

---

캡스톤디자인(1)

# 시나리오 및 Usecase 옵션

Motus+er

# SLAM의 원리

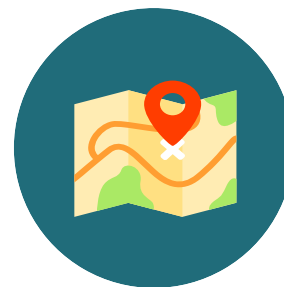
Simultaneous localization and mapping

항해하는 배가 별자리를 관찰하여 배의 위치를 파악하는 것처럼  
SLAM은 여러 특징점의 정보를 수식으로 계산하여 자기 위치를 알아낸다.  
이렇게 특징점으로 계산된 위치 정보를 기반으로  
케어버디는 집 안의 지도를 그릴 수 있다.



## LIDAR 데이터 수집

데이터는 로봇 주변의 물체까지의 거리를 나타내며,  
이를 통해 주변 환경의 지도를 만든다.



## 특징 추출 및 매칭

수집된 LIDAR 데이터에서 주요 지형지물이나 특징들이 추출된다. 이는 특정 점,  
모서리, 평면 등이 될 수 있다. 새로 수집된 특징들은 이전에 수집된 데이터 또는  
이미 알려진 지도 정보와 비교(매칭)된다.



## 그 외 : 오류 최소화 및 보정

모든 센서 데이터와 위치 추정 과정은 오류를 포함할 수 있다. 따라서 SLAM 시스  
템은 오류를 최소화하기 위해 다양한 알고리즘과 필터(예: 칼만 필터)를 사용하여  
데이터를 정제하고 보정한다.

# SLAM

SLAM은 해당 맵 내에서 로봇이 자신의 위치를 추적하는 동시에 주변 환경의 맵을 구축하는 과정  
즉, 동시적 "위치 추정" 및 "지도 작성"

## RPLIDAR 패키지 사용시 들어오는 데이터 형태

TIME

스캔하여 데이터를 획득한 시간

FLOAT32 ANGLE\_MIN [RAD]

스캔의 시작 각도

FLOAT32 ANGLE\_MAX [RAD]

스캔의 끝 각도

FLOAT32 ANGLE\_INCREMENT [RAD]

측정되는 각의 단위

FLOAT32 SCAN\_TIME [SEC]

측정되는 시간 간격

FLOAT32 RANGE\_MIN [M]

최소 RANGE(거리) 값

FLOAT32 RANGE\_MAX [M]

최대 거리 값

FLOAT32[] RANGES [M]

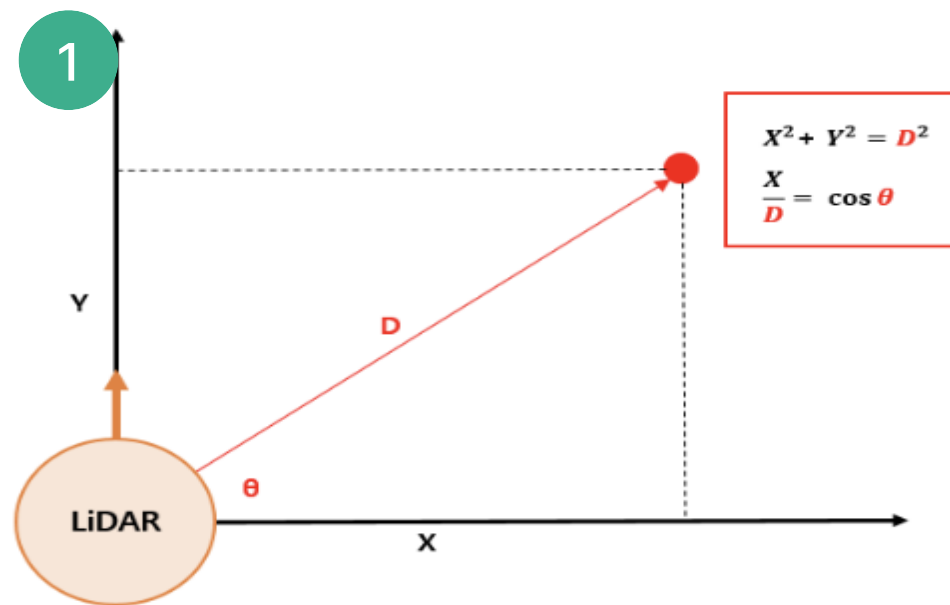
최소 거리(RANGE\_MIN) 이상, 최대 거리(RANGE\_MAX)  
이하의 거리 값들

FLOAT32[] INTENSITIES

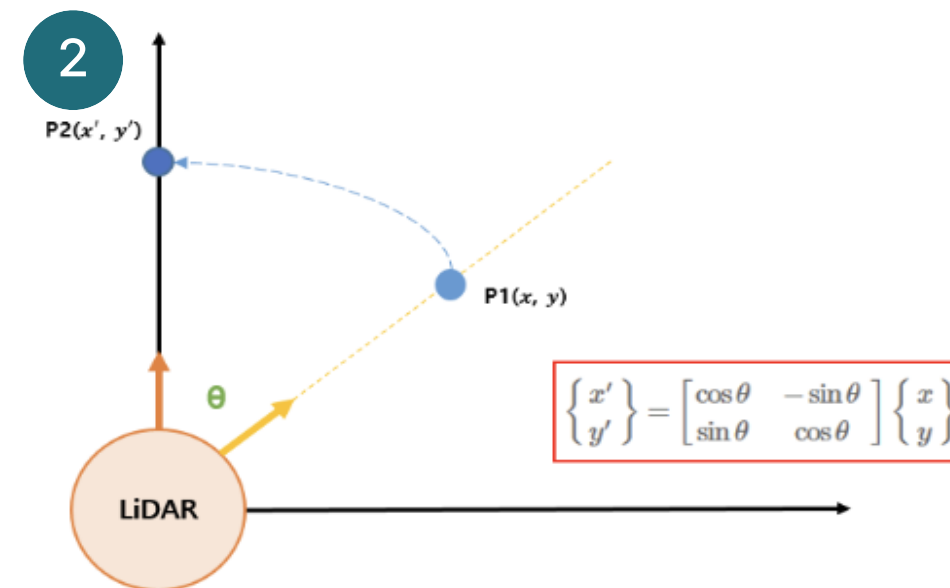
레이저가 부딪혀 돌아오는 강도

# LiDAR에서 제공하는 Data로 위치 계산

LiDAR에서 제공하는 Data는 Distance와 Angle을 이용해  
공간에서의 Object의 위치를 계산하는 과정



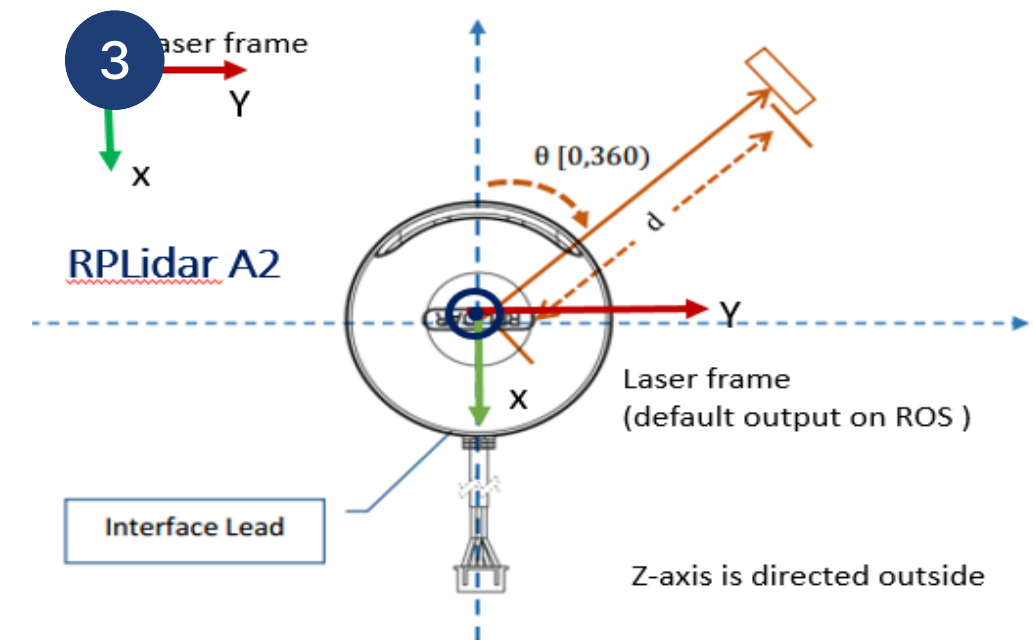
위 이미지에서의 빨간 점이 OBJECT의 위치라고 할 때 간단한 수학 식을 통해 좌표를 측정할 수 있다.



LIDAR의 0도 LINE을 LIDAR를 설치할 때 정확하게 배치할 수 없다면 CALIBRATION을 수행하는 프로그램을 하나 만들어 설치된 방향을 저장하고 좌표 값을 회전 변환시켜 다음과 같은 계산을 통해 MAPPING 해주면 된다.

(X, Y)를  $\theta$  만큼 회전시키면 (X', Y')를 얻을 수 있게 된다.  
이때  $\theta$  는 CALIBRATION을 수행할 때 설정해주어야 하는 값이다.

## 각도 및 거리의 표기



- Z축 양의 방향으로(반시계방향, 오른손 엄지를 위로 했을 때 손가락이 감아지는 방향) 각도를 증가시킨다.
- 0°는 X축을 따라 뻗어진다

# LiDAR에서 제공하는 Data로 위치 계산

LiDAR에서 제공하는 Data는 Distance와 Angle을 이용해  
공간에서의 Object의 위치를 계산하는 과정

## RPLIDAR ROS PACKAGE : 각도 및 거리의 표기

- 1 RPLIDAR의 SLAM을 위한 기본적인 소프트웨어 패키지에는  
복잡한 수식과 알고리즘이 이미 구현되어 있다.  
(github에서 다운 가능)  
즉, 이러한 패키지를 사용하여 직접 수식을 계산할 필요 없이  
센서 데이터를 처리하고 케어버디의 위치를 추정할 수 있다.

위 이미지에서의 빨간 점이 OBJECT의 위치라고 할 때 간단한  
수학 식을 통해 좌표를 측정할 수 있다.

LIDAR의 0도 LINE을 LIDAR를 설치할 때 정확하게 배치할 수 없다면  
CALIBRATION을 수행하는 프로그램을 하나 만들어 설치된 방향을  
저장하고 좌표 값을 회전 변환시켜 다음과 같은 계산을 통해  
MAPPING 해주면 된다.

$(X, Y)$ 를  $\theta$  만큼 회전시키면  $(X', Y')$ 를 얻을 수 있게 된다.  
이때  $\theta$  는 CALIBRATION을 수행할 때 설정해주어야 하는 값이다.

- Z축 양의 방향으로(반시계방향, 오른손 엄지를 위로 했을 때  
손가락이 감아지는 방향) 각도를 증가시킨다.
- 0°는 X축을 따라 뻗어진다

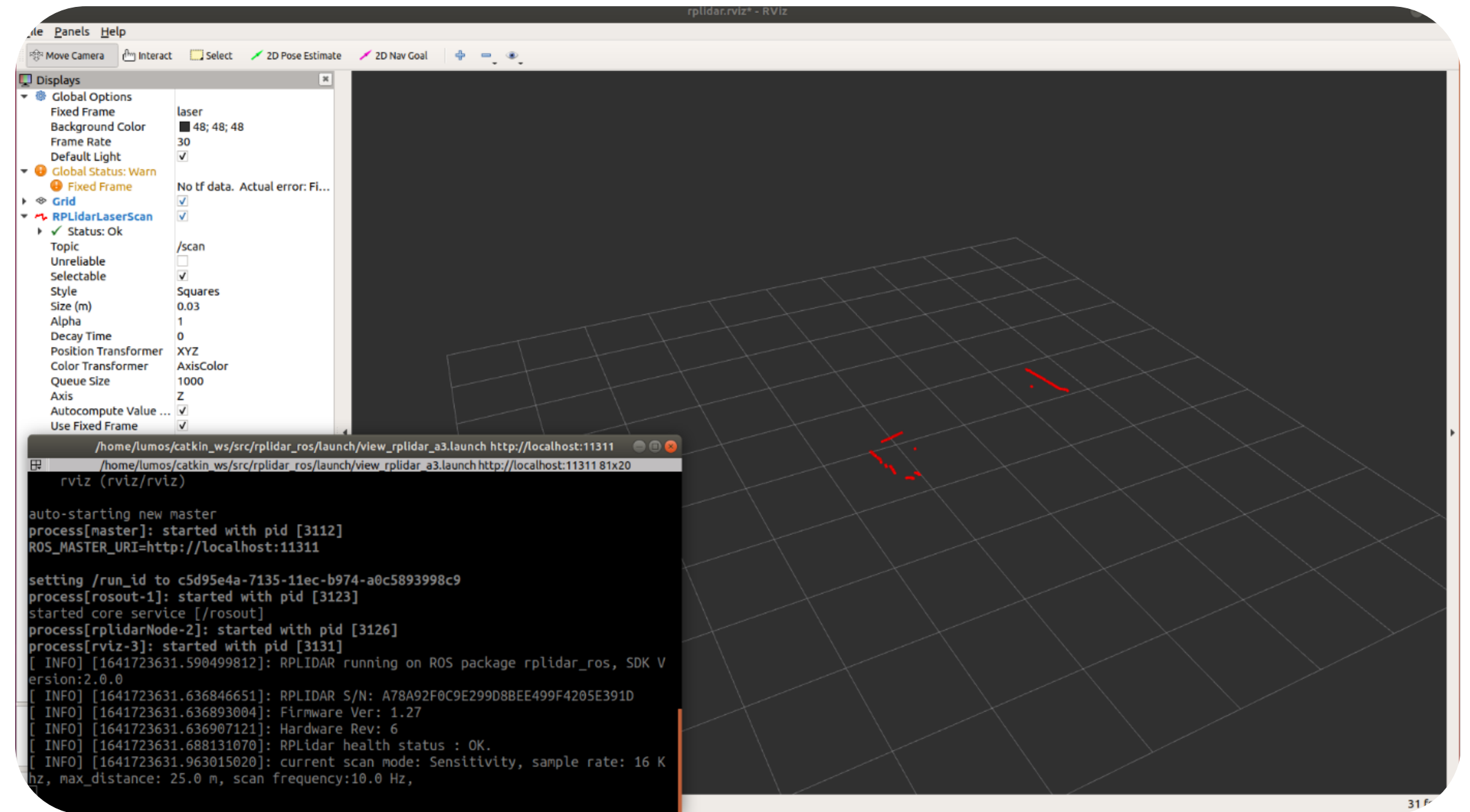
이러한 계산들이 패키지에 전부 존재함

# 시각화 방법 RoS - rviz

RoS에서 제공하는 시각화툴

센서에서 수집되는 값  
시각화 가능

경로 및  
특정 공간에 대한  
설정 가능



# 옵션으로 변경할 Usecase

1

## 장애물 감지 및 제동

다른 메인 기능들을 우선적으로 하고  
나중에 여유가 생기면 추가하기로 하였다.

2

## 음성 인식

다른 메인 기능들을 우선적으로 하고  
나중에 여유가 생기면 추가하기로 하였다.  
이에 제어는 자동적으로 혹은 디스플레이를 통해 이루어진다.

3



## 최적 위치

삼성에서 공기청정기 개발자에게 물어본 결과, 벽이나 다른 가구에 흡입구나 센서가 막히면  
공기청정 효과와 먼지농도 센싱 성능이 떨어질 수 있다고 하였다.  
이에 좌/우 60CM, 벽과 25CM 이상의 충분한 공간을 확보해야 한다고 하였다.

즉, 측정을 하며 최적의 위치를 찾는 것이 아니라 보편적인 최적의 위치로 USECASE를 작성하고자 한다.

---

# Thank you