

# 동적 환경에서의 전방위 이동 로봇을 위한 서클 리스트(Circle List) 기반의 장애물 회피

## Circle List-Based Obstacle Avoidance for Omni-directional Mobile Robots in Dynamic Environments

천 홍 석\*, 김 병 국  
(Hongseok Cheon<sup>1</sup> and Byung Kook Kim<sup>1</sup>)  
<sup>1</sup>KAIST

**Abstract:** An effective method of obstacle avoidance for omni-directional mobile robots is proposed to avoid moving obstacles in dynamic environments. Our method uses the concept of circle lists which represent the trajectories of robot and obstacles. This method predicts not only collision position but also collision time, and hence it enables the robot avoiding the most urgent collision first. In order to avoid obstacles, our method uses artificial repulsive force and contraction force. Simulation results show that the robot could avoid obstacles effectively.

**Keywords:** omni-directional mobile robot, obstacle avoidance, moving obstacles, dynamic environment, artificial force

### I. 서론

자율 이동 로봇이 계획하는 궤적이 갖추어야 될 주요 조건은 크게 다음 네 가지이다. 첫째, 장애물과 충돌 없는 궤적이라야 한다. 지도 상에 주어지지 않은 장애물을 피하기 위해서는 이동 중에 실시간으로 장애물을 인식하고 궤적을 변경해야 된다. 둘째, 목적지를 향하는 궤적이라야 한다. 셋째, 로봇의 성능을 고려하여, 로봇이 따라갈 수 있는 궤적이라야 한다. 넷째, 주어진 시간에 도착하도록 최선을 다해야 한다. 이 조건은 첫째 조건과 상충되는 부분이 있기 때문에 항상 지켜질 수는 없다. 장애물이 많은 경우에는 장애물을 피하기 위해 궤적을 변경하다 보면, 주어진 시간에 도착하지 못할 수 있다. 하지만 주어진 시간에 도착하기 위해 최선을 다하는 궤적 생성 방법이 좋다. 본 논문에서는 이러한 조건들을 만족하는 자율 이동 로봇의 새로운 궤적 생성 방법을 제안한다.

이동 중에 피하는 장애물의 정보는 센서를 통해 얻게 되는 데, 장애물이 존재하는 환경은 크게 정지된 장애물만 존재하는 정적 환경과 움직이는 장애물을 포함하는 동적 환경으로 구분된다.

정적 환경에서의 장애물 회피 방법을 위한 많은 연구가 진행되었다. 로봇이 목적지로 이동하는 도중에 정지된 장애물이 보이면 로봇은 이를 피해서 이동한다. 정적 환경에서의 장애물 회피 방법은 크게 속도기반의 접근법[1-3]과 APF (Artificial Potential Field) 기반의 접근법[4-6]으로 구분할 수 있다.

정적 환경에서의 기본적인 장애물 회피는 안정적으로 동작되는 방법들이 많이 연구되었기 때문에 더 이상 중요한 문

제가 아니다. 최근에는 구체적이고 복잡한 제한 조건에서의 장애물 회피 방법[17-18]과 움직이는 장애물이 존재하는 동적 환경에서의 장애물 회피가 많이 연구되고 있다. 노약자들은 이동 중에 능동적으로 이동 로봇을 신속히 피하기 힘들기 때문에, 로봇의 동적 환경에서의 장애물 회피는 인간과 같은 공간에서 이동하는 로봇의 경우에는 인간의 안전을 위해서 필수적이다. 동적 환경에서의 장애물 회피 방법은 크게 속도기반의 접근법의 확장[7-10,16]과 APF (Artificial Potential Field) 기반의 접근법의 확장[11-14]으로 구분할 수 있다.

속도기반의 접근법의 확장 방법은 로봇과 장애물의 속도를 고려하여 충돌이 일어나지 않도록 로봇의 속도를 주기적으로 결정하는 방법이다. 예를 들어, Owen의 방법은 Dynamic Window 방법을 확장한 것이다[8]. 이 방법은 움직이는 장애물과의 충돌을 피하기 위해 DOVS (Dynamic Objects Velocity Set)를 정의하였다. DOVS는 로봇이 해당 속도로 특정 기간 동안 일정하게 이동하면 장애물과 충돌이 발생할 수 있는 속도들을 표현한다. 이러한 속도기반의 접근법의 확장 방법은 예상 충돌 시간을 고려하지 않아서, 여러 충돌이 예상 될 때, 충돌에 따른 우선 순위를 둘 수 없다는 문제점이 있다. 따라서 여러 충돌이 예상될 때, 충돌을 회피하지 못할 수도 있다.

APF 기반의 접근법의 확장 방법은 장애물과의 충돌 예상 지점으로부터 로봇을 밀어내는 APF 방법을 이용한다. Kurihara의 방법은 로봇의 계획된 경로와 장애물의 예상 경로의 교점으로부터 로봇에 가상의 힘을 가한다[13]. 로봇이 장애물보다 먼저 지나갈 것으로 예상되면, 가상 인력을 가하고, 반대의 경우에는 척력을 가한다. 직선의 경로가 결정된 상황에서 로봇의 속도 변경만으로 장애물을 회피하는 방법이기에 정해진 경로 위에 있는 정지된 장애물이 있으면 피할 수 없다.

우리는 기존의 방법들의 문제점을 해결하기 위하여 [14]에서 제안된 서클 리스트 기반의 장애물 회피 방법 (CLOA:

\* 책임저자(Corresponding Author)

논문접수: 2010. 12. 15., 수정: 2011. 3. 26., 채택확정: 2011. 10. 5.

천홍석, 김병국: KAIST 전기 및 전자공학부

(hscheon@rtcl.kaist.ac.kr/bkkim@kaist.ac.kr)

※ 본 연구는 (고려대학교 로봇자율주행기술 전문인력양성센터를 통한) 지식경제부/한국산업기술진흥원 융복합형로봇전문인력양성사업의 지원으로 수행되었음.

Circle List-based Obstacle Avoidance) 을 개선 하였다. CLOA는 서클 리스트라는 개념을 이용하여 충돌 위치뿐 아니라 충돌 시간도 예상하여 로봇이 장애물과의 충돌을 피할 수 있도록 한다. 서클 리스트를 이루는 서클은 특정시간에 로봇과 장애물의 예상 위치를 표현하는 방법이다. 서클 리스트는 로봇의 계획된 궤적 또는 장애물의 예상 궤적을 표현하는 방법으로, 특정 시간 간격의 여러 서클의 집합이다. CLOA는 장애물의 예상 궤적과 로봇의 계획된 궤적을 이용하여 미래의 충돌 시점과 위치를 예측하고 이를 바탕으로 충돌이 없는 궤적으로 로봇 궤적을 수정하는 방법이다. CLOA에서는 척력과 수축력을 활용하여 경로를 변경하는 Elastic Band [6]의 척력과 수축력을 활용하였지만, 시간에 대한 내용을 포함하여 단순히 경로가 아닌 궤적을 변경하도록 하여, 움직이는 장애물이 있는 동적 환경에서 장애물의 움직임을 미리 예측하여 피할 수 있도록 하였다. CLOA는 기존의 속도기반의 접근법의 단점을 보완하여 로봇과 장애물 사이의 충돌 위치뿐 아니라 충돌 시간까지 고려하여 예상된 충돌 시각이 가까워지면 로봇이 해당 충돌을 피하도록 한다. CLOA는 충돌 예측에 따라 즉각적인 반응을 하는 것이 아니고, 계획된 궤적만 변경하여 각 충돌시점에 맞게 반응하도록 한다. 따라서, CLOA는 충돌 시점에 따라 우선순위를 두고 반응하기 때문에 여러 충돌이 예상될 때, 급한 충돌에 대해서는 빠른 회피 반응을 하고 먼 미래의 예상 충돌일수록 느긋한 회피 반응을 하여 불필요한 로봇의 움직임을 줄이고, 많은 장애물이 있을 때도 효율적으로 충돌을 회피할 수 있다. 그리고 기존의 APF 기반의 접근법과는 달리 미래의 충돌 예측에 따라 로봇이 직접적으로 반응하여 움직이지 않고, 로봇의 예상 궤적을 표현하는 로봇 서클 리스트의 형태만 변화되기 때문에, 상황에 따라 로봇이 우왕좌왕하지 않는다. 하지만 기존의 CLOA는 상황에 맞게 이동 시간을 유연하게 조절하지 않아, 실제 로봇이 CLOA에서 생성한 궤적을 따라가지 못하는 경우가 발생할 수 있었다. 그리고 센서로 측정된 장애물의 위치의 오차를 고려하지 않았다. 우리는 로봇 서클 리스트에 로봇이 이동할 수 있도록 로봇 서클을 추가 또는 삭제하는 기능을 추가하여, 주행 시간을 유연하게 할 수 있도록 하였고, 이 과정에서 로봇의 최대 속도를 고려하였다. 그리고 센서로 인식한 장애물의 위치 오차를 줄이기 위해 칼만 필터(Kalman Filter) [15]를 이용하였다. 본 논문은 II장에서 주요 개념인 서클과 서클 리스트에 대해 설명하고, III장에서 CLOA 방법을 구체적으로 설명한다. 그리고 IV장에서 시뮬레이션을 통해 다양한 장애물이 존재하는 환경에서의 CLOA를 통해 장애물을 피한 결과를 보인다. 마지막으로 V장에서 결론과 추후 과제에 대한 설명을 한다.

## II. 서클(Circle)과 서클 리스트(Circle List)

특정 시간의 로봇과 장애물의 몸체를 단순화하여 표현하기 위해 서클이라는 개념을 이용한다. 그리고 로봇과 장애물의 궤적을 표현하기 위해 서클 리스트라는 개념을 이용한다.

### 1. 서클

로봇 서클(RC)  $\tilde{R}_{k,n}$ 은 현재의 시각을  $kT$ 라 할 때, 미래의 특정 시각  $t=(k+n)T$ 에 계획된 위치  $\tilde{\mathbf{p}}_{k,n}=(\tilde{x}_{k,n}, \tilde{y}_{k,n})$ 에 있는 로봇의 반경  $r_R$ 보다 안전 거리  $d_s$ 만큼 큰 반경  $r$ 의 원이다.

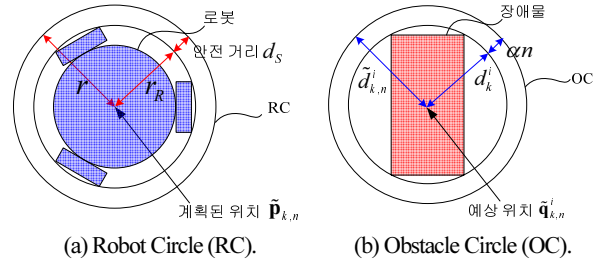


그림 1. 서클.

Fig. 1. Circle.

$$\tilde{R}_{k,n} = (\tilde{\mathbf{p}}_{k,n}, r) \quad (1)$$

$$r = r_R + d_s \quad (2)$$

장애물 서클(OC)  $\tilde{O}_{k,n}^i$ 은 미래의 특정 시각  $t=(k+n)T$ 에 장애물  $O^i$ 의 예상된 몸체를 포함하는 원, 즉 예상된 위치  $\tilde{\mathbf{q}}_{k,n}^i = (\tilde{x}_{k,n}^i, \tilde{y}_{k,n}^i)$ 와 예상 반지름  $\tilde{d}_{k,n}^i$ 으로 나타낸다.

$$\tilde{O}_{k,n}^i = (\tilde{\mathbf{q}}_{k,n}^i, \tilde{d}_{k,n}^i), \quad 1 \leq i \leq N_o \quad (3)$$

여기서  $N_o$ 는 인식된 장애물의 개수이다.  $\tilde{d}_{k,n}^i$ 는 OC 위치  $\tilde{\mathbf{q}}_{k,n}^i$ 의 불확실성에 비례하며, 현재로부터 먼 미래의 OC일수록 예상한 장애물의 위치에 대한 불확실성은 커진다. 즉, OC의 반지름은 센서부에서 측정된 장애물의 반경  $d_k^i$ 과 장애물의 위치에 대한 불확실성을 반영한 시간에 비례한 값  $\alpha n$ 의 합이다. 여기서  $\alpha$ 는 상수이다.

$$\tilde{d}_{k,n}^i = d_k^i + \alpha n \quad (4)$$

### 2. 서클 리스트

서클 리스트는 서클의 집합이다. 로봇 서클 리스트(RCL)  $L_k$ 는 현재 시각  $t=kT$ 부터 미래 시각  $(k+N_p)T$ 까지의 시간  $T$  간격의  $N_p+1$ 개 로봇 서클들로 구성되어 로봇의 계획된 궤적을 나타낸다.

$$L_k = \{\tilde{R}_{k,n} | n=0,1,2,\dots,N_p\} \quad (5)$$

인접한 RC  $\tilde{R}_{k,n}$ 와  $\tilde{R}_{k,n+1}$ 간의 거리  $l_{k,n}$ 는 해당 구간에서 시간  $T$  동안 로봇이 이동할 거리이기 때문에 해당 구간에서의 로봇의 평균 선속도에 비례한다.

장애물 서클 리스트(OCL)  $M_k^i$ 은 현재 시각  $t=kT$ 부터 미래 시각  $(k+N_p)T$ 까지의 시간 간격  $T$ 의  $N_p+1$ 개의 장애물 서클들로 구성되어 장애물  $O^i$ 의 예상된 궤적을 나타낸다.

$$M_k^i = \{\tilde{O}_{k,n}^i | n=0,1,2,\dots,N_p\} \quad (6)$$

CLOA 수행 주기이자 서클 리스트에서 서클 간의 시간 간격인  $T$ 와 서클 리스트 내의 서클 개수를 결정하는  $N_p$ 는 CLOA 성능을 결정하는 중요한 값이다.  $T$ 가 작으면 더 정확하게 장애물의 움직임을 감지할 수 있고, 더 세밀하게 로봇의 궤적을 계획할 수 있지만, 계산량이 늘어난다.  $N_p$ 가 크면

먼 미래의 충돌까지 예측하여 미리 대비할 수 있지만, 계산량이 늘어난다. 따라서 장애물이 많은 복잡한 환경일수록  $T$ 를 작게 하고  $N_p$ 를 크게 하면 좋지만, 로봇 시스템의 사양을 고려하여 설정해야 된다. 로봇이 최대 속도로 이동하는 구간이라도 인접한 RC간에 빈 공간이 없이 이어지도록  $T$ 를 설정하는 것이 적합하다. 즉, 로봇의 최대속도가  $v_{max}$  일 때,  $T = \frac{2r}{v_{max}}$  로 설정한다.  $N_p$ 는 시스템 사양에 따라 주기  $T$ 동안 서클리스트 관련 연산이 주기의 10%이내에 이루어지도록 설정한다.

### III.3. 서클 리스트를 이용한 장애물 회피

CLOA를 이용한 장애물 회피 문제: 매 주기  $T$  마다 로봇의 위치  $\mathbf{p}_k$ 와 각 장애물  $O^i$ 의 위치  $\mathbf{q}_k^i$ 와 반경  $d_k^i$ 를 인식하는 센서부를 포함하는 최대속도가  $v_{max}$ 인 전방위 이동 로봇을 대상으로, 목적지  $\mathbf{p}_g$ 와 로봇이 목적지에 도착하는 시간  $N_g T$ 가 주어져 있다. 동적 환경에서 로봇이 장애물과의 충돌 없이 목적지에 도착할 수 있도록, 서클 리스트를 이용하여 주행 중에 로봇의 궤적을 생성 및 변경하라.

CLOA는 주기  $T$ 마다 크게 다음 다섯 단계를 통해 수행하여 로봇의 궤적을 생성한다.

단계 1: 이전 주기의 RCL과 목적지와 이동 시간을 이용하여 RCL을 생성한다.

단계 2: 센서부에서 주어진 장애물의 위치와 크기를 이용하여 OCL을 생성한다.

단계 3: 로봇과 장애물의 충돌을 예상한다.

단계 4: 예상된 충돌을 회피할 수 있고, 부드러운 궤적으로 이동할 수 있도록 RCL을 변경한다.

단계 5: 로봇의 속도 제한을 고려하고 주어진 시간에 목적지에 로봇이 도착할 수 있도록 하기 위해 RCL에서 RC를 삽입/제거 한다.

#### 1. 로봇 서클 리스트 생성

RCL의 RC 개수  $N_p$ 는 최대  $N_{p,max}$ 개이고, 목적지 근처에서는 남은 시간에 맞게 설정된다.

$$N_p = \max(N_{p,max}, N_g - k) \quad (7)$$

로봇이 처음 출발할 때 ( $k=0$ )는, 목적지  $\mathbf{p}_g$ 와 목적지에 도착하는 시간  $t=N_g T$ 을 이용하여 현재 시각  $t=kT$ 에서  $t=(k+N_p)T$ 까지 시간 간격  $T$ 의  $N_p+1$ 개의 RC의 위치를 설정한다. 현재 시점의 RC  $\tilde{R}_{k,0}$ 의 위치  $\tilde{\mathbf{p}}_{k,0}$ 는 센서부에서 인식된 현재 시각으로 설정한다.

$$\tilde{\mathbf{p}}_{k,0} = \mathbf{p}_k \quad (8)$$

그 외의 RC들은  $\tilde{R}_{k,0}$ 와 목적지  $\mathbf{p}_g$  사이에 목적지에 도착하는 시간  $t=N_g T$ 을 이용하여 각 시각에 맞게 설정한다.

$$\tilde{\mathbf{p}}_{k,n} = \frac{n}{N_g}(\mathbf{p}_g - \tilde{\mathbf{p}}_{k,0}) \quad (9)$$

처음 출발할 때가 아니면, 현재 시각의 RC  $\tilde{R}_{k,0}$ 의 위치  $\tilde{\mathbf{p}}_{k,0}$ 은 센서부에서 인식된 현재 시각으로 설정하고, 그 외의 RC는 이전 주기의 수행 시각  $t=(k-1)T$ 의 RCL  $L_{k-1}$ 을 이용한다. RCL  $L_{k-1}$ 의 RC  $\tilde{R}_{k-1,n+1}$ 과  $L_k$ 의 RC  $\tilde{R}_{k,n}$ 는 동일한 시각의 로봇의 예상 위치이기 때문에 그대로 유지한다.

$$\tilde{\mathbf{p}}_{k,n} = \tilde{\mathbf{p}}_{k-1,n+1} \quad \text{for } n=1,2,\dots,N_p-1 \quad (10)$$

$L_{k-1}$ 에 없는 RCL의 마지막 RC  $\tilde{R}_{k,N_p}$ 의 위치  $\tilde{\mathbf{p}}_{k,N_p}$ 는 목적지  $\mathbf{p}_g$ 와 바로 이전 RC  $\tilde{R}_{k,N_p-1}$ 의 위치  $\tilde{\mathbf{p}}_{k,N_p-1}$ 와 남은 시간을 고려하여, 다음과 같이 결정한다.

$$\tilde{\mathbf{p}}_{k,N_p} = \tilde{\mathbf{p}}_{k,N_p-1} + \frac{\mathbf{p}_g - \tilde{\mathbf{p}}_{k,N_p-1}}{N_g - k - (N_p - 1)} \quad (11)$$

#### 2. 장애물 서클 리스트 생성

주기  $T$ 마다 CLOA는  $N_o$ 개의 장애물의 위치  $\mathbf{q}_k^i = (x_k^i, y_k^i)$  및 반지름  $\tilde{d}_{k,n}^i$ 을 센서부로부터 받아서 장애물의 궤적을 예상하여 각 장애물의 OCL을 생성한다 ( $i=1,2,\dots,N_o$ ). 센서 데이터를 이용하여 구한 장애물의 위치는 오차가 있다. 예상 궤적의 오차를 줄이기 위해서 장애물의 위치와 속도를 칼만 필터(Kalman Filter) [15]를 이용하여 계산한다. 사람, 로봇 등의 모든 장애물은 다른 운동을 하기 때문에 모든 장애물의 미래의 위치를 정확히 예측하기는 불가능하다. 하지만 장애물의 대부분이 직선운동을 하며, 등속구간에 비해 가속 및 감속 구간의 비율이 크지 않다. 따라서 장애물들의 대부분은 등속운동을 하고 있다고 할 수 있다. 즉, 등속 운동으로 장애물의 이동을 모델링하여 대부분의 장애물의 동작을 정확히 예측할 수 있다. 등속운동을 기반으로 설정한  $i$ 번째 장애물  $O^i$ 의 위치  $\mathbf{q}_k^i = (x_k^i, y_k^i)$ 와 속도  $\mathbf{v}_k^i = (\dot{x}_k^i, \dot{y}_k^i)$ 를 포함하는 상태  $\mathbf{x}_k^i = [\mathbf{q}_k^{iT}, \mathbf{v}_k^{iT}]^T$ 에 대한 상태방정식은 다음과 같다.

$$\mathbf{x}_{k+1}^i = \mathbf{F}_{k+1,k}^i \mathbf{x}_k^i + \mathbf{w}_k^i$$

$$\text{or } \begin{bmatrix} x_{k+1}^i \\ y_{k+1}^i \\ \dot{x}_{k+1}^i \\ \dot{y}_{k+1}^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^i \\ y_k^i \\ \dot{x}_k^i \\ \dot{y}_k^i \end{bmatrix} + \mathbf{w}_k^i. \quad (12)$$

그리고 센서부에서 장애물의 위치를 얻는 측정방정식은 다음과 같다.

$$\mathbf{q}_k^i = \mathbf{H}_k^i \mathbf{x}_k^i + \mathbf{v}_k^i$$

$$\text{or } \begin{bmatrix} x_k^i \\ y_k^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^i \\ y_k^i \\ \dot{x}_k^i \\ \dot{y}_k^i \end{bmatrix} + \mathbf{v}_k^i \quad (13)$$

$\mathbf{w}_k^i$ 와  $\mathbf{v}_k^i$ 는 각각 공분산행렬  $\mathbf{Q}_k^i$ 와  $\mathbf{R}_k^i$ 을 가지는 모델 잡음이다. 위 두 식을 기반으로 칼만필터를 통해 장애물의 보정된 위치  $\hat{\mathbf{q}}_k^i = (\hat{x}_k^i, \hat{y}_k^i)$ 와 속도  $\hat{\mathbf{v}}_k^i = (\hat{\dot{x}}_k^i, \hat{\dot{y}}_k^i)$ 를 포함하는

알고리즘	충돌 예측
입력: $L_k, M_k^i$	
출력: $c_{k,n}(j), b_{k,n}$	
for $n=1:N_p$	
$j=0$	
for $i=1:N_o$	
if $\ \tilde{\mathbf{p}}_{k,n} - \tilde{\mathbf{q}}_{k,n}^i\  < r + \tilde{d}_{k,n}^i$	
$j = j + 1$	
$c_{k,n}(j) = i$	
endif	
end	
$b_{k,n} = j$	
end	

그림 2. 충돌 예상 알고리즘.

Fig. 2. Collision prediction algorithm.

상태  $\hat{\mathbf{x}}_k^i$  를 구한다.  $i$ 번째 장애물의 미래 시각  $t = (k+n)T$  위치, 즉 OC  $\tilde{O}_{k,n}^i$  의 위치  $\tilde{\mathbf{q}}_{k,n}^i$  는 등속도 운동 공식을 통하여 다음과 같이 예측할 수 있다.

$$\tilde{\mathbf{q}}_{k,n}^i = (\tilde{x}_{k,n}^i, \tilde{y}_{k,n}^i) = \hat{\mathbf{q}}_k^i + \hat{\mathbf{v}}_k^i nT \quad (14)$$

### 3. 충돌 예상

RC  $\tilde{R}_{k,n}$  와 OC  $\tilde{O}_{k,n}^i$  가 겹치면 시각  $t = (k+n)T$  에 장애물  $O^i$  와 로봇이 충돌한다고 예상한다. RC  $\tilde{R}_{k,n}$  와 OC  $\tilde{O}_{k,n}^i$  가 겹치는 조건은 다음 부등식과 같다.

$$\|\tilde{\mathbf{p}}_{k,n} - \tilde{\mathbf{q}}_{k,n}^i\| < r + \tilde{d}_{k,n}^i, \quad (15)$$

이를 통해, 각 시각의 충돌 예상 개수  $b_{k,n}$ , 각 충돌을 일으키는 시각  $t = (k+n)T$  의  $j$  번째 충돌에 해당하는 장애물 번호  $c_{k,n}(j)$  를 알 수 있다 ( $j=1,2,\dots,b_{k,n}$ ). 시각  $b_{k,n}$  와  $c_{k,n}(j)$  를 구하는 상세한 알고리즘은 위 그림 2와 같다.

### 4. 로봇 서클 리스트 변경

로봇과 장애물의 충돌이 예상되면, RCL 내의 RC들의 위치를 변경하여, 충돌을 피해야 된다. 충돌을 피할 수 있도록 RC의 위치를 변경해야 되고, 실제 로봇이 따라갈 수 있도록 각 RC가 부드러운 경로로 되도록 일정한 간격을 유지해야 된다. 이를 위해 가상 힘을 이용하여 RC의 위치를 변경시킨다. 이 가상의 힘은 Quinlan 이 정적 환경에서 사용한 가상의 힘을 동적 환경에 적합하도록 수정 및 보완하였다[6]. 각 RC에 가해지는 가상 힘은 인접해 있는 RC 사이의 내부 수축력  $\mathbf{F}_{k,n}^{con}$  과 충돌 예상되는 위치로부터의 척력  $\mathbf{F}_{k,n}^{rep}$  으로 이루어진다.

$$\mathbf{F}_{k,n} = \mathbf{F}_{k,n}^{con} + \mathbf{F}_{k,n}^{rep} \quad (16)$$

그림 3(a)에서 보이는 내부 수축력  $\mathbf{F}_{k,n}^{con}$  은 고무밴드의 수축력을 기초로 한 것이다.

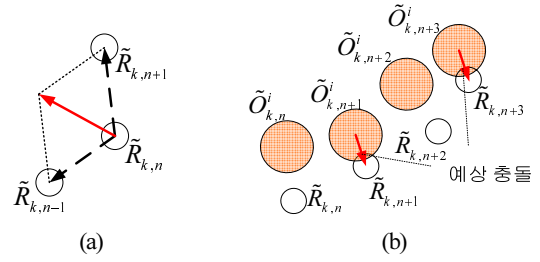


그림 3. 가상 힘.

Fig. 3. Artificial force.

$$\mathbf{F}_{k,n}^{con} = \zeta_1 (\tilde{\mathbf{p}}_{k,n-1} + \tilde{\mathbf{p}}_{k,n+1} - 2\tilde{\mathbf{p}}_{k,n}) \quad (17)$$

$\zeta_1$  는 상수이다. 내부 수축력은 RCL 내의 느슨함을 없애준다. 따라서 예상되었던 충돌이 장애물의 궤적이 바뀌어서 충돌이 안 일어나게 예상된다면, 충돌을 피하기 위해 변경되었던 로봇의 궤적 또한 원래의 궤적으로 돌아가게 된다. 그리고 내부 수축력은 RC들 사이의 간격을 유지하고 부드러운 궤적을 유지하게 한다.

그림 3(b)에서 보이는 외부 척력  $\mathbf{F}_{k,n}^{rep}$  은 RC  $\tilde{R}_{k,n}$  를 각 예상 충돌 위치  $\tilde{\mathbf{q}}_{k,n}^{c_{k,n}(j)}$  으로부터의 힘  $\mathbf{f}_{k,n}^{rep}(j)$  의 합력을 받게 된다.

$$\mathbf{F}_{k,n}^{rep} = \sum_{j=1}^{b_{k,n}} \mathbf{f}_{k,n}^{rep}(j) \quad (18)$$

척력의 세기는 RC에서 안전거리를 뺀 로봇을 포함하는 최소 반경의 원과 OC  $\tilde{O}_{k,n}^{c_{k,n}(j)}$  사이의 거리  $\psi_{k,n}(j)$  를 기준으로 결정된다.

$$\psi_{k,n}(j) = \left\| \tilde{\mathbf{q}}_{k,n}^{c_{k,n}(j)} - \tilde{\mathbf{p}}_{k,n} \right\| - \tilde{d}_{k,n}^{c_{k,n}(j)} - r_R \quad (19)$$

$\psi_{k,n}(j)$  이 클수록 약한 힘이 발생하며,  $\psi_{\min}$  이하에서는 최대 크기의 힘이 발생한다. 힘이 미치는 최대거리는 안전거리 범위, 즉  $\psi_{k,n}(j) = d_S$  까지 이다. 즉, 척력  $\mathbf{f}_{k,n}^{rep}(j)$  는 다음과 같다.

$$\mathbf{f}_{k,n}^{rep}(j) = \eta \left( \frac{1}{\psi'_{k,n}(j)} - \frac{1}{d_S} \right) \frac{1}{\psi'^2_{k,n}(j)} \nabla \psi_{k,n}(j) \quad (20)$$

이때,  $\eta$  는 상수이고, 힘의 세기를 결정하는 거리  $\psi'_{k,n}(j)$  와 힘의 방향  $\nabla \psi_{k,n}(j)$  는 다음과 같다.

$$\psi'_{k,n}(j) = \max(\psi_{k,n}(j), \psi_{\min}) \quad (21)$$

$$\nabla \psi_{k,n}(j) = \frac{\tilde{\mathbf{p}}_{k,n} - \tilde{\mathbf{q}}_{k,n}^{c_{k,n}(j)}}{\left\| \tilde{\mathbf{p}}_{k,n} - \tilde{\mathbf{q}}_{k,n}^{c_{k,n}(j)} \right\|} \quad (22)$$

가상 힘으로 인해 RC는 힘을 가하기 전의 위치  $\tilde{\mathbf{p}}_{k,n}$  에서 힘을 가한 후의 위치  $\tilde{\mathbf{p}}'_{k,n}$  로 이동한다.

$$\tilde{\mathbf{p}}'_{k,n} = \tilde{\mathbf{p}}_{k,n} + \mu \mathbf{F}_{k,n} \quad (23)$$

$$\mu = \frac{T^2}{2m_R} \quad (24)$$

$m_k$ 은 로봇의 질량이다.

5. 로봇 서클 삽입 및 제거

로봇이 정해진 시각에 목적지에 도착하기 힘든 경우에는, 이동시간을 늘려야만 한다. 이는 RCL에 RC를 추가함으로써 가능하다. 늘어났던 경로가 다시 줄어드는 경우, 추가한 RC가 필요가 없을 때도 있다. 이 경우 RC를 제거하여 정해진 시각에 로봇이 목적지에 도착하게 할 수 있다.

인접한 RC 사이의 거리를 고려하여 로봇이 이동하지 못하는 구간이라고 판단이 되면 새로운 RC가 추가된다. 로봇의  $T$  동안의 최대 평균 속도를 의미하는 RC간의 최대 거리  $D_{th}$  ( $=2r$ )에 대비 RC간의 거리  $l_{k,n}$ 이 1이 넘지 않도록 하여, 로봇의 속도를 제한한다.

$$\frac{l_{k,n}}{D_{th}} \leq 1 \tag{25}$$

식 (25)를 만족하지 않으면, RC를 삽입하게 되는데, RC가 삽입되는 위치의 공간적인 균형을 위해 인접한 두 RC의 위치를 이동한다. 먼저,  $l_{k,n+1}$ 와  $l_{k,n-1}$ 를 구한다. 두 거리 중 예 긴 것을 선택하고(그림 4(b)),  $l_{k,n}$ 와 앞서 선택된 거리를 각각 삼등분하고(그림 4(c)), 그림 4(d)와 같이 공간적 균형이

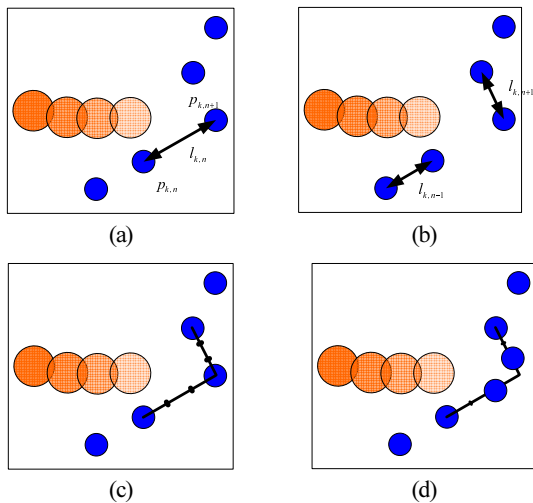


그림 4. 로봇 서클 추가.  
Fig. 4. Addition of a RC.

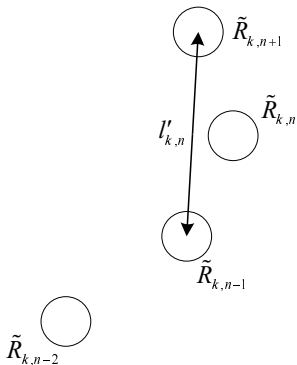


그림 5. 로봇 서클 삭제.  
Fig. 5. Deletion of a RC.

맞게 두 거리 사이의 RC를 이동시키고, RC를 삽입한다.

추가된 RC가 존재할 때, 하나의 RC를 삭제해도 로봇이 갈 수 있다고 판단되면 해당 RC는 삭제한다. 이는 주어진 시간에 로봇이 목적지에 도착하도록 유도한다. RC 삭제와 추가가 반복되는 것을 방지하기 위해, 추가와 삭제 사이에 여유  $\epsilon$ 를 둔다. 즉, 그림 5과 같이 RC  $\tilde{R}_{k,n}$ 가 삭제되었을 때의  $D_{th}$  대비 인접한 RC 간의 거리  $l'_{k,n}$ 가  $(1-\epsilon)$  이하이면, 즉 식 (26)을 만족하면, RC  $\tilde{R}_{k,n}$ 를 삭제한다.

$$\frac{l'_{k,n}}{D_{th}} < 1 - \epsilon \tag{26}$$

IV. 시뮬레이션

그림 6은 30개의 정지된 장애물과 15개의 움직이는 장애물이 있는 환경에서, 로봇이 (0, 0)에서 출발하여 (0, 16)으로 40초만에 이동하는 시뮬레이션 예제를 보여준다. RC의 반지름은 0.25이고, 모든 장애물의 반지름은 0.4이다. 그리고 RC간의 최대 거리는  $D_{th}$ 는 0.5, 추가와 삭제 사이에 여유  $\epsilon$ 은 0.3으로 설정하였다. 인식된 장애물 위치에는 x, y 각 축에 대해 평균이 0이고 분산이 0.05인 가우시안 분포의 랜덤 에러

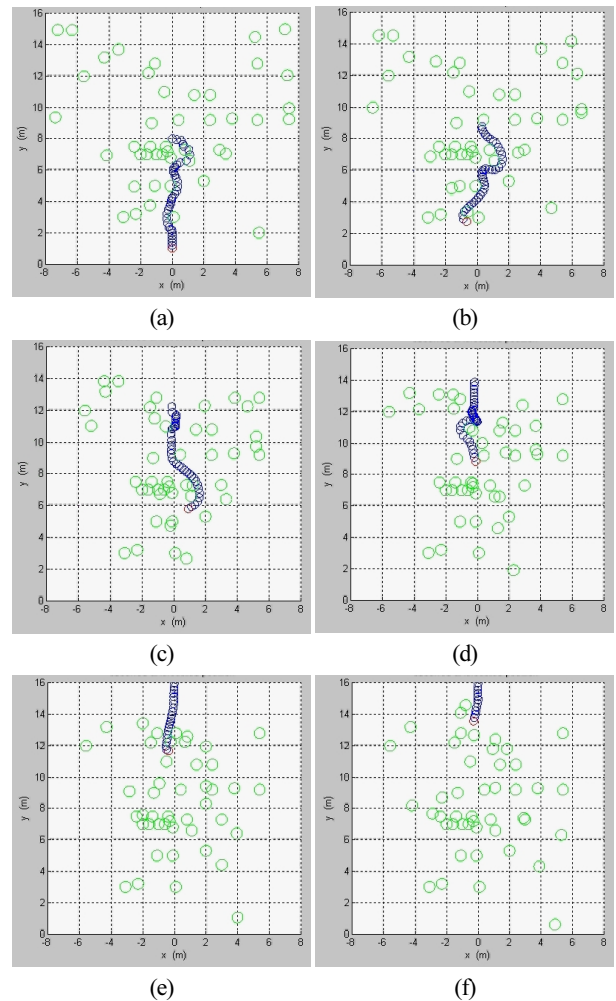


그림 6. CLOA 시뮬레이션.  
Fig. 6. Simulation of CLOA.

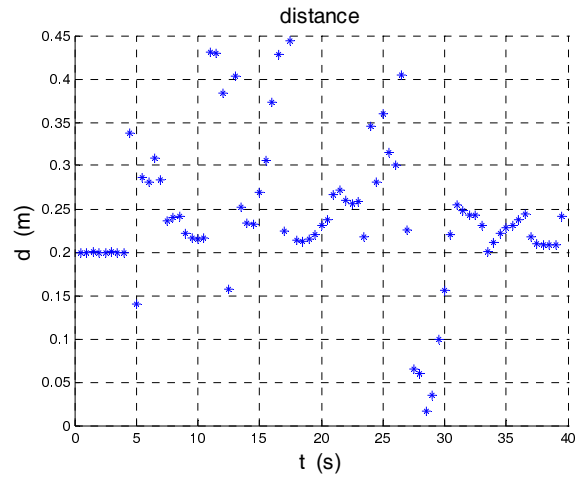
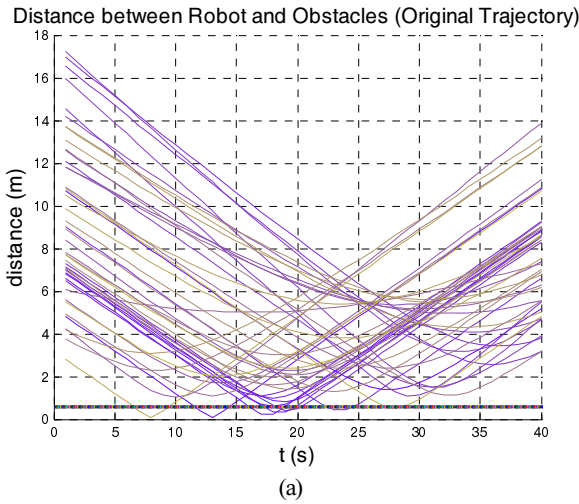


그림 8. 로봇의 움직임.

Fig. 8. Robot's movement.

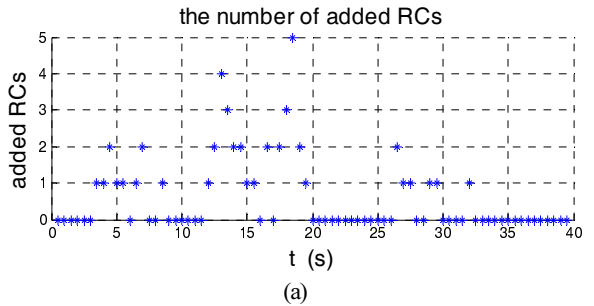
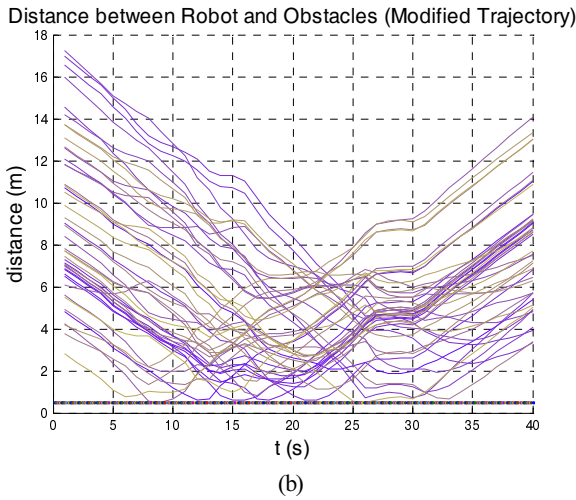


그림 9. 로봇 서클의 추가/삭제.

Fig. 9. Addition/deletion of RC.

그림 7. 장애물과 로봇 간의 거리.

Fig. 7. Distance between robot and obstacles.

값을 더하였다. 각각의 장면은 로봇과 장애물의 위치와 RCL을 그린 것이다. 로봇이 목적지로 이동하면서, 목적지로 충돌 없이 무사히 이동하는 과정을 시간 순으로 6컷을 나열한 것이다.

그림 7은 그림 6의 상황에서 로봇의 중심과 장애물의 중심 사이의 거리가 시간에 따라 어떻게 변화되는지를 보여준다. 충돌 유무를 구분하는 기준 거리를 그래프상에 선으로 표시했다. 로봇이 최초의 계획된 궤적으로 이동하는 경우(그림 7(a)), 로봇과 장애물들과의 거리가 기준선 아래로 내려오는 경우가 많지만, CLOA를 이용하여 궤적을 변경한 경우(그림 7(b))에는 장애물들과 로봇과의 거리가 기준선 아래로 내려오지 않는다. 즉, CLOA를 이용하여 궤적을 변경한 그래프에서는 충돌하지 않는 것을 확인할 수 있다.

그림 8은 그림 6의 상황에서 매 주기  $T$  마다 로봇의 이동 거리가 시간에 따라 어떻게 변화되는지를 보여준다. 이 값이 항상  $D_m$ 보다 작기 때문에, 설정한 조건을 잘 만족하여 전방위 이동 로봇이 잘 이동 하였음을 알 수 있다.

그림 9는 그림 6의 상황에서 늘어난 RC의 개수(그림 9(a))와 줄어든 RC의 개수(그림 9(b))를 보여준다. RC를 추가한 시점에 RCL 내에 RC를 삭제하여도 되는 부분에 삭제하기 때

그림 9. 로봇 서클의 추가/삭제.

Fig. 9. Addition/deletion of RC.

문에 동일한 시점에 추가와 삭제가 동시에 이루어지는 경우도 있다. 최종적으로 추가된 RC와 삭제된 RC의 개수가 동일하여, 계획한 시각에 도착하였음을 알 수 있다.

V. 결론

CLOA는 새로운 동적 환경의 장애물 회피 방법이다. 서클리스트라는 개념을 이용하여, 로봇의 계획된 궤적과 장애물의 예상 경로를 표현하여 충돌을 예상하고 회피한다. 가상의 척력을 이용하여 충돌이 없도록 로봇의 궤적을 변경하고, 가상의 수축력을 이용하여 궤적을 부드럽게 만든다. 충돌 예측 후에 충돌을 피하기 위해 즉각 적으로 반응하지 않고, 계획한 궤적만 변경하기 때문에, 예상 충돌의 위치와 시간에 맞게 반응하게 된다. 즉, 예상 충돌 위치뿐 아니라 예상 충돌 시각도 고려하여, 다양한 움직이는 장애물과의 충돌을 효과적으로 회피하고 불필요한 움직임을 최소화한다. 그리고 로

봇의 주행 성능을 고려하며 주어진 도착시각에 로봇이 목적지에 도착하도록 유도한다.

하나의 RC가 동일한 시각에 여러 OC들과 겹쳐지는 경우, 그 사이에 간혀 빠져 나오지 못하는 문제가 발생할 수 있다. 그리고 현재의 CLOA는 벽과 같은 긴 직선 형태의 장애물 회피에 적합하지 않다. 이에 대한 문제 해결이 필요하다. 그리고 전방위 이동 로봇뿐 아니라 차륜 구동 이동 로봇이나 자동차 형태의 이동 로봇에도 적용할 수 있도록 기구학을 고려한 CLOA로 확장할 필요가 있다. 또한 등속운동으로 단순화한 장애물 운동 모델을 가감속 구간을 포함하는 모델로 구체화하여 장애물 움직임에 대한 예측의 정확도를 높일 필요가 있다.

**참고문헌**

[1] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23-33, March 1997.

[2] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *Proc. of the IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, pp. 341-346, 1999.

[3] E. Demeester, M. Nuttin, D. Vanhooydonck, G. Vanacker, and H. Van Brussel, "Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section," *Proc. of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, pp. 2694-2699, 2005.

[4] J. Borenstein, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst. Man Cybern.*, vol. 19, pp. 1179-1187, 1989.

[5] R. B. Tilove, "Local obstacle avoidance for mobile robots based on the method of artificial potentials," *IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, vol. 2, pp. 566-571, 1990.

[6] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 802-807, 1993.

[7] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans*, vol. 28, no. 5, pp. 562-574, Sep. 1998.

[8] E. Owen and L. Montano., "A robocentric motion planner for dynamic environments using the velocity space," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 4368-4374, Oct. 2006.

[9] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760-772, July 1998.

[10] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving

obstacles," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1986-1991, 2007.

[11] Y. S. Nam, B. H. Lee, and N. Y. Ko, "An analytic approach to moving obstacle avoidance using an artificial potential field," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, USA, pp. 482-487, Aug. 1995.

[12] J. Ren, K. A. Mclsaac, and R.V. Patel, "A fast algorithm for moving obstacle avoidance for vision-based mobile robots," *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pp. 209-214, Aug. 2005.

[13] K. Kurihara, N. Nishiuchi, J. Hasegawa, and K. Masuda, "Mobile robots path planning method with the existence of moving obstacles," *ETFA 2005. 10th IEEE Conference on*, vol. 1, no. 8, pp. 195-202, 2005.

[14] H. S. Cheon and B. K. Kim, "Circle list based obstacles avoidance of mobile robots in dynamic environments," *KACC 2009*, Sep. 2009.

[15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Ser. D., J. Basic Engineering*, vol. 82, pp. 34-45, 1960.

[16] S.-H. Park and B.-H. Lee, "Reactive path planning in a dynamic environment," *Robotics, IEEE Transactions on*, vol. 25, pp. 902-911, 2009.

[17] G. I. Jee, J. H. Im, S. H. You, and D. H. Lee, "A path generation algorithm for obstacle avoidance in waypoint navigation of unmanned ground vehicle," *Journal of Control, Robotics and Systems (in Korean)*, vol. 17, no. 8, pp. 843-850, 2011.

[18] T. H. Kang and B. K. Kim, "Efficient online path planning algorithm for mobile robots in dynamic indoor environments," *Journal of Control, Robotics and Systems (in Korean)*, vol. 17, no. 7, pp. 651-658, 2011.



**천 흥 석**

2005년 고려대학교 전기전자전공공학부 졸업. 2007년 KAIST 대학원 전기전자전공 석사. 2007년~현재 동 대학원 전기전자전공 박사과정 재학 중. 관심분야는 장애물 회피 로봇



**김 병 국**

1975년 서울대학교 전자공학과 졸업. 1975년 KAIST 전기 및 전자 공학과 석사. 1981년 동 대학원 박사. 1981년~1986년 우진계기(주) 연구실장. 1982년~1984년 University of Michigan 방문연구. 1986년~현재 KAIST 전기 및 전자공학과 교수. 관심분야는 실시간 시스템, 로보틱스, 임베디드 제어.