

불안정한 다변수 시스템에 대한 퍼지 학습제어

Fuzzy Learning Control for Multivariable Unstable System

임윤규, 정병목, 소범식

(Yeun-Kyu Lim, Byeong-Mook Chung, and Bum-Sik Soo)

Abstract : A fuzzy learning method to control an unstable and multivariable system is presented in this paper. Because the multivariable system has generally a coupling effect between the inputs and outputs, it is difficult to find its modeling equation or parameters. If the system is unstable, initial condition rules are needed to make it stable because learning is nearly impossible. Therefore, this learning method uses the initial rules and introduces a cost function composed of the actual error and error-rate of each output without the modeling equation. To minimize the cost function, we experimentally got the Jacobian matrix in the operating point of the system. From the Jacobian matrix, we can find the direction of the convergence in the learning, and the optimal control rules are finally acquired when the fuzzy rules are updated by changing the portion of the errors and error rates.

Keywords : fuzzy learning, multivariables system, unstable system, cost function, Jacobian matrix

I. 서론

산업 현장에서 사용되는 대부분의 시스템은 하나 이상의 입력과 출력으로 이루어진 다변수 시스템(multivariable system)이다. 이러한 시스템은 입력과 출력이 일 대 일로 대응되는 경우는 잘 없고, 서로 간에 간섭(coupling)효과를 가지고 있다. 그리고 변수들의 증가로 인한 수학적 모델링이 어려우므로 이러한 시스템의 제어를 설계가 어렵다[1],[2]. 20세기에 들어서 제어 이론의 많은 발전이 있었으나 다변수 시스템에 대한 관심을 가지고 연구를 시작한 것으로 불과 얼마 되지 않는다. 처음에는 제어기 설계의 용이성을 위해 여러 개의 독립된 단일 입출력으로 해석하여 설계하려는 시도가 많으며[3],[4], 현대 제어에서는 최적의 제어를 설계하는 기법을 제시하기도 한다. 이 설계 방법들은 시스템의 모델링이 필요하므로 현장 적용에 어려움이 있다. 최근에는 학습에 의해 스스로 제어를 설계하는 방법에 대한 연구가 활발하다[5]. 그 대표적인 방법이 신경망이론을 이용하여 학습을 하거나 유전자 이론을 이용한 방법이 있는데 이 방법들은 학습에 소요되는 시간이 너무 길기 때문에 온라인(on-line)학습의 성격보다는 오프라인(off-line)학습의 성격이 더 강하다. 따라서 본 논문에서는 시스템의 수학적 모델링을 구하기 힘든 현장에 적용할 수 있으며, 학습 시간이 적게 걸리기 때문에 실시간 학습이 가능한 학습법을 이용하여 불안정한 다변수 시스템의 제어기 설계 방법을 제안하고자 한다. 이 학습법은 시스템의 원하는 출력이 시스템의 역모델의 출력과 실제 출력의 차인 에러와 에러 변화율로 이루어진 비용함수를 정의한 다음 이 비용함수를 최소화하기 위해 기울기법(gradient descent approach)을 이용하여 원하는 출력을

설계하는 방법이다. 이 학습법을 비선형이며 불안정한 다변수 시스템인 도립 진자 시스템의 진동차과 대차의 위치를 동시에 제어하는 제어기 설계에 적용하였다. 2장에서는 다변수 시스템의 퍼지 제어기와 퍼지 학습법에 대해 살펴보고, 3장에서는 시뮬레이션을 위한 시스템의 운동 방정식과 자코비안 매트릭스의 단조성을 검증하여 학습 적용 여부를 살펴본다. 그리고 4장에서는 시뮬레이션 방법과 결과를 고찰하고 5장에서 결론을 도출한다.

II. 다변수 시스템에 대한 퍼지 학습법

1. 퍼지 제어

퍼지 논리 제어기란 프로세스의 입력과 출력의 애매한 관계를 인간의 경험과 지식을 기초로 한 언어적 제어 규칙으로 표현하여, 이를 제어원리로 변환시킨 것이다. 일반적인 제어입력 $u(k)$ 는 다음과 같이 표현할 수 있다.

$$u(k) = f [y_{d1}(k+n), \dots, y_1(k), \dots, y_1(k-p_1+n), \\ \vdots \\ y_{dJ}(k+n), \dots, y_J(k), \dots, y_J(k-p_J+n), \\ u(k-1), \dots, u(k-m)] (j=1,2,\dots,J) \quad (1)$$

여기서, J 는 변수의 개수, n 은 시스템의 시간지연, m 은 입력 차수, p_j 는 시스템의 j 번째 변수의 차수이며, $y_j(k)$ 는 j 번째 변수의 k 번째 출력이고, $y_{dj}(k+n)$ 는 j 번째 변수의 원하는 $(k+n)$ 번째 출력이다. (1)에서 알 수 있듯이 현재의 제어 입력을 추정하기 위해서는 미래의 출력에 대한 예측이 가능해야 한다. 따라서 시스템의 역모델을 구하기 위해 (2)와 같이 퍼지 제어 입력을 변수들의 에러와 에러 변화율에 대한 함수로 나타낼 수 있다.

$$u(k) = F[e_1(k), c_1(k), \dots, e_j(k), c_j(k), \dots, e_j(k), c_j(k)] \quad (2)$$

여기서, e_j 는 j 번째 변수의 에러, 즉 $e_j(k) = y_{d_j}(k) - y_j(k)$

이고, c_j 는 j 번째 변수의 에러 변화율, 즉 $c_j(k) = (e_j(k) - e_j(k-1)) / \Delta t$ 이다. 퍼지 합인 R은 (3)과 같이 표현할 수 있다.

$$R : E_l(e_l) \wedge C_l(c_l) \cdots \wedge E_j(e_j) \wedge C_j(c_j) \rightarrow v \quad (3)$$

여기서, 퍼지 후건부의 값 v 는 계산의 용이성 때문에 비퍼지 값(defuzzified value)으로 하였다. 위 규칙을 1번째 제어규칙 R1로 확장하면 (4)와 같이 표현할 수 있다.

$$R_l : E_{l1}(e_{l1}) \wedge C_{l1}(c_{l1}) \wedge \cdots \wedge E_{lj}(e_{lj}) \wedge C_{lj}(c_{lj}) \rightarrow v_l \quad (4)$$

$(l=1,2,\dots,L)$

여기서, L 은 전체 규칙의 개수이다. 위 (4)를 멤버십 함수로 표현하면 (5)와 같이 표현할 수 있다.

$$\mu_{R_l} = \mu_{E_{l1}(e_{l1})} \wedge \mu_{C_{l1}(c_{l1})} \wedge \cdots \wedge \mu_{E_{lj}(e_{lj})} \wedge \mu_{C_{lj}(c_{lj})} \rightarrow v_l \quad (5)$$

위 식을 토대로 후건부의 값을 얼마나 반영해야 할 것인가를 결정하는 l 번째 규칙의 적합도(fitness) ϕ_l 는 (6)과 같이 계산할 수 있다.

$$\phi_l = \frac{\mu_{E_{l1}(e_{l1})} \wedge \mu_{C_{l1}(c_{l1})} \wedge \cdots \wedge \mu_{E_{lj}(e_{lj})} \wedge \mu_{C_{lj}(c_{lj})}}{\sum_{l=1}^L \mu_{E_{l1}(e_{l1})} \wedge \mu_{C_{l1}(c_{l1})} \wedge \cdots \wedge \mu_{E_{lj}(e_{lj})} \wedge \mu_{C_{lj}(c_{lj})}} \quad (6)$$

위에서 구한 적합도 ϕ_l 와 후건부 값 v_l 을 곱하면 제어 입력을 구할 수 있다. 이를 (7)에 나타내었다.

$$u(k) = \sum_{l=0}^L (\phi_l \times v_l) \quad (7)$$

제어 입력 값을 구하기 위해서는 퍼지 규칙을 얻어야만 한다. 그러나 많은 변수들로 구성된 퍼지 규칙을 경험자 혹은 전문가의 지식으로는 구하기 어렵기 때문에 다음 장에서 언급되는 학습에 의해 이 규칙들을 구하고자 한다.

2. 학습 알고리즘

도립 진자 시스템을 제어하기 위한 제어 규칙의 학습은 모델규범형 퍼지 학습 제어기를 사용하였다. 이 학습법은 원하는 출력과 시스템의 실제 출력의 차이를 구한 다음, 이를 퍼지 제어 규칙에 피드백하여 과거에 사용되었던 규칙을 수정하는 과정을 반복하여 제어기를 설계하는 학습법이다. 이 학습법에 대한 블록선도를 그림 1에 나타내었다. 여기서 n 은 시간 지연 스텝(step)을 나타낸다. 이 블록선도는 모델의 $(k-n)$ 번째 입력인 $r(k-n)$ 에 대해 모델의 k 번째 출력 매트릭스인 $Y_m(k)$ 과 실제 시스템의 k 번째 출력 매트릭스인 $Y_p(k)$ 의 차이인 $E(k)$ 를 구한다. 이를 n 스텝 시간지연을 시킨 $E(k)$ 를 학습에 반영한다. 다변수 시스템에 적용하기 위해서 앞에서 언급한 각각의 변수들에 대한 것을 (8)-(11)에 나타내었다.

$$Y_m(k) = [y_{1m}, y_{2m}, \dots, y_{jm}, \dots, y_{Lm}] \quad (8)$$

$$Y_p(k) = [y_{1p}, y_{2p}, \dots, y_{jp}, \dots, y_{Lp}] \quad (9)$$

$$E(k) = [e_1, c_1, \dots, e_j, c_j, \dots, e_L, c_L] \quad (10)$$

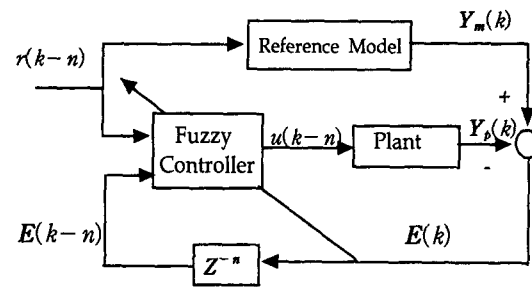


그림 1. 퍼지 학습법에 대한 블록선도.
Fig. 1. Block diagram of the fuzzy learning.

$$E(k-n) = [e_1(k-n), c_1(k-n), \dots, e_j(k-n), c_j(k-n)] \quad (11)$$

비용함수를 (12)와 같이 에러와 에러변화율들의 함수로 설정하였다.

$$J_j(k) = \frac{w_{e_j}}{2} e_j(k)^2 + \frac{w_{c_j}}{2} c_j(k)^2 \quad (12)$$

여기서, w_{e_j} 와 w_{c_j} 는 j 번째 에러와 에러 변화율에 대한 가중치 상수이다. 이때 최적의 제어 규칙을 찾기 위해서는 현재와 미래의 전구간의 비용 함수를 최소화하여야 한다. 이를 최소화하기 위해 기울기법을 사용하여 다음 (13)과 같이 표현하였다.

$$\Delta v_l(k) \propto - \frac{\partial \left[\sum_{j=1}^L \sum_{k=0}^{\infty} J_j(k+n) \right]}{\partial v_l(k)} \quad (13)$$

그러나 전구간을 탐색한다는 것은 불가능하다. 그래서 입력이 가장 크게 영향을 미치는 출력에 대해서만 고려하려고 한다. 즉 입력 $u(k)$ 가 n 스텝 후의 출력, $y_j(k+n)$ 에 가장 크게 영향을 준다고 가정한다면 (13)은 (14)와 같이 나타낼 수 있다.

$$\Delta v_l(k) \propto - \frac{\partial \left[\sum_{j=1}^L J_j(k+n) \right]}{\partial v_l} \quad (14)$$

(14)는 다음 (15)와 같이 바꾸어 표현할 수 있다.

$$\Delta v_l(k) \propto - \frac{\partial \left[\sum_{j=1}^L J_j(k+n) \right]}{\partial u(k)} \frac{\partial u(k)}{\partial v_l(k)} \quad (15)$$

(12)를 이용하여 (15)를 풀면 (16)과 같이 된다.

$$\Delta v_l(k) \propto \sum_{j=1}^L \left[w_{e_j} e_j(k+n) \frac{\partial y_j(k+n)}{\partial u(k)} + w_{c_j} c_j(k+n) \frac{\partial [y_j(k+n) - y_j(k+n-1)]}{\partial u(k)} \right] \frac{\partial u(k)}{\partial v_l(k)} \quad (16)$$

위 식에서 $\frac{\partial y_j(k+n) - y_j(k+n-1)}{\Delta t} = \dot{y}_j(k+n)$ 이며,

(7)을 이용하면 $\frac{\partial u(k)}{\partial v_l(k)} = \phi_l$ 와 같이 된다. 여기서 Δt 는 샘플링 시간(sampling time)이다. 이를 (16)에 대입하면 (17)과 같이 나타낼 수 있다.

$$\Delta v_i \propto \sum_{j=0}^L [w_{e_j} \frac{\partial y_j(k+n)}{\partial u(k)} e_j(k+n) + w_{c_j} \frac{\partial \dot{y}_j(k+n)}{\partial u(k)} \Delta t c_j(k+n)] \phi_i \quad (17)$$

여기서, 각 변수의 에러와 에러 변화율에 곱해진 $\frac{\partial y_j(k+d)}{\partial u(k)}$, $\frac{\partial \dot{y}_j(k+d)}{\partial u(k)}$ 는 자코비안 매트릭스이다. 이를 정리하면 (18)과 같이 나타낼 수 있다.

$$f_{jaco} = \left[\frac{\partial y_1(k+n)}{\partial u(k)} \quad \frac{\partial \dot{y}_1(k+n)}{\partial u(k)} \quad \dots \quad \frac{\partial y_j(k+n)}{\partial u(k)} \quad \frac{\partial \dot{y}_j(k+n)}{\partial u(k)} \right] \quad (18)$$

이 자코비안 매트릭스 f_{jaco} 가 단조 함수일 때, 즉 부호가 변하지 않는다면 부호가 같은 임의의 상수값을 사용해도 학습이 수렴한다는 것을 입증한 바가 있기[2] 때문에 (18)은 임의의 상수 매트릭스로 대체할 수 있다. 따라서 상수항들을 묶어 (19)와 같이 학습률로 정의할 수 있다.

$$w_{e_j} \frac{\partial y_j(k+n)}{\partial u(k)} = \eta_{e_j}, \quad w_{c_j} \frac{\partial \dot{y}_j(k+n)}{\partial u(k)} \Delta t = \eta_{c_j} \quad (19)$$

위 (19)를 (17)에 대입하면 (20)과 같이 간단히 나타낼 수 있다.

$$\Delta v_i = \eta_i \sum_{j=0}^L [\eta_{e_j} e_j(k+n) + \eta_{c_j} c_j(k+n)] \quad (20)$$

여기서, η_i 는 전체 학습률로 학습에서의 탐색 구간이고, η_{e_j} 는 j 번째 변수의 에러의 학습률이고, η_{c_j} 는 j 번째 변수의 에러 변화율의 학습률이다. (20)은 비유함수를 최소화하는 방향의 기울기를 나타내므로, 규칙의 수정은 (21)과 같이 나타낼 수 있다.

$$v_{k(new)} = v_{k(old)} + \Delta v_i \quad (21)$$

III. 시스템의 운동 방정식 및 학습 적용

도립 진자 시스템은 그림 2와 같이 대차가 도립진자가 달려있어 힘을 가하게 되면 대차가 이동하고, 이 이동 때문에 도립 진자의 운동에 영향을 주게되는 시스템이다. 아래 (22)와 (23)에서 보듯이 시스템의 출력이 진자의 운동각과 대차의 이동 위치인 다변수 시스템이다. 이 두 변수를 제어하는 퍼지 제어를 학습을 이용하여 설계하고자 한다. 퍼지 시스템을 사용하기 때문에 시스템의 모델링은 필요가 없으나 본 논문에서는 실제 실험을 하지 않고 시뮬레이션을 통해 그 결과를 보려고 하기 때문에 그림 2와 같이 모델링을 하였다. 여기서 F 는 시스템의 입력인 힘이고, x 는 첫 번째 출력인 대차의 이동 거리이고 θ 는 두 번째 출력인 진자의 진동각이다. 이 모델링을 바탕으로 운동방정식을 구하면 (22)와 (23)과 같이 나타낼 수 있다. 여기서 (22)는 도립 진자의 진동각에

$$\ddot{\theta} = \frac{F \cos \theta - (m_p + m_c) g \sin \theta + m_p l_p \cos \theta \sin(\theta) \dot{\theta}^2}{m_p l_p \cos^2 \theta - 4/3 l (m_p + m_c)} \quad (22)$$

$$\ddot{x} = \frac{-4/3 F - 4/3 m_p l_p \sin \theta \dot{\theta}^2 + m_p g \sin \theta \cos \theta}{m_p \cos^2 \theta - 4/3 (m_p + m_c)} \quad (23)$$

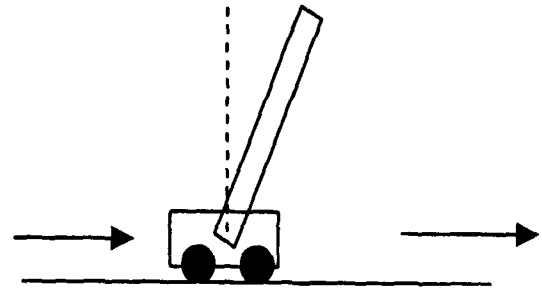


그림 2. 도립 진자 시스템의 모델링.
Fig. 2. Modeling for inverted pendulum system.

관한 식이고, (23)은 대차의 이동거리에 관한 식이다.

여기서, m_c 는 대차의 무게, m_p 는 진자의 무게, l_p 는 진자의 길이, F 는 제어 입력인 힘, 그리고 g 는 중력 가속도이다. 위 시스템을 퍼지 제어에 적용하기 위해서 앞 (12)를 도립 진자 시스템에 적용하면 (24)와 같이 적용할 수 있다.

$$J(k) = \frac{w_{e_\theta}}{2} e_\theta(k)^2 + \frac{w_{c_\theta}}{2} c_\theta(k)^2 + \frac{w_{e_x}}{2} e_x(k)^2 + \frac{w_{c_x}}{2} c_x(k)^2 \quad (24)$$

그리고 $e_\theta(k)$ 와 $c_\theta(k)$ 는 진자의 진동각 θ 의 k 번째 에러와 에러의 변화율이고 $e_x(k)$ 와 $c_x(k)$ 는 대차의 이동 거리 x 의 k 번째 에러와 에러의 변화율이다. 위 식을 최소화하기 위해서는 규칙을 다음과 같이 수정해야한다.

$$\Delta v_i = \eta_i [\eta_{e_\theta} e_\theta(k+n) + \eta_{c_\theta} c_\theta(k+n) + \eta_{e_x} e_x(k+n) + \eta_{c_x} c_x(k+n)] \quad (25)$$

이때 η 는 각 상태변수에 대한 학습률이고 n 은 학습 지연 스텝이다. 자코비안 매트릭스에 대한 (18)은 (26)과 같이 나타낼 수 있다.

$$f_{jaco} = \left[\frac{\partial \theta(k+n)}{\partial F(k)} \quad \frac{\partial \dot{\theta}(k+n)}{\partial F(k)} \quad \frac{\partial x(k+n)}{\partial F(k)} \quad \frac{\partial \dot{x}(k+n)}{\partial F(k)} \right] \quad (26)$$

앞에서도 언급한 것처럼 (25)를 이용하여 학습을 수행하기 위해서는 자코비안 매트릭스가 단조함수일 때에만 가능하므로 이를 검증해야 한다. 시스템 변수는 대차의 무게 m_c 를 1 Kg, 진자의 무게 m_p 를 0.1 Kg, 진자의 길이 l_p 를 0.5 m로 하였고 샘플링 시간을 0.01sec로 하여 입력 F 의 변화에 따른 4개의 출력($\theta, \dot{\theta}, x, \dot{x}$)의 변화를 실험하였다. 이 때 도립 진자 시스템이 원점에서 수직인 상태($\theta=0.0rad, \dot{\theta}=0.0rad/sec, x=0.0m, \dot{x}=0.0m$)에서 힘을 -100N에서 100N까지 변화시켰을 때 입력의 영향을 충분히 받을 수 있는 10스텝이 지난 후의 각 출력의 변화를 그림 3에서 6가지에 나타내었다. 그림에서 알 수 있듯이 입력의 변화에 대해 $\theta, \dot{\theta}, x, \dot{x}$ 의 기울기 방향이 변하지 않으므로 자코비안 매트릭스가 단조함수임을 알 수 있고 따라서 (25)를 이용하여 자코비안 매트릭스를 임의의 상수로 대체하여[2],[6], 학습을 수행하였다.

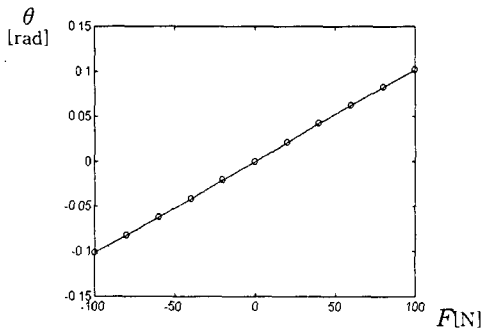


그림 3. 입력 F 의 변화에 대한 θ 의 변화.
Fig. 3. Graph of θ in changing input F .

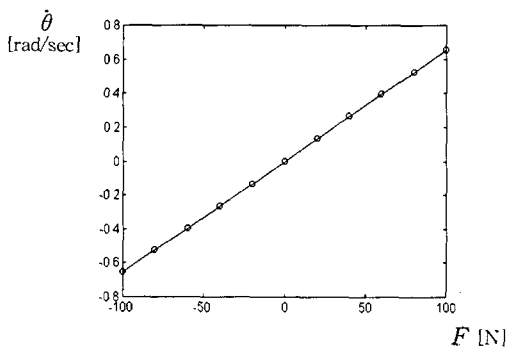


그림 4. 입력 F 의 변화에 대한 $\dot{\theta}$ 의 변화.
Fig. 4. Graph of $\dot{\theta}$ in changing input F .

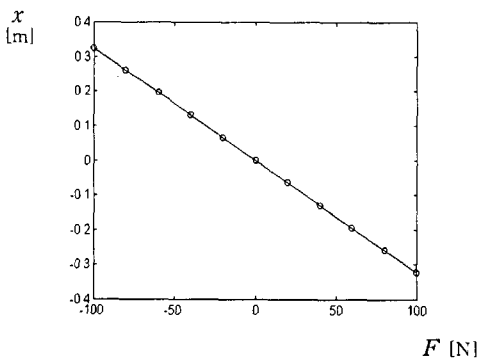


그림 5. 입력 F 의 변화에 대한 x 의 변화.
Fig. 5. Graph of x in changing input F .

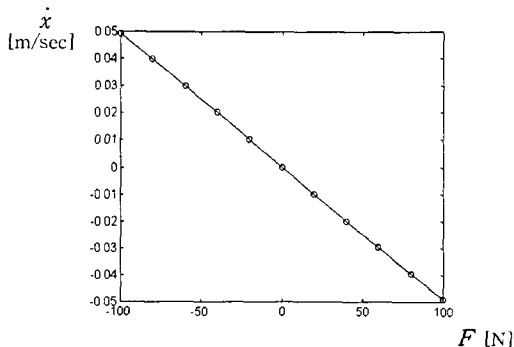


그림 6. 입력 F 의 변화에 대한 \dot{x} 의 변화.
Fig. 6. Graph of \dot{x} in changing input F .

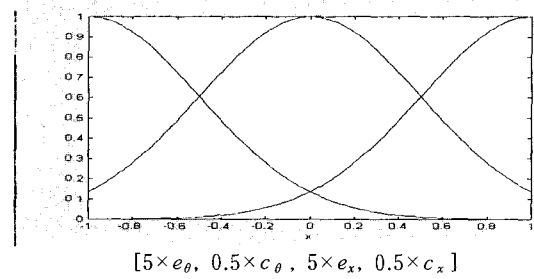


그림 7. $e_\theta, c_\theta, e_x, c_x$ 에 대한 양자화.
Fig. 7. Quantization of e_θ, c_θ, e_x and c_x .

IV. 시뮬레이션

1. 시뮬레이션 조건 및 초기 안정화

학습을 수행하기 위한 시뮬레이션 조건은 $m_c=1\text{kg}$, $m_p=0.1\text{kg}$, $l_p=0.5\text{m}$ 로, $\Delta t=0.01\text{sec}$ 로 하였으며, 초기 조건은 $(\theta=-0.3\text{rad}, \dot{\theta}=0\text{rad/sec}, x=0\text{m}, \dot{x}=0\text{m/sec})$ 으로 하였다. 그리고 각 변수($e_\theta, c_\theta, e_x, c_x$)의 퍼지 양자화(quantization)는 그림 7과 같이 3개씩으로 나누었기 때문에 사용한 규칙의 수는 $81(3 \times 3 \times 3 \times 3)$ 개였고 퍼지 후 건부의 값 v 는 앞에서도 언급한 것처럼 비퍼지 값을 사용하였다.

도립 진자 시스템은 불안정한 시스템이므로 먼저 안정화를 시켜야 한다[7],[8]. 이 시스템에서는 θ 가 불안정하므로 이를 안정화시키는 초기 규칙을 만들었다. 즉, θ 의 에러와 에러 변화율로 이루어진 2차원 규칙에 대해서는 다음과 같이 2개의 규칙만으로도 안정화된다. "If e_θ is P(positive) and c_θ is P, then u is N(negative). or if e_θ is N and c_θ is N, then u is P." 이를 θ 와 x 의 에러와 에러 변화율에 대한 4차원 퍼지 규칙으로 확장하면, "If e_θ is P and c_θ is P and e_x is all states(P, N or zero) and c_x is all states, then u is N. or if e_θ is N and c_θ is N and e_x is all states and c_x is all states, then u is P."와 같이 나타낼 수 있다. 여

표 1. 초기 규칙.

Table 1. Fuzzy rules for initial stability.

			θ									
			Negative			Zero			Positive			
			x			x			x			
			N	Z	P	N	Z	P	N	Z	P	
$\dot{\theta}$	N	\dot{x}	N	100	100	100	0	0	0	0	0	0
			Z	100	100	100	0	0	0	0	0	0
			P	100	100	100	0	0	0	0	0	0
	Z	\dot{x}	N	0	0	0	0	0	0	0	0	0
			Z	0	0	0	0	0	0	0	0	0
			P	0	0	0	0	0	0	0	0	0
	P	\dot{x}	N	0	0	0	0	0	0	-100	-100	-100
			Z	0	0	0	0	0	0	-100	-100	-100
			P	0	0	0	0	0	0	100	100	100

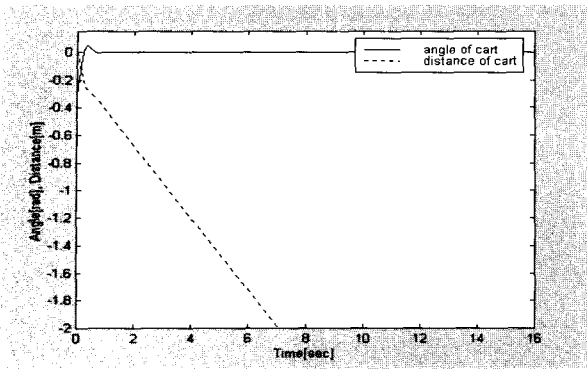


그림 8. 초기 안정화한 응답.
Fig. 8. Response of initial stability.

기서 입력 u 는 최대 최소값을 $\pm 100N$ 으로 각각 선정하여 초기 안정화를 하였다. 그 결과는 그림 8에 나타났고, 이 때의 퍼지 규칙을 표1에 나타내었다. 이 결과에서 불안정했던 θ 가 안정화 되었음을 알 수 있다. 그러나 x 는 무한히 발산하게 되었다. 이 응답을 초기 응답으로 사용하여 퍼지 학습법의 변수인 학습률을 조정하게 된다.

2. 시뮬레이션 결과 및 학습조건

3장에서 언급한 (25)에서 알 수 있듯이 학습 변수 6개, 즉 전체 학습률 η_t , 진자의 진동각에 대한 학습률 η_{e_θ} 와 η_{c_θ} , 대차의 이동거리에 대한 학습률 η_{e_x} 와 η_{c_x} , 그리고 학습지연 스텝 n 을 선정해야 한다. 단일 입출력 시스템에서 에러와 에러 변화율에 대한 학습률, 그리고 학습 지연이 가지는 성질을 보면, 에러의 학습률을 크게 하면 상승 시간(rising-time)이 짧아지지만 과도 응답(overshoot)이 커진다. 반면에 에러 변화율을 크게 하면 상대적으로 과도 응답을 줄일 수 있지만 상승 시간이 길어진다. 그리고 학습 지연을 크게 하게 되면 에러 변화율과 유사한 성질이 있다. 일반적으로 학습 지연은 시스템의 지연 보다 크게 선정해야 하므로[7],[8] 이 시스템의 지연은 2스텝이므로 학습 지연을 5스텝으로 정하였다. 그리고 각 변수에 대한 학습률을 선정 방법은 초기 안정화를 시킨 응답, 그림 8을 토대로 하여 선정한다. 따라서 그림 8에서 보듯이 θ 의 에러 보다 x 의 에러가 매우 크기 때문에 x 의 에러를 더 많이 고려해야 한다. 그래서 학습률을 $\eta_{e_\theta}=1.0(0.9\%)$, $\eta_{c_\theta}=0.1(2.2\%)$, $\eta_{e_x}=0.1(78\%)$, $\eta_{c_x}=0.1(18.7\%)$ 로 선정하였다. 이는 시뮬레이션의 시간인 16초 동안의 e_θ , c_θ , e_x , c_x 의 합을 100%로 볼 때 퍼지 규칙의 변화에 기여한 기여도는 진동각의 에러에 대해서는 0.9%, 에러 변화율에 대해서는 2.2%, 이동거리의 에러에 대해서는 78%, 에러 변화율에 대해서는 η_t 18.7%이다. 그리고 전체 학습률 η_t 은 1.5로 하여 학습하였다.

그 결과는 그림 9에 나타내었다. 이 응답을 보면 x 의 진동이 크음을 알 수 있다. 따라서 이를 줄이기 위해 η_{c_x} 의 값을 0.2로 증가시켜 학습하였다. 그 결과는 그림 10에 나타낸 것처럼 x 의 큰 진동은 줄었지만 θ 의

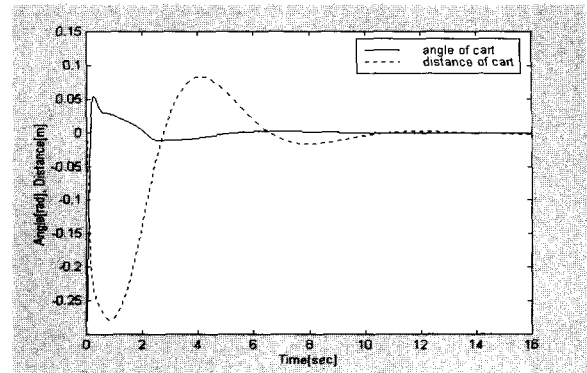


그림 9. 퍼지 학습한 결과 I.
Fig. 9. Fuzzy learning response I.
($\eta_t=1.5$, $n=5$, $\eta_{e_\theta}=1.0$, $\eta_{c_\theta}=0.1$, $\eta_{e_x}=0.1$, $\eta_{c_x}=0.1$.)

표 2. 학습 후의 퍼지 규칙.

Table 2. Fuzzy rules after fuzzy learning.

		θ										
		Negative			Zero			Positive				
		x			x			x				
		N	Z	P	N	Z	P	N	Z	P		
N	\dot{x}	N	100.57	101.99	99.95	2.93	1.30	-2.27	-3.21	1.63	-0.22	
		Z	102.62	115.05	101.86	8.80	6.30	-4.26	13.16	17.38	-0.12	
		P	100.80	103.80	102.10	4.19	1.46	-2.86	1.92	1.55	-0.44	
θ	Z	\dot{x}	N	2.74	0.65	-2.80	18.47	3.52	-20.06	-15.35	-7.31	-2.67
			Z	4.07	0.35	1.72	31.55	-0.09	-31.85	-1.66	-0.34	-4.13
			P	2.68	7.45	15.53	20.37	-2.63	-18.81	2.88	-0.34	-2.74
P	\dot{x}	N	0.44	-1.47	-1.91	2.76	-1.21	-4.10	-102.07	-103.76	-100.79	
		Z	0.12	-17.47	-13.19	4.28	-6.35	-8.92	-101.86	-115.15	-102.65	
		P	0.22	-1.59	3.28	2.36	-0.90	-2.91	-99.94	-101.93	-100.56	

진동이 발생하면서 x 또한 초기에 떨림이 생김을 알 수 있다. 따라서 θ 의 진동을 줄이기 위해 η_{c_θ} 를 0.15로 증가시켰다. 그 결과 그림 11에 나타낸 것처럼 θ 와 x 의 좋은 응답을 얻을 수 있었다. 이때의 학습 변수들은 $\eta_t=1.5$, $\eta_{c_\theta}=1.0$, $\eta_{c_\theta}=0.15$, $\eta_{e_x}=0.1$, $\eta_{c_x}=0.2$, $n=5$ 이다.

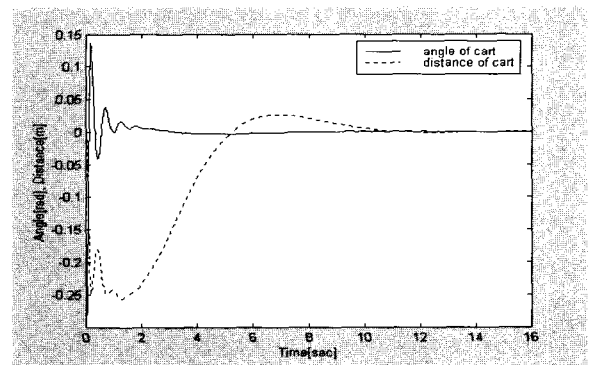


그림 10. 퍼지 학습한 결과 II.
Fig. 10. Fuzzy learning response II.
($\eta_t=1.5$, $n=5$, η_{e_θ} , $\eta_{c_\theta}=0.1$, $\eta_{e_x}=0.1$, $\eta_{c_x}=0.1$.)

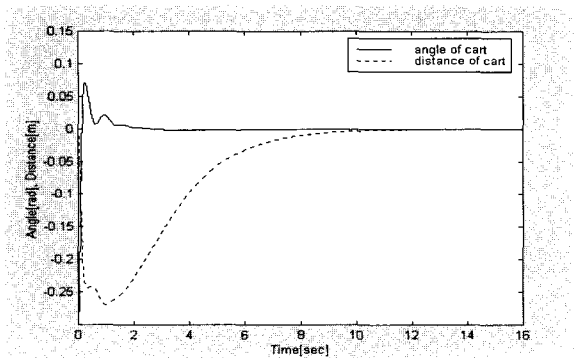


그림 11. 퍼지 학습한 결과 III.
Fig. 11. Fuzzy learning response III.

($\eta_i = 1.5, n = 5, \eta_{e_s} = 1.0, \eta_{c_s} = 0.15, \eta_{e_r} = 0.1, \eta_{c_r} = 0.2$).

이때 학습 횟수는 248번이고, 학습에 소요된 시간은 펜티엄 PC(150MHz)에서 대략 3시간 정도 소요되었다. 그리고 학습 후에 얻은 퍼지 규칙을 표 2에 나타내었다. 위 시뮬레이션 결과에서 알 수 있듯이 초기 응답을 기준으로 응답성이 나쁜 출력의 학습률을 상대적으로 증가시킴으로써 각각의 출력에 대해 만족할 만한 성능을 갖는 퍼지 제어기를 얻을 수 있었다.

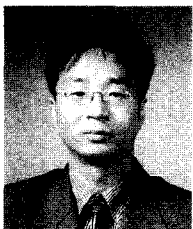
V. 결론

본 논문에서는 수학적 모델링을 구하기 어려운 불안정한 다변수 시스템을 제어하기 위한 퍼지 학습법을 제안하였다. 이 퍼지 학습법을 도입 진자 시스템의 두 변수, 대차의 이동 거리와 진자의 진동각을 동시에 제어하는 제어기 설계에 적용하였다. 먼저, 시뮬레이션을 통해 자코비안 매트릭스의 부호를 살펴보고 불안정한 진자를 안정화시키기 위해 초기 퍼지 규칙을 설계하였다. 그리고 진자의 진동각에 대한 에러와 에러 변화율을 비용함수를 정의하였다. 이 비용함수를 리와 에러 변화율과 대차의 이동거리에 대한 에러와 최소화하기 위

해 기율기법을 이용하여 도출한 결과 식의 학습률을 조정하여 진자의 진동각과 대차의 이동 위치를 동시에 제어하는 좋은 성능의 퍼지 규칙을 작은 학습시간을 통해 설계할 수 있었다.

참고문헌

- [1] B. Chung and Y. Lim, "Hybrid fuzzy learning controller for an unstable nonlinear dystem", *IASTED*, vol. 1, pp. 79-81, 1996.
- [2] B. Chung, "Control of nonlinear multivariable Systems using direct fuzzy learning method", *Journal of Intelligent & Fuzzy Systems*, vol. 5, pp. 297-310, 1998.
- [3] W. Gharieb and G. Nagib, "Fuzzy control to multivariable systems case study : Helicopter model", *Fuzzy-IEE*, vol. 1, pp. 400-405, September, 1996.
- [4] M. A. Goodrich, W. C. Stirling and R. L. Frost, "A satisficing fuzzy logic controller", *Fuzzy-IEE*, vol. 1, pp. 272-276, Sep., 1996.
- [5] S. Lee and R. Kil, "A Gaussian potential function network with hierarchically self-organizing learning", *Neural Network*, vol. 4, pp. 207-224, 1991.
- [6] T. J. Procyk, *A Self-Organizing Controller For Dynamic Processes*, Ph. D. Thesis, Queen Mary College, Univ. of London, 1977.
- [7] B. Chung and J. Oh, "Control of dynamic systems using fuzzy learning algorithm", *Fuzzy Sets and Systems*, vol. 59, no. 1, pp. 1-14, 1993.
- [8] B. Chung and J. Oh, "Autotuning method of membership function in a fuzzy learning control", *Journal of Intelligent & Fuzzy Systems*, vol. 1, no. 4, pp. 335-349, 1993.
- [9] T. Procyk and E. Mandani, "A linguistic self-organizing process controller", *Automatica* vol. 15, no. 1, pp. 15-30, 1979.



임윤규

1969년생. 1994년 영남대 기계공학과 학사. 동대학원 석사(1996), 1996년-현재 동대학원 박사과정, 관심 분야는 인공지능 제어(퍼지 이론, 신경망 이론, 유전자 이론), PID 튜닝, 로봇틱스, 개방형구조의 제어기를 이용한 공장자동화.



정병묵

1960년생. 1983년 부산대학교 기계공학과 학사. 1985년 한국과학기술원 기계공학과 석사. 1994년 한국과학기술원 정밀공학과 박사. 1985년-1995년 LG전자 책임연구원. 1995년-현재 영남대학교 기계공학부 조교수. 관심분야는 인공지능제어, 최적제어, 공장자동화 등.



소범식

1973년생. 1997년 영남대학교 기계공학과 학사. 1997년-현재 동대학원 석사과정, 관심분야는 CAD/CAM, 최적배치, 로봇틱스.