

2024 HADA Technical Report

School of Mechanical and Control Engineering
Handong Global University

곽진, 최윤석, 이찬근, 서현규, 최정빈, 김희원, 전세인, 이다원, 김유민

HADA 4

2024.11.29

차 례

I. 서론	2
1.1. 연구목표	2
1.2. 연구 필요성	2
1.3. 개발 환경	2
II. 알고리즘 및 설계 정리	7
2.1. 전역경로계획	7
2.2. 지역경로계획	12
2.3. 미션 알고리즘	14
2.4. 하드웨어	31
III. 문제점 및 해결 방안	34
IV. 결론	38
참고문헌	39
부록	39

I. 서론

1.1. 연구 배경

모델 기반 자율주행 기법은 이미 항공우주공학에서 주된 논제였으므로 상당수의 기법이 존재한다. HADA 팀은 자율 이동체의 위치 및 자세 정보를 갖고 유도 조종 기법을 적용해 원하는 목적지에 도달하는 전역경로계획을 핵심 알고리즘으로 한다. 돌발 상황, 주행 환경에 대한 즉각적인 대처를 위해 카메라 및 라이다 센서를 이용해 주변 상황을 인지하며, 식별된 정보를 이용해 주변 환경과 상호작용을 통해 미션을 수행하고, 전역경로계획의 최종 목표점까지 안전하게 도달한다.

또다른 자율주행의 핵심 키워드는 실시간성이다. 실시간성을 확보한 시스템에서는 빠른 상황 인지 및 의사 결정, 제어를 구현할 수 있다. 실시간성을 확보한 시스템은 일정한 샘플링주기로 인해 주파수 영역에서의 정량적인 해석 또한 가능케한다. 자율주행은 전역경로계획, 지역경로계획, 주변환경과의 상호작용으로 인해 조향각, 속력 등은 상시 변할 수 있다. 샘플링에서 발생할 수 있는 문제를 최대한 억제하며 상시 변이하는 태스크시간에 대한 실시간 불확실성을 Time-idling 을 통해 예방한다. HADA 팀은 알고리즘, 함수 등으로 나뉜 시스템에 대해 Loopback test 를 진행하며 이를 통해 샘플링 주파수를 적절히 설정한다.

1.2. 연구 필요성

HADA 동아리는 2010 년도 이후 본격적으로 연구되고 있는 자율주행자동차 기술에 대한 연구를 하는 연구 동아리이다. 본 보고서는 2024 년도 HADA 4 기 기술 보고서이며 HADA 4 기의 알고리즘, 하드웨어 및 향후 HADA 의 개선 사항 등을 소개해, 4 기 이후의 HADA 인원들의 빠른 지식 습득 및 추후의 개발에 차질을 미리 예방한다. 현재까지 HADA 에서 주된 알고리즘을 인수인계하는 방식으로 진행되어왔으나 문서화가 되어있는 부분은 많지 않다. 문서화의 부재는 해를 거듭하여 HADA 의 연구 개발 능력 향상에 상당한 어려움을 겪고 있다. 따라서 해당 보고서를 통해 향후 HADA 팀의 알고리즘을 문서화하고, 정리된 알고리즘을 바탕으로 지도교수인 나원상 교수님과 소통하며 이를 통해 자율주행 연구에서 발전할 수 있는 기회를 제공한다.

1.3. 개발 환경

HADA팀이 사용하는 차량은 HADA 팀이 2024년도 4월부터 8월까지 자체 제작한 차량이며 기존에 사용하던 ERP42와는 차별점을 둔다. 자체 제작 플랫폼의 가장 큰 장점으로 차량 외 하드웨어/센서 등을 배치할 수 있는 공간이 커졌다는 것이다. 축간 거리, 전폭 등 차량 제원은 HADA팀이 매년 참가하는 대학생 창작 자동차 경진대회 무인 모빌리티 부문에 맞게 설계되어 있으며, 차량 제원은 Table 1과 같다.

Table 1 차량 제원

구분	제원	보기
플랫폼 중량	약 160 [kgf]	
플랫폼 최대 길이	1885 [mm]	
플랫폼 최대 폭	1070 [mm]	
플랫폼 최대 높이	1017 [mm]	
플랫폼 축간 거리	1150 [mm]	
조향 모터 최대 RPM	80 [RPM]	
조향 모터 최대 Torque	9 [Nm]	
조향 모터 동작 전압	12 [V]	
구동 모터 최대 RPM	300 [RPM]	
구동 모터 최대 Torque	50 [Nm]	
구동 모터 동작 전압	24 [V]	
배터리 #1	24[V], 20[A]	최대 전력 500[W]
배터리 #2	24[V], 20[A]	최대 전력 500[W]

자율 주행 플랫폼에 장착된 센서 목록은 Table 2와 같다. 카메라 외 센서는 HADA 1기부터 사용되었으며, 컴퓨터는 딥러닝 및 연산량을 고려해 2대를 사용하였다. 컴퓨터 간 통신은 UDP/IP 통신으로, 서버/클라이언트 간 송수신을 통하여, 통신량, 통신 지연 시간을 최소화하였다.

Table 2 센서 제원

구분	제원	센서 사용 목적
LiDAR #1 (VLP-16 PUCK)	길이: 122 (mm) 폭: 122 (mm) 높이: 149 (mm)	
LiDAR #2 (SICK LMS511)	길이: 165 (mm) 폭: 155 (mm) 높이: 185 (mm)	장애물 감지
Camera #1 (iNavi MC10)	길이: 68 (mm) 폭: 34 (mm) 높이: 26.5 (mm)	객체 검출
Camera #2 (iNavi MC10)	길이: 68 (mm) 폭: 34 (mm) 높이: 26.5 (mm)	차선 검출

IMU (Xsens MTi-600 계열)	길이: 31.5 (mm) 폭: 25.4(mm) 높이: 13.0(mm)	
GPS (NovAtel OEM 7720)	길이: 46 (mm) 폭: 71 (mm) 높이: 11 (mm)	
GPS Antenna (NovAtel GNSS-850)	길이: 176 (mm) 폭: 176 (mm) 높이: 55 (mm)	
PC #1		메인 PC
PC #2		객체 인지 용 PC
Controller	STM32F411RE	제어용 MCU

HADA 팀에서 센서 배치를 고려하여 자율 주행 플랫폼 하드웨어를 설계하였다. 자율주행차량은 센서로부터 환경을 인지하고 이를 바탕으로 판단 과정을 거치며 제어된다. 센서 신호 처리와 센서 융합이 자율주행차량의 기초이자 핵심이기에 자율주행에 필요한 데이터를 선정 및 이를 얻기 위한 센서의 적절한 위치선정이 필수적이다. 플랫폼에 장착되는 센서 위치는 Figure 1에 도시하였다.

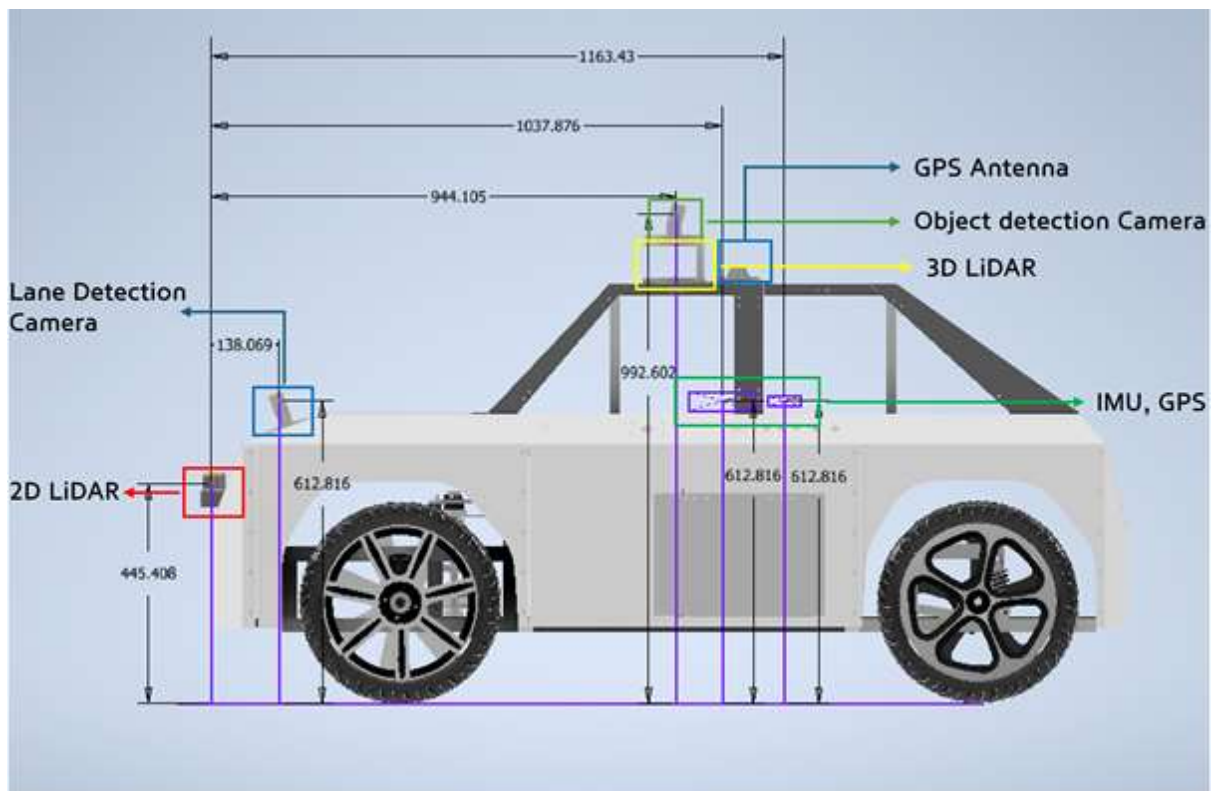


Figure 1 하드웨어 아키텍처

IMU는 관성 항법 필터에 쓰이는 동체의 각도, 각속도 데이터와 가속도 데이터를 얻

을 수 있다. 차량 동체 좌표계 원점 기준으로 동체가 받는 가속도, 행하는 각속도 및 자세각을 고려해야 하기 때문에 IMU의 위치는 동체의 중앙 부근에 고정되어 Strapdown-INS를 수행한다. 차량의 가속/감속, 속도가 매우 크지 않고 자세각 변화율 또한 크지 않기 때문에 센서의 위치가 차량 무게 중심에서 이동하여도 문제가 없다. 따라서 IMU 센서를 차체 중심 뒤에 거리를 띄워 장착하였다. GPS의 안테나는 위성으로부터 거리 정보를 받아온다. 안테나는 위성 신호를 받으며 신호를 안정적으로 받는데에 여러 제약상황이 있으므로 차체의 최대한 높은 곳에 장착하였다. GPS 수신기는 위치선정에 있어 특별한 제약이 없으므로 안테나와 연결이 용이한 곳에 장착하였다. LiDAR의 경우, 2D 라이다와 16채널 3D 라이다를 각각 하나씩 사용하였다. LiDAR 센서는 레이저를 발산하여 돌아오는 시간을 측정하는 TOF 시스템을 기반이기 때문에, 차체 다른 센서 및 하드웨어의 간섭을 받지 않도록 적절한 위치를 선정하는 것이 중요하다. 그에 따라, 2D LiDAR 센서의 경우, 플랫폼 가장 앞부분에 장착을 하였고, 3D LiDAR 센서는 플랫폼의 차체 위에 GPS 안테나 전면부에 설치하였다. 카메라 센서는 총 2대를 사용하였다. 카메라 센서의 내부 파라미터를 사용해 왜곡을 보정한 후 사용된다. 카메라 1대는 Bird's eye view(BEV) 알고리즘을 통해 차선 또는 정지선을 인지하여 차량이 GPS 음영 구간과 같은 전역경로 계획이 불가능한 시점에서 차선 기반 운행을 할 수 있도록 사용되었다. 나머지 1대는, 높은 위치에 장착해 표지판, 신호등, 장애물 등 객체를 인지하는데 활용하였다. 3D 라이다와 카메라의 융합을 고려하여 3D 라이다 위에 장착하였다. HADA팀이 자체 제작한 플랫폼을 제어하기 위해 MCU 1대가 사용되었다. 이는 조향 모터 제어 및 구동 모터 제어를 담당하고 있으며, 컨트롤러와 통신을 통해 수동으로 조작 가능하도록 설정되었다.

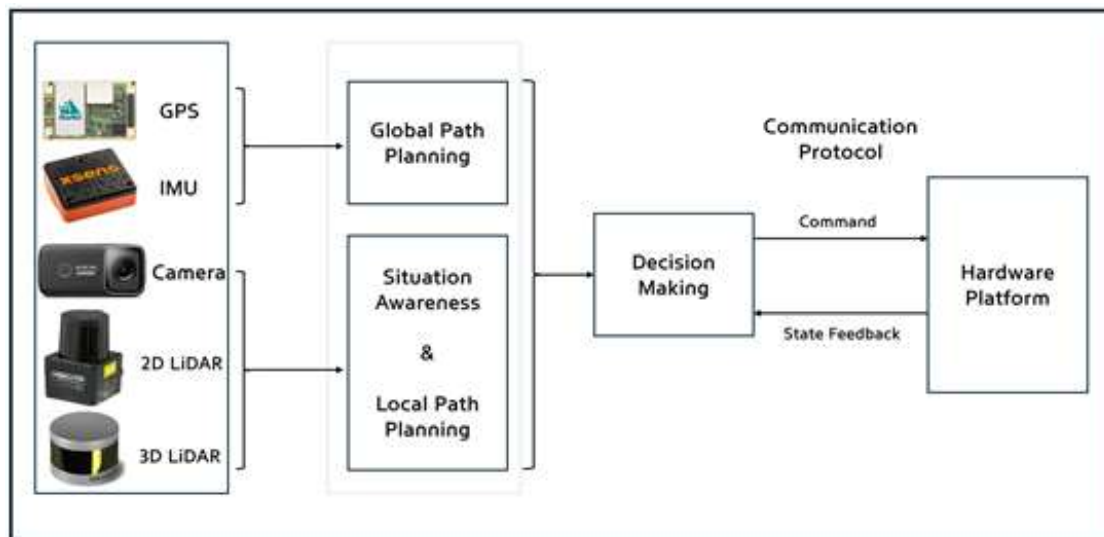


Figure 2 소프트웨어 아키텍처

HADA팀은 인지-판단-제어로 이뤄진 소프트웨어를 실시간으로 처리한다. 실시간 성능을 확보하기 위해 전체 프로세스는 C언어에 기반하였으며 카메라 영상 처리 및 딥러닝을 수행하기 위하여 C++언어를 추가로 사용하였다. HADA 4기 이전에 공유 메모리를 통해 센서의 정보를 다른 프로세스에서 취득하여 메인 프로세스로 전달한 반면, 올해부터 프로세스 내에서 센서로부터 정보를 취득하는 작업을 멀티 쓰레딩을 통하여 구현하였

다. 따라서 이를 통해 추가적인 실시간 성능 및 용이성 등을 확보하였다. 프로그래밍 언어 버전 및 빌드 환경, 통합 개발 환경(IDE)는 Table 3과 같다.

Table 3 소프트웨어 개발 환경

툴체인	Clang-cl
빌드 도구	Ninja/ Visual Studio
IDE	Clion
C 언어	17
C++ 언어	17
CUDA	12.1
TensorRT	10.2
OpenCV	4.10.0

HADA 에서 사용하는 센서의 샘플링 주파수 및 성능은 Table 4 로 정리하였다.

Table 4 센서/프로세스 별 샘플링 주파수 및 성능

Sensor/Process	Sampling Frequency [Hz]	Performance
GPS	10	RTK
IMU	100	
2D LiDAR	25	0.25°
3D LiDAR	10	좌우 분해능: 0.2° 상하 분해능: 2.0°(-15°~15°)
Camera	60	대각 시야각 142°, 해상도: FHD(1080)
Main Process	250	MCU 실시간 제어기 성능 확보

II. 알고리즘 및 설계 정리

2.1. 전역경로계획

자율 이동체에 유도조종기법을 적용하기 위해 가장 먼저 좌표계 정의를 하였다. 좌표계 정의 및 좌표계 간 관계 정의는 GPS, IMU 등 각 센서 별 받아오는 정보의 좌표계가 다르기 때문에 이를 하나로 통일하기 위한 작업을 거친다. 좌표계 변환 수식은 전역 경로계획 뿐 아니라, 센서 간 관계, 카메라 좌표계 변환 등에서 여러 형태로 쓰이게 된다. 좌표계 변환 방법으로는 오일러각, 쿼터니언, 방향 코사인 행렬(DCM)이 있으며, HADA팀에서는 DCM을 사용하여 좌표계 변환을 수행한다. DCM의 수식은 직교 좌표계에서 각 축에 대한 회전 행렬로 이뤄지며, 왜대칭 행렬(Skew-symmetric matrix)이라는 특성을 갖는다. 왜대칭행렬의 역행렬은 전치행렬로 보다 쉽게 구할 수 있으며 그 의미는 역 방향으로 좌표계 변환이다. 1에서 2로 좌표변환행렬 DCM의 표기는 C_1^2 와 같이 쓰이며 회전행렬로 이뤄진다. 좌표변환의 편의성을 위하여 모든 좌표계는 오른손 좌표계로 통일한다.

$$C_1^2 = R_x(\alpha)R_y(\beta)R_z(\gamma)$$

$$C_1^{2^{-1}} = C_1^{2T} = C_2^1$$

각 좌표계 별 상관관계는 Figure 3 에 도시하였다.

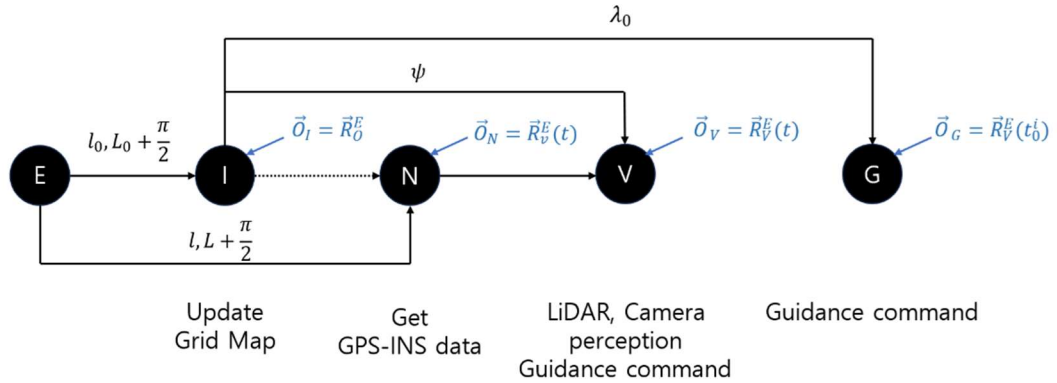


Figure 3 좌표계간 관계

- 1) 지구 좌표계, Earth Frame (E-Frame): 지구의 중심을 원점으로 하며, X축은 적도평면과 본초자오선의 교차점 방향, Z축은 지구 중심에서 북극점을 향하는 벡터로 표현하는 오른손 좌표계이다.
- 2) 항법 좌표계, Navigation Frame(N-Frame): 지구에서 자율이동체의 위치를 원점으로 WGS 84 지구 모델의 접평면이 X-Y축으로 이뤄진 평면이다. X축은 정북 방향, Y축은 정동 방향, Z축은 해당 접평면에 지구 원점 방향으로 직각하다.
- 3) 기준 항법 좌표계, Reference Navigation Frame: 자율이동체의 움직임이 표현되는 고정 좌표계로서, 전역경로계획의 중심이 되는 관성 좌표계이다. K-City의 경우, 위도

37.2408943374741, 경도 126.773828152913이 기준 항법 좌표계의 원점이 된다.

- 4) 동체 좌표계, Body Frame(B-Frame): 동체의 무게 중심을 원점으로 하고 동체의 전면 방향을 X축, 수직 하 방향을 Z축으로 하는 오른손 좌표계이다.
- 5) 유도 좌표계, Guidance Frame: 유도 좌표계는 자율이동체의 유도 명령이 산출되는 좌표계이다. X축은 최초의 시선각 방향, Z축은 NED 좌표계의 Z 방향과 동일한 오른손 좌표계이다.

Figure 3에 도시하였던 좌표계간의 변환 관계는 다음과 같다.

1) E-Frame(Latitude, Longitude to X_E, Y_E, Z_E)

GPS는 위경도를 출력한다. 따라서 위경도를 지구 좌표계로 변환하는 수식이 필요하다. 이 때, HADA팀은 WGS84 지구 모델을 사용하여 이를 표현한다.

$$R = \frac{a^2}{\sqrt{a^2 \sin^2(L) + b^2 \cos^2(L)}}$$

$$x_E = (R + h) \cos(L) \cos(l)$$

$$y_E = (R + h) \cos(L) \sin(l)$$

$$z_E = (R(1 - f^2) + h) \sin(L)$$

a : 지구 장축 반지름, b : 지구 단축 반지름, R : 곡률 반지름,

L : 위도, l : 경도, h : 고도, f : 평편도

2) E-Frame to N-Frame

지구 좌표계에서 정의된 벡터를 N-Frame에 대하여 정의할 때 쓰인다. GPS에서 받은 위경도 좌표를 통하여, 지구좌표에서 항법 좌표로 변환한다.

$$C_E^N = R_x(0)R_y\left(-L - \frac{\pi}{2}\right)R_z(l)$$

3) N-Frame to B-Frame

N-Frame(NED 좌표계 기준)의 X축, Y축, Z축에 대한 동체 좌표계의 회전각을 구한다. 이는 IMU 센서에서 나오는 자세각 정보로부터 구할 수 있다.

$$C_N^B = R_x(\phi)R_y(\theta)R_z(\psi)$$

4) N-Frame to Guidance Frame

각 경로점에 대하여, 유도 좌표계가 정의된다. 유도 좌표계는 관성 좌표계로 유도 명령이 정의된다. 경로점을 항법 좌표계에서 최초로 바라보는 시선각(Line of Sight Angle, λ)을 통하여 항법 좌표계에서 유도 좌표계로의 변환이 가능하다. HADA팀은 자율이동체가 평면 운동을 한다는 가정하에, Z축에 대한 변환만 수행한다.

$$C_N^G = R_z(\lambda_0)$$

자율이동체 시스템의 주요 목적은 목표점(경로점)까지의 정밀한 경로 계획 및 경로 추정에 그 주안점을 둔다. 해당 목적을 달성하기 위해, GPS와 IMU를 활용해 경로 추종 기법을 설계한다. 경로점까지의 경로 추종 기법은 자율이동체의 유도 제어 알고리즘 문제로 정의할 수 있다. 목표점까지의 유도제어 알고리즘은 다음 특성을 만족하는 유도 기법을 설계하는 것으로 귀결된다.

- 1) 동체와 목표점의 베어링 각도가 0° 로 수렴한다.
- 2) 동체와 목표점까지의 상대거리가 $0 [m]$ 로 수렴한다.

자율이동체와 목표점 간 상관관계를 Figure 4에 있는 상대기하(교전기하)로 도식화하고 각 기호는 Table 5에 나타내었다. HADA에서 사용하는 자율이동체는 차량의 속도가 $20 km/h$ 이하이며, 타이어의 슬립이 작다고 판단하여, 받음각은 무시하였다.

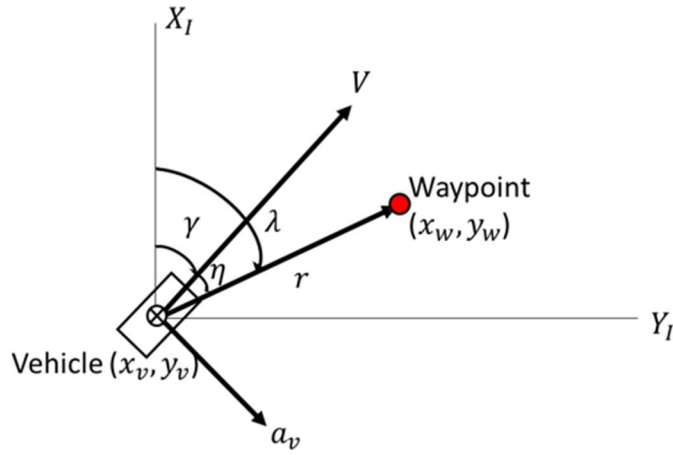


Figure 4 상대기하

아래 변수 정의를 사용해 상대기하에 대한 수식을 전개할 수 있다. 먼저 상대거리 및 시선각, 차량의 경로각을 정의한다.

$$r = \sqrt{(x_w - x_v)^2 + (y_w - y_v)^2}$$

$$\lambda = \tan^{-1} \left(\frac{y_w - y_v}{x_w - x_v} \right)$$

$$\gamma = \tan^{-1} \left(\frac{\dot{y}}{\dot{x}} \right)$$

이를 통하여 베어링 각을 다음과 같이 정의한다.

$$\eta = \gamma - \lambda$$

위 수식을 통해 차량의 경로각이 시선각으로 수렴한다면 제어 목표 1을 만족한다는 것을 확인할 수 있다. 이를 상대 운동 방정식을 통해 자세 제어 명령을 산출해야 한다.

Table 5 상대 기하 변수 정의

변수 명	정의
(X_I, Y_I)	관성 좌표계의 X, Y 축
(X_B, Y_B)	동체 좌표계의 X, Y 축
(x_w, y_w)	관성 좌표계에서 정의된 경로점 좌표
(x_v, y_v)	관성 좌표계에서 정의된 동체 좌표계의 원점
γ	동체의 경로각
λ	동체에서 경로점까지 시선각
η	헤딩 오차/베어링 각
r	동체와 경로점 간 상대거리
a_v	동체의 가속도
V	동체의 속력

제어 명령을 도출하기 이전, 일관된 수렴 특성과 용이한 해석, 경로점까지의 강건한 추종 제어를 위해 2가지 가정을 도입하자.

- 1) 사전에 미리 정의된 경로점이 있으며, 이는 관성 좌표계에서 고정된 위치를 갖는다.
- 2) 자율이동체는 등속 운동을 하며, 속도 방향의 수직방향으로 가속도를 갖는다.

자율이동체의 자세 정보를 통해, 시선각 변화율을 관성 좌표계에서 표현하자.

$$\dot{\lambda} = -\frac{V \sin(\eta)}{r}$$

시선각 변화율과 경로각 변화율 차이로 베어링 각의 변화율을 기술할 수 있다. 이 때 Pursuit Guidance를 기반해 yaw-rate에 대한 명령을 다음 수식과 같이 기술할 수 있으며, 이는 조향에 직접 들어가지는 않으나, 조향 명령에 비례한 관계식을 갖는다.

$$\dot{\gamma} = -\frac{K_p V \sin(\eta)}{r} = K_p \dot{\lambda}$$

유도 이득 K_p 는 안정성을 만족하기 위해 2 이상의 값을 갖는다. 유도 이득이 클수록 수렴 특성 등 좋은 성능을 보이나, 유도 명령 및 속도 명령 그리고 이동체의 회전 반경, 조향 반경, 속도 등 기구학적인 한계를 고려하여 유도 이득을 산출한다. HADA팀에서 최대 조향 각도를 20도로 제한하였기 때문에, 입력 포화를 최소화하며, 점근 안정한 명령을 내릴 수 있도록, 유도이득을 4.5로 선정하였다. 유도제어의 블록 선도는 Figure 5와 같이 표현할 수 있다. 실제 명령에 대입하기 위해 유도 명령을 자전거 모델(Bicycle model)에 대입하여 조향각 명령으로 변환한다.

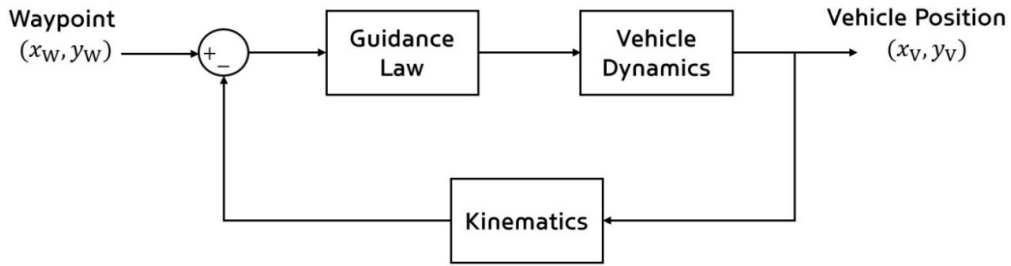


Figure 5 유도제어 루프

2.2. 지역경로계획

지역 경로 계획은 전역 경로 계획 상에서 현재 자율이동체 주위의 환경에 대한 인식과, 장애물과의 상호작용 없이 목적지까지 안전하게 갈 수 있도록 하는 알고리즘이다. 경로상에 충돌을 효과적으로 회피하는 기술로 사고를 미연에 방지하여 정상적인 자율 이동체 운용을 가능케한다. 장애물 회피를 위해 HADA팀은 2D LiDAR를 사용하였으며, 장애물 회피 알고리즘은 1-Dimensional 가상 조향장 알고리즘이다. 이는 적은 계산량을 바탕으로 하기 때문에 실시간 계산 성능이 뛰어나다. 올해 HADA 팀이 도전하였던 부분은 [2] DBSCAN를 응용하여 LiDAR에서 얻은 포인트 클라우드 기반 장애물장(Obstacle Force Field, OFF) 처리에 있다. OFF는 LiDAR를 이용해 탐지된 물체에 밀도 기반 클러스터링(Density-based spatial clustering of applications with noise, DBSCAN) 알고리즘을 거쳐 물체 중심위치를 계산하고, 동체 좌표계(B-Frame)에서 표현한다. 장애물은 PE 드럼통이나 일반적인 리버콘으로 모두 원형인 형태를 띄고 있기 때문에 장애물 중심 위치를 계산 후 원형으로 근사하여 OFF를 새롭게 제작하고 장애물 회피 알고리즘을 적용하였다.

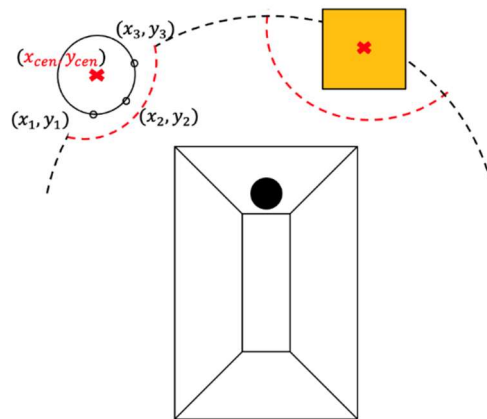


Figure 6 OFF 생성 방식

Figure 7의 왼쪽 이미지는 터널 구간에서의 라이다 raw data이며, 노이즈가 많이 발생하는 것을 확인할 수 있다. 외부에서 라이다를 가용할 때 전처리가 필수적이었고, 이를

DBSCAN으로 효과적으로 처리할 수 있었다. 정적 장애물의 크기를 고려하여 평균적으로 7m 내에 맷히는 포인트 개수가 장애물 당 최소 15개였기 때문에, DBSCAN 포인트 개수를 15로 설정하였고, 각 포인트별 최소거리를 10 cm로 선정하였다.

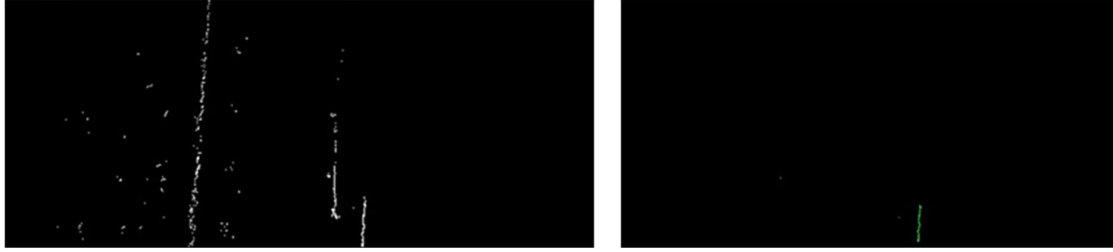


Figure 7 a) 라이다 포인트 클라우드 데이터 b) DBSCAN 결과

Figure 8은 DBSCAN을 통해 야외에서 러버콘을 인식한 데이터를 시각화한 것이고 오른쪽 사진은 그 데이터에 기반해 차량 좌표계로 좌표변환을 진행 후 OFF를 시각화한 것이다.

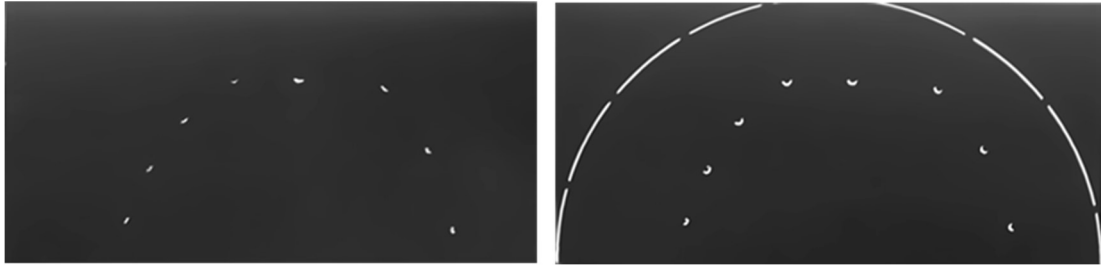


Figure 8 DBSCAN 기반 OFF 생성 결과

DBSCAN으로 얻은 데이터를 기반해 물체의 중심 좌표 기준으로 장애물을 원형으로 근사하였다. 원으로 가정한 이유는 형상이 회전에 영향을 받지 않으며, 물체를 보는 각도가 다르더라도 같은 공식을 적용할 수 있기 때문이다. 이 경우, Figure 6과 같이 가정할 수 있으므로 다음과 같은 수식을 유도할 수 있다.

- 장애물의 원의 방정식

$$(x - x_{cen})^2 + (y - y_{cen})^2 = r^2$$

- 차량 중심을 지나는 직선의 방정식

$$y = ax$$

- 장애물 인식 지점

$$x = \begin{cases} \frac{x_{cen} + ay_{cen} - \sqrt{(x_{cen} + ay_{cen})^2 - (a^2 + 1)(x_{cen}^2 + y_{cen}^2 - r^2)}}{a^2 + 1} & (x_{cen} \geq 0) \\ \frac{x_{cen} + ay_{cen} - \sqrt{(x_{cen} + ay_{cen})^2 - (a^2 + 1)(x_{cen}^2 + y_{cen}^2 - r^2)}}{a^2 + 1} & (x_{cen} < 0) \end{cases}$$

$$\text{단, } \tan^{-1} \delta_{Wmax} \geq a \geq \tan^{-1} \delta_{Wmin}$$

이 때 r 은 차량 크기와 위치에 따라 지정해야 하는 값이며, 장애물 인식 범위 내에

있고 차량 조향 명령 구간에 포함되어 있는 모든 점유 공간에 대해 반복 계산한다. 차량의 좌우 폭과 무게중심으로부터 모서리까지의 길이를 계산해 물체 중심과 x 축이 이루는 각이 작을수록 더 큰 r 값을 갖는다. 현재 차량의 속도, 차량의 전폭, 무게 중심으로부터 전면부까지의 거리인 $95cm$ 를 고려해, x 축과 이루는 각에 대한 1차 함수를 작성해 r 값을 결정했고, 장애물을 안정적으로 회피하는 모습을 관측하였다.

2.3. 미션 알고리즘

2.3.1. 협로 주행

협로 주행(트랙 주행)은 리버콘으로 만들어진 원형 트랙을 자율적인 차량의 운용으로 빠르고, 리버콘 사이를 안정적으로 주행하는 미션이다.



Figure 9. 트랙 주행 테스트 장소

상세 규정은 다음과 같다.

- 1) 리버콘 간 간격은 최소 $2m$ 최대 $5m$
- 2) 출발 위치 및 트랙의 형태 정보를 사전에 제공하지 않음
- 3) 트랙 내부에서 출발
- 4) 리버콘 색상은 무작위
- 5) 리버콘의 받침대를 밟는 것까지는 허용되나 리버콘과 충돌하는 경우 이전 리버콘 위치에서 재출발

당일 대회 특징으로 작년에 비해, 내부 리버콘과 외부 리버콘의 사이 폭이 늘어났으며($2m \rightarrow 2.5m$) 외부 리버콘(Figure 9 기준 노란색 리버콘)의 간격이 작년에 비해 줄었다. 어려운 점으로는 급커브 구간의 반경이 감소하였다.

HADA 팀의 기본 아이디어는 AMZ Formula 대회의 *flüela* 팀 영상을 참고하였다. 본 영상에서는 RRT*알고리즘을 이용해 물체에 충돌하지 않는 무작위 경로를 생성하고(Figure 10), 생성된 경로 중 최적의 경로를 찾아 주행하는 방식이었다.

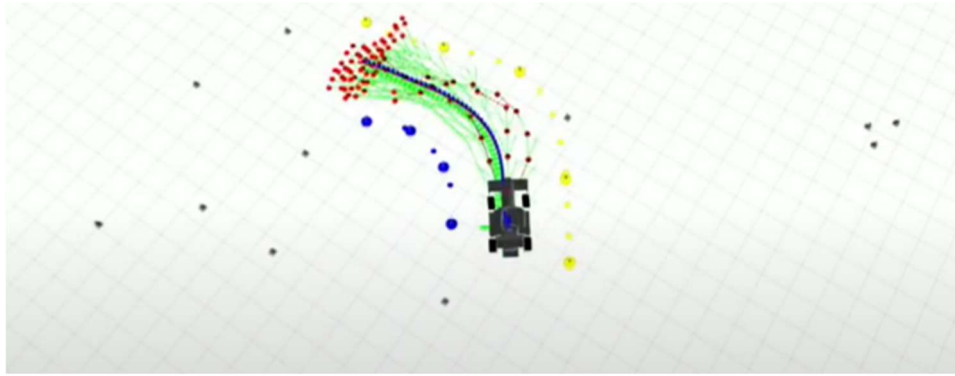


Figure 10 AMZ Formula *flüela* 팀의 RRT* 알고리즘 시각화 자료

HADA팀 또한 RRT*와 같이 경로를 생성하지만, 후보 지점이 무작위가 아닌 방법을 고안하였다. 무작위로 생성되는 경로로 인해 문제가 발생하였을 경우, 원인 분석이 어렵다는 것이 이유다. 해당 아이디어를 구현하기 위해 경로점을 생성하는 기준을 찾았다. 영상에서 발생하는 경로 대부분은 안쪽 리버콘과 바깥쪽 리버콘을 교차로 연결했을 때, 그 중심을 지나가는 것으로 확인할 수 있었다.

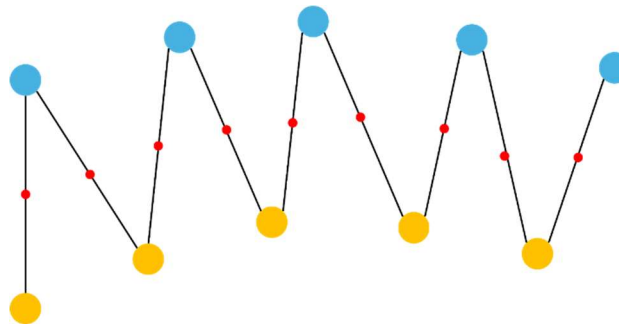


Figure 11 경로가 생성되어야 할 지점

HADA팀은 Figure 11과 유사한 경로점을 산출하고, 이를 바탕으로 경로를 결정하는 방법을 고안하였다. 알고리즘은 Figure 12의 순서도를 따른다.

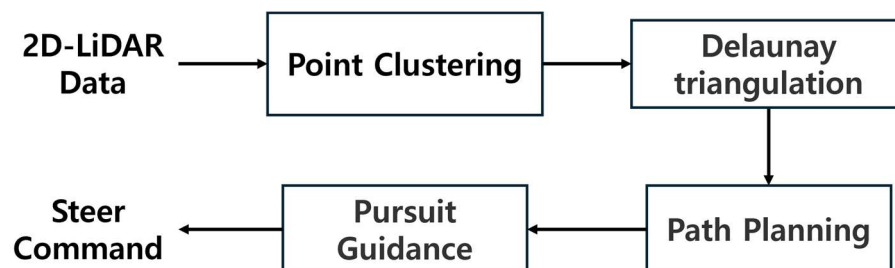


Figure 12 알고리즘 순서도

센서는 2D LiDAR를 사용한다. 2D LiDAR에서 나온 데이터를 군집화하여, 물체를 구분하고 노이즈를 제거한다. 또한, 실험적으로 리버콘의 경우, 클러스터 분산 값이 0.005를 넘지 않는 것을 확인했다. 따라서 클러스터링 된 포인트의 평균과 분산을 구하여, 다른 물체를 분리할 수 있다. 이 프로세스의 결과로 리버콘으로 추정되는 물체들의 중심 위치

를 반환할 수 있다.

이후 들로네 삼각분할 알고리즘을 사용해 리버콘 중심 좌표끼리 잇는 겹치지 않는 선을 Figure 13과 같이 생성한다.

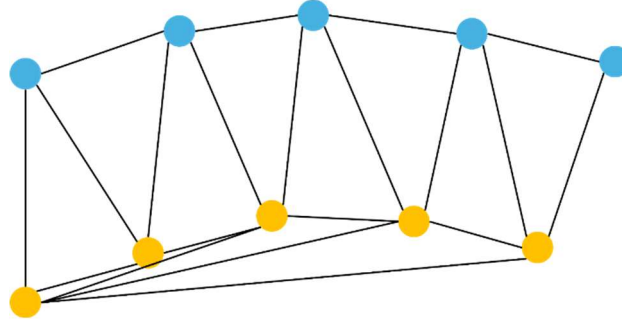


Figure 13 들로네 삼각분할 결과

먼저, 삼각분할로 만들어진 선들의 중심위치를 계산한다. 이때, 세가지 예외사항을 적용한다.

- 1) 차량 폭보다 좁은 선 사이는 지나갈 수 없으므로 제외한다.
- 2) 규정상 리버콘 사이 간격이 5m가 넘지 않으므로, 5m를 초과하는 경우 제외한다.
- 3) 1), 2)를 통해 나온 점을 후보점으로 가정한다. 이후 차량의 회전 반경 밖에 있는 점은 차량이 진행 불가능하므로 제외한다.

이를 통해 Figure 14와 같은 경로점 후보를 선정할 수 있다.

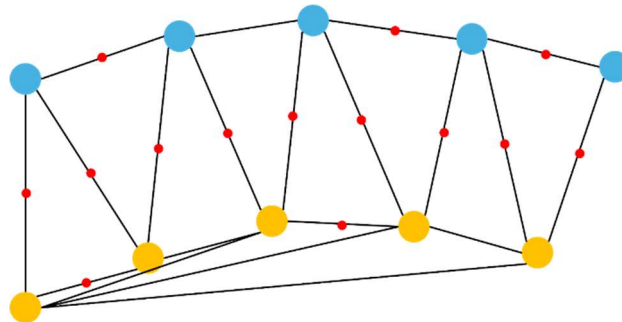


Figure 14 경로점 후보 선정 결과

경로점 후보 중 적합한 후보점을 가리기 위해, 아래의 비용함수 f_c 를 정의하자. 이때, d 는 현 위치와 점 사이의 거리며, θ 는 현재 위치에서 바라본 점의 시선각이다.

$$f_c(d, \theta) = \alpha d + \beta |\theta|$$

해당 비용함수가 최소가 되는 점 5개를 탐색하고, 이를 2차 함수 형태로 회귀한다. Figure 15에 도시하였듯 2차 함수의 전방 1m 지점을 경로점으로 선정하였다. 해당 경로점은 동체 좌표계에서의 좌표로 표현할 수 있다. 이를 Pursuit Guidance를 이용해 산출하

였다.

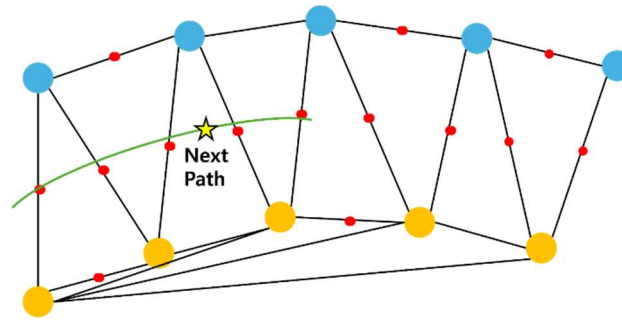


Figure 15 최종 경로점 결과

2.3.2. 표지판 인식(배달 미션)/ 신호등

표지판 인식은 작년과 동일하게 도로 주행 중 표지판, 신호등 등 주변 환경 인식을 위해 객체 검출 기법으로 YOLO V8 딥러닝 모델을 사용하였다. 작년보다 더 빠른 객체 인식을 위해 배달 미션을 위한 학습에서 segmentation data가 아닌, bounding box를 통해 학습 영역 지정으로 학습을 진행하였다. 이미지 데이터셋 학습을 하기 위해 Roboflow 라벨링 프로그램을 사용했다. Roboflow는 데이터 증강 기능을 통해 보다 견고한 모델 생성이 가능하다.

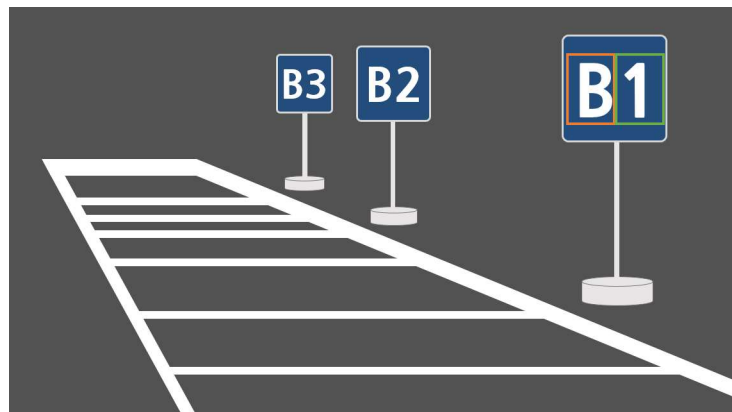


Figure 16 Bounding box를 이용한 표지판 학습 영역 지정

배달 미션을 수행하기 위해 학습한 데이터는 표지판, 알파벳, 숫자 정보로 나뉜다. 센서 데이터에 딥러닝 모델을 적용한 데이터를 사용해 배달 표지판을 비교한다. 표지판 내부 정보 인지를 명확히 하기 위해 confidence threshold는 0.75로 인지된 알파벳과 숫자 정보 중 표지판 내부에 있는 정보만 유효하게 판단해 표지판의 내용만 파악할 수 있었다. 실험을 통하여 Figure 17의 표지판 내용을 확인할 수 있었다.



Figure 17 배달 미션을 위한 표지판 인식 결과

배달에 필요한 배달 미션 학습 모델의 혼동 행렬은 Figure 18과 같다.

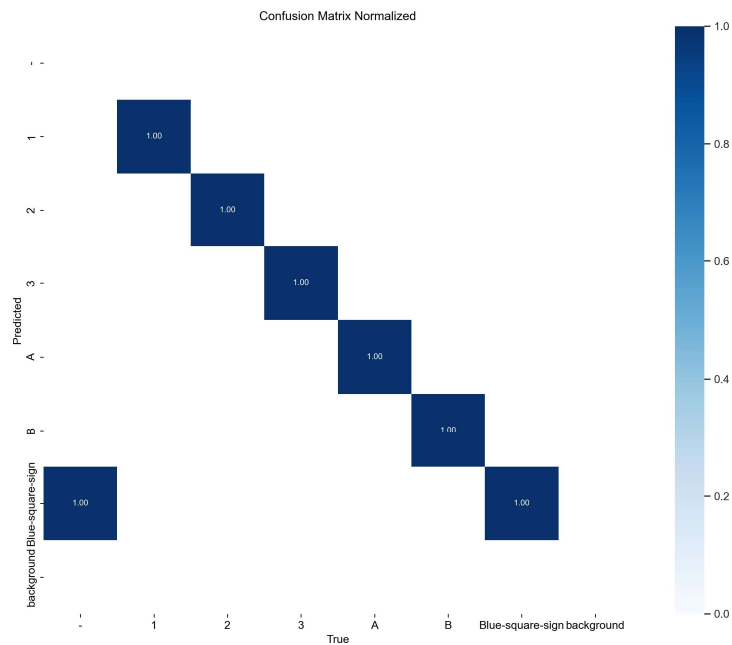


Figure 18 배달 미션 학습 모델의 혼동 행렬

배달 미션에서 배달 표지판과의 거리 정보를 판단하기 위해 카메라-3D LiDAR 캘리브레이션을 수행하였다. 빠른 연산을 위해, 전체 프레임의 왜곡 보정이 아닌, 인지된 표지판의 카메라 픽셀 좌표에 대한 왜곡 보정을 수행했다. 인지된 표지판의 보정된 카메라 픽셀과 라이다의 3D좌표를 이용해 아래 수식과 같이 캘리브레이션을 수행하였다. 이를 통해 동체와 표지판의 거리 정보를 추출할 수 있다.

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & skew & f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [R|T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

s : scaling parameter
 x, y : pixel coordinate
 f_x, f_y : focal length
 c_x, c_y : principal point
 R : rotation matrix
 T : translation matrix
 X, Y, Z : 3D – LiDAR coordinate

Figure 19처럼 동체와 표지판 간의 거리를 탐지하기 위하여 Figure 20에 도시한 플로우차트 형태로 배달 알고리즘을 구성하였다.

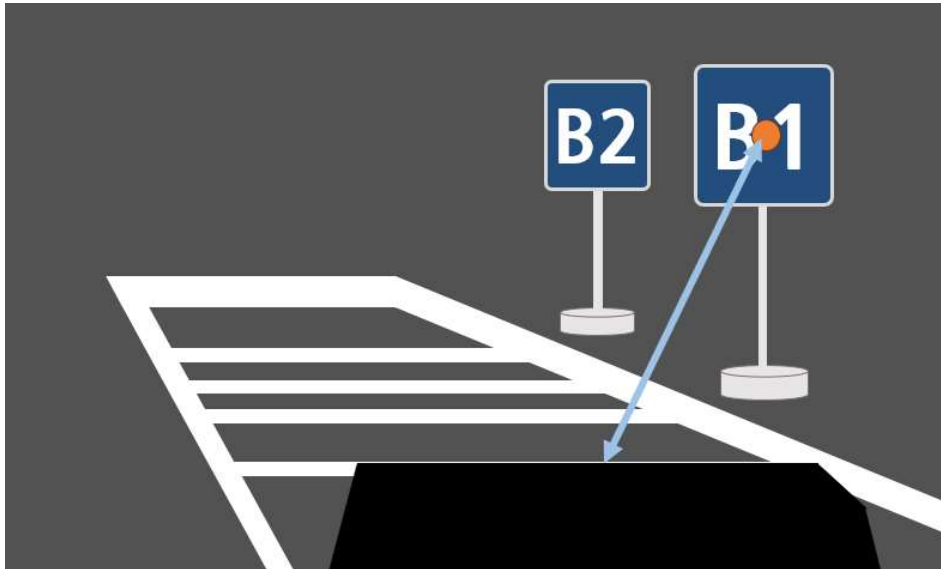


Figure 19 카메라와 3D 라이다 캘리브레이션을 통한 동체와 표지판 거리 탐지

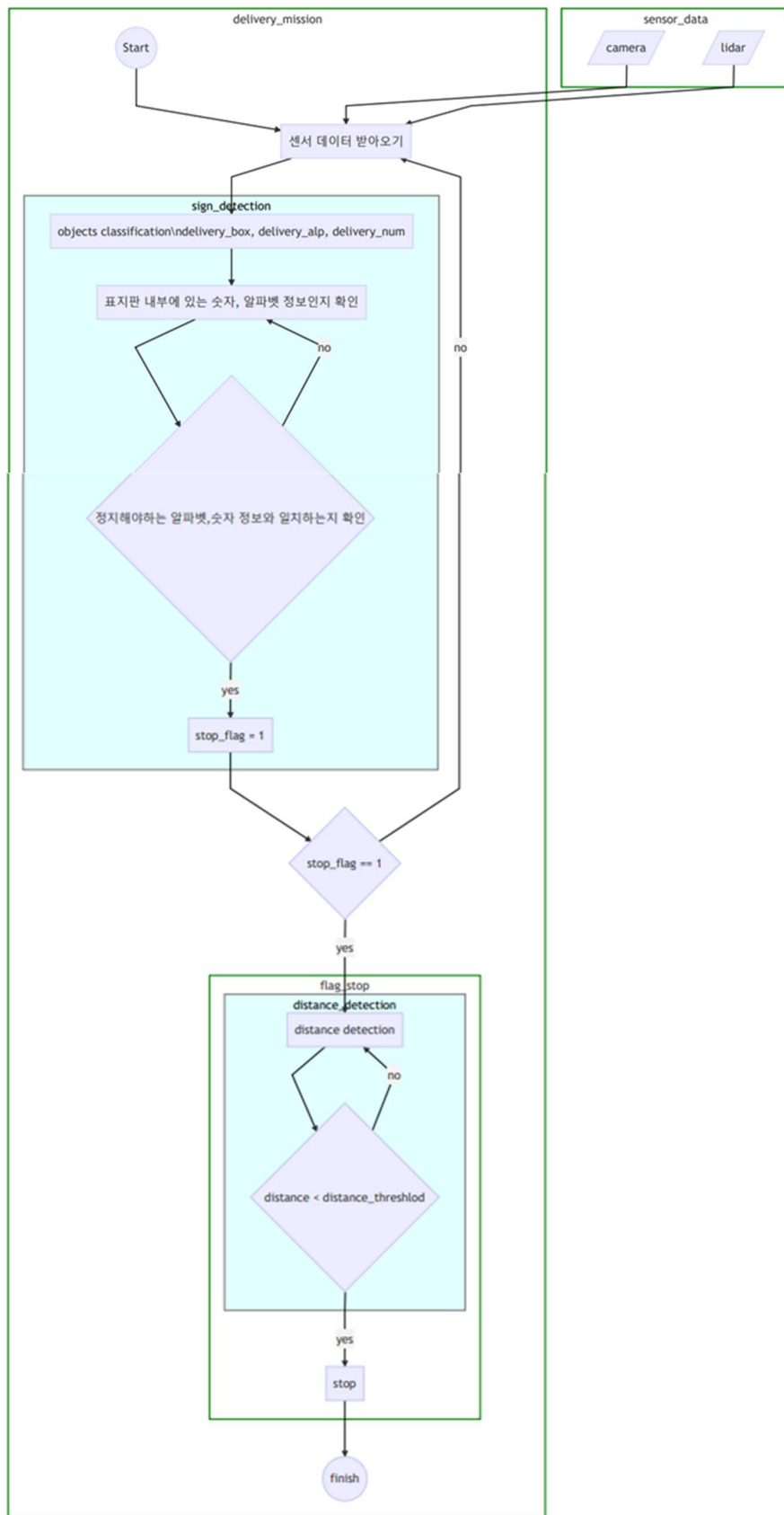


Figure 20. 표지판 인식 플로우차트

2.3.3 신호등 인식

신호등 인지에 필요한 학습 모델의 혼동 행렬은 Figure 21과 같다. 신호등 학습은 온전한 신호 인지를 위해 신호 각각이 아닌 신호등 자체를 학습시켰으며, 순서대로 4구 초록불 신호등, 4구 초록불과 우회전 신호등, 4구 빨간불 신호등, 3구 빨간불과 우회전 신호등, 4구 빨간불과 우회전 신호등, 4구 노란불 신호등을 학습하였다.

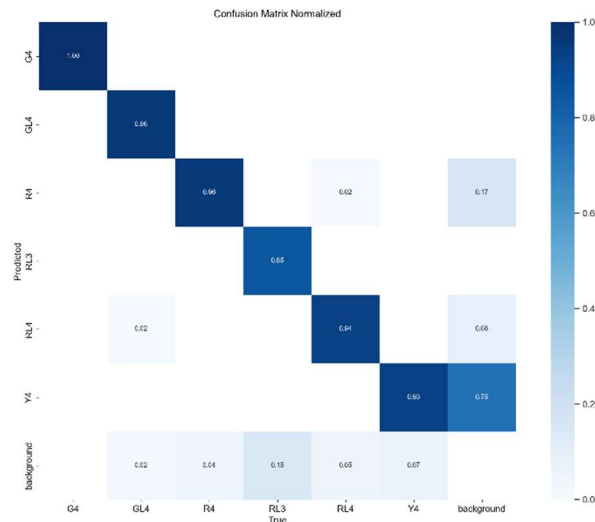


Figure 21. 신호등 미션 학습 모델의 혼동 행렬

동체와 표지판의 인지 후 거리탐지를 필요로 했던 배달 표지판 인지와 달리 신호등 인지에서는 신호등 라벨 정보를 요한다. Blink 현상이 심한 신호등 모델을 보완하여 신호등 인지를 위한 플로우차트를 Figure 22에 도시하였다.

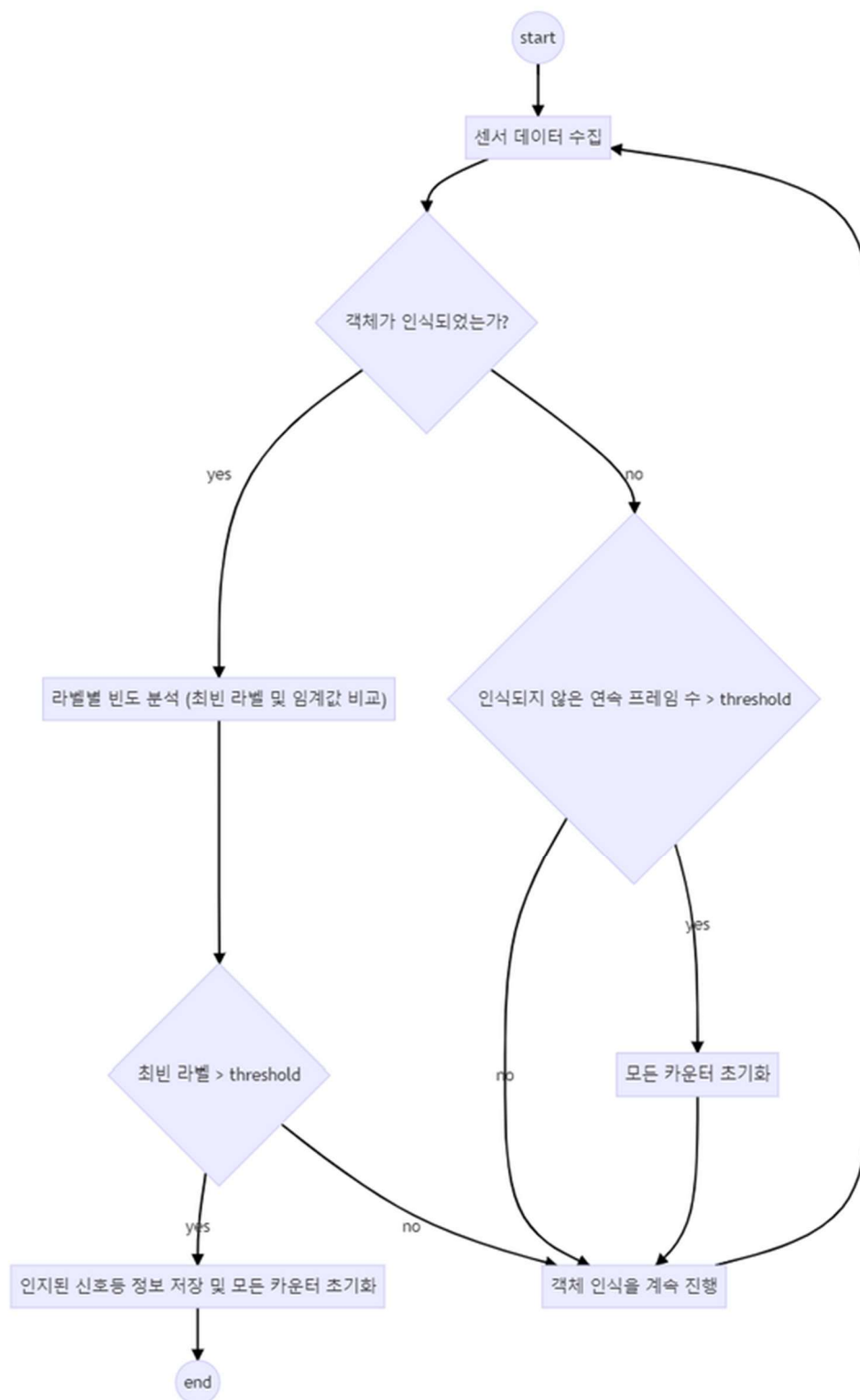


Figure 22. 신호등 인식 플로우차트

2.3.4 평행 주차

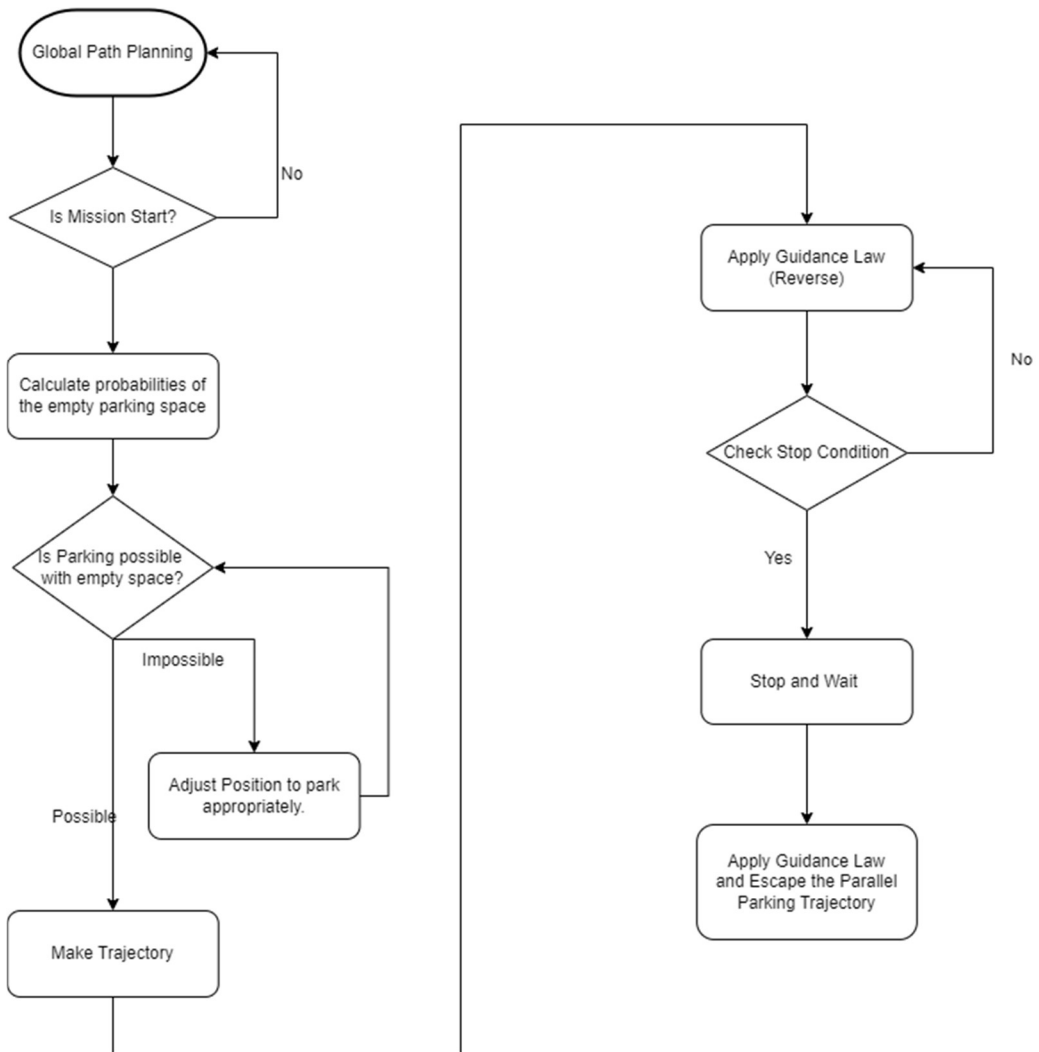


Figure 23 평행주차 알고리즘 플로우차트

이번 HADA팀 또한 가장 먼저 각 주차칸 입구의 위경도 정보를 갖고 있기 때문에, 예년과 비슷하게 주차칸 입구 중심의 1m 반경에서 검출되는 장애물을 라이다, GPS, 자세를 통하여 검출한다. 3칸 중 2칸이 점유되었기 때문에 다음과 같은 수식을 얻을 수 있다.

$$P(Occupied|Parking) = 2/3$$

이를 로그오즈 형태로 만들어, 로지스틱 회귀를 수행한다. n 은 1m 반경에서 검출된 포인트 클라우드 개수이다.

$$\text{logit}(P) = \log\left(\frac{p}{1-p}\right), \quad P(Occupied) = 1 - \frac{1}{1 + e^{n \text{logit}(P)}}$$

확률적인 접근을 통해 점유 공간에 대한 판단을 수행할 수 있었다. 3칸 중 가장 확률이 낮은 곳을 주차 가능한 구역으로 판단하였다. [3]과 같은 상황에서 두 원의 접선이 같은 반경의 원을 그려 궤적을 Figure 23과 같이 생성하였다. 주차칸의 크기와 차량의 크기, 차선의 폭을 고려하여, 주차장 진입 이전에 부딪히지 않는 위치(Figure 22)로 이동하는 것을 해당 알고리즘의 안전장치로 사용했다. 이는 [4]와 같이 두 원호를 통하여 구현하였다.

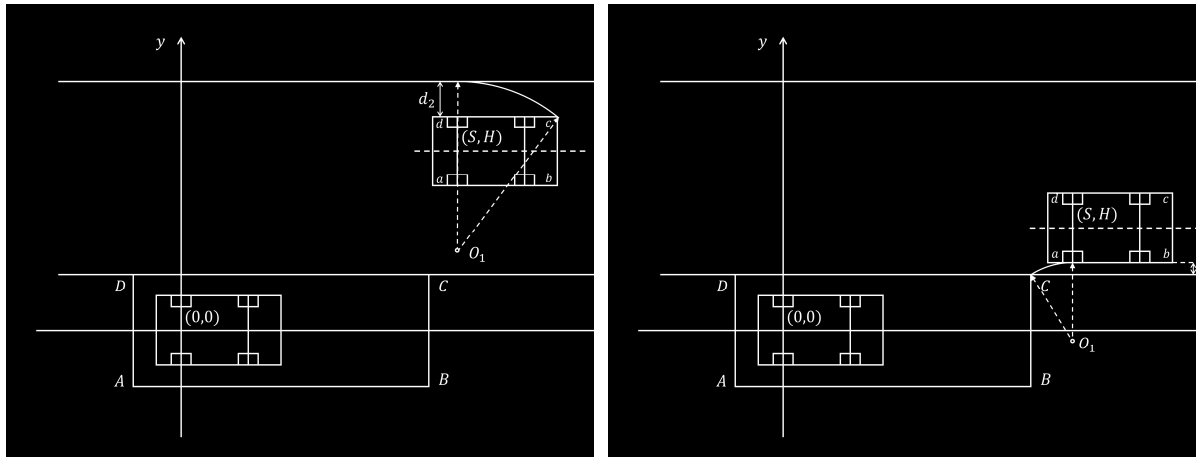


Figure 24 차량 충돌 방지를 위한 영역 설정

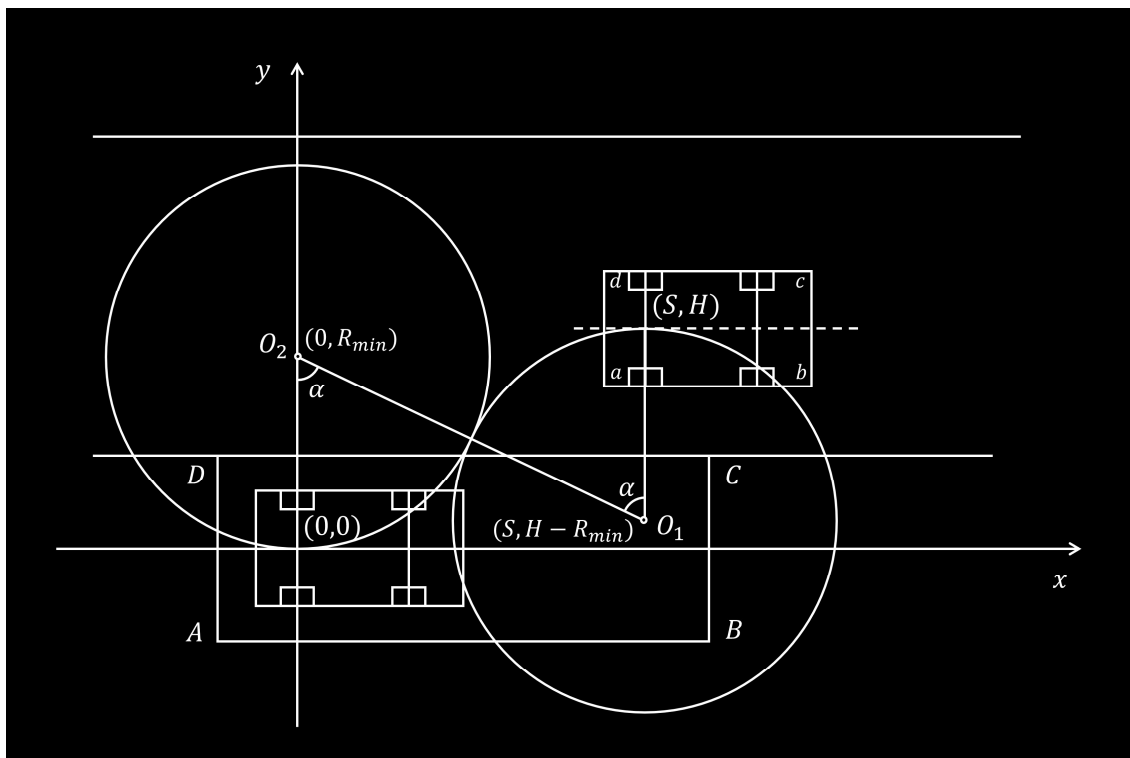


Figure 25 주차 궤적

2.3.5 BEV 변환 및 차선 유지 알고리즘

차량 앞부분에 탑재된 카메라로 도로 전방의 영상 정보를 수집, Raw 이미지에 왜곡 보정을 적용 후 색상 변환, 그라디언트 등을 사용하여 이진화된 이미지를 생성한다. 원근 변환, 차선 픽셀 피팅을 거치며 효율적인 차선 인식 기법을 제시한다. Figure 24의 순서도는 이미지 처리가 어떻게 진행되는지 보여준다.

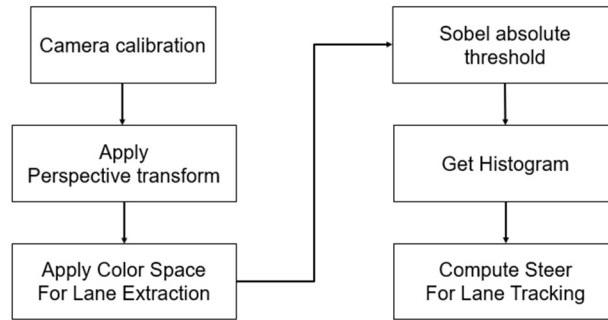


Figure 26 영상처리 순서도

본 과정은 차선 유지를 위한 조향 명령을 산출해 내기 위해 효과적인 방식으로 기존 도로 상황을 하늘에서 수직방향으로 내려다 보는 형태로 변환하는 방식이다. 이 변환을 이용하여 차선 검출 시, 차선의 기울기, 노이즈 제거 측면에서 처리가 쉬워지는 장점이 있다. BEV 변환은 다음 수식을 통해 이루어진다.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} : \text{Rotation Matrix}$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} : \text{Translation Matrix}$$

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} : \text{Camera Intrinsic Matrix}$$

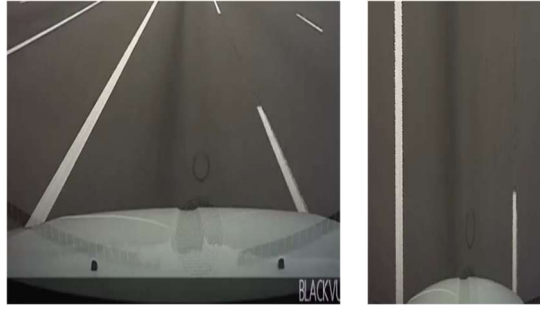


Figure 27 BEV transform

효과적인 차선 인식을 위해서는 노란색과 흰색 도로 차선 이미지가 표시되는 두 가지 주요 색상을 감지 해야 한다. 기존 그레이 스کیل 방식은 햇빛에 의한 음영, 도로 상태, 노란색과 흰색 차선 구분에 있어서 임계값을 설정하는데 어려움이 있었다. 이를 보완하기 위해 HSV, HLS 및 LAB 총 3가지 색상 공간을 테스트하였다. Figure 28, 29, 30은 흰색 차선 이미지를 각 색상 공간 채널을 적용시킨 차선 결과와 색 공간 다이어그램이다.

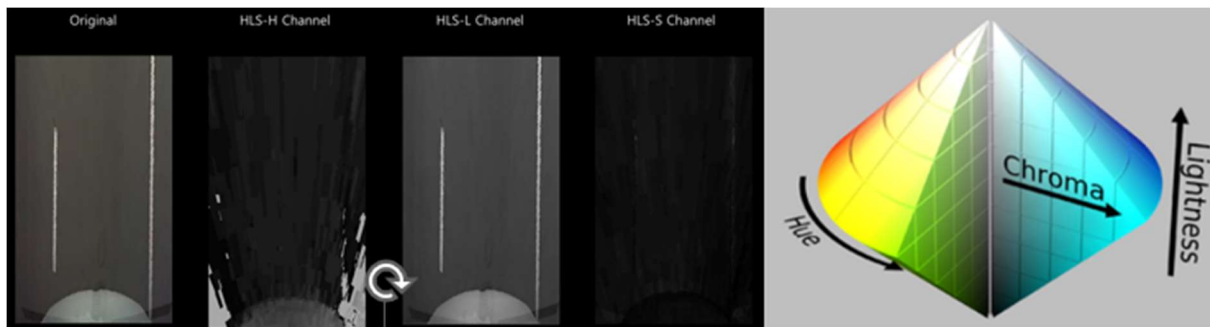


Figure 28 HLS 색상공간 채널

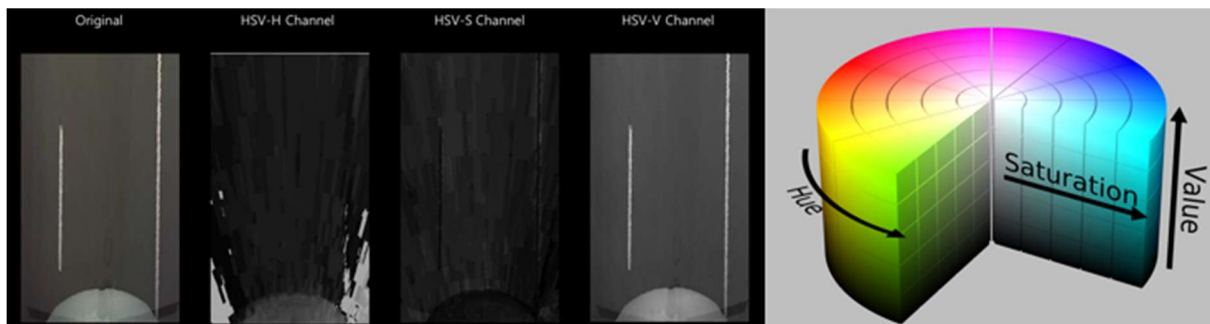


Figure 29 HSV 색상공간 채널

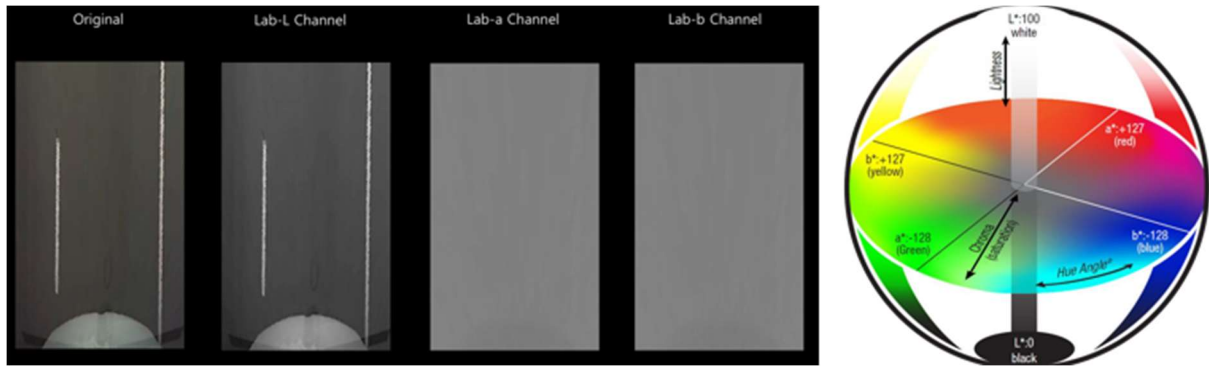


Figure 30 LAB 색상공간 채널

하나의 채널만으론 흰색 차선과 노란색 차선 둘 다 효과적으로 검출할 수 없었기 때문에 LAB-A채널을 통해 노란색 차선을, HLS-L채널을 통해 흰색차선을 따로 검출 후 결합하는 방식으로 기존 그레이 스케일 방식을 발전시키고자 하였다.

기존 Canny edge 검출 방식은 도로 이미지의 가장자리를 감지할 때 모든 가장자리를 감지하는 연산량의 문제가 있었다. 불필요한 엣지 감지를 최소화하기 위해 Sobel 연산을 사용하였다. 3×3 matrix를 이용하여 중심 픽셀을 기준으로 각 방향의 픽셀 변화량을 검출 후 임계 값을 적용하여 출력 이미지를 이진화하는 방식을 사용하였다.

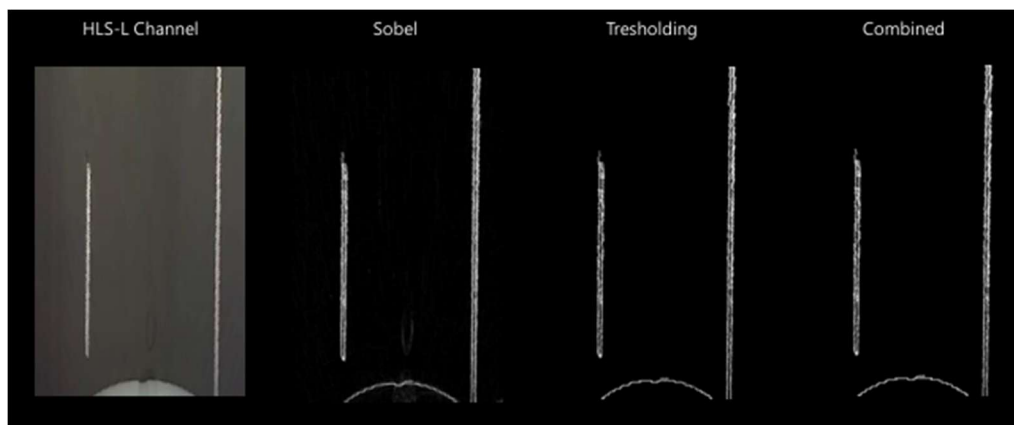


Figure 31. Sobel 연산 후 임계 값 적용 결과 (이진화 이미지)

기존 허프 방식은 검출된 모든 엣지들을 차선으로 인식하는 문제가 있었다. 이러한 문제를 개선하기 위해 Histogram data point를 생성한다. Data가 반환하는 값은 생성된 Histogram의 x,y 좌표이며, 차선의 x 좌표를 나타낼 수 있는 좋은 지표로 활용될 수 있다. 차선 경계 주변에서 평균 흰색 픽셀 밀도는 비교적 다른 영역보다 높게 나타난다. Histogram의 peak 값 x 좌표를 각 차선의 x 좌표로 인식하여 이후 차선유지를 위한 조향점 계산에 사용한다.

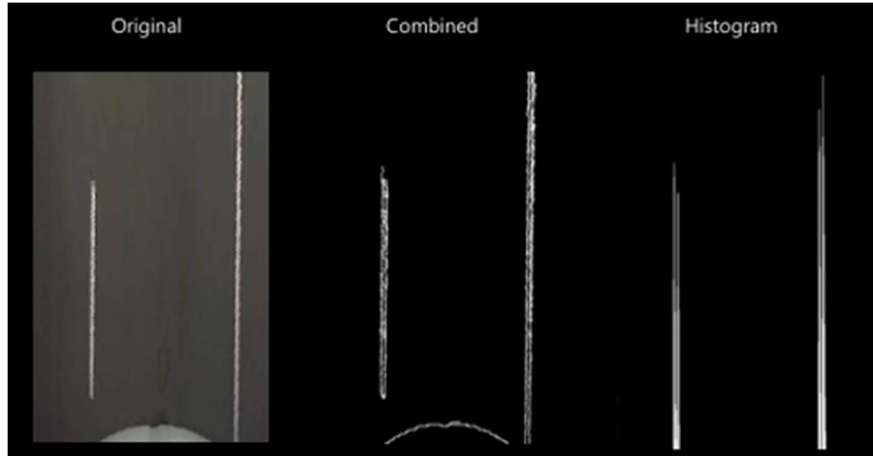


Figure 32. Histogram 검출

위의 Histogram 검출을 통해 얻은 x 좌표를 차선 검색 윈도우 창의 시작점으로 사용할 수 있다. 인식된 차선을 추적하는 슬라이딩 윈도우 창을 형성하여 선택한 창을 반복하여 차선의 곡률을 계산할 수 있다. 2차 이상으로 근사할 경우 차선 픽셀이 적게 검출 되었을 때 차선 정보가 불명확하게 검출될 경우가 빈번하게 발생하여 본 알고리즘에서는 1차 다항식으로 근사하여 다항식 계수들을 추정하였다. 1차 다항식으로 근사할 경우 차선의 기울기와 x 절편 정보를 얻을 수 있으며 이후 추가적인 조향점 계산에 활용할 수 있다.

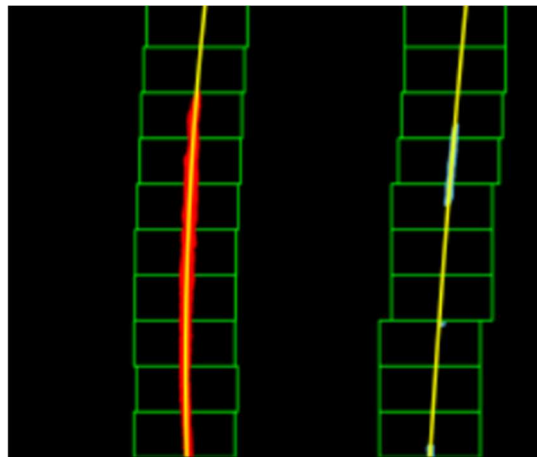


Figure 33. Sliding window 창을 통한 polynomial fitting

Camera Pixel Coordinate System은 Cartesian Coordinate System을 x 축 대칭 시킨 것을 기본으로 한다. Pursuit algorithm을 적용시키기 위해 좌표계 변환이 필요하다.

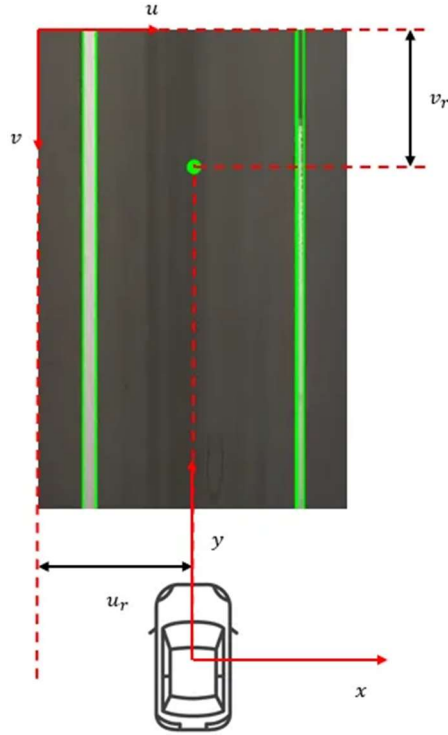


Figure 34. BEV이미지를 통한 조향점 계산

Table 6. 저속 선회 모델 (Ackerman geometry model parameter)

$G(g_x, g_y)$	Body-frame 상 목표점
l_d	예견거리(m)
α	헤딩각
δ	조향각

본 알고리즘에서는 저속 선회 모델 중 하나인 애커만 기하학 모델을 이용하여 조향각을 도출한다. 횡방향 힘이 없고, 즉 후륜 슬립이 발생하지 않는다는 가정하에 회전 중심은 후륜 구동축 연장선상에 있다. 이때 조향각은 $\delta \approx \arctan\left(\frac{L}{R}\right)$ 로 도출된다. 삼각함수 공식을 이용하면 조향각 명령을 산출할 수 있다. 정확한 조향각 계산을 위해 카메라와 후륜 구동축 사이의 거리 또는 안테나까지의 거리 정보가 필요하다.

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(90 - \alpha)} = \frac{R}{\cos(\alpha)}$$

$$R = \frac{l_d \cdot \cos(\alpha)}{2 \cdot \sin(\alpha) \cdot \cos(\alpha)} = \frac{l_d}{2 \sin(\alpha)}$$

$$\delta(t) = \tan^{-1}\left(\frac{L}{R}\right) = \tan^{-1}\left(\frac{2L\sin(\alpha(t))}{l_d}\right)$$

Figure 35은 본 알고리즘에서 사용한 저속 선회 모델을 기하학적으로 나타낸 그림이다.

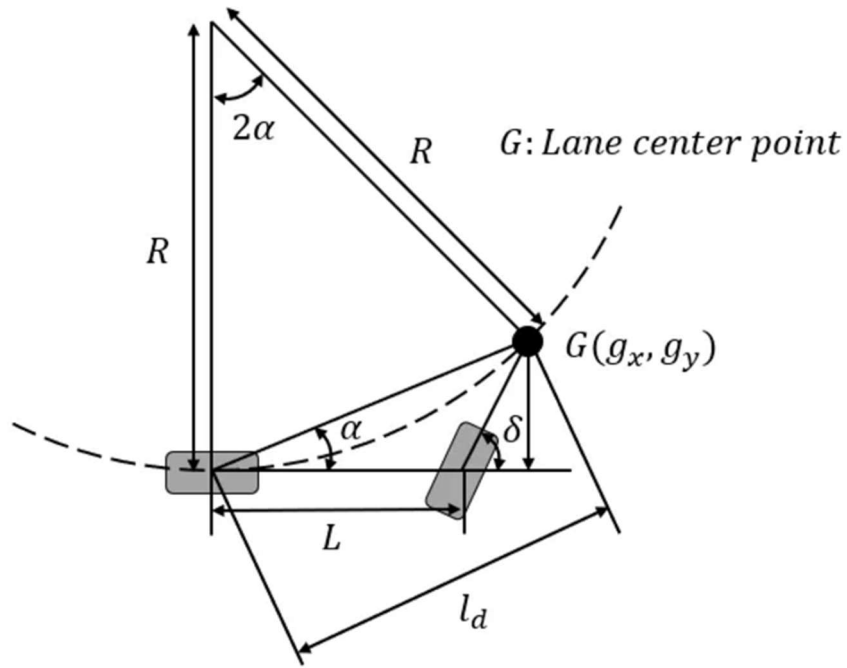


Figure 35. 저속 선회 모델

고전영상처리 방식만을 이용했을 경우 다양한 환경에 대해서 높은 신뢰성을 확보하기 어려웠다. 딥러닝을 추가적으로 사용하여 차선을 검출하는 시도는 기존 방식보다 검출 및 인식 부분에서 성능을 크게 확보할 수 있었다. 본 알고리즘에서 사용한 YOLO V8 segmentation 모델은 차선의 label 번호, 클래스 세그멘테이션이 적용된 된 픽셀 좌표, bounding box의 시작 x,y 좌표를 반환한다. 이를 통해 차선 픽셀을 1차 함수로 근사할 수 있으며 차선의 기울기 및 조향점을 산출해 낼 수 있었다.

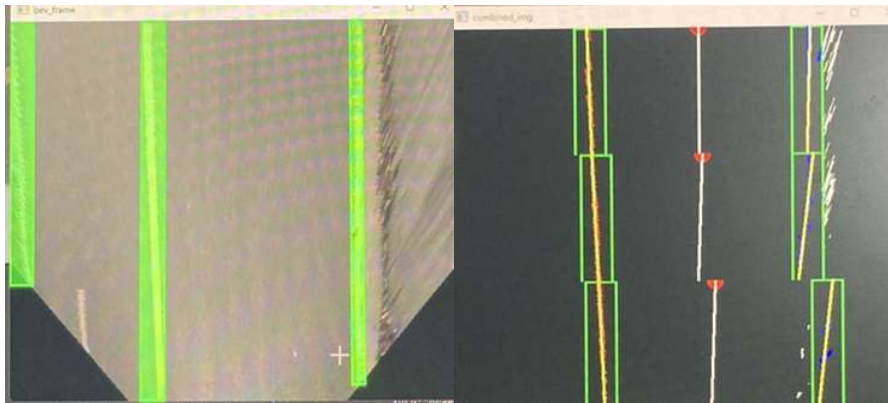


Figure 36. Yolo model 적용 및 Polynomial fitting 적용 결과

그럼에도 기존 BEV 변환을 통한 고전영상처리 차선유지 기법은 주어진 환경에 따라 영향을 많이 받아 많은 예외처리를 필요로 한다. 보다 포괄적이고 일관된 재구성을 위해 다중 카메라 이미지의 체적 점유율을 예측하는 Surround Occ 방법을 제안한다. 다중 카메라 3D 장면 재구성 모델을 구성하기 위해 LiDAR Point Cloud data를 생성하여 추출된 camera 이미지에 대한 멀티 스케일 특징과 융합하여 occupancy map을 구성한다.

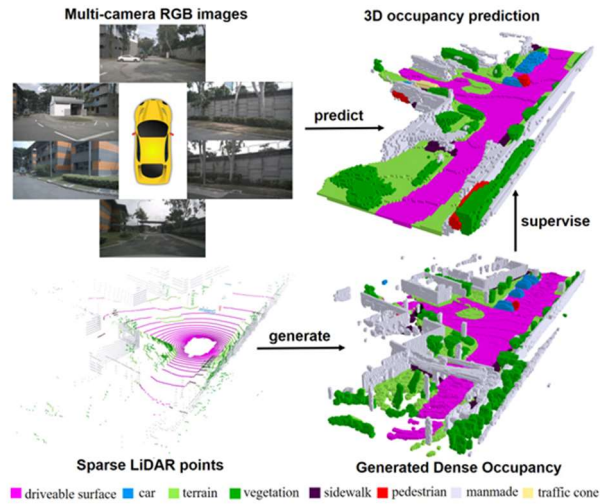


Figure 37. The overview of Surround Occ

2.4. 하드웨어

Wego측에서 제공하는 ERP 42 플랫폼은 하드웨어적 접근성이 떨어진다. 예를 들면 내장된 엔코더가 고장이지만 따로 분리하여 수리하거나 직접 교체를 할 수 없는 구조이다. 접근성 이외에도 경량화 문제, 저장 공간 문제 등으로 인해 HADA 4기에서 자체 플랫폼 제작을 하였다. 새로 만들어진 플랫폼의 문제점 또한 존재한다. 향후 현재 가용하는 플랫폼의 문제점을 빠르게 파악하여 수정하기 위한 기본 지식을 문서화 하였다.



Figure 38 차량 도안

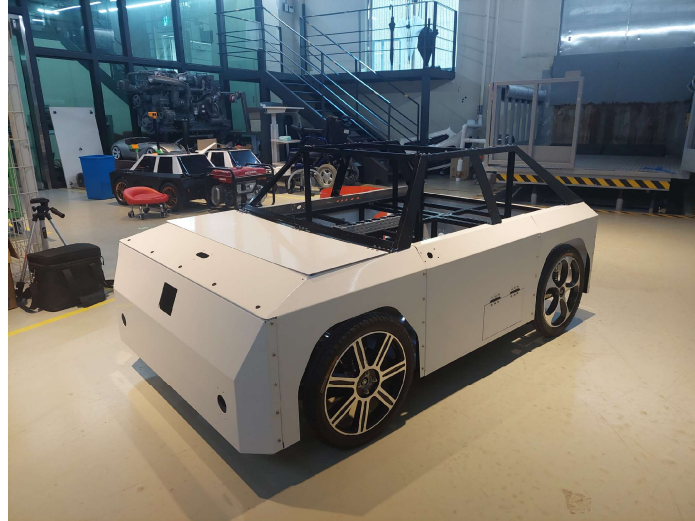


Figure 39 HADA 플랫폼 실물

Figure 38, 39은 현재 HADA에서 사용 중인 플랫폼의 시뮬레이션 모델과 실제 차량의 사진을 나타낸다. 본체 프레임은 2020 알루미늄 프로파일로 제작되었으며, 외판 판넬은 1.5T 금속판으로 구성되었다. 창문은 2T 아크릴로 제작되었으며, 센서 설치 층의 바닥과 모니터 지지대는 3T 아크릴을 사용하였다. 바퀴는 14인치 스쿠터용 바퀴를 사용하였다.

두 카메라 중 하나는 차선 인식을 목적으로 바닥면과 평행한 기준에서 약 19.5° 하향 각도로 배치되도록 하우징을 제작하였다. 다른 카메라는 신호등과 표지판 인식을 위해 약 105° 의 상향 각도로 배치되도록 하우징을 제작하였다. 2D 라이다는 장애물 인식을 목적으로 센싱 위치를 바닥으로부터 약 40cm 로 설치하였고 3D 라이다는 표지판 인식을 목표로 약 90cm 높이에 설치되었다. 라이다는 센싱 범위가 15m 이상으로 넓기 때문에, 보다 정밀한 센싱을 위해 pitch가 바닥과 평행을 유지하도록 유의하여 설치하였다.

조향은 NF 소나타 윈도우 모터를 사용하고 평기어를 사용하여 Potentiometer로 $\pm 2.5\text{V}$ 전압을 MCU ADC 기능을 이용해 각도를 계산하였다. 사용되는 PC 두대는 EcoFlow EF RIVER Pro에 연결되고 구동 모터는 24V 배터리에 연결된다. 조향 모터는 Regulator를 통해 12V를 인가 받는다.

자세한 3D 모델링 파일과 설계는 하다 공용 저장 공간에 저장하였다. 모델링 작업을 위해 사용된 tool은 *Autodesk Inventor Professional 2024*이다. 현재 플랫폼에 사용되는 센서와 차량 제원은 Table 1에서 확인 할 수 있다.

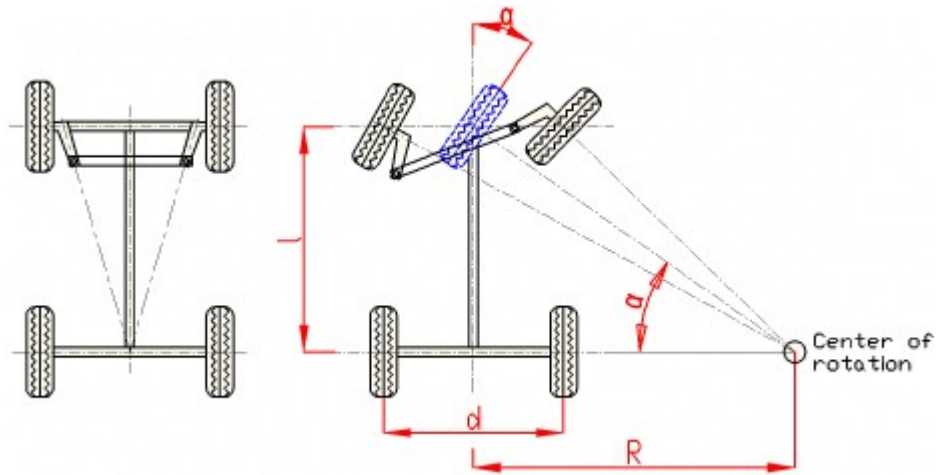


Figure 40 아커만 조향

현재 HADA의 새로운 플랫폼은 아커만 조향으로 설계 되어있다. 곡선 주행에서 코너 안쪽 조향바퀴가 코너 바깥쪽 조향 바퀴보다 더 큰 각도를 가지는 토아웃 현상이 일어난다. 이때 모든 바퀴가 동일한 회전 중심을 공유하기에 안정성있는 조향이 가능하다. 고속으로 곡선 주행시 좌우로 치지는 하중에 코너 바깥쪽 타이어의 슬립 앵글이 증가한다. 하지만 대회 규정과 모터 스펙으로 약 15km/h로 주행하는 HADA플랫폼은 큰 영향을 받지 않아 아커만 조향이 채택되었다. Figure 40는 아커만 조향 원리를 보여준다.

III. 문제점 및 해결 방안

아래에서부터 순서대로 II장의 절과 같은 번호로서, 번호 별 알고리즘의 문제점 및 해결방안에 대하여 작성하였다.

1. 전역경로 계획

- 유턴하는 부분에서 최대 조향 명령이 들어갔음에도, 최소회전 반경으로 인하여, 한동대학교 뉴턴홀 앞 2차선에서 유턴을 수행할 수 없었다. 이는 하드웨어 절에서 자세히 다루도록 한다.
- 2024년도 HADA팀의 전역경로 계획에서의 가장 큰 문제점은 GPS Jamming영향이다. 해당 문제점은 약 20개의 출전 팀 중 HADA를 포함한 2~3팀만 생기는 문제였다. HADA에서 심한준 박사에게 문의한 결과 Appendix 표 2와 같이 답이 왔다. 대회측에 문의한 결과, GPS Jamming 설정을 함부로 바꿀 수 없다는 입장이었다. GPS 안테나의 위치를 높여서 성능을 개선하려 시도하였으나 지면과의 multipath로 인해 오차가 증가할 수 있다. 따라서 GPS 수신기를 구매할 필요가 있다. 이는 Appendix 표2를 참고하였다.

2. 지역경로 계획

- 지역경로계획에서 산출된 조향 명령의 채터링 현상으로 인하여 모터의 무리를 준다. 차량이 순간적으로 움직이는 경우 장애물의 라이다 데이터 자체의 분포가 떨어지는 것을 확인했다. 이런 경우 DBSCAN 밀도를 유동적으로 조절해야 한다.
- 장애물을 OFF 원형 근사한 장애물을 안정적으로 회피하지만, 높은 밀집도를 가진 장애물 사이를 지나가는 경우, 장애물 사이를 들어가는 것이 아니라 아예 회피하여 지나가는 경우가 생긴다. 이를 극복하기 위해 물체를 원형으로 근사하는 계수(마진 계수)를 차량의 위치와 상황을 고려해 유동적으로 적용한다.

3. 미션 알고리즘

- **협로주행:** 교내에서 테스트할 때 발생한 문제로 가로수, 가로등과 같이 러버콘과 비슷한 폭의 물체가 있을 경우 러버콘으로 가정하는 경우가 있으며, 러버콘 간격을 넓게 설정할 경우 잘못된 선택을 하는 경우가 있었다. 이를 해결하기 위해 카메라를 같이 사용하는 방식을 제시한다. 카메라를 통하여 러버콘과 다른

물체를 판별할 수 있기 때문이다. 추가적인 해결방식으로는 OGM 생성이 있다. 첫번째 바퀴는 해당 알고리즘으로 주행하며 OGM을 생성하고, 이후 바퀴부터 OGM 기반으로 최적화된 경로를 기반하여 주행할 수 있다. 다만 트랙의 초기 위치 및 자세각을 모르기 때문에 이를 초기에 추정하는 알고리즘이 필요하다.

- **신호등/배달 미션/평행주차 미션:** 올해 HADA팀은 배달, 신호등, 평행주차 미션에 대한 실험을 하지 못했다.

4. 하드웨어

- NF 소나타 윈도우 모터를 사용하여 조향 실험을 진행한 결과, 다음과 같은 문제가 발견되었다.
 - 1) **필요 토크 증가에 따른 동작 멈춤:** 0V에서 2.5V까지 조향 명령에 해당하는 전압을 주었을 때 타이 로드가 일자로 정렬되며, 이로 인해 회전에 필요한 토크가 순간적으로 급격히 증가했다. 이 상황에서 모터에서 스위치 소리가 나며 몇 분간 동작이 멈췄다. 정확한 원인은 불명확하나, 내부 온도 기반 스위치가 고부하로 인해 발생한 발열을 감지하여 모터를 비활성화한 것으로 추정된다.
 - 2) 아스팔트 주행 시 마찰력 증가로 더 많은 토크가 요구되었고, 과열로 인해 스위치가 작동하여 조향 불능 상황이 발생했다. 임시 해결 방안으로 조향각을 20°로 제한하여 최소 회전반경을 3.5m로 설정했다. 그러나 최소 회전반경이 3.5m(회전 지름 7m)인 상태에서는 트랙 주행 및 U턴 상황에서 만족스러운 성능을 기대하기 어렵다.

$$R = \frac{\text{축간거리}}{\sin(\text{외측조향바퀴의 조향각도})} = \frac{1.2}{\sin(20^\circ)} = 3.5m$$



Figure 41 모터 내부 온도 기반 전류 차단 모듈

해당 문제를 해결하기 위해 “웜감속기어모터 WGM88-75112 DC 24V”를 구매하여 교체하였다. 해당 모터의 주요 사양은 다음과 같다.

- 1) 감속비: 75

2) 정격 토크: $280\text{kgf} \cdot \text{cm}$

3) 회전 속도: 40 [RPM]



Figure 42 월감속기모터 WGM88-75112 DC 24V

• 교체한 모터의 성능 평가 및 문제점

차량의 지면에서 최대 조향각을 주었을 때도 즉각적으로 회전하며 발열 문제가 발생하지 않았다. 하지만 제어기 설계 실험 도중 최대 조향 상태에서 빠른 속도로 회전시키는 과정에서 과열음이 발생, 이후 모터가 헛돌기 시작했다. 모터를 분해한 결과 내부 기어가 플라스틱으로 제작되어 파손된 것을 확인했다.

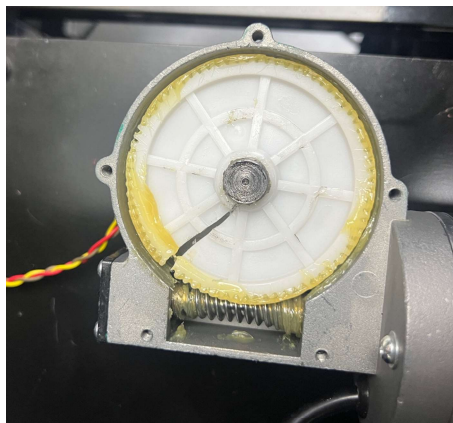


Figure 43 부서진 헬리컬 기어

○ 파손된 헬리컬 기어 스펙:

- 바깥 지름: 78.4 mm
- 안쪽 지름: 75.4 mm
- 기어 두께: 12 mm
- 축 직경: 14 mm (꼭 끼임 공차)
- 톱니 높이: 1.5 mm

$$m = \frac{D_o - 2h}{Z} = \frac{78.4 - 2 * 1.5}{75} = 1.005 \text{ mm}$$

기어 규격에 맞는 대체품을 검색했으나, 적절한 기어를 찾지 못하여 교체가 불가능한 상황이다. 따라서 다음과 같은 해결 방식을 제안한다.

1. 새로운 High Torque DC Motor 구매 검토

기존 플라스틱 기어의 내구성 문제를 방지하기 위해, 다음과 같은 조건을 만족하는 모터를 탐색 중이다:

- 금속 기어 사용
- 기존 모터와 유사한 크기 및 설치 구조
- 높은 정격 토크 제공

올해 HADA 남은 비용을 통하여 필요 조건에 맞는 제품을 구매할 계획이다.

• 후방 디트로네 구동 모터 문제점

기존에 사용 중인 디트로네S 트랜스엑슬 모터는 스펙상 300W를 필요로 하나, 실제 운용 시 100~260kW의 범위에서 구동 가능하며, 충분한 토크를 제공하지 못한다는 문제점이 있다. 같은 모터를 사용한 자율이동체 차량과 비교했을 때 차체 무게 증가(100kg → 160kg)와 바퀴 직경 증가(12인치 → 14인치)와 같은 변화가 생겼다. 이에 따른 속도 변화는 아래와 같은 계산 결과로 도출하였다.

1. 차체 무게 증가로 인한 속도 감소

$$v_2 = 16\text{km/h} \times \sqrt{\frac{100}{160}} = 12.649\text{km/h}$$

2. 바퀴 직경 증가로 인한 속도 보상

12인치 → 14인치 변환하였을 때 속도비는 다음과 같다.

$$\text{ratio} = \frac{14 \times 2.54}{12 \times 2.54} = \frac{35.56}{30.48} \approx 1.17$$

새롭게 도출한 속도 공식은 다음과 같다.

$$12.649 \times 1.167 = 14.761\text{km/h}$$

최종적으로 약 15km/h로 기존 속도를 비슷하게 유지하는 데 성공하였으나, 경사로에서 자력으로 오르지 못하는 현상이 발생했다. 이는 바퀴 직경 증가 및 축길이 증가로 인한 비틀림 변형이 토크 손실을 초래했을 가능성이 있다. 전동휠체어 모터와 같은 더 높은 토크를 제공하는 모터 탐색을 통해 안정적인 조향축에 동력전달을 목표로 한다.

IV. 결론

HADA팀의 올해 최종 목표는 자율주행 플랫폼을 낮은 단계에서 설계해 자율주행 및 전기차의 하드웨어 및 전자장비에 대한 이해도를 높이는 것이었다. 예년에 비해 올해 하드웨어와 소프트웨어(센서 인터페이스, 알고리즘, 센서 정렬, 멀티 쓰레딩 등)에서 차별점을 두었다. 또한 문서화 및 정규적인 버전 제어를 하기 위해 노력하고 있다. 본 보고서뿐만 아니라 문서화와 프로그램의 인수인계에 기반하여 추후에 더 많은 발전이 예상된다.

본 HADA는 학술 동아리이기에 논문 서치 및 논문 공부, 자료 조사를 통해 학술적인 부분에 집중할 것이다. 기본에 충실하여 온전히 이해한 상태에서 구현하고 실험하는 것이 목표이다. 아직 논문 조사나 자료조사에 있어서 미숙한 부분이 있기에 동아리로서 함께 성장해 나아갈 것이다. 전문가의 설명이 필요 할 때 그리고 현재 상황보고를 드릴 때 지도 교수님께 여쭙볼 예정이다.

HADA는 동아리이기에 학업보다 우선이 되면 안 된다. 기초가 되는 수업에 충실하며 추가적인 학습을 위해 동아리 활동을 진행할 예정이다. 수업에서 배우거나 논문 서치를 통해 학습한 이론을 시뮬레이션과 실험을 통해 비교 분석을 할 예정이다.

참고문헌

- [1] Kim, C. Y., Hong, D., & Ra, W. S. (2018). Autonomous Ground Vehicle Localization Filter Design Using Landmarks with Non-Unique Features. The Transactions of The Korean Institute of Electrical Engineers, 67(11), 1486-1495.
- [2] 김찬영, 이제혁, 배진근, & 나원상. (2016). 자율주행 자동차 응용을 위한 예측 통합조향장 기반 이동 장애물 회피 알고리즘. 대한전기학회 학술대회 논문집, 1468-1469.
- [3] Lv, Z., Zhao, L., & Liu, Z. (2013). A path-planning algorithm for parallel automatic parking. In *3rd IEEE international conference on instrumentation, measurement, computer, communication and control* (pp. 474-478).
- [4] 신희석, & 김정하. (2018). 네 개의 원호를 이용한 자동 평행주차 Path Planning 개발. 한국자동차공학회 춘계학술대회, 689-693.
- [5] AMZFormulaStudent . (2017). <https://youtu.be/FbKLE7uar9Y?feature=shared>.

Appendix

표 1 HADA 4 2024년도 사용 내역

사용 내역	비용 (만원)
디트로네 차량 구매 및 플랫폼 제작	919.5
배터리	118
PCB, 소자, 컴퓨터 부품	106.2
아크릴	15.5
카메라 * 2	38.1
컴퓨터 1 (메인 컴퓨터)	100
컴퓨터 2 (인지용 컴퓨터)	125.1
사전 주행 및 대회 비용 (숙박비, 유류비, 렌트비, 용달비 포함)	452.1
대회 참가비 (인당 3만원)	30
총합	1904.5

표 2 심한준 박사 메일 내용 일부

일단 제가 말씀하신 부분에 대해 파악한 바로는,

- 1) GPS 전파 교란기 상황이 주행상황 내에 존재함
- 2) 현재 시작지점에서는 GPS 전파 교란 상황이 있으면 안되는데 영향을 받음
- 3) 동일하게 적용 받는 팀이 2-3팀 존재

인것 같습니다.

주어진 상황 만을 놓고 판단을 해 보겠습니다.

1) GPS 재밍의 경우에는 수신기의 신호 획득을 방해하는 원리입니다. 신호 및 시스템에서 convolution과 동일한 것인데, 재밍은 획득을 못하도록 해당 주파수의 신호에 큰 잡음을 섞는 것과 동일합니다. 친구랑 이야기를 하는데 옆에서 큰 소리로 소리지르면 안들리는 것을 이해해 보시면 됩니다.

2) 그럼 왜 어떤 팀은 되고 어떤 팀은 안되느냐... 이것은 신호 획득에 대한 하드웨어 문제입니다. 비싼 장비 쓰면 당연히 잘되겠지요. 제가 재밍 쪽에 대해서 실제로 실험한 바는 없지만(이게 잘못 하면 아주 큰 문제가 됩니다.) 현재로서는 안테나 보다는 수신기 성능 문제가 큰것으로 보입니다. 노바텔 OEM7720이 매우 고가 수신기는 아닌것 같아요. 그래서 신호추적이 잘되는 것도 성능 문제라고 해석됩니다.

3) 어떤 팀은 되고 어떤 팀은 안되는 이유도 하드웨어의 차이로 보이는데, 이것은 주최측에서 GPS 재밍 신호의 세기를 낮춰야 해결되는 문제입니다. 주최측에서 다양한 하드웨어를 가지고 확인 안해본것 같습니다.

도움이 되셨을지 모르겠으나 현재 저에게 주어진 상황만으로는 해결할수 있는 부분이 없어보입니다. 좋은 수신기 쓰면 신호 추적 더 잘될거 같아서, 비싼 하드웨어를 가진 팀만 초기 신호를 잡을수 있는걸로 보입니다. 이것은 주최측에서 고려해야될것 같다는 생각입니다.

혹시 더 문의주실 내용 있으시면 카톡으로 연락부탁드립니다. 010-4482-0808

심한준 드림

