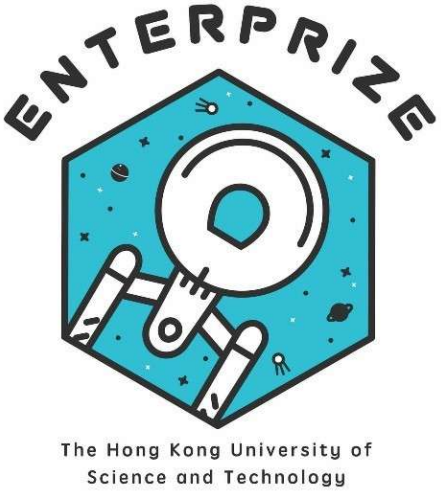
香港科技大学 ENTERPRIZE 战队
Robomaster 2026 赛季工程装配算法开源报告

ENTERPRIZE 战队工程组

2026 年 6 月 23 日

摘要

本报告面向香港科技大学 ENTERPRIZE 战队 RoboMaster 2026 赛季工程机器人装配任务，详细阐述一套上位机侧辅助装配算法。该项目的出发点是在保留下位机基本的操控功能的情况下，协助操作手进行自动化的精细装配操作。报告详细分析了合成数据生成、神经网络训练、位姿估计、轨迹规划、任务状态机和系统设计。该方案已部署到 ENTERPRIZE 战队 Robomaster 2026 赛季工程机器人上并经过赛场验证，现已开源至 项目仓库，供社区参考。



目录

1 问题定义与技术路线

1.1 任务动机、项目定位与问题抽象

RoboMaster 2026 工程机器人需要在比赛环境中完成对科技核心完成能量单元的装配，该类任务要求操作手操控机械臂完成接近、插入、滑移、翻转等操作。在规则理解和试验中，我们观察到：如果完全依赖操作手通过自定义控制器手动完成整段流程，操作负担会明显增加，尤其是在翻转步骤中，操作手需要在关注装配站相对位姿、机械臂姿态、关节限位和碰撞风险的同时进行高精度的遥操动作。因此，引入上位机辅助算法的主要动机，是把其中一部分可自动化的步骤交给算法处理，降低操作手在复杂步骤中的瞬时负担。由于相关规则首年引入，队伍缺少可直接复用的历史开发经验，同时工程机器人本体还需要完成下位机主线功能，实机调试时间也较有限，所以我们将该算法项目定位为独立于下位机控制链路的增量辅助项目，下位机仍负责工程机器人主要操控逻辑，上位机侧则在前者的基础和已有硬件与控制接口之上，提供视觉估计和在线规划能力。

算法端可总体建模为如下问题：

$$\mathrm{F} : (\mathbf{I}_{\mathrm{rgb}},\ \mathbf{q}_{\mathrm{cur}}) \mapsto (\mathbf{T}_{b,E},\ \mathbf{q}_{\mathrm{traj}}(t)) , \tag{1.1}$$

其中 $\mathbf{I}_{\mathrm{rgb}}$ 为 RGB 图像，即由红、绿、蓝三通道构成的普通彩色图像；$\mathbf{q}_{\mathrm{cur}} \in \mathbb{R}^n$ 为当前机械臂关节状态。SE(3) 表示三维刚体位姿空间，包含三维平移和三维旋转。$\mathbf{T}_{b,E} \in \mathrm{SE}(3)$ 表示当前使用的初始装配基准系 E 在机械臂基座坐标系 b 下的位姿，$\mathbf{q}_{\mathrm{traj}}(t)$ 为可执行的关节轨迹。坐标系标签与变换记号见表 1。

算法的技术难点来自感知误差、操作精度和规划约束之间的耦合。视觉端需要在灯条变化、高亮反光和局部遮挡条件下稳定估计装配站位姿；操作端需要在结构紧凑的装配站内完成插入、滑移和翻转等动作，因此厘米级平移误差和小角度姿态误差都可能被放大为执行失败；规划端则需要在任务几何约束之外同时考虑关节限位和碰撞安全。由于实机调试成本较高，仿真平台也需要作为算法迭代的基础设施同步建设，用于在不占用真实机器人主线开发时间的情况下验证坐标系、规划约束和恢复流程。

1.2 总体技术路线

本项目采用自底向上的分层技术路线。感知侧使用 “RGB 图像、关键点、PnP” 的流程：网络负责在图像中找结构点，PnP 负责根据二维/三维点对应关系恢复六自由度位姿。

数据侧采用 “Blender 目标素材、真实背景合成、网络训练” 的合成数据路线。Blender 是三维建模和渲染工具，用于生成具有几何标注的目标素材。由于规则第一年发布，在我们目前的方案中，合成数据承担了主要数据来源角色。

规划侧将运动问题拆分为 I 型、II 型和 III 型。I 型表示下位机预置固定动作，II 型表示在线点到点规划，III 型表示沿任务约束流形搜索整段可执行轨迹。这样可以用固定

轨迹播放处理重复动作，用点到点规划处理普通避障，用任务约束规划处理装配过程中的受约束运动。

系统侧将上位机主节点层和交互环境层分开，使同一套感知、规划和任务编排逻辑可以接入真实环境或 MuJoCo 仿真环境。仿真环境用于支持并行开发和低成本试错，同时将真实场景评估作为算法可行性的最终依据。

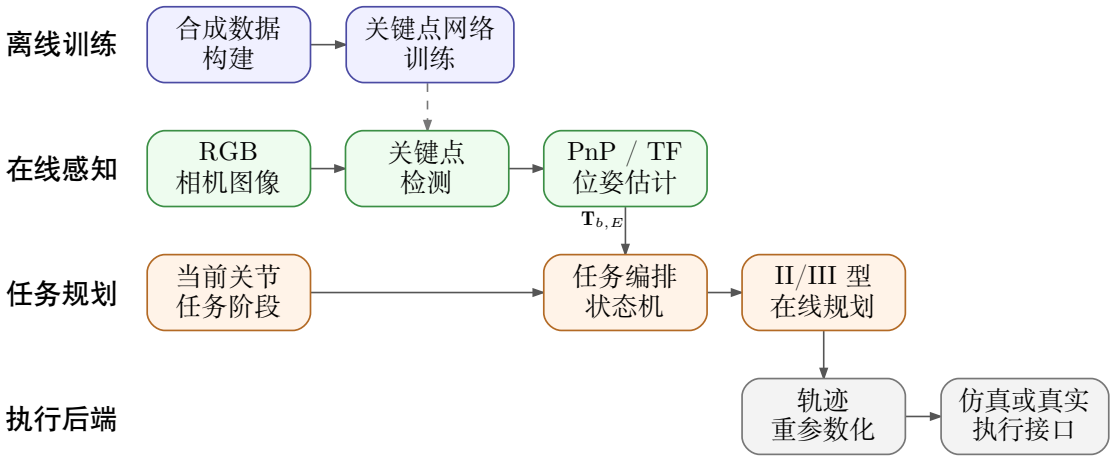


图 1: 总体技术路线流程框图。合成数据用于训练关键点检测器；在线阶段由 RGB 图像恢复装配基准位姿，再由任务编排层调用规划器并交给仿真或真实执行接口。

2 视觉识别与位姿估计

2.1 RGB 关键点路线与点位定义

装配站位姿估计可以直接采用 6D 物体位姿估计算法。例如 FoundationPose [20] 能够对新物体进行 6D 位姿估计与跟踪，在通用物体位姿任务上具有很强能力。然而，这类方法通常需要比单张 RGB 图像更丰富的输入，例如深度图像（Depth）、目标物体分割掩码（Mask）和物体 3D CAD 模型等。对于工程机器人装配任务，这些输入会带来额外的传感器、标定、同步和模型训练以及维护成本。

因此，我们采用了更轻量化的纯 RGB 感知方案，不额外引入激光雷达、深度相机或主动光传感器。装配站的可观测线索主要来自灯条、外壳边缘和局部几何结构，普通彩色图像已经包含这些外观信息。对应到算法上，我们采用了 RGB 关键点 PnP 路线：网络只负责定位二维结构点，PnP 和重投影误差负责估计位姿。这样可以降低硬件安装、外参标定、同步和点云处理链路的复杂度，同时保留较明确的几何解释和调试入口。

关键点提取可以由传统几何视觉完成，也可以由神经网络完成。传统方法通常依赖颜色或亮度阈值分割灯条区域，再提取轮廓、端点或中心线并与已知结构匹配。在灯条过曝、装配柱侧面灯条干扰、局部遮挡和透视变化下，手工阈值和局部轮廓质量容易成为不稳定因素，而神经网络关键点方案将目标检测和结构点定位交给数据驱动模型，则可以隐式地学会在这些不同环境条件下的结构点特征分布。具体来说，感知模块输入 RGB

图像 $\mathbf{I}_{\mathrm{rgb}}$，输出科技核心上 $K = 12$ 个关键点的二维观测：

$$\mathrm{D}(\mathbf{I}_{\mathrm{rgb}}) = \{(\mathbf{u}_i, c_i, v_i)\}_{i=1}^{12}, \tag{2.1}$$

其中 $\mathbf{u}_i \in \mathbb{R}^2$ 是第 i 个关键点坐标，c_i 为置信度，v_i 为可见性或有效性标记。三维关键点集合记为 $\mathcal{X}_E = \{\mathbf{X}_i^E\}_{i=1}^{12}$，其中 $\mathbf{X}_i^E$ 表示第 i 个关键点在装配基准系 E 下的位置。

为了保证预测稳定，除了视觉识别特征 MARKER 的四个角点，我们还让模型额外学会预测分散布置在灯条端点、外壳边缘和装配柱附近的稳定结构关键点。这样不仅可以为 PnP 提供更长的空间基线，在局部结构被遮挡或灯条过曝时，其他区域仍可能保留可用点。设可见关键点集合为 $\mathcal{V} \subseteq \{1, \ldots, 12\}$，当可见点数量足够且覆盖目标不同区域时，PnP 求解对单个局部点误差的敏感性会降低。

2.2 关键点识别网络与 PnP 位姿估计

关键点检测可以借用姿态估计中的 *bottom-up* 与 *top-down* 两类表述。*bottom-up* 方法通常先在整幅图像上预测关键点或局部部件，再进行实例关联 [4]；*top-down* 方法则先检测目标实例框，再在裁剪后的局部区域中估计关键点 [21]。在装配站识别任务中，*bottom-up* 方案对应整图直接预测：一个模型同时输出目标框和关键点，YOLOPose 即属于这一类整图预测思路 [12]，链路短、部署简单，适合作为后备方案和对照。*top-down* 方案则先检测装配站目标框 $\mathbf{b}$，根据目标框裁剪并归一化局部图像，再由关键点网络在局部图像中估计关键点。推理结束后，关键点统一转换回原图坐标，记为 $\mathbf{u}_i$，后续 PnP 直接使用这些二维观测。

top-down 方案的优势主要来自于检测精度，在检测器稳定的前提下，局部关键点网络可以更专注于装配站结构特征，减少场地背景、车体结构和其他高亮区域对关键点预测的干扰。装配站位姿识别主要用于接近规划和插入后的任务基准重建，目标与机械臂基做坐标系相对运动大多静止，对端到端帧率的要求低于高动态视觉任务。基于这一特点，我们以 *top-down* 方案作为主力方案，以 *bottom-up* 方案作为后备方案和对照。图 2 给出了两类流程的接口关系。

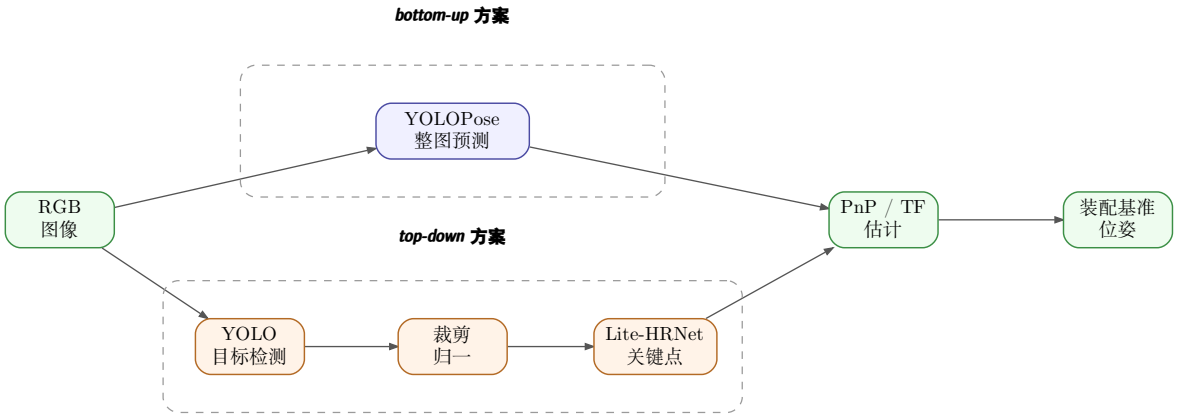


图 2: *bottom-up* 与 *top-down* 关键点识别流程对比。

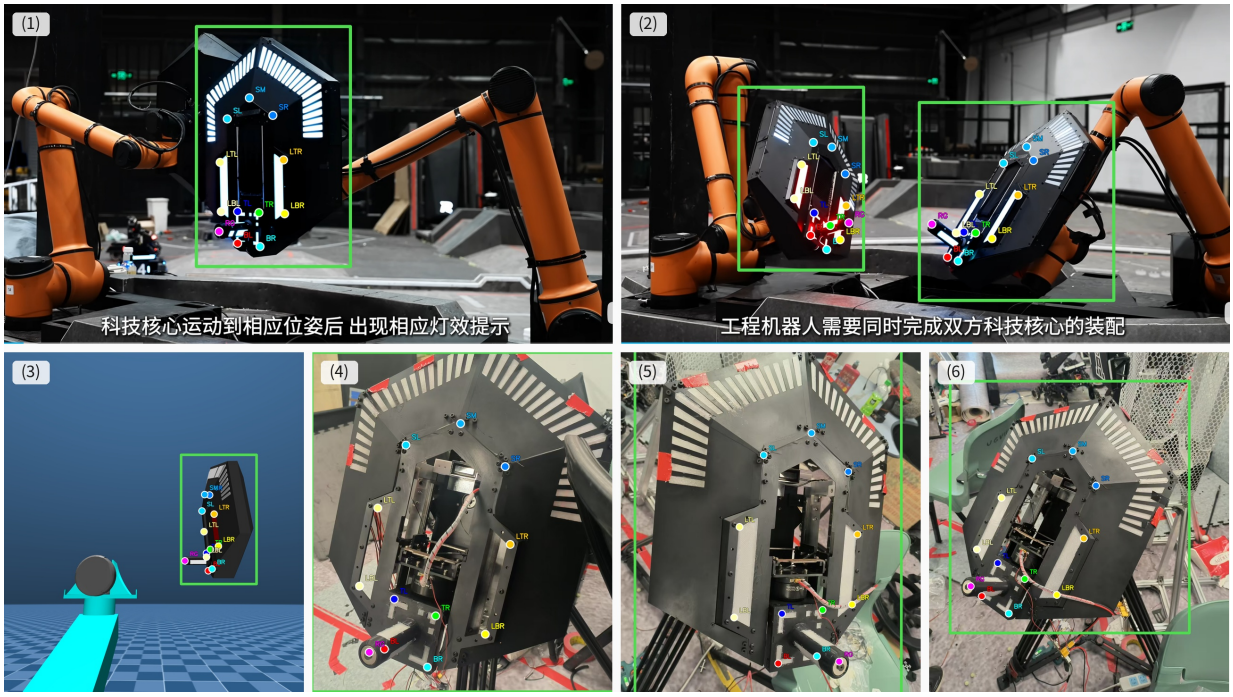


图 3: *top-down* 关键点模型推理结果示例，覆盖官方比赛场地（1）（2）、MuJoCo 仿真场景（3）以及本战队自制科技核心场景（4）（5）（6）。后续 PnP 使用这些二维关键点估计装配基准位姿。

当感知网络给出图像空间中关键点预测之后，我们用 PnP 算法来估计位姿。具体来说，我们首先使用 RANSAC/SQPnP [5, 17, 2] 进行初始位姿筛选与外点过滤，再以筛选后的内点集合为基础，用 Levenberg–Marquardt 方法最小化重投影误差进行精修，最终通过 TF 链路将 $\mathbf{T}_{c,E}$ 转换为规划侧使用的 $\mathbf{T}_{b,E}$。

3　合成数据与训练数据构建

上述视觉路线把学习问题集中在目标检测和关键点定位上。由于规则第一年发布，科技核心的真实图像和标注数据样本数都不足。若完全依赖真实采集，一方面难以覆盖足够多的目标姿态、光照、遮挡和灯条状态，另一方面手动获取真实六自由度位姿和关键点标注的成本也较高。合成数据的优势是可以自动获得目标框、关键点、可见性和相机参数等监督信息，并快速覆盖新规则下目标对象更多可能的外观与几何状态。

受时间和人力分配限制，当前训练主线采用纯合成数据，尚未进行真实数据微调。尽管如此，实践表明，通过数据增强等训练技巧，纯合成方案仍然可以满足训练需求，并泛化到真实场景中（见图 3）。实际运用中，我们也手工标注了少量真实数据，用于独立验证和失效案例分析。

图 4: 合成训练集样本示例。样本由渲染目标素材与真实比赛背景合成得到，用于覆盖不同视角、光照和灯条状态。

3.1 目标素材与几何标注

我们参考规则手册，在 Blender 场景中尽可能一比一还原科技核心模型，建模装配柱、MARKER 视觉特征灯条、主侧灯条和外壳主体几何结构。为了更加贴近真实场景，我们使用 Principal BSDF 节点对仿真科技核心进行了材质临摹。

在实际素材生成阶段，每个合成样本首先确定 Blender 相机的视角、距离和内参，并由相机位姿得到目标相对于相机的刚体变换。随后将装配基准系 E 下的三维关键点投影到图像平面，得到二维关键点标注，并同时用光线追踪、目标可视区域和图像边界判断关键点是否可见。最后，我们剔除可见关键点数过少的坏样本。完成几何标注后，渲染阶段输出透明背景目标素材，并同步输出对应的关键点投影和可见性标注。

3.2 渲染随机化

我们在渲染目标素材时引入随机化策略。随机化主要覆盖相机视角、距离、分辨率、灯条颜色、灯条强度、开关状态、环境光、辅助光、曝光和色彩变化。样本过滤则依据目标姿态、目标框尺寸和可见关键点数量完成，避免生成对训练无益或标注语义不明确的样本。训练集需要覆盖会影响关键点观测的外观变化。对于装配站任务，灯条亮灭、局部过曝和视角变化往往比精细材质更影响识别稳定性，因此随机化重点放在这些因素上。

3.3 真实背景合成与亮度匹配

为了降低纯 3D 渲染背景与真实比赛图像之间的域差，我们参考 DeepArUco++ 中的合成数据构建思路：该工作将合成标记叠加到真实背景上，并利用背景的亮度分量调制前景标记，使合成目标的光照更接近背景环境 [1]。如图 4 所示，Blender 渲染得到的是透明背景科技核心素材，背景则来自官方历年比赛图片、直播画面和场地相关图像，以尽力降低合成数据与真实场地数据的域差。合成时基于亮度进行背景与素材的匹配，选

择与素材亮度更接近的背景区域，尽量减少简单贴图带来的突兀边界。

亮度分量可近似理解为彩色图像转换为灰度后得到的明暗信息。若直接把高亮目标贴入暗背景，或者把暗目标贴入强光区域，模型容易学到不稳定的边界伪特征。亮度匹配可以让素材与背景局部统计更接近，降低合成痕迹。合成样本时需要同步传播目标框、关键点和可见性标注。

3.4 训练阶段数据增强

渲染随机化和真实背景合成发生在样本生成阶段，训练时仍需要进一步进行图像级数据增强。纯合成数据虽然可以提供精确标注，但材质细节、相机响应、压缩噪声、运动模糊、局部遮挡和现场曝光分布仍难以完全覆盖真实比赛环境。因此，纯合成训练对增强强度和覆盖面的依赖更高，需要通过更丰富的图像扰动扩大外观变化范围，降低模型对渲染风格、背景贴合痕迹或固定光照模式的依赖。除常规数据增强之外，我们额外加入遮挡增强，用随机裁剪、随机擦除或块状 dropout 的方式模拟局部结构被机械臂遮挡以及相机视角过偏的情况。这类增强会迫使网络在部分局部纹理或者结构不可见时仍利用剩余图像信息进行预测，提高在复杂工况下的鲁棒性。

4 规划问题抽象

4.1 运动问题形式与执行分工

装配任务包含一系列具有语义的操作阶段，系统可划分为三层。任务编排层决定当前处于哪个阶段、调用哪类规划器以及是否需要人工确认或手动操作。规划算法层在给定输入与约束下生成轨迹。执行层负责将自动规划轨迹重参数化，并发送给底层控制器。

设机械臂有 n 个关节，关节空间为 $\mathcal{Q} \subset \mathbb{R}^n$，关节状态为 $\mathbf{q} \in \mathcal{Q}$。机械臂正运动学定义为

$$\mathrm{FK} : \mathcal{Q} \to \mathrm{SE}(3), \qquad \mathbf{q} \mapsto \mathbf{T}_{b,t}, \tag{4.1}$$

其中 t 表示 TCP 坐标系。规划器输出的几何路径可以表示为连续映射 $\mathbf{q}(\xi) : [0,1] \to \mathcal{Q}$，其中 ξ 为无量纲路径参数。离散实现中则使用 $\{\mathbf{q}_i\}_{i=0}^{N-1}$ 表示路径点序列。

本项目使用 I/II/III 型作为规划问题的算法层分类。I 型动作全部由下位机按预置轨迹执行，上位机不参与轨迹规划、插值或重参数化。II 型和 III 型依赖在线视觉位姿、当前关节状态、碰撞模型和任务约束，主要由上位机规划。下位机负责重复动作的稳定播放，上位机负责依赖感知和碰撞模型的在线决策。

4.2 通用可行性与碰撞约束

II 型和 III 型规划都需要在同一套几何约束下判断轨迹可行性。设 $\mathbf{q}_{\min}$ 和 $\mathbf{q}_{\max}$ 分别为关节下限和上限，$\mathrm{collide}(\mathbf{q})$ 为碰撞指示函数：$\mathrm{collide}(\mathbf{q}) = 1$ 表示状态 $\mathbf{q}$ 与障碍物或

自身发生碰撞，$\mathrm{collide}(\mathbf{q}) = 0$ 表示未碰撞。可行状态集合定义为

$$\mathcal{Q}_{\mathrm{free}} = \{\mathbf{q} \in \mathcal{Q} \mid \mathbf{q}_{\min} \le \mathbf{q} \le \mathbf{q}_{\max},\ \mathrm{collide}(\mathbf{q}) = 0\}\,. \tag{4.2}$$

结构障碍物可使用三角网格表示，机械臂连杆可使用胶囊体等简化几何近似。硬碰撞状态直接判为不可用。靠近障碍物或关节限位时，可以通过软代价降低路径偏好：

$$\mathrm{C}_{\mathrm{safe}}(\mathbf{q}) = \lambda_d \max(0, d_{\mathrm{safe}} - \mathrm{d}(\mathbf{q})) + \lambda_l\, \mathrm{L}_{\mathrm{limit}}(\mathbf{q}), \tag{4.3}$$

其中 $\mathrm{d}(\mathbf{q})$ 表示机械臂与障碍物之间的距离估计，d_{safe} 为安全距离阈值，$\mathrm{L}_{\mathrm{limit}}$ 表示接近关节限位的惩罚，λ_d 和 λ_l 为非负权重。

4.3 动作类型与在线规划框架

机械臂动作按来源和约束形式分为 I/II/III 型。三者在系统接口上都表现为一段关节轨迹。I 型由下位机播放预置动作，II 型和 III 型由上位机根据当前位姿、碰撞模型和任务约束在线规划。

I 型表示下位机内部预置的固定动作。执行 I 型动作时，上位机不参与轨迹规划、插值或重参数化，只在任务编排中选择动作编号或切换阶段。这类动作适用于结构位置固定、误差可控、动作高度重复的场景，例如固定结构中的存取能量单元动作。I 型本质上是下位机中的固定轨迹播放，不处理由视觉模块输入的动态装配位姿。

II 型和 III 型都需要在关节空间中寻找在线路径。设规划类型为

$$\tau \in \{\mathrm{II}, \mathrm{III}\}, \tag{4.4}$$

$\mathcal{H}_\tau$ 表示对应任务条件。在线规划问题写为

$$\begin{aligned} \mathbf{q}^*_\tau(\cdot) = \arg\min_{\mathbf{q}(\cdot)} & \int_0^1 \left(\mathrm{D}_{\mathcal{Q}}\!\left(\frac{\mathrm{d}\mathbf{q}(\xi)}{\mathrm{d}\xi}\right) + \mathrm{C}_{\mathrm{safe}}(\mathbf{q}(\xi)) \right) \mathrm{d}\xi \\ \text{s.t.} \quad & \mathbf{q}(\xi) \in \mathcal{Q}_{\mathrm{free}}, \qquad \xi \in [0,1], \\ & \mathbf{q}(\cdot) \in \mathcal{H}_\tau. \end{aligned} \tag{4.5}$$

ξ 是当前规划请求内部的路径进度。装配过程中，任务编排层会根据已经完成的阶段设置规划区间：插入确认后规划剩余滑移段，滑移确认后规划 P 轴动作段，并在 P 轴终点检查或缓存后续 Q 轴调整所需的候选轨迹。$\frac{\mathrm{d}\mathbf{q}(\xi)}{\mathrm{d}\xi}$ 描述关节状态随路径进度的变化量，用于衡量几何路径长度和平滑性，不直接等同于执行时的关节速度。$\mathrm{D}_{\mathcal{Q}}$ 表示路径代价，$\mathrm{C}_{\mathrm{safe}}$ 表示靠近障碍物或关节限位的软惩罚。

后续离散规划会把连续路径拆成相邻路径点之间的转移。关节空间误差向量记为

$$[\mathbf{e}_{\mathcal{Q}}(\mathbf{q}, \mathbf{q}')]_\ell = \begin{cases} \mathrm{wrap}_\pi(q'_\ell - q_\ell), & \ell \in \mathcal{J}_{\mathrm{cont}}, \\ q'_\ell - q_\ell, & \ell \notin \mathcal{J}_{\mathrm{cont}}, \end{cases} \tag{4.6}$$

$$\mathrm{wrap}_\pi(\theta) = \mathrm{atan2}(\sin\theta, \cos\theta),$$

其中 ℓ 为关节索引，$\mathrm{wrap}_\pi : \mathbb{R} \to [-\pi, \pi]$，$\mathcal{J}_{\mathrm{cont}}$ 表示连续旋转关节的索引集合，即没有角度限位、按 2π 周期等价的关节。其他关节按关节坐标直接相减。在具体算法中，$\mathbf{e}_{\mathcal{Q}}$ 可进一步加权并聚合为相邻路径点的关节空间代价 $\mathrm{d}_{\mathcal{Q}}(\mathbf{q}_i, \mathbf{q}_{i+1})$。后续图搜索算法使用 $\mathrm{d}_{\mathcal{Q}}$ 与 $\mathrm{C}_{\mathrm{safe}}$ 构造边代价。

$\mathcal{H}_\tau$ 决定 II 型和 III 型的具体形态。II 型约束起点和终点，III 型约束每个任务进度点。

4.4 II 型点到点规划

II 型表示从当前状态走到一个明确目标。目标可以是末端位姿，也可以是关节状态。设当前关节状态为 $\mathbf{q}_{\mathrm{cur}}$，目标末端位姿为 $\mathbf{T}_{\mathrm{goal}} \in \mathrm{SE}(3)$。II 型只要求路径起点和终点满足条件：

$$\mathcal{H}_{\mathrm{II}} = \left\{ \mathbf{q}(\cdot) \mid \mathbf{q}(0) = \mathbf{q}_{\mathrm{cur}},\ \mathrm{d}_{\mathrm{SE}(3)}(\mathrm{FK}(\mathbf{q}(1)), \mathbf{T}_{\mathrm{goal}}) \le \epsilon_{\mathrm{goal}} \right\}. \tag{4.7}$$

其中 $\mathrm{d}_{\mathrm{SE}(3)}$ 表示末端位姿误差度量，ϵ_{goal} 是允许的终点误差阈值。若目标以关节状态 $\mathbf{q}_{\mathrm{goal}}$ 给出，则终点约束可替换为 $\mathbf{q}(1) = \mathbf{q}_{\mathrm{goal}}$。II 型只约束路径端点，中间路径主要由碰撞、关节限位和路径质量代价决定。它适合解决普通避障、回到安全位、到达接近位姿等问题。

点到点运动规划器可以按求解方式大致分为几类。树搜索式方法以 RRT 及其改进形式为代表，从起点向自由空间随机扩展，通常能较快找到一条可行路径 [10, 11]。RRT* 在扩展过程中加入重连和路径代价优化，在常见技术条件下具有渐近最优性，即采样数量增加时路径代价几乎必然收敛到最优代价 [9]。图搜索式方法先构造离散状态图，再用 A* 或 Dijkstra 等最短路算法求解，适合状态离散结构清晰的场景。轨迹优化式方法通常从给定初值出发，在目标函数中加入平滑性、碰撞距离和关节限位等代价，对整段轨迹进行局部连续优化，代表性方法包括基于序列凸优化的 TrajOpt [15]。混合式方法则显式串联可行路径搜索和局部优化：前者负责找到可执行的轨迹骨架，后者再改善路径长度、平滑性或安全裕度。

II 型规划需要在有限时间内稳定给出可执行的关节空间路径。RRT/RRT* 适用范围广，尤其适合作为可行性基线。随机树直接输出的路径可能包含折返、高曲率片段和局部抖动，通常还需要进一步平滑。BIT*（Batch Informed Trees）在批量采样点上形成隐式随机几何图，并用类似 A* 的启发式顺序搜索该图，使路径代价优化更直接地参与搜索过程 [6]。在本任务中，II 型规划更关注短路径、少折返和易于重参数化的关节空间路径，因此可采用 BIT* 作为主力点到点规划器，并以 RRT* 作为对照基线。

图 5 给出同一障碍物场景下 RRT* 与 BIT* 的仿真路径对比。II 型规划输出首先在关节空间中表现为 $\{\mathbf{q}_i\}$，再通过 FK 映射为 TCP 笛卡尔路径。实际评估时需要同时检查关节曲线、累计关节折转角和 TCP 空间轨迹。累计关节折转角统计相邻关节空间路径段方向变化的总量，用于反映局部折返程度。若关节路径存在明显折返，即使起终点可行，也会给后续重参数化和执行跟踪带来额外压力。

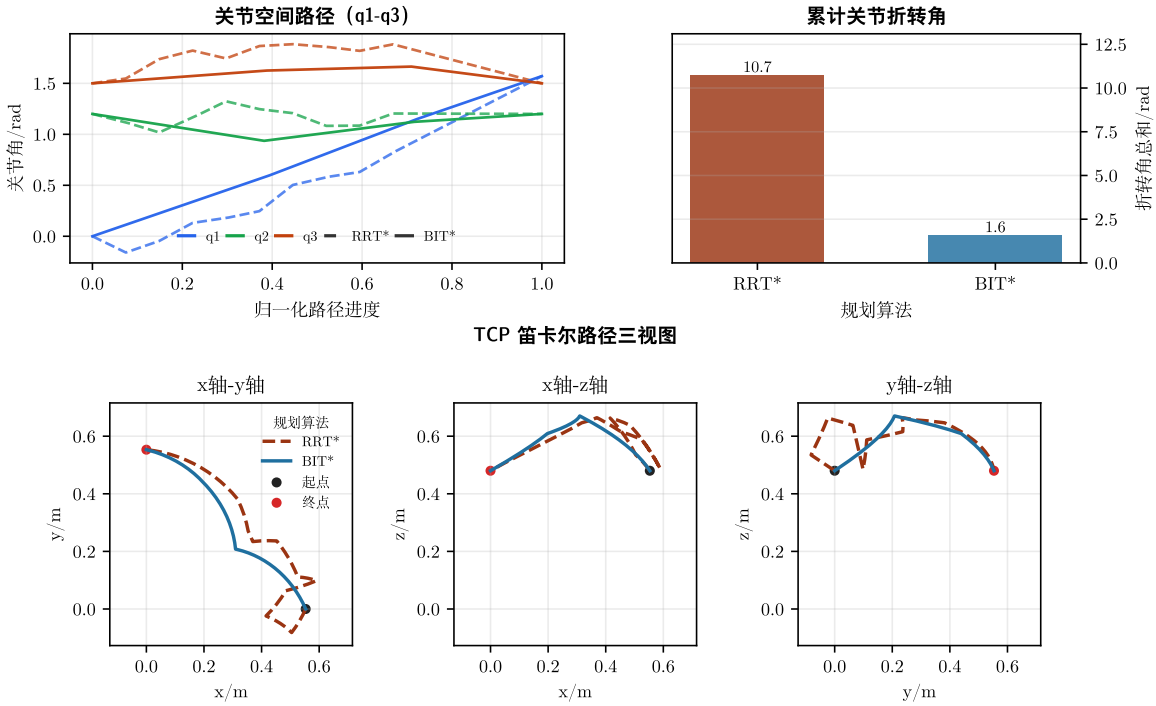


图 5: II 型点到点规划中 RRT* 与 BIT* 的仿真路径对比。两条路径使用相同起终点、关节限位和碰撞约束；图中从关节曲线、累计关节折转角和 TCP 笛卡尔路径三视图观察路径连续性。

4.5 III 型五自由度任务约束

III 型表示沿着任务本身规定的路径执行。一般地，任务约束可以规定“每个任务进度下，末端允许处在哪些位姿上”：

$$\mathcal{H}_{\mathrm{III}}^{\mathrm{gen}} = \{\mathbf{q}(\cdot) \mid \mathrm{FK}(\mathbf{q}(\xi)) \in \mathcal{M}(\xi),\ \xi \in [0,1]\}, \tag{4.8}$$

其中 $\mathcal{M}(\xi) \subset \mathrm{SE}(3)$ 是任务进度 ξ 对应的允许末端位姿集合。

在本项目的装配任务中，感知层输出的是初始装配基准系 E 的位姿。E 系的定义与规则手册和机构模型中科技核心装配基准系的定义保持一致，用于描述装配站/科技核心在初始装配状态下的空间位姿。在 III 型轨迹开始时，随动任务系 s 与 E 重合。随着插入、滑移、P 轴动作和 Q 轴动作推进，s 沿任务轨迹运动。

设 $\mathbf{p}_s^E(\xi) = [x_s(\xi), y_s(\xi), z_s(\xi)]^{\mathsf{T}}$ 为任务进度 ξ 下随动任务系 s 的原点，$\mathbf{n}_s^E(\xi) \in \mathbb{R}^3$ 为对应的单位任务轴，并令 $\mathbf{e}_x = [1,0,0]^{\mathsf{T}}$。选取参考旋转 $\mathbf{R}_s^E(\xi) \in \mathrm{SO}(3)$，使其满足 $\mathbf{R}_s^E(\xi)\mathbf{e}_x = \mathbf{n}_s^E(\xi)$。再设 $\mathbf{T}_{s,t}^{\mathrm{tool}} \in \mathrm{SE}(3)$ 为随动任务系到 TCP 坐标系的固定装配偏置，用于描述工具安装方向、夹爪参考点或安全距离等固定几何关系。

将绕任务轴的自由转角称为 roll，记为 r。由于 roll 是角度变量，使用 $[-\pi, \pi]$ 作为主值区间。实际允许或搜索的 roll 范围记为 $\mathcal{R} \subseteq [-\pi, \pi]$。由 roll 参数化的随动任务系位

姿定义为

$$\mathbf{T}_{E,s}^{\text{task}}(\xi, r) = \begin{bmatrix} \mathbf{R}_s^E(\xi)\mathbf{R}_x(r) & \mathbf{p}_s^E(\xi) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \qquad \xi \in [0,1],\ r \in \mathcal{R} \subseteq [-\pi, \pi], \tag{4.9}$$

其中 $\mathcal{R}$ 为允许的 roll 角集合，$\mathbf{R}_x(r)$ 表示绕 x 轴旋转 r 的旋转矩阵。若工具参考点恰好选在随动任务系原点且姿态与其一致，则 $\mathbf{T}_{s,t}^{\text{tool}}$ 可以取单位阵。一般情况下它不应被省略。五自由度约束对应的允许 TCP 位姿集合可先在 E 系下写为

$$\mathcal{M}_{\text{ex}}(\xi) = \left\{ \mathbf{T}_{E,s}^{\text{task}}(\xi, r)\mathbf{T}_{s,t}^{\text{tool}} \mid r \in \mathcal{R} \right\}. \tag{4.10}$$

该集合固定任务点和任务轴，并通过 $\mathbf{T}_{s,t}^{\text{tool}}$ 保留工具坐标系与随动任务系之间的固定几何偏置；剩余的一维自由度由 roll 角 r 表示。给定当前用于规划的装配基准位姿 $\mathbf{T}_{b,E}$，III 型轨迹需要满足下面的约束。这里 $\mathbf{T}_{b,E}$ 可以来自视觉初值，也可以是在插入确认或阶段确认后由当前 FK 重建得到的基准位姿。$\mathbf{T}_{b,E}\mathbf{T}_{E,s}^{\text{task}}(\xi, r)\mathbf{T}_{s,t}^{\text{tool}}$ 给出基座坐标系下的 TCP 位姿：

$$\mathcal{H}_{\text{III}} = \left\{ \mathbf{q}(\cdot) \mid \exists r(\xi) \in \mathcal{R},\ \text{FK}(\mathbf{q}(\xi)) = \mathbf{T}_{b,E}\mathbf{T}_{E,s}^{\text{task}}(\xi, r(\xi))\,\mathbf{T}_{s,t}^{\text{tool}},\ \xi \in [0,1] \right\}. \tag{4.11}$$

III 型要求整段路径都满足任务约束，输出为满足该约束的一串关节路径点。

5 III 型离散图搜索算法

5.1 从任务约束到离散搜索

式 (4.11) 可以从连续轨迹优化角度求解。将路径离散为 $\{\mathbf{q}_i\}_{i=0}^{N-1}$ 后，一个典型目标是同时最小化路径代价、安全代价和任务误差：

$$\begin{aligned} \min_{\mathbf{q}_0,\ldots,\mathbf{q}_{N-1}} \quad & \sum_{i=0}^{N-2} \mathrm{d}_{\mathcal{Q}}(\mathbf{q}_i, \mathbf{q}_{i+1}) + \sum_{i=0}^{N-1} \mathrm{C}_{\text{safe}}(\mathbf{q}_i) + \sum_{i=0}^{N-1} \mathrm{E}_{\text{task}}(\mathbf{q}_i, \xi_i) \\ \text{s.t.} \quad & \mathbf{q}_i \in \mathcal{Q}_{\text{free}}, \qquad i = 0, \ldots, N-1. \end{aligned} \tag{5.1}$$

连续优化形式适合一般高维轨迹问题，但变量规模随路径点数量增长，碰撞、关节限位和 IK 分支选择也会引入明显的非凸性，使求解更依赖初值和调参。本任务的五自由度约束已经给定 TCP 位置和任务轴方向，剩余自由度主要是 roll 选择，实际部署还进一步固定了解析 IK 分支。因此，本项目采用离散图搜索的方案，在更低维的状态集合上求解路径点，在保证全局最优性的同时亦能确保复杂度可控。

具体做法是先采样任务进度 ξ 和 roll 角 r：

$$\xi_i = \frac{i}{N-1}, \quad i = 0, \ldots, N-1, \qquad r_j \in \mathcal{R}_M, \quad j = 0, \ldots, M-1, \tag{5.2}$$

其中 N 是任务进度采样数，$\mathcal{R}_M \subset \mathcal{R}$ 是包含 M 个候选角度的离散 roll 集合。每个 (ξ_i, r_j) 由式 (4.11) 生成一个候选 TCP 位姿：

$$\mathbf{T}_{i,j}^{\text{task}} = \mathbf{T}_{b,E}\mathbf{T}_{E,s}^{\text{task}}(\xi_i, r_j)\mathbf{T}_{s,t}^{\text{tool}}. \tag{5.3}$$

对 $\mathbf{T}_{i,j}^{\text{task}}$ 求 IK 候选，得到关节状态 $\mathbf{q}_{i,j,k}$，其中 k 表示 IK 分支。实际部署时，根据机械臂关节限位和任务工作区预先选定一个解析分支 k_0，即取 $\mathcal{B} = \{k_0\}$。

这样，III 型规划转化为一个分层路径选择问题：每个任务进度层提供一组候选关节状态，搜索算法从每层选择一个节点组成完整路径。节点和边分别表达硬约束与软代价。这里的最优性限定在给定采样、图连接方式和代价函数之后，即求该离散图上的最小总代价路径。

5.2 图建模与代价函数

构造分层有向无环图 $\mathcal{G} = (\mathcal{V}_{\text{g}}, \mathcal{E}_{\text{g}})$，其中 $\mathcal{V}_{\text{g}}$ 为候选关节状态节点集合，$\mathcal{E}_{\text{g}}$ 为相邻进度层之间的转移边集合。节点定义为

$$\mathbf{v}_{i,j,k} = (\xi_i,\ r_j,\ k,\ \mathbf{q}_{i,j,k}), \tag{5.4}$$

其中 i 是任务进度层编号，j 是自转角采样编号，k 是 IK 分支编号。顶点集合为

$$\mathcal{V}_{\text{g}} = \{\mathbf{v}_{i,j,k} \mid i = 0, \dots, N-1,\ j = 0, \dots, M-1,\ k \in \mathcal{B}\}, \tag{5.5}$$

其中 $\mathcal{B}$ 为参与搜索的 IK 分支集合。实际部署中 $\mathcal{B} = \{k_0\}$。边只连接相邻任务进度层：

$$\mathcal{E}_{\text{g}} = \left\{ (\mathbf{v}_{i,j,k}, \mathbf{v}_{i+1,j',k}) \;\middle|\; \begin{array}{l} i = 0, \dots, N-2, \quad j, j' = 0, \dots, M-1, \\ k \in \mathcal{B}, \quad |j' - j|_{\text{wrap}} \le W \end{array} \right\}. \tag{5.6}$$

其中 $|j' - j|_{\text{wrap}}$ 表示 roll 采样环上的最短索引距离，例如 $j = 0$ 与 $j = M-1$ 相邻。带宽参数 W 限制相邻任务进度层之间允许连接的 roll 变化范围，用来减少远距离跳转，同时降低图搜索复杂度。通用形式中暂不允许跨 IK 分支切换，因为不同解析分支之间的关节角通常存在较大跳变。实际配置中 $\mathcal{B}$ 只包含一个预选分支，因此在线搜索不会遇到分支切换问题。

节点是否可用由 IK 存在性、关节限位和碰撞检测共同决定。严格搜索中，不可用节点不参与搜索。容错搜索可以保留不可用节点，但会给这些节点额外惩罚。软安全代价作为到达下一节点的边代价处理，转移代价可定义为

$$\mathrm{C}_e(\mathbf{v}_{i,j,k}, \mathbf{v}_{i+1,j',k}) = \mathrm{d}_{\mathcal{G}}(\mathbf{v}_{i,j,k}, \mathbf{v}_{i+1,j',k}) + \mathrm{C}_{\text{safe}}(\mathbf{q}_{i+1,j',k}), \tag{5.7}$$

其中 $\mathrm{d}_{\mathcal{G}}$ 是图边上的运动连续性度量。一个常用选择是

$$\mathrm{d}_{\mathcal{G}}(\mathbf{v}_{i,j,k}, \mathbf{v}_{i+1,j',k}) = \lambda_q\, \mathrm{d}_{\mathcal{Q}}(\mathbf{q}_{i,j,k}, \mathbf{q}_{i+1,j',k}) + \lambda_r\, |j' - j|_{\text{wrap}}, \tag{5.8}$$

其中 $\mathrm{d}_{\mathcal{Q}}$ 衡量两组关节状态之间的变化量，可由式 (4.6) 中的 $\mathbf{e}_{\mathcal{Q}}$ 加权后取范数得到。第二项直接惩罚相邻进度层之间的 roll 参数跳变。λ_q 和 λ_r 为非负权重。总代价为所选路径上所有转移边代价之和。从式 (4.5) 的连续形式看，每条边代价对应连续目标在小区间

$[\xi_i, \xi_{i+1}]$ 上的离散近似。为简化符号，记 $\mathbf{q}_i = \mathbf{q}_{i,j,k}$、$\mathbf{q}_{i+1} = \mathbf{q}_{i+1,j',k}$，并令 $\Delta\xi_i = \xi_{i+1} - \xi_i$，则直接离散化时可写成

$$\Delta\xi_i \, \mathrm{D}_{\mathcal{Q}}\left(\frac{\mathbf{q}_{i+1} - \mathbf{q}_i}{\Delta\xi_i}\right) + \Delta\xi_i \, \mathrm{C}_{\mathrm{safe}}(\mathbf{q}_{i+1}). \tag{5.9}$$

实际图搜索把 $\Delta\xi_i$、具体关节度量和 roll 连续性统一写入 $\mathrm{d}_{\mathcal{G}}$ 与权重中，最短路代价就是这些边代价的累加。

5.3 分层最短路求解

对固定 IK 分支 k，III 型图搜索是在任务进度分层图上求一条最小代价路径。实际部署中 $k = k_0$，下面保留 k 是为了维持通用记号。图的第 i 层对应任务进度 ξ_i，层内状态对应 roll 采样编号 j。由于边只连接相邻层，该最短路可以用 Viterbi 递推逐层求解。令

$$\mathcal{W}(j') = \{\, j \mid |j' - j|_{\mathrm{wrap}} \le W \,\}, \tag{5.10}$$

其中 $\mathcal{W}(j')$ 是由带宽 W 决定的前驱 roll 窗口。令 $\rho_i(j,k)$ 表示节点 $\mathbf{v}_{i,j,k}$ 的附加代价，则递推关系为

$$\mathrm{D}_{i+1}(j') = \min_{j \in \mathcal{W}(j')} \left[\mathrm{D}_i(j) + \mathrm{C}_e(\mathbf{v}_{i,j,k}, \mathbf{v}_{i+1,j',k}) + \rho_{i+1}(j',k)\right], \tag{5.11}$$

其中 $\mathrm{D}_i(j)$ 为到达第 i 层第 j 个 roll 状态的最小累计代价，C_e 为边代价。初始层可取 $\mathrm{D}_0(j) = \rho_0(j,k)$。若需要考虑当前关节状态到首个节点的连续性，也可以在 $\mathrm{D}_0(j)$ 中加入对应初始转移代价。通用形式下，可在所有 IK 分支和终点 roll 状态中选择总代价最小的路径。单分支部署时，只需在预选分支 k_0 的终点 roll 状态中取最小值。

节点有效性按两步处理。首先只允许 IK 存在、关节限位和碰撞检查均通过的节点参与搜索，即将无效节点的 $\rho_i(j,k)$ 置为 $+\infty$。若此时不存在完整路径，失败原因可能只是少数坏点，例如某个采样点刚好落在关节限位附近、超出解析 IK 的可达工作范围，或贴近碰撞判定边界。此时再将无效节点的附加代价改为有限高惩罚 $\eta_{\mathrm{bad}} > 0$，重新求解同一个分层最短路问题。该候选路径会倾向于绕开无效节点；若仍需经过少数无效节点，则只对这些节点调用最优近似 IK，并重新进行碰撞检查。

roll 选择和 IK 解选择会影响后续路径的可达性。关节限位较紧或障碍物靠近时，某个局部代价较低的节点未必能够稳定连接到终点。分层最短路搜索在完整规划区间内累计转移代价和安全代价。若严格可行路径被少数坏点截断，则使用带惩罚搜索给出候选路径，再把其中需要修复的节点交给最优近似 IK 做局部处理。

5.4 最优近似 IK 局部修复

最优近似 IK 只用于容错搜索路径上的少数无效节点。它在硬关节限位内寻找一个尽量接近目标 TCP 位姿、同时尽量贴近前驱关节状态的局部解，用来修补严格图搜索

Algorithm 1 III 型分层最短路与最优近似 IK 修复

Require: 任务约束 $\mathbf{T}_{E,s}^{\mathrm{task}}(\xi, r)$，装配基准位姿 $\mathbf{T}_{b,E}$，工具偏置 $\mathbf{T}_{s,t}^{\mathrm{tool}}$，采样参数 N, M，预选 IK 分支集合 $\mathcal{B}$，带宽 W

Ensure: 关节路径点序列 $\{\mathbf{q}_i\}_{i=0}^{N-1}$

初始化分层图 $\mathcal{G}$
for 每个 $i = 0, \ldots, N-1$ **do**
  for 每个 $j = 0, \ldots, M-1$ **do**
    生成 TCP 位姿 $\mathbf{T}_{b,E}\mathbf{T}_{E,s}^{\mathrm{task}}(\xi_i, r_j)\mathbf{T}_{s,t}^{\mathrm{tool}}$
    对 $\mathcal{B}$ 中的解析分支求解 IK 候选 $\mathbf{q}_{i,j,k}$
    标记节点有效性，计算软安全代价
  end for
end for
按 $|j' - j|_{\mathrm{wrap}} \le W$ 构造相邻进度层之间的转移边
令无效节点的 $\rho_i(j,k) = +\infty$
for 每个 IK 分支 $k \in \mathcal{B}$ **do**
  用 Viterbi 递推求解分层最短路
  记录该分支的最优路径与总代价
end for
if 严格搜索找到完整路径 **then**
  return 严格图上的最小代价路径
end if
将无效节点的 $\rho_i(j,k)$ 改为有限高惩罚 η_{bad}
for 每个 IK 分支 $k \in \mathcal{B}$ **do**
  用 Viterbi 递推求解带惩罚的分层最短路
  记录该分支的候选路径与总代价
end for
取带惩罚代价最小的候选路径
for 候选路径上的每个节点 **do**
  if 节点严格有效 **then**
    沿用解析 IK 结果
  else
    用 $\mathbf{T}_{b,E}\mathbf{T}_{E,s}^{\mathrm{task}}(\xi_i, r_j)\mathbf{T}_{s,t}^{\mathrm{tool}}$ 生成目标 TCP 位姿
    以前驱节点关节状态为参考，求解最优近似 IK
  end if
end for
if 修复路径存在碰撞 **then**
  return 容错修复失败
end if
return 修复后的关节路径点序列

被少数无效节点截断的情况。设目标 TCP 位姿和 FK 在关节状态 $\mathbf{q}$ 下给出的 TCP 位姿分别为

$$\mathbf{T}_d = (\mathbf{p}_d, \mathbf{R}_d), \qquad \mathbf{T}(\mathbf{q}) = (\mathbf{p}(\mathbf{q}), \mathbf{R}(\mathbf{q})). \tag{5.12}$$

位置残差直接由两者的位置差给出。姿态残差使用旋转群上的对数映射表示：

$$\mathbf{e}_p(\mathbf{q}) = \mathbf{p}(\mathbf{q}) - \mathbf{p}_d, \qquad \mathbf{e}_R(\mathbf{q}) = \log\!\left(\mathbf{R}_d^{\mathsf{T}}\mathbf{R}(\mathbf{q})\right). \tag{5.13}$$

其中 $\log : \mathrm{SO}(3) \to \mathbb{R}^3$ 表示旋转群到旋转向量的对数映射。由于相对旋转写成 $\mathbf{R}_d^{\mathsf{T}}\mathbf{R}(\mathbf{q})$，$\mathbf{e}_R(\mathbf{q})$ 的三个分量可理解为沿目标 TCP 坐标系三轴展开的旋转误差。在本项目装配约束中，自由自转方向对应目标 TCP 坐标系的 x 轴，该方向与随动任务系 s 的 x 轴方向一致。

为使修复后的关节状态与前一节点连续，还引入关节连续性残差

$$\mathbf{e}_q(\mathbf{q}) = \Delta\mathbf{q}(\mathbf{q}, \mathbf{q}_{\mathrm{ref}}), \tag{5.14}$$

其中 $\mathbf{q}_{\mathrm{ref}}$ 是候选路径中前驱节点对应的关节状态。$\mathbf{e}_q$ 约束修复结果在关节空间中贴近前一节点，降低局部最小二乘跳到另一组等价但不连续 IK 解的概率。数值迭代也以 $\mathbf{q}_{\mathrm{ref}}$ 作为初始值，使结果更容易停留在当前连续分支附近。

记加权范数 $\|\mathbf{x}\|_{\mathbf{W}}^2 = \mathbf{x}^{\mathsf{T}}\mathbf{W}\mathbf{x}$。通常取 $\mathbf{W}$ 为对角权重矩阵，用各维度权重表达位置、姿态和关节连续性的相对重要性。在本任务中，roll 轴是允许放开的方向，因此可将 $\mathbf{W}_R$ 在目标 TCP 的 x 轴方向上的权重取为很小的正数，或在完全释放该方向时退化为 0。姿态残差仍使用完整的三维旋转误差 $\mathbf{e}_R(\mathbf{q})$。最优近似 IK 的局部最小二乘问题可统一写为

$$\mathbf{q}^* = \underset{\mathbf{q}_{\min} \le \mathbf{q} \le \mathbf{q}_{\max}}{\arg\min} \left(\|\mathbf{e}_p(\mathbf{q})\|_{\mathbf{W}_p}^2 + \|\mathbf{e}_R(\mathbf{q})\|_{\mathbf{W}_R}^2 + \|\mathbf{e}_q(\mathbf{q})\|_{\mathbf{W}_q}^2 \right). \tag{5.15}$$

其中 $\mathbf{W}_p$、$\mathbf{W}_R$ 和 $\mathbf{W}_q$ 为权重矩阵，分别控制位置、姿态和关节连续性残差的尺度。实际求解时使用带边界约束的 Trust Region Reflective 最小二乘方法，并将关节上下限作为优化边界。修复后的节点仍需重新通过碰撞检查。该机制用于处理少数解析 IK 断点或局部数值问题，避免整条 III 型轨迹因个别节点失效而直接失败。

6 任务编排层：半自动装配流程

6.1 人机分工与状态机

装配流程采用半自动方式完成。上位机负责视觉定位、接近规划、受约束动作段规划、自动段执行和恢复规划；操作手负责确认、手动插入以及必要微调。插入段完全由操作手通过自定义控制器执行，上位机不为该段生成自动轨迹。任务编排层维护人工操作和自动规划之间的状态，并在每次确认后整理当前关节状态、装配基准位姿和任务进度。

从执行顺序看，流程为：视觉给出 $\mathbf{T}_{b,E}$ 后执行 II 型接近规划；操作手完成插入；插入确认后根据当前 FK 重建装配基准位姿，并进入后续 III 型约束段；完成、中止或重试

时进入恢复链路。图 6 给出了六个典型任务阶段的仿真示意，用于直观展示机械臂与能量单元在半自动装配流程中的相对状态。需要说明的是，该展示场景中的装配柱未随能量单元一起运动，只用于表达任务阶段本身。

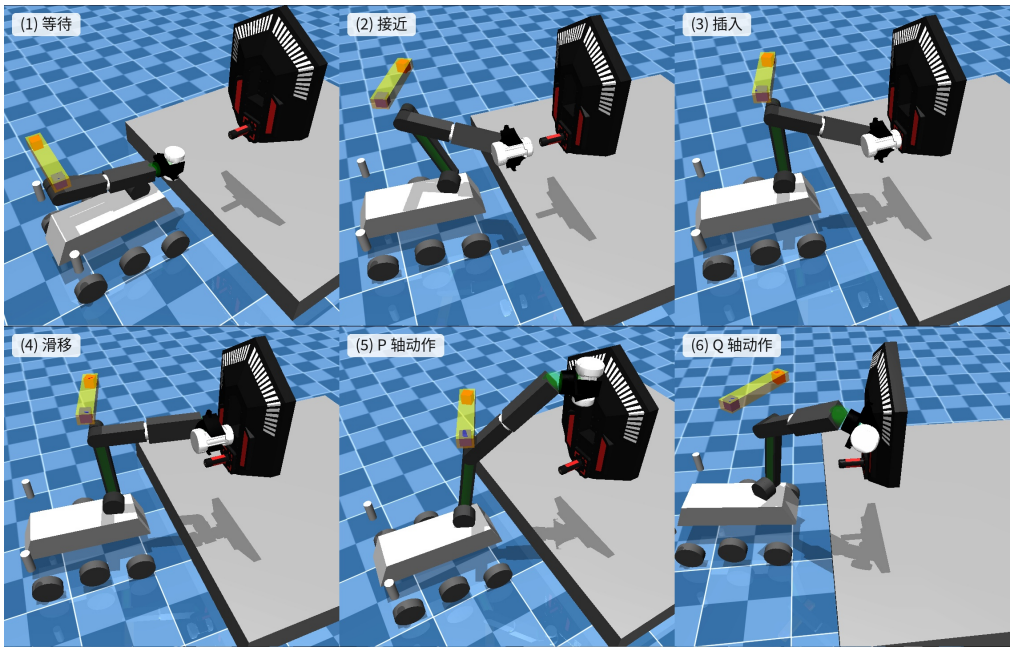
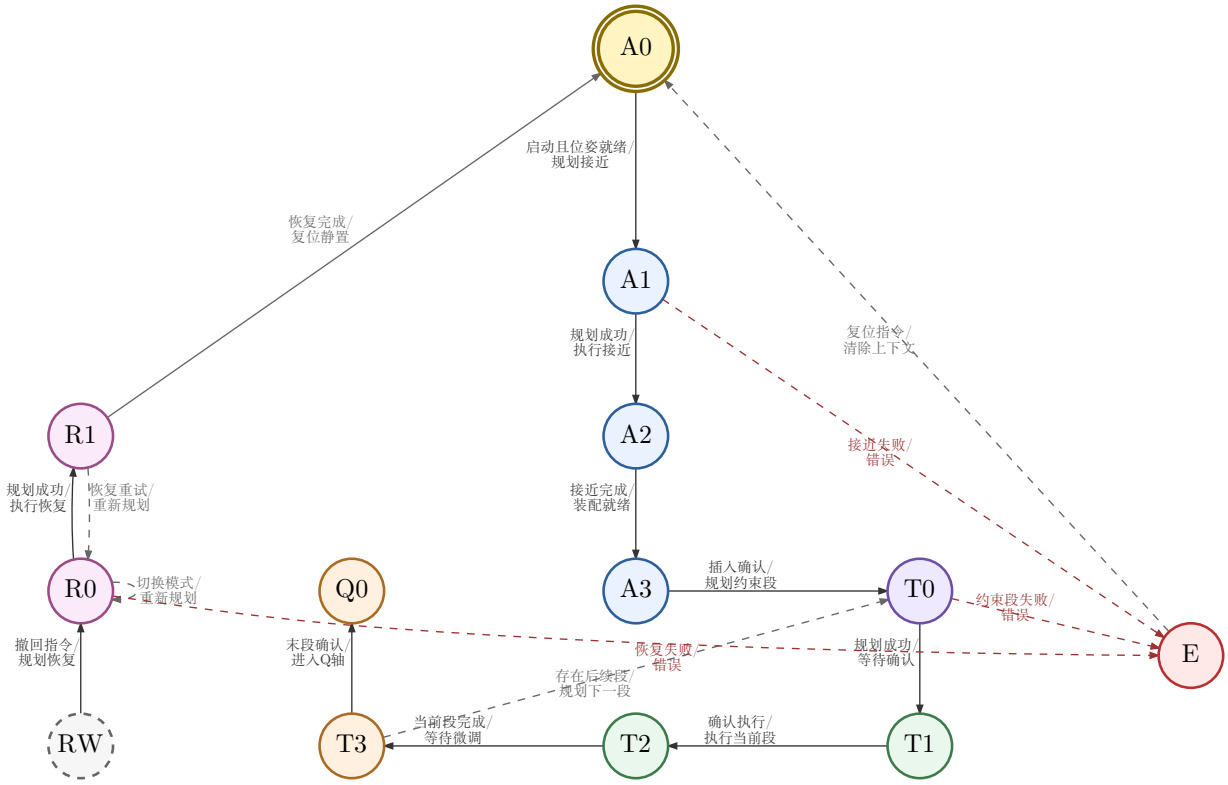


图 6: 半自动装配流程仿真示意图。(1) 等待；(2) 接近；(3) 插入；(4) 滑移；(5) P 轴动作；(6) Q 轴动作。

上位机侧任务状态机负责串接人工确认、规划请求和自动执行回调，其简化表示如图 7 所示。边标注采用“事件（条件）/动作”的形式。事件来自操作手输入、规划结果或执行完成回调，动作表示状态切换时触发的等待、规划或执行请求；连续手动控制量被省略，只保留与自动段交接的确认事件。



代号	状态含义	代号	状态含义	代号	状态含义	代号	状态含义
A0	待命/感知等待	A1	接近规划	A2	接近执行	A3	装配就绪
T0	III 型规划	T1	段就绪	T2	段执行	T3	段微调
Q0	Q 轴手动	RW	可恢复集合	E	错误	R0	恢复规划
R1	恢复执行						

图 7: 上位机侧任务状态机简化图。节点使用短代号表示任务状态，状态含义见表。

A3 表示接近执行完成后的装配就绪状态。T1–T3 表示当前 III 型动作段，可实例化为滑移段或 P 段。RW 表示可由撤回指令触发恢复的状态集合，恢复入口只在装配上下文已经建立后接受；R0–R1 表示局部释放和回撤执行过程。

6.2 接近与手动插入

感知阶段输出初始装配基准位姿 $\mathbf{T}_{b,E}$。任务编排层据此生成接近位姿，并调用 II 型点到点规划。到达接近位姿后，机械臂保持等待，操作手通过自定义控制器将末端插入装配结构内部。插入完成确认后，系统读取当前关节状态，并调用第 6.4 节的方法重建 $\mathbf{T}_{b,E}$。该位姿随后作为 III 型规划和恢复上下文的基准。

6.3 约束段执行、重参数化与恢复

插入确认后的自动段包括滑移/抬升、P 轴动作和 Q 轴相关动作。任务编排层为每一段提交当前所需的目标位姿或任务约束，并维护阶段顺序、确认和重试逻辑。

II 型和 III 型规划器输出的是几何路径点，自动执行前需要转换为带时间戳的轨迹

$$\mathbf{q}_{\mathrm{traj}}(t), \quad t \in [0, T].$$

插入阶段由操作手连续控制，不经过该链路。当前自动段采用固定时长或固定速度的简化重参数化方法，使关节空间速度模长近似固定；在几何路径较平滑时，该方法已能满足当前装配动作的执行需求。

恢复流程用于装配完成、中止或重试后的离场。它以当前关节状态为轨迹起点，先执行结构内局部释放，再回撤到接近起点。局部释放分为两类：抽出复位沿当前装配局部 x 轴把能量单元从结构中直接抽出；侧向释放沿 TCP 接触方向离开约束，可用于装配成功后的离场，也可在直接抽出不可行时先从侧向接触约束中脱出。

6.4 基于当前 FK 的插入后五自由度位姿恢复

视觉位姿需要经过相机内参、外参标定、机械臂连杆参数和坐标系转换等链路进入规划侧。这些建模误差会累积到装配基准位姿中，因此系统在插入完成确认时根据当前关节状态重建 $\mathbf{T}_{b,E}$；后续任务段也使用同一方法更新任务上下文。设当前确认阶段为 g，例如插入完成、滑移完成或 P 轴动作完成。任务模型在装配基准系 E 下给出该阶段 TCP 应处的位置和朝向，记为 $\mathbf{p}_g^E$ 与 $\mathbf{R}_g^E$。当前关节状态通过 FK 给出实际 TCP 位姿：

$$\mathbf{T}_{b,t}^{\mathrm{cur}} = \mathrm{FK}(\mathbf{q}_{\mathrm{cur}}) = \left(\mathbf{p}_t^{b,\mathrm{cur}}, \mathbf{R}_t^{b,\mathrm{cur}}\right). \tag{6.1}$$

五自由度约束中绕任务轴的 roll 为自由量。恢复时只用当前 TCP 位置和 TCP 的 x 轴方向校正装配基准，并沿用上一轮基准中未被约束确定的 roll（绕任务轴自转）。令当前任务上下文中的基准旋转为 $\mathbf{R}_E^{b,\mathrm{ref}}$，第一次插入确认时它来自视觉位姿，后续阶段确认时来自上一轮重建结果。参考基准预测的任务轴和当前 FK 观测到的任务轴分别为

$$\mathbf{a}_{\mathrm{ref}}^b = \mathbf{R}_E^{b,\mathrm{ref}} \mathbf{R}_g^E \mathbf{e}_x, \qquad \mathbf{a}_{\mathrm{cur}}^b = \mathbf{R}_t^{b,\mathrm{cur}} \mathbf{e}_x. \tag{6.2}$$

接下来构造一个最小旋转 $\mathbf{R}_{\mathrm{align}}$，使 $\mathbf{R}_{\mathrm{align}} \mathbf{a}_{\mathrm{ref}}^b = \mathbf{a}_{\mathrm{cur}}^b$。在两轴不退化时，令

$$d = (\mathbf{a}_{\mathrm{ref}}^b)^{\mathsf{T}} \mathbf{a}_{\mathrm{cur}}^b, \qquad \mathbf{v} = \mathbf{a}_{\mathrm{ref}}^b \times \mathbf{a}_{\mathrm{cur}}^b, \qquad \mathbf{K} = [\mathbf{v}]_\times, \tag{6.3}$$

则

$$\mathbf{R}_{\mathrm{align}} = \mathbf{I}_3 + \mathbf{K} + \mathbf{K}^2 \frac{1-d}{\mathbf{v}^{\mathsf{T}}\mathbf{v}}. \tag{6.4}$$

其中 $[\mathbf{v}]_\times$ 表示叉乘矩阵。两轴近似同向时取单位旋转，近似反向时绕参考旋转中的稳定横轴旋转 π，避免除以很小的 $\mathbf{v}^{\mathsf{T}}\mathbf{v}$。恢复出的基准旋转和平移为

$$\mathbf{R}_E^{b,\mathrm{rec}} = \mathbf{R}_{\mathrm{align}} \mathbf{R}_E^{b,\mathrm{ref}}, \qquad \mathbf{p}_E^{b,\mathrm{rec}} = \mathbf{p}_t^{b,\mathrm{cur}} - \mathbf{R}_E^{b,\mathrm{rec}} \mathbf{p}_g^E. \tag{6.5}$$

最终使用

$$\mathbf{T}_{b,E}^{\mathrm{rec}} = \left(\mathbf{p}_E^{b,\mathrm{rec}}, \mathbf{R}_E^{b,\mathrm{rec}}\right). \tag{6.6}$$

作为后续 III 型规划和恢复流程的装配基准。该方法属于任务编排中的状态重建，不作为独立规划算法。

7 系统运行与接口分层

7.1 ROS2 运行架构、接口切换与仿真验证

工程实现中，上位机主节点层由感知节点、任务编排节点和轨迹规划节点组成。感知节点负责把视觉反馈转换为装配站观测位姿，轨迹规划节点只处理规划请求和规划轨迹，任务编排节点则维护半自动装配过程中的状态、人工确认和节点间数据流。这样的划分使感知误差、规划失败和任务阶段切换可以分别定位。

图 8 将系统分为上位机主节点层和交互环境层。交互环境层只暴露视觉反馈接口、关节反馈接口和执行器接口。真实环境中，这三类接口分别连接相机驱动、电机编码器/下位机状态和电机执行器；仿真环境中，则分别连接 MuJoCo 虚拟相机、MuJoCo 关节反馈和 MuJoCo 执行器。上位机算法只依赖这些接口语义，不直接绑定具体硬件或仿真器。

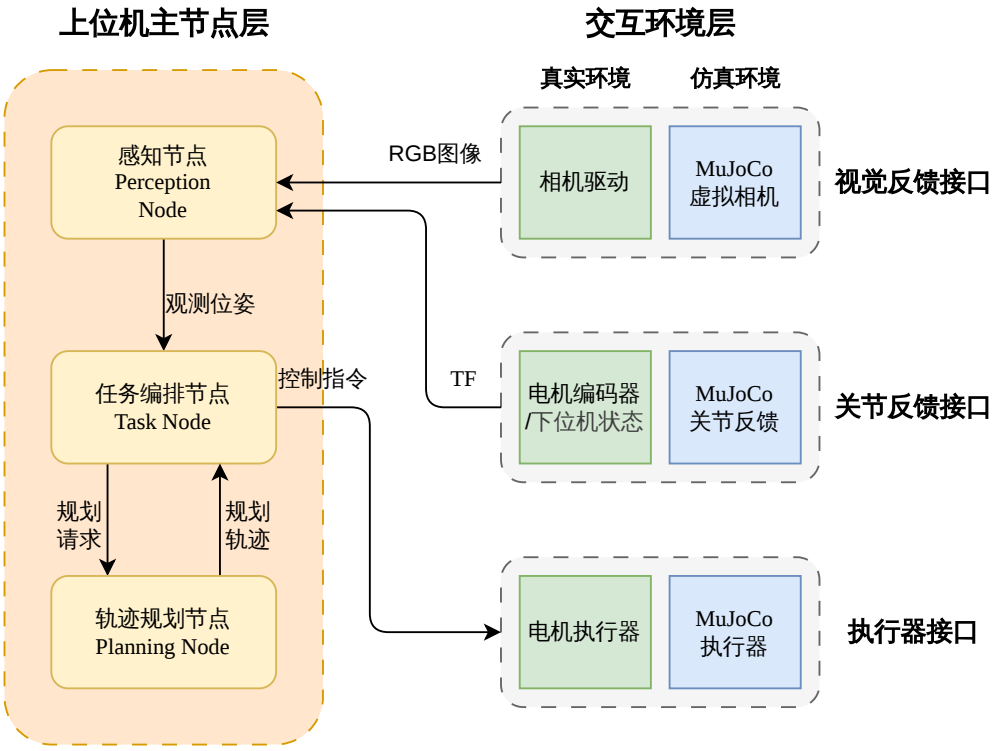


图 8: 上位机主节点层与交互环境层的数据流示意图。

MuJoCo 仿真侧的设计重点是把算法调试从实机占用中解耦出来。装配算法同时依赖视觉位姿、坐标系转换、FK/IK、碰撞检查和任务状态切换，若所有问题都放到真实机器人上验证，调试成本高且容易与底层通信、电机控制和整车联调相互阻塞。仿真环境提供可重复的相机图像、关节反馈和执行反馈，使坐标系约定、任务约束、规划可达性和恢复流程可以先在闭环环境中检查。它不能替代真实场景评估，但能在实机调试前暴露几何建模和接口语义错误，降低技术原型阶段的试错成本。

7.2 技术栈与模块选型

系统实现尽量复用成熟库提供的底层能力，把工程精力集中在任务建模、接口组织和调试闭环上。碰撞检查部分使用 **hpp-fcl** [14, 7]，对应第 4.2 节和第 5.2 节。装配站以 OBJ 网格形式保留，并在机械臂基座坐标系下构建 `BVHModelOBBRSS`；机械臂主要连杆用 `Capsule` 近似。每个候选关节状态先由 FK 更新胶囊体位姿，再通过 `hppfcl.collide` 查询胶囊体与装配站网格的碰撞关系。未发生硬碰撞时，碰撞查询结果中的距离下界提供保守安全裕度，随后转化为 III 型离散图搜索中的安全代价。

II 型点到点避障规划接入 **OMPL** [16]，对应第 4.4 节。OMPL 已经提供关节空间建模、状态有效性检查、优化目标以及 BIT*、RRT、RRT* 等采样规划器。上位机侧只需要给出关节上下界、碰撞有效性函数和起终点状态，即可复用库内规划器生成关节空间路径。

感知侧主要使用 **Ultralytics** [8] 和 **MMPose** [13] 提供的训练封装与模型接口，对应第 2.2 节。*top-down* 方案先采用 YOLO 目标检测模型，从 RGB 图像中给出装配站目标框；随后采用 LiteHRNet 关键点网络，在裁剪后的局部图像上回归 12 个关键点。Ultralytics 封装了 YOLO 的训练、导出和推理流程，便于快速完成检测模型部署；YOLO 训练阶段同时使用 Ultralytics 内置增强和 **Albumentations** [3] 扩展增强。MMPose 则提供关键点数据格式、LiteHRNet 模型配置和训练流程。**Albumentations** 也用于关键点模型训练中的图像增强，包括模糊、噪声、阴影、亮度变化和遮挡增强。运行阶段再通过统一检测器接口接入 PyTorch 或 OpenVINO 推理后端。

最优近似 IK 的局部修复由 **SciPy** [19] 提供数值优化支持，对应第 5.4 节。该步骤调用 `least_squares`，并使用 Trust Region Reflective 方法处理带关节上下界的非线性最小二乘问题。残差项包括 TCP 位置误差、任务轴姿态误差、可选 roll 误差，以及相对于前驱节点的关节连续性误差。

MuJoCo [18] 是面向机器人和接触场景的物理仿真引擎，提供刚体动力学、关节、执行器、传感器和接触求解等基础能力。在第 7.1 节所述的交互环境层中，它被用来提供虚拟相机、关节反馈和执行器反馈，使同一套上位机主节点可以脱离实机闭环运行。坐标系约定、FK/IK、碰撞模型、规划请求、状态切换和恢复流程都可以先在 MuJoCo 中验证，再进入真实机器人调试。

8 局限与后续改进方向

当前框架仍处于原型代码优化阶段。主要算法开发和系统联调以 Python 为主，重点放在任务建模、快速验证和开源复现上，尚未对运行性能进行深层次工程优化。

其次，III 型规划中的最优近似 IK 修复也存在边界。该方法的工程目标是避免少数坏点直接截断整段任务路径，在局部不可达、贴近关节限位或靠近碰撞边界时提供一种 best-effort 的修复手段。当无效节点数量较多时，优化调用次数会显著增加，整体规划时间也会变长。与此同时，当前修复策略主要通过代价项约束位置、姿态和关节连续性，其

修复结果的可行性、连续性以及对整段路径质量的影响尚未给出严格理论证明。

此外, 真实装配仍会受到感知误差、机械结构误差、吸附状态和接触条件的影响。尤其在翻转 P 轴并执行 Q 轴动作时，能量单元存在脱离吸附的风险。该问题本质上不仅是几何轨迹规划问题，还与接触力、下位机执行器的控制精度和执行过程中的扰动有关。后续可以考虑引入力控或力位混合控制，使系统能够感知接触状态并在执行过程中主动调整，从而更彻底地提高装配稳定性。

9 总结

本报告整理了 ENTERPRIZE 战队在 RoboMaster 2026 赛季工程辅助装配算法中的主要设计。整套方案以增量辅助为定位，在不改变下位机主线控制职责的前提下，把可形式化的视觉估计和在线规划放到上位机侧处理。感知侧采用 RGB 关键点 PnP 路线，并通过 Blender 合成数据、真实背景合成和训练增强缓解新规则下真实数据不足的问题。规划侧将动作分为下位机预置固定动作、II 型点到点规划和 III 型任务约束规划，其中 III 型规划使用离散 roll 域上的分层图搜索与动态规划求解，并通过最优近似 IK 对少数局部坏点进行修复。任务编排层将自动规划段、操作手手动插入和恢复流程组织成半自动装配状态机。仿真与真实后端通过统一接口接入同一套上位机节点，使算法能够先在 MuJoCo 中完成坐标系、碰撞、规划和状态切换验证，再进入真实机器人调试。该方案已在真实的赛场环境得到初步验证, 现已详细公开以供后续改进和参考。

参考文献

[1] Rafael Berral-Soler, Rafael Muñoz-Salinas, Rafael Medina-Carnicer, and Manuel J. Marín-Jiménez. Deeparuco++: Improved detection of square fiducial markers in challenging lighting conditions. *arXiv preprint arXiv:2411.05552*, 2024.

[2] Gary Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3067–3074, 2015.

[7] Humanoid Path Planner and Gepetto Team. hpp-fcl: An extension of the flexible collision library. https://github.com/humanoid-path-planner/hpp-fcl, 2026.

[8] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo. https://github.com/ultralytics/ultralytics, 2023.

[9] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[10] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University, 1998.

[11] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[12] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. *arXiv preprint arXiv:2204.06806*, 2022.

[13] MMPose Contributors. Mmpose: Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[14] Jia Pan, Sachin Chitta, and Dinesh Manocha. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3859–3866, 2012.

[15] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[16] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[17] George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. *European Conference on Computer Vision*, pages 478–494, 2020.

[18] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033, 2012.

[19] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[20] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *arXiv preprint arXiv:2312.08344*, 2023.

[21] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

A 坐标系与术语表

采用 $\mathbf{T}_{u,v}$ 表示坐标系 v 到坐标系 u 的刚体变换，即

$$\bar{\mathbf{p}}^{u} = \mathbf{T}_{u,v}\bar{\mathbf{p}}^{v}, \tag{A.1}$$

其中 $\bar{\mathbf{p}}^{u}$ 和 $\bar{\mathbf{p}}^{v}$ 是同一点在不同坐标系下的齐次坐标。表 1 中的坐标系标签只作为上下标标记，不表示待优化的标量变量。

表 1: 主要坐标系与变换记号

符号	名称	含义
b	机械臂基座系	机械臂 FK/IK 和规划使用的根坐标系。
E	初始装配基准系	由视觉给出初值，并可在插入确认等阶段根据当前 FK 重建更新的装配站初始基准系；III 型轨迹开始时与随动任务系 s 重合。
s	随动任务系	随插入、滑移、P 轴动作和 Q 轴动作等任务进度运动的局部任务系。
c	相机系	关键点投影和 PnP 求解使用的相机光学坐标系。
t	TCP 系	夹爪或工具的任务参考坐标系。