

휠구동방식의 자유이동로봇을 위한 조향제어방법

고경철, 조형석
한국과학기술원 생산공학과 제어및 자동화 연구실

A Steering Control Method for Wheel-driven Mobile Robot

K.C. Koh and H.S. Cho
Production Engineering Department, KAIST

ABSTRACT

This Paper proposes a steering control algorithm for non-holonomic mobile robots. The steering control algorithm is essential to navigate autonomous vehicles which employ combination of the dead reckoning and absolute sensor system such as a machine vision for detecting landmarks in order to estimate the current location of the mobile robot. The proposed algorithm is based on the minimum time BANG-BANG controller and curvature-continuity curve design method. In the BANG-BANG control scheme we introduce velocity/acceleration limiter to avoid any slippage of driving wheels. The proposed scheme is robot-independent and hence can be applied to various kinds of mobile robot or vehicles. To show the effectiveness of the proposed control algorithm, a series of computer simulations were conducted for two-wheel driven mobile robot.

1. 서론

최근 고정 궤도식 무인운반차(AGV)의 결점을 보완하고 유연성을 높이기 위해 자유주행(free-ranging)방식을 채택한 이동로봇(mobile robot)에 관한 연구가 활발하다. 이 자유주행 방식은 가야할 경로를 탑재된 메모리에 기억함으로써 경로변경이 용이하며, 충돌회피등 돌발적인 상황에 유연하게 대처할 수 있는 장점을 지니고 있다. 반면 이동로봇이 주어진 기준경로를 정확히 따라가도록 하기 위해서는, 현재 운행되고 있는 로봇의 자세(절대위치와 진행각)를 얻기위한 각종 센서시스템 및 그 처리알고리즘과, 벗어난 현재의 자세로부터 기준경로에 복귀하도록 하는 경로제어 알고리즘등을 필요로 한다. 그러나 휠구동 방식의 이동로봇의 제어는 기구학적으로 Non-holonomic Constraint를 갖고있어, 안정된 경로제어 알고리즘을 구현하는데 어려움이 있다. 이전의 연구들에서 이러한 문제점을 극복하기위한 여러가지 경로제어방법들을 제시된 바 있다. Tsumura[1]는 일련의 기준점들로 기준경로를 표현하고 이를 경로제어(path control)의 매 사이클에서 로봇의 기준점과 미래 위치를 비교하여, 다음 조향(steering)을 결정하는 방법을 제시하였고, Kanayama[2]는 기준열 대신 로봇의 이동을 위한 직선(straight line)기준경로를 사용하는 방법을 제시하였다. Crowley등[3]은 3개의 계층구조를 갖는 이동제어시스템(locomotion control system)을 개발하였는데, 여기서 가상의 이동체(virtual vehicle) 개념을 도입하였다. 이방법은 직진과 선회운동의 독립적인 제어가 가능하며, 부드러운 궤적을 얻을 수 있는 특징이 있다. Singh[4]은 경로궤적오차를 보상하기 위해, 현재점과 미래기준점을 부드러운 곡선으로 연결하는 Quintic polynomial방법을 사용한 바 있다. Kanayama[5]는 기준점과 현재점의 자세(posture)로부터, 국부오차(local error)좌표계를 설정하여, PID제어의 형태의 직진과 회전속도를 위한 제어율을 제시한 바 있다. 또한 최근 Kanayama[6]는 비선형형태를 갖는 조향(steering)및 선속도 제어율을 발표하였는데, 그 안정성을 Liapunov 함수를 이용하여 증명하였다. 본 논문에서는 이러한 이동로봇이 가지고 있는 기하학적한계를

동적한계를 극복하는 경로제어알고리즘의 새로운 형태로써 좀더 유연하고 안정된 경로제어방법을 제시한다. 이방법에서는 현재 운행중인 이동로봇의 정보를 현재위치벡터(X_c, Y_c, θ_c)^T와 현재속도벡터(V_c, ω_c)로 표현하고, 기준경로상의 목표지점(target point)에 관한 정보를 기준위치벡터(X_r, Y_r, θ_r)와 기준속도벡터(V_r, ω_r)로 표현하여, 이들을 이용하여 이동로봇이 기준경로점에 복귀할 수 있도록 다음 사이클의 속도명령(V_d, ω_d)을 계산하는 구조를 갖는다. 제안된 알고리즘의 효과를 알아보기 위해 Two wheel-driven방식의 이동로봇에 대해 일련의 컴퓨터시뮬레이션을 수행하였으며, 만족스러운 경로제어 성능을 얻을 수 있었다.

2. 문제점 제기

문제점설명에 앞서 몇가지 사전 정의를 한다.

2.1 경로표현과 기구학식

그림.1과 같은 2개의 구동휠의 속도차로써 직진과 선회를 할 수 있는 Two wheel-driven 이동로봇을 고려한다. 그리고 이 로봇은 2차원 평면상에서 주행한다고 가정하며, 이 평면상에서 정의된 고정된 좌표계(또는 절대좌표계)에서 이동로봇의 위치는 이동로봇의 중심점(C_R)의 (X, Y)좌표값 및 이동로봇의 길이방향의 축과 X축 사이의 각(일명 heading angle)인 θ 로 표현된다. 이를 벡터로 표현한 것을 자세(posture)라 하며 이를 P로 정의한다.

$$P=(X, Y, \theta)^T \quad (1)$$

좌우구동휠의 회전속도를 각각 (ω_R, ω_L)로 정의할 때, 이로부터 로봇몸체 중심의 선속도 및 회전속도 (V_c, ω_c)는 아래식으로 결정된다.

$$V_c = R_w(\omega_R + \omega_L)/2 \quad (2.a)$$

$$\omega_c = R_w(\omega_R - \omega_L)/D_w \quad (2.b)$$

여기서 R_w 는 구동휠의 반경이고, D_w 는 양구동휠간의 거리이다. 그러면 이러한 heading각(θ_c)방향으로의 선속도(V_c)와 vehicle의 회전중심(X_c, Y_c)에서 회전속도(ω_c)로부터 로봇의 자세(P_c)는 다음식들의 시간에 대한 적분으로 구해진다.

$$dX_c/dt = V_c \cos(\theta_c) \quad (3.a)$$

$$dY_c/dt = V_c \sin(\theta_c) \quad (3.b)$$

$$d\theta_c/dt = \omega_c \quad (3.c)$$

이 때 식.3은 비선형이고 그 적분해가 존재하지 않는다. 이를 "Non-holonomic problem"이라 하며, 이로 인해 경로궤적이나 제어에 제약과 어려움이 따른다.

2.2 경로오차의 정의

기준경로의 자세를 P_r 로 정의 하고, 이동로봇의 현재의 자세를 P_c 로 정의할 때, 이 둘간의 차이를 자세에러(P_e)라 정의한다. 이 P_e 는 기준좌표계를 어디에 설정하는가에 따라 다음의 2가지로 표현될 수 있다.

(i) 경로기준자세에러(P_{er})

이는 그림.2와 같이 기준좌표계를 기준경로의 자세(P_r)에서 설정하여 정의되는 것으로, 이 좌표계를 기준으로 본 자세에러를 경로기준자세에러(P_{er})라 하며, 다음식으로 구해진다.

$$P_{er} = (X_{er}, Y_{er}, \theta_{er})^T = T_{er} (P_r - P_c) \quad (4)$$

여기서 $T_{er} = \begin{vmatrix} \cos(\theta_r) & \sin(\theta_r) & 0 \\ -\sin(\theta_r) & \cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{vmatrix}$

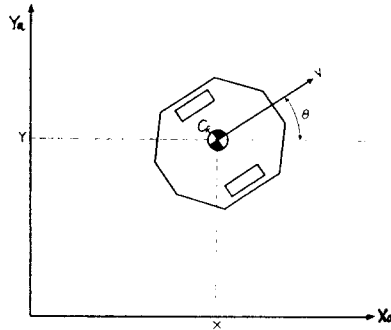


그림 1 이동로봇의 자세
Fig.1 Definition of posture

(ii) 로봇기준자세에러(P_{ec})

이는 그림.2와 같이 기준좌표계를 로봇의 현재자세(P_c)에서 설정하여 정의되는 것으로 이 좌표계에서 기준으로 본 자세에러를 로봇기준 자세에러(P_{ec})라 하며, 다음식으로 구해진다.

$$P_{ec} = (X_{ec}, Y_{ec}, \theta_{ec})^T = T_{ec} (P_r - P_c) \quad (5)$$

여기서 $T_{ec} = \begin{vmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{vmatrix}$

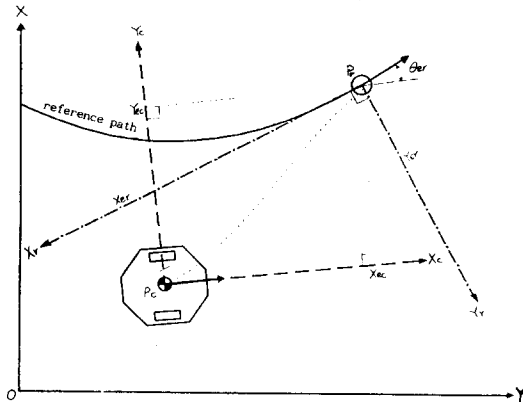


그림 2 상대좌표계의 정의
Fig.2 Relative coordinates for mobile robot

2.3 Dead reckoning

"Dead reckoning"이란 휠에 부착된 상대센서(엔코더 및 타코미터)의 정보를 이용하여, 이동로봇의 자세(posture)를 실시간으로 추정하는 것을 말한다. 이는 이론적으로는 식.2를 이용하여 회전속도(ω_c)와 선속도(V_c)를 구한후, 기구학식 식.3을 시간에 따라 적분하여 수행될 수 있다.

$$X_c = \int V_c \cos(\theta_c) dt \quad (6.a)$$

$$Y_c = \int V_c \sin(\theta_c) dt \quad (6.b)$$

$$\theta_c = \int \omega_c dt \quad (6.c)$$

그러나 실제로는 정확한 휠의 속도(ω_R, ω_L)를 측정하는데 한계가 있으며, 식.6의 적분해를 구할 수없으므로 매 샘플마다 근사적으로 계산한다. 이 과정을 설명하면 다음과 같다. 먼저 좌우 구동휠에서 측정된 회전각의 변화량($\delta\theta_L, \delta\theta_R$)으로부터 경로증분(δS_c)과 경로에 대한 접선방향의 각도변화량($\delta\theta_c$)을 아래식을 사용하여 계산한다.

$$\delta S_c = R_w(\delta\theta_R + \delta\theta_L)/2 \quad (7.a)$$

$$\delta\theta_c = R_w(\delta\theta_R - \delta\theta_L)/D_w \quad (7.b)$$

그리고 아래의 결과식을 이용하여 현재자세(P_c)를 계산한다.

($\delta\theta_c \neq 0$ 일때): 원호운동

$$\theta_c = \theta_c' + \delta\theta_c \quad (8.a)$$

$$X_c = X_c' + \delta S_c / \delta\theta_c [\sin(\theta_c) - \sin(\theta_c')] \quad (8.b)$$

$$Y_c = Y_c' - \delta S_c / \delta\theta_c [\cos(\theta_c) - \cos(\theta_c')] \quad (8.c)$$

($\delta\theta_c = 0$ 일때): 직선운동

$$\theta_c = \theta_c' \quad (9.a)$$

$$X_c = X_c' + \delta S_c \cos(\theta_c') \quad (9.b)$$

$$Y_c = Y_c' + \delta S_c \sin(\theta_c') \quad (9.c)$$

3. 경로제어 알고리즘

이동로봇이 주어진경로를 따라갈때, 이 경로를 벗어나지 않도록 로봇의 선속도 명령 V_d 및 회전속도 명령 ω_d 을 때 제어 순간마다 새로 정해 주는 것을 경로제어라 한다. 본연구에서는 기준경로의 자세(P_r) 및 속도(V_r, ω_r)에 대해 이동로봇의 현재자세(P_c) 및 속도(V_c, ω_c)를 귀환하여 수행하는 경로제어 알고리즘을 소개한다.

3.1 문제점 제기

이동로봇의 부드러운 가속속이나 유연한 회전은 미끄러짐을 피하는 중요한 인자이다. 따라서 경로계획에 있어 좀더 부드러운 경로를 만들기 위해 곡률연속곡선(예: clothoid, spiral curve 등)에 관한 연구가 있어왔다[7-8]. 그러나 대부분 이러한 곡선에 의한 경로설계는 매우 복잡하거나 곡선의 explicit한 해가 존재하지 않는 단점이 있다. 이 문제를 해결하기 위해 곡률연속적인 움직임보장하는 경로제어방법을 제안한다. 따라서 단순한 직선/원호의 조합만으로 경로계획이 가능하도록 하였다. 이방법은 최소시간 제어기의 설계개념과 곡률연속커브 설계기법에 바탕을 둔다.

3.2 최소시간 BANG-BANG제어

먼저 경로제어알고리즘 설명에 앞서 최소시간 BANG-BANG제어에 대해 알아본다. 어떤 이동체가 직진만 한다고 가정하고, 그 속도(V_c) 및 가속도(A_c)가 아래와 같이 제한된다고 가정한다.

$$\begin{vmatrix} V \\ A \end{vmatrix} \leq \begin{vmatrix} V_{max} \\ A_{max} \end{vmatrix} \quad (10.a)$$

$$\begin{vmatrix} V \\ A \end{vmatrix} \leq \begin{vmatrix} V_{max} \\ A_{max} \end{vmatrix} \quad (10.b)$$

여기서 V_{max} 는 최대허용속도이고 A_{max} 는 최대허용가속도이다. 그리고 이동체의 제어입력을 가속도 A_c 로 할때, 식.10의 제한 조건을 만족하면서 최소시간내에 현재의 위치및 속도(X_c, V_c)부터 목표점의 위치및 속도(X_r, V_r)에 도달하기 위한 가속도 A_c 를 구하는 방법을 고려한다. 먼저 이동체의 목표점의 위치(X_r)및 속도(V_r)에 대해, 현재의 위치 X_c 에서 안전속도 V_s' 를 아래의 식으로 정의한다.

$$V_s' = V_r + [2A_{max}|X_r - X_c|]^{1/2} \text{SGN}(X_r - X_c) \quad (11)$$

그리고 최대허용 속도조건 식(3.2.1)에 의해 그 크기는 아래와 같이 제한된다.

$$V_s = \mu(V_s', V_{max}) \quad (12)$$

여기서 $\mu(a, b)$ 는 일종의 Clamping함수로 다음과 같이 정의된다.

$$(1) \begin{cases} a \geq b & \text{일때 } \mu(a, b) = b \text{ SGN}(a) \\ a < b & \text{일때 } \mu(a, b) = a \end{cases}$$

일단 V_s 가 구해지면, 샘플링주기를 δT 라 할 때, 가속도(A_c)는 다음과 같이 결정할 수 있다.

$$A_c = \mu[(V_s - V_c)/\delta T, A_{max}] \quad (13)$$

이는 수정된 BANG-BANG제어알고리즘으로 볼수 있으며, 즉 기존의 BANG-BANG제어가 스위칭조건에 의해 최대가속도의 부호만을 결정하는 것에 비해 본 방법은 식.11과 같이 V_s 의 개념을 도입하여 식.13과 같이 선형화 시킴으로써 급격한 가속도의 변화를 줄이도록 하였다. 만약 이동체가 속도제어되는 시스템이라면 이때의 제어입력은 다음 주기에서의 원하는 새로운 속도 V_d 가 되며 이경우 아래식으로 구할 수 있다.

$$V_d = V_c + A_c \delta T \quad (14)$$

3.3 선속도 제어알고리즘

위의 최소시간제어를 이동로봇의 선속도제어 적용한다. 경로 기준 상대오차가 주어졌을 때 이를 로봇의 현재진행 방향선과 기준 경로자세(P_r)에서 잡은 좌표계의 까지의 거리및 경로점의 기준속도는 다음과 같다.

$$X_{ec} = X_{er} / \text{SIN}(\theta_{er}) \quad (15.a)$$

$$V_{rc} = V_r / \text{SIN}(\theta_{er}) \quad (15.b)$$

이와 같이 진행방향에 대한 목표점까지의 거리(X_{ec})및 목표점의 기준속도(V_{rc})가 구해지면, 앞절에서 유도한 최소시간 제어방법을 적용하여 아래와 같이 선속도명령(V_d)을 구할 수 있다.

$$V_s' = V_{rc} + [2A_{max}|X_{ec}|]^{1/2} \cdot \text{SGN}(X_{ec}) \quad (16.a)$$

$$V_s = \mu(V_s', V_{max}) \quad (16.b)$$

$$dV_c/dt = \mu[(\omega_s - \omega_c)/\delta T, A_{max}] \quad (16.c)$$

$$V_d = V_c + [dV_c/dt]\delta T \quad (16.d)$$

3.4 회전속도제어

본절에서는 직선 경로에 대해서는 X-Y좌표계에서, 원호경로에 대해서는 R- ϕ 좌표계에서 유도되는 회전속도(또는 조향)제어 알고리즘을 제시한다. 기본개념은 경로계획에서 사용하는 곡률 연속곡선의 설계기법을 온라인화하는 데서 출발하였으며, 이에 최소시간 제어방법을 도입하였다.

3.4.1 직선경로에 대한 조향제어

만약 ω_r 이 0인 경우 그때의 기준경로는 직선(straight line)이라고 볼수있으며, 이때 이동로봇이 자세에러 P_{er} 를 갖고 이 직선경로에 복귀하기 위한 이동로봇의 회전속도 ω_d 를 구하는 문제를 생각한다. 이를 단계적으로 설명하면 다음과 같다.

(1) 곡률연속 커브의 설계

먼저 이동로봇의 중심을 통과하며 직선기준경로에 접하는 가상의 곡률연속 커브를 가정한다. 이 커브의 좌표계로서 그 접하는 점을 원점으로 하고 직선기준경로의 진행방향을 X축으로 하는 X-Y 좌표계를 설정한다. 그리고 그 곡률연속커브 $Y_L(X)$ 의 한 형태로서 X에 대한 n차 polynomials로 가정할 때, 아래의 조건들을 만족하도록 하여 그 해를 구한다.

$$i) Y_L(0) = 0 \quad (17.a)$$

$$ii) dY_L(0)/dX = 0 \quad (17.b)$$

$$iii) d^2Y_L(0)/dX^2 = 0 \quad (17.c)$$

$$iv) d^3Y_L(0)/dX^3 = C_x \quad (17.d)$$

$$v) |x(X)| \leq X_{max} \quad (17.e)$$

여기서 $x(X)$ 는 곡률함수로 X-Y좌표상의 커브의 곡률은 아래의 관계식을 갖는다.

$$x(X) = (d^2Y/dX^2)/[1+(dY/dX)^2]^{3/2} \quad (18)$$

여기서 i)-iii)은 곡선이 곡률의 연속성을 만족하면서 직선경로에 접하는 조건이고, vi)는 곡선의 C_x 를 정하기 위한 조건으로 이 곡선의 최대곡률의 제한조건이다. 따라서 위의 조건들을 만족하는 커브식은 아래와 같다.

$$Y_L(X) = C_x X^5 \quad (19)$$

편의상 이렇게 설계된커브를 랜딩커브라 부른다.

(2) 랜딩각도의 계산

만약 이동로봇이 이 직선경로의 수직방향으로 Y_{er} 만큼 떨어진 거리에 위치할 때, 그 지점을 통과하는 랜딩커브의 접선과 로봇의 진행방향과 이루는 각 θ_L 을 아래와 같이 구할 수 있다.

$$\theta_L = \text{TAN}^{-1}[S_L] + \theta_r \quad (20.a)$$

여기서

$$S_L = 3C_x(Y_{er}/C_x)^{2/3} \cdot \text{SGN}(Y_{er}). \quad (20.b)$$

(3) 랜딩각속도(ω_L)의 계산

이 랜딩각 θ_L 은 이동로봇이 주어진 직선경로에 대해 Y_{er} 의 수직경로오차와 θ_{er} 의 방향오차를 가지고 선속도 V_c 의 속도로 움직이고 있을 때 시간에 따라 변하게 된다. 이 랜딩각도의 변화율을 랜딩각속도(ω_L)라 정의 하며, 이는 식.20을 시간에 대해 미분하여 구할수 있다.

$$\omega_L = 2C_x(|Y_{er}|/C_x)^{-1/3} Y_{er}' / (1+S_L^2) \cdot \text{SGN}(Y_{er}) \quad (21.a)$$

여기서 Y_{er}' 은 이동로봇이 기준경로에 수직방향으로 접근하는 속도성분으로 아래와 같다.

$$Y_{er}' = dY_{er}/dt = V_c \text{SIN}(\theta_{er}) \quad (21.b)$$

(4) 회전속도명령(ω_d)의 계산

지금까지 구한 랜딩각(θ_L)과 그각속도(ω_L)는 매순간 이동로봇이 이 랜딩커브를 추종하기위해 취해야할 방향및 회전속도가 된다. 이때 이동로봇에 대해 최대허용 각속도(ω_{max})와 최대 허용 각가속도(a_{max})가 주어지면, 아래와 같이 최소시간

어방법을 적용하여, 최종적으로 회전속도명령(ω_d)을 구할 수 있다.

$$\begin{aligned} \omega_s' &= \omega_L + [2a_{max}|\theta_L - \theta_c|]^{1/2} \text{SGN}(\theta_L - \theta_c) & (22. a) \\ \omega_s &= \mu(\omega_s', \omega_{max}) & (22. b) \\ d\omega_c/dt &= \mu[(\omega_s - \omega_c)/\delta T, a_{max}] & (22. c) \\ \omega_d &= \omega_c + [d\omega_c/dt]\delta T & (22. d) \end{aligned}$$

3.4.2 원호경로에 대한 조향제어 알고리즘

이번에는 기준경로가 원호(circular arc)인 경우 즉 ω_r 이 0이 아닌 경우에, 경로기준 상대오차(P_{or})를 갖는 이동로봇이 이 원호경로에 복귀하는 경우를 생각해 본다. 이 경우 앞절의 직선 경로에 대한 알고리즘을 그대로 적용해도 부드러운 경로접근이 가능하지만 기준경로의 방향(θ_r)이 시간에 따라 변하므로 정상 상태 오차가 존재하게 된다. 이를 해결하기 위해서는 식.17의 조건 iii)에서 원호의 곡률을 고려하여 0이 아닌 값으로 변경해야 한다. 그러나 이는 3차방정식의 해를 구하는 문제가 되어 알고리즘의 유도가 매우 복잡해진다. 그러나 좌표계를 원호의 중심을 기준으로 한 R- ϕ 좌표계로 하면, 이문제를 해결할 수 있다. 이에대한 유도과정은 앞의 직선경로의 경우와 같은과정을 수행하여 유도되므로, 본논문에서는 지면의 한계로 그결과식만을 제시한다.

(1) landing curve의 설계

원호경로에 대한 랜딩커브식 $R_L(\phi)$ 는 다음식으로 표현된다.

$$R_L(\phi) = R_r + C_r\phi^3 \quad (23)$$

여기서 C_r 는 최대곡률제한 조건으로 결정되는 랜딩계수로 참고로 R- ϕ 좌표계에서의 곡률공식은 아래와 같다.

$$\kappa(\phi) = (R^2 + 2R'R'' - RR''') / (R^2 + R'^2)^{3/2} \quad (24)$$

$$\begin{aligned} \text{여기서 } R &= R(\phi) \\ R' &= dR(\phi)/d\phi \\ R'' &= d^2R(\phi)/d\phi^2. \end{aligned}$$

(2) 랜딩각도의 계산

이때의 랜딩각 θ_L 은 다음과 같은 결과식을 이용하여 구해진다.

$$\theta_L = \text{TAN}^{-1}[S_L] + \text{TAN}^{-1}[X_{or}/(R_r - Y_{or})] + \theta_r \quad (25. a)$$

$$\begin{aligned} \text{여기서 } S_L &= 3C_r\phi^2/R_c \cdot \text{SGN}(\Delta R) & (25. b) \\ \phi &= (|\Delta R|/C_r)^{1/3} & (25. c) \\ \Delta R &= R_r - R_c & (25. d) \\ R_c &= [X_{or}^2 + (R_r - Y_{or})^2]^{1/2} & (25. e) \\ R_r &= V_r / \omega_r & (25. f) \end{aligned}$$

(3) 랜딩각속도(ω_L)의 계산

랜딩각속도(ω_L)는 식.25를 시간 t에 대해 미분을 하여 구한 아래의 결과식을 이용하여 계산된다.

$$\omega_L = R_c' (2/\phi + S_L) / (1 + S_L^2) / R_c + \beta' + \omega_r \quad (26. a)$$

$$\begin{aligned} \text{여기서 } R_c' &= [X_{or}X_{or}' - (R_r - Y_{or})Y_{or}'] / R_c & (26. b) \\ X_{or}' &= Y_{or}\omega_r - V_r + V_c \text{COS}(\theta_{or}) & (26. c) \end{aligned}$$

$$Y_{or}' = -X_{or}\omega_r + V_c \text{SIN}(\theta_{or}) \quad (26. d)$$

$$\beta' = [(R_r - Y_{or})X_{or}' + X_{or}Y_{or}'] / R_c^2 \quad (26. e)$$

(4) 회전속도명령(ω_d)의 계산

따라서 원호경로에 대해 랜딩각(θ_L)과 그각속도(ω_L)가 매순간 계산되면, 직선경로의 경우와 같이 식(22)의 최소시간 제어알고리즘을 적용하여, 회전속도명령(ω_d)을 결정할 수 있다.

3.5 랜딩계수의 선정

본절에서는 앞절에서 설명한 랜딩커브의 계수(직선경로시 C_x , 원호경로시 C_r) 선정방법을 다루기로한다. 그림 3(a)는 직선 경로에 대해 랜딩계수(C_x)를 0.05, 0.1, 0.15, 0.2로 달리 하였을 때, 랜딩곡선의 변화를 보여준다. 그림 3(b)는 각각의 랜딩곡선들에 대한 곡률(κ)의 변화를 곡선의 길이방향에 대해 그려본 결과이다. 그림에서 보는 바와 같이 C_x 값을 크게할수록 최대곡률(κ_m)도 증가함을 알 수 있다. 그림 4(a)는 원호경로(반경 $R_r = 5m$)에 대해 랜딩계수(C_r)를 0.4, 0.8, 0.12, 0.16, 0.2로 변화 시켰을 때의 랜딩커브들의 결과이며, 이에 대응하는 곡률곡선을 그림 4(b)에 도시하였다. 그림 4(b)에서 보는바와 같이 곡률곡선이 직선의 경우와는 달리 (+)방향의 랜딩커브와 (-)방향의 랜딩커브가 서로 비대칭인 것을 알 수있다.

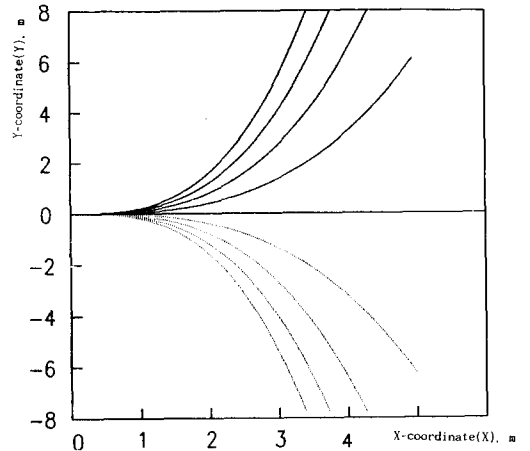


그림 3(a) C_x 변화에 따른 직선경로에 대한 랜딩커브

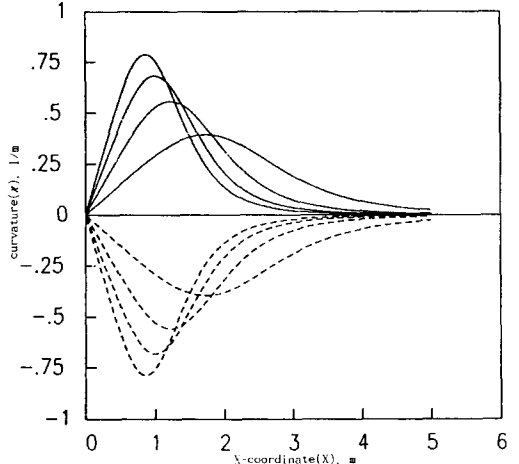


그림 3(b) C_x 변화에 따른 랜딩커브의 곡률 곡선

이 경우 원의 안쪽의 랜딩커브(-C_r)만이 곡률의 피크치를 갖고 있음을 알 수 있다. 이러한 랜딩커브의 최대곡률은 이동로봇의 회전운동시 원심력에 직접 영향을 미치며, 그 관계식은 다음과 같다.

$$F_c = M_r J_c V_r^2 \quad (27)$$

따라서 만약 미끄러짐을 방지하는 조건이 $|F_c| < F_{cmax}$ 라면, 최대곡률(κ_m)의 제한조건은 아래와 같이 구할 수 있다.

$$(\kappa_m)_{max} = F_{cmax}^2 / J_c V_c^2 \quad (28)$$

따라서 랜딩계수 C_x, C_r은 이러한 곡률의 제한조건을 고려하여 적절히 선택될 수 있다.

4. 시뮬레이션 결과 및 토의

제안된 경로제어 알고리즘의 효과를 알아보기 위해 일련의 컴퓨터 시뮬레이션을 수행하였다. 그림 5는 이동로봇이 직선 경로에 대해 Y_r 방향의 초기 위치오차가 있을 때 경로 추종결과를 도시한다. 이때 조건은 아래와 같이 하였다.

- i) 경로기준속도 : $V_r = 1, \omega_r = 0$
- ii) 초기경로오차 : $X_{or} = 0, Y_{or} = 3m, \theta_{or} = 0$
- iii) 이동로봇의 초기속도 : $V_c = 0, \omega_c = 0$

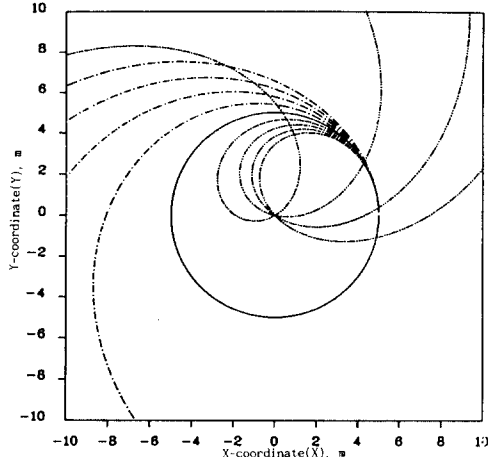


그림 4(a) C_r 변화에 따른 원호경로에 대한 랜딩커브

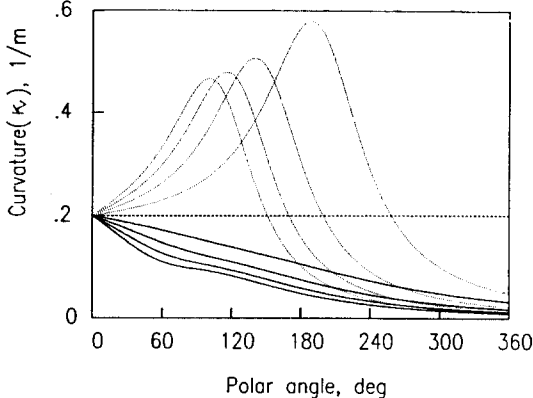


그림 4(b) C_r 변화에 따른 랜딩커브의 곡률 곡선

그림 5(a)는 제안된 종래의 단순비례 경로제어 알고리즘을 적용하여 보았을 때의 결과이다. 이때 이동로봇의 운동궤적을 절대 좌표계에서 그렸으며 실선은 기준경로를 의미하고 점선은 시뮬레이션된 로봇의 운동궤적을 의미한다. 이때 비례제어기의 계인들을 다음과 같이 4가지 경우로 변화시켜가며 시뮬레이션을 수행하였다.

- (case 1) $K_x = 0.3, K_y = 0.4, K_\theta = 0.4$
- (case 2) $K_x = 0.3, K_y = 0.8, K_\theta = 0.8$
- (case 3) $K_x = 0.3, K_y = 1.2, K_\theta = 1.2$
- (case 4) $K_x = 0.3, K_y = 1.6, K_\theta = 1.6$

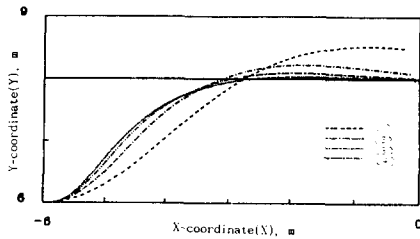
그림에서 보는바와같이 K_y와 K_θ가 작을 때 작은 곡률의 운동궤적을 보이나, 경로수렴도가 나빠지며 오버슈트현상이 있음을 알 수 있고, 그 값들이 커질수록 로봇의 운동궤적의 곡률도 커지면서 기준경로에 빨리 수렴함을 볼 수 있다. 그림 5(b)는 제안된 알고리즘을 적용하였을 때의 결과이며 랜딩계수 C_x를 0.05, 0.1, 0.15, 0.2등으로 변화시켜 보았다. 그림 5(b)에서 보는바와같이 랜딩계수(C_x)를 작게하면 운동궤적이 작은 곡률을 갖고 느리게 기준경로에 접근하고, 그 값을 크게 할수록 기 그 곡률이 커지면서 기준경로에 빨리 수렴함을 알 수 있다. 이때 특징적인 것은 C_x값에 무관하게 오버슈트를 피하면서 부드러운 운동을 얻을 수 있다는 점이다. 이는 앞절에서 언급한 바와같이 이동로봇의 회전시 선속도에 따라 최대허용곡률을 제한할 필요가 있으므로 중요한 의미를 갖는다. 즉 제안된 알고리즘의 하나의 장점으로써, C_x의 선택에 따라 이동로봇이 갖는 운동궤적의 최대곡률을 임의로 정할 수 있고, 오버슈트나 급격한 곡률변화없이 부드럽게 기준경로에 복귀할 수 있는 것이다. 그림 6은 반경 5m의 원호경로에 대해 경로제어를 수행한 결과이다. 마찬가지로 그림 6(a)는 종래의 비례제어방법을 적용하여 본 결과이고, 이때의 비례제어계인은 직선경로에 대한 (case 1), (case 2) 그리고 (case 3)의 3가지 경우로 하였다. 그림 6(b)는 제안된 경로제어 방법을 랜딩계수 C_r을 0.8, 1.2, 1.6등으로 변화시켜가면서 적용한 결과를 보여준다. 그림으로부터 알 수 있듯이 원호경로에 대한 제어결과도 제안된방법이 부드럽고 개선된 경로추종결과를 보임을 알 수 있다.

5. 결 론

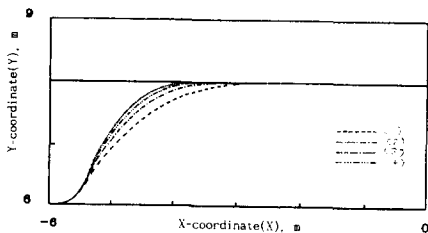
본 연구에서는 휠구동방식의 이동로봇에 대한 새로운 경로제어 알고리즘을 제시하였다. 제안된 알고리즘은 부드러운 운동궤적을 얻을 수 있도록 한 것이 특징이며, 특히 미끄러짐을 피할 수 있도록 이동로봇이 진행방향에 대해서는 선속도의 변화율, 즉 가속도에 제한을 주고, 이 제한조건을 만족하는 최소시간제어(BANG-BANG Control)를 채택한 선속도 제어방법을 제안하였고, 최대곡률 제한조건으로부터 설계한 랜딩커브를 추종하는 조향제어기를 제시함으로써, 원심력에 의한 미끄러짐을 피하면서 곡률연속적인 운동을 얻을 수 있도록 하였다. 따라서 제안된 경로제어방법은 경로계획시 단순한 직선과 원호의 조합으로 설계된 경로에 대해서도 곡률 불연속 구간을 부드럽게 연결하는 실시간 보간기능을 제공한다. 또한 제안된 방법은 Dead reckoning에 의한 경로제어뿐 아니라 Vision system 등과 같은 절대센서방식에 의한 경로수정 알고리즘으로 적용될 수 있다.

REFERENCES

- [1] T. Tsumura, et al., "An Experimental System for AGV, Following the Route Stored in Memory", *Proc. 11th I.S.R.R.*, 1981
- [2] Y. Kanayama and S.Y. Yuta, "Vehicle Path Specification by a Sequence of Straight Lines", *IEEE J. of R & A*, Vol. 4, No3, 265-276, 1988
- [3] J. Crowley, "Asynchronous Control of Orientation and Displacement in a Robotic Vehicle", *Proc. IEEE Conf. on R & A*, 1989
- [4] S. Singh and D.H. Shin, "Position Based Path Tracking for Wheeled Mobile Robots", *Proc. IEEE IROS '89*
- [5] Y. Kanayama, et al., "A Locomotion Control Method for Autonomous Vehicles", *IEEE Int. Conf. on R & A*, 1989
- [6] Y. Kanayama, et al., "A Stable Tracking Control Method for an Autonomous Mobile Robot", *Proc. IEEE Conf. on R & A*, 1990
- [7] Y. Kanayama and B.I. Hartman, "Smooth Local Path Planning for Autonomous Vehicles", *IEEE Int. Conf. on R & A*, 1989
- [8] W. Nelson, "Continuous-Curvature Path for Autonomous Vehicles", *IEEE Int. Conf. on R & A*, 1989

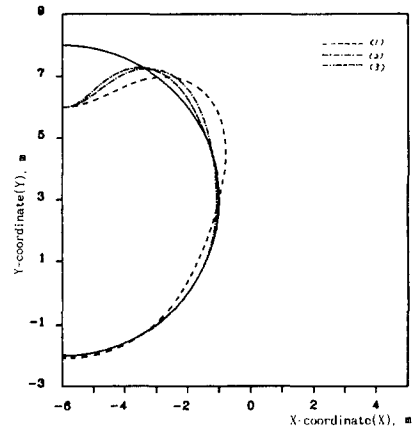


(a) 중력의 비례제어 알고리즘의 적용 결과

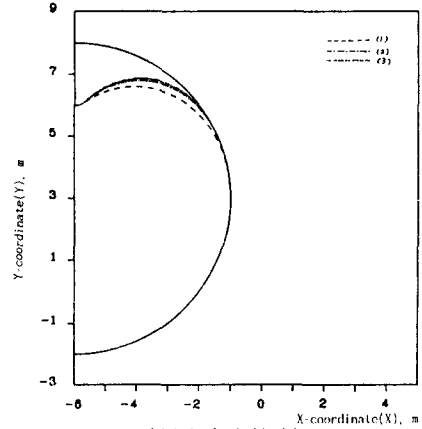


(b) 제한된 제어 알고리즘의 적용 결과

그림 5 직선경로에 대한 경로제어 시뮬레이션 결과



(a) 비례제어 알고리즘의 적용 결과



(b) 제한된 알고리즘의 적용 결과

그림 6 원호경로에 대한 경로제어 시뮬레이션 결과