mathematics

MDPI

Article

Secure Data Transmission and Image Encryption Based on a Digital-Redesign Sliding Mode Chaos Synchronization

Jiunn-Shiou Fang [1], Jason Sheng-Hong Tsai [1], Jun-Juh Yan [2,*], Li-Huseh Chiang [1] and Shu-Mei Guo [3]

[1] Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan; fjshow611@gmail.com (J.-S.F.); shtsai@mail.ncku.edu.tw (J.S.-H.T.); Chiang@stu.edu.tw (L.-H.C.)
[2] Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41107, Taiwan
[3] Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan; guosm@mail.ncku.edu.tw
* Correspondence: jjyan@ncut.edu.tw

Abstract: In this paper, a novel image encryption algorithm based on chaotic synchronization is proposed. First, a digital-redesign sliding mode controller (SMC) is developed to guarantee the chaos synchronization. The digital redesign method makes it possible to transform a proposed continuous-time SMC to discrete-time SMC whilst maintaining the performance of the robust synchronization. Then, the secret keys are embedded in the state equations of the master chaotic system, such that the secret keys do not appear in the public channel, and utilize the chaotic synchronization to achieve secure communication for transmitting the secret keys from transmitter to receiver. Second, an image encryption algorithm integrating the S-box with chaotic synchronization is established, where the S-box is created by the secret key transmitted from the transmitter. Finally, a detailed analysis of the image encryption algorithm based on chaos synchronization is included to verify the feasibility and effectiveness of this proposed approach.

Keywords: image encryption; digital redesign; sliding mode control; synchronization; S-box

Citation: Fang, J.-S.; Tsai, J.S.-H.; Yan, J.-J.; Chiang, L.-H.; Guo, S.-M. Secure Data Transmission and Image Encryption Based on a Digital-Redesign Sliding Mode Chaos Synchronization. *Mathematics* **2022**, *10*, 518. https://doi.org/10.3390/math10030518

Academic Editor: Xinsong Yang

Received: 3 January 2022
Accepted: 4 February 2022
Published: 5 February 2022

1. Introduction

Chaotic systems are nonlinear systems with many complex characteristics. Due to the randomness of chaotic systems, they can be applied in many aspects, especially in secure communication. For application in secure communication, chaos synchronization is the most important issue, and some control methods for chaos synchronization have been developed in the literature. In the report [1], the adaptive fuzzy control approach is designed to deal with the synchronization for time-delay uncertainty chaotic systems. In [2], the sliding mode control is utilized in the process of synchronization for the chaotic system with disturbances. It is well known that the sliding mode control (SMC) is a nonlinear control method using a discontinuous control signal to force state trajectories to hit the switching surface and enter the sliding manifold such that the Lyapunov stability for the controlled systems can be ensured. Moreover, SMC is an effective method to eliminate the influence of the matched disturbances [3–5], and the design of SMC often combines with the disturbance estimator or observer to achieve better robustness [6–8]. Due to this reason, SMC has been widely used to solve the robust control problems for many problems in engineering. However, there is a chattering phenomenon due to the utilization of sign function in SMC, which causes the high-frequency oscillation in the controller. To solve this problem, there are some alternatives, such as high-order SMC [9,10], second-order SMC [11,12], and saturation function [13]. Thus, in this paper, we also introduce the saturation function to avoid the chattering phenomenon and achieve the chaos synchronization.

To develop the digital-redesign SMC, we introduce the digital redesign approach to transform a well-designed continuous-time SMC to a corresponding discrete-time SMC

directly and keep the control performance. There have been many control laws designed with the digital-redesign method, for example, the H_∞ and H_2 continuous-time controllers are redesigned to the discrete-time controllers by the digital-redesign method [14]. In [15], the authors also presented a new digital-redesign control scheme applied to a high-gain analogy controller and improve the state responses. Thus, one can know that the digital redesign method is an effective approach to discretize the continuous-time controller and maintain the performance of the designed continuous controller as possible.

Chaotic synchronization methods are often used in secure communications by embedding the private information in the state equations of the transmitter and synchronizing the states on both sides to achieve the secure communication [16]. However, secure communication with the chaotic system may be destroyed by noise and external disturbances [17], in other words, the response of the chaotic system is sensitive to the initial conditions; therefore, the proposed digital-redesign SMC-based controller utilizes this property to reconstruct the transmission message for achieving secure communication. In [18], the sliding mode observer is designed for the secure communication and recovery of the desired message from the chaos trajectory. In [19], an adaptive terminal sliding mode tracking scheme is proposed for synchronizing the chaotic systems. Therefore, we combine the chaos secure communication with the SMC to the image encryption algorithm, and the designed secure key as a message is embedded in the state equation of the transmitter and securely sent to the receiver through the synchronization to perform the decryption in the receiver part. Due to the properties of the pseudo-randomness sensitivity to initial condition and unpredictability in the chaotic systems, the image encryption algorithm can be strengthened. Furthermore, the S-box is established to have secure encryption. The S-box is a core component to provide higher security properties and is widely used in image encryption. There have been many approaches proposed for image encryption based on chaotic systems [20–23]. In [20], an image encryption scheme based on the pseudo-orbits of 1D chaotic maps is proposed for image encryption. However, the synchronization problem was not considered. In [21], Moon et al. introduced the self-synchronization approach for generalized Lorenz chaotic systems and applied it to image encryption. But the proposed synchronization approach was difficult to extend to general types of chaotic systems. In [22,23], the authors integrated the randomness of the chaotic signal to construct S-boxes and apply some methods to increase the execution efficiency. However, the mentioned researches above are all used the continuous controller to achieve chaos synchronization. Today, with the advancement and popularization of digital signal processing (DSP) technology, to simplify control circuit realization and reduce design costs, there is a current trend to use digital microcontroller to implement control solutions. However, to utilize the DSP microcontroller to implement the control schemes, the traditional continuous-time control design methods mentioned above for chaos synchronization cannot been applicable. Therefore, we aim to propose a novel discrete digital-redesign SMC which can be easily realized by using the microcontrollers to guarantee the chaos synchronization and then applied to image encryption. To complete the design, the analogy chaotic systems are established and the secret message is embedded in the master chaotic system such that it does not appear on the public channel. Then, the digital redesign sliding mode controller is proposed for synchronizing chaotic behavior. After the chaotic systems are synchronized, the embedded message can be reconstructed at the receiver and adopted for the image encryption.

The structure of this paper is given as follows. In Section 2, the structure of the secure communication based on synchronization is introduced. In Section 3, the image encryption algorithm is proposed and the performance index to show the strength of the algorithm is discussed. In Section 4, the results are concluded.

In this paper, x^T denotes the transport for a matrix x. $\|x\|$ represents the Euclidean norm of a vector x. $I_n \in \Re^{n\times n}$ denotes the identity matrix. $|\ x\ |$ is the absolute value of a constant x. $x^\dagger = (x^T x)^{-1} x^T$ denotes to the pseudo inverse matrix for a matrix $x \in \Re^{n\times m}$ and $x^\dagger x = I_m$.sgn(x) is the nonlinear sign function of x and if $x > 0$, sgn$(x) = 1$; if $x < 0$, sgn$(x) = -1$; and sgn$(x) = 0$ if $x = 0$. sgn$(x) = [\text{sgn}(x_1)\ ,\ \text{sgn}(x_2)\ ,\ \cdots\ ,\ \text{sgn}(x_m)]^T \in \Re^m$.

2. Communication Structure Based on Synchronization

A. The secure communication based on chaos synchronization.

In this section, a secure communication based on synchronization is formulated. Due to the characteristic of chaos synchronization, the message $m(t)$ can be embedded in the master system and obtained in the slave system through the SMC synchronization control method.

First, the structure of the master system can be described as

$$\dot{y}_m(t) = Ay_m(t) + B\big(g_y(y(t)) + m(t)\big) \tag{1}$$

and the slave system is described as

$$\dot{x}_s(t) = Ax_s(t) + B(g_x(x(t)) + u(t)), \tag{2}$$

where $A \in R^{n\times n}$ and $B \in R^{n\times m}$ are the system matrices, $g_x(x(t))$ and $g_y(y(t))$ are the nonlinear terms of chaotic systems, and $m(t)$ is the embedded message.

The error vector is defined as $e(t) = y_m(t) - x_s(t)$ and the error dynamics can be obtained as follows:

$$\begin{aligned} \dot{e}(t) &= \dot{y}_m(t) - \dot{x}_s(t) \\ &= Ae(t) + B(m(t) - u(t) + g_e(t)), \end{aligned} \tag{3}$$

where $g_e(t) = g_y(y(t)) - g_x(x(t))$. When the synchronization is ensured, it means the error state converges to zero, i.e., $e(t) = y_m(t) - x_s(t) = 0$, and $g_e(t)$ will converge to zero as well. Then, the equivalent error dynamics can be rewritten as

$$\dot{e}(t) = B(m(t) - u(t)) = 0. \tag{4}$$

B is the system matrix and not be equal to zero, therefore, one can derive that $m(t) = u(t)$. As a result, the message can be recovered in the slave system if the synchronization is ensured.

B. SMC design for message communication.

First, to achieve the sliding mode control for completing chaos synchronization, the sliding mode function can be selected as

$$s(t) = C_s e(t) + \int_0^t (-C_s Ae(\tau) + K_c e(\tau))d\tau, \tag{5}$$

where $C_s = B^{\dagger}$, and K_c is the designed controller gain. By differentiating (5), we have

$$\begin{aligned} &\dot{s}(t) = C_s\dot{e}(t) - C_s Ae(t) + K_c e(t) \\ &= C_s(Ae(t) + B(m(t) - u(t) + g_e(t))) - C_s Ae(t) + K_c e(t) \\ &= -u(t) + m(t) + K_c e(t) + g_e(t). \end{aligned} \tag{6}$$

When the controlled state trajectories enter the sliding manifold, the equivalent controller with the fact of $s(t) = \dot{s}(t) = 0$ can be derived as

$$u_{ceq}(t) = K_c e(t) + m(t) + g_e(t) \tag{7}$$

Thus, one can know that the equivalent controller $u_{eq}(t)$ is equal to $m(t)$ when the error converges to zero which is mentioned in (4).

To guarantee the minimization of the state error $e(t)$, we use the linear-quadratic method [24] to calculate the controller gain K_c which will be used in the sliding mode function (5). We consider the cost function as follows

$$J = \frac{1}{2}\int_0^{t_{end}} \Big\{e(\tau)^T Qe(\tau) + u^T(\tau)Ru(\tau)\Big\}d\tau, \tag{8}$$

where $Q = 10^q \times I_n$, R is a positive define matrix. According to the above performance, we have the Riccati equation as

$$A^T P + PA - PBR^{-1}B^T P + Q = 0_n, \tag{9}$$

where P is a positive symmetric define matrix. Thus, one can obtain the gain K_c:

$$K_c = R^{-1}B^T P. \tag{10}$$

To ensure the controlled dynamics, (3) can enter the sliding manifold, the controller is designed as follows:

$$u(t) = -K_c e(t) - \gamma_1 s(t) - \gamma_2 \text{sgn}(s(t)) - r_3 \text{sgn}(s(t)). \tag{11}$$

Replacing (11) into (6), (6) becomes

$$\dot{s}(t) = g_e(t) + m(t) - \gamma_1 s(t) - \gamma_2 \text{sgn}(s(t)) - \gamma_3 \text{sgn}(s(t)), \tag{12}$$

where γ_1 and γ_2 are positive parameters and γ_3 is chosen as $\|g_e(t)\| + \gamma$, where $\|m(t)\| < \gamma$. To verify the designed controller can ensure the occurrence of the sliding mode, a Lyapunov function is chosen as follows:

$$V(s(t)) = \frac{1}{2} s^T(t) s(t), \tag{13}$$

and then one differentiates (13) and obtains

$$\begin{array}{c} \dot{V}(s(t)) = s^T(t)\dot{s}(t) \\ = s^T(g_e(t) + m(t) - \gamma_1 s(t) - (\gamma_2 + \gamma_3)\text{sgn}(s(t))) \\ \leq \|g_e(t)\|\|s(t)\| + \|m(t)\|\|s(t)\| - \gamma_1\|s(t)\|^2 - \gamma_2\|s(t)\| - \gamma_3\|s(t)\| \\ \leq -\gamma_1\|s(t)\|^2 - \gamma_2\|s(t)\| \leq 0. \end{array} \tag{14}$$

From (14), one can know that $\dot{V}(s(t)) \leq 0$ when γ_3 is chosen as $\|g_e(t)\| + \gamma$. Thus, the design of SMC (12) is complete through the above derivation.

There is a chattering phenomenon when the design of SMC includes the nonlinear *sign* function. To overcome this situation, we replace *sign* function with saturation function $sat(s(t))$ [13]. The saturation function is shown as follows:

$$sat(s(t)) = \left[\frac{s_1(t)}{|s_1(t)| + \varepsilon} \ \cdots \ \frac{s_m(t)}{|s_m(t)| + \varepsilon} \right]^T, \tag{15}$$

where the parameter ε is an arbitrarily small but positive constant. Therefore, one can obtain the continuous-time SMC-based control law for secure communication as follows:

$$u_c(t) = -K_c e(t) - \gamma_1 s(t) - \gamma_2 sat(s(t)) - \gamma_3 sat(s(t)). \tag{16}$$

While the sliding mode is reaching, the synchronized error approaches to zero, the desired message $m(t)$ can be established by the control law (16) in the continuous time.

C. Digital redesign of H_2 SMC for message communication

After obtaining the continuous-time SMC-based controller, we utilize the digital-redesign method to make the transformation and obtain the corresponding discrete-time SMC controller. First, one can discretize (9) with Euler's method [25] and get

$$\begin{array}{c} s(kT_s + T_s) = s(kT_s) + T_s(g(x(kT_s)) - g(y(kT_s)) + m(kT_s) - \gamma_1 s(kT_s) - \gamma_2 \text{sgn}(s(kT_s)) - \gamma_3 \text{sgn}(s(kT_s))) \\ \Delta s(kT_s) = -\gamma_1 T_s s(kT_s) + T_s(g(x(kT_s)) - g(y(kT_s)) + m(kT_s) - \gamma_2 \text{sgn}(s(kT_s)) - \gamma_3 \text{sgn}(s(kT_s))) \end{array} \tag{17}$$

where T_s is the sampling time. To guarantee the occurrence of the sliding manifold in the discrete-time domain, Lemma 1 is given as follows.

Lemma 1 The following reaching condition is considered. [26].

$$\begin{aligned} &\Delta s(kT_s) = s(kT_s + T_s) - s(kT_s) \\ &\leq -\gamma_1 T_s s(kT_s) - \gamma_2 T_s \text{sgn}(s(kT_s)) < 0, \text{ for } s(kT_s) > 0, \\ &\Delta s(kT_s) = s(kT_s + T_s) - s(kT_s) \\ &\geq -\gamma_1 T_s s(kT_s) - \gamma_2 T_s \text{sgn}(s(kT_s)) > 0, \text{ for } s(kT_s) < 0, \end{aligned}$$

where $(1 - \gamma_1 T_s) > 0$, $k = 0, 1, \cdots, \infty$. If the reaching conditions above are satisfied, then the controlled state trajectories can converge to $s(kT_s) = 0$ and enter the sliding manifold.

Proof of Lemma1: when $s(kT_s) > 0$, $\Delta s(kT_s) = s(kT_s + T_s) - s(kT_s) < 0$ means $s(kT_s + T_s) < s(kT_s)$ and the trajectory of $s(kT_s)$ converges towards the sliding surface $s = 0$. When $s(kT_s) < 0$, $\Delta s(kT_s) = s(kT_s + T_s) - s(kT_s) > 0$ means $s(kT_s + T_s) > s(kT_s)$ and the trajectory of $s(kT_s)$ also converges towards the sliding surface $s = 0$. Therefore, if the reaching conditions above are satisfied, the controlled state trajectories can converge to $s(kT_s) = 0$ and enter the sliding manifold.

Based on Lemma 1, we calculate

$$\begin{aligned} \Delta s(kT_s) = \quad & -\gamma_1 T_s s(kT_s) + T_s(g_e(kT) + m(kT_s) - (\gamma_2 + \gamma_3)\text{sgn}(s(kT_s))) \\ & \leq -\gamma_1 T_s s(kT_s) - \gamma_2 T_s \text{sgn}(s(kT_s)), \text{ for } s(kT_s) > 0, \end{aligned} \tag{18}$$

$$\begin{aligned} \Delta s(kT_s) = \quad & -\gamma_1 T_s s(kT_s) + T_s(g_e(kT) + m(kT_s) - (\gamma_2 + \gamma_3)\text{sgn}(s(kT_s))) \\ & \leq -\gamma_1 T_s s(kT_s) - \gamma_2 T_s \text{sgn}(s(kT_s)), \text{ for } s(kT_s) < 0, \end{aligned} \tag{19}$$

Therefore, according to Lemma 1, $s(kT_s)$ will always converge to zero and the controlled system enters the sliding manifold. In this paper, Euler's method is adopted to discretize the proposed sliding mode controller such that the existence of the sliding mode can be guaranteed and the equivalent control can be achieved.

When the system is in the sliding manifold, i.e., $u_c(t) = u_{ceq}(t)$ as given in (7), the error dynamics can be discretized as

$$e(kT_s + T_s) = Ge(kT_s) + Hu_d^*(kT_s), \tag{20}$$

where $G = e^{AT_s}$, and $H = (G - I_n)A^{-1}B$. Due to zero-order-hold (Z.O.H.), the controller $u_d^*(kT_s)$ can be approached to $u_c^*(t) = u_d^*(kT_s) \cong u_c^*(kT_s + T_s)$, for $kT_s \leq t < kT_s + T_s$, and $u_d^*(kT_s)$ can be approximated as

$$u_d^*(kT_s) \cong u_c^*(kT_s + T_s) = -K_c e(kT_s + T_s). \tag{21}$$

According to (20), (21) can be rearranged as

$$u_d^*(kT_s) = -(I + K_c H)^{-1} K_c G e(kT_s) = -K_d e(kT_s), \tag{22}$$

where $K_d = -(I + K_c H)^{-1} K_c G$. Since the reaching condition is satisfied, the controlled system is operated in the sliding manifold. The digital-redesign method is utilized to ensure that $(G - HK_d)$ is Hurwitz and the stability of the controlled system can be guaranteed. Furthermore, to achieve the digital-redesign-based SMC, one can discretize (8) to implement the discrete-time SMC as follows

$$s(kT_s) = C_s e(kT) + s_I(kT_s), \tag{23}$$

$$s_I(kT_s + T_s) = s_I(kT_s) + T_s(-C_s A e(kT_s) + K_c e(kT_s)). \tag{24}$$

Finally, with the utilization of the proposed digital-redesign method and the adoption of the saturation function (15), the discrete SMC synchronization scheme can be obtained as

$$u_d(kT_s) = -K_d e(kT_s) - \gamma_1 s(kT_s) - \gamma_2 sat(s(kT_s)) - \gamma_3 sat(s(kT_s)), \tag{25}$$

While the sliding mode is reaching, the synchronized error approaches to zero, the desired message $m(t)$ can be established by the control law (16) in the discrete time.

D. Simulation for message communication

After completing the design of the digital redesign SMC synchronization controller, the simulation for verifying the feasibility is performed with the Lorenz chaotic system which can be found in reference [27]. The Lorenz chaotic system is given as follows:

$$\begin{cases} \dot{x}(t) = \delta_1(y(t) - x(t)) \\ \dot{y}(t) = x(t)(\delta_2 - z(t)) - y(t) \\ \dot{z}(t) = x(t)y(t) - \delta_3 z(t) \end{cases},$$

where, $\delta_1 = 10$, $\delta_2 = 28$ and $\delta_3 = 8/3$.

First, one rearranges the Lorenz chaotic system as the structure of the master-slave system considered in (1) and (2) for secure communication as follows

$$\dot{y}_m(t) = Ay_m(t) + B\big(g_y(y_m(t)) + m(t)\big),$$

and

$$\dot{x}_s(t) = Ax_s(t) + B(g_x(x_s(t)) + u(t)),$$

where the system matrices are $A = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & 0 \\ 0 & 0 & -8/3 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, $g_x(x_s(t)) = \begin{bmatrix} -x_{s1}(t) \times x_{s3}(t) \\ x_{s1}(t) \times x_{s2}(t) \end{bmatrix}$, and $g_y(y_m(t)) = \begin{bmatrix} -y_{m1}(t) \times y_{m3}(t) \\ y_{m1}(t) \times y_{m2}(t) \end{bmatrix}$. Then, the message $m(t) = \begin{bmatrix} m_1(t) & m_2(t) \end{bmatrix}^T = \begin{bmatrix} 0.3\cos(2\pi ft) & 0.3\cos(2\pi f(t+1)) \end{bmatrix}^T$ is given with $f = 1/3$, and embedded in the master system. The parameter in the controller is chosen with $\gamma_1 = 40$ and $\gamma_2 = 0.3$, and the parameter in saturation function is selected as $\varepsilon = 0.01$. The weighting matrices of the cost function are chosen as $Q = 10^4 \times I_3$ and $R = I_2$. The controller gain is calculated as

$$K_c = \begin{bmatrix} 65.612 & 105.3637 & -3.379 \times 10^{-15} \\ 2.585 \times 10^{-14} & -3.379 \times 10^{-15} & 97.3689 \end{bmatrix}.$$

After using the proposed digital redesign approach, one can obtain G, H, K_d as below:

$$G = \begin{bmatrix} 0.9902 & 0.0099 & 0 \\ 0.0278 & 0.9991 & 0 \\ 0 & 0 & 0.9973 \end{bmatrix}, H = \begin{bmatrix} 4.9818 \times 10^{-6} & 0 \\ 9.995 \times 10^{-4} & 0 \\ 0 & 9.9867 \times 10^{-4} \end{bmatrix},$$

and

$$K_d = \begin{bmatrix} 61.4144 & 95.8046 & -2.7782 \times 10^{-15} \\ 2.3432 \times 10^{-14} & -2.559 \times 10^{-15} & 88.5036 \end{bmatrix}.$$

We select the initial conditions as $x(0) = \begin{bmatrix} -1 & 2 & 5 \end{bmatrix}^T$ and $y(0) = \begin{bmatrix} 2 & 5 & 3 \end{bmatrix}^T$ for simulation. Simulation results are shown as follows.

In Figure 1, one can observe that the synchronization errors are soon approaching zero, which means the designed controller can make the system achieve synchronization effectively. The errors between controller $u(t)$ and message $m(t)$ converge to zero as shown in Figures 2 and 3, which is in accordance with the derivation in (4). Figure 4 illustrates the sliding mode trajectory also converges to zero as expected. Thus, one can figure out that the digital-redesign SMC-based controller is an effective method to make the system achieve synchronization and establish a secure communication. In Figure 5, one can observe that the synchronization errors are not stable before 10 seconds, however, the synchronization errors are closing to zero while the control input is active after 10 seconds. Therefore, the proposed SMC controller is effectiveness and robust.

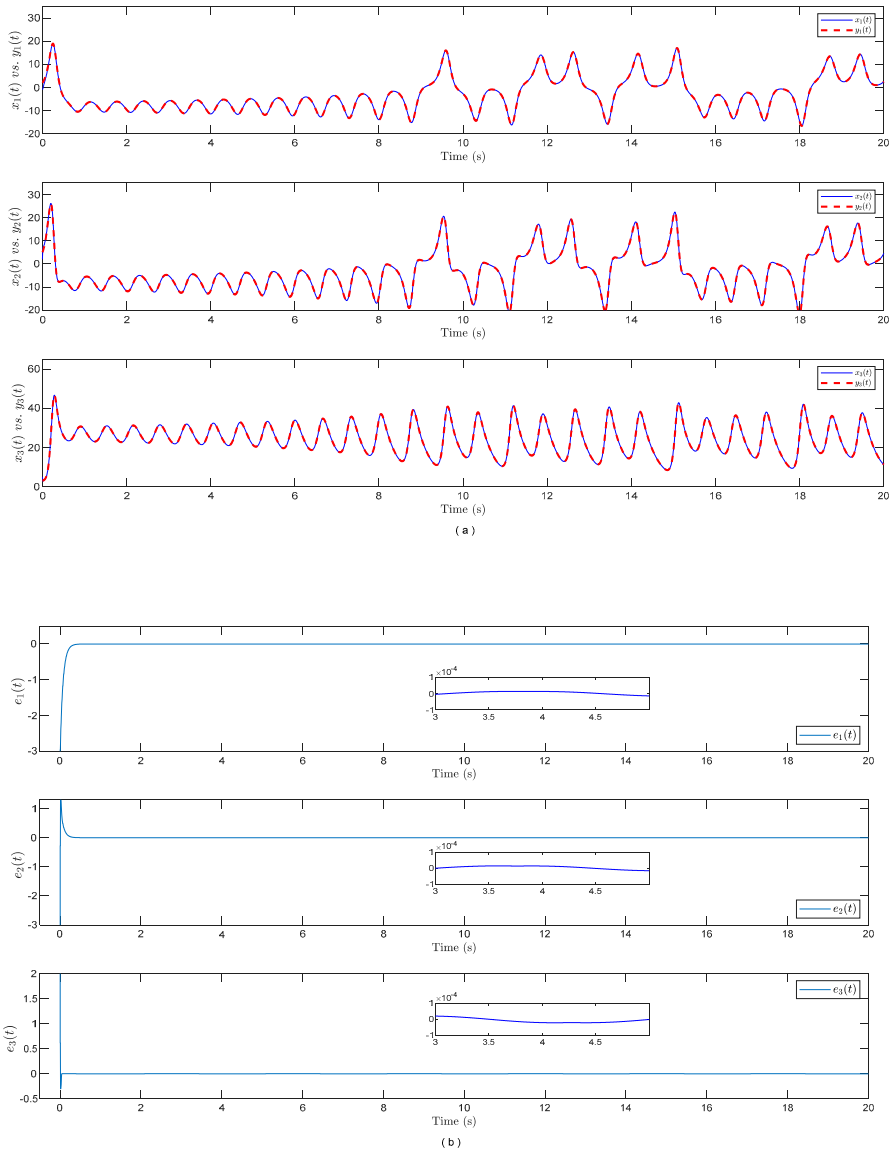


Figure 1. State responses: (**a**) the trajectories of the controlled master-slave system, (**b**) Synchronization errors between the master system and slave systems.

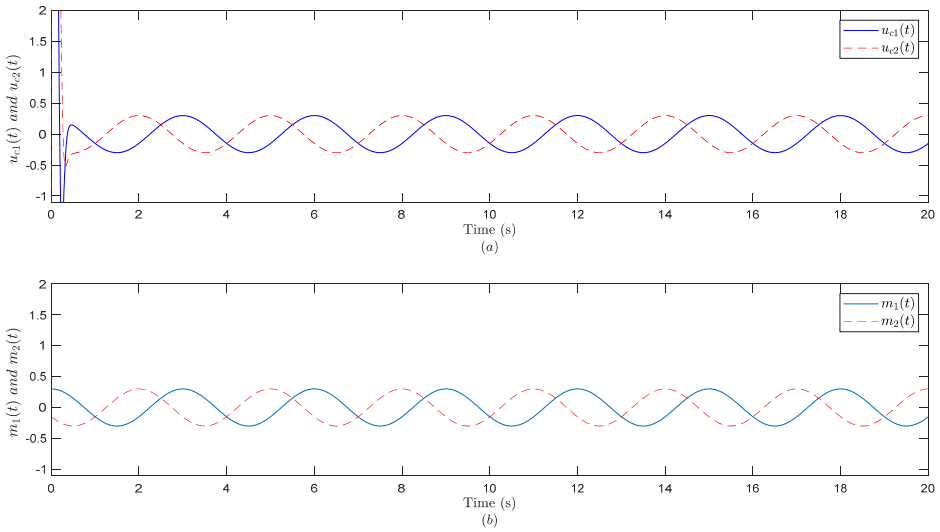


Figure 2. Message $m(t)$ and digital-redesign SMC-based controller $u_c(t)$: (**a**) Digital-redesign SMC-based controller; (**b**) Message $m(t)$.

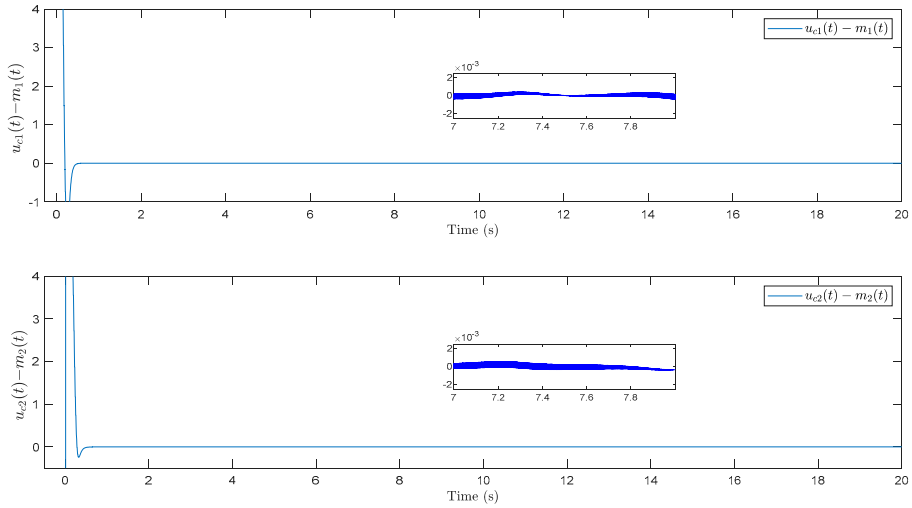


Figure 3. Errors between the message $m(t)$ and the controller $u_c(t)$.

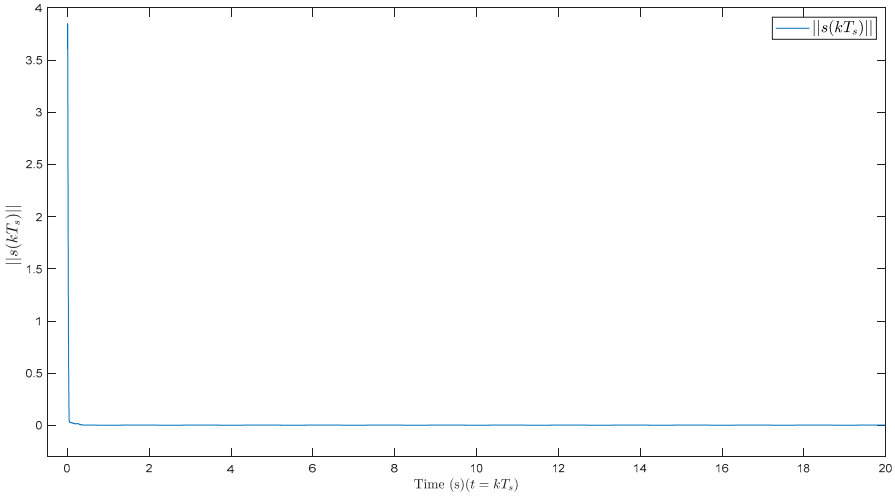


Figure 4. The response of the sliding mode function.

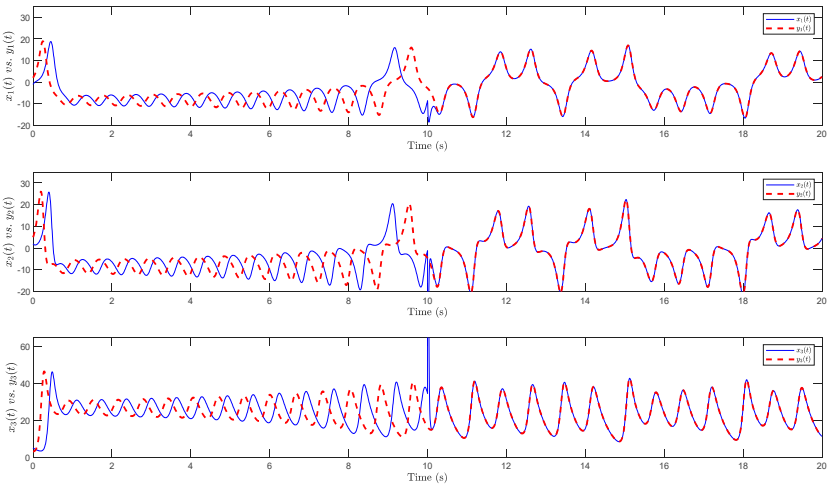


Figure 5. The state response with the control input enabled after 10 s.

3. Results

Image encryption based on secure communication

After completing the construction of secure communication with the hybrid synchronization control, the proposed controller is applied in image encryption in this section. First, the secret key for the image encryption algorithm is designed as the embedded message *m(t)* and the S-box is established by the secret key. The main function of the S-box is to increase encryption strength. With the secret key and S-box, the image encryption algorithm is proposed. The structure of the synchronization-based image encryption algorithm and flowchart are shown in Figure 6.

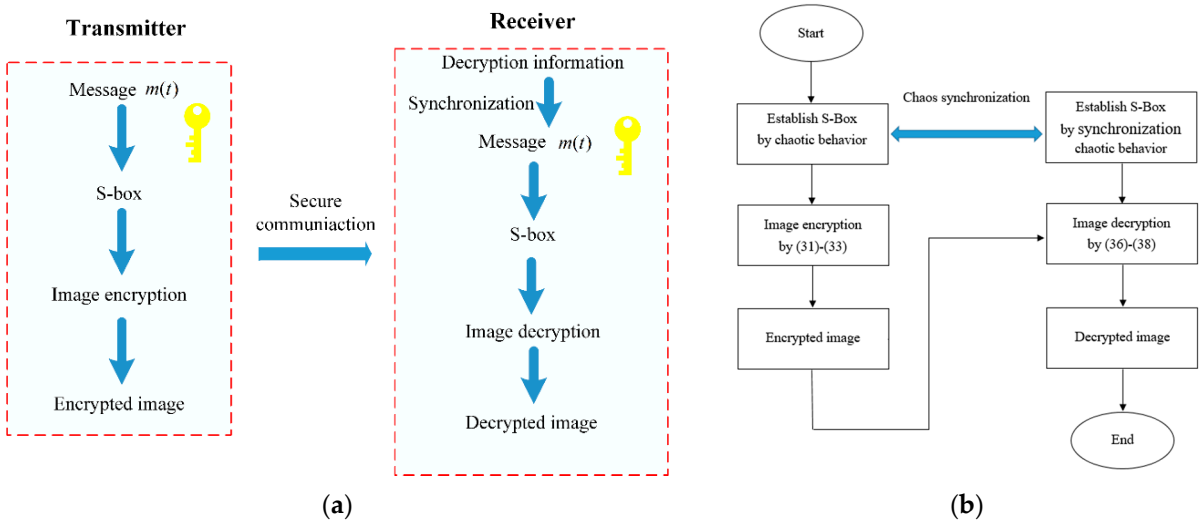


Figure 6. (**a**) The process of image encryption with secure communication; (**b**) The flowchart of the proposed algorithm.

A. Preliminary: establish S-Box by utilizing chaotic behavior

In this section, the secret key and the generation steps of the S-box are introduced. The reconstructed message $\hat{m}(t)$ is used for establishing the S-box and it is designed as

$$\begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} = \begin{cases} round(\hat{m}_{steady}(t)), & 0 < t < t_{end}/3 \\ round(\hat{m}_{steady}(t)), & t_{end}/3 < t < (2 \times t_{end})/3, \\ round(\hat{m}_{steady}(t)), & (2 \times t_{end})/3 < t < t_{end} \end{cases} \quad (26)$$

where m_1, m_2, and m_3 are appropriate positive integers, t_{end} is an entire time of synchronization simulation, and $m_{steady}(t)$ is the steady-state trajectory of $m(t)$. Then, one uses three constants in the message to establish the sample period m_{ts} for dealing with the state of the Lorenz chaotic system.

$$\begin{cases} m_{ts1} = round(1/(m_1 * t_{s1})), \\ m_{ts2} = round(1/(m_2 * t_{s2})), \\ m_{ts3} = round(1/(m_3 * t_{s3})), \end{cases} \tag{27}$$

where t_{s1}, t_{s2}, and t_{s3} are sampling times and $round(\bullet)$ is a function that rounds the element $\bullet$ to the nearest integer. The constants m_1, m_2, and m_3 are the important elements in image encryption and decryption.

After getting the m_{ts1}, m_{ts2}, and m_{ts3}, one can sample the states of the Lorenz chaotic system with m_{ts1}, m_{ts2}, and m_{ts3} respectively, and obtain sampled states $\begin{bmatrix} x_{d1} & x_{d2} & x_{d3} \end{bmatrix}^T$. Furthermore, the simple rules are applied in sampled states to get three sequences, x_{ts1}, x_{ts2}, and x_{ts3}. The rules are shown as follows

$$\begin{aligned} x_{ts1}(i) &= \begin{cases} 1, \text{ if } x_{d1}(i) > mean(x_{d1}), \\ 0, \text{ if } x_{d1}(i) < mean(x_{d1}), \end{cases} \\ x_{ts2}(i) &= \begin{cases} 1, \text{ if } x_{d2}(i) > mean(x_{d2}), \\ 0, \text{ if } x_{d2}(i) < mean(x_{d2}), \end{cases} \\ x_{ts3}(i) &= \begin{cases} 1, \text{ if } x_{d3}(i) > mean(x_{d3}), \\ 0, \text{ if } x_{d3}(i) < mean(x_{d3}), \end{cases} \end{aligned} \tag{28}$$

where $i = 0, 1, \ldots, s_2 - 1$, the size of the image is $s_1 \times s_2$, s_1 for row and s_2 for column, and $mean(x_d)$ is the function to get the mean value of the sequence x_d and it is a rule. The function of the sampled states is to permute the sequence and obtain the S-box.

After dealing with the states of the system, one uses it to obtain the S-box_x, S-box_y, and S-box_z. Take the S-box_x, for example. First, one generates a sequence $z_0 = [0, 1, 2, \ldots, s_2 - 1]$, and compare the sequence z_j with the sequence x_{ts1}, where j is the execution times. If the i term of x_{ts1} (i.e., $x_{ts1}(i)$) is one, the i term of z_0 (i.e., $z_0(i)$) is arranged to the far left side of the sequence z_0. If the i term of x_{ts1} (i.e., $x_{ts1}(i)$) is zero, the i term of z_0 (i.e., $z_0(i)$) is arranged to the far right side of the sequence z_0, and replaces the original z_0 after permutation. After completing the permutation from $i = 0$ to $s_2 - 1$, one gets the new sequence z_1. Then, one performs the same rules to get z_2. After running $N \times s_1$ times, one obtains an $S - box\_1$ with size $(N \times s_1) \times s_2$ and chooses the last s_1 rows of S-box_1 to build the S-box_x whose dimension is $s_1 \times s_2$. Rules for establishing the S-box is given in Figure 7.

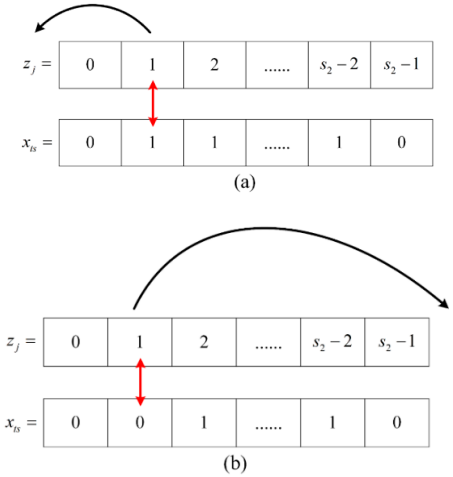


Figure 7. Rules for establishing the S-box: (**a**) when $x_{ts}(i) = 1$, permute $z_j(i)$ to the left-most term; and (**b**) when $x_{ts}(i) = 0$, permute $z_j(i)$ to the right-most term.

With the rules mentioned above, one can have another two S-box, S-box_2, and S-box_3, which are generated from the x_{ts2} and x_{ts3}, respectively. Then, one selects the last s_1 rows in S-box_2 and S-box_3 to obtain the S-box_y and S-box_z.

B. Image encryption algorithm

After constructing the S-boxes, the image encryption algorithm is proposed with the obtained S-boxes.

Step 1. Separate the image I with the size of $s_1 \times s_2$ into three grayscale images of red, green, and blue, respectively. Arrange the pixel from row to column, and one can obtain three sequences

$$\begin{cases} R = \{r_1, r_2, r_3, \ldots, r_{s_1 \times s_2}\}, \\ G = \{g_1, g_2, g_3, \ldots, g_{s_1 \times s_2}\} \\ B = \{b_1, b_2, b_3, \ldots, b_{s_1 \times s_2}\}, \end{cases}, \tag{29}$$

where r_i, g_i, and b_i are ith pixel of the red layer, green layer, and blue layer in a color image, respectively.

Step 2. Generate three S-boxes, S-box_x, S-box_y, and S-box_z, with the rules mentioned above and arrange three S-boxes from row to column as

$$\begin{cases} \mathrm{S-box\_x} = \{Sx_1, Sx_2, \ldots, Sx_{s_1 \times s_2}\}, \\ \mathrm{S-box\_y} = \{Sy_1, Sy_2, \ldots, Sy_{s_1 \times s_2}\}, \\ \mathrm{S-box\_z} = \{Sz_1, Sz_2, \ldots, Sz_{s_1 \times s_2}\}. \end{cases} \tag{30}$$

Step 3. Generate three sequences, cr, cg, and cb. Then, one can get

$$\begin{cases} ur_i = (r_i + Sz_i + cr_{i-1}) \bmod 256, \\ cr_i = ur_i \oplus Sx_i, \end{cases} \tag{31}$$

$$\begin{cases} ug_i = (g_i + cr_i + cg_{i-1}) \bmod 256, \\ cg_i = ug_i \oplus Sy_i, \end{cases} \tag{32}$$

$$\begin{cases} ub_i = (b_i + cg_i + cb_{i-1}) \bmod 256, \\ cb_i = ub_i \oplus Sz_i, \end{cases} \tag{33}$$

where $\oplus$ denote the bitwise exclusive or operation and $i = 1, 2, 3, \ldots, s_1 \times s_2$.

With the above three steps, one can obtain three encrypted layers cr_i,cg_i, and cb_i, and combine three layers to obtain the encrypted color image.

C. Decryption algorithm

In this section, the decryption algorithm is established. The decryption algorithm is just the inverse process of the encryption algorithm. One can obtain some decryption information from the transmitter such as the entire simulation time of synchronization, sampling time t_s, fast sampling time t_f, the state of the master system with the message embedded, and the formula of the sampling time for sampling the state of the system. With the above information, the message is obtained through the approach of synchronization, establishes the same S-box, and performs the following steps to obtain the decryption image.

Step 1. Perform the synchronization to have the message and calculate the sample time for sampling the state with the formula given in part A of Section 3. Then, sample the state to obtain $x_{tsd} = \begin{bmatrix} x_{tsd1} & x_{tsd2} & x_{tsd3} \end{bmatrix}^T$.

Step 2. Separate the encryption image I_{en} into three grayscale images of red, green, and blue, arrange the pixels from row to column, and get three sequences R_{en}, G_{en} and B_{en} as

$$\begin{cases} R_{en} = \left\{r'_1, r'_2, \ldots, r'_{s_{1 \times s_2}}\right\}, \\ G_{en} = \left\{g'_1, g'_2, \ldots, g'_{s_{1 \times s_2}}\right\}, \\ B_{en} = \left\{b'_1, b'_2, \ldots, b'_{s_{1 \times s_2}}\right\}. \end{cases} \tag{34}$$

Step 3. Establish the S-boxes with the state generated by synchronization and rule in part A of Section 3 and arrange three sequences as

$$\begin{cases} \mathrm{S-box\_}x_d = \{Sxd_1, Sxd_2, \ldots, Sxd_{s_1 \times s_2}\}, \\ \mathrm{S-box\_}y_d = \{Syd_1, Sxyd_2, \ldots, Syd_{s_1 \times s_2}\}, \\ \mathrm{S-box\_}z_d = \{Szd_1, Szd_2, \ldots, Szd_{s_1 \times s_2}\}. \end{cases} \tag{35}$$

Step 4. Generate the sequences, *udr*, *udg*, and *udb*, and get the decryption component *cdr*, *cdg*, and *cdb* as follows:

$$\begin{cases} udr_i = r'_i \oplus Sxd_i, \\ cdr_i = \left(udr_i - Szd_i - r'_{i-1}\right) \mathrm{mod} 256, \end{cases} \tag{36}$$

$$\begin{cases} udg_i = g'_i \oplus Syd_i, \\ cdg_i = \left(udg_i - r'_i - g'_{i-1}\right) \mathrm{mod} 256, \end{cases} \tag{37}$$

$$\begin{cases} udb_i = b'_i \oplus Szd_i, \\ cdb_i = \left(udb_i - g'_i - b'_{i-1}\right) \mathrm{mod} 256. \end{cases} \tag{38}$$

Finally, one can combine three layers, *cdr*, *cdg*, and *cdb*, and obtain the decrypted color image.

D. Simulation results

The Lena image with the size 512×512 is used as the test image to perform the image encryption. The secret keys are taken as $m_1 = 1$, $m_2 = 2$, and $m_3 = 3$. The plain image of Lena is shown in Figure 8 and its histogram is shown in Figure 9. After using the proposed encryption algorithm, one can obtain the encrypted image shown in Figure 10, and its histogram is shown in Figure 11. The outline of the Lena is hard to distinguish in the encrypted image. The encrypted image uses all the grayscales from 0 to 255 and the histogram of the encryption image is flat which means the encryption scheme is effective and secure.

Figure 8. Lena image.

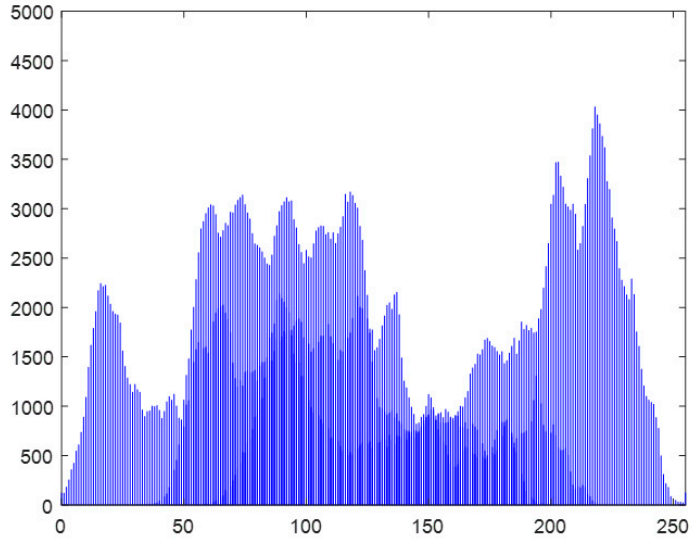


Figure 9. Histogram of Lena image.

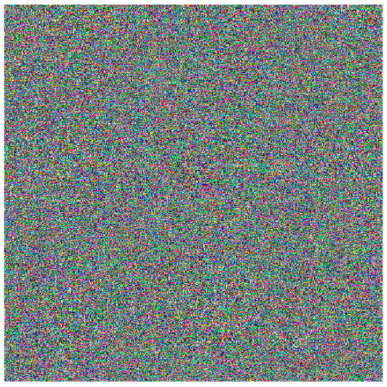


Figure 10. Encrypted image of Lena.

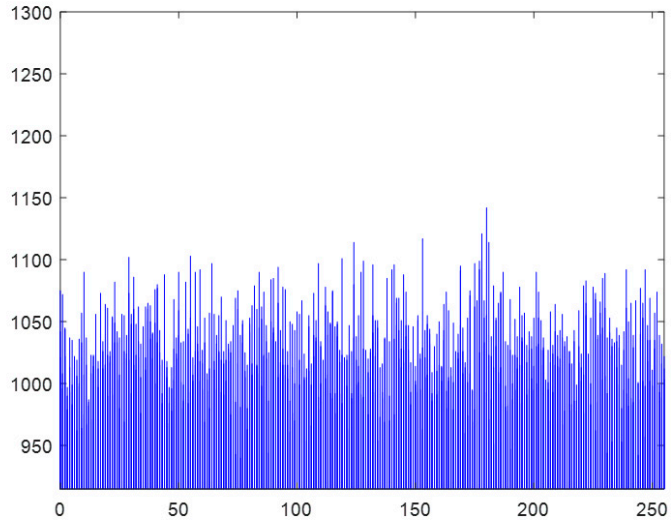


Figure 11. Histogram of encrypted Lena image.

Furthermore, another two images, baboon image with size 512 × 512, and the all-black figure with size 512 × 512, are used as the test images as well. The baboon image is shown in Figure 12 and its histogram is shown in Figure 13. The all-black figure is shown in Figure 14 and its histogram is shown in Figure 15. One can obtain the encrypted baboon

image and the encrypted all-black image shown in Figures 16 and 17, respectively, and the histograms are, respectively, shown in Figures 18 and 19. The encrypted image is done by utilizing the above algorithm with the same secret key. Then, we can observe that the histogram of the encrypted baboon image (Figure 18) and the histogram of the encrypted all-black image (Figure 19) are flat as well. Eventually, the image encryption algorithm can be applied in the image with any size, and the detailed analysis for the security is discussed in the next section.

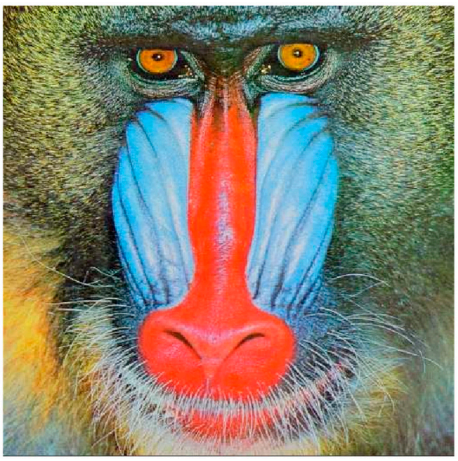


Figure 12. Baboon image.

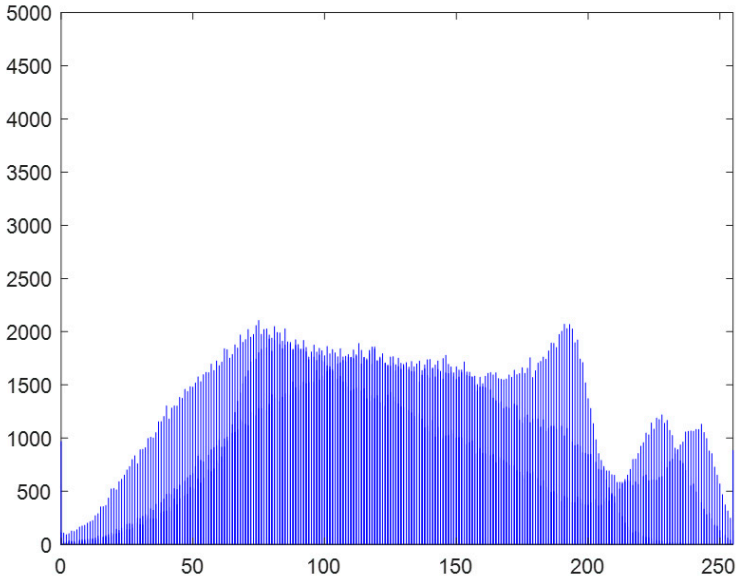


Figure 13. Histogram of baboon image.

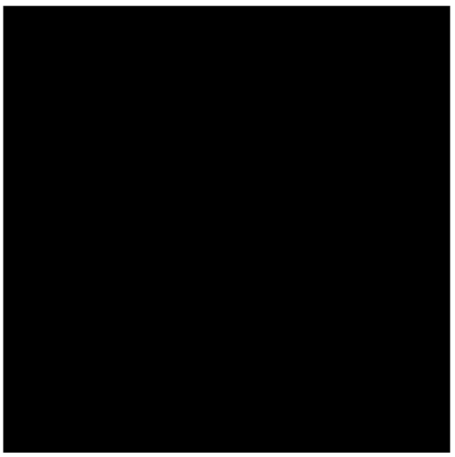


Figure 14. All-black image.

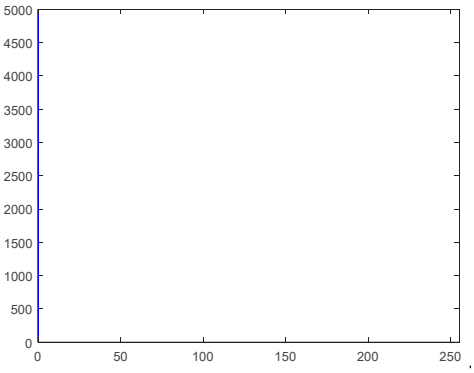


Figure 15. Histogram of all-black image.

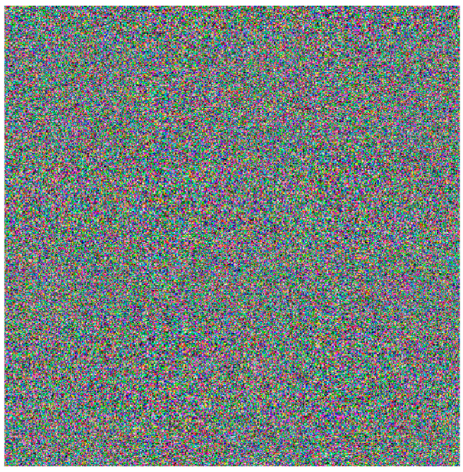


Figure 16. Encrypted baboon image.

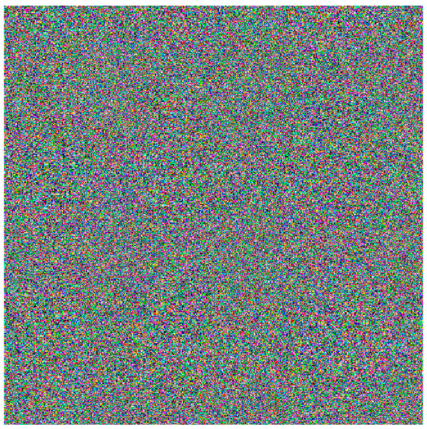


Figure 17. Encrypted image of all-black image.

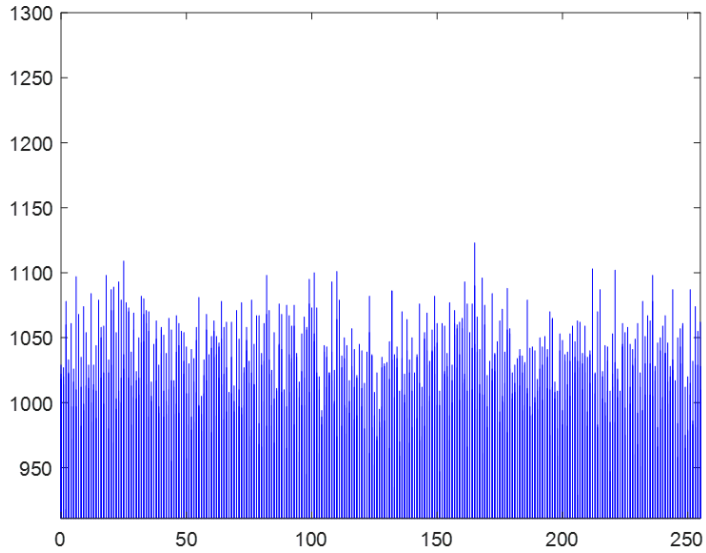


Figure 18. Histogram of encrypted baboon image.

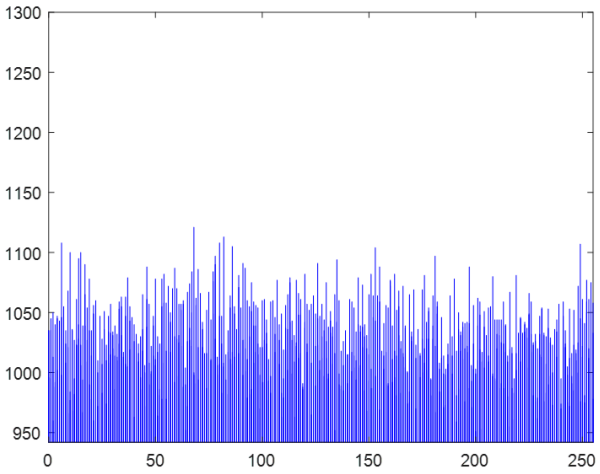
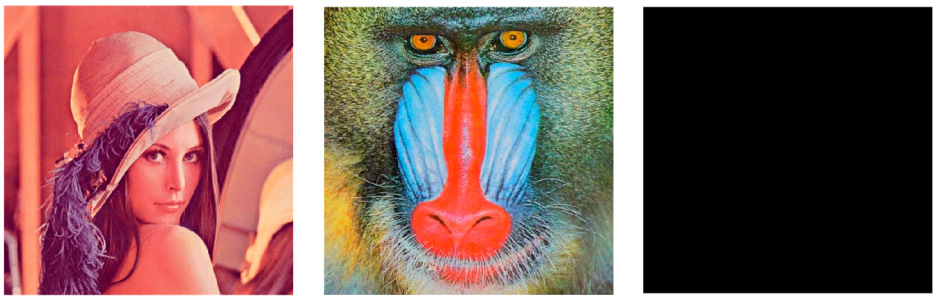


Figure 19. Histogram of Encrypted all-black image.

Once the encryption is completed, one starts the process of the decryption with decryption information. The file of decryption information includes the system states, the sampling time, the synchronization time, and the encrypted image. The file of decryption

information is established in the transmitter and sent to the receiver. Then, by using the proposed synchronization-based communication to obtain the secret key, one performs the decryption algorithm to obtain the decrypted image. Figure 20 shows the decrypted images.



(**a**) (**b**) (**c**)

Figure 20. Decrypted images: (**a**) Lena image, (**b**) baboon image, and (**c**) all-black image.

To show the effectiveness of the encrypted algorithm, the plain image is contrasted with the decrypted image pixel by pixel, and one can get the following result.

$$\text{image} - \text{image}_{decryption} = \sum_{i=1}^{s_1}\sum_{j=1}^{s_2}\Big(\text{image}(i,j) - \text{image}_{decryption}(i,j)\Big) = 0 \tag{39}$$

Thus, one can know that there is no difference between two images, and the encrypted and decrypted algorithms are feasible.

E. Security analysis

In this section, we analyze the security of the encryption algorithm with three images.

(1) Key space analysis

Key space means the total number of all different keys used in the encryption algorithm. The value of key space must be high so that it can make the brute-force attack ineffective. The secure key $m(t)$ in this proposed algorithm is utilized to generate the sampling time m_{ts} for sampling the states, and the sampling time m_{ts} is ranged in $10^0 \sim 10^3$. The key space can reach 10^9. The valid precision of the initial condition in chaotic systems is set to 10^{-14}, so the key space can reach $10^{14\times 3} = 10^{42}$. Thus, the number of different key combinations that can be used are $10^{42} \times 10^9 = 10^{51}$, which is large enough to resist the brute-force attack.

(2) Correlation

Adjacent pixels of an image will always be similar and with a strong correlation. A good image encryption algorithm is able to weaken the correlation of the adjacent pixels. The correlation function of the image is given as follows:

$$correlation = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{40}$$

where $\text{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$.

The correlation analysis demonstrates the similarity between two adjacent pixels in the vertical direction and the horizontal directions. The range of the correlation is $[\ -1 \quad 1\ ]$. The higher value means the higher correlation. One randomly selects 3000 pairs of the adjacent pixel from the plain image and encrypted image, respectively, to calculate the correlation coefficient. The horizontal correlation of the plain Lena image is shown in Figure 21 and the horizontal correlation of encrypted Lena image is shown in Figure 22. The vertical correlation of the plain Lena image is shown in Figure 23 and the vertical correlation of the encrypted Lena image is shown in Figure 24. One can find that the correlation of the adjacent pixel weakens after applying the encryption algorithm. The

value of the correlation is shown in Table 1. Then, one can compare the proposed algorithm with the algorithm in [22] on the encrypted Lena image. One can observe that the correlation coefficient in this paper is smaller than the correlation coefficient in [22] and this means the encryption algorithm in this paper is much more effective in decreasing the adjacent correlation as wells.

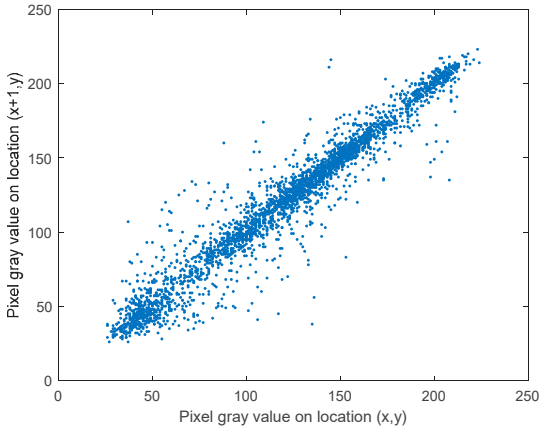


Figure 21. Correlation of the horizontal adjacent pixels of the Lena image.

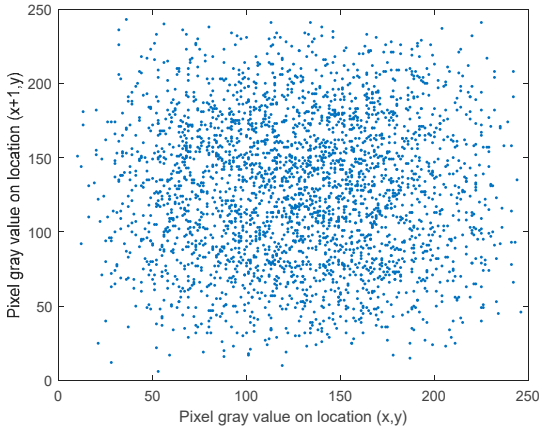


Figure 22. Correlation of the horizontal adjacent pixels of the encrypted Lena image.

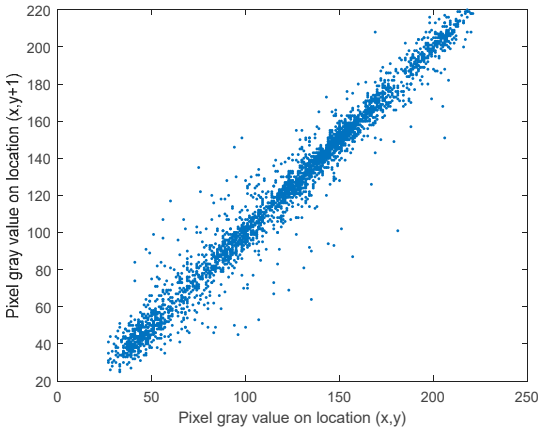


Figure 23. Correlation of the vertical adjacent pixels of the Lena image.

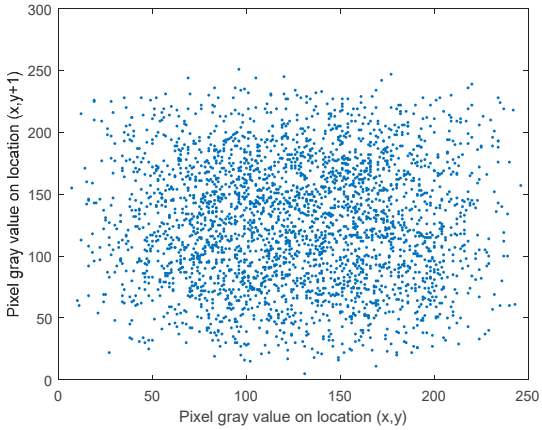


Figure 24. Correlation of the vertical adjacent pixels of the encrypted Lena image.

Table 1. The Correlation Coefficients in the Horizontal and Vertical Directions.

Correlation.	Horizontal	Vertical
Figure 8	0. 9681	0.9823
Figure 10	−0.0017	−0.0013
Figure 12	0.8761	0.7786
Figure 14	0.0018	−0.0031
Figure 16	NaN	NaN
Figure 18	−0.0042	−0.0048
Encrypted Lena image in Ref. [22]	0.003	

(3) Entropy

The entropy function of an image is given as follows:

$$Entropy = -\sum_{a,b} p(\rho(a,b)) \log_2 p(\rho(a,b)), \tag{41}$$

where a,b are the numbers of the rows and the columns of the image, $\rho(a,b)$ is the pixel value at the ath row and the bth column in the image, and $p(\rho(a,b))$ is the probability of image pixel at the ath row and the bth column. Entropy demonstrates the randomness of the image and the value range in $[\ 0 \quad 8\ ]$ for an image having 256 scales. The high value of entropy means the encrypted image has a greater amount of randomness. The value of entropy of the proposed algorithm is shown in Table 2. Comparing the encrypted Lena image in [22] with the encrypted Lena image in this paper, one can figure out that the entropy of the encrypted Lena image in this paper is larger than that in [22]. It means the Lena image encrypted by the proposed algorithm can get the higher randomness in the encrypted image.

Table 2. Entropies of the Encrypted Image.

Ecrypted Image.	R	G	B
Figure 10	7.9993	7.9992	7.9992
Figure 14	7.9992	7.9992	7.9993
Figure 18	7.9993	7.9993	7.9993
Encrypted Lena image in Ref. [22]	7.9808	7.9811	7.9814

F. Ability to resist differential attack

The differential attack is a good method to break the proposed encryption algorithm. If the encryption method has good sensitivity and diffusion property to the plain image,

it can resist those attacks. There are two indexes to evaluate the diffusion property, the number of pixel changing rate (NPCR) and unified averaged changed intensity (UACI). NPCR implies the change rate between two encrypted images C^1 and C^2 encrypted from two plain images with only one pixel different. NPCR and the UACI are defined as (42) and (43)

(1) NPCR :

$$N(C^1, C^2) = \sum_{i,j} \frac{D(i,j)}{N \times M} \times 100\%, \tag{42}$$

(2) UACI :

$$u(C^1, C^2) = \frac{1}{N \times M} \sum_{i,j} \frac{|C^1(i,j) - C^2(i,j)|}{T} \times 100\%, \tag{43}$$

where $D(i,j)$ is defined as $D(i,j) = \begin{cases} 0, \text{ if } C^1(i,j) = C^2(i,j) \\ 1, \text{ if } C^1(i,j) \neq C^2(i,j) \end{cases}$.

Here, we encrypt the Lena image, baboon image, and all-black image as C^1, and change the value of one pixel Lena image, baboon image, and all-black image then encrypt as C^2. Then, calculate the values of NPCR and UACI, and compare NPCR and UACI of the proposed method with the others [20,22] in Tables 3 and 4.

Table 3. NPCR Values of Encrypted Images with One Pixel Different in Plain Images.

NPCR	R	G	B	Average
Figure 8 Lena image	99.99%	99.61%	99.59%	99.73%
Figure 12 Baboon image.	99.99%	98.42%	99.99%	99.46%
Figure 16 All-black image.	99.99%	99.58%	99.59%	99.72%
Encrypted Lena image [22]	99.647%	99.623%	99.594%	99.63%

Table 4. UACI Values of Encrypted Images with One Pixel Different in Plain Images.

UACI	R	G	B	Average
Figure 8 Lena image	33.59%	33.56%	33.45%	33.53%
Figure 12 Baboon image.	33.42%	33.46%	33.41%	33.43%
Figure 16 All-black image.	33.37%	33.43%	33.48%	33.42%
Encrypted Lena image [22]	33.53%	33.27%	33.43%	33.41%

One can find that NPCR applied in color images is over 99% and NPCR applied in the all-black image can be also over 99%. UACI applied in the color images can be over 33% and UACI applied in all black image can approach 33%. The result shows that the proposed encryption algorithm is sensitive to a tiny change in image, even if the change is only in one pixel. The comparison results in Table 5 reveal that the method proposed in this paper has better results in NPCR and UACI tests. Thus, the proposed encryption algorithm is strong enough to make the differential attack ineffectively.

Table 5. Comparison of NPCR and UACI criteria of proposed method and the others.

Test Methods for Lena Image	NPCR	UACI
Proposed method	99.73%	33.53%
Erivelton et al. [20]	99.61%	33.46%
Liu et al. [22]	99.63%	33.41%

4. Conclusions

This paper proposes a digital-redesign SMC-based control law to achieve the chaotic synchronization. The proposed encryption/decryption algorithm integrates the synchronization technology of chaotic systems with the secret key transmission which embeds the secret keys in the chaos trajectory of the chaotic system. Therefore, the secret keys do not expose in the public channel and the security is improved. The designed controller makes the sliding trajectories converge to the sliding mode and complete synchronization. Furthermore, the proposed discrete digital-redesign SMC-based control law can make the controller easy to realize with high precision and low cost by using Raspberry Pi microcontrollers. After designing the controller, we construct the secure communication based on the chaotic synchronization. Furthermore, a chaos-based image encryption algorithm is established by the S-boxes to strengthen the complexity of the encryption algorithm. Security analysis has been included to verify the feasibility and effectiveness for the proposed encryption algorithm.

Author Contributions: All authors contributed to the paper. J.-S.F. wrote the manuscript with the supervision from J.S.-H.T. and J.-J.Y., L.-H.C. and S.-M.G. are responsible for the simulation of the sliding mode control and image encryption. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Ministry of Science and Technology, Taiwan, under grant MOST- 110-2221-E-167 -030 and MOST-110-2218-E-006 -014 -MBK.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhu, Z.Y.; Zhao, Z.S.; Zhang, J.; Wang, R.K.; Li, Z. Adaptive fuzzy control design for synchronization of chaotic time-delay system. *Inf. Sci.* **2020**, *535*, 225–241. [CrossRef]
2. Modiri, A.; Mobayen, S. Adaptive terminal sliding mode control scheme for synchronization of fractional-order uncertain chaotic system. *ISA Trans.* **2020**, *105*, 33–50. [CrossRef]
3. Jing, C.; Xu, H.; Niu, X. Adaptive sliding mode disturbance rejection control with prescribed performance for robotic manipulators. *ISA Trans.* **2019**, *91*, 41–51. [CrossRef]
4. Zheng, K.; Hu, Y.; Wu, B. Intelligent fuzzy sliding mode control for complex robot system with disturbances. *Eur. J. Control* **2020**, *51*, 95–109. [CrossRef]
5. Labbadi, M.; Cherkaoui, M. Robust adaptive nonsingular fast terminal sliding-mode tracking control for an uncertain quadrotor UAV subjected to disturbances. *ISA Trans.* **2020**, *99*, 290–304. [CrossRef]
6. Van, M. An enhanced tracking control of marine surface vessels based on adaptive integral sliding mode control and disturbance observer. *ISA Trans.* **2019**, *90*, 30–40. [CrossRef]
7. Han, S. Fractional-order command filtered backstepping sliding mode control with fractional-order nonlinear disturbance observer for nonlinear systems. *J. Frankl. Inst.* **2020**, *357*, 6760–6776. [CrossRef]
8. Ma, X.; Zhang, J.; Wang, J. Design of Disturbance Observer Based Sliding Mode Control for Fuzzy Systems. *IFAC Pap. OnLine* **2017**, *50–51*, 717–722.
9. Alipouri, Y.; Alipour, H.; Huang, B. Multiple step ahead prediction based high order discrete-time sliding mode control design with actuator and communication delays. *J. Frankl. Inst.* **2020**, *357*, 7845–7863. [CrossRef]
10. Wu, Y.; Huangfu, Y.; Ma, R.; Ravey, A.; Chrenko, D. A strong robust DC-DC converter of all-digital high-order sliding mode control for fuel cell power applications. *J. Power Sources* **2019**, *413*, 222–232. [CrossRef]
11. Abolvafaei, M.; Ganjefar, S. Maximum power extraction from a wind turbine using second-order fast terminal sliding mode control. *Renew. Energy* **2019**, *139*, 1437–1446. [CrossRef]
12. Merabet, A.; Labib, L.; Ghias, A.M.Y.M.; Aldurra, A.; Debbouza, M. Dual-mode operation based second-order sliding mode control for grid-connected solar photovoltaic energy system. *Electr. Power Energy Syst.* **2019**, *111*, 459–474. [CrossRef]
13. Sumantri, B.; Uchiyama, N.; Sano, S. Least square based sliding mode control for a quad-rotor helicopter and energy saving by chattering reduction. *Mech. Syst. Signal Processing* **2016**, *66–67*, 769–784. [CrossRef]
14. Morais, C.F.; Braga, M.F.; Tognetti, E.S.; Oliveira, R.C.L.F.; Peres, P.L.D. H_2 and H_∞ digital redesign of analog controllers for continuous-time polytopic systems. *IFAC Pap. OnLine* **2017**, *50–51*, 6691–6696. [CrossRef]

15. Tsai, J.S.H.; Cheng, H.; Moussighi, M.M.; Shieh, L.S. Digital redesign of observer-based weighting switch controller for cascaded analog systems with state saturation and external loads. *ISA Trans.* **2005**, *44*, 93–115. [CrossRef]
16. Feki, M. An adaptive chaos synchronization scheme applied to secure communication. *Chaos Solitons Fractls* **2003**, *18*, 141–148. [CrossRef]
17. Cheng, C.J. Robust synchronization of uncertain unified chaotic systems subject to noise and its application to secure communication. *Appl. Math. Comput.* **2012**, *219*, 2698–2712. [CrossRef]
18. Chan, J.C.L.; Lee, T.H.; Tan, C.P. Secure communication through a chaotic system and a sliding-mode observer. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, 1–13. [CrossRef]
19. Vaseghi, B.; Mobayen, S.; Hashemi, S.S.; Fekih, A. Fast reaching finite time synchronization approach for chaotic systems with application in medical image encryption. *IEEE Access* **2021**, *9*, 25911–25925. [CrossRef]
20. Erivelton, G.; Lucas, G.; Janier, A.G.; Denis, N.; Aleksandra, T. Image encryption based on the pseudo orbits from 1D chaotic map. *Chaos* **2019**, *29*, 061101.
21. Moon, S.; Baik, J.J.; Seo, J.M. Chaos synchronization in generalized Lorenz systems and an application to image encryption. *Commun. Nonlinear Sci. Numer. Simulat.* **2021**, *96*, 105708. [CrossRef]
22. Liu, H.; Kadir, A.; Gong, P. A fast color image encryption scheme using one-time S-Boxes based on complex chaotic system and random noise. *Opt. Commun.* **2015**, *338*, 34–347. [CrossRef]
23. Hussain, I.; Anees, A.; Alkhaldi, A.H.; Algarni, A.; Aslam, M. Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications. *Chin. J. Phys.* **2018**, *56*, 1609–1621. [CrossRef]
24. Malkapure, H.G.; Chidambaram, M. Comparison of Two Methods of Incorporating an Integral Action in Linear Quadratic Regulator. *IFAC Proc. Vol.* **2014**, *47*, 55–61. [CrossRef]
25. Li, S.; Du, H.; Yu, X. Discrete-Time Terminal Sliding Mode Control Systems Based on Euler's Discretization. *IEEE Trans. Autom. Control* **2013**, *59*, 546–552. [CrossRef]
26. Fang, J.S.; Tsai, S.H.; Yan, J.J.; Chen, P.L. Realization of DC-DC Buck Converter Based on Hybrid H2 Model Following Control. *IEEE Trans. Ind. Electron.* **2021**, *69*, 1782–1790. [CrossRef]
27. Zhuang, L.; Cao, L.; Wu, Y.; Zhong, Y.; Zhangzhong, L.; Zheng, W.; Wang, L. Parameter estimation of Lorenz chaotic system based on a hybrid Jaya-Powell algorithm. *IEEE Access* **2020**, *8*, 20514–20522. [CrossRef]