arXiv:2304.01583v1 [cs.RO] 4 Apr 2023

HALO：自主系统的危险感知着陆优化

Christopher R. Hayner*、Samuel C. Buckner*,†、Daniel Broyles‡、
Evelyn Madewell、Karen Leung、Beh›cet A›cıkme›se

摘要—随着自主飞行器执行安全关键任务（例如火星科学实验室"好奇号"火星车在火星着陆），自动识别并分析潜在危险着陆点至关重要。本文提出一种耦合的感知—规划解决方案，以应对在不确定环境中着陆时面临的危险检测、最优着陆轨迹生成和应急规划挑战。具体而言，我们开发并结合了两种新算法：风险感知着陆点选择（HALSS）和自适应延迟决策轨迹优化（Adaptive-DDTO），分别应对感知和规划方面的挑战。HALSS框架处理点云信息，以识别可行的安全着陆区；Adaptive-DDTO则是一种多目标应急规划器，可在接收到新的感知信息时自适应地重新规划。我们通过模拟火星环境展示了该方法的有效性，并表明，与非自适应DDTO方法相比，我们的耦合感知—规划方法着陆成功率更高，同时燃料效率也更高。

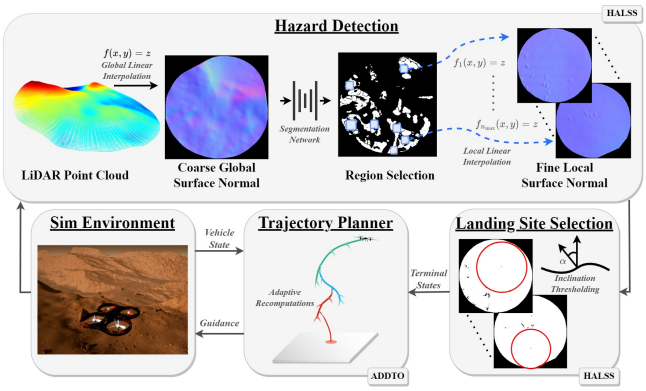


图 1：系统概览：HALSS 识别用于Adaptive-DDTO生成的多目标轨迹优化的可行着陆点。由此得到的制导用于控制（仿真的）四旋翼飞行器，并在处理新的测量数据时持续评估着陆点可行性。

一、引言

自主机器人系统有望执行多种安全关键任务，例如行星探测和灾害应对；若由人类执行，这些任务可能危险、成本高昂或难以实现。对于自主飞行机器人（如无人飞行器）而言，安全着陆对任务成功至关重要，因此机器人必须分析地形条件，识别并避开可能危及任务目标的危险着陆区域。

在本研究中，我们提出一种感知感知轨迹规划算法，使自主飞行机器人能够在未知或不确定的环境中安全着陆。具体而言，该算法将基于激光雷达的危险感知着陆点选择器与自适应多目标轨迹优化器相结合，从而生成燃料效率高的轨迹；随着接收新的观测数据，这些轨迹能够适应着陆点优先级顺序的变化。

相关工作。危险检测是美国国家航空航天局开发行星探测任务精确着陆技术的重要环节 [1]。NASA Safe与精确着陆集成能力演进（SPLICE）项目专门提出了下一代危险检测能力，这些能力主要利用激光雷达生成的数字高程模型（DEM）测量潜在着陆点的坡度和粗糙度 [2]。这些基于数字高程模型、用于危险分类的几何方法包括概率建模 [3]–[5] 以及基于模糊推理的安全地图 [6]。然而，近期研究通过采用深度学习（包括利用语义分割网络识别着陆点危险 [7], [8] ，以及采用贝叶斯深度学习分割和预测方差进行不确定性感知的安全着陆点识别 [9]），在性能上超越了传统方法。我们提出的方法兼具贝叶斯分割的优势，以及对危险地形进行直接几何表示所提供的理论保证。

感知感知型轨迹规划中一个重要的考量是如何管理应急计划：当地图和定位不确定性、激光雷达或其他机载传感器最初未能检测到的细微物体，以及环境中的不确定性导致原本可行的目标状态变得不可行或不切实际时，便需要管理这些应急计划。模型预测控制（MPC）方法已在动态复杂的着陆场景中展现出稳健的轨迹制导性能 [10], 然而，MPC并未考虑多种应急方案。延迟决策轨迹优化（DDTO） [11] 提供了一种框架，可为多个选定目标计算轨迹，同时尽可能延长必须选定目标之前的延迟时间，以满足给定的

*两位作者贡献相同。

† 本文所表达的任何观点、研究结果、结论或建议均属于作者，并不一定反映美国国家科学基金会的观点。

‡ 本文所表达的观点仅代表作者本人，不反映美国空军、美国国防部或美国政府的官方政策或立场。作者谨此感谢 Annika Singh 和 Michael Hayner。美国华盛顿大学航空航天系。电子邮箱：{haynec, sbuckne1}@uw.edu

©2023 IEEE。允许将本材料用于个人用途。其他所有用途均须事先获得 IEEE 的许可，无论使用何种现有或未来媒体，包括转载/再版本材料用于广告或促销目的、创作新的合集作品、转售或重新分发至服务器或列表，或在其他作品中重用本作品中受版权保护的任何组成部分。

次优性容差。DDTO 与用于处理不确定性的随机方法有显著区别：它完全是确定性的，不需要关于环境的概率知识。本文提出的轨迹生成方法（Adaptive-DDTO）结合了DDTO和MPC各自的优势。

贡献。本研究的贡献包括提出用于在不确定且危险的区域确定安全着陆点的新框架，即危险感知着陆点选择（HALSS），以及用于自适应多目标轨迹优化的框架，即（Adaptive-DDTO）。

组织结构。第 II 节和第 V-A 节分别介绍本研究提出的系统架构和所用仿真环境。第 III 节介绍 HALSS 框架，第 IV 节介绍Adaptive-DDTO 算法。第 V 节展示集成仿真与比较研究的结果，随后第 VI 节给出结论。

II. 架构

图 1 展示了所提出方案的整体概况。HALSS 算法根据地形的激光雷达测量数据执行危险检测，并确定以圆形区域表示的安全着陆点。这些着陆点作为终端状态传递给Adaptive-DDTO；随着飞行器经过轨迹中的相应分支点，若发现着陆点不安全或已无法到达，算法便将其移除。随后，Adaptive-DDTO求解着陆优化问题，并由底层控制器执行由此得到的多目标轨迹。着陆点会被传回HALSS ，以便在其范围内识别出新的不安全地形时更新其半径。该循环持续进行，直至自主系统达到截止高度，此时系统锁定最佳着陆点。

III. 危险感知着陆点选择

本节介绍 HALSS 框架，该框架从点云数据中选择安全着陆点。值得注意的是，HALSS 以多尺度执行危险检测，从而在不牺牲检测性能的前提下实现高效计算。我们将全局尺度定义为涵盖整个观测地图，将局部尺度定义为观测地图中较小的子集。HALSS 首先按照 III-A 所述流程，在全局尺度上进行粗略危险检测，以识别较大尺度的危险障碍物。随后，根据得到的全局安全图，按照 III-C 所述流程选取区域，以确定潜在着陆点。对于粗略危险检测识别出的每个区域，均按照 III-B 所述流程进行细粒度危险检测，以识别较小尺度的危险障碍物。最后，根据每个局部安全地图，按照 III-C 所述流程选择一个着陆点。

A. 粗略危险检测

粗略危险检测旨在识别着陆危险，例如大型斜坡和岩石，以生成全局安全图。这项任务通过一个

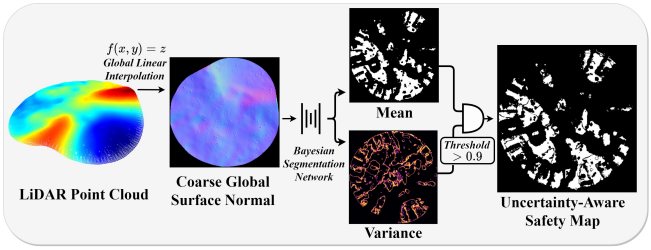


图 2：粗略危险检测流程：将激光雷达生成的表面法线图像输入 Bayesian SegNet，由其对平坦区域和坡地/岩石区域进行分类；输出为不确定性感知安全图，其中白色像素表示安全/平坦区域。

用于将地形分类为安全/不安全的贝叶斯分割网络来完成。本节将讨论粗粒度危险检测流程，如图 2 所示。

全局表面法向量插值： 由于地形表面法线表示能够体现地形坡度和粗糙度特征，因此将其作为分割网络的输入。该表示由三通道数据组成，用于编码每个像素的单位表面法向量，可由LiDAR点云数据推导得到（如图2所示）。为了根据点云生成全局表面法线图像，我们首先使用一种保持粗糙度的降采样算法裁剪点，以缩短插值时间。该算法构建一个二维均匀网格，并在每个网格单元中选取 Z值最大和最小的点，裁剪掉其余点。选取 Z值最大和最小的点可确保裁剪后的点云保留粗糙度；这是最保守的情况，也会产生最不安全的地形。随后，对裁剪后的点云进行线性插值，并以均匀的网格间距进行求值。

网络架构： 我们采用基于贝叶斯Seg-Net架构的贝叶斯分割网络[12]，，该网络非常适合学习高维特征（例如较大的斜坡和岩石）。每组卷积操作之后都加入蒙特卡洛（MC）Dropout层，以处理预测中的不确定性。为生成二值安全图，我们对网络输出应用Sigmoid变换。在推理过程中，全局表面法线通过网络传播，从而生成30个随机样本。这些输出的均值和方差分别用于对预测安全性及其贝叶斯不确定性进行建模，过程与 [9]中所述类似。随后，我们对方差进行归一化，并以0.9这一保守值进行阈值化；该值由实验选定。再对方差图和均值图执行逻辑与运算，得到具有不确定性感知能力的安全映射。

B. 细粒度危险检测

从全球安全图中识别出的潜在着陆点将接受进一步分析，以识别小型危险并为每个着陆区域创建详细的高分辨率安全图。本节将介绍图 3 所示的精细危险检测流程。

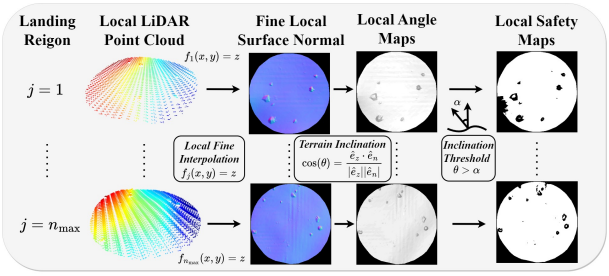


图 3：精细危险检测流程：通过对每个着陆区域的高分辨率局部表面法线进行角度阈值处理，获得局部安全地图。

局部表面法线插值：为了确保捕捉到各区域的较小局部细节，我们从已识别区域中心坐标半径（加上缓冲区）范围内的原始未裁剪点云中选取一个子集，从而得到一组局部点云，如图 3 所示。为避免随着全局点云中不断添加新点而导致计算时间无限增长，随后使用 III-A 节介绍的同一种保留粗糙度的降采样技术，分别对这些点云进行降采样，但采用更精细的网格间距。与全球表面法线的生成方式类似，局部表面法线通过对每个裁剪后的局部点云线性插值一个函数、按均匀网格间距采样，最后计算表面法线来生成。

倾斜度阈值处理： 为了识别更小且维度更低的危险障碍物，采用对地形坡度的纯几何表示非常合适，因为这种表示能够对较小的危险障碍物进行可解释建模。局部角度图通过计算每个局部点云中的单位表面法向量 $\hat{} e_n \in \mathbb{R}^3$ 与局部表面法线图中各点处的垂直单位向量 $\hat{} e_z \in \mathbb{R}^3$ 之间的点积获得。随后，将局部角度图 $\theta \in \mathbb{R}$ 与最大允许倾斜度 $\alpha \in \mathbb{R}$ 进行阈值比较，使得 $\theta > \alpha$ 表示不安全区域，从而生成局部安全地图。

C. 区域与着陆点选择

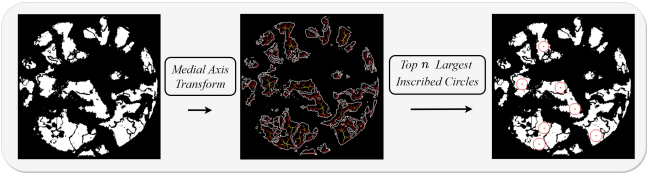


图 4：通过对二值安全映射进行中轴骨架化识别安全区域（红色圆圈）的示例。

通过考察安全区域内最大的内接圆，可以识别潜在安全区域和着陆点；该圆可通过中轴变换（MAT） [13] 轻松求得。中轴是沃罗诺伊图的连续形式；在二维域中，可以将其视为相对于某个边界的最大内切圆盘的圆心集合。对二值图像应用中轴变换后，会得到一个骨架数组，其中的值表示原始对象中各点到边界的距离；在本例中，边界由黑色像素构成。图 4 展示了安全映射中轴骨架化的示例。为找到一组候选着陆点，使用全局安全地图骨架快速、高效地找出排名前 n （所需目标着陆点数量）的最大圆形安全区。为精细检测危险，会对每个局部安全映射重复此过程，并使用 $n = 1$。

评分参数： 获得着陆点后，使用以下指标对每个着陆点进行评分：着陆点内的点云密度（密度）、北东地坐标系中着陆点与飞行器之间的欧几里得距离（接近度）、与其他着陆点的接近程度（聚类）、着陆点内不确定性地图的 99^{th} 百分位数（不确定性），以及着陆点边界圆的半径（尺寸）。

IV. 自适应延迟决策轨迹优化

在危险感知场景中，由于导航误差和环境不确定性，已获取的目标可能在未来某个时刻失去可行性，从而减少可用目标集合。本节介绍一种新算法，旨在始终将飞行器可达集中的目标数量维持在有界范围内，以供路径规划使用。该方法首先从（1）基线轨迹优化问题构建出发，并在此基础上提出（2）单次多目标路径规划方法（DDTO），进而将其改进为自适应重计算的多目标路径规划方法（Adaptive-DDTO）。

A. 基线问题构建

为对飞行器动力学（以多旋翼飞行器为背景）进行建模，考虑如下形式的三自由度（3-DOF）状态空间模型：

$$\dot{x}(t) = Ax(t) + Bu(t) + p \tag{1}$$

其中， $r(t) \in \mathbb{R}^3$ 表示飞行器的位置， $\dot{r}(t) \in \mathbb{R}^3$ 表示飞行器的速度， $x(t) = \left[r(t)^\top,\ \dot{r}(t)^\top\right]^\top \in \mathbb{R}^6$ 表示飞行器的状态。此外，控制信号 $u(t) \in \mathbb{R}^3$ 表示飞行器的指令推力（下文记作 $T(t)$）。因此，飞行器的简化动力学可描述为带有仿射重力项的双积分器系统，即：

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ \frac{I}{m} \end{bmatrix},\ p = \begin{bmatrix} 0 \\ -g \end{bmatrix} \tag{2}$$

其中， $g \in \mathbb{R}^3$ 表示重力加速度， $m \in \mathbb{R}$表示飞行器的质量， $I \in \mathbb{R}^{3\times3}$ 表示单位矩阵。我们考虑一个固定于地表的惯性坐标系，其基向量为 $e_j \in \mathbb{R}^3$，分别对应 $j = 1,\ 2,\ 3$，即顺航向、横航向和高度。状态空间模型 (2) 假设重力恒定，并忽略空气动力阻力。

单次轨迹优化问题（BaseProblem）的目标是在满足式（1）中的动力学以及状态 $x(t)$ 和控制 $u(t)$ 约束的条件下，引导目标从其当前空中状态 x_0 到达地面上的安全目标区域 x_f，同时最小化控制力（累积推力）：

BasePROBLEM（三自由度非凸） (3)

$$\begin{aligned}
&\text{最小化}_{T(\cdot)} && \int_0^{t_f} kT(\tau)k_2 d\tau && \text{(3a)}\\
&\text{满足以下条件} && \dot{x}(t) = Ax(t) + Bu(t) + p, && \text{(3b)}\\
& && 0 < T_{\min} \le \|T(t)\|_2 \le T_{\max}, && \text{(3c)}\\
& && \|T(t)\|_2 \cos(\gamma_{\max}) \le e_3^\top T(t), && \text{(3d)}\\
& && \|\dot{r}_{1:2}(t)\|_2 \le v_{\max}^{\text{lat}}, && \text{(3e)}\\
& && \text{abs}(\dot{r}_3(t)) \le v_{\max}^{\text{vert}}, && \text{(3f)}\\
& && x(0) = x_0, x(t_f) = x_f && \text{(3g)}
\end{aligned}$$

在问题（3）中，代价函数（3a）旨在最小化累积指令推力，并满足动力学约束（3b）、推力大小的下界（$T_{\min}$）和上界（$T_{\max}$）约束（3c）、相对于垂直方向的倾斜角上界 $\gamma_{\max}$ e_3 （3d）、横向速度上界 $v_{\max}^{\text{lat}}$（3e）、垂直速度上界 $v_{\max}^{\text{vert}}$ （3f），以及初始和终端边界条件（3g）；飞行时间固定为 $t_f \in \mathbb{R}$。

式 (3c) 中的下界导致该约束乃至整个问题变为非凸。无损凸化（LCvx）方法（见 [14], [15]）能够以无损（保留全局最优性）的方式对这些约束进行凸化，并提供可行性保证。为此，可以引入松弛变量 $\Gamma(t) \in \mathbb{R}$，在代价函数 (3a) 和约束 (3c)–(3d) 中替代$\|T(t)\|_2$ ，并增加一个松弛约束 $\|T(t)\|_2 \le \Gamma(t)$。本文其余部分均采用对 BasePROBLEM 的这一扩展，并将其称为L CvxPROBLEM。由于多旋翼飞行器的微分平坦性 [16], [17], LCvxPROBLEM提供的三自由度解可视为完整六自由度解（偏航不受约束）的替代方案。

B. 自适应延迟决策轨迹优化

延迟决策轨迹优化（DDTO） [11]扩展了 LCvxPROBLEM，使其能够进行多目标路径规划，同时将多个目标状态保持在飞行器的可达集中，并根据给定的次优容差（相对于针对同一目标的最优单次 LCvxPROBLEM 解）和目标拒绝序列，最大化到达每个目标的延迟时间（见图 5）。Adaptive-DDTO算法基于这一思路，构建了一个容错轨迹优化框架，使飞行器始终至少考虑 $n_{\min} \in \mathbb{N}$ 个目标、至多考虑 $n_{\max} \in \mathbb{N}$个目标（见图 6 和算法 1）。以下设计可视为类似于收缩时域模型预测控制问题（例如 [10]），并在非均匀时间间隔后有条件地执行。以下段落将介绍其先决条件

算法 1 背后的知识。[1]

[1]请注意，在算法 1 中调用底层算法时， [11], LCvxPROBLEM用作Opt- Problem，并使用相应的约束集来求解每个子问题。

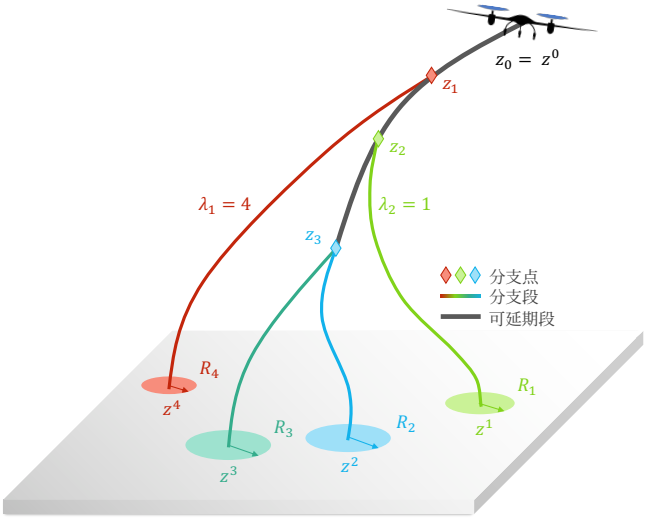


图 5：使用 $n = 4$ 个目标的多旋翼着陆问题的DDTO 求解结果示意图。

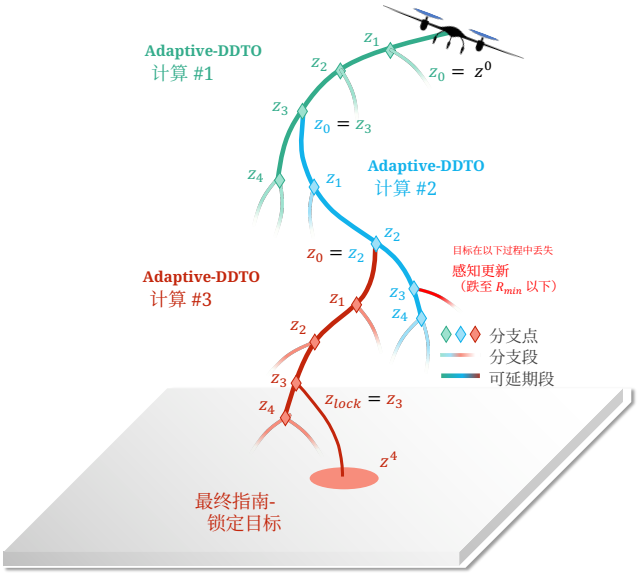


图 6：使用 $n_{\min} = 3$ 和 $n_{\max} = 5$解决多旋翼着陆问题的 Adaptive-DDTO 示意图。

考虑一个维度为n_x， $n_u \in \mathbb{N}$ 的状态空间系统，这两个维度分别对应状态数和控制输入数。飞行器的初始状态记为 $z^0 \in \mathbb{R}^{n_x}$，其 $n \in \mathbb{N}$ 已获取的目标状态 $z^j \in \mathbb{R}^{n_x}$ 对应于 $j \in J$，其中 $J \subset \mathbb{N}$ 是用于标识 n 目标的有限索引集。此外，还存在$n-1$ 个分支点$\{z_k\}_{k=1}^{n-1} \subset \mathbb{R}^{n_x}$，其对应的时间范围为$\{\tau\}_{k=1}^{n-1} \subset \mathbb{R}$。每个 z_{k-1} 与 z_k之间都存在 $n-1$ 可延期段，起始于$z_0 = z^0$；此外，每个目标都对应一个次优性容差$\{\epsilon\}_{k=1}^{n}$。每个目标还关联一个安全映射中的边界圆（见第 III 节），该边界圆会在每个分支点更新，半径为 $R^j \in \mathbb{R}$。为简化后续记法，这些半径将被隐式视为 z^j 的一个（动态更新的）参数（以及第 III-C 节讨论的评分参数）。

DDTO还需要指定目标拒绝顺序$\{\lambda_k\}_{k=1}^{n-2} \subset \mathbb{N}$，其中$J$中的标识标签按延后顺序指定。为此，为每个目标分配一个期望度评分，并用评价函数 g_{des} 表示：$\mathbb{R}^{n_x} \to [0,\infty)$。该评价函数的一种启发式方案是采用加权和 $g_{\text{des}}(z^j) = w_{\text{des}}^\top c_{\text{des}}$，其中 $c_{\text{des}} \in \mathbb{R}^5$ 包含与点云密度、与飞行器的接近程度、场地聚类、网络预测不确定性和着陆区大小相关的评分，如第 III-C 节所述；$w_{\text{des}} \in \mathbb{R}^5$ 则包含针对 c_{des}中各元素的相应可调权重。

函数 $g_{\text{rem}} : \mathbb{R}^{n_x} \to \{0, 1\}$ 用于判断是否应移除目标；如果目标包围半径超过最小安全包围半径，则返回 `true`，该半径记为 $g_{\text{rem}}(z^j) = R^j < R_{\min}$。

分支点处的切换决策由函数 $g_{\text{switch}} : \mathbb{R}^{n_x} \to \{0, 1\}$ 表示；如果待延后处理的目标所对应的期望度评分高于所有其他目标的评分，则返回 `true`，即 $g_{\text{switch}}(z^j) = g_{\text{des}}(z^j) > \max(\{g_{\text{des}}(z^i)\}^{i\in\{J/j\}})$

在某些条件下，最好将制导锁定到当前最优目标 $z^{j_{\text{lock}}}$，禁用Adaptive-DDTO 重新计算，并在当前状态 z_{lock} （隐式更新）下，以当前时间 τ_{lock} 为起点持续向该目标施加控制，直到达到终止条件 $z^{j_{\text{lock}}}$。可以根据最小“截止”高度 h_{cutoff} 设定合理的锁定条件，使 $g_{\text{lock}} : \mathbb{R} \to \{0, 1\}$ 定义为 $g_{\text{lock}}(h) = h < h_{\text{cutoff}}$。要索引 z_k 的高度，需要使用向量 $e_h = [0, 0, 1, 0, 0, 0]^\top$。

此外，还有其他几个适用于Adaptive-DDTO公式的实用函数和关系：

- $h_{\text{获取}}$： $\mathbb{N} \to \mathbb{R}^{n_x}$ 调用HALSS以获取$n_{\text{最多}} - n$ 个额外目标，其中$n \geq n_{\text{最少}}$是当前目标数量。
- $h_{\text{更新}}$： $\mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ 调用HALSS 以更新目标 z^j 及其相关属性。
- 二分： 问题$\to \mathbb{N}$ 针对固定飞行时间问题（例如LCvxProblem），在飞行时间范围内进行最小可行性二分搜索，返回给定离散化时间步长$\Delta t \in \mathbb{R}$ 时的最优离散化节点数 N^* （使得 $t_f^* = N^*\Delta t$）。• 排序排列$_n$：$\mathbb{R} \to \mathbb{N}$ 按单调递增顺序对基数为 $n \in N$ 的输入集合排序，¯并返回指示排序顺序的排列索引。
- 通信控制： $(\mathbb{R}^m \times \mathbb{N} \times \mathbb{R} \times \mathbb{R}) \to \varnothing$ 从DDTO返回的拼接向量 $V \in \mathbb{R}^m$ 中提取控制序列并执行控制（其中 m是控制元素的总数，见 [11]），使其朝向目标 j，时间范围为 t_1 至 t_2。值得注意的是，算法 1包含对自身的递归调用

用于在 $n < n_{\min}$ 时进行自适应 重新计算。

算法 1： Adaptive-DDTO

输入： z^0, J, $\{z^j\}_{j\in J}$

/* Begin initialization */

$z_0 \leftarrow z^0$

$\{z^j\}_{j=1}^{n_{\max}} \leftarrow h_{\text{acquire}}(n_{\max} - |J|)$

$n \leftarrow n_{\max}$

$J \leftarrow 1 : n$

$\{\lambda_k\}_{k=1}^{n-2} \leftarrow$ 排序置换$_n(\{g_{\text{des}}(z^j)\}_{j\in J})$

/* End initialization */

对于 $j \in J$ **执行**

  $N^j \leftarrow$ 二分搜索(LCvxProblem)

$\{\tau_k\}_{k=1}^{n-1}, \{z_k\}_{k=1}^{n-1}, V \leftarrow$ DDTO$(J, \{\lambda_k\}_{k=1}^{n-2}, z_0, \{z^j\}_{j\in J}, \{e^j\}_{j\in J}, \{N^j\}_{j\in J})$

对于 $k = 1, \ldots, n-1$ **执行**

  $z^j \leftarrow h_{\text{update}}(z^j, k)$

  如果 $g_{\text{switch}}(z^j)$**则**

    $J \leftarrow \{\lambda_k\}$

  else

    $J \leftarrow \{J/\lambda_k\}$

  对 $j \in J$ **执行**

    如果 $g_{\text{rem}}(z^j)$**，则**

      $J \leftarrow \{J/j\}$

  $n \leftarrow |J|$

  如果 $n < n_{\min}$**，则**

    Adaptive-DDTO$(z_k, J\{z^j\}_{j\in J})$

  如果 $g_{\text{lock}}(e_h^\top z_k)$**，则**

    $j_{\text{lock}} \leftarrow \text{argmax}(\{g_{\text{des}}(z^j)\}_{j\in J})$

    通信控制$(V, j_{\text{lock}}, \tau_{k-1}, N^{j_{\text{lock}}}\Delta t)$

  else

    通信控制$(V, \lambda_k, \tau_{k-1}, \tau_k)$

V. 结果

本节讨论了某一系统集成仿真的实现细节和实验结果，并介绍了组合式 HALSS Adaptive-DDTO 实现，以及 Adaptive-DDTO 与 DDTO 的比较研究分析结果。

A. 实验设置与仿真环境

我们使用虚幻引擎4的微软 AirSim 插件，创建了一个集成实现，结合HALSS和Adaptive-DDTO，用于在火星表面（由 NASA高分辨率成像科学实验成像）着陆四旋翼飞行器。该插件提供了一个仿真的多旋翼飞行器模型。值得注意的是，为简化起见，仿真飞行器采用地球的重力和大气条件；然而，本文提出的方法可推广至任意环境和飞行器配置，而不失一般性（例如火星直升机[18]）。仿真算法分别使用 Python（HALSS）和 Julia（Adaptive-DDTO）实现[2]。点云由 AirSim 的机体固定式激光雷达传感器模型生成，其最大量程为 500 米，采用圆形扫描模式，具有 30° 径向视场。

[2]代码可在此处获取： https://github.com/UW-ACL/HALO

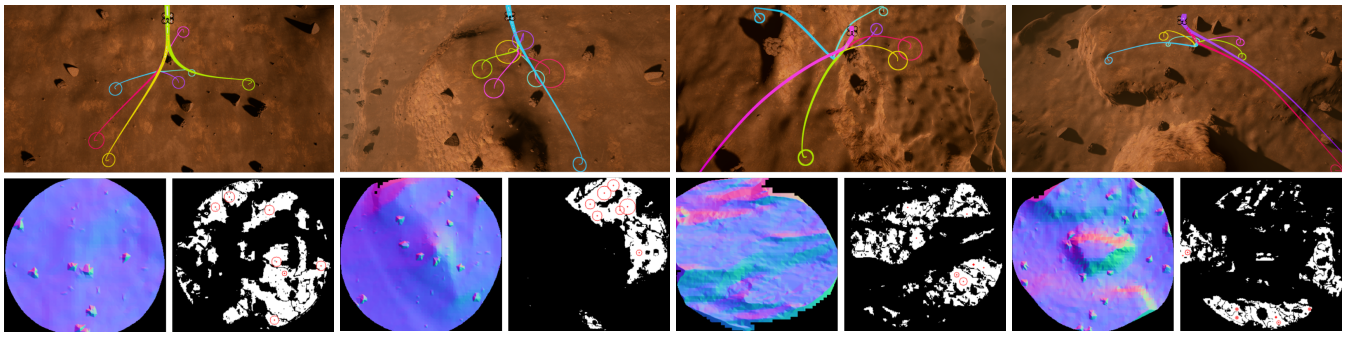
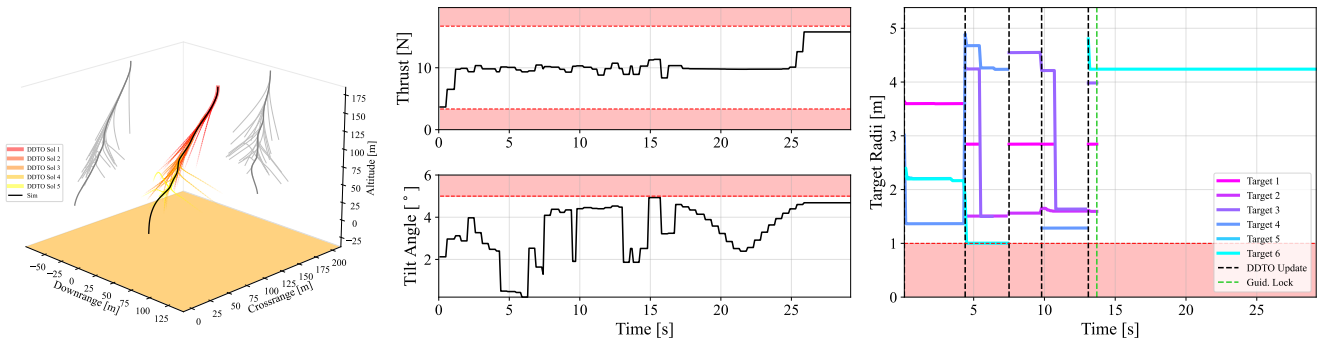


图 7：四个仿真实例在复杂度从左至右递增的不同地形图上的快照。上方图像展示了运行期间的仿真环境，其中包含 DDTO 分支和着陆点；下方图像展示了全球表面法线图（左）和安全性图（右），以及相应的着陆点（红色圆圈）。



（a）参数化三维轨迹　　（b）推力大小和目标半径历史

图 8： 图 7 中最复杂（最右侧）测试用例的累计仿真结果。

B. 综合仿真结果

图 7 展示了我们的耦合解在仿真环境[3]的四种不同环境中成功运行的四个案例。图 8a 展示了四旋翼飞行器的完整轨迹，以及每个 DDTO 解在三维空间中的分支。值得注意的是，在此次仿真中，四旋翼飞行器的初始状态位于一处大型岩层上方，这导致了较大的偏航。在图 8b 左侧，可以看到推力和倾斜角控制指令，它们满足优化约束（红色区域）。右侧展示了各目标半径随时间的变化。半径出现较大的垂直跳变，表示此前未观测到的岩石被识别为某个着陆点半径范围内的危险障碍物。较小的跳变则可能是因为圆周边界上的危险障碍物轮廓变得更加清晰，并随着四旋翼飞行器下降而逐渐侵入圆内。这些结果表明，我们针对 HALO 问题的耦合 HALSS-Adaptive-DDTO解能够适应较大的偏航，以及着陆点的不确定性。

C. 比较研究：*DDTO* 与 *Adaptive-DDTO*

在沙盒环境中对DDTO和Adaptive-DDTO 进行了比较研究，从一个150米的初始高度，且目标按均匀随机分布生成，其中每个目标每隔 Δt 秒有1%的失败概率。在100次蒙特卡洛试验中，DDTO 成功着陆的比例为72.41%，均值累计控制力为295.05牛·秒；而Adaptive-DDTO对应的数值分别为85.56%和293.72牛·秒。这意味着，与DDTO相比，Adaptive-DDTO的着陆成功率提高了18.16%，累计控制力降低了0.4%。

VI. 结论

在本研究中，我们提出了两个新框架——HALSS 和 Adaptive-DDTO，分别应对感知和规划方面的挑战，从而解决风险感知着陆优化问题。我们在逼真的仿真环境中展示了这些方法的有效性，并使HALSS 和 Adaptive-DDTO均实现了近实时性能。尽管本文介绍了一种集成实现，这些算法也可独立运行，且不依赖特定的仿真环境。此外，我们的方法适用于任何需要自主系统在不确定、动态且可能存在危险的环境中导航的应用场景。

[3]所有仿真均使用以下超参数：$n_{\min} = 3$、$n_{\max} = 7$、$\epsilon^j = 0.1\ \forall j \in J$、$w_{\text{des}} = [0,0,1,0,0]^\top$、$h_{\text{cutoff}} = 65$　$\alpha = 8^\circ$　米、$\Delta t = 0.5$ 秒、$T_{\min} = 3.34$ 牛、$T_{\max} = 16.72$ 牛、$\gamma_{\max} = 5^\circ$、$v_{\max}^{\text{lat}} = v_{\max}^{\text{vert}} = 5$ 米/秒和 $m = 1$ 千克。

参考文献

[1] C. D. Epp、E. A. Robertson 和 T. Brady，“自主着陆与危险规避技术（alhat）”，载于《2008 年 IEEE 航空航天会议》，2008 年，第 1–7 页。[2] C. I. Restrepo、P.-T. Chen、R. R. Sostaric 和 J. M. Carson，“下一代美国国家航空航天局危险检测系统的开发”，载于《AIAA 科技论坛 2020》，丛书《AIAA 科技论坛》。美国航空航天学会，2020 年 1 月。[3] T. Ivanov、A. Huertas 和 J. M. Carson，“面向自主安全着陆的概率危险检测”，载于《AIAA 制导、导航与控制（GNC）会议》，丛书《制导、导航与控制及同期会议》。美国航空航天学会，2013 年 8 月。[4] R. Brockers、J. Delaune、P. Proen ̃ca、P. Schoppmann、M. Dom{ } nik、G. Kubiak 和 T. Tzanetos，“面向未来火星科学直升机的自主安全着陆点检测”，载于《2021 年 IEEE 航空航天会议（50100）》，2021 年，第 1–8 页。[5] P. F. Proen ̧ca、J. Delaune、Rol 和 Brockers，“面向自主无人机的地形测绘与着陆点检测优化”，载于《2022 年国际机器人与自动化会议（ICRA）》，2022 年，第 9668–9674 页。[6] K. Konstantinidis 和 R. F ̈orstner，“在羽流源附近着陆的制导、导航与控制系统：以土卫二为例”，《制导、控制与动力学杂志》，第 43 卷，第 12 期，第 2289–2308 页，2020 年 12 月。[7] R. Moghe 和 R. Zanetti，“一种用于自主月球着陆危险检测的深度学习方法”，《航天科学杂志》，第 67 卷，第 4 期，第 1811–1830 页，2020 年 12 月。[8] K. Tomita、K. Skinner、K. Iiyama、B. Jagatia、T. Nakagawa 和 K. Ho，“基于语义分割的行星着陆危险检测算法”，载于《ASCEND 2020》，丛书《ASCEND》。美国航空航天学会，2020 年 11 月。[9] K. Tomita、K. A. Skinner 和 K. Ho，“用于自主安全行星着陆的分割贝叶斯深度学习”，arXiv:2102.10545，2021 年 2 月。[10] W. B. Greer 和 C. Sultan，“用于直升机—舰船着舰的缩短时域模型预测控制方法”，《制导、控制与动力学杂志》，第 43 卷，第 5 期，第 884–900 页，2020 年。[11] P. Elango、S. B. Sarsılmaz 和 B. A ̧cıkme ̧se，“多目标轨迹优化中的延迟决策”，载于《AIAA 科技论坛 2022》，2022 年，第 1583 页。[12] V. B. Alex Kendall 和 R. Cipolla，“Bayesian SegNet：用于场景理解的深度卷积编码器—解码器架构中的模型不确定性”，载于《英国机器视觉会议论文集（BMVC）》，G. B. Tae-Kyun Kim、Stefanos Zafeiriou 和 K. Mikolajczyk 编，BMVA Press，2017 年 9 月，第 57.1–57.12 页。 [在线]。可访问：https://dx.doi.org/ 10 5244 31 57 。/ C. . H. I. Choi、S. W. Choi 和 H. P. Moon，“中轴变换的数学理论”，《太平洋数学杂志》，第 181 卷，第 1 期，第 57–88 页，1997 年 11 月。[13] [14] B. A ̧cıkme ̧se 和 L. Blackmore，“一类具有非凸控制约束的最优控制问题的无损凸化”，《Automatica》，第 47 卷，第 2 期，第 341–347 页，2011 年。[15] D. Malyuta、T. P. Reynolds、M. Szmuk、T. Lew、R. Bonalli、M. Pavone 和 B. Acikmese，“用于轨迹生成的凸优化”，arXiv 预印本 arXiv:2106.09125，2021 年。[16] M. Fliess、J. L ́evine、P. Martin 和 P. Rouchon，“非线性系统的平坦性与缺陷：入门理论及示例”，《国际控制杂志》，第 61 卷，第 6 期，第 1327–1361 页，1995 年。[17] N. T. Nguyen、I. Prodan、F. Stoican 和 L. Lefevre，“基于微分平坦性的四旋翼轨迹跟踪可靠非线性控制”，《IFAC-PapersOnLine》，第 50 卷，第 1 期，第 6971–6976 页，2017 年。[18] B. Balaram、T. Canham、C. Duncan、H. F. Grip、W. Johnson、J. Maki、A. Quon、R. Stern 和 D. Zhu，“火星直升机技术验证机”，载于《2018 年 AIAA 大气飞行动力学会议》，2018 年，第 0023 页。