arXiv:2602.02826v1 [math.OC] 2 Feb 2026

面向结构化环境中全向车辆的快速近似时间最优运动规划

Louis Callens[1*], Bastiaan Vandewal[1], Ibrahim Ibrahim[1], Jan Swevers[1], Wilm Decr´e[1]

[1]MECO研究团队，机械工程系，鲁汶大学，Celestijnenlaan 300，鲁汶，3001，比利时。[1]Core Lab MPRO，Flanders Make@鲁汶大学，比利时。

*通讯作者。电子邮箱：louis.callens@kuleuven.be；贡献作者：bastiaan.vandewal@kuleuven.be；ibrahim.ibrahim@kuleuven.be；jan.swevers@kuleuven.be；wilm.decre@kuleuven.be；

摘要

本文提出了一种新颖且高效的基于优化的方法，用于生成全向车辆在复杂但结构化环境中行驶时的近似时间最优轨迹。该方法旨在解决采用磁悬浮技术的平面运动系统的运动规划问题，这类系统可应用于装配线、自动化实验室或洁净室。在这些应用中，需要能够实时计算的时间最优轨迹，以提高生产率，并使车辆在需要时具备反应能力。所提出的方法使用自由空间走廊对环境表示进行编码，并利用运动基元表示车辆通过此类走廊的运动。这些基元以启发式方式进行选择，并通过一个优化问题确定其有限数量的自由度，从而定义轨迹。因此，在固定走廊序列内，该方法在不显著牺牲最优性的情况下，相较于先进水平的方法实现了显著更低的计算时间，尤其是相较于求解完整最优控制问题（OCP）、OMG-tools或VP-STO。该方法通过仿真进行了广泛的基准测试，并在真实世界的Beckhoff XPlanar系统上进行了验证。

关键词： 约束运动规划、优化与最优控制、运动控制、碰撞避让 n

1 引言

移动机器人广泛应用于具有挑战性的工业环境中，通常需要自主绕过障碍物。全向车辆能够独立于自身朝向向任意方向移动，因此尤其适用于拥挤环境，因为它们所需的机动空间较小。轮式移动机器人可以被设计成通过采用特殊的车轮设计实现全向运动，将其转变为全向移动机器人（OMR）（Taheri & Zhao, 2020）。然而，这类全向轮面临打滑、遇到不平整表面或出现轮系奇异性等问题。轮式全向车辆的另一种选择是平面移动器系统，它利用磁悬浮实现悬浮运动（NithinJarralde, Vibin, Govardhan,

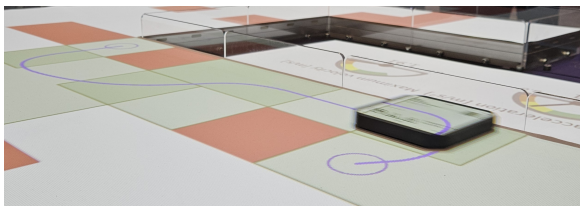


图 1：Beckhoff Automation 的 APM4220 XPlanar 移动器（不带任何有效载荷），尺寸为 113mm × 113mm，执行时间最优轨迹，该轨迹从上方投影后显示为蓝线。红色方块表示（虚拟）障碍物

& Raffik，2023）。在这些系统中，产生磁场的台面可以让永磁体以微米级精度移动（Scha-effel et al.，2016）（Jansen, van Lierop,Lomonova,& Vandenput，2008）。高定位精度和非接触式运动使该系统特别适用于小型装配线、洁净室或实验室。近年来，这项技术已应用于商业产品，例如 B&R工业自动化的 ACOPOS 6D、Planar Motor 的 XBot Flyway 或 Beckhoff 自动化的 XPlanar Mover System（见图1），并已在工业领域得到采用，大幅提升了制造业的灵活性。

为了充分发挥硬件所提供的灵活性，需要一种运动规划算法来规划移动器从一点到另一点的路径。这种算法需要满足若干要求。首先，它的计算效率必须足够高，以便在移动过程中能够重新规划路径，从而应对动态环境。车辆移动速度较快时，规划器尤其需要高效。其次，运动规划算法必须遵守速度限制以确保安全，并遵守加速度限制，以确保未固定的有效载荷不会在移动器上发生不必要的滑动。第三，移动器必须能够穿过狭窄通道，因为用于产生磁场的磁瓦成本高昂，可用的自由空间通常有限。这意味着在考虑车辆的占用轮廓时不能引入任何保守性。最后，为了提高这些运动系统在工业应用中的吞吐量，需要实现时间最优运动。

1.1 相关工作

为解决这一运动规划问题，可以采用基于采样的规划器，例如快速探索随机树（RRT）（Kuffner 和 LaValle，2000）。后来，这种方法得到扩展，并被证明具有渐近最优性，由此产生了 RRT*（Karaman 和 Frazzoli，2011）。研究者也尝试过提高其收敛速度。Nasir 等人（2013）提出了 RRT-smart，它采用智能采样技术，在障碍物附近进行采样，并通过改变图中的连接关系来优化中间路径。Gammell、Srinivasa 和 Barfoot（2014）提出了 informed-RRT，该方法在找到初始路径后，以更直接的方式对环境进行采样。关于基于RRT的方法的综述可参见 Noreen、Khan 和 Habib （2016）。需要注意的是，这些方法通常生成不光滑的距离最优轨迹，而不是时间最优轨迹（加速度有界）。

为在基于采样的方法中考虑车辆动力学，Webb 和 van den Berg(2013) 提出了动力学 RRT*，用于计算具有线性动力学（例如双积分器）的车辆的最优轨迹。采样点通过闭式形式的优化轨迹进行连接，而不是通过直线连接。然而，这些连接轨迹的目标函数除了行驶时间外，还包含加速度的非零二次惩罚项，因此得到的轨迹并非纯粹的时间最优轨迹。加速度限制仅通过在超出约束时拒绝采样点来间接考虑，这可能导致大量采样点被拒绝；因此，在追求时间最优时，所需的采样点数量会非常多。 Xie、van den Berg、Patil 和 Abbeel(2015年) 将动力学 RRT 的思想扩展* 到非线性系统，并解决了间接处理加速度约束的问题。具体而言，他们使用序列二次规划（SQP）求解边值问题（BPV），以寻找采样点之间满足约束的时间最优连接。然而，整个轨迹的最优性仍取决于采样点本身，并且只能渐近实现。这导致规划时间过长，无法用于实时应用。该方法并不优化采样点之间的轨迹，

基于途经点的随机轨迹优化（VP-STO）（Jankowski、Bruderm¨uller、Hawes 和 Calinon，2023）使用 OCP 的解析解连接采样点，并利用协方差矩阵自适应（CMA-ES）以无梯度方式优化采样点的位置，从而最小化用户指定的任意目标函数。该算法还提出了在线版本。然而，由于优化不会运行至收敛，且仅使用有限数量的途经点，其最优性有所牺牲。

与基于采样的方法相比，基于优化的方法通过求解所谓的最优控制问题（OCP），以更直接的方式考虑动力学、约束和目标函数。然而，求解完整的 OCP 往往太慢，无法用于实时场景，尤其是在环境中存在许多障碍物时。为了缩短求解 OCP 的计算时间，Mercy、Van Loock 和 Pipeleers（2016）提出了一种名为 OMG-tools 的工具箱，用样条表示状态轨迹和控制量。这种方法能够确保轨迹平滑，并减少优化问题中的自由度，从而加快求解速度。然而，时间最优轨迹中的控制量通常并不平滑。该方法通过分离超平面处理障碍物，但当障碍物较多时，其扩展性较差。

与其在优化问题中考虑所有障碍物，不如将自由空间建模为一个可供通行的凸区域。一种此类方法是走廊地图方法（Ger-aerts & Overmars，2007；Overmars, Karamouzas,& Geraerts，2008）。利用半定规划迭代区域膨胀（IRIS）（Deits&Tedrake，2015年），可以快速构造大型、无障碍物的椭圆区域。该算法交替计算分离超平面，并膨胀内切于这些超平面所定义多面体中的椭圆。利用搜索算法构造初始猜测后，Iwanaga andFujimoto（2025）使用改进版 IRIS 算法，在 OCP 网格点周围生成多面体走廊。该方法能够为自行车模型在杂乱环境中的狭窄通道内构造最优轨迹。OMG-tools 扩展后可利用表示自由空间的基本矩形走廊，为全向车辆规划一条最优轨迹（Mercy,Hostens,&Pipeleers，2018年）。矩形走廊能够准确表示结构化环境中的自由空间。Bos, Vandewal,Decr´e, andSwevers （2023）将矩形走廊的思路应用于卡车拖挂拖车的停车机动。在此，走廊表示道路上的车道、停车场或允许通行的机动空间。De Santis,Astudillo,Decr´e, andSwevers（2024年）利用仅以矩形走廊表示环境所带来的简化，解析地确定时间最优轨迹，从而完全消除对优化的需求。具体做法是根据矩形走廊的几何形状选取精确的运动基元。不过，该规划器仅适用于独轮车的速度控制，因此会生成加速度无限大的轨迹；对于搭载有效载荷的平面移动器系统而言，这并不可取。如何解析地扩展基元选择方法，使其纳入加速度限制，目前尚不明确。

OMG-tools 和Bos 等人（2023）均使用 IPOPT（W¨achter 和 Biegler，2006）求解由此得到的优化问题。IPOPT 是一种用于非线性规划（NLP）的通用内点求解器。然而，IPOPT 并未利用 OCP 中通常存在的拉格朗日海森矩阵和约束雅可比矩阵的稀疏结构。FATROP（Vanroye、Sathya、De Schutter 和 Decr´e，2023）基于 IPOPT 的理念，但能够利用这一已知结构，从而显著缩短求解 OCP 的计算时间。IPOPT 和 FATROP 均可通过`CasADi` （Andersson、Gillis、Horn、Rawlings 和 Diehl，2018年）轻松使用。CasADi 是一个开源工具箱，可构建能够通过算法微分（AD）轻松求导的符号表达式。作为`CasADi`一部分的 Opti 栈，能够轻松指定优化问题，并使用所选求解器进行求解。

1.2 主要贡献

本文提出一种基于优化的方法，用于高效计算全向车辆在结构化环境中的无碰撞近似时间最优轨迹，同时满足速度和加速度约束。我们的方法使用矩形走廊对结构化环境中的自由空间进行建模。类似

于Jankowski等人（2023）和Mercy等人（2018年）的思路，获得并优化了轨迹的低维表示。我们通过基于走廊几何形状选择运动基元来实现这一点，这与De Santis（2024年）的工作相关。

我们对该方法进行了广泛的基准测试，并验证了给定走廊序列时运动基元的选择与优化。结果表明，在大多数情况下，我们能以显著更短的时间找到最优解。此外，我们还通过与 VP-STO 进行基准测试，评估了通过走廊序列进行规划的思路。除了在仿真中进行基准测试外，我们还在真实系统上验证了该方法：Beckhoff 自动化公司的 XPlanar Mover System。验证结果表明，所提出的轨迹能够在真实硬件上准确执行。

本研究的主要贡献包括：(1) 仅根据矩形走廊的几何形状选择时间最优运动基元；(2) 对由此得到的优化问题进行建模，并在不显著牺牲最优性的前提下高效求解，以确定其余自由度。

1.3 论文结构

本文其余部分的结构安排如下：第2节首先详细介绍问题描述，随后在第3节说明所提方法。我们将在第4节对所提方法进行基准测试和分析。在第5节讨论在真实系统上的验证。最后，我们将在第6节总结全文并提出未来工作的建议。

1.4 符号说明

本文中，除非另有说明，粗体小写字母均表示二维向量。例如 $\boldsymbol{q} = (q^x,\ q^y)^\top \in \mathbb{R}^2$。对这些向量的运算始终逐元素进行。

2 问题描述

本文所考虑的动力学是全向车辆的动力学，建模为二维双积分器。我们考虑一辆具有矩形占地轮廓的车辆，宽度为 W，长度为 L。该车辆与世界坐标系的坐标轴对齐，且不发生旋转。需要注意的是，仅当车辆与坐标轴对齐时，XPlanar mover才能进行平移，这证明了关于车辆朝向所作假设的合理性。动力学由下式给出

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{v}(t) \qquad \dot{\boldsymbol{v}}(t) = \boldsymbol{a}(t) \tag{1}$$

其中，$\boldsymbol{p}(t)$、$\boldsymbol{v}(t)$、$\boldsymbol{a}(t)$分别表示车辆的几何中心、速度和加速度。

目标是使车辆从起始位置 $\boldsymbol{p}_0$ 尽可能快地到达目的地 $\boldsymbol{p}_n$，给定初始速度 $\bar{\boldsymbol{v}}_0$ 和零终端速度 $\boldsymbol{v}_n = 0$，同时满足以下速度和加速度限制：

$$||\boldsymbol{v}(t)||_\infty \leq v_{\max} \qquad ||\boldsymbol{a}(t)||_\infty \leq a_{\max}. \tag{2}$$

无穷范数用于对每个维度上的独立驱动进行建模，适用于全向车辆。龙门起重机等全向车辆显然具有解耦的驱动方式。对于平面移动器系统，最大加速度值则更为复杂，且取决于有效载荷的质心、移动器的高度及其他因素。所提方法采用无穷范数进行解耦，从而简化规划问题并减少计算时间。

车辆必须避开环境中的障碍物。我们假设环境是静态的，并且已有该环境的占据栅格地图。该栅格是一个与世界坐标轴对齐的矩阵，其中每个栅格单元的尺寸至少为 $W \times L$。所提方法针对的是能够以这种方式表示的结构化环境。对于采用磁悬浮的平面运动系统，环境通常由相互连接的方形磁瓦组成，这些磁瓦会产生磁场，因此真实世界环境天然适合采用所提环境表示。请注意，在此应用中，障碍物通常包括静止的移动器、环境边界或故障瓦片。停车场或仓库也是可以轻松用这种方式表示的环境示例。

由于当前的动力学彼此解耦，因此可以分别计算点到点运动

在两个维度上的运动。因此，最短时间运动总会有一个维度成为瓶颈。假设已分别计算出两个维度的最优运动时间，且 $T_x^* < T_y^*$，那么 x 方向上任何满足 $T_x < T_y^*$ 的可行解，其整体目标函数值都相同。这意味着在 x 维度上还存在额外的自由度。在这种情况下， y 维度可视为瓶颈方向（B），而 x 维度则是自由方向（F）。

3 方法

上述问题的所提方法包括三个步骤，如图 2所示。首先，构建矩形走廊序列，以简化环境的表示。这部分内容见第 3.1节，图 2b-2c对其进行了可视化。其次，在这些走廊中规划一条动力学可行且最优的轨迹。如第 3.2节所示，首先构造一个忽略走廊边界的解析解。如果该解析解无碰撞，则问题得以解决。否则，必须显式考虑走廊边界。具体做法是根据走廊的几何形状（第 3.4节）选择参数化运动基元（定义见第 3.3节）。图 2d 展示了一个示例中由所选基元表示的随机轨迹。随后，按照第 3.5节的说明，对剩余自由度进行优化。第 3.6 节讨论了优化问题构造的一些具体细节。第 3.7 节以伪代码形式总结了该方法。最后，第 3.8节列出了该方法的局限性。

由于所提方法基于参数化运动基元，因此我们称其为 PMP。实现代码可在此获取：https://github.com/meco-group/parametric-motion-planner#。

3.1 自由空间走廊序列的构建

构建 n 自由空间走廊 $\boldsymbol{C}$ 序列的目标是以简化且凸的方式表示环境。这些走廊是与世界坐标轴对齐的矩形，因此它们可以定义为

$$\mathcal{C}_i \coloneqq \{\boldsymbol{q} : q^x \in [x^i_{\min}, x^i_{\max}], q^y \in [y^i_{\min}, y^i_{\max}]\} \tag{3}$$

其中 $i = 0, \ldots\ldots, n-1$。此定义以及矩形走廊的选择使我们能够以解耦的方式处理约束；若考虑椭圆走廊（例如 IRIS 计算得到的走廊），则无法做到这一点（Deits 和 Tedrake，2015年）。更具体地说，要求车辆位于位置 $\boldsymbol{p}$且处于 $\mathcal{C}_i$ 内，会得到以下约束

$$x^i_{\min} + \frac{W}{2} \leq p_x \leq x^i_{\max} - \frac{W}{2} \tag{4}$$

$$y^i_{\min} + \frac{L}{2} \leq p_y \leq y^i_{\max} - \frac{L}{2}. \tag{5}$$

定义 $\mathcal{O}_{i,\ j} \coloneqq \mathcal{C}_i \cap \mathcal{C}_j$ 为走廊 i 和 j的交集。我们要求 $\mathcal{O}_{i,\ j} = \emptyset$ （若 $|i-j| >$为 $\mathbf{1}$），并且 $\mathcal{O}_{i-1,\ i}$ 足够大，能够容纳车辆$\forall i \in \{1, \ldots, n-1\}$。

构建 $\boldsymbol{C}$ 的第一步是计算一条由网格单元构成的路径，记作 g_i，其定义为

$$\begin{aligned} \mathcal{P} \coloneqq (g_0 \cdots g_m : g_i \text{ is free}, \\ D_M(g_i, g_{i+1}) = 1, \\ \boldsymbol{p}_0 \in g_0, \boldsymbol{p}_n \in g_m) \end{aligned} \tag{6}$$

其中， D_M 表示占据栅格中的曼哈顿距离。 $\mathcal{P}$ 通过执行广度优先搜索计算得到。定义集合

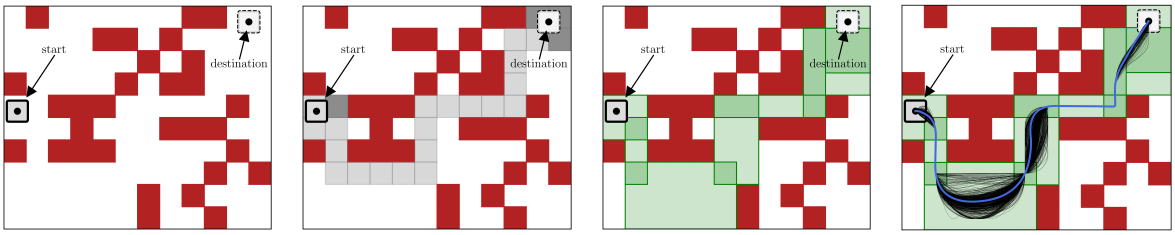
$$\mathcal{F}(\boldsymbol{p}) \coloneqq \left\{\boldsymbol{q} : |\boldsymbol{p} - \boldsymbol{q}| \leq \begin{bmatrix} \frac{W}{2} & \frac{L}{2} \end{bmatrix}^\top \right\}, \tag{7}$$

$$\mathcal{S}(\boldsymbol{p}) \coloneqq \{g_i : \exists \boldsymbol{q} \in \mathcal{F}(\boldsymbol{p}) \cap g_i\}. \tag{8}$$

$\mathcal{F}$ 表示车辆的占用轮廓，包含车辆位于 $\boldsymbol{p}$位置时占据的所有位置。 $\mathcal{S}$ 表示车辆位于 $\boldsymbol{p}$位置时被车辆（部分）占据的网格单元集合。为确保车辆在 4)-(5)中满足 $\boldsymbol{p}_0$ 和$\boldsymbol{p}_n$时， $\mathcal{F}(\boldsymbol{p}_0) \subseteq \mathcal{C}_0$ 以及 $\mathcal{F}(\boldsymbol{p}_n) \subseteq \mathcal{C}_{n-1}$ 必须成立。为此，我们将 $\mathcal{P}$ 扩展为包含 $\mathcal{S}(\boldsymbol{p}_0)$和 $\mathcal{S}(\boldsymbol{p}_n)$。由此得到如下定义

$$\mathcal{P}' \coloneqq \mathcal{S}(\boldsymbol{p}_0) \frown \mathcal{P} \frown \mathcal{S}(\boldsymbol{p}_n) \tag{9}$$

其中 $\frown$表示拼接。 $\mathcal{P}$ 和 $\mathcal{P}'$ 在图 2b中给出了一个示例环境。较深色的网格单元被添加，使得完整的

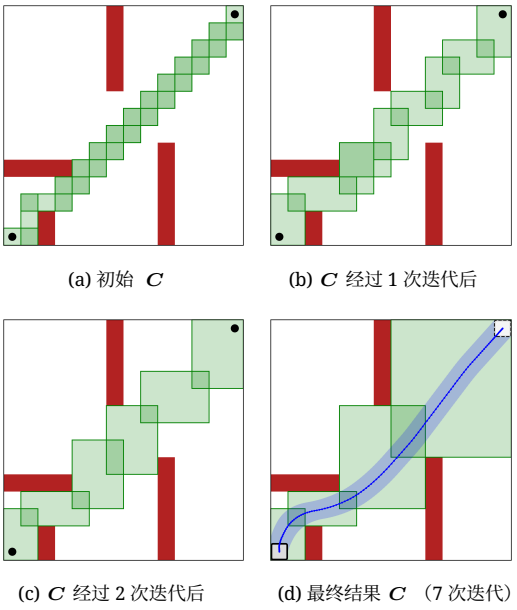


(a) (b) (c) (d)

图2：所提方法各个步骤的示意图。红色方块表示被障碍物占据的网格单元。图中还标出了起始位置、最终位置以及车辆占用空间。(a) 标出了起始位置和目标位置的环境。(b) 通过图搜索找到的路径以浅灰色表示，记为 $\mathcal{P}$。 $\mathcal{P}'$ 中不属于 $\mathcal{P}$ 的网格单元以深灰色表示。(c) 走廊序列 $\boldsymbol{C}$ 以浅绿色表示。两个相邻走廊的重叠部分以深绿色表示。(d) 黑线表示使用选定运动基元描述的500条轨迹（从$\boldsymbol{\Omega}_n$中随机采样）。蓝色轨迹为时间最优轨迹（$\Omega_n^* = \boldsymbol{\Phi}(\Pi^*)$，其中 $\Pi^* \in \mathcal{X}$）

车辆占用空间在 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_n$ 处均包含其中。由于对栅格单元维度的假设，在$\mathcal{P}'$中存在一条从 $\boldsymbol{p}_0$ 到 $\boldsymbol{p}_n$ 的可行路径。

$\boldsymbol{C}$ 通过将 $\mathcal{P}'$ 划分为一系列连续的网格单元区间得到，这些区间都位于同一行或同一列。为确保每个走廊尽可能宽，为车辆提供尽可能大的自由空间，采用迭代增长策略，通过检查每个走廊两侧栅格单元的占用状态来扩展 $\boldsymbol{C}$ 。每次扩展迭代后，如果 $\mathcal{C}_i$ ，则将走廊从序列中移除；如果 $\mathcal{C}_i \subseteq (\mathcal{C}_{i-1} \cup \mathcal{C}_{i+1})$ 或者$\mathcal{C}_{i-1} \cap \mathcal{C}_{i+1}$ 非空，也将其移除。该过程如图3所示。图2c展示了一个示例中构造的走廊。



(a) 初始 $\boldsymbol{C}$　(b) $\boldsymbol{C}$ 经过 1 次迭代后

(c) $\boldsymbol{C}$ 经过 2 次迭代后　(d) 最终结果 $\boldsymbol{C}$ （7 次迭代）

图3：示例环境中迭代增长策略的示意图。请注意，如果 $\mathcal{C}_i \subseteq (\mathcal{C}_{i-1} \cup \mathcal{C}_{i+1})$ 或 $\mathcal{C}_{i-1} \cap \mathcal{C}_{i+1}$ 非空，则会移除走廊 $\mathcal{C}_i$；这种情况在该环境中经常发生。走廊序列 $\boldsymbol{C}$ 以浅绿色显示。两个相邻走廊的重叠部分以深绿色显示。障碍物以红色显示。蓝线表示优化轨迹。

3.2 不考虑障碍物的解析解

假设走廊无限大，因此任意轨迹均无碰撞。此外，假设问题在每个维度上分别求解，得到最优运动时间 T_x^* andT_y^* ，并且不失一般性地假设 $T_x^* < T_y^*$。选择 $a_x(t) = 0 \quad \forall t \in (T_x^*, T_y^*]$。由于 $v_x(T_x^*) = 0$，这确保了 $p_x(T_y^*) = p_n^x$，并且 $v_x(T_y^*) = 0$。换言之，由于终端速度被视为零，两个维度上的运动可以完全解耦。

两个方向的解耦将问题简化为约束最短时间双积分器的运动，可使用庞特里亚金极大值原理完整求解，具体描述见 ˇ Feh´er、Straka 和 Sm´ıdl

(2017)。如果所得轨迹上的所有点都满足（4）和（5），且位于 $\boldsymbol{C}$ 中的某个走廊内，则已找到可行解。否则，将选择参数化基元并按后续章节所述进行优化。

3.3 参数化运动基元的定义

一维点到点运动的时间最优加速度曲线通常表示为

$$a(t) = \begin{cases} \alpha_k a_{\max} & t \in [t_0, t_1) \\ 0 & t \in [t_1, t_2) \\ \alpha_{k+1} a_{\max} & t \in [t_2, t_3) \end{cases} \tag{10}$$

其中 $\alpha_i \in \{-1, 1\}$ 可称为加速-匀速-减速加速度。

令 $\tau_i \coloneqq t_{i+1} - t_i$ 为各加速度阶段的持续时间。对该加速度曲线积分后，可得到由一段线性段、一段恒定段和另一段线性段组成的速度曲线。

进而得到位置曲线，其中包含一段抛物线段、一段线性段，随后是另一段抛物线段。这种一维运动可以完全用以下基元描述

$$\phi\left(\alpha_k, \alpha_{k+1}, p_k, v_k, \boldsymbol{\tau}\right) \coloneqq \left[p(t)\ v(t)\ a(t)\right]^\top \tag{11}$$

其中 $p(t_0) = p_k$，$v(t_0) = v_k$ 和 $\boldsymbol{\tau} \coloneqq \left[\tau_0\ \tau_1\ \tau_2\right]$是一个向量，用于收集一维单个基元中各加速度阶段的持续时间。示例如图 4。

二维运动基元因此定义为

$$\Phi\left(\boldsymbol{\alpha}_k, \boldsymbol{\alpha}_{k+1}, \boldsymbol{p}_k, \boldsymbol{v}_k, \boldsymbol{\tau}^x, \boldsymbol{\tau}^y\right) \coloneqq \begin{bmatrix} \phi\left(\alpha_k^x, \alpha_{k+1}^x, p_k^x, v_k^x, \boldsymbol{\tau}^x\right) \\ \phi\left(\alpha_k^y, \alpha_{k+1}^y, p_k^y, v_k^y, \boldsymbol{\tau}^y\right) \end{bmatrix}$$

令 $T_{(\cdot)} \coloneqq \sum_{i=0}^{2} \tau_i^{(\cdot)}$ 表示一维基元的总时间，则必须满足 $T \coloneqq T^x = T^y$。这一约束是确保基元在每个维度上同时到达终点所必需的。假设我们选择参数，使得 $p^x(T^x) = \tilde{p}^x$ 且 $p^y(T^y) = \tilde{p}^y$。在这种情况下，只有满足时间约束，车辆的轨迹才会实际经过点 $(\tilde{p}^x, \tilde{p}^y)$。此外，

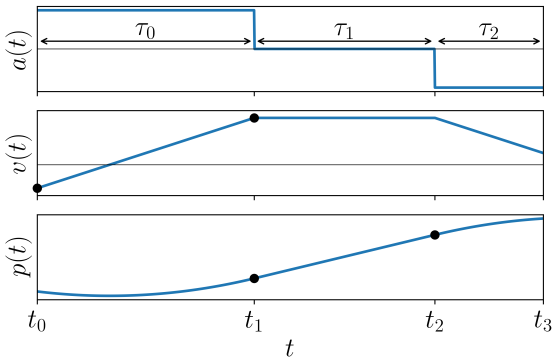


图 4：一维运动基元示意图 $\phi(1,\ -1,\ p_k,\ v_k,\ \boldsymbol{\tau})$。黑点表示施加盒约束的点，如第 3.5 节所述

对于 $t \in (\min(T^x, T^y), \max(T^x, T^y)]$，该基元将无法明确定义。令 $\boldsymbol{\lambda}_i \coloneqq \left[\tau_i^x\ \tau_i^y\right]^\top$。现在我们可以定义函数

$$f\left(\Phi\right) \coloneqq \begin{bmatrix} \boldsymbol{p}_k + \boldsymbol{v}_k T + a_{\max}\boldsymbol{\alpha}_k\boldsymbol{\lambda}_0\left(\boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2\right) \\ \qquad +0.5 a_{\max}\left(\boldsymbol{\alpha}_k\boldsymbol{\lambda}_0^2 + \boldsymbol{\alpha}_{k+1}\boldsymbol{\lambda}_2^2\right) \\ \boldsymbol{v}_k + a_{\max}\boldsymbol{\alpha}_k\boldsymbol{\lambda}_1 + a_{\max}\boldsymbol{\alpha}_{k+1}\boldsymbol{\lambda}_3 \end{bmatrix}$$

它对该基元的位置和速度进行解析积分。

完整轨迹由一系列 n参数化基元构成：每条走廊对应一个基元Φ。第 3.4 节中介绍了一种确定 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 和 $\{\boldsymbol{\alpha}_k\}_{k=0}^{n}$ 的策略，因此我们可以假设这些量已知。每个基元仅剩持续时间 $\boldsymbol{\tau}_k^x$和 $\boldsymbol{\tau}_k^y$ 以及初始速度 $\boldsymbol{v}_k$有待确定。这意味着还有 $8n$ 个待定参数，可将其收集到向量 $\Pi \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}^{6n}$ 中，如下所示：

$$\Pi \coloneqq \left[\boldsymbol{v}_0 \cdots \boldsymbol{v}_{n-1}\ \boldsymbol{\tau}_0^x \cdots \boldsymbol{\tau}_{n-1}^x\ \boldsymbol{\tau}_0^y \cdots \boldsymbol{\tau}_{n-1}^y\right] \tag{12}$$

它完全确定了这组 n 基元

$$\boldsymbol{\Phi}_n(\Pi) \coloneqq \{\Phi_k\left(\boldsymbol{\alpha}_k, \boldsymbol{\alpha}_{k+1}, \boldsymbol{p}_k, \boldsymbol{v}_k, \boldsymbol{\tau}_k^x, \boldsymbol{\tau}_k^y\right)\}_{k=0}^{n-1}. \tag{13}$$

通过对轨迹施加必要约束，可以消除部分自由度。更具体地说，每个基元都有一个时间约束，要求两个维度上的持续时间之和相等，如前文所述（n约束）。除此之外，每个基元都必须

到达已知的最终位置，并确保速度连续或达到终端速度（$4n$ 约束）。最后，给定初始速度（2 个约束）。由此得到 $3n-2$ 个有效自由度，并将可行参数集定义为

$$\begin{aligned}\mathcal{X} \coloneqq \{\Pi \in \mathbb{R}^{2n} \times \mathbb{R}^{6n}_{\geq 0} :\ & \boldsymbol{v}_0 = \bar{\boldsymbol{v}}_0, \\ & \textstyle\sum_{i=0}^{2} \tau^{x}_{k,i} = \sum_{i=0}^{2} \tau^{y}_{k,i} \quad k = 0...n-1, \\ & f(\Phi) = \begin{bmatrix}\boldsymbol{p}_{k+1} & \boldsymbol{v}_{k+1}\end{bmatrix}^{\top} \quad k = 0...n-1\}.\end{aligned} \tag{14}$$

请注意，我们没有通过这些约束显式消去 Π中的参数。相反，所有参数都保留为优化变量，并受到上述约束的限制，如公式19b所示。

我们考虑如下定义的轨迹集合$\boldsymbol{\Omega}_n$

$$\boldsymbol{\Omega}_n \coloneqq \{\boldsymbol{\Phi}_n(\Pi) : \Pi \in \mathcal{X}\}. \tag{15}$$

图2d中的黑线展示了从$\boldsymbol{\Omega}_n$ 中采样得到的轨迹，所用环境为一个示例环境。

3.4 参数化运动基元的选择

根据走廊序列$\boldsymbol{C}$的几何形状，可以通过启发式方法确定值 $\{\boldsymbol{\alpha}_k\}_{k=0}^{n}$ 和点 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 。 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_n$ 则由问题定义得出。

这些启发式方法基于这样一种假设：车辆从 $\mathcal{C}_{k-1}$ 移动到 $\mathcal{C}_k$时会转弯。由于膨胀前构造的走廊采用90度转弯，这一假设是合理的。因此，我们选择 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 来表示转弯内侧的顶点。该过程如图5所示。

顶点 $\boldsymbol{p}_k$ 从一组候选点中选取，这些候选点表示为 $\mathcal{W}_k$。为构造 $\mathcal{W}_k$，我们将 $\mathcal{O}_{k-1}$在水平方向两侧各缩小k ，并在垂直方向两侧各缩小 $W/2$ 。集合 $L/2$表示$\mathcal{W}_k$ 缩小版本的角点$\mathcal{O}_{k-1,\ k}$。此外，对于到某条走廊边缘的水平距离大于 $W/2$ ，且到同一走廊边缘的垂直距离大于 $L/2$ 的点，将从 $\mathcal{W}_k$ 中移除，以提高启发式方法的鲁棒性。 $\mathcal{W}_k$ 在图5中以蓝色圆圈表示。用于确定

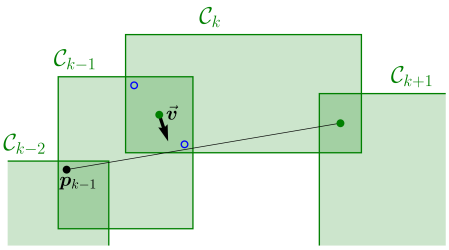


图 5：确定 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 的启发式方法示意图。 $\boldsymbol{C}$ 以绿色显示，绿色圆点表示 $\mathcal{O}_{k-1,\ k}$ 和 $\mathcal{O}_{k,\ k+1}$的中心。集合 $\mathcal{W}_k$ 中的点以蓝色圆圈显示。右下方的蓝色圆圈将被选为 $\boldsymbol{p}_k$

$\boldsymbol{p}_k$ 状态

$$\boldsymbol{p}_k = \underset{\boldsymbol{q} \in \mathcal{W}_k}{\arg\min} ||\boldsymbol{q} - \boldsymbol{i}_k||_2 \quad k = 1...n-1. \tag{16}$$

其中， $\boldsymbol{i}_k$ 表示转弯内侧的一个点，定义为 $\boldsymbol{i}_k$:= center($\mathcal{O}_{k-1,\ k}$) $+ \mu\vec{\boldsymbol{v}}$，且 $\vec{\boldsymbol{v}}$ 是一个单位向量，与连接 $\boldsymbol{p}_{k-1}$ 和 center($\mathcal{O}_{k,\ k+1}$) 的直线垂直。选择$\vec{\boldsymbol{v}}$ 的符号，使其从 center($\mathcal{O}_{k-1,\ k}$) 指向这条连接线。标量 $\mu \in \mathbb{R}_{>0}$是一个无量纲常数，必须选得足够大，以确保$\boldsymbol{i}_k \notin \mathcal{O}_{k-1,\ k}$ ，从而选出正确的候选航路点。只要满足这一条件， μ的具体取值就不会影响该方法的性能。在本工作中，我们设置 $\mu = 20$ ，使得 $\mu\vec{\boldsymbol{v}}$ 的长度为 20 米，大于基准测试中考虑的环境维度，因此也大于任何走廊重叠。换句话说，我们选择 $\boldsymbol{p}_k$ 作为 $\mathcal{W}_k$ 中最靠近转弯内侧的点。

Dmitruk 和 Kaganovich (2011) 讨论了通过一系列点的时间最优轨迹的解析解 $\{\boldsymbol{p}_k\}_{k=0}^{n}$ ，该解使用庞特里亚金极大值原理（见其论文中的例 4）推导得出。研究表明，加速度是分段常数函数，并且 $|a_x(t)| = |a_y(t)| = a_{\max} \quad \forall t \in [0, T]$。加速度的符号由庞特里亚金极大值原理最优性条件所导出的协态轨迹的符号决定。然而，求解非线性方程组并非易事，因为结果对协态变量的初始值十分敏感，而这些变量没有直接的物理意义。此外，该分析并未

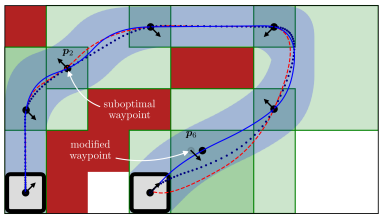


图 6：$\{\boldsymbol{p}_k\}_{k=0}^n$ 和 $\{\boldsymbol{\alpha}_k\}_{k=0}^n$的选择与修改示意图。 $\boldsymbol{C}$ 以绿色显示。黑色箭头表示 $\{\boldsymbol{\alpha}_k\}_{k=0}^n$。红色虚线：OCP 解。蓝色点线：中间的 PMP 结果。蓝色实线：最终的 PMP 解

考虑速度或位置约束。因此，在本研究中，点 $\{\boldsymbol{p}_k\}_{k=0}^n$ 处的加速度符号（记作 $\{\boldsymbol{\alpha}_k\}_{k=0}^n$）通过启发式方法确定如下：

$$\boldsymbol{\alpha}_k = \begin{cases} \text{sign}(\boldsymbol{p}_1 - \boldsymbol{p}_0) & k = 0 \\ \text{sign}(\boldsymbol{p}_{n-1} - \boldsymbol{p}_f) & k = n \\ \text{sign}(\boldsymbol{p}_k - \text{center}(\mathcal{O}_{k-1,k})) & \text{otherwise.} \end{cases} \tag{17}$$

在某些情况下，需要修改 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 和 $\{\boldsymbol{\alpha}_k\}_{k=1}^{n-1}$。如果车辆能够在走廊内从 $\boldsymbol{p}_{k-1}$ 移动到 $\boldsymbol{p}_{k+1}$，则“车辆从一个走廊移动到另一个走廊时会转弯”这一假设不成立。在这种情况下，启发式方法会产生 $\boldsymbol{p}_k$ 和 $\boldsymbol{\alpha}_k$，其结果可能是次优的，甚至不可行。因此，将加速度的符号翻转。形式上，如果 $\boldsymbol{q} \in \bigcup_{i=0}^n \mathcal{C}_i \forall \boldsymbol{q} \in \mathcal{F}(\boldsymbol{p}_{k-1} + s($ $\boldsymbol{p}_{k+1} - \boldsymbol{p}_{k-1}))$ 且 $\forall s \in [0,1]$，则 $\boldsymbol{\alpha}_k \leftarrow -\boldsymbol{\alpha}_k$ 其中 $\mathcal{F}$ 的定义与式（7）相同。此外，将点 $\boldsymbol{p}_k$ 放宽为$\mathcal{O}_{k-1,k}$中的任意一点；我们设定 $\boldsymbol{p}_k \leftarrow \boldsymbol{p}_k + \boldsymbol{\Delta}_k$，其中$\boldsymbol{\Delta}_k$ 是一个额外的受约束自由度，使 $\boldsymbol{p}_k$ 能够在重叠区域内移动。图6展示了一个示例环境，其中从点 $\boldsymbol{p}_5$ 到目的地存在一条可行直线，因此会触发 $\boldsymbol{\alpha}_6$ 翻转，并使$\boldsymbol{p}_6$ 可移动。注意，若不进行此更改，则$\boldsymbol{\Omega}_n = \emptyset$，因为不存在可行的基元组合。

3.5 参数优化

为了找到最优轨迹，必须找到使下式最小化的 $\Omega_n^* = \boldsymbol{\Phi}_n(\Pi^*) \in \boldsymbol{\Omega}_n$ $(\Pi^* \in \mathcal{X})$：

轨迹的总时间

$$J(\Pi) := \sum_{k=0}^{n-1} \left(\sum_{i=0}^{2} \tau_{i,k}^x \right). \tag{18}$$

为此，求解以下优化问题。

$$\begin{aligned} \min_{\boldsymbol{z}} \quad & J(\Pi) && \text{(19a)} \\ \text{s.t.} \quad & \Pi \in \mathcal{X}, && \text{(19b)} \\ & 0 \le g_k(\boldsymbol{p}_k, \boldsymbol{v}_k, \boldsymbol{\tau}_k) \quad k = 0 \ldots n-1, && \text{(19c)} \\ & |\alpha_0^F| \le 1, && \text{(19d)} \\ & |\alpha_n^F| \le 1 && \text{(19e)} \end{aligned}$$

其中 $\boldsymbol{z} = \begin{bmatrix} \Pi & \alpha_0^F & \alpha_n^F & \boldsymbol{\Delta}_1 \cdots \boldsymbol{\Delta}_{n-1} \end{bmatrix}$。

使用不等式 (19c) 强制施加走廊边界和速度限制。为了施加速度限制，可以利用速度曲线的结构。设 $v_k' := v_k + a_{\max}\alpha_k\tau_{k,0}$ 为经过第一个加速度阶段后得到的速度。由于速度曲线的初始段是线性的，$v(t) \in [v_k, v_k']$ 在这一段中成立。由于第二段为常数，该不等式仍然成立。在第三段（同样为线性）中，$v(t) \in [v_k', v_{k+1}]$ 成立。因此，只需施加 $\{v_k\}_{k=0}^{n-1} \in [-v_{\max}, v_{\max}]$ 和 $\{v_k'\}_{k=0}^{n-1} \in [-v_{\max}, v_{\max}]$，即可得到 $4n$ 个约束。注意，$\boldsymbol{v}_n = 0$。图4 中的黑点表示这些点。

同样，可以利用位置剖面的结构，以稀疏方式施加走廊边界约束。只要在基元的整个持续时间内 $|v(t)| > 0$，位置剖面就是单调的，这意味着只要基元的端点可行，便可保证基元可行。根据构造，这一点成立，因为选定的航路点 $\{\boldsymbol{p}_k\}_{k=0}^n$ 均满足走廊边界。若 $v(t) = 0$ 在某个抛物线段内，便可能出现走廊不可行性。第3.6节将对此进行处理。为防止求解器探索不可行解，从而提高鲁棒性，我们在滑行阶段的端点处添加走廊约束。图4中的黑点表示了这一点。请注意，使用矩形走廊可以分别考虑每个维度上的位置边界；若采用椭

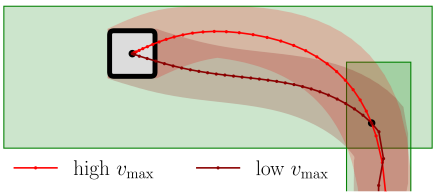


图 7：不同速度限制下 OCP 的解，显示了 y 方向上的初始加速度符号发生变化

圆（例如Deits 和 Tedrake(2015年)提出的椭圆），情况则并非如此。

如图 7所示，沿行驶距离最长的方向（称为自由方向）的初始加速度符号取决于车辆参数，甚至可能取决于走廊维度，因此 α_0^F 被保留为优化变量。同理，α_n^F 也被保留为优化变量。这一点体现在（19d）-（19e）中。

3.6 优化问题的公式细节

本节讨论优化问题公式中的一些细节，这些细节能够提高公式的鲁棒性或改善解的质量。

首先，自由方向上的初始和最终加速度分别记为 α_0^F 和α_n^F，并保留为优化变量。我们不直接强制执行（19d）和（19e）中的约束，而是对这些约束进行松弛；实践发现，这有助于更鲁棒地收敛到一个解。松弛后的约束为

$$|\alpha_0^F| \leq 1 + \left(s_a^0\right)^2, \tag{20}$$
$$|\alpha_n^F| \leq 1 + \left(s_a^n\right)^2. \tag{21}$$

在目标函数中加入惩罚项 $w||\boldsymbol{s_a}||_2^2$ where $w = 10^3$。

其次，当求解问题（19）后，会检查所得轨迹是否存在走廊可行性问题。如果 $v(t) = 0$ 在某个抛物线段内，则位置曲线上会到达一个极值点。如果该极值点违反走廊边界，就会添加附加约束并重新求解问题。例如，假设速度在第一个运动基元段 k 和 $p_k(t)$违反了走廊边界。在这种情况下，对于 $t = -v_k/(\alpha_k a_{\max})$，点 $p_k + v_k t + 0.5\alpha_k a_{\max} t^2$ 也会被约束在走廊边界内。这样可以确保位置曲线极值点满足走廊约束，从而使整个运动基元都满足走廊约束。由于这些约束在许多情况下会导致保守性，因此不能在求解问题之前添加。

第三，在获得最优且可行的轨迹 $\Omega_n^* \in \boldsymbol{\Omega}_n$后，可以检测出次优运动基元。更准确地说，如果发现车辆沿直线滑行经过航路点 $\boldsymbol{p}_k$，说明它没有利用该基元所允许的加速度，这表明施加的加速度曲线与真正的时间最优解不匹配。因此，我们设置$\boldsymbol{\alpha}_k \leftarrow -\boldsymbol{\alpha}_k$。请注意，对于新的曲线，当前解仍然可行，因为车辆滑行经过了 $\boldsymbol{p}_k$。对于新的运动基元，可以再次求解问题（19），同时将先前的解作为可行的初始猜测。这个过程有望找到更优的解，如图 6所示。第一个解以虚线表示，车辆滑行经过航路点 $\boldsymbol{p}_2$。更改 $\boldsymbol{\alpha}_2$后，得到了一条更快的新轨迹。

最后，优化问题原始变量的初始化并非易事，因为没有直观的方法来初始化加速度的持续时间。优化问题在每个走廊中通过设置$\tau_{0,k}^x = \tau_{0,k}^y = \tau_k'$, $\tau_{1,k}^x = \tau_{1,k}^y = 7\tau_k'$和$\tau_{2,k}^x = \tau_{2,k}^y = 0.2\tau_k'$进行初始化，其中 τ_k'是某个大于0.05 s 的数，且 $\boldsymbol{v}_k$ 的选取使得该基元到达下一个航路点 $\boldsymbol{p}_{k+1}$。

3.7 完整概述

算法1 总结了 PMP。第 1 行展示了走廊序列的构建过程。接下来，在第 2-4行中，如果解析解可行，则返回该解。第 5行使用启发式方法选择基元。随后在第 19行求解优化问题（8）。如第 3.6 节所述，该过程在 while 循环中进行，以确保走廊可行性（第 9-12行），并剔除次优运动基元（第 13-17行）。

算法 1 PMP 概述

需要： $\boldsymbol{p}_0$、$\boldsymbol{p}_n$、$\bar{\boldsymbol{v}}_0$ 以及环境的栅格地图-ment
确保： 时间最优、无碰撞轨迹

构建走廊序列 $\boldsymbol{C}$ ▷ 第3.1节
如果解析解可行**则** ▷节 3.2
  返回解析解
结束 if
确定 $\{\boldsymbol{p}_k\}_{k=1}^{n-1}$ 和 $\{\boldsymbol{\alpha}_k\}_{k=0}^{n}$ ▷第3.4
已完成 ← False
当未完成**时**
  求解问题（19） ▷第 3.5节
  当违反走廊边界 **时** ▷ 第 3.6节
    添加附加约束
    求解问题（19）
  结束 while
  如果 找到次优原语 **则** ▷ 第 3.6 节
    修改原语选择
  else
    完成 ←True
  结束 if
结束 while
返回解

3.8 方法局限性

所提方法存在一些局限性，本节将对此进行讨论。首先，走廊序列基于距离最优图搜索，因此可能产生非时间最优的走廊序列，如图 11a所示，VP-STO 在走廊序列之外找到了更优的轨迹。解决这一局限性的一种方法是探索属于不同同伦类的初始路径 $\mathcal{P}$。这些不同的同伦类可以通过在环境地图上构建 Voronoi图来寻找，例如 Demeester 等人（2008）所示。

其次，假设环境表示中的网格单元足够大，能够容纳车辆。这确保车辆始终可以从一个走廊转移到另一个走廊，但也对障碍物尺寸设定了下限。由于所提方法主要针对结构化环境，因此这一假设是合理的。请注意，对于所考虑的具体应用，障碍物通常是环境边缘、故障瓦片或静止移动器，它们都符合这一假设。然而，在存在不规则障碍物的随机杂乱环境中，仅凭这一假设无法确保走廊之间有足够的重叠，因此必须通过其他方式来实现。此外，在这类环境中，采用矩形走廊可能缺乏合理性，也可能无法恰当地表示可用的自由空间。

第三，选择运动基元的启发式方法无法提供理论保证。事实上，如图 11b所示，这些启发式方法可能会失效并生成次优轨迹。此外，这些启发式方法无法保证存在可行参数集。然而，第 4 节的基准测试结果表明，这些启发式方法在实践中是有效的。

最后，公式（2）使用无穷范数施加速度和加速度约束。当车辆在各个维度上的运动完全独立时，这种处理是准确的；但如果限制的是总速度或总加速度（欧几里得范数），而非无穷范数，则可能导致次优结果。即便如此，选择这种范数仍可将各维度上的加速度曲线解耦，从而显著缩短计算时间。

4 基准测试

为了评估所提方法（PMP），我们将其与第 4.1 节所述的方法进行比较。所有方法均用于计算结构化环境和随机杂乱环境中的轨迹。实现细节说明见第 4.2 节，基准测试本身详述于第 4.3 节。最后，我们在第 4.4 节比较不同方法构建动力学可行路径的效果。

4.1 所对比基准测试方法概述

所提方法与另外两种基于优化的方法进行了比较，这两种方法均通过同一走廊序列 $\boldsymbol{C}$进行规划。由此可以验证走廊内参数化运动基元的选择与优化。此外，我们还与不使用走廊序列 $\boldsymbol{C}$的VP-STO进行了比较。

4.1.1 完整最优控制问题（OCP）

可以构建一个完整的优化问题，其中轨迹采用

多重打靶方法进行离散化。最优控制问题（OCP）采用带时间缩放的多阶段表述，其中每个阶段表示穿过一个走廊的运动。速度和位置边界适用于每个离散点。该表述与Bos等人（2023）的研究中的问题（3）类似。为进行公平比较，我们对动力学采用解析积分器。每个走廊的网格点数量会影响问题规模，也会影响轨迹的可行性和最优性。更精细的网格更适合捕捉砰-砰解，也更不容易产生在网格点之间违反走廊边界的轨迹。然而，这会使问题规模增大，求解时间更长。本研究发现，每个走廊采用30个网格点是较为合理的折衷。

4.1.2 OMG-tools

OMG-tools 是一个工具箱，它使用样条表示生成平滑轨迹（Mercy等人，2018年，2016）。位置和速度均用样条表示，并对速度的导数施加加速度约束。

4.1.3 基于路点的随机轨迹优化（VP-STO）

VP-STO（Jankowski等人，2023）并非沿走廊序列进行规划，而是在空间中采样若干点（N_{via}），并通过 OCP 的解析解将这些点连接起来。使用随机黑箱无梯度优化方法，调整途经点以最小化通用代价函数。增大（N_{via}），即可表示更复杂的轨迹。此外，增大种群规模（即所考虑的轨迹数量），可以提高获得全局最优解的概率。然而，增大这些参数的值也会增加计算时间。在此基准测试中，我们采用的种群规模为 25，最大迭代次数为 3000，σ_{init} 设为 1.5，并使用不同的 N_{via} 值。我们将此方法称为 VP-STO-N_{via}。

Jankowski 和 Bruderm¨uller (2024年) 提供了一个 Python 开源实现。然而，该实现仅用于离线场景，而且很可能有很大的提速空间。作者还提出了在线VP-STO，并将其用于真实世界实验。在这种设置下，只考虑有限数量的途经点；轨迹不会一直求解到收敛，而是在达到MPC截止时间时中止，因此牺牲了最优性。此外，作者指出，当目标位置发生变化时，机器人有时无法足够快地找到有效运动，这表明该方法依赖于热启动过程。在本研究中，我们采用 VP-STO，并运行至收敛（或达到最大迭代次数）。

4.2 实现细节

所有优化问题均使用`CasADi`（Andersson等人，2018年）进行建模。对于 OCP，会针对每一种待考虑的走廊数量离线创建一个 opti 函数对象。这些优化问题使用 FATROP（Vanroye等人，2023）求解，这是一种基于 IPOPT 的最先进 OCP 求解器（W¨achter与Biegler，2006）。

同样，也会为 PMP 创建 opti 函数。然而，对于 n 条走廊，存在 $n-1$ 个可能的可移动航路点，因此需要枚举该优化问题的2^{n-1} 种变体。不过，这些问题规模较小，只要有一种允许足够多可移动点的变体可用，就不必枚举每一种可能的变体。这些优化问题同样使用 FATROP 求解。

OMG-tools 仅提供`Python` 版本，我们使用 IPOPT 及线性求解器 MA27 来求解这些优化问题。代码在第 12 代 Intel(R) Core(TM) i7-1280P 处理器上运行。需要注意的是，OMG-tools 的实现很可能仍可进一步提速。然而，OMG-tools 所构建的优化问题目前不符合 FATROP 所要求的问题结构，因此必须使用 IPOPT 求解。

4.3 基准测试描述

所考虑的方法应用于结构化环境和随机杂乱环境中。图10展示了结构化环境。该环境包含一些排列整齐的墙壁和障碍物。在此环境中，计算了 100 条具有不同参数值、起始位置和目的地的轨迹

。参数 $v_{\max}$ 和 $a_{\max}$ 分别从 $[0.5, 2.0]m/s$ 和 $[2.0, 6.0]m/s^2$ 中均匀采样。可行点 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_n$ 也采用均匀采样，但只有在满足 $||\boldsymbol{p}_0 - \boldsymbol{p}_n||_2 > 5W$ 时才会被接受，其中 W 为车辆宽度（此处等于车辆长度），以避免轨迹过短。

除了该结构化环境外，还创建了 100 个随机环境，其中栅格单元包含障碍物的概率设为 0.1。对于更高的障碍物密度，也发现了类似的结果。参数值、起始位置和终止位置均按照结构化环境中所述随机选取。

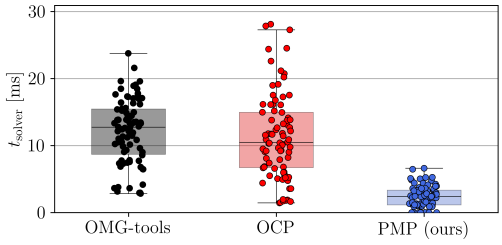
4.4 结果

对于基准测试中的每条轨迹，我们考察计算轨迹所需的时间、所计算轨迹的运动时间（该时间经过最小化），以及该方法的鲁棒性。结果汇总于表 1，并将在接下来的子区段中详细说明。

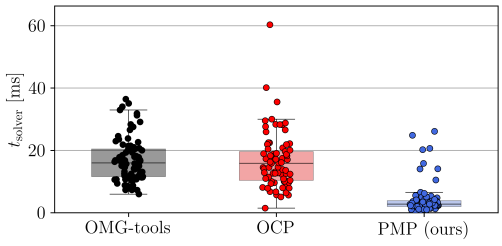
4.4.1 计算时间讨论

总计算时间（t_{total}）包括求解器时间（t_{solver}），即求解优化问题所需的时间；此外还包括走廊构建、可能涉及的运动基元选择、初始猜测设置以及轨迹采样。表1 给出了两种环境下 t_{solver} 和 t_{total} 的平均值和最坏情况值。PMP 求解优化问题的速度约为 OCP 或 OMG-tools 的 4 倍。需要注意的是，OMG-tools 的总时间显著较长，这是由于其优化问题构建效率低下，采用更好的实现方式即可改善。图8 展示了基于走廊的方法的求解器时间。在非结构化环境中，PMP 在某些情况下的求解器时间较长。这是因为，在从一个走廊转移到下一个走廊时车辆会转弯这一隐含假设，在非结构化环境中并非总是成立。这可能导致基元选择次优；检测到这一情况后，需要重新求解问题予以纠正，但代价是计算时间增加。

在结构化环境中，PMP 使用解析解求解了该问题。



（a）结构化环境



（b）非结构化环境

图 8：基于优化的方法在基准测试上的求解时间（t_{solver}）比较

以规划穿过第 3.2 节所述走廊序列，在 10 个案例中。在这 10 个案例中，有 8 个案例的走廊序列仅由一个走廊组成。其余 2 个案例的走廊序列则由两个走廊组成。在非结构化环境中，PMP 在 5 个案例中使用了解析解。在其中 4 个案例中，走廊序列由两个走廊组成，只有 1 个案例中仅有一个走廊。

VP-STO 的计算时间显著高于基于走廊的方法。此外，途经点越多，计算时间越长。这些结果并不完全具有代表性，因为正如第 4.1.3 节所述，该实现很可能还可以显著加速。

4.4.2 最优性讨论

为评估所生成轨迹的最优性，我们考察计算得到的运动时间t_{move}。表1展示了所有轨迹的平均运动时间，以及以 OCP 为基线时运动时间相对误差的中位数和标准差。图9展示了该相对误差的箱线图和单个样本。PMP 平均仅比 OCP 慢不到 0.5%，在所有方法中速度最快

表1：对100条轨迹进行比较，这些轨迹具有随机起始位置、目的地以及 $a_{\max}$ 和$v_{\max}$的取值

(a) 结构化环境

	PMP（我们的方法）	OCP	OMG-tools	VP-STO-5	VP-STO-10	VP-STO-15
平均 t_{solver} [毫秒]	**2.38**	11.33	12.03	(8.28e+03)	(2.30e+04)	(4.92e+04)
最大 t_{solver} [毫秒]	**6.61**	28.09	21.58	(3.12e+04)	(1.22e+05)	(1.57e+05)
平均 t_{total} [毫秒]	**4.30**	15.09	146.15	(8.42e+03)	(2.31e+04)	(4.94e+04)
最大 t_{total} [毫秒]	**9.83**	33.18	710.91	(3.17e+04)	(1.22e+05)	(1.57e+05)
平均 t_{move} [秒]	1.844	**1.839**	1.957	2.086	1.894	1.858
$\varepsilon_{\text{move}}$ [%]	0.0 ± 1.3	-	6.0 ± 4.9	6.7 ± 14.6	1.9 ± 11.8	0.7 ± 11.3
平均 $t_{\text{total}} + t_{\text{move}}$ [秒]	**1.849**	1.854	2.103	10.508	25.006	51.249
# 不可行情况	**0**	13	**0**	**0**	**0**	**0**
# 求解器故障	1	**0**	1	5	6	10

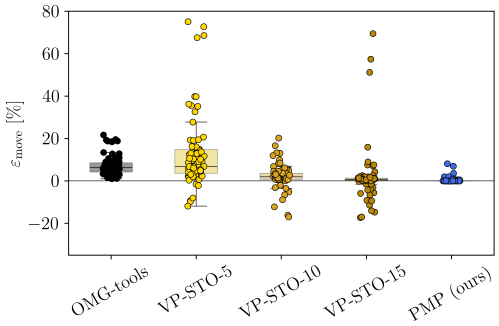
(b) 非结构化环境

	PMP（我们的方法）	OCP	OMG-tools	VP-STO-5	VP-STO-10	VP-STO-15
平均 t_{solver} [毫秒]	**4.28**	16.68	16.89	(3.13e+03)	(8.93e+03)	(1.85e+04)
max t_{solver} [ms]	**26.10**	60.29	36.43	(2.27e+04)	(4.01e+04)	(5.53e+04)
平均 t_{total} [毫秒]	**6.42**	20.90	181.51	(3.25e+03)	(9.05e+03)	(1.86e+04)
最大 t_{total} [毫秒]	**29.00**	64.37	648.17	(2.29e+04)	(4.02e+04)	(5.54e+04)
平均 t_{move} [秒]	1.709	**1.703**	1.785	1.893	1.724	1.694
$\varepsilon_{\text{move}}$ [%]	0.0 ± 1.7	-	4.2 ± 3.2	6.1 ± 19.9	0.9 ± 13.0	0.5 ± 11.2
平均 $t_{\text{total}} + t_{\text{move}}$ [秒]	**1.716**	1.724	1.966	5.146	10.774	20.297
# 不可行情况	**0**	7	**0**	**0**	**0**	**0**
# 求解器故障	3	**0**	4	5	3	3

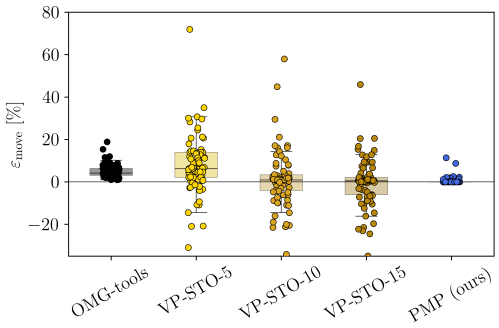
在两类环境中平均而言都是如此。OMG-tools 比 OCP 慢约 6%，这是由于其使用平滑样条表示控制量所致。在图9中可以清楚地看出，OMG-tools 始终比 OCP 略慢。PMP 大多收敛到与 OCP 相同的结果 t_{move} ，但在某些情况下可能产生次优结果。

不受走廊序列约束的 VP-STO 平均比 OCP 慢 16%，对于 $N_{\text{via}} = 5$。当 N_{via} 大于 10 时，运动时间的中位相对误差（$\varepsilon_{\text{move}}$）接近于零，表明途经点数量越多，产生的轨迹越好。这在图11d中也可以看到，所有方法都收敛到大致相同的轨迹。尽管如此，VP-STO-5 计算出一条更慢的轨迹，其原因在于

途经点通过平滑轨迹连接，这些轨迹并未显式优化时间。为实现时间最优性，需要更多的途经点。相对误差的较大标准差 t_{move} 对于 VP-STO 而言，这是因为该方法经常找到不同的轨迹。图 9 表明，尽管 VP-STO 能够收敛到全局最优解，但并不能保证这一点。在图10a 中，例如，VP-STO 收敛到了不同于基于走廊的方法所得到的局部最优解，且这些解更差。通过增大种群规模和初始采样标准差，可以提高 VP-STO 收敛到全局最优解的概率，但代价是更高的计算时间。此外，不同数量的途经点也会影响



(a) 结构化环境



(b) 非结构化环境

图9：所有环境下行驶时间的相对误差。将 OCP 视为基线

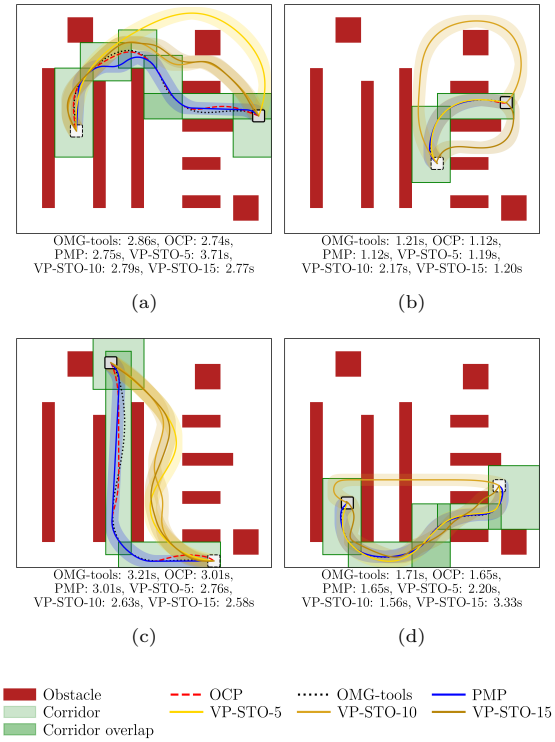


图10：结构化环境中的一些轨迹示例。图中显示了 $\boldsymbol{p}_0$（实线）和 $\boldsymbol{p}_n$（虚线）处的车辆占用 footprint。半透明阴影区域显示了车辆占用空间的轨迹

收敛性，如图 10b所示。图 10c展示了一个示例，其中 VP-STO 比基于走廊的方法收敛到了更好的局部最优解。图10d 展示了一个 VP-STO-10 未能找到可行解的情况。相反，它收敛到了一个局部最优解，在该解中，算法通过直穿墙壁来最小化碰撞。这是由于初始样本集质量较差，其中不包含绕过墙壁的轨迹。请注意，在此环境中，VP-STO-15 同样无法找到无碰撞轨迹。

在非结构化环境中也可以观察到类似现象，其中一些轨11中展示了迹。在图 11a中，走廊序列选取了一条需要在狭窄通道中转弯的轨迹，而 VP-STO 找到的轨迹则快得多。然而，在图 11b 和 11c中，VP-STO 难以找到一条穿过狭窄通道的轨迹，而在这些情况下，这种轨迹反而更快。这是因为采样到一条可行的穿过狭窄通道的轨迹的概率很低。

4.4.3 鲁棒性讨论

检查以100赫兹采样的轨迹是否发生碰撞后发现，OCP 解在结构化环境中有15次违反走廊边界，在非结构化环境中有7次。这是因为约束只施加在离散点上。若将每个走廊的网格点数量从30增加到60，则在结构化环境中仅发现2例走廊约束不可行的情况，而在非结构化环境中则没有。不过，这一调整会使 t_{solver} 的平均值增加一倍以上。与其增加网格点数量，不如额外考虑一定的余量来满足走廊边界要求，这也会影响 t_{move}。其他方法都能持续满足位置约束。

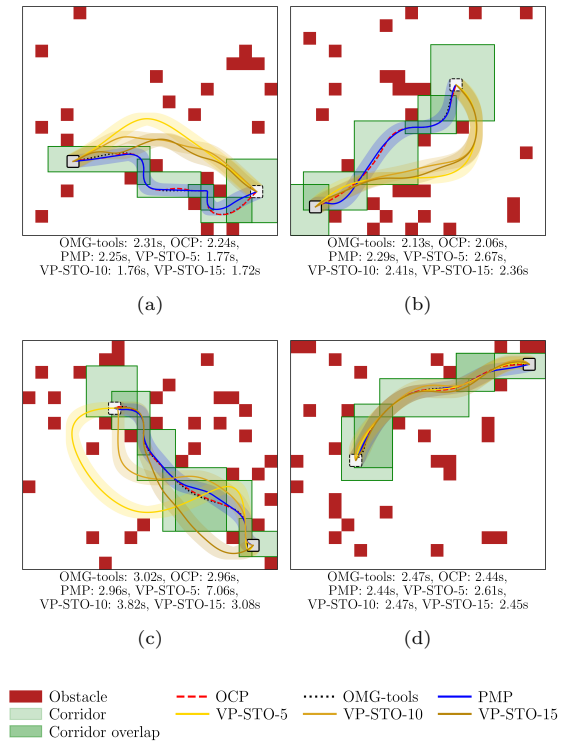


图11：非结构化环境中的一些示例轨迹。车辆在 $\boldsymbol{p}_0$（实线）和$\boldsymbol{p}_n$（虚线）时的占用 footprint 如图所示。半透明阴影区域显示了车辆占用空间的轨迹

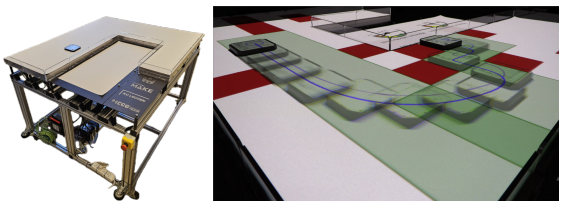
除 OCP 外，所有方法在某些情况下都无法找到解。对于基于走廊的方法，具体而言是 PMP 和 OMG-tools，这种情况在非结构化环境中更常见。在这种情况下，可以计算 $\boldsymbol{C}$ 各子区段内的轨迹，再将它们拼接起来。如果每个维度上的制动距离（由$\left(v^{(\cdot)}\right)^2/(2a_{\max})$给出）不超过走廊边界的可用余量，则总能通过解析方法求得穿过单个走廊的轨迹。如果规划器求解失败，且初始速度过高，以致无法再避开走廊边界，就需要采取紧急行为使车辆停下来。不过，由于所提方法计算的是终端速度为零的完整轨迹，只有在走廊序列突然发生变化时，车辆才可能在没有预先计算好的安全轨迹可用的情况下遇到这种状况。对于 VP-STO，求解器故障是指车辆移动时得到的解穿过障碍物，例如图 10d所示。这些失败在结构化环境中更常见，因为其中存在大型障碍物，且轨迹必须穿过狭窄通道。在此处考虑的非结构化环境中，VP-STO 往往会收敛到一条避开狭窄通道的轨迹，如图 11b所示。

5 实验验证

我们使用 Beckhoff Automation 的 XPlanar 移动器系统作为实验验证平台。系统包含 20 块 APS4322 XPlanar 瓷砖，每块尺寸为 240 mm $\times$ 240mm，以及一个尺寸为 113 mm 的 APM4220 移动器 $\times$ 113 mm。硬件如图 12所示。真实系统演示视频可在以下网址观看：https://www.youtube.com/watch?v=zhZ2Ko5VxUk

轨迹以 100赫兹 的频率采样，`C++` 脚本通过 ADS 通信将这些采样写入可编程逻辑控制器中的 FIFO 缓冲区。可编程逻辑控制器将采样传递给 Beckhoff 提供的 `ExternalSetPointGenerator()`功能。这些采样包含位置、速度和加速度设定值。内部底层跟踪控制器负责跟踪参考轨迹。

图 13展示了使用所提方法（PMP）计算得到的移动器实际行驶轨迹记录。对于该轨迹，移动器在3mm高度悬停，$a_{\max}$ 被设为6 m/s^2，且 $v_{\max}$ 被设为2 m/s（根据技术数据，这是最大速度）。速度限制在 x维度达到，且 $|v_y|$ 的最大值为1.8m/s。对于不同的参数值也得到了类似结果，尽管高得多的加速度限制会推高跟踪控制器的性能，导致移动器更加偏离所提供的设定值。这可能导致移动器在突然停止时进入错误状态。我们认为障碍物的位置是已知的，并且如第2节所述，所提方法获得了包含这些障碍物的环境地图。跟踪误差，即移动器测量位置与期望位置设定值之间的距离，由轨迹的颜色表示。跟踪误差始终小于1.2 mm，并且在以更高速度运动时更大。该结果表明



(a) XPlanar 硬件　　(b) 移动器执行轨迹

图 12：实验验证所用硬件。利用安装在天花板上的投影仪，可在硬件上实时可视化障碍物、走廊和轨迹

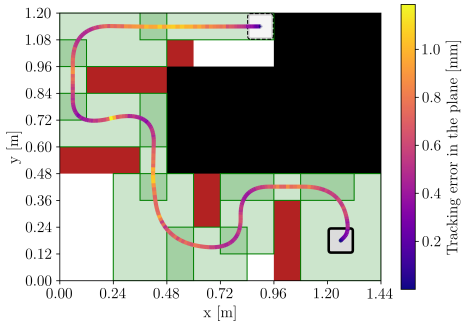


图 13：真实移动器跟踪采用所提方法计算的参考轨迹而执行的实测轨迹。红色方块表示障碍物。绿色矩形表示走廊序列，重叠区域以较深色调显示。轨迹颜色表示实测位置与给定位置设定点之间的距离。请注意，该误差以毫米表示。平面内最大跟踪误差为 1.19mm，平面内平均跟踪误差为 0.57mm

车辆能够准确跟踪由所提方法计算的轨迹，并展示了该硬件出色的跟踪能力。这也意味着可能无需以固定频率重新规划轨迹，而可以依靠内部参考轨迹跟踪控制器到达目的地。

6 结论与未来工作

在本工作中，我们提出了一种新方法，用于为复杂但结构化环境中的完整约束车辆计算近似时间最优轨迹。首先，我们构建矩形走廊序列以对自由空间进行建模。基于这些走廊的几何形状，我们使用启发式方法确定强制车辆经过的航路点位置，以及这些航路点处加速度的符号。利用这些信息，我们选择参数化运动基元，然后求解优化问题，以优化剩余自由度。

通过开展大量数值实验，我们表明，所得到的优化问题的求解速度显著快于最先进方法，同时在走廊序列内并未显著牺牲最优性。然而，在某些情况下，走廊序列是次优的，这一点通过对VP-STO进行基准测试而变得显而易见。我们计划通过探索属于不同同伦类的路径，并并行求解不同轨迹来解决这一问题。尽管如此，基于采样的方法只能渐近地实现全局最优性；尤其是在狭窄通道或大型障碍物的情况下，VP-STO可能收敛到一个（可能较差的）局部最优解。这也受到所考虑的途经点数量的影响，而该数量是一个难以调节的调参参数。在所提方法中，基元的数量由所构建走廊的数量自动确定。未来工作计划改进走廊构建，有可能生成多个可并行求解的序列。

我们还使用 Beckhoff Automation 的 XPlanar mover system，在真实的工业硬件上验证了我们的方法。这些结果表明，使用低级控制器可以高精度地跟踪所提出的轨迹。

合乎逻辑的下一步是在多智能体场景中使用所提出的规划器，这将需要在线重新规划。例如，可以让车辆认领环境中的区域，并规划穿越这些区域的路径，从而契合一个与deVos, Torta,Bruyninckx,Martinez,

以及 van de Molengraft (2024年）。需要注意的是，由于该规划器计算的是时间最优轨迹，框架应明确考虑递归可行性，以确保已经在行驶的车辆不会被要求突然躲避新的障碍物。

所提方法可以相对容易地扩展到包括旋转在内的多维情形，因为启发式方法以及由此产生的优化问题在每个维度上大体是解耦的。此外，可以修改运动基元的形状，以允许加加速度受限的运动。这些扩展可使所提方法适用于OMR。

最后，评估所提方法在更一般的走廊序列上的性能将会很有意义，例如略微倾斜、在某些应用中或许能更好地表示自由空间的走廊。在这种情况下，启发式方法的性能仍有待评估。

7 声明

伦理批准和参与同意。不适用。

出版同意。 所有作者均声明同意发表。

资助。 本工作由Flanders Make通过SBO项目ARENA（敏捷可靠导航）资助。

作者贡献。Louis Callens 撰写了论文的主要内容并完成了大部分研究。Bastiaan Vandewal 和 Ibrahim Ibrahim 协助开发算法、开展实验验证，并为稿件提供了重要反馈。Jan Swev- ers 和 Wilm Decr´e 提供了重要的研究指导，协助拟定稿件内容大纲，并审阅了稿件。

致谢。 作者感谢 Joris Gillis 博士在通过 CasADi 使用 FATROP 以及一般 C++ 实现方面提供的帮助。此外，作者感谢 Alejandro Astudillo Vigoya 博士对稿件的反馈。

参考文献

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M. (2018年). CasADi：用于非线性优化与最优控制的软件框架。*Mathematical ProgrammingComputation*, *11*, 1 - 36, https://doi.org/10.1007/s12532-018-0139-4

Bos,M.,Vandewal,B.,Decr´e, W., Swevers, J. (2023). 基于 MPC 的自主卡车-拖车机动运动规划。*IFAC-PapersOnLine*, *56*(2), 4877-4882,https://doi.org/10.1016/j.ifacol.2023.10.1258 (第 22 届 IFAC 世界大会)

Deits, R., & Tedrake, R. (2015). 通过半定规划计算无障碍空间中的大型凸区域。载于 H.L. Akin, N.M. Amato, V. Isler, & A.F. van der Stappen (编)，机器人学算法基础 *XI*：机器人学算法基础第十一届国际研讨会精选论文(第 109-124 页)。Springer International Publishing。https://doi.org/10.1007/978-3-319-16595-0 7

Demeester, E., Huentemann, A., Vanhooydonck, D., Vanacker, G., Brussel, H., Nuttin, M. (2008, 02). 用户适应型计划识别与用户适应型共享控制：一种用于半自主轮椅驾驶的贝叶斯方法。*Autonomous Robots*, *24*, 193-211, https://doi.org/10.1007/s10514-007-9064-5

De Santis,S., Astudillo,A.,Decr´e, W., Swevers, J. (2024年)。关于走廊环境中独轮机器人时间最优轨迹的解析计算。*2024*年欧洲控制会议（*ECC*）（第 3239-3246 页）。https://doi.org/10.23919/ECC64448.2024.10591028

de Vos, K., Torta, E., Bruyninckx, H., Martinez, C.L., van de Molengraft, R. (2024年). 面向可扩展多智能体的混合决策

导航：融合语义地图、离散协调和模型预测控制。 https://arxiv.org/abs/2410.12651

Dmitruk, A., & Kaganovich, A. (2011, 04). 带中间约束的最优控制问题的极大值原理。计算数学与建模，*22*，180- 215，https://doi.org/10.1007/s10598-011-9096-8

Feh´er,M.，Straka,O.，ˇSm´ıdl, V. (2017, jan). 双积分器系统的约束时间最优控制及其在模型预测控制中的应用。物理学杂志：会议系列，*783*(1), 012024, https://doi.org/10.1088/1742-6596/783/1/012024

Gammell, J.D., Srinivasa, S.S., Barfoot, T.D. (2014). Informed rrt*：一种基于采样的最优路径规划方法，通过直接采样可容许椭球启发式函数实现聚焦。*2014*年 *IEEE/RSJ*智能机器人与系统国际会议 (第 2997- 3004 页)。https://doi.org/10.1109/IROS.2014.6942976

Geraerts, R., & Overmars, M.H. (2007). 廊道地图方法：实时高质量路径规划。*2007*年*IEEE*机器人与自动化国际会议论文集(第1023–1028页)。https://doi.org/10.1109/ROBOT.2007.363119

Iwanaga, Y., & Fujimoto, Y. (2025). 通过高精度离散时间建模，为轮式移动系统规划杂乱环境中的最短时间轨迹。*IFAC*系统与控制期刊，*32*，100316，https://doi.org/https://doi.org/10.1016/j.ifacsc.2025.100316

Jankowski，J.，&Bruderm¨uller, L. (2024年)。*vp-sto*。https://github.com/JuJankowski/vp-sto。（访问日期：2025-07-22）

Jankowski, J., Bruderm¨uller,L.,Hawes,N., Calinon, S.（2023）。VP-STO：基于途经点的随机轨迹优化，用于反应式机器人行为。*2023 IEEE*国际机器人与自动化会议（*ICRA*）（第10125-10131 页）。https://doi.org/10.1109/ICRA48891.2023.10160214

Jansen, J.W., van Lierop, C.M.M., Lomonova, E.A., Vandenput, A.J.A.（2008）。采用移动磁体的磁悬浮平面执行器。*IEEE*工业应用汇刊，*44*（4），1108-1115，https://doi.org/10.1109/TIA.2008.926065

Karaman, S., & Frazzoli, E.（2011）。用于最优运动规划的基于采样的算法。机器人学研究国际期刊，*30*（7），846-894，https://doi.org/10.1177/0278364911406761

Kuffner, J.J., & LaValle, S.M.（2000）。RRT-Connect：一种用于单查询路径规划的高效方法。*2000*年*ICRA*千年会议论文集。*IEEE*机器人与自动化国际会议。专题会议论文集（目录号：*00CH37065*）（第2卷，第995-1001页）。https://doi.org/10.1109/ROBOT.2000.844730

Mercy, T., Hostens, E., Pipeleers, G.（2018年）。面向自动驾驶车辆的在线运动规划，适用于密集环境。*2018*年 *IEEE* 第*15*届先进运动控制国际研讨会（*AMC*）（第114-119页）。https://doi.org/10.1109/AMC.2019.8371072

Mercy, T., Van Loock, W., Pipeleers, G.（2016）。存在移动障碍物时的实时运动规划。*2016*年欧洲控制会议（*ECC*）（第1586-1591页）。https://doi.org/10.1109/ECC.2016.7810517

Nasir, J., Islam, F.,Malik, U.,Ayaz, Y., Hasan, O., Khan, M., Muhammad, M.S. (2013)。RRT*-Smart：一种实现 RRT* 快速收敛的方案。先进机器人系统国际期刊，*10*(7), 299,https://doi.org/10.5772/56718

Nithin Jarralde, J., Vibin, R., Govardhan, A., Raffik, R. (2023)。基于磁悬浮的工业物料搬运系统：未来展望综述。*2023* 年第二届电气、电子、通信、计算与自动化进展国际会议（*ICAECA*）（第 1–6 页)。https://doi.org/10.1109/ICAECA56562.2023.10199189

Noreen, I., Khan, A., Habib, Z.（2016年，11月）。基于 RRT* 方法的最优路径规划：综述与未来方向。《国际高级计算机科学与应用期刊》，*7*, 97–107, https://doi.org/10.14569/IJACSA.2016.071114

Overmars, M., Karamouzas, I., Geraerts, R. (2008). 使用走廊地图进行灵活路径规划。D. Halperin & K. Mehlhorn（编），算法——*ESA 2008*（第 1-12 页）。柏林，海德堡：Springer Berlin Heidelberg。https://doi.org/10.1007/978-3-540-87744-8 1Schaeffel, C., Katzchmann, M., Morh, H.-U., Gloess, R., Rudolf, C., Mock, C., Walenda, C. (2016). 六自由度平面磁悬浮系统——PiMag 6D。《机械工程学报》，*3*(1), 15-00111-15-00111, https://doi.org/10.1299/mej.15-00111Taheri, H., & Zhao, C.X. (2020). 全向移动机器人、机构与导航方法。*Mechanism and Machine Theory*，*153*，103958, https://doi.org/10.1016/j.mechmachtheory.2020.103958Vanroye, L.,Sathya,A.,De Schutter, J., Decr´e, W. (2023). FATROP：一种用于机器人轨迹优化与控制的快速受约束最优控制问题求解器。*2023 IEEE/RSJ* 智能机器人与系统国际会议（*IROS*）（第 10036-10043 页）。https://doi.org/10.1109/IROS55552.2023.10342336

W¨achter, A., & Biegler, L.T. (2006). 关于面向大规模非线性规划的内点滤波线搜索算法的实现。*Mathematical Programming*，*106*，25-57, https://doi.org/10.1007/s10107-004-0559-y

Webb, D.J., & van den Berg, J.（2013）。运动学-动力学 RRT*：针对具有线性动力学的机器人的渐近最优运动规划。*2013*年 *IEEE*机器人与自动化国际会议（第5054–5061页）。https://doi.org/10.1109/ICRA.2013.6631299

Xie, C., van den Berg, J., Patil, S., Abbeel, P.（2015年）。利用两点边值问题求解器实现运动学-动力学系统的渐近最优运动规划。*2015*年 *IEEE*机器人与自动化国际会议（*ICRA*）（第 4187–4194页）。https://doi.org/10.1109/ICRA.2015.7139776