

ICT-DNA 팀설계 프로젝트1

자율주행 2팀



담당 교수	조정훈 교수님	경북대 전자공학부	
참여 학생	이재훈	영남대 정보통신공학과	B202300163
	김영균	경북대 전자공학부	2017111243
	박종민	영남대 로봇공학과	B202300176
	임경준	경북대 전자공학부	2019114837
	최현석	경북대 전자공학부	2021427976

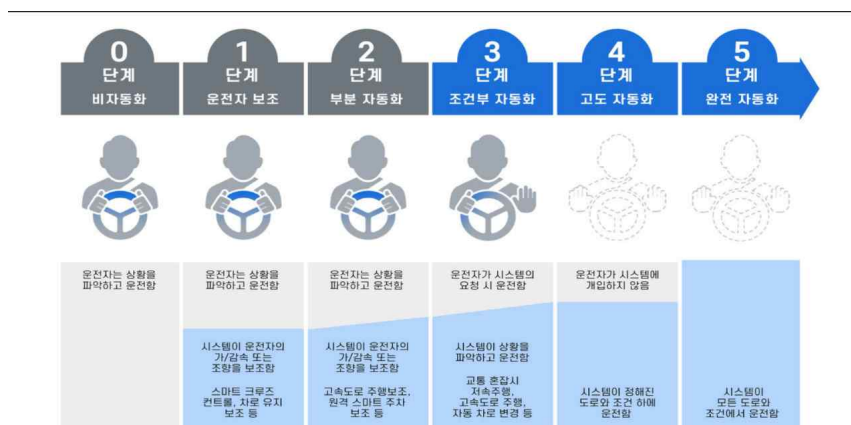
■ 과제 선정 배경

자율주행 기술은 빠르게 발전하고 있으며, 자동차 산업뿐만 아니라 로봇, 드론, 물류 및 운송 등 다양한 산업 분야에도 확장되고 있다. 이에 따라 자율주행 시스템을 이해하고 구현하는 능력은 미래의 기술 분야에서 필수적이다.

실제 자율주행차를 구현하긴 문제들이 따르기에, 자율주행차에서 사용되는 센서들(LiDAR, Camera, IMU, IR Depth Cameras)을 1/10 Scale 플랫폼에서 구현하여 일련의 감지(Sense), 인지(Perception), 계획(Path Planning), 동작(Control) 과정을 직접 구현해 본다.

구현하고자 하는 성능에 맞는 하드웨어 재료를 선정하고, 자율주행에 필요한 여러 알고리즘들을 목적에 맞추어 설계 및 조정한다. PID를 이용한 모터제어 부터 시작해 IMU센서 및 모터 엔코더를 이용한 바퀴 회전 수 측정을 통해 위치를 추적한다. SLAM을 통해 LIDAR 기반 Map을 작성하고, Stereo vision을 위한 카메라 센서를 사용하여 센서퓨전, 딥러닝을 통한 객체 탐지를 통해 주행과정에서 변수에 대응하도록 한다.

이를 통해 하드웨어 선정부터 소프트웨어 개발, 차량 제어, 주행 테스트, 성능 개선까지의 Lv.4 수준의 자율주행 프로세스 전 과정을 습득해본다.



■ 프로젝트 주요 목표

0. HW재료 선정 (Lidar, Stereo Camera, Controller, GPU, IMU, Joy Stick)
1. 모터 제어(pid 제어)(드라이버 찾기)
2. 위치 추적(IMU 사용, Encoder Motor)
3. 센서 데이터 송수신, 처리 및 퓨전
(IMU에서 발생하는 드리프트 오차 수정, 베이지 필터, 칼만 필터)
4. 지도 Mapping(SLAM)
5. 영상 처리(Stereo Camera, OpenCV)
6. 최적 경로 (벽 따라가기 → A* 성능향상, Pure Pursuit)
7. 딥러닝1(객체 탐지)

■ 프로젝트 상세 추진계획

3월

- HW 재료 선정 + 조립
- 노트북, 자동차 Ros 환경 세팅
- Ros 사용법 숙지(노드 구성, 통신방법, 시뮬레이션 등)

4월

- 모터제어 알고리즘 설계(pid 제어, 속도, 방향 등)
parameter 조정을 안정적 주행 구현
- 위치 추적 알고리즘 설계(imu, 모터 엔코더 이용)

imu 드리프트 오차 잡기, 바퀴 수를 통한 위치 추적
- Slam 으로 mapping

정확하게 mapping 하는 방법 찾기
- 센서 처리(IMU에서 발생하는 드리프트 오차 수정)

5월

- 영상 처리(SLAM 따기, stereo vision,)
- 최적 경로 알고리즘 설계

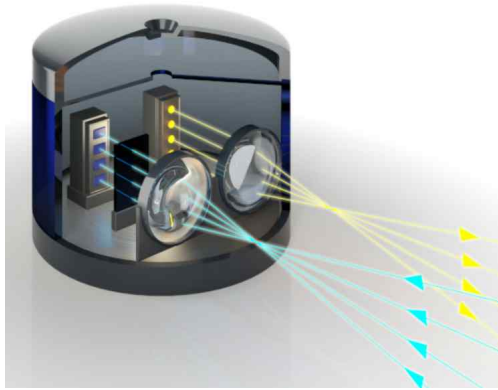
A-star algorithm 을 통한 최적 경로 추정
- 센서 퓨전(베이지스, 칼만 필터)

6월

- 주행대회(최단시간)

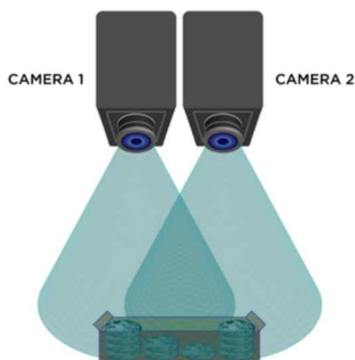
■ HardWare 재료 선정

1. Lidar



- **정의:**
레이저 광선을 사용하여 주변 환경의 거리와 방향을 측정하는 센서
- **이용 목적:**
지도 작성: SLAM으로 주변 환경의 거리 데이터를 수집해 IT 2호관 지도 작성.
자율주행 자동차는 SLAM으로 만든 지도를 기반으로 자율주행.
거리측정을 통한 위치 추정: SLAM 기술과 결합하여 자율주행 자동차의 실시간 위치를 추정하고 이를 토대로 주행 경로를 동적으로 조정

2. Stereo Camera



- **정의:** 두개의 렌즈가 있는 카메라 시스템으로, 두 렌즈간의 거리를 이용하여 깊이 정보를 측정. 측정을 통하여 주변 환경의 공간 구조를 인식
- **이용 목적:**
장애물 회피: Stereo Camera로 획득한 정보를 기반으로 주행중 나타나는 장애물에 대한 실시간 정보를 제공하여 장애물이 맵핑이 안된 SLAM을 보완

8. Motor Controller (아두이노 or something



- **정의:** 아두이노(or else)를 이용하여 모터의 속도와 방향을 제어
- **이용 목적:**
Controller 분리: 실제 자동차에서도 John Controller와 I/O Control이 분리되어 시스템이 이루어진다. 이와 같이 모듈의 유연성과 성능 향상에 도움을 준다.
모터 제어: PID 제어를 통한 주행중 거리를 측정하여 중심에 유지하기 위한 제어, PWM제어를 통한 모터에 가해지는 펄스 폭을 조절하여 모터의 회전 속도 제어

9. Encoder Motor



- **정의:** 모터의 회전축에 부착된 디스크 또는 스케일이 회전하면서 빛을 차단하거나 반사하여 이러한 빛의 변화를 센서가 감지하여 모터의 회전상태를 펄스 신호로 출력, 이러한 펄스 신호를 분석하여 모터의 회전 속도와 회전 방향을 측정할 수 있는 모터

- **이용 목적:**

정확한 위치 제어 및 속도 제어: 현 프로젝트와 같이 실내에서 자율주행하는 환경의 경우 GPS 신호가 닿지 않고 IMU 센서의 드리프트 발산 오차가 커 정확한 위치를 파악하기 어렵기 때문에 자신의 위치 및 현재 속도를 정확하게 파악하기 위해서는 엔코더 모터는 필수적이다.

10. GPU (jetson xavier)



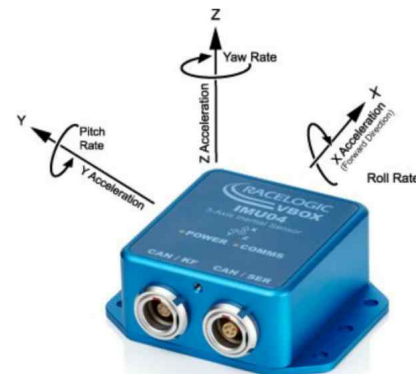
- **정의:** 병렬 처리 능력을 활용하여 복잡한 계산을 가속화하는데 사용되는 그래픽카드의 부품

- **이용 목적:**

이미지 처리: 카메라에서의 노이즈를 처리하고

답러닝:

11. IMU



- **정의:** 장치에 가속력과 진동력, 충격력 등 동적 힘이 발생하였을 때, 움직임의 변화에 따른 순간적인 가속도, 각속도의 변화를 감지하여 이를 토대로 장치의 현 위치를 계산하는 센서

- **이용 목적:**

위치 및 자세 측정: 가속도계와 자이로스코프를 사용하여 차량의 가속도, 회전 속도, 방향등을 측정하여 GPS 신호 없이도 차량의 자세 및 위치를 추정할 수 있다.

■ 구현 기술

ROS

Robot Operating System이며 오픈 소스 로봇 개발 플랫폼으로, 소프트웨어 개발에 필요한 여러 기능과 도구를 제공하는 프레임워크.

패키지 단위로 소프트웨어를 모듈화하고 패키지 간에 재사용이 용이하다. 또한 표준화된 메시지 통신 시스템을 제공하여 다양한 센서 및 액추에이터와의 통합이 용이하게 한다.

PID

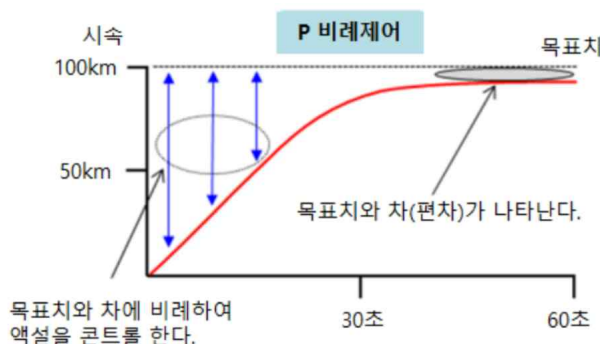
실제 응용 분야에서 가장 많이 사용되는 대표적인 형태의 제어기법이다. PID 제어기는 기본적으로 피드백 제어기의 형태를 가지고 있으며, 제어하고자 하는 대상의 출력값을 측정하여 이를 원하고자 설정값과 비교하여 오차를 계산하고, 이 오차값을 이용하여 제어에 필요한 제어값을 계산하는 구조로 되어 있습니다.

P 동작: Proportional (비례)

I 동작: Integral (적분)

D 동작: Differential (미분)

P 제어 (비례 제어)

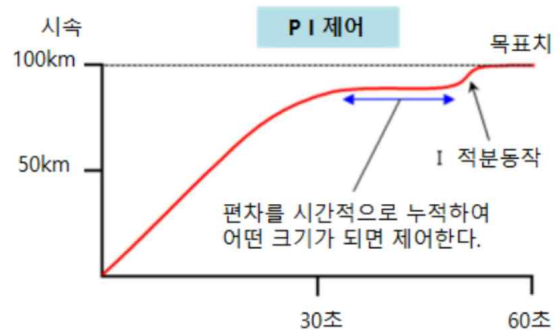


목표값과 현재 값의 차이에 비례한 조작량을 조절하는 제어 방식입니다. 범위의 조작량이 제어 대상 측정값의 변화에 따라 0 ~ 100% 사이를 연속적으로 변화시키는 것을 제한한 제어입니다.

그러나 측정값이 설정값에 도달하면 조작량이 너무 작아 제어할 수 없는 상태가 됩니다.

설정값에 근접한 상태에서 안정되어 측정값 = 설정값이 되지 않습니다.

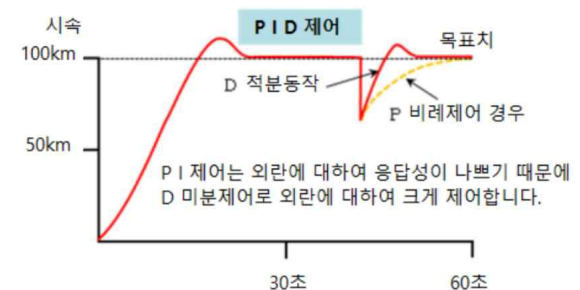
PI 제어 (비례 적분 제어)



P제어에서 편차를 없애기 위해 적분동작을 합니다. 편차를 시간적으로 축적하고 축적된 양이 크기로 된 곳에서 조작량을 증가시켜 편차를 없애는 방식으로 돌아갑니다.

그러나 PI제어는 편차를 시간 경과로 측정하기 때문에 원래의 값으로 복원하는데 시간이 걸리는 단점이 있습니다.

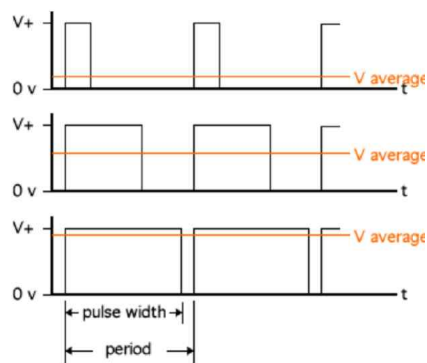
PID 제어 (비례 적분 미분 제어)



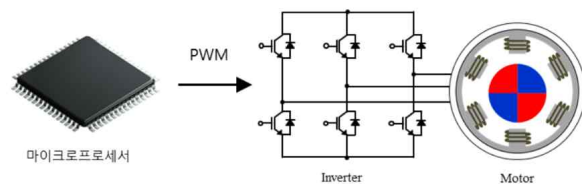
PI 제어는 시간이 걸리기 때문에 외력이 있을 때 빠르게 반응을 할 수 없습니다. 미분동작으로 이번 편차와 전회의 편차와 비교하여 편차의 대소 크기에 의해 조작량을 기민하게 반응하도록 합니다.

PWM 제어

PWM은 효율적인 스위칭 제어의 기본이 되는, 디지털 출력으로 아날로그 회로를 제어하는 기술.



on 시간과 off 시간의 비율을 변화시켜 전체적인 평균값을 조절해서 속도를 제어한다.



모터의 속도는 아날로그지만, 디지털 출력으로 제어를 합니다. 모터의 속도를 빠르게 할려면 duty ratio를 높이고, 속도를 늦출려면 duty ratio를 낮춥니다.

SLAM

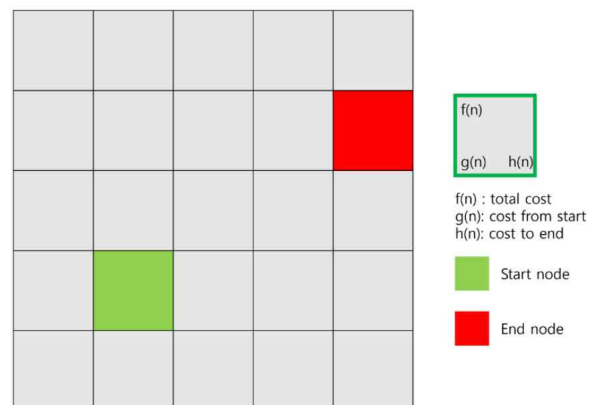
로봇의 위치 추정과 환경 지도 작성을 수행하는 기술입니다. SLAM은 로봇이 자신의 위치를 파악하고 동시에 주변 환경을 지도로 만듭니다.



다양한 센서를 (여기서는 Lidar) 사용하여 주변 환경 정보를 수집하고 센서 데이터 기반으로 현재 위치를 추정한다. 또한 이를 기반으로 지도로 작성한다. 주로 격자 혹은 특징점으로 표현되고 Grid-based SLAM, Feature-based SLAM으로 나뉜다. 또한 이동을 할때, 운동 모델을 업데이트하여 다음 위치를 예측한다. 예측된 위치와 실제 측정된 위치를 비교함으로써 위치 추정을 개선한다.

A* 알고리즘

A* 알고리즘은 주어진 출발 노드에서부터 목표 노드까지 가는 최단 경로를 찾아내는 그래프 탐색 알고리즘 중 하나이다. 주어진 지도에서 출발 지점부터 목표 지점까지의 최적의 경로를 찾는 기술인 전역 경로 계획 중 Path / Graph Search 알고리즘에 사용된다.



1. 출발 사각형에서 가장 작은 $f(n)$ 비용을 가진 사각형을 찾아 선택.

2. 다음 과정을 반복

(1) 열린 목록에서 가장 작은 $f(n)$ 비용을 가진 사각형을 찾아 선택

(2) 이것을 열린 목록에서 꺼내 닫힌 목록에 추가

(3) 선택한 사각형에 인접한 8개의 사각형에 대해 탐색

a. 인접 사각형중 닫힌 목록에 있거나 장애물인 것은 제외하고, 나머지 열린 목록에 사각형이 없다면 열린목록에 추가. 그리고 부모 노드 설정.

b.인접한 사각형이 이미 기존의 열린 목록에 있다면 현재 사각형을 기준으로 인접한 사각형까

지 이동할 때 $G(n)$ 비용이 낮아지지 않으면 아무 것도 하지 않고 낮으면 부모 노드르 현재노드로 바꾼다.

3. 탐색 중단 조건

(1) 경로 탐색 중 목표 사각형을 열린 목록에 추가 했을때

(2) 열린 목록이 비게 될때 (실패, 길이 없는 경우)

4. 목표 사각형으로부터 각각의 부모 사각형을 향하여 시작 사각형에 도착할때까지 거슬로 올라가면 최단 경로 도출이 된다.

■ 역할분담

소프트 웨어 역할		
1	현재 위치 파악	IMU 센서에서 얻은 가속도 및 각속도 값을 활용하여 자율주행 자동차의 방향 및 회전 정보를 추정하고, Encoder 모터의 회전 수를 활용하여 주행거리를 계산
2	제어	거리 정보를 데이터 값으로 얻고 PID 제어의 피드백을 통한 거리 조절, PWM 제어를 활용하여 모터에 가해지는 pulse를 조절해 속도를 제어하고 각도를 제어
3	최적 루틴 알고리즘	SLAM 지도 작성과 센서들의 데이터를 기반으로 주행 환경을 분석하고 최적의 경로를 계획하는 알고리즘 구현. A*

■ 고려 사항

1. Encoder 모터가 없을때 IMU 센서의 발산의 위치 부정확함을 대체하고자 SLAM 지도내의 위치와 카메라를 비교하여 오차가 생겼을때 위치 보정.
2. 자율 주행시 계단 혹은 다른 길로 경로 이탈을 방지하는 적절한 경로를 유지하는 알고리즘 혹은 제어 설계 .
3. 하드웨어 무게 중심에 따른 방향 치우침. 보정을 통해 정확한 주행 경로를 유지하고, 무게 중심에 따른 불안정성 보완.

■ 평가 내용

1학기: 경북대 IT 2호관 구름다리에서 시작해서 한바퀴 돌기

2학기: 사전에 알 수 없는 장애물을 설치하고 경북대 IT 2호관 구름다리에서 시작해서 한바퀴 돌기

감점 사항: 벽에 부딪혔지만 스스로 탈출하면 약간의 감점, 사람이 개입하면 큰 감점

성적: 팀 평가 60%, 개인 평가 40%

■ 기대효과 및 향후 계획

<https://www.schott.com/ko-kr/products/hermetic-packages-for-lidar-sensors-P1000281/applications>

<https://www.clearview-imaging.com/en/blog/stereo-vision-for-3d-mac-vision-applications>

https://eduino.kr/product/detail.html?product_no=61&gad_source=1&gclid=Cj0KCQiAxOauBhCaARIsAEbUSQQ9Lvddgw_d6AfUbi7QqZLq5DJjLuHr9e2OVcJVxKcZhGSoozkJEpAaAv_AEALw_wcB

<https://automaticaddison.com/calculate-pulses-per-revolution-for-a-dc-motor-with-encoder/>

<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/>

: <https://m.blog.naver.com/soyose1/221470449246>

<https://m.blog.naver.com/alsrb968/220728192740>

<https://blog.naver.com/lagrange0115/222345576071>