

자율이동로봇을 위한 경로제어에 관한 연구

조성배*, 박경훈, 이양우
 동의대학교 전기공학과

A Study on the Trajectory Control of a Autonomous Mobile Robot

Sung-Bae Cho*, Kyung-Hun Park, Yang-Woo Lee

Abstract - A path planning is one of the main subjects in a mobile robot. It is divided into two parts. One is a global path planning and another is a local path planning. This paper, using the formal two methods, presents that the mobile robot moves to multi-targets with avoiding unknown obstacles. For the shortest time and the lowest cost, the mobile robot has to find a optimal path between targets. To find a optimal global path, we used GA(Genetic Algorithm) that has advantage of optimization. After finding the global path, the mobile robot has to move toward targets without a collision. FLC(Fuzzy Logic Controller) is used for local path planning. FLC decides where and how faster the mobile robot moves. The validity of the study that searches the shortest global path using GA in multi targets and moves to targets without a collision using FLC, is verified by simulations.

1. 서 론

최근 이동로봇은 그 유용성으로 인하여 활용이 계속 증대되고 있다. 미지의 환경계척이나 열악한 환경의 탐험, 또는 서비스 등이 그것이다. 이러한 임무의 완수를 위해 로봇은 자신의 진행경로와 방향, 위치 등을 인식하고 있어야 하며 환경의 변화에 대한 대처방안이 필요하게 된다. 이를 위해 이동로봇에 관한 많은 연구가 진행 중이며 이들은 경로계획(path planning)과 위치보정, 장애물회피(obstacle avoidance) 등으로 이루어져 있다[1][2].

그 중 경로계획은 국지적 경로계획(local path planning)과 전역 경로계획(global path planning)으로 나눌 수 있는데 국지적 경로계획은 미지의 환경에서 로봇의 적응을 목적으로 장애물 회피와 더불어 센서를 통한 변화하는 환경의 인식을 통해 사전지식 없이 목표지점까지 주행하는 것이다[3][4]. 전역 경로계획은 목표지점까지의 전체경로를 설정하는 것으로 기지의 공간에서 off-line으로 최단경로를 찾아내는 것이다.

이상의 경로계획을 바탕으로 본 논문에서는 기지의 환경에서 다중 목표지점을 가지는 이동로봇의 경로계획을 세우고자 한다. 사용자가 요구하는 여러 지점으로 물건의 운송이나 전달을 위해 로봇은 다양한 경로의 목표지점들을 가지게 되는데 효율적인 주행을 위해 다중 목표지점들을 최단으로 연결해주는 전역 경로계획이 필요하게 된다. 본 논문에서는 수치최적화의 강점이 있는 GA(Genetic Algorithm)를 이용하여 가장 효율적인 경로를 설정하였다. GA를 이용한 경로계획은 이전의 많은 연구가 있으나[5], 여기서는 지도를 격자로 설정하지 않고 단순한 목표지점들의 좌표를 조합하고 평가함으로서 전체적인 경로를 설정해 주게 된다. 정해진 경로는

기지의 환경에서 다음 목표지점만을 정해줌으로 로봇의 이동은 국지적 경로계획을 통해 이루어진다. 로봇의 주행 중 지도상에 반영되어 있지 않은 장애물의 회피와 목표지점까지의 주행을 위해 본 논문에서는 FLC(Fuzzy Logic Controller)를 이용하였다. 목표지점과 로봇의 위치, 장애물의 위치를 고려하여 적절한 바퀴의 속도를 결정함으로써 로봇은 장애물을 회피하고 목표지점으로 이동한다.

2. 이동로봇의 주행

본 연구에서 사용된 이동로봇은 그림 1에서와 같이 두 바퀴의 속도차를 이용하여 조향각을 결정하고 이동하는 differential steering 방식의 로봇이다.

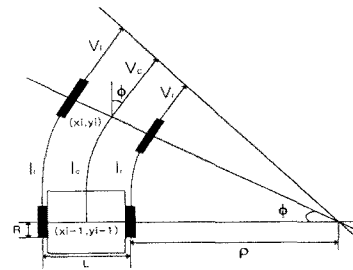


그림 1. 이동로봇의 주행
 Fig. 1. Movement of a mobile robot

로봇중심의 속도는 식 (1)에서 보인 바와 같이 각 바퀴 속도의 평균이며 곡률반경 ρ를 이용하여 이동하는 지점의 좌표를 구할 수 있다.

$$V_c = \frac{V_r + V_l}{2} \tag{1}$$

$$\rho = \frac{L(V_r + V_l)}{2(V_r - V_l)} \tag{2}$$

여기서 Vr은 왼쪽 바퀴의 속도이고 Vl은 오른쪽 바퀴의 속도, Vc는 로봇중심의 속도이다.

이 때 조향각 φ는 중심의 이동거리 lc와 곡률반경을 이용해 결정할 수 있는데 현재의 위치에서 샘플링 시간 동안의 좌표이동은 식 (3)을 이용하여 구한다.

$$\begin{aligned} x_i &= (\rho + L/2)(1 - \cos \phi) + x_{i-1} \\ y_i &= (\rho + L/2)\sin \phi + y_{i-1} \end{aligned} \tag{3}$$

$$\begin{bmatrix} X_i \\ Y_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} \cos \phi_i & -\sin \phi_i & 0 \\ \sin \phi_i & \cos \phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ \phi_i \end{bmatrix} \tag{4}$$

새로운 좌표 (xi, yi)는 로봇의 현재좌표계(current frame)에서 위치이므로 이를 식 (4)를 이용해 고정좌표계(fixed frame)의 좌표로 변환해야 한다. 이동로봇

은 목표지점까지 $[X_i, Y_i, \theta]$ 를 이용하여 로봇자신의 위치와 각도를 인식하게 되고 자기유도주행(dead reckoning)으로 이동하게 된다.

3. 경로제어

3.1 전역 경로계획

다중 목표지점을 가지는 기지의 환경에서 각 목표지점 간을 최단으로 연결해 주는 전역경로를 찾기 위해 유전자 알고리즘을 사용하여 경로를 설정하였다. 28개의 목표지점을 가지는 기지의 공간이 그림 2에 표시되어져 있다. 그림에서 목표지점을 원으로 나타내었고 유전자 알고리즘의 적용을 위해 목표지점에 번호를 부여하였다.

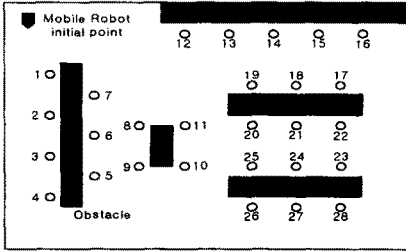


그림 2. 사용된 기지의 환경 지도
Fig. 2. The map of known environment

지도를 격자로 구성하지 않았고 유전자 알고리즘은 선택된 목표지점만을 진행순서대로 순환하게 나열함으로써 실제 진행경로까지 지정하지는 않는다. 따라서 로봇은 모든 장애물의 정확한 위치를 요구하지 않으며 목표지점의 위치와 장애물을 고려했을 때 이들 사이의 거리관계를 필요로 하게된다.

이것은 TSP(Traveling Salesman Problem)와 같은 순서조합 최적화의 문제로써 기존 유전자알고리즘의 연산자들과 같은 방식으로는 해의 적법성 문제로 해결에 어려움이 있다. 본 논문에서는 경로를 표시하는 각 스트링을 순서조합문제에 적합한 기호코딩(symbolic encoding)법을 사용하여 나타내었다. 임의의 목표지점이 지정되었을 때 목표들의 위치는 기호로 사용된 정수로 사상되어진다. 전역경로계획에 사용된 주요 연산을 다음에 나타내었다.

교배연산 방법은 부분사상교배(partially mapped crossover)를 사용하였다. 사상된 두 개의 스트링이 식 (5)와 같을 때 교배구간이 교환되고 나머지 유전자들은 교환된 유전자들의 사상관계에 따라 변환되게 된다. 교배 후의 스트링의 변화를 식 (6)에 나타내었다.

$$s_c^i = [1 \ 2 \ \dots \ 5 \ 6 \ 7 \ \dots \ n]$$

$$s_u^i = [6 \ 5 \ \dots \ 1 \ n \ 2 \ \dots \ 7] \quad (5)$$

$$s_c^{i+1} = [5 \ 7 \ \dots \ 1 \ n \ 2 \ \dots \ 6]$$

$$s_u^{i+1} = [n \ 1 \ \dots \ 5 \ 6 \ 7 \ \dots \ 2] \quad (6)$$

돌연변이연산자로는 역치(Inversion)를 사용하였다.

$$s_c^i = [1 \ 2 \ \dots \ 5 \ 6 \ 7 \ \dots \ n]$$

$$s_u^{i+1} = [1 \ 2 \ \dots \ 7 \ 5 \ 6 \ \dots \ n] \quad (7)$$

식 (7)처럼 역치연산자는 정해진 돌연변이 구간에서 유전자들의 순서를 바꾸는 역할을 한다. 이 두 연산은 순서의 중복 없이 해를 찾기 위해 사용되어졌다. 그 외,

엘리트 전략을 도입하여 최상의 개체가 보존되도록 하였으며 각 개체의 평가는 식 (8)에서처럼 각 목표지점간의 장애물을 고려한 거리의 합으로 하였다.

$$eval(s_c) = \frac{1}{\sum_{i=1}^{n-1} dist\{s_c(i), s_c(i+1)\} + dist\{s_c(n), s_c(1)\}} \quad (8)$$

(dist : 거리합수)

구해진 경로는 로봇의 현재위치와 각도를 고려하여 편리한 방향으로 순서를 유지하며 재배치되게 된다.

3.2 국지적 경로계획

이동로봇이 전역 경로계획을 통해 결정된 순서대로 각 목표지점들을 순환하는 것은 실제 주행경로의 지정이 없으므로 국지적 경로계획에 따라 FLC를 이용하게 된다.

첫 번째 목표점은 목표지점들이 순환경로이기 때문에 로봇의 초기위치에서 각도와 거리를 기준으로 선택하게 하였으며 목표지점들을 모두 순환하였을 때는 다음 명령을 고려해 초기위치로 돌아가 대기하는 것을 생각하여 최종 기각지는 초기위치가 된다.

FLC는 장애물의 탐색에 따른 센서의 입력과 현재의 위치를 고려하여 목표지점으로의 빠른 주행을 위해 적절한 회피경로를 설정하게 된다. 그림 3에 국지적 경로계획을 위한 제어기의 블록도를 나타내었다.

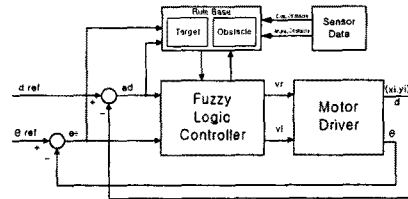


그림 3. 국지적 경로계획을 위한 제어기의 블록도
Fig. 3. Block diagram of controller for local path planning

퍼지입력으로는 현재 목표지점의 거리와 방향, 센서를 통한 장애물의 거리와 방향으로 하였으며 출력은 각 바퀴의 속도로 두었다. 삼각형과 사다리꼴 함수를 적절히 병행하여 멤버십 함수를 구성하였고 그림 4에 목표지점의 추종을 위한 입,출력 멤버십 함수를 나타내었다.

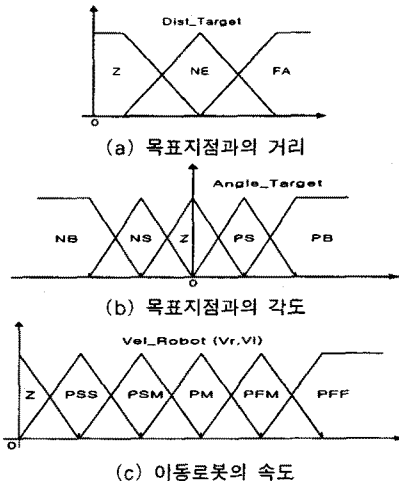


그림 4. 입출력변수의 멤버십 함수
Fig. 4. Membership function of input, output variables

표 1에 목표지점 추종을 위해 사용된 거리와 각도에 차에 따른 두 바퀴의 출력을 나타내었다. 최종출력은 무

계중심법을 이용하여 구동부의 입력으로 된다.

표 1. 목표지점과의 거리와 각도오차에 따른 룰 베이스
Table 1. Rule Base of distance and angle error between target and a mobile robot

Angle _{target} \ Dist _{target}	Z		NE		FA	
	V _l	V _r	V _l	V _r	V _l	V _r
NB	Z	Z	PM	PS	PFM	PFM
NS	Z	Z	PSM	PS	PFM	PM
Z	Z	Z	PSM	PSM	PFM	PFM
PS	Z	Z	PSS	PSM	PM	PFM
PB	Z	Z	PSS	PM	PFM	PFM

Z(Zero), NE(Near), FA(Far), NB(Negative Big), NS(Negative Small), PS(Positive Small), PB(Positive Big), PSS(Slow), PSM(Slow-Medium), PM(Medium) PFM(Fast-Medium), PFF(Fast)

FLC를 이용한 이동로봇의 주행은 실제 주행경로의 지정이 없기 때문에 전면에만 장애물이 감지된 경우, 주행 방향의 설정이 어렵게 된다. 이러한 경우 목표지점과 현재위치의 가장 가까운 거리로 연결해주는 장애물의 꼭지점을 검색하여 이동하는 방법을 이용하였다.

4. 모의실험

모의실험에서 사용된 지도는 그림 2와 같은 28개의 목표지점을 가지는 공간으로 그림 2에 나타난 번호를 이용하여 목표지점을 임의로 선택하였다. GA 연산에서 각 스트링의 유전자 수는 지정된 목표지점의 수로 하였고 개체수는 200개로 두었으며 세대수는 유전자 수에 비례하도록 설정하였다. 교배연산확률 P_c 와 돌연변이연산확률 P_m 은 각각 0.7과 0.2로 두었다.

그림 5와 6은 각각 목표지점이 11개와 12개로 GA에 의해 최적화된 목표지점을 순환하고 초기위치로 돌아가는 궤적을 나타낸 것이다. 검은 원은 선택된 목표지점을 나타낸다.

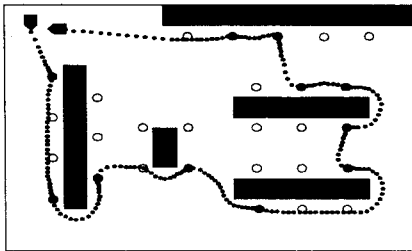


그림 5. 목표지점이 {1 4 5 10 13 14 17 18 22 23 26} 일 때의 이동 궤적

Fig. 5. Trajectory with target points {1 4 5 10 13 14 17 18 22 23 26}

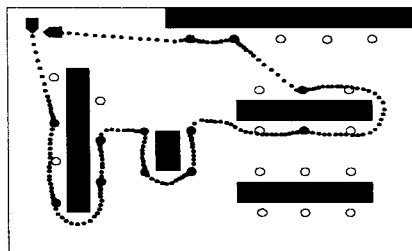


그림 6. 목표지점이 {2 4 5 6 8 9 10 11 12 13 18 21} 일 때의 이동 궤적

Fig. 6. Trajectory with target points {2 4 5 6 8 9 10 11 12 13 18 21}

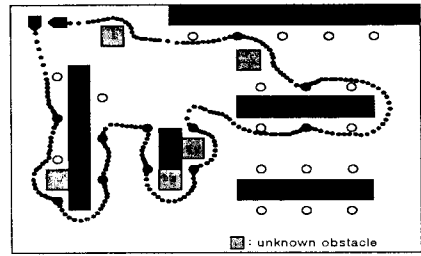


그림 7. 미지의 장애물이 있는 경우

Fig. 7. The case of being unknown obstacles

그림 5에서 목표지점은 {1 4 5 10 13 14 17 18 22 23 26}로 두었으며 최적화된 순서는 {1 4 5 10 26 23 22 17 18 14 13}으로 나타났다. 그림 6에서 목표지점은 {2 4 5 6 8 9 10 11 12 13 18 21}이었으며 최적화된 목표지점의 조합은 {2 4 5 6 8 9 10 11 21 18 13 12}이었다.

그림 7은 두 번째 모의실험과 비슷하나 미지의 장애물이 나타났을 때의 경로를 도시한 것이다. 이 때 목표지점은 {2 4 5 6 8 9 10 11 13 18 21}의 10개로 두었고 최단거리의 조합은 {2 4 5 6 8 9 10 11 21 18 13}로 나타났다.

모든 모의실험에서 non-slipping 조건과 pure rolling 조건을 만족한다고 가정하였으며 지도상의 장애물과 미지의 장애물 모두 고정된 것으로 하였다.

5. 결 론

본 연구는 다중 목표지점을 가지는 기지의 공간에서 여러 목표지점을 최단거리로 연결하여 효율적으로 주행하기 위한 경로계획을 제안하였고 이를 모의실험을 통해 확인하였다. 전역 경로계획으로 GA를 사용하여 각 목표지점들의 최적화된 순서를 찾아내었으며 국지적 경로계획으로 FLC를 사용하여 환경의 변화에 적응하게 하였다. GA는 순서조합에 적합한 기호코딩법과 이에 따른 연산자를 사용하였으며 퍼지변수로는 장애물과 목표지점의 위치, 각도를 이용하였다. 제안한 방법으로 모의실험을 실시한 결과 목표지점들의 효율적인 조합문제와 이들의 추종문제, 미지의 장애물이 있는 경우에 모두 만족할 만한 결과를 나타냄으로서 본 연구의 유용성을 확인할 수 있었다. 향후 과제로서는 기지의 공간내에서 이동장애물 발견시의 경로계획과 작업환경이 미지의 공간으로까지 확장되었을 때 제안한 방법의 확장 문제와 이를 실제 로봇에 적용하여 유효성을 확인해 보는 것이다.

[참 고 문 헌]

- [1] C. W. Warren, "Global Path Planning using Artificial Potential Field," Proc. IEEE int. Conf. on Robotics and Automation, pp. 316-321, 1989.
- [2] J. A. Janet, R. C. Lou, and M. G. Kay, "Autonomous Mobile Robot Global motion planning geometric beacon collection using traversability vectors," IEEE Trans. Robotics and Automat., vol. 13, no. 1, pp. 132-140, February, 1997.
- [3] A. Fujimori, P.N. Nikiforuk and M. M. Gupta, "Adaptive Navigation of Mobile Robots with Obstacle Avoidance," IEEE Trans. on Robotics & Automat., vol.13, no. 4 1997.
- [4] J. Borenstein, Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots," IEEE Trans. Syst., Man, Cybern., vol. 19, no. 5, pp. 1179-1187, 1989.
- [5] K. Sugihara, "GA Based On-line Path Planning for SAUVIM," Lecture Notes in Artificial Intelligence, vol. 1416, pp. 329-338, Springer-Verlag, 1998.