The Emerald Research Register for this journal is available at
http://www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
http://www.emeraldinsight.com/0332-1649.htm

Modelling and simulation with neural and fuzzy-neural networks of switched circuits

Yakup Demir and Ayşegül Uçar
Department of Electrical and Electronics Engineering, Firat University, Elazig, Turkey

Received August 2001
Revised July 2002
Accepted July 2002

Keywords *Artificial neural networks, Fuzzy logics, Electrical circuits, Simulation, Non-linear analysis*

Abstract *Recently, the modelling and simulation of switched systems containing new nonlinear components in electronics and power electronics industry have gained importance. In this paper, both feed-forward artificial neural networks (ANN) and adaptive network-based fuzzy inference systems (ANFIS) have been applied to switched circuits and systems. Then their performances have been compared in this contribution by developed simulation programs. It has been shown that ANFIS require less training time and offer better performance than those of ANN. In addition, ANFIS using "clustering algorithm" to generate the rules and the numbers of membership functions gives a smaller number of parameters, better performance and less training time than those of ANFIS using "grid partition" to generate the rules. The work not only demonstrates the advantage of the ANFIS architecture using clustering algorithm but also highlights the advantages of the architecture for hardware realizations.*

1. Introduction

In the last few decades, many types of new nonlinear components have appeared in electrical circuits and systems. As a result, the analysis of such circuits and systems has gained more importance and therefore, a lot of scientists have developed and published quite many analysis methods. One of these is the piecewise linearization approach and nonlinear components were modelled by linear time invariant components and ideal switches (Demir *et al.*, 1996; Koksal, 1984). However, unlike the linear case, the operating conditions (switching sequence and switching time) of these switches are not preknown in nonlinear systems. This necessitates the use of the so-called control inequalities additionally. Nevertheless in this paper, the switching sequence and switching time of the switches are assumed to be preknown.

Electrical circuit models can be broadly divided into two classes: first principles models and empirical models. First principles models are based on circuit knowledge, and hence, they are reliable. Whenever feasible, these should be developed and utilised. However, the development of principles models is sometimes time consuming and effort demanding, especially for nonlinear circuits.

COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering
Vol. 22 No. 2, 2003
pp. 253-272

0332-1649
DOI 10.1108/03321640310459199

Artificial neural networks (ANN) have been successfully applied to a wide variety of system modelling and control applications by many researchers (Borchers, 1998; Cannas *et al.*, 1998; Choi and Farrell, 2000). Indeed, they offer massive parallel distributed processing, generalising properties and especially the learning property. An overwhelming majority of these applications have used the backpropagation algorithm for optimising the networks. Although the backpropagation algorithm has unquestionably been a major factor for the success of past neural network applications, it is plagued with disadvantages, e.g. convergence to the global optimum is not guaranteed and convergence tends to be extremely slow. Therefore, several algorithms based on Newton's method have been developed, but these show several limitations, e.g. they can be difficult to interpret and can lack robustness when applied to data outside the training interval (Efe *et al.*, 1999a, b; Nørgaard, 1997).

One approach to improve the model robustness is based upon the combined use of both circuit knowledge and circuit input output data. Circuit knowledge can, for example, be used to decompose the circuit operation into a number of local operating regions such that, within each region, a reduced order linear model can be used to approximate the local behaviour of the circuit. In fact, a nonlinear circuit can always be locally linearized around a particular operating point. The locally linearized model is valid within a region around that operating point. Fuzzy sets provide an approximate means for defining operating regions since the definition of local operating regions is often vague in nature and there usually exists overlapping among different regions. This leads to the fuzzy modelling approach (Efe *et al.*, 1999a, b; Gomez-Skarmeta *et al.*, 1999; Park *et al.*, 1999; Zawirski *et al.*, 1999). The major difficulty in fuzzy modelling is how to decide on the parameters for the fuzzy membership functions (MFs). The parameters are often specified by users, either from their own experience or from trial and error. However, introducing some automation might relieve this difficulty, as shown in recent researches.

In the last decade, the similarity of nonlinear systems modelling using ANNs and fuzzy logic systems (FLS) has been researched and the advantages of fuzzy reasoning and neural networks by neuro-fuzzy networks have been combined. These advantages include fast and accurate learning, good generalisation capabilities, excellent explanation facilities in the form of semantically meaningful fuzzy rules, and the ability to accommodate both data and existing expert knowledge about the problem given below. Different fuzzy-neural networks architectures have been used for the solution of many engineering problems (Jantzen, 1998; Rashid *et al.*, 2000). One of these is the adaptive network-based fuzzy inference systems (ANFIS) architecture to represent fuzzy models (Jang and Chuen-Tsai, 1993). The ANFIS architecture cannot only use linguistic information but also adapt itself using numerical data to achieve better performance. In any case, the determination of the number and locations of fuzzy rules is necessary for its success. For this, several techniques proposed in the literature include techniques like min-max, clustering and grid partition techniques (Djukanovic *et al.*, 1997; Ellithy and Al-Naamany, 2000; Jang and Chuen-Tsai, 1995).

In this paper, the switched circuits have been modelled by both ANN and ANFIS architectures for the first time, and the behaviour for different inputs has been simulated. To train ANN and to generate the rules and the numbers of MFs of ANFIS, two different algorithms are used. For this purpose, all simulation programs have been developed in the contribution by The Mathworks's MATLAB. Simulation results for the example circuit are given. Furthermore, the simulation results of the developed ANN and ANFIS models have been comparatively evaluated in terms of error, parameter number, and training time.

2. Theoretical background

The aim of this section is to provide a theoretical background.

2.1 ANN for modelling

ANN is a network architecture which gives very good performance for the modelling of nonlinear systems. In Figure 1, the architecture of the feedforward ANN with n input neurons, one hidden layer with m hidden neurons and one output neuron is shown.

In this paper, the ANN has been trained with the backpropagation algorithm using an adaptive learning rate and the Levenberg-Marquardt algorithm. ANN-1 and ANN-2 indicate the ANNs trained with the backpropagation algorithm and the Levenberg-Marquardt, respectively. The Levenberg-Marquardt benefits from additional information coming from the second derivatives of the cost function for iteratively solving for the weights. Hence, it is faster than the backpropagation algorithm but has a larger computational complexity and too much hardware requirements (Efe *et al.*, 1999a, b; Jang *et al.*, 1997; Nørgaard, 1997).

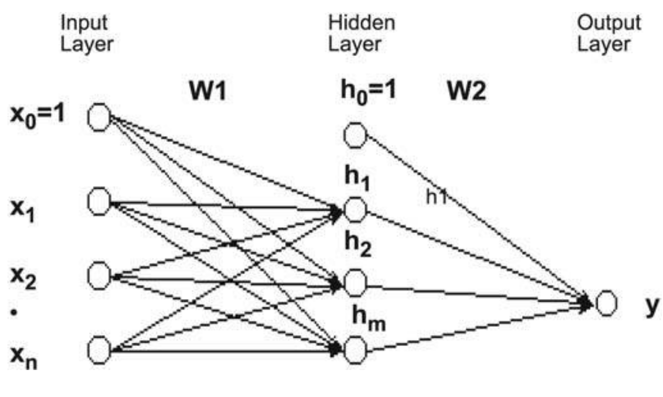


Figure 1.
ANN architecture

2.2 ANFIS for modelling

ANFIS is one of the proposed methods to combine fuzzy and neural systems. It contains rule base and database (knowledge base), fuzzification and defuzzification unit as well as a decision-making unit. The system makes use of a hybrid learning rule to optimise the fuzzy system parameters of a first order Sugeno system (Jang and Chuen-Tsai, 1993).

The ANFIS architecture with two inputs, two rules and one output is graphically represented in Figure 2 for a typical fuzzy rule set:

IF **x is A**$_1$ and **z is B**$_1$ THEN $y_1 = p_1x + q_1z + r_1$

IF **x is A**$_2$ and **z is B**$_2$ THEN $y_2 = p_2x + q_2z + r_2$

The layers for ANFIS in this architecture are defined as follows:

Layer 1: Every node "*i*" in this layer is an adaptive node with a node output defined by

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x), \quad \text{for } i = 1, 2 \text{ or} \\ O_{1,i} &= \mu_{B_{i-2}}(z), \quad \text{for } i = 3, 4 \end{aligned} \tag{1}$$

where x (or z) is the input relating to the node, A_i (or $B_{i\text{-}z}$) is a fuzzy set associated with this node, $\mu_{A_i}(x)$ is the MF of x in A. In this study, the Gaussian and the trapezoidal MFs characterised by equations (2) and (3) are used. $\{a_j, b_j, c_j, d_j, \sigma_j\}$ is the nonlinear premise parameter set in this layer.

$$\mu_{A_j}(x) = e^{-\left(\frac{x - c_j}{\sigma_j}\right)^2} \tag{2}$$

$$\mu_{A_j}(x) = \max\left(\min\left(\frac{x - a_j}{b_j - a_j}, 1, \frac{d_j - x}{d_j - c_j}\right), 0\right) \tag{3}$$

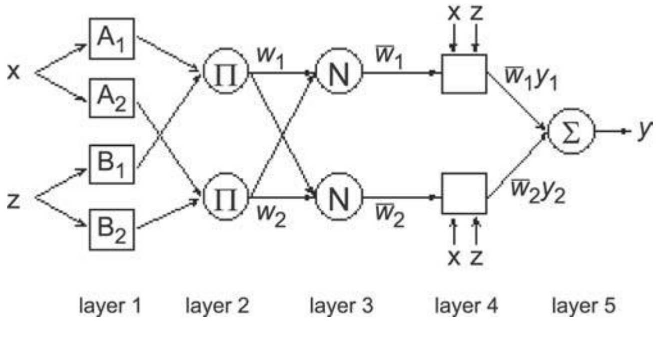


Figure 2.
ANFIS architecture for the two-input two-rule Sugeno fuzzy model

Layer 2: Every node "*i*" in this layer is a two-fixed node, which multiplies the incoming signals and outputs the product, as defined in equation (4).

$$O_{2,i} = w_i = \mu_{A_j}(x)^* \mu_{B_j}(z), \quad i = 1, 2; \; j = 1, 2 \tag{4}$$

Each node output represents the firing strength of the rule.
Layer 3: Every node "*i*" in this layer is a fixed node. The *i*th node calculates the normalized firing strength:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{5}$$

Layer 4: Every node "*i*" in this layer is an adaptive node with a node function

$$O_{4,i} = \bar{w}_i y_i = \bar{w}_i (p_i x + q_i z + r_i) \tag{6}$$

where $\{p_i, q_i, r_i\}$ is the linear consequent parameter set in this layer.
Layer 5: The single node in this layer is a fixed node, which computes the overall output as the summation of all incoming signals:

$$O_{5,1} = \text{overall output} = \sum_i \bar{w}_i y_i = \frac{\sum_i w_i y_i}{\sum_i w_i} \tag{7}$$

The hybrid learning algorithm of ANFIS adjusts the consequent parameters in a forward pass and the premise parameters in a backward pass. In the forward pass, the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backward and the premise parameters are updated by gradient descent.

Here, the rules and the numbers of MFs are obtained by two methods. In the first method, after the numbers of MFs for the inputs are determined, it generates rules by enumerating all possible combinations of MFs of all inputs. This way to generate the rules is called grid partition (Jang and Chuen-Tsai, 1995). It leads to an exponential explosion even when the numbers of inputs are moderately large. For a fuzzy inference system with 10 inputs, each with two MFs, the grid partitioning leads to $2^{10} = 1{,}024$ rules, which is inhibitively large for any practical learning method. In the second method, the rules and the MFs are generated by a subtractive clustering algorithm. The subtractive clustering is based on a measure of the density of data points. The aim is to find regions with high densities of data points. The data point with the highest potential is selected as the centre for a cluster. The data points within a prespecified fuzzy radius are removed (subtracted), and the algorithm looks for a new point with the highest number of neighbours. This process continues until all data points have been tested. An election of K data points is specified by m-dimensional vectors and normalised. Since each data point is a candidate for a cluster centre, a density measure at data point u_k is defined as

$$D_k = \sum_{j=1}^{K} \exp\left(-\frac{\|u_k - u_j\|^2}{(r_\mathrm{a}/2)^2}\right), \tag{8}$$

where $k = 1, 2, 3, \ldots, K$ and r_a is a positive constant. After calculating the density measure for each data point, the point with the highest density is selected as the first cluster centre. Let $u_{\mathrm{c}1}$ be the selected point and $D_{\mathrm{c}1}$ its density measure. Next, the density measure for each data point u_k is revised by;

$$D_k^1 = D_k - D_{\mathrm{c}1} \exp\left(-\frac{\|u_k - u_{\mathrm{c}1}\|^2}{(r_\mathrm{b}/2)^2}\right), \tag{9}$$

where r_b is larger than r_a to prevent closely spaced cluster centres. Therefore, the data points near the first cluster centre $U_{\mathrm{c}1}$ will have significantly reduced density measures, thereby making the points unlikely to be selected as the next cluster centre. After the density measure for each point is revised, the next cluster centre $U_{\mathrm{c}2}$ is selected and all density measures are revised again.

The process is repeated until a sufficient number of cluster centres are generated. When applying subtractive clustering to a set of input-output data, each of the cluster centres represents a rule. To generate rules, the cluster centres are used as the centres for Gaussian MFs in clustering algorithm (Jang *et al.*, 1997; Jantzen, 1998). In this paper, ANFIS-1 and ANFIS-2 indicate the ANFIS's used grid and clustering methods, respectively.

3. Formulation and modelling of switched circuits

The following steps outline the development of the neural and fuzzy-neural models.

(1) Collect ample training and checking data as produced by the target model.

(2) Use ANN and ANFIS to create the neural and the fuzzy-neural models that relates capacitor voltage signal to the source voltage of the switched circuits.

(3) Validate the new model through comparison of its output to the output of the target model, given identical inputs.

3.1 Data collection

By using piecewise linearization, nonlinear components are replaced by their equivalent linear models containing ideal switches so that depending on their position, each piecewise linear region (system state) in the nonlinear characteristics is realized. In each system state, the state and output equations of switched circuits are written in the form (10).

$$\begin{bmatrix} \dot{X}_k(t) \\ Y(t) \end{bmatrix} = \begin{bmatrix} A_k \\ B_k \end{bmatrix} [X_k(t)] + \sum_{i=0}^{j_k} \begin{bmatrix} B_{k,i} \\ D_{k,i} \end{bmatrix} \frac{d^i}{dt} U(t) \tag{10}$$

In this equation, $X_k(t)$ is the vector of state variables, $Y(t)$ is the vector of output (response) variables, and $U(t)$ is the source vector; A_k, B_k, $B_{k,i}$, $D_{k,i}$ are constant coefficient matrices of proper dimensions. Note that for each system state, k ($k = 1, 2, \ldots, n_s$; where n_s is the total number of the states) different state vectors and different coefficient matrices may occur. In this example, in order to clearly observe the performance of ANN and ANFIS models, the switched circuit shown in Figure 3 is considered. For this circuit, four system states are assumed and the remaining period in each system state is 1 s. The output to be modelled with respect to the input is the capacitor voltage, V_c. The program has produced the following state and output equations.

The first system state (S1 and S2 are off):

$$\frac{d}{dt}\begin{bmatrix} V_c \\ I_L \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ -2 & -2 \end{bmatrix}\begin{bmatrix} V_c \\ I_L \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix}[V_i], \tag{11a}$$

The second system state (S1 is off, S2 is on):

$$\frac{d}{dt}\begin{bmatrix} V_c \\ I_L \end{bmatrix} = \begin{bmatrix} -1.666 & 8.333 \\ -1.666 & -1.666 \end{bmatrix}\begin{bmatrix} V_c \\ I_L \end{bmatrix} + \begin{bmatrix} 1.666 \\ 1.666 \end{bmatrix}[V_i], \tag{11b}$$

The third system state (S1 is on, S2 is off):

$$\frac{d}{dt}\begin{bmatrix} V_c \\ I_L \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ -2 & 0 \end{bmatrix}\begin{bmatrix} V_c \\ I_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}[V_i], \tag{11c}$$

The fourth system state (S1 and S2 are on):

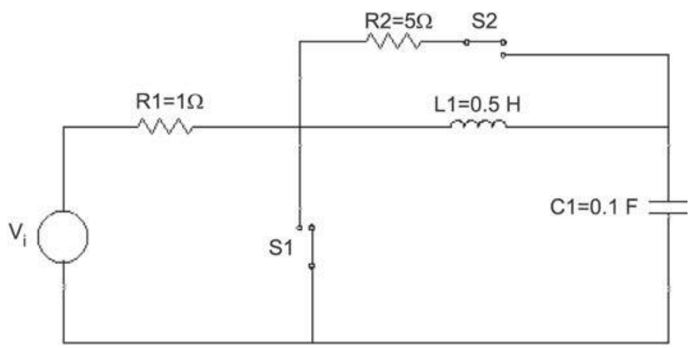


Figure 3.
Switched circuit relating to example

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} V_\mathrm{c} \\ I_L \end{bmatrix} = \begin{bmatrix} -2 & 10 \\ -2 & 0 \end{bmatrix}\begin{bmatrix} V_\mathrm{c} \\ I_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}[V_i], \tag{11d}$$

The output equations for all system states are produced to be

$$[V_\mathrm{c}] = [1 \quad 0]\begin{bmatrix} V_\mathrm{c} \\ I_L \end{bmatrix} + [0][V_i]. \tag{12}$$

In this study, data for training and checking are obtained from the mathematical models of the switched circuits proposed. One of the important features of nonlinear circuits is to vary nonlinearly depending upon the initial conditions. Herein, initial values relating to each system state are computed by the program. Firstly, the combinations are formed for all different positions of the switches, secondly this is combined with models relating to each linear state and lastly training and checking are performed on this single model by using simulation programs. Operation ranges of the input and the output signals in the circuit are ± 2 and ± 1 V, respectively. Limits of these quantities are dependent upon the specific application of the switched circuits. To insure creation of a valid model, data used for training must thoroughly cover the range of operation in which the circuit will function. For this reason, the training data contains the samples of response (output) of circuits to it and the input signal randomly distributed with 1,197 samples in the range $[-2{:}2]$ as shown in Figure 4. The circuit equations are used for a simulation time of 4 s. The time steps of 0.003304 and 0.0104 s are used to produce a total of 1,197 sets and 397 sets of data for both training and checking.

3.2 Training of the models

With training and checking data established, ANN and ANFIS are used to create a neural system and a fuzzy-neural system that emulates the behaviour of the switched circuits. Both the ANN and the ANFIS are designed with five inputs. Generally, the input number for the neural and the fuzzy neural models is the order of the transfer functions, and it is chosen based on system dynamics and delays. In this study, three of these inputs are from the input of the switched circuit through the tapped-delay line (TDL) block, the others are from the output of the switched circuit through another TDL block, as shown in Figure 5.

One of the most important problems in the modelling with ANN is what architecture should be used for a given problem. Architecture selection requires choosing both the appropriate number of hidden units and the connections.

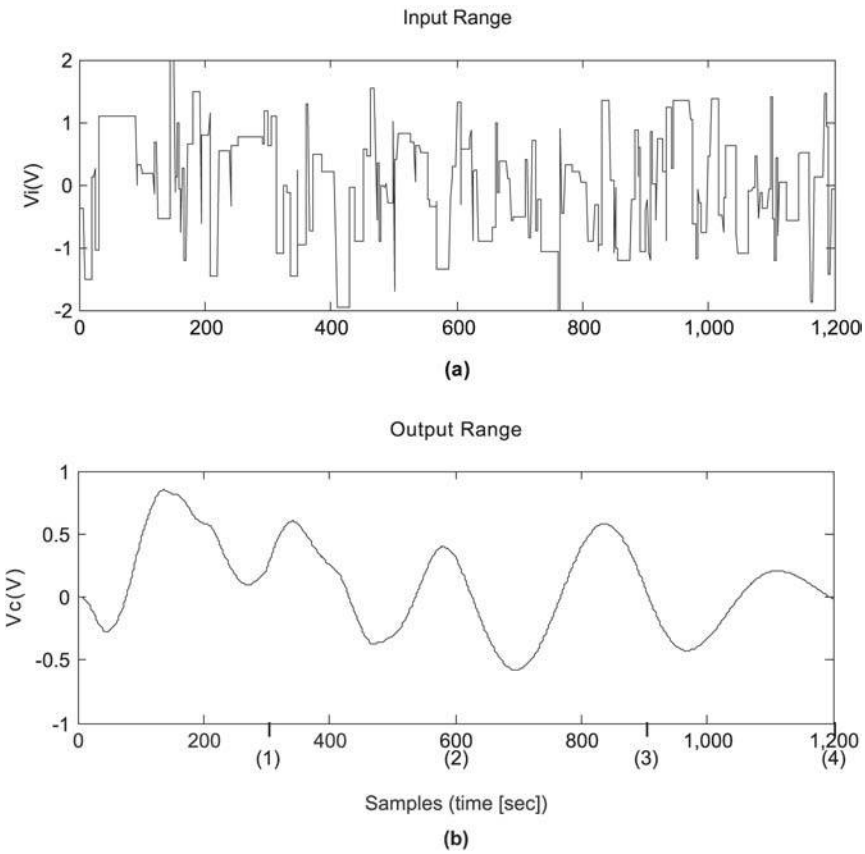


Figure 4. In order to train ANN and ANFIS models (a) the input signal randomly distributed in the range [−2:2] while data size is 1,197, and (b) the response of circuit in Figure 3 to the input signal randomly distributed

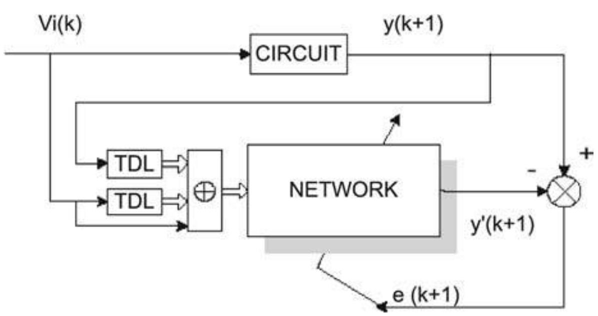


Figure 5. Modelling structure for switched circuits

In order to obtain a good model in this study, we used ANN models with 5,6,7,8,9. . .34 hidden neurons and a feedforward ANN with three layer having a logistic activation function in the hidden layer and a linear activation function in the output layer for this architecture. However, the ANFIS-1 used here contains two trapezoidal MFs assigned to each input variable. We also tried ANFIS-1 models with two and/or three MFs for each input, but these models are too complex and reduce the model performance. In addition, to reduce the number of parameters in the MFs in this study, different MFs can be used, but they are not too effective the model performance. The MFs before and after training for the ANFIS-1 and ANFIS-2 models are shown in Figure 6.

4. Numerical results and discussion

The aim of this section is to discuss the performances of all model architectures.

4.1 Assessment of model performance

In order to assess the accuracy of methods considered, the root mean square error (RMSE) criterion has been applied. Figure 7 shows the RMSE curves which indicate the learning of networks architectures according to epoch number. Table I also summarizes the training RMSE and checking RMSE values for some ANN and the ANFIS architectures by two different sized training and checking data sets. Each value is the result of averaged values obtained from ten runs. For the ANN-1 and ANN-2, these 10 runs were started from different sets of initial random weights for 75,000 epochs and 150 epochs, respectively. For ANFIS, these ten runs correspond to ten step size values ranging from 0.01 to 0.1 for 4 epochs. As seen from these numerical results, even using 397 training data, the ANFIS-2 architecture results in a good modelling performance in terms of the training and checking error. But compared to the ANFIS-2 architecture, the ANN-2 architecture achieves a good modelling performance with 1,197 data requiring more training data. In this study, for all comparisons we used the ANN 5-6-1 architecture since it has small number of parameters and a good modelling performance and trained both the ANN and ANFIS architectures using 1,197 data.

The overall assessment of the approaches is given in Table II. The approaches are evaluated for three different comparison measurements. The first column accounts for the need for training time. In this sense, ANFIS-2 is the best approach. The second column considers the training epoch number. Note that the ANN-2 and ANFIS structures are comparatively evaluated in 150, 4 epochs, respectively. Because even we increased the training epochs for the ANN-2 and ANFIS-1 and 2, their performances did not vary very much (see their curve levels off after 100 epochs, as shown in Figure 7). Unfortunately, the best result for ANN-1 is obtained in 75,000 epochs. The last column includes the number of parameters of models. The total number of modifiable

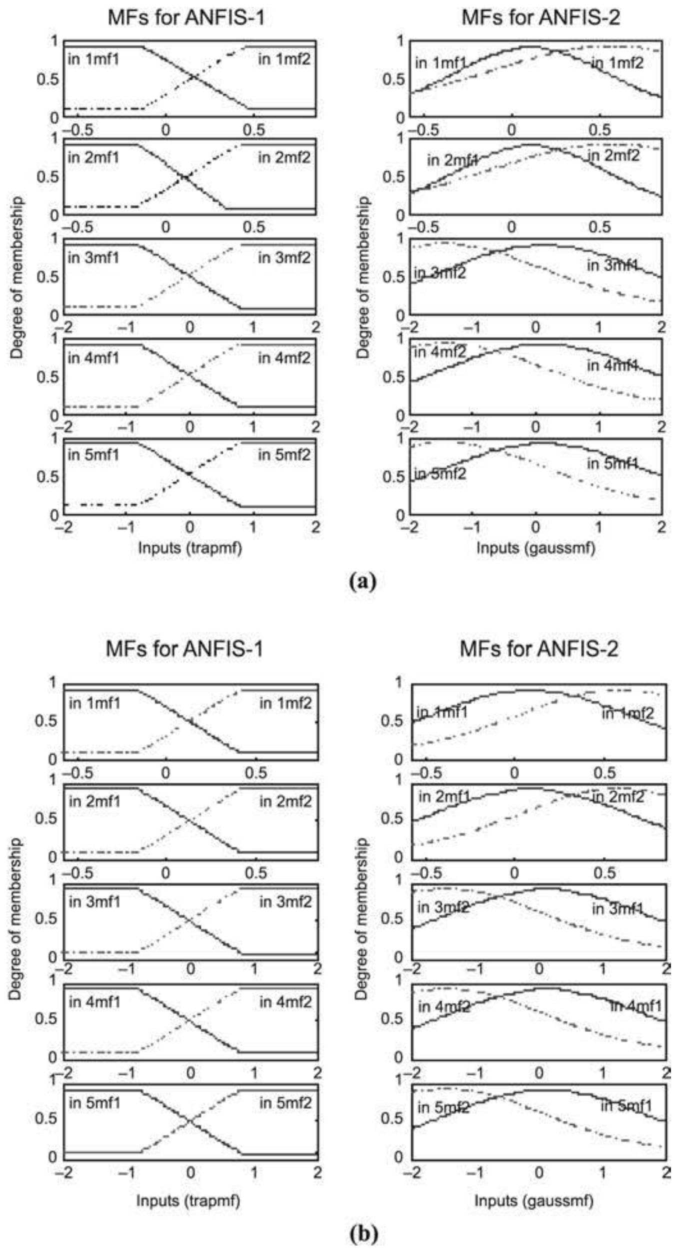


Figure 6.
ANFIS MFs (a) before training, and (b) after training

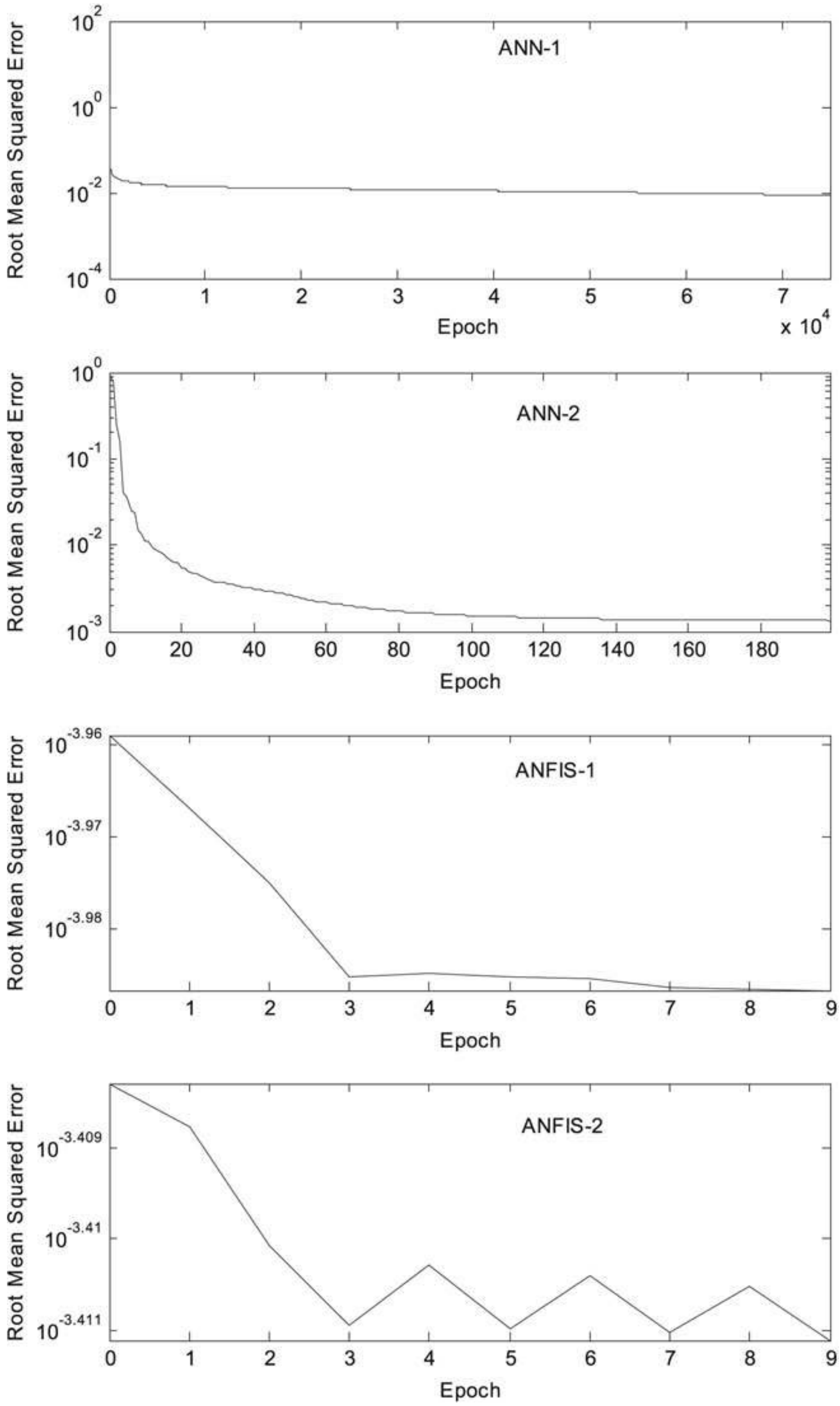


Figure 7.
Training RMSE curves for modelling while training data size is 1,197

parameters in the ANN is (5 input nodes $\times$ 6 hidden nodes) + (1 input bias $\times$ 6 hidden nodes) + (1 hidden node bias $\times$ 1 output node) + (6 hidden nodes $\times$ 1 output node) = 43 modifiable parameters. On the other hand, the total number of parameters available for tuning in the ANFIS-1 which equals the number of modifiable linear parameters is (number of rules) $\times$ (number of inputs+1) $= 2^5 \times (5+1) = 192$ and number of modifiable nonlinear parameters is (number of MF parameters) $\times$ (number of MFs $\times$ number of inputs) $= 4 \times 2 \times 5 = 40$ and total number of modifiable parameters is $192 + 40 = 232$ modifiable parameters. The total number of modifiable parameters available for tuning in the ANFIS-2 is 32 modifiable parameters by adopting the rules and two Gaussian MFs for each input itself with 5 input number.

From all numerical results, it is revealed that ANFIS-2 architecture has a better performance than other architectures in terms of low training and checking error and smaller number of parameters and smaller learning speed.

4.2 Model validation

To validate accuracy of the neural and fuzzy-neural models, it is necessary to compare graphically and numerically the behaviour of the mathematical models when subjected to identical inputs. To this end, the arithmetic model is used to generate four new sets of data. Attributes of these sets, including the unit step and sin(0.1**t*) input signals with 1,197 and 397 samples, are presented in Table III. This table clearly shows that the nonlinear dynamic behaviour of

	Training data size: 1,197 Checking data size: 397		Training data size: 397 Checking data size: 1,197	
Network	Train RMSE	Check RMSE	Train RMSE	Check RMSE
5-6-1 ANN-1	**0.00817**	**0.00978**	**0.00945**	**0.04911**
5-12-1 ANN-1	0.00791	0.00896	0.00835	0.04852
5-24-1 ANN-1	0.00762	0.00863	0.00812	0.04003
5-6-1 ANN-2	**0.00098**	**0.00937**	**0.00381**	**0.02233**
5-12-1 ANN-2	0.00091	0.00855	0.00304	0.02229
5-24-1 ANN-2	0.00087	0.00823	0.00291	0.02013
ANFIS-1	**0.00024**	**0.00110**	**0.00201**	**0.00989**
ANFIS-2	**0.00011**	**0.00101**	**0.00101**	**0.00557**

Table I. Comparison of tracking and checking errors for different data size

Network	Training time	Epoch number	Parameter number
ANN-1	16 min	75,000	43
ANN-2	15 s	150	43
ANFIS-1	1.3 min	4	232
ANFIS-2	13 s	4	32

Table II. Comparison of the various performance measures for ANN and ANFIS the switched circuit have been accurately captured essentially by the proposed ANN-2 and ANFIS-2 architectures.

Figures 8 and 9 depict a graphical comparison for two of validation sets 1, 2. These figures compare the output of the networks and mathematical models. The simulation results of the proposed models are shown in Figure 8 for unit step input and in Figure 9 for sin (0.1*t) input, respectively. It is worth pointing out that it is difficult to discern a difference between the responses of the mathematical model and neural and fuzzy neural networks confirming the capability of the proposed models to emulate behaviour of the switched circuits. These figures show that the use of an ANFIS or an ANN for modelling produces reasonable results for switched circuits of a type similar to the considered circuit. The highlight side of ANFIS system is that it is able to precisely model the uncertainty and imprecision within the data as well as to incorporate the learning ability of ANNs. Compared to ANNs, an important advantage of ANFIS is its reasoning ability (if-then rules) of any particular state. A simple example of a learned for ANFIS architecture is presented below.

Rule 1: *if input1 is in1mf1 and input2 is in2mf1 and input3 is in3mf1 and input4 is in4mf1 and input5 is in5mf1, then*;

$V_c = 2.0061.input1 - 1.0063.input2 - 0.0004.input3 + 0.0011.input4 + 0.0001.input5 + 0.0008$

. . ..

Rule 32: *if input1 is in1mf2 and input2 is in2mf2 and input3 is in3mf2 and input4 is in4mf2 and input5 is in5mf2, then*;

$V_c = 1.9412.input1 - 0.9402.input2 - 0.0002.input3 + 0.0002.input4 + 0.0001.input5 - 0.0009$

where: {in1mf1, in1mf2};{ in2mf1, in2mf2}; {in3mf1,in3mf2};{in4mf1, in4mf2}; {in5mf1 in5mf2} characterised by:

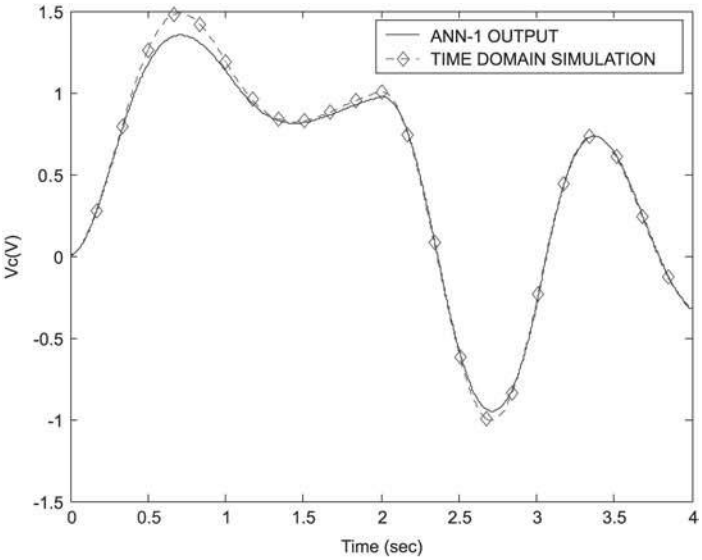
{−1.5785, −1.0069, −0.1270, 0.4746}, {−0.1224, 0.4448, 1.2795, 1.8511};

. . ..

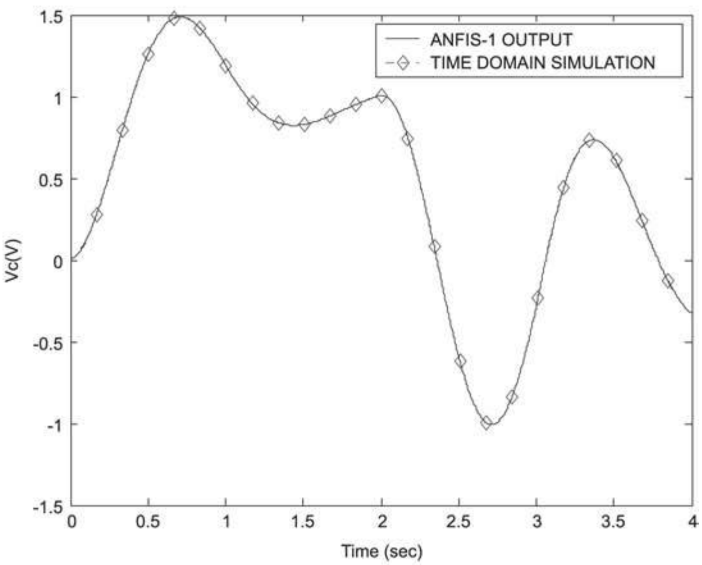
{ − 4.8000, −3.2000, −0.7982, 0.8042}, { − 0.8034, 0.8018, 3.2000, 4.8000}, respectively.

Table III. Comparison of the modelling performances for ANN and ANFIS in the validation data sets

Network	**1.**sin(0.1*t) input with 1,197 samples	**2.**Unit step input with 1,197 samples	**3.**sin(0.1*t) input with 397 samples	**4.**Unit step input with 397 samples
ANN-1	0.009251	0.00912	0.00965	0.01014
ANN-2	0.00062	0.000791	0.00900	0.009501
ANFIS-1	0.0000381	0.000476	0.001176	0.003570
ANFIS-2	0.000226	0.000317	0.001170	0.002576



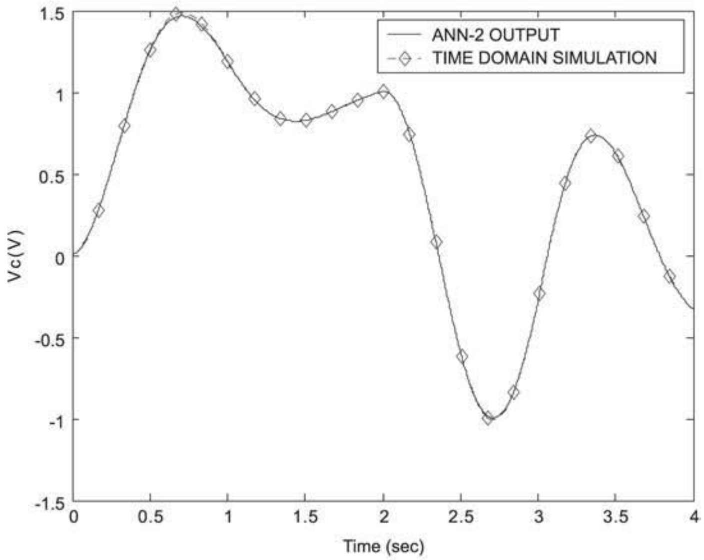
(a)



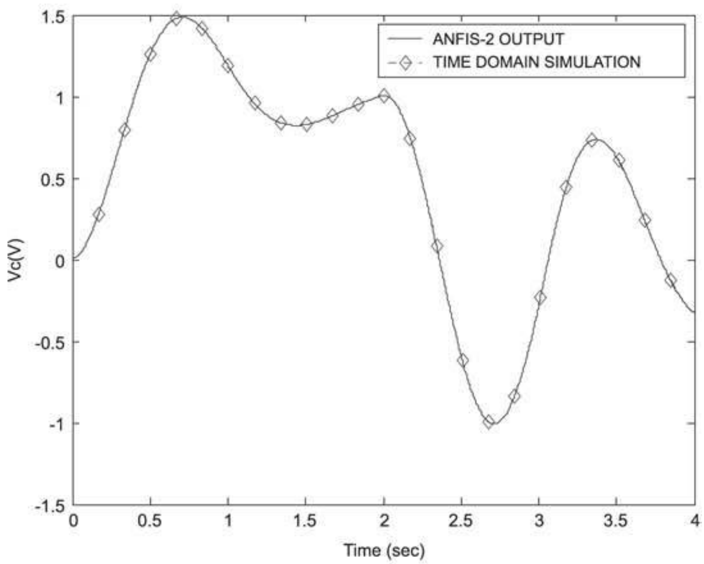
(b)

Figure 8.
The output of models relation to V_c and the circuit output for unit step input of circuit in Figure 3, (a) the ANN-1 model, (b) the ANFIS-1 model, (c) the ANN-2 model, and (d) the ANFIS-2 model

(*Continued*)

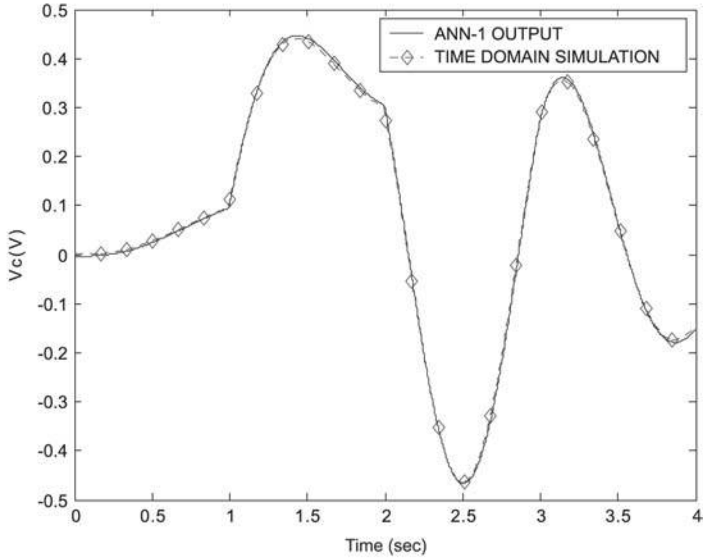

ANN-2 OUTPUT
TIME DOMAIN SIMULATION
Vc(V)
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
2
2.5
3
3.5
4

(c)

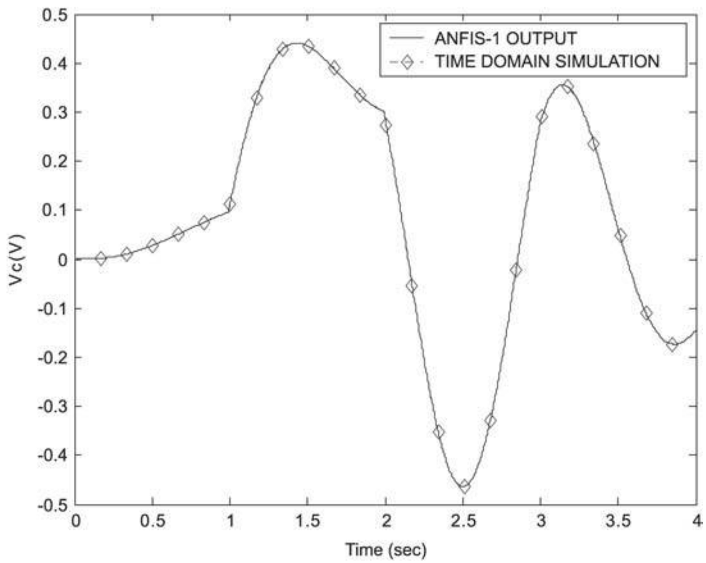

ANFIS-2 OUTPUT
TIME DOMAIN SIMULATION
Vc(V)
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
2
2.5
3
3.5
4

(d)

Figure 8.



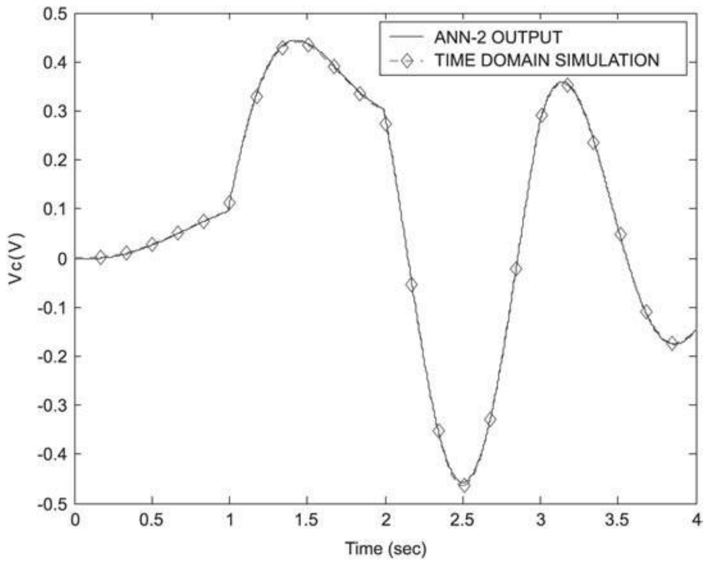
(a)



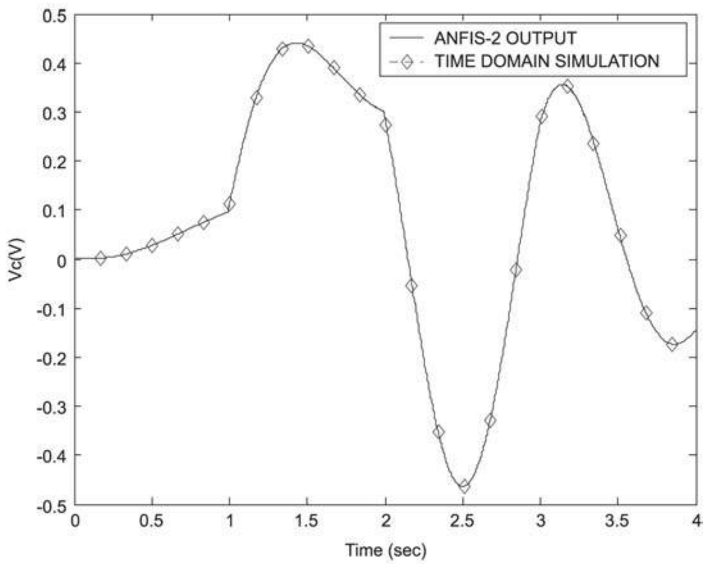
(b)

Figure 9.
The output of models relation to V_c and the circuit output for $V_i = \sin(0.1*t)$ input of circuit in Figure 3, (a) the ANN-1 model, (b) the ANFIS-1 model, (c) the ANN-2 model, and (d) the ANFIS-2 model

(*Continued*)


ANN-2 OUTPUT
TIME DOMAIN SIMULATION
Vc(V)
Time (sec)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4

(c)


ANFIS-2 OUTPUT
TIME DOMAIN SIMULATION
Vc(V)
Time (sec)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4

(d)

Figure 9.

5. Conclusions

In this paper, the modelling of switched circuits by ANN and ANFIS is discussed. It was found that both ANN and ANFIS produced reasonable results for modelling of switched circuits. The best performances, both in training and in validation modes, were also obtained with the ANFIS models that use grid partition and clustering algorithm to generate the rules and membership functions. It was seen that the ANFIS-2 gives better results than the other models in that the number of parameter, the training time, and the arrived errors are smaller.

In addition, the architectural simplicity of the ANFIS-2 as compared to others approaches is highlighted in tables. This illustrates that the ANFIS-2 significantly reduces the number of parameters required in the architecture with respect to comparable other network architectures (which can be interpreted as the increase in transparency removing the inherent redundant information in other network architectures). This has obvious benefits for the hardware realization of the architecture. The possibility that this architecture could be implemented in hardware would have considerable advantage as it would reduce the long times associated with this application.

An important practical benefit of neural and fuzzy neural models is that their speed of execution is faster than that of the mathematical model obtained by MATLAB. The price for this impressive speed is a relatively small amount of error. Decreasing of computational time increases the feasibility of essentially switched circuit design through the use of personal computers.

When nonlinear components in a nonlinear circuit are modelled by linear components and ideal switches, the most important problem is the determination of the unknown switching sequence and switching time. Our study has been progressing to obtain the switching sequence and to compute unknown switching time by using the control inequalities in which a nonlinear circuit contains the internally controlled switches (such as conducting of a diode) and the externally controlled switches (such as the triggering of a thyristor).

References

Borchers, C. (1998), "Symbolic behavior model generation of nonlinear analog circuits", *IEEE Trans. on Circuits and Systems*, Vol. 45 No. 10, pp. 1362-71.

Cannas, B., Cincotti, S., Fanni, A., Marchesi, M., Pilo, F. and Usai, M. (1998), "Performance analysis of locally recurrent neural networks", *COMPEL*, Vol. 17 Nos 5/6, pp. 708-16.

Choi, Y.J. and Farrell, A.J. (2000), "Nonlinear adaptive control using networks of piecewise linear approximators", *IEEE Trans. on Neural Networks*, Vol. 11 No. 2, pp. 390-401.

Demir, Y., Koksal, M. and Herdem, S. (1996), "State space suitable modelling of power semiconductors with ideal switches, resistors, constant voltage and current sources", *Proc. IMACS 5. Int. Conf.*, Saint-Nazaire, France, Vol. 1/3, pp. 97-102.

Djukanovic, M.B., Calovic, M.S., Vesovic, B. and Sobajic, D.J. (1997), "Neuro-fuzzy controller of low head hydropower plants using adaptive-network based fuzzy inference system", *IEEE Trans. on Energy Conversion*, Vol. 12 No. 4, pp. 375-81.

Efe, M.O., Kaynak, O. and Rudas, I.J. (1999a), "Increasing the noise rejection capability of neural networks in identification of nonlinear systems", *Int. Conf. on Mechatronic Technology (ICMT'99)*, 21-23 October, Pusan, Korea, pp. 363-8.

Efe, M.O., Kaynak, O. and Wilamowski, B.M. (1999b), "Training of fuzzy inference systems by combining variable structure systems technique and Levenberg-Marquardt algorithm", *Proc. of the 25th Annual Conference of the IEEE Industrial Electronics Society (IECON'99)*, 29 November - 3 December, California, USA, Vol. 2, pp. 514-19.

Ellithy, K. and Al-Naamany, A. (2000), "A hybrid neuro-fuzzy static var compensator stabilizer for power system damping improvement in the presence of load parameters uncertainty", *Electric Power Systems Research*, Vol. 56, pp. 211-23.

Gomez-Skarmeta, A.F., Delgado, M. and Vila, M.A. (1999), "About the use of fuzzy clustering techniques for fuzzy model identification", *Fuzzy Sets and Systems*, Vol. 106, pp. 179-88.

Jang, J.R. and Chuen-Tsai, S. (1993), "ANFIS: adaptive neural fuzzy inference systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 23 No. 3, pp. 665-85.

Jang, J.R. and Chuen-Tsai, S. (1995), "Neuro-fuzzy modelling and control", *Proceedings of the IEEE*, Vol. 83 No. 3, pp. 378-406.

Jang, J.R., Mizutani, E. and Chuen-Tsai, S. (1997), "Neuro-fuzzy modelling and soft computing: a computional approach to learning and machine intelligence", *PTR*, Prentice-Hall.

Jantzen, J. (1998), "Neurofuzzy modelling", *Technical Report 98-H-874*, Department of Automation Technical University of Denmark.

Koksal, M. (1984), "A fast convergent method for the analysis of nonlinear systems with periodic excitations", *IEEE Int. Symp. on Circuit and Syst. Proc.*, Montreal, Canada, Vol. 3 No. 3, pp. 1362-464.

Nørgaard, M. (1997), "Neural network based systems identification for use with Matlab", *Technical Report 96-E-830*, Department of Automation, Technical University of Denmark.

Park, M., Ji, S., Kim, E. and Park, M. (1999), "A new approach to the identification of a fuzzy model", *Fuzzy Sets and Systems*, Vol. 104, pp. 169-81.

Rashid, K., Ramirez, J.A. and Freeman, E.M. (2000), "Derivative extraction from a neuro-fuzzy model", *COMPEL*, Vol. 19 No. 3, pp. 850-65.

Zawirski, K., Ski, K.U. and Ferenc, J. (1999), "Fuzzy control of current in thyristor DC drive", *COMPEL*, Vol. 18 No. 4, pp. 675-83.

This article has been cited by:

1. Emrehan Yavşan, Ayşegül Uçar. 2016. Gesture imitation and recognition using Kinect sensor and extreme learning machines. *Measurement* **94**, 852-861. [CrossRef]
2. Ayşegül Uçar, Yakup Demir, Cüneyt Güzeliş. 2016. A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering. *Neural Computing and Applications* **27**:1, 131-142. [CrossRef]
3. Emrehan Yavsan, Aysegul UcarTeaching human gestures to humanoid robots by using Kinect sensor 1208-1211. [CrossRef]
4. Davut Hanbay, Ibrahim Turkoglu, Yakup Demir. 2010. Modeling switched circuits based on wavelet decomposition and neural networks. *Journal of the Franklin Institute* **347**:3, 607-617. [CrossRef]
5. REMZI TUNTAŞ, YAKUP DEMIR, MUHAMMET KÖKSAL. 2008. A NEW APPROACH FOR THE ANALYSIS OF THE TRANSIENT AND STEADY STATE OF PIECEWISE-LINEAR CIRCUITS BY NEURAL NETWORKS. *Journal of Circuits, Systems and Computers* **17**:06, 1091-1109. [CrossRef]
6. Brad S. Trinkle. 2005. Forecasting annual excess stock returns via an adaptive network-based fuzzy inference system. *Intelligent Systems in Accounting, Finance and Management* **13**:3, 165-177. [CrossRef]