

라인 트레이서 로봇의 주행 알고리즘에 관한 연구

이석원, 최완호, 송인근
우송대학교 컴퓨터전자정보공학부

A Study on the Driving Algorithm for Line-Tracer Robot

Seok-Won Lee, Wan-Ho Choi, In-Keun Song
School of Computer, Electronic and Communication Engineering, Woosong University

Abstract - In this paper, we propose an algorithm for driving line-tracer robot, considered as a kind of AGV. At first, we introduce the architecture of the line-tracer robot implemented for the experiment, and explain the driving algorithm using path information from infrared sensors. The proposed algorithm can overcome the difference between each sensor's characteristics. Finally, we make some experiments and show the performance of the algorithm in case of straight path.

2. 라인 트레이서 로봇의 구성 요소

일반적으로 라인 트레이서 로봇은 크게 기구부와 제어부로 나누어 볼 수 있으며, 특히 제어부는 그림 1과 같이 주제어를 담당하는 CPU(Central Processing Unit)를 중심으로 한 4가지 부분으로 구성되어 있다. 우선 주행의 기준인 라인과 로봇의 상대적인 위치를 파악하기 위한 센서부가 있어야 하고, 센서부로부터 입력된 정보를 바탕으로 로봇의 자세를 보정할 수 있는 모터의 구동 명령을 계산하기 위한 알고리즘 수행부가 필요하다. 알고리즘 수행 결과 출력된 구동 명령에 따라 로봇을 움직일 수 있는 구동부와, 전체 시스템을 동작시키기 위한 전원부가 추가로 필요하다.

1. 서 론

자율적으로 진행 방향을 변경할 수 있는 조향 기능을 갖고 지정된 경로를 지능적으로 주행할 수 있는 무인 반송차(AGV : Automatic Guided Vehicle)는 유도 경로의 형태에 따라 다음과 같이 분류할 수 있다. 지정된 경로를 연속적으로 추종하는 고정 경로식, 주행 경로 상에 표식을 적당한 위치에 설치하고 AGV가 이를 검출하여 주행하는 반경로식, 유도선이나 유도설비 등 외부의 지원 없이 차체에 설치된 바퀴의 엔코더나 비전 시스템 등으로 위치 측정을 하면서 스스로 주행 경로를 생성하는 자율 주행식으로 나눌 수 있다[1][2]. 이런 분류 중에 고정 경로식의 형태를 갖춘 로봇 중 하나가 라인 트레이서이다.

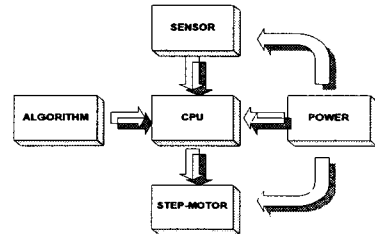


그림 1. 라인 트레이서 로봇의 구성도

라인 트레이서의 기본적인 목표는 주어진 주행선을 센서로 검출하여 목적 위치까지 정확하고 빠르게 이동하는 것이다. 라인 트레이서 대회는 현재 여러 대학교에서 매년 전국 학생과 일반인을 대상으로 개최되어지고 있으며 현재 대회의 수와 참가 인원수도 증가하고 있다. 라인 트레이서 대회는 고정된 라인을 가장 빨리 완주하는 로봇을 겨루는 속도 경기이기 때문에 하드웨어적인 설계와 소프트웨어의 차이에 따라 주행 시간의 차이가 많이 생긴다. 하드웨어적인 특성은 대부분 공개되고 대회를 참가함으로써 정보 교류를 통해 대부분 비슷하나 소프트웨어의 차이에 의해 주행 성능이 달라진다. 제어부는 하드웨어적인 보정 방식을 사용하지 않으면서 복잡한 알고리즘을 수행하기 위해 점차 고성능의 CPU가 사용되어지고 있다. 초기 라인 트레이서의 센서부는 하드웨어적으로 프로그램이 복잡하지 않은 op-amp를 이용한 ON/OFF 형태의 날개형 방식을 많이 사용하였다. 그러나, 시간이 지나면서 대부분 A/D 변환기를 사용한 광량 방식의 라인 트레이서가 주를 이루고 있다. 이러한 A/D 변환기의 사용으로 센서의 광량 데이터 처리와 주행 알고리즘의 구현이 복잡해지면서, 라인 트레이서의 주행 알고리즘에 대한 체계적인 연구가 필요하다. 본 논문에서는 라인 트레이서의 적외선 센서 특성 차로 인한 오차 보상 및 주행 알고리즘에 대해 설명하고자 한다.

2.1 센서 구동 방식 및 배치

라인 트레이서 로봇은 검정색 주행판 위에 백색 테이프가 목표 주행 경로로써 부착되어 있는 주행 환경을 가지고 있고, 빠른 시간 내에 주행 경로를 완주하고자 하는 특별한 목표가 있기 때문에 보다 저렴하고 구현이 용이한 센서를 사용하는 것이 가능하다. 그러므로, 본 연구에서는 라인의 위치를 검출하기 위한 센서로 적외선을 이용한 광센서를 채용하였고, 센서 구동 방식으로는 송신부에 전격 이상의 전류(1~2[A])를 짧은 주기(≈1[μsec.])를 갖는 펄스열의 형태로 흘려주어 송신부의 출력을 크게 하는 대신 수신부의 감도를 낮게 하여 상대적으로 외란광의 영향을 줄일 수 있는 펄스 점동(Pulse Drive) 방식을 사용하였다. 센서 값 입력방식으로는 발광부와 수광부 사이의 장애물 유무만을 판단하는 디지털 방식이 아니고, 물체에서 반사되는 빛의 양을 측정하여 센서와 물체 사이의 거리를 구하는 광량 방식을 사용하였다. 라인 트레이서의 센서의 배치는 그림 2와 같이 주행 라인을 중심으로 보정용 센서 L0와 R0가 위치한다. L0와 R0의 위치가 주행 라인에 너무 가까우면 보정을 자주 수행하여 주행 중 흔들림 현상이 생기고, 너무 멀면 보정의 범위를 넘기 때문에 보정이 늦게 수행되어 원활한 주행을 하지 못한다. 그러므로, 센서의 배치는 사용하는 센서의 종류에 따라 반사각의 범위를 고려하여 결정해야 한다. 본 연구에서는 이런 점을 고려하여 보정 센서 사이의 거리를 약 25[mm]로 설정하였다.

2.2 구동부

본 연구에서는 속도나 위치 정보 등이 없이 개루프 제어 동작이 가능한 스테핑 모터를 사용하였다. 두 개의

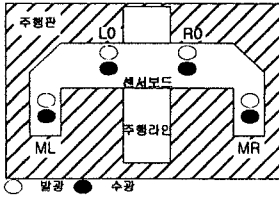


그림 2. 센서 배치도

모터 축을 좌우 바퀴에 연결하여 좌우 바퀴의 회전 방향과 속도 차이로 직선 운동과 곡률 회전 등을 가능하게 할 수 있다. 그리고, 1-2상 여자 방식을 사용하였으며, 모터 권선에 전류를 흘려 줄 드라이버 회로는 전용의 유니폴러 정전류 구동 드라이버 SLA7026을 사용하였다. 정전류를 위한 PWM 회로가 2개 내장되어 있고 파워 FET의 사용으로 고속 응답성이 우수하며, 최대 3[A]까지 전류를 흘려 줄 수 있다. 정전류 특성을 얻기 위해서 모터 정격 전압보다 3~5배 높은 전압을 인가하는데 최대값은 50[V]이다[3]. 토크 조절을 위해 전류 조절용 가변 저항을 배치하였으며, 모터는 JAPAN SERVO사의 2상 스테핑 모터 KH42JM2-901을 사용하였다.

2.3 주제어 CPU를 포함한 제어부

80C296SA는 32개의 I/O 포트와 19개의 인터럽트 소스, 4개의 외부 인터럽트 핀이 있고, 2개의 16비트 타이머와 3개의 PWM, 6개의 칩 선택 장치, 입력 캡처나 출력 비교를 위한 Event Processor Array(EPA)가 4개 있다[4]. 라인 트레이서의 모터부에 2개의 EPA 칩페어 인터럽트를 사용하여 타이머 값을 조절하여 모터의 속도를 제어하였다. 80C296SA 내부에 광센서 입력신호 처리를 위한 A/D 변환기가 없기 때문에 외부에 MAXIM사의 MX7828이라는 고성능 A/D Converter를 사용하였다.

2.4 전원부

CPU를 포함한 IC들은 어느 정도 일정한 전류를 소모하는 반면, 모터부에서는 순간적으로 1~2[A]의 큰 전류를 필요로 한다. 그리고, 센서부와 같은 아날로그 회로는 전원 노이즈에 상당히 민감하기 때문에 마이크로프로세서와 로직용 IC에는 +5V의 정전압이 필요하므로, 높은 입력 전압을 낮추어 정전압원을 만드는 전용 스위칭 방식의 레귤레이터 LM2575를 사용하였다[3].

3. 적외선 센서의 광량 오차 보상 알고리즘

3.1 적외선 센서의 튜닝

그림 3에서 R0, L0는 자세 보정용 센서이고 MR, ML은 턴 마크용 센서이다. 본 절에서는 자세 보정용 센서인 R0, L0의 수광부에 입사되는 광량의 오차를 보정하기 위한 알고리즘에 사용되는 기준 광량 값을 획득하는 광량 센서의 튜닝 방법을 소개한다. 그림 3과 같이 주행 라인의 주행 방향과 직각 방향으로 라인 트레이서의 센서 보드를 위치시키고 상하로 움직이면서 입력되는 센서값을 바탕으로 센서값 중 최소값과 최대값을 선정 후 최소값과 최대값의 평균값을 계산한다. 광량 오차 보상 알고리즘에서 사용하기 위하여 센서의 최소값과 최대값 사이를 다수 구간으로 분할하는 광량값의 경계치를 계산. 저장함으로써 적외선 센서의 튜닝 과정이 완료된다.

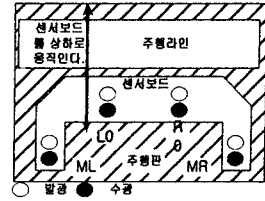


그림 3. 센서 튜닝 방법

3.2 광량 오차 보상 알고리즘

주행 라인을 중심으로 라인 트레이서를 대칭적으로 위치시켰다 할지라도 오른쪽 센서(R0)와 왼쪽 센서(L0)의 특성의 차이와 외부 환경의 영향으로 인해 두 센서로부터 동일한 광량값을 얻을 수 없다. 이러한 상대적 오차를 갖고 있는 두 센서의 광량값을 보정 과정 없이 그대로 라인 트레이서의 자세 보정 및 주행 방향 선정에 사용한다면 동일한 환경에서 곡선 주로의 턴 방향에 관계없이 동일한 보정을 수행하는 것이 어려워진다. 그러므로, 각 센서의 광량값의 오차를 보정한 후 자세 보정에 사용하여 정확한 주행 제어를 하기 위한 알고리즘이 필요하다.

본 연구에서는 광량 대 거리 테이블이라는 광량 마이크로 마우스에서 사용하는 알고리즘을 응용하였다. 광량 마우스의 경우 센서와 장애물까지 거리가 가변적이지만 라인 트레이서의 경우 센서와 주행판까지의 거리는 고정되어 있고, 주행판의 명암에 따라서 광량값이 변화한다. 검은색 주행판에서 반사된 광량값이 최소값을 구성하고 하얀색 주행 라인에서 반사된 광량값이 최대값을 구성한다. 이러한 라인 트레이서의 경우 그래프는 그림 4와 같이 나타낼 수 있다. 이 최대값과 최소값 사이를 그림 5와 같이 여섯 구간으로 분할하여 보정 경계치를 설정한다. 그림 5에서 C0는 광량의 최소값, C6는 광량의 최대값이고, C1은 C0와 C6의 평균값이다.

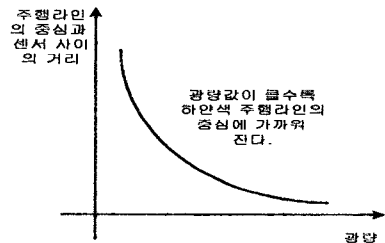


그림 4. 주행라인의 중심과 센서 사이의 거리 대 광량

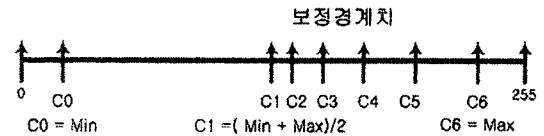


그림 5. 광량의 보정 경계치 설정

C2~C5를 선정할 때 보정 경계치와 보정 경계치 사이는 다음 식 (1)과 같은 관계를 가질 수 있도록 설정한다.

$$(C2-C1) \times (C3-C2) \times (C4-C3) \times (C5-C4) \times (C6-C5) \quad (1)$$

최소값 C0와 평균값 C1 사이는 한 구간으로 설정하는데 그 이유는 이 구간의 광량 값이 입력되는 경우는 현

재 센서 반사면을 검정색 주행판으로 간주할 수 있어 보정이 불필요하기 때문이다. 그리고, 센서의 위치가 하얀색 주행 라인의 중심 방향으로 접근하여 보정이 필요한 C2~C6 구간은 센서의 비선형적 특성을 반영하기 위하여 식 (1)과 같이 비선형적으로 분할한다. 이렇게 보정 경계치를 설정함으로써 두 광량 센서의 상대적인 광량 오차를 최소화할 수 있다.

3.3 주행 알고리즘

앞에서 설명한 보정 경계치 설정이 끝나면 각 센서로부터 입력된 광량을 바탕으로 모터 속도의 보정량을 결정해 주어야 한다. 이를 위해서 본 연구에서는 모터 보정을 위한 핸들 테이블용 포인터를 식 (2)를 이용하여 생성한다.

$$L_TRIM[i] = \frac{i - C_k}{C_{k+1} - C_k} + (TRIM * k) \quad \text{단, } k=0 \sim 5 \quad (2)$$

위 식 (2)에서 파라미터 TRIM에 의해서 좌우 바퀴의 핸들 테이블 포인터가 결정되어지며, 핸들 테이블의 데이터가 속도 테이블의 데이터와 곱해져 좌우 바퀴의 속도를 결정하게 된다. 그러므로, TRIM을 너무 크게 설정하면 보정량이 지나치게 커져 트래이서 운동이 부드럽지 못하게 되고, TRIM 값이 너무 작으면 반대로 충분한 보정이 이루어지지 못한다. 그림 6에서 수행 알고리즘의 핵심 코드를 나타내었다.

```

LSEN[0]~LSEN[5] := C0~C6
TRIM : 보정 경계값

for(i=0:(LSEN[0]:i++)) LEFT_TRIM(i) = 0;
for(i=0:(RSEN[0]:i++)) RIGHT_TRIM(i) = 0;
-----광량값이 평균값 이하면 0-----
for(i=LSEN[5]:i=256:i++) LEFT_TRIM(i) = TRIM_LIMIT;
for(i=RSEN[5]:i=256:i++) RIGHT_TRIM(i) = TRIM_LIMIT;
-----광량값이 최대값 이상이면 경계치의 최대값-----
for(j=0:(5:j++)) {
for(i=LSEN[j]:(LSEN[j+1]:i++))
LEFT_ANGLE(i) = ((ANGLE*i)/(LSEN[j+1]-LSEN[j]))
-(ANGLE*LSEN[j])/(LSEN[j+1] - LSEN[j]) + (j*ANGLE);

for(i=RSEN[j]:(RSEN[j+1]:i++))
RIGHT_ANGLE(i) = ((ANGLE*i)/(RSEN[j+1]-RSEN[j]))
-(ANGLE*RSEN[j])/(RSEN[j+1] - RSEN[j]) + (j*ANGLE);
}
-----보정 경계치를 광량의 변화량으로 나눈 부분.-----

```

그림 6. 주행 알고리즘 코드

4. 실험 결과 및 분석

4.1 실험 환경

- 센서 종류: ST-8L(수광), EL-8L(발광) 적외선 센서 2조
- 주행판 재질: 검은색 호미카판, 크기 = 1200(mm)×1200(mm)
- 주행 라인: 하얀색 3M 바닥 경계용 테이프, 폭 = 20(mm)
- 주행 라인 설명: 주행 라인 그림 7 참조, 직진 1구간 길이 = 310(mm)

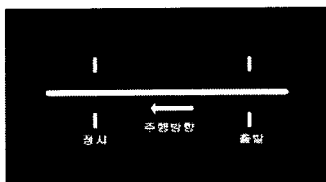


그림 7. 실험 주행판

4.2 실험 방법

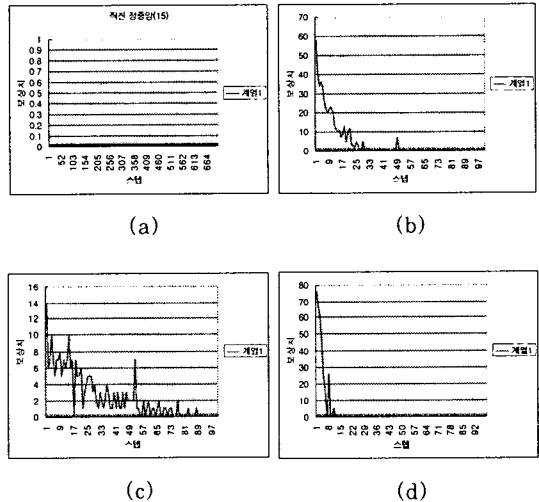
- ① 라인 트래이서의 R0, L0 센서 튜닝후 광량 오차 보

상 알고리즘을 수행시킨다.

- ② TRIM 값을 15, 속도 포인터는 300으로 설정한다.
- ③ 라인 트래이서를 출발 지점에 놓은 후 약 310(mm) 길이의 직진구간을 주행하도록 한다.
- ④ 총 4회에 걸쳐 테스트한다.
- ⑤ 1차 테스트는 R0, L0 센서의 광량값을 똑같이 한 후 (트래이서가 들어짐 없이) 출발시킨다.
- ⑥ 2차 테스트는 R0, L0센서의 광량값을 틀리게 한 후 (왼쪽으로 들어짐) 출발시킨다.
- ⑦ 3차 테스트는 TRIM 값을 25, 속도 포인터는 300으로 설정하고 실험 방법 ⑥과 같이 출발시킨다.
- ⑧ 4차 테스트는 TRIM 값을 5, 속도 포인터는 300으로 설정하고 실험 방법 ⑥과 같이 출발시킨다.
- ⑨ 주행 완료후 매 스텝마다 광량값 변화에 따른 광량 오차 보상 데이터를 저장한다.

4.3 실험 결과

아래에서의 실험 결과 그래프를 보면 들어짐 없이 출발할 경우 보상 없이 직진 주행을 하는 것을 볼 수 있다. 그리고, 왼쪽으로 치우쳐 출발할 경우 TRIM 값이 큰 경우 TRIM 값이 작은 경우보다 빠른 시간에 보상은 안정적임 직진 주행을 시작함을 확인할 수 있다.



- (a) 직선 주로를 들어짐 없이 출발할 경우(TRIM = 15)
- (b) 직선 주로를 왼쪽으로 치우쳐 출발할 경우(TRIM = 15)
- (c) 직선 주로를 왼쪽으로 치우쳐 출발할 경우(TRIM = 25)
- (d) 직선 주로를 왼쪽으로 치우쳐 출발할 경우(TRIM = 5)

그림 8. 실험 결과 그래프

5. 결론

본 논문에서는 적외선 센서를 이용한 라인 트래이서 로봇의 구현에 대하여 설명하고, 좌, 우 센서의 특성 및 환경차를 보상할 수 있는 주행 알고리즘을 제안하였다. 그리고, 구현한 라인 트래이서 로봇을 대상으로 직접 주행 실험을 수행하여 제안한 알고리즘의 성능 및 파라미터가 성능에 미치는 영향을 분석하였다.

(참고 문헌)

- [1] A.J.Bostel and V.K.Sagar, "Dynamic control systems", Computing & Control Engineering Journal, Jun
- [2] 임일선, 허옥열, "뉴로퍼지방식 광유도식 무인반송차의 경로 추종 제어", Journal of Control Automation and Systems Engineering, Vol. 7, No.9, September 2001.
- [3] 윤지영, "마이크로 로봇 바이블", 성안당, 2002.
- [4] "80C296SA User manual", www.intel.com