n-Dimensional Polynomial Chaotic System With Applications

Zhongyun Hua, *Member, IEEE*, Yinxing Zhang, Han Bao, *Member, IEEE*,
Hejiao Huang, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Designing high-dimensional chaotic maps with expected dynamic properties is an attractive but challenging task. The dynamic properties of a chaotic system can be reflected by the Lyapunov exponents (LEs). Using the inherent relationship between the parameters of a chaotic map and its LEs, this paper proposes an *n*-dimensional polynomial chaotic system (*n*D-PCS) that can generate *n*D chaotic maps with any desired LEs. The *n*D-PCS is constructed from *n* parametric polynomials with arbitrary orders, and its parameter matrix is configured using the preliminaries in linear algebra. Theoretical analysis proves that the *n*D-PCS can produce high-dimensional chaotic maps with any desired LEs. To show the effects of the *n*D-PCS, two high-dimensional chaotic maps with hyperchaotic behaviors were generated. A microcontroller-based hardware platform was developed to implement the two chaotic maps, and the test results demonstrated the randomness properties of their chaotic signals. Performance evaluations indicate that the high-dimensional chaotic maps generated from *n*D-PCS have the desired LEs and more complicated dynamic behaviors compared with other high-dimensional chaotic maps. In addition, to demonstrate the applications of *n*D-PCS, we developed a chaos-based secure communication scheme. Simulation results show that *n*D-PCS has a stronger ability to resist channel noise than other high-dimensional chaotic maps.**

***Index Terms*— Chaotic system, hardware implementation, random number generator, secure communication, nonlinear system.**

I. Introduction

Chaos theory is a type of nonlinear theory that describes the evolution of dynamical systems from ordered states to disordered states [1]. Chaos widely exists in many man-made and natural phenomena [2]–[4]. There are many methods for defining the existence of chaos. Among these, the Lyapunov exponent (LE) and Devaney's definition are two widely accepted measures [5], [6]. The LE method defines chaos by quantizing the divergence rate of the close trajectories of a dynamical system, whereas Devaney's definition defines chaos in terms of initial state sensitivity, topological transitivity, and dense periodic orbits [7]. Owing to these properties, chaos theory has attracted much attention in various research fields [8]–[11]. For example, the integer-order and fractional-order chaotic systems [12]–[14] are widely used in secure communication schemes because the unpredictability of chaotic outputs can embed secret data [15], [16].

With the rapid development of chaos theory, many studies have pointed out that existing chaotic systems manifest various weaknesses when applied to practical applications [17]. First, many chaotic systems have discontinuous chaotic ranges [18]. Small perturbations may cause the parameters to fall into the periodic windows if the chaotic ranges are discontinuous [19]. Second, chaos is defined in mathematical domain with infinite precision. When implemented in digital platforms with finite precision, a chaotic system will inevitably cause chaos degradation because of precision truncation [19]. However, if a chaotic system exhibits complex behaviors, chaos degradation may occur later. In particular, many existing chaotic systems have simple structures and low complexities, and thus chaos degradation may easily occur in digitized platforms [20]. In addition, researchers have found that the chaotic signals of many existing chaotic maps can be estimated using artificial intelligence technologies [21], [22]. These technologies include recognizing system prototypes [23], [24], prediction of chaotic signals [25], [26], and estimation of the control parameters and initial states [27], [28]. Most chaos-based applications are developed based on the assumption that chaotic signals are unpredictable when the exact initial states and parameters are unknown. If useful information about the chaotic signals can be predicted without knowing the initial states and parameters, the chaotic systems lose the property of being unpredictable, which may lead to the corresponding applications to become ineffective [29].

To solve the weaknesses of existing chaotic systems in practical applications, many studies have been devoted to enhancing the complexity of chaotic systems [30]. These efforts can be divided into two types: efforts for low-dimensional chaotic

Manuscript received May 19, 2021; revised August 24, 2021; accepted September 17, 2021. Date of publication October 18, 2021; date of current version January 28, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62071142, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515011406, in part by Shenzhen College Stability Support Plan under Grant GXWD20201230155427003-20200824210638001, and in part by the Research Committee at University of Macau under Grant MYRG2018-00136-FST. This article was recommended by Associate Editor A. Oliveri. *(Corresponding authors: Zhongyun Hua; Han Bao.)*

Zhongyun Hua, Yinxing Zhang, and Hejiao Huang are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China (e-mail: huazyum@gmail.com; yxzhang23@163.com).

Han Bao is with the School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213164, China (e-mail: charlesbao0319@gmail.com).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3117865.

Digital Object Identifier 10.1109/TCSI.2021.3117865

maps and high-dimensional chaotic maps. One strategy for low-dimensional chaotic maps is to disturb chaotic signals or control parameters [31]. This strategy can greatly increase the period length of digitalized chaotic maps [17]. The other strategy is to develop new low-dimensional chaotic maps with better complexity and performance [32]. These newly generated chaotic maps can overcome the weaknesses of existing low-dimensional chaotic maps, such as simple structures, discontinuous chaotic ranges, and a lack of hyperchaotic behaviors [33], [34]. However, these studies on low-dimensional chaotic maps have certain limitations. The strategy of disturbing existing low-dimensional chaotic maps cannot show stable performance [35], and the development of new chaotic maps rely heavily on the experience of researchers and, thus, this strategy lacks theoretical guarantees [33].

Compared with low-dimensional chaotic maps, high-dimensional chaotic maps have more complicated structures and affluent dynamic properties, and as a result, these have been widely researched in recent years. Because the complexity of a high-dimensional chaotic map can be reflected by its positive LEs [36], a key concern in developing high-dimensional chaotic maps is customizing the number of positive LEs and their values. Many high-dimensional chaotic maps with multiple positive LEs have been developed recently [37], [38]. Representative examples are provided below. Shen *et al*. developed a systematic method to construct nD continuous-time chaotic maps with $\lfloor n-1 \rfloor/2$ positive LEs [39]. Wu *et al*. generated new nD discrete-time Cat maps using Laplace expansion [40]. The generated Cat maps have multiple positive LEs, and thereby show better complexity compared with other Cat maps. These works can generate continuous-time and discrete-time hyperchaotic maps with multiple and even theoretically the maximum number of positive LEs [36], [41]. However, the effects of these methodologies rely heavily on the debugging parameters. In addition, these methodologies can only construct high-dimensional chaotic maps with the desired numbers of positive LEs, but the LE values cannot be determined exactly.

To generate high-dimensional chaotic maps with expected dynamic properties, this study proposes an nD polynomial chaotic system (nD-PCS). The nD-PCS is constructed from n parametric polynomials, and its parameter matrix is configured through dimension expansion. Theoretical analysis shows that the n LEs of the nD-PCS are determined by its partial control parameters. By customizing these parameters, high-dimensional chaotic maps can be obtained with any desired LEs. Because the dynamics of a high-dimensional chaotic map can be reflected by its LEs, the generated high-dimensional chaotic maps can exhibit complex and robust dynamic properties. To demonstrate the effectiveness of the nD-PCS, we provide two examples of new chaotic maps and implement them in a microcontroller-based hardware platform. The test results show that these two chaotic maps can generate chaotic signals with high randomness. Performance evaluations show that the nD-PCS can generate high-dimensional chaotic maps having desired LEs, better performance, and more uniformly distributed outputs, compared with existing high-dimensional chaotic maps. Finally, a chaos-based secure communication is developed, and the simulations show that the nD-PCS performs better than other high-dimensional chaotic maps in this application.

The remainder of this paper is organized as follows. Section II presents the nD-PCS and a methodology for configuring its parameter matrix and discusses its parameter space. Section III analyzes the chaotic behaviors of nD-PCS. Section IV presents two new high-dimensional chaotic maps generated by using nD-PCS as examples. Section V evaluates the properties of nD-PCS and compares its performance with those of other high-dimensional chaotic maps. Section VI applies the nD-PCS to a secure communication scheme, and Section VII concludes the paper.

II. *n*-Dimensional Polynomial Chaotic System

This section presents the nD polynomial chaotic system (nD-PCS). First, the structure of nD-PCS is presented. Then, a construction method is introduced to configure the parameter matrix of the nD-PCS. Finally, the parameter space for configuring the parameter matrix is discussed.

A. nD-PCS

The nD-PCS is generated from n parametric polynomials. First, we initialize the n general polynomials of arbitrary orders. Then, we perform a modular operation on these polynomials, and the nD-PCS is defined as

$$\mathbf{x}(i+1) = \mathbf{F}(\mathbf{x}(i)) \mod N, \tag{1}$$

where $\mathbf{x}(i) = \{x_1(i), x_2(i), \cdots, x_n(i)\}^T \in \mathbb{R}^{n\times 1}$ is the observed state of the system in the i-th observation time, N is the module coefficient that is an integer, and $\mathbf{F}(\mathbf{x})$ is a polynomial function of $I^n \to I^n$, which is defined as

$$\begin{cases} x_1(i+1) = a_{11}x_1(i) + a_{12}x_2(i)^{c_{12}} + \cdots + a_{1n}x_n(i)^{c_{1n}} \\ x_2(i+1) = a_{21}x_1(i)^{c_{21}} + a_{22}x_2(i) + \cdots + a_{2n}x_n(i)^{c_{2n}} \\ \qquad\vdots \\ x_n(i+1) = a_{n1}x_1(i)^{c_{n1}} + a_{n2}x_2(i)^{c_{n2}} + \cdots + a_{nn}x_n(i). \end{cases} \tag{2}$$

The polynomial function $\mathbf{F}(\mathbf{x})$ can be expanded as

$$\mathbf{F}(\mathbf{x}(i)) = \mathbf{D}\cdot\mathbf{x}(i), \tag{3}$$

where the matrix $\mathbf{D}$ is expressed as

$$\mathbf{D} = \begin{pmatrix} a_{11} & a_{12}x_2(i)^{c_{12}-1} & \cdots & a_{1n}x_n(i)^{c_{1n}-1} \\ a_{21}x_1(i)^{c_{21}-1} & a_{22} & \cdots & a_{2n}x_n(i)^{c_{2n}-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}x_1(i)^{c_{n1}-1} & a_{n2}x_2(i)^{c_{n2}-1} & \cdots & a_{nn} \end{pmatrix}. \tag{4}$$

All the parameters in $\mathbf{D}$ can be represented as two matrices $\mathbf{A}$ and $\mathbf{C}$, and

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{5}$$

and

$$\mathbf{C} = \begin{pmatrix} 1 & c_{12} & \cdots & c_{1n} \\ c_{21} & 1 & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & 1 \end{pmatrix}. \tag{6}$$

The parameter matrix $\mathbf{A}$ contains the parameters of the polynomials, whereas the coefficient matrix $\mathbf{C}$ contains the exponent coefficients of the variables in the polynomials. It is clear that the main diagonal elements of $\mathbf{C}$ are all one, and the elements $c_{11}, c_{22}, \ldots, c_{nn}$ can thus be omitted, whereas the other elements of $\mathbf{C}$ can be arbitrary constants. Because all the main diagonal elements of $\mathbf{C}$ are one, the dynamic properties of the proposed nD-PCS are mainly determined by the parameter matrix $\mathbf{A}$. By setting the elements of $\mathbf{A}$ as appropriate values, the nD-PCS can show robust and expected dynamic properties.

B. Methodology for Constructing A

The parameter matrix $\mathbf{A}$ was constructed using dimension expansion. Proposition 1 [42] is introduced.

Proposition 1: Let $\mathbf{A}_1$ *and* $\mathbf{A}_2$ *be an* iD *and a* jD *matrices, respectively. The* n *eigenvalues of the* $(i + j)D$ *matrix,*

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{M}_1 \\ \mathbf{M}_2 & \mathbf{A}_2 \end{pmatrix}, \tag{7}$$

are composed of the eigenvalues of $\mathbf{A}_1$ *and* $\mathbf{A}_2$ *if at least one of* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *is a zero-block matrix.* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are an* $i \times j$ *and a* $j \times i$ *matrices, respectively.*

Utilizing Proposition 1, an nD parameter matrix $\mathbf{A}$ is constructed from the special 1D matrix. Detailed descriptions of the construction procedure are as follows:

- *Step 1*: Generate a 1×1 special matrix as an initial matrix with a predefined matrix value.
- *Step 2*: Place the matrix and a predefined parameter in the main diagonal position of a block matrix of size 2×2, as in Eq. (7). To increase the parameter space, the location of the initial matrix was randomly set.
- *Step 3*: For the two matrix blocks $\mathbf{M}_1$ and $\mathbf{M}_2$ in the antidiagonal position, select one randomly and set the value of its element as zero. The elements in the other are assigned to some given values.
- *Step 4*: Set the current composite matrix to be the initial matrix.
- *Step 5*: Repeat the *Step 2* to *Step 4* $(n - 1)$ times using the new initial matrix each time.

Algorithm 1 presents the pseudocode of the above procedure for generating the parameter matrix $\mathbf{A}$ with inputs $\mathbf{a}$, $\mathbf{b}$, $\mathbf{g}$, $\mathbf{h}$, and n. The data sequence $\mathbf{a} = \{a_i\}_{i=1}^{n}$ contains the main diagonal elements of $\mathbf{A}$, and the data sequence $\mathbf{b} = \{b_i\}_{i=1}^{(n-1)\cdot n/2}$ contains the other elements of $\mathbf{A}$. The binary sequence $\mathbf{g} = \{g_i\}_{i=1}^{n-1}$ determines the spatial locations of $\mathbf{a}$, and the binary sequence $\mathbf{h} = \{h_i\}_{i=1}^{n-1}$ determines the spatial locations of $\mathbf{b}$ in the parameter matrix $\mathbf{A}$.

Algorithm 1 Generation of the Parameter Matrix **A**

```
Input: $\mathbf{a} = \{a_i\}_{i=1}^{n}$, $\mathbf{b} = \{b_i\}_{i=1}^{(n-1)\cdot n/2}$, $\mathbf{g} = \{g_i\}_{i=1}^{n-1}$ and $\mathbf{h} = \{h_i\}_{i=1}^{n-1}$, where $a_i$ and $b_i$ are predefined values, and $g_i, h_i \in \{0, 1\}$.
$\mathbf{A} = [a_1]$;
for $i = 2$ to $n$ do
    if $g_{i-1} == 1$ then
        $\mathbf{A}_1 = \mathbf{A}$, $\mathbf{A}_2 = [a_i]$,
        $\mathbf{M}_1 \in \mathbb{R}^{(i-1)\times 1}$, $\mathbf{M}_2 \in \mathbb{R}^{1\times(i-1)}$.
    else
        $\mathbf{A}_2 = \mathbf{A}$, $\mathbf{A}_1 = [a_i]$,
        $\mathbf{M}_1 \in \mathbb{R}^{1\times(i-1)}$, $\mathbf{M}_2 \in \mathbb{R}^{(i-1)\times 1}$.
    end if
    if $h_{i-1} == 1$ then
        $\mathbf{M}_1 = \mathbf{0}$, elements of $\mathbf{M}_2$ are fetched from $\mathbf{b}$.
    else
        $\mathbf{M}_2 = \mathbf{0}$, elements of $\mathbf{M}_1$ are fetched from $\mathbf{b}$.
    end if
    $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{M}_1 \\ \mathbf{M}_2 & \mathbf{A}_2 \end{pmatrix}$.
end for
Output: an $n$-D coefficient matrix $\mathbf{A}$.
```

C. Parameter Space

There are three types of parameters used to construct an n-D parameter matrix $\mathbf{A}$: main diagonal entity (MDE), block matrix entity (BME), and spatial location configuration (SLC). The elements in $\mathbf{a}$ and $\mathbf{b}$ are MDEs and BMEs, respectively, whereas the binary values in $\mathbf{g}$ and $\mathbf{h}$ are SLCs. Clearly, the sequence $\mathbf{a}$ contains n MDEs for constructing an nD parameter matrix $\mathbf{A}$. As shown in Algorithm 1, $n-1$ times of dimension expansion are required to construct an nD matrix, and $i-1$ BMEs are used in the i-th expansion. Thus, the total number of BMEs is $\#_{BDE} = (n-1)\cdot n/2$. For simplicity of calculation, all MDEs and BMEs were randomly chosen from a dataset with M possible values randomly. Then the parameter spaces of MDEs and BMEs are

$$P_{MDE} = M^n \quad \text{and} \quad P_{BME} = M^{(n-1)\cdot n/2},$$

respectively. A binary value in $\mathbf{g}$ indicates the direction of the matrix expansion, whereas that in $\mathbf{h}$ controls the position of the added zero sub-matrix in Eq. (7). A number of $n-1$ iterations are required to construct an nD parameter matrix, and thus, each of $\mathbf{g}$ and $\mathbf{h}$ has $n-1$ elements. The total number of SLCs required to construct an nD $\mathbf{A}$ is $\#_{SLC} = 2(n-1)$. Because each SLC is a binary number, the parameter space of all the SLCs in $\mathbf{g}$ and $\mathbf{h}$ is

$$P_{SLC} = 2^{\#_{SLC}} = 2^{2(n-1)}.$$

Because all the MDEs, BMEs, and SLCs are independent of each other, the entire parameter space to construct an nD parameter matrix $\mathbf{A}$ can be obtained by multiplying the parameter spaces of MDEs, BMEs, and SLCs, namely,

$$\begin{aligned} P_{\mathbf{A}} &= P_{MDE} \times P_{BME} \times P_{SLC} \\ &= 2^{2(n-1)} \cdot M^{(n+1)\cdot n/2}. \end{aligned}$$

TABLE I

PARAMETER SPACES OF CONSTRUCTING THE nD PARAMETER MATRIX **A** WITH DIFFERENT DIMENSIONS n

n	P_{MDE}	P_{BME}	P_{SLC}	$P_{\mathbf{A}}$
3	M^3	M^3	16	$16M^6$
4	M^4	M^6	64	$64M^{10}$
5	M^5	M^{10}	256	$256M^{15}$
6	M^6	M^{15}	1024	$1024M^{21}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	M^n	$M^{\frac{(n-1)\cdot n}{2}}$	$2^{2(n-1)}$	$2^{2(n-1)}\cdot M^{\frac{(n+1)\cdot n}{2}}$

Table I lists the parameter spaces of constructing the nD parameter matrix **A** with different dimensions n. It shows that a large number of parameter matrices can be generated using different settings of parameters. This indicates that one has a great flexibility to obtain a large number of high-dimensional chaotic maps with different parameter matrix **A**.

III. ROBUST CHAOTIC BEHAVIORS ANALYSIS

The robust chaotic behaviors indicate that a chaotic system has chaotic behaviors in the entire chaotic range, and no periodic windows exist [43]. A chaotic system with continuous chaotic ranges can exhibit robust chaotic behavior. When the parameter matrix **A** of the nD-PCS is constructed using Algorithm 1, the dynamic properties of the nD-PCS are determined using the partial parameters of **A**. By customizing these parameters, nD-PCS can exhibit complex and robust chaotic behaviors. This section investigates the relationship between the elements of **A** and the behavior of the nD-PCS. Among the different measures for describing chaos, the LE is one of the most widely used and accepted methods [5], [44]. This section first introduces chaos in the sense of LE for high-dimensional dynamical systems, and then analyzes the chaotic behaviors of the nD-PCS using LE.

A. Chaos in the Sense of LE

The LE defines chaos by characterizing the divergence rate of two trajectories of a dynamical system. If a dynamical system has a positive LE, the two close trajectories will be dissipative and separate in every unit time, thereby leading the system to be chaotic. If a dynamical system has more than one positive LE, its close trajectories are dissipative in several directions, and thus the system achieves hyperchaotic behaviors.

The definition of chaos in the sense of LE can be defined as Definition 1 [44].

Definition 1: A nonlinear system can generate chaos in the sense of LE if (1) at least one of its LEs is positive and (2) it is globally bounded.

Referring to Definition 1, one can judge whether a dynamical system has chaotic behavior. A multidimensional dynamical system has several LEs that describe the separation of close trajectories from different directions. For an nD discrete-time differentiable dynamical system

$$\mathbf{G}(\mathbf{x}):\begin{cases} x_1(i+1)=G_1(x_1(i),x_2(i),\cdots,x_n(i))\\ x_2(i+1)=G_2(x_1(i),x_2(i),\cdots,x_n(i))\\ \quad\vdots\\ x_n(i+1)=G_n(x_1(i),x_2(i),\cdots,x_n(i)),\end{cases}$$

its n LEs can be calculated as

$$LE_j=\lim_{t\to\infty}\frac{1}{t}\sum_{i=0}^{t-1}\ln|\lambda_j^{\mathbf{x}(i)}|,\tag{8}$$

where $j=1,2,\cdots,n$, and $\lambda_1^{\mathbf{x}(i)}\sim\lambda_n^{\mathbf{x}(i)}$ are the n eigenvalues of $\mathbf{J}(\mathbf{x}(i))$. $\mathbf{J}(\mathbf{x}(i))$ is the Jacobian matrix of the nD dynamical system with the observed state $\mathbf{x}(i)$, which can be calculated as

$$\mathbf{J}(\mathbf{x}(i))=\begin{pmatrix}\frac{\partial G_1}{\partial x_1(i)}|_{\mathbf{x}(i)} & \frac{\partial G_1}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \frac{\partial G_1}{\partial x_n(i)}|_{\mathbf{x}(i)}\\ \frac{\partial G_2}{\partial x_1(i)}|_{\mathbf{x}(i)} & \frac{\partial G_2}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \frac{\partial G_2}{\partial x_n(i)}|_{\mathbf{x}(i)}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial G_n}{\partial x_1(i)}|_{\mathbf{x}(i)} & \frac{\partial G_n}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \frac{\partial G_n}{\partial x_n(i)}|_{\mathbf{x}(i)}\end{pmatrix}.$$

B. Chaotic Behaviors of the nD-PCS

The n eigenvalues of the nD parameter matrix **A** are closely related to the partial parameters used in the construction of **A**. First, we introduce Lemma 1 to illuminate this relationship.

Lemma 1: The n eigenvalues of the nD matrix **A** *constructed by Algorithm 1 with parameters* ***a****,* ***b****,* ***g****, and* ***h*** *are the n elements of* ***a****.*

Proof: From the construction procedure in Algorithm 1, the nD matrix **A** is constructed by iteratively expanding the dimension, and the iD matrix is generated from the lower $(i-1)$D matrix and some parameters, namely

$$\mathbf{A}_{i\times i}=\begin{pmatrix} a_i & \mathbf{P}_{1\times(i-1)}\\ \mathbf{Q}_{(i-1)\times 1} & \mathbf{A}_{(i-1)\times(i-1)}\end{pmatrix},\tag{9}$$

or

$$\mathbf{A}_{i\times i}=\begin{pmatrix} \mathbf{A}_{(i-1)\times(i-1)} & \mathbf{Q}_{(i-1)\times 1}\\ \mathbf{P}_{1\times(i-1)} & a_i\end{pmatrix},\tag{10}$$

where a_i is the ith element of sequence **a**. One of the $\mathbf{P}_{1\times(i-1)}$ and $\mathbf{Q}_{(i-1)\times 1}$ is a zerovector, and the elements of the other are fetched from the sequence **b**.

Thus, the nD matrix $\mathbf{A}_{n\times n}$ in the last iteration is generated as follows:

$$\mathbf{A}_{n\times n}=\begin{pmatrix} a_n & \mathbf{P}_{1\times(n-1)}\\ \mathbf{Q}_{(n-1)\times 1} & \mathbf{A}_{(n-1)\times(n-1)}\end{pmatrix},\tag{11}$$

or

$$\mathbf{A}_{n\times n}=\begin{pmatrix} \mathbf{A}_{(n-1)\times(n-1)} & \mathbf{Q}_{(n-1)\times 1}\\ \mathbf{P}_{1\times(n-1)} & a_n\end{pmatrix}.\tag{12}$$

One of the $\mathbf{P}_{1\times(n-1)}$ and $\mathbf{Q}_{(n-1)\times 1}$ is a zerovector. The characteristic polynomial of matrix $\mathbf{A}_{n\times n}$ is calculated as follows:

$$\begin{aligned}&\det(\lambda\mathbf{E}_{n\times n}-\mathbf{A}_{n\times n})\\ &=\begin{vmatrix}\lambda\mathbf{E}_{(n-1)\times(n-1)}-\mathbf{A}_{(n-1)\times(n-1)} & \mathbf{Q}_{(n-1)\times 1}\\ \mathbf{P}_{1\times(n-1)} & \lambda-a_n\end{vmatrix}\\ &=0,\end{aligned}\tag{13}$$

or

$$\begin{aligned} &\det(\lambda \mathbf{E}_{n\times n} - \mathbf{A}_{n\times n})\\ &= \begin{vmatrix} \lambda - a_n & \mathbf{P}_{1\times(n-1)} \\ \mathbf{Q}_{(n-1)\times 1} & \lambda\mathbf{E}_{(n-1)\times(n-1)} - \mathbf{A}_{(n-1)\times(n-1)} \end{vmatrix}\\ &= 0, \end{aligned} \tag{14}$$

where λ denotes the eigenvalue of the matrix $\mathbf{A}_{n\times n}$, and $\mathbf{E}$ is an identity matrix. Referring to Proposition 1, when one of the $\mathbf{P}_{1\times(n-1)}$ and $\mathbf{Q}_{(n-1)\times 1}$ is a zerovector, the eigenvalues of $\mathbf{A}_{n\times n}$ are composed of a_n and the eigenvalues of $\mathbf{A}_{(n-1)\times(n-1)}$. This indicates that

$$\begin{aligned} &\det(\lambda \mathbf{E}_{n\times n} - \mathbf{A}_{n\times n})\\ &= (\lambda - a_n)\cdot \det(\lambda\mathbf{E}_{(n-1)\times(n-1)} - \mathbf{A}_{(n-1)\times(n-1)})\\ &= 0. \end{aligned} \tag{15}$$

Similarly, one can obtain that the characteristic polynomial of matrix $\mathbf{A}_{(n-1)\times(n-1)}$ is calculated as follows:

$$\begin{aligned} &\det(\lambda \mathbf{E}_{(n-1)\times (n-1)} - \mathbf{A}_{(n-1)\times (n-1)})\\ &= (\lambda - a_{n-1})\cdot \det(\lambda\mathbf{E}_{(n-2)\times(n-2)} - \mathbf{A}_{(n-2)\times(n-2)})\\ &= 0. \end{aligned} \tag{16}$$

Finally, the characteristic polynomial of the matrix $\mathbf{A}_{n\times n}$ can be obtained as

$$\begin{aligned} &\det(\lambda \mathbf{E}_{n\times n} - \mathbf{A}_{n\times n})\\ &= (\lambda - a_n)(\lambda - a_{n-1})\cdots(\lambda - a_2)(\lambda - \mathbf{A}_{1\times 1})\\ &= 0. \end{aligned} \tag{17}$$

Because $\mathbf{A}_{1\times 1}$ is the first element of the sequence $\mathbf{a}$, namely $\mathbf{A}_{1\times 1} = a_1$, then

$$\begin{aligned} \det(\lambda \mathbf{E}_{n\times n} - \mathbf{A}_{n\times n}) &= (\lambda - a_n)(\lambda - a_{n-1})\cdots(\lambda - a_1)\\ &= 0. \end{aligned} \tag{18}$$

Thus, the n eigenvalues of the matrix $\mathbf{A}_{n\times n}$, denoted as $\lambda_1 \sim \lambda_n$, are the n elements of the sequence $\mathbf{a}$, which means that

$$\lambda_j = a_j, \quad j = 1, 2, \cdots, n. \tag{19}$$

This completes the proof. ■

Referring to Lemma 1, we can deduce Lemma 2 that illuminates the chaotic behavior of the nD-PCS.

Lemma 2: The nD-PCS in Eq. (1) *with parameter matrix* $\mathbf{A}$ *constructed by Algorithm 1 exhibits chaotic behavior if the elements in sequence* $\boldsymbol{a}$ *satisfy* $\exists|a_j| > 1$ *for* $j = 1, 2, \cdots, n$.

Proof: The Jacobian matrix of nD-PCS can be calculated as Eq. (20), as shown at the bottom of the page. Let the nD matrix $\mathbf{O}$ be

$$\mathbf{O} = \begin{pmatrix} 1 & x_2(i)^{c_{12}-1} & \cdots & x_n(i)^{c_{1n}-1} \\ x_1(i)^{c_{21}-1} & 1 & \cdots & x_n(i)^{c_{2n}-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1(i)^{c_{n1}-1} & x_2(i)^{c_{n2}-1} & \cdots & 1 \end{pmatrix} \tag{21}$$

and denote the Hadamard product as $*$. The Jacobian matrix $\mathbf{J}(\mathbf{x}(i))$ in Eq. (20) can be represented as

$$\mathbf{J}(\mathbf{x}(i)) = \mathbf{A} * \mathbf{C} * \mathbf{O}. \tag{22}$$

According to the calculation in Eq. (18), the eigenvalues of the matrix $\mathbf{A}$ are determined only by the main diagonal elements a_{ii}. These eigenvalues cannot be changed when the off-main diagonal elements of $\mathbf{A}$ are multiplied by any value. From Eq. (22), we can obtain $\mathbf{J}(\mathbf{x}(i))_{jk} = a_{jk}\cdot c_{jk}\cdot o_{jk}$, where o_{jk} is an element in the matrix $\mathbf{O}$. Because $c_{kk} = o_{kk} = 1$, $\mathbf{J}(\mathbf{x}(i))$ has the same main diagonal elements as $\mathbf{A}$. Then, the n eigenvalues of $\mathbf{J}(\mathbf{x}(i))$ and $\mathbf{A}$ are the same.

Referring to Lemma 1, the n eigenvalues of $\mathbf{A}$ are the n elements of sequence $\mathbf{a}$. Thus, the n eigenvalues of the Jacobian matrix $\mathbf{J}(\mathbf{x}(i))$ are also n elements of the sequence $\mathbf{a}$. This indicates that the eigenvalues of the Jacobian matrix $\mathbf{J}(\mathbf{x}(i))$ in different observed states are the same and are independent of the observed state $\mathbf{x}(i)$. Suppose that the n eigenvalues of $\mathbf{J}(\mathbf{x}(i))$ are $\lambda_1^{\mathbf{x}(i)} \sim \lambda_n^{\mathbf{x}(i)}$. Combined with Eq. (19), we obtain:

$$\lambda_j^{\mathbf{x}(0)} = \cdots = \lambda_j^{\mathbf{x}(i)} = \cdots = \lambda_j^{\mathbf{x}(t-1)} = a_j, \quad j = 1, 2, \cdots, n. \tag{23}$$

According to the LE calculation of the discrete-time dynamical system in Eq. (8), the n LEs of the nD-PCS are calculated as

$$\begin{aligned} LE_j &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} \ln|\lambda_j^{\mathbf{x}(i)}|\\ &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} \ln|a_j|\\ &= \ln|a_j|, \end{aligned} \tag{24}$$

$$\begin{aligned} \mathbf{J}(\mathbf{x}(i)) &= \begin{pmatrix} \dfrac{\partial x_1(i+1)}{\partial x_1(i)}|_{\mathbf{x}(i)} & \dfrac{\partial x_1(i+1)}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \dfrac{\partial x_1(i+1)}{\partial x_n(i)}|_{\mathbf{x}(i)} \\ \dfrac{\partial x_2(i+1)}{\partial x_1(i)}|_{\mathbf{x}(i)} & \dfrac{\partial x_2(i+1)}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \dfrac{\partial x_2(i+1)}{\partial x_n(i)}|_{\mathbf{x}(i)} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial x_n(i+1)}{\partial x_1(i)}|_{\mathbf{x}(i)} & \dfrac{\partial x_n(i+1)}{\partial x_2(i)}|_{\mathbf{x}(i)} & \cdots & \dfrac{\partial x_n(i+1)}{\partial x_n(i)}|_{\mathbf{x}(i)} \end{pmatrix}\\ &= \begin{pmatrix} a_{11} & a_{12}\cdot c_{12}\cdot x_2(i)^{c_{12}-1} & \cdots & a_{1n}\cdot c_{1n}\cdot x_n(i)^{c_{1n}-1} \\ a_{21}\cdot c_{21}\cdot x_1(i)^{c_{21}-1} & a_{22} & \cdots & a_{2n}\cdot c_{2n}\cdot x_n(i)^{c_{2n}-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\cdot c_{n1}\cdot x_1(i)^{c_{n1}-1} & a_{n2}\cdot c_{n2}\cdot x_2(i)^{c_{n2}-1} & \cdots & a_{nn} \end{pmatrix} \end{aligned} \tag{20}$$

where $j = 1, 2, \cdots, n$. Because $\exists |a_j| > 1$, for $j = 1, 2, \cdots, n$, one obtains that the nD-PCS has at least one positive LE. Moreover, because the modular operation in Eq. (1) can transform any input to a fixed range, and nD-PCS is globally bounded. Thus, the nD-PCS can satisfy the requirements of chaos's definition in the sense of LE in Definition 1 and thereby exhibits chaotic behavior. This completes the proof. ■

Referring to Lemma 2, we can see that when the elements of the sequence **a** satisfy $\exists |a_j| > 1$, for $j = 1, 2, \cdots, n$, the constructed parameter matrix **A** exists eigenvalues that are outside the unit circle, thereby allowing the nD-PCS to exhibit chaotic behaviors. Each LE of the nD-PCS is independently determined through Eq. (24) by using one element of the sequence **a**. By customizing all the elements $a_1, a_2, \ldots, a_n$ of **a** to construct the parameter matrix **A**, one can allow the nD-PCS to own n LEs with any large values. Because the dynamic properties of a chaotic system can be reflected by its LEs [36], the nD-PCS can exhibit robust chaotic behaviors and expected dynamical properties by customizing the values of its n LEs. The n is the dimension of the proposed system. Therefore, the theoretical minimum value of n can be equal to one.

IV. Two Illustrative Examples

To demonstrate the effectiveness of nD-PCS in Eq. (1), this section provides two examples of high-dimensional chaotic maps constructed by the proposed method: a four-dimensional polynomial chaotic map (4D-PCM) and a seven-dimensional polynomial chaotic map (7D-PCM). Then, a hardware platform was developed to demonstrate the easy hardware implementation of the proposed chaotic maps. Finally, we tested the randomness properties of the chaotic signals generated by the two chaotic maps.

A. Construction of Two Examples of High-Dimensional Chaotic Map

When constructing the parameter matrix **A**, all the elements in sequences **a** and **b** are randomly selected from an integer set $M = \{1, 2, \cdots, 500\}$, and all the elements in sequences **g** and **h** are randomly generated binary numbers. For simplicity, all the exponent coefficients in **C** of Eq. (6) are set to $c_{ij} = 2$, and the modular coefficient is set to $N = 1$. One is the flexibility to set them as other values.

1) 4D-PCM: When generating a 4D-PCM, a 4D parameter matrix should be constructed. Then, the lengths of the data sequences **a**, **b**, **g**, and **h** are four, four, three, and three, respectively. These generated parameters are $\mathbf{a} = \{421, 33, 233, 251\}$, $\mathbf{b} = \{224, 171, 420, 492, 314, 91\}$, $\mathbf{g} = \{1, 1, 0\}$, and $\mathbf{h} = \{0, 1, 0\}$. According to the construction process in Algorithm 1, the 4D parameter matrix **A** can be generated as follows:

$$\mathbf{A} = \begin{pmatrix} 251 & 492 & 314 & 91 \\ 0 & 421 & 224 & 0 \\ 0 & 0 & 33 & 0 \\ 0 & 171 & 420 & 233 \end{pmatrix}. \tag{25}$$

When all the exponent coefficients in **C** are two, and the modular coefficient is set to one in Eq. (1), a 4D-PCM can be generated as:

$$\begin{cases} x_1(i+1) = 251x_1(i) + 492x_2(i)^2 + 314x_3(i)^2 \\ \qquad\qquad + 91x_4(i)^2 \mod 1 \\ x_2(i+1) = 421x_2(i) + 224x_3(i)^2 \mod 1 \\ x_3(i+1) = 33x_3(i) \mod 1 \\ x_4(i+1) = 171x_2(i)^2 + 420x_3(i)^2 + 233x_4(i) \mod 1. \end{cases} \tag{26}$$

The equilibrium point is a point of a dynamical system that maps to itself in its domain. For the 4D-PCM, its equilibrium points $(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4)$ are the solutions of the equations

$$\begin{cases} \hat{x}_1 = (251\hat{x}_1 + 492\hat{x}_2^2 + 314\hat{x}_3^2 + 91\hat{x}_4^2) \mod 1 \\ \hat{x}_2 = (421\hat{x}_2 + 224\hat{x}_3^2) \mod 1 \\ \hat{x}_3 = 33\hat{x}_3 \mod 1 \\ \hat{x}_4 = (171\hat{x}_2^2 + 420\hat{x}_3^2 + 233\hat{x}_4) \mod 1. \end{cases} \tag{27}$$

By resolving the above equation, one can get that the 4D-PCM has many equilibrium points including (0.9667, 0.662, 0.313, 0.362), (0.613, 0.982, 0.625, 0.737), (0.650, 0.627, 0.781, 0.761), (0.679, 0.246, 0.938, 0.520), (0.237, 0.359, 0.469, 0.800) and many other points. According to Eq. (23), the eigenvalues of the Jacobian matrix are equal to the four elements in sequence **a** and have no relationship with the observed states. Then, the four eigenvalues for every equilibrium point are $\lambda_1 = 251$, $\lambda_2 = 421$, $\lambda_3 = 33$, and $\lambda_4 = 233$. This indicates that all the equilibrium points of the 4D-PCM are unstable node-foci. According to Eq. (24), the four LEs of the 4D-PCM are $LE_1 = \ln(251) = 5.5255$, $LE_2 = \ln(421) = 6.0426$, $LE_3 = \ln(33) = 3.4965$ and $LE_4 = \ln(233) = 5.4510$. It is clear that the 4D-PCM has four positive LEs, and one can allow it to show expected and robust dynamic properties by customizing the elements in sequences **a**. With four positive LEs, the 4D-PCM exhibited hyperchaotic behavior.

2) 7D-PCM: To construct a 7D-PCM, the data sequences **a** and **b** are randomly generated from the data set M, and the binary sequences **g** and **h** are randomly generated. These parameters are $\mathbf{a} = \{93, 55, 51, 247, 99, 449, 51\}$, $\mathbf{b} = \{23, 279, 387, 156, 90, 170, 106, 256, 454, 315, 51, 196, 28, 251, 216, 499, 406, 243, 448, 69, 196\}$, $\mathbf{g} = \{0, 1, 1, 1, 0, 1\}$, and $\mathbf{h} = \{0, 1, 1, 1, 0, 1\}$. Then, the 7D parameter matrix **A** can be generated using Algorithm 1 as

$$\mathbf{A}_2 = \begin{pmatrix} 449 & 51 & 196 & 28 & 251 & 216 & 0 \\ 0 & 55 & 23 & 0 & 0 & 0 & 0 \\ 0 & 0 & 93 & 0 & 0 & 0 & 0 \\ 0 & 279 & 387 & 51 & 0 & 0 & 0 \\ 0 & 156 & 90 & 170 & 247 & 0 & 0 \\ 0 & 106 & 256 & 454 & 315 & 99 & 0 \\ 499 & 406 & 243 & 448 & 69 & 196 & 51 \end{pmatrix}. \tag{28}$$

When all the exponent coefficients in **C** are set to two, and the modular coefficient is set to one in Eq. (1), a 7D-PCM can be generated. When setting $\mathbf{x}(i+1) = \mathbf{x}(i)$, it can be calculated that the 7D-PCM also has many equilibrium points. According to Eq. (23), the eigenvalues of the Jacobian

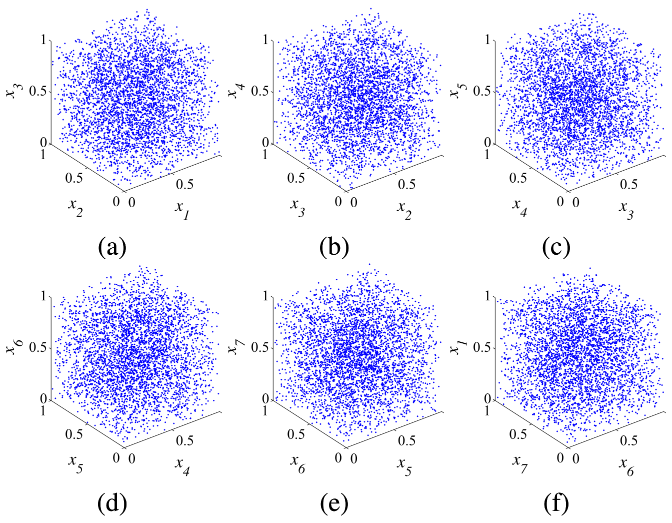


Fig. 1. The hyperchaotic sequences of the constructed 7D-PCM with seven positive LEs in 3D phase space. (a) $x_1 - x_2 - x_3$ space. (b) $x_2 - x_3 - x_4$ space. (c) $x_3 - x_4 - x_5$ space. (d) $x_4 - x_5 - x_6$ space. (e) $x_5 - x_6 - x_7$ space and (f) $x_6 - x_7 - x_1$ space.

matrix of the 7D-PCM are the seven elements of **a**, indicating that $\lambda_1 = 449$, $\lambda_2 = 55$, $\lambda_3 = 93$, $\lambda_4 = 51$, $\lambda_5 = 247$, $\lambda_6 = 99$, and $\lambda_7 = 51$. Then, all the equilibrium points of the 7D-PCM are unstable node-foci. According to Eq. (24), the seven LEs of the 7D-PCM are $LE_1 = \ln(449) = 6.1070$, $LE_2 = \ln(55) = 4.0073$, $LE_3 = \ln(93) = 4.5326$, $LE_4 = \ln(51) = 3.9318$, $LE_5 = \ln(247) = 5.5094$, $LE_6 = \ln(99) = 4.5951$ and $LE_7 = \ln(51) = 3.9318$. With seven positive LEs, the 7D-PCM exhibits complex dynamics and can generate hyperchaotic behavior. In Fig. 1, the hyperchaotic sequences of the 7D-PCM are plotted with the initial state $\mathbf{x}(0) = \mathbf{0.7}_{7\times 1}$ in the 3D phase space. These results show that the chaotic sequences are uniformly distributed in the 3D phase space.

B. Hardware Implementation

When a chaotic system is employed in industrial applications, implementing it on a hardware platform is a necessary step. A microcontroller-based platform was developed to demonstrate the easy hardware implementation of the 4D-PCM and 7D-PCM.

Because the microcontroller has many favorable characteristics such as easy programming, low power consumption, and strong environmental adaptability, it is widely used in industrial applications. In our experiment, a microcontroller-based platform was developed, which includes a high-performance microcontroller STM32F407VET6, 12-bit D/A converter TLV5618, oscilloscope TDS3054C, and other peripheral circuits. The mathematical models of 4D-PCM and 7D-PCM were first programmed using the C language, and the programs were then downloaded to the microcontroller. The D/A converter converts the generated digital chaotic signals into analog voltage signals, which are displayed by an oscilloscope. Fig 3 shows the hardware devices used in the microcontroller-based platform. Algorithm 2 presents the pseudocode of the microcontroller-based experiment, and its flowchart is shown in Fig. 2. In this flowchart, the vectors $\mathbf{x}(i) = \{x_1(i), x_2(i), \ldots, x_n(i)\}^T$, $\mathbf{step} = \{step_1, step_2, \ldots, step_n\}^T$, and $\mathbf{value} = \{value_1, value_2, \ldots, value_n\}^T$.

Algorithm 2 The Pseudocode of the Microcontroller-Based Hardware Experiment

Input: The output sequence length K, interpolation number M, parameter matrix **A** and coefficient matrix **C**, and initial value $\mathbf{x}(0)$.

```
for $i = 0$ to $K$ do
    $x_1(i+1) = a_{11}x_1(i) + a_{12}x_2(i)^{c_{12}} + \cdots + a_{1n}x_n(i)^{c_{1n}}$ mod 1;
    $\cdots$
    $x_n(i+1) = a_{n1}x_1(i)^{c_{n1}} + a_{n2}x_2(i)^{c_{n2}} + \cdots + a_{nn}x_n(i)$ mod 1;
    $step_1 = (x_1(i+1) - x_1(i))/M$;
    $\cdots$
    $step_n = (x_n(i+1) - x_n(i))/M$;
    for $j = 1$ to $M - 1$ do
        $value_1 = (x_1(i) + j \times step_1 + 15) \times 4096/30$;
        $\cdots$
        $value_n = (x_n(i) + j \times step_n + 15) \times 4096/30$;
        Transfer the $value_1, value_2, \cdots, value_n$ to TLV5618;
    end for
end for
```

Output: The results $value_1$, $value_2$, $\cdots$, $value_n$ in each iteration for oscilloscope displaying.

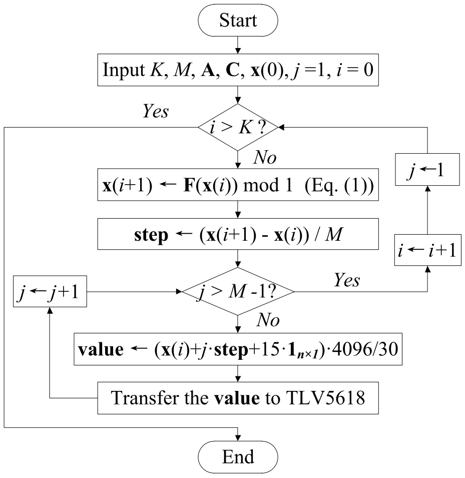

Start
Input K, M, A, C, x(0), j =1, i = 0
Yes
i > K ?
No
x(i+1) ← F(x(i)) mod 1 (Eq. (1))
step ← (x(i+1) - x(i)) / M
j ← 1
i ← i+1
j ← j+1
j > M -1?
Yes
No
value ← (x(i)+j·step+15·1n×1)·4096/30
Transfer the value to TLV5618
End

Fig. 2. The flowchart of the hardware implementation.

For the implementation resources, the MCU frequency is 168 MHz, and an external clock was chosen as its clock source. The core of STM32F407 is an ARM 32-bit Cortex-M4 CPU with an FPU. At room temperature, the experiment was operated with a 3.3V power supply voltage and an approximately 40 ~ 87mA power supply current. The initial states for the 4D-PCM and 7D-PCM were set to $\mathbf{x}(0) = \mathbf{0.7}_{4\times 1}$ and $\mathbf{x}(0) = \mathbf{0.7}_{7\times 1}$, respectively. Fig. 4 displays the outputs of the first 55 iterations of the 4D-PCM and 7D-PCM captured by the oscilloscope. Each sequence x_i oscillates randomly within a fixed range, which indicates the correctness and feasibility of the implementation of the 4D-PCM and 7D-PCM.

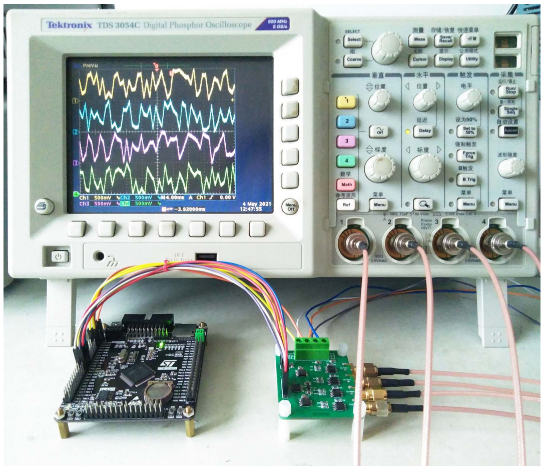


Fig. 3. Hardware devices of microcontroller-based platform.

C. Randomness Test

Because the state space in digital platforms cannot have infinite states, all self-evolved systems with external disturbances will finally result in periodic behaviors when implemented on a digital platform. Different chaotic systems may exhibit different chaos degradation times. In this subsection, we test the randomness properties of the chaotic sequences of the 4D-PCM and 7D-PCM to demonstrate their ability to resist chaos degradation. At present, there are many test standards for testing the randomness of random sequences, such as NIST, DIEHARD, and TestU01. To the best of our knowledge, the TestU01 standard can test much longer random sequences and is much stricter than any other test standard. Many existing chaotic systems cannot pass all the test suites of the TestU01 standard. This is because the longest binary sequence that can be measured by TestU01 can reach 10Tb. TestU01 offers a collection of utilities for empirical statistical testing of PRNGs [45]. TestU01 includes eight test batteries that aim to find the non-random area of random number sequences from different aspects. These eight test batteries include the SmallCrush, Crush, BigCrush, Rabbit, Alphabit, BlockAlphabit, PseudoDIEHARD, and FIPS-140-2 test suites. Each test suite applies a number of statistical tests to the tested random sequence, and each statistical test can generate a p-value. The tested random sequence can pass the corresponding statistical test if the generated p-value falls into the interval $[0.001, 0.999]$.

The TestU01 test standard can directly use the float numbers within the range $[0, 1]$ as input. However, the generated chaotic sequences in our systems are exactly float numbers within $[0, 1]$. Thus, these chaotic signals can be directly used as tested random numbers. Specifically, we set the initial values of the tested 4D-PCM and 7D-PCM as $\mathbf{x}(0) = \mathbf{0.7}_{4\times 1}$ and $\mathbf{x}(0) = \mathbf{0.7}_{7\times 1}$, respectively, and the produced first-dimension chaotic signal is used as random numbers. Because the Rabbit, Alphabit, and BlockAlphabit test suites can take variable lengths of random sequences as input, we use a fixed length of 32 Gb in these three test suites. The other five test suites take a random sequence with a fixed length of as input, and the BigCrush test suite is the most stringent test suite that applies 160 statistical tests to approximately 10 Tb data. Table II lists the test results of the 4D-PCM and 7D-PCM under these eight test suites. Since the initial state has only one significant digit, we exclude the first 40000 states. It can be seen that the 4D-PCM and 7D-PCM can pass all the statistical tests in all the test suites, which indicates that they can generate a large number of sequences with high randomness.

TABLE II

THE TEST RESULTS OF TESTU01 FOR THE 4D-PCM AND 7D-PCM

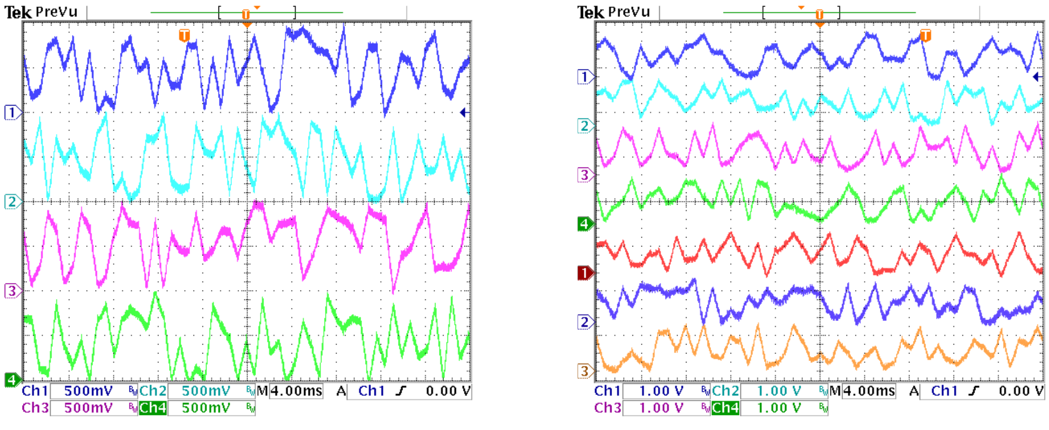
	Evaluated data	4D-PCM	7D-PCM
SmallCrush	About 6 Gb	15/15	15/15
Crush	About 1 Tb	144/144	144/144
BigCrush	About 10 Tb	160/160	160/160
Rabbit	32 Gb	40/40	40/40
Alphabit	32 Gb	17/17	17/17
BlockAlphait	32 Gb	102/102	102/102
PseudoDIEHARD	About 5 Gb	126/126	126/126
FIPS-140-2	About 19 Kb	16/16	16/16

V. PERFORMANCE EVALUATIONS

This section analyzes the performance of the nD-PCS from the following three aspects: LE, sample entropy (SE) [46], and information entropy (IE). When generating the parameter matrix $\mathbf{A}$, all the elements in sequences $\mathbf{a}$ and $\mathbf{b}$ are randomly selected from the integer set $M = \{1, 2, \cdots, 500\}$, and all the elements in $\mathbf{g}$ and $\mathbf{h}$ are randomly generated binary values. We also compared the nD-PCS with some existing high-dimensional chaotic map generation methods and 3D chaotic maps. These competing high-dimensional chaotic map generation methods include Shen's method [39], Liu's method [38], Chen's method [41] and Wu's method [47], whereas the competing 3D chaotic maps include the maps proposed by Xie [48], Aqeel [49], Lai [50], and Karawia [51].

When comparing the performance of high-dimensional chaotic maps produced by various chaotic map generation methods, a large number of high-dimensional chaotic maps with different dimensions should be generated. However, for these competing high-dimensional chaotic map generation methods, the authors only provided a few examples of chaotic maps with fixed dimensions in the literature. When generating chaotic maps with higher dimensions, some parameter settings reported cannot result in the expected properties, and therefore, these parameters should be adjusted. To provide a relatively fair comparison, we set the parameters in the chaotic map generation methods using the following rules: (1) if the parameter settings reported can result in any nD chaotic maps with expected properties, we directly reference these parameter settings; (2) if the parameters reported cannot result in chaotic maps for some dimensions with expected performance, we adjust these parameters to ensure that all the generated nD chaotic maps can show the properties expected in the literature; and (3) under the premise that the generated nD chaotic maps can possess the properties expected in the literature, we set the parameters to the same level as that in our proposed nD-PCS.

As a result, the parameters are set as $a = -0.1$ and $\epsilon \in [1, 20]$ for Shen's method, $a = 1$, $b \in [3, 5]$ and $c \in [0, 5]$



(a) (b)

Fig. 4. Simulation results captured from hardware platform for (a) 4D-PCM and (b) 7D-PCM. The time sequences from top to bottom in (a) are the outputs of x_1, x_2, x_3, x_4 in the 4D-PCM while in (b) are the outputs of $x_1, x_2, \cdots, x_7$ in the 7D-PCM.

for Liu's method, $\epsilon \in [100, 500]$ and $\sigma = 0.01$ for Chen's method, and $b = 0.1$ and $a \in [1.5, 1.7]$ for Wu's method. The parameters are randomly selected within their respective ranges when generating an nD chaotic map. The parameters in all the competing 3D chaotic maps are assigned the same values as those in the literatures.

A. LE

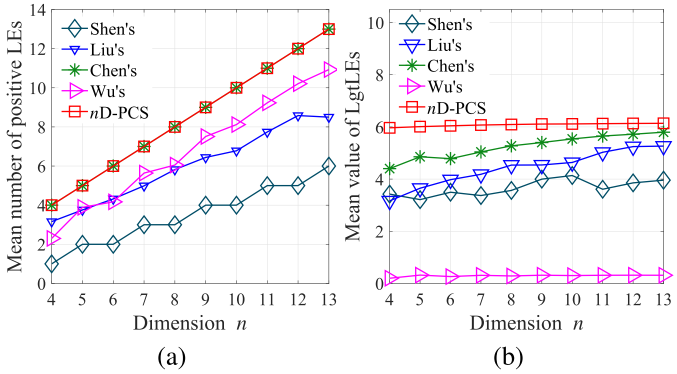
As mentioned in Section III-A, LE indicates the existence of chaos. A positive LE indicates that the corresponding dynamical system exhibits chaotic behavior if the system is globally bounded. A high-dimensional chaotic map has multiple LEs, and more than one positive LE indicates hyperchaotic behavior, which is a more complicated behavior than chaotic behavior. According to the discussions in [36], the dynamic properties of a high-dimensional chaotic map can be reflected by the number of positive LEs and their values. A high-dimensional chaotic map with more and larger positive LEs usually exhibits richer dynamic properties. Thus, for each chaotic map, we calculated the number of positive LEs and the largest LE (LgtLE).

Two groups of experiments were designed for the proposed nD-PCS and the aforementioned chaotic map generation methods and 3D chaotic maps. The first group tests the LEs of 3D chaotic maps. For each of the five chaotic map generation methods, we first randomly generated one 3D chaotic map and then calculated its LEs. Table III lists the three LEs of the 3D chaotic maps generated by the 3D chaotic maps and these 3D chaotic maps generated by five chaotic map generation methods. It can be seen that the 3D chaotic maps generated by Liu's method [38], Chen's method [41], Wu's method [47], and the proposed nD-PCS have three positive LEs. However, the 3D chaotic map generated by the nD-PCS has much larger LEs than the other 3D chaotic maps.

TABLE III

THE LEs FOR DIFFERENT 3D CHAOTIC MAPS AND THESE 3D CHAOTIC MAPS GENERATED BY DIFFERENT GENERATION METHODS

	LE_1	LE_2	LE_3	Number of positive LE
Xie's [48]	0.0365	0.0096	-0.6063	2
Aqeel's [49]	0.8785	0.0000	-14.5394	1
Lai's [50]	0.3111	0.0000	-1.2453	1
Karawia's [51]	0.9451	0.2557	-4.8134	2
Shen's [39]	1.4699	0.0000	-1.7689	1
Liu's [38]	3.2335	1.5496	0.2102	3
Chen's [41]	3.2425	3.2425	3.1820	3
Wu's [47]	0.3215	0.2060	-5.1326	2
nD-PCS	5.4190	5.2595	5.1338	3



(a) (b)

Fig. 5. The LEs for 100 nD chaotic maps generated by different generation methods with dimension $n \in \{4, 5, \cdots, 13\}$ (a) the mean numbers of positive; (b) the mean LgtLEs.

The second group of experiments tested five chaotic map generation methods with different system dimensions. For each generation method with dimensions $n \in \{4, 5, \cdots, 13\}$, we randomly generated 100 nD chaotic maps and calculated their mean numbers of positive LEs and mean LgtLEs. Fig. 5(a) plots the mean number of positive LEs, whereas Fig. 5(b) plots the mean LgtLEs. As shown in Fig. 5(a), both the proposed nD-PCS and Chen's method [41] can generate nD chaotic maps with the maximum number of positive LEs, namely, n. The nD chaotic maps generated by Shen's method [39] can own $\lfloor n-2 \rfloor/2$ positive LEs, and the high-dimensional chaotic maps generated by the other two methods are not fixed. From Fig. 5(b), it can be seen that the chaotic maps generated by the nD-PCS have the largest LgtLEs compared with the other methods. In addition, from the LE calculation in Eq. (24), the LEs of the chaotic maps by the nD-PCS are independently determined by the parameters

TABLE IV
THE SEs FOR DIFFERENT 3D CHAOTIC MAPS AND THESE 3D CHAOTIC MAPS GENERATED BY DIFFERENT METHODS

	SE_1	SE_2	SE_3
Xie's [48]	0.7882	0.7882	0.7882
Aqeel's [49]	0.1993	0.1887	0.1809
Lai's [50]	0.0432	0.0469	0.0464
Karawia's [51]	0.4980	0.3601	0.4157
Shen's [39]	0.0709	0.0672	0.0636
Liu's [38]	1.3577	1.3656	1.3611
Chen's [41]	2.1028	2.0852	2.1844
Wu's [47]	0.5341	0.5270	0.5362
*n*D-PCS	2.1968	2.1930	2.1911

in the sequence **a**. They can achieve any desired value by predefining the elements of **a** in the generation process.

B. SE

SE is a type of approximate entropy that can be used to test the degree of complexity of the time series output using chaotic maps. A time series with a positive SE shows irregular properties. A larger SE indicates more irregularity in the time series; it further indicates more complexity of the corresponding chaotic map.

Our experiments used the calculation method introduced in [46] to obtain the SEs of different high-dimensional chaotic maps. First, we tested the SEs of the three time series outputs by different 3D chaotic maps. Table IV lists the three SEs of the 3D chaotic maps produced by the chaotic map generation methods and other 3D chaotic maps. This shows that the 3D chaotic map generated by the proposed *n*D-PCS has the largest SE. In addition, we tested the SEs of five chaotic map generation methods with dimensions $n \in \{4, 5, \cdots, 13\}$. For each method, we randomly generated 100 *n*D chaotic maps and initial states $\mathbf{x}(0) \in [0, 1]$. Because each *n*D chaotic map can generate n time series and each time series has an SE, the SE of an *n*D chaotic map is calculated by averaging the n SEs. Fig. 6 plots the mean SEs of the 100 chaotic maps generated using different generation methods with different dimensions n. It can be seen that with an increase in dimension n, the *n*D chaotic maps generated by each generation method have similar SEs. The proposed *n*D-PCS can generate *n*D chaotic maps with the largest mean SEs. These results verify that the *n*D-PCS can produce high-dimensional chaotic maps with complex chaotic behaviors.

C. IE

IE is a widely used measure to characterize the randomness of a signal. It can be used to test the randomness of the observed states generated by an *n*D chaotic map. The observed state of an *n*D chaotic map $\mathbf{x}(i) = \{x_1(i), x_2(i), \cdots, x_n(i)\}^T \in R^{n\times 1}$ has n dimensions. When uniformly dividing the output range of each dimension into I intervals, the *n*D phase space can be divided into the I^n sub-phase space. Then, the IE of the output sequence of an *n*D chaotic map can be calculated as follows:

$$IE = -\sum_{k=1}^{I^n} Pr(k) \log_2 Pr(k), \tag{29}$$

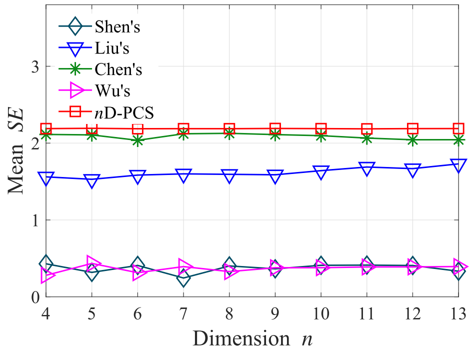
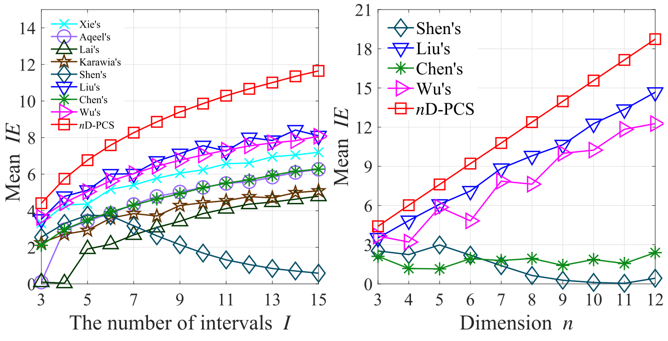


Fig. 6. The mean SEs for 100 *n*D chaotic maps generated by different generation methods with dimension $n \in \{4, 5, \cdots, 13\}$.



(a) (b)

Fig. 7. IEs for different chaotic maps. (a) The IEs for different 3D chaotic maps against different numbers of intervals; (b) the mean IEs for 100 *n*D chaotic maps generated by five chaotic map generation methods against different dimensions by fixing the number of intervals $I = 3$.

where $Pr(k)$ denotes the probability of the observed states in the k-th sub-phase space. It is clear that the IE is a positive value and has a theoretical maximum value when the probabilities of the observed states in each sub-phase plane are the same. This means that when $Pr(k) = 1/I^n$ $(k = 1, 2, \cdots, I^n)$, the IE can reach the maximum value and $IE_{\max} = n \log_2 I$. A larger IE indicates a more uniform distribution of the observed states, and further implies more complex properties of the related chaotic map.

Two groups of experiments were conducted to calculate the IE of the output sequences for different chaotic maps. The first group analyzes the IEs against the number of intervals $I = \{3, 4, \cdots, 15\}$ by fixing the dimension $n = 3$. For each 3D chaotic map, the initial state is set to $\mathbf{x}(0) \in [0, 1]$, and a number of I^4 output states are generated to calculate its IE. Fig. 7(a) plots the mean IEs of 100 3D chaotic maps generated by different chaotic map generation methods and the IEs of the other 3D chaotic maps. It is clear that the 3D chaotic map generated using the proposed *n*D-PCS has the largest IE among all the 3D chaotic maps. Table V shows the maximum IEs of the 100 3D chaotic maps randomly generated using the five chaotic map generation methods with different numbers of intervals. As shown, the 3D chaotic maps generated using the *n*D-PCS have the largest IEs, and their maximum IEs are very close to the theoretical maximum values.

The other group of experiments was designed to test the IEs against the system dimension $n = \{4, 5, \cdots, 12\}$ by

TABLE V

THE MAXIMUM IEs FOR 100 3D CHAOTIC MAPS GENERATED BY DIFFERENT CHAOTIC MAP GENERATION METHODS AGAINST DIFFERENT NUMBERS OF INTERVALS I

	Number of intervals I												
	3	4	5	6	7	8	9	10	11	12	13	14	15
Shen's [39]	3.2152	4.2126	5.1338	5.4789	4.2854	3.4973	2.7920	2.4973	1.9644	1.6873	1.2462	1.2544	1.5180
Liu's [38]	4.4842	5.7786	6.5592	7.5441	7.6273	8.2756	8.3487	8.9660	9.2095	9.7438	9.6947	10.1166	10.2891
Chen's [41]	2.2236	2.9872	3.5218	3.9496	4.3597	4.6666	4.9825	5.2591	5.5143	5.7275	5.9560	6.1597	6.3206
Wu's [47]	3.9273	4.6659	5.3400	5.8691	6.2594	6.6285	7.0772	7.4021	7.5949	7.8164	8.0016	8.2149	8.4829
nD-PCS	4.6424	5.8886	6.8560	7.6556	8.3273	8.9251	9.4375	9.9005	10.3185	10.6983	11.0491	11.3727	11.6746
$H_{\max}$	4.7549	6.0000	6.9658	7.7549	8.4221	9.0000	9.5098	9.9658	10.3783	10.7549	11.1013	11.4221	11.7207

TABLE VI

THE MAXIMUM IEs OF THE nD CHAOTIC MAPS GENERATED BY DIFFERENT METHODS WITH DIMENSION $n = \{4, 5, \cdots, 12\}$ BY FIXING $I = 3$

n	**nD chaotic map Generation Methods**					
	Shen's [39]	Liu's [38]	Chen's [41]	Wu's [47]	nD-PCS	$H_{\max}$
4	4.1705	6.0864	1.7290	5.0873	6.1608	6.3399
5	4.4563	7.6503	1.3479	6.4362	7.7183	7.9248
6	3.2189	9.1772	2.0628	7.8038	9.2759	9.5098
7	2.6805	10.7148	2.2942	9.1394	10.8516	11.0947
8	1.5354	12.3587	2.1756	10.2567	12.4252	12.6797
9	0.6026	13.8415	1.5461	11.9150	14.0070	14.2647
10	0.2393	15.4980	1.9498	12.9367	15.5899	15.8496
11	0.1061	17.0571	1.6220	14.1957	17.1734	17.4346
12	0.5207	18.5959	2.5147	15.7751	18.7576	19.0196

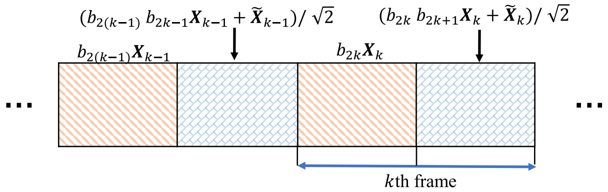
fixing the number of intervals $I = 3$. For each chaotic map generation method with different n, 100 nD chaotic maps are randomly generated. For each chaotic map, the initial state is set to $\mathbf{x}(0) \in [0, 1]$, and a number of 3^{n+1} output states are generated to calculate its IE. Fig. 7(b) displays the mean IEs of the 100 nD chaotic maps produced by different generation methods. This shows that the nD chaotic maps generated by the nD-PCS can achieve higher mean IEs than the nD chaotic maps generated by other methods. Table VI lists the maximum IEs of the 100 nD chaotic maps. As can be seen, the maximum IEs of the proposed method are significantly larger than those of other generation methods and are very close to the theoretical maximum values. These results demonstrate that the proposed nD-PCS can generate high-dimensional chaotic maps with uniform distribution outputs.

VI. SECURE COMMUNICATION APPLICATIONS

Differential chaos shift keying (DCSK) is a widely used technique in secure communication [13]. It exhibits a strong ability to resist noise interference. When a chaotic system is used to transmit data, its trajectory distribution can significantly affect the performance of resisting bit errors. Because the high-dimensional chaotic maps generated using the nD-PCS can produce more uniformly distributed chaotic outputs than existing chaotic maps, they can show better performance in resisting channel noise. In this section, we first propose a simple new DCSK and then test the performance of different chaotic maps in this DCSK.

A. Structure of the Proposed DCSK

Fig. 8 illustrates the transmitted signal of the proposed DCSK. It shows that each frame contains two sections. The previous section of the kth frame is the multiplication of the information bit b_{2k} with an M-length chaotic sequence $\mathbf{X}_k$, where M is a spreading factor in the DCSK. The latter section is formed by the addition of two parts. The first part is the multiplication of the information bits b_{2k}, b_{2k+1} and chaotic sequence $\mathbf{X}_k$, while the second part is a time-reversed chaotic sequence $\widetilde{\mathbf{X}}_k$. The $\widetilde{\mathbf{X}}_k$ is generated by flipping over the chaotic sequence $\mathbf{X}_k$ and can be represented as



$(b_{2(k-1)}\, b_{2k-1}\boldsymbol{X}_{k-1} + \widetilde{\boldsymbol{X}}_{k-1})/\sqrt{2}$
$(b_{2k}\, b_{2k+1}\boldsymbol{X}_k + \widetilde{\boldsymbol{X}}_k)/\sqrt{2}$
$b_{2(k-1)}\boldsymbol{X}_{k-1}$
$b_{2k}\boldsymbol{X}_k$
…
…
kth frame

Fig. 8. The transmitted signal format in the proposed DCSK.

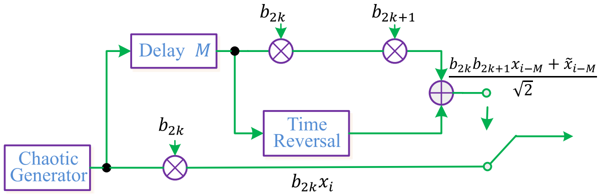


Fig. 9. The transmitter of the proposed DCSK.

$$\widetilde{\mathbf{X}}_k(i) = \mathbf{X}_k(M + 1 - i), \quad 0 < i \le M. \tag{30}$$

The proposed DCSK is mainly composed of the transmitter and receiver parts, and its structures are described as follows:

1) Transmitter: The transmitter of the proposed DCSK is illustrated in Fig. 9. It can be observed that the M-length chaotic sequence $\mathbf{X}_k = \{x_{2kM+1}, x_{2kM+2}, \cdots, x_{(2k+1)M}\}$ is generated by a chaotic generator. In one branch, it directly multiplies to information b_{2k}. In the other branch, $\mathbf{X}_k$ first goes through an M time delay to generate a delayed chaotic sequence $\mathbf{X}_{k-M}$, which is multiplied to the information bits b_{2k} and b_{2k+1}. Then, the multiplication result is added to the time-reversed chaotic sequence $\widetilde{\mathbf{X}}_{k-M}$. The generation of the k-th frame signal e_i can be expressed as

$$e_i = \begin{cases} b_{2k}x_i, \quad 2kM < i \le (2k+1)M; \\ (b_{2k}b_{2k+1}x_{i-M} + \tilde{x}_{i-M})/\sqrt{2}, \\ \qquad\qquad (2k+1)M < i \le 2(k+1)M. \end{cases}$$

2) Receiver: The receiver of the proposed DCSK is shown in Fig. 10. Because many transmission channels are noise channels, the transition signal can be blurred by noise during

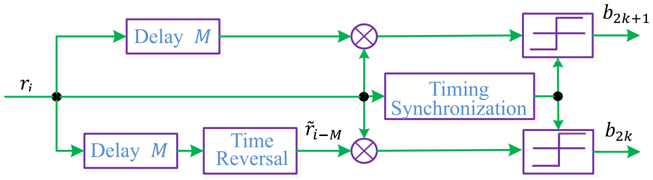


Fig. 10. The receiver of the proposed DCSK.

transmission. Thus, the received signal r_i is different from the transmitted signal e_i and can be represented as $r_i = e_i + \eta_i$, where η_i denotes the additive noise. For the k-th frame, the information bits b_{2k} and b_{2k+1} can be concurrently demodulated by the top and bottom branches of the receiver, respectively.

To extract the information bit b_{2k}, the receiver signal r_i is multiplied by $\tilde{r}_{i-M}$, where $\tilde{r}_{i-M}$ is the time-reversed signal of r_{i-M} and can be obtained using the same rule in Eq. (30). Then, the correlator output for bit b_{2k} can be calculated as

$$\begin{aligned}
Z_{2k} &= \sum_{i=(2k+1)M+1}^{2(k+1)M} r_i\tilde{r}_{i-M}\\
&= \frac{\sqrt{2}}{2}\sum_{i=(2k+1)M+1}^{2(k+1)M}(b_{2k}b_{2k+1}x_{i-M}+\tilde{x}_{i-M}+\eta_i)\\
&\quad\times(b_{2k}\tilde{x}_{i-M}+\tilde{\eta}_{i-M})\\
&= \frac{\sqrt{2}}{2}\sum_{i=(2k+1)M+1}^{2(k+1)M} b_{2k}x_{i-M}^2\\
&\quad+\frac{\sqrt{2}}{2}\sum_{i=(2k+1)M+1}^{2(k+1)M}(b_{2k+1}x_{i-M}\tilde{x}_{i-M}+\tilde{x}_{i-M}\tilde{\eta}_{i-M}\\
&\quad+b_{2k}b_{2k+1}x_{i-M}\tilde{\eta}_{i-M}+\eta_i\tilde{\eta}_{i-M}+b_{2k}\tilde{x}_{i-M}\eta_i). \quad (31)
\end{aligned}$$

To extract the information bit b_{2k+1}, the receiver signal r_i is multiplied by its delayed signal r_{i-M}. Then, the output of the correlator for bit b_{2k+1} can be calculated as

$$\begin{aligned}
Z_{2k+1} &= \sum_{i=(2k+1)M+1}^{2(k+1)M} r_ir_{i-M}\\
&= \frac{\sqrt{2}}{2}\sum_{i=(2k+1)M+1}^{2(k+1)M}(b_{2k}b_{2k+1}x_{i-M}+\tilde{x}_{i-M}+\eta_i)\\
&\quad\times(b_{2k}x_{i-M}+\eta_{i-M})\\
&= \frac{\sqrt{2}}{2}\sum_{i=(2k+1)M+1}^{2(k+1)M} b_{2k+1}x_{i-M}^2\\
&\quad+\frac{\sqrt{2}}{2}\sum_{i=(2k)M+1}^{2(k+1)M}(b_{2k+1}x_{i-M}\tilde{x}_{i-M}+\tilde{x}_{i-M}\eta_{i-M}\\
&\quad+b_{2k}b_{2k+1}x_{i-M}\eta_{i-M}+\eta_i\eta_{i-M}+b_{2k}x_{i-M}\eta_i). \quad (32)
\end{aligned}$$

The first items in Eqs. (31) and (32) are the main components for extracting the transmitted information bits, whereas the remaining items are interference components containing noise. Despite noise, the information bits can be extracted by

$$b_n = \begin{cases} 1, & \text{if } Z_n > 0,\\ -1, & \text{else.}\end{cases}$$

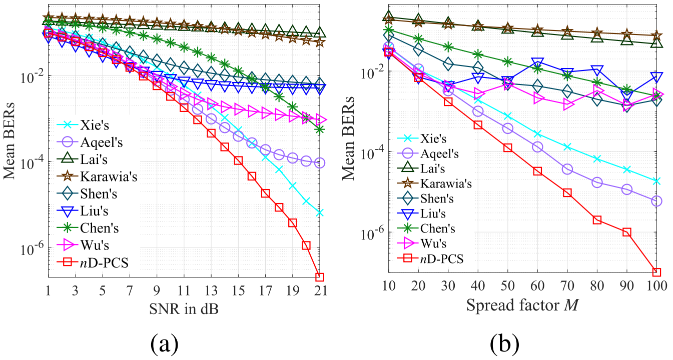


Fig. 11. The mean BERs of the proposed DCSK using different 3D chaotic maps under (a) spread factor $M = 40$ and $SNR \in \{1, 2, \cdots, 21\}$, and (b) noise level $SNR = 13$ dB and spread factors $M \in \{10, 20, \cdots, 100\}$.

B. Simulation Results

Here, we test the ability of the proposed DCSK to tolerate noise. Furthermore, we consider that the chaotic generator is a 3D chaotic map generated using the chaotic map generation methods and the other 3D chaotic maps. The parameters of the chaotic maps were set to be the same as those in Section V. We add the Gaussian noise in our simulations because it is the most widely occurring noise during transmission.

Two groups of experiments were developed to calculate the bit error rates (BERs) of the proposed DCSK for different 3D chaotic maps. The first group analyzes the BERs against various noise levels by fixing the spread factor $M = 40$. For each 3D chaotic map, the BERs of the proposed DCSK were calculated under different signal-noise-ratios (SNRs) $\{1, 2, \cdots, 21\}$. To obtain a stable BER, we repeated each experiment 100 times with different system parameters to calculate the mean BER value. Fig. 11(a) plots the mean BERs of the proposed DCSK with different 3D chaotic maps under a fixed spread factor $M = 40$ and $SNR \in \{1, 2, \cdots, 21\}$. It is clear that when using the 3D chaotic map generated using the proposed *n*D-PCS, the DCSK can achieve much smaller BERs than using other 3D chaotic maps.

The other group of experiments investigated the BERs against various spread factors M by setting the noise strength as $SNR = 13$ dB. The system parameters of different chaotic maps were set to be the same as those in the first group of experiments. Fig. 11(b) shows the mean BERs in 100 experiments for different 3D chaotic maps at $SNR = 13$ and spread factor $M \in \{10, 20, \cdots, 100\}$. The results show that the 3D chaotic maps generated by the *n*D-PCS can always achieve smaller BERs than the other 3D chaotic maps. Furthermore, the chaotic maps generated by the *n*D-PCS exhibit a strong ability to resist transmission noise in secure communication applications, compared with existing maps.

VII. Conclusion

This study first analyzed the relationship between the parameter matrix of a high-dimensional chaotic map and its LEs, and then proposed an *n*-dimensional polynomial chaotic system (*n*D-PCS) that can generate *n*D chaotic maps with desired LEs. The *n*D-PCS is constructed from n parametric polynomials, and its parameter matrix is constructed using matrix expansion. Theoretical analysis shows that the n LEs of the nD-PCS are determined by its partial control parameters. Thus, by customizing these parameters, one can obtain high-dimensional chaotic maps with any desired LEs and further exhibit the expected dynamic properties. To demonstrate the effects of the nD-PCS, we provided two examples of high-dimensional chaotic maps: a 4D polynomial chaotic map and a 7D polynomial chaotic map; we then implemented them in a microcontroller-based hardware platform and tested their randomness properties using TestU01. Property analysis shows that these two high-dimensional chaotic maps have the desired LEs and thus exhibit complex dynamics and hyperchaotic behaviors. The performance analysis shows that the high-dimensional chaotic maps generated by the nD-PCS have the desired LEs, more complex dynamic properties, and better distribution outputs, compared with other high-dimensional chaotic maps. Finally, we developed a secure communication scheme, and the simulation results demonstrate that the chaotic maps generated by the proposed nD-PCS exhibit much better performance in this application than other high-dimensional chaotic maps. Our future work will investigate the hyperchaotic behaviors in fractional order, because the fractional-order chaotic systems have many unique and different dynamic behaviors compared with integer-order chaotic systems.

References

[1] H. G. Schuster and W. Just, *Deterministic Chaos: An Introduction*. Hoboken, NJ, USA: Wiley, 2006.

[2] S. Vaidyanathan and C. Volos, *Advances and Applications in Chaotic Systems*. Berlin, Germany: Springer, 2016.

[3] C. Li, K. Tan, B. Feng, and J. Lu, "The graph structure of the generalized discrete Arnold's cat map," *IEEE Trans. Comput.*, early access, Jan. 13, 2021, doi: 10.1109/TC.2021.3051387.

[4] C. Wang, H. Xia, and L. Zhou, "A memristive hyperchaotic multiscroll jerk system with controllable scroll numbers," *Int. J. Bifurcation Chaos*, vol. 27, no. 6, 2017, Art. no. 1750091.

[5] A. Pikovsky and A. Politi, *Lyapunov Exponents: A Tool to Explore Complex Dynamics*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[6] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boulder, CO, USA: Westview Press, 2014.

[7] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Boulder, CO, USA: Westview Press, 2003.

[8] K. Tian, C. Grebogi, and H.-P. Ren, "Chaos generation with impulse control: Application to non-chaotic systems and circuit design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 7, pp. 3012–3022, Jul. 2021.

[9] Y. Luo, W. Wang, S. Best, Y. Wang, and X. Xu, "A high-performance and secure TRNG based on chaotic cellular automata topology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4970–4983, Dec. 2020.

[10] W. He, T. Luo, Y. Tang, W. Du, Y.-C. Tian, and F. Qian, "Secure communication based on quantized synchronization of chaotic neural networks under an event-triggered strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3334–3345, Sep. 2020.

[11] C. Pan, Q. Hong, and X. Wang, "A novel memristive chaotic neuron circuit and its application in chaotic neural networks for associative memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 3, pp. 521–532, Mar. 2021.

[12] C. Bai, H. P. Ren, and G. Kolumban, "Double-sub-stream M-ary differential chaos shift keying wireless communication system using chaotic shape-forming filter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3574–3587, Oct. 2020.

[13] G. Kaddoum, E. Soujeri, C. Arcila, and K. Eshteiwi, "I-DCSK: An improved noncoherent communication system architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 9, pp. 901–905, Sep. 2015.

[14] M. A. Platas-Garza, E. Zambrano-Serrano, J. R. Rodríguez-Cruz, and C. Posadas-Castillo, "Implementation of an encrypted-compressed image wireless transmission scheme based on chaotic fractional-order systems," *Chin. J. Phys.*, vol. 71, pp. 22–37, Jun. 2021.

[15] L. Zhang, Z. Chen, W. Rao, and Z. Wu, "Efficient and secure non-coherent OFDM-based overlapped chaotic chip position shift keying system: Design and performance analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 1, pp. 309–321, Jan. 2020.

[16] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm," *Opt. Lasers Eng.*, vol. 128, May 2020, Art. no. 105995.

[17] J. Zheng, H. Hu, and X. Xia, "Applications of symbolic dynamics in counteracting the dynamical degradation of digital chaos," *Nonlinear Dyn.*, vol. 94, no. 2, pp. 1535–1546, 2018.

[18] Z. Hua, B. Zhou, and Y. Zhou, "Sine chaotification model for enhancing chaos and its hardware implementation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1273–1284, Feb. 2019.

[19] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2322–2335, Jun. 2019.

[20] C. Fan, Q. Ding, and C. K. Tse, "Counteracting the dynamical degradation of digital chaos by applying stochastic jump of chaotic orbits," *Int. J. Bifurcation Chaos*, vol. 29, no. 8, Jul. 2019, Art. no. 1930023.

[21] Z. Xiao, S. Shan, and L. Cheng, "Identification of cascade dynamic nonlinear systems: A bargaining-game-theory-based approach," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4657–4669, Sep. 2018.

[22] R. Zhang, D. Zeng, J. H. Park, Y. Liu, and S. Zhong, "A new approach to stabilization of chaotic systems with nonfragile fuzzy proportional retarded sampled-data control," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3218–3229, Sep. 2019.

[23] M. Xu, G. Chen, and Y.-T. Tian, "Identifying chaotic systems using Wiener and Hammerstein cascade models," *Math. Comput. Model.*, vol. 33, nos. 4–5, pp. 483–493, Feb. 2001.

[24] J.-P. Yeh, "Identifying chaotic systems using a fuzzy model coupled with a linear plant," *Chaos, Solitons Fractals*, vol. 32, no. 3, pp. 1178–1187, May 2007.

[25] M. Han, K. Zhong, T. Qiu, and B. Han, "Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2720–2731, Jul. 2019.

[26] M. Xu, M. Han, T. Qiu, and H. Lin, "Hybrid regularized echo state network for multivariate chaotic time series prediction," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2305–2315, Jun. 2019.

[27] J. A. Lazzús, M. Rivera, and C. H. López-Caraballo, "Parameter estimation of Lorenz chaotic system using a hybrid swarm intelligence algorithm," *Phys. Lett. A*, vol. 380, nos. 11–12, pp. 1164–1171, Mar. 2016.

[28] L. Lin, M. Shen, H. C. So, and C. Chang, "Convergence analysis for initial condition estimation in coupled map lattice systems," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4426–4432, Aug. 2012.

[29] S. Ergün, "On the security of chaos based true random number generators," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 99, no. 1, pp. 363–369, 2016.

[30] Y. Deng, H. Hu, W. Xiong, N. N. Xiong, and L. Liu, "Analysis and design of digital chaotic systems with desirable performance via feedback control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 8, pp. 1187–1200, Aug. 2015.

[31] L. Liu, J. Lin, S. Miao, and B. Liu, "A double perturbation method for reducing dynamical degradation of the digital baker map," *Int. J. Bifurcation Chaos*, vol. 27, no. 7, Jun. 2017, Art. no. 1750103.

[32] Z. Hua and Y. Zhou, "One-dimensional nonlinear model for producing chaos," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 1, pp. 235–246, Jan. 2018.

[33] C. Cao, K. Sun, and W. Liu, "A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map," *Signal Process.*, vol. 143, pp. 122–133, Feb. 2018.

[34] S. Panahi, J. C. Sprott, and S. Jafari, "Two simplest quadratic chaotic maps without equilibrium," *Int. J. Bifurcation Chaos*, vol. 28, no. 12, Nov. 2018, Art. no. 1850144.

[35] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," *Int. J. Bifurcation Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005.

[36] C. Shen, S. Yu, J. Lü, and G. Chen, "Constructing hyperchaotic systems at will," *Int. J. Circuit Theory Appl.*, vol. 43, no. 12, pp. 2039–2056, 2015.

[37] Y. Chen and Q. Yang, "A new Lorenz-type hyperchaotic system with a curve of equilibria," *Math. Comput. Simul.*, vol. 112, pp. 40–55, Jun. 2015.

[38] W. Liu, K. Sun, and S. He, "SF-SIMM high-dimensional hyperchaotic map and its performance analysis," *Nonlinear Dyn.*, vol. 89, no. 4, pp. 2521–2532, 2017.

[39] C. Shen, S. Yu, J. Lü, and G. Chen, "Designing hyperchaotic systems with any desired number of positive Lyapunov exponents via a simple model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2380–2389, Aug. 2014.

[40] Y. Wu, Z. Hua, and Y. Zhou, "*n*-dimensional discrete cat map generation using Laplace expansions," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2622–2633, Nov. 2016.

[41] S. Chen, S. Yu, J. Lü, G. Chen, and J. He, "Design and FPGA-based realization of a chaotic secure video communication system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2359–2371, Sep. 2018.

[42] T. S. Shores, *Applied Linear Algebra and Matrix Analysis*. New York, NY, USA: Springer, 2007.

[43] P. Glendinning, "Robust chaos revisited," *Eur. Phys. J. Special Topics*, vol. 226, no. 9, pp. 1721–1738, Jun. 2017.

[44] U. Schwengelbeck and F. H. M. Faisal, "Definition of Lyapunov exponents and KS entropy in quantum dynamics," *Phys. Lett. A*, vol. 199, nos. 5–6, pp. 281–286, Apr. 1995.

[45] P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 22-1–22-40, 2007.

[46] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 278, no. 6, pp. 2039–2049, 2000.

[47] Q. Wu, F. Zhang, Q. Hong, X. Wang, and Z. Zeng, "Research on cascading high-dimensional isomorphic chaotic maps," *Cogn. Neurodyn.*, vol. 15, no. 1, pp. 157–167, Feb. 2021.

[48] Y. Xie, J. Yu, S. Guo, Q. Ding, and E. Wang, "Image encryption scheme with compressed sensing based on new three-dimensional chaotic system," *Entropy*, vol. 21, no. 9, p. 819, Aug. 2019.

[49] M. Aqeel and S. Ahmad, "Analytical and numerical study of Hopf bifurcation scenario for a three-dimensional chaotic system," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 755–765, Apr. 2016.

[50] Q. Lai, P. D. Kamdem Kuate, F. Liu, and H. H.-C. Iu, "An extremely simple chaotic system with infinitely many coexisting attractors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 6, pp. 1129–1133, Jun. 2020.

[51] A. Karawia, "Image encryption based on Fisher–Yates shuffling and three dimensional chaotic economic map," *IET Image Process.*, vol. 13, no. 12, pp. 2086–2097, Oct. 2019.

Zhongyun Hua (Member, IEEE) received the B.S. degree in software engineering from Chongqing University, Chongqing, China, in 2011, and the M.S. and Ph.D. degrees in software engineering from the University of Macau, Macau, China, in 2013 and 2016, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include chaotic systems, chaos-based applications, and multimedia security.

Yinxing Zhang received the M.S. degree in fundamental mathematics from Guilin University of Electronic Technology, Guilin, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His current research interests include chaotic systems and nonlinear system control.

Han Bao (Member, IEEE) received the B.S. degree in landscape design from Jiangxi University of Finance and Economics, Nanchang, China, in 2015, and the M.S. degree in art and design from Changzhou University, Changzhou, China, in 2018. He is currently pursuing the Ph.D. degree in nonlinear system analysis and measurement technology with the Nanjing University of Aeronautics and Astronautics, Nanjing, China, and is undergoing a one-year joint training at Changzhou University.

In 2019, he visited the Department of Computer Science, The University of Auckland, New Zealand. His research interests include memristive neuromorphic circuit, nonlinear circuits and systems, and artificial intelligence.

Hejiao Huang graduated from the City University of Hong Kong. She received the Ph.D. degree in computer science in 2004.

She was an Invited Professor with INRIA, Bordeaux, France. She is currently a Professor with Harbin Institute of Technology, Shenzhen, China. Her research interests include cloud computing, network security, trustworthy computing, and formal methods for system design and wireless networks.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

He is currently a Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau. His research interests include chaotic systems, multimedia security, computer vision, and machine learning.

Dr. Zhou is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macau Natural Science Award in 2014. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and several other journals.