

카오스 패턴 발생기의 구현

Implementation of chaotic pattern generator

박광현*, 박희윤, 황종선, 남호윤
한국원자력연구소

Park Kwang-Hyeon*, Park Hyi-Yun, Hwang Jong-Sun, and Nam Ho-Yun
Korea Atomic Energy Research Institute

요 약

카오스 함수인 베이커 함수를 이용하여 불규칙한 상태에 의해 카오스적인 패턴을 발생시키는 패턴 발생기를 구현하였다.

카오스적인 패턴 발생기는 예측할 수 있는 패턴들을 발생시킨다는 점에서 기존의 패턴 발생기와 흡사하지만, 패턴요소의 출력 순서는 결정론적 카오스 함수에 따른다.

I. 서론

카오스 역학(chaotic dynamics)을 디지털시스템의 순차시스템(sequential system)에 연관시키기 위하여 결정론적 카오스 함수(deterministic chaos function)인 베이커 함수(Baker's function)를 이용하여^{[1][2]} 카오스적인 패턴 발생기(chaotic pattern generator)를 구현하였다.

기존의 패턴 발생기(conservative pattern generator)는 자율시스템(autonomous system)에 속하므로, 입력순서(input sequence)에 종속되지 않고 미리 정의된 출력순서(predefined output sequence)를 순차적으로 만들어 낸다^{[3]-[6]}.

카오스시스템(chaotic system)도 역시 자율시스템이므로, 카오스적인 거동(chaotic behavior)을 내보이는 카오스적인 패턴 발생기는, 기존의 패턴 발생기와 유사하게 미리 예측된 출력들을 만들지만, 패턴의 출력순서는 결정론적 카오스 함수에 따른다^[7].

이 논문에서는 먼저 기존의 패턴 발생기에 관한 정의와 기술에 대해 살펴보고 나서, 카오스적인 패턴 발생기를 정의하고, 구현하며, 카오스 특성을 증명한다.

II. 카오스적인 패턴 발생기(chaotic pattern generator)

A. 기존의 패턴 발생기(conservative pattern generator)
기존의 패턴 발생기는 입력순서와 무관하게 미리 정의된 출력순서를 만들어 내므로, 자율시스템(autonomous system)으로 분류된다.

패턴 발생기는 정해진 순차에 따라 클럭펄스(clock pulse)에 의해 주기적으로 패턴을 발생시키는 연속패턴 발생기(continuous pattern generator)와 패턴발생을 요구하는 신호가 있을 때마다, 패턴을 정해진 순서에 따라 순차적으로 발생시키는 단발패턴 발생기(one-shot pattern generator)로 크게 나눌 수 있다^[4].

즉, 두가지 형태의 거동(two types of behavior)으로 패턴을 발생시킬 수 있다.

연속적으로 패턴을 발생시키는 경우의 시간거동(time behavior)은 일반적으로 다음 식과 같이 표현된다.

<시간거동 표현식(time behavior expression)>

$$Z(rK, (r+1)K-1) = P$$

여기서, $P = (P_0, \dots, P_{K-1})$ 는 길이 K 의 패턴이고, r 은 반복횟수.

단발적으로 패턴을 발생시키는 경우에 있어서의 시간거동은 다음과 같다.

<시간거동 표현식>

$$\begin{cases} Z(t, t+K-1) = P, & \text{if } x(t) = \text{start} \\ Z(i) = 0, & \text{if } x(t-i) \neq \text{start for } i=0, \dots, K-1. \end{cases}$$

여기서, 시작신호(start signal)가 단발패턴을 요구하는 신호.

앞의 두가지 거동형태 모두가, 정해진 순차에 의해 결정된 상태순서들(state sequences) 안에서, 순차적으로 출력을 발생시킨다는 점은 같지만, 한 거동의 패턴발생은 연속적이고 다른 거동은 단발적이다.

두가지 패턴발생 절차를 자율(autonomous)로 인식하는데에는 별문제가 없으므로, 다음 절 B.에서 제시할 카오스적인 패턴 발생기와 비교를 쉽게 하기 위해서, 이 절 A.에서는 단발 패턴 발생기에 대해서만, 시간거동표현식(time behavior expression), 상태기술(state description)과 상태전이(state transition)를 정의한다(물론, 카오스적인 패턴 발생기에서 연속 반복되는 패턴을 발생시키기 위해서는 시작신호(start signal) 대신에 클럭신호(clock signal)를 주면 된다).

[단발 패턴 발생기의 정의]

<시간거동 표현식>

$$\begin{cases} Z(t, t+K-1) = P, & \text{if } x(t) = \text{start} \\ Z(i) = 0, & \text{if } x(t-i) \neq \text{start for } i=0, \dots, K-1. \end{cases}$$

여기서, 시작신호(start signal)가 단발패턴을 요구하는 신호.

<상태기술>

단발 패턴 발생기는, 0에서 $K-1$ 까지의 순차적으로 번호가 주어진 K 개의 상태들(states)을 갖는다.

<상태전이>

상태전이는 다음 그림 1에서 보인 상태도(state diagram)에 해설된 바와 같이, 시작신호가 주어질 때, 한번

의 패턴이 상태순서에 따라 자율적인 양식(autonomous fashion)으로 변화한다. 상태 i 에 해당하는 출력(output)은 P_i 이다.

B. 카오스적인 패턴 발생기(chaotic pattern generator)

다음 식과 같이 정의되는 결정론적 카오스 함수인 베이커 함수를 이용하여,

$$S(x) = \begin{cases} 2x & \text{if } x < \frac{\text{단위구간}}{2} \\ 2x-1 & \text{if } x \geq \frac{\text{단위구간}}{2} \end{cases}$$

여기서, 단위구간(unit interval)은 $[0, 1)$ 이고, " $)$ "은 1을 포함하지 않는 것을 의미.

카오스적인 패턴 발생기를 기존의 단발패턴 발생기에 준하여 정의한다.

<시간거동 표현식>

$$\begin{cases} Z(t, t+K-1) = P, \text{ if } x(t) = \text{start} \\ Z(t) = 0, \text{ if } x(t-i) \neq \text{start for } i=0, \dots, K-1. \end{cases}$$

여기서, 시작신호(start signal)는 한벌의 패턴을 요구하는 신호이며, P는 길이 K의 패턴.

<상태기술>

카오스적인 패턴 발생기의 상태들의 순서(sequence of states)는 예측할 수는 있지만, 정해지지 않았으며 다음과 같은 상태전이함수(state transition function)에 따른다.

$$S(t+1) = D(S(t))$$

여기서, D는 결정론적 카오스 함수.

따라서 출력함수(output function)는 다음과 같이 정의된다.

$$Z(t) = H(S(t)) = P$$

여기서, H는 $S \times I \rightarrow 0$ 인 함수이며, S는 상태들의 셀(set), I는 입력셀, 0는 출력셀이다.

<상태전이>

상태전이는 다음 그림 2에 보인 상태도와 같이 상태전이함수에 따른다. 시작신호가 주어지면 상태들의 순서는 베이커 함수의 변환(transformation of Baker's function)에 의해, 자율적인 양식으로 결정된다.

즉, 상태들의 순서는 초기화(initialization)와 베이커 함수변환에 따른다.

C. 카오스적인 패턴 발생기의 구현

앞절의 카오스적인 패턴 발생기에 관한 상태전이함수와 출력함수를 이용하여 전형적인 구현(canonical implementation)을 그림 3에 보인다.

하나의 예로써, 그림 4에 보인 카오스적인 4-비트 패턴 발생기는, 베이커 함수변환을 수행하는 하드웨어 소자로 시프트(shift)를 사용하여 구현하였으며, 16개의 패턴요소셀(set of pattern elements $P=(P_0, \dots, P_{15})$ 중 에서, 길이 4비트인 임의의 패턴 즉, 카오스적인 패턴을 발생시킨다.

III. 카오스적인 패턴 발생기의 카오스 특성 증명

카오스적인 패턴 발생기는, 정량적인 현상들(qualitative phenomena)이 전형적인 카오스로 판명된 베이커 함수의 변환을 이용하여, 패턴요소를 갖고 있는 상태들의 순서를 결정함으로써, 카오스적인 패턴을 발생시킨다.

따라서, 베이커 함수 변환을 수행하는 시프트의 동작이 어떻게 카오스 역학(chaotic dynamics)의 특성들을 만족시키는가를 살펴 봄으로써 패턴 발생기의 카오스적인 특성을 증명한다.^{[1][2][8][9]}

A. 초기조건에의 민감성(Sensitive dependence on initial condition)

많은 결정론적 카오스 함수들이 카오스 거동을 나타낼 수 있지만 이 논문에서는 베이커 함수를 이용한다. 초기조건에 의한 민감성의 특성은 카오스 역학의 핵심으로, 카오스 상태머신의 초기조건에의 민감성을 설명하기 위해 우선 먼저 베이커 함수변환을 살펴본다. 다음과 같은 원래의 정의와 다른 베이커 함수에 관한 새로운 표기법부터 살펴보자.

$$\text{Frac}(X) = X - k \text{ if } k \leq X < k+1, \text{ } k \text{ integer}$$

와 같이 기술되며, 베이커 변환을 새로운 표기법으로 사용하여 정의하면

$$S(X) = \text{Frac}(2X) \text{ for } 0 \leq X \leq 1$$

으로 표현될 수 있고, 단위구간에서의 실수 X 또는 임의의 십진수를 이진수 표현에 의해 새롭게 해석할 수 있다^[6].

예를 들면, $1/2 = 0.1, 3/4 = 0.11, 1/3 = 0.010101 \dots = 0.01(\dots)$ 는 주기적인 반복을 나타냄, $7 = 111, 12 = 1100$, 등으로 표현된다. 이와 같이 이진수로 표현했을 때, 베이커 변환은 시프트동작(shift operation)으로 간주할 수 있으므로 방정식 $S(X) = \text{Frac}(2X)$ 에서의 오른쪽 항에 존재하는 두배 곱셈은, a_k 가 이진수 즉 0 또는 1 인 경우, $0, a_1 a_2 a_3 \dots a_k$ 에서 $a_1, a_2 a_3 \dots a_k$ 으로 되는 것을 의미한다^[6].

더우기, 만약 단위구간에서 실수 X의 이진 표현식에서 이진점(binary point)을 제거한다면 실수 X의 이진 표현식과 십진 표현식 간의 차이는 없다. 이진점이 없는 이진 표현식으로 그림 4에 보인 회로의 상태레지스터(state register)로 초기화한다.

이제, 시프트의 시프트동작에 의한 민감성을 검토하기 위해 어떤 변수들의 이진 전개(binary expansion)를 생각해 보자.

x, y 를 $x=1110$ 및 $y=1101$ 의 4 비트 이진 표현으로 된 십진수라 놓고, 표 1과 표 2에서, 그림 4에 보인 4 비트 시프트레지스터 동작을 비교 설명한다.

표 1과 표 2를 비교하면, x 와 y 에서 시작한 시프트동작의 결과가 결국에는 어느 정도의 한계값(threshold)만큼 차이가 있는 y 에 근접하는 x 가 존재함을 알 수 있다. 이 한계값은 십진수를 표현한 이진수의 비트수가 같은 한 이진 표현의 모든 실수 x 에 대해 동일하며, 민감상수(sensitivity constant)라 부른다.

그러나 민감성의 현상은 1 비트 편차에 의해서도 항상 확대된다. 표들과 설명으로 카오스적인 패턴 발생기의 민감성에 대한 증명은 완료되었다.

B. 밀집된 주기점(Dense of periodic points)

그림 4의 상태 레지스터의 초기상태를 다음과 같이 지정해 보자.

$$x = a_1 a_2 a_3 \dots a_8$$

의심할바 없이 8번째 왼쪽 시프트동작후에 8-순환 상태 반복을 갖게된다. 또한, 초기상태를 다음과 같이 지정하면

$$y = a_1 a_2 a_3 \dots a_k$$

왼쪽 시프트 동작에 의해 모든 2^k 부구간(subinterval)에서 주기점들을 발견할 수 있다. 더욱 중요한 것은 임의의 주어진 x 에 대해, 시프터에 의해 주기적이며, x 에 임의적으로 근접하는 y 를 찾을 수 있다.

이제, 그 수들이 어떻게 밀집된 주기점들로 되는지 알아본다.

어떤 n 에 대해 $x = a_1a_2a_3 \dots a_{n-1}a_n$ 이면, 어떤 k 에 대해 $y = a_1a_2a_3 \dots a_{k-1}a_k^*$ 이 되도록 선택한다.

$$a^* = \begin{cases} a=0 \text{ 이면 } 1 \\ a=1 \text{ 이면 } 0 \end{cases}$$

여기서 $n=k$ 이고 a^* 는 이중 이진수(dual binary digit)이다. 즉,

그러면 x 와 y 는 기껏해야 2^k 정도만이 차이가 나고 y 는 주기적이다. 이것은 주기점들이 시프트동작에 대해 밀집된 것을 의미한다.

표 1과 표 2를 사용해서 밀집된 주기점들에 대한 예를 그림 5에 나타내었다.

C. 믹싱(Mixing)

믹싱의 특성은 임의의 지점에서 모든 지점에 도달할 수 있는 것을 말한다. 다음과 같이 두개의 초기값을 선택하자.

$$x = 1110, y = 0111.$$

다음에는 왼쪽 시프트동작을 정확히 3번 반복하면 y 와 같아지는 x 로 상태 레지스터를 초기화 한다. 베이커 변환(시프트동작)의 경우에는, 시간 t 의 경과에 따라 그림 5의 (b)에서 처럼 목표 값과 상태에 도달할 수 있다.

이제, A, B, 및 C에 의해 카오스적인 패턴 발생기의 3가지 특성이 증명되었다.

IV. 결론

디지털 순차시스템에 카오스 역학을 적용하기 위해 결정론적 카오스 함수변환을 이용하여 카오스적인 패턴 발생기를 구현하였다.

이 논문에서는 시프트레지스터를 이용하여 베이커 변환을 수행함으로써 불규칙적으로 패턴요소를 선정하였으나, 텐트함수(tent function)와 같은 1차이상의 카오스적인 함수들로의 대치도 고려해 볼 수 있다.^{[2][10][11]}

카오스적인 패턴 발생기에 보다 복잡한 카오스 함수변환을 사용하여 설계한다면 훨씬 다양한 카오스적인 거동을 갖을 것이 분명하기 때문이다.

따라서 디지털시스템에 속하는 여러 회로들에 필요 용도에 따라 알맞는 카오스 역학을 적용시켜 나갈 수 있으리라 생각한다.

참고문헌

- [1] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe, "Chaos and Fractals", Springer-Verlag Newyork, Inc., pp. 509-583, 1992.
- [2] Heinz Georg Schuster, "Deterministic Chaos", Weinheim: VCH Verlags gesellschaft mbH, pp. 1, pp. 21-36, 1989.
- [3] ZVI KOHAVI, "Switching and finite automata theory", Newyork: McGraw-Hill, Inc., pp. 275-315, 1978.
- [4] Milos D. Ercegovac and Tomas Lang, "Digital sys-

tems and hardware/firmware algorithms", Newyork: John Wiley & Sons, Inc., pp. 269-381, pp. 393-407, 1985.

- [5] Goger C. Alfred, "Programmable Logic Designer's Guide", Howard W. Sams & Company, pp. 34-45, 1989.
- [6] Anthony Palson and Edwin D. Reilly, "Encyclopedia of computer science", Newyork: Van Nostrand Reinhold, pp. 1191-1193, 1993.
- [7] K.H. Park et al., "A State Machine using Deterministic Chaotic Function", Proc. of KITE Fall Conf. '95, pp. 741-744, 1995.
- [8] Denny Gulick, "Encounters with Chaos", Newyork: McGraw-Hill, Inc., pp. 96-103, 1992.
- [9] R.L. Devaney, "An Introduction to Chaotic Dynamical Systems", Addison-Wesley, Redwood City, 1989.
- [10] Heinz-Otto Peitgen et al., "FRACTALS", Springer-Verlag Newyork, Inc., pp. 71-99, 1991.
- [11] Joseph L. McCauley, "Chaos, dynamics, and fractals", Cambridge University Press, pp. 278-308, 1993.

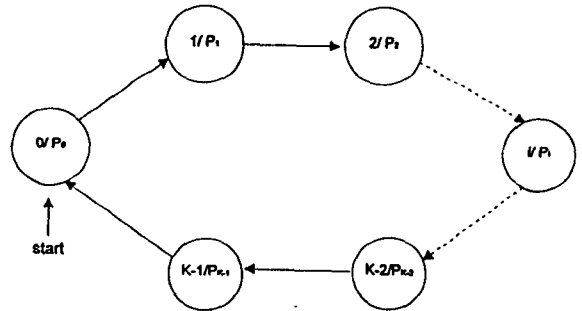


그림 1. 기존의 단발 패턴 발생기의 상태도
Fig. 1. State diagram of one-shot pattern generator.

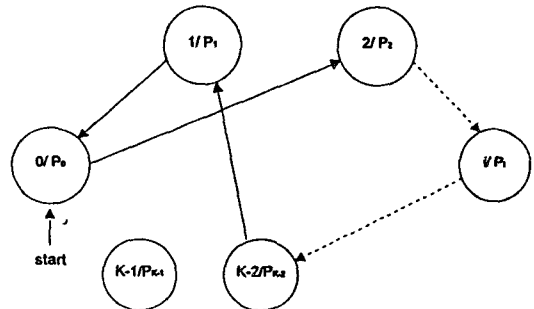


그림 2. 카오스적인 패턴 발생기의 상태도
Fig. 2. State diagram of chaotic pattern generator.

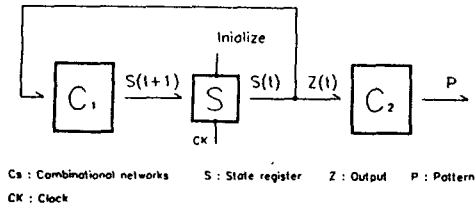
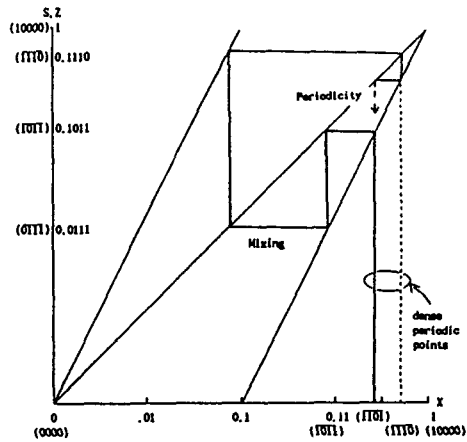


그림 3. 카오스적인 패턴 발생기의 전형적인 구현
 Fig. 3. Canonical implementation of chaotic pattern generator.



(b) Cycle of periodic 4 for initial state $y=1101$
 그림 5. 두가지 주기-4 순환을 이용한 믹싱과 밀집된 주기점들에 관한 설명
 Fig. 5. Illustration of dense periodic points and mixing by using period-4 cycles.

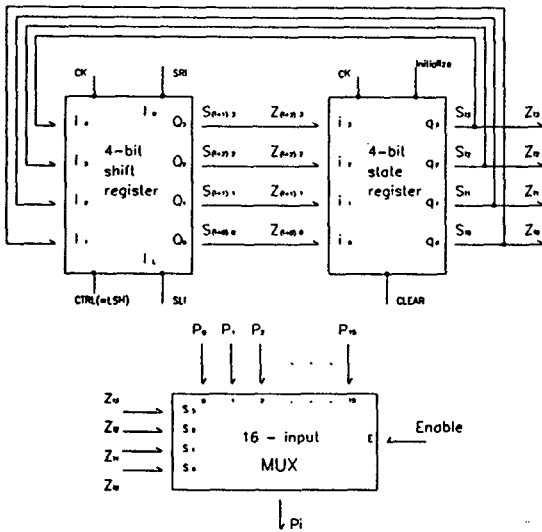


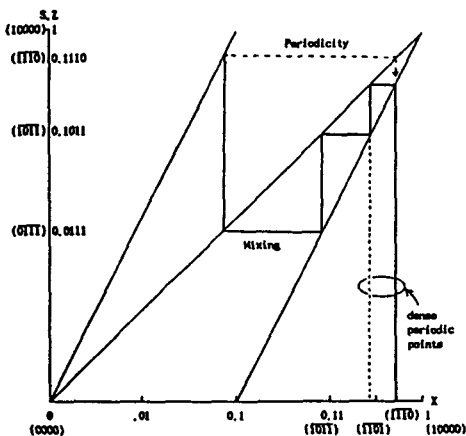
그림 4. 카오스적인 길이 4 비트 패턴 발생기
 Fig. 4. Chaotic pattern generator with length 4-bit pattern.

표 1. 초기상태 $x=1110$ 의 왼쪽 시프트 동작
 Table 1. Left shift operation for initial state $x=1110$.

Number of Shift Operations	Output Q	Decimal Numbers	Period
	INITIAL State LOAD 1=1110	1110 1110	14 14
1	LSH	1101	13
2	LSH	1011	11
3	LSH	0111	7
4	LSH	1110	14
5	LSH	1101	13
6	LSH	1011	11
7	LSH	0111	7
8	LSH	1110	14
9	LSH	1101	13
.	.	.	.
.	.	.	.
.	.	.	.

표 2. 초기상태 $y=1101$ 의 왼쪽 시프트 동작
 Table 2. Left shift operation for initial state $y=1101$.

Number of Shift Operations	Output Q	Decimal Numbers	Period
	INITIAL State LOAD 1=1110	1101 1101	13 13
1	LSH	1011	11
2	LSH	0111	7
3	LSH	1110	14
4	LSH	1101	13
5	LSH	1011	11
6	LSH	0111	7
7	LSH	1110	14
8	LSH	1101	13
9	LSH	1011	11
.	.	.	.
.	.	.	.
.	.	.	.



(a) Cycle of periodic 4 for initial state $x=1110$