

OID센서로 주행하는 교육용 로봇의 주행 개선을 위한 연구

송현주*

A Study on the Improvement of Driving of Educational Robots with OID Sensors

Hyun-Joo Song*

*Adjunct professor, Department of Software Engineering, Korea Cyber University, Seoul, 03051 Korea

요 약

본 연구는 교육용 로봇 중 한 종류인 스마트로봇을 대상으로 기존의 OID 센서 환경을 이용하되 주행의 문제점을 프로그램으로 처리하는 것을 제안 하고자 한다. 주행 정보를 가지고 있는 지도환경, 위치인식, 경로계획, 장애물 회피 및 경로 재설정 등을 중심으로 주행 테스트 환경을 구축했으며 평균적인 최종 오류 률이 아니라 오류가 증가하는 시점을 재보정의 순간으로 잡아야 한다는 가정을 통해 실험을 진행했다. 그 과정을 통해 선행실험에 비해 안정적인 주행 결과를 얻었다. 본 연구가 현재 판매되는 저가형 센서를 장착한 교육용 로봇들의 주행 성능을 올릴 수 있는 개발 방법이 될 것이라 본다.

ABSTRACT

In this research, we will use the existing OID sensor environment for smart robots, which are a type of educational robot, but we would like to propose that the problem of running be handled by a program. Maybe you have driving information We are building a driving test environment focusing on environment, position recognition, route planning, obstacle avoidance and path reset, and it is not the average final error rate, but the time when the error increases The experiment was conducted by a household that catches the moment of recalibration. Through the process, stable running results were obtained compared to the previous experiment. In this research, I think that it will be a development method that can improve the running performance of educational robots equipped with low-cost sensors currently on the market.

키워드 : 스마트로봇, OID센서, 로봇주행, 교육용로봇

Keywords : Smart robot, OID sensor, Robot driving, Educational robot

Received 24 February 2021, Revised 25 February 2021, Accepted 9 March 2021

* Corresponding Author Hyun-Joo Song(E-mail:dimfe11@gmail.com, Tel:+82-2-6361-2000)

Adjunct professor, Department of Software Engineering, Korea cyber University, Seoul, 03051 Korea

Open Access <http://doi.org/10.6109/jkiice.2021.25.4.549>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

2017년에 정규 교육과정에서 소프트웨어 교육이 등장하면서 다양한 코딩 방법들이 개발되고 있다. 이 중 교육용 로봇은 IoT(Internet of Things) 기반의 음성, 센서의 다양한 하드웨어를 사용하면서 학습자에게 맞춤형 서비스를 제공할 수 있다는 장점이 있어 많이 사용되고 있다[1].

교육용 로봇은 사용자가 로봇을 움직이고 그 로봇으로 코딩의 결과를 표현하기 때문에 원활한 주행은 교육에 중요 고려요소이다. 다만 교육용 로봇이 일종의 교육 교보재로 사용되고 있기에 기기의 성능이나 센서 장치가 제한적일 수밖에 없다. 그러다 보니 현재 시장에 있는 교육용 로봇들은 교육에 연계되기에 어려울 만큼 주행이나 이동에서 부정확한 움직임을 보여주거나 수동적 주행을 처리하는 경우도 많다[2].

본 연구는 현재 교육용 로봇에서 많이 사용되는 스마트 로봇[3,4]을 대상으로, 기존의 센서환경을 이용하되 주행의 문제점을 프로그램으로 처리하는 것을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 교육용 로봇 중 하나인 스마트 로봇과 교육용 로봇의 주행 기술 정보, 닷코드에 관한 관련 연구를 다룬다. 3장에서는 주행테스트 환경 구성을 서술하고 선행 주행을 통해 주행시의 문제점을 분석했다. 4장에서는 이동속도와 위치인식이 가능한 환경에서 주행 오류를 줄일 수 있는 보정방법을 제안하고 5장에서는 4장에서 제안한 방법에 대한 실험결과를 정리한다.

II. 관련 연구 및 제품 분석

2.1. 스마트 로봇

교육용 로봇은 교육지원을 위한 다양한 역할을 수행하는 로봇으로 정의한다[5]. 형태에 따라 두 가지로 분류하는데 키트형 로봇과 완제품 로봇이다[2]. 스마트 로봇은 완제품 로봇 내에 속한다.

스마트로봇은 완제품 로봇의 단점이었던 경제적인 면과 개발의 어려움을 스마트폰이나 태블릿PC를 이용해 해결한 형태이다[6]. 완제품 로봇이 본체 안에 운영체제, 소프트웨어 등이 탑재해 업그레이드나 콘텐츠개

발이 어렵다는 점을 스마트폰을 이용해 해결한 것이다. 대표적으로 그림 1의 알버트[7]를 들 수 있다.



Fig. 1 AlbertBT[7]

2.2. 교육용 로봇의 주행 기술 및 정보

일반적인 로봇의 주행을 위해 필요한 기술은 지도환경 확인, 위치인식, 경로계획, 장애물 회피 및 경로 재설정 등을 볼 수 있다[8]. 교육용 로봇도 큰 차이는 없지만 상대적으로 일반 로봇의 센서나 기기 성능을 갖고 있지는 못하기 때문에 수동적 주행을 처리하거나 부정확한 움직임을 보여주는 원인이 되고 있다.

2.2.1. 지도환경 확인

로봇의 지도환경은 격자지도, 위상지도, 특징지도, 하이브리드 지도가 있다[8].

교육용 로봇에서는 환경을 셀로 표현하는 격자지도를 많이 사용하는데, 위치인식 시 소수점을 처리하지 않고 임의로 격자 개수만 확인하므로 데이터양이 상대적으로 적고 처리가 빠르다. 그림 2는 격자형 콘텐츠의 예인 카미봇의 낚시게임이다.



Fig. 2 Kamibot. Fishing game[9]

2.2.2. 위치인식

위치인식은 로봇의 처음 위치를 기준으로 현재 위치를 추정하는 지역 위치인식, 로봇의 초기 위치를 모르는 경우에 지도 작성으로 절대 위치를 추정하는 전역 위치인식이 있다.

일반적인 로봇의 위치인식을 위한 센서는 비전이나 초음파 혹은 RF를 사용[10]하지만 교육용 로봇의 경우 IR(적외선, Infrared Ray)센서나 OID(Optical Identification) 센서를 1~2개 정도 사용하는 수준이다. 표 1은 기존 연구[11]를 기반으로 스마트 로봇의 입력과 출력을 분석해 놓은 표이다.

Table. 1 Types and characteristics of complete Educational robots

| Name | Input/Output |
|----------|--|
| Nuratti | Input : Touch sensor, IR sensor, OID sensor Output : Microphone, LED, Wheel. |
| Dio | Input : Ultrasonic sensors, IR sensors Output : Wheel, Buzzer, LED, Sub-motors. |
| truetrue | Input : IR sensor, light sensor, Acceleration sensor, Color sensor Output : Wheel, Buzzer, LED |
| Hamster | Input : IR sensor, Acceleration sensor, Brightness sensor, Temperature sensor Output : Wheel, LED, buzzer |
| Altino | Input : IR sensor, Temperature sensor, Acceleration sensor, Compass. Output : Wheel, dots, matrices, LED |
| Epro | Input : Infrared sensor, Touch sensor, Brightness sensor, Camera Output: Wheel, speaker, LCD, LED |
| Turtle | Input : Color sensor, Acceleration sensor Output : Wheel, LED, speaker. |
| Albert | Input : OID sensor, IR sensor, Acceleration sensor, Brightness sensor, Temperature sensor Output : Wheel, LED, Speaker, buzzer. |
| Genibot | Input : OID sensor, Acceleration sensor, Floor detection(IR) sensor, Brightness sensor Output : Wheel, LED |

IR센서는 적외선으로 바닥의 검은 색을 측정해서 이동 가능여부를 알아보는 방식이다. 그래서 로봇의 위치 정보는 프로그램적으로 가상 추정한다. 실제 로봇의 위치정보를 확인할 수 없으므로 로봇이 주行的 문제를 일으켜도 오류를 감지할 수 없다.

이에 비해 OID센서의 경우 닷코드를 읽는 방식인데, 움직이면서 실시간으로 바닥판의 위치 정보를 읽어 들이기 때문에 좀 더 진화된 형태라고 볼 수 있다. 이 방법은 로봇의 위치 파악이 용이하고, 주행 시 문제점이 생기면 즉시 알 수 있다는 장점이 있다.

또 공통적으로 두 방식 모두 위치인식 여부를 센서 1~2개에 의존하는 방식이라 이동속도가 높아지면 센서

인식률의 저하를 예상할 수 있다.

2.2.3. 경로계획

경로계획은 현재 로봇의 위치로부터 목표점까지의 최적의 경로를 찾아내는 기술이다. 사용자의 수동적 이동만 처리하는 교육용 로봇의 경우 현재 위치를 알 수 없으니 경로계획도 세울 수 없다. 그러니 경로계획은 센서를 이용해 위치인식이 가능하거나 프로그램적으로 처리해 추정할 수 있는 로봇을 대상으로 한다.

로봇의 경로계획은 로봇의 현 위치에서 목표점까지 이동하는 하는 PTP방식, 전체영역을 경로로 설정해서 이동하는 전체영역 경로방식[12]을 사용한다. 대부분의 스마트로봇이 PTP방식을 사용한다. 중간 경우를 하는 경우 현 위치, 중간위치, 목표점까지를 처리한다.

2.2.4. 장애물 회피 및 경로 재설정

이동 중인 로봇이 오류 없이 주행이 가능하도록 하는 기술로 장애물 확인은 전 후방에 설치된 근접 IR 센서를 사용한다.

회피나 경로 재설정의 경우 추가 프로그램을 통해 진행하는데, 위치인식을 사용하지 않는 로봇의 경우 센서 확인 후 후진이나 경고음을 내보내며 정지한다. 위치인식과 경로계획이 가능한 경우, 현재 위치를 추정한 후 PTP방식으로 목표점까지 재 경로를 처리한다.

2.3. 닷코드와 닷코드를 이용한 위치인식

교육용 로봇에서 가장 사용빈도가 높은 것은 2.2.2에서 거론한 IR센서 방식이다. 하지만 이 방식은 이동 가능 지역에 검은색을 사용하기 때문에 콘텐츠를 만들 때 디자인의 제약이 생긴다. 그리고 기능적으로도 이동 가능 여부만 확인할 뿐 경로처리나 위치인식, 경로 재설정 구현에는 어려움이 있다.

OID센서는 IR센서의 단점을 개선할 수 있어서 근래 출시되는 스마트로봇에서 적용되는 방식이다. 이 방식은 지도환경에 닷코드(Dot Code)를 인식해서 위치정보를 바로 확인할 수 있다.

닷코드는 눈에 보이지 않는 점으로 구성된 코드를 생성, 인쇄, 활용하는 기술이다[13]. 적외선을 반사하지 않는 탄소 잉크로 닷코드를 인쇄 한 후 OID센서로 패턴에 숨겨진 정보를 읽는다. 눈에 보이지 않는 미세한 점으로 인쇄되므로 지도 정보가 콘텐츠를 방해하지 않는다. 그림 3은 닷코드의 일부를 확대한 것이다. 이처럼 컨텐츠

상태의 인쇄물일 때는 닷코드가 잘 보이지 않으므로 디자인에 방해가 되지 않는다.

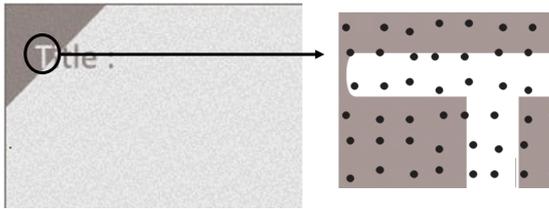


Fig. 3 Dot Code

OID에 인식된 닷코드의 최종 인식률이 80% 이하가 되면 오류로 판단한다[14]. 80% 이하라고 해도 실제로 센서에는 닷코드 10~100여개가 동시 입력된다. 그럼 최종적인 인식률은 계산식 (1)과 같다.

$$P = 100\% - (L)^C \quad (1)$$

P는 최종적인 인식률이다. 모든 닷코드가 인식된다면 100%이며 실패한 닷코드 인식률은 L로. 동시 입력이 되는 닷코드의 수는 C로 표현했다.

(1)의 내용을 기반으로 닷코드 인식률이 80%. 닷코드 10개가 동시 입력된다고 하면 최종 인식률 L은 99.999%, 즉 100%에 가까운 인식이 된다. 닷코드 방식은 이처럼 인식률이 워낙 뛰어나기 때문에 아동용, 교육용 콘텐츠에서도 많이 사용된다.

다만 (1)의 인식률은 정지상태일 때의 인식률이다. OID 역시 센서이기 때문에 움직임이 생기면 빨라지만 센서 인식률이 떨어질 것을 예상할 수 있다. 특히 주행같이 지속적이며 빠른 이동은 센서 인식의 저하를 가져온다.

III. 주행 테스트 환경 개요

스마트 로봇 대부분은 IR과 OID센서를 이용한 위치 확인을 한다. 본 연구에서는 이런 센서 환경을 그대로 사용하면서 주행의 문제점을 프로그램적으로 처리하고자 한다. 그러기 위해서 먼저 주행 테스트를 위한 환경을 구축했다.

다만 IR방식의 경우 이동 여부만 확인하는 방식이라 실제 주행 데이터를 받아서 확인하는 것에 어려움이 있다. 그래서 실험 대상은 실시간 위치확인 가능한 OID 센서 기기로 진행한다.

로봇의 주행 시스템은 주행 프로그래밍 후 업데이트가 가능한 스마트로봇, 주행을 위한 기술 정보 4가지로 나뉜다.

3.1. 주행 대상 스마트로봇

이번 실험에 사용할 스마트로봇은 O사의 알버트 BT[9]이다. 11cm 정도의 크기에 무게 250g의 소형 로봇으로, 3축 가속도 센서, 전방 좌우측 구분 근접 센서, 하방 전 후 측 닷코드를 읽을 수 있는 OID센서 두 개, 조도 센서와 온도센서를 장착했다. OID를 통한 위치 정밀도는 2.5mm이며 각도 정밀도는 2.5도이다.

3.2. 지도 환경

알버트의 일반적인 콘텐츠는 다른 교육용 로봇들처럼 격자지도 형태로 되어 있지만 지도 환경은 그림 2처럼 닷코드(Dot Code)가 인쇄된 형태이다.

실험환경의 지도는 해당 기기용으로 시판되어 사용되는 표준 네비게이션 패드의 사이즈를 사용했다. 사이즈는 가로 108cm, 세로 76cm이지만 최종적으로 COMMAND_PAD_SIZE에 데이터 값으로 입력되므로 사이즈 변경도 가능하다. 그림 4는 지도환경인 네비게이션 패드의 일부 이미지이다.

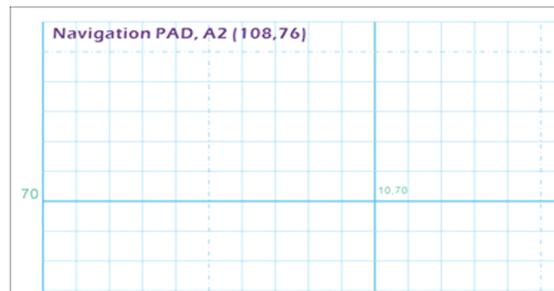


Fig. 4 Navigation PAD[15]

주행 테스트를 위해서 긴 코스가 필요했기 때문에 표준 네비게이션 패드를 4조각으로 나눠 길게 붙여서 이동전용 네비게이션 패드를 다시 제작했다. 가로 25cm, 세로 300cm 이며 이것은 다시 데이터로 전환해서 COMMAND_PAD_SIZE에 300, 25로 입력되었다.

3.3. 위치인식

실험기기의 위치인식은 닷코드와 하단의 OID센서를

통해서 이루어진다. 그림 5는 기기 하단에 있는 두 개의 OID센서이다. 다만 주행 시스템에서의 위치 인식은 정지상태의 위치 인식과 다르다.



Fig. 5 Dot Code Sensor : OID Sensor[7]

정지되어 있는 상태라면 (1)의 계산식처럼 100%에 가까운 닷코드 인식을 보이기 때문에 현 위치의 정확한 인식이 가능하다.

하지만 주행을 시작하면, 센서의 특징 상 인식률이 떨어질 수밖에 없다. 결국 이동이라는 요소는 인식률이 낮아지는 주 원인이 된다. 개발 경험 상 이동속도가 높아지면 인식률이 떨어지고, 이동속도가 낮아지면 인식률은 높아진다[16].

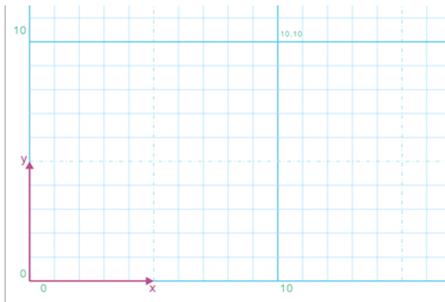


Fig. 6 Navigation Location Data[15]

인식이 이루어지면 해당 닷코드를 통해 입력된 위치 데이터는 SENSOR_POSITION에 x, y에 해당하는 2개의 배열로 들어온다. 각각 -1부터 39999까지의 값을 가질 수 있는데 좌표 값은 그림 6처럼 0부터 시작하기 때문에 0부터 39999까지는 유효한 값이고, -1은 현재 위치를 알지 못할 때 표시하는 값이다.

즉 -1의 값이 나타나는 경우가 계산식 (1)의 P가 80% 이하, 인식이 안 되는 순간으로 볼 수 있다.

3.4. 경로계획

경로계획은 PTP방식으로 목표점과 현재 스마트로봇의 위치를 이용한다. 현 위치는 닷코드를 이용해서 읽을 수 있으므로 중간 경로점과 최종점의 데이터만 배열값으로 COMMAND_WAYPOINTS에 입력한다.

3.5. 장애물 회피 및 경로 재설정

테스트 기기는 전방 좌우측에 IR센서가 있어 장애물을 인지할 수 있다. 본 연구의 목적인 센서를 이용한 주행에서의 문제점 해결을 위해서는 경로를 벗어났을 때의 현상을 이해해야 한다.



Fig. 7 Out-of-Road Robot

그림 7은 이동 시 로봇이 길을 벗어나는 상황을 가정한 이미지이다. 올바른 길로 이동하다가 인식의 오류가 발생하면 점차 벗어난 길로 이동하는데, ②에서 ③상태까지는 인식률이 기하급수적으로 크게 떨어진다. 최종 인식률 L이 80%가 되는 상황이 ③인데, 이미 그 로봇은 길을 벗어난 상황이 된다.

이 현상을 줄이기 위해서는 평균 인식률이 아니라, 위치데이터가 -1. 즉 오류를 나타내는 위치인식을 늘어나는 시점을 확인하는 것이 경로를 벗어날 여지를 미리 알아보는 지표가 될 수 있다. 직선으로 주행하고 있지만 -1 위치데이터가 늘어난다는 것은 현재 위치데이터 인식에 문제가 있다는 뜻이기 때문이다.

이 경우 최종점의 위치는 변함이 없으므로 현 로봇의 위치를 중간점으로 잡아서 경로계획을 세운다.

3.6. 주행 방식

스마트로봇의 주행형태는 아래 그림 8처럼 네 가지 주행 형태를 갖는다[2,15]. A는 전진-후진, B는 라운드

이동, C는 피봇이동, D는 제자리돌기이다.

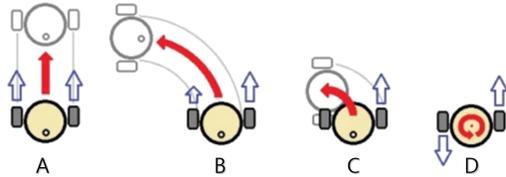


Fig. 8 4 Different Types of Robot Movements[2]

주행 방식의 제어는 바퀴의 속도 데이터를 이용해서 진행하며 A의 경우 모터의 회전방향을 변경해 후진처리를 할 수 있다.

주행 시 생기는 현상들의 발생 요인 및 검증을 위해 바퀴의 속도 데이터를 기준으로 속도를 올리는 것으로 실험을 진행했으며 이동시 OID센서의 인식률을 확인하기 위해 이동 시 좌표 데이터를 기록하도록 추가해서 위치데이터 값이 -1을 표시하는 것을 확인하도록 했다.

IV. 주행 오류 보정 방법 설계

교육용 로봇의 주행 특징을 확인하기 위해 3장의 내용을 근거로 위치정보와 이동속도를 제어 할 수 있는 환경을 구축했다.

선행 실험을 통해 주행환경의 도출되는 문제점을 관찰하고 이동 시 움직임 오류 탐지 및 보정 방법을 제안하고자 한다. 전체 도식을 표현하면 그림 9와 같다.

4.1. 선행실험, 센서를 이용한 주행 테스트

주행 테스트는 3.2에서 서술한 주행테스트용으로 만든 네비게이션 패드에서 진행했다. 각 속도의 구현은 좌우 바퀴의 속도에 해당하는, 왼쪽 바퀴는 EFFECTOR_LEFT_WHEEL에, 오른쪽 바퀴는 EFFECTOR_RIGHT_WHEEL에 각각의 값을 넣어 구현한다. 스마트 로봇의 경우 회전을 각 바퀴의 속도 데이터로 조정하기 때문에 왼쪽과 오른쪽 바퀴 수는 보통 이동이 가능한 로봇이라면 따로 조정이 가능하도록 만들어져 있다.

개발사에서 제공하는 개발가이드의 표준 속도 20[14]을 중심으로 10, 20, 30, 40으로 조정해서 진행했으며 SENSOR_POSITION에서 들어오는 위치데이터 x, y 값을 별도 배열로 저장한다.

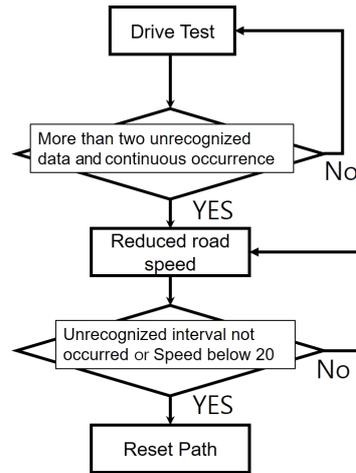


Fig. 9 Driving modification Method

각각 속도에 따라 50회씩, 총 200회를 실시했다. 위치데이터가 -1로 기록된 경우, 인식률의 오류가 생긴 경우의 평균값은 표 2이다.

Table. 2 Perception error depending on robot speed

| Wheel Speed Value | Per-drive recognition error average value |
|-------------------|---|
| 10 | 0.01% |
| 20 | 0.3% |
| 30 | 3.1% |
| 40 | 10.5% |

위치데이터 오류, 즉 인식률의 오류 평균값을 확인하면 속도 40일 때도 10.5%. 즉 오류 기준인 평균 20%에 미치지 못하는 공식 테스트 기준의 정상 범위이다[14]. 하지만 실제로는 속도 40에서 인식오류가 발생하면 평균값과 관계없이 로봇은 완전히 구간을 이탈해 버린다. 그 이유는 대부분의 인식오류가 한 번에 몰아서 발생하고 속도가 빠를수록 잘못된 코스로의 진입이 빠르기 때문이다.

4.2. 위치인식 오류가 증가하는 상황의 확인

인식 오류의 평균값은 주행에서 문제점이 있는지의 여부는 짐작할 수 있으나, 해당 오류를 대처할 수는 없다. 3.5에서 서술했듯 오류의 숫자가 중요한 것이 아니라 위치인식 오류가 연속해서 일어나는 시작 시점이 중요하기 때문이다. 표 3은 속도 30을 기준으로 50회 테스트

트 중 3번째 테스트의 일부 위치데이터 값이다. 중심을 벗어나지 않고 주행한다면 x 값은 네비게이션 패드의 중심 값인 13이어야 하며 y 값만 증가하면 된다.

Table. 3 The 3 of the 50 tests. Some location data values (based on speed 30)

| | | | | | | | | | | | | | | | | |
|---------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Index | ... | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | ... |
| x value | ... | 13 | 13 | 12 | -1 | 13 | 13 | -1 | -1 | 14 | 14 | 15 | -1 | -1 | 16 | ... |
| y value | ... | 56 | 59 | 61 | -1 | 64 | 67 | -1 | -1 | 73 | 76 | 79 | -1 | -1 | 82 | ... |

x값의 ±1~2는 인식 후 좌표 변경 시 생기는 노이즈로 판단할 수 있다. y값은 직진 주행을 하고 있는 상황으로 증가한다. 표 3에서 인식 불가 구간은 73, 76, 77과 81, 82인데, 이 중 단일 인식불가 구간 73이다. 74를 보면 좌표가 회복이 되었다. 하지만 인식불가 구간이 두 개 이상 연속인 경우 그 다음 좌표가 크게 증가한 것을 볼 수 있다. 83의 경우 x가 기준점 13에서 16까지 증가했다. 이것은 로봇이 오른쪽으로 치우치고 있는 것을 의미한다. 속도가 빠르면 이런 인식불가 구간 후 좌표가 더 크게 치우치고 그것은 다시 인식불가 상태를 야기한다.

이런 상황에서 주행 상태를 바로잡기 위해서는, 인식불가 구간이 두 개 이상 그리고 연속으로 나타날 때를 재보정의 순간으로 기준을 잡아야 한다는 가정을 얻었다.

4.3. 재보정 시점의 확인 및 보정방법

스마트 로봇의 주행 개선 효과를 얻기 위해서 가장 효과적인 것은 이 위치인식 오류의 직접적인 원인이 되는 속도를 줄이는 것이다. 그것만으로도 위치 인식의 문제는 크게 줄어든다. 그 다음 현재 위치데이터를 읽을 수 있는 상태가 되면 그것을 기반으로 경로를 재설정한다.

4.3.1. 속도 감소

안정 속도 기준은 개발 사이트의 표준 개발 속도인 20[13]을 기준으로 했다. 실제 표 2에서도 속도가 10, 20일 때는 인식률의 문제가 미비하다.

본 실험에서는 재보정의 기준이 되는 인식불가 구간이 두 개 이상, 연속으로 나타나는 경우를 필터링했고 프로그램 상에 인지되면 계산식 (2)를 기반으로 속도를 줄였다.

$$V_{moter} = V_{real} * K \quad (2)$$

(2)는 재보정을 위한 감속 모터 속도 V_{moter} 를 구하는 식이다. 현재 속도 V_{real} 에 감쇠계수 K 를 곱해서 조건이 될 때까지 반복한다. 본 실험에서는 K 를 0.95로 계산했으며, 이 부분은 기기별로 조정이 필요하다.

V_{moter} 가 20이 되거나 혹은 두 개 이상의 연속되는 인식 불가 구간이 나타나지 않는다면 속도 조정이 끝났다고 가정하고 경로 재설정을 진행했다.

4.3.2. 경로 재설정

로봇의 경우 보통은 측면부에 두개의 바퀴를 가진다. 그러기 때문에 목표 경로를 입력했다고 해도 즉각적으로 보정 값으로의 이동을 기대할 수는 없다.

로봇은 이미 주행 중에 있기 때문에 y값은 속도에 맞춰 증가하므로 보정이 필요한 값은 위치데이터 중 x값이 된다. 3.4에서 서술한대로 중간 경로 값과 최종점의 데이터를 배열 값으로 COMMAND_WAYPOINTS에 입력하면 PTP로 경로계획을 잡는다. 이 중 최종점은 그대로 두고 중간 경로 값을 사용한다.

직선에서는 처음 출발점의 위치데이터의 x값이 계속 유지되어야 한다. 이 값이 그대로 중간 경로 값으로 사용되면 된다.

$$\begin{aligned} x &: \text{출발점의 위치데이터의 } x\text{값} \\ y &: \text{현재 위치데이터의 } + \text{ 추가 값} \end{aligned} \quad (3)$$

계산식 (3)은 직선구간에서 중간 경로값을 계산하는 식이다. x값은 출발 데이터의 x값을 그대로 사용하며 y값의 경우 현재 데이터에서 좀 더 증가한 값을 쓴다. 그래야 해당 위치에 도달했을 때 중간 경로로 인지한다. 실험에서는 +15를 사용했다.

V. 실험 결과

주행테스트를 통해 속도가 빨라지만 센서의 위치 인식률이 나빠진다는 것을 알게 되었으며 이 상태에서 주행 상태를 바로잡기 위해서는, 인식불가 구간이 두 개 이상 그리고 연속으로 나타날 때를 재보정의 순간으로 기준을 잡아야 한다는 가정을 얻었다.

해당 가정을 기반으로 재보정을 해야 할 상황을 체크하고 해당 조건이 발생하면 일차로는 속도를 감소시키고 고 인식률을 높인 다음 현재 위치데이터를 기반으로 중

간 경로값을 계산하는 설계 제안을 했다. 이 진행조건은 그림 9에 도식으로 표현했다.

표 4는 이 방법을 적용한 실험의 결과로 앞서 선행실험을 했던 표 2의 실험조건으로 진행했다. 각각 속도에 따라 50회씩, 총 200회를 실시했으며 위치데이터가 -1로 기록된 경우, 즉 인식률의 오류가 생긴 경우의 평균을 기록했다.

Table. 4 Recognition error rate after error correction depending on robot speed

| Wheel Speed Value | Per-drive recognition error average value |
|-------------------|---|
| 10 | 0.01% |
| 20 | 0.2% |
| 30 | 0.4% |
| 40 | 0.6% |

30에서는 3.1%였던 인식률의 오류가 0.4%대로 떨어졌으며 40에서는 10.5%였던 오류가 0.6%로 떨어졌다. 이것은 일반적인 테스트 기준인 최종 인식률 20%에는 미치지 않으며[14] 실제 실험에서도 경로를 이탈하는 오작동의 상황은 나타나지 않아서 효과적으로 사용되고 있다고 관찰할 수 있었다.

VI. 결론

본 연구는 교육용 로봇 중 한 종류인 스마트로봇을 대상으로 추가적인 센서나 기기를 사용하지 않고 원활한 주행이 가능하도록 해당 문제점을 프로그램적으로 처리하는 것을 제안하고자 한다.

그러기 위해서 우선은 기존의 교육용 로봇과 스마트 로봇 기기들의 분석과 관련 연구를 진행했고 특히 주행 정보를 가지고 있는 지도환경, 위치인식, 경로계획, 장애물 회피 및 경로 재설정을 중심으로 교육용 로봇의 현재 기술들을 확인해 보았다. 그리고 주행 테스트 환경을 구축해서 위치 인식의 오류율을 확인하며 속도가 높아지면 오류가 늘어난다는 것을 확인했다. 늘어난 오류를 줄이기 위해 위치 인식 오류가 증가하는 상황을 확인하는 지표를 만들고 그 순간에 속도가 감소하고 그 다음 위치 인식을 다시 해서 현 위치를 중간값으로 계산해 재보정을 하는 과정을 진행했다. 이 과정을 통해 선행실험

에 비해 안정적인 주행 결과를 얻었다.

센서의 경우 속도가 오르면 오류가 증가할 것이라는 상황은 어느 정도 예상은 되지만 실제로 적용하고 문제점을 분석해 본 것에 의미가 있는 실험이라고 보겠다. 다만 속도를 줄이는 과정에서 좀 더 효과적인 감쇠계수의 계산이 추가적으로 필요하다. 그리고 본 실험에서 제외했던 IR센서를 기반으로 하는 가상의 위치 인식을 진행해서 현 실험의 과정을 테스트 해 보는 것도 추가적으로 필요한 연구라고 생각한다.

위에 언급한 추가적인 연구를 진행한다면 현재 판매되는 저가형 센서를 장착한 교육용 로봇들의 주행 성능을 올릴 수 있는 개발 방법이 될 것이라 본다.

References

- [1] K. C. Kim and K. E. Lee, "The Effects of Activities with Teaching Assistant Robots on Children's Social Competence," *Korean Journal of Early Childhood Education Research*, vol. 15, no. 2, pp. 29-49, 2013.
- [2] K. B. Park, S. H. Cho, and B. J. Seo, "An Observation-based Movement Control for Educational Coding Robots," *Journal of Korea Game Society*, vol. 12, no. 6, pp. 131-142, 2016.
- [3] D. J. Kim, "Analyses of the Region and Gender Differences in Creativity, Problem Solving and Flow of Robot," Ph. D. dissertation, Konkuk University of Educational Technology, 2016.
- [4] J. S. Lee and D. W. Rhee, "The development of Smart Robot Game," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 12, pp. 1596-1601, 2019.
- [5] Korea Technology and Information Promotion Agency for SMEs, "Technology Roadmap For SMEs 2016-2018 - Robot Application," 2016.
- [6] W. R. Lee, "The Effects of the Smart Robot Albert BT-Based Coding Education on the Communication Competence and Creativity of Five-Year-Olds," Masters. dissertation, Chongshin University, 2017.
- [7] Albert Robot Website [Internet]. Available : <http://www.albert.school>.
- [8] J. B. Song, "Trends in the Research and Development of Driving Technologies for Mobile Robots," *Korea Robotics Society, Korea Robotics Society Review*, vol. 2, no. 2, pp. 40-45, 2005.

- [9] Kamibot Website
[Internet]. Available : <http://www.kamibot.com>.
- [10] J. C. Jin, "An Introduction to the Development Trends of Location Recognition System," *Korean Institute of Communication Sciences*, vol. 25, no. 4, pp. 5-10, 2008.
- [11] Korea Institute of Education, Academic Information, Ministry of Trade, Industry and Energy, Korea Robot Industry Promotion Agency "Job training by SW education level using robots," Korea Institute of Education, 2017.
- [12] S. H. Nam and S. B. Moon, "Minimal Turning Path Planning for Cleaning Robots Employing Flow Networks," *Journal of Control, Automation and Systems Engineering*, vol. 11, no. 9, pp. 789-794, 2005.
- [13] Ministry of Culture, Sports and Tourism, "The Development of Technology of Scanning a Dot Code Using Smartphone in order to learn and Teach Using Educational Contents," Content Industry Technical Support Project Report, 2013.
- [14] Genie Kids and Korea Creative Content Agency, "Content Industry Technology Support Project, Robots Contents dependency on Smart Decive," Korea Creative Content Agency, 2011.
- [15] Smartbot Website [Internet].
Available : <http://www.smartrobot.org>.
- [16] Androidpup in Smart Robot Development Information noticeboard [Internet]. Available : https://www.androidpub.com/android_dev_robot.



송현주(Hyun-Joo Song)

2005 상명대학교 게임학 석사
2015 상명대학교 게임학 박사
2017~현재 고려사이버대학교 소프트웨어공학과 겸임교수
※관심분야 : 게임개발, 스마트로봇, 코딩교육