

혼돈맵들을 사용하는 합성 상태머신

*박광현, **서용원, ***Khalifa Eltayeb kh Mustafa

*충주대학교 전자통신과, **청주대학교 전자공학과, ***University Technology Malaysia

The composition state machine using the chaotic maps

Kwang-Hyeon Park*, Yong-Won Seo**, Khalifa Eltayeb kh Mustafa***

Chung-Ju University*, Cheong-Ju University*, ***University Technology Malaysia***

Abstract - 기율기 S=2인 톱니맵 $S_2(x)$ 와 텐트맵 $T_2(x)$, 두 혼돈맵들을 차례로 연결시킨 합성맵을 사용하여 합성상태머신을 설계하였다. 혼돈맵의 합성논리를 이용하여 설계한 합성상태머신에서 발생하는 난수적인 상태들을 그래프적으로 보였으며, 난수적인 상태들의 주기는 정밀도와 이산화된 진리표에 따른 길이를 발생시켰다.

1. 서 론

혼돈역학의 난수성(randomness)를 전기적인 상태머신(State machine)에 활용하고자 이산화된 혼돈맵들(discretized chaotic maps)을 사용하여 혼돈적인 합성머신(Chaotic Composition Machine)을 설계하였다. 디지털 회로나 시스템들은 그동안 출력에 관여하는 변수들의 종류에 따라 크게 Mealy Machine 과 Moore Machine 으로 대별되어져 왔으며, 이 두 상태머신들은 조합논리회로(Combinational logic circuit)의 설계에 따라 다양한 다음상태들과 출력을 발생시키고 있다.

따라서, 이 논문에서는 상태머신의 조합회로 부분으로 혼돈함수(Chaotic Function)의 변환을 수행하는 두 가지 혼돈맵들(Chaotic Maps)을 사용함으로써 [1] 보다 난수적인 상태들을 발생시키는 합성상태머신(Compositive state Machine)을 설계하여 제시하였다.

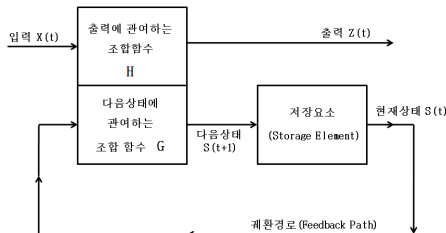
사용한 두 가지 혼돈맵들로는 이산화된 톱니맵(Discretized Saw-tooth map)과 텐트맵(Tent map)이며[2], [3], 합성상태머신은 8비트의 2진수(Binary number) 값으로 그 랜덤성을 보이도록 하였다.

2. 본 론

2.1 상태머신과 혼돈맵들

2.1.1 상태머신

다음 [그림 1]의 디지털 회로와 시스템의 블록도(Block Diagram)에 보인 디지털순차회로[Digital Sequential Circuit]에 관련시킬 수 있는 Mealy 와 Moore, 두 상태머신들의 상태전이함수(State-Transition Function)나 출력함수(Output Function)은 각각 다음 식들 (1), (2)로 정의 할 수 있다[4].



<그림 1> 전형적인 디지털 회로의 블록도

저자 소개

- *박광현 : 충주대학교 전자통신과 정교수
- **서용원 : 청주대학교 전자공학과 박사과정
- ***Khalifa Eltayeb kh Mustafa : 말레이시아 공과대학 박사과정

본 연구는 충북테크노파크 (맞춤형과제 "인휠 모터 기반의 미니 전기차 제어시스템 개발", 주관기업 (주) 이씨엠) 지원에 의한 것입니다.

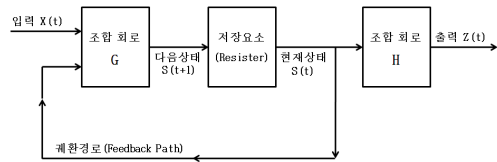
[Mealy 상태 머신의 상태전이함수와 출력함수]

$$\begin{aligned} \text{상태전이함수} : S(t+1) &= G(S(t), X(t)) \\ \text{출력함수} : Z(t) &= H(S(t), X(t)) \end{aligned} \quad (1)$$

[Moore 상태 머신의 상태전이함수와 출력함수]

$$\begin{aligned} \text{상태전이함수} : S(t+1) &= G(S(t), X(t)) \\ \text{출력함수} : Z(t) &= H(S(t)) \end{aligned} \quad (2)$$

와 같이 각각 기술 할 수 있고, 설계한 합성머신의 경우는 Mealy 상태머신의 형태이므로 앞의 식 (2)에 의해 [그림 2]의 블록도를 갖도록 설계 하였다[5].



<그림 2> 설계한 합성(상태)머신의 블록도

2.1.2 이산화된 톱니파

기율기 S=2를 갖는 다음 식 (3)으로 정의되는 톱니맵을 8비트로 이산화시킨다면

$$S(x) = \begin{cases} 2x, & 0.0 < x \leq 0.5 \\ 2x - 1, & 0.5 < x \leq 1.0 \end{cases} \quad (3)$$

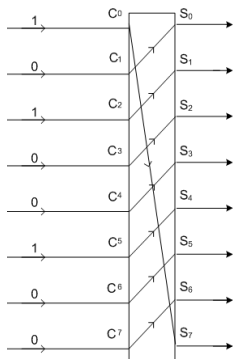
십진수값 x=0.0을 제외한 다음 [표 1]과 같이 2진수(Binary Number)의 8비트 유리수들(8-bit Rational Number)로 이산화된 진리표(Truth Table)를 얻을 수 있고, 이 진리표로부터

<표 1> 8비트로 이산화된 톱니맵의 부분적 진리표

	입력변수	출력변수
	S ₇ S ₆ S ₅ S ₄ S ₃ S ₂ S ₁ S ₀	C ₇ C ₆ C ₅ C ₄ C ₃ C ₂ C ₁ C ₀
1	00000001	00000010
2	00000010	00000100
3	00000011	00000110
4	00000100	00001000
5	00000101	00001010
6	00000110	00001100
7	00000111	00001110
8	00001000	00010000
9	00001001	00010010
10	00001010	00010100
⋮	⋮	⋮
126	01111110	11111100
127	01111111	11111110
128	10000000	00000001
129	10000001	00000011
130	10000010	00000101
⋮	⋮	⋮
252	11111100	11111001
253	11111101	11111011
254	11111110	11111101
255	11111111	11111111

식 (4)와 같은 간략화 된 부울대수(Simplified Boolean algebra)를 찾아냄으로써 [그림 3] 과 같이 입력과 출력의 관계를 배선의 변경만으로 설계 하였었다[2].

$$C_0 = S_{15}, C_1 = S_0, C_2 = S_1, C_3 = S_2, C_4 = S_3, \\ C_5 = S_4, C_6 = S_5, C_7 = S_6 \quad (4)$$



<그림 3> 이산화된 튜니맵의 조합회로

이렇게 8비트로 이산화 한 경우, 튜니함수와 이산화 된 튜니맵과의 오차는 구간 (0, 1]을 갖는 튜니함수의 기능을 맵의 해당구간 (0, 1]내에서 반복하기위한 이산화 된 튜니맵과의 오차는 예 불과하다는 것을 진리표에 통해서도 확인할 수 있었고, 식(5)에 의해 표현 할 수 있는 이산화된 상태의 순서들도

$$S_n = (0.x_1x_2x_3 \dots x_{16})_2 \\ = \sum_{i=1}^{16} x_i 2^{-i} \quad (5)$$

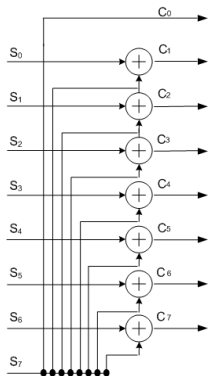
역시 난수성(Randomness)을 갖는 것이 입증되었다.

튜니 함수의 경우와 마찬가지로 다음 식 (6)으로 정의되는 텐트함수의 이산화된 텐트맵도

$$T(x) = \begin{cases} 2x, & 0.0 < x \leq 0.5 \\ 2(1-x), & 0.5 < x \leq 1.0 \end{cases} \quad (6)$$

[그림 4]처럼 배타적합논리게이트(Exclusive-OR logic gate)만을 사용하고, 간략화된 부울식에 일치하는 선 연결만으로 설계하였었다[3].

$$C_0 = S_7, C_1 = S_0 \oplus S_7, C_2 = S_1 \oplus S_7, \\ C_3 = S_2 \oplus S_7, C_4 = S_3 \oplus S_7, C_5 = S_4 \oplus S_7, \\ C_6 = S_5 \oplus S_7, C_7 = S_6 \oplus S_7 \quad (7)$$



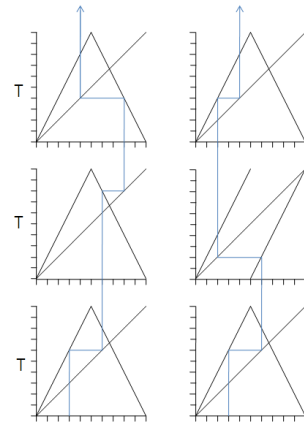
<그림 4> 이산화된 텐트맵의 조합회로

2.2 튜니맵과 텐트맵의 합성논리

텐트맵 변환을 n회 반복하는 것은 튜니맵 변환을 n-1회 반복한 후에 마지막 일회만을 텐트맵 변환으로 수행한 결과와 같다는 것이 다음 수식 (8)에 의해 정의되어졌고 [1],

$$T^n(x) = T(S^{n-1}(x)) \quad (6)$$

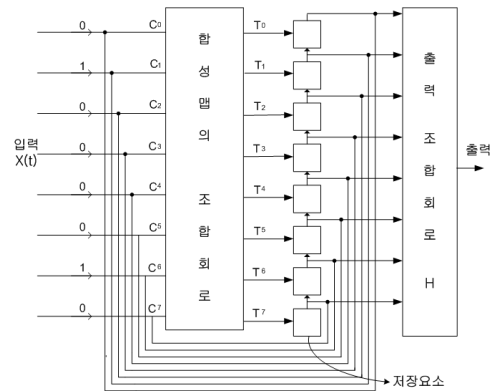
더 나아가, 텐트맵과 튜니맵이 뒤섞이는 경우 일지라도, 마지막 일회 변환이 텐트맵이라면 역시 같은 결과가나온다는 사실을 다음 [그림 5]를 통해서도 알 수 있다.



<그림 5> T(T(T(x))) = T(S(T(x))) 인 경우

2.3 합성상태머신

앞의 [그림 2]에 보인 블록도와 일치하도록 설계한 합성상태머신을 다음 [그림 6]에 제시하였다.



<그림 6> 합성상태머신

그리고 발생하는 상태의 주기 P는, 이산화과정에서 8비트의 왼쪽 이동 레지스터(Left Shift Register:LSR)를 사용하였으므로, 하나의 입력마다 8개의 상태를 발생시키는 P=8의 길이를 갖는다. ([그림 7]에 한 입력마다 주기 8인 발생하는 상태의 혼돈적인 주기성을 보인다.)

3. 결 론

디지털회로나 시스템에 관한 분류의 한 방법으로 두가지 상태머신 - Mealy머신과 Moore머신 - 을 정의하고, 사용용도에 따라 두 종류중 하나나, 혼합된 형태를 설계하여 사용해왔다. 그리고 기존의 상태머신들이 정해진 상태들을 발생에 목표를 두었다면, 이 논문에서 설계하여 제시한 합성상태머신은 혼돈맵들의 변환에 의한 비선형적 움직임(Non Linear Dynamics)을 이용하여 난수적인 상태, 즉 혼돈적인 상태(Chaotic state)를 발생시키고자 하였다. [그림 8]에 의해 합성상태 머신에서 발생하는 혼돈적인 상태들을 기하학적으로 입증하였고, 사용된 이산화진리표들에 따라 혼돈적인 상태들의 주기도 확인할 수 있다.

[참 고 문 헌]

- [1] Heinz-Otto Peitgen, Hartmut Jurgens, Dietmar Saupe, Evan Maletsky, Terry Percianate and Lee Yunker, "Fractals for the classroom", Springer-verlag(New York), 1991.
- [2] 박광현, 백승재, "혼돈맵을 사용한 난수성 2진 순서발생기의 설계", 한국콘텐츠학회논문지 제8권 제7호 별책, 논문08-08-07-7, 2008년 07월.
- [3] 백승재, 박진수, "이산화된 텐트맵의 설계", 한국콘텐츠학회논문지 '04 Vol. 4 No.
- [4] Milos D.Ercegovac and Tomas Lang, " Digital Systems and Hardware / Firmware Algorithms", John Wiley & Sons, 1985.
- [5] Kwang-Hyeon Park, Jong-Sun Hwang and Chong-Eun Chung, "Implementation of Chaotic State Machine using Deterministic Chaos Function", Journal of Electrical Engineering and Information Science, Vol. 3, No. 2, 1998.