arXiv:2303.01614v1 [cs.RO] 2 Mar 2023

STEP：面向风险感知越野导航的随机可通行性评估与规划；DARPA地下挑战赛成果

Anushri Dixit[*1]、**David D. Fan**[*2]、**KyoheiOtsu**[2], **Sharmita Dey**[2], **Ali-Akbar Agha-Mohammadi**[2], 和 **Joel Burdick**[1]

[1] 加州理工学院[2] NASA喷气推进实验室、加州理工学院[*] 表示贡献相同。©2023，加州理工学院。保留所有权利

摘要

尽管自主导航已在结构化和受控环境中得到广泛应用，但在未知的越野地形中实现机器人自主性仍是一项艰巨挑战。未开发荒野、洞穴、瓦砾以及其他灾后场地等极端、越野和非结构化环境，为自主导航带来了独特且严峻的难题。基于我们参加DARPA地下挑战赛的经验，我们提出一种名为 STEP（随机可通行性评估与规划）的可通行性评估与规划框架，旨在提升机器人在感知条件恶劣且完全未知的地下环境中的自主通行能力。我们介绍：1）快速的不确定性感知建图与可通行性评估；2）采用条件风险价值（CVaR）进行尾部风险评估；3）基于序列二次规划（SQP）的模型预测控制（MPC），实现高效且兼顾风险与约束的运动学-动力学运动规划；4）针对可能导致失败的意外情况设计的快速恢复行为；以及5）面向四足机器人的基于风险的步态调整。我们展示并验证了大量实验结果：在加利福尼亚州 Valentine Cave（洞穴环境）、肯塔基州 Kentucky Underground（矿井环境），以及肯塔基州 Louisville Mega Cavern（DARPA地下挑战赛的最终比赛场地，包含隧道、城市和洞穴环境）开展实地研究，使用轮式和足式机器人平台进行测试。

1 引言

在极端危险地形中实现稳健自主通行仍是一个尚待解决的问题，在外星环境、 [42], 灾区 [1], 和地下环境 [34] 中有着诸多应用。机器人依靠带噪声的传感器测量值和定位不确定性，在崎岖且风险极高的地形中运行。在此类条件下进行自主运动规划，需要一个能够考虑由瓦砾、陡降地形、泥泞／湿滑区域等因素带来的可通行性风险，同时兼顾机器人状态估计不确定性的框架。该框架还必须能够对这些风险所引发的非凸约束进行便于求解的改写，以便实时规划反应式运动。

在本研究中，我们对地形可通行性的评估会考虑崎岖地形带来的危险因素（几何分析与置信度感知分析）以及地形类型（语义分析），见图 1。我们将把可通行性代价视为随机变量，并使用条件风险价值（CVaR）来处理可通行性代价中的不确定性。基于对地形的 CVaR 评估，我们得出一条几何路径，并制定一个考虑地形风险的底层模型预测控制（MPC）方案。

大多数可通行性分析依赖于用于获取地形信息的传感器类型，并基于几何、外观或本体感知方法 [45]。基于几何的方法通常依赖于



图 1：左上：波士顿动力 Spot 四足机器人正在加利福尼亚州拉瓦床国家纪念碑的 Valentine Cave 中探索。中上、下中（从左数第二张图）：Clearpath Husky 机器人正在西弗吉尼亚州贝克利的 Arch Mine 中探索。左下：Spot 正在华盛顿州埃尔马废弃的 Satsop发电厂中探索。

构建 2.5D地形图，以提取地形高度和坡度的最大值、最小值及方差等特征 [28]。此类方法的规划算法会考虑机器人在地形上的稳定性 [29]。在 [43, 27], 中，作者基于车辆的运动学模型和地形高度不确定性估计状态的概率分布。此外，[23]中还探讨了一种将传感器和状态不确定性纳入考虑的方法，以基于栅格的高程地图形式获得概率地形估计。基于学习的可通行性框架正变得越来越受欢迎。在 [26], 中，作者提出了一种基于逆强化学习的方法，通过原始外感受和本体感觉传感器测量值学习可通行性代价。在 [10], 中，作者根据语义信息学习速度地图表示；而在 [39], 中，作者使用基于自监督学习的地形粗糙度分类器。在 [22],中，作者提出了一种神经网络架构，以监督方式利用激光雷达获取的点云，学习由几何风险产生的可通行性代价分布。本文以经典几何方法的可通行性分析为基础，同时利用激光雷达提供的语义信息，并纳入这些方法的不确定性，以进行风险感知规划 [2, 51, 20]。

风险可以通过多种不同的方法纳入运动规划，包括机会约束[42, 52],指数效用函数 [35], 分布鲁棒性 [54, 53], 以及分位数回归 [21, 13]。风险度量常用于金融和运筹学，它将随机变量（通常是代价）映射为一个实数。这些风险指标应满足特定公理，才能得到良好定义，并适用于机器人应用 [38]。条件风险价值（CVaR）就是一种具备这组理想性质的风险度量，属于称为一致性风险度量 的一类风险指标 [7]。一致性风险度量已用于多种决策问题，尤其是马尔可夫决策过程（MDP） [12, 4, 3, 5]。当系统模型存在不确定性时 [49, 15], , 当不确定性源于测量噪声或运动障碍物时 [30, 18], , 以及当不确定性同时来自测量噪声和过程噪声时 [17], 一致性风险度量都已用于模型预测控制框架。我们的工作将CVaR风险扩展到基于风险的规划框架，该框架利用不同来源的可通行性风险（例如碰撞风险、台阶风险、打滑风险等）。此外，本文首次提出了一种经过实地验证且具有理论依据的可通行性评估与基于CVaR指标的风险约束规划方法。利用CVaR评估可通行性风险，只需改变一个值——风险水平——即可动态调整整个系统的行为，使其从激进转为高度保守。当风险水平较低时，可通行性评估的行为等同于基于期望的评估，类似于以往文献中的几何可通行性评估。反之，当风险水平较高时，我们会根据测量中的不确定性，将地形评估取为可能的最差值。

模型预测控制长期以来一直用于对更复杂的系统进行鲁棒控制，包括时变、非线性或多输入多输出系统 [11]。简单的线性PID控制器足以应对较简单的系统，而模型预测控制则适用于更复杂的任务，同时在计算上仍然可行：模型预测控制能够考虑非线性机器人动力学，对机器人将要遵循的轨迹进行多步预测，并纳入复杂的状态约束和控制约束，例如条件风险价值约束、避障约束和非完整约束。因此，模型预测控制提供了一个基于优化的框架，能够在计算上可处理的前提下模拟次优的无限时域控制器。具体而言，我们采用序列二次规划（SQP），通过迭代求解局部二次子问题，收敛到全局（更）优解[9]。这种方法降低了计算代价，并能够灵活处理各种代价项和约束[47, 36, 40, 8, 25]。对基于SQP的模型预测控制方法（以及一般的非线性模型预测控制方法）的一种常见批评是，它们容易陷入局部极小值。我们通过引入轨迹库（可预先定义和/或随机生成，例如 [33]中的方法）来解决这一问题，并将其用于初步的轨迹选择过程。该方法能够找到更接近全局最优的初始猜测，供SQP问题在局部进一步优化。风险约束非线性模型预测控制问题的另一常见难点是确保递归可行性 [37]。我们通过动态放宽风险约束的严格程度，同时在代价函数中惩罚条件风险价值，绕过了这一问题。在本工作中，我们提出STEP（随机可通行性评估与规划），突破当前技术水平的限制，实现对未知环境的安全、风险感知、高速地面通行。具体而言，我们的贡献包括：

1. 不确定性感知的2.5D可通行性评估，综合考虑定位误差、传感器噪声和遮挡，并结合多种可通行性风险来源。2. 一种将这些可通行性风险整合到统一的风险感知条件风险价值规划框架中的方法。3. 一种高效的模型预测控制架构，用于稳健地求解非凸风险约束最优控制问题。4. 一套恢复行为，用于快速应对故障情景。5. 面向四足机器人的基于风险的步态调整（在我们的案例中，机器人平台为波士顿动力 Spot）。6. 在未知且高风险的环境中，针对轮式和足式机器人平台进行实时条件风险价值规划的真实世界演示。

本研究在我们先前的工作 [20] 基础上进行了大幅扩展，除了几何风险来源外，还纳入了语义和置信度感知的可通行性风险来源，从而利用激光雷达提供一套完整的可通行性分析。此外，我们在分层规划框架中提供了恢复行为，以应对几何规划器和运动学-动力学规划器未能考虑的高风险情形。最后，我们还为四足机器人提供了风险感知步态选择技术。我们在多个地下环境中对这一分层框架进行了广泛测试。

2 风险感知的可通行性与规划

2.1 问题陈述

令 x_k、 u_k、 z_k 分别表示机器人在第 k个时间步的状态、动作（或控制输入）和观测（或传感测量）。路径 $x_{0:N}=\{x_0,\ x_1,\ \cdots x_N\}$ 由一系列位姿组成。策略是从状态到控制 $u_k=\pi(x_k)$的映射。地图表示为 $m=(m^{(1)},\ m^{(2)},\ \cdots)$，其中 m^i 是地图中的第 i个元素（例如栅格地图中的一个栅格单元，或基于特征的地图中的一个特征）。机器人的动力学模型描述车辆运动的物理属性，例如惯性、质量、尺寸、形状，以及运动学和控制约束：

$$x_{k+1}=f(x_k,u_k) \tag{1}$$
$$g(u_k)\succ 0 \tag{2}$$
$$h(m,x_k)\succ 0 \tag{3}$$

其中， $g(u_k)$ $h(m,\ x_k)$ 分别为用于编码控制约束/限制和状态约束/限制的向量值函数。

在 [45], 之后，我们将可通行性 定义为地面车辆在满足约束的可行状态下驻留于某个地形区域的能力。我们将可通行性表示为一种代价，即使用地形模型、机器人车辆模型和运动学约束计算得到的连续值，用于表示我们希望机器人避开给定状态的程度：

$$r = \mathcal{R}(m, x, u) \tag{4}$$

其中 $r \in \mathbb{R}$，$\mathcal{R}(\cdot)$ 是一种地形评估模型。该模型涵盖了各种不利事件，例如碰撞、陷车、倾覆和严重打滑等。每种移动平台都有自己的评估模型，以反映其移动能力。

与真实可通行性值相关联的是一个基于当前对环境和机器人动作的认知所形成的可能值分布。在大多数感知能力有限且存在噪声的实际应用中，真实值可能具有很大的不确定性。为处理这种不确定性，考虑一个地图信念，即在可能集合 $\mathcal{M}$ 上的概率分布 $p(m|x_{0:k}, z_{0:k})$。此时，可通行性估计也表示为一个随机变量 $R: (\mathcal{M} \times \mathcal{X} \times \mathcal{U}) \longrightarrow \mathbb{R}$。我们将这种从地图信念、状态和控制量映射到可能的可通行性代价值的概率映射称为地形评估模型。

风险度量 $\rho(R) : R \to \mathbb{R}$ 是一个将随机变量映射为实数的函数，用于量化某种风险。为了评估沿路径 $x_{0:N}$ 按照策略 π行进的风险，我们希望定义与该路径相关联的累积风险度量 $J(x_0, \pi)$。为此，我们需要评估一系列随机变量 $R_{0:N}$。为了将随机结果量化为实数，我们采用通过复合单步风险度量 [46]得到的动态时间一致风险度量：

$$J(x_0, \pi; m) = R_0 + \rho_0\big(R_1 + \rho_1\big(R_2 + ... + \rho_{N-1}\big(R_N\big)\big)\big) \tag{5}$$

其中， $\rho_k(\cdot)$是在时间 k 的单步相干风险度量。该单步风险从时间步 k的角度，给出了时间步 $k+1$ 产生的代价。按照公式（5）复合的任何扭曲风险度量都具有时间一致性（有关扭曲风险度量和时间一致性的更多信息，请参见 [38] ）。我们使用条件风险价值（CVaR）作为单步风险度量：

$$\rho(R) = \mathrm{CVaR}_\alpha(R) = \inf_{z \in \mathbb{R}} \mathbb{E}\left[z + \frac{(R - z)_+}{1 - \alpha} \right] \tag{6}$$

其中，$(\cdot)_+ = \max(\cdot, 0)$，且 $\alpha \in (0, 1]$ 表示风险概率水平。我们注意到，本文的结果也可以很容易地扩展到其他尾部风险度量，例如熵风险价值 [6] 和基于全变差距离的风险 [16]。

我们将该问题的目标表述如下：给定初始机器人构型 x_S 和目标构型 x_G，寻找一个最优控制策略 π^*，使机器人从 x_S 移动到 x_G，同时满足：1）最小化穿越时间；2）最小化沿路径的累积风险度量；3）满足所有运动学约束和动力学约束。对于像 Boston Dynamics Spot机器人这样的四足机器人，该框架还必须根据从 x_S 移动到 x_G 过程中累积的风险，选择最佳步态类型，作为最优策略 π^*的一部分。

2.2 分层风险感知规划

我们提出一种分层方法，通过将运动规划问题拆分为几何域和运动学与动力学域，来解决上述风险感知运动规划问题。我们在较长时域内考虑几何域，同时在较短时域内求解运动学与动力学问题。这样做有以下几个优点：1）在较长时间尺度／时域内求解完整的受约束条件风险价值（CVaR）最小化问题，在实时条件下会变得难以处理。2）在较长时域内，几何约束的作用要大得多；而在较短时域内，运动学与动力学约束的作用要大得多（以确保每个时间步都满足动力学可行性）。3）仅考虑位置信息即可得到良好的风险估计（上界）。具体做法是构建基于位置的可通行性模型 $\mathcal{R}_{\mathrm{pos}}$： 从地形评估模型中边缘化掉与位置无关的变量。也就是说，若状态 $x = [p_x, p_y, x_{\mathrm{other}}]$ 由位置变量和非位置变量（例如方向、速度）组成，则

$$\mathcal{R}_{\mathrm{pos}}(m, p_x, p_y) \geq \mathcal{R}(m, x, u) \quad \forall x_{\mathrm{other}}, u \tag{7}$$

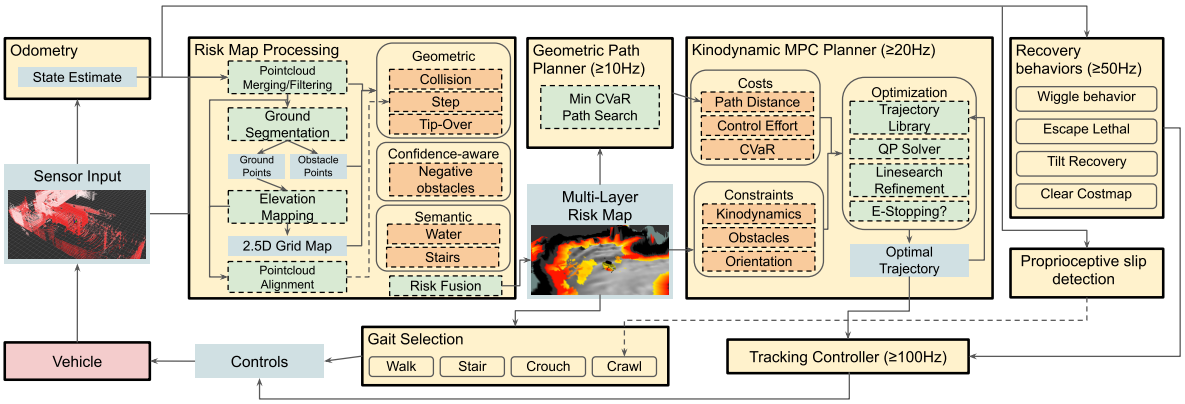

图 2：STEP 系统架构概览。从左到右：里程计汇总传感器输入和相对位姿。接下来，风险地图处理模块融合这些点云并创建多层风险地图。该地图由几何路径规划器和运动学与动力学模型预测控制规划器使用。系统求得一条最优轨迹并将其发送至跟踪控制器，后者生成施加于机器人的控制输入。

几何规划： 几何规划的目标是搜索一条< style id='3'>乐观的 风险最小化路径，即使真实 CVaR 值的上界近似最小化的路径。为提高效率，我们将搜索空间限制在几何域内。我们搜索位姿序列 $x_{0:N}$，该序列以 x_G 结束，并使式（5）中的仅位置风险度量最小化，我们将其定义为 $J_{\text{pos}}(x_{0:N})$。路径优化问题可写为：

$$x^*_{0:N} = \underset{x_{0:N}}{\operatorname{argmin}} \left[J_{\text{pos}}(x_{0:N}) + \lambda \sum_{k=0}^{N-1} \|x_k - x_{k+1}\|^2 \right] \tag{8}$$

$$s.t. \quad \phi(m, x_k) \succ 0 \tag{9}$$

其中，约束 $\phi(\cdot)$ 对位置相关的可通行性约束进行编码（例如约束车辆避开障碍物并禁止进入致命风险等级），而 $\lambda \in \mathbb{R}$ 对风险与路径长度之间的权衡进行加权。动力学规划： 随后，我们求解一个运动学动力学规划问题，以跟踪最优几何路径、最小化风险度量，并遵守运动学与动力学约束。目标是在局部规划时域 $T \leq N$ 内找到一个控制策略π^*，使其跟踪路径 $X^*_{0:N}$。最优策略可通过求解以下优化问题获得：

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}} \left[J(x_0, \pi) + \lambda \sum_{k=0}^{T} \|x_k - x^*_k\|^2 \right] \tag{10}$$

$$s.t.\ \forall k \in [0, \cdots, T]: \qquad x_{k+1} = f(x_k, u_k) \tag{11}$$

$$g(u_k) \succ 0 \tag{12}$$

$$h(m, x_k) \succ 0 \tag{13}$$

其中，约束 $g(u)$ 和 $h(m, x_k)$ 分别是对控制器限制和状态约束进行编码的向量值函数。

不利状态恢复： 恢复规划是一类特殊的规划问题，其中初始状态并不安全。例如，机器人可能需要从保险杠接触墙壁的状态开始规划，或者从机身倾斜在瓦砾顶部的状态开始规划。从这些不利状态中恢复，需要在较小裕度条件下，在不违反安全约束的情况下找到安全控制。

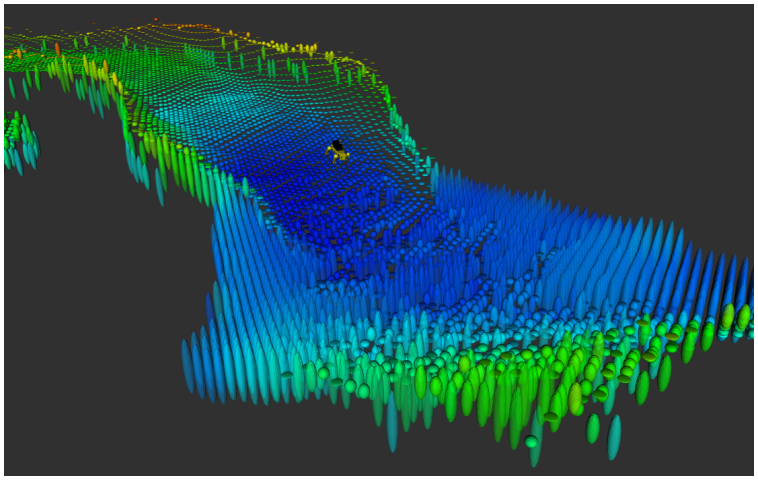

图 3：带不确定性的地面高度估计

3 STEP：面向非结构化地形

本节将讨论如何计算可通行性风险，以及如何高效求解风险感知轨迹优化问题。总体而言，我们的方法包括以下步骤（见图 2）：1）使用某种带有不确定性的定位来源，汇总传感器测量值，以创建不确定性感知地图；2）执行地面分割，以分离出地图中机器人可能通行的部分；3）利用点云的几何属性计算风险和风险不确定性（可选地纳入其他风险来源，例如语义信息或其他传感器）；4）汇总这些风险，计算 2.5D CVaR风险地图；5）使用几何路径规划器，在较长范围内求解使 CVaR 最小化的乐观路径；7）求解一条运动学-动力学可行轨迹，在满足所有约束的同时，使 CVaR 最小化并保持接近几何路径。

3.1 点云处理与建图

多传感器融合： 我们的点云处理流程从融合不同传感器的点云开始。一台机器人可以配备多个同类型传感器以提高覆盖度，也可以配备利用不同机制生成点云的异构传感器（例如主动式激光雷达、RGB-D相机、被动双目相机）。应用用于去除噪声或机体遮挡影响的传感器专用滤波器后，我们利用外参标定信息融合这些点云。如果传感器未实现时间同步，我们会使用里程计补偿运动偏移。

时序点云融合： 我们将融合后的点云在固定时间窗口内聚合，以构建局部点云地图。我们在该时间窗口内维护一个位姿图，以纳入里程计估计中的历史更新（例如回环闭合）。根据最新的位姿图，我们重建完整的聚合点云。我们为每个点标注其与最新点云之间的时间偏移。这样便可以在融合阶段将里程计不确定性传播到每个点。

地面分割： 我们采用三维点云分割技术，将聚合点云分割为地面点和障碍物点。我们借鉴了 [31] 中的工作，该方法基于柱坐标中的直线拟合实现高效的地面分割。我们对该方法进行了扩展，使其也能应对地下环境中常见的挑战，例如低矮天花板或负障碍物。在遮挡环境中，地面可能无法被传感器观测到，而墙壁和天花板的测量结果会形成虚假地面，因此地面分割对于高程建图至关重要。

高程建图： 我们使用分割后的地面点构建2.5D高度图。将点分配到各个栅格地图单元并按时间戳排序后，我们应用卡尔曼滤波器估计地面高度。我们对滤波器进行设置，使其更加重视近期测量值，因为这些测量值受里程计误差的影响较小。图3展示了带有不确定性的地面高度估计可视化结果。请注意，测量值较少的区域具有更高程度的不确定性。这些建图不确定性将用于在后续的可通行性估计过程中调整置信度。

3.2 建模假设

在崎岖地形的众多表示选项中，本文出于处理和数据存储效率的考虑，采用2.5D 网格地图 [24]。该地图表示为均匀网格上地形属性（例如高度、风险）的集合。

对于不同的车辆，我们采用不同的机器人动力学模型。例如，对于能够产生纵向/横向速度并进行转向的系统（如足式平台），可将状态和控制量定义为：

$$x = [p_x, p_y, p_\theta, v_x, v_y, v_\theta]^\intercal \tag{14}$$

$$u = [a_x, a_y, a_\theta]^\intercal \tag{15}$$

动力学模型 $x_{k+1} = f(x_k, u_k)$ 可写为：

$$x_{k+1} = x_k + \Delta t \begin{bmatrix} v_x \cos(p_\theta) - v_y \sin(p_\theta) \\ v_x \sin(p_\theta) + v_y \cos(p_\theta) \\ \kappa v_x + (1-\kappa) v_\theta \\ a_x \\ a_y \\ a_\theta \end{bmatrix}.$$

我们令 $\kappa \in [0,1]$ 为一个常数，用于调整允许车辆原地转向的幅度。对于差速驱动或阿克曼转向车辆，可以从这些动力学模型中去掉横向速度分量。不过，我们的通用方法适用于任何车辆动力学模型。（差速驱动模型参见附录 A）

3.3 地形评估模型

可通行性代价由多个风险因素综合评估得出。这些因素旨在捕捉特定环境中目标机器人可能遇到的危险因素（图 4），包括：

- 碰撞：通过与最近障碍物点的距离来量化。
- 台阶高度：栅格地图中相邻单元之间的高度差。
- 倾覆：取决于坡度角和机器人的方向。
- 传感器不确定性：传感器误差和定位误差会增加可通行性估计的方差。
- 负障碍物：通过检查栅格单元中是否缺少测量点来检测。
- 打滑：根据地面的几何形状和表面材料进行量化。

3.3.1 基于几何的风险源

基于几何的风险源包括与障碍物碰撞、地形台阶高度过大以及不可通行的坡度。这些基于几何的风险通过对高程地图估计和激光雷达点云点进行几何分析构建。我们使用以下方法，针对每个网格单元构建这些风险。首先，利用地面分割流程得到的地面估计，确定地面高度的下界，即机器人占据该栅格单元时，其足部或车轮所处的高度。以此地面估计为基准，我们可以确定时间合并点云中其他激光雷达点相对于地面的高度。对于高于或低于某一高度阈值（该阈值界定了机器人本体范围）的点，将其视为障碍物，并将这些点对应的网格单元标记为不可通行。台阶高度风险也采用类似分析，通过检查高程图中相邻单元之间的高度差来评估。对于高度差过大的相邻单元，将其标记为不可通行。最后，通过检查高程图的法向量 [24] ，可以整体估算坡度（法向量计算的范围按机器人大致尺寸取平均）；坡度值过高的区域则标记为高风险或不可通行。

请注意，在所有这些分析中，高程图的不确定性都起着重要作用。传感器和定位不确定性会以不同程度影响高程测量值，且影响程度与机器人之间的距离成正比。因此，我们根据与机器人的距离调整各种检测阈值或风险阈值，以获得更稳健的

结果。例如，对于 Husky机器人，机器人底盘下方的离地间隙为 10 厘米。要在 100 米外检测到 10 厘米的高程台阶，需要达到 < 0.06 度的角度定位精度。这可能不可行，因此根据距离调整台阶高度阈值有助于减少较远距离处的误报。

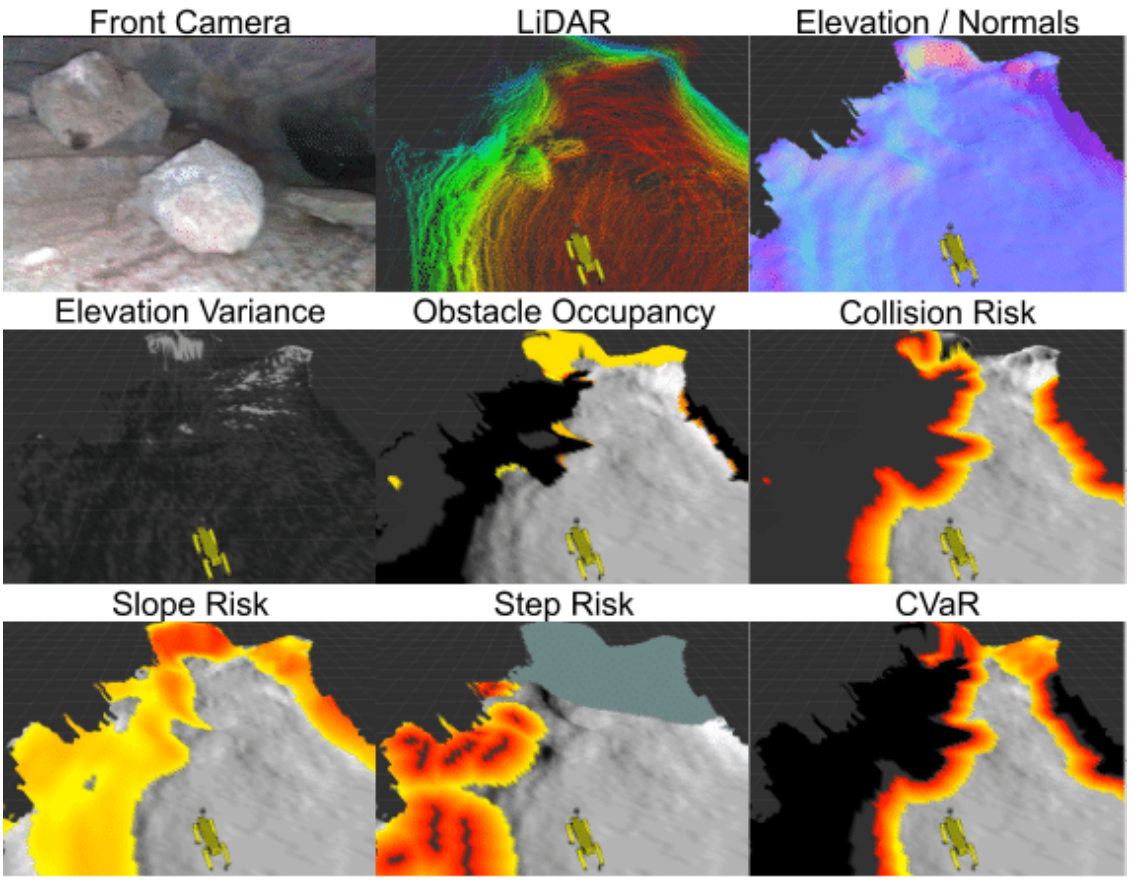

图4：多层几何风险分析，首先汇总近期的点云（上）。随后，每种类型的分析（坡度、台阶、碰撞等）都会生成风险图及其不确定性（中间各行）。最后汇总这些风险，计算出最终的条件风险价值（CVaR）地图（下）。

3.3.2 基于置信度的风险源

负障碍物检测是基于置信度的风险源的一个示例。为检测负障碍物，我们估计点云中没有回波的区域，并使用二维光线追踪找出未被障碍物、小台阶或上坡遮挡的“间隙”区域。然而，如果仅以此作为检测负障碍物的标准，就会出现许多负障碍物假阳性。图5展示了一个例子：机器人从狭窄通道转弯进入一间新房间时，在新房间各处的激光雷达回波到达传感器所需的时间内，检测到了负障碍物假阳性。为解决这一问题，我们通过估计点云中的这些间隙区域迄今是否已得到充分覆盖，来考虑对这些间隙的置信度；也就是说，机器人是否已从不同位置感知该区域，并持续足够长的时间，以确保点云中的间隙并非由传感器和环境配置造成。有关检查给定区域是否通过覆盖检查的方法的详细说明，请参见算法1。

3.3.3 基于语义的风险源

在与传感器距离相同的情况下，不同材料的激光雷达强度回波也不同。在 [32], 中，作者展示了利用激光雷达强度和高度回波进行地形分类的有效性。我们根据激光雷达回波强度的变化来考虑地形特征的变化。由大量泥浆和水构成的区域，其回波强度较低。这使得机器人能够检测泥泞区域

算法1用于计算给定区域是否未被遮挡且覆盖充分的算法，即，$m_{\mathrm{MSE}}^{\mathrm{cover}} >= 0$

输入： m^{unocc}：根据当前机器人状态 x_t （使用光线追踪）生成的未遮挡网格点栅格地图，$x_t = [p_x, p_y, p_\theta, v_x, v_y, v_\theta]^T$：当前机器人状态，$d^{\mathrm{cover}}$：达到充分覆盖所需的阈值（单位：米）。输出： m_N^{unocc}：截至目前，机器人观测到栅格地图中每个网格单元未被遮挡的次数，$m^{\mathrm{cover}} = \{m_{\mu_x}^{\mathrm{cover}}, m_{\mu_y}^{\mathrm{cover}}, m_{\mathrm{MSE}}^{\mathrm{cover}}\}$：多层地图，用于指示正在评估是否存在负障碍物的每个网格单元是否已从不同位置被观测为未遮挡，以确保达到充分覆盖。1：对于 $i=0$到 $|m^{\mathrm{unocc}}|$执行 2：# 更新 $m_N^{unocc} m^{\mathrm{unocc}}, {}_N{}^{(i)} = \begin{cases} 1 + m^{\mathrm{unocc}}, {}_N{}^{(i)} \\ m_N^{\mathrm{unocc}\ (i)} \end{cases}$ 若 $m^{\mathrm{unocc}}, {}^{(i)} = 1$（即 i^{th}网格单元未被遮挡），若 $m^{\mathrm{unocc}}, {}^{(i)} = 0$（即 i^{th}网格单元被遮挡）3：# 更新观测到每个网格单元未被遮挡的位置的平均值 x 和 y 机器人位置 $\{m^{\mathrm{cover}}, {}_{\mu_x}{}^{(i)}, m^{\mathrm{cover}}, {}_{\mu_y}{}^{(i)}\} = \begin{cases} \{m^{\mathrm{cover}}, {}_{\mu_x}{}^{(i)} + {}^{p_x - m}\text{覆盖}, \ ({}_i)_m{}^{\mu_x}\text{未遮挡}, \ ({}_i)_N, m^{\mathrm{cover}}, {}_{\mu_y}{}^{(i)} + {}^{p_y - m}\text{覆盖}, \ ({}_i)_m{}^{\mu_y}\text{未遮挡}, \\ ({}_i)_N\} \text{若 } m^{\mathrm{unocc}}, {}^{(i)} = 1 \{m^{\mathrm{cover}}, {}_{\mu_x}{}^{(i)}, m^{\mathrm{cover}}, {}_{\mu_y}{}^{(i)}\} \text{若 } m^{\mathrm{unocc}}, {}^{(i)} = 0 \end{cases}$ 4：# 更新观测到给定网格单元时机器人位置均值的均方偏差，以检查其是否超过充分覆盖阈值， $d^{cover} m^{\mathrm{cover}}, {}^{(i)}$ 均方误差 $= \begin{cases} (\\ p_x - m_{\mu_x}^{\mathrm{cover}\ (i)\ 2} + p_y - m_{\mu_y}^{\mathrm{cover}\ (i)\ 2} - d^{\mathrm{cover}\ 2} \quad m^{\mathrm{unocc}\ (i)} = 1,) \ (,) \ ( ) \text{若}, (d^{\mathrm{cover}}) \text{若 } m^{\mathrm{unocc}}, {}^{(i)} = 0 \end{cases}$ 5：结束循环

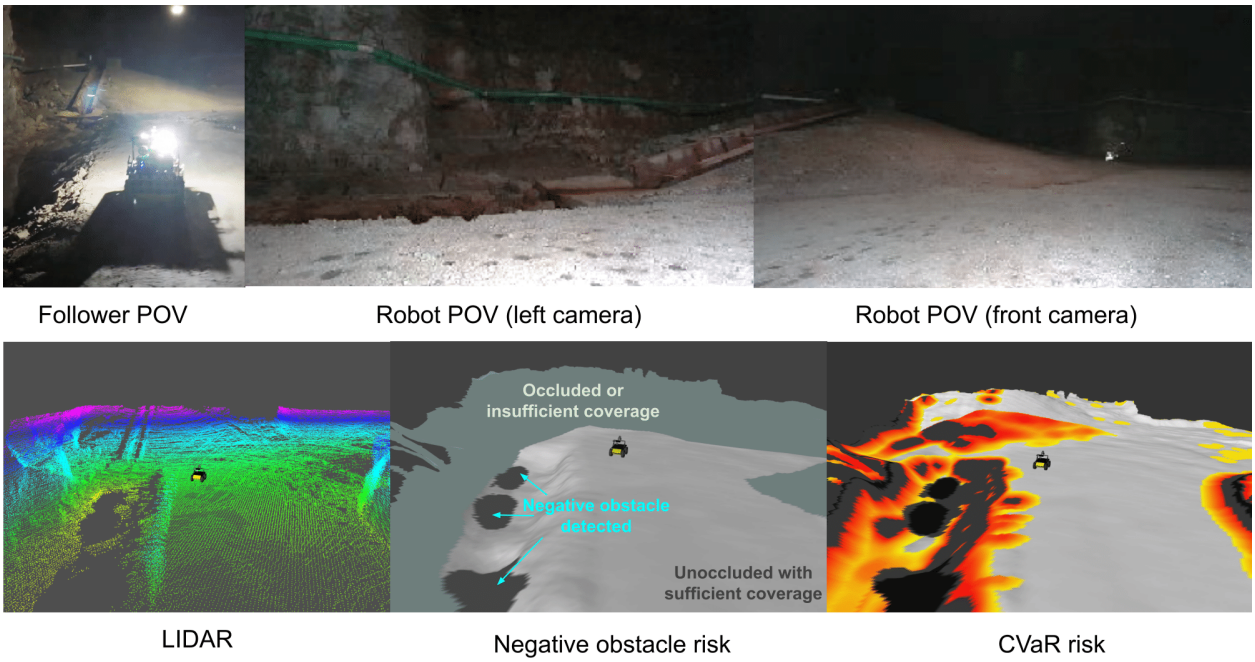

图5：基于置信度的风险分析：场景通过第三人称视角（POV）（左上）和机器人视角（上方中间和右侧）进行展示。聚合点云（左下）中，机器人左侧区域存在无回波的区域。只有当这些区域未被遮挡且受到激光照射模式充分覆盖，但仍然没有回波时，才会将点云中的这些空洞在风险层（下方中间）中标记为负障碍物。该风险层与几何风险层聚合，以计算最终的条件风险价值地图（右下）。

可能会陷住或摔倒。深水水位区域同样会被检测为负障碍物，因为激光雷达回波非常稀疏。值得注意的是，结合负障碍物和低回波强度，我们能够检测出深水水位的致命区域。图6展示了机器人进行水体检测的一个实例，此时

机器人靠近一个水坑。需要注意的是，相反，在图5中，虽然激光雷达点云中存在空洞，但这些空洞周围并没有低强度回波，因此被归类为负障碍物。

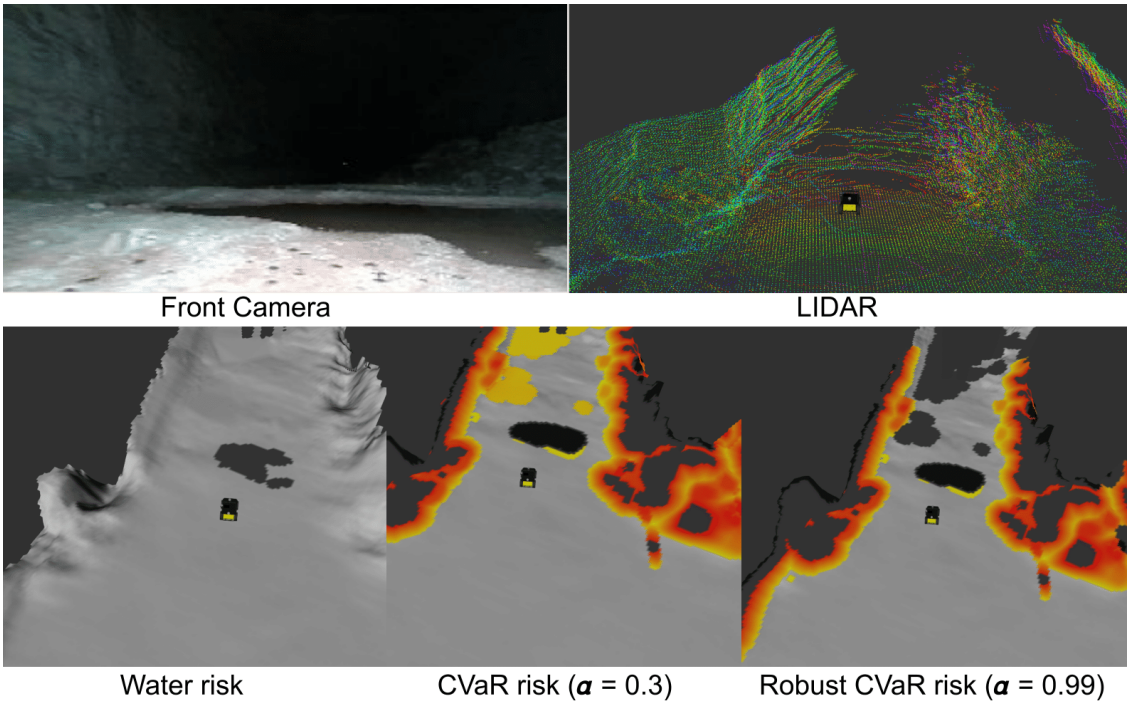

图6：基于语义的风险分析：机器人摄像头（左上）显示机器人前方有一滩水。聚合点云（右上）中，机器人前方区域存在无回波的区域。当点云中的这些空洞附近出现低强度回波时，风险层（左下）会将这些点云空洞标记为水。随后，将此风险层与几何风险层聚合，以计算最终的条件风险价值地图（中下和右下），并呈现不同程度的鲁棒性 α。我们可以清楚地看到这对风险图的影响： α 值越高，未知区域被赋予的代价越高。

3.4 可通行性代价

为高效计算 $l>1$ 风险因素的条件风险价值（CVaR）可通行性代价，我们假设每个风险因素 R_l 都是相互独立且服从正态分布的随机变量，其均值为 μ_l，方差为σ_l。我们对各风险因素加权求平均，得到地形评估模型 $R=\sum_l w_l R_l$；该模型也服从正态分布，表示为 $R\sim\mathcal{N}(\mu,\ \sigma^2)$。令 φ 和 $\mathbf{\Phi}$ 分别表示标准正态分布的概率密度函数和累积分布函数。相应的条件风险价值计算如下：

$$\rho(R)=\mu+\sigma\frac{\varphi(\mathbf{\Phi}^{-1}(\alpha))}{1-\alpha} \tag{16}$$

条件风险价值代价考虑地形评估模型分布尾部（$1-\alpha$）中的预期代价，即 R，因此也考虑了高风险低概率事件。我们构造 R，使 R的期望为正，从而保持条件风险价值为正，$\mu\geq 0$。

每种风险因素的均值和方差的构建方式取决于风险的类型。例如，碰撞风险通过检查高程图上方是否存在点来确定，其方差则根据高程图方差推导得出，而高程图方差主要取决于定位误差。相比之下，负障碍物风险通过查找传感器测量值中的空隙来确定。这些空隙通常与传感器稀疏性有关，因此，距离传感器坐标系越远，风险方差越大。

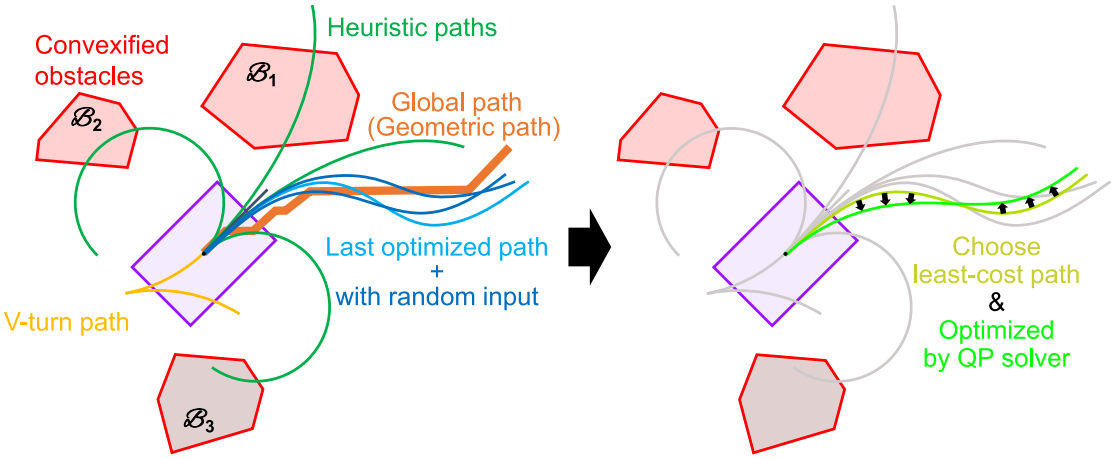

图 7：运动学与动力学模型预测控制规划器示意图。该规划器首先评估轨迹库中的各种路径，选择代价最低的路径作为候选项，并通过二次规划求解器进行优化。

3.5 风险感知几何规划

为优化式 (8) 和式 (9)，几何规划器计算一条最优路径，使沿路径的位置相关动态风险度量 (5) 最小。将式 (16) 代入式 (5)，可得：

$$J_{\text{pos}}(x_{0:N}) = \mu_0 + \sum_{k=1}^{N} \left[\mu_k + \sigma_k \frac{\varphi(\mathbf{\Phi}^{-1}(\alpha))}{1-\alpha} \right] \tag{17}$$

（证明见附录B。）
我们使用 A* 算法在二维网格上求解（8）。A* 需要路径代价 g（n）和可采纳启发式代价 h（n），其定义如下：

$$g(n) = J_{\text{pos}}(x_{0:n}) + \lambda \sum_{k=0}^{n-1} \|x_k - x_{k+1}\|^2, \tag{18}$$

$$h(n) = \lambda \|x_n - x_G\|^2. \tag{19}$$

对于启发式代价，我们采用到目标点的最短欧几里得距离，该距离始终小于实际路径代价。参数 lambda 表示距离惩罚与风险惩罚之间的相对权重，其单位可理解为（可通行性代价 / 米）。我们的实验采用了一个相对较小的值，这意味着我们主要关注最小化可通行性代价。

3.6 风险感知运动学-动力学规划

几何规划器生成一条路径，即一系列位姿。我们希望找到一条运动学-动力学可行轨迹，使其始终靠近这条路径，同时满足所有约束并最小化条件风险价值代价。为求解式（10）—（13），我们使用风险感知运动学与动力学模型预测控制规划器，其步骤概述如下（图7、算法2、附录C）。轨迹库：我们的运动学与动力学规划器首先从轨迹库中选择最佳候选轨迹。该轨迹库存储了多组初始控制与状态序列。所选轨迹将作为求解完整优化问题的初始解。轨迹库可以包含：1）在上一次规划迭代中接受的轨迹；2）停止（制动）轨迹；3）沿几何规划的轨迹；4）启发式定义的轨迹（包括 V 形转弯、U 形转弯和不同曲率的轨迹）；以及 5）随机扰动的控制输入序列。

二次规划优化：接下来，我们构建一个包含代价项和约束的非线性优化问题（10–13）。我们在初始解附近对问题进行线性化，并以序列二次规划（SQP）的方式迭代求解 [41]。令 $\{\hat{x}_k, \hat{u}_k\}_{k=0:T}$ 表示初始解。令 $\{\delta x_k, \delta u_k\}_{k=0:T}$ 表示偏差

算法2 运动学与动力学模型预测控制规划器（为简洁起见，序列 {var$_k$}$_{k=0:T}$ 表示为 {var}） 输入：当前状态 x_0，当前控制序列（先前的解）{u^*}$^{(j)}$ 输出：重新规划的轨迹 {x^*}$^{(j+1)}$，重新规划的控制序列 {u^*}$^{(j+1)}$# 初始化 1：{x^r}= updateReferenceTrajectory() 2：{u^*}$^{(j)}$ = stepControlSequenceForward({u^*}$^{(j)}$) # 循环过程 3：对 $i=0$到 qp_次迭代执行 4：l =generateTrajectoryLibrary(x_0) 5：[{x^c},{u^c}]= chooseCandidateFromLibrary(l) 6：[{δx^*},{δu^*}]= solveQP({x^c},{u^c},{x^r}) 7：[γ,$solved$]= lineSearch({x^c},{δx^*},{u^c},{δu^*}) 8：$u_k^c = u_k^c + \gamma \delta u_k^*, \forall k = 0$: T9：{x^c}= rollOutTrajectory(x_0,{u^c}) 10：结束循环 11：如果已求解 12：{x^*}$^{(j+1)}$,{u^*}$^{(j+1)}$ ={x^c},{u^c} 13：否则 14：{x^*}$^{(j+1)}$,{u^*}$^{(j+1)}$ = getStoppingTrajectory() 15：结束判断 16：返回 {x^*}$^{(j+1)}$,{u^*}$^{(j+1)}$

初始解。我们引入解向量变量 X：

$$X = \begin{bmatrix} \delta x_0^{\mathrm{T}} & \cdots & \delta x_T^{\mathrm{T}} & \delta u_0^{\mathrm{T}} & \cdots & \delta u_T^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{20}$$

随后，我们可以将（42–45）写成以下形式：

$$\text{minimize} \quad \frac{1}{2} X^{\mathrm{T}} P X + q^{\mathrm{T}} X \tag{21}$$

$$\text{subject to} \quad l \leq AX \leq u \tag{22}$$

其中， P 是一个半正定权重矩阵， q 是用于定义目标函数一阶项的向量， A 定义不等式约束，而 l 和 u 分别给出其下限和上限。（见附录 E。）下一小节（第 3.6.1 节）将详细介绍这些代价项和约束。这是一个二次规划问题，可以使用常见的二次规划求解器进行求解；例如，我们的实现使用 OSQP 求解器。OSQP 是一种稳健且高效的通用凸二次规划问题求解器 [50]。

线搜索： 序列二次规划问题的解返回控制序列 $\{\delta u_k^*\}_{k=0:T}$ 的一个优化变体。然后，我们使用线搜索过程确定要添加到当前候选控制策略 π 中的偏差量 $\gamma > 0$： $u_k = u_k + \gamma \delta u_k^*$。（见附录 F。）

停止序列：如果线搜索未找到可行的良好解，我们就从轨迹库中选取一条无碰撞且代价最低的轨迹。如果所有轨迹都会发生碰撞，我们就生成一条紧急停止序列，尽可能减缓机器人速度（虽然可能发生碰撞，但希望碰撞时的能量尽可能小）。

跟踪控制器：找到一条可行且能最小化条件风险价值的轨迹后，我们将其发送给跟踪控制器，以较高频率（>100 Hz）生成闭环跟踪行为。跟踪控制器根据机器人类型而定（例如，简单的级联PID或足式运动控制器）。

3.6.1 优化代价项与约束

代价项：请注意，代价（10）包含条件风险价值。为了在线性化代价并将其表示为二次规划，我们计算 ρ 关于状态 x 的雅可比矩阵和海森矩阵。我们通过数值微分高效地近似这些项。

运动学与动力学约束：与代价类似，我们针对 x 和 u 对系统动力学（11）进行线性化。根据动力学模型的不同，也可以通过解析方法完成。

控制限制： 式（12）中的约束函数 $g(u)$ 限制了控制输入的范围。例如，在 6 状态动力学情况下，我们限制最大加速度：$|a_x| < a_x^{\max}$、 $|a_y| < a_y^{\max}$ 和 $|a_\theta| < a_\theta^{\max}$。

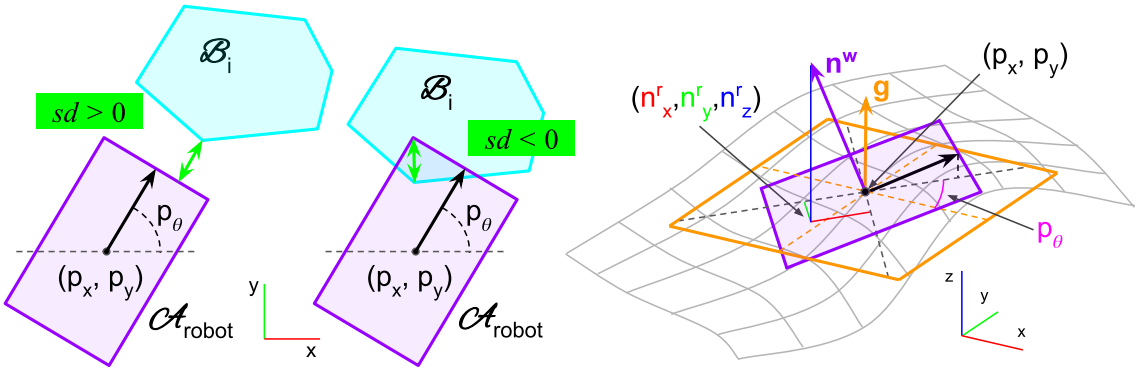

图 8：左：计算机器人足迹与障碍物之间凸多边形到凸多边形的有符号距离函数。没有相交时，有符号距离为正；相交时为负。右：根据经机器人偏航角旋转后的表面法线计算机器人俯仰角和横滚角。紫色矩形表示机器人足迹，其表面法线为 n^w。 $\boldsymbol{g}$ 表示重力向量，$n^r_{x、y、z}$ 是用于计算俯仰角和横滚角的机器人坐标系表面法线分量。

状态限制： 在式（13）的约束函数 $h(m、x)$ 中，我们编码了速度约束： $|v_x| < v_x^{\max}$、$|v_y| < v_y^{\max}$ 和 $|v_\theta| < v_\theta^{\max}$。我们还将车辆的速度限制为低于该区域风险的某个标量倍数，同时设置最大允许速度：

$$|v_\theta| < \gamma_\theta \rho(R_k) \tag{23}$$

$$\sqrt{v_x^2 + v_y^2} < \gamma_v \rho(R_k) \tag{24}$$

这一约束会在风险较高的情况下降低机器人的动能，从而避免造成更严重的损害。
位置风险约束：函数 $h(m、x_k)$ 还对机器人的位置和方向施加约束，以防止机器人撞上障碍物。这一约束的一般形式为：

$$\rho(R_k) < \rho^{\max}. \tag{25}$$

为构建约束 (25)，我们在地图上找出风险 ρ 高于最大允许风险的区域。这些区域被标记为障碍物，且通常高度非凸。为了获得对这一高度非凸约束的凸且易于处理的近似，我们将障碍物分解为互不重叠的二维凸多边形，并创建一个有符号距离函数，用于确定机器人足迹（同样假设为凸多边形）与各障碍物 [47] 之间的最小距离。设 $\mathcal{A}$、$\mathcal{B} \subset \mathbb{R}^2$ 为两个凸集，并将它们之间的距离定义为：

$$\mathrm{dist}(\mathcal{A},\mathcal{B}) = \inf\{\|T\| \mid (T+\mathcal{A}) \cap \mathcal{B} \neq \emptyset\} \tag{26}$$

其中 T 是一次平移。当两个集合重叠时，将穿透距离定义为：

$$\mathrm{penetration}(\mathcal{A},\mathcal{B}) = \inf\{\|T\| \mid (T+\mathcal{A}) \cap \mathcal{B} = \emptyset\} \tag{27}$$

然后，我们可以将两个集合之间的带符号距离定义为：

$$\mathrm{sd}(\mathcal{A},\mathcal{B}) = \mathrm{dist}(\mathcal{A},\mathcal{B}) - \mathrm{penetration}(\mathcal{A},\mathcal{B}) \tag{28}$$

We 然后，在函数 $h(m,x_k)$ 中加入一个约束，以确保满足以下不等式 y:

$$\mathrm{sd}(\mathcal{A}_{\mathrm{robot}},\mathcal{B}_i) > 0 \quad \forall i \in \{0,\cdots,N_{\mathrm{obstacles}}\}. \tag{29}$$

请注意，机器人足迹$\mathcal{A}_{\mathrm{robot}}$ 取决于当前机器人位置和方向：$\mathcal{A}_{\mathrm{robot}}(p_x,p_y,p_\theta)$；而每个障碍物$\mathcal{B}_i(m)$则取决于地图信息（见图 8）。
方向约束： 为了约束机器人在倾斜地形上的方向，防止机器人

翻滚或执行危险动作，我们在函数 $h(m,\ x_k)$ 中添加约束，以限制机器人在地面表面稳定下来时的横滚角和俯仰角。将机器人的位置记为 $p=[p_x,p_y]$，将位置/偏航角记为 $s=[p_x,p_y,p_\theta]$。令 ψ和 ϕ分别表示机器人在机体坐标系中的俯仰角和横滚角。令$\omega=[\psi,\phi]$。约束形式为 $|\omega|\prec\omega^{\max}$。令点 p处的表面法向量在世界坐标系中表示为 $n^w=[n_x^w,n_y^w,n_z^w]$。令 $n^r=[n_x^r,n_y^r,n_z^r]$ 表示机体坐标系中的表面法向量，其中按机器人的偏航角进行旋转：$n^r=R_\theta n^w$（见图 8），其中旋转矩阵 R_θ表示绕世界坐标系 z 轴旋转角度 θ。然后，我们将机器人的俯仰角和横滚角定义为 $\omega=g(n^r)$，其中：

$$\omega=g(n^r)=\begin{bmatrix}\text{atan2}(n_x^r,n_z^r)\\-\text{atan2}(n_y^r,n_z^r)\end{bmatrix} \tag{30}$$

创建线性　　约束优化问题需要对约束进行线性近似　　t:

$$|\nabla_s\omega(s)\delta s+\omega(s)|<\omega^{\max} \tag{31}$$

线性化通过分别求取方向约束相对于位置和偏航角的梯度来实现（见附录 D）。

盒约束：请注意，如果 δx 和 δu 过大，线性化误差将占主导。为减轻这一影响，我们还在 (12) 和 (13) 中加入盒约束，以使解相对于初始解的偏差保持有界：$|\delta x|<\epsilon_x$和 $|\delta u|<\epsilon_u$。

添加松弛变量：为进一步提高优化问题的可行性，我们针对状态限制、位置风险和方向约束引入辅助松弛变量。对于给定约束 $h(x)>0$，我们引入松弛变量 ϵ，并将约束修改为 $h(x)>\epsilon$和 $\epsilon<0$。随后，我们通过二次代价惩罚较大的松弛变量：$\lambda_\epsilon\epsilon^2$。这些松弛变量被纳入二次规划问题 (21) 和 (22) 中。

3.7 动态风险调整

条件风险价值指标允许我们动态调整愿意接受的风险水平和严重程度。选择较低的 α 会回归使用平均代价作为指标，从而做出乐观的决策，同时忽略低概率但高代价的事件。相反，选择较高的α则倾向于采取保守策略，在减少致命事件可能性的同时，也会缩小可行路径的集合。我们根据以下两个标准调整 α：1）任务级状态：根据机器人的角色，或环境与机器人能力之间的平衡，不同机器人可能采取不同的风险态势。2）恢复行为：如果机器人被困在不利条件下，通过逐步降低 α，可以以最小风险找到逃脱方案。这些启发式方法在风险感知规划中尤其有用，因为在线非线性模型预测控制的可行性难以保证。当针对给定风险水平 α 找不到可行解时，可以快速找到并执行一个风险更高但可行的解决方案。

3.8 风险驱动的步态选择

我们选用两种平台来测试可通行性框架：Clearpath Husky（轮式机器人）和波士顿动力 Spot（四足机器人）。对于 Spot，有多种运动步态可供选择，分别是：行走步态 （标准操作，最稳定）、爬楼梯步态 （降低机器人速度，并调整机器人的俯仰角，以便在下楼梯时更好地观察楼梯）、爬行步态 （始终有三只脚接触地面，是最稳定的步态）[1]。此外，还可以动态调整机器人的高度；此后，将为适应狭小空间而降低高度的动作称为蹲伏步态。这些运动步态使 Spot 能够在采用标准行走步态时无法通行的区域移动。

- **楼梯步态选择：**通过对激光雷达和相机视差点云数据进行几何分析来检测楼梯。为检测楼梯，我们采用了与 [48] 类似的平面拟合方法。我们计算法向量并进行聚类，以识别点云中坡度相似且彼此相连的表面。随后，我们分离出水平面和垂直面，以检测潜在的楼梯表面。找到候选项楼梯表面后，确定楼梯的主要方向和整体平面。最后，利用机器人当前所在的地面与楼梯平面的交线来确定楼梯的起点。确定楼梯起点（向上或向下）后，机器人即可调整自身位置对准该处，并开始采用楼梯行走步态。

[1]有关这些步态的更多说明，请参见此处：https://support.bostondynamics.com/s/article/ Operating-Spot

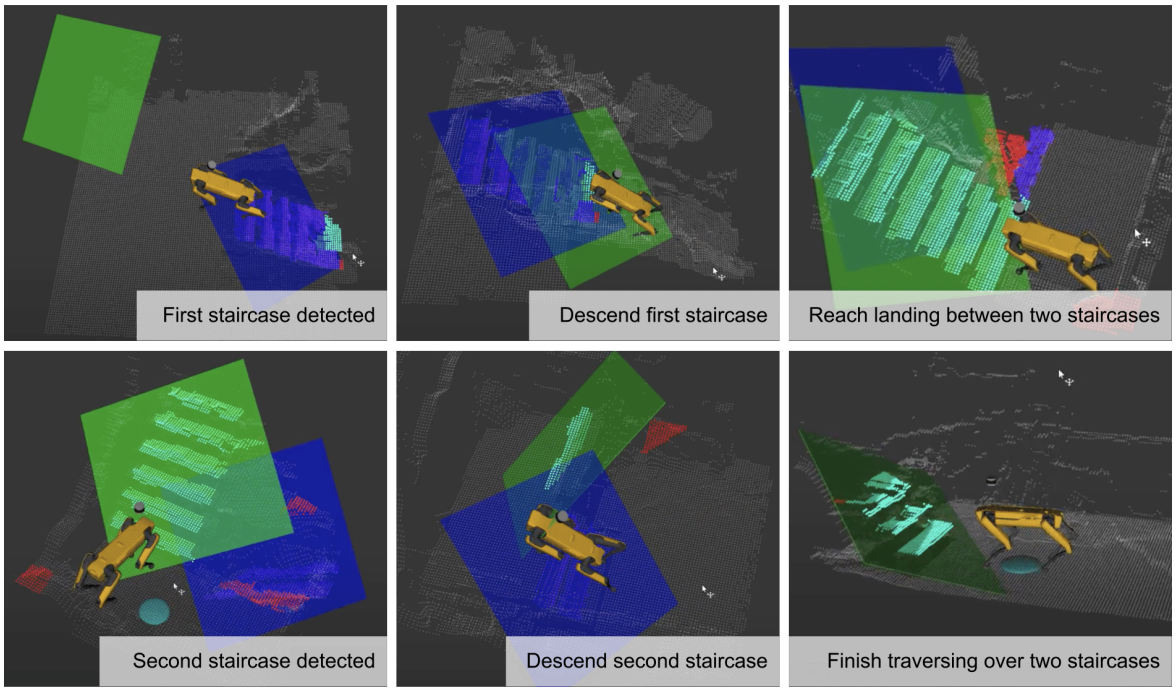

图9：通过平面拟合方法进行楼梯检测

- **通过滑移检测实现蹲伏步态：**在 [14] 中实现了一个基于学习的滑移预测模块，该模块利用四足机器人每个时刻的关节状态信息。我们训练了一个集成模型，根据机器人肢体运动学和动力学，预测每个时刻发生滑移的概率， $p(slip)$。多个现场试验中采集的、涵盖不同地形（例如岩石、沙地和泥泞地形）的滑移标注构成了模型训练的真实标注。使用训练好的模型进行滑移预测时，如果 $p(slip) > 0.5$，则启用机器人的爬行步态模式，以通过可能湿滑的地形。有关滑移预测模型实现的更多详情，请参见 [14]。

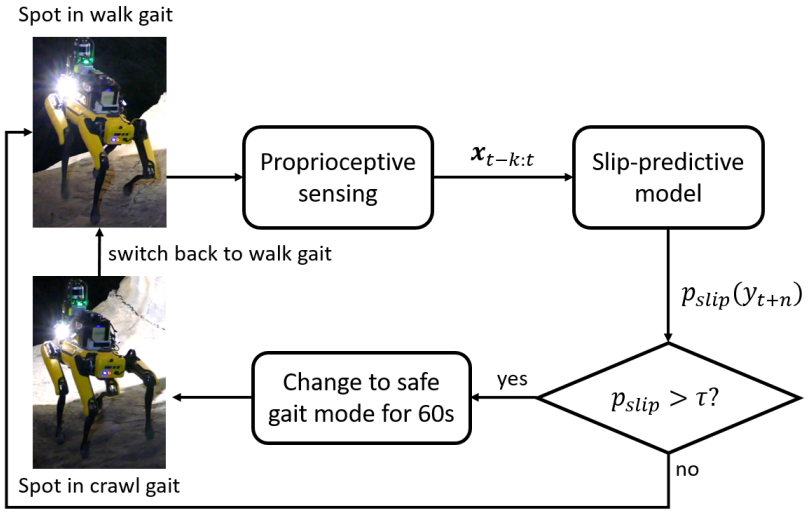

图10：通过滑移检测进行步态切换

- **低矮天花板区域中的蹲伏行为：**如果机器人检测到一个低矮顶棚区域，且顶棚高度与机器人高度相差不超过 16 厘米（这是机器人能够弯曲的极限），机器人就会启用蹲伏步态，使机器人的高度降低 16 厘米。机器人通过低矮顶棚区域后，会切换

回到行走步态。

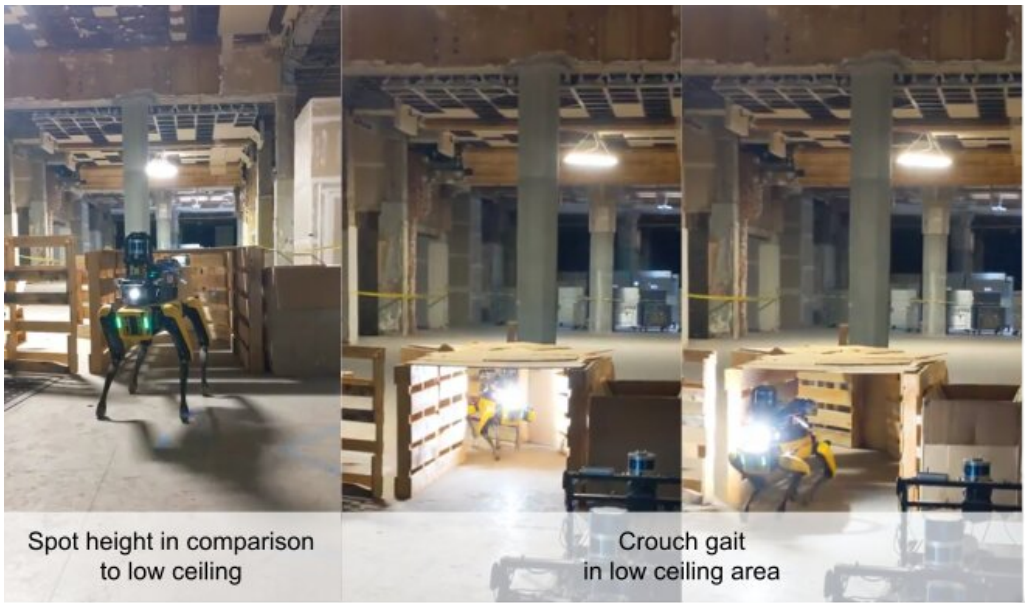

图 11：低矮天花板区域中的蹲伏行为

3.9 恢复行为

恢复行为可确保机器人针对危险且可能导致任务终止的情况，快速采取反应性动作，从而快速、安全地恢复。在激活恢复行为时，机器人会暂时停止遵循几何与运动学规划流水线发出的命令，以进入安全状态，并从该状态恢复分层规划流水线（参见图 2）。

倾斜恢复： 倾斜恢复行为会跟踪轮式机器人从状态估计流水线接收的里程计数据。里程计数据由轮式里程计和惯性测量组成，包含位姿和速度测量。如果机器人的俯仰角超过阈值，机器人会沿着导致其进入危险俯仰角状态的同一路径，以低速倒退。

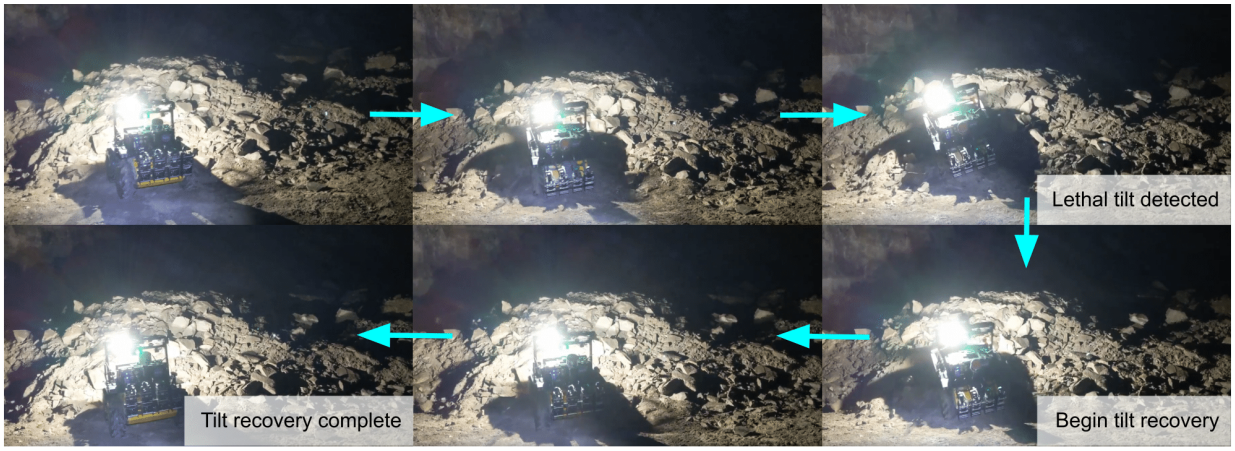

图 12：当机器人俯仰角超过安全限值时的倾斜恢复行为。

摆动行为：我们监测通过车轮和惯性里程计获得的机器人速度，以确保机器人在任务过程中没有陷入停滞。如果模型预测控制发出机器人移动指令时，机器人在一段时间内没有移动，就会被视为陷入停滞。在这种情况下，我们会启动一种摆动

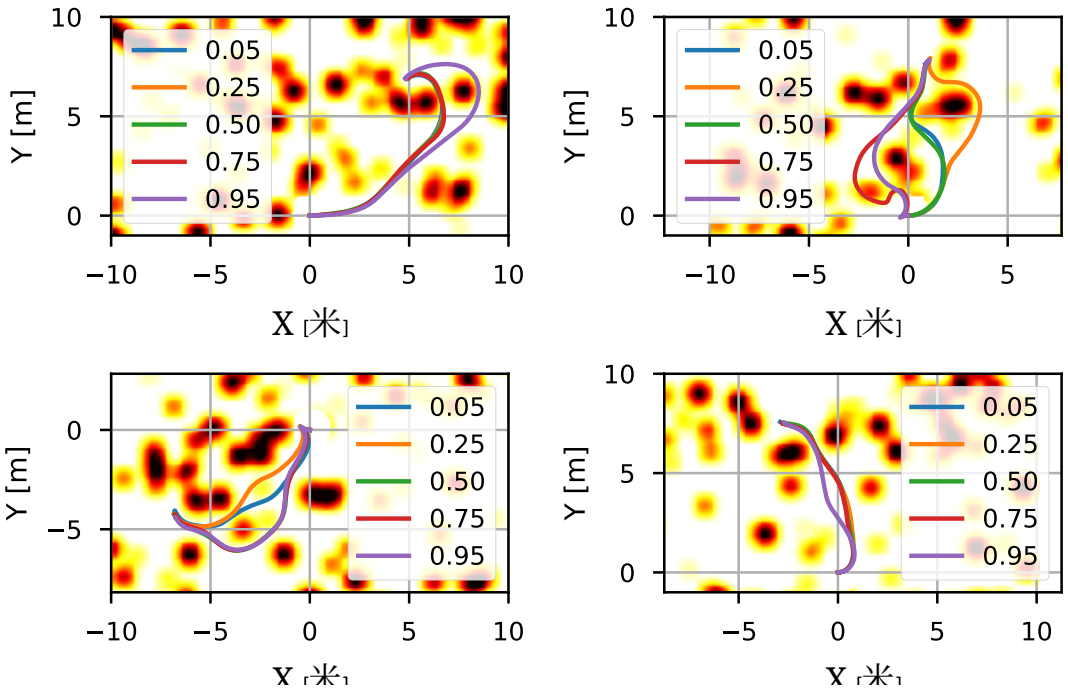

图13：四次仿真运行的路径分布。风险水平 α 的范围为0.1（接近均值）至0.95（保守）。较小的 α 通常会产生较短的路径，而较大的 α 则会选择统计上更安全的路径。

行为，使机器人的车轮以特定方式运动；若不受干扰，机器人将沿 8 字形轨迹行进。这样机器人便能来回摆动，直到摆脱停滞状态。如果摆动行为未能使机器人恢复运动，则机器人会被视为陷入停滞，任务也随之结束。

逃离致命区域：如果机器人发现自己被困在致命区域一段时间，采用一种尽可能降低风险、尝试逃离高风险区域的行为会很有用。具体做法如下：首先，选择一个目标位置，该位置不在致命区域内、靠近机器人，并且使机器人沿直线到达目标点时需要穿越的致命区域尽可能少。接着，调整可通行性代价地图的风险阈值，以允许机器人穿越通常会被视为约束或障碍物的高风险地图单元。然后，将目标点发送给全局和局部规划器，并由机器人执行。如果到达目标点，则结束恢复行为；如果目标点仍然无法到达，则可以进一步调整风险阈值，逐步允许更冒险的行为，以尝试脱困。通过这种方式，CVaR风险阈值有助于实时调整机器人的风险特征。

4 实验

本节报告 STEP 的实验表现和实地性能。我们首先在轮式差速驱动平台上，通过仿真对比不同的可调风险阈值。随后，我们展示了在充满杂乱障碍物的废弃地铁中部署轮式机器人的真实环境表现，以及在熔岩管环境中部署足式平台的表现。

4.1 仿真研究

为评估统计性能，我们使用随机生成的地图和目标点进行了 50 次蒙特卡洛仿真。这些结果也已纳入我们此前的工作 [20]。为每个网格单元分配随机可通行性代价。我们作出以下假设：1）不存在定位误差；2）不存在跟踪误差；3）采用带有人工噪声的简化感知模型。我们设置一个距离 8 m 的随机目标点，并评估路径代价和距离。我们采用差速驱动动力学模型（无横向速度）。

我们比较了采用不同 α 水平的 STEP。图13展示了不同规划配置下路径的分布。乐观型（接近均值）的规划器 $\alpha = 0.05$ 通常生成更短的路径，而保守设置 $\alpha = 0.95$ 则会进行较长的绕行路径，以选择在统计意义上更安全的路径。其他 α 设置

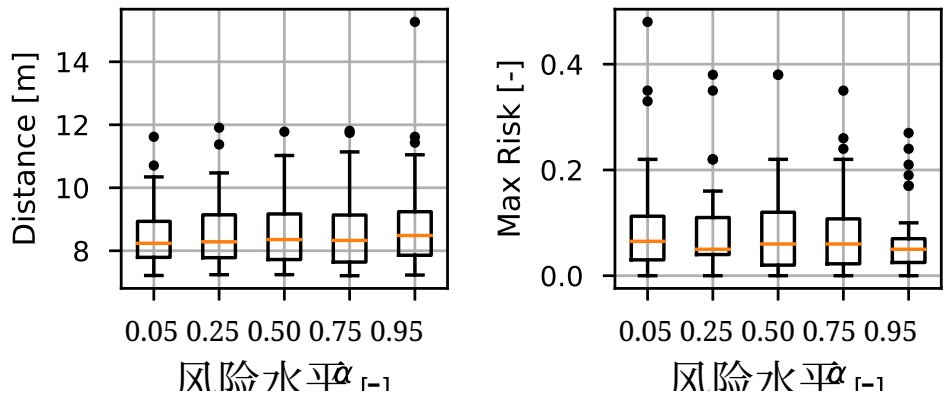

图14：50 次蒙特卡洛仿真中距离与风险之间的权衡。左：路径距离的分布。右：已行进路径上的最大风险分布。箱线图采用标准四分位数格式，圆点表示离群值。

显示出介于这两个极端之间的分布，其中较大的 α 生成与保守规划器相似的路径，而较小的 α 生成更多时间最优路径。统计数据如图14所示。

4.2 硬件结果

我们在不同的复杂环境中（包括废弃地铁、熔岩管、石灰岩矿，以及DARPA地下挑战赛中多种地下环境的组合）将STEP部署于两种不同的机器人（轮式和足式）上。机器人配备了定制传感与计算单元，并由喷气推进实验室的NeBula自主软件驱动 [1]。使用了3台Velodyne VLP-16采集激光雷达数据。定位由机载的基于激光雷达的同步定位与建图方案提供 [44, 19]。整个自主技术栈运行在一台英特尔酷睿 i7 处理器上。可通行性技术栈的典型CPU使用量约为单个核心。

4.2.1 此前结果

较早版本的STEP（不包含语义与置信度感知风险源、步态适应以及整套恢复行为）已在多个实地地点进行了测试。为完整起见，我们纳入了这些已发表于 [20], 的结果，来源如下：

1. 图15中，加利福尼亚州洛杉矶市中心一处充满工业杂物的废弃地铁，
2. 图 16 中，位于加利福尼亚州图莱莱克拉瓦床国家纪念碑的 Valentine Cave

4.2.2 肯塔基地下储存设施和洛杉矶地铁的测试结果

我们在肯塔基州威尔莫的肯塔基地下储存设施（KU）进行测试时，通过全自主运行测试了可通行性流程中的多个新部分。图5和图6展示了基于置信度的风险和基于语义的风险的代价地图结果。我们还在 KU 的一堆瓦砾上测试了倾斜恢复行为：通过迫使机器人进入危险倾斜姿态，机器人成功从这一高风险构型中恢复，如图 12 所示。

我们还在加利福尼亚州洛杉矶市中心一处多层、充满工业杂物的废弃地铁中测试了新成果。例如，洛杉矶地铁环境被用作测试平台，用于测试图9和图 11 分别展示的楼梯步态和蹲伏步态行为。

4.2.3 DARPA 地下挑战赛的测试结果

该可通行性框架部署于在肯塔基州路易维尔 Louisville Mega Cavern 举办的 DARPA地下挑战赛中。赛道由3种不同的环境组成——隧道、城市和洞穴。图17展示了赛道地图以及机器人探索的不同区域。图17所示区域按照 DARPA 发布的赛道布局指南，均被评为高难度[2]。各区域面临的地形挑战如下：

[2]课程指南可在此处访问：https://bitbucket.org/subtchallenge/finals_ground_truth/src/ master/course_design/Finals_Course_Callouts.pdf



图15：Husky 在废弃地铁实验中的可通行性分析结果。左上至右：三个不同风险水平下的风险地图：$\alpha=0.1$、0。5、0。9，分别对应。颜色表示条件风险价值（白色：安全（$r<=0.05$），黄色至红色：中等（$0.05<r<=0.5$），黑色：有风险（$r>0.5$））。图中还显示了最新的激光雷达测量结果（绿色点）。左下和中下：观察同一位置的前向和右侧车载摄像头。右下：自主探索后完成的环境俯视地图。亮点为柱子，在摄像头图像中清晰可见。这些风险分析结果产生了直观的效果——例如，一小堆金属虽然可能可以通行，但如果可能，应尽量避开。当某个区域的传感器覆盖不足时，风险会较高。当机器人更接近目标且传感器覆盖良好时，条件风险价值代价会降低，从而得到更准确的风险评估。与确定性方法相比，这能实现更高效、更安全的规划。

区域A 一个类似办公室的区域，由狭窄的走廊（$5'\times8'$）和小型房间（$8'\times8'$）组成，供机器人探索。

区域B 一个类似仓库的区域，拥有大量货架和杂乱障碍物，模拟地震后的工业仓库。

区域C 赛道城市部分与隧道部分之间的连接区域。楼梯对于轮式机器人而言构成负障碍。在狭窄走廊中存在这种负障碍，使得落差更难被检测到。

区域D 一条受限的通道，包含地面、墙壁和天花板障碍物——设有垂直管道和碎片 .

区域E 一个狭窄的洞穴入口，模拟人类必须爬行通过的区域。地面向上倾斜，而天花板向下倾斜，给地面分割和低矮天花板检测带来问题。

区域F 一个有瓦砾和松散岩石堆的小型石灰岩洞穴。

图18展示了上述不同场景中机器人可通行性分析的结果，以及机器人的几何规划和运动学与动力学规划。可以看出，机器人能够在所有这些场景中规划安全路径，并成功避开不同的地形危险因素。

5 讨论

我们介绍了随机可通行性评估与规划方法，以及该方法在 DARPA地下挑战赛中的表现。该方法成功应对了先验未知环境中的不确定性，并在各种地形中导航。在这项对 [20], 的扩展工作中，除了此前介绍的几何风险源外，我们还考虑了语义与置信度感知风险源。此外，我们还允许四足机器人根据观察到的地形风险以及从本体感觉传感器获取的信息调整步态。我们还增加了一套丰富的恢复行为，以增强可通行性技术栈的鲁棒性，从而应对框架其他部分无法处理、但会导致任务终止的意外情况。

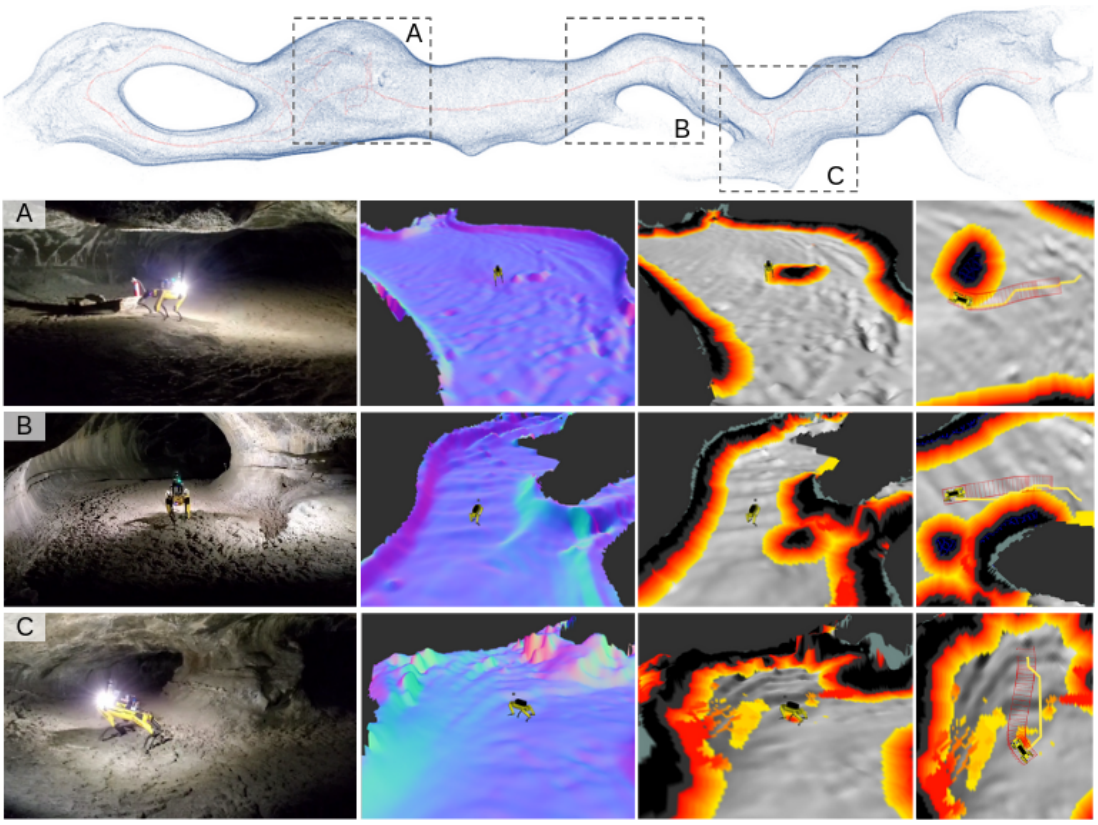

图16：Valentine Cave 实验的可通行性分析结果。从左到右依次为：第三人称视图、高程图（按法线方向着色）、风险图（按风险水平着色。白色：安全（$r <= 0.05$），黄色至红色：中等（$0.05 < r <= 0.5$），黑色：危险（$r > 0.5$）），以及规划的几何/运动动力学路径（黄色线条/红色框）。

我们的框架还有许多改进方向。STEP 考虑了静态环境产生的可通行性风险，但不会预测动态障碍物。因此，机器人无法规划反应式运动来避开坠落碎屑，也无法应对机器人下方瓦砾的突然移动。此外，对于机器人的所有速度，可通行性分析都相同。动态可通行性分析将支持更大范围的运动。基于激光雷达强度的语义地形分析可以粗略区分泥泞地形与普通地形；若在激光雷达强度信息之外再结合基于视觉的语义分类，就能更深入地了解机器人行进路径上的表面。

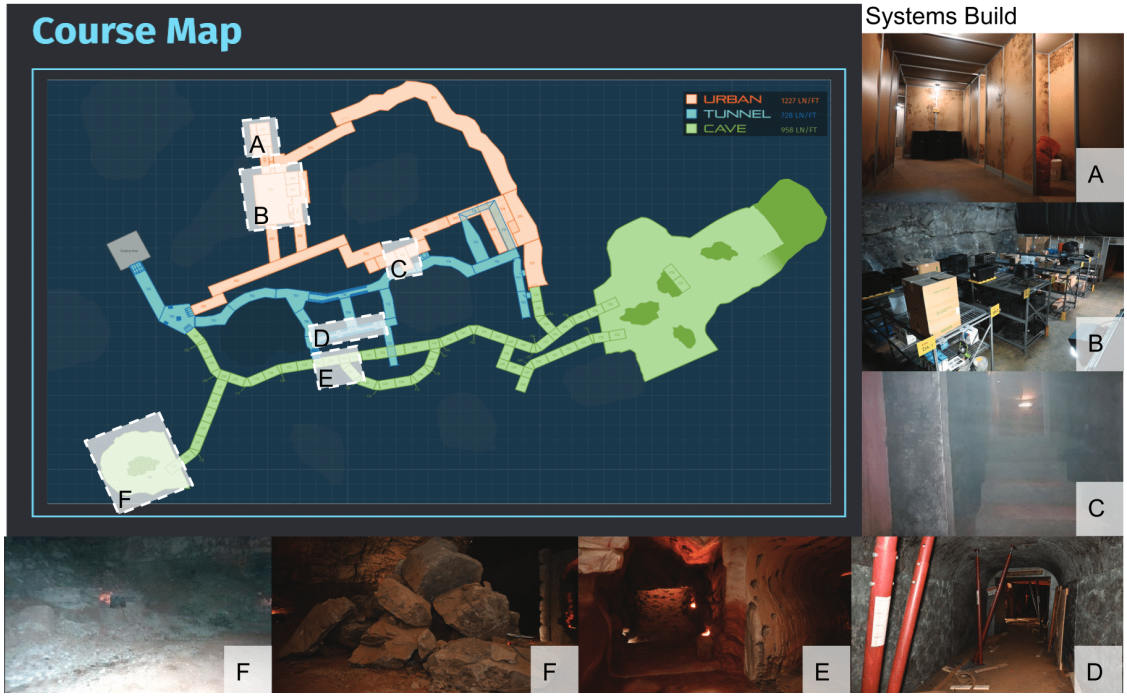

图17：竞赛路线的布局，以及机器人 traversed 区域的系统构建快照，这些区域具有有趣的地形且难度较高。

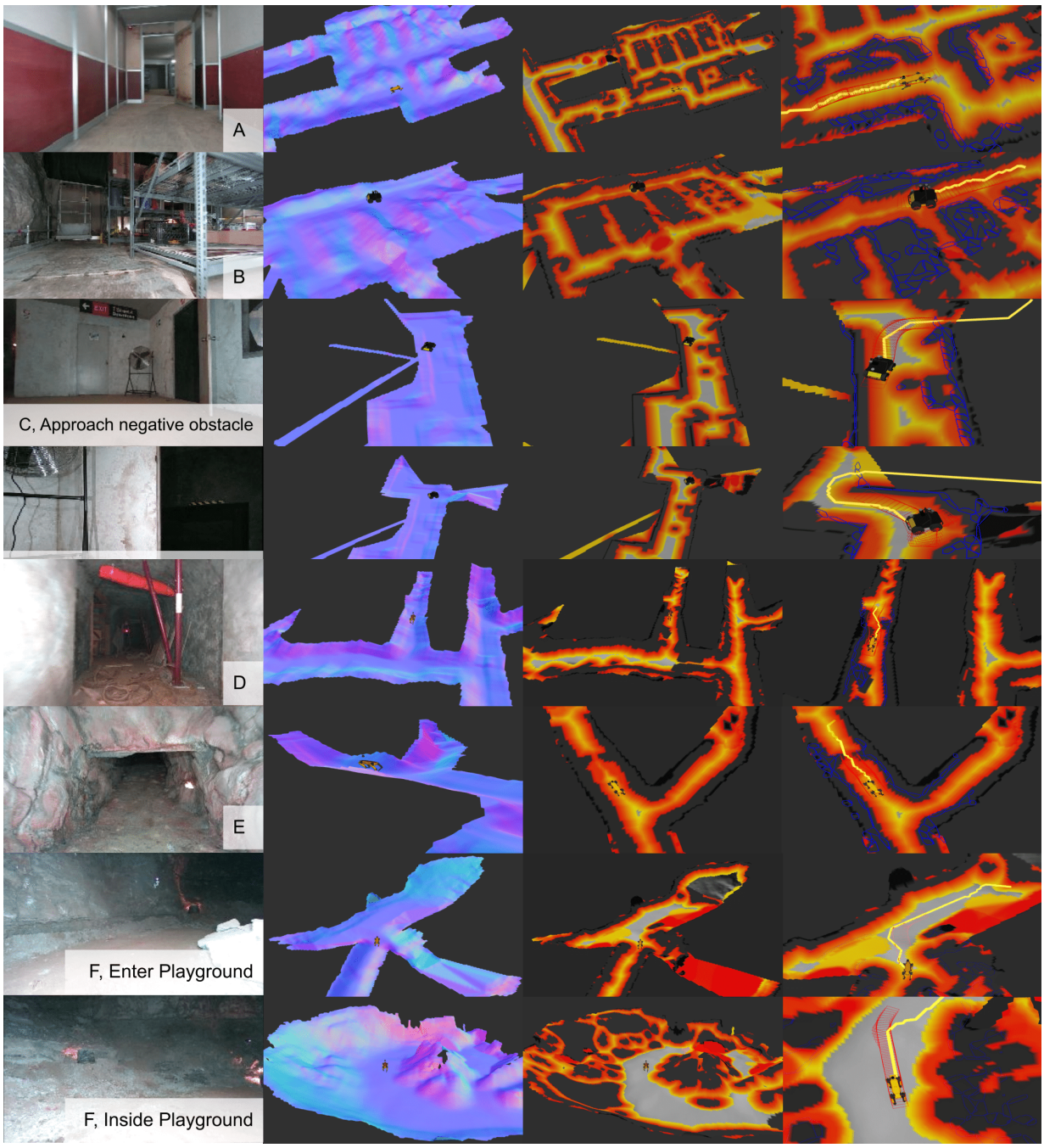

图18：DARPA地下挑战赛的可通行性分析结果。从左至右：机器人前置摄像头视图、高程图（按法线方向着色）、风险图（按风险水平着色——白色：安全（$r <= 0.05$），黄色至红色：中等（$0.05 < r <= 0.5$），黑色：高风险（$r > 0.5$）），以及规划的几何/运动动力学路径（黄色线条/红色方框）。

致谢

本研究由喷气推进实验室和加州理工学院在与美国国家航空航天局（80NM0018D0004）及国防高级研究计划局（DARPA）签订的合同项下开展。

参考文献

[1] A. Agha-mohammadi 等. “NeBula：在复杂环境中探索机器人自主性；TEAM CoSTAR 参加DARPA地下挑战赛”。载于：《现场机器人学》（2021）。[2] A. Agha-Mohammadi 等. “置信度丰富的栅格地图构建”。载于：《国际机器人研究杂志》38.12-13（2017），第 1352–1374 页。[3] Mohamadreza Ahmadi 等. “受约束的风险规避马尔可夫决策过程”。载于：人工智能 AAAI 会议。2021。[4] Mohamadreza Ahmadi 等. “不确定性下的风险规避规划”。载于：美国控制会议。2020，第 3305–3312 页。[5] Mohamadreza Ahmadi 等. “风险规避随机最短路径规划”。载于：arXiv:2103.14727（2021）。[6] A. Ahmadi-Javid. “熵风险价值：一种新的相干风险度量”。载于：优化理论与应用杂志 155.3（2012），第 1105–1123 页。[7] Philippe Artzner 等. “相干风险度量”。载于：《金融数学》9.3（1999），第 203–228 页。[8] F. Augugliaro、A.P Schoellig 和 R. D’Andrea. “四旋翼机队无碰撞轨迹的生成：一种序列凸规划方法”。载于：智能机器人与系统国际会议（2012）。[9] Paul T Boggs 和 Jon W Tolle. “序列二次规划”。载于：《数值学报》（1995），第 529–562 页。[10] Xiaoyi Cai 等. “通过学习得到的速度分布图实现风险感知越野导航”。载于：arXiv 预印本 arXiv:2203.13429（2022）。[11] Eduardo F Camacho 和 Carlos Bordons Alba. 模型预测控制。Springer Science & Business Media，2013。[12] Yinlam Chow 等. “风险敏感与鲁棒决策：一种条件风险价值优化方法”。载于：《神经信息处理系统进展》。2015，第 1522–1530 页。[13] Will Dabney 等. “利用分位数回归进行分布式强化学习”。载于：第三十二届人工智能 AAAI 会议。2018。[14] Sharmita Dey 等. “PrePARE：用于极端地形机器人探索中敏捷故障事件检测的预测性本体感觉”。载于：arXiv 预印本 arXiv:2208.00322（2022）。[15] Anushri Dixit、Mohamadreza Ahmadi 和 Joel W Burdick. “基于全变差距离的分布鲁棒模型预测控制”。载于：arXiv 预印本 arXiv:2203.12062（2022）。[16] Anushri Dixit、Mohamadreza Ahmadi 和 Joel W. Burdick. “基于全变差距离的分布鲁棒模型预测控制”。载于：《IEEE控制系统快报》6（2022），第 3325–3330 页。doi: 10.1109/LCSYS.2022.3184921。[17] Anushri Dixit、Mohamadreza Ahmadi 和 Joel W. Burdick. 风险规避滚动时域运动规划。2022。arXiv: 2204。09596 [eess.SY]。[18] Anushri Dixit、Mohamadreza Ahmadi 和 Joel W. Burdick. “使用熵风险价值进行风险敏感运动规划”。载于：欧洲控制会议。2021。[19] K Ebadi 等. “LAMP：面向感知能力受限地下环境探索的大规模自主地图构建与定位”。载于：IEEE机器人与自动化国际会议。2020，第 80–86 页。[20] D.D. Fan 等. “STEP：面向风险感知越野导航的随机可通行性评估与规划”。载于：机器人：科学与系统。2021。[21] David D Fan、Ali-akbar Agha-mohammadi 和 Evangelos A Theodorou. “用于管束模型预测控制的深度学习管束”。载于：机器人：科学与系统（RSS）（2020）。

[22] David D. Fan、Ali-akbar Agha-mohammadi 和Evangelos A.Theodorou. “学习复杂环境中可通行性的风险感知代价地图”。载于：IEEE机器人与自动化快报 7.1（2022），第 279–286 页。doi: 10.1109/LRA.2021.3125047。[23] Péter Fankhauser、Michael Bloesch 和 Marco Hutter. “面向不确定定位移动机器人的概率地形映射”。载于：IEEE机器人与自动化快报 3.4（2018），第 3019–3026 页。[24] Péter Fankhauser 和 Marco Hutter. “通用栅格地图库：崎岖地形导航的实现与应用案例”。载于：机器人操作系统（ROS）：完整参考指南（2016），第 99–120 页。[25] R. Foust、S.J. Chung 和 F.Y. Hadaegh. “利用序列凸规划实现非线性动力学下的最优制导与控制”。载于：制导、控制与动力学杂志 43.4（2020），第 633–644 页。[26] Lu Gan 等. “通过深度逆强化学习进行基于能量的足式机器人地形可通行性建模”。载于：IEEE机器人与自动化快报 7.4（2022），第 8807–8814 页。[27] S. Ghosh、K. Otsu 和 M. Ono. “行星漫游车运动规划的概率运动学状态估计”。载于：IEEE/RSJ智能机器人与系统国际会议。2018，第 5148–5154 页。[28] Steven B Goldberg、Mark W Maimone 和 Larry Matthies. “用于行星探索的立体视觉与漫游车导航软件”。载于：IEEE航空航天会议。IEEE。2002。[29] Alain Haït、Thierry Simeon 和 Michel Taïx. “崎岖地形轨迹规划算法”。载于：先进机器人学 16.8（2002），第 673–699 页。[30] Astghik Hakobyan、Gyeong Chan Kim 和 Insoon Yang. “使用CVaR约束优化进行风险感知运动规划与控制”。载于：IEEE机器人与自动化快报 4.4（2019），第 3924–3931 页。[31] Michael Himmelsbach、Felix V Hundelshausen 和 H-J Wuensche. “面向地面车辆的三维点云快速分割”。载于：2010年IEEE智能车辆研讨会（IV）。IEEE。2010，第 560–565 页。[32] Li Hui 等. “激光强度在激光雷达点云数据分类中的应用”。载于：IGARSS 2008——2008年IEEE国际地球科学与遥感研讨会。第 2 卷。2008，第 II-1140–II–1143 页。doi: 10.1109/IGARSS.2008.4779201。[33] Mrinal Kalakrishnan 等. “STOMP：用于运动规划的随机轨迹优化”。载于：IEEE机器人与自动化国际会议。2011，第 4569–4574 页。[34] Himangshu Kalita 等. “外星熔岩管与洞穴内部的路径规划与导航”。载于：IEEE/ION定位、位置与导航研讨会。2018，第 1311–1318 页。[35] Sven Koenig 和 Reid G Simmons. “基于概率决策图的风险敏感规划”。载于：知识表示与推理原理。Elsevier。1994，第 363–373 页。[36] Thomas Lew、Riccardo Bonalli 和 Marco Pavone. “用于鲁棒轨迹优化的机会约束序列凸规划”。载于：2020年欧洲控制会议（ECC）。IEEE。2020，第 1871–1878 页。[37] Johan Löfberg. “哎呀！我不能再来一次：检验模型预测控制中的递归可行性”。载于：Automatica 48.3（2012），第 550–555 页。[38] Anirudha Majumdar 和 Marco Pavone. “机器人应如何评估风险？迈向机器人学中的风险公理化理论”。载于：机器人研究。Springer，2020，第 75–84 页。[39] Travis Manderson 等. “利用机载摄像头和稀疏航拍图像学习在非结构化环境的平坦地形上越野行驶”。载于：2020年IEEE机器人与自动化国际会议（ICRA）。IEEE。2020，第 1263–1269 页。[40] D. Morgan、S.J. Chung 和 F. Hadaegh. “利用序列凸规划对航天器集群进行模型预测控制”。载于：《制导、控制与动力学杂志》37（2014年4月），第 1–16 页。doi: 10.2514/1.G000218。[41] Jorge Nocedal 和 Stephen Wright. 数值优化。Springer Science & Business Media，2006。[42] Masahiro Ono 等. “应用于风险感知机器人空间探索的机会约束动态规划”。载于：自主机器人 39.4（2015），第 555–571 页。[43] Kyohei Otsu 等. “面向铰接式悬挂系统漫游车的快速近似净空评估”。载于：《现场机器人学》37（5，2020），第 768–785 页。[44] Matteo Palieri 等. “LOCUS：一种用于实时高精度里程计与三维建图的多传感器激光雷达中心化方案”。载于：IEEE机器人与自动化快报 6.2（2020），第 421–428 页。

[45] Panagiotis Papadakis. “无人地面车辆地形可通行性分析方法综述”。载于：《人工智能工程应用》26.4 (2013)，第 1373–1385 页。[46] Andrzej Ruszczynski. “马尔可夫决策过程的风险规避动态规划”。载于：《数学规划》75.2 (2014)，第 235–261 页。[47] John Schulman 等。“采用序列凸优化和凸碰撞检测的运动规划”。载于：《国际机器人研究杂志》33.9 (2014)，第 1251–1270 页。[48] Tobias Schwarze 和 Zhichao Zhong。“基于自我中心立体视觉的楼梯检测与跟踪”。载于：*2015 IEEE*国际图像处理会议 *(ICIP)*。IEEE，2015，第 2690–2694 页。[49] Sumeet Singh 等。“时间一致性风险敏感模型预测控制框架：理论与算法”。载于：《*IEEE*自动控制汇刊》64.7 (2018)，第 2905–2912 页。[50] B. Stellato 等。“OSQP：用于二次规划问题的算子分裂求解器”。载于：《数学规划计算》(2020)，第 1–36 页。[51] Rohan Thakker 等。“在感知条件具有挑战性的极端地形上进行自主越野导航”。载于：国际实验机器人学研讨会 (2020)。[52] Allen Wang、Ashkan Jasour 和 Brian Williams。“智能体不确定性下自动驾驶车辆的非高斯机会约束轨迹规划”。载于：《机器人学与自动化快报》5.4 (2020)，第 6041–6048 页。[53] Skylar X Wei 等。“移动障碍物避碰：一种数据驱动的风险感知方法”。载于：《*IEEE*控制系统快报》(2022)。[54] Huan Xu 和 Shie Mannor。“分布鲁棒马尔可夫决策过程”。载于：《神经信息处理系统进展》。2010，第 2505–2513 页。

差速驱动系统的动力学模型

对于一个可产生前向速度和转向的简单系统（例如差速驱动系统），我们可能希望将状态和控制量定义为:

$$x=[p_x,p_y,p_\theta,v_x]^\intercal \tag{32}$$

$$u=[a_x,v_\theta]^\intercal \tag{33}$$

例如　　，简单差速驱动系统的动力学 $x_{k+1}=f(x_k,\ u_k)$ 可写为:

$$x_{k+1}=x_k+\Delta t\begin{bmatrix} v_x\cos(p_\theta) \\ v_x\sin(p_\theta) \\ \gamma v_x+(1-\gamma)v_\theta \\ a_x \end{bmatrix} \tag{34}$$

其中 $\gamma\in[0,1]$ 是一个常数，用于调节车辆允许原地转向的幅度　　.

B 使用正态分布的条件风险价值（CVaR）定义动态风险度量

$$\begin{aligned} J(x_0,\pi)&=R_0+\rho_0\big(R_1+\rho_1\big(R_2+...+\rho_{T-1}(R_T)\big)\big) \\ &=R_0+\rho\Bigg(R_1+\rho\Big(R_2+...+\rho(R_{T-1}+ \\ &\qquad \mu_T+\sigma_T\frac{\varphi(\boldsymbol{\Phi}^{-1}(\alpha))}{1-\alpha})\Big)\Bigg) \\ &=R_0+\rho\Bigg(R_1+\rho\Big(R_2+...+\rho(R_{T-2}+ \\ &\qquad \mu_{T-1}+\mu_T+(\sigma_{T-1}+\sigma_T)\frac{\varphi(\boldsymbol{\Phi}^{-1}(\alpha))}{1-\alpha})\Big)\Bigg) \\ &\vdots \\ &=R_0+\sum_{i=1}^{T}\left(\mu_i+\sigma_i\frac{\varphi(\boldsymbol{\Phi}^{-1}(\alpha))}{1-\alpha}\right) \\ &=\sum_{i=0}^{T}\rho(R_i) \end{aligned}$$

C 动力学规划示意图

见图19。

D 方向约束的梯度

我们将进一步详细说明方向约束的推导。将位置记为 $p=[p_x,p_y]$，将位置/偏航角记为 $s=[p_x,p_y,p_\theta]$。我们希望求出机器人在机体坐标系中的俯仰角 ψ和横滚角 ϕ。令$\omega=[\psi,\phi]$。约束的形式为 $|\omega(s)|<=\omega_{max}$。在 p处，我们计算表面法向量，并将其记为 $n^w=[n_x^w,n_y^w,n_z^w]$，该向量位于世界坐标系中。为了将法向量转换到机体坐标系中，记为 $n^r=[n_x^r,n_y^r,n_z^r]$，我们根据机器人的偏航角对其进行旋转： $n^r=R_\theta n^w$（见图 8），其中 R_θ是绕世界坐标系的 z 轴旋转 θ角度的基本旋转矩阵:

$$R_\theta=\begin{bmatrix} \cos p_\theta & \sin p_\theta & 0 \\ -\sin p_\theta & \cos p_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{35}$$

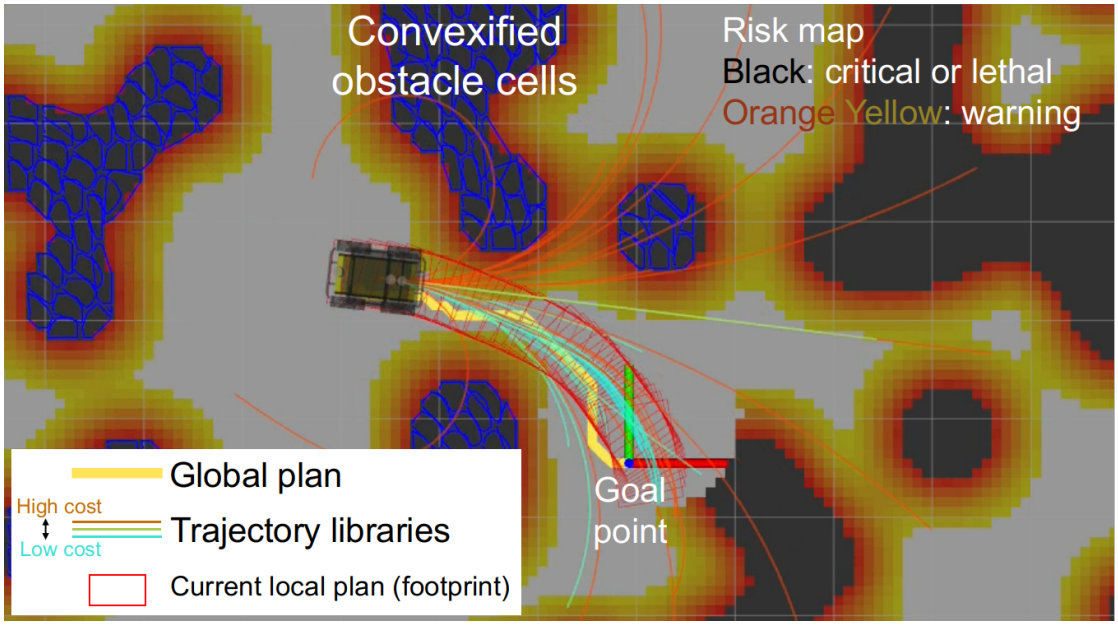

图19：运动学与动力学模型预测控制规划器示意图。该规划器首先评估轨迹库中的多条路径，随后选取代价最低的路径作为候选项，并由二次规划求解器对其进行优化。

令机器人俯仰角和横滚向量 ω 定义为 $\omega = g(n^r)$，其中：

$$\omega = g(n^r) = \begin{bmatrix} \text{atan2}(n_x^r, n_z^r) \\ -\text{atan2}(n_y^r, n_z^r) \end{bmatrix} \tag{36}$$

创建线性约束问题需要对约束进行线性近似 t:

$$|\nabla_s \omega(s) \delta s + \omega(s)| <= \omega_{max} \tag{37}$$

方便的是，计算 $\nabla_s \omega(s)$ 可分别求出相对于位置和偏航角的梯度。令 $\nabla_s \omega(s) = [\nabla_p \omega(s) \ \nabla_\theta \omega(s)]$ ，，则：

$$\nabla_p \omega(s) = (\nabla_{n^r} g)(R_\theta)(\nabla_p n^w) \tag{38}$$

$$\nabla_\theta \omega(s) = (\nabla_{n^r} g)(\frac{d}{d\theta} R_\theta)(n^w) \tag{39}$$

其中：

$$\nabla_{n^r} g = \begin{bmatrix} \frac{n_z^r}{(n_x^r)^2 + (n_z^r)^2} & 0 & \frac{-n_x^r}{(n_x^r)^2 + (n_z^r)^2} \\ 0 & \frac{-n_z^r}{(n_y^r)^2 + (n_z^r)^2} & \frac{n_y^r}{(n_y^r)^2 + (n_z^r)^2} \end{bmatrix} \tag{40}$$

and

$$\nabla_p n^w = \begin{bmatrix} \frac{\partial n_x^w}{\partial p_x} & \frac{\partial n_x^w}{\partial p_y} \\ \frac{\partial n_y^w}{\partial p_x} & \frac{\partial n_y^w}{\partial p_y} \\ \frac{\partial n_z^w}{\partial p_x} & \frac{\partial n_z^w}{\partial p_y} \end{bmatrix} \tag{41}$$

形式为 $\frac{\partial n_x^w}{\partial p_x}$ 的项相当于计算 2.5D 地图上高程的二阶梯度。可使用数值方法高效地完成此计算 [24]。

E 将非线性模型预测控制问题转换为二次规划问题

方程（10-13）中给出的模型预测控制问题是非线性的。为了高效地求解，我们围绕一个初始解将问题线性化，并采用序列二次规划（SQP）方式迭代求解 [41]。令 $\{\hat{x}_k,\hat{u}_k\}_{k=0,\cdots,T}$ 表示一个初始解。令 $\{\delta x_k,\delta u_k\}_{k=0,\cdots,T}$ 表示相对于初始解的偏差。我们将（10-13）近似为一个问题，其中代价项为二次函数，约束为关于 $\{\delta x,\delta u\}$ 的线性函数：

$$\begin{aligned}\{\delta x^*,\delta u^*\}=\underset{\delta x,\delta u}{\operatorname{argmin}}&\sum_{k=0}^{T}\|\hat{x}_k+\delta x_k-x_k^*\|_{Q_k}\\&+\lambda J(\hat{x}_k+\delta x_k,\hat{u}_k+\delta u_k)\\ s.t.\quad &\forall k\in[0,\cdots,T]:\end{aligned}\tag{42}$$

$$\hat{x}_{k+1}+\delta x_{k+1}=f(\hat{x}_k,\hat{u}_k)+\nabla_x f\cdot\delta x_k+\nabla_u f\cdot\delta u_k \tag{43}$$

$$g(\hat{u_k})+\nabla_u g\cdot\delta u_k\succ 0 \tag{44}$$

$$h(m,\hat{x}_k)+\nabla_x h\cdot\delta x_k\succ 0 \tag{45}$$

其中 $J(\hat{x}_k+\delta x_k,\ \hat{u}_k+\delta u_k)$ 可用二阶泰勒近似表示（暂且假设其不依赖于控制量）：

$$J(\hat{x}+\delta x)\approx J(\hat{x})+\nabla_x J\cdot\delta x+\delta x^{\intercal}H(J)\delta x \tag{46}$$

而 $H(\cdot)$ 表示海森矩阵。现在，该问题是一个具有二次代价和线性约束的二次规划（QP）。为求解方程（42-45），我们引入解向量变量 X：

$$X=\begin{bmatrix}\delta x_0^{\mathrm{T}} & \cdots & \delta x_T^{\mathrm{T}} & \delta u_0^{\mathrm{T}} & \cdots & \delta u_T^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{47}$$

于是，我们可以将方程（42-45）写成以下形式：

$$\text{minimize}\quad \frac{1}{2}X^{\mathrm{T}}PX+q^{\mathrm{T}}X \tag{48}$$

$$\text{subject to}\quad l\le AX\le u \tag{49}$$

其中，P 是半正定权重矩阵，q 是用于定义目标函数一阶项的向量，A 定义不等式约束，而 l 和 u 分别给出其下限和上限。

F 用于 SQP 解细化的线搜索算法

序列二次规划问题的解会返回一个优化后的控制序列 $\{u_k^*\}_{k=0:T}$。然后，我们使用线搜索程序，利用算法 3 找到合适的校正系数 γ。最终得到的校正系数会沿用到下一个路径规划循环中。

算法 3 线搜索算法

输入： 候选控制序列 $\{u_k^{\mathrm{c}}\}_{k=0:T}$，二次规划解 $\{\delta u_k^*\}_{k=0:T}$

输出： 校正系数 γ

初始化

将 γ 初始化为默认值或上次使用的值

$[c,o]$=获取代价和障碍物($\{u_k^{\mathrm{c}}\}_{k=0:T}$)

线搜索循环

对 $i=0$至最大值 _迭代 **执行**

  对 $k=0$至 T**执行**

    $u_k^{\mathrm{c}(i)}=u_k^{\mathrm{c}}+\gamma\delta u_k^*$

  结束循环

  $[c^{(i)},o^{(i)}]$=获取代价和障碍物($\{u_k^{\mathrm{c}(i)}\}_{k=0:T}$)

  如果 $(c^{(i)}\le c$且 $o^{(i)}\le o)$ **则**

    γ=min($2\gamma,\gamma_{max}$)

    跳出

  else

    γ=最大值($\gamma/2,\gamma_{min}$)

  结束条件判断

结束循环

返回 γ