이동로봇을 중심으로 LEGO MINDSTORM을 응용한 로봇공학 교육용 실습 로봇개발

Development of Experimental Mobile Robots for Robotics Engineering Education by Using LEGO MINDSTORM

박 준 형[1], 정 슬[†]

June Hyung Park[1], Seul Jung[†]

Abstract This paper introduces several mobile robots developed by using LEGO MIDSTORM for experimental studies of robotics engineering education. The first mobile robot is the line tracer robot that tracks a line, which is a prototype of wheel-driven mobile robots. Ultra violet sensors are used to detect and follow the line. The second robot system is a two-wheel balancing robot that is somewhat nonlinear and complex. For the robot to balance, a gyro sensor is used to detect a balancing angle and PD control is used. The last robot system is a combined system of a line tracer and a two-wheel balancing robot. Sensor filtering and control algorithms are tested through experimental studies.

Keywords: LEGO Mindstorm, robot education, line tracer, balancing robot

1. 서 론

우리나라처럼 인적 자원이 많고 수출에 의존하는 나라는 산업의 발달이 매우 중요하다. 산업은 점차 무인화 및 자동화되어 가고 있으며, 산업의 발달을 뒷받침하는 것이 바로 공학교육이다.

최근에는 산업현장에서 이론 보다는 실습위주의 공학교육이 요구되고 있고, 한 분야의 지식보다는 다양한 분야의 지식, 즉 지식의 융합이 요구되고 있는 실정이다.

특히 신성장 동력산업의 하나로 선정된 지능 로봇 분야는 미래 우리나라를 책임질 항목의 하나로 선정된 바 있다. 로봇 분야의 기대가 커지고 발달이 전망되면서 그에 따른 로봇 공학 교육 프로그램 개발도 절실히 필요하게 되었다.

로봇공학은 한 분야의 지식으로 개발할 수 있는 항목이 아니라 다양한 학문의 지식을 바탕으로 융합을 필요로 하는 학문이다. 또한 로봇공학은 이론만으로는 무의미하며 로봇의 부품, 설계, 제작, 센싱, 제어, 지능 이론 등의 모든 공학 분야의 기술을 통합하여 구현되어지는 시스템이다[1].

대학 교육에서 로봇공학의 이론을 실험적으로 구현하기 위해서는 고가의 로봇 장비들이 필요하지만 현실적으로 모두 갖추기가 쉽지 않다. 로봇 공학의 커리큘럼을 보면 대부분 로봇공학이론을 중심으로 시뮬레이션을 통해 검증하는 프로젝트나 연구가 주를 이루고 있다. 또한 최근의 로봇의 추세를 보면 이동로봇분야의 비중이 매우 커짐을 알 수 있다. 하지만, 대학 교육에서 이동로봇을 가르치기 위한 커리큘럼을 만들기가 어려운 것이 사실이다.

이를 보완하기 위해서는 연구 프로젝트를 통해서 제작한 로봇을 교육에 사용하는 것이 한 방법이다. 하지만, 이는 로봇의 몸체를 만들고 제작하고 필요한 부품을 조립하는 것에 대한 시간과 예산의 어려움이 따른다. 실습중심의 로봇교육이 해결해야 할 숙제인 것이다.

시간과 예산문제에 대한 대안으로 LEGO(레고) 블록은 좋은 도구가 된다[2]. 레고의 가장 큰 장점은 작은

Received : Jul. 27. 2011; Reviewed : Oct. 20. 2011; Accepted : Mar. 8. 2012
※ 이 논문은 한국연구재단의 특정기초와 지식경제부의 융복합형 로봇전문인력 양성사업의"자율지능형 매니퓰레이션 연구센터(AIM)"를 통하여 지원에 의하여 연구되었으며, 이에 감사를 드립니다.
† 교신저자 : 충남대학교 메카트로닉스공학과 교수(jungs@cnu.ac.kr)
[1] 충남대학교 메카트로닉스공학과 석사(vij232@hyundai-wia.com)

블록들을 이용하여 여러 가지 다양하고 창의적인 모델을 만들 수 있도록 다양한 모양과 크기의 제품들을 제공하는 것이다.

최근에는 단순히 정적인 모델의 제작이 아닌 동적인 움직임들이 가능한 모델을 제작할 수 있도록 MCU (Micro Control Unit), 각 종 센서, 모터 등을 제공하고 있다. 그 중 NXT라는 제품군은 쉽고 빠른 시간에 센서와 모터를 연동한 하나의 모델을 제작할 수 있고, ROBOTC언어 기반의 소프트웨어를 통하여 센서의 신호처리와 모터제어를 할 수 있도록 되어 있다. 이러한 장점을 이용하여 레고 마인드스톰을 초등학생 로봇 교육에 이용한 사례가 있다[3].

대학의 로봇공학교육에서 한 학기 동안 센서, 신호처리, 제어와 하드웨어 제작을 모두 배운다는 것은 다소 무리가 있다. 이 경우에 NXT를 사용하면 하드웨어의 구성이 쉽고, 소프트웨어도 쉽게 접근할 수 있어서 한 학기 동안 여러 가지 센서와 함께 신호처리 및 모터 제어를 실습할 수 있다.

외국 대학에서는 공학교육에 LEGO를 사용하여 실습 효과를 높이고 있으며[4], 최근에 국내대학에서도 레고 NXT시스템을 공학설계입문교육에 사용하고[5], 이동로봇 자기 위치인식 연구에 사용하며[6], 프로그램 교육과 로봇 교육과정의 도구로 사용하는 등[7-10] 공학교육에의 응용이 늘어가고 있다.

본 논문에서는 로봇 공학의 효율적인 교육을 위해 로봇실습 재료로 LEGO를 사용한 예를 제시한다. 이동로봇을 중심으로 세 가지 로봇 시스템을 제안한다. 라인트레이서를 비롯하여 선행 로봇 연구 결과를 로봇공학 교육에 접목하여 레고 NXT시스템기반의 밸런싱 로봇을 제작하여 실험 하였다. 레고 NXT를 이용하여 실습이 가능한 로봇 시스템을 난이도와 연결성을 중심으로 제작하고 실험하여 실험교재로의 사용 가능성을 확인하였다.

2. 로봇공학 교육 시스템 개발

이 장에서는 로봇공학 교육실습에 적합한 로봇을 실제로 구현하고 소개한다.

첫 번째는 이동로봇의 기본인 라인트레이서 로봇의 제작이다. 적외선 센서를 이용하여 선을 검출, 추종하고 초음파 센서를 이용하여 장애물을 감지하는 라인트레이서의 개념은 미래 전기 자동차의 개념과 일치하므로 교육용으로 매우 적합하다.

두 번째 시스템은 두 바퀴로 움직이는 이동로봇 기능과 역진자 시스템의 균형을 유지하는 기능을 합한 모바일 역진자 시스템이다. 자이로 센서를 이용하여 밸런싱 각도를 검출하여 항상 균형을 유지하도록 제어한다.

마지막으로 라인트레이서 기능과 모바일 역진자 기능을 하나로 합친 밸런싱 라인트레이서 시스템이다. 소프트웨어는 ROBOTC를 사용하였다. 표 1은 각 로봇에 대한 교육 내용과 대상을 설명한다.

표 1. 실습 가능한 로봇 레고 시스템
Table 1. Robot LEGO system for experiments

	교육내용	교육대상
Line tracer	• 이동로봇기구학 • 제어 방법 • 적외선 센서기술	학부생
Balancing robot	• 역진자 시스템 개요 • 제어 방법 • 자이로 센서기술	학부생 대학원생
Balancing line tracer	• 역진자 시스템 개요 • 제어 방법 • 적외선및자이로센서기술	학부생 대학원생

2.1 이동 로봇의 기구학

라인트레이서는 이미 많은 공학도들과 고등학생들에게 알려진 로봇이다. 국내에는 매년 여러 대학에서 라인트레이서 대회를 개최한다. 레고를 통한 라인트레이서의 구현은 적외선 센서 두 개를 이용하여 선을 인식하고 모터 두 개로 구동하는 형태이다. 또한 초음파 센서를 통해 전방에 장애물이 감지되면 정지하고, 없어지면 전진하는 동작을 구현하였다. 그림 1은 제작된 라인트레이서의 모습이다.

무게 중심에서 이동로봇의 위치는 x, y 그리고 오리엔테이션은 Φ에 의해서 정의된다. 따라서 입력은 두

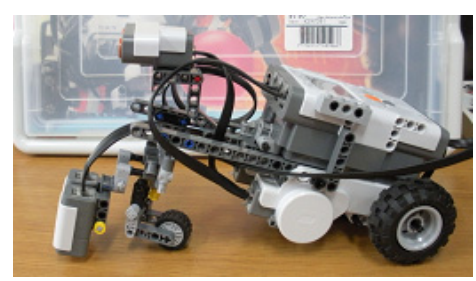

그림 1. NXT를 이용한 라인트레이서
Fig.1. Line tracer using NXT

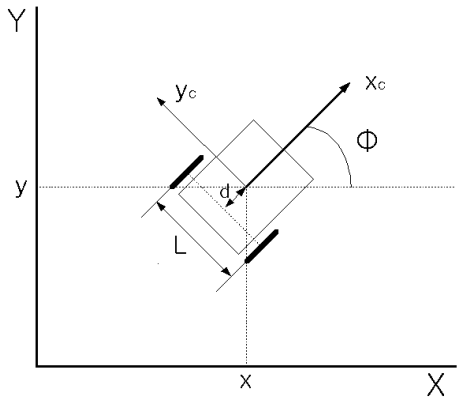

그림 2. 라인트레이서 기구학
Fig. 2. Kinematics of line tracer

바퀴의 구동 토크로 둘인 반면에 로봇의 상태를 나타내는 변수는 x,y,ϕ로 셋인 대표적인 under-actuated 시스템의 하나이다.

이동로봇의 선속도를 v와 각속도를 w라 할 때 위치 (x,y)에서의 속도는 다음의 행렬식을 만족한다.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos\phi - d\sin\phi\\ \sin\phi \quad d\cos\phi\\ 0 \quad\quad 1\end{bmatrix}\begin{bmatrix}v\\ w\end{bmatrix} \tag{1}$$

여기서 d는 바퀴 중심점에서 무게 중심점까지의 거리이다. 이동로봇의 선속도 v는 다음과 같이 각 바퀴의 평균 각속도로 표현될 수 있다.

$$v = \frac{1}{2}(r\dot{\theta}_R + r\dot{\theta}_L) \tag{2}$$

여기서 θ_R,θ_L는 각각 오른쪽, 왼쪽 바퀴의 회전각이고 r은 바퀴의 반지름이다.

로봇의 각속도 w는 다음과 같이 각 바퀴의 선속도에 의해 표현될 수 있다.

$$w = \frac{v_R}{L} - \frac{v_L}{L} = r\left(\frac{\dot{\theta}_R}{L} - \frac{\dot{\theta}_L}{L}\right) \tag{3}$$

여기서 L은 두 바퀴 사이의 거리이다.

식(1)-(3)을 정리하면 다음과 같은 기구학 식을 구하게 된다. 따라서 식(4)를 사용하여 각 바퀴의 속도를 제어하여 이동로봇의 속도를 제어할 수 있다.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\frac{r}{2}\cos\phi - \frac{rd}{L}\sin\phi & \frac{r}{2}\cos\phi + \frac{rd}{L}\sin\phi\\ \frac{r}{2}\sin\phi + \frac{rd}{L}\cos\phi & \frac{r}{2}\sin\phi - \frac{rd}{L}\cos\phi\\ \frac{r}{L} & \frac{-r}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}_R\\ \dot{\theta}_L\end{bmatrix} \tag{4}$$

기구학 식(4)는 본 논문의 모든 이동 로봇의 기구학에 적용된다.

2.2 라인트레이서 로봇

2.2.1 하드웨어 제작

적외선 센서는 최대한 바닥에 가깝게 놓이도록 해서 외부의 간섭을 받지 않도록 하였고, 두 적외선 센서를 붙여 그 사이에 라인이 위치하도록 하였다. 초음파 센서는 정면을 향하도록 기울기를 맞춰서 조립하였고, 앞바퀴는 전 방향으로 회전이 가능하도록 하며, 방향 전환 시 상승, 하강이 최대한 적도록 설계하였다. 그림 3에서는 각 센서와 바퀴의 부착된 모습을 나타낸다.

그림 3. 라인트레이서 센서 및 바퀴
Fig. 3. Line tracer sensor and wheels

2.2.2 ROBOTC 프로그램

레고를 제어하는 프로그램으로 ROBOTC를 사용하였다. 그림 4는 ROBOTC 프로그램의 창으로 Robot - Motors and Sensors Setup을 클릭하여 사용할 모터와 센서의 이름을 지정해주고 연결 포트를 설정해준다.

프로그램 방법 중 중요한 몇 가지를 열거하여 보았다.

1. Task : ROBOTC는 Multitasking을 지원한다. Task라는 키워드를 사용하여 최대 256개의 태스크를 사용할 수 있다. 하지만 task main() 문은 하나만 존재해야하며 다른 태스크 문들이 진행 중일 때 main 문이 종료되면 안된다. main 문이 종료되면 모든

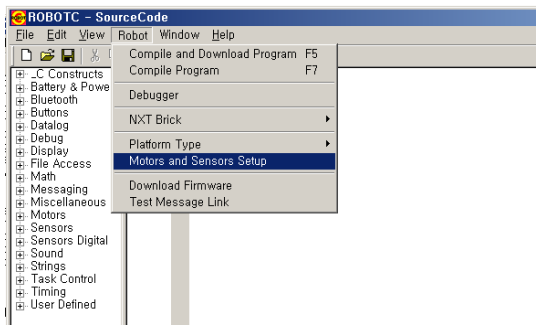

그림 4. ROBOTC 프로그램
Fig. 4. ROBOTC Program

태스크 문이 자동 종료된다.

2. Motor : 모터를 구동하기 위해서는 사용할 모터를 설정해주고 'motor[motorA]=변수' 의 형태로 사용하면 된다. 모터는 A, B, C 세 포트를 사용할 수 있으며 그 값은 ±100 까지 사용가능하다.
3. Sensor : 센서 값을 받기 위해서는 사용할 센서와 연결포트, 그리고 변수명을 설정해주고 'SensorValue (변수명)'을 통해 값을 입력받으면 된다. 레고에서 제공하는 기본 센서들은 0~100의 값이 출력된다. 타 회사 센서의 경우 값의 범위가 다르므로 그 값에 맞게 사용하도록 한다.
4. Delay : 시간지연을 발생시키기 위해서는 'wait1Msec (숫자)'를 사용하면 된다. 숫자는 msec 단위 이므로 1초의 시간지연을 위해서는 1000을 입력하면 된다.
5. Display : NXT화면에 특정값을 나타내려면 nxtDisplayText(위치, "형식", 변수)를 사용한다. 위치는 NXT화면의 행을 말하며 0부터 시작된다. 형식은 %d(정수형), %u(부호없는 정수형), %c(문자형), %s (문자열), %f(실수형)을 나타내고, 변수에는 나타내고자 하는 변수값을 넣어주면 된다.

일례로 컴파일과 다운로드를 실행한 후에 Robot - Debug windows - NXT Device를 선택하면 그림 5와 같은 창이 생성된다. 현재 로봇의 상태를 알 수 있다.

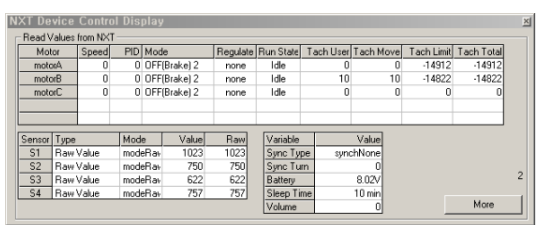

그림 5. Debug windows 프로그램
Fig. 5. Debug window program

그림 6은 라인트레이서 프로그램의 예이다.

```
  #pragma config(Sensor,S1,Line_1,sensorLightActive)
  #pragma config(Sensor,S2,Line_2,sensorLightActive)
  #pragma config(Sensor,S3,Ultra, sensorSONAR)
  #pragma config(Motor,motorA,AAA,tmotorNormal,PIDControl)
  #pragma config(Motor,motorB,BBB,tmotorNormal,PIDControl)
  //*!!Code automatically generated by 'ROBOTC'
configuration wizard !!*//
  int Left=0;
  int Right=0;
  task main()
  {
   while(true)
   {
    if(SensorValue(Ultra)>50)
    {
    Left=SensorValue(Line_1);
    Right=SensorValue(Line_2);

     if(Left > 50 && Right > 50)
     {
      motor[motorA]=40;
      motor[motorB]=40;
     }
     else if(Left < 50 && Right > 50)
     {
      motor[motorA]=20;
      motor[motorB]=40;
     }
     else if(Left > 50 && Right < 50)
     {
      motor[motorA]=40;
      motor[motorB]=20;
     }
    }
    else
    {
      motor[motorA]=0;
      motor[motorB]=0;
    }
   }
  }
```

그림 6. 라인트레이서 프로그램 예
Fig. 6. Program example of line tracer

2.2.3 제어 방법

그림 7에서처럼 라인과 장애물을 감지하는 센서의 값을 통해 오른쪽, 왼쪽 바퀴의 속도를 조절하는 제어 방식을 사용하였다. 적외선 센서로부터 라인이 인식되지 않은 상태의 값을 측정해서 이를 표준으로 사용하였다. 라인트레이서 작동 시 정면을 향하고 있는 초음파 센서의 일정 거리 안에 장애물이 없으면 라인트레이서는 주행을 하게 된다. 주행 시 장애물이 감지되면 다시 멈추고 장애물이 제거되면 다시 주행한다.

라인을 인식하는 적외선 센서에 표준 값보다 일정

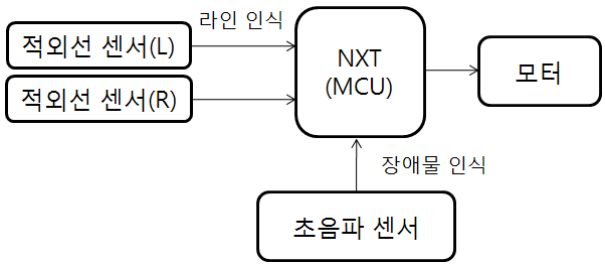

그림 7. 라인트레이서 제어 순서도
Fig. 7. Control flow of line tracer

값 이상 작은 값이 들어오면 감지된 쪽의 센서 반대쪽 모터의 속도를 높여주고, 같은 쪽 모터의 속도를 낮춰줌으로써 라인을 따라 움직이게 하였다. 적외선 센서가 둘이므로 모터의 속도를 너무 빠르게 하면 라인의 곡률에 반응하기 어렵다.

2.2.4 실험

라인은 흰색 바탕에 검정색으로 그렸다. 속도를 조절하여 아래 그림 8에서와 같은 곡률이 큰 라인도 잘 따라가는 것을 확인하였다. 센서를 통해 정면에 장애물이 있을 경우에 멈추는 것을 확인 하였다.

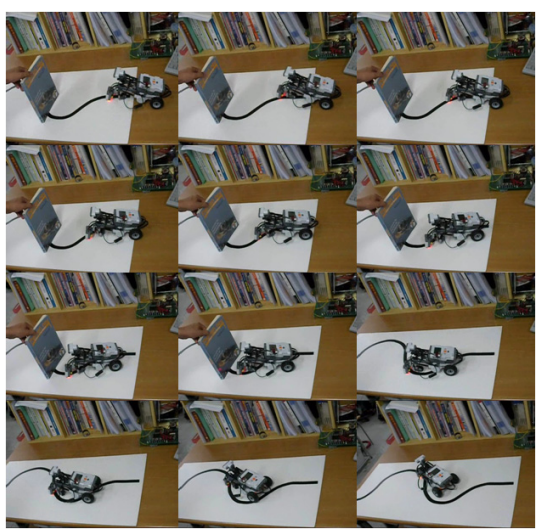

그림 8. 라인트레이서 동영상
Fig. 8. Image cut of line tracer

2.3 두 바퀴 구동 밸런싱 로봇

두 바퀴 구동 역진자 시스템은 매우 도전적인 비선형 제어 응용 시스템이다. 라인트레이서와 달리 두 바퀴로 균형을 유지해야 하므로 제어가 쉽지 않다. 선행연구 결과를 바탕으로 레고 기반으로 투 휠 밸런싱 로봇을 제작하였다. HiTechnic 사의 자이로 센서를 이용하여 로봇의 기울기 값을 얻었고, 센서의 특성에 맞게 샘플링 타임과 Dead zone을 설정하여 제자리에서 두 바퀴로 균형을 잡도록 하였다.

2.3.1 하드웨어

두 바퀴 구동 밸런싱 로봇의 경우에 균형이 가장 중요하므로 앞뒤가 최대한 같도록 설계하였다. 자이로 센서는 진동에 의한 간섭이 심하므로 최대한 몸체에 고정을 시켰고, 케이블을 감아서 진동을 더 감소시켰다. 몸체와 모터 또한 완전히 고정을 시킬 수는 없으나 최대한 몸체와 고정을 시켜서 균형을 잡는데 외란으로 작용하지 않도록 하였다. 그림 9는 두 바퀴 구동 밸런싱 로봇을 나타낸다.

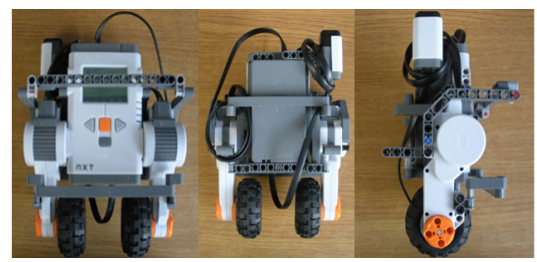

그림 9. NXT를 이용한 투휠 밸런싱 로봇
Fig. 9. Two-wheel balancing robot using NXT

2.3.2 제어방법

자이로 센서의 값이 민감하게 바뀌므로 오히려 제어에 방해가 되는 것을 확인할 수 있었다. 이를 해결하기 위해 Dead Zone을 설정해주고, 제어주기가 필요 이상으로 빨라서 자이로 센서의 샘플링 속도를 프로그램으로 낮춰주었다.

전체적인 제어를 위해 자이로 센서에서 각속도 값을 얻게 되고 이를 적분하여 기울기 값을 구한다. 로봇이 기울어지게 되면, PD 제어기의 계산에 의한 제어

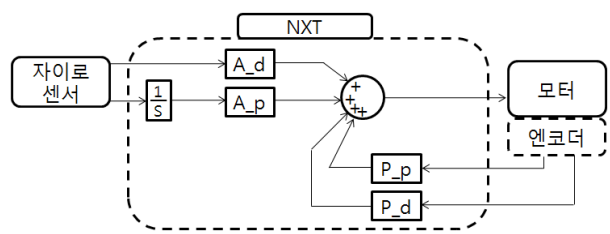

그림 10. 투휠 밸런싱 로봇 제어 블록다이아그램
Fig. 10. Control block diagram for two-wheel balancing robot

표 2. 제어기 이득값
Table 2. Control gains

	Angle	Position
P	4 (A_p)	17 (P_p)
D	50 (A_d)	68 (P_d)

입력 값이 모터에 전달되고 모터의 엔코더를 통해 이동거리와 속력 값을 얻어낼 수 있다. 이 값들이 서로 상쇄되면서 로봇의 균형점을 찾게 되고 그 위치에서 균형을 유지하고 서있게 된다. 그림 10에 제어 블록도가 나타나 있고, 표 2는 실제 사용된 이득값이다.

2.3.3 실험

실내 바닥에서 균형 실험을 하였다. ROBOTC를 이용한 NXT의 기본 제어주기는 4*ms* 이다. 자이로 센서의 경우 4*ms*를 사용하면 흔들림이 심해서 샘플링 타임을 100*ms*로 설정하였다. 그림 11은 로봇이 균형을 유지하는 실험이다. 로봇이 앞, 뒤로의 작은 움직임은 있지만 균형을 유지하며 잘 서는 것을 확인하였다.

그림 12는 0초에서 8초까지의 각도 값을 나타낸다. 각도가 잘 유지되는 것을 볼 수 있다. 바닥이 미끄러워 중간 중간 흔들림이 있지만 균형을 유지하고 서있는 것을 확인하였다.

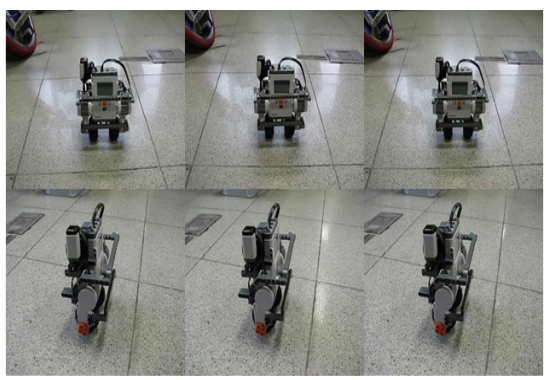

그림 11. 투휠 밸런싱 동영상
Fig. 11. Image cuts of Two-wheel balancing

2.4 두 바퀴 구동 밸런싱 라인트레이서 로봇

앞의 두 실험을 통해 학습한 라인트레이서와 투휠 밸런싱 기술을 접목하여 투 휠 밸런싱 라인트레이서를 제작하였다. 두 바퀴로 균형을 유지한 상태에서 선을 추종 할 수 있도록 제어하였다. 자이로 센서를 사용하여 균형을 유지하고 적외선 센서에 의해 선을 인식하

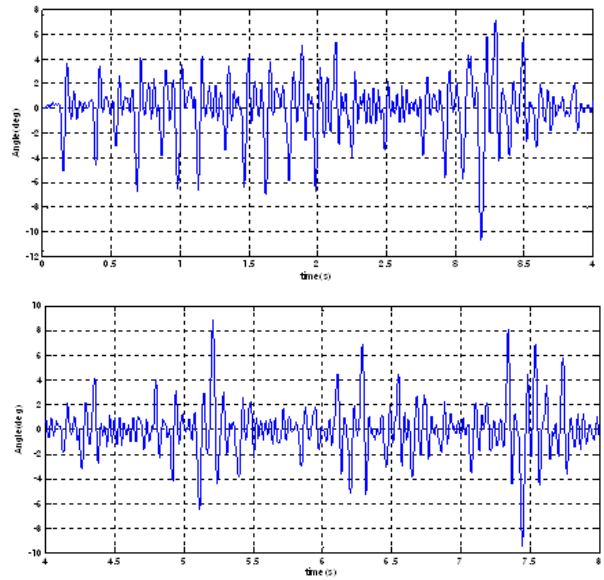

그림 12. 0초에서 8초 까지 자이로 센서 데이터
Fig. 12. Gyro sensor data for 0 to 8 seconds

여 선을 따라서 방향을 바꾸도록 하였다.

2.4.1 하드웨어

밸런싱 로봇에서 앞쪽에 적외선 센서만 설치해 주었다. 앞의 라인트레이서 제작과 마찬가지로 두 적외선 센서가 균형상태일 때 바닥과 최대한 밀착하도록 하고 균형을 유지하는데 방해가 되지 않도록 제작하였다. 그림 13에 밸런싱 라인트레이서가 나타나 있다.

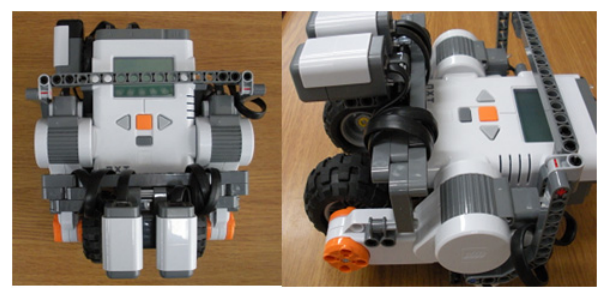

그림 13. NXT를 이용한 투휠 밸런싱 라인트레이서
Fig. 13. Two-wheel balancing line tracer using NXT

2.4.2 제어방법

적외선 센서를 추가하여 그림 14와 같이 사용하였다. 적외선 센서는 주변 빛에 의한 간섭이 심하므로 장소 또는 시간에 따라 값이 다르게 나올 수 있다. 따라서 트랙에 올려놓았을 때 초기 값이 설정될 수 있도록 동작 시 초기 3초 정도는 균형을 유지하고, 이후에

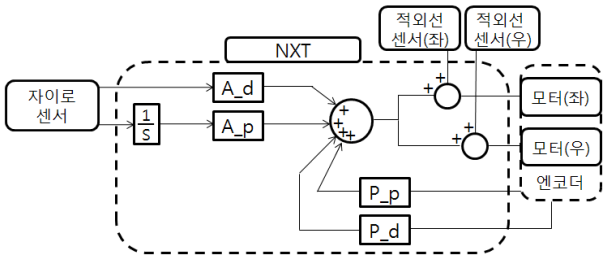

그림 14. 투휠 밸런싱 라인트레이서 제어 블록다이아그램
Fig. 14. Control block diagram for two-wheel balancing line tracer

는 라인트레이싱으로 전환하기 전 흰 바탕의 바닥 값을 표준 값으로 설정하도록 하였다.

라인트레이싱을 시작하면 임의의 기울기 값을 앞으로 주어 기울어 진 오차 값으로 로봇이 앞쪽으로 전진하게 하였다. 이를 위하여 오차 값을 주기적으로 적절히 0 과 -2를 넣어줘서 균형이 깨지지 않는 속도로 전진하도록 하였다.

또한 적외선 센서가 라인을 인식하면 회전해야 하는 방향으로 양쪽 모터에 반대 값(+8, -8)을 넣어줘서 균형이 잡힌 상태에서 회전을 할 수 있도록 하였다.

2.4.3 실험

그림 15는 실제 실험 동영상을 캡처한 그림이다. 로봇이 선을 추종하며 균형을 유지하는 것을 볼 수 있다.

그림 16에서 보면 자이로 센서값과 엔코더 값의 비교를 나타낸다. 자이로센서는 균형을 유지하면서 전진하기 때문에 균형만 유지할 때 보다는 떨림이 더 심한 것을 볼 수 있다. 흔들림이 있지만 '0'점을 기준으로 균형을 유지하고 있는 자이로 센서 데이터와 엔코더의

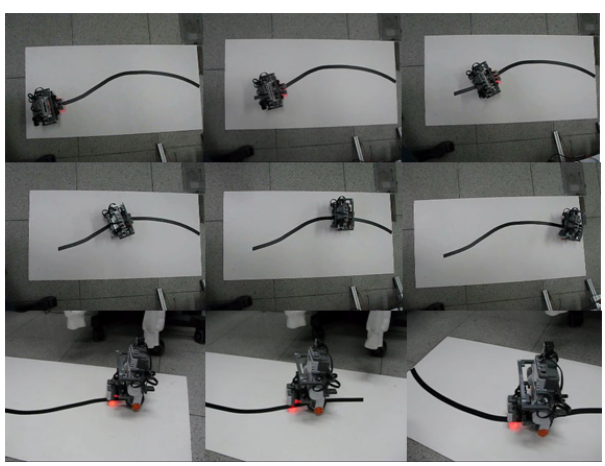

그림 15. 투휠 밸런싱 라인트레이서 동영상
Fig. 15. Image cuts of two-wheel balancing line tracer

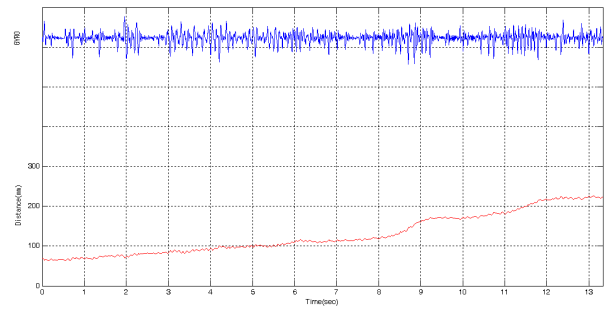

그림 16. 자이로 센서 값과 엔코더 데이터 비교
Fig. 16. Comparison between gyro and encoder data

증가하는 데이터를 통해 로봇이 균형을 유지하며 전진하고 있음을 알 수 있다.

3. 결 론

본 논문에서는 로봇 공학 교육의 하나인 이동로봇에 대한 실습으로 레고를 사용하는 실험방법을 소개하였다. 세 가지 이동 로봇 시스템을 개발하여 실험을 수행하여 교육 자료로서의 타당성을 검증하였다. 먼저 라인트레이서를 제작하고 밸런싱 로봇을 제작 한 다음 밸런싱 라인트레이서를 실습하도록 하여 실험의 연관성을 강조하였다.

실험을 통하여 적외선, 초음파, 자이로 센서의 원리와 사용방법을 익힐 수 있었고, 모터의 엔코더 값을 통한 거리 및 속도 제어를 할 수 있었다. 레고 NXT를 사용하여 하드웨어 구성을 쉽게 할 수 있었고, 프로그래밍 또한 쉽게 할 수 있어서 여러 가지 센서와 모터의 제어, 여러 형태의 로봇을 제작 할 수 있었다. 본 실습 방법은 로봇 공학 실습에 필요한 센서의 신호처리와 모터제어, 제어기의 값 설정 등 여러 가지 내용을 짧은 기간 동안 익힐 수 있는 교육용 자료가 된다.

참고문헌

[1] K. S. FU, R. C. Gonzales, C. S. G. Lee, "ROBOTICS", *McGraw-Hill*, 1987

[2] LEGO, "http://lego.com".

[3] 홍기천, “레고 NXT 로봇을 활용한 예비교사의 프로그래밍 언어 수업 방안 :미로 찾기 문제를 중심으로”, 정보교육학회논문지, 13권 1호, pp. 71-78, 2009.

[4] P. J. Gawthrop and E. McGookin, “A LEGO-based control experiment”, IEEE Control Systems Magazine, pp. 43-56, October 2004.
[5] 이강, “LEGO MINDSTORMS NXT를 이용한 이용한 공학설계 입문 운영사례”, 한국 공학교육 학회, 12권 2호, pp. 83-88, 2009.
[6] 박종진, 전창희, “USN과 LEGO MINDSTORMS NXT를 이용한 이동로봇의 위치인식과 주행시스템 개발”, 제어 로봇 시스템학회, 16권 3호, pp. 215-221, 2010.
[7] 김태희,강문설, “레고 마인드스톰 로봇을 이용한 프로그램 입문 교육의 효과측정”, 한국인터넷정보학회, 11권 4호, pp. 159-173, 2010.
[8] 웅홍우, 소원호, “가상 로봇 교육 시스템 설계 및 구현”, 전자공학회논문지, 48권 CI편 1호, pp. 108-115, 2011.
[9] 이경희, “ROBOTC 기반 LEGO MINDSTORMS NXT 로봇을 이용한 교육과정 개발 및 교육 효과 분석”, 정보처리 학회 논문지, 18 A권 5호, pp. 165-176, 2011.
[10] S. H. Kim and J. W. Jeon, “Introduction for freshmen to embedded systems using LEGO mindstorms”, *IEEE Trans. on Education*, vol. 52, no.1, pp. 99-108, 2009.

박 준 형

2009 충남대학교 메카트로닉스공학과(공학사)
2012 충남대학교 메카트로닉스공학과 지능로봇시스템전공(석사)
2012～현재 현대 위아 의왕 연구소
관심분야 : 지능로봇 시스템, 이동로봇, 밸런싱 메커니즘 응용
E-mail : vij232@hyundai-wia.com

정 슬

1988 미국 웨인 주립대학교 전기컴퓨터공학과
1996 미국 캘리포니아 대학 데이비스 전기컴퓨터공학과 석사, 박사
1997～현재 충남대학교 메카트로닉스공학과 교수
관심분야 : 지능 로봇 시스템, 가정용 서비스 로봇 응용, 교육 로봇 개발
홈페이지 : http:// isee.cnu.ac.kr
E-mail : jungs@cnu.ac.kr