

論文94-31B-12-5

# 자기구성 퍼지제어의 궤적연구 및 도립진자 제어 적용 (Trajectory Study of Self-organizing Fuzzy Control and Its Application to Inverted Pendulum Control)

朴 楨 日\*, 柳 在 奎\*\*

(Jung Il Park and Jae Kyu Ryu)

## 要 約

본 논문에서는 룩업 테이블을 사용하는 자기구성 퍼지제어에서 룩업테이블의 새로운 수정법을 제안한다. 이 방법은 빠른 응답성을 갖도록 룩업테이블을 수정하는 특징을 갖는다. 원리는 제어기가 상태평면도에서 기준입력이나 외란이 변하는 초기시점에 궤적을 빠른 응답성을 갖는 범주, 즉 오차보다 오차미분치가 큰 쪽으로 유도하는 것이다. 또 스위칭면과의 절대 거리에 비례하는 가변 학습속도 계수를 도입한다. 그리고 제안한 방법의 응용성을 보기 위하여 몇가지 예에 대해서 시뮬레이션을 수행하였으며 도립진자를 이용하여 진자의 도립실험을 수행하였다.

## Abstract

In this paper, we propose a new modification method of the look-up table in self-organizing fuzzy control using look-up table. This method has the property that look-up table is modified to have fast response property. Its principle is that the controller forces the trajectory to go into the fast response region which the error change amount is larger than the error at initial time whenever the reference or disturbance change. Also we introduce the variable learning speed coefficient which is proportional to distance from switching curve. And to demonstrate the applicability of the proposed method, we had simulation study for some examples and executed pole balance experiments with inverted pendulum.

## 1. 서 론

\*正會員, 嶺南大學校 電子工學科  
(Dept. of Elec. Yeongnam Univ.)  
\*\*準會員, (株)金星電線 電力研究所  
(Goldstar Cable Co., Ltd. Elec. Power  
Research Institute)  
接受日字: 1994年 6月 16日

퍼지이론이 1965년 버클리 대학의 L.A. Zadeh교수에 의해서 발표된 이래로 제어분야에 도입되어 퍼지제어에 관한 많은 이론적 발전이 있었으며<sup>1,2</sup>, 산업현장에 성공적으로 적용된 연구사례가 발표되고 있다. 퍼지 제어기는 전문가의 경험이나 지식으로부터

제어규칙을 구성할 때, 그 과정은 상당히 주관적이게 되고 전문가의 경험이나 지식이 얼마나 정확히 제어기 설계자에게 전달되는가가 설계의 중요한 요인이 된다. 퍼지 제어기의 성능은 퍼지화 방법, 제어규칙, 비퍼지화 방법, 입출력 언어변수의 모양에 관계된다. 특히 퍼지제어기의 설계에 비교적 많은 영향을 미치는 것은 제어기의 전건부와 후건부의 언어변수의 소속함수모양과 제어규칙이다. 이들은 경험과 시행착오를 거쳐서 최적값으로 조절된다. 그 중에서 제어규칙을 구성하는 과정이 중심이 되는데 제어규칙 설계법에는 전문가의 지식·경험에 의한 방법, 조작자의 조작모델에 의한 방법, 플랜트의 퍼지모델에 의한 방법, 제어규칙의 자기구성에 의한 방법 등이 있다. 이와 같은 이유로 인해 최근에는 소속함수의 자동조정(Auto-tuning)과 제어규칙의 자기구성(Self-organizing)에 관한 연구가 많이 이루어 지고 있다.<sup>[2,5]</sup>

제어규칙의 조절은 각 규칙을 조절할 수도 있고 Off-line으로 추론을 행해서 제어기의 입출력 매핑(mapping)을 나타내는 룩업테이블(Look-up table)을 만든후 평가함수에 의해서 제어오차를 줄이는 방향으로 룩업 테이블의 값을 변화시켜 나가는 방법등이 있다. 룩업 테이블은 실시간 제어에서 계산량을 줄이기 위한 목적으로 많이 쓰이고 있다.

자기구성 퍼지제어기(Self-organizing Fuzzy Controller)는 T.J. Procyk 과 E.H. Mamdani에 의해서 처음으로 제안되었는데<sup>[2]</sup>, 추론을 위한 룩업 테이블과 이 추론 룩업 테이블을 수정하고 재구성해 나가기 위한 성능측정 결정(Performance Measurement Decision) 룩업 테이블이 있다. 그 후, B.S. Zhang 과 J.M. Edmunds는 시스템상태의 제적방향을 이용하여 제어규칙의 생성과 수정을 위한 Supervision rule에 관한 논문을 발표하였다.<sup>[3]</sup>

본 논문에서는 [3]의 논문을 근간으로 하여  $e$ 와  $\dot{e}$ 의 상태 평면상에서 시스템 상태의 제적방향 분석을 통한 룩업 테이블의 새로운 학습방법을 제안한다. 또 각종 시뮬레이션을 통하여 본 방법의 학습특성을 살펴본다. 또한 비선형이고 불안정한 도립진자 시스템에 적용하여 실험적으로 본 논문에서 제안한 방법의 우수성도 보인다.

## II. 룩업 테이블을 사용하는 자기구성 퍼지제어

퍼지제어에서 추론과정은 많은 계산량을 요구한다. 입력변수의 갯수가 많아지고 각 입력변수의 언어변수의 갯수가 많아지면 제어규칙의 수는 기하급수적으로 늘어나게 된다. 따라서, 빠른 제어를 필요로하는 시

스템에서는 실시간 제어에 어려움이 따른다.

그래서 off-line으로 추론을 행하여  $e$ 와  $\dot{e}$ 에 대해서 퍼지제어기의 입출력 매핑을 나타낸 룩업 테이블을 이용하여 제어를 수행한다. 룩업 테이블을 쓰면 계산시간을 많이 줄일 수 있으나 퍼지제어기의 입출력관계를 실제추론에 가깝게 구현하기 위해서는 많은 양자화 레벨 즉, 많은 기억공간을 필요로 하게 된다.

[2]의 성능측정결정표(Performance Measurement Decision Table)를 이용한 학습방법은 시스템 상태의 절대적 위치에 의존한 학습방법이어서 목표치가 변하거나 외란이 발생할때 민감한 반응을 보인다. 그래서 Daley와 Gill<sup>[5]</sup>은 룩업 테이블상의 수정하지 않는 영역을 넓혀서 목표치가 변하거나 외란이 발생할때 수정의 빈도를 줄여 보고자 하는 연구를 하였다.

본 논문에서는 [3]의 학습방법을 근간으로 하여 새로운 학습방법을 제안한다. [3]의 논문에서는 다음과 같은 룩업 테이블을 수정하는 기본원칙이 있다. 기준 입력이나 외란에 의해 바람직한 제적을 벗어나게 되면 제어기는 바람직한 제적 범위내로 유도해야 한다. 이 때 제어기가 바람직한 제적내로 유도를 못할 때는 제어법칙이 수정되어야 한다. 또 제어기가 충분히 빠르게 제적을 바람직한 범주내로 움직이고 있다면 제어법칙은 수정될 필요가 없다.  $e$ 와  $\dot{e}$ 는 다음식으로 정의한다.

$$e(k) = (r - y(k)) \cdot Ke \quad (1)$$

$$\dot{e}(k) = (e(k) - e(k-1)) \cdot Kc \quad (2)$$

여기에서  $Ke$ 와  $Kc$ 의 값을 적절히 조정하여  $e(k)$ 와  $\dot{e}(k)$ 량의 크기가 정규화된 구간내에 들도록 한다.  $r$ 은 기준입력이고  $y(k)$ 는 출력이다.

그림 1에서 ' $v \cdot \dot{e} < 0$ '인 영역(II, I)에서 시스템의 상태가  $a(e, \dot{e})$ 에서  $b(e + \Delta e, \dot{e} + \Delta \dot{e})$ 로 변할때  $\Delta \dot{e} / \Delta e \geq Q_1 > 0$ 이면 시스템의 상태가 충분히 빠른 속도로 스위칭면(Switching curve)으로 움직여 가기 때문에 제어규칙을 수정할 필요가 없다. 마찬가지로 ' $v \cdot \dot{e} > 0$ '인 영역(III, IV)에서도  $\Delta \dot{e} / \Delta e \leq Q_2 < 0$ 이면 시스템의 상태가 스위칭면으로 충분히 빠른 속도로 움직여 가고 있기 때문에 제어규칙을 수정할 필요가 없다. 이 방법은 시스템 상태의 절대적인 위치가 아닌 움직임의 방향에 따른 제어규칙 수정법이다. 즉, 스위칭면에서 현 상태의 위치가 멀리 있더라도 현재의 제적이 스위칭면으로 빠르게 접근하고 있다면 제어 규칙은 수정할 필요가 없는 것이다. 이것을 요약하면 아래와 같이

두개의 Super-vision 규칙으로 나타낼 수 있다.<sup>[3]</sup>

$$\text{IF } v \cdot \dot{e} < 0 (\text{I, II region}) \quad \text{THEN } m = Q_1 - \Delta \dot{e} / \Delta e \quad (3)$$

$$\text{IF } v \cdot \dot{e} > 0 (\text{III, IV region}) \quad \text{THEN } m = -Q_2 + \Delta \dot{e} / \Delta e \quad (4)$$

$m > 0$ 인 경우는 제어규칙의 수정량을  $m$ 에 비례하는 값으로 하고  $m \leq 0$ 인 경우는 제어규칙을 수정하지 않는다. 룩업 테이블 수정량은 다음식과 같다.

$$\Delta r = \text{sat}(\Delta r_{\max}, l_m \cdot m \cdot \text{sgn}(v)) \quad (5)$$

여기에서,  $l_m$ 은 학습속도를 결정하는 계수이고  $\Delta r_{\max}$ 는 1회 수정의 최대값이다.  $\text{sat}(a, x)$ 는  $x > a$ 인 구간에서는  $a$ ,  $|x| \leq a$ 인 구간에서는  $x$ ,  $x < -a$ 인 구간에서는  $-a$ 를 나타내는 saturation 함수이고,  $\text{sgn}(\cdot)$ 는 signum 함수이다.

룩업 테이블 수정은 다음식으로 계산된다.

$$\begin{aligned} & r(i(e(k-1)) + \Delta i, j(\dot{e}(k-1)) + \Delta j) \\ &= \text{sat}(r_{\max}, r(i(e(k-1)) + \Delta i, j(\dot{e}(k-1)) + \Delta j) + w_{ij} \cdot \Delta r(k)) \end{aligned} \quad (6)$$

여기에서  $\Delta i$ 와  $\Delta j$ 는  $-1.0, 1.0$ 의 값을 가진다.

$W_{ij}$ 는 가중인자(weighting factor)로서 다음식과 같다.

$$w_{ij} = \frac{1}{(1 + \text{abs}(\Delta i)) \cdot (1 + \text{abs}(\Delta j))} \quad (7)$$

여기에서  $\text{abs}(\cdot)$ 는 절대치(absolute) 함수이다. (6)식과 (7)식을 살펴보면 룩업 테이블은 한번에 9개의 요소가 수정이 되는데 가운데 요소의 가중인자가 1, 가운데와 인접하는 4개의 요소의 가중인자는 1/2, 가운데 요소와 인접하지 않는 4개 요소의 가중인자는 1/4이다. 제어입력은 다음과 같이 계산된다.

$$u(k) = Ku \cdot r(i(e(k)), j(\dot{e}(k))) \quad (8)$$

1. 빠른 응답성을 위한 룩업 테이블 수정

본 절에서는 앞절에서 설명한 룩업 테이블을 수정하는 기본원칙은 그대로 이용하고, 상태 평면도에서 제어기가 바람직한 궤적 범주내로 가능한한 빠르게 시스템의 상태를 유도할 수 있는 방법을 설명한다.

일반적으로 기준입력이 구형파로 입력되는 경우,

(1)식과 (2)식으로 정의된  $e$ 와  $\dot{e}$ 는 초기순간에는  $e$ 가 기준입력의 양,  $\dot{e}$ 는 0이므로 궤적의 첫 출발지점은 그림1의 상태평면도의  $e$ 축상의 오른쪽 끝부분이

된다. 또 기준입력이 0으로 변하는 순간에는 궤적은  $e$ 축상의 왼쪽 끝부분에서 출발하게 된다. 이 때 제어기가 출력을 기준입력에 빠르게 추종하도록 하기 위해서는 기준입력이 변하는 초기순간이 중요하게 된다. 이러한 관점에서 보면 상태도의 I, II의 영역에서의 궤적의 형태가 시스템의 속응성을 크게 좌우한다. 빠른 응답성을 갖기 위해서는 초기에  $\dot{e}$ 의 절대크기를 가능한한 크게 하는 것이 중요하다.

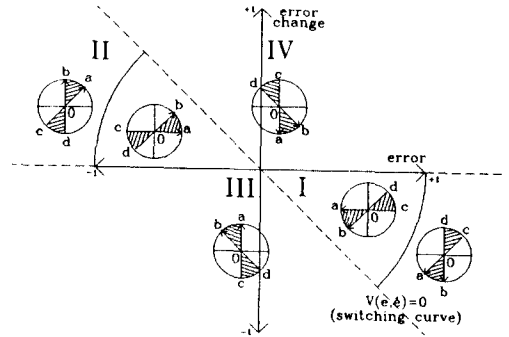


그림 1. 상태평면상에서의 궤적 분석

Fig. 1. Trajectory analysis on the state plane.

그래서 I, II의 영역에 관계하는 Supervision 규칙인 (3)식을 (9)식과 같이 수정한다. (4)식은 시스템의 속응성에 크게 관계하지 않으므로 그대로 이용한다.

$$\text{IF } v \cdot \dot{e} < 0 (\text{I, II region}) \quad \text{THEN} \quad (9)$$

$$\begin{aligned} & \{ \text{IF } (e^2(k) + \dot{e}^2(k) < L^2) \text{ THEN } m = Q_1 - \Delta e / \Delta \dot{e} \\ & \quad \text{ELSE } m = Q_1 - \Delta \dot{e} / \Delta e \} \end{aligned}$$

여기에서  $L$ 은  $e$ 와  $\dot{e}$ 를 각각 같은 양으로 정규화했을 때의 최대값을 나타낸다. (9)식의 Supervision 규칙을 상태도 위에 나타낸 것이 그림 1이다. 그림에서의  $L$ 의 값은 1이다. 궤적이  $o$ 에서 출발하여  $a$ 와  $b$ 의 빗금친 영역사이로 움직이게 되면 이 위치에 매핑된 룩업 테이블은 수정되지 않는다. 이렇게 룩업 테이블이 수정되지 않는 방향범위는  $Q_1$ 과  $Q_2$ 의 값에 따라서 결정이 되며  $Q_1$ 과  $Q_2$ 의 절대값이 커질수록 룩업 테이블이 수정되는 영역이 넓어지게 된다(빗금친 영역이 좁아지게 됨).

이번에는 룩업 테이블을 수정할 때 얼마의 양으로 수정을 해야 하는가를 설명하기로 한다. (9)식을 보면 I, II영역에서 룩업 테이블을 수정하는 양은 원의 내부에서는 궤적이  $o$ 에서  $b$ 로 움직여 갈 때 보다  $o$ 에서  $c$ 의 방향으로 움직여 갈 때가 룩업 테이블의 수정량이 더 커지게 된다. 룩업 테이블의 값은 제어입력

으로 되기 때문에 제어입력은 더 커지게 되며, 다음 번에 기준입력이 변해서 이 범주대로 제적이 지나갈 때  $\dot{e}$  가 더 큰쪽으로 유도할 수가 있다. 제적이 a와 b의 빗금친 사이영역으로 움직일 때는 이에 해당하는 룩업 테이블은 수정하지 않는다. 이렇게 기준입력이 바뀔 때마다 학습이 되면 제적의 방향이 초기시점에서는 원의 바깥 쪽으로 가려는 경향이 있으며 빨리 스위칭면으로 도달할 수가 있게 된다. 그러나 이와 같이 기준입력이 변할 때 마다 학습이 끊임없이 이루어 지게 되면 제적이 계속 바깥쪽으로 가게 되어 제어입력도 과도하게 된다. 그래서 이와 같이 학습이 과도하게 이루어 지는 것을 막아 주기 위하여 상태도에서 원의 외부에서는 바깥쪽(빗금친 영역내의 방향)으로 제적이 움직일 때는 룩업 테이블의 수정이 이루어 지지 않도록 제어규칙을 바꾸어 준다. 그러면 이러한 Supervision 규칙하에서는 학습이 반복할 수록 제적을 가능한한 크게 하면서 일정한 크기로 원호를 그리면서 스위칭면에 도달하게 할수가 있어서 빠른 응답성을 갖도록 할 수 있다.

2. 빠른 학습을 위한  $l_m$  의 선정

(5)식을 살펴보면  $l_m$  의 값 선정에 따라서도 학습속도가 차이남을 알 수 있다. 그래서 본 논문에서는 다음과 같은  $l_m$  의 값을 사용한다.

$$l_m = abs(v) + \alpha \tag{10}$$

여기에서  $\alpha$  는 스위칭면 위/아래 근방에서도 기본적으로 일정량의 학습이 이루어 지도록 하는 상수이다. (10)식의 의미는 룩업 테이블의 수정을 앞절에서 설명한대로  $Q_1, Q_2$  의 값과 제적의 방향만을 보고 하는 것이 아니라 상태평면 위에서의 현재의 절대위치와 스위칭면과의 절대거리에 비례해서 룩업 테이블 수정을 병행하도록 하는 것이다. 즉, 스위칭면과 멀리있을 때는 수정량을 많이해서 제어입력을 크게하여 학습 속도를 높이고, 가까이 있을 때는 수정량을 적게해서 학습속도를 낮춘다. (10)식에서의  $v$ 는 다음식과 같다.

$$v = Q_s \cdot e + \dot{e} \tag{11}$$

여기에서  $Q_s (> 0)$ 는 상태평면상에서의 스위칭면의 기울기 이다.

III. 시뮬레이션 및 도입진자 실험

1. 시뮬레이션

이상에서 제안한 방법의 효과를 보기 위하여 다음

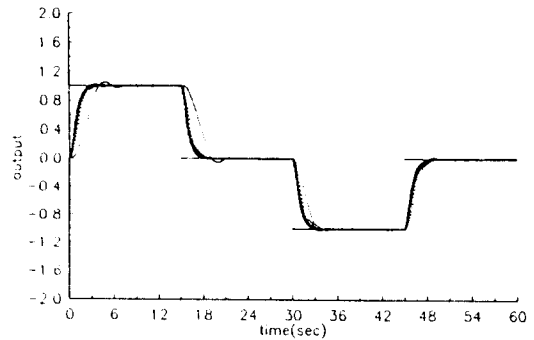
과 같은 시간지연을 갖는 적분기에 대해서 Zhang의 방법<sup>1)</sup>과 비교 시뮬레이션을 수행하였다.<sup>19)</sup>

$$\text{Model}; \frac{3}{s(3s+1)}$$

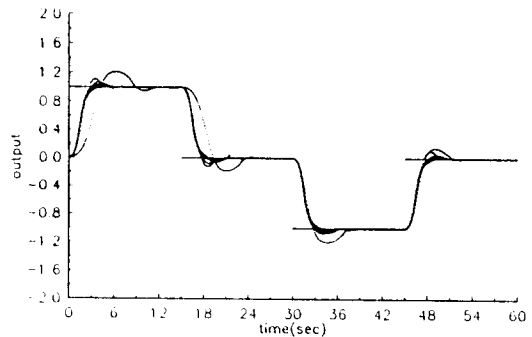
스케일링 인자는 다음과 같이  $K_p = 1, K_c = 1, K_i = 0.1$ 로 선정하였으며 샘플링 시간은 0.1초로 하여 Runge-kutta 방법으로 시뮬레이션을 하였다. 또 스위칭면의 기울기( $Q_s$ )는 1.5.  $Q_1, Q_2$  의 값은 각각 1.5로 하였다. 정규화된  $e$ 와  $\dot{e}$ 를 각각 -1과 1사이에서 16단계로 양자화를 하여 16x16의 룩업 테이블을 이용하였으며, 룩업 테이블은 초기에 비워 둔 상태에서 시작하였다. 학습계수  $l_m$ 의 기본 바이어스 값  $\alpha$ 는 0.3으로 하였다. 또  $\Delta r_{MAX}$  값은 1로  $r_{MAX}$  값은 100으로 하였다.

먼저 제어기의 학습능력을 알아 보기 위해 위의 모델에 대해서 룩업 테이블이 비어 있는 상태에서 구형과 입력에 대하여 20번 반복시켜 가면서 출력응답을 살펴 보았다.

그림 2를 살펴보면 (b)의 경우 보다 본 논문에서 제안한 (a)의 경우가 학습이 빨리 이루어져 기동시간이 짧아지는 것을 볼 수 있다.



(a)



(b)

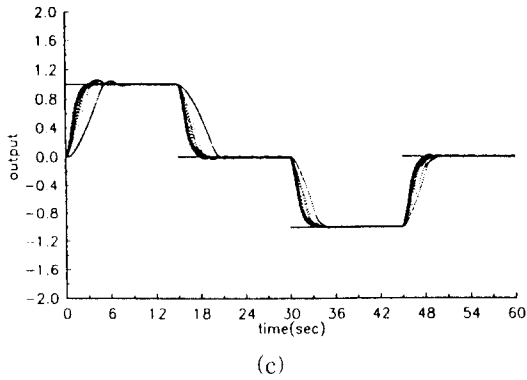


그림 2. 자기구성 퍼지제어 시스템의 구형파 응답

- (a) 제안 방식,  $l_m = \text{abs}(v) + 0.3$
- (b) Zhang의 방식,  $l_m = \text{abs}(v) + 0.3$
- (c) 제안방식,  $l_m = 0.3$

Fig. 2. Square wave responses of self-organizing fuzzy control system.  
 (a) proposed method,  $l_m = \text{abs}(v) + 0.3$   
 (b) Zhang's method,  $l_m = \text{abs}(v) + 0.3$   
 (c) proposed method,  $l_m = 0.3$

(c)의 경우는 학습속도 계수를 본 논문에서 제안한 (10)식을 사용하지 않고 일정한 값을 이용했을 경우의 응답이다. 이 경우는 (a)의 경우 보다 학습속도가 느린 것을 확인할 수 있었다.

그림 3은 그림 2에서 수행한 결과의  $e$ 와  $\dot{e}$ 의 궤적을 상태평면도 위에 도시한 것이다. (a)의 경우가 본 논문에서 의도한 대로 궤적이 초기에 바깥 쪽으로 원을 그리면서  $\dot{e}$ 가 빨리 커지면서 스위칭 면으로 가까이 감을 알 수 있다. (b)의 경우는 궤적이 초기에  $e$ 와  $\dot{e}$ 가 일정하게 변하여 안쪽으로 직선을 그리면서 스위칭 면으로 가까이 감을 알 수 있다. (c)의 경우는 (a)의 경우와 같은 제어규칙을 사용했지만 학습속도 계수를 일정하게 했기 때문에 학습 속도가 느려서 궤적이 빨리 바깥쪽으로 가지 못함을 알 수 있다.

그림 4는 그림 2에서 수행한 (a),(b)의 경우에 대하여 20번 학습을 하고난 최종의 룩업 테이블 즉, 제어평면을 3차원 공간상에 나타낸 것이다. 본 논문에서 제안한 (a)의 경우를 그림 3의 (a)와 같이 매칭시켜서 보면 궤적이  $\dot{e}$ 가 큰쪽으로 움직여 갈 때가 학습량이 크고,  $e$ 축과 평행하게 움직이는 곳에서는 학습량이 적었기 때문에 내부보다는 궤적이 출발하는 초기 부분에서 시작하여 원을 그리는 부분에서 학습량이 많았음을 알 수 있다. 학습량이 많아지면 다음 궤적이 그곳으로 지나갈 때 제어 입력이 커지게 되어 속응성이 있는 제

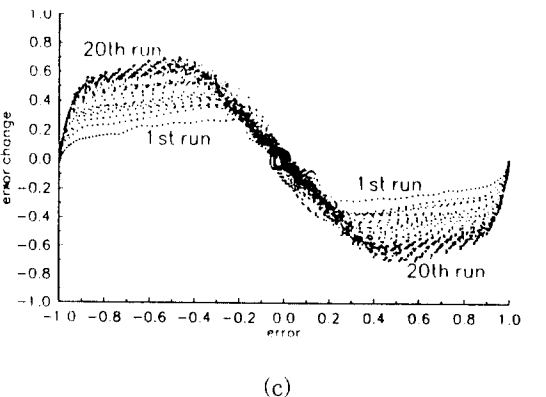
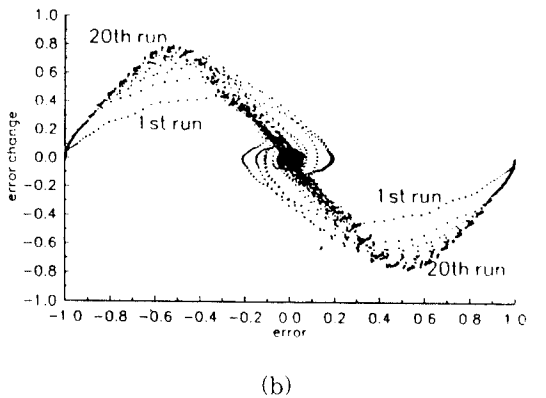
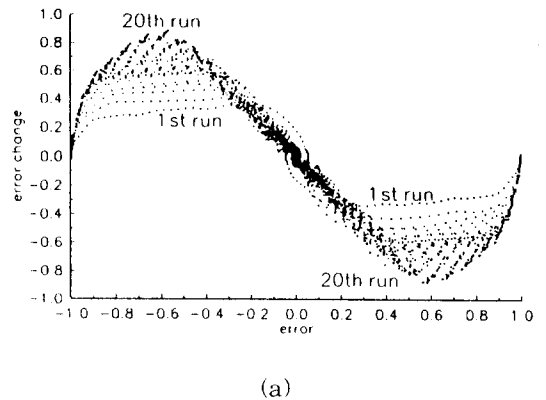
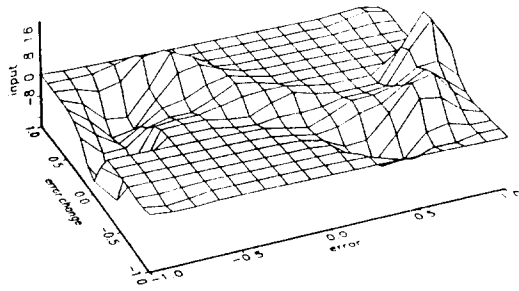


그림 3. 구형파 응답에 대한 위상 평면상의 궤적

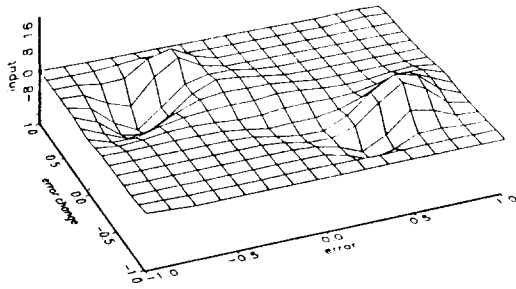
- (a) 제안 방식,  $l_m = \text{abs}(v) + 0.3$
- (b) Zhang의 방식,  $l_m = \text{abs}(v) + 0.3$
- (c) 제안 방식,  $l_m = 0.3$

Fig. 3. Trajectories on the phase plane for square wave response.  
 (a) proposed method,  $l_m = \text{abs}(v) + 0.3$   
 (b) Zhang's method,  $l_m = \text{abs}(v) + 0.3$   
 (c) proposed method,  $l_m = 0.3$

어를 할 수 있게 된다. (b)의 Zhang의 방법에서는 (a)에 비해서 학습량이 적었음을 알 수 있었다. 학습 방향은 스위칭면 위쪽으로는 + 방향, 아래쪽으로는 - 방향으로 학습이 되었음을 알 수 있다.



(a)



(b)

그림 4. 20번 학습후의 룩업 테이블(제어) 평면  
(a) 제안 방식 (b) Zhang의 방식

Fig. 4. Look-up table(control) surface after 20 times learning. (a) proposed method (b) Zhang's method

그림 5는 Zhang의 방법과 본 논문에서 제안한 방법을 구형과 입력에 대하여 20번 학습시킨후의 출력과 제어입력을 비교한 것이다. 기동시간이 차이가 남을 알 수 있으며 제어입력을 비교하면 이 원인을 찾을 수 있다. 기준입력이 변한 초기시점에 본 논문에서 제안한 방식이 제어입력이 크게 인가됨을 알 수 있다.

그림 6은 스텝 외란에 대해서 비교한 것이다. 스텝 외란은 수행후 7.5초 이후부터 출력에 직접 -0.1의 크기로 인가 되어서 37.5초 까지 지속된 후 해제되었다. 역시 본 논문에서 제시한 방식이 외란제거 능력이 뛰어난 것을 알 수 있다.

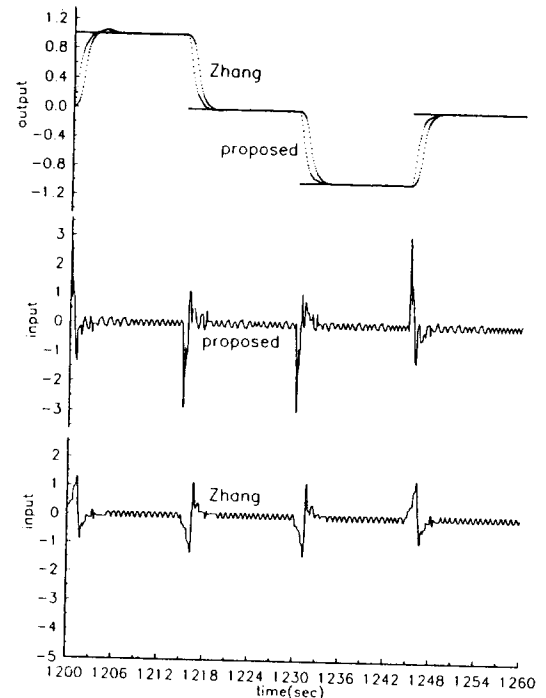


그림 5. 제안 방식과 Zhang의 방식의 입출력 비교  
Fig. 5. Comparison of the output and input between proposed and Zhang's method.

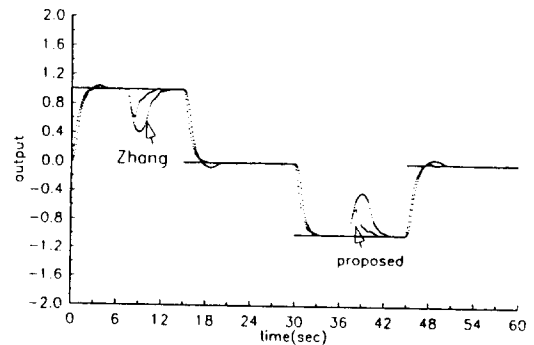


그림 6. 스텝외란에 대한 출력응답의 비교  
Fig. 6. Comparison of output responses for step disturbance.

## 2. 도립진자 실험 및 고찰

### 1) 실험장치의 구성

도립진자를 제어하는 전체적인 시스템의 블록도를 그림 7에 나타낸다. PC는 80287 coprocessor를 장착한 286AT를 사용한다. 드라이버는 SINANO Electric Co.의 아날로그속도형의 TEA081-12(120W)를 사용한다. 도립진자 시스템에서 모터는 DC Motor TDA12-7S (120W)를 사용하고, 모터와

진자 회전축에 각각 한바퀴당 1000개의 펄스를 발생시키는 오픈 콜렉터 방식의 인크리멘탈 엔코더를 부착한다. 또한, 모터에는 드라이버로의 속도귀환전압을 발생시키는 타코제너레이터가 부착되어 있다. 진자의 길이는 60cm이고 지름은 6mm 이다. 수레(cart)가 움직일 수 있는 레일의 길이는 2m이다.

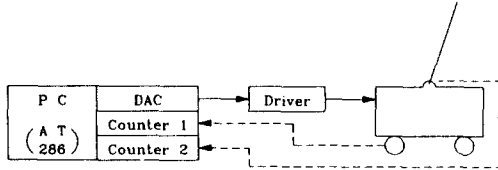


그림 7. 실험장치의 전체 블럭도

Fig. 7. Overall block diagram of the experimental apparatus.

수레와 진자(pendulum)의 엔코더에서 발생된 펄스는 인터페이스 보드에서 업/다운 카운트되어 PC로 입력되고, PC에서 C언어로 구현된 퍼지제어기에서 제어입력이 계산되고 이 값이 다시 인터페이스 보드를 통해서 드라이버에 인가된다. 이를 위한 인터페이스 보드의 블럭도를 그림 8에 나타낸다.

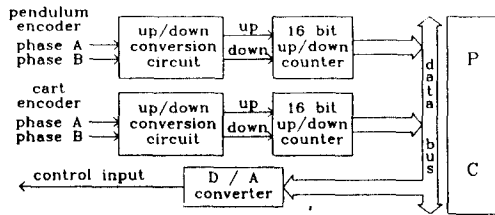


그림 8. PC 인터페이스 보드 블럭도

Fig. 8. PC interface board block diagram.

진자와 수레에 각각 부착된 엔코더로부터 발생된 펄스의 A상과 B상을 이용하여 방향변환회로를 통해 업/다운 클럭을 만든다. 이 클럭은 4 비트 업/다운 카운터 4개를 직렬연결하여 만든 16 비트 업/다운 카운터로 입력되며 카운트된 펄스의 갯수는 래치를 통해서 PC 데이터 버스에 연결되고 PC로 읽혀진다. 이를 이용해서 PC에서 계산된 제어입력은 래치와 12 비트 D/A 변환기를 통해서 드라이버에 인가된다. 각 래치와 8253에는 I/O 어드레스중 사용하지 않는 주소를 할당한다.

샘플링 주기는 도립진자시스템의 특성과 PC의 계산능력등을 고려하여 20msec로 한다. 인터럽트 주기를 만들기 위해서 8MHz의 PC시스템 클럭을 8분주

한 후 8253에서 다시 50Hz신호로 분주한다. 8253에서 만들어지는 50Hz신호를 8259의 인터럽트신호로 이용하여 하드웨어 인터럽트방식으로 20msec의 샘플링 주기를 만든다.

2) 도립진자 제어 시퀀스

진자는 아래로 늘어뜨려진 상태 즉, 안정된 상태에서 스윙업(swing up)과정을 거쳐서 세워진다. 목표치와 현재 진자위치와의 오차가 설정치( $\alpha$ )이내에 들어가면 스윙업 과정에서 퍼지제어과정으로 넘어간다. 스윙업 과정에서의 제어입력은 그림9에서와 같이 발생시킨다. 제어입력은 진자가 회전하는 평면을 4개로 나누어서 각 사분면에 양, 음의 부호를 교대로 하여 각 사분면에서 그 사분면의 45°선으로 갈수록 선형적으로 커져서 45°선에서 최대가 되었다가 45°를 넘어가게 되면 줄여 준다. 이렇게 스윙을 하다가 오차가 어느일정범위 이내에 들어오게 되면 앞절에서 설명한 대로 on-line으로 학습을 진행하며 균형제어를 시작한다. 그림 10은 제어알고리즘 순서도이다.

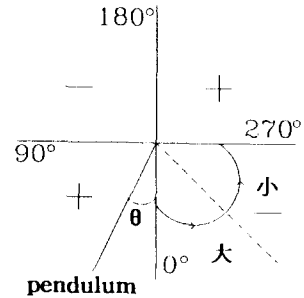


그림 9. 스윙업 과정동안의 제어입력 발생

Fig. 9. Control input generation during swing up process.

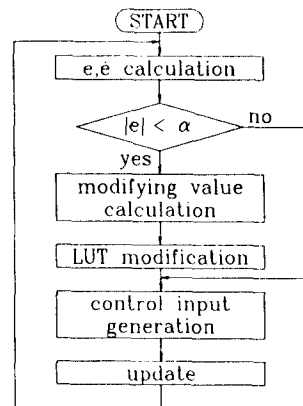


그림 10. 진자 균형제어의 시퀀스

Fig. 10. Sequence of pole balance control.

3) 실험결과 및 고찰

e와  $\dot{e}$  값은 -1에서 +1사이값으로 각각 정규화 이득 Ke와 Kc를 곱해서 정규화 시킨후 양의 영역과 음의 영역을 각각 10단 계로 나눈다. 대각선으로 대칭인 이 기본적인 룩업 테이블을 본 논문에서 제안한 학습방법에 의해서 수정한다.

Y.F. Li 와 C.C Lau<sup>[6]</sup>는 coarse 룩업 테이블과 fine 룩업 테이블을 사용하여 이 문제를 해결하고 있다. 비교적 오차가 큰 영역에서는 양자화크기를 크게 하여 coarse 룩업 테이블을 구성하고, 오차가 작은 영역에서는 정밀제어를 위하여 양자화 크기가 작은 fine 룩업 테이블을 구성 하였다. 본 실험에서도 양자화 크기를 달리한 두개의 룩업테이블을 이용하여 룩업 테이블의 불연속성과 정밀제어의 어려움을 극복한다. Coarse 룩업테이블과 fine 룩업 테이블의 이동은 fine 제어와 coarse제어의 Ke, Kc를 다르게 하므로써 룩업테이블의 스위칭이 일어날 때, 같은 e와  $\dot{e}$  에 대해서 같은 량의 제어입력이 발생하도록 하여 제어입력의 불연속성을 없앤다. Coarse 룩업 테이블만 학습 알고리즘을 이용해서 수정하고, fine 룩업 테이블은 목표치 근처에서 수정량이 불연속적이고 신뢰도가 떨어지므로 수정하지 않는다. 양자화된 e와  $\dot{e}$  를 각각  $i(e)$  와  $j(\dot{e})$ 로 나타낸다.

$i(e)$ 와  $j(\dot{e})$ 는 -10 에서 10사이의 정수값을 가진다. 즉, 20x20의 룩업 테이블이다. 제어입력 u는 다음식과 같이 생성된다.

$$u(k) = u(k-1) + r(i(e(k)), j(\dot{e}(k))) \quad (12)$$

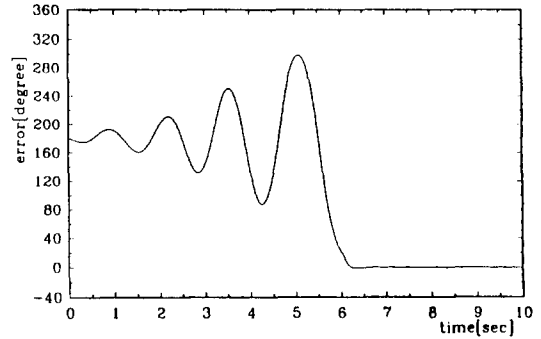
여기에서,  $r(i(e), j(\dot{e}))$ 는 룩업 테이블의 값이다. 도립진자의 균형제어 실험에서는 초기의 룩업 테이블을 비워둔 상태에서는 제어가 어려움으로 다음식을 이용하여 채워둔다. 즉, 상태평면도에서 스위칭면을 중심으로 절대거리에 비례해서 대칭적으로 초기 룩업 테이블을 만든다.

$$r(i(e), j(\dot{e})) = i(e) + j(\dot{e}) \quad (13)$$

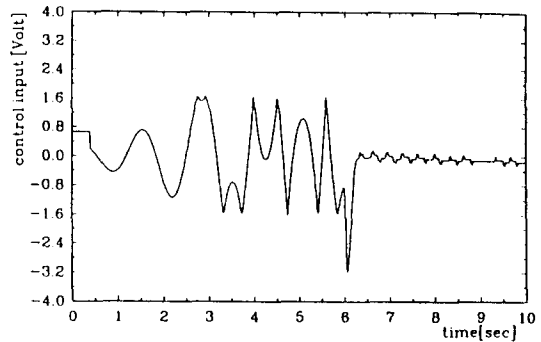
실험은 PID제어, 기본적인 룩업 테이블을 이용한 제어, Zhang에 의한 학습방법으로 룩업 테이블을 수정하는 경우, 본 논문에서 제안한 학습방법으로 룩업 테이블을 수정하는 경우로 구성된다.

그림 11에서 본 논문에서 제안한 학습방법으로 5회 반복한 룩업 테이블로 진자를 제어한 결과를 보여준다. 약 6초간의 스윙업 과정을 거쳐서 e가 21.6° (60개 엔코더 펄스)이하면 퍼지제어 과정으로 들어간다.

그림 12는 도립된 상태의 사진이다.



(a)



(b)

그림 11. 제안방식으로 5번 학습후의 진자 균형제어의 결과 (a)오차, (b) 제어입력

Fig. 11. The results of pole balance control after 5 times learning using proposed method. (a) error (b) control input

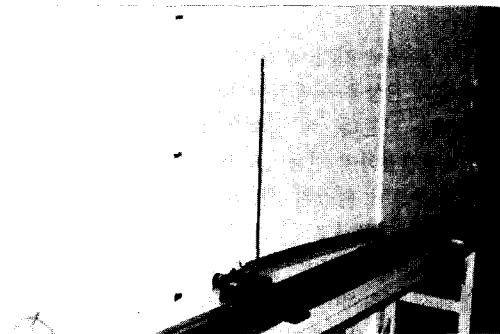


그림 12. 수레 진자 균형제어의 장면

Fig. 12. The scene of the cart pole balance control.



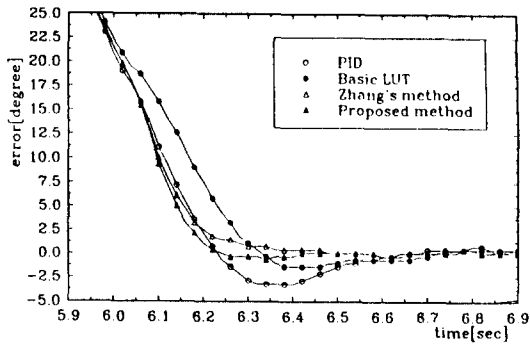
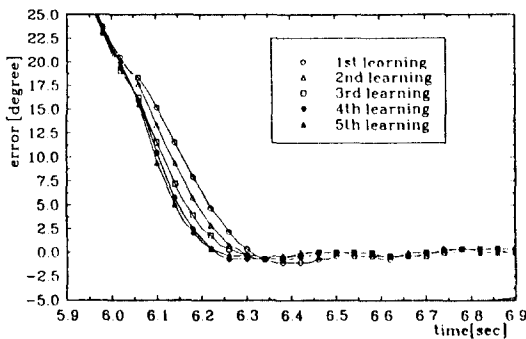
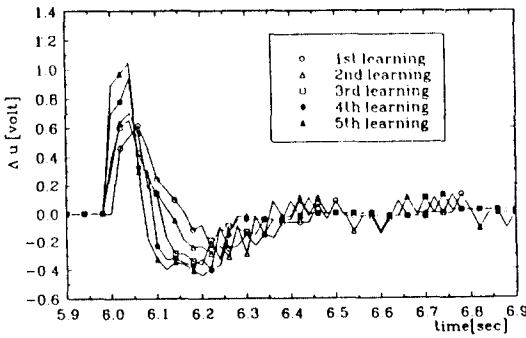


그림 13. 4방식의 결과 비교

Fig. 13. Results comparison of the 4 methods.



(a)



(b)

그림 14. 5번 반복 학습 동안의 제어 결과

(a) 오차, (b) 제어입력 변화분

Fig. 14. The control results during 5 times iterative learning. (a) error (b) changes in control input

그림 13은 PID제어, 기본적인 룩업 테이블을 이용한 제어, Zhang에 의한 학습방법으로 룩업 테이블을 5회 반복한 경우, 본 논문에서 제안한 학습방법으로

룩업 테이블을 5회 반복한 경우에 대한 결과이다. 상승시간과 오버슈트에 대해서 자세히 살펴보기 위해서 퍼지제어부분만 확대한 그림이다. 본 논문에서 제안한 방식의 세팅 시간이 100 ms 이상 짧아짐의 알 수 있다.

그림 14는 본 논문에서 제안한 방법의 반복학습과정의 결과와 반복에 따른  $\Delta u$ 의 값을 보여주고 있다. 학습이 반복됨에 따라  $\Delta u$ 의 값이 커짐과 세팅 시간이 짧아지는 것을 알 수 있다.

#### IV. 결론

본 논문에서는 Zhang<sup>[3]</sup>의 논문을 근간으로 하여  $e$ 와  $\dot{e}$ 의 상태평면상에서 시스템 상태의 궤적방향 분석을 통한 룩업 테이블의 새로운 학습방법을 제안하였다. 원리는 기준입력이나 외란이 변하는 초기에  $\dot{e}$ 의 성분을 크게하여 응답성을 빠르게 하였다. 이올러 스윙칭면과의 절대거리에 비례하는 가변 학습속도 계수를 도입하였다. 이 두가지를 도입한 본 방법이 Zhang의 방법보다도 기준입력이나 외란의 변화에 대하여 응답성이 빠름을 시뮬레이션을 통하여 살펴보았다.

또한 도입된 자 실험에서는 룩업 테이블을 오차가 큰 영역과 작은 영역으로 나눈 coarse 룩업 테이블과 fine 룩업 테이블을 이용하여 양자화 크기가 클때 정밀제어가 어려운 점을 극복할 수 있었으며, 또한 본 논문에서 제안한 스윙업 과정과 퍼지제어 과정이 자연스럽게 이어질 수 있었고, 이올러 세팅시간을 줄임과 동시에 오버슈트도 억제할 수 있음을 보였다.

#### 參考文獻

- [1] C.C.Lee, "Fuzzy logic in control systems: Fuzzy logic controller-Part I, II," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-435, March/April 1990.
- [2] T.J.Procyk & E.H.Mamdani, "A linguistic self-organizing process controller," *Automatica* vol. 15, pp. 15-30, 1979.
- [3] B.S.Zhang & J.M.Edmunds, "Self-organising fuzzy logic controller," *IEE Proceeding-D*, vol. 139, no. 5, september 1992.
- [4] S. Shao, "Fuzzy self-organizing controller and its application for dynamic

- process." *Fuzzy Set and Systems*, vol.26, pp.151-164, 1988.
- [5] Daley S., and Gill K.F., "A design study of a self-organizing fuzzy logic controller," *Proc. Instn Mech. Engrs*, 200, pp.59-69, 1986.
- [6] Li, Y.F. and Lau, C.C., "Development of fuzzy algorithm for servo systems." *IEEE control system magazine*, vol.9, no.3, pp.65-72, 1989.
- [7] 이영노, 김태원, 서일홍, "룩업테이블을 이용한 자동학습 퍼지제어기의 설계에 관한 연구," 대한 전자 공학회 논문집, vol.29, no.9, pp.76-87, 1992.
- [8] Chin E. Lin, and Yih-Ran Sheu, "A Hybrid-Control Approach for Pendulum-Car Control", *IEEE Trans. Industrial Electronics*, vol. 39, no. 3, pp. 208-214, 1992.
- [9] 류재규, 박정일, "빠른학습을 위한 자기구성 퍼지제어기의 궤적에 관한 연구," 대한전자공학회 하계 종합 학술대회 논문집, 제17권, 1호, pp.917-920, 1994.

---

 著 者 紹 介
 

---



朴 楨 日(正會員)

1958年 4月 8日生. 1981年 2月 경북 대학교 전자공학과 졸업(공학사). 1983年 2月 서울 대학교 대학원 전자공학과 졸업(공학석사). 1989年 2月 서울 대학교 대학원 전자공학과 졸업(공학박사). 1983年 4月 ~ 1984年 12月 삼성전자 종합 연구소 연구원. 1989年 4月 ~ 1992年 2月 삼성 종합 기술원 선임 연구원. 1992年 3月 ~ 현재 영남대학교 전자공학과 조교수. 주관심 분야는 적응 제어, 퍼지 제어, 신경망 제어, 공장 자동화 및 컴퓨터 제어 등임.



柳 在 奎(準會員)

1969年 9月 15日生. 1992年 2月 경북대학교 전자공학과 졸업(공학사). 1994年 8月 영남대학교 대학원 전자공학과 졸업(공학석사). 1994年 8月 ~ 현재 금성전선(주) 전력연구소 연구원. 주관심 분야는 퍼지제어, 신경망제어, 무인전력감시 시스템 등임.