

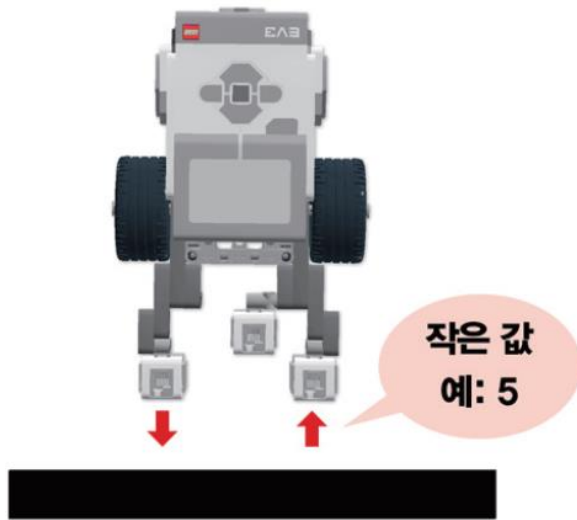
MindStorm EV3

(Line trace)

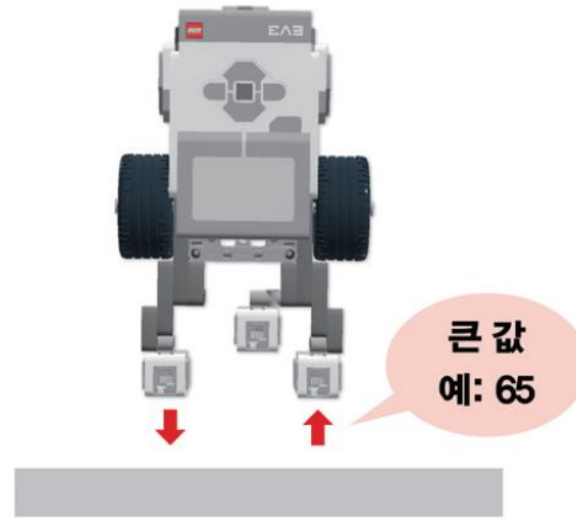
이건희, 조한진

1.1 Linetrace: 라인감지 원리1

- 어두운 바닥면 – 빛 많이 흡수, 빛 반사값 작게 측정
- 밝은 바닥면 – 빛 적게 흡수, 빛 반사값 크게 측정



어두운 바닥면



밝은 바닥면

1.2 Linetrace: 라인감지 원리2

- 센서의 상태, 조명의 상태, 로봇이 움직이는 속도 등으로 인한 오차 발생 가능
- 라인과 흰색 바탕면 경계 근처에서 애매한 값 측정
→ 바닥과 라인의 구별 시 어느 정도의 오차 범위 고려 필요
경계값 : 판단의 범위의 경계에 있는 값
- 항상 하드웨어나 주변 환경 등의 요인을 같게 유지할 수 없음
→ 로봇이 실행될 때마다 새로 경계값을 정하는 것이 바람직
- 1센서 라인 트레이싱을 시작할 때 경계값 자동 계산 함수 호출
→ 현재 센서 값을 통해 자동으로 경계값 결정
※ 로봇 출발 시 컬러 센서(c2)가 검정색 라인에 위치해야 함

$$\text{경계값} = (\text{바닥측정값} + \text{라인측정값}) / 2$$

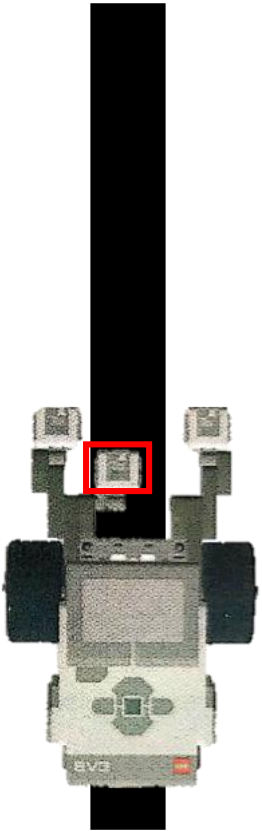
2. On/Off control

2.1 One sensor boundary calculate

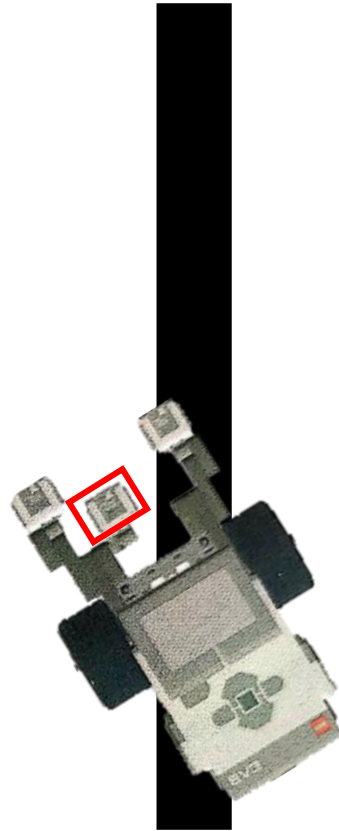
On/Off 라인트레이싱 원리: 검은선이 검출되면 우회전, 흰선이 검출되면 좌회전을 한다.

①, ② 구현하기

검은 선 값 계측

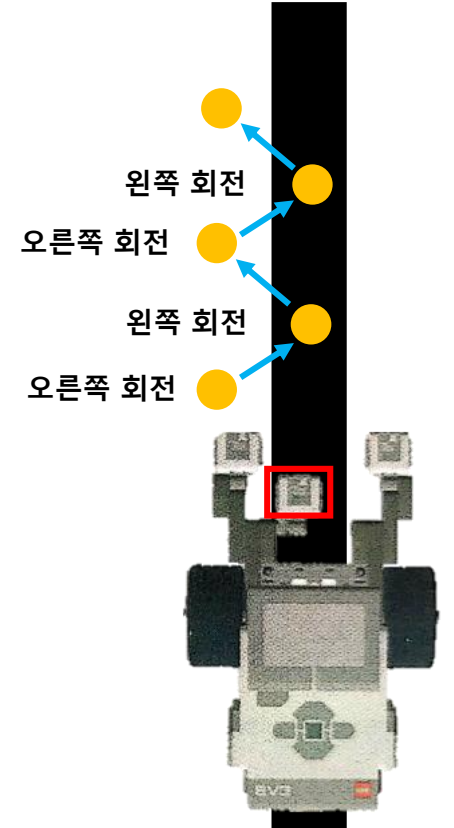


흰 선 값 계측



경계 값 계산

한쪽 경계선 따라 이동



①

②

2. On/Off control

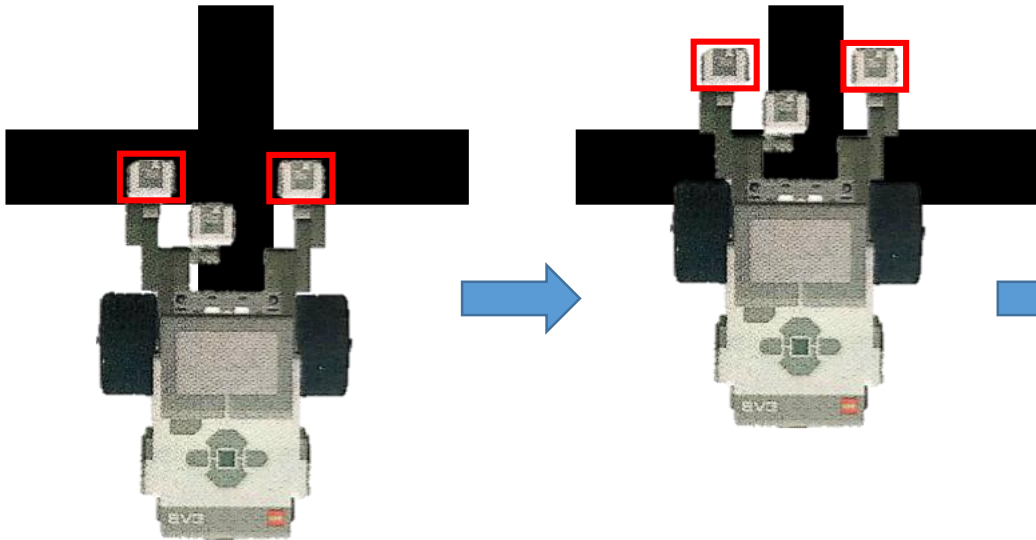
2.3 Two sensor boundary calculate

①, ② 구현하기

On/Off 라인트레이싱 원리: 검은선이 검출되는 방향으로 회전한다. 둘 다 흰색이면 직진

검은 선 값 계측

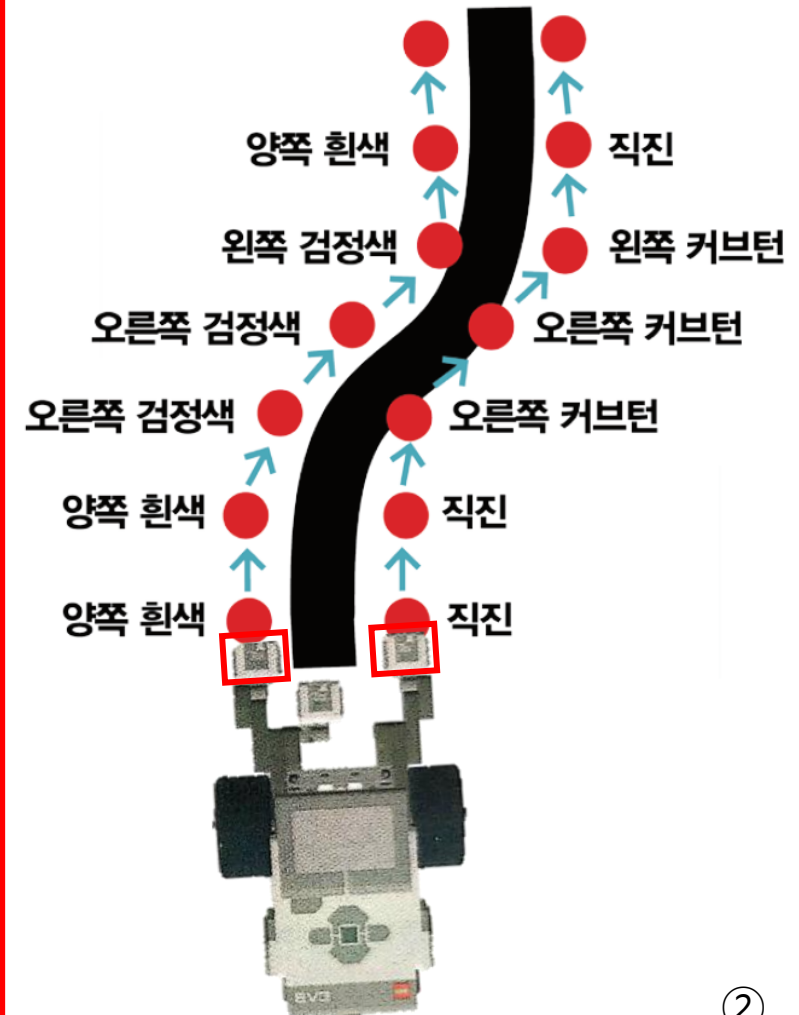
흰 선 값 계측



경계 값 계산

①

양쪽 경계선 따라 이동



②

3. PID control

3.1 P(Proportional) Control Linetrace

P Control Linetrace 기본 원리

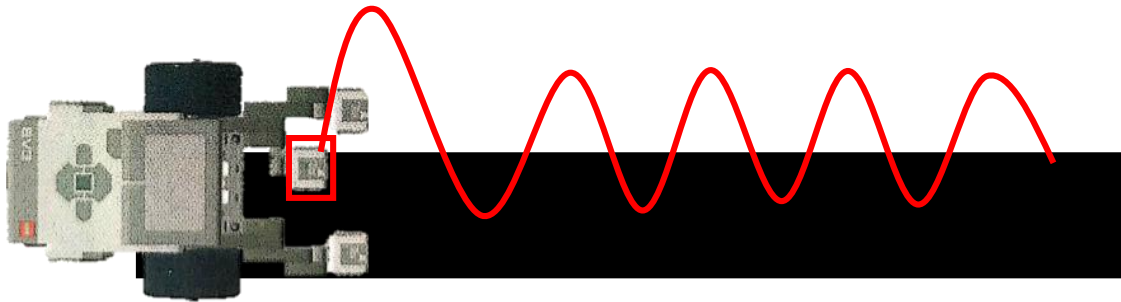
- On/Off 제어처럼 검은선이 검출되면 우회전, 흰선이 검출되면 좌회전을 한다.
- 우회전, 좌회전 속도를 에러 크기에 따라 다르게 한다.

→ 빠르게 경계선에 접근 가능

→ 경계선에 가까워지면 속도가 줄어서 관성에 의해 에러가 커지는 것을 방지

→ 오차가 누적되어 정상상태(Steady-state)에 가도 완전한 수렴이 되지 않는다.

→ 실습 해보기



3. PID control

3.2 PI (Proportional Integral) Control Linetrace

PI Control Linetrace 기본 원리

- On/Off 제어처럼 검은선이 검출되면 우회전, 흰선이 검출되면 좌회전을 한다.
- 우회전, 좌회전 속도를 에러 크기에 따라 다르게 한다.
- 누적 오차에 의한 영향도 제어 인자에 반영(오차를 적분하여 더해준다.)

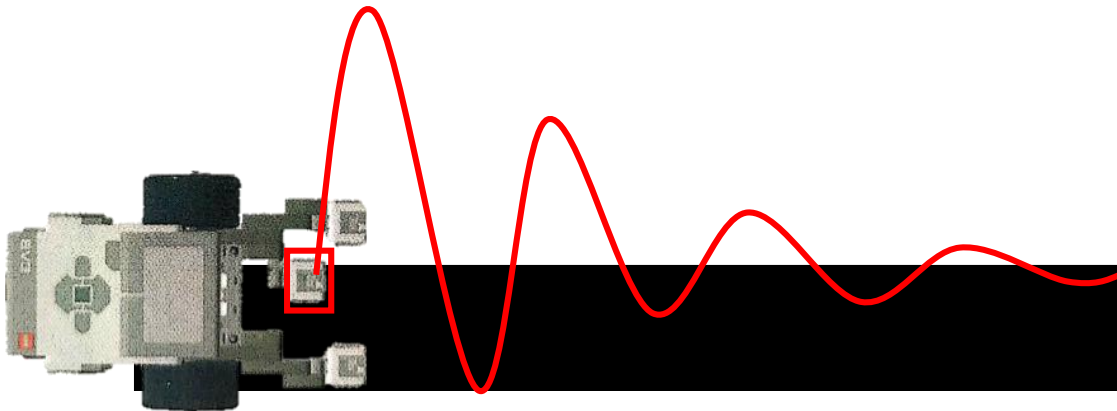
→ 빠르게 경계선에 접근 가능

→ 경계선에 가까워지면 속도가 줄어서 관성에 의해 에러가 커지는 것을 방지

→ 정상상태(Steady-state)에 이르러도 잔류 오차가 적다.

→ 제어 시작 부근에서 오버슈트가 크게 발생하고 그에 따라서는 라인 이탈이 발생할 수 있다.

→ 실습 해보기



3. PID control

3.3 PID (Proportional Integral Differential) Control Linetrace

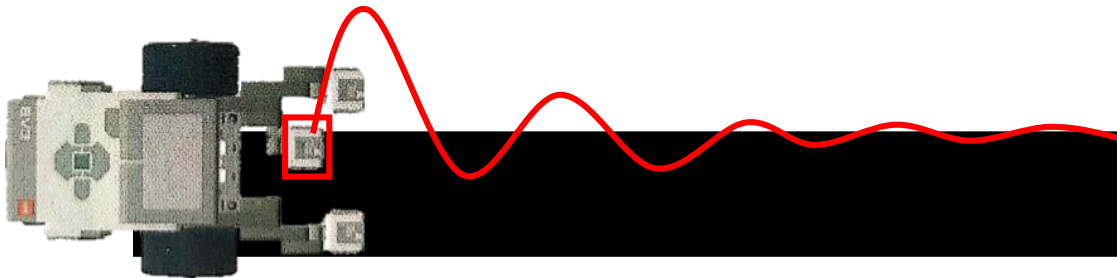
PID Control Linetrace 기본 원리

- On/Off 제어처럼 검은선이 검출되면 우회전, 흰선이 검출되면 좌회전을 한다.
- 우회전, 좌회전 속도를 에러 크기에 따라 다르게 한다.
- 누적 오차에 의한 영향도 제어 인자에 반영(오차를 적분하여 더해준다.)
- 오차의 시간에 따른 변화율을 고려한다. (오차를 미분하여 더해준다.)

*지글러-니콜스 기법으로 I값을 구하는 것도 추가 수행 ($K_i = \frac{K_p^2}{4 \times K_d}$)

- 빠르게 경계선에 접근 가능
- 경계선에 가까워지면 속도가 줄어서 관성에 의해 에러가 커지는 것을 방지
- 정상상태(Steady-state)에 이르러도 잔류 오차가 적다.
- 제어 시작 부근에서 오버슈트가 크게 발생하고 그에 따라서는 라인 이탈이 발생할 수 있다.

→ 실습 해보기



3. PID control

3.3 PID Control Linetrace

PID 제어 정리

	P 제어	PI 제어	PID 제어
장점	<ul style="list-style-type: none">- 목표 값에 근접- 동작이 부드러움 (On/Off 대비)	<ul style="list-style-type: none">- 목표 값과 일치- 잔류 오차 제거	<ul style="list-style-type: none">- 목표값과 일치- 오버슈트 제거
단점	<ul style="list-style-type: none">- 목표 값과 불일치- 잔류 오차 발생	<ul style="list-style-type: none">- 안정 상태까지 오래 걸림- 오버슈트 발생	<ul style="list-style-type: none">- K_p, K_i, K_d를 적절히 맞추기 어려움

PID 각 계수 크기 변화에 따른 경향성
* 절대적인 것은 아닙니다. 시스템에 따라 달라질 수 있음

	K_p	K_i	K_d
크다	<ul style="list-style-type: none">- 속도의 변화가 큼- 진폭이 커짐(불안정)	<ul style="list-style-type: none">- 잔차 제거- 오버슈트 발생	<ul style="list-style-type: none">- 오버슈트 제거- 진폭이 커짐(불안정)
작다	<ul style="list-style-type: none">- 속도 변화가 작음- 이탈 가능성이 높음	<ul style="list-style-type: none">- 잔차 제거 실패	<ul style="list-style-type: none">- 오버슈트 제거 실패