

# 바람직한 제어 방향의 학습을 통한 퍼지 제어기의 자기 구성 방법

## A Method of Self-Organizing for Fuzzy Logic Controller Through Learning of the Proper Direction of Control

李 然 定\*, 崔 鳳 烈\*  
Yun-Jung Lee\*, Bong Yeol Choi\*

### 요 약

본 논문에서는 바람직한 제어 방향의 학습을 통한 퍼지 제어기의 새로운 자기 구성 방법을 제안한다. 기울기 강하법에 기반하여 특성을 모르는 동적 플랜트에 대한 퍼지 제어기를 자기 구성할 때 풀어야할 문제 중 하나는 오차를 줄이도록 하는 바람직한 제어입력의 변화방향을 알아내는 것이다. 이 문제를 해결하기 위한 한 방법으로서, 제어입력에 따른 오차의 변화 방향에 대한 대표 값을 분할된 상태영역에 할당하고, 반복적인 시행을 통해 강화 학습된 이 대표값을 이용하여 퍼지 제어 규칙을 학습하는 방법을 제안하였다. 제안된 자기구성 퍼지제어기는 간단한 구조를 가질 뿐 아니라 설계하기도 쉬운 장점을 갖는다. 제안된 방법의 타당성은 역진자 시스템에 대한 모의실험을 통하여 검증하였다.

### ABSTRACT

In this paper, a method of self-organizing for fuzzy logic controller(FLC) through learning of the proper direction of control is proposed. In case of designing a self-organizing FLC for unknown dynamic plants based on the gradient descent method, it is difficult to identify the desirable direction of the change of control input in which the error would be decreased. To resolve this problem, we propose a method as follows: at first, assign representative values for the direction of change of error with respect to control input to each partitioned region of the states, and then, learn the fuzzy control rules using the reinforced representative values through iterative trials. The proposed self-organizing FLC has simple structure and it is easy to design. The validity of the proposed method is proved by the computer simulation for an inverted pendulum system.

### I. 서 론

퍼지이론은 애매성을 포함한 인간의 지식이나 언어적 표현을 수학적으로 다루는 데 있어 효과적인 하나의 도구로서, 제어분야에 적용되어 가전제품부터 산업공정 제어에 이르기까지 광범위한 실제 응용과

---

\*경북대학교 공과대학 전자전기공학부

아울러 퍼지 제어 이론에 관한 많은 연구가 이루어져 왔으며 현재에도 활발히 연구되고 있는 분야이다.

초기에 퍼지 제어기의 설계 방법에 있어서는 숙련 운전자의 언어적 규칙이나 전문가 지식을 토대로 퍼지규칙을 설계하므로써, 수학적 모델이 없이 복잡하고 비선형적인 시스템 제어에 인간을 대치하는 제어기로 실제 시스템에 많은 적용이 이루어졌다<sup>[1][2]</sup>. 그러나, 이러한 설계방식에서는 원하는 제어성능을 갖는 퍼지 제어 규칙을 생성하는데 있어 반복적인 시행착오 과정을 필요로 함에 따라, 제어기 설계에 많은 시간을 요하게 되고, 설계된 제어기를 더욱 최적화시킬 수 있는지가 불분명하게 될 뿐만 아니라 외부환경의 변화나 시스템 노화에 따른 제어기 파라미터의 재조정 작업이 어려운 문제점을 갖고 있다.

이와 같은 퍼지 제어기 설계의 문제점을 해결하기 위한 한 방안으로 학습을 통해 스스로 제어규칙을 생성하는 이른바 자기구성 퍼지 제어기(self-organizing fuzzy logic controller(SOFLC))에 관한 연구가 진행되어 왔다. 자기구성 퍼지 제어기들의 분류는 직·간접 적용 퍼지 제어기로 구분하거나<sup>[3][4]</sup>, 다른 한편으로는 규칙 생성 방법에 따라 온라인(On-line), 오프라인(Off-line) 자기구성 퍼지 제어기로 크게 나누어 볼 수 있다.

그런데, 기존의 퍼지 제어기 자기 구성 방법들을 살펴보면, 많은 결과들이 기울기 강하법(gradient descent method)을 근간으로 하고 있다. 예를들면, 오프라인 방식에 있어서는 신경회로망과 퍼지를 혼합한 구조에서 역전파 알고리즘을 이용하는 방법<sup>[5]</sup>, 퍼지 제어기를 하나의 함수근사기(function approximator)로 간주하고 제어기 입출력 데이터를 이용하여 소속 함수와 규칙을 설정하는 방법<sup>[6]</sup> 등을 꼽을 수 있고, 온라인 방식에 있어서는 자기구성 퍼지 제어기에 관한 연구의 시초라 볼 수 있는 Procyk에 의해 제안된 방법<sup>[7]</sup>과 강화배환(reinforcement feedback)을 통한 학습 방법<sup>[8][10]</sup> 등이 이 부류에 속한다고 할 수 있다.

그러면, 기울기 강하법을 사용한 학습 시 해결해야 할 문제점에 대해 살펴보자.

먼저, 제어 성능 평가를 위한 성능 지수는 k 번째 시간에 대해서

$$J_k = \frac{1}{2} (y_k^d - y_k)^2 = \frac{1}{2} e_k^2 \quad (1)$$

라 하자. 여기서,  $y_k^d$ ,  $y_k$ ,  $e_k$ 는 각각 k번째 시간에서의 기준입력, 플랜트 출력, 그리고 기준입력과 플랜트 출력과의 오차이다.

식 (2)의 성능 지수를 최소화시키기 위한 학습 제어기의 파라미터  $w_i$ 의 변화량은 기울기 강하법을 이용하여 다음과 같이 계산된다.

$$\Delta w_i = w_i(k+1) - w_i(k) = -\eta \cdot \frac{\partial J_{k+1}}{\partial w_i(k)} \quad (2)$$

여기서,  $\eta$ 는 학습율 계수이다.

또한, chain-rule에 의해

$$\frac{\partial J_{k+1}}{\partial w_i(k)} = \frac{\partial J_{k+1}}{\partial e_{k+1}} \cdot \frac{\partial e_{k+1}}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_i(k)} \quad (3)$$

로 구해진다. 여기서,  $u_k$ 는 k번째 시간에서의 제어기 출력 값이다.

이제 식 (3)의 우변의 각 항들을 계산해야 하는데, 이 중 첫째 항  $\frac{\partial J_{k+1}}{\partial e_{k+1}}$ 은 식(1)에서 쉽게 구해지고, 셋째 항  $\frac{\partial u_k}{\partial w_i(k)}$ 은 학습 제어기의 입출력 관계로부터 간단히 구할 수 있다.

그러나, 식 (3)의 우변의 두 번째 항인  $\frac{\partial e_{k+1}}{\partial u_k}$ 은 중요한 의미를 가질 뿐 아니라, 구하기 어려운 값이다.

이 값은  $e_{k+1} = y_{k+1}^d - y_{k+1}$ 이므로  $-\frac{\partial y_{k+1}}{\partial u_k}$ 와 같은 것으로, k번째 플랜트 제어 입력의 변화에 따른 출력의 변화율을 의미한다. 즉, 제어 입력을 증가 또는 감소시킬 경우 플랜트 출력이 증가하는지 또는 감소하는지의 방향을 결정하는 값인 것이다. 따라서,  $\frac{\partial e_{k+1}}{\partial u_k}$ 의 크기보다는 바람직한 제어입력의 변화 방향을 결정하는 이 값의 부호가 더욱 중요한 의미를 갖고, 그 부호는 플랜트의 특성과 밀접한 관계를 가지므로 제어 성능을 최적화 하는 방향으로 퍼지 제어기를 학습시키는데 있어 매우 중요한 파라미터이다.

그런데, 기존의 연구에서는 이 값의 부호를 안다고 가정하거나<sup>[11]</sup>, 플랜트 자코비안 모델을 통해 결정하는 경우<sup>[7]</sup>가 있는데, 이는  $\frac{\partial e_{k+1}}{\partial u_k}$  값을 모델 등을 통해

미리 주어진다고 가정하는 것으로 특성을 잘 모르는 플랜트의 제어를 위한 자기구성 제어기를 설계하고자 하는 관점에서 보면 이는 그 기본적 동기에 위배되는 것으로 합당하지 않다. 또한,  $\frac{\partial e_{k+1}}{\partial u_k} \approx \frac{e_{k+1} - e_k}{u_k - u_{k-1}}$  과 같은 근사화를 하고, 이로부터 부호 만을 취하는 방법<sup>[10]</sup>이 있는데, 이는 인접한 시간에서의 오차와 입력의 변화율을 구한 것으로 엄밀하게 말하면  $\frac{\partial e_{k+1}}{\partial u_k}$  의 합당한 근사화라 할 수 없다. 그 이유는  $\frac{\partial e_{k+1}}{\partial u_k}$  가 플랜트의 상태에 관한 함수로 같은 하나의 상태에서 구한 오차와 입력의 변화율로부터 구해야 하는데, 인접한 시간에서의 동적 플랜트의 상태는 일반적으로 같지 않기 때문이다.

이러한 문제를 해결하기 위하여 본 논문에서는 다음과 같은 관찰을 통하여, 하나의 새로운 퍼지 제어기의 자기 구성 방법을 제안한다. 한 예로 사람이 역진자(Inverted Pendulum)를 똑바로 세우기 위해 학습하는 과정을 살펴보자. 특성을 잘 모르는 역진자 시스템의 제어를 위해 사람은 일단 몇번의 시도를 통해 임의의 입력을 주고 그때마다의 시스템 응답을 관찰한다. 이렇게 시행착오를 거치면서 사람은 막대의 무게나 길이 등을 알아내는 것(즉, 시스템 파라미터 또는 모델을 동정하는 것)이 아니고, 가해진 입력에 대한 출력의 변화를 보고 성능 향상을 위한 입력 증감의 바람직한 방향을 학습하고, 학습된 방향으로 입력의 크기를 조금씩 변화시켜 가면서 원하는 제어 성능을 갖도록 함을 관찰할 수 있다. 이는 파블로프의 조건 반사의 학습 과정과도 매우 유사한 것이다.

본 논문에서 제안하는 자기구성 퍼지 제어기는 상기한 바와 같은 반복적인 인간의 학습 양식을 기반으로 하여 적합한  $\frac{\partial e_{k+1}}{\partial u_k}$  의 부호, 즉 시스템 상태에 따른 바람직한 제어기 출력의 변화 방향을 온라인 학습 방법으로서, 강화학습 방법에서와 같이 오류 신호 등을 이용한 반복적 학습을 하는 방식을 갖는다. 제안된 방법은 시스템 모델을 필요로 하지 않을 뿐 아니라 설정해야 할 계수가 적고 간단한 구조를 가지므로 설계하기 쉬운 장점을 갖는다.

제안된 방법에 대해 구체적으로는, 2장에서 제안된

자기구성 퍼지 제어기의 구조 및 반복적 시행착오 과정을 통한 학습 방법에 대해 기술하고, 역진자 시스템에 대한 모의실험을 통하여 제안된 방법의 타당성을 3장에서 보이며, 끝으로 4장에서 결론을 맺도록 한다.

## II. 반복적 학습에 의한 자기구성 퍼지 제어기

반복적 학습을 기반으로 하는 자기구성 퍼지 제어기의 전체적인 구조는 그림 1과 같이 크게 퍼지 제어기 부분(FLC)과 규칙조정부(Rule modifier)로 나뉜다. 각 부분의 개괄적인 동작 원리는 다음과 같다. 퍼지 제어기는 플랜트 출력( $y$ )과 그 목표치( $y_d$ )와의 차로부터 계산된 오차 및 오차의 변화율 등을 입력으로 하여, IF-THEN 형태의 규칙들로부터 추론하여 비퍼지화된 제어 입력( $u$ )을 생성한다. 규칙조정부에서는 과거와 현재의 플랜트의 상태와 그 때마다 생성된 제어 입력으로부터 바람직한 제어 입력의 변화 방향을 학습하고 이를 바탕으로 퍼지 규칙을 수정한다. 또한, 제어 과정에서 플랜트의 출력이 일정 범위를 벗어나거나(오류(Failure) 발생), 오류 발생 없이 일정 시간이 지나면(Time-Out) 재시도(Retry) 신호를 발생시켜 다시 초기화된 플랜트 상태에서 제어 및 학습을 반복하게 된다.

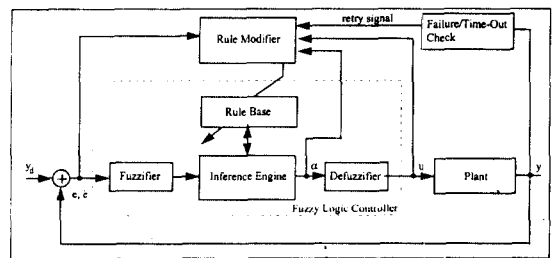


그림 1. 제안된 자기구성 퍼지 제어기의 블록도  
Fig. 1 Block diagram of the proposed SOFLC

### 1. 퍼지 제어기

퍼지 제어기의 제어 규칙은 IF-THEN 형태를 갖으며, 예를 들어 2 입력-1 출력 제어기의 경우에는 퍼지 입력변수를  $x, y$ , 출력변수를  $u$ 로 나타낼 때 다음과 같은 형태의 규칙으로 표현된다.

$i$ 번째 규칙: IF  $x$  is  $A_j$  and  $y$  is  $B_k$  THEN  $u$  is  $w_i$ ,  
 ( $i = 1, 2, \dots, n$ )

여기서,  $A_j, B_k$ 는 각 입력변수의 퍼지값 (또는, Label) 들이고, 규칙의 총 개수  $n$ 은 각 입력변수의 퍼지값들의 개수의 곱(즉, 퍼지값들의 가능한 조합의 개수)으로 정해진다.

퍼지값들의 소속함수 형태는 후건부(THEN-part)의  $w_i$ 는 crisp한 단일값을 갖는 것으로 하고, 전건부(IF-part)의  $A_j, B_k$ 들의 소속함수는 삼각형 모양을 갖는 것으로 정한다<sup>[2]</sup>(참조 그림 2).

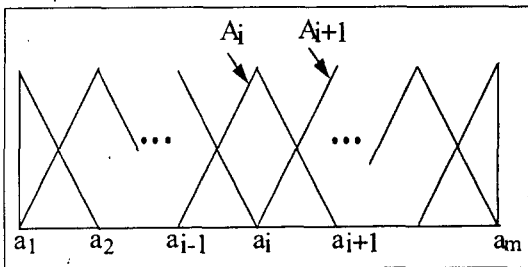


그림 2 퍼지 제어 규칙의 전건부의 삼각형 소속함수들  
 Fig. 2 Triangular membership functions of premise part of fuzzy control rules

그림 2에서 퍼지집합  $A_j$ 의 소속함수를 수식으로 나타내면

$$\mu_{A_j}(x) = \begin{cases} (x - a_{j-1}) / (a_j - a_{j-1}), & \text{만일 } a_{j-1} \leq x < a_j \\ (-x + a_{j+1}) / (a_{j+1} - a_j), & \text{만일 } a_j \leq x < a_{j+1} \\ 0, & \text{이외의 범위} \end{cases} \quad (4)$$

이며,  $A_1, A_m$ 는 대집합의 범위 제한에 의해 각각 좌우가 잘린 삼각형 모양을 갖는다.

이와 같이 퍼지변수의 퍼지값들이 동일한 형태의 삼각형 소속함수를 갖는 경우 몇 가지 특성에 대해 살펴보면, 대집합(Universe of discourse)을  $U = [x_{\min}, x_{\max}]$  라하고,  $m$  개의 소속함수로 나눌 때  $a_{j+1} - a_j = (x_{\max} - x_{\min}) / (m - 1)$ , ( $j = 1, 2, \dots, m$ )이고, 모든  $x \in U$ 에 대해,  $\sum_{i=1}^m \mu_{A_i}(x) = 1$ 이며, 모든  $x \in [a_j, a_{j+1}]$ 에 대하여,  $\mu_{A_j}(x) = 1 - \mu_{A_{j+1}}(x)$ 인 성질을 갖는다.

한편, 퍼지 제어 규칙 이외의 퍼지 제어기 각 부분에 대하여는, 퍼지화(Fuzzification)는 단일값(Singleton) 퍼지화 방법으로 하고, 비퍼지화(Defuzzification) 및 추론 방법은 Takagi-Sugeno 방법<sup>[4]</sup>으로 설정한다.

이상과 같이 설정된 퍼지 제어기의 추론 및 비퍼지화 과정을 다음과 같은 2개의 규칙에 대해 예를 들어 살펴보자.

규칙 1: IF  $x$  is  $A_1$  and  $y$  is  $B_1$  THEN  $u$  is  $w_1$   
 규칙 2: IF  $x$  is  $A_2$  and  $y$  is  $B_2$  THEN  $u$  is  $w_2$

그림 3에 나타낸 바와 같이, 단일 값 입력  $x_0, y_0$ 에 대해 각 규칙의 적합도( $\alpha_1, \alpha_2$ )는 전건부 소속함수들의 곱으로부터  $\alpha_1 = \mu_{A_1}(x_0) \cdot \mu_{B_1}(y_0)$ ,  $\alpha_2 = \mu_{A_2}(x_0) \cdot \mu_{B_2}(y_0)$ 로 계산되고, 이 적합도와 crisp한 단일 값을 갖는 후건부의 소속함수를 곱(Product) 한 후, 그 결과들을 가중 평균(weighted average)을 취하면, 최종적으로

$$\text{얻어지는 제어기 출력}(u) \text{는 } u = \frac{\alpha_1 \cdot w_1 + \alpha_2 \cdot w_2}{\alpha_1 + \alpha_2} \text{로 계산된다.}$$

이러한 결과를 일반화 시켜,  $r$ 개의 퍼지 입력 변수를 갖는  $n$ 개의 규칙에 대해 적용하면  $i$ 번째 규칙의 적합도는  $\alpha_i = \prod_{j=1}^r \mu_{ij}(x_j)$ 이고, 추론을 통해 최종적으로 얻어지는 제어기 출력은

$$u = \frac{\sum_{i=1}^n (\prod_{j=1}^r \mu_{ij}(x_j)) \cdot w_i}{\sum_{i=1}^n (\prod_{j=1}^r \mu_{ij}(x_j))} = \frac{\sum_{i=1}^n \alpha_i \cdot w_i}{\sum_{i=1}^n \alpha_i} \quad (5)$$

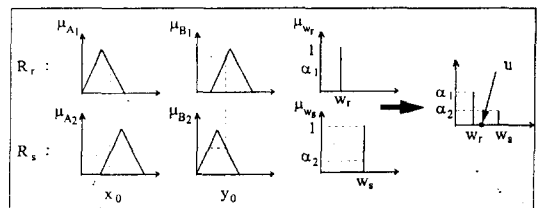


그림 3 퍼지추론 과정의 도식적 표현  
 Fig. 3 Diagrammatic representation of fuzzy inference procedure

이다. 단, 여기서  $\mu_{ij}(x_j)$ 는  $i$ 번째 규칙의  $j$ 번째 입력 퍼지변수에 대한 소속함수 값이고,  $x_j$ 는  $j$ 번째 입력 퍼지변수에 해당하는 crisp 입력 값이다.

이상과 같이 설정된 퍼지 제어기의 학습 변수를 후건부의 소속함수, 즉 단일값을 갖는  $w_i$  ( $i=1, 2, \dots, n$ )로 하면, 식 (1)과 같은 성능 지수에 대한 기울기 강하법의 적용시 식 (3)의 우변의 첫째항과 셋째항은 다음과 같이 계산된다.

$$\frac{\partial J_{k+1}}{\partial e_{k+1}} = e_{k+1} \quad (6)$$

$$\frac{\partial u_k}{\partial w_i(k)} = \frac{\prod_{j=1}^r \mu_{ij}(x_j(k))}{\sum_{i=1}^n \prod_{j=1}^r \mu_{ij}(x_j(k))} = \frac{\alpha_i(k)}{\sum_{i=1}^n \alpha_i(k)} \quad (7)$$

$\frac{\partial e_{k+1}}{\partial u_k}$ 의 부호를 결정하기 위한 적절한 방법에 대해서는 다음 절에서 기술하기로 하고, 이와 같은  $\frac{\partial e_{k+1}}{\partial u_k}$ 의 부호 값을 이용한 학습규칙은 식 (2)와 식 (6), (7)로부터 다음과 같이 정리할 수 있다.

$$\Delta w_i = -\eta \cdot \frac{\partial J_{k+1}}{\partial w_i(k)} = -\eta \cdot e_{k+1} \cdot \text{sgn}\left(\frac{\partial e_{k+1}}{\partial u_k}\right) \cdot \frac{\alpha_i(k)}{\sum_{i=1}^n \alpha_i(k)} \quad (8)$$

## 2. 제안된 $\frac{\partial e_{k+1}}{\partial u_k}$ 의 부호 학습 방법

앞서 언급한대로  $\frac{\partial e}{\partial u}$ 는 상태에 관한 함수이므로, 각 상태에 대해  $\frac{\partial e}{\partial u}$  값을 저장하고 있다가, 학습과정에서는 그때 그때의 플랜트 상태에 따라 저장된 값을 참조하여  $k$ 번째 시간에서의  $\frac{\partial e_{k+1}}{\partial u_k}$  값으로 사용하여야 한다. 그러나, 모든 플랜트 상태에서의  $\frac{\partial e}{\partial u}$  값을 저장하려면 무한히 많은 메모리를 필요로 할 뿐 아니라, 학습과정에서 제어되고 있는 플랜트로부터 똑같은 상태를 찾아내는 것도 어려운 문제점을 갖는다.

이러한 문제점을 해결하는 하나의 방법으로 본 논문에서는, 상태 공간을 몇개의 영역으로 세분화하고

나닌 하나의 상태 영역에 대해  $\frac{\partial e}{\partial u}$ 의 대표값 하나를

갖게 하는 방법을 제안한다. 상태영역의 분할을 위해서는 규칙조정부 내에 별도의 블록을 필요로 하지 않고, 퍼지 제어기의 전건부 소속함수들에 의한 상태영역 분할을 그대로 사용한다. 즉, IF-THEN 형태의 퍼지 제어규칙의 정성적인 의미는 “만일 이러 이러한 상태이면, 이런 제어입력을 생성한다”는 식이기 때문에, 퍼지 제어규칙은 전건부의 소속함수들에 의해 규칙의 개수만큼 플랜트의 상태 영역을 나눈 것과 같은 점을 이용하는 것이다.

따라서,  $\frac{\partial e}{\partial u}$ 에 대한 대표값은 분할된 상태영역 각각에 할당되는 것으로 규칙의 수와 같게되며, 대표값은 나닌 상태영역 중 하나의 영역에서의  $\frac{\partial e}{\partial u}$  값을 의미하는데 그 대표상태(Representative state)는 각 상태 영역의 중심 즉, 전건부 소속함수 값들이 모두 1이 되는 상태로 정한다.

이러한 대표값을 얻기 위해서는 학습 과정에서 제어되고 있는 플랜트 상태로부터 대표상태인 것을 찾아내야 하는데, 대표 상태와 똑같은 상태를 얻는 것이 일반적으로 매우 어려우므로, 오프라인으로 각 상태에 대한 제어를 통해 대표값을 구할 수도 있겠으나, 이 또한 실제적으로는 불가능하므로, 본 논문에서는 인접한 시간에서의 정보를 이용하는 다음과 같은 온라인 방법을 제안한다.

일반적으로 인접한 시간에서의 상태가 서로 다를 뿐 아니라, 대표 상태와도 다르므로 이들 상태와 대표 상태와의 유사정도(Similarity)에 따라  $\frac{\partial e}{\partial u}$  값을 결정하도록 한다. 예를 들어, 그림 4에서와 같이  $k-1$  시간에서  $e_{k-1}, \dot{e}_{k-1}$ 인 상태에서 퍼지 추론에 의해 결정된  $u_{k-1}$ 이 플랜트에 인가되고, 이  $u_{k-1}$  입력에 의해 다음 시간에서 플랜트의 상태가 변하여  $k$  시간에서  $e_k, \dot{e}_k$ 로 변하고, 이로부터 결정된  $u_k$ 에 의해 다시  $k+1$  시간에서  $e_{k+1}, \dot{e}_{k+1}$ 로 변하는 일련의 과정을 생각해보자. 이 경우  $u_{k-1}$ 을 결정하는데 초기상태 조건은  $e_{k-1}, \dot{e}_{k-1}$ 인데 이 상태가  $i$ 번째 상태영역의 대표값과 유사한 정도는  $e_{k-1}, \dot{e}_{k-1}$ 이 입력되었을때  $i$ 번째 퍼지규칙의 적합도( $\alpha_i(k-1)$ )로 정할 수 있다. 이와 같이 인접

한 시간에서의 초기상태  $e_k, \dot{e}_k$ 에 대해  $i$ 번째 상태영역의 대표값과 유사한 정도를 적합도( $\alpha_i(k)$ )라 하면, 이 두 적합도 값의 곱 즉,  $\alpha_i(k-1) \cdot \alpha_i(k)$ 는 인접한 시간에서의 두 초기 조건이 대표 상태와의 유사정도를 나타내게 된다.

다음은  $\frac{\partial e}{\partial u}$ 의 근사값 계산에 대해서 살펴보자.  $\frac{\partial e}{\partial u}$ 의 의미는 앞서 언급한바와 같이 제어입력의 변화에 따른 플랜트 상태의 변화를 또는 오차의 변화율로써, 이 값의 부호는 입력을 증가시킬 때 오차가 줄어드는지(즉, 바람직한 방향으로의 제어가 되는지) 입력을 감소시킬 때 그러한지를 나타낸다. 예를 들어, 입력을 증가시켰을 때 오차가 감소하였다면  $\frac{\partial e}{\partial u}$  값의 부호는 -가 된다. 그런데, 동적 플랜트의 경우는 플랜트의 다이내믹 특성으로 인해 입력이 증가하더라도 오차는 감소하지 않게 될 수도 있다. 이러한 문제를 해결하기 위해서는 시스템 동특성에 대한 예측이 필요한데, 본 논문에서 다루는 문제는 플랜트 특성을 모르는 경우이므로 다음과 같은 단순한 예측 방법을 사용한다.

$\frac{\partial e}{\partial u}$ 를 그림 4와 같은 인접한 시간에서의 정보로부터 계산하기 위해서는 예측을 통하여,  $e_{k-1}, \dot{e}_{k-1}$  상태에서  $u_{k-1}$ 을 인가한 후  $k$  시간에서 더 이상의 제어입력을 가하지 않았을 경우  $k+1$ 에서의 예상되는 오차

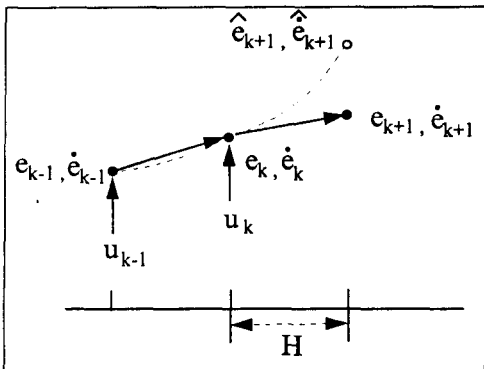


그림 4. 인접한 시간에서의 상태와 제어입력 예  
Fig. 4 Example of states and control inputs between adjacent time steps

$\hat{e}_{k+1}$ 를 계산하고,  $k$  시간에서  $u_k$ 를 인가하였을 때의 오차  $e_{k+1}$ 로부터  $k$ 번째 시간에서의  $\frac{\partial e}{\partial u}$ 는

$$\frac{\partial e_{k+1}}{\partial u_k} \approx \frac{e_{k+1} - \hat{e}_{k+1}}{u_k - u_{k-1}} \quad (9)$$

와 같이 근사적으로 계산할 수 있다.

식 (9)는 입력을  $u_{k-1}$ 에서  $u_k$ 로의 변화가 다음 시간에서 오차변화에 바람직한 방향인지를 판정할 수 있도록 한다. 예를 들어,  $u_{k-1}$ 보다  $u_k$ 를 증가시켰는데 예상오차  $\hat{e}_{k+1}$ 보다 실제오차  $e_{k+1}$ 가 작다면  $\frac{\partial e_{k+1}}{\partial u_k}$ 는 음수가 되어 입력의 증가가 바람직함을 나타내는 것이다.

이상의 내용을 종합하여,  $k+1$ 번째 시간에서  $i$ 번째 상태영역의  $\frac{\partial e}{\partial u}$ 의 부호를 결정하기 위한 대표값( $\phi_i$ )은

$$\phi_i(k+1) = \alpha_i(k-1) \cdot \alpha_i(k) \cdot \frac{e_{k+1} - \hat{e}_{k+1}}{u_k - u_{k-1}} + \phi_i(k) \quad (10)$$

로 구한다. 식 (10)에서  $k$ 번째 시간에서  $\frac{\partial e}{\partial u}$ 의 근사값과  $\alpha_i(k-1) \cdot \alpha_i(k)$ 를 곱하는 것은  $\frac{\partial e}{\partial u}$ 의 근사값을 구하는 두 인접한 상태가  $i$ 번째 대표 상태와의 유사한 정도를 고려하여, 이 근사값을  $i$ 번째 대표값으로 하기 위함이고,  $\phi_i(k)$  값을 더하는 것은 비슷한 상태에 대한 과거의 대표값을 이용하여 대표값( $\phi_i$ )을 강화학습시키는 의미를 갖는다.

한편, 예상되는 오차의 예측은 외삽(extrapolation) 방법을 사용하여 다음과 같이 구한다. 그림 4에서의 인접한 두 점  $(e_{k-1}, \dot{e}_{k-1}), (e_k, \dot{e}_k)$ 에 대해 매개 변수  $t$ 를 두어  $t=0$ 일때  $(e_{k-1}, \dot{e}_{k-1})$ 를,  $t=H$ (단,  $H$ 는 제어기의 샘플링 주기)일때  $(e_k, \dot{e}_k)$ 를 지나는, 즉 인접한 두 점에서의 위치와 속도를 만족하는 다항식은 주어진 경계 조건식이 4개이므로 다음과 같은 3차식으로 주어지고,

$$e(t) = at^3 + bt^2 + ct + d \quad (11)$$

각 계수는

$$\begin{aligned}
a &= (H(\dot{e}_k + \dot{e}_{k-1}) - 2(e_k - e_{k-1}))/H^3, \\
b &= (-H(\dot{e}_k + 2\dot{e}_{k-1}) + 3(e_k - e_{k-1}))/H^2, \\
c &= \dot{e}_{k-1}, \\
d &= e_{k-1}
\end{aligned} \quad (12)$$

로 계산되고,  $\hat{e}_{k+1}$ 는 식 (11)에서  $t=2H$  일 때의 값으로

$$\hat{e}_{k+1} = 8aH^3 + 4bH^2 + 2cH + d \quad (13)$$

로 계산된다. 이와 같은 외삽에 의한 예측 방법이 적절할 것인가는 모의 실험 부분에서 검증해 보았다.

그런데, 식 (10)으로 주어지는  $i$ 번째 상태영역의  $\frac{\partial e}{\partial u}$ 에 대한 대표값( $\phi_i$ )은 각 상태에서 구한 정확한  $\frac{\partial e}{\partial u}$ 가 아니고, 근사값이므로 퍼지규칙 조정을 행할 때는 그 크기보다는 부호, 즉 제어입력의 바람직한 변화 방향에 중점을 두어 식 (8)에서와 같이 이 값의 부호만을 사용하여 결과적으로 학습규칙은 다음과 같이 정리된다.

$$\Delta w_i = -\eta \cdot \frac{\partial J_{k+1}}{\partial w_i(k)} = -\eta \cdot e_{k+1} \cdot \text{sgn}(\phi_i(k)) \cdot \frac{\alpha_i(k)}{\sum_{i=1}^n \alpha_i(k)} \quad (14)$$

만일,  $k$  번째 시간에 대해서 성능 지수를 오차뿐 아니라 오차의 변화율도 최소화되도록

$$J_k = \frac{1}{2} (e_k^2 - \dot{e}_k^2) \quad (15)$$

로 정하면, 상기한 내용을 유사하게  $\dot{e}_k$ 에 대해 적용하여 학습규칙은

$$\Delta w_i = -(\eta e_{k+1} \cdot \text{sgn}(\phi_i(k)) + \delta \dot{e}_{k+1} \cdot \text{sgn}(\dot{\phi}_i(k))) \cdot \frac{\alpha_i(k)}{\sum_{i=1}^n \alpha_i(k)} \quad (16)$$

가 된다. 여기서,  $\dot{\phi}_i(k)$ 는  $k$ 번째 시간에서  $i$ 번째 상태 영역의  $\frac{\partial \dot{e}}{\partial u}$ 에 대한 대표값이고,  $\eta$ ,  $\delta$ 는 학습율이다.

### 3. 규칙조정부 알고리즘

규칙조정부는 성능평가에 따라 제어규칙을 조정하는 부분으로, 시스템 상태에 따른 제어기 출력( $u$ )의 바람직한 변화 방향을 학습하고, 플랜트 출력에 따라 발생하는 재시도 신호에 의해 반복적으로 퍼지 제어규칙을 조정한다.

재시도 신호는 플랜트의 출력이 규정된 범위를 벗어나거나(Failure), 제어시간이 일정 주기를 지나면(Time-out) 발생되며, 재시도 신호가 발생되면 플랜트의 상태를 다시 초기화하여 반복적인 학습 과정을 되풀이한다. 또한, 식 (10)에서 알 수 있듯이, 학습시 초기의 두 스텝동안에는  $\frac{\partial e}{\partial u}$ 의 갱신이 이루어지지 않고 오차 및 오차의 변화율 등과 제어입력의 과거 값을 저장하고 있다가, 3번째 단위 시간부터 학습을 한다.

전 절의 내용을 종합하여 규칙조정부 전체 알고리즘을 정리하면 다음과 같다.

#### [단계 1] 퍼지 제어기 초기화

식 (4)의 삼각형 형태를 갖는 전건부 소속함수들을 정해진 제어기 입력에 대해 정해진 퍼지 값 개수만큼 전체 대집합에 대해 정의하고, 후건부 소속함수의 위치( $w_i$ ) 및 대표값( $\phi_i$ ,  $\phi_{di}$ )을 초기화한다. 기준입력을 정한다.

#### [단계 2] 제어 대상 플랜트의 초기화

제어 대상 플랜트의 상태를 초기화하고,  $u = 0$ ,  $\alpha_i(0) = 0$ ,  $k = 1$ 로 초기화한다.

#### [단계 3] 상태 계환을 통해 $e_k$ , $\dot{e}_k$ 를 구한다.

#### [단계 4] $u_{k-1}$ , $e_k$ , $\dot{e}_k$ , $\alpha_i(k-1)$ 를 저장한다.

#### [단계 5] $k > 2$ 이면 단계 6으로, 그렇지 않으면 단계 7로 간다.

#### [단계 6] 제어규칙 및 $\frac{\partial e}{\partial u}$ 의 대표값 학습

식 (10), (14) 또는 식 (16)을 이용하여 대표값( $\phi_i(k)$ ,  $\phi_{di}(k)$ )과 후건부 소속함수 위치( $w_i(k)$ )를 갱신한다.

#### [단계 7] 식 (5)와 같이 퍼지추론을 통해 제어입력( $u_k$ )과 각 규칙의 소속정도 $\alpha_i(k)$ 를 구한다.

#### [단계 8] 제어대상 플랜트에 $u_k$ 를 인가한다.

#### [단계 9] 재시도 신호 발생 및 성능지수 계산

플랜트 상태가 특정범위를 벗어나면 단계 2

로 가고,  $k > T$ 이면 성능지수  $J (= \sum_{k=1}^T J_k)$ 를 계산하여  $J < \epsilon$ 이면 반복학습을 종료한다. 그렇지 않으면  $k = k + 1$ 로 하고 단계 3으로 간다.

### III. 모의실험

제안된 반복적 학습을 통한 자기구성 퍼지 제어기의 성능 평가를 위해 그림 5와 같은 역진자 시스템<sup>[13]</sup>에 대해 모의실험을 행한다. 역진자 시스템의 각 파라미터는 표 1과 같다.

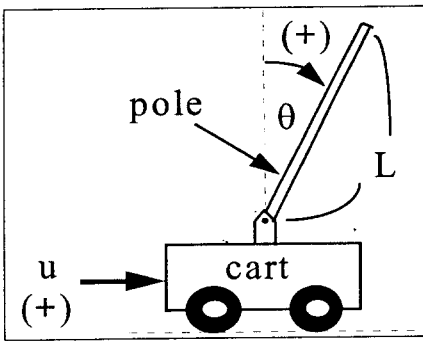


그림 5. 역진자 시스템  
Fig. 5 Inverted pendulum system

표 1. 역진자 시스템 파라미터들  
Table 1. Parameters of an inverted pendulum system

파라미터	설정값
막대 무게(m)	0.1 kg
cart 무게(M)	1 kg
막대 길이(L)	1 m
샘플링 주기(H)	0.01 sec
중력가속도(g)	9.8 m/s <sup>2</sup>

또한, 막대의 각도에 대한 역진자 시스템의 모델식은 다음과 같다.

$$\ddot{\theta} = \frac{(m + M)g \sin \theta - \cos \theta (u + m \frac{L}{2} \dot{\theta}^2 \sin \theta)}{\frac{2}{3} (m + M)L - m \frac{L}{2} \cos^2 \theta} \quad (17)$$

퍼지 제어기의 구성에 있어서, 기준입력( $y_d$ )은  $0^\circ$ 이고 입력은 오차( $e$ )와 오차의 변화율( $\dot{e}$ )로써 각각의 퍼지 값(또는 label)수는 7개씩으로 하고, 각각의 대집합의 범위는 오차의 경우  $[-0.2618, 0.2618]$  rad이고, 오차의 변화율에 대해서는  $[-2.618, 2.618]$  rad/sec로 정한다. 출력은 차체(cart)에 가해지는 힘( $u$ )으로  $[-10, 10]$  N 범위로 제한한다.

그리고, 규칙조정부의 파라미터로써 학습율들은 각각  $\eta = 0.8, \delta = 0.1$ 로 정하며, 오류신호는 막대의 각도가  $\pm 0.349$  rad ( $= \pm 20^\circ$ )의 범위를 벗어날때로 하고, 한 반복학습 주기시간  $T = 5$  sec로 한다. 매 반복시의 초기 막대의 각도는  $+0.1745$  rad ( $= 10^\circ$ )로 정한다.

또한, 각 규칙 후건부의 단일 값( $w_i$ )은  $[-10, 10]$  N의 범위에서 랜덤하게 초기화하였으며,  $i$ 번째 상태영역의  $\frac{\partial e}{\partial u}$ 에 대한 대표값( $\phi_i$ )은  $[-0.01, 0.01]$ 의 범위에서 랜덤하게 초기화하고  $\frac{\partial \dot{e}}{\partial u}$ 에 대한 대표값( $\phi_{di}$ )은 0

으로 초기화하였다. 초기화된 값들은 그림 6과 같으며, 7개로 나누어진  $e$ 와  $\dot{e}$  (ce)에 대한 퍼지값들은 편의상 각각  $A_1 \sim A_7, B_1 \sim B_7$ 로 표시하였다. 따라서, 그림 6(b)의 각 칸은 각 행, 열의 해당 소속함수를 갖는 하나의 퍼지규칙에 해당된다. 그런데, 제어입력의 변화 방향은  $\phi_i$ 의 크기가 아닌 부호에 의해 결정되는 것이므로, 이러한  $\phi_i$ 의 초기화는 초기에 제어입력의 변화 방향을 모르는 것으로 정한 것과 같은 의미를 갖는다. 이와 같은 랜덤 초기화는 자기구성 퍼지 제어기의 학습에 있어 매우 까다로운 조건으로 제안된 방법의 성능평가를 위해 충분한 조건이다.

이와 같이, 설정된 퍼지 제어기와 규칙조정부 파라미터에 대해 식 (3)과 같이 오차의 크기만을 고려한

ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	-99	-93	-56	-60	+89	-11	+39
B2	-91	+63	+52	+91	-14	+67	+62
B3	+20	+19	+44	-18	+34	-01	-31
B4	-47	-15	-67	+29	-99	-46	-22
B5	+16	-44	+61	-33	-51	-89	+96
B6	-18	-85	+94	-19	-60	-50	+77
B7	+47	+06	-73	-03	-49	+40	+77

(a)



ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	-9.78	-3.29	-2.88	+0.73	+4.00	-4.50	-7.82
B2	+1.28	-6.69	+3.71	+6.55	-5.61	+9.05	+8.46
B3	-0.97	+3.23	+0.98	-7.72	-7.57	-0.50	+1.28
B4	+7.37	-6.40	+3.89	+0.76	+2.46	+5.74	-0.77
B5	-2.46	+2.06	-6.58	-4.10	-1.82	-1.72	-0.11
B6	-9.96	-8.60	-4.91	-2.89	-6.05	-5.76	+5.16
B7	+8.10	-0.77	-9.29	-0.83	+1.93	-7.70	+2.98

(b)

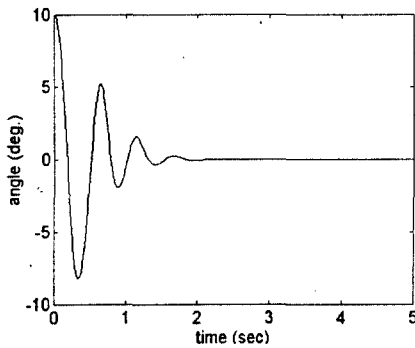
그림 6.  $\phi_i$ 와  $w_i$ 의 랜덤 초기화 값

(a) 10,000 배한 대표값( $\phi_i$ )들, (b)  $w_i$  값들

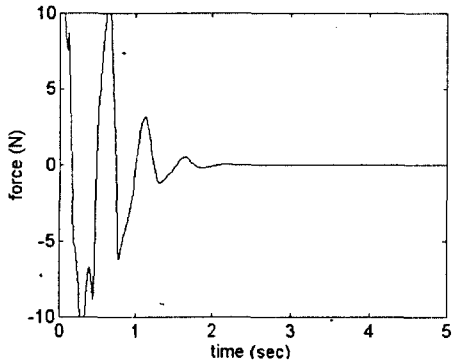
Fig. 6 Randomly initialized values of  $\phi_i$  and  $w_i$

(a) Representative values( $\phi_i$ ) multiplied by 10,000,

(b) Values of  $w_i$ ;



(a)



(b)

ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	-99	-93	-56	-60	89	-11	39
B2	91	63	57	113	11	67	62
B3	121	56	223	265	140	1	29
B4	2	502	69	11408	213	78	124
B5	16	98	495	495	288	76	268
B6	18	84	159	56	52	7	141
B7	47	06	73	03	49	40	77

(c)

ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	-9.78	-3.29	-2.88	+0.73	+4.00	-4.50	-7.82
B2	+1.12	-6.35	+6.18	+7.84	-4.68	+9.05	+8.46
B3	+10	+10	+10	+1.82	-10	+1.81	+3.69
B4	-7.26	+10	+10	+0.00	-10	-10	-10
B5	-2.39	+10	+10	-7.82	-10	-10	-10
B6	-9.96	-10	+8.41	-4.53	-1.45	+10	-10
B7	+8.10	-0.77	-9.29	-0.83	+1.93	-7.7	+2.17

(d)

그림 7. 성능지수  $J_k = \frac{1}{2} e_k^2$ 에 대한 150회 반복 학습의 결과

(a) 시간에 따른 막대의 각도, (b) 시간에 따른 제어입력

(c) 10,000배한 학습 후  $\phi_i$ , (d) 학습 후  $w_i$

Fig. 7 Learning results after 150th iteration with performance index of  $J_k = \frac{1}{2} e_k^2$

(a) Pole angle versus time, (b) Control input versus time

(c)  $\phi_i$  multiplied by 10,000 after learning, (d) Learned  $w_i$ ;

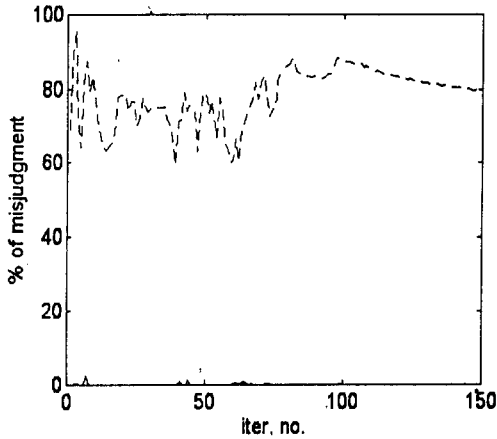
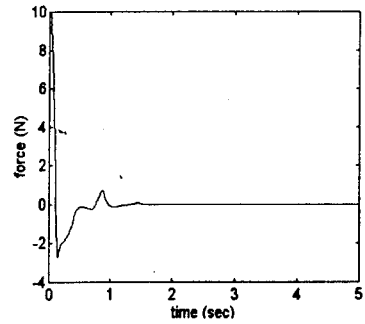


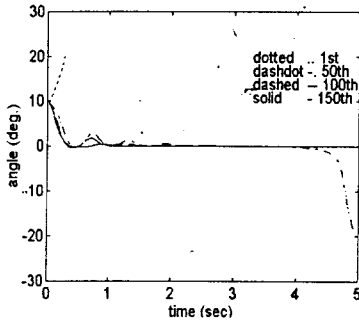
그림 8. 바람직한 제어 방향의 오판율의 비교  
 Fig. 8 Comparison of the rate of misjudgement for the proper direction of control



(c)

ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	+99	+93	+56	+60	+89	+11	+39
B2	+91	+63	+52	+91	+14	+57	+62
B3	+104	+36	+51	+21	+34	+01	+31
B4	+07	+425	+328	+21832	+22	-45	+01
B5	+16	+119	+556	+176	+115	-13	+392
B6	+18	+85	+94	+19	+57	+18	+96
B7	+47	+106	+73	+03	+49	+40	+77

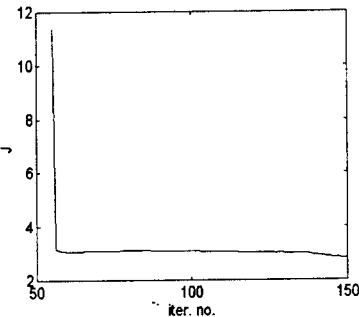
(d)



(a)

ce \ e	A1	A2	A3	A4	A5	A6	A7
B1	-9.78	-3.29	-2.88	+0.73	+4.00	-4.50	-7.82
B2	+1.25	-6.69	+3.71	+6.55	-5.61	+9.05	+8.46
B3	+10	+10	+6.26	-2.12	-7.56	-0.50	+1.28
B4	-2.88	+10	+9.68	-0.002	+1.62	+6.79	+6.87
B5	-2.33	+9.99	-3.87	-9.99	-6.58	+6.29	-10
B6	-9.5	-8.7	-4.92	-2.89	-6.85	-1.69	-10
B7	+8.10	-0.77	-9.29	-0.83	-1.93	-7.7	+2.98

(e)



(b)

그림 9. 성능지수  $J_k = \frac{1}{2} (e_k^2 + \dot{e}_k^2)$ 에 대한 반복 학습의 결과

- (a) 반복횟수에 따른 막대의 각도,
- (b) 반복횟수에 따른 성능지수
- (c) 150회째 시간에 따른 제어입력,
- (d) 10,000배한 150회 학습 후  $\phi_i$ , (e) 150회 학습 후  $w_i$

Fig. 9 Learning results with performance index of  $J_k = \frac{1}{2}$

$$(e_k^2 + \dot{e}_k^2)$$

- (a) Pole angle at each iterations,
- (b) Value of performance index versus iterations,
- (c) Control input versus time at 150th iteration,
- (d)  $\phi_i$  multiplied by 10,000 after 150th iteration,
- (e)  $w_i$  after 150th iteration

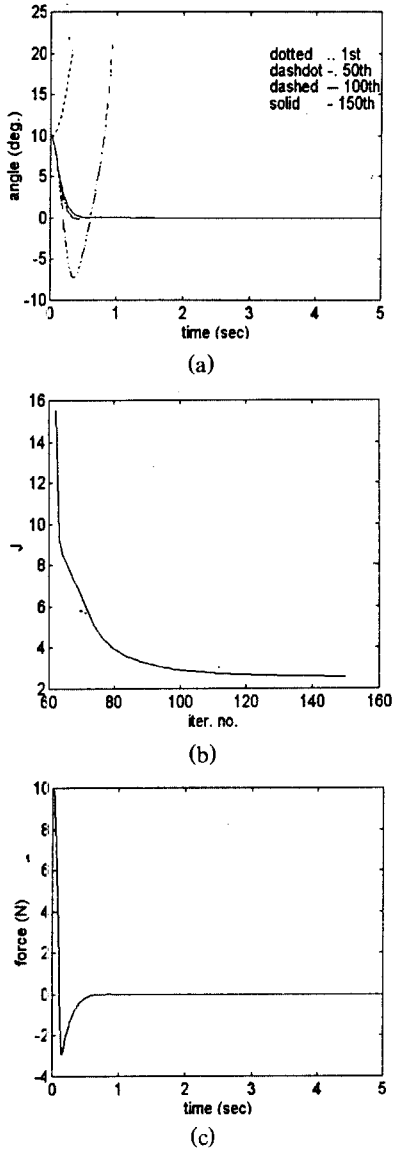


그림 10.  $w_i=0$ 로 초기화한 경우  $J_k = \frac{1}{2} (e_k^2 + \dot{e}_k^2)$ 에 대한

반복 학습의 결과

- (a) 반복횟수에 따른 막대의 각도,
- (b) 반복횟수에 따른 성능지수,
- (c) 150회째 시간에 따른 제어입력

Fig. 10 Learning results with initialization of  $w_i=0$  and  $J_k = \frac{1}{2} (e_k^2 + \dot{e}_k^2)$

$$= \frac{1}{2} (e_k^2 + \dot{e}_k^2)$$

- (a) Pole angle at each iterations,
- (b) Value of performance index versus iterations,
- (c) Control input versus time at 150th iteration

과정에서 이 부분의 상태에 해당된 적이 거의 없어 학습이 이루어지지 않은 부분을 나타낸 것이다.

한편, 그림 8은 바람직한 제어 방향을 알아내는데

있어 기존의 방법에서와 같이  $\frac{\partial e_{k+1}}{\partial u_k} \approx \frac{e_{k+1} - e_k}{u_k - u_{k-1}}$ 로

한 경우(점선)와 식 (9)와 같이 제안한 예측 방법을 적용한 경우(실선)에 대해 각 반복 학습시 잘못 알아내는 방향의 횟수를 퍼센트로 나타낸 것이다. 이로부터, 본 논문에서 제안한 방법이 바람직한 제어 방향을 매우 정확히 알아낼 수 있음 검증할 수 있었으며, 더우기 이 값을 강화 학습시킴으로써 약간의 오판이 발생하는 경우도 학습이 잘 이루어짐을 알 수 있다.

반면, 앞서의 모의실험과 같은 파라미터, 같은 초기화를 하고 단지 성능지수를 오차뿐만 아니라 오차의 변화율의 크기도 고려한 식 (15), (16)을 사용한 경우는 그림 9(a)와 같이 만족할만한 결과가 얻어졌다. 이는 오차의 변화율의 크기 또한 최소화하므로써 제어 입력이 과도하게 한 방향으로 치우치는 것이 조절되었기 때문으로 여겨진다. 이는 그림 9 (d), (e)를 그림 7 (c), (d)와 비교하므로써 알 수 있는 것이다.

또한, 1번째와 50번째 반복시의 결과에서는 오류가 발생하였으나, 횟수를 더할 수록 만족할만한 성능을 갖도록 퍼지 제어기가 자기구성됨을 알 수 있다. 55번째 반복 횟수 이후에는 한 주기 동안에 오류 발생이 없이 제어되는 경우와 오류가 발생하는 경우가 반복적으로 발생하다가 126번째 이후에는 오류의 발생없이 계속적으로 학습 제어가 이루어진다. 그림 9(b)는 오류가 발생되지 않는 반복 제어시의 성능지수 합( $J = \frac{1}{2} \sum_{k=1}^T (e_k^2 + \dot{e}_k^2)$ )을 나타낸 것으로 전체적으로 계속 감소하는 변화를 갖는 것을 볼 수 있다. 그림 9(b)에서, 성능지수 값이 약간의 증감을 반복하면서 줄어드는데 이것은 반복학습시 이전까지 해당되지 않았던 새로운 상태영역에 대한 학습이나  $J_k$ 를 줄이기 위한  $u_{k-1}$ 가  $J_{k+1}$ 을 증가시킬 수도 있기 때문에 나타나는 것이다.

상기한 모의 실험의 결과는 물론 초기화 방법에 따라 달라지게 되는데, 참고로 각 규칙 후건부의 단일 값( $w_i$ )을 모두 0으로 초기화하고 그 외의 파라미터는 같을 때, 식 (15)의 성능지수를 사용한 경우의 결과는 그림 9의 결과보다 우수하게 그림 10에 나타낸 것과 같이 얻어진다.

#### IV. 결 론

본 논문에서는 특성을 잘 모르는 시스템의 제어를 위한 자기구성 퍼지 제어기의 설계를 위해, 반복적인 시도를 통해 바람직한 제어입력의 변화방향과 퍼지 규칙(후건부 소속함수의 위치)을 학습하는 새로운 방법을 제안했다. 특히, 제어입력 변화에 따른 플랜트 출력(또는 오차)의 변화율의 학습은 기존의 연구에서 부족하게 다루어진 것으로서, 제안된 방법은 개념적으로 강화학습을 이용한 제어기 설계 방법과 유사하다. 그러나, 강화학습 방법에 비해 간단한 구조를 가지며, 학습하는 내용이 제어입력의 변화 방향으로 그 물리적 의미를 쉽게 알 수 있으며, 설계가 용이한 장점들을 갖는다.

제안된 자기구성 제어기를 역진자 시스템 제어에 적용하여 모의 실험한 결과, 후건부 소속함수의 위치 및 제어입력 변화 방향에 대한 대표값들을 랜덤하게 초기화한 까다로운 조건에 대해서도 학습과 제어가 모두 잘 이루어짐을 보였으며, 성능지수가 오차와 오차의 변화율의 크기의 합으로 주어질때 그 결과가 더욱 우수함을 검증하였다.

앞으로, 제안된 방법을 보다 일반화시키기 위해서는 다양한 플랜트에 대한 검증과 상태영역의 분할을 미리 정하는 것이 아니라 학습과정에서 갱신하면서  $\frac{\partial e}{\partial u}$ 의 대표값을 정하는 방법 등에 관한 연구가 진행되어야 할 것으로 사려된다.

#### 참 고 문 헌

1. M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science, 1985.
2. E. Mamdani, "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant," *Inst. Electr. Eng.*, Vol. 121, pp. 1569-1588, 1974.
3. 김용태, 이연정, 이승하, 변중남, "학습/적용 능력을 갖는 퍼지 제어시스템들에 관한 고찰," *한국퍼지 및 지능시스템학회지*, 5권 3호, pp. 11-35, 1995.
4. C. Moore and C. Harris, "Indirect Adaptive Fuzzy Control," *Int. J. Control*, Vol. 56, pp. 441-468, 1992.
5. C. Lin and C. S. G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," *IEEE Trans. on Computers*, Vol. 40, No. 12, pp. 1320-1336, 1991.
6. L.-X. Wang and J. Mendel, "Generating Fuzzy Rules by Learning From Examples," *IEEE Trans. on Syst., Man, Cybern.*, Vol. 22, pp. 1414-1427, 1992.
7. T. J. Procyk and E. H. Mamdani, "A Linguistic Self-Organizing Process Controller," *Automatica*, Vol. 15, pp. 15-30, 1979.
8. A. G. Barto, R. C. Sutton, and C. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans. on Syst., Man, Cybern.*, Vol. 13, No. 5, pp. 834-846, 1983.
9. C. C. Lee, "Intelligent Control Based on Fuzzy Logic and Neural Net Theory," *Int'l Conf. on Fuzzy Logic and Neural Network*, pp. 759-764, July 1990.
10. H. R. Berenji and P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements," *IEEE Trans. on Neural Net.*, Vol. 3, No. 5, pp. 724-740, 1992.
11. B. Chung and J. Oh, "Autotuning Method of Membership Function in a Fuzzy Learning Controller," *J. of Intelligent and Fuzzy Systems*, Vol. 1, pp. 335-349, 1994.
12. T. Sudkamp and R. J. Hammell II, "Interpolation, Completion, and Learning Fuzzy Rules," *IEEE Trans. Syst., Man, Cybern.*, Vol. 24, No. 2, pp. 332-342, 1994.
13. S. Geva and J. Sitte, "A Cartpole Experiment Benchmark for Trainable Controllers," *IEEE Control System Magazine*, pp. 40-51, Oct. 1993.
14. T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. on Syst., Man, Cybern.*, Vol. 15, pp. 116-132, 1985.



이 언 정(Yun-Jung Lee) 정회원

1961년 9월 10일생

1984년 2월:한양대학교 전자공학과 졸(학사)

1986년 2월:한국과학기술원 전기 및 전자공학과 졸(석사)

1994년 2월:동대학원 졸업(박사)

1986년 3월~1989년 8월:한국기계연구원 자동제어실 연구원

1994년 3월~1995년 2월:한국과학기술원 정보전자연구소 연수연구원

1995년 3월~현재:경북대학교 전자전기공학부 전임 강사

※주관심분야: 퍼지제어, 학습제어 및 보행로봇 제어 등임



최 봉 열(Bong-Yeol Choi) 정회원

1961년 5월 20일생

1983년 2월:경북대학교 전자공학과 졸업

1985년 2월:한국과학기술원 전기 전자공학과 졸업(석사)

1993년 2월:동 대학원 졸업(박사)

1990년:경북대 전임강사

1993년~현재:경북대 조교수

※주관심분야:가변구조제어, 적응제어, 로봇틱스, 건설제어