**Modular Pipeline 전체 보고서**

고소실 월요일 분반

20211530 김희진

1. **실험목표**

총 3주차에 걸친 자율주행 모듈공정 실험에 대해 전체적인 흐름과 구현모듈에 대해 정리하는 시간을 가진다. 또한, 본인이 구현한 모듈공정 기반 자율주행 에이전트의 행동을 파라미터에 따라 분석해본다.

1. **전체적인 흐름**

해당 실험은 lane detection, path planning, vehicle control로 이루어진다. Lane detection 주차에서는 차선을 감지하여 주황색 선으로 나타낸다. path planning 주차에선 lane detection을 수행한 것을 바탕으로 자동차가 나아가야 할 길을 예측하여 점으로 나타낸다.

스크린샷, 다채로움, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Lane detection과 path planning을 구현한 결과는 위와 같다.

Vehicle control주차에서는 lateral control과 longitudinal control을 구현해 우리가 직접 조종하지 않아도 앞선 lane detection과 path planning을 바탕으로 자동차가 움직일 수 있게 하였다. 이때, lateral control에서는 stanley controller를 사용했고 longitudinal control에서는 pid controller를 사용했다.

1. **1주차 : Lane detection에 대한 분석**

해당 주차에서는 차선 탐지를 위한 모듈을 구현하는 것을 목표로 했다.

* 1. 구현한 부분에 대해 함수별로 살펴보고 해당 주차의 전체적인 흐름도 살펴보자
* Cut\_gray()

인자로 받은 이미지를 차량 앞부분만큼 잘라내고 그레이 스케일로 변환하는 함수다.

* Edge\_detection()

Cut\_gray()를 거쳐 그레이 스케일로 변환된 이미지에서 에지를 탐지하기 위해 그레디언트를 계산하고 설정된 임계값(threshold)보다 작은 그레디언트를 무시하여 차선으로 인지될 수 있는 유의미한 에지만을 추출한다.

우리가 자동차를 운전할 맵에는 따로 장애물은 없지만, 잔디에 사각형 모양으로 무늬가 있다. 임계값은 이 무늬를 거르고 차선을 탐지해줄 정도의 값만 설정해주면 된다는 말이다.

* Find\_maxima\_gradient\_rowwise()

각 행별로 로컬 최대값을 찾아서 차선이 될 가능성이 있는 지점을 탐지한다. 이 과정에서 scipy.signal.find\_peaks()를 사용한다.

* First\_find\_lane\_point()

자동차의 앞부분에서 시작하여 아래로 내려가면서 첫번째 차선 지점을 찾는다. 해당 함수의 결과가 차선의 시작점을 결정하고 스플라인 피팅 과정의 기준점이 된다.

* Lane\_detection()

앞선 함수들을 이용해 얻은 정보를 바탕으로 실제 차선을 탐지하는 함수이다. Find\_first\_lane\_point함수에서 리턴된 첫번째 차선 지점에서 시작하여 find\_maxima\_gradient\_rowwise를 사용해 차선 지점을 식별한다. 찾은 차선 지점들을 바탕으로 왼쪽 차선과 오른쪽 차선에 대해 spline 피팅을 수행하여 연속적인 차선 경로를 생산한다. 이때, spline 피팅에 실패한 경우 이전단계에 저장한 차선정보를 사용하도록 한다.

* Plot\_state\_lane()

앞서 수행한 결과를 시각화 하는 함수이다.

② 파라미터에 따라 해당 실험의 결과가 어떻게 달라지는지 살펴보자.

-distance\_maxima\_gradient

이 값을 높게 설정한다면 차가 코너를 돌 때 차선을 올바르게 감지할 확률이 낮아지게 된다. 다음은 이 값을 10으로 설정했을 때의 결과이다.

스크린샷, 그래픽, 그래픽 디자인, 다채로움이(가) 표시된 사진

자동 생성된 설명

한 화면에 급격한 커브가 나올 경우 distance\_maxima\_gradient를 높게 설정하면 lane detection이 잘 되지 않는 것을 확인할 수 있다.

반대로 이 값을 너무 낮게 설정한다면 차가 차선과 근접했을 때 차선을 올바르게 감지할 확률이 낮아진다. 다음은 이 값을 1로 설정했을 때의 결과이다.

스크린샷, 그린, 다채로움, 픽셀이(가) 표시된 사진

자동 생성된 설명

자동차가 차선에서 조금 벗어났을 때 lane detection이 올바르게 이뤄지지 않는 것을 확인할 수 있다.

* Cut\_size

Cut size를 크게 설정했을 때에는 차선 감지가 자동차 앞바퀴부터 시작된다. 그렇기 때문에 코너를 돌 때나 자동차가 lane의 가장자리로 달릴 때 제대로 차선 감지가 되지 않는 문제가 발생했다. 다음은 cut\_size를 68로 설정했을 때의 결과이다.

스크린샷, 그래픽 디자인, 픽셀이(가) 표시된 사진

자동 생성된 설명

Cut size를 줄여주었을 때 위와 같은 문제점이 해결되었다. 아래는 cut\_size를 67로 설정해준 결과 화면이다.

스크린샷, 다채로움, 픽셀, 그린이(가) 표시된 사진

자동 생성된 설명

* Gradient\_threshold

앞서 언급한대로, 우리가 자동차 운전을 할 맵에는 별다른 장애물이 없어서 이 값을 크게 해주거나 작게 해주는 데에 큰 차이가 없었다.

* Spline\_smoothness

Spline 함수의 부드러움 정도를 조절하는 변수이다. 즉, 이 값이 커지면 전체적인 차선의 경향성을 부드럽게 표현하고 이 값이 작아지면 spline이 데이터 포인트들에 더욱 밀착되어 표시된다. 이 값을 조절해봤지만 육안으로 큰 차이가 느껴지지 않아 건드리지 않았다.

③ 실험을 수행하며 어려웠던 점

화면에 도로가 두 선으로 나타나는, 즉 급격한 커브가 나타날 때 정확한 차선 감지가 안되어 어려움을 겪었다. 그래서 여러 파라미터들을 바꿔보던 중 cut size를 변화하면 차선감지 시작점만 바뀐다고 생각했는데 전체적인 결과 또한 바뀌는 것을 발견했다. 그래서 cut size를 줄이고 이에 맞는 가능한 작은distance\_maxima\_gradient값을 찾아 이 문제를 해결했다.

1. **2주차 : path planning에 대한 분석**

해당 주차는 차량의 주행경로를 최적화하는 모듈을 구현하는 것을 목표로 한다.

* 1. 구현한 부분에 대해 함수별로 살펴보고 해당 주차의 전체적인 흐름도 살펴보자
* Curvature

경로의 곡률을 계산하는 함수다.   
이를 계산하는 방법은 연속된 세 점에서 각 점에서의 벡터를 구하고 이를 정규화 한 후 내적을 계산해 구한다. 이 값들의 합이 곡률의 척도가 될 것이다.

* smoothing\_objective

주행경로를 부드럽게 만드는 목적의 함수로 경로를 최적화하는 과정에 사용된다. 주어진 way point와 way point center사이의 평균 제곱 오차와 curvature를 기반으로 목적함수 값을 계산해 사용할 것이다.

* waypoint\_prediction

lane의 spline을 바탕으로 waypoint를 예측하는 목적의 함수이다. Center와 smooth 두가지 방법을 제공하는데, center는 두 차선의 중간지점을 waypoint로 설정하고 smooth는 이 waypoint를 부드럽게 만들기 위해 최적화 과정을 거치는 방법이다.

* target\_speed\_prediction

주어진 waypoint curvature를 바탕으로 목표 속도를 예측하는 함수이다. Lane의 curvature를 계산하고 이를 기반으로 지수 함수를 사용해 목표속도를 결정한다.

* 1. 파라미터에 따라 해당 실험의 결과가 어떻게 달라지는지 살펴보자.

아래 3주차 분석에서 살펴볼 것이다.

1. **3주차 : vehicle control에 대한 분석**

Stanley controller를 이용해 lateral control을 하는 모듈과 pid controller를 이용해 longitudinal control를 하는 모듈을 구현하는 것이 해당 주차의 목표이다.

Lateral control이란 자동차 주행 중에 차선을 유지하면서 제어를 수행하는데 사용되며 longitudinal control은 자동차의 종방향 즉, 앞뒤 방향을 제어하기 위한 제어장치이다. 각 함수에 작성한 코드는 수학 식을 구현한 것에 지나지 않으므로 자세한 설명은 생략하겠다.

Modular pipeline의 모든 과정이 끝났음으로 우리는 전체적인 실험결과를 확인해 볼 수 있으며, 파라미터들을 조절하며 결과가 어떻게 바뀌는지 관찰하고 최적의 파라미터 값을 찾을 것이다.

* 1. 파라미터에 따라 해당 실험의 결과가 어떻게 달라지는지 살펴보자.

실험 결과를 보기위해 각 파라미터의 비교에서 해당 파라미터를 제외한 파라미터들은 default값으로 설정해주었다.

* Max speed

말 그대로 최대 속력을 제한하는 파라미터다.

아래는 max\_speed를 60으로 뒀을 때의 결과이다

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

아래는 max\_speed를 100으로 뒀을 때의 결과이다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

Episode 0, 1 2, 5, 7에서는 max\_speed가 60일 때와 비교했을 때 점수가 확실히 높아졌음을 확인할 수 있지만 episode 3, 8, 9는 오히려 점수가 줄어 trade off가 있음을 확인했다. 이로 인해 해당 에피소드들은 커브가 많은 맵일 것이라는 예측을 했으며 속도를 높이는 것은 점수를 올려주지만 커브가 많은 맵에서는 오히려 낮은 점수를 야기하기에 적당히 조절해야 함을 느꼈다.

* Exp\_constant

이는 target\_speed\_prediction()에서 지수함수에 사용되는 상수로 curvature에 따른 target speed의 감소율을 조정하는데 사용된다. 즉 이 값이 커지면 곡률이 증가할 때 속도를 많이 감소하겠다는 뜻이다.

아래는 exp\_constant를 4.5로 뒀을 때의 결과이다

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

아래는 exp\_constant를 10으로 뒀을 때의 결과이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

exp\_constant를 4.5로 설정했을 때 눈에 띄게 점수가 낮았던 episode 3, 9의 점수가 exp\_constant를 10으로 설정했을 때 많이 올랐음을 확인할 수 있다. 이로 인해 episode 3과 9는 커브를 꺾을 때 속도를 낮추는 것이 필요한 맵이므로 급격히 꺾이는 커브나 구불구불한 도로를 가진 맵임을 예측할 수 있었다.

* Weight\_curvature

이는 lane의 curvature에 부여되는 가중치이다. Weight\_curvature가 높으면 경로를 줄이는데 중요성을 부여해 경로를 더 부드럽게 만든다.

아래는 weight\_curvature를 40으로 뒀을 때의 결과이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

아래는 weight\_curvature를 60으로 뒀을 때의 결과이다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

전체적인 결과와 양상은 비슷하나 weight\_curvature가 높아졌을 때 episode 3의 점수는 높아지고 episode 8의 점수는 낮아지는 결과를 확인했다. 이로 인해 weight-curvature는 3번 episode와 8번 episode에 영향을 준다는 사실을 확인했고 episode 3번은 커브가 계속 있는 맵일거라 예상했다.

* Etc

나머지 파라미터(Kp, Ki, Kd, Vmin)들은 변경했을 때 결과에 의미 있는 차이가 보이지 않아서 건드리지 않았다.

* 1. 실험을 수행하며 어려웠던 점

스코어를 에피소드별로 확인할 수 있었지만 해당 에피소드가 어떤 맵에 해당하는 지를 알 수가 없었다. 그래서 어떤 맵에서 주행이 제대로 안되는지를 파악하는게 어려웠다. 어떤 파라미터 값을 변경했을 때 에피소드의 점수가 올라가고 내려가는지에 대한 관찰을 바탕으로 해당 에피소드가 어떤 맵일지 추측했고 이를 통해 앞선 어려움을 해결했다.

예를 들어, 에피소드 9번 같은 경우에는 exp\_constant를 높였을 때 다른 경우와 비교해 스코어가 높아짐을 확인할 수 있었다. 그래서 9번 맵이 구불구불한 도로가 지속되는 맵이라는 판단이 들어서 최대속도를 낮추고 exp\_constant를 높이고 curvature를 낮추는 등의 시도를 통해 최적의 결과를 찾았다.