

### 3차원 장애물에서의 이족보행로봇을 위한 이동경로계획 알고리즘의 설계

민 승 기\*, 김 대 원  
 명지대학교 제어계측공학과

#### A Design of Path Planning Algorithm for Biped Walking Robot in 3-D Obstacle Environment

Seung Ki Min\*, Dae Won Kim  
 Dept. of Control and Instrumentation Eng., MyongJI University

**Abstract** - This paper presents a path planning algorithm for biped walking robot in 3-D workspace. Since the biped walking robot can generate path on some 3-D obstacles that cannot generate path in case of mobile robot, we have to make a new path planning algorithms. A 3-D-to-2-D mapping algorithm is proposed and two kinds of path planning algorithms are also proposed. They make it easier to generate an efficient path for biped walking robot under given environment. Some simulation results are shown to prove the effectiveness of proposed algorithms.

#### 1. 서 론

이동로봇의 경로를 계획하는 문제는 지금껏 많은 연구가 거듭되어왔다. 그러나 이러한 연구들은 이족형 로봇 중 바퀴구동형 로봇에 집중되어 이족보행로봇에는 적합하지 않고 또한 경로생성에도 여러 가지 제한요소가 생기게된다. 이족보행로봇은 인간과 유사한 걸음걸이를 가졌기 때문에 바퀴형 로봇이 생성할 수 없는 경로, 즉, 계단이나 깊이 패인 지형 등에도 경로생성이 가능하다. 따라서 기존의 바퀴형 로봇의 알고리즘에 이족보행로봇의 구조적 요건[6]을 고려하여 새로운 경로계획 알고리즘을 제안할 수 있다.

3차원 환경을 2차원으로 바꾸어주는 문제는 3차원상의 복잡한 문제를 간단히 2차원 상에서 해결할 수 있다는 측면에서 중요한 의미를 가진다고 볼 수 있다. 이것을 작업공간상의 3차원 장애물에 적용하게 되면 경로를 계획할 때 발생하는 어려움을 기존에 2차원 상에서 적용 가능한 여러 알고리즘 [1][2][5]으로 문제를 해결해 갈 수 있다. 본 논문에서는 이와 같이 3차원환경을 2차원으로 바꾸어주는 알고리즘을 제안하고, 바뀌어진 환경에서 적용할 수 있는 궤도 추적 알고리즘과 가상 장애 알고리즘을 설계하고자 한다.

#### 2. 3차원 장애물에서의 이족보행로봇

일부 3차원 장애물은 바퀴구동형 로봇에게는 장애물로만 인식된다. 그러나 이족보행로봇에 있어서 3차원 장애물은 로봇의 구조나 걸음걸이 알고리즘에 따라 장애물이 아닌 경로생성이 가능한 공간으로 변환될 수 있다. 또한 주행 중 걸음걸이 알고리즘의 변화를 주어 실제 바퀴형 로봇이 생성해낼 수 없는 경로를 계획해 낼 수 있다는 점과 이런 특징을 이용하여 이전에 연구해온 여러 알고리즘을 신축성 있게 적용해서 새로운 경로계획 알고리즘을 만들어 낼 수 있다.

##### 2.1 3-D-to-2-D Mapping algorithm

3차원 공간의 장애물, 즉, 계단이나 경사면 등을 2차원 공간상으로 변환하는 것이 목적이며 여기에는 2차원 상에서는 없었던 선 장애물과 가상 장애물의 개념이 더해진다. 그림 1에서 보면 2차원 변환 알고리즘의 매개변수들과 구조를 볼 수 있다. 이것은 이족보행로봇의 구조에 따라 변형될 수 있으며 로봇이 대칭구조일때와 비대칭구조일 때 두 가지로 분류하여 변환 알고리즘을 세울 수 있다.

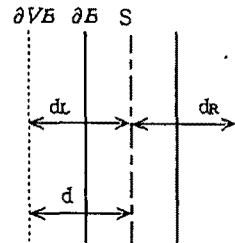


그림 1. 2차원 변환과 매개변수

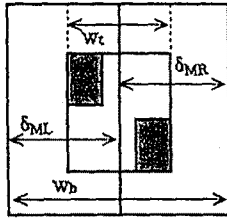


그림 2. 이족보행로봇의 걸음새

(1) 대칭구조

$$d = \min_{p \in \partial B} \|S - p\| + \frac{w_r - w_k}{2} \quad (1)$$

$$\partial B \triangleq \{ \beta \mid \beta \in W_p \text{ and } \| \beta - S \| \triangleq d \}$$

여기서,

$\partial B$  : 선 장애물의 경계  
 $W_p$  : 작업영역

(2) 비대칭구조

$$d_R = \min_{p \in \partial B} \|S - p\| + w_r - \left( \delta_{ML} + \frac{w_k}{2} \right) \quad (2)$$

$$d_L = \min_{p \in \partial B} \|S - p\| + w_r - \left( \delta_{MR} + \frac{w_k}{2} \right) \quad (3)$$

$$\partial VB \triangleq \{ \beta_l \mid \beta_l \in W_p \text{ and } \beta_l \in HP_L(S) \text{ and } \| \beta_l - S \| \triangleq d_L \} \cup \{ \beta_r \mid \beta_r \in W_p \text{ and } \beta_r \in HP_R(S) \text{ and } \| \beta_r - S \| \triangleq d_R \}$$

여기서,

$\partial VB$  : 가상 장애물  
 $HP_L(S)$  : 직선 S의 좌 반평면  
 $HP_R(S)$  : 직선 S의 우 반평면

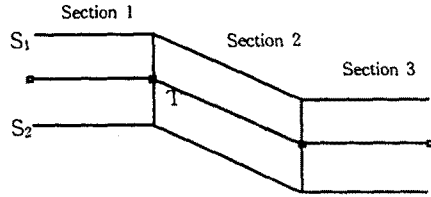
그림 2에서는 로봇의 걸음새와 실제 로봇의 몸체가 투영된 모습을 나타내고 있으며 계산의 편의를 위해 직사각형으로 근사화 시켰다. 이하 표시되는 로봇의 그림은 걸음새를 근사화 시킨 직사각형으로 표현한다. 3차원 장애물은 위 식에서와 같이 로봇의 구조에 따라 변형될 수 있으며 가상 장애물은 로봇의 몸체가 실제 닿는 부분이고 선 장애물은 로봇의 걸음걸이 알고리즘에 의해 만들어진 영역이 닿는 부분이다.

## 2.2 궤도추적 방법

로봇이 장애물에 닿지 않고 경로를 생성하는 가장 좋은 방법은 장애물 사이의 가장 먼 지점을 지나는 것이다. 만일 양쪽에 장애물이 있다면 두 장애물간의 거리의 절반을 지나는 것이 가장 안전한 경로라고 할 수 있다. 이러한 방법을 가지고 하나의 경로계획 알고리즘을 만들 수 있다.

3차원 장애물을 2차원으로 변형시킨 후 두 선 장

애물에서 같은 거리에 위치하는 점들의 집합으로 하나의 궤도를 만든다. 로봇의 중점은 궤도 위를 움직이며 어떤 회전도 가능하다. 그림 3은 궤도추적방법에 의해 생성된 선 장애물사이의 궤도를 보여주고 있다.



□ : Segment Point

그림 3. 궤도생성 예

각 구간마다 로봇이 움직일 수 있는 회전각이 다르게 나타나며 한 구간 안에서는 자유 회전각이 정해져 있어야 한다. 이것은 구간이 변할 때 조사되어야 하고 절점에서 가능한 회전각을 결정해야 한다. 따라서 이전 절점에서 결정한 회전각은 다음 절점에 로봇의 중심이 오기 전까지 유효하다. 단 두 선 장애물 사이의 거리는 일정하다고 가정한다.

$$x = \min_{s_1 \in S_1, s_2 \in S_2} \|s_1 - s_2\|$$

$$T = \left\{ \theta \mid \theta \in W_p \text{ and } \theta = \frac{x}{2} \right\} \quad (4)$$

로봇의 중심이 궤도에 놓일 수 없는 경우에는 경로생성이 불가능하게 된다. 그림 4에서 보는 것과 같이 로봇이 회전해야 하는 각도가  $\frac{\pi}{2}$ 보다 크거나 같을 경우 L의 길이가 로봇의 중심에서 각 꼭지점에 이은 선분의 길이 K보다 클 때 로봇의 회전이 가능하다.

$K > L$  : 회전 불가능  
 $K \leq L$  : 회전 가능

회전해야 하는 각도가  $\frac{\pi}{2}$ 보다 작을 때는 그림 5와 같은 경우이고 다음과 같은 조건으로 회전여부를 판단한다.

$$a_1 = \pi - 2\omega$$

$$\phi = a_1 + a_2$$

$$\phi \geq \frac{\pi}{2} \text{ 일 때,}$$

$K \leq L$  : 회전가능

$$\phi < \frac{\pi}{2} \text{ 일 때,}$$

$K\sin\theta \leq L$  : 회전 가능  
 $K\sin\theta > L$  : 회전 불가능

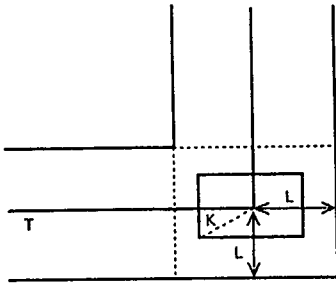


그림 4. 회전각이  $\frac{\pi}{2}$  보다 크거나 같을 때

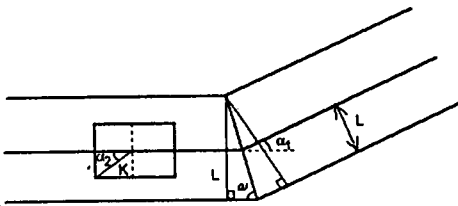


그림 5. 회전각이  $\frac{\pi}{2}$  보다 작을 때

로봇의 걸음걸이 알고리즘에 따라 회전체의 크기를 조절할 수 있으므로 알고리즘의 변화를 주어 경로를 생성해낼 수도 있다. 즉, 어느 시점에서 로봇의 크기, 즉, 걸음걸이 알고리즘을 바꾸어야 하는가를 알 수 있게 된다. 그러나 본 논문에서는 걸음걸이 알고리즘은 고정되었다고 가정한다.

## 2.2 가상장벽 방법

로봇의 회전축을 궤도추적 방법과는 달리 그림 6과 같이 O에 둔다. 회전축 O를 기준으로 회전 가능각을  $\theta$ 라하고 단계별 회전가능각을  $\theta(n)$ , 시계방향의 회전각을  $\theta_1$ , 반시계방향의 회전각을  $\theta_2$ 로 정의하면 장애물의 모양이 바뀌지 않는다고 가정했을 때 다음과 같은 식이 성립한다.

$$\theta_1(n) = \theta_1(n-1) - \xi(n-1) + \gamma \quad (6)$$

$$\theta_2(n) = \theta_2(n-1) - \xi(n-1) + \gamma \quad (7)$$

여기서,

$\xi$  : 실제 회전각 ( $\xi < \theta$ )  
 $\gamma$  : 직진시 생기는 변화각

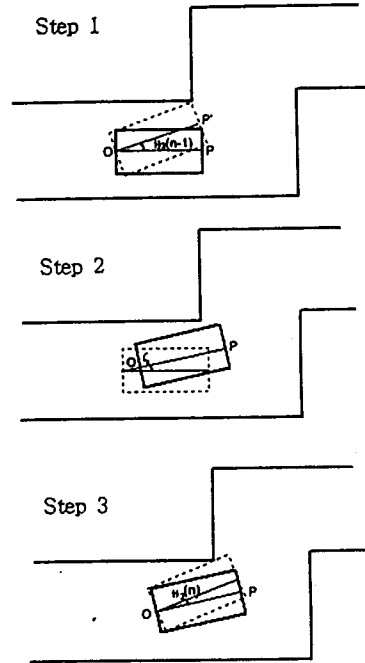


그림 6. 가상장벽 알고리즘

$\theta(n)$ 은  $\theta(n-1)$ 에 보정을 해줌으로써 이전단계에서의 장애물의 위치를 파악하고 장애물에 변화가 없거나 로봇의 위치가 변하지 않을 경우 같은 값을 가지게 된다. 그러나 장애물이나 로봇의 위치가 변하는 경우에는  $\theta(n)$ 과  $\theta(n-1)$ 사이엔 변화가 생겨 로봇은 변화하는 값에 따라 위치를 조정하게 된다. 로봇이 회전하는 실제각은  $\xi$ 로 나타나고 이 값은  $\theta$ 보다는 작은 값을 가져야 한다.  $\gamma$ 는 로봇의 위치가 이동함으로써 생기는 변화각을 나타내는데 그림 7에서 그에 관한 변화각을 나타내고 있으며 다음과 같은 식으로 구할 수 있다.  $d_1$ 에서  $d_2$ 로의 위치변화를 가지면서 변화각  $\gamma$ 는  $\phi$ 와 같은 값을 가진다.

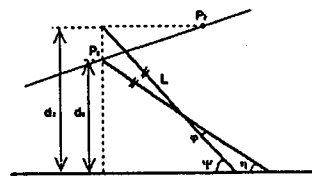


그림 7. 로봇 위치에 따른 변화각

$$\begin{aligned} \psi &= \phi + \eta \\ \gamma &= \phi \text{ 이므로} \\ \phi &= \psi - \eta \end{aligned}$$

$$\sin^{-1} \frac{d_1}{L} = \eta, \quad \sin^{-1} \frac{d_2}{L} = \psi \quad (8)$$

위 식들을 바탕으로 로봇은 다음과 같은 조건에 의해 현재위치에서 작업영역이 넓어지는 방향으로 이동하게 된다.

$$\begin{aligned} \theta_1(n) > \theta_1(n-1) - \zeta(n-1) + \gamma \text{ 이면} \\ \zeta(n) > 0 \end{aligned} \quad (9)$$

$$\begin{aligned} \theta_2(n) > \theta_2(n-1) - \zeta(n-1) + \gamma \text{ 이면} \\ \zeta(n) < 0 \end{aligned} \quad (10)$$

두 조건을 동시에 만족한 경우

$$\begin{aligned} \theta_1(n) > \theta_1(n-1) - \zeta(n-1) + \gamma \text{ 이고} \\ \theta_2(n) > \theta_2(n-1) - \zeta(n-1) + \gamma \text{ 일 때} \\ \zeta(n) = \text{sgn}[\theta_1(n) - \theta_2(n)] \end{aligned} \quad (11)$$

여기서,  $\text{sgn}[a] = \begin{cases} +1 & \text{if } a > 0 \\ -1 & \text{if } a < 0 \end{cases}$

### 2.2.1 직선통로에서의 안정도

가상장벽 방법은 직선통로 주행시 진동하는 불안정한 움직임을 보인다. 회전 가능한 각도  $\theta$ 가 로봇의 움직임에 관계없이 진동하기 때문이다. 먼저 진동의 여부를 판단해야 하려면 다음과 같은 식을 따른다.

$$\begin{aligned} \dots \theta_1(n-2), \theta_2(n-1), \theta_1(n) \dots \\ \dots \theta_2(n-2), \theta_2(n-1), \theta_2(n) \dots \end{aligned}$$

$$\begin{aligned} \theta_1(n-1) + \theta_2(n-1) = \theta_1(n) + \theta_2(n) \text{ 일 때} \\ \theta_1(n-2) = \theta_1(n) \text{ 또는 } \theta_2(n-2) = \theta_2(n) \end{aligned}$$

의 식을 만족하면 진동이라고 판단한다. 따라서 로봇의 실제 회전각  $\zeta$ 는

$$\zeta = \frac{\zeta}{c} \quad (12)$$

여기서,  $c$  : 일치한 조건의 회수

와 같이 정해진다.

### 3. 모의실험 및 결과

모의실험은 먼저 똑같이 경로를 생성할 수 있는 경우에 궤도추적 방법과 가상장벽 방법 두 가지 알고리즘이 걸리는 시간을 비교해 보았다. 동일한 장애물이 있는 작업환경에서 각각의 알고리즘을 적용해보아 생성해낼 수 있는 경로를 그림 8,9,10에 나타내었다. 실험은 Pentium-133MHz, Windows-NT 4.0 환경에서 수행하였고 알고리즘은 Visual C++

4.2 컴파일러로 프로그램 되었다. 그림에서 보아 알 수 있듯이 그림 7은 로봇의 중심에서 각 꼭지점까지의 거리가 통로길이의 절반보다 작은 경우 손쉽게 경로를 생성하는 모습을 보여주는 것이고 그림 9,10은 모두 로봇의 중심에서 각 꼭지점까지의 거리가 통로의 길이의 절반보다 큰 예로서 궤도추적 방법으로는 경로를 생성해 낼 수 없었지만 가상장벽 방법으로는 경로를 생성해 낼 수 있다는 것을 보여주고 있다. 따라서 협소한 장애물 사이를 통과할 때 유용한 알고리즘이라고 생각된다. 표 1은 각 알고리즘 별 통로의 통과시간을 보여주고 있는 때 궤도 추적방법에 비해 가상장벽 방법이 알고리즘 계산에 걸리는 시간이 매우 큰 것을 알 수 있다.

표 1. 두 가지 제안된 알고리즘 별 성능비교

알고리즘 비교치	궤도추적 방법	가상장벽 방법
시간성능	0.01Sec	0.901Sec
K < L	경로생성 가능	경로생성 가능
K > L	경로생성 불가	경로생성 가능

### 4. 결 론

본 논문에서는 일반적인 이동로봇이 생성해낼 수 없는 경로를 3차원의 장애물 환경에서 이족보행로봇에 적용했을 때 경로생성이 가능한 알고리즘을 제안하였고, 이에 적용할 수 있는 두 가지 경로계획 방법을 설계했다. 3차원환경을 2차원으로 변환시켜주는 알고리즘은 기존의 알고리즘이 이족보행로봇이 생성해낼 수 있는 경로를 장애물로 인식하는 불합리한 점을 개선하였고 그 바탕 위에 적용된 두 경로계획 방법은 장애물의 모양에 따라 최적의 경로를 생성할 수 있었다. 모의실험의 결과로 볼 때 궤도추적 방법은 알고리즘 계산시간의 이득이 있었고 가상장벽 방법은 경로생성의 이득이 있었다.

### [참 고 문 헌]

- [1] Latombe, J. C. "Robot Motion Planning", Kluwer Academic Publisher, 1991.
- [2] Yun-Hui Liu, Suguru Arimoto, "Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles", The Int. Journal of Robotics Research, Vol 11, No. 4, pp. 376-382, 1992.
- [3] Jason A. Janet, Ren C. Luo and Michael G. Kay, "The Essential Visibility Graph: An Approach to Global Motion Planning for Autonomous Mobile Robots", IEEE Int. Conf. on Robotics and Automation, pp. 1958-1962, 1995.

[4] Yun-Hui Liu, Suguru Arimoto, "Proposal of Tangent Graph and Extended Tangent Graph for Path Planning of Mobile Robots", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 312-317, 1991.

[5] 민승기, 김대원, "오목한 장애물이 많은 작업환경에서 ETG를 이용한 이동로봇의 경로계획 알고리즘의 설계", 대한전기학회 로보틱스 및 자동화 연구회, pp. 60-63, 1997.

[6] Yuan F. Zheng and Fred R. Sias, Jr., "Design and Motion control of Practical Biped Robots", Int. J. of Robotics and Automation, 1988.

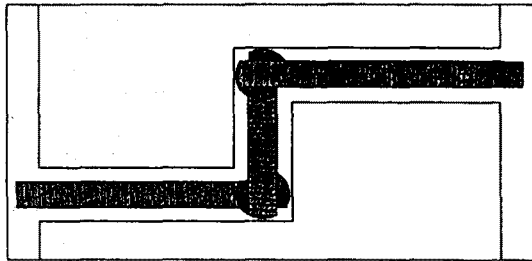


그림 8. K가 L 보다 작은 경우

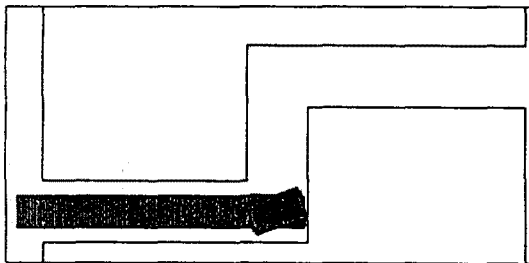


그림 9. K가 L보다 큰 경우

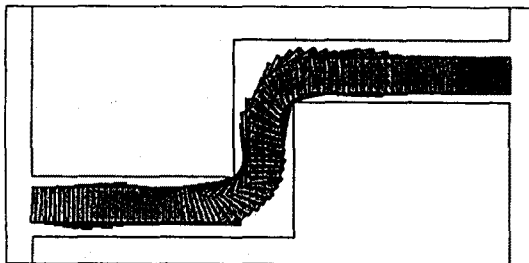


그림 10. 가상장벽 방법 ( $K > L$ )