

## 코딩 학습용 로봇 제어를 위한 관찰 기반 움직임 제어

박경복, 조성현, 서범주  
홍익대학교 일반대학원 게임학부  
testadmara@gmail.com, {scho, bseo}@hongik.ac.kr

An Observation-based Movement Control for Educational Coding Robots

Kyeongbok Park, Sung Hyun Cho, and Beomjoo Seo  
School of Games, Hongik University

### 요약

소프트웨어 교육이 초중등 정규 교과목으로 편제됨에 따라 코딩 교육용 로봇을 활용한 코딩 교구 시장이 나타나고 있다. 하지만 현재 로봇용 코딩 교구 시장은 로봇의 부정확한 움직임 제어 때문에 코딩교육의 본질을 벗어나 장난감으로서의 기능에만 충실한 경우가 많다. 이 점에 착안하여 본 연구에서는 6개의 라인센서의 관찰 정보를 바탕으로 로봇의 전진 이동, 회전 이동을 보정하는 방안을 제시한다. 본 방안의 효용성을 검증하기 위해 보드 게임과 유사한 코딩 학습용 테스트베드를 구축하고 코딩을 통해 로봇을 제어할 수 있는 모바일 앱을 활용하여 로봇의 실제 움직임 정확도를 측정하였다. 측정결과에 따르면 다양한 명령어를 수행한 결과 거리측면에서는 최대 1.57cm, 각도측면에서는 2.38도의 오차를 보고하고 있다. 이 결과를 바탕으로 본 논문에서 제안한 방식을 활용하면 향후 코딩교구 시장에서의 로봇의 활용도가 더욱 커질 것으로 기대된다.

### ABSTRACT

Recently, a variety of methodologies based on the use of educational coding robots have been proposed and experimented in newly emerging Korean educational coding markets. Due to imprecise movement accuracy of the existing coding robots, however, such robots are mainly used as a toy. Inspired by these observations, we developed a simple adjustment method during robot movements, which is based on the observation results of 6 line-tracking sensors: 3 sensors in the center of a robot and other 3 sensors at the front. These sensors help to determine whether the robot is in the right location and whether it rotates as expected. Through the well-designed experiments, we report the effectiveness of our proposed solution: the average distance error of 1.57cm and the average degree error of 2.38 degree before and after complex movements. In near future, we hope our method to be popularly used in various educational coding robot platforms.

**Keyword** : Serious Educational Game(교육 게임), Coding Puzzle Platform for Kids(아동용 코딩 플랫폼), Arduino(아두이노), Robot Car(로봇 차), Line Sensors(라인센서), Observation-based Movement Control(관찰 기반 움직임 제어)

Received: Nov. 16, 2016 Revised: Dec. 16, 2016  
Accepted : Dec, 15, 2016  
Corresponding Author: Sung Hyun Cho(Hongik University)  
Email: scho@hongik.ac.kr

© The Korea Game Society. All rights reserved. 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.

ISSN: 1598-4540 / eISSN: 2287-8211

## 1. 서 론

2017년 초중등 교육과정에서 소프트웨어 교육이 정규 교과목으로 편제됨에 따라 다양한 코딩방법론이 대두되고 있다[1]. 지식기반의 현대 사회에서는 고차원적인 컴퓨터적 사고력 배양이 필요하다. 이를 위해 초등학교 저학년부터 코딩에 대한 흥미를 갖고 학습활동을 할 수 있도록 쉽게 접근 가능한 코딩교구 개발에 중점을 두고 있으며 널리 사용되는 코딩학습 방법으로 로봇을 활용한 코딩 교육이 있다[2,3,4,5].

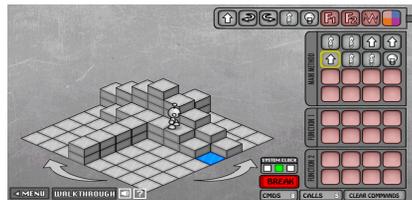
현재 널리 사용되는 코딩교구들은 다양한 IoT(Internet of Things)기기들과 연동되어 코딩 교육에 활용되고 있다[6,7]. 이것들을 분류해 보면, 학생들이 직접 센서를 회로에 연결하고 센서를 구동해보도록 테스트하는 기판형 기기 제어 방식과 외형을 갖춘 로봇에게 필요한 명령을 프로그래밍하여 제어하는 로봇 구동형 제어 방식(예를 들어, Bee[8], 알버트 로봇[9])으로 나눌 수 있다. 비트블릭[10]으로 대변되는 기판형 기기 제어 방식에서는 다양한 센서들을 직접 제어할 수 있다는 점에서 흥미를 유발할 수 있으나 하드웨어적인 센서 제어에 국한되어 있으며 사전 기반 프로그래밍과 같이 고급 코딩 기법을 저연령대 학생에게 학습시키는 것이 부담스럽다. 이에 반해 로봇 구동형 제어 방식은 직접 로봇의 움직임을 제어한다는 측면에서 초기 흥미를 지속적으로 유지시킬 수 있는 반면 일정 수준 이상의 움직임을 위해서는 고도의 추상적 사고력을 요구하는 경우가 많다. 또한 섬세한 프로그래밍 기반 움직임 제어가 어렵다. 이러한 연유로 초기에 개발된 코딩교육용 로봇들은 극히 제한적인 코딩교육에 활용되고 있으며 수동으로 움직임을 제어하는데 국한된 경우가 많다.

현재 시장에 선보인 로봇 기반 코딩교구들은 코딩교육에 바로 연계되기에 어려울 만큼 전진 이동, 회전 이동 등에서 부정확한 움직임을 보여주고 있다. 예를 들어, 미로에서 장애물을 만났을 때 어떤 방식으로 회피할지를 간단하게 알고리즘적으로 설

계하고 미로를 탈출하는 예제에 로봇 제어를 활용한 경우가 있으나 실제 로봇의 움직임이 학생의 의도대로 정확하게 움직이지 못한 경우가 많다.

격자 공간(Grid Space) 상에서 로봇이 정확히 움직이는 것은 코딩용 로봇에 가장 필수적인 기능이나 현재 국내 시장에 출시된 로봇들은 이런 기능을 제공하는 예가 전무하다. 정밀하지 못한 로봇 제어는 향후 로봇 코딩용 교육콘텐츠의 부족을 초래하게 되며 이에 따라 학생들의 코딩교육에 대한 흥미를 저하시킨다.

본 논문에서 이 점을 착안하여 아이들의 학습 동기부여 및 지속성 있는 코딩교육용 콘텐츠 제작을 위해 (1) 안정적인 움직임을 제공하는 아두이노 기반 로봇 제공, (2) 무선통신을 통한 프로그래밍적 로봇 제어용 모바일앱 개발을 목표로 한다. 로봇은 직진 이동, 90도 회전 이동을 기본으로 하며 저가의 센서와 커스터마이징이 가능한 외관 제작이 가능하도록 구성한다. 모바일앱은 초등학교 저학년이 비교적 쉽게 코딩학습에 접할 수 있도록 LightBot[Fig. 1][11] 형식의 격자 분할 공간 내에서 주어진 명령어 패턴에 따라 로봇을 움직일 수 있는 로봇 기반의 코딩교육용 모바일앱 시스템 개발을 목표로 한다.



[Fig. 1] Screenshot of LightBot Application

본 논문의 구성은 다음과 같다. 2장에서는 본 논문이 다루는 시스템의 하드웨어적, 소프트웨어적 구성을 개괄하여 설명한다. 3장에서는 저가 하드웨어 센서 제어를 통한 격자 방식의 보드 환경 하에서 효율적으로 이동, 회전하는 방법 및 보정법 등을 상술하며 4장에서는 실제 실험 환경 및 실험 결과를 제시하고 마지막으로 5장에서는 본 연구의 연구 결과를 정리하며 향후 과제 등을 다룬다.

## 2. 시스템 개요

이 장에서는 본 논문이 목표로 하는 시스템에 대해 기술한다. 시스템은 아두이노 기반 하드웨어 로봇, 로봇 구동용 소프트웨어, 그리고 격자 형태 보드게임용 고무판으로 구성되어 있으며 학습자의 게임 플레이 환경 및 플레이 방식은 다음과 같다.

### 2.1 게임 플레이 환경

사각타일 모양 격자 보드게임판은 코딩교육이 실시되는 실내 교실 공간을 가정하고 있다. 로봇이 이동할 최소 거리는 20cm, 선의 두께는 1.8cm로 선정하고 있으며 실제 교육 공간 내 네 개의 책상을 붙여 연결한 크기에 해당하는 140cm x 140cm (7 x 7 격자 공간 [Fig. 2])이다. 정사각형 타일 위에는 로봇과 각종 타일을 배치할 수 있어 학생들의 몰입감을 높여준다.



[Fig. 2] Our Target Coding Platform

### 2.2 명령어 입력용 모바일앱

#### 2.2.1 시스템 개요

코딩교육용 모바일앱[Fig. 2]은 사용자 명령 입력 큐, 명령어 스케줄러, 센서 제어 로직으로 구성

되어 있다. 사용자 명령 입력 부에는 과제로 주어진 미션을 해결하기 위해 학습자가 로봇을 전진, 회전하는 명령어들을 순차적으로 구성할 수 있다. 또한 자주 사용되는 명령어들을 추상화시킨 함수 개념을 도입함으로써 추상화된 명령어들을 구성하고 반복적으로 호출함으로써 진일보된 알고리즘을 구축할 수 있게 된다.

명령어 스케줄러는 입력된 명령어 큐에서 하나씩 명령어를 꺼내 로봇에게 무선 통신부를 통해 명령어를 전달하여 수행을 지시하고 명령어 수행이 완료되면 명령어 큐에 명시된 다음 명령어들을 순차적으로 가져와 큐에 있는 모든 명령어들을 소진시키는 역할을 담당한다.

이 논문의 설계검증을 위해 제작된 코딩교육용 모바일앱은 Unity3D[12] 게임플랫폼을 기반으로 하고 있으며 안드로이드 환경에서 구동된다.

#### 2.2.2 코딩용 명령어 구성

코딩교육용 모바일 앱에서 제공하는 명령어는 직진, 90도 우회전, 90도 좌회전, 액션 네 가지로 구성되어 있다. 직진 명령어는 로봇의 현재 시점을 기준으로 타일위에서 한 칸 앞으로 이동한다. 두 개의 회전 명령어는 좌측, 우측으로 회전하는 기능만을 담당한다. 액션 명령어는 주어진 목표 지점에 도착하였음을 알려주는 기능으로 비프음 소리를 울리는 역할만을 수행한다. 반면 한 칸 후진과 같은 명령어는 본 로봇에서 제공하지 않는다. 물리적으로 모터를 반전시켜 이동함으로써 후진을 할 수 있으나 그것대신 좌회전 2번, 전진, 좌회전 2번 과 같은 명령어 조합을 권고하고 있다.

이 명령어 집합은 주어진 미션을 클리어하기 위해 여러 개의 명령어 큐를 담아 놓아 이를 순차적으로 아두이노 기기와 통신하여 전송해 로봇을 제어한다.

#### 2.2.3 무선 통신부

명령어 스케줄러와 로봇은 무선망을 통해 클라

이언트/서버 방식으로 동작하며 스케줄러는 클라이언트 역할을 수행한다. 실제 하나의 명령어 수행을 위해 아두이노 기기가 복수개의 센서를 제어하기 때문에 이것을 조율하기 위해 무선 통신부를 통한 클라이언트/서버 모델로 개발되었다.

본 연구에서는 다양한 모바일 통신 방법론 중에서 적은 소비 전력, 제한된 실내 공간에서 활용이 용이한 블루투스를 채택하였다. 대표적인 모바일 환경(iOS와 안드로이드)에 모두 대응할 수 있도록 블루투스 4.0이상 버전으로 사용하였다.

서버역할을 수행하는 아두이노 블루투스 수신부 모듈은 시리얼 연결 방식으로 명령어들을 전송받으며 동기 방식으로 명령어를 하나씩 수행하고 수행 결과를 송신부에 반환하는 방식으로 구현되어 있다.

### 2.3 로봇 하드웨어 구성

하드웨어인 아두이노 로봇은 보드위에서 게임 말과 같이 동작한다. 로봇은 타일 선상을 따라 전진 이동, 회전 이동 기능을 수행한다.

로봇은 로봇 구성물을 담는 베이스 프레임과 베이스 프레임에 덮여질 외형 프레임부로 구성되어 있으며 외형 프레임부에는 다양한 로봇 모형을 3D 프린터를 이용하여 출력하고 해당 요소들을 베이스 프레임에 담을 수 있도록 설계되어 있다. 로봇의 전체 크기는 보드게임 타일 크기의 1/2 정도이다. 외부 전원 없이 구동할 수 있도록 충전가능하며 소형화에 용이한 리튬배터리가 채택되었다.

현재 많이 사용되고 있는 아두이노 로봇은 아두이노 우노(Arduino Uno)[13]를 기본 베이스 모델로 사용하고 있으며 이동 기능을 위해 DC모터를 부착한 바퀴 2개가 가운데 위치하고 보조바퀴가 앞에 달린 모형으로 제작되어 있다.

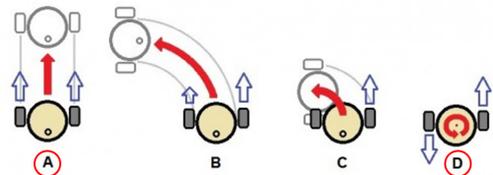
## 3. 모터 오류 보정 방법 설계

본 장에서는 코딩 로봇의 움직임 제어 문제에 대한 정확한 이해를 위해 제어가 가능한 실행환경

에서 선행 실험을 수행하고, 선행 실험에서 관찰된 다양한 현상을 기반으로 본 논문이 다루는 문제를 다루고자 한다. 도출된 문제를 바탕으로 센서 관찰 정보를 바탕으로 한 전진 이동, 회전 이동시 움직임 오류 탐지 및 자동 보정 방법을 제안한다.

### 3.1 선행 실험

#### 3.1.1 모터 제어 모델



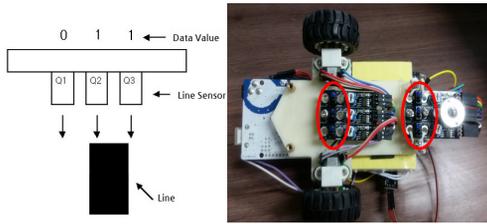
[Fig. 3] 4 Different Types of Robot Movements

일반적 코딩 로봇은 [Fig. 3]에서 보는 바와 같이 A,B,C,D 네가지 주행형태를 가지며 모터의 출력과 주행시간으로 이동거리를 제어한다. 본 논문에서는 A의 방식의 일정 거리를 이동하는 전진이동과 D방식의 지정 위치로부터의 회전 이동을 사용한다.

본 논문에서 다루는 모터 제어는 Pulse Width Modulation(PWM)[14]을 통해 아두이노 기본 전압인 5V를 듀티 사이클(duty cycle)을 이용하여 정수 출력 값(0~255)으로 선행 사상하여 개별 모터의 움직임을 제어하는 방식을 사용한다.

모터의 정회전 혹은 역회전을 위해 하드웨어적으로 전류 방향을 바꾸는 H-Bridge회로[15]를 사용하고 있으며 신호 설정에 따라 전진, 후진, 좌/우 회전 움직임을 제어한다.

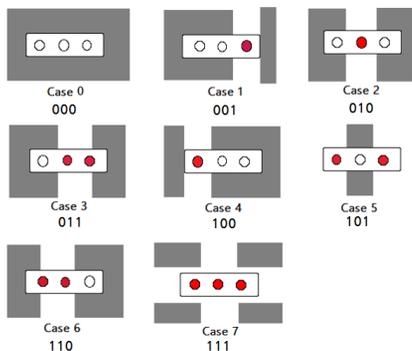
이동시 생기는 현상들의 발생 요인 탐색 및 검증을 위해 다음과 같은 방식으로 실험을 진행하였다. 먼저 로봇에는 엔코더[16] 장치를 부착하여 광학 센서를 활용하여 실시간으로 모터의 회전수를 측정하며, 적외선 라인 추적 센서[17][Fig. 4] (일명 라인센서)를 추가하여 타일 경로를 따라 이동 여부를 확인하도록 하였다.



[Fig. 4] Two Three-Paired Line Sensors : Center For Tile Crossing Detection and Front For 90-Degree Rotation Completion Detection

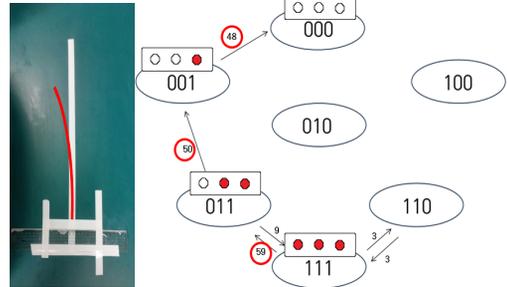
이중 라인센서는 바닥에 적외선을 발사하여 반사된 명암의 세기에 따른 결과 값이 일정 수치 이상이면 High, 그 이하이면 Low로 인식함으로써 이동 경로상 존재하는 라인 존재 여부를 확인한다. 본 실험에서는 [Fig. 4]에서 도시하는 바와 같이 한 타일에서 다른 타일로 이동하는 전진 이동시 라인 경로가 교차하는 지점을 인식할 수 있도록 로봇의 정중앙에 3개의 라인센서를, 회전 이동시 90도 마다 측정 가능한 라인 유무를 측정하기 위해 추가로 3개의 라인센서를 로봇의 전면부에 부착하였다.

3개 단위로 묶인 라인센서들은 [Fig. 5]에서 제시된 바와 같이 총 8개의 상태 정보를 가지고 있으며 로봇의 이동 상황 및 현재 진행 현황을 추정하는데 사용된다. 예를 들어, 중앙에 위치한 3개의 라인센서가 111이라면 현재 타일이 다른 타일로 옮기기 바로 직전 상태로 타일 교차점에 위치한 것으로 판정할 수 있다.



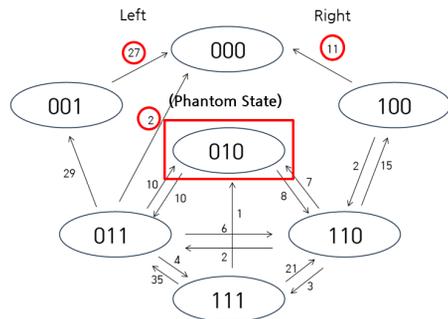
[Fig. 5] All Possible 8 Different Cases of Three-Paired Line Sensor States

### 3.1.2 전진 이동



[Fig. 6] State Transitions of 50 Moving Forward Requests Starting From a Fixed Origin Position. Digits on Edge Represent the Number of Occurrences Among 50 Trials.

본 실험에서는 총 50회의 전진 이동을 시행하였으며 [Fig. 6]에서는 지정된 타일의 중앙위치를 기점으로 시간이 지남에 따라 개별 라인센서들에서 측정된 비트 상태 정보를 상태로 표현하고 이들간 전이 관계를 기술하고 있다. 이 그림에서는 라인센서 정보가 111 -> 011 -> 001로 변화하고 있으며 이는 로봇이 기준위치에서 왼쪽으로 서서히 벗어나고 있음을 보여준다.



[Fig. 7] State Transitions of 50 Different Moving Forward Requests Starting From a Nearby Fixed Center Position.

[Fig. 7]은 지정된 중앙위치에서 약간 벗어난 위치(아이들이 중앙위치에 놓는 것은 오류가 있을 것으로 가정한 상태)에서 50번의 전진 명령을 수행하

였을 때 관찰된 3 라인센서의 상태 정보를 도식화하고 있다. 이 그림에서는 3개의 라인센서가 모두 라인선상에 있음을 알려주는 111 상태에서 오른쪽에 있는 2개의 라인센서값이 011로 (즉, 왼쪽으로 미끄러지기 시작하는) 상황이 50번의 시도 중 35번 관찰되었음을 보여주고 있다. 유사하게, 111에서 오른쪽으로 미끄러지는 경우가 21번 발생했음을 의미한다. 반면 011에서 111로 변화하는 횟수는 4번 발생하였다. 이 현상은 왼쪽으로 미끄러지다 다시 원래 상태로 돌아왔음을 의미하며 왼쪽으로 미끄러지거나(111 → 011) 오른쪽으로 미끄러지는(111 → 110) 경우가 발생하였음을 보여다. 이 결과는 주어진 실험환경에서 보정이 없는 전진 이동의 경우, 일정하게 오른쪽 혹은 왼쪽으로 미끄러지는 경향성이 있음을 보여주고 있다.

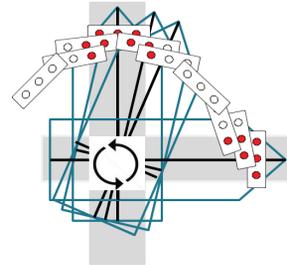
이 미끄러짐 현상에서 재미있게 관찰된 현상은 011 혹은 111, 110 등 라인에 걸쳐있는 상황에서 적지 않은 횟수로 절대 나타날 수 없는 010 상태가 발생했다는 점이다. 이 상태는 실제 011에 110으로(혹은 그 반대 경우) 변경되는 실험에서 관찰되었으며 로그 분석 결과, 011→111→110으로 변경되는 상황이 지속적으로 발생하였음을 확인하였다.

본 저자는 센서 잡음에 의해 유령상태(Phantom State)인 010이 나타나는 것으로 해석하고 있으며 해당 상태가 나타난 모든 경우에 대한 분석을 통해 010 상태는 그 이전상태와 동일한 상태로 판정해도 전혀 무방함을 확인하였다. 기존 관찰 기반 모델에서는 이와 같은 유령상태에 대한 보고가 없으며 본 연구에서는 유령상태에 대응하여 그 이전상태로 변이하여도 전체 평가 결과에 영향을 미치지 않음을 확인할 수 있었다.

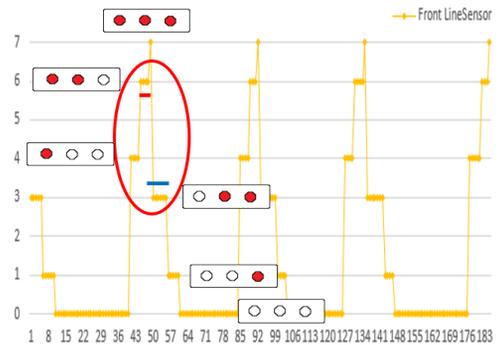
### 3.1.3 회전 이동

로봇의 회전 이동은 전진 이동과는 상이하다. 중앙에 위치한 3개의 라인센서로는 회전상태를 평가할 수 없기 때문에 본 실험에서는 추가로 로봇의 전면부에 라인센서를 3개를 부착하여 90도 방향마다 위

치한 라인을 탐지하여 원하는 회전을 검증하는 용도로 사용하였다. [Fig. 8]은 회전이동에 따른 전면부 라인센서의 상태정보 변화를 도식화시켜 보여준다.

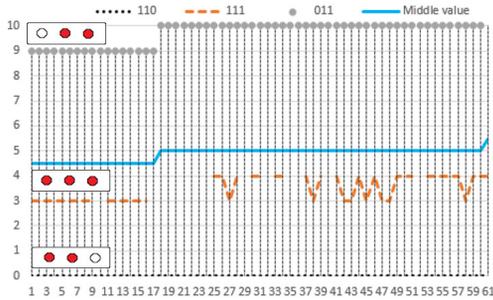


[Fig. 8] State Transitions During Rotation



[Fig. 9] Typical Transition Patterns of Frontal Three-Paired Line Tracking Sensors During Rotation-Right.

[Fig. 9]에서 보여주는 바와 같이 우회전의 경우, 라인을 발견하지 못하는 상태(000)에서 100 → 110 → 111 → 011 → 001 → 000 과 같이 라인센서의 탐지 결과가 변화한다. 하지만 라인센서가 111을 탐지하거나 혹은 110에서 011로 라인을 탐지하지 못하고 지나치는 경우가 관찰되었다. 따라서 전면부 3 라인센서의 정보가 111 혹은 110 → 011 혹은 011 → 110으로 변경하는 시점을 탐지할 수 있으며 그 결과 변경지점을 90의 배수 형태의 회전으로 평가할 수 있다고 결론지을 수 있었다.



[Fig. 10] Statistics of 50 Different Right Rotation Trials (X-Asix) Between 011 Signal Appearance (Entrance: 0) and 110 Signal Disappearance (Exit), Sorted by Maintenance Period

[Fig. 10]에서는 우회전의 경우, 90도 회전 즈음에 발생한 라인센서의 상태 정보의 변화 추이를 보여주고 있다. 이 그림들에서 점선으로 표시한 부분은 111이라는 정지선을 탐지한 시점이 정지선 근처에 진입한 이후 진행된 경과 시간을 보여주고 있으며 파란색 단선으로 표시한 지점은 정지선 근처에 진입한 시간과 정지선 근처를 탈출한 시점의 중간값(median) 정보를 보여주고 있다. 이 그림들로부터 정지선의 정확한 중앙 지점을 탐지하는 것은 회전 관성 때문에 느리게 혹은 빠르게 관정됨을 알 수 있다. 따라서 정확한 중앙 지점(111이라는 라인센서 정보가 탐지된 시점)에서 회전을 멈출 것인지 아니면 정지선 진입 시점 이래 몇 시간 사이클 후에 멈출지에 대한 두 가지 선택이 가능하다. 이들 선택 방법에 대한 상세한 내용은 3.2.2절

에 기술되어 있다.

### 3.1.4 문제원인 및 선행실험 결론

선행실험을 수행한 결과, 개별 DC 모터에 동일 출력 값을 주어 일정 주기마다 반복적으로 수행하는 전진 이동의 경우만 해도 미끄러지는 현상이 관찰되었다. 이 현상은 하드웨어의 각 바퀴의 평행 여부, 진행방향과 모터 방향의 일치 여부, 바닥의 기울기 여부, 바닥과 각 모터바퀴의 마찰계수 로봇의 무게에 의한 관성, 로봇의 무게중심, 회전축 마지막으로 DC모터를 아두이노가 제어할 때 생기는 전압의 노이즈 등 다양한 원인에 기인한다. 또한 전진 이동과 회전 이동이 혼합되어 동작할 때 정확한 위치 도달 여부와 회전 여부에 따라 지정된 타일의 라인을 이탈할 수 있으며 한 타일에서 다른 타일로 이동한 후 정지하는 시점을 결정하기 어려운 문제가 발생하였다.

## 3.2 관찰 기반 모터 제어 보정 방법

본 장에서는 3.1장에서 기술한 관찰 결과를 바탕으로 한 모터 제어 보정 방식을 제안한다.

### 3.2.1 전진 이동

전진 이동의 경우, 유령상태만을 보정함으로써 우측 혹은 좌측 미끄러짐 관찰시 반대 방향으로 기

[Table 1] Voltage Adjustment Value Table From a Previous Line Sensor State To a Current Line Sensor State. The Values in Each Entry Contain Two Adjustment Values: Left Value for Left Motor and Right Value for Right Motor.

Current state \ Previous state	000	001	010	011	100	101	110	111							
000	-	-	5	-5	-	-	1	-	-5	5	-	-	1	-	-
001	3	-3	3	-3	-	-	1	-	-2	2	-	-	-1	1	-
010	-	-	4	-4	-	-	2	-2	-4	4	-	-	-2	2	-
011	2	-2	3	-3	-	1	1	-1	-2	2	-	-	-1	2	-
100	-3	3	-	-	-	-	1	-1	-3	3	-	-	-	1	-
101	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
110	-2	2	2	-	1	-	2	-1	-3	3	-	-	-1	1	-
111	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

본 모터 입력 값에 보정 값을 더하는 방식이다. 주기적으로 중앙에 위치한 3개의 라인센서 정보를 갱신하고 갱신된 값이 이전 갱신된 정보와 현재 관찰된 정보를 비교하여 [Table 1]에서 제시된 결과 값을 반영하여 개별 모터의 보정 출력 값을 지정한다. 이를 통한 보정테이블은 다음과 같으며 개별 조정 값은 모터에 따른 상수 일반화된 값으로 조절된다. 이 보정테이블 값은 현재상태의 기본 보정값에 이전상태로부터의 추가 보정값을 합한 값을 의미하며 정규화한 것이며 다음과 같은 공식을 따른다.

$$v_{motor} = v_{Base} + p \times K$$

예를 들어, 버퍼에 저장된 이전 라인센서의 값이 110이고 현재가 100이면 로봇은 라인기준으로 진행방향의 오른쪽방향으로 이탈중임을 알려준다. 이 경우 보정테이블에서 이전상태가 110이며 현재상태가 100일 때의  $p$ 에 해당하는 모터별 보정값 -3, 3을 로봇 속도상수  $K$ 만큼 곱한 값만큼 더해주어 왼쪽모터의 속도를 낮추고 오른쪽모터의 속도를 높여 좁아오도록 왼쪽으로 쏠린 로봇을 라인 안으로 들어오도록 보정하는 방식이다.

### 3.2.2 회전 이동

회전 이동의 보정방법은 전진부에 배치된 3-라인센서의 상태 정보를 바탕으로 하고 있으며 크게 두 가지 방식의 보정 방법을 제시한다. 첫 번째는 라인센서 정보가 111 신호를 탐지하였을 때 즉시 정지방식이고 다른 하나는 좌회전 혹은 우회전시 정지선 진입 지점 기준으로 일정 사이클 후에 강제 정지방식이다. 전자의 경우, 현재 회전 상황을 바로 반영하는 방법이고 후자의 경우, 현재 회전에서 진입지점 탐지 시점 이래 계산에 의해 정지선의 중앙 위치에 위치하였을 때 자동으로 정지하는 방식이다.

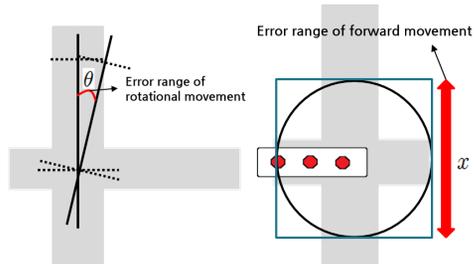
### 3.2.3 오차 허용 범위

모든 움직임에는 오차 허용 범위가 있으며 본

문에서는 움직임별 오차 허용 범위를 라인의 특징값들로부터 도출하는 식을 제시한다.

$$E(dist) = 3\text{라인센서 길이} \times 2$$

$$E(\theta) = \text{atan} \frac{\text{라인두께}/2}{\text{전면부라인센서와 중앙부라인센서거리차}}$$

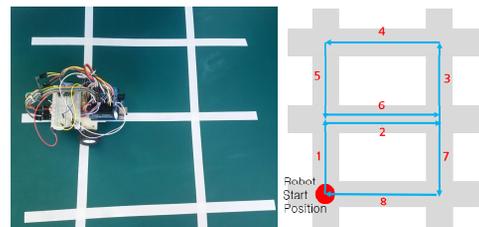


[Fig. 11] Illustration of Allowable Margin of Errors for Distance Difference and Angle Difference Before and After Movements, Respectively.

본 식에 따라 실험에서 사용한 개별 3 라인센서의 길이는 2.3cm, 전면부와 중앙부의 거리는 6.5cm, 라인의 두께는 1.8cm일 때 거리 오차 허용 범위는 4.6cm, 각도 오차 허용 범위는 7.5도로 정한다.

## 4. 실험 결과

선행 실험 결과를 바탕으로 구축된 실험환경은 [Fig. 12]와 같이 실제 보드환경을 축소한 두 개의 정사각형 타일로 구성된 그리드를 타겟 실험공간으로 설정하였다.



[Fig. 12] Experimental Setup With Two Tiles

- 1) 로봇 속도 상수는 로봇 무게에 의해 결정됨.

미끄러짐 현상을 최소화하기 위해 고무매트로 구성된 그리드판을 사용하였으며 보드 상에 흰색 테이프로 개별 타일 공간을 구분하였다.

본 실험에서의 사용자 명령은 전진이동, 90도 좌회전, 90도 우회전 이동 등 세 가지의 상이한 로봇 코딩 명령어로 구성되어 있다. 3장에서 제시한 보정 알고리즘의 정확도 여부를 측정하기 위해 8자 모양으로 이동 및 회전을 혼합한 명령어를 수행하고 최종 시작 위치로 돌아왔을 때 회전각도 및 위치 차이의 변이를 측정하여 정확도를 산출하였다.

의도적으로 원래 위치와 최종 이동 위치를 동일한 위치로 설정하였으며 이때 최종 위치의 회전각도를 별도로 측정하여 원 위치와 비교하였다. [Fig.13,14]는 총 10회 8자형 움직임후의 로봇의 위치 분포를 도시하고 있다. 요약하면, 10회 반복시 회전 각도 차이의 절대값 평균은 2.38도이며 거리 차이의 절대값 평균은 1.57cm이었다. 이 결과는 오차 허용 범위의 34%(거리), 32%(각도)에 해당 정도로 본 논문에서 제안한 방식이 매우 훌륭하게 작동하고 있음을 보여주고 있다.

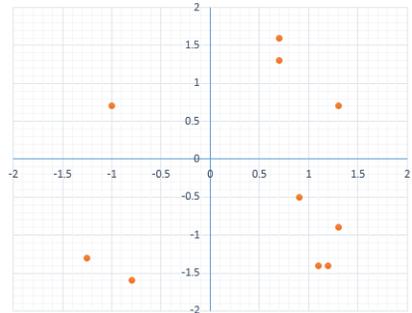
## 5. 결 론

본 논문에서는 타일 기반 코딩교구에서 활용할 수 있는 격자 공간 내 정확하게 전진 이동, 회전 이동하는 로봇의 모터 움직임 현상을 관찰하고 그 관찰 결과를 바탕으로 모터 제어의 정확도를 높이는 전진이동, 회전 이동 방법을 제안하였다.

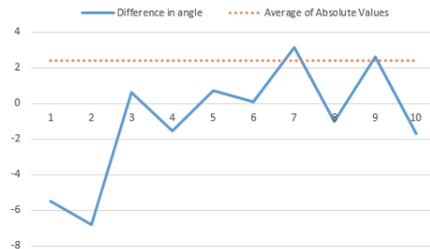
제안한 방식을 도출하기 위해 현실 공간에서 이용할 수 있는 실험 환경을 구축하였으며 모바일 앱을 개발하여 블루투스를 통해 실제 명령어를 수행시켜 로봇이 원하는 목표로 정확히 이동할 수 있음을 검증하였다. 또한 총 6개의 라인센서(회전 탐지를 위한 전면부 3개, 타일 중앙부 탐지를 위한 중앙부 3개) 사용 모델을 제안하고 활용함으로써 고가의 모터나 IMU센서 장치 없이 저가격대의 안정성 높은 로봇 모터 제어 방식의 활용가능성을

확인하였다.

하지만 테스트 중에 배터리 용량문제로 실제 사용 시간이 1시간에 그치고 있으며 과도한 센서 샘플링에 따른 배터리 소비문제 등 다양한 문제가 산적해 있다. 또한 현재 사용 중인 총 6개의 라인센서의 개수를 최소화시키는 문제는 향후 본 연구진에서 지속적으로 해결해야 할 것이다.



[Fig. 13] The Distribution of the Final Positions of A Robot After 10 Trials of 8-shape Movements. The Origin Position Represents the Desired Location After Movement.



[Fig. 14] Angular Difference Between Start Position and End Position, Red-Dotted Line Represents the Average of Absolute Values of Angular Difference.

위에 제기된 문제가 해결하게 된다면 저가격으로 높은 정밀도를 갖춘 코딩용 로봇을 상용화할 수 있으며 향후 코딩용 교구 시장에서 다양한 코딩 교육용 콘텐츠 개발에 일익을 담당할 것으로 기대한다.

## ACKNOWLEDGEMENT

This work was supported by 2015 Hongik University Research Fund.

## REFERENCES

- [1] Korean Ministry of Education, Software Education Guidance, 2015.
- [2] SungSu Jin, PhanWoo Park, The Effects of Programming Learning on the Improvement of Problem Solving Ability Using MCU, Journal of The Korea Association of Information Education, Vol 14, No. 3, pp. 319-328, 2010.
- [3] Pamela B Lawhead et al, A Road Map for Teaching Introductory Programming Using LEGO Mindstorms Robots, ACM SIGCSE Bulletin, Vol. 35, No. 2, pp. 191-201, 2002.
- [4] Jeong-Hyun Seo, Young-Sik Kim, Study on educational utilization of physical computing using Arduino, In Proceedings of the Korean Association of Computer Education, Vol. 16, No. 2, pp. 103-107, 2012.
- [5] Seok-Jeon Kim, Young-Ju Jeon, Tae-Young Kim, A Practical Approach to Arduino Programming for The Physical Computing Section of The Informatics Curriculum in Korean Middle School, In Proceedings of the Korean Association of Computer Education, Vol. 20, No. 2, pp. 29-34, 2016.
- [6] A Harvard Business Review Analytic Services Report, Internet of Things: Science Fiction or Business Fact? Harvard Business Review, 2014.
- [7] Ki-hyuk Kim, Gwi-Im Ahn, Hwan-Seob Lim, Deok-Gil Jung, Research about senior citizen IT start-up education linking the IoT, Journal of Korea Institute of Information and Communication Engineering, Vol. 19, No. 11, pp. 2710-2716, 2015.
- [8] <https://www.bee-bot.us/>
- [9] <http://albert.school/>
- [10] <http://hellogeeks.kr/bitbrick/>
- [11] <https://lightbot.com/hocflash.html>
- [12] <https://unity3d.com/>
- [13] <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [14] Katharina Hausmair, Shuli Chi, Peter Singerl, Christian Vogel, Aliasing-Free Digital Pulse-Width Modulation for Burst-Mode RF Transmitters. IEEE Transactions on Circuits and Systems, Vol. 60, No. 2, pp. 415 - 427, 2013.
- [15] Al Williams, Microcontroller Projects Using the Basic Stamp (2nd Ed.), CRC Press, ISBN 978-1578201013, 2002.
- [16] G. K. McMillan, D. M. Considine (ed), Process Instruments and Controls Handbook Fifth Edition, McGraw Hill, pp. 5-26, ISBN 978-0070125827, 1999.
- [17] [https://www.dfrobot.com/wiki/index.php/Line\\_Tracking\\_Sensor\\_for\\_Arduino\\_\(SKU:SEN0017\)](https://www.dfrobot.com/wiki/index.php/Line_Tracking_Sensor_for_Arduino_(SKU:SEN0017))



박 경 복 (Park Kyeong bok)

2015 홍익대학교 게임학부 학사  
2016- 홍익대학교 게임공학 석사

관심분야 : 교육용 게임 개발, 기능성 게임

---



조 성 현 (Cho, Sung Hyun)

1978 서울대학교 계산통계학과 이학사  
1980 서울대학교 계산통계학과 이학석사  
1995 UCLA 컴퓨터과학과 이학박사  
1996- 홍익대학교 게임학부 교수

관심분야 : 게임프로그래밍, 게임인공지능

---



서 범 주 (Seo Beom joo)

2001 LG전자 DTV연구소 주임연구원  
2012 싱가포르국립대 Senior Research Fellow  
2013- 홍익대학교 게임학부 조교수

관심분야 : 교육용 게임 개발, 가상현실, 분산 멀티미디어  
데이터베이스 시스템 설계

---

