The 5th International Conference on Electrical Engineering – Boumerdes (ICEE-B)
October 29-31, 2017, Boumerdes, Algeria.

Improved Pi-Sigma Neural Network for nonlinear system identification

LADJOUZI Samir
Department of Electrical Engineering University of Bouira
Bouira, Algeria
ladjouzi_samir2001@yahoo.fr

GROUNI Said
Department of Industrial Maintenance University of Boumerdes Boumerdes, Algeria
said.grouni@yahoo.com

KACIMI Nora
Process Control Laboratory
National Polytechnic School, Bp 182
El Harrach, Algiers, Algeria
kacimi.nora@yahoo.fr

SOUFI Youcef
Department of Electrical Engineering
University of Tébessa
Tébessa, Algeria
y_soufi@yahoo.fr

Abstract— **In this paper, we propose a modified architecture of a Pi-Sigma Neural Network (PSNN) based on two modifications: extension of the activation function and adding delays to neurons in the hidden layer. These new networks are called respectively Activation Function Extended Pi-Sigma (AFEPS) and Delayed Pi-Sigma (DPS) are obtained first by adding an activation function to all hidden neurons and secondly by modifying the PSNN so its hidden layer outputs are fed to temporal adjustable units that permit to this new network to be capable to identify nonlinear systems. Architecture and dynamic equations of these networks are given in details with their training algorithm. To ensure the effectiveness of our proposed networks, examples of nonlinear system identification are provided. The obtained results show the capacity of HONNs for the nonlinear systems identification. In particular, the proposed neural architectures (AFEPS and DPS) provide better results due to the modifications made on them.**

Keywords: *Pi-Sigma Neural Network, Extended Activation Function Pi-Sigma, Delayed Pi-Sigma, temporal adjustable units , nonlinear systems identification.*

I. Introduction

A class of Artificial Neural Networks (ANN) called High Order Neural Networks (HONNs) appears around 1990 to overcome limitation of traditional topologies of ANN like the Multi Layer Perceptron [1]. The majority of neural networks can be classified as linear networks due to the linearity input neurons combination, in HONNs, higher-order combinations of neurons give a superior dynamic to the network and perform nonlinear mapping efficiently owed of the inputs nonlinearly [2].

HONNs have been the focus of significant researches and have generally the advantage to have superior generalization, but they are more expensive in terms of computations and their training are sometimes more complicit [3].

HONNs are used in several applications like: prediction of non-linear time-series [4], pattern recognition [5], financial time series prediction and misclassification cost for different financial distress prediction models [6, 7], a nonlinear classification based on Chemical Reaction Optimization HONN [8].

Among most popular HONNs we interest to the Pi-Sigma Neural Network. This network is an idea developed by Shin and Ghosh in 1992. Rather, adding the input's sum of product like in other networks, PSNN utilizes as output the product of processing units in the hidden layer which are obtained by a linear summation. Compared to other types of HONNs, PSNN do not require a very important number of weights, but a fewer is sufficient [9].

Many works have been proposed to achieve performances of the PSNN: authors in [10] have proposed a recurrent Pi–Sigma neural network used as a predictor structure in Differential Pulse Code Modulation systems. A sigma-pi network trained with an online learning algorithm have been presented in [11] for solving the frame of reference transformation problem. A chemical reaction optimization based Pi–Sigma neural network was proposed in [12] by keeping the population size fixed for solving parity problems.

In this paper, we suggest two different architectures of the PSNN based on two modifications. Firstly, we propose to extend the activation function to all hidden neurons, and secondly by modifying the PSNN so its hidden layer outputs are fed to temporal adjustable units that permit to this new network to be capable to identify nonlinear systems. These new networks are called respectively Activation Function Extended Pi-Sigma (AFEPS) and Delayed Pi-Sigma (DPS). To show the effectiveness of our networks some examples based on the identification of dynamical systems are provided.

The remaining part of the paper is structured as follows: section 2 illustrates architecture and equations of the PSNN. Section 3 describes the new modifications of the PSNN. Nonlinear systems identification examples are presented in section 4 and finally section 5 concludes the work.

II. Pi-Sigma Neural Network

The architecture of the PSNN is composed of three layers, an input, single hidden and output layer. The input layer is formed by external inputs which are weighted and their summation is sent to the single hidden layer. Then outputs of this layer are multiplied and passed to the output layer by a nonlinear function (see Fig. 1).

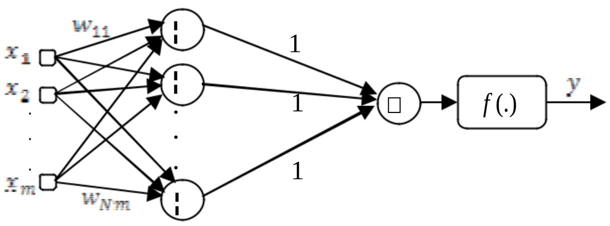

Figure 1. Pi-Sigma Neural Network architecture

Only weights which are connected between the input and hidden layers are adjusted, while those linked the hidden neurons to the output layer are fixed to one and they are not modified.

We use the following notations relative to our work:

$x_j(k)$: the j^{th} input of the network at time *k*.

$s_i(k)$: the i^{th} input in the hidden layer at time *k*.

$v_i(k)$: the i^{th} delayed unit output associated to the i^{th} hidden neuron at time *k*.

$h_i(k)$: the i^{th} output in the hidden layer at time *k*.

$w_{ij}(k)$: the weight of the connection from j^{th} input to i^{th} hidden neuron at time *k*.

$b_i(k)$: the i^{th} bias in the hidden layer at time *k*.

f : the activation function of hidden neurons.

N: the hidden neurons number.

y(k) : the output of the network at time *k*.

$(1 \le i \le N)$

Let us consider a Pi-Sigma network with *m* external inputs, equations of the Pi-Sigma are given by:

$$h_i(k) = \sum_{j=1}^{N} w_{ji}(k)\, x_j(k) + b_i(k) \qquad (1)$$

$$prodh(k) = \prod_{i=1}^{N} h_i(k) \qquad (2)$$

$$y(k) = f\big(prodh(k)\big) \qquad (3)$$

Let consider the following criterion to be minimized:

$$E = \frac{1}{2}\sum_{k=1}^{T}\big(y(k) - y_d(k)\big)^2 \qquad (4)$$

When $y_d(k)$ is the desired output and T is the length of the training sequence.

Equations for updating the weights are given below:

$$w_{ji}(k+1) = w_{ji}(k) - \mu \frac{\partial E}{\partial w_{ji}(k)} \qquad (5)$$

$$b_i(k+1) = b_i(k) - \mu \frac{\partial E}{\partial b_i(k)} \qquad (6)$$

When µ is the step-size.

The learning algorithm to train the Pi-Sigma can be obtained by the chain rule:

$$\frac{\partial E}{\partial w_{ji}(k)} = \frac{\partial E}{\partial y(k)} \frac{\partial y(k)}{\partial prodh(k)} \frac{\partial prodh(k)}{\partial h_i(k)} \frac{\partial h_i(k)}{\partial w_{ji}(k)} \qquad (7)$$

$$\frac{\partial E}{\partial b_i(k)} = \frac{\partial E}{\partial y(k)} \frac{\partial y(k)}{\partial prodh(k)} \frac{\partial prodh(k)}{\partial h_i(k)} \frac{\partial h_i(k)}{\partial b_i(k)} \qquad (8)$$

After derivation, we obtain:

$$\frac{\partial E}{\partial w_{ji}(k)} = \big(y(k) - y_d(k)\big)\acute{f}\big(prodh(k)\big) \prod_{\substack{p=1 \\ p\neq i}}^{N} h_p(k) x_i(k) \qquad (9)$$

$$\frac{\partial E}{\partial b_i(k)} = \big(y(k) - y_d(k)\big)\acute{f}\big(prodh(k)\big) \prod_{\substack{p=1 \\ p\neq i}}^{N} h_p(k) \qquad (10)$$

III. The Proposed Modifications in the PSNN

A. ***First Modification:*** *Activation Function Extension*

Like in the Multi Layer Perceptron we propose for the first modification on the Pi-Sigma architecture to apply the activation function for all hidden neurons and the network output does not have an activation function (see Fig. 2). We can name this first network by Activation Function Extension Pi-Sigma (AFEPS). According to this transformation equations of the AFEPS are given by:

$$s_i(k)) = \sum_{j=1}^{m} w_{ji}(k)\, x_j(k) + b_i(k) \qquad (11)$$

$$h_i(k) = f\big(s_i(k)\big) \qquad (12)$$

$$y(k) = \prod_{i=1}^{N} h_i(k) \qquad (13)$$

The learning algorithm to train the AFEPS can be obtained by the chain rule:

$$\frac{\partial E}{\partial w_{ji}(k)} = \big(y(k) - y_d(k)\big) \prod_{\substack{p=1 \\ p\neq i}}^{N} h_p(k)\, f'\big(s_i(k)\big) x_i(k) \qquad (14)$$

$$\frac{\partial E}{\partial b_i(k)} = \big(y(k) - y_d(k)\big) \prod_{\substack{p=1 \\ p\neq i}}^{N} h_p(k)\, f'\big(s_i(k)\big) \qquad (15)$$

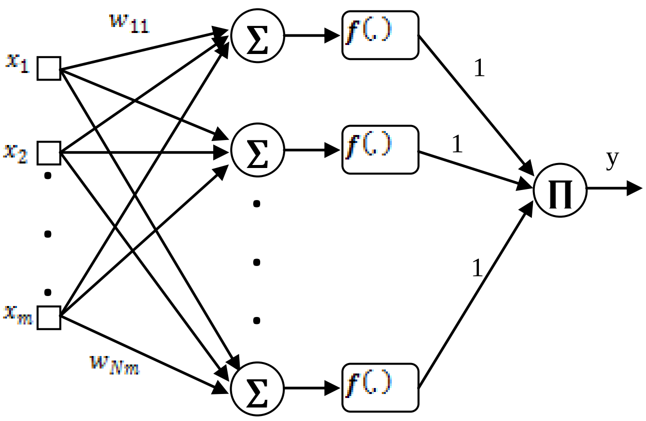

Figure 2. Activation Function Extension Pi-Sigma architecture

B. ***Second Modification:*** *Adding Delayed Units to Hidden Neurons*

As in [13], we add for each hidden neuron a delayed unit with a corresponding weight ${}_i(k)$ between hidden neuron and its unit and (1- $\alpha_i(k)$) between the input unit and his output. The network output is computed by the summation of both outputs hidden neuron product and output unit delay product (see Fig. 3). We call this new architecture the Delayed Pi-Sigma.

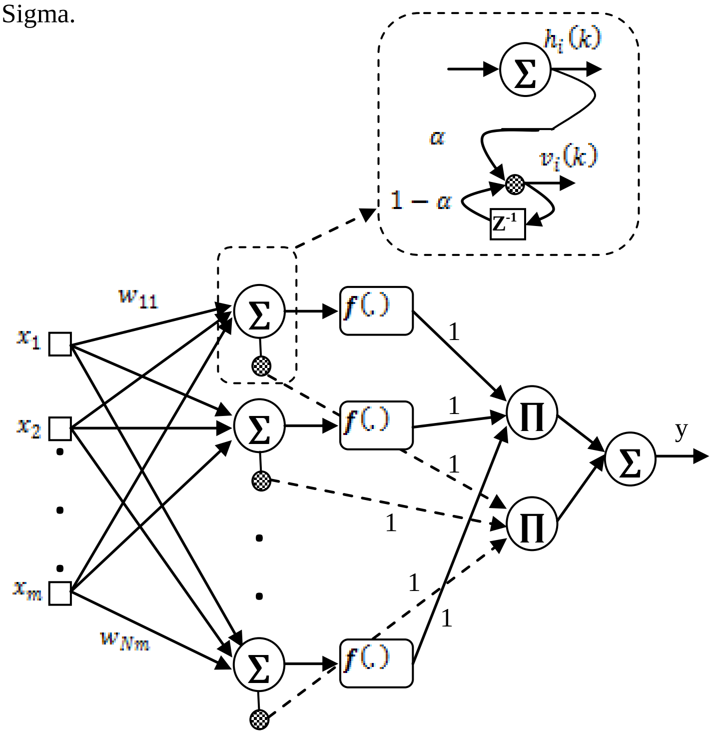

Figure 3. Delayed Pi-Sigma architecture

The same equations (11) and (12) are used here, we add a new equation given by:

$$v_i(k) = \alpha_i(k) h_i(k) + \big(1 - \alpha_i(k)\big) v_i(k-1) \qquad (16)$$

And the output of the network is computed with this next equation:

$$y(k) = \prod_{i=1}^{N} h_i(k) + \prod_{i=1}^{N} v_i(k) \qquad (17)$$

Updating weights $w_{ij}(k)$ and $b_i(k)$ is done by these equations [14]:

$$\frac{\partial E}{\partial \alpha_i(k)} = \frac{\partial E}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial v_i(k)} \frac{\partial v_i(k)}{\partial \alpha_i(k)} \qquad (18)$$

And after derivation, we obtain:

$$\frac{\partial E}{\partial \alpha_i(k)} = \big(y(k) - y_d(k)\big) \prod_{\substack{p=1 \\ p\neq i}}^{N} v_p(k) \big(h_i(k) - v_i(k-1)\big) \qquad (19)$$

IV. SIMULATIONS AND RESULTS

To demonstrate the efficiency of our network architectures, we have tested them on two nonlinear dynamic systems by using the identification scheme illustrated in Fig. 3.

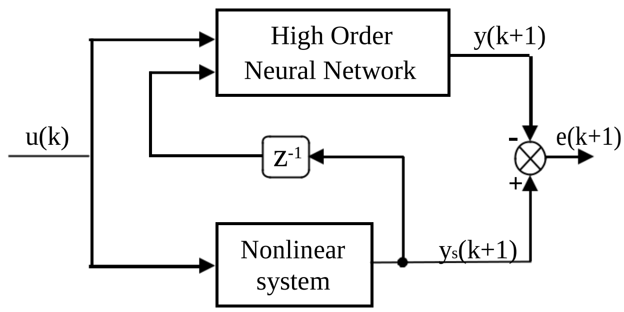

Figure 4. Serie-parallel identification scheme with HONN

For all simulations, we have taken five neurons in the hidden layer, the activation function used is the sigmoid bipolar and the value of the step-size is equal to 0.1(μ =0.1)

System 1: the first example of nonlinear system identification is defined by:

$$y_s(k+1) = \frac{y_s(k)\left[y_s(k-1)+2\right]\left[y_s(k)+2.5\right]}{8.5 + \left[y_s(k)\right]^2 + \left[y_s(k-1)\right]^2} + u(k) \qquad (20)$$

The phase training of the HONN is made during 1000 epochs. During this period the system input u(k) is taking randomly in the interval [-1, 1]; then for 500 epochs the network is tested with the following input:

$$u(k) = \begin{cases} 2\cos\left(\frac{2\pi k}{100}\right) & si\ k \leq 200 \\ 1.2\sin\left(\frac{2\pi k}{20}\right) & si\ 200 < k \leq 500 \end{cases} \qquad (21)$$

Figures (5), (6) and (7) illustrate the plant output with respectively the three HONNs: PSNN, AFEPS and the DPS.

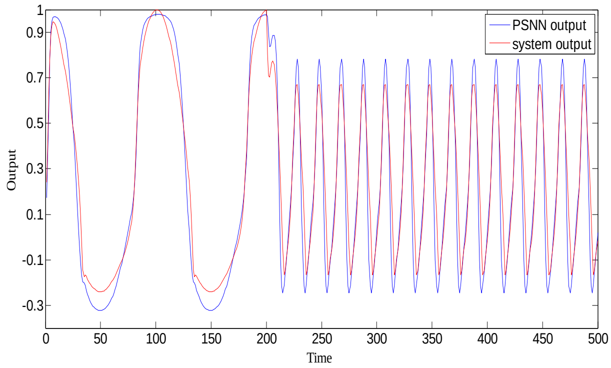

Figure 5. Outputs of the system 1 and the PSNN

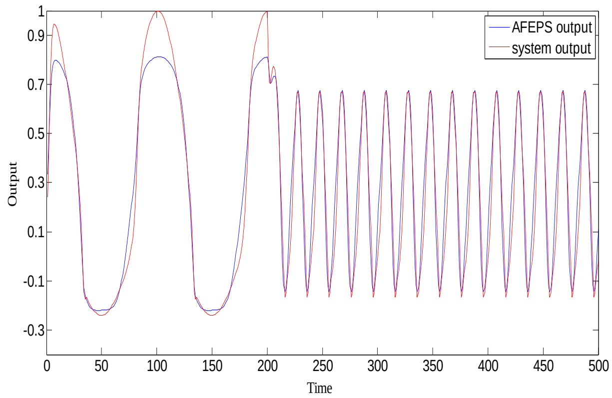

Figure 6. Outputs of the system 1 and the AFEPS

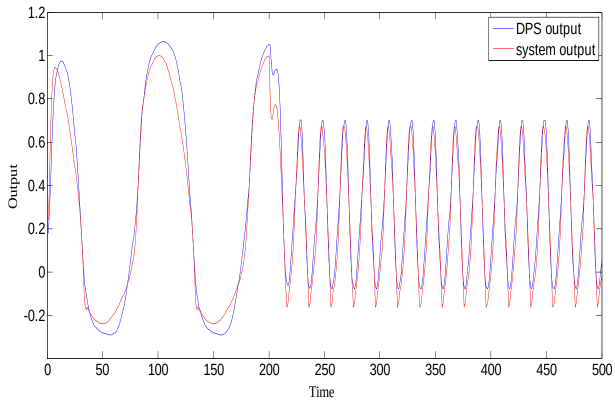

Figure 7. Outputs of the system 1 and the DPS

System 2: the second dynamic system to be identified is given by the following set of equations:

$$y_p(k+1) = f\left(x_p(k)\right) \qquad (22)$$

$$f(v) = \left(\frac{v}{3.5}\right)^3 \qquad (23)$$

$$x_p(k) = W(z)h\left(u(k)\right) \qquad (24)$$

$$W(z) = \frac{z+0.3}{z^2 - 0.8z + 0.15} \qquad (25)$$

$$h(v) = \frac{4v^3}{1+4v^2} \qquad (26)$$

As for the first example, we take here 2000 epochs for the learning step and 1000 epochs when the network is tested with the following input:

$$u(k) = \begin{cases} sin\left(\frac{\pi k}{25}\right) & if\ \ k \leq 250 \\ 1 & if\ 250 < k \leq 500 \\ -1 & if\ 500 < k \leq 750 \\ 0.3sin\left(\frac{\pi k}{25}\right) + 0.1sin\left(\frac{\pi k}{32}\right) + & \\ 0.6sin\left(\frac{\pi k}{10}\right) & if\ 750 < k \leq 1000 \end{cases} \qquad (27)$$

Figures (8), (9) and (10) show the obtained results for the second example with the three HONNs.

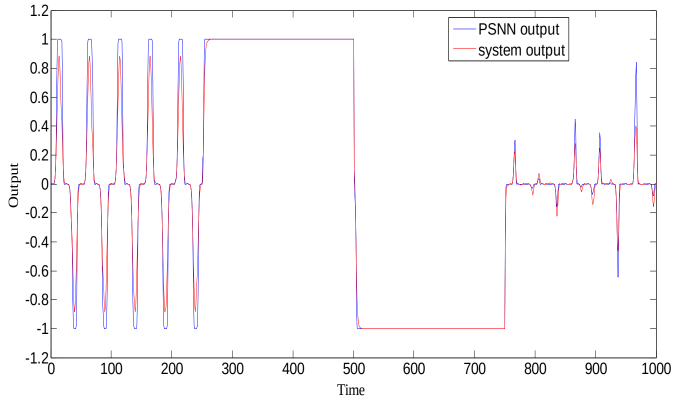

Figure 8. Outputs of the system 2 and the PSNN

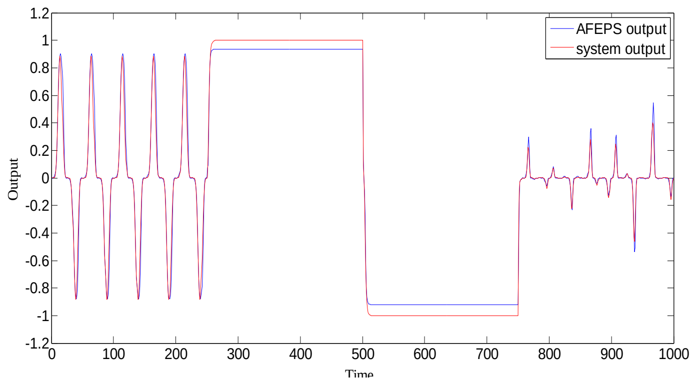

Figure 9. Outputs of the system 2 and the AFEPS

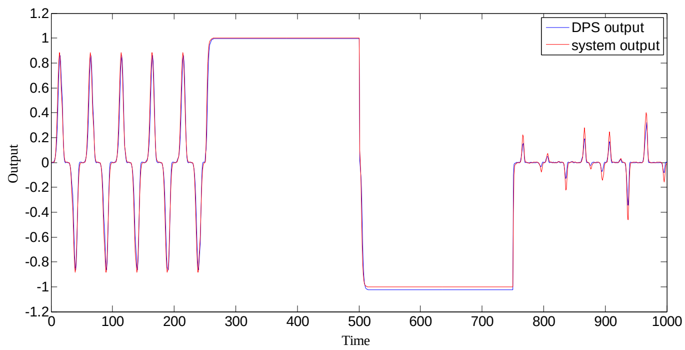

Figure 10. Outputs of the system 2 and the DPS

In summary, we can resume our results in the following table by using the Mean Squared Error (MSE) of the different errors between the system output and HONN output:

TABLE I. MSE FOR THE TWO EXAMPLES

	PSNN	AFEPS	DPS
SYSTEM 1	0.0133	0.0079	0.0066
SYSTEM 2	0.0096	0.0049	0.0018

From the results showed in the different figures and the MSE values, we can see that all HONNs are capable to identify the nonlinear systems. Especially for the proposed ones (AFEPS and DPS) the identification process provides better results compared to the initial PSNN.

The advantage of adding activation function for each neuron in the hidden layer makes the dynamic of the neuron near of the behavior of a real biological neuron. Also, it gives for the network to have the non linearity ability for the task to doing.

The presence of the delayed unit for each hidden neuron allows to the network to achieve more dynamic and hence to be able to approximate any non linear system.

V. CONCLUSION

In this article we have discussed on two novel architectures of HONNs inspired from the PSNN with some modifications. The first modification is about the extension of the activation function to all hidden neurons instead to place it in the output layer. The idea of the second modification is to add for each hidden neuron a delayed unit with a corresponding weight between hidden neuron and its unit and between the input unit and his output. The obtained results show that these new networks are suitable for nonlinear system identification.

We hope that this work will be extended to other types of HONNs, and applied to nonlinear system control.

REFERENCES

[1] H. C. Tsai, "Modeling concrete strength with higher-order neural networks," Neural Computing and Applications vol. 27, pp. 2465–2473, 2016.

[2] I. El-Nabarawy, A.M. Abdelbar, D.C. Wunsch,"Levenberg-Marquardt and Conjugate Gradient methods applied to a high-order neural network,'' International Joint Conference on Neural Networks (IJCNN), pp. 1-7, 2013.

[3] M. A. Ashraf, M. S.Khalid,"Ant Colony Optimization Applied to the Regularization Process of a High Order Neural Network,'' IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2016.

[4] H. Tawfik, P. Liatsis," Prediction of non-linear time series using higher order neural networks,'' In Proceeding IWSSIP1997 conference, Poznan, Poland, 1997.

[5] T. Kaita, S. Tomita, J. Yamanaka," On a higher order neural network for distortion invariant pattern recognition,'' Pattern Recognit Letters, vol. 23, pp. 977–984, 2002.

[6] R.Ghazali, D.al-Jumeily,"Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction,'' In Artificial Higher Order Neural Networks for Economics and Business, pp. 271-293, 2009.

[7] J.Fulcher, M.Zhang, S.Xu,"The Application of Higher-Order Neural Networks to Financial Time Series,'' In J. Kamruzzaman (Ed), Artificial Neural Networks in Finance, Health and manufacturing : Potential and Challenges, pp. 80-108, 2006.

[8] J.Nayak, B.Naik, H.S.Behera," A novel chemical reaction optimization based higher order neural network (CRO-HONN) for nonlinear classification,'' Ain Shams Engineering Journal, Vol. 6, No. 3, pp.1069–1091, 2015.

[9] J. Ghosh and Y. Shin,"Efficient higher-order neural networks for classification and function approximation," International Journal of Neural Systems, vol. 3,no. 4, pp. 323-350, 1992.

[10] A.J. Hussain, P.Liatsis ,"Recurrent Pi–Sigma networks for DPCM image coding,'' Neurocomputing, vol.55, pp. 363–82, 2002.

[11] C.Weber, S.Wermter,"A self-organizing map of Sigma-Pi units,'' Neurocomputing, vol. 70, pp. 2552-2560, 2007.

[12] K.K. Sahu, S. Panigrahi, H.S. Behera,"A novel chemical reaction optimization algorithm for higher order neural network training,'' Journal of Information Technology Theory and Application, vol. 53, pp. 402-409, 2013.

[13] P. S. Sastry, K. P. Unnikrishnan,"Memory Neuron Networks for Identification and Control of Dynamical systems,'' IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 306-319, 1994.

[14] B.Subudhi, D.Jena,"A differential evolution based neural network approach to nonlinear system identification,'' Applied Soft Computing, vol. 11, No. 1, pp. 861-871, 2011.