Sigmoid Function Approximation for ANN Implementation in FPGA Devices

Djalal Eddine KHODJA[1], Aissa KHELDOUN[2], and Larbi REFOUFI[2]
[(1)]*Faculty of Engineering Sciences, University Muhamed Boudiaf of M'sila*
B.P N° 116 Ichebilia (28000), Algeria, Tel/Fax: +213 35 55 18 36,
E-mail: djalal_ed@yahoo.fr
[(2)] Signals & Systems Laboratory
Institute of Electronics and Electrical Engineering
Boumerdes University
Aissa1973@gmail.com

Abstract - The objective of this work is the implementation of Artificial Neural Network on a FPGA board. This implementation aim is to contribute in the hardware integration solutions in the areas such as monitoring, diagnosis, maintenance and control of power system as well as industrial processes. Since the Simulink library provided by Xilinx, has all the blocks that are necessary for the design of Artificial Neural Networks except a few functions such as sigmoid function. In this work, an approximation of the sigmoid function in polynomial form has been proposed. Then, the sigmoid function approximation has been implemented on FPGA using the Xilinx library. Tests results are satisfactory.

Keywords - ANN, FPGA, Xilinx, Sigmoid Function, power system..

1 Introduction

Monitoring, control and maintenance of any element of wide area power system become an important issue for ensuring the continuity of electric supply and avoiding the black out. It is important to early detect the defects that can occur in these elements by monitoring their operations and then developing methods for applying adaptive control and preventive maintenance [1, 2, 3].

The fast implementation of the developed methods is necessary needs for the industry and the complex (smart) power systems. These methods may be investigated using several techniques that have different characteristics to solve the encountering problems [4, 5]

The most commonly used techniques are the artificial intelligence-based techniques such as Artificial Neural Networks (ANN) [5, 6, 7] that they are easier to implement on electronic circuit board such as: Digital Signal Processing (DSP) chips, Application Specific Integrated Circuits (ASICs) or Field programmable gate array (FPGAs) [8, 9, 10]. The objective of this work is the implementation of artificial neural network on a FPGA. This implementation aim to contribute in hardware integration solutions using FPGA applied to different areas such as monitoring, diagnosis, maintenance and control of power systems.

In this study, we begin by adapting ANN to allow optimal implementation. This implementation must ensure efficiency, timeless and a minimum possible space on the FPGA. Then, we schedule the ANN on the System Generator. The System Generator to generate VHDL code. This code is verified and implemented on a FPGA Spartan-like by the ISE software from Xilinx Fondation.

2 ANN and simulation

To ensure efficiency, timeless and a minimum possible space on the FPGA, we propose a network that contains two hidden layers the first layer has three neurons and the second has two neurons, and output layer has two neurons.

We conducted a machine learning using the MATLAB software to where we obtain a smallest squared error (2.7401 e^{-015}) for 202 inputs, (see Fig .1).

The design of ANN through the use Simulink is shown in the figure 2.

ISBN: 978-960-474-262-2

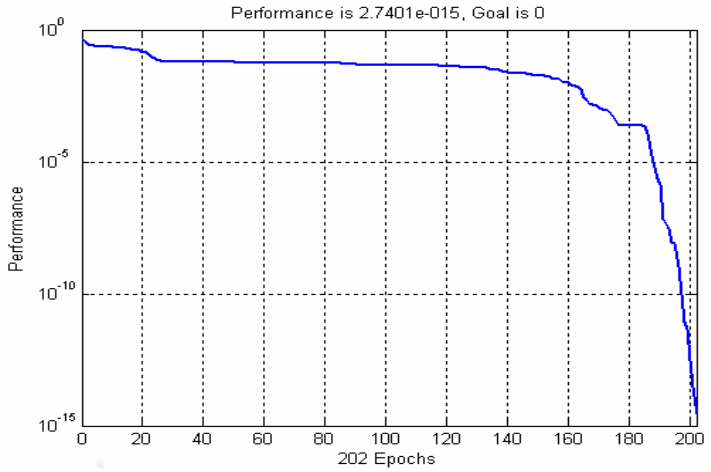

Figure 1 Evolution of average quadratic error of ANN

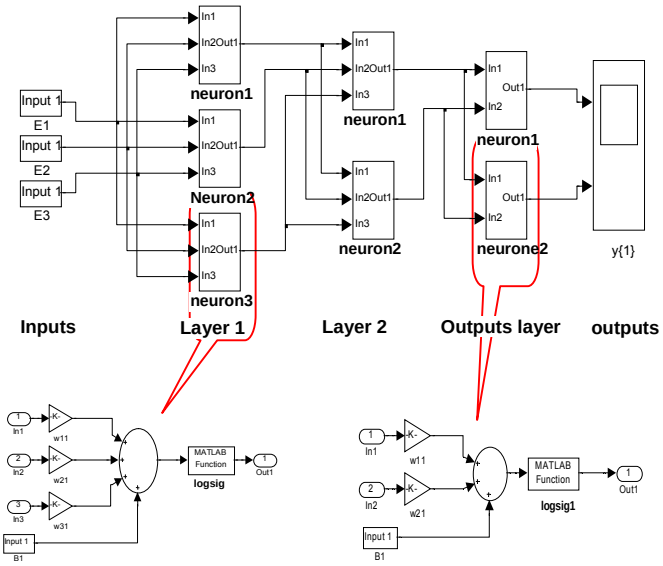

Figure 2 ANN using Simulink library

2.2 Test of ANN

Once the ANN that is implemented and learning performance has reached. Satisfactory tests are investigated, their results are presented in table1.

Table.1: Simulation (test) Results of ANN for different cases.

Output of ANN	Safety case	Fault1	Fault2
S1	$1.2222\ e^{-018}$	$1.0000\ e^{+000}$	$6.2836\ e^{-016}$
S2	$1.0000\ e^{+000}$	$1.6661\ e^{-007}$	$1.0000\ e^{+000}$

From the obtained results in the testing phase, it can be found that almost ANN in accordance (with error E^{-008}) with the desired predetermined outputs.

3 ANN Review and Simulation of the System Generator

The Simulink library provided by Xilinx, works in the same principle as the other elements of Simulink. It contains blocks representing different functions that are subjected and interconnected to form algorithms [8]. These blocks do not only serve to simulation, but can also generate VHDL or Verilog code.

In addition, the library provided by Xilinx to Simulink, has all blocks that are necessary for the design of ANN except a few functions such as sigmoid function..

For this reason, we suggested an implementation of the sigmoid function through the use of Taylor series [9], the resulting function (of order 5) causes an error of 0.51% with respect to the continuous model.

In our work, the sigmoid function has been approximated in polynomial form as given in the following :function,

$$f\ (x) = \frac{1}{1+e^{-cx}} \tag{1}$$

Equation (1) can be rewritten as follows :

$$f\ (x) = c + bx + ax^2 \tag{2}$$

Where the coefficients c, b, a are given by:

first:
$$\begin{cases} c = \dfrac{1}{1+e^{-a1}} \\ b = \dfrac{1}{1+e^{-a2}} \\ a = \dfrac{1}{1+e^{-a3}} \end{cases}$$

with:

$$\vec{a} = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \quad where \quad \vec{a} = \vec{V}/\vec{F}$$

ISBN: 978-960-474-262-2

$$and: \quad \vec{a} = \begin{bmatrix} 1 & 0^1 & 0^2 \\ 1 & 2^1 & 2^2 \\ 1 & 4^1 & 4^2 \end{bmatrix},$$

$$\vec{F} = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix} .$$

Modeling the resulting function is given by Xilinx as shown in Fig.3:

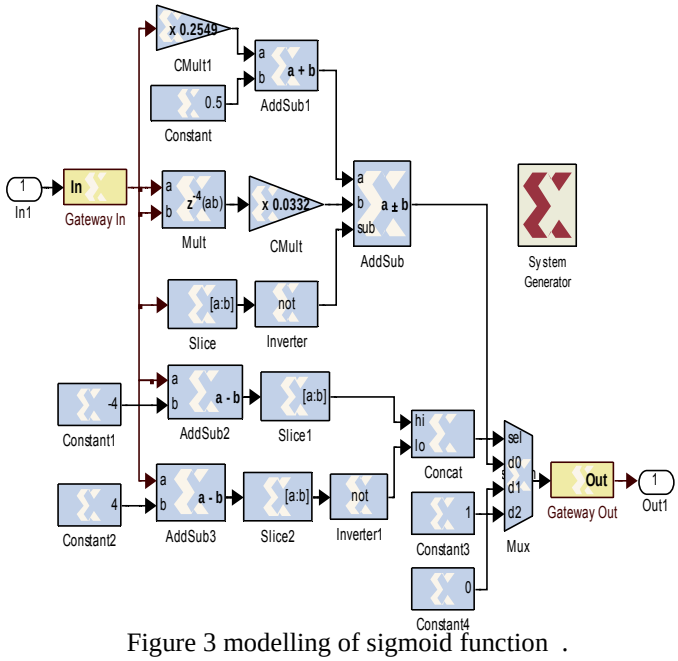

Figure 3 modelling of sigmoid function .

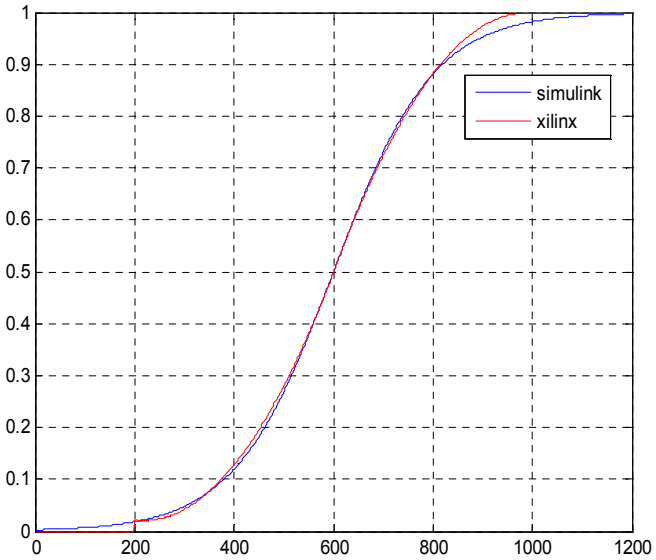

Figure 4 Comparaison of Sigmoid functions of Simulink and Xilinx

Figure.4 shows that the sigmoid function curve obtained by Simulink is in accordance with the curve of sigmoid function obtained by Xilinx. The design of ANN using Simulink is shown in Figure.5

According to the construction of ANN in the Simulink/Xilinx the system generator can also generate the VHDL code.

4 Implementation of ANN in FPGA

We place the entire contents of the block of ANN in the same subsystem, with a single generator system, then, it generates the VHDL code (see figure.6).

This part is devoted to description of the implementation of ANN on FPGA. In order, to do this, we use the Xilinx System Genrator.

Synthesizer: you click on Synthesizer to find space resources occupied by the VHDL (see figure.7)

The result is displayed in the Synthesizer as given in table.2.

From the obtained results shown in the table, we notice that the VHDL code occupies an area of 208% on FPGA, means that the FPGA type Spartan2 xc2s200e-6tq does not support this VHDL code.

However, the implementation of the code obtained in a system Spartan3 gives the results shown in table 3. From this table, we found that the code implemented in Spartan3 takes only 63% of storage space of FPGA.

5 Conclusion

In this work, we proposed a simple algorithm for the implementation of the ANN. The proposed hardware synthesis algorithm is performed by the System Generator. Routing and implementation in FPGA type Spartan2E was made by ISE Foundation.

The use of high-level design tool, System Generator is very beneficial for the verification of the behaviour of the algorithm in Simulink. The simulations for the sigmoid function show that the obtained results using Xilinx give the same performance as the sigmoid function obtained by Simulink.

Furthermore, the implementation of the obtained code using two different systems Spartan2 and Spartan3 leads to the conclusion that the FPGA type Spartan3 supported the generated VHDL code gives adequate results.

Finally, one can conclude that the implementation of functions in FPGAs is easier because now VHDL code of any circuit can be generated not only by an advanced language but also by the software MATLAB. This gives us the ability to simulate any function in simulink, using the library of Xilinx after that the latter is directly converted into VHDL by System Generator.

ISBN: 978-960-474-262-2

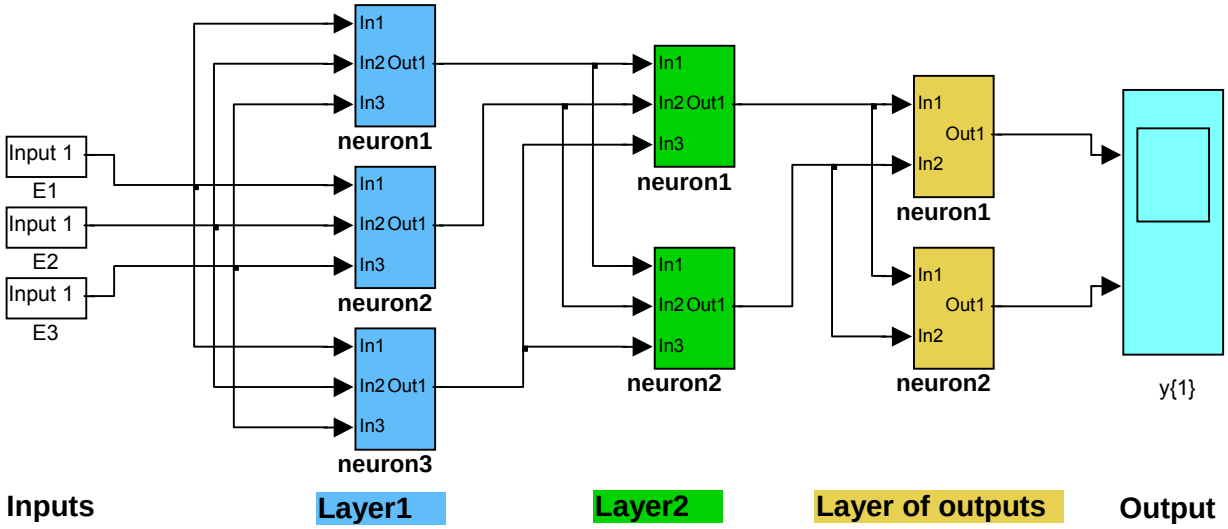

Fig.5 programming ANN using SIMULINK

Table.2 : Space used on FPGA

RNA_CW Project Status			
Project File:	rna_cw.ise	**Current State:**	Synthesized
Module Name:	rna_cw	**• Errors:**	No Errors
Target Device:	xc2s200e-6pq208	**• Warnings:**	775 Warnings
Product Version:	ISE, 8.1i	**• Updated:**	

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4895	2352	208%
Number of Slice Flip Flops	4777	4704	101%
Number of 4 input LUTs	8820	4704	187%
Number of bonded IOBs	167	146	114%
Number of GCLKs	1	4	25%

Detailed Reports					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current		0	775 Warnings	15 Infos
Translation Report					
Map Report					
Place and Route Report					
Static Timing Report					
Bitgen Report					

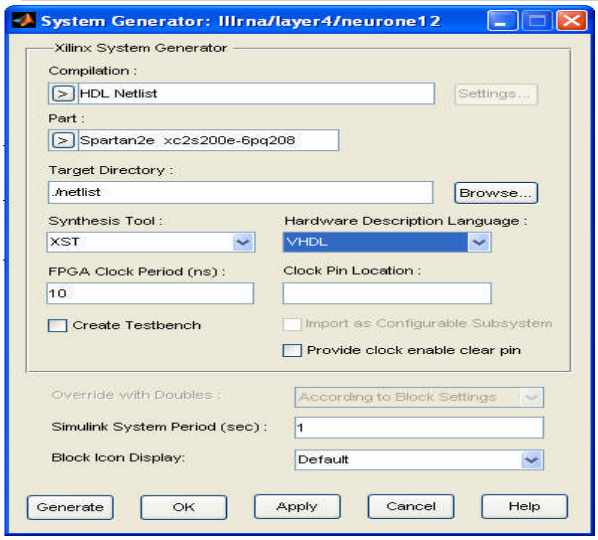

Figure 6 System Generator interface.

Table.3: Summary of used space on the FPGA**.**

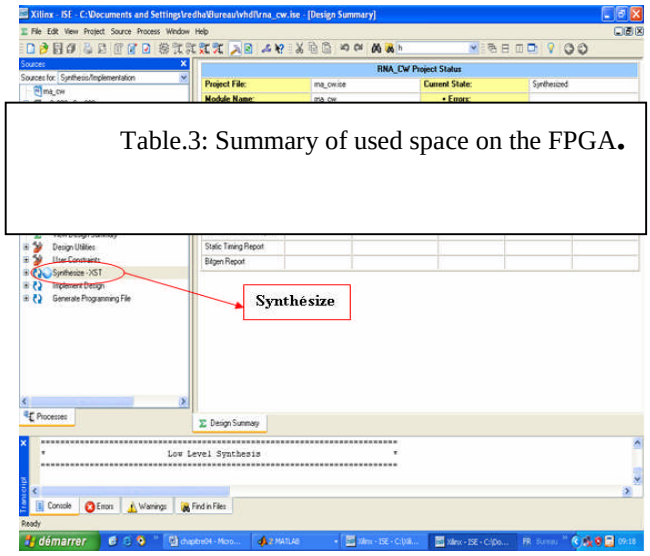

Fig.7 Space occupied by the implementation.

References

[1] G.Zwinngelsten, « *Diagnostic des défaillances: théorie et pratique pour les systèmes industriels* », Ed. Hermès Paris. 1995.

[2] B.Dubuisson ; « *Détection et diagnostic des pannes sur processus* », Technique de l'ingénieur. R7597,1992.

[3] B.Chetate, DJ.Khodja, « *Diagnostic en temps réel des défaillances d'un ensemble Moteur asynchrone-convertisseur électronique en utilisant les réseaux de neurones artificiels*», Journal Electrotekhnika, Moscou 12/2003, pp:16-20

ISBN: 978-960-474-262-2

Table 3 Summary of used space on the FPGA

SUBSYSTEM1_CW Project Status			
Project File:	subsystem1_cw.ise	**Current State:**	Synthesized
Module Name:	subsystem1_cw	**• Errors:**	No Errors
Target Device:	xc2s200e-6pq208	**• Warnings:**	775 Warnings
Product Version:	ISE, 8.1i	**• Updated:**	

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4895	7680	63%
Number of Slice Flip Flops	4777	15360	31%
Number of 4 input LUTs	8820	15360	57%
Number of bonded IOBs	167	333	50%
Number of GCLKs	1	8	12%

[4] DJ.Khodja, B.Chetate, «*ANN system for identification and localisation of faillures of anatomawed electric asynchronous drive*», 2nd International Symposium on Electrical, Electronic and Computer engineering en Exhibition, March, 11-13, NEU-CEE 2004, NICOSIA, TRNC pp : 156-161.

[5] DJ.Khodja, B.Chetate, «*Development of Neural Network module for fault identification in Asynchronous machine using various types of reference signals*», 2nd International Conference PHYSICS and CONTROL, August,24-26, Physcon 2005, St Ptersburg, Russia, 2005,. pp : 537-542.

[6] V.A.Tolovka. « *Réseaux de neurones : Apprentissage, Organisation et Utilisation* », Ed, Entreprise de rédaction de journal 'Radiotekhnika' Moscou, 2001.

[7] N. Moubayed, *"Detection et Localisation Des Défauts Dans Les Convertisseurs Statiques"*, 6TH International Conference on Electromechanical and Power Systems, Chişinău, Rep. Moldova, October, 2007.

[8] K. Roy-Neogi and C. Sechen, *"Multiple FPGA Partitioning with Performance Optimization"*, IEEE Computer Society, Proceedings of the Third International ACM Symposium on Field-Programmable Gate Arrays (FPGA'95), 1995.

[9] K. Renovell, P. Faure, J.M. Portal, J. Figueras and Y. Zorian, *"IS-FPGA: A New Symmetric FPGA Architecture With Implicit SCAN"*, IEEE ITC International Test Conference, Paper 33.1, 0-7803-7169-0/2001.

[10] Areibi, G. Grewal, D. Banerji and P. Du *"Hierarchical FPGA Placement"*, CAN. J. ELECT. COMPUT. ENG., VOL. 32, NO. 1, Winter 2007.