



## 이륜 구동 로봇의 DWA를 활용한 경로 추종

이현석<sup>a</sup>, 유경현<sup>a</sup>, 이승원<sup>a</sup>, 김종민<sup>b</sup>, 이철수<sup>b,\*</sup>

### Path Pursuit using Coverage DWA of Two Wheel Drive Robot

Hyun-suk Lee<sup>a</sup>, Kyung-hyun Yoo<sup>a</sup>, Seung-woon Lee<sup>a</sup>, Jong-min Kim<sup>b</sup>, Cheol-soo Lee<sup>b,\*</sup><sup>a</sup> Department of Mechanical Engineering, Graduate School, Sogang University<sup>b</sup> Department of Mechanical Engineering, Sogang University

#### ARTICLE INFO

##### Article history:

Received	21	July	2023
Revised	9	August	2023
Accepted	14	August	2023

##### Keywords:

Differential drive wheeled robot  
 Path pursuit  
 Dynamic window approach  
 ROS(robot operating system)  
 RMSE(root mean square error)

#### ABSTRACT

This study presents a path pursuit algorithm designed for map coverage. While traditional path planning focuses on generating an optimal path between a robot's current location and a target point, coverage path planning aims to efficiently cover a designated area without any gaps. Therefore, a path pursuit algorithm is required to minimize path deviation. The coverage path was first calculated based on the map floor plan. Using a dynamic window, we established a new objective function that allowed the robot to minimize path deviation while pursuing the designated coverage path. This was achieved by considering the errors between the paths and the robot's current location, rather than just focusing on the robot's current position and the target point. Experimental results showed that our algorithm significantly reduced path deviation by 62% compared to pure pursuit algorithms.

## 1. 서론

모바일 로봇분야에서의 일반적인 경로 계획은 로봇의 현재 위치에서 목표점까지의 최적의 경로를 생성하는 것을 목표로 한다. 그러나 로봇 청소기 등에서는 커버리지 경로 계획은 주어진 이동 가능한 영역을 빈틈없이 효율적으로 이동할 수 있는 것을 목표로 한다. 따라서 커버리지 경로 계획에는 주어진 경로를 벗어나지 않고 추종할 수 있는 경로 추종 방법이 매우 중요한 역할을 한다.

경로 추종에 관한 연구는 경로와 로봇의 현재 위치에서 계산한 위치 오차와 방향오차를 줄이도록 제어하는 방법으로 pure pursuit<sup>[1-5]</sup>, follow the carrot<sup>[6-8]</sup>과 vector pursuit tracking<sup>[9,10]</sup> 경로 추종 방법이 있다. Pure pursuit 알고리즘은 전방 주시거리와 로봇의 중심 좌표를 이용하여 제어를 수행하는데, 이때 로봇의 방향에 접

하는 원의 곡률을 사용한다. 이 원의 곡률에 따라 제어가 이루어지므로, 경로 추종 시 부드럽고 자연스러운 움직임을 보이는 특징이 있다.

Follow the carrot 경로 추종 방법은 이동로봇의 중심좌표와 목표지향점을 지나는 직선을 찾아, 이동로봇의 방향이 직선의 방향과 일치하도록 이동로봇을 제어하는 방법이다. Vector pursuit tracking은 강체의 운동을 중심선을 따라 병진 운동하는 것처럼 표현하는 방법을 통하여 이동로봇의 선회 반경을 제어하는 방법이다.

최근에는 MPC(model predictive controller)<sup>[11,12]</sup> 이나 LQR(linear-quadratic regulator)<sup>[13-14]</sup>를 사용한 경로 추종에 관한 연구도 이루어지고 있다. MPC는 매 업데이트 시간 간격마다 수학적 모델을 사용하여 시스템의 미래 동작을 예측하고, 목적함수를 최소화하는 제어 값을 찾는다. LQR는 경로 추종상황 시 시스템의

\* Corresponding author. Tel.: +82-2-705-8646

E-mail address: cscam@sogang.ac.kr (Cheol-soo Lee).

측면 동역학을 고려하여 추종 오차를 최소화 시키는 최적의 제어 값을 찾는다.

이러한 경로 추종 방법은 주위 환경이나 장애물에 영향을 받지 않고 오직 주어진 경로와 로봇의 위치만을 고려하기 때문에 커버리지 경로가 장애물이나 벽과 근접하게 생성이 되었을 때, 충돌에 비교적 취약한 커버리지 경로 추종에는 적합하지 않다.

DWA(dynamic window approach)<sup>[15,16]</sup>는 속도 기반 충돌 방지 계획 알고리즘으로서 로봇의 위치에서부터 목적지까지 장애물을 고려한 최적의 속도를 계산한다. 이는 DWA로 하여금 지도 등 사전정보가 없는 장애물이 있는 환경에서 안전하게 목표지까지 도달할 수 있게 한다. 하지만 DWA는 목표점에 도달하는데 장애물과 로봇의 거리만을 고려하여, 주어진 경로가 장애물이나 벽과 근접하게 생성이 되었을 때, 실제 로봇은 장애물과 최대한 멀어지는 방향으로 움직이게 된다. 이는 마찬가지로 경로 이탈을 최소화 해야 하는 커버리지 경로 추종에 적합하지 않다.

본 연구에서는 로봇의 위치와 주어진 목표점만을 고려하는 것이 아닌, 주어진 경로와 로봇 위치 간의 오차를 고려하여, 경로 이탈을 최소화하며 동시에 장애물을 고려하는 경로 추종 방법을 제안한다.

환경 구성을 위해, ROS(robot operating system)<sup>[17]</sup>의 네비게이션 스택(navigation stack)과 지도 서비스(map service)를 사용하였다. 지도 서비스의 평면도를 사용하여 커버리지 경로를 생성하고, 경로 추종을 위해 새로운 로컬 플래너(local planner)로 커버리지 DWA를 구성했다.

본 연구에서 사용된 차동 이륜 구동형 로봇의 기구학적 모델링과 파라미터에 대해 기술하였다. 기존 DWA의 한계점에 대해 간략하게 기술한 후, 지도 서비스를 이용한 커버리지 경로 생성에 대해 설명한 후, 커버리지 DWA의 목적함수에 대해 기술하였다.

실제 환경 실험에서 횡방향 오차의 RMSE(root mean square error)와 전제 영역 대비 커버된 영역을 중점으로 커버리지 DWA를 대표적인 경로 추종 방법인 pure pursuit 알고리즘과 비교하여, 제안한 커버리지 DWA의 우수성을 검증하였다.

## 2. 차동 이륜 구동 로봇

본 연구에서는 DC 모터 2개와 앞뒤로 캐스터 2개가 결합된 차동 이륜 구동형 로봇 Fig. 1을 사용하였다.

로봇은 지름이 320 mm 높이가 610 mm인 원통형 로봇이다. 구동륜의 반지름 R 은 35 mm, 구동륜 간 사이 거리 2 L은 250 mm이다.

본 연구는 환경설정 면에서 ROS를 사용하였다. 이를 통해 Lidar센서와 AMCL를 활용한 위치 추정, 지도 서비스, 등을 이용할 수 있다.

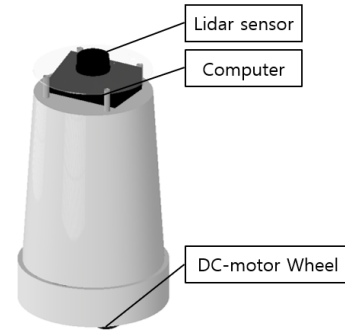


Fig. 1 Differential drive wheeled robot

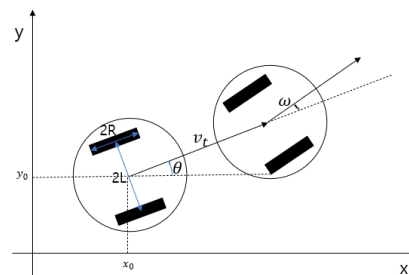


Fig. 2 Kinematics of differential drive robot

Intel Celeron N4020 CPU 코어, 4 GB RAM 환경에서 ROS를 구동하고 Lidar 센서는 ydlidar사의 TG50을 사용하였다.

차동 이륜 구동 로봇의 기구식<sup>[18]</sup>은 Fig. 2와 같이 표현할 수 있다. R은 구동륜의 반지름, 2 L은 구동륜 간 사이 거리를 나타낸다.

로봇 좌표계에서 차동 구동 로봇의 구동 륜 각속도를 각각 ( $\omega_l, \omega_r$ )라 나타낼 때, 로봇의 선속도  $v_i$  와 각속도  $\omega$ 는 식 (1), 식 (2)로 나타낸다.

$$v_i = \frac{v_r + v_l}{2} = R \left( \frac{\omega_r + \omega_l}{2} \right) \quad (1)$$

$$\omega = \frac{v_r - v_l}{2L} = R \left( \frac{\omega_r - \omega_l}{2L} \right) \quad (2)$$

전역 좌표계에서 차동 구동 로봇의 요우(yaw)각도가  $\theta$  이고, 전역 좌표계에서의 로봇의 위치와 속도를 각각  $(x, y)$ 와  $(\dot{x}, \dot{y})$  라 할 때, 이륜 구동형 로봇의 운동방정식은 식 (3)과 같이 나타낼 수 있다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega \end{bmatrix} \quad (3)$$

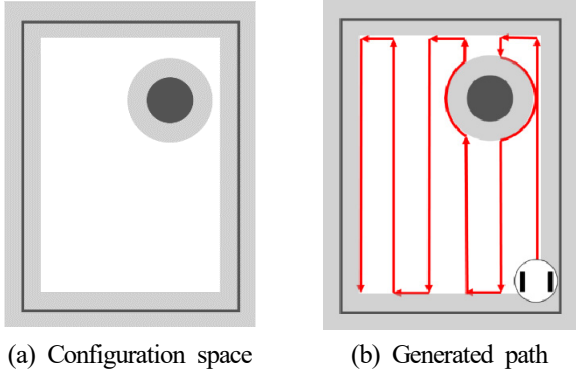


Fig. 3 Configuration space and generated path

전역 좌표계에서의 로봇의 위치와 차동 이륜 구동 로봇의 기구학적 파라미터는 선속도  $v_t$  와 각속도  $\omega$ 로 나타낼 수 있다.

### 3. 커버리지 경로 생성 및 커버리지 경로 추종 구현

본 연구에서는 Fig. 3(a)와 같이 주어진 영역에 대하여 로봇의 반지름을 고려하여 팽창시킨 configuration space를 만들고 이 C-space에 대하여, Fig. 3(b)와 같은 커버리지 경로를 생성시킨다. 생성된 경로를 추종함에 있어 dynamic window를 활용한다. Dynamic window에서 정의된 속도 후보군에 대한 새로운 경로 추종 목적함수를 설정하고 최적의 속도를 선택한다.

#### 3.1 DWA(dynamic window approach)

DWA는 실시간으로 센서로부터 장애물 정보를 받고, 로봇 하드웨어의 기구학적인 구속조건을 포함시켜 경로를 생성하는 온라인 충돌 방지 전략이다.

DWA에 사용되는 dynamic window는 Fig. 4과 같이 로봇의 기구학적인 최대 가속도를 고려하여, 단위시간 ( $\Delta t$ ) 당 최고 가속도로 도달할 수 있는 속도를 선속도와 각속도 좌표계  $(v_t, \omega)$ 에서 나타내어진 영역이다. 현재 속도를  $(v_a, \omega_a)$ 라 하고 최대 가속도

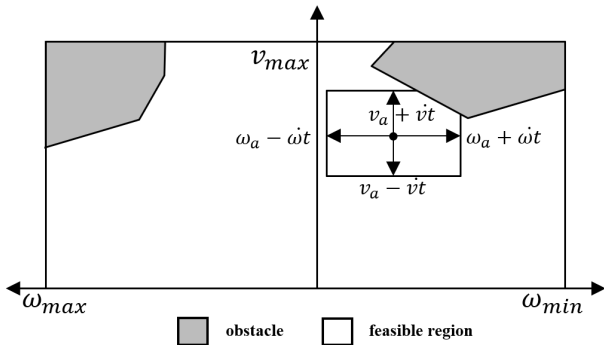


Fig. 4 Diagram of dynamic window

를 각각  $(\dot{v}, \dot{\omega})$ 라 했을 때, dynamic window는 식 (4)와 같이 나타낼 수 있다.

$$V_d = \{(v, \omega) | v \in [v_a - \dot{v}t, v_a + \dot{v}t] \wedge \omega \in [\omega_a - \dot{\omega}t, \omega_a + \dot{\omega}t]\} \quad (4)$$

DWA는 dynamic window 안에서 각각의 속도  $(v_t, \omega)$ 마다 목적 함수를 식 (5)와 같이 정의를 하게 된다.  $v_t$  는 선속도를 나타내고  $\omega$ 는 각속도를 나타낸다.

$$c(v_t, \omega) = c_{target}w_{target} + c_{vel}w_{vel} + c_{obstacle}w_{obstacle} \quad (5)$$

$c_{target}(v_t, \omega)$ ,  $c_{vel}(v_t, \omega)$ ,  $c_{obstacle}(v_t, \omega)$  은 로봇의 목표점과의 각도, 로봇의 속도, 경로 오차, 에 관한 목적 함수이고,  $w_{target}(v_t, \omega)$ ,  $w_{vel}(v_t, \omega)$ ,  $w_{obstacle}(v_t, \omega)$ 은 각각의 목적 함수의 가중치이다.

목표에 관한 목적함수  $c_{target}$  는 식 (6)으로 나타낼 수 있다. dynamic window 내에서 모든  $(v_t, \omega)$ 당 단위 시간 후의 로봇의 방향과 목표점까지의 각도 차이( $\theta$ )를  $\pi - \theta$ 로 나타낸 후  $[0, 1]$ 사이로 정규화 한다.

$$c_{target} = \frac{\pi - \theta}{\pi} \quad (6)$$

속도에 관한 목적함수  $c_{vel}$ 은 식 (7)과 같이 선 속도  $(v_t)$ 로 나타내게 된다.  $c_{vel}$  는 선속도에 비례하며 목적함수  $c_{vel}$ 의 증가가 이루어지게 되고,  $v_{max}$ 에서  $c_{vel}$ 는 1이 된다.

$$c_{vel} = \left| \frac{v}{v_{max}} \right| \quad (7)$$

장애물과의 거리에 관한 목적함수  $c_{obstacle}$  는 장애물과 로봇사이의 거리를 나타낸다. 장애물을 중심으로 거리가 늘어남에 따라 식 (8)과 같이  $c_{obstacle}$ 가 정의된다.

$$c_{obstacle} = \begin{cases} c_f & (d < r) \\ c_f e^{(-r_f(d-r))} & (r < d < r_f) \\ 0 & (r_f < d) \end{cases} \quad (8)$$

장애물로부터 멀어지면서  $c_{obstacle}$ 가 0이 되는 거리를 인플레이션 반지름 ( $r_f$ ) 이라고 한다.  $c_f$ 는 비용 스케일링 팩터다.  $d$  는 장애물로부터 떨어진 거리,  $r$  은 로봇의 반지름이다. Table 1은 실험에 사용된 obstacle parameters를 나타낸다.

DWA는 dynamic window에서 각각의 속도마다 목적함수를 정의한 후, 최적의 속도를 선택하게 되고, 이를 목적지에 도달할 때까지

Table 1 Obstacle parameters

Inflation radius	1 m
Cost scaling factor	255
Robot radius	0.15 m

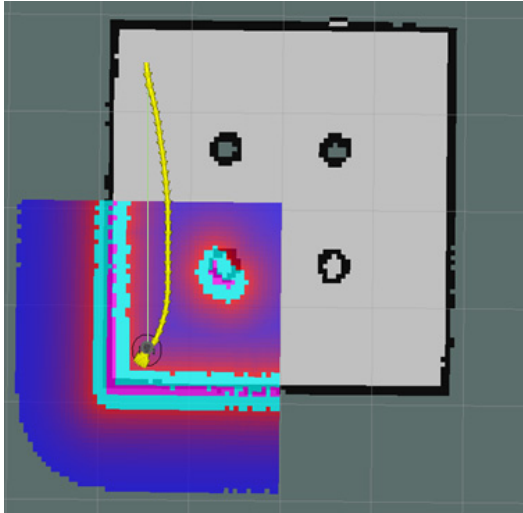


Fig. 5 Path produced by DWA

지 반복하게 된다.

Fig. 5와 같은 로봇의 경로 시작점과 목표점까지의 일직선 거리 상 장애물이 없는 환경에서, 기존의 DWA는 로봇의 현재 위치와 목표점, 장애물만을 고려한다. 따라서 실제 로봇이 택한 경로는 기존에 생성된 경로와는 상관이 없는, 장애물과 최대한 멀리 떨어지는 방향으로 움직인다. 기존의 DWA는 주어진 경로를 따라 모든 곳을 방문해야 하는 커버리지 상황보다 장애물을 자유롭게 피해야 하며 주어진 목표점에 도달하는 네비게이션 상황에 최적화 되어있다.

### 3.2 커버리지 경로 생성

커버리지 영역은 ROS의 지도 서비스를 통해 미리 주어졌다. 지도 서비스는 지도의 메타데이터와 지도의 평면도로 나뉘어지고, 실험에 사용된 지도의 해상도는 0.05 m이다.

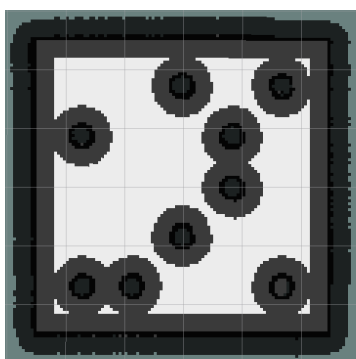


Fig. 6 Configuration space

Table 2 Coverage path planning algorithm

1. **Input** : line gap, c-space map image  
 $p = \{\}$   
 upflag = true
2. **Output** : coverage path p
3. Compute the most right down corner free space pixel  $f(x, y)$
4. Compute the most left up corner free space pixel  $f(x_f, y_f)$
5. **If** upflag
6.   **while**  $i < \text{map width}$
7.      $x = x + 1$
8.     **if**  $f(x, y)$  is free space
9.       Insert  $(x, y)$  into p
10.    **else**
11.     **for**  $i < \text{line gap}$
12.        $y = y + j$
13.       **if**  $f(x, y)$  is free space
14.         Insert  $(x, y)$  into p
15.         **break**
16.    **else**
17.     **while**  $i < x$
18.        $x = x + 1$
19.       **if**  $f(x, y)$  is free space
20.         Insert  $(x, y)$  into p
21.       **else**
22.         **for**  $i < \text{line gap}$
23.          $y = y + j$
24.         **if**  $f(x, y)$  is free space
25.         Insert  $(x, y)$  into p
26.         **break**
27.     **If**  $< y + \text{line gap}$
28.       **Return** p
29.    **while**  $k < \text{line gap}$
30.      $y = y + k$
31.     Insert into p

지도의 평면도는 점유된 공간, 자유공간, 미확인 공간 3가지 공간으로 나뉜다. 점유된 공간은 벽이나 장애물 등을 나타내고, 자유공간은 자유롭게 이동할 수 있는 공간을 나타낸다.

Fig. 6과 같이 이미지에서 나타난 점유된 공간을 로봇의 크기와 안전계수를 고려하여 팽창시킨 C-space (configuration space)를 만들었다.

C-space를 바탕으로 Table 2의 알고리즘을 통해 Fig. 7과 같은 커버리지 경로를 생성한다.

커버리지 경로는 평면도 상의 픽셀로 나타내고, 지도의 해상도를 고려하여 전역 좌표계로 식 (9)와 같이 나타냈다.

$$p_i = (x_i, y_i) \tag{9}$$

### 3.3 커버리지 DWA의 목적함수

로봇이 모든 구역을 방문해야 하는 커버리지 상황에서는, 커버리

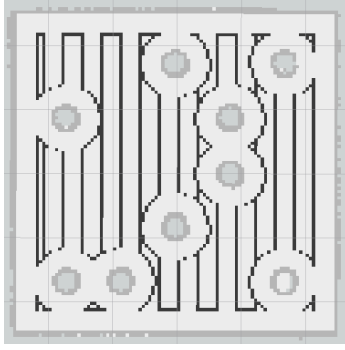


Fig. 7 Coverage path plan

지 경로 이탈은 커버리지 실패로 이어질 수 있다. 따라서 주어진 경로와 로봇의 위치를 고려하지 않는 기존 DWA는 커버리지 경로 추종 알고리즘에 쓰이기 어렵다. 본 논문에서는 기존 DWA의 목적함수를 경로 오차에 관한 항을 추가하여 커버리지 상황에 적합한 수정된 커버리지 DWA를 제안했다.

로봇의 현재 위치에서 각각의 속도 벡터  $(v_t, \omega)$ 에 따라 나타내어지는 로봇의 예상 경로는 길이가  $v_t \Delta t$  이고, 중심각은  $\pi - \omega \Delta t$  인 고유한 호로 나타낼 수 있다.

이때 호의 끝점이 다음 단위 시간의 위치를 나타내게 된다. 로봇의 현재 위치와 전역 좌표계로 주어진 커버리지 경로와 장애물 등의 위치는 이 관계를 통해 Fig. 8와 같이 전역 좌표계에서 dynamic window 상의 좌표계로 변환된다.

커버리지 DWA의 목적함수는 식 (10)으로 나타내었다. Dynamic window에 나타나는 후보 속도군 중, 목적함수 값이 가장 작은 속도를 다음 시간 간격의 속도로 정한다.

$$c(v_t, \omega)^* = c_{target} w_{target} + c_{vel} w_{vel} + c_{error} w_{error} + c_{obstacle} w_{obstacle} \quad (10)$$

장애물에 대한 목적함수  $c_{obstacle}$  은 장애물과 로봇사이의 거리를 나타낸다.  $c_{target}$  은 지역 목표점과 로봇의 방향각 차이이고,  $c_{vel}$  는 지역 목표점과 로봇의 현재 속도의 비를 의미하고,  $c_{error}$  는

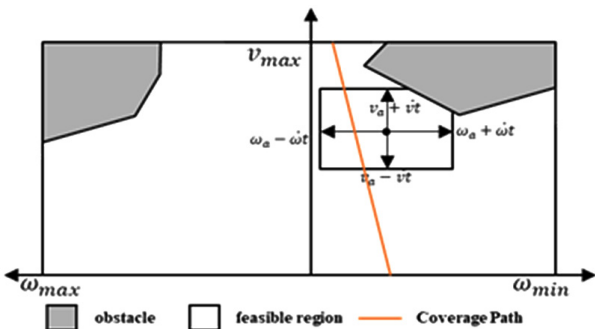


Fig. 8 Coverage path plan conversion into dynamic window

경로 오차를 나타내게 된다.

$c_{target}$  은 전역 좌표계에서의 로봇의 방향(yaw)값과 지역 목표점의 방향각의 차를 라 할 때, 식 (11)로 나타낸다.

$$c_{target} = 1 - \frac{\alpha}{\pi} \quad (11)$$

$c_{vel}$ 은 로봇의 속도에 관한 목적함수이다. 기존 속도에 관한 목적함수  $c_{vel}$ 은  $v_{max}$  에서 최대값 1을 가졌지만 커버리지 DWA 목적함수는 지역 목표점까지의 거리를 이용한다. 지역 목표점을 나타내는 선속도를  $v_{target}$ 라 하면  $c_{vel}$ 은 식 (12)와 같이 나타낼 수 있다.

$$c_{vel} = \begin{cases} 1 - \frac{v_{target} - v_t}{v_{target}} & (v_{target} < v_{max}) \\ \left| \frac{v}{v_{max}} \right| & (v_{target} > v_{max}) \end{cases} \quad (12)$$

경로 오차에 관한 목적함수  $c_{error}$ 은 커버리지 경로로부터의 거리 오차를 나타낸다. 현재 속도  $(v_a, \omega_a)$ 와 가장 가까운 커버리지 경로의 좌표를  $(v_i, \omega_i)$ 라 할 때 각 속도에 관한 거리 오차  $d$ 는 식 (13)으로 나타낼 수 있다.

$$d_i = \frac{|(v_{i+1} - v_i)(\omega_i - \omega) + (v_i - v)(\omega_{i+1} - \omega_i)|}{\sqrt{(v_{i+1} - v_i)^2 + (\omega_{i+1} - \omega_i)^2}} \quad (13)$$

경로 오차 목적함수는 거리 오차를 정규화 하여 식 (14)로 나타냈다.

$$c_{error} = \left| \frac{d}{d_{max}} \right| \quad (14)$$

커버리지 DWA는 경로 오차에 관한 목적함수  $c_{error}$ 를 통해 장애물 뿐만 아니라, 주어진 경로와의 로봇의 위치를 고려하여 경로오차를 최소화하면서 커버리지 경로 추종을 할 수 있다.

목적함수 항들의 가중치  $w_{target}(v_t, \omega), w_{vel}(v_t, \omega), w_{error}(v_t, \omega), w_{obstacle}(v_t, \omega)$  은 실험을 통해 구하였다.

## 4. 실험 결과

### 4.1 실험 평가

실험 환경은 Fig. 9와 같이 건물 복도를 변형시킨 공간에서 커버리지 실험을 진행하였다. 앞서 기술했듯이, 환경설정에서 ROS를 사용하여, AMCL를 활용한 위치 추정, 지도 서비스를 이용할 수 있었다. Lidar센서로 Ydlidar사의 TG50을 사용하였다.



Fig. 9 Experiment environment

Table 3 Test parameter of pure pursuit and coverage DWA

Parameter	Pure pursuit	Coverage DWA
Linear velocity	0.3 m/s	-
Look ahead distance	0.4 m	0.4 m

커버리지 경로는 특성상 장애물이나 벽면에 근처에 경로 생성이 된다. 기존 DWA는 경로를 따라 움직이며 모든 구역을 방문해야 하는 커버리지 상황보다 장애물을 피하며 새롭게 목표 지점까지 경로를 만들어가는 네비게이션 상황에 특화되어있다.

따라서 경로 추종에 대한 대조군으로 경로 추종에 사용되는 대표적인 알고리즘 중 하나인 pure pursuit 알고리즘을 선정했다. Pure pursuit 알고리즘은 전방주시거리에 따라 다른 특성을 보인다. 전방 주시거리가 짧은 경우, 로봇은 빠른 피드백을 받아 경로를 따라가기 때문에 경로 추종이 빠르게 이루어진다. 하지만 이는 급격한 경로 변화에서 오버슈팅이 발생할 수 있는 단점을 가지고 있다. 반면 전방 주시거리가 큰 경우, 로봇은 급격한 경로 변화에서 오버슈팅 없이 경로로 수렴하게 된다. 그러나 이 경우 로봇은 경로 안쪽으로 이동하게 되며 추종 성능이 저하될 수 있는 특징이 있다.

Pure pursuit 알고리즘과 커버리지 DWA의 파라미터를 Table 3와 같이 설정하였다. 커버리지 경로와 실제 경로의 횡방향 오차의 RMSE<sup>[19,20]</sup>와 커버리지 영역 커버율을 평가 지표로 경로 추종에 대한 평가를 한다.

Look ahead distance를 0.4m보다 더 늘릴 경우, 급격한 방향 전환에 따른 반응이 느려져 지형과 충돌하는 현상이 일어나, pure pursuit과 제안된 경로 추종방법의 look ahead distance를 0.4m로 지정하였다.

#### 4.2 장애물이 없는 환경

장애물이 없는 환경에서 제안한 경로 추종과 pure pursuit 경로 추종에 따른 이동 경로를 Fig. 10로 나타냈다. 흑색선은 생성된

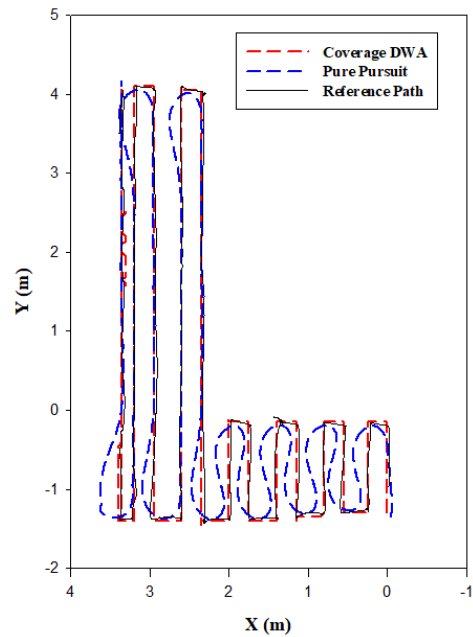


Fig. 10 Path tracking on obstacle free environment

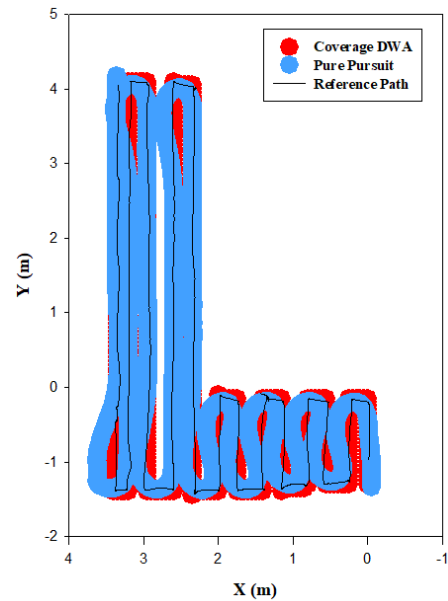


Fig. 11 Coverage area on obstacle free environment

커버리지 경로를 나타내고, 경로 추종 시 기준 경로로 사용되었다. 적색선과 청색선은 각각 커버리지 DWA와 pure pursuit의 경로 추종 시 로봇의 실제 경로다.

직선 경로 시 Fig. 11과 같이 pure pursuit와 커버리지 DWA는 경로 추종 시 횡 방향 거리 오차가 0.03 m로 유지되었다. 하지만 코너에서도 커버리지 DWA경로 추종은 횡방향 거리 오차를 0.05 m안으로 유지하는 반면, pure pursuit은 코너부분에서 횡방향 거리 오차가 0.2 m까지 증가하는 것을 Fig. 12에서 확인할 수 있다.

Table 4에 커버리지 실험에 소요된 시간과 횡방향 거리의

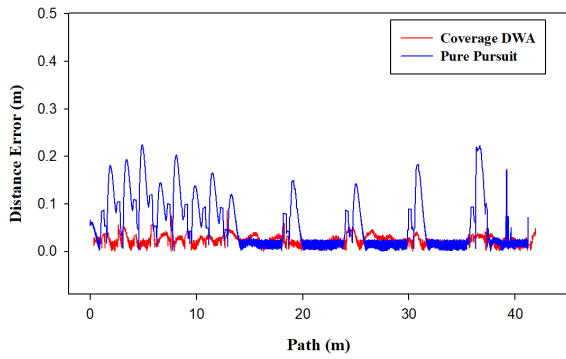


Fig. 12 Comparison of track error between coverage DWA and pure-pursuit algorithm on obstacle free environment

Table 4 Coverage time and RMSE of track error between coverage DWA and pure-pursuit algorithm on obstacle-free environment

	Coverage DWA	Pure-pursuit
Time	860 s	604 s
RMSE	0.029 m	0.077 m

RMSE를 나타내었다. 커버리지 DWA의 횡방향 거리의 RMSE가 pure pursuit 알고리즘에 비해 62% 감소했다.

Fig. 12에 pure pursuit 알고리즘과 coverage DWA 두가지 경로 추종완료 후 커버리지 영역을 나타내었다. Pure pursuit 알고리즘은 coverage DWA 경로 추종과 비교 시 모든 영역을 커버하지 못했다. 이는 pure pursuit경로 추종에 있어 코너점에서 오프셋이 일어났기 때문이다.

### 4.3 장애물이 있는 환경

커버리지 경로 생성시에도 장애물이 없는 환경보다 장애물이 존재하는 환경에서 급격한 경로 변화가 많이 생성되었다. 실험 환경은 Fig. 9와 같은 복도에 지름이 0.15 m인 장애물 5개가 있는 환경이다.

직선 경로 추종 시 pure pursuit과 커버리지 DWA는 Fig. 15와 같이 횡방향 거리 오차가 0.04 m 이내로 안정적으로 경로 추종을 수행함을 알 수 있었다.

반면 코너와 장애물 부근처럼 경로의 급격한 변화가 일어날 때, 커버리지 DWA와는 다르게 pure pursuit의 횡방향 오차가 크게 상승하는 것을 확인 할 수 있다.

Fig. 14에 두가지 경로 추종 알고리즘의 커버리지 영역을 나타내었다. Coverage DWA가 경로오차가 적은 추종 방법을 보이기 때문에, 주어진 영역을 빈틈없이 커버하는 것을 확인 할 수 있다.

Table 5에 장애물이 있는 실험에 소요된 시간과 횡방향 거리의 RMSE를 나타내었다. Coverage DWA는 장애물의 영향으로 인해 RMSE가 0.005 m 증가한 반면 pure pursuit 알고리즘은 장애

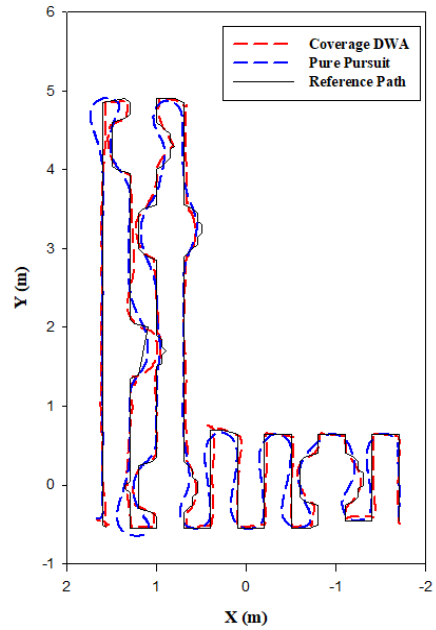


Fig. 13 Path tracking on obstacle environment

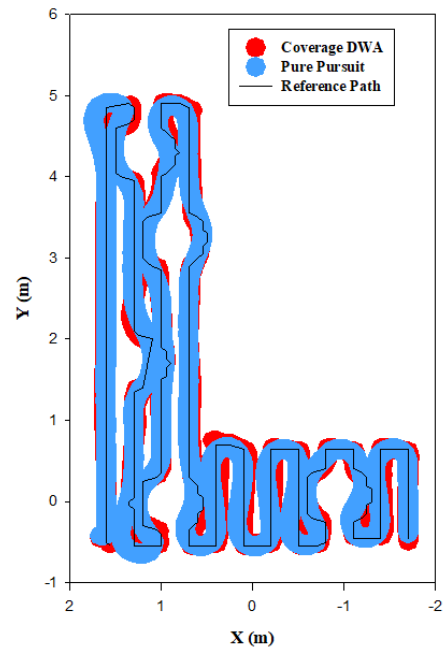


Fig. 14 Coverage area on obstacle environment

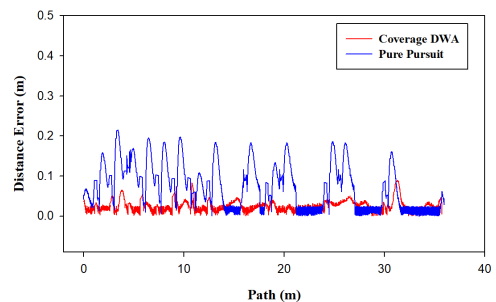


Fig. 15 Comparison of track error between coverage DWA and pure-pursuit algorithm on obstacle environment

**Table 5 Coverage time and RMSE of track error between coverage DWA and pure-pursuit algorithm on obstacle environment**

	Coverage DWA	Pure-pursuit
Time	1018 s	850 s
RMSE	0.034 m	0.095 m

물의 영향으로 인하여, RMSE가 0.018 m가 증가했다. 커버리지 DWA의 횡방향 거리의 RMSE가 pure pursuit 알고리즘에 비해 65% 감소했다.

## 5. 결론

본 논문은 로봇의 위치와 주어진 목표점 만을 고려하는 것이 아닌, 경로와 로봇 위치 간의 오차를 고려하여, 경로 이탈을 최소화하며 동시에 장애물을 고려하는 경로 추종 방법을 제안하였다. 이를 위하여 주어진 지도에 커버리지 경로를 생성하고 경로 추종 시 dynamic window를 활용하여 장애물을 고려하고, 주어진 경로를 이탈하지 않는 커버리지 DWA 경로 추종 방법을 사용하였다.

커버리지 경로를 dynamic window계의 좌표계로 변환 한 후, 로봇의 위치에 따른 지역 목표점 설정 하였다. 기존 DWA와는 다르게 경로와의 거리 오차를 고려하여, 장애물을 여유롭게 피해가 면서도 주어진 경로를 이탈하지 않는 경로 추종을 수행하였다.

실험을 통해 일직선과 같이 급격하게 변하지 않는 구간에서 커버리지 DWA 경로 추종 방법과 pure pursuit 알고리즘의 경로 추종의 거리오차가 0.06 m 안으로 유지하는 것을 확인했다. 하지만 코너나 장애물이 있는 구간처럼 경로가 급격하게 변하는 구간에서는 커버리지 DWA의 횡방향 거리의 RMSE가 pure pursuit 알고리즘에 비해 62% ~ 65%까지 감소한 것을 실험을 통해 확인 할 수 있었다. 또한 로봇의 위치와 지역 목표점만을 고려하는 pure pursuit 알고리즘보다 경로 오차까지 고려하는 커버리지 DWA를 사용 했을 때, 더 많은 영역을 커버할 수 있음을 Fig. 11와 Fig. 14에서 확인하였다.

논문에서 사용된 가중치 파라미터는 실험을 통해 설정하였으나, 향후 파라미터의 최적화를 위한 연구를 통해 알고리즘을 향상할 수 있을 것이다.

## References

- [1] Ollero, A., Heredia, G., 1995, Stability Analysis of Mobile Robot Path Tracking, IEEE International Workshop on Intelligent Robots and Systems (IROS), 5100108, <https://doi.org/10.1109/IROS.1995.525925>.
- [2] Morales, J., Martínez, J. L., Martínez, M. A., Mandow, A., 2009, Pure-pursuit Reactive Path Tracking for Nonholonomic Mobile Robots with a 2D Laser Scanner, EURASIP J. Adv. Signal Process., 2009 935237, <https://doi.org/10.1155/2009/935237>.
- [3] Coulter, R. C., 1992, Implementation of the Pure Pursuit Path Tracking Algorithm, Carnegie-Mellon Univ., Pittsburgh PA.
- [4] Wit, J. S., Crane, C. D., Armstrong, D., 2004, Autonomous Ground Vehicle Path Tracking, Journal of Robotics Systems, 21:8 439-449, <https://doi.org/10.1002/rob.20031>.
- [5] Samuel, M., Hussein, M., Mohamad, M. B., 2016, A Review of Some Pure-pursuit based Path Tracking Techniques for Control of Autonomous Vehicle, Int. J. Comput. Appl., 135:1 35-38, <https://doi.org/10.5120/ijca2016908314>.
- [6] Hellstrom, T., 2006, Follow the Past: A Path-tracking Algorithm for Autonomous Vehicles, International Journal of Vehicle Autonomous Systems, 4:2-4 216-224, <https://doi.org/10.1504/IJVAS.2006.012208>.
- [7] Wit, J. S., 2000, Vector Pursuit Path Tracking for Autonomous Ground Vehicles, Doctoral Dissertation, University of Florida, USA.
- [8] Yoon, S., Yeu, T. K., Park, S. J., Hong, S., Kim, S. B., 2012, A Simulation Study for Performance Analysis of Path Tracking Method of Follow the Carrot and Pure Pursuit, The Korean Association of Ocean Science and Technology Societies 2012 Joint Conference, 1582-1585.
- [9] Yeu, T.-K., Park, S., Hong, S., Kim, H., Choi, J., 2006, Path Tracking using Vector Pursuit Algorithm for Tracked Vehicles Driving on the Soft Cohesive Soil, SICE-ICASE International Joint Conference, 9430207, <https://doi.org/10.1109/SICE.2006.314707>.
- [10] Kim, D. H., Yu, B. G., Han, C. S., Lee, J. Y., 2008, A Study of The Safety Cornering of Nonholonomic Mobile Robot Using Vector Pursuit Path Tracking Algorithm, Proc. Korean Society of Precision Engineering Autumn Conference, 169-170.
- [11] Wang, H., Liu, B., Ping, X., An, Q., 2019, Path Tracking Control for Autonomous Vehicles based on an Improved MPC, IEEE Access, 7 161064-161073, <https://doi.org/10.1109/ACCESS.2019.2944894>.
- [12] Sun, C., Zhang, X., Zhou, Q., Tian, Y., 2019, A Model Predictive Controller with Switched Tracking Error for Autonomous Vehicle Path Tracking, IEEE Access, 7 53103-53114, <https://doi.org/10.1109/ACCESS.2019.2912094>.
- [13] d'Andrea-Novel, B., Campion, G., Bastin, G., 1995, Control of



Nonholonomic Wheeled Mobile Robots by State Feedback Linearization, *Int. J. Robotics Research*, 14:6 543-559, <https://doi.org/10.1177/027836499501400602>.

- [14] Piao, C., Liu, X., Lu, C., 2019, Lateral Control using Parameter Self-tuning LQR on Autonomous Vehicle, *Int. Conf. Intelligent Computing, Automation and Systems (ICICAS)*, 19495174, <https://doi.org/10.1109/ICICAS48597.2019.00197>.
- [15] Fox, D., Burgard, W., Thrun, S., 1997, The Dynamic Window Approach to Collision Avoidance, *IEEE Robot. Autom. Mag.*, 4:1 23-33, <https://doi.org/10.1109/100.580977>.
- [16] Ogren, P., Leonard, N. E., 2005, A Convergent Dynamic Window Approach to Obstacle Avoidance, *IEEE Trans. Robot.*, 21:2 188-195, <https://doi.org/10.1109/TRO.2004.838008>.
- [17] Malu, S. K., Majumdar, J., 2014, Kinematics, Localization and Control of Differential Drive Mobile Robot, *Global Journal of Researches in Engineering: H*, 14:1 1-8.
- [18] Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A., 2009, ROS: An Open-source Robot Operating System, *ICRA Workshop on Open Source Software*.
- [19] Lozano-Pérez, T., 1983, Spatial Planning: A Configuration Space Approach, *IEEE Trans. Comput.*, C-32:2 108-120, <https://doi.org/10.1109/TC.1983.1676196>.
- [20] Zakaria, M. A., Zamzuri, H., Mamat, R., Mazlan, S. A., 2013, A Path Tracking Algorithm using Future Prediction Control with Spike Detection for an Autonomous Vehicle Robot, *Int. J. Adv. Robot. Syst.*, 10:8 309, <https://doi.org/10.5772/56658>.



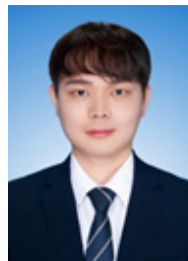
**Hyun-suk Lee**

Master Course in the Department of Mechanical Engineering, Sogang University. His research interest is Mobile robot, ROS.  
Email: leessd2004@gmail.com



**Kyung-hyun Yoo**

Master Course in the Department of Mechanical Engineering, Sogang University. His research interest is Mobile Robot, ROS.  
Email: bggh9395@naver.com



**Seung-won Lee**

Master Course in the Department of Mechanical Engineering, Sogang University. His research interest is Robotics.  
Email: seuungwon426@naver.com



**Jong-min Kim**

Research Professor in the Department of Mechanical Engineering, Sogang University. His research interests are Robot System and Control and Radiopharmaceutical Automatic Manufacturing System.  
Email: griffon0@gmail.com



**Chul-soo Lee**

Professor in the Department of Mechanical Engineering, Sogang University. His research interest is CNC Controller, Geometric Modeling and CAM, Robot & Automation System.  
Email: cscam@sogang.ac.kr