

비전 시스템에 기초한 이동 로봇의 장애물 회피

남문호\*, 박태준\*, 김민수\*\*, 정찬수\*

\*숭실대학교 전기공학과, \*\*세종-특히드마틴 우주항공연구소

Obstacle Avoidance Behavior of the Mobile Robot based on the Vision System

\*Mun-Ho Nam, \*Tae-Jun Park, \*\*Min-Soo Kim, \*Chan-Soo Chung

\*Dept. of Electrical Eng. Soongsil Univ.

**Abstract** - 최근 로봇 기술의 발전으로 인해 로봇에 관련된 제반 기술에 대한 연구가 활발히 진행되고 있다. 이러한 연구 중의 하나가 이동 로봇이 스스로 장애물을 감지하여 회피하도록 적절한 이동 경로를 생성해주는 연구이다.

따라서 본 논문에서는 이동 경로 상에 존재하는 장애물을 회피하기 위한 방법으로 비전 시스템에 기반하여 주행 거리 및 주행 시간을 최소화하여 목표점에 도달할 수 있도록 장애물을 회피하는 방법에 대하여 연구하였다. 시뮬레이션을 통해 제안한 방법의 성능을 검증하여보았다.

1. 서 론

기존의 로봇은 그 작업 공간의 한계로 그 쓰임의 한계가 제한되었고, 특정 작업에만 사용 되었으나 최근 들어 이동 로봇에 대한 연구가 활발해 진행됨에 따라 산업 분야 뿐만 아니라 사무실이나 가정 등 그 활용의 범위가 광범위 해지고 있다. 특히, 이동 로봇은 넓은 범위에서 작업 수행이 가능하므로 고장 설비의 점검 및 수리, 우주에서의 정보 수집, 원자력 시설의 점검 등 인간이 행하기 어려운 작업을 대신 할 수 있는 장점이 있다.

또한, 최근에는 건물내의 자동화 서비스를 제공하기 위해 이동 로봇이 스스로 판단하여 경로 생성 및 장애물 회피를 목적으로 하는 이동 로봇의 적용 범위가 확대되고 있다. 이의 한 형태인 자율 주행에 관한 연구로서는 경로상에 존재할 수 있는 장애물제외의 충돌을 회피하여 최적의 경로를 찾아가는 경로 계획(path planning)에 대한 연구가 진행되고 있다.

장애물 회피를 위한 지금까지의 논문들에서는 정적(static)환경에서 자율 주행 로봇이 움직인다고 가정되어 연구가 진행되어 왔다. 그 중 대표적인 방법이 포텐셜 함수(potential function)을 이용한 것이다[1]. 그러나 로봇의 제어 시스템이 안정함에도 불구하고, 포텐셜 함수를 사용한 방법은 바람직하지 않은 국부 최소값(local minima)을 가질 수 있다. 이 문제를 해결하기 위하여 몇 가지의 연구가 시도 되었다. 장애물에 대한 C-space를 설정한 후 포텐셜력 의해 경로를 결정하고 그 후 전체적인 경로를 고려한 Global Path Planning[2], 포텐셜력을 설정할 때 수치적으로 위치장을 설정한 Numerical Potential Field Techniques[3], 그리고 장애물로부터 공차를 고려하여 포텐셜력을 이용한 Path Transforms[4] 등이 있다.

한편 자율 주행 로봇이 동적(dynamic)환경에서 움직이는 경우는 포텐셜 함수 방법으로는 비효율적이다[5]. 즉, 장애물들이 움직임에 따라 계산에 드는 비용이 커지기 때문이다. 이것은 주로 동적인 환경에서 자율 주행 로봇의 안정한 위치는 계속적인 계산을 필요하기 때문이다. 그러므로 동적인 환경에서의 주행 경로를 설계하기 위한 여러 가지 방법들이 제시되었다.

주행 공간에 지표값을 설정한 spatial indexing[6], Hopfield 형식의 신경망 회로를 사용한 Neural Network Dynamics[7], 로봇이 장애물 보다 빠른 경우 장애물의 움직임을 예측하여 Accessibility Graph를 작성하는 방법[8] 등이 있다.

여러 가지 회피 방법 중 본 논문에서는 비전 시스템을 기반으로 이동 경로 상에 존재하는 장애물을 회피하여 목표점까지의 주행거리 및 주행 시간을 최소화하여 목표점에 도달하는 방법을 연구하였다.

2. 비전 시스템

비전 시스템은 CCD 카메라에서 나오는 NTSC 방식의 신호를 프레임 그레버(frame grabber)를 통하여 PC에서 사용할 수 있도록 변환한다.

변환된 정보는 하나의 2D 영상으로 나타나며, 이러한 영상을 해석함으로써 원하는 정보를 얻는다. 본 논문에서는 가장 일반적인 위치 추정 방법인 영상에서 물체의 기울어진 각도를 측정하고 고정된 높이의 값을 이용하여 상대 위치를 계산한다.

비전 시스템의 정확도에 따라 이동 로봇의 위치 추종 성능이 좌우되며, 동적 환경에서 이동 로봇의 위치 제어를 정확하게 하기 위하여 비전 시스템을 사용하였다.

3. 장애물 회피 방법

3.1 장애물의 위치 및 속도

주행 환경에서 장애물과 자율주행 로봇과의 관계는 그림 1와 같다.

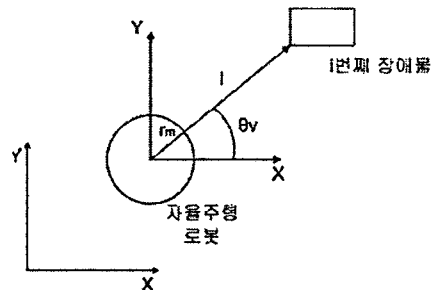


그림 1. 자율 주행 로봇과 장애물

장애물의 절대위치를 계산하기 위해서는 먼저 로봇과 장애물간의 상대위치를 계산하여야 한다. i번째 장애물의 상대위치는 식 (1)와 같다.

$$x_{pi/m} = (r_m + \ell) \cos\theta_v \quad (1)$$

$$y_{pi/m} = (r_m + \ell) \sin\theta_v$$

식 (1)에서  $r_m$ 은 자율주행 로봇의 몸체 반경,  $\ell$ 은 로봇과 장애물과의 거리,  $\theta_v$ 는 비전 센서가 자율주행로봇에 부착된 위치의 절대각도를 나타낸다. 그러므로 장애물의 절대위치는 식 (2)로 표현된다.

$$x_{pi} = x_m + x_{pi/m} \quad (2)$$

$$y_{pi} = y_m + y_{pi/m}$$

식 (2)에서  $x_m$ 은 로봇 중심점의 x축 절대위치,  $y_m$ 은 로봇 중심점의 y축 절대위치를 나타낸다.

장애물의 절대속도 계산도 같은 방법으로 장애물의 상대속도부터 계산하면 식 (3)과 같다.

$$v'_{pix/m} = \frac{x'_m - x_{pi}}{h} \quad (3)$$

$$v'_{piy/m} = \frac{y'_m - y_{pi}}{h}$$

식 (3)에서  $h$ 는 제어기의 샘플링 시간,  $x'_m$ 은  $h$ 초 후의  $i$ 번째 장애물 x축 상대좌표,  $y'_m$ 은  $h$ 초 후의  $i$ 번째 장애물 y축 상대좌표를 나타낸다.

그러므로 장애물의 절대속도는 식 (4)과 같이 표현할 수 있다.

$$v_{pix} = v_{mx} + v_{pix/m} \quad (4)$$

$$v_{piy} = v_{my} + v_{piy/m}$$

식 (4)에서  $v_{mx}$ 는 로봇의 x축 방향의 절대속도,  $v_{my}$ 는 로봇의 y축 방향의 절대속도를 나타낸다. 장애물의 절대위치와 절대속도를 계산하였으므로 장애물과 자율주행 로봇의 충돌 가능성을 검사할 수 있다.

### 3.2 충돌 가능성

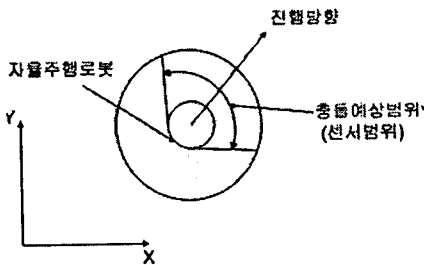


그림 2. 장애물 충돌 가능성 예측

센서 영역 중에서 충돌 가능 영역은 그림 2와 같다. 센서의 범위를  $R_{sensor}$ 라하고  $v_m$ 을 자율주행 로봇의 주행 속도라 하면 현재의 위치에서 센서의 범위까지 자율주행 로봇이 주행하는데 걸리는 시간은 식 (5)과 같이 나타낼 수 있다.

$$t_{sensor} = \frac{R_{sensor}}{v_m} \quad (5)$$

그러므로 자율주행 로봇은  $t=0$ 에서  $t=t_{sensor}$ 까지 로봇과 장애물을 가상적으로 움직이면서 그 거리가  $D_w/2$ 보다 작게 되는 장애물을 찾는다.

### 3.3 충돌 예상점

#### 3.3.1 고정 장애물

고정 장애물일 경우 장애물의 절대 속도는 0이 된다.

그러므로 고정 장애물과 로봇과의 충돌 예상점은 고정 장애물의 절대 위치가 되며, 이 경우 충돌 예상점은 식 (6)과 같이 표현할 수 있다.

$$x_{pi.c} = x_{pi} \quad (6)$$

$$y_{pi.c} = y_{pi}$$

#### 3.3.2 이동 장애물

이동 장애물의 충돌 예상점을 계산하기 위하여 먼저 장애물의 진행 방향을 식 (7)로부터 구하여야 한다.

$$\theta_{pi} = \text{atan}\left(\frac{v'_{piy}}{v'_{pix}}\right) \quad (7)$$

장애물의 속도와 장애물의 진행 방향은 일정하다고 가정하므로, 장애물의 주행 경로는 식 (8)과 같은 직선의 방정식으로 나타낼 수 있다.

$$y - y_{pi} = \tan\theta_{pi} \times (x - x_{pi}) \quad (8)$$

자율주행 로봇의 주행 경로도 직선의 방정식으로 나타내면 식 (9)과 같다.

$$y - y_m = \tan\psi \times (x - x_m) \quad (9)$$

식 (8), (9)로부터 두 직선의 충돌 예상점을 구하면 식 (10)와 같다.

$$x_{pi.c} = \frac{x_m \tan\psi - x_{pi} \tan\theta_{pi} + y_{pi} - y_m}{\tan\psi - \tan\theta_{pi}} \quad (10)$$

$$y_{pi.c} = (x_{pi.c} - x_{pi}) \tan\theta_{pi} + y_{pi}$$

### 3.4 임시 목표점

#### 3.4.1 고정 장애물

고정 장애물일 경우 목표점, 장애물, 그리고 로봇의 현재 위치를 기하학적으로 고려하면 임시 목표점은 식 (11)과 같이 표현된다.

$$x_t = x_{de} + \text{sgn}(\Delta\psi) \times \frac{D_w}{2} \times \sin\psi_g \quad (11)$$

$$y_t = y_{de} - \text{sgn}(\Delta\psi) \times \frac{D_w}{2} \times \cos\psi_g$$

식 (11)에서

$$\psi_{de} = \text{atan}\left(\frac{y_{de} - y_m}{x_{de} - x_m}\right)$$

$$\psi_g = \text{atan}\left(\frac{y_g - y_m}{x_g - x_m}\right) \quad (12)$$

$$\Delta\psi = \psi_{de} - \psi_g$$

$$\text{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

식 (12)에서  $(x_{de}, y_{de})$ 는 가장 가까운 충돌 예상점,  $(x_g, y_g)$ 는 목표점의 좌표를 나타낸다.

#### 3.4.2 이동 장애물

이동 장애물의 경우 임시 목표점을 정하는 방법을 2가지 제안하였다.

첫 번째 방법은 자율주행 로봇이 이동 장애물보다 속도가 빠른 경우 장애물의 진행 방향으로 장애물을 앞질러 가는 방법으로 식 (13)과 같이 표현된다.

$$\text{if } \Delta V > 0 \quad x_t = x_{de} + \frac{D_w}{2} \times \cos\theta_{de} \quad (13)$$

$$y_t = y_{de} + \frac{D_w}{2} \times \sin\theta_{de}$$

두 번째 방법은 자율주행 로봇이 이동 장애물보다 속도가 느릴 경우 장애물의 반대 방향으로 속도를 줄이면서

서 돌아가는 방법으로 식 (14)와 같이 표현된다.

$$\text{if } \Delta V \leq 0 \quad x_t = x_{de} - \frac{D_u}{2} \times \cos\theta_{de} \quad (14)$$

$$y_t = y_{de} - \frac{D_u}{2} \times \sin\theta_{de}$$

식 (13)과 (14)에서  $\Delta V = V_m - V_{de}$ 이고,  $(x_{de}, y_{de})$ 는 가장 가까운 충돌 예상점이고,  $\theta_{de}$ 는 이동 장애물의 진행 방향이다.

#### 4. 시뮬레이션

##### 4.1 시뮬레이션 방법

자율주행 로봇의 주행 환경은 1000mm×1000mm의 정사각형의 형태이다. 시뮬레이션에서 거리의 기본 단위는 [cm] 이고 각도의 기본단위는 [radian]이다. 그러므로 자율주행 로봇의 주행 환경 좌표계는 시뮬레이션 환경에서 좌측 아래가 (0, 0) 이고 우측 위가 (100, 100)으로 정해진다. 자율주행 로봇의 폭( $D_u$ )은 75mm이므로, 좌표상에서 자율주행 로봇과 이동 장애물 간에 거리가 80mm 안에 들어 올 경우 충돌에 위험이 발생한다. 장애물의 감지하기 위한 센서의 측정 범위는 200mm로 하였다.

먼저, 고정 장애물의 경우 일정 속도(20m/sec)로 주행하다가 이동 장애물을 감지할 경우 속도를 이동 장애물의 속도보다 빠르게(30m/sec)하는 방법과 속도를 이동 장애물의 속도보다 느리게(10m/sec)하는 방법으로 시뮬레이션을 하였다.

이동 장애물의 경우도 위와 동일한 방법을 적용하였으며, 이동 장애물의 움직이는 속도(20m/sec)와 방향을 일정하다고 가정하였다.

마지막으로 고정 장애물과 이동 장애물이 함께 있는 경우는 고정 장애물의 경우 빠른 속도로 장애물을 회피한 후 뒤에 이동 장애물을 회피하는 방법은 위와 동일한 방법으로 시뮬레이션 하였으며, 4m/sec당 한 픽셀이 출력되게 하였다.

##### 4.2 시뮬레이션 결과

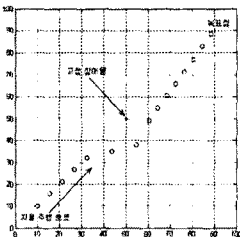


그림 3. 고정 장애물1

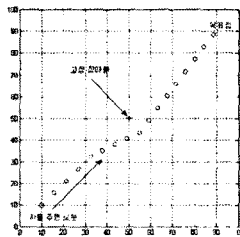


그림 4. 고정 장애물2

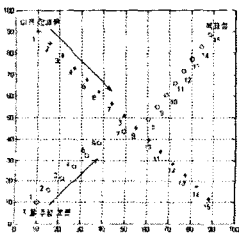


그림 5. 이동 장애물1

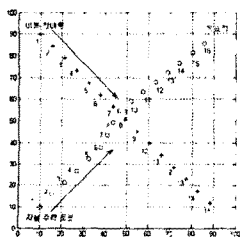


그림 6. 이동 장애물2

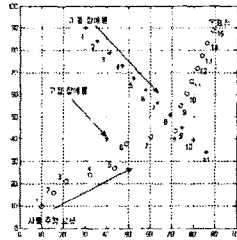


그림 7. 종합 장애물1

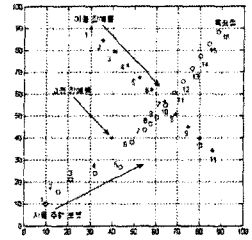


그림 8. 종합 장애물2

#### 5. 결 론

본 논문에서는 자율 주행 로봇이 고정 장애물과 이동 장애물, 종합 장애물을 회피하여 목표점을 향하여 주행할 때, 주행 거리와 주행 시간을 최소화하는 방법을 제시하였다.

고정 장애물인 경우, 이동 장애물인 경우, 종합 장애물인 경우 모두 속도를 빠르게 주행하면 주행 시간을 최소화 되었고, 속도를 느리게 주행하면 주행 거리가 최소화 되었다. 그러므로 적절한 주행 속도와 주행 거리를 최적화 시키는 것이 후후 연구과제이다.

#### [참 고 문 헌]

- [1] E. Rimon and D.E. Koditschek, "Exact Robot Navigation Using Artificial Potential Functions," IEEE Transactions on Robotics and automation, vol. 8, No. 5, pp.501-518, 1992.
- [2] C.W. Warren, "Global Path Planning Using Artificial Potential Fields," IEEE International conference on Robotics and Automation, pp.316-321, 1989.
- [3] J. Barraquand, B. Langlois and J.C. Latombe, "Numerical potential field techniques for robot path planning," IEEE Transactions on Systems, Man, and Cybernetics, pp.224-241, 1992.
- [4] Z. Alexander, "Using Path transforms to Guide the Search for Findpath in 2D," The international Journal of Robotics Research, vol 13, No. 4, pp.315-325, 1994.
- [5] 남문호, 김민수, 정찬수, "포텐셜 함수를 이용한 자율주행 로봇의 장애물 회피에 관한 연구," 정보 및 제어 심포지엄 논문집, pp.266-268, 2005.
- [6] K. Fujimura and H. Samet, "Path planning among moving obstacles using spatial indexing," ICRA, pp.1662-1667, 1988.
- [7] R. Glasius, A. Komoda and S.C.A.M. Gielen, "Neural Network Dynamics for Path Planning and Obstacle Avoidance," Neural Networks, vol. 8, No. 1, pp.125-133, 1995.
- [8] K. Fujimura and H. Samet, "Time-optimal Paths among moving obstacles," IEEE International Conference on Robotics and Automation pp.1110-1115, 1989.