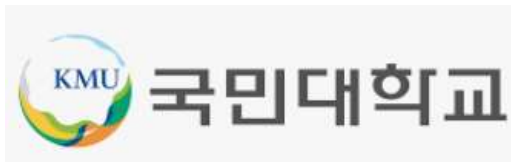


다학제간캡스톤디자인_중간발표

2025.04.29(화)

자동차IT융합학과

20203334 김지원 / 20203336 남대현 / 20203358 이동원



1. 프로젝트 개요

DeepRacer 대회를 준비하기 위한 핵심 목표

- 보상함수 설계: 주행 성능과 일반화 능력을 극대화 하기 위한 보상 구조 설계방안 탐색
- 보상함수 학습 진행: Curriculum Learning, Domain Randomization 등을 포함한 효과적인 학습 전략 수립
- 강화학습 외 rule-based 방법: 강화학습을 보완하거나 병행할 수 있는 전략 구성
- SimtoReal Gap: 학습된 에이전트를 실제 차량에 적용할 때 발생할 수 있는 요인 문제점 및 해결 방안 탐색

각 Task 별 핵심 목표

- 하나의 Task 안에서도 세부적으로 고려해야 할 Sub-task 존재
- Sub-task간의 우선순위를 명확히 설정하고, 그에 맞춰 Reward Function을 구조적으로 설계하여야 함

Time - Trial

- 가능한 한 짧은 시간 내 트랙 완주
- 빠른 속도만으로는 안정적인 완주가 어려우므로, 주행의 안전성과 효율성을 동시에 고려하여 설계

Obstacle Avoidance

- 경로상의 장애물을 인식하고, 충돌 없는 회피
- Time-Trial과 마찬가지로, 안전성과 효율성을 동시에 고려하되, 장애물 회피 기반으로 설계

Random Tracks

- 트랙의 랜덤성을 고려하여 트랙 고유 파라미터 (ex. waypoint index)에 의존하는 보상 설계 지양
- 일반성과 강건성에 중점을 두어, 간결하고 명확한 보상 구조 설계

2. 개발 주안점

Action space(Discrete vs Continuous)

- 강화학습에서의 행동 공간은 학습의 난이도 및 정밀도에 큰 영향을 미치므로 알고리즘과 학습 전략에 따라 적절하게 선택하여야 함
 - Discrete space: 행동이 유한한 집합에서 선택되며, 모델링이 간단하지만 정밀한 제어에 덜 유연함
 - Continuous space: 행동이 연속된 범위 내에서 선택되며, 정밀한 제어가 가능하지만 학습이 어렵고 최적화가 복잡함

Action space에 따른 학습 전략

- 비교적 튜닝이 쉽고 안정적인 PPO(Proximal Policy Optimization)로 전체적인 보상 구조를 정립하고, Discrete Action space로 정책 수렴 도모
- 안정화 후 정밀한 제어를 요구하는 시점에서 Continuous Action space로 확장하여 실제 주행에 가까운 환경을 구축하고, Off-policy 기반 SAC(Soft-Actor-Critic)를 적용하여 연속된 제어 문제에서의 성능 달성
- 초기에는 안정성과 단순성을 사용하여 전체적인 Framework를 구축하고, 이후에는 정밀성과 일반화 성능을 향상하는 방식으로 학습

PPO + Discrete -> PPO + Continuous -> SAC + Continuous

● 2. 개발 주안점

Rule-based Method

- 강화학습 기반 정책을 보완하거나 대체 수단으로 도입

1) 경로 추종(Path Following)

- 현재 차량의 x, y 좌표와 Waypoint 기반 주행 경로 정보 활용
- Pure Pursuit, Stanley와 같은 횡 방향 제어 방식 및 경로의 곡률 정보를 기반으로 차선 정보 추정 기능 적용

2) 장애물 회피를 위한 로직 구성

- [is_left_of_center]와 [objects_left_of_center] 파라미터 활용
- 차량과 장애물이 같은 방향에 위치하면 반대 방향으로 회피, 그 외에 상황에서는 직진을 유지하는 로직 설계

SimtoReal

- 목표: 시뮬레이션 환경에서 학습된 에이전트를 실제 차량에 적용할 때 발생하는 성능 차이(SimtoReal gap)를 최소화

- Deepracer 논문 분석: 시뮬레이션과 현실 간의 간극(SimtoReal gap)을 줄이기 위한 구체적인 방법론 파악 후 정리
- 강화학습 후 SimtoReal 성능 평가: 학습된 에이전트를 실제 환경에서 주행한 뒤, 정량적 및 정성적 평가 수행
- 프로젝트 적용 가능성 검토: 실제 주행 결과를 바탕으로, 논문에서 제시한 기법을 프로젝트에 적용할 수 있는 방안 탐색

3. 시도한 방법들

보상함수 설계 - Time-Trial

- Time-Trial에서는 안전성(Safety)와 효율성(Efficiency) 두 가지 Task를 정의하고 그 안에서도 세부적으로 고려해야 할 Sub-task 존재
- Sub-task는 독립적인 Reward 함수로 구성되며, 이들을 선형 결합하여 총 Reward 구성(ex. $Reward = w_1R_1 + w_2R_2 + w_3R_3$)

안전성(Safety)	효율성(Efficiency)
<u>Sub-Task 1)</u> Waypoint 기반 중심선 경로를 따른 안정적인 주행 <u>Sub-Task 2)</u> 차량의 주행 방향과 트랙의 진행 방향 정렬하여 경로 이탈 방지 <u>Sub-Task 3)</u> 조향 및 속도 변화의 안정성 확보	<u>Sub-Task 1)</u> 트랙의 형태(직선/좌회전/우회전)에 따른 속도 및 조향 전략 학습 유도 <u>Sub-Task 2)</u> 차량이 트랙을 더 많이 완주하도록 유도하고, 곡선 주행 전략 확보

보상함수의 가중치 조정 문제

- Task-Level에서의 가중치 조정**
 - 우선 설계자의 의도에 따라 가중치(w_1, w_2 등)을 설정하고, 시뮬레이션을 통해 학습 경향 관찰
 - 실험적으로 도출된 경향성을 바탕으로 각 task의 우선순위를 검증하고 보완
- Micro-Level에서의 가중치 조정**
 - 보상함수 내부에서 사용되는 미시적인 연산 방법(ex. $reward *= 1.5$)과 계수 값에 대해서도 일정한 기준 정립
 - Task 간 구조적으로 일관되도록 유지하고 학습이 편향되는 것을 방지하기 위해 통일성 유지

3. 시도한 방법들

Curriculum Learning

- 학습 목표: 안전성/효율성 보상함수 각각 설계 후, 단계적으로 학습하는 Curriculum Learning 전략을 통해 정책의 안전성과 일반성 향상
 - 1) Safety -> Efficiency 학습: 안전한 주행 전략부터 학습한 후, 속도/조향 최적화를 통해 점진적으로 고도화
 - 2) Efficiency -> Safety 학습: 시간 단축 전략으로 exploration을 유도한 후, 안정화하는 방향으로 유도
 - 3) 동시 학습: Safety와 Efficiency를 동시에 적용하여 통합된 목표 학습

Curriculum Learning 실험 전략

- DeepRacer의 Clone 기능을 활용하여 Curriculum Learning 진행 예정



- 시뮬레이션 시간은 1시간으로 고정
- 보상함수를 나누어서 시뮬레이션 진행 시, 정책의 수립 여부와는 관계없이 각 보상함수의 시뮬레이션은 30분으로 고정
- 수립 여부에 대한 명확한 판단 기준은 시뮬레이션 결과 분석 후 추후 수립 예정

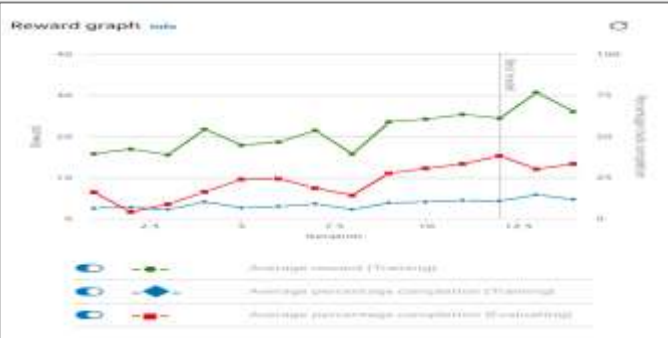
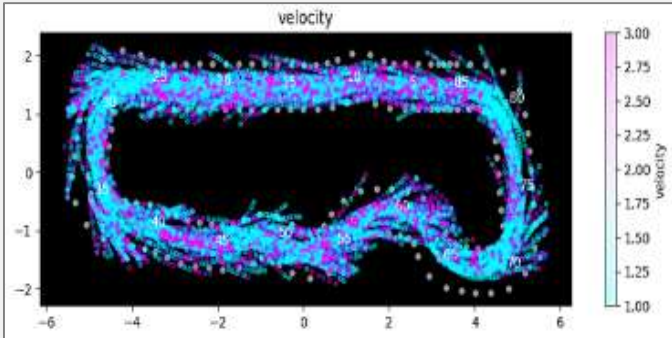
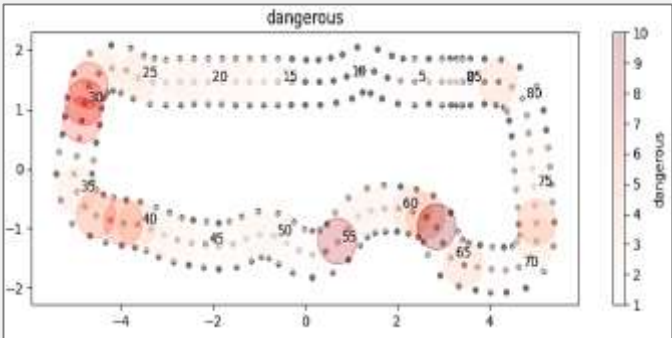
3. 시도한 방법들

Curriculum Learning 성능 비교

- Curriculum Learning과 동시 학습의 성능을 비교할 때, **각 학습 전략에 최적화된 가중치와 파라미터를 사용**
- 두 방식을 같은 파라미터로 통일하면 각 전략의 잠재력을 상쇄할 수 있으므로, '최상의 성능'을 낼 수 있도록 학습한 후 성능 비교

Reward 학습 평가 기준: 로그 파일 분석

- 에이전트가 어떻게 주행했는지를 분석하고 보상함수의 효과를 평가하는 핵심적인 자료
- 보상 함수 그래프, 구간별 속도 및 위험도를 추적**함으로써 학습이 어떻게 진행되었는지 평가

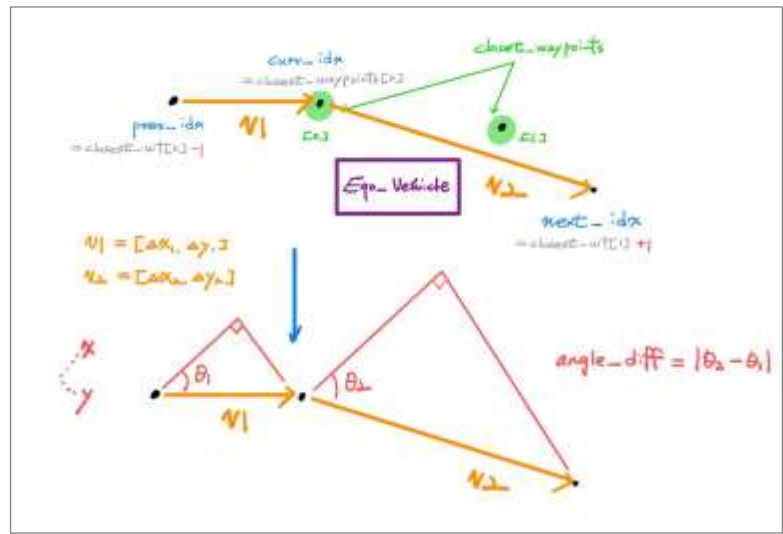
보상 함수 그래프	구간별 속도 추적	구간별 위험도 추적
<div></div> <div><p>➢ Graph가 상승하며 수렴하는 형태가 이상적</p><p>➢ 진동 및 하락했을 시, 보상 구조 / 탐색에 문제가 있는지 검토 필요성 있음</p></div>	<div></div> <div><p>➢ 보상함수에서 의도한 속도로 주행하였는지 평가</p><p>➢ 곡선 구간에서 감속하고, 직선 구간에서는 속도를 높이는 경향이 나타났는지 분석</p></div>	<div></div> <div><p>➢ 트랙 이탈이 발생한 구간 파악</p><p>➢ 급커브 이후 가속을 시작하는 타이밍에서 위험도가 높게 나타나는 경향이 있음</p></div>

3. 시도한 방법들

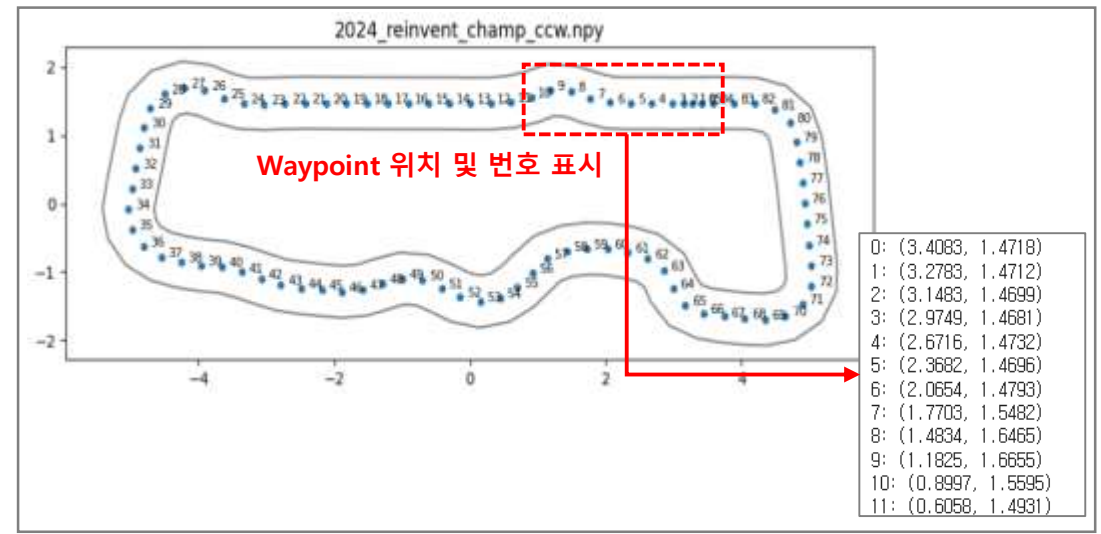
효율성(Efficiency) 보상함수에서의 Map Data 활용

- Github에 공개된 AWS DeepRacer 트랙의 **Waypoint 데이터(x, y)를 시각화** 하여 트랙의 전체 형태 및 구간별 특성 분석
- 연속된 세 Waypoint를 기준으로 두 개의 벡터(v1, v2)를 생성하고, 이 벡터들 사이의 **각도 차이(angle diff)를 계산하여 곡률(curvature) 추정**
- **angle_diff의 절댓값이 Threshold보다 작으면 직선으로 분류**하고, 그 외 부호에 따라 좌/우회전으로 분류

< angle_diff 계산 >



< 트랙의 Waypoint 출력 >



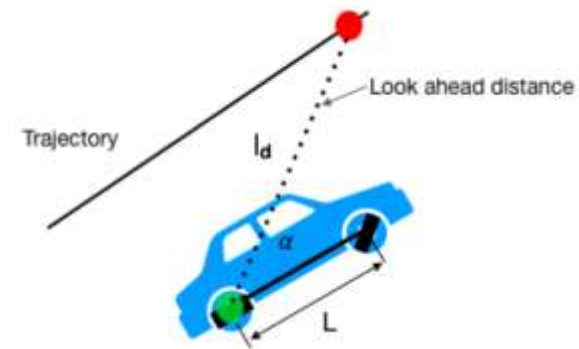
angle_diff의 절댓값 Threshold 조절

- Threshold 낮게 조절: 커브 구간을 엄격하게 판단하므로 속도 제어에 유리하나, 자주 감속하게 될 시 **소극적인 주행을 할 우려가 있음**
- Threshold 높게 조절: 직선으로 간주되는 구간이 넓어 가속에 유리하나, **커브에서 급가속하여 트랙에서 벗어날 수 있음**

3. 시도한 방법들

보상 함수 고도화 전략: Pure Pursuit 조향 보정

- Pure Pursuit은 단순하고 직관적인 기하학 기반 경로 추종 알고리즘으로, 차량이 일정 거리 앞의 목표점(lookahead distance)을 향해 조향각(steering angle) 계산
- 조향각의 방향성을 제시하는 reference로 적합하다고 판단해 보상함수 구조에 추가



학습 목표

- Curriculum Learning 기반 학습 구조를 기반으로, Safety와 Efficiency와 같은 기본 주행 전략을 안정적으로 학습한 후 적용할 예정
- 곡률이 큰 상황에서만 보조적인 조향 reference로 사용하여 학습의 정밀도 향상 도모

시뮬레이션 시 고려해야 할 내용

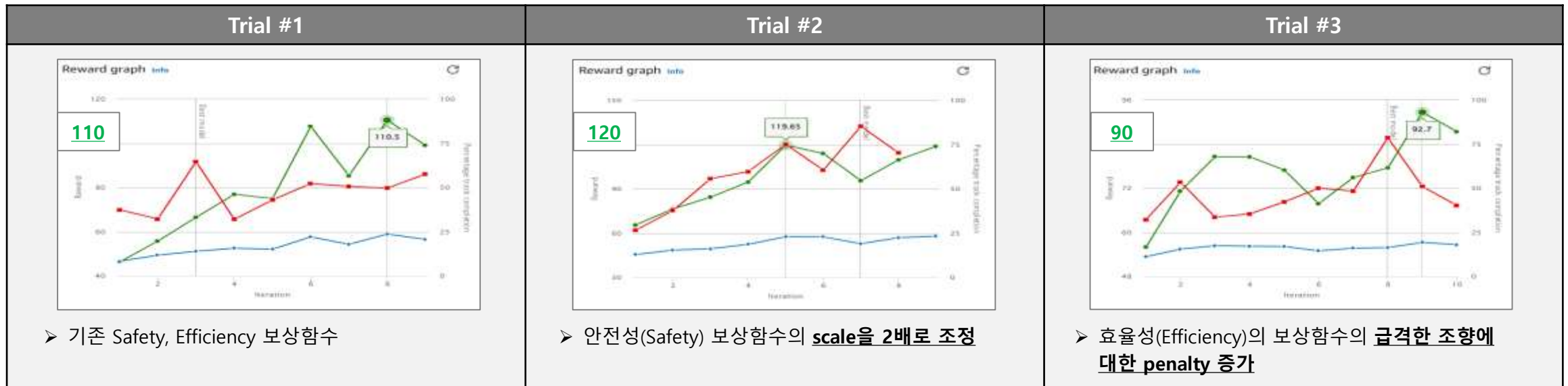
- Pure Pursuit의 조향각이 실제 주행과 맞지 않는 경우, 불필요한 조정이 발생하였는가?
- 기존 학습한 Safe 및 Efficiency 전략이 안정화된 상황에서, 새로 적용한 정책이 기존 학습과 충돌하지는 않았는가?
- Pure Pursuit을 적용해야 할 곡률의 임계값이 적절하게 설정되었는가?

4. 중간 결과

Curriculum Learning #1(Safety -> Efficiency)

- PPO + Discrete, Action space는 [**Steering Angle: +30,+15,+5,0 / Speed = 1,2,4**] 총 20개로 정의
- 정책 수렴 여부와는 관계없이 Safety 30분 학습 후, Efficiency 30분 학습

평균 보상
 평균 완료율(평가)
 평균 완료율(학습)



학습 분석

- Trial #1: 직선 구간에서는 대체로 트랙을 잘 따라갔으나, 급커브 구간에서는 **과도한 조향으로 인해 잦은 이탈**이 발생함
- Trial #2: 총 Reward와 평균 완료율(평가)이 **전반적으로 우상향**했으나, 급커브 구간에서 여전히 **조향이 불안정함**
- Trial #3: 급커브 구간에서 **조향 부족으로 트랙 이탈**, 완만한 구간에서는 **과도한 조향이 발생**하는 등 조향 제어가 불안정하게 나타남

4. 중간 결과

로그 분석 [Trial #2]

구간별 속도 추적	구간별 Reward 추적	구간별 위험도 추적
 <ul style="list-style-type: none"> ➢ 대체로 트랙에서 낮은 속도로 주행 ➢ 직선 구간에서 의도한 속도에 충분히 도달하지 못함 	 <ul style="list-style-type: none"> ➢ 대체로, 트랙 중앙을 따라 reward가 고르게 이어짐 ➢ 급커브 구간에서 의도한 대로 인코스 주행 시 높은 reward를 기록함 	 <ul style="list-style-type: none"> ➢ waypoint 30 부근 급커브 구간에서의 위험도가 가장 높음 ➢ 직진 구간 이후, 급커브에 제대로 대응하지 못하는 모습이 빈번히 관찰됨

분석 및 고찰

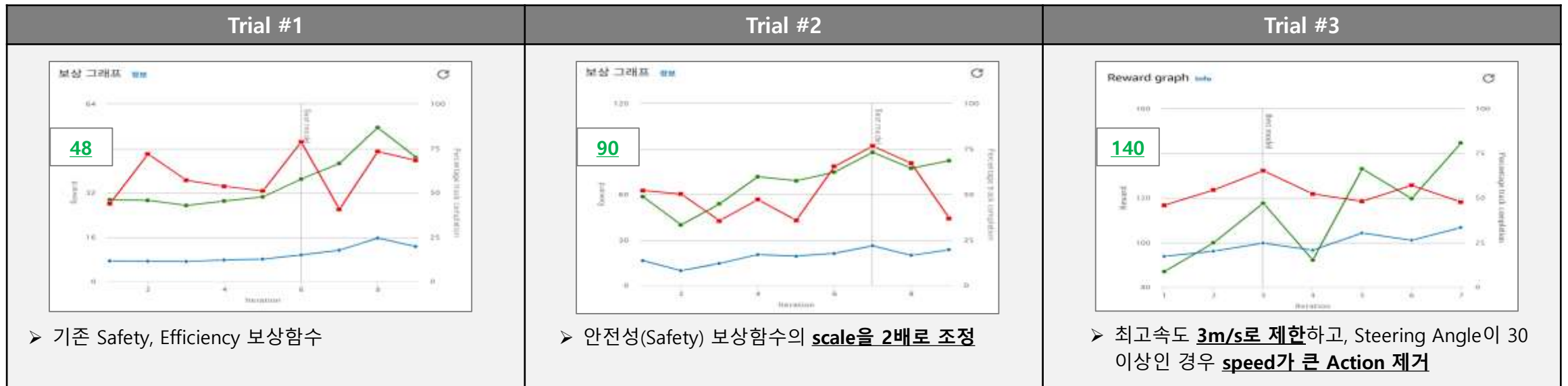
- **위험 구간 대응 전략 보완:** Waypoint 30 부근(급커브 구간)에서 높은 위험도가 지속적으로 나타나, 조향 및 속도 제어의 예측 가능성 향상이 요구됨
→ 급커브 진입 전 감속 및 조향 준비를 유도하는 reward shaping 필요
- **속도 학습 개선 필요:** 전 구간에서 속도가 전반적으로 낮게 설정되어 있으며, 직선 구간에서 의도한 속도에 충분히 도달하지 못함
→ 에이전트의 속도 보상 민감도 보완을 위해 직선 구간에서의 가속을 유도하는 보상 도입 검토

4. 중간 결과

Curriculum Learning #2(Efficiency -> Safety)

- PPO + Discrete, Action space는 [**Steering Angle: +30,+15,+10,0 / Speed = 1,2,4**] 총 20개로 정의
- 정책 수렴 여부와는 관계없이 Efficiency 30분 학습 후, Safety 30분 학습

평균 보상
 평균 완료율(평가)
 평균 완료율(학습)



학습 분석

- Trial #1: 인코스 주행을 시도하였으나, 트랙 곡선 구간에서 정밀한 조향을 하지 못하고 **급가속으로 인해 잦은 이탈이 발생함**
- Trial #2: 총 Reward만 약간 상승하였으며 **조향각 및 속도 제어(Action)의 유의미한 변화가 관찰되지 않음**
- Trial #3: 총 Reward가 유의미하게 상승하였고 안정성도 일부 향상되었지만, 여전히 **트랙 곡선 구간에서 이탈이 빈번하게 발생**

4. 중간 결과

로그 분석 [Trial #3]



분석 및 고찰

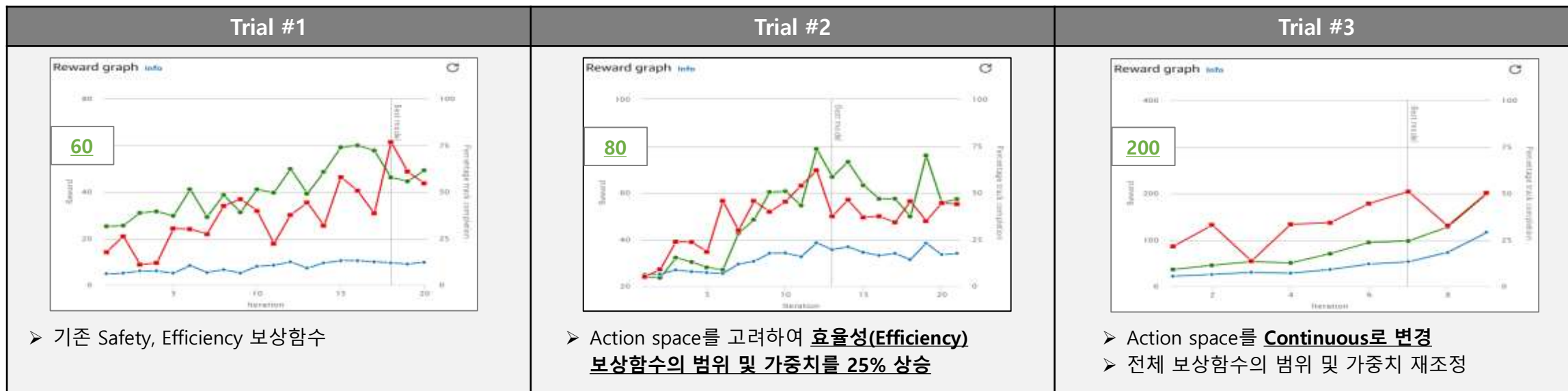
- **학습 순서 및 방법 재검토:** 안전성(Safety) 정책이 수렴되지 않은 상태에서, 효율성(Efficiency) 학습이 진행됨
- 단순히 보상 스케일을 확대하는 것만으로는 부족하며, critical한 상황에 대한 **penalty**를 반영하는 방향으로 보상 구조 개선할 필요가 있음
- Trial #3에서의 Action space 조정이 일정 부분 효과가 있었으나, 속도 및 조향에 대해 **Action space**를 세부적으로 조정할 필요가 있음
- 전체 학습과 각 단계별 보상 함수 설계에 대해 **충분한 시간과 검토가 필요함**

4. 중간 결과

Curriculum Learning #3(동시 학습)

- PPO + Discrete/Continuous, Action space는 [**Steering Angle: $\pm 30, \pm 15, \pm 10, 0$ / Speed = 1,2,4**] 총 20개로 정의
- 정책 수렴 여부와는 관계없이 Safety + Efficiency 보상함수 1시간 학습

평균 보상
 평균 완료율(평가)
 평균 완료율(학습)



학습 분석

- Trial #1: 직선 구간에서 가속이 저조했으며, 곡선 구간에서는 **불필요한 감속 및 잦은 이탈 발생**
- Trial #2: 총 Reward 및 직선 구간 가속이 증가했으나, 여전히 **곡선 구간에서 잦은 이탈 발생**
- Trial #3: 총 Reward가 유의미하게 상승하였고 안정성도 일부 향상되었지만, 여전히 **트랙 곡선 구간에서 이탈이 빈번하게 발생**

4. 중간 결과

로그 분석 [Trial #3]



분석 및 고찰

- 트랙의 형태(직선, 곡선)에 따른 명확한 경계 설정과 보상 구조 개선 필요
- 곡선 구간에서 발생할 수 있는 Critical case를 구체적으로 정의하고, 감속 및 조향 범위를 정밀하게 조정해야 함
- 동시 학습 특성상, 각 보상 항목에 대한 가중치 조합에 따라 학습 방향이 달라지므로 이를 명확히 분석할 필요가 있음
- 복잡한 보상함수 설계가 요구되는 만큼 충분한 학습 시간 확보 필요

5. 토의

논문 참조[A Review of Reward Functions for Reinforcement Learning in the context of Autonomous Driving]

[2. Challenges of Reward Design in the Autonomous Driving Domain] 문단 중

자율주행은 본질적으로 다중 목표 문제이기 때문에, **보상함수는 서로 다른 목표들을 동시에 고려하고, 이들 간의 충돌을 적절히 조율할 수 있는 구조를 갖추어야 한다.** 또한, 자율주행은 문맥(context)에 따라 요구되는 행동이 달라지기 때문에, 보상함수 설계 역시 **문맥을 반영할 수 있어야 한다.** 이 외에도, 어떤 보상함수가 좋을지 명확히 평가하기가 어렵고, 어떤 행동의 효과가 시간 차를 두고 나타나므로 학습이 어려워질 수 있으며, **잘못 설계된 보상 속성이 실제로는 바람직하지 않거나 오히려 위험한 주행 행동을 유도**할 수 있으므로 보상함수 설계 초기 단계부터 세심한 주의가 필요하다.

[4. General Limitation] 문단 중

보상함수는 **일반적으로 설계자의 경험과 직관에 기반**하여 수작업으로 설계되기 때문에, 특정 행동만을 과도하게 강화하거나 설계자가 미처 염두에 두지 못한 시나리오에서는 에이전트가 비합리적인 행동을 학습할 가능성이 존재한다. 더욱이, 보상함수의 **작은 변경조차도 최종적으로 학습된 정책에 상당한 영향**을 미칠 수 있으며, 예기치 못한 방식으로 정책이 왜곡될 수 있다. 이는 보상함수가 올바른 행동을 명시적으로 알려주는 것이 아니라, 어디까지나 그 방향을 암시하는 간접적인 유도 신호의 역할을 하기 때문이다. 결과적으로, 보상이 잘못 설계되었을 경우 의도하지 않은 행동이 강화될 수 있는 위험이 존재하며, 특히 **현실 환경은 복잡하고 다양한 상황이 존재하기 때문에 극단적인 시나리오**까지 포괄하지 못하는 한계도 여전히 크다.

토의의 주안점

- 다중 목표 상황 속에서, **어떤 Task를 우선적으로 고려하여 보상함수를 설계**할 것인가? ex. 안전성(Safety), 효율성(Efficiency)
- 우선순위가 정해졌다면, **설계 의도와 상황적인 문맥**을 보상함수 구조에 어떻게 효과적으로 반영할 것인가? ex. 안전성(Safety) 내 Sub-Task 간 구성
- 설계된 보상함수 구조 내에서 **Sub-Task 간에 보상이 충돌하거나 간섭**하지는 않는가?
- 설계자의 의도를 정량적으로 표현하기 어려운 미시적 요소들은 **어떤 기준으로 설계에 반영**할 것인가? ex. 튜닝 및 연산 방식(reward*=0.9) 등

5. 토의

(1) 보상함수 구조화 및 가중치 조정

- Time-Trial에서는 논문과 같이 안전성(Safety) 및 효율성(Efficiency)을 큰 목표로 두고, 설계자의 의도를 반영한 2~3개의 Sub-Task로 보상함수를 설계
- 초기에는 각 Sub-task의 가중치를 설계자의 의도대로 부여하고, 학습 결과에 따라 우선순위를 검토 및 보완
- $\text{reward}^*=0.9$ 와 같은 미시적인 연산과 및 계수의 값은 통일된 기준을 따르되, 그 안에서 세부적인 이견이 있었음

남대현 학생	김지원 학생	이동원 학생
<ul style="list-style-type: none">▪ 각 task 내에서 사용되는 reward 및 penalty 항목에 대해, 최대한 조건문 및 <u>파라미터를 간결하게 유지</u>▪ 가급 reward는 1.5, penalty는 0.5로 통일하되, 조건 분기가 3개 이상이면 예외적으로 추가 설정	<ul style="list-style-type: none">▪ 조건문이 많더라도 각 상황을 <u>의미 있게 구분할 수 있도록 유지</u>▪ 보상 구조나 파라미터를 지나치게 단순화하기보다는 DeepRacer 예제와 같은 <u>Reference 구조 참고</u>	<ul style="list-style-type: none">▪ reward 항목은 전체 보상의 균형을 위해 상한선을 가급적 일정하게 유지▪ penalty 항목은 트랙 이탈 등의 <u>critical 한 상황에 대해 충분히 반영</u>할 수 있도록 하한선을 인위적으로 고정하지 않아야 함

협의 결과

- DeepRacer 예제 설계 의도를 적극적으로 반영하되, Sub-Task 간 기능 중복 제거를 위해 보상 구조의 단순성과 명확성 확보
- 보상 항목의 독립성과 선형 결합 구조 유지를 통해 정책이 편향되는 것을 최소화하고, 실험적으로 조정이 용이하도록 함

5. 토의

(2) 차선 유형을 분류하여 보상함수 반영

- 효율성(Efficiency) 보상함수에 차선의 유형(좌회전/직진/우회전)을 출력하여
- 차선 유형에 따라 조향 및 속도 조건을 다르게 하여 보상함수를 설계할 수 있으므로, 상황에 맞는 효율적인 주행 전략 유도
- DeepRacer에서 Waypoint index를 직접 활용하는 map data 방식과, 곡률 계산을 통해 차선 정보를 판단하여 보상함수에 반영하는 방식 두 가지 고려

회전구간: 인코스 주행, 직선: 고속 주행

상황에 맞는 효율적인 주행 전략

Map data 방식	곡률 방식
<ul style="list-style-type: none">▪ 도로 형태를 사전에 알고 있어 특정 트랙에서 안정적인 주행 성능을 유도할 수 있으며, Waypoint 번호 기반으로 보상함수 조건 조정에 유리▪ 하지만, <u>트랙에 과도하게 의존(overfit)</u>하므로, 새로운 환경에서 성능이 저하될 수 있으며, 트랙이 바뀔 때마다 <u>index를 재설정해야 하는 번거로움</u>이 있음	<ul style="list-style-type: none">▪ <u>도로 형태를 실시간으로 계산</u>하므로, 사전에 정의된 Waypoint index에 의존하지 않고 트랙의 구조를 자동으로 파악할 수 있음▪ DeepRacer에서는 여전히 waypoint 기반 데이터를 사용하므로 <u>완전하게 독립적이지는 않으며</u>, waypoint 밀도에 따라 곡률 계산이 불안정해 질 우려가 있음

협의 결과

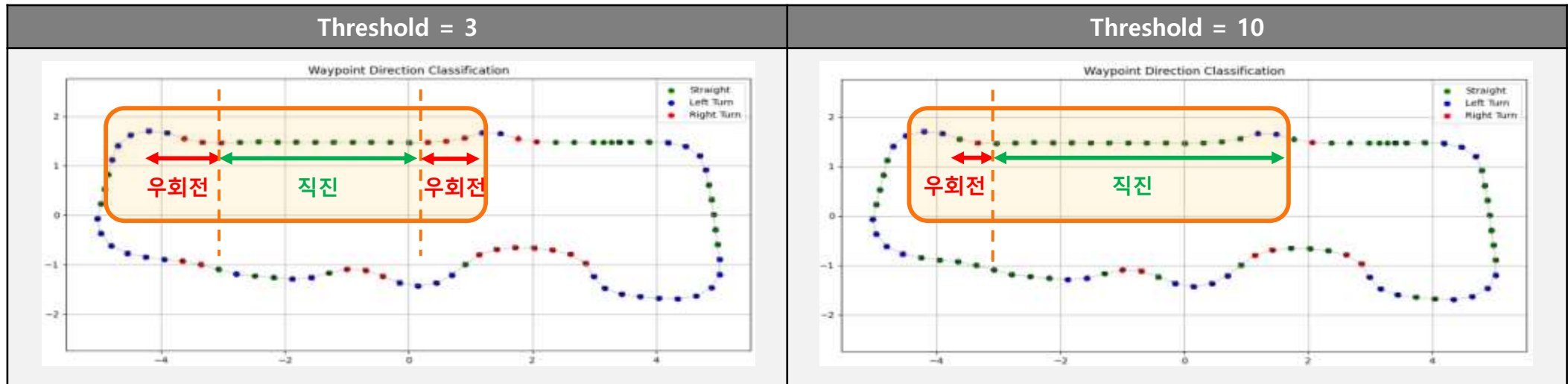
- 사전 제공된 Map data를 기반으로 곡률 계산 알고리즘을 적용해 차선 형태를 미리 분류하고, 보상함수 학습에 반영하는 방식으로 진행
- 도로 형태에 대한 사전 정보가 제공되지 않는 경우에는, 불확실한 환경에서도 강건하게 대응할 수 있도록 보상함수 재구성 필요

Waypoint를 이용하지 않는 보상함수

5. 토의

(3) angle_diff의 절댓값 Threshold에 따른 보상 전략

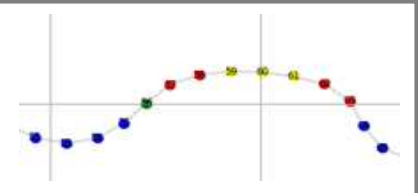
- Threshold 낮게 조절: 자주 감속하게 될 시 소극적인 주행을 할 우려가 있음
- Threshold 높게 조절: 커브에서 급가속하여 트랙에서 벗어날 수 있음



협의 결과

- 시뮬레이션을 통해 **Threshold를 실험적으로 조정**(ex. 차량이 커브 구간에서 급가속하는 경향이 관찰되면 Threshold를 낮춤)
- 또한, **두 가지 Threshold를 사용하는 보상 함수 설계**를 고려하여 세 가지 구간으로 세분화

직선: 6도 이하
완만한 곡선: 6-10도
좌회전/우회전: 10도 이상



5. 토의

(4) Rule-Based 활용 방안

강화학습을 활용할 경우 모두 다른 Agent 활용

- Time-Trial, Obstacle Avoidance는 강화학습을 활용하되, Random Track은 일반성 및 강건성에 초점을 맞추어 Rule-based 활용 고려
- Rule-based를 활용하기 위해서는 ROS가 필수적이며, waypoint에 의존해야 하는 제어 알고리즘은 Random Track에서 활용이 어려움

Rule-based 활용 시 고려해야 할 내용

- ROS 활용: 차량의 현재 위치, heading 등 센서 데이터를 실시간으로 받아야 하므로 Rule-based에서 필수적
- 제어 알고리즘 활용: Random Track에서는 waypoint 자체를 알 수 없으므로, waypoint에 의존하는 Pure Pursuit, Stanley는 부적합

협의 결과

- 카메라로 트랙의 경계 검출 -> 중앙선 추정 -> 그 중앙선에 맞게 PID Controller로 조향 각(Steering angle) 계산
- Throttle(속도)는 복잡한 계산 없이 상수 값(ex. 0.3-0.5) 유지

Edge Detection / Midline Following	Simple PID Controller
<ul style="list-style-type: none">▪ 트랙 경계를 카메라의 이미지에서 <u>검출하여 중앙선 계산</u>(ex. Canny Edge Detection)▪ 트랙 경계가 명확할 시 유용함	<ul style="list-style-type: none">▪ 중앙선과 차량의 heading 차이를 <u>오차로 계산</u>하여, PID Controller을 통해 <u>조향 각 (Steering angle) 조절</u>▪ 간단하게 튜닝이 가능하나, noise 발생 가능

● 6. 향후 계획

향후 계획(2025.04.29~)

- Curriculum Learning 실험 결과 기반 Action space 및 강화학습 알고리즘 선정
 - 1) **Action space**: PPO 알고리즘 기반 **Discrete Action space → Continuous Action space** 순으로 학습
 - 2) **학습 알고리즘**: Continuous Action space 기반 **PPO → SAC** 순으로 학습
- Random Track 주행과 Obstacle Avoidance(장애물 회피)를 고려한 보상함수 설계 및 시뮬레이션 진행
- SimtoReal 적용
 - 1) 학습이 완료된 Agent를 실제 AWS DeepRacer 차량에 업로드하여 **실차 주행 실험**
 - 2) 주행 데이터를 수집하여, **시뮬레이션과 실차 주행 간 차이(SimtoReal Gap) 정량적/정성적 평가**
 - 3) 논문을 참고하여 분석한 **SimtoReal Gap 극복 기법 검토 후 적용**(ex. 도메인 랜덤화, 센서 노이즈 추가, 보상 구조 재설계)