

# Path Planning and Execution for Real Worlds

심현철

## 경로 및 운동계획법

- 경로 및 운동 계획

로보틱스에서 가장 중요한 분야들 중 하나

**복잡도**와 **구속환경** : 완벽한 계획법을 저해하는 요소

**최적의 경로**를 가능한 빨리 찾아내는 기술



- 관련 분야

로봇의 조작 기술 및 계획법

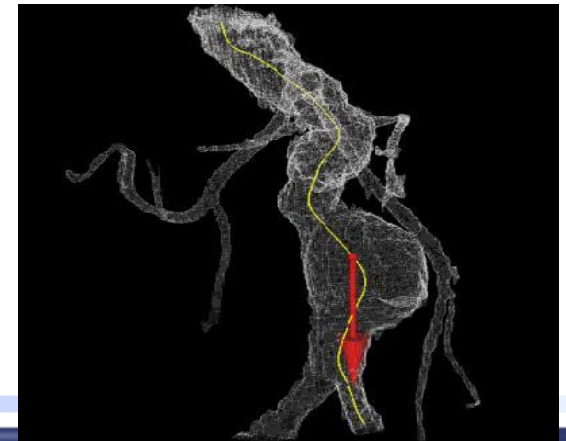
Nonholonomy 및 동역학 방정식 고려

위험 감지 및 회피 계획

최적의 경로 탐색

시간에 따라 변하거나 움직이는 장애물 회피

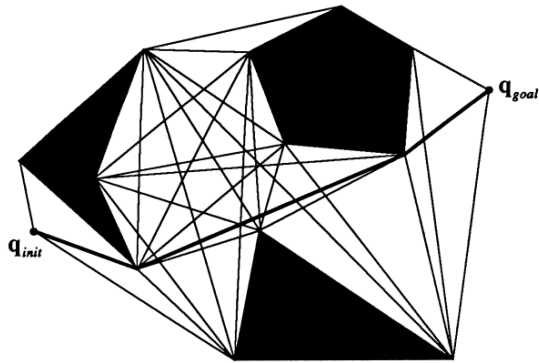
감지 및 예측에 따른 불확실성 파악



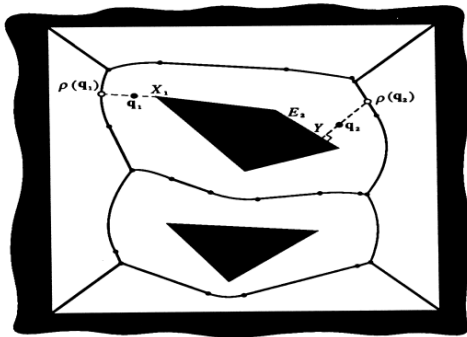
## Path Planning

### Roadmap

#### Visibility Graph

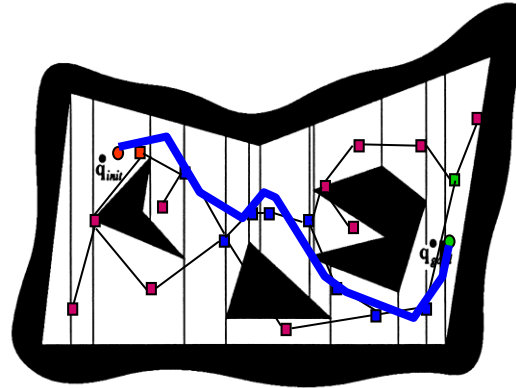


#### Voronoi Diagram

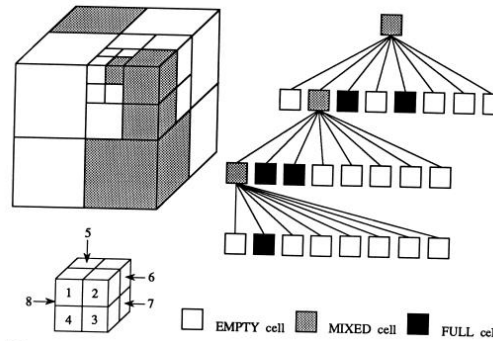


### Cell Decomposition

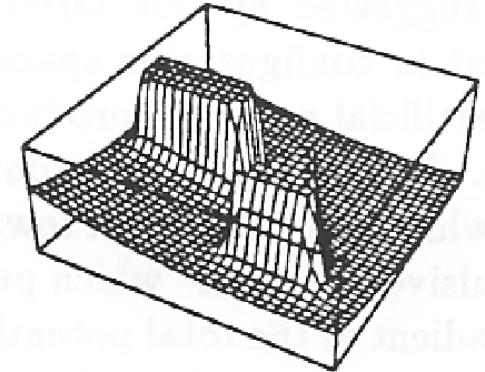
#### Trapezoidal Decomposition



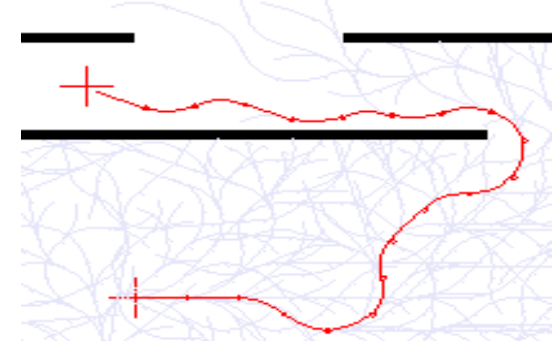
#### Octree Decomposition



### Potential Field



### Rapidly-Exploring Random Tree



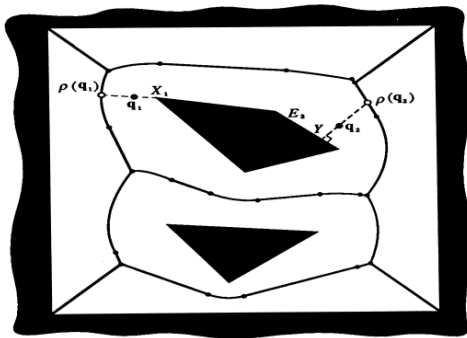
## Real Time Path Planning

### Roadmap

경로 생성 결과를 예측 가능

환경의 복잡도에 따라  
계산량 증가  
→ 실시간 사용이 가능  
하도록 수정

### Voronoi Diagram



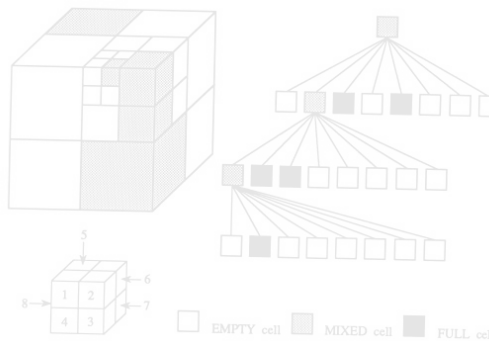
### Cell Decomposition

#### Trapezoidal Decomposition



지정된 구역이 아닌  
**Open Field (실제환경)에  
사용하기 적합하지 않음**

#### Octree Decomposition



### Potential Field



**Local Minima**에 빠져  
Solution을 구하지  
못할 수 있음

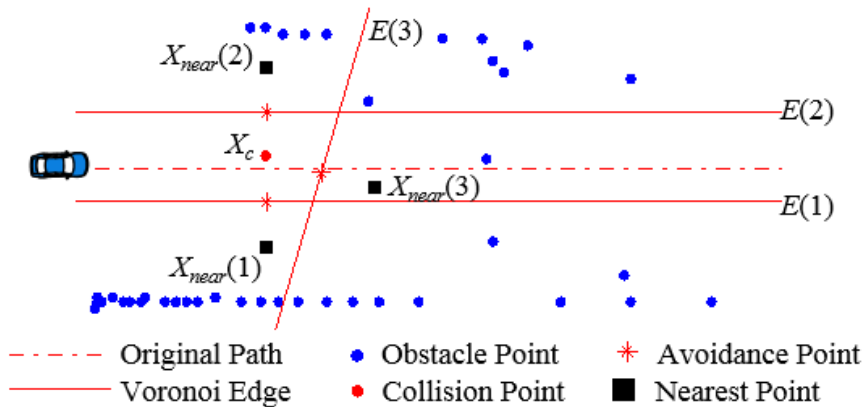
### Rapidly-Exploring Random Tree

실시간 경로 계획에  
사용하기 위해 계산 시간을  
줄이면 **동일한 조건에도  
매번 다른 결과를 도출**

개발자(탐승자)로 하여금  
**불안감 조성**

## One Voronoi Cell Algorithm

### One Voronoi Cell Algorithm

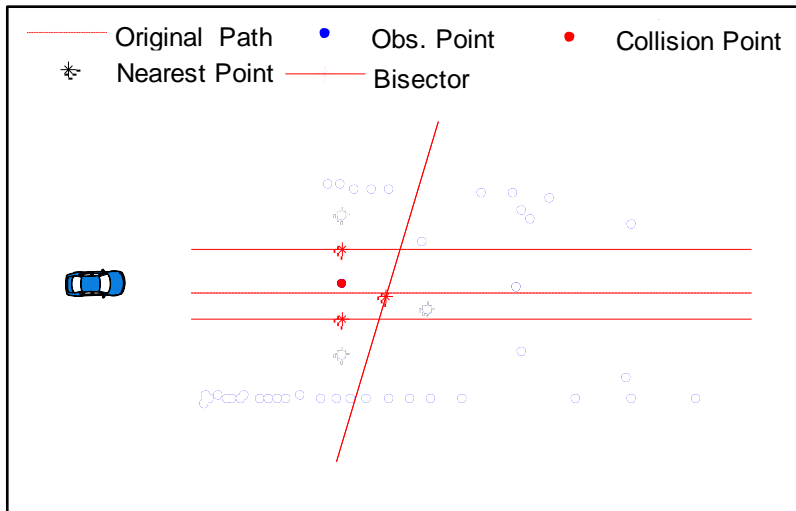


Voronoi diagram의 정의를 이용하여 장애물 사이의 **최대 Clearance를 보장**

경로상의 충돌하는 한개의 voronoi core 만을 이용하여 계산  
**실시간 장애물 회피 경로 생성 가능**

장애물 간의 간격이 큰 경우 크게 회피 하는 것을  
방지하지 위해 Voronoi cell의 크기를 제한시켜  
**최소한의 기동으로 장애물 회피**

**동일한 장애물 환경에서 동일한 결과 도출**



**One Voronoi Cell Algorithm** ( $X_{obs}, X_c$ )

$n \leftarrow 0$

do

$n \leftarrow n + 1$

$X_{near}(n) \leftarrow GetNearestPoint(X_{obs}, X_c)$

$(E(n), X_{avoid}(n), t_{avoid}(n)) \leftarrow GetVoronoiEdge(X_{near}, X_c)$

$X_{obs} \leftarrow RemovePointSet(X_{obs}, E(n))$

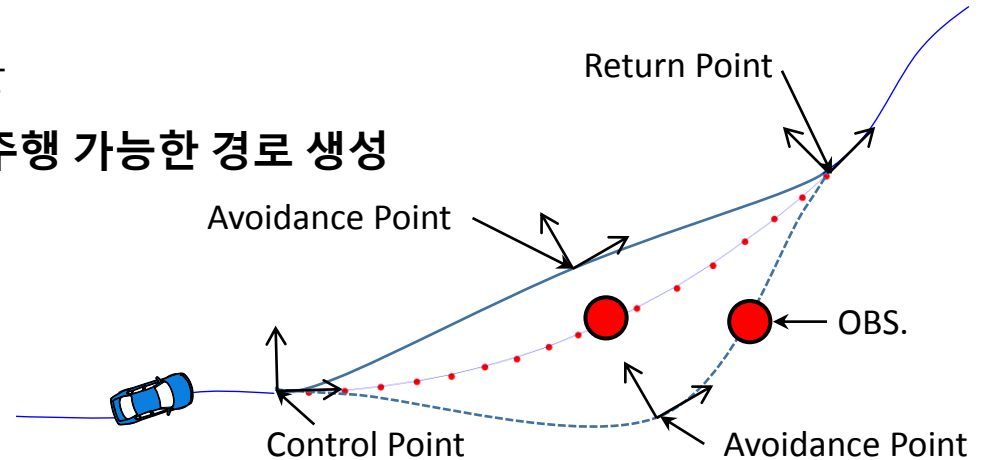
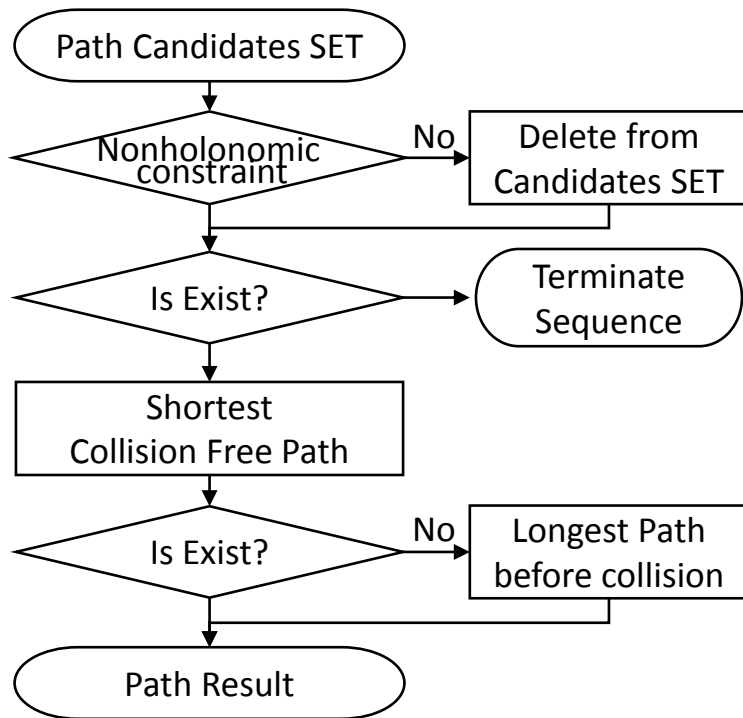
while ( $X_{obs} \neq Null$ )

return ( $X_{avoid}, t_{avoid}$ )

### Path Generation

One Voronoi Cell Algorithm 으로부터 구한

Avoidance points 후보들로 부터 차량이 주행 가능한 경로 생성



경로 선정 조건

- 차량이 주행 가능한 경로중
- 조건 1. 충돌하지 않는 가장 짧은 경로를 선택
- 조건 2. 충돌하기까지 가장 긴 경로를 선택
- 차량이 주행 가능한 경로가 없는 경우
- ➔ 충돌 전까지 기존 경로를 따라 주행 (경로 생성이 안된 경우를 대비)
- ➔ 충돌 전까지 주행해도 경로 생성이 안되는 경우 후진 하면서 경로 생성

### Path Representation

모든 경로는 Path Segment의 Array로 구성  
 각각의 Path Segment는 **3차 방정식**으로 이뤄짐.

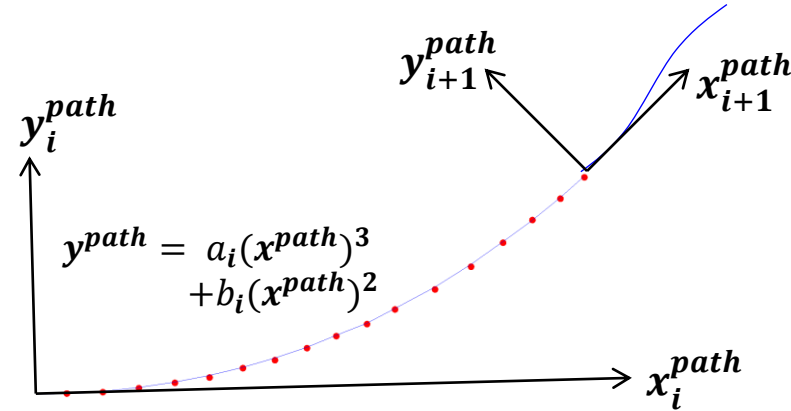
이점 :

1. 3차방정식으로 구성  
 → 양끝의 **경계조건(위치, 기울기)** 만으로 생성 가능  
 → Coefficients를 제한하여 **Nonholonomic 조건부합 가능**

2. 매개변수를 이용한 표현법 보다 직관적임

3. WPT를 이용한 경로 관리보다 효율적임

- 경로의 연속성으로 제어 및 장애물 검사가 연속적임
- 데이터 저장의 효율성 증대
- 경로 수정 용의



$$P_i = [x_i^{ENU}, y_i^{ENU}, h_i^{ENU}, a_i, b_i, x_{end_i}]^T$$

- $x_i^{ENU}, y_i^{ENU}$  Path Segment Start Point's Pos. in ENU Frame.
- $h_i^{ENU}$  Path Segment Start Point's Heading in ENU Frame.
- $a_i, b_i$  Cubic Function Coefficients.
- $x_{end_i}$  Path Segment End Point's x value in Path Frame.

### Collision Detection

#### 1. 경로 위 장애물 검출

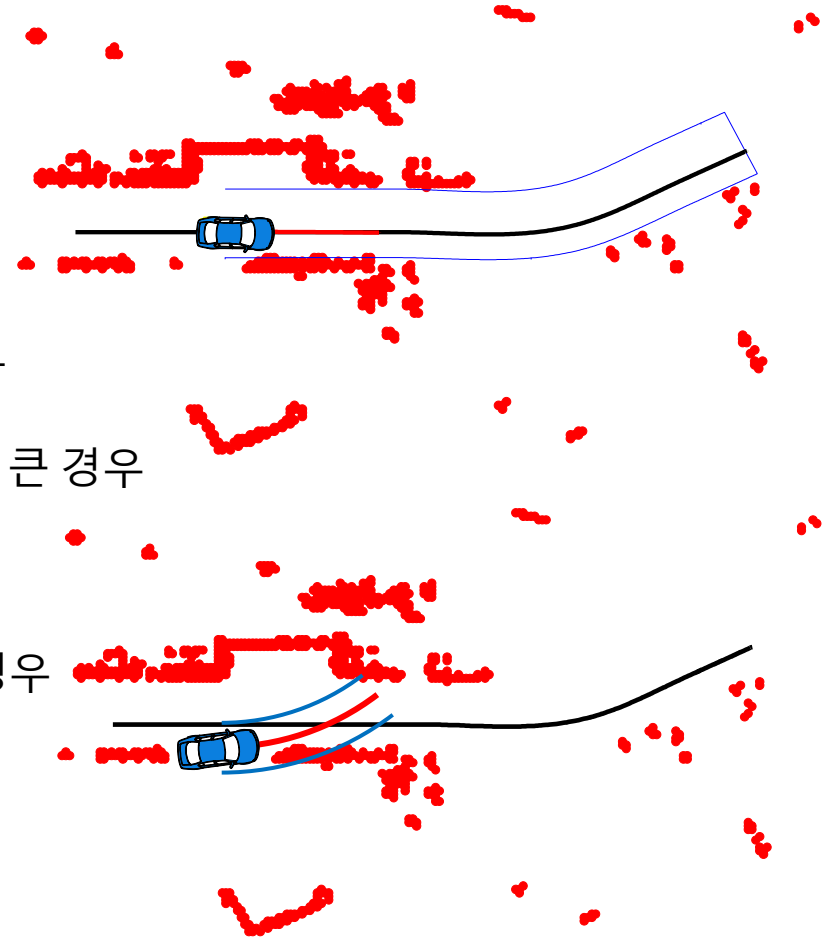
- 주어진 경로 위의 장애물 검출  
3차방정식으로 이루어진 경로를  
좌우로 벌려 충돌 감지

이점 : 단시간 내에 주행 가능 영역을 판별  
경로를 이용하여 연속적인 판별 가능

단점 : 차량 위치와 경로사이의 거리 오차가 큰 경우

#### 2. 주행경로 위 장애물 검출

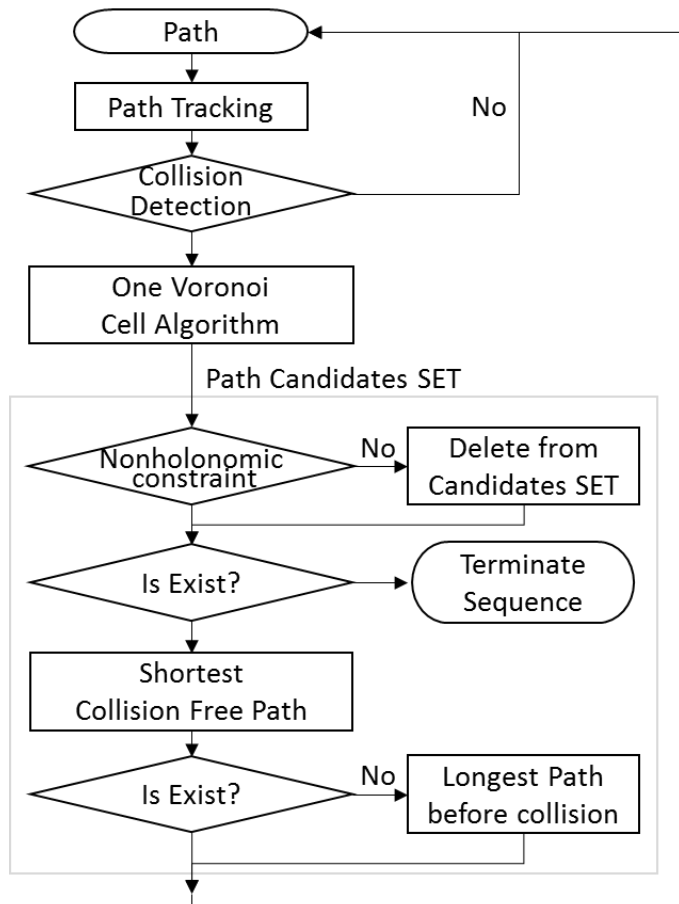
- 차량 위치와 경로사이의 거리 오차가 큰 경우  
차량의 현재 속도와 Steering 각을 이용해  
차량의 이동 궤적을 예측  
이동 궤적 상의 장애물 판별



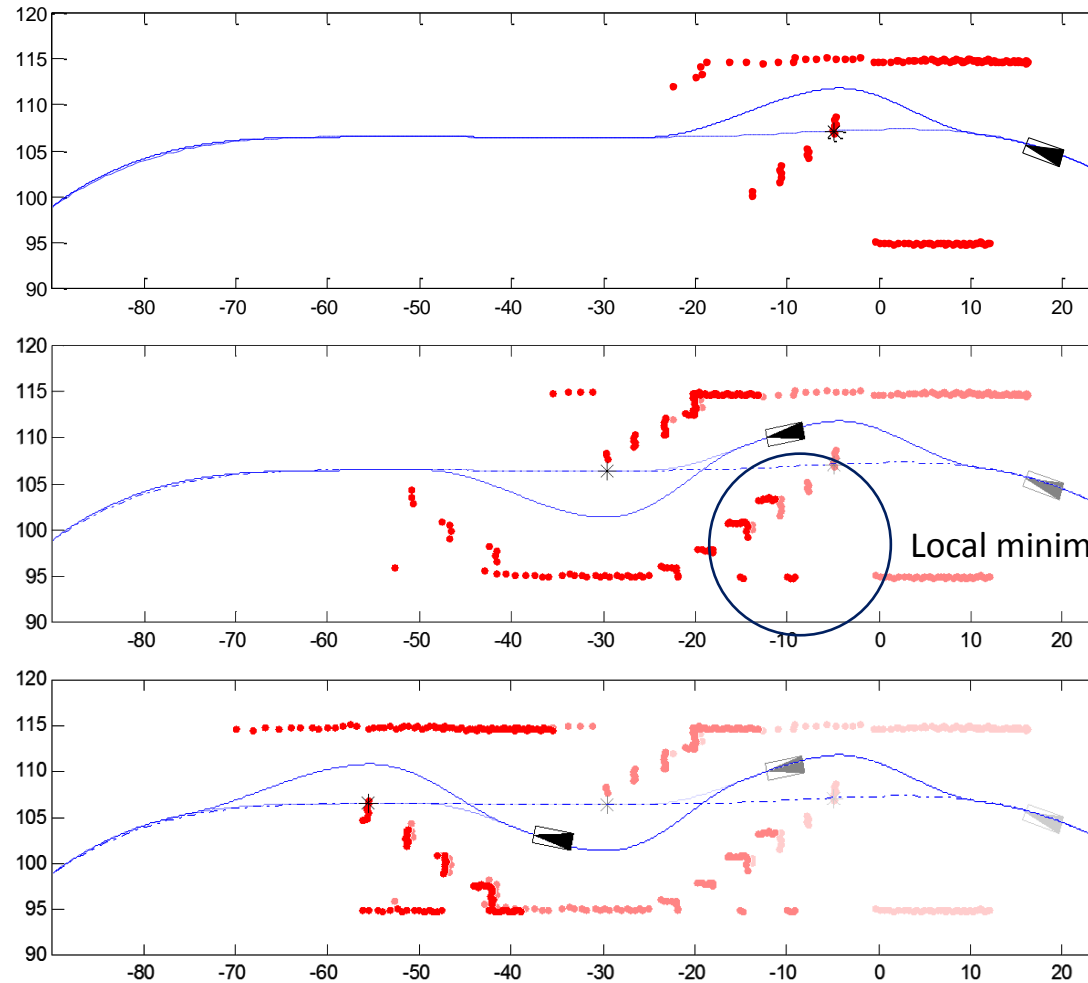


## Path Planning Sequence & Simulation Result

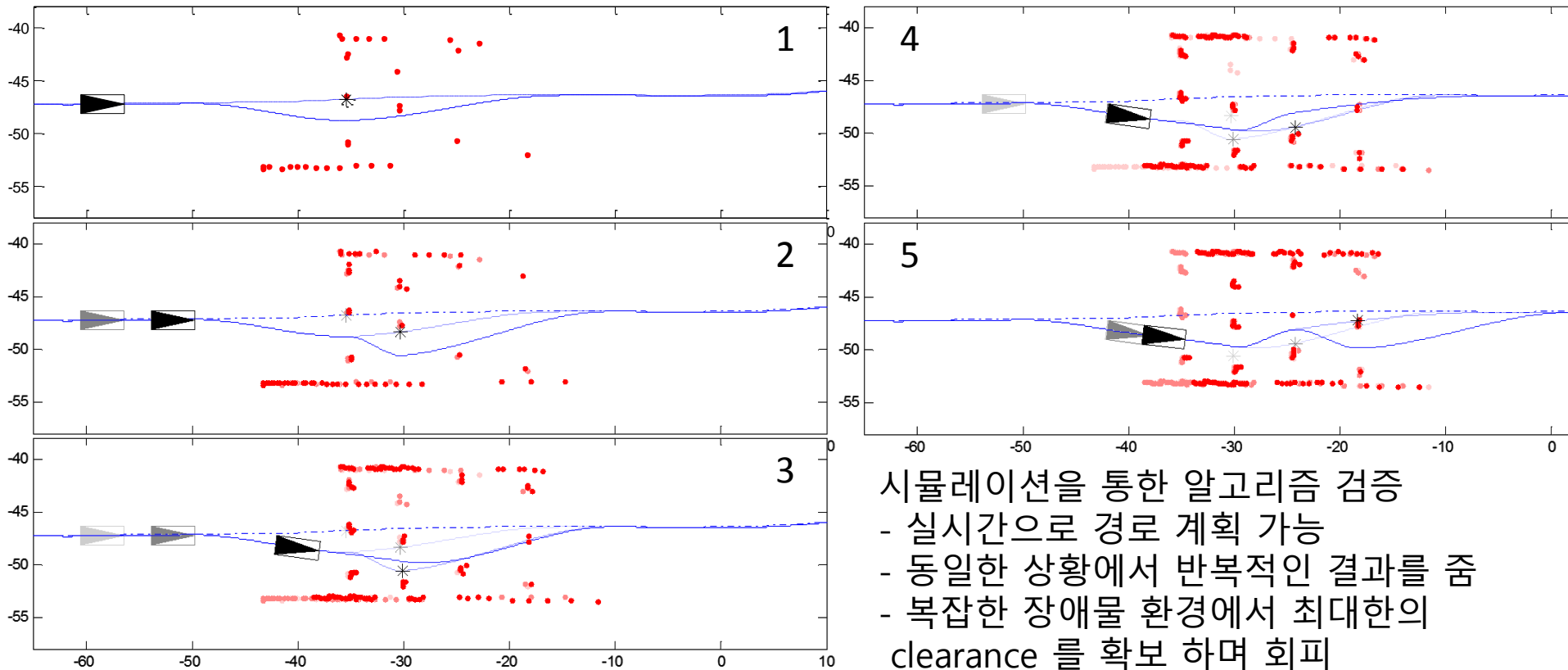
### Path Planning Sequence



### Simulation Result Case 1



### Simulation Result Case 2



시뮬레이션을 통한 알고리즘 검증

- 실시간으로 경로 계획 가능
- 동일한 상황에서 반복적인 결과를 줌
- 복잡한 장애물 환경에서 최대한의 clearance 를 확보 하며 회피

## 자율 주행 차량 개발

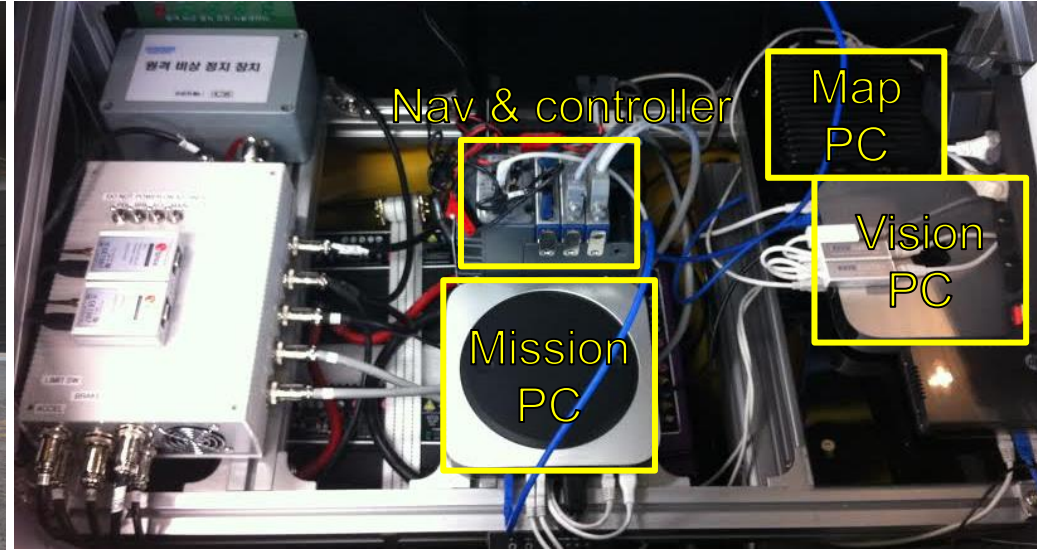
### EURECAR TURBO – System Configuration

최소한의 센서와 저가의 GPS를 이용한 자율 주행 시스템의 센서 구성



EureCar Turbo 센서 및 피씨 구성

single GPS	U-blox EVK-6T (자율주행용)
DGPS	Novatel OEMV-2 (항법 비교용)
Laser Scanner	Velodyne HDL-32E
Nav. & Controller	NI CompactRio
Path Planner PC	Apple Mac mini
Vision PC	Apple Mac mini
Obs. Processing PC	CompuLab Intense PC

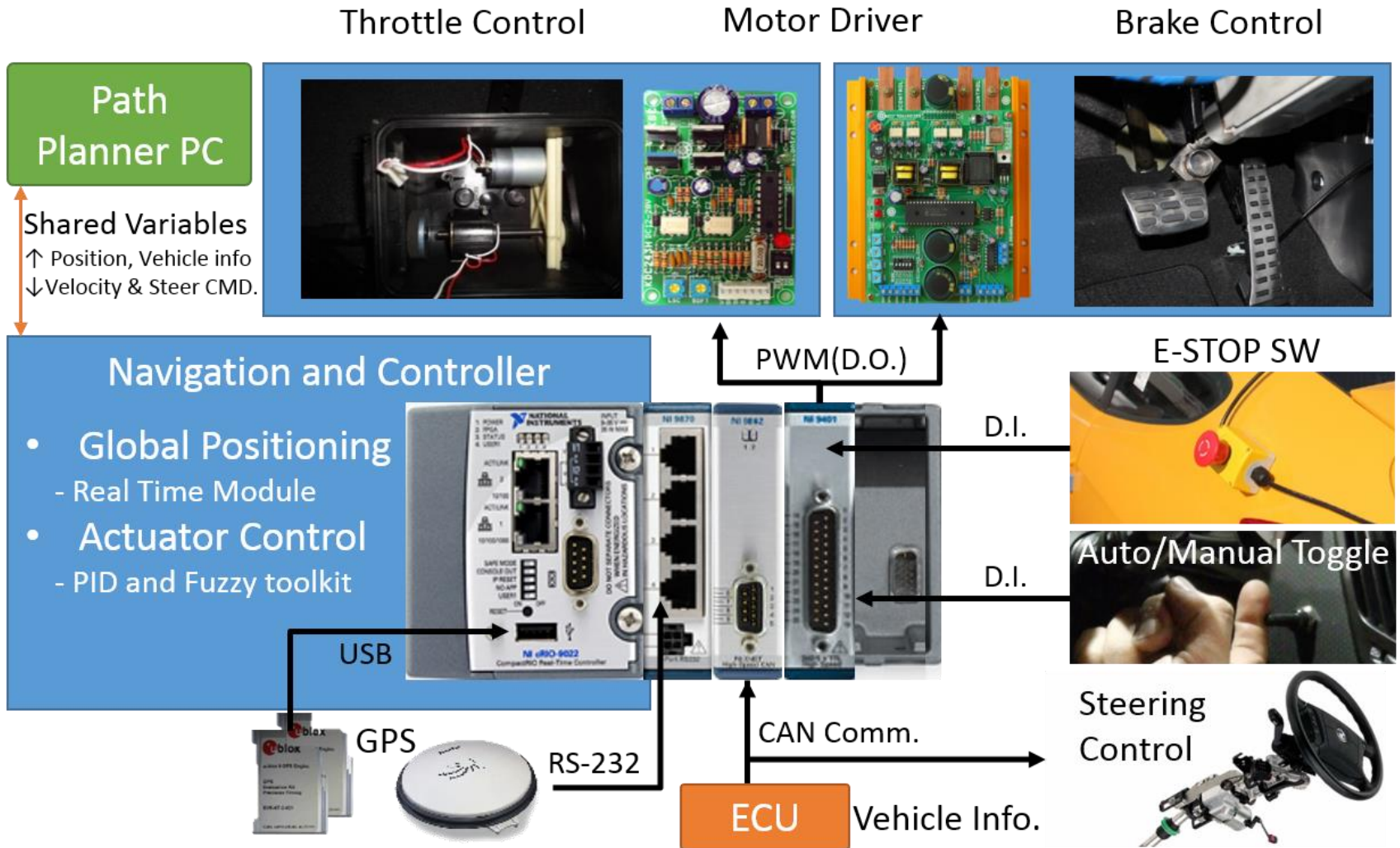


EureCar Turbo 주요 기능

- Path Tracking - Sensor Fusion & GPS  
single GPS + Vehicle ECU(Speed, Yaw rate) → **Position, Heading**  
Laser Scanner → **Road Boundary Detection, Obstacles**  
Camera → **Road Detection**
- Collision avoidance  
Voronoi cell 기반의 실시간 장애물 회피 알고리즘 탑재
- 신호등 횡단보도 보행자 인식/ 풀 시퀀스 직각 자동 주차

# KIDCS C-FRIEND Field Robotic Systems Center

## Development of Self-driving car



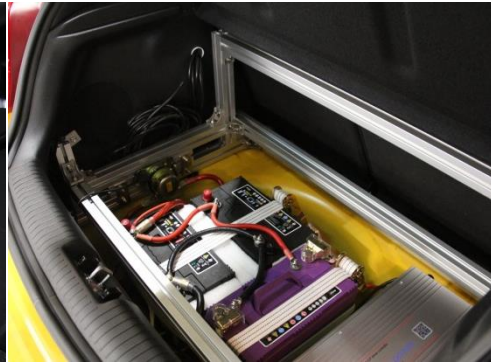
### Generator 추가



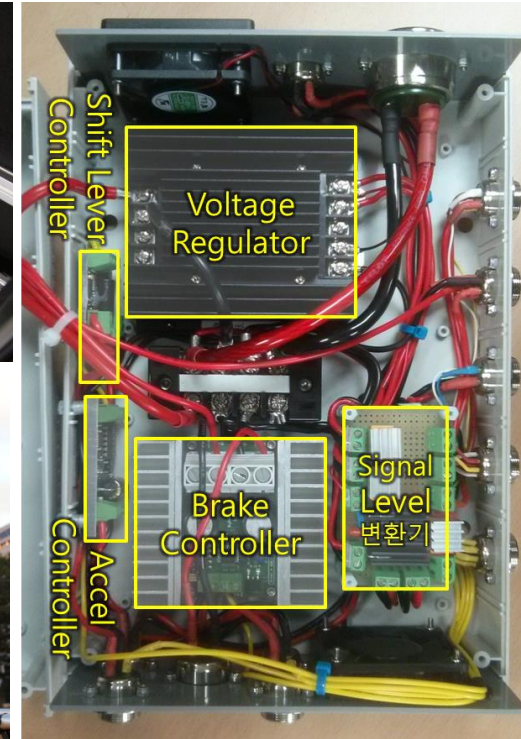
### Brake & Throttle Control Actuator



### Batteries & Inverter



### Actuator Controller



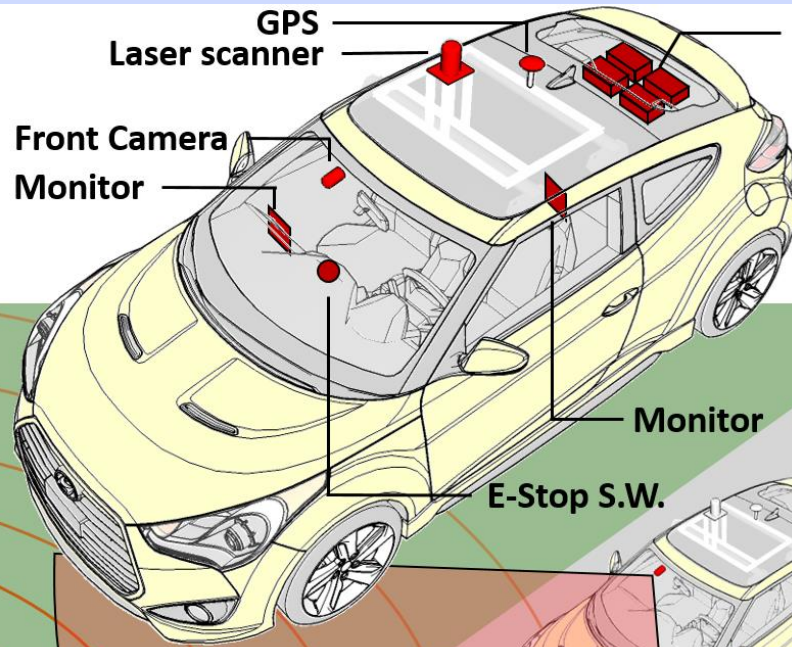
### 비상 정지 스위치

### Low Cost GPS

### 360° Lidar

## KAIST Self-Driving Car, Eurecar

KAIST is developing Self-Driving Car that makes surrounding environment map with various sensor data to steer, speed up and down to a destination.



Processor analyses sensor data to locate the car position and find the non-colliding path.

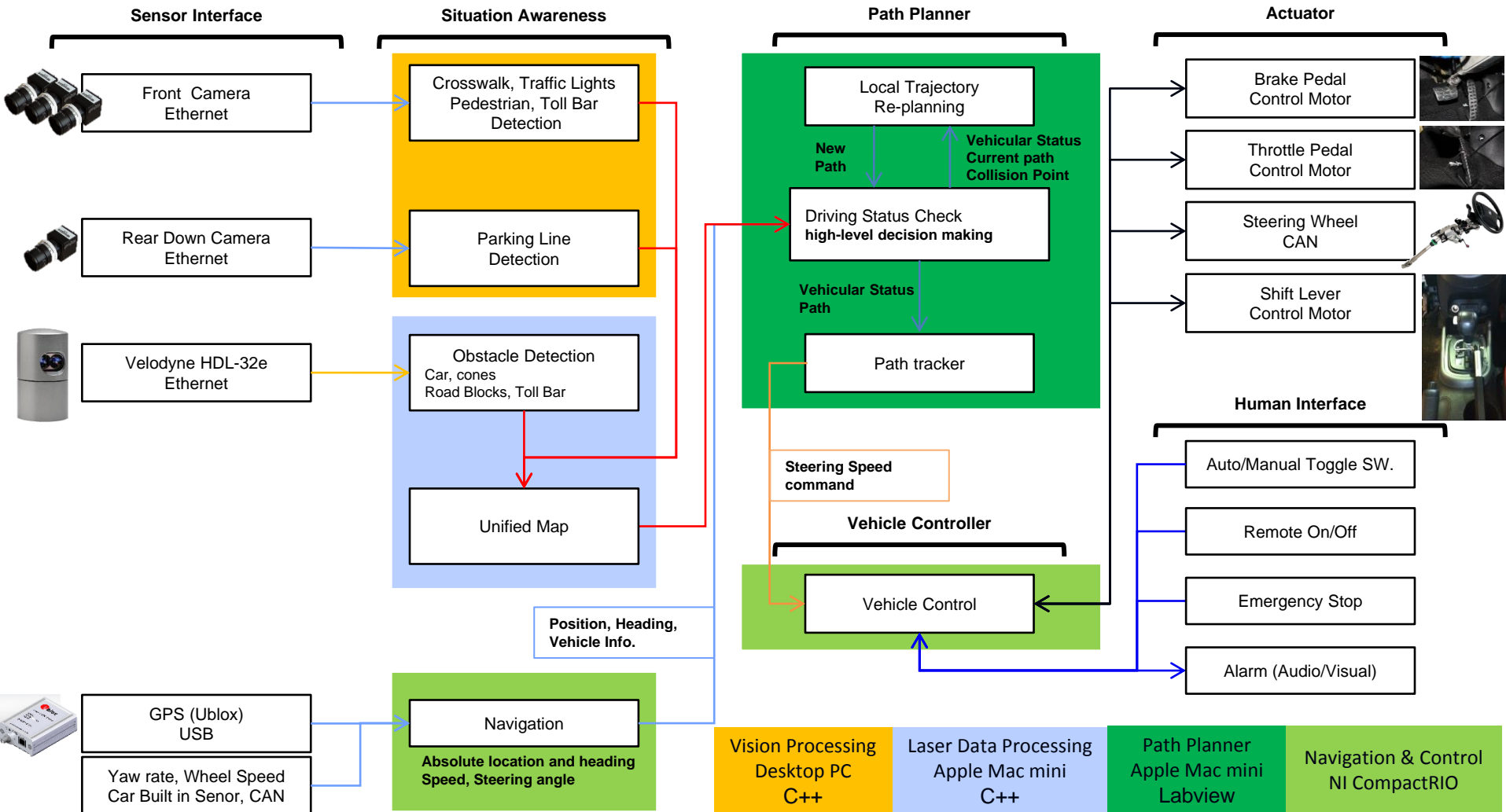
Velodyne HDL-32E  
Horizontal angle : 360deg  
Vertical angle : -30~10deg  
Range : ~70m

**Laser Scanner**  
detects car and other road blocking obstacles. It has 360-degree view and up to 75m detection range

**Front Camera**  
detects lane, crosswalk, traffic signs, pedestrians and other objects

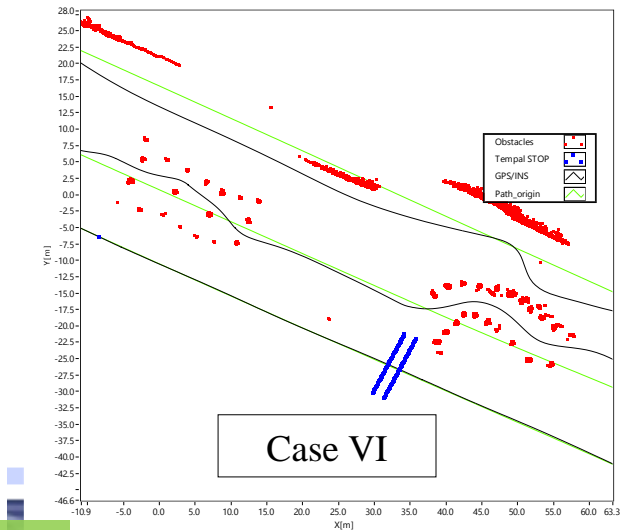
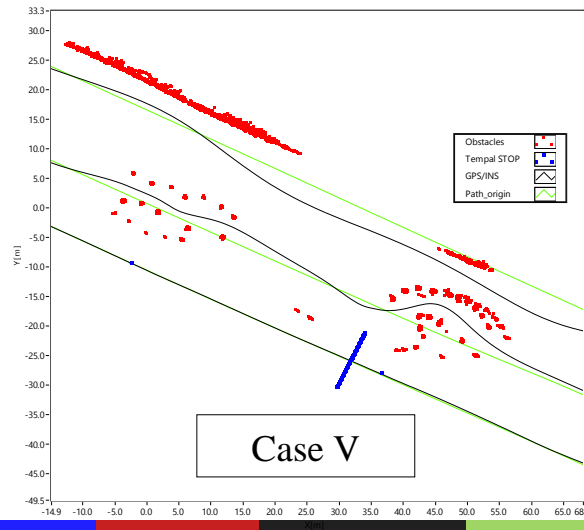
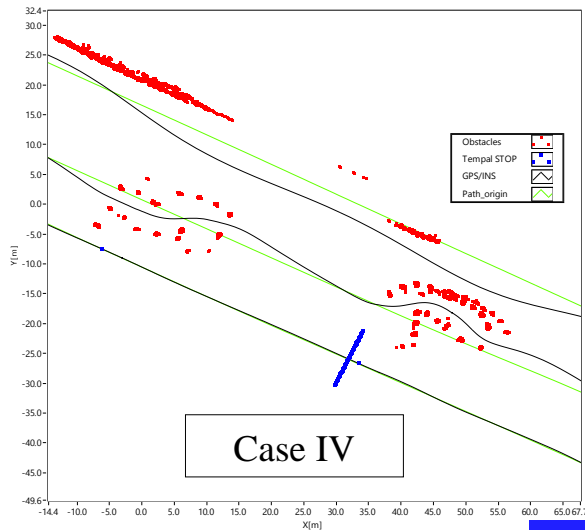
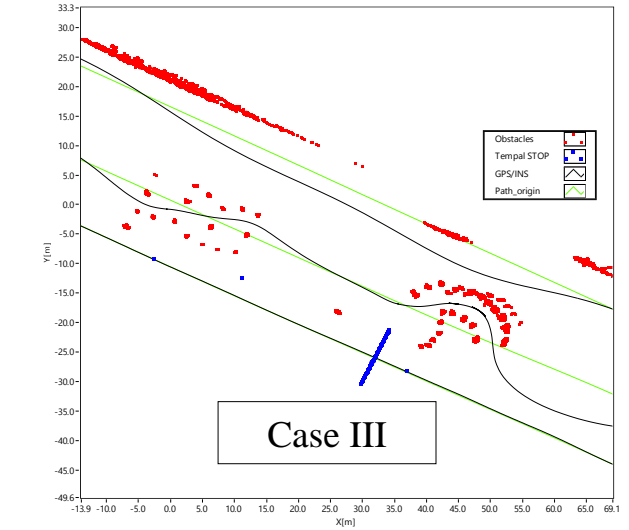
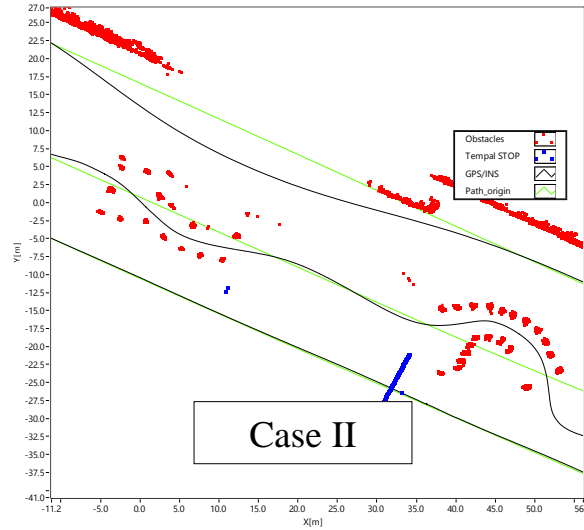
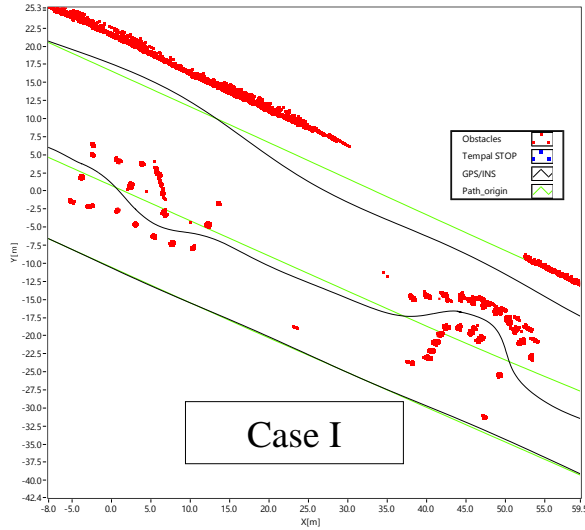
# KIDCS C-FRIEND Field Robotic Systems Center

## System Configurations



## The experiment result

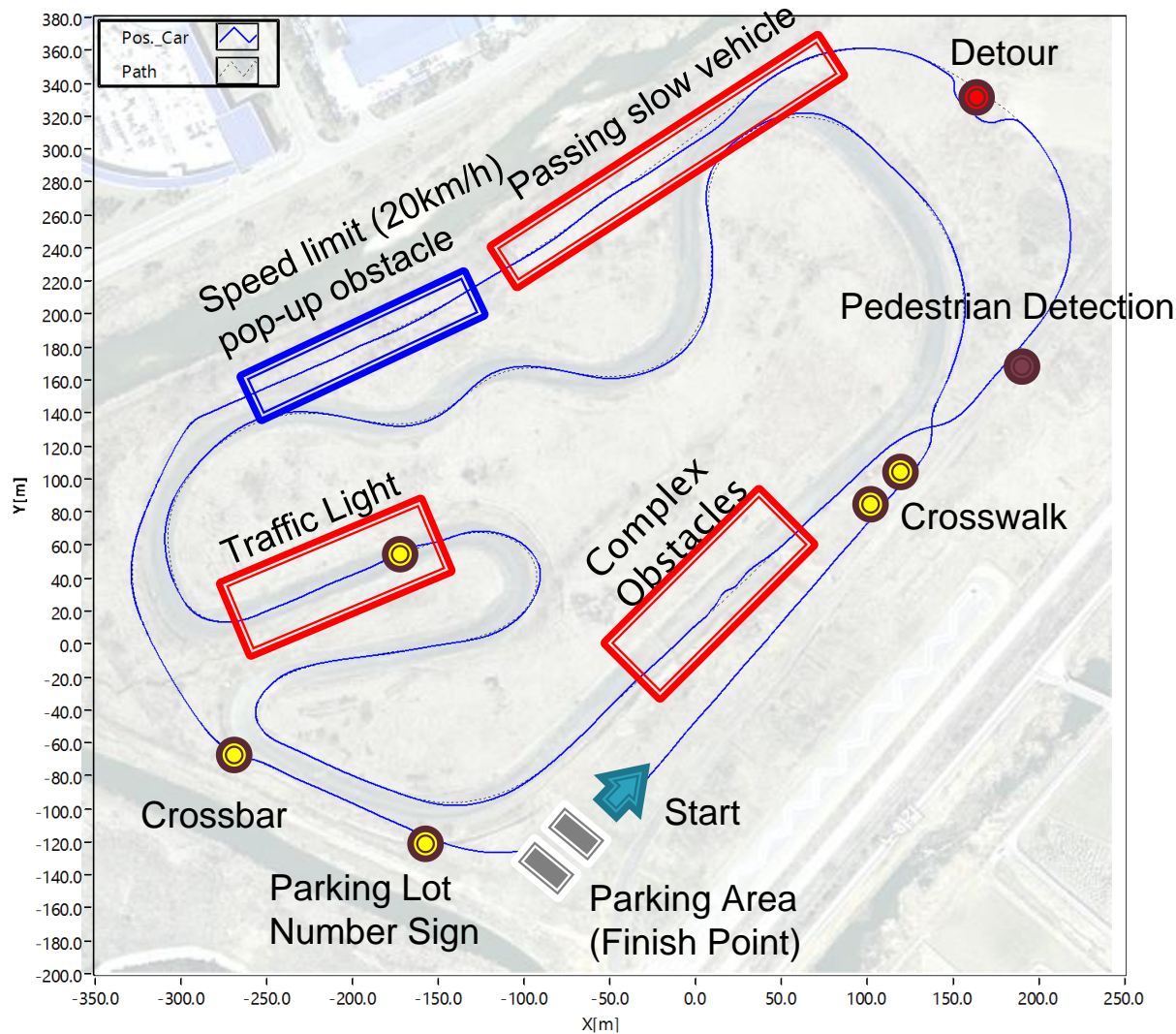
- Obstacle avoidance result – 다수의 Case에 관하여 알고리즘의 반복성 테스트





# KIDCS C-FRIEND Field Robotic Systems Center

## “Hyundai Challenge” Competition-at-a-glance



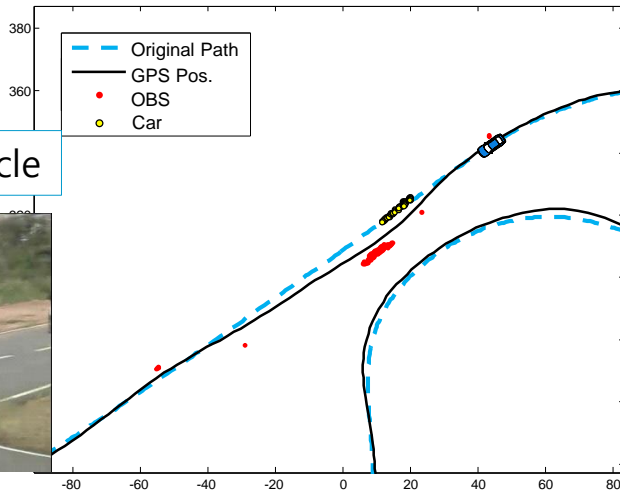
Mission	Result
Crosswalk detection	Success
Traffic light	Success
Complex obstacles	Success (zero cone)
Pedestrian Detection	Failure
Detour	Success (zero cone)
Passing slow vehicle	Success
Pop-up obstacle	Success
Crossbar Detection	Success
Parking	Success (rearward)

Speed [km/h]	
Ave.	22.43
Max.	51.75
time	9min 07sec

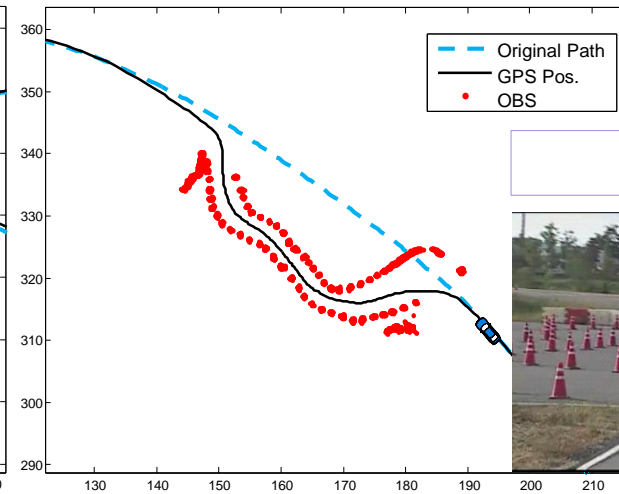
## Path Planning Result In Competition

- 대회에서의 장애물 회피 결과 - 다양한 미션에 관하여 완결성 있는 결과 도출

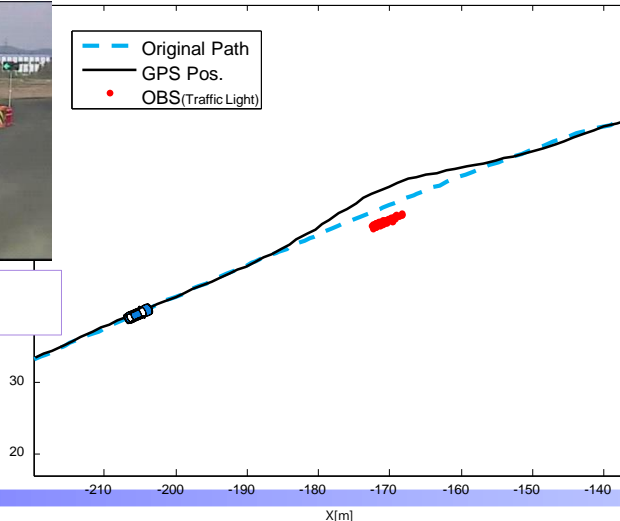
Passing slow vehicle



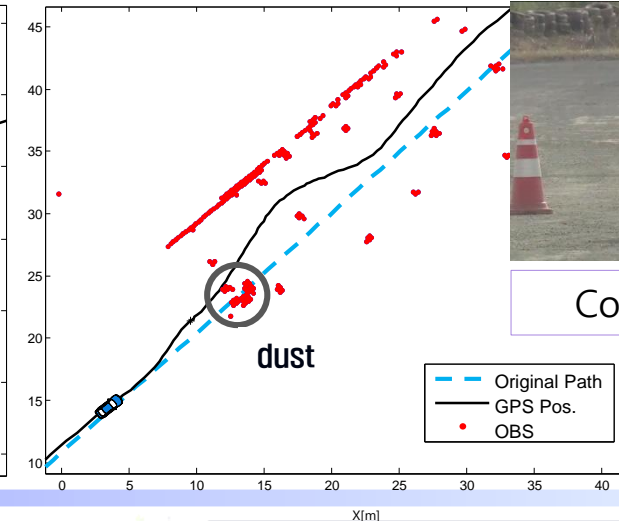
Detour



Traffic Light



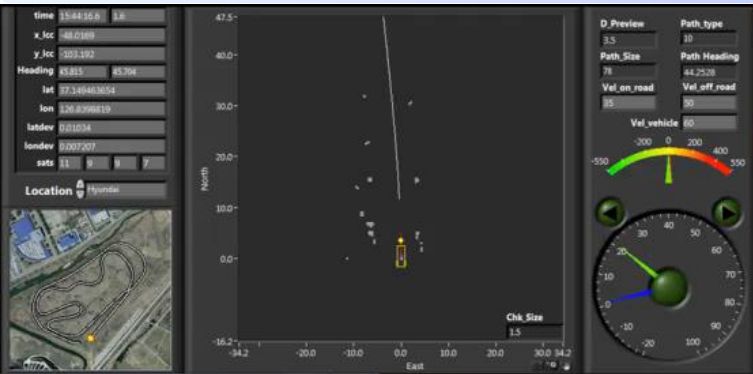
dust



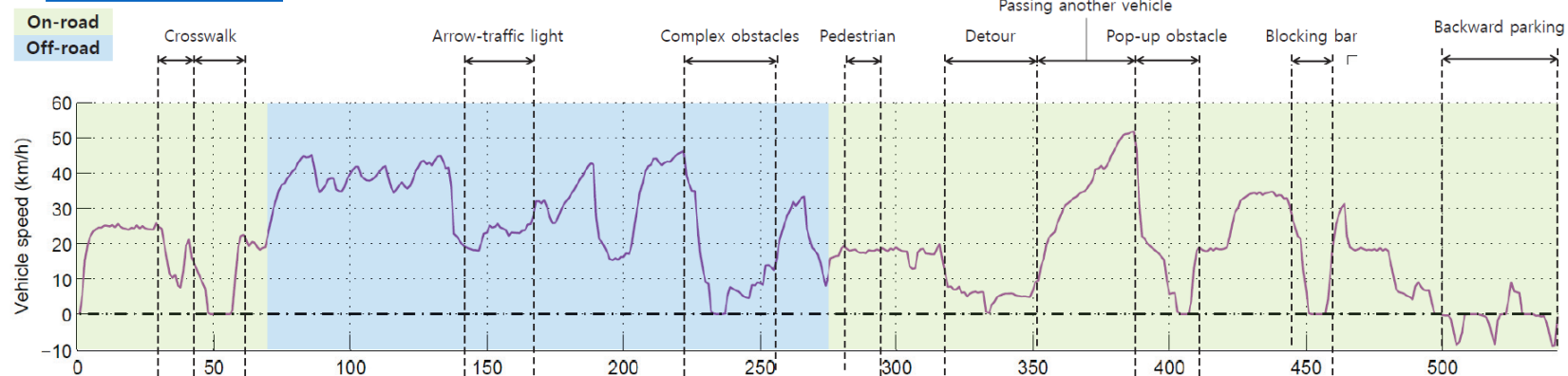
Complex Obstacles

# KIDCS C-FRIEND Field Robotic Systems Center

## Competition Video



**USRGTube** <https://www.youtube.com/user/USRGTube>



### 자율 주행 차량 개발 사항

- Single GPS와 Odometer를 이용한 주행
- 캠퍼스 내의 완전 자율 주행
- 장애물 회피 주행
- 고속 주행

### 차후 Application

- 무인택시, 교내 셔틀
- 다양한 실제 도로 환경에서 자율 주행

