

비용함수와 서브 골을 이용한 비선형 최적화 방법 기반의 이동로봇 장애물 회피 주행

Mobile Robot Navigation with Obstacle Avoidance based on the Nonlinear Least Squares Optimization Method using the Cost Function and the Sub-Goal Switching

정 영 중* · 김 곤 우†
(Young-Jong Jung · Gon-Woo Kim)

Abstract - We define the mobile robot navigation problem as an optimization problem to minimize the cost function with the pose error between the goal position and the position of a mobile robot. Using Gauss-Newton method for the optimization, the optimal speeds of the left and right wheels can be found as the solution of the optimization problem. Especially, the rotational speed of wheels of a mobile robot can be directly related to the overall speed of a mobile robot using the Jacobian derived from the kinematic model. When the robot detects the obstacle using sensors, the sub-goal switching method is adopted for the efficient obstacle avoidance during the navigation. The performance was evaluated using the simulation and the simulation results show the validity of the proposed method.

Key Words : Mobile Robot, Navigation, Nonlinear Least Squares, Sub-Goal, Obstacle Avoidance

1. 서 론

이동로봇의 경로계획이란 로봇이 원하는 목적지점까지 장애물과의 충돌 없이 도착할 수 있게 해주는 경로를 찾는 과정을 의미한다. 또한 이동로봇의 경로 추종이란 경로 계획 과정을 통해 찾아진 경로를 이동로봇이 최소한의 오차를 보이면서 따라가도록 제어하는 과정을 의미한다. 이동로봇을 위한 경로계획을 진행하고자 할 때 고려해야 할 사항들로 이동로봇의 경로와 장애물과 충돌가능성, 목적지까지 가는 최단경로, 빠른 시간 내에 경로계획 과정 수립 등등이 있으며 이러한 고려사항들을 만족시키고자 하는 다양한 경로계획 방법들이 제안되었다.

버그 알고리즘[1]은 로봇의 형태를 공간상의 한점으로 가정한 상태에서 가장 직관적이고도 기본적인 알고리즘이며 로드맵기반 경로 계획[2]은 공간상에 로봇이 안전하게 달릴 수 있는 특별 도로를 의미하며 가시성 그래프와 보로노이(Voronoi) 다이어그램이 이에 속한다. 셀 분할 기반 경로계획[3]은 주어진 형태공간을 다수의 셀들로 분할하고, 이때 각각의 셀들을 그래프의 정점으로 그리고 이웃한 셀들 간의 관계를 그래프의 간선으로 표현한 뒤, 주어진 출발 지점이 속해져 있는 셀부터 목표지점이 속해져 있는 셀까지의 최단 경로를 생성 시키는 방법이다. 포텐셜 필드 기반의 경로계

획[4]은 장애물들과의 충돌 없이 목적지점을 향해 나아갈 수 있게 하기 위해 형태공간상에 가상의 포텐셜 필드를 형성시키는 방법이다. Dynamic Window Approach(DWA)[5]는 로봇의 기구학적 구속조건을 포함시켜 로봇에 의하여 변화할 수 있는 속도영역에서, 장애물을 회피하면서 목표점으로 빠르게 주행할 수 있는 속도를 선택하는 실시간 장애물 회피 알고리즘이다. 그러나 실시간 장애물 회피 알고리즘만을 이용한 로봇주행은 전체 공간에 대한 정보 결여로 주행 중 국부최소(local minimum)에 빠져 목표점에 도달하지 못할 수 있는 단점을 가지고 있다. 이 문제를 해결하기 위해 Brock[6]이 제시한 주행함수(navigation function)[7]를 이용하여 환경정보를 추가한 전역 DWA 방법이 나왔다.

지금까지 살펴 본 방법들은 본 논문에서 사용하는, 실시간으로 경로주행을 하는 비선형 최적화 방법[8]과 맞지 않는 방법이므로 새로운 장애물 회피 알고리즘이 필요했다. 본 논문에서 경로계획은 비선형 최적화 방법 [8-11]을 사용하여 별도의 경로 계획 과정을 생략하고 실시간으로 직접 이동로봇의 경로 주행이 가능하도록 하였으며 장애물 회피에 있어서 서브 골을 이용한 회피방법을 제안한다. 제안하는 장애물 회피 알고리즘은 가중치를 이용하여 주행 전에 이동로봇의 headings 방향을 목적지와 일치시켰으며 주행 경로에 장애물이 존재할시 장애물 근처에 서브 골을 생성하여 비교적 적은 거리를 이동하며 장애물을 회피 주행하도록 하는 방법이다.

본 논문에서는 자율주행으로 장애물을 회피하기 위해서 구현된 방법에 대해서 설명한다. 2장에서는 이동로봇의 기구학 모델과 비선형 최소자승법을 이용한 이동로봇의 주행, 서브 골을 이용한 장애물 회피 방법에 관해 설명을 하며 3장에서는 시뮬레이션을 통해서 제안한 알고리즘을 검증하였다.

† Corresponding Author : School of Electronics Engineering, Chungbuk Nat'l University, Korea.

E-mail: gwkim@cbnu.ac.kr

* Dept. of Control and Robotics Engineering, Chungbuk Nat'l University, Korea.

Received : March 20, 2014; Accepted : July 14, 2014

2. 비선형 최소자승법을 이용한 이동로봇의 주행

2.1 차분 구동형 이동로봇 기구학 모델

이동 로봇의 위치는 전역 기준 좌표계와 로봇 기준 좌표계의 관계식을 이용하여 정의할 수 있다. 이동로봇의 위치와 방위각을 표현하는 자세정보는 전역 기준 좌표계 상에서 다음과 같이 정의한다.

$$q_R = [x_R \ y_R \ \theta_R]^T \quad (1)$$

여기서 x_R, y_R, θ_R 은 각각 이동로봇의 위치와 방위각을 나타낸다.

휠 반경 r 과 휠 간 거리 W 를 갖는 차분 구동형 이동로봇의 기구학 모델은 자코비안 행렬(Jacobian matrix)을 이용하여 다음과 같이 구할 수 있다.

$$\dot{q}_R = J_{kin} \cdot \dot{\varphi} \quad (2)$$

$$\text{where } J_{kin} = \begin{pmatrix} r \cos \theta_R / 2 & r \cos \theta_R / 2 \\ r \sin \theta_R / 2 & r \sin \theta_R / 2 \\ r / W & -r / W \end{pmatrix}$$

여기서 $\dot{\varphi} = [\dot{\varphi}_r \ \dot{\varphi}_l]^T$ 는 좌우측 휠의 회전 속도, $\dot{\varphi}_r$ 과 $\dot{\varphi}_l$ 로 구성된 벡터를 나타내며 이러한 기구학 모델을 이용하면 전역 기준 좌표계에서의 로봇의 속도 예측이 가능하다.

2.2 이동로봇 주행을 위한 비용함수

비선형 최소자승법을 이용하여 이동로봇을 주행시키기 위해서는 최적화를 위한 비용함수의 정의가 필요하다. 여기서 최적화의 목적은 이동로봇의 현재 자세와 목적지의 자세의 오차를 최소화하는데 있으며 다음과 같이 오차 함수를 정의한다.

$$e(\varphi) = q_R(\varphi) - q_{Goal} \quad (3)$$

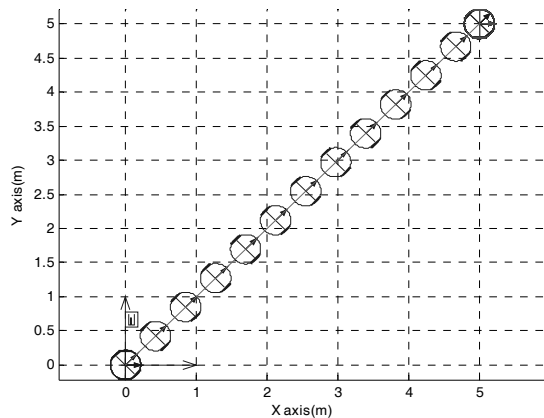


그림 1 장애물이 없을 때 가중치를 이용한 로봇 주행
Fig. 1 Navigation without obstacles using weight factors

여기서 이동로봇이 목적지를 향해 주행을 할 때 목적지 자세 정보와의 오차를 사용하면 적절한 주행을 할 수 없으므로 목적위치에 일정 오차 이내로 접근할 때까지는 로봇의 방위각과 목적지 방향의 오차를 이용하여 다음과 같이 정의한다.

$$\dot{\theta} = \begin{cases} \tan^{-1} \left(\frac{y_{Goal} - y_R}{x_{Goal} - x_R} \right) & \text{if } \text{pos.error} > \varepsilon \\ \theta_{Goal} & \text{else} \end{cases} \quad (4)$$

여기서 pos.error는 이동로봇의 현재위치와 목적지간의 유클리드 거리(Euclidean distance)이며 ε 은 충분히 작은 양의 값으로 정의한다.

식 (4)를 이용하여 수정된 목적지 자세정보 \dot{q}'_{Goal} 과 위치 오차와 방위 오차를 효율적으로 최소화하기 위한 가중치 행렬을 이용하여 오차 함수를 다음과 같이 다시 정의한다.

$$e(\varphi) = A \cdot (q_R(\varphi) - \dot{q}'_{Goal}) \quad (5)$$

$$\text{where } A = \begin{bmatrix} 1-r & 0 & 0 \\ 0 & 1-r & 0 \\ 0 & 0 & r \end{bmatrix}$$

로봇이 목적지를 인식하고 주행을 시작할 때 먼저 로봇의 현재 방향과 골의 방향의 차이를 계산해서 설정해 놓은 범위 안의 값이 들어올 때 까지 가중치 r 의 값을 1을 준다. 그 후 로봇의 방향과 골의 방향의 값이 설정해 놓은 범위 안으로 들어오면 가중치 r 의 값을 0을 주어서 로봇이 위치만 계산해서 주행할 수 있도록 한다. 이는 로봇이 장애물을 만났을 때 조금 더 효율적인 주행을 하도록 하는데 일조를 한다. 그림 1은 장애물이 없을 때 위의 가중치를 가지고 로봇이 목적지까지 가는 방법을 실험한 결과이다.

이동로봇을 목적지로 이동시키기 위해 식 (5)에서 정의한 오차 함수를 최소화하기 위한 비용 함수(cost function)를 다음과 같이 정의한다.

$$E(\varphi) = \frac{1}{2} e(\varphi)^T e(\varphi) \quad (6)$$

비용함수는 비선형 함수로 오차 함수의 제곱 형태로 정의되며 이를 최소화하기 위해 비선형 최소자승법을 이용한 최적화 방법을 적용한다.

2.3 Gauss-Newton 방법을 이용한 주행 제어

이동 로봇을 원하는 위치와 자세로 주행시키기 위한 비선형 최소자승 최적화 문제를 아래와 같이 정의한다.

$$\min_{\varphi \in R^2} E(\varphi) = \min_{\varphi \in R^2} \left\{ \frac{1}{2} e(\varphi)^T e(\varphi) \right\} \quad (7)$$

식 (7)에서 φ^* 주위가 선형적이고 미분가능하다면 $\partial E(\varphi^*) / \partial \varphi = 0$ 를 만족하는 휠 위치, φ^* 에서 비용 함수가 최소

값을 가지며, 비용 함수는 테일러 급수 확장(Taylor series expansion)을 통해 다음과 같은 근사화된 선형 모델을 얻을 수 있다.

$$\tilde{E}(\varphi) = E(\varphi^*) + \nabla E(\varphi^*) \cdot (\varphi - \varphi^*) \quad (8)$$

비용 함수가 φ 에 대해 2차 미분 가능하다면 근사 모델 (approximation model)의 변화율(gradient)은 아래와 같다.

$$\frac{\partial \tilde{E}(\varphi)}{\partial \varphi} = \nabla E(\varphi^*) + \nabla^2 E(\varphi^*) \cdot (\varphi - \varphi^*) \quad (9)$$

이 때 φ_k 에서 식 (9)의 근사 모델의 변화율이 0이 된다면 비용 함수를 최소화 하는 해가 될 것이다.

따라서 식 (2)에서 정의한 기구학 모델의 자코비안 행렬, $J_{kin}(\varphi) = \partial q_R / \partial \varphi$ 을 이용하여 비용 함수의 1차, 2차 미분을 구하면 다음과 같다.

$$\nabla E(\varphi) = J_{kin}(\varphi)^T e(\varphi) \quad (10)$$

$$\nabla^2 E(\varphi) = J_{kin}(\varphi)^T J_{kin}(\varphi) + R(\varphi) \quad (11)$$

$H(\varphi) = \nabla(J_{kin}(\varphi))^T e(\varphi)$ 는 헤시안(Hessian) 항목을 포함하는 레지듀얼(residual) 항목으로 무시할 수 있으며 식 (10), (11)에서 구한 1차, 2차 미분을 식 (9)에 적용하여 비용 함수를 최소화 하는 횡 위치 해를 반복적 형태로 구하면 다음과 같다.

$$\varphi_{k+1} = \varphi_k - (J_{kin}(\varphi_k)^T J_{kin}(\varphi_k))^{-1} J_{kin}(\varphi_k)^T e(\varphi_k) \quad (12)$$

위 식을 이용하여 직접 좌우측 휠의 속도 제어를 통해 목적지까지 이동로봇을 주행시키며 단순하고 효과적인 주행이 가능하다. 속도 제어를 위해 일정한 제어 주기 Δt 를 이용하여 휠 속도를 얻을 수 있다.

다만 식 (12)를 통하여 얻은 휠의 속도가 이동로봇의 물리적인 속도 제한 범위를 넘는 해를 제공할 경우 제한 범위를 넘은 속도를 정규화 하여 비례적으로 속도 제어를 한다. 이 방법은 회전 속도 제어의 방해요소로 작용하지 않기 때문에 효율적인 주행을 가능케 한다[8].

2.4 서브 골을 이용한 이동로봇 장애물 회피 주행

지금까지 제안한 방법으로 장애물이 없을 때 이동로봇의 주행은 문제없이 수행 할 수 있다. 이제부터는 장애물이 있을 때의 회피 주행에 대해서 제안하고자 한다.

이동로봇의 장애물 회피 주행에 다양한 방법이 있겠지만 본 논문에서는 장애물을 만났을 때 서브 골을 이용하여 회피 주행을 한 후 목적지로 주행하는 방법을 제안한다.

먼저 서브 골을 구하기 전에 안전거리에 대해서 정의하고자 한다. 이동로봇이 주행을 하다가 경로 상에 장애물이 존재할 때 충돌을 회피하기 위해서는 장애물을 인식 할 수 있는 약간의 시간과 충분한 거리가 필요하다. 약간의 시간은

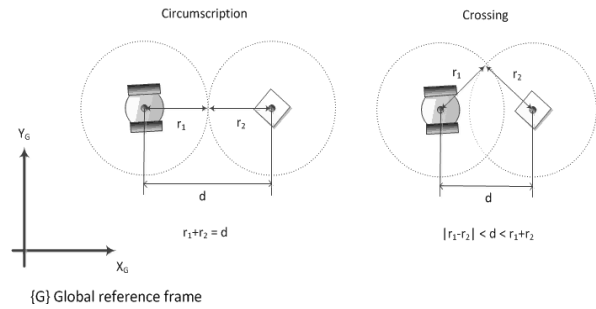


그림 2 안전거리에 대한 정의
Fig. 2 Definition of the distance of the safety

충분히 먼 거리에서부터 장애물을 인식하도록 하면 얻을 수 있으므로 거리에 대한 정의만 하면 된다. 이동로봇과 장애물의 충돌을 방지하기 위한 최대한의 안전거리 d_{max} 를 1로 정의 하였을 때 이동로봇 좌표와 장애물의 좌표를 각각의 중심으로 하고 최대안전거리를 반지름으로 하는 원이 있다고 가정하면 두 원이 만나는 지점부터 접지기 전까지를 안전거리라고 정의한다.

이동로봇과 장애물의 두 원이 만난다는 의미는 서로 가까이 있다는 의미이므로 이동로봇의 주행방향을 바꿔야 되는 데 여기서 서브 골이 필요하다. 그러나 서브 골의 좌표를 구하기 전에 이동로봇의 주행방향을 살펴봐야한다. 이동로봇과 장애물의 두 원이 만난다고 해서 무조건 이동로봇의 경로를 방해하는 위치에 장애물이 있지는 않으므로 이동로봇의 주행방향에 장애물이 있을 때만 서브 골을 생성하기로 한다. 또한 이동로봇의 좌표가 장애물의 원 밖에 있을 수도 있고 안에 있을 수도 있으므로 두 가지 경우로 나누어서 서브 골을 생성하기로 한다.

먼저 이동로봇의 좌표가 장애물의 원 밖에 있을 때 이동로봇 주행방향과 장애물의 좌표를 계산하여 서브 골을 생성할지 아니면 지나갈지를 판단한다. 다음 식을 이용하여 이동로봇의 현재 좌표 (x_r, y_r) 와 장애물의 좌표 (x_o, y_o) 를 지나가는 직선의 방정식을 구하고 그 직선의 기울기 값을 θ_1 이라 정의한다. 또한 원 밖의 한 점에서 그은 접선의 방정식과 원의 반지름은 직교한다는 성질을 이용하여 θ_2 값을 정의한다.

$$y - y_r = \left(\frac{y_o - y_r}{x_o - x_r} \right) * (x - x_r) \quad (13)$$

$$\theta_1 = \tan^{-1} \left(\frac{y_o - y_r}{x_o - x_r} \right) \quad (14)$$

$$\theta_2 = \sin^{-1} \left(\frac{d_{max}}{d} \right) \quad (15)$$

로봇 주행방향 θ_R 이 두 접선사이($\theta_1 - \theta_2 < \theta_R < \theta_1 + \theta_2$)에 있다고 판단되면 서브 골의 좌표를 계산한다. 이동로봇의 현재 위치에서 장애물의 원과 접하는 접선의 방정식을 구하여 원과 접선이 만나는 점의 좌표를 구한 후 접선 위에 있는 점들 중 구한 좌표 이후의 점 하나를 선택해 서브 골로

정의한다. (실험결과 원과 접선이 만나는 점을 서브 골로 잡는 것보다 그 점에서 1.2배한 위치가 더 좋은 결과를 보였다.) 단, 두 개의 접선(Tangent line A, B)으로 인해 서브 골이 두 개(s_1, s_2)가 생성되므로 다음 식을 이용하여 각각의 서브 골의 위치와 목적지의 위치 사이의 거리를 계산해서 목적지와 더 가까운 점을 서브 골로 선택한다.

$$q_{Sub\ goal} = \begin{cases} s_1 & \text{if } \sqrt{(q_{goal} - s_1)^2} < \sqrt{(q_{goal} - s_2)^2} \\ s_2 & \text{else} \end{cases} \quad (16)$$

다음으로 이동로봇의 좌표가 장애물의 원 안에 있을 경우이다. 앞의 경우와 마찬가지로 이동로봇 주행방향과 장애물의 좌표를 계산하여 서브 골을 생성할지 아니면 지나갈지를 판단한다. 앞에서는 두 접선사이를 이동로봇의 주행범위로 정의하였지만 이번 경우에는 이동로봇의 좌표가 장애물의 원 안에 있으므로 접선을 구할 수 없다. 따라서 접선 대신 현재 이동로봇의 좌표와 각각의 두 원의 교점을 동시에 지나가는 직선의 방정식을 이용한다. 식 (13)을 사용하여 θ_3 으로 정의하고 다음 식으로 θ_4 를 정의한다.

$$\theta_3 = \tan^{-1} \left(\frac{y_o - y_r}{x_o - x_r} \right) \quad (17)$$

$$\theta_4 = \cos^{-1} \left(\frac{0.5 * d}{d_{max}} \right) \quad (18)$$

로봇 주행방향 θ_R 이 두 직선사이($\theta_3 - \theta_4 < \theta_R < \theta_3 + \theta_4$)에 있다고 판단되면 서브 골의 좌표를 계산한다. 이번 경우의 서브 골의 좌표는 위에서 구한 직선의 방정식 위의 좌표를 선택한다. 현재 이동로봇의 좌표와 각각의 두 원의 교점을 동시에 지나가는 직선의 방정식을 구하기 위해서는 먼저 두 원의 교점의 좌표가 필요하다. 교점의 좌표는 두 원의 교점을 지나가는 직선의 방정식과 원의 방정식을 연립하여 얻을 수 있다. 아래 식은 두 원을 $x^2 + y^2 + ax + by + c = 0$, $x^2 + y^2 + a'x + b'y + c' = 0$ 이라고 가정했을 때 두 원의 교점을 지나가는 직선의 방정식이다.

$$x^2 + y^2 + ax + by + c - (x^2 + y^2 + a'x + b'y + c') = 0 \quad (19)$$

두 원의 교점의 좌표와 현재 이동로봇의 좌표를 이용하여 두 좌표를 지나가는 직선의 방정식을 구하고 서브 골을 선택한다. (본 논문에서는 두 원의 교점에서 1.2배 한 위치의 좌표를 서브 골로 설정했다.) 위의 경우와 마찬가지로 두 개의 직선(Straight C, D)에서 두 개의 서브 골(s_3, s_4)이 나오므로 식(20)를 이용하여 목적지의 좌표와 더 가까운 점을 서브 골로 선택한다.

$$q_{Sub\ goal} = \begin{cases} s_3 & \text{if } \sqrt{(q_{goal} - s_3)^2} < \sqrt{(q_{goal} - s_4)^2} \\ s_4 & \text{else} \end{cases} \quad (20)$$

서브 골이 정의되면 로봇은 서브 골에 도달할 때까지 전

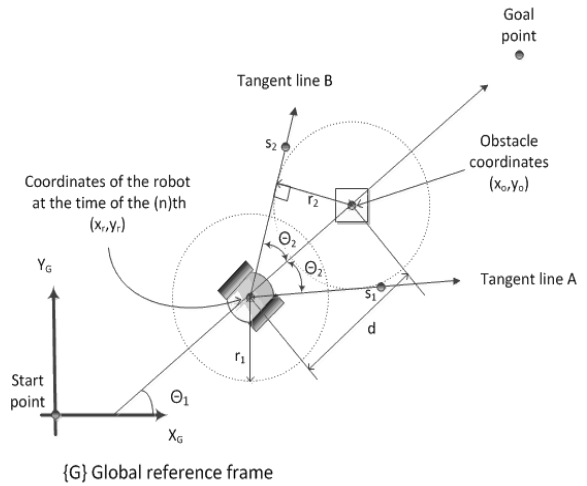


그림 3 로봇좌표가 장애물의 원 밖에 있을 때 서브골 정의

Fig. 3 Definition of the sub-goal when the robot center exists outside of the circle of the obstacle

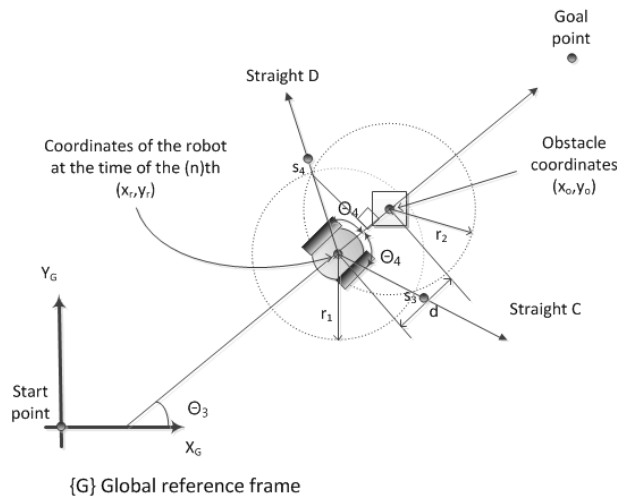


그림 4 로봇좌표가 장애물의 원 안에 있을 때 서브골 정의

Fig. 4 Definition of the sub-goal when the robot center exists inside of the circle of the obstacle

역 골에 도달할 때와 마찬가지로 주행한다. 서브 골에 도달했다는 것은 현재 인식된 장애물을 극복했다는 것을 의미하며, 따라서 또다른 장애물이 인식되지 않을 경우 전역 골로 목적지를 스위칭하게 된다. 따라서 로봇은 서브 골을 통해서 장애물을 회피한 후 다시 장애물이 없을 때처럼 주행해서 목적지에 도달한다.

서브 골 스위칭을 통한 장애물 회피 주행 알고리즘은 다음과 같다.

```

1  while  $q_R(\varphi) = q_{Global\ goal}$  do
2       $q_R(\varphi_k) = q_R(\varphi_{k-1}) + J_{kin}(\varphi_{k-1})(\varphi_k)$ 
3       $e(\varphi_k) = q_R(\varphi_k) - q_{Global\ goal}$ 
4       $d = \|q_R(\varphi_k) - q_O\|_2$ 
5      if  $0 < d < 2 * d_{max}$  then
6          if  $d_{max} < d$ 
7              if  $\theta_1 - \theta_2 < \text{robot's heading direction} < \theta_1 + \theta_2$ 
8                  "select the nearest sub-goal to goal..and go"
9              else  $d_{max} \geq d$ 
10                 if  $\theta_3 - \theta_4 < \text{robot's heading direction} < \theta_3 + \theta_4$ 
11                     "select the nearest sub-goal to goal..and go"
12                 else "go goal"
13              $\varphi_{k+1}$  calculation
14         end

```

3. 시뮬레이션 결과 및 성능 비교

본 장에서는 제안한 주행 알고리즘을 적용하여 시뮬레이션을 수행하였다.

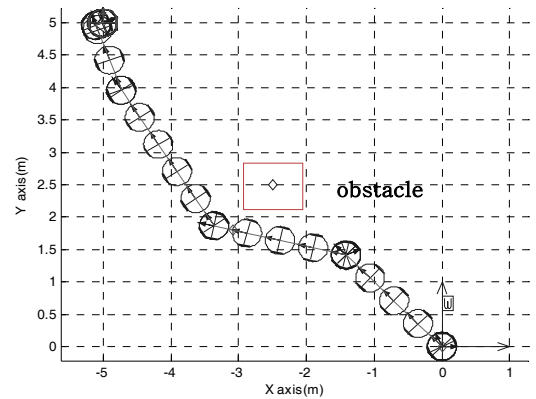
이동로봇의 초기 자세는 $[0 \ 0 \ 0^\circ]^T$ 이며 장애물과의 충돌을 막기 위하여 최대안전거리를 1m로 하였으며 제어 주기는 100ms로 가정하였다. 또한 휠의 속도 제어를 위한 이동로봇의 최대 속도를 5m/s로 설정하였으며 휠의 반경을 20cm, 휠 간 거리를 50cm로 설정하여 휠의 최대 속도가 25rad/s를 넘지 않도록 제한하였다. 이때의 로봇의 회전 속도와 방위각은 그림 5 (b), (c)와 그림 6 (b), (c)와 같다. 제안된 알고리즘의 효율성을 검증하기 위해 목적지의 위치 및 방위각을 달리하여 이동로봇의 장애물 회피 궤적, 이동로봇의 직선 및 회전 속도, 그리고 위치오차 및 방위각에 대한 결과를 분석하였다.

먼저 목적지의 자세정보가 $[-5 \ 5 \ 90^\circ]^T$, 장애물의 자세정보가 $[-2.5 \ 2.5 \ 0^\circ]^T$ 일 때의 시뮬레이션 결과는 그림 5와 같다. 이동로봇은 주행하기 전에 제자리에서 방위각을 조정 한 후 움직이기 시작했으며 파란색 점으로 표시된 서브 골을 경유하여 목적지로 접근함으로써 효율적인 주행을 보여주었다.

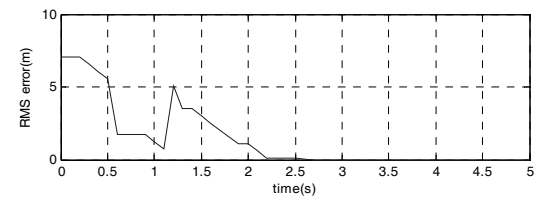
그림 6은 목적지의 자세정보가 $[5 \ 5 \ 90^\circ]^T$, 장애물의 자세정보가 $[2.5 \ 2.5 \ 0^\circ]^T$ 로 일 때의 시뮬레이션 결과로 목적지와 장애물 자세정보가 바뀌어도 생성된 서브 골을 경유하여 장애물을 회피하는 결과를 확인할 수 있다.

본 논문에서 제안하는 장애물 회피 알고리즘의 성능을 비교를 통해 검증하기 위해 기존 알고리즘 중 가장 보편적인 포텐셜 필드 방법과 동일한 조건에서 시뮬레이션을 수행하여 결과를 비교하였다.

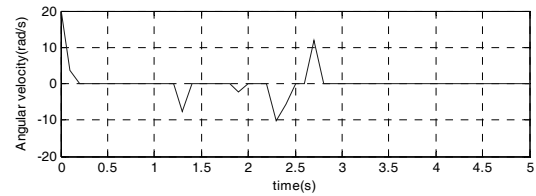
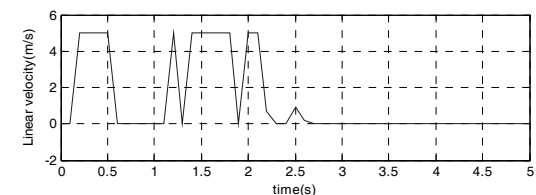
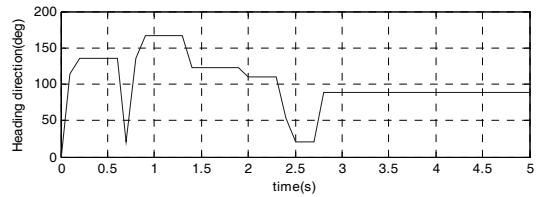
포텐셜 필드 방법을 이용한 시뮬레이션은 그림 6에서의 실험과 기구학적 요소, 동역학적 요소, 환경, 장애물의 위치 등 모든 변수에 대해 동일한 조건으로 설정하여 수행되었다.



(a)



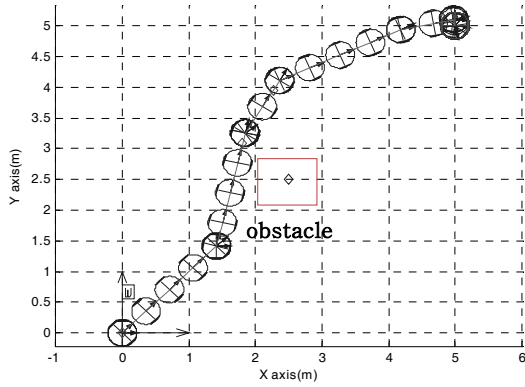
(b)



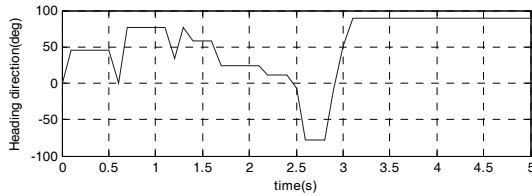
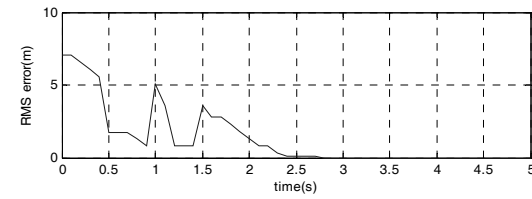
(c)

그림 5 시뮬레이션 결과 : 목적지의 자세정보 $[-5 \ 5 \ 90^\circ]^T$, 장애물의 자세는 $[-2.5 \ 2.5 \ 0^\circ]^T$ 일 때 (a) 이동로봇의 궤적, (b) 위치 오차와 이동로봇의 방위각, (c) 이동로봇의 직선 및 회전 속도

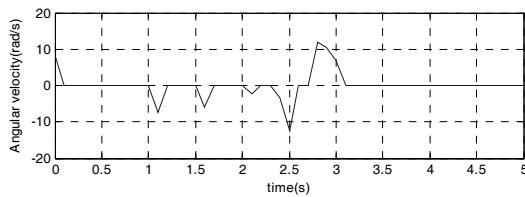
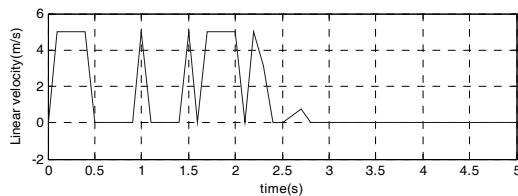
Fig. 5 Simulation results: (a) trajectory of the mobile robot, (b) pose error and the orientation, and (c) the linear and angular velocity of the robot



(a)



(b)

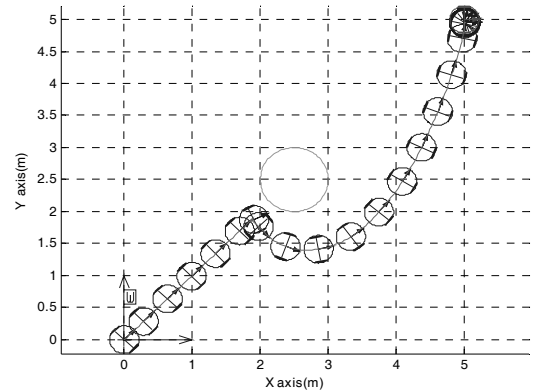


(c)

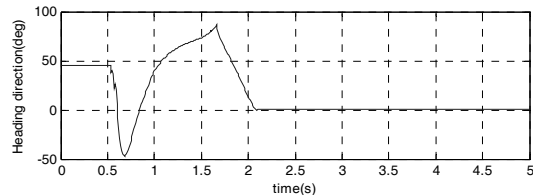
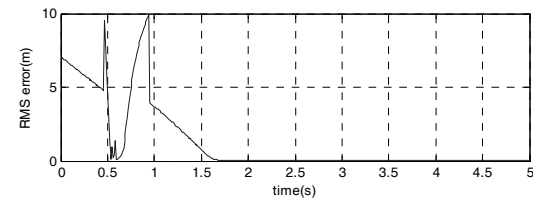
그림 6 시뮬레이션 결과 : 목적지의 자세정보 $[5 \ 5 \ 90]^T$, 장애물의 자세는 $[2.5 \ 2.5 \ 0]^T$ 일 때 (a) 이동로봇의 궤적,(b) 위치 오차와 이동로봇의 방위각 (c) 이동로봇의 직선 및 회전속도

Fig. 6 Simulation results: (a) trajectory of the mobile robot, (b) pose error and the orientation, and (c) the linear and angular velocity of the robot

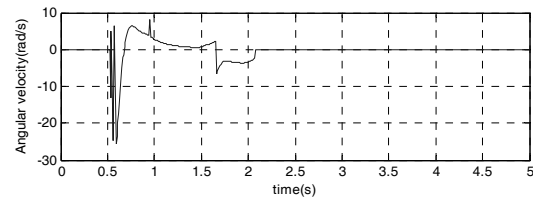
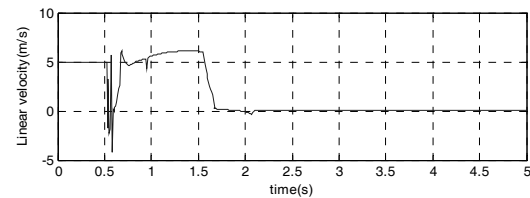
그림 7은 동일한 조건에서 수행된 포텐셜 필드 방법에 대한 시뮬레이션 결과를 보여준다. 그림 8은 제안된 방법과



(a)



(b)



(c)

그림 7 시뮬레이션 결과 : 목적지의 자세정보 $[5 \ 5 \ 90]^T$, 장애물의 자세는 $[2.5 \ 2.5 \ 0]^T$ 일 때 (a) 이동로봇의 궤적,(b) 위치 오차와 이동로봇의 방위각 (c) 이동로봇의 직선 및 회전속도

Fig. 7 Simulation results: (a) trajectory of the mobile robot, (b) pose error and the orientation, and (c) the linear and angular velocity of the robot

포텐셜 필드 방법을 이용한 시뮬레이션에 대한 비교 결과를 보여주고 있다. 제안된 방법과 포텐셜 필드 방법은 알고리

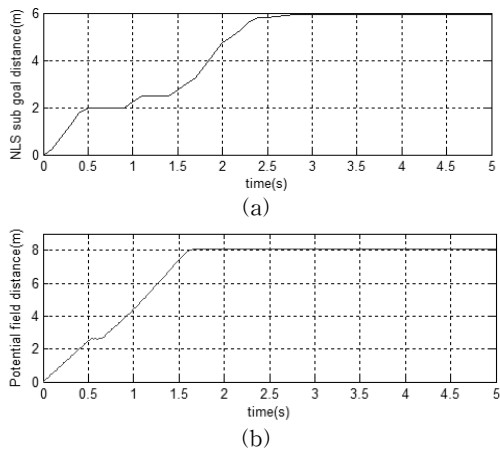


그림 8 로봇의 이동거리 비교: (a) 제안된 방법, (b) 포텐셜 필드 방법
 Fig. 8 Comparison of the path length: (a) the proposed method, (b) potential field method

증의 성능을 정량적으로 비교하기는 어려우며, 비교를 통해 얻은 개선된 성능이라는 것을 객관적으로 보이기는 어렵다. 하지만, 두 방식에 대한 특성을 분석하면 먼저 제안된 방식의 경우 전체적인 주행 시간의 경우 포텐셜 필드 방식에 비해 조금 더 지체된 것을 볼 수 있다. 하지만 주행거리의 경우 좀 더 짧다는 것을 알 수 있다. 제안된 방법의 경우 0.5초와 1.5초 사이에서 볼 수 있듯이 지체되는 구간이 발생한다. 이는 제안된 방법의 경우 로봇의 방위각에 대한 가중치 설정으로 회전주행 후 직진주행을 하도록 설정하였는데 회전주행을 하는 구간으로 분석되었으며, 이에 따라 전체적인 이동시간이 증가한 것으로 분석되었다. 이는 서브 골로의 변환과정에서 발생하는 것으로 포텐셜 필드 방법과 같이 전역환경에 대해 사전에 포텐셜 필드를 형성하는 것이 아니라 실시간으로 장애물 정보를 확인하여 처리를 하는 과정에서 발생하게 된다. 이는 센서기반의 장애물 회피 주행 알고리즘에서는 다수 발생하는 현상으로 당연한 결과라 할 수 있다.

4. 결 론

본 논문에서 제안한 방법을 이용하면 비교적 효율적으로 많은 거리를 돌아가지 않고도 장애물을 회피하면서 이동로봇을 목적지까지 주행할 수 있음을 시뮬레이션을 통해 선보였다. 제안한 방법은 실내/외 이동로봇의 장애물 회피 주행에 새로운 길을 제시했다고 생각된다.

향후 다수의 장애물이 존재하는 경우에 효율적으로 장애물을 회피하여 주행을 할 수 있는 방법에 대한 연구를 진행하고 실험을 통해 이를 검증할 예정이다.

감사의 글

이 논문은 2012년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음

References

[1] V. Lumelsky and T. Skewis, "Incorporating range

sensing in the robot navigation function," *IEEE Trans. on System, Man, and Cybernetics*, vol. 20, pp. 1058-1068, 1990.

[2] T. Lozano-Perez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560-570, 1979.

[3] R. A. Brooks and T. Lozano-Perez, "A subdivision algorithm in configuration space for find path with rotation" *Proc. of the 8th Joint Conf. on Artificial Intelligence*, pp. 799-806, 1983.

[4] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, pp. 412-442, 1990.

[5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, pp. 23-33, 1997.

[6] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *Proc of IEEE International Conf. on Robotics and Automation*, vol. 1, pp. 341-346, 1999.

[7] J. C. Latombe, *Robot Motion Planning*, 4th Edition, Kluwer academic publishers, 1996.

[8] G. W. Kim and Y. Y. Cha, "Navigation of a mobile robot using nonlinear least squares optimization," *Trans. of the Korean Institute of Electrical Engineers*, vol. 60, no. 7, pp. 1404-1409, 2011.

[9] G. W. Kim, "Visual servoing of a wheeled mobile robot with the obstacle avoidance based on the nonlinear optimization using the modified cost function," *Trans. of the Korean Institute of Electrical Engineers*, vol. 58, no. 12, pp. 2498-2504, 2009.

[10] G. W. Kim, K. T. Nam and S. M. Lee, "Visual servo navigation of a mobile robot using nonlinear least squares optimization for large residual," *Journal of Korea Robotics Society*, vol. 2, no. 4, pp. 326-332, 2007.

[11] G. W. Kim, and B. H. Lee, "Target tracking using the efficient estimation of the image Jacobian with large residual," *ROBOTICA*, vol. 24, pp. 325-327, 2006.

저 자 소 개



정 영 중 (鄭煥鍾)

2014년 충북대학교 제어로봇공학과 대학원 석사졸업. 관심분야는 mobile robot, navigation, 영상처리



김 곤 우 (金坤佑)

1974년 5월 10일생. 2006년 서울대학교 전기컴퓨터공학부 대학원(공학박사). 2006년~2008년 한국생산기술연구원 로봇기술본부 위촉연구원, 2008년~2011년 원광대학교 전자및제어공학부 조교수, 2012년~현재 충북대학교 전자공학부 조교수. 관

심분야는 Visual servoing, Robot Navigation, Sensor Fusion, Map Building and Localization.