Eur. Phys. J. Special Topics **225**, 119–126 (2016)

DOI: 10.1140/epjst/e2016-02609-8

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

Robust hyperchaotic synchronization via analog transmission line

S. Sadoudi[1,a] and C. Tanougast[2]

[1] Laboratoire Systèmes de Communication, Ecole Militaire Polytechnique, Algiers, Algeria
[2] LCOMS, Université de Lorraine, Metz, France

Received 29 July 2015 / Received in final form 29 January 2016
Published online 29 February 2016

Abstract. In this paper, a novel experimental chaotic synchronization technique via analog transmission is discussed. We demonstrate through Field-Programmable Gate Array (FPGA) implementation design the robust synchronization of two embedded hyperchaotic Lorenz generators interconnected with an analog transmission line. The basic idea of this work consists in combining a numerical generation of chaos and transmitting it with an analog signal. The numerical chaos allows to overcome the callback parameter mismatch problem and the analog transmission offers robust data security. As application, this technique can be applied to all families of chaotic systems including time-delayed chaotic systems.

1 Introduction

Since the pioneering work of Pecora and Carroll [1] about the first chaotic synchronization technique, many efforts have been made to show that the synchronization problem of chaotic systems could be solved. In fact, different synchronization schemes have been developed such as inverse system approach [2], linear and nonlinear feedback control [3,4], symbolic dynamics approach [3] and [5], observer design approach [6,7], impulsive synchronization approach [8,9] and recently, the application of complete synchronization technique to synchronize, through FPGA implementation, the Ikeda time-delayed chaotic system [10,11], without taking off the channel noise. But to the best of our knowledge, none of them gives a robust experimental solution to the chaotic synchronization problem. This is due to the fact that these techniques are based on continuous or impulsive synchronization, which is extremely sensitive to small perturbations present in the drive signal (the butterfly effect). For more precision, the drive signal is transmitted and injected continuously (or impulsively) into the dynamics of the chaotic system at the receiver (slave system). In this scheme, the chaotic dynamics of the slave system is perturbed, thus it will generate non identical chaotic signal compared to that of the master. Consequently, the synchronization cannot be achieved. This finding is confirmed recently by Pecora and Carroll in their last research paper [12], where they present details of the history and early work in

[a] e-mail: sadoudi.said@gmail.com

the area of chaotic synchronization. In fact, they affirm that identical synchronization between two chaotic systems in the presence of noise becomes impossible for noise whose amplitude is greater than approximately 10% of the signal amplitude. This is due to the fact that the chaotic response system mixes noise and signal in a nonlinear fashion, making it impossible to separate the two by conventional methods, such as filtering or correlation. Thus, the question is how to ensure experimentally this condition? Or, is it possible to do it? We think that it is very difficult to overcome this problem. Thus, we agree with the proposition of Pecora and Carroll at the end of their paper, where they encouraged to continue research on the problem of chaotic synchronization by exploring larger topic of collective behavior and dynamical patterns in networks [12].

Recently, a new idea for chaotic synchronization has been experimentally explored. This idea consists in avoiding the perturbation of the chaotic dynamics of the slave system with the injection of the received signal. To realize experimentally this interesting solution, the authors in [13] have used the asynchronous serial communication protocol to synchronize two hyperchaotic systems implemented separately in two Xilinx FPGA platforms. Initially, this technique has been tested and validated through experimental realization of wireless hyperchaotic communication system based on the ZigBee communication protocol [14]. The only inconvenient of this technique is an eventual error in the starting bit, which is under control in the actual communication system.

In this paper, we propose for the first time a robust hyperchaotic synchronization via an analog transmission line between two Lorenz hyperchaotic systems [15], implemented separately in Xilinx FPGA platforms [16]. Precisely, we exploit the idea based on avoiding the perturbation of the chaotic dynamics of the slave system and the power of the hardware FPGA implementation of chaotic system, which overcomes the parameter mismatch problem, to synchronize embedded hyperchaotic system interconnected by an analog transmission line. For this, the Digital to Analog Converter (DAC) of the FPGA platform is used to convert the hyperchaotic samples to an analog signal. This signal is then transmitted to the receiver through analog line. At the receiver, an analog to digital conversion is used and at the obtaining of the first hyperchaotic sample, a control signal triggers the slave system to generate hyperchaotic samples identical to those of the master system under the same parameters and initial conditions values. By controlling the sampling time operation efficiently at the receiver, we can obtain a perfect synchronization. The rest of this paper is organized as follows. Section 2 presents the proposed approach. In Sect. 3, we detail the hardware architecture of the transmitter and receiver implemented on FPGA technology with the obtained synthesis results. The obtained real-time results are presented in Sect. 4. Section 5 gives the conclusion.

2 Proposed hyperchaotic synchronization scheme

The proposed analog hyperchaotic synchronization technique is presented by the scheme of Fig. 1. The basic idea consists in associating the analog transmission mode of information with the digital hardware implementation of chaotic (or hyperchaotic) sources. For that reason, we name this technique Chaotic Synchronization via Analog Transmission (CS-AT). To realize this idea or this association, we must use a Digital to Analog Converter (DAC) and an Analog to Digital Converter (ADC). In this work, we use the hyperchaotic Lorenz system modeled by the following equations system:

$$\begin{cases} \dot{x} = a(y-x) \\ \dot{y} = x(b-z) - y + w \\ \dot{z} = xy - cz \\ \dot{w} = -dx. \end{cases} \tag{1}$$

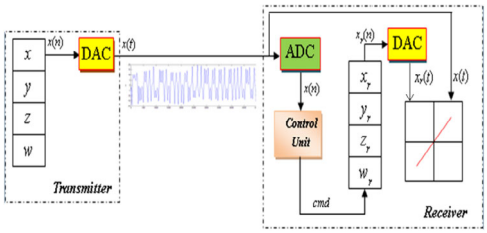

Fig. 1. Principle scheme of the hyperchaotic synchronization via analog transmission.

At the transmitter, as shown in Fig. 1, a DAC is used to convert the generated chaotic samples $x(n)$, of the master system, with a resolution of $(n+1)$ bits. The obtained analog chaotic signal $x(t)$ is then transmitted to the receiver via analog mode.

At the receiver, an ADC is used to convert the received analog signal to chaotic samples $x(n)$ and at the end of this operation, a control unit generates a command signal cmd which triggers and controls the generation process of the hyperchaotic generator of the receiver (slave system). For more precision, through the control signal (cmd), the slave system generates automatically its chaotic (or hyperchaotic) signals (x_r, y_r, z_r and w_r), which are evidently identical to those of the master system because they are generated with the same parameters and initial values via digital hardware implementation. At the end, a DAC is used to convert the samples $x_r(n)$ to analog signal x_r and then a synchronization block is used to synchronize the drive signal $x(t)$ with the generated signal $x_r(t)$, by viewing them simultaneously on oscilloscope screen.

As you can see in the principle scheme of this technique, whatever the perturbations present in the drive signal caused by channel noise, the chaotic dynamics of the slave system will not be perturbed. Thus, the chaotic synchronization will be ensured effectively. In addition, note that this technique can be applied for all families of chaotic systems such as continuous 3 dimensional, hyperchaotic, discrete, time-delayed chaotic systems etc. This affirmation is confirmed experimentally in the next section, where more details are also given.

3 Hardware FPGA implementation

To test and validate experimentally the proposed solution, we have implemented separately two hyperchaotic Lorenz systems in Xilinx Virtex-II Pro XC2VP30 platforms [16]. We have interconnected the two platforms with an analog line (or wire) as it is shown by the photo of the experimental design presented in Fig. 2. This analog connection is realized through the inputs/outputs of the CODEC AC97 of each FPGA platform [16]. Thus, the real-time results of the analog chaotic synchronization are viewed on oscilloscope screens.

3.1 Transmitter architecture

The schematic of the implemented transmitter architecture is presented in Fig. 3. It is composed mainly of three modules, a clock generator (div_clk), the *lorenz* module and the codec AC97 module which realize the digital to analog conversion.

- ***div_clk***: This module generates and provides the clock (div_clk) and *reset* signals to the two others modules. The frequency of the clock signal is imposed by the

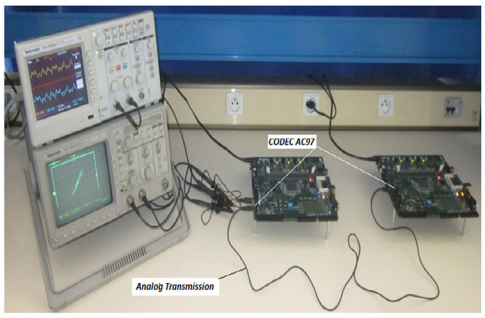

Fig. 2. Experimental device.

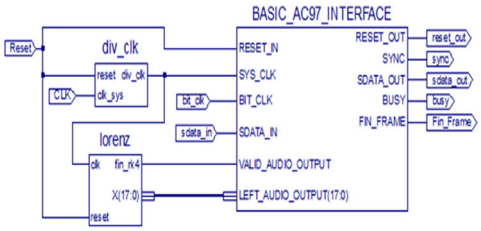

Fig. 3. Hardware architecture of the transmitter.

Table 1. Synthesis Results of the Transmitter/Selected device: XC2vp30-7ff896.

Logic Utilization	Available	Used	(%)
Number of Slices	13696	2661	19
Number of Slice Flip Flops	27392	1211	4
Number of 4 input LUTs	27392	4267	15
Number of bonded IOBs	556	8	1
Number of MULT18x18s	136	36	26
Number of GLKs	16	3	18
Maximum Frequency (MHz)	25.318		

maximum operating frequency of the codec AC97 which is 12.5 MHz. To obtain the desired frequency, we use in this module a clock divider. Knowing that the global clock signal *clk_sys* of the Xilinx Virtex-2 platform is 100 MHz [16].

- ***Lorenz***: This module represents the main module of the proposed architecture. It is based on the hardware architecture detailed and presented in [14] in which an optimal VHDL hardware description of the RK-4 method based on a Moore finite state machine is used. This module generates the hyperchaotic samples X encoded on 18-bit. This signal is then transmitted to the codec under the command signal *fin_rk4*.
- ***BASIC_AC97_INTERFACE***: This module converts the generated hyperchaotic samples to analog hyperchaotic signal under the command signal *fin_rk4* at the frequency of 12.5 MHz. In fact, when the data is available at the output of the *lorenz* module, the signal *fin_rk4* is set to "1". Then, the digital to analog conversion operation begins. The Lorenz module waits the end of this operation to set *fin_rk4* to "0" and generates the next sample.

The synthesis results after place and route in terms of logic resources and performance analysis of the proposed architecture implementation inner the Xilinx FPGA Virtex-2 are detailed in the Table 1. The maximum frequency and the hardware

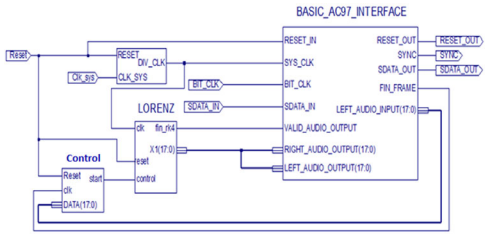

Fig. 4. Hardware architecture of the receiver.

resources consumption are specified. It can be stated that an attractive tradeoff between high speed and low logic resources is achieved. Indeed, our implementation of the transmitter on a Xilinx Virtex-2 platform uses only 2661 slice registers (19%), 36 (26%) of multipliers and no block RAMs under the maximum frequency of 25.318 MHz.

3.2 Receiver architecture

The schematic of the receiver hardware architecture is presented in Fig. 4. Contrary to that of the transmitter, it is composed of four modules, a clock generator, hyperchaotic Lorenz generator module which is identical to that of the transmitter, a control module (*Control*) and the codec AC97 (*BASIC_AC97_INTERFACE*) module. Each module operates as follow:

- ***div_clk***: This module generates the same clock (*div_clk*) as that of the transmitter (12.5 MHz) and reset signals to the two others modules.
- ***BASIC_AC97_INTERFACE***: At the receiver, this module receives the analog hyperchaotic (or chaotic) signal through its input (*LEFT_AUDIO_INPUT*). Then, it converts the analog signal to hyperchaotic samples and sends them to the Control module (Fig. 4). In addition, in this work, this module is used to view the synchronization line and the analog hyperchaotic attractors of the master and slave systems.
- ***Control***: At the end of the analog to digital conversion of each hyperchaotic samples (Fig. 4), this module use a command signal (*start*) to trigger the generation of the identical hyperchaotic sample at the *LORENZ* module. In fact, this command signal is set to "1" when the converted data is available else it is set to "0". This module is the basic element of the proposed solution. The detail of the functioning of this module is presented by the flowchart of Fig. 5.
- ***LORENZ***: This module contains the same hardware architecture of the hyperchaotic Lorenz system implemented at the transmitter. Under the command of the *Control* module, it generates an identical hyperchaotic samples X encoded on 18-bit. These samples are sent to the codec AC97 for digital to analog conversion. At the end of this operation, the analog hyperchaotic signals will be viewed on the oscilloscope screen under the command signal *fin_rk4*. The detail of the functioning of this module is presented by the flowchart of Fig. 6.

The synthesis results after place and route in terms of logic resources and performance analysis of the proposed receiver architecture implementation inner the Xilinx FPGA Virtex-II are depicted in the Table 2. As the results of Table 1, it can be noted that also an attractive tradeoff between high speed and low logic resources is achieved. Indeed, the implementation consumes slightly more logic sources than the transmitter architecture (Table 1), 2664 slice registers, 36 multipliers and no block RAMs under the same value of maximum frequency of 25.318 MHz.

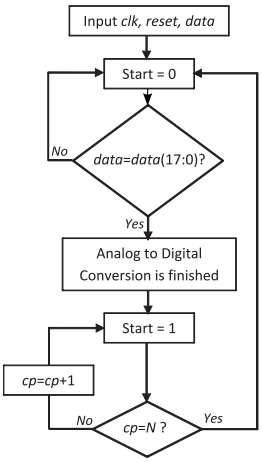

Fig. 5. Flowchart of the *Control* module.

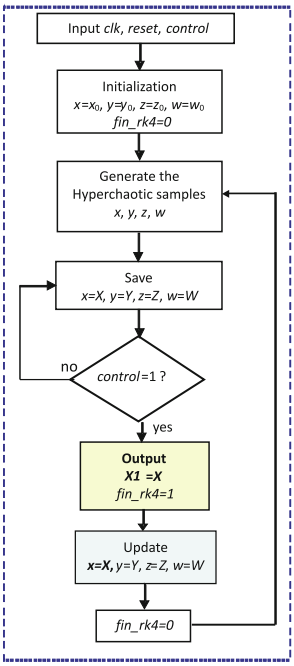

Fig. 6. Flowchart of the *LORENZ* module.

Table 2. Synthesis Results of the Receiver/Selected device: XC2vp30-7ff896.

Logic Utilization	Available	Used	(%)
Number of Slices	13696	2664	19
Number of Slice Flip Flops	27392	1198	4
Number of 4 input LUTs	27392	4261	15
Number of bonded IOBs	556	12	2
Number of MULT18x18s	136	36	26
Number of GLKs	16	3	18
Maximum Frequency (MHz)	25.318		

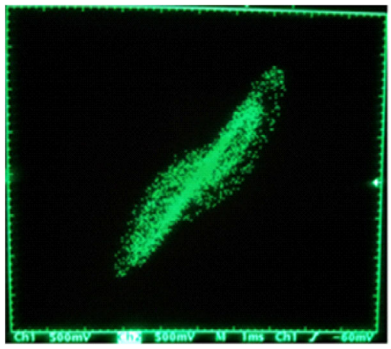

Fig. 7. Real-time 2D hyperchaotic Lorenz Attractor.

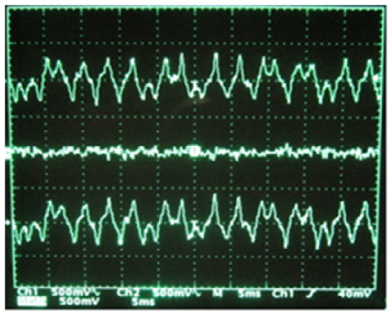

Fig. 8. Transmitted hyperchaotic signal (top), the generated signal at the receiver (bottom) and the synchronization error (middle).

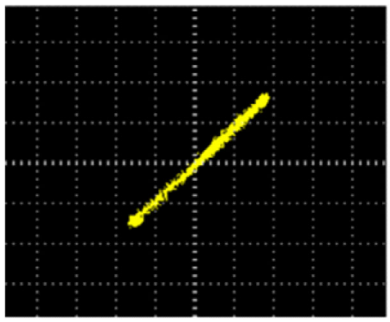

Fig. 9. Real-time synchronization line.

4 Real-time results

In order to test experimentally the proposed analog chaotic synchronization, real-time tests are realized by synchronizing two implemented hyperchaotic Lorenz systems in FPGA platforms interconnected via an analog transmission line (see Fig. 2). The validation of the results is done by displaying the synchronization line, between the received and the locally generated hyperchaotic signals, on digital oscilloscope screen.

The obtained real-time results are presented in Figs. 7, 8 and 9. Figure 6 presents the real-time Lorenz 2D attractor, which confirms that the hyperchaotic dynamic of the slave system is not perturbed, it is conserved and generates a proper chaotic regime. Figure 8 presents the comparison of the received (generated by the master) and the locally generated (by the the slave) signals with the synchronization

error between them which is very small (middle). Figure 9 validates the hyperchaotic synchronization through the obtaining of a perfect synchronization line. Thus, these results validate experimentally the proposed synchronization technique based on an analog transmission line.

Through the experimental tests, we have noted that the quality of the synchronization depends only on the performances of the CODEC AC97 which realizes the analog to digital conversion. In other words, it depends on the sampling time operation. Thus, by using high performed analog to digital converter, the performances of the proposed technique can be improved significantly.

5 Conclusion

This paper presents an experimental validation through FPGA implementation of an interesting chaotic synchronization technique based on analog transmission. The idea consists in avoiding the perturbation of the chaotic dynamics of the slave system. For realizing this idea, a control module is used to trigger the execution of the slave system at the reception of each chaotic sample. The obtained real-time results have shown a very small synchronization error due to the time sampling and thus validate the effectiveness of the proposed synchronization technique. To the best of our knowledge, it is the first time a robust analog chaotic synchronization is validated experimentally between two chaotic generators. The advantage of this technique is its insensitivity to channel noise and to the parameter mismatch problem. The proposed synchronization technique can be applied efficiently to synchronize all families of chaotic systems such as time-delayed chaotic systems, and it is very useful to application in chaos-based speech encryption.

References

1. L.M. Pecora, T.L. Carroll. Phys. Rev. Lett. **64**(8), 821 (1990)
2. U. Feldmann, M. Hasler, W. Schwarz, Int. J. Circuit Theory Appl. **24**, 551 (1996)
3. E.R. Bollt, Intl. J. Bifurcation Chaos **13**, 269 (2003)
4. B.R. Andrievskii, A.L. Fradkov, Automation Remote Control **64**, 673 (2003)
5. G.M. Maggio, G. Galias, IEEE Trans. Circuits Syst.-I **49**, 1729 (2002)
6. T.L. Liao, N.S. Huang, IEEE Trans. Circuits Syst.-II **46**, 1144 (1999)
7. M. L'Hernault, J.P. Barbot, A. Ouslimani, IEEE Trans. Circuits Syst.-I **55**, 614 (2008)
8. T. Yang, Int. J. Computational Cognition **1**, 81 (2004)
9. T. Yang, L.-B. Yang, C.-M. Yang, Phys. Lett. A **226**(6), 349 (1997)
10. B. Muthuswamy, S. Banerjee, *A Route to Chaos Using FPGAs*, Vol. I: Experimental Observations (Springer, Switzerland, 2015)
11. D. Valli, B. Muthuswamy, S. Banerjee, M.R.K. Ariffin, A.W.A. Wahab, K. Ganesan, C.K. Subramaniam, J. Kurths, Eur. Phys. J. Special Topics **223**, 1465 (2014)
12. L.M. Pecora, T.L. Carroll, Chaos **25**, 097611 (2015)
13. S. Sadoudi, C. Tanougast, M.S. Azzaz, Proc, CoDIT'2014, Metz, France (2014)
14. S. Sadoudi, C. Tanougast, M.S. Azzaz, Progr. Electromagnetic Research C **32**, 181 (2012)
15. R. Barboza, Int. J. Bifurcation Chaos **17**, 4285 (2007)
16. *Xilinx University Program Virtex-II Pro Development System* (UG069, April 9, v1.1, 2008)