Design and synchronization of large-scale random number generators

Teh-Lu Liao¹, Pei-Yen Wan¹, Shih-Che Tseng², Jun-Juh Yan^{2*}

¹Department of Engineering Science, National Cheng Kung University
Tainan 701, Taiwan, R.O.C.

^{2*} Department of Computer and Communication, Shu-Te University,
Kaohsiung 824, Taiwan, R.O.C.

*Corresponding Author: jjyan@stu.edu.tw
(Tel: +886-7-6158000 #4806)

Abstract

This paper is concerned with the design of synchronized large-scale random number generators. First, we introduce some parameters into the chaotic Duffing map system to regulate and obtain richer chaotic state responses. Then according to the butterfly effect of chaos systems, we implement many adjustable Duffing map systems in the microcontroller and complete the design of large-scale random number generators. Next, a controller is proposed to complete the synchronization of the master-slave large-scale random number generators.

Keywords: Duffing map system, butterfly effect, large-scale random number generators, discrete sliding mode control.

1. Introduction

Chaotic system is a complex nonlinear system. In recent years, it has attracted extensive attention in the field of engineering research. The reason is that chaotic systems have very rich dynamic behavior, unpredictable trajectory, white noise-like broadband and the initial value sensitivity of the butterfly effect. In 1990, O.G.Y (Ott, Grebogi and Yorke)[1] proposed the study of chaos control. Researches indicate that, when parameter values of chaotic systems are slightly adjusted, the chaotic behavior will produce huge and unpredictable changes. Although the chaos behavior might be undesired in most mechanical systems, the noise-like behavior of chaos is suitable to the application of secure communication [2]. As a result, several feedback control methods leading to suppression or synchronization of chaos have been proposed in the literature, such as OGY method, passive control [3], impulsive control [4], sliding mode control [5].

In these two decades of research, the vast practicality of chaotic systems has been applied in the fields of communication, medicine and biology, and has solved several important engineering problems, especially in the security field of communication systems for data confidential transmission is one of the most

popular research. Recently, due to the appearance of quantum computers, many traditional encryption methods might become unsafe and many scholars have indicated that chaotic encryption method may be one of the solutions. Because the chaos system has rich dynamics, it can generate huge random signals in a very short time. The encryption method designed by the chaotic system is difficult to decrypt even with a quantum computer. Therefore, to generate large-scale random number generators is an import issue for resisting the attack of quantum computers.

For the above reasons, in this paper, we study the design of large-scale random number generators and its synchronization for the application to secure communication. First, we introduce a modulation approach by introducing some parameters into the chaotic Duffing map system to regulate and obtain richer chaotic state responses, which can be effectively applied to all general classes of discrete chaotic systems. Then we implement many adjustable Duffing map systems in the microcontroller just with different initial conditions. According to the butterfly effect of chaos systems, we can complete the design of large-scale random number generators. Next, a discrete sliding mode scheme is proposed to solve the synchronization problem of the master-slave large-scale random number generators.

2. Design of a large-scale random number generator

In this paper, we will discuss the design and realization of a large-scale random number generators. To complete the large-scale random number generators, in the following, we first introduce the modulation of the Duffing map chaotic system such that richer random chaotic state responses can be obtained for large-scale random number generators.

The proposed Duffing map system is described by the following system

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= -0.2x_1(k) + 2.75x_2(k) - x_2^3(k) \end{aligned} \quad (1)$$

, where $x_1, x_2(k)$ are the system states. In order to adjust the amplitude and DC offset of system state responses, we let

$$y_i(k) = a_i x_i(k) + d_i \quad (2)$$

where $a_i, i=1,2$ are used to adjust the amplitude and the DC offset of states, respectively. From equation (2), we have

$$x_i(k) = \frac{y_i(k) - d_i}{a_i}, i=1,2 \quad (3)$$

By (1) and (3), a new discrete Duffing map chaotic system with adjustable parameters can be obtained as follows

$$\begin{aligned} y_1(k+1) &= \lambda_1 y_2(k) - \lambda_2 \\ y_2(k+1) &= \beta_1 y_1(k) + \beta_2 y_2^3 + \beta_3 y_2^2 + \beta_4 y_2 + \beta_5 \end{aligned} \quad (4)$$

, where

$$\begin{aligned} \lambda_1 &= \frac{a_1}{a_2}, \lambda_2 = -\frac{a_1}{a_2} d_2 + d_1, \beta_1 = -\frac{0.2a_2}{a_1} \\ \beta_2 &= -\frac{1}{a_2^2}, \beta_3 = \frac{3d_2}{a_2^2}, \beta_4 = 2.75 - \frac{3d_2^2}{a_2^2}, \beta_5 = \frac{0.2a_2 d_1}{a_1} - 2.75d_2 + \frac{d_2^3}{a_2^2} + d_2 \end{aligned}$$

Numerical simulations for the adjustable Duffing map system

For evaluating the modulation of the system amplitude and DC offset of random number generator, we give the following simulation analysis. In the simulation, the modulation parameters are given as $a_1=2, a_2=4, d_1=-5, d_2=5$ and the initial conditions are selected as $x_1=0.4, x_2=-0.2$. The simulation results are given in Figures 1-3

Fig.1. The states of the original Duffing map chaotic system (before modulation).

Fig.2. The states of the adjusted Duffing map chaotic system (after modulation).

Fig.3. The strange attractors of the original and adjusted Duffing map chaotic systems, respectively (left is before modulation, right is after modulation).

From discussed above, the random number generator can be implemented by using the adjusted Duffing map chaotic system. As well known, the state response of a chaotic system is very sensitive to the initial value, and different initial values will lead to a great difference in state responses of the system. This is the butterfly effect. Therefore, we can utilize the butterfly effect to design large-scale random number generators. In addition, due to the high capacity and fast computing speed of the digital microcontroller, we can implement this large-scale random number generator with microcontroller, and can get huge random numbers in the very short time according to the computing speed of the microcontroller. To show the butterfly effect, we use three sets of random generators (Duffing map systems) for comparison. x_1, x_2 is the random number of the first random number generator, the second group is x_{11}, x_{12} and the third group is x_{21}, x_{22} . The initial values of each group are setting very close to observe the butterfly effect. The initial conditions are selected as $(x_{10}, x_{20}) = (0.9, -0.5)$, $(x_{110}, x_{120}) = (0.900001, -0.500012)$, $(x_{210}, x_{220}) = (0.9000011, -0.500001)$, respectively. From the simulation results in Fig.4, it can be found that the initial values with very small difference, as expected, will produce completely different random state responses.

Fig.4. Butterfly effect

Now we are ready to implement a large-scale random number generator using a microcontroller. Its structure is designed as shown in Fig.5. In Fig. 5, we construct n Duffing maps with the same structure in the microcontroller. Each Duffing map has different initial values. According to the butterfly effect mentioned above, we can get $2N$ random numbers. It is worth mentioning that the microcontrollers have a large amount of program memory capacity and fast computing speed, which means that we can use the

microcontroller to complete the large-scale random number generator and obtain a huge amount of random numbers in a very short period time.

Fig.5. The structure of a large-scale random number generator

In the following, we continue to discuss the realization of a large-scale random number generator. First, we construct ten sets of random number generators by using the HT32F1765 microcontroller as shown in Fig. 6. Due to the butterfly effect, the output responses of the random number generators of each set will be completely different. When we set up ten sets of random number generators with different initial values, we can get the twenty random numbers, and then according to the IEEE 754 standard specifies, each random number is with 32 bits. Therefore, we can get 640 binary random numbers ($10 \times 32 \times 2$), so we can write multiple sets of random number generators according to the memory size of the microcontroller to get huge random numbers. By using 8x8 LED matrix, Fig.7 shows 256 random binary signals obtain from the proposed large-scale random number generators.

Fig.6. Holtek 32-bit microcontroller.

Fig.7. 256 random binary signals.

3. Synchronization of master-slave large-scale random number generators

We first consider the design of a single master-slave random number generator. The master-slave random number generators are defined as below.
Master random number generator:

$$\begin{aligned} x_1(k+1) &= \lambda_1 x_2(k) - \lambda_2 \\ x_2(k+1) &= \beta_1 x_1(k) + \beta_2 x_2^3(k) + \beta_3 x_2^2(k) + \beta_4 x_2(k) + \beta_5 \end{aligned} \quad (5)$$

Slave random number generator:

$$\begin{aligned} y_1(k+1) &= \lambda_1 y_2(k) - \lambda_2 \\ y_2(k+1) &= \beta_1 y_1(k) + \beta_2 y_2^3(k) + \beta_3 y_2^2(k) + \beta_4 y_2(k) + \beta_5 + u(k) \end{aligned} \quad (6)$$

, where $x_i, i=1,2$ and $y_i, i=1,2$ are the state variables of the master and slave systems. $u(k) \in R$ is the control input introduced to achieve synchronization between the master and slave random number generators. By defining $e_i(k) = y_i(k) - x_i(k), i=1,2$, the error dynamics can be described by the following equation:

$$\begin{aligned} e_1(k+1) &= \lambda_1 e_2(k) \\ e_2(k+1) &= \beta_1 e_1(k) + \beta_2 (y_2^3(k) - x_2^3(k)) + \beta_3 e_2(k)(y_2(k) + x_2(k)) + \beta_4 e_2(k) + u(k) \end{aligned} \quad (7)$$

In the following, the discrete sliding mode control (DSMC) is utilized to achieve the synchronization. Generally speaking, the DSMC design is composed of two steps. To achieve the synchronization based on DSMC, a switching function is selected as:

$$s(k) = e_2(k) + \alpha e_1(k) \quad (8)$$

Assuming the error dynamics are already on the sliding manifold ($s(k)=0$), we have

$$e_2(k) = -\alpha e_1(k) \quad (9)$$

Substituting (9) into (7), we can obtain

$$e_1(k+1) = -\alpha \lambda_1 e_1(k) \quad (10)$$

From (10), we can see that if α is specified to satisfied $|\alpha \lambda_1| < 1$, then e_1 converges to zero. Furthermore, since $e_2(k) = -\alpha e_1(k)$ in the sliding manifold, we obtain $e_2 = 0$ when $e_1 = 0$.

After discussing the selection of the switching surface, we still have to design the controller to ensure that the system can smoothly enter the sliding manifold. The controller design is described as follows. According to (7) and (8), we have

$$\begin{aligned} s(k+1) &= e_2(k+1) + \alpha e_1(k+1) \\ &= \beta_1 e_1(k) + \beta_2 (y_2^3(k) - x_2^3(k)) + \beta_3 e_2(k) \\ &\quad (y_2(k) + x_2(k)) + \beta_4 e_2(k) + u(k) + \alpha \lambda_1 e_2(k) \end{aligned} \quad (11)$$

If the controller $u(k)$ is properly designed as:

$$u(k) = -(\beta_1 e_1(k) + \beta_2 (y_2^3(k) - x_2^3(k)) + \beta_3 e_2(k)(y_2(k) + x_2(k)) + \beta_4 e_2(k) + \alpha \lambda_1 e_2(k)) + \gamma s(k) \quad (12)$$

, where $|\gamma| < 1$. Substituting (12) into (11), we can get

$$s(k+1) = \gamma s(k) \quad (13)$$

Since $|\gamma| < 1$, the error system can smoothly enter the sliding manifold, (i.e. $\lim_{k \rightarrow \infty} s(k) = 0$).

In the following, for evaluating the synchronization effect of master and slave random number generators, the parameters are given as $a_1=1, a_2=5, d_1=3, d_2=-3$. Therefore, the sliding mode control law can be obtained by (8) with $\alpha=0.3, \gamma=0.2$. In numerical simulations, the initial conditions are selected as $x_{10}=3.1, x_{20}=-4.5, y_{10}=-3.3, y_{20}=-4$. The simulation results are shown in Figs.8-11. Fig.8 shows the corresponding state responses of master and controlled slave random number generators. Fig.9 shows the error response between master and controlled slave random number generators. From the simulation results, it shows the proposed sliding mode control works well and the chaotic behavior of controlled master and slave random number generator can be asymptotically synchronized.

Fig.8 The state responses of the controlled master and slave random number generators.

Fig.9 The error responses of the controlled master and slave random number generators

In order to reduce the design complexity of synchronization controller for the master-slave large-scale random number generator, we divide the controller $u(k)$ (12) into two parts, $u_m(k)$ and $u_s(k)$ to satisfy $u(k) = u_m(k) + u_s(k)$, where $u_m(k)$ and $u_s(k)$ are the combination signals of master and slave random number generator, respectively.

After completing the design of synchronized master-slave random number generators, the large-scale synchronized master-slave random number generators is given as shown in Fig. 10.

Fig.10 Synchronized large-scale random number generators.

In Fig. 10, First, the master $RNG_i, i=1,2,...,n$, sequentially generates the information $u_{mi}(k)$ required by each RNG_i synchronization controller, and in the slave $RNG_i, i=1,2,...,n$, also sequentially generates the required information $u_{si}(k)$. $u_{mi}(k)$ and $u_{si}(k)$ are, respectively, time-sharing selected by the selector in the master and slave side., then $u_{mi}(k)$ and $u_{si}(k)$ is integrated to form the aforementioned controller $u_i(k) = u_{mi}(k) + u_{si}(k)$ and then the synchronization between master RNG_i and slave $RNG_i, i=1,2,...,n$ can be guaranteed.

Conclusions

This paper proposes the design and synchronization for large-scale random number generators. The proposed large-scale random number generators can adjust the amplitude and DC offset by some introduced parameters and can produce a huge amount of random numbers. The the discrete sliding mode control method is used to solve the synchronization problem of the master and slave large-scale random number generators. Throughout the paper, the simulation and the experimental results verify that the methods are correct.

References

- [1] E. Ott, C. Grebogi, and J. A. Yorke : "Controlling Chaos, Pages" 77-80, 1996
- [2] M. Z. D. Hoz, L. Acho, and Y. Vidal : "A modified Chua chaotic oscillator and its application to secure communications", Applied Mathematics and Computation, 247(15), 712-722, 2014
- [3] X. J. Wu, J. S. Liu, and G. R. Chen : "Chaos synchronization of Rikitake chaotic attractor using the passive control technique", Nonlinear Dynamics 53(1), 45-53, 2008
- [4] S. Chen, Q. Yang, and C. Wang : "Impulsive control and synchronization of unified chaotic system", Chaos Solitons Fractals 20(4), 153-160, 2004
- [5] M. C. Pai : "Global synchronization of uncertain chaotic systems via discrete-time sliding mode control", Applied Mathematics and Computation, 228, 663-671, 2014