ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

FPGA-based implementation of characterization of epileptic patterns in EEG using chaotic descriptors

Walther Carballo, Hugo G. González, David Antonio, Azgad Casiano

Abstract*—In this work an integration of different mathematical and technological tools are presented in order to automatically classify through softwarea sort of preprocessed time series from electroencephalograms (EEG) obtained from healthy patients and patients suffering epilepsy. Time series are analyzed using different chaotic descriptors like the Largest Lyapunov Exponent (LLE) and the Correlation Dimension (Dg). These descriptors are used to describe the different patterns in EEG for a diagnosis of the current status patients. Once the descriptors are obtained these descriptive algorithms are embedded into a multicore Field Programmable Gate Array (FPGA)Zynq ZC702® from Xilinx®, once the algorithm is tested in software.

***Index Terms*— Epilepsy, FPGA, EEG, chaotic descriptors, Largest Lyapunov Exponent (LLE), Correlation Dimension (Dg), MPSOC, Hardware Design Level (HDL), False nearest neighbor (FNN), embedded dimension, chaotic attractors, probabilistic neural network (PNN), decision trees, Bayesian network and data mining.**

I. INTRODUCTION

The identification of epilepsy disorder using *EEGs* as an indicator of the pattern created because of the deficit of neurotransmitters in the exchange of ions in bioelectrical connections is a difficult task even for the trained eye of the medical professionals in the diagnosis stage of the disorder. Therefore further analyses are required in order to successfully classify biosignals of patients who suffer from it and those who do not. However, with the constant evolution and micro-integration of transistors and electronic devices,it is possible to implement powerful technologies that help us to do the most difficult processing in short time intervals. In the same way some software architectures [1], like real time operative systems and multitasking computing, make the task easier to realize because of the increase of the power and computing speed [6] [7].

In this document, an implementation of a previously tested work based on chaotic descriptors using the *EEG* for a future classification with *PNN* [2] and [3], Decision trees [10] and Bayesian Networks [11] also proposed in these papers is proposed. The implementation was realized using an electronic platform multicore *FPGA* with instances executed with parallel computing techniques [2]. The main objective was to show a stable approach for an automatic classification of the epilepsy stages in patients as fast and accurate as possible.

II. THEORETICAL BACKGROUND AND VALIDATION

A. Methodology

The methodology proposed in [2] is a good point to start the software development of the algorithm. However this methodology was adjusted using some more modern tools and improved so that it can be embedded into an electronic interface. The procedure of the implementation from section III follows the diagram in Fig. 1.

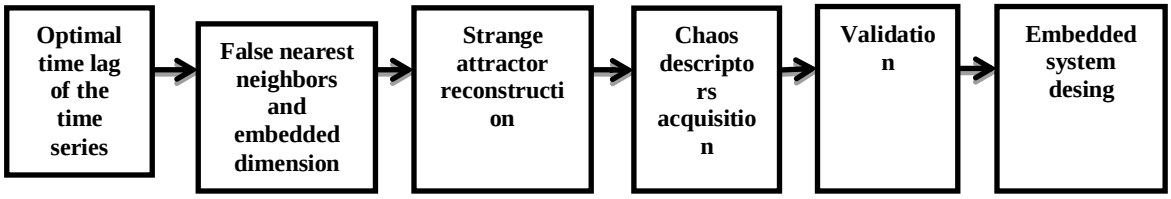


Fig. 1 Block diagram of software development and testing.

B. Chaotic descriptors

One of the main stages of the algorithm is the reconstruction of the epileptic time seriesas it is used to obtain a

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

visual representation of the dynamical system using phase states in an*n*-dimensional Euclidian space. This approximation can show some information from the signals. In most of the cases the validation of the algorithms is done in comparison against the results of well-known dynamical systems. Previously the algorithms were tested before they were implemented in the Linux-based hardware platform based on [5] [9].

C. Attractor reconstruction using delayed coordinates

Before the reconstruction of the attractors based on the time series from the dynamical systems a time delay has to be done in order to decrease the correlation of the signals,so more information of the different dimensions can be obtained.

There are some methods that propose different approaches in order to obtain the delayed time series. The Whitney's embedding theorem states any manifold in m-dimension can be embedded in a 2m-dimension, while Taken's theorem states that the embedding dimension [4] is obtained using multiples of the time-delay*([y(t),y(t−τ),y(t−2τ),…,y(t−(n-1)τ)]).* However in this work some statistical and probabilistic information is obtained from the time series and the correlation is evaluated using the mutual information. The optimal delay is obtained when the correlation from both, the original time series and the delayed time series, reaches its first minimum value.

The mutual information is obtained using (1).

$$I(X,Y) = \sum_{y\in Y}\sum_{x\in X} P(x,y)\log\left(\frac{P(x,y)}{P(x)\,P(}\right. \qquad (1)$$

where*x* is the original time series, *y* is the delayed time series, *P(x,y)* is the joint probability distribution and *P(x)* and *P(y)* are the marginal probabilities. This method iterates the sum in*I(X,Y)* until the first minimum is reached and τ is obtained as seen in (2).

$$\tau = argmin\left[I\left(x(t), y(t\right)\right. \qquad (2)$$

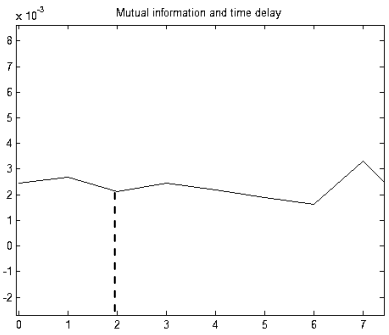


Fig. 2 First minimum value in mutual information of a time series.

After the first optimal delay is obtained the embedding dimension is computed in order to obtain the reconstructed attractor in the space state representation. In terms of the algorithm development both the time delay and the embedding dimension can be computed simultaneously. The method to identify which embedding dimension is selected depends in the number of the false nearest neighbors. This is computed using Kennel's method [4] to obtain the percentage of FNN with (3).

$$R_i = \frac{|R_{i+1} - R_{j+1}|}{||\overrightarrow{R_i} - \overrightarrow{R_j}||} \qquad (3)$$

whereR_{i+1} is a point in the attractor, R_{j+1} is a neighbor in the attractor and the $||\overrightarrow{R_i} - \overrightarrow{R_j}||$ is the distance between these two points in the attractor. If R_i is greater than a heuristic based in the absolute and relative tolerances then these two points are false neighbors as indicated by (4).

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

$$R_i > h(atol, rtol) \rightarrow \%FVC = FVC(R_i) \quad (4)$$

If the number of false nearest neighbors is 0 or low enough then the embedding dimension is taken where the neighbors reached this value in (5).

$$m = \arg(\%FVC(R_i) \approx 0) \quad (5)$$

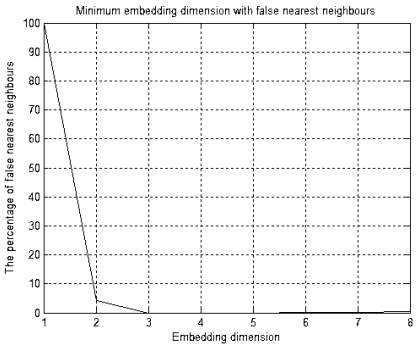


Fig. 3*%FNN* vs embedding dimension.

Using both parameters a reconstruction of the strange attractor can be made using the original time series with no delay, a second time series delayed by τ and a third time series with a delay of 2τ in order to plot a 3D space state representation. However if the parameters of embedding dimension or time delay in (6) are not optimal the attractor will not be fully expanded.

$$x(t), x_1(t-\tau), x_2(t-2\tau), x_3(t-3\tau) \ldots x_{m-1}(t-(m-1)\tau) \quad (6)$$

On the other hand if the embedded dimension and the time delay are chosen with the optimal parameters, the strange attractor will be completely expanded.

D. LLE and Dg computed using Rosenstein's algorithm for small time series

Following Grassberger- Procaccia's algorithm [8] the dimension correlation is calculated creating a sphere of dimension *m* and a variable radius *R* in some point of the strange attractor and counting the number of points of the neighbors inside the *m*-dimensional sphere. This can be calculated using (7) and (8).

$$C(R) = \lim_{N\to\infty}\left[\frac{2}{N(N-1)}\sum_{i,j=1}^{N} H\left(R - \|x_i - x_j\|\right)\right],\ i \neq j \quad (7)$$

$$C(R) \sim R^{dg} \quad R \rightarrow 0 \quad (8)$$

where $H(x)$ is the Heaviside function (1 if $x \geq 0$ and 0 otherwise) $\|x_i - x_j\|$ is the Euclidian norm (distance between neighbors) and *N* is the number of points in the attractor. *Dg* is calculated as a logarithmic ratio between $C(R)$ and R. Using a linear approximation the pendant of the line estimates the *Dg* value.

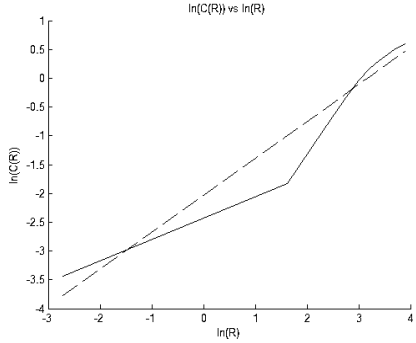


Fig. 4 Linear approximation (segmented line) of the curve $\ln(C(R))$ and $\ln(R)$ (solid line) to obtain *Dg*.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

The computation of LLE can be done using Rosenstein's algorithm [5] based on getting the divergence from trajectories on an m-dimensional sphere, whose surface is created with the neighborhood of points in the reconstructed attractor. Each point from the surface has a trajectory described by the dynamic system. The divergence (or convergence) of the system is plotted and stabilizes at some point through the time evolution from (9).

$$\lambda_i = \lim_{t\to\infty}\left\{\frac{1}{t} ln\left[\frac{L_i(t)}{L_i(0)}\right]\right\} \qquad (9)$$

where $L_i(0)$ is the initial trajectory and $L_i(t)$ are the divergent trajectories shown in (10). λ_i is the exponent of such trajectory in that dimension. The time series can have many Lyapunov exponents, if one of those is positive it hints some chaotic behavior for that system; however if the sum of all the exponents is negative it means that the system is dissipative and it tends to converge at some point. *LLE* is the highest λ.

$$\epsilon(t) = \epsilon(0)\, e^{(\lambda_1+\lambda_2+\cdots+\lambda_n)t} \qquad (10)$$

When the divergence stabilizes a linear approximation can be done in order to estimate LLE.

III. IMPLEMENTATION

A. *Attractor reconstruction of EEG time series from the database of the Bonn University Hospital, Germany*

Using the previously described algorithms, the analysis of the *EEG* with the database of the hospital in Bonn can be done. The first step is to analyze the attractor reconstruction of the time series from the 5 sets of patients (Z, O, F, N and S). Each sample data has 100 time series of 4097 points, frequency of 173 Hz (23.6 seconds). A band-pass filter from 0 Hz to 60Hz was required in order to avoid instrumentation noise from the data acquisition device. The sets Z and O were acquired from patients with no previous diagnosis of epilepsy with open and closed eyes respectively. The sets F and N are from patients with a previous diagnosis of epilepsy with no attacks at the moment of the acquisition with open and closed eyes respectively. Finally the set S was acquired from patients with a previous diagnosis of epilepsy during a seizure [11].

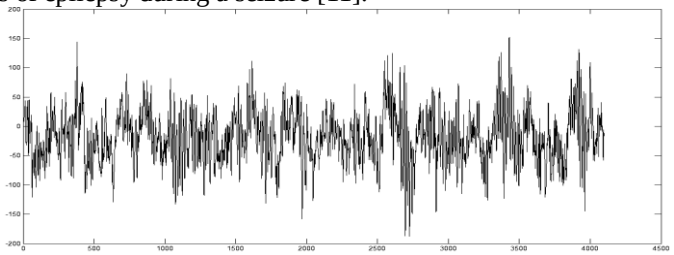


Fig. 5 Sample 1 of cluster Z

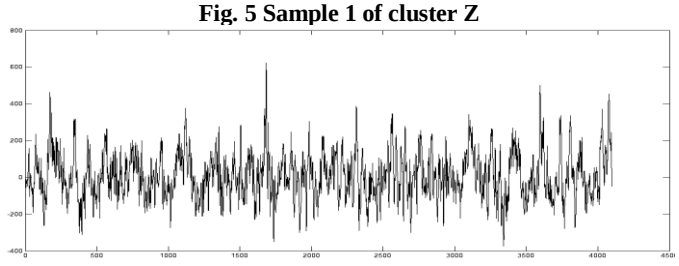


Fig. 6 Sample 1 of cluster F

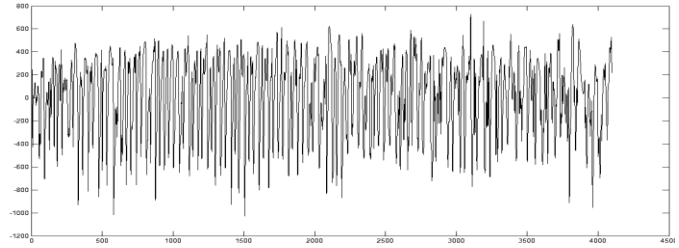


Fig. 7 Sample 1 of cluster S

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

B. Software results

Using the attractor reconstruction algorithm described before, the strange attractors in Fig. 8, 9 and 10 are obtained.

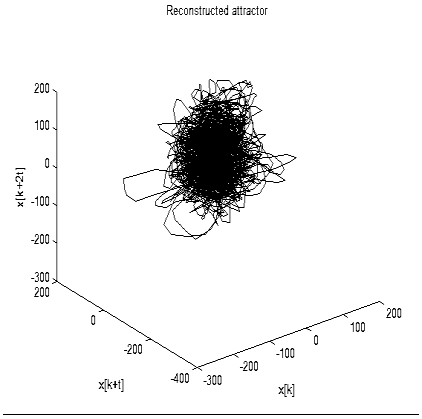


Fig. 8 Example of reconstructed attractor from cluster Z with m=15 and τ=11

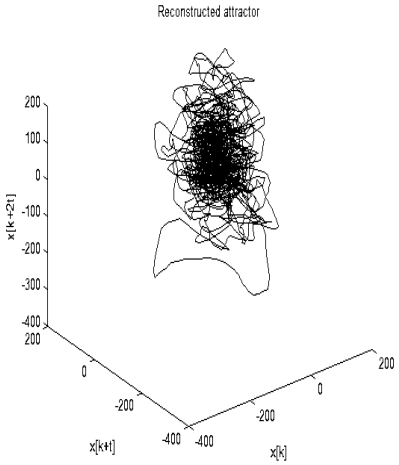


Fig. 9 Example of reconstructed attractor from cluster F with m=7 and τ=13

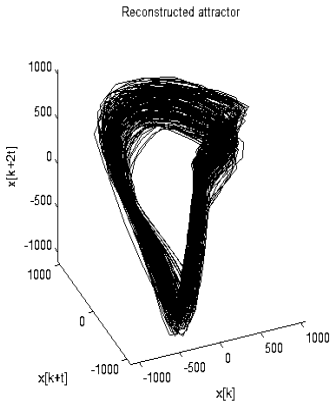


Fig. 10 Example of reconstructed attractor from cluster F with m=6 and τ=3

Computing the Dg and LLE of every sample of each cluster a collection of data was created showing some of the results from table I.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

#	Mutual Info.	Tau (Time lag)	FNN	m	Dg	LLE	Cluster
1	0.0019	4	0.0000	13	4.96	0.3142	F
2	0.0000	14	0.0279	14	3.38	0.0760	F
3	0.0020	2	0.0270	6	2.92	0.3309	F
…	…	…	…	…	…	…	…
1	0.0017	1	0.0000	6	2.38	0.0297	S
2	0.0020	1	0.0000	6	2.37	0.1709	S
3	0.0014	3	8.3045	8	3.42	0.1696	S
…	…	…	…	…	…	…	…
1	0.0018	5	0.0261	14	1.37	0.3321	Z
2	0.0022	1	0.0000	5	1.05	0.0392	Z
3	0.0015	1	0.0000	6	1.21	0.0225	Z
…	…	…	…	…	…	…	…

Table I some results from the algorithms described above.

Note that most *LLE* are positive which suggests a chaotic behavior from the signals. Using the software platform for data mining analysis *Weka*® many classification methods like *Decision trees* using *C4.0, C5.0 and Bayesian networks* with *Naïve Bayes* were developed in order to obtain better performance in the classification. The *J48* is based in the association and pruning rules described in [10].The *Bayesian Network* assumes completely independency between the classifications attributes and estimates probability tables given the values of such parameter [10]. In Fig. 11 and Fig. 12 the *J48*tree and naïve Bayes model were implemented respectively.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

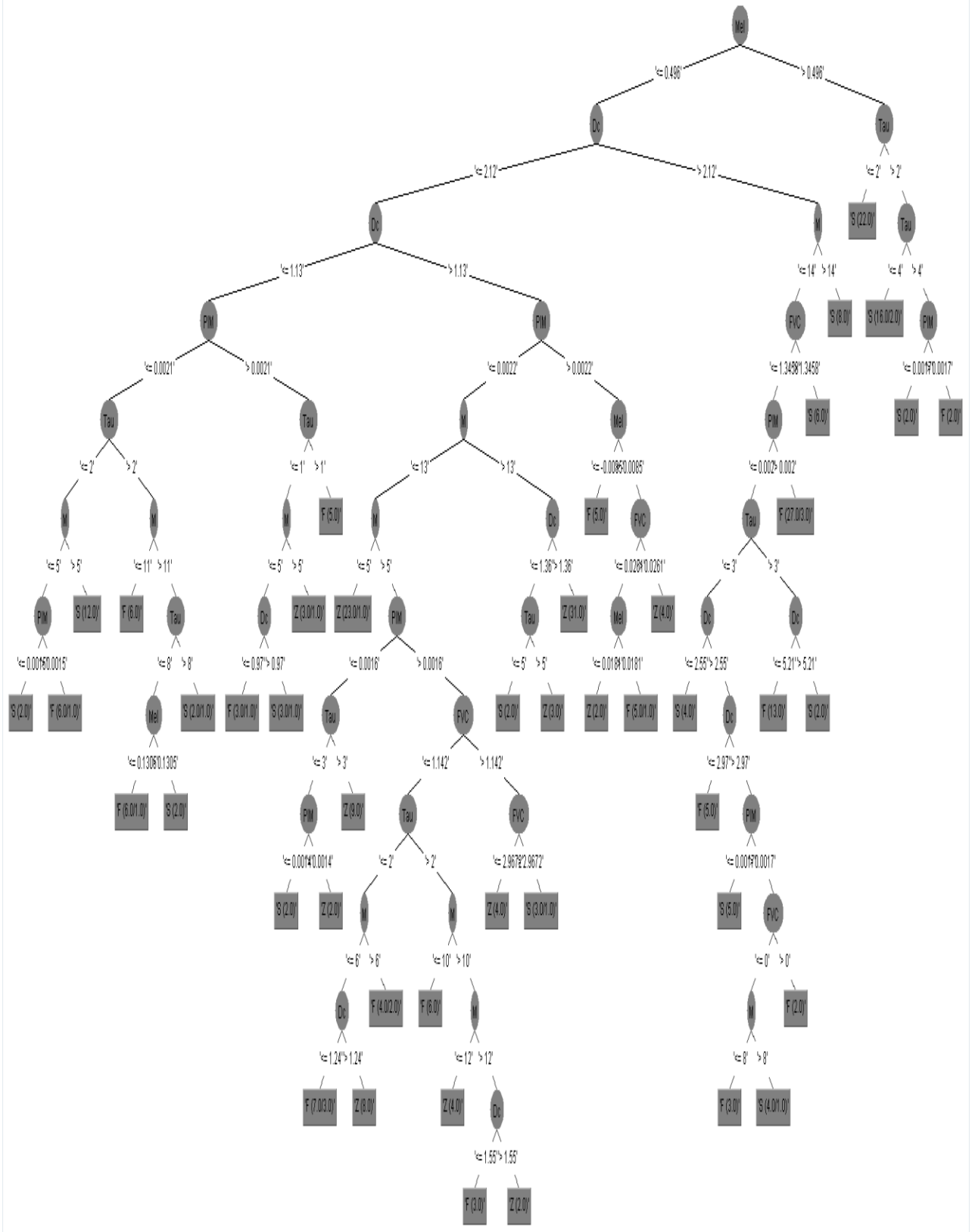


Fig. 11 Decision tree using *J48* algorithm on *Weka*®.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

Where *PIM* means mutual information, τ is the time delay, *FVC* means % of k-nearest neighbors embedding dimension, *Dc* is the dimension correlation and *Mel* is the largest Lyapunov Exponent.

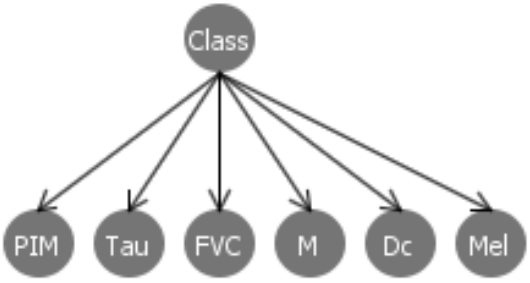


Fig. 12 Naïve Bayes model using *Weka*® from left to right mutual information, time delay, % of k-nearest neighbors, embedding dimension, *Dg* and *LLE*.

The percentage of correct classified instances from the decision tree was 94% while the Bayesian model had 76%. Both models can be easily implemented using basic comparisons and taking advantage of the probability matrix respectively.

Real set / Classified as set	F	S	Z
F	96	3	1
S	8	91	1
Z	4	3	93

Table II Confusion matrix of Decision tree *J48* algorithm.

Real set / Classified as set	F	S	Z
F	60	16	24
S	26	64	10
Z	3	2	95

Table III Confusion matrix of *Naïve Bayes* algorithm.

Table II and Table III show the results of the classification. True positives and true negatives can be seen as a diagonal in the confusion matrix, while the errors false positives and false negatives are both outside of the diagonal.

C. Embedded design with MATLAB®, Simulink®, Embedded Coder®, HDL Coder® and Xilinx Zynq ZC702®.

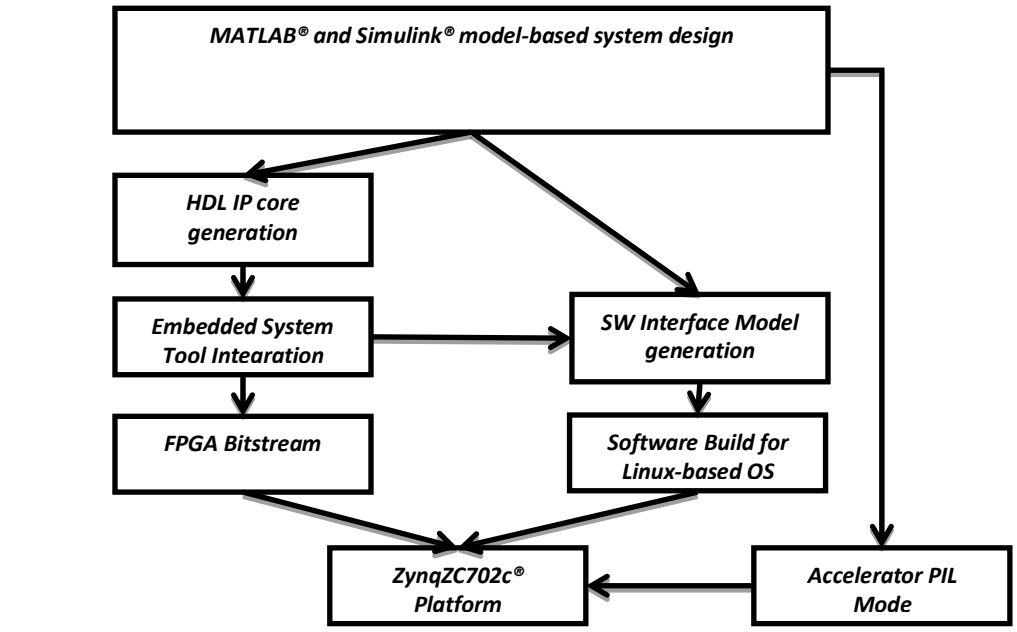
Based on the algorithms of the chaotic descriptors, tools and techniques from different platforms the whole system can be embedded on a *FPGA* device with 2 ARM-Cortex9 cores of 800 MHz using *Simulink*® and the Embedded Coder® and *HDL* Coder®toolboxes.The former generates automatically *C-code* for the Linux-based operative system using the two cores of the device running the model on software. The latter generates *HDL* code for the programmable logic running entirely on hardware. The diagram in Fig. 13 explains how the target device assigns workflows on the different parts of the *FPGA* based on *Xilinx Zynq ZC702*® architecture.

The model was created on Simulink® importing all the functions from MATLAB®. All was embedded into two charts. The first one works as a buffer for memory allocation for the time series from the workspace on MATLAB®, this way the time series can be processed as a whole. This block also works as a controller triggering the processing of the time series with the algorithms already described.

In Fig. 14 the time series was imported from the MATLAB® workspace and on each rising pulse from the pulse generator (period of 2 samples with sampling time of 0.01 times the simulation time) the data will be stored on the Buffer. Because the variable “serie” and the pulse generator are not synchronized the buffer must handle asynchronous data acquisition using the time stamp of the variable “serie” with simulation time. As seen in Fig. 15 when the Buffer reaches the size of the buffer the trigger is set to one and the signal processing starts.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

Fig. 13 MATLAB® and Simulink® with support with Zynq platform based on [12].



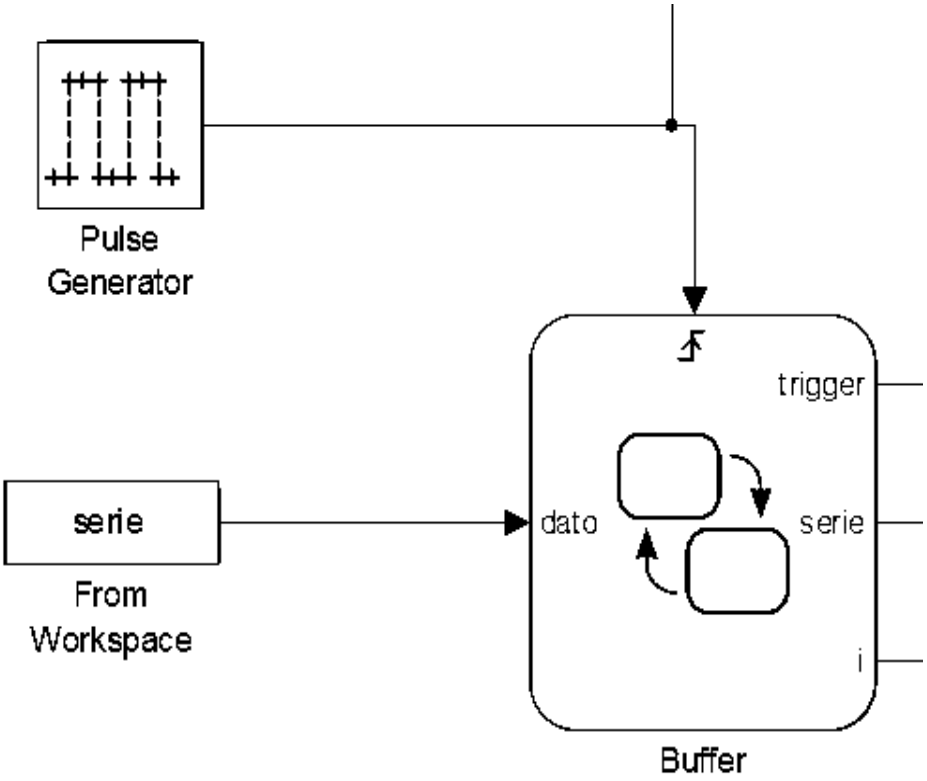


Fig. 14 Chart with states and obtaining time series from workspace.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

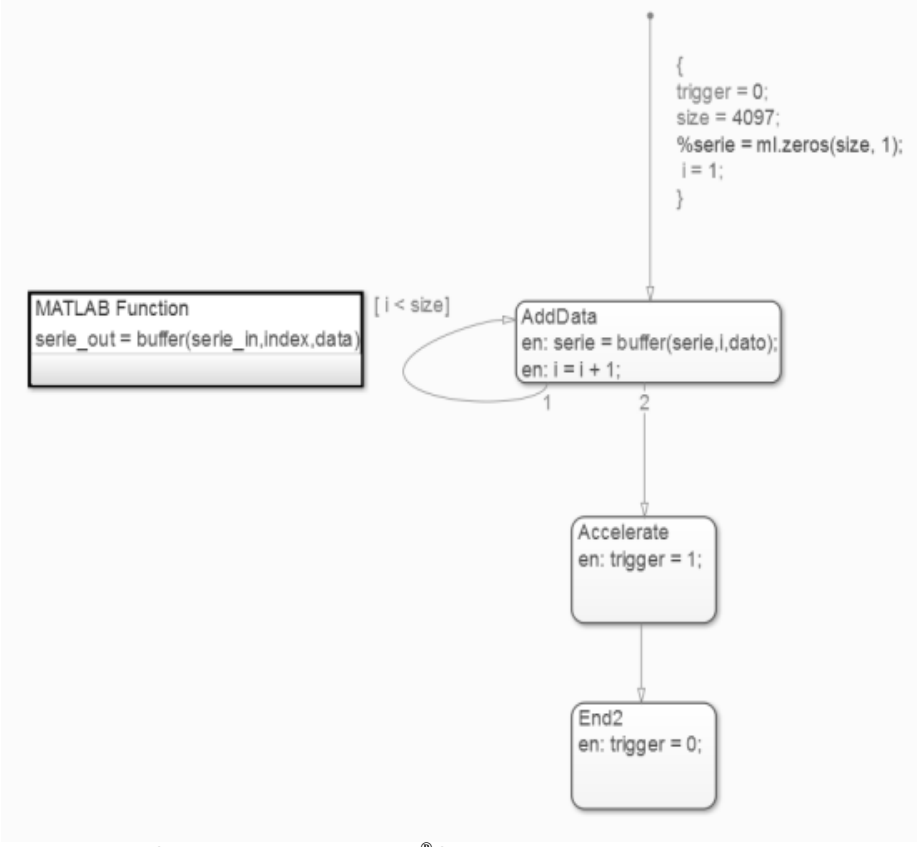


Fig. 15 State flow chart with MATLAB® function embedded and triggering controller.

As seen in Fig. 15, the *trigger* variable starts the execution of a stateless chart, whose main objective is generating the attractor reconstruction algorithm. The connections can be seen in Fig. 16.

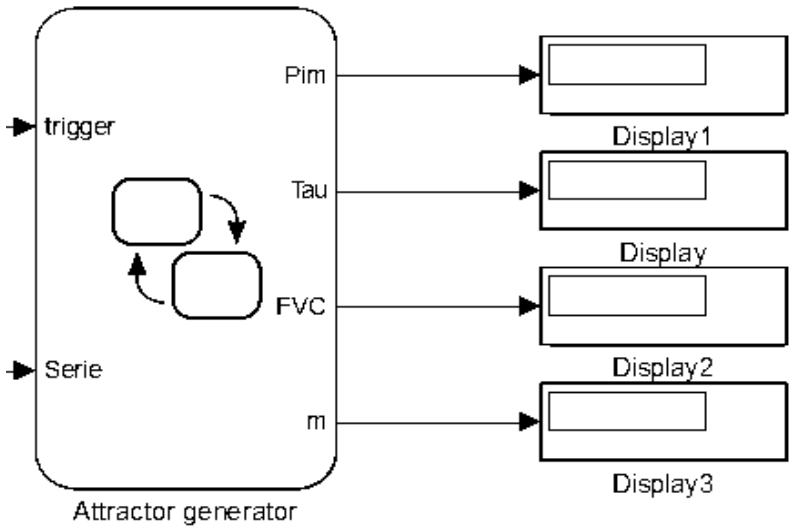


Fig. 16 Attractor reconstruction chart with 4 outputs for classification

The chart is driven by stateless operations and the execution of two MATLAB® functions triggered when the buffer is ready. The process can be seen in the Fig. 17.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

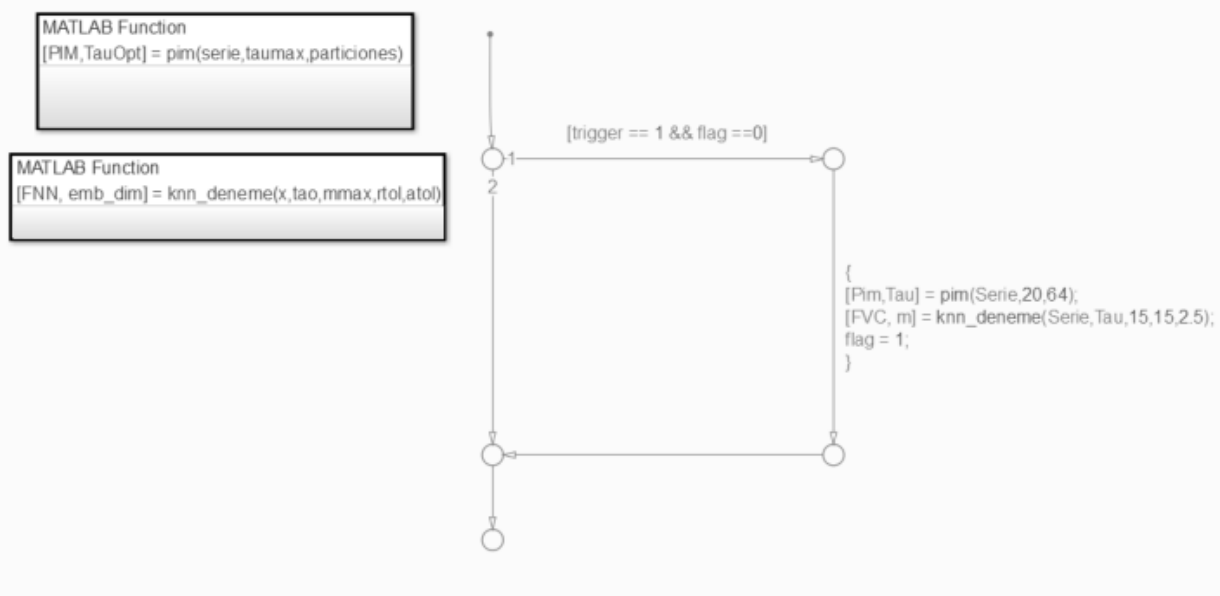


Fig. 17 Stateless chart delay coordinates and embedding dimension.

IV. CONCLUSIONS AND FUTURE WORK

In this approach for an automatic classification a prototype was created in order to generate strange attractors of the signals obtained from the database from the Bonn University Hospital. The main objective was to implement an embedded system design taking into consideration the architecture of the device *Zynq ZC702*® using the algorithms of the chaotic descriptors of *EEG* time series from patients who suffer from epilepsy. With these results the next step will be to embed the chaos descriptors (*LLE* and *Dg*) algorithms on the Xilinx® platform and create a hardware model in the programmable logic based on the results or the classifiers from *Weka*® and the *PNN* from previous work [2]for an atomic classifier. The *Zynq*® results matches so far the results from *MATLAB*® and Simulink®, however some optimization may be required at the end of the embedding system. This means adding some directives or objectives to make the processing faster, allocating more wisely some memory in the algorithms and probably transferring some processing between the SW interface to the *HDL IP* core.

REFERENCES

[1] D. Antonio Torres and D. Soler Delgado, «Evaluation of MPSoC Operating Systems,» International Journal of Engineering Practical Research, vol. 2, nº 2, pp. 49-54, 2013.

[2] H. G. González Hernández and A. Casiano Ramos, «Automatic classification of EEG epileptic patterns using chaotic descriptors» fromCongreso Nacional de Control Automático, 2005.

[3] Azgad Casiano Ramos and Hugo G. González Hernández. <<Extracting dynamical invariantsfrom EEG time series in epilepsy>>.Memorias del primer Congreso de Control Aplicado a Ciencias Biomédicas (CACIB), 2005.

[4] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, <<Determining embedding dimension for phase-space reconstruction using a geometrical construction>>, Phys. Rev. A 45, 3403 (1992).

[5] M.T. Rosenstein, J.J. Collins, C.J. De Luca. <<A practical method for calculating largest Lyapunov exponents from small data sets>> Physica D 65:117-134, 1993.

[6] Hekim, M., Ozer, M., & Provaznik, I. (2011). <<Epilepsy diagnosis using probability density functions of EEG signals>>., (pp. 626 - 630). Istanbul.

[7] Khobragade, P., & Panda, R. (2011). <<Expert system design for classification of brain waves and epileptic-seizure detection>>. Students' Technology Symposium (TechSym), 2011 IEEE, (pp. 187-192).

[8] Baker, G. L., & Gollub, J. P. (1996). <<Chaotic dynamics an introduction>> (pag. 109-132). Second edition. Cambridge university press.

[9] Wolf, A., Swift, J.B., Swinney, H.L., & Vastano, J.A. (1985). <<Determining Lyapunov exponents from a time series>> Physica, (pp 273-290).

[10] H. Witten, Ian. Frank, Eibe. Hall,& Mark A.(2011) <<Data Mining. Practical Machine Learning Tools and Techniques>>. Third edition. Chapters 6,7 and 8 .Morgan Kaufmann series in data management systems. Elsevier.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 3, May 2015

[11] K. Lehnertz. <<Klinik für Epileptologie Bonn Universität: Neurophysik/Vorlesungen>> Obtained from http://epileptologie-bonn.de/cms/front_content.php?idcat=203 on September 2013.

[12] Mathworks <<Motor control design with Simulink>> Obtained from http://www.mathworks.com/discovery/motor-control.html on April 2014.