【ROS机械臂入门教程】
第8讲 运动规划

小五

日期　2023/1/18

目录

为什么要路径规划

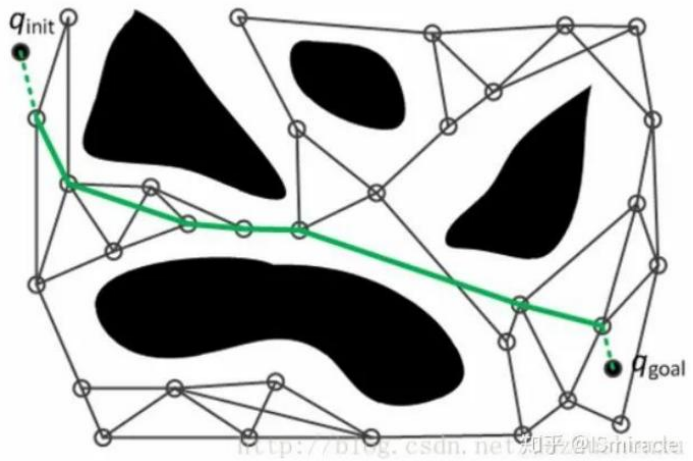
- **避障**：避免与桌子等机械臂附近的**静态物体**发生碰撞；避免与(突然走近的)人等**动态物体**发生碰撞
- **任务对运动的路径有要求**：具有运动学或动力学约束，如焊接、抓取装有水的杯子等

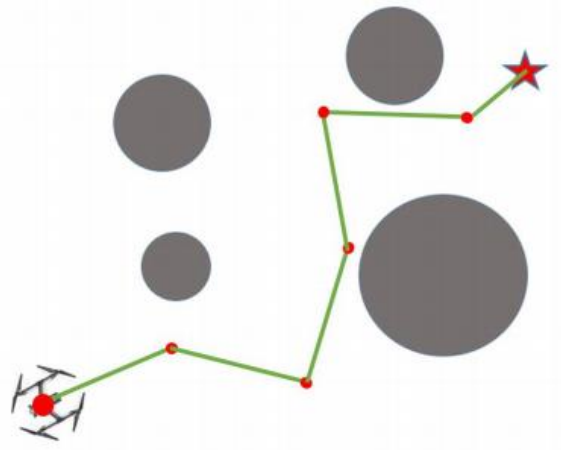
路径规划分类

- 基于搜索，Dijkstra，A*，Anytime A*、ARA*、D*
- **基于采样**，PRM，RRT，RRT-connect，RRT*，Kinodynamic-RRT*(符合动力学)，Anytime RRT*，Informed RRT*
- 智能算法如遗传算法、蚁群算法

基于采样的路径规划算法

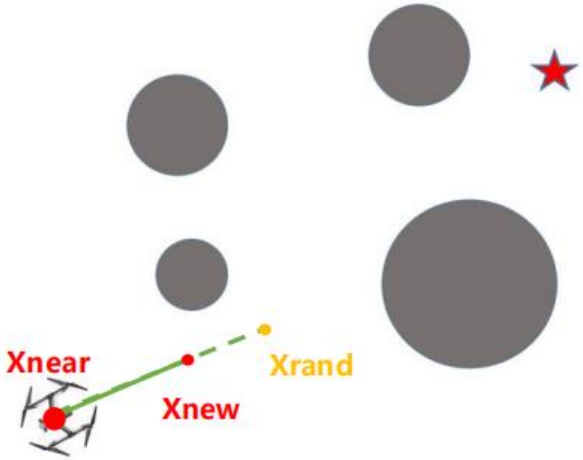
- 基础知识：概论完备：时间无穷大必能找到解；
- PRM：1.先随机采样n个点构成图 2.基于图进行搜索
- RRT：快速搜索随机树，**非最优解**；查询near可用kdtree等数据结构，快速找到树中最近结点
- RRT-connect：双向扩展的RRT，从start和goal同时扩展，搜索速度比较快，**ros-moveit默认算法**
- RRT*：作了两个改进，一是改变了新节点连接到树的规则，二是对搜索树进行“**剪枝**”操作，使之更接近于真实的最优路径

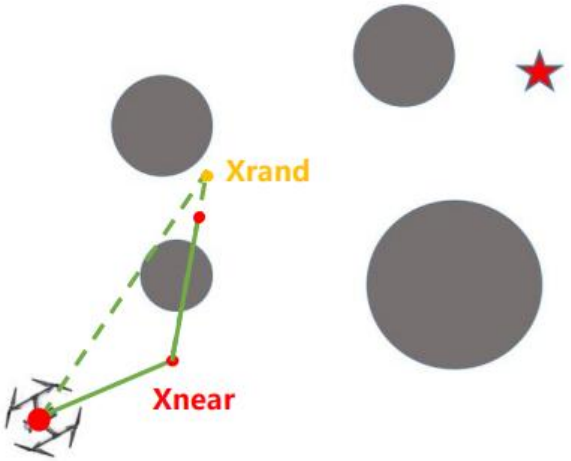


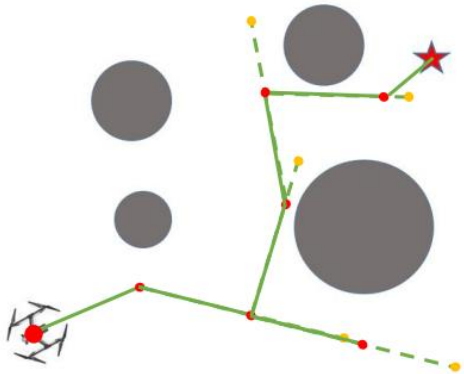


图片参考自《机器人路径规划、轨迹优化系列课程》

RRT算法







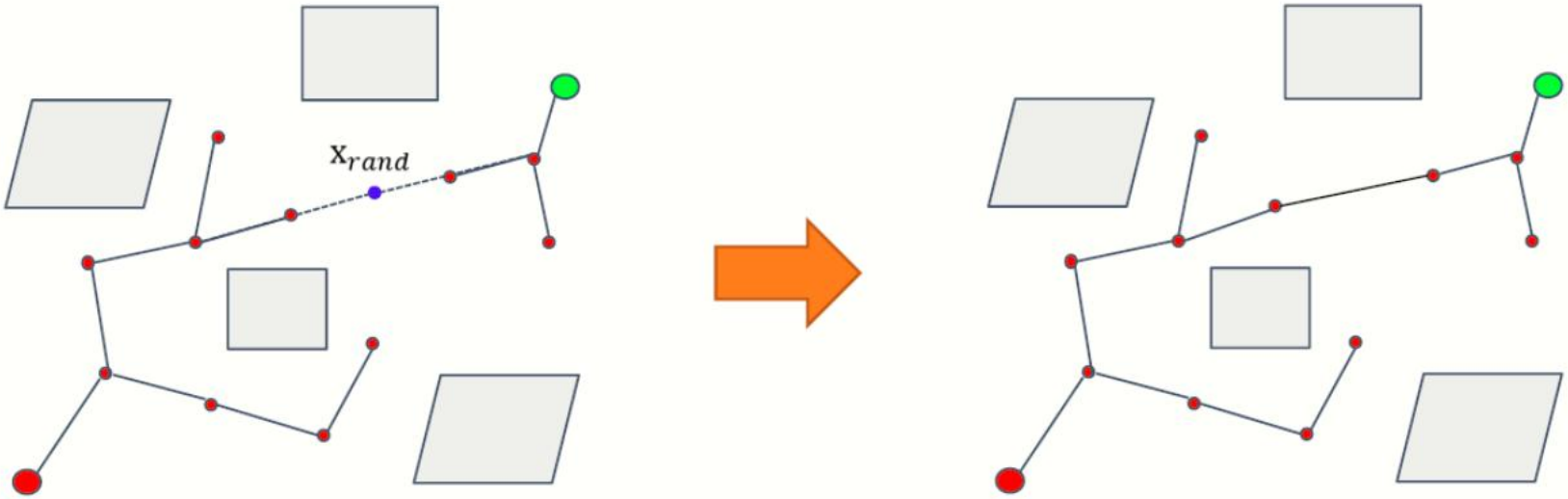
Algorithm 1: RRT Algorithm

Input: $\mathcal{M}$,x_{init},x_{goal}
Result: A path Γ from x_{init} to x_{goal}
$\mathcal{T}$.init();
for $i = 1$ *to* n **do**
  $x_{rand} \leftarrow Sample(\mathcal{M})$;
  $x_{near} \leftarrow Near(x_{rand}, \mathcal{T})$;
  $x_{new} \leftarrow Steer(x_{rand}, x_{near}, StepSize)$;
  $E_i \leftarrow Edge(x_{new}, x_{near})$;
  if *CollisionFree*$(\mathcal{M}, E_i)$ **then**
    $\mathcal{T}.addNode(x_{new})$;
    $\mathcal{T}.addEdge(E_i)$;
  if $x_{new} = x_{goal}$ **then**
    Success();

图片参考自《机器人路径规划、轨迹优化系列课程》

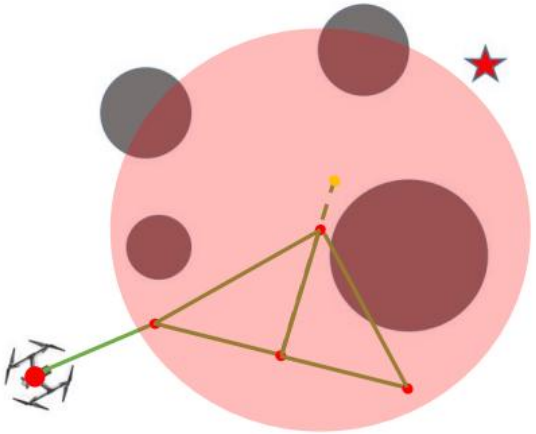
RRT-Connect算法

Bidirectional RRT / RRT Connect



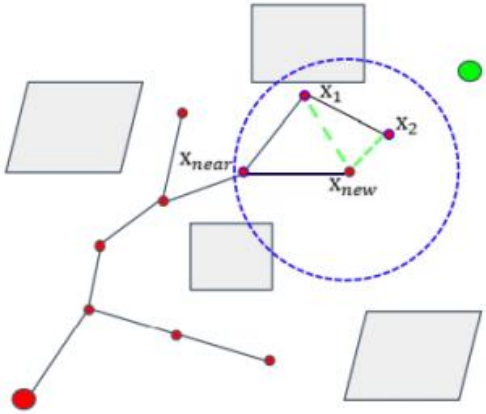
RRT*算法

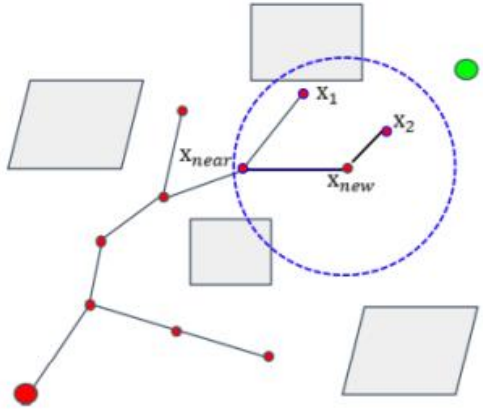
- RRT*：能找到一条最优的路径?
- 改进1：重新选择父节点
- 改进2：节点重新连接



当前节点重新选择父节点

范围内的节点重新连接（rewire）

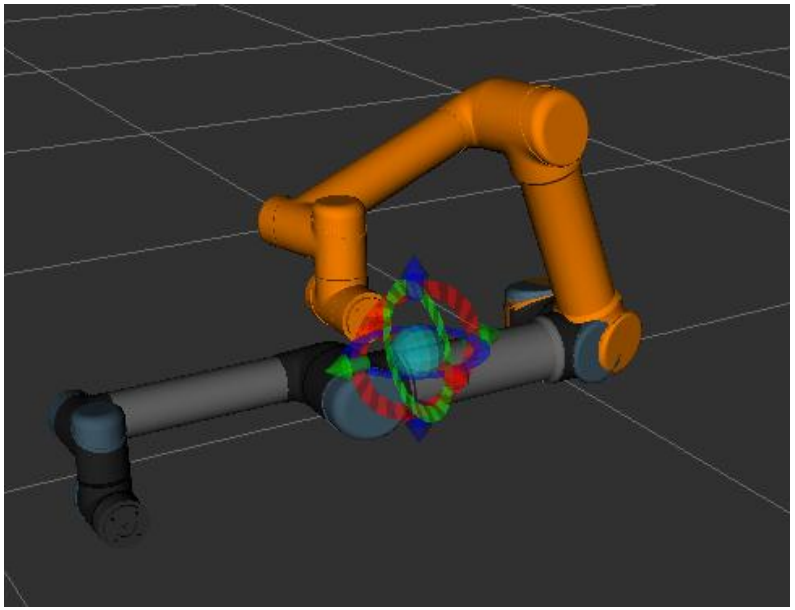


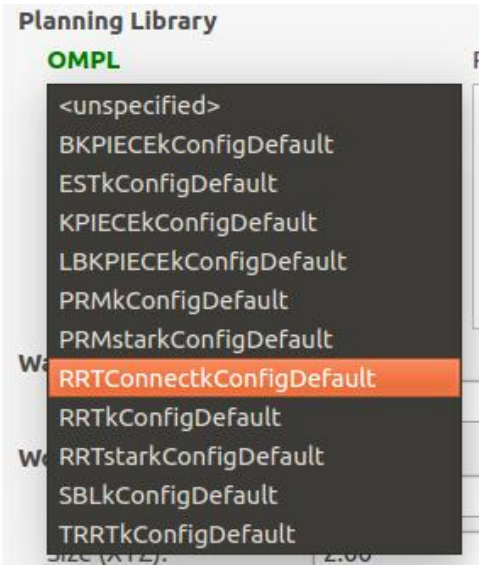


图片参考自《机器人路径规划、轨迹优化系列课程》

Moveit内置算法

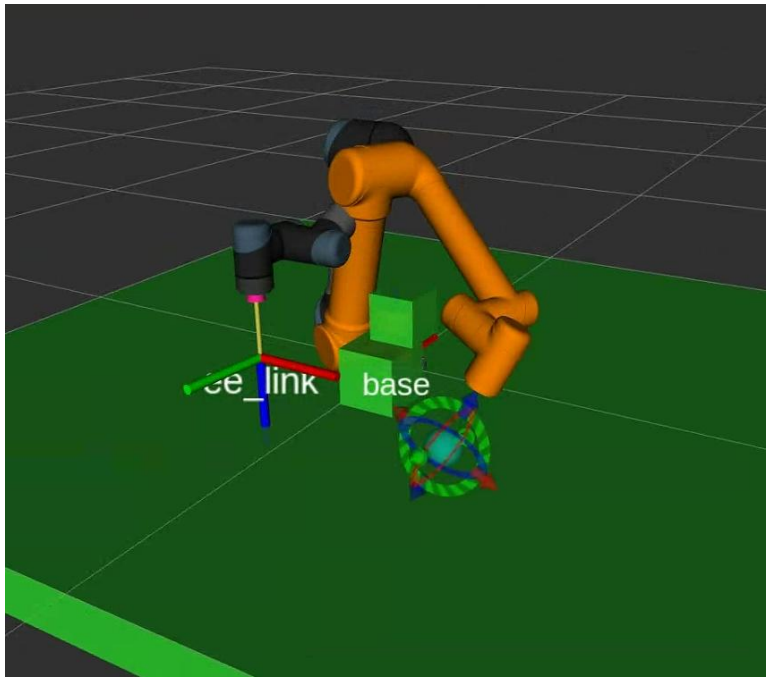
SBL	BFMT
EST	PDST
LBKPIECE	STRIDE
BKPIECE	BiTRRT
KPIECE	LBTRRT
RRT	BiEST
RRTConnect	ProjEST
RRTstar	LazyPRM
TRRT	LazyPRMstar
PRM	SPARS
PRMstar	SPARStwo
FMT	



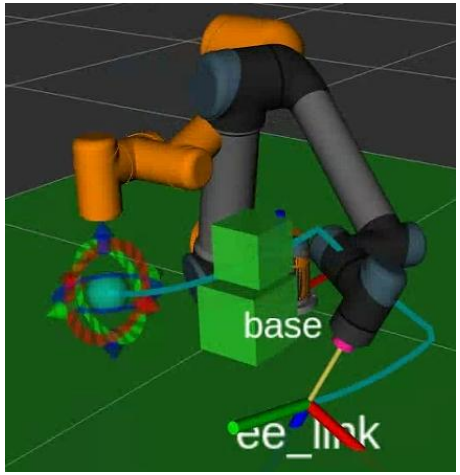


Moveit!默认使用OMPL库，且默认使用RRTConnect算法!

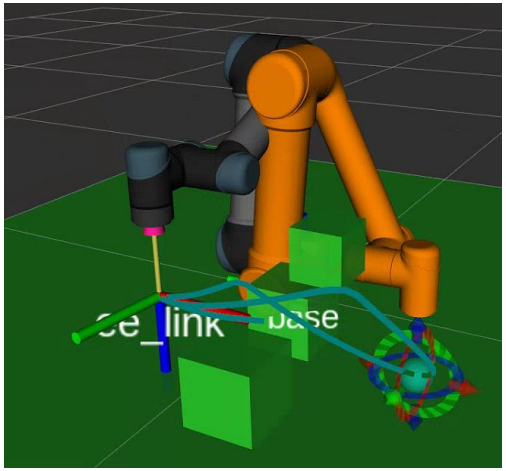
2 如何选择路径规划算法



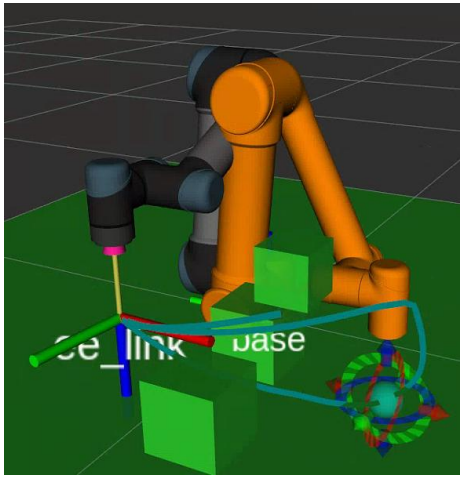
BKPIECEkConfigDefault	失败
ESTkConfigDefault	0.5s 还不错
KPIECEkConfigDefault	失败
LBKPIECEkConfigDefault	失败
PRMkConfigDefault	5s 较满意
PRMstarkConfigDefault	5s 满意
RRTConnectkConfigDefault	0.1s **最快** 不是特别好
RRTkConfigDefault	0.5-5s 还行
RRTstarkConfigDefault	**5s 满意**
SBLkConfigDefault	失败
TRRTkConfigDefault	**1-5s 满意**



EST



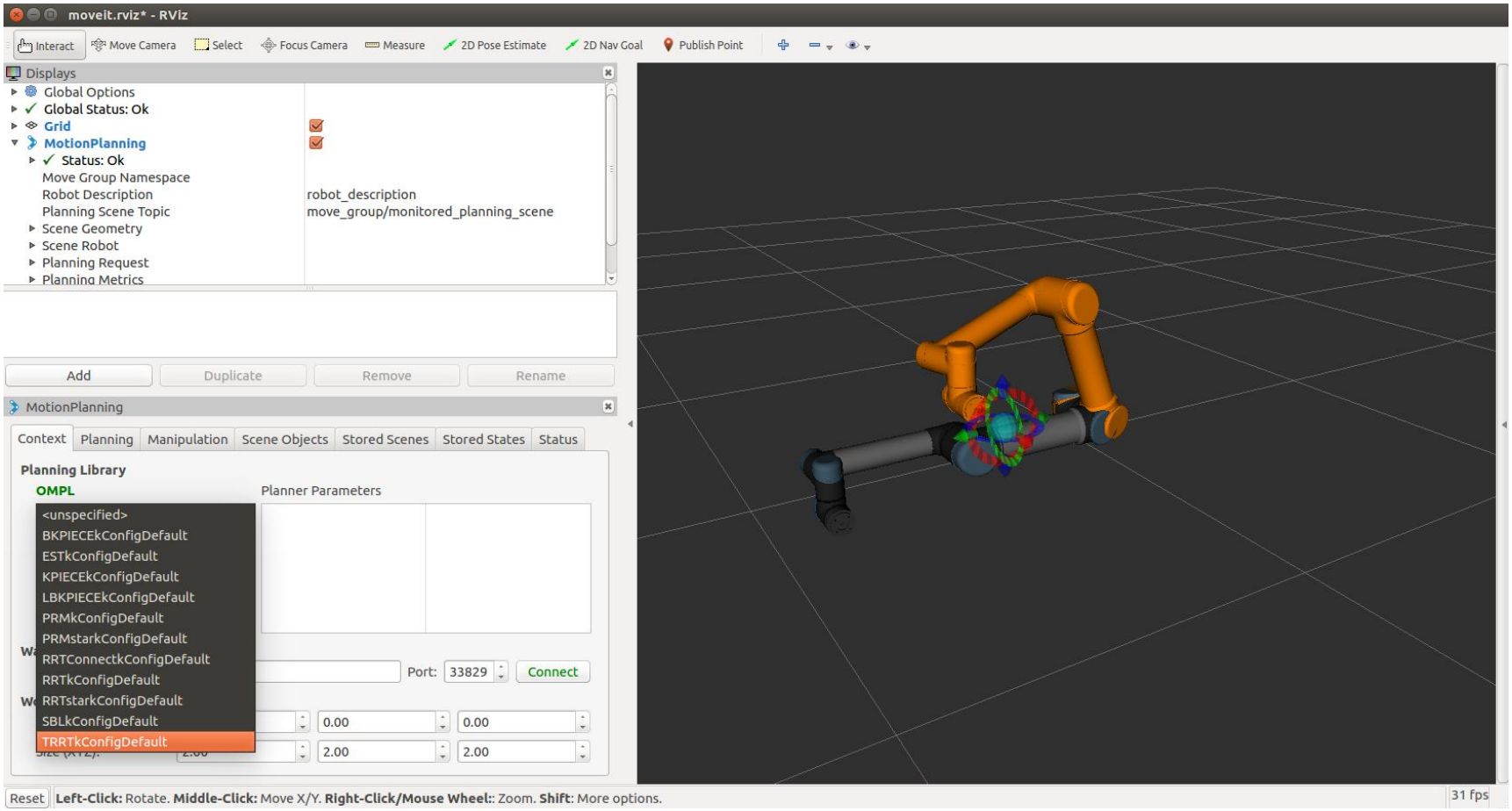
RRT*



TRRT

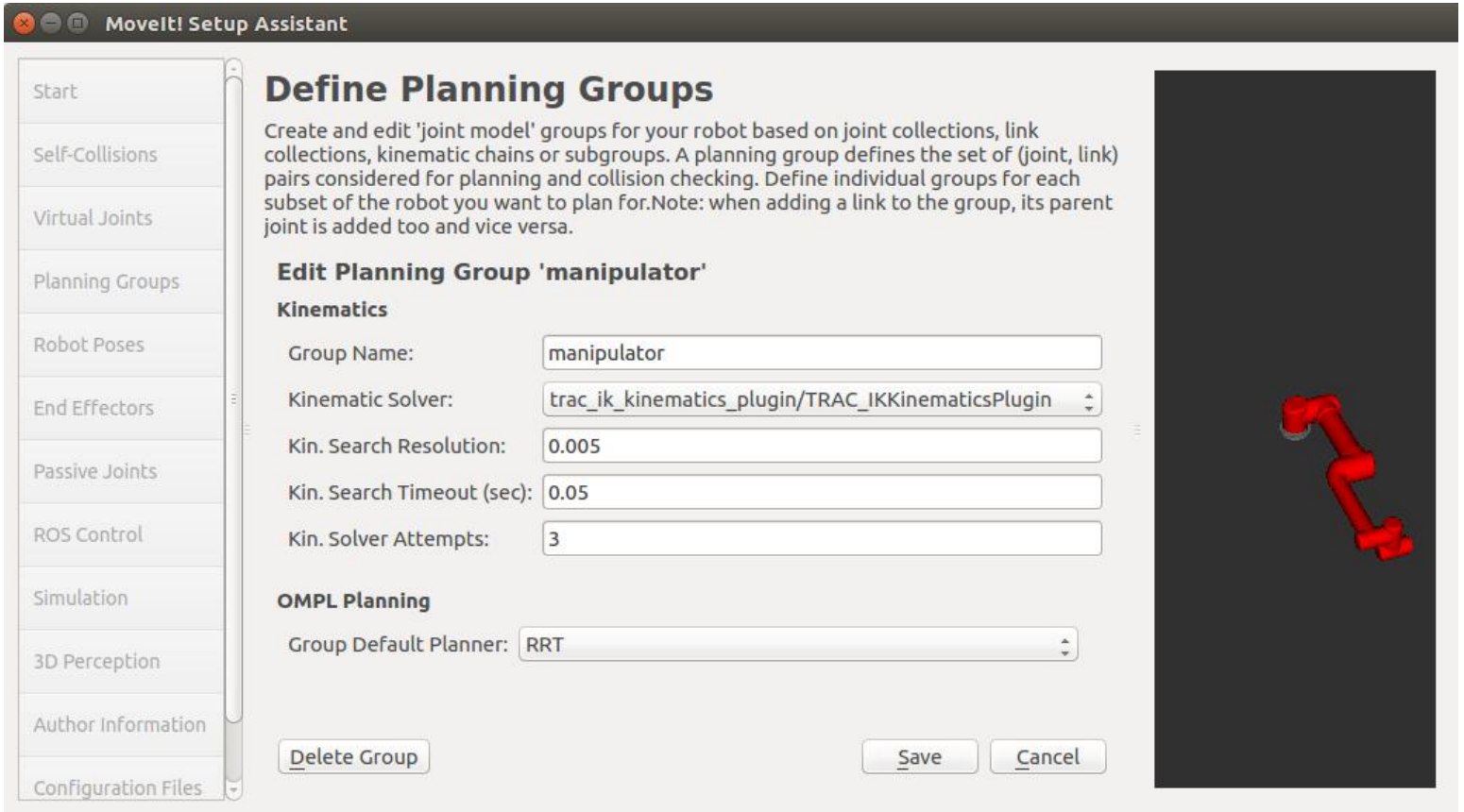
Moveit更换默认算法

- 1.打开Rviz，在Rviz中更换算法



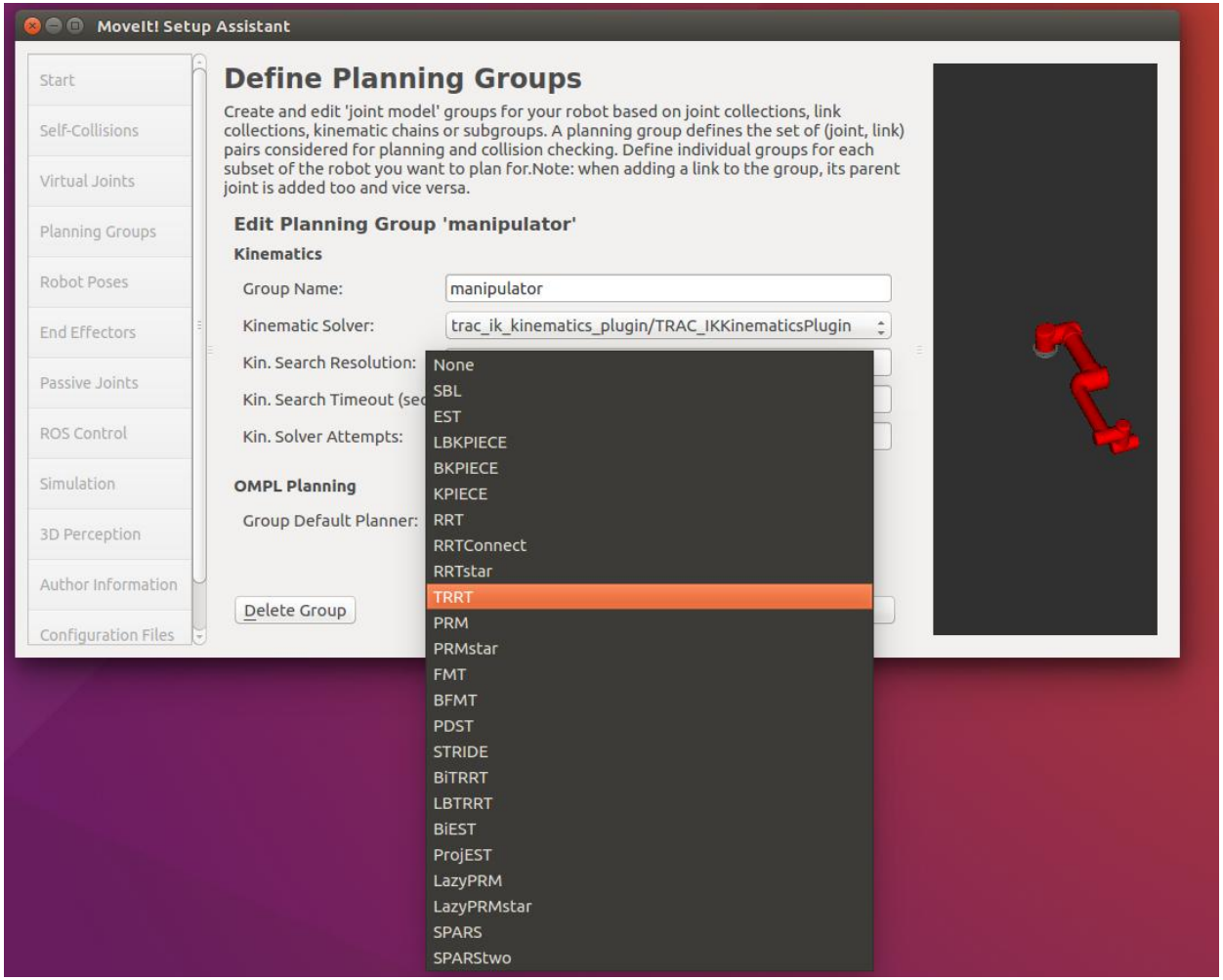
Moveit更换默认算法

- 2.在moveit_setup_assistant中更换



Moveit更换默认算法

- 2.在moveit_setup_assistant中更换

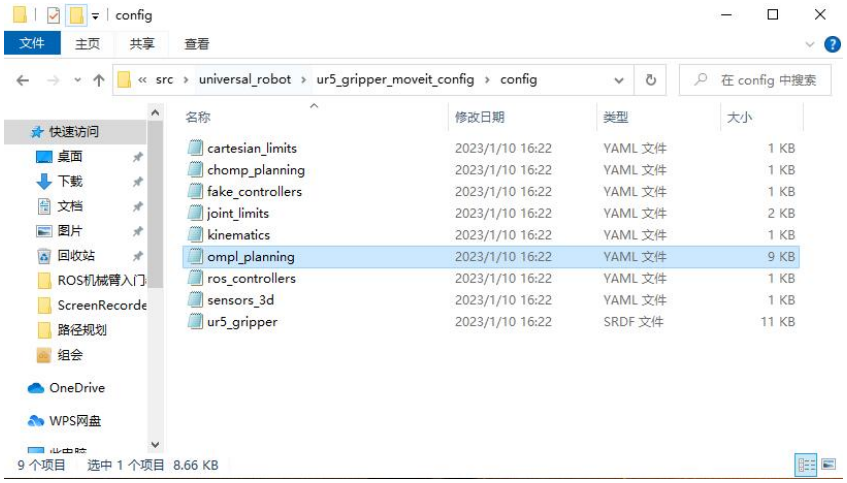


Moveit更换默认算法

- 3.在程序中调用相关接口更换
- group.set_planner_id(“RRT”)　　(以python为例)

```
#self.arm.set_planner_id("RRTConnect")
self.arm.set_planner_id("TRRT")
```

- 可更换的算法可以再xxx_moveit_config/config/ompl_planning.yaml中查看



Moveit自定义算法

- ompl自定义算法

参考资料：https://blog.csdn.net/sinat_23853639/article/details/87854461
参考资料：https://blog.csdn.net/weixin_36965307/article/details/105312020

- 尝试使用chomp库和Stomp库

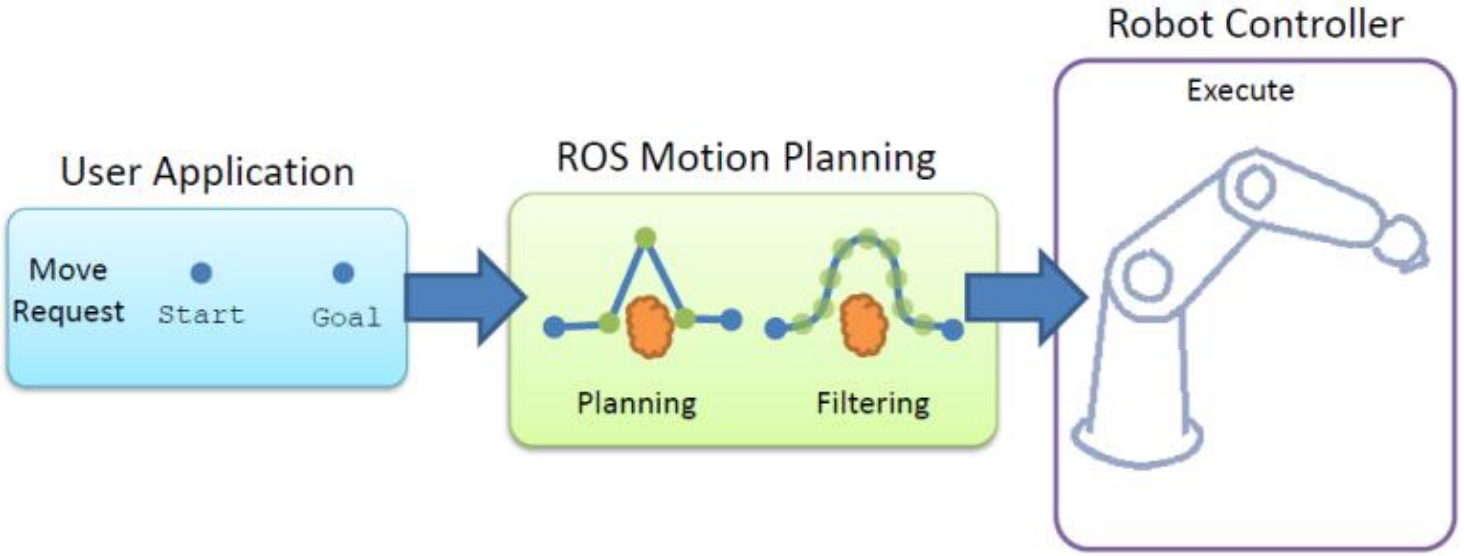
参考资料：https://www.guyuehome.com/33896

大家多多探索！~

教程视频会持续更新

敬请期待！

路径规划知识补充



路径规划 -> 轨迹规划 -> 电机执行