

Unicycle-type 이동로봇의 비선형 제어

김 용 진 , 문 인 혁

연세대학교 의용전자공학과

첨단의료기기 기술혁신센터, 의용계측 및 재활공학 연구센터, 의공학연구소

Nonlinear control of unicycle-type mobile robot

Yong-Jin Kim , In-Hyuk Moon

Dept. Biomedical engineering, Technology Innovation Center for Medical Instrument, Reserch Institute for Medical Instruments & Rechabilitation Engineering, Reserch Institute of Medical Engineering, Yonsei University

E-mail: tonko@hanmail.net

Abstract

This paper proposes a stable control rule for nonli-near unicycle-type mobile robot. The control method uses a local error coordinate system, velocity and distance constants k_x, k_y and k_θ . Stability of control rule is proved Liapunov function. System input to the mobile robot is reference posture $(x_r, y_r, \theta_r)^T$ and referenc e velocity $(v, \omega)^T$. System output of the mobi-ble robot is velocity of driving wheels. We introduce limit veloc ity for preventing high initial speed. From simulation results, we can see the proposed control rule is stable.

1. 서론

unicycle-type 이동로봇은 직선과 회전제어 입력으로 궤도를 제어할 수 있기 때문에 로봇의 연구에 많이 사용 된다. 하지만, unicycle-type의 운동모델은 비선형이므로 선형제어 방식을 적용시키기에 어려움이 있다.

unicycle-type 이동로봇의 제어입력을 Kanayama[1] 는 오차상태방정식을 이용하여 구하고, 최대속도와 최대 가속도로 제어입력을 제한하는 방법을 제안하였다.

Aicardi[2]는 극좌표계를 사용한 제어 방법을 제안하 였다.

본 논문에서는 unicycle-type 이동로봇의 목표상태 (위치, 방향)와 목표속도(직진, 회전)를 기준입력으로 하 고, 직진과 회전 제어입력 $u = [v \ \omega]^T$ 를 구하는 방법을 제안한다. 로봇의 현재상태와 목표상태간의 오차를 현재

상태의 지역좌표계로 변환하여 오차상태방정식으로 나타내고, 현재속도 v_c 에 대한 속도상수 k_x 와 현재 회전 속도 ω_c 에 대한 거리상수 k_y, k_θ 를 도입하여 제어입력 을 결정한다. 제어입력의 안정성은 *Liapunov* 안정도 해석법을 사용하여 그 안정성을 보인다.

제안된 제어 방법은 시뮬레이션을 통하여 그 유효성 을 입증한다.

2. 이동로봇의 운동관계식

2.1 이동로봇의 운동방정식

본 논문에서는 독립적인 두 개의 구동 바퀴를 가진 unicycle-type 이동로봇을 제어 대상으로 한다. 각각 의 좌우 바퀴속도의 입력으로 로봇의 직진속도 v 와 회 전 속도 ω 를 제어한다.

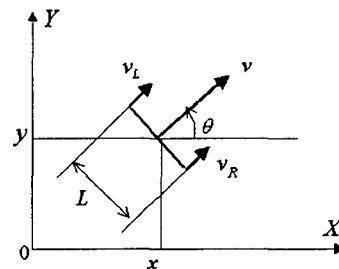


그림 1. 로봇의 위치와 속도

이동로봇의 상태는 2차원 평면상의 위치와 방향으로 나타낸다(그림1 참조).

$$p = [x \ y \ \theta]^T \quad (1)$$

직선운동속도 v , 회전운동속도 ω 를 제어입력이라고 할 때 운동방정식은 다음과 같다.

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = J \cdot u_2 \quad (2)$$

여기서, 이동로봇의 두 바퀴 사이의 거리를 L 이라 할 때 $v = \frac{v_R + v_L}{2}$, $\omega = \frac{v_R - v_L}{L}$ 이다. 따라서 v 와 ω 가 주어지면 v_R , v_L 은 다음과 같다.

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} 1 & \frac{L}{2} \\ 1 & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

2.2 이동로봇의 오차상태방정식

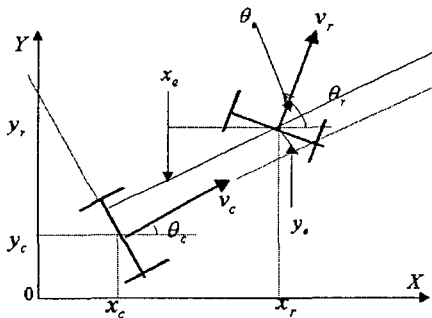


그림 2. 이동로봇 상태간의 오차

목표상태와 목표속도를 각각 $p_r = [x_r \ y_r \ \theta_r]^T$, $u_r = [v_r \ \omega_r]^T$ 라하고, $p_c = [x_c \ y_c \ \theta_c]^T$ 를 현재의 로봇 상태라고 할 때, 두 상태간의 오차 p_e 는 다음과 같다(그림2 참조).

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_c & \sin \theta_c & 0 \\ -\sin \theta_c & \cos \theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} = T_e \cdot (p_r - p_c) \quad (4)$$

식 (4)을 미분하면 \dot{p}_e 를 얻을 수 있고, 다음과 같이 \dot{p}_e 를 분리하여 나타낼 수 있다.

$$\dot{p}_e = \begin{bmatrix} y_e \cdot \omega_c - v_c + v_r \cdot \cos \theta_e \\ -x_e \cdot \omega_c + v_r \cdot \sin \theta_e \\ \omega_r - \omega_c \end{bmatrix} = \begin{bmatrix} 0 & \omega_c & 0 \\ -\omega_c & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} p_e \quad \text{만약, } v_r = \omega_r = 0 \text{ 이 되면 식 (7)의 선형화 모델에}$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin \theta_e \\ 0 \end{bmatrix} v_r \quad (5)$$

여기서, $u_1 = -v_c + v_r \cos \theta_e$, $u_2 = \omega_r - \omega_c$ 이다.

2.3 제어 시스템의 블록선도

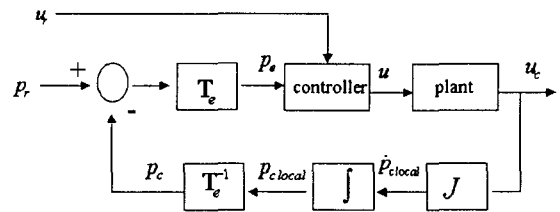


그림3은 이동로봇의 제어 블록선도를 나타낸다. 목표 상태 p_r 과 목표속도 u_r 를 입력으로 하여 제어입력 u 를 구하고, 로봇의 양 바퀴의 속도 v_R, v_L 로 부터 로봇의 현재속도 u_c 를 구할 수 있다. u_c 와 식 (2)를 사용하여 \dot{p} 를 구한 후 적분하여 $p_{c local}$ 을 얻는다. $p_{c local}$ 을 역좌표계 변환하여 절대좌표계의 p_c 를 구하고 그 위치를 되먹임한다.

3. 제어입력의 결정

3.1 제어 입력 u 의 결정

제어 입력 u 는 목표속도 u_r 과 상태오차 p_e 에 의해 결정되고, 현재의 출력속도는 u_c 는 제어입력에 추종한 다고 가정하고, 식 (5)로 부터 다음과 같다.

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -v_c + v_r \cos \theta_e \\ \omega_r - \omega_c \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = u_c \quad (6)$$

3.2 제어 입력을 위한 상수 도입

식 (5)를 공칭점(equilibrium point) $p_e = 0$ 에서 선형화 하 하면 다음과 같다.

$$\dot{p}_e = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} p_e = \bar{A} \cdot p_e \quad (7)$$

Unicycle-type 이동로봇의 비선형 제어

서 \bar{A} 는 0행렬이 되어 제어 능력을 상실하게 된다. 이런 상황을 고려해서 양의 상수 k_x, k_y, k_θ 를 도입한다. 그림 2에서 v_c 는 x_e 와 시간상수 k_x 에 의해 $v_c = -k_x$

$\cdot x_e$ 로 나타나고, ω_c 는 y_e, θ_e 와 거리상수 k_y, k_θ 에 의해 $\omega_c = -k_y \cdot v_r \cdot y_e - k_\theta \cdot v_r \cdot \sin \theta_e$ 로 나타낼 수 있다. 식 (5)에 대입하면 다음과 같은 제어입력을 정의 할 수 있다.

$$u = \begin{bmatrix} x_e \cdot k_x + v_r \cdot \cos \theta_e \\ \omega_r + v_r \cdot k_y \cdot y_e + v_r \cdot k_\theta \cdot \sin \theta_e \end{bmatrix} \quad (8)$$

3.3 \dot{p}_e 의 선형화

새롭게 정의된 제어입력인 식 (8)을 식 (5)에 대입하여 공칭점 $p_e = 0$ 에서 선형화 하면 다음과 같이 된다.

$$\dot{p}_e = \begin{bmatrix} -k_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -v_r \cdot k_y & -v_r \cdot k_\theta \end{bmatrix} p_e = A \cdot p_e \quad (9)$$

3.4 제어입력 u 에 대한 공칭점 $p_e = 0$ 에서 안정성

3.4.1 Liapunov 안정도 해석

Theory 1. Liapunov 안정도

$\dot{x} = f(x, u)$ 에서 모든 t 에 대해서 $f(0, t) = 0$ 이고, 연속인 일차변도함수 $\dot{V}(x, t)$ 가 존재 할 때 $V(x, t)$ 가 양의 한정 ($V(x, t) > 0$)이고, $\dot{V}(x, t)$ 가 음의 한정 ($\dot{V}(x, t) < 0$)이면, 평형 상태인 원점은 균일하게 점근 안정하다[3].

따라서, 식 (8)에 의해 제어입력 u 를 사용시 $v_r > 0$ 일 때 $p_e = 0$ 의 안정성을 증명하기 위해서 다음과 같이 Liapunov 함수 V 를 제안한다.

$$V = \frac{1}{2}(x_e^2 + y_e^2) + (1 - \cos \theta_e)/k_y \quad (10)$$

이때, 식 (10)은 $p_e = 0$ 이면 $V = 0$ 이고, $p_e \neq 0$ 이면

$V > 0$ 이므로, 항상 $V \geq 0$ 이 되어 양의 한정이 된다.

또한, 연속인 일차변도함수 $\dot{V}(x, t)$ 는 다음과 같다.

$$\dot{V} = \dot{x}_e \cdot x_e + \dot{y}_e \cdot y_e + \dot{\theta}_e \cdot \sin \theta_e / k_y \quad (11)$$

식 (8)을 (4)식에 대입하여 그 결과를 식 (11)에 대입하면 \dot{V} 는 다음과 같다

$$\dot{V} = -k_x \cdot x_e^2 - v_r \cdot k_\theta \cdot \sin^2 \theta_e / k_y \quad (12)$$

여기서, $\dot{V} \leq 0$ 이 된다. 따라서, $V \geq 0$ 과 $\dot{V} \leq 0$ 에 의해서 V 는 **Theory 1**을 만족한다. 따라서 제어입력 u

를 사용시, $v_r > 0$ 인 조건에서 $p_e = 0$ 은 안정한 공칭점이라 할 수 있다.

3.4.2 $p_e = 0$ 에서 안정도 조건

Theory 2. Routh-Hurwitz 안정도

특성방정식의 모든 근이 복소평면상의 좌반평면 위에 존재하면 시스템은 안정하다[3].

식 (9)에서 $\dot{p}_e = A \cdot p_e$ 이고, 공칭점 $p_e = 0$ 에서 선형화된 식을 다음과 같이 표현할 수 있다.

$$\dot{p}_e = A \cdot p_e + B \cdot u \quad (13)$$

이때, 식 (13)의 특성방정식은 다음과 같다.

$$s^3 + (k_x + v_r \cdot k_\theta) \cdot s^2 + (k_x \cdot v_r \cdot k_\theta + \omega_r^2 + v_r^2 \cdot k_y) \cdot s + \omega_r^2 \cdot v_r \cdot k_\theta + v_r \cdot k_y \cdot k_x = 0 \quad (14)$$

따라서, Routh-Hurwitz 안정도 해석법을 사용하면,

$$a_3 = 1$$

$$a_2 = v_r \cdot k_\theta + k_x$$

$$a_1 = k_y \cdot v_r^2 + k_x \cdot k_\theta \cdot v_r + \omega_r^2$$

$$a_0 = k_x \cdot k_y \cdot v_r^2 + \omega_r^2 \cdot k_\theta \cdot v_r$$

이 된다. 또한, a_3, a_2, a_1, a_0 가 모두 양의 값이고, 그때 $a_2 \cdot a_1 - a_3 \cdot a_0 > 0$ 이 되어, 특성방정식의 모든 근이 좌반평면에 존재하므로 불안정 극점이 존재하지 않는다.

4. 이동로봇의 현재상태 추정.

로봇의 각 바퀴에서 측정된 v_R 과 v_L 로 부터 식 (2)와 (3)을 사용하여 로봇좌표계에서 \dot{p}_c 를 구하면 다음과 같다.

$$\dot{p}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta_c}{2} & \frac{\cos \theta_c}{2} \\ \frac{\sin \theta_c}{2} & \frac{\sin \theta_c}{2} \\ \frac{1}{L} & \frac{-1}{L} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix} \quad (15)$$

식 (15)의 \dot{p}_c 를 적분하면 현재의 상태 $p_{c, local}$ 을 구할 수 있고, 좌표축변환을 하여 절대좌표계에서의 p_c 를 계산할 수 있다.

5. 실험

이동로봇의 목표위치 $p_r = [x_r, y_r, \theta_r]^T$ 을 주었을

때의 로봇의 제어에 대한 시뮬레이션을 수행하였다. 목표 속도 $u_r = [v_r, \omega_r]^T$ 을 고정된 입력으로 하고, p_r 을 주었을 때, 제어입력에 대한 상수 k_x, k_y, k_θ 에서 k_x 는 그 값이 크면 x_e 가 빨리 감소 하지만, k_x 가 너무 크면 로봇의 주행이 불안정하게 되므로 실험적으로 가장 적절한 값을 설정했다.

\dot{p}_c 가 언더슈트나 오버슈트가 발생되지 않는 임계값

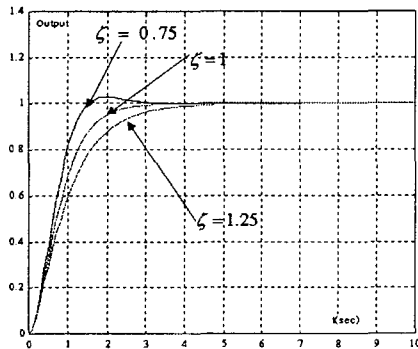


그림 4

핑(critical damping) 상태에서 실험적으로 가장 적절한 k_y, k_θ 를 설정했다(그림 4참조).

시뮬레이션은 $v_r = 30\text{cm/sec}$, $\omega_r = 0.18\text{rad/sec}$, $k_x = 15/\text{sec}$, $k_y = 0.0064/\text{cm}$, $k_\theta = 0.001/\text{cm}$, 샘플링

시간을 $T_s = 10\text{ms}$ 로 하여 시뮬레이션을 수행하였다.

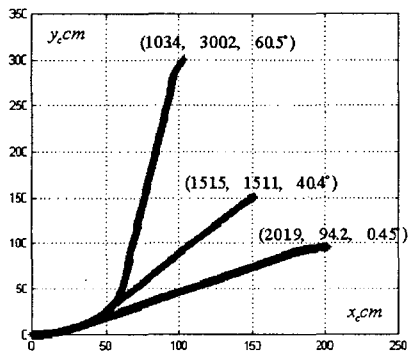


그림 5

그림 5는 목표상태를 (100, 300, 60°), (150, 150, 40°), (200, 100, 0°)로 주었을 때 그때의 주행경로 및 정지상태를 나타낸다.

로봇의 초기 주행시 직진속도와 회전속도는 큰 값이 요구되어지만 로봇의 최대속도로 제한한다. 본 실험에 사용된 시스템은 직진과 회전속도를 각각 50cm와 0.8°를 최대값으로 제한하여 가용속도 범위에서 동작하도록 제어하였다.

그 결과 로봇의 주행제어가 안정적으로 이루어졌지만 1% 정도의 오차를 가짐을 볼 수 있다.

6. 결론

이 논문에서는 목표상태 p_r 과 목표속도 u_r 의 입력과 상수 k_x, k_y, k_θ 의 도입으로 제어입력 u 를 결정하고, Liapunov 함수로 그 안정성을 입증하였다. 또한 로봇의 각 바퀴에서 측정된 속도 v_R, v_L 을 사용하여 로봇의 현재상태를 구하고, 그 상태를 되먹임 하는 제어 방법을 제안하였다.

로봇의 초기 주행시 나타나는 높은 속도를 제어하기 위해서 최대 직진속도와 최대 회전속도를 사용하는 방법을 보였다.

그림 5의 결과로 부터 로봇의 주행 후 상태가 목표상태로 수렴하고 있음을 볼 수 있다. 따라서, 이 논문에서 제시한 unicycle-type 이동로봇의 비선형제어가 안정적으로 이루어짐을 알 수 있다.

본 시스템은 거리계(odometry)만을 이용한 dead reckoning 제어였는데, 차후 자이로 센서 등의 방향계 센서를 적용하면 좀더 정확한 주행제어가 가능 할 것이다.

7. 참고문헌

- [1]Yutaka Kanayama 외. "A Stable Tracking Control Method for an Autonomous Mobile Robot " Pro. Int. Conf. R&A. 1990.
- [2]M.Aicard 외. "Closed Loop Steering of Unicycle-like Vehicles via Lyapunov Techniques" Ieee Robotics & Automation Magazine. 1995. 3.
- [3]Katsuhiko Ogata 외. "Modern Control Engineering" Scitech Media. pp.885-896, 1999. 7.