

# 整身3D避障技术点

## 关键技术并解决什么问题

### • 解决移动机械臂3D实时可靠（自我）的动态避障问题

- 3D。增量式构建3D ESDF地图，基于体素哈希与广度优先搜索方法来增量式更新环境障碍信息，实现动态环境中机器人周围障碍环境的高频率感知。
  - 指标：在机器人本体2m范围内增量更新，以3cm的栅格大小可以达到30hz，满足动态环境下的操作需求。
  - 对比1：高老师目前无人机的框架，以elastic tracker和fast planner为例，所采用的是非增量式的ESDF构建方式，每次都需要从空白开始算，这种好处是算法实现简单，但是是一个 $O(N^3)$ 时间复杂度的算法，在无人机的应用中，采用10cm左右的栅格即可快速构建，但是在复杂环境中移动机械臂操作场景，此栅格大小无法满足需要。而我们采用的增量式方法在3cm栅格大小下达到30hz的更新频率。
  - 为什么高老师不用FIESTA这种增量式更新的方法呢，我认为主要是因为，1、算法本身实现难，虽然开源了但是开源部分的代码质量不高 2、算法实现不完整，只有核心部分，如，核心部分建立出来的是EDF，不带有符号信息，在障碍物里面的距离都是0，不具有梯度信息，再如，不具有保存和加载地图模块。
  - 对比2：Moveit所用的框架的感知模块。moveit的感知方法是基于点云转成OctoMap，然后将OctoMap和机械臂模型都输入FCL中，再进行环境凸分解，利用FCL进行碰撞检测。所用方法，碰撞检测的方式繁琐，且这个常规流程不具有梯度信息，无法进行优化。在moveit的基础上实现的优化方法CHOMP或Trajopt依然得利用其他步骤获取梯度信息。整个感知的pipeline频率低。
  - 对比3：ETH的perceptive MPC所用的voxblox也是增量式的建图，但是无论是时间还是空间上都落后于我们所用的FIESTA。所以在后来22年他们的工作中也切换到了FIESTA，但是我倾向于他们也并未解决FIESTA中的一些问题，仅仅是拿来用（如符号问题->无法获取障碍物内部梯度信息，局部更新问题->无法获得大范围内的梯度信息）。
- 实时。基于序列二次规划（SLQ）的微分动态规划方法（DDP），求解连续时间内的滚动时域优化问题，获得可靠的前馈+反馈MPC控制率。关键是基于哈密顿-雅克比-拉格朗日方程来推导最优控制的更新率，结合自适应线搜索来获取最优局部控制率。
  - 指标：状态空间20维，可以实现50Hz的滚动时域优化，实现复杂环境下的实时（重）规划。
  - 难点1：相比于直接转为纯数值优化的方法来求解MPC问题（利用shooting或者collocation法转为非线性优化问题），基于DDP方法的MPC求解具有速度快，并且具有反馈控制率的优点。但传统DDP方法无法很好地处理带约束问题（如LQR只能无约束），因此如何推导可以处理约束的DDP问题是当今的一大难点。具体而言，我们利用松弛障碍函数来处理状态约束问题，利用约束投影和增广拉格朗日乘子来解决状态和输入联合约束问题，使我们的DDP求解器可以通用于复杂环境下的规划问题。
  - 难点2：传统的DDP方法求解大概分为backward后向求解再一个forward rollout的过程（参考ILQR），但是这个过程是串行的，当时间窗口长和离散精度高时，这个过程限制了整个

DDP问题的求解速度。我们对耗时的riccati-based backward过程进行子系统的划分，引入前后不同时刻的状态联系来解决子系统之间的连接问题，实现多线程并行的Riccati方程的求解。

- 对比：ETH就是用的这一套。我们和ETH的区别大概只是问题的formulation不一样。
- 可靠。利用滚动时域优化和系统模型来预测未来移动机械臂未来的全状态，利用预测模型进行机器人与环境的碰撞检测，基于碰撞模型和ESDF构建关节空间的拓扑地图引导多个MPC的并行优化，并选取最优的结果作为执行。（这是之后的工作计划）
  - 指标：对于低动态障碍物0.1m/s实现可靠的实时重规划。
  - 对比：这个可以直接对标甚至超越ETH的效果，ETH的perceptive MPC和collision-free MPC中都是不存在重规划模块的，因此MPC只要陷入局部最优了比较难出来，任务直接会失败。这种重规划的工作是高老师的强项。
  - 难点：将无人机的重规划方法用于机械臂的属于升维问题，难点主要集中在1、无人机是质点，而移动机械臂是一个强非凸的 2、维度更大，在关节空间构建拓扑地图难度指数级增大。 3、无人机由于微分平坦的良好特性，整个优化问题可以在几毫秒就解决，而移动机械臂不行
- 避障。利用自动圆球近似算法对非凸的移动机械臂进行高精度的圆球近似，在制定一个近似误差的前提下，算法可以直接输出近似的结果。
  - 难点：速度与精度的权衡。从前端与后端两部分进行设计，保持与障碍物的大距离 clearance。
  - 对比1：目前业界论文中的避障算法再感知理想的情况可以达到很好的效果，但是感知是实际部署中不可忽略的问题，一来获取环境全局障碍信息需要较大的算力，二来现实情况下普遍存在部分遮挡问题。我们的方法基于ESDF方法能达到的移动机械臂的避障效果是业界最好的，虽然ETH的框架与我们的相似，但是在帧率、系统可靠性上都不如我们。
  - 对比2：moveit有实现一部分基于基于凸包的避障，但是速度慢，因为那个对环境的处理更加复杂
  - 对比3：相比于业界大部分基于reactive controller的方法，我们的方法具有预测和重规划能力，因此能够提前采取避障措施，更为可靠
- 解决移动机械臂整身平滑运动的最优控制问题
  - 整身。将移动机械臂问题构建成一个整身MPC问题，根据实时的运动误差自适应调整MPC的权重系数以实现合理的运动分配。
    - 效果：作业效率提升，实现 无停式 的整身运动规划。
    - 难点：整身规划相比与分体规划的问题更加复杂，规划空间更大，但是也因此具有了更强大的移动机械臂操作能力，能够在更加受限的环境运动。但是如何保持运动的协调，如何避免两者的碰撞，如何避免机械臂陷入奇异构型都给整身规划问题提出的难点。
    - 对比：
  - 平滑。在框架的前端与后端进行kinodynamic的规划和优化，构建加速度层面的MPC使机械臂的运动平滑
    - 效果：运动的jerk大大降低，更为平滑
    - 对比：目前ETH的狗和平台都未在加速度层面进行规划，仅仅是考虑到速度层面，一是为了不让模型变得更加复杂，二是不规划也能完成任务，只是效果没那么顺滑而已（jerk大）。但是实际在运动中必须要考虑这个问题，比如我们的平台，假如仅仅在速度层面进行规划那么就会很 冲。

## 核心技术点

- 大范围高分辨率ESDF地图实时增量化更新

- 轻量化前端次优轨迹生成
  - 应该可以做到几十毫秒级别，未测试
- 基于高精度凸分解的机器人本体点云滤波
  - 指标：对于realsense 30hz的输出，可以实时滤除点云
  - 方法：利用V-HACD法对移动机械臂进行精确凸包分解（输入URDF，输出凸包mesh），再根据凸包的快速包含检测算法实时滤除机械臂本体点云。
- 并行化微分动态规划方程的求解
  - 具有反馈控制率，无论平台性能高低，可保证在一个低算力平台中可靠控制一个机器人。
- 基于增广拉格朗日乘子法和约束投影的约束处理
- 函数的自动微分与梯度的求解
  - 最优化问题求解的快速线性化
  - 优化过程的局部梯度

## 比较

- 与开源框架Moveit比较
  - 速度有较大的提升，我们的框架更加简洁高效
- 与ETH比较
  - 在系统可靠性方面可以超越，对于两轮平台的整身操作效率也可以超越
  - ETH在感知部分和前端模块做的工作较少且效果不是很理想，这方面也会超越
  - ETH的底层优化框架跟我们的一样，但是我们对于问题的formulation更符合实际移动机械臂场景的需求。