Progress In Electromagnetics Research C, Vol. 32, 181–196, 2012

FIRST EXPERIMENTAL SOLUTION FOR CHANNEL NOISE SENSIBILITY IN DIGITAL CHAOTIC COMMUNICATIONS

S. Sadoudi[1, *], C. Tanougast[2], and M. S. Azzaz[1]

[1]Ecole Militaire Polytechnique, Bordj el Bahri, BP 17, Algiers, Algeria

[2]Paul Verlaine University of Metz, 7 Rue Marconi, Metz 57070, France

Abstract—An interesting and original solution to the high channel noise sensibility problem of digital chaotic communications is proposed. The solution idea consist of avoiding disruption of the slave/receiver dynamics by injecting the driving signal. To realize experimentally this pertinent idea, an FPGA-based hardware architecture is developed, firstly to trigger the generation of the slave/receiver chaotic dynamics at each received data detection, and secondly to synchronize the driving signal with the slave generated chaotic signal for the demodulation operation. We have tested and validated the proposed solution through experimental realization of a wireless hyperchaotic communication system based on ZigBee communication protocol. Real-time results of experimental wireless communication tests are presented. The obtained results show the effectiveness and the robustness of the proposed solution against real channel noise in digital chaotic communications.

1. INTRODUCTION

Since the discovery by Pecora and Carroll that two identical chaotic generators can be synchronized [1], the synchronization of chaotic circuits for securing communication has been a topic of great interest, because utilizing chaotic generators offers some advantages in securing communication systems, such as broadband noise-like waveform, unpredictable properties and extreme sensitivity to initial conditions variations. Until now, several methods based on the identical synchronization, such as Additive Chaotic Masking (ACM) [2], Chaotic Switching [3, 4], Chaotic Parameter Modulation [5], improved chaotic

Received 9 August 2012, Accepted 3 September 2012, Scheduled 13 September 2012

* Corresponding author: Said Sadoudi (sadoudi.said@gmail.com).

additive masking [6] and impulsive synchronization methods [7], have been developed. However, recent results have shown that most of these techniques fail to realize experimentally a performed chaotic communication system because they ignored the channel noise, which is ubiquitous in the transmission of the driven signal, and then the noise's effect should be taken into account when to evaluate the performances of a chaos communication schemes. Precisely, the main problem with the ACM-based schemes is that any difference between the master/transmitter and slave/receiver systems will break the symmetry between the two systems. Therefore, the behavior of the two systems will no longer be identical. Consequently, the information signal will not be recovered faithfully at the receiver. This difference can be caused by both, an additive information or an additive noise channel. For the first one, the additive information could act like an external perturbation in the coupling signal. But, if the information signal is too small (in amplitude) [2], it is more possible to recover faithfully the information. However, for the second one, i.e., if additive noise is considered in the transmission channel, there will be difficulties to recover faithfully the information signal, because the additive noise channel contaminates the driving signal and then effects considerably the dynamics of the slave/receiver system. As a consequence, chaotic communication in the presence of channel noise is becoming an important issue nowadays. In order to overcome this drawback, some interesting method has been developed and some theoretical results have already been obtained. In this way, in [8] a method for securing transmission of encrypted message using chaos and noises is proposed. The authors in [9] considered the robust demodulation problem when there exist disturbances and noises in the channel. Numerically investigation of the secure communication based on the heterogenous chaotic systems with channel noise and nonidentity of parameters is proposed in [10]. In [11], the authors propose a new communication system which is able to separate noise successfully by using Independent Component Analysis (ICA), and a parameter modulation method based on a Lorenz chaotic system is employed for recovery of the source signals. The authors in [12] investigate numerically an alternative model to decrease the master-slave synchronization error when there is additive white Gaussian noise between master and slave: using coupled lattices instead of coupled single maps. An investigation of the characteristics of time-delay systems in the presence of Gaussian noise is given in [13]. In [14], the authors use a set of qualitatively different models of coupled oscillators (genetic, membrane, Cametabolism, and chemical oscillators) to study dynamical regimes in the presence of small detuning. Finally, the authors in [15] present a new scheme for the secured transmission of information based on master-slave synchronization of chaotic systems, using unknown-input observers. Other related results, see [16, 17]. However, the major of these proposed methods are not tested through a real channel under real transmission conditions for the best of our knowledge. It should be noted that a real channel is subjected to significant noise, has limited bandwidth, and undergoes attenuation effect.

In this paper, we present an original solution to the problem of chaotic synchronization high sensibility to channel noise. This solution is tested and validated, through a real channel under real transmission conditions, in a realized wireless hyperchaotic communication system using RF communication modules based on ZigBee communication protocol. The basic idea of the proposed solution is to avoid disturbing the chaotic dynamics of the slave/receiver by the injection of the transmitted driving signal. But rather, to trigger the generation of the slave/receiver chaotic signals at each reception of transmitted data and then to synchronize the received driving signal with the generated slave/receiver signal for the demodulation operation. To realize experimentally this pertinent idea, we develop an FPGA-based hardware architecture for interconnecting and adapting an embedded hyperchaotic generators (master/transmitter and slave/receiver) to the XBee RF modules based on ZigBee communication protocol. We recall that, this protocol, identified by the IEEE 802.15.4 standard [18], is designed to communicate data through hostile RF environments and to provide an easy-to-use wireless data solution characterized by secure, low-power and reliable wireless network architectures. The used hyperchaotic generator is the well-known hyperchaotic Lorenz system [19]. This last is implemented on FPGA technology by using an extension of the technique developed in [20, 21] for 3-dimensions chaotic systems. This technique is optimal since it uses directly VHDL (VHSIC Hardware Description Language) description of a numerical resolution method of continuous chaotic system models. Many transmission tests are carried out for different distances between the transmitter and receiver. The obtained real-time results validate the developed hardware architecture. Furthermore, it confirms the efficiency and the robustness of the proposed solution against channel noise in digital chaotic communication systems.

The remainder of this paper is organized as follows: Section 2 details the proposed solution principle. Section 3 presents our experimental design developed to realize and validate the proposed solution through real wireless communication tests and the different obtained real-time results and discussions. Finally, Section 4 draws appropriate conclusions.

2. PROPOSED SOLUTION

Since chaotic behavior depends highly and sensitively on tiny perturbations to the initial conditions and/or parameter values, our idea is to avoid the injection of the transmitted masking signal in the chaotic dynamics of the slave/receiver, as it is done in the most proposed chaos-based communication schemes [2, 6, 7, 22]. This means that, we propose a new technique to synchronize the master/transmitter and the slave/receiver, in digital chaotic communication scheme, without disturbing the dynamics of the used chaotic generators. In this scheme, whatever the disturbances affecting the received masking signal, the chaotic dynamics of the slave/receiver will not be disturbed. Therefore, the slave/receiver will generate chaotic behavior identical to that of master/transmitter. By triggering the slave/receiver chaotic signal generation at the reception of the transmitted masking signal and after synchronization of the two signals, the information signal will be recovered correctly after the demodulation operation. The principle of the proposed idea is illustrated by the scheme of Figure 1. At the transmitter, the transmitted signal after the Additive Chaos Masking (ACM) is,

$$s(t) = x(t) + m(t) \tag{1}$$

where, $m(t)$ is the information signal and $x(t)$ the chaotic carrier.

The transmitted signal $s(t)$ will be affected by additive noise $n(t)$ through the transmission channel. Thus, the received signal can be modeled as,

$$\hat{s}(t) = s(t) + n(t) \tag{2}$$

At the receiver, contrary to the proposed techniques in [2, 6, 7, 22], the receiver is composed by two modules, *signal detector* module and *chaotic generator* module which is identical to that of the transmitter, i.e., both generate chaotic signals with the same parameters and initial

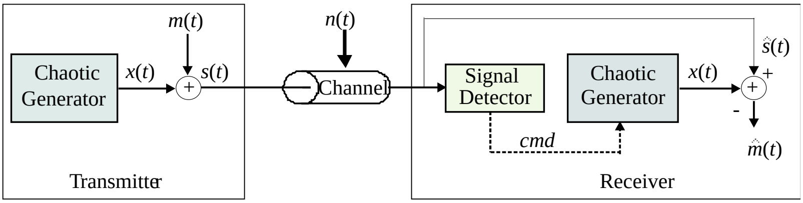


Figure 1. Proposed solution principle.

conditions values. At the detection of the transmitted signal $\hat{s}(t)$ and through the command signal *cmd*, the *signal detector* module orders the *chaotic generator* module to start the generation of the chaotic signal $x(t)$, which is identical to that of the master/transmitter. After synchronization of the regenerated chaotic signal $x(t)$ with the received signal $\hat{s}(t)$ (more details of the synchronization step are given in the next section), and the demodulation operation, we recover the noisy information signal as follow,

$$\hat{m}(t) = \hat{s}(t) - x(t) \tag{3}$$

thus, from (2) and (3), we obtain,

$$\hat{m}(t) = m(t) + n(t) \tag{4}$$

Eq. (4) shows that the noise affecting the recovered noisy information signal $\hat{m}(t)$ depends only on the additive noise $n(t)$ of the transmission channel affecting the transmitted masking signal $s(t)$ and do not depends on a perturbed chaotic behavior generation of the slave/receiver like the case in [2,6,7,22]. Indeed, the slave/receiver generates the chaotic signal $x(t)$, needed to the demodulation operation, without any perturbations of its chaotic dynamics. Consequently, the regenerated chaotic signal $x(t)$ is identical to that of the master/transmitter. However, by using a robust signal detector at the receiver, we can recover the transmitted masking signal $s(t)$ without additive noise, this means that $\hat{s}(t) = s(t)$. In this situation, we can recover then the information signal $m(t)$ correctly as follow,

$$m(t) = s(t) - x(t) \tag{5}$$

At this level, the asked question is: How can we realize experimentally the proposed solution? The response is investigated in the next section.

2.1. Experimental Implementation

To realize experimentally the proposed solution (Figure 1), we have used and associated the following three elements

(i) **Embedded chaotic generators**: Contrary to the analog generation, the numerical generation of chaos permits to overcome the problem of parameter mismatches and then offers the possibility to conceive separately two identical chaotic generators using the same parameters and initial conditions values. In this way, the technique proposed in [20,21] for conceiving embedded chaotic generators, based on FPGA implementation technology, responds appropriately to our needs. In fact, this technique uses an optimal hardware architecture based on direct VHDL description of the forth order Runge-Kutta (RK-4) method for implementing continuous chaotic generators. This architecture offers the advantage to adapt and interface easily the implemented chaotic generator to external devices as RF communications modules.

(ii) **RF communication modules**: To test and validate our solution through a real channel under real transmissions conditions, we have chosen to use the XBee RF modules based on ZigBee communication protocol [18]. This protocol is designed to communicate data through hostile RF environments and to provide an easy-to-use wireless data solution characterized by secure, low-power and reliable wireless network architectures [18]. In addition, XBee modules offer the advantage to interface to a host device through a well-known logic-level asynchronous serial port. In fact, devices having an UART interface can connect directly to the pins of the RF modules [18].

(iii) **UART interface**: To connect directly the embedded chaotic generators to the XBee RF modules, we have developed an FPGA-based hardware architecture realizing an UART interface compatible with the logic-level asynchronous serial port of the XBee modules [13] (more details are given later).

2.2. Experimental Design

The proposed solution is tested through real-time tests performed in the realized wireless hyperchaotic communication system presented by the scheme of Figure 2. The transmitter and receiver are implemented separately in two XUP Virtex-II Pro platforms of Xilinx [23]. The

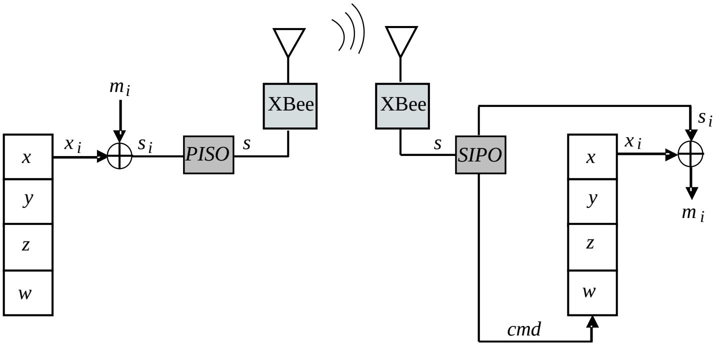


Figure 2. Realized wireless hyperchaotic communication system.

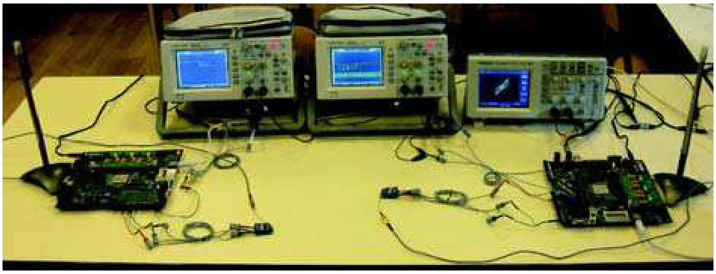


Figure 3. Experimental device.

used RF modules are the XBee Pro modules of Digi [18]. Instead of 3-Dimensional (3D) chaotic systems, we implement the hyperchaotic Lorenz generator [19] by using an extension of the technique developed in [20, 21] for implementing 3D continuous chaotic systems on FPGA technology. The hyperchaotic systems (4D continuous chaotic systems) are known for their very complex chaotic behaviors then 3D chaotic systems. Contrary to the ASM-based techniques [2, 6, 7, 22] and in order to increase the security level, we secure the information message by using the XOR operator (Modulation/Demodulation operation). A photo of the experimental design is given in Figure 3.

At the transmitter, the information samples m_i, generated on 32 bit parallel data format, are XORed with the hyperchaotic samples x_i, generated also on 32 bit parallel data format by the embedded hyperchaotic generator. Thus, the transmitted signal is as follow:

$$s_i = x_i \oplus m_i \tag{6}$$

The encrypted signal samples s_i are then converted to serial data format s by the *PISO* (Parallel Outputs/Serial Inputs) module and transmitted to the XBee Pro RF module at 115 kbps (the maximum Serial Interface Data Rate of the XBee Pro modules [18]). This latter transmits the encrypted serial data s to the receiver. At the receiver, the XBee Pro RF module transmits the received data, at the same bit rates of 115 kbps, to the *SIPO* (Serial Inputs/Parallel Outputs) module according to the asynchronous serial communication protocol. Note that the *SIPO* module represents the *Signal detector* bloc of the proposed scheme depicted in Figure 1. Indeed, at each start bit detection of the received serial data frame s, the *SIPO* module start the serial to parallel conversion and in the same time it orders, through the command signal *cmd*, the slave/receiver to generate the corresponding hyperchaotic sample x_i, which is evidently identical to that used at the transmitter for encrypting data information. Finally, when the two operations, serial to parallel conversion and the

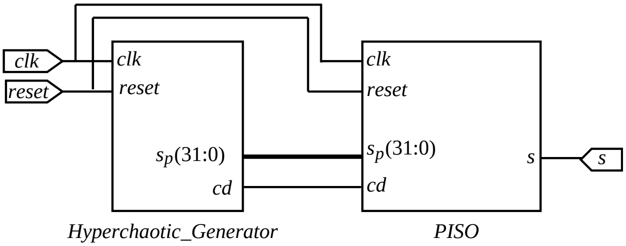


Figure 4. Bloc diagram of the transmitter architecture.

generation of x_i on parallel data format, are finished, it is easy to synchronize them (details are given in subsection 3.3) and then recover the information as follow:

$$m_i = s_i \oplus x_i \tag{7}$$

2.3. Transmitter Architecture

In Figure 4, we present the hardware architecture of the transmitter implemented in the first XUP Virtex-II Pro platform of Xilinx. This architecture is composed by two main modules operating and reinitializing under the signals *clk* and *reset* respectively. The frequency of the clock signal *clk* is imposed by the serial interface data rate of the XBee RF module [18]. Thus, we choose to work with the maximum data rate provided, which is 115 kbps and then the *clk* signal frequency must be 115 kHz.

The *Hyperchaotic_Generator* module generates the hyperchaotic samples on 32 bit parallel data format and after the modulation operation (Eq. (6)), it send the encrypted signal samples s_p to the *PISO* module. Under the command signal values *cd*, this last converts the samples s_p to serial data frames s and send them to the XBee module (see Figure 2). Note that, for the well comprehension of our solution we gives the details of the two modules, *hyperchaotic_Generator* and *PISO*, in the next section.

2.4. Receiver Architecture

In the second XUP Virtex-II Pro platform, the implemented hardware architecture, for realizing the receiver proposed in Figure 2, is presented by the scheme of Figure 5. It is composed by three main modules, the *SIPO* module, the *hyperchaotic_generator* and a *PISO* module. These modules operate under the clock signal *clk*, generated at the frequency of 115 kHz, and reinitialized by the signal *reset*. Note that the *PISO*

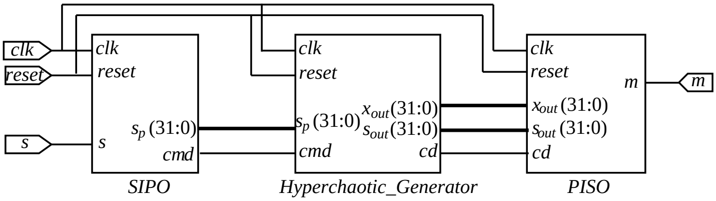


Figure 5. Bloc diagram of the receiver architecture.

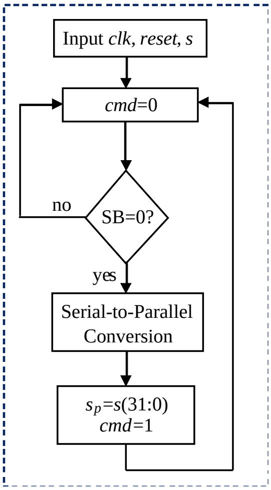


Figure 6. *SIPO* module flowchart.

module is added to the receiver architecture only in order to view the recovered information in serial data format on oscilloscope screen for comparison and validation.

(i) ***SIPO* module**: The operation of this module is described by the flowchart presented in Figure 6. Initially, the command signal *cmd* is set to 0, just at the detection of the Start Bit (SB), of the received serial data s, the serial to parallel conversion operation begins. At the end of this operation the parallel 32 bit samples s_p are assigned to the output of the module, at the same time *cmd* is set to 1.

(ii) ***Hyperchaotic_generator* module**: The developed architecture of this module is presented by the flowchart of Figure 7. When, the receiver is turned on, the module generates the first hyperchaotic samples x, y, z and w of the hyperchaotic Lorenz system, saves them in the variables X, Y, Z and W respectively, and according

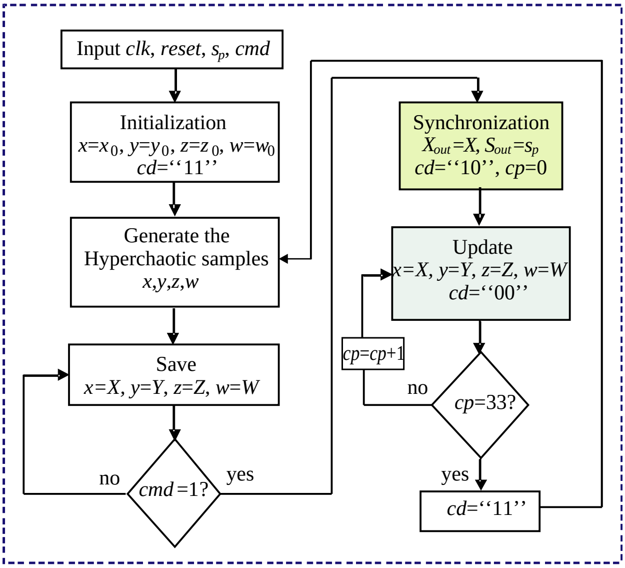


Figure 7. Flowchart of the hyperchaotic generator architecture at the receiver.

to the command signal value, it passes to the next step or it stay waiting. When *cmd* is set to "1", i.e., the paralleled 32 bit data samples s_p are available at the *SIPO* output, the generated hyperchaotic sample X is synchronized with the received samples s_p by affecting them simultaneously to the outputs X_{out} and s_{out} respectively. It should be noted that, the received data samples s_p undergo no change in the *hyperchaotic_generator* module. After synchronization, a step of updating is performed by assigning the computed hyperchaotic samples X, Y, Z and W to the variables x, y, z et w respectively for computing the next solutions. This last step is the basic idea of the proposed solution. Indeed, unlike the techniques [2, 6, 7, 17] based on the synchronization of Pecora and Carroll [1], where in the update step, the received signal is injected into the chaotic dynamics of the slave/receiver as shown in Figure 8. In the proposed solution (Figure 7), the hyperchaotic samples generation of the slave/receiver is not perturbed because at the update step we assign, at each time, the computed hyperchaotic samples to the variable x for computing the next samples ($x = X$). This permits to generate the same and identical hyperchaotic samples of the master/transmitter. Once the update is performed, the process will wait 33 clock cycles to

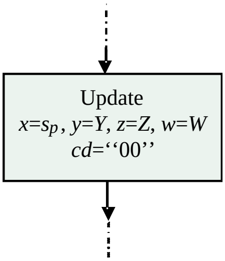


Figure 8. Synchronization principle in ASM-based techniques.

Table 1. *cd* command values.

cd values	The commands
"11"	Wait, does nothing
"10"	X_{out} and S_{out} are available: Demodulation $m = S_{out} \oplus X_{out}$
"00"	Parallel-to-Serial Conversion

repass to the hyperchaotic samples generation step.

(iii) ***PISO* module**: The operation of this module is controlled by the previous module through the command signal *cd*. This module has the role of recovering the information data frames m by demodulation and convert them in serial data format to real-time viewing on oscilloscope. Table 1 summarizes the *cd* command values.

2.5. Electronic Design Considerations

(i) **Composition**: The experimental design (Figure 3) is composed by three digital oscilloscopes, the first one is used to verify the preservation of the hyperchaotic behaviors, the second one is used to view the transmitted serial data frames and the third one permits to view and validate the recovered information in serial data format (Figure 9), two XUP Virtex-II Pro platforms of Xilinx, two 2.4 GHz antennas and two XBee Pro RF modules of Digi.

(ii) **Tests environment**: Experimental tests are performed in the same floor of the laboratory. The transmitter and receiver are placed separately in two non adjacent rooms to a distance of about 30 m. Knowing that the indoor/urban maximum range of the Xbee Pro modules is about of 100 m.

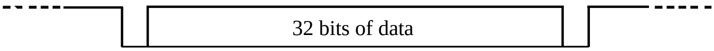


Figure 9. Visualized information data frames format.

(iii) **Experimental tests**: The real-time transmission test consists of transmitting a same serial data frame, generated locally on 32 bit, to the transmitter. This approach can easily check transmission errors in real-time, by visualizing the recovered data frames on oscilloscope and comparing them with those transmitted. To facilitate the reading of retrieved data and viewed on the oscilloscope screen, we add to the data frames, originally coded on 32 bit, one bit "0" on each side as it is shown in Figure 9.

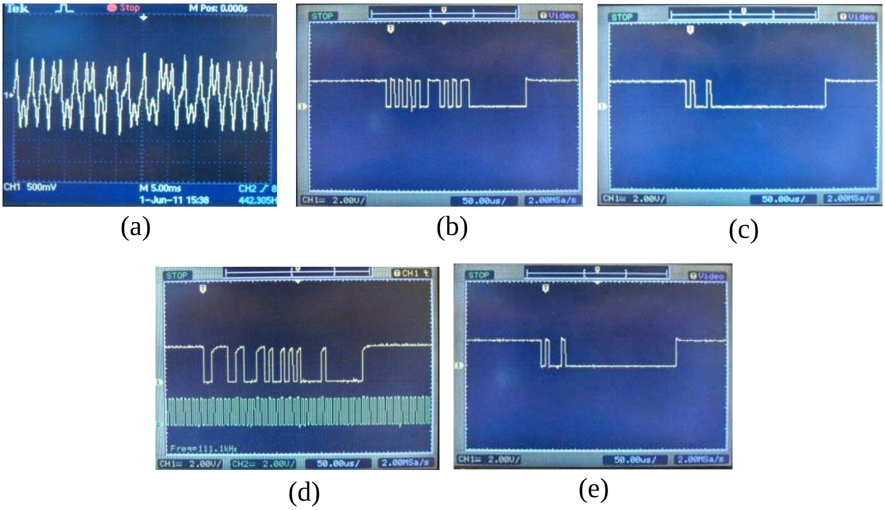
(iv) **XBee Pro module connections**: In our experiment, we use the XBee modules in their Transparent Mode with the minimum connections VCC, GND, Tx and Rx [18]. When operating in this mode, the modules act as a serial line replacement, i.e., all UART data, consisting of a start bit (low), eight data bit (least significant bit first) and a stop bit (high), received through the Rx pin is queued up for RF transmission. When RF data is received, the data is sent out the Tx pin.

(v) **XBee Pro modules configuration**: The main parameters configured in the used XBee Pro modules are the Serial Interface Data Rate (115 kbps), the Packetization Timeout ($RO = 0$) parameter, we use the channel 12 and the AES encryption is disabled.

(vi) **Performances**: It is known that, in a communication system the quality of the transmission is usually quantified by either the Bit Error Rate (BER) or the Packet Error Rate (PER), where a packet contains a number of bits. However, it should be noted that, in the proposed and realized wireless hyperchaotic communication system, we don't need to compute the BER of the transmission tests because it is equal to that characterizing the XBee Pro modules expressed by PER. The PER is the ratio of the incorrectly transferred data packets divided by the number of transferred packets (for details see [24]). For the XBee Pro modules the PER is 1% at −100 dBm [18].

2.6. Experimental Results and Discussions

In Figure 10, we show the real-time results of transmission test example consisting of transmitting the serial data frame "00000011" encoded



(a) (b) (c)

(d) (e)

Figure 10. Real time results, (a) the hyperchaotic carrier signal x, (b) the hyperchaotic sample x_i, (c) the serial information data frame m_i, (d) the transmitted serial masked data frame s, and (e) the recovered information data (value "00000011").

on 32 bits and represented on hexadecimal format. The hyperchaotic carrier signal x and a 32 bit hyperchaotic sample x_i (on serial data format) are shown in Figures 10(a) and 10(b) respectively. Figure 10(c) presents the serial information data frame m_i (see the scheme of Figure 2). An example of the transmitted serial masked data frame s with the corresponding clock signal *clk* is presented in Figure 10(d). Finally, the recovered information data (only for the constant value ("00000011") is shown in Figure 10(e). These results show that the information is recovered correctly without any bits error. After several experimental tests at various distances and dispositions of the transmitter and receiver and various data values, the obtained results have demonstrated the reliability of the proposed wireless hyperchaotic communication system.

However, in order to demonstrate the robustness of the proposed solution to the transmission channel noise and the relevance of our idea, we have added voluntary a noise to the masking signal for realizing (2). It is known that the impact of additive noise in a digital communication system results in the appearance of erroneous bits at the reception. We put ourselves in this situation by adding a bit to the transmitted signal s (Eq. (6)), and to facilitate the understanding of our approach, we have added the constant value "00001000" as the

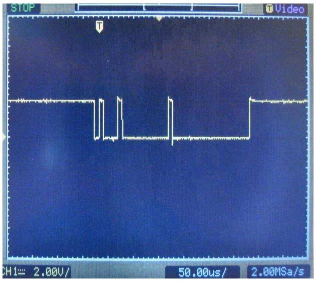


Figure 11. Real time results: recovered information with noise.

value of the noise. At the receiver outputs, we have obtained the results presented in Figure 11 which shows that the information data frame is recovered correctly but with the added noise ("00001000"), which is easily localized, confirming thus the Eq. (4). This means that the hyperchaotic dynamics of the slave/receiver is not disturbed by the added noise and the last really generates hyperchaotic samples identical to those of the master/transmitter. We resume this by saying that the slave/receiver chaotic dynamics generation is insensible to the channel noise.

3. CONCLUSION

An original solution to the hard problem of chaotic synchronization high sensibility to channel noise has been proposed. This solution has been tested and validated experimentally in a real channel noise environment through a realized wireless hyperchaotic communication system based on ZigBee protocol. The pertinent idea of the solution is focused on synchronizing two embedded hyperchaotic systems, implemented on FPGA circuits, without disruption of the slave/receiver dynamics. This means that, whatever the perturbations suffered by the received signal through the channel, the slave/receiver will generate chaotic solutions identical to that of the master/transmitter which will permit to recover the information correctly. The obtained results show the effectiveness and the robustness of the proposed solution against real channel noise in embedded digital chaotic communications.

REFERENCES

1. Pecora, L. M. and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Lett. A*, Vol. 64, 821–824, 1990.

2. Cuomo, K. M., "Circuit implementation of synchronized chaos with applications to communications," *Phys. Rev. Lett.*, Vol. 71, No. 1, 65–68, 1993.
3. Parlitz, U., "Transmission of digital signals by chaotic synchronization," *Int. J. Bifurcation and Chaos*, Vol. 2, 973–997, 1992.
4. Dedieu, H., "Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing," *IEEE Trans. Circ. Syst.-II.*, Vol. 40, No. 1, 634–641, 1993.
5. Yang, T., "Secure communication via chaotic parameter modulation," *IEEE Trans. Circ. Syst.-I*, Vol. 43, 817–819, 1996.
6. Milanovic, V., "Improved masking algorithm for chaotic communications systems," *Elec. Lett.*, Vol. 32, 11–12, 1996.
7. Yang, T., "A survey of chaotic secure communication systems," *Int. J. Computational Cognition*, Vol. 2, 81–130, 2004.
8. Minai, A. A., "Communicating with noise: How chaos and noise combine to generate secure encryption keys," *Chaos*, Vol. 8, 621–627, 1998.
9. Wang, X., "A robust demodulation application communication using chaotic signals," *Int. J. Bifurcation and Chaos*, Vol. 13, 227–231, 2003.
10. Murali, K., "Heterogeneous chaotic systems based cryptography," *Phys. Lett. A*, Vol. 272, 184–192, 2000.
11. Zhang, Y.-Q. and X.-Y. Wang, "A parameter modulation chaotic secure communication scheme with channel noises," *Chin. Phys. Lett.*, Vol. 28, No. 2, 02050, 2011.
12. Eisencraft, M. and A. M. Batista, "Discrete-time chaotic systems synchronization performance under additive noise," *Signal Processing*, Vol. 91, 2127–2131, 2011.
13. Senthilkumar, D. V. and J. Kurths, "Characteristics and synchronization of time-delay systems driven by a common noise," *Eur. Phys. J. Special Topics*, Vol. 187, 87–93, 2010.
14. Koseska, A., E. Volkov, and J. Kurths, "Parameter mismatches and oscillation death in coupled oscillators," *Chaos*, Vol. 20, 023132, 2010.
15. Dimassi, H., A. Loria, and S. Belghith, "A new secured transmission scheme based on chaotic synchronization via smooth adaptive unknown-input observers," *Comm. in Nonl. Sci. and Num. Simul.*, Vol. 17, No. 9, 3727–3739, 2012.
16. Chen, M., "A new private communication scheme based on the idea of fault detection and identification," *Phys. Lett. A*, Vol. 531, 177–183, 2006.

17. Li, S., "Breaking a chaos-noise-based secure communication scheme," *Chaos*, Vol. 15, 013703, 2005.
18. Digi International Inc., *Xbee/XBee-PRO ZB RF Modules,* 2010.
19. Barbosa, R., "Dydnamics of a hyperchaotic Lorenz systems," *Int. J. Bifurcation and Chaos*, Vol. 17, 4285–4294, 2007.
20. Sadoudi, S., "Embedded Genesio-Tesi chaotic generator for cipher communications," *Proc. 7th Int. Symp. Comm. Syst., Networks Dig. Signal Proc.*, 234–238, 2010.
21. Sadoudi, S., "An FPGA real-time implementation of the Chen's chaotic system for chaotic communications," *Int. J. Nonlinear Science*, Vol. 7, 467–474, 2009.
22. Kocarev, L., "Experimental demonstration of secure communications via chaotic synchronization," *Int. J. Bifurcation and Chaos*, Vol. 2, 709–713, 1992.
23. Xilinx, "Xilinx University program Virtex-II Pro development system," *Xilinx, UG069*, Vol. 1.1, 2008.
24. Centeno, A. and N. Alford, "Measurment of ZigBee wireless communications in mode-stirred and mode-tuned reverberation chamber," *Progress In Electromagnetics Research M*, Vol. 18, 171–178, 2011.