

자율가변 구조의 신경망 모델을 이용한 구륜 이동 로봇의 위치 제어

김기열*/김성희**/김현**/임호***/정영화****

요 약

본 논문에서는 퍼지모델의 최적 입-출력 소속함수들(membership functions) 및 규칙기반(rulebase) 얻기 위한 자율가변구조의 신경망 알고리즘을 제안하였으며 구륜 이동 로봇(WMR : Wheeled Mobile Robot)의 위치, 속도 방향제어를 위한 퍼지-신경망 제어기 설계를 설계하였다. 제안된 알고리즘에서 입-출력 소속함수의 파라미터들을 찾기 위하여 유전알고리즘을 응용한다. 유전알고리즘에 의해 출력술어의 원소가 증가되며, 규칙기반이 원소의 증가에 의하여 조절된다. 새롭게 조절된 제어기는 출력술어의 증가를 수행하지 않은 제어기와 경쟁하며, 만약 새롭게 조절되어진 퍼지-신경망 제어기가 경쟁에서 진다면, 그 제어법칙은 소멸한다. 그 반대로 조절된 제어기가 생존한다면, 출력술어의 증가된 각 원소들 및 변화된 시스템의 규칙기반이 제어기에 적용된다. 출력술어 및 규칙의 조절이 완료된 후 입력소속함수들에 대한 탐색이 제약조건을 가지고 수행되며 입력소속함수들의 탐색이 완료된 후 출력소속함수의 미세 조절이 수행된다.

1. 서론

현재 구륜 이동 로봇의 효율적 제어를 위해 많은 연구들이 수행되고 있다. 구륜 이동 로봇은 지금까지는 수학적 모델링에 기초한 제어기가 설계 되어왔으나 non-holonomic constraints등의 특성 때문에 제어에 많은 어려움을 지녀왔다. 이와 같은 문제를 해결하기 위해 제어 대상에 대한 정확한 수학적 모델링이 없이 전문가의 지식을 이용한 퍼지 제어기가 널리 연구되어 왔다. 그러나 퍼지

제어기는 원하는 성능을 얻기 위해 많은 시행착오를 거쳐야만 하며 다양한 시스템에 대해 일관되고 구조화된 퍼지 규칙과 입출력 소속함수를 찾는다는 것은 어렵다. 이런 단점을 보완하기 위해서 신경망과 퍼지 알고리즘이 결합하는 연구가 진행되어 왔다^[1-5]. 퍼지 신경망은 퍼지의 구조를 신경망을 이용하여 생성한 뒤 학습을 통해 자신의 구조와 매개변수들을 변화시켜 논리적 제어를 수행한다^[6-9].

본 논문에서는 2자유도 구륜 이동 로봇의 실시간 제어를 위한 퍼지-신경망 제어기를 설계한다. 즉 퍼지 시스템의 규칙들과 소속함수들이 실시간 조절된다. 학습 알고리즘을 제시한다. 본 논문에서는 기존의 신경망 시스템이 갖는 입출력 데이터에 의한 반복적인

* 부천대학 컴퓨터제어과 교수
** 미래산업주식회사 미래연구센터
*** 부천대학 컴퓨터제어과 교수
**** 성남기능대학 전자과 교수
***** 남서울대학교 전자정보통신공학부 교수

훈련을 통한 학습이 수행되지 않으며 제어 대상의 입력 정보를 이용하여 실시간으로 퍼지 제어기의 규칙 및 소속함수를 변화시킨다. 또한 기존의 퍼지 신경망과 달리 x축과 y축에 별도의 독립된 퍼지 신경망을 사용한다. 제안된 알고리즘과 퍼지 신경망 구조의 효율성을 입증하기 위하여 2 자유도 구륜 이동 로봇에 대하여 모의 실험을 수행하였다.

II. 가변 구조의 퍼지 신경망

2.1 일반적인 퍼지 제어기

퍼지 제어기는 입력 u_1, u_2, \dots, u_i 들과 출력 o_1, o_2, \dots, o_k 들이 논리적으로 연결되어 제어기가 구성된다. 퍼지 제어기는 퍼지화부, 퍼지 규칙부, 규칙 추론부, 비퍼지화 부로 나누어진다. 입력부는 실제 시스템의 입력 u_i 를 정규화된 범위의 확률값으로 변환시키며, 퍼지 규칙부는 일반적으로 식 (1)처럼 if-then 구문으로 언어적으로 표현된다. 퍼지 규칙의 수는 입력변수의 수에 비례한다.

$$IF u_i = PM \text{ and } u_j = NB \text{ then } y_{ij} = PM \quad (1)$$

퍼지 추론부는 시스템에 실제 입력된 값을 설정된 규칙에 따라 퍼지 출력값을 얻기 위해 추론을 수행하는 부분이다. 비퍼지화부는 퍼지 추론의 결과를 실제 출력 영역에 맞도록 변환시켜주며 보통 무게 중심법을 많이 사용한다. 이와 같은 퍼지 제어기의 단점은 제어기의 퍼지 규칙과 퍼지 소속 함수의 설

계에 일관성이 없으며 퍼지 규칙과 퍼지 소속 함수에 대한 최적값을 얻기 어려운 것이다.

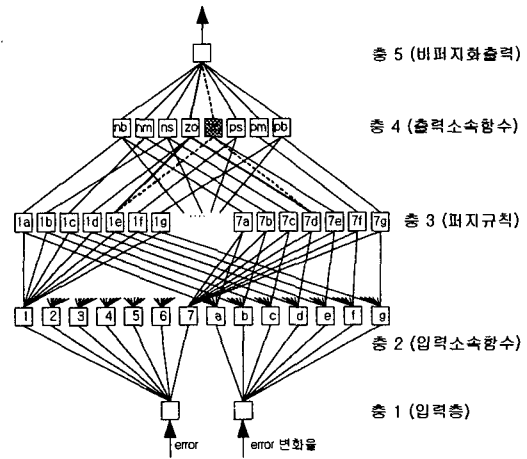


그림 1. 퍼지 신경망의 구조

Fig. 1 Structure of fuzzy neural network

2.2 퍼지 신경망의 구조

퍼지 신경망 제어기는 퍼지 제어기에 신경망의 학습기능, 구조조정특징을 도입한 제어기이다. 이 모델은 구조 학습과 매개변수 학습의 2가지 방법을 사용한다. 구조 학습은 반복적인 학습에 의해 새로운 출력 퍼지 소속함수가 생성이 되며 규칙부의 조절이 동시에 이루어진다. 매개변수 학습은 신경망의 노드를 연결하는 연결 가중치의 변경에 의해 이루어진다^[5]. Lin and Lee 신경망의 구조조정 방법은 반복적인 학습을 통해 노드를 임시 생성하여 성능이 향상되는 제어기를 찾는 것이다. 이런 경우 퍼지 신경망의 구조가 점점 커지게 되고 신경망의 최적성을 얻기가 힘들다.

Lin and Lee 신경망은 모든 입력에 대해

서 노드를 구성한다. 이 구조는 모든 입력이 각각의 관계를 모두 고려할 수 있다는 장점이 있으나 신경망의 크기가 커지고 계산시간이 오래 걸리는 단점이 있다. 그래서 본 논문은 기준 궤적값들에 대해 완전히 분리된 퍼지 신경망을 사용한다. 이 구조에서 퍼지 신경망의 크기는 작아지고 계산속도가 향상된다. 각각의 퍼지 신경망은 5개의 층으로 구성되며 입력은 기준궤적과 실제 궤적의 오차와 오차변화율로 이루어진다.

2.3 퍼지 신경망의 동작

본 논문에서 퍼지 신경망의 전방향 동작은 다음과 같다. 그림 1에서 층 1은 2개의 노드로 이루어져 있다. 각 입력들은 오차와 오차 변화율이다. 각 노드는 식 (2)에 의해 입력 $u_i^{(1)}$ 을 출력 a 으로 전달한다.

$$a = u_i^{(1)} \quad (2)$$

층 2는 퍼지 제어기의 입력 소속함수부분을 구성한다. 한 개의 입력 $u_i^{(2)}$ 는 7개의 노드와 연결되며 입력 $u_i^{(2)}$ 는 식 (3)에 의해서 출력값 a_{ij} 으로 변환한다.

$$a_{ij} = e^{-\frac{(u_i^{(2)} - m_{ij})^2}{2\sigma_{ij}^2}} \quad (3)$$

층 3는 퍼지 규칙을 나타낸다. 층 3의 1개의 노드는 층 2의 노드 2개와 연결되어 퍼지 제어기의 if-then문장의 동작을 수행한다. 오차입력과 연결된 7개의 노드와 오차 변화율

입력과 연결된 7개의 노드가 2개씩 조합되어 총 49개의 노드가 존재한다. 입력 $\mu^{(3)}(x)$ 에 대해 식 (4)에 의해 출력 a 가 생성된다.

$$\begin{aligned} a &= T(\mu_A^{(3)}(x), \mu_B^{(3)}(x)) \\ &= \mu_A^{(3)}(x) \wedge \mu_B^{(3)}(x) \\ &= \min(\mu_A^{(3)}(x), \mu_B^{(3)}(x)) \end{aligned} \quad (4)$$

층 4는 퍼지 제어기의 출력 소속함수에 해당된다. 그리고 층 3의 노드들과 층 4의 노드의 연결구조가 퍼지 추론의 역할을 한다. 층 4의 7개의 노드는 층 3의 출력값과 연결함으로써 퍼지 규칙이 설정된다. 노드는 모든 입력값에 대해서 식 (5)의 T-conom동작으로 한번에 처리된다.

$$a = \max(u_1^{(4)}, u_2^{(4)}, \dots, u_n^{(4)}) \quad (5)$$

층 5는 비퍼지화 동작을 하며 사용된 비퍼지화 방법은 식 (6)과 같다.

$$a = \frac{\sum_j (m_{ij} \sigma_{ij}) u_{ij}^{(5)}}{\sum_j \sigma_{ij} u_{ij}^{(5)}} \quad (6)$$

E 는 오차함수로 역전파 학습은 $\frac{\partial E}{\partial \omega}$ 을 최소화시키는 방향으로 가중치를 조정한다. 층 4와 층 2에서 소속함수의 중앙값 m_{ij} 과 너비값 σ_{ij} 은 아래의 식 (7)에 의해 변경된다.

$$\omega(t+1) = \omega(t) + \eta \left(-\frac{\partial E}{\partial \omega} \right) \quad (7)$$

η 는 학습률을 나타내는 매개변수이고 0과 1사이의 값으로 설정되며 클수록 매개변수의 변화량은 커진다. 본 논문에서는 역전파 학습에 의한 매개 변수 학습과 구조조정은 바로 이루어지지 않는다. 일단 노드를 연결하는 가중치의 변화량을 일반적인 역전파 학습 방법을 이용해 계산한다. 층 5에서는 퍼지 신경망의 출력 $y(t)$ 과 기준 데이터 $y^d(t)$ 의 오차를 층 4로 전달된다.

$$\delta^{(5)} = -\frac{\partial E}{\partial y} = y^d(t) - y(t) \quad (8)$$

층 4에서는 층 5에서 전파된 역전파 오차를 이용하여 중앙값 m_i 과 너비값 σ_i 은 식 (9)에서 구해진다.

$$m_i(t+1) = m_i(t) + \eta \delta^{(5)} \frac{\sigma_i u_i^{(5)}}{\sum_i \sigma_i u_i^{(5)}} \quad (9)$$

$$\sigma_i(t+1) = \sigma_i(t) + \eta \delta^{(5)}$$

$$\frac{m_i u_i^{(5)} \left(\sum_i \sigma_i u_i^{(5)} \right) - \left(\sum_i m_i \sigma_i u_i^{(5)} \right) u_i^{(5)}}{\left(\sum_i \sigma_i u_i^{(5)} \right)^2}$$

아래층으로 전파되는 오차는 식 (10)과 같다.

$$\delta_i^{(4)} = \delta^{(5)} \frac{m_i \sigma_i \left(\sum_i \sigma_i u_i^{(4)} \right) - \left(\sum_i m_i \sigma_i u_i^{(4)} \right) \sigma_i}{\left(\sum_i \sigma_i u_i^{(4)} \right)^2} \quad (10)$$

식 (9)의 중앙값과 너비값은 유사성을 이용한 출력노드의 생성 및 결합에 사용한다. 층 3에서는 퍼지 규칙 노드와 출력 노드가

가중치 1로 연결되어 있다. 따라서 역전파 오차만 계산된다.

$$\begin{aligned} \delta_i^{(3)} &= -\frac{\partial E}{\partial a_i^{(3)}} = -\frac{\partial E}{\partial u_i^{(4)}} \\ &= -\frac{\partial E}{\partial a_i^{(4)}} \frac{\partial a_i^{(4)}}{\partial u_i^{(4)}} = -\frac{\partial E}{\partial a_i^{(4)}} = \delta_i^{(4)} \end{aligned} \quad (11)$$

역전파 오차는 $\delta_i^{(3)} = \delta_i^{(4)}$ 이다. 만약 출력노드의 연결이 여러 개인 경우 $\delta_i^{(3)} = \sum_k \delta_k^{(4)}$ 의 동작으로 아래층으로 전달한다. 층 2에서는 노드의 중앙값과 너비값이 식 (12)에 의해 계산한다.

$$m_{ij}(t+1) = m_{ij}(t) + \eta \delta_i^{(2)} e^{f_i} \frac{2(u_i^{(2)} - m_{ij})}{\sigma_{ij}^2} \quad (12)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \eta \delta_i^{(2)} e^{f_i} \frac{2(u_i^{(2)} - m_{ij})^2}{\sigma_{ij}^3}$$

여기서 $\delta_i^{(2)} = \sum_k a_k$ 이며 a_k 는 가장 작은 출력을 발생한 노드와 연결된 층 3의 역전파 오차 $\delta_i^{(3)}$ 이다. 가장 작은 출력을 발생한 노드의 매개변수만 고려한다.

2.4 유사성을 이용한 구조조정 알고리즘

퍼지 제어기는 고정된 구조이며 소속함수나 퍼지 규칙부의 수정이 용이하지 않다. 하지만 퍼지 신경망은 노드의 매개변수와 노드를 연결하는 가중치의 수정이 가능하다. 또한 일반적인 퍼지 신경망은 매개변수를 조정하면서 새로운 노드를 생성하여 구조적인 수

정을 수행한다. 본 논문에서는 퍼지 신경망의 구조조정에서 유사성을 이용한 노드의 생성, 결합 방법을 제시한다. 노드는 퍼지 제어기의 소속함수가 사용하는 중앙값 m_{ij} 과 너비값 σ_{ij} 을 매개변수로 사용하는데 이 중앙값과 너비값의 변화량이 노드의 유사성을 판별하는 기준이 된다. 유사성은 노드의 연결가중치의 변화량으로 판단한다. 유사성이 작으면 새로운 노드를 생성하고 유사성이 크면 기존의 노드가 갱신된다. 유사성이 큰 노드가 2개 이상 존재한다면 노드의 결합이 가능하다. 노드의 생성, 수정, 통합은 제어기의 성능을 향상시키고 퍼지 신경망의 크기를 변화시킨다. 구조조정은 퍼지 신경망의 층 4의 노드에서 일어난다. 층 4의 노드는 식 (13)의 PM에 해당한다. 새로운 노드가 생성되면 식 (14)이 추가된다.

$$\text{if } u_{ij}^1 = PS \text{ and } u_{ij}^2 = NM \text{ then } o_{jk} = PM_{old} \quad (13)$$

$$\text{if } u_{ij}^1 = PS \text{ and } u_{ij}^2 = NM \text{ then } o_{jk} = PM_{new} \quad (14)$$

새로운 노드의 생성은 퍼지 신경망에 새로운 퍼지 규칙의 추가를 의미한다. 층 4의 i 번째 노드에서 매개변수 m_{i-new} , σ_{i-new} 을 계산하여 원래 노드와의 유사성 $M(m_i, \sigma_i)$ 을 계산한다. 그리고 유사성이 높은 다른 노드가 있는지 $m_{i-closest}$, $\sigma_{i-closest}$ 을 검색한다. 이 2가지가 계산되면 표 1에 제시한 방법을 통해 층 4에 새로운 출력 노드의 생성과 결합을 한다. 판단기준은 유사성과 노드 생성 임계점(strength threshold α)의 크기로 판단한다. 표 1의 조건(1)은 예상된 매개변수가 현재의 매개변수와 유사한 경우이다.

조건 (1)을 만족하는 경우 새로운 노드를 생성되지 않고 기존 출력 노드의 매개변수를 수정한다. 단 유사성이 높은 다른 노드가 검색된다면 원래의 출력 노드는 갱신되지 않는다. 대신에 유사성이 높은 다른 노드의 매개변수가 수정된다. 표 1의 조건 (2)은 예상된 매개변수와 기존의 매개변수의 유사성이 작은 경우이다. 유사성이 임계점 α 보다 큰 다른 노드가 검색되지 않는다면 새로운 노드가 생성된다. 만약 유사성이 임계점 α 보다 큰 노드가 검색되면 새로운 노드를 생성하지 않고 유사성이 큰 노드를 갱신한다.

표 1. 층 4에서 구조 조정 알고리즘

Table. 1 Self-structured algorithm in 4th layer

IF $M(m_i, \sigma_i) > \alpha$

유사한 경우	조건 (1)
매개변수 수정을 한다.	
$\sigma_i(t+1) = \sigma_{i-new}$	
$m_i(t+1) = m_{i-new}$	
만약 유사성이 더 큰 노드가 존재한다면 다른 노드를 선택하여 매개변수를 수정한다.	
$\sigma_{i-closest}(t+1) = \sigma_{i-new}$	
$m_{i-closest}(t+1) = m_{i-new}$	

ELSE IF $M(m_{i-closest}, \sigma_{i-closest}) \neq M(m_i, \sigma_i)$

유사하지 않은 경우	조건 (2)
새로운 노드 생성	
$\sigma_{i-new\ node}(t+1) = \sigma_{i-new}$	
$m_{i-new\ node}(t+1) = m_{i-new}$	
만약 $M(m_{i-new}, \sigma_{i-new})$ 가 큰 다른 노드가 존재	
새로운 노드 생성 대신 유사성이 큰 노드를 수정한다.	
$\sigma_{i-closest}(t+1) = \sigma_{i-new}$	
$m_{i-closest}(t+1) = m_{i-new}$	

END

노드의 생성과 수정이 끝난 후
생성되거나 수정된 노드는 자신을 제외한 모든 노드에 대해서 서로의 유사성을 검사하여 결합

표 1에서 조건 (2)에 의해 추가된 노드는 기존 노드와 같은 연결 구조로 연결한다. 마지막으로 생성이나 수정이 된 노드는 자신을 제외한 모든 노드와의 유사성을 검사하여 결합한다.

층 2의 노드들도 유사성을 이용하여 노드의 매개변수를 갱신한다. 노드의 유사성이 노드 생성 임계점 α 보다 큰 경우에만 매개변수의 갱신이 표 2에서 보여진다. 그렇지 않은 경우 매개변수의 변화는 일어나지 않는다. 일반적인 역전파 학습 방법을 이용하면 갑작스런 외란이나 제어 시스템의 초기 상태를 그대로 학습하게 된다. 노드의 유사성을 이용한 매개변수 학습은 이런 단점을 보완한다.

표 2. 층 2에서 학습 알고리즘

Table. 2 Learning algorithm in second layer

IF $M(m_i, \sigma_i) > \alpha$

유사한 경우
$\sigma_i(t+1) = \sigma_{i-new}$
$m_i(t+1) = m_{i-new}$

ELSE

유사하지 않은 경우
노드의 매개변수를 수정하지 않는다.

퍼지 제어기가 전방향으로 출력동작을 진행할 때 층4의 추가된 노드가 있는 경우 층 3에서 출력된 퍼지 규칙 값이 추가된 노드와 연결될지 결정해야 한다. 여기서 우리는 입력 생성 임계점(input threshold) β 을 설정한다. 만약 노드의 입력이 β 보다 큰 경우는 추가된 노드로 연결된다. β 보다 작은 경우는 원래의 노드로 연결된다. 그리고 퍼지 신

경망이 역전파 학습을 진행할 때 층 4의 노드에 새로 추가된 노드가 있다면 학습시킬 노드도 표 3의 방법을 사용한다.

표 3. 층 4의 노드 선택 방법

Table. 3 Node selection method in 4th layer

IF $\alpha_3^i \geq \beta$

추가된 노드를 학습하거나 연결

ELSE

기존 노드를 학습하거나 연결

III. 모의 실험 및 고찰

제안된 퍼지 신경망 알고리즘에 대한 모의 실험이 2 자유도 구륜 이동 로봇에 대하여 수행되었으며 적용된 2 자유도 로봇이 그림 2에서 보여진다. 이륜 구동 로봇의 기구학 방정식은 식 (15)과 같다.

$$\begin{pmatrix} \omega_R \\ \omega_L \end{pmatrix} = \frac{1}{R(I_b^2 + 1)} \begin{pmatrix} -I_a I_b & -I_b^2 - 1 & -I_a \\ I_a I_b & -I_b^2 - 1 & I_a \end{pmatrix} \begin{pmatrix} V_x^b \\ V_y^b \\ V_\theta^b \end{pmatrix} \quad (15)$$

입력 V_x^b, V_y^b, V_θ^b 는 로봇 몸체의 속도이다. 출력 $\omega_{left}, \omega_{Right}$ 는 로봇 몸체의 바퀴속도이다. 사용하는 기준 데이터는 로봇이 놓인 전체 좌표에서의 속도로 표시되어 있다. 따라서 전체 좌표 속도를 로봇 몸체의 속도로 식 (16)에서 변환한 뒤 입력한다.

$$\begin{pmatrix} V_x^b \\ V_y^b \\ V_\theta^b \end{pmatrix} = \begin{pmatrix} \cos^f \theta_b & \sin^f \theta_b & 0 \\ -\sin^f \theta_b & \cos^f \theta_b & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x^f \\ V_y^f \\ V_\theta^f \end{pmatrix} \quad (16)$$

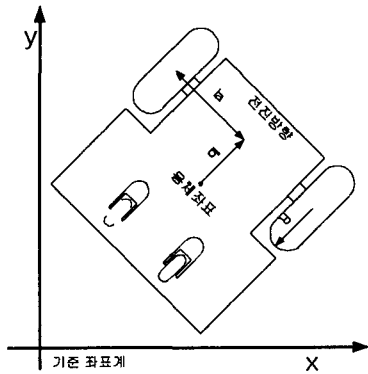


그림 2. 2자유도 구륜 이동 로봇의 구조
Fig. 2 Structure of 2 D.O.F. Wheeled Mobile Robot

로봇의 기준궤적의 초기위치는 (0, 1, -90°)이며, 최종위치는 (16.47, 1, -90°)이다. 그리고 총 주행시간은 60초이며 각각 5초 가속, 50초 등속, 5초 감속하였다.

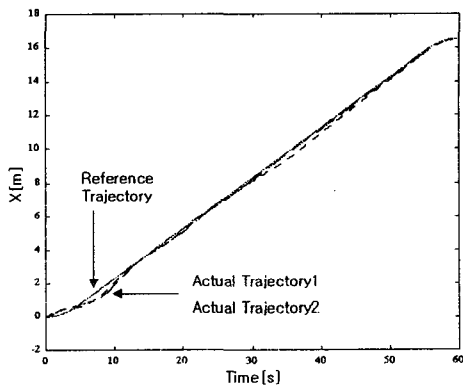


그림 3. x축 기준궤적에 대한 로봇의 실제궤적
Fig. 3 Real trajectory of robot along x-reference trajectory

시간의 변화에 따라 x축 기준궤적에 대한 이동 로봇의 실제궤적의 수렴 결과가 그림 3에서 보여진다. 본 논문에서의 두 제어기 모두 x축 기준궤적에 대해 수렴함을 알 수 있다.

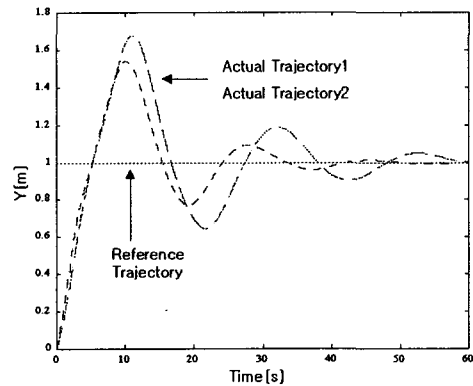


그림 4. y축 기준궤적에 대한 로봇의 실제궤적
Fig. 4 Real trajectory of robot along y-reference trajectory

y축 기준궤적에 대한 수렴 결과가 그림 4에서 보여진다. 10초와 15초 사이에서 과도 응답이 1.67m에서 1.53m로 줄었으며 상승시간도 다소 향상되었음을 알 수 있다.

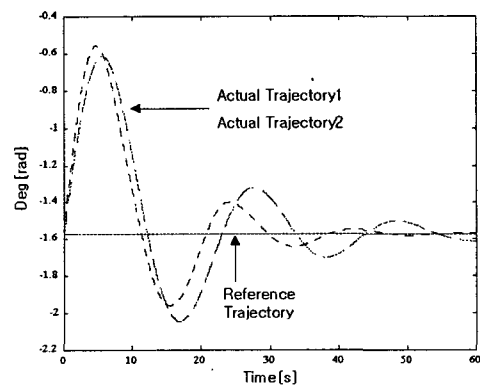


그림 5. 기준궤적에 대한 로봇의 방향각
Fig. 5 Orientation angle of robot along the reference trajectory

이동 로봇의 방향각 θ 의 기준궤적에 대한 수렴결과가 그림 5에서 보여진다. 방향각 θ 의 기준궤적에 대한 수렴이 상승시간이 짧아졌기 때문에 y축 기준궤적에 대한 상승시간이 짧아졌다. 15초 이내에 과도응답이 더 높게 나타났으나 이로 인해 상승시간을 줄였고 15초 이후 방향각 θ 의 정착시간도 적고 과도응답도 감소하였다.

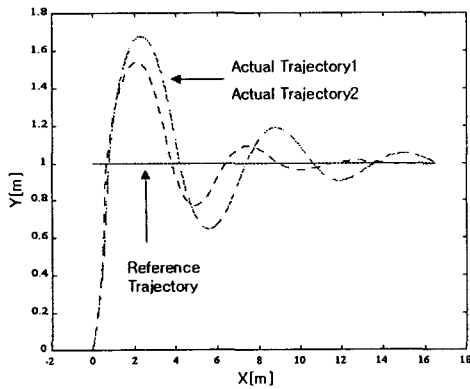


그림 6. 직선 기준궤적에 대한 로봇의 실제궤적
Fig. 6 Actual trajectory of robot along line reference trajectory

기준 좌표계에서 이동 로봇의 위치는 그림 6에서 보여진다. x축 좌표에서 1m이동할 때 퍼지 신경망이 y축으로 빠른 상승 속도를 보이고 있다. x축으로 1m부터 16.47m까지 이동하는 동안 퍼지 신경망의 y축 동작에서 과도응답과 정착시간이 향상되었다. 본 논문에서는 x축과 y축에 독립된 2개의 퍼지 신경망을 사용하고 있다. 그림 7과 그림 8은 각각의 퍼지 신경망이 60초 동안 실시간 학습시킨 각 퍼지 신경망에서 총 4의 노드들의 결과를 보여준다.

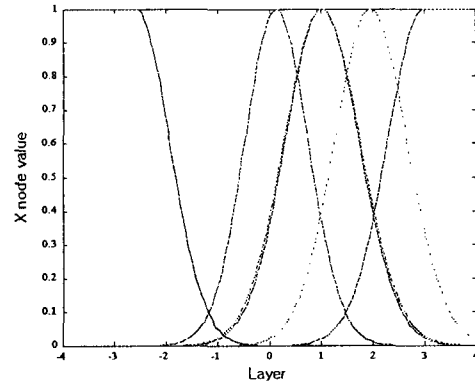


그림 7. x축 출력 노드의 변화

Fig. 7 Value of output node along x-axis

그림 7은 x축을 제어하는 퍼지 신경망의 총 4의 노드들이다. 시작할 때 노드는 총 7개였으며, -3, -2, -1, 0, 1, 2, 3에 노드의 중앙값이 배치되어 있었다. 모의 실험은 노드의 최소 개수를 7개로 정의해 놓았기 때문에 최소 7개의 노드는 존재한다. 그림 7에서 두 개의 노드가 겹쳐져서 한 개의 노드처럼 구성되어 있다. 본 퍼지 신경망의 x축의 총 4의 노드는 5개로 구성되면 충분하다는 것을 보여준다.

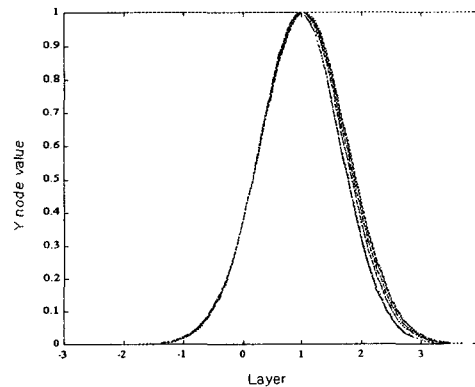


그림 8. y축 출력 노드의 변화

Fig. 8 Value of output node along y-axis

그림 8은 y축의 퍼지 신경망의 층 4의 노드들의 60초 동안 학습시킨 결과이다. 시작할 때 노드의 수와 초기 위치는 x축과 y축이 동일하다. 모의 실험에서 최소 노드의 수를 7개로 설정했다. 60초 동안 학습이 이루어지면서 y축의 기준 궤적 1m에 수렴하기 위해서 노드의 중앙값들이 1을 중심으로 모여있다. 그림 8에서 1이 실제 y축 궤적의 1m에 대응함을 알 수 있다. 제안된 퍼지 신경망은 y축에 대한 층 4의 노드는 3개로 구성되면 충분하다는 것을 보여준다.

IV. 결론

본 논문에서는 2 자유도 구륜 이동 로봇의 제어를 위한 퍼지 신경망의 구조와 그 학습 방법을 제시하였다. 제안된 퍼지 신경망의 구조는 구륜 이동 로봇의 x축, y축의 위치오차와 오차변화율을 입력으로 사용하며 각 위치오차를 보상하는 독립된 2개의 제어기를 구성하였다. 각각의 제어기의 입력은 축의 위치오차와 오차변화율로 하고 유사성을 이용한 학습 방법을 채택하여 시스템의 최적성을 높였다. 이의 실용성을 증명하기 위해 2 자유도 구륜 이동 로봇에 대한 모의실험 결과 제안된 퍼지 신경망의 구조를 바탕으로 유사성을 이용한 학습 방법이 실시간 자기 구조 조정 및 매개변수 조정을 실행하였음을 확인하였으며 퍼지 신경망 제어기의 과도응답이 작고 정상상태에서 수렴속도도 향상되었으며, 실시간 학습으로 인하여 실험 시간이 단축되었음을 확인하였다. 유사성을 이용한 학습 방법은 실시간 학습 결과를 보이기

때문에 제어기가 종종 지역 국소점에 정착하여 빠져나오지 못하는 경우가 있다. 이것은 신경망의 노드들이 학습이 진행되는 동안에 한곳으로 집중하거나 노드들이 전체적으로 한 방향으로 이동하기 때문이다. 향후 이런 동작결과들을 좀더 자세하게 분석하여 향상되고 최적화된 실시간 퍼지 신경망 제어기의 설계에 대한 연구가 진행되어야 할 것이다.

참고문헌

- [1] Spyros Raptis and Spyros Tzafestas, "A Blueprint for a Genetic Meta-Algorithm", EUFIT '98, September 7-10, 1998
- [2] Joao Pedro Pedroso, "Niche Search: an Application in Vehicle Routing", IEEE 1998
- [3] Chin-chih Hsu, Shin-ichi Yamada, "A Multi-Operator Self-tuning Genetic Algorithm for Fuzzy Control Rule Optimization", IEEE 1996
- [4] Henning Heider and Thorsten Drabe, "Fuzzy System Design with a Cascaded Genetic Algorithm", IEEE 1997
- [5] Gregory V. Tan and Xiheng HU, "On Designing Fuzzy Controllers using Genetic Algorithms", IEEE 1996
- [6] Kevin M. Passino and Stephen Yurkovich, "Fuzzy Control", Addison Wesley, pp. 21-103
- [7] D. Driankov and M. Reinfrank, "An

-
- Introduction to fuzzy control",
Springer 2nd Edition, pp 103-141
- [8] David E. Goldberg, "Genetic Algorithms
in Search Optimization & Machine
Learning", Addison Wesley, pp 60-
88
- [9] Zbigniew Michalewicz, "Genetic Algo-
rithms + Data Structures = Evo-
lution Programs", Springer-Verlag,
pp 95-172

Position Control of Wheeled Mobile Robot using Self-Structured Neural Network Model

Ki-Yeoul, Kim*/Sung-Hoe, Kim**/Hyun, Kim***/Ho, Lim****/Young-Hwa, Jeong*****

Abstract

A self-structured neural network algorithm that finds optimal fuzzy membership functions and rule base to fuzzy model is proposed and a fuzzy-neural network controller is designed to get more accurate position and velocity control of wheeled mobile robot. This procedure that is composed of three steps has its own unique process at each step. The elements of output term set are increased at first step and then the rule base is varied according to increase of the elements. The adjusted controller is in competition with controller which doesn't include any increased elements. The adjusted controller will be removed if the control-law lost. Otherwise, the controller is replaced with the adjusted system. After finished regulation of output term set and rule base, searching for input membership functions is processed with constraints and fine tuning of output membership functions is done.

* Dept. of Computer Control, Bucheon College

** Mirae Coporation, R&D Center

*** Dept. of Computer Control, Bucheon College

**** Dept. of Electronic, Sungnam Polytechnic College

***** School of Electronics & Information and Communication, Namseoul University