

## 화재 진압을 위한 자율주행 로봇 개발

조 현 습<sup>1)</sup> · 서 석 현<sup>\*1)</sup>

한국공학대학교 전자공학부<sup>1)</sup>

## Development of Autonomous Driving Robot for Fire Suppression

Hyunseup Jo<sup>1)</sup> · Sukhyun Seo<sup>\*1)</sup>

<sup>1)</sup>Department of Electronics Engineering, Tech University of Korea, 237 Sangidaehak-ro, Siheung-si, Gyeonggi, Korea

**Abstract** : As of 2020, there were about 40,000 fires in Korea, resulting in the loss of over 2,000 people and approximately 600 billion won in property damage. In particular, in industrial facilities such as factories, it is easy to spread as a large-scale fire, and the damage is enormous. Therefore, in the event of a fire, it is very important to suppress the initial fire so as not to cause a bigger accident. However, when a fire breaks out, people inside the building start to get confused, which can cause the fire to spread and cause property damage and personal injury. Therefore, for the initial detection and suppression of fire on behalf of humans, autonomous driving to the point of fire and the ability to fire a throwing fire extinguisher at the recognized fire point are required.

The robot manufactured for this purpose creates a map of the driving area in advance using Hector SLAM, an open source based on 2D lidar for indoor location recognition, and estimates the current location through sensing data.

In addition, a fire detector capable of communication was produced in order to recognize fire and register the data in the server in real time. The robot subscribes to the fire data and finally sets the waypoint of the point where the fire detector is installed as the destination, and when it autonomously moves to the point, it recognizes the fire and launches a fire extinguisher.

**Key words** : SLAM(동시적 위치 추정 및 지도작성), Mobile Robot(모바일 로봇), PI Control(PI 제어)

### Nomenclature

$v$  : linear velocity, m/s

$w$  : angular velocity, rad/s

$tr$  : target rpm, rpm

$cr$  : current rpm, rpm

$r$  : wheel radius, m

$RW$  : robot width, m

$ad$  : average traveled distance, m

$ld$  : left wheel traveled distance, m

$rd$  : right wheel traveled distance, m

$\theta$  : theta, rad

### Subscripts

ppr: pulses per revolution

ppm: pulses per meter

$x_r$  : relative x-coordinate

$y_r$  : relative y-coordinate

## 1. 서론

2020년 기준 우리나라에서는 약 4만 건의 화재로 2000여 명의 인명피해와 6000억 가까이 되는 재산 피해가 발생했다. 특히 공장과 같은 산업시설에서는 대형화재로 번지기 쉬워 그 피해가 막대한 상황이다. 따라서 화재 발생 시 더 큰 사고가 발생하지

---

\* 서석현, E-mail: shseo@tukorea.ac.kr

않도록 초기진압이 매우 중요하다. 하지만 화재가 발생하면 건물 내부의 사람들은 혼란이 오기 시작하고 이로 인해 화재가 번져 재산피해, 인명피해를 일으킬 수 있다. 따라서 사람을 대신하여 화재의 초기 감지 및 진압을 위해선 화재 발생 지점까지 자율 주행, 인식된 화재지점에 투척용 소화기를 발사하는 기능이 필요하다.

이를 위해서 제작된 해당 로봇은 실내 위치 인식을 위해 2D 라이다 기반의 오픈소스인 Hector SLAM<sup>1)</sup>을 이용하여 주행 영역에 대한 지도를 사전에 생성하고 센싱 데이터를 통해 현재 위치를 추정한다.

또한, 화재 인식 및 실시간으로 해당 데이터를 서버에 등록하기 위해 통신이 가능한 화재 감지기를 제작하였다. 로봇에서는 화재 데이터를 구독하고, 최종적으로 화재 감지기가 설치된 지점의 waypoint를 목적지로 설정하여 해당 지점까지 자율적으로 이동하면 화재를 인식 후 투척용 소화기를 발사하는 로봇을 제작하였다.

## 2. 시스템 구성

### 2.1 HW 구성

제작된 로봇의 시스템 구성은 Fig.1과 같다. 센서로는 이동 거리와 속도를 측정하기 위한 엔코더 모터 2대, 주행 환경 인식을 위한 2D 라이다 1대, 화재 인식 및 거리 추정을 위한 USB 카메라 1대가 장착되어있다. 또한, 센서 처리를 위한 메인 프로세서인 Jetson Xavier NX와 모터 제어를 수행함과 동시에 엔코더 데이터를 수신하는 하위 제어기인 Arduino Mega 2560으로 구성하였다.

화재 감지기의 시스템 구성은 Fig.2와 같다. 화재 발생 여부를 실시간으로 감지하고 로봇에게 MQTT 통신을 이용하여 화재 감지기 위치 좌표를 전송하기 위해 라즈베리파이를 사용하였다. 또한, 불을 감지하기 위한 화재 감지 센서, 경고음을 발생시켜 위험을 인식하게 하기 위한 부저가 사용되었다.

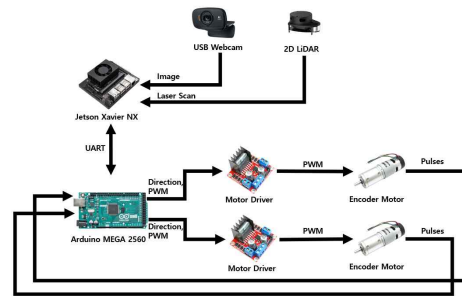


Fig. 1 Robot System Diagram

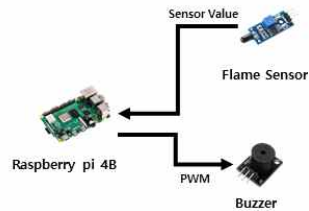


Fig. 2 Fire Detector System Diagram

### 2.2 프레임 구성

프레임은 Fig.3와 같다. 교내 건물의 통로 중 가장 좁은 폭을 고려하여 화재 초기진압 중 사람이 대피한다는 상황을 가정하고 대피경로를 방해하지 않는 크기로 제작하기 위해 너비는 0.375m, 길이는 0.435m, 높이는 0.420m와 같이 설계하였다.

LiDAR와 Camera의 배치는 로봇의 주시 방향과 일치시키기 위해 각각 상단 중앙, 전면 중앙에 배치하였다. LiDAR의 경우 전방 120도, 최대 인식 거리 5m 이하만으로 ROI를 설정하여 연산량을 줄이고 필요 없는 데이터인 후면부 영역을 제거하였다. 액추에이터인 모터의 배치는 전륜구동으로 구성하여 별도의 조향 시스템을 제작하지 않고 좌측 전륜과 우측 전륜의 회전수 차이에 따라 조향을 조절하는 차동 방식으로 조향 및 회전 동작을 수행하게 하였다.

마지막으로 투척용 소화기를 발사하기 위한 발사대의 구성은 Fig.4와 같다. 원통 형태의 PVC 파이프에 스프링을 부착하고 압축된 스프링을 서보모터가 밀어내어 투척용 소화기를 발사하게 하였다.

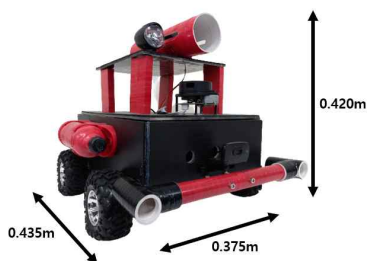


Fig. 3 Robot frame



Fig. 4 Fire-extinguisher launcher

### 3. 위치 추정 및 이동 거리 계산

실내 환경에서 로봇이 지정된 목적지까지 자율적으로 주행하기 위해서는 지도 작성(SLAM), 위치 인식(Localization), 경로 계획(Path planning), 장애물 회피(Obstacle avoidance)의 과정이 필요하다.<sup>2)</sup>

#### 3.1 지도 생성

로봇의 주행 환경은 교내 건물인 산학 융합관으로 선정하였다. 이에 따라 직접 생성한 맵은 Fig.5와 같다. 우측 상단의 빨간색 지점을 로봇이 출발하는 시작점으로 설정하고, 상대적으로 하단에 위치한 지점을 화재 발생 지점으로 사전에 waypoint를 지정하여 해당 지점까지 주행하도록 하였다.

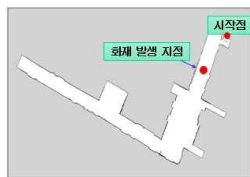


Fig. 5 Map

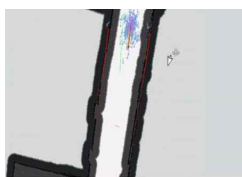


Fig. 6 Path planning

#### 3.2 경로 계획 및 위치 추정

본 논문에서는 ROS의 오픈소스 패키지인 AMCL(Adaptive Monte Carlo Localization)을 이용하여 현재 로봇의 위치 추정을 진행했다. AMCL은

LiDAR와 엔코더를 이용해 로봇의 위치와 자세 정보를 파티클 필터 알고리즘을 통해 현재 로봇의 위치를 추정한다.<sup>3)</sup> 또한, 장애물 회피를 위해 DWA(Dynamic Window Approach) 알고리즘을 이용하였다. 이는 지역 경로 계획으로 이동체의 속도, 방향 및 센서 정보로부터 얻는 장애물과의 거리를 토대로 최적의 선속도와 각속도를 도출하여 장애물을 회피하고 목적지에 도달하는 방법이다.<sup>4)</sup>

##### 3.2.1 Odometry 변환

AMCL의 엔코더 데이터 입력은 Odometry 변환이 필요하다. Odometry란 주행 거리를 뜻하며 GPS와 같이 절대적인 위치가 아닌 출발 지점으로부터 얼마나 이동했는가를 측정하는 상대적인 위치를 추정하는 방식이다. 이는 엔코더의 펄스를 카운트하여 이동 거리를 추정할 수 있다.

$$ld = rd = \frac{pulses_{curr} - pulses_{prev}}{ppr} \quad (1)$$

이는 현재 엔코더 값과 이전 엔코더 값을 1m 당 발생하는 펄스의 개수로 나눈 값이며 좌측, 우측 바퀴 모두 동일하게 적용된다.

$$ad = \frac{ld + rd}{2} \quad (2)$$

두 바퀴의 이동 거리를 평균을 내어 평균 이동 거리를 계산한다.

$$\theta = \frac{rd - ld}{RW} \quad (3)$$

작은 각도의 경우 대략적인 로봇의 회전 각도를 추정할 수 있다. 따라서 로봇의 상대적인 이동 위치는 직교 좌표계로 변환했을 때 상대 좌표는 다음과 같다.

$$x_r = \cos(\theta) * ad \quad (4)$$

$$y_r = -\sin(\theta) * ad \quad (5)$$

이를 토대로 현재 지도상에서의 위치를 추정하기 위해 전역 좌표로 변환하기 위해서 엔코더 데이터의 발행주기마다 상대 좌표를 누적 합하여 로봇의 위치를 추정하였다.

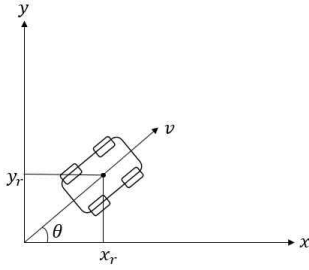


Fig. 7 Robot coordinate plane

### 3.3 명령 속도 RPM 계산

DWA를 통해 최종적으로 도출된 속도 명령은 선속도  $v$ 와 각속도  $w$ 로 구성되어 있다. PI 제어를 위해 이를 목표 rpm으로 변환한다.

$$rpm_{target} = \frac{velocity * 60}{2\pi r} \quad (8)$$

여기서 선속도  $v$ 와 각속도  $w$ 를 고려하여 (8)의 식을 변형하면 (9)와 같다.

$$rpm_{target} = \frac{v \mp \left(\frac{w}{2}\right)}{2\pi r} * 60 \quad (9)$$

$r$ 은 장착된 바퀴의 지름을 의미한다. 로봇의 구조적 특성에 따라 각각 왼쪽, 오른쪽 모터가 앞으로 나아가기 위해서는 서로 반대 방향으로 회전해야만 하므로 좌·우측 모터에 따른 목표 rpm은 +의 순서로 계산되었다.

### 3.4 엔코더 체배

엔코더 체배란, 2개의 엔코더 채널 A, B를 이용하여 A rising, A falling, B rising, B falling 등의 edge를 사용하여 각도를 측정하는 것을 의미한다. 대표적으로 1체배, 2체배, 4체배 방법이 존재하여 1체배 방식의 경우 분해능이 부족하여 잘 쓰이지 않는다는 단점이 존재한다. 4체배 방법의 경우 저성능 MCU에서 연산의 지연으로 인해 해당 논문에서는 4개의 edge중 2개를 이용하는 2체배 방식을 택하여 모터의 회전 각도에 따른 rpm을 구하였다.

$$rpm_{current} = \frac{60}{t * ppr * 2} * pulses \quad (8)$$

여기서  $t$ 는 하위 제어기가 모터에게 보내는 모터 명령 주기,  $ppr$ 은 1회전당 펄스 발생 개수,  $pulses$ 는 현재 엔코더에서 발생한 펄스의 개수이다.

### 3.5 PI 제어

PI 제어는 비례-적분 제어로서 Fig.8와 같이 구성하였다. 현재 스텝의 목표 rpm과 현재 엔코더를 통해 계산된 값인 현재 rpm의 error를 구하고 이에 따른 Gain 값은 다음과 같은 절차를 통해 구하였다.

1) 모든 Gain 값을 0으로 초기화

2) P Gain을 증가시켜 모터가 급하게 회전하면서 속도가 지속적으로 변화하는 경우 overshoot 현상이므로 해당 현상이 발생하지 않는 값을 선택

3) 정상상태에 도달하기 위해 잔류편차를 누적하기 위한 I Gain을 높여 목표 속도에 도달

최종적으로 비례 제어 항과 적분 제어 항을 합산하여 나온 결과에 대한 rpm을 pwm으로 변환하여 이를 액추에이터인 motor에 입력하였다.

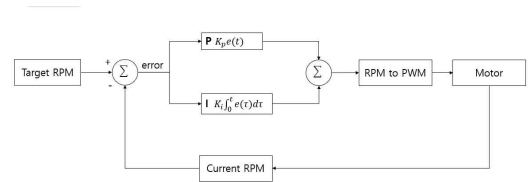


Fig. 8 PI Controller

## 4. 화재 거리 추정 및 화재 진압

최종적으로 화재 감지기가 설치된 지점에 도달하면 USB 카메라를 이용하여 화재를 인식한다. 인식된 화재 주위에 바운딩 박스를 그려 이의 넓이에 따른 거리를 실험적으로 측정하였을 때 Fig.9과 같은 결과를 얻을 수 있었다. 해당 데이터를 바탕으로 2차 다항식으로 가정하여 이에 대한 곡선 피팅을 수행하였다.

$$dist = w_0 * x^2 + w_1 * x + b \quad (9)$$

여기서  $x$ 는 바운딩 박스의 넓이를 의미한다.

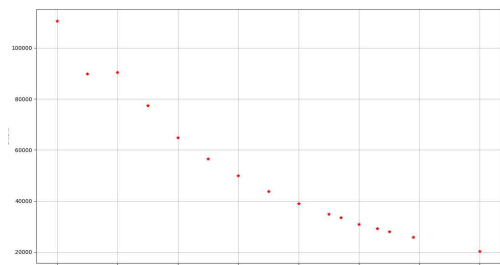


Fig. 9 Distance value according to the bounding box area



Fig. 10 Fire detection and distance estimation result

파라미터에 따라 화재까지의 거리를 추정하고 주위에 그려진 바운딩 박스의 중심점과 카메라 중심의 차이를 구하여 조향각을 결정하였다. 또한, 인식된 화재까지의 거리가 특정 임계값 이하면 투척용 소화기를 발사하게 하였다.

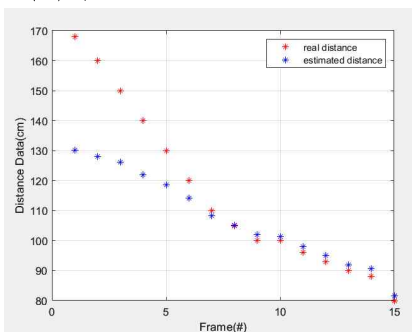


Fig. 11 Fire detection and distance estimation result

최종적으로 도출된 곡선 피팅 모델에 따라 15개의 프레임을 기준으로 실제 거리 데이터와 추정 데이터를 실제 측정결과의 오차를 측정하기 위해 rmse를 손실함수로 사용하여 계산한 결과 15.3632와 같이 측정되었다.

## 5. 결론

본 논문에서는 화재 감지기가 설치된 지점의 waypoint를 설정하여 해당 지점을 목적지로 설정하여 자율적으로 주행하고 투척용 소화기를 발사하는

로봇을 제작하였다. 하지만 사용된 라이다의 부족한 분해능과 엔코더의 노이즈, 타이어와 지면의 마찰에 의해 추정 위치에 대한 오차가 누적되는 현상이 발생하여 설정 목적지 도달에 60%의 성공률을 보였다. 따라서 이를 위해 IMU를 추가하여 엔코더와 라이다로 부족한 위치 추정능력을 보완하고 화재 추정을 위한 다량의 데이터를 확보하여 개선할 계획이다.

## Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, MSIT (No.2020R1F1A1071819). This work was supported by project for Industry-Academic Cooperation Based Platform R&D funded Korea Ministry of SMEs and Startups in 2022. In addition, this research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2018-01426) supervised by the IITP(Institute for Information & communications Technology Promotion).

## References

- 1) Shubham Nagla, "2D Hector SLAM of Indoor Mobile Robot using 2D Lidar", 2020 2nd International Conference on Power, Energy, Control and Transmission Systems, 2020
- 2) Hakjun Kim, Sumin Kang, Donghan Kim, "Development of Autonomous Driving Algorithm for Robots in High-rise Buildings Using Gazebo Simulator", Journal of Institute of Control, Robotics and Systems, 2022
- 3) Byeoung Min Ahn, Yunho Ko, Jihong Lee, "Indoor Location and Pose Estimation Algorithm using Artificial Attached Marker", Journal of Korean Multimedia Society, Vol. 19, No.2, 2016
- 4) Jinyeong Heo, Jeesoo Ha, Junsik Lee, Jaekwan Ryu, Yongjin Kwon, "Dynamic Widow Approach with path-following for Unmanned Surface Vehicle based on Reinforcement Learning", Journal of the KIMST, Vol. 24, No. 1, pp. 61-69, 2021