

## 장남 이동 로봇의 힘 제어 : 분석, 시뮬레이션 및 실험

### Force Control of a Blind Mobile Robot : Analysis, Simulations and Experiments

전 풍 우, 정 슬\*

(Poong-Woo Jeon and Seul Jung)

**Abstract :** We propose a blind mobile robot force control algorithm that uses force information as a guidance toward to the goal position. Based on the mobile robot dynamics, the control law is formed from explicit force errors. Simulation studies are conducted based on the kinematics and the dynamics of the mobile robot. Simulation results show that good force tracking can be achieved. In order to confirm simulation results, experiments are performed. The robot is commanded to follow unknown environment with maintaining a certain desired force. Experimental results show that the blind mobile robot successfully maintains contact with a regulated desired force and arrives at the goal position.

**Keywords :** mobile robot, kinematics, dynamics, force tracking control, path generation.

#### I. 서론

일반적으로 이동로봇은 이동성을 가지고 있다는 점에서 다른 산업용 로봇 시스템과는 다른 특유의 장점을 가지고 있다. 이러한 이동성의 장점을 충분히 달성하기 위해서는 로봇의 자세 제어가 매우 중요한 요소가 된다.

현재 이러한 이동로봇의 자세 제어를 위한 방법으로는 매우 다양한 방법들이 연구되고 있는데, 대부분의 연구는 로봇의 자율주행에 관한 것이다[1-5]. 가장 기본적인 주행 방법으로는 로봇의 양쪽 바퀴에 장착되어있는 엔코더 센서를 사용한 Dead-reckoning 방법이 사용된다. 그러나, 이러한 Dead-reckoning 방법은 로봇바퀴의 미끄러짐과 같은 물리적 현상이나 기구학적 계산의 오차에 의한 프로그램적 오차의 누적과 같은 많은 문제점을 야기할 수 있다. 이와 같은 단점을 보완하기 위해서 현재의 많은 연구에서는 기본적인 엔코더 센서와 더불어 다양한 센서들과의 융합을 통한 연구가 활발히 이루어지고 있다[5].

이처럼 엔코더 센서에 의한 Dead-reckoning 방법을 보완하기 위해 사용되는 센서들은 주로 자이로, 비전, 초음파, 레이저 센서 등이 사용되고 있고 특히, 최근에는 광역 센서로 GPS나 LBL과 같은 시스템도 자주 사용되고 있다. 본 논문에서는 이동 로봇의 자세 제어를 위해 위에서 언급했던 다양한 센서들의 사용을 배제하였고 오직 힘 센서만을 사용하여 이동로봇을 제어해 보고자 한다. 힘 제어라 함은 위치와 힘을 동시에 제어하는 방식을 말하는데 크게 두 가지 형태, 임피던스와 하이브리드, 가 있다[6][7]. 본 논문에서 제안하는 힘 제어는 기존의 로봇제어에서의 힘 제어 방식과는 달리[8-11], 바퀴 구동 형태의 이동로봇에 힘 센서를 장착한 후, 일정한 힘의 크기와 방향을 유지하도록 제어하

면서 임의의 경로를 추종하도록 한 것이다. 이는 마치 장남이 손만을 사용하여 눈으로 볼 수 없는 벽을 더듬으면서 길을 찾는 것처럼 오직 힘 센서만을 이용하여 알려지지 않은 경로를 추종하도록 하는 것과 같다. 그와 동시에 로봇은 설정한 힘을 조절하며 움직이도록 하였다.

이를 위해 우선, 제안된 이동로봇에 대한 기구학 및 동력학적 분석에 기초한 힘 제어 알고리즘을 구성하여 시뮬레이션을 통해 다양한 환경에서 제안된 제어 알고리즘을 검증해 보았다. 모양이 직선이거나 원과 같은 환경에서 대상의 강성도를 달리하며 시뮬레이션함으로써 제안한 힘 추종제어 알고리즘을 검증해 보았다. 또한 시뮬레이션 결과를 검증하기 위해 다양한 형태로 설계된 경로에 대한 실제 실험을 통해 시뮬레이션 결과와의 차이점을 비교해보고 힘 센서를 이용한 이동로봇의 제어에 대한 효율성을 검증해 보고자 하였다.

이와 같은 취지의 본 논문의 구성은 아래와 같다. 우선, 서론에 이어 2장에서는 제안된 이동로봇의 기구학에 대해 기술하였고 3장에서는 본 이동로봇의 동력학을 유도해 보았다. 4장에서는 실제 실험에 앞서 시뮬레이션을 통해 제안된 알고리즘의 유효성을 검증해보았고 5장에서는 실제 실험을 위해 제안된 힘 센서가 장착된 이동로봇의 전체적인 구조 및 내부 제어 구조, 그리고 실험을 위해 제작된 경로 환경에 대해 언급하였다. 6장에서는 실험 결과에 대해 상세하게 분석하였고 7장의 결론에서는 본 시스템 제어의 효율성에 대한 입장과 앞으로의 발전 방향에 대해 제시하였다.

#### II. 장남 로봇의 기구학

그림 1은 본 논문에서 제안된 장남 이동로봇의 기구학을 나타낸 것이다.

장애물과 Force 센서와의 접촉점,  $P_f$ 에서의 속도 벡터

\* 책임저자(Corresponding Author)

논문접수 : 2003. 1. 23., 채택확정 : 2003. 7. 10.

전풍우, 정 슬: 충남대학교 메카트로닉스공학과  
(windrainer@hanmail.net/jungs@cnu.ac.kr)

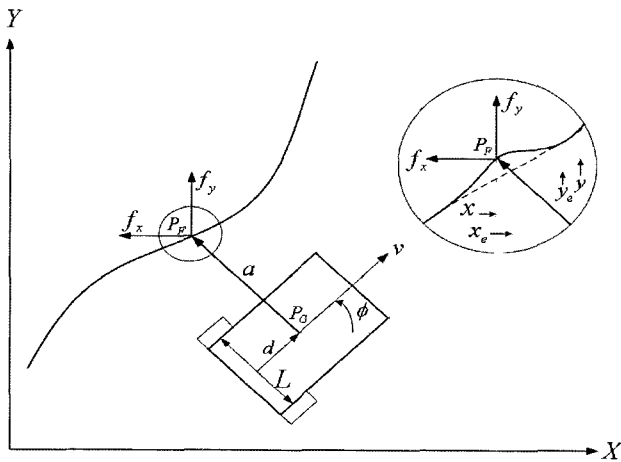


그림 1. 이동로봇의 기구학.  
Fig. 1. Kinematics of Mobile Robot.

들은 다음과 같이 로봇의 선속도,  $v$  와 각속도,  $\omega$ 로 나타내어질 수 있다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & -d \cdot \sin \phi & -a \cdot \cos \phi \\ \sin \phi & d \cdot \cos \phi & -a \cdot \sin \phi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

단, 여기서  $\phi$  는 절대좌표에 대한 로봇의 방향각,  $d$  는 구동축의 중심에서 로봇의 무게 중심,  $P_G$  까지의 거리이고  $a$ 는 로봇의 무게중심에서 Force 센서까지의 거리이다. 이때, 로봇의 선속도와 각속도는 다음과 같이 로봇 양쪽 바퀴의 회전 속도로 나타낼 수 있다.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \quad (2)$$

단,  $r$  은 바퀴의 반지름이고  $L$  은 바퀴 양축간의 거리이다.

(1),(2)로부터 로봇 양쪽 바퀴의 회전속도와 로봇의 속도 벡터들과의 관계를 다음과 같이 나타낼 수 있다.

$$\dot{X} = J \dot{q} \quad (3)$$

단,  $\dot{X} = [\dot{x}, \dot{y}, \dot{\phi}]^T$ ,  $J = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \\ j_{31} & j_{32} \end{bmatrix}$ ,  $\dot{q} = [\dot{\theta}_R, \dot{\theta}_L]^T$

이고, 자코비안 행렬,  $J$ 의 각 원소는 다음과 같다.

$$\begin{aligned} j_{11} &= \frac{r}{2} \cos \phi - \frac{r}{L}(d \cdot \sin \phi + a \cdot \cos \phi), \\ j_{12} &= \frac{r}{2} \cos \phi + \frac{r}{L}(d \cdot \sin \phi + a \cdot \cos \phi) \\ j_{21} &= \frac{r}{2} \sin \phi + \frac{r}{L}(d \cdot \cos \phi - a \cdot \sin \phi), \\ j_{22} &= \frac{r}{2} \sin \phi - \frac{r}{L}(d \cdot \cos \phi - a \cdot \sin \phi) \\ j_{31} &= \frac{r}{L}, \quad j_{32} = -\frac{r}{L} \end{aligned}$$

바퀴구동형태의 이동로봇은 2개의 제어입력( $\dot{\theta}_R, \dot{\theta}_L$ )에 대하여 3개의 출력상태( $\dot{x}, \dot{y}, \dot{\phi}$ )를 갖는 non-holonomic 시스템이므로 위에서 보는바와 같이 자코비안 행렬이 비정방 행렬의 형태이다.

### III. 이동 로봇의 동력학

일반적으로 이동로봇은 평면상에서 움직이므로 위치에너지는 무시되고 운동에너지에 의해 동적 특성이 결정된다. 이러한 일반적인 이동로봇의 동력학은 라그랑지안 방식을 사용하여 유도해보면 다음과 같은 형태로 나타낼 수 있다 [12][13].

$$M(q) \ddot{q} + V(q, \dot{q}) = P(q) \tau - A^T \lambda \quad (4)$$

단,

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

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

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

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

여기서  $m_c$  는 바퀴의 질량을 제외한 이동로봇의 질량이고  $m$  은 바퀴를 포함한 이동로봇의 전체 질량,  $I$  는  $\dot{\phi}$  에 관련된 모든 회전관성,  $I_W$  는 양쪽 바퀴의 회전관성이다.

그리고 (4)의  $M(q)$ 는 관성행렬,  $V(q, \dot{q})$ 는 원심력 및 코리올리스 벡터,  $P(q)$ 는 입력 변환 행렬,  $\tau$ 는 입력 토크 벡터,  $A(q)$ 는 자코비안 행렬,  $\lambda$ 는 라그랑지안 multiplier를 나타낸다.

시스템 분석을 간단히 하기 위해 위의 동력학식으로부터 라그랑지안 multiplier를 소개한다[12].

우선, 이동로봇의 속도 벡터는 다음과 같이 나타낸다.

$$\dot{q} = S(q) \dot{\nu}(t) \quad (5)$$

단,  $\dot{q} = [\dot{x}, \dot{y}, \dot{\phi}, \dot{\theta}_R, \dot{\theta}_L]^T$ ,  $\dot{\nu}(t) = [\dot{\theta}_R, \dot{\theta}_L]^T$

(5)를 (4)에 대입하면 다음과 같다.

$$M(S \dot{\nu}(t) + S \ddot{\nu}(t)) + V(q, \dot{q}) = P \tau - A^T \lambda \quad (6)$$

라그랑지안 multiplier를 소개하기 위해  $S^T A^T \lambda = 0$ 을 만족하도록 하는  $S^T$ 를 (6)의 양변에 곱하여 정리하면 다음

과 같다.

$$S^T[M(\ddot{S}v(t) + S\ddot{v}(t)) + V(q, \dot{q})] = S^T P\tau \quad (7)$$

$S^T P\tau = \tau$  이므로 (7)을 다시 정리하면 다음과 같다.

$$S^T M S \ddot{v}(t) + S^T M \dot{S} \dot{v}(t) + S^T V(q, \dot{q}) = \tau \quad (8)$$

이때,  $S^T A^T \lambda = 0$ 을 만족하도록 하는  $S$  는 다음과 같이 주어진다.

$$S(q) = \begin{pmatrix} \left( \begin{array}{cc} \left( \frac{r}{2} \cos \phi - \frac{rd}{L} \sin \phi \right) & \left( \frac{r}{2} \cos \phi + \frac{rd}{L} \sin \phi \right) \\ \left( \frac{r}{2} \sin \phi + \frac{rd}{L} \cos \phi \right) & \left( \frac{r}{2} \sin \phi + \frac{rd}{L} \cos \phi \right) \end{array} \right) & \\ \frac{r}{L} & -\frac{r}{L} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (9)$$

이때,  $b \equiv \frac{L}{2}$ ,  $c \equiv \frac{r}{L}$  라 정의하고 (8)에 (9)를 대입하여 정리하면 일반적인 이동로봇의 동력학은 다음과 같이 나타낼 수 있다[13].

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} + \begin{bmatrix} 2c^3bd(m+m_c)(\dot{\theta}_R \dot{\theta}_L - \dot{\theta}_L^2) \\ 2c^3bd(m+m_c)(\dot{\theta}_R \dot{\theta}_L - \dot{\theta}_R^2) \end{bmatrix} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (10)$$

단,

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

그런데, 본 논문에서 적용된 이동로봇의 경우, 2차원 평면상에서 매우 느린 속도로 움직이므로 (10)의 원심력 및 코리올리스 성분이 무시할 수 있을 정도로 매우 작다고 가정하면 (10)은 다음과 같이 더욱 간략화된 관성 시스템이 된다.

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (11)$$

즉, 2차원 평면상을 비교적 천천히 움직이는 이동로봇은 (11)과 같이 간단한 inertial 시스템으로 간주될 수 있다. 이러한 이유로 이동로봇의 제어에서는 동력학적 요소보다 기구학적 요소를 기준으로 제어하는 경우가 일반적이다 [12][13].

#### IV. 시물레이션

##### 1. 제어 법칙

위에서 살펴본 이동로봇의 기구학 및 동력학적 분석으로부터 시물레이션을 위한 PD제어 형태의 로봇의 구동토크에 관한 관계식을 유도해보도록 하겠다.

우선, 마찰이 없는 2차원 평면상에서 비교적 천천히 움직이는 시스템의 동력학식은 다음과 같이 나타낼 수 있다. 제어법칙을 유도하기 위해 다시 쓰면 다음과 같다.

$$D(q)\ddot{q} = \tau - J^T(q)h \quad (12)$$

$$\tau = D(q)\ddot{q} + J^T(q)h \quad (13)$$

단,  $h = \begin{bmatrix} f \\ \mu \end{bmatrix}$  이며,  $f$  는 로봇에 가해지는 외부 힘이고

$\mu$  는 로봇에 가해지는 외부 모멘트이다.

일반적인 로봇 시스템의 경우, 작업공간으로부터 조인트 공간으로의 변환관계는 다음과 같이 나타내어진다.

$$\dot{q} = J^{-1} \dot{X} \quad (14)$$

이를 한번 미분하면 다음과 같다.

$$\ddot{q} = J^{-1}(\ddot{X} - \dot{J}J^{-1}\dot{X}) \quad (15)$$

그러나, 이동로봇의 경우, (3)에서 보는바와 같이 자코비안 행렬,  $J$  가 정방행렬이 아니므로 (15)처럼 역행렬을 바로 구할 수 없다. 따라서, 다음과 같이 Pseudo inverse 자코비안을 구하여 사용하고자 한다.

$$\dot{q} = J^{*-1} \dot{X} \quad (16)$$

이를 한번 미분하면 다음과 같다.

$$\begin{aligned} \ddot{q} &= J^{*-1}(\ddot{X} - \dot{J}J^{*-1}\dot{X}) \\ &= (J^T J)^{-1} J^T(\ddot{X} - \dot{J}(J^T J)^{-1} J^T \dot{X}) \end{aligned} \quad (17)$$

단, 여기서  $\dot{X} = [\dot{x} \ \dot{y} \ \dot{\phi}]^T$ ,  $\dot{q} = [\dot{\theta}_R \ \dot{\theta}_L]^T$ ,  $J^{*-1} \equiv (J^T J)^{-1} J^T$  이다.

실제 시스템에서 측정 가능한 출력변수,  $Y$ 를 다음과 같이 정의하자.

$$Y = [f_x \ f_y \ \phi]^T \quad (18)$$

그럼 1에서 보여진 것 처럼 물체와 이동로봇의 접촉에 의해 발생하는 2차원적인 힘( $f_x, f_y$ )을 다음과 같이 모델링할 수 있다고 가정하자.

$$\begin{aligned} f_x &= k_e \delta x \\ f_y &= k_e \delta y \end{aligned} \quad (19)$$

단,  $\delta x = x - x_e$ ,  $\delta y = y - y_e$  이고,  $x_e, y_e$  는 이동로봇과 장애물 사이의 접촉이 일어나기 직전의 장애물 상의 접촉점의 절대 좌표를 나타낸 것이며  $x, y$ 는 접촉에 의해 변형이 발생한 장애물의 접촉점의 위치를 나타낸다. 그리고,  $k_e$ 는 접촉되는 물체의 강성도를 나타낸다. 실제 실험의 경우에서, 로봇과 힘 센서가 장착되어 장애물과 접촉되는 부분이 장애물에 비해 비교적 강성도가 강한 재료인 알루미늄 합금과 스틸로 만들어진 반면, 외부 장애물의 경우 강성도가 비교적 작은 나무판을 사용하였다. 이에 따라 시물레이션 상에서도 외부 장애물 측의 강성도만을 고려하였다.

(19)에 의해 (18)의  $Y$  벡터는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 Y &= K_e \Delta X \\
 &= K_e (X - X_e) \\
 &= \begin{bmatrix} k_e & 0 & 0 \\ 0 & k_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_e \\ y - y_e \\ \phi \end{bmatrix}
 \end{aligned} \tag{20}$$

그런데,  $\dot{X}_e = \ddot{X}_e = 0$  이므로 (20)을  $\dot{X}$ 와  $\ddot{X}$ 에 대해 정리하면 다음과 같다.

$$\dot{X} = K_e^{-1} \dot{Y}, \tag{21}$$

(17)에 (21)을 대입하여 정리하면 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 \ddot{q} &= J^{*-1} (\ddot{X} - \dot{J} J^{*-1} \dot{X}) \\
 &= J^{*-1} K_e^{-1} (\ddot{Y} - \dot{J} J^{*-1} \dot{Y})
 \end{aligned} \tag{22}$$

(22)를 (13)에 대입하면 이동로봇의 제어 법칙은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 \tau &= D(q) \ddot{q} + J^T(q) \dot{h} \\
 &= D [J^{*-1} K_e^{-1} (\ddot{Y} - \dot{J} J^{*-1} \dot{Y})] + J^T \dot{h}
 \end{aligned} \tag{23}$$

다음 단계로서, PD제어 형태의 다음과 같은 이상적인 오차 방정식을 설정해보자.

$$\ddot{e} + K_D \dot{e} + K_P e = 0 \tag{24}$$

단,  $e = Y_d - Y$ 이다.

(24)를 만족하는 제어입력을 설정하기 위해  $\dot{Y}$ 에 대해 정리하면 제어 입력은 다음과 같다.

$$\dot{Y} = \dot{Y}_d + K_D (\dot{Y}_d - \dot{Y}) + K_P (Y_d - Y) \tag{25}$$

(25)를 (23)에 대입하여 정리하면 본 논문에서 제시된 이동로봇의 제어법칙은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 \tau &= D J^{*-1} K_e^{-1} [\dot{Y}_d + K_D (\dot{Y}_d - \dot{Y}) - \dot{J} J^{*-1} \dot{Y} \\
 &\quad + K_P (Y_d - Y)] + J^T \dot{h}
 \end{aligned} \tag{26}$$

단,  $D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$ ,  $K_e = \begin{bmatrix} k_e & 0 & 0 \\ 0 & k_e & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ,

$K_D = \begin{bmatrix} k_{DX} & 0 & 0 \\ 0 & k_{DY} & 0 \\ 0 & 0 & k_{D\phi} \end{bmatrix}$ ,  $K_P = \begin{bmatrix} k_{PX} & 0 & 0 \\ 0 & k_{PY} & 0 \\ 0 & 0 & k_{P\phi} \end{bmatrix}$ 이다

다음으로 (26)의 PD제어 형태의 제어기를 사용하여 이동로봇의 힘 제어 성능에 대해 시뮬레이션을 통해 알아본다. 그림 2는 본 논문에서 제시된 로봇 시스템에 관한 시뮬레이션 분석을 위한 제어 구조를 나타낸 것이다.

시뮬레이션의 편의를 위해 주어진 전체 경로에 대해 매 샘플링 시간마다의  $Y_d$ 의 값, 즉,  $(f_x, f_y, \phi)$ 의 목표 값을 스텝 파형으로 형성하였다. 즉,  $\dot{Y}_d = \ddot{Y}_d = 0$ 이 되도록 목표 값을 설정하여 시뮬레이션 하였다.

2. 시뮬레이션 환경

로봇 동력학은 앞 절의 동력학을 사용하였고, 제어입력

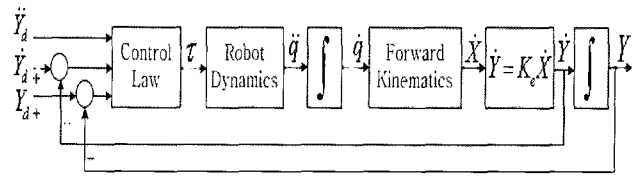


그림 2. 제어 구조.

Fig. 2. Control Structure.

은 (26)의 PD형태의 제어기를 사용하였다. 상용프로그램으로 MATLAB 6.0을 이용하여 시뮬레이션 하였다.

표1은 시뮬레이션을 위한 실제 이동로봇의 기구학적 파라미터 및 각종 제어 파라미터들을 나타낸 것이다.

표 1. 시뮬레이션 파라미터.

Table 1. Simulation Parameters.

기구학적 파라미터	설정값	제어 파라미터	설정값
$L$	20cm	$D_{11}, D_{22}$	1
$a$	15cm	$D_{12}, D_{21}$	0
$d$	4cm	$k_{PX}, k_{PY}$	20,000
$r$	4cm	$k_{DX}, k_{DY}$	1,500
Sampling time	1ms	$k_{P\phi}, k_{D\phi}$	0
		$k_e$	10,000

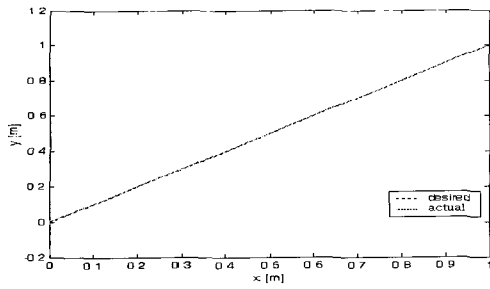
제어 파라미터의 경우, 다양한 시뮬레이션 환경에 따른 PD 제어기의 이득값에 약간의 변화만 있었을 뿐, 이외의 파라미터들은 모든 경우에 동일한 값을 적용하였다. 제어 파라미터 중 PD 제어기의 이득값이 매우 크게 설정되어 있는데, 이는 (26)에서 알 수 있듯이 경로에 대한 강성도,  $k_e$ 에 의해 제어 토크의 값이 나뉘어지는 것을 보상하기 위함이다.

또한, 본 논문에서 제시한 바퀴구동 형태의 이동로봇은 2개의 제어입력( $\theta_R, \theta_L$ )에 대하여 3개의 출력상태( $\dot{x}, \dot{y}, \dot{\phi}$ )를 갖는 Non-holonomic 시스템이므로, 3개의 출력상태 중 2개의 상태,  $\dot{x}, \dot{y}$ 만을 제어하였고 그에 따라 나머지 하나의 출력상태,  $\dot{\phi}$ 까지 제어되도록 하였다.

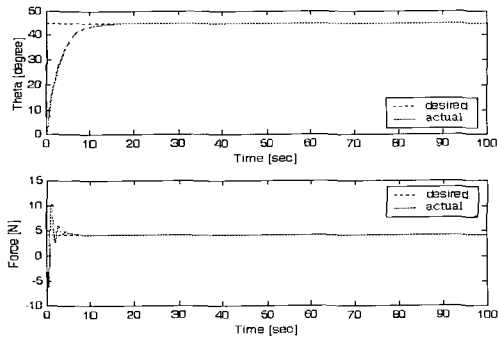
3. 시뮬레이션 결과

3.1 다양한 경로에 따른 시뮬레이션 결과

실제로 구현 가능한 비교적 다양한 경로를 설정한 후, 동일한 시뮬레이션 파라미터를 적용하여 로봇의 위치 및 방향각 그리고 반발력 등을 제어해 보았다. 그림 3, 4, 5, 6, 7의 (a)는 각각 직선, 원형, 곡선, 계단형, 사각형 경로에 대한 위치 제어의 결과로서, 그림에서 보는 바와 같이 비교적 잘 제어되었음을 확인할 수 있다. 위치 제어의 경우, 목표 경로는 로봇과 접촉하는 가상적인 실제 경로로부터 일정한 반발력이 발생되도록 계산되어진 경로로서 가상의 실제 경로로부터 일정한 거리만큼 떨어져 물체 내부에 형성된 경로를 나타낸다.



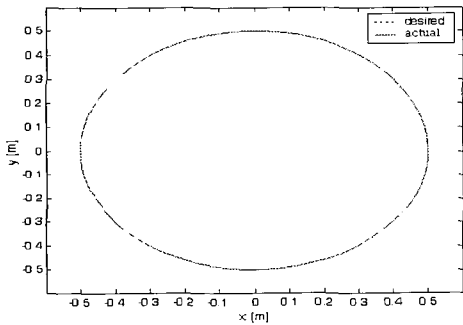
(a) 위치 추종



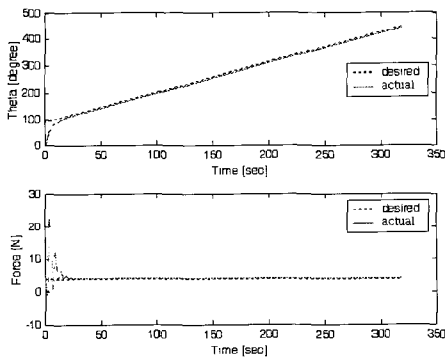
(b) 힘과 방향각 추종

그림 3. 직선 경로.

Fig. 3. Straight Line Tracking.



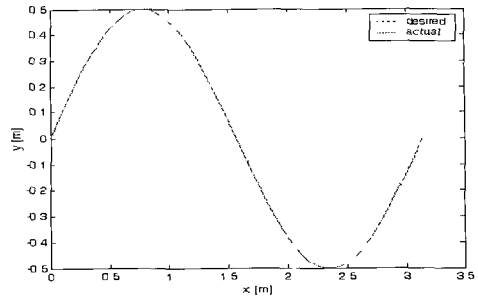
(a) 위치 추종



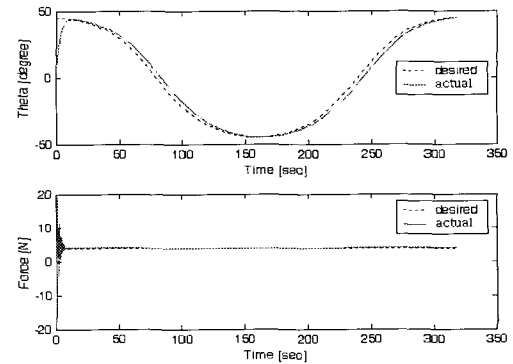
(b) 힘과 방향각 추종

그림 4. 원형 경로.

Fig. 4. Circular Trajectory.



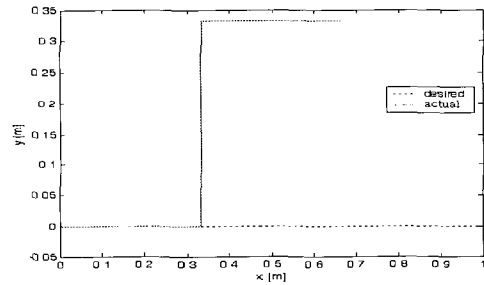
(a) 위치 추종



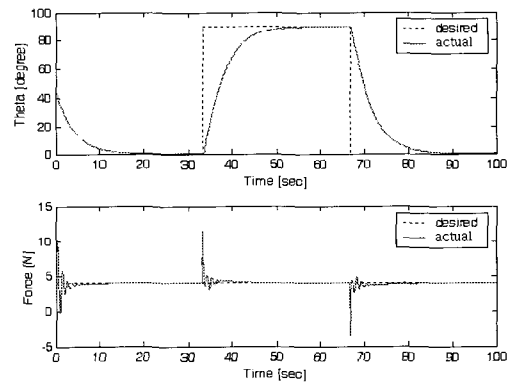
(b) 힘과 방향각 추종

그림 5. 곡선 경로.

Fig. 5. Curved Trajectory Tracking.



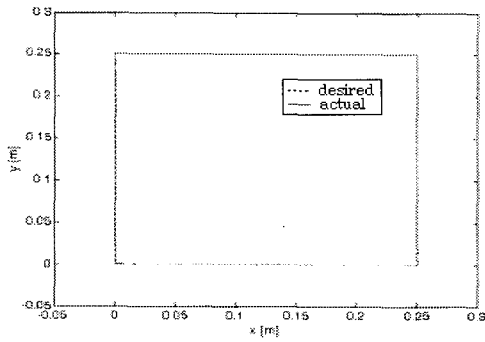
(a) 위치 추종



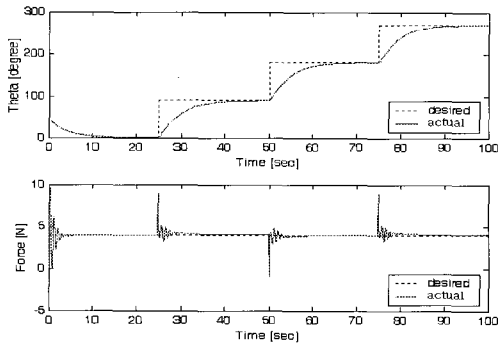
(b) 힘과 방향각 추종

그림 6. 계단형 경로.

Fig. 6. Step Trajectory.



(a) 위치 추종



(b) 힘과 방향각 추종

그림 7. 사각형 경로.

Fig. 7. Rectangular Trajectory.

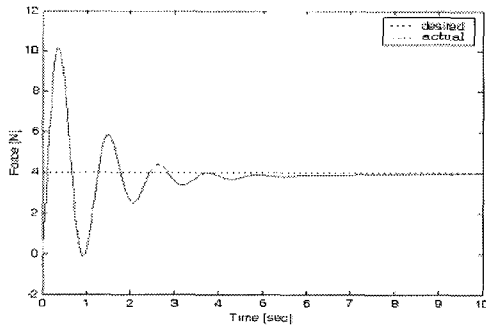


그림 8. 이동속도 : 1cm/s,  $k_e = 10,000$  N/m.

Fig. 8 Velocity : 1cm/s,  $k_e = 10,000$  N/m.

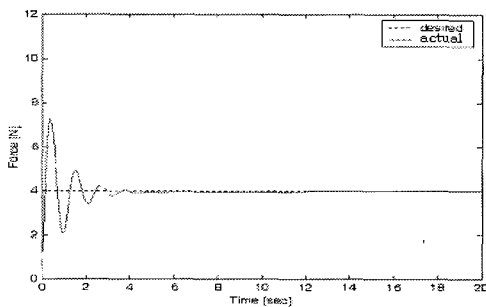


그림 9. 이동속도 : 0.5cm/s,  $k_e = 10,000$  N/m

Fig 9. Velocity : 0.5cm/s,  $k_e = 10,000$  N/m

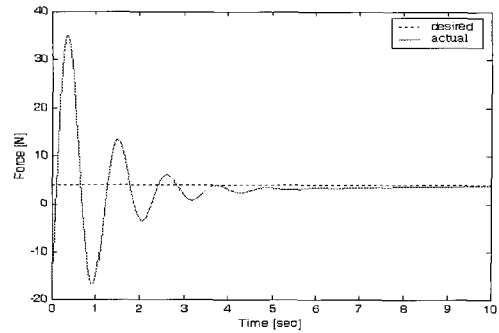


그림 10. 이동속도 : 1cm/s,  $k_e = 50,000$  N/m.

Fig. 10. Velocity : 1cm/s,  $k_e = 50,000$  N/m.

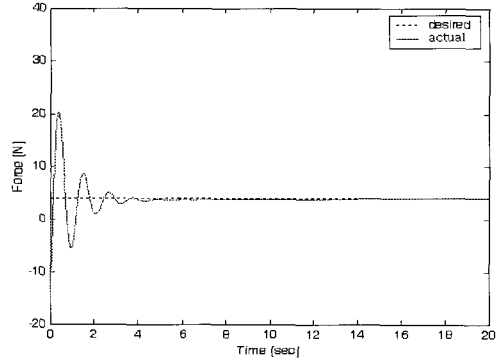


그림 11. 이동속도 : 1cm/s,  $k_e = 50,000$  N/m.

Fig. 11. Velocity : 1cm/s,  $k_e = 50,000$  N/m.

그림 3, 4, 5, 6, 7의 (b)는 각각의 경로에 대한 로봇의 방향각과 힘 제어에 대한 결과이다. 우선, 로봇 방향각은 설정된 경로와 로봇 기구학을 바탕으로 계산되어진 값, 즉, 로봇의 이동 방향이 경로의 접선방향과 평행하도록 하는 값을 목표 값으로 설정하였는데, 위에서 밝힌바와 같이 방향각의 제어는 실제 시뮬레이션 제어구조에서는 실행되지 않았고 두 방향의 힘 제어에 따라 내부적인 기구학에 의해 제어된 결과이다. 각각의 그림에서 확인할 수 있듯이 로봇 방향각의 제어는 대부분의 경로에서 잘 이루어 졌다. 다만, 그림 4의 (b)와 그림 5의 (b)의 경우, 목표 값과는 약간의 윽셋 값을 가지고 있는데, 이는 프로그램에서 삼각함수 등을 계산하는 과정에서 발생한 오차의 누적에 의한 결과로 사료된다.

힘 제어의 경우, 목표 값은 모든 경로에 대해 동일하게 설정하였고 경로에 수직을 이루는 방향으로 4N을 제어하도록 설정하였다. 이 목표 값은 실제로 제작한 이동로봇의 크기와 무게, 바퀴의 구동 토크 등을 고려하여 설정한 값이다. 그림에서 보는데와 같이 대부분의 경우에서 제어 초기에 많은 과도 응답을 보이지만 정상상태에서는 정상상태오차 없이 잘 제어됨을 확인할 수 있다. 다만, 계단형 경로나 사각형 경로의 경우, 직각을 이루는 모서리 부근에서는 과도한 응답이 확인되었으나, 곧 수렴함을 볼 수 있다.

3.2 파라미터의 변화에 따른 힘 제어 결과의 변화

이절에서는 동일한 직선 이동거리(10cm)를 설정한 후, 경로의 강성도와 로봇의 이동속도 등을 변화시켜가면서 힘 제어의 변화된 결과를 확인해 보았다.

그림 8은 경로의 강성도,  $k_e$ 를 비교적 작은 10,000N/m으로 설정하였고, 로봇의 이동속도를 약 1cm/s의 속도를 갖도록 설정하여 시뮬레이션한 결과로서 과도상태 초기에 약 10N정도의 오버슈트를 나타낸다. 그림 9는 경로의 강성도를 일정하게 유지하고 이동속도를 0.5cm/s로 설정하여 실험한 결과로서 과도응답이 약 7N정도로 그림 8에 비해 줄어든 것을 확인할 수 있다. 그림 10은 이동속도를 1cm/s로, 강성도를 50,000N/m으로 설정한 것으로 과도상태에서 30N이 넘는 오버슈트가 발생함을 확인할 수 있다. 그림 11은 이동속도를 0.5cm/s로, 강성도를 50,000N/m으로 설정하여 실험한 것으로 그림 10에 비해 적은 오버슈트를 나타내지만 그림 8에 비해서는 2배 정도의 오버슈트를 보인다. 하지만 모든 경우 추종 힘의 오차가 ‘영’으로 수렴하였다.

이와 같은 로봇의 이동속도와 경로의 강성도의 변화는 과도상태에서의 응답에는 많은 영향을 주었지만 정상상태에서는 그 차이를 알 수 없을 정도로 유사한 응답을 나타내었다. 이러한 오버슈트는 로봇과 환경과의 동적인 관계를 고려함이 없이 힘오차를 직접 줄이도록 제어하기 때문이다.

V. 실험 환경

그림 12는 본 논문에서 구현된 실제 실험 시스템이다. 그림 12에서 보는바와 같이 전체 시스템은 힘 센서를 장착한 이동로봇과 힘 센서로부터 데이터를 전송받기 위한 인터페이스 카드가 설치되어있는 컴퓨터로 구성되어있다. 이동로봇은 바퀴 구동용 로봇으로 컴퓨터로부터 구동명령을 전송받아 동작되는 형태이다.

그림 13은 본 실험에 사용된 힘 센서로서, 3축의 힘과 3축의 모멘트를 측정할 수 있도록 구성되어 있는데 실제 실험에서는  $f_x, f_y$ 의 두 방향의 힘에 대한 정보만을 이용하였다. 힘 센서와 인터페이스 카드는 RS-422 통신 규약에 의해 데이터를 주고받는다. 컴퓨터에서는 힘 센서로부터 데이터를 전송받아 제안된 제어 알고리즘을 수행한 후, 로봇에게 구동명령을 전송하게 된다. 힘 센서의 끝단에는 베어링

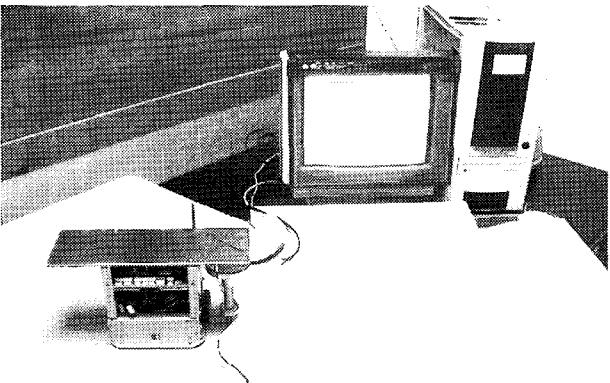


그림 12. 전체시스템 환경.  
Fig. 12. Overall Experimental Setups.

을 장착하여 경로와의 마찰을 최소화하고자 하였다. 로봇을 구동하기 위한 컨트롤러는 80C196KC를 사용하였고 컴퓨터와는 RS232 시리얼 통신을 이용하여 구동명령을 전송받게 된다.

이와 같은 제어구조는 그림 14에 보여진 것과 같다. 그림 15는 실제 실험에 사용된 실험 경로로서 그림에서 보는 바와 같이 다양한 형태의 경로로 구성되어 있고 재질은 나무로 제작되었다. 로봇은 주어진 경로를 알지 못하고 시작점에서 목표점까지 일정한 힘을 유지하며 주행하여야 한다.

- path 1 : 직선 경로
- path 2 : 모서리 경로
- path 3 : 오목 경로 ( curve radius > L )
- path 4 : 볼록 경로
- path 5 : 오목 경로 ( curve radius < L )

VI. 실험결과

위의 시뮬레이션의 경우는 프로그램에서 로봇의 위치와 방향을 설정하고 계산할 수 있는 반면, 실제 실험의 경우에는 로봇과 외부 환경의 절대 위치를 알 수 없다. 이에 따라 실제 제어 구조의 형태는 로봇의 절대 위치와는 무관한 임의의 경로를 추종하면서 일정한 힘의 크기와 방향각 만을 제어하도록 하는 구조이다. 이처럼 실제 실험과 시뮬레이션에서는 힘에 대한 에러 값을 설정하고 이에 대하여 PD형태의 제어를 설계하여 에러 값과 에러의 미분 값을 ‘영’으로 수렴토록 하는 제어 방식을 취하고 있다.

실제 실험은 그림 15와 같은 경로를 추종하면서 일정한 힘의 크기와 방향을 제어하도록 실험하였다[14]. 제어 목표 값은 힘의 크기의 경우, 경로와 수직인 방향으로 4N을 목표 값으로 설정하였고, 로봇의 방향각의 경우는 로봇의 이동 방향이 항상 경로의 접선방향과 평행하도록 설정하였다. 즉, 힘의 크기,  $F_{magnitude}$ 와 로봇의 방향각,  $F_{direction}$ 은



그림 13. 힘 센서의 외형과 6분력.  
Fig. 13. Force Sensor.

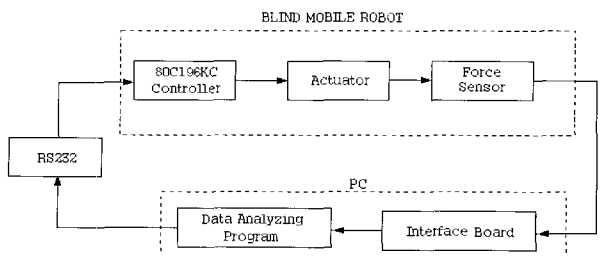


그림 14. 제어 구조 블록 다이어그램.  
Fig. 14. Control Block Diagram.

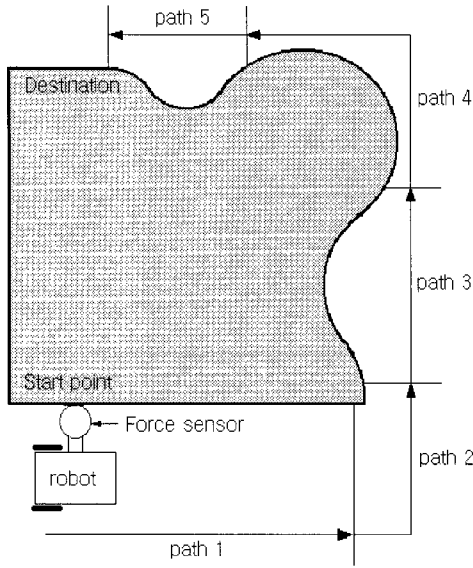


그림 15. 경로 환경.  
Fig. 15. Overall Trajectory.

다음과 같이 나타낸다.

$$F_{\text{magnitude}} = \frac{|F_{\text{normal}}|}{\sqrt{f_x^2 + f_y^2}} \quad (27)$$

$$F_{\text{direction}} = \tan^{-1}\left(\frac{f_y}{f_x}\right) \quad (28)$$

실제 실험 시에는 로봇의 방향각을 힘 센서로부터  $f_x, f_y$  정보를 얻어 (28)의 관계식을 이용하여 측정하였는데, 하드웨어 설계시에 로봇의 이동 방향이 경로의 접선방향과 평행한 자세가 될 때 방향각이  $45^\circ$ 가 되도록 힘 센서를 장착하였기 때문에 방향각의 목표 값은  $45^\circ$ 로 설정하였다.

그림 15에서 보는바와 같이 경로는 다양한 형상의 조합으로 구성되어 있는데, 우선 각각의 형상에 대해 독립적으로 실험한 후, 전체 경로에 대해 실험하였다. 그림 16은 직선 경로에 대한 실험 결과이다. 그림 16으로부터 로봇의 방향각의 경우 약  $4^\circ$  정도의 오프셋 에러가 발생함을 확인할 수 있는데, 이는 힘의 크기에 대한 목표 값을 만족시키기 위해 로봇이 물리적으로 경로를 미는 형상을 취하게 됨으로써 발생한 오차이다. 힘의 크기의 경우, 일정한 범위 내에서 목표 값을 유지함을 확인할 수 있다. 그림 17은 오목한 형상의 경로에 대한 실험결과로서 로봇의 방향각의 경우 그 오차가 평균적으로 '영'에 가까운 값을 나타내고, 힘의 크기 역시 비교적 잘 제어되었음을 확인할 수 있다. 그림 18은 불룩한 형상에 대한 실험결과로서 위의 두 실험과 마찬가지로 로봇의 방향각과 힘의 크기 모두 비교적 잘 제어되었다. 그림 19는 그림 15의 전체 경로에 대한 실험 결과를 나타낸 것이다. 그림에서 보는바와 같이 전체 주행 거리는 1m 정도이고 주행시간은 약 50초가 소요되며 약

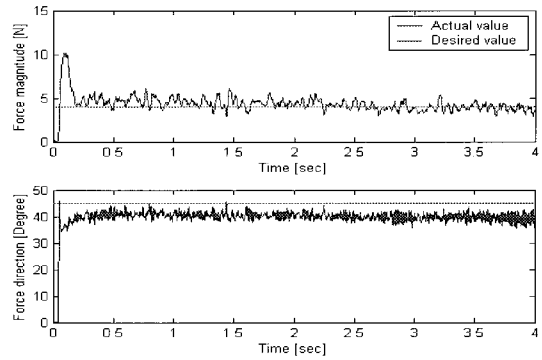


그림 16. 직선 경로의 힘 추종.  
Fig. 16. Force Tracking of Straight Line.

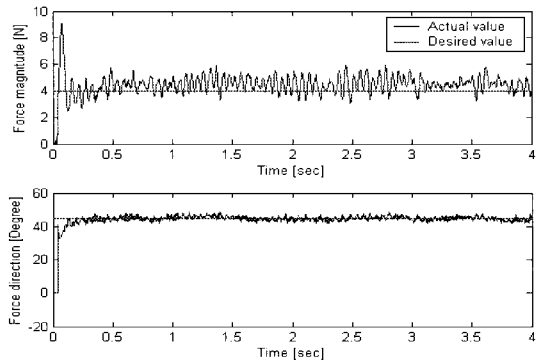


그림 17. 오목한 경로의 힘 추종.  
Fig. 17. Force Tracking of Concave Trajectory.

2cm/s의 속도로 움직였다. 이와 같은 로봇의 속도는 제어하고자 하는 힘의 목표 값이 비교적 작아서 로봇의 속도를 증가시키게 되면 과도상태에서 오버슈트가 증가하고 장애물의 표면 상태의 영향에 더욱 민감해지는 문제점과 함께, 로봇 자체의 구동 토크 등을 고려하여 설정한 결과로서, 이는 시뮬레이션상의 로봇의 이동속도에 따른 힘 제어 결과에서도 확인할 수 있다.

방향각의 오차에 관한 결과에서 약 8초부터 13초 정도까지 에러가  $45^\circ$ 가 됨을 확인할 수 있는데, 이는 로봇이 직선 경로를 경유하여 모서리 부근을 지나면서 경로로부터 떨어졌음을 나타낸 것이다. 이 부근의 힘의 크기에 관한 결과에서도 영으로 나타남을 확인해 볼 수 있다. 이외에 힘의 크기에 관한 결과에서 약 23초와 45초 정도에 비교적 큰 값의 과도응답이 나타남을 볼 수 있는데, 이는 로봇이 오목한 형상의 경로를 지날 때를 나타내는 것으로서 오목한 경로의 곡률 반경이 로봇 바퀴의 축간 거리보다 작게 되는 경우, 로봇의 오른쪽 바퀴가 역회전을 하게 될 때 발생하는 현상이다. 이러한 현상에서 알 수 있듯이, 알려지지 않은 다양한 형태로 구성된 경로에 대한 제어에서는 제어 이득에 대한 세심한 선택과 주의가 필요하게 된다.

**VII. 결론**

본 논문에서는 힘 센서를 장착한 바퀴 구동용 이동로봇의 힘 제어에 관해 기술하였다. 제안된 이동로봇의 기구학



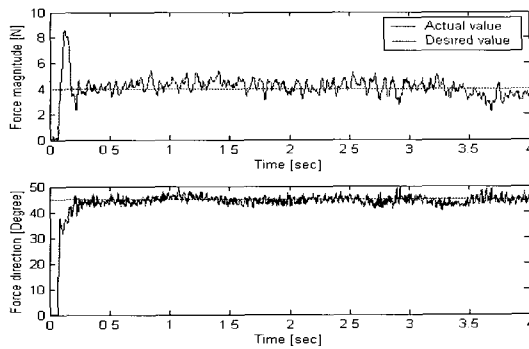


그림 18. 볼록한 경로의 힘 추종.

Fig. 18. Force Tracking of Convex Trajectory.

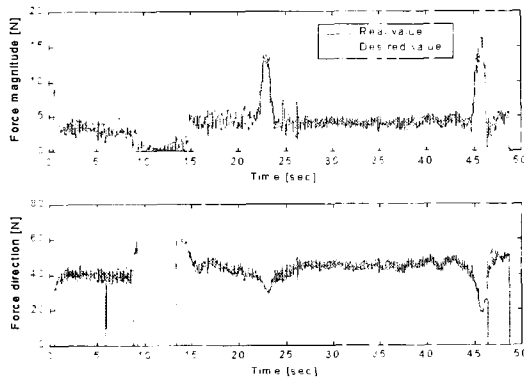


그림 19. 전체 경로의 힘추종.

Fig. 19. Force Tracking of Whole Trajectory.

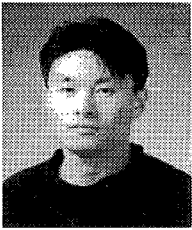
과 동력학식을 유도해보고 PD제어 형태의 제어기를 설계하여 이동로봇의 힘 제어에 관한 제안된 알고리즘을 시뮬레이션을 통해 검증해 보았다. 이를 바탕으로 힘 센서가 장착된 실제 이동로봇을 구현하여 알려지지 않은 경로에 대해 일정한 힘의 크기와 방향을 유지한 채 경로를 추종하도록 제어해 보았다. 실험 결과는 시뮬레이션 결과에 비해 적지 않은 오차가 발생됨을 관찰할 수 있었지만 유효범위 내에서는 힘의 크기와 방향 모두 비교적 잘 제어되었다.

추후 과제로는 다양한 경로에 대해 더욱 적절히 대응할 수 있는 제어 알고리즘의 개발과 바퀴 구동 토크의 부족 및 지면과의 미끄러짐과 같은 이동로봇의 하드웨어적 문제점의 해결함으로써 선박 용접용 로봇과 같은 실제 산업용 로봇으로의 적용 등이 있을 것이다.

#### 참고문헌

- [1] 특집(이동로봇 기술), 제어 자동화 시스템 공학회, pp. 717-797, 제 7 권, 제 9 호, 2001
- [2] 문종우, 김종수, 박세승, "차동 구륜 이동 로봇의 기구

- 학적 보정과 모터 제어기의 가속도 해상도 제약을 고려한 기준 속도 궤적의 설계", 제어 자동화 시스템 공학회, pp. 498-505, 제 8 권, 제 6 호, 2002
- [3] 김희국, 김도형, 이병주, "확장된 좌표계 전환기법에 의한 모바일 로봇의 기구학 모델링", 제어 자동화 시스템 공학회, pp. 233-242, 제 8 권, 제 3 호, 2002
- [4] 김희국, 이승은, 이병주, "평면형 모바일 로봇의 모빌리티에 대한 분석", 제어 자동화 시스템 공학회, pp. 243-248, 제 8 권, 제 3 호, 2002
- [5] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots," *IEEE Trans. on Robotics and Automations*, vol. 15, no. 2, pp. 219-229, April, 1999
- [6] N. Hogan, "Impedance control : An approach to manipulator, part i, ii, iii", *ASME Journal of Dynamics Systems, Measurements, and Control*, vol. 3, pp. 1-24, 1985
- [7] M. Raibert and J. Craig, "Hybrid position/force control of manipulators", *ASME Journal. of Dynamic Systems, Measurements, and Control*, vol. 102, pp. 126-133, 1981
- [8] H. Seraji, "Adaptive admittance control : an approach to explicit force control in compliant motion", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2705-2712, 1994.
- [9] S. Jung and T. C. Hsia, "Robust neural force control scheme under uncertainties in robot dynamics and unknown environment", *IEEE Trans. on Industrial Electronics*, vol. 47, no. 2, pp.403-412, April 2000.
- [10] 정슬, "로봇의 디버링 작업이나 표면 광택 작업을 위한 새로운 힘제어 기술 개발", *제어 자동화 시스템 공학 논문지*, pp.786-795, 제6권, 9호, 2000년
- [11] 정슬, 임선빈, "미정보 환경하에서의 신경회로망 힘 추종 로봇제어 기술의 실험적 연구", *제어 자동화 시스템 공학 논문지*, pp. 338-344, 제8권, 4호, 2002
- [12] N. Sarkar, X. Yun, and V.Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots," *International Journal of Robotics Research*, vol. 31, pp. 55-69, February, 1994.
- [13] 정슬, 장평수, 원문철, 홍섭, "신경회로망을 이용한 비전기반 이동로봇의 위치제어에 대한 연구", *제어 자동화 시스템 공학 논문지*, pp. 515-526, 제9권 7호, 2003
- [14] J.S. Jung, P. W. Jeon, "Force guided tracking control of blind mobile robot," *IASTED International conference : ROBOTICS AND APPLICATIONS*, pp. 148-151, November. 2001



**전 봉우**

1975년 12월 17일생. 2001년 충남대 메카트로닉스공학과 졸업. 2003년 동대학 석사졸업. 현재 동대학 박사과정. 관심분야는 로봇 설계 및 제작, 하드웨어 제작, 힘제어 알고리즘 응용.



**정 슬**

1964년 9월 11일 생. 1988년 미국 웨인주립대 전기 및 컴퓨터 공학과 졸업. 1991 미국 캘리포니아 대학 데이비스 석사, 1996년 동대학 박사(로봇 전공). 1997년~현재 충남대학교 메카트로닉스공학과 부교수. 관심분야는 지능 제어 및 감성 공학, 로봇 설계 및 제어, S.o.C 제어기 설계.