

이동로봇의 바퀴 속도 제한을 고려한 최대 속도궤적 생성 방법

Maximum Velocity Trajectory Planning for Mobile Robots Considering Wheel Velocity Limit

양길진, 최병욱*
(Gil Jin Yang¹ and Byoung Wook Choi^{2,*})

¹Graduate School in Dept. of Electrical Engineering, Seoul National University of Science and Technology

²Dept. of Electrical Engineering, Seoul National University of Science and Technology

Abstract: This paper presents a maximum velocity trajectory planning algorithm for differential mobile robots with wheel velocity constraint to cope with physical limits in the joint space for two-wheeled mobile robots (TMR). In previous research, the convolution operator was able to generate a central velocity that deals with the physical constraints of a mobile robot while considering the heading angles along a smooth curve in terms of time-dependent parameter. However, the velocity could not track the predefined path. An algorithm is proposed to compensate an error that occurs between the actual and driven distance by the velocity of the center of a TMR within a sampling time. The velocity commands in Cartesian space are also converted to actuator commands to drive two wheels. In the case that the actuator commands exceed the maximum velocity the trajectory is redeveloped with the compensated center velocity. The new center velocity is obtained according to the curvature of the path to provide a maximum allowable velocity meaning a time-optimal trajectory. The effectiveness of the algorithm is shown through numerical examples.

Keywords: trajectory planning, time-optimal trajectory, convolution operator, wheel velocity limit, mobile robots

I. 서론

두 바퀴 이동로봇은 작업공간 내에서 목표 위치까지 이동하여 주어진 작업을 수행할 수 있는 로봇으로 현재 지능형 서비스로봇, 로봇청소기 그리고 운반로봇 등으로 활용되고 있다. 이러한 이동로봇의 주행을 위해서 경로 계획 방법과 궤적 계획 방법, 궤적을 추종하는 추종 제어기 및 구동 제어기의 연구와 함께 최적의 궤적을 생성하기 위한 연구가 진행되고 있다[2-8]. 궤적 계획은 미리 알려진 작업공간의 상황 정보를 이용하여 계획하는 오프라인 방식과 실시간으로 주변 상황 정보를 인지하여 주행하는 온라인 방식이 있다. 오프라인 방식의 궤적 계획에서는 불확실한 상황 정보에 대한 적응성이 없으며, 구동기의 동적인 특성을 고려하지 않을 경우 구동기에서 실제로 추종하기 어려운 궤적이 생성될 수 있다[1]. 반면에 미리 알려진 작업공간내의 상황 정보를 이용하여 최적의 궤적을 생성할 수 있으며, 시작 점부터 목표 위치까지 이동로봇의 주행을 미리 계획할 수 있다는 장점이 있다[7].

최근 궤적 계획에서 로봇의 동적인 특성을 고려하기 위한 연구가 진행되고 있으며, 로봇 구동기의 물리적 제한을 고려하기 위한 방법으로 컨볼루션 (Convolution) 연산자의 특성을 이용한 궤적 생성방법이 제안되었다[1]. 하지만 이를 두 바퀴

이동로봇에 적용하여 곡선의 경로를 주행할 경우 두 구동기의 물리적 제한을 만족하지 못한다. 이러한 문제점을 해결하기 위한 방법으로 곡선경로에 대하여 이동로봇의 물리적 제한을 고려한 궤적 계획 방법이 선행 연구되었다[4]. 그러나 최소 주행 시간을 이루기 위한 두 바퀴의 최대 허용 속도를 구현하는 방법은 연구되지 않았다.

본 논문에서는 두 바퀴 이동로봇의 궤적 생성과정에서 이동로봇의 두 바퀴의 물리적인 제한을 고려하여, 로봇의 구동기가 추종할 수 있는 궤적 생성 방법을 연구한다. 이 때, 이동로봇의 각 구동기의 최대 허용 속도를 고려하여 최소시간 주행이 가능한 궤적을 생성한다.

II. 경로 및 궤적계획

1. 베지어 곡선기반 경로계획

로봇이 목표 위치까지 이동하여 주어진 작업을 수행하기 위해서는 미리 알려진 작업 공간 내의 환경 정보와 함께 로봇의 상태 정보를 이용한 경로 계획 방법이 필요하다. 로봇의 상태 정보 중 하나인 로봇의 자세를 고려한 경로 계획 방법은 시작점과 목표위치에서의 로봇이 바라보고 있는 방향각을 고려하여 궤적 추종제어기에서 추종 가능한 경로를 생성할 수 있다. 이를 위해 본 연구에서는 시작점과 목표점에서의 방향각을 고려할 수 있으며, 연속적인 베지어 곡선 (Bezier curve)기반 경로 생성방법을 이용한다. 그림 1은 3차 베지어 곡선으로 생성된 경로이며, 다음 식 (1)은 베지어 곡선의 방정식을 나타낸다.

$$\begin{aligned}x(u) &= x_s(1-u)^3 + 3x_1u(1-u)^2 + 3x_2u^2(1-u) + x_e u^3 \\y(u) &= y_s(1-u)^3 + 3y_1u(1-u)^2 + 3y_2u^2(1-u) + y_e u^3\end{aligned}\quad (1)$$

* Corresponding Author

Manuscript received December 9, 2014 / revised January 5, 2015 / accepted January 30, 2015

양길진: 서울과학기술대학교 일반대학원 전기공학과
(gijyang@seoultech.ac.kr)

최병욱: 서울과학기술대학교 전기정보공학과
(bwchoi@seoultech.ac.kr)

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(NRF-2012R1A1A2006057).

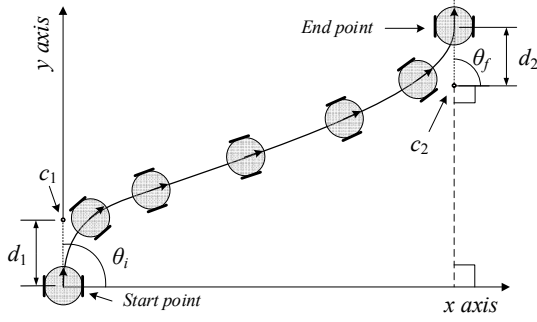


그림 1. 베지어 곡선 기반 S-곡선 경로.

Fig. 1. The S-curve path based on Bezier curve.

여기서 x_1, y_1, x_2, y_2 는 제어점 c_1, c_2 의 좌표를 나타내며, 시작점 (x_s, y_s, θ_s) 에서 목표점 (x_e, y_e, θ_e) 까지의 경로를 나타낸다. 식 (1)에서 u 는 $0 \leq u \leq 1$ 의 값이며, 곡선 경로의 파라미터를 나타낸다.

곡률 $\kappa(u)$ 는 베지어 곡선의 파라미터 u 에 대하여 식 (2)의 미분방정식으로 나타낼 수 있다.

$$\kappa(u) = \frac{1}{\rho(u)} = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'(u)^2 + y'(u)^2)^{3/2}} \quad (2)$$

$$\begin{cases} x_1 = x_s + d_1 \cos \theta_s \\ y_1 = y_s + d_1 \sin \theta_s \end{cases} \quad (3)$$

$$\begin{cases} x_2 = x_e + d_2 \cos(180 + \theta_e) \\ y_2 = y_e + d_2 \sin(180 + \theta_e) \end{cases} \quad (4)$$

식 (3)과 (4)는 제어점 $c_1(x_1, y_1)$ 과 $c_2(x_2, y_2)$ 를 결정하는 식이며, d_1, d_2 를 결정하는 방법에 따라 경로의 곡률이 결정된다. 변수 d_1, d_2 결정 방법은 이동로봇의 접선가속도 제한을 고려하는 방법을 이용하였다[2]. 이는 시작점과 목표점에서 이동로봇의 속도에 따른 접선가속도 제한을 고려한 경로를 생성할 수 있는 방법이다. 제어점을 찾은 후 식 (1)에 의해 베지어 곡선의 경로를 파라미터화 할 수 있다.

2. 컨볼루션 기반 궤적 계획

파라미터화된 경로는 시작점부터 목표점까지의 이동로봇이 수행할 위치를 나타내지만 시간에 따른 위치정보인 속도 정보는 없다. 이동로봇이 생성된 경로를 추종하기 위해서는 속도정보를 포함한 궤적 생성 방법이 필요하다. 로봇의 궤적 생성방법으로는 컨볼루션 연산자의 특성을 이용한 물리적 제한을 만족하는 궤적 생성 방법을 기반으로 한다[1].

컨볼루션 기반 궤적 생성방법은 컨볼루션 연산의 특성을 이용하며 두 번의 연산을 통해 구동기의 가속도와 저크를 고려, 최대 속도궤적을 생성하는 방법이다.

다음 식 (5)의 $y_0(t)$ 는 물리적 제한을 고려하지 않고 일정한 속도 v_0 로 이동거리 S 만큼 이동하기 위한 사각파형 함수를 나타낸다. 이 때 $t_0 = S/v_0$ 이다.

$$y_0(t) = \begin{cases} v_0, & 0 \leq t \leq t_0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

컨볼루션 연산은 $y_n(t)$ 와 $h_n(t)$ 의 컨볼루션이며, $n = [0, 1, 2]$ 일 때 구동기의 물리적 제한을 만족하는 속도궤적을 생성한다.

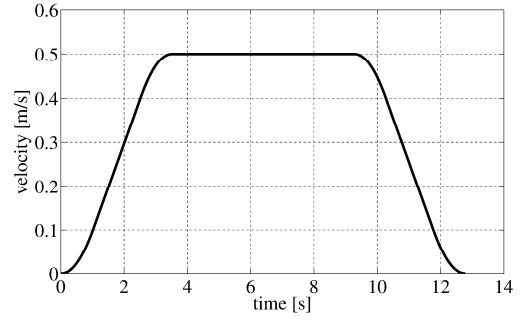


그림 2. 컨볼루션 연산 기반 이동로봇의 중심 속도궤적.

Fig. 2. The trajectory of mobile robot's center velocity based on convolution operation.

$h_n(t)$ 는 사각파형 함수로 $0 \leq t \leq t_n$ 의 구간에서 다음 식 (5)와 같이 정의한다. $y_1(t)$ 는 $y_0(t)$ 와 $h_1(t)$ 의 컨볼루션으로 가속도를 고려하기 위한 컨볼루션이며, $y_2(t)$ 는 $y_1(t)$ 와 $h_2(t)$ 의 컨볼루션으로 가속도가 고려된 속도궤적 $y_1(t)$ 에서 저크를 고려하기 위한 컨볼루션이다.

$$h_n(t) = \begin{cases} 1/t_n, & 0 \leq t \leq t_n \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$\text{where, } t_n = \frac{v_{\max}^{(n-1)}}{v_{\max}''} \quad (6)$$

식 (6)에서 $v_{\max}^{(0)}$ 는 구동기의 최대 속도, $v_{\max}^{(1)}$ 는 최대 가속도, $v_{\max}^{(2)}$ 는 최대 저크를 나타낸다. 로봇의 구동기의 최대 속도궤적을 생성하기 위한 $v_0 = v_{\max}^{(0)}$ 이며, 2번의 컨볼루션 연산을 통해 물리적 제한이 고려된 미분 가능한 S-곡선의 속도궤적이 생성된다.

일반적인 로봇의 제어시스템은 일정한 샘플링 주기로 속도명령을 처리하므로 컨볼루션 연산은 다음 식 (7)과 (8)의 디지털 컨볼루션 식을 이용한다[1].

$$y_n[k] = \frac{y_{n-1}[k] - y_{n-1}[k - m_n]}{m_n} + y_n[k - 1], \quad (7)$$

$$k = [1/T_s], \quad m_n = [t_n/T_s], \quad (8)$$

여기서 T_s 는 샘플링 시간, $[x]$ 는 가우스 양자화 기호로 x 보다 크지 않은 최대 정수를 나타내며, 위의 디지털 컨볼루션의 결과는 샘플링 시간에 따른 오차가 발생할 수 있다.

그림 2는 이동거리 $S = 4.63$ m, $v_{\max} = 0.5$ m/s, $a_{\max} = 0.2$ m/s², $j_{\max} = 0.2$ m/s³일 때, 컨볼루션 연산 결과 이동로봇 중심의 속도궤적을 나타내며 미분 가능한 S-곡선의 속도궤적 $y_2(t)$ 이다.

3. 경로궤적 생성 방법

컨볼루션 연산자 기반 궤적 생성방법은 이동거리와 속도, 가속도, 저크를 기반으로 속도궤적을 생성하며, 이 때 이동로봇의 방향각은 고려되지 않는다. 이동로봇의 물리적 제한과 방향각을 고려하기 위해서는 새로운 경로궤적의 생성이 필요하며, 베지어 곡선의 매개변수 u 를 결정하는 방법으로 경로궤적을 생성하는 방법이 제안되었다[3].

베지어 곡선의 경로에 대한 중심속도궤적을 생성하기 위하여 식 (1)에 일정 매개변수 변위 du 를 사용하여 생성된 경

로를 $R(u)$ 라고 정의하며, 시작점부터 목표점까지의 경로 $R(u)$ 의 길이 S_b 는 다음 식 (9)와 같이 정의한다.

$$S_b = \int_0^1 R(u) du \quad (9)$$

$$R(u) = \sqrt{x'(u)^2 + y'(u)^2}$$

베지어 곡선의 매개변수 u 를 결정하는 방법은 S_b 를 계산하고, 컨볼루션기반 속도계획의 이동거리 S 가 S_b 와 일치하도록 속도함수 $y_2(t)$ 를 생성한다. 생성된 $y_2(t)$ 의 적분과 S_b 를 이용하여 0에서 1까지의 범위로 속도함수를 변환하며, 식 (10)과 같이 나타낼 수 있다.

$$u(t_n) = \frac{\int_0^{t_n} y_2(t) dt}{S_b} \quad (10)$$

식 (10)에 의하여 매개변수는 시간의 함수로 나타낼 수 있으며, $u(t_n)$ 은 $0 \leq u(t) \leq 1$ 구간에서 정의 된다. 이는 중심속도계획의 시간에 따른 이동거리를 고려하여 매개변수의 증가량을 결정하는 것과 같은 의미이다. 매개변수 $u(t)$ 를 식 (1)에 대입하여 생성된 경로 $R(u(t))$ 는 물리적 제한인 속도, 가속도, 저크가 고려되었으며, $R(u)$ 를 시간의 함수로 나타낸 경로계획이다. 주어진 속도를 이용하여 이동로봇이 실제 주행하게 될 경로계획을 생성하기 위해서는 베지어 곡선으로부터 방향각을 식 (11)과 같이 계산하여 식 (12)와 (13)을 통해 실제 주행하게 될 궤적을 계산할 수 있다.

$$\theta_c(t) = \tan^{-1} \left(\frac{y'(u(t))}{x'(u(t))} \right) \quad (11)$$

$$v_x(t) = v_c(t) \cos(\theta_c(t)) \quad (12)$$

$$v_y(t) = v_c(t) \sin(\theta_c(t))$$

$$x_c(t) = \int_0^t v_x(t) dt \quad (13)$$

$$y_c(t) = \int_0^t v_y(t) dt$$

그림 3은 이동로봇이 시작점 (0, 0, 0°)에서 목표점 (2, 4, 0°)까지 계획된 베지어 곡선의 경로와 컨볼루션 연산으로 생성된 속도계획 $y_2(t)$ 를 수식 (12)의 $v_c(t)$ 로 하여 계산된 경로계획이다. 즉 중심속도는 컨볼루션에 의하여 만들어지고 로봇의

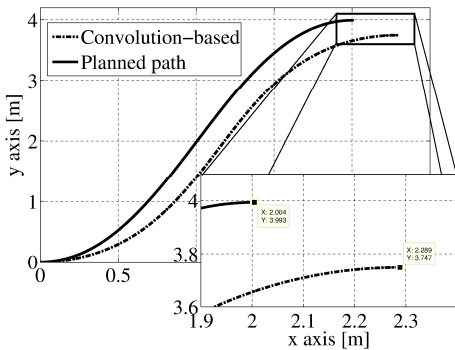


그림 3. 이동로봇 중심의 물리적 제한이 고려된 경로계획.

Fig. 3. The path trajectory considering physical limit of mobile robot's center.

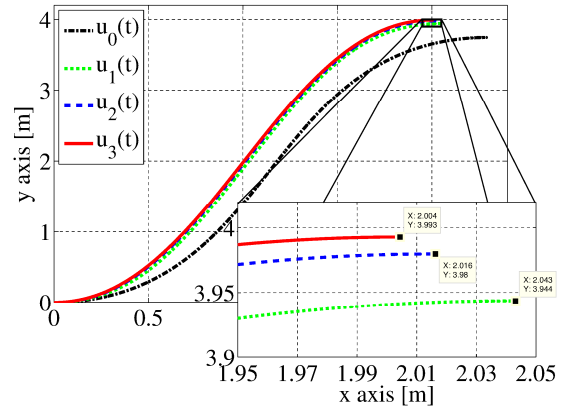


그림 4. 에러 보정 $u(t)$ 에 의해 생성된 경로계획.

Fig. 4. Path trajectory according to the error considered $u(t)$.

방향각 정보는 수식 (12)을 통하여 만들어진 경로이다. 생성된 경로계획의 마지막 점의 좌표는 (2.287, 3.745, 0°)으로 기준 경로를 추종하지 못하며, 큰 오차를 갖는다. 이는 시간의 함수로 생성된 베지어 곡선인 $R(u(t))$ 가 계획된 경로와 일치하지 못하여 발생하는 오차이다.

본 연구에서는 보다 정밀한 경로계획을 생성하기 위해 $u(t)$ 를 결정하는 방법을 제안한다. 수식 (10)에 의하여 $y_2(t)$ 를 통해 계산된 $u(t)$ 와 $u(t)$ 에 의해 생성된 시간의 함수인 경로 $R(u(t))$ 를 이용하여 단위 시간 $du = u(t_n) - u(t_{n-1})$ 에서 곡률에 따른 에러를 보정하는 방식을 제안한다.

$$e(t_n) = u_0(t_n) - \frac{\int_{t_{n-1}}^{t_n} R(u_k(t_n)) dt}{S_b} \quad (14)$$

$$u_{k+1}(t_n) = u_k(t_n) + e(t_n) \quad (15)$$

식 (14)와 (15)를 이용하여 속도계획에 적합한 경로계획을 생성할 수 있는 $u(t)$ 를 계산한다. 여기서 $k = [0, 1, 2, 3, \dots]$ 의 자연수로 보정 횟수를 의미한다. $u_k(t)$ 는 식 (10)을 이용해 처음 계산된 매개변수이며, 식 (14)는 $u_k(t)$ 와 $u_k(t)$ 에 의해 생성된 경로의 오차를 나타내는 에러 값이다. 최적의 경로계획을 얻기 위한 $u(t)$ 로 $\max(|e(t)|) < 0.005$ 를 만족할 때까지 보정을 수행하며 보정횟수에 따른 경로계획은 그림 4에 나타내었다.

그림 4는 최종적으로 결정된 $u(t)$ 에 의해 생성된 경로계획을 나타낸다. 제안된 방법을 통해 $k = 3$ 일 때까지 수행되었으며, 생성된 궤적의 마지막 점의 좌표는 (2.004, 3.993, 0°)으로 기준경로에 매우 근접하게 경로계획이 생성되는 것을 확인할 수 있다.

III. 두 바퀴 이동로봇의 바퀴 속도 제한을 고려한 궤적 계획

1. 두 바퀴 이동로봇의 기구학

물리적 제한을 만족하는 속도계획은 이동로봇의 중심을 기준으로 생성되었으며, 이를 이동로봇의 속도 명령으로 인가하기 위해서는 두 바퀴 이동로봇의 기구학을 고려하여 관절공간에서 두 바퀴의 속도명령으로 변환해 주어야 한다.

이동로봇의 위치는 전역 기준 좌표계와 로봇기준 좌표계로 구성되며, 그림 5에서와 같이 X-Y 직교 좌표계에서 이동로봇의 파라미터들을 나타낼 수 있다. 이동로봇의 위치와 방

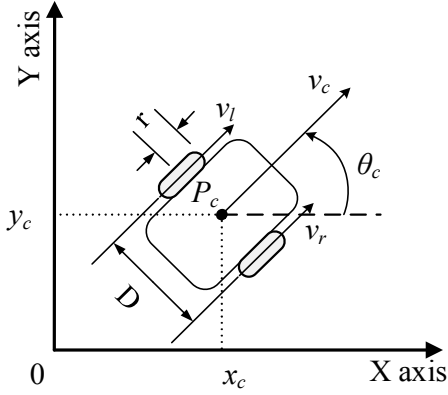


그림 5. 직교 좌표계에서 두 바퀴 이동로봇의 파라미터.

Fig. 5. The parameters of two-wheeled mobile robot in Cartesian coordinates.

향각을 나타내는 P_c 는 전역 기준 좌표계 상에서 다음 식 (16)과 같이 정의 된다.

$$P_c = [x_c, y_c, \theta_c]^T \quad (16)$$

여기서 x 와 y 는 이동로봇의 위치를 나타내는 변수이며, θ_c 은 이동로봇의 방향각을 나타낸다. 직교 좌표계 (Cartesian coordinate)에서 이동로봇의 기구학은 식 (17)과 같이 나타낼 수 있으며, 이동로봇의 선속도는 식에 의해 시간의 함수로 표현이 가능하다.

$$\dot{P}_c = \begin{bmatrix} v_x \\ v_y \\ \omega_c \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta_c & \frac{r}{2} \cos \theta_c \\ \frac{r}{2} \sin \theta_c & \frac{r}{2} \sin \theta_c \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (17)$$

식 (17)에서 r 은 바퀴의 반지름을 의미하고, D 는 이동로봇 두 바퀴 사이의 거리를 의미한다. ω_r 은 오른쪽 바퀴의 회전속도, ω_l 은 왼쪽 바퀴의 회전속도이다. 각 바퀴의 회전속도와 바퀴의 반지름을 이용하여 이동로봇의 각 구동 바퀴의 각속도를 식 (18)과 같이 계산할 수 있으며, 두 바퀴의 선속도는 식 (19)로 계산할 수 있다.

$$\begin{cases} \omega_r = \frac{1}{r} \left(v_c + \frac{D}{2} \omega_c \right) \\ \omega_l = \frac{1}{r} \left(v_c - \frac{D}{2} \omega_c \right) \end{cases} \quad (18)$$

$$\begin{cases} v_r = r \omega_r = v_c + \frac{D}{2} \omega_c \\ v_l = r \omega_l = v_c - \frac{D}{2} \omega_c \end{cases} \quad (19)$$

이동로봇의 두 바퀴의 선속도는 이동로봇 중심의 속도인 v_c 를 기준으로 방향각에 따라 다른 값을 갖게 된다. 그림 6은 속도 제한을 0.5 m/s로 가정하여 컨볼루션 연산 기반으로 생성된 속도계획 $y_c(t)$ 와 이동로봇의 기구학을 이용해 계산된 두 바퀴의 선속도 $v_r(t)$, $v_l(t)$ 를 나타낸다. 계산된 두 바퀴의 선

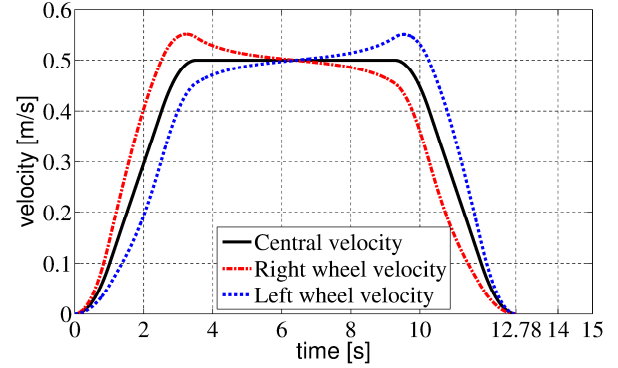


그림 6. 두 바퀴 이동로봇의 속도 명령.

Fig. 6. The velocity commands for two-wheeled mobile robot.

속도는 이동로봇 중심의 최대 속도인 0.5 m/s를 초과하는 속도 명령이 존재하는 것을 확인할 수 있다.

2. 두 바퀴의 속도 제한을 고려한 궤적 계획 방법

차분 구동형 두 바퀴 이동로봇의 최대 속도는 일반적으로 두 바퀴 모두 최대 속도로 구동 되어 직선 주행을 하는 경우의 속도를 의미하며, 곡선 주행에서 두 바퀴는 최대 속도를 넘는 속도명령은 구동기가 추종할 수 없다. 그림 6에서 이동로봇의 기구학에 의해 생성된 두 바퀴의 속도 명령은 미리 설정된 최대속도 0.5 m/s를 만족하지 못하며, 이는 이동로봇의 속도명령으로 적합하지 않다. 이러한 문제점을 해결하기 위한 방법으로 최대 속도를 재 설정하여 속도계획의 전반적인 속도를 작게 생성하는 방법이 선행 연구되었다[4]. 하지만 이 방법은 주행시간이 크게 증가하는 단점이 있다.

본 연구에서는 두 바퀴 이동로봇의 각 구동 바퀴의 물리적 제한을 만족하도록 두 바퀴의 최대 허용속도를 고려한 최대 속도계획을 생성하기 위한 방법을 제안한다. 다음 식 (20)은 곡선 경로의 곡률과 이동로봇의 두 바퀴 사이의 거리에 의해 물리적 제한을 만족하지 못하는 $v_c(t)$ 를 보정하기 위한 식이다.

$$v_{c(adj)}(t) = v_c(t) - \frac{D}{2} \text{sgn}(\kappa(t)) \omega_c(t) \quad (20)$$

식 (20)의 $\kappa(t)$ 는 $u(t)$ 에 의해 생성된 경로계획의 곡률을 의미하며 식 (2)에 의해 계산된다. $v_{c(adj)}(t)$ 는 보정된 이동로봇의 중심속도이며, 그림 7은 $v_{c(adj)}(t)$ 와 이를 기반으로 계산된 두

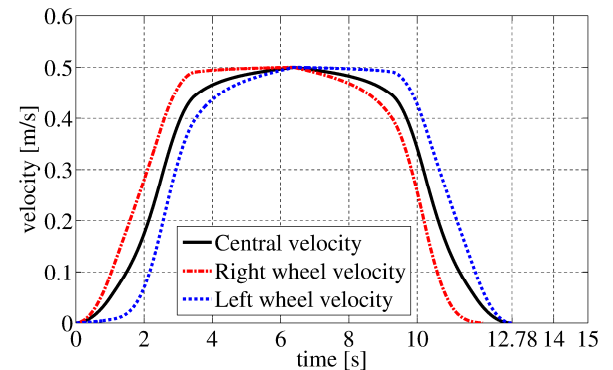


그림 7. 두 바퀴의 물리적 제한을 고려한 속도 명령.

Fig. 7. Velocity commands considering two wheels velocity limits.

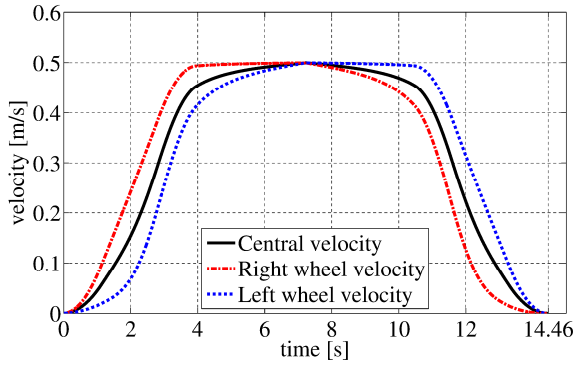


그림 8. 이동거리 S 가 고려된 두 바퀴 이동로봇의 속도 명령.
Fig. 8. Velocity commands considering the distance of S for two-wheeled mobile robot.

바퀴의 속도계획이다. 이동로봇 중심과 두 바퀴의 속도계획 모두 물리적 제한을 만족하는 것을 확인할 수 있다.

그림 6과 그림 7에서 주행 시간은 동일하게 나타나며 속도가 변화된 것은 결론적으로 보정된 중심속도계획 $v_{c(adj)}(t)$ 의 이동거리 S_{adj} 가 $v_c(t)$ 의 이동거리 S 보다 적은 거리를 이동하는 것을 의미한다. 따라서 계획된 경로를 추종하기 위하여 S_{adj} 와 S 가 같은 이동거리를 갖도록 하기 위해 식 (21)에 의해 샘플링 시간 T_s 의 비율을 조정한다.

$$T_{s,adj} = T_s \frac{S}{S_{adj}} \quad (21)$$

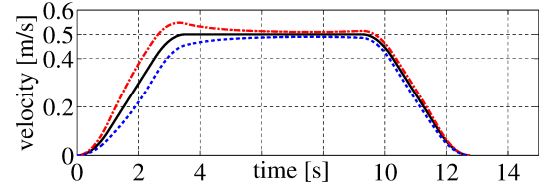
$$t_{adj} = nT_{s,adj}$$

여기서 $T_{s,adj}$ 는 보정된 샘플링 시간, n 은 속도계획의 샘플의 수, t_{adj} 는 보정된 시간이다. 본 연구는 시작점 $(0, 0, 0^\circ)$ 에서 목표점 $(4, 2, 0^\circ)$ 까지 이동하는 경우 베지어 곡선의 d_1 과 d_2 는 0.8083 m이며, 이동거리는 $S = 4.63$ m이다. 이 때 T_s 는 0.01 s, $t = 12.78$ s로 보정 전 $n = 1278$ 이다. 이 때 $S_{adj} = 4.092$ m이며, 식 (14)에 의해 보정된 시간은 14.46 s가 되며, $n_{adj} = 1446$ 이다.

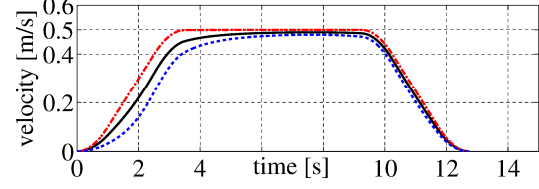
그림 8은 이동거리 S 를 고려하여 샘플링 시간이 보정된 두 바퀴 이동로봇의 속도 명령이며, 그림 7과 비교하여 주행 시간이 증가하였다. 이는 속도 명령의 면적인 거리를 동일하게 하기 위해 그림 7의 낮아진 속도에 따라 주행 시간이 증가된 것을 알 수 있다. 이 때, 속도 명령의 샘플링 주기는 식 (14)에 의해 0.011315 s로 일정한 샘플링 시간을 갖지만 일반적인 제어시스템에서 사용하는 샘플링 주기를 위해서 본 연구에서는 선형 보간법 (Linear interpolation)으로 0.01 s 주기의 샘플링 시간에서 속도 값을 추정하여 실제 이동로봇의 속도 명령에 사용한다. 이 때 보간된 속도계획과 함께 곡선 경로의 θ 또한 선형보간법을 적용하여 식 (21)을 통해 두 바퀴의 속도계획을 생성해야 한다. 보간된 속도와 각도는 보간법 오차가 발생하지만 그림 8의 속도 명령과 거의 일치한다.

IV. C-곡선 경로의 속도계획

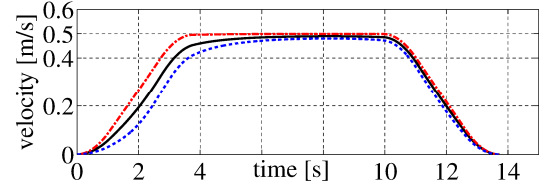
베지어 곡선의 경우 목표 위치에서 이동로봇의 방향각을 고려하여 경로를 생성 하며, 방향각에 따라 주행경로는 직선 또는 'C', 'S' 형태의 곡선으로 이루어 진다. 이 장에서는 'C'형태의 곡선 경로에서 제안된 방법에 의해 속도계획을



(a) Convolution-based.



(b) Wheel velocity limit consideration.



(c) Adjusting T_s ratio.

— Central velocity — Right wheel — Left wheel

그림 9. C-곡선 경로에서의 속도계획.

Fig. 9. Velocity trajectory for C-curve path.

확인한다.

그림 9는 시작점 $(0, 0, 0^\circ)$ 에서 목표점 $(2, 4, 90^\circ)$ 까지 이동로봇의 주행을 계획할 경우 'C'형태의 곡선 경로에 대한 속도계획이다. 그림 (a)와 같이 C-곡선의 경로에 대한 속도계획에서도 컨볼루션 기반의 계획 계획을 통해서도 각 바퀴의 물리적 제한을 모두 만족하지 못한다. 그림 (b)와 같이 속도를 조절하면 실제 경로를 추종하지 못하며, 각 바퀴의 속도 제한을 고려하기 위해 S-곡선의 계획 계획 방법과 동일한 방법으로 그림 (c)와 같은 속도계획을 생성해야 하며, 생성된 속도계획에 대하여 경로계획을 생성해야 한다. 제안된 방법으로 생성된 경로계획은 그림 10에 나타내었으며, 생성된 경로계획은 기준 경로와 매우 근사한 계획을 갖는다.

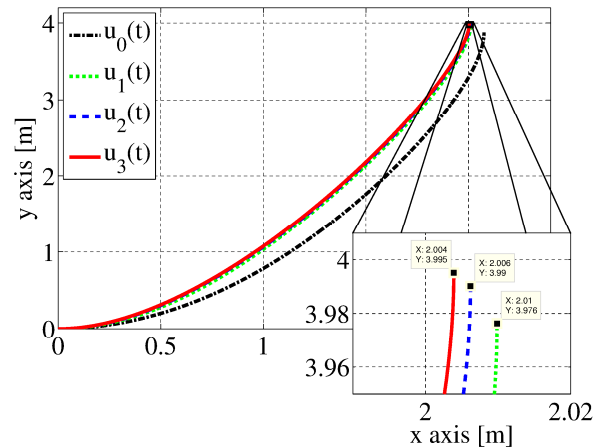


그림 10. 수치적으로 계산된 경로계획 (C-curve path).

Fig. 10. Numerically computed trajectory for C-curve path.

V. 결론

본 논문 이동로봇의 두 바퀴의 속도 제한을 고려한 최대 속도제적 생성 방법을 제안하였다. 물리적 제한을 만족하기 위하여 컨볼루션 기반 속도제적 생성방법을 사용하였으며, 베지어 곡선의 경로를 추종하기 위한 매개변수 결정 방법으로 기존에 제안된 매개변수를 시간의 함수로 변환하는 방법에서 오차를 보정하는 방법을 제안하였다.

제안된 베지어 곡선의 매개변수를 결정하는 방법은 기존의 매개변수를 시간의 함수로 변환하는 방법에서 매개변수에 의해 생성되는 경로제적을 주어진 속도제적에 적합하도록 생성하는 방법이며, 기존의 방법보다 정확히 경로를 추종할 수 있다.

본 연구에서 적용된 속도제적 생성방법은 컨볼루션 연산자의 특성을 기반으로 두 바퀴 이동로봇의 구동기에서 주행 가능한 최대 속도제적을 생성하는 방법이며, 곡선 경로에서 물리적 제한을 만족하지 못하는 구간에 대하여 곡선의 방향각을 고려하여 속도제한을 만족하는 속도제적을 생성하였다. 이는 두 바퀴 이동로봇의 구동기에서 실제로 구동 가능한 속도제적으로 이동로봇의 속도명령으로 사용이 가능하다. 또한 제안된 속도제적은 구동기의 최대 허용속도를 고려하여 최소 주행시간으로 주행이 가능하다.

REFERENCES

- [1] G. Lee, D. Kim, and Y. Choi, "Faster and smoother trajectory generation considering physical system limits under discontinuously assigned target angles," *IEEE International Conference on Mechatronics and Automation*, pp. 1196-1201, Aug. 2012.
- [2] K. G. Jolly, R. S. Kumar, and R. Vijayakumar, "A bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robotics and Automation Systems*, vol. 57, pp. 23-33, Jan. 2009.
- [3] G. J. Yang and B. W. Choi, "Smooth trajectory planning along bezier curve for mobile robots with velocity constraints," *International Journal of Control and Automation*, vol. 6, no. 2, pp. 225-234, Apr. 2013.
- [4] G. J. Yang and B. W. Choi, "Joint space trajectory planning considering physical limits for two-wheeled mobile robots," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 6, pp. 540-546, Jun. 2013.
- [5] J. H. Han, S. H. Park, D. H. Lee, K. W. Noh, and J. M. Lee, "A mobile robot estimating the real-time moving sound sources by using the curvature trajectory," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 20, no. 1, pp. 48-57, 2014.
- [6] J. H. Koh, G. J. Yang, and B. W. Choi, "Elderly assistance system development based on real-time embedded linux," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 11, pp. 1036-1042, 2014.
- [7] S. Y. Lee, S. H. Lee, and Y. S. Hong, "A real-time collision-free trajectory planning and control for a car-like mobile robot," *Journal of Control, Automation and System Engineering (in Korean)*, vol. 5, no. 1, pp. 105-114, Jan. 1999.
- [8] M. Brezak and I. Petrovic, "Time-optimal trajectory planning along predefined path for mobile robots with velocity and

acceleration constraints," *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Budapest, Hungary, pp. 942-947, 2011.



최 병 옥

1988년~1992년 한국과학기술원 전기및 전자공학과 석사 및 박사졸업. 1988년~2000년 LG산전 중앙연구소 책임연구원. 2000년~2005년 선문대학교 제어계측공학과 부교수. 2003년~2005년 임베디드 웹 대표이사. 2007년~2008년 Nanyang

Technological University, Senior Fellow. 2005년~현재 서울과학기술대학교 전기정보공학과 교수. 관심분야는 실시간 시스템 설계, 임베디드 시스템, 임베디드 리눅스, 지능형 로봇 소프트웨어.



양 길 진

2013년 원광대학교 전자공학과 졸업. 2015년 서울과학기술대학교 전기공학과 석사졸업. 2015년~현재 세연테크 연구원. 관심분야는 실시간 시스템 설계, 임베디드 리눅스, 지능형 로봇 소프트웨어.