상용 시뮬레이터 기반 차량 동역학을 고려한 강화학습 환경 구축

Development of Simulation Environment for Vehicle Dynamics-based Reinforcement Learning Using CarMaker Simulator

팀 명	TOCA
팀 원	Kim, Junghyo (2018114346) Lee, Hyojae (2020118112) Jung, Sebin (2018111273)
지도 교수	Prof. Han, Kyoungseok

1. Motivation

- With the rapid growth of the autonomous driving industry, research is underway on related driving and behavioral control algorithms.
- IPG CarMaker is a simulation program capable of realistic driving environment emulation.
- The TD3 reinforcement learning algorithm is well-suited for problems in continuous action spaces, such as vehicle speed and steering angle.

2. Project Overview

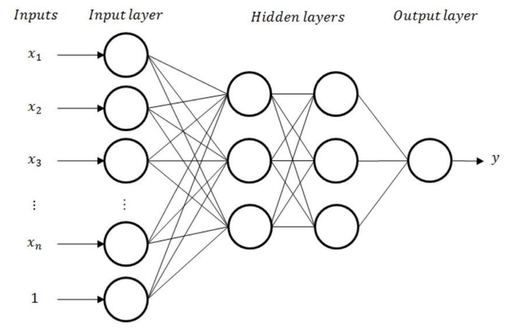
- **TD3 Algorithm**

Algorithm 1 TD3

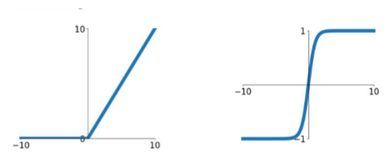
Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \min_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end if
end for

<TD3 Algorithm>



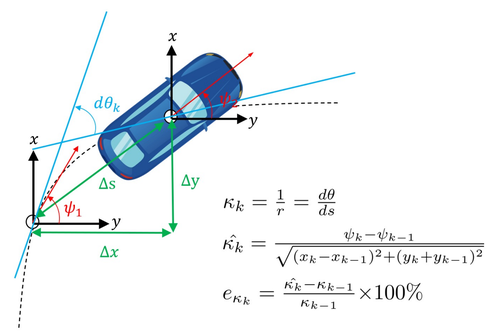
<MLP Structure>



<Activation Function – ReLu, tanh>

- TD3, a member of the Actor-Critic family, adopts the DDPG-based architecture, making it suitable for application in continuous action spaces.
- TD3 comprises two independent Critic neural networks (MLP), utilizing the smaller value between the two calculated value estimates from each network as the target for Q-learning. This approach effectively addresses overestimation and variance issues.

- **RL Environment**

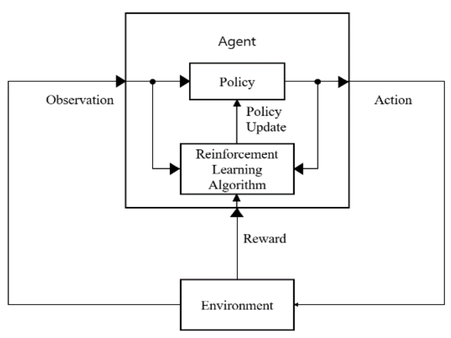


$$\kappa_k = \frac{1}{r} = \frac{d\theta}{ds}$$

$$\hat{\kappa_k} = \frac{\psi_k - \psi_{k-1}}{\sqrt{(x_k - x_{k-1})^2 + (y_k + y_{k-1})^2}}$$

$$e_{\kappa_k} = \frac{\hat{\kappa_k} - \kappa_{k-1}}{\kappa_{k-1}} \times 100\%$$

< Vehicle dynamics>



< RL Agent-Environment Interaction >

$$s(\mathrm{k}) = [x(k), \mathrm{y}(\mathrm{k}), v_x(\mathrm{k}), v_y(\mathrm{k}), \tau(\mathrm{k}), \delta(\mathrm{k}), \psi(\mathrm{k}), \ \mathrm{y}^*(\mathrm{k}), \kappa_\mathrm{p}(\mathrm{k}), \mathrm{s}(\mathrm{k}-1)]^\mathrm{T}$$

$$\vec{a} = [\delta, \ \tau]^T$$

$$w_1 = 0.5, w_2 = 0.1, w_3 = 0.2, w_4 = 0.2$$

$$\mathrm{r}_{tot} = \mathrm{r}_{\mathrm{y}^*} + \mathrm{r}_\mathrm{s} + \mathrm{r}_{\mathrm{v_x}} + \mathrm{r}_{\kappa_\mathrm{t}} + \mathrm{r}_\mathrm{T}$$

$$\mathrm{B}(x) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}}$$

$$\mathrm{r}_{\mathrm{y}^*}(y^*) = w_1 \frac{10}{1 + \left|\frac{y^* + 1.7}{0.4}\right|^{2(7.5)}}$$

$$r_s(s_k) = w_2(s_k - s_{k-1})$$

$$\mathrm{r}_{\mathrm{v_x}}(v_x) = w_3 \frac{10}{1 + \left|\frac{v_x - 15}{4}\right|^{2(2)}}$$

$$r_{\kappa_\mathrm{k}} = \begin{cases} w_4(10) & \mathrm{e}_{\kappa_\mathrm{k}} < 15\% \\ w_4(-10) & else \end{cases}$$

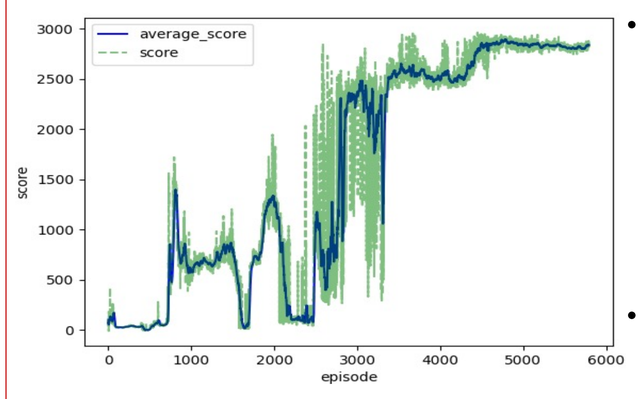
$$r_T = -3$$

$$y^* > 3m \ \ or \ \ y^* < 0.1m$$

$$\psi > 1.4rad$$

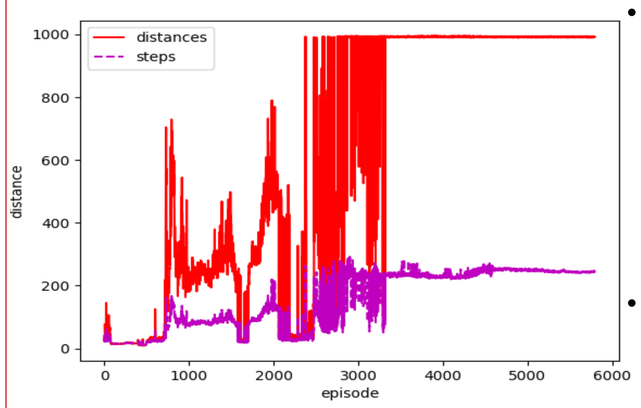
$$v < 4m/s$$

3. Results



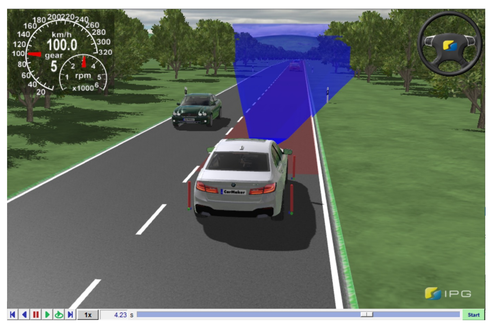
< Reward / Average Reward>

- Our investigation reveals a consistent increase in the Score and Step metrics per episode as learning progresses. This observation indicates the effectiveness of the policy learned through the generated reward function.
- With approximately 6000 episodes conducted, the model reliably completes the given road length of 1000m after a certain number of episodes.



< Distance / Time Step >

- After reaching the 1000m road length, the Score per episode exhibits a gradual increase to converge, signifying the completion of learning and stable lane tracking at the center.
- The trained model demonstrates the capability to follow the center of the lane reliably across diverse road configurations.



<IPG Carmaker driving simulation>

4. Expected Outcomes

- This research presents the implementation of a learning-based autonomous driving model in the commercial simulator environment, IPG Carmaker.
- Learning-based autonomous algorithms are scarcely commercialized due to challenges in acquiring driving data and the inherent risks involved, hindering performance assessment in real driving environments.
- Reinforcement learning empowers the driving system to make real-time decisions, monitor the surroundings, and choose optimal actions. Moreover, the system autonomously learns suitable responses to new situations.
- Thus, the reinforcement learning environment developed in this study holds the potential for versatile applications across diverse scenarios.

5. Future Work

- This study applied reinforcement learning to a straightforward road model. Future research aims to implement reinforcement learning models in more complex scenarios, incorporating constraints like other vehicles and traffic signals on intricate paths.
- While this study focused on controlling a single vehicle, future work involves leveraging ROS communication to train multiple vehicles simultaneously in diverse situations or exploring alternative control methods such as platoon driving.

상용 시뮬레이터 기반 차량 동역학을 고려한 강화학습 환경 구축

Development of Simulation Environment for Vehicle Dynamics-based Reinforcement Learning Using CarMaker Simulator

팀 명	TOCA
팀 원	김정효 (2018114346) 이효재 (2020118112) 정세빈 (2018111273)
지도 교수	한경석

1. 과제 선정 배경 및 필요성

- 최근 자율주행 자동차 산업이 급격히 성장함에 따라 관련된 주행 및 행동제어 알고리즘이 연구되고 있다.
- IPG CarMaker는 현실적인 주행환경 모사가 가능한 시뮬레이션 프로그램이다.
- TD3 강화학습 알고리즘은 차량의 속도, 조향각과 같은 연속적인 행동공간에서의 문제에 적용하기 적합하다.

2. 수행내용

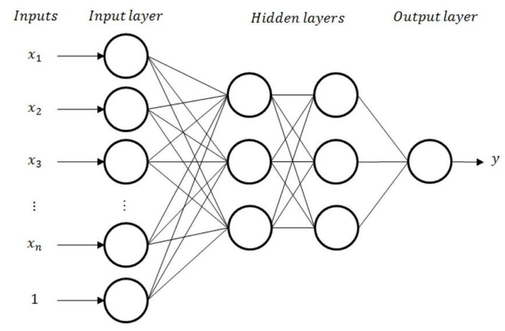
- **TD3 Algorithm**

Algorithm 1 TD3

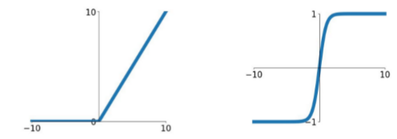
Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \min_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end if
end for

<TD3 Algorithm>



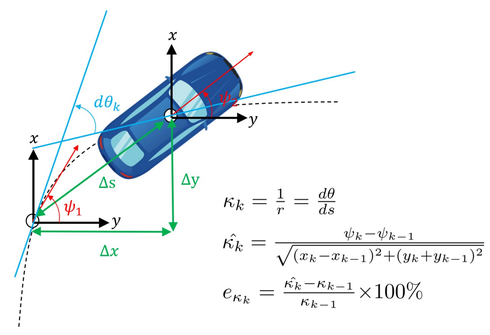
<MLP Structure>



<Activation Function – ReLu, tanh>

- Actor-Critic 계열의 TD3는 DDPG기반 구조를 가지고 있어 연속적인 행동공간에 적용할 수 있다.
- TD3는 2개의 독립적인 Critic신경망(MLP)으로 구성되며 각각의 신경망에서 계산된 두 개의 가치 추산치에서 더 작은 값을 Q-learning의 Target으로 이용함으로써 과도한 가치 추산 문제와 분산 문제를 해결하였다.

- **RL Environment**



< Vehicle dynamics>

$$s(\mathrm{k}) = [x(k), \mathrm{y}(\mathrm{k}), v_x(\mathrm{k}), v_y(\mathrm{k}), \tau(\mathrm{k}), \delta(\mathrm{k}), \psi(\mathrm{k}), \ \mathrm{y}^*(\mathrm{k}), \kappa_\mathrm{p}(\mathrm{k}), \mathrm{s}(\mathrm{k}-1)]^\mathrm{T}$$

$$\vec{a} = [\delta, \ \tau]^T$$

$$w_1 = 0.5, w_2 = 0.1, w_3 = 0.2, w_4 = 0.2$$

$$\mathrm{r}_{tot} = \mathrm{r}_{\mathrm{y}^*} + \mathrm{r}_\mathrm{s} + \mathrm{r}_{\mathrm{v_x}} + \mathrm{r}_{\kappa_\mathrm{t}} + \mathrm{r}_\mathrm{T}$$

$$\mathrm{B}(x) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}}$$

$$\mathrm{r}_{\mathrm{y}^*}(y^*) = w_1 \frac{10}{1 + \left|\frac{y^* + 1.7}{0.4}\right|^{2(7.5)}}$$

$$r_s(s_k) = w_2(s_k - s_{k-1})$$

$$\mathrm{r}_{\mathrm{v_x}}(v_x) = w_3 \frac{10}{1 + \left|\frac{v_x - 15}{4}\right|^{2(2)}}$$

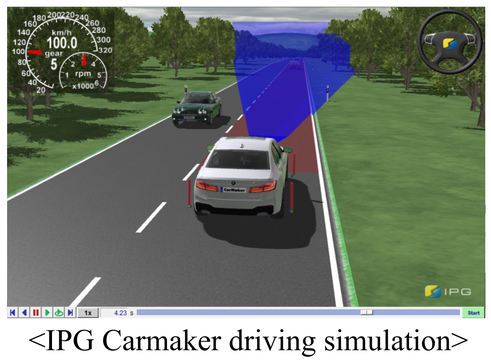
$$r_{\kappa_\mathrm{k}} = \begin{cases} w_4(10) & \mathrm{e}_{\kappa_\mathrm{k}} < 15\% \\ w_4(-10) & else \end{cases}$$

$$r_T = -3$$

$$y^* > 3m \ \ or \ \ y^* < 0.1m$$

$$\psi > 1.4rad$$

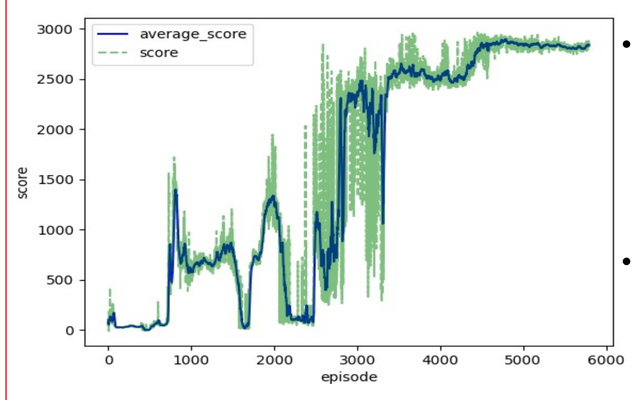
$$v < 4m/s$$



< RL Agent-Environment Interaction >

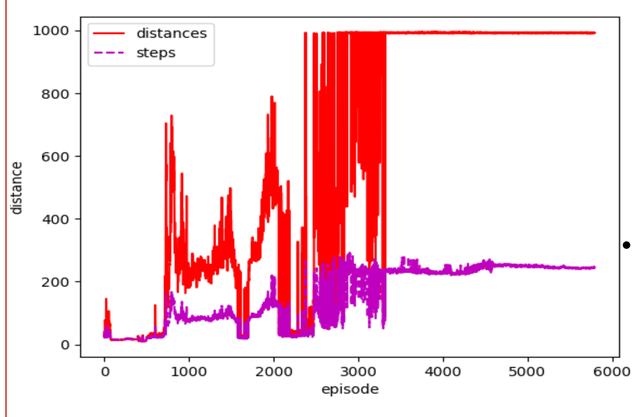
<IPG Carmaker driving simulation>

3. 최종 결과물



< Reward / Average Reward>

- 학습이 진행됨에 따라 에피소드당 Score, Step이 증가함을 확인할 수 있다. 이를 통해 만들어진 보상함수로 학습된 정책이 효과를 보인다는 것을 알 수 있다.
- 에피소드는 약 6000번 수행되었으며, 에피소드가 어느 정도 수행된 후 주어진 도로의 길이인 1000m까지 안정적으로 완주하는 것을 확인할 수 있다.



< Distance / Time Step >

- 도로 길이인 1000m까지 완주한 이후에도 에피소드당 Score는 점진적으로 증가하다가 수렴하게 되는데, 이를 통해 학습이 완료됨과 차선의 중앙을 안정적으로 추종함을 확인할 수 있다.
- 학습을 통해 만들어진 모델은 다양한 형태의 도로에서 차선 중앙을 추종하며 주행할 수 있을 것으로 판단된다.

4. 기대효과

- 본 연구에서는 상용 시뮬레이터인 IPG Carmaker 환경에서 학습기반의 주행모델을 구현하였다.
- 학습을 기반으로 한 자율주행 알고리즘은 상용화 된 것이 거의 없다. 주행 데이터를 얻기 힘들고 통제를 벗어날 경우 위험요소가 많아 실제 주행 환경에 적용하여 학습 성능을 측정하기에 어려움이 있기 때문이다.
- 강화학습은 주행 시스템이 실시간으로 의사결정을 내릴 수 있는 능력을 제공한다. 주행 시스템은 주변 환경을 모니터링하고, 현재 상황에 맞는 최적의 행동을 선택한다. 또한 새로운 상황에 대한 적절한 행동을 스스로 학습한다.
- 따라서 본 연구를 통해 구현된 강화학습 환경은 다양한 경우에 응용될 수 있을 것으로 기대된다.

5. 향후 계획

- 본 연구에서는 단순한 도로 모델에서 강화학습을 적용하였다. 향후에는 더욱 복잡한 경로에서 상대 차량, 신호등과 같은 여러 제약이 있는 조건으로 강화학습 모델을 구현하고자 한다.
- 본 연구에서는 하나의 차량만을 제어할 수 있었으나, 향후 ROS통신 등을 이용하여 하나의 제어기로 여러 대의 차량을 동시에 다양한 상황에서 학습시키거나, 군집주행 등의 다른 방식의 제어를 해보고자 한다.