We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

6,600
Open access books available

177,000
International authors and editors

195M
Downloads

154
Countries delivered to

TOP 1%
Our authors are among the most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
Abstract

In recent years, exploring and investigating chaotic systems with hyperbolic sine nonlinearity has gained the interest of many researchers. With two back-to-back diodes to approximate the hyperbolic sine nonlinearity, these chaotic systems can achieve simplicity of the electrical circuit without any multiplier or sub-circuits. In this chapter, the genesis of chaotic systems with hyperbolic sine nonlinearity is introduced, followed by the general method of generating nth-order (n > 3) chaotic systems. Then some derived chaotic systems/torus-chaotic system with hyperbolic sine nonlinearity is discussed. Finally, the applications such as random number generator algorithm, spread spectrum communication and image encryption schemes are introduced. The contribution of this chapter is that it systematically summarizes the design methods, the dynamic behavior and typical engineering applications of chaotic systems with hyperbolic sine nonlinearity, which may widen the current knowledge of chaos theory and engineering applications based on chaotic systems.

Keywords: chaotic systems, torus chaos, hyperbolic sine nonlinearity, spread spectrum communication, image encryption

1. Introduction

Since Lorenz discovered chaos in a third-order ordinary differential equations, a new field of science has been launched [1]. The fact that simple equations can exhibit incredible complex behavior continues enthral engineers to apply chaotic systems to cryptosystem, secure communication, spread spectrum communication, etc. [2].

There is no doubt that nonlinear term is very important to design chaotic systems, which has peculiar complex properties such as ergodicity, highly initial value sensitivity, non-periodicity and long-term unpredictability. According to the literature, the nonlinearities can be piecewise nonlinear function [3], trigonometric function [4], absolute value function [5], or power function [6]. With different nonlinearities, the chaotic system can have various strange attractors as single-scroll [7], double-scroll [8], multi-scroll [9], etc. The majority of such chaotic systems are known for many years, and some chaotic systems with hidden attractors are derived from them [10–12].

In recent years, chaotic systems with hyperbolic sine nonlinearities have gained the interest of many researchers. With two back-to-back diodes to approximate the
hyperbolic sine nonlinearity, these chaotic systems can achieve simplicity of the electrical circuit without any multiplier or sub-circuits. Compared to single-scroll chaotic systems, the chaotic system with hyperbolic sine nonlinearity has richer dynamic behavior because it is symmetrical and can exhibit symmetry breaking, and offers the possibility that attractors will split or merge as some bifurcation parameter is changed [13].

In this chapter, we will systematically summarize the design method, the dynamic behavior and typical engineering applications of chaotic systems with hyperbolic sine nonlinearity. The genesis and general method of generating nth-order \((n > 3)\) chaotic systems with hyperbolic sine nonlinearity are introduced in Section II. Some derived chaotic systems/torus-chaotic system with hyperbolic sine nonlinearity is discussed in Section III. The application such as random number generator algorithm, spread spectrum communication and image encryption schemes are introduced in Section IV. Conclusions are finally drawn in Section V.

2. General chaotic systems with hyperbolic sine nonlinearity

2.1 The genesis of chaotic systems with hyperbolic sine nonlinearity

In 2011, Sprott and Munmuangsaen proposed an exponential chaotic system [14], which happens to be an example of the simplest chaotic system [15]. In the same year, Sprott used common resistors, capacitors, operational amplifiers, and a diode to successfully implement this system in a circuit [16]. Few years later, the simplest hyperbolic sine chaotic system is proposed [17]. Compared to the exponential chaotic system, the hyperbolic sine chaotic system changed the nonlinearity from exponential function (asymmetric function) to hyperbolic sine function (symmetric function), which can exhibit symmetry breaking, and offers the possibility that attractors will split or merge as some bifurcation parameter is changed [18].

The simplest chaotic system with a hyperbolic sine is described as follows:

\[
\ddot{x} + c\dot{x} + x + \rho \sinh (\varphi x) = 0
\]  

(1)
Where $c$ is considered as the bifurcation parameter, 
\[
\sinh (\varphi x) = \frac{e^{\varphi x} - e^{-\varphi x}}{2},
\]
\[
\rho = 1.2 \times 10^{-6} \quad \text{and} \quad \varphi = \frac{1}{0.020},
\]
which have been chosen to facilitate circuit implementation using diodes. The corresponding circuit schematic diagram of Eq. (1) is shown as Figure 1.

When $c = 0.75$, the Eq. (1) can exhibit chaotic behavior, which is shown as Figure 2.

2.2 The general equations of generating chaotic systems with hyperbolic sine nonlinearity

It is obvious that Eq. (1) can be written in the form with jerk equations:

\[
\begin{align*}
  \dot{x}_1 &= x_2 \\
  \dot{x}_2 &= x_3 \\
  \dot{x}_3 &= -c x_3 - f(x_2) - x_1
\end{align*}
\] (2)

where $f(x_2) = \rho * \sinh (\varphi x_2)$. Therefore, the higher order chaotic systems with hyperbolic sine nonlinearity can be generated by adding jerk cabins, which is described by:

\[
\begin{align*}
  \dot{x}_1 &= x_2 - x_1 \\
  \dot{x}_2 &= x_3 - x_2 \\
  & \vdots \\
  \dot{x}_{n-3} &= x_{n-2} - x_{n-3} \\
  \dot{x}_{n-2} &= x_{n-1} \\
  \dot{x}_{n-1} &= x_n \\
  \dot{x}_n &= -c x_n - f(x_{n-1}) - n x_{n-2} - n x_{n-3} - \cdots - \frac{1}{2n} x_1
\end{align*}
\] (3)

where $\dot{x}_{k-1} = x_k - x_{k-1}$ is the jerk cabin. With Eq. (3), we can construct nth-order ($n > 3$) chaotic systems with hyperbolic sine nonlinearity.

When $n = 4$, the equations of fourth-order chaotic systems will be:
\[
\begin{align*}
\dot{x}_1 &= x_2 - x_1 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -x_4 - f(x_3) - 5x_2 - 0.125x_1
\end{align*}
\] (4)

Figure 3. The corresponding circuit schematic diagram of Eq. (4).

Figure 4. Numerical and actual circuit state space plot in \( x_2 - x_3 \) plane and \( x_3 - x_4 \) plane.
The corresponding circuit schematic diagram of Eq. (4) is shown as Figure 3.

Its numerical and actual circuit state space plot is shown as Figure 4.

When n = 5, the equations of fifth-order chaotic systems will be:
The corresponding circuit schematic diagram of Eq. (5) is shown as Figure 5. Its numerical and actual circuit state space plot is shown as Figure 6.

3. Derived chaotic systems/torus-chaotic system with hyperbolic sine nonlinearity

3.1 Multi-nonlinearities hyperbolic sine chaotic system

One way to construct the derived chaotic systems is to add more nonlinear terms of the equations. For example, the new chaotic system can be constructed by Eq. (4), which is described as follows:

\[
\begin{align*}
\dot{x}_1 &= x_2 - x_1 \\
\dot{x}_2 &= x_3 - x_2 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= x_5 \\
\dot{x}_5 &= -x_5 - f(x_4) - 5x_3 - 5x_2 - 0.1x_1
\end{align*}
\] (5)

Where \( \rho = 1.2 \times 10^{-6}, \varphi = \frac{1}{0.026} \). These equations can exhibit chaotic behavior as shown in Figure 7.

Figure 7.
Numerical phase space plot of Eq. (6).
3.2 Simple chaotic system with hyperbolic sine nonlinearity

The other way to construct the derived chaotic systems is to simplify the known chaotic systems. For example, if we remove the parameter \( \rho \) and \( \phi \), search the parameter space, we will have the following chaotic system:

\[
\begin{aligned}
\dot{x}_1 &= 6x_2 - x_1 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -x_4 - \sinh (x_3) - x_1
\end{aligned}
\]  

(7)

When initial conditions are set to be \((x_1, x_2, x_3, x_4) = (0.7, 0.9, 1.0, 1.3)\), or \((x_1, x_2, x_3, x_4) = (-0.7, -0.9, -1.0, -1.3)\), the system exhibits period behavior. When the initial conditions are set to be \((x_1, x_2, x_3, x_4) = (7, 9, 10, 13)\) and \((x_1, x_2, x_3, x_4) = (-7, -9, -10, -13)\), the system exhibits chaotic behavior. Therefore, this system has four coexistence attractors [19], as shown in Figure 8.

3.3 Torus-chaotic system with hyperbolic sine nonlinearity

By introducing a nonlinear feedback controller to system Eq. (5), the following system is obtained:

\[
\begin{aligned}
\dot{x}_1 &= x_2 - \rho \sinh (\rho x_3) \\
\dot{x}_2 &= x_3 - x_2 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= x_5 \\
\dot{x}_5 &= -cx_5 - \rho \sinh (\rho x_4) - 5x_3 - 5x_2 - 0.1x_1
\end{aligned}
\]  

(8)

When \( c = 1 \), the Lyapunov exponents are \((\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (0.47, 0.0, -1.10, -1.37)\), which suggests Eq. (8) is exhibiting torus-chaos behavior [20].

When \( c = 1.55 \) and the initial conditions are set to be \((x_1, x_2, x_3, x_4, x_5) = (-0.1, -0.1, -0.1, -0.1, -0.1)\) and \((x_1, x_2, x_3, x_4, x_5) = (0.1, 0.1, 0.1, 0.1, 0.1)\), the system has two coexisting attractors as shown in Figure 9.

Figure 10 shows the Lyapunov exponent spectrum, Kaplan–Yorke dimension spectrum and bifurcations of Eq. (8) as the coefficient \( c \) is varied over the range \( c \in [0.3, 2] \). Those figures suggest there is an interesting route leading to chaos [21].

1. When \( c \in [0.3, 0.4639] \), there exists a period-doubling behavior along with \( \dot{x}_2 \) and \( \dot{x}_3 \) subspace. However, the system shows torus behavior along with \( \dot{x}_2 \) and \( \dot{x}_3 \) subspace. It is like saddle point: the system is stable in one direction but unstable in the other direction.

2. When \( c \in [0.4640, 0.5574] \), the system exhibits two-torus-chaos behavior except for some 2-torus windows. When the parameter passed \( c = 0.4639 \) to \( c = 0.4640 \), two-torus-chaos is born by replacing the 2-torus behavior. The Lyapunov exponents at these two critical values are \((\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (0, 0, -0.01, -0.57, -0.88)\) for \( c = 0.4639 \) and \((\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (0.02, 0, -0.60, -0.88)\) for \( c = 0.4640 \). This may cause by the period-doubling route along with \( \dot{x}_2 \) and \( \dot{x}_3 \) subspace.
Figure 8.
Coexistence attractors of Eq. (7).

Figure 9.
Coexistence attractors of Eq. (8).
3. When \( c \in [0.5575, 0.5901] \), the system exhibits 2-torus behavior.

4. When \( c \in [0.5902, 1.5575] \), the system exhibits 2-torus-chaos behavior except for 2-torus windows. The route leading to chaos is same to point 3.

5. When \( c \in [1.5575, 2] \) the system exhibits 2-torus behavior, except for some 3-torus windows like \( c = 1.6157 \).
4. Engineering applications with chaotic systems with hyperbolic sine nonlinearity

4.1 Random number generator

Sensitivity to initial conditions is one of the most important properties of chaotic systems. Therefore, chaotic systems are very suitable for the cryptography purpose. But before that, it should be noticed that the probability density distributions (PDD) of chaotic systems are not uniform distributions. Figure 11(a) and 11(b) are the waveform and PDD of $x_4$ of Eq. (4). It shows that PDD of the output sequences has physical characteristic. The cryptosystem with these sequences cannot resist side channel attack.

To remove physical characteristic, one can use the following de-correlation operation:

$$ S_{out} = S_{in} \times 10^6 - \text{floor}(S_{in} \times 10^6) $$

(9)

In fact, Eq. (9) can be applied in all chaotic/torus-chaotic/hyperchaotic systems. The output sequences can pass fifteen random tests of NIST 800-22, as shown in Table 1, which indicated the proposed method can provide high security level. This proposed method can be used as a part of some cyber security systems such as the verification code, secure QR code and some secure communication protocols.

Figure 11.
Waveform and PDD before and after de-correlation operation of $x_4$ of Eq. (4): (a) is the waveform of $x_4$ before de-correlation operation; (b) is the PDD of $x_4$ before de-correlation operation; (c) is the waveform of $x_4$ after de-correlation operation; (d) is the PDD of $x_4$ after de-correlation operation.
4.2 Image encryption

Image encryption is another widely used engineering application of chaotic system. In this section, we will use Eq. (7) for image encryption purpose.

A flowchart of the encryption scheme is shown in Figure 12.

The detailed encryption process includes the following steps.

Input: Plain image; Initial conditions for the chaotic system; Control parameters of the chaotic system.

Output: Ciphered image.

Step 1: Calculate the average pixel value of the plain image and generate the pseudorandom sequence.

Step 2: Transform the pseudorandom sequence and change pixel value of the image via XOR.

Step 3: Sort the pseudorandom sequence for permutation.

Step 4: Shift the pixel positions by column using the sorted elements.

Step 5: Shift the pixel positions by row using the sorted elements.

To provide a better understanding of this scheme, the pseudocode is provided in Table 2.

<table>
<thead>
<tr>
<th>Test</th>
<th>P-value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.841481</td>
<td>Success</td>
</tr>
<tr>
<td>Block frequency</td>
<td>0.900704</td>
<td>Success</td>
</tr>
<tr>
<td>Runs</td>
<td>0.744455</td>
<td>Success</td>
</tr>
<tr>
<td>Longest run</td>
<td>0.172897</td>
<td>Success</td>
</tr>
<tr>
<td>Rank</td>
<td>0.368065</td>
<td>Success</td>
</tr>
<tr>
<td>FFT</td>
<td>0.762020</td>
<td>Success</td>
</tr>
<tr>
<td>Non-overlapping template</td>
<td>0.813121</td>
<td>Success</td>
</tr>
<tr>
<td>Overlapping template</td>
<td>0.532736</td>
<td>Success</td>
</tr>
<tr>
<td>Universal</td>
<td>0.856573</td>
<td>Success</td>
</tr>
<tr>
<td>Linear complexity</td>
<td>0.408679</td>
<td>Success</td>
</tr>
<tr>
<td>Serial</td>
<td>0.967366</td>
<td>Success</td>
</tr>
<tr>
<td>Approximate entropy</td>
<td>0.433157</td>
<td>Success</td>
</tr>
<tr>
<td>Cumulative sums</td>
<td>0.688582</td>
<td>Success</td>
</tr>
<tr>
<td>Random excursions</td>
<td>0.075229</td>
<td>Success</td>
</tr>
<tr>
<td>Random excursions variant</td>
<td>0.102049</td>
<td>Success</td>
</tr>
</tbody>
</table>

Table 1. Pseudo-random properties of $x_i$ of Eq. (8) after de-correlation operation.

4.2 Image encryption

Image encryption is another widely used engineering application of chaotic system. In this section, we will use Eq. (7) for image encryption purpose.

A flowchart of the encryption scheme is shown in Figure 12.

The detailed encryption process includes the following steps.

Input: Plain image; Initial conditions for the chaotic system; Control parameters of the chaotic system.

Output: Ciphered image.

Step 1: Calculate the average pixel value of the plain image and generate the pseudorandom sequence.

Step 2: Transform the pseudorandom sequence and change pixel value of the image via XOR.

Step 3: Sort the pseudorandom sequence for permutation.

Step 4: Shift the pixel positions by column using the sorted elements.

Step 5: Shift the pixel positions by row using the sorted elements.

To provide a better understanding of this scheme, the pseudocode is provided in Table 2.

![Figure 12](image.png)
The decryption process of the proposed algorithm is the reverse process of the encryption algorithm. A flowchart of the decryption process is shown in **Figure 13**.

The detailed decryption process includes the following steps.

**Input:** Plain image Org Img, Initial conditions for the chaotic system, Control parameter for the chaotic system, Average pixel value of the plain image

**Output:** Decrypted image

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Generate the pseudorandom sequence via the initial conditions and the average pixel values of the plain image.</td>
</tr>
<tr>
<td>2.</td>
<td>Sort the pseudorandom sequence for row and column recovery.</td>
</tr>
</tbody>
</table>

**Mathematical formulation:**

\[
\begin{align*}
    x(1) &\leftarrow x(1) + \text{Avg. pixel value} \\
    y(1) &\leftarrow y(1) \\
    z(1) &\leftarrow z(1) \\
    u(1) &\leftarrow u(1) \\
    s(1) &\leftarrow u(1)^*10^{-4} \cdot \text{floor}(u(1)^*10^{-4})
\end{align*}
\]

For \(i=1:1:m*n\) % Generate pseudorandom sequence that will be used for diffusion and permutation

\[
\begin{align*}
    [dx, dy, dz, du] &\leftarrow \text{Runge-Kutta} \ (x(i), y(i), z(i), u(i)) \\
    x(i+1) &\leftarrow x(i) + dx \\
    y(i+1) &\leftarrow y(i) + dy \\
    z(i+1) &\leftarrow z(i) + dz \\
    u(i+1) &\leftarrow u(i) + du \\
    s(i+1) &\leftarrow u(i+1)^*10^{-4} \cdot \text{floor}(u(i+1)^*10^{-4})
\end{align*}
\]

**Table 2.**

**Image encryption scheme.**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Count = 1</td>
</tr>
<tr>
<td>2.</td>
<td>For (i=1:m) % Diffusion Operation</td>
</tr>
<tr>
<td>3.</td>
<td>For (j=1:n)</td>
</tr>
<tr>
<td>4.</td>
<td>diff(Count) \leftarrow \text{mod} \ (s(Count)^*10^{-14}, 256)</td>
</tr>
<tr>
<td>5.</td>
<td>En_Dif(i,j) = \text{bitxor}(Org Img(i,j), diff(Count));</td>
</tr>
<tr>
<td>6.</td>
<td>Count = Count + 1;</td>
</tr>
<tr>
<td>7.</td>
<td>End</td>
</tr>
<tr>
<td>8.</td>
<td>End</td>
</tr>
<tr>
<td>9.</td>
<td>S_index \leftarrow \text{Sort}(s)</td>
</tr>
<tr>
<td>10.</td>
<td>For (i=1:n) % Column-wise permutation</td>
</tr>
<tr>
<td>11.</td>
<td>For (j=1:m)</td>
</tr>
<tr>
<td>12.</td>
<td>En_per_col(i,j) \leftarrow \text{Sort}(En_Dif, S_index)</td>
</tr>
<tr>
<td>13.</td>
<td>End</td>
</tr>
<tr>
<td>14.</td>
<td>End</td>
</tr>
<tr>
<td>15.</td>
<td>For (i=1:m) % Row-wise permutation</td>
</tr>
<tr>
<td>16.</td>
<td>For (j=1:n)</td>
</tr>
<tr>
<td>17.</td>
<td>En_Img(i,j) \leftarrow \text{Sort}(En_per_col, S_index)</td>
</tr>
<tr>
<td>18.</td>
<td>End</td>
</tr>
<tr>
<td>19.</td>
<td>End</td>
</tr>
</tbody>
</table>

The decryption process of the proposed algorithm is the reverse process of the encryption algorithm. A flowchart of the decryption process is shown in **Figure 13**.
Input: Ciphered image $\text{En}_\text{Img}$, Initial conditions for the chaotic system, control parameter for the chaotic system, $\text{Avg\_pixel\_value}$ of $\text{Org\_Img}$

Output: Plain Image $\text{Org\_Img}$

- $[m,n] \leftarrow \text{size}(\text{En}_\text{Img})$;
- $x(1) \leftarrow x(1) + \text{Avg\_pixel\_value}$
- $y(1) \leftarrow y(1)$
- $z(1) \leftarrow z(1)$
- $u(1) \leftarrow u(1)$
- $s(1) \leftarrow u(1) \times 10^4 - \text{floor}(u(1) \times 10^4)$

For $i=1:1:m \times n$ % Generate a pseudorandom sequence that will % be used for decryption
- $[dx, dy, dz, du] \leftarrow \text{Runge-Kutta} (x(i), y(i), z(i), u(i))$
- $x(i+1) \leftarrow x(i) + dx$
- $y(i+1) \leftarrow y(i) + dy$
- $z(i+1) \leftarrow z(i) + dz$
- $u(i+1) \leftarrow u(i) + du$
- $s(i+1) \leftarrow u(i+1) \times 10^4 - \text{floor}(u(i+1) \times 10^4)$
End

$S\text\_index \leftarrow \text{Sort}(s)$ % Row-wise permutation recovery
For $i=1:m$
- For $j=1:n$
- $\text{De\_per\_row} (i,j) \leftarrow \text{Sort} (\text{En\_Img}, S\text\_index)$
End
End

For $i=1:n$ % Column-wise permutation recovery
For $j=1:m$
- $\text{De\_per\_col} (i,j) \leftarrow \text{Sort} (\text{De\_per\_row}, S\text\_index)$
End
End

Count=1 % Count flag
For $i=1:m$ % Diffusion recovery
For $j=i:n$
- $\text{diff}(\text{Count}) \leftarrow \text{mod} (s(\text{Count}) \times 10^4, 256)$ % transform s, which could be used for XOR
- $\text{Org\_Img} (i,j) = \text{bitxor} (\text{De\_per\_col} (i,j), \text{diff} (\text{Count}))$; % Bitwise exclusive OR
End
End

Table 3. Image decryption scheme.
Step 3: Shift the pixel positions by row
Step 4: Shift the pixel positions by column
Step 5: Transform the pseudorandom sequence and recover the pixel values of the image via XOR

To provide a better understanding of this scheme, the pseudo-code is provided in Table 3.

In this system, all the initial conditions and control parameters can be considered as secret keys. Because the basin of attraction of each initial condition is greater than 1, it could have more than $10^{15\times4} \times 10^{60}$ choices via a resolution of $10^{-15}$, in terms of a numeric calculation. Moreover, if a range of control parameters are considered for the key space, the key space of this system would far exceed $10^{90}$. Such a large key space provides sufficient security against brute-force attacks.

Figure 14. The testing results of encryption and decryption: (a) is the plain image of cameraman; (b) is the encrypted image of cameraman; (c) is the decrypted image of cameraman; (d) is the plain image of breast CT image; (e) is the encrypted image of breast CT image; (f) is the decrypted image of breast CT image; (g) is the plain image of thorax CT image; (h) is the encrypted image of thorax CT image; (i) is the decrypted image of thorax CT image.
Correlation coefficients of adjacent pixels in the plain and encrypted image are shown in Table 4. The NPCR and UACI score of CT image are 99.5804% and 33.3227%.

From the above security analysis, the proposed scheme can provide high security for cryptographic applications.

4.3. Spread spectrum communication

Chaotic systems can also use for spread spectrum communication propose. Different chaos shift keying (DCSK) technology employs nonperiodic and wideband chaotic signals as carriers so as to achieve the effect of spectrum spreading in the process of digital modulation. Figure 15 shows the scheme of modulation for DCSK.

In this scheme, every bit has two time slots. The first time slot is used for transmission of a chaotic sequence for the reference signal. The second time slot is used for transmission of another chaotic sequence for the reference signal which has the same length as the first time slot. If the information bit is +1, then the information signal is exactly the same as the reference signal. If the information signal bit is −1, then the information signal is the negative of the reference signal. For bits \( b_k \), the signal at time \( k \) is:

\[
s_i = \begin{cases} 
  x_i & 2k\beta < i \leq (2k + 1)\beta \\
  b_kx_i / \beta & (2k + 1)\beta < i \leq 2(k + 1)\beta 
\end{cases}
\]

(10)

Where \( \beta \) is the number of sampling points. The spreading factor (SF) in the DCSK system is \( SF = 2\beta \).

<table>
<thead>
<tr>
<th>Figure name</th>
<th>Direction</th>
<th>Plain-image</th>
<th>Ciphered image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman Image</td>
<td>Horizontal</td>
<td>0.983146</td>
<td>0.001731</td>
</tr>
<tr>
<td>Cameraman Image</td>
<td>Vertical</td>
<td>0.990025</td>
<td>0.004141</td>
</tr>
<tr>
<td>Cameraman Image</td>
<td>Diagonal</td>
<td>0.973249</td>
<td>0.000324</td>
</tr>
<tr>
<td>Breast CT image</td>
<td>Horizontal</td>
<td>0.978292</td>
<td>0.002500</td>
</tr>
<tr>
<td>Breast CT image</td>
<td>Vertical</td>
<td>0.955481</td>
<td>0.006207</td>
</tr>
<tr>
<td>Breast CT image</td>
<td>Diagonal</td>
<td>0.940737</td>
<td>0.003071</td>
</tr>
<tr>
<td>Thorax CT image</td>
<td>Horizontal</td>
<td>0.994585</td>
<td>0.001267</td>
</tr>
<tr>
<td>Thorax CT image</td>
<td>Vertical</td>
<td>0.994761</td>
<td>0.001267</td>
</tr>
<tr>
<td>Thorax CT image</td>
<td>Diagonal</td>
<td>0.991973</td>
<td>0.001558</td>
</tr>
</tbody>
</table>

Table 4. Correlation coefficients of adjacent pixels in the plain and encrypted image.

Figure 15. Scheme of DCSK modulation.
For demodulation as shown in Figure 16, the receiver calculates the correlation between the received signal \( r_i \) and the signal \( r_i - \beta \), which is \( r_i \) delayed by \( \beta \). After a time \( k \), the output of the correlator is:

\[
Z_k = \sum_{i=(2k+1)\beta+1}^{i=(2k+1)\beta+1} r_i r_i - \beta
\]  

(11)

Thus, the information bit \( b_k \) can be restored by the sign of the decision variable:

\[
\hat{b}_k = \text{sgn} \left[ Z_k \right]
\]  

(12)

The obtained BER performance under additive white Gaussian noise (AWGN) channels for spreading factor \( 2\beta = 200 \) is shown in Figure 17. From the comparison results, DCSK can have a lower BER when using this system as a carrier signal in the presence of noise.

5. Conclusions

In this chapter, we first described a third order chaotic system with hyperbolic sine nonlinearity, then we introduced the method to expand this chaotic system to high order chaotic systems. After that, we introduced the method to construct the derived chaotic torus-chaotic systems. Finally, we introduced some applications such as random number generator algorithm, spread spectrum communication and image encryption schemes. The contribution of this chapter is that it systematically summarizes the design method, the dynamic behavior and typical engineering application of chaotic systems with hyperbolic sine nonlinearity, which may widen the current knowledge of chaos theory and engineering applications based on chaotic systems.
Acknowledgements

Jizhao Liu has received research grants from Sun Yat-sen University. This study was supported by the Fundamental Research Funds for the Central Universities. No. 19lgpy230.

Conflict of interest

The authors declare that they have no conflict of interest.

Notes/thanks/other declarations

The authors would like to thank professor Julien Clinton Sprott for helpful discussion.

Author details

Jizhao Liu,1,2*, Yide Ma,3, Jing Lian3 and Xinguo Zhang2

1 School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China

2 School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China

3 School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu, China

*Address all correspondence to: liujizhao@mail.sysu.edu.cn

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
References


