

Fuzzy Inverse Kinematics Mapping을 이용한 로봇의 지능제어에 관한 연구

*김 관 형 , 이 상 배

韓國海洋大學校 電子通信工學科

The study on the Intelligent Control of Robot using Fuzzy Inverse Kinematics Mapping

*Gwan Hyoung Kim , Sang Bae Lee

Dep. of Electro. & Comm. Eng, Korean Maritime Univ

Abstract

Generally, when we control the robot, we should calculate exactly Inverse Kinematics. However, Inverse Kinematics calculation is complex and it takes much time for the manipulator to control in real-time. Therefore, the calculation of Inverse Kinematics can result in significant control delay in real time.

In this paper, we will present that Inverse Kinematics can be calculated through Fuzzy Logic Mapping, based on an exact solution through fuzzy reasoning instead of Inverse Kinematics calculation. Also, the result provides sufficient precision and transient tracking error can be controlled based on a fuzzy adaptive scheme proposed in this paper.

Based on the Denavit-Hartenberg parameters specification , after the Jacobian matrix of arbitrary manipulator is calculated, we will construct Fuzzy Inverse Kinematics Mapping(FIKM) using fuzzy logic and represent a good control efficiency through simulation of 2-DOF manipulator.

I. 서 론

오늘날 robot manipulator의 제어에 큰 문제 중의 하나는 실시간에서 Inverse Kinematics를 계산하는 것이다. Inverse Kinematics 계산은 매우 복잡하고, 일반적으로 robot manipulator의 실시간 제어에 많은 양의 시간이 소요된다. 일반적으로, Inverse Kinematics는 직각좌표와 각 관절 사이는 1:1로 사상되지 않는다.

또한 로봇트가 움직일 때 어떤 특이형상에서 Jacobian은 역행렬을 갖지 않는 경우가 있다. 그러므로 특이형상을 결정하는 것은 여러 가지 이유로 중요하며 특이형상에서 어떤 무한소 운동이 일어날 수 없다. 즉, 어떤 방향으로 manipulator의 end-effector를 움직일 수 없다는 것이다.

그러므로 특이형상에서 Inverse Kinematics의 해는 비 유일성과 관계가 있다.

만약 Manipulator가 특이형상을 통과한다면, 그 특이형상의 한 해에서 다른 해의 형상으로 진행 할 수 없다. 그러므로 로봇에 적용할 때, 특이형상을 피하여 manipulator를 움직이도록 계획하는 것이 중요하다.

이러한 점들로 보아 퍼지이론을 응용한다면 이러한 문제점을 보다 효율적으로 해결할 수 있을 것이다.

II. 퍼지제어 시스템의 모델링

Fig. 1. 의 신호 흐름도에서 Fuzzy Inverse Kinematic Mapping(FIKM)의 입력으로 원하는 robot manipulator의 위치 r_d 와 실제의 end-effector 위치 r 과의 오차인 dr 과 robot manipulator의 Jacobian을 입력으로 한다.

이러한 입력으로부터 Fuzzy Inverse Kinematic Mapping(FIKM)은 robot manipulator를 제어하기 위한 각 관절의 제어각 $d\theta$ 를 생성한다. 이러한 system을 식으로 나타내면 다음과 같이 설명된다.

$$dr = J(\theta) d\theta \quad (\text{Eq.1})$$

$J(\theta)$: Jacobian matrix

dr : differential Cartesian rate

$d\theta$: differential joint rate

(Eq.1)을 (Eq.2)의 형태로 유도할 수 있다.

$$d\theta = J(\theta)^{-1} dr \quad (\text{Eq.2})$$

식(Eq.2)에서 $J(\theta)^{-1}$ 계산이 매우 복잡하고 많은 시간이 소요되기 때문에 이 문제를 Fuzzy logic으로 해결하여 좋은 효과를 제시하고자 한다.

우리는 다시 새로운 변수 $d\theta_{ij}$ 를 J_{ij} 와 dr_i 의 관계로 정의한다.

$$J_{ij} d\theta_{ij} \approx dr_i \quad (\text{Eq.3})$$

J_{ij} : dr 벡터의 i번째 구성되어지는 Jacobian 각각의 요소.

$d\theta_{ij}$: J_{ij} 와 dr_i 의 관계 요소.

dr_i : i번째 dr 벡터.

Fuzzy mapping은 (Eq.4)로 주어질 수 있다.

$$d\theta_{ij} \approx \frac{dr_i}{J_{ij}} \quad (\text{Eq.4})$$

Fuzzy mapping의 $d\theta_{ij}$ 을 알맞게 조절한 $d\theta_j$ 는 end-effector를 원하는 위치로 움직이게 할 관절각(joint angles)을 적절하게 제어할 것이다.

Fig. 2. 에서 fuzzy model을 생성하기 위한 입력 DH parameters는 다음 테이블과 같이 주어지고, 이 논문에서 robot manipulator의 응용 예는 2-DOF를 취할 것이다.

Table. I. DH Parameters for a 2-DOF, Planar Manipulator

i	a_{i-1}	a_i	d_i	θ_i
1	0	0	0	θ_1
2	0	1.0	0	θ_2
3	0	0.5	0	0

III. 퍼지 모델의 생성

$d\theta_{ij}$ 의 fuzzy mapping 생성을 위한 세부 알고리즘은 Fig. 3. 과 같다. Fuzzy mapping에서 입력 변수 dr_r, J_{ij} 의 membership function은 Fig. 4. 와 같이 정의하고 사용한다.

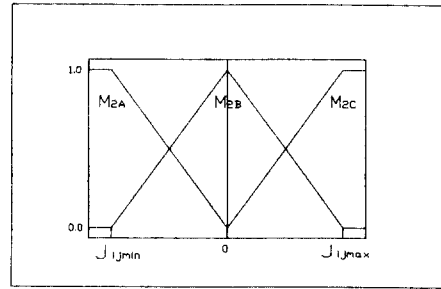


Fig. 4. 입력 변수(dr_r, J_{ij})의 membership function.

퍼지제어 모델의 rule base는 (Eq.5)와 같이 정의한다.

R^0 : If dr_i is M_{1A} and J_{ij} is M_{2A}
then $d\theta_{ij}$ is w_0

R^1 : If dr_i is M_{1A} and J_{ij} is M_{2B}
then $d\theta_{ij}$ is w_1

⋮

R^{N-1} : If dr_i is M_{1C} and J_{ij} is M_{2C}
then $d\theta_{ij}$ is w_{N-1} (Eq.5)

여기에서 R^k 은 k번째 rule이고, N 은 총 rule의 수이다.

IV. 컴퓨터 시뮬레이션

이제 robot manipulator의 경로 추적을 위하여 Fuzzy Inverse Kinematic Mapping(FIKM)을 이용한 제어의

효과를 제시하고자 하며, 동시에 각관절의 각의 변화량도 시간에 따라 변하는 것을 제시하고자 한다.

여기서 초기값 θ 와 원하는 위치를 fuzzy model의 입력으로 주고, 원하는 위치에 도착 할 때까지 계속 θ 를 보증 해 가면서 manipulator를 움직이고, 또 초기값 θ 와 원하는 궤도의 연속된 위치 점을 fuzzy model의 입력으로 주어 여기에서 새로운 위치점 마다 θ 를 보정한다.

컴퓨터 시뮬레이션에서 Table. 1. 과 같은 2-DOF Planar Manipulator 를 실험 모델로 취하였으며 그때의 실험경로 (예제1)과 (예제2)와 같이 취하여 실험을 하였다.

(예제.1) Two-DOF, Planar Manipulator

- ① 경로 : $90^\circ \leq \theta \leq 180^\circ$
 $x = 0.8 \text{ m}$
 $y = 1.0 \sin(\theta - 90^\circ) \text{ m}$
- ② 경로 : $180^\circ < \theta \leq 360^\circ$
 $x = 0.8 \cos(\theta - 90^\circ) \text{ m}$
 $y = 1.0 \text{ m}$

위의 경로로 컴퓨터 시뮬레이션 한 결과는 Fig. 5. 와 같으며 Fig. A. 는 2차원 평면에서 end-effect의 위치의 궤적을 나타내고, Fig. B. 는 각각의 추론 번호에 해당하는 x, y 좌표를 나타낸다. 그리고, Fig. C. 은 각각의 sampling 에서 원하는 위치로 이동하는데 소요 되는 추론의 수를 막대그래프로 나타낸 것이다.

(예제.2) Two-DOF, Planar Manipulator

- 경로 : $0^\circ \leq \theta \leq 180^\circ$
 $x = 0.2 + 1.3 \cos(\theta) \text{ m}$
 $y = 0.8 \sin(\theta) \text{ m}$

위의 경로로 컴퓨터 시뮬레이션 한 결과는 Fig. 6. 과 같으며 Fig. A. 는 2차원 평면에서 end-effect의 위치의 궤적을 나타내고, Fig. B. 는 각각의 추론 번호에 해당하는 x, y 좌표를 나타낸다. 그리고, Fig. C. 은 각각의 sampling 에서 원하는 위치로 이동하는데 소요 되는 추론의 수를 막대그래프로 나타낸 것이다.

V. 결론

위의 결과로 부터 Fuzzy Inverse Kinematics Mapping 을 2-DOF 경로 제어에 응용한 결과 제어 성능이 한층 더 개선 되었음을 보였고, manipulator 가 어떤 위치나 경로를 추종하는 동안 정확한 역기구학방정식의 계산 이 필요 없어지는 것도 보였다. 나아가서 4-DOF, 7-DOF등에 응용하여 보다 좋은 결과를 보일 것이며,

한국해양대 Fuzzy & Neuro Control Laboratory 에서 계속 연구중에 있다.

참고 문헌

- [1] S. Arki, H. Nomura, I. Hayashi and N. Wakami, "A Self-Generating Method of Fuzzy Inference Rules," in *Proceeding: 1991 IFES Conference*, pp. 1047-1058, 1991.
- [2] J.J. Craig, *Introduction to Robotics: Mechanics and Control, 2nd ed.* New York, Addison-Wesley Publishing Company, 1989.
- [3] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Parts I and II," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404-435, 1990.
- [4] W. Boyce and R. DiPrima, *Elementary Differential Equations*, 3rd edition, John Wiley and Sons, New York, 1977
- [5] Benerjee, S. and Woo, P.Y. (1993). Fuzzy Logic Control of a Robot Manipulator, *Proc. 2nd IEEE Conf. Control Applications*, Vancouver, Canada, Vol. 1, pp. 87-89.

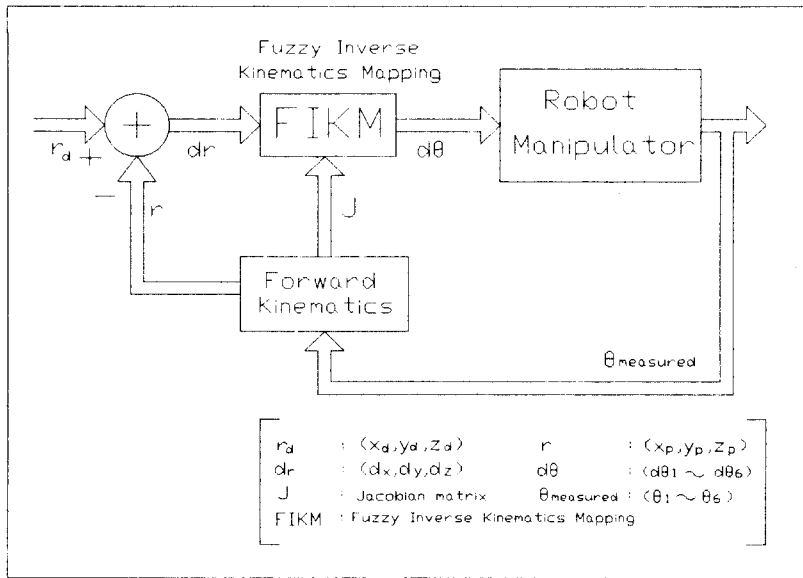


Fig. 1. 퍼지 제어기의 신호 흐름도

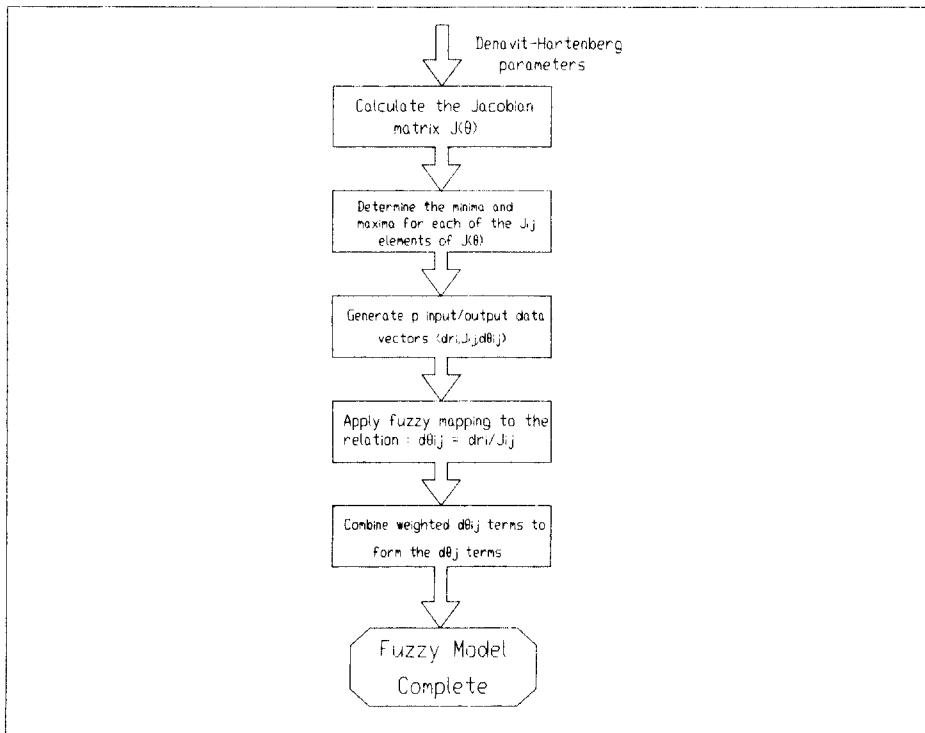


Fig. 2. DH parameters 입력에 대한 퍼지모델의 생성 알고리즘

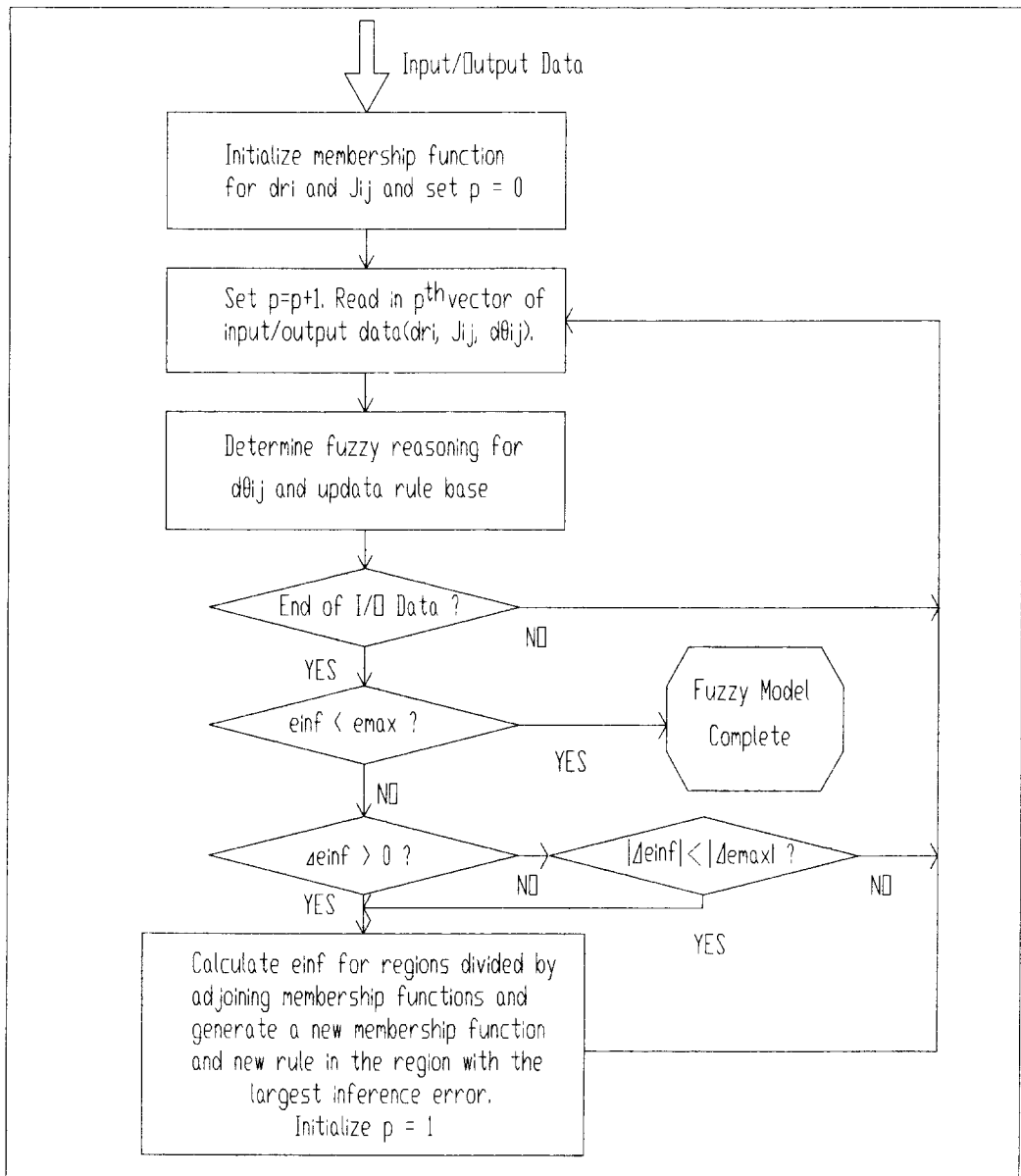


Fig. 3. Fuzzy Mapping에 의한 $d\theta_{ij}$ 생성의 세부 알고리즘

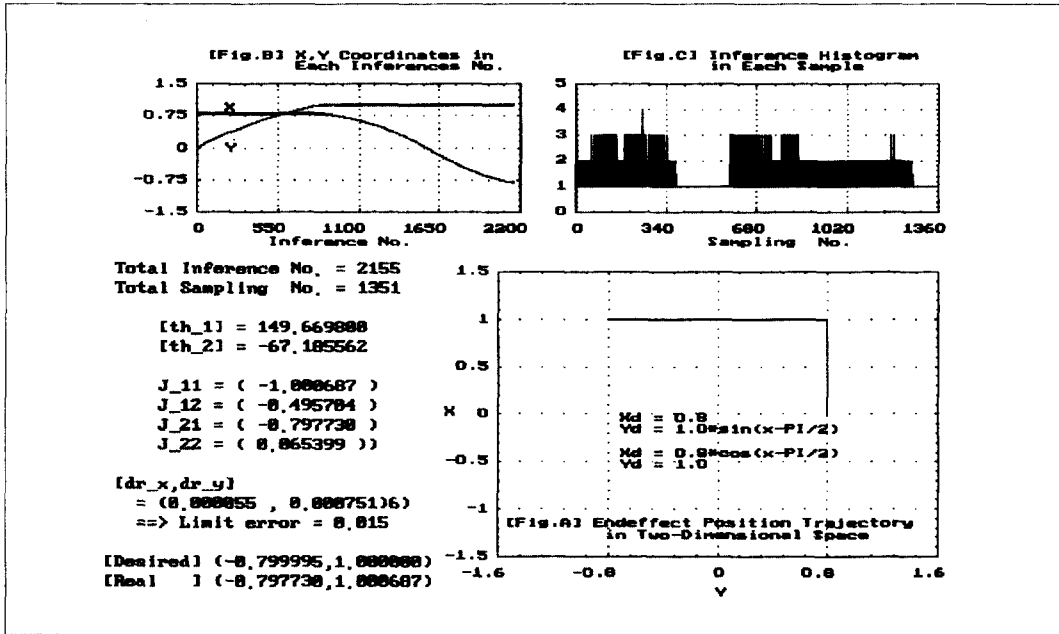


Fig. 5. (예제1)에 대한 퍼지제어기의 실험결과

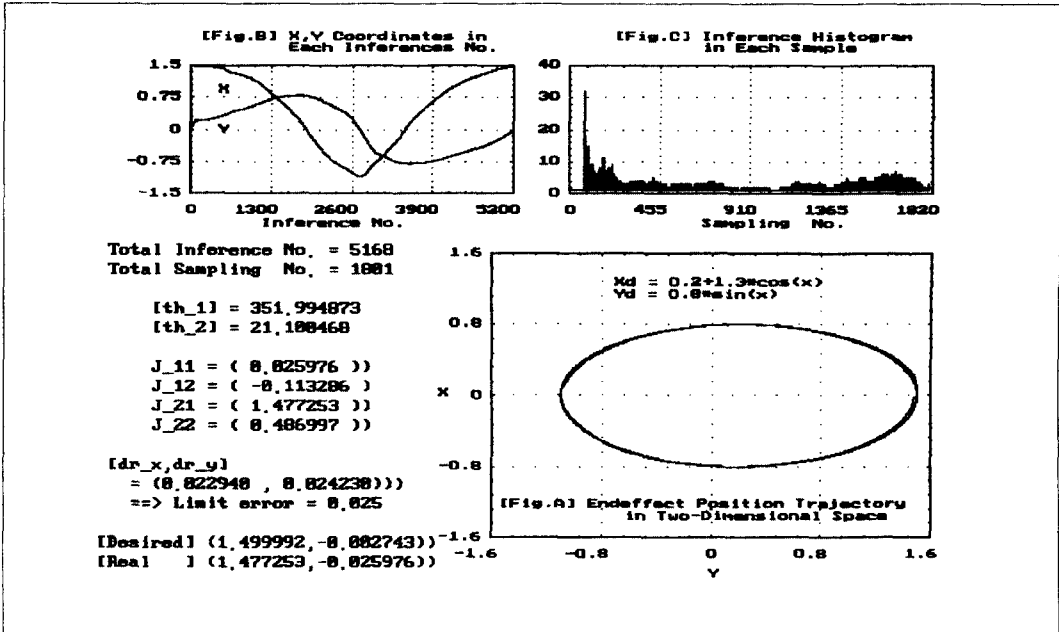


Fig. 6. (예제2)에 대한 퍼지제어기의 실험결과