

본 연구는 1997년도 단국대학교 대학연구비의 지원으로 수행되었음

큰 외란이 존재하는 CNC 이송 구동계를 위한 적응 퍼지논리 제어기

지 성 철*

Self-Organizing Fuzzy Logic Controller for CNC Feed Drive Systems with Large Disturbances

Sungchul Jee*

ABSTRACT

This paper introduces a new self-organizing fuzzy logic controller (SOFLC) for precision contour machining in the presence of large disturbances which adjusts both input and output membership functions simultaneously. The parameters of the proposed controller are self-tuned in real-time according to a continuous measurement of the performance of the controller itself and estimated disturbance values. The proposed controller as well as a conventional fuzzy logic controller and a PID controller were simulated and implemented on a 3-axis milling machine in contour milling. Both the simulations and experiments show that the self-organizing fuzzy logic controller has superior performance in terms of contour tracking accuracy compared with the other two controllers.

Key Words : CNC(컴퓨터 수치제어), Machine Tools(공작기계), Feed Drive System(이송 구동계), Fuzzy Logic Control(퍼지논리 제어), Self-Organizing Mechanism(자기조정 기구)

1. 서 론

고정밀도의 윤곽가공을 실현하기 위해 정교한 컴퓨터 수치제어(CNC) 시스템을 개발하고자 하는 많은 노력들이 경주되어 왔다. 그러나, 공작기계 운동부의 마찰과 같은 외란으로 인해 커다란 가공오차가 발생할 수 있다. 마찰외란이 존재하는 경우의 윤곽추적 정밀도를 향상시키

기 위해서 모델에 기초하는 보정방법이 아닌, 규칙에 기초하는 퍼지논리 제어기를 이용한 공작기계의 서보제어가 제시된 바 있다⁽¹⁾. 이와 같이 규칙에 기초한 제어기가 제안된 이유는 마찰 특성들의 복잡성과 비선형성으로 인해 마찰모델에 기초한 보정 제어기를 구현하는 데 제한이 있기 때문이다.

본 논문은 정밀도를 요하는 윤곽가공에 있어 추적 성능

* 단국대학교 기계공학과

을 개선하기 위한 마찰 보정방법으로 퍼지논리 제어를 다루고 있는 일련의 두 논문 중 두번째 논문이다. 첫번째 논문⁽¹⁾에서는 고정된 파라미터를 갖는 퍼지논리 제어가 해석되었으며, 시뮬레이션과 실제의 밀링머신을 이용한 실험을 통해 윤곽 정확도 면에서 제어기의 성능이 입증되었다. 윤곽 정확도는 원하는 윤곽에 직교하는 방향의 기하학적 에러성분(즉, 원하는 윤곽으로부터 벗어난 양)으로 정의되는 윤곽오차(contour error)에 의해 결정된다. 퍼지논리 제어 자체가 제어되는 프로세스가 변화하는 것을 허용하는 적응능력을 갖추고 있지만, 고정된 제어기 파라미터와 제어규칙을 가진 퍼지논리 제어기로써는 변화하는 작동조건과 프로세스의 거동에 효과적으로 대처하기에 충분하지 못하다. 그 이유는 고정된 제어기 파라미터와 제어규칙은 제한된 범위의 시스템 입력과 작동조건에 대해서만 효과적일 수 있고, 따라서 다른 형태의 입력과 작동조건에 대해서는 제어기의 성능이 제한될 수 있기 때문이다. 그러므로, 제어를 변화하는 환경에 효과적으로 적응시키기 위해서는 제어기 파라미터와 제어규칙을 자동으로 조정해줄 필요가 있다. 이와 같은 자기조정 기구(self-organizing mechanism)를 갖추고 있는 퍼지논리 제어를 적응 또는 자기조정 퍼지논리 제어기(self-organizing fuzzy logic controller: SOFLC)라고 부른다. 자기조정 기구를 이용하면 퍼지논리 제어를 서로 다른 동작기계에 설치하는 데 필요한 복잡한 조율(tuning) 절차를 줄일 수 있고, 변화하는 가공조건에 효과적으로 대처할 수 있다.

1979년에 Procyk과 Mamdani⁽²⁾가 처음으로 적응 퍼지논리 제어를 도입한 이래 여러가지 형식의 적응 퍼지논리 제어가 제안되었다.⁽³⁾⁽⁴⁾⁽⁵⁾⁽⁶⁾⁽⁷⁾ 현재까지 제안된 적응 퍼지논리 제어에 쓰인 자기조정 방법들은 다음과 같이 크게 세 종류로 구분될 수 있다⁽³⁾.

- (a) 제어규칙들을 변경시키는 방법
- (b) 소속함수의 형상을 조정하는 방법
- (c) 소속함수를 이동하여 그 지지(support)를 변경시키는 방법

Procyk과 Mamdani⁽²⁾는 (a)의 방법으로 적응 퍼지논리 제어를 제안하였다. 이 방법은 제어기 입출력간의 상관행렬(relation matrix)로 표현되는 추론과정을 변경시킴으로써 제어기 구조를 근본적으로 바꿀 수 있는 장점이 있는 반면, 초기의 제어규칙들을 쉽게 복원할 수 없고 필요

로 하는 데이터 저장량이 많아 실시간 제어에 구현되기 어려운 단점이 있다. Batur와 Kasparian⁽⁸⁾, 그리고 Jee와 Koren⁽⁴⁾은 소속함수들의 기울기를 변화시켜 제어기 계인을 조정하는 (b)의 방법을 사용하였는데, 이들의 연구에서는 제어기 출력 소속함수들만이 조정되었다. Bare 등⁽⁵⁾은 제어기 입력에 대한 스케일링 계수(scaling factor)를 조정하여, Zhao 등⁽⁶⁾은 제어기 입출력 모두에 대한 스케일링 계수를 조정하여 제어를 조율하는 방법을 제안하였는데, 이들은 위의 방법 (c)에 속한다. (a)가 직접적인 제어구조의 조정방법이라면 (b)와 (c)는 소속함수들의 조정을 통한 간접적인 방법이라고 볼 수 있다. 그러나, (b)와 (c)는 (a)에 비해 계산 시간상의 부담이 작고 짧은 시간내에 제어기의 성능을 효과적으로 향상시킬 수 있는 장점이 있다. (b)의 방법은 (c)에 비해 제어규칙의 조정 효과가 나은 반면, 제어기 계인은 (c)의 방법을 통하여 더 용이하게 조정된다고 볼 수 있다. 특히, Langari와 Tomizuka⁽⁷⁾의 연구에서는 (b)와 (c)의 방법을 결합하였는데, 파라미터로 표시된 입력 소속함수들을 변경하는 적응 알고리즘을 제안하였다. 즉, 전 시간단계에서 작동된 제어규칙들을 평가하여 그에 해당하는 입력 소속함수들을 이동시키거나 확대·축소함으로써 입력 소속함수들을 조정하였다. 그러나, 입력 소속함수들만 조정하게 되면 제어규칙들을 제한된 정도로 밖에 변경시킬 수 없으므로 제어성능의 개선 또한 제한받게 된다.

따라서, 본 논문에서는 (b)와 (c)의 자기조정 방법을 결합하여 입력과 출력 소속함수들을 동시에 조정하는 새로운 적응 퍼지논리 제어를 제안한다. 제안된 적응 퍼지논리 제어기는 실시간 성능 측정에 따라 퍼지 소속함수(membership function)의 범위를 이동시키거나 확장 또는 축소시킴으로써 제어를 자동적으로 조율해 준다. 이와 같은 방법은 계산 시간상의 부담이 작고, 제어기 장착시에 제어기 파라미터들을 효율적으로 자기조정하며, 시변 프로세스에 파라미터들을 적응시킬 수 있다. 또한, 동작기계 안내면의 부착마찰에 의한 윤곽오차를 줄이기 위해 저속에서의 마찰보정 방법이 제안된 제어기에 포함되었다. 즉, 추정된 마찰모델⁽¹⁾을 이용하여 저속에서 출력 소속함수가 추가적으로 조정되었다.

2. 퍼지논리 제어

이절에서는 본 연구에서 제시된 적응 퍼지논리 제어기에 포함되는 일반적인 퍼지논리 제어기의 구성요소와 구

조에 대해 간략하게 기술한다.

• **소속함수**: 퍼지집합은 소속함수에 의해 특성화되는데, 함수값은 0과 1 사이의 값으로 그 퍼지집합에 속하는 정도를 나타낸다. 일반적으로 원하는 프로세스의 상태와 실제 프로세스의 상태의 차이값인 에러와 에러의 변화량이 퍼지논리 제어기의 입력으로 쓰인다. 본 논문에서 쓰인 에러 e 에 대한 소속함수들은 퍼지집합 A_i 를 표시함에 있어 각각 좌단점(A_i^l), 중앙점(A_i^c), 우단점(A_i^r)을 함수의 파라미터로 한다(Fig. 1a 참조). Fig. 1에 그 형태를 나타내었으며, 수학적으로 다음과 같이 표현될 수 있다.

$$\mu_{A_i}(e) = \begin{cases} 0 & \text{for } e < A_i^l \\ \frac{e - A_i^l}{A_i^c - A_i^l} & \text{for } A_i^l \leq e < A_i^c \\ \frac{A_i^r - e}{A_i^r - A_i^c} & \text{for } A_i^c \leq e < A_i^r \\ 0 & \text{for } e \geq A_i^r \end{cases} \quad (1)$$

같은 방법으로 에러 변화량 Δe 에 대한 퍼지집합 B_j 의 소속함수들도 파라미터 B_j^l, B_j^c, B_j^r 로써 정의될 수 있다.

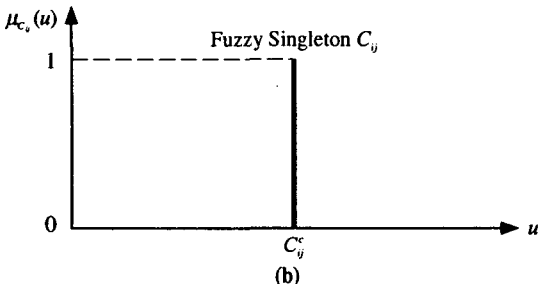
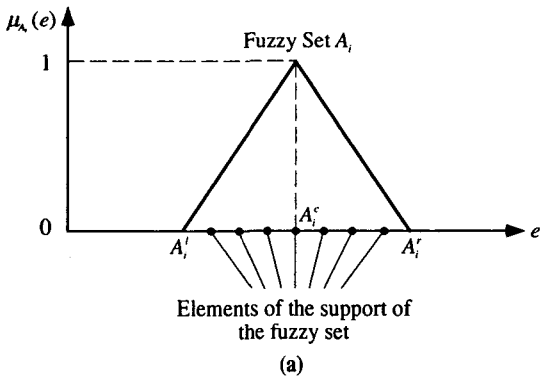


Fig. 1 Parameters describing a membership function: (a) typical; (b) singleton.

퍼지집합의 지지(support)는 전체집합(universe of discourse), 즉 퍼지집합이 정의되는 공간에서 그 퍼지집합에 대한 소속도(degree of membership)가 영이 아닌 모든 점들의 집합으로 정의되는데, 특별히 퍼지집합의 지지가 한점인 경우에 그 퍼지집합을 퍼지 싱글톤(singleton)이라 부른다. 출력 퍼지집합 C_{ij} 에 대해서는 그에 해당하는 소속함수들의 도심(centroid)들만 고려하는 단순화된 비퍼지화(defuzzification) 방법을 적용하였다. 다시 말해 퍼지 출력에 대해서는 퍼지 싱글톤들을 이용하였으며, 그들은 다음과 같이 나타내어진다.

$$\mu_{C_{ij}}(u) = \begin{cases} 1 & \text{if } u = C_{ij}^c \\ 0 & \text{if } u \neq C_{ij}^c \end{cases} \quad (2)$$

여기서 u 는 이산화된 제어명령값이고, C_{ij}^c 는 출력 소속함수의 도심을 나타낸다.

• **제어규칙**: 퍼지논리 제어기의 작동은 제어규칙 베이스에 들어있는 퍼지 규칙들을 토대로 하여 이루어진다. 이 규칙들은 위치에러, 에러의 변화량과 제어명령에 대해 각각 정의된 퍼지집합들(A_i, B_j, C_{ij})에 부여된 언어값들을 이용하며, 다음의 형태를 갖는다.

$$\text{If } e \in A_i \text{ and } \Delta e \in B_j \text{ then } u = C_{ij} \quad (3)$$

입력 소속함수의 좌단점과 우단점 사이의 거리 $|A_i^l - A_i^r|$ 또는 $|B_j^l - B_j^r|$ 이 크거나 작을수록, 그 소속함수에 해당하는 퍼지집합을 포함하는 제어규칙이 작용하게 될 확률이 각각 커지거나 작아진다. 그러므로, 좌단점과 우단점의 값들을 변화시킴으로써 제어규칙의 작용효과를 변화시킬 수 있다. 본 논문에서 제시된 제어기에서는 입력(A_i, B_j)과 출력(C_{ij})에 대해 각각 언어값 NL(Negative Large), NM(Negative Medium), NS(Negative Small), ZR(Zero), PS(Positive Small), PM(Positive Medium), PL(Positive Large)이 부여된 7개의 퍼지집합을 정의하였다. 위치에러 e 에 대해 정의된 소속함수와 파라미터들을 Fig. 2에 나타냈으며, 위치에러의 변화량 Δe 에 대해서도 마찬가지로 정의하였다. 제시된 제어기에서 2개의 제어 입력(e 와 Δe)이 쓰이고 각각의 입력에 대해 7개의 퍼지집합이 정의되므로, 모두 49개의 제어규칙이 결정되어 제어규칙 베이스에 저장되었다. 제시된 제어기에 쓰인 규칙들을 Table 1에 나타내었다.

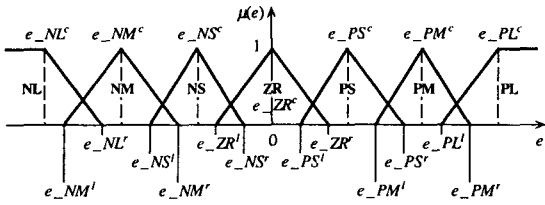


Fig. 2 Parameters of the membership functions for the position error.

Table 1 The fuzzy control rule base inside the proposed controller.

Control	Actions	If $\Delta e \in$						
		NL	NM	NS	ZR	PS	PM	PL
if $e \in$	NL	NL	NL	NL	NL	NL	NL	NL
	NM	NL	NL	NM	NM	NS	NS	NS
	NS	NL	NM	NM	NS	NS	NS	ZR
	ZR	ZR	ZR	ZR	ZR	ZR	ZR	ZR
	PS	ZR	PS	PS	PS	PM	PM	PL
	PM	PS	PS	PS	PM	PM	PL	PL
	PL	PL	PL	PL	PL	PL	PL	PL

• 퍼지논리 제어의 구조: 일반적인 퍼지논리 제어기의 블록선도를 Fig. 3의 상단에 나타내었다. 퍼지논리 제어기는 퍼지화 요소, 제어규칙을 포함한 추론기구, 비퍼지화 요소로 이루어진다. 전술한 바와 같이 본 연구에서는 현재 시간단계에서의 위치에러 $e(k)$ 와 현재와 전 시간단계 사이의 위치에러의 변화량 $\Delta e(k)$ 를 제어기 입력으로 하였다. 퍼지화 과정을 통해 제어기 입력이 퍼지 변수(A_i 와 B_j)로 변환되는데, 각각의 퍼지 변수에 대해 “양의 값으로 작다”(PS), “양의 값으로 크다”(PL), “음의 값으로 중간 정도이다”(NM)와 같은 명칭이 붙여진다. 퍼지화 과정을 거쳐 변환된 퍼지 입력은 규칙 베이스를 갖춘 추론기구를 통해 퍼지 출력(C_{ij})으로 전환된다. 또한, 추론기구는 개별적으로 작용한 규칙들로부터의 제어 출력들(C_{ij})을 결합하여 포괄적인 퍼지 출력(C)을 생성하게 된다. 비퍼지화 요소는 퍼지 변수 C 를 받아서 하나의 제어 출력 u_0 를 발생시킨다.

3. 제시된 적응 퍼지논리 제어

일반적으로 퍼지논리 제어기의 성능은 미리 정의된 퍼지집합 또는 소속함수와 제어규칙들에 좌우된다. 따라서, 퍼지집합이나 제어규칙들이 적절하게 정의되지 않거나 제어대상인 프로세스의 거동이 변하게 되면 그 제어기는 효과적이지 못하게 된다. 그러므로, 이와 같은 제어기를 효과적으로 만들기 위해서는 소속함수나 제어규칙들을

변경하고 재조율해야만 한다.

본 논문에서 새롭게 제시되는 자기조정 방법은 입력 소속함수들뿐만 아니라(즉, A_i' , A_i'' , B_j' , B_j'')을 조정) 출력 소속함수들(즉, C_{ij}^c)을 함께 조정한다. 제어명령에 직접적으로 영향을 주는 출력 소속함수들도 함께 조정하게 되면, 제어규칙들을 보다 포괄적으로 변경시킬 수 있고, 따라서 제어성능을 크게 향상시킬 수 있게 된다. 이와 같은 입력출력 소속함수들의 조정은 뒤에 설명될 (i)성능지수와 (ii)각기 작용된 제어명령과 포괄적인 명령(u_0)과의 비교에 의해 자동적으로 이루어진다. 성능지수(performance index)는 전 시간단계에서의 제어명령이 에러 감소에 효과적이었는지를 측정하는 역할을 하고, 제어명령을 비교함으로써 전 시간단계에서 각기 작용한 제어규칙이 포괄적인 제어명령의 결정에 어떻게 기여했는지를 평가하게 된다. 제안된 자기조정 방법에서 실시간으로 성능지수를 계산하고 작용한 제어규칙을 평가해서 입력출력 소속함수들을 조정하기 때문에, 퍼지논리 제어기를 변화하는 작동조건과 시변 프로세스에 효율적으로 적용시킬 수 있다. 또한, 제어규칙들을 직접 변경하는 방법에 비해 계산상의 부하가 작으며, 적은 양의 데이터 저장으로 구현될 수 있는 장점이 있다.

본 논문에서 제시된 자기조정 기구의 구조를 Fig. 3의 하단에 나타내었으며, 아래에 상술하였다.

• 제어규칙의 성능 평가: 전술한 바와 같이 제안된 제어기는 제어규칙들을 자동으로 평가함으로써 제어성능을 개선시키게 된다. 전 시간단계 $k-1$ 에서 작용한 제어규칙을 평가하기 위해서 시간단계 $k-1$ 에서의 에러 $e(k-1)$ 과 현 시간단계 k 에서의 에러 $e(k)$ 가 쓰인다.

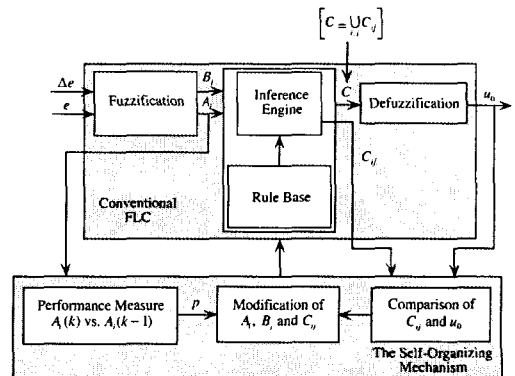


Fig. 3 Block diagram of the proposed SOFLC.

자기조정 기구는 각기 작용된 제어규칙이 전 시간단계와 현 시간단계 사이에 어느 정도의 빠르기로 에러를 변화시켰는가를 결정하게 된다. 시간단계 $k-1$ 에서 작용한 각각의 제어규칙을 현 시간단계 k 에서 평가한 성능지수 $p(k)$ 의 값은 $\{-2, -1, 0, 1, 2\}$ 중의 하나로 결정되는데, 작용한 규칙에 의한 에러 감소율을 나타낸다. 성능지수 2는 가장 빠른 성능을, -2는 가장 느린 성능 또는 에러가 줄어들지 않고 오히려 증가함을 나타낸다. 성능지수 0은 에러가 적당한 속도로 감소하고 있거나 원하는 에러 범위안으로 감소해서 다음 단계에서 변경이 필요 없는 경우를 나타낸다. 예를 들어 $e(k-1)$ 이 PS인 경우에 대해, (i) $e(k)$ 가 PL이면 $p(k)$ 는 -2로 너무 느린 성능, 다시 말해 에러가 증가함을, (ii) $e(k)$ 가 PS이면 $p(k)$ 는 -1로 에러가 감소하지 않았음을, (iii) $e(k)$ 가 NS이면 $p(k)$ 는 0으로 에러가 적당히 감소하고 있는 적절한 성능을, (iv) $e(k)$ 가 NL이면 $p(k)$ 는 1로 에러가 원하는 양 이상으로 감소된 조금 빠른 성능을 의미한다. 다른 예로, $e(k-1)$ 이 PL인 경우에 대해, $e(k)$ 가 NL이면 $p(k)$ 는 2로 정의되며 에러 감소가 과대하여 원하는 위치의 반대 방향으로 너무 멀리 벗어나게 됨을 의미한다.

본 논문에서 제시된 방법의 새로운 면의 하나는 성능지수의 결정에 대한 원리에 있다. 이 원리는 다른 제어규칙 베이스와 퍼지집합을 갖는 유사 시스템에 대해서도 적용될 수 있다. 편의상 E_l ($l = -3, -2, -1, 0, 1, 2, 3$)을 위치 에러에 대한 소속함수로 정의했는데, 여기서 첨자 -3은 앞에서 정의한 NL을, -2는 NM을, -1은 NS를, 0은 ZR을, 1은 PS를, 2는 PM을, 3은 PL을 각각 나타낸다. 또한 E_{-3}, E_{-1}, E_1, E_3 을 기소속함수(odd membership function)로, E_{-2}, E_0, E_2 를 우소속함수(even membership function)로 정의하였다. 소속함수의 좌단점과 우단점에 대해, 이웃하는 소속함수의 중심을 넘어서지 않도록 제한하는 구속조건을 두었는데, 이렇게 하면 기소속함수들과 우소속함수들은 서로 겹쳐질 수 없게 된다. 성능지수를 결정하는 원리에 대해 아래에 설명하였다.

먼저, 기소속함수들 사이의 또는 우소속함수들 사이의 성능지수를 정의하였다. 어떤 위치어려값에 해당되는 소속함수가 연속하는 두 시간단계 사이에 E_{2m} 에서 E_{2n} 으로(즉, 우소속함수에서 우소속함수로) ($m, n = -1, 0, 1$), 또는 E_{2m-1} 에서 E_{2n-1} 로(즉, 기소속함수에서 기소속함수로) ($m, n = -1, 0, 1, 2$) 변화했다고 가정하면, 성능지수 $p(k)$ 는 다음과 같이 나타내어진다.

$$p(k) = \begin{cases} \frac{1}{2}[x - y - 2] & \text{for } x > 0 \\ \frac{1}{2}[y - x - 2] & \text{for } x < 0 \end{cases} \quad (4)$$

여기서, (x, y) 는 $(2m, 2n)$ 또는 $(2m-1, 2n-1)$ 을 나타낸다. 한편, 소속함수가 기소속함수에서 우소속함수로 또는 그 반대로 변화된 경우에 대해서는 성능지수를 아래와 같이 정의하였다.

$$p(k) = \begin{cases} \frac{1}{2}[x - y - 1] & \text{for } x > 0 \\ \frac{1}{2}[y - x - 1] & \text{for } x < 0 \end{cases} \quad (5)$$

여기서, (x, y) 는 $(2m-1, 2n)$ 또는 $(2m, 2n-1)$ 이다. 위의 원리에 대한 예외로 에러가 E_{-3} 또는 E_3 에서 E_0 으로 되었을 경우에는, 에러가 원하는 작은 범위로 감소했기 때문에 성능지수를 1대신 0으로 정의하였다. 본 연구에서 정의된 성능지수를 Table 2에 나타내었다. 특히, Table 2에서 *로 표시된 경우들은 전 시간단계에서의 에러 $e(k-1)$ 이 E_0 이었으나 현재의 시간단계에서는 에러가 E_0 밖으로 증가한 경우에 해당한다. 따라서, 이 경우들에 대해서는 에러의 감소율로 정의되는 성능지수가 무의미하게 되므로, 성능지수를 정의할 필요없이 에러를 증가시키는 작용을 한 규칙들에 관계되는 소속함수들을 조정하여 다음 시간단계에서 전체 제어명령에 대한 그 규칙들의 영향을 줄일 수 있도록 하였다. 반면에 $e(k-1)$ 이 E_0 이고 $e(k)$ 가 E_0 인 경우에 대해서는 에러가 여전히 작은 범위안에 있으므로 성능지수를 0으로 정의하였다.

Table 2 The performance indices.

Performance	Index	If $e(k)$ is						
		E_{-3}	E_{-2}	E_{-1}	E_0	E_1	E_2	E_3
If $e(k-1)$ was	E_{-3}	-1	0	0	0	1	2	2
	E_{-2}	-1	-1	0	0	1	1	2
	E_{-1}	-2	-1	-1	0	0	1	1
	E_0	*	*	*	0	*	*	*
	E_1	1	1	0	0	-1	-1	-2
	E_2	2	1	1	0	0	-1	-1
	E_3	2	2	1	0	0	0	-1

• **소속함수의 조정:** 퍼지 입력 소속함수의 변경은 소속함수들의 좌단점 (A'_l, B'_l)과 우단점 (A'_r, B'_r)을 동시에

움직임에 따라 그 소속함수에 의해 둘러싸이는 면적을 좁히거나 넓힘으로써 이루어진다. 퍼지 출력 소속함수에 대해서는 도심(C_{ij}^c)을 이동시킴으로써 소속함수들을 변경하였다. 이와 같은 소속함수들의 조정 원리를 다음의 두 예제를 통해 설명하고자 한다.

• **예제1:** 전 시간단계 $k-1$ 에서 편의상 오직 하나의 제어 규칙이 작용하였고, 그 규칙은 다음과 같았다고 가정한다.

$$\text{If } e(k-1) = \text{PS and } \Delta e(k-1) = \text{PS then } u(k-1) = \text{PM} \quad (6)$$

위의 제어 작용에 의해 Fig. 4에 나타난 바와 같이 예러가 $e(k-1) = \text{PS}$ 에서 두 소속함수 ZR과 PS에 속하는 $e(k)$ 로 변화된 경우를 생각한다. 이때에 Eq. (6)의 규칙은 두 성능지수를 갖는데, 예러가 $e(k-1) = \text{PS}$ 에서 $e(k) = \text{ZR}$ 로 변했기 때문에 성능지수가 0인 동시에, 한편으로 예러가 여전히 $e(k) = \text{PS}$ 에 속해 있기 때문에 성능지수가 -1이 될 수 있다(Table 2 참조). 그러나, 제어 규칙의 작용효과에 대한 결정을 내리기 위해서는 제어 규칙에 대해 오직 하나의 성능지수를 부여해야 한다. $e(k) = \text{ZR}$ 이 $e(k) = \text{PS}$ 에 비해 소속도가 높기 때문에, 즉 소속함수값이 1에 더 가깝기 때문에, 자기조정 기구는 이 규칙을 대표하는 성능지수가 0이라고 결정하게 된다. 따라서, 다음 시간단계에서 이 규칙에 관련된 파라미터들을 변경할 필요가 없다.

• **예제2:** 두번째 예제에서는 예러 $e(k-1)$ 이 두 소속함수에 해당된 경우에 대해 생각한다. 시간 $k-1$ 에서 다음의 제어규칙들이 작용했다고 가정하자.

$$\begin{aligned} (a) \text{ If } e(k-1) = \text{PS and } \Delta e(k-1) = \text{ZR then } u(k-1) = \text{PS} \\ (b) \text{ If } e(k-1) = \text{PM and } \Delta e(k-1) = \text{ZR then } u(k-1) = \text{PM} \end{aligned} \quad (7)$$

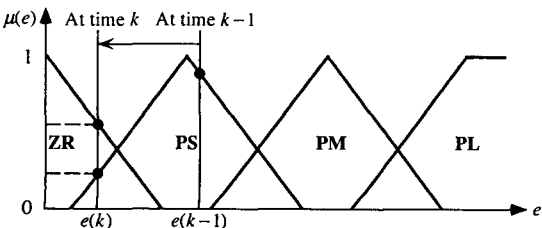


Fig. 4 Error change in Example 1.

$e(k-1)$ 에 해당하는 소속함수값들이 Fig. 5a에 나타난 바와 같다고 가정한다. 시간 $k-1$ 에서 제어를 통해 각각의 제어규칙에 대한 출력 소속도가 계산되어졌고 (Fig. 5b 참조), 이를 바탕으로 2개의 출력들은 하나의 포괄적인 제어명령(u_0)으로 결합되었다. 이 포괄적 제어 명령이 작용한 결과, 시간 k 에서의 예러($e(k)$)가 Fig. 5a와 같이 증가되었다고 하자. 예제 1과 같은 방법으로 작용한 각 규칙에 대해 성능지수들이 주어지는데, Eq. (7)의 규칙 (a)의 경우 $e(k)$ 가 PM과 PL에 속하기 때문에 성능지수는 -1(PS로부터 PM으로 변함) 또는 -2(PS로부터 PL로 변함)이다. 여기서, PL이 PM보다 소속도가 높기 때문에(Fig. 5a 참조) 규칙 (a)에 대한 성능지수는 -2로 결정된다. Eq. (7)의 규칙 (b)에 대해서는 $e(k) = \text{PM}$ 과 $e(k) = \text{PL}$ 모두 성능지수가 -1로서 동일하다(Table 2 참조). 결국 두 제어규칙 모두 성능지수 (-1과 -2)가 만족스럽지 못하기 때문에 이 제어규칙들에 관련된 파라미터들은 변경될 필요가 있다.

Fig. 5b에서 보는 바와 같이 규칙 (a)의 제어명령으로 인해 포괄적 제어명령 $u_0(k-1)$ 이 왼쪽으로 당겨져서 작아졌기 때문에 실질적으로 포괄적 제어명령이 나빠졌음을 알 수 있다. 즉, 예러를 줄이기 위해서는 $u_0(k-1)$ 이 더 큰 값일 필요가 있었다. 제어규칙 (a)가 포괄적인 제어명령을 작게 하는 부정적인 효과를 주기 때문에, 이 제어규칙에 관련된 입력 퍼지집합의 범위를 좁혀서 제어 규칙을 약화시키게 된다. 따라서, $e(k-1) = \text{PS}$ 와 $\Delta e(k-1) = \text{ZR}$ 에 해당하는 소속함수들의 면적을 좁힌다. Fig. 5b에서 알 수 있듯이 규칙 (b)로부터의 제어명령은 제어규칙 (a)의 명령보다 크므로, 포괄적 제어명령보다 큰 값이 된다. 따라서, 규칙 (b)에 관계되는 입력 소속함수들($e(k-1) = \text{PM}$ 과 $\Delta e(k-1) = \text{ZR}$)은, 그 제어명령인 PM이 더 높은 소속도를 가져서 포괄적인 제어명령의 결정에 더 큰 기여를 하고 결과적으로 u_0 를 크게 할 수 있도록 변경되어야 한다. 이와 같은 이유에서 두 입력 소속함수들은 넓혀지게 된다. 각 규칙의 퍼지 출력 싱글톤에 대해서는 함께 작용한 다른 퍼지 출력 싱글톤들에 비해 포괄적 제어명령에 부정적 효과를 준 것만을 변경하였다. 따라서, 규칙 (a)로부터의 퍼지 출력 싱글톤인 $u(k-1) = \text{PS}$ 가 더 큰 값을 갖도록 오른쪽으로 이동되고, 규칙 (b)의 출력 싱글톤 $u(k-1) = \text{PM}$ 은 이동되지 않는다. 이 예제를 정리하면, 성능지수를 결정하고 작용한 각각의 제어규칙으로부터의 제어명령과 포괄적 제어명령을 비교함으로써 어떤 제어규칙이 포괄적 제어명령

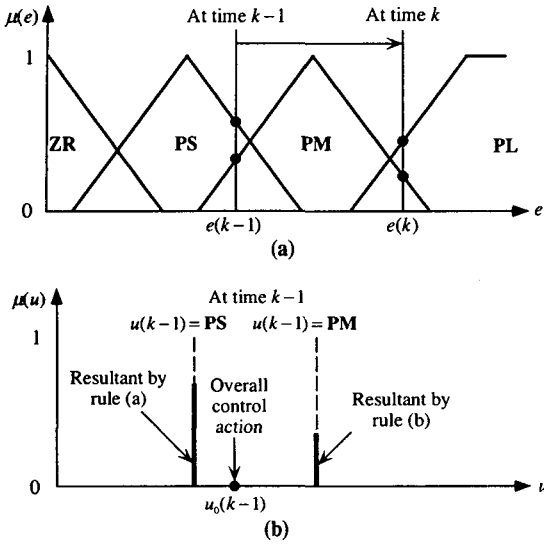


Fig. 5 Membership functions in Example 2:
(a) error change; (b) fuzzy outputs.

에 긍정적 또는 부정적인 영향을 미쳤는가를 결정할 수 있다는 것이다.

일반적으로 파라미터들의 변경 정도, 즉 입력 소속함수들에 대한 좌단점과 우단점의 이동 크기와 퍼지 출력 싱글톤의 이동 크기는 성능지수값으로부터 결정될 수 있다. 다시 말해 제어규칙에 대한 성능지수가 ±1이면 그에 해당하는 입력 소속함수들의 좌단점과 우단점은 1단위 만큼 이동되어지고, 퍼지 출력 싱글톤 역시 1단위 이동되어진다. 성능지수가 ±2이면 위의 파라미터들은 2단위씩 조정되고, 성능지수가 0인 경우에는 변경이 필요없다.

본 연구에서 제시된 적응 퍼지논리 제어기의 주요 알고리즘을 정리하면 다음과 같다. 여기서, K_A, K_B, K_C 는 각각 에러와 에러 변화량에 대한 소속함수, 그리고 퍼지 출력 싱글톤에 대한 조정 단위의 크기를 나타내는 상수이다.

전 시간단계 $k-1$ 에서 작용한 제어법칙이 “If $e(k-1) = A_i(k-1)$ and $\Delta e(k-1) = B_j(k-1)$ then $u(k-1) = C_{ij}(k-1)$ ”이고, 이 규칙에 대해 결정된 성능지수가 $p(k)$ 라고 하면,

- (I) if $e(k-1) > 0$ and $p(k) > 0$, then
if $C_{ij}(k-1) \geq u_0(k-1)$, then

$$A_i'(k) = A_i'(k-1) + K_A p(k); A_i'(k) = A_i'(k-1) - K_A p(k)$$

$$B_j'(k) = B_j'(k-1) + K_B p(k); B_j'(k) = B_j'(k-1) - K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1) - K_C p(k)$$

else

$$A_i'(k) = A_i'(k-1) - K_A p(k); A_i'(k) = A_i'(k-1) + K_A p(k)$$

$$B_j'(k) = B_j'(k-1) - K_B p(k); B_j'(k) = B_j'(k-1) + K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1)$$

- (II) if $e(k-1) > 0$ and $p(k) < 0$, then

if $C_{ij}(k-1) > u_0(k-1)$, then

$$A_i'(k) = A_i'(k-1) + K_A p(k); A_i'(k) = A_i'(k-1) - K_A p(k)$$

$$B_j'(k) = B_j'(k-1) + K_B p(k); B_j'(k) = B_j'(k-1) - K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1)$$

else

$$A_i'(k) = A_i'(k-1) - K_A p(k); A_i'(k) = A_i'(k-1) + K_A p(k)$$

$$B_j'(k) = B_j'(k-1) - K_B p(k); B_j'(k) = B_j'(k-1) + K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1) - K_C p(k)$$

- (III) if $e(k-1) < 0$ and $p(k) > 0$, then

if $C_{ij}(k-1) > u_0(k-1)$, then

$$A_i'(k) = A_i'(k-1) - K_A p(k); A_i'(k) = A_i'(k-1) + K_A p(k)$$

$$B_j'(k) = B_j'(k-1) - K_B p(k); B_j'(k) = B_j'(k-1) + K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1)$$

else

$$A_i'(k) = A_i'(k-1) + K_A p(k); A_i'(k) = A_i'(k-1) - K_A p(k)$$

$$B_j'(k) = B_j'(k-1) + K_B p(k); B_j'(k) = B_j'(k-1) - K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1) + K_C p(k)$$

- (IV) if $e(k-1) < 0$ and $p(k) < 0$, then

if $C_{ij}(k-1) \geq u_0(k-1)$, then

$$A_i'(k) = A_i'(k-1) - K_A p(k); A_i'(k) = A_i'(k-1) + K_A p(k)$$

$$B_j'(k) = B_j'(k-1) - K_B p(k); B_j'(k) = B_j'(k-1) + K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1) + K_C p(k)$$

else

$$A_i'(k) = A_i'(k-1) + K_A p(k); A_i'(k) = A_i'(k-1) - K_A p(k)$$

$$B_j'(k) = B_j'(k-1) + K_B p(k); B_j'(k) = B_j'(k-1) - K_B p(k)$$

$$C_{ij}'(k) = C_{ij}'(k-1)$$

• **저속 마찰보정:** 본 연구의 자기조정 알고리즘에 추정된 마찰모델을⁽¹⁾ 부분적으로 포함시켰다. 마찰에 의한 윤곽 오차를 줄이기 위해, 정적마찰(static friction)과 역점성 마찰(negative viscous friction)이 존재하여 마찰의란

값이 크게 되는 저속영역(본 연구에 쓰인 기계에 대해 12 mm/sec 이하)을 정의하였다. 이 저속영역에 대해 위의 마찰모델에서 쿨롬마찰을 제외한 나머지 부분을 정적마찰과 역점성마찰에 의한 효과로 간주하여 이를 속도(V : mm/sec)의 함수로 다음과 같이 정의하였으며,

For the X - axis :

$$u_d(V) = \begin{cases} 0.05V^2 - 1.16V + 6.54 & \text{for } 0 \leq V < 12 \\ -0.04V^2 - 1.00V - 6.43 & \text{for } -12 \leq V < 0 \end{cases} \quad (8)$$

For the Y - axis :

$$u_d(V) = \begin{cases} 0.04V^2 - 1.12V + 7.23 & \text{for } 0 \leq V < 12 \\ -0.02V^2 - 0.68V - 5.04 & \text{for } -12 \leq V < 0 \end{cases}$$

속도 피드백에 따라 출력 소속함수들을 조정하였다. 즉, 전술한 자기조정 기구에 의해 조정된 퍼지 출력 싱글톤들 ($C_{ij}^c(k)$)에 u_d 가 더해져서 싱글톤들의 값은 다음과 같이 조정되며,

$$C_{ij}^c(k) = C_{ij}^c(k) + u_d(V) \quad (9)$$

이는 결과적으로 적응 퍼지논리 제어기의 포괄적 제어명령값 u_0 에 u_d 를 더한 것과 같다.

또한, 위의 마찰보정방법의 연장으로서, 속도가 0에 가까운 경우에 원활한 운동을 위한 부차마찰의 보정방법으로 다음의 조건을 추가하였다. 즉, 시스템이 실제의 피드백을 가져서 제어명령을 내보내고 있는 상태에서만 부차마찰을 보정할 수 있도록 마찰보정 알고리즘에 아래의 조건을 추가하였다.

$$\begin{aligned} \text{If } (V \approx 0 \text{ and } |e(k)| \geq 2 \text{ BLUs}) \text{ then } u_d = u_d \\ \text{If } (V \approx 0 \text{ and } |e(k)| < 2 \text{ BLUs}) \text{ then } u_d = 0 \end{aligned} \quad (10)$$

여기서, BLU(basic length-unit)는 시스템의 해상도(resolution)(본 논문에 쓰인 시스템의 경우 0.01 mm)에 해당되므로, 예러가 $-1 \text{ BLU} \leq e \leq 1 \text{ BLU}$ 의 범위에 있게 되면 시스템은 개루프(open loop) 상태에 놓이게 된다. 예러의 절대값이 1 BLU를 초과할 때만 시스템은 피드백을 갖게 되고, 운동 제어명령을 발생시킬 수 있다. 따라서, $e \geq 2 \text{ BLU}$ 인 예러 신호에 대해서만 제어 시스템이 운동 명령을 내보내고 있다고 볼 수 있다.

4. 적응 퍼지논리 제어의 평가

적절하게 조율된 일반적인 퍼지논리 제어기의 성능이 PID 제어기의 성능보다 우수하다는 것을 앞의 논문에서¹¹⁾ 보인 바 있다. 이절의 목적은 일반적인 퍼지논리 제어기(FLC)와 적응 퍼지논리 제어기(SOFLC)의 성능을 비교하고, 적응 퍼지논리 제어기의 성능에 있어 자기조정 효과의 효과를 연구하는 데 있다. 이를 위해 퍼지논리 제어기와 제시된 적응 퍼지논리 제어기를 이용하여 컴퓨터 시뮬레이션과 3마력 CNC 밀링머신을 통한 실제의 윤곽추적 실험을 행하였다. 이 밀링머신은 범용 PC에 의해 제어되므로 다양한 인터폴레이션(interpolation) 및 제어 알고리즘을 용이하게 실행시킬 수 있는 장점을 갖는다. 제어 컴퓨터(PC)는 디코더 보드(decoder board)와 디지털 I/O 보드를 통해 각각 선형 인코더(linear encoder)와 digital-to-PWM(pulse width modulation) 변환기(converter)와 연결되어 있다. 선형 인코더는 밀링머신의 각 축에 장착되어 제어기에 테이블의 위치 피드백을 보내며, digital-to-PWM 변환기는 각 축의 제어기로부터 출력된 8 bit 제어명령값을 5 volt의 PWM 신호로 변환한다. 각 축에 대한 PWM 신호는 밀링머신 내부의 전력 증폭기(power amplifier)를 통해 증폭되어 각 축의 직류 서보 모터로 보내진다. Fig. 6은 제시된 적응 퍼지논리 제어기를 밀링머신의 한 축에 적용한 경우의 실험장치 구성도를 나타내는데, 두 축에 대해 적응 퍼지논리 제어기를 각각 구현하여 윤곽추적 시뮬레이션 및 실험을 하였다. Fig. 6에서 R_x 와 R_y 는 각각 X축과 Y축에 대한 기준 위치입력을, P 는 인코더(encoder)로부터의 위치 측정값을 나타낸다. V 는 속도 피드백으로, 연속하는 두 시간단계 사이의 위치 변화 ΔP 를 샘플주기(sampling period) T (본 연구에서 쓰인 시스템의 경우 10 msec)로 나눠서 계산된 값이다. 정상상태에서의 추적 정확도를 향상시키기 위해 적응 퍼지논리 제어 알고리즘에 적분(I) 제어명령을 결합하였으며, U_c 는 모터에 보내지게 될 조합된 제어명령을 나타낸다. 적응 퍼지논리 제어기, 퍼지논리 제어기, 그리고 일반적인 PID 제어기를 CNC 밀링머신에 구현하였다. 실험장치 구성요소들의 사양은 Table 3과 같다.

적응 퍼지논리 제어시스템에 대한 전체 제어 프로그램은 (i)주어진 파트 프로그램(part program)으로부터 각 축의 기준 위치입력을 계산하는 인터폴레이션 루틴(routine), (ii)자기조정 기구를 포함하는 적응 퍼지논리 제어 루틴 및 (iii)인코더로부터 위치 데이터를 얻고 적응 퍼지

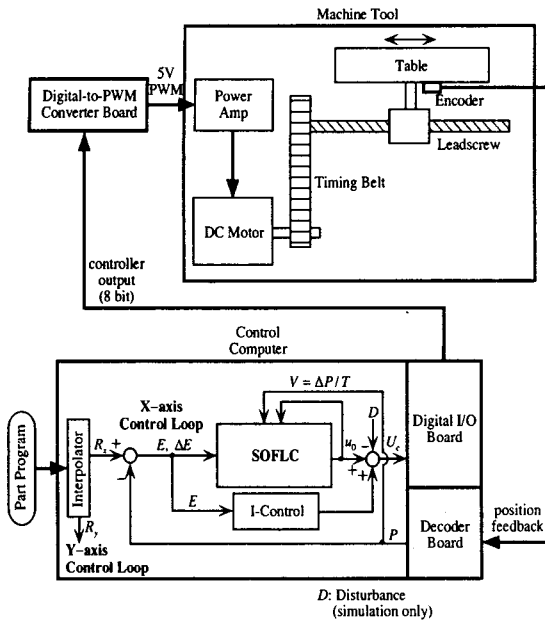


Fig. 6 Overall structure of the experimental testbed.

Table 3 Specifications of the experimental apparatus.

Components	Specifications
Milling Machine	3-hp Knee Type
Control Computer	33MHz 80486-based PC
Digital I/O Board	8 bit I/O (for each channel)
Decoder Board	Direction Sensor and Up-down Counter
Digital-to-PWM Converter	5V PWM Output
Linear Encoders	10 μm Resolution

논리 제어 루틴으로부터 계산된 제어명령을 내보내는 인터페이스 보드들을 제어하는 드라이버(driver) 루틴으로 구성되었다. 이들은 C언어로 쓰여졌으며, DOS 운영체제 하에서 작동되었다.

Fig. 6에 보인 바와 같이 본 논문에서 제시된 적응 퍼지논리 제어기에서 각 축에 대해 기준 위치입력과 밀링머신의 실제 위치의 차로 정의되는 축방향 위치에러(axial position error), 위치에러의 변화량, 속도 피드백과 퍼지 제어기 출력이 자기조정 알고리즘의 입력으로 쓰였다. 위치에러에 대한 퍼지 입력 소속함수들에 대해 그 경계인 좌단점과 우단점들을 기본 길이단위(BLU)의 0.5배를 최소단위로 하여 움직여서(즉, $K_A = 0.5$ 로 해서) 소속함수

들을 넓히거나 좁혔다. 위치에러의 변화량에 대해 정의한 소속함수들에 대해서는 0.125 BLU를 최소단위로 하여 조정하였고($K_B = 0.125$), 퍼지 출력 소속함수들에 대해서는 전력 증폭기에 보내지는 제어명령값(8 bit)의 기본 단위를 최소단위로 하여($K_C = 1$) 싱글톤들을 이동하였다. 또한, 실험에서 싱글톤들은 마찰력의 추정값⁽¹⁾에 따라서도 조정되었다.

자기조정 기구가 없는 일반적인 퍼지논리 제어기와 적응 퍼지논리 제어기 모두에 대해 제어기 입출력 소속함수들의 초기 파라미터값들을 같게 하였으며, 의도적으로 이 값들을 잘 조율되지 않은 값으로 택하였다. 입력 소속함수에 대해서는 시뮬레이션과 실험에서 동일한 초기 파라미터값들을 사용하였다.

4.1 시뮬레이션 해석

본 논문에서 제시된 자기조정 기구의 효과를 관찰하기 위해 적응 퍼지논리 제어기와 퍼지논리 제어기에 대한 시뮬레이션을 수행하였다. 시뮬레이션이 좀 더 실제적으로 되기 위하여 실제의 밀링머신에 대해 측정된 마찰값⁽¹⁾을 시스템의 외란으로 시뮬레이션에 포함시켰다. 시뮬레이션 프로그램 상에서 계산되어지는 밀링머신의 X축과 Y축 속도에 따라, 각 축에 대해 속도의 함수로 정의된 마찰 모델을 통해 마찰외란값들을 계산하고 이 값들을 각 축의 제어기 출력에서 빼는 방식으로 마찰외란을 반영하였다. 즉, 이송 구동부의 입력이 마찰외란의 크기 만큼 작게 되도록 하였다.

•자기조정의 효과: Fig. 7은 밀링머신이 반경 40 mm의 원을 반시계 방향으로 0.754 m/min의 이송속도를 가지고 움직이는 경우에 대한 시뮬레이션 결과를 나타낸다. 이 경우에 X축과 Y축에 대한 각각의 기준 위치입력은 다음과 같이 주어진다.

$$R_x(k) = 4000 \sin\left(\frac{\pi}{10} kT\right) \quad [\text{BLU}]$$

$$R_y(k) = 4000 - 4000 \cos\left(\frac{\pi}{10} kT\right) \quad [\text{BLU}] \quad (11)$$

이 시뮬레이션에서는 본 논문에서 제시된 자기조정 기구만의 효과를 관찰하기 위해 적분제어 계인을 0으로 하였다. 한번의 원운동(이를 1 사이클이라 정의하였음)에 20초가 소요되는데, 이 기간 동안 적응 퍼지논리 제어기

는 제어기 파라미터들을 자기조율하면서 자신을 주어진 환경에 적응시키게 된다. 적응하는 데 또 하나의 사이클이 필요하다면, 두번째 사이클에서는 첫번째 사이클에 비해 성능이 향상될 것이다. 자기조정 효과를 보이기 위해 연속 3 사이클의 원운동을 시뮬레이션하였다. 처음 3 사이클(즉, 0초부터 60초까지) 동안의 X축 방향의 위치 에러를 Fig. 7a에 나타내었고, Y축 방향에 대해서도 유사한 결과를 얻었다. 시작단계에서 적응 퍼지논리 제어기와 퍼지논리 제어기에 대해 제어기 파라미터값들을 같게 놓았다. 적응 퍼지논리 제어기의 결과(SOFLC로 표시)를 보면 사이클이 진행되면서 에러가 감소한 것을 알 수 있고, 실효값(root mean square: RMS) 에러가 첫째 사이클의 9.6 BLU에서 셋째 사이클(즉, 40초에서 60초)에는 5.6 BLU로 감소하였다. 반면에 일반적인 퍼지논리 제어기(FLC로 표시)에 대해서는 실효값 에러가 약 16.2 BLU로 남아있는 것을 알 수 있으며, 이는 적응 퍼지논리 제어기의 자기조정 효과를 보여준다.

윤곽오차를 비교해 보면 15초후에(첫째 사이클의 3분원을 지난 후에) 자기조정 기구가 윤곽오차를 효과적으로 줄인 것을 알 수 있다. 세 사이클에 대한 윤곽오차를 Fig. 7b에 비교하였는데, 두 제어기의 성능에 큰 차이가 있음을 볼 수 있다. 결론적으로 자기조정 기구에 의해 첫째 사이클 후에 실효값 윤곽오차가 퍼지논리 제어기에 비하여 1/4로 감소하게 된다. 자기조정 기구에 의해 자동 조율된 파라미터들을 가지고 시뮬레이션을 계속 실행시키면 윤곽오차는 셋째 사이클과 같은 값들로 남아 있게 된다.

X축과 26.6°의 각을 갖는 직선 궤적에 대해 0.6 m/min의 이송속도로 움직이는 경우의 시뮬레이션 결과를 Fig. 8에 나타내었다. 이 시뮬레이션에서 정상상태의 위치에러를 없애기 위해 적분제어 게인을 8.0으로 놓았다. 퍼지논리 제어기에 대해서는 위치에러가 수렴하지 않았지만 적응 퍼지논리 제어기의 경우에는 정상상태 에러가 0으로 수렴하였다. 윤곽오차를 살펴보면 두 제어기의 차이가 더욱 확실해진다. 퍼지논리 제어기의 윤곽오차는

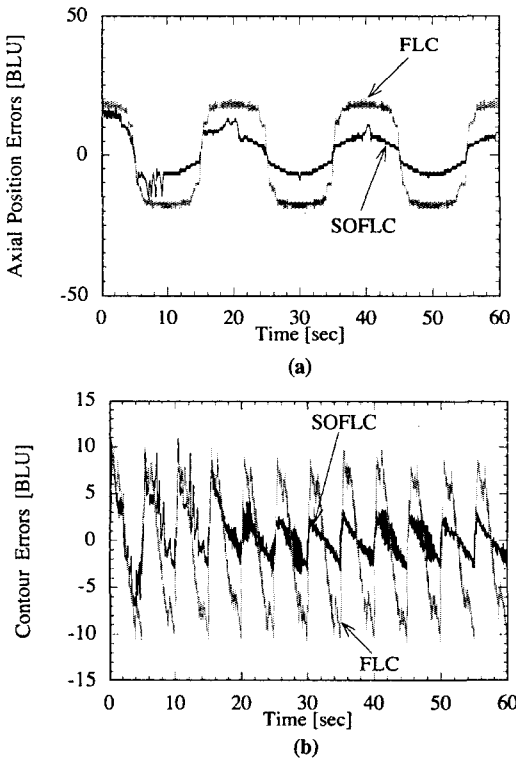


Fig. 7 Simulation comparison of (a) axial position errors and (b) contour errors for a circular contour (radius = 40 mm; feedrate = 0.754 m/min).

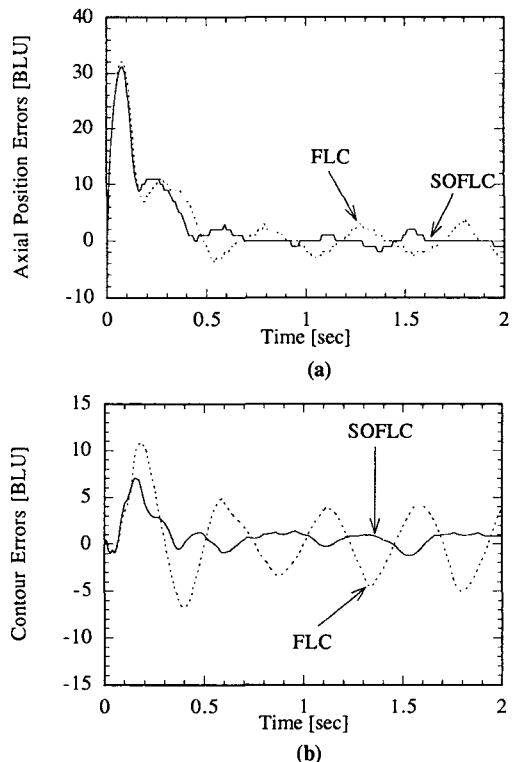


Fig. 8 Simulation comparison of (a) axial position errors and (b) contour errors for a linear contour (feedrate = 0.6 m/min).

±5 BLU의 진폭을 가지고 계속적으로 진동한 반면에 적응 퍼지논리 제어기의 윤곽오차는 약 ±1.5 BLU이다.

● **마찰외란 제거능력:** 마찰외란이 적응 퍼지논리 제어기에 미치는 영향을 살펴보기 위해 앞에서의 원운동에 대한 시뮬레이션을 통해 얻은 조율된 파라미터값들로 이루어진 퍼지논리 제어기를 이용하여 시뮬레이션을 하였다. 반경이 20 mm인 원형 윤곽에 대해 마찰외란의 효과를 증대시키기 위해 0.377 m/min의 저속 이송속도를 사용하였으며, 외란을 입력했을 때와 입력하지 않았을 때에 대해서 3 사이클 동안의 윤곽오차를 비교하였다. 외란이 존재하는 경우가 그렇지 않은 경우에 비해 실효값 윤곽오차가 0.4 BLU에서 1.3 BLU로 미소하게 증가하였다. 그러나, 마찰외란이 존재하는 경우에 전통적인 PID 제어기를 사용한 경우의 윤곽오차와 비교해 보면 이 차이는 매우 작은 양이다. PID 제어기의 경우 동일한 조건하에서 매 90° 회전마다 부착마찰에 의해 반복되는 윤곽오차 값은 14.3 BLU였다. 따라서, 본 논문에서 제시된 적응 퍼지논리 제어기에 뛰어난 외란 제거 성능이 있음을 알 수 있다.

4.2 실험결과

실험 시스템에서는 시뮬레이션 해석에서 쓰였던 자기조정 방법 외에 3절에서 언급한 저속 마찰보정 방법을 포함시켰다. 따라서, 퍼지 출력 싱글톤들은 제어기의 성능뿐만 아니라 저속에서 추정된 정적마찰 및 역점성마찰값에 따라 조정되었다.

제시된 자기조정 방법의 효과를 살펴보기 위해 적응 퍼지논리 제어기와 퍼지논리 제어기를 구현하였고, 이들의 윤곽오차를 비교하였다. 그 대표적인 결과를 Fig. 9와

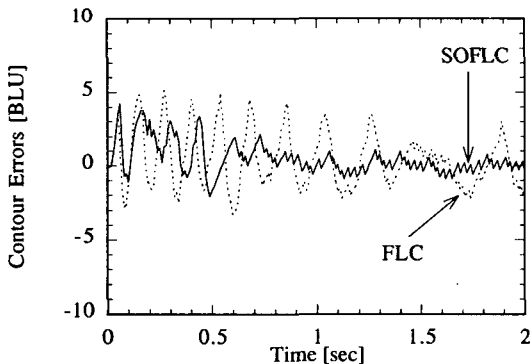


Fig. 9 Experimental comparison of linear contour errors (feedrate = 0.8 m/min).

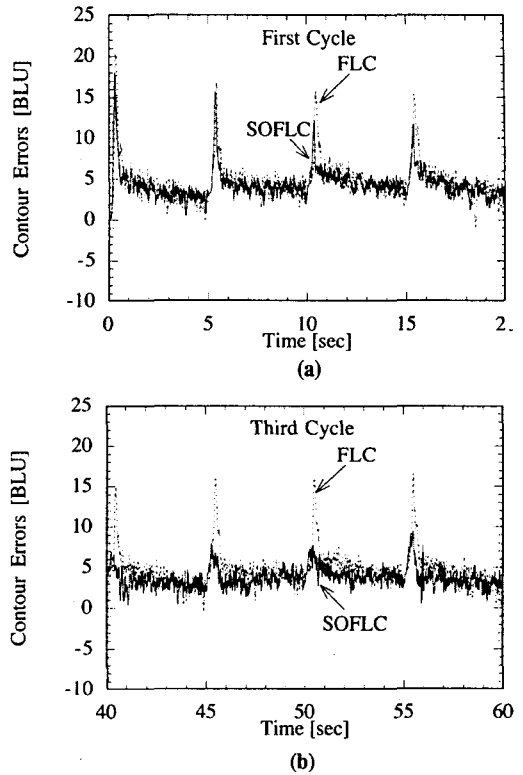


Fig. 10 Experimental comparison of circular contour errors (feedrate = 0.754 m/min): (a) first cycle; (b) third cycle.

Table 4 Initial parameter values for the membership functions.

Membership Functions for e	Left Edge	Center	Right Edge
NL	$-\infty$	-21	-16
NM	-19	-14	-9
NS	-12	-7	-2
ZR	-5	0	5
PS	2	7	12
PM	9	14	19
PL	16	21	∞
Membership Functions for Δe	Left Edge	Center	Right Edge
NL	$-\infty$	-4.5	-3.5
NM	-4.0	-3.0	-2.0
NS	-2.5	-1.5	-0.5
ZR	-1.0	0.0	1.0
PS	0.5	1.5	2.5
PM	2.0	3.0	4.0
PL	3.5	4.5	∞
Membership Functions for u	Centroid		
NL	-32		
NM	-16		
NS	-8		
ZR	0		
PS	8		
PM	16		
PL	32		

Fig. 10에 나타내었는데, Fig. 9는 밀링머신이 선형 윤곽 $y = 10x$ 를 따라 0.8 m/min의 이송속도로 움직이는 경우이고, Fig. 10은 반경이 40 mm인 원형 윤곽을 0.754 m/min으로 움직이는 경우이다. 원운동의 경우 첫째와 세째 사이클에 대한 윤곽오차를 각각 Fig. 10a와 Fig. 10b에 비교하였다. 퍼지논리 제어기의 소속함수들에 대한 파라미터값들(적응 퍼지논리 제어기의 경우에는 초기 파라미터값들)을 Table 4에 나타내었으며, 적분제어 계인은 8.0으로 하였다. 실험결과에서 볼 수 있듯이 자기조정 기구를 통해 일반적인 퍼지논리 제어기를 사용할 때 나타나는 윤곽오차의 진동을 실제적으로 감소시켰다. 위의 실험에서 윤곽오차의 실효값은 Table 5에 나타낸 바와 같이 1/2로 감소하였다. 직선운동에 대해 1초 이

후의 실효값 윤곽오차를 비교하면 1.5 BLU(퍼지논리 제어기)에서 0.4 BLU(적응 퍼지논리 제어기)로 감소하였다. 원운동에 대해서는 매 90° 마다의 최대 윤곽오차가 14.3 BLU에서 6.4 BLU로 감소하였다.

절삭력 외란이 존재하는 조건하에서, 제시된 자기조정 방법의 성능을 살펴보기 위해 적응 퍼지논리 제어기를 이용하여 실제의 가공 실험을 행하였다. 알루미늄 공작물을 엔드밀을 이용하여 슬로팅(slotting) 가공할 때의 대표적인 결과를 Fig. 11에 나타내었다. 이 경우 원형 윤곽의 반경은 20 mm, 이송속도는 0.3 m/min이고, 절삭깊이와 절삭폭은 각각 2 mm와 10 mm이며, 스펀들 회전속도는 1600 rpm이다. 실제의 가공조건하에서도 사이클이 진행됨에 따라 자기조정 기구에 의해 제어기 파라미터들이 자동으로 조율되어서 윤곽오차가 감소하였다. 첫째 사이클과 세째 사이클에 대한 윤곽오차를 각각 Fig. 11a와 Fig. 11b에 나타내었다. 이 그림에서 보는 바와 같이 자기조정 방법에 의해 최대 윤곽오차가 약 1/2로 감소하였다.

원운동과 직선운동에 대해, 제시된 자기조정 기구를 통해 조율된 파라미터들을 갖는 퍼지논리 제어기와 조율된

Table 5 Comparison of the RMS contour errors (unit: 10 μ m).

Contour	SOFLC	FLC
Straight Line	0.9	1.7
Circle	1.9	4.0

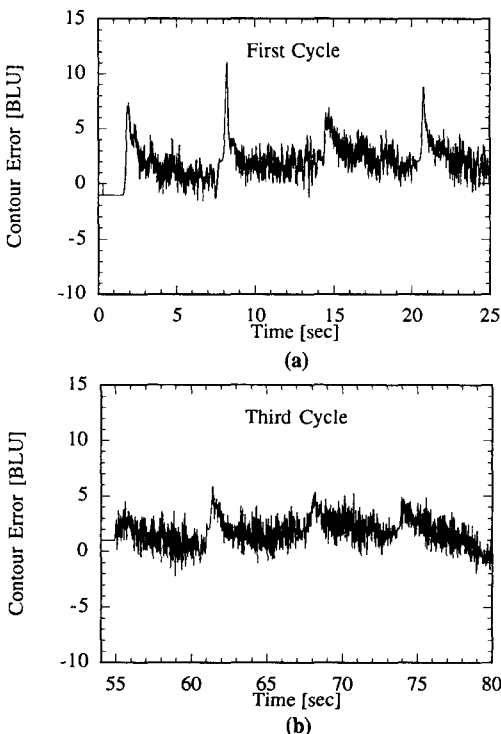


Fig. 11 Experimental contour errors for circular slot milling: (a) first cycle; (b) third cycle.

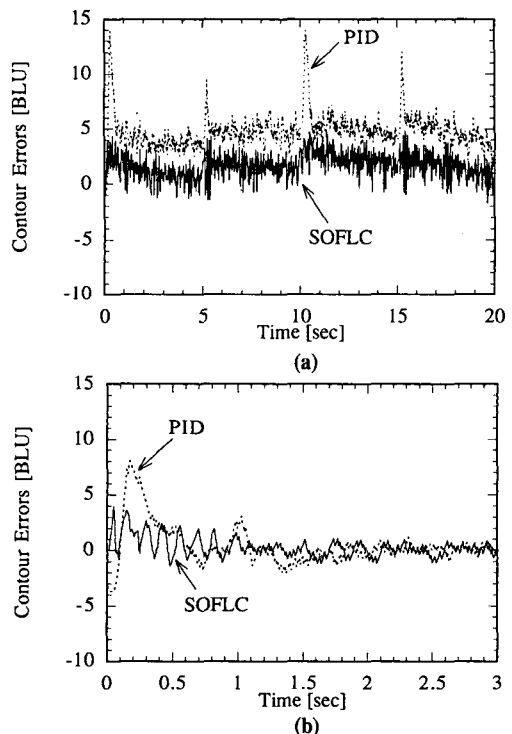


Fig. 12 Experimental comparison of contour errors: (a) circular motion; (b) linear motion.

PID 제어기의 윤곽오차를 Fig. 12에 비교하였다. 본 논문에서 쓰인 PID 제어기의 비례, 적분, 미분 게인은 각각 1.5, 8.1, 0.1이었다. 원운동의 경우 모든 실험에 대해 적응 퍼지논리 제어기가 PID 제어기에 비해 훨씬 좋은 윤곽추적 성능을 보였다. 예를 들어 Fig. 12a의 원운동에 대해 적응 퍼지논리 제어기의 실효값 윤곽오차는 PID 제어기의 윤곽오차와 비교해 1/2.6로 감소하였으며, 최대 윤곽오차는 1/2.9의 감소를 보였다. 직선운동에 대해서는 정상상태에서 두 제어기 사이에 큰 차이가 없었으나, PID 제어기를 사용할 때 생기는 마찰에 의한 초기 윤곽오차가 적응 퍼지논리 제어기에 의해 1/3로 감소하였다 (Fig. 12b참조). 따라서, 가공하고자 하는 윤곽이 다수의 짧은 선분들로 이루어져 있다면, 적응 퍼지논리 제어기에 의해 윤곽추적 성능을 크게 향상시킬 수 있을 것이다.

5. 결 론

공작기계 이송장치의 마찰과 같은 외란이 존재하는 경우에 양호한 윤곽 정밀도를 얻기 위해 새로운 적응 퍼지논리 제어기가 제안되었다. 제시된 방법에서 퍼지논리 제어기의 입력 및 출력 소속함수에 대한 파라미터들은 각 제어규칙의 실시간 성능평가에 따라 실시간으로 조율되었다. 제시된 적응 퍼지논리 제어기는 밀링머신의 각 이송축에 대해 매 샘플주기마다 0.4 msec의 계산 시간 (33 MHz 80486 PC 사용)이 필요했으며, 일반적 퍼지논리 제어기의 경우 0.3 msec가 필요했다. 따라서, 자기조정 기구에 의해 추가되는 계산 시간상의 부담이 작기 때문에 제시된 적응 퍼지논리 제어기는 다축 공작기계에도 적용될 수 있다.

시뮬레이션과 실험을 통하여 제시된 적응 퍼지논리 제어기가 제어기 파라미터들을 효과적으로 조정할 수 있고, 일반적인 퍼지논리 제어기와 PID 제어기에 비해 실제로 윤곽 정밀도를 향상시킬 수 있음을 보였다. 또한, 제시된 제어기는 마찰외란에 강인하고 외란에 의한 윤곽오차를 상당히 줄일 수 있음을 보였다.

참 고 문 헌

1. Jee, S., "A Study on the Friction Compensation in CNC Servomechanisms by Fuzzy Logic Control," *Journal of the Korean Society of Precision Engineering*, Vol. 15, No. 9, pp. 56-67, 1998.
2. Procyk, T. J. and Mamdani, E. H., "A Linguistic Self-Organizing Process Controller," *Automatica*, Vol. 15, pp. 15-30, 1979.
3. Batur, C. and Kasparian, V., "Fuzzy Adaptive Control," *International Journal of Systems Science*, Vol. 24, No. 2, pp. 301-314, 1993.
4. Jee, S. and Koren, Y., "Friction Compensation in Feed Drive Systems Using an Adaptive Fuzzy Logic Control," *Proceedings of the 1994 ASME Winter Annual Meeting, DSC-Vol. 55-2*, pp. 885-893, Chicago, IL., 1994.
5. Bare, W. H., Mulholland, R. J., and Sofer, S. S., "Design of a Self-Tuning Rule Based Controller for a Gasoline Refinery Catalytic Reformer," *IEEE Transactions on Automatic Control*, Vol. 35, No. 2, pp. 156-164, 1990.
6. Zhao, Z.-Y., Tomizuka, M., and Sagara, S., "A Fuzzy Tuner for Fuzzy Logic Controllers," *Proceedings of the 1992 American Control Conference*, pp. 2268-2272, Chicago, IL., 1992.
7. Langari, G. and Tomizuka, M., "Self Organizing Fuzzy Linguistic Control with Application to Arc Welding," *Proceedings of the 1990 IEEE International Workshop on Intelligent Robots and Systems*, pp. 1007-1014, Tsuchiura-shi, Japan, 1990.