

# 자율 이동 로봇을 위한 안정한 주행 제어 방법

심현식<sup>o</sup>, 김종환  
한국과학기술원 전기및전자공학과

## A Stable Locomotion Control Method for an Autonomous Mobile Robot

Hyun-Sik Shim and Jong-Hwan Kim  
Dept. of Elec. Eng., KAIST

### Abstract

This paper proposes a stable locomotion control rule for non-holonomic mobile robot. Stability of the rule is proved through the use of a Liapunov function. We have two controller for locomotion control. One is velocity controller, the other is position controller. The proposed controller is position controller whose input to robot are a reference posture and reference velocities. The major objective of this paper is to propose a control rule to find a reasonable velocity command under a assumption which is velocity controller is ideal controller.

### 1 서론

최근 고정 궤도식 무인운반차(AGV)의 결점을 보완하고 유연성을 높이기 위해 자유주행방식을 채택한 이동로봇에 관한 연구가 활발하다. 로봇이 주어진 기준경로를 정확히 따라 가도록 하기 위해서는 벗어난 현재의 자세로부터 기준 경로에 복귀하도록 하는 경로제어 알고리즘들을 필요로 한다. 그러나, 휠 구동 방식의 이동로봇의 제어는 기구학적으로 Non-holonomic Constraint를 가지고 있어 안정된 경로제어 알고리즘을 구현하는데 어려움이 있다. 이전의 연구들에서 이러한 문제점을 극복하기 위한 여러가지 경로 제어 알고리즘이 제시되었다. Kanayama[2]는 로봇의 이동을 위한 직선 기준경로를 사용하는 방법을 제시하였다. Crowley[3]등은 3개의 계층구조를 갖는 이동 제어 시스템을 개발하였는데, 여기서 가장의 이동개념을 도입하였다. 이 방법은 직진과 선의운동의 독립적인 제어가 가능하며, 부드러운 궤적을 얻을 수 있는 특징이 있다. Kanayama [6]는 기준점과 현재점의 자세로부터, 극부오차좌표를 설정하여, PID제어의 형태와 직진과 회전속도를 위한 제어법칙을 제시한 바 있다. 또한 최근 Kanayama [1]는 비선형형태를 갖는 조향 및 선속도 제어법칙을 발표하였는데, 그안정성을 Liapunov함수를 사용하여 증명하였다. 그러나  $v_c$ 이 0이면  $y$ 의 오차를 0으로 보내지 못하는 단점이 있다.

본 논문에서는 위의 알고리즘의 단점을 극복하고 유연한 경로제어를 할수있는 방법을 제시한다. 먼저 2절에서는 이동 로봇의 기구학과 동역학에 대한 내용을 다룬다. 이것들은 속도 제어를 구성할때 쓰일뿐만 아니라 시뮬레이션시 사용한다. 그다음 3절에서는 이동로봇의 주행제어를 설계하고 Liapunov 정리를 사용해 제안된 알고리즘의 안정도를 증명한다. 제안한 제어 방법에 대한 개요도를 그림 3에 나타낸다. 여기서, 두 개의 제어기가 존재하는데 하나는 속도제어기로서 어떠한 상황 아래에서도 주어진 속도 명령을 잘 따라가도록 구성된 제어기이고 위치제어기는 속도제어기가 이상적인 제어기라는 가정아래서 구성된 것이다. 이 논문에서는 속도제어기는 구성되었다고 가정하고 위치 제어기만 다룬다. 속도제어기는 로봇트 메니플레이터에 사용된 제어 알고리즘들을 비슷한 방법으로 사용할수있다. 그 다음 4절에서는 시뮬레이션을 하고 같은 결론을 내린다.

### 2 기구학과 동역학

몇가지 가정아래서 정역학과 동역학식을 구한다.

#### 2.1 경로표현과 기구학식

직교좌표에서 정의된 2차원의 평면에 놓여진 로봇트가 그림 1에서 처럼 있다. 로봇트는 위치를 나타내는데 3자유도를 가지고 있으며 자세라고 정의한다.

$$p = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (2.1)$$

여기서, 방향각  $\theta$ 는 X축에서 반시계방향으로 취해진 값이다. 로봇트는 평면하에서 주행 능력을 가지고 있으므로 실제로 자세  $p$ 는 시간  $t$ 의 함수이다.  $(x(t), y(t))$ 은 궤적(path trajectory)라고 부른다.

로봇트의 이동은 시간 함수인 선속도  $v$ 와 각속도  $\omega$ 으로 제어되고 로봇트의 정역학은 자코비안 행렬  $J$ 의 의해 다음과 같이 정의된다:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \dot{p} = Jq = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} q \quad (2.2)$$

여기서,  $q = (v \ \omega)^T$ .

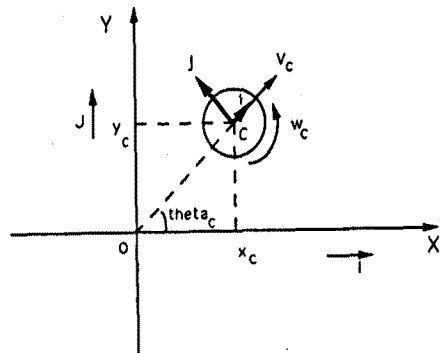


그림 1. 이동 로봇트의 절대좌표와 상대 좌표

## 2.2 자세 오차

이 논문에서, 두가지의 자세가 사용된다; 기준 자세 (reference posture)  $p_r = (x_r, y_r, \theta_r)^T$  과 현재 자세 (current posture)  $p_c = (x_c, y_c, \theta_c)^T$ . 기준 자세는 로봇의 목표 자세이고 현재 자세는 그 순간의 실제 위치이다. 지금 경로기준자세오차 (reference based error posture)  $p_{er}$  를 정의한다. 그것은 기준좌표를 기준경로의 자세  $p_r$  에서 설정하여 정의되는 것으로, 이 좌표계로  $p_c$  를 좌표변환한것이고 다음식으로 구해진다.

$$p_{er} = \begin{pmatrix} x_{er} \\ y_{er} \\ \theta_{er} \end{pmatrix} = \begin{pmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{pmatrix} (p_r - p_c) \quad (2.3)$$

다음 보조 정리는 다음 절에서 사용될것이다.

### 보조정리 1

$$\dot{p}_{er} = \begin{pmatrix} \dot{x}_{er} \\ \dot{y}_{er} \\ \dot{\theta}_{er} \end{pmatrix} = \begin{pmatrix} y_{er}\omega_r + v_r - v_c \cos\theta_{er} \\ -x_{er}\omega_r + v_c \sin\theta_{er} \\ \omega_r \end{pmatrix} \quad (2.4)$$

Proof : 이 보조정리는 Kanayama [1]의 것과 비슷한 방법으로 유도된다. 식 (2.3)과 (2.2)로부터 부동식  $\dot{x} \sin\theta = \dot{y} \cos\theta$  를 사용하여 다음 식들을 유도할수있다.

$$\begin{aligned} \dot{x}_{er} &= y_{er}\omega_r + v_r - \dot{x}_c \cos\theta_r - \dot{y}_c \sin\theta_r \\ &= y_{er}\omega_r + v_r - v_c \cos\theta_c \cos\theta_r - v_c \sin\theta_c \sin\theta_r \\ &= y_{er}\omega_r + v_r - v_c \cos\theta_{er} \\ \dot{y}_{er} &= -x_{er}\omega_r + \dot{x}_c \sin\theta_r - \dot{y}_c \cos\theta_r \\ &= -x_{er}\omega_r + v_c \cos\theta_c \sin\theta_r - v_c \sin\theta_c \cos\theta_r \\ &= -x_{er}\omega_r + v_c \sin\theta_{er} \\ \dot{\theta}_{er} &= \dot{\theta}_r - \dot{\theta}_c = \omega_r - \omega_c = \omega_{er} \quad \blacksquare \end{aligned}$$

## 2.3 동역학

우리는 동역학식을 유도하기 위해 몇가지 가정을 한다:

- 1) 로봇트는 강체로만 이루어져 있다.
- 2) 모든 회전축은 표면에 수직이다.
- 3) 로봇트는 평평한 표면위만 움직인다.
- 4) 바퀴와 표면사이에는 미끄러짐이 없다.
- 5) 무게중심과 도심은 일치한다.

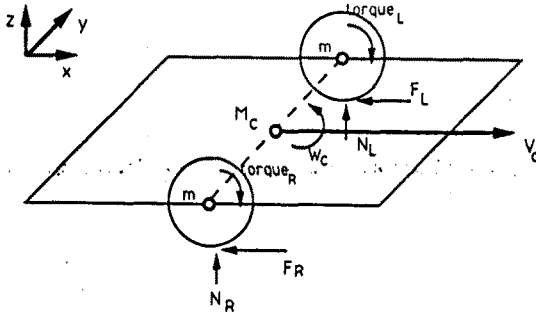


그림 2. 이동 로봇트의 Free body diagram

로봇트의 무게중심에서 제각기 선속도와 각속도벡터인  $v_c \vec{i}$  과  $\omega_c \vec{k}$  를 가진다.

$$v_c \vec{i} = \dot{r}_c \vec{i} = v_c \cos\theta_c \vec{i} + v_c \sin\theta_c \vec{j} \quad (2.5)$$

$$\omega_c \vec{k} = \dot{\theta}_c \vec{k} \quad (2.6)$$

동역학식을 얻기위하여, 우리는 Lagrange's method 를 사용한다. 다음과 같이 운동에너지  $T$  를 취한다:

$$T = \frac{1}{2} M v_c^2 + \frac{1}{2} I_c \omega_c^2 + \frac{1}{2} I_w (\omega_L^2 + \omega_R^2) \quad (2.7)$$

여기서,  $M, m, I_c, I_w, \omega_L$  and  $\omega_R$  는 각각 전체 로봇트의 무게, 바퀴 무게, 물체의 관성모멘트, 바퀴의 관성모멘트, 왼쪽 바퀴의 각속도, 오른쪽 바퀴의 각속도이다.

다음식은 물체의 속도와 바퀴의 각속도사이의 관계를 나타낸다.

$$\begin{aligned} \dot{r}_c &= v_c = \frac{r}{D} (\omega_R + \omega_L) \\ \dot{\theta}_c &= \omega_c = \frac{r}{D} (\omega_R - \omega_L) \end{aligned}$$

여기서,  $r$  과  $D$  는 바퀴의 반지름과 물체의 지름이다. 위 식들을 각각 제곱하고 더하면, 다음 식을 얻는다.

$$\omega_L^2 + \omega_R^2 = \frac{2}{r^2} (v_c^2 + D^2 \omega_c^2) \quad (2.8)$$

그래서 식 (2.7) 을 다시 쓰면,

$$T = \frac{1}{2} (M + \frac{2}{r^2} I_w) v_c^2 + \frac{1}{2} (I_c + \frac{D^2}{2r^2} I_w) \omega_c^2 \quad (2.9)$$

Lagrange's method 을 적용하면

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{r}_c} \right) &= (M + \frac{2}{r^2} I_w) \dot{r}_c & \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\theta}_c} \right) &= (I_c + \frac{D^2}{2r^2} I_w) \dot{\theta}_c \\ \frac{\partial T}{\partial r_c} &= 0 & \frac{\partial T}{\partial \theta_c} &= 0 \\ Q_1 &= \frac{\tau_R}{r} + \frac{\tau_L}{r} - F_R - F_L & Q_2 &= \frac{D}{2r} \tau_R - \frac{D}{2r} \tau_L - \frac{D}{2} F_R + \frac{D}{2} F_L \end{aligned}$$

결과적으로 다음과 같은 동역학식을 얻는다:

$$\begin{pmatrix} \tau_L \\ \tau_R \end{pmatrix} = \begin{pmatrix} a & -b \\ a & b \end{pmatrix} \begin{pmatrix} \dot{r}_c \\ \dot{\theta}_c \end{pmatrix} + r \begin{pmatrix} F_L \\ F_R \end{pmatrix} \quad (2.10)$$

여기서,

$$\begin{aligned} a &= \frac{5}{2} (M + \frac{2}{r^2} I_w) \\ b &= \frac{1}{2} (\frac{2D}{r} I_c + \frac{D}{r} I_w) \end{aligned}$$

위 식을 다시 쓰면, 다음 형태의 식을 쓸수 있다.

$$\tau = M \ddot{\Theta} + r F = M \dot{q}_c + r F \quad (2.11)$$

여기서,  $\Theta = (r_c, \theta_c)^T$

동역학식으로부터,  $F_R$  과  $F_L$  을 볼수 있다. 이것들은 마찰뿐만 아니라 구름 저항 (rolling resistance) 요소를 포함하고 있다. 구름저항은 바퀴와 표면사이의 상호 접촉에서의 변형에 기인한것이다. 이 구름저항은 마찰과는 약간 다르다. 이 논문에서는  $F_L$  과  $F_R$  을 마찰이 아니라 구름저항의 영향이 큰것으로 한다.

## 3 경로제어 알고리즘

이 절에서는 이동 로봇트의 주행 제어에 관해 다룬다. 제어를 하기 위해 두개의 제어를 사용한다. 하나는 속도 제어가므로 속도 명령이 주어지면 그 속도를 따라가도록 하는 것이고 다른 하나는 어느 한 목표점이 주어지면 그 목표까지 가도록 속도 명령을 주는 위치 제어이다. 이 논문에서의 목적은 속도 제어가 구성되어 있다는 가정아래 위치 제어의 설계를 다루는 것이다. 속도 제어는 적용 제어나 퍼지 제어등의 고급 알고리즘을 사용해 주어진 명령을 잘 따라간다는 가정을 한다.

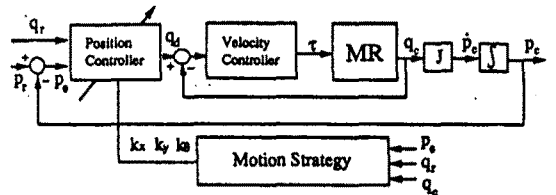


그림 3. 제어기 구조도

### 3.1 제어기 설계

$q_m$ 을 명령 속도(command velocity)라고 정의하고 이것은 위치 제어기의 출력이다.

지금 다음과 같은 가정을 한다.

A:  $q_m = q_c$   
즉 속도제어기는 이상적인 제어기다.

정리 1 (2.18)로 주어진 이동 로봇, 다음 제어법칙 (3.1) 과 가정 A를 생각하자

$$q_m = \begin{bmatrix} k_x x_{er} \cos \theta_{er} - k_y y_{er} \sin \theta_{er} + v_r \sec \theta_{er} \\ \omega_r + y_{er} v_r \sec \theta_{er} + k_\theta \sin \theta_{er} + 2k_x k_y x_{er} y_{er} \cos \theta_{er} \end{bmatrix} \quad (3.1)$$

여기서,  $k_x, k_y, k_\theta$ 들은 양의 상수이고  $v_r$ 과  $\omega_r$ 는 목표점에서의 원하는 도달 속도이다.

그때, 로봇트의 위치 오차는 위의 제어 법칙을 사용하면 원하는 목적으로 간다.

증명: 다음과 같은 Liapunov function candidate를 선택하자

$$V(t) = \frac{1}{2}(x_{er}^2 + y_{er}^2) + (1 - \cos \theta_{er}) \quad (3.2)$$

여기서,  $-\pi < \theta_{er} \leq \pi$ . 그때,(3.2)의 시간 미분은 다음과 같다.

$$\dot{V}(t) = x_{er} \dot{x}_{er} + y_{er} \dot{y}_{er} + \sin \theta_{er} \dot{\theta}_{er}$$

(2.5),  $q_m = q_c$ 과 (3.1)을 사용하면,

$$\begin{aligned} \dot{V}(t) &= x_{er} v_r - v_c x_{er} \cos \theta_{er} + v_c y_{er} \sin \theta_{er} + (\omega_r - \omega_c) \sin \theta_{er} \\ &= x_{er} v_r - (x_{er} \cos \theta_{er} - y_{er} \sin \theta_{er})(k_x x_{er} \cos \theta_{er} \\ &\quad - k_y y_{er} \sin \theta_{er} + v_r \sec \theta_{er}) + \omega_r \sin \theta_{er} - \\ &\quad \sin \theta_{er} (\omega_r + y_{er} v_r \sec \theta_{er} + k_\theta \sin \theta_{er} + 2k_x k_y x_{er} y_{er} \cos \theta_{er}) \\ &= -k_x x_{er}^2 \cos^2 \theta_{er} - k_y y_{er}^2 \sin^2 \theta_{er} - k_\theta \sin^2 \theta_{er} \leq 0 \end{aligned}$$

그때, Lasalle의 정리에 의해 오차는 점근적으로 0으로 간다. ■

위 알고리즘은 목표점과 목표점에서의 원하는 도달 속도를 주면 목표까지 도달하고 그때의 속도도 우리가 원하는 속도가 된다. 그러므로 위에 제안된 알고리즘은 궤적을 일일이 계산하여 줄 필요가 없이 단지 목표점을 줌으로써, 쉽게 원하는 목적을 이룰 수 있다. 그러나 위에 제안된 알고리즘이 비록 안정하지만,  $k_x, k_y, k_\theta$ 에 따라 이동 행태가 달라진다. 그러므로 이 값들을 잘 조합함에 따라서 우리가 원하는 주행형태를 만들어 낼 수 있다. 그래서 그림 3에서처럼 Motion strategy 부분이 주면 상황에 따라 상수들을 잘 조정하여 여러가지 주행형태를 만들어내도록 한다. 예를 들면, 움직이는 장소의 넓고 좁음에 따라, 이동 시간에 따라, 에너지 효율에 따라 원하는 주행형태는 각각 다를 것이다. 이 논문에서 사용되는 strategy는 먼저 속도를 줄여 방향각의 오차를 줄이고 어느 오차 범위내에 들어오면 속도를 올려  $x, y$ 의 오차를 줄이도록 하는 전략을 사용한다.

### 4 시뮬레이션

시뮬레이션을 위해서, 전에 구한 동역학식을 사용한다. 속도 제어기로서는 강인한 적응제어 기법을 사용하여 가정 A를 필수있는한 만족하도록 하였다. 시뮬레이션시 구름 저항이 있고 정확한 관성 모멘트를 모르고, 속도는 최대  $2m/s$ , 가속도는 최대  $5m/s^2$ 인 상황에서 시행되었다.

### 5 결론

본 연구에서는 휠 구동방식의 이동로봇에 대한 새로운 경로 제어 알고리즘을 제시하였고 몇가지 가정하에서 동역학식을 유도하였다. 유도한 동역학식은 시뮬레이션 및 강인한 적응 제어 속도 제어기 설계시 사용한다. 제안된 알고리즘은 Liapunov 함수를 사용해 안정도를 증명했으며 여러가지 주행형태를 결정할 수 있는 Motion strategy 중 한가지 경우의 전략을 제안했다.

### 참고 문헌

- [1] Yutaka Kanayama, et al, "A Stable Tracking Control Method for an Autonomous Mobile Robot," *Proc.IEEE Int. Conf. Robotics Automat.*, pp.384-389, 1990.
- [2] Y.Kanayama and S.Y.Yuta, "Vehicle Path Specification by a sequence of Straight lines," *IEEE J.of R & A.*, vol. 4, No.3, pp.265-276, 1988.
- [3] J.Crowley, "Asynchronous Control of Orientation and Displacement in a Robotic Vehicle," *Proc.IEEE.conf. Robotics and Automa.*, pp.1277-1282,1989.
- [4] Y.Kanayama and N.Miyake, "Trajectory Generation for Mobile Robots," *Robotics Research*, vol.3, pp.333-340, 1986.
- [5] Patrick F. Muir and Charles P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," *Journal of Robotics Systems*, vol.4, pp.281-340, 1987.

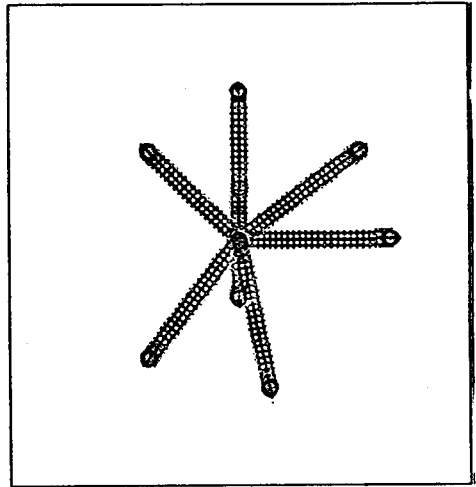


그림 4. 초기 속도가  $v=0, w=0$ 이고 목표 도달속도는 0인 경우

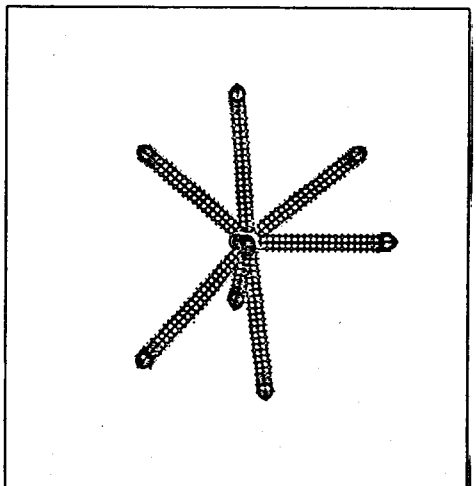


그림 5. 초기 속도가  $v=2, w=0$ 이고 목표 도달속도는 0인 경우