Microelectronic Engineering 90 (2012) 163–167

Contents lists available at ScienceDirect

Microelectronic Engineering

journal homepage: www.elsevier.com/locate/mee

A current mode CMOS noise generator using multiple Bernoulli maps

J. López-Hernández [a], A. Díaz-Méndez [b], J.L. Del-Río-Correa [c], M. Cruz-Irisson [a], R. Vázquez-Medina [a,*]

[a] *Instituto Politécnico Nacional, ESIME-Culhuacan, Santa Ana 1000, 04430 D.F., Mexico*
[b] *Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro 1, Tonantzintla, Puebla, Mexico*
[c] *Universidad Autónoma Metropolitana Iztapalapa, San Rafael Atlixco 186, 09340 D.F., Mexico*

ARTICLE INFO

Article history:
Available online 10 May 2011

Keywords:
Noise generators
Chaotic noise
CMOS circuits

ABSTRACT

This paper presents the analysis and design of a chaotic noise generator using statistical mechanic tools. The noise generator is a CMOS analog circuit operating in current mode, which generates chaotic signals using four Bernoulli chaotic maps with topological dependency on each other. They are randomly or deterministically selected to be applied by iterating an initial condition. These variants of the Bernoulli map are different in their slope and attenuation process. The initial condition can be considered as the secret key in the generation of chaotic noise. The advantage of this chaotic noise generator is that its' control parameter can be selected within an interval greater than the one obtained when basic Bernoulli map is used.

1. Introduction

Noise generators have multiple applications in different areas. In daily life the noise generators are used to emulate rain, waterfall or surf. They are used for better sleeping, privacy enhancement, blocking distractions, masking tinnitus, pacifying children and pets, soothing migraines among others. Also, they are useful in spread spectrum and watermarking systems and they are also an essential component in stream and block cryptosystems to obtain the confusion and diffusion effects in the plaintext.

The implementation of noise generating systems depends on the mathematical model used. Some reported works which are related with the design of discrete noise generator can be revised in [1,2], whereas the works related with the analog noise generators can be revised in [3].

There are few studies reporting the design of chaotic noise generators based on the Bernoulli map, most of the works use the logistic map because it is a function that is obtained from nonlinear behavior of the transistors. The implementations of noise generators with piecewise linear maps are reported using the tent map and Bernoulli map [4–7]; Table 1 shows a comparison of different alternatives to generate chaotic noise.

In all works reported in Table 1 it can be seen that the bifurcation diagram has a chaotic behavior limited to less than a third of the total area of plane α vs $f(x)$. On the other hand, this work proposes a noise generator that uses four chaotic maps, which are topologically conjugated. This feature of the proposed noise generator circuit allows that three different maps can be obtained from a basic Bernoulli map. Additionally, considering that the proposed noise generator uses four chaotic maps, the chaotic behavior region in the plane α vs $f(x)$ is much larger than when only a chaotic map is used. The size of the chaotic region and selection interval of the control parameter in the bifurcation diagram using four chaotic maps are about four times higher than those obtained with a single chaotic map.

Besides, looking forward that a noise generator circuit reaches the desired behavior it must be designed with precision. For this reason, when an electronic circuit is designed, it is necessary to establish a model that describes its dynamic behavior, this allows to find the valid operating regions. But, many of these circuits generate in fact, pseudorandom numbers instead of truly random numbers. Within that limitation, the most produced sequences should pass one or more of the randomness statistical tests (See NIST Special Publication 800-22 and RFC 4086) [8,9]. The main reasons for using such pseudorandom number generators are its low cost and easy implementation.

In the same way that the discrete noise signals the analog noise signals should have randomness and unpredictability conditions. The randomness condition is related with the statistical distribution of the output signal in the noise generator, whereas the unpredictability condition is related with the security and production of the seed.

2. Model and calculation scheme

The chaotic noise generator is a CMOS analog circuit operating in current mode, which generates chaotic signals using four Bernoulli chaotic maps, with topological dependency to each other and they are randomly or deterministically selected to be applied

[*] Corresponding author. Tel./fax: +52 55 5656 2058.
E-mail address: ruvazquez@ipn.mx (R. Vázquez-Medina).

0167-9317/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.mee.2011.05.009

Table 1
The summary of different works related with the noise generation using chaotic maps.

Authors	Year	Noise Generator Type	Chaotic Map	Relevance
I Campos-Canton et al.	2009	Analog	Tent Map	Realization of a circuit to implement the tent map
R Vazquez-Medina et al.	2009	Analog	Logistic Map	Design chaotic analog noise generators using MOS transistors
K Nakadaatal	2007	Analog	Bernoulli Map	Analog current-mode subthreshold CMOS circuit implementing a piecewise linear neuromorphic oscillator
VD Juncuetal	2006	Discrete	Map based in transistors nonlinearities	Two chaotic maps are used in the construction of noise generator
P Dudeketal	2003	Discrete	Map based in transistors nonlinearities	Two chaotic maps are used in the construction of noise generator
M Eisencraft et al.	2010	Digital	Skew Tent Map	Calculation of the Power Spectral Density of chaotic orbits generated by individual skew tent maps
Tommaso Addabbo et al.	2008	Digital	Bernoulli Map	Evolution of any initial distribution towards a statistical distribution invariant
Lwaa Faisal Adbudl	2010	Digital	Twisted Map	Noise generator using twisted map

by iterating an initial condition. The Bernoulli map has been extensively studied [10,11] and its model has great mathematical simplicity, which allows its implementation in both software and hardware. The variants of the Bernoulli map used in this work are different in their slope and attenuation process, and the initial condition can be considered as the secret key in the chaotic noise generation.

The implementation required for chaotic noise generating systems depends on the mathematical model used. There are models that use differential equations which require an analog circuit that operates in continuous time [12,13].Mathematically, the proposed architecture can be represented by:

$$x_{n+1} = \begin{cases} f_1(x_n); & 0 \leqslant \alpha \leqslant 0.5,\ \phi \leqslant 0.5 \\ f_2(x_n); & 0 \leqslant \alpha \leqslant 0.5,\ \phi > 0.5 \\ f_3(x_n); & 0.5 < \alpha \leqslant 1,\ \phi \leqslant 0.5 \\ f_4(x_n); & 0.5 < \alpha \leqslant 1,\ \phi > 0.5 \end{cases} \tag{1}$$

where,

$$f_1(x_n) = 1 - (1-\alpha)[2x_n \bmod 1] \tag{2}$$

$$f_2(x_n) = (1-\alpha)[(1-2x_n) \bmod 1] \tag{3}$$

$$f_3(x_n) = 1 - \alpha(2x_n \bmod 1) \tag{4}$$

$$f_4(x_n) = \alpha[(1-2x_n) \bmod 1] \tag{5}$$

In an equivalent form, the Eq. (1) can be written in the following way,

$$x_{n+1} = \begin{cases} 1 - \kappa(2x_n - S) & \phi \leqslant 0.5 \\ \kappa(1 - [2x_n - S]) & \phi > 0.5 \end{cases} \tag{6}$$

where

$$\kappa = \begin{cases} 1-\alpha & 0 \leqslant \alpha \leqslant 0.5 \\ \alpha & 0.5 < \alpha \leqslant 1 \end{cases} \tag{7}$$

$$S = \begin{cases} 0 & 0 \leqslant x_t \leqslant 0.5 \\ 1 & 0.5 < x_t \leqslant 1 \end{cases} \tag{8}$$

Based on this mathematical model, the proposed architecture for multiple Bernoulli maps is shown in Fig. 1. In this noise generator circuit, the basic Bernoulli map (2*x*–*S*) is a common block used to implement four necessary chaotic maps. The other processing blocks are subtractions that must be switched to obtain the four different chaotic Bernoulli maps. The switching elements are activated or deactivated according to signals m_1, m_2, k_1 and k_2. Signals k_1 and k_2 are complementary digital signals, that is, when k_1 is activated k_2 is deactivated and vice versa. These signals are responsible of the κ value according to Eq. (7); when $0 \leqslant \alpha \leqslant 0.5$ k_1 is activated and $\kappa = (1-\alpha)$ and, when $0.5 < \alpha \leqslant 1$ k_2 is activated and $\kappa = \alpha$. Signals m_1 and m_2 are also complementary digital signals responsible for controlling the sequence that must be used, which is produced by the respective map depending on the probability of use of the maps based on a selection probability ϕ.

When m_1 is active the Bernoulli map is multiplied by κ, and then the result is subtracted to 1 according to Eq. (6) if $\phi \leqslant 0.5$. When m_2 is active the Bernoulli map is subtracted to 1, and then the result is multiplied by κ according to Eq. (6) if $\phi > 0.5$. Signals k_1 and k_2 are generated by a comparator circuit, which compares the value of α with a reference value, if $0 \leqslant \alpha \leqslant 0.5$ the comparator circuit generates a digital signal for the activation of k_1 and, if $0.5 < \alpha \leqslant 1$ the comparator circuit generates a digital signal for the activation of k_2. Signals m_1 and m_2 must be generated randomly with a certain probability occurrence. The last block needed to generate analog signals in discrete time is the delay circuit, which must take a sample of the signal and hold it while it is processed.

From Eqs. (6)–(8) it can be observed that the required processing operations are subtraction and multiplication, available in current-mode techniques for circuit design. The common processing block in the system is the original Bernoulli map (2*x*–*S*), which amplifies the input signal by a factor of 2 and then the step function is subtracted to the amplified signal.

On the other hand, when discrete time models are used, analog or digital circuits can be implemented [14,15]. In this work, an analog implementation in discrete time and current-mode is shown. One of the advantages to use current-mode design techniques is the simplicity with which the addition or subtraction operations are implemented. In this alternative, Kirchhoff's current law in a single node is required in order to perform these operations, and it makes extensive use of the current mirror circuits. The basic function of the current mirror is to reflect or create multiple copies of a signal. However, it is possible to create circuits composed of several current mirrors able to perform various functions such as addition, subtraction and multiplication among others. In the design of the chaotic noise generator proposed it is needed to replicate some signals several times.

Fig. 2 shows the required circuit to generate the Bernoulli map, where all the transistors of the current mirrors have been designed to operate in the saturation region for strong inversion. In this circuit the input signal is amplified using current mirrors with a gain of 2, formed by transistors M1 to M6. Transistors M3 and M4

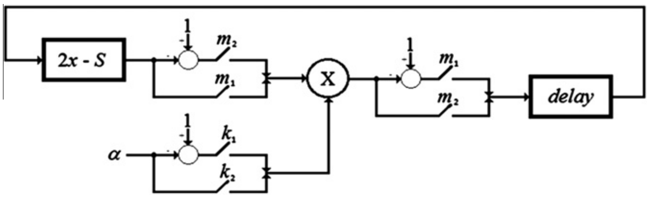

Fig. 1. Proposed architecture for multiple Bernoulli maps.

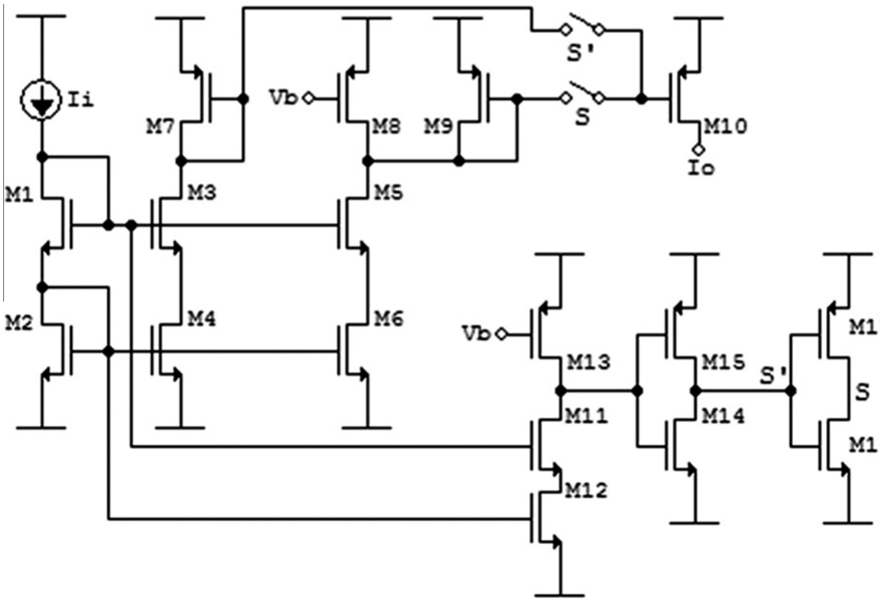

Fig. 2. CMOS implementation of Bernoulli map.

generate an amplified copy of the input signal and loaded by the half mirror M7. Transistors M5 and M6 generate an amplified copy of the input signal too, but this time the signal is subtracted from the bias current *Ib*, which is produced by M8 and *Vb*, and loaded by the half mirror M9.

In the Bernoulli map the output signal Io is 2*x*, where *x* is Ii, for $0 \leqslant x \leqslant 0.5$ and (2*x*–*S*) for $0.5 < x \leqslant 1$. Half mirrors M7 and M9 provide the signals 2*x* and (2*x*–*S*) respectively, having the two expected results separated. In order to have only one result, two switches controlled by the signals *S* and *S*' are added. These are complementary digital signals. When *S*' is active only the copy of the signal 2*x* can be copied by others mirrors and, when *S* is active only the copy of the signal (2*x*–*S*) can be copied. Again, only current mirrors and the subtraction are required for the implementation of the Bernoulli circuit. A 100 kHz sampling frequency is used at the delay block in order to guarantee the correct functionality of the chaotic noise generator. The circuit was polarized by *V*dd = 3.3 V and *Ib* = 100μA.

3. Results

Bernoulli map was selected because the stability islands in its bifurcation diagram and its Lyapunov exponent do not exist.

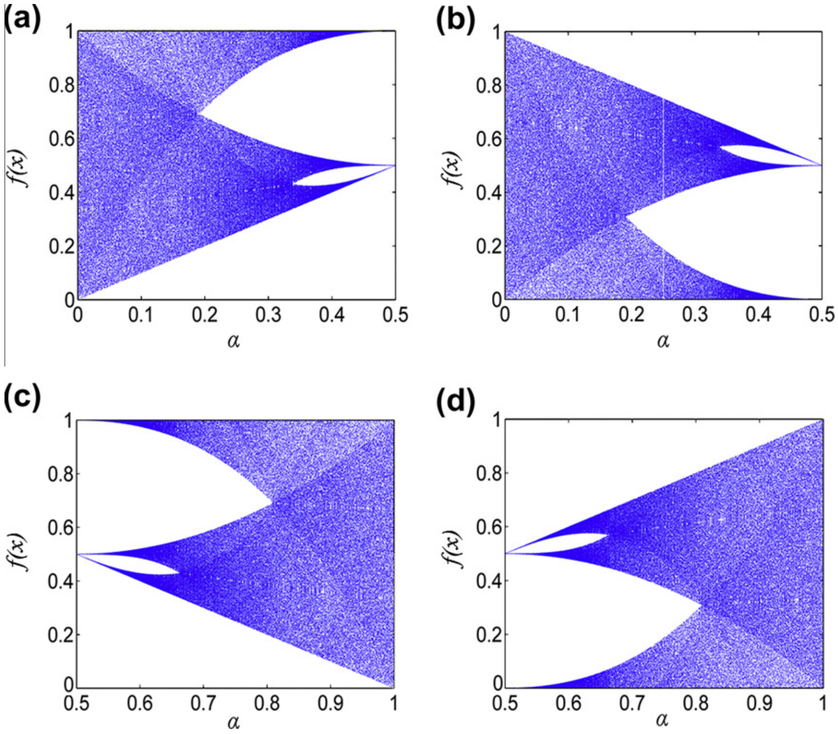

Fig. 3. Bifurcation diagram for of each variant, which were obtained in independent form: (a) $f_1(x_n)$, (b) $f_2(x_n)$, (c) $f_3(x_n)$ and (d) $f_4(x_n)$.

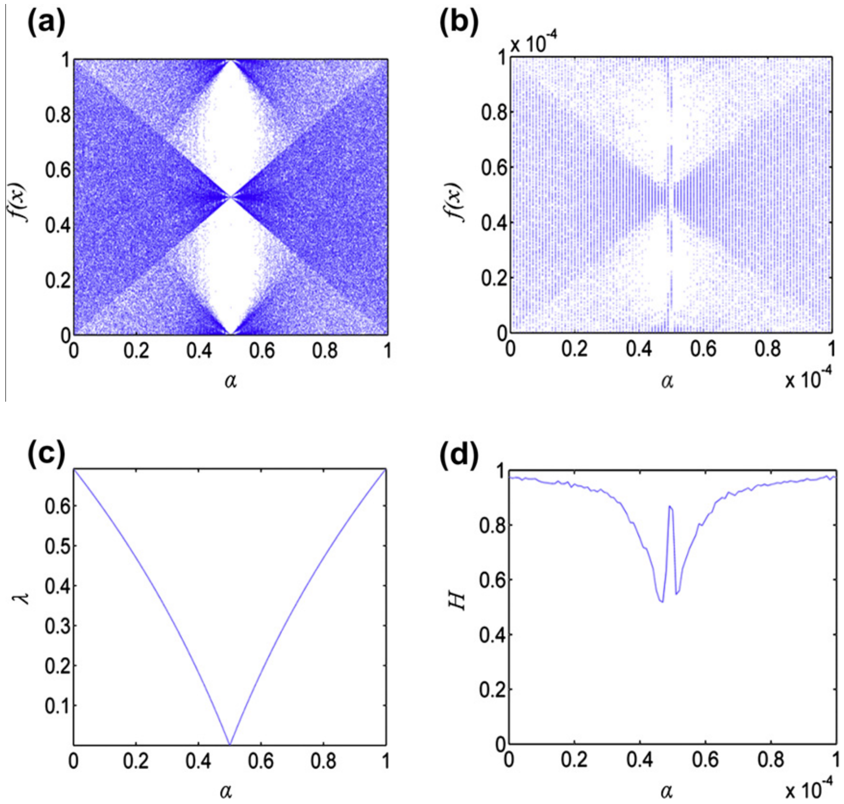

Fig. 4. (a) Theoretical bifurcation diagram using MatLab^{TM} simulations. (b) Experimental bifurcation diagram using HSPICE^{TM} simulations. (c) Lyapunov Exponent by proposed chaotic noise generator. (d) Entropy of signals produced by proposed chaotic noise generator.

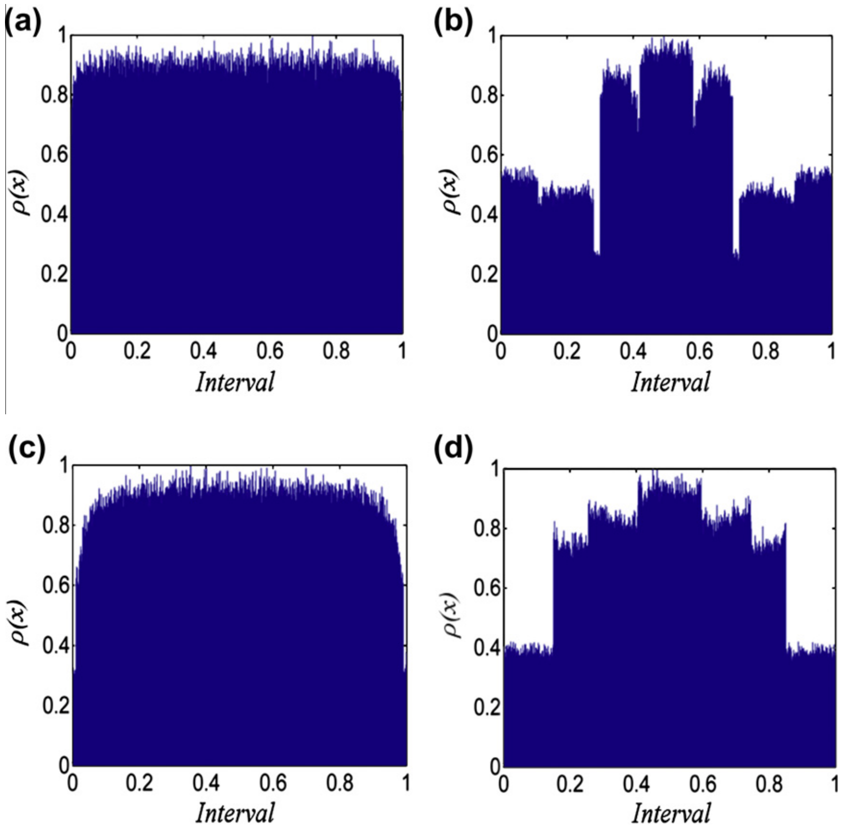

Fig. 5. Statistical distribution of output signal produced by proposed chaotic noise generator using = 0.001, = 0.7, α = 0.01 and α = 0.85.

Fig. 3 shows the bifurcations diagrams of each variant, which were obtained in independent form. It is worth important observe that each diagram of bifurcation does not have islands of stability within the chaotic regions.

Fig. 4a and b show the theoretical and experimental bifurcation diagram of the signals produced by proposed chaotic noise generator: (a) MatLab™ simulations and (b) HSPICE™ simulations. Notice that the obtained bifurcation diagram of the proposed chaotic noise generator has a greater chaotic region, compared independently with each one of the Bernoulli maps used. In addition, the obtained bifurcation diagram does not have stability islands. Fig. 4c shows Lyapunov exponent of the signals produced by proposed chaotic noise generator. This result confirms that the proposed chaotic noise generator has not stability islands. Fig. 4d shows the experimental normalized entropy of signals produced by proposed chaotic noise generator. The results of normalized entropy are congruent with Lyapunov exponent results.

The statistical distribution calculated using Birkhoff's theorem [16] was compared with the simulated statistical distribution of the circuit output signal. Fig. 5 shows the statistical distribution for different values of α.

4. Conclusions

In summary, the analysis and design of a chaotic noise generator using statistical mechanic tools are presented. Bernoulli map was selected because the stability islands in its bifurcation diagram and its Lyapunov exponent do not exist. The absence of stability islands avoids a periodic behavior in the operation of the circuit. In this work, the statistical distribution of output signal of circuit designed is shown and it is similar to the uniform distribution. Simplicity of the models and the required processing blocks in the variants of the Bernoulli map causes that CMOS circuit is easy to implement in current mode. The results show that it is possible to define a specific behavior of the circuit, selecting the control parameter of the Bernoulli map within the chaotic region. In this way, the control parameter can be used as additional key to the initial condition in the generation of chaotic noise. The advantage of this chaotic noise generator is that its control parameter can be selected within an interval greater than the one available when basic Bernoulli map is used.

Acknowledgments

The authors are grateful to the financial support of the SIP-IPN 20110670 and ICYTDF 270/2010 projects.

References

[1] V.D. Juncu, M. Rafiei-Naeini, P. Dudek, Analog Integrated Circuits and Signal Processing 46 (2006) 275–280.
[2] P. Dudek, V.D. Juncu, Electronics Letters 39 (2003) 1431.
[3] A. Kandangath, S. Krishnamoorthy, Y.C. Lai, J.A. Gaudet, IEEE Transactions on Circuits and Systems I 54 (2007) 1109–1119.
[4] I. Campos-Cantón, E. Campos-Cantón, J. S. Murguía, H.C. Rosu, Chaos, Solitons & Fractals, 42 (2009) 12-16.
[5] K. Nakada, T. Asai, T. Hirose, H. Hayashi, Y. Amemiya, Neurocomputing 71 (2007) 3–12.
[6] M. Eisencraft, D.M. Kato, L.H.A. Monteiro, Signal Processing 90 (2010) 385–390.
[7] R. Vázquez-Medina, A. Díaz-Méndez, J.L. del Río-Correa, J. López-Hernández, Chaos, Solitons & Fractals, 40 (4) (2009) 1779-1793.
[8] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST: SP 800–22 Rev. 1a, Maryland, USA, 2010.
[9] J. Schiller, S. Crocker, Network Working Group RFC- 4086 (2005).
[10] H.O. Peitgen, H. Jürgens, D. Saupe, Springer Science 509 (2004).
[11] J. Argyris, G. Faust, M. Haase, Elsevier Science Ltd 234 (1994).
[12] K. Ozdemir, S. Kilinc, S. Ozoguz, Communication and Applications Conference (2008) 1–4.
[13] Q. Li, X. Yang, F. Yang, Electronics Letters 39 (2003) 1306–1307.
[14] Y. Simin, Chinese Control Conference (2007) 409–413.
[15] S. Rocchi, V. Vignoli, Proc. of IEEE International Symposium on Circuits and Systems 5 (1999) 463–466.
[16] I.P. Cornfeld, S.V. Fomin, Ya.G. Sinai, Ergodic Theory, Springer-Verlag, New York, 1982.