International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 4, No. 1
February 2015

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 4, No. 1, February 2015

Research Paper

A HARDWARE REALIZATION OF MULTIPOINT ELECTRONIC DEVICE USING FPGA AND NEURAL NETWORK

Shefa A Dawwd[1] and Ula F Ahmed[2]*

*Corresponding Author: ***Shefa A Dawwd*** ✉ shefadawwd@gmail.com

Neural Networks (NN) play a crucial rule in all intelligent system design, as its ability to learn and mimic brain behavior. Many applications use Neural Networks in their implementations such as identify, simulate and control nonlinear system. In this work, a neural network mapped to reconfigurable platform Field Programmable Gate Array (FPGA) to impersonator first order system (RC) circuit. Due to the inability to predict output of electronics nonlinear mathematical way so used feedback system to calculate the exact output of any electronic device. Elman Neural Network is recurrent neural networks with special feedback from output layer to hidden layer, it is the best candidate for this kind of application as it is ability to remember the previous result. Elman Neural Network has a limited memory to store the previous output. Neural Networks in general contain three main layers: input layer, hidden layer (in between) and output layer. In Elman Neural Network hidden layer contains a feed back to reserve the old output of this layer. In this paper, Elman Neural Network has been used with four hidden layer and multi neurons in each hidden layer. The input layer has three inputs and the output layer has one outputs. Multi-points of the electronic device are selected to act as the input of analog to digital (A/D) converter, and then to the network input layer, the electronic device is connected to neural network during the training phase. The neural network training by matlab, c^{++} language, c under zynq. After the training, the neural network will replace the functionality of electronic device. The resulted (trained) neural network saved in DDR (RAM) and mapped to hardware using FPGA by using zedboard.

Keywords: Neural Networks, Elman Neural Network, Zedboard, FPGA, Tansig function

INTRODUCTION

Artificial Networks (ANNs) is a mathematical method designed to simulate the way in which the human brain a specific task by a huge processing distributed parallelism, and made up of simple processing elements, these units are only computational elements called neurons or contract which has a nervous property, from where they store experimental practical knowledge and information to make it available to the user and

[1] Assistant Professor, University of Mosul, Al-Majmoa'a Street, Mosul, Iraq.
[2] Lecturer, University of Mosul, Al-Majmoa'a Street, Mosul, Iraq.

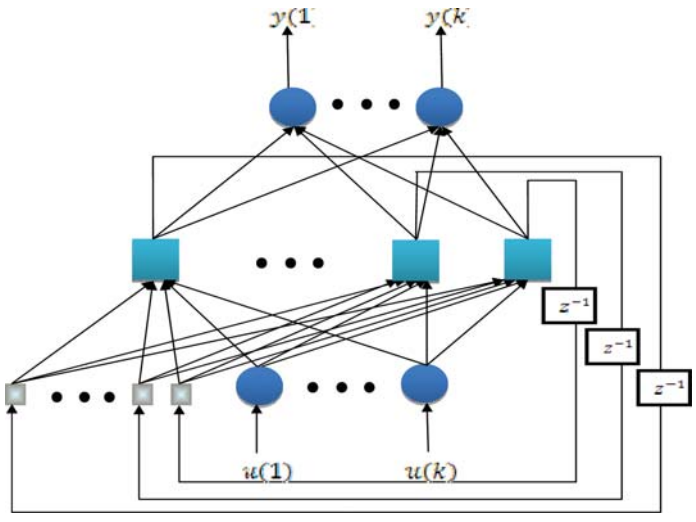
by adjusting the weights (Suzuki, 2011). Simple mathematical operation 'to represent ANN is : multiplication, summation and activation .this operation worked as: the inputs of layer multiplied by weighted that means every input has individual weight. In the second step done by summation operation that mean sums all result of first step with bias, the result of second step is passing through activation (Nielson, 2011). The first successful attempt to represent the nerve cell artificially had in 1943 where it was made up of two layers input layer and the rest of the cell and fired name on this cell acronym for the names of scientists inventors (MCP) (Warren S McCulloch and Walter Pitts) (Mc Culloch). Elman Neural Networks or can be called a simple recurrent neural network invented by Jeffry Elman in 1990, it was a small network content from (1 node in input layer, 2 node in hidden layer, 2 node in context layer,1node in output layer) to represent XOR gate (Elman, 1990). In 1993 (D T Pham) used the Elman neural network and modified Elman neural network to recognize the linear and non-linear system by using C language in IBM-386 processor (D T Pham, 1992) in 2006 (A Kalinli) was devise a new type of neural network by make a company between the Elman neural network with NARX network to make Elman Network with Embedded Memory and applied it to recognize linear and non-linear system (Kalinli, 2006). In 2009 F J Linto control Linear Ultrasonic Motor and implementation it in FPGA type (XC2V1000) (Lin *et al.,* 2009).

ELMAN NEURAL NETWORKS AND MODIFICIT ELMAN NEURAL NETWORKS

Basic Elman Networks

Elman Neural Networks contain four layer: (input layer, hidden layer, context layer, output layer), input layer acts as a buffer layer pass the extern inputs to hidden layer without any changes, hidden layer contain (multiplier, summation, transfer function), while the context layer acts as memories of hidden layer to save the previous value of hidden layer without any change .the output layer have the same contain of hidden layer. The result of output layer is the output of neural network to external environment. The

Figure 1:Elman Neural Network



structure of Elman neural network can represent in Figure 1 (Pham, 1992).

From the Figure 1 can noticed the external input (u) passed through input layer to hidden layer ,in the hidden layer the input multiply by their weight and the same thing about context unit , after that sum all result of multipliers and pass the result to transfer function the transfer function in hidden layer usually acts as non-linear function after that pass the result to output layer. In the output layer multiply the input (passed from hidden layer) by their weight and passed the result to transfer function after summation it. The transfer function in output layer usually can be linear. Can obtain this by following equation:

$$\overset{\backslash}{x}_i(k) = \sum_{j=1}^{m} u(k) * w_i^u(k-1) + x_{i,j}^c(k) * w_{i,j}^c(k-1) \quad ...(1)$$

$$x_i(k) = f\left(\overset{\backslash}{x}_i(k)\right) \quad ...(2)$$

$$x_{ci}(k) = x_i(k-1) \quad ...(3)$$

$$\overset{\backslash}{y}_i(k) = \sum_{i=1}^{n} x_i(k) * w_i^y(k-1) \quad ...(4)$$

$$y(k) = g\left(\overset{\backslash}{y}_i(k)\right) \quad ...(5)$$

where

u: Input vector.

w_i^u : Input's weight.

$w_{i,j}^c$: Hidden's weight.

w_i^y : Output's weight.

x_i : Hidden layer's output.

f: Transfer function of hidden layer.

x_{ci} : Context layer's value.

g: Transfer function of output layer.

y: Output vector.

Modified Elman Networks

The structure of Modified Elman Neural Network is the same as Elman Neural Network with self-connection in context layer, the self-feedback increase dynamic memorization of the network, can be depicted in Figure 2 can noticed the last value of context layer effected to current value of context layer with and summation with the value of hidden layer.

Can obtain this by following equation:

$$\overset{\backslash}{x}_i(k) = \sum_{j=1}^{m} u(k) * w_i^u(k-1) + x_{i,j}^c(k) * w_{i,j}^c(k-1) \quad ...(6)$$

$$x_i(k) = f\left(\overset{\backslash}{x}_i(k)\right) \quad ...(7)$$

$$x_{ci}(k+1) = x_i(k) + wc * x_i^c(k) \quad ...(8)$$

$$\overset{\backslash}{y}_i(k) = \sum_{i=1}^{n} x_i(k) * w_i^y(k-1) \quad ...(9)$$

$$y(k) = g\left(\overset{\backslash}{y}_i(k)\right) \quad ...(10)$$

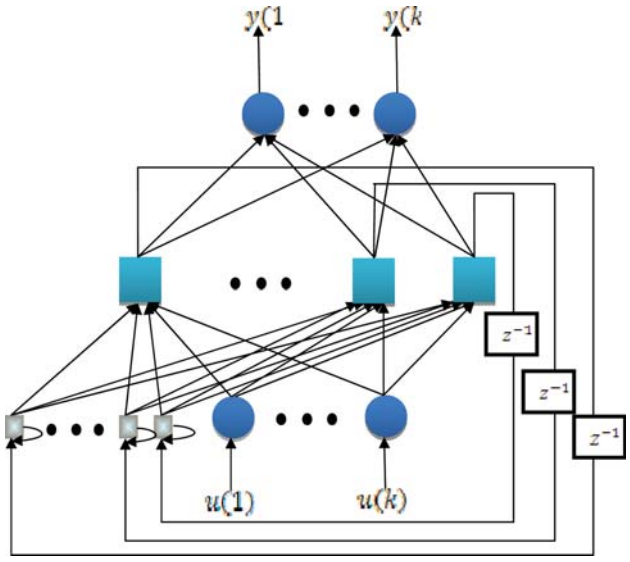
where

wc: context's weight constant.

Dynamic Backpropagation Algorithm

Backpropagation algorithm is method of training Artificial Neural Networks was invented by (Rumelhart and McClelland, 1986) for feed-forward neural networks and develop to applied to content recurrent neural network by Pineda (1989),

Figure 2: Modified Elman Networks



Dynamic backpropagation algorithm used to train Modified Elman neural network based on the Flowchart 1, this work done by chosen a random value of weight and change is depended on minimum square error rate and that change weight depend on following Equation (Pham, 1996):

$$\Delta w_i^y = -y_d((k)-y(k))x_i(k)$$

$$\Delta w_i^u = -y_d((k)-y(k))w_i^y \frac{df}{d\,x_i'(k)} x_i(k-1)$$

$$\Delta w_i^c = -(y_d(k)-y(k))w_i^y \frac{df}{d\,x_i'(k)} x_i(k-1)$$

$$w_{new} = w_{old} + \alpha\Delta w$$

After calculating the new value of weight we test the network that happen by applying the new weight and calculate the square error rate if the result error lese or equal the target then the training is finish or continuously the operation until to reach the target.

THE PROPOSED SYSTEM

In this paper used RC circuit to representation first order system and RLC circuit to representation a second order system to impersonation these system, used MENN with one node in the input layer, four nodes in hidden and context layer and one node in the output layer and used tansig (Hyperbolic tangent sigmoid transfer) function as a transfer function in the hidden layer and hardlim function in the output layer.

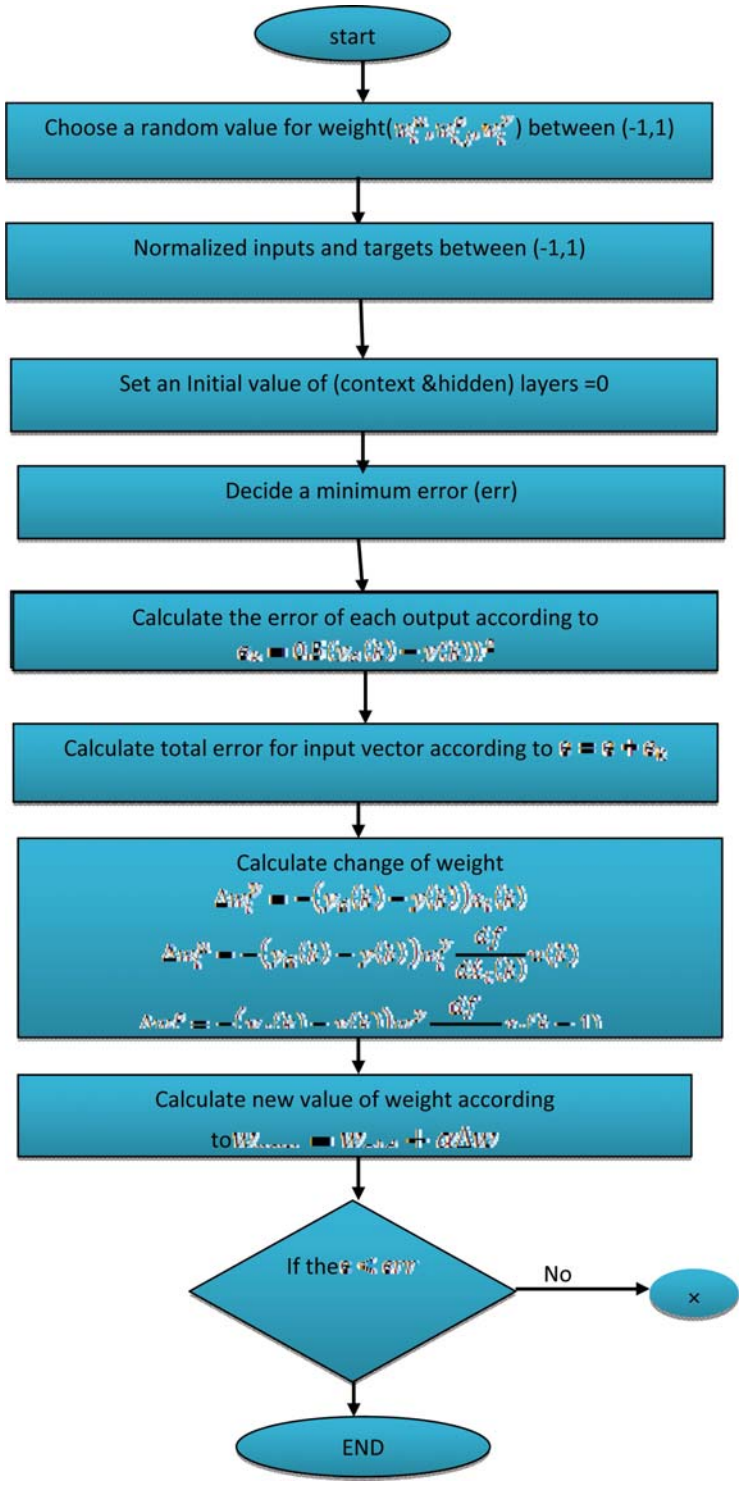
Training Method

In this paper, training algorithm had been done in three different way (matlab, c++, c under ZYNQ).

Usually training of neural networks done offline, it mean training the network in software and take the result of weight's value and put it in the hardware that represent the neural network. The work in neural network divide in two phase: training phase and running phase.

In training phase represent the number by floating point to compute a exactly weight this mean in matlab the number represent by 64 bit, in c++ represent by 32 bits and the same thing c

Chart 1: Dynamic Backpropagation algorithm



under zynq, while in running phase used fixed point to represent number 16 bits because the mathematic computation doesn't need to high precision.

Implementation Modified Elman Neural Network in FPGA

In general Elman neural network contains (multiplier, adder and transfer function). In this work the input layer works as a buffer to pass the input to hidden layer, in the hidden layer ,there are 20 multipliers (5 multipliers in each nodes), 16 adders (4 adder in each nodes), in the context layer contains 4 multiplier (one for each nodes) and 4 adder (one for each nodes) and finally in the output layer there are 4 multipliers, 3 adders and this can represent it in Figure 3.

Implementation Platform

Zedboard is used in the evaluation and development of programs, depending on the processor (Xilinx ZynqTM-7000 All Programmable SoC) as it combines 85,000 logic gate system (Series-7 Programmable Logic cells) and dual-processor (Corex-A9).

The strength of (Zedboard) to contain the peripheral devices on board as well as the ability to expand this and make it an ideal platform for both novices and experienced designers (Pham, 1996).

The work in this board dived to three part ,firs part select the hardware needed to done work (select processor (processing_system7_0), select memory(DDR), select type of input output (Fixed_IO)), the second part built elman hardware in VHDL and export it as bloke, the last part write program in c language to make interface between hardware of the system as show in Figure 4.

Figure 3: Implementation Modified Elman Neural Network in FPGA

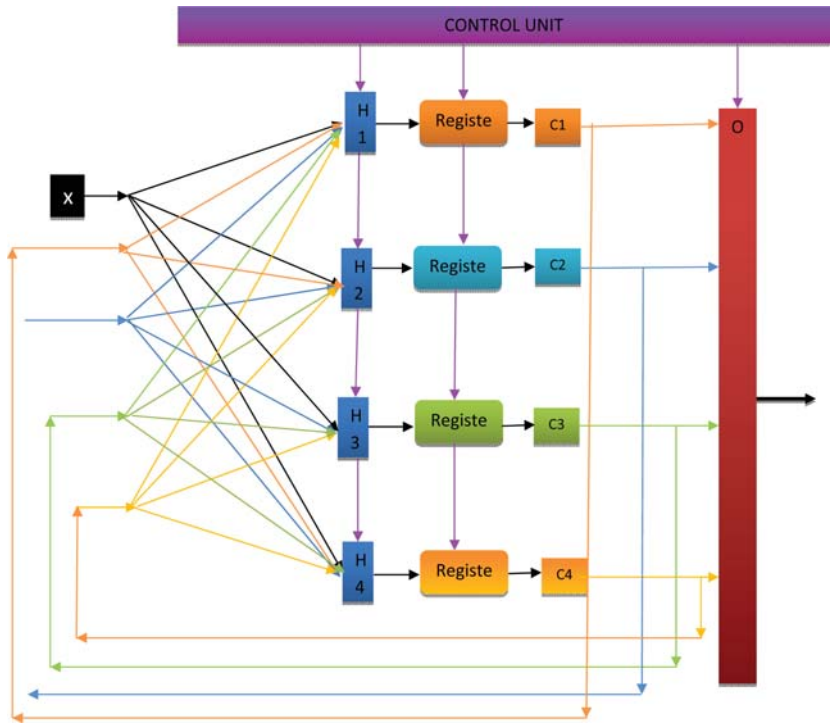


Figure 4: Implementation Platform

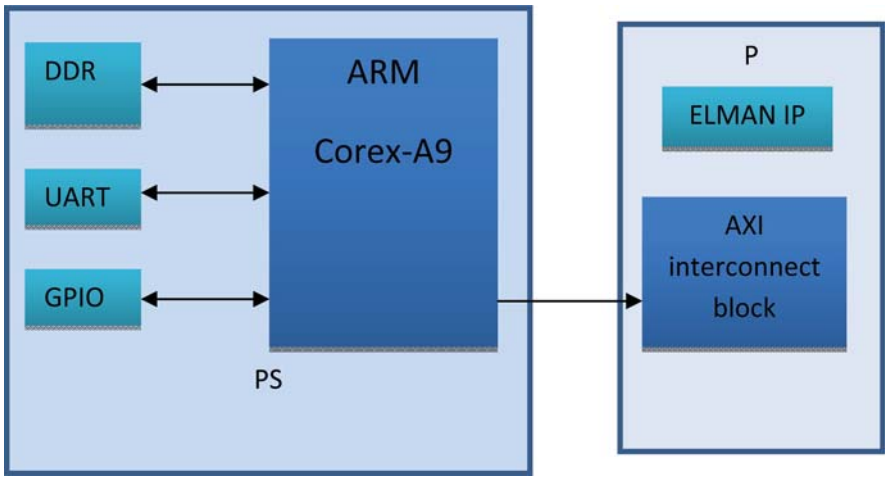


Figure 5: RC Circuit

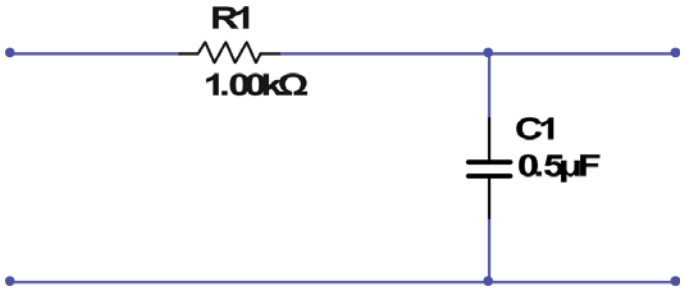
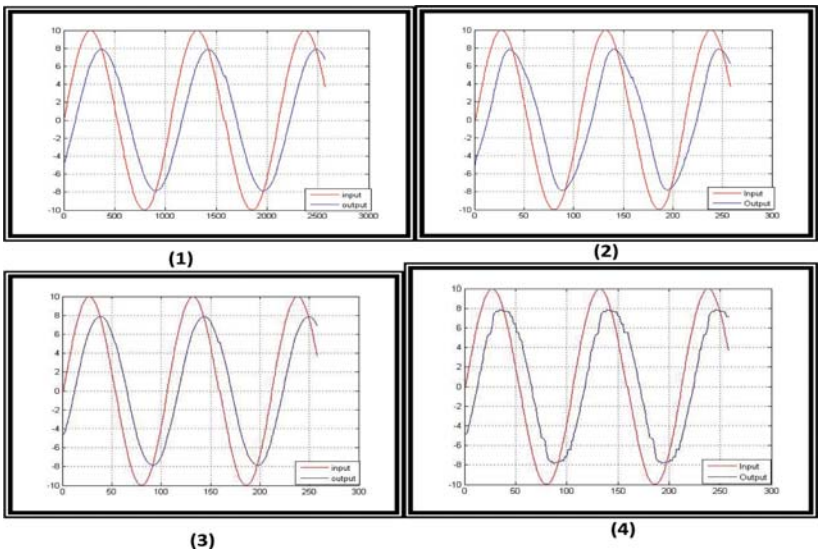
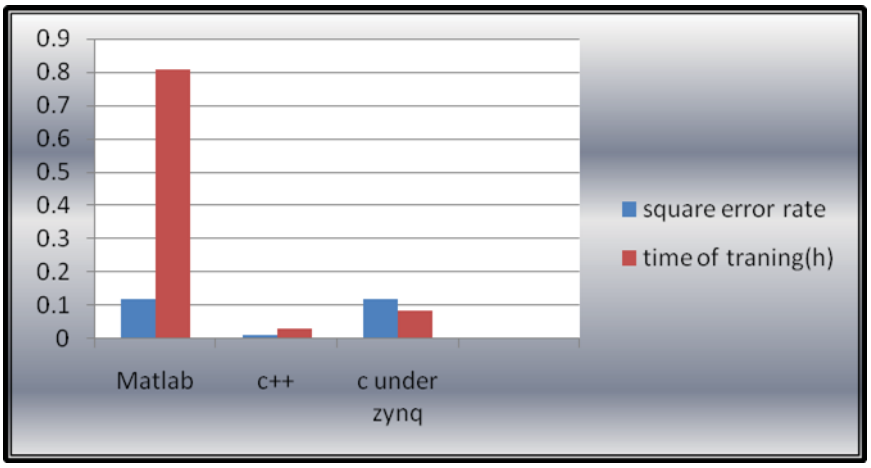


**Figure 6: The relationship between input/output to first order system :
1-the normal way, 2-after training in matlab, 3- after training in c++, 4- after training**



Flow Chart 2: Compare Between Three Way of Traning



RESULTS

In this work , the system applied is first order system (RC circuit) as shown in Figure 5, the system trained by matlab, C++ language, C language under ZYNQ, and tested it by ZedBoard and the result as shown in Figure 6.

CONCLUSION

From the result can see that the training in matlab takes more time compare with training in c++ and c under zynq that return to number of represent number in matlab is double than in c++ and c under zynq.

REFERENCES

1. Bailey T (2005), ”An Introduction to the C Programming Language and Software Design”.
2. Botros N M, and Abdul-Aziz M (1994), ”Hardware Implementation of an Artificial Neural Network Using Field programmable Gate Arrays (FPGA’s)”, *IEEE Transaction on Industrial Electronics,* Vol. 41.
3. Cernansky M and Benuskova L, ”Simple Recurrent Network Trained By Rtrl And Extended Kalman Filter Algorithms”, *Neural Network World,* Vol. 13, No. 3, pp. 223-234.
4. Coban R (2013), ”A context layered locally recurrent neural network for dynamic system identification“, *Engineering Applications of Artificial Intelligence*, Vol. 26, pp. 241–250.
5. Jamel T M and Khammas B M (2012), ” Implementation of a Sigmoid Activation Function For Neural Network Using FPGA“, Published in the 13th Scientific Conference of Al-Ma’moon University College -18 April 2012.
6. Elman J L (1990), “Find Structure In Time”, *Cognitive Science*, Vol. 14, pp. 179-211.
7. Gao X Z, Ovaska S J and Dote Y (2000), ”Motor Fault Detection Using Elman Neural Network with Genetic Algorithm-aided Training”, 0-7803-6583-6/00/$10.00 © IEEE.
8. Ghosh M (2007), ”Design And Implementation Of Different Multipliers Using VHDL“, Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela.

9. Houcque D (2005), "Introduction to Matlab for Engineering Students", Northwestern University, version 1.2, August.

10. "IEEE Standard for Binary Floating-Point Arithmetic", The Institute of Electrical and Electronics Engineers, Inc345 East 47th Street, New York, NY 10017, USA, 1985.

11. Pham D T and Liu X (1992)," Identification of linear and nonlinear dynamic systems using recurrent neural networks", *Artificial Intelligence in Engineering,* Vol. 8, pp. 67-75.

12. Kalinli A and Sairoglu S (2006), "Elman Network with Embedded Memory for System Identification", *Journal of information Science and Engineering*, Vol. 22, pp. 1555-1568.

13. Lewis F L, and Ge S S (2005), "Neural Networks in Feedback Control Systems", *Mechanical Engineer's Handbook*, John Wiley, New York.

14. Lotric U and Bulic P (2012), "Applicability of approximate multipliers in hardware neural networks", *SciVerse Science Direct*.

15. Li X, Moussa M and Areib S (2005), "Arithmetic formats for implementing artificial neural networks on FPGAs", *Canadian Journal on Electrical and Computer Engineering* (in print).

16. Lin F J, Hung Y C and Chen S Y (2009), "FPGA-Based Computed Force Control System Using Elman Neural Network for Linear Ultrasonic Motor", *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 4.

17. Mcculloch W and W H.Pitts (1943), "A Logical Calculus Of The Ideas Immanent In Nervous Activity", *Bulletin of mathematical Biophysics*, Vol. 5, pp. 115-133.

18. Nielsen F (2011), "Neural Networks – algorithms and applications".

19. Pham D T and Liu X (1996), "Training of Elman networks and dynamic system modelling", *International Journal of System Science*, Vol. 27, No. 2, pp. 221, 226.

20. Raghatate R P, Rajurkar S S, Badhe P U and Turale T L (2013), "Design and Implementation of Full Adder Using Vhdl and Its Verification in Analog Domain", *International Journal of Engineering Science Invention,* ISSN (Online), pp. 319 – 6734, ISSN (Print): pp. 2319 – 6726, Vol. 2, Issue 4, April.

21. Schoenauer T, Jahnke A, Roth U and Klar H (1998), "Digital Neurohardware :Principles and Perspectives", *Neuronal Networks in Applications - NN*'98 - Magdeburg.

22. Suzuki K (2011), "Artificial Neural Networks Methodological Advances And Biomedical Applications", Janeza Trdine 9, 51000 Rijeka, Croatia, Copyright © 2011 InTech.

23. ZedBoard (Zynq™Evaluation and Development) Hardware User's Guide", Version 2.2,27 January 2014.

24. •ilková J, Timko J and Girovský P (2006), "Nonlinear System Control Using Neural Networks", *Acta Polytechnica Hungarica*, Vol. 3, No. 4.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com