Chaotic Synchronization of Neural Networks in FPGA

Elias de Almeida Ramos(✉), Vitor Bandeira, Ricardo Reis, and Guilherme Bontorin

Institute of Informatics, PPGC/PGMicro, Federal University of Rio Grande do Sul (UFRGS), Mailbox 15.064, Porto Alegre, RS 91.501-970, Brazil
{elias.ramos, vvbandeira, reis}@inf.ufrgs.br, bontorin@ufpr.br

Abstract. The objective of this work is to obtain a complete synchronization of Hopfield Neural Networks (HNN) with a delay using a Field Programmable Gate Array (FPGA) simulating in real-time a Natural Neural Networks (NNN). This work is motivated by research in Neurosciences involving the implantation of chips between the skull and the brain to prevent or ameliorate diseases such as Parkinson's, Epilepsy and Depression. Our contribution is the introduction of new synchronization techniques based on the Qualitative Theory of Differential Equations, Chaos Theory and Algebraic Topology substituting calculations using the Lyapunov Stability Criterion (LSC). The presented technique does not depend on the Neural Networks to be synchronized but also presents a lower computational cost in comparison with previous works. The results show that FPGAs are good platforms for such experiments.

Keywords: FPGA · Neurosciences · Synchronization · Neural Networks · Chaos Theory

1 Introduction

The brain is a system composed of neurons arranged in a highly complex network. In biological studies, it was observed that isolated neurons can emit signals irregularly, but analyzing a network, the signals are synchronized over time [1]. Synchronization is fundamental in certain stages of sleep, in addition to be related to some pathology such as Parkinson's disease [2]. In recent years, brain implants have emerged as an alternative to the usual medications for treating neurological diseases. Brain implants are electronic devices connected directly to the brain, usually on the surface below the skull. Implants aim to block, record or stimulate signals from networks of neurons or even isolated neurons.

It is of scientific knowledge that diseases of cerebral origin originate by the deficiency of the neurons in the sending and receiving signals. Epilepsy, Parkinson's disease, and depression problems are being treated with Deep Brain Stimulation [2] which consists of the stimulation of isolated neurons or whole deficient regions. Among the current deficiencies, we can highlight synchronization failures of the neurons. Following research in Computational Neuroscience [3–5], information

D.A.C. Barone et al. (Eds.): LAWCN 2017, CCIS 720, pp. 17–30, 2017.
https://doi.org/10.1007/978-3-319-71011-2_2

processing in the brain is modeled by chaotic systems, more precisely, by nonlinear differential equations (Dynamical Systems).

In the study of the brain, the last decades show that it is necessary to introduce new techniques as nonlinear dynamics, since nerve cell activities or pattern recognition can't be totally modeled by classical tools [5]. In addition, experiments show that small disturbances in neural activities generate significant changes, the so-called "Butterfly Effect" [41].

In the last 25 years, techniques for Dynamic System Synchronization have been developed [11–13]. Such synchronizations have also application in several areas of computing, such as Cryptography and Signal Processing [15, 16]. On the other hand, to simulate Natural Neural Networks (NNN) we use Artificial Neural Networks (ANN). They are computational models inspired by the central nervous system of an animal where they are widely applied in problems of a computational nature. ANNs has evolved with robust techniques for solving machine learning problems, and pattern recognition [17, 18]. Among the models of ANNs, we can cite the Hopfield Neural Networks (HNN), which employs a principle called the storage of information in the form of dynamically unstable attractors [18]. The information retrieval happens through a dynamic process of updating the states of the neurons where the neuron to be updated is chosen randomly. Both the ANNs and synchronization problems can use parallel processing [18]. We are using a Field Programmable Gate Array (FPGAs). They are truly parallel because different processing operations do not have to struggle for the same resources. This allows users to create multiple specific task cores, possibly processed in parallel using multiples cores from a same chip. Hardware execution provides better performance and determinism compared to the vast majority of general purpose processor-based solutions. FPGAs have applications in various areas of computing such as Signal Processing, TV, Radio, Cellular Telephony, as well as in Computational Neurosciences [23, 24, 32]. They are used in research as a prototyping platform as well as for a final application [34, 36]. The main goal is to develop a complete synchronization method between ANNs using Algebraic Topology, Chaos Theory and Qualitative Theory of ODEs in FPGA reducing computational cost compared to Lyapunov Stability Criterion (LSC) for a possible brain implant that synchronizes signals from neurons. This paper is organized as follows. Section 2 describes the brain and the behavior of neurons defining the concepts of synchronization, chaotic synchronization and how they are present in the processing of the brain. In Sect. 3 the dynamic systems as well as the strange attractors, essential for the understanding of the dynamics of the neural signals and how the synchronization of these systems occurs, are formally presented. In Sect. 4, HNNs will be formally defined, which will model NNNs. In Sect. 5 Homotopies are defined, applications that deform a subset of topological spaces. In Sect. 6 the software and hardware experiment will be presented. In the last Section, the conclusion is presented in addition to future work.

2 The Brain

The brain is the main component of the Central Nervous System located inside the skull, divided into more than forty distinct areas, where each one performs a specific activity. We can say that it is the most formidable carbon-based processor. It consists of approximately 86 billion neurons (i.e., nerve cells), where each one has 10,000 connections called synapses [1]. Neurons are divided into sets known as Neural Networks. Each neuron is responsible for processing and disseminating electrical signals, and they control all activities of the organism. They consist of: (a) dendrites, that receive the stimuli being the input terminals; (b) central body, which performs information processing; (c) axons, transmits the stimuli, they are the output terminals.

2.1 Synchronization Between Neurons

Synchronization between neurons was observed in the works [6, 7]. In a healthy nervous system, the firing of neurons is not performed randomly. Usually, there is a low-frequency rhythm determining its activity. Studies show that synchronization plays a key role in brain functions involving memory and movement. Understanding how synchronizations occur is fundamental to understand the functioning of the brain and, as a corollary we can identify and avoid cerebral origin pathologies [33]. These pathologies are related to abnormal synchronization of neural firing. Hence, the need to construct theoretical models so that such results can be used for the treatment of diseases. In recent years, a new technique has brought interesting results. The implant of small devices between the skull and the surface of the brain. For Parkinson's disease [2] the device electrodes trigger electrical stimuli to the region of the brain that has symptoms of the disease. Neurology attests that chaotic behaviors are found in the brain [5], both at the neurons and global activity levels. So, unique tools are needed to understand the phenomena of the brain that are introduced in the next Section.

3 Dynamical Systems

Natural phenomena, even governed by simple equations have chaotic behavior. The presence of variables at first negligible results in generating unpredictable behaviors in the System. The Nonlinear Dynamic Systems, Chaos Theory, is composed of a range of tools derived from areas of Mathematics and their generality allow it to be used in the solution of problems of linear character [8].

At the end of the nineteenth-century, Poincare, in his works about celestial mechanics it was realized that for some differential equations it was not possible to find an understanding of the quantitative methods. Hence, Poincare introduced other unconventional elements of mathematics to attack such problems as Algebra, Topology and Differential Geometry [9]. Dynamic Systems are divided into two classes. (a) Conservatives: Those who conserve energy, in other words, system energy is constant. In practice, this measure will depend on the nature of the system (e.g., temperature, pressure, volume). (b) Dissipative: Those in which there is the loss of energy through dissipative factors (e.g., temperature drop, volume decrease). The biggest problem in Dynamic Systems theory is to understand its asymptotic behavior. Along the interactions, it can be generated sets of complicated geometry and chaotic behavior. They are named Strange Attractors.

3.1 Strange Attractors

Arise after a considerable number of interactions in a Dynamic System. Considering the effects produced, such attractors are extremely sensitive to the simplest variations in the initial conditions of their development, as the interactions advance over time. Thus, a certain pattern of disorder will be developed [9, 10]. The asymptotic behavior of a Dynamic System can evolve to: (a) a fixed point. (e.g. $f : [0,1] \rightarrow [0,1]$, $f(x) = ax$ with $a < 1$); (b) a periodic attractor. (e.g., the behavior of a pendulum); (c) a strange attractor. (e.g., Henon, Chua, Plikin, [10]).

3.2 Synchronization of Dynamic Systems

Occurs when two or more dissipative Dynamic Systems are coupled. Even if their trajectories deviate exponentially, this phenomenon is obtained experimentally and theoretically well studied [11–14]. We have two types of Chaotic Synchronization. (a) Generalized Synchronization, which occurs when the oscillators are distinguished as Master and Slave. (b) Phase Synchronization, that occurs with identical oscillators where the phase difference is limited, but their amplitudes are not correlated. Formally, (a) is defined as a function that varies with the time when applied to the Slave. It then receives the topological characteristics of the Master. In other words: the Slave is defined from the Master.

4 ANNs

They are computational dynamical systems inspired by the central nervous system of an animal. They were first presented in 1943 in the studies of McCulloch and Pitts until in 1949 Heeb published the article: 'The Organization of Behavior' [17], where he proposed a law of organization for neurons. It has several applications in computing such as processing and signals, speech recognition (e.g., images, speech) [18, 19], and in recent years are mainly used to solve Convex Optimization problems. ANNs present three types of learning: (a) Supervised, (b) Not Supervised, and, (c) By Reinforcement. In (a) the training set is presented where outputs will converge to this set. Others network such as (b), update their weights without a training set or any added reinforcement. Nonetheless, (c) has each entry reinforced to adjust the network outputs.

4.1 HNNs

First, we formulate the General Equation of the HNN. For this, we consider its synaptic weights $w_{i1}, w_{i2}, \ldots, w_{in}$, which represent the conductances where n is the number of neurons. Consider also $u_1(t), u_2(t), \ldots, u_n(t)$ the inputs (voltages). According to Kirchhoff's Current Law we have:

$$\frac{dv_i(t)}{dt} = -\frac{v_i(t)}{C_i R_i} + \sum_{j=1}^{n} w_{ij} g_i(v_i(t)) + \frac{I_i}{C_i} \quad (1)$$

where R_i is a Leak resistance, C_i a leakage capacitance and v_i is the field induced at the input of the neuron's Activation Function g_i [18].

Later the time delay was introduced $\tau > 0$ in the above equation resulting in our general form:

$$\frac{dv_i(t)}{dt} = \frac{-v_i(t)}{C_i R_i} + \sum\nolimits_{j=1}^{n} w_{ij} g_i(v_i(t-\tau)) + \frac{I_i}{C_i} \tag{2}$$

4.2 Synchronization of HNNs

This is a Methodology to develop HNN Synchronization of Chaotic Nature [18]. This Methodology consists of taking an HNN in its general equation given by its matrix form:

$$\phi^{'}(t) = -C\phi(t) + Af(\phi(t)) + Bf(\phi(t-\tau)) + I \tag{3}$$

By which we will call the Master System. Where $\phi(t)$ is the Neural Network state vector,

$$C = \begin{bmatrix} a_1 & & \\ & \ldots & \\ & & a_n \end{bmatrix} \in R^{n\times n}, a_i > 0,$$

$A, B \in R^{n\times n}$, f is the activation function satisfies the global Lipschitz Condition i.e.:

$$\|f(x) - f(y)\| \le k\|x - y\| \text{ for all } x, y \in R^n, k \ge 0 \tag{4}$$

τ is the delay time, and I is the input vector of the network. We will also take a System with the same Equation corresponding, but with different initial conditions by which we will call Slave System given by:

$$\varphi^{'}(t) = -C\varphi(t) + Af(\varphi(t)) + Bf(\varphi(t-\tau)) + I + U(t) \tag{5}$$

where U (t) is the portion that over time will generate synchronization [9, 10, 39].

The complete synchronization error dynamics are defined as:

$$e_t^{'} = \phi^{'}(t) - \varphi^{'}(t) \tag{6}$$

Master and Slave are synchronizable if:

$$\lim_{t\to\infty} \|e_t\| = 0 \tag{7}$$

In this work, we will synchronize two networks that do not use the U (t) portion. The discovery is an expensive task involving many equations and algebraic formulations, and $U(t)$ depends on the dynamic system [20]. This simplified method will be defined in the next Section, as well as the tool responsible for this simplification.

5 Homotopy

Algebraic Topology is an area of Mathematics that has recently presented satisfactory results in applications in Computing. Algebraic Topology presents valuable tools when studying space deformations, symmetries, translations, for example [21].

Among the deformations, we highlight Homotopies. Transformations that deform sub-sets of spaces continuously [22]. The main tool of this work.

Let be Ω (Omega) a Topological Space [22] and $I = [0,\ 1]$. Two continuous functions f and g are said Homotopic if there is a continuous function: $H : \Omega \times I \rightarrow \Omega$. Such that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$ for all $x \in \Omega$. The H function is called Homotopy between f and g.

In other words, for $t \in [0, 1]$ sufficiently close to 1, f and g are identical.

Theorem 1: *Let Φ and Ψ differentiable dynamic systems synchronizable. There is a Homotopy H such that*: $H(x,\ 0) = \Phi\ (x)$ *and* $H(x,1) = \Psi\ (x)$, $x \in R^n$.

Proof. Consider: $k = \frac{1}{t}, t \in [1, \infty)$. Define:

$$H(x, k) = (1 - k)\Phi(x) + k\Psi(x), x \in R^n \tag{8}$$

We have:

$$H(x, 0) = \Phi(x) \, \text{and} \, H(x, 1) = \Psi(x)$$

So:

$$\lim_{t \to \infty} \|\Phi(x) - H(x, k)\| = \|\Phi(x) - \Phi(x)\| = 0 = \lim_{t \to \infty} \|e_t\| \tag{9}$$

Q.E.D.

Corollary: *Let ϕ and ψ HNNs defined by Eq.* (3), *synchronizable. There is a Homotopy what synchronizes ϕ and ψ.*

Proof. Trivial.

Theorem 1 guarantees the use of the Eq. (8) in the synchronization implementation justifying the absence of the term $U(t)$ described in Sect. 4. This fact is responsible for the lower complexity of the algorithm since the calculation of $U\ (t)$ involves many algebraic operations [12–14, 37–39]. The Eq. (8) also ensures that we do not need to have the same dynamic system.

6 The Experiment

In this work, as previously mentioned, the method was simplified generating lower computational cost and execution time. Consider the HNN described by Eq. (10):

$$\phi'(t) = -C\phi(t) + Af(\phi(t)) + Bf(\phi(t-\tau)) + I \tag{10}$$

$$\text{Where } C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} -2 & 0 & 6 \\ -4 & 1 & -1 \\ -6 & -4 & -1 \end{bmatrix},$$

$$f(x) = tanh(x), \quad B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The solution of the system of ODEs in (10) has the solution a chaotic attractor Asteriscus described in Fig. 1.

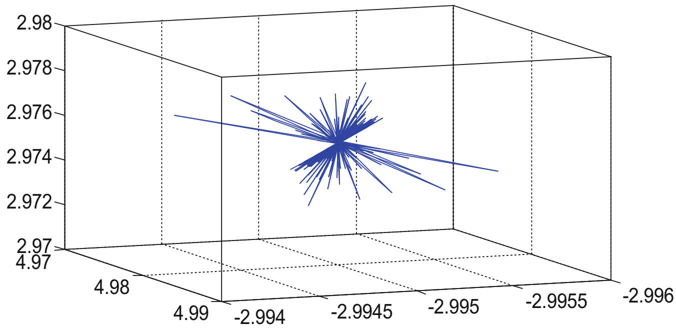

Fig. 1. Asteriscus attractor generated in the experiment

The system Master has the following initial conditions: (−3, 1, 3). Take system Slave with the following initial conditions (4, −1, 5).

$$\varphi'(t) = -C\varphi(t) + Af(\varphi(t)) + Bf(\varphi(t-\tau)) + I \tag{11}$$

In the implementation, we used the software MATLAB-2014. For the generation of the Master and the Slave was used the fourth-order Runge-Kutta Method and the synchronization was executed through the Linear Homotopy defined in Eq. (8). Let's take the errors in each coordinate after the synchronization defined by:

$$\begin{gathered} e_x = x_m - x_s, e_y = y_m - y_s, e_z = z_m - z_s \\ \text{where } (x_m, y_m, z_m) \in \text{ Master and } (x_s, y_s, z_s) \in \text{ Slave.} \end{gathered} \tag{12}$$

At this point are observed: (a) the systems to be synchronized need not necessarily to have the same system of equations, but in some cases, the approximation (i.e., synchronization) is only continuous and not differentially continuous; (b) on the other hand, in some cases, the only continuous approximation can be approximated to a differentially continuous. These cases will be exposed in next works.

6.1 Implementation

We adopted the number of interactions as a metric for the comparison between our implementation and related works. Since the hardware is different, this presents a straightforward comparison [39].

The experiment was run with an Intel Core 2 Duo processor, 4 GB of memory and Windows 7 Operating System and MATLAB performed the processes of resolution of the ODEs and synchronization in 5.6×10^{-2} s. The average time to obtain the complete synchronization occurred in 3 interactions. Being an interaction equal to 1.2×10^{-4} s the complete synchronization was obtained in 3.6×10^{-4} s, where following [12–14, 37, 38] the systems are synchronized when the average error is less than 0.5 units of measurement.

Considering the errors between the coordinates of system Master and Slave, the graphs below present the performance of the experiment, where the complete synchronization was tested using 500 interactions. The implementation by Homotopy attests to its functionality. Over time, the error reaches a stationary stage, and by Theorem 1, in the implementation of synchronization, we can adopt Homotopy by replacing the calculations established by the LSC used in the state-of-the-art [13, 14, 37–39].

Figure 2 show in the first three graphs the synchronization on each axis between the Master and the Slave and the last graph shows the synchronization error.

Comparing our method with other works, it was used the complexity and the minimum number of interactions of each method to perform a full synchronization. All

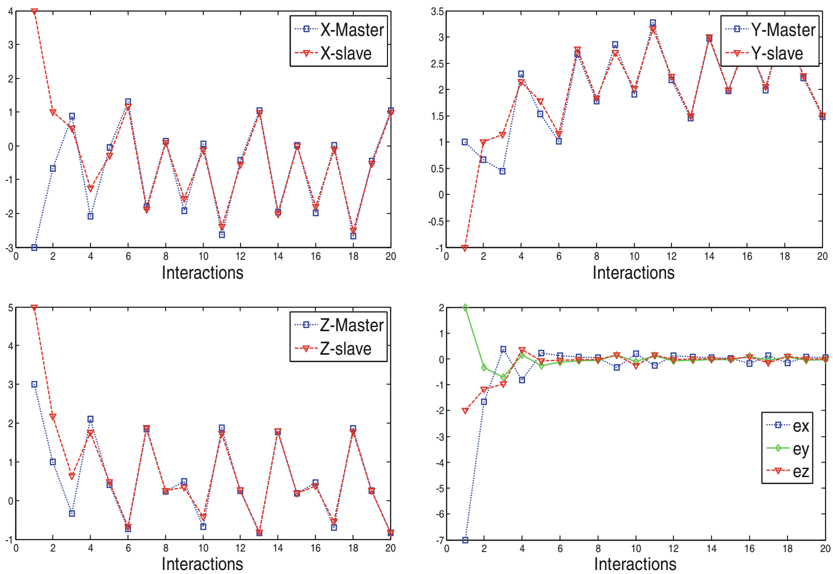

Fig. 2. On the three first graphs, it is presented the synchronization in the three axes x, y, z of Master and Slave respectively till 20 interactions. On the last, we see the errors between the Masters and the Slaves till 20 interactions.

use the Runge-Kutta method for the solution of the equation system. It is defined the total complexity of each method by:

$$TotalComplexity = n(Operations + RK)$$
$$Complexity = n \times Operations$$

where n is the interaction number to the solution (Runge-Kutta). The Homotopy method has a Complexity of $15n$. Table 1 shows the latest results in Chaotic Synchronization and the method involved. Homotopy was applied in each dynamic system of each work. A quick calculation on the complexities of the methods shows that we have a reduced number of operations compared to the others, in addition to a minimum number of interactions for full synchronization (column 'Time' in Table 1) is smaller or equivalent.

Table 1. Comparison of results (state-of-the-art)

Ref	Method	Complexity	Attractor	Time	Our time
[40]	LSC	$22n$	Lorenz	4	4
[39]	Adaptive control	$60n + n^2$	HNN	5	5
[12]	Active disturbance rejection	$50n$	ODEs	5	4
[13]	LSC*	$(23 + k)n$ **	Lorenz	5	4
[37]	Probability theory	kn^2 **	N. network	4	4

(*) LSC modified. (**) $k >=1$

6.2 Simulation in FPGA

The previous work shows that the FPGA implementation of the synchronizations of dynamic systems generates satisfactory results [25–27], where the synchronization is given by the LSC [26, 28]. In this work, as previously mentioned, the synchronization is simplified, generating a lower computational cost and execution time. Table 2 represents the state-of-the-art on chaotic synchronization problems in FPGAs. The most used attractors and the comparison with our work are presented. Unlike the previous work, the main tool is to approach the problem in general through the Qualitative Theory of ODEs (Algebraic Topology).

Table 2. Comparison of results (state-of-the-art)

Ref	Attractor	FPGA	Qualitative theory
[29]	Lorenz	Yes	No
[30]	Chua	Yes	No
[31]	Neural network	Yes	No
This	General	Yes	Yes

We chose to implement the chaotic synchronization on an FPGA as a demonstration of the concept. The tests consist of: (a) simple combinatory implementation of the equations using hardware description language (i.e. SystemVerilog); (b) behavioral simulations using ModelSim®; (c) circuit synthesis using QuartusII®; also, (d) programming an Altera® DE4 FPGA board to extract data after the execution.

A simple combinatory implementation was preferred as it can takes advantage of the parallel nature of FPGAs. For this reason, the calculation of each of the three dimensions (i.e. *x*, *y*, *z*) of both Master and Slave networks is concurrent. Subsequently, these outputs are operated together for the chaotic synchronization. An initial validation using behavioral simulation was performed using ModelSim®. After, we developed and compiled a synthesizable version with QuartusII®. Finally, the object code was uploaded to a DE4 board and results extracted. Altera manufactured the board with a Stratix IV chip that has 182,400 Arithmetic LookUp Tables (ALUTs). The circuit uses a total of 10,378 ALUTs, 139 Registers, 30 Digital Signal Processing (DSPs): 24 of 18bit and 6 of 36bit. The circuit has a maximum frequency of operation at 165.62 MHz (as per timing analysis performed by QuartusII). For better stability, the clock frequency was reduced to 150 MHz for the final implementation and results from extraction. At this frequency, the execution time to perform 500 iterations of synchronization was 6.7 microseconds. As the brain operates at a frequency that is an order of magnitude lower [34, 35]. The implementation allows for a real-time application with a brain interface. Nonetheless, it is 8.4×10^3 times faster than a MATLAB implementation.

Figure 3 shows in the first three graphs the synchronization on each axis. Synchronization errors are displayed in the fourth graph. Note that the error reaches its steady state (minimum value) in the second interaction.

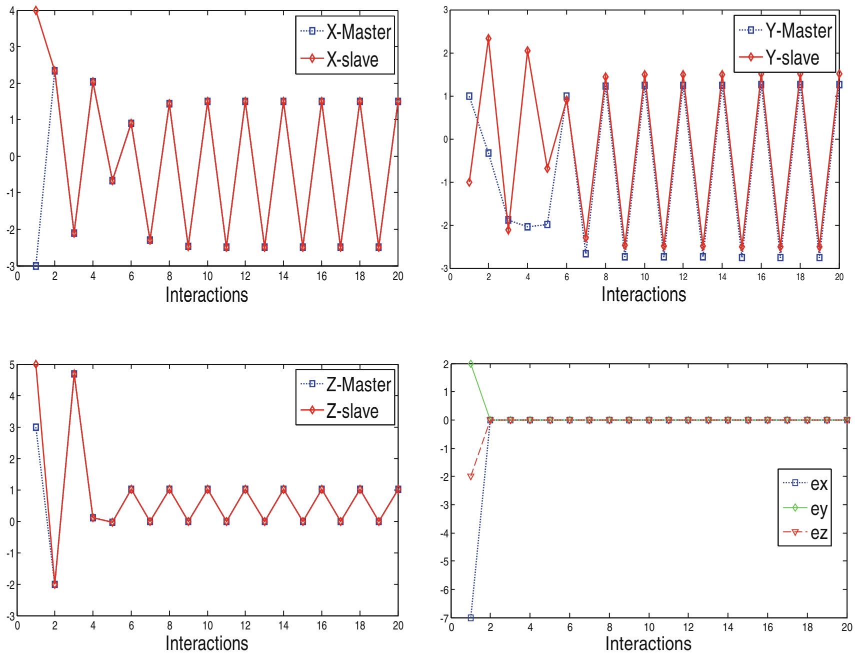

Fig. 3. Our implementation using a FPGA. On three first graphs, it is presented the synchronization in the three axes *x*, *y*, *z* of Master and Slave respectively. The last one present the errors between the Master and the Slave.

6.3 Comparison of Results (Hardware vs Software Implentation)

To compare the results of the implementations, software and hardware, due to the change of scale and speed of processing and different metrics were used the Largest Lyapunov Exponent (LLE) of each series (x, y, and z). The Lyapunov Exponents describes the speed at which two near points move away or approach in the course of time subjected to a dynamic system. These values defined in Eq. (13) represent a study in the behavioral sense between the time series, due to their mathematical properties being used for topological comparison among attractors. The LLE is a powerful tool that unites analytical and topological analysis [5, 9, 10].

Let be $\phi : R^n \to R^n$, differentiable. Define:

$$\lambda(x) = \lim_{\mathrm{n}->\infty} \ln\left|\frac{d\phi^n}{dx}\right| \tag{13}$$

$$\mathrm{LLE} = \max\{\lambda(x), \text{for all}\, x \in R^n\}$$

Table 3 shows the deviation of the LLEs of the implementations, where the maximum deviation is 0.06. So, the implementation in hardware preserves the topological properties of the implementation in software making viable the application in hardware.

Table 3. Comparison of results (LLE)

	X-Master	Y-Master	Z-Master	X-Slave	Y-Slave	Z-Slave
MATLAB (*LEE*)	1.69	1.32	1.91	1.39	1.38	1.91
FPGA (*LEE*)	1.67	1.38	1.89	1.42	1.38	1.89
ΔLEE	0.02	0.06	0.02	0.03	0.00	0.02

7 Conclusions

In this paper, it was proposed a methodology to perform the complete synchronization between HNNs simulating NNNs in FPGA using Algebraic Topology tools. The motivation originates in the study of implants in the human brain, with the objective of treating neurological diseases caused by the synchronization deficiencies of the neurons. The methodology is independent of dynamic systems and it has a lower computational cost compared to the state-of-the-art. The application of Homotopy presented satisfactory results, which were verified analytically and experimentally. The results of numerical simulations indicate that the proposed methodology is efficient and convenient for synchronization using FPGAs, where the topological and differential properties of the software implementation were preserved, and the implementation allows for a real-time application with a brain interface.

In future works, we will study simultaneous synchronizations in two or more neural networks. Finally, the experiment confirms the possibility, in theory, that if a brain implant is worked to perform the synchronization of neurons we have appropriate tools for this purpose. To generalize the method for any dynamic system, we first want to analyze systems with high differentiation (second and third derivative) as well as systems defined in larger dimensions. Finally, we want to implement other classes of Homotopy following the implementations in FPGA.

References

1. Odekerken, V.J., Boel, J.A., Geurtsen, G.J., Schmand, B.A., Dekker, I.P., de Haan, R.J., Schuurman, P.R., de Bie, R.M.: Neuropsychological outcome after deep brain stimulation for Parkinson disease. Neurology **84**, 1355–1361 (2015)
2. Little, S., Pogosyan, A., Neal, S., Zavala, B., Zrinzo, L., Hariz, M., Foltynie, T., Limousine, P., Ashkan, K., Fitzgerald, J., Green, A.L., Aziz, T.Z., Brown, P.: Adaptive deep brain stimulation in advanced Parkinson disease. Ann. Neurol. **74**(3), 449–457 (2013)
3. Bob, P.: Chaos, Cognition and Disordered Brain. Activitas Nervosa Super. **50**(4), 114–117 (2008)
4. Cerutti, S., Carrault, G., Cluitmans, P.J., Kinie, A., Lipping, T., Nikolaidis, N., Pitas, I., Signorini, M.G.: Non-linear algorithms for processing biological signals. Comput. Methods Programs Biomed. **51**(1–2), 51–73 (1996)
5. Maron, G., Barone, D.A.C., Ramos, E.A.: Measuring the differences between spatial intelligence in different individuals using Lyapunov exponents. In: Proceedings of the 7th International Conference on Mass-Data Analysis of Images and Signals, MDA 2012, Berlin (2012)
6. Linas, R.R.: Intrinsic electrical properties of mammalian neurons and CNS function: a historical perspective. Front Cell Neurosci. **8**, 320 (2014)
7. Cabral, J., Luckhoo, H., Woolrich, M., Joensson, M., Mohseni, H., Baker, A., Kringelbach, M.L., Deco, G.: Exploring mechanisms of spontaneous functional connectivity in MEG: how delayed network interactions lead to structured amplitude envelopes of band-pass filtered oscillations. NeuroImage **90**, 423–435 (2014)
8. Frederickson, P., Kaplan, J.L., Yorke, E.D., Yorke, J.A.: The Liapunov dimension of strange attractors. J. Differ. Equ. **49**(2), 185–207 (1983)
9. Viana, M.: Dynamical systems: moving into the next century. In: Engquist, B., Schmid, W. (eds.) Mathematics Unlimited and Beyond. Springer, Heidelberg (2001). https://doi.org/10.1007/978-3-642-56478-9_32
10. Viana, M., Alves, J.F., Bonatti, C.: SRB measures for partially hyperbolic systems whose central direction is mostly expanding. Invent. Math. **140**, 298–351 (2000). Reprinted in the theory of chaotic attractors. Dedicated to J.A. Yorke in commemoration of his 60th birthday. Edited by B.R. Hunt, J.A. Kennedy, T.-Y. Li and H.E. Nusse. Springer Verlag, 443–490 (2004)
11. Pecora, L.M., Carroll, T.L.: Physical review letters. Phys. Rev. Lett. **64**, 821 (1990)
12. Khadra, F.A.: Synchronization of chaotic systems via active disturbance rejection control. Intell. Control Autom. **8**, 86–95 (2017)
13. Ouannas, A., Abdelmaleka, S., Bendoukhaba, S.: Coexistence of some chaos synchronization types in fractional-order differential equations. Electron. J. Differ. Eqn. **2017**(128), 1–15 (2017)
14. Zhang, Q., Lu, J.-A.: Chaos synchronization of a new chaotic system via nonlinear control. Chaos Solitons Fractals **37**(1), 175–179 (2008)
15. González-Miranda, J.M.: Synchronization and Control of Chaos. An Introduction for Scientists and Engineers. Imperial College Press, London (2004)

16. Al-Sawalha, M.M.: Projective reduce order synchronization of fractional order chaotic systems with unknown parameters. J. Nonlinear Sci. **10**, 2103–2114 (2017)
17. Barone, D.A.C.: Sociedades Artificiais: a nova fronteira da inteligência nas máquinas. Bookman, Porto Alegre (2003)
18. Haykin, S.: Redes neurais: princípios e prática. Trad. Paulo Martins Engel. 2. edn. Porto Alegre, Bookman (2001)
19. Hebb, D.O.: Distinctive features of learning in the higher mammal. In: Delafresnaye, J.F. (ed.) Brain Mechanisms and Learning. Oxford University Press, London (1961)
20. Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., Zhou, C.: Synchronization in complex networks. Phys. Rep. **469**(3), 93–153 (2008)
21. Lima, E.L.: Grupo Fundamental e Espaços de Recobrimento, 4ª edição. IMPA (2012)
22. Lamure, H., Michelucci, D.: Solving geometric constraints by Homotopy. In: Third ACM Symposium on Solid Modeling and its Applications, pp. 263–269. ACM Press (1995)
23. Ahmed, E., Rose, J.: The effect of LUT and cluster size on deep-submicron FPGA performance and density. In: ACM Symposium on FPGAs, FPGA 2000, pp. 3–12 (2000)
24. Lewis, D., Ahmed, E., Baeckler, G., Betz, V., Bourgeault, M., Casshman, D., Galoway, D., Hutton, M., Lane, C., Lee, A., Leventis, P., Marquardt, S., McClintock, C., Padalia, K., Pedersen, B., Powell, G., Ratchev, B., Reddy, S., Sghleicher, J., Stevens, K., Yuan, R., Cliff, R., Rose, J.: The Stratix II logic and routing architecture. In: ACM Symposium on FPGAs, FPGA 2005, pp. 14–20 (2005)
25. Yau, H.T., Pu, Y.C., Li, S.C.: An FPGA-based PID controller design for chaos synchronization by evolutionary programming. Discrete Dyn. Nat. Soc. **2011**, 1–11 (2011)
26. Atoche, A.C., Perales, G.S., Gamboa, A.M., Enseñat, R.A.: Synchronization of chaotic systems: field programable gate array and nonlinear control feedback approach. In: IBERCHIP-2006 (2006)
27. Rajagopal, K., Guessas, L., Vaidyanathan, S., Karthikeyan, A., Srinivasan, A.: Dynamical analysis and FPGA implementation of a novel hyperchaotic system and its synchronization using adaptive sliding mode control and genetically optimized PID control. Math. Prob. Eng. **2017,** Article ID 7307452, 14 p. (2017)
28. Karthikeyan, R., Prasina, A., Babu, R., Raghavendran, S.: FPGA implementation of novel synchronization methodology for a new chaotic system. Indian J. Sci. Technol. **8**, 2 (2015)
29. Vaidyanathan, S., Volos, C.: Advances and Applications in Chaotic Systems. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-30279-9
30. Muthuswamy, B., Banerjee, S.: A Route to Chaos Using FPGAs: Volume I: Experimental Observations. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18105-9
31. Park, J., Sung, W.: FPGA based implementation of deep neural networks using on-chip memory only. In: ICASSP 2016 (2016)
32. Cuevas-Arteaga, B., Dominguez-Morales, J.P., Rostro-Gonzalez, H., Espinal, A., Jimenez-Fernandez, A.F., Gomez-Rodriguez, F., Linares-Barranco, A.: A SpiNNaker application: design, implementation and validation of SCPGs. In: Rojas, I., Joya, G., Catala, A. (eds.) IWANN 2017. LNCS, vol. 10305, pp. 548–559. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59153-7_47
33. WHO: World Health Statistics 2017: Monitoring health for the SDGs. http://www.who.int/gho/publications/world_health_statistics/2017/en/. Accessed 20 June 2017
34. Cassidy, A., Andreou, A.G.: Dynamical digital silicon neurons. In: Biomedical Circuits and Systems Conference, BioCAS 2008, pp. 289–292. IEEE (2008)
35. Ambroise, M., Levi, T., Bornat, Y., Saighi, S.: Biorealistic: spiking neural network on FPGA. In: 2013 47th Annual Conference on Information Sciences and Systems (CISS) (2013)

36. Thomas, D.B., Luk, W.: Biorealistic spiking neural network on FPGA. In: 47th Annual Conference on Information Sciences and Systems, (CISS), pp. 1–6 (2013)
37. Zhu, Q., Song, A., Fei, S., Yang, Y., Cao, Z.: Synchronization control for stochastic neural networks with mixed time-varying delays. Sci. World J. **2014**, Article ID 840185, 10 p. (2014). http://dx.doi.org/10.1155/2014/840185
38. Yue, L., Yixin, Z., Wei, H.: Robust synchronization of uncertain chaotic neural networks with time-varying delay via stochastic sampled-data controller. In: Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE (2016)
39. Abdurahman, A., Hu, C., Muhammadhaji, A., Jiang, H.: Adaptive control strategy for projective synchronization of neural networks. In: Cong, F., Leung, A., Wei, Q. (eds.) ISNN 2017. LNCS, vol. 10261, pp. 253–260. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59072-1_30
40. Park, J.H.: Chaos synchronization of a chaotic system via nonlinear control. Chaos Solitons Fractals **25**, 579–584 (2005)
41. London, M., Roth, A., Beeren, L., Häusser, M., Latham, P.E.: Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. Nature **466**(7302), 123–127 (2010)

http://www.springer.com/978-3-319-71010-5

Computational Neuroscience
First Latin American Workshop, LAWCN 2017, Porto Alegre, Brazil, November 22–24, 2017, Proceedings
Barone, D.A.C.; Teles, E.O.; Brackmann, C.P. (Eds.)
2017, X, 195 p. 77 illus., Softcover
ISBN: 978-3-319-71010-5