

論文00-37SC-5-1

이동 로봇의 자율 주행을 위한 적응 퍼지 제어기의 설계

(The Design of Adaptive Fuzzy Controller for Autonomous Navigation of Mobile Robot)

吳俊變*, 崔允浩**, 朴珍培*
(Joon Seop Oh, Yoon Ho Choi, and Jin Bae Park)

요 약

본 논문에서는 이동 로봇의 자율 주행을 위해 퍼지 제어에 기초한 적응 퍼지 제어기 설계 기법을 제안하였다. 제안한 방법에서는 첫째, 퍼지 추론 과정에서 불필요한 규칙으로 인하여 데이터 처리 시간이 증가된다는 단점을 이동 로봇의 현재 상태에 적합한 퍼지 추론 규칙과 소속함수를 생성함으로써 향상시켰다. 이 과정은 데이터 쌍을 이용한 클러스터링 방법을 통하여 구현되며, 이동 로봇은 보다 적은 퍼지 추론 규칙을 가지고 빠른 데이터 처리 속도로 주행 가능해진다. 둘째, 기존의 퍼지 제어기가 입·출력 변수의 고정된 소속 함수로 인하여 느린 수렴성을 갖는 단점을 클러스터링 과정에서 생성된 소속함수를 스케일링함으로써 향상시켰다. 한편 본 논문에서 제안한 제어 방법의 성능 평가를 위해 퍼지 제어를 사용한 기존의 제어 방법과 컴퓨터 모의 실험 및 실제 이동 로봇 주행 실험을 통해 비교 및 고찰하였다.

Abstract

In this paper we propose a design method of the adaptive fuzzy controller for autonomous navigation of mobile robots based on the fuzzy theory. We present two improvements. First, unnecessary rules in the fuzzy inference process make data processing time increase. We reduce this data processing time by generating suitable fuzzy inference rules and membership functions according to the current state of a mobile robot. It is implemented with the clustering method using input and output data pairs, and then it is possible for a mobile robot to navigate in shorter processing time with less fuzzy inference rules. Second, existing algorithms used fixed membership functions of input and output variables, hence converged slowly. We improve convergence time via scaling membership functions generated by the clustering method. To evaluate and compare the performance of the proposed method with the existing fuzzy navigation controller, computer simulations and navigation experiments of a mobile robot are presented.

I. 서론

20세기 후반에 들어 매니플레이터의 지역성 극복과

* 正會員, 延世大學校 電氣·컴퓨터工學科

(Dept. of Electrical and Computer Engineering
Yonsei University)

** 正會員, 京畿大學校 電子·機械工學部

(Dept. of Electronic and Mechanical Engineering
Kyonggi University)

接受日字:1999年10月7日, 수정완료일:2000年8月1日

고임금으로 인한 원가 상승 및 인력난 등의 문제를 해소하기 위한 공장 자동화의 개념이 도입되면서 임의의 작업 공간에서 원하는 위치로 이동이 가능한 이동 로봇의 연구가 산업계 여러 분야에서 활발히 연구되고 있다.

이동 로봇의 연구 내용으로는 가장 기본적인 로봇 자체의 하드웨어 설계나 전동기 제어, 주위 환경 인식을 위한 여러 가지 센서의 융합 문제^[1], 목적 지까지의 경로 계획 문제, 진행 중 나타나는 장애물 회피 문제^{[2][3]} 그리고 이동 로봇의 작업 영역 내의 지도를 작성하는 문제^[4] 등이 있으며, 현재 연구 동향

을 보면 이동 로봇의 제어기 설계 및 하드웨어 설계보다는 장애물 회피나 주위 환경 인식을 통한 지도 작성 및 경로 계획 등과 같은 응용 분야에 초점을 두고 있다. 한편 이동 로봇이 미지의 환경에서도 정확한 경로 추종 및 제어 목적에 빠른 수렴성을 갖도록 하기 위한 제어기의 설계 방법도 연구 대상이 될 수 있다.

최근에는 기존의 수학적인 모델로 만족스러운 결과를 얻지 못하는 복잡한 시스템이나 제어 대상의 파라미터가 시간에 따라 변할 때 기존의 방법을 적용하기 어려운 시스템 등을 제어하기 위한 연구의 일환으로 지능 제어가 활발히 연구되고 있다. 퍼지 이론을 이용한 퍼지 제어, 신경 회로망을 이용한 제어 그리고 퍼지 이론과 신경 회로망을 이용한 퍼지-신경망 제어 등이 연구되고 있다^{[5]-[7]}. 그 중에서도 복잡한 수학적 모델링 과정을 거치지 않고도 기존의 제어 기법으로는 다루기 힘든 잘 정의되지 않는 시스템의 제어와 실시간 제어에 적합한 퍼지 제어가 활발히 진행되고 있다. 그러나 퍼지 제어는 IF-THEN 형식의 논리형 제어 기법으로서 인간의 정성적인 지식을 표현할 수 있는 제어 기법이지만 제어 대상이 제어 목적대로 동작하도록 제어 대상과 환경에 적합한 입·출력 변수의 소속 함수의 파라미터와 퍼지 추론 규칙이 정확히 표현되어야 한다는 제약 조건이 있다. 따라서 최근 여러 분야에서 퍼지 이론이 실용화되면서 소속 함수의 파라미터나 퍼지 추론 규칙을 자동으로 동조하는 방법이 제안되고 있다. Sugeno는 전진부는 심플렉스 방법으로, 후진부는 선형식으로 하여 비선형 프로그램 기법을 사용한 동조 방법을 제안하였으며^[8], Maeda는 계산적인 순서 구조를 사용하여 기존의 방법보다 처리가 빠른 동조 기법을 제안하였다^[9]. 또한 Hong은 학습 데이터의 집합을 클러스터링을 이용하여 자동으로 소속 함수와 퍼지 추론 규칙을 유도하는 방법을 제안하였다^[10].

본 논문은 이동 로봇의 자율 주행 제어를 위해서 지능 제어의 관점에서 퍼지 이론에 기초한 적응 퍼지 제어 기법을 제안한다. 기존의 퍼지 제어기는 고정된 퍼지 추론 규칙에서 불필요한 규칙으로 인하여 데이터 처리 시간이 길어지고 또한 고정된 입·출력 변수의 소속 함수로 인하여 제어 목적에의 수렴이 느리다는 단점을 가지는데, 본 논문에서 제안한 방법은 이동 로봇의 자율 주행 제어를 위해 기존의 고정된 소속 함수와 퍼지 추론 규칙에서 유도한 전체 데이터 쌍을 기

반으로 이동 로봇이 학습에 필요한 주행 데이터 쌍을 on-line으로 이동 로봇의 현재 상태에 적합한 데이터 쌍들을 습득하고, Hong이 제안한 클러스터링(clustering) 방법을 통하여 이동 로봇의 주행 제어 목적에 적합한 새로운 소속 함수와 퍼지 추론 규칙을 생성 보다 적은 퍼지 추론 규칙을 이용하여 빠른 처리 속도로 주행을 가능케 하며 생성된 소속 함수를 스케일링(scaling) 함으로써 제어 목표에 빠른 수렴을 하도록 한다.

본 논문의 주행 실험에서 주변 환경 인식을 위한 정보 습득은 다른 센서에 비하여 데이터 처리 시간이 짧으며 가격이 저렴한 초음파 센서를 이용하고, 본 논문에서 제안한 주행 알고리즘은 벽면 추종 주행(wall-following navigation)을 기본으로 한다. 그리고 본 논문에서 제안한 방법에 의해 설계된 적응 퍼지 제어기의 성능을 평가하기 위해, 기존의 퍼지 제어를 사용한 기존 제어 방법과 모의 실험 및 실제 주행 실험을 통해 비교한다.

II. 적응 퍼지 제어기 설계

본 장에서는 기존의 퍼지 제어기의 고정된 입·출력 변수의 소속 함수로 인한 제어 목적으로의 비효율적인 수렴 속도 및 고정된 퍼지 추론 규칙에서 불필요한 퍼지 추론 규칙으로 인한 데이터 처리 속도의 비효율성을 향상시키기 위하여 이동 로봇의 주행 환경에 따라 환경에 적합한 입·출력 변수의 소속 함수와 퍼지 추론 규칙을 자동 동조하는 방법을 제안한다. 데이터 처리 속도의 향상을 위해서는 Hong이 제안한 학습 데이터의 쌍을 클러스터링 기법^[10]을 이용하여 자동으로 소속 함수와 퍼지 추론 규칙을 유도하는 방법을 사용하며, 제어 목적으로의 빠른 수렴성을 위해서는 클러스터링 기법에 의해 생성된 입력 변수의 소속 함수 매개변수를 스케일링함으로써 주행 환경에 적합한 새로운 입력 변수의 소속 함수를 유도하는 방법을 사용한다. 이동 로봇의 주행 중 추가되는 입·출력 데이터 쌍들은 on-line 상태에서 매 시간 멤버십 함수의 튜닝 방법에 사용되며, 튜닝된 멤버십 함수의 파라미터가 다시 스케일링 방법에 의해 조정된다. 즉, 매 시간 멤버십 함수의 파라미터들이 이동 로봇의 상태에 적합하도록 튜닝되는 것이다.

1. 클러스터링을 이용한 퍼지 추론 규칙 및 소속 함수의 유도

본 논문에서 제안한 클러스터링 방법은 크게 3단계로 나뉠 수 있다. 첫 단계에서는, 기존의 퍼지 제어기의 입·출력 변수의 소속 함수와 퍼지 추론 규칙 테이블을 이용하여 전체 25가지의 학습 데이터 쌍 중에서 이동 로봇의 현재 상태와 관련된 데이터 쌍을 선택한다. 두 번째 단계에서는 얻어진 학습 데이터 쌍을 이용하여 입·출력 변수의 소속 함수를 생성하며, 마지막 단계에서는 퍼지 추론 규칙을 생성한다.

(1) 학습 데이터의 획득

클러스터링 방법에 사용되는 전체적인 학습 데이터 쌍은 기존의 퍼지 제어기의 입·출력 변수의 소속 함수와 퍼지 추론 규칙 테이블에서 얻을 수 있다.

표 1. 언어적인 변수를 일반적인 수치 값으로 사상

Table 1. Mapping of linguistic variable to crisp value.

		진 행 방 향 (θ)				
		-60°	-30°	0°	30°	60°
거리차 (dif)	-60cm	60°	60°	30°	0°	-30°
	-300cm	60°	30°	30°	0°	-30°
	0cm	60°	30°	0°	-30°	-60°
	300cm	30°	0°	-30°	-30°	-60°
	600cm	30°	0°	-30°	-60°	-60°

기존의 퍼지 제어기의 퍼지 추론 규칙 테이블에서 각각의 언어 변수를 수치적인 값으로 사상하면 표 2.1과 같고 이로부터 전체 데이터 쌍 25개를 구할 수 있다. 두 개의 입력 기준선과의 거리차와 진행 방향이 주어지면 각각을 기존의 퍼지 제어기의 입력 변수에 대한 소속 함수들의 영역에 사상하여 입력 값들을 포함하는 소속 함수들의 중심값을 구한 후 전체 데이터 쌍에서 구해진 중심값과 같은 데이터 쌍들을 선택한다.

(2) 클러스터링을 이용한 입·출력 변수의 소속 함수와 퍼지 추론 규칙의 동조

본 논문에서 제안한 학습 과정은 주요한 5단계로 나눌 수 있다. 첫 번째 단계에서는, 학습 데이터 쌍들에서 출력 변수의 데이터 값들을 집단화하고, 집단화된 값들에 의하여 출력 변수의 소속 함수를 생성해 낸다. 두 번째 단계에서는 입력 변수의 특성에 대한 초

기 소속 함수를 생성한다. 세 번째 단계에서는 초기 퍼지 추론 규칙 테이블을 구성한다. 네 번째 단계에서는 초기 퍼지 추론 규칙 테이블을 간략화한다. 마지막 단계에서는 간략화된 퍼지 추론 규칙 테이블을 이용하여 입력 변수들의 새로운 소속 함수와 퍼지 추론 규칙을 유도한다. 각 단계를 구체적으로 서술하면 다음과 같다.

① 출력 데이터의 그룹화 및 퍼지화

이 단계에서는 모든 학습 데이터 쌍들의 출력 값이 클러스터링 과정을 통하여 적당한 그룹으로 되며, 출력 값에 대한 소속 함수들이 이에 적합하게 유도된다. 클러스터링 과정은 α 값 이상의 집단에 대한 소속 정도를 갖는 데이터 쌍들의 출력 값들이 같은 집단에 속하게 되고, 그 집단에 적합한 출력 값들과 각 출력 값의 소속 정도에 따라 소속 함수가 유도된다. 5 단계의 하위 단계가 여기에 포함된다. 첫 번째로 오름차순으로 학습 데이터의 출력 값을 정렬한다. 이는 인접한 데이터 사이의 관계를 구하기 위하여 학습 데이터의 출력 값을 정렬하는 것이다. 정렬 후에 변경된 차원은 다음과 같다.

$$y_1, y_2, \dots, y_n \tag{1}$$

여기서 $y_i \leq y_{i+1}$ (for $i=1, \dots, n-1$)이다. 두 번째로 인접한 데이터 사이의 차를 구한다. 인접한 데이터 사이의 차이는 그들 사이의 유사성에 대한 정보를 준다. 각각의 쌍 y_i 와 y_{i+1} ($i=1, 2, \dots, n-1$)에 대한 차이는 $diff_i = y_{i+1} - y_i$ 이다. 세 번째로 인접한 데이터 사이의 유사성을 구한다. 인접한 데이터 사이의 유사성의 값을 획득하기 위하여 식 (2)를 이용하여 차이 $diff_i$ 를 0과 1사이의 실수 값 s_i 로 변환한다.

$$s_i = \begin{cases} 1 - \frac{diff_i}{C * \sigma_s} & \text{for } diff_i \leq \sigma_s \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

여기서 s_i 는 y_i 와 y_{i+1} 사이의 유사성을 나타내고, σ_s 는 $diff_i$ 의 표준 편차이며 C 는 유사성의 소속 함수의 모양을 결정하는 제어 매개 변수이고, C 값이 큰 경우에는 강한 유사성을 나타낸다. 네 번째로 유사성에 따른 학습 표본을 클러스터링한다. 여기서는 표본을 집단화하기 위하여 유사성의 α -cut을 사용한다. α 값은 같은 집단에 속한다고 생각되어지는 두 개의

인접한 데이터에 대한 문턱값으로 결정된다. 큰 α 는 작은 집단의 수를 결정짓는다. 만약 $s_i \leq \alpha$ 이면 두 개의 이웃한 데이터는 다른 집단으로 나뉘고, $s_i > \alpha$ 이면 같은 집단으로 결정된다. 집단화 결과로 (y_i, R_j) 의 형식의 결과를 얻을 수 있다. 이것은 i 번째 출력 데이터는 R_j 내로 집단화되어진다는 의미이고, 여기서 R_j 는 j 번째 생성된 퍼지 영역을 의미한다. 마지막 다섯 번째로 출력 공간의 소속 함수를 결정한다. 여기서는 간략하게 각각의 언어적인 변수들에 대하여 부등변 삼각형 모양의 소속 함수를 사용한다.

본 논문에서 세 개의 매개 변수를 다음 과정을 통해 결정한다. 처음 중심값 b 를 집단의 중심에 위치하게 놓는다. 다음으로 집단 내 두 개의 학습 출력 경계선(training outputs boundary)의 소속 정도를 구한다. 여기서, 경계선은 집단의 최소와 최대 출력을 의미한다. 출력 소속 함수의 두 개의 끝점인 a 와 c 는 b 의 외삽법과 두 개의 출력 데이터의 경계선을 통하여 구할 수 있다. 이 과정은 다시 4단계로 나뉜다. 첫 번째로 중심 값 b_j 를 찾는다. 만약 y_i, y_{i+1}, \dots, y_k 가 j 집단에 속하면 집단의 중심 값은 식 (3)에 의해 정의된다.

$$b_j = \frac{y_i * s_i + y_{i+1} * \frac{s_i + s_{i+1}}{2} + y_{i+2} * \frac{s_{i+1} + s_{i+2}}{2} + \dots}{s_i + \frac{s_i + s_{i+1}}{2} + \frac{s_{i+1} + s_{i+2}}{2} + \dots} \quad (3)$$

$$\dots + \frac{y_{k-1} * \frac{s_{k-2} + s_{k-1}}{2} + y_k * s_{k-1}}{\dots + \frac{s_{k-2} + s_{k-1}}{2} + s_{k-1}}$$

두 번째로 y_i 와 y_k 의 소속 정도를 결정한다. 집단에서 최소의 유사성은 두 개의 경계점 y_i 와 y_k 의 소속 정도로서 선택된다. 식 (4)는 $\mu_j(y_i)$ 와 $\mu_j(y_k)$ 을 계산하기 위하여 사용된다.

$$\mu_j(y_i) = \mu_j(y_k) = \min(s_i, s_{i+1}, \dots, s_{k-1}) \quad (4)$$

여기서 μ_j 는 j 번째 집단에 속하는 소속 정도를 나타낸다. 이 방법은 두 개의 경계 데이터들이 같은 집단에서의 다른 데이터들보다 낮은 소속 정도를 갖고 집단화된다는 의미가 있다. 세 번째로 정점 a 를 결정한다. 두 개의 점 $(b_j, 1)$ 과 $(y_i, \mu_j(y_i))$ 에 따라 식 (5)에 의하여 점 $(a, 0)$ 를 구한다.

$$a = b_j - \frac{b_j - y_i}{1 - \mu_j(y_i)} \quad (5)$$

마지막 네 번째로 정점 c 를 결정한다. 두 개의 점 $(b_j, 1)$ 과 $(y_k, \mu_j(y_k))$ 에 따라 식 (6)에 의하여 점 $(c, 0)$ 를 구한다.

$$c = b_j + \frac{y_k - b_j}{1 - \mu_j(y_k)} \quad (6)$$

② 입력 변수에 대한 초기 소속 함수 구성

이 단계에서는 입력 변수들에 대한 초기 소속 함수를 만든다. 각각의 입력 변수들에 대하여 초기 소속 함수를 할당하며, 소속 함수의 매개 변수 값 $b-a=c-b$ 가 최소 단위로 정의되는 부등변 삼각형 (a, b, c) 으로 결정된다. 여기서 각각의 매개 변수의 최소 단위를 a_0 , 그리고 매개 변수 값의 최대를 a_n 이라 놓는다. 입력 변수에 대한 초기 소속 함수는 그림 2.1과 같이 정의할 수 있다. 여기서 $a_i - a_{i-1} = a_{i+1} - a_i$ 가 최소 단위로 정의되고, R_x 는 x 번째 초기 영역 ($i=2, 3, \dots, n-1; x=1, 2, \dots, n$)을 의미한다. 이 정의는 많은 초기 소속 함수가 존재하므로 전체 범위 위에서 부분적인 값들에 대해서는 부적절하게 보이나 실제적인 설계에서는 단지 학습 데이터의 입력 값들과 관련 있는 소속 함수들이 고려되고 다른 소속 함수들은 그들에 속하게 되므로 소속 함수의 수는 학습 데이터 값들에 따라 감소하게 된다. 고려되지 않는 소속 함수는 다음 단계에서 합병된다.

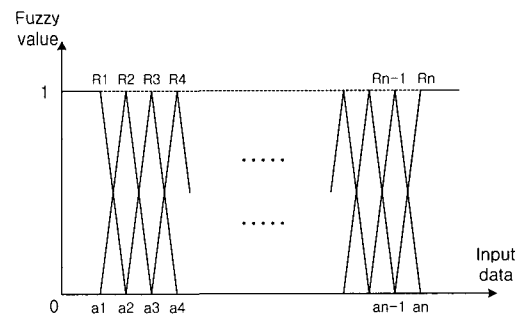


그림 1. 초기 소속 함수
Fig. 1. Initial membership function.

③ 초기 추론 규칙 테이블의 구성

이 단계는 초기 소속 함수에 따른 다차원의 추론 규칙 테이블 구성하는 것이다. 하나의 셀(cell)은 추론

규칙 테이블에서 입력 영역에 따른 출력 데이터들의 퍼지 관계로 정의된다. 셀 $(d_1, d_2, \dots, d_i, \dots, d_m)$ 은 추론 규칙 테이블에서 위치 $(d_1, d_2, \dots, d_i, \dots, d_m)$ 의 내용으로 표현된다. 여기서 m 은 추론 규칙 테이블에서의 입력 영역 차원을 그리고 d_i 는 i 번째 차원에서 위치 값이다. 추론 규칙 테이블에서 각각의 셀들은 비어 있을 수도 있고, 출력의 퍼지 관계를 포함할 수도 있다. 실제적인 설계에서는 단지 비어 있지 않는 셀들만을 취한다. 나머지 비어 있는 셀들은 주위의 비어 있지 않는 셀들에 합병된다.

④ 초기 추론 규칙 테이블의 간략화

이 단계는 필요 없는 즉, 비어 있는 셀들을 비어 있지 않는 셀들로 합병시킴으로써 초기 추론 규칙 테이블을 간략화하는 것이다. 이 과정은 다음의 다섯 단계로 나뉘며 그림 2에서 나타난다. 첫 번째로, 만약 그림 2(a)와 같이 두 개의 인접한 열들 또는 행들에서의 셀들이 같으면 이 두 개의 열 또는 행을 하나로 합병한다. 두 번째로, 만약 두 개의 열이나 행에서 그들 어느 한 쪽이 비어 있고, 적어도 두 개의 행이나 열에서 하나의 셀이 비어 있지 않으면, 이 두 개의 열이나 행들은 하나로 합병한다. 이 과정은 그림 2(b)에서 보여준다. 세 번째로, 만약 그림 2(c)와 같이 열 또는 행에서 모든 셀들이 비어 있고, 그것의 인접한 두 개의 열 또는 행에서 셀들이 같으면 이 세 개의 열 또는 행들을 하나로 합병한다. 네 번째로, 만약 열 또는 행의 모든 셀들이 비어 있고, 그것의 인접한 두 개의 열 또는 행들이 같거나 그들 중에 하나가 비어 있으면 세 개의 열 또는 행들을 합병한다. 이 과정은 그림 2(d)와 같다. 다섯 번째로, 만약 열 또는 행에서 모든 셀들이 비어 있고, 그것의 왼쪽으로의 열 또는 행의 비어 있지 않은 모든 셀들이 같은 영역을 갖으며, 그것의 오른쪽으로의 열 또는 행의 비어 있지 않은 모든 셀들이 같은 영역을 갖는다면 이 세 개의 열 또는 행을 두 개의 영역으로 합병한다. 이 과정은 그림 2(e)와 같다.

⑤ 간략화 과정을 통한 새로운 소속 함수의 형성.

여기서는 전 단계의 합병 과정을 거친 추론 규칙 테이블을 이용하여 입력 변수들에 대한 소속 함수를 다시 세운다. 과정 1과 과정 2에 대하여 만약 (a_i, b_i, c_i) 와 (a_j, b_j, c_j) 가 d_i 와 d_{i+1} 가 되는 소속함수라면, 새로운 소속 함수는 $(a_i, (b_i + b_j)/2, c_i)$ 가 된다. 과정 3과 과정 4에 대하여 만약 (a_i, b_i, c_i) ,

(a_j, b_j, c_j) 및 (a_k, b_k, c_k) 가 d_{i-1}, d_i 와 d_{i+1} 가 되는 소속 함수라면, 새로운 소속 함수는 $(a_i, (b_i + b_j + b_k)/3, c_i)$ 가 된다. 과정 5에 대하여 만약 (a_i, b_i, c_i) , (a_j, b_j, c_j) 및 (a_k, b_k, c_k) 가 d_{i-1}, d_i 와 d_{i+1} 가 되는 소속 함수라면, 새로운 소속 함수는 (a_i, b_i, c_i) 와 (a_j, b_k, c_k) 가 된다.

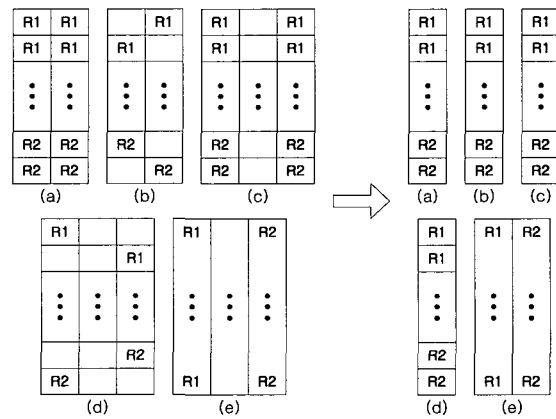


그림 2 추론 규칙 테이블의 간략화
Fig. 2. Simplify of inference rule table.

(3) 퍼지 추론 규칙의 유도

지금까지는 이동 로봇의 현재 상태 즉, 기준선과의 거리차(dif)와 이동 로봇의 진행 방향(θ)에 적합한 데이터 쌍들을 획득하여 이 데이터 쌍으로부터 입·출력 변수의 소속 함수를 유도하였다. 이 단계에서는 유도된 입·출력 변수와 추론 규칙 테이블을 이용하여 퍼지 추론 규칙들을 생성한다. 추론 규칙 테이블에서 각각의 셀 $(Cell_{(d_1, d_2, \dots, d_m)} = R_i)$ 은 식 (7)과 같은 규칙을 유도하는데 사용된다.

$$\text{If } input_1 = d_1, input_2 = d_2, \dots, \text{ and } input_m = d_m$$

$$\text{Then } output = R_i \tag{7}$$

2. 수렴 속도 향상을 위한 스케일링 방법

본 논문에서는 이동 로봇이 기준선까지 빠른 속도로 수렴 할 수 있는 방법의 하나로 퍼지 제어기 소속 함수의 매개 변수를 스케일하는 방법을 사용한다. 이는 고정된 퍼지 제어 시스템의 경우에는 입력 영역의 전체 범위가 넓어서 이동 로봇이 기준선에 가까이 있을 경우에는 조향각($\Delta\theta$)이 작은 값으로 변하기 때

문에 기준선까지 느린 수렴을 초래한다. 따라서 이동 로봇의 현재 상태인 기준선과의 거리차(dif)에 따라 적응 퍼지 제어 시스템의 입력 변수 거리차(dif)의 소속 함수 매개 변수를 스케일함으로써 이동 로봇이 기준선에 가까이 있어도 큰 조향각($\Delta\theta$)을 갖고 진행 방향을 변경하므로 빠른 수렴성을 갖게 된다.

스케일 척도를 구하는 것은 또 하나의 퍼지 제어 시스템으로서 생각할 수 있다. 입력 변수로 이동 로봇의 현재 상태인 기준선과의 거리차(dif)를 입력 받아 출력으로 스케일 척도를 구하는 것이다. 입·출력 변수의 소속 함수는 그림 3과 같으며 소속 함수의 매개 변수는 경험적인 방법에 의해 결정한다.

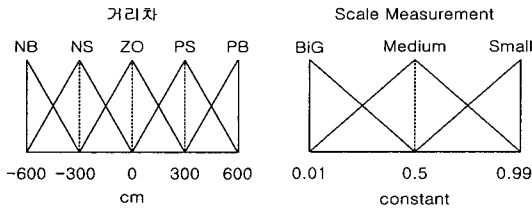


그림 3. 스케일링을 위한 입·출력 변수의 소속 함수
Fig. 3. Membership function of input and output variable for scaling.

또한, 다음과 같은 퍼지 추론 규칙을 실험을 통하여 경험적으로 유도할 수 있다.

$$\begin{aligned}
 & \text{If (dif) is NB Then scale measurement is Small} \rightarrow \text{Rule}_1 \\
 & \text{If (dif) is NS Then scale measurement is Medium} \rightarrow \text{Rule}_2 \\
 & \text{If (dif) is ZE Then scale measurement is Big} \rightarrow \text{Rule}_3 \\
 & \text{If (dif) is PS Then scale measurement is Medium} \rightarrow \text{Rule}_4 \\
 & \text{If (dif) is PB Then scale measurement is Small} \rightarrow \text{Rule}_5
 \end{aligned}
 \tag{8}$$

3. 퍼지 추론 과정

퍼지 추론 과정에서 입력 데이터는 이동 로봇의 현재 상태에서 얻어지며, 추론은 유도된 퍼지 추론 규칙들과 입·출력 변수의 소속 함수들을 통하여 이루어진다. 본 논문에서 제안한 스케일링 기법은 유도된 지식의 형식이 공통적으로 사용되는 퍼지 지식으로 일관되므로 추론 과정에서 일반적인 Mamdani의 추론법이 적용 가능하다. 추론 과정은 크게 3가지 단계로 나눌 수 있다. 첫째, 주어진 소속 함수에 따라서 수치적인 입력 값들을 언어적인 항들로 변환한다. 수치적인

입력 (I_1, I_2, \dots, I_m)을 그들이 관련된 소속 정도를 가진 퍼지 언어 항들로 사상한다. 둘째, 출력 집단들을 찾기 위해서 퍼지 추론 규칙들에 언어적인 항들을 사상시킨다. 셋째로 출력 집단을 비퍼지화한다. 최종 출력값은 출력 집단들의 평균으로 획득된다. 출력 변수에 대한 소속 함수는 (a_i, b_i, c_i) 로 표현된다. 출력값은 식 (9)에 의해 계산된다.

$$y = \frac{\sum_{i=1}^K \mu_{R_i}(I) * b_i}{\sum_{i=1}^K \mu_{R_i}(I)}
 \tag{9}$$

여기서 K 는 가능한 출력 집단의 수이고, $R_i(I)$ 는 입력 값을 퍼지 언어 항들로 사상할 때의 소속 정도를 나타낸다.

III. 이동 로봇의 적응 퍼지 제어 시스템

본 논문에서는 이동 로봇의 벽면 추종 주행시 데이터 처리 속도와 수렴성의 향상을 위하여 2장에서 서술한 클러스터링 방법과 스케일링 방법을 사용한다. 전체 시스템은 크게 2개의 독립적인 퍼지 제어기로 구성된다. 하나는 클러스터링 방법에 의하여 이동 로봇의 현재 상태에 적합하게 유도된 퍼지 추론 규칙과 입·출력 변수의 소속 함수를 이용하여 이동 로봇가 다음 상태로 진행하게 할 조향각($\Delta\theta$)을 구해내는 퍼지 제어기로 두 개의 입력과 하나의 출력을 갖는 MISO(Multi Input Single Output)시스템이고, 다른 하나는 빠른 수렴성을 갖기 위해 입력 영역을 스케일하는, 즉 스케일 척도를 구하는 퍼지 제어기로서 SISO(Single Input Single Output)시스템이다. 본 논문에서 제안한 이동 로봇의 적응 제어 시스템의 블록 선도는 그림 4와 같으며 크게 3가지 부분으로 나뉜다. 그림 4에서 A부분은 초기 상태에서 알고 있는 이동 로봇의 입·출력 데이터를 가지고 클러스터링을 함으로 B부분 퍼지 제어기의 초기 퍼지 규칙 및 멤버십 함수를 생성해 낸다. 이동 로봇이 주행을 시작하면 기존의 데이터를 포함해 단위 시간(1sec)마다 얻어지는 입·출력 데이터 쌍을 이용하여 현재의 이동 로봇의 상태에 적합한 멤버십 함수 및 퍼지 규칙을 생성한다. 따라서 B부분 퍼지 제어기의 멤버십 함수 및 퍼지 규칙은 매 시간 튜닝된다. 그림 4의 B부분은 현

재의 이동 로봇의 위치와 제어 목적인 벽과의 간격의 차를 이용하여 이동 로봇이 제어 목적에 빠른 수렴을 할 수 있도록 A부분에서 생성된 멤버쉽 함수의 파라미터들을 단위 시간마다 스케일링하는 하나의 퍼지 제어기이다. 그림 4의 C부분은 A와 B의 과정을 통해 생성된 멤버쉽 함수와 퍼지 규칙을 이용하여 이동 로봇이 주행할 방향을 구하는 퍼지 제어기 부분이다.

이동 로봇의 벽면 추종 주행은 다음과 같이 크게 7단계로 나눌 수 있으며, 이 과정을 반복하므로 이동 로봇은 원하는 기준선으로 빠른 속도를 가지고 수렴할 수 있다.

변수 거리차(dif)의 소속 함수를 스케일링한다.

[단계 6] 단계 3과 단계 5에서 유도된 퍼지 추론 규칙 및 스케일링 된 소속 함수를 이용하여 이동 로봇의 현재 상태에 적합한 다음 상태로의 조향각($\Delta\theta$)을 구한다.

[단계 7] 구해진 조향각($\Delta\theta$)에 따라 이동 로봇은 주행하게 된다.

여기서, T는 주행 중 이동 로봇의 시간적인 현재를 나타낸다.

IV. 실험 결과

본 논문에서 제안한 알고리즘의 효율성을 검증하기 위한 실험은 컴퓨터 모의 실험과 실제 이동 로봇을 이용한 주행 실험을 통째 수행하였다. 컴퓨터 모의 실험은 공학용 프로그램인 MATLAB을 이용하여 주행 환경과 실제 이동 로봇을 설정하여 실험하였고, 실제 이동 로봇의 주행 실험에서 사용한 이동 로봇은 TRC사의 본체를 사용하였으며, 주위 환경 인식을 위한 센서로는 Polaroid사의 송수신 겸용 초음파 센서를 이용하였다. 2개의 부시스템을 제어하기 위해서 Pentium급 컴퓨터를 사용하였고, 컴파일 환경은 Visual c++이다.

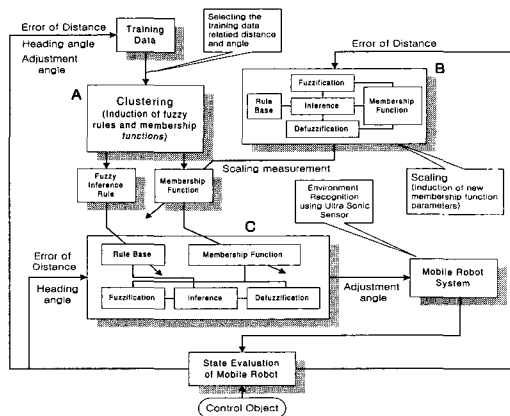


그림 4. 본 논문에서 제안한 적응 퍼지 제어 시스템의 블록 선도

Fig. 4. Block diagram of adaptive fuzzy control system proposed in this paper.

[단계 1] 이동 로봇의 현재 상태(T)를 평가한다. 이것은 주어진 환경 내에서 현재 이동 로봇과 기준 선과의 거리차(dif), 그리고 이동 로봇의 진행 방향(θ)을 계산하는 단계이다.

[단계 2] 계산된 거리차(dif)와 진행 방향(θ)에 따른 학습 데이터 쌍(T)을 선택한다.

[단계 3] 선택된 학습 데이터 쌍(T)으로 클러스터링 과정을 통하여 퍼지 추론 규칙 및 소속 함수(T)를 생성해 낸다. 여기서, 선택된 학습 데이터 쌍(T)이 전 단계의 데이터 쌍(T-1)과 같으면 클러스터링 과정은 생략하고 퍼지 추론 규칙 및 소속 함수(T)는 전(T-1)과 동일하게 간주한다.

[단계 4] 평가된 거리차(dif)에 따른 스케일 척도(T)를 구한다.

[단계 5] 구해진 스케일 척도(T)에 따라서 입력

1. 모의 실험 및 결과 고찰

본 절의 모의 실험에서의 성능 평가는 이동 로봇이 주행 목적으로 수렴하는 정도와 주행 중 데이터 처리 시간에 초점을 두며, 크게 3가지로 비교 분석하였다. 첫 번째는 기존의 퍼지 제어기를 이용한 이동 로봇의 오른쪽 벽면 추종 주행, 두 번째는 기존의 퍼지 제어기에 본 논문에서 제안한 클러스터링 기법을 적용한 적응 퍼지 제어기를 이용한 이동 로봇의 오른쪽 벽면 추종 주행 그리고 세 번째로는 클러스터링 기법과 스케일링 기법을 함께 적용한 적응 퍼지 제어기를 이용한 이동 로봇의 오른쪽 벽면 추종 주행이며, 각각 30cm/1step으로 총 98step에 대한 실험 결과를 나타낸다. 또한 각각의 실험마다 이동 로봇의 위치(x, y)와 초기 진행 방향(θ)을 3가지의 경우로 나누어 실험하였다. 3가지의 위치 및 방향은 각각의 실험에서 동일하며, 실험 공간은 폭이 2m이고, 길이는 30m로 설정하였다. 또한 제어 목적도 벽면에서 30cm를 따라 주행한다는 점은 동일하다. 이동 로봇

의 3가지 출발 위치 (x, y) 및 진행 방향 (θ) 은 표 2와 같다.

표 2. 이동 로봇의 초기 위치 및 초기 진행방향
Table 2. Initial position and heading angle of mobile robot.

x 좌표	y 좌표	초기 진행 방향
100cm	200cm	30°
100cm	150cm	0°
100cm	220cm	-30°

그림 5는 기존의 퍼지 제어기를 이용하여 각기 다른 3가지의 초기 위치 및 진행 방향 (θ) 으로 이동 로봇이 오른쪽 벽면 추종 주행을 한 결과이고, 그림 6은 각각에 대한 기준선과의 오차를 나타낸다.

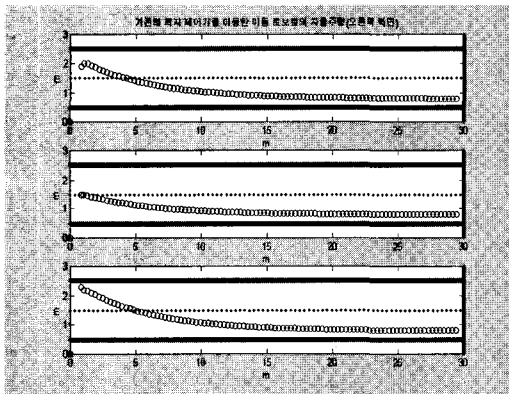


그림 5. 기존의 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig 5. The right wall following navigation using existing fuzzy controller.

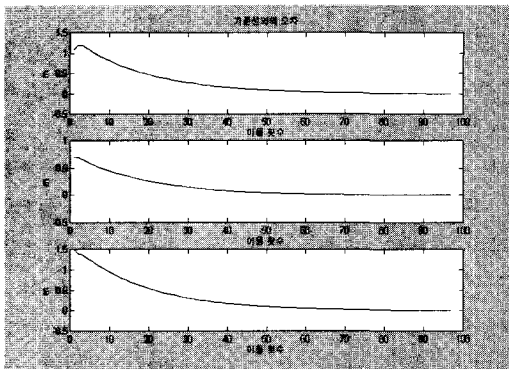


그림 6. 기준선과의 오차
Fig. 6. Error from reference line.

그림 5 및 그림 6의 결과로부터, 기존의 퍼지 제어기를 이용한 이동 로봇의 벽면 추종 주행은 제어 목적인 벽면 30cm에 안정적으로 수렴함을 알 수 있다. 3가지 경우를 비교하면 대략 15m 이상 주행, 즉 50번 정도 주행하면 기준선으로 거의 근접함을 확인할 수 있다. 3가지 경우 결과가 조금 다른 것은 초기 진행 방향의 영향에 기인한다고 추측할 수 있다. 초기 진행 방향이 벽면을 향하고 있으면 다른 방향에 비해 더욱 빨리 기준선으로의 수렴함을 알 수 있다.

그림 7은 기존의 퍼지 제어기에 본 논문에서 제안한 클러스터링 기법을 첨가한 후의 이동 로봇의 오른쪽 벽면 추종 주행의 결과를 나타내고 그림 8은 이동 로봇과 기준선과의 오차를 나타낸다.

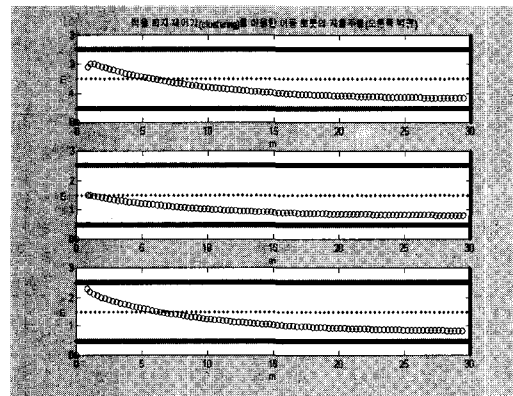


그림 7. 적응 퍼지 제어기(클러스터링)을 이용한 오른쪽 벽면 주행
Fig. 7. The right wall following navigation using adaptive fuzzy controller(clustering).

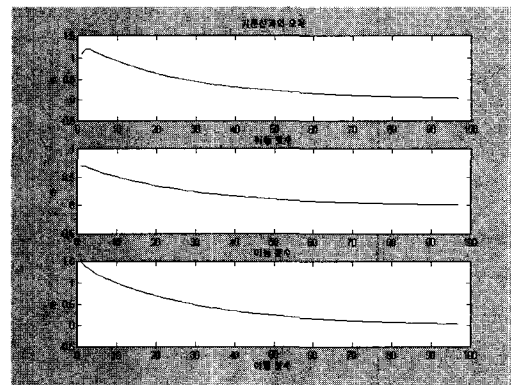


그림 8. 기준선과의 오차
Fig. 8. Error from reference line.

그림 7 및 그림 8의 결과로부터, 클러스터링 기법만을 사용한 적응 퍼지 제어기에 의한 이동 로봇의 벽

면 추종 주행은 마찬가지로 제어 목적인 벽면 30cm에 안정적으로 수렴함을 알 수 있다. 그러나 수렴 속도는 기존의 퍼지 제어기에 비하여 다소 떨어진다. 대략 20m 이상 주행, 즉 70번 정도 주행하면 기준선으로 거의 근접함을 확인 할 수 있었다.

이 경우에도 3가지 경우 결과가 조금 다른 것은 초기 진행 방향의 영향에 기인한다고 추측할 수 있다. 그림 9는 기존의 퍼지 제어기에 클러스터링 기법과 스케일링 기법을 모두 사용한 퍼지 제어기에 의한 이동 로봇의 벽면 추종 주행의 결과를 나타내고, 그림 10은 이동 로봇과 기준선과의 오차를 나타낸다.

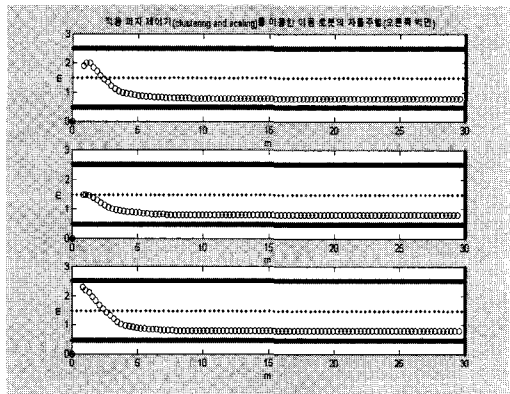


그림 9. 적응 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig. 9. The right wall following navigation using adaptive fuzzy controller (clustering and scaling).

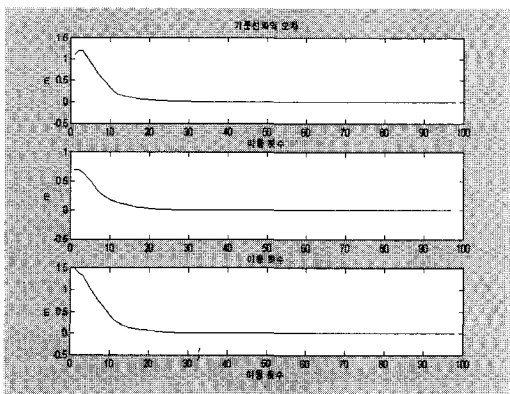


그림 10. 기준선과의 오차
Fig. 10. Error from reference line.

그림 9 및 그림 10의 결과로부터, 클러스터링 기법과 스케일링 기법을 모두 사용한 적응 퍼지 제어기에 의한 이동 로봇의 벽면 추종 주행은 앞의 두 가지의

경우에 비하여 훨씬 빠른 수렴성을 가진다. 단지 5m 정도 이상, 다시 말해 10번 이상 주행하면 제어 목적인 벽면 30cm에 안정적으로 수렴한다. 이 경우에도 마찬가지로 초기 진행 방향의 영향이 조금 미치고 있음을 알 수 있다. 표 4.2는 각각의 3가지의 주행 실험에 대한 평균 오차를 나타낸다.

표 3. 주행 실험의 평균 오차
Table 3. Mean error for navigation experiment.

		기존의 퍼지 제어기	적용 퍼지 제어기(클러스터링)	적용 퍼지 제어기(클러스터링과 스케일링)
이동 로봇의 시작 상태	100cm, 200cm, 30°	0.2493m	0.3547m	0.0953m
	100cm, 150cm, 9°	0.1359m	0.1923m	0.0564m
	100cm, 220cm, -30°	0.2905m	0.3885m	0.1135m

평균 오차를 보면 기존의 퍼지 제어기에 비하여 클러스터링 기법을 적용한 적응 퍼지 제어기의 평균 오차가 더 크음을 알 수 있다. 그만큼 기준선으로의 수렴 속도가 느리다는 것을 의미하는 것이다. 반면에 클러스터링 기법과 스케일링 기법을 모두 적용한 적응 퍼지 제어기의 평균 오차는 앞의 두 경우에 비해 월등히 작다는 것을 확인하였다. 다시 말해서 스케일링 기법은 수렴 속도를 향상시키는 점에서는 월등함을 알 수 있다.

표 4는 각각의 3가지 실험에 대한 시뮬레이션 동안 데이터 처리 시간을 나타낸다. 각각의 3가지의 시뮬레이션 동안의 데이터 처리 속도를 비교해 보면, 기존의 퍼지 제어기는 퍼지 추론 시간이 전체 시간의 64%를 차지할 만큼 많은 시간을 요구한다. 반면에 클러스터링 기법을 통하여 퍼지 추론 규칙을 줄이고, 간략화한 비퍼지화 방법을 사용한 적응 퍼지 제어기의 시뮬레이션 시간은 많은 시간이 주행시에 소요되지만, 퍼지 추론 과정에서는 많은 시간을 요하지 않다는 것을 알 수 있다. 클러스터링 기법만을 적용한 적응 퍼지 제어기와 클러스터링 기법과 스케일링 기법을 모두 사용한 적응 퍼지 제어기의 데이터 처리 속도를 비교해 보면 후자가 또 하나의 퍼지 제어기인 스케일링 부분에서 전자보다 조금 더 시간이 소요 될 뿐 그리 큰 차이가 나지 않음을 알 수 있다. 따라서 클러스터링 기법은

퍼지 추론 규칙을 줄임으로써 데이터 처리 속도를 향상시킬 수 있다.

표 4. 시뮬레이션 동안 데이터 처리 시간
Table 4. Data processing time for simulation.

		기존의 퍼지 제어기	적응 퍼지 제어기 (클러스터링)	적응 퍼지 제어기 (클러스터링과 스케일링)
데이터 처리 시간	주행 시간(%)	28%	61%	65%
	퍼지 추론 시간(%)	64%	16%	22%
	그외의 시간 (graphic 등등)(%)	8%	23%	13%
	전체 시간(seconds)	72.54 seconds	15.2 seconds	20.72 seconds

그림 11은 본 논문에서 제안한 알고리즘(클러스터링과 스케일링)을 적용한 이동 로봇의 주행에 따른 퍼지 제어기의 변화되는 규칙 수를 나타낸다. 그림에서 알 수 있듯이 이동 로봇의 상태 즉, 기준선과의 거리차(dif)에 따라 퍼지 추론 규칙수가 변화됨을 알 수 있다. 초기의 3가지의 이동 로봇 상태에 따른 각각의 결과는 비슷하며, 제어 목적에 수렴하였을 경우는 3가지 모두 원래의 25개의 퍼지 추론 규칙에서 단지 8개만으로 줄어든 퍼지 추론 규칙을 가진다.

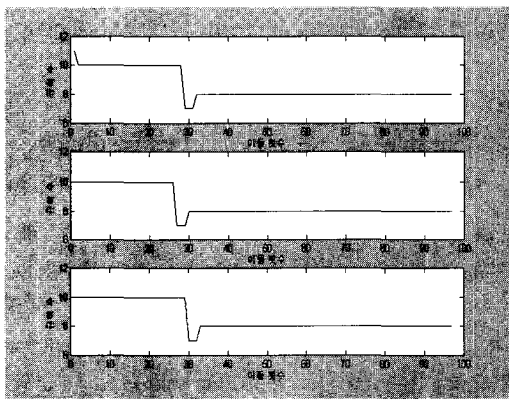


그림 11. 퍼지 제어기의 변화되는 규칙수
Fig. 11. Changing rules of fuzzy controller.

2. 주행 실험 및 결과 고찰

본 절에서는 오른쪽 벽면 추종 주행을 기본으로 기존의 퍼지 제어기를 이용한 주행 실험과 본 논문에서 제안한 적응 퍼지 제어기를 이용한 주행 실험을 각기

두 가지의 경우에 대하여 비교 검토하였다. 첫 번째는 이동 로봇의 초기 진행 방향이 기준선과 평행한 경우이고, 두 번째는 이동 로봇의 초기 진행 방향이 기준선을 향한 경우이다. 그리고 기준선은 오른쪽 벽면으로부터 60cm이며, 폭이 2.5m인 통로에서 20cm/sec로 약 10m정도 주행하였다. 그림 12 및 그림 13은 이동 로봇의 초기 진행 방향이 기준선과 평행한 경우의 주행 실험 결과이다. 결과 그림의 1 눈금은 2m이다.

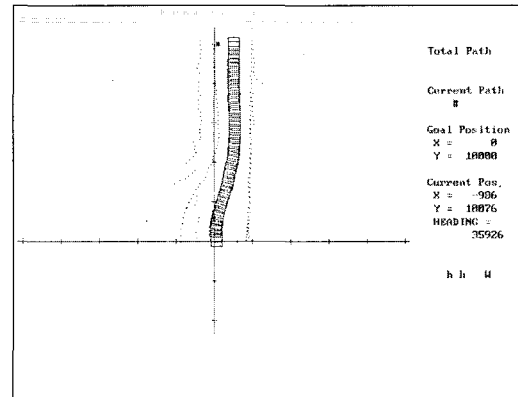


그림 12. 기존의 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig. 12. The right wall following navigation using existing fuzzy controller.

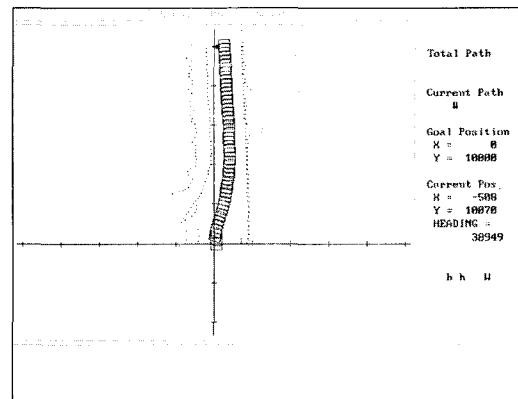


그림 13. 적응 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig. 13. The right wall following navigation using adaptive fuzzy controller.

그림 12의 결과를 보면 기준선과 평행한 초기 진행 방향을 가지고 기존의 퍼지 제어기를 이용한 주행은 약 4m이상 진행한 후에 제어 목적인 벽과 60cm 떨어진 기준선에 수렴함을 알 수 있었다. 반면에 그림 13을 보면 적응 퍼지 제어기를 이용한 주행은 약 2.3m

이상 진행하면 기준선에 수렴함을 볼 수 있다. 따라서 적응 퍼지 제어기는 기존의 고정된 퍼지 제어기의 성능에 비하여 좀 더 빠른 수렴성을 갖도록 함을 알 수 있다.

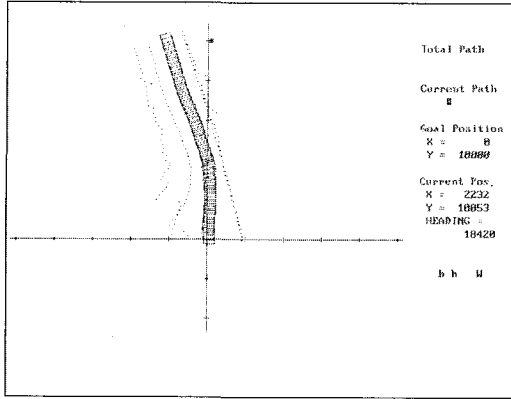


그림 14. 기존의 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig. 14. The right wall following navigation using existing fuzzy controller.

그림 14 및 그림 15는 이동 로봇의 초기 진행 방향이 기준선을 향한 경우의 주행 실험 결과이며 마찬가지로 적응 퍼지 제어기를 이용하여 벽면 주행 한 경우가 기준선으로 빠른 수렴성을 가짐을 확인할 수 있다.

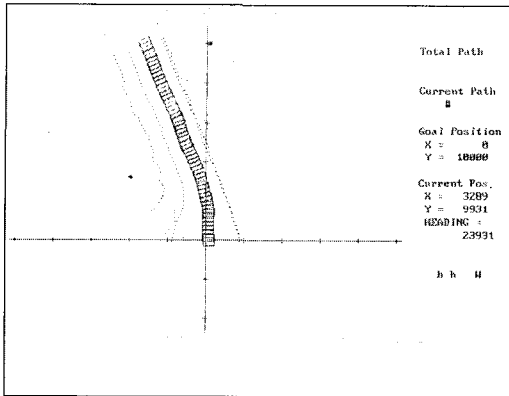


그림 15. 적응 퍼지 제어기를 이용한 오른쪽 벽면 주행
Fig. 15. The right wall following navigation using adaptive fuzzy controller.

V. 결론

본 논문은 이동 로봇의 자율 주행 제어를 위해 지능 제어의 관점에서의 퍼지 이론에 기초한 적응 퍼지 제어기 설계에 관한 연구로서, 고정된 입·출력 변

수의 소속 함수를 갖는 기존의 퍼지 제어기의 단점인 긴 데이터 처리 속도와 제어 목적으로의 비효율적인 수렴 속도를 향상시키는 방법을 제안하였다. 향상된 데이터 처리 속도를 위해서는 이동 로봇이 주행 중 획득한 학습 데이터의 쌍을 클러스터링 기법을 이용하여 이동 로봇 주행 상태에 적합한 소속 함수와 필요한 퍼지 추론 규칙만을 자동으로 유도하는 방법을 사용하였고, 제어 목적으로의 빠른 수렴성을 위해서는 클러스터링 기법에 의해 생성된 입력 변수의 소속 함수 매개 변수를 스케일링함으로써 주행 환경에 적합한 새로운 입력 변수의 소속 함수를 유도하는 방법을 사용하였다.

모의 실험 결과, 이동 로봇의 벽면 추종 주행은 이동 로봇의 초기 진행 방향과 주위 환경의 영향을 받을 수 있었다. 그리고 기존의 퍼지 제어기에 클러스터링 기법만을 적용한 적응 퍼지 제어기는 데이터 처리 속도 면에서는 5배 이상 크게 향상되었지만 기준선으로의 평균 오차는 약간 큼을 알 수 있었다. 이는 기존의 퍼지 제어기의 고정된 25개의 퍼지 추론 규칙에 비하여 클러스터링 기법을 사용한 적응 퍼지 제어기의 퍼지 추론 규칙은 약 8~11로 감소했기 때문이다. 그리고 기존의 퍼지 제어기의 데이터 처리 시간은 64% 이상이 퍼지 제어기의 추론 과정에서 소요되지만 클러스터링 기법을 사용한 적응 퍼지 제어기의 데이터 처리 시간은 추론 과정에서 16% 정도로 감소됨을 확인하였다. 그리고 실제 벽면 주행 실험 결과는 모의 실험의 경우에 비하여 다소 떨어지기는 하지만 기존의 퍼지 제어기를 이용한 경우에 비하여 본 논문에서 제안한 적응 퍼지 제어기를 이용한 경우에 이동 로봇이 좀 더 빠른 수렴성을 가짐을 확인하였다.

VI. 참고 문헌

- [1] R. Kuc and B. Barshan, "Navigation Vehicles through an Unstructured Environment with Sonar", Proc. IEEE Int. Conf. Robotics Automation, pp.1422-1426, May, 1989.
- [2] J. Borenstein and Y. Koren, "Obstacle Avoidance with Ultrasonic Sensors," IEEE Robotics Automation., vol. RA-4, No. 2, pp. 213-218, 1988.
- [3] J. Borenstein and Y. Koren, "Real time

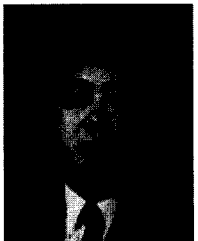
- Obstacle Avoidance for Fast Mobile Robots", IEEE Trans. Systems Man Cyben, vol. 19, No.5 pp. 1179-1187, 1989.
- [4] J.L.Crowley, "World Modelling and Position Estimation for a Mobile Robot using Ultra sonic Ranging," Proc. IEEE Int. Conf. Robotics Automation, pp.674-680, May 1989.
- [5] S. Ishikawa, "A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control", IEEE/RSJ International Workshop on Intelligent Robot and Systems, pp. 1013-1018, 1991.
- [6] A. Chohra, F. Sif and S. Talaoubrid "Neural Navigation Approach of an Autonomous Mobile Robot in a Partially Structured Environment", 2nd IFAC Conf. on Ini. Auto. Veh., pp. 238-243, 1995.
- [7] A. Moran, and M. Nagai, "Autonomous Parking of Vehicles with Intelligent Fuzzy-Neural Networks", Proc. of AVEC'94, pp. 270-275, Tsukuba, Japan, 1994.
- [8] T. Takagi, and M. Sugeno, "Fuzzy Identification of Systems and It's Application to Modeling and Control", IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-15, No.1, pp. 116-132, 1985.
- [9] A.Maeda, and M.Funabashi, "A Self-Tuning Algorithm for Fuzzy Membership Functions Using Computational Flow Network", Proc. of 4th IFSA '91 pp.129-132, 1991.
- [10] T. P. Hong, and C. Y. Lee, "Induction of Fuzzy Rules and Membership Functions from Training Examples", Fuzzy Sets and Systems, vol. 84, pp. 33-47, 1995.
- [11] R. L. Madarasz, L. C. Heiny, R. F. Crompt, and N. M. Mazur, "The Design of an Autonomous Vehicle for the Disabled", IEEE Jour. of Robotics and Automation, vol. 2, No.3, pp. 117-126, 1986
- [12] B. D. Andrea-Novel, G. Bastin and G. Campion, "Modelling and Control of Nonholonomic Wheeled Mobile Robots", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1130-1135, April, 1991
- [13] X. yun, Y. Yamamoto, "Internal Dynamics of a Wheeled Mobile Robot", Proc. of IROS'93, pp. 1288-1294, July, 1993

저 자 소 개



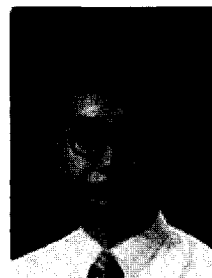
吳 俊 燮(正會員)

1997년 경기대학교 전자공학과 학사. 1999년 경기대학교 전자공학과 석사. 현재 연세대학교 전기 및 컴퓨터 공학과 박사과정 재학 중



崔 允 浩(正會員)

1980년 연세대학교 전기공학과 학사. 1982년 연세대학교 대학원 전기공학과 석사. 1991년 동 대학원 박사. 현재 경기대학교 전자·기계공학부 부교수



朴 珍 培(正會員)

1977년 연세대학교 전기공학과 학사. 1985년~1990년 Kansas State University 전기 및 컴퓨터 공학과 박사. 1990년~1991년 Kansas State University 전기 및 컴퓨터 공학과 조교수. 현재 연세대학교 전기 및 컴퓨터 공학과 부교수