强化学习环境

roslaunch moveit\_setup\_assistant setup\_assistant.launch

roslaunch p\_robot\_gazebo ur\_pick\_gazebo.launch

roslaunch p\_robot\_moveit\_config demo.launch sim:=true

roslaunch t\_robot\_gazebo ur\_pick\_gazebo.launch

roslaunch t\_robot\_moveit\_config demo.launch sim:=true

roslaunch p\_robot\_gazebo ur\_pick\_gazebo.launch

roslaunch p\_robot\_moveit\_config demo.launch sim:=true

roslaunch p\_robot\_learning\_demo demo\_node.launch

rosrun p\_robot\_learning\_demo p\_robot\_go\_server

重开一个终端

5. condapy3 // 查看：<https://blog.csdn.net/penge666/article/details/88692605>

6. rosrun p\_robot\_learning\_demo p\_robot\_rl.py

|  |  |  |  |
| --- | --- | --- | --- |
| dlut\_robot\_learning | | | |
| // meta-packages  // Author：卢彬鹏  // email：lbp\_and1@163.com | | | |
| 1. | gazebo\_grasp\_plugin | | |
|  | // gazebo抓取插件，用于仿真环境中robotiq抓取物体  // 来自：<https://github.com/JenniferBuehler/gazebo-pkgs/tree/master/gazebo_grasp_plugin>，参数设置自行查看wiki。 | | |
| 2. | gpml | | |
|  | // 高斯过程matlab工具箱  // 来自：<http://www.gaussianprocess.org/gpml/>。 | | |
| 3. | LSTM Networks\_u | | |
|  | // LSTM神经网络程序 | | |
|  |  | LSTM\_train.m | |
|  |  | // 训练主函数 | |
|  |  | Lstm\_Layer.m | |
|  |  | // LSTM神经网络单元，权值保存到weights文件夹 | |
|  |  | LSTM\_predict.m | |
|  |  | // 示范如何使用训练好的权值文件 | |
|  |  | // 其余的自行理解 | |
| 4. | p\_robot\_description | | |
|  | // 机器人模型文件 | | |
|  |  | meshes | |
|  |  | // 主要为stl模型 | |
|  |  | urdf | |
|  |  | // 统一的机器人描述格式文件 | |
|  |  |  | ur\_robotiq\_env.xacro |
|  |  |  | // 包括UR5机械臂、robotiq手抓、kinectv1、机器人桌子和支架等 |
|  |  |  | ur\_robotiq\_env.urdf |
|  |  |  | // 命令行模式直接由ur\_robotiq\_env.xacro转换得到 |
|  |  |  | ur\_gripper\_env.xacro |
|  |  |  | // 包括UR5、气动手抓、kinectv1、机器人桌子和支架等 |
|  |  |  | ur\_gripper\_env.urdf |
|  |  |  | // 同上 |
|  |  |  | // 其余文件是被调用的文件 |
|  |  | world\_models | |
|  |  | // 环境模型文件，可以添加到gazebo中 | |
|  |  |  | models |
|  |  |  | // 模型文件 |
|  |  |  | worlds |
|  |  |  | // 世界文件 |
| 5. | p\_robot\_gazebo | | |
|  | // gazebo启动文件，格式参照了universal\_robot功能包 | | |
|  |  | controller | |
|  |  | // 自己配置的controller文件，也可以直接掉机器人功能包中的配置文件，如universal\_robot包中的 | |
|  |  | launch | |
|  |  | // 启动文件 | |
|  |  |  | ur\_pick\_gazebo.launch |
|  |  |  | // 启动仿真环境 |
|  |  |  | // 里面的调用关系自己看 |
| 6. | p\_robot\_learning\_demo | | |
|  | // 自己写的核心程序包，包括路径规划、动作规划、用于控制机械臂运动的action服务端和客户端、强化学习程序等。 | | |
|  |  | action | |
|  |  | // 自定义的action消息类型 | |
|  |  | include | |
|  |  | // 规划部分头文件 | |
|  |  |  | /p\_robot\_learning\_demo/action\_planing |
|  |  |  | // 动作规划文件，通过路径组合得到  // 注意：程序中可以设置机械臂的速度（static double speed = 2;）。仿真环境中速度可以设置的快一些，最大设置为2比较好，超过2规划的结果就不太好。实际环境速度设置的小一些，0.5左右。  // 函数pick\_obj用于仿真环境抓取，pick\_obj2用于实际环境抓取，两者手抓不一样，控制方式也不一样。  // 其他函数类推 |
|  |  |  | /p\_robot\_learning\_demo/action\_planing |
|  |  |  | // 路径规划文件，通过KDL库实现。 |
|  |  | launch | |
|  |  | // 启动的是robotiq的节点和add\_object节点 | |
|  |  | msg | |
|  |  | // action编译之后自动生成的消息文件 | |
|  |  | scripts | |
|  |  | // 强化学习部分程序都是用python写的，程序都在该文件夹下  // 强化学习算法基于pytorch和python3实现，因此这部分程序是运行在conda建立的虚拟环境中（虚拟环境在隐藏目录下，home文件夹下ctrl+h，虚拟环境在目录 .conda/envs/robotlab），如何实现参照本人博客：<https://blog.csdn.net/penge666/article/details/88692605>。 | |
|  |  |  | data\_rl |
|  |  |  | // 保存的神经网络参数 |
|  |  |  | p\_robot\_env.py |
|  |  |  | // 参照OpenAI Gym写的强化学习环境，主要是一些规范化的函数接口 |
|  |  |  | p\_robot\_env\_two.py |
|  |  |  | // 同上，环境中物体数量变了 |
|  |  |  | p\_robot\_rl.py |
|  |  |  | // 强化学习算法程序 |
|  |  | src | |
|  |  | // C++程序文件夹。包括基于action的服务端、客户端，在moveit中增加物体 | |
|  |  |  | p\_robot\_go\_server |
|  |  |  | // 机械臂控制服务端 |
|  |  |  | p\_robot\_go\_client |
|  |  |  | // 机械臂控制客户端 |
|  |  |  | // 其他类似，根据文件名类推 |
|  |  |  | add\_object |
|  |  |  | // 在moveit中的桌子上增加一个桌面 |
| 7． | p\_robot\_moveit\_config | | |
|  | // moveit!助手配置好后自动生成的配置包，需要注意的是controller.yaml文件是自己另外添加的 | | |
| 8． | p\_robot\_msggen | | |
|  | // 用matlab将ROS自定义的消息类型转化为matlab能用的消息类型  // 参照本人博客：<https://blog.csdn.net/penge666/article/details/79141215>。 | | |
| 9. | p\_robot\_real\_moveit | | |
|  | // 实际机器人环境moveit包 | | |
| 10. | robotiq\_85 | | |
|  | // robotiq手抓功能包，从github下载 | | |
| 11. | universal\_robot | | |
|  | // UR机械臂功能包，从github下载  // 其中ur\_kinematics中是机械臂的运动学程序，moveit配置时可以选里面的求解器，需要先编译 | | |
| 12. | ur\_kinect2\_calibration | | |
|  | // 手眼标定功能包  // 使用方法参照：<https://www.ncnynl.com/archives/201611/1117.html>。 | | |
| 13. | ur\_matlab\_simu | | |
|  | // matlab程序。主要在RobLearnGUI文件夹下，实现RGB-D图像交互示教、模仿学习预测、图像处理、物体识别等 | | |
|  |  | RobLearnGUI | |
|  |  | // 核心程序文件夹 | |
|  |  |  | data |
|  |  |  | // 数据文件夹。包括示教时保存的图像、示教数据等  // state\_action.mat为堆叠积木任务示教数据  // b\_state\_action.mat为pick and place任务示教数据  // 其他的带b的一个道理 |
|  |  |  | weights |
|  |  |  | // LSTM训练后得到的权值文件，用于模仿学习预测过程 |
|  |  |  | obj\_hsv\_detect.m |
|  |  |  | // 基于HSV颜色空间识别物体 |
|  |  |  | RobLearning.m |
|  |  |  | // GUI主程序  // 基于robotics工具箱与ROS通讯  // 实现了机械臂、robotiq action客户端  // 可以选择demonstrtaion和imitation模式，点一下按钮为进入该模式，再点一下为退出该模式  // 由于仿真环境和实际环境的不同，也需要选择simu或real模式，会有不同的相机到世界坐标系的变换矩阵及其他参数 |
|  |  |  | // GUI操作过程为  // 1.运行界面🡪2.点击setting，选择simu or real🡪3.点击client\_on，启动与ros连接🡪4.点击start，开始传输图像🡪5.点击demo，进入示教模式，示教时会记录数据。点击select，在图像中选中一个物体然后按enter，等下会计算得到三维坐标。点击action\_list中的一个动作，机械臂开始运动。再次点击demo，退出示教，不在记录数据。🡪6.点击imit，进入模仿模式，根据当前环境状态预测动作和目标物，点击go，机器人运动。每次都要点击一次imit，才能预测一次。 |
|  | | | |
| 如何使用  仿真环境：  roslaunch robot\_gazebo ur3e.launch  bash /home/wangzitu/ur3e/src/robot\_moveit\_config/launch/moveit\_start.sh  roslaunch p\_robot\_learning\_demo demo\_node.launch  rosrun p\_robot\_learning\_demo p\_robot\_go\_server  重开一个终端  5. condapy3 // 查看：<https://blog.csdn.net/penge666/article/details/88692605>  rosrun p\_robot\_learning\_demo p\_robotrl.py  source activate CondaRL  rosrun p\_robot\_learning\_demo GUI.py | | | |