项目笔记

2025.4.18

资料汇总

宇树G1: https://support.unitree.com/home/zh/G1_developer/about_G1
SMPL论文: https://files.is.tue.mpg.de/black/papers/SMPL2015.pdf
Hybrik: https://github.com/jeffffffli/HybrIK

考核目标

将已经有的提取到的关节坐标信息，转换成电机的参数。

参考教程：

Anaconda安装教程

/tmp空间不足解决方案

装cuda时的驱动冲突问题

安装推荐驱动教程

cuda和cudnn安装教程

/home盘满了解决方案

ubuntu更换清华源

cuda和cudnn版本对应关系

由于之前没有用ubuntu系统跑过深度学习环境，这次task第一步是要先配置anaconda、显卡驱动、cuda和cudnn。之前尝试配置显卡驱动但碰到了一些问题（装完后无法启动电脑等），这次通过多种方法尝试并解决。

部分遇到的问题：

1. 磁盘空间不足导致的问题

/tmp空间不足（一开始也怀疑是安装包损坏），导致cuda安装失败。通过在/home中创建代替/tmp的文件夹，并修改--tmpdir解决。

```
(base) yth@yth-Legion-Y9000P-IAH7H:~$ sudo sh cuda_11.8.0_520.61.05_linux.run
Extraction failed.
Ensure there is enough space in /tmp and that the installation package is not co
rrupt
Signal caught, cleaning up
```

```
(base) yth@yth-Legion-Y9000P-IAH7H:~$ df -h /tmp
文件系统        容量  已用  可用 已用% 挂载点
/dev/nvme1n1p9   14G  7.5G  5.9G   57% /
```

/home空间不足，导致无法安装cuda。找教程删文件解决。

```
------------+
(base) yth@yth-Legion-Y9000P-IAH7H:/home$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

/usr满了（cuda占空间较大），导致电脑无法开机。起初怀疑无法开机是安装驱动版本不对导致的，删掉驱动重装了一遍，发现依然没用。最后换内核进恢复模式clean了才能重启。

注：后悔以前装ubuntu没有多分配空间，导致现在碰到各种问题。

2. Isaac Gym例程部署

gpu调用

起初直接用了文档里的torch安装指令，但发现torch版本太旧，导致和cuda版本不兼容。

（报错显示physics device只调用了cpu）

```
(rl-g1) yth@yth-Legion-Y9000P-IAH7H:~/project/isaacgym/python$ python examples/1080_balls_of_solitude.py
Importing module 'gym_38' (/home/yth/project/isaacgym/python/isaacgym/_bindings/linux-x86_64/gym_38.so)
Setting GYM_USD_PLUG_INFO_PATH to /home/yth/project/isaacgym/python/isaacgym/_bindings/linux-x86_64/usd/plugInf
o.json
WARNING: Forcing CPU pipeline.
[Error] [carb.gym.plugin] Sim CUDA device 0 can't be set, the total number of available devices is -1
Not connected to PVD
/buildAgent/work/99bede84aa0a52c2/source/cudamanager/src/CudaContextManager.cpp (404) : warning : cuInit failed

[Warning] [carb.gym.plugin] Failed to create a valid PhysX CUDA Context Manager. Falling back to CPU.
Physics Engine: PhysX
Physics Device: cpu
GPU Pipeline: disabled
[Error] [carb.gym.plugin] Gym cuda error: unknown error: ../../../source/plugins/carb/gym/impl/Gym/GymCuda.h: 1
10
[Error] [carb.gym.plugin] Failed to create primary CUDA context
[Warning] [carb.gym.plugin] Failed to create primary CUDA context on graphics device
MESA-INTEL: warning: Performance support disabled, consider sysctl dev.i915.perf_stream_paranoid=0

*** Failed to load '../../assets/urdf/ball.urdf'[Error] [carb.gym.plugin] Failed to parse URDF file 'urdf/ball.
urdf'
[Error] [carb.gym.plugin] Failed to import URDF file 'urdf/ball.urdf'
[Error] [carb.gym.plugin] *** Failed to load 'urdf/ball.urdf' from '../../assets'
段错误 (核心已转储)
```

最后还是上官网找对应版本手动安装了torch，成功调用gpu。

模型训练-代码框架测试

出现部分报错：

1. numpy版本问题(（dtype=np.float)）：

```
AttributeError: module 'numpy' has no attribute 'float'.
`np.float` was a deprecated alias for the builtin `float`. To avoid this error in existing code, use `float` by
 itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type,
 use `np.float64` here.
The aliases was originally deprecated in NumPy 1.20; for more details and guidance see the original release not
e at:
    https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
```

```
# 换成：
dtype=float
```

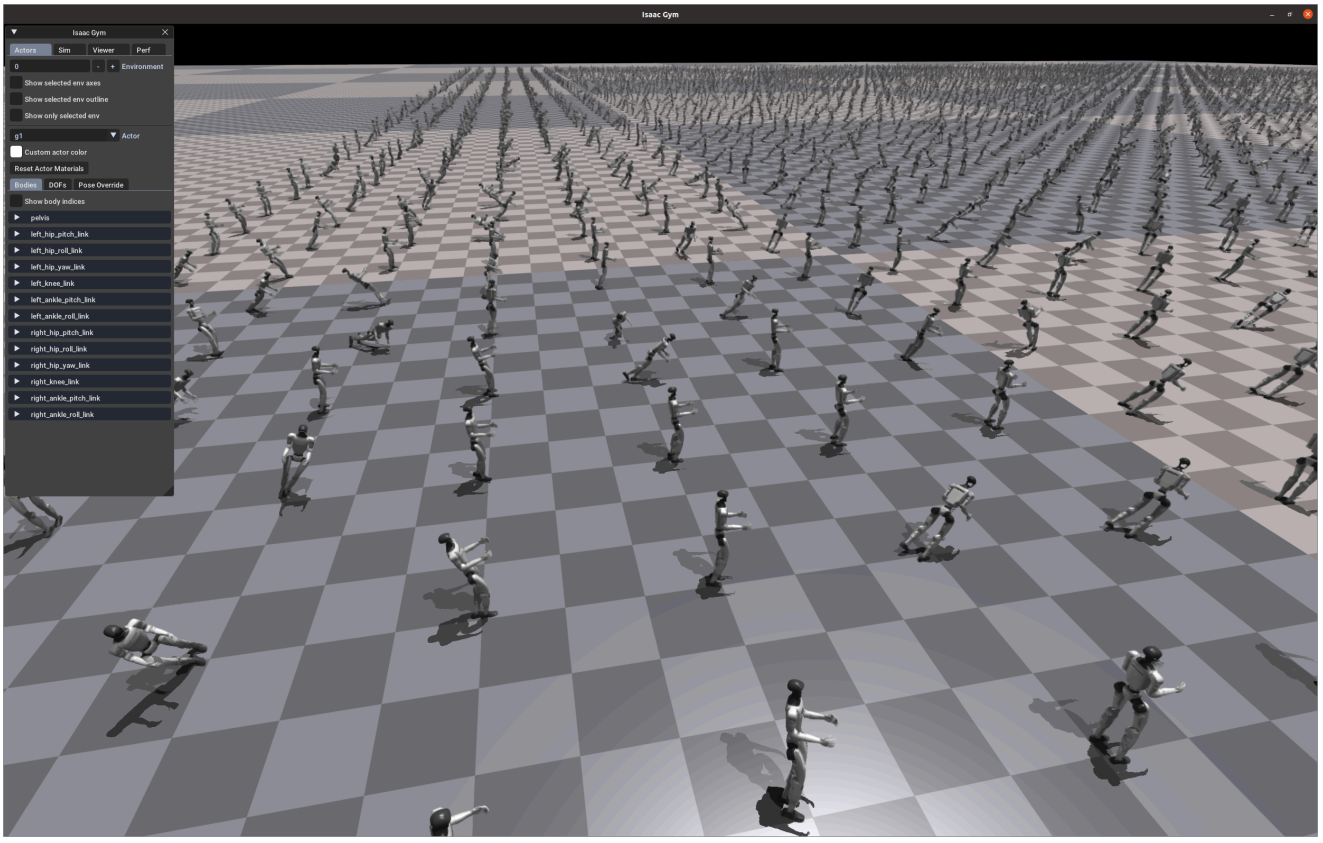
2. rsl_rl外部库可见性（需配置环境变量）：

```
ModuleNotFoundError: No module named 'rsl_rl'
```

```
# 终端输入
export PYTHONPATH=$PYTHONPATH:/home/yth/project/rsl_rl
source ~/.bashrc
# 验证安装成功的命令
python -c "import rsl_rl; print(rsl_rl.__file__)"  # 应显示库路径
```

解决后即可运行train文件并可视化。

我的电脑跑起来比较卡顿，打算挂着训练一晚上看看效果。



3. tensorboard模块未安装

```
pip install tensorboard
```

4. legged_gym（module）可见性

```
# 代码：目录路径的添加（需要父目录）
import sys
sys.path.append("/home/yth/project/unitree_rl_gym")  # 添加这一行
sys.path.append("/home/yth/project/unitree_rl_gym/legged_gym")
```

2025.4.19

跑了一晚上训练结果如下：

```
yth@yth-Legion-Y9000P-IAH7H: ~/project/unitree_rl_gym/legged_gym/scripts

                    Total timesteps: 64389120
                     Iteration time: 44.93s
                         Total time: 29483.30s
                                ETA: 420688.4s

################################################################################
                      Learning iteration 655/10000

                       Computation: 2190 steps/s (collection: 44.692s, learning
0.178s)
               Value function loss: 0.0091
                    Surrogate loss: -0.0113
             Mean action noise std: 0.62
                       Mean reward: 8.23
               Mean episode length: 743.42
      Mean episode rew_action_rate: -0.0991
            Mean episode rew_alive: 0.1065
       Mean episode rew_ang_vel_xy: -0.0532
      Mean episode rew_base_height: -0.0042
          Mean episode rew_contact: 0.1697
   Mean episode rew_contact_no_vel: -0.0463
          Mean episode rew_dof_acc: -0.0897
   Mean episode rew_dof_pos_limits: -0.0153
          Mean episode rew_dof_vel: -0.0854
Mean episode rew_feet_swing_height: -0.0156
          Mean episode rew_hip_pos: -0.0392
        Mean episode rew_lin_vel_z: -0.0600
      Mean episode rew_orientation: -0.0075
          Mean episode rew_torques: -0.0286
 Mean episode rew_tracking_ang_vel: 0.1192
 Mean episode rew_tracking_lin_vel: 0.4100
--------------------------------------------------------------------------------
                    Total timesteps: 64487424
                     Iteration time: 44.87s
                         Total time: 29528.17s
                                ETA: 420641.4s
```

测试结果：

```
yth@yth-Legion-Y9000P-IAH7H: ~/project/unitree_rl_gym/legged_gym/scripts
Traceback (most recent call last):
  File "/home/yth/project/unitree_rl_gym/legged_gym/scripts/play.py", line 2, in <module>
    from legged_gym import LEGGED_GYM_ROOT_DIR
ModuleNotFoundError: No module named 'legged_gym'
(base) yth@yth-Legion-Y9000P-IAH7H:~/project/unitree_rl_gym/legged_gym/scripts$ python play.py --task=g1
Traceback (most recent call last):
  File "/home/yth/project/unitree_rl_gym/legged_gym/scripts/play.py", line 9, in <module>
    import isaacgym
ModuleNotFoundError: No module named 'isaacgym'
(base) yth@yth-Legion-Y9000P-IAH7H:~/project/unitree_rl_gym/legged_gym/scripts$ conda activate rl-g1
(rl-g1) yth@yth-Legion-Y9000P-IAH7H:~/project/unitree_rl_gym/legged_gym/scripts$ python play.py --task=g1
Importing module 'gym_38' (/home/yth/project/isaacgym/python/isaacgym/_bindings/linux-x86_64/gym_38.so)
Setting GYM_USD_PLUG_INFO_PATH to /home/yth/project/isaacgym/python/isaacgym/_bindings/linux-x86_64/usd/plugInf
o.json
PyTorch version 2.4.1
Device count 1
/home/yth/project/isaacgym/python/isaacgym/_bindings/src/gymtorch
Using /home/yth/.cache/torch_extensions/py38_cu118 as PyTorch extensions root...
Emitting ninja build file /home/yth/.cache/torch_extensions/py38_cu118/gymtorch/build.ninja...
Building extension module gymtorch...
Allowing ninja to set a default number of workers... (overridable by setting the environment variable MAX_JOBS=
N)
ninja: no work to do.
Loading extension module gymtorch...
Setting seed: 1
Not connected to PVD
+++ Using GPU PhysX
Physics Engine: PhysX
Physics Device: cuda:0
GPU Pipeline: enabled
MESA-INTEL: warning: Performance support disabled, consider sysctl dev.i915.perf_stream_paranoid=0

/home/yth/anaconda3/envs/rl-g1/lib/python3.8/site-packages/torch/functional.py:513: UserWarning: torch.meshgrid
: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at /opt/cond
a/conda-bld/pytorch_1724789115405/work/aten/src/ATen/native/TensorShape.cpp:3609.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
'train_cfg' provided -> Ignoring 'name=g1'
Actor MLP: Sequential(
  (0): Linear(in_features=64, out_features=32, bias=True)
  (1): ELU(alpha=1.0)
  (2): Linear(in_features=32, out_features=12, bias=True)
)
Critic MLP: Sequential(
  (0): Linear(in_features=64, out_features=32, bias=True)
  (1): ELU(alpha=1.0)
  (2): Linear(in_features=32, out_features=1, bias=True)
)
Actor RNN: Memory(
  (rnn): LSTM(47, 64)
)
Critic RNN: Memory(
  (rnn): LSTM(50, 64)
)
Loading model from: /home/yth/project/unitree_rl_gym/logs/g1/Apr19_03-18-51_/model_650.pt
/home/yth/project/rsl_rl/rsl_rl/runners/on_policy_runner.py:222: FutureWarning: You are using `torch.load` with
 `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possi
ble to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.
com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default
value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unp
ickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allo
wlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=Tr
ue` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for a
ny issues related to this experimental feature.
  loaded_dict = torch.load(path)
Exported policy as jit script to:  /home/yth/project/unitree_rl_gym/logs/g1/exported/policies
```

代码框架测试没问题，开始尝试上手改代码。

G1配置修改

首先需要知道G1的配置在哪里，如何设定初始状态和数量等初始化条件（现在的代码还是初始化N个机器人）。经过对例程代码的初步熟悉，确定了在 `legged_gym/envs/g1/g1_config.py` 中。

```
    class asset( LeggedRobotCfg.asset ):
        file =
'{LEGGED_GYM_ROOT_DIR}/resources/robots/g1_description/g1_12dof.urdf'
        name = "g1"
        foot_name = "ankle_roll"
        penalize_contacts_on = ["hip", "knee"]
        terminate_after_contacts_on = ["pelvis"]
        self_collisions = 0 # 1 to disable, 0 to enable...bitwise filter
        flip_visual_attachments = False
```

尝试用AI写了个单机器人站立和简单运动的demo测试一下，发现上半身关节检测不到，整个机器人只有12个自由度，怀疑是模型的问题。于是在官方文档-关于G1栏目中找到了电机相关信息：

12自由度版本应该只含有下半身，不符合task需求。于是重新找了文件里G1的urdf模型（位于legged_gym/resources/robots/g1_description）。考虑到灵巧手较为复杂暂时用不到，选用了23-DOF版本。

直接运行会报错，需在config文件中加入另外11个关节(class init_state)。对应的，环境中的动作维度和观察维度也需要增加（class env）。

```
  File "/home/yth/project/IK_humanoid/un
    super().__init__buffers()
  File "/home/yth/project/IK_humanoid/un
    angle = self.cfg.init_state.default_
KeyError: 'waist_yaw_joint'
```

此外，在demo中加入了固定基座的功能，从而稳定下半身。

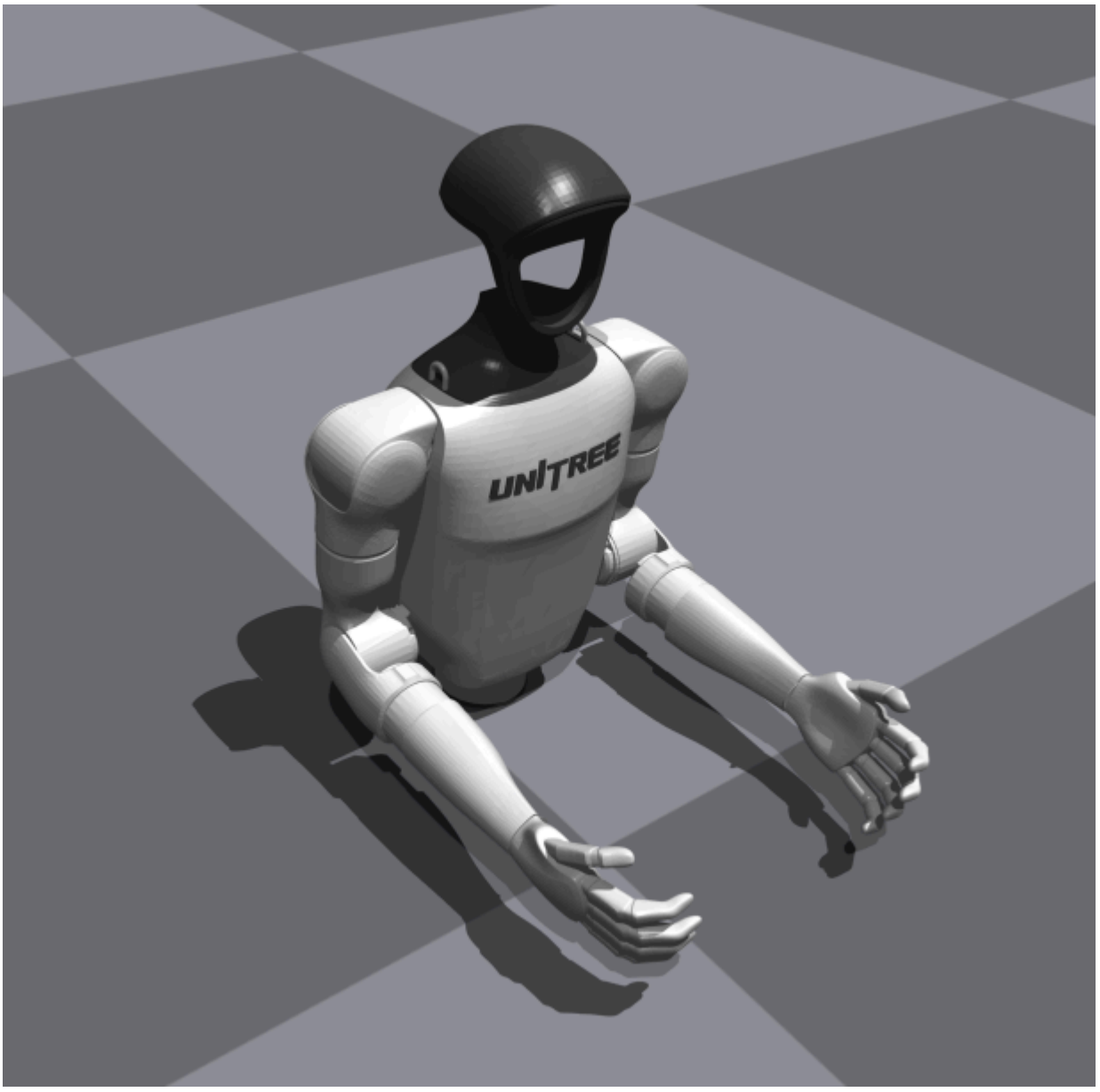
```
env_cfg.asset.fix_base_link = True # 固定基座
```

再次运行显示新加入关节的PD未定义：

```
PD gain of joint waist_yaw_joint were not defined, setting them to zero
PD gain of joint left_shoulder_pitch_joint were not defined, setting them to zero
PD gain of joint left_shoulder_roll_joint were not defined, setting them to zero
PD gain of joint left_shoulder_yaw_joint were not defined, setting them to zero
PD gain of joint left_elbow_joint were not defined, setting them to zero
PD gain of joint left_wrist_roll_joint were not defined, setting them to zero
PD gain of joint right_shoulder_pitch_joint were not defined, setting them to zero
PD gain of joint right_shoulder_roll_joint were not defined, setting them to zero
PD gain of joint right_shoulder_yaw_joint were not defined, setting them to zero
PD gain of joint right_elbow_joint were not defined, setting them to zero
PD gain of joint right_wrist_roll_joint were not defined, setting them to zero
获取初始观察...
G1机器人已固定，保持站立状态
```

在class control中加入阻尼和刚度参数即可。加入后手部由原本的松弛平摊状态转为固定状态。

UNITREE



手部动作尝试，确认关节电机映射关系

根据官网上的关节序列映射（底层运动控制-关节电机顺序），在代码中加入了对应的joint_indices。测试后发现对应关系和仿真里不一样，直接改成了按顺序排列。（官网放的可能是实物版本）

23DOF 版本 (LowCmd_.mode_machine == 1)

Joint Index in IDL	Joint Name (LowCmd_.mode_pr or LowState_.mode_pr == 0)	Joint Name (LowCmd_.mode_pr ==
0	L_LEG_HIP_PITCH	L_LEG_HIP_PITCH
1	L_LEG_HIP_ROLL	L_LEG_HIP_ROLL
2	L_LEG_HIP_YAW	L_LEG_HIP_YAW
3	L_LEG_KNEE	L_LEG_KNEE
4	**L_LEG_ANKLE_PITCH**	**L_LEG_ANKLE_B**
5	**L_LEG_ANKLE_ROLL**	**L_LEG_ANKLE_A**
6	R_LEG_HIP_PITCH	R_LEG_HIP_PITCH
7	R_LEG_HIP_ROLL	R_LEG_HIP_ROLL
8	R_LEG_HIP_YAW	R_LEG_HIP_YAW
9	R_LEG_KNEE	R_LEG_KNEE
10	**R_LEG_ANKLE_PITCH**	**R_LEG_ANKLE_B**
11	**R_LEG_ANKLE_ROLL**	**R_LEG_ANKLE_A**
12	WAIST_YAW	WAIST_YAW
13	(empty)	(empty)
14	(empty)	(empty)
15	L_SHOULDER_PITCH	L_SHOULDER_PITCH
16	L_SHOULDER_ROLL	L_SHOULDER_ROLL

分步尝试关节，检查对应关系是否正确。测试腰部yaw角时，发现2pi并不对应一圈，说明需要运动学建模。

部分测试代码如下：

```
print("演示3: 躯干旋转")
for t in range(300):
    actions = zero_action.clone()
    phase = t / 300.0 * 2 * math.pi

    # 躯干左右旋转
    actions[0, joint_indices['waist_yaw']] = 5 * math.sin(phase)

    obs, _, _, _, _ = env.step(actions)
    time.sleep(0.01)
```

```
        # 回到零位
        print("回到初始位置...")
        for i in range(100):
            obs, _, _, _, _ = env.step(zero_action)
            time.sleep(0.01)
```

左肩前举测试：

右肘上抬测试：

> 经过验证发现，按照对应视角，左手角度设为负值是往前上方抬手（即电机顺时针旋转），右手角度设为负值是往前上方抬手（即电机逆时针旋转），腰部yaw角设为负值是向右转（即顺时针）。

> 这里也发现一个问题，貌似是换成23dof后仿真的画面放缩比例变了，导致机器人看起来很小。暂时没有研究怎么解决。

逆运动学建模尝试

逆运动学首先需要肢体末端的位置点序列。

张量维度问题

经过查找，打算用张量API：**envs.rigid_body_states**来获取每个刚体的状态属性（位置），但维度上出了一些问题：

```
/home/yth/anaconda3/envs/rl-g1/lib/python3.8/site-packages/torch
o pass the indexing argument. (Triggered internally at /opt/cond
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-de
获取初始观察...
G1机器人已固定，总关节数量: 23
初始化刚体状态...
刚体总数: 13
使用左手索引: 12, 右手索引: 12
让机器人稳定下来...
刚体状态张量形状: torch.Size([24, 13])
执行过程中发生错误: too many indices for tensor of dimension 2
Traceback (most recent call last):
```

根据输出结果可以看出，rigid_body_states应当是被强行展平成二维后形状变成了[24,13]，然而一开始我设想的是[1,23,13]，分别代表环境数量、刚体数量（关节电机共23个）、刚体属性（position、orientation、linear_velocity、angular_velocity共13个），根据数据特点，我猜测展平操作导致env被迫压缩加了上去变成了24个。

于是针对问题，给对应手部的序列号+1，相当于跳过了第一位的env，使得数据形状符合预期。

> 推测思路：通过以下输出（左右手的位置）确定了env是在第一位而不是最后一位，因为输出的两个位置x和z位置几乎相同，只有y相差0.3米，符合实际情况。

```
G1机器人已固定，总关节数量: 23
左手索引: 18, 右手索引: 23
让机器人稳定下来...
刚体总数: 24
使用左手索引: 18, 右手索引: 23
刚体状态张量形状: torch.Size([24, 13])
左手初始位置: tensor([ 0.8896, -0.7984,  0.0889], device='cuda:0')
右手初始位置: tensor([ 0.8916, -1.0918,  0.0899], device='cuda:0')
```