

동적 프로그래밍을 이용한 자율이동로봇의 최소시간 경로계획

윤희상, 박태형
충북대학교 제어계측공학과

Near-Time-Optimal Path Planning of Mobile Robot Using Dynamic Programming

Hee-Sang Yoon, Tae-Hyoung Park
Dept. of Control and Instrumentation Eng.

Abstract - 자율이동로봇의 기구학 및 동역학 모델을 통해 로봇의 특성을 고려하여 목표점까지 이동시간이 최소화 되는 경로 생성 방법을 제안한다. 본 논문에서는 초기 전역 경로를 골격선 그래프와 디스트라알고리즘으로 설정하고, 로봇의 기구학 및 동역학 제약조건을 고려하여 동적 프로그래밍으로 경로를 개선한다. 개선된 경로는 자율이동로봇이 이동하는데 걸리는 시간을 단축한다. 마지막으로 시뮬레이션을 통해 제안하는 방법의 성능을 검증한다.

1. 서 론

최근 들어 다양한 형태의 로봇이 개발 되고 있다. 특히 자신의 위치를 인식하여 원하는 목적지 까지 로봇 스스로 찾아가는 자율이동기능은 가정의 청소 및 애완용로봇, 산업현장의 안내 및 운반로봇, 군사용의 경비 로봇 등 여러 분야의 다양한 로봇에서 핵심적인 기능을 담당하고 있다. 이런 자율이동기능이 적용된 자율이동로봇들의 주행성능을 높이기 위해서, 현재 지점에서 목표 지점까지의 이동 경로를 생성하는 경로계획 기능이 중요시 된다 [1]. 경로계획은 시작점, 목표점, 장애물의 위치 등의 환경정보로부터 시작점과 목표점까지의 경로를 생성하는 문제로, 많은 이동 가능 경로 중에서 최단거리, 최소시간, 최소에너지 등을 고려하여 경로를 생성하는 연구가 수행되었으며 본 논문에서는 이동시간을 최소로 되는 경로계획방법을 다룬다.

로봇의 이동 시간을 고려한 경로계획의 최근 연구는 로봇의 기구학[2] 및 동역학을 고려한 방법들이 연구되었다. [3]에서 제안한 방법은 쉐분해 방식으로 초기경로를 생성하고, B-스플라인을 이용하여 스푸스(smooth)한 경로를 기구학 및 동역학 제약조건에서 생성하는 방법이다. 그러나 이런 방법들은 복잡한 연산을 필요로 한다.

본 논문은 동적 프로그래밍으로 자율이동로봇의 동역학적 특성을 고려하여, 목표지점까지 최소시간으로 운행할 수 있는 경로생성방법을 제안한다. 경로생성은 세션화 알고리즘을 이용하여 골격선 그래프를 구성하고, 탐색알고리즘인 디스트라(Dijkstra) 알고리즘으로 초기경로를 생성한다. 생성된 초기경로에 대하여 동적 프로그래밍을 이용하여 로봇의 동역학 제약조건 범위 안에서 경로를 반복적으로 개선한다. 개선된 경로는 로봇의 이동하는데 걸리는 시간을 줄인다. 마지막으로 시뮬레이션을 통해 제안하는 방법의 성능을 검증한다.

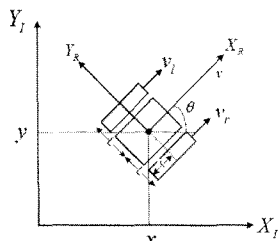
2. 경로 계획 방법

2.1 자율이동로봇 기구학 모델

본 논문에서 사용하는 자율이동로봇은 그림 1과 같이 2개의 구동휠(differential-drive)로 구성된 차동형 로봇(differential-drive robot)이다. 로봇은 2개의 구동휠의 속도차에 의해 이동 궤적이 생성된다. 로봇의 진행방향에 대한 로봇좌표계를 $\{X_R, Y_R\}$ 직교좌표계를 $\{X_I, Y_I\}$ 라 할 때, 로봇의 무게 중심의 위치는 (x, y) , 자세는 θ 로 정의한다. 로봇의 바퀴가 미끄러짐(slip)이 없다고 가정할 때, 차동형 로봇의 기구학 식은 다음과 같다.

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

여기서 \dot{x} 은 로봇의 x 축 방향 속도, \dot{y} 은 y 축 방향 속도, $\dot{\theta}$ 은 로봇 자체의 각속도를 나타낸다. 또한 차동형 로봇의 휠의 반지름을 r , 각 바퀴의 각속도를 w_R, w_L 이라 할 때 각 바퀴에 의한 속도 v_R, v_L 은 식(2)와 같이 나타낼



<그림 1> 로봇 모델

수 있으며, 각각의 바퀴에 의한 로봇 전체의 선속도 및 로봇의 회전 속도는 각각 식(3) 식(4)와 같다.

$$\begin{bmatrix} v_R = rw_R \\ v_L = rw_L \end{bmatrix} \quad (2)$$

$$v = \frac{v_R + v_L}{2} \quad (3)$$

$$\dot{\theta} = \frac{v_R - v_L}{2L} \quad (4)$$

로봇의 회전은 각 바퀴의 속도차에 의해 결정되며 로봇의 회전 반경을 나타내는 곡률 κ 는 식 (5)와 같다.

$$\kappa = \frac{\dot{\theta}}{v} \quad (5)$$

또한 로봇은 회전할 때 최대 곡률 $|\kappa| \leq \kappa_{max}$ 범위에서 회전하게 된다.

2.2 자율이동로봇 동역학 모델

자율이동로봇의 일반적인 동역학식은 로봇의 휠 및 프레임의 관성 및 모터의 토크를 고려하여 구할 수 있으며 세부적인 유도과정은 참고문헌[2]에 유도되어 있다. 최소시간 경로계획을 위해서는 식(6)과 같이 모터의 토크 τ_R, τ_L 과 휠의 각속도 간의 관계식을 통해 모터의 최대 토크 범위 $|\tau| \leq \tau_{max}$ 에서의 각가속도 $\ddot{\theta}_R, \ddot{\theta}_L$ 를 구하여 로봇의 속도를 구할 수 있다

$$\begin{bmatrix} (mL^2 + I)\frac{r^2}{4L^2} + I_w & \frac{r^2}{4L^2}(mL^2 - I) \\ \frac{r^2}{4L^2}(mL^2 - I) & (mL^2 + I)\frac{r^2}{4L^2} + I_w \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (6)$$

여기서 m 은 이동로봇의 전체 질량이며, I_w 는 휠 관성, I 는 로봇이 회전시 발생하는 회전관성을 나타낸다.

2.3 경로계획방법

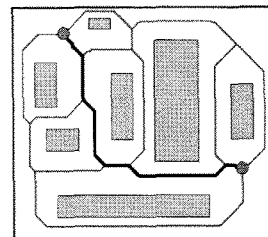
자율이동로봇의 최소시간 경로계획 문제는 로봇의 기구학 및 동역학 제약조건을 고려하여 푸는 전형적인 비선형 문제이다. 본 논문에서는 복잡한 비선형 문제인 최소시간 경로계획 문제를 단순화하기 위해 [4]에서 제안한 동적경로계획방법을 사용한다. 동적경로계획방법은 초기 전역 경로를 생성하고, 동적프로그래밍을 이용하여 개선하는 방법으로, 본 논문에서는 기구학 및 동역학 제약조건을 고려한 동적프로그래밍을 풀어 로봇의 이동시간을 단축한다.

2.3.1 초기경로생성

주어진 환경에서 경로의 연결 관계를 보여주는 골격선 그래프를 생성하는 방법은 다양하다. 그 중 세션화 알고리즘은[5]은 영상처리 과정의 하나인 골격화에 사용되는 방법으로서 경로계획 문제에도 적용할 수 있다. 골격선 그래프를 구성한 후에는 초기경로생성을 위해 시작점에서 목표점까지의 이동경로를 탐색해야 한다. 탐색알고리즘은 최적화가 입증된 디스트라 알고리즘을 사용한다. 골격선 그래프의 구성과 초기경로생성의 예를 그림 2에 나타내었다.

2.3.2 경로추정

골격선 그래프를 이용하여 생성한 초기경로는 최적에 가까운 경로는 아



<그림 2> 골격선 그래프 및 초기경로

니다. 기존의 동적경로계획방법에서는 경로의 경유점들을 위치를 바꾸어 동적프로그래밍으로 직선거리의 합이 최소가 되는 경로를 탐색하였다. 이 경로는 직선으로 이루어져 이동거리가 짧지만 직선이 만나는 경유점에서는 로봇이 무한대의 가속도로 회전해야 하므로 실제 로봇이 수행하지 못한다. 따라서 로봇이 이동 가능하도록 경로를 생성해야하며, 또한 로봇의 기구학 조건인 곡률과 동력학 조건인 최대 토크 조건을 만족하는 로봇 바퀴의 속도를 결정하여 경로를 추정해야한다.

k 번째 경유점을 v_k 라 할 때, 로봇의 위치는 (x_k, y_k, θ_k) , 각 바퀴의 속도는 $[v_R(k), v_L(k)]$ 라 하면, 경로는 최대 곡률 κ_{max} 과 최대 토크 τ_{max} 의 범위 안에서 경유점 v_{k+1} 까지 빠르게 이동하는 $[v_R(k+1), v_L(k+1)]$ 을 찾아 위치 $(x_{k+1}, y_{k+1}, \theta_{k+1})$ 를 구하는 문제이다. 여기서 $[v_R(k+1), v_L(k+1)]$ 을 구하기 위해, 로봇의 선속도 v 를 일정하다고 가정하였으며, v_k 와 v_{k+1} 사이에 이동시간을 Δt_k 라 정의 하였다. v_{k+1} 의 위치는 (x_{k+1}, y_{k+1}) 정해져 있으나 자세 θ_{k+1} 와 각 바퀴의 속도 $[v_R(k+1), v_L(k+1)]$ 를 추정하기 위해 그림 3과 같이 v_{RP1} 과 v_{RP2} 를 선정한다. v_{RP1} 은 v_k 에서의 각 바퀴의 속도 $[v_R(k), v_L(k)]$ 로 속도 변화 없이 Δt_k 로 이동하였을 때의 위치이며, v_{RP2} 는 최대 곡률 조건 안에서 최대 토크로 이동 하였을 때의 위치로 바퀴의 속도는 $[v_R^+(k), v_L^+(k)]$ 로 정의된다. 여기서 구하려는 v_k 의 각 바퀴의 속도 $[\hat{v}_R(k+1), \hat{v}_L(k+1)]$ 와 자세 $\hat{\theta}_{k+1}$, 곡률 $\hat{\kappa}_{k+1}$ 은 식(7)-(9)로 추정가능하다.

$$\hat{\theta}_{k+1} = \theta_k + (\angle v_{k+1} - \angle v_{RP1}) \times \frac{\theta_{RP2} - \theta_{RP1}}{\angle v_{RP2} - \angle v_{RP1}} \quad (7)$$

$$\hat{\kappa}_{k+1} = \kappa_k + (\angle v_{k+1} - \angle v_{RP1}) \times \frac{\kappa_{RP2} - \kappa_{RP1}}{\angle v_{RP2} - \angle v_{RP1}} \quad (8)$$

$$\begin{cases} \hat{v}_R(k+1) = (\hat{\kappa}_{k+1} + 1)v \\ \hat{v}_L(k+1) = (1 - \hat{\kappa}_{k+1})v \end{cases} \quad (9)$$

여기서 $\angle v_{k+1}, \angle v_{RP1}, \angle v_{RP2}$ 는 각 v_k 와 이루는 각이다.

2.3.3 동적프로그래밍을 이용한 경로개선

앞서 경로추정을 통해 다음 경유점까지 로봇이 이동 가능한 상태를 추정할 수 있다. 그러나 이렇게 추정한 경로 또한 최적에 가까운 경로는 아니며 경로를 최소시간에 가깝도록 개선해야한다. 본 논문에서는 경로개선을 위해 경유점들의 위치를 그림 4와 같이 경유점을 중심으로 경로에 수직방향으로 $w \times 1$ 개의 격자를 두었다. 이는 다른 격자점으로 경유점이 변경 될 경우 총 이동시간이 단축 될 수 있다.

격자점을 선택하는 문제는 동적 프로그래밍 알고리즘[6]을 적용한다. DP의 적용을 위해서는 스테이지(stage)와 스테이트(state)로 구성된 탐색 평면이 필요하다. 탐색평면은 초기경로에서 탐색된 경유점들을 일정 간격으로 나누어 M 개의 스테이지를 구성하고, 새롭게 구성된 경유점에 대해 스테이트로 구분한다.

동적프로그래밍을 이용한 경로 개선 알고리즘은 다음과 같다.

- S1. 스테이지 $k=1$ 에 대하여 이동 시간 $t_{k,1}^* = 0$ 으로 설정한다.
- S2. 스테이지를 $k=2, \dots, M-1$ 로 증가시키면서, 격자점 $v_{k,\alpha}$ 에 대해 v_{RP1}, v_{RP2} 를 각 v_k 구하여 식 (7)-(9)를 계산하고, 이동시간 $t_{k,\alpha}^*$ 를 다음과 같이 계산하여 저장한다.

$$t_{k,\alpha}^* = \min_{\beta} (t_{k-1,\beta}^* + t_{(k-1,\beta),(k,\alpha)}) \quad (10)$$

$t_{k,\alpha}^*$ 가 생성되는 β 값을 지시자 $\pi_{k,\alpha}$ 로 저장한다.

- S3. 최종 스테이지 $k=M$ 에 대하여, 시작점부터 목표점까지의 최소수직거리 $t_{M,1}^*$ 를 (7)와 같이 구하고 지시자 $\pi_{M,1}$ 을 저장한다.
- S4. 스테이지를 $k=M, \dots, 2$ 로 감소시키면서 지시자를 역 추적하며, 각 스테이지의 최적 격자점 v_k^* 을 찾는다. 개선 경로 p^* 는 각 스테이지의 최적 격자점으로 다음과 같이 구성된다.

$$p^* = \langle v_1, v_2^*, \dots, v_{M-1}^*, v_M \rangle \quad (11)$$

- S5. 개선 경로의 경유점을 중심으로 다시 격자지도를 구성하여, 일정 값 이상의 경로 개선이 없을 때 까지 S1-S4의 과정을 반복한다.

3. 시뮬레이션

제안한 경로계획 방법의 성능을 평가하기 위해서 시뮬레이션을 수행하였다. 시뮬레이션은 $10[m] \times 10[m]$ 크기의 평면에서 다음과 같은 파라미터(parameters)를 가지고 수행되었다.

$$\begin{cases} r = 0.10m, & m = 51.0kg, & I = 27.17kgm^2 \\ L = 0.75m, & & I_w = 0.0050kgm^2 \end{cases} \quad (12)$$

또한 로봇의 선속도는 $v=1[m/sec]$, 각 바퀴의 최대 속도 $v_{L_{max}} = v_{R_{max}} = 1.5[m/s]$ 이며, 최대 토크 $\tau_{max} = 1[Nm]$ 로 설정하였다.

그림 5는 골격선 그래프로 초기 경로를 생성하고, 동적프로그래밍으로 개선한 경로를 보여준다. 초기 경로의 경우 완전한 곡선이 아니므로 동적프로그래밍을 1회 적용하여, 반복 적용된 개선 경로와 비교하였다. 초기 경로와 개선 경로의 이동시간은 표 1과 같으며, 개선 경로가 초기 경로에 비해 약

13% 정도 빠른 것을 확인할 수 있다. 따라서 개선 경로가 좀 더 부드럽고 빠르게 이동하는 것을 확인 할 수 있다.

<표 1> 이동시간 비교

	초기경로	개선경로
이동 시간 [sec]	9.74	8.41

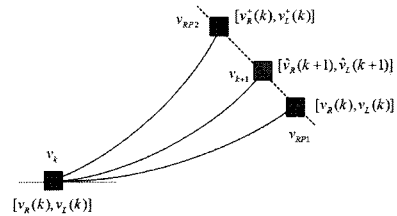
4. 결 론

본 논문은 자율이동로봇의 최소시간 경로계획방법을 제안하였다. 제안 방법은 초기 경로를 생성하고 생성된 경로를 개선하는 단계로 구성된 방법이다. 초기 경로는 세선화 기법으로 골격선 그래프를 구성하고, 딥스트라알고리즘을 사용하였다. 생성된 초기경로는 로봇의 기구학 및 동력학을 고려하여 동적프로그래밍에 의해 경로를 개선하였다. 기존의 동적경로계획방법에서[4]는 이동거리를 줄이는 방법이었으나, 본 논문은 이동시간을 단축 하였으며, 시뮬레이션을 통해 성능을 확인하였다.

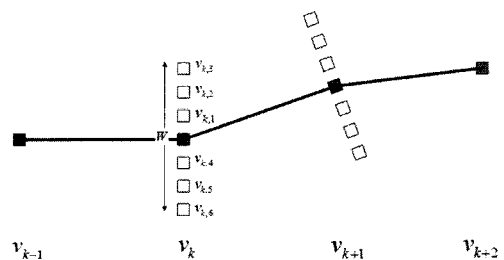
제안한 방법은 경로계획문제를 단순화하기 위하여 로봇의 선속도를 일정하다고 가정하였다. 후속 연구를 통하여, 일정속도가 아닌 로봇의 속도 정보를 가지고 보다 능동적인 경로계획 문제의 해법을 제시하고자 한다.

<참고 문헌>

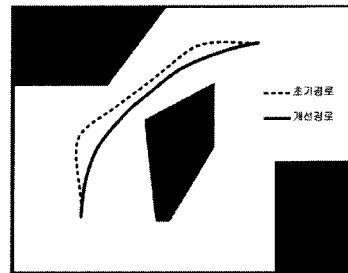
- [1] R. Siegwart, I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robot*, MIT press, 2004.
- [2] M. Yamamoto, M. Iwamura, A. Mohri, "Quasi-Time-Optimal Motion Planning of Mobile Platforms in the Presence of Obstacles," Proc. of the 1999 IEEE Int. Conf. on Robotics & Automation, pp.2958-2963, 1999.
- [3] W. W. Chen, P. P. Woo, "Time Optimal Path Planning for a Wheeled Mobile Robot," Journal of Robotic System, vol.17, pp.585-591, 2000.
- [4] 윤희상, 유진오, 박태형, "자율이동로봇을 위한 동적 경로 계획 방법," 제어로봇시스템학회논문지, 제14권, 제4호, pp.392-398, 2008.
- [5] E. R. Davies, *Machine Vision*, 3rd Edition, Morgan Kaufmann publishers, 2004.
- [6] T. H. Cormen, C. E. Leiserson, et al, *Introduction to Algorithms*, 2nd Ed., The MIT Press, 2001.



<그림 3> 휠 속도 추정 방법



<그림 4> 격자구성



<그림 5> 생성 경로