

Simulation Design of an Intelligent System for Automotive Transmission Gearbox Based on FPGA

Azzad Bader Saeed

Electrical Engineering Department
University of technology, Baghdad, Iraq
Email : 30065@uotechnology.edu.iq

Abstract

In this paper, a simulation of artificial intelligent system has been realized and then implemented in FPGA (Field Programmable Gate Array). This intelligent system acts as a controller for electrical types of Automotive Transmission Gearbox ATG, which is used to control five steps (1,2,3,4,5) of speed ratios of a vehicle. The first speed ratio (step 1) is characterized by highest torque and lowest velocity, while the fifth speed ratio (step 5) is characterized by lowest torque and highest velocity. The Back-propagation neural network has been used as an intelligent system for this work. The proposed neural network involves eight neurons in the input layer, five neurons in the hidden layer, and five neurons in the output layer. In this work, accurate results have been obtained due to the following reasons: 1) The MATLAB linear activation functions, 'Satlins' and 'Satlin' have been used for the hidden and output layers respectively, 2) The MATLAB training function 'Trainlm' (Levenberg-Marquardt training) has been used as a learning method for the proposed neural network, which is powerful algorithm. The Mean-Square-Error MSE of this work has reached to 3.94×10^{-25} in 15 iterations, which is acceptable value with fast training process. After implementing the proposed system in FPGA, the execution time has reached to 10 nsec, which is also an acceptable value. The proposed simulation system has been designed using MATLAB software package, and has been implemented in FPGA using ISE Design Suit software package.

Keywords: Artificial intelligent system, Back-propagation neural network, FPGA, Levenberg-Marquardt training, Automotive transmission gearbox.

1. INTRODUCTION

The transmission gearbox of a vehicle is defined as a system is used for transporting the mechanical output power of the engine to the driving wheels. It is always linked to the fly wheel at the back side of the engine. The duty of this device is reducing the rotation speed, at same time increasing

the torque of the crankshaft of the engine to a reasonable value that is proportional to the rotation speed and required torque of the driving wheels [1].

The transmission gearbox is usually composed from number of internal shafts, and several gears with different speed ratios, each one of these gears can convert the rotation speed from a specific value to another value. There are two main types of transmission gear: Manual, and automatic, whereas, in this work the automatic type has been interested [2][3].

In the automatic transmission gearbox, the speed ratio is selected automatically, which make the car driver free to press on the clutch and change the speed ratio manually, which make him comfortable in car driving along his way. Increasing the no. gears in the automatic transmission gearbox will increase the smoothing of acceleration of the vehicle. There are several types of automatic transmission gearboxes, for example, hydraulic, continuously variable, and semi-automatic transmission [4][5].

The hydraulic automatic transmission is the traditional type, which is most applied technique, it is used in most types of automobiles. This technique does not use the friction clutch such that used in manual transmission gear, but it utilizes a fluid coupling. The system of this transmission type consists of planetary gears, whereas, the gear changing can be achieved by hydraulically locking and unlocking to the system gear. The continuously variable and semi-automatic transmission technique differ from the traditional hydraulic transmission because, they depend on a computer to change the speed ratio of the gearbox, in this technique the computer calculate the speed of the driving wheels continuously using speed sensors, and then it select and change the speed ratio to a suitable choice. In continuously variable transmission CVT, an infinite speed ratios can be obtained using a belt and torque transmission scheme. The semi-automatic transmission type uses the clutch like the manual transmission type, but here, the clutch is controlled by using electro-hydraulic method. Generally, the continuously variable and semi-automatic transmissions types select the suitable speed ratio according to electric signals incoming from specific gears in the transmission gearbox [6][7].

The Back-propagation neural network is a powerful type of the intelligent systems, and it is the most used network in the intelligent applications.

The Back-propagation is a common method for training the artificial neural networks, which is used in conjunction with an optimization method such as gradient descent. This method calculates the gradient of a loss function with respect to all weights in the network. The gradient is fed to the optimization method, which is utilized to update the weights of the overall network in an attempt to minimize the loss function [8][9].

Back-propagation algorithm requires a known, desired output for each input value, in order to calculate the loss function gradient, therefore, it is classified as a supervised neural network, and is usually considered to be a method. It is a generalization of the delta rule to multi-layered feed-forward networks, made possible by using the chain rule to iteratively compute

gradients for each layer. Back-propagation requires an activation function (that is used by the artificial neurons) to be differentiable [10][11][12].

The whole Back-propagation process is intuitively very clear. When a learning pattern is clamped, the activation values are propagated to the output units, and the actual network output is compared with the desired output values, this process usually end up with an error in each of the output units. The final error value that is obtained from the output layer is used to generate the new (or updated) weights of connections of the whole network [13][14][15].

Generally, The Back-propagation neural network consists of three main layers: Input, hidden, and output layers. The input layer is single layer of a specific number of neurons, the hidden layer is a single or more layers of number of neurons which they use activation functions to present their outputs to the next layer (output layer), the output layer is a single layer of number of neurons, and it uses an activation function similar or different from that used in the hidden layer, to generate the error function for updating the weights of all connections of the whole network [16][17][18].

In this work, a simulation of Back-propagation neural network has been designed, and implemented in FPGA (type: Xilinx Spartan 6 SP-605 evaluation kit). This simulation software has been used for controlling the ratio speed of the electrical type of automatic transmission gearbox, whereas, this simulation selects and activates the proper speed ratio for the gear box according to the incoming data from the gearbox itself, this data represents the real speed of the driving wheels of the vehicle or car. For example, if the incoming data to the simulation software at the range (00-08)_H, i.e. the real speed of the automobile is at the range (0-20) km/h, so, the simulation software selects and presents the speed ratio number (1), and if the incoming data to the simulation software at the range (09-10)_H, i.e. the real speed of the automobile is at the range (21-40) km/h, so, the simulation software selects and presents the speed ratio number (2), and so on.

There are two reasons for using the Back-propagation neural network as an intelligent system in this work, they are: 1) The fast processing of this neural network due to the calculation of the gradient descent in its algorithm at each training iteration, the gradient descent is the partial derivative of the error function with respect to any weight and bias of the network, it represents the fast changing in the error function due to the changing of the weights and biases at each iteration, i.e. the gradient descent starts with maximum value at first iteration, and then it quickly decreases to minimum value in few iterations. 2) The high accuracy training of this neural network, whereas, this network generates two error functions, the first is generated from the output layer, which is used to update the weights and biases of input connections the output layer, while the second is generated from the hidden and output layers, which is used to update the weights and biases of input connections of the hidden layer, so, the Back-propagation neural network can present minimum error function in few iterations [19][20].

The proposed simulation system consists of three layers: Input layer, single hidden layer, and output layer. The input layer contains 8 neurons

because the resolution of the input data is 8 bits, the hidden layer involves 5 neurons, and the output layer involves 5 neurons. In this work, accurate results has been obtained due to the following reasons: 1) 'Trainlm' MATLAB function has been used as training function for the proposed neural network, 2) 'Satlins' MATLAB function has been used as linear activation function for the hidden layer, 3) 'Satlin' MATLAB function has been used as linear activation function for the output layer. Due to using 'linear' activation functions for the proposed intelligent system, this system has been easily converted to VHDL (Very high-Speed Integrated Circuit Hardware Design Language) code program, and then has been uploaded in FPGA successfully, whereas, if non-linear activation functions are used, the simulation system cannot be converted to VHDL code program and then cannot be uploaded in FPGA.

The proposed simulation system has utilized the FPGA for implementing its network due to the following reasons: 1) Fast processing of the FPGA because of its parallel processing, 2) Low power consumption, 3) Easy of implementing any network in FPGA.

The MATLAB software package has been used to built the proposed simulation system, and the ISE Design Suit software package has been used to upload the proposed VHDL code program in FPGA.

2. RELATED WORKS

Several previous works have related with topic of the proposed work, whereas, they have designed a simulation system of Back-propagation neural network for various practical applications.

In 2013, Nazri M. *et al.* [21] have proposed a Cuckoo Search CS Model, it is a new meta-heuristic search algorithm depends on behavior of the cuckoo birds that is used in training the Back-propagation neural network for achieving fast rate of convergence and avoiding problems of local minima. They have compared their proposed Cuckoo Search Back-propagation CSBB with artificial bee colony using Back-propagation algorithm and other hybrid variants, and they have used OR and XOR datasets specifically. They have presented simulation results that ensures and supports the high enhancing of the computational efficiency of the Back-propagation algorithm when it is coupled with the proposed hybrid method. They have used linear activation functions for the hidden and output layer, so , they have uploaded their simulation system in FPGA.

In 2014, Ramya J. *et al* [22] have proposed a recognition system of Tamil palm leaf manuscripts using Feed-Forward Back-propagation Neural Network FFNN technology. First, they have segmented the palm leaf manuscripts characters by exploiting the sliding window show and adaptive histogram calculation, then they have assembled the segmented characters and stored them in the database. The designed simulation program of this work is not uploaded in FPGA because they have used non-linear activation functions. The performance result graph of this work will be presented and discussed in the section of experiment and analysis in this paper.

In 2018, Wei L. *et al.* [23] have simulated three systems of tidal flow and a system of continuous vertical flow filled with synthetic wastewater, whereas, the influent and effluent concentrations has been compared to test: 1) nutrient removal in artificial Tidal Flow Constructed wetlands TFCWs, 2) the ability of a Back-propagation (BP) artificial neural network to predict nutrient removal. Non-linear activation functions have been used in BP neural network of this work, so, the proposed simulation system of this work is not uploaded in FPGA. The graph of behavior of the Mean-Square-Error MSE of this work will be presented and discussed in the section of the experiment and analysis of this paper.

3. PROPOSED METHOD

According to Table 1, there are several requirements must be considered for designing the proposed simulation system, they are:

- 1) The real speed of the driving wheels is represented by 8-bits data, which is entered to the input of the proposed simulation system, so, 8-neurons must be consisted in the input layer of the proposed simulation system.
- 2) Single hidden layer is used in the proposed system, which consists of 5-neurons.
- 3) The output of the proposed system is represented by 5-lines, they represent five control signals relate to 5-speed ratios for the automatic transmission gearbox, so, 5-neurons must be consisted in the output layer.
- 4) Two linear activation function must be used for the hidden and output layers of the proposed system, to make sure that the simulation system can be converted to VHDL code program, and then can be uploaded into FPGA successfully. The MATLAB activation functions: 'Satlins', and 'Satlin' have been used for the hidden and output layers of the proposed system.
- 5) For obtaining accurate results, the MATLAB 'Trailm' (Levenberg-Marquardt training) function has been used for learning the proposed system, which is an effective and powerful function.

Table 1. The relation between input and output data of the proposed system.

Speed Ratio	Real Speed (km/h)	Input Data (Hexadecimal)	Proposed System Outputs				
			m	n	o	p	q
1	0-20	00-08	1	0	0	0	0
2	21-40	09-10	0	1	0	0	0
3	41-60	11-18	0	0	1	0	0
4	61-100	19-28	0	0	0	1	0
5	≥101	29-FF	0	0	0	0	1

After execution the proposed software program that is ended by the statement {gensim(net)}, a simulation system block will be presented on the screen of the computer, which is illustrated in Figure 1. As shown, this system

block has 8-input lines for the input data, this data represents the real speed of the driving wheel, and has 5-output lines that represent 5-controlling signals for 5-speed ratios for the automatic transmission gearbox.

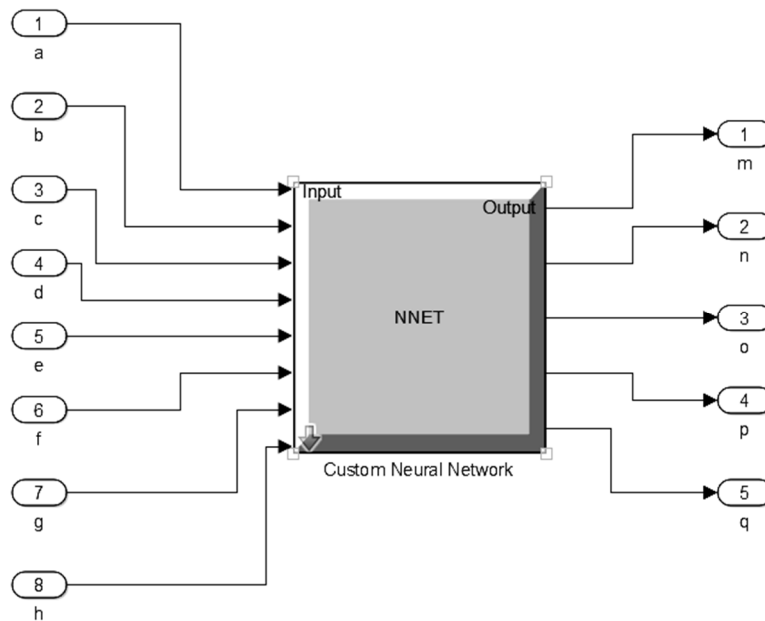


Figure 1. Simulation system block of the proposed system.

By clicking on "Look Under Mask" option of the proposed system block, a simulation neural network will be presented on the screen of the computer, which is demonstrated in Figure 2. As shown in this figure, the proposed simulation system consists of the following elements: 1) '8-input ports' are used for entering 8-bit binary data, where this data represents the real speed of the vehicle, 2) A 'multiplexer' is used for converting the 8-bit parallel input data to 8-bit serial data, 3) A 'data type conversion 1' block is used for converting the "Binary" data type of the proposed system to "Fixed" data type, 4) 'Process input 1' block is used for primary processing the serial input data, 5) 'Layer 1' block which represents the hidden layer of the proposed simulation system, 6) 'Layer 2' block which represents the output layer of the proposed simulation system, 7) 'Process output 1' block is used for final processing the output data, 8) Another 'data type conversion' block is used for converting the output "Fixed" data type to "Binary" data type, 9) A 'de-multiplexer' is used for converting the output serial data to parallel data.

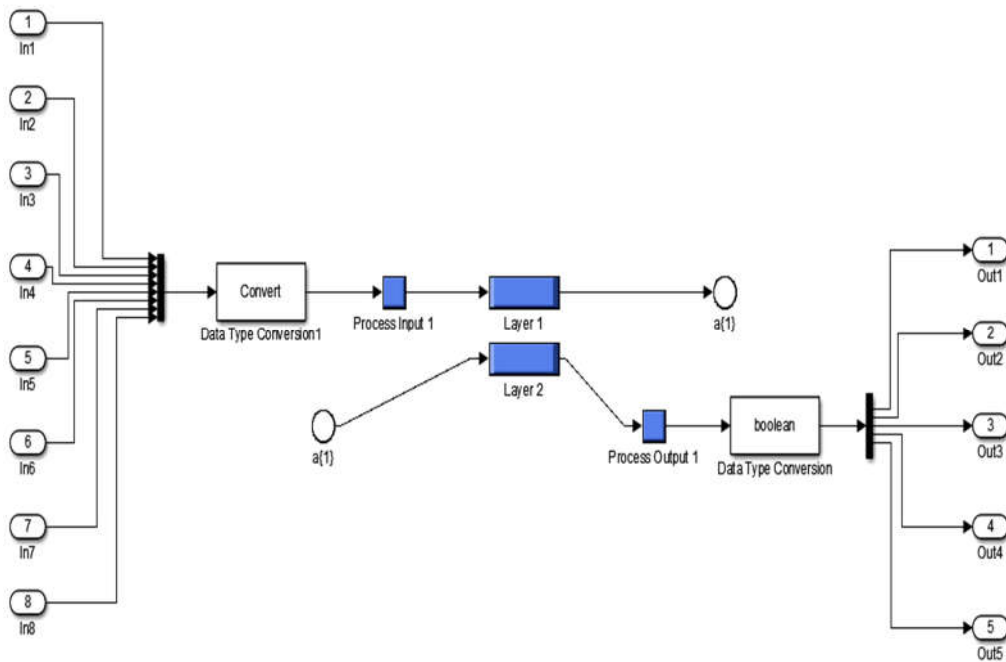


Figure 2. Details of the proposed neural network.

By clicking on the weight block of 'Layer 1' block of the proposed neural network in Figure 2, another network will be presented, which is illustrated in Figure 3, it represents the multiplication of the weights with the input connections of the hidden layer, this weight block consists of 5-weight units related to 5-neurons of the hidden layer.

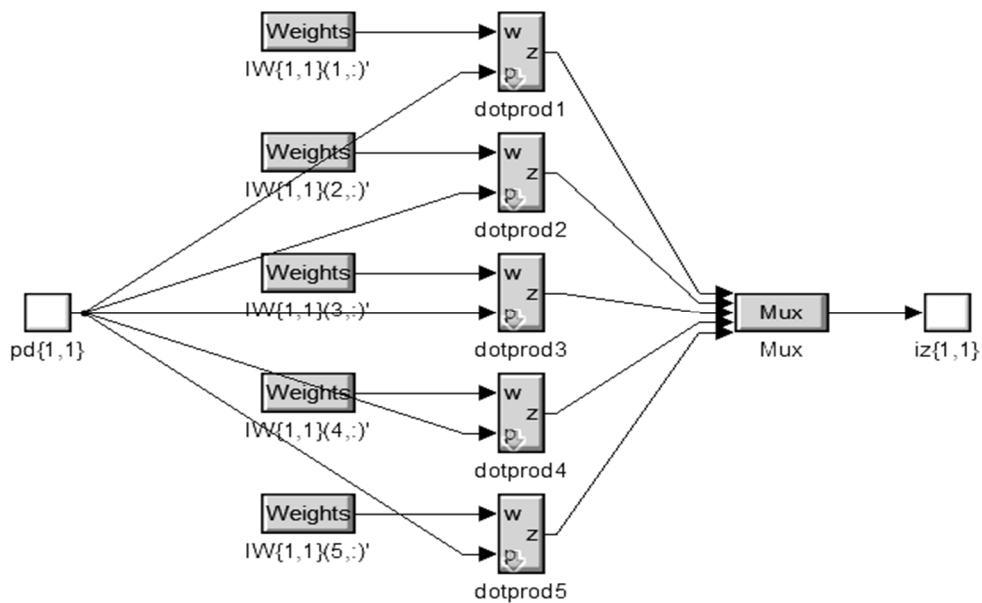


Figure 3: The weight block of the hidden layer of the proposed system.

By clicking on the weight block of 'Layer 2' block of the proposed neural network in Figure 2, another network will be presented on the screen of the computer, which is shown in Figure 4, it represents the multiplication of the related weights with the input connections of the output layer, this weight block consists of 5-weight units related to 5-neurons of the output layer.

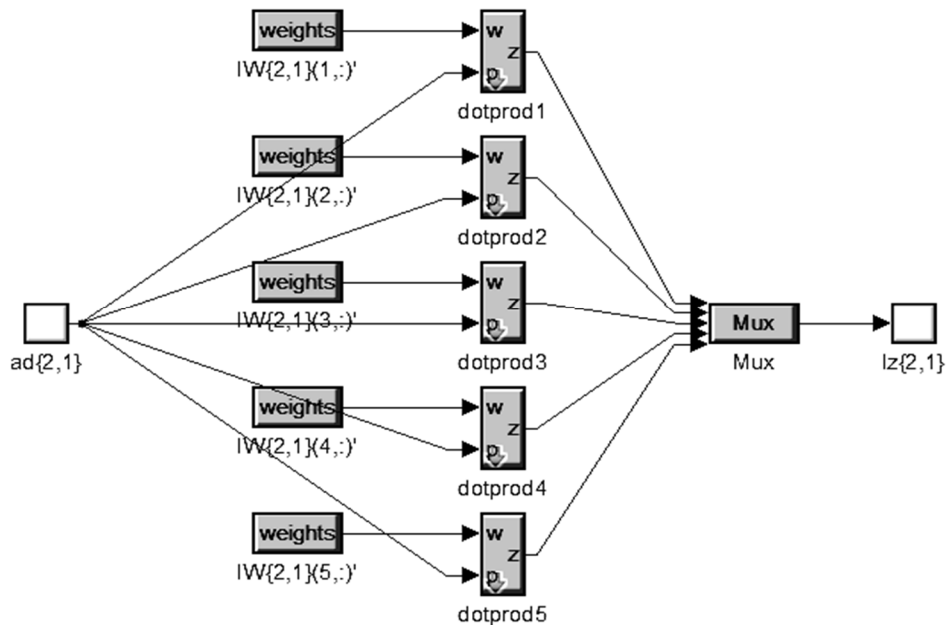


Figure 4. The weight block of the output layer of the proposed system.

The block diagram of the proposed software system with the related hardware system is illustrated in Figure 5. As shown in this figure, the overall process is performed by the following steps: 1) writing a suitable MATLAB program for building the proposed Back-propagation neural network, 2) this proposed MATLAB program is executed to generate a simulation block system on the screen of the computer, 3) the generated simulation block system is then converted to VHDL code program using MATLAB software package, 4) this VHDL code program is then uploaded in FPGA through JTAG (Joint Test Action Group) serial input bus, i.e. in this step the proposed simulation system will be implemented in FPGA. According to these four steps, the FPGA has been designed to have 8-useful input lines, and 5-useful output lines. The 8-input lines is connected to the output of the Speed-to-(8-bits) data converter, this converter is used to convert the output signal of the speed sensor to 8-bits binary data. The 5-output lines of the FPGA is connected to the 5-electrical controlling inputs of the Automatic Transmission Gearbox, these 5-controlling output lines of the FPGA correspond to the 5-related speed ratios of the gearbox.

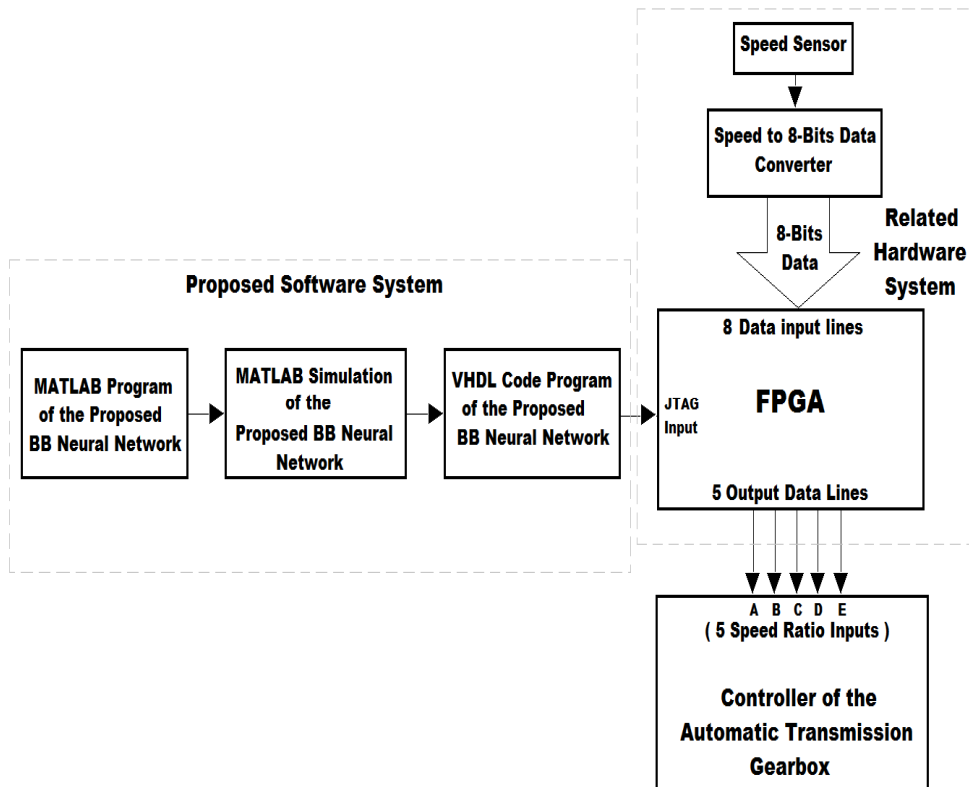


Figure 5. The block diagram of the proposed software system with the related hardware system.

4. EXPERIMENT AND ANALYSIS

After executing the proposed simulation system, a performance output result has been presented on the screen window, which is shown in Figure 6. This figure represents the relation between the Mean-Square-Error MSE and the number of iterations. As illustrated from Figure 6, the value of the MSE starts from 1 at epoch (iteration) 0, and it is gradually decreased to 3.94×10^{-25} at epoch 15, which is good and acceptable value if is compared with performance results of the previous related designs. The proposed simulation system has been trained with 15 iterations, which means that the training process has been performed in slight time if is compared with the results of previous related works, therefore, the proposed neural network is characterized by its high speed of training.

Another result has been presented on the computer screen by executing the proposed simulation system, it is the regression output result. This output result represents the relation between the desired (target) output and real output of the proposed simulation system, which is illustrated in Figure 7.

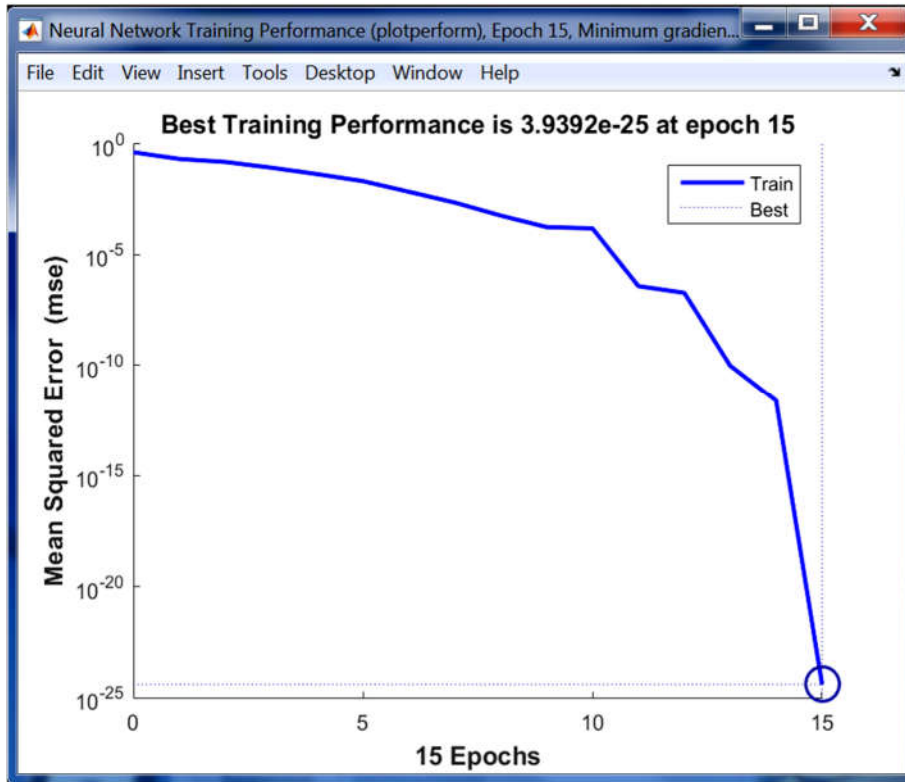


Figure 6. Performance output results of the proposed neural network.

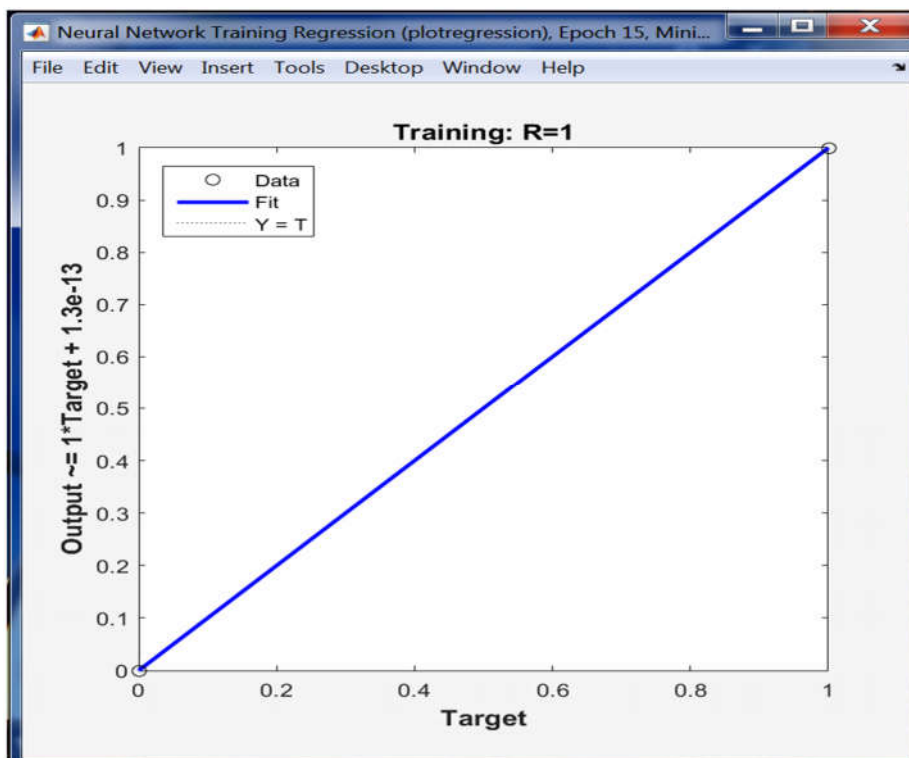


Figure 7. The regression output result of the proposed simulation system.

There are two fitted linear curve lines in Figure 7, the first is solid blue line, which represents the real relation between real and desired outputs of proposed system, and the second is the dotted black color line, which represents the fitting between real and desired outputs of the proposed system. As shown, the blue solid line is fitted to the dotted black line, which supports and ensures the 100% fitting between the real and desired output of the proposed system.

Ramya J. *et al* [22] have presented the result of performance of their proposed system, which is shown in Figure 8. As shown in this figure, The Mean-Square-Error MSE value starts from 1 at epoch 0, and then it has finished to 4.3×10^{-14} at epoch 24, which is higher than that of the proposed simulation system that is illustrated in Figure 6, and in same time, the simulation system of Ramya J. *et al.* has been trained slower than that of the proposed system of this paper, whereas, the simulation system of the Ramya J. *et al.* has been fully trained by 24 iterations, while the proposed system of this paper has been trained by 15 iterations.

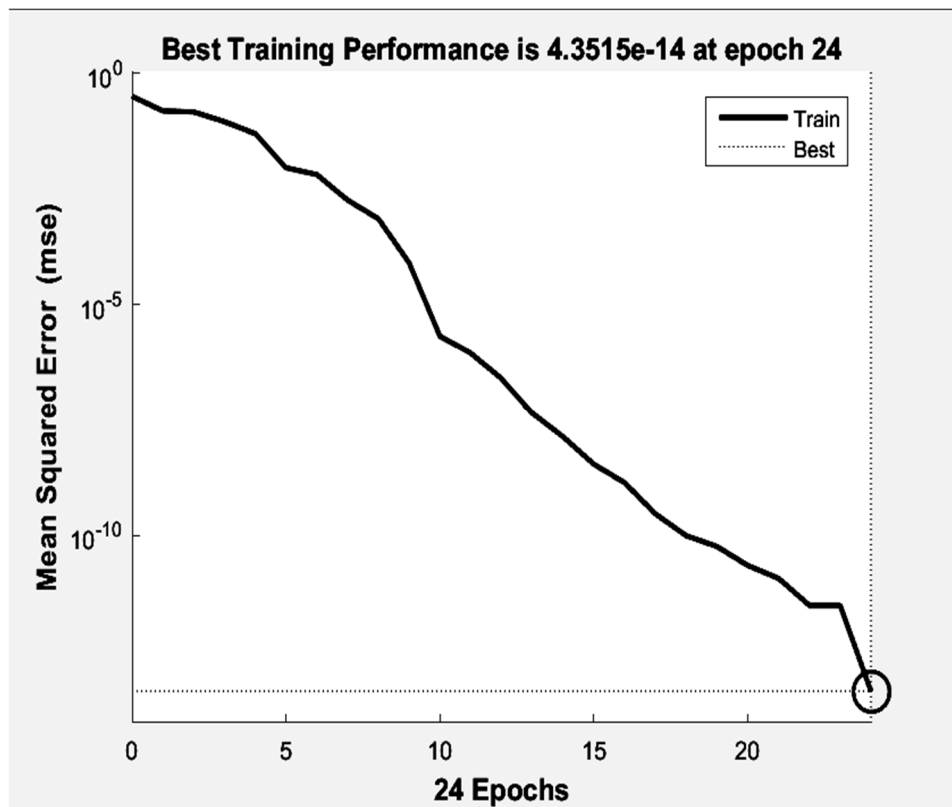


Figure 8. The performance result of Ramya J. *et al.* [20].

Wei L. *et al.* [23] have presented the performance result of their work, which is illustrated in Figure 9. As shown in this figure, the MSE value has started from 1 at epoch 0 and then has finished gradually to 1.3×10^{-15} at epoch 36, whereas, it is higher than that of the proposed system that is

shown in Figure 6, so one can say, the real output of the proposed system of this paper has higher accuracy than that of the simulation system of Wei L. et al. work. One can be observe the learning process of Wei L. et al. has been performed in 36 iterations which is more than that of the proposed system of this paper (which is 15 iterations), so, the learning speed of the proposed system of this paper is higher than that of the simulation system of the Wei L. et al.

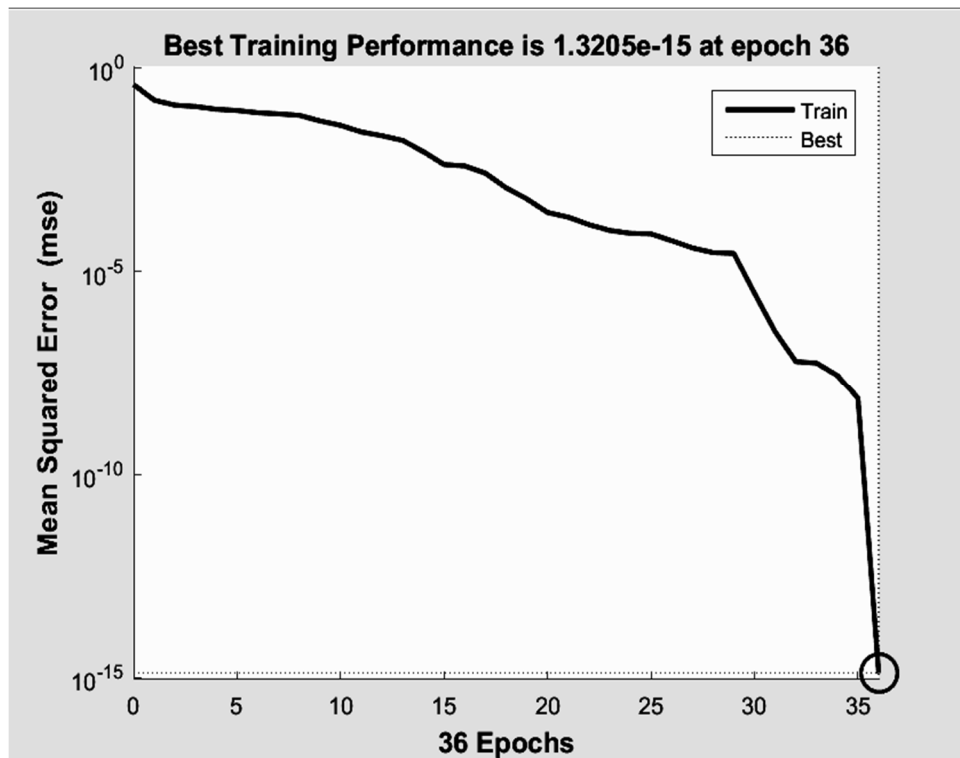


Figure 9. The performance result of Wei L. et al. [21].

The proposed Back-propagation neural network has been implemented successfully in FPGA (type: Xilinx SPARTAN 6 SP605 Evaluation Kit), which is performed using ISE Design Suit software package. The fundamental part of final results have been presented in Figure 10.

After implementing the proposed Back-propagation neural network in the FPGA, the execution time of the overall system has reached to 10 nsec, which is so slight and acceptable value. This execution time has been obtained due to the using of the Back-propagation neural network (which is fast intelligent system), and the FPGA (which is fast hardware tool).

There are four main characteristics must be considered for implementing the proposed simulation system in FPGA: the speed (bit rate), consumption power, size, and capability of upgrading. In this work, the proposed system has been implemented using intelligent simulation system based on FPGA, as known, the FPGA is characterized by its low power consumption, high speed (high bit rate reaches to 200 Mbits/sec), small size

of implementing, and the capability of upgrading. The low power consumption and high speed of processing is due to the using of FPGA which is hardware), and the capability of upgrading is due to the using of intelligent simulation system (which is software).

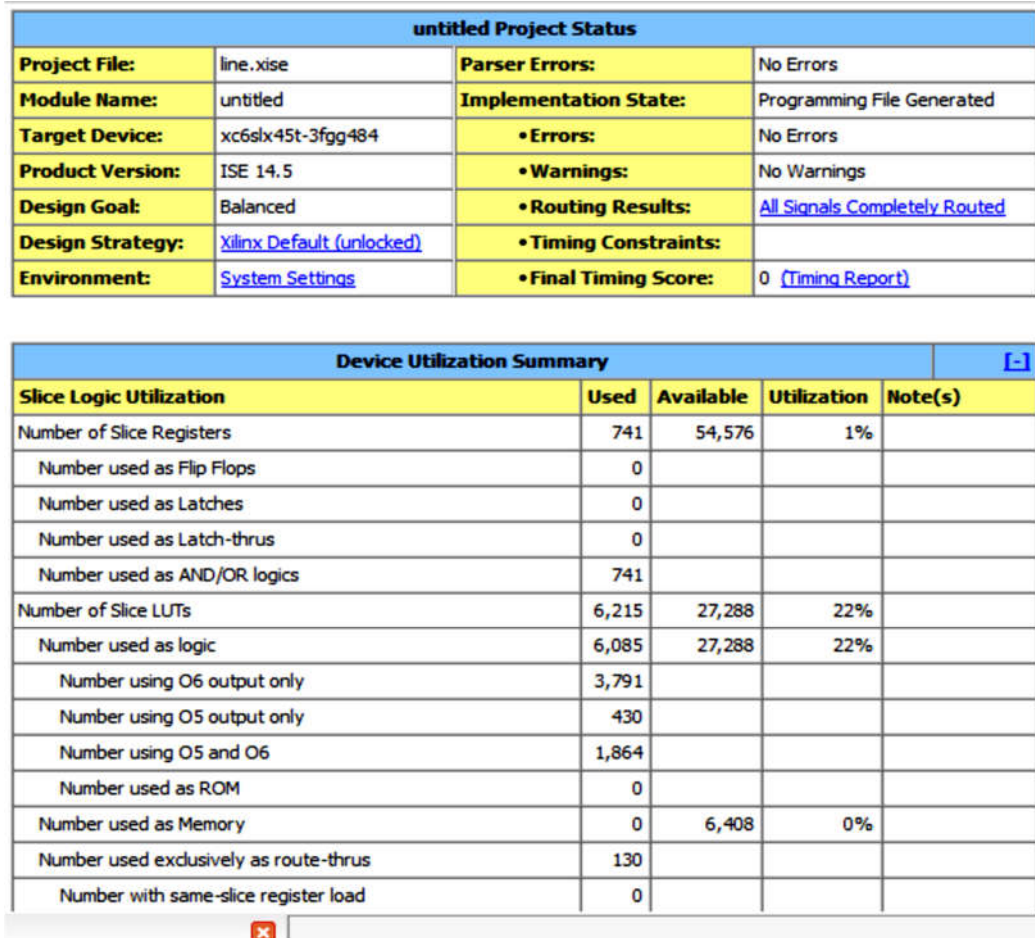


Figure 10. Fundamental part of final results for uploading the proposed neural network into FPGA.

The proposed system can be implemented using fast logic combinational circuit (Hardware) with the following characteristics: 1) the bit rate can reach to more than 550 Mbits/sec, which is fast than that of this work, 2) the consumed power nearly equals to that of this work, 3) the overall system can not be upgraded, 4) small size of building. Another technique can be used for the proposed system using fast RAM (Random Access Memory) or ROM (Read Only Memory) (which is hardware) , the characteristics of this technique are: 1) the bit rate can reach up to 550 Mbits/sec, which is fast than that of this work, 2) the consumed power nearly equals to that of this work, 3) the overall system can not be upgraded, 4) small size of building. Also, the proposed system can be built using low power microprocessor system(software with hardware) with the following characteristics: 1) The bit rate of this system is so slower than that of this

work, which depends on low clock frequency of the microprocessor and low speed of the used RAM and ROM of the system, 2) the consumed power is so lower than that of this work, 3) the software of this system can be upgraded, 4) small size of building. Finally, the proposed system can be built using high speed microprocessor system(software with hardware) with the following characteristics: 1) The bit rate of this system is higher than that of this work, which depends on the high clock frequency of the microprocessor and high speed of the used RAM and ROM of the system, 2) the consumed power is higher than that of this work, 3) the software of this system can be upgraded, 4) large size of building. Table 2 illustrates the characteristics of the various systems that can be used by the proposed system in this work.

Table 2. Comparison among various systems which can used by the proposed work.

The proposed system	Bits Rate (speed)	Power Consumption	Capability to Upgrade	Size
In this work	200 Mbits/sec	Low	Yes	Small
Using fast logic combinational circuit	>550 Mbits/sec	Low	No	Small
Using fast RAM or ROM	≤550 Mbits/sec	Low	No	Small
Using low power microprocessor System	< 200 Mbits/sec	so lower than in this work	Yes	Small
Using high speed microprocessor system	> 200 Mbits/sec	Higher than in this work	Yes	Large

5. CONCLUSIONS

The proposed intelligent system can be only utilized with electrical type of Automatic Transmission Gearboxes ATGs. High speed processing and high accuracy output can be achieved by designing an intelligent system to control and select the proper speed ratio for the ATG with the following steps: 1) choosing the Back-propagation neural network as the intelligent system, 2) using the Leveberg-Marquardt algorithm as a training method for the proposed neural network, 3) using MATLAB linear functions 'Satlins' and 'Satlin' as the activation functions for the hidden and output layers of the proposed neural network, 4) logic type of data can be used for the input and output information for the proposed system. With these considerations, one can observe the value of Mean-Square-Error MSE can reach to 3.94×10^{-25} in 15 iterations. For obtaining high speed of processing for the hardware system (where the bit rate can reach to 200 Mbits/sec and the execution time can reach to 10 nsec), the proposed neural network can be

implemented in FPGA, as known, the FPGA is a high speed and low consumed power tool, and it is faster from all low power microprocessors and microcontrollers due to its parallel processing. The essential condition for implementing the proposed neural network in FPGA, is the using of linear activation functions for the hidden and output layers of the proposed network, whereas, if the non-linear activation function is used for the proposed neural network, then this network is not implemented in FPGA.

REFERENCES

- [1] Mayur R. Mogre, **Comparative Study between Automatic and Manual Transmission Car**, International Conference on Mechanical, Automobile and Biodiesel Engineering, Dubai (UAE), Oct. 6-7, 2012.
- [2] Darko Stanojevi, Vladimir Spasojevi, Igor Stevanovi, and Aleksandar Nedi, **The Contemporary Automatic Gearboxes- review of the current state and interpretation of advantages and disadvantages of their use with respect to vehicle performance and terrific safety**, University of Belgrade, Serbia, 2013.
- [3] Andrew Moskalik, Aaron Hula, Daniel Barba, and John Kargul, **Investigating the Effect of Advanced Automatic Transmissions on Fuel Consumption Using Vehicle Testing and Modeling**, SAE International J. Engines, April 01, 2016.
- [4] Ngo D.V., **Gear Shift Strategies for Automotive Transmissions**, Technische Universiteit Eindhoven TU/e Co., January, 2012.
- [5] Wenchen Shen, Huilong Yu, Yuhui Hu, and Junqiang Xi, **Optimization of Shift Schedule for Hybrid Electric Vehicle with Automated Manual Transmission**, Energies Journal, 2016.
- [6] Jian Yao, Li Chen, and Fengyu Liu, **Experimental Study on Improvement in the Shift Quality for an Automatic Transmission Using a Motor-Driven Wedge Clutch**, Jian Yao et al., January, 2014.
- [7] Long-Chang Hsieh, and Hsiu-Chen Tang, **The Innovate Design of Automatic Transmission for Electric Motorcycle**, Canadian Society for Mechanical Engineering, Vol. 37, No. 3, 2013.
- [8] Jing Li, Ji-hang Cheng, Jing-yuan Shi, and Fei Huang, **Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement**, Springer-Verlag Berlin Heidelberg, Vol. 2, pp. 553-558, 2012.
- [9] Chih-Yao Lo, **Back Propagation Neural Network on the Forecasting System of Sea Food Material Demand**, Springer-Verlag Berlin Heidelberg, Part II, pp. 147-154, 2011.
- [10] Jyh-Woei Lin, Chun-Tang Chao, and Juing-Shian Chiou, **Back-propagation Neural Network as Earthquake Early Warning Tool using a new Elementary Modified Levenberg-Marquardt Algorithm to minimise Back-propagation Errors**, Author(s), May, 2018.
- [11] Bin Lin¹, Gaotong Lin, Xianyun Liu, Jianshe Ma, Xianchuan Wang, Feiyan Lin, and Lufeng Hu, **Application of back-propagation artificial neural**

- network and curve estimation in pharmacokinetics of losartan in rabbit**, Int J Clin Exp. Med., ISSN:1940-5901, 2015.
- [12] Taufik Ari Gunawan¹, M. Syahril Badri Kusuma, M. Cahyono, and Joko Nugroho, **The application of backpropagation neural network method to estimate the sediment loads**, EDP Sciences Co., 2017.
- [13] Amit Ganatra, Y. P. Kosta, Gaurang Panchal, and Chintan Gajjar, **Initial Classification Through Back-propagation in a Neural Network Following Optimization Through GA to Evaluate the Fitness of an Algorithm**, International Journal of Computer Science and Information Technology (IJCSIT), Vol. 3, No. 1, February, 2011.
- [14] Ali Hossain, Mijanur Rahman, Uzzal Kumar Prodhan, and Farukuzzaman Khan, **Implementation Of Back-Propagation Neural Network For Isolated Bangla Speech Recognition**, International Journal of Information Sciences and Techniques (IJIST) Vol.3, No.4, July, 2013.
- [15] Rashmi Amardeep, and K. Thippe Swamy, **Training Feed forward Neural Network with Back-propagation Algorithm**, International Journal Of Engineering And Computer Science, ISSN: 2319-7242, Vol. 6, Issue 1, January, 2017.
- [16] Kamil Dimililer, **Backpropagation Neural Network Implementation for Medical Image Compression**, Hindawi Publishing Corporation Journal of Applied mathematics, 2013.
- [17] Rama Kishore, and Taranjit Kaur, **Backpropagation Algorithm: An Artificial Neural Network Approach for Pattern Recognition**, International Journal of Scientific & Engineering Research, Vol. 3, Issue 6, June, 2012.
- [18] Omaima N. A. AL-Allaf, **Improving the Performance of Backpropagation Neural Network Algorithm for Image Compression / Decompression System**, Journal of Computer Science, ISSN 1549-3636, 2010.
- [19] Hanan A. R. Akkar, Aied K. Al-Samarrie, and Azzad B. Saeed, **Simulation Design of a Back-propagation Neural System of Sensor Network Trained by Particle Swarm Optimization**, International Journal of Scientific & Engineering Research IJSER, Vol. 7, Issue 4, ISSN: 2229-5518, April, 2016.
- [20] Hanan A. R. Akkar, Aied K. Al-Samarrie, and Azzad B. Saeed, **Design of an Artificial Neural Network for Sensor Network Applications** 15th Conference of Al-Mansoor collage, Baghdad, Iraq, 23-24, April, 2016
- [21] Nazri Mohd. Nawi, Abdullah Khan, and Mohammad Zubair Rehman, **A New Back-Propagation Neural Network Optimized with Cuckoo Search Algorithm**, Springer-Verlag Berlin Heidelberg, Part I, LNCS 7971, pp. 413–426, 2013.
- [22] Ramya J., and B. Parvathavarthini, **Feed Forward Back-propagation Neural Network Based Character Recognition System for Tamil Palm Leaf Manuscripts**, Journal of Computer Science, ISSN: 1549-3636, 2014.
- [23] Wei Li, Lijuan Cui, Yaqiong Zhang, Zhangjie Cai, Manyin Zhang, Weigang Xu, Xinsheng Zhao, Yinru Lei, Xu Pan, Jing Li, and Zhiguo Dou,

Using a Back-propagation Artificial Neural Network to Predict Nutrient Removal in Tidal Flow Constructed Wetlands, Water Journal, January, 2018.