

실험적 진화 프로그래밍을 이용한 초정밀 위치 제어

High-Precision Controller Design Using Evolutionary Programming

전 정열, 김 종환

한국 과학기술원 전기 및 전자 공학과

Abstracts Conventional linear control fails to provide precise positioning of a control object under the influence of friction, deadzone, saturation, etc. This paper proposes a high-precision control scheme for a precise point-to-point positioning system, called an X-Y table, even under the same influences above. The proposed scheme is composed of a fuzzy precompensator and a PD controller. The fuzzy precompensator is employed to improve the performance of the PD controller. Its fuzzy rules are obtained from experimental evolutionary programming (EP), not from an expert. The effectiveness of the scheme is demonstrated by experiments on the X-Y table, with a positioning error of within 1 μm .

I 서론

X-Y 테이블은 선반, 밀링머신, SMD 마운터등의 산업기계에 널리 사용되고 있다. 그러므로 X-Y 테이블의 위치 제어에 관한 많은 연구가 있어왔다. 일반적으로 X-Y 테이블은 넓은 의미의 선형 시스템으로 간주될 수 있기 때문에 기존의 PID (비례-적분-미분) 제어기와 같은 선형제어기만으로 적절한 성능을 갖는 위치 제어가 가능하다고 알려져 있다. X-Y 테이블의 제어 사양은 그 적용 대상에 따라 다양하게 주어진다. 최근에는 X-Y 테이블이 정밀 작업에 이용되는 일이 많아지고, 있고 이에 따라 X-Y 테이블 제어기의 높은 정밀도와 빠른 응답속도가 요구되어지고 있다. 이 경우 마찰, 백래쉬 (backlash), 히스테리시스 (hysteresis), 구동기의 포화등 제어 대상 시스템의 비선형성의 영향으로 선형제어기만으로 정밀제어를 수행하기는 매우 어렵다. 예를 들면, 구동기의 포화는 시스템의 과도응답 특성에 나쁜영향을 미친다. 또한, 정지마찰력은 정상상태오차의 원인이 되며, 적분항이 포함된 제어기를 사용하였을 때에는 기준위치 근처에서 진동을 유발한다.

일반적으로 기계부를 정밀하게 제어하고자 할 경우 두가지 접근방식이 있다. 첫째로 보다 정밀한 기계를 설계 제작하여 비선형성등을 제거하여 제어 성능을 높이는 방법이다. 예로, X-Y 테이블에서 운동부의 접촉부에 사용되는 베어링을 볼베어링대신에 에어베어링으로 교체하여 사용한다면 마찰의 영향을 대부분 없앨 수 있다. 그러나 이러한 방법은 전체 시스템을 제작하는 비용을 상승시키기 때문에 상품 경쟁력을 약화 시킬 수 있다. 또 다른 접근 방법은 좀 더 좋은 성능의 제어기를 설계하는 것이다. 최근에 마이크로 프로세서의 비약적인 발전으로 복잡한 제어기를 저렴한 가격으로 실현 시킬 수 있게 되었다. 따라서 두번째 방법에 적합한 제어 알고리즘을 개발할 수 있다면 이 방법이 시스템의 가격 상승없이 정밀 제어를 수행할 수 있는 적절한 방법이 될것이다.

최근에 다양한 위치 제어방법이 제시되었다. 정밀 위치 제어를 위하여 Yang 과 Tomizuka [1] 는 위치 오차에 따라 적절한 넓이를 가지는 펄스를 제어 대상 시스템에 인가하여 제어 목적을 달성하는 PWC (Pulse Width Controller)를 이용한 적응 펄스폭 제어기를 제안하였다. 그러나 이 방법은 한 번의 펄스를 가한후 시스템이 정지할 때까지 기다린 뒤에 다시 제어입력을 계산하여야 하기 때문에 시스템이 목표위치에 도달하기까지 많은 시간이 필요하다는 단점을 가지고 있다. Tung 등 [2]은 반복 제어에 기초를 둔 마찰력보상 기법을 제안하였으나, 적용대상이 제한되어 있다는 단점을 가지고 있다. 또다른 연구자들 [3, 4]은 적응제어 기법을 이용한 제어 방법들도 제안하였으나, 과도응답 시간이 길어 빠르게 수렴해야 하는 시스템에 적용하기가 어렵다.

한편, 퍼지논리에 기반을 둔 제어기법은 제어 대상 시스템의 모델이 불가능하거나 부정확하여도 제어가 가능하다는 장점을 가지고 있다. 이러한 퍼지제어기는 또한 기존의 제어 기법에 비하여 나은 성능을 보이는 경우도 있다. Kim 등 [5]은 사구간을 가지는 시스템을 제어하기위해 퍼지 전단 보상기와 PD (비례-미분) 제어기로 구성된 이층구조의 제어기를 제안하였다. 또한 이들은 이 제어기가 기존의 선형 제어기에 비하여 향상된 과도응답 특성을 가지고 있고, 정상 상태 오차를 제거할 수 있음을 보여 주었다. 이와 같은 이층구조에 의한 제어기의 실험결과는 Kim 등 [6]에 의하여 발표되었으나, 여기서는 퍼지 전단보상기와 PID 제어기로 구성되었다.

본 논문에서는 마찰과 사구간, 제어기 포화등의 비선형성을 가지는 X-Y 테이블의 초 정밀 위치 제어를 위하여 Kim 등 [5]의 이층 구조 퍼지 제어기에 실험적 진화 프로그래밍을 도입한 초 정밀 위치 제어기에 관하여 기술하고자 한다. 여기서 말하는 초 정밀 위치 제어란, X-Y 테이블의 헤드(Head)를 원하는 위치에 1 μm 이내의 정상상태 오차로 도달하도록 하는 것이며, 이 때 헤드가 정상상태에 머무르는 시간은 0.15 초이다. 사용된 위치 검출 센서의 해상도는 1 μm 이다.

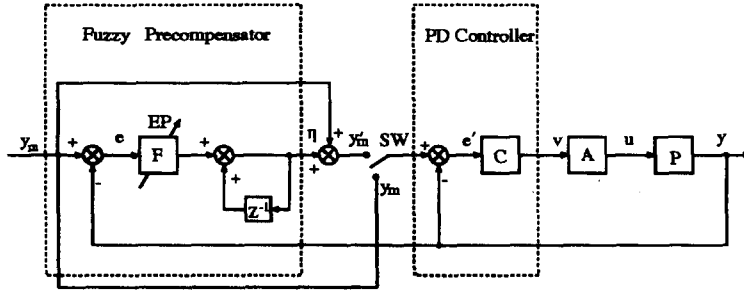


그림 1: 제안된 제어기의 구조

일반적으로 퍼지 제어기의 규칙들은 전문가나 오퍼레이터의 경험으로부터 얻어진다. 그러나 정밀제어기의 경우 이것은 많은 시간을 요하거나 매우 어려운 작업이다. 본 논문에서는 정밀 위치 제어를 위한 퍼지 제어기 결본부의 실수값들을 실험적 진화 프로그래밍 (EEP, Experimental Evolutionary Programming) 기법을 이용하여 구하였다. X-Y 테이블에 적용한 실험결과는 제안된 기법의 우수성을 보여주고 있다.

II 초정밀 위치 제어기

본 절에서는 초정밀 위치제어를 위한 “이층 구조” 제어기에 대하여 설명한다. 그림 1은 전체 제어기의 구조를 보여주고 있다. 전체 제어기는 퍼지 전단 보상기와 PD 제어기(그림 1에서 C로 표시)로 구성된 이층으로 구조로 되어있다. 전단 보상기의 입력은 오차 $e(k)$ 와 오차의 변화량 $\Delta e(k)$ 이고 출력은 $\eta(k)$ 이다.

그림 1에서 스위치는 제어 모드 (mode)를 정상적인 PD 제어기 모드에서 “이층 구조” 제어기 모드로 전환하기 위하여 사용되어진다. 정상 제어모드로서 제어될 대상이 미리 정해진 위치에 도달하기까지 PD 제어기만이 작동한다. 만약 제어대상이 정해진 위치에 도달하면 그 이후부터는 스위치(SW)가 y_m 로 옮겨져 퍼지 전단보상기가 작동 하게되고, 이 퍼지 전단보상기의 출력이 바로 PD 제어기의 입력으로 인가된다.

“이층 구조” 제어기는 PD 제어기의 전단에 퍼지 보상기를 도입함으로써 PD 제어기의 과도응답특성을 향상시키고, 정상상태오차를 제거할 수 있도록 설계 되었다.

지금부터는 “이층 구조”의 첫 번째층을 이루는 퍼지 전단 보상기에 대하여 설명하도록 한다. 본 논문에서 사용되는 퍼지 제어 법칙은 전형적인 퍼지 논리에 기반을 두고 있다. 좀더 자세한 설명이 필요한 독자는 참고문헌 [7]을 참고하기 바란다. 퍼지 전단 보상기의 입력은 “오차”와 “오차의 변화량” 이고 출력은 $\eta(k)$ 이다. 여기서 $e(k)$ 는 $y_m(k) - y(k)$ 로 정의되는 시스템의 오차이며, $\Delta e(k)$ 는 $e(k) - e(k-1)$ 로 정의되는 오차의 변화량이다. 출력 $\eta(k)$ 는 다음 식에 의하여 결정된다:

$$\eta(k) = \eta(k-1) + F[e(k), \Delta e(k)]$$

여기서 $F[e(k), \Delta e(k)]$ 퍼지논리에 기반을 둔 비선형 함수이다.

함수 $F[e(k), \Delta e(k)]$ 의 구현은 퍼지논리에 기반을 두고 있고, 퍼지화와 퍼지추론, 비퍼지화의 3단계로 구성된다. 퍼지 전단 보상기의 입력 “오차”와 “오차의 변화량”은 전단 보상기에 입력되기전에 퍼지화되어야 한다. 퍼지화단계에서 입력 $e(k)$ 와 $\Delta e(k)$ 는 퍼지 집합의 언어변수로 전환되게 된다. 본 논문에서는 다음 5개의 언어변수와 이에 따르는 적합도 함수를 사용하였다:

$$L = \{NB, NS, ZO, PS, PB\}$$

$$M = \{M_{NB}, M_{NS}, M_{ZO}, M_{PS}, M_{PB}\}.$$

각각의 적합도 함수는 실수축에서 구간 $[0, 1]$ 로 변환하는 일종의 맵이다. 적합도 함수는 convex 하다는 조건이 만족되면 어느 형태라도 사용될 수 있으며, 그 예로 종형, 삼각형형, 사다리꼴형등이 있다. 여기서는 계산의 간편함을 위하여 삼각형형의 적합도 함수를 사용하였다. 각 적합도 함수는 그림 2와 같으며 각 언어변수의 의미는 그 약자로부터 알아볼 수 있다. 예를 들면 NB는 negative-big이며, NS는 negative-small, ZO는 zero를 의미한다. 마찬가지로 방법으로 positive (P)에도 적용된다.

각 언어변수 $l \in L$ 에 있어서 관련된 적합도 함수 M_l 에 의하여 입력 $e(k)$ 과 $\Delta e(k)$ 은 다음과 같이 $n_e(l)$ 과 $n_{\Delta e}(l)$ 인 한 쌍의 변수로 설정될 수 있다:

$$n_e(l) = M_l(C_e e(k))$$

$$n_{\Delta e}(l) = M_l(C_{\Delta e} \Delta e(k))$$

이때 C_e 와 $C_{\Delta e}$ 는 배율이다. 이러한 변수들, $n_e(l)$, $n_{\Delta e}(l)$, $l \in L$, 은 다음에 논할 퍼지 논리 판단 과정에서 사용된다.

퍼지 논리 판단 과정에서 퍼지 규칙들의 집합 $R = \{R_1, R_2, \dots, R_{25}\}$ 이 사용된다. 표 1은 퍼지 전단 보상기에서 사용되는 규칙들을 보여주고 있다. 제안된 제어기는 “height defuzzification” 을 사용하기 때문에 각 규칙의

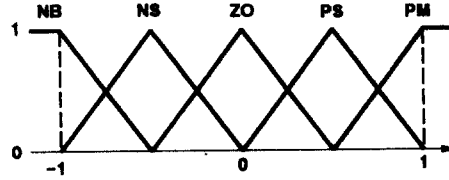


그림 2: 적합도 함수

표 1: 퍼지 전단보상기의 규칙

		$e(k)$				
		NB	NS	ZO	PS	PB
$\Delta e(k)$	NB	μ_1	μ_6	μ_{11}	μ_{16}	μ_{21}
	NS	μ_2	μ_7	μ_{12}	μ_{17}	μ_{22}
	ZE	μ_3	μ_8	μ_{13}	μ_{18}	μ_{23}
	PS	μ_4	μ_9	μ_{14}	μ_{19}	μ_{24}
	PB	μ_5	μ_{10}	μ_{15}	μ_{20}	μ_{25}

결론부인 μ_i 에 언어변수 대신에 실수값을 부여함으로써 퍼지 규칙을 완성할 수 있다. 또한 μ_i 는 진화 프로그래밍에 의하여 최적화 되어질 수 있다. 각 규칙, $R_i, i = 1, \dots, 25$,는 세계의 변수를 ($l_e, l_{\Delta e}, \mu_i$)와 같이 사용한다. 여기서 $l_e, l_{\Delta e}$ 는 언어 변수 집합 L 에 속하는 변수들이다. 규칙들은 보통 다음과 같은 형태로 쓰여진다. “만일 $e(k)$ 가 l_e 이고 $\Delta e(k)$ 가 $l_{\Delta e}$ 이면, μ_i 이다.” 각 규칙, $R_i = (l_e, l_{\Delta e}, \mu_i)$, 들은 주어진 입력쌍 $e(k)$ 와 $\Delta e(k)$ 로부터 다음과 같은 함수 $p_i(e(k), \Delta e(k))$ 로 표현된다.

$$p_i(e(k), \Delta e(k)) = \min(n_e(l_e), n_{\Delta e}(l_{\Delta e})).$$

비퍼지화는 제어동작인 정의된 퍼지 공간에서 비퍼지 공간으로 전환하는 과정이다. 비록 많은 비퍼지화 방법이 있지만 [8]. 본 논문에서는 다음과 같은 “height defuzzification”을 이용하였다:

$$F[e(k), \Delta e(k)] = C_F \frac{\sum_{i=1}^{25} \mu_i p_i(e(k), \Delta e(k))}{\sum_{i=1}^{25} p_i(e(k), \Delta e(k))}$$

여기서 C_F 는 배율이다.

앞에서도 언급했듯이 전단보상기는 다음식과 같은 차등 방정식에 의하여 구해진다:

$$\eta(k) = \eta(k-1) + F[e(k), \Delta e(k)]$$

제안된 “이층 구조” 제어기의 두번째 층은 기존의 PD 제어기의 형식을 따른다. 정상 제어 모드에서의 PD 제어기와 제안된 제어기의 두번째 층에서의 PD 제어기와의 차이는 제안된 제어기의 두번째 층에서는 입력으로 $e(k)$ 와 $\Delta e(k)$ 대신 $e'(k)$ 와 $\Delta e'(k)$ 를 사용한다는 것이다. 여기서 $e'(k) = e(k) + \eta(k)$, $\Delta e'(k) = e'(k) - e'(k-1)$ 이고 $\eta(k) = F[e(k), \Delta e(k), \eta(k-1)]$ 는 전단 보상기의 출력이다. 따라서 PD 제어기의 출력을 표현한 식은 다음과 같이 표현된다:

$$v(k) = K_P e'(k) + K_D \Delta e'(k).$$

그림 1에서 함수 C 는 $e'(k)$ 와 $\Delta e'(k)$ 의 함수로 표현되는 일반적인 제어기를 나타내는 것이다. 즉, 이 경우 $v(k) = C[e'(k), \Delta e'(k)]$ 이다.

이층구조 제어기의 전체적인 동력학식을 정리하면 다음과 같다.

$$\begin{aligned} e(k) &= y_m(k) - y(k) \\ \Delta e(k) &= e(k) - e(k-1) \\ \eta(k) &= F[e(k), \Delta e(k)] + \eta(k-1) \\ y'_m(k) &= y_m(k) + \eta(k) \\ e'(k) &= y'_m(k) - y(k) \\ \Delta e'(k) &= e'(k) - e'(k-1) \\ C[e'(k), \Delta e'(k)] &= K_P e'(k) + K_D \Delta e'(k) \\ v(k) &= C[e'(k), \Delta e'(k)] \\ u(k) &= A[v(k)] \\ y(k) &= P(z)[u(k)] \end{aligned}$$

표 2: 퍼지 전단보상기의 초기 규칙

		$e(k)$				
		NB	NS	ZO	PS	PB
$\Delta e(k)$	NB	-1.0	-1.0	-0.5	-0.5	0.0
	NS	-1.0	-0.5	-0.5	0.0	0.5
	ZE	-0.5	-0.5	0.0	0.5	0.5
	PS	-0.5	0.0	0.5	0.5	1.0
	PB	0.0	0.5	0.5	1.0	1.0

단, 이때 $A[\cdot]$ 는 v 를 u 로 변환하는 포화가 있는 전류 증폭기를 나타낸다.

III 실험적 진화 프로그래밍에 의한 규칙의 최적화

앞 절에서는 제안된 제어기의 구조에 대하여 기술하였다. 다음은 퍼지제어기의 가장 중요한 부분인 퍼지 규칙들을 어떻게 구할 것인가에 관하여 설명한다. 본 절에서는 실험적 진화 프로그래밍을 이용하여 퍼지 규칙을 결정하는 방법에 대하여 기술한다. 진화 프로그래밍은 돌연변이법을 이용하여 새로운 자손을 만들어 내고 이를 확률적인 방법으로 선택하여 점차 최적의 개체를 찾아가는 방법이다 [9]. 여기서 하나의 개체 벡터는 다음과 같이 25개의 퍼지 규칙들로 구성되어 있다.

$$\vec{m}^i = [\mu_1^i, \dots, \mu_{25}^i]^T$$

본 논문에서 사용된 실험적 진화 프로그래밍 과정은 다음과 같이 요약될 수 있다.

1. (초기화) 초기화 단계에서는 퍼지 전단 보상기의 규칙중 후건부의 값 $\mu_j, \forall j \in \{1, \dots, 25\}$ 를 변수로 갖는 실수 벡터쌍 $(\vec{m}^i, \vec{\sigma}^i), \forall i \in \{1, \dots, 20\}$ 20쌍을 초기 개체들로 생성한다. 일반적으로 진화 프로그래밍의 초기화 단계에서 같은 확률을 가지는 임의수를 초기 개체로서 정의한다. 그러나 본 논문에서 사용하는 실험적 진화 프로그래밍은 실제 시스템에서 동작하기 때문에 임의의 규칙을 사용할 경우 전체 시스템이 불안정해질 수 있으며, 심지어 시스템에 피해를 줄 수 있다. 그러므로 본 논문에서 사용되는 초기개체는 퍼지 제어기에서 널리 사용되어지고 전체 시스템의 안정도를 보장하는 규칙들로 구성되어 있다. 표 2는 본 논문에서 사용하는 초기 규칙들을 나타내고 있다. 적용 표준편차인 $\sigma_i, \forall i \in \{1, \dots, 20\}$ 의 초기값은 2.5로 초기화되었다.
2. (평가) 각 개체 \vec{m}^i 에 대하여 적합도에 해당하는 목적 함수인 $\Phi(\vec{m}^i)$ 을 이용하여 다음과 같이 계산한다.

$$\Phi(\vec{m}^i) = \sqrt{\sum_k e(\vec{m}^i, k)^2 + \zeta e_f(l)^2} \quad (1)$$

이 때 k 는 시간을 나타내는 양의 정수이고, l 은 미리 정해진 목표점의 수이다. 또한 ζ 는 양의값의 배율이다. $e(\vec{m}^i, k)$ 는 퍼지 전단 보상기가 작동하기 시작한 이후의 시스템 오차로서 다음과 같이 정의 된다:

$$e(\vec{m}^i, k) = y_m(k) - y(\vec{m}^i, k)$$

여기서 $y_m(k)$ 는 기준 입력이고, $y(\vec{m}^i, k)$ 은 \vec{m}^i 를 퍼지 전단 보상기의 규칙으로 사용했을 경우의 시스템 출력이다. $e_f(l)$ 는 l 번째 정상상태 오차로서 $e_f(l) = y_f(l) - y(\vec{m}^i, k)$ 로 정의된다. 이 때 $y_f(l)$ 은 l 번째 목표점이다. $e_f(l)$ 은 정상상태 오차를 없애기 위해 도입되었으며, ζ 는 4000을 사용하였다.

3. (자손생성) 자손 개체 $(\vec{m}^j, \vec{\sigma}^j)$ 를 다음과 같은 돌연변이 연산자를 이용하여 각 부모 개체 $(\vec{m}^i, \vec{\sigma}^i)$ 로부터 생성해낸다:

$$\begin{aligned} \vec{m}^j(j) &= \vec{m}^i(j) + \vec{\sigma}^i(j) \cdot N_j(0, 1) \\ \vec{\sigma}^j(j) &= \vec{\sigma}^i(j) \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_j(0,1)} \end{aligned}$$

$\forall j \in \{1, \dots, 25\}$, 여기서 $\vec{m}^i(j)$ 와 $\vec{m}^j(j)$, $\vec{\sigma}^i(j)$, $\vec{\sigma}^j(j)$ 는 각각 \vec{m}^i 와 \vec{m}^j , $\vec{\sigma}^i$, $\vec{\sigma}^j$ 의 j 번째 항을 나타낸다. 즉, $\vec{m}^i(j)$ 는 퍼지 규칙의 μ_j^i 를 나타낸다. $N(0, 1)$ 는 평균이 0이고, 표준 편차가 1인 일차의 가우시안 (Gaussian) 랜덤 함수이다. $N_j(0, 1)$ 는 각 개체의 j 번째 항을 생성할 때마다 새로이 얻어지는 랜덤값을 나타낸다. τ, τ' 는 외부 영향력을 나타내는 값으로서 각각 $(\sqrt{10})^{-1}$ 와 $(\sqrt{50})^{-1}$ 로 정의되었다 [10]. 제어기가 안정한 영역에서 동작하여야 하기 때문에 다음의 제한 조건이 자손 개체를 생성하는 조건에 포함되었다.

$$\begin{cases} -2 \leq \vec{m}^i(j) \leq 0 & \text{for } j \in \{1, 2, 3, 4, 6, 7, 8, 11, 12, 16\}, \\ 0 \leq \vec{m}^i(j) \leq 2 & \text{for } j \in \{10, 14, 15, 18, 19, 20, 22, 23, 24, 25\} \\ -2 \leq \vec{m}^i(j) \leq 2 & \text{for } j \in \{5, 9, 17, 21\} \\ \vec{m}^i(j) = 0 & \text{for } j = 13. \end{cases}$$

4. (평가) 자손세대의 목적함수 $\Phi(\bar{m}')$ 를 계산한다.
5. (선택) 각 개체에 대하여 \bar{m}^i 와 \bar{m}^j 의 $2 \times 20 = 40$ 개체중에서 똑같은 선택될 확률을 가지고 선정된 8개의 개체와 성능 비교를 한다. 이때 만약 개체의 성능이 비교하는 개체의 성능보다 더 좋거나 같을 경우 "win"의 수를 하나 추가한다.
 \bar{m}^i 와 \bar{m}^j , $\forall i \in \{1, \dots, 20\}$ 중에서 "win"의 수가 많은 20개의 개체를 다음 세대의 부모로 선택한다. 이 때 이에 따르는 σ_i 또는 σ_j' 도 함께 포함시킨다.
6. (종료 조건 검사) 만족할만한 해가 나오거나 지정된 시간이 지날때까지 단계 3으로 돌아가 전과정을 반복한다.

IV 실험

이번절에서는 제안된 제어기의 효용성을 실험을 통하여 보이고자 한다. 실험의 목표는 X-Y 테이블의 헤드를 원하는 위치에 위치 검출 센서의 해상도인 $1\mu\text{m}$ 이내의 정상상태 오차로 도달하도록 하는 것이며, 이 때 헤드가 목표점에 도달하여 0.15초내에 위치 오차없이 정지해 있어야 한다 본 실험에서는 X 축 위치제어만을 수행하였다.

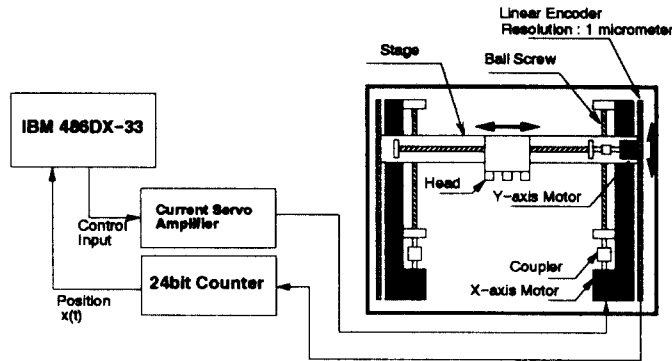


그림 3: 실험장치 구성도

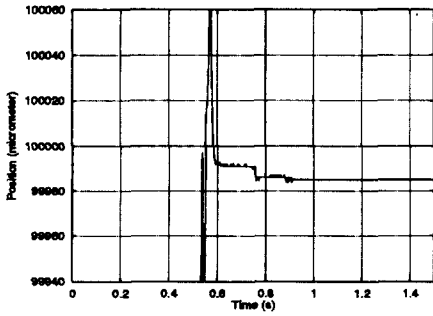
그림 3은 실험장치의 구조를 보여주고 있다. 실험장치는 이동장치와 위치검출을 위한 측정장치, IBM PC로 구성되어 있다. IBM PC에는 아날로그 출력을 위한 D/A 컨버터와 24비트 카운터를 가지고 있는 자작한 카드가 장착되어 있다. 그림에서 볼 수 있듯이 구동기에서 발생한 토크는 lead screw와 ball-nut에 의하여 이동부에 전달된다. 스크류의 리드피치는 0.5 cm 이다. 위치 검출 센서의 해상도는 $1\mu\text{m}$ (10^{-6} m)로 $1\mu\text{m}$ 를 움직일때마다 하나의 펄스가 출력되는 형태이다. 이 펄스를 24 비트 카운터로 세어서 위치를 파악한다. 12비트 D/A 컨버터의 아날로그 출력은 전류 증폭기에 입력되어 전동기에 입력된다. 주 제어 알고리즘은 인텔 i486DX-33 마이크로 프로세서가 장착된 IBM PC 호환 컴퓨터에서 실행되었다. 본 실험에서 샘플링은 1 ms 마다 수행되었고,

그림 4(a)는 PD 제어기를 사용하였을 때의 실험 결과를 목표위치 근처에서 확대한 것이다. 이 때, 목표위치는 $100,000\mu\text{m}$ 이고, 실험결과 $18\mu\text{m}$ 의 정상상태 오차가 발생하였다. 이는 PD 제어기만으로는 마찰이 있는 시스템의 정밀 제어가 불가능하다는 것을 나타내어 준다 [11]. 일반적으로 시스템의 정상상태 오차를 제거하기 위하여 적분항 포함하는 제어기를 사용한다. 따라서 위 PD 제어기에 정상상태 오차를 제거하기 위한 적분항이 더해진 PID 제어기를 구성하여 실험을 수행하였다. 이 때의 결과가 그림 4(b)에 도시되어 있다. 그러나 이 때에도 과도응답특성이 좋지않고, 시스템이 정지할 때까지 걸리는 시간이 작지 않음을 알 수 있다.

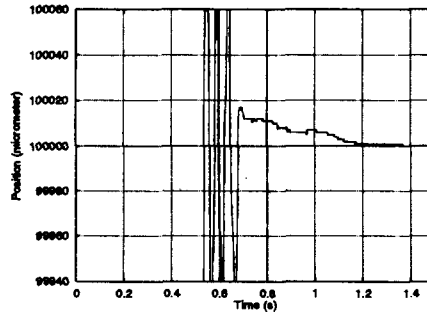
제안된 제어기의 실험을 위하여 먼저 퍼지 전단보상기의 규칙을 얻어야 한다. 앞 절에서 설명한 바와 같이 본 논문에서는 실험적 진화 프로그래밍을 이용하여 퍼지 규칙을 최적화하였다. 퍼지 규칙을 실험적 진화 프로그래밍으로 최적화 하기 위하여 이동부가 직접 이동한 데이터를 가지고 성능을 평가하였다. 본 실험에서는 이동부가 1 cm 에서 출발하여 10 cm 까지 왕복운동한 후 다시 20 cm 까지 왕복운동을 하도록 하였다. 즉, 이 실험의 Evaluation 단계에서는 $y_f(l) = 10\text{ cm}, 1\text{ cm}, 20\text{ cm}, 1\text{ cm}, l = 1, 2, 3, 4$ 를 사용하였다. 여기서 주목하여야 할 것은 진화 프로그래밍이 컴퓨터 시뮬레이션이 아닌 실제 시스템을 제어하여 Evaluation 되었다는 것이다. X-Y의 이동부는 주어진 기준 곡선을 따라 목표점에 도달하여 0.15초 안에 위치 오차없이 정지하여야 한다. 여기서 기준 곡선은 최대 속도가 0.25 m/s 이고 가속도가 1.7 m/s^2 인 2차 곡선이다. 퍼지 전단 보상기의 배율은 다음과 같이 선택되었다:

$$C_e = 0.1, C_{\Delta e} = 0.1 \text{ and } C_F = 9.$$

이동부가 목표점에 $25\mu\text{m}$ 거리내로 접근하였을 때 제어기가 정상 제어기 모드에서 "이층 구조" 제어기모드로 전환된다. 즉, 퍼지 전단보상기는 이동부가 목표점에 $25\mu\text{m}$ 거리내로 접근한 이후 작동된다.



(a) PD 제어기의 실험결과



(b) PID 제어기의 실험결과

그림 4: PD와 PID 제어기의 실험결과

위의 기준 입력을 가지고 III절에 설명한 실험적 진화 프로그래밍이 퍼지 규칙을 최적화 하기 위하여 100 세대에 걸쳐 실행되었다. 100 세대 후에 가장 좋은 성능을 보인 개체를 퍼지 전단 보상기의 규칙으로 결정하였다. 그 결과 얻어진 규칙을 표 3에 나타내었다.

표 3: 실험적 진화프로그래밍으로 얻어진 100세대 후의 규칙

	$e(k)$					
		NB	NS	ZO	PS	PB
$\Delta e(k)$	NB	-1.503	-1.419	-0.147	-0.896	0.045
	NS	-0.403	-0.218	-0.577	-0.013	0.643
	ZE	-0.187	-0.359	0.000	0.755	0.492
	PS	-0.583	0.075	0.618	0.197	0.347
	PB	-0.128	0.658	0.371	1.097	1.483

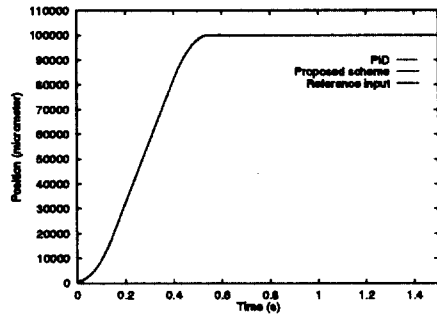
표 3과 같은 규칙을 갖는 제안된 제어기를 가지고 X-Y 테이블의 정밀 위치 제어를 실험하였다. 그림 5(a)는 목표위치가 100,000 μm 일 때 PID 제어기와 제안된 제어기의 결과를 보여주고 있다. 그림 5(b)는 그림 5(a)를 목표위치 근처에서 확대한 그림이다. 실험결과 제안된 제어기가 PID 제어기보다 좋은 과도응답 특성을 가지고 있으며, 더 빠른 시간안에 오차없이 목표 위치에 도달할 수 있음을 보여준다.

V 결론

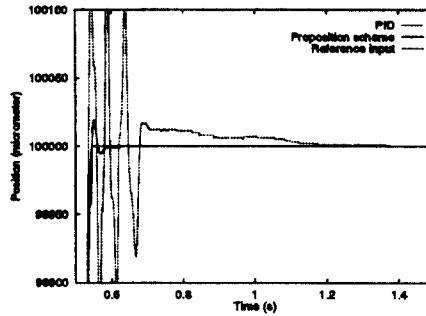
본 논문에서는 위치제어를 위한 초정밀 지능 제어기를 제안하였다. 제안된 제어기는 퍼지 전단보상기와 PD 제어기의 이층 구조로 구성되어 있다. 퍼지 전단보상기의 규칙들은 출력오차를 최소화 하고 정상상태 오차를 제거할 수 있도록 실험적 진화 프로그래밍에 의하여 최적화 되었다. 제안된 제어기는 엔코더 해상도 ($1\mu\text{m}$) 이내의 위치 오차를 가지도록 X-Y 테이블을 제어할 수 있음을 보여주었다. 이러한 초정밀 지능 제어 기술은 SMD 마운터, CNC, 반도체 장비등의 정밀 위치제어가 필요한 장비에 적용될 수 있으며, 이러한 공작기계 및 장비의 고성능화에 기여할 수 있으리라 기대된다.

참고 문헌

- [1] S. Yang and M. Tomizuka, "Adaptive pulse width control for precise positioning under the influence of stiction and coulomb friction," *Journal of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 221-227, September 1988.
- [2] E. D. Tung, G. Anwar, and M. Tomizuka, "Low velocity friction compensation feedforward solution based on repetitive control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 279-284, June 1993.



(a) 시스템의 출력



(b) (a)를 확대한 그림

그림 5: 제안된 제어기와 PID 제어기의 비교

- [3] P. I. Ro and P. I. Hubbel, "Model reference adaptive control of dual-mode micro/macro dynamics of ball screws for nanometer motion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 103-115, March 1993.
- [4] L.-C. Fu, Y. Lin, H.-T. Lee, and J.-S. Chang, "High precision adaptive robust controller for x-y table," in *Proc. of the Asian Control Conference*, pp. 601-603, 1994.
- [5] J.-H. Kim, J.-H. Park, S.-W. Lee, and E. K. P. Chong, "Fuzzy Precompensation of PD Controllers for Systems with Deadzones," *Journal of Intelligent and Fuzzy Systems*, vol. 1, pp. 404-435, 1993.
- [6] Jong-Hwan Kim and Kwang-Choon Kim and Edwin K. P. Chong, "Fuzzy precompensated pid controllers," *IEEE Trns. on Control Systems Technology*, vol. 2, no. 4, pp. 406-411, 1994.
- [7] C. C. Lee, "Fuzzy logic in control systems:fuzzy logic controller - part i, part ii," *IEEE Trans. on Sys., Man, and Cyber.*, vol. 20, no. 2, pp. 404-435, 1990.
- [8] H. Hellendoorn and C. Thomas, "Defuzzification in fuzzy controllers," *Journal of Intelligent and Fuzzy Systems*, vol. 1, no. 2, pp. 109-123, 1993.
- [9] N. Saravanan and D. B. Fogel, "Learning Strategy Parameters in Evolutionary Programming: An Empirical Study," in *Proc. of third Annual Conf. on Evolutionary Programming* (A. V. Sebald and L. J. Fogel, eds.), (Ruer Edge, NJ), pp. 269-280, World Scientific, 1994.
- [10] Thomas Bäck and Günter Rudolph and Hans-Paul Schwefel, "Evolutionary programming and evolutionary strategies: Similarities and differences," in *Proc. of Second Annual Conf. on Evolutionary Programming* (D. Fogel and W. Atmar, eds.), (San Diego, CA), pp. 11-22, Evolutionary Programming Society, February 1993.
- [11] J.-H. Kim, H.-K. Chae, J.-Y. Jeon, and S.-W. Lee, "Identification and control of systems with friction using accelerated evolutionary programming." will be published in *IEEE Control Systems Magazine*, August 1996.