

&lt;학술논문 or 응용논문 택1&gt;

DOI <http://dx.doi.org/10.3795/KSME-AB.2014.38.0.000>

ISSN 1226-0000

2288-0000(Online)

# GVG를 이용한 복도 환경에서의 180° 센서 기반 모바일 로봇 탐색 알고리즘 설계

황지만\* · 이성학\* · 권혁삼\*\*

\* 성균관대학교 기계공학부

## Design of Sensor-based Mobile Robot Exploration Algorithm in Corridor Environment using Generalized Voronoi Graph with 180° sight

Jiman Hwang\*, Sunghak Lee\*, and Hyuksam Kwon\*

\* Dept. of Mechanical Engineering, Sungkyunkwan Univ.

(Received January 1, 2014 ; Revised January 2, 2014 ; Accepted January 3, 2014)

**Key Words:** Generalized Voronoi Graph(일반 보로노이 그래프), Sensor-based(센서 기반), Navigation Algorithm(탐색 알고리즘)

**초록:** 거리 센서를 장착한 모바일 로봇을 이용하여 지도가 주어지지 않은 지역을 탐색하고, 해당 지역에 대한 지도를 만들어낼 수 있다. 이전 연구에서는 거리 센서로 로봇을 360°로 둘러싸아 로봇으로 하여금 앞쪽 영역과, 뒤쪽 영역 모두 탐지할 수 있게 하였다. 로봇의 뒤쪽에 대한 정보는 상당히 유용하게 쓰이는데 모퉁이와 교차로에서 중요한 역할을 한다. 본 논문에서는 로봇의 가격을 낮추고 전력 소모량을 줄일 수 있도록, 로봇의 앞쪽 180°에만 거리 센서를 장착하여 모르는 영역을 탐색한다. 뒤쪽 시야의 부재 때문에 발생하는 문제점을 소개하고, 해결책을 제안한다. 또한 사용자가 로봇의 최대, 최소 속력을 지정하여 로봇의 안정적인 주행이 가능하도록 하였다.

**Abstract:** A mobile robot equipped with range sensors is able to explore unknown areas, building the entire map data in its memory or file. The prior works have taken advantage of a ring of sensors, which surrounds the robot usually, so that the robot has backward area as well as forward one in its sight. The backward sight is critical, especially at corners and intersections. In this paper, a half of ring of sensors, which reduces the cost of robot and power consumption, is used to explore unknown areas. A challenge derived from the limited view is introduced and its solution is also suggested in this paper. Moreover, we suggest a method that a user can specify the maximum and minimum speed of robot explicitly, which helps a stable drive.

### 1. 서 론

센서 기반의 탐색은 이전 연구<sup>(1~3)</sup>에서 활발히 이루어 졌는데, 이는 로봇이 전방뿐만 아니라 후방도 탐지가 가능한 로봇에 국한되었다. 이 후방에 대한 정보는 로봇이 모퉁이를 돌때나 교차로를 지날 때 쓰이며, 특히 GVG에서는 노드를 판단하는데 사용된다. 로봇이 후방에 대한 정보를 얻지 못 할 때에는 노드를 판단하는 방법이 이전과<sup>(3)</sup>는 약간 다르게 되는데, 이는 4.1절에서 자세히 설명하였다.

또한 만들어진 GVG위로 로봇을 움직이는 control law<sup>(4)</sup>는 로봇이 거리 센서로부터 읽은 데이터가 주어졌을 때, 로봇이 GVG위로 가기 위해서 가야 하는 속도를 제시한다. 하지만 제한된 시스템은 오차가 심하고, 때에 따라서는 갑자기 속력이 크게 상승해 안정적인 주행을 하지 못한다. 3장에서는 안정적인이고, 적은 오차를 갖는 시스템을 소개한다.

### 2. 이전 연구

GVG에 대한 사전 지식은 이전 자료에서 충분히 찾을 수 있으므로 자세한 설명은 생략하기로 한다.

### 2.1 Control Law

로봇이 거리 센서로부터 읽는 값들로부터 로봇이 어느 방향으로 얼마만큼의 속력을 내야 하는지를 알려주는 것이 Control law이며 다음과 같이 주어진다.

$$\dot{x} = \alpha \text{Null}(\nabla G(x)) + \beta (\nabla G(x))^+ G(x) \quad [1]$$

단,  $\alpha, \beta$ 는 스칼라(scalar) 상수,

$$G(y, \lambda) = \begin{bmatrix} d_1(y, \lambda) - d_2(y, \lambda) \\ \vdots \\ d_1(y, \lambda) - d_m(y, \lambda) \end{bmatrix},$$

$\text{Null}(\nabla G(x))$ 은  $\nabla G(x)$ 의 null space,

$d_i$ 는 점  $(\lambda, y)$ 에서  $i$ 번째로 가까운 장애물의 거리함수 값이다.

이전에는<sup>(4)</sup> 다음과 같은 조건으로 구현하였다.

$$\alpha = \frac{1}{\|\nabla G(x)\|}, \quad \beta = -1 \quad [2]$$

그 결과 직선구간에서 로봇의 경로가 다음과 같이 나왔다.



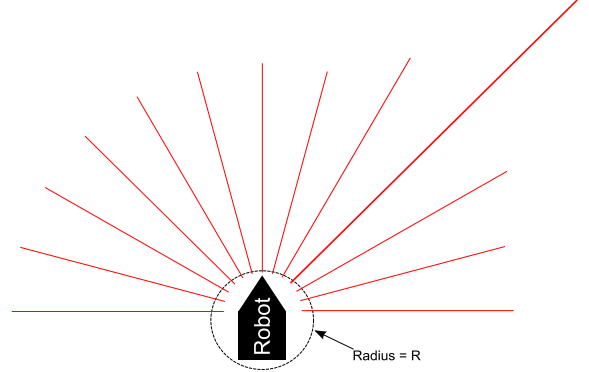
**Fig. 1** Wave phenomenon in control law

또,  $\|\nabla G(x)\| \rightarrow 0$ 일 때, 로봇의 속력이 크게 증가하므로 안정적인 주행을 보장하지 못한다.

## 3. The New Control Law

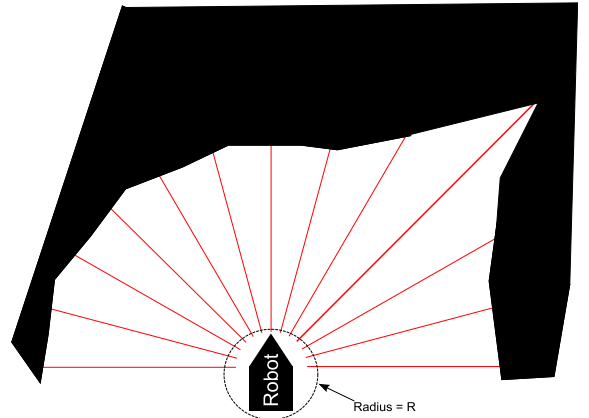
### 3.1 장애물의 판별

센서가 주는 값으로 거리함수 값을 계산하려면 읽은 값 중에 어느 값들이 하나의 장애물에 있는지를 알아야 한다.

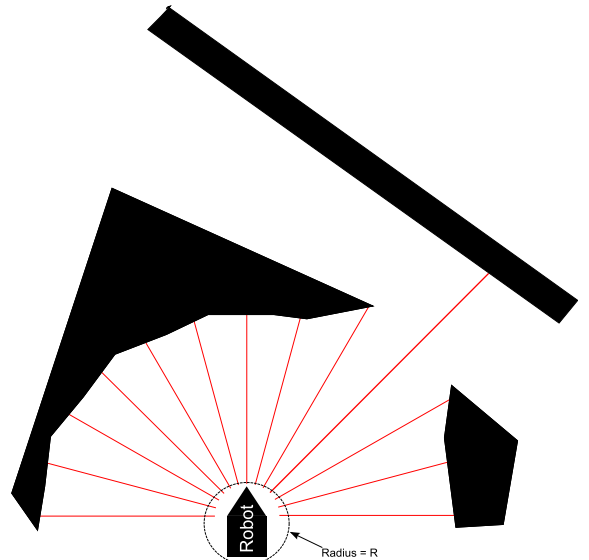


**Fig. 2** Sample robot's sight

예로, 위와 같이 장애물을 인지되었다고 하면, 로봇은 아래 그림들처럼 실제 장애물이 어떻게 구성되어있는지 알 수가 없다.



**Fig. 3** Obstacle arrangement 1



**Fig. 4** Obstacle arrangement 2

이를 해결하기 위해 아래와 같이 로봇지름( $D_R$ )을 정하고, 가정을 설정한다.

로봇지름( $D_R$ ) : 로봇에 외접하는 가상의 원보다 살짝 큰 원의 지름.

가정 : “두 연속된 센서로부터 읽은 점간의 거리가 로봇지름보다 작으면, 두 점은 하나의 장애물 위에 있다.”

센서로 읽은 두 좌표의 거리가 로봇지름보다 짧으면 로봇은 그 둘 사이를 지나갈 수 없으므로, 그 두 점을 하나의 장애물 위에 있다고 보아도 무방하다.

### 3.2 웨이브 현상 제거

#### 3.2.1 GVG와 로봇의 어긋난 정도 측정

로봇의 이동경로가 휘어서 나오는 이유는 거리 센서가 주는 정보를 제대로 활용하지 않아서이다. 특히 장착된 센서의 개수가 적을수록 이 현상은 심해진다. 본 논문에서는 기존 거리 센서를 그대로 사용하되, 활용할 수 있는 정보를 뽑아내어 GVG알고리즘 적용에 사용하도록 하였다.

$H_1$ 은 거리센서가 주는 값의 수가 적을 때, 부족한 정보를 보충하는 용도로 쓸 수 있다.

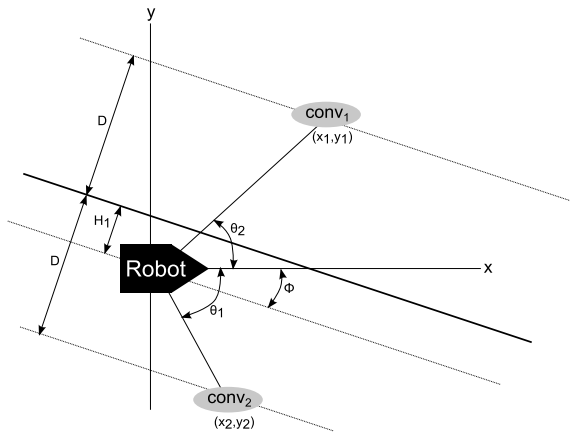


Fig. 5 Calculation of  $H_1$

$H_1$ 을 식으로 나타내면 다음과 같다.

$$H_1 = \frac{1}{2}(-x_1 \sin \phi + y_1 \cos \phi - x_2 \sin \phi + y_2 \cos \phi) \quad [3]$$

단,  $\phi = (\theta_1 + \theta_2)/2$

$H_2$ 는 가장 인접한 두 장애물이 GVG edge를 사이에 두고 대칭에 놓여 있다고 가정한 뒤, 로봇부터 그 edge까지의 거리를 나타낸다. 인접한 두 장애물과 로봇이 이루는 각이 커질수록  $H_2$ 는 edge와 로봇간의 실제 거리 값에 근접하게 되는데, 이는 직선으로 된 복도에서 흔히 발생하는 현상이다.

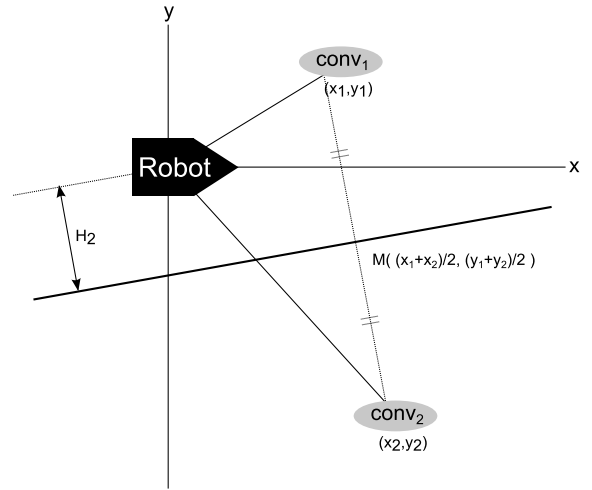


Fig. 6 Calculation of  $H_2$

$H_2$ 를 식으로 나타내면 다음과 같다.

$$H_2 = \frac{\frac{1}{2} |x_2^2 - x_1^2 + y_2^2 - y_1^2|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad [4]$$

이 값은 [8]에서  $\psi$ 를 계산하는데 쓰인다.

$H_1$ 을 이용하여 GVG로부터 어긋난 정도를 다음과 같이 정의하였다.

$$E = \phi + k_1 \frac{H_1}{D_R} \quad [5]$$

단,  $\phi = (\theta_1 + \theta_2)/2$ ,  $k_1$ 은 양의 실수,  $D_R$ 은 로봇의 직경이다.

#### 3.2.2 안정적인 주행을 위한 설계

$E$ 를 이용하여 로봇이 GVG edge와 가까우면 속력은 빠르고 회전 각도는 작게 하며, 반대로 멀리 떨어져있다면 속력은 줄이고 회전 각도를 크게 하여 edge로 수렴이 용이하도록 하였다. 정규분포와 sigmoid함수를 활용하여 로봇의 속력, 회전각의 범위를 제한하고, 계산된 값이 갑작스

렵게 크게 바뀌는 현상을 없애고자 하였다.

$$speed(E) = (v_{\max} - v_{\min}) \exp(-k_2 E^2) + v_{\min} \quad [6]$$

$$angle(E) = \frac{2\theta_{\max}}{1 + \exp(-\psi E)} - \theta_{\max} \quad [7]$$

단,  $v_{\max}$ ,  $v_{\min}$ ,  $\theta_{\max}$ 는 각각 사용자가 입력한 최대 속도, 최소 속도, 최대 회전 각도이며,  $k_2$ 는 양의 상수이다.

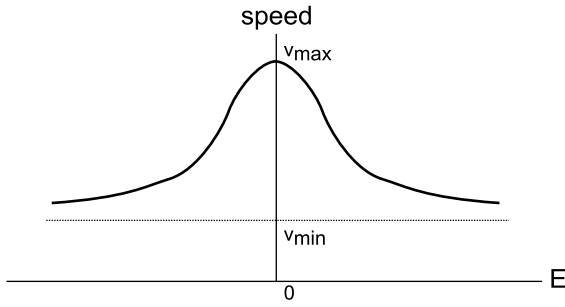


Fig. 7 Speed vs  $E$

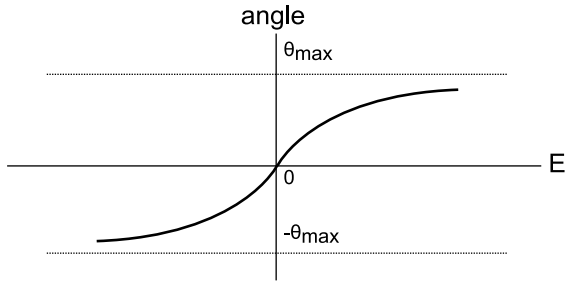


Fig. 8 Angle vs  $E$

$\psi$ 는 각도 변화의 민감도를 나타내며 아래와 같이 주어진다.

$$\psi = (\psi_{\max} - \psi_{\min}) \exp\left(-k_3 \frac{H_2}{D_R}\right) + \psi_{\min} \quad [8]$$

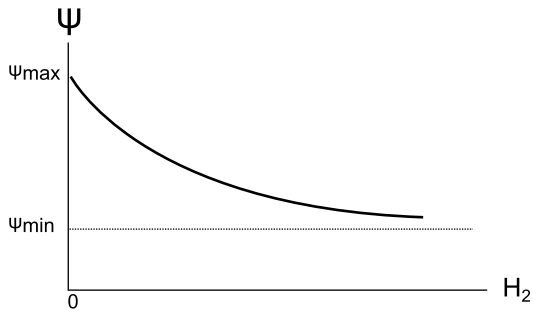


Fig. 9  $\psi$  vs  $H_2$

복도에서 edge를 따라 주행할 때는  $H_2$ 가 작아지고,  $\psi$ 가 커지게 되어  $E$ 가 조금 변하더라도 로봇의 각도는 크게 바뀌게 된다. 반대로 직선이

아닌 코너에서는  $H_2$ 가 커지고,  $\psi$ 가 작아짐에 따라  $E$ 의 작은 변화에도 로봇의 각도는 크게 변하지 않는다. 이 차이는 직선구간에서의 웨이브 현상을 없애고 코너에서의 안정적인 주행을 돕는다.

## 4. 노드 탐색 정책

### 4.1 180도 센서의 제한

시중에 판매되고 있는 모바일 로봇 중에서 거리센서가 로봇을 360도로 둘러싸고 있어 전방뿐만 아니라 후방도 탐지할 수 있는 제품도 있지만, 그 절반인 180도 전방만 탐지할 수 있는 제품도 있다.

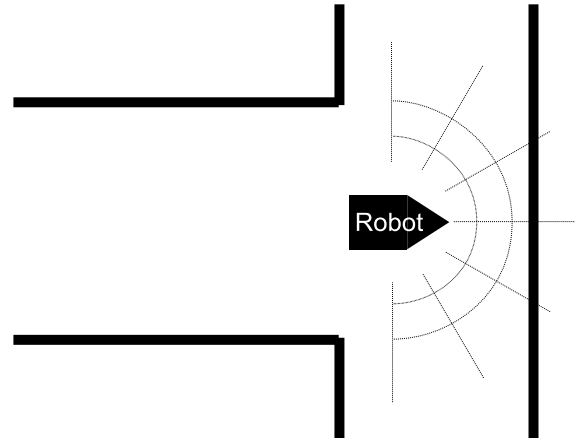


Fig. 10 Node detection failure

후방 탐지가 불가능한 로봇은 노드에 진입하기 전에 미리 앞에 노드가 있음을 탐지해야 한다. 위의 그림은 로봇이 앞에 노드가 있음을 알지 못하고 노드의 위치까지 움직인 모습이다. 이 경우 로봇이 노드상에 있다는 것을 판별하기 위해서는 360도 범위로 거리를 측정할 필요가 있다. 하지만 전방만 측정이 가능하다면, 노드를 발견할 수 없다. 여기서는 후방 탐지가 불가능한 로봇이 전방의 노드를 미리 파악하고, 발견한 노드에 도착했을 때의 행동 정책을 제안한다.

### 4.2 노드의 탐지

2차원 상에서, 어떤 한 점을 기준으로 가장 가까운 세 개의 장애물과의 거리가 서로 같을 때, 그 점을 노드라고 한다. 로봇의 앞에 노드가 있는지를 검사하기 위해서 로봇 앞에 일정거리

$DT$ 만큼 떨어진 가상의 점  $P_D$ 를 찍는다. 이 점  $P_D$ 는 로봇의 병진 운동과 회전 운동에 상관없이 로봇의 관점에서는 항상 일정한 위치에 있게 된다.

거리 센서로부터 데이터를 읽을 때마다 다음 과정을 수행하여 노드를 찾는다. 과정에서 나오는 “좌표”는 로봇을 기준으로 한다.

- 1) 센서로 읽은 점들 중에서 점  $P_D$ 를 기준으로 가장 가까운 3개의 점을 뽑는다.  
이 세 개의 점은 각각 다른 장애물상의 점이어야 한다.
- 2) 1)에서 뽑은 세 개의 점( $A, B, C$ )을 꼭지점으로 하는  $\triangle ABC$ 의 외심의 좌표를  $P_N$ 이라고 한다.
- 3)  $\overline{P_D P_N}$ 이 일정 문턱 값(threshold)보다 작으면  $P_N$ 을 노드로 설정한다.

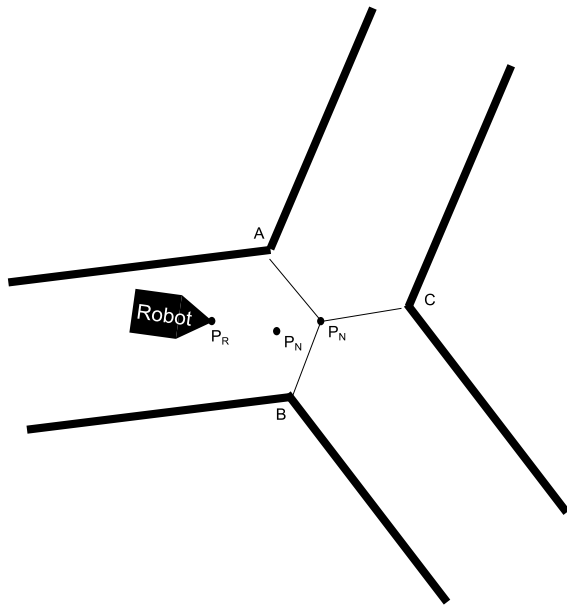


Fig. 11 Node detection

노드가 발견된 후 로봇이  $P_N$ 을 중심으로 하고 반지름이  $R$ 인 원 안에 있을 때, 로봇이 노드 위에 있는 것으로 간주한다. 복도의 폭에 따라 적절히 설정하여야 한다.

#### 4.3 노드 위에서의 행동

로봇이 노드상에 있을 때 다음 길(edge)을 선택하는 방법으로 다음 두 가지 정책 중 하나를 선택한다.

- 1) 무조건 오른쪽 길로 가기

#### 2) 무조건 왼쪽 길로 가기

이를 위해 노드 위에서는 회전 하고자 하는 방향의 반대쪽 센서 데이터를 지운다. 예를 들어 1)의 정책을 선택하였다고 하자. 로봇이 노드에 들어서면 로봇의 왼쪽에 장착된 거리센서부터 받은 모든 값들을 무효화시켜 로봇이 오른쪽 시야만 볼 수 있게 한다. 이제 남은 오른쪽 방향의 데이터와 3절에 설명한 알고리즘을 사용하면 로봇은 오른쪽으로 향하게 된다.

#### 4.4 노드의 해제

로봇이 노드를 탐지하고, 그 노드를 통과한 후엔 지나간 노드를 해제시켜줘야 한다. 물론 해제시키지 않고 메모리상에 노드 좌표를 저장하여 차후 로봇이 이 길로 다시 돌아왔을 때 재활용해도 되지만, 그렇게 하지 않는 편이 좋다. 주행을 하면서 엔코더로 계산한 로봇의 위치에 오차가 쌓이면, 로봇이 저장해두었던 노드 좌표에 있지 않는데 로봇의 좌표가 노드 좌표에 있어서 노드로 잘못 판별할 수 있고, 또 반대의 경우도 발생할 수 있다. 로봇의 위치를 오차 없이 정확히 판단하기는 힘들기 때문에 지나간 노드는 해제한 후, 다시 이 길을 찾았을 때 한 번 더 노드를 판별하는 것이 안전하다.

**Algorithm. 1** Node Detection & Invalidation

Input:  $R$ (predefined radius of node),  
robot's location

Output: the status of detected node and whether  
the robot is on the node

---

```

1 static isNodeValid  $\leftarrow$  FALSE
2 static isOnNode  $\leftarrow$  FALSE
3 if isNodeValid == TRUE
4    $D$  = distance between the node and robot
5   if  $D > R$  and isOnNode == TRUE
6     isNodeValid  $\leftarrow$  FALSE
7     isOnNode  $\leftarrow$  FALSE
8   else if  $D \leq R$ 
9     isOnNode  $\leftarrow$  TRUE
10 else
11   Try to discover a new node
12   if node is discovered
13     isNodeValid  $\leftarrow$  TRUE
14 return isOnNode

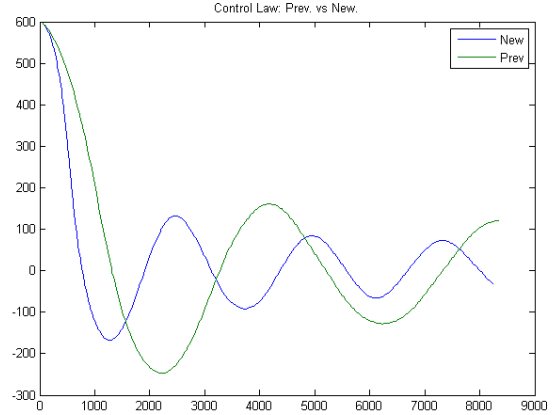
```

---

위의 의사코드(pseudo-code)는 노드를 발견, 해제하는 함수이다. 거리센서에서 값을 읽을 때 마다 이 함수를 실행한다. static 변수인 isNodeValid와 isOnNode는 C언어에서의 static 변수처럼 함수를 실행 후 빠져나와도 그 값이 보존되며, 초기 값은 둘 다 FALSE이다. isNodeValid는 노드가 발견 되었는지를 나타내는 변수이며 발견했을 시 TRUE, 아니면 FALSE이다. isOnNode는 로봇이 발견한 노드 위에 있는지를 나타내며, 노드 위에 있으면 TRUE, 아니면 FALSE이다. isOnNode가 TRUE이면 isNodeValid는 반드시 TRUE이다. 로봇과 노드간의 거리가 노드의 반지름  $R$ 보다 크고, isOnNode가 TRUE라면, 로봇은 노드를 빠져나가는 중이라는 뜻이므로 노드를 해제시킨다(isNodeValid = isOnNode = FALSE).

## 5. 실험

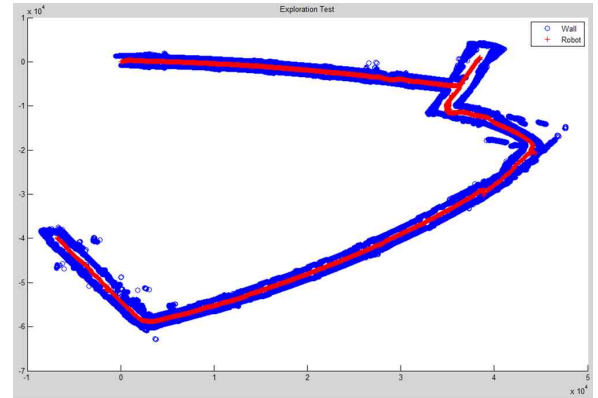
### 5.1 시뮬레이터



**Fig. 12** Previous vs New Control Law

이전 연구에서 제시된 control law와 3장에서 제안한 새로운 control law와의 비교이다. Peak time, Maximum Overshoot 등에서 기존 방법에 비해 새로운 방법이 더 좋은 결과를 나타내고 있다.

### 5.2 실제 환경



**Fig. 13** Real machine experiment

실제 건물의 한 층을 탐색한 결과이다. 로봇이 회전한 처음 가동될 때를 기준으로 회전한 각도를 바퀴에 달린 엔코더로 측정하는데, 이때의 오차가 탐색을 계속하면서 쌓여, 전체 경로가 휘어져 나왔다.

## 6. 결론

새로운 control law를 작성함으로써 웨이브 현상을 감소시키고 안정적인 주행이 가능해졌고, 또 180°의 시야만으로도 주행이 가능하다.

참고문헌  
(References)

- (1) YUAN, Quan; LEE, Ji Yeong; HAN, Changsoo., 2011, "Sensor-based navigation algorithm for car-like robot to generate completed GVG.", In: *Control, Automation and Systems (ICCAS), 2011 11th International Conference on. IEEE*, p. 1442-1447.
- (2) NAGATANI, Keiji; IWAI, Yosuke; TANAKA, Yutaka., 2011, "Sensor Based Navigation for car-like mobile robots using Generalized Voronoi Graph.", In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on. IEEE*, p. 1017-1022.
- (3) NAGATANI, Keiji; CHOSET, Howie; THRUN, Sebastian., 1998, "Towards exact localization without explicit localization with the generalized voronoi graph.", In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. IEEE*, p. 342-348.
- (4) CHOSET, Howie; KONUKSEVEN, Ilhan; RIZZI, Alfred., 1997, "Sensor based planning: A control law for generating the generalized Voronoi graph.", In: *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on. IEEE*, p. 333-338.