A Modified Elman Network with Memory Units for 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

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

***Abstract*— In this paper we propose a modified Elman network structure called memory Elman neural network. The idea of this new architecture is based on adding memory units to the neurons of the classic Elman network. These memory units are trainable temporal elements that make the output history-sensitive. By virtue of this capacity, this new architecture can take into account the past information of the neurons and use them in order to accomplish the task of the network. In order to show the performance of this new network, some dynamical systems are used for identification and results are compared with the conventional Elman network.**

Keywords— Elman network structure; memory units; dynamical systems; identification

I. Introduction

Recurrent neural networks are a special type of artificial neural networks where the associations between neurons are formed by a closed cycle. This configuration allows showing dynamic temporal behavior of the network. Among these recurrent architectures, we cite the Elman Neural Network (ENN) which consists of three layers: an input, hidden and output. In addition to these three layers, the Elman network has an additional layer called context layer consisting of the delayed hidden neurons output. The context neurons allow to the Elman network to have certain dynamical advantages over static neural network; and through the different papers the ENN is commonly applied in dynamical systems identification and control [1]. The Elman network has been used in various applications like in face and speech recognition [2, 3], system identification [4], wind power [5].

Many modifications on the Elman network architecture have been proposed in literature to improve its performance, in [6] authors propose adding new adaptable weights between the context neurons and output neurons. A new embedded memory unit has been added to the Elman structure in [7]. An hybrid model based on Elman Neural Network and differential evolution algorithm was performed in [8] and in [9] authors suggest to enhance the performance of the Elman network by modifying the learning rate.

In this paper, we propose a modified Elman architecture by adding temporal adjustable units to the hidden neurons. This new network is called Memory Elman Neural Network (MENN). The performance of our network is tested on some examples based on the identification of dynamical systems.

The rest of paper is ordered as follows: section 2 illustrates architecture, equations and training procedure of the MENN. Nonlinear systems identification examples are presented in section 3 and finally section 4 concludes the work.

II. Memory Elman Neural Network

The Elman neural network was first proposed by Jeffrey Elman, an American psycholinguist and professor of cognitive science at the university of California, in 1990 [10].

The Elman network can be seen as a special feedforward neural network with a single hidden layer in which was added an additional layer called the context layer representing the delayed outputs of hidden neurons as shown in Figure 1.

We note that the black circles do not symbolize real neurons but they are only an illustration of the context layer. Thus, the context neurons represent virtual nodes where the output and the input are the same.

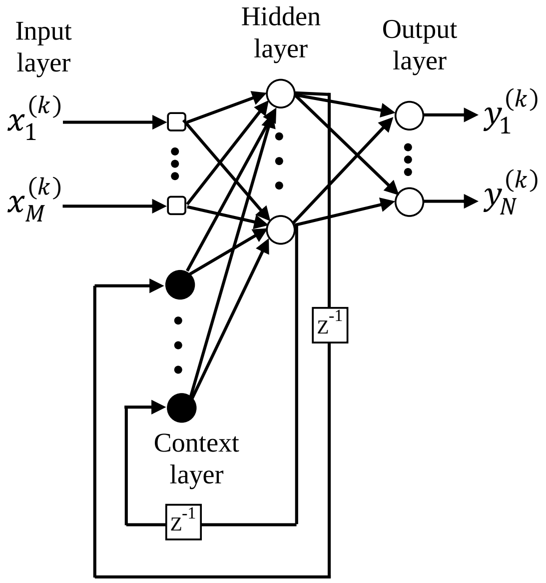

Fig. 1. Architecture **of** the Elman neural network

We use the following notation:

w_{ij}^1: The weight connection between the neuron ***i*** in the input layer and the neuron ***j*** in the hidden layer

w_{ij}^2: The weight connection between the neuron ***i*** in the hidden layer and the neuron ***j*** in the output layer

w_{ij}^3: The weight connection between the neuron ***i*** in the context layer and the neuron ***j*** in the hidden layer

h_i: The input of neuron ***i*** in the hidden layer

o_i: The output of neuron ***i*** in the hidden layer

c_i: The input of neuron ***i*** in the context layer

s_j: The input of neuron ***j*** in the output layer

f_1 (x): The activation function in the hidden layer

f_2 (x): The activation function in the output layer

M, N, L: The number of: inputs, outputs and hidden neurons respectively

$x_i^{(k)}$, $y_j^{(k)}$: Inputs and outputs of the Elman neural network

where i = 1,2, …M and j = 1,2, …N.

z^{-1}: unit delay

Equations of the Elman network are given below:

$$h_i^{(k)} = \sum_{i=1}^{M} w_{ij}^1 x_i^{(k)} + \sum_{i=1}^{L} w_{ij}^3 c_i^{(k)} \tag{1}$$

$$o_i^{(k)} = f_1\left(h_i^{(k)}\right) \tag{2}$$

$$s_j^{(k)} = \sum_{i=1}^{L} w_{ij}^2 o_i^{(k)} \tag{3}$$

$$y_j^{(k)} = f_2\left(s_j^{(k)}\right) \tag{4}$$

Let consider the following criterion to be minimized:

$$E = \frac{1}{2}\sum_{k=1}^{T}\sum_{i=1}^{N}\left(y^{(k)} - y_d^{(k)}\right) \tag{5}$$

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

The general equation for updating the weights is given below:

$$w_{ij}^p(k+1) = w_{ij}^p(k) - \mu \frac{\partial E}{\partial w_{ij}^p(k)} \tag{6}$$

When μ is the step size and p = 1, 2, 3.

The partial derivative of the global error with respect to the weights can be obtained by the backpropagation algorithm:

$$\frac{\partial E}{\partial w_{ij}^1(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial o_i^{(k)}} \frac{\partial o_i^{(k)}}{\partial h_i^{(k)}} \frac{\partial h_i^{(k)}}{\partial w_{ij}^1(k)} \tag{7}$$

$$\frac{\partial E}{\partial w_{ij}^2(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial w_{ij}^2(k)} \tag{8}$$

$$\frac{\partial E}{\partial w_{ij}^3(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial o_i^{(k)}} \frac{\partial o_i^{(k)}}{\partial h_i^{(k)}} \frac{\partial h_i^{(k)}}{\partial w_{ij}^3(k)} \tag{9}$$

After derivation, we obtain:

$$\frac{\partial E}{\partial w_{ij}^1(k)} = \left(y^{(k)} - y_d^{(k)}\right) f_2'\left(s_j^{(k)}\right) w_{ij}^2(k) f_1'\left(h_i^{(k)}\right) x_i^{(k)} \tag{10}$$

$$\frac{\partial E}{\partial w_{ij}^2(k)} = \left(y^{(k)} - y_d^{(k)}\right) f_2'\left(s_j^{(k)}\right) o_i^{(k)} \tag{11}$$

$$\frac{\partial E}{\partial w_{ij}^3(k)} = \left(y^{(k)} - y_d^{(k)}\right) f_2'\left(s_j^{(k)}\right) w_{ij}^2(k) f_1'\left(h_i^{(k)}\right) c_i^{(k)} \tag{12}$$

Authors in [11] have suggested to add memory units for each neuron in a multi layer perceptron, in our paper, we propose the same modification in the ENN by adding for each neuron a memory unit with a corresponding weight $\alpha_i(k)$ between the output of the neuron and its memory unit and the weight (1-α_i(k)) between the input unit and its delayed output. Another input is added to the neurons in the output layer produced by the delayed output of the memory unit weighted by the value β_i(k)

The Memory Elman Neural Network architecture with the detailed weighted links is depicted in Figure 2.

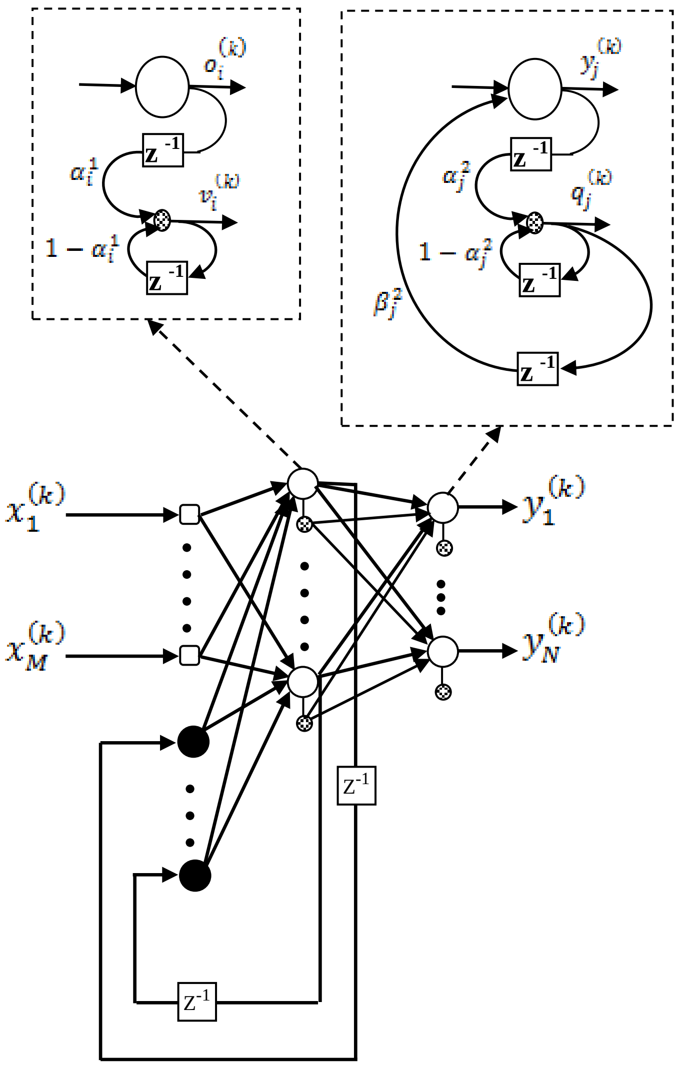

Fig. 2. Architecture **of** the Memory Elman Neural Network

We use the new following notation:

$v_i^{(k)}$: the output of the i^{th} delayed memory unit associated to the i^{th} neuron in the hidden layer.

$q_j^{(k)}$: the output of the j^{th} delayed memory unit associated to the j^{th} neuron in the output layer .

α_i^1: The weight connection between the delayed output of the neuron ***i*** in the hidden layer and the input of its memory unit.
α_j^2: The weight connection between the delayed output of the neuron ***j*** in the output layer and the input of its memory unit.
β_j^2: The weight connection between the input of the neuron ***j*** in the output layer and the delayed output of its memory unit.
F_{ij}^1: The weight connection between the memory unit of the neuron ***i*** in the hidden layer and the neuron ***j*** in the output layer

The equations (1), (2) and (4) are the same for the MENN, and equation (3) is rewritten as:

$$s_j^{(k)} = \sum_{i=1}^{L} w_{ij}^2 o_i^{(k)} + \sum_{i=1}^{L} F_{ij}^1 v_i^{(k)} + \beta_j^2 q_j^{(k-1)} \qquad (13)$$

The outputs of the memory unit in the hidden layer and output layer are given, respectively, by these equations:

$$v_i^{(k)} = \alpha_i^1 o_i^{(k-1)} + (1-\alpha_i^1) v_i^{(k-1)} \qquad (14)$$

$$q_j^{(k)} = \alpha_j^2 y_j^{(k-1)} + \left(1-\alpha_j^2\right) q_j^{(k-1)} \qquad (15)$$

The adjusted weights w_{ij}^1, w_{ij}^2 and w_{ij}^3 are done by equations (6), (10), (11) and (12).

For others weights, we follow the same training procedure. Thus, the updating weight is made by the following equations:

$$F_{ij}^1(k+1) = F_{ij}^1(k) - \mu \frac{\partial E}{\partial F_{ij}^1(k)} \qquad (16)$$

$$\alpha_i^1(k+1) = \alpha_i^1(k) - \mu' \frac{\partial E}{\partial \alpha_i^1(k)} \qquad (17)$$

$$\alpha_j^2(k+1) = \alpha_j^2(k) - \mu' \frac{\partial E}{\partial \alpha_j^2(k)} \qquad (18)$$

$$\beta_j^2(k+1) = \beta_j^2(k) - \mu' \frac{\partial E}{\partial \beta_j^2(k)} \qquad (19)$$

It may be noted that we have using another step size parameter μ' for the memory coefficients α_i^1, α_j^2 and β_j^2this choice is made to ensure that the change in the values of the memory coefficients will be slow compared to the other weights of the network.
After derivation of the above equations, we obtain:

$$\frac{\partial E}{\partial F_{ij}^1(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial F_{ij}^1(k)} \qquad (20)$$

$$\frac{\partial E}{\partial \alpha_i^1(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial v_i^{(k-1)}} \frac{\partial v_i^{(k-1)}}{\partial \alpha_i^1(k)} \qquad (21)$$

$$\frac{\partial E}{\partial \alpha_j^2(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial q_j^{(k-1)}} \frac{\partial q_j^{(k-1)}}{\partial \alpha_j^2(k)} \qquad (22)$$

$$\frac{\partial E}{\partial \beta_j^2(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial \beta_j^2(k)} \qquad (23)$$

III. SIMULATIONS AND RESULTS

To show the effectiveness of our network architecture, simulation results of three nonlinear systems identification are presented. Both the Elma network and the memory Elman network have two inputs u(k) and y(k) with one neuron in the output layer y(k+1).

We take four neurons in the hidden layer with the bipolar sigmoid function. For the output layer we choose the linear function. The values of the different step-size are chosen as μ= 0.1and μ'= 0.05.

System 1: the first example of nonlinear system identification is a second order system defined by [12]:

$$y(k+1) = \frac{y(k)y(k-1)[y(k)+2.5]}{1+[y(k)]^2+[y(k-1)]^2} + u(k) \qquad (24)$$

The phase training of the two networks ENN and MENN is made during 2000 epochs, then for 100 epochs the network is tested with the following input:

$$\boldsymbol{u(k) = sin\left(\frac{2\pi k}{25}\right)} \qquad (25)$$

Figures (3) and (4) show the identification performance for the system 1 using, respectively, the ENN and the MENN.

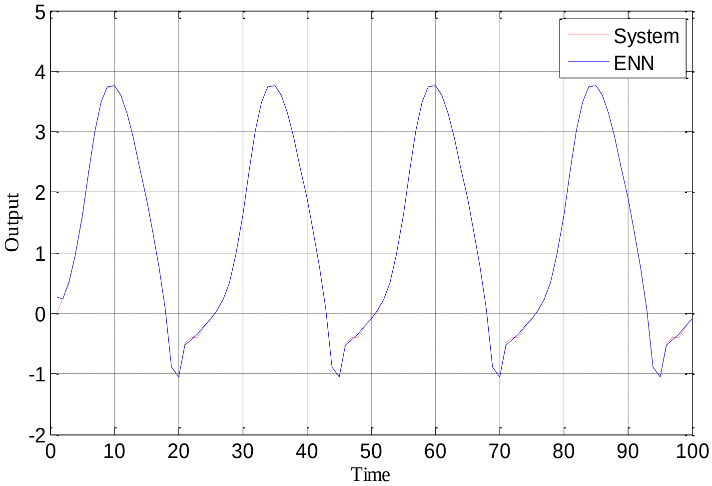

Fig. 3. Identification performance for the system 1 using the Elman network

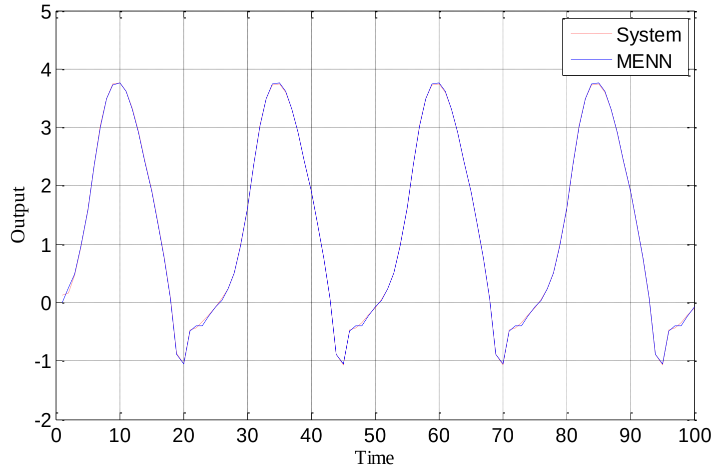

Fig. 4. Identification performance for the system 1 using the Memory Elman network

System 2: the second example is a second order nonlinear system described with the following discrete-time equation [13]:

$$y(k+1) = A_1 y(k) + A_2 y(k-1) + A_3 y^3(k-1) + B_1 u(k-1) \qquad (26)$$

Where: A_1 = 1.04, A_2 = -0.824, A_3 = 0.130667 and B_1 = -0.16

After training, we test our networks with the following sequence for 1000 epochs:

$$u(k) = 0.6sin\left(\frac{2\pi k}{250}\right) - 0.2cos\left(\frac{2\pi k}{100}\right) \qquad (27)$$

Figures (5) and (6) show the identification performance for the system 2 using, respectively, the ENN and the MENN.

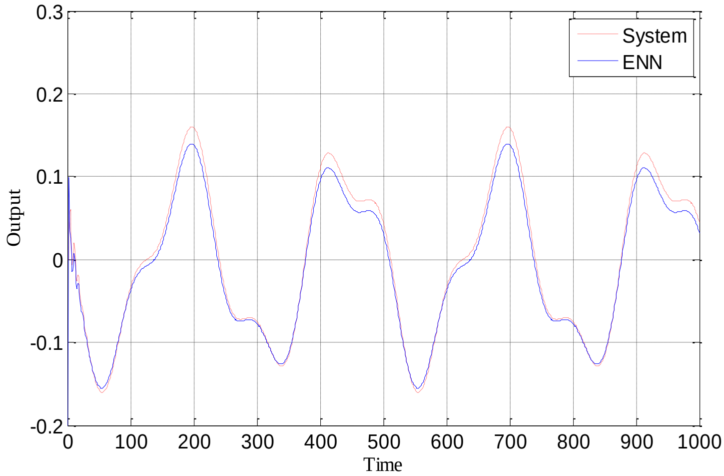

Fig. 5. Identification performance for the system 2 using the Elman network

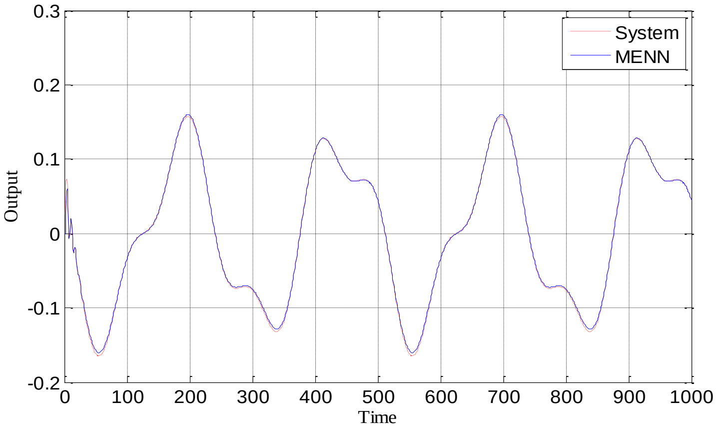

Fig. 6. Identification performance for the system 2 using the Memory Elman network

System 3: the third example is a third order nonlinear system described with the following relation [14]:

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)u(k-1)[y(k-2)-1]+u(k)}{1+[y(k-1)]^2+[y(k-2)]^2} \qquad (28)$$

The sequence test is given by:

$$u(k) = \begin{cases} sin\left(\frac{\pi k}{125}\right) & if\ k \leq 500 \\ 0.8 sin\left(\frac{\pi k}{125}\right) + 0.2 sin\left(\frac{\pi k}{12.5}\right) & if\ 500 < k \leq 1000 \end{cases} \qquad (29)$$

Figures (7) and (8) show the identification performance for the system 3 using, respectively, the ENN and the MENN.

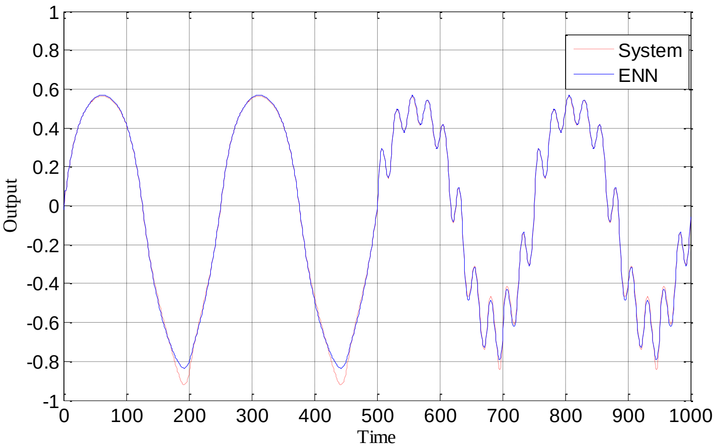

Fig. 7. Identification performance for the system 3 using the Elman network

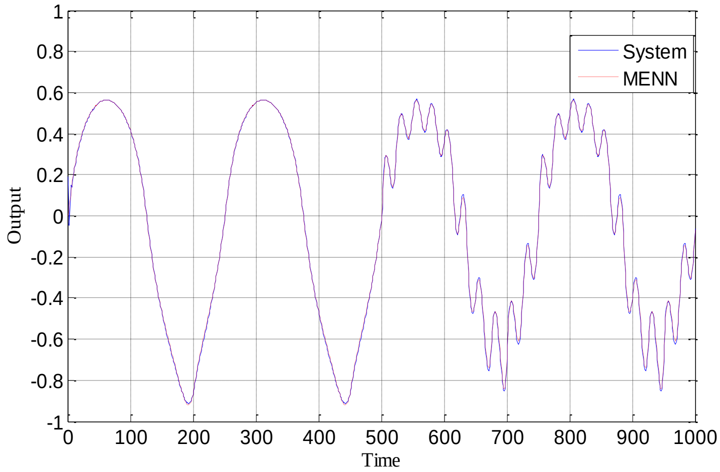

Fig. 8. Identification performance for the system 3 using the Memory Elman network

We can resume the graphical results by using the Mean Squared Error (MSE) defined by:

$$MSE = \frac{1}{T}\sum_{k=1}^{T}\left(y^{(k)} - y_d^{(k)}\right)^2 \qquad (30)$$

The table 1 show the different values of the MSE for the different systems obtained with the two networks ENN and MENN:

Table 1 MSE values for the three examples

	ENN	MENN
SYSTEM 1	**1.015*10^{-3}**	**5.233*10^{-4}**
SYSTEM 2	**2.217*10^{-4}**	**9.944*10^{-6}**
SYSTEM 3	**3.546*10^{-4}**	**9.596*10^{-5}**

The simulation results of the three nonlinear systems identification exhibit that the two recurrent neural networks Elman and Memory Elman can track the nonlinear systems dynamics with good approximations. Especially for the MENN, the results are better due to the memory unit added to the neurons.

IV. CONCLUSION

In this paper we have proposed a new modification on the Elman network architecture based on a transformation made on the different neurons of the ENN so for each neuron a memory unit has been added. These memory units have the capacity to take into account the past outputs of the neuron with its associated memory unit. The new memory network was used for identify nonlinear plants, and the obtained results show clearly the effectiveness of the MENN to the nonlinear system identification. Finally, we hope that this work will be extended to another types of recurrent networks, and applied to nonlinear system control.

References

[1] CH. Lin, “Recurrent modified Elman neural network control of PM synchronous generator system using wind turbine emulator of PM synchronous servo motor drive,” Int. J. Electr. Power Energy Syst., vol. 52, pp. 143–160, 2013.

[2] Saba Q. H. Al-Hashimy, “Face Recognition Using Elman Neural Network,” Int. J. Eng. and Innov. Tech. (IJEIT), vol. 7, pp. 38-42, 2017.

[3] A. Graves, A. Mohamed and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," Acoustics, Speech and Signal Processing (ICASSP), IEEE Int. Conf. pp. 6645–6649, 2013.

[4] I. K. Ibraheem, "System Identification of Thermal Process using Elman Neural Networks with No Prior Knowledge of System Dynamics," Int. J. Comp. App., vol. 161 (11), pp. 38-46, 2017.

[5] T.Xinran, Y.Jing, L.Teng and L. Jicheng, "Short-term Load Forecasting Study of Wind Power Based on Elman Neural Network," 7th Int. Conf. Elec. Infor. Eng., vol. 10322, 2017.

[6] X. Z. Gao, X. M. Gao, and S. J. Ovaska, “A modified Elman neural network model with application to dynamical systems identification,” Proc. IEEE Int. Conf. Sys., Man and Cyber, vol. 2, pp.1376-1381,1996.

[7] A. Kalinli, S. Sagiroglu, "Elman network with embedded memory for system identification," J. Inform. Sci. Eng., vol 22(6), pp.1555–1568, 2006.

[8] Z. Liouane, T. Lemlouma, P. Roose, F. Weis and H. Messaoud, " An improved elman neural network for daily living activities recognition,"Int. conf. int. sys. design app., pp. 697–707, 2017.

[9] G.H.Ren, Y.T.Cao, S.P. Wen, T.W. Huang and Z.G. Zeng,’’A modified Elman neural network with a new learning rate scheme,’’ Neurocomp., vol. 286,pp.11–18, 2018.

[10] J.L. Elman, “Finding structure in time,” Cogn. Sci., vol. 14, pp. 179-211, 1990.

[11] P. S. Sastry and K. P. Unnikrishnan, “Memory Neuron Networks for Identification and Control of Dynamical systems, ” IEEE Trans. Neur. Net., vol.5(2), pp. 306-319, 1994.

[12] J. Codina, J. C. Aguado, and J. M. Fuertes, “Neural network identification of non linear systems using state space techniques,” Int. Symp. Sig. Proces. Robot. and Neur. Net. (SPRANN’94), Lille, France, 1994.

[13] A. Kalinli, "Simulated annealing algorithm-based Elman network for dynamic system identification," Turk. J. Electr. Eng. Co., vol. 20, pp. 569-582, 2012.

[14] L. Qiang, Z.Yang and L. Huican,’’ A method of dynamic system identification based on memory RBF network,’’ Cont. Dec. Conf. (CCDC), pp. 3051-3054, 2016.