UR5e 로봇의 모방학습 시뮬레이션 API

과제명	인간을 물리적 상호작용을 통해 지원하는 로봇을 위한 가상현실 환경 인간시연을 모방하는 학습 기술 개발
작성자	써로마인드 이정민
작성 날짜	2022-05-20
문서 리뷰 날짜	2022-05-27
승인자	써로마인드 김병희
승인 날짜	2022-05-28

API 요약

API 종류	API 명	설명
neem data load	neem.load_neem	neem 데이터 불러오기
	neem.select_sc	neem 시나리오 선택
	neem.select_object	시나리오에서 원하는 데이터 추출
	neem.time_step_object	시나리오상 원하는 time step 경로 데이터 추출
	neem.data_augmentation	경로 데이터 증강
Ur5 Test	urx_move. test_pose	ur5 임의위치 이동 Test
	urx_move. test_Grasping	ur5 잡기 Test
	urx_move.move_dummy	ur5 더미 잡기 Test
UR5 reaching	reaching.reaching	ur5 특정 object로의 ur5 조인트
	reaching.obstacle	ur5 장애물을 피하면서 오브젝트에 도달
UR5 placing	placing.obstacle	ur5 장애물을 피하며 물체를 옮김
	placing.drop	ur5 물체를 옮긴 뒤 특정위치 drop
RL_base	RL_base.get_state	ur5 전체 정보 가져오기
	RL_base.get_reward	강화학습 보상 가져오기
	RL_base.rl_step	시뮬레이션 1 스텝 실행
	RL_base.rl_reset	시뮬레이션 초기화

Neem data load

neem.load_neem

neem.load_neem(path)

- Input
  - path : Universität Bremen 의 OpenEASE에서 받은 Neem Data 최상위 경로
- Return
  - Neem_Data : Universität Bremen 의 OpenEASE에서 받은 모든 Neem Data를 딕셔너리 형태로 반환

neem.select_sc

neem.select_sc(scenario, ep)

- Input
  - scenario: 'sc1' mount에 더미물체 걸기 or
    'sc2' 더미물체 mount에서 빼기 선택
  - ep: '0' ~ '9' 각 시나리오 당 각기 다른 10개의 episode중 원하는 episode 고르기
- Return
  - neem_episode : 선택한 scenario 및 episode에 맞는 neem data 반환

neem.select_object

neem.select_object(neem_episode, object)

- Input
  - neem_episode :선택한 neem data
  - object : neem data 중 원하는 물체
- Return
  - object_trajectory: 선택한 object의 trajectory data

neem.time_step_object

neem.time_step_object(object_trajectory, time_step)

- Input
  - object_trajectory: 개별 object에 대한 경로
  - time_step: 시작 time_step과 끝 time_step
- Return
  - object_time_step: 선택한 time_step의 경로 반환

neem.data_augmentation

neem.data_augmentation(trajectory, aug, start_shift, end_shift)

- Input
  - trajectory: 경로 데이터
  - aug : 노이즈 추가 강도 '0~10'
  - start_shift : 기존 시작 위치에서의 x,y,z 변경 값

    - end_shift : 기존 경로에서 끝나는 위치의 x y z 변경 값
- Return
    - aug_trajectory: 변경된 경로

UR5 Test

urx_move.test_pose

urx_move.test_pose(num_trials = 3)

- Input
  - num_trials : 시도 횟수
- output
  - None
- Action
  - num_trials 회수 만큼 로봇 end point 이동

urx_move.test_Grasping

urx_move.test_Grasping(num_trials=3, select_object = 0):

- Input
  - num_trials : 시도 회수
  - select_object : pybullet에서 지정한 object 번호
- output
  - None
- Action
  - num_trials 회수 만큼 select_object에서 지정한 object를 ur5 로봇이 잡기 들어올리기 놓기를 반복

urx_move.move_dummy

urx_move.move_dummy(num_trials=3)

- Input
  - num_trials : 시도 회수

- output
  - None

- Action
  - num_trials만큼 더미 모델의 위치를 랜덤하게 이동

UR5 reaching

- 좌표 (-1.52, 1, -0.564)에 위치한 UR5e 로봇을 이용하여 특정 오브젝트에 도달하는 태스크입니다.
- 입력되는 좌표들은 모두 UR5e 좌표로부터 [0.4, 0.9] 범위 내에 위치해야 합니다.

reaching.reaching

reaching.reaching(X,Z)

- Input
  - X: x좌표
  - Y: y좌표

- output
  - action: UR5e 로봇의 6 개의 joint 값을 컨트롤 할 6 차원 list

reaching.obstacle

reaching.obstacle(X_1, Z_1, X_2, Z_2) -> 6차원 list

- Input
  - X_1, Z_1 : Target object의 좌표
  - X_2, Z_2 : 장애물의 좌표

- output
  - action list : 장애물을 피해 오브젝트에 도달하기 위한 action list

UR5 placing

- 입력되는 좌표들은 모두 UR5e 좌표로부터 [0.4, 0.9] 범위 내에 위치해야 합니다.

placing.drop

placing.drop(Oh, Ow, Oz, X, Z, R)

- Input
  - Oh * Ow * Oz: 물체를 좌표
  - X, Z 지점에 있는 반경 R 떨어뜨릴 영역
- Return
  - action: 물체를 일정 영역에 떨어뜨릴 UR5e 로봇의 6개의 joint 값

placing. obstacle

placing.obstacle(Oh, Ow, Oz, X_1, Z_1, R_1, X_2, Z_2, R_2)

- Input
  - Oh * Ow * Oz: 물체
  - X_1, Z_1 : 좌표
  - R_1 : 반경
  - X_2, Z_2 : 장애물의 좌표
  - R_2 : bounding sphere의 반경
- Return
  - joint action list : 크기 Oh * Ow * Oz인 물체를 좌표 X1, Z1 지점에 있는 반경 R1 영역으로 옮기면서(suction 사용 가정), 장애물의 좌표 X2, Z2와

bounding sphere의 반경 R2의 장애물을 피해 오브젝트에 도달하기 위한 UR5e 조인트 리스트

UR5 RL_base

RL_base.get_state

RL_base. get_state()

- Input
  - None
- Return
  - info_dict : ur5, 더미, mount tip 관련 정보를 딕셔너리로 반환

```
        # ur5
        info_dict['ur5_current_joint_state'] = current_joint_state
        info_dict['ur5_tool_tip_pose']       = tool_tip_pose
        # dummy
        info_dict['dummy_hole']              = dummy_hole
        info_dict['dummy_center']            = dummy_center
        # mount
        info_dict['mount_tool_tip']          = mount_tool_tip
        info_dict['mount_center']            = mount_center
```

RL_base.get_reward

RL_base.get_reward()

- Input
  - None
- Return
  - goal: 강화학습이 목표에 도달했는지 성공 또는 실패를 반환

RL_base.rl_step

RL_base. get_state(ur5_joint)

- Input
  - ur5_joint : 움직일 ur5 joint list
- Return
  - None
- Action
  - 시뮬레이션이 1 step 시간이 진행되면서 ur5가 목표 위치로 이동

RL_base.rl_reset

RL_base. get_state(dummy_pose = None, dummy_quat = None, mount_pose = None)

- Input
  - dummy_pose : 초기화 할 때의 dummy 위치
  - dummy_quat : 초기화 할 때의 dummy 방향
  - mount_pose : 초기화 할때의 mount 위치
- Return
  - None
- Action
  - 처음상태로 초기화