LAB MEETING

국민대학교 지능형 차량 신호 처리 연구실 학부연구생 김지원

2024.03.06(목)

국민대학교
KOOKMIN UNIVERSITY

DDPG 알고리즘을 활용한
차선 유지 제어 강화학습

국민대학교
KOOKMIN UNIVERSITY

DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

DDPG Algorithm

Actor-Critic 알고리즘 기반

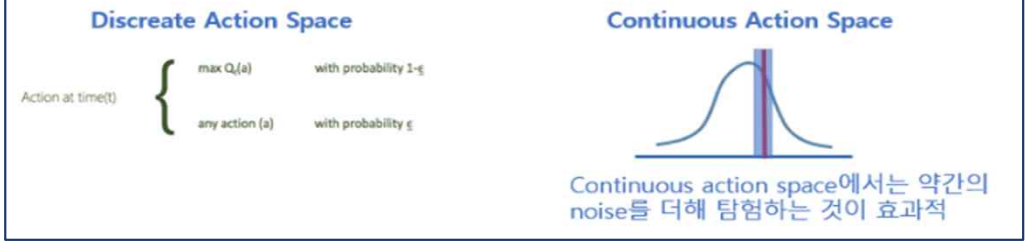
1. Value Function을 기반으로 Optimal Policy를 찾는 알고리즘 + Policy 자체를 강화하는 알고리즘
2. Policy based를 사용하는 Actor Network, Value based를 사용하는 Critic Network로 구성
3. 최소 2개 이상의 Network를 사용

Deterministic polices 기반

1. 확률적으로 행동을 취하는 것이 아닌, 주어진 상태에서 하나의 행동만을 선택하는 결정론적 정책(Deterministic Policy) 기반
2. Action Space를 Discrete에서 Continue로 확장
3. Input 개수 감소 및 효율적인 학습 가능

ε-greedy 방법 사용

1. 초기에 높은 Exploration(**ε**)을 진행하다가 점차 줄여나가는 방법
2. Exploration과 Exploitation을 적절한 비율로 조절 가능
3. Continuos Action Space를 사용한 DDPG에서는 Noise를 활용하여 **ε-greedy** 구현



DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

DDPG Algorithm m-file code 및 parameter

m-file code

```
stang = 90;

% action Info
numAct = 1;
actInfo = rlNumericSpec([numAct 1],"LowerLimit",-stang,"UpperLimit", stang);   %
actInfo.Name = 'SteeringAng';

% define environment
env = rlSimulinkEnv(mdl,agentblk,obsInfo,actInfo);
```

```
%% Critic Network

criticLayerSizes = [100 100];

statePath = [
    featureInputLayer(numObs,'Normalization','none','Name', 'observations')
    fullyConnectedLayer(criticLayerSizes(1), 'Name', 'CriticStateFC1', ...
            'Weights',2/sqrt(numObs)*(rand(criticLayerSizes(1),numObs)-0.5), ...
            'Bias',2/sqrt(numObs)*(rand(criticLayerSizes(1),1)-0.5))
    reluLayer('Name','CriticStateRelu1')
    fullyConnectedLayer(criticLayerSizes(2), 'Name', 'CriticStateFC2', ...
            'Weights',2/sqrt(criticLayerSizes(1))*(rand(criticLayerSizes(2)
            'Bias',2/sqrt(criticLayerSizes(1))*(rand(criticLayerSizes(2),1)
    ];

actionPath = [
    featureInputLayer(numAct,'Normalization','none', 'Name', 'action')
    fullyConnectedLayer(100, 'Name', 'CriticActionFC1', ...
            'Weights',2/sqrt(numAct)*(rand(100,numAct)-0.5), ...
            'Bias',2/sqrt(numAct)*(rand(100,1)-0.5))
    ];

commonPath = [
    additionLayer(2,'Name','add')
    reluLayer('Name','CriticCommonRelu1')
    fullyConnectedLayer(1, 'Name', 'CriticOutput',...
            'Weights',2*5e-3*(rand(1,criticLayerSizes(2))-0.5), ...
            'Bias',2*5e-3*(rand(1,1)-0.5))
    ];

criticNetwork = layerGraph(statePath);
criticNetwork = addLayers(criticNetwork, actionPath);
criticNetwork = addLayers(criticNetwork, commonPath);
criticNetwork = connectLayers(criticNetwork,'CriticStateFC2','add/in1');
criticNetwork = connectLayers(criticNetwork,'CriticActionFC1','add/in2');

% Create critic representation
criticOptions = rlRepresentationOptions('LearnRate',1e-04,'GradientThreshol
criticOptions.UseDevice = 'gpu';
%
critic = rlQValueRepresentation(criticNetwork,obsInfo,actInfo,...
    'Observation',{'observations'},'Action',{'action'},criticOptions);
```

```
%% Actor Network
actorLayerSizes = [100 100];
actorNetwork = [
    featureInputLayer(numObs,'Normalization','none','Name','observations')
    fullyConnectedLayer(actorLayerSizes(1), 'Name', 'ActorFC1', ...
            'Weights',2/sqrt(numObs)*(rand(actorLayerSizes(1),numObs)-0.5), ...
            'Bias',2/sqrt(numObs)*(rand(actorLayerSizes(1),1)-0.5))
    reluLayer('Name', 'ActorRelu1')
    fullyConnectedLayer(actorLayerSizes(2), 'Name', 'ActorFC2', ...
            'Weights',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),actorLayerSizes(1))-0.5), ...
            'Bias',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),1)-0.5))
    reluLayer('Name', 'ActorRelu2')
    fullyConnectedLayer(actorLayerSizes(2), 'Name', 'ActorFC3', ...
            'Weights',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),actorLayerSizes(1))-0.5), ...
            'Bias',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),1)-0.5))
    reluLayer('Name', 'ActorRelu3')
    fullyConnectedLayer(numAct, 'Name', 'ActorFC4', ...
            'Weights',2*5e-3*(rand(numAct,actorLayerSizes(2))-0.5), ...
            'Bias',2*5e-5*(rand(numAct,1)-0.5))
    tanhLayer('Name','ActorTanh1')
    scalingLayer("Name","Actorscaling","Scale",stang)
    ];

actorOptions = rlRepresentationOptions('LearnRate',5e-05,'GradientThreshold',1);

actorOptions.UseDevice = 'gpu';

actor = rlDeterministicActorRepresentation(actorNetwork,obsInfo,actInfo,'Observation',{'observations'},

% rlDDPGAgentOptions Options
agentOptions = rlDDPGAgentOptions(...
    'SampleTime',Ts,...
    'TargetSmoothFactor',1e-3,...
    'ExperienceBufferLength',1e7,...   % 수정 필요 *** 1e7
    'DiscountFactor',0.99,...
    'MiniBatchSize',128);    % 수정 필요 ***
agentOptions.NoiseOptions.Variance = 0.6;
agentOptions.NoiseOptions.VarianceDecayRate = 1e-6; % 1e-6;

agent = rlDDPGAgent(actor,critic,agentOptions);

% Train Agent
maxepisodes = 20000;   % 수정 필요 ***
maxsteps = 2000;       % 수정 필요 ***
```

parameter

Stang: 130 % 조향각 제한 값
numAct: 1 % 행동 변수 개수

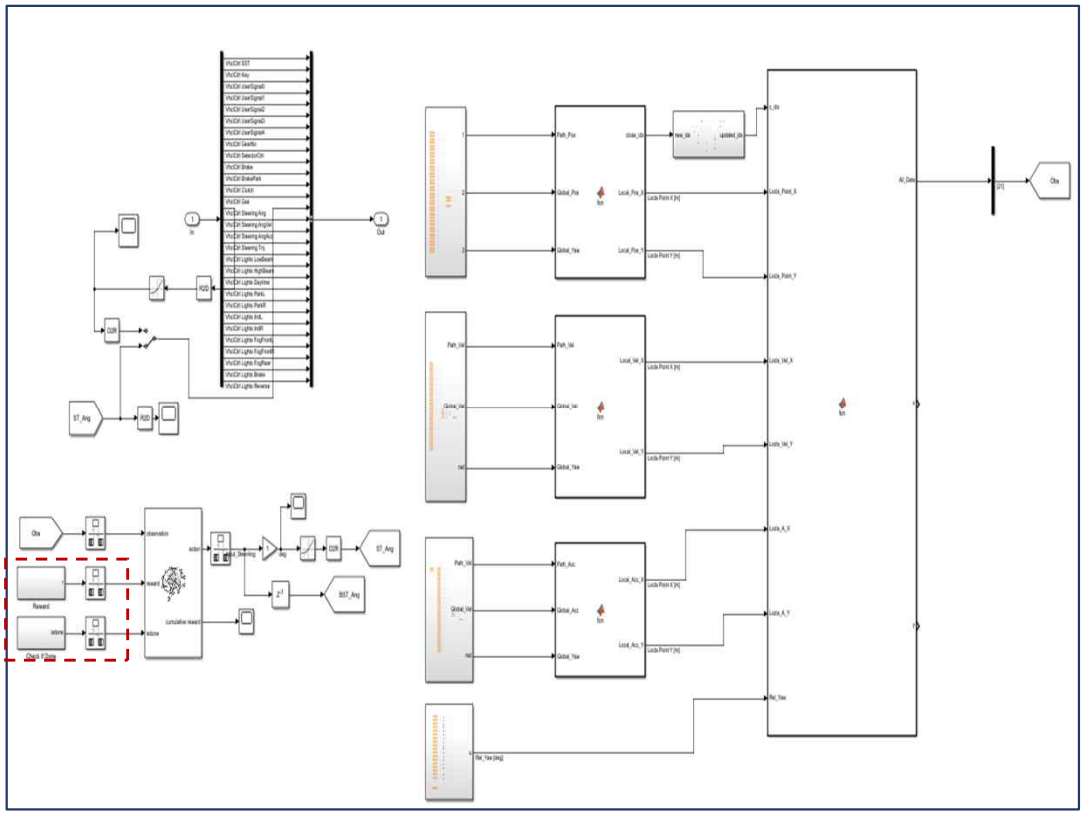
(actor)LearnRate: 1e-03
(critic)LearnRate: 1e-03

TargetSmoothFactor: 1e-3 % 목표 추정 스무딩 계수
ExperienceBufferLength: 1e7 % 경험 재생 버퍼 크기
DiscountFactor: 0.99 % 감가율
MiniBatchSize: 64 % 미니배치 크기
NoiseOptionsVariance: 0.6 % 행동탐색 노이즈 크기
NOVDecayRate: 1e-6 % 노이즈 감소율

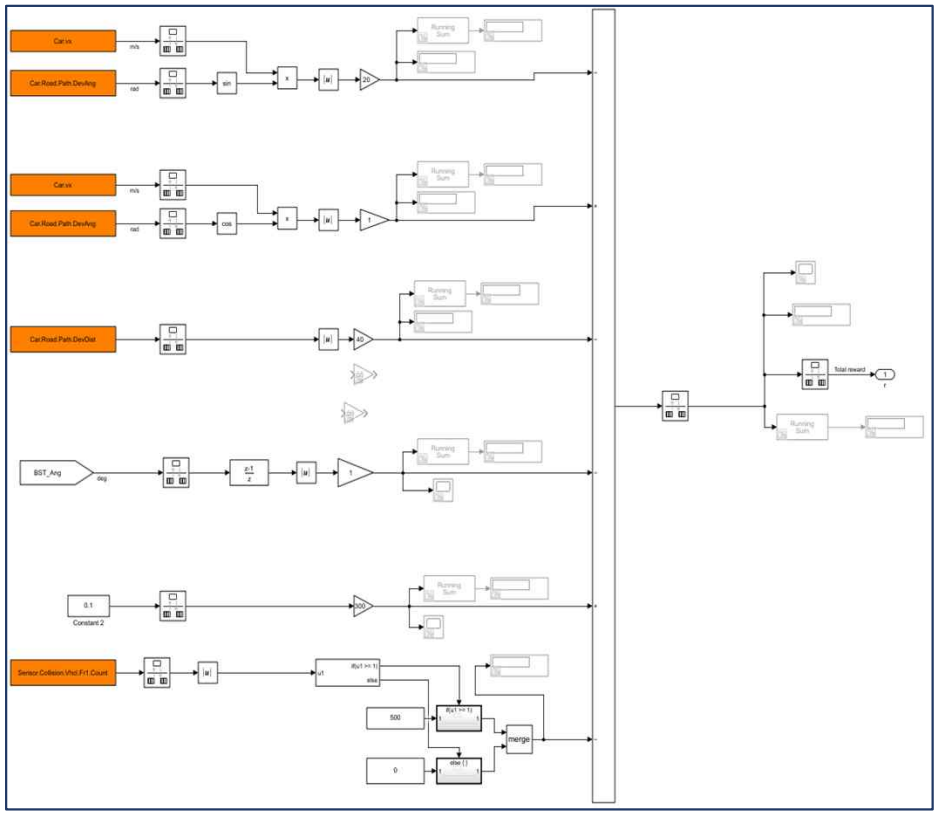
DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

DDPG Algorithm m-file Simulink model 및 Reward function

Simulink model

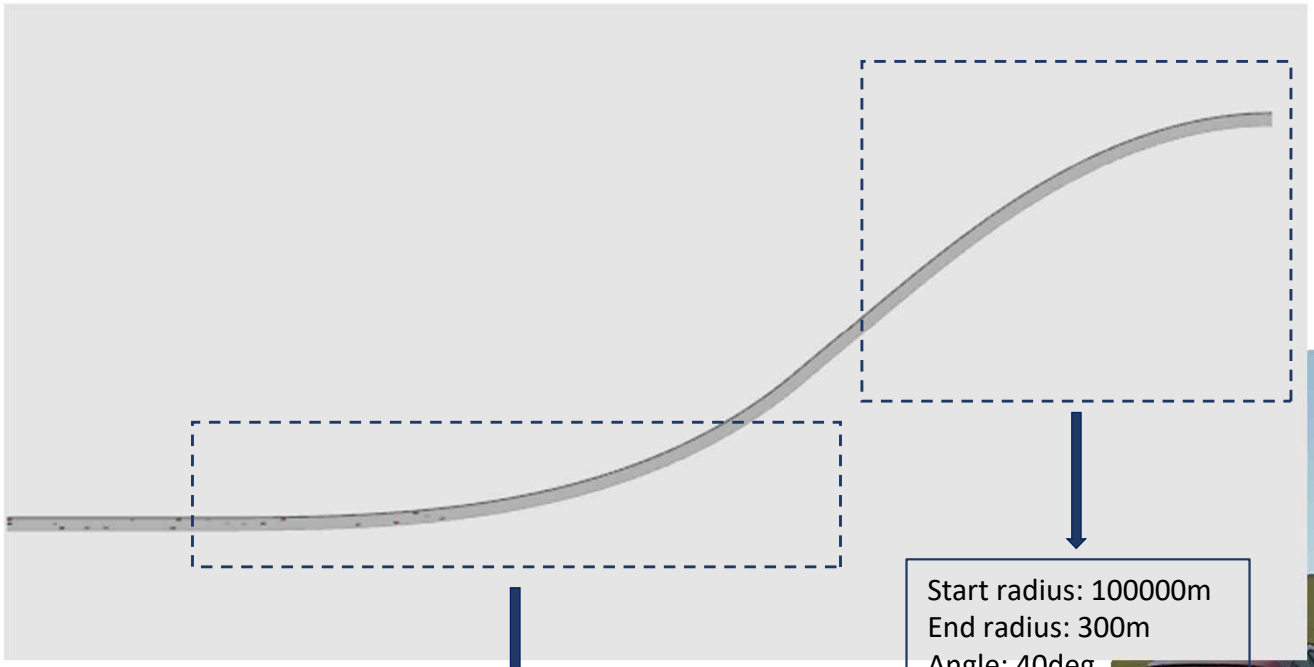


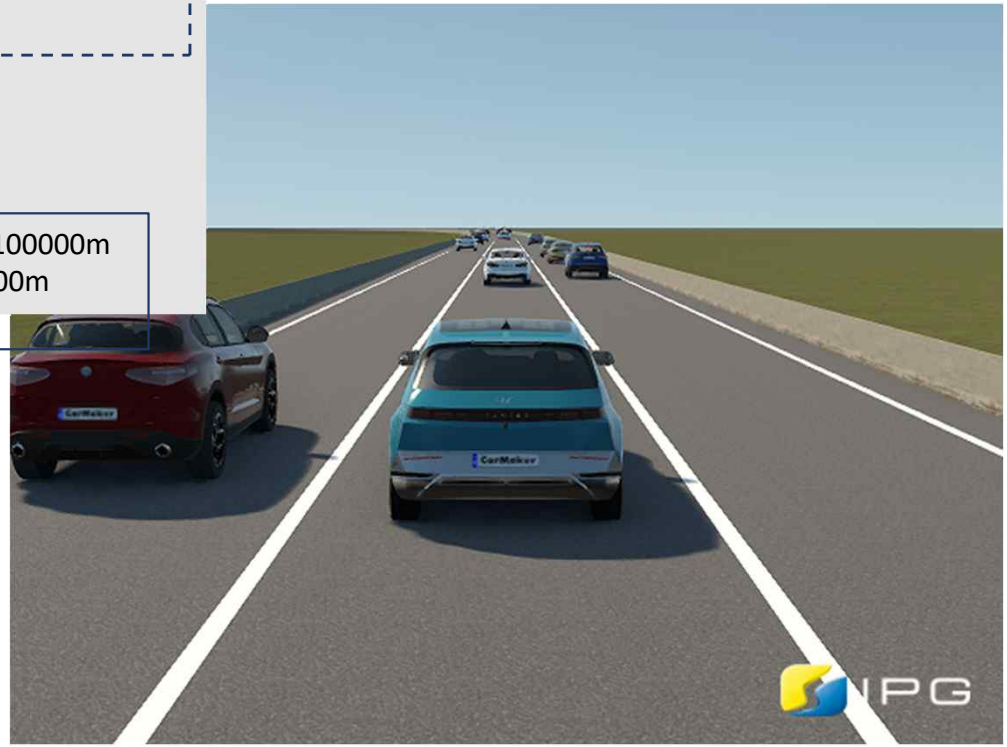
Reward function



DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

Scenario(1.2km)





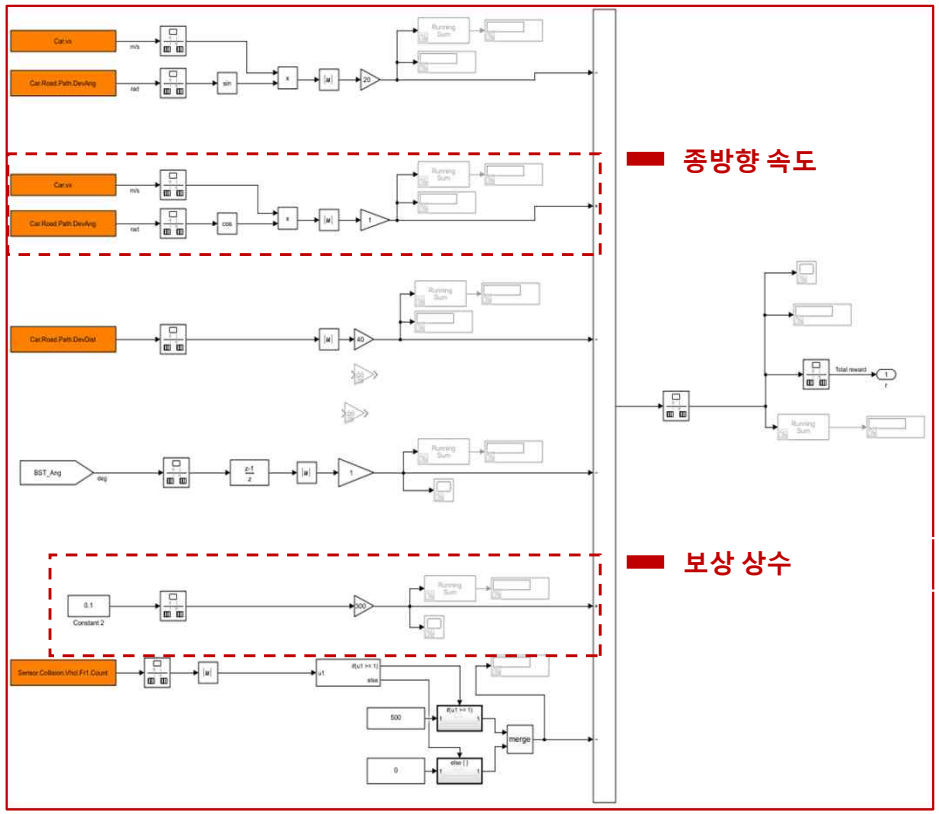
DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

변경전 강화학습 결과

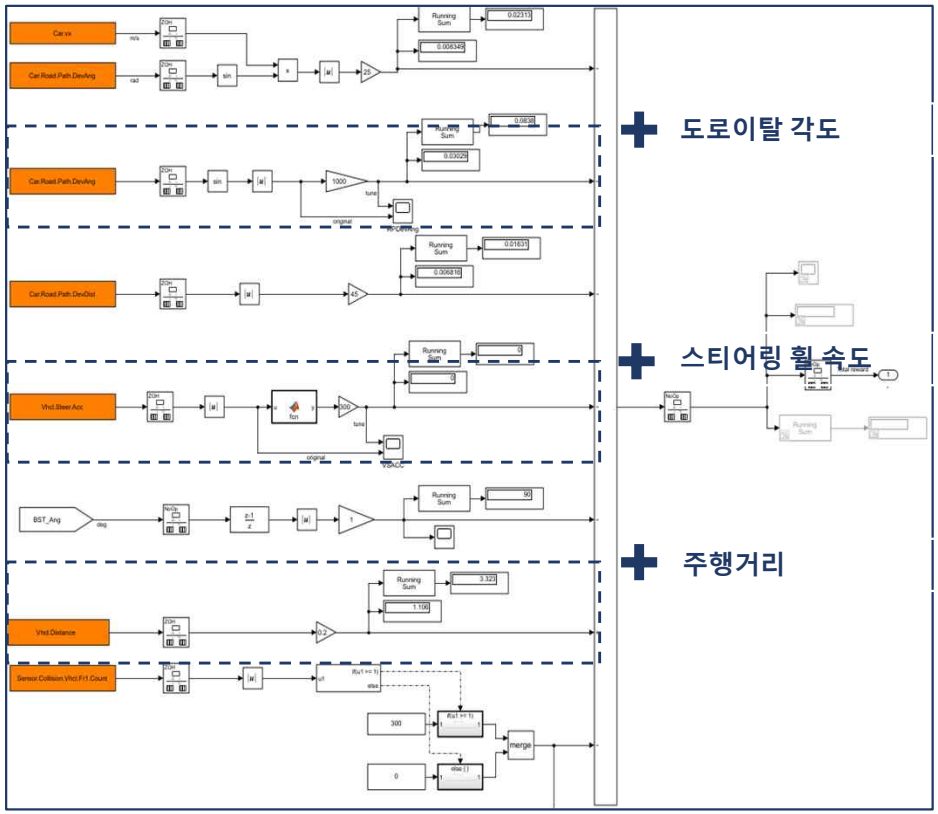
DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

Reward function 변경

변경 전 Reward function



변경 후 Reward function



DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

M-file code parameter 변경

변경 전 m-file code parameter

```
Stang: 130          % 조향각 제한 값
numAct: 1           % 행동 변수 개수
-------------------------------------------------------------------------------------------------
(actor)LearnRate: 1e-03
(critic)LearnRate: 1e-03
-------------------------------------------------------------------------------------------------
TargetSmoothFactor: 1e-3            % 목표 추정 스무딩 계수
ExperienceBufferLength: 1e7         % 경험 재생 버퍼 크기
DiscountFactor: 0.99                % 감가율
MiniBatchSize: 64                   % 미니배치 크기
NoiseOptionsVariance: 0.6           % 행동탐색 노이즈 크기
NOVDecayRate: 1e-6                  % 노이즈 감소율
```

변경 후 m-file code parameter

```
Stang: 130          % 조향각 제한 값
numAct: 1           % 행동 변수 개수
-------------------------------------------------------------------------------------------------
(actor)LearnRate: 5e-05
(critic)LearnRate: 1e-04
-------------------------------------------------------------------------------------------------
TargetSmoothFactor: 1e-3            % 목표 추정 스무딩 계수
ExperienceBufferLength: 1e7         % 경험 재생 버퍼 크기
DiscountFactor: 0.99                % 감가율
MiniBatchSize: 128                  % 미니배치 크기
NoiseOptionsVariance: 0.6           % 행동탐색 노이즈 크기
NOVDecayRate: 1e-6                  % 노이즈 감소율
```

DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

변경후 강화학습 결과

DDPG 알고리즘을 활용한 차선 유지 제어 강화학습

3월 개인연구 계획

강화학습

1. M-file code 심화 분석(Actor-Critic Network, 연결 구조, 활성화 함수 등)
2. Simulink model 심화 분석(보상 함수 외에 RL Agent 블록, UAQ 블록 등)
3. 강화학습 전체 이론 복습 및 TD3, SAC 이론 학습
4. TD3, SAC 알고리즘 m-file code 및 Simulink model 활용실습

감사합니다.

국민대학교 지능형 차량 신호 처리 연구실 학부연구생 김지원

2024.03.06(목)

국민대학교
KOOKMIN UNIVERSITY