Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.5, No.12, 2015
www.iiste.org

On the Comparative Study of Some Mathematical Tools for Specific Sequences Design

Cuong Nguyen Le[1] Thang Pham Xuan[1] Quynh Le Chi[2]
1.Electric Power University, 235 Hoang Quoc Viet Hanoi, Viet Nam
2.Van Lang University, 45 Tran Khac Nhu TPHCM, Viet Nam

Abstract
In modern communication system, cryptography and automatic test patterns, some specific sequences with strictly defined properties are required to meet the application demands. [1,2,3,4…] These requirements are:

- Good pseudorandom (PN) properties (large period length, uniform distrbution…).
- Low periodic correlation property.
- Low aperiodic correlation property.
- Large linear complexity.
- Large cardinality (number of sequences in the set).

Unfortunately, there is not any set of sequence satisfying all these requirements, despite the fact that a lot of efforts have been given for design such sequences. For this purpose, different mathematical tools have been widely used such as: matrix, d-transform and trace function representations. However, to the best of our knowledge, there has been no any comparative study on these mathematicals tools carried out so far!
In this contribution, we try to fill some parts of this gap by considering some typical applications of these tools. The paper is constructed as below:

- In the introduction, the basic concepts and definitions of matrix, d-transform and trace are given (briefly).
- In section II, some typical applications (for demonstration) will be shown. In this regard, we will give some discussions and suggestions for choosing the appropriate mathematical tools for each application.
- Especially, in section III, we will show the relationship (interchanging or equivalencies) between them.

Keywords: specific sequences, mathematical tool, matrix, d-transform and trace.

1. Introduction

1.1 Matrix and State-machine representation of LFSR

In this section, two typical configurations of linear feedback shift register for the same output i.e., the Galois and Fibonacci, are considered and showed as Fig 1.1a and Fig 1.1b, which can be used to generate the m-sequence for mobile communication and cryptography application [5,6]. The state at the time instant n is the contents of the shift register represented by a column vector S_n:

$$S_n = \left[S_{0,n} S_{1,n} \ldots S_{m-2,n} S_{m-1,n}\right]^T \quad (1)$$

Let consider the Fig 1.1a first. Given the contents of the LFSR at the time instant n, then the content of the shift register at instant (n+1) can be obtained:

$$\begin{aligned} S_{0,n+1} &= S_{1,n} + g_1 S_{0,n} \\ S_{1,n+1} &= S_{2,n} + g_2 S_{0,n} \\ &\vdots \\ S_{m-2,n+1} &= S_{m-1,n} + g_{m-1} S_{0,n} \\ S_{m-1,n+1} &= g_r S_{0,n} \end{aligned} \quad (2)$$

Where addition and multiplication are mod2 and $g_i \in \{0,1\}$ for i = {1,2,…,m}. These equations can be written in matrix form as:

$$S_{n+1} = \begin{bmatrix} S_{0,n+1} \\ S_{1,n+1} \\ S_{2,n+1} \\ \vdots \\ S_{m-1,n+1} \end{bmatrix} = \begin{bmatrix} g_1 & 1 & 0 & \ldots & 0 \\ g_2 & 0 & 1 & \ldots & 0 \\ g_3 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_m & 0 & 0 & \ldots & 0 \end{bmatrix} \times \begin{bmatrix} S_{0,n} \\ S_{1,n} \\ S_{2,n} \\ \vdots \\ S_{m-1,n} \end{bmatrix} \quad (3)$$

which can be written for short:

$$S_i = G^i S_0 \text{ for } i = \{1,2,\ldots,n\} \quad (4)$$

Where G is a square Galois state transition matrix of order m, S_0 is the vector of initial state.

The output b_n of the shift register at the given instant n (the content of the right most shift register) can be determined as:

$$b_n = [1 \quad 0 \quad 0 \quad \dots \quad 0] \times G^n \times S_0 \tag{5}$$

Which presents the relationship between output sequence and the state transition matrix and the initial state S_0 of the LFSR.

For the shift register configuration in Fig 1.1b, a similar result may be obtained in the same way as:

$$S_{n+1} = F \times S_n \tag{6}$$

Where:

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ g_m & g_{m-1} & g_{m-2} & g_{m-3} & \cdots & g_1 \end{bmatrix} \tag{7}$$

And the output b_n at the instant of time n is:

$$b_n = [g_m \quad g_{m-1} \quad \cdots \quad g_1] \times F^n \times S_0 \tag{8}$$

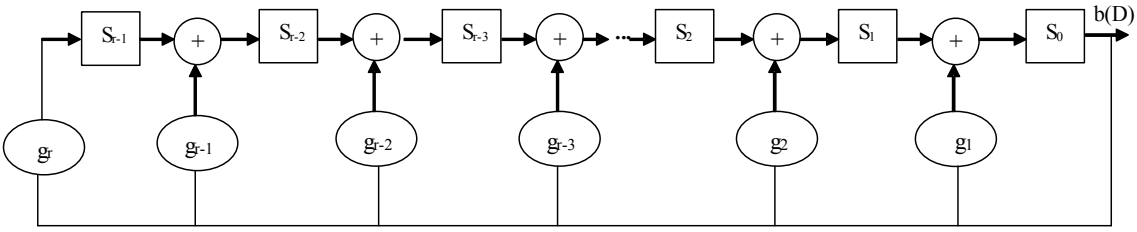

Figure 1.1a. Galois feedback generator.

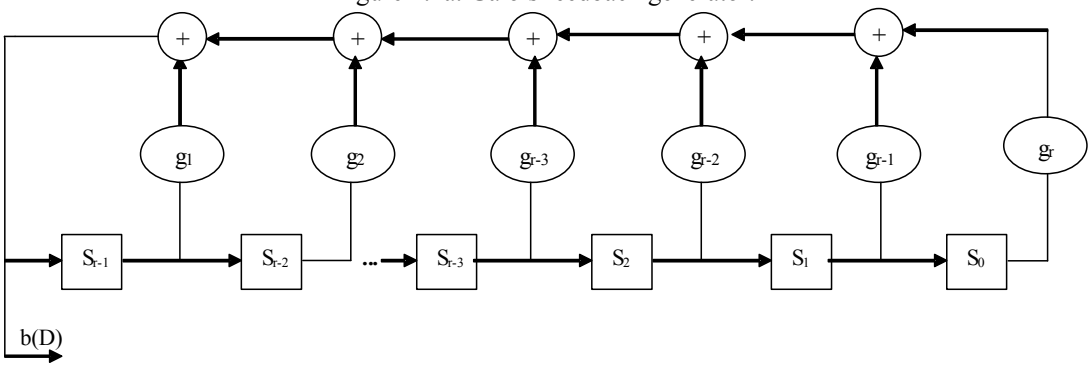

Figure 1.1b. Fibonaci feedback generator.

1.2 D-transform

The D-transform of a sequence $\{b_n\}$ over GF(p) (Galois filed (mod p), a primitive integer) is denoted by $D[b_n]$ [4,7,8,9]:

$$D[b_n] = \sum_{i=0}^{m} b_i D^i \ , \ b_i \in \{GF(p)\} \tag{9}$$

Thus, the D-transform of the sequence will have the form of a polynomial in d over GF(p) and has been conveniently used in signal and system analysis in data transmission and CDMA [8,9].

The D-transform of the generator sequence $\{b_n\}$ of a linear feedback shift register (LFSR) is then given by:

$$b(D) = \frac{S(D)}{g(D)} \tag{10}$$

Where g(D) of degree n is the generating polynomial of a LFSR and S(D) of degree ≤ n-1 specifies the initial condition corresponding to a particular shifted version of $\{b_n\}$.

1.3 Representation the m-sequence in term of the power of primitive elements (Trace function)

Let α be the root of the primitive polynomial g(D) of the degree m. The trace function of α is mapping from GF(q^m) to GF(q) and is defined as:

$$Tr_q^{q^m}(\alpha) = \sum_{i=0}^{m-1} \alpha^{q^i} \tag{11}$$

In other words, the trace of α over GF(q) is the sum of conjugate of α with respect to GF(q).

2. Application cases

2.1 Matrix representation

2.1.1 State and configure description

As clearly seen in 1.1 previous section, the matrix presentation may be best suitable to describle FSR in term of Finite State Machine (FSM). It shows the close relations in the set of four element $\{S_n, S_{n+1}, F, b_n\}$ which defines the FSM.

Observing the fig 1.1a, 1.1b one can easily get elements of the feedback tap g_n, state vector S_n and vice versa. In word: matrix presentation is no doubt the most intuitive.

2.1.2 Interleaved structure representation

It is well known that the specific sequences must have very large length. Except for few Mersenne primitive integers, the natutally posses interleaved structure which draws a lot of attentions in the literature [1,2,3,4,10,11…]. For this case, the matrix representation can offer a very intuitive approach in represent the interleaved structure [4,10,11…] but it may look complicated due to the great length of those sequences.

Decomposition of m-sequence $\{b_n\}$ and rearrange it Matrix form [3,4,12]. It is well known that if we decimate $\{b_n\}$ by T, we obtain:

$$\{a_n\} = \{b_{nT}\} = Tr_1^n(\alpha^{Tn}) = Tr_0^n(\beta^n) \tag{12}$$

With $\beta = \alpha^T$. Since α^T is primitive element in GF(2^m), $\{a_n\}$ is also an m-sequence of length $N=(2^n-1)/T$. When the decimation start at the first bit, we will obtain the subsequence: $\{a_0, a_T, \ldots, a_{(2^m-2)T}\}$

Similarly, we'll obtain the subsequence $\{a_{t-1}, a_{T+t-1}, \ldots, a_{(2^m-2)T+t-1}\}$ when the decimation start at the t^{th}-bit. Thus, on the time-domain, these subsequence (arrange in columns) can be considered as T time-multiplexed sequences $\{a_{nT}\}\{a_{nT+1}\}\ldots\{a_{(n+1)T-1}\}$ onto T time slots:

$$\underline{\underline{M}} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{T-1} \\ a_T & a_{T+1} & \cdots & a_{2T-1} \\ \vdots & \vdots & & \vdots \\ a_{(2^m-2)T} & a_{(2^m-2)T+1} & \cdots & a_{(2^m-1)T-1} \end{bmatrix} = \{a_{nT}\}\{a_{nT+1}\}\ldots\{a_{(n+1)T-1}\} \tag{13}$$

This representation is closely related to the time multiplex technique, which well understood by engineers.

Example 1: Decomposition of the sequence of 1023 bits long into 33 subsequences, each of length 31:

1001001001101001101011111100110001
1111001000111011111100001111000000
0111111111100011100010011110110010
1011101111010100011111010010101000
0010111111111010101010111101000011
1010010001100101101011001111101011
0001100111111001010101001100110011
0100111110100111000010001101100010
0010100110111101110101011100110011
1101110111001110101001110100000011
1101101110000110001001010010110011
1010001000101101001011101001100011
0110000001010010010111111011110001
1000110111011000011111001001110010
1100010000110111111100111000110100
1001010000100001001011011111101011
1000101110010000111111011010101000
1011110110011100111111000001110010
0101011001011110010111000001010111
0110011000011010110111010001010111
1111010001110011011100101000110100
0000011001001000100000100110110100
0111100110101011000010111011010000
1100001001111111101110001111000000
1110110110001010001001100100000011
0100100111101111100010101011010000
0101000000010110110111100111100001
0001111110110001110101101010000011
0011011000001100000000011011011010
1110101111000010101001000010110011
0011000001000100100000010000000000

Based on the phase shifts between the subsequences represented by the columns and define the Null sequence by ∞ we can easily get the interleaving order [4,9,12]

I_p^T={12,21,1,29,6,17,0,6,6,2,4,16,21,7,22,4,29,16,13,16,15,8,24,12,4,5,25,15,10,18,∞,17,8}

Note:

- This sequences is still too short for practical application.
- One can get the interleaving order I_p^T (shift sequence e) only after knowing the sequences.

2.2 D-transform representaion

2.2.1 State and configure description

Refer to (10) we see that in D-transform the information about configuration g(D), state S(D) and signal b(D) are given. It is short and effective to handle the relation between these three parameters. If S(D) and g(D) are known, we can find out the corresponding output sequence by the long division algorithm [5,13] as in example 2.

Example 2: Apply the long division algorithm to determinate shift register output when S(D)=1, $g(D)=1+D+D^4$

Solution:

The shift register output is:

$$b(D)=\frac{1}{1+D+D^4}$$

$b(D)=1+D+D^2+D^3+D^5+D^7+D^8+...$ and $b_n=1,1,1,1,0,1,0,1,1,...$

2.2.2 Interleaved structure representation

Since interleaving process and D-transform are both sort of time multiplexing [1, 3, 4, 9…] one can easily derive the interleaving order I_p^T straightforwardly from D-transform of the sequences. In fact, there are two methods for derive I_p^T, namely: expanding and decomposition. For the sake of simplicity we just show the result as in example 3.

Example 3: Let m=3, n=6 and let α be a primitive element of $GF(2^6)$ with primitive polynomial $b(D)=D^6+D^5+1$ over GF(2). Let $\{b_n\}$ denote the m-sequence generated by b(D)

$\{b_n\}$={0 1 1 1 1 1 1 0 1 .0 1 0 1 1 0 0 1 1 .0 1 1 1 0 1 1 0 1 .0 0 1 0 0 1 1 1 0 .0 0 1 0 1 1 1 1 0 .0 1 0 1 0 0 0 1 1 .0 0 0 0 1 0 0 0 0}.

Decimation of $\{b_n\}$ by T=9, we obtain $\{a_n\}=\{b_{n9}\}$ and rearrange is as:

$$M=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can see that the columns are shift equivalent and I_p^T={∞,5,3,5,6,3,3,2,5}, where ∞ represents Null sequence.

2.2.3 Hardware implementation of interleaving

D-transform can offer this representation, since it is time multiplexing from nature! It is clear that all m-subsequences are shift versions generated by same LFSR. Therefore, the software controlled switch S just picks up these shift versions and deliver them in the order determined by I_p^T… This approach have been successfully have been successfully simulated and implemented in [14,15,16].

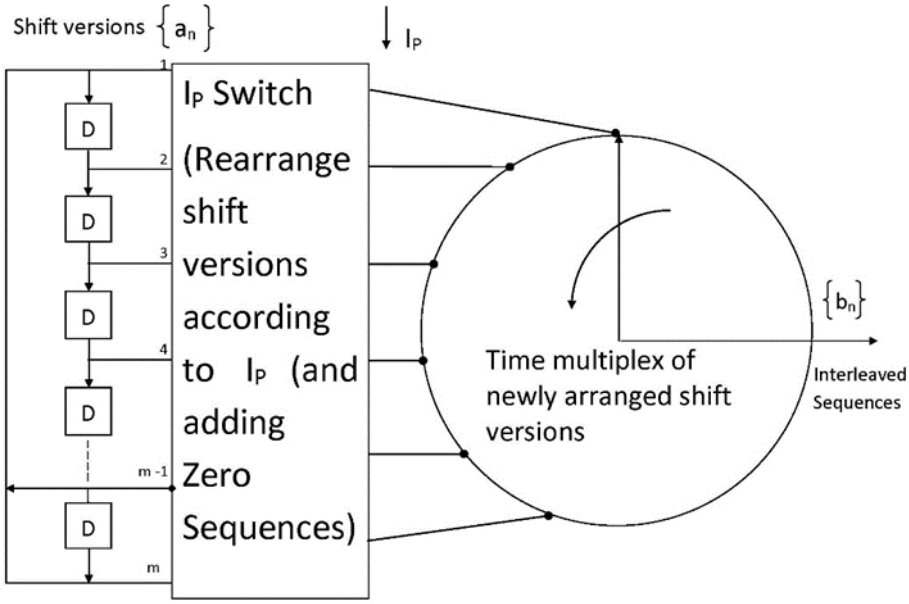

Figure 2. Shift versions for interleaving

2.3 Trace function representation

2.3.1 Field structure and sequences

As seen in the definition, this is also very short and compact representation: the information about the extension Field and subfield is given (mapping between them). One can derive the sequence from it trace function (see example 4).

Example 4: The binary m-sequence $\{a_n\}$ with period 2^m-1 can be represented through the trace function as:

$$\{a_n\}=\{a_0,a_1,a_2,...,a_{2m-1}\}=Tr_1^m(\alpha^0),Tr_1^m(\alpha^1),...,Tr_1^m(\alpha^{2^m-1})=Tr_1^m(\alpha^n)$$

It is well known that, the shift version of sequence $\{a_n\}$ can be created bu its decimation.

The sequence $\{a_{kn}\}$ obtained by sampling every k^{th} bit of $\{a_n\}$, starting from the first bit is referred to as decimation of $\{a_n\}$. A decimation by k of $\{a_n\}$ results in the sequence: $\{a_{nk}\}=Tr_1^m(\alpha^n)$.

Let $\{a_n\}$ be m-sequence generator by $g(D)=D^5+D^2+1$ over GF(2).

The corresponding binary from is: 10010.11001.11110.00110.11101.010000

Below are few sequences result from decimation of $\{a_n\}$ by difference k [17]

k	Sequence	Characteristic Polynomial (in octal)
1	10010110011111000110111010100 00	45 47 61 56 50 0
3	11111011100010101101000011001 00	76 70 53 20 62 0
5	11101000100101011000011100110 11	72 11 26 07 15 4
7	10010011000010110101000111011 11	44 60 55 21 67 4
11	11101100111000011010100100010 11	73 16 06 51 05 4
15	10000101011101100011111001101 00	41 27 30 76 32 0

2.3.2 Interleaved structure representation

By nature, trace function shows the "interleaving" of subfileds inside the extension field. Therefore it is convenient to describle the interleaved structure.

Base on trace function one can easily derive the interleaving order directly as in example 5.

Example 5: Let trace function from GF(3^4) onto GF(3^2) with primitive polynomial 10021: f(x)=x^4+2x+1 with n=4, L=3^4-1=80, m=2, N=3^2-1=8, S=L/N=10.

Calcualting the trace function of x from GF(3^4) into GF(3^2) we obtained:

$$Tr_m^n(\alpha)=\sum_{k=0}^{n/m-1}\alpha^{3^{2k}}=\alpha+\alpha^9$$

α^{Si} table:

$$i=0\Rightarrow\alpha^0=1$$

$$i=1\Rightarrow\alpha^{10}=1+2\alpha+\alpha^2+\alpha^3$$

$$i=2\Rightarrow\alpha^{20}=\alpha+2\alpha^2+2\alpha^3$$

$$i=3\Rightarrow\alpha^{30}=1+\alpha+2\alpha^2+2\alpha^3$$

$$i=4\Rightarrow\alpha^{40}=2$$

$$i=5\Rightarrow\alpha^{50}=2+\alpha+2\alpha^2+2\alpha^3$$

$$i=6\Rightarrow\alpha^{60}=2\alpha+\alpha^2+\alpha^3$$

$$i=7\Rightarrow\alpha^{70}=2+2\alpha+\alpha^2+\alpha^3$$

With j run from 0 to S-1 we get:

$$j=0\Rightarrow Tr(\alpha^0)=Tr(1)=1+1=2=\alpha^{40}\Rightarrow I_P^0=4$$

$$j=1\Rightarrow Tr(\alpha^1)=\alpha+\alpha^9=2\alpha+\alpha^2+\alpha^3=\alpha^{60}\Rightarrow I_P^1=6$$

$$j=2\Rightarrow Tr(\alpha^2)=\alpha^2+\alpha^{18}=2\alpha+\alpha^2+\alpha^3=\alpha^{60}\Rightarrow I_P^2=6$$

$$j=3\Rightarrow Tr(\alpha^3)=\alpha^3+\alpha^{27}=\alpha+2\alpha^2+2\alpha^3=\alpha^{20}\Rightarrow I_P^3=2$$

$$j=4\Rightarrow Tr(\alpha^4)=\alpha^4+\alpha^{36}=2+\alpha+2\alpha^2+2\alpha^3=\alpha^{50}\Rightarrow I_P^4=5$$

$$j=5\Rightarrow Tr(\alpha^5)=\alpha^5+\alpha^{45}=0\Rightarrow I_P^5=\infty$$

$$j=6\Rightarrow Tr(\alpha^6)=\alpha^6+\alpha^{54}=\alpha+2\alpha^2+2\alpha^3=\alpha^{20}\Rightarrow I_P^6=2$$

$$j=7\Rightarrow Tr(\alpha^7)=\alpha^7+\alpha^{63}=1=\alpha^0\Rightarrow I_P^7=0$$

$$j=8\Rightarrow Tr(\alpha^8)=\alpha^8+\alpha^{72}=2+\alpha+2\alpha^2+2\alpha^3=\alpha^{50}\Rightarrow I_P^4=5$$

$$j=9\Rightarrow Tr(\alpha^9)=\alpha^9+\alpha^{81}=2\alpha+\alpha^2+\alpha^3=\alpha^{60}\Rightarrow I_P^1=6$$

The interleaving order is:

$$I_p^T=\{4,6,6,2,5,\infty,2,0,5,6\}$$

The subsequences can be interleaved like this [3]

$$M=\begin{bmatrix}1&0&0&0&1&0&0&2&1&0\\1&1&1&2&0&0&2&2&0&1\\0&2&2&1&1&0&1&0&1&2\\1&2&2&1&2&0&1&2&2&2\\2&0&0&0&2&0&0&1&2&0\\2&2&2&1&0&0&1&1&0&2\\0&1&1&2&2&0&2&0&2&1\\2&1&1&2&1&0&2&1&1&1\end{bmatrix}$$

Even though trace function is very effective, it is abstract (pure mathematic) and one cannot expect it to be intuitive. Note: It has shown that the interleaving order (or shift sequence) calculated by D-transform and by trace representation are identical for the same composite sequence [4,9,12].

3. The links between above mentioned tools

3.1 Between Matrix and D-transform

In order to determinate S(D) (in D-transform) which specifies the corresponding memory content of LFSR we have use the matrix equation system. This fact can be best demonstrated by an example. Unfortunately the matrix representation for specific sequences which have very large length would take a lot of spaces, let alone the manipulation with these matrices. Therefore we consider here only the subsequence (shorter length) P_n generated by:

$$g(D)=1+D^5+D^6$$

Example 6: The D-transform of P_n is $W_N(D)$ with $N=2^m-1=63$, so, the length of P_n is then=63 and

$$W_i(d^T)=\frac{S_i(d^T)}{g_S(d^T)}$$

With T denotes the number of subsequences in the specific sequence.
The corresponding $S_i(d^T)$ of memory state: S_i can be calculated as:

$$\begin{pmatrix}p_0\\p_1\\\vdots\\p_{N-1}\end{pmatrix}=\begin{pmatrix}1&0&\cdots&0\\w1&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\w_{N-1}&w_{N-2}&\cdots&1\end{pmatrix}\begin{pmatrix}g_0\\g_1\\\vdots\\g_{N-1}\end{pmatrix}$$

And $[g_i]$ denote the feedback taps of the LFSR characterized by g(D).

Table i the corresponding phase shifts of sequences specified by $S_i(d^T)$

phase shifts	$S_i(d^T)$	phase shifts	$S_i(d^T)$
0	$1+ D^5$	32	$1+D^3 +D^5$
1	D^5	33	$D^3 +D^5$
2	D^4	34	$D+D^4$
3	D^3	35	$1+D^3$
4	D^2	36	$D^2 +D^4 +D^5$
5	D	37	$D +D^3 +D^5$
6	1	38	$1+D^2 +D^3$
7	D^4+D^5	39	$D+D^2 +D^4 +D^5$
8	D^3+D^4	40	$1+D+D^3 +D^4$
9	D^2+D^3	41	$1+D^2 +D^3 +D^4 +D^5$
10	$D+D^2$	42	$D +D^2 +D^3 +D^5$
11	1+D	43	$1+D +D^2 +D^4$
12	$1+D^4+D^5$	44	$1+D +D^3 +D^4 +D^5$
13	D^3+D^5	45	$1+D^2 +D^3 +D^5$
14	D^2+D^4	46	$D +D^2 +D^5$
15	D^1+D^3	47	$1+D +D^4$
16	$1+D^2$	48	$1+D^3 +D^4 +D^5$
17	$D+D^4+D^5$	49	$D^2 +D^3 +D^5$
18	$1+D^3+D^4$	50	$D +D^2 +D^4$
19	$D^2+D^3+D^4 +D^5$	51	$1+D +D^3$
20	$D+D^2+D^3 +D^4$	52	$1+D^2 +D^4 +D^5$
21	$1+D+D^2 +D^3$	53	$D +D^3 +D^5$
22	$1+D+D^2 +D^4 +D^5$	54	$1+D^2 +D^4$
23	$1+D+D^3 +D^5$	55	$D+D^3 +D^4 +D^5$
24	$1+D^2 +D^5$	56	$1+D^2 +D^3 +D^4$
25	$D+D^5$	57	$D+D^2 +D^3 +D^4 +D^5$
26	$1+D^4$	58	$1+D+D^2 +D^3 +D^4$
27	$D^3 +D^4 +D^5$	59	$1+D+D^2 +D^3 +D^4 +D^5$
28	$D^2 +D^3 +D^4$	60	$1+D+D^2 +D^3 +D^5$
29	$D+D^2 +D^3$	61	$1+D+D^2 +D^5$
30	$1+D+D^2$	62	$1+D +D^5$
31	$1+D +D^4 +D^5$		

Note that T may be few thousands or more to create the specific sequences used in the practice!
On the other hand from g(D) we can easily get the feedback tap of fig 1.1a, 1.1b and therefore G_i of (4), the state transition matrix F of (7). If S(D) is known, then we can get the output b_n.

3.2 Between D-transform and trace

The equivalence can be proven as follows:
Let consider the Decomposition of m, sequences $\{b_n\}$
Let $\{b_n\}$ be an m-sequence generated by g (D) of degree n = lm, and denote $T=(2^n-1)/(2^m-1)$
From $\{b_n\}$ sequences, we try to find out $\{a_n\}$ and I_p^T

Decimate $\{b_n\}$ by T, we will obtain: $\{a_n\}= Tr_1^n(\alpha^{Tn}) = Tr_0^n(\beta^n)$
Since (α^{Tn}) and (β^n) are primitive element in GF(2^n), $\{a_n\}$ is also an m-sequence of length N with $N=(2^n-1)/T$.
Let arrange $\{a_n\}$ in the column of the decimation matrix

$$M = \begin{vmatrix} a_0 & a_1 & \cdots & a_{T-1} \\ a_T & a_{T+1} & \cdots & a_{2T-1} \\ a_{2T} & a_{2T+1} & \cdots & \\ a_{(2^m-2)T} & a_{(2^m-2)T+1} & \cdots & a_{(2^m-1)T-1} \end{vmatrix}$$

Each column is a decimation of $\{b_n\}$ by T, an m-subsequence [17].
On other hand, from d-transform property: $S(d^T)/g(d^T)$ represents a decimations of b_n by $T=(2^n-1)/(2^m-1)$, which turn out to be $\{a_n\}$, an m-sequence also because: $S(d^T)$ and $g(d^T)$ are relative prime.
So, the subsequences in the interleaving are identical. Now, we need only to check I_p^T. It can be seen that I_p^T

calculated by [3] or lookup table [12] (similar to table 1, example 1) are identical also (they must be the same because they are derived from matrix M). That implies D-transform and trace function are equivalent in creating the sequences with interleaved structure.

4. Conclusion remarks and future works

In this contribution we try to compare some mathmatical Tools widely used in representation of specific sequences applied in such areas like: advanced communications, cryptography and automatic testing.

The matrix representation is very intuitive and easily understandable in describling the configuration, state transition. It can also be used for determination the particular memory content when the D-transform of either the sequences or the corresponding initial state are given. However, for the sequences of great length, the dimension of the Matrix is unbearable great, which makes the manipulation difficult.

The trace function is compact and very comfortable for investigation of the m-sequences in general and interleaved m-sequences in particular. Since the length of sequences defined in trace function is $L=p^m-1$ only, it cannot be used for an arbitrary interleaved sequences ($L\neq p^m-1$) without some complicated modification as in D-transform case. Furthermore, one cannot get the information about states and configuration of LFSR in trace representation. However, from shift sequence ($e=I_p^T$) one can also figure out the hardware implementation (interleaving order).

The D-transform representation is also short and easy to handle. Furthermore, it contains information about the state (S(D)) and configuration, g(D) as well. Like Trace function representation, it also indicates the hardware implementation (I_p^T). We notice once more that D-transform can be applied for any periodic sequence without any new concept. We hope to prove this statement in coming papers!

The link between those tools are also pointed out, for the first time.

As mentioned in our introduction and [1,2,3,4,…] all the above requirements need to be carefully evaluated before application. The static properties of the sequences (correlation functions, symbol distribution…) have to satisfy some random criteria so that the processed signal can be seen as pseudo noise and the linear complexities have to be large enough to make the sequence estimation difficult [4,11,12,18,19,20]. The evaluation process is obviously more complicated so that the trace function and D-transform need to be modified correspondingly. However, due to the limited scope of this paper we concentrate only on the comparison of represenration methods and leave other issues such as analysis and sequence selection for the future contributions.

The authors express their deep sense of gratitude toward the reviewers for their constructive comments. Thanks also given to engineers Phuong n m, Thang l m for their help in checking the paper's format and drawing the pictures.

References

[1] Fan.P.Z and Darnell.M (1996), "Sequence Design for Communications Applications", New York: Wiley, 1996.
[2] Golomb S. W and. Gong. G (2005) "Signal Design for Good Correlation - for Wireless Communication, Cryptography and Radar", Cambridge University Press, 2005.
[3] Lin. X.D and Chang K.H, (1997): "Optimal PN Sequence Design for Quasi synchronous" IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 45, NO. 2, FEBRUARY 1997 p 222-226.
[4] Hieu l .M et al ((2015) "Design and Analysis of Ternary m-sequences with Interleaved Structure by d-Transform", Journal of Information Engineering and Applications 8/2015 p 93-101
[5] Peterson R.L.et al (1995): "Introduction to spread spectrum communication". Prentice Hall International 1995.
[6] Weisun. A, Klapper, Y Yang.X(2001): "On the correlation of a family of generalized geometric sequences", IEEE Trans, Information theory, vol47, No6 September 2001, pp 2609-2618.
[7] Gill A (1996), "Linear sequential circuits", McGraw-Hill New york1996.
[8] Gitlin R.G & Hayer J. F (1975), "Timing recovery and scramblers in data transmission", Bell. Syst. Tech Journal, vol54, no3, pp 589-593, March 1975.
[9] Quynh .L.C, Prasad. S (1985): "A class of binary cipher sequences with best possible correlation function". IEEE Proceeding Part F .Dec 1985. Vol 132 pp.560-570
[10] An B.l (2013) on multi dimension cascaded interleaved sequences PhD thesis RIPT Hanoi 2013
[11] He. J (2013) interleaved sequences over Finite Field PhD thesis Carleton University Ottawa, Ontario 2013
[12] Hieu L.M &. Quynh L.C (2005): "Design and Analysis of Sequences with Interleaved Structure by d-Transform", IETE Journal of Research, vol. 51, no. l, pp.61-67, Jan-Feb. 2005.
[13] Zigangirov .k.sh (2002) Theory of code division multiple access John Wiley &son ltd 2002
[14] Lam. N.V (2004) Master Thesis Hanoi Technology University 2004.
[15] Cuong. N. l (2007) Master Thesis Military technology University 2007

[16] Thanh N.V (2010) Master Thesis Hanoi Technology University 2010
[17] Simon M, K et al (2002) Spread spectrum communications Handbook McGraw-Hill 2002.
[18] Key E. L, "An analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators", IEEE Trans. Inform. Th., vol. IT-22, pp. 732- 736, Nov. 1976.
[19] Prasad.S, Quynh L.C, "Equivalent linear span analysis of binary sequences having interleaved structure", i.ee proceedings Vol 133,part F No3 june 1986 p 288-291
[20] Helleseth. T et al (2012) State Space Cryptanalysis of The MICKEY Cipher University of Bergen Netherlands 2012.