of partitioning a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subsets are retained for subsequent use in confirming and validating the initial analysis. In this study, 96 experimental data of encapsulation weight and encapsulation thickness are used. In 10-fold cross-validation, the experimental data (sample) is partitioned into 10 sub-samples. 9 sub-samples are used for training and 1 sample is used for testing (validation). The cross-validation process is then repeated 10 times, with each of the 10 sub-samples used exactly once as the validation data. The 10 results can be averaged to produce an average training and testing (validation) results.

For comparison purpose, wavelet neural network [3] and feed-forward neural network (FFNN) [6] models trained by HPSOWM [1] are used to model the MFD-EP system. The basic settings of the parameters of the HPSOWM and the neural networks are shown as follows:

For HPSOWM:

- Shape parameter of the mutation [1] (ζ_{wm}): 2
- Constant (φ_1 and φ_2)of HPSOWM: 2.05
- Maximum velocity $v_{\max}$: 0.2
- Swarm size (γ): 50
- Number of runs: 2000
- Probability of mutation (p_m): 0.1

For Neural Networks:

- The initial ranges of the weights of the neural networks are bounded between –4 and 4.
- The initial ranges of the κ_j for VTWNN are bounded between 0.1 to 1.5.
- The number of hidden nodes (n_h) of the neural network for encapsulation weight and thickness are set at 5 and 7 respectively.

The average training results comparison between different neural network topologies (VTWNN, WNN, and FFNN) trained with HPSOWM for encapsulation weight and encapsulation thickness are tabulated in Table I (a) and (b) respectively. Their convergence rates are given in Fig.7. In these tables, the mean value, best value, standard deviation (S.D.), and run time in second are given. Comparing with different neural network topologies, the VTWMM gives a better performance. The average mean error of VTWNN for encapsulation

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

weight is 3.6492 which imply 45% and 75% improvement compared with WNN and FFNN. Similarly, average mean error of VTWNN for encapsulation thickness is 0.5251×10^{-3} which implies 28% and 63% improvement compared with WNN and FFNN. VTWNN gives a better performance because the translation parameters of the wavelet neural network are variable. With this property, VTWNN has the ability to model the input-output function with input-dependent network parameters. It works as if several individual networks are handling different sets of input data, which exhibits a better performance. In order to test the generalization ability of the proposed network, a 10-fold cross-validation process is given. The results in terms of the mean testing error, standard deviation are tabulated in Table II. In this table, the proposed VTWMM gives a better testing result compared with others.

V. Conclusion

In this paper, a hybrid PSO has been proposed to optimize the variable translation wavelet neural network VTWNN. Thanks to the variable translation parameters in the network, the proposed VTWNN becomes adaptive and is able to improve the learning ability of the neural networks. An industrial application on modelling the fluid dispensing for electronic packaging using the PSO based-wavelet neural network has been discussed. Experimental results have been given to show the improvement.

Acknowledgments

The work described in this paper was substantially supported by grants from the University of Western Australia and from the Hong Kong Polytechnic University (Project code G-U414).

References

[1] S.H. Ling, C.W. Yeung, K.Y. Chan, H.H.C. Iu, and F.H.F. Leung, "A new hybrid particle swarm optimization with wavelet mutation for neural network training," in *Proc. Congress on Evolutionary Computation (CEC2007), Singapore*, Sep 2007, pp. 1977-1984.

[2] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Information Theory*, vol. 36, no.5, pp. 961-1005, Sep. 1990.

[3] S. Yao, C.J. Wei, and Z.Y. He, "Evolving wavelet neural networks for function approximation," *Electron. Lett.*, vol.32, no.4, pp. 360-361, Feb. 1996.

[4] S.J. Yoo, J.B. Park, and Y.H. Choi, "Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks," *IEEE Trans. on Systems, Man and Cybernetics: Part B: Cybernetics*, vol. 36, no. 6, pp. 1342-1355, Dec. 2006.

[5] Y.C. Huang and C.M. Huang, "Evolving wavelet networks for power transformer condition monitoring," *IEEE Trans. Power Delivery*, vol. 17, no. 2, pp. 412-416, Apr. 2002.

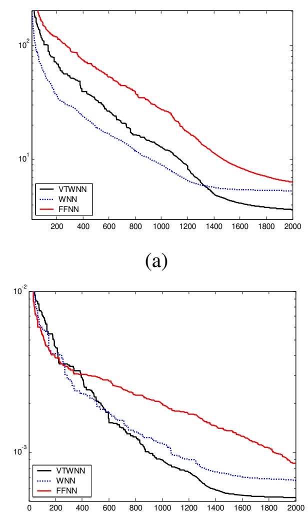
[6] B. Widrow and M.A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442, Sept. 1990.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, vol. 4, 1995, pp.1942-1948.

[8] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Trans. Industrial Electronics*, vol.52, no. 6, pp.1478-1498, Dec. 2005.

[9] Y. Song, Z, Chen, and Z. Yuan, "New chaotic PSO-based neural network predictive control for nonlinear process," *IEEE Trans. Neural Networks*, vol. 18, no. 2, pp.595-601, Mar. 2007.

[10] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence* vol.12, no. 2, 1995. pp. 1137–1143.



(a)

(b)

Fig. 7 Comparisons between different neural network for MFD-EP (a)encapsulation weight, (b)encapsulation thickness.

TABLE I Comparison between different neural network topologies for MFD-EP (Training): (A) encapsulation weight, (B) encapsulation thickness.

(A)

	VTWNN	WNN	FFNN
Mean	3.6492	5.3277	6.3729
Best	2.6033	4.4550	5.1207
S.D.	0.1857	0.1671	0.5987
Run Time	154.44	145.78	101.76

(B)

	VTWNN	WNN	FFNN
Mean($\times10^{-3}$)	0.5251	0.6728	0.8542
Best ($\times10^{-3}$)	0.4420	0.5012	0.6016
S.D. ($\times10^{-3}$)	0.0322	0.0238	0.0883
Run Time	167.33	150.88	112.23

TABLE II Comparison between different neural network topologies for MFD-EP (Testing): (A) encapsulation weight, (B) encapsulation thickness.

(A)

	VTWNN	WNN	FFNN
Mean	6.41	7.76	11.83
S.D.	0.2711	0.4920	1.5832

(B)

	VTWNN	WNN	FFNN
Mean($\times10^{-3}$)	0.5280	0.6892	0.8901
Best ($\times10^{-3}$)	0.0454	0.0487	0.1678

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 6, 2009 at 23:21 from IEEE Xplore. Restrictions apply.