

신경망을 이용한 이동 로봇의 실시간 고속 정밀제어

High Speed Precision Control of Mobile Robot using Neural Network in Real Time

주진화, 이장명
(Jin Hwa Ju, and Jang Myung Lee)

Abstract : In this paper we propose a fast and precise control algorithm for a mobile robot, which aims at the self-tuning control applying two multi-layered neural networks to the structure of computed torque method. Through this algorithm, the nonlinear terms of external disturbance caused by variable task environments and dynamic model errors are estimated and compensated in real time by a long term neural network which has long learning period to extract the non-linearity globally. A short term neural network which has short learning period is also used for determining optimal gains of PID compensator in order to come over the high frequency disturbance which is not known a priori, as well as to maintain the stability. To justify the global effectiveness of this algorithm where each of the long term and short term neural networks has its own functions, simulations are performed. This algorithm can also be utilized to come over the serious shortcoming of neural networks, i.e., inefficiency in real time.

Keywords : mobile robot, long term neural network, short term neural network

I. 서론

이동로봇의 활용은 정보화 산업화와 더불어 그 의의가 더욱 커지고 있는 실정이다. 이동로봇은 non-holonomic 시스템으로 그 위치제어에 있어서 베이스가 고정된 로봇과는 달리 속도 적분 값에 의하여 위치가 결정되어 진다. 따라서, 정밀한 위치 제어를 위하여서는 동적인 상황속에서의 정밀한 속도제어가 필연적으로 요구되어진다. 또한, 이동로봇은 로봇의 환경 즉, 이동 표면과의 끊임없는 접촉이 행해지는 가운데 이동하기 때문에 이동 로봇 자체의 모델링에 의존한 제어 방식만으로는 충분하지 못하다.

종래의 이동로봇의 활용은 단순히 어떤 물체를 이동시켜주는 의미만을 강조하였기에 주어진 경로를 가능한 충돌 없이 또는 최적의 조건으로 따라가는 목적에 초점을 맞추어 연구가 진행되어왔다[1]-[3].

최근에 이동로봇이 다른 작업로봇을 적절한 작업 장소로 이동시켜주고 이동 후에 작업로봇과 협동하면서 작업을 수행하는 잉여로봇에 대한 관심이 고조되면서, 이동 로봇의 동적 모델링에 의한 정밀제어의 필요성에 대하여 연구가 시작되고 있다[14]. 이동로봇의 구조는 직렬로 연결된 다관절 로봇에 비하여 단순하기 때문에 그 동적 모델이 비교적 간략하게 주어진다. 그러나, 이 동적 모델을 바탕으로 동적 제어를 실현하더라도 항상 이동 표면과 접촉하고 있는 이동로봇의 정밀한 제어를 위하여서는 모델링 되지 않은 표면과의 마찰력 및 표면에 존재하는 외란으로 인하여 정밀제어에 상당한 제약이 있다.

이러한 비선형 성분들을 학습 능력을 갖춘 신경망을

사용하여 그 값들을 추정하여 정제환으로 보상해 주고, 또한 그 구조가 간단하면서도 잡음이나 파라미터 변동에 강인한 PID 제어기를 활용한다면 이동로봇의 정밀한 제어가 가능할 것으로 제안한다. 로봇의 제어에 신경망을 적용하는 경우 일반적으로, 로봇을 black box로 보고 이의 모델 즉, 특성을 학습을 통해서 구하여 정밀한 제어를 실현하고자 하는 것이 일반적이다[8]-[12]. 이러한 경우 신경망의 구조가 복잡해질 뿐만 아니라 학습하는데 걸리는 시간의 과다로 인하여 실시간에 적용하기는 어려움이 있다. 또 다른 방향으로는, 복잡하고 시변인 로봇의 모델을 학습하는 대신에 적절한 귀환 제어기를 통하여 우수한 제어 성능을 얻고자 하는 방법도 있다[13]. 이는 비록 신경망의 구조는 간단하여 실시간 실현의 가능성이 높으나 시스템에 대한 정보를 활용하지 않기때문에 그 수렴속도를 높이기 힘들고 정밀한 제어를 위하여서는 만족할 만한 성능을 기대하기 힘들다.

본 논문에서는 고속 정밀제어를 실시간에 실현하기 위하여 이동로봇의 동적 모델을 사용하는 동적 제어기법을 기본으로 하여, 일반적으로 그 비선형 및 시변특성으로 인하여 동적 모델에 포함시키지 않는 구동바퀴와 표면간의 마찰을 쿨롱 마찰과 점성마찰로 모델링하고 이를 통하여 미지의 마찰계수를 실시간에 Long Term Neural Network(LTNN)을 활용하여 학습하였다. 학습된 동적 모델은 정제환되어 요구되는 토크를 계산하여 입력시켜주므로 로봇 시스템은 선형화 되어 외란이 없는 상황에서는 완벽한 Tracking 제어기가 될 수 있다. 그러나, 실제적으로 모델링에서 시 불변으로 가정한 항들의 시변 가능성 및 이동 표면에 상존하는 외란(순간적인 표면 기울기 변화) 등으로 인하여 강인성에 대한 고려가 없는 LTNN만으로는 만족할 만한 성능을 기대하기 힘들다. 이에, 강인한 제어 알고리즘으로 널리 알려진 PID

접수일자 : 1998. 7. 29., 수정완료 : 1998. 12. 29.

주진화, 이장명 : 부산대학교 전자공학과

※ 본 논문은 1998년도 한국학술진흥재단의 자유공모과제 연구비에 의하여 연구되었습니다.

제어기를 활용하여 그 제어 이득을 실시간에 적절히 학습시켜주는 Short Term Neural Network(STNN)의 도입으로 고속 정밀제어를 실시간에 실현하고자 한다.

본 논문은 서론을 포함하여 6개의 절로 구성되며, 2절에서는 이동로봇의 동적 제어방식에 대해 기술하고 그 문제점을 제시한다. 3절에서는 Long term 신경망(LTNN)의 필요성과 구조를, 4절에서는 Short term 신경망(STNN)의 사용목적과 구조에 따른 학습알고리즘에 관해서 설명한다. 5절에서는 제안된 제어기의 성능을 평가하기 위해서 시뮬레이션을 통해 타당성을 검증하고, 6절에서 결론과 더불어 향후 과제를 검토한다.

II. 이동로봇의 강인한 역 동적제어 알고리즘

많은 경우에 이동로봇의 제어를 위하여 선형시스템을 위하여 이론적으로 잘 정립된 PID 제어 알고리즘을 사용하고 있다. 이와 같은 알고리즘은 무 부하나 저속의 동작에서는 안정화되어 있고 만족스럽게 작업을 수행할 수 있지만, 부하가 있을 때나 고속의 동작에서는 관성, centrifugal, coriolis 및 중력 향이 활성화되어 정밀하지 않게 된다. 이러한 비 선형성, coupling 영향을 극복하고 외란에 강인한 제어기를 만들기 위해서 역 동적 제어 알고리즘에 가변 PID 보상기를 결합한 강인한 동적 제어 알고리즘의 필요성이 요구된다. 따라서 이동 로봇의 고속 정밀제어를 위하여 이동 로봇의 동적 모델링을 바탕으로 한 역 동적제어를 실현하고 아울러 가변 PID 보상기를 결합하고자 한다. 이를 위하여, 동적 모델링은 일반적으로 사용되는 Euler-Larange 방정식을 이용하여 구한다. 이때, 마찰 등의 손실에 관련된 항들은 모델링에 포함되지 않는다는 사실을 주목해야 할 것이다.

본 연구에서 사용된 이동로봇은 그림 1의 구조로 되어 있으며, 이 이동로봇의 운동 방정식은 Euler-Lagrange방법에 의해 유도한 후 모타의 이너시어, 마찰력 항들 및 바퀴와 표면과의 마찰을 모델링하여 결합하면 다음과 같이 구해진다.

$$(J_m \gamma^{-1} + M) \ddot{q} + (B_m \gamma^{-1} + F_B) \dot{q} + F_C \cdot \text{sign}(\dot{q}) = \tau \quad (1)$$

(1)에서,

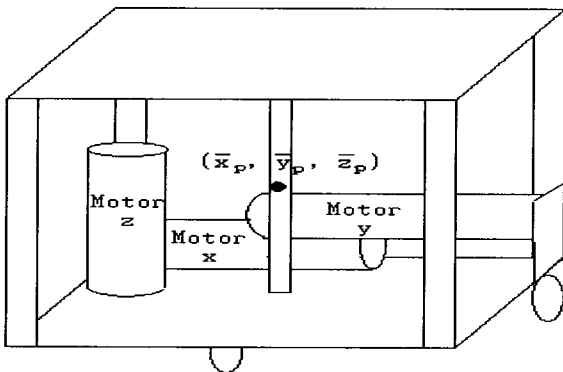


그림 1. 이동 로봇의 구조.
Fig. 1. Structure of a mobile robot.

$$\tau = \begin{bmatrix} \tau^l \\ \tau^r \end{bmatrix},$$

$$M = \begin{bmatrix} \frac{R_m^2}{4} (W + \frac{L_z}{L^2}) & \frac{R_m^2}{4} (W - \frac{L_z}{L^2}) \\ \frac{R_m^2}{4} (W - \frac{L_z}{L^2}) & \frac{R_m^2}{4} (W + \frac{L_z}{L^2}) \end{bmatrix}$$

$\dot{q} = \begin{bmatrix} \dot{q}^l \\ \dot{q}^r \end{bmatrix}$ 이며, R_m 은 모터의 반경을, W 는 이동로봇 전체의 질량을, L 은 이동 로봇의 무게중심과 바퀴 사이의 거리를 나타낸다. 일반적인 이동로봇의 운동 방정식에 모터의 관성행렬, $J_m = \begin{bmatrix} J_m^l & 0 \\ 0 & J_m^r \end{bmatrix}$ 및 점성마찰계수 $B_m = \begin{bmatrix} B_m^l & 0 \\ 0 & B_m^r \end{bmatrix}$ 를 포함시켰으며, 아울러 바퀴와 표면 사이의 Coulomb 마찰계수 $F_C = \begin{bmatrix} F_C^l & 0 \\ 0 & F_C^r \end{bmatrix}$ 및 점성마찰계수 $F_B = \begin{bmatrix} F_B^l & 0 \\ 0 & F_B^r \end{bmatrix}$ 를 포함시킨 것이다. 또한, γ 는 기어 비를 나타내며 위 첨자 l 및 r 은 왼쪽과 오른쪽바퀴를 각각 의미하며, Coulomb의 정지 마찰력과 운동 마찰력은 동일한 것으로 가정하였다.

기저가 고정된 로봇의 경우에는 로봇의 각 관절에서 발생하는 비 선형 마찰 항들이 고속정밀제어에 가장 큰 어려움으로 취급되나, 이동로봇의 경우에는 바퀴와 지표면 사이의 마찰 항들이 가장 큰 장애로 대두된다. 또한, 이동로봇이 운동하는 표면에 놓여있는 작은 장애물들이 이동로봇의 운동에 외란으로 작용되며 상당히 큰 영향을 미친다.

이동로봇의 고속 정밀제어를 위하여 역 동적제어에 의하여, 복잡한 비 선형 동역학 관련 항들을 직접 보상해줄 수 있다. 즉, 정 궤환 토크, τ_f 를 구하면 아래와 같이 구해진다.

$$(J_m \gamma^{-1} + M) \ddot{q}_d + (B_m \gamma^{-1} + \hat{F}_B) \dot{q} + \hat{F}_C \cdot \text{sign}(\dot{q}) = \tau, \quad (2)$$

비선형 성분들을 보상하고 모델링 오차 및 coupling 항들을 외란으로 처리하여 선형의 오차방정식을 유도하기 위하여 계산토크법과 PID 알고리즘의 결합에 의한 제어규칙 u 는 다음과 같이 표시된다.

$$u = (J_m \gamma^{-1} + M) \ddot{q}_d + (B_m \gamma^{-1} + \hat{F}_B) \dot{q} + \hat{F}_C \text{Sign}(\dot{q}) + K_P e + K_I \int e dt + K_D \dot{e} \quad (3)$$

여기서 $K_P = \text{diag}\{K_P^l, K_P^r\}$, $K_I = \text{diag}\{K_I^l, K_I^r\}$, $K_D = \text{diag}\{K_D^l, K_D^r\}$ 를 나타내고, $e = q_d - q$ 으로 정의되며, \wedge 는 추정치를 의미한다. 위의 제어규칙을 블록 선도로 나타내면 그림 2와 같다.

제어규칙 u 와 τ 를 동일하게 두면 (1)과 (3)으로부터 오차 방정식은 다음과 같이 구해진다.

$$(J_m \gamma^{-1} + M) \ddot{e} + (\hat{F}_B - F_B) \dot{q} + (\hat{F}_C - F_C) \cdot \text{sign}(\dot{q}) + K_P e + K_I \int e dt + K_D \dot{e} = 0 \quad (4)$$

순시적으로 $\ddot{q} = \ddot{q}_d$ 라고 가정하면 $\ddot{e} = 0$ 이 되어

$$K_P e + K_I \int e dt + K_D \dot{e} = \eta_0 \quad (5)$$

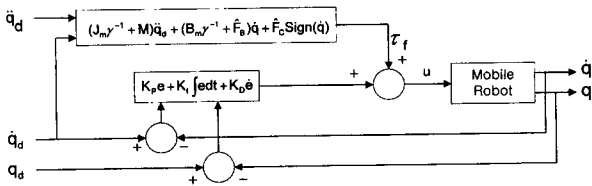


그림 2. 강인한 동적제어 알고리즘.
Fig. 2. Robust dynamic control algorithm.

의 식이 얻어진다. (5)에서 $\eta_0 = (F_B - \hat{F}_B) \dot{q} + (F_C - \hat{F}_C) \cdot \text{sign}(\dot{q})$ 를 나타낸다. 이 오차 방정식은 K_P , K_I , K_D 가 대각 행렬이므로 오차 방정식이 각각의 바퀴변수 ($q^{L,r}$ 및 $q^{R,r}$)에 대해 독립적인 2차의 오차방정식으로 구해지며, K_P , K_I 및 K_D 를 적절히 조절하여 안정도와 오차 특성을 만족시킬 수 있다.

동역학 방정식의 모델링 오차 및 외란 등의 특성으로 인하여 (5)로 표시된 오차방정식에서 그 오차특성이 설계된 PID 이득 값을 일반적으로 정확히 따라가지 않는다. 따라서, 가변 PID 제어를 신경망을 사용하여 설계하여 이동 로봇이 표면 외란에 강인한 특성을 갖도록 할 필요성이 있다. (4)에서 $\tilde{e} = 0$ 로 가정하여 (5)를 구하는 대신에 $\|\tilde{e}\| = \epsilon$ 로 가정한다. 그러면 (4)는 다음과 같이 변환된다.

$$K_P e + K_I \int e dt + K_D \dot{e} = \eta \quad (6)$$

여기서, $\eta = \eta_0 - (J_m \gamma^{-1} + M)\epsilon$, $\epsilon \triangleq [\epsilon_1 \epsilon_2]^T$ 이다.

본 연구에서는 이러한 강인한 역 동적 알고리즘의 수행을 위하여 필연적으로 요구되어 지는 다음의 두 가지 관점에 초점을 두고 문제를 기술한다.

1) 이동로봇이 바퀴와 지면사이에 존재하는 마찰력 항들을 Long term 신경망(LTNN)을 활용하여 학습하도록 하여 $\eta \rightarrow 0$ 의 목적을 달성한다.

2) 이동로봇에 존재하는 외란 즉, 표면의 불 균일성 및 제어시스템에 미치는 순간적인 외란들은 강인한 제어기로 잘 알려진 PID 보상기의 이득을 Short term 신경망(STNN)을 활용하여 동적으로 보정해 주고자 한다.

이를 통하여 이동로봇의 고속정밀제어가 가능할 것으로 제한한다.

III. Long term 신경망

(1)에서 표현된 로봇의 이동에 따른 구동바퀴와 표면 간에 발생하는 비 선형 마찰 항들을 그림 3과 같이 모델링 한다.

여기서, $F(\dot{q})$ 는 로봇이 표면을 따라 이동할 때 발생하는 구동바퀴와 표면간의 마찰 항으로 점성마찰과 Coulomb 마찰의 결합에 의해 표현되며, 점성마찰계수 F_B 와 Coulomb 마찰계수 F_C 를 이용하여 (7)과 같이 표현되며, 이들 값은 표면 상태에 종속되어 그 크기가 결정되어진다.

$$F(\dot{q}) = F_B \dot{q} + F_C \cdot \text{sign}(\dot{q}) \quad (7)$$

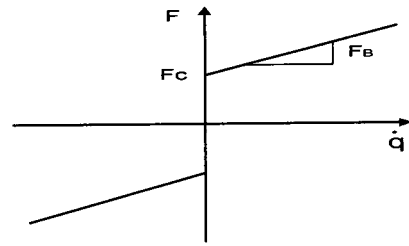


그림 3. 바퀴와 표면과의 모델링.
Fig. 3. Modelling of the interface between a wheel and surface.

LTNN은 (7)로 표현된 비 선형 마찰 모델의 점성 마찰계수 및 Coulomb 마찰계수를 신경망의 학습에 의해 결정함으로써 로봇이 표면을 따라 이동할 때 발생하는 마찰에 의해 야기되는 속도편차를 보상하는 것이다. 여기서는 표면의 상태가 비교적 느리게 변화하는 것으로 가정하고 이동 로봇이 이동하는 동안 발생하는 마찰 항을 비교적 정밀하게 보상할 수 있도록 하기 위해서 신경망의 학습율을 비교적 낮게 유지하도록 한다. 따라서 신경망이 학습되기까지는 비교적 긴 시간이 소요된다. 이러한 의미에서 이 신경망을 Long term 신경망(LTNN)으로 지칭하게 된 것이다. 그러나, 신경망의 구조가 그림 4에서 보인 바와 같이 간단하게 설정하여 수렴속도가 지나치게 길어지는 것을 방지하고 있다. 따라서 LTNN의 학습은 수행속도면 보다는 정밀성에 중점을 둔 학습에 의해 구동바퀴와 표면간의 마찰에 따른 비 선형 항들을 보상해 줌으로써 로봇의 고속 정밀제어가 가능하도록 한다. 이동로봇이 작업 중에 발생하는 표면의 불 균일성 및 제어시스템에 미치는 순간적인 외란 등은 STNN을 사용하여 PID보상기의 이득조정에 의해 실시간으로 보상되어진다. 따라서 LTNN의 입력 층과 은닉 층 그리고 은닉 층과 출력 층간의 각 가중치의 초기 값은 임의의 값으로 설정하였다. 신경망의 입력으로는 이동로봇의 두 개의 바퀴에 가해지는 속도 요구 값, \dot{q}_d^L , \dot{q}_d^R 과 그에 따른 실제 출력 값 즉, 센서로부터 측정된 속도 값 \dot{q}^L , \dot{q}^R 로 하고 신경망의 출력은 이들 입력에 의해 오차 역 전파 학습알고리즘에 의해 학습되어져 추종되어진 마찰 계수 \hat{F}_C^L , \hat{F}_C^R 및 \hat{F}_B^L , \hat{F}_B^R 이다. 또한, 신경망의 은닉 층의 활성화함수로는 마찰력과 같은 실제 모델이 어려운 비 선형 항들을 적절하게 학습할 수 있도록 하기 위하여 시그모이드 함수를 사용하고, 출력 층의 활성화함수는 선형함수를 사용함으로써 입력 값에 따른 추종된 마찰계수 값이 임의의 영역으로 사상(mapping)되는 것이 가능하도록 한다.

이동로봇이 표면상에서 이동하고 있을 때, 마찰력들은 실제 이동 속도와 방향의 함수에 의하여 결정되어지므로, LTNN의 평가 함수를 C_{LTNN} 으로 표현하고 속도 오차를 기초로 다음과 같이 정의한다.

$$C_{LTNN} = (J_m \gamma^{-1} + M) \tilde{e} \quad (8)$$

이제 신경망의 학습 방법 즉, 가중치들의 변화 방법에

대해 살펴본다. 설명의 편의를 위하여, PID 보상기를 일단 고려하지 않고 보면 이동로봇의 입력, u 는 다음과 같다.

$$u = (J_m \gamma^{-1} + M) \ddot{q}_d + (B_m \gamma^{-1} + \hat{F}_B) \dot{q} + \hat{F}_C \cdot \text{sign}(\dot{q}) \quad (9)$$

여기서, \hat{F}_B 와 \hat{F}_C 는 LTNN의 학습에 의해 추정된 점성 마찰계수와 Coulomb 마찰계수이다. (1)의 r 와 u 를 같게 두면

$$0 = (J_m \gamma^{-1} + M) \ddot{e} + (\hat{F}_B - F_B) \dot{q} + (\hat{F}_C - F_C) \text{sign}(\dot{q}) \quad (10)$$

이 된다. 또한 윗식은 다시

$$(J_m \gamma^{-1} + M) \ddot{e} = (F_B - \hat{F}_B) \dot{q} + (F_C - \hat{F}_C) \text{sign}(\dot{q}) \quad (11)$$

로 표현된다.

급강하법을 이용하여 은닉 층에서 출력 층으로의 가중치 조절은 신경망의 수렴속도를 향상시키기 위하여 과거의 가중치 변화를 반영하는 모멘텀항을 도입하여 정리하면 다음과 같다.

$$\Delta W_{kj}(k) = -\eta \frac{\partial C_{LTNN}}{\partial W_{kj}} + \alpha \Delta W_{kj}(k-1) \quad (12)$$

여기서, α 는 모멘텀 상수이고 η 는 학습율을 나타낸다.

윗식에서

$$\begin{aligned} \frac{\partial C_{LTNN}}{\partial W_{kj}} &= \frac{\partial C_{LTNN}}{\partial \hat{F}_B} \frac{\partial \hat{F}_B}{\partial W_{kj}} + \frac{\partial C_{LTNN}}{\partial \hat{F}_C} \frac{\partial \hat{F}_C}{\partial W_{kj}} \quad (13) \\ &= -\dot{q} \frac{\partial \hat{F}_B}{\partial W_{kj}} - \text{sign}(\dot{q}) \frac{\partial \hat{F}_C}{\partial W_{kj}} \end{aligned}$$

로 정의되며, 여기서 $\frac{\partial \hat{F}_B}{\partial W_{kj}}$ 와 $\frac{\partial \hat{F}_C}{\partial W_{kj}}$ 는 연쇄법칙(chain rule)에 의해 다음과 같이 구해진다.

$$\begin{aligned} \frac{\partial \hat{F}_B}{\partial W_{kj}} &= \frac{\partial \hat{F}_B}{\partial (net_k)} \frac{\partial (net_k)}{\partial W_{kj}} \quad (14) \\ \frac{\partial \hat{F}_C}{\partial W_{kj}} &= \frac{\partial \hat{F}_C}{\partial (net_k)} \frac{\partial (net_k)}{\partial W_{kj}} \end{aligned}$$

여기서, $\frac{\partial (net_k)}{\partial W_{kj}}$ 는 은닉 층에서 출력 층 k번째 뉴런으로의 입력 $i_k(k=1,2,\dots,n)$ 이며,

$$\begin{aligned} \frac{\partial \hat{F}_B}{\partial (net_k)} &= f'(net_k) = \frac{1}{2}(1 - \hat{F}_B^2) \quad (15) \\ \frac{\partial \hat{F}_C}{\partial (net_k)} &= f'(net_k) = \frac{1}{2}(1 - \hat{F}_C^2) \end{aligned}$$

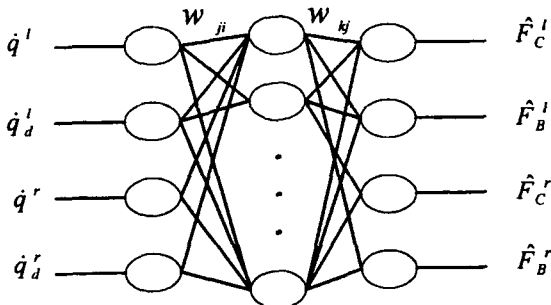


그림 4. LTNN의 구조.
Fig. 4. Structure of LTNN.

이다. (14) 및 (15)에 의해 구해진 $\frac{\partial C_{LTNN}}{\partial W_{kj}}$ 의 값을 (12)에서 사용하여 신경망의 가중치 조절이 학습을 통하여 평가함수를 줄이는 방향으로 진행된다

또한, 은닉 층에서의 가중치 조절은

$$\Delta W_{ji}(k) = -\eta \frac{\partial C_{LTNN}}{\partial W_{ji}} + \alpha \Delta W_{ji}(k-1) \quad (16)$$

에 의해서 (13)-(15) 수식들을 같은 방법으로 적용하여 얻어진다.

IV. Short term 신경망

이동 로봇이 이동하는 표면에는 순간적인 표면의 상태 변화들이 존재할 수 있다. 물론, 로봇 제어 시스템 자체에서의 외란도 존재할 수 있다. 이러한, 외란에 대하여 강인한 특성을 갖도록 하기 위하여 PID 보상기를 계산 토크법과 함께 사용한다. 이 PID 보상기의 성능은 최적으로 결정된 이득 값에 따라 결정되어지므로 실시간에 외란 등의 영향에 강인하게 하기 위하여서는 지속적으로 학습에 의하여 이득 값을 최적화 해주어야 한다.

그림 5는 이동로봇의 고속 고정도 제어를 위해 앞 절에서 언급한 LTNN과 PID보상기의 이득을 동적으로 보정해 주기 위한 Short term 신경망(STNN)이 결합된 제안된 로봇제어기의 구조를 보이고 있다.

LTNN의 구조와 필요성에 대해서는 앞 절에서 이미 언급하였으므로 여기서는 STNN의 역할과 학습방법에 대해서만 논의하도록 한다. STNN은 순간적인 변화 상황에 대하여 빠르게 적응해야 하므로 신경망의 학습률을 높게 설정하였으며, 실시간 적용을 위하여서는 빠르게 수렴하는 것이 요구된다.

그림 5에서 보인 STNN의 구조는 다층 신경망 구조로 되어 있으며, 그 내부 구조는 그림 6과 같다. 신경망의 입력으로는 계획계획에 따라 매 샘플링 주기마다 각 제어기로 전송되는 속도 요구 값 및 위치 요구 값에 대해 센서로부터 측정된 각 구동바퀴의 실제 속도 값과 위치 값에 대한 각각의 편차인 속도 오차 값, e^i , e^r 및 위치 오차 값 e^l , e^r 그리고 이동로봇의 제어입력, $u^{i(k-1)}$, $u^{r(k-1)}$ 로 하고, 이들 입력에 의해 신경망에 적용된 역전파 학습알고리즘에 의해 학습된 결과로서 각 구동 바퀴별 PID 보상기의 이득을 결정하도록 한다.

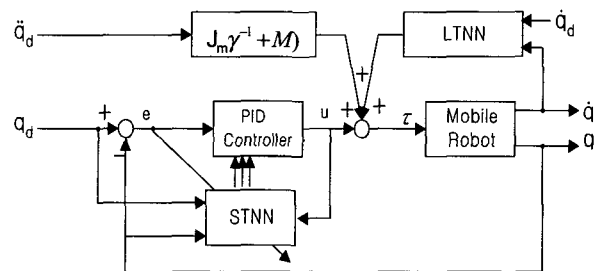


그림 5. STNN과 LTNN을 사용하는 제안된 이동로봇 제어기.

Fig. 5. Proposed mobile robot controller with STNN and LTNN.

급강하법에 입각한 오차 역 전파 알고리즘[4]-[7]을 사용하여 PID 보상기의 이득 값을 순간적인 외란 등으로부터 최적의 상태로 결정해 주기 위해서 평가 함수 C_{STNN} 을 다음과 같이 정의한다.

$$C_{STNN}(k) = \frac{1}{2} [q_d(k) - q(k)]^2 \quad (17)$$

마찰 등의 비 선형 함수의 존재도 궁극적으로는 위치 값의 변화를 초래하게 되므로 현재 원하는 위치 값에 대한 실제 로봇의 위치 값의 편차를 줄이는 쪽으로 즉, 정의된 평가 함수의 값을 줄이는 방향으로 가중치를 역 전파 학습에 의하여 학습시킴으로써 K_p , K_i 및 K_d 의 값을 최적으로 결정해 주고자 하는 것이다. (17)에서 정의된 오차함수를 이용하여 STNN의 학습기법은 다음과 같다.

먼저, 오차함수를 이용한 은닉 층에서 출력 층으로의 가중치 조절을 급강하법을 사용하여 구하기 위해 우선 평가함수의 가중치에 대한 기울기를 구한다.

$$\frac{\partial C_{STNN}(k)}{\partial W_{kj}(k-1)} = -e(k) \frac{\partial (q_d(k) - q(k))}{\partial W_{kj}(k-1)} \quad (18)$$

윗식의 우변 두 번째 항은 다음과 같이 chain rule에 의해 전개된다.

$$\frac{\partial q(k)}{\partial W_{kj}(k-1)} = \frac{\partial (q_d(k) - q(k))}{\partial u(k-1)} \frac{\partial K_{gain}(k-1)}{\partial W_{kj}(k-1)} \frac{\partial u(k-1)}{\partial K_{gain}(k-1)} \quad (19)$$

여기서, $q(k)$ 는 이동로봇의 출력 즉 위치 값을 나타내며, 제어입력 $u(k)$ 에 대해서 미분 가능하다고 가정한다.

(19)의 우변 두 번째 항은 chain rule을 사용하여 다음과 같이 전개된다.

$$\frac{\partial K_{gain}(k-1)}{\partial W_{kj}(k-1)} = \frac{\partial K_{gain}(k-1)}{\partial (net_k(k-1))} \frac{\partial (net_k(k-1))}{\partial W_{kj}(k-1)} \quad (20)$$

여기서, $\frac{\partial (net_k(k-1))}{\partial W_{kj}(k-1)}$ 는 은닉 층에서 k번째 출력 층 뉴런으로의 입력 값 $i_k(k=1, 2, \dots, n)$ 이며,

$$\begin{aligned} \frac{\partial K_{gain}(k-1)}{\partial (net_k(k-1))} &\triangleq f_{(k-1)}(net_k(k-1)) \\ &= \frac{1}{2}(1 - K_{gain}^2(k-1)) \end{aligned} \quad (21)$$

이다.

또한, (19)의 우변 세 번째 항은 다음 식과 같이 주어진다.

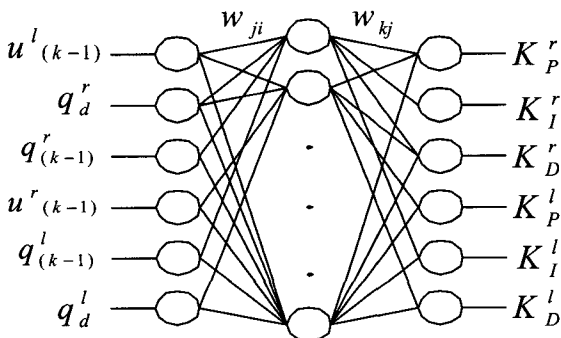


그림 6. STNN의 구조.
Fig. 6. Structure of STNN.

$$\begin{aligned} \frac{\partial u(k-1)}{\partial K_P(k-1)} &= e(k-1) - e(k-2) \\ \frac{\partial u(k-1)}{\partial K_I(k-1)} &= \frac{1}{2} [e(k-1) + e(k-2)] \\ \frac{\partial u(k-1)}{\partial K_D(k-1)} &= e(k-1) - 2e(k-2) + e(k-3) \end{aligned} \quad (22)$$

매 샘플링 순간의 위치 오차값이 주어지므로 (21), (22)에 의해 $\frac{\partial K_{gain}(k-1)}{\partial W_{kj}(k-1)}$ 의 값이 구해지며, $u(k)$, $q_d(k)$, $q(k)$ 및 $K_{gain}(k)$ 값들도 매 샘플링마다 구해지므로 (19)의 $\frac{\partial q(k)}{\partial W_{kj}(k-1)}$ 의 값은 구해진다. 따라서, (18)에 의하여 은닉 층과 출력 층 사이의 가중치를 위한 $\frac{\partial C_{STNN}(k)}{\partial W_{kj}(k-1)}$ 의 값이 구해진다. 이를 이용하여 은닉 층과 출력 층간의 가중치 조절은 다음과 같이 진행된다.

$$\begin{aligned} \Delta W_{kj}(k) &= -\eta \frac{\partial C_{STNN}}{\partial W_{kj}(k)} \\ W_{kj}(k) &= \Delta W_{kj}(k) + W_{kj}(k-1) \end{aligned} \quad (23)$$

입력 층과 은닉 층 사이의 가중치 조절은 입력 층 뉴런으로 들어오는 입력 값과 은닉 층 뉴런에서 출력되는 출력 값의 관계에 의해 위에서와 같은 방법으로 진행되어 다음 수식에 결정된다.

$$\begin{aligned} \Delta W_{ji}(k) &= -\eta \frac{\partial C_{STNN}}{\partial W_{ji}(k)} \\ W_{ji}(k) &= \Delta W_{ji}(k) + W_{ji}(k-1) \end{aligned} \quad (24)$$

V. 시뮬레이션 및 검토

본 절에서는 컴퓨터 시뮬레이션을 통하여 제안된 이동로봇 제어기의 성능과 타당성을 검증하기 위해서 제안된 제어기법에 의해서 로봇이 계획된 경로를 추종하였을 때 궤적 추종결과와 발생한 위치 및 속도 오차 값을 계산토크 법에 의한 결과와 다음의 두 가지 경우에 대한 결과를 비교 분석하도록 한다.

1) STNN만을 사용하여 구동바퀴와 표면간의 비선형 마찰항의 보상 없이 PID보상기의 이득 값을 실시간으로 보상한 경우, 이때 PID 보상기의 이득 초기 값은 계산토크 법에서 사용한 값으로 한다.

2) LTNN만을 사용하여 구동바퀴와 표면간의 마찰항을 보상에 준 경우, 이때 PID 보상기의 이득 값은 계산토크 법에서 사용한 값으로 한다.

시뮬레이션에 사용된 이동로봇의 파라미터는 제작된 실제 로봇에서 측정된 값으로서 표 1과 같다.

또한 로봇의 궤적은 그림 7과 같이 로봇의 두 바퀴간의 중간지점(중심점이라 한다)이 X-Y 평면 (0,0)지점에서 출발하여 Y축으로 2m, X축으로 3m인 직사각형 궤적을 이동한 후 초기 지점으로 되돌아오도록 계획하였다. 또한, 궤적계획에 있어서, 로봇의 각 구동바퀴가 정 방향으로만 회전한다고 가정한다. 따라서 직사각형의 각 모서리 지점에서 로봇이 다음 변으로 이동하기 위해서 회전운동을 할 때 예상되는 로봇의 기구학적 구조에 의한 위치 오차의 발생을 최소화하기 위해서 각 모서리 지점에서 일정한 변위의 기울기를 갖도록 하였다. 각 모서리에서 로봇의 회전하는 방식은 편의를 위해 우측 바퀴

표 1. 시뮬레이션에 사용된 파라미터 값.

Table 1. Parameter values used in computer simulation.

표시	값	설명
m	2.1 [Kg]	모터의 질량
J_m	0.2×10^{-3} [Kgf · m/sec ²]	모터 이너시아 + 기어 이너시아
B_m	0.2 [Kgf · m/sec]	모터의 마찰계수
γ	1/0.0322	기어비
W_p	7.2 [Kg]	평판의 질량
W	20.7 [Kg]	이동로봇의 전체 질량

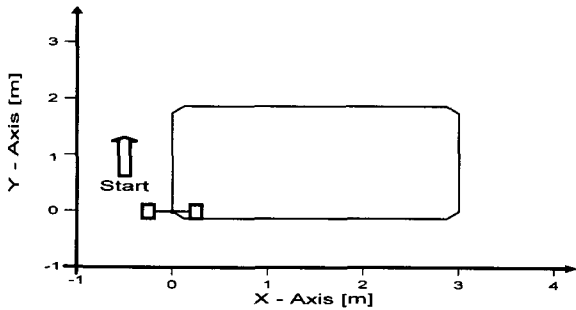


그림 7. 이동로봇의 궤적 계획.
Fig. 7. Trajectory planing of the mobile robot.

는 로봇의 중심점이 기구학적 계산에 의해 각 구간의 끝점에 도착 가능한 위치에서부터 좌측바퀴가 다음 구간의 시작점에 도착할 때까지는 정지상태로 두었으며, 좌측바퀴의 다음 구간 도착 후에는 두 바퀴가 동시에 가속상태로 되도록 하기 위해 직사각형의 시작점과 끝점에서 각각 0.8sec씩의 가속구간과 감속구간을 각각 설정하였다. 또, 정상 이동시의 속도 요구 값은 50mm/sec로 하고 샘플링 주기는 20msec로 하였다. 따라서 각 샘플링 주기마다 로봇은 1mm씩 이동하여 궤적추종을 완료하는데까지는 222.8sec가 걸린다. 그리고 로봇이 이동할 때 야기되는 구동바퀴와 표면간에 발생하는 마찰력은 로봇이 이동하는 표면의 조건에 따라 다르지만, 여기서는 로봇이 이동하는 표면의 변화가 비교적 완만한 기울기를 가진다고 가정하고, 쿨롱마찰과 점성마찰의 각 계수 값은 다음과 같이 사용하였다.

$$\begin{aligned} F_C &= 0.1 + \sin(0.001 \times t) \quad [\text{N m}] \\ F_B &= \sin(0.0125 \times t) \quad [\text{N/m/s}] \end{aligned} \quad (25)$$

그림 8-12는 계산토크 법에 의해 계획된 궤적을 이동하였을 때의 로봇 중심점의 궤적 추종 결과와 이때 발생한 위치 오차 및 속도 오차 그리고 각 구동바퀴의 속도 응답특성을 나타내고 있다. 계산토크 법에서도 비교적 우수한 결과를 보이고 있지만, 다음 구간으로 이동하는데 따른 가속 및 감속 구간에서 위치 값의 경우 최대 2.3mm의 오차를 그리고 속도 값은 최대 4.6mm/sec의 오차를 발생하였다.

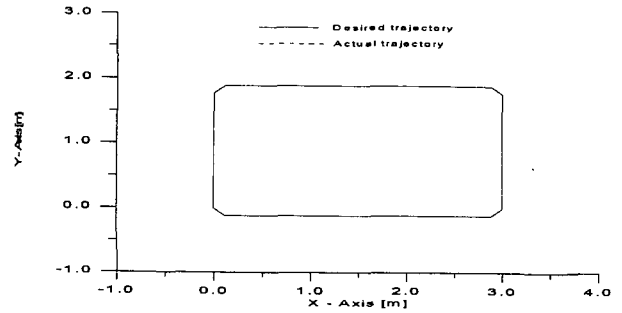


그림 8. 계산토크 법에서 이동로봇 위치응답.
Fig. 8. Position response of the mobile robot in computed torque method.

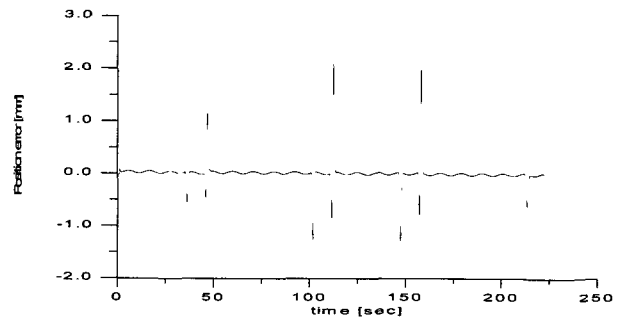


그림 9. 계산토크 법에서 위치 오차.
Fig. 9. Position error of the mobile robot in computed torque method.

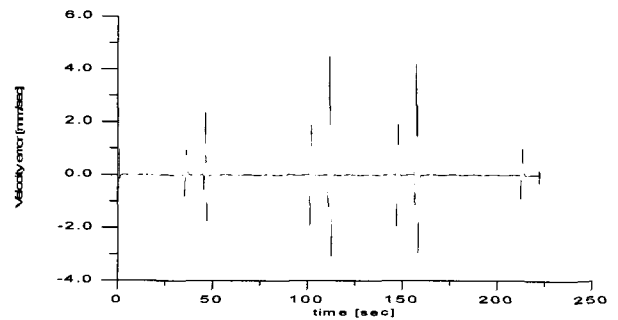


그림 10. 계산토크 법에서 속도오차.
Fig. 10. Velocity error of the mobile robot in computed torque method.

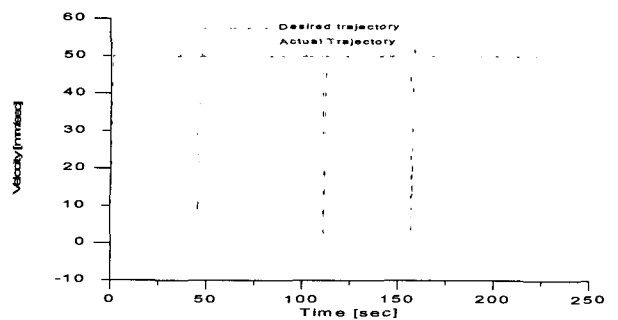


그림 11. 좌측바퀴의 속도 응답.
Fig. 11. Velocity response of left wheel in computed torque method.

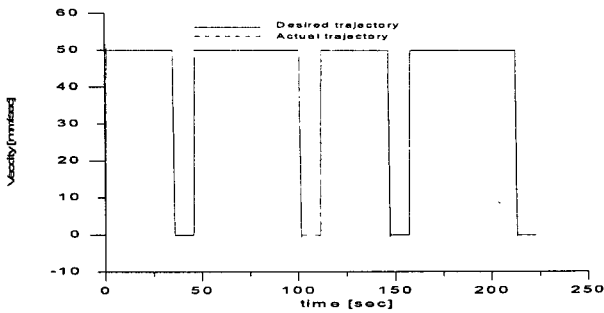


그림 12. 우측 바퀴의 속도응답.
Fig. 12. Velocity response of right wheel in computed torque method.

또한 등속구간에서도 유계된 외란에 의해서 약간의 위치편차가 내포되어 있음을 알 수 있다. 이때 사용된 PID보상기의 이득 값은 여러번의 시행착오에 의해서 속도 및 위치 추종이 최적의 상태가 되도록 하였을 때의 값을 사용하였으며, 좌측바퀴의 PID 이득값은 $K_p=15$, $K_i=30$, $K_d=1.15$ 로 하였고, 우측바퀴의 PID 이득값은 $K_p=10$, $K_i=20$, $K_d=0.8$ 이다. 따라서 계산토크법에서 발생하는 이러한 위치 및 속도편차의 원인중의 하나인 외부에서 유계된 외란의 영향을 최소화하기 위하여 STNN을 사용하여 PID보상기의 이득을 실시간으로 보상해 보았다. 이때 PID보상기의 각 초기 값은 계산토크 법에서 사용된 값으로 하였으며, STNN의 은닉 층의 뉴런갯수는 PID 이득 값을 실시간으로 보상하기 위해서는 계산속도가 빠르고 또한 이득 값 선정시의 정밀성을 함께 고려하여 13개로 하였을 때가 가장 최적의 상태가 되었다. STNN의 은닉 층 활성화함수는 시그모이드 함수를 그리고 출력 층의 활성화함수는 선형 활성화함수를 사용하였으며 학습률(η)을 0.75로 하였다. 대부분의 경우 신경망을 제어기나 또는 다른 목적으로 사용할 경우에 신경망의 각각의 가중치의 초기 값을 주로 0에서 1사이의 값인 임의의 값으로 설정하고 원하는 출력을 얻기 위해 학습을 시키는 경우가 일반적이다. 그러나 본 연구에서 사용한 PID 외부 루프 보상기의 이득을 실시간으로 보상해 주기 위해서는 앞에서 언급한 일반적인 제어시스템에서 적용하는 신경망 가중치의 초기치 적용방식을 채택할 경우 신경망의 학습에 소비되는 시간의 과다 소요로 인하여 PID이득의 실시간 보상이 어려울 뿐만 아니라 초기 학습결과에 대해서 보상된 PID 이득계수의 최적 여부를 결정하는 어려움이 있다. 따라서 이러한 문제를 해결하는 방법으로 대부분의 경우 유전진화 알고리즘 등을 사용하여 신경망 가중치의 초기 값을 결정해 주는 기법도 연구되어지고 있으나 본 연구에서는 제어 시스템의 구성상의 복잡성을 배제하기 위하여 STNN의 각 가중치의 초기 값을 임의의 값으로 선정하지 않고, STNN의 초기 출력이 계산토크 법에서 사용한 PID 이득 값의 근사치를 출력 가능하도록 가중치의 초기 값을 계산에 의해 선정하였다. 이것은 적용한 STNN의 구조를 비교적 간단히 구성함으로써 가능하며, 또한 신경망의 학습시간을

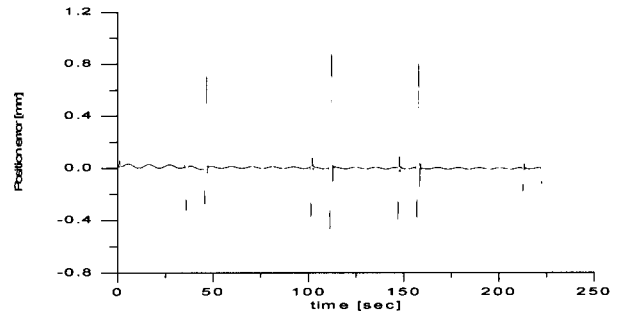


그림 13. STNN에 의한 이동로봇의 위치오차.
Fig. 13. Position error of the mobile robot in computed torque method with STNN.

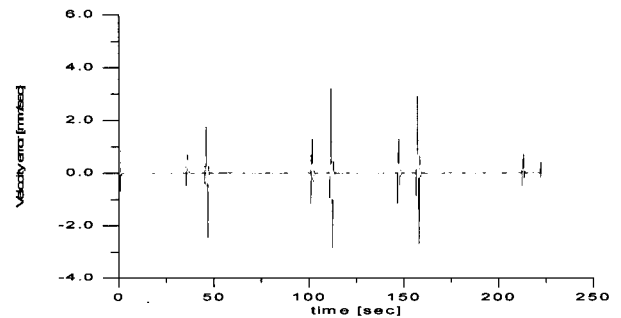


그림 14. STNN에 의한 이동로봇의 속도오차.
Fig. 14. Velocity error of the mobile robot in computed torque method with STNN.

단축함으로써 PID 보상기의 이득을 실시간 보상해주는 효과를 도모하였다. 따라서 학습단계를 거치지 않고 궤적 추종시 학습 및 실행단계를 동시에 수행 가능하도록 하였다.

그림 13과 14에서 위에서 언급한 계산토크 법에서 STNN에 의해 PID 보상기의 이득을 실시간 보상했을 때의 위치 및 속도추종에 대한 편차를 나타내고 있다. 그림에서는 외부에서 유입된 외란의 영향은 PID 이득 값의 실시간으로 보상해 줌으로써 로봇의 궤적추종에 따른 위치 편차를 다소 줄일 수 있음을 보이고 있다. STNN의 평가함수를 위치 값의 오차를 최소화하도록 하였으므로 위치 추종오차의 경우 등속 구간에서도 처음 시작지점에서의 오차에 비해 시간이 경과함에 따라 그 값이 점차적으로 줄어들고 있음을 알 수 있다. 속도오차의 경우도 외란의 영향을 보상해 줌으로써 계산토크 법에 비해 가속 및 감속구간에서도 다소 줄어들었음을 알 수 있다. 그러나 STNN에 의해 외부에서 유입되는 외란의 영향을 보상하여 이동로봇이 궤적을 추종하였을 때 그 결과 위치 및 속도 편차는 기대 이상의 값에는 미치지 못하고 있다.

따라서 이때 발생하는 위치 및 속도 편차의 원인을 구동 바퀴와 표면간에서 발생하는 마찰의 영향으로 간주하고 이들의 영향을 LTNN을 사용하여 보상하였을 때의 결과를 고찰하기로 한다.

3절에서 모델링한 마찰력에 대한 마찰계수의 값들을

LTNN을 이용하여 이들 값을 추정하여 피드포워드로 보상해 줌으로써 마찰에 대한 속도 및 위치편차의 영향을 최소화하도록 한다. 이때 PID 이득 값은 계산토크 법에서 사용한 값으로 고정시켜두었으며, LTNN의 은닉 층 뉴런 개수는 23개로 하였다. 은닉 층 및 출력 층의 활성화 함수는 STNN의 경우와 같이 시그모이드 함수 및 선형 활성화 함수를 각각 사용하였으며, 학습방법은 B.P 학습 알고리즘을 사용하고, 학습률은 마찰계수 값을 정밀하게 추정 가능하도록 하기 위하여 0.2로 하였다. 그림 15와 16은 LTNN에 의해 이동로봇의 각 구동바퀴와 표면간의 마찰을 보상했을 때의 위치 및 속도 추종 편차를 나타내고 있다.

LTNN에 의한 마찰에 대한 보상은 위치 및 속도 값 추종에 따른 오차가 앞의 두 경우에 비해 현저히 줄어들었음을 알 수 있지만, 이들 오차 값에는 고정된 PID 보상기의 이득 값으로 인해 외란에 의한 영향이 다소 포함되었음을 알 수 있다.

따라서, 앞에서 사용한 LTNN과 STNN을 동시에 사용하여 마찰 및 외란 등에 따른 영향을 동시에 보상해주는 제안된 제어기의 궤적추종 결과와 그에 따른 위치 및 속도오차 그리고 각 구동바퀴에서의 속도 추종결과가 그림 17-21에 보이고 있다. 이때 각 신경망의 사용 조건은 앞에서 각각 개별적으로 사용하였을 때와 동일하게 적용하였다.

그림에서 로봇 중심점의 이동에 따른 계획된 궤적과 실제 궤적추종의 결과는 계획된 경로의 각 구간 값[단위

[m]]에 대해 실제 이동에 대한 위치 값의 편차[단위 [mm]]가 너무 작은 값이므로 그림에서는 계산토크 법에서의 제안된 알고리즘에서 궤적추종 결과가 유사하게 보인다. 그러나, 위치오차의 경우에서 보면 LTNN 또는 STNN만을 사용하였을 때 보다 그 편차가 궤적추종의 시간이 경과함에 따라 현저히 줄어들고 있음을 알 수 있다. 그리고 속도 오차 값도 또한 LTNN만에 의해서 마찰을 보상했을 때 보다 등속 구간 및 가,감속 구간에서도 다소 줄어들었음을 알 수 있다. 그리고 그림 20과 21은 제안된 제어기법에서의 각 구동바퀴의 속도 추종 값을 나타내고 있으며, 계산토크법과 비교했을 때 속도 추종의 정밀도가 매우 좋아 졌음을 알 수 있다. 따라서, LTNN과 STNN을 동시에 적용하여 구동바퀴와 표면간

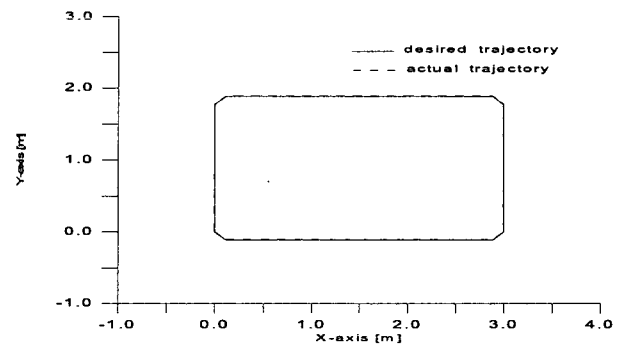


그림 17. 제안된 기법에서의 로봇 위치응답.
Fig. 17. Position responses of the mobile robot in proposed method.

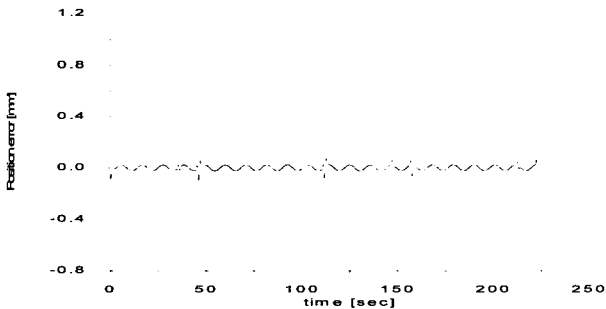


그림 15. LTNN에 의한 이동로봇의 위치오차.
Fig. 15. Position error of the mobile robot in computed torque method with LTNN.

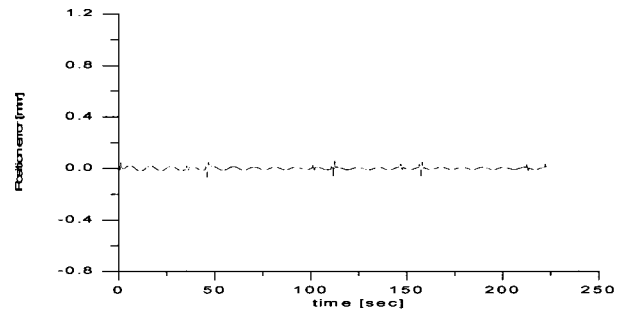


그림 18. 제안된 기법에서의 위치 오차.
Fig. 18. Position error of the mobile robot in proposed method.

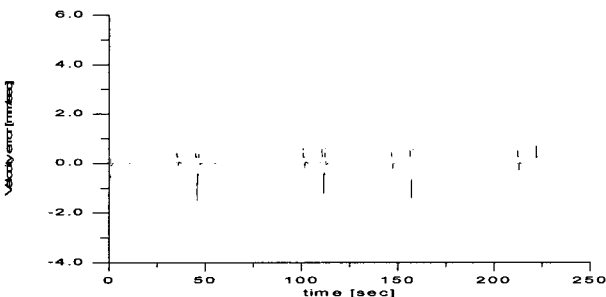


그림 16. LTNN에 의한 이동로봇의 속도오차.
Fig. 16. Velocity error of the mobile robot in computed torque method with LTNN.

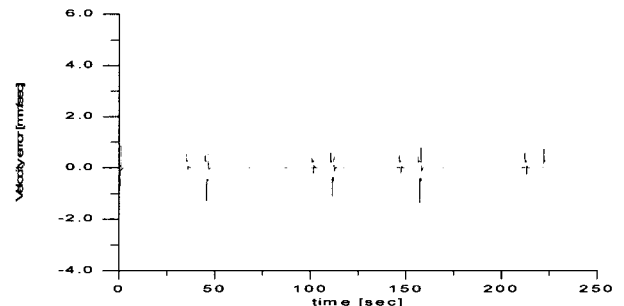


그림 19. 제안된 기법에서 로봇 속도오차.
Fig. 19. Velocity error of the mobile robot in proposed method.

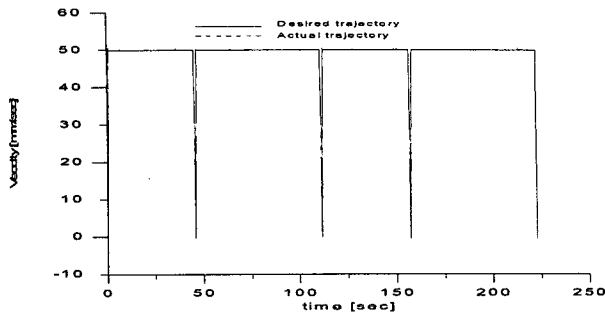


그림 20. 좌측 바퀴의 속도응답.
Fig. 20. Velocity response of left wheel in proposed method.

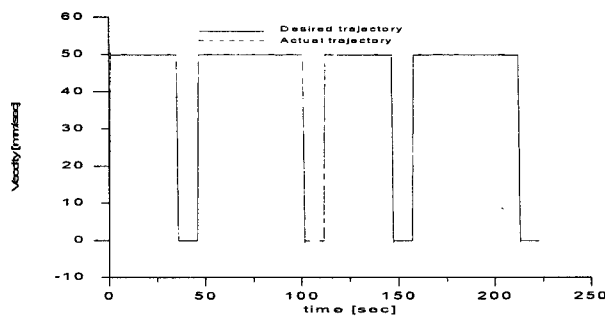


그림 21. 우측 바퀴의 속도응답.
Fig. 21. Velocity response of right wheel in proposed method.

의 마찰과 외부에서 유입되는 외란 등의 영향을 줄이기 위해 PID 이득 값의 실시간으로 보상을 동시에 수행함으로써 이동로봇의 정밀제어가 가능함을 알 수 있다.

VI. 결론

본 논문에서는 종래의 로봇제어기들이 갖는 비 선형 현상에 의한 정밀성의 한계를 극복하기 위하여 로봇의 동적 모델에 포함되지 않은 로봇의 표면이동에 따른 구동바퀴와 표면간의 마찰항을 모델링하고, 그 마찰모델의 계수 값을 LTNN으로 칭해진 다층 구조 신경망의 학습에 의해서 추정하여 피드포워드로 보상하고, 또한, 로봇의 이동표면에 상존하는 외란 등의 영향을 보상하기 위한 PID 보상기의 성능향상을 위해 그 이득상수를 STNN 신경망에 의하여 실시간으로 최적의 값으로 결정해 줌으로써 로봇의 고속 고 정밀제어를 가능하게 하는 제어 알고리즘을 제안하였다. 전체적인 제어 알고리즘의 성능을 실제 모델을 사용하여 컴퓨터 시뮬레이션으로 보이고, 제안된 알고리즘의 타당성을 검증하기 위해 계산토크 법에 의한 동적제어의 결과 및 각 신경망을 개별적으로 적용하였을 때와 비교 검토하였다. 시뮬레이션 결과에서 알 수 있듯이 STNN과 LTNN을 각각 개별적으로 사용하였을 경우에 두 경우 모두 계산토크 법에 의한 결과보다는 비교적 우수한 성능을 보이고 있다. 그러나 STNN만 사용한 경우에는 PID 보상기에 의해 외란 등에 대한 보상을 실시간으로 해 주더라도 구동 바퀴와 표면간의 마찰항 등에 의해 위치 및 속도 편차를 일정 값

이상을 내포하고 있음을 알 수 있고, 또한 LTNN만을 사용하여 마찰항에 대한 영향을 보상해 준 경우에는 속도 편차가 현저히 감소함에도 불구하고 외란 등에 의한 영향으로 인해 위치 오차 값이 일정한 값을 유지하고 있다. 따라서, LTNN과 STNN을 함께 사용하여 마찰항에 대한 영향과 PID 이득 값을 동시에 보상해 줌으로써 위치 오차의 경우 궤적추종의 시간이 경과함에 따라 점차적으로 감소하였고, 속도 오차의 경우도 PID 보상기의 이득 값을 실시간으로 보상하여 외란 등의 영향을 보상해 줌으로써, LTNN만을 사용하여 마찰항만을 보상해 주었을 때보다 감소함을 알 수 있었다. 따라서 제안된 알고리즘을 사용하여 이동로봇의 궤적추종에 대한 로봇의 고속 고 정밀 제어가 가능함을 알 수 있다.

참고문헌

- [1] A. Moran, N. Tokunaga and M. Hayase, "Autonomous motion control of mobile robots using fuzzy-neural networks," *Third International Conference on Motion and Vibration Control*, pp. 104-109, 1996. 9.
- [2] T. Hamel and D. Meizel, "Robust control laws for wheeled mobile robots," *Int. Journal of Systems Science*, vol. 27, no. 8, pp. 695-704, 1996.
- [3] H. Miyamoto, M. Kawato, T. Setoyama and R. Susuki, "Feedback-error-learning neural network for trajectory control of a robotic manipulator," *Neural Networks*, vol. 1, pp. 251-265, 1988.
- [4] C. L. Chen and F. Y. Chang, "Design and analysis of neural/fuzzy variable structural PID control systems," *IEE Proc. Control Theory and Applications*, vol. 143, no. 2, pp. 200-208, 1996. 3.
- [5] R. Vitthal, P. Sunthar and C. D. Rao, "The generalized proportional-integral-derivative(PID) gradient descent back propagation algorithm," *Neural Networks*, vol. 8, no. 4, pp. 563-569, 1995.
- [6] J. Tanomaru and S. Omatu, "Process control by on-line trained neural control," *IEEE Trans. Industrial Electronics*, vol. 39, no. 6, pp. 511-521, 1992.
- [7] 성 홍석, 이 쾌희, "다층 신경회로망을 이용한 비선형 시스템의 건설한 제어" 전자공학회 논문지-S, vol. 34, no. 9, pp. 41-49, 1997. 9.
- [8] 문 영주, 오 세영, "로봇 매니플레이터의 동력학적 신경제어 구조" 전자공학회 논문지-B, vol. 29, no. 8, pp. 15-23, 1992. 8.
- [9] C. Wu and C. H. Huang, "Back-propagation neural networks for identification and control of a direct drive robot," *Journal of Intelligent and Robotic Systems*, vol. 16, pp. 45-64, 1996.
- [10] S. Jangannathan and F. L. Lewis, "Identification of nonlinear dynamical systems using multilayered

neural networks," *Automatica*, vol. 32, no. 12, pp. 1707-1712, 1996.

- [11] X. Cui and K. G. Shin, "Direct control and coordination using neural networks," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 686-697, 1993.
- [12] Y. H. Kim and F. L. Lewis, "Output feedback control of rigid robots using dynamic neural networks," *International Conference on Robotics*

and Automation, pp. 1923-1928, 1996. 4.

- [13] L. Jin, P. N. Nikiforuk and M. M. Gupta, "Fast neural learning and control of discrete-time nonlinear systems," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 25, no. 3, pp. 478-488, 1995. 3.
- [14] 이 장명, 김 용태, "이동 로봇의 동적 제어 구현" *전자공학회 논문지-S*, vol. 34-S, no. 1, pp. 54-64, 1997. 7.



주 진 화

1990년 부경대학교 전자공학과 졸업. 동아대학교 대 학원 석사(1993), 현재 부산대학교 전자공학과 박사과정 수료. 주 관심분야는 로봇 시스템 설계 및 제어, 마이크로 프로세서 응용, 신경 회로망 응용 등.



이 장 명

1980년 서울대학교 전자공학과 졸업. 동대학원 석사(1982), USC(남가주대학교) 공학박사(1990). 1992년-현재 부산대학교 전자공학과 부교수, 부산대학교 컴퓨터 및 정보통신 연구소 전자기술부장. 주 관심분야는 다관절 로봇 시스템의 설계 및 제어, 마이크로 프로세서 응용 시스템 설계, 모터 구동방식 및 제어 등.