

퍼지로직을 이용한 위치 예측과 매니플레이터의 제어

이 승 환 임 종 태
한국과학기술원 전기및전자공학과

Fuzzy Logic for a Position Prediction and Manipulator Control

Seung-Hwan Lee Jong-Tae Lim
Dept. of Electrical Engineering
KAIST

ABSTRACT

A solution to the problem of robot manipulator tracking of a smoothly moving object is given. It is shown that fuzzy prediction rule, fuzzy control can compensate the adverse effects of noise, time delay, unknown object trajectory, and robot modeling uncertainty. Simulations show that the fuzzy logic control results in acceptable precision.

리들은 다른방법[3]에 비해서 최소화(min), 최대화(max) 연산을 하므로 간단해서 실시간에 처리가 가능하고 기존의 퍼지를 이용한 전문가시스템처럼 정확한 물음 세울 수가 없으므로 이를 보완하기 위해 Mamdani가 제안한 자기구성제어기[4] 이론을 이용하여 학습을 통해서 수정시키도록 했다.

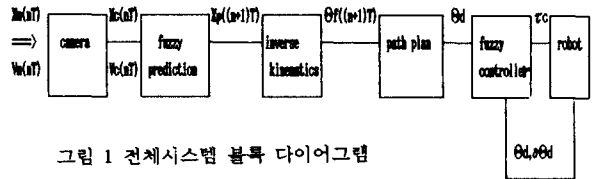


그림 1 전체시스템 블록 다이어그램

Fig 1 Overall System Block Diagram

1. 서 론

현대산업이 급속히 발전하면서 생산성을 높이기 위해 공장자동화에 대한 연구가 활발히 진행되고 있다. 그 중에 한 분야는 로봇의 작업능력 향상을 위해 로봇에 지능을 부여하는 것이다. 로봇트가 완만히 움직이는 물체를 추적하거나 잡는 경우는 하기와 같은 문제가 발생한다. 첫째는 측정된 물체의 위치와 속도에는 noise가 섞여 정확한 값을 얻을 수가 없고, 둘째는 영상처리로 인한 시간지연과 물체의 궤적을 모르므로 물체의 위치를 예측하기가 어렵고, 셋째는 로봇트 매니플레이터를 정확하게 모델링하기가 어려워 효과적인 제어가 어렵다. 이 문제점을 해결하기 위해서 인공지능의 한분야인 퍼지이론[1]을 이용했다. Zadeh교수[2]가 제안한 퍼지 이론은 수리적인 모델을 세우기 힘든 공정, 불확실성이 있는 공정의 제어에 효과적으로 응용되고있다.

2. 본 론

본 논문에서는 첫째 문제점을 측정된 물체의 위치 및 속도를 퍼지화한 후 비퍼지화(defuzzification)하기 전에 α -cut을 함으로 해결하고 둘째 문제점을 예측 룰(prediction rule)을 세워 학습을 통해서 이 룰을 수정시켜서 해결하고 셋째 문제점을 퍼지제어 자체의 강인성(robustness)을 이용하여 해결하고자 한다. 위의 문제를 다루는데 있어서 하기와 같은 가정을 하였다. 물체는 로봇트의 작업 공간에서 smooth하게 이동하고, 움직임이 주기성을 가진다고 했다. 이는 실제공장에서 로봇트가 작업을 할때 예를들면 페인팅을 하거나 납땜을 할 경우 작업할 대상물체는 각각이 비슷한특성을 가짐으로 타당한 가정이다. 이 논문 에서 제안한 퍼지를 이용한 예측 알고

2.1 영상정보의 처리

$X_m(nT)$ 는 물체의 시간 nT 에서 위치이고 $V_m(nT)$ 는 시간 nT 에서 속도이다. 여기서 T 는 영상처리시간과 로봇트매니플레이터의 구동시간을 합친것이다.

$$T = T_v + T_r \quad \text{----- (1)}$$

T_v : 영상처리시간, T_r : 로봇트 매니플레이터 구동시간

카메라가 물체를 찍어서 영상처리 후 나온 값에는 noise가 섞여서 실제 값과 다른 값이 나오므로 카메라에서 나온 값을 $X_c(nT), V_c(nT)$ 라고했다.

$$\begin{aligned} X_c(nT) &= X_m(nT) + G_1(nT) \\ V_c(nT) &= V_m(nT) + G_2(nT) \end{aligned} \quad \text{----- (2)}$$

G_1, G_2 는 gaussian noise를 나타낸다.

이 값을 선형 양자화(linear quantization)를 했는데 퍼지 정의구역(universe of discourse)은 -6 에서 6까지 정수값을 사용하였다. 그리고 이값을 퍼지화 함으로써 어느정도는 noise의 영향을 줄일 수 있었다.

2.2 예측 알고리즘과 학습방법

앞서 가정에서 밝힌 것처럼 물체가 완전히 이동하므로 현재위치와 현재속도를 알면 다음에 물체가 위치할 곳을 대강은 알 수 있다. 예를 들면 현재속도가 크고 현재위치가 중간 정도 이면 다음에 위치할 곳은 중간에서 좀 더 떨어진 위치에 물체가 있을 것이다.

이런 것을 퍼지화하면 아래 표 1 과 같다.

<표 1> 예측 룰

pos\vel	NB	NM	NS	ZR	PS	PM	PB
NB			PS	PS	PS	PM	
NM	NB	NM	NM	ZR	PM	PM	PB
NS	NB	NM	NM	ZR	PM	PM	PB
ZR	NB	NB	NM	ZR	PM	PB	PB
PS	NB	NM	NM	ZR	PM	PM	PB
PM	NB	NM	NM	ZR	PM	PM	PB
PB		NM	NS	NS	NS		

위의 예측 룰에서 나온 출력을 U라고 할 경우 예측 한 위치는 다음과 같이 계산한다.

$$Xp((n+1)T) = Xc(nT) + k1 * U \quad \text{----- (2)}$$

k1은 상수이다. k1 값은 퍼지화할 경우의 정의구역과 물체가 단위시간 여기서는 T동안에 움직일 수 있는 최대거리를 어느정도 안다고 할경우 정의구역의 최대값을 최대거리로 나눈 값을 택하는 것이 제일 좋은 선택법이다.

그러나, 여기서 세운 룰은 전문가에 의해 세워진 것이 아니므로 오차를 포함할 수 밖에 없어서 이것을 학습을 통해서 수정하여 이 물체의 움직임을 예측할 수 있었다. 여기서 자기구성제어기에서 제안한 평가표를 이용하여 룰을 수정한다. 예를 들면 룰에의해 나온출력 값이 실제값 보다 작으면 이때 룰을 증가하도록 membership value의 중심값을 이동 함으로써 학습을 시킬 수 있다. 이때 오차와 오차 변화율은 식(3)에 나타냈고 표2에 평가표를 나타냈다.

$$\begin{aligned} e(nT) &= Xc(nT) - Xp(nT) \\ \dot{e}(nT) &= e(nT) - e((n-1)T) \quad \text{----- (3)} \\ e(nT), \dot{e}(nT) &: nT \text{에서 오차, 오차변화율} \end{aligned}$$

이때 오차와 오차변화율은 비선형 양자화를 하고 퍼지화하였다.

<표 2> 평가표(performance table)

e \ \dot{e}	NB	NM	NS	ZR	PS	PM	PB
NB	NB	NB	NM	NM	NS	NS	ZR
NM	NB	NM	NM	NS	NS	ZR	PS
NS	NM	NM	NS	NS	ZR	PS	PS
ZR	NM	NS	NS	ZR	PS	PS	PM
PS	NS	NS	ZR	PS	PS	PM	PM
PM	NS	ZR	PS	PS	PM	PM	PB
PB	ZR	PS	PS	PM	PM	PB	PB

$$* e = e(nT) \quad , \dot{e} = e(nT) - e((n-1)T)$$

이때 오차변화율은 아래와 같다.

$$\begin{aligned} \dot{e}(nT) &= Xc(nT) - Xp(nT) - [Xc((n-1)T) - Xp((n-1)T)] \\ &= [Xc(nT) - Xc((n-1)T)] - [Xp(nT) - Xp((n-1)T)] \end{aligned}$$

이므로 물체의 위치 증가분과 예측한 증가분의 차이 이므로 실제 변화율과 같으므로 오차보다 가중치를 더 주었다. 그리고 룰 수정방법은 평가표에서 나온값을 P 라고 할 경우 새로운 룰을 New_Rule 이라고 하고 이전 룰을 Old_Rule 이라고 하면 다음과 같다.

$$New_Rule = Old_Rule \oplus P \quad \text{----- (4)}$$

⊕이 연산자는 P만큼 membership value의 중심값을 이동시키는 연산을 뜻한다. 룰 수정시 모든 룰을 변경하지 않고 중심되는 룰만 수정시켰는데 이때 중심적인 룰은 membership value가 1에 해당하는 룰을 말한다. 또 시간이 지남에따라 학습율을 감소시켰다. 이렇게 만든 룰은 max-min composition 을 행했는데 이때 모든 룰을 composition하지 않고 α-cut을 함으로써 잡음의 영향을 줄였다.

비퍼지화는 무게중심법(center of area)을 취했다. 퍼지룰을 실지로 학습 시킬 경우는 몇가지 문제가 있는데 룰 자체가 양자화로 인해 이산적으로 변할 수 밖에 없음으로 시간이 무한대로 감에따라 오차가 0근처의 일정한 한계안에서 수렴하게된다. 아래 그림에 컴퓨터 모의 실험 결과를 보이겠다.

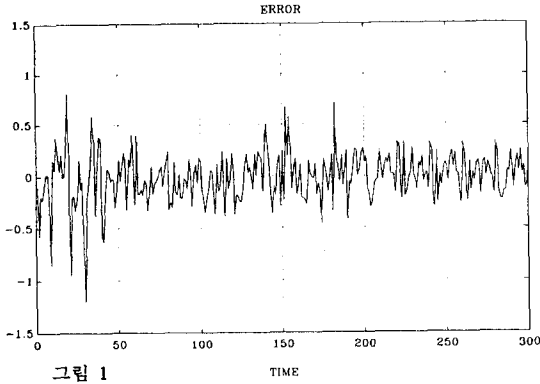


그림 1
fig 1

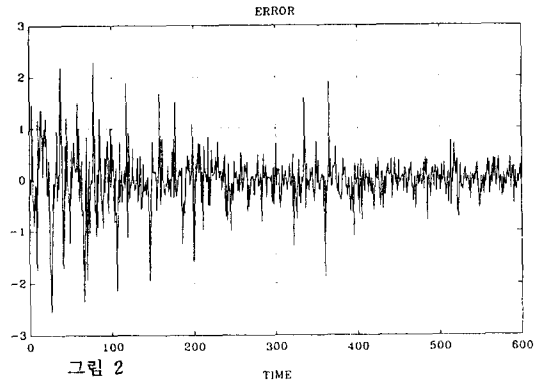


그림 2
fig 2

위의 그림1은 $V_m(nT) = 3.0 * \sin(\pi/10 * (nT))$ 일 경우 예측한 위치와 실제 위치와의 오차를 나타낸 그림이다. 이때 $T_v=0.6\text{sec}$, $T_r=0.4\text{sec}$ 실재 물체의 주기가 20sec이므로 약 10번의 학습후 오차의 절대평균값이 약 0.1 정도로 나타났다. 이 결과는 카메라에서 나온 값은 noise가 섞여 있으므로 물론 잡음의 평균값은 0.0 을 주고 표준편차는 각각 0.2, 0.4 를 주었다. 평균값만 가지고 비교하면 예측 오차는 0.1 정도이다. 그러므로 물체가 움직일 수 있는 위치공간이 약 0에서 20이므로 약 1% 정도의 오차가 발생한다. 그림2는 $V_m(nT)=3.0 * \sin(\pi/10 * (nT)) - 2.0 * \sin(\pi/5 * (nT))$ 일 경우를 나타낸 것이다. 이것은 전의 경우보다 약간복잡한 경우 이므로 오차의 절대 평균 값이 약 0.26 정도 이다. 이 경우는 약 2% 정도의 오차가 발생한다.

2.3 로봇트의 퍼지제어

위의 결과에서 나온 위치를 joint space로 바꾸어 경로 계획 (trajectory path planning)은 $T_r = 0.4\text{sec}$ 이므로 이 시간안에 바라는 joint angle까지 가도록 계획했다. 여기서 나온 joint angle에 도달하도록 퍼지제어기를 설계했다. 이때 로봇트 매니플레이터 동태 방정식(dynamic equation)은 아래와 같다.

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + \tau_d \quad \text{----- (5)}$$

이 식에서 M : mass matrix, V : centriolius force, G : gravity force τ_d : noise torqe
 τ_d 를 넣은 것은 로봇트 동특성의 불확실성을 나타내기위 해서이다.

이때 제어 톨은 표2의 평가표를 그대로 사용했는데 이 이유는 오차와 오차변화율을 식(6)과 같이 정의함으로써 해본 결과 일반적인 퍼지 제어 톨에서 cover 할수 없는 부분이 발생했기 때문이다.

$$\begin{aligned} e(nTc) &= \theta_d(nTc) - \theta(nTc) \\ \dot{e}(nTc) &= \dot{\theta}_d(nTc) - \dot{\theta}(nTc) \quad \text{----- (6)} \end{aligned}$$

이렇게 해서 나온 출력을 $dU(nTc)$ 라고 할 경우 $U(nTc) = U((n-1)Tc) + dU(nTc)$ 이다. 그러므로 전체적인 제어 토오르는 식(7) 과 같다.

$$\tau_c(nTc) = M * (k_2 * U(nTc)) + V(\theta, \dot{\theta}) + G(\theta) \quad \text{----- (7)}$$

T_c 는 제어 샘플링(sampling) 시간으로 T_r 에 맞게 조겐 값이다. 이 제어 톨도 위의 예측 톨을 학습시킨 것과 같이 평가표로 학습을 시켰다.

아래 그림에 퍼지제어와 PD 제어를 비교하여 나타냈다.

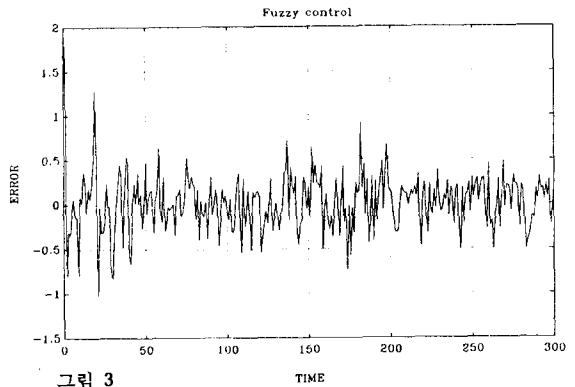


그림 3
fig 3

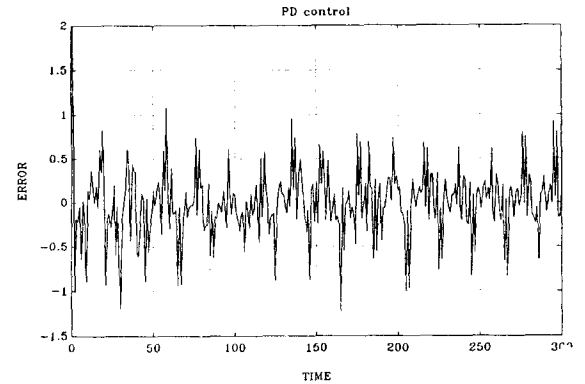


그림 4
fig 4

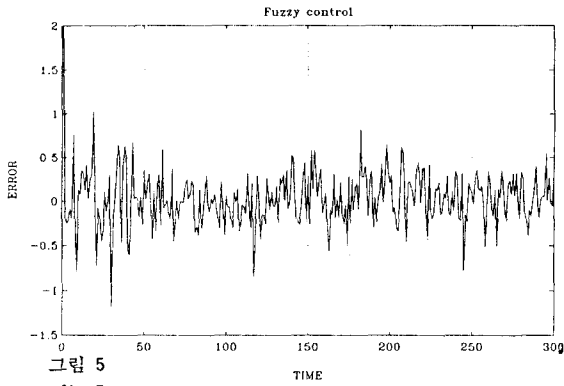


그림 5
fig 5

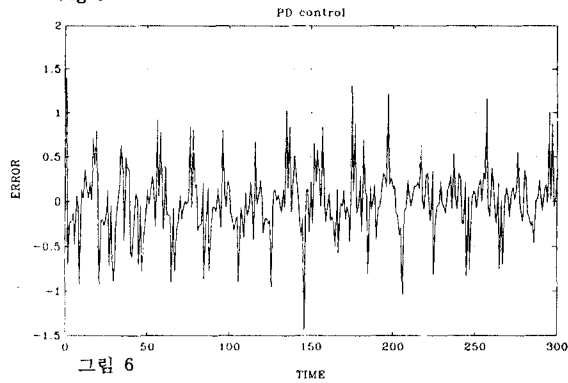


그림 6
fig 6

위의 그림3 과 그림4는 τd 잡음에서 표준편차를 0.1로했을 경우 퍼지제어와 PD제어 했을때 로봇 매니플레이터와 물체의 거리오차를 나타낸 것인데 200초가 지난후 퍼지는 오차의 절대값의 평균이 0.19 이고 PD의 경우는 0.23 이었다. 그림 5과 6는 표준편차가 0.4인 경우로 각각이 0.21 이고 0.28 이었다. 이에서 알 수있듯이 퍼지가 PD의 경우보다 같은 크기의 잡음에 대해서 조금 더 좋은 특성을 보였다. 이때 $T_c=0.02\text{sec}$, $k_2=1.0$ 이다.

3. 결 론

앞에서 제안한 예측 물은 움직이는 물체의 특성이 복잡해도 평가표에 의한 학습을 통해서 잘 예측했다. 로봇트의 퍼지제어는 PD제어 보다 잡음이 심한 경우에도 좋은 특성을 나타내 주었다.

참고문헌(reference)

- [1] G. J. Klir, T. A. Folger, Fuzzy sets, uncertainty, and information, Prentice Hall, 1988.
- [2] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision process", IEEE Trans. SMC-1, 28-44, 1973.
- [3] A. J. Koivo, "Real-Time vision feedback for servoing robotic manipulator with self-tuning controller", IEEE Trans. SMC-21, 134-142, 1991.
- [4] T. J. Procyk, E. H. Mamdani, "A linguistic self-organizing process controller", Automatica, vol-15, 15-30, 1979.