

LabVIEW를 이용한 이동로봇 위치제어를 위한 PI제어기 구현

論 文

57-7-19

An Implementation of PI Controller for the Position Control of Mobile Robot Using LabVIEW

朴永煥[†] · 李在璟^{*}
(Young-Hwan Park · Jae-Kyung Lee)

Abstract - The dynamics of mobile robot is nonlinear. To cope with this nonlinearity, many advanced control schemes have been proposed recently. Generally, the advanced control schemes are complicated and not good for the practical real-time control when they are implemented as control programs. So, in this paper, a relatively simple PI controller is proposed and applied to the position control of mobile robot with the adoption of reference trajectory calculation method used for the AUV(Autonomous Underwater Vehicle) control. The proposed PI controller is programmed using LabVIEW which is popular for its graphical programming characteristics. The simulation and experimental results show the feasibility and effectiveness of the proposed PI controller.

Key Words : Mobile Robot, PI control, Position Control, LabVIEW

1. 서 론

지금까지 이동로봇의 주행제어에 관한 여러 연구가 있었다[1-3]. 이동로봇의 동특성식은 기본적으로 비선형이다[4]. 따라서 이러한 비선형 동특성식을 다룰 수 있는 고급의 제어방식들이 여러 논문에서 제시되어 왔다. 예를 들어 백스텝핑(Backstepping)을 이용한 건설적응제어 방식[5], 모델예측 제어(Model Predictive Control) 방식[6], 입출력 변환 선형화(Input-Output Feedback Linearization) 방식[7] 등이 이동로봇 제어에 도입되었다. 그러나 이러한 고급 제어방식들은 일반적으로 알고리즘이 복잡하게 되며 실제 제어 프로그램으로 구현하였을 때 샘플링 시간이 길어지게 되어 실시간 제어에 적합하지 못한 단점을 갖고 있다.

이에 반해 PID 제어기는 구조가 단순하여 제어파라미터 값들만 잘 선정되면 실제 프로그램으로 구현하였을 때, 짧은 샘플링 시간으로 인하여 실용적으로 성능을 잘 발휘하는 경우가 많다. 이러한 관점에서 본 논문에서는 가급적 단순한 구조의 PI 제어기를 이동로봇 제어에 도입하더라도 기준계적의 추종 제어가 가능하고 짧은 샘플링 시간도 확보할 수

있음을 확인하고자 한다. 본 논문에서는 수중로봇(Autonomous Underwater Vehicle)에 적용되어[8] 그 효용성이 입증된 기준계적 유도방식을 이동로봇의 위치제어에 채택하였다. PI 제어기의 파라미터는 시행착오적으로 선정하였으며 시뮬레이션 결과, 제안된 PI 제어기가 이동로봇 동특성식의 비선형성에도 불구하고 만족할 만한 성능을 발휘함을 확인하였다. 제안한 PI 제어기는 LabVIEW[9]를 이용하여 프로그램하였다. LabVIEW는 그래픽 프로그램 언어(graphical programming language)라는 특성 때문에 최근 많이 이용되고 있는데 본 논문에서는 이동로봇 위치제어용 프로그램으로도 사용될 수 있음을 확인하였다. 실험결과로 얻은 로봇의 이동계적은 시뮬레이션에서 예상했던 궤적과 거의 같은 결과를 얻을 수 있었으며 이로부터 제안한 PI 제어기와 LabVIEW를 이용한 구현방식이 유용함을 확인할 수 있었다.

2. 이동로봇

2.1 이동로봇 동특성식 및 제어목적

이동로봇의 동특성식은 다음 그림 1로부터 정의된다[4]. 그림 1에서

- {I} : 전역적(global) 좌표계
- {R} : 국소적(local) 좌표계

[†] 교신저자, 正會員 : 忠州大 電氣電子 및 情報 工學部
정보제어전공 副教授 · 工博

^{*} 非會員 : 忠州大 電氣電子 및 情報 工學部 정보제어전공
教授 · 工博

接授日字 : 2008年 1月 18日

最終完了 : 2008年 4月 18日

- $\dot{\phi}_1$: 우측바퀴의 회전각속도[rad/s]
- $\dot{\phi}_2$: 좌측바퀴의 회전각속도[rad/s]
- l : 그림 1에서 로봇 폭의 1/2 [m]
- r : 그림 1에서 로봇 바퀴의 직경[m]

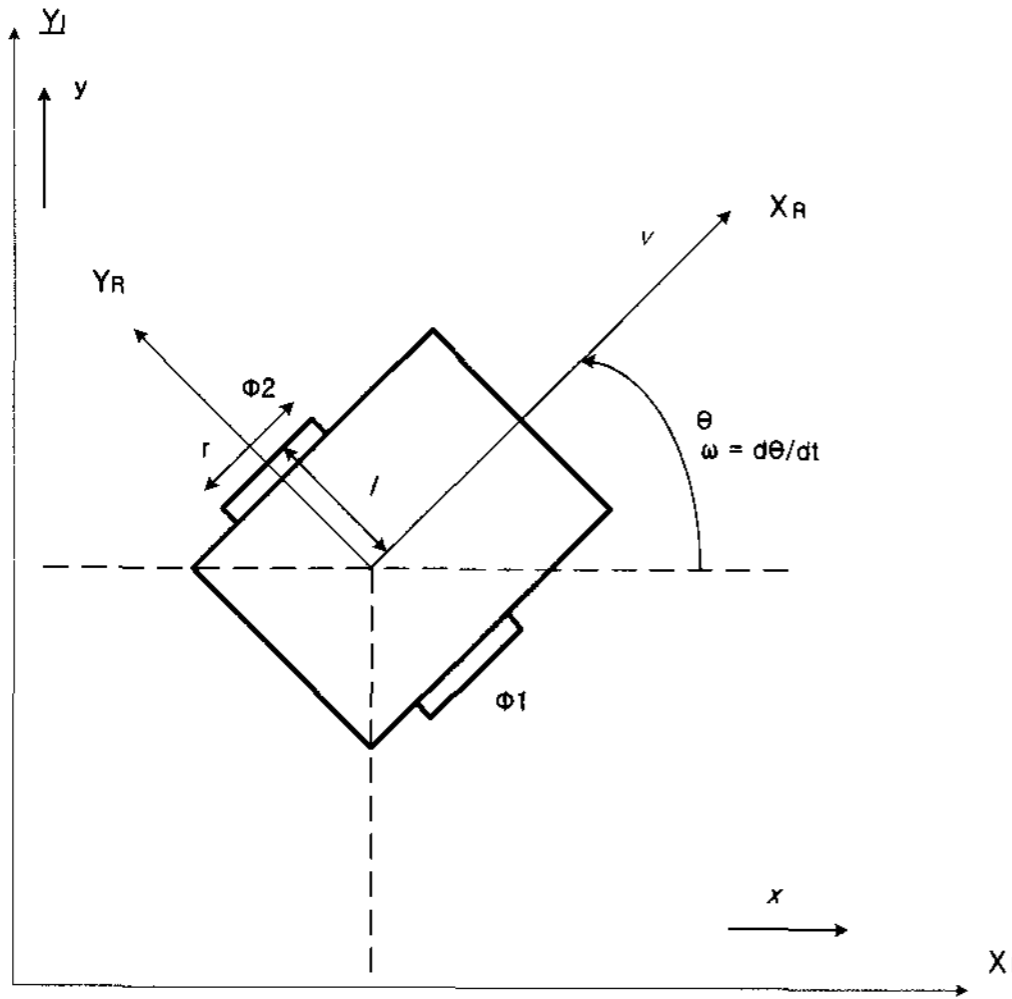


그림 1 이동로봇의 좌표계
Fig. 1 Coordinates for the Mobile Robot

그림 1에서 전역적 좌표계의 좌표축에 따른 직선 및 회전운동 성분은 다음 벡터로 나타낼 수 있다.

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

그리고 그림 1에서 국소 좌표계 {R}에서의 극좌표상의 운동 $\{v, \dot{\theta}\}$ 를 전역 좌표계 {I}에서의 운동으로 변환시켜 주는 변환행렬은 다음과 같다.

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (2)$$

즉,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta) \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (3)$$

또한 바퀴의 회전 각속도 $\dot{\phi}_1, \dot{\phi}_2$ 와 국소 좌표계 상의 운동 속도 $\{v, \dot{\theta}\}$ 사이에는 다음과 같은 관계가 있다[4].

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \quad (4)$$

(3), (4)로 부터 이동로봇의 운동모델(kinematic model)은 다음과 같다.

$$\dot{\xi}_I = \begin{bmatrix} \frac{r}{2} \cos\theta & \frac{r}{2} \cos\theta \\ \frac{r}{2} \sin\theta & \frac{r}{2} \sin\theta \\ \frac{r}{2l} & \frac{-r}{2l} \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \quad (5)$$

위의 식들은 이동로봇이 비선형 계통임을 보여준다. 본 논문의 제어목적은 (5)에서 상태변수 x, y 즉 로봇의 전역좌표계 상의 위치값이 원하는 값이 되도록 하는 것이다.

2.2. 제어기 설계 및 시뮬레이션

그림 2에는 이동로봇이 추종해야할 기준궤적이 나타나 있다.

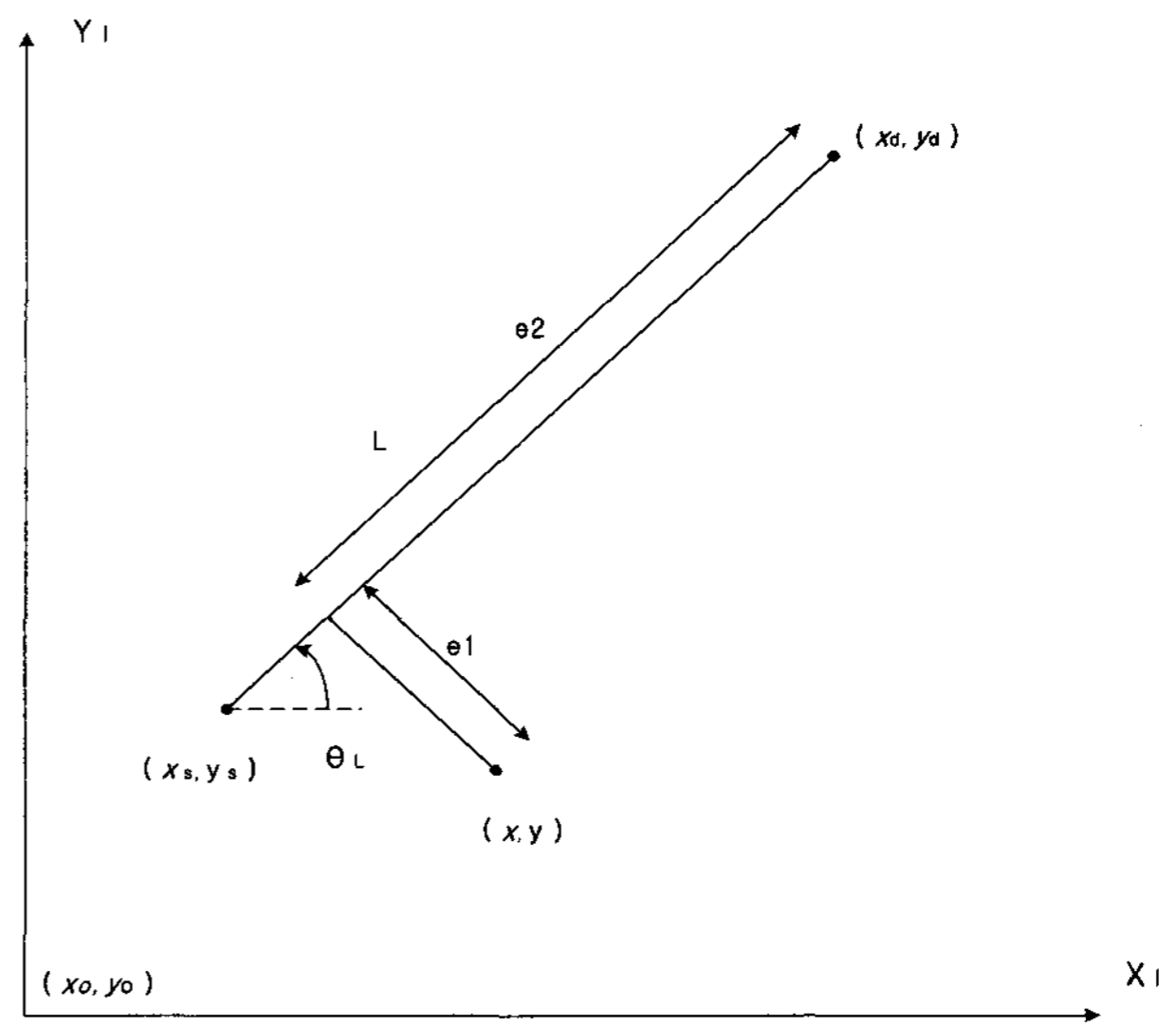


그림 2 이동로봇에 대한 기준궤적
Fig. 2 Reference Trajectory of Mobile Robot

시작점 (x_s, y_s) 로부터 목표점 (x_d, y_d) 로 향하는 직선 L이 기준궤적이며 이동로봇의 현위치 (x, y) 와 기준궤적과의 거리는 오차 e_1 으로, L상의 목표점으로 부터의 거리는 오차 e_2 로 정의 된다. 여기에서 기준궤적 L의 방향각은 다음과 같다.

$$\theta_L = \tan^{-1} \left(\frac{y_d - y_s}{x_d - x_s} \right) \quad (6)$$

e_1 은 GPS 수신좌표(동경, 북위좌표)로부터 실시간으로 얻은

로봇의 위치 (x, y) 를 이용하여 부록과 같이 계산된다. 그리고 이동로봇의 방향각에 대한 지령치를 θ_c 라 하면 $|e_1|$ 이 정해진 거리 d_1 이상 떨어져 있을 때에는 θ_c 가 빨리 θ_L 값에 가까이 가도록, d_1 보다 가까이 있을 때에는 $|e_1|$ 에 비례하도록 다음과 같이 선택적으로 θ_c 를 선정한다[8].

$$\theta_c(t) = \begin{cases} \theta_L + K \times e_1, & |e_1| \leq d_1 \\ \theta_L + \text{sgn}(e_1) \times 45^\circ, & |e_1| > d_1 \end{cases} \quad (7)$$

여기서 $K = \frac{45^\circ}{d_1}$ 이고 e_1 은 다음과 같은 부호를 갖는다.

$$\begin{cases} \text{sgn}(e_1) > 0, & \text{이동로봇이 기준궤적 } L \text{의 우측에 있을때} \\ \text{sgn}(e_1) < 0, & \text{이동로봇이 기준궤적 } L \text{의 좌측에 있을때} \end{cases}$$

그리고 운동방정식 (5)에서 이동로봇의 방향각 $\theta(t)$ 가 $\theta_c(t)$ 를 추종하도록 하는 바람직한 이동로봇 방향각 각속도 지령치 $\omega_d(t)$ 를 다음 PI 제어기 식으로부터 얻는다.

$$\omega_d(t) = K_p e_\theta(t) + K_I \int_0^t e_\theta(\tau) d\tau \quad (8)$$

여기서 $e_\theta(t) = \theta_c(t) - \theta(t)$ 이고 $\omega_d(t)$ 는 로봇 방향각 $\theta(t)$ 의 바람직한 각속도 값을 나타낸다. 또한 그림 1에서 이동로봇 국소 좌표계에서의 직선운동 속도 v 에 대한 지령치를 $V_d(t)$ 라 하면 $V_d(t)$ 는 목표점과의 오차 e_2 가 일정한 값 이상일 때는 정속도로 설정하고, 오차 e_2 가 정해진 값 이내일 때는 목표점에 근접한 것으로 간주하고 속도가 줄어들도록 $V_d(t)$ 를 e_2 에 비례하도록 설정한다. 또한 이동로봇이 목표점으로부터 정해진 오차거리 R 이내에 도달하게 되면 위치센서의 오차를 고려하여 목표점에 도달한 것으로 간주하고 $V_d(t)$ 값을 영으로 선정한다. 즉,

$$V_d(t) = \begin{cases} \text{상수 } V_{const}, & |e_2| > d_2 \\ K_p v_c \times e_2, & |e_2| < d_2 \\ 0, & |e_2| < R, \quad R < d_2 \end{cases} \quad (9)$$

이렇게 함으로써 Farrell의 논문[8]에서 나타나는 단점을 없앨 수 있다. 즉, Farrell의 논문에서는 오차 e_2 가 어느 값(R) 이내로 목표점에 도달했을 때 $V_d(t)$ 를 영으로 선정하는데, 이 경우에 가속도로 인해 오차 e_2 가 다시 R 이상으로 커져버릴 가능성이 생긴다. 그러나, 본 논문에서와 같이 $V_d(t)$ 를 e_2 에 비례시킬 경우 속도가 오차에 비례하며 천천히 줄어들어 그와 같은 현상이 없어진다.

그림 1에서 국소 좌표계의 직선운동 속도 v , 회전운동 각속도 $w(=\dot{\theta})$ 는 이동로봇 바퀴의 회전 각속도 $\dot{\phi}_1, \dot{\phi}_2$ 와 (4)의 관계를 가지므로 지령치 (8), (9)을 얻기 위한 이동로봇 바퀴 회전속도 지령치 $\dot{\phi}_{1d}, \dot{\phi}_{2d}$ 는 다음 식으로부터 얻을 수 있다.

$$\begin{bmatrix} \dot{\phi}_{1d} \\ \dot{\phi}_{2d} \end{bmatrix} = \frac{l}{r} \begin{bmatrix} \frac{1}{l} & 1 \\ \frac{1}{l} & -1 \end{bmatrix} \begin{bmatrix} V_d(t) \\ \omega_d(t) \end{bmatrix} \quad (10)$$

(10)의 $\dot{\phi}_{1d}(t), \dot{\phi}_{2d}(t)$ 는 이동로봇의 바퀴회전을 위한 지령치로 사용된다. 모터 드라이버와 모터 부분의 시정수가 이동로봇의 바퀴를 포함한 기구적인 부분의 시정수보다 상대적으로 매우 작고(즉, 응답속도가 매우 빠르고) 안정적이라 생각하면

$\dot{\phi}_1(t) \simeq \dot{\phi}_{1d}(t), \dot{\phi}_2(t) \simeq \dot{\phi}_{2d}(t)$ 이고 $\omega(t)(\equiv \dot{\theta}(t)) \simeq \omega_d(t), v(t) \simeq V_d(t)$ 라 할 수 있다. 이 경우 (8)에서 얻은 $\omega_d(t)$ 와 로봇의 방향각 회전 각속도 $\omega(t)$ 가 거의 같다고 할 수 있고 실제 로봇 방향각의 동특성식은 다음 (11)과 거의 같다고 할 수 있다.

$$\theta(t) = K_p(\theta_c(t) - \theta(t)) + K_I \int_0^t (\theta_c(\tau) - \theta(\tau)) d\tau \quad (11)$$

이 식은 선형계통에 대한 식이며 다음과 같은 블록 다이어그램으로 나타낼 수 있다.

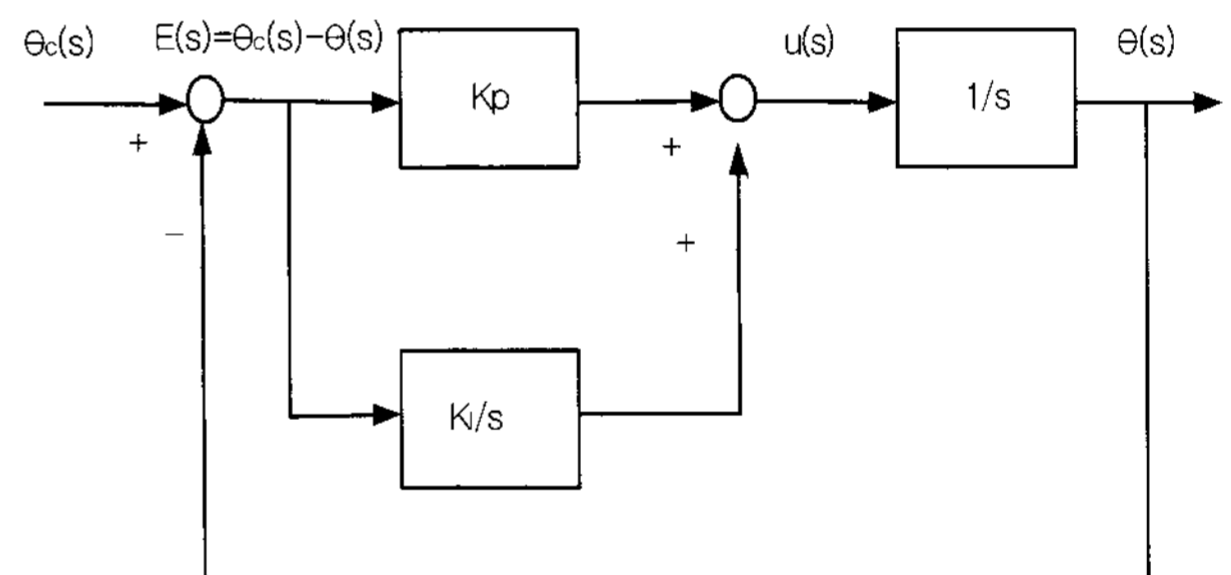


그림 3 식 (16)에 대한 블록 다이어그램
Fig. 3 Block Diagram for (16)

(11)의 우변을 PI제어기의 출력 $u(s)$ 라고 보면 그림 3의 블록 다이어그램에서 계통 전달함수 $G_p(s)$ 는 다음과 같고

$$G_p(s) = \frac{1}{s} \quad (12)$$

PI제어기 전달함수 $G_c(s)$ 와 궤환 경로의 전달함수 $H(s)$ 를 다음과 같이 정의하면

$$G_c(s) = K_p + \frac{K_I}{s}, \quad H(s) = 1 \quad (13)$$

$\theta_c(s)$ 와 $\theta(s)$ 사이의 폐루프 전달함수는 다음과 같다.

$$\frac{\theta(s)}{\theta_c(s)} = \frac{G(s)H(s)}{1+G(s)H(s)}, \quad G(s) = G_c(s)G_p(s) \quad (14)$$

$$\begin{aligned} \frac{\theta(s)}{\theta_c(s)} &= \frac{G(s)H(s)}{1+G(s)H(s)} \\ &= \frac{K_p s + K_I}{s^2} \\ &= \frac{K_p s + K_I}{1 + \frac{K_p s + K_I}{s^2}} \\ &= \frac{K_p s + K_I}{s^2 + K_p s + K_I} \end{aligned}$$

따라서 특성방정식은 다음과 같고

$$s^2 + K_p s + K_I = 0 \quad (15)$$

특성방정식의 근은 시뮬레이션과 실험에서 사용된 PI 제어기 파라미터 $K_p=1, K_I=2$ 에 대해 다음과 같이 구해진다.

$$\begin{aligned} s &= \frac{-K_p \pm \sqrt{K_p^2 - 4K_I}}{2} \\ &= -0.5 \pm j1.3229 \end{aligned} \quad (16)$$

이 값은 $\frac{\theta(s)}{\theta_c(s)}$ 폐루프 계통이 안정함을 말해준다. 이상의 과정을 정리하면, (7)에 의해 바람직한 방향각 $\theta_c(t)$ 가 얻어지고 이로부터 (8)을 이용하여 바람직한 방향각 각속도 $\omega_d(t)$ 가 얻어지며, (9)를 이용하여 바람직한 직선운동 속도 $V_d(t)$ 가 얻어진다. 이렇게 얻어진 $\omega_d(t), V_d(t)$ 는 (10)에서 바람직한 바퀴회전 속도 $\dot{\phi}_{1d}, \dot{\phi}_{2d}$ 를 구하는데 사용된다. 상대적으로 작은 시정수를 가진 모터드라이버와 모터 시스템에 의해 실제 바퀴회전속도 $\dot{\phi}_1(t), \dot{\phi}_2(t)$ 가 안정적으로 $\dot{\phi}_{1d}, \dot{\phi}_{2d}$ 에 급속히 수렴한다고 생각하면 다음 (17)과 같이 놓고 전체 계통을 해석할 수 있다.

$$\dot{\phi}_1(t) \approx \dot{\phi}_{1d}(t), \dot{\phi}_2(t) \approx \dot{\phi}_{2d}(t) \quad (17)$$

(17)은 (4)의 관계로부터 $v(t) \approx V_d(t), \theta(t) \approx \theta_c(t)$ 를 보장한다. 그리고 $\theta(t) \approx \theta_c(t)$ 는 $\omega_d(t)$ 를 (8)과 같이 설정하였으므로 (11)의 관계가 성립하도록 한다. 그리고 (11)~(16)은 $\theta(t)$ 가 $\theta_c(t)$ 를 안정적으로 추종함을 보여준다. 타당성 확인

을 위해 (5), (7), (8), (9), (10)을 이동로봇에 관련된 계수값들을 다음 표 1과 같이 설정하여 시뮬레이션한 결과 그림 4를 얻었다. 시뮬레이션에서 시작점과 목표점은 표 2와 같이 좌표를 선정하였다. 그림 4를 표 2와 비교해 보면 이동로봇이 주어진 목표점을 잘 추종하고 있음을 알 수 있다. GPS 데이터에 섞인 노이즈(noise)를 고려하여 R값을 10으로 선정하였고 따라서 그림 4에서 기준궤적의 끝점($x_d = 196.517$)으로 부터 몇[m] 전방에서 정지했음을 알 수 있다.

표 1 이동로봇 계수들

Table 1 Parameters of the Mobile Robot

계수	값
r	0.12 [m]
l	0.1 [m]
d1	30 [m]
d2	20 [m]
R	10 [m]
Vconst	0.2 [m/s]
Kp	1
Ki	2
KpVc	0.01

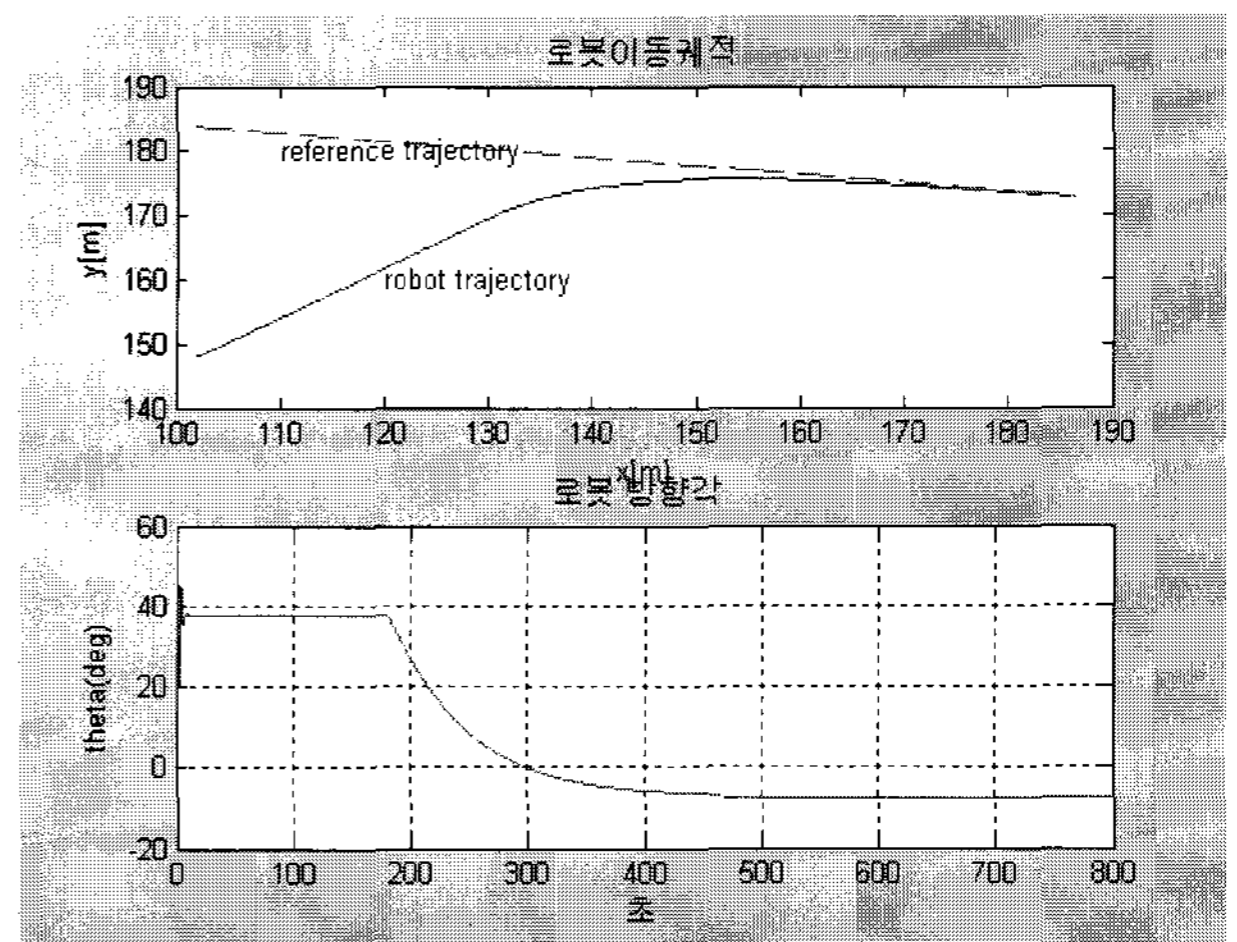


그림 4 로봇 이동궤적과 방향각(시뮬레이션)

Fig. 4 Robot Trajectory and Heading Angle(simulation)

표 2 기준궤적의 시작점, 끝점과 로봇의 출발점

Table 2 Start and End Point of the Reference Trajectory and the Robot Start Point

x_s	y_s [m]	x_d	y_d [m]	x	y [m]
99.8284	184.139	196.517	171.609	102.05	148

2.3. 실험결과

본 논문에서 제안한 제어알고리즘의 유용성을 확인하기 위해 그림 5와 같은 자율이동로봇을 제작하였다. 그림 6는 자율이동로봇의 개략적인 블록도이다. 그림 7은 본 논문에서 사용한 PI제어기 LabVIEW프로그램의 일부분을 보여주는

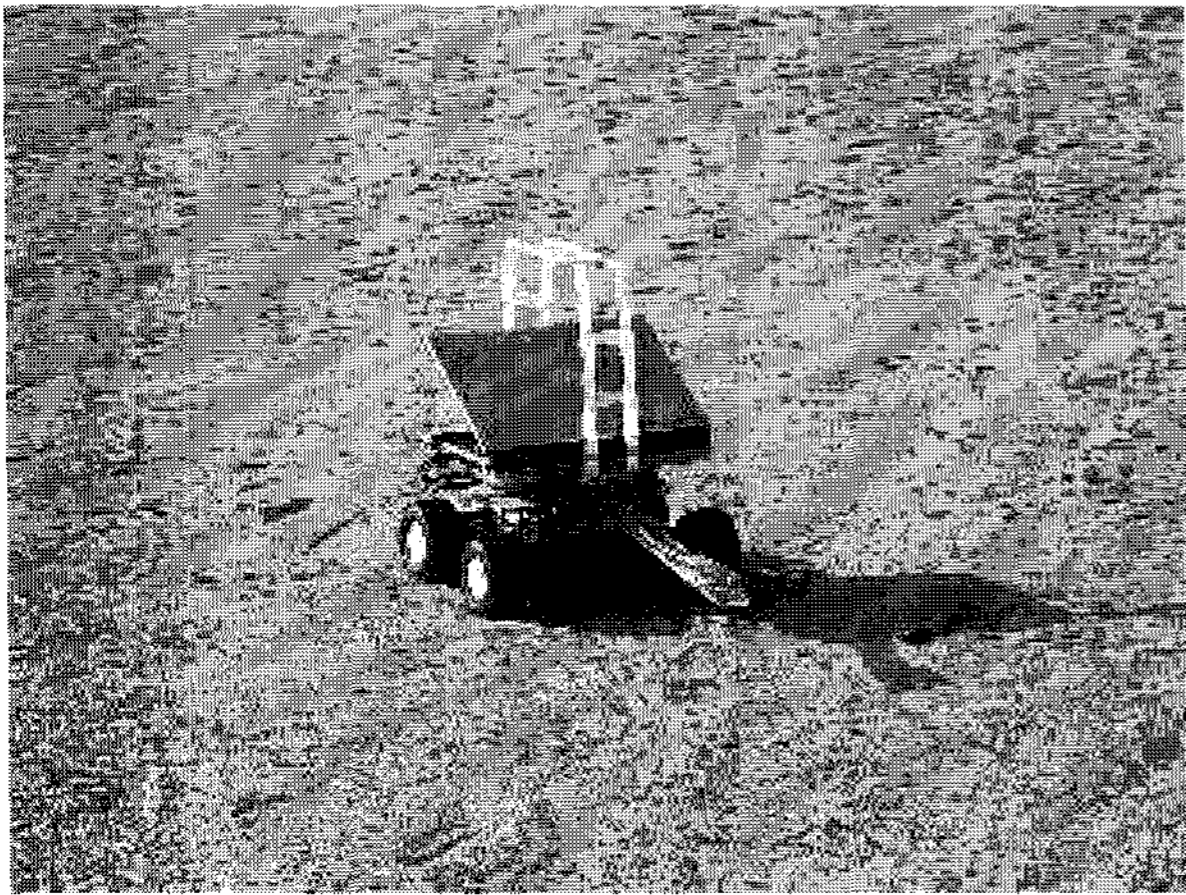


그림 5 자율 이동 로봇
Fig. 5 Autonomous Mobile Robot

화면이다. 실험에서 설정한 그림 2의 $(x_s, y_s), (x_d, y_d), (x, y)$ 의 실제적인 위도, 경도값과 위도 36.96649, 경도 127.86751인 위치를 기준으로한 국소적(local)인 위도, 경도값은 표 3과 같다.

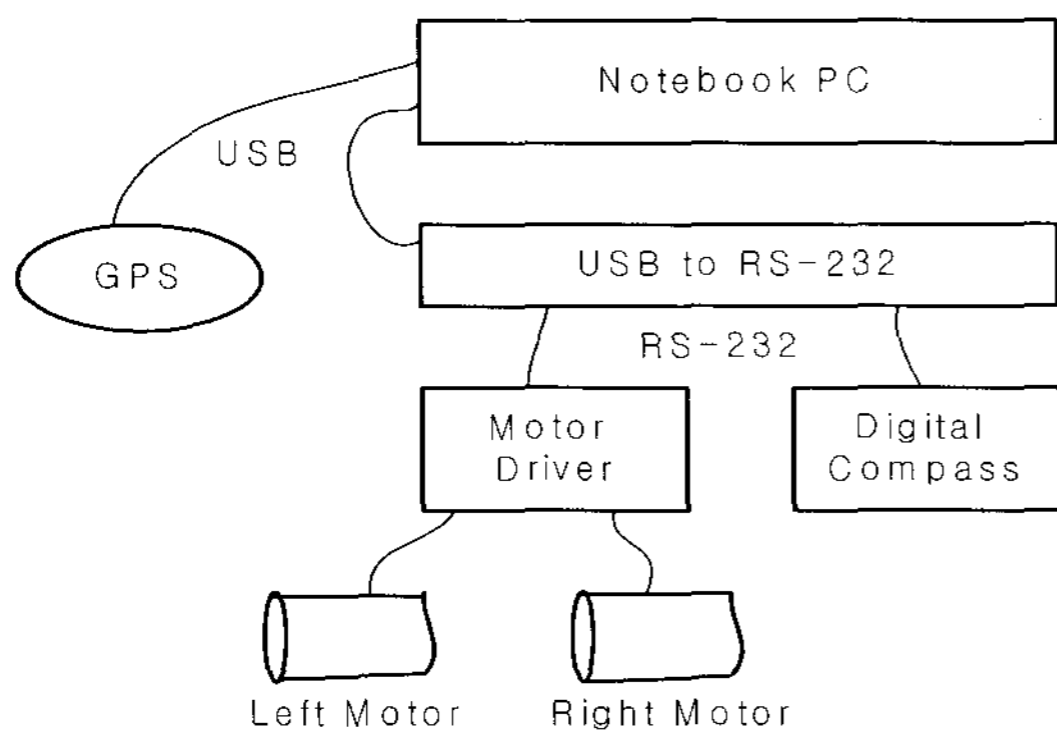


그림 6 자율이동로봇 블록도
Fig. 6 Block Diagram of Autonomous Mobile Robot

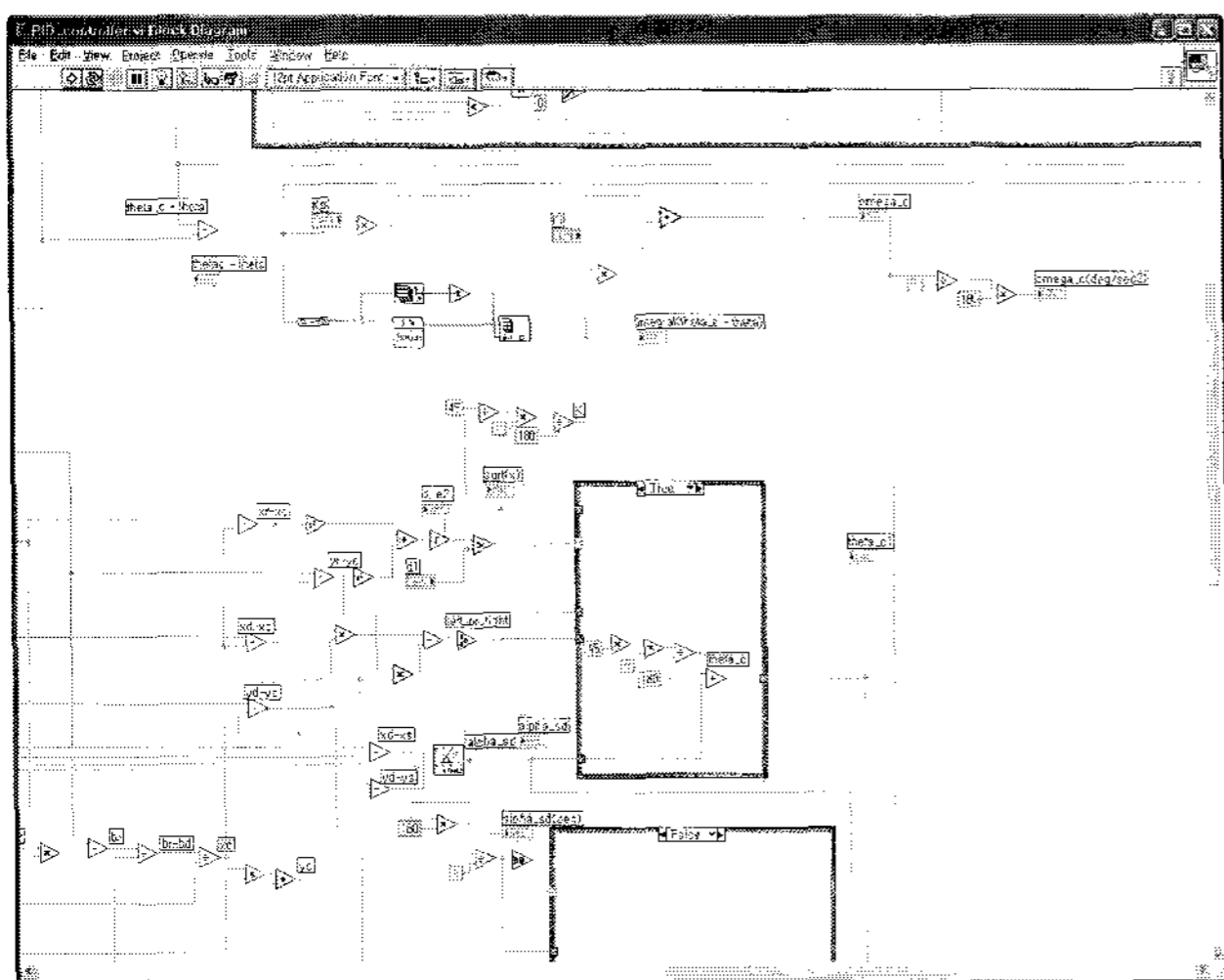


그림 7 LabVIEW 프로그램
Fig. 7 LabVIEW Program

표 3 실험에 사용된 실제 위도 경도값 및 국소적 위도 경도값

Table 3 Latitude and Longitude Coordinates for the Experiment

	x_s, y_s	x_d, y_d	x, y
실제위도값	36.968146	36.96803	36.96786
실제경도값	127.86863	127.86972	127.86867

	x_s, y_s	x_d, y_d	x, y
국소적위도값[m]	184.139	171.609	148
국소적경도값[m]	99.8284	196.517	102.05

그림 8은 실험결과로 나온 이동로봇의 궤적도이며 GPS 데이터의 노이즈가 섞여있기는 하지만 시뮬레이션에서 기대했던 것과 유사한 로봇의 이동궤적을 얻을 수 있었다. 그림 8에서 5000~8000 step 사이의 로봇 방향각은 $35^\circ \sim 36^\circ (= -1^\circ \sim 0^\circ)$ 이며 그림 4의 시뮬레이션 결과와 유사함을 알 수 있다.

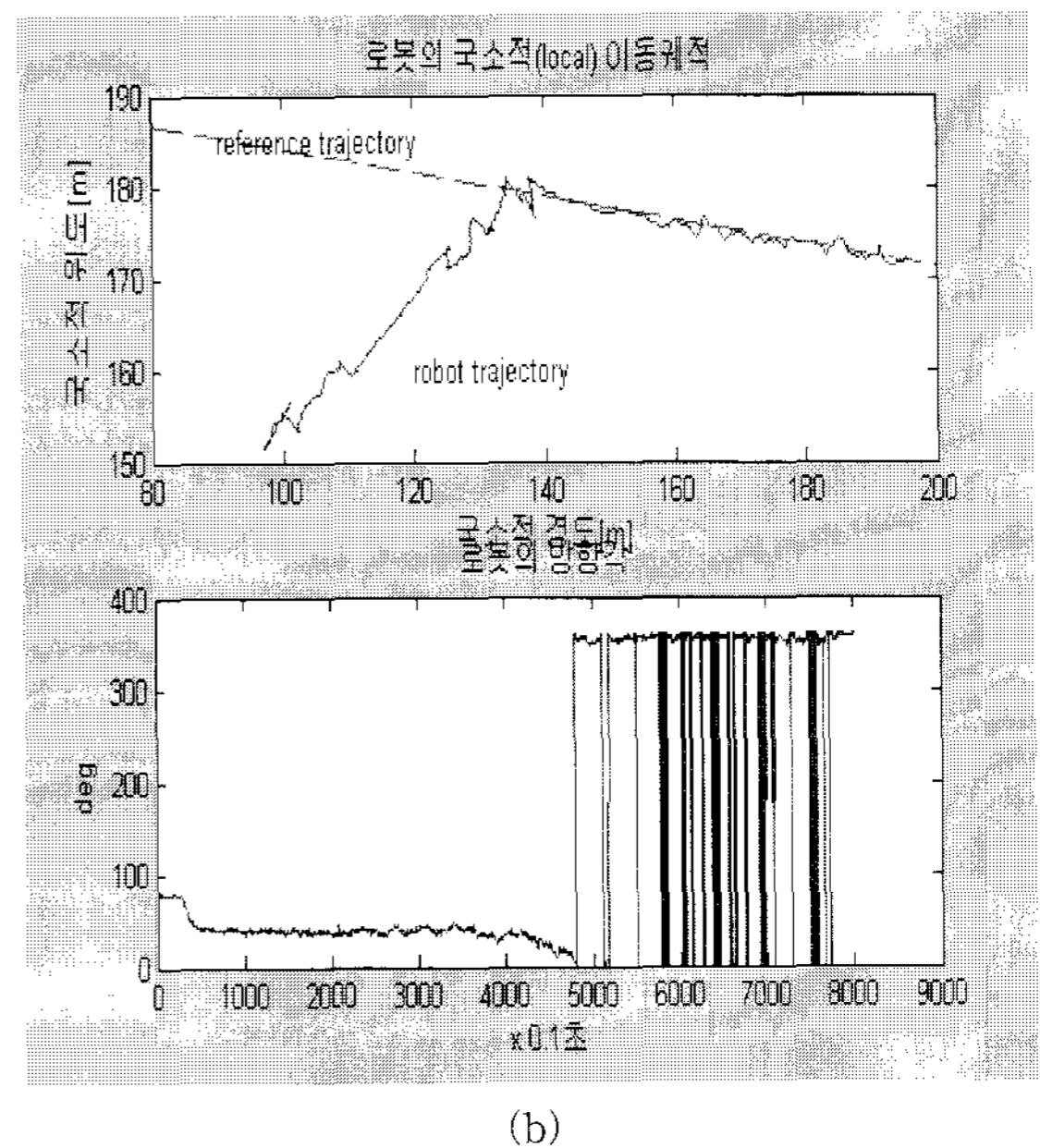
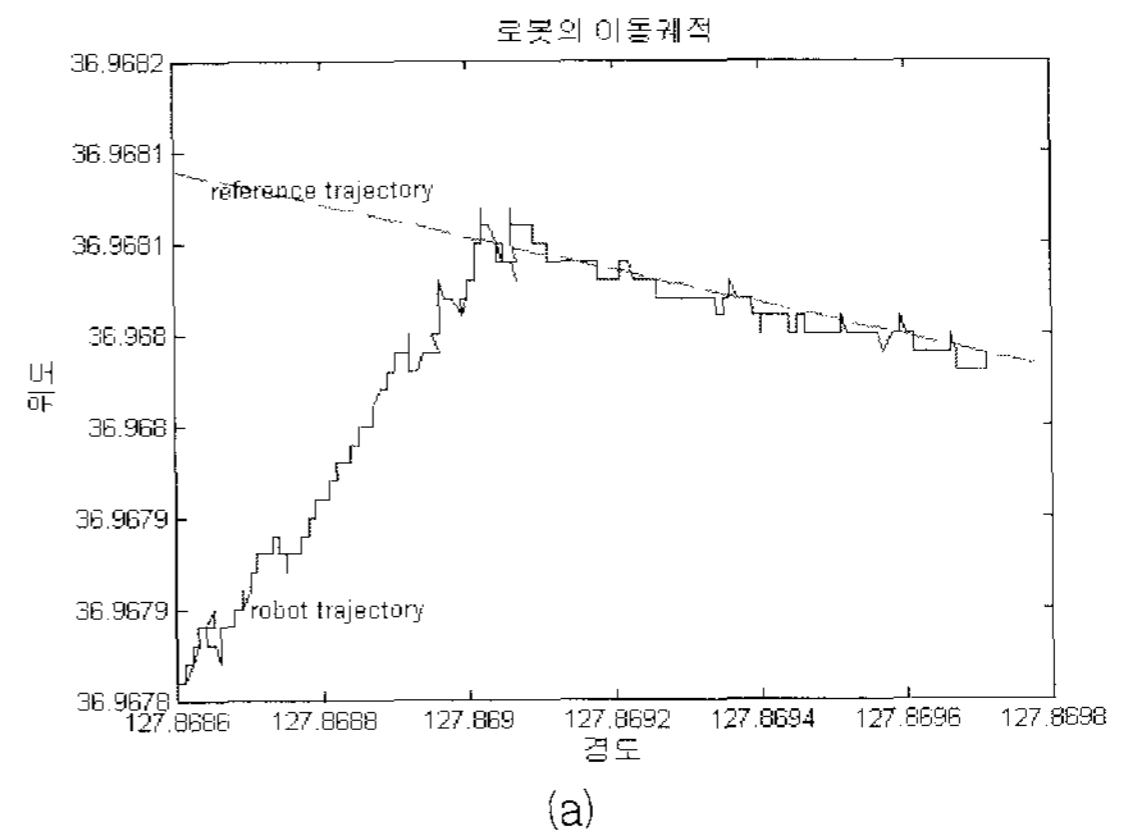


그림 8 로봇 이동궤적 및 방향각(실험) (a)실제, (b)국소적
Fig. 8 Robot Trajectory and Heading Angle(experiment)

3. 결 론

본 논문에서 고려한 대상은 야외에서 GPS와 디지털 컴퍼스에 의해 위치와 방향 정보를 얻어 움직이는 이동로봇이다. 본 연구에서는 비교적 간단한 PI 제어 알고리즘을 이동로봇의 위치제어에 적용하였고 시뮬레이션과 실험을 통해 좋은 결과를 얻을 수 있었다. 제시된 제어기는 간단한 PI 제어기의 구조상 특성으로 인해 실제 실시간 제어 프로그램화 하였을 때 샘플링 시간이 짧아지고 따라서 이동로봇 위치제어 기 구현에 적합한 형태임을 실제 실험을 통해 확인하였다. 또한 본 연구를 통해 최근 많이 사용되고 있는 그래픽 프로그램 언어인 LabVIEW가 이동로봇의 위치제어 프로그래밍에도 유용하게 사용될 수 있음을 확인하였다.

부 록

그림 2에서 로봇의 위치 (x, y) 는 GPS 수신좌표(동경, 북위좌표)로부터 실시간으로 얻으며 $(x_s, y_s), (x_d, y_d)$ 는 주어진 좌표값이다. 이들 $(x, y), (x_s, y_s), (x_d, y_d)$ 로부터 e_1 은 수학적으로 다음과 같이 계산된다. 그림 2에서 (x_s, y_s) 와 (x_d, y_d) 를 연결하는 직선의 기울기를 a_d 라 하면 a_d 는 다음과 같다.

$$a_d = \frac{y_d - y_s}{x_d - x_s}$$

그러면 (x_s, y_s) 와 (x_d, y_d) 를 연결하는 직선의 식은 다음을 만족해야 하며

$$y = a_d x + b_d, \quad b_d = y_d - a_d x_d$$

이 직선에 수직이고 (x, y) 을 지나는 직선의 식은 다음과 같다.

$$y = -\frac{1}{a_d} x + b_r, \quad b_r = y_r - a_r x_r$$

이 두 직선의 교점 (x_c, y_c) 는 다음과 같고

$$x_c = \frac{b_r - b_d}{a_d + \frac{1}{a_d}}, \quad y_c = a_d x_c + b_d$$

따라서 두 점 (x_r, y_r) 과 (x_c, y_c) 사이의 거리 $|e_1|$ 은 다음과 같이 계산된다.

$$|e_1| = \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2}$$

참 고 문 헌

- [1] 주영훈, 우광방, 김경배, "이동 로봇의 주행 위치 추정에 관한 연구", 대한 전기학회 논문지, 44권 3호, pp. 372-380, 1995
- [2] 노영식, 박경섭, "PDS를 이용한 이동 로봇의 위치보정", 대한 전기학회 논문지, 44권 8호, pp. 1100-1107, 1995

- [3] 유원필, 정명진, "실내 주행 이동 로봇을 위한 컬러 이미지를 이용한 표식점 위치 측정 방법", 대한 전기학회 논문지, 45권 2호, pp. 310-318, 1996
- [4] Siegwart, Roland and Nourbakhsh, Illah R. *Introduction to Autonomous Mobile Robot*, The MIT Press, 2004
- [5] Dong, Wenjie and Kuhnert, K. D., "Robust Adaptive Control of Nonholonomic Mobile Robot With Parameter and Nonparameter Uncertainties," IEEE Trans. Robotics, Vol. 21, No. 2, pp. 261-266, 2005
- [6] Gu, Dongbing and Hu, Huosheng, "A Stabilizing Receding Horizon Regulator for Nonholonomic Mobile Robots," IEEE Trans. Robotics, Vol. 21, No. 5, pp. 1022-1028, 2005
- [7] Coelho, Paulo and Nunes, Urbano, "Path-Following Control of Mobile Robots in Presence of Uncertainty," IEEE Trans. Robotics, Vol. 21, No. 2, pp. 252-261
- [8] Farrell, Jay A., Pang, Shuo, and Li Wei, "Chemical Plume Tracing via Autonomous Underwater Vehicle," IEEE Journal of Oceanic Engineering, Vol. 30, No. 2, pp. 428-442, 2005
- [9] Bruce Mihura, *LabVIEW for Data Acquisition*, Prentice Hall, 2001

저 자 소 개



박 영 환 (朴 永 煥)

1986년 고려대 전기공학과 졸업. 1998년 동대학원 전기공학과 졸업(공학). 1998년~현재 충주대 전기 전자 및 정보공학부 부교수.

Tel : 043-841-5328

Fax : 043-841-5320

E-mail : pyh@cjnu.ac.kr



이 재 경 (朴 永 煥)

1980년 서울대학교 기계공학과 졸업. 1996년 한국과학기술원(KAIST) 기계공학과 졸업(박사). 1997년~현재 충주대 전기전자 및 정보공학부 교수.

Tel : 043-841-5327

Fax : 043-841-5320

E-mail : jklee@cjnu.ac.kr