

## 동적 장애물의 속도를 고려한 이동로봇의 궤적 재생성 기법

문 창 배\* · 정 우 진\*\*

\* 고려대학교 기계공학부

### Trajectory Regeneration Considering Velocity of Dynamic Obstacles Using the Nonlinear Velocity Obstacles

Chang-bae Moon\* and Woojin Chung\*\*†

\* Dept. of Mechanical Engineering, Korea Univ.

(Received December 9, 2013 ; Revised August 30, 2014 ; Accepted September 1, 2014)

**Key Words:** Mobile Robot(이동로봇), Navigation(주행), Obstacle Avoidance(장애물회피), Trajectory Regeneration(궤적 재생성)

**초록:** 서비스 로봇이 충돌안전성을 확보한 상태에서 고속 주행 임무를 수행하기 위해서는 동적 장애물의 속도를 고려한 궤적 계획이 필요하다. 정적 장애물만을 고려한 상태에서 궤적을 계획하는 경우 장애물과의 상대속도로 인해서 로봇이 장애물과 충돌할 수 있다. 본 연구에서는 동적 장애물의 속도를 고려한 궤적시간조정기법을 제안한다. 제안된 기법을 통해서 기존에 생성된 궤적의 시간을 조정해서 장애물 회피가 가능한지를 평가할 수 있다. 만일 회피가 불가능할 경우 생성된 경로가 아닌 다른 경로를 선택할 수 있다. 모의 시험 결과를 통해서 제안된 기법을 통해서 짧은 시간 내에 장애물 회피를 수행할 수 있음을 보였다.

**Abstract:** To achieve safe and high-speed navigation of a mobile service robot, velocity of dynamic obstacles should be considered while planning the trajectory of a mobile robot. Trajectory planning schemes without considering the velocity of the dynamic obstacles may collide due to the relative velocities or dynamic constraints. However, the general planning schemes that considers the dynamic obstacle velocities requires long computational times. This paper proposes a velocity control scheme by scaling the time step of trajectory to deal with dynamic obstacle avoidance problem using the RNLVO (Robot Nonlinear Velocity Obstacles). The RNLVO computes the collision conditions on the basis of the NLVO (Nonlinear Velocity Obstacles). The simulation results show that the proposed scheme can deal with collision state in a short period time. Furthermore, the RNLVO computes the collisions using the trajectory of the robot. As a result, accurate prediction of the moving obstacles trajectory does not required.

### 1. 서 론

최근 다양한 목적의 이동형 서비스 로봇이 개발되고 있다. 이동로봇이 서비스 임무를 효과적으로 수행하기 위해서는 고속으로 주행 가능해야 한다. 이동로봇이 고속으로 주행하기 위해서는 동역학적 특성을 고려한 장애물 충돌에 대한 안전성이 확보되어야 한다. 이동로봇의 주행제어 기법은 국소 장애물 회피 기법<sup>(1-3)</sup>과 전역 경로생성 기법<sup>(4,5)</sup>로 크게 분류할 수

있다. 국소 장애물 회피 기법은 센서정보를 이용해서 매번 속도출력을 계산하기 때문에 미지 환경에 적합하다. 전역 경로생성 기법은 소수의 동적 장애물이 존재하고 환경에 대해서 부분적으로 지도에 표현되지 않은 장애물이 존재할 때 적합하다. 전역 경로생성 기법은 고속 주행을 위해서 사전에 계획된 경로를 이용해서 충돌회피불가능상태(Inevitable collision state)<sup>(6)</sup>를 계산할 수 있기 때문에 유용한 제어 기법이다.

서비스 로봇이 충돌안전성을 확보한 상태에서 고속 주행 임무를 수행하기 위해서는 동적 장애물의 속도를 고려한 궤적 계획이 필요하다. 정적 장애물만을 고려한 상태에서 궤적을 계획하는 경우 장애물과의

† Corresponding Author, smartrobot@korea.ac.kr  
© 2014 The Korean Society of Mechanical Engineers

상대속도로 인해서 로봇이 장애물과 충돌할 수 있다. 이동 장애물이 로봇보다 고속으로 움직이는 경우에는 장애물을 회피하기 보다는 장애물이 먼저 지나가도록 감속해서 주행하는 것이 보다 효과적인 회피 방법이다. 장애물의 속도를 고려한 장애물과의 충돌 조건 계산 및 장애물회피 방법이 제시되었다.<sup>(7,8)</sup> VO (Velocity Obstacle)<sup>(7)</sup>은 장애물이 특정 속도로 이동할 경우 이동로봇의 속도공간에서 동적 장애물에 충돌하는 속도 집합을 나타낸다. VO 를 이용한 충돌회피를 수행하기 위해서는 무한대의 예측 시간을 사용할 수 없기 때문에 시지평(Time Horizon) 이내에서 계산하게 된다. 만일 주어진 시지평이 짧은 경우 동적 장애물에 충돌할 수 있다.<sup>(9)</sup> 이와는 반대로 시지평이 지나치게 길 경우 모든 속도 공간이 충돌하게 되는 것으로 평가되어서 로봇이 이동할 수 없게 된다.<sup>(10)</sup> NLVO(Nonlinear Velocity Obstacle)<sup>(8)</sup>은 동적 장애물의 이동 궤적이 주어진 경우 장애물에 충돌하는 VO 의 집합이다. NLVO 는 장애물의 이동궤적이 시간에 따라 변하고 불확실성을 가지는 경우에 적용하기에는 한계가 있다.

작업공간(Configuration Space)와 시간공간(Time Space)에서 동적 장애물을 고려한 경로계획기법<sup>(11)</sup> 고차원공간을 포함한 일반적인 경로생성 문제에 적합하다. 하지만 이동로봇이 주행하는 공간은 2 차원 공간이고 상대적으로 적은 수의 이동 장애물이 존재하는 공간이기 때문에 제안된 경로계획 기법<sup>(11)</sup> 적용하기에는 계산량이 많기 때문에 한계가 있다. 격자지도를 이용한 경로 생성 방법인 SIPP(safe interval path planning)<sup>(12)</sup> 기법이 제시되었다. 2 단계로 궤적을 변형하는 2-STD(two step trajectory deformer)<sup>(13)</sup> 기법이 제시되었다. 2-STD 기법은 우선 기존에 생성된 경로를 이동장애물 회피를 위해서 경로를 변형한다. 두 번째 단계로 변형된 궤적이 기구동역학적 제한 조건을 만족하도록 궤적을 재생성 한다.

추종경로를 유지하면서 속도나 토크를 조정하는 방법인 궤적시간조정(Trajectory Time Scaling) 방법은 매니플레이터의 경로추종오차를 최소화하고 토크한계를 초과하는 문제를 방법이 제안되었다.<sup>(14,15)</sup> 제안된 방법은 사전에 정의된 시간조정함수를 이용해서 궤적 추종시에 요구되는 토크가 매니플레이터 토크한계보다 큰 경우 궤적시간조정을 수행한다. 경로추종오차를 줄이기 위한 궤적시간조정기법이 제안되었다.<sup>(16-18)</sup>

본 논문에서 서비스로봇의 고속주행은 로봇의 최고 속도로 주행가능하며 동적 장애물을 감지한 경우 동적 장애물의 속도를 고려하여 회피 가능함을 의미한다. 이를 실내 환경에서 이동 장애물은 보행자이기 때문에

본 연구에서는 장애물회피를 위해서 서비스로봇이 가속하지 않는 것을 전제로 한다. 충돌이 예측되는 상황에서 장애물 회피를 위해 로봇이 감속하지 않을 경우 보행자는 로봇이 본인을 감지하지 못한 것으로 판단하여 보행자가 먼저 로봇을 회피하고자 하는 반응을 보인다. 보행자에 의한 충돌회피 시도는 장애물의 이동궤적을 예측하는데 어려움을 준다. 또한, 서비스로봇은 보행자의 움직임에 불편함을 주어서는 안 된다. 본 연구에서는 장애물회피를 위해서 로봇이 주행 경로를 변경하지 않고 속도를 감속하여 보행자가 로봇보다 먼저 로봇과 보행자가 교차하는 지점을 지나가도록 하는 궤적시간조정기법을 이용한 속도제어기법을 제안 하고자 한다.

NLVO 를 이용하여 장애물의 궤적이 주어진 경우 장애물과 충돌하는 로봇의 속도를 계산할 수 있다. 실제 환경에서는 동적 장애물의 정해진 궤적을 예측하기 어렵기 때문에 NLVO 를 이용해서 동적 장애물회피를 수행하는 것은 한계가 있다. 장애물의 궤적이 아닌 로봇의 궤적은 사전에 계산할 수 있다. 본 연구에서는 동적 장애물의 속도가 주어진 경우 로봇의 궤적을 NLVO 에서 장애물의 궤적으로 계산하는 RNLVO(Robot-NLVO)를 제시한다. 궤적의 시간 스텝이 일정할 때, 각 시간 스텝을 늘릴 경우 결과적으로 속도를 감속하는 효과가 나타나게 된다. RNLVO 를 이용해서 동적 장애물과의 충돌 조건을 계산하고 궤적시간조정을 통해서 동적 장애물을 회피하는 궤적을 재생성 하게 된다. 본 연구에서 제안된 방법은 계산량 측면에서 유용한 기법이며 기존에 사용되는 임의의 궤적생성기법에도 적용 가능하다는 장점이 있다. 서비스로봇의 고속주행을 위해서 빠르게 동적 장애물 회피 궤적을 생성할 수 있어 유용하다.

본 논문은 다음과 같이 구성되어 있다. RNLVO 를 이용한 충돌상태 조건계산 방법을 2장에서 제시한다. 3장은 시간조정함수와 장애물회피 궤적생성방법을 설명한다. 4장에서는 모의시험결과를 제시하고 5장에서 본 연구의 결론을 제시한다.

## 2. RNLVO 와 충돌 상태 조건 계산

### 2.1 Robot-NLVO

초기 시작시간  $t_0$  에서  $c(t_0)$ 이고 최종 시간  $t_f$  에서  $c(t_f)$ 인 이동로봇의 궤적  $c(t)$ 를 이용해서 RNLVO 를 계산한다. RNLVO 의 시지평  $t_h$  는  $t_f$  가 된다. RNLVO 는 장애물과 이동로봇이 충돌할 수 있는 속도공간에서의 집합이다. 만일 장애물의 속도가 RNLVO 에 포함되는 경우 장애물은 로봇과 충돌하게 된다. RNLVO 는 시작시간  $t_0$  에서  $t_f$ 까지

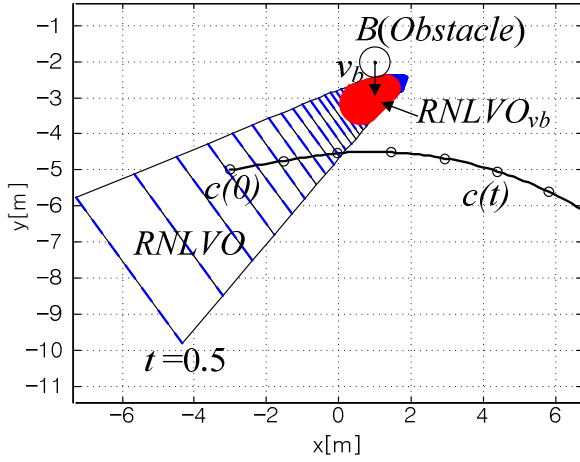


Fig. 1 Illustration of the RNLVO

다음 식을 이용해서 계산된다.

$$RNLVO_{t_0}^{t_f} = \bigcup_{t_f \geq t > t_0} \frac{c(t) \oplus \bar{A}}{t - t_0} \quad (1)$$

식 (1)에서  $\bar{A}$  는 로봇 A 에 포함되는 모든 영역을 나타낸다. 시간이  $t$  일 때 로봇의 궤적  $c(t)$ 에 로봇이 위치할 때 로봇이 점유하는 영역이 된다.  $\oplus$  는 민코프스키합(Minkowski sum)이다. 식 (1)의 상세한 계산방법은 Zvi Shiller 및 Frederic Large 의 연구에 제시되어 있다.<sup>(8)</sup>

Fig. 1 은 장애물 B 가 (1, -2)에 위치하고 로봇의 궤적이  $c(t)$ 일 경우 RNLVO 를 나타내고 있다. 만일 장애물의 속도가  $v_b$  일 경우 장애물과 로봇이 충돌하는 영역인  $RNLVO_{vb}$  는 다음 식을 이용해서 계산할 수 있다.

$$RNLVO_{vb} = \{RNLVO(t) | B + \gamma v \in RNLVO_{t_0}^{t_f}\} \quad (2)$$

식 (2)에서  $\gamma$  의 범위는  $[0, dist(B, c(t))]$ 이고  $dist(B, c(t))$ 는 B 와  $c(t)$ 의 거리이다.

### 2.2 충돌 상태 조건 계산

Fig. 2 는 장애물과 로봇이 충돌할 경우 시간이  $t_{v,min}$  일 때 최저 속도  $v_{min}$  에서 시간이  $t_{v,max}$  일 때 최대 속도  $v_{max}$  인 경우 RNLVO<sub>v</sub> 를 나타내고 있다. RNLVO<sub>v</sub> 는 로봇의 궤적을 장애물이 통과할 때 존재하게 된다. 만일 장애물의 속도  $v_b$  가  $v_{min}$  보다 작을 경우 로봇은 장애물과 교차하는 지점을 장애물보다 먼저 이동하게 된다. 장애물의 속도  $v_b$  가  $v_{min} \leq v_b \leq v_{max}$  범위인 경우 장애물은 로봇과 충돌하게 된다. 장애물과의 충돌을 회피하기 위한 궤적시간조정을 위한 계수  $\alpha$  는 장애물의 속도  $v_b$  를 이용해서 다음 식과 같이 정규화된 계수이다.

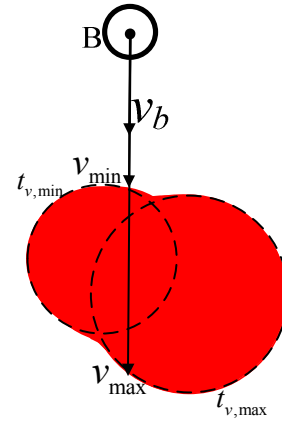


Fig. 2 Illustration of the RNLVO<sub>v</sub>

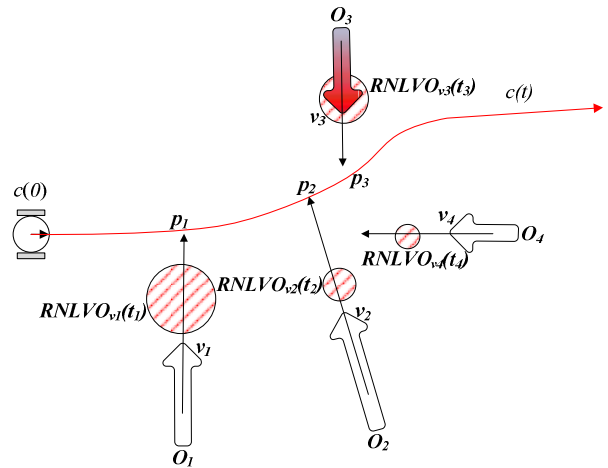


Fig. 3 Examples of the collision cases

$$\alpha = \begin{cases} \alpha_{min} = \|v_{min}\| / \|v_b\| \\ \alpha_{max} = \|v_{max}\| / \|v_b\| \end{cases} \quad (3)$$

장애물이 로봇보다 충돌하는 궤적을 사전에 통과하도록 하는 시간조정계수  $\alpha$  의 범위는  $\alpha \geq \alpha_{max}$  가 된다. 로봇이 장애물과 공통으로 지나는 영역에서 장애물이 해당 지점을 완전히 이탈하는 시간  $t_{v,max}$  일 때 로봇이 충돌을 회피하기 위해서 필요한 시간은  $\tau_p = \alpha_{max} \cdot t_{v,max}$  가 된다.

## 3. 장애물 충돌 회피 우선 순위 계산 및 궤적 재생성

### 3.1 장애물 회피 우선 순위 계산

Fig. 3 은 로봇과 장애물이 충돌하는 경우를 나타내고 있다. 장애물  $O_1$  과  $O_2$  는 로봇과 장애물이 공통으로 지나는 지점  $p_1$  과  $p_2$  를 장애물의 속도가 RNLVO 보다 작기 때문에 로봇이

지나친 이후에 해당 지점을 통과하게 된다. 장애물  $O_3$  는 장애물의 속도가  $RNLVO$  에 포함되기 때문에 로봇이 해당 궤적을 속도제어 없이 통과하게 되는 경우 장애물과 시간  $t_3$  일 때  $p_3$  에서 충돌하게 된다. 장애물  $O_4$  는 로봇의 초기 위치를 통과하게 되며 로봇이 초기 위치에서 정지해 있을 경우 장애물과 충돌하게 된다.

장애물회피를 위해서 충돌하는 장애물을 충돌회피 후보에 등록하게 된다. Fig. 3 의 경우 장애물  $O_3$  가 첫 번째 장애물로 등록된다. 장애물  $O_3$  를 회피하기 위한 시간 조정계수  $\alpha$  를 식 (1)을 이용해서 계산한다. 장애물  $O_3$  를 회피하기 위해서 속도를 줄일 경우 장애물  $O_1, O_2$  에 충돌할 수 있다. 따라서  $\alpha$  를 이용해서 시간을 조정할 경우  $O_3$  를 제외한 장애물에 충돌하는 경우 증가된 시간을 이용해서 충돌이 발생하지 않는 최소  $\alpha_{min}$  을 계산하여 충돌이 없는 궤적의 재생성이 가능하게 된다. 만일 충돌이 발생하지 않는  $\alpha_{min}$  이 존재하지 않을 경우 해당 경로를 통과할 경우 장애물에 충돌하게 된다.

3.2 궤적시간조정 방법

장애물이 로봇보다 먼저 교차지점을 통과하게 하기 위해서는 로봇이 시간  $\tau_p = \alpha_{min} \cdot t_{v,max}$  에 해당 지점을 통과해야 한다. 최종 통과 지점에서의 속도제한 조건을 이용해서 속도프로파일을 생성하는 기법<sup>(19)</sup> 제시된 바 있다. 본 연구에서는 시간제한조건 및 동역학적제한조건을 이용하여 궤적을 재생성하는 기법을 제안한다. 제안하는 기법은 로봇이 감속할 수 있는 최소의 궤적을 생성하여 결과적으로 장애물 회피를 위한 최적의 궤적을 생성하게 된다.

Fig. 4 는 장애물회피를 위해서 궤적을 재생성 할 때 감속하는 구간에서의 시간  $\tau_0$  에서  $\tau_p$  까지의 시간 조정 함수를 나타낸다. 시간이 지연된 궤적에서 충돌 지점을 지나는 시간은  $\tau_p$  로 증가하게 되며 원래의 궤적에서 통과하는 시간은  $t = \int_{\tau_0}^{\tau} f_1(\tau) d\tau$  이

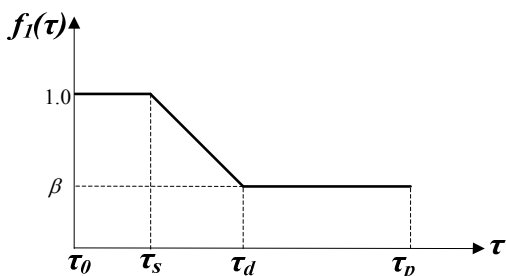


Fig. 4 Time scale function  $f_1(\tau)$  for generating deceleration trajectory

된다.  $\tau_s$  에서  $\tau_d$  구간은 동역학적 제한 조건을 고려한 감속 구간이다. 예를 들어, 원래 궤적의 제어 주기가 0.1s 인 경우,  $\beta = 0.7$  일 때 기존 속도를 30% 감속하는 결과가 된다.  $\beta$  와  $\tau_d$  는 다음 식을 만족해야 한다.

$$\int_{\tau_0}^{\tau_p} f_1(\tau) d\tau = t_{v,max} \tag{4}$$

$$\alpha_{min} \cdot t_{v,max} = \tau_p \tag{5}$$

식 (4)와 (5)를 이용해서  $\beta$  와 제한 조건을 고려한  $\tau_d$  의 허용 가능한 범위는 다음 식과 같다.

$$\beta = \frac{t_{v,max} - 0.5(\tau_s + \tau_d)}{\alpha t_{v,max} - 0.5(\tau_s + \tau_d)} \tag{6}$$

$$\tau_{d,min} \geq \tau_s + \max\left(\left|\frac{\beta v_d - v_s}{\dot{v}_{max}}\right|, \left|\frac{\beta \omega_d - \omega_s}{\dot{\omega}_{max}}\right|\right) \tag{7}$$

식 (6)에서 초기치  $\tau_d$  를 이용해서  $\beta$  를 계산하게 된다. 식 (7)에서  $v_d$  는 기존 궤적의 병진속도이고  $v_s$  는 감속을 시작하는 시간  $\tau_s$  에서의 병진속도이다.  $\omega_d$  는 기존 궤적의 각속도이고  $\omega_s$  는 감속을 시작하는 시간  $\tau_s$  에서의 병진속도이다.  $\dot{v}_{max}$  와  $\dot{\omega}_d$  는 각각 최대 병진가속도 및 각가속도를 나타낸다. 식 (6)에서 계산된  $\beta$  를 이용해서 식 (7)을 이용해서  $\tau_d$  를 계산할 수 있다. 식 (6)과 (7)을  $\tau_{d,min}$  보다 작을 때 까지 계산하여 최종적인  $\tau_d$  를 얻게 된다. 만일  $\tau_d$  가 0 보다 작을 경우 제한 조건을 만족하는 궤적을 생성할 수 없다.

Fig. 5 는 장애물과의 충돌 지점을 통과한 후 감속된 궤적을 원래의 궤적으로 복원하기 위한 시간 조정함수를 나타낸다.  $\tau_a \sim \tau_{af}$  구간에서 가속을 해서 시간  $\tau_{af}$  이후에 원래의 궤적과 동일하게 된다. 시간  $\tau_{af}$  와  $\tau_f$  는 다음 제약 조건을 가지게 된다.

$$\int_{\tau_p}^{\tau_f} f_2(\tau) d\tau = t_f - t_{v,max} \tag{8}$$

$$\tau_f = (\alpha - 1)t_{v,max} + \frac{1}{2}(1 - \beta)(\tau_{af} + \tau_a) \tag{9}$$

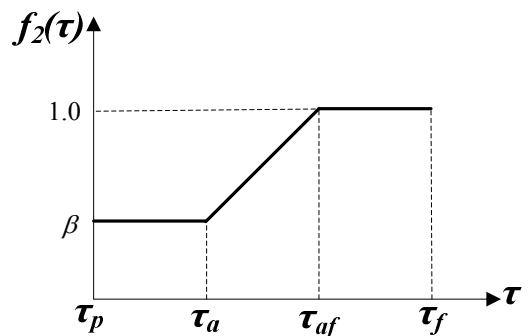


Fig. 5 Time scale function  $f_2(\tau)$  for generating acceleration trajectory

식 (11)과 (12)의 제한 조건을 만족하는  $\tau_{af}$  와  $\tau_f$  는 다음 식을 통해서 계산된다.

$$\tau_{af} = \frac{2(\alpha - 1)t_{v,max} + (1 - \beta)\tau_a - 2\tau_f}{(\beta - 1)} \quad (10)$$

$$\tau_{af} \geq \tau_a + \max\left(\left|\frac{v_{af} - \beta v_a}{\dot{v} \rightarrow_{max}}\right|, \left|\frac{\omega_{af} - \beta \omega_a}{\dot{\omega}_{max}}\right|\right) \quad (11)$$

식 (10)과 (11)을 통해서 식 (6)과 (7)과 같이 반복적으로 계산하여 식 (11)을 만족하는  $\tau_{af}$  를 계산할 수 있다.

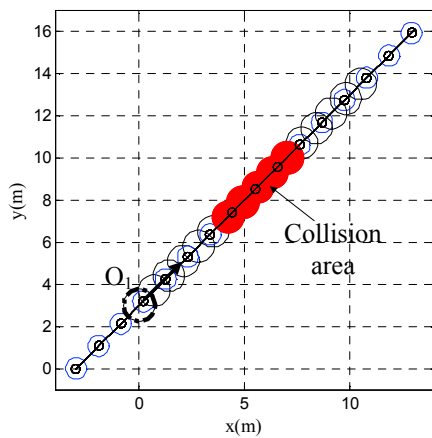
#### 4. 모의 시험 결과

모의 시험을 위해서 생성된 궤적은 최대 병진속도  $1.5m/s$ , 각속도  $30deg/s$  이고 최대 병진가속도  $0.6m/s^2$ , 각가속도  $60deg/s^2$  로 설정하였다. 기준 궤적의 시간 간격은  $0.1$  초로 생성하였다.

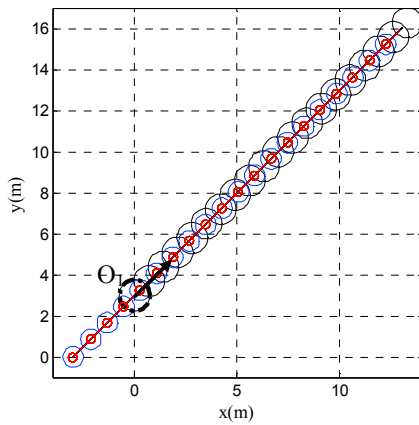
Fig. 6은 이동 장애물이 로봇과 동일한 경로를 로봇보다 앞선 지점에서 로봇보다 느린 속도로 이

동할 때의 장애물회피 결과를  $1s$  시간간격으로 나타내고 있다. 장애물과 충돌하는 경우 Fig. 5 와 같이 적색으로 나타나며 Fig. 6 에서와 같이 충돌한 영역은 없음을 알 수 있다. Fig. 6 과 같은 경우는 로봇이 좁은 복도를 이동할 경우 발생한다. 장애물은 초기 위치  $(0,3)$ 에서  $v_x=0.7m/s$ ,  $v_y=0.7m/s$  ( $v=0.99m/s$ )로 이동한다. Fig. 5 에 제시된 바와 같이 로봇의 이동속도가 장애물의 이동속도보다 빠르기 때문에 로봇과 장애물은 약  $(4,7)$ 에서 충돌하게 된다. Fig. 6 은 제안된 기법을 통해서 재 생성된 궤적을 나타내고 있다. 로봇은 장애물의 인접한 후방을 따라서 주행하게 되며 궤적의 종료시점까지 충돌은 발생하지 않는다. 제시된 결과는 로봇이 장애물로 인해서 지연되는 시간을 최소화한다.

Fig. 7 은 Fig. 5 와 같은 충돌이 일어날 때 제안된 기법에 의해서 생성된 궤적시간조정함수를 나타내고 있다. 로봇이 고속으로 이동하고 있기 때문에 충돌이 발생하는 시점 이전에 감속을  $0.51$  초



(a)



(b)

Fig. 6 Obstacle avoidance result when the obstacle moving along the trajectory of the robot

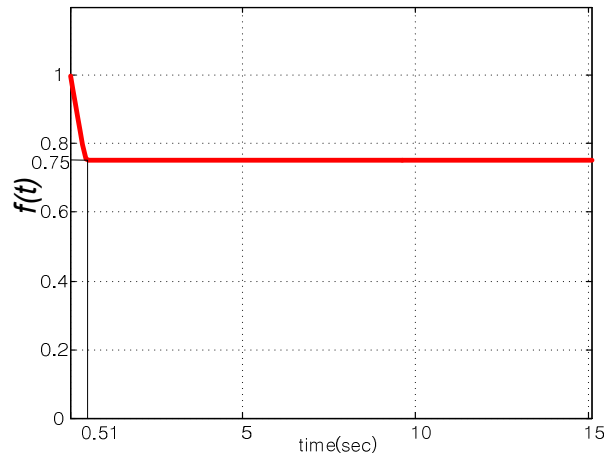


Fig. 7 Time scaling function of Fig. 6

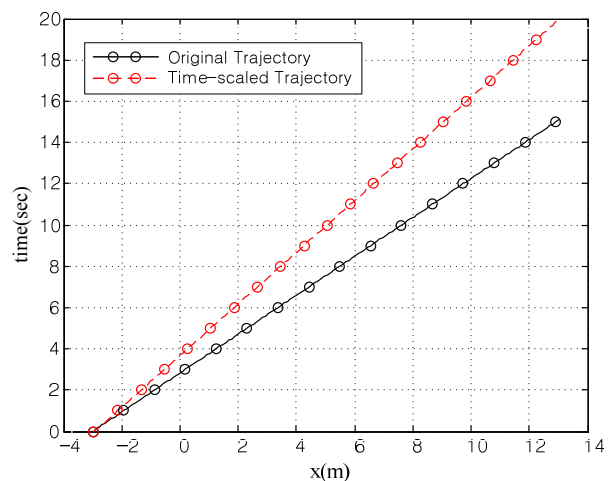


Fig. 8 Time scaled trajectory of Fig. 6

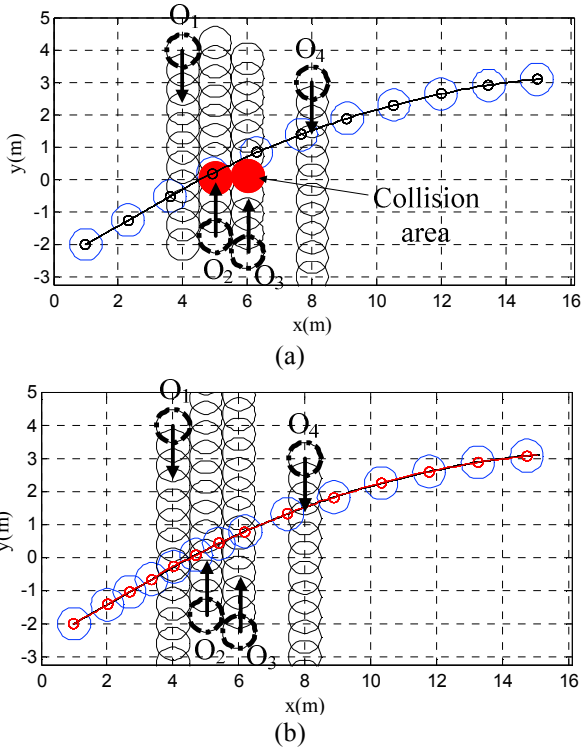


Fig. 9 Obstacle avoidance result for recursive collisions

까지 수행하고 이후에는 등속으로 이동하는 결과를 나타낸다. Fig. 8은 Fig. 7을 이용해서 재생성된 로봇의 궤적을 나타내고 있다. 장애물의 속도는 로봇에 의해서 영향을 받지 않고 로봇의 속도를 Fig. 8과 같이 감속하여 장애물의 후방을 추종하는 형태로 주행하게 된다. 장애물이 로봇의 최종궤적까지 로봇보다 저속으로 같은 경로를 이동하기 때문에 로봇의 궤적은 최종시간까지 초기궤적으로 복원하지 않는다. Fig. 6의 장애물회피를 위한 계산 시간은 측정 가능한 최소 시간 단위인 1ms 이내로 계산되었다.

Fig. 9는 다수의 장애물이 로봇의 궤적을 통과하는 경우의 장애물 회피결과를 제시하고 있다. 각각의 장애물은 Fig. 9에 표시된 방향으로 0.6m/s로 이동한다. Fig. 3에 제시된 바와 같이 복수의 장애물과의 충돌이 발생하는 경우 제안한 기법을 이용하여 충돌 순서를 계산하여 충돌 하는 순서대로 장애물 회피를 위한 궤적 재생성을 수행하게 된다. 초기궤적을 통해서 이동할 경우 장애물 O<sub>3</sub>에 의해서 1차 충돌이 발생하게 되며 충돌회피를 위해서 감속하는 궤적을 생성하게 된다. 이로 인해서 로봇이 천천히 주행하게 된다. 장애물 O<sub>3</sub> 회피를 위해서 속도를 줄일 경우 장애물 O<sub>2</sub>에 의한 2차 충돌이 발생한다. 최종적으로 감속하여 장애물 O<sub>1</sub>과의 장애물 회피를 위한 궤적이 생성된다. Fig. 9(b)는 제안된 기법에 의해서 생성된 궤적을 통해 장애물 회피 결과를 제시하고 있다.

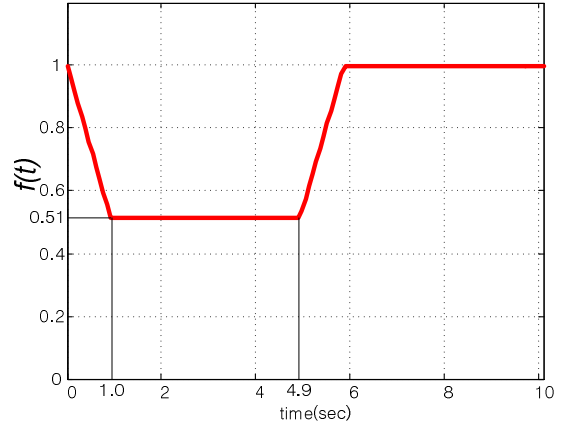


Fig. 10 Time scaling function of Fig. 9

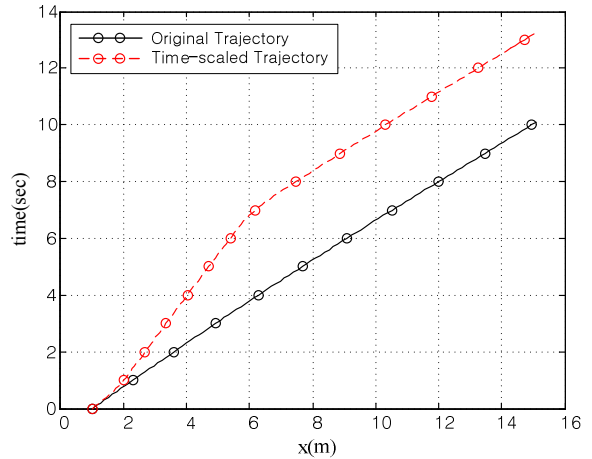


Fig. 11 Time scaled trajectory of Fig. 9

Fig. 10은 Fig. 9에 제시된 장애물회피를 수행하기 위한 시간조정함수를 나타내고 있다. 장애물회피를 수행한 이후에 발생하는 충돌이 없기 때문에 로봇의 원래궤적으로 복귀하게 된다. Fig. 10에 제시된 바와 같이 원래궤적으로 복귀를 위해서 4.9초 이후에 가속을 시작하여 6초 이후에 원래의 궤적으로 복귀하게 된다. Fig. 11은 생성된 궤적을 나타내고 있다. Fig. 9에 제시된 장애물회피를 위해서 2ms의 계산시간이 소요되었다.

15m 이상의 궤적을 통해서 제안된 방법의 장애물회피 방법을 시험하였다. 장애물 감지를 위한 센서군의 주기는 대부분 2ms를 훨씬 상회하는 주기를 가지기 때문에 센서의 측정 주기보다 매우 짧은 시간내에 계산할 수 있음을 알 수 있다. 모의 실험결과 계산 시간은 2ms 이내로 매우 짧은 시간내에 제안된 기법을 이용해서 장애물회피를 수행할 수 있음을 검증하였다. 제안된 기법은 장애물의 속도를 선형으로 고려하였으나, 매우 빠른 시간내에 경로를 재생성할 수 있기 때문에 장애물의 속도가 변하는 경우 실시간으로 대처 가능하다.

### 5. 결 론

본 연구에서는 동적 장애물의 속도를 고려한 궤적시간조정기법을 제안하였다. RNLVO 는 동적 장애물의 회피를 위한 충돌 조건을 계산할 수 있다. 제안된 기법을 통해서 기존에 생성된 궤적의 시간을 조정해서 장애물회피가 가능한지를 평가할 수 있다. 만일 회피가 불가능할 경우 생성된 경로가 아닌 다른 경로를 선택할 수 있다. 모의 시험 결과를 통해서 제안된 기법을 통해서 짧은 시간 내에 장애물 회피를 수행할 수 있음을 보였다. 제안된 기법은 기존의 궤적 생성기법에 적용이 가능하며, 장애물의 궤적을 정확하게 예측하지 않아도 동적 장애물 회피가 가능한 장점이 있다.

### 후 기

본 연구는 산업통상자원부의 산업융합핵심기술개발사업 (10048474, 고령화 세대에게 개인별 특화된 복지 서비스를 제공하기 위한 빅데이터 기반의 서비스 로봇 개발)의 지원으로 수행되었음. 본 연구는 농림축산식품부 농림축산식품연구센터지원사업에 의해 이루어진 것임.

### 참고문헌 (References)

(1) Minguez, J., Montano, L., Simeony, T. and Alamiy, R., 2001, "Global Nearness Diagram Navigation (GND)," In *Proceedings of IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 21~26.

(2) Fox, D., Burgard, W. and Thrun, S., 1997, "The Dynamic Window Approach to Collision Avoidance," *IEEE Robotics and Automation Magazine*, Vol. 4, No. 1, pp. 23~33.

(3) Borenstein, J. and Koren, Y., 1991, "The Vector Field Histogram - Fast Obstacle-Avoidance for Mobile Robots," *IEEE Journal of Robotics and Automation*, Vol. 7, No. 3, pp. 278~288.

(4) Konolige, K., 2000, "A Gradient Method for Realtime Robot Control," In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, Oct. 30 - Nov. 5.

(5) LaValle, S. M. and Kuffner, J. J., Jr, 2001, "Randomized Kinodynamic Planning," *International Journal of Robotics Research*, Vol. 20, No. 5, pp. 378~400.

(6) Fraichard, Th. and Asama, H., 2004, "Inevitable Collision States—A Step Towards Safer Robots?" *Advanced Robotics*, Vol. 18, No. 10, pp. 1001~1024.

(7) Fiorini, P. and Shillert, Z., 1998, "Motion Planning in Dynamic Environments Using Velocity Obstacles,"

*International Journal of Robotics Research*, Vol. 17, pp. 760~772.

(8) Shiller, Z., Large, F., Sekhavat, S. and Laugier, C., 2001, "Motion Planning in Dynamic Environments: Obstacles Moving Along Arbitrary Trajectories," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3716-3721, Seoul, Korea, May 21-26.

(9) Martinez-Gomez, L. and Fraichard, T., 2009, "Collision Avoidance in Dynamic Environments: an ICS-Based Solution and Its Comparative Evaluation," *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17.

(10) van den Berg, J., Lin, M. C., and Manocha, D., 2008, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 19-23.

(11) Zucker, M., Kuffner, J. and Branicky, M., 2007, "Multipartite RRTs for Rapid Replanning in Dynamic Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, April 10-14.

(12) Phillips, M. and Likhachev, M., 2011, "SIPP: Safe Interval Path Planning for Dynamic Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13.

(13) Delsart, V. and Fraichard, T., 2008, "Navigating Dynamic Environments Using Trajectory Deformation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, Sept. 22-26.

(14) Valasek, M., 1987, "Dynamic Time Parameterization of Manipulator Trajectories," *Kybernetika*, Vol. 23, No. 2, pp. 154~174.

(15) Dhal, O. and Nielsen, L., 1990, "Torque-Limited Path Following by On-Line Trajectory Time Scaling," *IEEE Trans. Robotics and Automation* Vol. 6, No. 6.

(16) Kiss, B. and Sz'adeczky-Kardoss, E., 2007, "On-Line Time-Scaling Control of a Kinematic Car with One Input," *Proceedings of the 15th Mediterranean Conference on Control & Automation*, Athens - Greece, July 27-29,.

(17) Gerelli, O. and Bianco, C. G. L., 2009, "Nonlinear Variable Structure Filter for the Online Trajectory Scaling," *IEEE Trans. Ind. Electron.*, Vol. 56, No. 10, pp.3921~3930.

(18) Sz'adeczky-Kardoss, E. and Kiss, B., 2009, "On-Line Trajectory Time-Scaling to Reduce Tracking Error," *Intelligent Engineering Systems and Computational Cybernetics*, Springer Netherlands, pp. 3~14.

(19) Au, T.-C. and Stone, P., 2010, "Motion Planning Algorithms for Autonomous Intersection Management," *AAAI 2010 Workshop on Bridging The Gap Between Task And Motion Planning (BTAMP)*, July 11-12, Atlanta, Georgia, USA.