Computers, Materials & Continua CMC, vol.62, no.2, pp.833-847, 2020

Stability of Nonlinear Feedback Shift Registers with Periodic Input

Bo Gao[1, *], Xuan Liu[1], Xiaobo Wu[1, *], Shudong Li[2, *], Zhongzhou Lan[1], Hui Lu[2, *] and Boyan Liu[1]

Abstract: The stability of Non-Linear Feedback Shift Registers (NFSRs) plays an important role in the cryptographic security. Due to the complexity of nonlinear systems and the lack of efficient algebraic tools, the theorems related to the stability of NFSRs are still not well-developed. In this paper, we view the NFSR with periodic inputs as a Boolean control network. Based on the mathematical tool of semi-tensor product (STP), the Boolean network can be mapped into an algebraic form. Through these basic theories, we analyze the state space of non-autonomous NFSRs, and discuss the stability of an NFSR with periodic inputs of limited length or unlimited length. The simulation results are provided to prove the efficiency of the model. Based on these works, we can provide a method to analyze the stability of the NFSR with periodic input, including limited length and unlimited length. By this, we can efficiently reduce the computational complexity, and its efficiency is demonstrated by applying the theorem in simulations dealing with the stability of a non-autonomous NFSR.

Keywords: Non-Linear Feedback Shift Register (NFSR), Boolean Network (BN), Semi-Tensor Product (STP), transition matrix, stability, periodic input.

1 Introduction

Non-liner feedback shift registers (NFSRs), a generalization of Linear feedback shift registers (LFSRs) [Golomb (1982)], have many advantages, such as efficient implementation, large period and good statistical performance. They can be successfully applied in the coder/decoder, stream cipher and pseudo-random number generator. As a main block of a coder/decoder, each feedback function of the NFSR represents a decoding algorithm [Massey and Liu (1964)], which can be aptly used in many communication applications, such as mobile communications and satellite communications [Riad and Ke (2018)]. In addition, by virtue of their large period, the NFSRs are also used to design a stream cipher, in which the controlling NFSRs are cascaded with controlled NFSRs, such that the outputs of the controlling NFSRs are the

[1] School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot, 010051, China.

[2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China.

[3] School of Electrical and Electronic Engineering, University of Manchester, Manchester M13 9PL, UK.

[*] Corresponding Authors: Bo Gao. Email: gaobonmghhht@163.com;
Shudong Li. Email: lishudong@gzhu.edu.cn.

inputs of the controlled NFSRs [Zhong and Lin (2016)], as demonstrated by Grain [Hell, Johansson and Meier (2007)], Trivium and Mickey. Furthermore, NFSRs are of great interest because they show good statistical performance on the generation of pseudo-random sequences, which are hard to break by current cryptanalysis attacks [Zeng, Yang, Wei et al. (1991)].

Nonlinear systems have been applied in many fields [Li, Wu, Zhao et al. (2018); Han, Tian, Huang et al. (2018); Qiu, Chai, Liu et al. (2018); Wang, Liu, Qiu et al. (2018)] in which the stability analysis is a main component in the research of the controlled nonlinear system [Xu, Xiang and Sachnev (2018); Tang, Ling, Yao et al. (2018)]. Since stable NFSRs can limit error propagation during the process of decoding, the security of NFSRs can be well-developed through the use of the theories of stability of nonlinear systems. NFSRs consists of two basic kinds, autonomous NFSRs (without input) and non-autonomous NFSRs (with inputs). Some studies focus on the stability of autonomous NFSRs [Mowle (1966, 1967); Ma, Qi and Tian (2013); Zhong and Lin (2015)], while less attention is given to the non-autonomous NFSRs. Massay and Liu introduced the concept of stability of non-autonomous NFSR, and designed a progress-driven stable version of an NFSR [Massey and Liu (1964)]. Due to the lack of efficient mathematical tools, research achievements in this field are rare. In 2012, Cheng and his co-workers introduced a new method to achieve the algebraic form of a Boolean network by using the semi-tensor product (STP) [Cheng and Qi (2010)]. Thanks to the mathematical tool, some problems, in fields such as physics [Gao, Li, Peng et al. (2013); Gao, Deng, Zhao et al. (2017); Li (2016)] and system science [Gao, Shi, Yang et al. (2014); Gao, Deng, Zhao et al. (2017); Li, Yan and Karimi (2018)], involving Boolean control networks, can be converted into algebraic problems. This is also very helpful for analyzing non-autonomous NFSRs [Gao, Liu, Lan et al. (2018)]. Based on these contributions, Zhong et al. [Zhong and Lin (2016)] proposed a novel way to study the stability of NFSRs, via a technique that has lower computational complexity than the exiting method. However, due to the complexity of controlled non-linear systems, the study of stability of the NFSR with different kinds of input has yet to reach a satisfactory conclusion.

On the basis of these efficient mathematical tools, this paper mainly describes the state transition of the non-autonomous NFSR. In order to achieve a less computationally complex method, we focus on the transition matrix of the NFSR corresponding with different periodic input sequences during the process of expression. Based on these works, we can provide a method to analyze the stability of the NFSR with periodic input, including limited length and unlimited length.

This paper is organized as follows. In Section 2, some basic concepts and related definitions are introduced. Section 3 provides the main results including a description of the stability of an NFSR with periodic input. We also discuss the stability of an NFSR with periodic input of limited length and unlimited length, respectively. Section 4 ends the paper with a brief conclusion.

2 Background

2.1 Semi-tensor product (STP)

The STP is a more general form of a matrix product that allows us to perform a matrix product when the size of the column of one matrix is not equal to the size of the row of the other. The definition is defined as follows:

Definition 2.1.1. Assume that M is a matrix of dimensions $m \times n$, N is a matrix of dimensions $p \times q$, and let a be the least common multiple of n and p. The STP of M and N is defined as

$$M \ltimes N = (M \otimes I_{\frac{a}{n}})(B \otimes I_{\frac{a}{p}}) \tag{1}$$

where $\otimes$ is the Kronecker product and I_k is an identity matrix of dimension k.

Obviously, if $n = p$, then the STP of M and N in Definition 2.1 will result ina conventional matrix product. For the sake of convenience, the symbol "$\ltimes$" can be omitted from the definition unless specifically required.

2.2 Boolean network (BN) and Boolean control network (BCN)

These autonomous NFSRs can be regarded as a Boolean network, which consists of Boolean functions with finite logical variables. The variables in a Boolean network can be classified as "1" and "0". So an autonomous NFSR can be described as a Boolean network as follows:

$$x_i(t+1) = g_i(x_1(t), x_2(t), ..., x_n(t)), (i = 1, 2, ...n) \tag{2}$$

where $g_i(i = 1, 2, ..., n)$ are logical functions, and $x_i(t)(i = 1, 2, ..., n)$ are the states of the nodes $i(i = 1, 2, ..., n)$ in the time t.

Non-autonomous NFSR is the autonomous NFSR with an input sequence, which can be expressed as a Boolean control network with n nodes and m inputs as

$$x_i(t+1) = G_i(x_1(t), x_2(t), ..., x_n(t), u_1(t), ..., u_m(t))(i = 1, 2, ...n), \tag{3}$$

where $G_i(i = 1, 2, ..., n)$ are logical functions with inputs, $x_1(t), x_2(t), ..., x_n(t)$ are the states of nodes $i(i = 1, 2, ..., n)$ in the time t, and $u_1(t), ..., u_m(t)$ are the inputs in the time t.

Denote $x(t) = \{x_1(t), x_2(t), ..., x_n(t)\}$ as the state of the network in the time t. Through the NFSR (2) and the NFSR (3), we can then describe the state of Boolean (control) network in the time $t+1$ according to the state $x(t)$.

In order to effectively describe the state of the network in the time t, we transform the network into an algebraic form, and introduce the definitions related to the transfer of the logic-based problems into algebraic problems below.

Assume that I_k can be represented as $I_k = \left\{ \delta_k^i \middle| i = 1, 2, ...k \right\}$, where δ_k^i is the ith column of an identity matrix. The logical variables "0" and "1" can be denoted by $\delta_2^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and

$\delta_2^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, respectively. Through the use of the mathematical tool, the STP, a Boolean function can be transferred into an algebraic form.

Definition 2.2.1. Given a Boolean function $g(x_1,...,x_i,...,x_n)$ with logical variables $x_1,...,x_i,...,x_n$, it can be further expressed as a multi-linear form via the STP,

$$g(x_1,...,x_i,...,x_n) = Mx_1,...,x_i,...,x_n, \tag{4}$$

where g is the logical function, $x_i \in I_2 (i = 1,2,...,n)$, and M is the structure matrix of g, which is expressed as

$$M = \begin{bmatrix} a_1 & a_2 & ... & a_j & ... & a_{2^n} \\ 1-a_1 & 1-a_2 & ... & 1-a_j & ... & 1-a_{2^n} \end{bmatrix} \tag{5}$$

with $a_j = 0$ *or* 1 $(j = 1,2,....,2^n)$.

Through the use of the STP, a Boolean function can be represented as a muti-linear form. We then denote the structure matrix corresponding to the node x_i by a 2×2^n matrix M_i. Through Definition 2.2.1, any Boolean function in NFSR (2) can be represented as

$$x_i(t+1) = M_i x_1(t)...x_i(t)...x_n(t) = M_i x(t), \tag{6}$$

where $i = 1,2,...,n$.

Based on the properties of the STP, a further transformation of the algebraic form of the Boolean (control) network is shown as follows, such that a Boolean network can be converted into a conventional discrete-time linear system.

Definition 2.2.2. Assuming $x(t) = \ltimes_{i=1}^n x_i(t)$, the NFSR (2) has an equivalent algebraic representation

$$x(t+1) = M_1 x(t) M_2 x(t)...M_n x(t) \coloneqq Lx(t), \tag{7}$$

where $L = [\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, ..., \delta_{2^n}^{i_{2^n}}]$, $i_1,...,i_{2^n} \in \{1,2,...,2^n\}$ is the transition matrix of NFSR (2), a $2^n \times 2^n$ matrix, satisfying

$$Col_j(L) = \ltimes_{i=1}^n Col_j(M_i), \tag{8}$$

where $i = 1,...,n$, $j = 1,...,2^n$. For convenience, according to the special properties of the STP, the transition matrix L can be rewritten as $L = \delta_{2^n}[i_1, i_2, ..., i_{2^n}]$.

When the input sequence is added into the autonomous NFSR, the study is turned into a non-autonomous NFSR.

Similarly, if we assume that $x(t) = \ltimes_{i=1}^n x_i(t)$, the algebraic representation of the non-autonomous NFSR is performed as shown in the following cases. We begin with discuss the algebraic representation of a non-autonomous NFSR with multi-steps.

Case 1: If there exists a series of input sequence $u^i(t)$ in a period $i = 1,...,l$, after *l* steps, the state of non-autonomous NFSR in the *l*-th step can be solved as

$$x(t+1) = (\tilde{L})^l u^1(t)u^2(t)...u^l(t)x(t), \tag{9}$$

where $x(t)$ is state of the system in the time *t*, the $u^i(t) \in I_2$ represents the input in the *i*th step, and $(\tilde{L})^l$ is a $2^n \times 2^{n+l}$ matrix. Through the transition matrix $\tilde{L}$, we can naturally get the state of the NFSR with a regular periodic input sequence.

If, however, when there exist a series of inputs $u_1(t), u_2(t), ..., u_m(t)$ in one step, the state transition of the NFSR with single step can be performed as follows:

Case 2: Assume there exists a series of inputs $u_1(t), u_2(t), ..., u_m(t)$ of the system, after one step, the NFSR (3) with the input in time *t* can be described as an algebraic representation,

$$x(t+1) = \tilde{L}u_1(t)u_2(t)...u_m(t)x(t), \tag{10}$$

where $x(t)$ is the state of the network in time *t*, the $u_1(t), u_2(t), ..., u_m(t)$ are inputs of the network in one step, and $\tilde{L}_{2^{n+m}}$ is the transition matrix of NFSR (10).

In most cases, there only exists one input per step in the NFSR. Thus, NFSR (10) can be rewritten as

$$x(t+1) = \tilde{L}u(t)x(t), \tag{11}$$

where $\tilde{L}$ is a $2^n \times 2^{n+1}$ matrix, which consists of transition matrix L and matrix $\overline{L}(\overline{L} \neq L)$, namely the matrix $\tilde{L} = [\overline{L} | L]$, and $u(t)$ is the only input in one step.

Case 2 is a special situation of Case 1. In this paper, we will focus on the state transition under Case 1.

3 Results

3.1 State of the NFSR with input

In this subsection, we mainly focus on the description of the transition matrix so that we can describe the state transition of a stable NFSR with different input sequences. Since the logical variables in a system can be classified as "0" and "1", the corresponding inputs and transition matrices are also different. Next, we analyze the state transition in the circumstances of the two different kinds of input.

Theorem 3.1.1. Consider NFSR (11) with single input. Let the logical variables δ_2^2 and δ_2^1 correspond to the logical variables "0" and "1", respectively, and denote the corresponding transition matrix of logical variables "0" and "1" by $\tilde{L}_0$ and $\tilde{L}_1$. When $u(t) = \delta_2^2$, NFSR (11) can be rewritten as

$$x(t+1) = \tilde{L}\delta_2^2 x(t), \tag{12}$$

where $Col(\tilde{L}u_0) = Col(L)$, and $\tilde{L} \in M_{2^{n+1}}$. Here, we note that $\tilde{L}u_0 = L$, satisfying

$$x(t+1)=\tilde{L}\delta_2^2 x(t)=Lx(t), \tag{13}$$

where $L\in M_{2^n}$.

Conversely, if the input is δ_2^1, then NFSR (11) can be rewritten as

$$x(t+1)=\tilde{L}\delta_2^1 x(t), \tag{14}$$

where $Col(\tilde{L}u_1)\neq Col(L)$, and $\tilde{L}_1\in M_{2^{n+1}}$. Similarly, we note that $\tilde{L}_1u_1=\overline{L}$, satisfying

$$x(t+1)=\tilde{L}\delta_2^1 x(t)=\overline{L}x(t), \tag{15}$$

where $\overline{L}\in M_{2^n}$.

Proof. Let us consider NFSR (11), where $\tilde{L}u(t)$ is different in different circumstances. Since the $\tilde{L}$ is a $2^n\times 2^{n+1}$ matrix and $u(t)$ is a 2×1 matrix, the ranks of matrices are not equal, so we use the STP to determine the product of $\tilde{L}$ and $u(t)$. By the definition of the STP, $\tilde{L}$ is split into two equal blocks, namely $\tilde{L}=[\overline{L}\,|\,L]$, which are $2^n\times 2^n$ matrices, respectively. The effects of different input are considered in the following cases.

Case 1: If $u(t)=\delta_2^2$, then the matrix product of $\tilde{L}$ and δ_2^2 can be defined as

$$\tilde{L}\ltimes\delta_2^2=\left[\overline{L}\,|\,L\right]\ltimes\begin{bmatrix}0\\1\end{bmatrix}=L. \tag{16}$$

The input "0" is a invalid input.

Case 2: If the input is "1", then the product between $\tilde{L}$ and δ_2^1 is defined as

$$\tilde{L}\ltimes\delta_2^1=\left[\overline{L}\,|\,L\right]\ltimes\begin{bmatrix}1\\0\end{bmatrix}=\overline{L}. \tag{17}$$

The input "1" is a valid input.

Next, we provide an example to illustrate the different effects for the inputs of "0" and "1". It should be noted that, in most cases, the logical relationship between the state nodes and inputs is exclusive-or $A+B$. Thus, we assume that the logical relationship between the state nodes and inputs is the exclusive-or in the following examples.

Example 3.1.2. Consider a 4-stage autonomous NFSR, in which the network consists of feedback functions

$$\begin{cases}x_1(t+1)=x_2(t)\\x_2(t+1)=x_3(t)\\x_3(t+1)=x_4(t)\\x_4(t+1)=x_1(t)+x_2(t)+x_3(t)+x_2(t)x_3(t)\end{cases}, \tag{18}$$

where $x_i(t)(i=1,2,3,4)$, are states of the nodes.

The corresponding algebraic representation can be expressed as,

$$x(t+1) = Lx(t), \tag{19}$$

where the transition matrix $L = \delta_{16}[2\ 4\ 6\ 8\ 10\ 12\ 13\ 15\ 1\ 3\ 5\ 7\ 9\ 11\ 14\ 16]$.

The BN(19) with single input can be represented as

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = x_4(t) \\ x_4(t+1) = x_1(t) + x_2(t) + x_3(t) + x_2(t)x_3(t) + u(t) \end{cases}, \tag{20}$$

where $x_i(t)(i = 1,2,3,4)$ are states of the nodes *i*, and *u(t)* is the single input.

With the periodic inputs, the corresponding algebraic representation of the non-autonomous NFSR (20) can be described as

$$x(t+1) = \tilde{L}u(t)x(t), \tag{21}$$

where $\tilde{L}$ satisfies .

$$\tilde{L} = \delta_{16}[1\ 3\ 5\ 7\ 9\ 11\ 14\ 16\ 2\ 4\ 6\ 8\ 10\ 12\ 13\ 15\ 2\ 4\ 6\ 8\ 10\ 12\ 13\ 15\ 1\ 3\ 5\ 7\ 8\ 11\ 14\ 16] \tag{22}$$

The aim of this example is to illustrate the different effects caused by the different inputs "0" and "1". When the input is "0", NFSR (21) can be further expressed as

$$x(t+1) = \tilde{L}\delta_2^2 x(t) = \tilde{L}_0 x(t), \tag{23}$$

where δ_2^2 is the logical variable corresponding to the input "0", and the matrix $\tilde{L}_0$ is

$$\tilde{L}_0 = L = \delta_{16}[2\ 4\ 6\ 8\ 10\ 12\ 13\ 15\ 1\ 3\ 5\ 7\ 8\ 11\ 14\ 16]. \tag{24}$$

On the other hand, when the input is "1", NFSR (21) can be further described as

$$x(t+1) = \tilde{L}\delta_2^1 x(t) = \tilde{L}_1 x(t), \tag{25}$$

where δ_2^1 is the logical variable corresponding to the input "1", and the matrix $\tilde{L}_1$ is

$$\tilde{L}_1 = \overline{L} = \delta_{16}[1\ 3\ 5\ 7\ 9\ 11\ 14\ 16\ 2\ 4\ 6\ 8\ 10\ 12\ 13\ 15]. \tag{26}$$

If we divide the transition matrix $\tilde{L}$ into two matrices L and $\overline{L}$ of the same dimensions $2^4 \times 2^4$, respectively, i.e., $\tilde{L} = \left[L \mid \overline{L} \right]$, then we note that $\tilde{L}_0 = L$ and $\tilde{L}_1 = \overline{L}$, which effectively illustrates the theorem mentioned above.

3.2 Limited length of periodic input

The periodic input sequence can be classified as limited or unlimited in length according to their length properties. In this subsection, we analyze the problem of the limited length of periodic inputs, where the short-period repeats finite times. Then, based on the stability of the NFSR, we focus on the state transition after periodic input in a stable NFSR.

The definition for representing the method of limited length of periodic input is shown as follows:

Definition 3.2.1. Assume that there exists a series of periodic input sequences, written as

$$\underbrace{i_1,i_2,...,i_c}_{1st},\underbrace{i_1,i_2,...,i_c}_{2nd},...,\underbrace{i_1,i_2,...,i_c}_{kth}, \tag{27}$$

where $i_1,i_2,...,i_c$ are all the logical numbers in a period, and k is the number of times the period repeats. For convenience, we will denote the periodic input sequence by $(i_1,i_2,...,i_c)\big|\_k$, in which the sequence includes k folds of a period, and the length of the periodic input is $l=c\cdot k$. The k folds means the period is repeated k times.

For example, if there exists a fold of periodic input including logical numbers $(0,0,1,1,0)$, which repeat 8 times, which performed as

$$\underbrace{0,0,1,1,0}_{1st},\underbrace{0,0,1,1,0}_{2nd},...,\underbrace{0,0,1,1,0}_{8th}, \tag{28}$$

then the sequence of periodic input is denoted by $(0,0,1,1,0)\big|\_8$.

Next, we analyze the state transition generated by limited length of periodic input sequence.

Theorem 3.2.2. Given the previously assumed periodic inputs $(i_1,i_2,...,i_c)\big|\_k$, assume there exists a matrix Θ, which satisfies

$$\Theta=\tilde{L}\ltimes\begin{bmatrix} i_c \\ 1-i_c \end{bmatrix}\ltimes\tilde{L}\ltimes\begin{bmatrix} i_{c-1} \\ 1-i_{c-1} \end{bmatrix}\ltimes...\ltimes\tilde{L}\ltimes\begin{bmatrix} i_1 \\ 1-i_1 \end{bmatrix} \tag{29}$$

where $i_\alpha=0\ or\ 1$, $\alpha=1,2,...,c$, are logical numbers in a complete short-period, k is the number of times the short-period repeats, and Θ represents the transition matrix of NFSR (9).

NFSR (9) with input sequence $(i_1,i_2,...,i_c)\big|\_k$ can be further expressed as

$$x(t+1)=\Theta^k x(t), \tag{30}$$

where k is a positive integer.

Proof: Beginning with the state $x(0)$, the state transition attached to a periodic input of one fold is performed as

$$\begin{aligned} & x(0) \\ \to\ & x(1)=\tilde{L}\ltimes u^1\ltimes x(0) \\ \to\ & x(2)=\tilde{L}\ltimes u^2\ltimes x(1)=\tilde{L}\ltimes u^2\ltimes\tilde{L}\ltimes u^1\ltimes x(0) \\ \to\ & ... \\ \to\ & x(c)=\tilde{L}\ltimes u^c\ltimes x(c)=\tilde{L}\ltimes u^c\ltimes\tilde{L}\ltimes u^{c-1}\ltimes x(c-1)=\underbrace{\tilde{L}\ltimes u^c\ltimes...\ltimes\tilde{L}\ltimes u^1}_{c}\ltimes x(0). \end{aligned}$$

Constructing a matrix $\Theta = \underbrace{\tilde{L} \ltimes u^c \ltimes ... \ltimes \tilde{L} \ltimes u^1}_{c}$, we can conclude that

$$x(t+1) = \Theta x(t). \tag{31}$$

If the input sequence increases to k folds, namely, there exists a series of periodic input sequence $(i_1, i_2, ..., i_c)\big|\_k$, based on the Eq. (30) and starting from the state x_0, the corresponding state transition can be constructed as

$$x(0) \to x(c) = \Theta x(0) \to x(2c) = \Theta x(c) = \Theta^2 x(0) \to ... \to x(kc) = \Theta^k x(0)$$

$$x_1 \to x(c+1) = \Theta x(1) \to x(2c+1) = \Theta x(c+1) = \Theta^2 x(1) \to ... \to x(kc+1) = \Theta^k x(1)$$

...

$$x(c-1) \to x(2c-1) = \Theta x(c-1) \to x(3c-1) = \Theta x(2c-1) = \Theta^2 x(c-1) \to ...$$

$$\to x((k+1)c-1) = \Theta^k x(c-1)$$

Therefore, we can conclude that

$$x(t+1) = \Theta^k x(t), \tag{32}$$

where the Θ satisfies

$$\Theta = \tilde{L} \ltimes \begin{bmatrix} i_c \\ 1-i_c \end{bmatrix} \ltimes \tilde{L} \ltimes \begin{bmatrix} i_{c-1} \\ 1-i_{c-1} \end{bmatrix} \ltimes ... \ltimes \tilde{L} \ltimes \begin{bmatrix} i_1 \\ 1-i_1 \end{bmatrix}, \tag{33}$$

where, $i_\alpha = 0 \ or \ 1, \alpha = 1, 2, ..., c$.

The matrix Θ is the transition matrix of the network, which is helpful in transferring the network into a linear representation. The value of the matrix Θ is influenced by the periodic input and the length of the attractor is also affected by the length of these inputs. After periodic inputs of length-*c* in a fold, the NFSR can generate length-*kc* attractors (*k=1,2,...c*).

According to Theorem 3.2.2, we can determine the state after *kc* steps in an NFSR with periodic inputs. In the calculation of the Θ^k, we compute $\Theta^2 = \Theta \ltimes \Theta$ at first, then compute $\Theta^4 = \Theta^2 \ltimes \Theta^2$,..., and by this analogy, we can get the value of the matrix Θ^k. In the process of the calculation, the computational complexity is $O(\log_2 l)$.

Next, we provide an example to illustrate Definition 3.2.1 and Theorem 3.2.2.

Example 3.2.3. Consider the feedback functions of the NFSR from Example 3.1.2. If there exist a series of inputs sequence (111101000110010) in a period, the expression of the non-autonomous NFSR can be described as

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = x_4(t) \\ x_4(t+1) = x_1(t) + x_2(t) + x_3(t) + x_2(t)x_3(t) + u^i(t) \end{cases} \tag{34}$$

where $\{i \mid 1 \le i \le 15, i \in Z^*\}$, and $u^i(t)$ is the ith logical variable attached to the ith logical number in the input sequence (111101000110010), which are ordered as $(u^1, u^2, ..., u^c)$. The aim of this example is to show the state transition in this NFSR with periodic input.

Based on Theorem 3.2.2, we begin by calculating the value of the matrix Θ as follows:

$$\Theta = L \ltimes \overline{L} \ltimes L \ltimes L \ltimes \overline{L} \ltimes \overline{L} \ltimes L \ltimes L \ltimes L \ltimes \overline{L} \ltimes L \ltimes \overline{L} \ltimes \overline{L} \ltimes \overline{L} \ltimes \overline{L}. \quad (35)$$

In Example 3.1.2, we found that, $L = \delta_{16}[2\ 4\ 6\ 8\ 10\ 12\ 13\ 15\ 1\ 3\ 5\ 7\ 9\ 11\ 14\ 16]$ and $\overline{L} = \delta_{16}[1\ 3\ 5\ 7\ 9\ 11\ 14\ 16\ 2\ 4\ 6\ 8\ 10\ 12\ 13\ 15]$. We can now substitute the values of matrix L and matrix $\overline{L}$ into the expression of matrix Θ, and through the definition of the matrices product, calculate the matrix Θ as follows:

$$\Theta = \delta_{16}[3\ 13\ 6\ 10\ 7\ 8\ 16\ 11\ 12\ 4\ 1\ 5\ 9\ 15\ 14\ 2]. \quad (36)$$

When the period repeats k times, which means there exists a periodic input sequence $(111101000110010)\big|_{\_} k$, the state transition can be expressed as

$$x(t+1) = \Theta^k x(t). \quad (37)$$

Through the above equation, we can calculate the state in the NFSR after periodic inputs. For the state δ_{16}^1, we want to find the state after the periodic input $(111101000110010)\big|_{\_} 8$. Through Theorem 3.2.2, we can determine the value in the matrix to be $\Theta^8 = [8\ 13\ 11\ 4\ 7\ 1\ 16\ 3\ 12\ 10\ 6\ 5\ 9\ 14\ 15\ 2]$. Note that we should calculate $\log_2 8$ times in this method, while we would calculate eight times in the traditional method. The convenience of multiple matrices production is more obvious if the k is large enough. Then, we calculate the state after the periodic input $(111101000110010)\big|_{\_} 8$, in which $x(t+1) = \Theta^8 \delta_{16}^1 = \delta_{16}^8$.

Based on Theorem 3.2.2, when k=5, we note that $\delta_{16}^1 = \Theta^5 \delta_{16}^1$. We then find the state δ_{16}^1 can return to the initial state after 75 state transitions, and the length of the corresponding attractor is 75. At this moment, the NFSR described in the system (34) reaches stability.

3.3 Unlimited length of periodic input

In the previous section, we analyzed the periodic input with limited length, but in the actual application, unlimited-time input sequences are more likely. Therefore, we now consider the stability of NFSR with unlimited length of periodic input. Then we introduce a method for representing the unlimited length of periodic input sequence.

Definition 3.3.1. Let the symbol "∞" expresses the periodic input $i_1, i_2, ..., i_c$ repeated unlimited folds. The logical variable without predecessors is called the starting number. In the previous assumed input sequence, when the starting logical variable is i_1, the input

sequence can be represented as $(i_1, i_2, ..., i_c)\big|_{-}\infty$, indicating that the period input repeats unlimited times.

Because of the unlimited length of the periodic input, the starting number is different, the input sequence is also different. This means that given an unlimited input sequence, which started with different logical numbers, may produce a different order in a period, such as $(i_1, i_2, ..., i_{c-1,} i_c)\big|_{-}\infty$, $(i_2, i_3, ..., i_c, i_1)\big|_{-}\infty$, ..., $(i_c, i_1, ..., i_{c-2}, i_{c-1})\big|_{-}\infty$.

Considering the above analysis, c corresponding transition matrices $\Theta_i (i = 1, 2, ..., c)$ in the network with input preciously assumed are defined as follows:

Definition 3.3.2. Let $S_1 = (i_1, i_2, ..., i_{c-1,} i_c)\big|_{-}\infty$, $S_2 = (i_2, i_3, ..., i_c, i_1)\big|_{-}\infty$, ..., $S_c = (i_c, i_1, ..., i_{c-2}, i_{c-1})\big|_{-}\infty$, in which the starting nodes are different, and let the matrix Θ attached to the different inputs sequence $S_1, S_2, ..., S_c$ be denoted by $\Theta_1, \Theta_2, ..., \Theta_c$, such that

$$\begin{aligned}
\Theta_1 &= \tilde{L} \ltimes \begin{bmatrix} i_c \\ 1 - i_c \end{bmatrix} \ltimes ... \ltimes \tilde{L} \ltimes \begin{bmatrix} i_2 \\ 1 - i_2 \end{bmatrix} \ltimes \tilde{L} \ltimes \begin{bmatrix} i_1 \\ 1 - i_1 \end{bmatrix}, \\
\Theta_2 &= \tilde{L} \ltimes \begin{bmatrix} i_1 \\ 1 - i_1 \end{bmatrix} \ltimes ... \ltimes \tilde{L} \ltimes \begin{bmatrix} i_3 \\ 1 - i_3 \end{bmatrix} \ltimes \tilde{L} \ltimes \begin{bmatrix} i_2 \\ 1 - i_2 \end{bmatrix}, \\
&... \\
\Theta_c &= \tilde{L} \ltimes \begin{bmatrix} i_{c-1} \\ 1 - i_{c-1} \end{bmatrix} \ltimes ... \ltimes \tilde{L} \ltimes \begin{bmatrix} i_1 \\ 1 - i_1 \end{bmatrix} \ltimes \tilde{L} \ltimes \begin{bmatrix} i_c \\ 1 - i_c \end{bmatrix},
\end{aligned} \tag{38}$$

where $\tilde{L}$ is a $2^n \times 2^{n+1}$ matrix, and $i_1, i_2, ..., i_c$ are logical variables in the one period.

Accordingly, we can construct a corresponding representation of the NFSR with a periodic input of unlimited length as follows,

$$\begin{aligned}
&x(t+1) = \Theta_1 x(t) \\
&x(t+1) = \Theta_2 x(t) \\
&... \\
&x(t+1) = \Theta_c x(t)
\end{aligned}, \tag{39}$$

where $x(t)$ and $x(t+1)$ are states of the network in the time t and $t+1$.

From the point of cryptographical security in NFSRs, the stability is a fundamental condition, and the maximum period is preferable. Next, employing Definition 3.3.1, we provide a method to check the stability of the NFSR with periodic input of unlimited length so that we can enhance the cryptographical security.

Theorem 3.3.3. Assume an NFSR has a series of periodic input of unlimited length. Let i_α be the starting input node in a period of the input sequence, and denote Θ_α as the

transition matrix corresponding to the starting number i_α . Given a state $\delta_{2^n}^i (i = 1, 2, ..., 2^n)$, the non-autonomous NFSR is in stability if and only if,

$$\delta_{2^n}^i = \Theta_\alpha^k \delta_{2^n}^i , \tag{40}$$

where $\alpha \in \{1, ..., c\}$, and $\delta_{2^n}^i$ indicates a certain state for the system depicted by Eq. (9). The $k \in \{1, 2, ..., c\}$ is the minimum positive number satisfying Eq. (40).

Proof: As Definition 3.3.1 shows, the different starting nodes correspond to different Θ matrices. Therefore, the matrix Θ_α^k mentioned in Eq. (40) is also different. If we let the length of a period in the input sequence be *c*, we can deduce that the length of the attractor corresponding to the input with starting node i_α is *kc* (where $k \in \{1, 2, ..., c\}$, $\alpha \in \{1, 2, ..., c\}$). In other words, the state of the system described in Eq. (40) could return to the initial state after *kc* steps.

Assuming that there exists a previous assumed input sequence $(i_1, i_2, ..., i_{c-1}, i_c)\big|\_\infty$, we strive to give an algorithm in order to describe the state transition corresponding to the different transition matrices.

Algorithm 3.3.4. Check stability of the NFSR with unlimited length periodic input

1. Initialize set $S_i = \delta_{2^n}^i$, $P_i = \Phi$
2. For $\alpha = 1$ to c do
3. $P_i = S_i$, $X = S_i$, $L = \Theta_\alpha$
4. For $k = 1$ to c do
5. $X = L^k X$
6. If $X == P_i$
7. Break
8. Output α, k
9. End for
10. End for

When the NFSR is stable, the minimum length of the periodic input sequence is $1 \cdot c$, and the maximum length is $c \cdot c$. Through Algorithm 3.3.4, when the starting node of the input sequence is i_α , the state $\delta_{2^n}^i$ is stable after kc steps of transition.

Example 3.3.5. Consider feedback functions of the non-autonomous NFSR in Example 3.2.3. If there exists a series of unlimited length periodic input, it is unknown which one is the starting node in the input sequence. Assume that the periodic input is $(111101000110010)\big|\_\infty$. By different starting nodes, let the input sequence in a period

be classified as $S_1 = \{111101000110010\}$, $S_2 = \{011110100011001\}$, ..., $S_{15} = \{111010001100101\}$. Next, we discuss different matrices attached to different starting nodes. According to Definition 3.3.2, let $\Theta_1, \Theta_2, ..., \Theta_{15}$ be the transition matrices attached to the different input sequence $S_1, S_2, ..., S_{15}$. We can calculate the value of the matrices respectively as follows:

$$\Theta_1 = \delta_{16}[3\ 13\ 6\ 10\ 7\ 8\ 16\ 11\ 12\ 4\ 1\ 5\ 9\ 15\ 14\ 2], \tag{41}$$

$$\Theta_2 = \delta_{16}[5\ 8\ 10\ 7\ 11\ 1\ 4\ 9\ 14\ 2\ 16\ 13\ 12\ 15\ 3\ 6], \tag{42}$$

$$\Theta_3 = \delta_{16}[9\ 12\ 16\ 3\ 4\ 15\ 14\ 10\ 6\ 8\ 1\ 13\ 5\ 7\ 11\ 2], \tag{43}$$

$$\Theta_4 = \delta_{16}[2\ 11\ 8\ 16\ 15\ 1\ 5\ 10\ 7\ 9\ 13\ 14\ 6\ 12\ 3\ 4], \tag{44}$$

$$\Theta_5 = \delta_{16}[3\ 14\ 6\ 2\ 16\ 10\ 15\ 12\ 13\ 11\ 1\ 8\ 5\ 9\ 7\ 4], \tag{45}$$

$$\Theta_6 = \delta_{16}[9\ 6\ 5\ 11\ 2\ 12\ 15\ 4\ 10\ 16\ 1\ 3\ 14\ 13\ 7\ 8], \tag{46}$$

$$\Theta_7 = \delta_{16}[2\ 4\ 11\ 15\ 9\ 1\ 6\ 5\ 3\ 12\ 8\ 10\ 14\ 13\ 16\ 7], \tag{47}$$

$$\Theta_8 = \delta_{16}[6\ 4\ 7\ 8\ 15\ 5\ 3\ 14\ 11\ 1\ 9\ 2\ 12\ 16\ 10\ 13], \tag{48}$$

$$\Theta_9 = \delta_{16}[5\ 12\ 2\ 8\ 1\ 13\ 4\ 15\ 7\ 14\ 16\ 10\ 6\ 3\ 11\ 9], \tag{49}$$

$$\Theta_{10} = \delta_{16}[13\ 10\ 11\ 7\ 16\ 4\ 3\ 15\ 12\ 2\ 6\ 9\ 8\ 5\ 14\ 1], \tag{50}$$

$$\Theta_{11} = \delta_{16}[10\ 8\ 4\ 3\ 6\ 11\ 14\ 2\ 15\ 16\ 7\ 9\ 12\ 5\ 1\ 13], \tag{51}$$

$$\Theta_{12} = \delta_{16}[4\ 13\ 16\ 15\ 7\ 14\ 5\ 2\ 11\ 8\ 6\ 9\ 1\ 12\ 10\ 3], \tag{52}$$

$$\Theta_{13} = \delta_{16}[5\ 8\ 15\ 9\ 12\ 16\ 1\ 14\ 2\ 13\ 7\ 11\ 10\ 3\ 4\ 6], \tag{53}$$

$$\Theta_{14} = \delta_{16}[4\ 10\ 9\ 15\ 13\ 14\ 5\ 1\ 3\ 7\ 6\ 16\ 2\ 8\ 11\ 12], \tag{54}$$

$$\Theta_{15} = \delta_{16}[7\ 5\ 4\ 14\ 2\ 11\ 13\ 15\ 10\ 3\ 12\ 16\ 6\ 9\ 8\ 1]. \tag{55}$$

The above equations demonstrate that the different starting nodes attach to different transition matrices. If we then choose the δ_{16}^1 as the initial state, and apply Algorithm 3.3.4, the state is stable after the state transitions as follows:

$\delta_{16}^1 = \Theta_1^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_2^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_3^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_4^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_5^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_6^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_7^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_8^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_9^2 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{10}^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{11}^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{12}^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{13}^5 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{14}^7 \delta_{16}^1$, $\delta_{16}^1 = \Theta_{15}^7 \delta_{16}^1$.

Thus, the modified non-autonomous NFSR mentioned in this example is stable.

4 Conclusions

This paper applies the Boolean network to reduce the computational complexity about the method of description, in cases of the state transition in an NFSR with different kinds of

periodic input. Notably, we developed a novel method to analyze the stability of the system with unlimited length of periodic input. For a non-autonomous NFSR, the more stable it is, and the larger the period generated, the more security it will offer when used. Developing effective algorithms or approximate techniques for the present approach will be a challenging problem in future work.

Acknowledgement: This work is supported by the National Natural Science Foundation of China (Grants Nos. 61672020, U1803263, 61662069, 61762068, 31560622, 31260538, 30960246, 31672385, 71761029), Project funded by China Postdoctoral Science Foundation (2013M542560, 2015T81129), and A Project of Shandong Province Higher Educational Science and Technology Program (No. J16LN61), Inner Mongolia Colleges and Universities Scientific and Technological Research Projects (Grant No. NJZC17148), CERNET Innovation Project (No. NGII20161209), Natural Science Foundation of Inner Mongolia Autonomous Region of china (No. 2017MS0610，No. 2017MS717), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (No. NJYT-18-A13), Inner Mongolia Key Laboratory of economic data analysis and mining China-Mongolia Scientific Research Capacity Building of Incubator, Joint Laboratory and Technology Transfer Center, Education research project of national finance and economics (No. MZCJYB1803), Postgraduate research and innovation project of Inner Mongolia university of finance and economics.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Cheng, D; Qi, H. (2010): A linear representation of dynamics of Boolean networks. *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251-2258.

Gao, B.; Li, L.; Peng, H.; Kurths, J.; Zhang, W. et al. (2013): Principle for performing attractor transits with single control in Boolean networks. *Physical Review E*, vol. 88, no. 6, 062706.

Gao, B.; Peng, H.; Zhao, D.; Zhang, W.; Yang, Y. (2013): Attractor transformation by impulsive control in Boolean control network. *Mathematical Problems in Engineering*, vol. 2013, pp. 1-5.

Gao, B.; Shi, Y.; Yang, C.; Li, L.; Wang, L. et al. (2014): STP-LWE: a variant of learning with error for a flexible encryption. *Mathematical Problems in Engineering*, vol. 2014, pp. 1-7

Gao, B.; Deng, Z. H.; Zhao, D. W.; Song, Q. (2017): State analysis of Boolean control networks with impulsive and uncertain disturbances. *Applied Mathematics and Computation*, vol. 301, pp. 187-192.

Gao, B.; Liu, X.; Lan, Z.; Fu, R. (2018): A novel method for reconstructing period with single input in NFSR. *Chaos, Solitons & Fractals*, vol. 109, pp. 36-40.

Golomb, S. W. (1982): *Shift Register Sequences*. World Scientific Press.

Li, F.; Yan, H.; Karimi, H. R. (2018): Single-input pinning controller design for reachability of Boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3264-3269.

Han, W.; Tian, Z.; Huang, Z.; Li, S.; Jia, Y. (2018): Bidirectional self-adaptive resampling in internet of things big data learning. *Multimedia Tools and Applications*, no. 3, pp. 1-16.

Hell, M.; Johansson, T.; Meier, W. (2007): *Grain: A Stream Cipher for Constrained Environments*. Inderscience Publishers Press.

Li, F. (2016): Pinning control design for the stabilization of Boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1585-1590.

Li, S.; Wu, X.; Zhao, D.; Yang, Y. (2018): An efficient dynamic ID-based remote user authentication scheme using self-certified public keys for multi-server environments. *PLoS One*, vol. 13, no. 10, pp.1-19.

Massey, J.; Liu, R. (1964): Application of Lyapunov's direct method to the error-propagation effect in convolutional codes (Corresp.). *IEEE Transactions on Information Theory*, vol. 10, no. 3, pp. 248-250.

Mowle, F. J. (1966): Relations between PN cycles and stable Feedback Shift Registers. *IEEE Transactions on Electronic Computers*, no. 3, pp. 375-378.

Mowle, F. J. (1967): An algorithm for generating stable feedback shift registers of order n. *Journal of the ACM*, vol. 14, no. 3, pp. 529-542.

Ma, Z.; Qi, W. F.; Tian, T. (2013): On the decomposition of an NFSR into the cascade connection of an NFSR into an LFSR. *Journal of Complexity*, vol. 29, no. 2, pp. 173-181.

Qiu, J.; Chai, Y.; Liu, Y.; Gu, Z.; Li, S. et al. (2018): Automatic non-taxonomic relation extraction from big data in smart city. *IEEE Access*, vol. 6, pp. 74854-74864.

Riad, K.; Ke, L. (2018): Roughdroid: operative scheme for functional android malware detection. *Security and Communication Networks*, vol. 2018, pp. 1-10.

Wang, Z.; Liu, C.; Qiu, J.; Tian, Z.; Cui, X. et al. (2018): Automatically traceback RDP-Based targeted ransomware attacks. *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-13.

Xu, W.; Xiang, S.; Sachnev, V. (2018): A cryptograph domain image retrieval method based on Paillier Homomorphic block encryption. *Computers Materials & Continua*, vol. 55, no. 2, pp. 11-21.

Zeng, K.; Yang, C. H.; Wei, D. Y.; Rao, T. R. N. (1991): Pseudorandom bit generators in stream-cipher cryptography. *Computer*, vol. 24, no. 2, pp. 8-17.

Zhong, J.; Lin, D. (2015): A new linearization method for nonlinear feedback shift registers. *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 783-796.

Zhong, J.; Lin, D. (2016): Driven stability of nonlinear feedback shift registers with inputs. *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2274-2284.