신경회로망을 이용한 비전 기반 이동 로봇의 위치제어에 대한 실험적 연구

Experimental Studies of Vision Based Position Tracking Control of Mobile Robot Using Neural Network

정 슬*, 장 평 수, 원 문 철, 홍 섭
(Seul Jung, Pyoung Soo Jang, Moon Cheol Won, and Sup Hong)

Abstract : Tutorial contents of kinematics and dynamics of a wheeled drive mobile robot are presented. Based on the dynamic model, simulation studies of position tracking of a mobile robot are performed. The control structure of several position control algorithms using visual feedback are proposed and their performances are compared. In order to compensate for uncertainties from unknown dynamics and ignored dynamic effects such as slip conditions, neural network based position control schemes are proposed. Experiments are conducted and the results show the performance of the vision based neural network control scheme turned out to be the best among several proposed schemes.

Keywords : mobile robot, neural network, visual feedback, reference compensation technique.

I. 서론

이동로봇에 관한 연구는 장애물 회피나 localization, 빠른 움직임의 속도제어 등이 대부분이다. 다양한 센서의 융합과 함께 필터 설계를 통한 위치 추정과 장애물 회피 등은 이동로봇의 자율주행을 완성하는데 필요한 기술들로 많은 연구가 진척되어 있으며 활발하다[1-3]. 이는 대부분의 이동 로봇의 작업 환경을 살펴보면 정확한 경로 Tracking이 필요 없이 속도 제어를 통해 목표점에 도달하는 것이 가능하기 때문이다.

하지만 이동로봇을 사용하여 글자를 쓴다거나 그림을 그리는 경우에는 정확한 위치제어가 필요하며, 특히, 이동로봇을 사용하여 선박과 같은 큰 물체를 용접하거나 grinding을 하는 경우, 도로의 갈라진 틈새를 추종하는 경우 등과 같은 작업에 필요한 힘제어를 하는 경우에는 정확한 위치제어가 필요하게 된다[4-6].

이동로봇은 *non-holonomic* 시스템으로 상태변수와 제어 입력변수의 수가 다르므로 동력학을 기반으로 하는 위치 제어시 구속조건을 만족해야 하고, 제어 변수를 선택적으로 취해야 하므로 제어가 어렵다[7]. 또한 바퀴의 슬립이나 바퀴의 불균형 등의 동적인 요소를 고려하지 않는데서 오는 불확실성에 의해 위치오차가 발생한다. 이러한 오차를 보상하는 강인한 제어 알고리즘으로 백스테핑 제어 방식이 제안되었다[8-11]. 또한 불확실성 요소들을 보상하고자 신경회로망을 사용한 지능 제어 방식도 시뮬레이션을 통해 제안되었다[14-15]. 하지만 이러한 제어방식은 센서의 정확성을 가정한 상태에서 시뮬레이션한 결과이므로 실제의 경우에 있어서는 localization을 해야 한다. 그러므로, 제어 알고리즘과 상관없이 위치제어를 정확하게 하기 위해서는 dead-reckoning을 보정하는데 필요한 절대센서가 필요하다. 절대센서가 없이는 위치제어를 정확하게 할 수 없게 된다. 절대센서로는 공간에서 사용할 수 있는 DGPS, 바다 속에서 사용하는 LBL(Long Base Line) 등이 있다.

따라서, 본 논문에서는 DGPS에서 LBL을 사용하는 위치 추종의 경우를 축소한 한 예로서 간단한 실험을 통해 visual feedback을 사용하는 신경회로망 제어의 성능을 실험적으로 검증하고자 한다. 비전 센서의 정확도에 따라 이동로봇의 위치 추종성능이 좌우된다. 실내 환경에서 이동로봇의 위치 제어를 정확하게 하기 위해 비전 센서를 사용하여 visual feedback을 이용하면 visual servoing이 가능하다. 비전으로 로봇의 위치정보를 추출하여 기준 경로와 비교하여 오차를 만든 다음, 제어기를 거쳐 생성된 제어 입력을 무선으로 로봇에 전달하여 위치를 매 샘플마다 보정할 수 있다. 비전 정보를 사용하는 제어방식으로는 기존의 내부 제어루프가 있고, 바깥에 외부 제어루프를 첨가함으로써 추종오차를 줄이는 방식을 제안한다. 제안하는 방식들로는 visual feedback PD 제어 방식, visual feedback PD 제어 + 신경회로망 제어 방식, 그리고 내부 엔코더 기반 PD 제어 + 외부 신경회로망 제어 방식이 있다. visual feedback PD 제어 방식에서 발생하는 오차는 신경회로망을 사용하여 실시간으로 줄이고자 하였다. 신경회로망을 사용하는 입력보상방식은 이동로봇 자체의 내부제어기를 수정할 필요 없이 그대로 둔 상태로 외부에서 격리된 상태에서 무선으로 제어가 가능한 장점이 있다. 이는 외부제어기와 내부제어기가 공간적으로 완전히 분리된 상태에서 제어할 수 있는 이점이 있으므로 특이한 상황에서의 제어 응용에 사용할 수 있는 장점이 있다.

본 논문에서는 wheeled drive 형태의 이동로봇의 기구학, 동력학식, 그리고 오차 좌표계를 구하는 과정을 2장, 3장 그리고 4장에 각각 정리하고, 5장에서는 다양한 제어방식을 제안하였다. 6장에서는 동력학식을 기반으로 시뮬레이션 하

---

* 책임저자(Corresponding Author)
논문접수 : 2002. 11. 12., 채택확정 : 2003. 4. 19.
정슬, 장평수, 원문철 : 충남대학교 메카트로닉스공학과
(jungs@cnu.ac.kr/j-water@hanmail.net/mcwon@cnu.ac.kr)
홍 섭 : 한국 해양 연구소(suphong@kriso.re.kr)
※ 본 연구는 해양수산특정 연구 사업으로 한국 해양 연구소 위탁과제로 연구되었으며 이에 감사드립니다.

였으며, 7장에서 실제 실험을 통해 각 제어 방식의 위치 추종 결과를 비교 분석하였다. 결과적으로 보면 신경회로망을 사용한 제어방식의 추종 결과가 가장 우수한 것으로 나타났다.

II. 이동로봇의 기구학

이 장에서는 이동로봇의 기구학을 튜토리알 형식으로 자세하게 유도한다. 아래 그림 1은 일반적인 WDR(Wheeled Drive Robot)의 기구학을 나타낸다. 이동 로봇의 움직임은 x-y 평면에서 움직이므로 그림 1과 같이 나타낼 수 있다. X,Y는 전체 좌표를 나타내며 x,y는 로봇 좌표를 나타낸다.

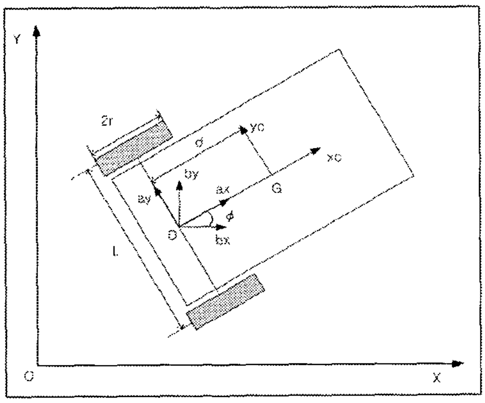


그림 1. 이동로봇의 좌표계.
Fig. 1. Coordinates of Mobile Robot.

그림 1에서 보면 r은 바퀴의 반지름, L은 바퀴사이의 거리, G는 이동로봇의 무게중심, D는 바퀴축의 중심, d는 무게중심에서 바퀴축 간의 최단거리 그리고 a_x, a_y와 b_x, b_y는 유닛 벡터이다.

로봇의 무게 중심 G는 바퀴축의 중심 D로부터 수직으로 d거리만큼 떨어진 곳에 있다. 바퀴축의 중심 D에서 로봇의 무게 중심 G점에서의 로봇의 선속도를 구하고자 하면 다음과 같다.

$$\begin{aligned} v &= v_D + w_C \times (d a_x) \\ &= \dot{x} b_x + \dot{y} b_y + \dot{\phi} a_z \times d a_x \\ &= \dot{x} b_x + \dot{y} b_y + d\dot{\phi} a_y \end{aligned} \tag{1}$$

여기서 v_D는 카트의 D 점에서의 선속도이고 w_c는 각속도이다. a_y를 유닉벡터 b로 나타내면 그림 1로부터 다음과 같다.

$$a_y = b_y \cos\phi - b_x \sin\phi \tag{2}$$

(2)를 (1)에 대입하여 정리하면

$$\begin{aligned} v &= \dot{x} b_x + \dot{y} b_y + d\dot{\phi}(b_y \cos\phi - b_x \sin\phi) \\ &= (\dot{x} - d\dot{\phi}\sin\phi) b_x + (\dot{y} + d\dot{\phi}\cos\phi) b_y \end{aligned} \tag{3}$$

(3)에서 무게중심에서의 선속도를 v라하고 각 속도 $\dot{\phi}$를 w라 하면 이는 각 축 방향의 속도로 분할되어 다음과 같이 얻어 질 수 있다.

$$\begin{aligned} \dot{x} &= v\cos\phi - dw\cos(90-\phi) = v\cos\phi - dw\sin\phi \\ \dot{y} &= v\sin\phi + dw\cos\phi, \\ w &= \dot{\phi} \end{aligned} \tag{4}$$

위 식을 행렬의 형태로 바꾸어 보면 다음과 같다. 이 행렬식이 로봇의 기구학이며 직교 공간에서 제어 입력과 속도와의 관계를 나타내준다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -d\sin\phi \\ \sin\phi & d\cos\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{5}$$

무게 중심이 바퀴의 중심축의 연장선상에 있을 때, 즉 $d=0$일 때 (5)는 다음과 같이 간단하게 된다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{6}$$

그렇다면 조인트 공간에서 로봇의 바퀴 속도와 카테시안 공간에서 속도와의 관계를 알아보자. 오른쪽 바퀴의 선속도를 v_R이라 하고 왼쪽 바퀴의 선속도를 v_L이라 하면 무게중심에서 이동로봇의 선속도 v는 다음과 같다.

$$v = \frac{1}{2}(v_R + v_L) \tag{7}$$

바퀴의 반지름을 r이라 하고 오른쪽 바퀴의 움직인 각을 θ_R, 왼쪽 바퀴의 움직인 각을 θ_L이라 할 때 오른쪽 바퀴의 각속도는 w_R, 왼쪽 바퀴의 각 속도는 w_L이 되므로 각과 각속도는 다음의 관계가 있다.

$$w_R = \dot{\theta}_R,\ w_L = \dot{\theta}_L \tag{8}$$

또한 각 바퀴의 선속도는 바퀴의 각속도와 외적관계인데 이루는 사이각이 90도이므로 각각 다음과 같은 관계가 있다.

$$\begin{aligned} v_R &= w_R \times r = w_R r \sin\theta = w_R r, \\ v_L &= w_L \times r = w_L r \sin\theta = w_L r \end{aligned} \tag{9}$$

(9)를 (7)에 대입하여 정리하면 이동로봇의 선속도는 다음과 같이 각 바퀴의 평균 각 속도로 표현될 수 있다.

$$v = \frac{1}{2}(r w_R + r w_L) \tag{10}$$

두 바퀴사이의 거리를 L라 하고 오른쪽 바퀴에 의한 로봇의 각속도를 $\dot{\phi}_R$ 그리고 왼쪽 바퀴에 의한 로봇의 각속도를 $\dot{\phi}_L$이라 하면 로봇의 각 속도 w는 다음과 같이 각 바퀴의 선속도에 의해 표현될 수 있다.

$$w = \dot{\phi}_R - \dot{\phi}_L = \frac{v_R}{L} - \frac{v_L}{L} = \frac{r w_R}{L} - \frac{r w_L}{L} \tag{11}$$

(10)과 (11)로 부터 바퀴의 각속도와 로봇의 속도와의 관계는 다음과 같다.

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{12}$$

그러므로 각 바퀴의 각속도에 의해서 결정되어지는 직교 좌표는 (5)와 (12)를 연립하므로 다음과 같이 구할 수 있다.

$$P = J\theta$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\phi - \frac{rd}{L}\sin\phi & \frac{r}{2}\cos\phi + \frac{rd}{L}\sin\phi \\ \frac{r}{2}\sin\phi + \frac{rd}{L}\cos\phi & \frac{r}{2}\sin\phi - \frac{rd}{L}\cos\phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{13}$$

여기서 J는 로봇의 자코비안이다.

무게 중심이 바퀴의 중심축의 연장선상에 있을 때, 즉 $d=0$일 때 위 식은 다음과 같이 간단하게 된다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{14}$$

III. 이동로봇의 동력학

이 장에서는 이동로봇의 동력학 및 제어에 대해 튜토리알 형식으로 자세하게 전개하고자 한다. 이동로봇의 움직임은 평면에서 움직이므로 위치에너지는 없고 운동에너지에 의해 좌우된다. 이 장에서는 라그란지안 방식을 통하여 동력학식을 유도하고자 한다.

먼저 바퀴를 제외한 카트의 운동에너지 K_C는 다음과 같다.

$$K_C = \frac{1}{2} I_C w_C^2 + \frac{1}{2} m_C v^2 \tag{15}$$

여기서 I_C는 카트의 관성모멘트, m_C는 카트의 질량, w_C는 카트의 각속도이다. 카트의 각속도는 그림 1로부터 다음과 같이 표현된다.

$$w_C = \dot{\phi} a_z \tag{16}$$

(3)과 (16)을 (15)에 대입하면 카트의 운동에너지 K_C는 다음과 같다.

$$\begin{aligned} K_C &= \frac{1}{2} I_c\dot{\phi}^2 + \frac{1}{2} m_c[(\dot{x} - d\dot{\phi}\sin\phi)^2 + (\dot{y} + d\dot{\phi}\cos\phi)^2] \\ &= \frac{1}{2} I_c\dot{\phi}^2 + \frac{1}{2} m_c[\dot{x}^2 + \dot{y}^2 + d^2\dot{\phi}^2 - 2d\dot{\phi}(\dot{x}\sin\phi - \dot{y}\cos\phi)] \end{aligned} \tag{17}$$

오른쪽 바퀴의 운동에너지 K_{RW}는 다음과 같다.

$$K_{RW} = \frac{1}{2}[I_{bx}^R w_{bx}^2 + I_{by}^R w_{by}^2 + I_{bz}^R w_{bz}^2] + \frac{1}{2} m_{RW} v_{RW}^2 \tag{18}$$

여기서 $I_{bx}^R, I_{by}^R, I_{bz}^R$는 각각 bx, by, bz방향의 관성 모멘트로 $I_{bx}^R = I_{bz}^R = \frac{1}{4} m_{RW} r^2, I_{by}^R = \frac{1}{2} m_{RW} r^2$이고 m_{RW}는 오른쪽 바퀴의 질량이다.

각방향의 각속도는 다음과 같다.

$$w_{bx} = 0, w_{by} = \dot{\theta}_R, w_{bz} = \dot{\phi} \tag{19}$$

오른쪽 바퀴의 선속도는 다음과 같이 표현된다.

$$\begin{aligned} v_{RW} &= v_D + w_C \times (-\frac{L}{2} a_y) \\ &= \dot{x}b_x + \dot{y}b_y + \dot{\phi}a_z \times (-\frac{L}{2} a_y) \\ &= \dot{x}b_x + \dot{y}b_y + \frac{L}{2}\dot{\phi}a_x \end{aligned} \tag{20}$$

$a_x = b_x\cos\phi + b_y\sin\phi$이므로 (20)에 대입해서 정리하면 아래와 같다.

$$\begin{aligned} v_{RW} &= \dot{x}b_x + \dot{y}b_y + \frac{L}{2}\dot{\phi}(b_x\cos\phi + b_y\sin\phi) \\ &= (\dot{x} + \frac{L}{2}\dot{\phi}\cos\phi)b_x + (\dot{y} + \frac{L}{2}\dot{\phi}\sin\phi)b_y \end{aligned} \tag{21}$$

그러므로 오른쪽 바퀴의 운동에너지 K_{RW}는 (20)을 (18)에 대입하여 다음과 같이 구할 수 있다.

$$\begin{aligned} K_{RW} &= \frac{1}{2}[I_{by}\dot{\theta}_R^2 + I_{bz}\dot{\phi}^2] \\ &+ \frac{1}{2} m_{RW}[\dot{x} + \dot{y} + (\frac{L}{2})^2\dot{\phi}^2 + 2(\frac{L}{2})\dot{\phi}(\dot{x}\cos\phi + \dot{y}\sin\phi)] \end{aligned} \tag{22}$$

마찬가지로 왼쪽 바퀴의 운동에너지 K_{LW}를 구하면 다음과 같다.

$$K_{LW} = \frac{1}{2}[I_{bx}^L w_{bx}^2 + I_{by}^L w_{by}^2 + I_{bz}^L w_{bz}^2] + \frac{1}{2} m_{LW} v_{LW}^2 \tag{23}$$

여기서 $I_{bx}^L = I_{bz}^L = \frac{1}{4} m_{LW} r^2, I_{by}^L = \frac{1}{2} m_{LW} r^2$이고 m_{LW}는 왼쪽 바퀴의 질량이다.

각 방향의 각속도는 다음과 같고

$$w_{bx} = 0, w_{by} = \dot{\theta}_L, w_{bz} = \dot{\phi} \tag{24}$$

선속도는

$$\begin{aligned} v_{LW} &= v_D + w_C \times \frac{L}{2} a_y \\ &= \dot{x}b_x + \dot{y}b_y + \dot{\phi}a_z \times \frac{L}{2} a_y \\ &= \dot{x}b_x + \dot{y}b_y - \frac{L}{2}\dot{\phi}a_x \end{aligned} \tag{25}$$

$a_x = b_x\cos\phi + b_y\sin\phi$이므로 대입하면

$$\begin{aligned} v_{LW} &= \dot{x}b_x + \dot{y}b_y - \frac{L}{2}\dot{\phi}(b_x\cos\phi + b_y\sin\phi) \\ &= (\dot{x} - \frac{L}{2}\dot{\phi}\cos\phi)b_x + (\dot{y} - \frac{L}{2}\dot{\phi}\sin\phi)b_y \end{aligned} \tag{26}$$

그러므로 왼쪽 바퀴의 운동에너지 K_{LW}는 다음과 같다.

$$\begin{aligned} K_{LW} &= \frac{1}{2}[I_{by}\dot{\theta}_L^2 + I_{bz}\dot{\phi}^2] \\ &+ \frac{1}{2} m_{LW}[\dot{x} + \dot{y} + (\frac{L}{2})^2\dot{\phi}^2 - 2(\frac{L}{2})\dot{\phi}(\dot{x}\cos\phi + \dot{y}\sin\phi)] \end{aligned} \tag{27}$$

따라서, 이동로봇의 전체 운동에너지 K는 (17), (22), (27)을 통합하여 다음과 같이 표현된다.

$$\begin{aligned} K &= K_C + K_{RW} + K_{LW} \\ &= \frac{1}{2} I_C\dot{\phi}^2 + \frac{1}{2} m_C[\dot{x}^2 + \dot{y}^2 + d^2\dot{\phi}^2 - 2d\dot{\phi}(\dot{x}\sin\phi - \dot{y}\cos\phi)] \\ &+ \frac{1}{2}[I_{by}^R\dot{\theta}_R^2 + I_{bz}^R\dot{\phi}^2] + \frac{1}{2} m_{RW}[\dot{x}^2 + \dot{y}^2 \\ &+ (\frac{L}{2})^2\dot{\phi}^2 + 2(\frac{L}{2})\dot{\phi}(\dot{x}\cos\phi + \dot{y}\sin\phi)] \\ &+ \frac{1}{2}[I_{by}^L\dot{\theta}_L^2 + I_{bz}^L\dot{\phi}^2] + \frac{1}{2} m_{LW}[\dot{x}^2 + \dot{y}^2 + (\frac{L}{2})^2\dot{\phi}^2 \\ &- 2(\frac{L}{2})\dot{\phi}(\dot{x}\cos\phi + \dot{y}\sin\phi)] \end{aligned} \tag{28}$$

(28)을 정리하면 다음과 같다.

$$\begin{aligned} K =& \frac{1}{2}(m_C+m_{RW}+m_{LW})(\dot{x}^2+\dot{y}^2) \\ &+ m_C d\dot{\phi}(\dot{y}\cos\phi-\dot{x}\sin\phi)+\frac{1}{2}I_{by}^R\dot{\theta}_R^2+\frac{1}{2}I_{by}^L\dot{\theta}_L^2 \\ &+ \frac{1}{2}[I_C+m_C d^2+I_{bz}^R+m_{RW}(\frac{L}{2})^2+I_{bz}^L+m_{LW}(\frac{L}{2})^2]\dot{\phi}^2 \end{aligned} \quad (29)$$

(29)에서 오른쪽 바퀴와 왼쪽 바퀴의 회전관성이 같다고 하면 다음과 같다.

$$I_w = I_{by}^R = I_{by}^L = \frac{1}{2}m_W R^2 \quad (30)$$

이동 로봇의 전체질량을 m이라 하면 다음과 같이 표현된다.

$$m = m_c + m_{RW} + m_{LW} = m_c + 2m_W \quad (31)$$

로봇의 회전속도 $\dot{\phi}$에 관련된 관성들을 모두 I라 설정하면

$$\begin{aligned} I =& I_C + m_C d^2 + I_{bz}^R + m_{RW}(\frac{L}{2})^2 + I_{bz}^L + m_{LW}(\frac{L}{2})^2 \\ =& I_C + m_C d^2 + 2I_{bz} + 2m_W(\frac{L}{2})^2 \end{aligned} \quad (32)$$

(30), (31), (32)를 (29)에 대입하여 정리하면 이동로봇 전체 운동에너지는 다음과 같이 간단하게 표현할 수 있다.

$$\begin{aligned} K =& \frac{1}{2}m(\dot{x}^2+\dot{y}^2)+m_C d\dot{\phi}(\dot{x}\cos\phi-\dot{y}\sin\phi) \\ &+\frac{1}{2}I_w(\dot{\theta}_R^2+\dot{\theta}_L^2)+\frac{1}{2}I\dot{\phi}^2 \end{aligned} \quad (33)$$

이동로봇의 위치에너지는 없기 때문에 (33)은 라그란지안 함수 L과 같으므로 다음과 같이 (33)을 미분하여 구한 전체 동력학식은 다음과 같다.

$$\begin{aligned} &\frac{d}{dt}[\frac{\partial L}{\partial \dot{q}_i}]-\frac{\partial L}{\partial q_i} = \\ &= \begin{bmatrix} m & 0 & -m_C d\sin\phi & 0 & 0 \\ 0 & m & m_C d\cos\phi & 0 & 0 \\ -m_C d\sin\phi & m_C d\cos\phi & I & 0 & 0 \\ 0 & 0 & 0 & I_W & 0 \\ 0 & 0 & 0 & 0 & I_W \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \\ \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} \\ &+ \begin{bmatrix} -m_C d\dot{\phi}^2\cos\phi \\ -m_C d\dot{\phi}^2\sin\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (34)$$

(34)를 동력학 식에 맞게 다시 정리하면 각 행렬 및 벡터는 다음과 같다.

$$M(q)\ddot{q}+V(q,\dot{q}) = P(q)\tau - A^T\lambda \quad (35)$$

여기서

$$M(q) = \begin{bmatrix} m & 0 & -m_c d\sin\phi & 0 & 0 \\ 0 & m & m_c d\cos\phi & 0 & 0 \\ -m_c d\sin\phi & m_c d\cos\phi & I & 0 & 0 \\ 0 & 0 & 0 & I_W & 0 \\ 0 & 0 & 0 & 0 & I_W \end{bmatrix}$$

$$V(q,\dot{q}) = \begin{bmatrix} -m_C d\dot{\phi}^2\cos\phi \\ -m_C d\dot{\phi}^2\sin\phi \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad P(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

$$A(q) = \begin{bmatrix} -\sin\phi & \cos\phi & -d & 0 & 0 \\ \cos\phi & \sin\phi & \frac{L}{2} & -r & 0 \\ \cos\phi & \sin\phi & -\frac{L}{2} & 0 & -r \end{bmatrix}$$

$M(q)$는 $n\times n$ 관성행렬, $V(q,\dot{q})$는 $n\times 1$ 원심력 및 코리올리스 벡터, $P(q)$는 $n\times r$ 입력 변환 행렬, τ는 $r\times 1$ 입력 토크 벡터, $A(q)$는 $m\times n$ 자코비안 행렬, 그리고 λ는 $m\times 1$ 라그란지안 multiplier 또는 구속된 힘의 벡터를 나타낸다.

먼저 동력학식을 간단히 하기 위해 라그란지안 multiplier를 소거해 보자. 속도는 다음과 같이 나타낼 수 있다.

$$\dot{q} = S(q)\nu(t) \quad (36)$$

여기서 S는 $\dot{q}$와 ν사이의 관계 행렬이고 $\dot{q}=[\dot{x}\ \dot{y}\ \dot{\phi}\ \dot{\theta}_R\ \dot{\theta}_L]^T$이고 $\nu(t)=[\nu_1\ \nu_2 \cdots \nu_{n-m}]^T$이다. (36)을 한 번 미분하면

$$\ddot{q} = \dot{S}(q)\nu(t)+S(q)\dot{\nu}(t) \quad (37)$$

(37)을 (35)에 대입하고 정리하면 다음과 같다.

$$M(q)[\dot{S}(q)\nu(t)+S(q)\dot{\nu}(t)]+V(q,\dot{q}) = P(q)\tau - A^T\lambda \quad (38)$$

$S^TA^T\lambda=0$를 만족하도록 양변에 S^T를 곱하면

$$S^T[M(q)(\dot{S}(q)\nu(t)+S(q)\dot{\nu}(t))+V(q,\dot{q})] = S^TP(q)\tau \quad (39)$$

(39)를 정리하면 다음과 같은 형태가 된다.

$$S^TM(q)S(q)\dot{\nu}(t)+S^TM(q)\dot{S}(q)\nu(t)+S^TV(q,\dot{q}) = S^TP(q)\tau \quad (40)$$

여기서 $S^TP=I_{2\times 2}$, 이므로 $S^TP\tau=\tau$가 된다. 따라서 다음과 같은 일반 산업 로봇의 동력학 식과 같은 형태가 된다.

$$S^TM(q)S(q)\dot{\nu}(t)+S^TM(q)\dot{S}(q)\nu(t)+S^TV(q,\dot{q}) = \tau \quad (41)$$

(41)에서 $\nu(t)=[\dot{\theta}_R\ \dot{\theta}_L]^T$를 대입하면 다음과 같다.

$$S^TM(q)S(q)\ddot{\theta}(t)+S^TM(q)\dot{S}(q)\dot{\theta}(t)+S^TV(q,\dot{q}) = \tau \quad (42)$$

여기서 행렬 $S(q)$는 다음과 같이 $S^T(q)A^T(q)=0$를 만족하도록 주어진다.

$$S(q) = \begin{bmatrix} \frac{r}{2}\cos\phi-\frac{rd}{L}\sin\phi & \frac{r}{2}\cos\phi+\frac{rd}{L}\sin\phi \\ \frac{r}{2}\sin\phi+\frac{rd}{L}\cos\phi & \frac{r}{2}\sin\phi-\frac{rd}{L}\cos\phi \\ \frac{r}{L} & -\frac{r}{L} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (43)$$

여기서 $b=\frac{L}{2}$, $c=\frac{r}{L}$로 정의하면 $S(q)$는 다음과 같이 간단해 진다.

$$S(q) = \begin{bmatrix} c(b\cos\phi-d\sin\phi) & c(b\cos\phi+d\sin\phi) \\ c(b\sin\phi+d\cos\phi) & c(b\sin\phi-d\cos\phi) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (44)$$

(44)의 $S(q)$행렬은 (35)의 $A(q)$행렬을 통해 $A(q)S(q)=0$ 또는 $S^T(q)A^T(q)=0$를 만족하도록 구해진다.
(42)에 (44)를 대입해서 정리하면 결국 조인트 공간에서의 토크 방정식은 다음과 같다.

$$D_{11}\ddot{\theta}_R + D_{12}\ddot{\theta}_L + 2c^3bd(m+m_c)(\dot{\theta}_R\dot{\theta}_L - \dot{\theta}_L^2) = \tau_R$$
$$D_{21}\ddot{\theta}_R + D_{22}\ddot{\theta}_L + 2c^3bd(m+m_c)(\dot{\theta}_R\dot{\theta}_L - \dot{\theta}_R^2) = \tau_L \quad (45)$$

여기서 각 관성성분은 다음과 같다.

$$D_{11} = c^2m(b^2+d^2) + 2c^2d^2m_c + c^2I + I_w$$
$$D_{12} = c^2[(m(b^2-d^2) - 2d^2m_c - I]$$
$$D_{21} = c^2[(m(b^2-d^2) - 2d^2m_c - I]$$
$$D_{22} = c^2m(b^2+d^2) + 2c^2d^2m_c + c^2I + I_w$$

무게 중심이 바퀴의 중심축의 연장선상에 있을 때, 즉 $d=0$일 때 동력학식은 다음 (46)과 같이 간단하게 된다. 결국 이동로봇은 원심력이나 코리올리스 힘에 관계없이 간단한 inertial 시스템으로 간주될 수 있다. 이러한 이유로 이동로봇의 제어에서는 동력학을 기반으로 제어하기보다는 간단하게 기구학을 기준으로 제어하는 경우가 많다[1-3, 7-13].

$$\begin{bmatrix} mb^2c^2+Ic^2+I_w & c^2(mb^2-I) \\ c^2(mb^2-I) & mb^2c^2+Ic^2+I_w \end{bmatrix}\begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (46)$$

로봇의 위치제어를 하기 위해서는 (3)의 기구학 변환을 통해 조인트 공간과 카테시안 공간과의 관계를 나타내는 자코비안을 고려해야 한다. 본 논문에서는 동력학식이 주어지지 않는 상태에서 로봇의 위치를 제어하는 실험을 하고자 한다.

IV. 오차 좌표계

이동 로봇의 위치를 추종하고자 할 때 다음과 같이 오차 좌표계를 설정한다[5,7]. 목표 위치의 좌표 $P_r=[x_r\, y_r\, \phi_r]^T$에서 현재 로봇의 좌표로 빼면 오차 좌표계가 구해진다. 아래 그림 2에서는 새로운 오차 좌표계를 구성하여 전역적으로 가능함을 나타낸다.

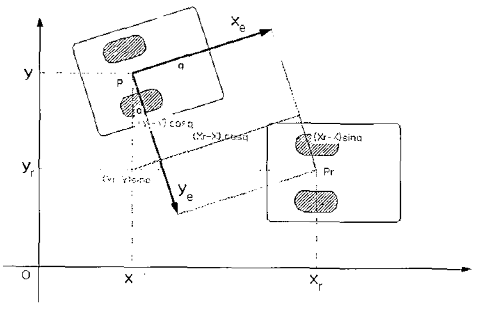


그림 2 새로운 오차 좌표계의 설정.
Fig. 2. New Error Transformation.

$[x_r-x,\ y_r-y]^T$에서 $[x_e,\ y_e]^T$로 전역 변환 좌표는 오차 변수가 전체 좌표계에서 각각 독립적으로 선택할 수 있게 만든다. 그림 2로부터 새로운 좌표계에서의 출력 오차 식은 다음과 같다.

$$\begin{aligned} x_e &= \cos\theta(x_r-x) + \sin\theta(y_r-y) \\ y_e &= -\sin\theta(x_r-x) + \cos\theta(y_r-y) \\ \phi_e &= \phi_r - \phi \end{aligned} \quad (47)$$

이러한 오차 식을 통해 기구학을 근거로 한 다양한 제어 방식이 제안되었다[10-13].

V. 제어 방식

1. 제어방식 1 : 엔코더 기반의 PD 제어

제어방식 1은 로봇 내부에 장착되어 있는 엔코더를 사용하는 엔코더 기반의 PD 제어 방식을 말한다. 엔코더 정보에만 의존하는 이 방식은 바퀴와 마찰면과의 미끄러짐에 의해 발생되는 오차와 이동로봇 자체의 구조에 의한 불균형 등에 의해 생기는 오차, 그리고 헤딩각 정보의 부재 때문에 dead-reckoning 상태에 처하게 되어 정확한 위치 제어가 어렵게 된다. 그림 3은 로봇이 내부적으로 엔코더 정보에 의존하는 PD 제어 방식을 보여주고 있다. 로봇이 자율 주행시 바퀴의 회전에 의한 제어는 헤딩각을 측정할 수 있는 센서 정보가 필요하지만, 이 경우에는 단지 엔코더 정보에 의존하여 위치를 제어한다.

결과적으로 이 제어 방식은 (46)의 동력학 식을 근거로 각 바퀴의 토크를 제어한 것이다. 물론 조인트 공간과 직교공간과의 관계를 고려하지 않아 dead-reckoning에 처하게 된다.

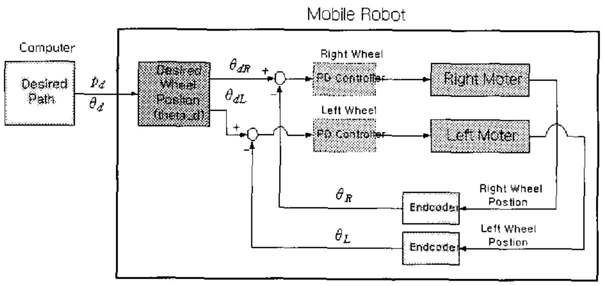


그림 3. PD 제어 방식.
Fig. 3. PD control.

엔코더의 정보를 받아 입력받은 값과 비교하여 오차를 만들어 PD제어를 하고 있다. 이동로봇의 조인트 공간에서의 제어법칙은 다음과 같다.

$$\tau = K_D(\dot{\theta}_d - \dot{\theta}) + K_P(\theta_d - \theta) \quad (48)$$

여기서 K_D, K_P는 2 x 2 제어 이득값이고, θ_d는 원하는 바퀴의 회전수의 2 x 1 벡터 $\theta_d=[\theta_{dR}\,\theta_{dL}]^T$, 그리고 θ는 실제 바퀴의 회전수의 2 x 1 벡터 $\theta=[\theta_R\theta_L]^T$로 엔코더 데이터를 나타낸다.

2. 제어 방식 2 : 비전기반 PD제어

Dead_reckoning 방식의 경우 기준 좌표가 로봇이 되기 때문에 자체의 오차를 수정할 방법이 없다. 즉, 헤딩각 센서를 사용한다고 하더라도 센서의 정확도에 따라 성능이 좌우된다[18]. 예를 들어 자이로 센서는 오차가 누적되는 현상이

발생하고 마그네틱 컴파스는 자장에 의해 영향을 받게된다. 이처럼 상대 센서의 오차의 문제를 해결하기 위해 엔코더 대신 비전 센서만을 이용하여 이동로봇을 제어할 수 있다. 비전 시스템을 사용하면 로봇 위에 장착된 마커를 통해 로봇의 위치와 자세정보를 알아내고 로봇이 움직여야 할 위치와 현재 위치를 계산하여 오차를 만든다. 이 오차를 사용하여 PD 제어기를 통과하면 로봇 구동부의 제어 입력을 만든 다음, 무선으로 그 값을 로봇에게 전송하여 로봇을 구동한다. 모든 계산은 PC에서 하고 모터의 PWM 값만 무선으로 전송하는 방식이다.

그림 4에 비전 정보 기반의 PD 제어 방식이 잘 나타나 있다.

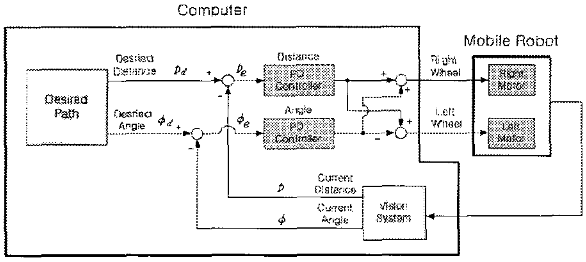


그림 4. 비전 PD 제어 방식.

Fig. 4. Vision Based PD Control.

비전 정보를 통해 (47)로부터 다음과 같이 직교좌표의 오차를 만든다.

$$
\begin{aligned}
p_e &= \sqrt{x_e^2 + y_e^2} \\
\phi_e &= \phi_d - \phi
\end{aligned}
\tag{49}
$$

여기서 p_e는 위치에 대한 오차이고 ϕ_e는 오리엔테이션에 대한 오차이다.

(49)의 오차로 구성된 각 바퀴의 PD 제어입력은 다음과 같다.

$$
\begin{aligned}
\tau_R &= k_{pp}p_e + k_{dp}\dot{p}_e + k_{p\phi}\phi_e + k_{d\phi}\dot{\phi}_e \\
\tau_L &= k_{pp}p_e + k_{dp}\dot{p}_e - (k_{p\phi}\phi_e + k_{d\phi}\dot{\phi}_e)
\end{aligned}
\tag{50}
$$

여기서 $\dot{p}_e, \dot{\phi}_e$는 p_e, ϕ_e를 미분한 것이고 k_{pp}, k_{dp}는 위치제어 이득값이고 $k_{p\phi}, k_{d\phi}$는 헤딩각제어 이득값이다. 로봇이 회전시에는 바퀴의 회전 방향이 틀리게 되므로 헤딩각 제어부분의 사인을 반대로 해야만 회전이 가능하다. 궁극적으로 제어 법칙의 목적은 직교좌표의 오차를 영으로 만드는 것이다. 여기서 계산된 토크값은 PWM으로 바뀌어 로봇에게 무선으로 전송된다.

3. 제어 방식 3 : 비전기반 신경회로망 PD 제어

제어 방식 3은 비전 기반 PD 제어방식인 제어방식 2에 신경회로망을 추가한 방식으로 그림 5에 잘 나타나 있다. 여기서 신경회로망의 사용 목적은 PD 제어로 인해 발생하는 추종 오차를 더 줄이기 위함이다. 신경회로망의 연산은 비전 처리와 함께 따로 컴퓨터에서 행해지며 무선으로 결과적인 토크값을 로봇에게 전달한다.

PD 제어기를 사용했으므로 입력 τ를 다음과 같이 정의한다. (50)으로부터 신경회로망의 출력을 입력에 더하면 다음과 같다.

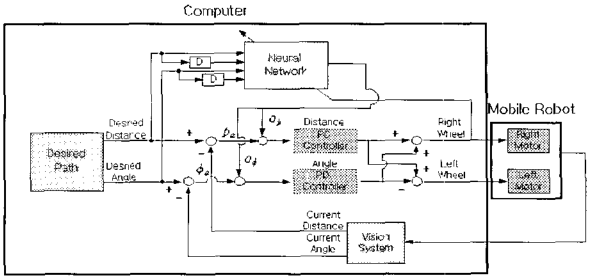


그림 5. 비전 PD 제어 + 신경회로망.

Fig. 5. Vision based PD Control Using Neural Network.

$$
\begin{aligned}
\tau_R &= k_{pp}(p_e + o_p) + k_{dp}(\dot{p}_e + \dot{o}_p) \\
&\quad + k_{p\phi}(\phi_e + o_\phi) + k_{d\phi}(\dot{\phi}_e + \dot{o}_\phi) \\
\tau_L &= k_{pp}(p_e + o_p) + k_{dp}(\dot{p}_e + \dot{o}_p) \\
&\quad - [k_{p\phi}(\phi_e + o_\phi) + k_{d\phi}(\dot{\phi}_e + \dot{o}_\phi)]
\end{aligned}
\tag{51}
$$

여기서 o_p, o_ϕ은 각각 신경회로망의 출력이고 $\dot{o}_p, \dot{o}_\phi$는 신경망 출력을 미분한 값들이다.

On-line 학습 제어를 실현하기 위해서는 학습신호를 잘 설정하여야 한다. 일반적으로 유용한 오차로 설정하는데 (51)에서 신경회로망을 학습하기 위한 학습신호는 p_e, ϕ_e를 기준으로 다음과 같이 정의한다.

$$
\begin{aligned}
tn_R &= k_{pp}p_e + k_{dp}\dot{p}_e + k_{p\phi}\phi_e + k_{d\phi}\dot{\phi}_e \\
tn_L &= k_{pp}p_e + k_{dp}\dot{p}_e - (k_{p\phi}\phi_e + k_{d\phi}\dot{\phi}_e)
\end{aligned}
\tag{52}
$$

(52)를 (51)에 대입하면 (51)은 다음과 같다.

$$
\begin{aligned}
tn_R &= \tau_R - (k_{pp}o_p + k_{dp}\dot{o}_p + k_{p\phi}o_\phi + k_{d\phi}\dot{o}_\phi) \\
tn_L &= \tau_L - (k_{pp}o_p + k_{dp}\dot{o}_p - k_{p\phi}o_\phi - k_{d\phi}\dot{o}_\phi)
\end{aligned}
\tag{53}
$$

(53)에서 학습신호가 영으로 수렴하도록 신경회로망을 학습하는 것이 목적이므로 이때 신경회로망의 학습 신호를 $tn = [tn_R \; tn_L]^T$이라 하고 목적함수는 다음과 같이 정의한다.

$$
E = \frac{1}{2} tn^T tn \tag{54}
$$

역전파 알고리즘을 적용하기 위해서는 그래디언트 값을 구한다.

$$
\frac{\partial E}{\partial w} = \frac{\partial E}{\partial tn}\frac{\partial tn}{\partial w} = tn^T \frac{\partial tn}{\partial w} \tag{55}
$$

(53)에서 살펴보면 $\frac{\partial tn}{\partial w}$를 각각 구할 수 있다.

$$
\begin{aligned}
\frac{\partial tn_R}{\partial w} &= -(k_{pp}\frac{\partial o_p}{\partial w} + k_{dp}\frac{\partial \dot{o}_p}{\partial w} + k_{p\phi}\frac{\partial o_\phi}{\partial w} + k_{d\phi}\frac{\partial \dot{o}_\phi}{\partial w}) \\
\frac{\partial tn_L}{\partial w} &= -(k_{pp}\frac{\partial o_p}{\partial w} + k_{dp}\frac{\partial \dot{o}_p}{\partial w} - k_{p\phi}\frac{\partial o_\phi}{\partial w} - k_{d\phi}\frac{\partial \dot{o}_\phi}{\partial w})
\end{aligned}
\tag{56}
$$

역전파 알고리즘을 사용해서 가중치 값을 매 샘플마다 업데이트한다. 출력층이나 은닉층의 가중치값을 업데이트하고자 할 때 아래와 같이 하면 된다.

$$
w(t+1) = w(t) + \eta \frac{\partial E}{\partial w} \tag{57}
$$

(53)에서 오차가 영으로 수렴하면 아래와 같이 inverse dynamics control을 이루게 된다.

$$\begin{aligned} \tau_R &= (k_{pp}o_p + k_{dp}\dot{o}_p + k_{p\phi}o_\phi + k_{d\phi}\dot{o}_\phi) \\ \tau_L &= (k_{pp}o_p + k_{dp}\dot{o}_p - k_{p\phi}o_\phi - k_{d\phi}\dot{o}_\phi) \end{aligned} \quad (58)$$

4. 제어방식 4:엔코더 기반 PD 제어+비전 기반 PD 제어

아래 그림 6은 내부 엔코더 PD 제어와 외부 비전 PD 제어를 사용한 경우를 나타낸다. 내부 루프는 엔코더 기반의 PD 제어이고 외부 루프는 비전 기반의 PD 제어를 사용하였다. 이 제어 구조는 시스템 자체의 제어기를 수정하지 않고 외부에서 시스템의 출력을 조절할 수 있는 장점이 있다. 외부 PD 제어는 내부 제어기로 인해 발생하는 오차를 줄인다. 이 제어 방식의 문제는 외부 루프의 샘플링시간이 충분히 빨라야 내부 제어기로 생기는 dead-reckoning을 보정하여 오차를 줄일 수 있다.

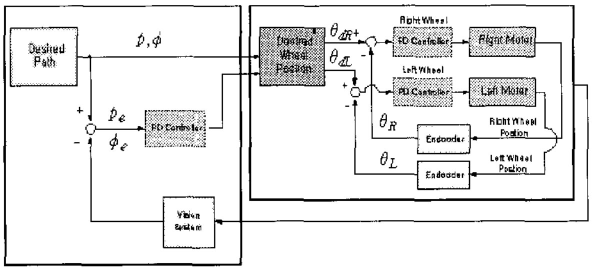


그림 6. 내부 엔코더 PD 제어 + 외부 비전 PD 제어.
Fig. 6. Encoder PD control + Vision PD Control.

비전을 통하여 현재 로봇의 자세 오차를 구하여 다음에 로봇이 이루어야할 자세를 수정하여 전달한다. 각 바퀴를 구동하는 토크는 다음과 같이 조인트 공간에서의 PD 제어기에 직교좌표 공간의 PD 제어기가 더해진 형태로 나타내어진다.

$$\begin{aligned} \tau_R &= k_{pp}[\theta_{dR} - \theta_R + k^j_{pp}p_e + k^j_{p\phi}\phi_e] \\ &\quad + k_{dp}[\dot{\theta}_{dR} - \dot{\theta}_R + k^j_{dp}\dot{p}_e + k^j_{d\phi}\dot{\phi}_e] \\ \tau_L &= k_{pp}[\theta_{dL} - \theta_L + k^j_{pp}p_e \\ &\quad - k^j_{p\phi}\phi_e] + k_{dp}[\dot{\theta}_{dL} - \dot{\theta}_L + k^j_{dp}\dot{p}_e - k^j_{d\phi}\dot{\phi}_e] \end{aligned} \quad (59)$$

여기서 θ_{dR}, θ_{dL}은 각각 원하는 오른쪽, 왼쪽 바퀴의 회전수이고 θ_R, θ_L는 로봇의 오른쪽, 왼쪽 바퀴의 회전수이고 $k^j_{pp}, k^j_{p\phi}, k^j_{dp}, k^j_{d\phi}$는 자코비안을 고려한 이득값들이다.

5 제어방식 5 : 신경회로망을 이용한 PD 제어

제어방식 5는 제어방식 4에 신경회로망을 추가한 것으로 그림 7에 나타나 있다.

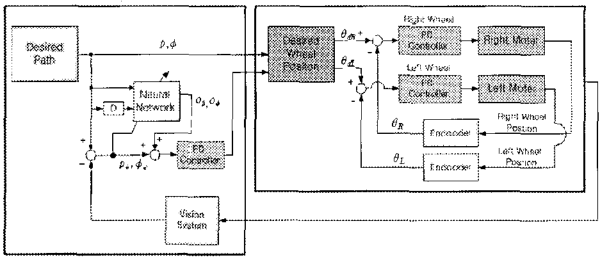


그림 7. 내부 PD 제어 + 외부 신경회로망제어.
Fig. 7. Encoder Based PD Control Using NN.

이 제어방식은 제어방식 4에 출력되는 값에 신경회로망의 출력을 더한 것으로 PC에서 출력되는 최종 출력은 다음과 같다.

PD 제어기를 사용했으므로 입력 τ를 다음과 같이 정의한다.

$$\begin{aligned} \tau_R &= k_{pp}[\theta_{dR} - \theta_R + k^j_{pp}(p_e + o_p) + k^j_{p\phi}(\phi_e + o_\phi)] \\ &\quad + k_{dp}[\dot{\theta}_{dR} - \dot{\theta}_R + k^j_{dp}\dot{p}_e + k^j_{d\phi}\dot{\phi}_e] \\ &= k_p p_e + k_{dp}\dot{p}_e + k_\phi\phi_e + k_{d\phi}\dot{\phi}_e + k_{pp}o_p \\ &\quad + k_{p\phi}o_\phi + k_{pp}(\theta_{dR} - \theta_R) + k_{dp}(\dot{\theta}_{dR} - \dot{\theta}_R) \\ \tau_L &= k_{pp}[\theta_{dL} - \theta_L + k^j_{pp}(p_e + o_p) - k^j_{p\phi}(\phi_e + o_\phi)] \\ &\quad + k_{dp}[\dot{\theta}_{dL} - \dot{\theta}_L + k^j_{dp}\dot{p}_e - k^j_{d\phi}\dot{\phi}_e] \\ &= k_p p_e + k_{dp}\dot{p}_e - (k_\phi\phi_e + k_{d\phi}\dot{\phi}_e) + k_{pp}o_p \\ &\quad - k_{p\phi}o_\phi + k_{pp}(\theta_{dL} - \theta_L) + k_{dp}(\dot{\theta}_{dL} - \dot{\theta}_L) \end{aligned} \quad (60)$$

여기서 o_p, o_ϕ은 각각 신경회로망의 출력이고 $k_p = k_{pp}k^j_{pp}$, $k_d = k_{dp}k^j_{dp}, k_\phi = k_{p\phi}k^j_{p\phi}, k_{d\phi} = k_{d\phi}k^j_{d\phi}$이다.
학습신호를 다음과 같이 정의하면

$$\begin{aligned} tn_R &= k_p p_e + k_{dp}\dot{p}_e + k_\phi\phi_e + k_{d\phi}\dot{\phi}_e \\ tn_L &= k_p p_e + k_{dp}\dot{p}_e - (k_\phi\phi_e + k_{d\phi}\dot{\phi}_e) \end{aligned} \quad (61)$$

(60)은 다음과 같다.

$$\begin{aligned} tn_R &= \tau - [k_{pp}(\theta_{dR} - \theta_R) + k_{dp}(\dot{\theta}_{dR} - \dot{\theta}_R)] \\ &\quad - (k_{pp}o_p + k_{dp}\dot{o}_p + k_{p\phi}o_\phi + k_{d\phi}\dot{o}_\phi) \\ tn_L &= \tau - [k_{pp}(\theta_{dL} - \theta_L) + k_{dp}(\dot{\theta}_{dL} - \dot{\theta}_L)] \\ &\quad - (k_{pp}o_p + k_{dp}\dot{o}_p - k_{p\phi}o_\phi - k_{d\phi}\dot{o}_\phi) \end{aligned} \quad (62)$$

(62)를 영으로 수렴시키기 위해서 목적함수는 다음과 같다.

$$E = \frac{1}{2} tn^T tn \quad (63)$$

여기서 $tn = [tn_R tn_L]^T$이다. 역전파 알고리즘을 적용하기 위해서는

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial tn}\frac{\partial tn}{\partial w} = tn^T\frac{\partial tn}{\partial w} \quad (64)$$

(61)에서 살펴보면 $\frac{\partial tn}{\partial w}$를 각각 구할 수 있다.

$$\frac{\partial tn}{\partial w} = -\begin{bmatrix} k_{pp}\frac{\partial o_p}{\partial w} + k_{p\phi}\frac{\partial o_\phi}{\partial w} \\ k_{pp}\frac{\partial o_p}{\partial w} - k_{p\phi}\frac{\partial o_\phi}{\partial w} \end{bmatrix} \quad (65)$$

이동로봇의 내부동작은 제어방식 4와 같다.
제안한 제어방식들의 특징을 비교하면 표 1과 같다.

VI. 위치 제어 시뮬레이션

아래 그림은 앞장에서 구한 동력학 식을 모델로 하여 이동로봇의 위치 추종을 시뮬레이션 한 것이다. 동적 모델에 슬립이나 마찰력 등과 같은 불확실성을 포함하지 않고 정확한 모델로 시뮬레이션 하였다.

아래 그림 9에서는 이동로봇이 원하는 경로를 추종하고 있는 것을 보여준다. 원하는 경로는 초기 위치 (0,0)m에서 출발

표 1. 각 제어 방식 구조 비교

Table 1. Structural Comparison of Control Methods.

실험	엔코더 사용 유무	비젼 센서 사용유무	신경회로망 사용 유무
제어 방식 1	O	X	X
제어 방식 2	X	O	X
제어 방식 3	X	O	O
제어 방식 4	O	O	X
제어 방식 5	O	O	O

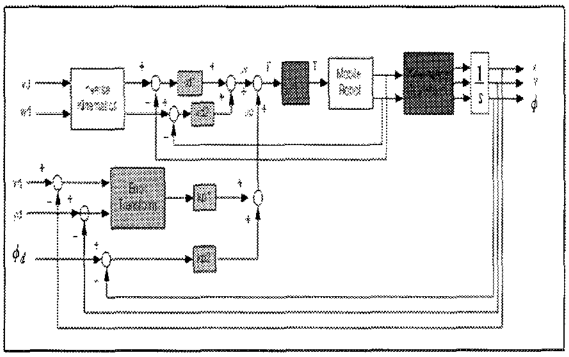


그림 8. 이동 로봇의 제어 블록도.

Fig. 8. Control Block Diagram of Mobile Robot.

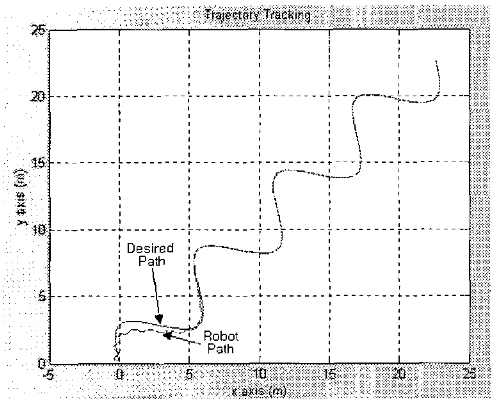


그림 9. 이동로봇의 움직임.

Fig 9. Movement of Mobile Robot

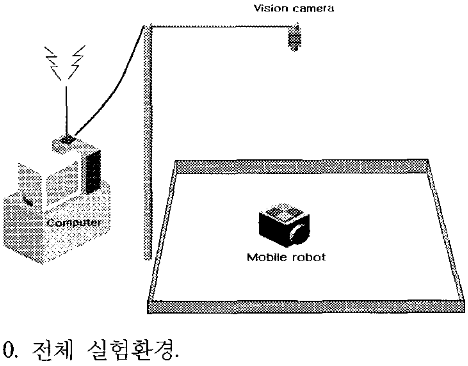


그림 10. 전체 실험환경.

Fig. 10. Experimental Setup.

하여 사인파 형태로 다양한 헤딩각을 나타내려 하였다. 실제 경로는 초기에 오차가 발생하다가 잘 추종하는 것을 볼 수 있다.

VII. 실험 환경

1. 전체구조

실험환경의 전체 구조는 다음 그림 10과 같다. 이동로봇, 비전 시스템, 그리고 컴퓨터로 구성되어 있다.

2. 이동로봇

이동로봇은 바퀴에 의해 조향이 가능한 wheeled drive 형태이다. 컨트롤러로는 80C196KC를 사용하였고 80C196KC 내부에는 소프트 타이머가 있어 내부적으로 일정 샘플링 타임(2msec)마다 제어가 가능하다. 하지만, 컴파일러에서 실수 연산이 지원되지 않아 소수점이하는 생략해 버리는 수가 많다. 따라서, 버려지는 data를 줄이기 위해 임의의 수를 곱해서 계산하고, 다시 곱해진 임의의 수를 나누는 방법으로 버려지는 수를 줄였다. 컨트롤러에서 엔코더를 읽을 수 있지만, 내부 연산이 많아지는 경우 엔코더를 제대로 카운터하지 못하고, 카운터를 놓치는 경우가 발생한다. 따라서, 엔코더를 읽기 위해 전용 카운터 칩인 7266(24bit)을 사용하여 메인 컨트롤러의 부하를 줄여주었다. 24bit까지 카운터가 가능한 7266을 사용함으로써 이동 로봇이 주행할 수 있는 거리를 모두 기록하도록 했다. 사용된 내부 제어기로는 PD 제어기를 사용하였다. 원하는 자세에 대한 값이 이동로봇에 전달되면, 이동로봇은 입력받은 값을 엔코더 값으로 환산하여 오차를 계산한다. 입력받은 값이 카운터 값으로 변환한 값을 $Desired_Dist_cnt$, $Desired_Angl_cnt$ 하자. Act_Dist_cnt, Act_Ang_cnt는 실제 엔코더 값으로부터 얻은 값이다.

$$\begin{aligned} Act_Dist_cnt &= \frac{L_cnt + R_cnt}{2} \\ Act_Ang_cnt &= \frac{L_cnt - R_cnt}{2} \end{aligned} \tag{66}$$

여기서 R_cnt, L_cnt는 각각 오른쪽, 왼쪽 바퀴의 엔코더 신호이다. 따라서, 오차값을 구하면 다음과 같다.

$$\begin{aligned} D_cnt &= Desired_Dist_cnt - Act_Dist_cnt \\ A_cnt &= Desired_Ang_cnt - Act_Ang_cnt \end{aligned} \tag{67}$$

D_cnt는 왼쪽과 오른쪽 엔코더 값을 읽어 실제 이동거리를 구한 것이다. 또한, A_cnt는 왼쪽과 오른쪽 엔코더 값 차이에 따른 이동로봇의 회전을 알기 위한 것이다. 실제 로봇에 인가되는 PWM_R, PWM_L값은 다음 식으로 계산된다.

$$\begin{aligned} PWM_R &= Kp \times D_cnt + Kd \times D_cnt_d \\ &\quad + (Kp \times A_cnt + Kd \times A_cnt_d) \\ PWM_L &= Kp \times D_cnt + Kd \times D_cnt_d \\ &\quad - (Kp \times A_cnt + Kd \times A_cnt_d) \end{aligned} \tag{68}$$

헤딩각 오차에 대해서는 PWM_R, PWM_L가 반대 값으로 입력되는 이유는 로봇이 회전을 해야되기 때문이다. D_cnt_d A_cnt_d는 이전 값과 현재 값을 sampling값으로 나눈 값이다.

사용된 구동 모터로는 Mini-Motor사의 FI 모델을 사용하였으며, 엔코더는 500 펄스로 기어비가 14:1이므로 한바퀴 회

전하는데 7000펄스가 출력된다. 따라서, 1펄스로 제어 가능한 이동 거리와 회전 각도는 0.0357mm, 0.0514도 이다. 로봇이 회전하는 경우 회전 반지름은 143mm(바퀴 사이거리=L/2)이다.

3. 비전 시스템

본 논문에서는 RGB를 사용하여 영상을 처리하였다. 영상처리는 비전 보드를 사용하지 않고, TV 수신 카드를 사용하여 영상을 입력받아 Direct X를 사용하여 처리하였다. 로봇은 기본적으로 로봇의 칼라와 그룹 칼라를 갖는다. 영상처리는 기본적으로 2진 화상을 만들기 위해 임계값 처리를 하고, 각각의 물체를 구분하기 위하여, 화소들이 서로 연결되어 있는가 또는 떨어져 있는가를 알기 위해 라벨링 작업을 한다. 카메라의 해상도나 조명의 불균일성 등에 의해, 2진 화상에는 잡음이 있을 수 있다. 이러한 잡음은 보통 불규칙적으로 발생하므로 잡음에 해당하는 연결성분들의 화소의 수는 작은 값을 갖는다. 따라서, 라벨링 후에 얻어진 연결성분들에 대해, 어느 화소 수 미만인 연결성분들을 제거하기 위해 크기 필터링을 한다. 크기 필터링을 거쳐 각각의 로봇 칼라와 그룹 칼라의 위치를 파악하게 된다. 이 정보는 면적 중심법을 통해 로봇의 방향과 위치를 파악하는데 사용된다.

카메라는 높이 2.4m의 높이에 설치되어 있으며, 사용된 카메라는 삼성 SDC-410을 사용하였다. 화상의 크기는 320×240 으로 실제 비율을 따져보면, 한 픽셀 당 약 5mm의 실제 크기를 나타낸다. 따라서, 카메라를 사용하는 경우 5mm 이하의 오차는 사실상 보정이 어렵다.

실험에 사용된 실제 시스템은 다음 그림 11과 같다.

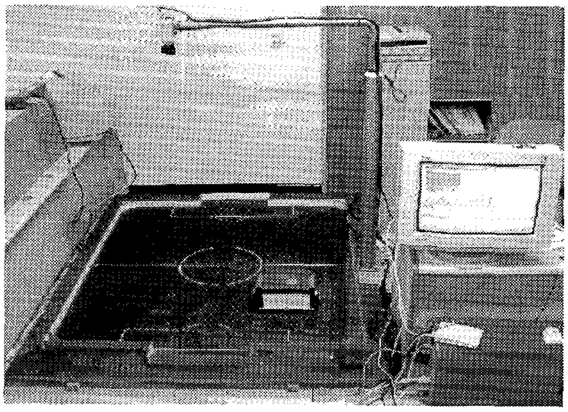


그림 11. 실제 실험 환경.
Fig. 11. Real Experimental Setup.

4. 신경회로망

신경회로망은 그림 12에서 보여진 입력층-은닉층-출력층으로 구성된 일반적인 feedforward multi-layered 신경회로망을 사용하였다. 입력으로는 기준 입력과 실제 상태 값의 차이인 오차를 사용하였으며, 은닉층의 뉴론수는 9개를 사용하였고 출력층은 모두 2개의 뉴론을 사용하였다. 출력층의 두 출력은 PD 제어기를 거치면서 각각 미분된 뒤에 더해진다. 학습율은 시행착오를 거쳐 최적화하였다. 실시간 신경회로망제어로는 DSP를 사용하여 처리하였다[19].

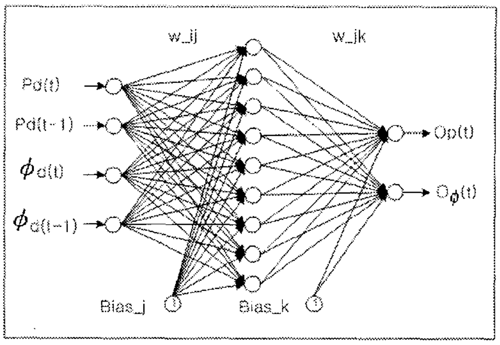


그림 12. 신경회로망 구조.
Fig. 12. Neural network structure.

VIII. 실험결과

1. 제어방식 1

그림 13에서 보면 직선주행 중에 로봇은 기준 궤적을 벗어나 천천히 오차가 커지고 있다. 오차가 커지는 원인은 엔코더 환산 계수 오차, 오른쪽 왼쪽 바퀴 사이의 거리오차와 로봇 바퀴와 바닥면사이의 미끄러짐 때문이다.

회전하는 부분에서 로봇이 오실레이션하는 것을 볼 수 있는데, 이 부분에서 큰 미끄러짐이 발생하여 엔코더 카운터 오차가 커졌기 때문이다. 내부적으로 오차를 수정하기 위한 PD 제어가 이루어지고 있지만, 미끄러짐에 의한 오차를 줄이지 못하고 있음을 볼 수 있다. 목표점에 도달하지 못한 것을 볼 수 있다.

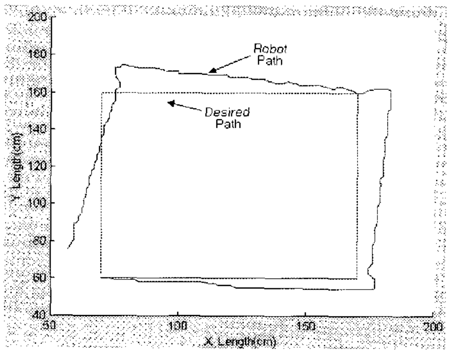


그림 13. PD 제어 : 제어방식 1.
Fig. 13. PD control : Scheme 1.

2. 제어방식 2

비전으로 visual feedback을 사용한 결과가 그림 14에 나타나 있다. 제어방식 1의 결과와 비교해 볼 때 성능이 많이 좋아진 것을 알 수 있다. 목표점에 정확하게 도달하였으나 코너 부분에서 회전시에 오버슈트가 발생한 것이 관찰되었다. PC에서는 20msec 마다 영상처리를 하여 error를 얻어 PD 제어를 하고 매 번 실제 모터를 구동하는데 사용되는 PWM값 을 출력하였다. 이 경우에는 영상처리를 얼마나 잘 하는가가 실험 결과에 큰 영향을 미친다.

3. 제어방식 3

그림 15는 제어방식 3의 결과를 보여준다. 제어방식 2와 비교해서 위치 추종 성능이 매우 좋아진 것을 알 수 있다.

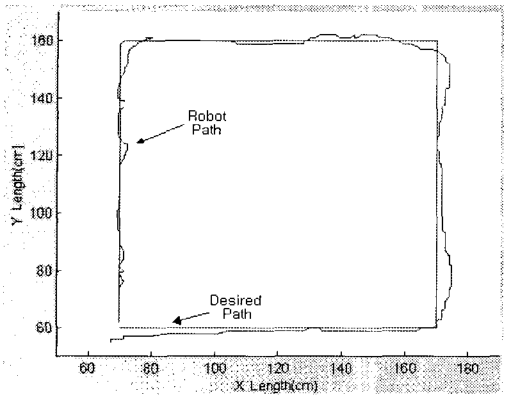


그림 14. 비전 PD 제어:제어방식 2.
Fig. 14 .Vision Based PD control : Scheme 2.

신경회로망을 추가한 경우에도 몇 가지 setting 해주어야 하는 값들이 있다. 특히, On-Line으로 실행되는 경우에는 하드웨어적인 성능에 따라 임의의 값으로 선택되는 bias값과 weight 값의 범위가 중요하다. 초기에 bias값과 weight 값이 너무 작으면 신경회로망이 성능을 발휘하는데 어느 정도의 시간이 소요된다. 또한, 학습율을 설정하는데 있어서도 몇 번의 실험을 거쳐 최적으로 설정했다. 학습율의 경우 실험에서는 η = 0.5를 사용하였고, 0.5가 넘는 경우 오히려 결과가 좋지 않았다.

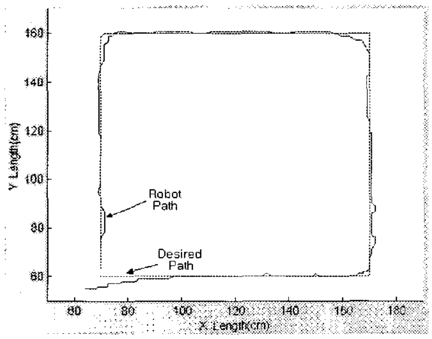


그림 15. 비전 신경회로망 제어 : 제어방식 3.
Fig. 15. Vision Based Control : Scheme 3.

4. 제어방식 4

방식1에 방식3을 더한 경우의 제어 성능을 알아보자. 로봇내부에 엔코더가 있고, 로봇의 자체 오차를 수정하기 위해 비전 센서를 사용하였다. 그림 16에서 보면 계속 오실레이션이 발생하는데 그 이유는 로봇 자체의 시스템 루프시간은 2msec 이지만 비전을 통하여 로봇에 명령이 전달되는 시간은 20msec로 10배가 느리기 때문이다. PC에서 비전 처리를 통하여 한번의 명령이 전달되면 다음 명령이 다시 전달될 때까지 로봇은 엔코더로부터 받은 정보를 이용하여 로봇의 위치 오차를 보정하려고 노력한다. 따라서, 다음 명령이 전달 될 때까지 로봇이 오차를 더 크게 만들 수 있다는 것이다. 결국, 비전에서 얻게되는 정보와 로봇이 구동하면서 얻게 되는 정보와의 샘플링 시간 문제가 발생하는 것이다. 이로 인해 계속된 오실레이션이 발생하게 된다. 오실레이션

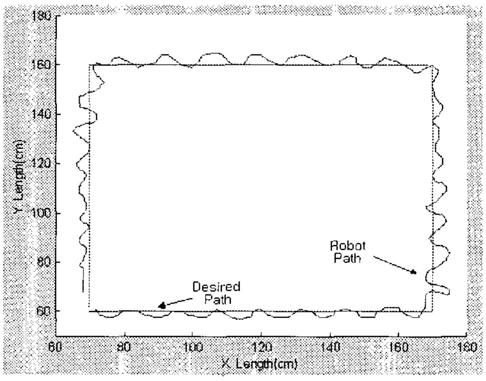


그림 16. 비전PD + PD 제어 : 제어방식 4.
Fig. 16. Vision based PD control + Encoder Based PD control : Scheme 4.

이 많은 이유 중 하나는 로봇이 주행하는데 있어서 먼 거리를 한번에 주행하도록 하지 않고 매 sampling time 마다 로봇의 원하는 위치와 자세 각을 주기 때문이다.

5. 제어방식 5

제어방식 5는 제어방식 4에서 나타난 오실레이션을 줄이기 위해 신경회로망을 사용하였다. 그림 17에서 보면 그림 16의 결과보다 위치 추종이 향상된 것을 볼 수 있다. 하지만, 제어방식 4와 마찬가지로 전체 시스템 루프 시간이 로봇 자체의 내부 루프보다 10배가 늦기 때문에 진동이 발생하는 것 자체를 없앨 수는 없었다. 하지만, 결과적으로 실험을 통하여 신경회로망을 사용했을 경우에 더 좋은 결과를 얻음을 볼 수 있다. 비전 처리에 있어 충분히 빠른 비전 전용보드를 사용하면 더 좋은 추종 결과를 얻을 수 있을 것이다.

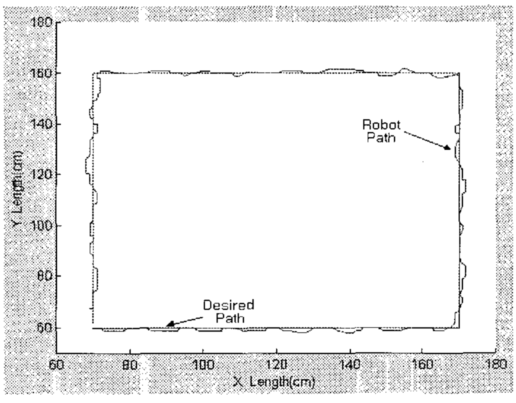


그림 17. 비전 신경회로망 PD 제어 : 제어방식 5.
Fig. 17. ision Based PD Control Using NN:Scheme 5.

IX. 결론

이동 로봇이 주어진 환경에서 작업하기 위해서는 자신의 위치를 알고 정확한 위치 추적이 이루어져야 한다. 엔코더값에 의한 위치 추정은 실시간으로 로봇의 위치를 가장 빠르고 쉽게 얻을 수 있지만, 오차의 누적으로 실용성이 떨어진다. 이런 오차를 보정하기 위해 절대 위치를 알 수 있는 비전 센서를 사용함으로써 기존의 PD제어기는 수정하지 않고 신경

회로망을 외부적으로 사용하여 실험하였다. 다양한 형태의 제어 알고리즘의 성능을 비교 분석해 보았는데, 실험 결과를 보면 신경회로망을 이용한 이동 로봇 tracking control이 좋은 성능을 보임을 확인 할 수 있다. 하지만 이동로봇의 내부 제어기를 사용하는 경우에는 외부 루프에서 영상을 실시간으로 처리하는데 걸리는 샘플링 시간이 로봇 제어기 루프의 샘플링보다 현저하게 늦어지면 보정하는데 있어서 어려움이 발생하는 것을 발견했다. 추후 연구로는 이 점을 보완하여 비전의 영상처리 속도를 빠르게 하여 위치 제어의 성능을 좋게 하는 것이다. 또한 실제 DGPS 센서를 사용하여 이동로봇을 제어하는 것이다.

참고문헌

[1] L. Jetto, S. Longhi, Giuseppe Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots", *IEEE Transactions on Robotics and Automation,* vol. 15, no. 2, pp. 219-229, April 1999.

[2] T. Larsen, K. Hansen, N. Anderson, O. Ravn,"Design of kalman filters for mobile robots ; Evaluation of the kinematics and odometric approach", *IEEE Control Applications,* pp. 1021-1026, 1999.

[3] C. C. Tsai, "A localiztion system of a mobile robot by fusing dead-reckoning and ultrasonic measurements", *IEEE Instrumentation and Measurement Technology,* pp. 144-149, 1998.

[4] S. Jung and T. C. Hsia, "Robust neural force control scheme under uncertainties in robot dynamics and unknown environment", *IEEE Trans. on Industrial Electronics*, pp. 403-412, vol. 47, no. 2, April, 2000.

[5] 정슬, 임선빈, "미정보 환경하에서의 신경회로망 힘추종 로봇제어 기술의 실험적 연구", 제어 · 자동화 · 시스템 공학 논문지, pp. 338-344, 제8권 제4호.

[6] P. W. Jeon, P. S. Jang and Seul Jung, "Path generation for collision avoidance of blind mobile robot using force sensor with memories", *FIRA 2002 World Congress on Robotics*, May 2002, Seoul, Korea.

[7] N. Sarkar, X. Yun, V. Kumar,"Control of mechanical systems with rolling constraints : Application to dynamic control of mobile robots", *The International Journal of Robotics Research,* vol. 13, no. 1, February 1994, pp. 55-69.

[8] Z. P. Jang and H. Nijmeijer,"Tracking control of mobile robots: A case study in backstepping," *Automatica,* vol. 33, no.7, pp. 1393-1399, 1997.

[9] W. Weiguo, C. Huitang, W. Yuejuan, "Backstepping design for path tracking of mobile robots", *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 1999, pp. 1822-1827.

[10] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi,"A stable tracking control method for an autonomous mobile robot," *in Proceedings of the 1990 IEEE International Conference on Robotics and Automation,* 1990, pp. 384-389.

[11] Y. Kanayama., Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for a non-holonomic mobile robot", *IEEE/RSJ International Workshop on Intelligent Robots and System IROS,* pp. 1236-1241, 1991.

[12] Y. J Kanayama and F. Fahroo, "A new line tracking method for nonholonomic vehicles", *IEEE Robotics and Automations*, pp. 2908-2913, 1997.

[13] S. Singh, and D. H. Shin, "Position based path tracking for wheeled mobile robots", Proc. of *IEEE IROS 89,* pp.386~391. 1989.

[14] 주진화, 이장명, "신경회로망을 이용한 이동 로봇의 실시간 고속 정밀제어", 제어 · 자동화 · 시스템공학회, pp. 95-104, 제5권 1호, 1999.

[15] 김무진, 이영진, 박성준, 이만형 "신경회로망을 이용한 이동 로봇의 정밀제어", 제어 · 자동화 · 시스템공학회, pp. 689-696, 제6권 8호, 2000.

[16] R. Fierro and F. Lewis, "Robust practical point stabilization of a nonholonomic mobile robot using neural networks", *Journal of Intelligent and Robotic Systems*, pp. 295-317, 1997.

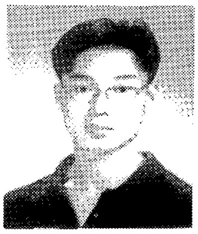
[17] W. K. Kim, D. H. Kim, B. J. Yi, B. J. You, "Kinematic modeling of mobile robots by transfer method of augmented generalized coordinates", *IEEE Robotics and Automations*, pp. 2413-2418, 2001

[18] 전상운, 정슬, 원문철, 홍 섭 , 최종수, "추측항법과 LBL 시스템 융합을 통한 UTV의 위치 추정", pp. 1899-1903, ICCAS, 2001, 제주.

[19] 정슬, 임선빈, "신경회로망을 이용한 비선형 시스템 제어의 실험적 연구", pp. 918-926, 제어 · 자동화 · 시스템 공학 논문지 제 7 권 제 11 호.

정 슬

1964년 9월 11일생. 1988년 미국 웨인 주립대 전기 및 컴퓨터 공학과 졸업. 1991 미국 캘리포니아 데이비스 석사. 1996년 동대학 박사. 1997년~현재 충남대학교 메카트로닉스공학과 부교수. 관심분야는 지능 제어 및 감성 공학, 로봇 설계 및 제어, SOC 제어기 설계.



장 평 수

1974년 11월 20일생, 2001년 충남대학교 메카트로닉스공학과 졸업(학사). 현재 동대학 석사과정. 관심분야는 이동로봇 제작 및 센서 융합 기반의 이동로봇 제어.

홍 섭

1959년 6월 2일생. 1983년 서울대학교 조선공학과 졸업. 1985년 동대학원 졸업, 1992년 독일 Aachen 대학교 박사. 현재 한국 해양 연구소 책임연구원. 관심분야는 해양장비 및 동력학.

원 문 철

1960년 10월 21일생. 1983년 서울대학교 조선공학과 및 1985년 동대학원 졸업, 1987년-1990년 기계연구원 연구원. 1994년 미국 캘리포니아 버클리 박사 및 PATH연구소 연구원. 1995년~현재 충남대 메카트로닉스공학과 부교수. 관심분야는 지능차량제어, 해양시스템제어.