

4-17 네비게이션 (Navigation)

강의 요약

01

패스 파인딩 vs.
모션 플래닝

02

멀티 로봇 패스
파인딩

- Task Allocation
- Multi-robot Path Finding

03

Conflict-
Based Search
(CBS)

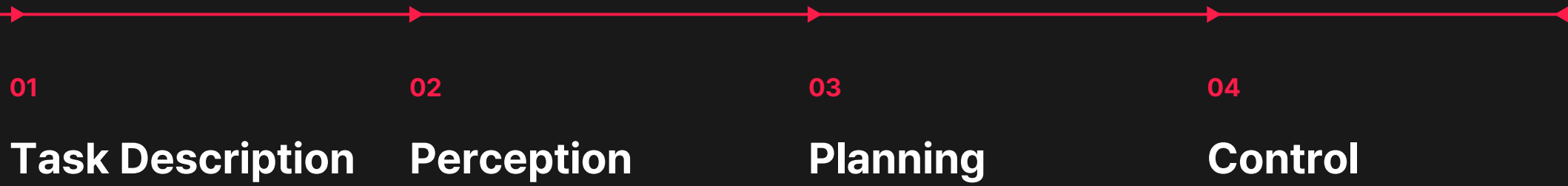
- Two-level Search
- 완전성
- 최적성

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



- 네비게이션

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템

01

Task Description

- 네비게이션

02

Perception

- 카메라, 라이다
- 로봇의 위치 추정
- 장애물 확인

03

Planning

04

Control

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템

01

Task Description

- 네비게이션

02

Perception

- 카메라, 라이다
- 로봇의 위치 추정
- 장애물 확인

03

Planning

- 경로 계획

04

Control

- 로봇을 구동
- 외란 (미끄럼) 의 작용

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템

01

Task Description

- 네비게이션

02

Perception

- 카메라, 라이다
- 로봇의 위치 추정
- 장애물 확인

03

Planning

- 경로 계획

04

Control

- 로봇을 구동
- 외란 (미끄럼) 의 작용

Localization

나의 위치가 어디에 있는지 결정

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템

01

Task Description

- 네비게이션

02

Perception

- 카메라, 라이다
- 로봇의 위치 추정
- 장애물 확인

환경 정보를 사전에 제공하는가?

→ O: Mapping

→ X: Simultaneous Localization And Mapping (SLAM)

03

Planning

- 경로 계획

Localization

나의 위치가 어디에 있는지 결정

04

Control

- 로봇을 구동
- 외란 (미끄럼) 의 작용

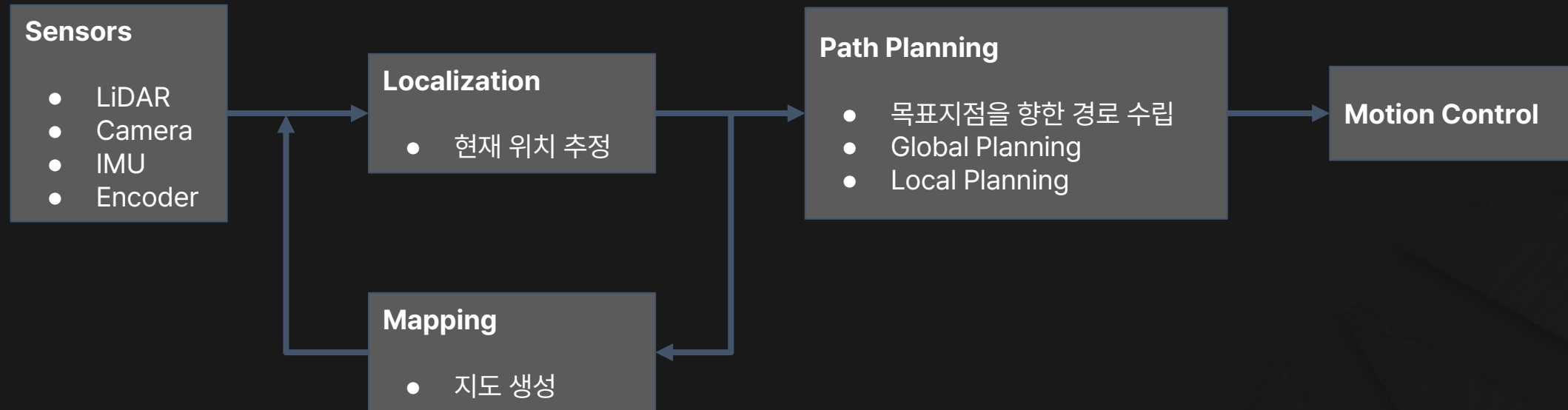
네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



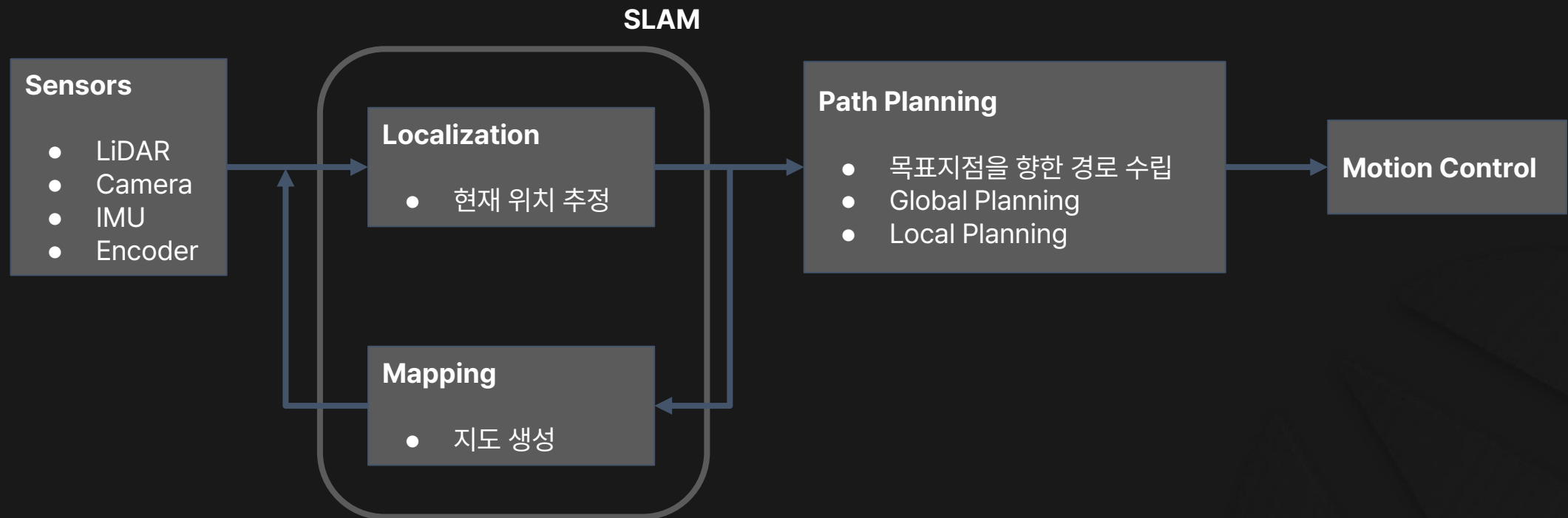
네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



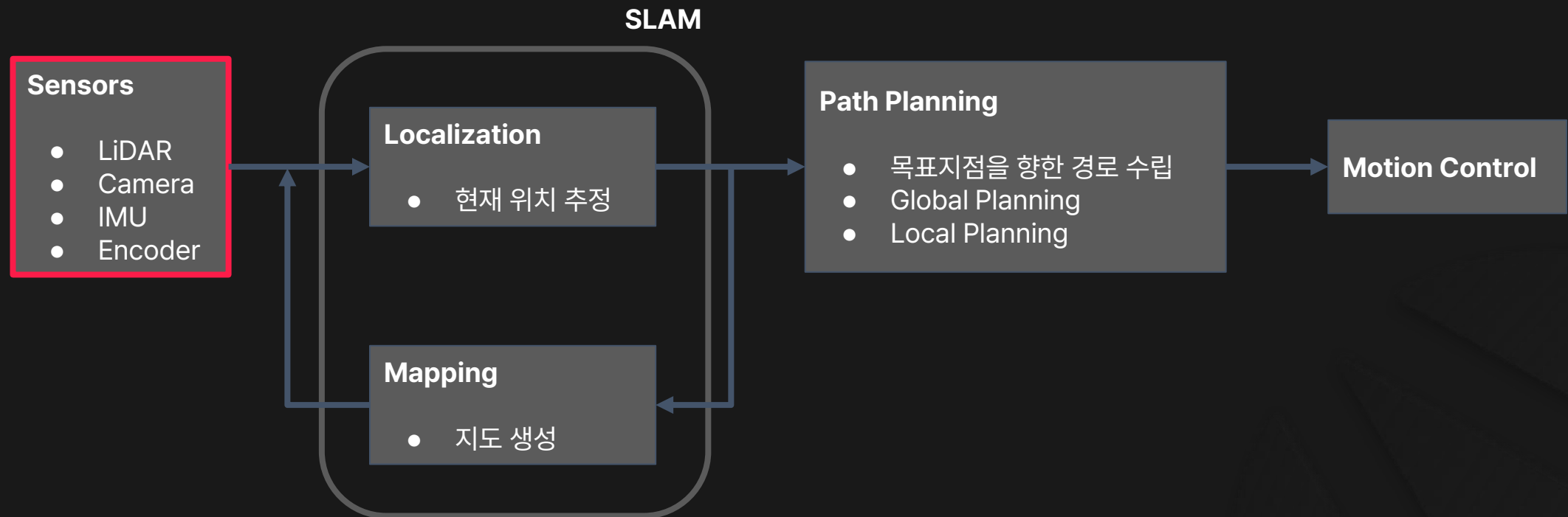
네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

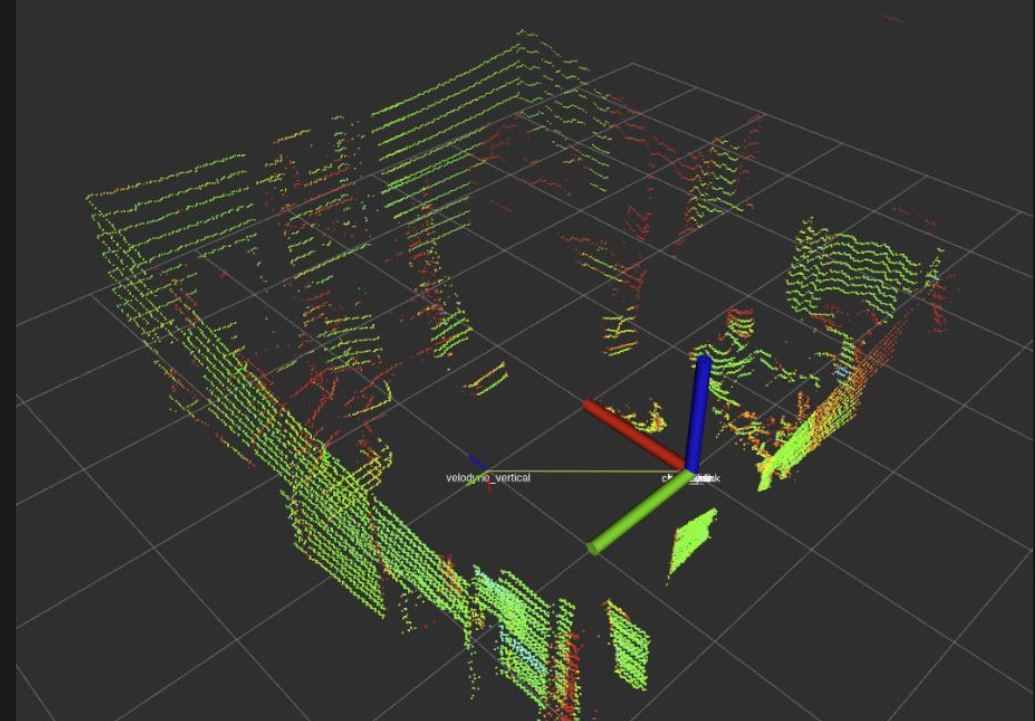
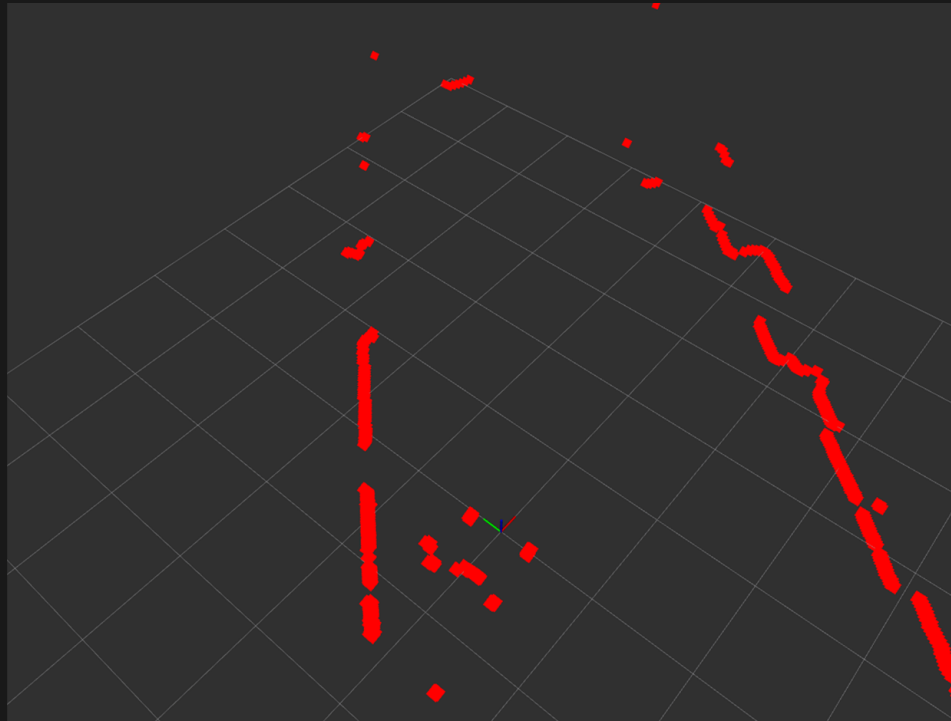
- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



Sensors

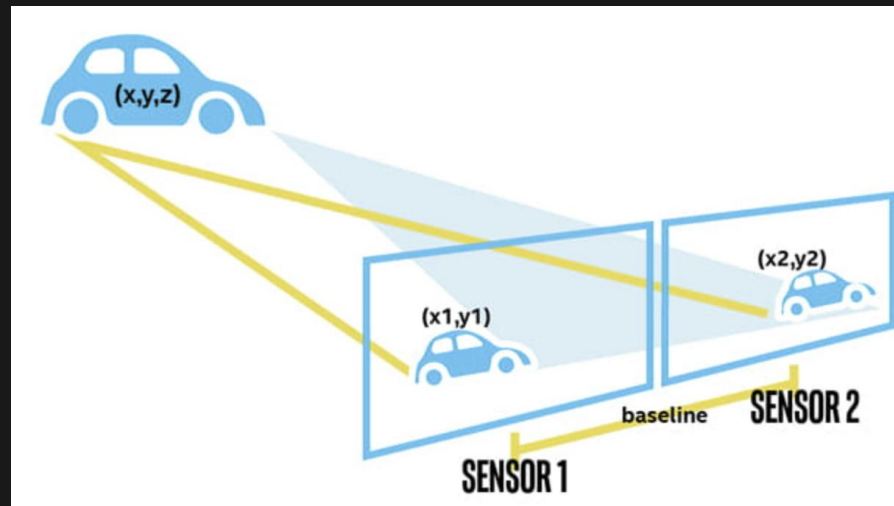
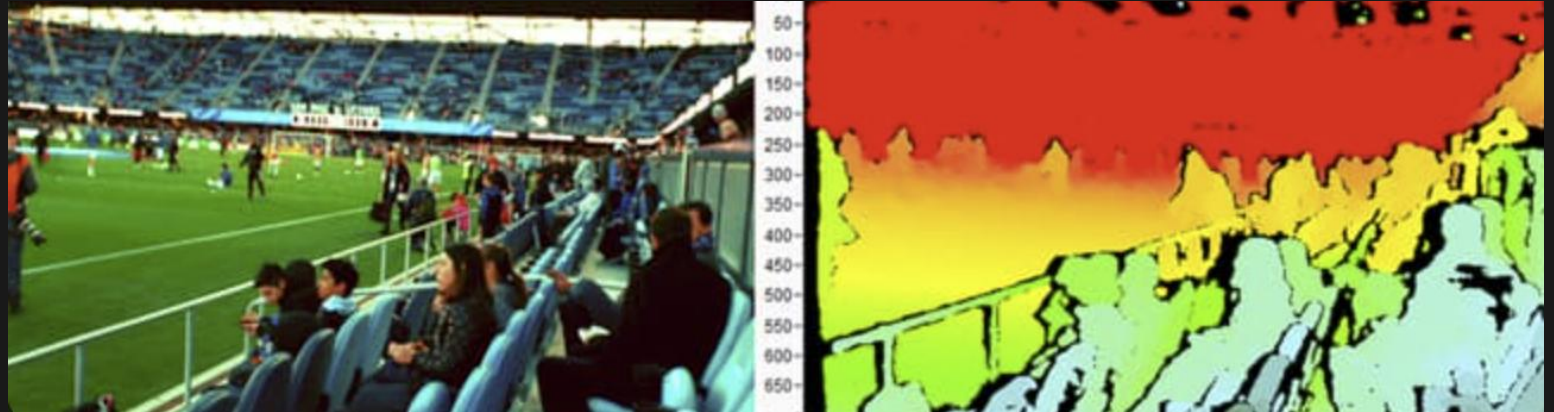
Sensors

- LiDAR
 - 시간의 정보가 없는 “포인트 클라우드 (Point Cloud)”
 - 3차원의 정보를 저장



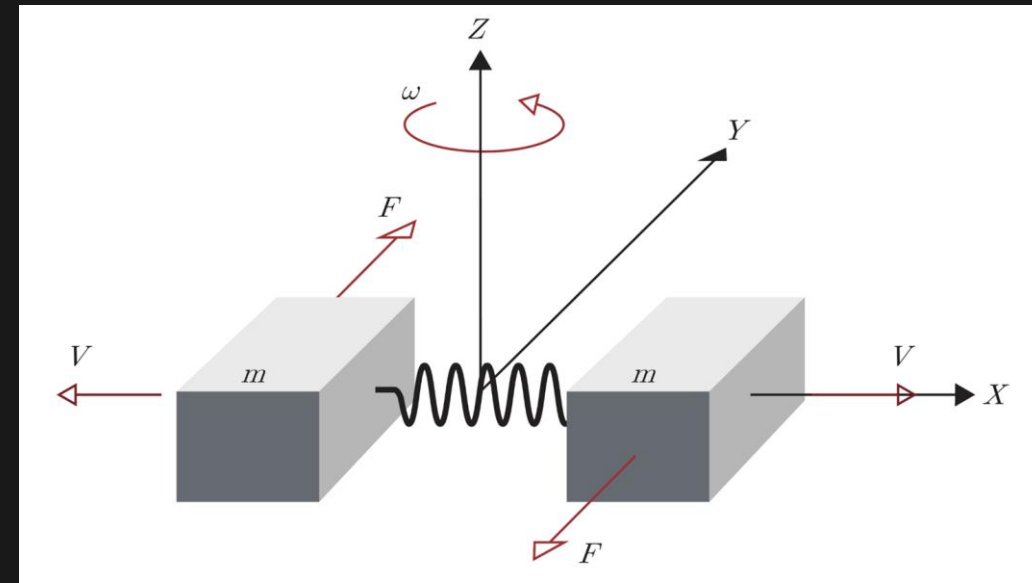
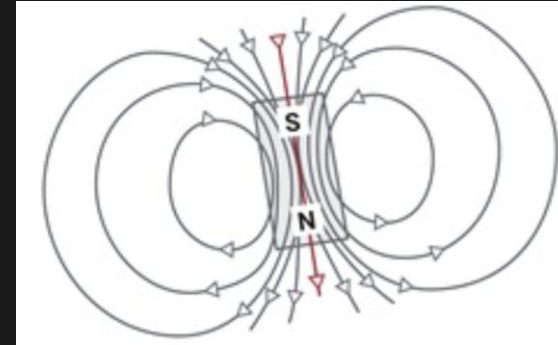
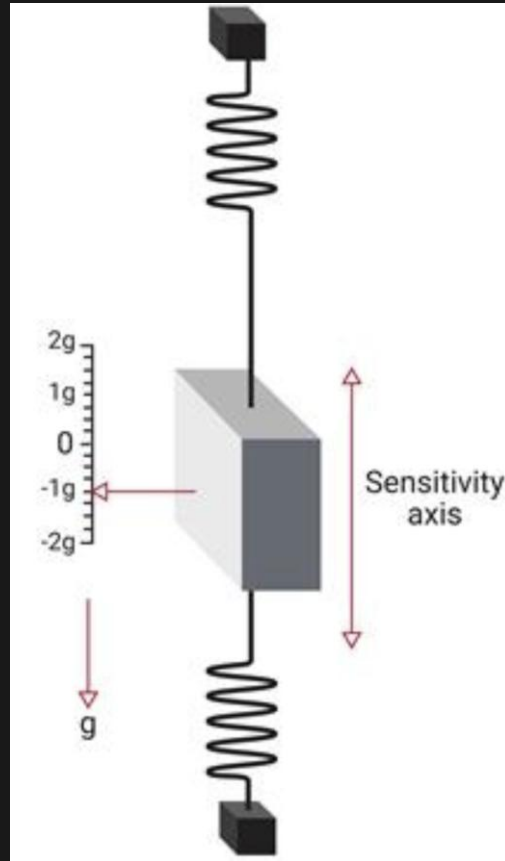
Sensors

- Camera
 - RGB: 픽셀별 색 정보
 - Depth: 픽셀별 깊이 정보
- Stereo Camera
 - 두개의 카메라로 깊이 정보 계산



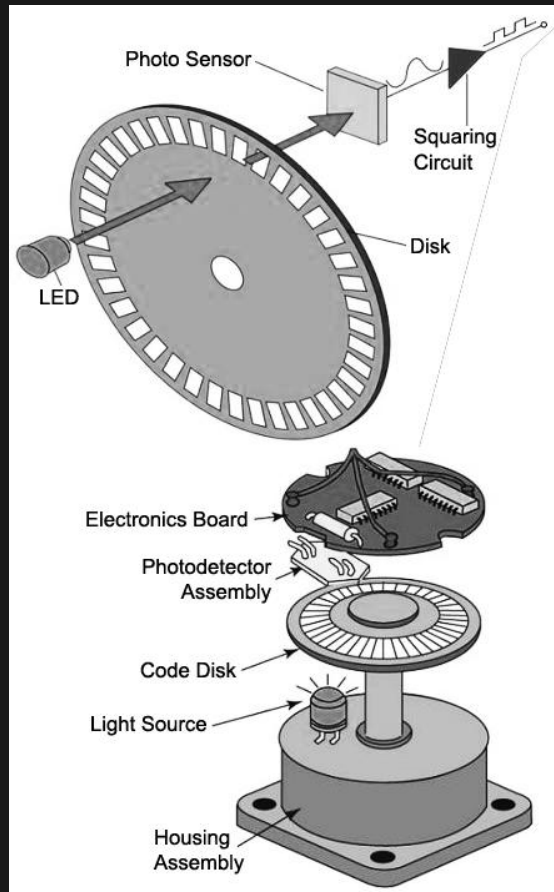
Sensors

- Inertial Measurement Unit
 - 선 가속도 (Linear acceleration)
 - 회전 가속도 (Rotational acceleration)
 - 자기장 (Magnetic Field)



Sensors

- Encoder
 - 모터의 회전 각도 계산



Sensors

- 종류
 - LiDAR
 - Camera
 - IMU
 - Encoder

Sensors

- 종류
 - LiDAR
 - Camera
 - IMU
 - Encoder
- } 정적인 정보 → 동적 정보를 위해서는 추가 작업이 필요
- } 동적인 정보

Sensors

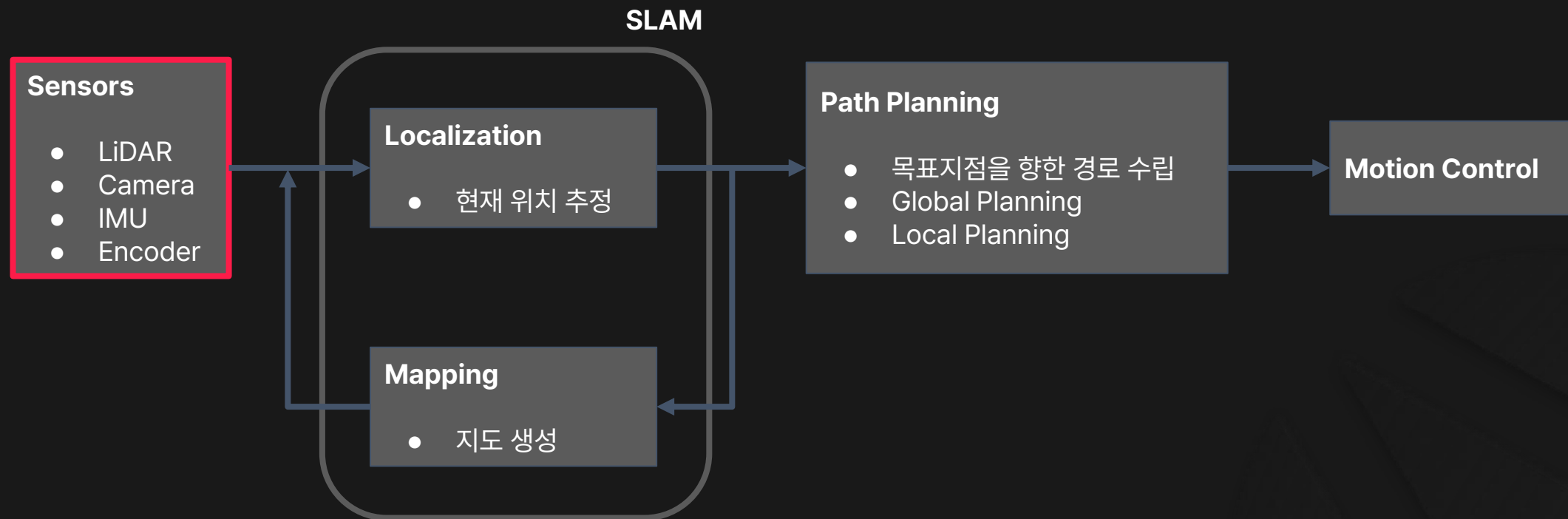
- 종류
 - LiDAR
 - Camera
 - IMU
 - Encoder

정적인 정보 → 동적 정보를 위해서는 추가 작업이 필요

동적인 정보
- 센서 오차
 - 각 센서의 상호 보완 관계
 - Filtering, Sensor Fusion

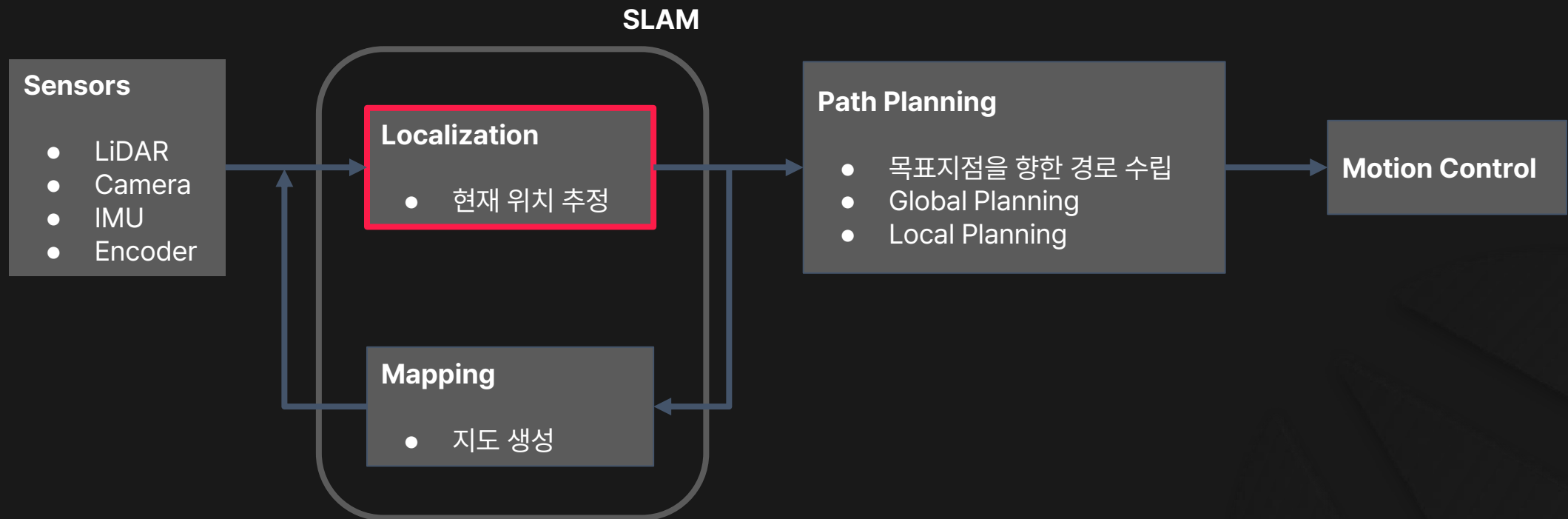
네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



Localization

- 길을 찾기 위해서는 "내가 어디 있는지" 알아야 함
- 지도 상 좌표 없이 경로를 계획할 수 없음
- 실제 도로 주행, 실내 이동 모두 localization이 필수

Localization

- 길을 찾기 위해서는 "내가 어디 있는지" 알아야 함
- 지도 상 좌표 없이 경로를 계획할 수 없음
- 실제 도로 주행, 실내 이동 모두 localization이 필수
- 로봇이 자신의 현재 위치(위치 \times 방향 = Pose)를 이미 알고 있는 맵 위에서 추정하는 과정

2D: (x, y, θ)

3D: $(x, y, z, \phi, \theta, \psi)$

Localization

- 길을 찾기 위해서는 "내가 어디 있는지" 알아야 함
- 지도 상 좌표 없이 경로를 계획할 수 없음
- 실제 도로 주행, 실내 이동 모두 localization이 필수
- 로봇이 자신의 현재 위치(위치 \times 방향 = Pose)를 이미 알고 있는 맵 위에서 추정하는 과정

2D: (x, y, θ)

3D: $(x, y, z, \phi, \theta, \psi)$

- 센서 활용
 - Encoder: 바퀴의 회전수를 측정
 - IMU: 가속도 정보를 이용하여 이동 속도 및 회전 계산
 - LiDAR: 주변의 거리정보 측정
 - Camera: 주변 특징점들을 파악

Localization

- Dead Reckoning
 - 이전 위치와 이동거리를 통해 현재 위치를 계산하는 방법
 - 사용센서: Encoder (Odometry), IMU
 - 장점: 구현이 매우 간단함, 외부 환경 정보 없이도 작동 가능
 - 단점: 누적 오차 발생, 시간에 따라 위치 부정확

$$x_t = x_{t-1} + \Delta d \cdot \cos(\theta_{t-1})$$

$$y_t = y_{t-1} + \Delta d \cdot \sin(\theta_{t-1})$$

$$\theta_t = \theta_{t-1} + \Delta\theta$$

Δd : 시간 간격 동안 이동한 거리 (Odometry 기반)

$\Delta\theta$: 회전한 각도

x_t, y_t : 현재 위치

θ_t : 현재 방향(heading)

Localization

- Adaptive Monte Carlo Localization (AMCL)

- 몬테 카를로 방법론 (Monte Carlo Methods): 복잡한 수학적 문제를 정확하게 계산하기 어려울 때, 무작위 샘플 (Random Sampling)을 이용해 통계적으로 근사하는 기법.

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{i=1}^N f(x^{(i)})$$

$x^{(i)}$: 확률 분포 $p(x)$ 에서 샘플링한 샘플

예시) 원주율 근사

Localization

- Adaptive Monte Carlo Localization (AMCL)
 - 몬테 카를로 방법론 (Monte Carlo Methods)
 - 입자 필터 (Particle Filter) 기반의 알고리즘
 - 로봇의 가능한 위치 후보들을 입자(particle)로 표현하고, 센서 정보에 따라 이 입자들의 신뢰도 (가중치)를 갱신

Localization

- **Adaptive Monte Carlo Localization (AMCL)**
 - 몬테 카를로 방법론 (Monte Carlo Methods)
 - 입자 필터 (Particle Filter) 기반의 알고리즘
 - 로봇의 가능한 위치 후보들을 입자(particle)로 표현하고, 센서 정보에 따라 이 입자들의 신뢰도 (가중치)를 갱신
 - 구성요소
 - **Particles:** 로봇의 위치 (x, y, θ)
 - **Motion Model:** Odometry
 - **Sensor Model:** 입자의 신뢰도를 계산하기 위한 보조 수단
 - **Resampling:** 신뢰도가 높은 입자를 중심으로 새로운 입자를 샘플

Localization

- **Adaptive Monte Carlo Localization (AMCL)**
 - 몬테 카를로 방법론 (Monte Carlo Methods)
 - 입자 필터 (Particle Filter) 기반의 알고리즘
 - 로봇의 가능한 위치 후보들을 입자(particle)로 표현하고, 센서 정보에 따라 이 입자들의 신뢰도 (가중치)를 갱신
 - 구성요소
 - **Particles:** 로봇의 위치 (x, y, θ)
 - **Motion Model:** Odometry
 - **Sensor Model:** 입자의 신뢰도를 계산하기 위한 보조 수단
 - **Resampling:** 신뢰도가 높은 입자를 중심으로 새로운 입자를 샘플
 - 과정

Localization

- Adaptive Monte Carlo Localization (AMCL)
 - 몬테 카를로 방법론 (Monte Carlo Methods)
 - 입자 필터 (Particle Filter) 기반의 알고리즘
 - 로봇의 가능한 위치 후보들을 입자(particle)로 표현하고, 센서 정보에 따라 이 입자들의 신뢰도 (가중치)를 갱신
 - 구성요소
 - Particles: 로봇의 위치 (x,y,theta)
 - Motion Model: Odometry
 - Sensor Model: 입자의 신뢰도를 계산하기 위한 보조 수단
 - Resampling: 신뢰도가 높은 입자를 중심으로 새로운 입자를 샘플
 - 과정
 - 초기화: 지도 위에 수백~수천개의 입자를 무작위로 뿌림
 - Motion Update: 로봇을 이동시키고, 입자들도 동일하게 이동
 - Sensor Update: 각 입자와 함께 센서값들도 동일하게 이동시키고, 실제 센서 데이터와 비교
 - Resampling: 유사도에 따라 가중치를 부여하고 입자 분포를 정제

Localization

- Kalman Filter: 센서 측정값을 기반으로 시스템의 상태를 추정하는 알고리즘

Localization

- Kalman Filter: 센서 측정값을 기반으로 시스템의 상태를 추정하는 알고리즘
 - 예측(Prediction): 현재 상태에서부터 다음 상태를 예측
 - 보정(Update): 실제 측정값이 들어오면 예측값을 보정

Localization

- Kalman Filter: 센서 측정값을 기반으로 시스템의 상태를 추정하는 알고리즘
 - 예측(Prediction): 현재 상태에서부터 다음 상태를 예측
 - 상태 예측: 내가 어디에 있을지 예상
 - 공분산 예측: 내 예상을 얼마나 신뢰할 수 있는지 계산
 - 보정(Update): 실제 측정값이 들어오면 예측값을 보정
 - 칼만 게인 계산: 센서를 믿을 것인가? 아니면 내 예측을 믿을 것인가?
 - 상태 보정: 센서의 정보를 반영하여 상태를 보정
 - 공분산 보정: 센서의 정보를 반영하여 내 예상의 신뢰도 보정

Localization

- Kalman Filter: 센서 측정값을 기반으로 시스템의 상태를 추정하는 알고리즘
 - 예측(Prediction): 현재 상태에서부터 다음 상태를 예측
 - 상태 예측: 내가 어디에 있을지 예상
 - 공분산 예측: 내 예상을 얼마나 신뢰할 수 있는지 계산
 - 보정(Update): 실제 측정값이 들어오면 예측값을 보정
 - 칼만 게인 계산: 센서를 믿을 것인가? 아니면 내 예측을 믿을 것인가?
 - 상태 보정: 센서의 정보를 반영하여 상태를 보정
 - 공분산 보정: 센서의 정보를 반영하여 내 예상의 신뢰도 보정

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-$$

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



- 로봇이 센서 정보를 이용해 주변 환경의 구조를 지도 형태로 표현 (Mapping)
- 이 과정이 잘 되어야 Localization, Path Planning도 정확히 수행될 수 있음

Mapping

- Occupancy Grid Mapping

- 환경 표현: Grid World

$$p(m_{i,j}) = \mathbb{P}(\text{cell } (i, j) \text{ is occupied})$$

$$\mathbf{M} = \{p(m_1), p(m_2), \dots, p(m_N)\}$$

- 센서 기반 환경 예측:

$$p(m \mid z_{1:t}, x_{1:t})$$

서 z 와 로봇 위치 x 를 바탕으로, 이 셀이 장애물일 확률은 얼마일까?"

→ Bayesian Update:

$$p(m \mid z_{1:t}, x_{1:t}) = \frac{p(z_t \mid m, x_t) \cdot p(m \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

해서 계산

→

$$l(m) = \log \left(\frac{p(m)}{1 - p(m)} \right)$$

"있을 가능성 vs. 없을 가능성"

$$l_t^{(i)} = l_{t-1}^{(i)} + \log \left(\frac{p(m_i \mid z_t)}{1 - p(m_i \mid z_t)} \right) - l_0$$

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



Path Planning

- 시작 위치에서 목표 위치까지 충돌 없이 갈 수 있는 경로를 찾는 과정
 - Global Planner (전역 경로 계획자)
 - 로봇이 목적지까지 가는 큰 그림을 계획
 - A*, Dijkstra, RRT, PRM 등
 - Local Planner (지역 경로 계획자)
 - 로봇 주변의 짧은 거리만 고려하여 동작 결정
 - 센서 정보로 주변 장애물을 반영해서 경로를 수정 → 동적 장애물 회피
 - Dynamic Window Approach (DWA)

Local Planning

- Dynamic Window Approach (DWA): 단기적인 안전하고 효율적인 경로를 선택

- 로봇의 즉각적인 제어명령을 위해 속도 조합 다음 커맨드를 결정

- 단계:

- 현재 속도 기준으로 가능한 속도-회전 속도 후보들을 샘플링

- 각 속도 조합으로 단기 시뮬레이션

- 세 가지 평가 함수로 점수 계산: Goal heading, Obstacle distance, velocity

$$G(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{distance}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)$$

- 총 점수 가장 높은 후보 선택 → 실제 속도 명령으로 보냄

네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



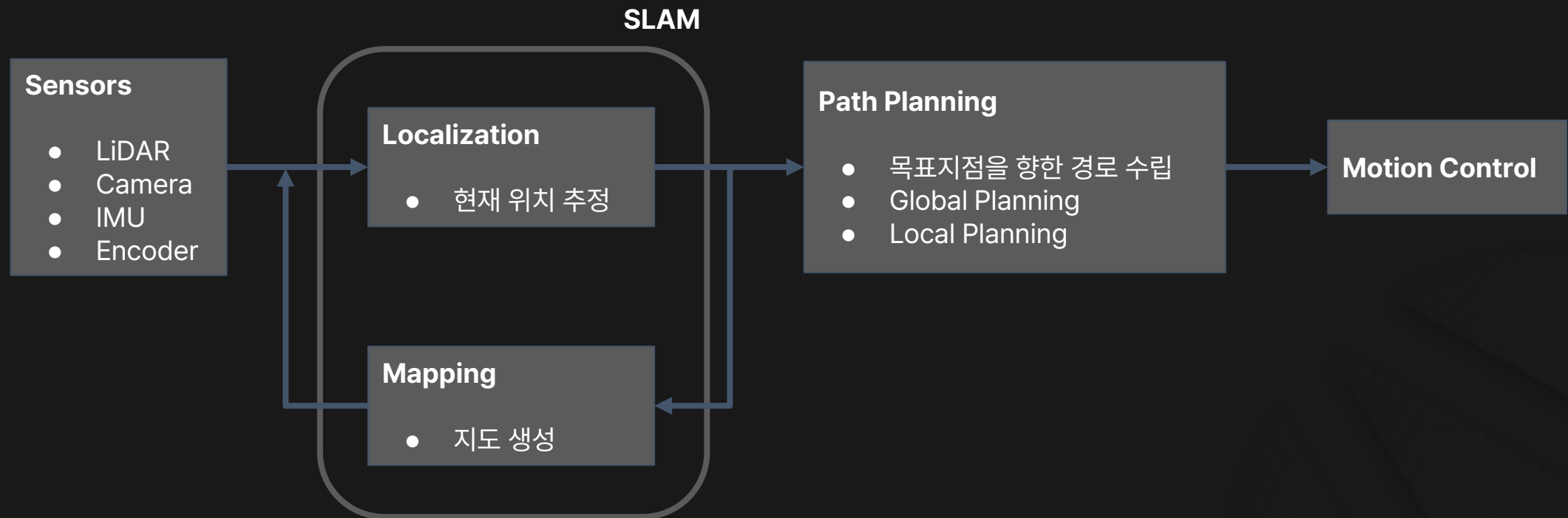
네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



네비게이션의 정의

- 로봇이 주어진 목표 지점까지 이동하기 위한 시스템



강의 요약

01

네비게이션의 구성요소

- Sensor
- Localization
- Mapping
- Path Planning
- Motion Control

02

Localization

- Dead Reckoning
- Adaptive Monte Carlo Localization
- Kalman Filter

03

Occupancy Grid Mapping

- Grid world
- Bayes update
- Log-odds

04

Path Planning

- Global Planner
- Local Planner:
Dynamic Window
Approach (DWA)