entropy

MDPI

Article

A New Chaotic System with a Self-Excited Attractor: Entropy Measurement, Signal Encryption, and Parameter Estimation

Guanghui Xu [1,2], Yasser Shekofteh [3,*], Akif Akgül [4], Chunbiao Li [5,6] and Shirin Panahi [7]

[1] School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China; xuguanghui29@163.com

[2] Hubei Collaborative Innovation Center for High-efficiency Utilization of Solar Energy, Hubei University of Technology, Wuhan 430068, China

[3] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran 1983969411, Iran

[4] Department of Electrical and Electronic Engineering, Faculty of Technology, Sakarya University, Serdivan 54187, Turkey; aakgul@sakarya.edu.tr

[5] Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science & Technology, Nanjing 210044, China; goontry@126.com

[6] School of Electronic & Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China

[7] Department of Biomedical Engineering, Amirkabir University of Technology, Tehran 1591634311, Iran; sh.panahi89@gmail.com

* Correspondence: y_shekofteh@sbu.ac.ir; Tel./Fax: +98-21-2243-1804

Received: 28 December 2017; Accepted: 21 January 2018; Published: 27 January 2018

Abstract: In this paper, we introduce a new chaotic system that is used for an engineering application of the signal encryption. It has some interesting features, and its successful implementation and manufacturing were performed via a real circuit as a random number generator. In addition, we provide a parameter estimation method to extract chaotic model parameters from the real data of the chaotic circuit. The parameter estimation method is based on the attractor distribution modeling in the state space, which is compatible with the chaotic system characteristics. Here, a Gaussian mixture model (GMM) is used as a main part of cost function computations in the parameter estimation method. To optimize the cost function, we also apply two recent efficient optimization methods: WOA (Whale Optimization Algorithm), and MVO (Multi-Verse Optimizer) algorithms. The results show the success of the parameter estimation procedure.

Keywords: chaotic systems; circuit design; parameter estimation; optimization methods; Gaussian mixture model

1. Introduction

A chaotic system has been considered with great potential in engineering applications, in which many chaotic systems with different properties have been studied. Specifically, some systems have the properties of amplitude control and offset boosting [1–4]. In this paper, we use a new three-dimensional (3D) chaotic system in random number generation and signal encryption, which are important engineering applications of chaotic systems [5–10]. To do this, an electronic design of the system is implemented as a real electronic circuit to generate random numbers. Finally, the one-dimensional (1D) and two-dimensional (2D) parameter estimation of the system is reported based on a non-traditional parametric model cost function and two new optimization methods.

The topic of self-excited and hidden attractors is a new attractive topic in dynamical systems [11–13]. Recent studies have classified dynamical attractors as self-excited or hidden [14–17]. A self-excited attractor has a basin of attraction which intersects with at least one unstable equilibrium. If that is not the case, the attractor is hidden [18–21]. According to the above definition, most of the classical chaotic attractors are self-excited [22,23]. It has been demonstrated that the attractors in dynamical systems with no equilibria [24–32], with stable equilibria [33,34], with lines [35,36] and curves of equilibria [37–41], and with plane [42] or surface of equilibria [43] are hidden attractors. Even some systems can belong to more than one category [3,44–47]. Hidden attractors cannot simply be located. There have been some efforts in literature to solve this problem [18,48,49].

As we know, modeling of the real world chaotic systems has received great attention in recent decades [50–55]. Choosing proper values for model parameters is essential in chaotic systems, since they are very sensitive, both to model parameters and initial conditions. A slight change in the parameters of the chaotic system may cause important bifurcation in its behavior, because of the butterfly effect of the chaotic system [56]. Therefore, the parameter estimation problem of chaotic system models is a complex problem [57–59].

There are some widely used methods for the parameter estimation of the chaotic systems which are based on optimization methods [60–62]. In these methods, the problem of the parameter estimation is generally formulated as a cost function based on an error function between a time series obtained from a real system and a time series obtained from a known model with unknown parameters of that system. The goal of the parameter estimation method will then be to find the best values of the unknown parameters of the model which minimize the cost function. In addition, the optimization approaches have been used algorithms for this problem to find the best values of the unknown parameters as quickly as possible. They are algorithms such as genetic [63], particle swarm optimization [64], and evolutionary programming [65]. However, approaches that utilize cost function based on the error function seem to bear major limitations because of the butterfly effect of the chaotic systems [57–59].

It was remarked that the state space would be a proper domain to analyze the chaotic systems rather than the time-domain. The time series generated by the chaotic systems have random-like behavior in the time-domain, but they are ordered in the state space. They can show specific topologies in the state space named strange attractors. In this paper, we use a non-conventional metric as a useful cost function for the parameter estimation method. Accordingly, we model the attractor distribution of a real chaotic system by a parametric model named the Gaussian mixture model (GMM). It can provide flexible and probabilistic modeling for data distributions. GMM is also a commonly used parametric model in the pattern recognition and machine learning domain [66]. For example, in the speech recognition field, a set of GMMs was introduced to model phone attractors in a reconstructed phase space (RPS) in which the RPS is a time-independent domain similar to the state space [67–69]. The phone classification results showed that the GMM could be a useful model to capture the structure and topology of the speech attractors in the RPS. In addition, models of Gaussian mixture were recently used as the parameter identification method for some chaotic systems [70–72].

Here, to optimize the cost function, two recent efficient optimization methods are applied, including the WOA (Whale Optimization Algorithm), and MVO (Multi-Verse Optimizer) algorithm. Also, for testing the parameter estimation method in the chaotic systems, a real circuit is utilized based on a new chaotic system in this paper. All the data (time series) are obtained from the circuit that is designed based on the new chaotic system.

The contributions of this paper are described as:

- A new 3D chaotic system with saddle equilibriums is proposed by a set of ordinary differential equations.
- Dynamical properties of the 3D chaotic system are then reported that exhibit its dynamics.
- The electronic circuit implementation of the 3D chaotic system is studied and used to present a random number generator (RNG), and its signal encryption is then introduced as an engineering application.
- 1D and 2D parameter estimation of the electronic circuit is done by a GMM based cost function.

- The cost function is optimized using two new efficient optimization methods called the WOA and the MVO algorithms.
- By comparing the experimental data with numerically generated time series, the best-fitting parameters are found because the circuit had (almost) the same dynamics as the 3D chaotic system.

The structure of the paper is organized as follows: in the next section we introduce and analyze the new chaotic system with saddle equilibriums. In Section 2, we investigate it carefully through bifurcation analysis, spectrum of Lyapunov exponents, and its entropy. Section 3 deals with the circuit implementation of this new system and a real circuit application based on mobile RNG design. In the next section, the cost function based on the GMM is introduced. Two meta-heuristic optimization algorithms (WOA and MVO) are presented in Section 5. Results of the cost function and the parameter estimation of the new chaotic system using the WOA and MVO methods are available in Section 6. Finally, Section 7 is the conclusion of the paper.

2. A New Chaotic System and Its Analysis

In this section we introduce a new 3D system which can show chaotic behavior. Consider a system described with the following ordinary differential equations:

$$\begin{aligned} \dot{x} &= gz \\ \dot{y} &= dx^2 + ey^2 - f \\ \dot{z} &= -ax - bx^2 + cy^2 \end{aligned} \tag{1}$$

This system is in the chaotic state when $a = 4.0$, $b = 1.0$, $c = 1.0$, $d = 1.0$, $e = 1.0$, f = 4.0 and $g = 1.0$. Different projections of the phase portrait for this system are plotted in Figure 1, which shows its strange attractor in 2D state spaces. System (1) is a new offset-boostable one [1–4] in which the variable z can be boosted with a direct constant in the first dimension.

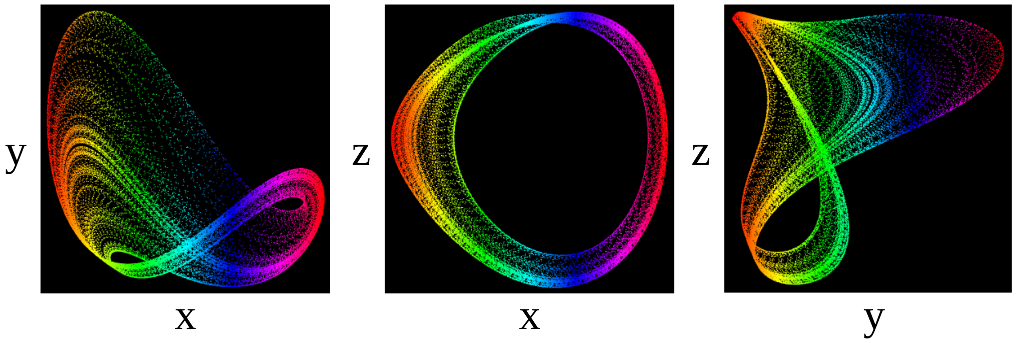


Figure 1. Different projections of the chaotic attractor of system (1) with the initial conditions $(-1.8,\ -1.5,\ -2.5)$.

This system has fixed-points in every (x^*, y^*, z^*) which satisfy the following equation,

$$\left\{ \begin{array}{l} \dot{x} = 0 \\ \dot{y} = 0 \\ \dot{z} = 0 \end{array} \right. \rightarrow \left\{ \begin{array}{l} 0 = z \\ 0 = x^2 + y^2 - 4 \\ 0 = -4x - x^2 + y^2 \end{array} \right. \tag{2}$$

According to Equation (2), the system (1) has two equilibria in $A = (0.7321, 3.4641, 0)$ and $B = (0.7321, -3.4641, 0)$. The Jacobian matrix of the system (1) is

$$J = \begin{bmatrix} 0 & 0 & 1 \\ 2x & 2y & 0 \\ -4 - 2x & 2y & 0 \end{bmatrix} \tag{3}$$

and the corresponding eigenvalues for A and B are

$$A:\begin{cases}\lambda_1 = 3.9784\\ \lambda_{2,3} = -0.12798 \pm i2.5428\end{cases} \quad B:\begin{cases}\lambda_1 = -3.9784\\ \lambda_{2,3} = 0.12798 \pm i2.5428\end{cases} \tag{4}$$

Therefore, both equilibria are saddle-foci. Thus, the attractor is self-excited.

3. Bifurcation and Entropy Analysis

3.1. Bifurcation Analysis

In this part, we investigate the behaviors of the system (1) with respect to changing parameter g. In part (A) of Figure 2 the bifurcation diagram of the system is shown and in part (B) of this figure Lyapunov exponents can be observed. It is important to be careful about numerical calculation of Lyapunov exponents, since improper use of usual methods may cause some issues [14,15,73–77]. We have used the algorithm of [78] for computation of Lyapunov exponents.

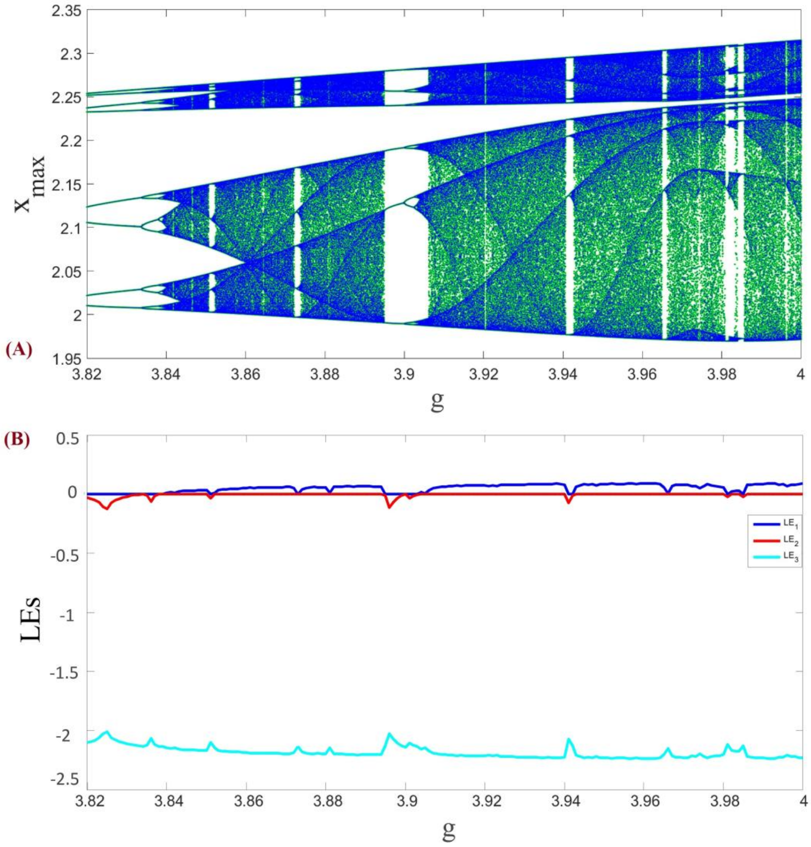


Figure 2. (**A**) Bifurcation diagram of the system (1) with respect to parameter g, and (**B**) Lyapunov exponents of the system (1) with respect to parameter g.

As can be seen in Figure 2A, changing parameter g causes a familiar period doubling route to chaos. In addition, positive values of the Lyapunov exponents in Figure 2B show that the underlying system is the chaotic system.

3.2. Entropy Analysis

There are many techniques to evaluate the system complexity from data. One of the most famous method which had been used since 1991 is Approximate Entropy (ApEn) [79]. ApEn can be applied to short and noisy data with outliers [80]. Therefore, many systems can be categorized by means of complexity [81]. Consider the N data sample $u(1),\ u(2),\ \ldots,\ u(N)$ with the vector sequence $x(1),\ x(2), \ldots, x(N-m+1) \in R^m$ which can be defined as:

$$x(i) = [u(i),\ u(i+1),\ \ldots,\ u(i+m-1)] \tag{5}$$

where m is an integer and determines the dimension of $x(i)$ as the length of compared run of data. Then, for each i in the $1 \le i \le N-m+1$, the following equation is defined:

$$c_i^m(r) = \frac{J}{N-m+1} \tag{6}$$

$$d[x(i), x(j)] \le r\ ,\ 1 \le j \le N-m+1 \tag{7}$$

$$d[x(i), x(j)] = \max_{k=1,2,\ldots,D}(|u(i+k-1) - u(j+k-1)|) \tag{8}$$

where J is the number of correct vectors in Equation (7), the number of vectors that the distance (infinity norm or maximum norm) between them and $x(i)$ is lower than r, and r is also a tolerance threshold that is defined by the product of a constant C to the standard deviation of data.

$$r = C \times std(u(t)) \quad 0.1 \le C \le 0.2 \tag{9}$$

Then, the *ApEn* can be written as:

$$\phi^m(r) = \frac{\sum_{i=1}^{N-m+1} logC_i^m(r)}{N-m+1} \tag{10}$$

$$ApEn(m, r) = \lim_{N \to \infty}\left[\phi^m(r) - \phi^{m+1}(r)\right] \tag{11}$$

The estimation of Equation (11) for N data sample is as follows,

$$ApEn(m, r, N) = \phi^m(r) - \phi^{m+1}(r) \tag{12}$$

It can be derived that the *ApEn* values determine the similarity between chosen window and the sliding window of the data. Therefore, m determines the length of the window to be compared, and r is the tolerance threshold for accepting similar pattern between two windows. Figure 3 represents the *ApEn* diagram of the system (1) with respect to parameter g.

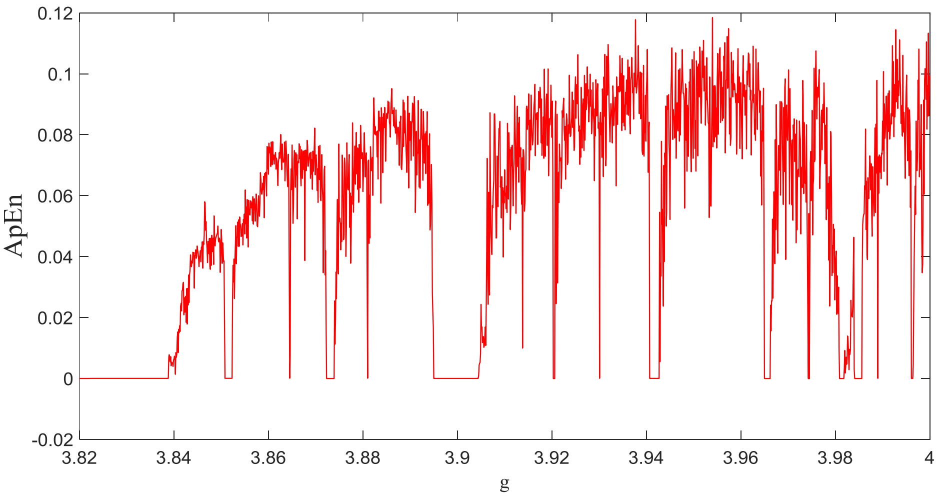


Figure 3. ApEn of the system (1) with respect to parameter g.

4. Real Circuit Design of the New Chaotic System as a Mobile RNG and Its Application for Signal Encryption

Random number generator (RNG) algorithms produce a sequence of numbers with properties of randomness and they are a research subject since a few decades. Chaotic systems are commonly used in the random numbers generation algorithms because they are complex and very sensitive. In this section, a mobile RNG design is implemented based on the introduced chaotic system (1) and then signal encryption application is realized with the RNG.

The micro-computer based mobile RNG can be used in many fields especially in encryption studies with low cost and high performance. It is aimed at encryption of multimedia data (audio, image, video, text etc.) with the realized mobile RNG to be flexible and user friendly.

As far as we know, random number generators require high cost hardware like computers and FPGA in order to successfully pass the universal tests [82–86]. In this paper, the design of a microcomputer-based mobile RNG and a signal encryption application with the designed RNG is realized without needing hardware such as FPGA, computers, etc. Therefore, “Raspberry Pi 3” is used here as hardware which supports 64-bit processing capability. Since the “Raspberry Pi 3” card has 64-bit processing capability, it can generate very sensitive decimal numbers; thus, randomness of these generated numbers is very high. BCM2837 SoC (system-on-chip) 64-bit ARMv8 quad core Cortex A53 processor running @1.2GHz produced by Broadcom is available on the card. The general view of “Raspberry Pi 3” is as given as in Figure 4.

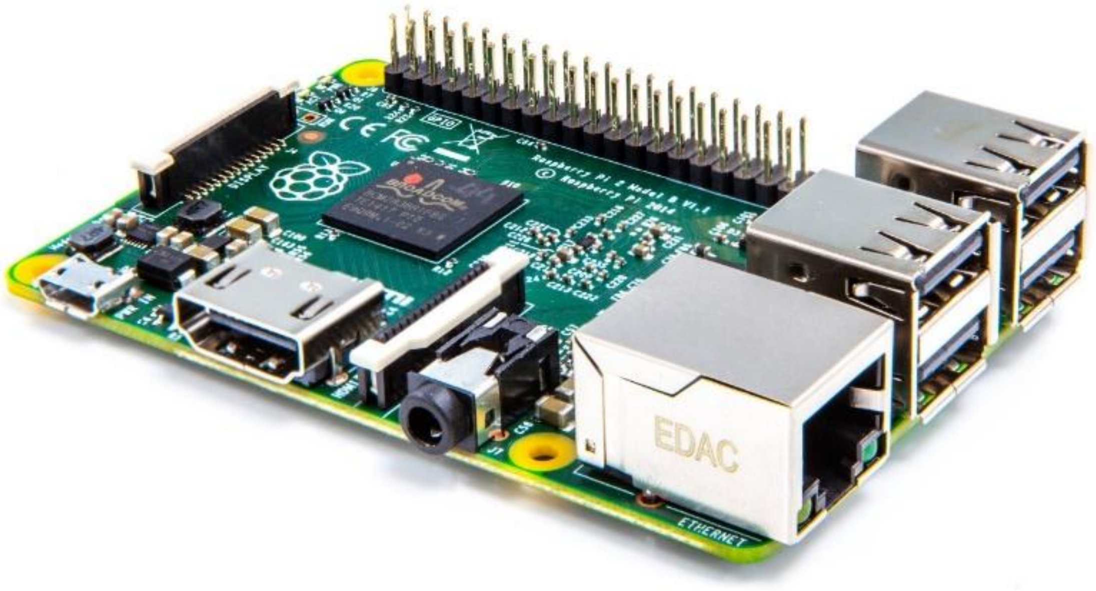


Figure 4. The general outlook of “Raspberry Pi 3”.

Our proposed circuit is used as an entropy source for RNG. Then, the NIST-800-22 tests are performed on random numbers to evaluate the performance of the designed RNG. In the next step, a signal encryption application is realized as an example application in “Raspberry Pi 3”. Also, an electronic circuit implementation of the chaotic circuit is done in OrCAD-PSpice and on the oscilloscope.

4.1. Micro-Computer-Based Mobile RNG Design

As before mentioned, the “Raspberry Pi 3” board is used as a micro-computer for RNG design and encryption application. The chaotic system of (1) is also utilized in the RNG design. The RNG design steps are given in Algorithm 1 as a pseudo code.

Algorithm 1. Mobile RNG design algorithm pseudo code.

```
 Start
 Entering parameters and initial condition of the chaotic system
 Determination of the value of $\Delta h$
 Sampling with determination $\Delta h$ value
 while (least 1 M. Bit data) do
   Solving the chaotic system with RK4
   Convert float to binary number (32 bit)
   Select the bits (LSB-16 bit) from 32 bit binary number
 end while
The implementation of NIST Tests for 1 M. Bit data
if test results == pass then
    Successful results (Ready tested 1 M. Bit data)
    RNG applications (Cryptology, data hiding, watermarking, etc.)
else (test results == false)
return the previous steps and generate bits again
end if
End
```

After entering parameters and initial condition of the system (1), the outputs are discretized with the RK4 differential equation solving method. Then, float numbers are obtained and converted into 32 bits binary numbers. Later, the RNG design is executed with obtained binary numbers. The last 16 bits of the outputs (x, y and z variables) are used in the design. The NIST-800-22 statistical tests are also used to prove the success of the RNG design [87]. The NIST-800-22 tests consist of 16 different tests such as monobit, serial and discrete Fourier transform tests. The p-values of the test should be greater than 0.001 in order to be counted as successful in NISTS-800-22 tests.

Our experiments show that the random numbers generated from x, y and z outputs successfully passed all the tests with the last 16 bits. The NIST-800-22 tests results are given in Table 1. The ready tested random numbers that pass all of the NIST-800-22 tests can be used in applications that require high security such as cryptology, data hiding, watermarking, etc.

Table 1. RNG NIST-800-22 tests for x, y and z outputs.

Statistical Tests	p-Value-x (X_16bit)	p-Value-y (Y_16bit)	p-Value-z (Z_16bit)	Result
Frequency (Monobit) Test	0.5741	0.2209	0.9904	Successful
Block-Frequency Test	0.5692	0.2711	0.4011	Successful
Cumulative-Sums Test	0.6255	0.1218	0.4619	Successful
Runs Test	0.7012	0.1846	0.5313	Successful
Longest-Run Test	0.6207	0.1881	0.6901	Successful
Binary Matrix Rank Test	0.4378	0.9036	0.9755	Successful
Discrete Fourier Transform Test	0.0796	0.5819	0.6931	Successful
Non-Overlapping Templates Test	0.1685	0.0011	0.0803	Successful
Overlapping Templates Test	0.8824	0.1699	0.5441	Successful
Maurer's Universal Statistical Test	0.5665	0.3602	0.8932	Successful
Approximate Entropy Test	0.1364	0.7072	0.6264	Successful
Random-Excursions Test ($x = -4$)	0.9005	0.3467	0.6683	Successful
Random-Excursions Variant Test ($x = 9$)	0.5249	0.9845	0.5880	Successful
Serial Test-1	0.1784	0.6299	0.5716	Successful
Serial Test-2	0.5467	0.4709	0.7633	Successful
Linear-Complexity Test	0.7039	0.3601	0.2000	Successful

To obtain the random numbers, the pins x, y, and z GPIO (General purpose input/output) are utilized as shown in Figure 5. They are the 37th pin for x output, the 35th pin for y output, and the 38th pin for z output from “Raspberry Pi 3”.

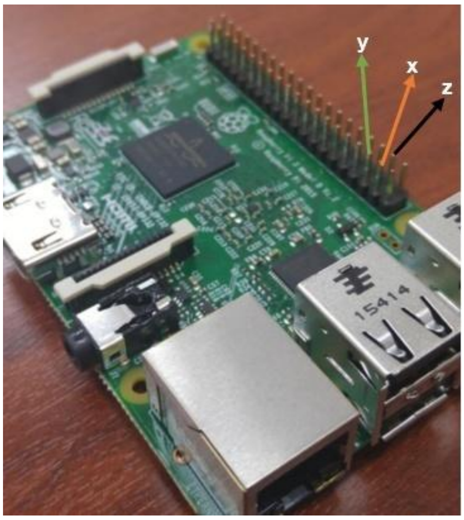
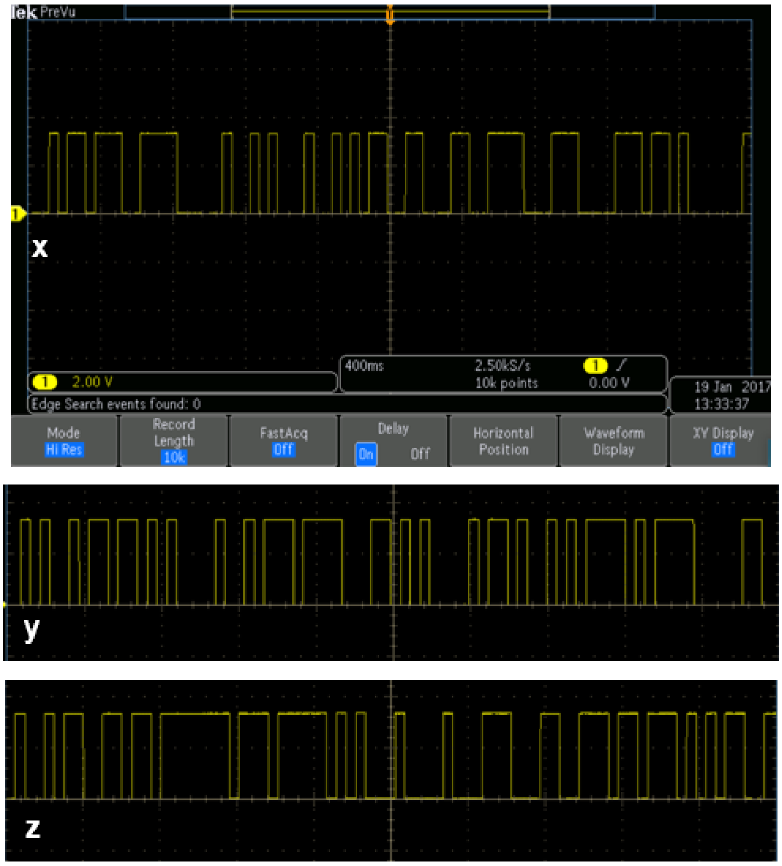


Figure 5. Pins of x, y and z for chaotic system outputs from “Raspberry Pi 3”.

The generated x, y and z outputs (first 50 bits) are shown in Figure 6 as real-time oscilloscope outputs. The 35th, 37th, and 38th GPIO pins give x, y and z outputs in Figure 5, respectively. They are used for real-time oscilloscope outputs.



1 2.00 V
400ms 2.50kS/s 10k points 1 / 0.00 V
Edge Search events found: 0
19 Jan 2017 13:33:37
Mode Hi Res
Record Length 10k
FastAcq Off
Delay On Off
Horizontal Position
Waveform Display
XY Display Off

Figure 6. x, y and z outputs on the oscilloscope (first 50 bits).

4.2. Signal Encryption Application Using "Raspberry Pi 3"

In this section, a signal encryption application with RNG that was generated from the proposed chaotic system is realized in "Raspberry Pi 3". The steps of the encryption and decryption process are given in Algorithm 2. In the encryption application, a signal that consists of 512 bits is used and shown (first 50 bits) in Figure 7 as the real-time oscilloscope outputs.

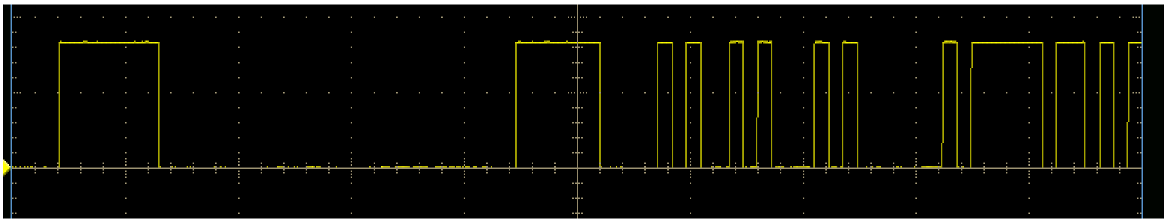


Figure 7. Original signal data (first 50 bits).

Algorithm 2. Chaos based encryption and decryption algorithm pseudo code.

```
Start
Getting ready to test random numbers for keys
Getting signal data to be encrypted
for $i$ = 1 for all original data
    random number bit xor original data bit
end
Encrypted data
for $i$ = 1 for all encrypted data
    random number bit xor encrypted data bit
 end
Decrypted data
End
```

For the encryption process, the 'XOR' operator is used. Figure 8 shows the first 50 bits of the encrypted signal as real-time oscilloscope outputs. Since the encryption process is performed for each bit, the size of the encrypted data is also 512.

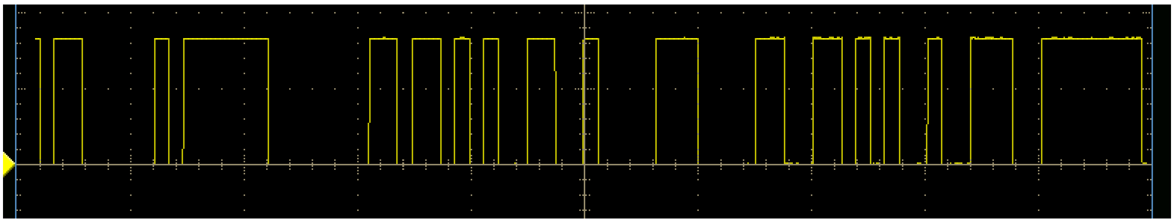


Figure 8. Encrypted signal data (first 50 bits).

The same keys generated from the chaotic system are needed for decryption. With these keys, the original data can be obtained, again. The first 50 bits of the decrypted signal are shown in Figure 9 as the real-time oscilloscope outputs. As can be seen, comparing Figures 7 and 9, for the first 50 bits, there is no deformation.

In the implemented method, a cryptoanalyser who wants to crack the encrypted data must know exactly all of the parameters and initial values of the chaotic system used in the encryption. Also, encrypted data will be not decrypted without "Raspberry Pi 3".

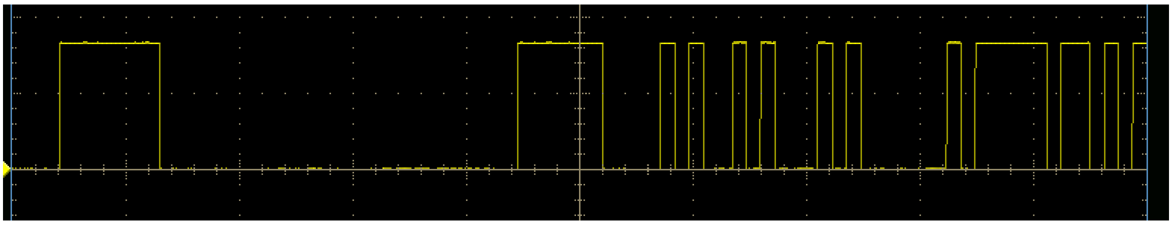


Figure 9. Decrypted Signal Data (first 50 bits).

4.3. Electronic Circuit Implementation of the Chaotic System in OrCAD-PSpice and on the Oscilloscope

In this part, we design an electronic circuit based on system (1) in OrCAD-PSpice (Figure 10) and on the board (Figure 11). The circuit includes simple electronic elements such as resistors, multipliers, capacitor, and opamps. Note that PSPICE simulation of chaotic circuits is quite trivial. In the literature, such systems are implemented with integrated circuit technology [88].

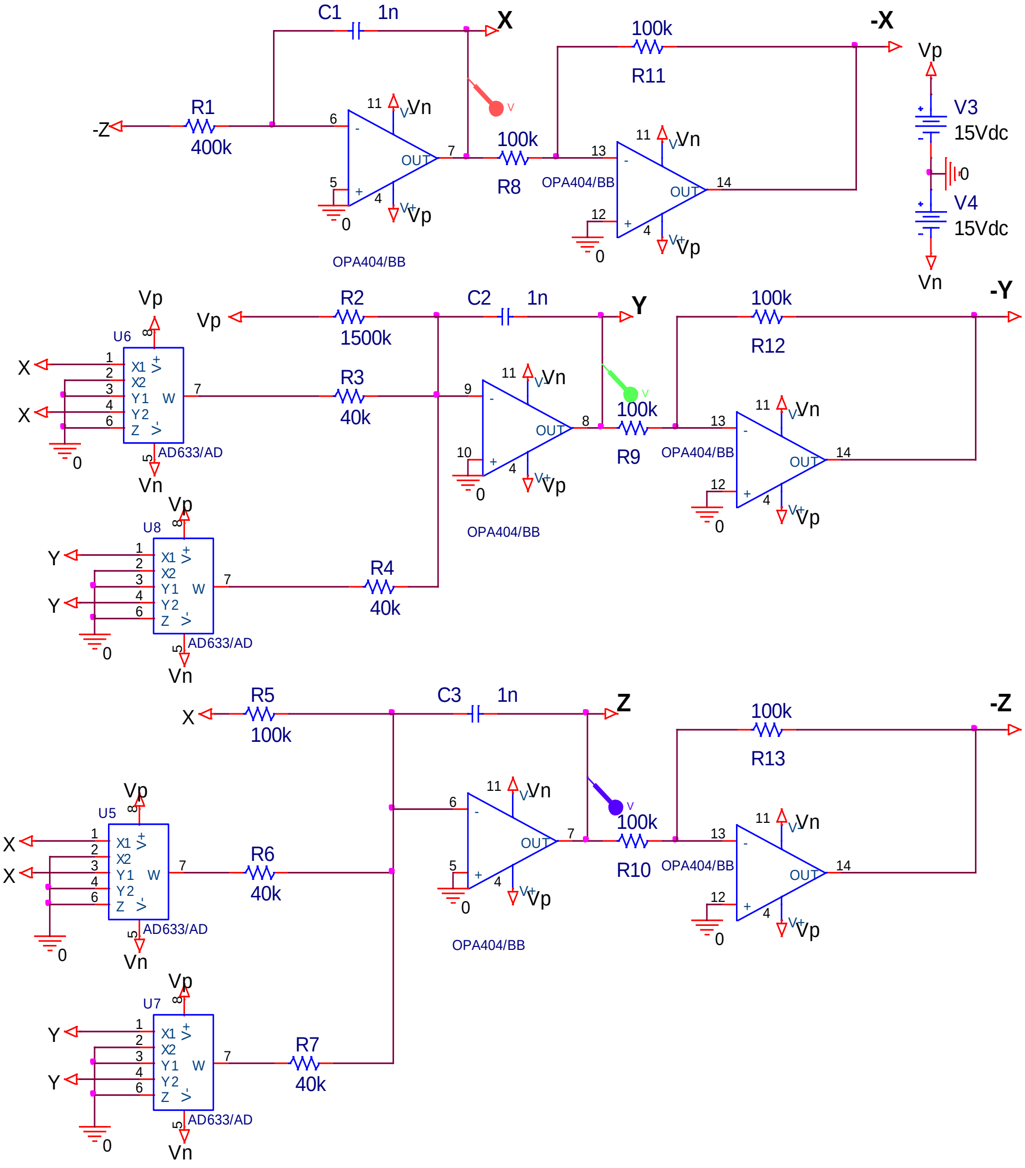


Figure 10. The electronic circuit schematic of system (1).

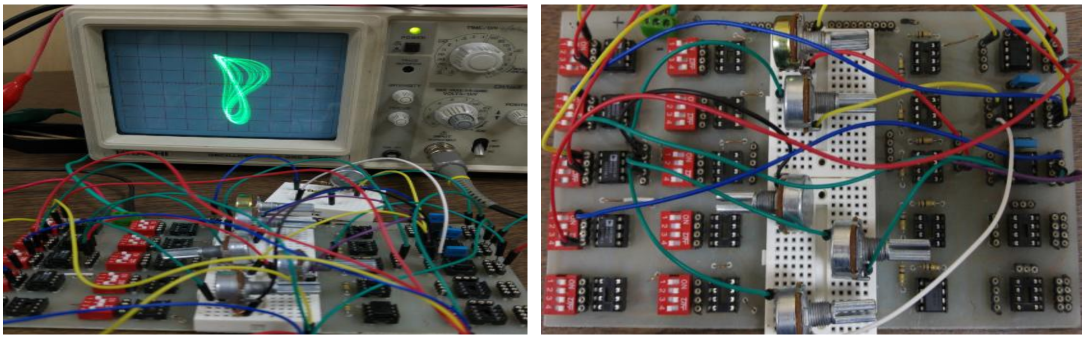


Figure 11. The experimental circuit of the chaotic circuit and the phase portraits of system (1) on the oscilloscope.

The OrCAD-PSpice simulation outputs, which are two-dimensional phase portraits of the chaotic system, are seen in Figures 12 and 13, respectively. As can be seen from the ORCAD-PSpice outputs in Figure 12 and oscilloscope outputs in Figure 13, the results are similar.

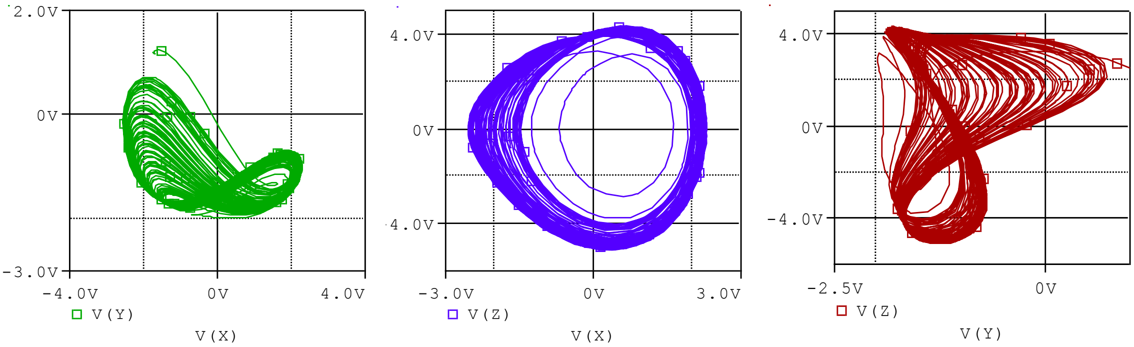


Figure 12. The phase portraits of the system (1) in ORCAD-Pspice.

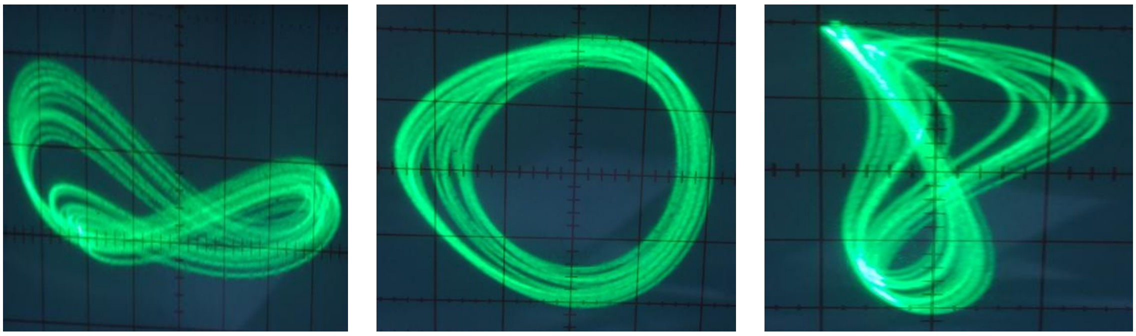


Figure 13. The phase portraits of system (1) on the oscilloscope.

5. Parameter Estimation of the Chaotic System

In this section, we introduce the parameter estimation method used for the chaotic circuit. This method utilizes a cost function which was adopted for the chaotic systems. The cost function of the parameter estimation method is based on a similarity metric using a parametric model of strange attractors in the state space. It was shown that this cost function could yield better results than the conventional error-based cost function over the time-domain [71]. The time-independent property of the state space is a sufficient reason to use this cost function because the state space can show complex behaviors of the strange attractor of chaotic systems [89].

As before mentioned, the utilized cost function is based on the attractor modeling; therefore, we need a model to represent the distribution of the attractor points in the state space. As a smooth parametric model, a Gaussian mixture model (GMM) can model the chaotic attractor geometry in the state space [67]. The GMM is a parametric probability density function represented by a weighted sum of Gaussian component densities [90]. It can model the distribution of the attractor points in the state space based on its powerful characteristics [91]. So far, metrics such as Kullback–Leibler divergence (also called relative entropy) were defined to measure distance between GMMs [90]. In addition, similarity-based metrics such as likelihood functions have been used to measure distance between a time series and a GMM. This idea was recently used as phone classification methods by parametric models of the distribution points of the speech signal in a high-dimension domain named RPS [67–69].

The GMMs have also been used for parameter estimation of some chaotic systems [70,71]. They were utilized similar to the task of the phone classification method. Suppose we have a chaotic system with a known model and its trajectory was recorded. We can then generate a GMM for the strange attractor of the chaotic system in the state space. Utilizing a distance-like metric over a likelihood function, we can compute dissimilarity between the learned GMM model of the real system attractor (with unknown parameters) and a distribution of a new attractor obtained by a system's model (with known parameters) in the state space to complete the parameter estimation method. Therefore, the score of the distance-based metric will be equal to the cost function of the parameter estimation method.

5.1. The GMM Computation as a Cost Function

A GMM with M mixtures is a weighted sum of M individual Gaussian densities. Each Gaussian density as a component of the GMM is represented by three main factors, mixture weight, mean vector, and covariance matrix. Therefore, they can be shown by a set of parameters, λ, as follows,

$$\lambda = \{w_m, \mu_m, \Sigma_m\},\ m = 1, \ldots, M p(v|\lambda) = \sum_{m=1}^{M} w_m \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_m|^{1/2}} \exp\left\{ \frac{-1}{2} (v - \mu_m)^T {\Sigma_m}^{-1} (v - \mu_m) \right\} \quad (13)$$

and $p(v|\lambda)$ is the conditional probability of a D-dimensional single observation vector v given the GMM of λ. The $p(v|\lambda)$ can show a likelihood score. It expresses how probable the observed vector v is for the GMM of λ. In Equation (13) $|.|$ is the determinant operator, exp(.) denotes the exponential function, and M is the number of mixtures (Gaussian components). In addition, for m-th mixture, $w_m \in [0,1]$ is an scalar and named m-th mixing coefficient or mixture weight, μ_m is the m-th D-dimensional mean vector, and Σ_m is the m-th $D \times D$ covariance matrix. The mean vector and covariance matrix of a Gaussian component can show the center and the shape of points distribution around those of the component. It should be noted that the mixing coefficients w_m are constrained to sum to 1, i.e., $\sum_{m=1}^{M} w_m = 1$.

As a problem which depends on the complexity of the data distribution, there is no analytical solution to determine the optimum number of GMM mixtures, M, needed for modeling of the attractors. Therefore, it is common to use a trial-and-error method to choose an adequate value of M. In our attractor modeling problem, to obtain a proper GMM model of the attractor in the state space, we evaluate some values of M. Generally, we need a higher value of M for attractor modeling if it has a very complex dynamic in the state space. One should note that while a higher number of mixtures can increase the performance of the cost function, it also increases the computational cost.

To find the similarity score between the attractor of a real system and the state space points of a specific model obtained from a chaotic system with known parameters (for example chaotic system (1)), the likelihood score can be calculated. Therefore, the parameter estimation of a known chaotic system with unknown parameters can be performed using the following two phases; a learning phase, here named "phase A", which includes fitting the GMM to the attractor of the real system, and an evaluation phase, named "phase B", to select the best values of parameters for the known chaotic model which causes the maximum similarity score or equally minimum distance score (cost function) over the learned GMM. Following are those phases in details:

5.2. Phase A

The first phase of the parameter estimation approach is the learning phase to find the GMM parameters, λ in Equation (13). The GMM learns the attractor's distribution of a real system, e.g., a chaotic circuit. Suppose $S = \{s_1, s_2, \ldots, s_N\}$ is an $N \times D$ matrix consisting of N-samples of the time series of the real data in the D-dimensional state space. Therefore, each sample is a D-dimensional observation vector. To find the GMM parameters, an iterative expectation-maximization (EM) algorithm is utilized as follows:

5.2.1. Initialization Step

Initialize the mean vector μ_m, covariance matrix Σ_m and mixing coefficients w_m in Equation (13) and evaluate the initial value of the logarithm of the likelihood score obtained from the input time series as follows,

$$logp(\mathbf{S}|\lambda) = \sum_{n=1}^{N} log(p(s_n|\lambda)) \tag{14}$$

5.2.2. Expectation Step

Evaluate values of $r(s_i, m)$, named responsibility of i-th sample of $\mathbf{S}$ given the m-th Gaussian component, using the current values of the GMM parameters:

$$r(s_i, m) = \frac{w_m \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_m|^{1/2}} exp\left\{ \frac{-1}{2} (s_i - \mu_m)^T \Sigma_m{}^{-1} (s_i - \mu_m) \right\}}{\sum_{j=1}^{M} w_j \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_j|^{1/2}} exp\left\{ \frac{-1}{2} \left(s_i - \mu_j\right)^T \Sigma_j{}^{-1} \left(s_i - \mu_j\right) \right\}} \tag{15}$$

5.2.3. Maximization Step

Re-estimate the parameters of the GMM utilizing the estimated values of the responsibilities as follows:

$$N_m = \sum_{i=1}^{N} r(s_i, m) \tag{16}$$

$$\mu_m = \frac{1}{N_m} \sum_{i=1}^{N} r(s_i, m) s_i \tag{17}$$

$$\Sigma_m = \frac{1}{N_m} \sum_{i=1}^{N} r(s_i, m)(s_i - \mu_m)(s_i - \mu_m)^T \tag{18}$$

$$w_m = \frac{N_m}{N} \tag{19}$$

5.2.4. Likelihood Computation Step

Evaluate the logarithm of the likelihood score in Equation (14) and check for convergence criterion. If the convergence criterion is not satisfied, return to Section 5.2.2.

5.3. Phase B

The second phase is finding the best parameters of the known model of the chaotic system (with unknown parameters) using the learned GMM in the phase A. Here, the search space will be formed from a set of acceptable values of the model parameters. Now we suppose that the values of the parameters $a\&b$ of the system (1) are unknown. Then, for each pair of parameters (a, b), the chaotic system of (1) will be simulated, and a trajectory $T(a,b) = (t_1, t_2, t_3, \ldots, t_K|a,b)$ with K samples will be obtained where each t_K is D-dimensional measured data in the state space. Finally, using an average point-by-point log-likelihood score obtained from the learned GMM, λ, a similarity-based score is computed as follows,

$$log\left(p(T^{(a,b)}|\lambda)\right) = \frac{1}{K}\sum_{K=1}^{K} log(p(t_k|\lambda)) \tag{20}$$

where $T^{(a,b)}$ is a matrix whose rows are composed from the state space vectors of the system trajectory with the model's parameters (a,b), and K is the number of the state space point. The parameter estimation method of the model is accomplished by computing Equation (20) and selecting the parameters of the model that can obtain the best similarity-based score, which here means the maximum score. If we use the negative of the similarity-based score, then the parameter estimation becomes a cost function minimization. Therefore, the best parameter selection, $(a,b)^*$, would be conducted by the following criteria, $J(.)$, based on the negative of mean log-likelihood score,

$$(a,b)^* = argmin\{J((a,b))\}\& \ J((a,b)) = -p\left(T^{(a,b)}\middle|\lambda\right) \tag{21}$$

Equation (21) shows the utilized cost function and (a,b) is the set of the system parameters (1). Here, λ is the learned GMM of the real system attractor obtained from the phase A. The objective of the parameter estimation method is to determine the parameters of the system, (a,b) when the cost function is minimized to result in the minimum value of $J((a,b))$. The minimum value of the cost function guarantees the best solution with the proper parameters.

5.4. The GMM of Chaotic Circuit

Based on the observation vector v of the chaotic circuit, in this work, $D = 3$ is selected for the dimension of the state space according to system (1). Using the prepared real training data for the attractor by the chaotic circuit as a real system, the GMM will be specialized in order to model the geometry of that attractor. Figure 14 shows the attractor of the chaotic system in a three-dimensional state space with its GMM modeling using 256 Gaussian components, $M = 256$, where every three-dimensional ellipsoid corresponds to one of the Gaussian components.

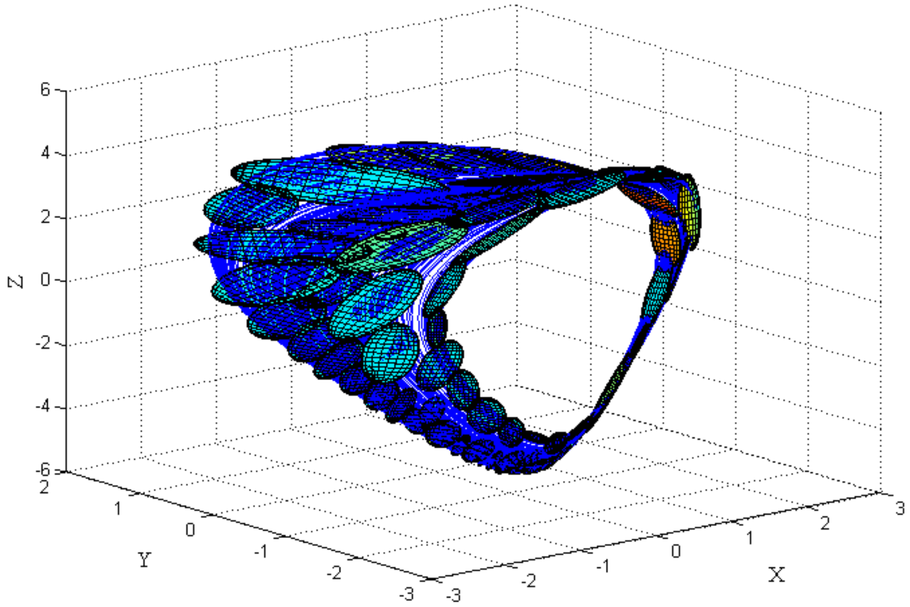


Figure 14. Plot of the attractor and its GMM modeling with $M = 256$ components for the chaotic system (1) with $a = 4$ & $b = 1$, in the 3-D state space.

6. Optimization Algorithm

There are four major categories to which different kinds of optimization methods belong: Enumerative methods, Calculus-Based methods, Heuristic methods, and Meta-heuristic methods [62]. Meta-heuristic optimization algorithms, which cover a wide range of problems, are becoming more and more popular in engineering applications [92,93]. Nature can be named as one of the most important sources of inspiration for new meta-heuristic algorithms. On this subject, black and white holes in cosmology and Humpback whales in the sea aid in constructing the MVO (Multi-Verse Optimizer) and WOA (Whale Optimization Algorithm) meta-heuristic algorithm [94,95]. We now introduce these methods:

6.1. The Whale Optimization Algorithm

The WOA algorithm is based on the hunting behavior of Humpback whales which can encircle the recognized location of prey. The WOA algorithm assumes that the current best candidate solution is the target prey or is close to the optimum. The next step is about the attacking strategy which is the bubble-net strategy. Putting it all together, the proposed WOA method includes three major steps in the simulation: the search for prey, encircling prey, and the bubble-net foraging behavior of humpback whales. For complete details see [94].

6.2. Multi-Verse Optimizer: A Nature-Inspired Algorithm for GlobalOptimization

Another novel nature-inspired algorithm is Multi-Verse Optimizer (MVO). Cosmology (white hole, black hole, and wormhole) is the main inspiration of this algorithm. As mentioned before, every search process in the optimization algorithm consists of two phase: exploration and exploitation. The MVO supports this by white and black holes in order to respond to the exploration phase and wormholes for the exploitation phase. Further details are described in [95].

6.3. Experimental Results

In this section, some simulations are done to investigate the acceptability of the parameter estimation method of the chaotic circuit. We have used a fourth-order Runge-Kutta method with a step size of 10 ms and a total of 30,000 samples corresponding to a time of 300 s. Here, we assume that the original chaotic system of (1) should be estimated by minimization of the GMM-based cost function.

First, using some 1D parameter estimation methods, different number of the GMM's components, $M = (64, 96, 128, 192, 256)$, are used to show the sufficiency of the cost function. The experimental results of the cost function versus the values of the parameters *a*&*b* are depicted in Figures 15 and 16, respectively.

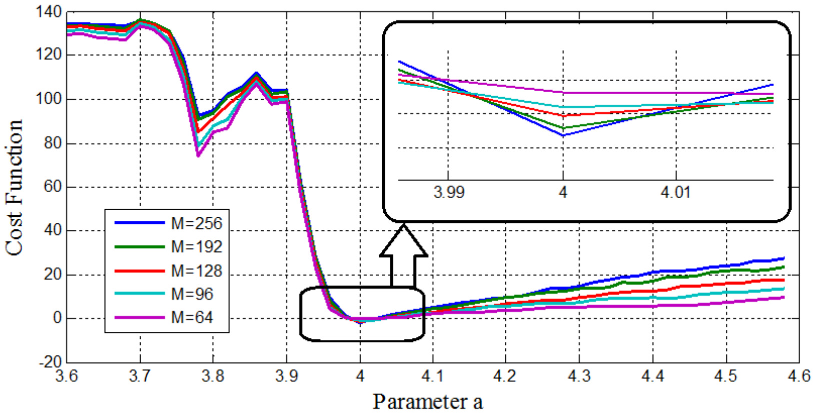


Figure 15. Cost function versus parameter *a*, with different number GMM components (M) for the 1D parameter estimation method.

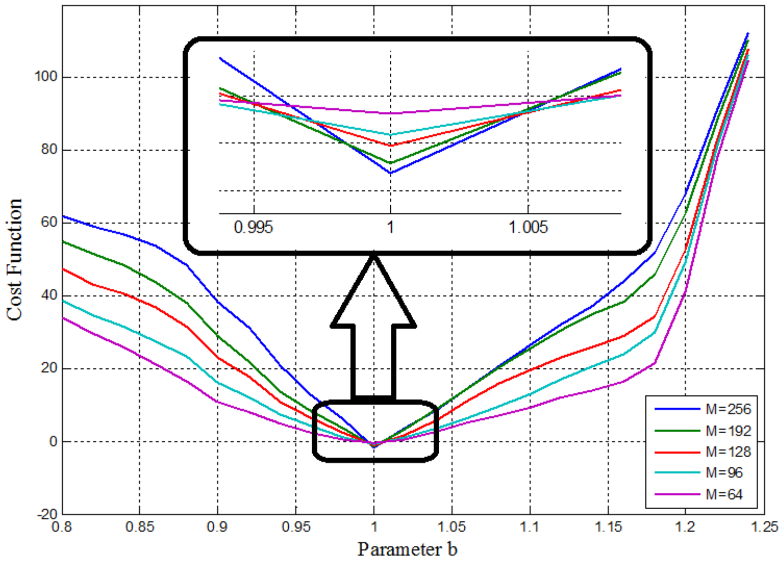


Figure 16. Cost function versus parameter b, with different number of GMM components (M) for the 1D parameter estimation method.

As can be seen, all of the cost functions show convex functions around the desired point. Therefore, they are acceptable for the parameter estimation methods. Specifically, they show the effect of changing the parameter of the model as a monotonically trend along with a global minimum at the exact expected value of the desired parameters $(a = 4.00,\ b = 1.00)$. Therefore, the GMM—based cost function has the desired ideal properties for the parameter estimation problem. Moreover, Figures 15 and 16 show the effect of increasing the number of GMM components, M, used in the GMM modeling. In this case, $M = 256$ represents better performance to identify the parameters $a\&b$.

In Figures 17 and 18, a contour plot of the cost function and its "cost surface" are respectively shown for the chaotic system (1) with $M = 256$ along with variation in the parameters, $a\&b$. They show dissimilarity between the real system attractor and each model attractor for a 2D parameter estimation problem. The minimum value of the point on those plots gives the parameters for the best model.

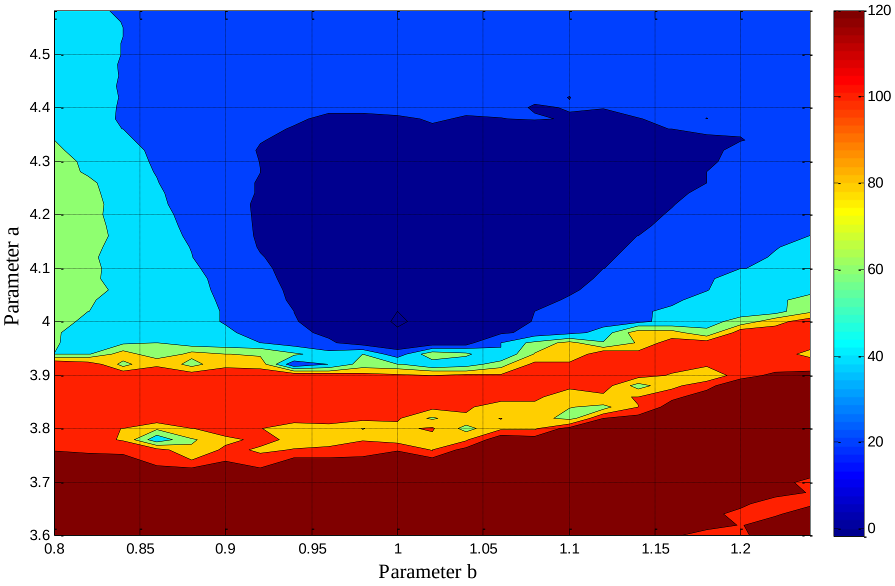


Figure 17. The contour plot of the GMM-based cost function for the introduced chaotic system ($M = 256$) along with variations in the parameters, $a\&b$.

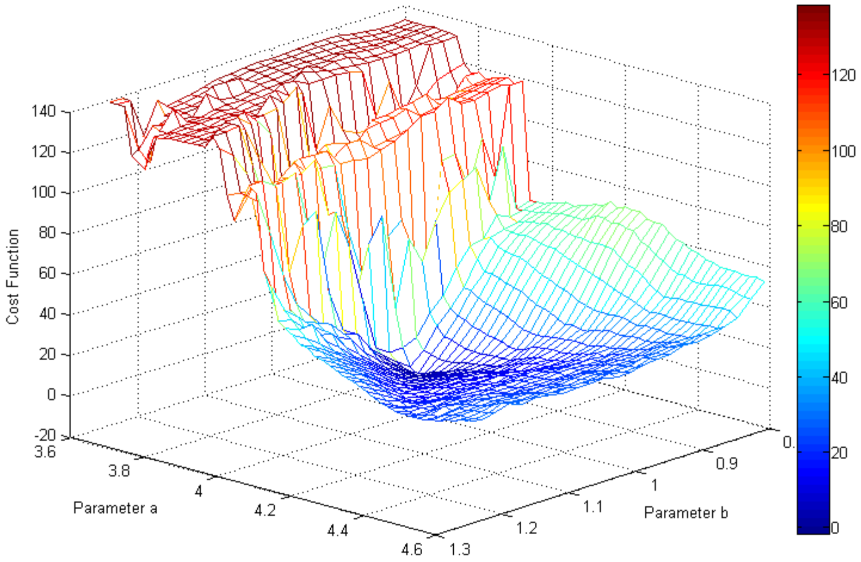


Figure 18. The "cost surface" of the GMM-based cost function for the introduced chaotic system ($M = 256$) along with variations in the parameters, $a\&b$.

As can be seen in Figures 17 and 18, the global minimum of the cost function is in the right place $(a = 4.00 \text{ and } b = 1.00)$. Furthermore, the surface of the cost function is almost convex near the best parameters, which makes it an easy case for any optimization approach that moves downhill.

In order to examine the efficiency of the cost function in the parameter estimation, two mentioned meta-heuristic optimization methods are applied. All the basic parameters, such as maximum number of iterations (50) and number of search agent (25), are the same in both algorithms. For further details about the algorithms and their particular parameters, see [96]. Comparison between the performances of MVO and WOA optimization algorithm is shown in Figure 19.

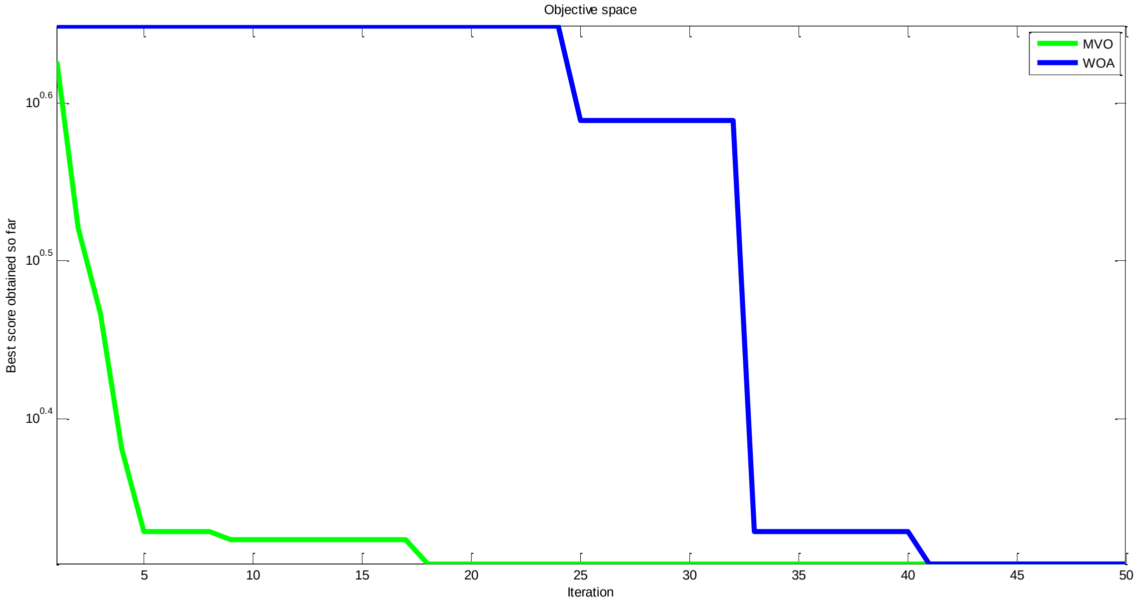


Figure 19. Comparison between the performances of the MVO and WOA optimization algorithm.

Based on the results of Figure 19, the MVO optimization method showed a superior performance in comparison with the WOA algorithms. In addition, Figure 20 represents the process of finding the best parameters using the WOA algorithm performed once for every 10 iterations. As can be seen, the individuals converge to the optimum area ($a = 4.00$ and $b = 1.00$).

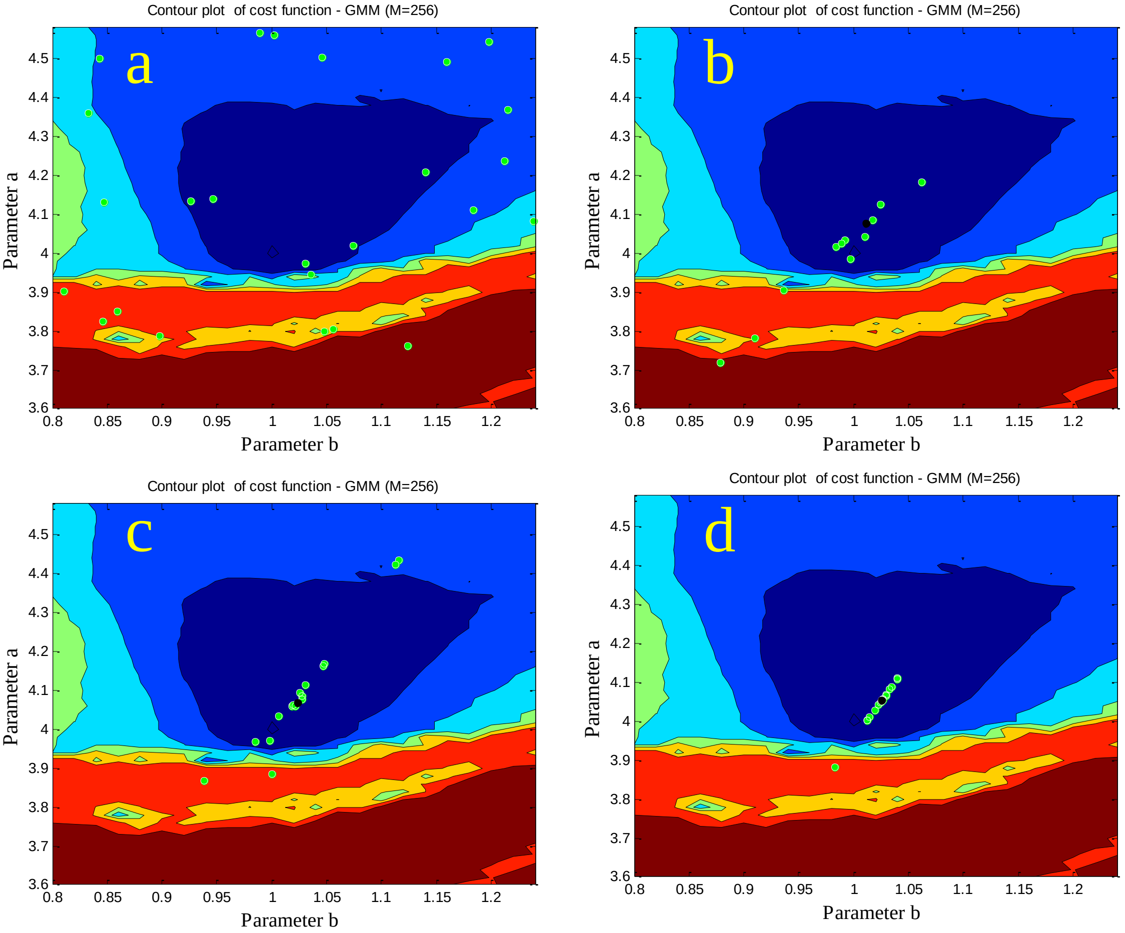


Figure 20. Process of finding the best parameters using the WOA algorithm. (**a**–**d**) represent the first, 10th, 20th, and 30th iteration, respectively.

7. Conclusions

In this paper, a new chaotic system has been investigated carefully through bifurcation, the largest Lyapunov exponent, ApEn, and stability analysis. Then, an engineering application of that system was proposed using a random number generator and its signal encryption application. After that, a GMM-based cost function was utilized in the parameter estimation of the chaotic circuit designed from the chaotic system. The cost function was based on the minimization of dissimilarity between the phase portrait obtained from the real system and that obtained from the model of the chaotic system. In order to minimize the cost function and to obtain the correct parameters, we used two new efficient optimization methods, the Whale Optimization Algorithm (WOA), and Multi-Verse Optimizer (MVO) algorithm. The MVO optimization method showed superior performance in comparison with the WOA algorithm.

Acknowledgments: This work was supported by the PhD launch fund of Hubei University of Technology (BSQD 2015024).

Author Contributions: Guanghui Xu and Yasser Shekofteh designed the study and wrote the paper. Akif Akgül and Shirin Panahi contributed to the experiment and algorithm design. Yasser Shekofteh and Chunbiao Li partially undertook the theoretical analysis and simulation work of the paper. All authors read and approved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, C.; Sprott, J.C. Variable-boostable chaotic flows. *Opt.-Int. J. Light Electron Opt.* **2016**, *127*, 10389–10398. [CrossRef]
2. Li, C.; Sprott, J.C.; Akgul, A.; Iu, H.H.; Zhao, Y. A new chaotic oscillator with free control. *Chaos Interdiscip. J. Nonlinear Sci.* **2017**, *27*, 083101. [CrossRef] [PubMed]
3. Jafari, M.A.; Mliki, E.; Akgul, A.; Pham, V.-T.; Kingni, S.T.; Wang, X.; Jafari, S. Chameleon: The most hidden chaotic flow. *Nonlinear Dyn.* **2017**, *88*, 1–15. [CrossRef]
4. Li, C.; Sprott, J.C.; Xing, H. Hypogenetic chaotic jerk flows. *Phys. Lett. A* **2016**, *380*, 1172–1177. [CrossRef]
5. Tlelo-Cuautle, E.; Carbajal-Gomez, V.; Obeso-Rodelo, P.; Rangel-Magdaleno, J.; Nuñez-Perez, J.C. FPGA realization of a chaotic communication system applied to image processing. *Nonlinear Dyn.* **2015**, *82*, 1879–1892. [CrossRef]
6. De la Fraga, L.G.; Torres-Pérez, E.; Tlelo-Cuautle, E.; Mancillas-López, C. Hardware implementation of pseudo-random number generators based on chaotic maps. *Nonlinear Dyn.* **2017**, *90*, 1661–1670. [CrossRef]
7. Pano-Azucena, A.D.; de Jesus Rangel-Magdaleno, J.; Tlelo-Cuautle, E.; de Jesus Quintas-Valles, A. Arduino-based chaotic secure communication system using multi-directional multi-scroll chaotic oscillators. *Nonlinear Dyn.* **2017**, *87*, 2203–2217. [CrossRef]
8. Valtierra, J.L.; Tlelo-Cuautle, E.; Rodríguez-Vázquez, Á. A switched-capacitor skew-tent map implementation for random number generation. *Int. J. Circuit Theory Appl.* **2017**, *45*, 305–315. [CrossRef]
9. García-Martínez, M.; Ontañón-García, L.; Campos-Cantón, E.; Čelikovský, S. Hyperchaotic encryption based on multi-scroll piecewise linear systems. *Appl. Math. Comput.* **2015**, *270*, 413–424. [CrossRef]
10. Tlelo-Cuautle, E.; Rangel-Magdaleno, J.; Pano-Azucena, A.; Obeso-Rodelo, P.; Nuñez-Perez, J.C. FPGA realization of multi-scroll chaotic oscillators. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *27*, 66–80. [CrossRef]
11. Danca, M.-F.; Kuznetsov, N. Hidden chaotic sets in a Hopfield neural system. *Chaos Solitons Fractals* **2017**, *103*, 144–150. [CrossRef]
12. Danca, M.-F.; Kuznetsov, N.; Chen, G. Unusual dynamics and hidden attractors of the Rabinovich–Fabrikant system. *Nonlinear Dyn.* **2017**, *88*, 791–805. [CrossRef]
13. Kuznetsov, N.; Leonov, G.; Yuldashev, M.; Yuldashev, R. Hidden attractors in dynamical models of phase-locked loop circuits: Limitations of simulation in MATLAB and SPICE. *Commun. Nonlinear Sci. Numer. Simul.* **2017**, *51*, 39–49.
14. Leonov, G.; Kuznetsov, N.; Mokaev, T. Hidden attractor and homoclinic orbit in Lorenz-like system describing convective fluid motion in rotating cavity. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *28*, 166–174. [CrossRef]
15. Leonov, G.A.; Kuznetsov, N.V.; Mokaev, T.N. Homoclinic orbits, and self-excited and hidden attractors in a Lorenz-like system describing convective fluid motion. *Eur. Phys. J. Spec. Top.* **2015**, *224*, 1421–1458. [CrossRef]
16. Sharma, P.; Shrimali, M.; Prasad, A.; Kuznetsov, N.; Leonov, G. Control of multistability in hidden attractors. *Eur. Phys. J. Spec. Top.* **2015**, *224*, 1485–1491. [CrossRef]
17. Sharma, P.R.; Shrimali, M.D.; Prasad, A.; Kuznetsov, N.; Leonov, G. Controlling Dynamics of Hidden Attractors. *Int. J. Bifurc. Chaos* **2015**, *25*, 1550061. [CrossRef]
18. Leonov, G.; Kuznetsov, N.; Vagaitsev, V. Localization of hidden Chua's attractors. *Phys. Lett. A* **2011**, *375*, 2230–2233. [CrossRef]
19. Leonov, G.; Kuznetsov, N.; Vagaitsev, V. Hidden attractor in smooth Chua systems. *Phys. D Nonlinear Phenom.* **2012**, *241*, 1482–1486. [CrossRef]
20. Leonov, G.A.; Kuznetsov, N.V. Hidden attractors in dynamical systems. From hidden oscillations in Hilbert–Kolmogorov, Aizerman, and Kalman problems to hidden chaotic attractor in Chua circuits. *Int. J. Bifurc. Chaos* **2013**, *23*, 1330002. [CrossRef]
21. Leonov, G.; Kuznetsov, N.; Kiseleva, M.; Solovyeva, E.; Zaretskiy, A. Hidden oscillations in mathematical model of drilling system actuated by induction motor with a wound rotor. *Nonlinear Dyn.* **2014**, *77*, 277–288. [CrossRef]

22. Dudkowski, D.; Jafari, S.; Kapitaniak, T.; Kuznetsov, N.V.; Leonov, G.A.; Prasad, A. Hidden attractors in dynamical systems. *Phys. Rep.* **2016**, *637*, 1–50. [CrossRef]
23. Tlelo-Cuautle, E.; de la Fraga, L.G.; Pham, V.-T.; Volos, C.; Jafari, S.; de Jesus Quintas-Valles, A. Dynamics, FPGA realization and application of a chaotic system with an infinite number of equilibrium points. *Nonlinear Dyn.* **2017**, *89*, 1–11. [CrossRef]
24. Pham, V.-T.; Volos, C.; Jafari, S.; Kapitaniak, T. Coexistence of hidden chaotic attractors in a novel no-equilibrium system. *Nonlinear Dyn.* **2017**, *87*, 2001–2010. [CrossRef]
25. Pham, V.-T.; Kingni, S.T.; Volos, C.; Jafari, S.; Kapitaniak, T. A simple three-dimensional fractional-order chaotic system without equilibrium: Dynamics, circuitry implementation, chaos control and synchronization. *AEU Int. J. Electron. Commun.* **2017**, *78*, 220–227. [CrossRef]
26. Pham, V.-T.; Jafari, S.; Volos, C.; Gotthans, T.; Wang, X.; Hoang, D.V. A chaotic system with rounded square equilibrium and with no-equilibrium. *Opt.-Int. J. Light Electron Opt.* **2017**, *130*, 365–371. [CrossRef]
27. Pham, V.-T.; Volos, C.; Gambuzza, L.V. A memristive hyperchaotic system without equilibrium. *Sci. World J.* **2014**, *2014*. [CrossRef] [PubMed]
28. Jafari, S.; Sprott, J.C.; Hashemi Golpayegani, S.M.R. Elementary quadratic chaotic flows with no equilibria. *Phys. Lett. A* **2013**, *377*, 699–702. [CrossRef]
29. Pham, V.-T.; Jafari, S.; Volos, C.; Wang, X.; Hashemi Golpayegani, S.M.R. Is that really hidden? The presence of complex fixed-points in chaotic flows with no equilibria. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450146. [CrossRef]
30. Pham, V.-T.; Volos, C.; Jafari, S.; Wang, X. Generating a novel hyperchaotic system out of equilibrium. *Optoelectron. Adv. Mater.-Rapid Commun.* **2014**, *8*, 535–539.
31. Pham, V.-T.; Volos, C.; Jafari, S.; Wei, Z.; Wang, X. Constructing a novel no-equilibrium chaotic system. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450073. [CrossRef]
32. Tahir, F.R.; Jafari, S.; Pham, V.-T.; Volos, C.; Wang, X. A Novel No-Equilibrium Chaotic System with Multiwing Butterfly Attractors. *Int. J. Bifurc. Chaos* **2015**, *25*, 1550056. [CrossRef]
33. Pham, V.-T.; Jafari, S.; Kapitaniak, T.; Volos, C.; Kingni, S.T. Generating a Chaotic System with One Stable Equilibrium. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750053. [CrossRef]
34. Wang, X.; Pham, V.-T.; Jafari, S.; Volos, C.; Munoz-Pacheco, J.M.; Tlelo-Cuautle, E. A new chaotic system with stable equilibrium: From theoretical model to circuit implementation. *IEEE Access* **2017**, *5*, 8851–8858. [CrossRef]
35. Kingni, S.T.; Pham, V.-T.; Jafari, S.; Woafo, P. A chaotic system with an infinite number of equilibrium points located on a line and on a hyperbola and its fractional-order form. *Chaos Solitons Fractals* **2017**, *99*, 209–218. [CrossRef]
36. Jafari, S.; Sprott, J.C. Simple chaotic flows with a line equilibrium. *Chaos Solitons Fractals* **2013**, *57*, 79–84. [CrossRef]
37. Pham, V.-T.; Jafari, S.; Volos, C. A novel chaotic system with heart-shaped equilibrium and its circuital implementation. *Opt.-Int. J. Light Electron Opt.* **2017**, *131*, 343–349. [CrossRef]
38. Pham, V.T.; Volos, C.; Kapitaniak, T.; Jafari, S.; Wang, X. Dynamics and circuit of a chaotic system with a curve of equilibrium points. *Int. J. Electron.* **2017**, *105*, 1–13. [CrossRef]
39. Pham, V.-T.; Jafari, S.; Wang, X.; Ma, J. A Chaotic System with Different Shapes of Equilibria. *Int. J. Bifurc. Chaos* **2016**, *26*, 1650069. [CrossRef]
40. Pham, V.-T.; Jafari, S.; Volos, C.; Giakoumis, A.; Vaidyanathan, S.; Kapitaniak, T. A chaotic system with equilibria located on the rounded square loop and its circuit implementation. *IEEE Trans. Circuits Syst. II Express Br.* **2016**, *63*, 878–882. [CrossRef]
41. Pham, V.-T.; Jafari, S.; Volos, C.; Vaidyanathan, S.; Kapitaniak, T. A chaotic system with infinite equilibria located on a piecewise linear curve. *Opt.-Int. J. Light Electron Opt.* **2016**, *127*, 9111–9117. [CrossRef]
42. Jafari, S.; Sprott, J.C.; Molaie, M. A simple chaotic flow with a plane of equilibria. *Int. J. Bifurc. Chaos* **2016**, *26*, 1650098. [CrossRef]
43. Jafari, S.; Sprott, J.C.; Pham, V.-T.; Volos, C.; Li, C. Simple chaotic 3D flows with surfaces of equilibria. *Nonlinear Dyn.* **2016**, *86*, 1349–1358. [CrossRef]
44. Rajagopal, K.; Akgul, A.; Jafari, S.; Karthikeyan, A.; Koyuncu, I. Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. *Chaos Solitons Fractals* **2017**, *103*, 476–487. [CrossRef]

45. Rajagopal, K.; Jafari, S.; Laarem, G. Time-delayed chameleon: Analysis, synchronization and FPGA implementation. *Pramana* **2017**, *89*, 92. [CrossRef]
46. Pham, V.-T.; Wang, X.; Jafari, S.; Volos, C.; Kapitaniak, T. From Wang–Chen System with Only One Stable Equilibrium to a New Chaotic System without Equilibrium. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750097. [CrossRef]
47. Pham, V.-T.; Volos, C.; Jafari, S.; Vaidyanathan, S.; Kapitaniak, T.; Wang, X. A chaotic system with different families of hidden attractors. *Int. J. Bifurc. Chaos* **2016**, *26*, 1650139. [CrossRef]
48. Nazarimehr, F.; Saedi, B.; Jafari, S.; Sprott, J.C. Are perpetual points sufficient for locating hidden attractors? *Int. J. Bifurc. Chaos* **2017**, *28*, 1750037. [CrossRef]
49. Dudkowski, D.; Prasad, A.; Kapitaniak, T. Perpetual Points: New Tool for Localization of Coexisting Attractors in Dynamical Systems. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750063. [CrossRef]
50. Faure, P.; Korn, H. Is there chaos in the brain? I. Concepts of nonlinear dynamics and methods of investigation. *C. R. l'Acad. Sci.-Seri. III-Sci.* **2001**, *324*, 773–793. [CrossRef]
51. Korn, H.; Faure, P. Is there chaos in the brain? II. Experimental evidence and related models. *C. R. Biol.* **2003**, *326*, 787–840. [CrossRef] [PubMed]
52. Molaie, M.; Falahian, R.; Gharibzadeh, S.; Jafari, S.; Sprott, J.C. Artificial neural networks: Powerful tools for modeling chaotic behavior in the nervous system. *Front. Comput. Neurosci.* **2014**, *8*. [CrossRef] [PubMed]
53. Falahian, R.; Mehdizadeh Dastjerdi, M.; Molaie, M.; Jafari, S.; Gharibzadeh, S. Artificial neural network-based modeling of brain response to flicker light. *Nonlinear Dyn.* **2015**, *81*, 1951–1967. [CrossRef]
54. Jafari, S.; Sprott, J.C.; Hashemi Golpayegani, S.M.R. Layla and Majnun: A complex love story. *Nonlinear Dyn.* **2016**, *83*, 615–622. [CrossRef]
55. Aram, Z.; Jafari, S.; Ma, J.; Sprott, J.C.; Zendehrouh, S.; Pham, V.-T. Using chaotic artificial neural networks to model memory in the brain. *Commun. Nonlinear Sci. Numer. Simul.* **2017**, *44*, 449–459. [CrossRef]
56. Hilborn, R.C. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*; Oxford University Press: Oxford, UK, 2000.
57. Jafari, S.; Hashemi Golpayegani, S.M.R.; Daliri, A. Comment on 'Parameters identification of chaotic systems by quantum-behaved particle swarm optimization' [Int. J. Comput. Math. 86(12) (2009), pp. 2225–2235]. *Int. J. Comput. Math.* **2013**, *90*, 903–905. [CrossRef]
58. Jafari, S.; Hashemi Golpayegani, S.M.R.; Rasoulzadeh Darabad, M. Comment on "Parameter identification and synchronization of fractional-order chaotic systems" [Commun Nonlinear Sci Numer Simulat 2012; 17: 305–16]. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 811–814. [CrossRef]
59. Jafari, S.; Hashemi Golpayegani, S.M.R.; Jafari, A.H.; Gharibzadeh, S. Some remarks on chaotic systems. *Int. J. Gen. Syst.* **2012**, *41*, 329–330. [CrossRef]
60. He, Q.; Wang, L.; Liu, B. Parameter estimation for chaotic systems by particle swarm optimization. *Chaos Solitons Fractals* **2007**, *34*, 654–661. [CrossRef]
61. Tang, Y.; Guan, X. Parameter estimation for time-delay chaotic system by particle swarm optimization. *Chaos Solitons Fractals* **2009**, *40*, 1391–1398. [CrossRef]
62. Wang, L.; Xu, Y. An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems. *Expert Syst. Appl.* **2011**, *38*, 15103–15109. [CrossRef]
63. Weile, D.S.; Michielssen, E. Genetic algorithm optimization applied to electromagnetics: A review. *IEEE Trans. Antennas Propag.* **1997**, *45*, 343–353. [CrossRef]
64. Kennedy, J. Particle swarm optimization. In *Encyclopedia of the Sciences of Learning*; Springer: Berlin, Germany, 2011; pp. 760–766.
65. Yao, X.; Liu, Y. Fast Evolutionary Programming. *Evolut. Program.* **1996**, *3*, 451–460.
66. Arı, Ç.; Aksoy, S.; Arıkan, O. Maximum likelihood estimation of Gaussian mixture models using stochastic search. *Pattern Recognit.* **2012**, *45*, 2804–2816. [CrossRef]
67. Povinelli, R.J.; Johnson, M.T.; Lindgren, A.C.; Roberts, F.M.; Ye, J. Statistical models of reconstructed phase spaces for signal classification. *IEEE Trans. Signal Process.* **2006**, *54*, 2178–2186. [CrossRef]
68. Shekofteh, Y.; Almasganj, F. Feature extraction based on speech attractors in the reconstructed phase space for automatic speech recognition systems. *ETRI J.* **2013**, *35*, 100–108. [CrossRef]
69. Shekofteh, Y.; Almasganj, F.; Daliri, A. MLP-based isolated phoneme classification using likelihood features extracted from reconstructed phase space. *Eng. Appl. Artif. Intell.* **2015**, *44*, 1–9. [CrossRef]
70. Lao, S.-K.; Shekofteh, Y.; Jafari, S.; Sprott, J.C. Cost function based on gaussian mixture model for parameter estimation of a chaotic circuit with a hidden attractor. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450010. [CrossRef]

71. Shekofteh, Y.; Jafari, S.; Sprott, J.C.; Golpayegani, S.M.R.H.; Almasganj, F. A gaussian mixture model based cost function for parameter estimation of chaotic biological systems. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *20*, 469–481. [CrossRef]
72. Jafari, S.; Sprott, J.C.; Pham, V.-T.; Hashemi Golpayegani, S.M.R.; Jafari, A.H. A New Cost Function for Parameter Estimation of Chaotic Systems Using Return Maps as Fingerprints. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450134. [CrossRef]
73. Kuznetsov, N.; Mokaev, T.; Vasilyev, P. Numerical justification of Leonov conjecture on Lyapunov dimension of Rossler attractor. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1027–1034. [CrossRef]
74. Leonov, G.; Kuznetsov, N.; Mokaev, T. Homoclinic orbit and hidden attractor in the Lorenz-like system describing the fluid convection motion in the rotating cavity. *arXiv* **2014**, arXiv:1412.7667.
75. Kuznetsov, N.; Leonov, G.; Mokaev, T. The Lyapunov dimension and its computation for self-excited and hidden attractors in the Glukhovsky-Dolzhansky fluid convection model. *arXiv* **2015**, arXiv:1509.09161v2.
76. Leonov, G.; Kuznetsov, N.; Mokaev, T. The Lyapunov dimension formula of self-excited and hidden attractors in the Glukhovsky-Dolzhansky system. *arXiv* **2015**, arXiv:1509.09161v1.
77. Kuznetsov, N. The Lyapunov dimension and its estimation via the Leonov method. *Phys. Lett. A* **2016**, *380*, 2142–2149. [CrossRef]
78. Wolf, A.; Swift, J.B.; Swinney, H.L.; Vastano, J.A. Determining Lyapunov exponents from a time series. *Phys. D Nonlinear Phenom.* **1985**, *16*, 285–317. [CrossRef]
79. Pincus, S.M. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [CrossRef] [PubMed]
80. Pincus, S. Approximate entropy (ApEn) as a complexity measure. *Chaos Interdiscip. J. Nonlinear Sci.* **1995**, *5*, 110–117. [CrossRef] [PubMed]
81. Chon, K.H.; Scully, C.G.; Lu, S. Approximate entropy for all signals. *IEEE Eng. Med. Biol. Mag.* **2009**, *28*. [CrossRef]
82. Koyuncu, İ.; Özcerit, A.T. The design and realization of a new high speed FPGA-based chaotic true random number generator. *Comput. Electr. Eng.* **2016**, *58*, 203–214. [CrossRef]
83. Akgul, A.; Moroz, I.; Pehlivan, I.; Vaidyanathan, S. A new four-scroll chaotic attractor and its engineering applications. *Opt.-Int. J. Light Electron Opt.* **2016**, *127*, 5491–5499. [CrossRef]
84. Çavuşoğlu, Ü.; Akgül, A.; Kaçar, S.; Pehlivan, İ.; Zengin, A. A novel chaos-based encryption algorithm over TCP data packet for secure communication. *Secur. Commun. Netw.* **2016**, *9*, 1285–1296. [CrossRef]
85. Avaroğlu, E.; Koyuncu, İ.; Özer, A.B.; Türk, M. Hybrid pseudo-random number generator for cryptographic systems. *Nonlinear Dyn.* **2015**, *82*, 239–248. [CrossRef]
86. Akgul, A.; Calgan, H.; Koyuncu, I.; Pehlivan, I.; Istanbullu, A. Chaos-based engineering applications with a 3D chaotic system without equilibrium points. *Nonlinear Dyn.* **2016**, *84*, 481–495. [CrossRef]
87. Rukhin, A.; Soto, J.; Nechvatal, J.; Barker, E.; Leigh, S.; Levenson, M.; Banks, D.; Heckert, A.; Dray, J.; Vo, S. *Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; NIST Special Publication; Booz-Allen and Hamilton Inc.: McLean, VA, USA, 2010.
88. Trejo-Guerra, R.; Tlelo-Cuautle, E.; Jimenez-Fuentes, J.; Sánchez-López, C.; Muñoz-Pacheco, J.; Espinosa-Flores-Verdad, G.; Rocha-Pérez, J. Integrated circuit generating 3-and 5-scroll attractors. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4328–4335. [CrossRef]
89. Kantz, H.; Schreiber, T. *Nonlinear Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2004; Volume 7.
90. Bishop, C.M. Pattern recognition. *Mach. Learn.* **2006**, *128*, 1–58.
91. Nakagawa, S.; Wang, L.; Ohtsuka, S. Speaker identification and verification by combining MFCC and phase information. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 1085–1095. [CrossRef]
92. Yang, X.-S. *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
93. De la Fraga, L.G.; Tlelo-Cuautle, E. Optimizing the maximum Lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators. *Nonlinear Dyn.* **2014**, *76*, 1503–1515. [CrossRef]
94. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]

95. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]

96. Tang, W.; Wu, Q. Biologically inspired optimization: A review. *Trans. Inst. Meas. Control* **2009**, *31*, 495–515. [CrossRef]