Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 5 (7): 1232-1236, 2005
ISSN 1812-5654

Modeling of Switching Systems with Artificial Neural Networks

Remzi Tuntaş and Yakup Hameş
Ercis Vocational Collage, 65400, Yil University, Ercis, Van, Turkey

Abstract: In this study, for the first time, an alternative method was presented intended for analysis of circuits including switches. The aim of in this study was to reduce simulating time and to easily investigate real physical systems in computer media by using this modeling in place of the circuits is used in real applications. In recent years, the analysis has gained importance. Here a method is presented to analysis of switching systems by using the Artificial Neural Networks. In this method, forward ANN which is convenient to analyse of switching systems has been applied by use of back propagation algorithm. The method is illustrated with examples and the output of the computer program called YPMA is presented.

Key words: Systems modeling, switching systems, FNN

INTRODUCTION

Nonlinear elements are being used on the circuits for a long time and nowadays many other new nonlinear electronic and power electronic circuit elements started to be used. These nonlinear elements some special properties such as quick switching, working quietly, being controllable with small amount of energy and having small size, long lifetime, low cost place them among the most significant circuit elements. Therefore, the analyses of the circuits that contain these nonlinear components become more important. Many methods have been developed for these analyses and one of them is piecewise linearity approach[1,2]. This approach considers nonlinear elements as piecewise linear elements thus the circuits are modeled by replacing nonlinear components with equivalent ones that contain linear elements, power sources and switches[3]. In recent years this approach becomes more important. Consequently, switches become one of the most used components on circuits.

The number of studies modeling nonlinear systems by using Artificial Neural Networks (ANN) has increased in the recent years[4-7]. ANN has been used very frequently in nonlinear modeling due to its parallel processing, generalization and learning abilities. For this purpose feedforward and feedback multilayer ANN is used. In order to train these networks dynamic and static back propagation learning algorithms have been developed[8,9].

In this study, switching circuits are modeled by using ANN. Studies have been conducted in order to determine the learning and generalization abilities of ANN. In addition, a software program called YPAM is developed[10]. The result of the analysis done with this program on sample circuits is given.

MATERIALS AND METHODS

Artificial neural networks: For years many researchers focus on conducting studies on intelligent systems that mimics human behaviours such as ANN, a data processing system. Generally, ANN can be considered as a black box which gets input and produce output[11]. ANN is composed of processing elements and weighted connections. Furthermore, thresholds function and input/output layers are also among the key elements for ANN's design, application and utilization.

Feedforward Multilayered Neural Networks (FNNs): In this paper, the multilayer FNNs proven rigorously to be a universal function aproximator was used. Given sufficient input-output data [x, y], FNN was determined by the well-known backpropagation algorithm as if the objective

$$E(k) = \frac{1}{2}\sum_{j=1}^{m}\left(d_j - y_j\right)^2 \qquad K=1, 2, \ldots T \qquad (1)$$

where, $d \in \Re^m$ and $y \in \Re^m$ represent desired and network outputs.

The network is then used for feedforward computation with various inputs. Such training of the network is normally depicted by the block diagram (Fig.1). Generally, the inputs for the FNN models are from the input of the nonlinear systems through the tapped-delay line block, the others are from the output of the nonlinear circuit through another the tapped-delay line block (Fig. 2). In this study the output of FNN

Corresponding Author: Dr. Remzi Tuntaş, Yuzuncu Yil University, Ercis Vocational Collage, 65400, Ercis, Van, Turkey
Tel: 904323882167 Email: rtuntas@hotmail.com

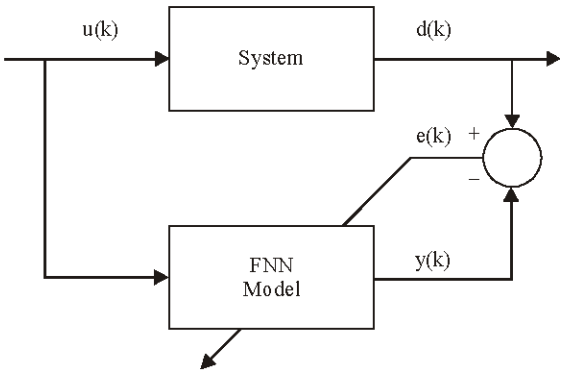

Fig. 1: Training of neural network

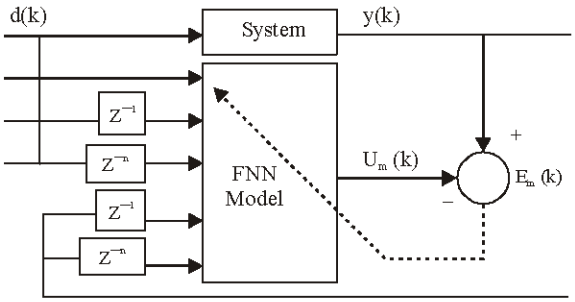

Fig. 2: Modelling structure for nonlinear systems

structure for modelling each system state was taken as the output vector belonging to the state out of the output vectors used in the control equations while its inputs are reference signal, its delayed values and the delayed output values[12]. In addition, architecture selection is completed choosing both the appropriate number of hidden units and the connections by authors' experience.

The schematic diagram of the internal structure of the neural network proposed for modelling the system states is shown in Fig. 3. In each system state, it is used same network structure consisting of the input layer, hidden layers and output layer, each having a number of units, depicted as circles. Each unit is connected to units in the neighbouring layer with a weight, shown as a line in the Fig. 3. The actual neural network is thus parametrized by a set of weights.

Feedforward neural networks training: In this study an FNN trained to mimic the real system was used. The trained system gives an output to the given input. This output was compared with the real system performance and according to the error between them; the trained system was adjusted till it represents the real system accurately. Network entry was composed of system's separated time equation and the input function that is applied to the system. Therefore, the number of the cells

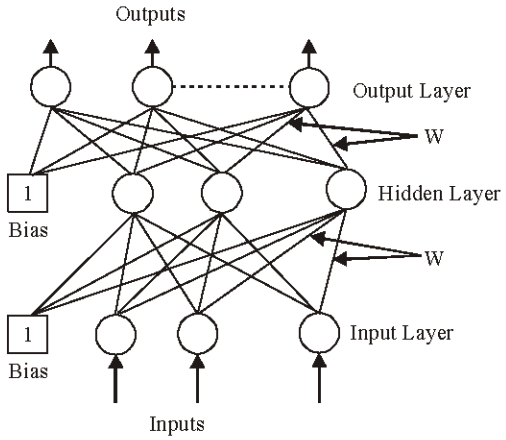

Fig. 3: Multilayer feedforward neural network

at the system entry must be selected according to system state. If the number of the cells at the input layer are M and the number of the cells at the output layer are N then the number of the cells at the hidden layers are selected as $N(M+1)$[13].

Network training was executed by using error back prohibition. If output's sum of square of error is E , desired output level is d_j and the real system output is y, then the kth sample output error and training signal like the following.

$$e_j(k) = d_j(k)-y_j(k) \qquad j=1,2....m \qquad (2)$$

$$E(k) = \frac{1}{2}\sum_{j=1}^{m} e_j^2(k) \qquad K=1, 2, N \qquad (3)$$

The weight between the input layer and hidden layer is W^1_{il} and the weight between the hidden layer and output layer W^2_{jil} input vector is x, hidden layer`s vector is v, the baises are θ1 and θ2, The transfer activation function Φ(.). This multilayer neural network has three layers. The equations between the layers of the network like the following.

$$net_i = \sum_{l=0}^{n} w^1_{il} x_1 \qquad 1=1,....n \qquad (4)$$

$$v_i = \Phi_i(net_i) \qquad i=1,.... .p \qquad (5)$$

$$y_j = \sum_{i=0}^{p} w^2_{ji} v_1 \qquad j=1,..... k \qquad (6)$$

In order to minimize training signal according to weights, it is necessary to determine the training. Therefore, determination of weights at hidden layers for each kth sample is given in Eq. 7 and 8 and the input layer`s weights are determined in Eq. 9 th and 10 th.

$$w^2_{ji}(k) = w^2_{ji}(k-1)+\Delta w^2_{ji}(k) \qquad (7)$$

$$\Delta w_{ji}^{2}(k) = -\alpha\frac{\alpha E_j(k)}{\alpha w_{ji}^{2}(k)} + \beta\Delta w_{ji}^{2}(k-1) \quad (8)$$

$$w_{i1}^{1}(k) = w_{i1}^{1}(k-1) + \Delta w_{i1}^{1}(k) \quad (9)$$

$$\Delta w_{i1}^{1}(k) = -\alpha\frac{\partial E_j(k)}{\partial w_{i1}^{1}(k)} + \beta\Delta w_{i1}^{1}(k-1) \quad (10)$$

If the number of the iterations for weight determination is S and N samples is prepared, then the determination of the collected weight can be determined by the following equation:

$$\Delta w(S) = -\alpha\frac{\partial E}{\partial w(S)} + \beta\Delta w(S-1) \quad (11)$$

$$w(S) = w(s-1) + \Delta w(S) \quad (12)$$

For training the adaptive weight vector, iterative training vector is used. According to weights the real gradient of training signal can be determined via collected training[14].

Artificial neural networks selection and simulation to model a selected system: Let the continuous time transfer of a chosen second degree system be

$$G(s) = \frac{5}{S^2+3S+6} \quad (13)$$

using the related commands in MATLAB we obtain the discrete time transfer function as follows:

$$G(z) = \frac{0.0107+0.0215z^{-1}+0.0107z^{-2}}{1-1.6910z^{-1}+0.7425z^{-2}} \quad (14)$$

since discrete time systems are modeled by difference methods, the input-output correlation of discerete time transfer function is:

$$y(k) = 0.0107u(k)+0.0215u(k-1)+0.0107u(k-2) +1.6910y(k-1)-0.7425y(k-2) \quad (15)$$

by this way, the number of the cells in the input layer is calculated as 5. Moreover we add a bias term with value 1, so the total number of input layers is 6. To calculate the number of cells in the hidden layer, we generally add 1 to the number of cells in the input layer which is 7. Since the system is one-input one output system, there is only one cell in the output layer of the network. The momentum coefficient and training rate which are used in FNN training are adjusted inside the network model adaptively[15]. For the simulation of FNN, it is trained by YPMA software which is prepared in TC++ software language with a total training algorithm. The training ends when the error value is minimal.

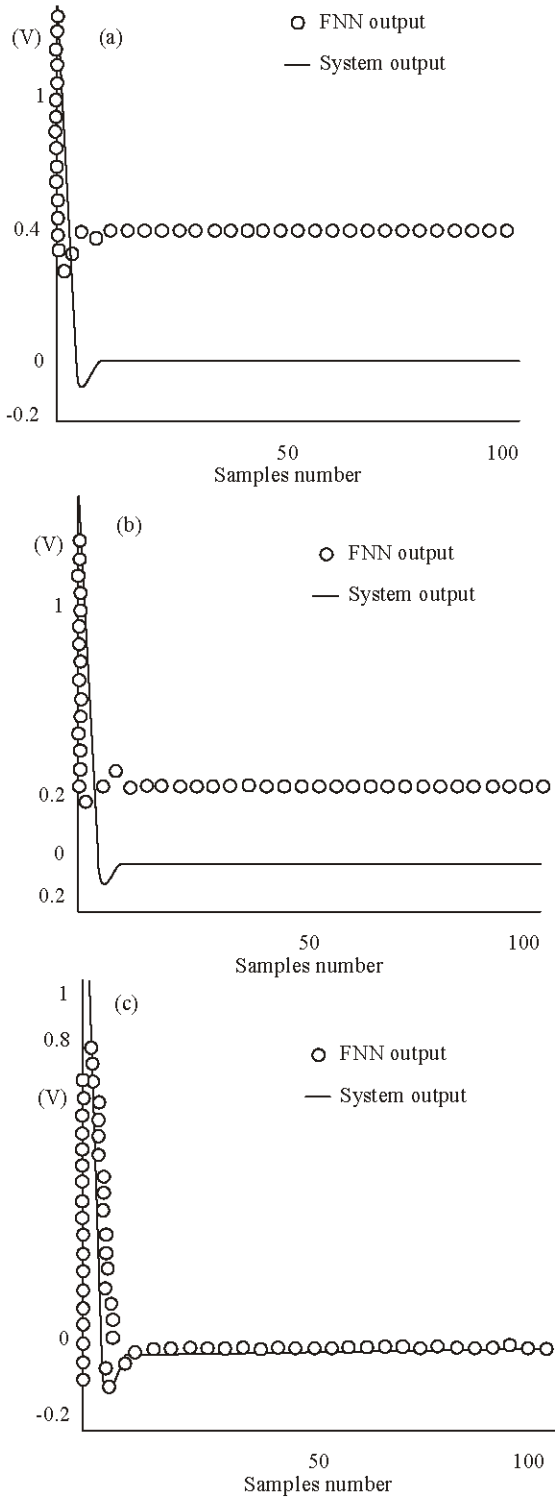

Fig. 4: The response of the system and network model trained in Eq. 13 to the sudden impact reference input sign for 100 samples a) with error E = 1 b) with error E = 0.5 c) with error E=0.0097

E gets smaller the training of FNN becomes more satisfactory (Fig. 4). The training is ended when the most accurate output is obtained at the least E-level, which is 0.0097-error. From now on, the network model represents the system successfully. In the following iterations the network model will not be trained and the weight-values which are obtained from the last training will be used.

After the FNN which is used for the modeling of the system, was trained with the weighted-value that is mentioned above. The ability of the system modeling of the network model, which is trained according to sudden impact input sign will be observed in order to examine generalization ability which is the most important quality of FNN. For this purpose, the network model output can be model the system totally with unit steps and sinusoidal input signals which are shown in Fig. 5a and b.

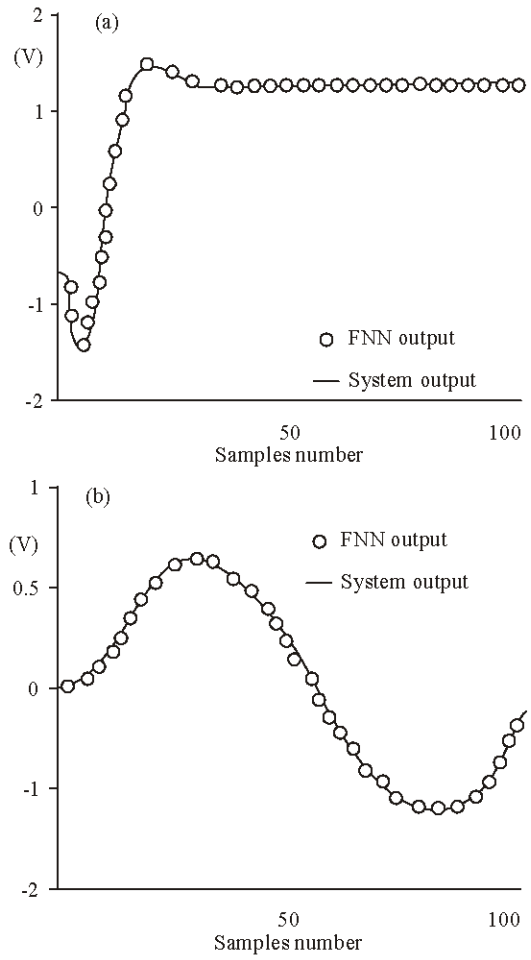

Fig. 5: The response of the system in Eq. 13 and the network model trained with E = 0.0097 error. a) to the unit-step, b) to the sinusoidal reference input signal for 100 samples.

RESULTS AND DISCUSSION

As a result of piecewise linear modeling approach to the nonlinear circuit model, switches occurs in equivalent circuits. As a result of different combination of switches, each system model which refers to a linear region is solved by YPMA software. The outputs of all linear regions combined in MATLAB to simulate the system.

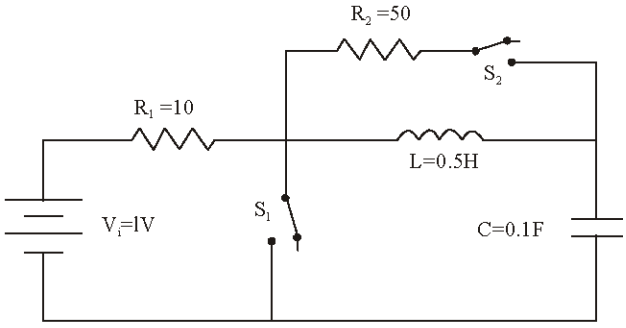

Fig. 6: Sample circuit

In this example the circuit in Fig. 6 is analyzed. In the first phase s1-s2 are on, in the 2 nd phase s1-s2 are off, in the 3 rd phase s1 is on s2 is off and in the last phase s1 is off and s2 is on. System stays for 1 sec at each phase. The system and network model output related to Vc of the circuits are obtained as in Fig. 7.

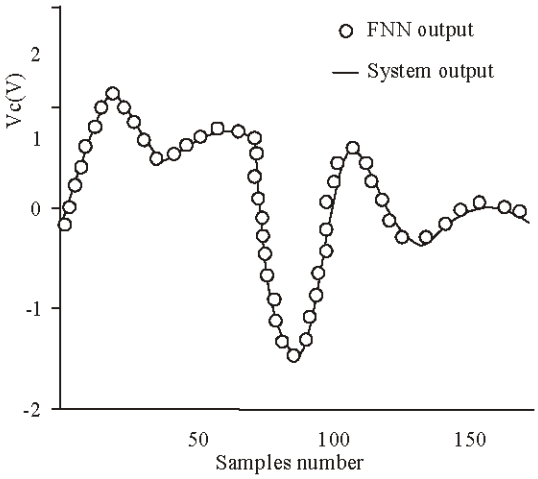

Fig. 7: System network model output related to Vc of the circuit in Fig. 6

CONCLUSIONS

In this study we obtained transfer functions related to system state of FNN and switching circuits. Transfer functions are coded proper to the MATLAB programming language commands and transferred to the difference functions used in FNN to train the network model. The trained network model has the ability to generalize and adapt to the several parameter changes in the system by using the experience it gets.

We can use this generalization property on different input applications of systems with known mathematical models to obtain output signs easily. Excellent agreement was obtained between the FNN outputs and the system output as seems at the above graphs.

With this method, in spite of several hard physical systems in real applications in terms of energy, cost, speed and dimension, it was constituted the models of these systems which can represent these systems correctly. These models are provided calculation efficient advantage for simulating time with eliminating the effect of environmental factor.

REFERENCES

1. Billing, S.A., H.B. Jamaluddin and S. Chen, 1992. Properties of neeural networks with aplication to modelling nonlineer systems. Intl. J. Control, pp: 193-224.
2. Cinquin, O. and J. Demongeot, 2005. High dimensional switches and the modelling of cellular differentiation. J. Theoretical Biol., 233: 391-411.
3. Demir, Y., 1993. The existing of state equations of nonlinear circuits in different region and the calculating of switching equations for trasition interregions. Ph.D. Thesis, Firat University.
4. Garea, A., J.A. Marques and A. Irabien, 2005. Mechanistical and non-linear modelling approaches to in-duct desulfurization. Chem. Eng. Proc., 44: 709-715.
5. Torrecilla, J.S., L. Otero and P.D. Sanz, 2005. Artificial neural networks a promising tool to design and optimize high-pressure food processes. J. Food Eng., 69: 299-306.
6. Lars, S. and O. Klaus, 2003. Modeling the adaptive visual system. A Survey of principled approaches. Neural Networks, 16: 1353-1371
7. Verikas, A. and M. Bacauskiene, 2003. Using artificial neural networks for process and system modelling. Chemometrics and Intelligent Laboratory Systems, 67: 187-191
8. Chen, Y. T.C., Huang and R.C., Hwang, 2004. An effective learning of neural network by using RFBP learning algorithm. Information Sciences, 167: 77-86.
9. Tokuda, I., R. Tokunaga and K. Aihara, 2003. Back-propagation learning of infinite-dimensional dynamical systems. Neural Networks, 16: 1179-1193
10. Tuntaş, R., 1999. Modeling of nonlinear devices with artificial neural networks. M.Sc. Thesis, Firat University.
11. Panerai, R.B., M. Chacon, R. Pereira and D.H. Evans, 2004. Neural network modelling of dynamic cerebral autoregulation: Assessment and comparison with established methods. Med. Eng. Phy., 26: 43-52.
12. Tuntaş, R., A. Uçar and Y. Demir, 2004. A new approach for the modelling and simulation of piecewise-linear circuits by neural networks. ASYU-INISTA, pp: 108-111.
13. Zhang, Y., P. Sen and G.E., Hearn, 1995. An online trained adaptive neural controller. IEEE Control Systems, pp: 67-75.
14. Mellit, A., M. Benghanem, A.H. Arab and A. Guessoum, 2005. An adaptive artificial neural network model for sizing stand-alone photovoltaic systems: Application for isolated sites in Algeria. Renewable Energy, 30: 1501-1524.
15. Diao, Y. and K.M. Passino, 2002. Immunity-based hybrid learning methods for approximator structure and parameter adjustment. Engineering Applications of Artificial Intelligence, 15: 587-600.