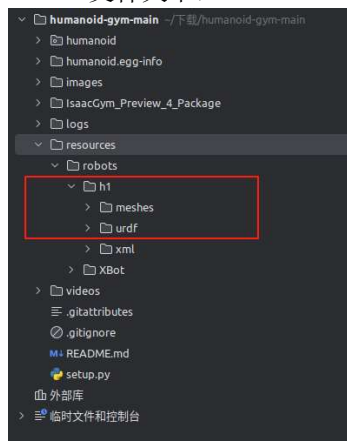


## 一. 导入机器人模型

(1) 下载某个人形机器人模型，包括urdf以及mesh文件等。

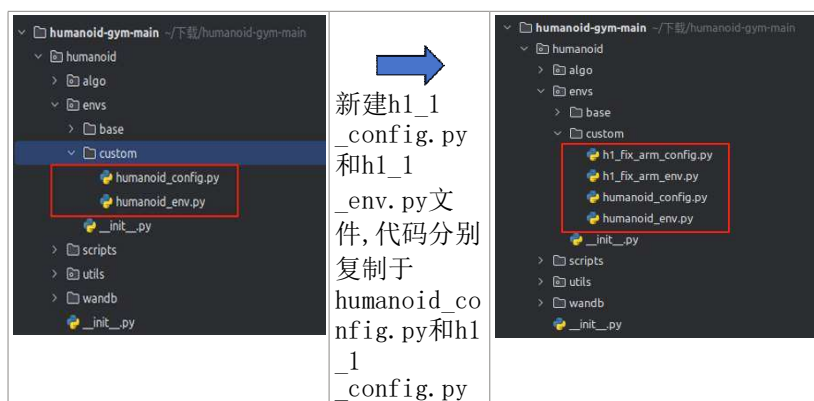
常用机器人模型下载路径：<https://github.com/robot-descriptions/awesome-robot-descriptions>

(2) 机器人模型加载（将下载好的机器人模型放置在robots文件夹下）



## 二. 注册任务

(1) 新建\*config.py(配置文件)和\*env.py(任务文件)，如h1\_config.py和h1\_env.py。将humanoid\_config.py代码复制到h1\_config.py文件中，humanoid\_env.py代码复制到h1\_env.py文件中。



(2) 修改h1\_env.py与h1\_config.py中的类名

将h1\_fix\_arm\_config中的class

XBotLCfg(LeggedRobotCfg): **修改为** class H1

\_fix\_arm\_Cfg(LeggedRobotCfg):

```
h1_fix_arm_config.py x
No Python interpreter configured for the project

25 # OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
26 # OF THIS SOFTWARE, EVEN IF ADVISED OF THE POS
27 #
28 # Copyright (c) 2024 Beijing RobotEra TECHNOLO
29
30
31 from humanoid.envs.base.legged_robot_config import
32
33
34 class H1_fix_arm_Cfg(LeggedRobotCfg):
35
```

将h1\_fix\_arm\_config中的class  
XBotLCfgPP0(LeggedRobotCfgPP0):修改为class H1  
\_fix\_arm\_CfgPP0(LeggedRobotCfgPP0):

```
h1_fix_arm_config.py x
No Python interpreter configured for the project

34 class H1_fix_arm_Cfg(LeggedRobotCfg):
231     class normalization:
232         class obs_scales:
234             ang_vel = 1.
235             dof_pos = 1.
236             dof_vel = 0.05
237             quat = 1.
238             height_measurements = 5.0
239             clip_observations = 18.
240             clip_actions = 18.
241
242
243     class H1_fix_arm_CfgPP0(LeggedRobotCfgPP0):
244         seed = 5
245         runner_class_name = 'OnPolicyRunner' # DWL0nPolicyRunner
246
```

将h1\_fix\_arm\_env.py中的class  
XBotLFreeEnv(LeggedRobot):修改为class H1  
\_fix\_arm\_FreeEnv(LeggedRobot):

```
h1_fix_arm_config.py h1_fix_arm_env.py x
No Python interpreter configured for the project

37 from humanoid.envs import LeggedRobot
38
39 from humanoid.utils.terrain import HumanoidTerrain
40
41
42 class H1_fix_arm_FreeEnv(LeggedRobot):
43
```

### 三. 修改h1\_config.py文件

(1) 查看机器人的关节自由度数量，并修改h1\_1  
\_config.py文件中的num\_single\_obs参数。如h1机器人上  
肢锁定后的自由度数量为10。



```
h1_fix_arm.urdf
此文件已显式重新分配为纯文本
1/10
91     <geometry>
92         <cylinder length="0.07" radius="0.06" />
93     </geometry>
94 </collision>
95 </link>
96 <joint
97     name="left_hip_yaw_joint"
98     type="revolute">
```

查看机器人urdf模型中的revolute数量(自由度数量)，对应得到privileged\_obs\_buf的数量

```
h1_fix_arm.config.py
No Python interpreter configured for the project
42 class H1_fix_arm_FreeNode(LeggedRobot):
287 def compute_observations(self):
288     diff = self.dof_pos - self.ref_dof_pos
289     self.privileged_obs_buf = torch.cat((
290         self.command_input, # 2 + 3
291         (self.dof_pos - self.default_joint_pd_target) * \
292         self.obs_scales.dof_pos, # 10
293         self.dof_vel * self.obs_scales.dof_vel, # 10
294         self.actions, # 10
295         diff, # 10
296         self.base_lin_vel * self.obs_scales.lin_vel, # 3
297         self.base_ang_vel * self.obs_scales.ang_vel, # 3
298         self.base_euler_xyz * self.obs_scales.quat, # 3
299         self.rand_push_force[:, :2], # 2
300         self.rand_push_torque, # 3
301         self.env_frictions, # 1
302         self.body_mass / 30., # 1
303         stance_mask, # 2
304         contact_mask, # 2
305     ), dim=-1)
```

```
h1_fix_arm.urdf
此文件已显式重新分配为纯文本
1/10
91     <geometry>
92         <cylinder length="0.07" radius="0.06" />
93     </geometry>
94 </collision>
95 </link>
96 <joint
97     name="left_hip_yaw_joint"
98     type="revolute">
```

查看机器人urdf模型中的revolute数量(自由度数量)，对应得到obs\_buf的数量

```
h1_fix_arm.config.py
No Python interpreter configured for the project
42 class H1_fix_arm_FreeNode(LeggedRobot):
287 def compute_observations(self):
288     self.body_mass / 30., # 1
289     stance_mask, # 2
290     contact_mask, # 2
291     ), dim=-1)
292
293 obs_buf = torch.cat((
294     self.command_input, # 5 = 20(sin cos) + 30(vel_x
295     q, # 100
296     dq, # 100
297     self.actions, # 100
298     self.base_ang_vel * self.obs_scales.ang_vel, # 3
299     self.base_euler_xyz * self.obs_scales.quat, # 3
300     ), dim=-1)
```

```
h1_fix_arm.config.py
No Python interpreter configured for the project
42 class H1_fix_arm_FreeNode(LeggedRobot):
287 def compute_observations(self):
288     self.body_mass / 30., # 1
289     stance_mask, # 2
290     contact_mask, # 2
291     ), dim=-1)
292
293 obs_buf = torch.cat((
294     self.command_input, # 5 = 20(sin cos) + 30(vel_x
295     q, # 100
296     dq, # 100
297     self.actions, # 100
298     self.base_ang_vel * self.obs_scales.ang_vel, # 3
299     self.base_euler_xyz * self.obs_scales.quat, # 3
300     ), dim=-1)
```

(1) 计算obs\_buf的总数，得到num\_single\_obs。  
(2) 根据机器人自由度数量，修改num\_actions的值

```
h1_fix_arm.config.py
No Python interpreter configured for the project
33
34 2个用法
35 class H1_fix_arm_Cfg(LeggedRobotCfg):
36     """
37     Configuration class for the XBot1 humanoid robot.
38
39     class env(LeggedRobotCfg.env):
40         # change the observation dim
41         frame_stack = 15
42         c_frame_stack = 3
43         num_single_obs = 41 #47
44         num_observations = int(frame_stack * num_single_obs)
45         single_num_privileged_obs = 65 #73
46         num_privileged_obs = int(c_frame_stack * single_num_privileged_obs)
47         num_actions = 10 #12
48         num_envs = 4096
49         episode_length_s = 24 # episode length in seconds
50         use_ref_actions = False # speed up training by using reference actions
```

(2) 修改机器人的h1\_fix\_arm\_config.py中的asset类  
对应修改urdf的路径机器人  
name、foot\_name、knee\_name (foot\_name,knee\_name  
可由urdf文件得知)，以及设置碰撞后重置和惩罚的刚体。

```
h1_fix_arm_config.py  h1_fix_arm_env.py
No Python interpreter configured for the project

34 class H1_fix_arm_Cfg(LeggedRobotCfg):
51     class safety:
55         torque_limit = 0.85
56
57     class asset(LeggedRobotCfg.asset):
58         file = '{LEGGED_GYM_ROOT_DIR}/resources/robots/h1/urdf/h1_fix_arm.urdf'
59
60         name = "h1"
61         foot_name = "ankle"
62         knee_name = "knee"
63
64         terminate_after_contacts_on = ['pelvis', 'elbow']
65         penalize_contacts_on = ["pelvis"]
66         self_collisions = 0 # 1 to disable, 0 to enable...bitwise filter
67         flip_visual_attachments = False
68         replace_cylinder_with_capsule = False
69         fix_base_link = False
70
```

(3) 修改机器人的h1\_fix\_arm\_config.py中的init\_state类。对应修改机器人质心的初始位置pos，初始关节角度default\_joint\_angles。

```
h1_fix_arm_config.py
No Python interpreter configured for the project

34 class H1_fix_arm_Cfg(LeggedRobotCfg):
100     class init_state(LeggedRobotCfg.init_state):
101         pos = [0.0, 0.0, 1.01]
102
103         default_joint_angles = { # = target angles [rad] when action = 0.0
104             'left_hip_yaw_joint': 0.0,
105             'left_hip_roll_joint': 0.,
106             'left_hip_pitch_joint': -0.349,
107             'left_knee_joint': 0.698,
108             'left_ankle_joint': -0.349,
109
110             'right_hip_yaw_joint': -0.0,
111             'right_hip_roll_joint': -0.,
112             'right_hip_pitch_joint': -0.349,
113             'right_knee_joint': 0.698,
114             'right_ankle_joint': -0.349,
115         }
```

(4) 修改机器人的h1\_fix\_arm\_config.py中的control类。对应修改机器人的刚度(stiffness)和阻尼(damping)

```
h1_fix_arm_config.py h1_fix_arm_env.py
No Python interpreter configured for the project

34 class H1_fix_arm_Cfg(LeggedRobotCfg):
100 class init_state(LeggedRobotCfg.init_state):
115 }
116
117 class control(LeggedRobotCfg.control):
118     # PD Drive parameters:
119     stiffness = {'hip_yaw': 200,
120                 'hip_roll': 200,
121                 'hip_pitch': 200,
122                 'knee': 300,
123                 'ankle': 40,
124                 'torso': 300,
125                 'shoulder': 100,
126                 "elbow":100,
127                 } # [N*m/rad]
128     damping = { 'hip_yaw': 5,
129                 'hip_roll': 5,
130                 'hip_pitch': 5,
131                 'knee': 6,
132                 'ankle': 2,
133                 'torso': 6,
134                 'shoulder': 2,
135                 "elbow":2,
136                 } # [N*m/rad] # [N*m*s/rad]
```

(5) 修改机器人的h1\_fix\_arm\_config.py中的rewards类。例如：修改max\_contact\_force改为450，修改min\_dist和max\_dist(两脚和两膝的距离)，base\_height\_target高度等。

```
h1_fix_arm_config.py
No Python interpreter configured for the project

34 class H1_fix_arm_Cfg(LeggedRobotCfg):
177 class commands(LeggedRobotCfg.commands):
183 class ranges:
184     lin_vel_x = [-0.3, 0.6] # min max [m/s]
185     lin_vel_y = [-0.3, 0.3] # min max [m/s]
186     ang_vel_yaw = [-0.3, 0.3] # min max [rad/s]
187     heading = [-3.14, 3.14]
188
189 class rewards:
190     base_height_target = 0.98 #0.89
191     min_dist = 0.35 #0.2
192     max_dist = 0.6 #0.5
193     # put some settings here for LLM parameter tuning
194     target_joint_pos_scale = 0.2 #0.17 # rad
195     target_feet_height = 0.06 # m
196     cycle_time = 0.64 # sec
197     # if true negative total rewards are clipped at zero (avoids early termination problems)
198     only_positive_rewards = True
199     # tracking reward = exp(error*sigma)
200     tracking_sigma = 5
201     max_contact_force = 1000 #700 # Forces above this value are penalized
```

(6) 修改机器人的h1\_fix\_arm\_config.py中的runner类，将experiment\_name改为H1\_ppo。

```
h1_fix_arm_config.py h1_fix_arm_env.py
No Python interpreter configured for the project

243 class H1_fix_arm_CfgPPO(LeggedRobotCfgPPO):
252 class algorithm(LeggedRobotCfgPPO.algorithm):
253     num_learning_epochs = 2
254     gamma = 0.994
255     lam = 0.9
256     num_mini_batches = 4
257
258 class runner:
259     policy_class_name = 'ActorCritic'
260     algorithm_class_name = 'PPO'
261     num_steps_per_env = 60 # per iteration
262     max_iterations = 3001 # number of policy updates
263
264     # logging
265     save_interval = 100 # Please check for potential savings every 'save_interval' iterations.
266     experiment_name = 'H1_ppo'
267     run_name = ''
268     # Load and resume
269     resume = False
270     load_run = -1 # -1 = last run
271     checkpoint = -1 # -1 = last saved model
272     resume_path = None # updated from load_run and chkpt
```



## 四. 修改H1\_fix\_arm\_FreeEnv.py文件

### (1) 得到参考关节轨迹

```
42 class H1_fix_arm_FreeEnv(LeggedRobot):
    1个用法
121     def compute_ref_state(self):
122         phase = self._get_phase()
123         # print("phase", phase)
124         sin_pos = torch.sin(2 * torch.pi * phase)
125         sin_pos_l = sin_pos.clone()
126         sin_pos_r = sin_pos.clone()
127         self.ref_dof_pos = torch.zeros_like(self.dof_pos)
128         scale_1 = self.cfg.rewards.target_joint_pos_scale
129         scale_2 = 2 * scale_1
130         # left swing
131         sin_pos_l[sin_pos_l > 0] = 0
132         sin_pos_l[torch.abs(sin_pos) < 0.1] = 0
133         #根据腿的比例调参, 以及默认角度时, 左右腿的角度若相同, 则需要加负号 (左右腿相反)
134         #H1与小星在膝关节以及腕关节的角度的方向不同
135         self.ref_dof_pos[:, 2] = sin_pos_l * scale_1 + self.cfg.init_state.default_joint_angles["left_hip_pitch_joint"]
136         #print("self.ref_dof_pos[:, 2]", self.ref_dof_pos[:, 2])
137         self.ref_dof_pos[:, 3] = -sin_pos_l * scale_2 + self.cfg.init_state.default_joint_angles["left_knee_joint"]
138         self.ref_dof_pos[:, 4] = sin_pos_l * scale_1 + self.cfg.init_state.default_joint_angles["left_ankle_joint"]
139         # right
140         sin_pos_r[sin_pos_r < 0] = 0
141         sin_pos_r[torch.abs(sin_pos) < 0.1] = 0
142         self.ref_dof_pos[:, 7] = -sin_pos_r * scale_1 + self.cfg.init_state.default_joint_angles["right_hip_pitch_joint"]
143         #print("self.ref_dof_pos[:, 8]", self.ref_dof_pos[:, 8])
144         self.ref_dof_pos[:, 8] = sin_pos_r * scale_2 + self.cfg.init_state.default_joint_angles["right_knee_joint"]
145         self.ref_dof_pos[:, 9] = -sin_pos_r * scale_1 + self.cfg.init_state.default_joint_angles["right_ankle_joint"]
```

### (2) 根据状态空间修改对齐噪声维度

```
42 class H1_fix_arm_FreeEnv(LeggedRobot):
207     def compute_observations(self):
239         self.body_mass / 30., # 1
240         stance_mask, # 2
241         contact_mask, # 2
242     ), dim=-1)
243
244     obs_buf = torch.cat((
245         self.command_input, # 5 = 20(sin cos) + 3D(vel x
246         q, # 100
247         dq, # 100
248         self.actions, # 100
249         self.base_ang_vel * self.obs_scales.ang_vel, # 3
250         self.base_euler_xyz * self.obs_scales.quat, # 3
251     ), dim=-1)
```

→

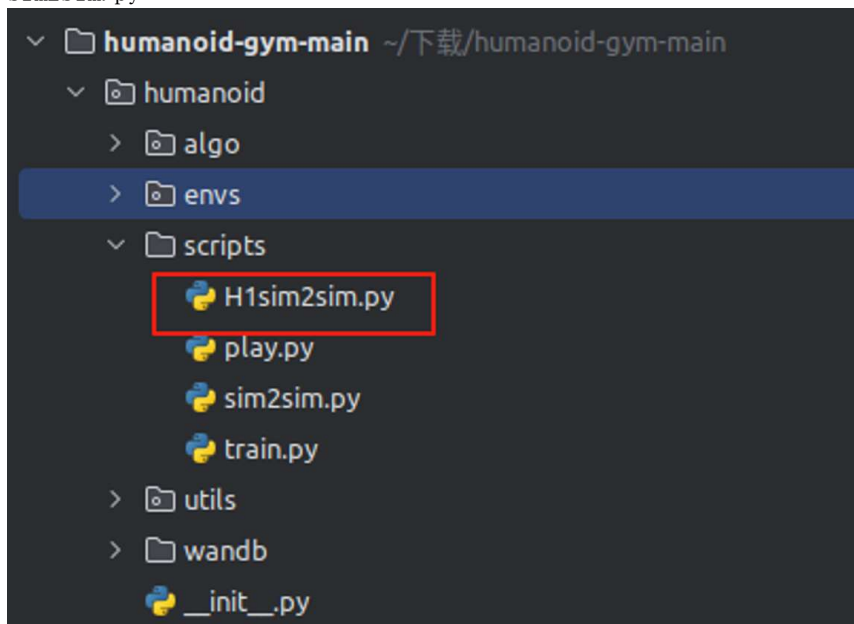
```
42 class H1_fix_arm_FreeEnv(LeggedRobot):
173     def _get_noise_scale_vec(self, cfg):
174
175         Args:
176             cfg (Dict): Environment config file
177
178         Returns:
179             [torch.Tensor]: Vector of scales used to multiply a uniform distribution in
180
181         """
182         noise_vec = torch.zeros(
183             self.cfg.env.num_single_obs, device=self.device)
184         self.add_noise = self.cfg.noise.add_noise
185         noise_scales = self.cfg.noise.noise_scales
186         noise_vec[0: 5] = 0. # commands
187         noise_vec[5: 15] = noise_scales.dof_pos * self.obs_scales.dof_pos
188         noise_vec[15: 25] = noise_scales.dof_vel * self.obs_scales.dof_vel
189         noise_vec[25: 35] = 0. # previous actions
190         noise_vec[35: 38] = noise_scales.ang_vel * self.obs_scales.ang_vel # ang vel
191         noise_vec[38: 41] = noise_scales.quat * self.obs_scales.quat # euler x
192
193         return noise_vec
```

### (3) 设置self.default\_dof\_pos为机器人初始关节角度

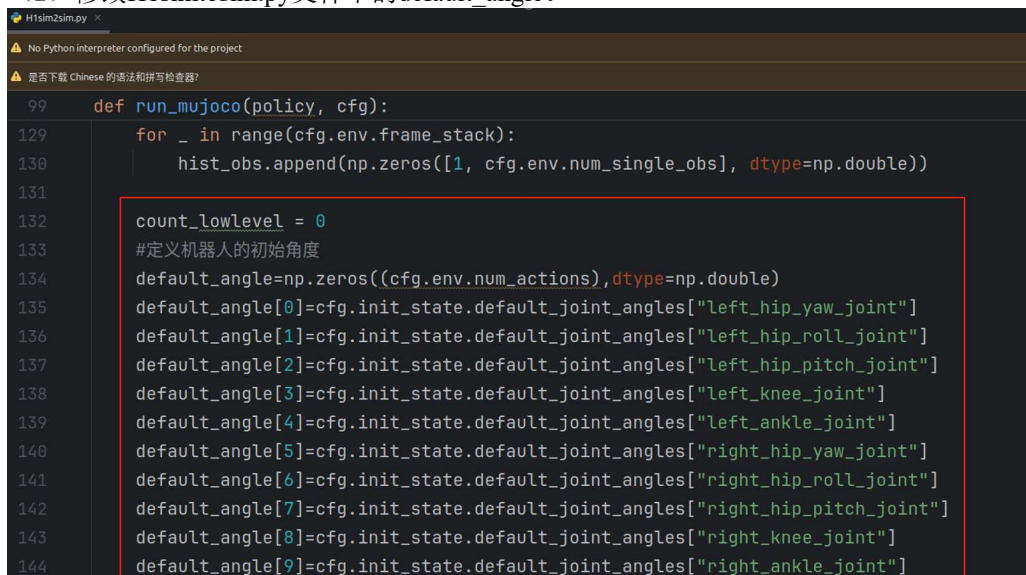
```
42 class H1_fix_arm_FreeEnv(LeggedRobot):
76     def __init__(self, cfg: LeggedRobotCfg, sim_params, physics_engine, sim_device, headless):
77         super().__init__(cfg, sim_params, physics_engine, sim_device, headless)
78         self.last_feet_z = 0.05
79         self.feet_height = torch.zeros((self.num_envs, 2), device=self.device)
80         self.reset_idx(torch.tensor(range(self.num_envs), device=self.device))
81         self.compute_observations()
82         self.default_dof_pos = torch.zeros(self.num_dof, dtype=torch.float, device=self.device, requires_grad=False)
83         # default_angle = np.zeros((cfg.env.num_actions), dtype=np.double)
84         self.default_dof_pos[0] = cfg.init_state.default_joint_angles["left_hip_yaw_joint"]
85         self.default_dof_pos[1] = cfg.init_state.default_joint_angles["left_hip_roll_joint"]
86         self.default_dof_pos[2] = cfg.init_state.default_joint_angles["left_hip_pitch_joint"]
87         self.default_dof_pos[3] = cfg.init_state.default_joint_angles["left_knee_joint"]
88         self.default_dof_pos[4] = cfg.init_state.default_joint_angles["left_ankle_joint"]
89         self.default_dof_pos[5] = cfg.init_state.default_joint_angles["right_hip_yaw_joint"]
90         self.default_dof_pos[6] = cfg.init_state.default_joint_angles["right_hip_roll_joint"]
91         self.default_dof_pos[7] = cfg.init_state.default_joint_angles["right_hip_pitch_joint"]
92         self.default_dof_pos[8] = cfg.init_state.default_joint_angles["right_knee_joint"]
93         self.default_dof_pos[9] = cfg.init_state.default_joint_angles["right_ankle_joint"]
```

## 五. Hlsimtosim

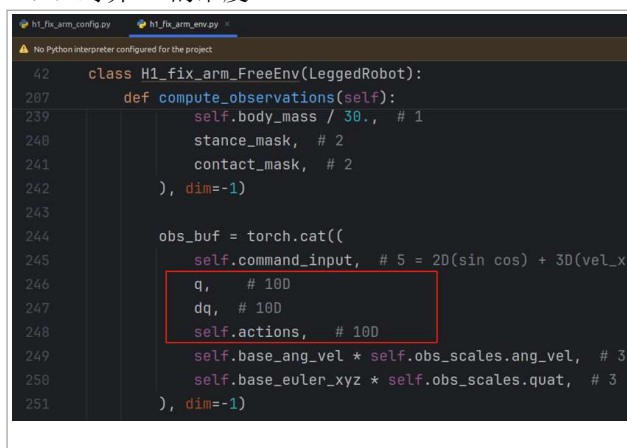
(1) 创建H1sim2sim.py文件，文件中代码复制于  
sim2sim.py



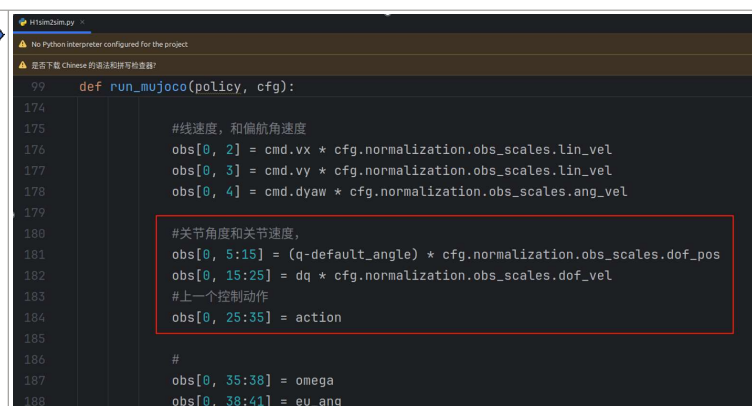
(2) 修改H1sim2sim.py文件中的default\_angle。



(3) 对齐obs的维度



根据  
状态  
空间  
维度  
进行  
修改



参考链接:

[https://www.bilibili.com/video/BV19r421T7M8/?spm\\_id\\_from=333.1391.0.0](https://www.bilibili.com/video/BV19r421T7M8/?spm_id_from=333.1391.0.0)

