IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 9, 2020, accepted March 24, 2020, date of publication March 30, 2020, date of current version April 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984311

Modified Elman Spike Neural Network for Identification and Control of Dynamic System

NADIA ADNAN SHILTAGH AL-JAMALI[1,2], (Member, IEEE), AND HAMED S. AL-RAWESHIDY[2], (Senior Member, IEEE)
[1]Department of Computer Engineering, University of Baghdad, Baghdad 10071, Iraq
[2]Electrical and Computer Engineering Department, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, U.K.

Corresponding author: Nadia Adnan Shiltagh Al-Jamali (nadiaadnanshiltagh.aljamli@brunel.ac.uk)

ABSTRACT The utilization of conventional modeling strategies in the identification and control of a nonlinear dynamical system suffers from some weaknesses. These include absence of precise, conventional knowledge about the system, a high degree of uncertainty, strongly nonlinear and time-varying behavior. In this paper, a modified training algorithm for the identification and control of a nonlinear system using a soft-computing approach is proposed. Specifically, a modified structure of the Elman neural network with spike neural networks is proposed. This modified structure includes self-feedback, which provides a dynamic trace of the training algorithm. This self-feedback has weights, which can be trained during the training process. The simulation results show that the modified structure with the modified training algorithm is capable of the identification and control of a dynamic system in a more robust manor than when solely applying the other types of neural networks by 70% in terms of minimization of the percentage of error.

INDEX TERMS Identification, dynamic system, modified Elman spike neural network, spike neural network.

I. INTRODUCTION

Neural networks (NNs) for identification and control have been receiving increasing attention, because they can establish optimal identification and control signals due to the online training process [1]. A survey reported in [1] led to the proposal of an intelligent adaptive dynamic control system based on a recurrent wavelet Elman NN for an induction motor servo drive. The results showed that the proposed structure increases the uncertainty identifier and speeds up the convergence time, which means that both the convergence precision and convergence time are enhanced better than with the basic Elman NN. The most employed algorithm for NNs is the Backpropagation Neural Network (BPNN). It is popular, because of its power in studying difficult multidimensional mapping on non-linear systems, usually termed ''beyond regression''. Moreover, it has an easy structure design, so many researchers use this backpropagation to solve their problems [2]. Suprapto and Kusumoputro [2] proposed an algorithm based on an Elman recurrent NN for controlling the heavy-life hexacopter. The results were compared with BPNN and showed that the proposed algorithm has the capability of controlling the system with a smaller mean square error (MSE) value. Current typical NNs use a comparatively simple mathematical model of a neuron that is computationally effective, but not accurate in terms of biological settings and hence, more accurate biological neuron models are being developed. These models, like real neurons, use small portions of energy, called spikes, for communication with others. Hence, they are called spiking neural networks (SNNs) and from the taxonomy viewpoint, they belong to the third generation of NNs [3]. SNNs are strong computational modelling outfits that have attracted much attention, because of the effective bio-inspired modelling of synaptic interactions between neurons [4]. The basic Elman NN, which was introduced by [5] as one kind of partial recurrent NN model, is used for system identification and control. A combination of the Elman NN and SNN can give optimal results in these contexts. The use of artificial intelligence in the field of identification and prediction control, as in [6]–[9], has received increasing interest in recent years. Shou-Ping and Xue-Fei [10] proposed an architecture of an interval Elman NN used to model uncertain dynamic system, the simulation results showed that the proposed architecture

The associate editor coordinating the review of this manuscript and approving it for publication was Qi Zhou.

has better performance than the conventional interval feed forward BPNN. Sadek ***et al.*** [11] proposed an adaptive Elman to solve the mismatched uncertainty problem of under actuated robotic systems of a nonlinear system, with simulation results showing that the proposed control algorithm provides good performance. Deo and Chandra [12] undertook an empirical study on the minimal timespan required for robust prediction based on Elman recurrent NNs with two different training methods. Lin and Boldbaatar [13] developed a control system with a recurrent wavelet Elman NN, which enhanced the capabilities of a commercial aircraft to land automatically, with simulation results showing that the develop control system can achieve better performance than other control schemes. Dahmani ***et al.*** [14] proposed an approach involving online adjustment of the connection weights of an Elman NN controller for a greenhouse. A dynamic system, such as a stochastic jumping one is widely implemented in practice, which can arise among a finite number of system modes governed by a stochastic process. As a special class of stochastic jumping systems, semi-Markovian jump systems (SMJSs) have become a hot research topic in the past few years due to their general practical applications, such as communications, transportation systems, networks and dependability analysis [15]. Based on the above, a modified structure of an Elman NN with an SNN is introduced in this work, which is named as the Modified Elman Spike Neural Network (MESNN). The modified training algorithm is implemented to train the MESNN, with the proposed model being subsequently used for the identification and control of a dynamic system. The dynamic action of a spiking neuron is near to its biological equivalent. Signals from the adjacent presynaptic neurons are created by the dendrites of the postsynaptic neuron and are inherited to the soma. If the total irritation caused by the input is sufficient, i.e. above a threshold, an action potential, or spike, is emitted and propagated along the axon and its branches to other neurons. The axon-branches are placed at the end of the axon and connected to the dendrites of the other neurons, with this being called a synaptic connection. Each such connection is described by a threshold value. When a spike attains this threshold level, it causes a variation of membrane potential in the dendrites of the drawing neuron, called a postsynaptic neuron. The membrane potential is assigned as the postsynaptic potential, which can excite the neuron by lifting the potential or prevent the neuron by lowering it. The lifting of the postsynaptic potential causes the firing of the neuron for a certain period of time and if the synapse is fired, this is called excitation. The lowering of the potential leads to it being harder to fire the neuron and the synapse is called inhibitory. This process is comparatively slow, so the effect is delayed, with a specific ideal time for that synapse [16]. The strength of an SNN is obtained from precise modeling of the synaptic interactions between biological neurons, taking into account the time of spike firing. The computational power of SNNs outstrips that of classical neural networks that use threshold or sigmoidal activation functions. Furthermore, SNNs have the potential for quick adaptation [17]–[23]. Given these advantages, as above mentioned, an SNN with an Elman NN is considered in this paper for the identification and control of dynamic plants.

The main contributions of this paper can be summarized as follows:

1. A modified structure based on Elman NN and a spike neural network, named the Modified Elman Spike Neural Network (MESNN);
2. A modified training algorithm for MESNN based on updating its weights, delay and the threshold values;
3. The modified structure with the modified training algorithm is used for the identification and control of a dynamic system.

The remainder of this paper is organized as follows. Section II presents the proposed structure introduced in this paper and section III presents the system identification, whilst section IV explains the proposed controller. Section V introduces the modified training algorithm, with the performance of identification of the plant being explained in section VI and that of the controller in section VII. Finally, in section VIII the conclusion to the paper is provided.

II. THE PROPOSED STRUCTURE

Fig. 1 shows the proposed structure. The developments that have occurred in the science of artificial intelligence have prompted the use of these technologies in various aspects of life. Our focus is on the application of the proposed model in the field of systems identification and control. The proposed MESNN is a modified type of basic Elman NN, which was introduced by [5] as one of kind of partial recurrent spike NN model. The proposed structure consists of four layers: an input layer, a context layer, an invisible layer and an output layer. The modified structure has self-feedback with variable gain in the context layer, whilst the feedback from the invisible layer to the context layer has feedback weights, W^{hc}, which are adaptive during the training process. The spike criteria of the training algorithm speed up the training process, such that just the active nodes that reach the threshold value need to be updated. The broken line portion shown in Fig 2 (a, b) represents two neurons with a series of time delayed synaptic link.

The dynamics of the MESNN are explained in the following equations:

$$X(k) = f\,(W^{xc}X^{c}(k), W^{xu}U(k)) \tag{1}$$

$$X^{c}(k) = \alpha(k)X^{c}(k-1) + W^{hc}X(k-1) \tag{2}$$

$$Y^{m}(k+1) = W^{yx}X(k) \tag{3}$$

where, $Y^{m}(k)$ and $U(k)$ represent the output and input of MESNN, sequentially. $X^{c}(k)$, *and* X(k) represent the nodes state vector of the context layer and invisible layer, respectively. W^{xu}, W^{xc}, and W^{yx} are the weights vector between the input and invisible layers, between the context and invisible layers and between the invisible and output layers, respectively. $f(\cdot)$ is a nonlinear function which represents

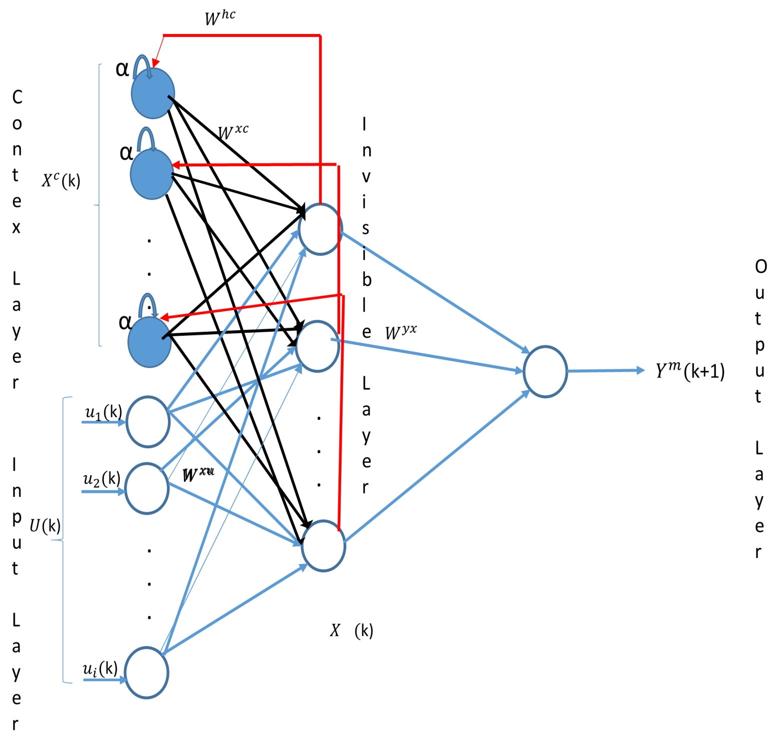


FIGURE 1. The Proposed modified ELMAN Spike Neural Network.

the performance of the MESNN. The self-feedback α, in the context layer is updated in the proposed model until it reaches an accurate value. Fig. 2a represents the internal connection between two neurons in MESNN structure, while, Fig. 2b represents the single synaptic terminal between them. The neuron i in the Fig. 2 is not allowed to spike anymore through the remaining period of time interval T, when the threshold has been exceeded at a particular instant t_i and it will be reset in the next, $t_i + d^k$. Every single connection among the layers in MESNN is composed of a group with the same number of synaptic terminals. Every sub-connection is related with a different weight and delay as it is clear in Fig 2a. The difference between the time of the postsynaptic potential and the firing time of presynaptic neurons i is defined as the delay of the synaptic terminals. The time of postsynaptic potential begins to rise, as seen in Fig.2b,and there is a synapse sequence in the connection.The weight of each synapse effect on the spike-response function ζ represents the activation function of the neuron.

III. SYSTEM IDENTIFICATION

Fig.3 shows the proposed model used for system identification. The input to MESNN is just the present one that is applied to the plant and so, the proposed structure does not need as much information to identify the unknown plant as with a traditional NN. The output of the plant is compared with the output of MESNN to obtain the error, which is used to update its weights. In sum, the idea behind using an Elman NN with an SNN is to enhance the power of the identification process.

IV. THE PROPOSED CONTROLLER

Fig.4 shows MESNN as a controller for the plant, the weights of which are updated during the modified training algorithm, as described latter. $y_p(k)$ is the output of the plant, $r(k)$ is the reference signal, whilst $e(k)$ and $e^{\cdot}(k)$ are the error and difference in error, respectively, whilst D represents the change of error. The parameters of the MESNN structure as a controller learn based on these signals in a closed-loop manner. The one layered MESNN has two input neurons, six neurons in the context layer and invisible layer, with there being one in the output layer. The inputs are the error and the change of error, with seven synapses being used for each connection. The initial values of the weights of MESNN are generated randomly within the interval $[-0.5, 0.5]$ and the learning rate and the self-feedback are initiated at 0.01 and 0.5, respectively, with the training being carried out for 100 epochs. During the synthesis of the input signals of the MESNN control system, the error and the difference of error are converted into spike times. The output signal of MESNN is also spike characterized with a spike time, which is converted into a real value applied as input to the plant. This will be explained in section V.

V. MODIFIED TRAINING ALGORITHM

In this section, the training algorithm used to train MESNN is explained for identifying and controlling the response of dynamic plants. The proposed algorithm is based on the scope of the negative gradient descent method for minimizing the difference between the desired and actual response of the

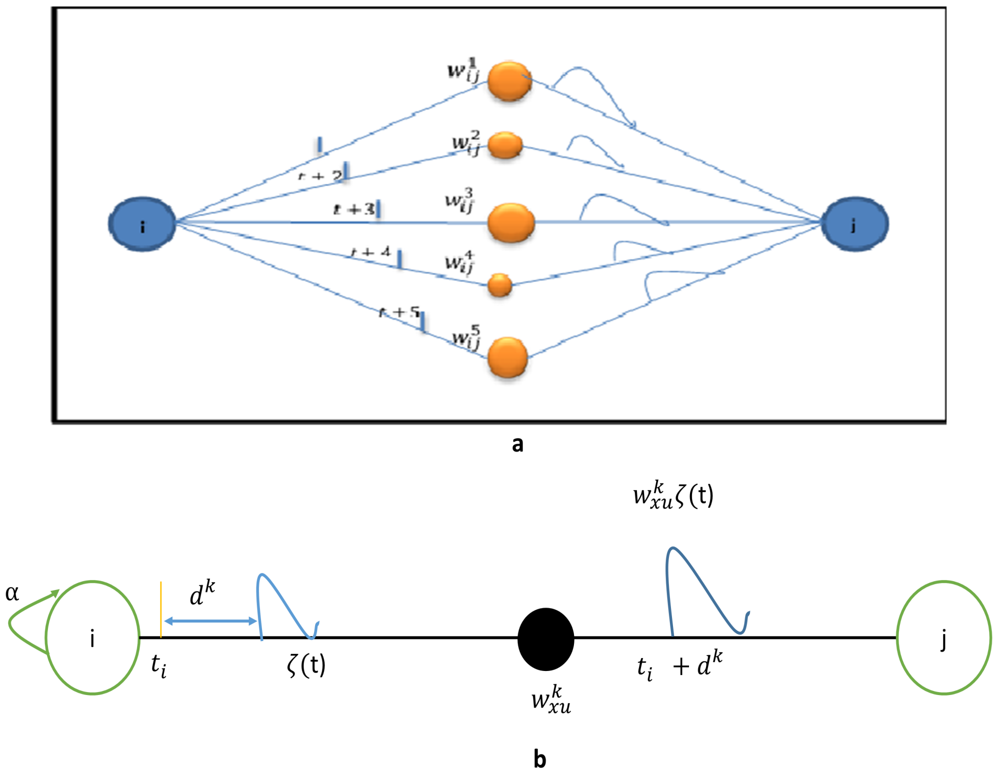


FIGURE 2. a) Internal connection of MESNN. b) Single synaptic terminal.

system to be identified. The parameters trained for this algorithm are the: weights, sub-connection or synaptic delays and threshold. Merging an Elman NN with an SNN is undertaken in order to exploit the strengths that the latter can provide. The presence of many invisible layers decreases the speed of the training process and increases network complexity. The number of sub-connections or synapses in the relationship between the input and invisible layers is updated as well as those between the invisible and output layers. Initially, the weights are randomly chosen and then, after implementing rounds of training, the weights values are tuned more efficiently with a specific learning rate.

The response of the plant is firstly encoded into spike times based on the following equation:

$$t_j^f = t_{max} - \lfloor \frac{t_{min}(rf(t) - rf_{min})(t_{max} - t_{min})}{(rf_{max} - rf_{min})} \rceil. \quad (4)$$

where, rf_{max} and rf_{min} represent the maximum and minimum real response, whilst t_{max} and t_{min} are the maximum and minimum interval time, respectively. The $\lfloor \rceil$ is a round function.

The actual response decoding equation is described as follows:

$$rf(t_j) = \frac{(t_{max} - t_j - t_{min})(rf_{max} - rf_{min})}{(t_{max} - t_{min})} + rf_{min}. \quad (5)$$

There are two modes for the training algorithm. The first is called the feed-forward mode, where each neuron spikes at each time interval T only once at most and this happens when the value of the threshold is exceeded by the membrane potential m. The feed-forward mode always begins from the invisible layer and the neuron is continuously examined to see whether it is spiked or not. The algorithm uses the next neuron when the previous neuron has been spiked. The membrane potential, $m_i(t)$, is calculated by the training algorithm according to the following equation based on the input spikes t_j^f of the neuron at the input layer.

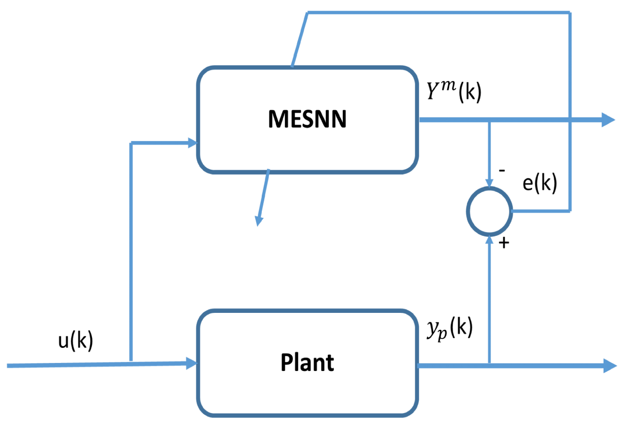


FIGURE 3. The system identification based on MESNN.

$$m_i(t) = \sum_{j=1}^{NH} \sum_{k=1}^{del} w_{xu}^k \zeta(t - t_j^f - d^k)$$
$$+ \alpha * \sum_{i=1}^{NH} \sum_{k=1}^{del} w_{xc}^k * rf_{hi}^k(t-1) \quad (6)$$

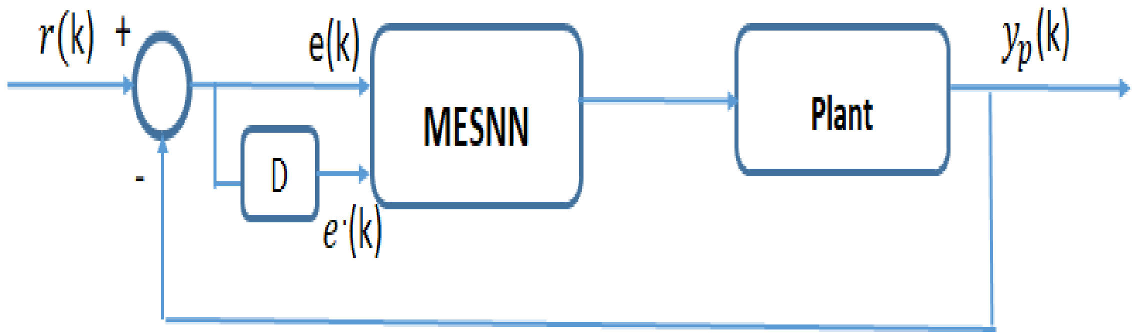


FIGURE 4. The structure of the MESNN based control system.

The term $rf_{hi}^{k}(t-1)$ represents the previous output from the invisible layer and the present input. The function $\zeta(t-t_j^f-d^k)$ is defined as follows:

$$\zeta(t-t_j^f-d^k) = -\sigma * exp[\frac{-(t-t_j^f-d^k)}{\tau}]. \tag{7}$$

The synapse weights of the connection are updated when the feed-forward mode has finished. Opposite to feed-forward, back-propagation begins from the output layer and returns back to the invisible layer. The synapses of the output layer will be updated according to following equations:

$$w_{yx}^k(t+1) = w_{yx}^k(t) - \Delta w_{yx}^k(t). \tag{8}$$

where,

$$\Delta w_{yx}^k(t) = \eta.\delta_j.X^k. \tag{9}$$

The error between the desired spike time of the output neuron and its actual firing time is defined as:

$$E = (T_j^d - t_j^f). \tag{10}$$

The δ_j can be computed as:

$$\delta_j = \frac{E}{\sum_{(i=1)}^{NH}\sum_{(k=1)}^{del} w_{yx}^k \frac{\partial X_i^k}{\partial t}}. \tag{11}$$

The synapses of the invisible layer will be updated according to following equations:

$$w_{xu}^k(t+1) = w_{xu}^k(t) - \Delta w_{xu}^k(t). \tag{12}$$

where,

$$\Delta w_{xu}^k(t) = \eta.\delta_i.U_i^k. \tag{13}$$

δ_i is defined as:

$$\delta_i = \frac{\sum_{(j=1)}^{(NI)} \delta_j \sum_{(k=1)}^{del} w_{xu}^k \frac{\partial U^k}{\partial t}}{\sum_{(k=1)}^{del} w_{xu}^k \frac{\partial xc^k}{\partial t}}. \tag{14}$$

Likewise, the weights of the context layer are updated according to the following equation:

$$w_{xc}^k(t+1) = w_{xc}^k(t) - \Delta w_{xc}^k(t). \tag{15}$$

TABLE 1. Parameters of the MESNN training algorithm.

Symbol	Meaning
m_i	Membrane potential of neuron i
w_{yx}^k	synapses weights of the output layer
w_{xc}^k	synapses weights of the invisible layer
w_{xu}^k	synapses weights of the input layer
k	Step time
η	Learning rate
del	Number of delayed-synapses per connection
T_j^d	desired spike time of output neuron.
t_j^f	actual spike time of output neuron.
d^k	delay of the connection
NH	Number of neurons in the invisible layer
NI	Number of neurons in the input layer
max epoch	Maximum number of epochs
σ	Some constant between (0-1)
T	Time interval
ρ_d	Learning rate of the synaptic delay
ρ_θ	Learning rate of the synaptic thresholds
t	Time counter
θ	The Threshold value
τ	The Time constant
δ	The Delta function

All the symbols in the above equations are described in table 1. The update of the synaptic delay and neuron thresholds are explained in the following equations:

$$\Delta_{del}^k = -\rho_d \sum_{(i=1)}^{(NH)} \frac{\partial E}{\partial t_j^f}\frac{\partial t_j^f}{\partial U_i(t)}\frac{\partial U_i(t)}{\partial del^k}. \tag{16}$$

$$\Delta\theta = -\rho_\theta \sum_{(i=1)}^{(NH)} \frac{\partial E}{\partial t_j^f}\frac{\partial t_j^f}{\partial U_i(t)}\frac{\partial U_i(t)}{\partial \theta_i}. \tag{17}$$

MESNN is adaptive according to the data dynamics of the input pattern that consists of one set for training and others for testing.The training algorithm of MESNN is shown in Fig. 5. and Fig.6,

VI. THE PERFORMANCE OF IDENTIFICATION

In this paper, the simulations are implemented by using real data that consist of one set for training and another for testing in a MATLAB simulator. The training of real data using the modified training algorithm proposed is for identification of

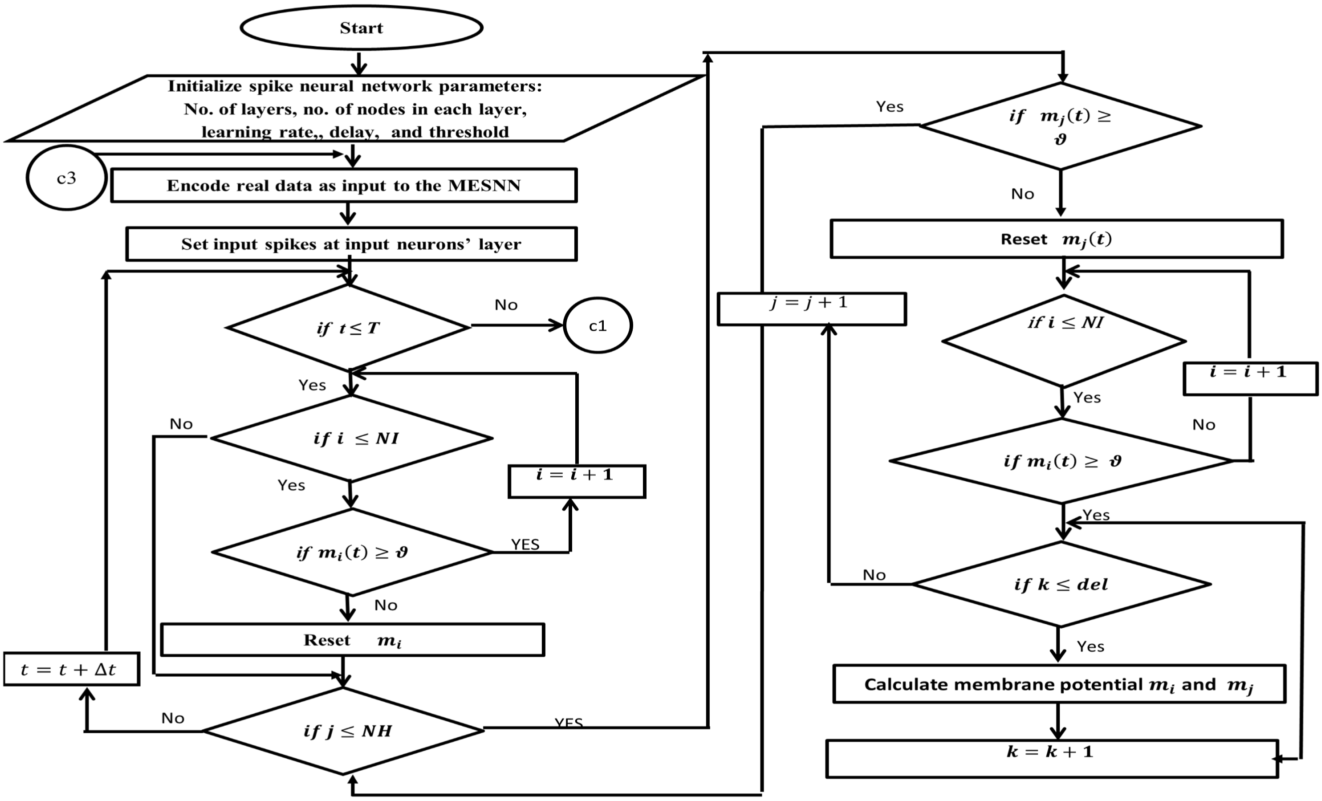


FIGURE 5. The proposed training algorithm.

the plant described by (18) as a case study.

$$y(k) = 0.72 * y(k-1) + 0.025 * y(k-2) * u(k-1) + 0.01 * u^2(k-2) + 0.2 * u(k-3). \quad (18)$$

The MESNN configuration for the identification system of a plant model consists of an input layer with one neuron, a context layer with seven, an invisible layer with seven and an output layer with one. This number of neurons in the context layer is equal to the number in the invisible layer [5]. In the training phase, a set of random inputs is applied to the plant and the response is taken as a target for the training of MESNN with the same input as that applied to the plant. Then in the testing phase, the stimulus signal given in (19) is applied to the system.

$$u(k) = \begin{cases} \sin(\pi k/25), & k < 200 \\ 1.0, & 201 \le k < 400 \\ -1.0, & 401 \le k < 600 \\ 0.4\sin(\pi k/25) & \\ +0.2\sin(\pi k/32) & \\ +0.8\sin(\pi k/10), & 601 \le k < 800. \end{cases} \quad (19)$$

It is clear from (19) that after every 200 units of time the stimulus signal is changed with different types of the shape signals (e.g. sinewave, step, complex sinewave), thus showing the efficiency of the proposed structure in identifying the plant.

Fig. 7 shows the minimization of error during the training process as comparisons between the proposed structure and other structures proposed in [4] and [14]. It shows that MESNN can reach a lower error as compared with the Elman Neural Network and SNN. The error goal is set to (10^-5), so the MESNN reaches to the error goal faster than ENN and SNN. From the results obtained from the Fig. 7 and based on the general formula for calculation of the percentage of error improvement δe in [24], it emerges that MESNN improves by 70% more than ENN in terms of minimizing the error.

$$\delta_e = \frac{E_{ENN} - E}{E_{ENN}} * 100\%. \quad (20)$$

where, E_{ENN} and E represent the error rate of the training with ENN and MESNN, respectively.

The previous studies that we have compared with the present one relied in their proposals on different structures of neural networks. For example, in [4] the researchers adopted SNN with its training algorithm, while in the [14] the researchers used ENN with self feedback.

It can be seen in Fig. 7 that the performance of the proposed structure is better than that for the other structures, due to the combination of the advantages of SNN and ENN. Moreover, it can also be observed that ENN is faster in training at the beginning than SNN, but the latter reaches the error goal in less time than the former. The reason for this is that SNN is approaching the behavior of the human mind, as it relies on coding the data set and converting it into time, instead

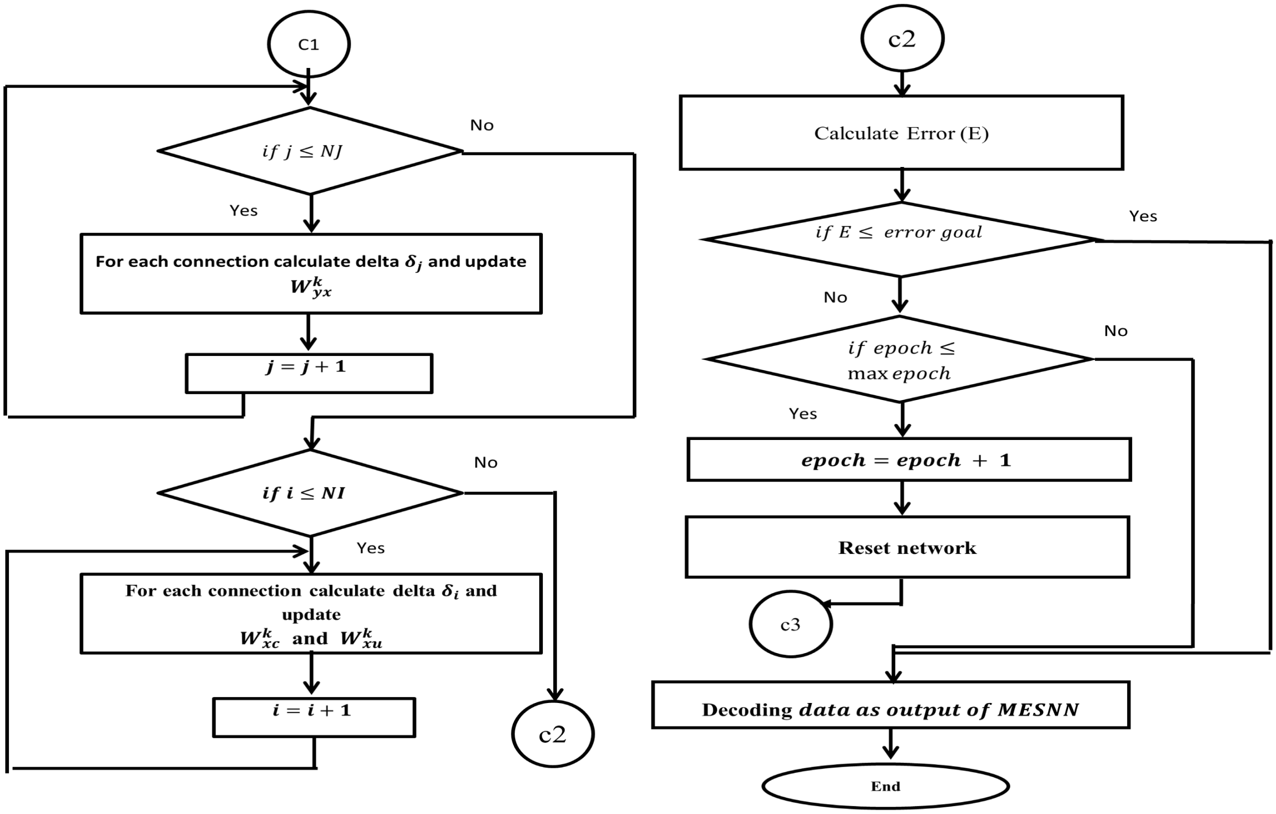


FIGURE 6. Continue: the proposed training algorithm.

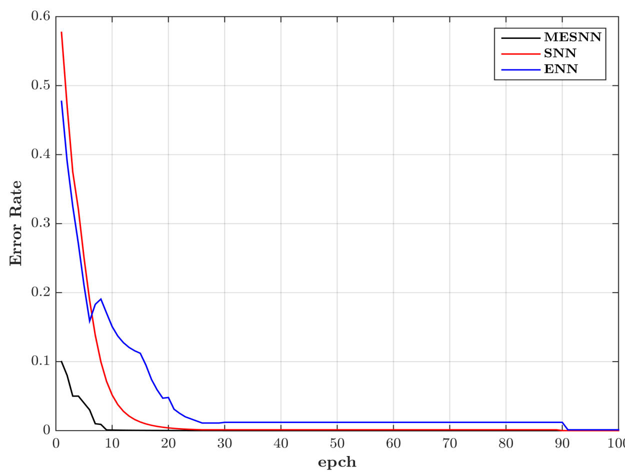


FIGURE 7. The minimization of error during training.

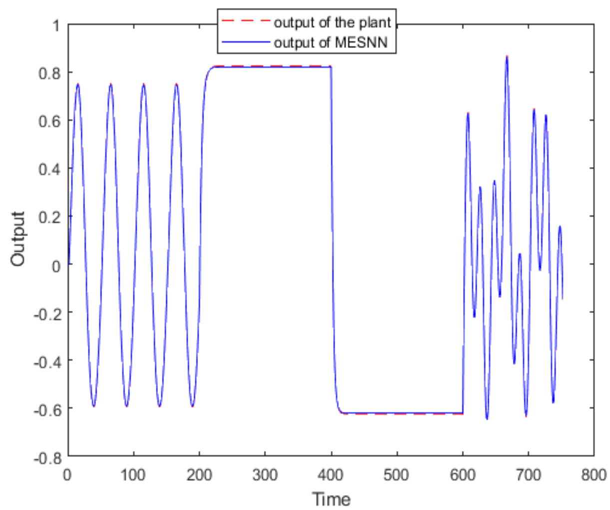


FIGURE 8. The response of the system identification based on MESNN.

of using them as a real data. The self-feedback in MESNN gives the power to it to speed up the training process, thus helping the network to identify and control of dynamic systems more efficiently. In the identification process, enough information about the system needs to be identified and the dynamic behavior of the ENN helps in providing this. In this paper, the proposed structure comprises a combination of the accuracy of SNN and the speed of ENN in the training process.

Fig. 8 shows the response of the plant after applying the stimulus signal,as shown in Fig. 8, with the output of MESNN being very close to that desired, which means that the accuracy of the proposed structure is very high. Fig. 9 shows the response of the plant when ENN is applied as the identifier of a plant as compared with the response when SNN is applied to this end. It is clear from the figure that the performance of SNN is better than that of ENN, which is because of the ability of the former to train the system more efficiently than the latter.

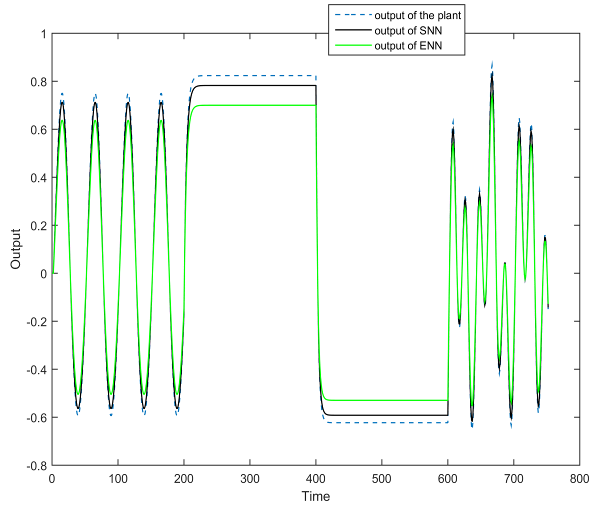


FIGURE 9. **The response of the system identification system based on ENN and SNN.**

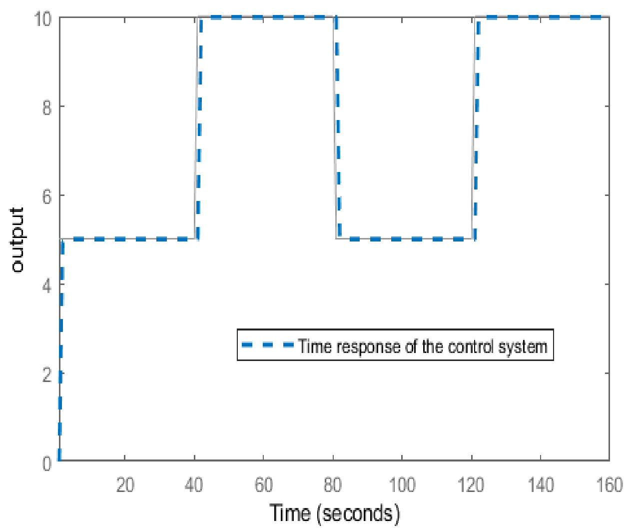


FIGURE 10. **The response of the control system for different set points when MESNN is used.**

VII. THE PERFORMANCE OF THE CONTROLLER

The performance of the proposed structure to the control of dynamic plant is discussed in this section.The input of the control system for different set points $r(k)$ is described as follows:

$$r(k) = \begin{cases} 5, & \text{k} < 40 \\ 10, & 41 \leq \text{k} < 80 \\ 5, & 81 \leq \text{k} < 120 \\ 10, & 121 \leq \text{k} < 160. \end{cases} \tag{21}$$

The response of the control system is shown in Fig.10. As it is shown in Fig. 10, the MESNN is having the potential of reaching out the given set of data points quickly. Fig.11 shows the comparison of the response of the control system for different set points when ENN and SNN are used as the controller. It is clear from the figure that both of the controllers cannot track the set points, but the performance of SNN is better than that of ENN. This is because the ENN is not efficient enough as a controller and its training is not powerful as that of SNN.

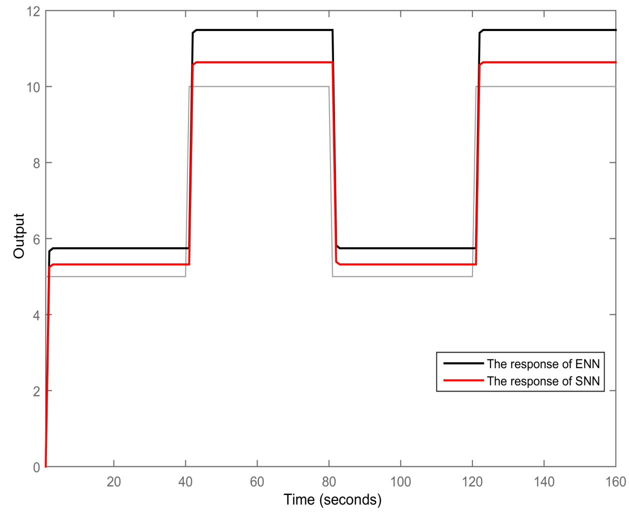


FIGURE 11. **The comparison of the response of the control system for different set points when ENN and SNN are used.**

VIII. CONCLUSION

In this paper, the identification and control of a dynamic plant based on a modified Elman spike neural network MESNN has been proposed. The proposed structure is operationalized by a modified training algorithm. Comparisons between the proposed model and other structures have been made. The performance of the proposed model in the field of system identification is better than for ENN and SNN. This is because MESNN combines the advantages of accuracy in SNN and the structure of ENN in the training phase. The performance of the proposed model is better than that of ENN by 70%, which is evidenced in the simulation results.

In the field of using MESNN as a controller, it is clear that it is better than ENN and SNN, for it can track the set points more accurately than these others. That is, the proposed structure can automatically adapt their parameters and this makes the controller based on MESNN more efficient.

REFERENCES

[1] F. F. M. El-Sousy and K. A. Abuhasel, "Intelligent adaptive dynamic surface control system with recurrent wavelet Elman neural networks for DSP-based induction motor servo drives," *IEEE Trans. Ind. Appl.*, vol. 55, no. 2, pp. 1998–2020, Mar. 2019.

[2] B. Y. Suprapto and B. Kusumoputro, "A comparison of back propagation neural network and Elman recurrent neural network algorithms on altitude control of heavy-lift hexacopter based on direct inverse control," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. (ICECOS)*, Oct. 2018, pp. 79–84.

[3] J. Khun, M. Novotny, and M. Skrbek, "High-performance spiking neural network simulator," in *Proc. 8th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2019, pp. 1–4.

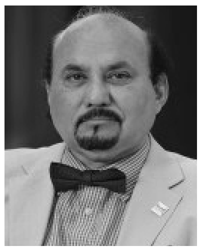
[4] H. Qiu, M. Garratt, D. Howard, and S. Anavatti, "Evolving spiking neural networks for nonlinear control problems," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1367–1373.

[5] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[6] H. Taghavifar, "Neural network autoregressive with exogenous input assisted multi-constraint nonlinear predictive control of autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6293–6304, Jul. 2019.

[7] L. Dong, J. Yan, X. Yuan, H. He, and C. Sun, "Functional nonlinear model predictive control based on adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4206–4218, Dec. 2019.
[8] D. Xu, J. Liu, X.-G. Yan, and W. Yan, "A novel adaptive neural network constrained control for a multi-area interconnected power system with hybrid energy storage," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6625–6634, Aug. 2018.
[9] L. Zhang, Y. Zhu, H. Shi, and K.-S. Hwang, "Adaptive dynamic programming approach on optimal control for affinely pseudo-linearized nonlinear system," *IEEE Access*, vol. 7, pp. 75132–75142, 2019.
[10] G. Shou-Ping and P. Xue-Fei, "An architecture of interval Elman network and its numerical analysis," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 332–336.
[11] A. M. Sadek, W. M. Elawady, and A. M. Sarhan, "Design of a backstepping controller based on an adaptive Elman neural network for a two-link robot system," in *Proc. 13th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2018, pp. 481–487.
[12] R. Deo and R. Chandra, "Identification of minimal timespan problem for recurrent neural networks with application to cyclone wind-intensity prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 489–496.
[13] C.-M. Lin and E.-A. Boldbaatar, "Autolanding control using recurrent wavelet Elman neural network," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 9, pp. 1281–1291, Sep. 2015.
[14] K. Dahmani, K. Elleuch, F. Fourati, and A. Toumi, "Adaptive neural control of a greenhouse," in *Proc. 19th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. (STA)*, Mar. 2019, pp. 59–63.
[15] Y. Tian, H. Yan, H. Zhang, S. X. Yang, and Z. Li, "Observed-based finite-time control of nonlinear semi-Markovian jump systems with saturation constraint," *IEEE Trans. Syst., Man, Cybern., Syst.*, Jan. 22, 2020, doi: 10.1109/TSMC.2019.2962318.
[16] Y. Oniz, O. Kaynak, and R. Abiyev, "Spiking neural networks for the control of a servo system," in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, Feb. 2013, pp. 94–98.
[17] R. H. Abiyev, O. Kaynak, and Y. Oniz, "Spiking neural networks for identification and control of dynamic plants," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2012, pp. 1030–1035.
[18] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227–3235, Aug. 2017.
[19] A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5394–5407, Nov. 2018.
[20] Y. Miao, H. Tang, and G. Pan, "A supervised multi-spike learning algorithm for spiking neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
[21] F. Han, R. Li, and D. Qian, "Short-term wind speed forecasting model based on spiking neural network," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Aug. 2018, pp. 359–363.
[22] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.
[23] R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Machine learning techniques for traffic identification and classifiacation in SDWSN: A survey," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 4645–4650.
[24] B. M. Mahmmod, A. R. Ramli, T. Baker, F. Al-Obeidat, S. H. Abdulhussain, and W. A. Jassim, "Speech enhancement algorithm based on super-Gaussian modeling and orthogonal polynomials," *IEEE Access*, vol. 7, pp. 103485–103504, 2019.

NADIA ADNAN SHILTAGH AL-JAMALI (Member, IEEE) received the B.Sc. degree in control and systems engineering, the M.Sc. degree in control engineering, and the Ph.D. degree in computer engineering from the University of Technology, Baghdad, Iraq. She is currently working with Brunel University London, U.K. Her fields of interest are computer control, wireless sensor networks, intelligent systems, neural networks, and robotics.



HAMED S. AL-RAWESHIDY (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the master's Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991, all in electronic engineering. He has worked with the Space and Astronomy Research Center, Baghdad, Perkin Elmer, Waltham, British Telecom, Oxford University, Manchester Metropolitan University, and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network and Communications Centre, Brunel University London, U.K.