

# 뱀형 로봇에 대한 이동궤적과 장애물 회피 시뮬레이션

## Locomotion of Snake Robot and Obstacle Avoidance Simulation

이지우\*, 이창훈, 김용호

J.W. Lee, C.H. Lee, Y.H. Kim

\* 배재대학교 전자공학과(전화:(042)520-5663, 팩스:(042)520-5773, E-mail : lucky77777@hanafos.com)

**요약** : 뱀형 로봇은 일반적인 바퀴형 이동로봇과 운동 메카니즘이 상이하며 다관절로 이루어져 있기 때문에 장애물 회피에 있어 빠른 정보의 처리와 이를 위한 특별한 정보가 요구된다. 이를 실현하기 위하여 로봇은 자신의 위치를 지속적으로 파악하면서 장애물의 좌표 값과 일정한 거리의 간격을 두고 움직여야 한다. 주행 궤도 및 장애물 회피를 위한 알고리즘을 검증하기 위하여 가상 뱀형 시뮬레이터를 제작하였다. 시뮬레이터는 이동 주행 궤도를 생성하고, 지나온 궤도를 재현할 수 있는 재현기(Back Tracker), 앞으로 이루어질 뱀형 로봇의 위치와 자세를 알아보는 예견기(Predictor)로 구성된다. 시뮬레이터를 통하여 주위의 장애물을 안전하게 통과할 수 있는 일반적인 알고리즘인 포텐셜함수의 특성을 알아보고, 국소 최소점(Local Minima)에 빠지기 쉬운 단점을 극복하기 위한 방안을 제시한다. 본 논문에서는 뱀의 이동 주행 궤적을 알아보고, 주위의 장애물을 안전하게 통과할 수 있도록 하는 알고리즘에 대한 고찰과 제안한 알고리즘을 소프트웨어적인 3D 시뮬레이션을 통하여 결과를 분석하고 검증한다.

**Keywords** : Snake Robot, Simulation, Local Minima, Obstacle avoidance, Potential function, Convex hull

### I. 서론

지면위에서 동작하는 바퀴형 로봇은 엔진에 의한 회전을 통하여 움직이게 된다. 이러한 바퀴형 로봇은 부드러운 표면에 적합하며, 높은 지형을 오르거나 이동하기에는 부적절하다. 이러한 이유로 다관절 로봇에 대한 연구가 활발하게 진행되고 있다. 다관절 로봇에 속하는 뱀형 로봇은 나무를 오르고, 물위를 지나가며, 작은 관속을 통과하거나, 좁은 공간을 지나가는 운동 메카니즘을 지니고 있다. 이러한 운동 메카니즘으로 인하여 활용도가 매우 다양하며, 여러 가지 방향으로 연구 되고 있다.

본 논문에서는 뱀형 로봇의 움직임에 대한 운동 메카니즘을 알아보고 가상 시뮬레이션을 통하여 결과를 분석하고 검증한다. 뱀형 로봇에 대한 주행을 알아보고, 한 단계 높은 수준인 유닛 스스로 장애물 피하면서 출발지점에서 목표지점으로 이동하는 방법을 결정하도록 하는 방법에 대하여 알아본다. 이를 구현하기 위해서는 복잡한 장애물을 간단한 장애물로 재구성해야 할 필요성이 있다. 본 논문에서는 뱀형 로봇이 장애물을 인식하기 쉽도록 재구성하는 알고리즘에 대한 고찰과 실험을 통하여 검증한다. 본 논문의 결과를 분석하고 검증하기 위하여 3차원 시뮬레이터를 제작하였다. 뱀형 로봇을 위한 시뮬레이션 환경 프로그램인 뱀형 시뮬레이터는 이동 주행 궤도를 생성하고, 지나온 궤도를 재현하는 재현기(Back Tracker), 앞으로 이루어질 뱀형 로봇의 위치와 자세를 알아보는 예견기(Predictor)로 구성되었다. 시뮬레이터의 기능에 대하여 알아보고 본 논문에서 제안한 장애물 재구성 방법을 이용하여 장애물 회피 시뮬레이션 결과에 대한 분석과

검증을 통하여 결론을 맺는다.

### II. 뱀형 로봇의 주행 시뮬레이션

#### 2.1 뱀형 로봇의 주행

뱀형 로봇은 다관절 로봇으로 다른 바퀴형 로봇과 다른 운동 메커니즘을 가지고 있다. 뱀형 로봇의 주행에 사용된 알고리즘은 다음과 같다.

#### 뱀 모양 곡선(Serpennoid Curve)

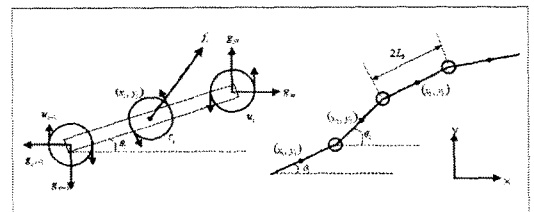


그림 1. 뱀형 로봇 다이어그램

x-y평면 위의 임의의 점(x, y)가 다음과 같은 곡선위에 있으면 이 곡선을 뱀 모양 곡선이라고 정의하며 곡선의 방정식은 식(1)과 같다.

$$x(s) = \int_0^s \cos(\xi_\sigma) d\sigma, y(s) = \int_0^s \sin(\xi_\sigma) d\sigma$$

$$\xi_\sigma := a \cos(b\sigma) + c\sigma \quad (1)$$

여기서 a, b는 스칼라이고, s는 원점에서 임의의 점까지의 호의 길이이다. 식(1)을 미소 호의 길이에 대하여 미분하면 식(2)과 같다.

$$\frac{dx}{d\sigma} = \cos(\xi_\sigma), \quad \frac{dy}{d\sigma} = \sin(\xi_\sigma)$$

$$\tan \theta_i = \frac{dx}{dy} / \frac{dy}{d\sigma} = \cos(\xi_\sigma) / \sin(\xi_\sigma) \quad (2)$$

이것을  $i$ 번째 절대 편절각  $\theta_i$ 을 구하면,  $\theta_i = \xi_\sigma$ 이다. 이 식으로부터 상대 각도값을 구하면 식(3)이 된다.

$$\phi_i = \frac{d\theta_i}{d\sigma} = a \sin(b\sigma) + c \quad (3)$$

위의 식(3)을 이용하여 뱀모양 움직임을 얻을 수 있다.  $\phi_i$ 의 입력에 의해 로봇 전체가 움직이는 속도는 다음 식에서 얻을 수 있다.

$$v := \dot{w}_x \cos \xi + \dot{w}_y \sin \xi \quad \xi := \frac{1}{n} \sum_{i=1}^n \theta_i \quad (4)$$

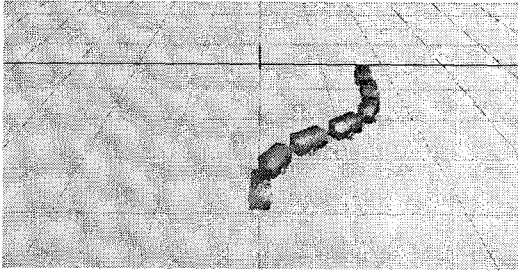


그림 2.뱀의 움직임을 묘사한 시뮬레이션

## 2.2 포텐셜 함수를 이용한 장애물 회피

이동 로봇은 어떤 환경 하에서 미리 알고 있는 장애물과 알지 못하는 장애물을 만나게 된다. 센서를 이용하여 새로운 장애물이 관측되었을 경우 이 로봇이 장애물을 피해 원하는 목적지까지 도달할 수 있는 일반적인 방법이 포텐셜(Potential)함수를 이용한 방법이다.

포텐셜 함수:

$$\text{direction} = \frac{\text{goal}_{loc} - \text{curr}_{loc} + \sum_n \left[ \frac{\text{curr}_{loc} - \text{obst}_{loc} \times k}{|\text{curr}_{loc} - \text{obst}_{loc}|^p} \right]}{\quad} \quad (5)$$

이러한 국지적 경로계획(Local Path Planning)은 이동로봇이 최적의 경로를 얻을 수 없다는 단점을 가지고 있으며, 국소 최소점(Local Minima)에 빠지게 되는 단점이 있다.

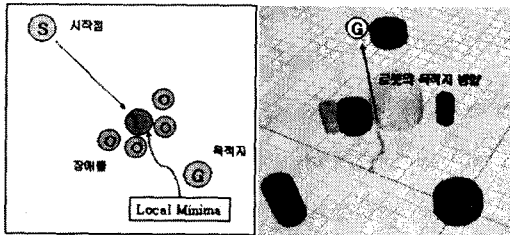


그림 3. 로봇이 국소 최소점에 빠진 모습

위의 그림에서처럼 국소 최소점에 빠지는 단점을 보완하기 위하여 본 논문에서는 장애물 재구성에 관한 알고리즘을 제시한다.

## III. 국소 최소점 극복을 위한 알고리즘

본 논문에서는 포텐셜 함수의 단점인 국소 최소점에 빠지는 단점을 보완하기 위하여 아래와 같은 알고리즘을 제안하였다. 아래의 그림은 국소 최소점에 빠지는 장애물의 일부 모형이다.

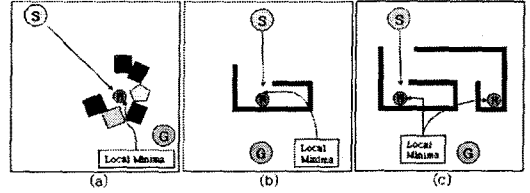


그림 4. 국소 최소점의 다양한 구조

본 논문에서 제안하는 알고리즘은 모든 장애물의 정보를 알고 있다는 가정을 통하여 제안한다. [그림 4]의 (a)는 장애물들이 모여 하나의 큰 U자형 장애물을 형성하고 있어 국소 최소점에 빠지게 되는 경우이다. 본 논문에서는 이동로봇이 국소 최소점에 빠지지 않기 위해서 다음과 같은 방법을 제안한다.

불록 다각형 장애물의 특성은 포텐셜함수를 통하여 물 흐르듯이 지나갈 수 있기 때문에 convex hull을 이용하여 불록 다각형을 만드는 방법을 사용하였다. convex hull이란 정점들의 부분집합을 연결하는 다각형을 의미한다. convex hull은 원본 객체를 완전하게 둘러싸는 가장 간단한 다각형에 의해서 복잡한 다각형 모양의 형상을 표현하는데 사용된다. 그림에서 보느냐와 같이 각각의 다각형 장애물에 대하여 꼭지점 정보를 취득한다. 취득한 꼭지점 정보를 가지고 모든 장애물을 포함하는 불록 다각형을 만들기 위한 convex hull을 찾는다. convex hull을 찾는 알고리즘에는

그림 5. convex hull의 검색  
Pac-  
kage Wrapping,  
Grah-  
am Scan 방법 등  
이 있다. 본 논문에서는 R. L. Graham에 의해 제안된  
Graham Scan방법을 사용하였다.

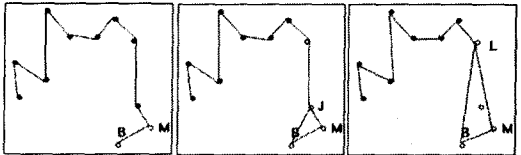


그림 6. Graham Scan방법

Graham Scan방법은 방위각을 이용하여 그리고 angular sorting 단계로부터 정렬된 점집합의 스캔(scan)으로 진행된다. B점에서 시작하여 점들의 연속하는 삼각형을 조사한다. 만일 연속하는 삼각형  $P_1P_2P_3$  내각이  $\Pi$ 값을 넘어서게 되면, 정점  $P_2$ 는 제거되고 삼

각형  $P_0P_1P_3$ 이 조사된다. Graham Scan 방법은 시작점인 B점과 조우할 때 종결된다. convex hull을 이용하여 복잡한 다각형의 모습에 대하여 간단화된 하나의 다각형으로 재구성한다.

[그림 4]의 (b)는 여러 개의 블록이 모여 만들어진 장애물의 형태이다. 본 장애물의 경우는 국소 최소점에 들어가기 전에 장애물을 우회하여 장애물을 회피하여야 한다. 이러한 경우 [그림 4]의 (a)에서 사용된 convex hull의 확장을 통하여 간단화된 다각형으로 재구성 된다. 아래의 그림은 시작점의 변화에 따라 장애물의 재구성을 보여주고 있다.

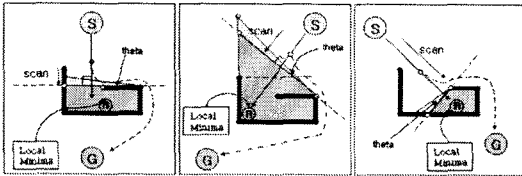


그림 7. 스캔을 이용한 블록 다각형 구성

[그림 7]에서 보는 바와 같이 시작점이 변경됨에 따라 재구성된 블록 다각형의 크기변화가 일어난다. 로봇의 방향과 위치에 따라 블록다각형의 모양이 변함으로써 기본적인 틀(기존의 장애물)은 변함이 없고, 로봇의 이동 비용을 줄이기 위한  $\theta$ 값에 의해서 블록 장애물이 변형된다. 본 알고리즘의 구현은 다음과 같은 절차를 따른다.

- 1) 로봇의 방향에 수직성분인 직선의 방정식 생성
- 2) 로봇의 방향에 따라 선분의 방정식 스캔(서로 다른 두개의 블록으로 구성된 좌표점 검색)
- 3) 로봇의 이동 비용 값 적용( $\theta$ )

[그림 4]의 (c)는 뱀형 로봇이 장애물을 통과할 수 있는 경우이다. 이와 같은 경우 앞 절에서 사용한 convex hull을 이용하여 두개의 단절된 블록을 하나의 장애물로 인식하여 우회할 수 있지만, 본 논문에서는 로봇이 블록 사이를 지나갈 수 있다는 가정을 통하여 두개의 장애물로 구성된 길을 따라 이동하는 방법을 제안한다.

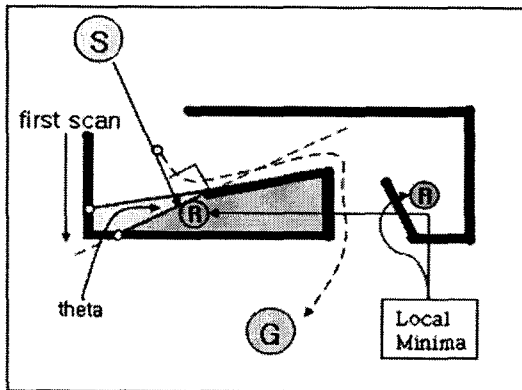


그림 8. 한번의 스캔을 통하여 적용시키는 사례

위의 그림은 [그림 4]의 (c)번 그림과 유사한 형태를 가지고 있는 장애물의 모습이다. 단지 국소 최소점에 빠질 확률을 증가시킨 그림이다. 포텐셜함수는 물이 높은 곳에서 낮은 곳으로 이동하는 방식과 같기 때문에 첫 번째 장애물을 지나치게 되면 두 번째 장애물과는 상관없이 목표점에 도달할 수 있다. 만약 이러한 시뮬레이션을 하기 위해서는 일반화된 방식을 통하여 두 번째 장애물에 대해서도 같은 방식으로 재구성 하여도 무방하다. 이와 같은 사례를 아래 그림을 통하여 설명한다.

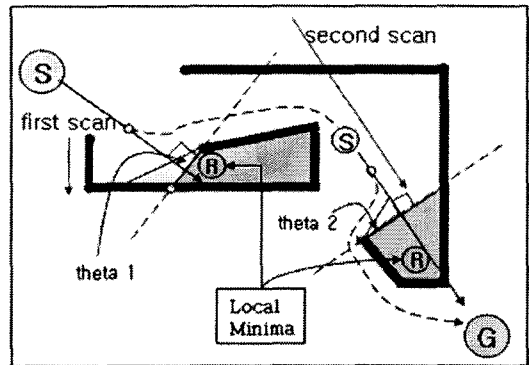


그림 9. 두 번의 스캔을 적용시킨 사례

앞 절과 조금 달라진 곳은 시작점과 목적지가 달라 졌다는 것과 시작점의 위치에서 로봇이 국소 최소점에 빠지기 쉽도록 블록을 재구성한 것이다. 위의 그림에서 보듯이 첫 번째 블록을 재구성한 장애물을 우회하였을 경우 로봇은 목적지를 향하여 새로운 방향을 가지게 되며, 로봇은 두 번째 국소 최소점에 빠지게 된다. 이러한 경우 두 번째 블록에 또 한번의 스캔을 통하여 재구성된 장애물을 생성하여 로봇은 목적지에 도달할 수 있게 된다.

본 논문에서는 포텐셜 함수를 이용하여 시뮬레이션 하기 위한 장애물의 재구성 방법에 대하여 제안하였다. 포텐셜 함수의 단점이 국소 최소점에 쉽게 빠진다는 단점 때문에, 다른 이동 알고리즘들과 같이 사용하여 우회하는 경우가 많다. 하지만, 본 논문에서 제안한 알고리즘은 포텐셜함수의 특성은 살리지만, 장애물의 특성을 변형 하여 이동 로봇이 최적의 궤적을 통하여 이동할 수 있는 방안을 제시하였다.

#### IV. 시뮬레이션 결과 및 분석

앞 절에서 제안된 뱀형 로봇의 주행과 장애물 회피를 위한 장애물 재구성 알고리즘을 구현하기 위하여 뱀형 시뮬레이터(Snake Simulator)를 제작하였다. 뱀형 로봇의 시뮬레이터는 3차원 인터페이스를 사용하였다. 3차원을 구성하기 위해 OpenGL(Graphic Library)를 사용하였으며, 지형 및 장애물을 담당하는 클래스와,

로봇의 움직임을 제어하는 클래스 그리고 사용자와 대화하는 클래스로 크게 세 가지로 나뉘어 진다.

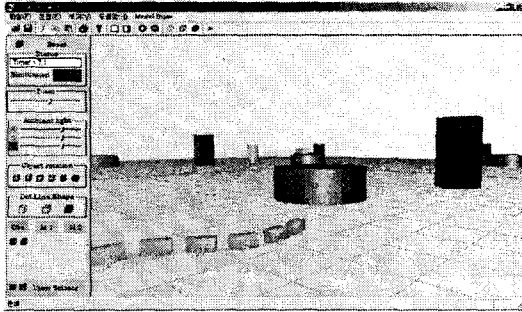


그림 10. 뱀형 시뮬레이터의 구성모습

지형 및 장애물을 관리하는 클래스 CObstacle은 동적으로 지형을 생성 및 제거 가능하다. 일반적인 장애물은 부피를 가지므로 둘러싸는 가상의 구를 이용하여 처리하였으며, 복잡한 형태의 장애물은 블록들의 합성을 통하여 표현하였다. 로봇의 움직임을 제어하는 클래스 CSnakeModel은 뱀형 로봇의 주행에 대한 제어를 한다. 움직임 제어에 있어 실제 동작시간 값을 활용한다. 사용자와 대화가 가능하도록 하는 유저 인터페이스(User Interface)를 담당하는 클래스 CSimRobotFormView으로 구성된다. 시뮬레이터는 장애물 지도를 이용하여 시뮬레이션 하며, 지도의 정보를 이용하여 객체의 정보를 획득한다. 본 논문에서 제안한 알고리즘을 통하여 장애물 재구성 시뮬레이션 결과 아래의 그림과 같이 장애물을 회피하여 목적지에 도달하였다.

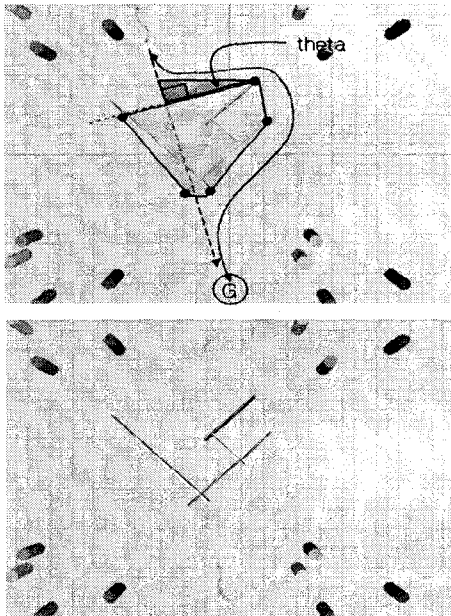


그림 11. 뱀형 로봇의 시뮬레이션 결과

## V. 결 론

본 논문에서는 뱀형 로봇의 이동체계에 대한 고찰과 포텐셜 함수를 통하여 장애물을 회피하여 이동하기 위한 장애물의 재구성에 대하여 알아보았다. 포텐셜 함수의 단점인 국소 최소점에 빠지는 단점을 보완하기 위하여 최적의 블록 다각형으로 재구성하는 방법을 사용하였다. 이전에 만들어진 장애물은 복잡하고 컴퓨터가 이해하기 어려운 형태를 가지고 있지만, 이를 convex hull을 통하여 단순화 시킬 수 있는 기능을 이용하였으며, 로봇이 지나쳐 가는 부분에 대하여 부분적인 블록 다각형을 생성하여 로봇이 지나쳐 갈수 있도록 장애물 재구성에 대한 방안을 제시하였다. 본 논문에서 사용된 알고리즘과 제어방법들을 가상 시뮬레이터를 통하여 그 유효성 및 타당성을 검증하였다. 실제 세계에서는 보다 다양한 장애물이 있으며, 예상치 않은 상황이 발생하게 된다. 현재 시뮬레이션하기 위해서 사용된 블록들의 기능은 컴퓨터가 인식하기 쉽도록 구성하였기 때문에 복잡한 블록들을 사용하여 테스트 하는 과정이 필요할 것이라 생각되며, 실제 이동 로봇 시스템에 장애물을 검출하는 PSD소자 거리센서를 이용하여 장애물을 검출하고, 본 논문에서 제안한 장애물 재구성을 통하여 이동 로봇이 실제로 회피할 수 있도록 하는 연구과정이 필요할 것이라 생각된다.

## 참 고 문 헌

- [1] Masashi Saito, Masakazu Fukaya, and Tetsuya Iwasaki, "Serpentine Locomotion with Robotic Snakes" IEEE Control Syst. Mag, pp. 64/81, February 2002
- [2] Shugen MA, Wen J.LI, Yuechao WANG, "A Simulator to Analyze Creeping Locomotion of a Snake-like Robot", in Proc 2001 IEEE Int. Conf. On Robotics and Automation, pp. 3656/3661
- [3] S. Ma, Analysis of Snake Movement Forms for Realization of Snake-like Robots, in Proc. 1999 IEEE Int. Conf. on Robotics and Automation(ICRA'99), pp.3007/3013, 1999
- [4] M. Raibert, Legged Robots that Balance. Cambridge, MA:MIT Press, 1986.
- [5] B. Klaassen and K. Paap, "GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control, "in Proc. IEEE Int. Conf. Robot. Auto. Detroit, MI, 1999, pp. 3014/3019.
- [6] Gray and H. Lissmann, "The kinetics of locomotion of the grass-snake", J. Exp. Biol., vol. 26, no. 4, pp. 354/367, 1950.