

이동 로봇의 경로 추종을 위한 웨이블릿 신경 회로망 기반 일반형 예측 제어에 관한 연구

A Study on Wavelet Neural Network Based Generalized Predictive Control for Path Tracking of Mobile Robots

송용태*, 오준섭**, 박진배**, 최윤호***

Yong Tae Song*, Joon Seop Oh**, Jin Bae Park** and Yoon Ho Choi***

*울산시 울주군 삼남면 가천리 818 삼성 전자 SDI 기술지원본부 생산기술팀

*Technology Support Division Production Engineering Team, SAMSUNG SDI CO.,LTD, 818
Kachun-Ri, Samnam-Myun, Ulju-Gun, Ulsan, 689-701

**서울시 서대문구 신촌동 134 연세대학교 전기전자 공학과

**Dept. of Electrical & Electronic Eng., Yonsei University, 134 Shinchon-dong, Seodaemun-ku,
Seoul, 120-749

***경기도 수원시 영통구 이의동 산 94-6 경기대학교 전자공학부

***School of Electronic Eng., Kyonggi University, San 94-6 Yui-dong, Yongtong-ku, Suwon,
Kyonggi-Do, 442-760,

요 약

본 논문에서는 다중 입·출력을 갖는 이동 로봇의 경로 추종을 위해 웨이블릿 신경 회로망에 기반한 예측 제어 방법을 제안한다. 제안된 방법에서 상태 예측기로는 학습 능력이 뛰어난 신경 회로망의 특성 및 웨이블릿 분해의 특성을 합성한 웨이블릿 신경 회로망을 사용한다. 예측기는 경사 하강법을 사용하여 웨이블릿 신경회로망의 출력에 대한 실제 이동 로봇의 상태 오차를 최소화하도록 학습된다. 또한 이동 로봇의 제어 신호인 직진 속도 및 각속도는 추종하고자 하는 기준 경로에 대한 이동 로봇의 예측 상태 오차를 이용하여 정의된 비용 함수를 최소화하도록 구해진다. 컴퓨터 모의 실험에서 변화되는 기준 경로에 대한 경로 추종 성능을 통해 제안한 예측 제어 시스템의 적용 가능성 및 효율성을 보인다.

핵심어: 이동 로봇, 경로 추종, 웨이블릿 신경 회로망, 예측 제어

Abstract

In this paper, we propose a wavelet neural network(WNN) based predictive control method for path tracking of mobile robots with multi-input and multi-output. In our control method, we use a WNN as a state predictor which combines the capability of artificial neural networks in learning processes and the capability of wavelet decomposition. A WNN predictor is tuned to minimize errors between the WNN outputs and the states of mobile robot using the gradient descent rule. And control signals, linear velocity and angular velocity, are calculated to minimize the predefined cost function using errors between the reference states and the predicted states. Through a computer simulation for the tracking performance according to varied track, we demonstrate the efficiency and the feasibility of our predictive control system.

Keyword: Mobile Robot, Path tracking, Wavelet neural network, Predictive control

1. 서 론

로봇 기술이 급속히 발전함에 따라 로봇은 산업 현장에서 단순 반복 작업을 하는 기계의 이미지에서 탈피하여 생활 속

에서 인간과 친숙한 휴먼 로봇이나 인간에게 오락이나 서비스 등을 제공하는 개인용 지능 로봇으로 발전해 가고 있다. 이러한 다양한 지능 로봇들이 가져야 할 가장 중요한 기능들 중의 하나가 작업 공간 내의 자율 이동성(autonomous mobility)이며 이를 위한 다양한 제어 기법들이 연구되어 왔다. 특히 자율 이동성을 위한 정확한 경로 추종 문제의 해결을 위해 PID 제어, 슬라이딩 모드(sliding mode) 제어, 예측 제

접수일자 : 2004년 12월 1일
완료일자 : 2005년 7월 8일

어, 강인 제어 등과 같은 제어 이론들을 기반으로 한 이동 로봇의 다양한 경로 추종 제어 기법들이 연구되어 왔다[1]-[4]. 한편 비선형 시스템의 복잡성, 정확한 수학적 모델의 필요성 및 다양한 불확실성 등의 문제 해결을 위해 지능 제어의 관점에서 신경 회로망[5][6], 퍼지 이론 및 유전 알고리즘 등에 기반한 제어 방법들이 소개되었고 이동 로봇의 추종 제어에 적용되어 왔다. 그러나 신경 회로망 제어기의 경우 제어 능력 부족 및 국부 최솟치(local minima)에 빠질 가능성 등과 같은 단점을 가지며, 퍼지 제어기의 경우 환경 변화에 따른 퍼지 규칙을 얻거나 동조하기 힘든 단점 등을 가진다. 이와 같은 단점들을 개선하기 위해 퍼지 신경망, 뉴로 퍼지 및 생물학적인 학습 방법 등을 이용한 다양한 지능 제어 방법들이 제안되었지만, 여전히 변하는 환경에서의 실시간 제어 능력이 부족한 단점을 가진다. 한편 최근에 시간과 주파수 영역에서 해석 능력이 우수한 웨이블릿 변환(wavelet transform: WT)과 신경 회로망을 결합한 웨이블릿 신경 회로망(wavelet neural network: WNN)이 제안되었고, 이를 이용하여 시스템의 모델링, 신호 예측 및 제어에 적용하고자 하는 연구가 활발히 진행되고 있다[7].

따라서 본 논문에서는 웨이블릿 이론 및 신경 회로망의 장점을 결합한 WNN을 이용하여 기존의 단일 입·출력 시스템 예측에 대한 이론을 바탕으로 다중 입·출력을 갖는 비선형 이동 로봇의 경로 추정 제어에 적합한 일반형 예측 제어(generalized predictive control: GPC) 시스템을 설계하는 방법을 제안하고자 한다. 본 논문에서 설계하고자 하는 WNN 기반 일반형 예측 제어 시스템에서, WNN은 이동 로봇의 상태인 좌표 및 진행 방향의 예측을 위한 예측기(predictor)로 사용하고, 예측기로부터의 예측된 상태를 이용하여 이동 로봇의 제어 입력인 직진 속도 및 각속도를 생성하는 제어기를 구성한다. 아울러 WNN 기반 예측기는 이동 로봇의 과거 상태 및 과거 제어 입력으로 구성함으로써 보다 정확한 이동 로봇의 상태 예측이 가능하게 하고, 이동 로봇의 제어 입력을 위한 예측 제어기는 WNN으로부터의 예측 상태 및 이동 로봇의 실제 출력을 이용하여 추종하고자 하는 경로와의 오차로 정의되는 성능 지표 함수를 최소화시킬 수 있도록 설계한다. 한편 설계된 WNN의 파라미터들은 대표적인 학습 방법인 경사 하강(gradient descent: GD)법을 통해 빠르게 동정한다. 본 논문은 2장에서 WNN의 구조 및 학습 방법에 대해 서술한다. 3장에서는 이동 로봇 모델과 이동 로봇의 경로 추종을 위한 WNN 기반 일반형 예측 제어 시스템 설계 방법에 대해 기술하며, 4장에서는 모의 실험을 통해 이동 로봇의 경로 추정 문제에 대한 WNN 기반 일반형 예측 제어 시스템의 효율성 및 적용 가능성을 검증하고, 마지막 5장의 결론으로 끝맺는다.

2. 웨이블릿 신경 회로망

2.1 웨이블릿 프레임 및 웨이블릿 네트워크

웨이블릿 네트워크는 정적 모델링의 구조에서 처음 연구되어 지금까지 다양한 분야에 적용되어져 왔다[8]-[12]. 일반적으로, 웨이블릿 함수의 성질은 다음과 같은 특성을 갖는다. $L^2(R)$ 공간의 모든 함수는 한정된 웨이블릿의 합으로 근사화 되어질 수 있으며 따라서 웨이블릿 네트워크는 신경망 또는 방사 기저 함수 신경망으로 고려되어질 수 있다. 한편, 웨이블릿 프레임은 모체 웨이블릿으로 불리는 고정 함수에 대

하여 변형 인자 및 확장인자와 같은 단순 연산에 의해 구성되어 진다.

모체 웨이블릿 $\phi_j(x)$ 로부터 웨이블릿 함수 $\phi(z_j)$ 를 구하는 관계는 식 (2.1)과 같다.

$$\phi_j(x) = \phi\left(\frac{x - m_j}{d_j}\right) = \phi(z_j) \quad (2.1)$$

여기서, m_j 및 d_j 는 각각 변형 인자 및 확장 인자로 R_+ 공간의 실수 값을 가진다. 웨이블릿 ϕ 에 의해 구성되는 함수들의 집합은 다음과 같이 정의 된다.

$$\Omega_c = \frac{1}{\sqrt{d_j}} \phi\left(\frac{x - m_j}{d_j}\right), m_j \in R \quad \text{and} \quad d_j \in R_+^*$$

또한 집합 Ω_c 가 $L^2(R)$ 공간에서 웨이블릿 프레임이 되기 위해서는 식 (2.2)를 만족해야 하며, 이러한 웨이블릿 프레임의 집합은 $L^2(R)$ 공간에서의 전역 근사화기로 사용된다[13].

$$A \|f\|^2 \leq \sum_{j, \phi_j \in \Omega_c} |\langle \phi_j, f \rangle|^2 \leq B \|f\|^2. \quad (2.2)$$

여기서, $A > 0$ 과 $B < +\infty$ 인 조건을 만족해야 하며, $\|f\|$ 및 $\langle f, g \rangle$ 은 각각 함수 f 의 놈과 함수 f 와 g 의 내적을 나타낸다. 한편, 다변수 프로세서에 대한 모델링으로 다차원의 웨이블릿이 정의될 수 있다. 다차원 웨이블릿 함수를 단일 차원의 웨이블릿 함수의 텐서 곱(tensor product)으로 나타내면 식 (2.3)과 같다.

$$\phi(x) = \phi_1(x_1) \cdots \phi_n(x_n) \quad (2.3)$$

여기서, 단일 차원의 웨이블릿 변환은 n 개의 직교 방향의 요소로 분리 되어진다고 가정하면 각 항의 푸리에 변환은 식 (2.4)와 같다.

$$\tilde{\phi}(w) = \tilde{\phi}_1(w_1) \cdots \tilde{\phi}_n(w_n) \quad (2.4)$$

여기서, $\tilde{\phi}(w)$ 은 웨이블릿 $\phi(x)$ 의 푸리에 변환이며, 식 (2.5)의 조건을 만족해야 한다.

$$\int_{-\infty}^{\infty} \frac{|\phi_i(w_i)|}{w_i} dw_i < \infty \quad (2.5)$$

감쇄 조건 하에서 음과 양의 무한대에서 0으로 수렴하는 $\phi_i(x_i)$ 에 대하여 식 (2.6)을 만족한다면 식 (2.5)를 만족시킬 수 있다.

$$\int_{-\infty}^{\infty} \phi_i(x_i) dx_i = 0 \quad (2.6)$$

따라서, 식 (2.2)와 식 (2.6)을 만족하는 $\phi_i(x_i)$ 를 웨이블릿 프레임으로 설정한다. 본 논문에서는 두 조건을 만족하는 식 (2.7)과 같은 가우시안(Gaussian) 확률 밀도 함수의 1차 미분 함수를 모체 웨이블릿 함수로 선정한다.

$$\phi(z) = -z \exp\left(-\frac{1}{2} z^2\right) \quad (2.7)$$

또한, 식 (2.8)과 같은 N_j 개의 스칼라 웨이블릿의 곱으로 다차원 웨이블릿을 구성한다.

$$\Phi_j(x) = \prod_{k=1}^{N_j} \phi(z_{jk}) \quad \text{with} \quad z_{jk} = \frac{x - m_{jk}}{d_{jk}} \quad (2.8)$$

2.2 웨이블릿 신경 회로망

WNN에서 웨이블릿 함수를 결정하는 방법에는 크게 세 가지가 있다. 첫 번째 방법은 직교 웨이블릿 분해 이론으로부터 적절한 모체 웨이블릿(mother wavelet)을 선택하고, 이때 결정되는 m 과 d 를 고정시키고 단지 웨이블릿 노드와 출력 노드 사이의 가중치만을 학습하여 갱신하는 방법이다[11]. 두 번째 방법으로는 데이터의 공간 주파수 해석 방식에 따라 결정된 m 과 d 를 가지고 웨이블릿 노드와 출력 노드간의 가중치를 학습 및 갱신하는 방법으로서, 이것은 첫 번째 방식과 달리 반드시 직교성을 가질 필요가 없다[16][17]. 마지막 방법은 m 과 d 를 학습 및 갱신하는 방법이다[18]. 본 논문에서는 미지의 시스템에 적용 가능성을 보이기 위해 마지막 방법을 이용하여 시스템 모델에 대한 예측과 제어를 수행한다.

2.2.1 웨이블릿 신경 회로망의 구조

그림 2.1은 N_i 개의 입력과 단일 출력 및 웨이블릿 층으로 구성된 WNN의 구조를 나타낸다. 그림 2.1에서 A 층은 WNN의 입력을 나타내며, 입력은 벡터 $z = [x_1, \dots, x_{N_i}]^T$ 로 표현된다. 또한 B 층은 웨이블릿 함수 층으로서 각 노드에 대한 웨이블릿 함수는 식 (2.9)와 같이 정의된다.

$$\phi_j(z) = \phi\left(\frac{x - m_j}{d_j}\right) \quad (2.9)$$

여기서 m_j 와 d_j 는 각각 j 번째 웨이블릿 노드의 변형 인자 및 확장 인자로서 실수값을 가지며, $d_j > 0$ 이다. 또한 모체 웨이블릿은 식 (2.10)과 같은 가우시안(Gaussian) 함수의 1차 미분형으로 Haar 모체 웨이블릿의 변형으로 통합 근사화(universal approximation property) 특성을 갖는다[19].

$$\phi(z) = -z \exp\left(-\frac{1}{2}z^2\right) \quad (2.10)$$

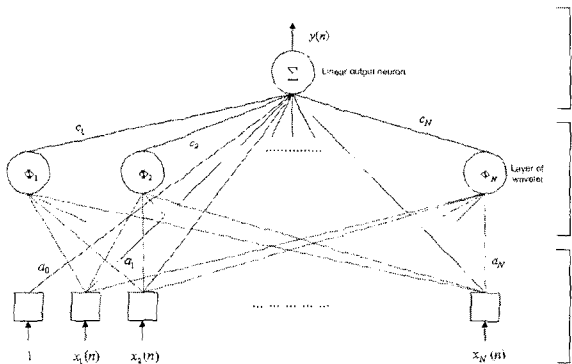


그림 2.1 웨이블릿 신경 회로망의 구조
Fig. 2.1 Structure of wavelet neural network

B 층의 노드의 각 웨이블릿은 식 (2.11)과 같이 각 입력에 대한 모체 웨이블릿의 곱으로 표현된다.

$$\Phi_j(z) = \prod_{k=1}^{N_j} \phi(z_{jk}) \quad (2.11)$$

여기서, $z_{jk} = \frac{x_k - m_{jk}}{d_{jk}}$ 이고, $k = 1, \dots, N_i$, $j = 1, \dots, N_w$ 이며, N_i 는 입력의 수와 웨이블릿 차원을, N_w 는 웨이블릿 노드의 웨이블릿 수이다.

그림 2.1에서 WNN의 출력 값은 식 (2.12)와 같이 입력, 웨이블릿 및 각 파라미터들로 구성된다.

$$y = \Psi(x, \theta) = \sum_{j=1}^{N_w} c_j \Phi_j(x) + a_0 + \sum_{k=1}^{N_i} a_k x_k \quad (2.12)$$

여기서, a_0 와 a_k 는 입력 노드와 출력 노드 사이의 가중치(weight), c_j 는 웨이블릿 노드와 출력 노드 사이의 가중치이고 θ 는 학습을 통하여 갱신되는 WNN의 파라미터 집합으로 다음과 같다.

$$\theta = a_0, a_k, c_j, m_{jk}, d_{jk} \quad (2.13)$$

2.2.2 웨이블릿 신경 회로망의 학습 방법

WNN의 학습은 식 (2.14)와 같은 비용 함수를 최소화하는 파라미터 집합 θ 를 구하는 것이다.

$$J(\theta(n)) = \frac{1}{2} (y_r(n) - y(n))^2 = \frac{1}{2} e^2(n) \quad (2.14)$$

여기서, $y(n)$ 은 n 번째 WNN의 출력이며, $y_r(n)$ 은 $y(n)$ 의 기준 출력 값이다. 비용 함수의 최소화 과정은 파라미터 집합 θ 를 식 (2.15)와 같은 경사 하강법을 이용하여 반복적으로 학습함으로써 구해진다.

$$\theta(n+1) = \theta(n) - \Delta\theta(n) = \theta(n) - \eta \frac{\partial J(\theta(n))}{\partial \theta(n)} \quad (2.15)$$

여기서, η 는 WNN의 모델 예측에 사용되는 학습률(learning rate)을 나타내고, $\theta(n)$ 은 n 번째 파라미터 집합을 나타낸다. $\frac{\partial J(\theta(n))}{\partial \theta(n)}$ 은 $\theta(n)$ 에 대한 비용 함수의 기울기로서 식 (2.16)에 의해 계산된다.

$$\frac{\partial J(\theta(n))}{\partial \theta(n)} = -e(n) \frac{\partial y(n)}{\partial \theta(n)} \quad (2.16)$$

식 (2.16)에서 $\theta(n)$ 에 대한 $y(n)$ 의 기울기 $\frac{\partial y(n)}{\partial \theta(n)}$ 는 출력을 각각의 파라미터에 대해 미분하여 식 (2.17)에서 식 (2.20)과 같이 구해진다.

- 파라미터 a_0 :

$$\frac{\partial y(n)}{\partial a_0} = 1 \quad (2.17)$$

- 직접 연결 파라미터 a_k :

$$\frac{\partial y(n)}{\partial a_k} = x_k \quad (2.18)$$

- 연결 하중 파라미터 c_j :

$$\frac{\partial y(n)}{\partial c_j} = \Phi_j(x) \quad (2.19)$$

- 변형 인자 파라미터 m_{jk} :

$$\frac{\partial y(n)}{\partial m_{jk}} = -\frac{c_j}{d_{jk}} \frac{\partial \Phi_j(x)}{\partial z_{jk}}, \quad k=1, \dots, N_i; j=1, \dots, N_w \quad (2.20)$$

여기서, $\frac{\partial \Phi_j(x)}{\partial z_{jk}} = \phi(z_{j1})\phi(z_{j2})\dots\phi(z_{jk})\dots\phi(z_{jN_i})$ 이며, $\phi(z_{jk})$ 은 z_{jk} 에서의 스칼라 모체 웨이블릿 $\phi(z_{jk})$ 에 대한 z_{jk} 의 미분값을 의미하며 식 (2.21)과 같이 표현된다.

$$\phi(z_{jk}) = \frac{d\phi(z_{jk})}{dz_{jk}} = (z_{jk}^2 - 1) \exp\left(-\frac{1}{2} z_{jk}^2\right) \quad (2.21)$$

변형 인자 m_{jk} 의 학습에 대한 식 (2.20)은 식 (2.21)을 이용하여 식 (2.22)와 같이 정의 할 수 있다.

$$\begin{aligned} \frac{\partial y(n)}{\partial m_{jk}(n)} &= -\frac{c_j}{d_{jk}} \frac{\partial \Phi_j}{\partial z_{jk}} = -\frac{c_j}{d_{jk}} \phi(z_{j1})\phi(z_{j2})\dots\phi(z_{jk})\dots\phi(z_{jN_i}) \\ &= -\frac{c_j}{d_{jk}} \left[\left\{ -\exp\left[-\frac{1}{2} \left(\frac{x-z_{j1}}{d_{j1}}\right)^2\right] \right\} \left\{ -\exp\left[-\frac{1}{2} \left(\frac{x-z_{j2}}{d_{j2}}\right)^2\right] \right\} \dots \right. \\ &\quad \left. \left\{ (z_{jk}^2 - 1) \exp\left(-\frac{1}{2} z_{jk}^2\right) \right\} \dots \left\{ -\exp\left[-\frac{1}{2} \left(\frac{x-z_{jN_i}}{d_{jN_i}}\right)^2\right] \right\} \right] \end{aligned} \quad (2.22)$$

• 파라미터 d_{jk} :

$$\frac{\partial y(n)}{\partial d_{jk}} = -\frac{c_j}{d_{jk}} z_{jk} \frac{\partial \Phi_j(x)}{\partial z_{jk}}, \quad k=1, \dots, N_i; \quad j=1, \dots, N_w \quad (2.23)$$

각 파라미터 벡터는 식 (2.17)에서 식 (2.20)의 과정을 반복적으로 수행하여 식 (2.15)에 의해 학습 및 갱신된다.

3. 웨이블릿 신경 회로망 기반 예측 제어 시스템 설계

3.1 이동 로봇 모델

본 논문에서 사용한 이동 로봇은 두 개의 구동륜 및 네 개의 캐스터(caster)로 구성되며, 이동 로봇의 위치는 두 개의 구동륜 사이의 중심 좌표에 의해 표현되는 3 자유도로 고려된다. 또한 이동 로봇의 방위각은 그림 3.1과 같이 표현되며, 이동 로봇의 동적 방정식은 식 (3.1)과 같다.

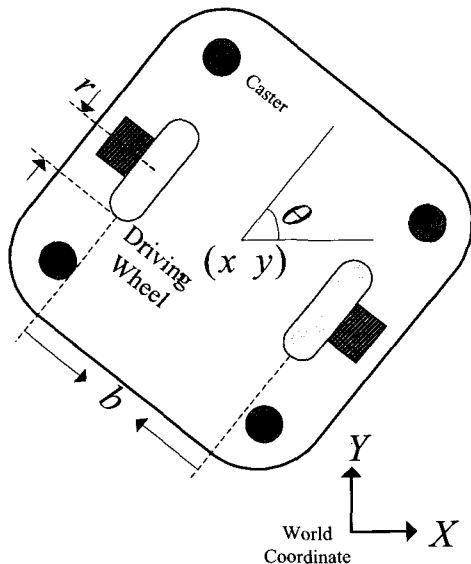


그림 3.1 이동 로봇 모델 및 절대좌표

Fig. 3.1 Mobile robot model and absolute coordinate

$$\begin{aligned} \begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} &= \begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} + \begin{pmatrix} \delta d_n \cos\left(\theta_n + \frac{\delta \theta_n}{2}\right) \\ \delta d_n \sin\left(\theta_n + \frac{\delta \theta_n}{2}\right) \\ \delta \theta_n \end{pmatrix} \\ \delta d_n &= \frac{d_r + d_l}{2} \quad \delta \theta_n = \frac{d_r - d_l}{b} \end{aligned} \quad (3.1)$$

여기서, x_n, y_n 및 θ_n 은 이동 로봇의 현재 상태를 나타내며, δd_n 및 $\delta \theta_n$ 은 직진 속도 및 각속도를 나타낸다. 또한 이동 로봇의 모델에서 제어 입력은 $u_n = [\delta d_n, \delta \theta_n]^T$ 로 정의된다. 제어 입력 u_n 을 위한 d_r, d_l 및 b 는 각각 좌·우 구동륜의 변위 및 구동륜 사이의 거리를 나타낸다.

3.2 웨이블릿 신경 회로망에 기반한 예측 제어 시스템

WNN에 기반한 이동 로봇 시스템의 예측 제어에 사용되는 적응 제어 방법은 직접 적응 제어(direct adaptive control) 방법과 간접 적응 제어(indirect adaptive control) 방법으로 구분될 수 있다. 간접 적응 제어 방법은 시스템에 대한 예측 모델을 오프 라인(off-line) 상에서 우선 생성하고, 생성된 모델을 이용하여 제어 대상의 공정을 온 라인(on-line)으로 제어하는 방법이며, 예측의 정확성은 양호하나 동정 모델을 얻기 위해 많은 학습 시간이 소요된다는 단점을 가진다. 반면에, 직접 적응제어 방법은 제어기 파라미터가 제어 대상에 대한 오프 라인 예측 과정 없이 시스템 출력과 모델 출력 사이의 오차에 의해 온 라인으로 직접 갱신된다. 본 논문에서는 직접 적응제어 방법에 기반한 일반형 예측 제어 기법을 이용하여 다중 입·출력을 갖는 이동 로봇을 위해 웨이블릿 신경망 기반 일반형 예측 제어 설계 방법을 제안한다.

3.2.1 웨이블릿 신경 회로망을 이용한 예측기 설계

본 논문에서 제안한 WNN 구조를 이용한 다중 입력과 다중 출력을 갖는 이동 로봇의 상태 예측은 식 (3.1)과 같은 시스템 출력 $Y(n) = [x_n \ y_n \ \theta_n]^T$ 과 WNN 출력 $\Upsilon(n)$ 사이의 오차를 이용하여 각 파라미터들을 학습시킴으로써 주어진 시스템을 예측한다. 그림 3.2는 예측 오차를 이용하여 각 파라미터를 학습시키기 위한 WNN 기반 예측기의 구조를 나타낸다.

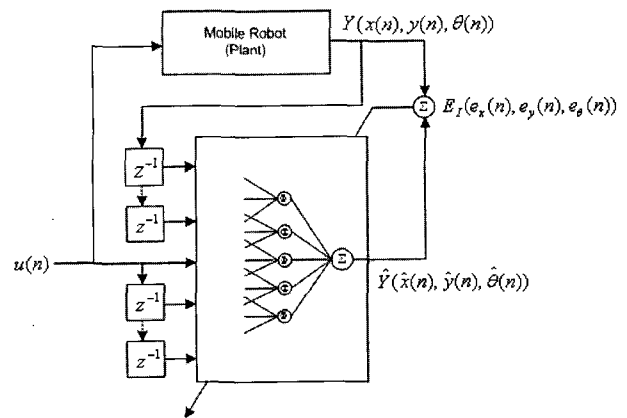


그림 3.2 시스템 예측을 위한 웨이블릿 신경 회로망의 구조
Fig. 3.2 Structure of wavelet neural network for system prediction

WNN의 출력은 식 (3.2)와 같이 표현될 수 있다.

$$\Upsilon(n) = f(Y(n-1), Y(n-2), \dots, Y(n-N_s), u(n-2), u(n-1), \dots, u(n-N_e)) \quad (3.2)$$

여기서 $\Upsilon(n)$ 은 n 번째 WNN의 출력이고, $f(\cdot)$ 는 실제 시스템의 비선형 함수이며, N_s 와 N_e 는 각각 WNN의 상태 입력의 수와 제어 입력의 수를 의미한다. 한편 WNN의 입력 x 는 식 (3.3)과 같다.

$$x = [Y(n-1), Y(n-2), \dots, Y(n-N_s), u(n-2), u(n-1), \dots, u(n-N_e)]^T \quad (3.3)$$

여기서 이동 로봇 시스템의 출력과 WNN 출력의 차인 예측 오차 $E_I(n)$ 은 2장에서 서술한 바와 같이 반복적인 경사하강법을 통해 WNN의 파라미터 값을 조정하는데 사용된다.

3.2.2 웨이블릿 신경 회로망 기반 일반형 예측 제어기의 설계

본 논문에서 제안한 WNN 기반 예측 제어 시스템의 구조

는 그림 3.3과 같다. 그림에서 기준 출력과 WNN 출력의 차로 정의되는 $E_C(e_x, e_y, e_\theta)$ 은 제어기의 파라미터를 조정하여 제어 오차를 최소화 하며, 이와 같이 제어 오차를 최소화함으로써 시스템 출력 $Y(x, y, \theta)$ 가 기준 신호 $Y_r(x_r, y_r, \theta_r)$ 을 추종하도록 한다. 플랜트가 다중 입·출력 시스템일 경우 식 (3.1)과 같은 플랜트의 입출력 관계를 고려하여 그림 3.3과 같이 웨이블릿 신경망을 구성한다. 그림 3.3에서의 웨이블릿 신경회로망의 출력인 \hat{x} , \hat{y} 및 $\hat{\theta}$ 값은 식 (3.4)에서 식 (3.6)까지의 식들로 나타낼 수 있다.

$$\hat{x}(n) = f_x(x(n-1), \theta(n-1), x(n-2), \theta(n-2), \dots, x(n-N_s), \theta(n-N_s), u_{\delta d}(n-1), u_{\delta \theta}(n-1), u_{\delta d}(n-2), u_{\delta \theta}(n-2), \dots, u_{\delta d}(n-N_e), u_{\delta \theta}(n-N_e)) \quad (3.4)$$

$$\hat{y}(n) = f_y(y(n-1), \theta(n-1), y(n-2), \theta(n-2), \dots, y(n-N_s), \theta(n-N_s), u_{\delta d}(n-1), u_{\delta \theta}(n-1), u_{\delta d}(n-2), u_{\delta \theta}(n-2), \dots, u_{\delta d}(n-N_e), u_{\delta \theta}(n-N_e)) \quad (3.5)$$

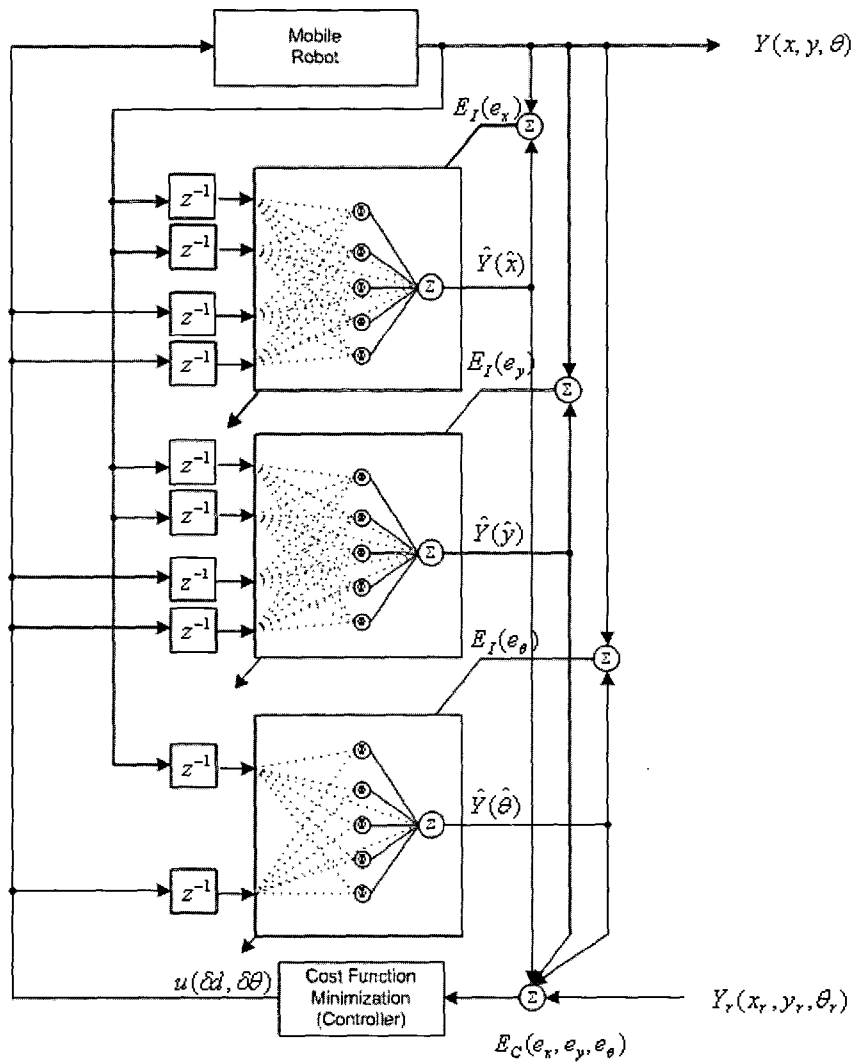


그림 3.3. 웨이블릿 신경 회로망 기반 예측제어 시스템
Fig. 3.3 Wavelet neural network based predictive control system

$$\begin{aligned} \mathcal{V}(n) &= f_{\theta}\{\theta(n-1), \theta(n-2), \dots, \\ &\theta(n-N_s), u_{\delta\theta}(n-1), \dots, u_{\delta\theta}(n-N_e)\} \end{aligned} \quad (3.6)$$

여기서 $\hat{x}(n)$, $\hat{y}(n)$, $\mathcal{V}(n)$ 은 n 번째 WNN의 출력이고 각 $f_{x,y,\theta}\{\cdot\}$ 는 실제 시스템의 비선형 함수이며 N_s 와 N_e 는 각각 웨이블릿 신경회로망의 상태 입력의 수와 제어 입력의 수를 나타낸다. 일반형 예측 제어기의 설계를 위한 비용 함수는 식 (3.7)과 같이 정의한다.

$$J = \frac{1}{2}(E_c^2(n+1) + \lambda \Delta u^2(n)) \quad (3.7)$$

식 (3.7)의 $\Delta u(n)$ 은 제어 입력의 변화치를 나타내고, $E_c(n+1)$ 은 제어 오차로서 식 (3.8)과 같이 정의한다.

$$E_c(n+1) = \mathcal{Y}(n+1) - \mathcal{X}(n+1) \quad (3.8)$$

여기서 $\mathcal{Y}_x(n+1)$ 은 기준 신호이다.

WNN을 통과한 시스템 출력은 식 (3.9)과 같이 얻을 수 있다.

$$\mathcal{Y} = \sum_{j=1}^{N_r} c_j \Phi_j(\vec{x}) + a_0 + \sum_{k=1}^{N_i} a_k x_k \quad (3.9)$$

여기서 $\Phi_j(\vec{x}) = \prod_{k=1}^{N_i} \phi(z_{jk})$ 이고, $z_{jk} = \frac{x_k - m_{jk}}{d_{jk}}$ 이다. 이 때, $k=1, \dots, N_i$ 이며 $j=1, \dots, N_r$ 이다.

식 (3.7)의 비용 함수 J 를 최소화 하기 위해서 제어 신호 $u(n)$ 은 경사 하강법에 의해 반복적으로 계산되며, 식 (3.10)에 의해 갱신된다.

$$\begin{aligned} u_{\delta x}(n+1) &= u_{\delta x}(n) - \eta \frac{\partial J}{\partial u_{\delta x}(n)} \\ u_{\delta \theta}(n+1) &= u_{\delta \theta}(n) - \eta \frac{\partial J}{\partial u_{\delta \theta}(n)} \end{aligned} \quad (3.10)$$

여기서, $\eta > 0$ 은 학습률로서 WNN 출력 $\mathcal{Y}(n)$ 을 기준 신호 $\mathcal{Y}_x(n)$ 으로 점근적으로 근사화 시키는데 사용된다.

식 (3.1)과 같은 플랜트 모델에 의해 비용 함수 J 는 직진 속도의 제어 신호 $u_{\delta x}(n)$ 와 각속도의 제어 신호 $u_{\delta \theta}(n)$ 에 의해 결정된다. 식 (3.10)에서의 $\frac{\partial J}{\partial u_{\delta x}(n)}$, $\frac{\partial J}{\partial u_{\delta \theta}(n)}$ 은 제어 신호 $u_{\delta x}(n)$, $u_{\delta \theta}(n)$ 에 대한 비용 함수의 편미분으로 식 (3.11)과 (3.12)에 의해 구해진다.

$$\begin{aligned} \frac{\partial J}{\partial u_{\delta x}(n)} &= [(\hat{x}(n+1) - x_x(n+1)) \frac{\partial \mathcal{Y}_x(n+1)}{\partial u_{\delta x}(n)} \\ &+ (\hat{y}(n+1) - y_x(n+1)) \frac{\partial \mathcal{Y}_y(n+1)}{\partial u_{\delta x}(n)}] + \lambda \Delta u_{\delta x}(n) \end{aligned} \quad (3.11)$$

$$\frac{\partial J}{\partial u_{\delta \theta}(n)} = [(\mathcal{V}(n+1) - \theta_y(n+1)) \frac{\partial \mathcal{Y}_{\theta}(n+1)}{\partial u_{\delta \theta}(n)}] + \lambda \Delta u_{\delta \theta}(n) \quad (3.12)$$

또한 식 (3.11)과 (3.12)의 $\Delta u_{\delta x}(n)$ 및 $\Delta u_{\delta \theta}(n)$ 은 각 제어 입력의 변화값으로 각각 $\Delta u_{\delta x}(n) = u_{\delta x}(n) - u_{\delta x}(n-1)$ 및 $\Delta u_{\delta \theta}(n) = u_{\delta \theta}(n) - u_{\delta \theta}(n-1)$ 로 나타내지며,

$\frac{\partial \mathcal{Y}_x(n+1)}{\partial u_{\delta x}(n)}$, $\frac{\partial \mathcal{Y}_y(n+1)}{\partial u_{\delta x}(n)}$, $\frac{\partial \mathcal{Y}_{\theta}(n+1)}{\partial u_{\delta \theta}(n)}$ 은 각 제어 신호 $u_{\delta x}(n)$, $u_{\delta \theta}(n)$ 에 대한 WNN의 기울기로서 식 (3.13), (3.14), (3.15)로 각기 나타낼 수 있다.

$$\begin{aligned} \frac{\partial \mathcal{Y}_x(n+1)}{\partial u_{\delta x}(n)} &= \sum_{j=1}^{N_r} c_{x_j} \frac{\partial \Phi_{x_j}(x)}{\partial u_{\delta x}(n)} + a_{x_k} \mid_{k=N_s+1} \\ &= \sum_{j=1}^{N_r} \frac{c_{x_j}}{d_{x_{jk}}} \frac{\partial \Phi_{x_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta x}(n)} + a_{x_k} \mid_{k=N_s+1} \end{aligned} \quad (3.13)$$

$$\begin{aligned} \frac{\partial \mathcal{Y}_y(n+1)}{\partial u_{\delta x}(n)} &= \sum_{j=1}^{N_r} c_{y_j} \frac{\partial \Phi_{y_j}(x)}{\partial u_{\delta x}(n)} + a_{y_k} \mid_{k=N_s+1} \\ &= \sum_{j=1}^{N_r} \frac{c_{y_j}}{d_{y_{jk}}} \frac{\partial \Phi_{y_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta x}(n)} + a_{y_k} \mid_{k=N_s+1} \end{aligned} \quad (3.14)$$

$$\begin{aligned} \frac{\partial \mathcal{Y}_{\theta}(n+1)}{\partial u_{\delta \theta}(n)} &= \sum_{j=1}^{N_r} c_{\theta_j} \frac{\partial \Phi_{\theta_j}(x)}{\partial u_{\delta \theta}(n)} + a_{\theta_k} \mid_{k=N_s+1} \\ &= \sum_{j=1}^{N_r} \frac{c_{\theta_j}}{d_{\theta_{jk}}} \frac{\partial \Phi_{\theta_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta \theta}(n)} + a_{\theta_k} \mid_{k=N_s+1} \end{aligned} \quad (3.15)$$

여기서, $\frac{\partial x}{\partial u(n)} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ 이다. 따라서 제어 신호 $u_{\delta x}(n)$, $u_{\delta \theta}(n)$ 은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} u_{\delta x}(n+1) &= u_{\delta x}(n) - \eta \frac{\partial J}{\partial u_{\delta x}(n)} \\ &= u_{\delta x}(n) - \eta [(\hat{x}(n+1) - x(n+1)) \\ &\quad \left[\sum_{j=1}^{N_r} \frac{c_{x_j}}{d_{x_{jk}}} \frac{\partial \Phi_{x_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta x}(n)} + a_{x_k} \mid_{k=N_s+1} \right] \\ &\quad + [(\hat{y}(n+1) - y(n+1)) \\ &\quad \left[\sum_{j=1}^{N_r} \frac{c_{y_j}}{d_{y_{jk}}} \frac{\partial \Phi_{y_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta x}(n)} + a_{y_k} \mid_{k=N_s+1} \right] + \lambda \Delta u_{\delta x}(n)] \end{aligned} \quad (3.16)$$

$$\begin{aligned} u_{\delta \theta}(n+1) &= u_{\delta \theta}(n) - \eta \frac{\partial J}{\partial u_{\delta \theta}(n)} \\ &= u_{\delta \theta}(n) - \eta [(\mathcal{V}(n+1) - \theta(n+1)) \\ &\quad \left[\sum_{j=1}^{N_r} \frac{c_{\theta_j}}{d_{\theta_{jk}}} \frac{\partial \Phi_{\theta_j}(x)}{\partial z_{jk}} \frac{\partial x}{\partial u_{\delta \theta}(n)} + a_{\theta_k} \mid_{k=N_s+1} \right] + \lambda \Delta u_{\delta \theta}(n)] \end{aligned} \quad (3.17)$$

식 (3.1)의 플랜트 모델에 따라 제어 신호의 관계를 수식적으로 연관성이 있는 입·출력끼리 구분하여 구성하고, 제어 신호를 생성시키는 제어기 또한 모델 식의 제어 입력에 따라 구분한다. 따라서 식 (3.16)과 식 (3.17)의 $\frac{\partial x}{\partial u(n)}$ 도 각각의 웨이블릿 신경망 입력 벡터를 X_x , X_y , X_{θ} 라고 하면 N_s 가 2일 때 식 (3.18)과 같이 쓸 수 있다.

$$\begin{aligned} \frac{\partial x_x}{\partial u_{\delta x}} &= [0 \ 0 \ 1 \ 0]^T, \frac{\partial x_y}{\partial u_{\delta x}} = [0 \ 0 \ 1 \ 0]^T, \frac{\partial x_{\theta}}{\partial u_{\delta x}} = [0 \ 0]^T \\ \frac{\partial x_x}{\partial u_{\delta \theta}} &= [0 \ 0 \ 0 \ 0]^T, \frac{\partial x_y}{\partial u_{\delta \theta}} = [0 \ 0 \ 0 \ 0]^T, \frac{\partial x_{\theta}}{\partial u_{\delta \theta}} = [0 \ 1]^T \end{aligned} \quad (3.18)$$

이상은 단 단계 예측 제어에 사용되는 제어 신호 학습에 관한 알고리즘으로 일반형 예측 제어 기법 이론에 의해 다

단계 예측 제어로 확장 가능하다. 다 단계 예측을 위한 일반형 예측 제어기의 제어 신호를 정의하기 위해 기준 신호 $R_{N|n}$ 시스템 출력 $\mathcal{Y}_{N|n}$ 및 제어 오차 $E_{N|n}$ 은 식 (3.19), (3.20), (3.21)과 같이 각기 나타낼 수 있다.

$$R_{N|n} = [r_{n+1}, r_{n+2}, \dots, r_{n+N}]^T \quad (3.19)$$

$$\mathcal{Y}_{N|n} = [\hat{y}_{n+1}, \hat{y}_{n+2}, \dots, \hat{y}_{n+N}]^T \quad (3.20)$$

$$E_{N|n} = [e_{n+1}, e_{n+2}, \dots, e_{n+N}]^T \quad (3.21)$$

식 (3.19)의 기준 신호 벡터 및 식 (3.20)의 WNN 출력 벡터와 두 벡터의 오차 벡터인 식 (3.21)으로부터 식 (3.22)와 같이 제어 신호 벡터를 정의할 수 있다.

$$U_{N|n} = [u_{n+1}, u_{n+2}, \dots, u_{n+N}]^T \quad (3.22)$$

식 (3.7)의 비용 함수 J 는 식 (3.23)로 다시 정의 될 수 있다.

$$J = \frac{1}{2} [E_{N|n}^T E_{N|n} + \lambda \Delta U_{N|n}^2] \quad (3.23)$$

즉, 제어의 목적은 비용 함수 J 를 최소화 시키는 제어 신호 $U_{N|n}$ 을 구하는 것이며, 제어 신호 벡터 $U_{N|n}$ 은 경사 하강법을 통해 식 (3.24)에 의해 갱신된다.

$$U_{N|n} = U_{N|n} - \eta D_{N|n} \quad (3.24)$$

식 (3.24)의 $D_{N|n}$ 는 식 (3.25)과 같다.

$$D_{N|n} = \frac{\partial J}{\partial U_{N|n}} = \frac{\partial \mathcal{Y}_{N|n}}{\partial U_{N|n}} E_{N|n} + \lambda M \Delta U_{N|n} \quad (3.25)$$

여기서 λ 는 제어 신호의 가중 인자(weighting factor)이다. 또한 M 은 제어 입력 $U_{N|n}$ 의 시간별 차이를 나타내는 값으로 식 (3.26)과 같이 나타낼 수 있다.

$$M = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \vdots \\ \vdots & 0 & \ddots & -1 & 0 \\ \vdots & \vdots & 0 & 1 & -1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (3.26)$$

식 (3.25)의 $\frac{\partial \mathcal{Y}_{N|n}}{\partial U_{N|n}}$ 는 식 (3.27)로 나타낼 수 있다.

$$\frac{\partial \mathcal{Y}_{N|n}}{\partial U_{N|n}} = \begin{bmatrix} \frac{\partial \hat{y}_{n+1}}{\partial u_n} & \frac{\partial \hat{y}_{n+2}}{\partial u_n} & \dots & \frac{\partial \hat{y}_{n+N}}{\partial u_n} \\ 0 & \frac{\partial \hat{y}_{n+2}}{\partial u_{n+1}} & \dots & \frac{\partial \hat{y}_{n+N}}{\partial u_{n+1}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \hat{y}_{n+N}}{\partial u_{n+N-1}} \end{bmatrix} \quad (3.27)$$

여기서, $\frac{\partial \mathcal{Y}_{N|n}}{\partial U_{N|n}}$ 는 WNN 출력 \mathcal{Y}_{Mn} 의 제어 신호 U_{Mn} 에 대한 편미분이다. 본 논문에서 사용된 WNN에 기반한 이동 로봇 시스템의 예측 제어 과정을 단계별로 나타내면 표 3.1와 같다.

표 3.1 웨이블릿 신경 회로망 기반 예측 제어 시스템 과정
Table. 3.1 Procedure for wavelet neural network based predictive control system

단계 1	WNN과 제어기의 파라미터를 초기화 한다. $a_0, a_k, c_j, m_{jk}, d_{jk} = 1, a_k, c_j, m_{jk}, d_{jk} = random,$ $u(n) = 0$
단계 2	이동 로봇의 출력 $Y(n)$ 과 WNN의 출력 $\mathcal{Y}(n)$ 의 차로 정의된 예측 오차 $E_f(n)$ 을 계산한다. $E_f(n) = Y(n) - \mathcal{Y}(n)$
단계 3	예측 오차 E_f 를 이용하여 WNN의 각 파라미터를 경사 하강법을 통하여 학습시킨다. $\theta(n+1) = \theta(n) - \Delta\theta(n) = \theta(n) - \eta \frac{\partial J}{\partial \theta(n)}$
단계 4	WNN의 출력 $\mathcal{Y}(n)$ 과 원하는 기준 신호의 출력 $Y_r(n)$ 사이의 오차인 제어 오차 $E_c(n)$ 을 계산한다. $E_c(n) = \mathcal{Y}(n) - Y_r(n)$
단계 5	제어 오차를 이용하여 이동 로봇의 제어 입력을 경사 하강법으로 갱신한다. $u_{\delta d}(n+1) = u_{\delta d}(n) - \eta \frac{\partial J}{\partial u_{\delta d}(n)}$ $u_{\delta \theta}(n+1) = u_{\delta \theta}(n) - \eta \frac{\partial J}{\partial u_{\delta \theta}(n)}$
단계 6	이동 로봇의 주행 시간 동안 단계 2에서 단계 6까지를 반복 수행한다.

4. 모의 실험 및 결과 고찰

본 논문의 모의 실험에서는 제어 대상으로 위치 값으로 x, y, θ 로 표현되는 좌표와 진행 방향을 출력으로 하고 제어 입력으로는 직진 속도 δd 와 각속도 $\delta \theta$ 를 갖는 이동 로봇의 정확한 경로 추종을 위해 WNN 기반한 예측 제어 시스템을 구성하여 시뮬레이션 함으로써 제안된 WNN 기반 예측 제어 방법이 다중 입·출력을 갖는 시스템에 효율적으로 적용 가능함을 보이고자 한다.

4.1 모의 실험 환경

일반적으로 신경망 모델의 특징은 신경망의 파라미터 초기치, 노드 수 및 학습률 등의 실험 환경에 영향을 받는다. 따라서 모의 실험 환경은 로봇 궤적 설정과 예측기의 파라미터를 동일하게 설정하고, 초기 연결 강도는 랜덤(random)하게 발생하였다. 실험을 위한 이동 로봇 경로는 이동 로봇의 움직임을 총 20초로 하여 5초씩 간의 구간별 제어 입력인 직진 속도 및 각속도를 표 4.1과 같이 설정하였다. 또한 WNN은 출력 \hat{y} 의 경우 $x, \theta, u_{\delta d}, u_{\delta \theta}$ 의 입력을, 출력 \hat{y} 는 $x, y, u_{\delta d}, u_{\delta \theta}$ 의 입력을, θ 의 경우 $\theta, u_{\delta \theta}$ 의 입력을 WNN의 입력 층에 연결하고, 각각 한 주기의 플랜트 과거 값으로 예측기를 구성하였다. 은닉 층에는 각각 5개씩의 웨이블릿 노드를 구성하였다. 그리고 샘플링 주기는 이동 로봇의 움직임을 비교적 상세히 파악할 수 있도록 0.01초로 하였다. 표 4.2는 모의실험을 위한 초기 파라미터들을 나타낸다.

표 4.1 이동 로봇의 경로를 위한 기준 제어 입력
Table 4.1 Reference control input for the reference path of a mobile robot

구간	직진 속도(δd)	각속도($\delta \theta$)
$0 < t \leq 5$	0.1m/sec	0 °/sec
$5 < t \leq 10$	0.2m/sec	5.93 °/sec
$10 < t \leq 15$	0.2m/sec	-5.93 °/sec
$15 < t \leq 20$	0.1m/sec	-0.593 °/sec

표 4.2 모의 실험을 위한 초기 파라미터
Table 4.2 Initial parameters for simulation

Number of wavelet function	5
Number of past state	1
Number of past input	1
Sampling time	0.01

4.2 모의 실험 결과 및 고찰

이동 로봇의 기준 신호에 대한 WNN의 예측 오차는 표 4.3에서와 같이 x , y , θ 가 각각 0.0089cm, 0.1178cm, 0.3094 °로 양호한 수준의 평균 제곱 오차(mean square error: MSE)를 보여준다. 또한 그림 4.1에서와 같이 이동 로봇의 비선형적인 궤적을 상당히 정확하게 예측함을 알 수 있다. 그림 4.2는 이동 로봇의 기준 신호와 WNN의 예측 오차를 보여주며 시간에 따라 오차가 점점 감소하여 0으로 수렴함을 알 수 있다. 또한 그림 4.2의 결과로부터 기준 제어 입력이 변하는 부분, 즉 속도나 방향이 변하는 부분에서 오차값의 변화가 상대적으로 다른 구간 보다 커졌다가 안정되는 모습을 보이는데, 이는 WNN이 기준 신호의 x , y , θ 값이 변할 때 신경망의 고유 특성인 학습을 통하여 오차값이 줄어들어 안정화 되고, 학습된 WNN은 그 이후의 기준 신호의 변화가 있을 때 오차값을 대폭 감소시키는 것을 볼 수가 있다.

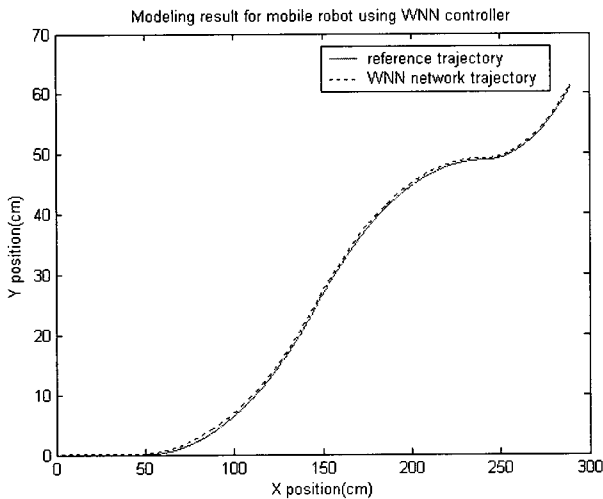


그림 4.1 WNN을 이용한 이동 로봇의 경로 예측 결과
Fig. 4.1 Path prediction result of a mobile robot using WNN

표 4.3 이동 로봇의 경로 예측 오차
Table 4.3 Path prediction errors of a mobile robot

	MSE
x	0.0089 cm
y	0.1178 cm
θ	0.3094 °

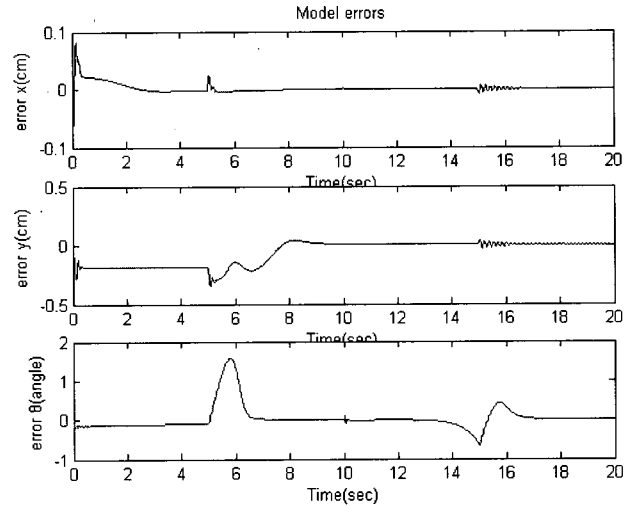


그림 4.2 이동 로봇의 경로 예측 오차
Fig. 4.2 Path prediction errors of a mobile robot

기준 신호에 대한 WNN의 차이를 이용하여 제어 오차 e_x , e_y , e_θ 를 구하고 이 값을 이용하여 경사 하강법으로 제어 신호를 생성시켜 이동 로봇을 제어한 결과는 표 4.4와 같다. 표의 결과에서 나타나듯이 표 4.3에서와 같이 x , y , θ 가 각각 0.3916cm, 0.3090cm, 0.0172 °로 양호한 수준의 평균

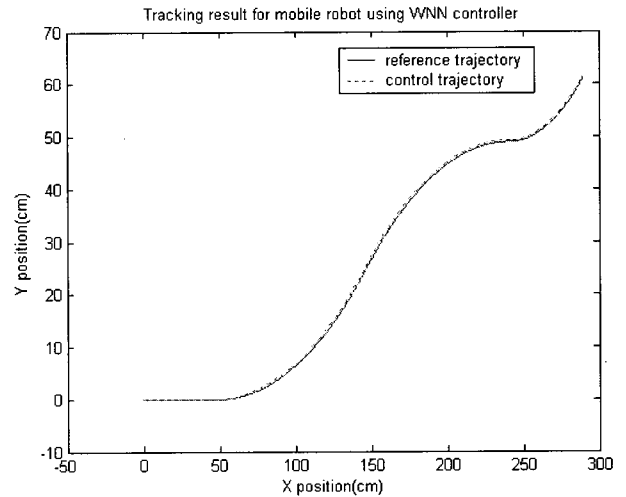


그림 4.3 WNN 기반 예측 제어 시스템을 이용한 이동 로봇의 경로 추종 결과
Fig. 4.3 Path tracking result of a mobile robot using WNN based predictive control system

제곱 오차 대해 모두 양호한 수준의 평균 제곱 오차(MSE)를 보여준다. 또한 그림 4.3에서와 같이 이동 로봇의 비선형적인 궤적을 상당히 정확하게 추종함을 알 수 있다. 그림 4.4는 기준 신호와 웨이블릿 이동 로봇의 위치 오차를 나타내며, 그림의 결과로부터 이동 로봇의 속도와 방향이 변하는 부분에서 오차가 발생하나, 비교적 빠른 시간에 안정화됨을 알 수 있다. 그림 4.5는 이동 로봇의 제어 입력 신호를 나타내는데, 기준 경로를 생성시키는 제어 신호에 근사한 값이 생성됨을 보여준다.

본 논문에서는 웨이블릿 이론 및 신경 회로망의 장점을 결합한 WNN에 기반하여 기존의 단일 입·출력에 대한 시스템의 예측에 대한 이론적 바탕으로 다중 입·출력을 갖는 비선형 이동 로봇의 경로 추종 제어 문제에 적용하였다. 모의실험 결과들로부터, 이동 로봇의 상태 예측 및 추종 제어에서 본 논문에서 제안한 WNN 기반 예측 제어 시스템의 적용 및 양호한 예측 및 제어 성능을 확인할 수 있었다.

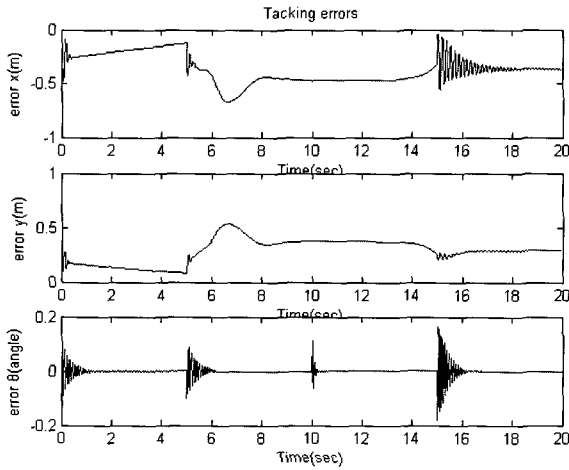


그림 4.4 이동 로봇의 경로 추종 오차
Fig. 4.4 Path tracking errors of a mobile robot

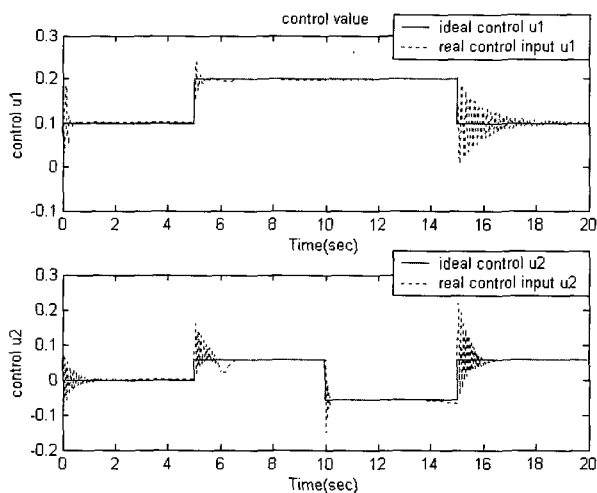


그림 4.5 제어 입력
Fig. 4.5 Control inputs

표 4.4 이동 로봇의 경로 추종 오차
Table 4.4 Path tracking errors of a mobile robot

	MSE
x	0.3916cm
y	0.3090cm
θ	0.0172°

5. 결론

본 논문에서는 다중 입·출력을 갖는 비선형 시스템의 하나인 이동 로봇의 경로 추종 제어를 위하여 WNN을 이용한 예측 제어 시스템의 설계 방법을 제안하였다. 본 논문에서 제안한 제어기 설계 방법에서 모델의 예측을 위하여 웨이블릿과 신경 회로망의 시간 및 주파수 영역의 해석 능력이 뛰어난 학습 능력을 결합한 WNN을 사용하였고, 성능 지표 함수인 비용 함수가 최소가 되도록 경사 하강법을 사용하여 학습함으로써 제안한 예측 제어기의 여러 파라미터들을 온라인으로 예측하여 필요한 제어 신호를 생성하였다. 본 논문에서는 기존의 단일 입·출력 시스템 예측에 대한 이론을 바탕으로, 다중 입·출력을 갖는 이동 로봇에 WNN을 결합하였고, 다중 출력으로부터 다중 제어 신호를 생성하여 이동 로봇의 경로를 추종하게 하였다. 본 논문에서 사용한 이동 로봇의 모델은 3개의 출력 x , y , θ 와 플랜트 제어를 위한 2개의 제어 입력 δd , $\delta \theta$ 로 구성하였다. 3개의 출력 x , y , θ 를 동정하기 위하여 예측기의 내부에 3개의 서로 다른 입력을 갖는 WNN을 구성하였고, 구성된 WNN의 출력 $Y(x)$, $Y(y)$, $Y(\theta)$ 를 플랜트 출력과 비교하여 동정 오차 $E_f(x)$, $E_f(y)$, $E_f(\theta)$ 를 구하였다. 3개의 동정 오차를 경사 하강법을 통하여 WNN을 학습시키고, 학습된 결과를 기준 출력 $Y(x)$, $Y(y)$, $Y(\theta)$ 와 비교하여 3개의 제어 오차 $E_c(x)$, $E_c(y)$, $E_c(\theta)$ 를 계산한 후, 이 값과 출력에 대한 WNN의 기울기를 구하여 이동 로봇을 제어할 직진 속도 $u_{\delta d}$ 와 각 속도 $u_{\delta \theta}$ 를 구하였다.

모의 실험을 통해 본 논문에서 제안된 방법의 적용 가능성과 효율성을 검증하기 위하여 임의로 주어진 경로를 추종하게 한 결과, 이동 로봇의 상태 예측 및 경로 추종 제어에서의 양호한 예측 및 제어 오차를 보여 단일 입·출력뿐만 아니라 다중 입·출력을 갖는 시스템에 대해서도 웨이블릿 신경망 기반 예측 제어 시스템이 적용 가능함을 확인하였다. 또한 설계한 웨이블릿 신경망 기반 예측 제어 시스템에서 예측기는 플랜트의 출력의 개수만큼 WNN의 수를 설정하고, 각각의 WNN의 입력은 플랜트 모델의 관계식의 요소 항목으로 입력을 구성하는 것이 효율적임을 확인하였다.

참고 문헌

[1] R. M. Desantis, "Path-Tracking for Car-like Robots with Single and Double Steering," *IEEE Trans. on Vehicular Technology*, Vol. 44, No. 2, pp. 366-377, 1995.

[2] X. Yang, K. He, M. Guo, and B. Zhang, "An Intelligent Predictive Control Approach to Path Tracking Problem of Autonomous Robot," *Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics*, Vol. 4, pp. 350-355, 1998.

[3] Z. P. Jiang and H. Nijmeijer, "Tracking Control of Mobile Robots: A Case Study in Backstepping," *Automatica*, Vol. 33, No. 7, pp. 1393-1399, 1997.

[4] J. M. Yang and J. H. Kim, "Sliding Mode Motion Control of Nonholonomic Mobile Robots", *IEEE Control Systems*, Vol. 19, No. 2, pp. 15-23, 1990.

[5] M. L. Corradini, G. Ippoliti, S. Longhi and S. Micheline, "Neural Networks Inverse Model Approach for the Tracking Problem of Mobile Robot," *Proc. of RAAD 2000*, pp. 17-22, 2000.

[6] M. L. Corradini, G. Ippoliti, and S. Longhi, "The Tracking Problem of Mobile Robots: Experimental Results Using a Neural Network Approach," *Proc. of WAC*, pp. 33-37, 2000.

[7] Celso de Sousa, Elder Moreira Hemerly, Roberto Kawakami "Adaptive Control for Mobile Robot Using Wavelet Networks," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 32, No. 42, pp. 493-504, 2002

[8] C. S. Burrus, R. A. Gopinath, H. Guo, Introduction to Wavelet and Wavelet Transforms: A Primer, Prentice Hall, 1996.

[9] Y. Meyer, Wavelets: Algorithms and Applications, translated by R. D. SIAM. Philadelphia, PA: SIAM Press, 1993.

[10] I. Daubechies, Ten Lectures on Wavelets, PA: SIAM Press, 1992.

[11] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Transform," *IEEE Trans. on Pattern Anl. Machine Intell.*, Vol. 11, pp. 674-693, 1989.

[12] C. K. Chui, An Introduction to Wavelets. New York, Academic Press, 1992.

[13] Ciftcioglu O., "From Neural to Wavelet Network," *Proc. of NAFIPS '99*, New York, pp. 894-898, 1999.

[14] J. Zhang, G. G. Walter, Y. Miao, and N. W. Lee, "Wavelet Neural Networks for Function Learning," *IEEE Trans. on Signal Processing*, Vol. 43, No. 6, pp. 1485-1497, 1995.

[15] T. Kugarajah and Q. Zhang, "Multidimensional Wavelet Frames," *IEEE Trans. on Neural Network*, Vol. 6, No. 6, pp. 1552-1556, 1995.

[16] Y. C. Pati, P. S. Krishanaparasad, "Analysis and Synthesis of Feedforward Neural Networks using Discrete Affine Wavelet Transformations," *IEEE Trans. on Neural Networks*, Vol. 4, No. 1, pp. 73-85, 1993.

[17] M. Cannon, J. J. E. Slotine, "Space-Frequency Localized Basis Network for Nonlinear System estimation and Control," *Neurocomputing*, Vol. 9, pp. 293-342, 1995.

[18] Y. Oussar, I. Rivals, L. Personnaz, G. Dreyfus, "Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling," *Neurocomputing*, Vol. 20, pp. 173-188, 1998.

[29] Q. Zhang, A. Benvenste "Wavelet Networks," *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, pp. 889-898, 1992.

저 자 소 개

송용태(宋 庸 台)

1991년 : 연세대학교 전기공학과 학사
 2005년 : 연세대학교 전기 전자공학과 석사.
 현재 : 삼성 SDI 재직 중.

Tel : 051-363-7849
 Fax : 055-380-2004
 E-mail : yotasong@samsung.com

오준섭(吳 俊 燮)

1997년 : 경기대학교 전자공학과 학사
 1999년 : 경기대학교 전자공학과 석사
 현재 : 연세대학교 전기 전자 공학과 박사과정 재학 중.

Tel : 02-2123-2773
 Fax : 02-362-4539
 E-mail : jsoh@control.yonsei.ac.kr

박진배(朴 珍 培)

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 전자 공학과 교수.

Tel : 02-2123-2773
 Fax : 02-362-4539
 E-mail : jbpark@yonsei.ac.kr

최윤호(崔 允 浩)

1980년 : 연세대학교 전기공학과 학사
 1982년 : 연세대학교 대학원 전기공학과 석사
 1991년 : 동 대학원 박사. 현재 경기대학교 전자공학부 교수.

Tel : 031-249-9801
 Fax : 02-362-4539
 E-mail : yhchoi@kyonggi.ac.kr