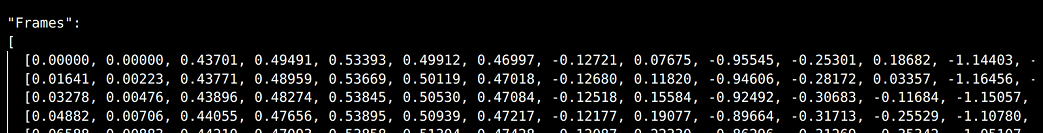
**Pros:**  
RL能减少手动调参，但是训练过程得到的策略会过度依赖仿真的Dynamics(运动参数)和虚拟环境，使得sim2real的过程很难实现。

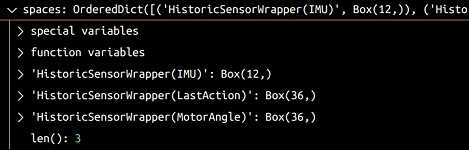
由于simulator无法做到很好地仿真真实世界的dynamics，就有了Domain randomization来be robust to variations in the dynamics。

**Motion Retargeting**  
根据每一帧关键点的位置逆向求解得12个关节电机对应的角度，存储在dog\_pace.txt中的数据并不是12维的，每一帧为19维，还包含3+4维的root pose：  


**Motion Imitation**

状态变量：

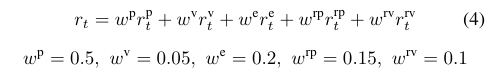
由t, t-1, t-2三个时刻的位姿和t-1, t-2, t-3三个时刻的动作组成，总共3\*(12+3)+3\*(12)=81维，在编程时得到的orientation是四元数所以的维度是84：



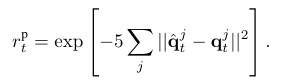
策略函数

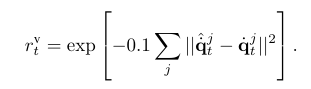
策略网络的输入包括状态变量和预期位姿变量，所以环境中的observation总维度，即策略网络的输入为84+4\*19=160维。

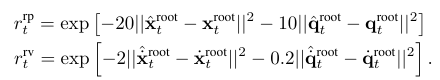
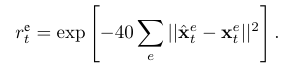
奖励函数

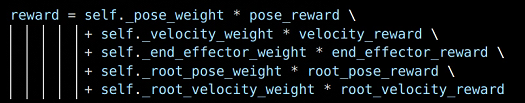
由五项组成：其实分别就是每一时刻的：①关节位置误差 ②关节速度误差 ③足端位置误差 ④躯体位置及线速度误差 ⑤躯体角度及角速度误差  


其中：

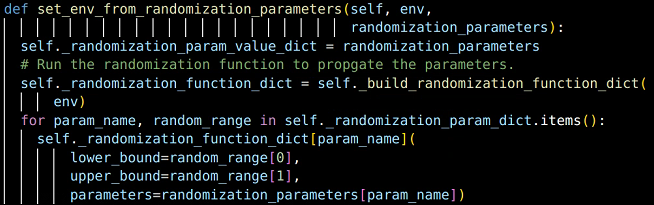




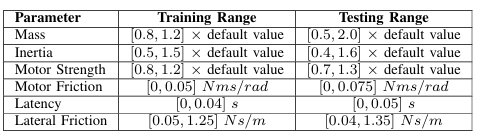


对应代码：  


**Domain adaptation**

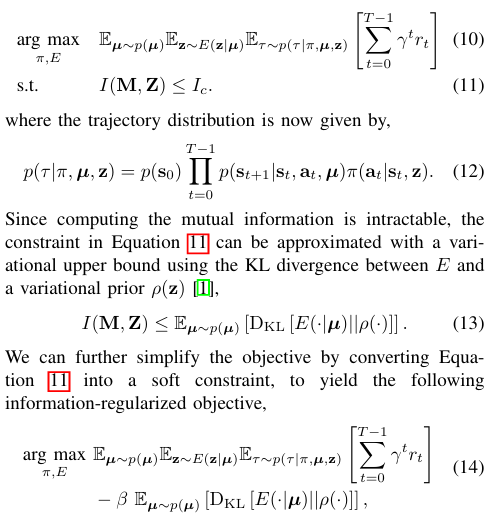
传统的Domain adaptation会通过随机化仿真环境中的参数dynamics来提高模型的鲁棒性，代码中在建立环境env时就会通过randomizer来实时调整环境参数，比如：  


代码中看到是在如下范围内均匀采样：

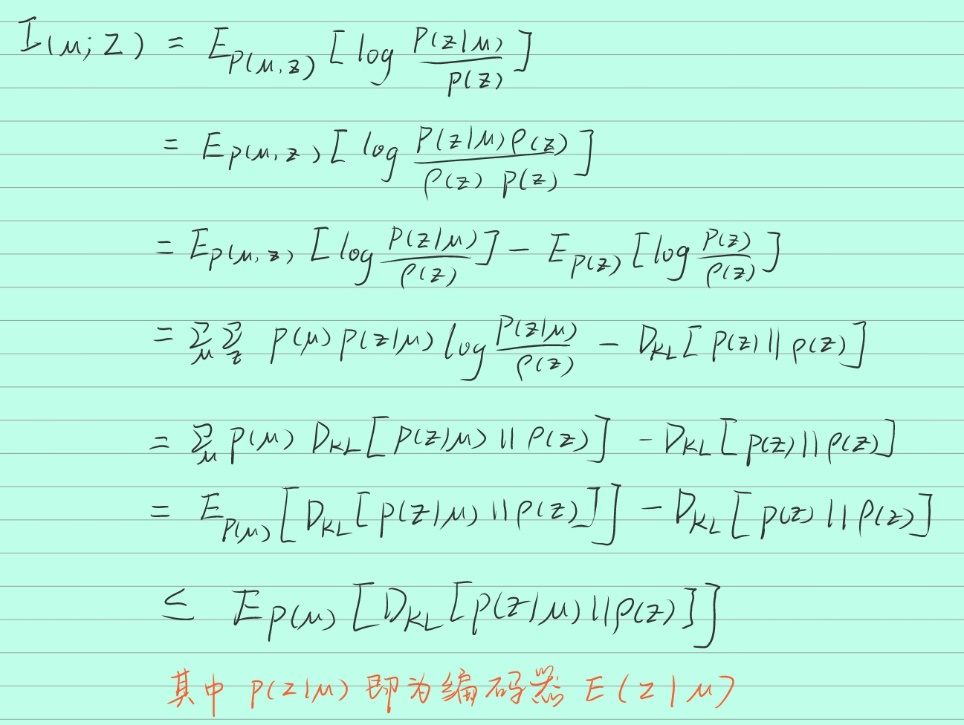


但是每当迁移到真实环境中时效果总是不佳。文中提出一种方法，先将采样到的环境参数

编码到一个隐空间：， 然后将编码得到的z一起作为输入给到策略函数：。如下，这种做法的关键在于减少和z的互信息，也就是说策略函数在计算action value的时候会把经过编码的环境参数也考虑在内，而为了避免策略函数过拟合于仿真环境参数，应减小编码前后环境参数分布的互信息。这里的互信息用了一个变分上界来进行优化：



公式（13）的具体推导应如下：



优化目标中用一个惩罚因子来调整互信息的惩罚。时对应过拟合，无穷时对应non-adaptive.

各部分网络结构如下：

