다족 모바일 로봇의 최적 경로 생성을 위한 3D 시뮬레이터의 개발

*김기우, *최우창, *유영국, **김진걸
*인하대학교 전기공학과 석사과정
**인하대학교 전자전기공학부 교수

Development of a 3D simulator for optimal path generation of a mobile multiped robot

*Ki-Woo Kim, *Woo-Chang Choi, *Young-Kuk Yoo, **Jin-Geol Kim
*School of Electrical Engineering, Inha University
**School of Electrical Engineering, Inha University

Abstract - This paper deals with generating multi-ped mobile robot's optimal path and its simulation. The multi-ped robot has six-legs which make it possible to move actively by attached driving wheel at the end of legs. The simulation environment is created similarly to the indoor environment as simple obstacles and walls. Also simulator can reconstruct an simulation environment. In this paper, the suggested simulator can generate the optimal path from starting point to destination by applying the A* algorithm and Bug2 algorithm. Then it is possible to check algorithms as 3D screen and we can simulate under the generated path.

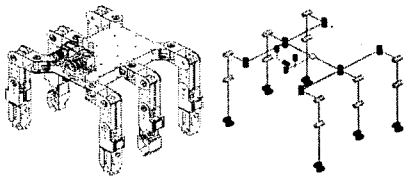
1. 서 론

로봇은 다리를 가진 보행로봇과 바퀴를 가진 주행로봇으로 크게 나뉠 수 있다. 보행로봇은 크게 2족 보행로봇과 다족 보행로봇으로 나뉘며 대표적인 2족 보행로봇은 ASIMO[1]가 있으며 다족 보행로봇으로는 AIBO[2]가 있다. 본 실험에 사용된 로봇은 6족의 다리를 가지며 4개의 바퀴로 구동되는 로봇으로 평탄한 바퀴로 구동을 할 수 있으며, 평탄하지 않거나 장애물이 많은 지형에서는 다리로 보행을 하는 효율적인 구조로 제작되어 있다. 본 논문에서의 실험은 로봇의 주행기능을 이용하여 목표점까지의 좌표까지의 경로를 생성하여 모의 주행하는 것이다. 로봇이 장애물의 좌표를 알고 있다는 가정 하에서 최적 경로를 생성하기 위하여 A*알고리즘[3]을 사용하였다. A*알고리즘은 여러 가지 탐색알고리즘 중 효율적인 알고리즘으로 여러 문제해결을 위해 사용되는데, 특히 게임에서와 같이 여러 장애물이 있는 지도상에서 목표까지의 경로를 계획하는데 효과적으로 사용된다. 또한 장애물의 좌표를 모른다는 가정 하에 초음파 센서의 정보를 바탕으로 로봇이 목표점까지의 경로를 추종하는 Bug2알고리즘[4]을 사용하여 본 시뮬레이터의 성능을 평가하였다. 상기 2가지 알고리즘을 사용하여 로봇이 계획된 최적경로를 바탕으로 주행하게 되며 시뮬레이션을 통하여 결과를 확인하였으며 본 모의실험에서는 바퀴의 슬립이 없다는 가정 하에 실험을 실시하였다.

2. 본 론

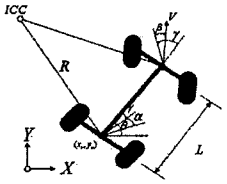
2.1 로봇의 기구적 구성

본 실험의 모델이 된 로봇인 ITIBO는 총 6개의 다리 중 4개만이 구동용 바퀴가 장착되어 있으며 주행에는 사용되지 않는다. 각 다리에는 4개의 주행용 모터가 장착되어 있으며 다리 종단부분은 2축으로 되어있어 개별제어가 가능하며 여러 동작을 구현할 수 있다. 그림 1은 CAD를 이용한 3D 모델링과 각 관절의 구조를 나타내고 있다.



<그림 1> ITIBO의 3D 모델링과 기구적 구조

주행로봇에 있어서 조향은 필수적인 요소이다. 본 로봇에서는 Ackerman steering을 사용하여 조향을 하였다. 본래 Ackerman steering은 자동차 조향에 쓰이는 장치로 모바일로봇에 많이 쓰이는 조향방식이나 미끄러짐이 많은 방식이다. 본 실험에서는 슬립이 적은 Double Ackerman steering 방식을 사용하였다. 본 로봇은 4개의 휠에 z축으로 모두 자유도가 있는 구조로 설계되어 있다. 즉 4개의 바퀴가 개별 동작할 수 있는 구조로 설계되어 있다.



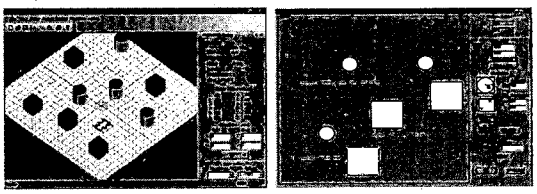
<그림 2> 로봇의 기구적 모델

그림 2는 로봇의 기구적 모델을 나타낸 그림으로써 앞바퀴와 뒷바퀴의 슬립각인 α와 β는 선형가속도와 로봇의 상대적인 뒷바퀴와 앞바퀴 사이의 각이나 본 실험에서는 슬립을 배제 하였다. 또한 x_r과 y_r은 상대적인 뒷바퀴의 좌표이며 각도 θ는 x축에 대한 오리엔테이션이다. L은 앞 축과 뒤 축 사이의 거리이며 로봇의 선형가속도는 V이며 조향 각은 γ로 정의된다. 로봇의 bi-steerable wheel은 앞쪽 휠 각 γ 와 반대편에서의 뒤쪽 휠 각 γ 가 같으며 슬립이 없다는 가정 하에 로봇의 기구학 모델은 식(1)과 같다.

$$\begin{bmatrix} \dot{x}_{ES} \\ \dot{y}_{ES} \\ \dot{\theta}_{ES} \end{bmatrix} = \begin{bmatrix} V\cos(\theta-\gamma) \\ V\sin(\theta-\gamma) \\ \frac{V\sin(2\gamma)}{L\cos(\gamma)} \end{bmatrix} \quad (1)$$

2.2 3D 시뮬레이터 구성

3D 시뮬레이터는 크게 시뮬레이션 창과 맵 에디터 창으로 나뉘어져 있으며 시뮬레이션 창은 OpenGL기반의 3D화면으로써 맵 에디터 창은 윈도우에서 제공하는 GDI(Graphics Device Interface)로 구성되어있다. 아래의 그림 3은 시뮬레이터 창 및 맵 에디터 창이다.



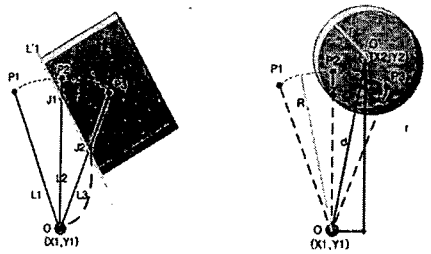
<그림 3> 시뮬레이션 창 및 맵 에디터

시뮬레이터 창은 Visual C++ 6.0으로 작성이 되었으며 GUI(Graphic User Interface)를 제공한다. 기본적으로 알고리즘 변환, 센서정보 디스플레이가 가능하고 맵 에디터와 연동하여 2D에서 맵을 작성 후 파일로 저장이 가능하며, 저장된 파일은 시뮬레이터에서 불러 사용을 할 수 있다.

2.3 장애물 검출 및 센서 모델링

장애물은 크게 사각형 및 원으로 근사화 될 수 있으며 검출원리는 다음과 같다. 그림 4의 사각형 검출원리는 L'1을 직선의 방정식으로 표현되며 식(2)와 같은 형식이 된다. 센서의 L1, L2 그리고 L3도 식(2)와 같이 표현이 되므로 연립하여 교점을 계산한다. 여기서 a는 직선의 기울기 이며 b는 y절편이다. 계산된 교점을 기반으로 O에서 J2까지의 거리는 식(3)을 이용하여 계산 할 수 있으며 여기서 (x_1, y_1)와 (x_2, y_2)는 각 점의 좌표이다. 원의 검출원리는 두 점 O와 O'의 거리는 식 (3)과 같이 표현되며 r은 장애물 원의 반지름이며 R은 센서의

측정거리이다. 여기서 d값에 따라서 원과 센서와의 관계를 기술할 수 있다. $d > r+R$ 이면 원과 접하지 않은 상태이며 $d = r+R$ 이면 원과 접합의 의미하며 $d < r+R$ 이면 충돌의 의미한다.



<그림 4> 사각형 및 원 검출원리

$$y = ax + b \tag{2}$$

$$d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \tag{3}$$

식(4)는 각 센서의 좌표를 나타내는 식이며 오프셋 및 센서의 각도를 나타낸다. 시뮬레이터 상에서 초음파 센서는 전방에 3조가 장착되어 있으며 좌우 측면에 각 1조씩 모두 5개를 사용하였으며 위치는 표 1과 같다.

$$\left(\sin(\frac{\deg \times \pi}{180})*offset\,,\ \cos(\frac{\deg \times \pi}{180})*offset\right) \tag{4}$$

<표 1> 센서의 위치

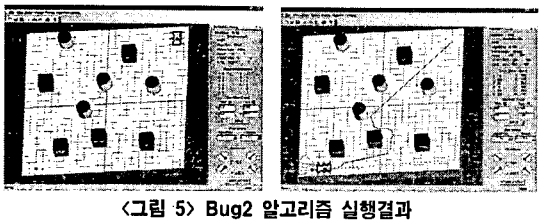
센서번호	deg(degree)	offset(cm)	센서(x,y)좌표
1(전방)	0	55	(0, 55)
2(좌 전방)	330	55	(-27.63, 47.55)
3(우 전방)	30	55	(27.48, 47.63)
4(왼쪽)	270	25	(-24.99, -0.059)
5(오른쪽)	90	25	(25.00, -0.099)

2.4 주행 알고리즘

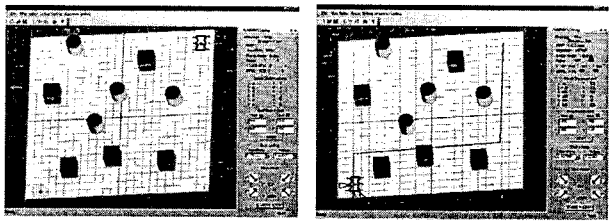
본 시뮬레이터에서 사용한 주행알고리즘은 A*와 Bug2알고리즘이다. A*알고리즘은 맵 크기 및 장애물의 위치를 알고 있을 때 사용하는 알고리즘이며 Bug2알고리즘은 시작점과 목표점만 알고 장애물의 위치를 모를 때 사용하는 알고리즘이다. A*알고리즘은 게임에 많이 사용되는 알고리즘으로써 다음과 같이 $'f = g + h$의 계산 값을 사용하여 다음 이동할 경로를 결정한다. 여기서 g는 goal로서 시작노드부터 이 노드까지 오는데 드는 비용이며 h는 heuristic으로 이 노드에서 목표까지 가는데 드는 추정 비용이다. f는 fitness로서 g와h의 합이고 이 노드를 거쳐 가는 경로의 비용에 대한 최선의 추측을 의미한다. f값이 낮을수록 이 경로가 최단 경로일 가능성이 크다. Bug2알고리즘은 장애물을 회피하여 목적지까지 최단거리를 설정하여 목적지까지 도달하는 알고리즘으로서 비교적 간단한 알고리즘에 속한다. 목적지까지 최단거리의 가상의 직선을 설정하여 이를 기준으로 목표점까지의 방향을 설정하고 만약 진행 방향에 장애물이 있으면 다시 가상의 직선에 닿을 때까지 벽 추종주행을 한다. 장애물을 벗어나면 목적지 방향으로 주행하고 목표점 도달까지 이를 반복하는 알고리즘이다.

2.5 실험

실험은 같은 시작좌표에서 출발하여 같은 도착좌표까지 2개의 알고리즘을 비교하는 것으로 진행하였다. 출발좌표는 (500, 500)이며 도착 좌표는 (-500, -500)으로 하였으며 로봇의 속도는 0.5m/s이다. 로봇의 효율을 산정하기 위하여 출발 시점부터 도착 시점까지의 소요된 시간을 계산하였고 이동거리를 산출하였다. 아래의 그림 5는 Bug2알고리즘을 사용하였을 때의 결과이고 그림 6은 A*알고리즘을 사용하였을 때의 결과이다.



<그림 5> Bug2 알고리즘 실행결과



<그림 6> A* 알고리즘 실행 결과

각각의 알고리즘에 의해 탐색된 경로의 효율(η)를 계산하기 위하여 식(5)을 이용하여 효율을 계산하였다.

$$\eta = \frac{A^*(t) - Bug2(t)}{Bug2(t)} \times 100 \tag{5}$$

여기서 $A^*(t)$는 A*알고리즘의 소요시간이며 $Bug2(t)$는 Bug2 알고리즘의 소요시간이다. 표2는 각각 다른 위치에서 식(5)을 이용하여 5번 실행하여 평균값을 구했을 때 Bug2알고리즘이 A*알고리즘보다 효율이 약 3.93% 더 높게 측정이 되었다. 표 2는 다른 시작좌표에서 5번 실험한 값이다.

<표 2> 알고리즘별 로봇의 소요시간과 이동거리

실험	시작좌표	알고리즘	소요시간 (ms)	이동거리 (m)	효율 (%)
1	x: 500 y: 500	Bug2	38.02	19.01	6.8822
		A*	40.83	20.415	
2	x: -500 y:500	Bug2	28.12	14.06	28.4733
		A*	39.35	19.675	
3	x: 0 y: 0	Bug2	32.83	16.415	9.1085
		A*	36.12	18.06	
4	x: 500 y:-500	Bug2	19.4	9.7	-33.7009
		A*	14.51	7.255	
5	x: 200 y: 200	Bug2	43.02	21.51	8.9138
		A*	47.23	23.615	

3. 결 론

본 실험은 OpenGL기반의 3D 시뮬레이터를 사용하여 주행로봇에 Bug2 및 A*알고리즘을 포팅하여 각 알고리즘의 효율을 알아보았다. A*알고리즘을 보면 특징점만 잡아 경로생성을 하여 직각인 경로가 많아짐을 알 수 있고 Bug2알고리즘은 경로 주행을 하기 때문에 원형인 경로가 많다. 또한 시작지점에 따른 알고리즘의 효율을 보면 비슷한 것을 알 수 있으나 Bug2의 알고리즘이 효율이 3.93%더 높게 나온 결과를 확인할 수 있었다. 본 연구는 카메라를 장착하여 장애물 인식 및 경로생성주행을 하기 위한 전 단계의 연구 중 경로생성 부분에 관한 연구이며 추후 카메라를 장착하여 이를 응용하여 연구를 진행하여야 할 것이다.

후 기

이 논문은 2007년도 산업자원부의 지원에 의하여 기초전력공학공동연구소 (R-2007-2-059) 주관으로 수행된 과제임.

[참 고 문 헌]

[1] Sakagami Y., Watanabe R., Aoyama C., Matsunaga S., Higaki, N. and Fujimura, K., "The intelligent ASIMO: system overview and integration", International Conference on Intelligent Robots and System IEEE/RSJ, Vol. 3, 2478 - 2483, 5 Oct., 2002

[2] Fujita M., "On activating human communications with pet-type robot AIBO", Proceedings of the IEEE, Vol. 92, 1804 - 1813 ,11 Nov., 2004

[3] Sankaranarayanar A. and Vidyasagar M., "Path planning for moving a point object amidst unknown obstacles in a plane: a new algorithm and a general theory for algorithm development", Decision and Control, Proceedings of the 29th conference, Vol. 2, 1111 - 1119, 5 Dec., 1990

[4] Hui, Y.C., Prakash, E.C. and Chaudhari, N.S, "Game AI: artificial intelligence for 3D path finding", IEEE Region 10 Conference, Vol. 2 306 - 309, 21 Nov., 2004