**首先，在预抓取训练的时候，可以在isaac里面调整手部的位姿，记录容易抓取物体手部的位姿，将其在训练的时候初始化为初始手部动作。----对于训练的转动,训练时间越久奖励可能会更低,要控制时间回合.**

**---**

**在这个强化学习里面，他的任务环境是通用的，就是利用了域随机化，将一些扰动因素进行随机化，训练一个通用且范化的一个策略，他将物体的size和mass都进行随机化，并且加入了一些干扰力。。。。对物体尺寸进行域随机化，，提高策略的范化能力。。。**

**------**

**SDK的编译和下载：在网上下载最新的SDK包之后，在编译时可能会出现缺失empy的问题。**

1. **首先查看python的版本和pip和pip3的版本，确认python版本在3.8.**
2. **然后进行pip install empy下载这个包。**
3. **下载完之后可以在pip show empy查看版本号，有时候需要特定版本的empy，这就需要我们先卸载这个版本的empy，然后在下载制定版本的empy。**
4. **这样就有了empy，之前如果编译过，需要先把build和devel这两个文件夹删除，然后在catkin\_make。**
5. **如果在缺少别的包，就直接下载就可以。**

**-----**

**进行旋转时，可能会出现的是手一直拿着方块做抖动以此来获取奖励，加入新的奖励函数，将转动方向一致性作为旋转奖励。**

**-----**

**在创建一个新的任务时，isaac里面的模型要尽可能的去贴近现实，比如说我在将右手换成左手时，左手可能是因为关节质量或者刚度和阻尼的设置不同，产生了因重力原因的自然下落。（我将重力设为0进行了实验）。。。然后我将刚度和阻尼和力矩进行了调整，可以不因为重力影响而下落。。。（没有考虑机械手的pd控制;;;待解决。。。）------在换手的时候应该是urdf的问题，我将urdf中的inertial惯性张量进行了替换，替换成了与右手完全相同的，解决了这个下垂的问题，具体影响还待研究。。。  
inertial惯性张量的理解：惯性张量的origin是惯性属性相对于物体的哪个点进行计算，他的xyz和rpy决定了惯性张量在物体的定位和旋转。**

**-----**

**在isaac中，每一个环境任务的智能体可能都不一样，利用rl-games的开源库里面的runner注册工厂，利用工厂进行注册的name进行create相对应的智能体。智能体就是对应的强化学习算法**

**------**

**num\_position\_iterations位置迭代次数的作用是发生碰撞的后，可能会因为计算误差导致位置偏移，这个参数的作用就是在每个仿真步中迭代修正这些误差。**

**在仿真中，出现关节松动或者夹不稳，可以适当的提高一下。关节可能有多个自由度**

**bounce\_threshold\_velocity反弹阈值速度，这个可以控制物体在发生碰撞时会不会反弹，低于设置的这个值就会反弹。**

**max\_depenetration\_velocity最大去重叠速度，当两个物体发生穿透时，最大允许的去重叠速度，可以减少物体在手指间弹跳。**

**collision\_group碰撞组，只有两个物体在同一个组内才会产生碰撞。在create\_actor内使用，collision\_filter是一个位掩码，如果设置了相同的位的话，就不会碰撞。0会碰撞.**

**------**

**fps小的原因是什么总结可能的原因。计算量太大？还没有弄清楚是哪里的计算量大，根据网上和isaac实例分析：**

**第一种可能：手部各个关节的effort是否过大，在转动的时候直接将方块推走。**

**第二种：碰撞过滤，是否没有将这个手部的碰撞进行过滤。刚体数量变多。**

**第三种：是否可以将horizon\_length这个参数进行减少，这个参数是训练horizon\_length个步长之后进行梯度更新策略，过低会不稳定，过高会增加训练时间。策略更新慢。**

**第四种：观测空间增大，神经网络推理变慢，是否可以利用减少网络结构来优化计算。**

**找到问题：我将isaac中的凸分解去掉之后，训练的fps显著增大，占用的显存也降低了很多，初步的分析应该是凸分解使手部模型的碰撞计算变的更大了，导致计算量增大，fps提高不上去。**

**凸分解是什么：凸分解（VHACD）会在加载模型时对复杂形状进行凸包分解，首先会增加初始化的时间，并且在多个凸包在物理仿真时会单独进行碰撞检测，增加了GPU计算量。凸分解的作用是优化复杂模型的碰撞检测，但会增加计算成本。凸分解将凹形状的刚体进行凸包分解,凹形状不容易进行碰撞,在物理仿真中无法得到充分考虑.**

**平衡凸分解和fps：可以将关键部位进行凸分解，或者将凸分解的程度降低**

**hand\_asset\_options.vhacd\_params.max\_convex\_hulls = 10 # 默认 64，改小点，减少凸包数量**

**hand\_asset\_options.vhacd\_params.resolution = 50000 # 默认 100000~300000，降低细节**

**-------**

**通过ssh连接本地网络的linux主机,知道ip地址和密码就可以,还可以使用scp命令进行传输数据,两台主机之间.**

**当pytorch加载不进去的时候，清理pytorch缓存。重新编译他**

**-------**

**在缓存抓取位姿的时候，为了记录手部姿态的多样性，随机化初始一个自由度张量，设定一个在这个姿态的抓取搜索半径，搜索半径就是一个权重，将其与随机初始化的自由度张量与记录的自由度相加，这样就会在设定自由度附近进行寻找合适的抓取姿态。**

**--------**

**在代码中，为了符合现实中的情况，对物体（方块）设定了一个轴和随机角度进行转动，目的就是稳定让手部抓取成功。让缓存数据更加健壮。**

**-------**

**他的执行顺序是这样的，根据每一个环境的索引进行重置环境，reset。查看哪个环境到达了重置环境的条件，就根据索引对其进行reset\_idx进行重置环境。**

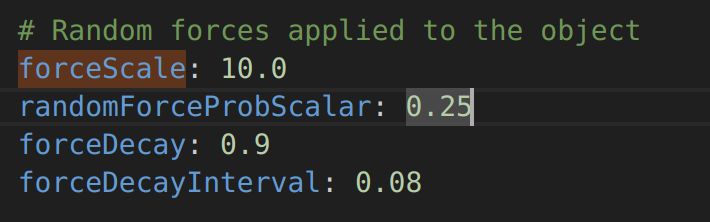
**--------**

**还有就是isaac 里面的api是读取urdf文件里面的自由度，关节，属性等东西，是直接利用函数将其自由度等信息存储在一个张量当中，**

1. **在文件里面leapsim/tasks/leap\_hand\_rot.py，这个文件是与搭建任务环境相关的文件，里面包含了奖励函数的设计，状态空间的观测，下面是我对这个文件里面函数的理解；**

**-------------**

**物体的相关设置:总的来说就是在训练环境中,生成不同尺寸,不同环境扰动力的一个方块.**



**这些参数是对物体施加一个扰动力和一些力衰减比例 ,随机扰动力设置,让物体在动态环境里更有鲁棒性.**

**randomizeScaleList: []**

**这个参数是物体随机尺寸的一个list,在里面随机选择尺寸,他在训练时是训练一个通用更加范化的策略,适用于绝大多数尺寸的方块.在创建并行环境时会随机初始化方块尺寸.**

**-------------**

\_create\_envs函数：

是一个在物理仿真环境下创建模型的一个函数，在里面配置手部的模型和一个物体的相关参数，并且将多个环境实例配置到gym中（并行训练）。------应该可以在这个函数里面修改模型，将自己的模型参数等导入进行训练。

compute\_reward(self, actions)函数：

这个函数就是根据智能体选择的动作进行计算奖励。并在必要时可以进行环境重置。

compute\_hand\_reward函数：

这个函数在compute\_reward函数里面调用，用途是计算手的相关控制的奖励或惩罚。

compute\_observations函数：

计算手部的观测信息，并更新 self.obs\_buf，它在环境中用于为策略提供状态输入。观测空间，更新obs\_buf作为强化学习的输入。

post\_physics\_step：

每个物理时间步结束后进行调用，就是每次执行智能体的动作后，这个函数就会被调用。用来更新环境并且提供最新的观测数据。

\_create\_ground\_plane(self):

创建仿真环境的函数

pre\_physics\_step(self, actions)

每一次物理仿真步骤以前调用，主要是更新动作，目标以及施加力等物理状态。

reset(self)

重置环境函数

check\_termination(self, object\_pos)

查看物体的状态并且检查是否需要重置环境。

\_refresh\_gym(self):

同步物理仿真的最新状态，获取手部和物体的位姿，速度等信息。

\_create\_object\_asset(self):

这个就是将手部和物体进行创建的文件。在物理仿真环境中加载并配置物体和手部模型的资产，以及对物体和手部刚体形状属性的设置。

\_init\_object\_pose(self)

这个函数就是初始化物体和手部位姿的函数。

\_refresh\_gym

这个函数就是刷新物理仿真的，这里面有手部的坐标，物体的坐标和位姿和线速度和角速度

hand\_asset\_options.flip\_visual\_attachments = True

这个是是否翻转视角的插件函数，true为可以翻转，false为不能翻转。

1. **cfg文件夹里面，这是一个任务和训练配置文件夹。里面包含了任务的各项配置和训练的各项配置。**

numObservations: 102

numActions: 16

**根据里面的配置知道，观测空间是102维，动作空间是16维的，手的关节是16，应该对应的是每一个关节**