

城市道路自动驾驶车辆运动规划和控制技术综述 *

原作: Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli

译作: 罗辉武 †

版本: 0.02

日期: 2019 年 12 月 16 日

摘要

自动驾驶汽车技术属于一类趋于成熟的技术 (a maturing technology), 通过提高汽车运输的安全性、可达性、效率性和便利性来重塑汽车的机动性。自动驾驶汽车必须执行对安全至关重要的任务包括: 通过与其他车辆和行人共享的动态环境信息来规划运动, 以及通过反馈控制执行稳健的驾驶。本文的目的是调查规划和控制算法的现状, 特别是与城市环境有关的规划和控制算法。对所选型的相关技术进行了简单回顾, 并对其有效性进行了讨论。注意: 所调研的方法所采用的车辆运动模型、对环境结构的假设以及计算资源的要求方面存在差异。本次调查中提出的并行比较有助于深入了解所调研方法的优点和局限性, 从而有助于在系统的设计阶段选择合适的方法进行部署。

目录

1 引言	2	4 运动规划	11
2 无人驾驶汽车决策层研究综述	3	4.1 路径规划	12
2.1 轨迹规划	4	4.2 轨迹规划	13
2.2 行为决策	4	4.3 变分方法	16
2.3 运动规划	6	4.4 图搜索方法	17
2.4 车辆控制	6	4.4.1 车道线图	18
3 规划与控制模型	6	4.4.2 几何方法	19
3.1 单轨运动学模型	6	4.4.3 基于采样的方法	19
3.2 惯性效应	8	4.4.4 图搜索策略	22
		4.5 增量搜索方法	23
		4.6 实际部署	26

*本文以下文献的中文翻译, 英文原版为: <https://arxiv.org/pdf/1604.07446.pdf>

B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," in *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, March 2016.

†Email: huiwu.luo@aliyun.com

注: 由于时间仓促和非科班出身, 对内容不熟悉, 难免存在错误, 请在在 GitHub Issues 中提交您的宝贵修改建议。

5 车辆控制	26		
5.1 基于运动学模型的路径稳定	28	5.3.1 具有运动学模型的无约束 MPC	36
5.1.1 纯追踪	28	5.3.2 路径跟踪器	36
5.1.2 后轮位置反馈	29	5.3.3 轨迹跟踪控制器	36
5.1.3 前轮位置反馈	30	5.4 线性变参控制器	37
5.2 基于运动学模型的轨迹控制	33		
5.2.1 设计 Control-Lyapunov 函数	33	6 结论	37
5.2.2 输出反馈线性化	33		
5.3 可预测的控制方法	34	【参考文献】	38

1 引言

过去的三十年里，学术界和工业界都在稳步加大研究力度，致力于发展无人驾驶汽车相关技术。传感技术和计算技术的最新进展，加上汽车运输和社会效益的变革的影响，进一步推动了它的发展：2014 年，与交通有关的死亡人数为 32675 人次，受伤人数为 230 万人次，上报的碰撞事故为 610 万人次 [1]。其中，保守估计 94% 归因于司机的错误驾驶行为，31% 涉及法律意义上的醉酒驾驶，10% 源于司机驾驶中的分心行为 [2]。在显著减少因驾驶员的驾驶错误和人为过失行为造成车辆发生碰撞方面，无人驾驶汽车有很大的应用潜力。此外，无人驾驶还可以为身体残疾或视觉残疾而无法驾驶车辆的人群提供个性化的机动方案。最后，对于美国 86% 的上班族来说，平均每天需要花费 25 分钟（单程）的驾驶时间 [3]，自动驾驶汽车有助于更有效地利用交通时间，同时消除因驾驶压力带来的可测量的不良影响 [4]。

考虑到这项新技术的潜在影响，自动驾驶汽车有着悠久的历史并不奇怪。这种想法早在 20 世纪 20 年代就已经出现，但直到 20 世纪 80 年代，无人驾驶汽车才似乎成为一种真正的可能。由 Ernst Dickmanns（例如，[5]）在 20 世纪 80 年代领导的开创性工作为自主汽车的发展铺平了道路。当时，普罗米修斯项目（PROMETHEUS project）是一项大型研究项目，该资助项目的目标是开发出一种自主汽车。1994 年，该项工作的一个显著成果是，VaMP 无人驾驶汽车行驶了 1600 公里，其中 95% 路段属于自动驾驶。与此同时，CMU NAVLAB 也在该地区取得了成果；并于 1995 年取得进一步的发展：它设计的汽车在美国境内行驶了 5000 公里，其中 98% 属于自动驾驶 [7]。

无人驾驶汽车技术的下一个重大里程碑是于 2004 年举办的第一届 DARPA 挑战赛。此次比赛的目标是无人驾驶汽车以最快速度驶过 150 英里的越野路线。与以往的比赛相比，此次比赛属于一个重大挑战，因为比赛期间不能人为干预汽车。尽管之前的研究表明汽车几乎可以实现自主行驶，但在关键时刻消除人为因素属于重大挑战。参加比赛的 15 辆车没有一辆完成比赛。2005 年，举行了一个类似的比赛：这次比赛 23 支队伍中有 5 支队伍到达终点线 [8]。2007 年末，DARPA 举办了城市挑战赛，要求车辆在模拟城市环境中自主驾驶。六支车队完成了比赛，证明了完全自主的城市驾驶行为是可以做到的 [9]。

自举办 DARPA 挑战赛之后，陆续举办了许多类似的活动、同时进行了很多自主车辆的系统测试。重要的例子包括 2009 年至 2013 年的智能汽车未来挑战赛 [10]、2010 年的韩国现代汽车

(Hyundai) 自主挑战赛 [11]、2010 年的 VisLab 洲际自主挑战赛 [12]、2013 年的公共道路城市无人驾驶汽车测试活动 [13] 和伯莎-奔驰 (Bertha-Benz) 历史路线的自动驾驶活动 [14]。与此同时，无论在学术界还是在工业界，都在加速进研究状态。谷歌 (Google) 的自动驾驶汽车 [15] 和特斯拉 (Tesla) 的自动驾驶系统 [16] 是两个典型的商业案例，受到媒体的广泛关注。

汽车的自动化程度可划分为从完全人工操作到完全自主。SAE J3016 分级标准 [17] 引入了一个从 0 到 5 的汽车自动化分级标准。在该标准中，0 级表示所有驾驶任务由人类完成。L1 级包括基本的辅助驾驶系统，如自适应巡航控制、防锁死刹车系统 (anti-lock braking systems) 和电子稳定控制系统 [18]。L2 级包含有先进的辅助措施，如为减少危险的纵向操控/横向操控 [19] 或紧急制动 [20], [21]，此级通常采用基于集合的控制理论和方法来计算无碰撞（即安全）状态下的“最坏情况”的集合 [22]-[24]。在 L3 级，系统监控周围环境，在特定条件下可以完全自主驾驶，但是如果驾驶任务不在自主系统的操作范围时，仍需要人工帮助控制车辆。具有 L4 级的车辆能够在特定条件下完全自主驾驶，并且在操作员未能在要求干预时能够采取正确的控制措施，对车辆进行安全控制。最后，L5 级系统在所有驾驶模式下都是完全自主的。

车载计算技术和无线通信技术使汽车能够与其他汽车或道路的基础设施交换信息，从而产生紧密相关的物联网智能汽车研究领域 [25]。该研究领域在透过个别车辆之间的信息分享与协调，提升道路运输的安全性与运输性能。例如，车联技术有可能提高交叉路口的吞吐量 [26]，或者防止交通事故的形成 [27]。

为缩小本次综述的范围，我们将重点放在自动驾驶汽车的决策、运动规划和控制方面，特别是 L3 级及以上的系统。基于相同的原因，省略了自动驾驶的广义领域，读者可参考关于这一主题的综合调研报告和最近的成果以加深了解 [28]-[31]。

现代自主驾驶系统的决策通常可划分为路径规划、行为决策、局部运动规划和基于反馈的控制。然而，不同文献对这些层次的划分也是比较模糊的。本文概述了解决自动驾驶汽车核心问题的方法，尤其是局部运动规划方法和控制方法。

本文的其余部分结构如下：在第二节中对决策过程的层次结构及其设计方法进行概述。基于运动规划和反馈控制为本综述的主要目的，第三节回顾了用于城市道路的汽车运动模型。第四节综述了很多关于运动规划的文献，并讨论了其在自动驾驶汽车中的应用。同样，第五节讨论了无人驾驶汽车的路径稳定性问题和轨迹稳定性问题，以及具体的反馈控制方法。最后，第六节总结了当前的技术研究现状、以及未来研究中可能的研究方向。

2 无人驾驶汽车决策层研究综述

在该节中，我们描述了一个典型的自动驾驶汽车的决策架构，并对每个组件的职能进行了点评。无人驾驶汽车本质上是一种自主决策系统，它处理来自车载传感器（如雷达、激光雷达、摄像机、GPS/INS 装置和里程计）的观测数据。基于这些观察结果，连同关于道路网络、道路规则、车辆动力学和传感器模型的先验知识，用于自动选择控制车辆运动的控制变量的值。智能车辆研究旨在尽可能多地实现驾驶任务的自动化。解决这一问题的常用方法是对感知和决策任务进行规划，并组织成层次结构。感知系统利用先验信息和收集到的观测数据，对车辆及其周围环境的状态进行估计；然后由决策系统利用这些估计对车辆进行控制，从而实现驾驶车辆到

达目的地的目标。

典型的自动驾驶汽车的决策系统可分为四个部分（见图 1）：在最高层次，通过道路网络规划出一条行驶路线。接下来是行为层，它主要决定一个局部的驾驶任务，该任务使汽车朝目的地前进，同时遵守交通规则。然后，运动规划模块通过感知的环境信息选择一条连续路径完成局部导航任务。其次，在执行计划路线的运动过程中，控制系统对该过程产生的误差反复进行修正。在本节的其余部分中，我们将更详细地讨论每个部分的功能。

2.1 路线规划 (Route Planning)

在决策的最高层，车辆的决策系统必须选择从当前位置到请求的目的地的道路网络的最佳路线。此时通过将道路网络表示为一个有向图，其边的权值与穿过该段路径的代价相对应，这样的路线可以归结为在道路网络上找到最小通行代价的连通路问题。然而，表示道路网络的图可能包含数百万条边，这样会使得应用经典的最短路径算法（如 Dijkstra[32] 或 A*[33]）变得不切实际。对于如何有效规划交通网络中的问题，已经引起了交通科学界的极大兴趣，从而发明了一系列算法，通过预处理步骤后，在毫秒时间内返回大规模道路网络的最优路径 [34], [35]。关于对人类驾驶车辆和自动驾驶车辆进行路线规划的实用算法的性能比较的全面综述文献，请参见 [36]。

2.2 行为决策 (Behavioral Decision Making)

制定计划的路线后，自动驾驶车辆必须能够按照道路的行驶习惯和规则在选定的路线上行驶，并与其他交通主体进行交互。给定一系列所选路线的路段，行为层负责根据其他交通主体的感知信息、道路状况、和基础交通设施发出的信号，在任何时间点选择适当的驾驶行为。例如，当车辆在交叉路口中到达停车线位置时，行为层将命令车辆停车；同时观察交叉路口其他汽车、自行车和行人的行为，当轮到车辆行驶时让车辆继续前进。

驾驶规则规定了特定驾驶环境下的可能操作。由于驾驶状态的上下文行为和每种上下文行为中可能产生的行为都可以建模为有限集，因此自动化此决策的自然方法是将每个行为建模为有限状态机中的状态，其转换由感知获得的驾驶上下文进行控制，例如相对于计划路线、附近车辆的驾驶位置。事实上，在 DARPA 城市挑战赛中，大多数团队将有限状态机与基于特定驾驶场景的不同启发式策略进行结合，用作行为控制机制 [9]。

然而，在真实环境中驾驶时，特别是在城市环境时，主要的特点是其他车辆主体的意图存在相当的不确定性。智能驾驶领域也对自身外的其他汽车、自行车及行人在意图预测和可能轨迹估计等问题进行了研究。在提出的解决方案中，有基于机器学习的技术，例如高斯混合模型 [37]、高斯过程回归 [38]；据报道在谷歌的自动驾驶系统中有用于意图预测的学习技术 [39]、以及根据传感器的测量直接估计意图的基于模型的方法 [40][41]。

使用概率规划（如马尔可夫决策过程 (MDPs) 及其推广）进行决策时，通常会在行为层考虑其他交通主体行为中的这种不确定性。例如，[42] 在 MDP 框架中定义了行为决策问题。几个工作 [43]-[46] 对未观察到的驾驶场景和行人意图使用一个部分可观测的马尔可夫决策过程 (POMDP) 框架来进行显式建模，并提出在特定条件下的近似解策略。

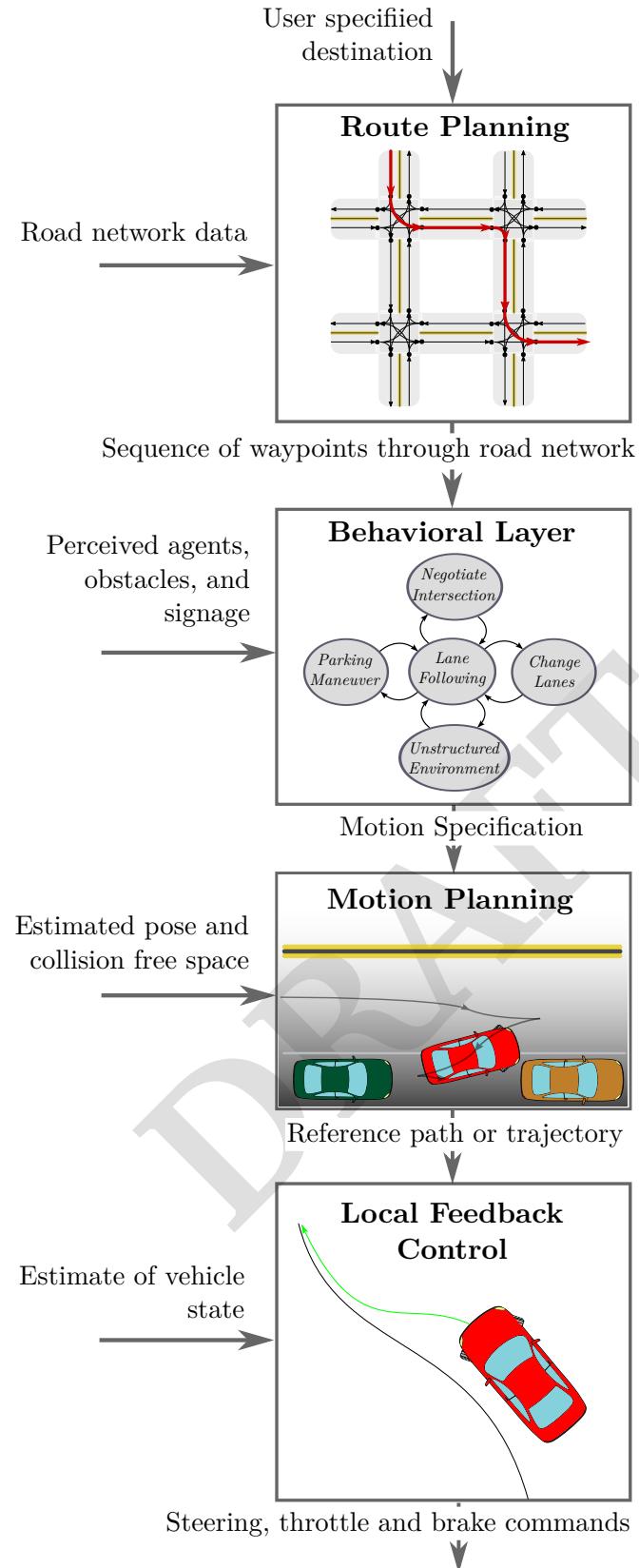


图 1: 决策的层次结构图。将终点传递给路线规划器，该规划器通过道路网络生成路线。行为层对环境进行推理，并生成一个运动规范，以便沿着选定的路线前进。然后，运动规划器求解完成该规范的可行运动。反馈控制调整驱动变量，以纠正车辆执行参考路径时的误差。

2.3 运动规划 (Motion Planning)

当行为层决定在车辆的当前上下文环境中执行具体的驾驶行为（例如，在车道上巡航、改变车道或右转）时，所选行为必须转换为可由低层级的反馈控制器跟踪的路径或轨迹。所产生的路径或轨迹必须对车辆动态可执行，对乘客舒适，并避免与车载传感器检测到的障碍物发生碰撞。而寻找这样一条可行路径或轨迹的任务是运动规划系统的职责。

自主车辆的运动规划对应于解决机器人学文献中讨论的标准运动规划问题。运动规划问题的精确解在大多数情况下是难以计算的。因此，通常在实际中使用数值逼近方法。当前最流行的数值方法有：变分法，它将问题转换为函数空间中的非线性优化问题；图搜索法，它将车辆的状态空间用图离散化技术 (Graphical Discretization)，并使用基于图优化的搜索法找到最短路径；以及基于增量树的方法，它从车辆的初始状态出发，构造一棵可到达的状态树，然后选择该树的最佳分支作为解。我们将在第四节详细地讨论与自动驾驶相关的运动规划方法。

2.4 车辆控制 (Vehicle Control)

为执行运动规划系统输出的参考路径或轨迹，使用反馈控制器选择恰当的输入传递给运动规划执行器并纠正跟踪误差。运动执行过程中产生的跟踪误差部分是由于车辆模型的不精确造成的。因此，要重视闭环系统的鲁棒性和稳定性。

为执行运动规划系统给出的参考路径，人们提出了许多有效的反馈控制器。相关的技术调研情况将在第五节进行详细介绍。

3 规划与控制模型

在本节中，我们将调研最常用的车辆动力学模型。这种模型广泛应用于控制和运动规划算法中，在相关操作条件下通过响应制动来近似车辆的行为。真实的控制模型可以准确反映车辆的响应，但由此增加的实现细节可能会使规划和控制问题复杂化。所以在模型准确性和决策难度之间会采取一个折衷方案。本节概述了通用汽车建模概念，并概述了用于运动规划和控制的车辆模型。

车辆建模从车辆构型 (Vehicle Configuration) 概念开始，车辆构型表示车辆在真实世界上的姿态或位置。例如，车辆构型可以用汽车上某个点的平面坐标以及汽车的航向表示。该坐标系即为汽车构形空间 (Configuration Space) 的坐标系。该坐标系描述平面刚体运动 (由二维的特殊欧几里德群 $SE(2)$ 表示)，且是常用的构型空间 [47]-[49]。之后，必须对车辆的运动进行规划和调节以完成驾驶任务，所以须为所选模型引入约束。

3.1 单轨运动学模型 (The Kinematic Single-Track Model)

实际应用的基本模型中，汽车由两个车轮组成，通过刚性连杆连接，并将其限制为只能在平面内移动 [48]-[52]。假设车轮在与地面的接触点不打滑，但可以围绕其旋转轴自由旋转。前轮有一个附加的自由度，允许前轮绕一个垂直于运动平面的轴旋转。此条件是为了对转向进行建模。这两个建模特征反映大多数乘客的经验，即汽车不能在不同时向前移动的情况下能在横向

方向进行移动。正式地说，对机动性的限制称为非完整约束 (nonholonomic constraint)[47], [53]。非完整约束表示汽车运动的微分约束。此表达式随坐标系的选择而变化。该模型的变种被称为类车机器人、自行车模型、运动学模型或单轨模型。

以下对常用构型坐标系的几种微分约束进行推导。参考图 2，向量 p_r 和 p_f 表示后轮和前轮在静止或惯性坐标系中的位置，该坐标系基向量为 $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ 。航向 θ 为一个角度，描述了车辆面对的方向，定义为向量 \hat{e}_x 和 $p_f - p_r$ 之间的夹角。

以下导出由角度 θ 、于文献 [54] 提到的点 p_r 的运动、以及于文献 [55] 提到的点 p_f 所组成的坐标系的微分约束。

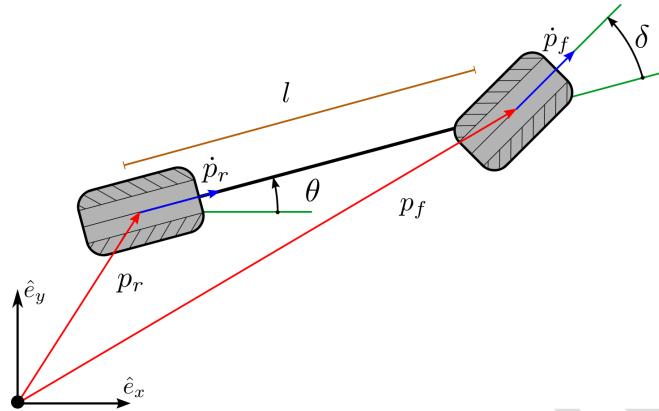


图 2：单轨运动学模型。 p_r 和 p_f 分别为车轮与地面的接触点； θ 为车辆朝向； p_r 和 p_f 的时间导数受非完整性约束（蓝色箭头所指向的方向）； δ 是前轮的转向角。

点 p_r 和点 p_f 的运动必须与车轮方向共线，以满足防打滑假设。后轮上的约束可用方程式表示为：

$$(\dot{p}_r \cdot \hat{e}_y) \cos(\theta) - (\dot{p}_r \cdot \hat{e}_x) \sin(\theta) = 0 \quad (1)$$

同样前轮的约束为：

$$(\dot{p}_f \cdot \hat{e}_y) \cos(\theta + \delta) - (\dot{p}_f \cdot \hat{e}_x) \sin(\theta + \delta) = 0 \quad (2)$$

通常将该表达式依据每个点沿基向量的分量进行重写。后轮沿 \hat{e}_x 方向的运动量为 $x_r := p_r \cdot \hat{e}_x$ 。类似地，对于 \hat{e}_y 方向， $y_r := p_r \cdot \hat{e}_y$ 。前进速度为： $v_r := \dot{p}_r \cdot (p_f - p_r) / \|p_f - p_r\|$ ，即为 \dot{p}_r 的大小并加上正确的符号以表示前进还是后退。对于标量 x_r ， y_r 和 θ ，微分约束为：

$$\begin{aligned} \dot{x}_r &= v_r \cos(\theta) \\ \dot{y}_r &= v_r \sin(\theta) \\ \dot{\theta} &= \frac{v_r}{l} \tan(\delta) \end{aligned} \quad (3)$$

或者，亦可以根据运动 p_f 重写微分约束：

$$\begin{aligned} \dot{x}_f &= v_f \cos(\theta + \delta) \\ \dot{y}_f &= v_f \sin(\theta + \delta) \\ \dot{\theta} &= \frac{v_f}{l} \sin(\delta) \end{aligned} \quad (4)$$

其中采用了前轮的前进速度 v_f 。前轮速度 v_f 与后轮速度的关联公式为：

$$\frac{v_r}{v_f} = \cos(\delta) \quad (5)$$

该模型的规划和控制为车轮的机械极限 $\delta \in [\delta_{\min}, \delta_{\max}]$ 内选择转向角度 δ , 并在可接受的范围内选择前进速度 v_r : $v_r \in [v_{\min}, v_{\max}]$ 。

有时亦采用简化式, 例如 [56] 选择的是航向率 ω 而不是转向角 δ :

$$\delta = \arctan\left(\frac{l\omega}{v_r}\right) \quad (6)$$

这样可将航向动力学简化为:

$$\dot{\theta} = \omega, \quad \omega \in \left[\frac{v_r}{l} \tan(\delta_{\min}), \frac{v_r}{l} \tan(\delta_{\max}) \right] \quad (7)$$

此种情况下的模型称为独轮车模型 (unicycle model), 因为可以通过仅考虑单个车轮的运动来进行推导。

该模型的一个重要变化模型是当固定 v_r 的情况下。此情况下的车辆模型亦称为 Dubins 车辆模型: Lester Dubins 推导出一个点与规定切线之间车辆在最短时间内运动路线 [57]。另一个著名的变型模型是当 v_r 采用一个固定的前进速度和后退速度时称为 Reeds-Shepp 车辆模型, 此时已知车辆的最小路径长度 [58]。这两个模型已被证实对运动规划具有重要的意义, 将在第四节进一步讨论。

运动学模型适用于低速 (如停车场和城市驾驶) 条件下的路径规划, 与防滑假设对机动性的限制相比, 惯性效应很小。该模型的一个主要缺点是它允许瞬时转向角变化, 如果运动规划模块产出具有这种瞬时变化速度的方案, 则可能会出现问题。

转向角的连续性可通过增强式(4)来实现, 其中转向角整合了文献 [49] 提到中的指令速度。此时方程式(4)变成:

$$\begin{aligned} \dot{x}_f &= v_f \cos(\theta + \delta) \\ \dot{y}_f &= v_f \sin(\theta + \delta) \\ \dot{\theta} &= \frac{v_f}{l} \sin(\delta) \\ \dot{\delta} &= v_\delta \end{aligned} \quad (8)$$

除了转向角的限制外, 现在还可以限制转向率 (Steering Rate): 同样, 问题也可能出现在车速 v_r 上, 可以用同样的思路解决。这种模型的缺点是当输入维数增加时, 会使运动规划和控制问题变得很复杂。

关于坐标系统的选, 并不局限于将其中一个车轮的位置作为坐标。对于使用经典力学原理导出的模型, 可以方便地使用质心位置作为原点坐标, 例如 [59] 和 [60]; 或者, 采用振动中心作为原点坐标, 例如 [61]、[62]。

3.2 惯性效应 (Inertial Effects)

当加速度足够大时, 轮胎和地面之间的防滑假设将失效。此种情况下, 精确的车辆模型应该是满足基本动量原理的刚体。也就是说, 加速度与在地面上给轮胎施加的力成正比。假如 p_c 为车辆的重心, 并将它作为构型空间的坐标原点 (参见图 3), 此时车辆的运动为:

$$\begin{aligned} m\ddot{p}_c &= F_f + F_r \\ I_{zz}\ddot{\theta} &= (p_c - p_f) \times F_f + (p_c - p_r) \times F_r \end{aligned} \quad (9)$$

式中, F_r 和 F_f 是地面通过地面-轮胎相互作用施加到车辆上的力, m 是车辆的总质量, I_{zz} 是围

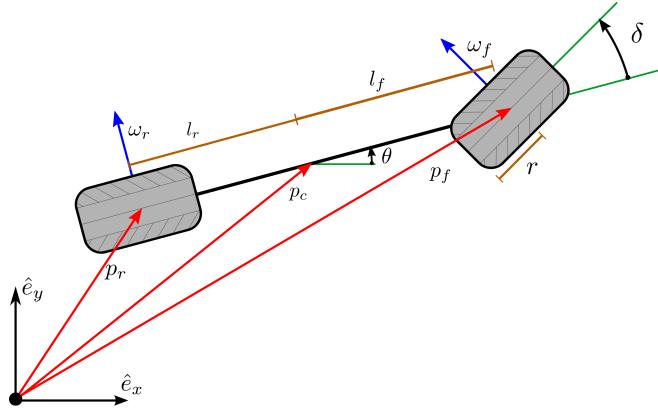


图 3: 无防滑假设的单轨运动学模型图解。 $\omega\{r,f\}$ 是车轮相对于车辆的相对角速度。

绕质量中心沿 \hat{e}_z 方向的极惯性矩 (polar moment)。在下面的推导中，我们默认忽略 p_c 在 \hat{e}_z 方向上的运动，并且假设道路是水平的，悬架 (suspension) 是刚性物体，而且车一直在道路上。

F_r 和 F_f 的表达式依据建模时的假设而变化 [18]、[59]、[60]、[62]，但在任何情况下，该表达式的推导都可能很繁琐。因此，式(10)-(15)提供详细的推导过程作为参考。

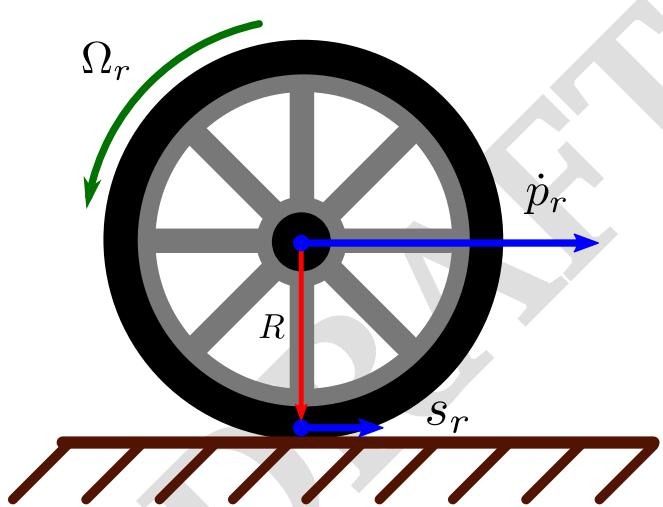


图 4: 后轮运动学的二维图解，展示了车轮打滑 s_r 与后轮速度 \dot{p}_r 和角速度 ω_r 的关系。通常， s_r 和 \dot{p}_r 并不共线，并且可能具有垂直于所示平面的非零分量。

地面和轮胎之间的力大小在建模上取决于轮胎在地面上的打滑率。虽然以质心作为构型的

坐标，但还需要每个车轮相对于地面的速度来确定此相对速度。这三点之间的运动学关系是：

$$\begin{aligned}
 p_r &= p_c + \begin{pmatrix} -l_r \cos(\theta) \\ -l_r \sin(\theta) \\ 0 \end{pmatrix} \\
 \dot{p}_r &= \dot{p}_c + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix} \times \begin{pmatrix} -l_r \cos(\theta) \\ -l_r \sin(\theta) \\ 0 \end{pmatrix} \\
 p_f &= p_c + \begin{pmatrix} l_f \cos(\theta) \\ l_f \sin(\theta) \\ 0 \end{pmatrix} \\
 \dot{p}_f &= \dot{p}_c + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix} \times \begin{pmatrix} l_f \cos(\theta) \\ l_f \sin(\theta) \\ 0 \end{pmatrix}
 \end{aligned} \tag{10}$$

这些运动学关系用于确定与地面接触的每个轮胎上的点的速度： s_r 和 s_f 。这些点的速度称为轮胎滑移速度 (tire slip velocity.)。一般来说， s_r 和 s_f 区别于 \dot{p}_r 和 \dot{p}_f 的地方在于车轮的角速度。它们的运动学关系为：

$$\begin{aligned}
 s_r &= \dot{p}_r + \omega_r \times R \\
 s_f &= \dot{p}_f + \omega_f \times R
 \end{aligned} \tag{11}$$

车轮的角速度由下式给出：

$$\omega_r = \begin{pmatrix} \Omega_r \sin(\theta) \\ -\Omega_r \cos(\theta) \\ 0 \end{pmatrix}, \quad \omega_f = \begin{pmatrix} \Omega_f \sin(\theta + \delta) \\ -\Omega_f \cos(\theta + \delta) \\ 0 \end{pmatrix} \tag{12}$$

并且 $R = (0, 0, -r)^T$ 。车轮半径是标量 r ， $\Omega_{r,f}$ 为每个车轮相对于汽车的角速度。如图 4 所示。

在静态条件下，或者当质心的高度可以近似为 $p_c \cdot \hat{e}_z \approx 0$ 时，可以从静力矩平衡 (static force-torque balance) 计算出与地面垂直的力分量 $F_{\{r,f\}} \cdot \hat{e}_z$:

$$F_f \cdot \hat{e}_z = \frac{l_r mg}{l_f + l_r}, \quad F_r \cdot \hat{e}_z = \frac{l_f mg}{l_f + l_r} \tag{13}$$

然后使用法线方向的力计算每个轮胎上的牵引力，以及轮胎本身的打滑和摩擦系数。后轮胎上的牵引力按分量分别由下式给出：

$$\begin{aligned}
 F_r \cdot \hat{e}_x &= -\frac{(F_r \cdot \hat{e}_z) \mu \left(\frac{\|s_r\|}{\Omega_r r} \right) s_r}{\|s_r\|} \cdot \hat{e}_x \\
 F_r \cdot \hat{e}_y &= -\frac{(F_r \cdot \hat{e}_z) \mu \left(\frac{\|s_r\|}{\Omega_r r} \right) s_r}{\|s_r\|} \cdot \hat{e}_y
 \end{aligned} \tag{14}$$

通过替换下标 r 为 f 可得到前轮的表达式。上述公式对牵引力的建模为：方向与打滑方向垂直，且其大小与法向力成比例，与滑移率 (由后轮通过 $\Omega_r r$ 标准化的滑移量，前轮由 $\Omega_f r$ 标准化) 具有非线性关系。结合(10)-(15)即可得到由控制变量、广义坐标系统、以及速度所组成的施加于每个车轮上的净力表达式。方程式(14)以及以下公式中的 μ 表达式：

$$\mu \left(\frac{\|s_r\|}{\Omega_r r} \right) = D \sin \left(C \arctan \left(B \frac{\|s_r\|}{\Omega_r r} \right) \right) \tag{15}$$

是轮胎与地面相互作用的常用模型。式(15)是 Pacejka[63] 得出的著名模型的简化版。

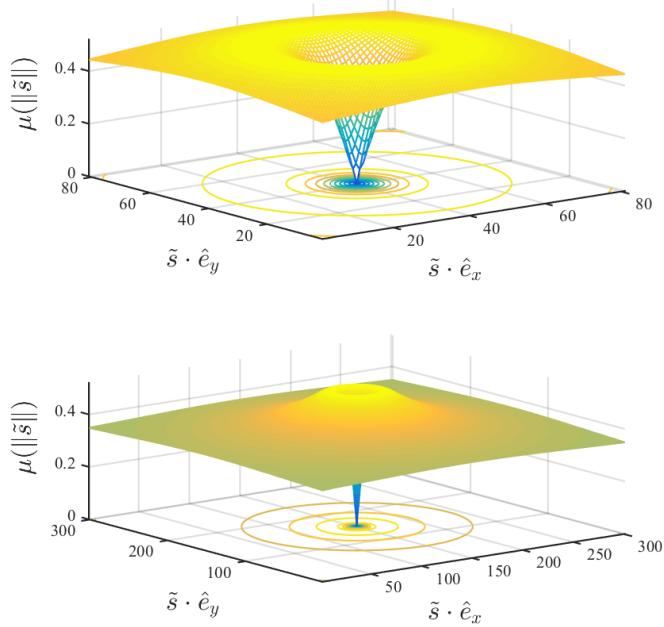


图 5: 每个轮胎的打滑到牵引力的放大效果图 (顶部), 以及强调定义摩擦圈峰值的缩小效果视图 (底部); 参见式(15)。

式(14)的旋转对称性与(15)中的峰值可得到轮胎在任意方向上施加的最大的力的数值。该峰值为图 5 所示的摩擦圈 (friction circle)。

本节讨论的模型经常出现在无人驾驶汽车运动规划和控制的文献中。它们适用于本次综述中讨论的运动规划和控制任务。然而, 较低级别的控制任务, 如电子稳定控制 (electronic stability control) 和主动悬架系统 (active suspension systems), 通常使用更复杂的底盘、转向和传动系统模型。

4 运动规划

运动规划层为车辆的当前构型到决策层行为层要求的目标构型计算一条安全、舒适和动态可行的轨迹。根据上下文的不同, 目标构型可能不同。例如, 目标位置可以是当前车道的中心点 (行驶方向前方数米处)、下一个交叉口的停车线中心或下一个停车点。运动规划组件接收车辆周围静态和动态障碍物信息, 并生成满足车辆动态运动约束的无碰撞轨迹。通常, 运动规划器也会最小化一个给定的目标函数。除了运动时间外, 目标函数还会惩罚危险动作或引起乘客不舒适的动作。在典型的设置中, 运动规划器的输出传递到本地反馈控制层。反过来, 反馈控制器生成一个输入信号, 以调节车辆跟随指定的运动路径。

车辆的运动路径可以采取路径或轨迹的形式。在路径规划框架中, 求解的路径用函数表示为 $\sigma(\alpha) : [0, 1] \rightarrow X$, 其中 X 是车辆的构型空间。注意, 这样的解并没有规定应该如何跟随路径, 您可以为路径选择速度配置文件, 也可以将此任务委托给决策层的较低层。在轨迹规划框架内, 仅明确考虑了控制的执行时间。这种考虑可以对车辆动力学和动态障碍物进行直接建模。在这种情况下, 解的轨迹可表示为一个以时间为参数的函数 $\pi(t) : [0, T] \rightarrow X$, 其中 T 是规划时间。与路径不同, 轨迹规定了车辆的构型如何随时间发生演变。

在以下两个部分，我们对路径规划和轨迹规划问题提供了一个正式的定义，并为这两个定义回顾了主要的问题和相应的算法效果。

4.1 路径规划

路径规划是在车辆（或更一般的机器人）的构型空间 X 中找到一条路径 $\sigma(\alpha) : [0, 1] \rightarrow X$ ，该路径从初始位置开始，满足给定的全局约束和局部约束的情况下到达目标区域。根据是否考虑求解路径的质量，用可行性 (feasible) 和最优性 (optimal) 两个术语来描述该路径。可行路径规划是指在不关注解的质量的情况下，确定一条满足给定问题约束的路径；而最优路径是指在给定约束条件下，找到一条满足某些优化准则的路径。

最优路径规划问题可以形式化地表述如下：设 X 为车辆的构型空间， $\Sigma(X)$ 表示所有的连续函数 $[0, 1] \rightarrow X$ 的集合。车辆的初始构型为 $X_{init} \in X$ ，路径目标为区域 $X_{goal} \subseteq X$ 。所有允许的车辆构型的集合称为自由构型空间 (free configuration space)，并用 X_{free} 表示。通常，自由构型是指那些不会与障碍物发生碰撞的构型，但是自由构型集也可以表示路径上的其他完整性约束 (holonomic constraints)。路径上的差分约束由谓词 $D(\mathbf{x}, \mathbf{x}', \mathbf{x}'', \dots)$ 表示，它在某种程度上可用于平滑车辆经过的路径，例如路径曲率和/或曲率速率 (the rate of curvature) 的界限。例如，在 $X \subseteq \mathbb{R}^2$ 情况下，差分约束可以使用 Frenet-Serret 公式获得路径的最大曲率 κ ：

$$D(\mathbf{x}, \mathbf{x}', \mathbf{x}'', \dots) \Leftrightarrow \frac{\|\mathbf{x}' \times \mathbf{x}''\|}{\|\mathbf{x}'\|^3} \leq \kappa \quad (16)$$

此外，记 $J(\sigma) : \Sigma(X) \rightarrow \mathbb{R}$ 为代价函数，则优化版本的路径规划问题可概括描述为：

Problem IV.1 (最优路径规划问题) 给定一个 5-元组 $(X_{free}, \mathbf{x}_{init}, X_{goal}, D, J)$ ，找出 $\sigma^* =:$

$$\begin{aligned} & \arg \min_{\sigma \in \Sigma(X)} J(\sigma) \\ & \text{subj. to } \sigma(0) = \mathbf{x}_{init} \text{ and } \sigma(1) \in X_{goal} \\ & \quad \sigma(\alpha) \in X_{free} \\ & \quad D(\sigma(\alpha), \sigma'(\alpha), \sigma''(\alpha), \dots) \quad \forall \alpha \in [0, 1] \end{aligned} \quad (17)$$

过去的几十年中，可行的最优路径规划问题得到了广泛的研究。该问题的复杂性已得到很好的理解，并已经开发了许多实用算法。

复杂性：很多文献致力于研究运动规划问题的复杂性。以下对关于计算复杂性问题的一些主要结论进行简要概述。

根据 **Problem IV.1** 中的公式，已知解决受完整性约束和差分约束的最优路径问题是 PSPACE-hard[64] 的。这意味着它的困难程度至少和解决任何 NP 完全问题是相同的困难，因此，假设 $P \neq NP$ ，目前还不存在能在多项式时间内能够解决该问题的有效算例。研究重点转向近似方法，或者在通用运动规划问题的子集上的方法。

最初的研究主要集中在多边形/多面体环境下完整性车辆模型的可行（而非最优）路径的规划上。也就是说，假设障碍物为一个多边形/多面体，并且在结果路径上没有差分约束。1970 年，Reif[64] 发现完整飞行器的无障碍路径，其足迹可以描述为一个多面体，在二维和三维环境中都可以在多项式时间内找到。Canny[65] 证明了在用多项式表示的自由空间找到可行路径规划问题在 PSPACE 的范围内，使得解决无微分约束的可行路径规划决策问题是一个完备 PSPACE 问题。

对于最优规划公式，其目标是找到最短的无障碍路径。众所周知，在具有多边形障碍物的二

维环境中，完整性车辆 (holonomic vehicle) 的最短路径可以在多项式时间内完成 [66], [67]。更准确地说，最短路径可以在 $O(n^2)$ 的时间内进行计算，其中 n 是多边形障碍物的顶点数 [68]。可以通过构造和搜索所谓的可视图 (visibility graph) 来取得解 [69]。相反，Lazard, Reif 和 Wang[70] 证明了在多边形障碍物（即类车机器人 (car-like robot) 的路径）中寻找二维平面上的最短曲率的有界路径属于一个 NP-hard 问题，这表明在多边形障碍物中寻找类车机器人的最短路径问题不存在能在多项式时间内完成的算法。一个相关的结论是，在多边形环境中受曲率约束的路径的存在性可以在指数时间 (EXPTIME) 内 [71] 决定。

在无障碍环境中，一种能有效获得解的特殊情况是最短曲率有界路径 (the shortest curvature bounded path)。Dubins[57] 已经证明，以曲率 κ 为界、并且分别具有正切角 θ_1 、 θ_2 的两点 p_1 、 p_2 之间的最短路径是一段由至多三段线段组成的曲线，每个线段都是圆弧线段或直线。Reeds 和 Shepp[58] 将这种方法推广到了既能向前行驶也能向后退的移动汽车上。另一个显著的例子是 Agraval 等人的算法是一个计算复杂度为 $O(n^2 \log n)$ 的算法，用于寻找凸多边形内曲率有界的最短路径。类似地，Bois-sonnat 和 Lazard[73] 提出了一种多项式时间算法，用于在障碍物具有有界曲率边界的环境中精确寻找曲率有界的路径。

自动驾驶中的大多数问题，精确算法与实际计算复杂度不可兼得 [70]，须求助于更一般的数值求解方法。这些方法一般不寻找精确解，而是试图寻找一个满意解或者一系列可行解，并收敛到最优解。这些方法的实用程度和计算性能通常由它们适用的问题类别以及它们收敛到最优解的程度来进行保证。路径规划的数值方法大致可分为三大类：

变分方法 (Variational methods) 将路径表示为一个由有限维向量参数化的函数，并利用非线性连续优化技术对向量参数进行优化以寻求最优路径。这类方法因其快速收敛到局部最优解而特别具有吸引力；然而，除非提供合适的初始值，否则会很难找到全局最优解。有关变分方法的详细讨论，请参见第4.3节。

图搜索方法将车辆构型空间离散为图，其中顶点表示车辆构型的有限集合，边表示顶点之间的转换。通过在这样的图中搜索最小代价路径，就可以找到所需的路径。图搜索方法不容易陷入局部极小，但它们仅限于有限路径集上的优化，即那些可以由图中的原子运动原语构造的路径集。关于图搜索方法的详细讨论，请参见第4.4节。

增量搜索方法对构型空间进行采样，并增量地构建一个可达图（通常为一棵树），该图维护一组离散的可达构型和它们之间的可行转换。一旦图足够大，大到使得至少一个节点位于目标区域中，则通过跟踪从开始位置指向该节点的边来取得所需的路径。与基本的图搜索方法相比，基于采样的方法逐渐增大图的大小规模，直到在图中找到满意解为止。关于增量搜索方法的详细讨论，请参阅第4.5节。

显然，可以通过组合方式发挥每种方法的优点。例如，正如文献 [74] 和 [75] 中所报告的，可以使用一个粗略的图搜索来获得变分方法的初始猜测。表1对选取部分的路径规划方法的关键特性进行了比较。在本节的剩余部分，我们将详细讨论路径规划算法及其特性。

4.2 轨迹规划

动态环境中或具有动态约束的运动规划问题可以在轨迹规划框架中得到更合适的表述，其中问题的解为轨迹，即以时间为参数的函数 $\tau(t) : [0, T] \rightarrow \mathcal{X}$ ，它规定了车辆构型如何随时间而

	Model assumptions	Completeness	Optimality	Time Complexity	Anytime
Geometric Methods					
Visibility graph [33]	2-D polyg. conf. space, no diff. constraints	Yes	Yes ^a	$O(n^2)$ [68] ^b	No
Cyl. algebr. decomp. [76]	No diff. constraints	Yes	No	Exp. in dimension. [65]	No
Variational Methods					
Variational methods (Sec IV-C)	Lipschitz-continuous Jacobian	No	Locally optimal	$O(1/\epsilon)$ [77] ^{k,l}	Yes
Graph-search Methods					
Road lane graph + Dijkstra (Sec IV-D1)	Arbitrary	No ^c	No ^d	$O(n + m \log m)$ [78] ^{e,f}	No
Lattice/tree of motion prim. + Dijkstra (Sec IV-E)	Arbitrary	No ^c	No ^d	$O(n + m \log m)$ [78] ^{e,f}	No
PRM [79] ^g + Dijkstra	Exact steering procedure available	Probabilistically complete [80] [*]	Asymptotically optimal * [80]	$O(n^2)$ [80] ^{h,f,*}	No
PRM* [80], [81] + Dijkstra	Exact steering procedure available	Probabilistically complete [80]–[82] ^{*,†}	Asymptotically optimal [80]–[82] ^{*,†}	$O(n \log n)$ [80], [82] ^{h,f,*,†}	No
RRG [80] + Dijkstra	Exact steering procedure available	Probabilistically complete [80] [*]	Asymptotically optimal [80] [*]	$O(n \log n)$ [80] ^{h,f,*}	Yes
Incremental Search					
RRT [83]	Arbitrary	Probabilistically complete [83] ^{i,*}	Suboptimal [80] [*]	$O(n \log n)$ [80] ^{h,f,*}	Yes
RRT* [80]	Exact steering procedure available	Probabilistically complete [80], [82] ^{*,†}	Asymptotically optimal [80], [82] ^{*,†}	$O(n \log n)$ [80], [82] ^{h,f,*,†}	Yes
SST* [84]	Lipschitz-continuous dynamics	Probabilistically complete [84] [†]	Asymptotically optimal [84] [†]	N/A ^j	Yes

表 1: 路径规划方法的比较。图例: *a*: 最短路径问题; *b*: n 为定义障碍物所使用的点数量; *c*: 仅在给定图(Graph)诱导的路径集上完备; *d*: 仅由给定图诱导的路径集上最优; *e*: n 和 m 分别是图中的边数和顶点数; *f*: 假设进行复杂度为 $O(1)$ 的碰撞检查; *g*: 采用固定半径连接策略的批量版本; *h*: n 是样本数/算法迭代次数; *i*: 某些变量; *j*: 未明确分析; *k*: ϵ 是距最优代价所需的距离; *l*: 在增加假设的情况下可能更快; ***: 标记为无微分约束的系统; *†*: 标记某些非完整性系统。

演化。

设 $\Pi(\mathcal{X}, T)$ 为所有连续函数的集合 $[0, T] \rightarrow \mathcal{X}$ 并且 $x_{init} \in \mathcal{X}$ 为车辆的初始构型。目标区域为 $X_{goal} \subseteq \mathcal{X}$ 。 $t \in [0, T]$ 时刻的所有可用构型的集合表示为 $X_{free}(t)$, 用于编码完整性约束, 例如在经过的路径上避免与静态的或者 (可能的) 动态障碍物发生碰撞。轨迹上的微分约束用谓词 $D(x, x', x'', \dots)$ 表示, 可用于强制轨迹上的动态约束。进一步, 设 $J(\pi) : \Pi(\mathcal{X}, T) \rightarrow \mathbb{R}$ 为代价函数。在这些假设下, 最优化版本的轨迹规划问题通常可以表述为:

Problem IV.2 (最优轨迹规划问题). 给定 6-元组 $(\mathcal{X}_{free}, x_{init}, X_{goal}, D, J, T)$, 寻找 $\pi^* = :$

$$\begin{aligned} & \arg \min_{\pi \in \Pi(\mathcal{X}, T)} J(\pi) \\ & \text{subj. to } \pi(0) = x_{init} \text{ and } \pi(T) \in X_{goal} \\ & \quad \pi(t) \in \mathcal{X}_{free} \\ & \quad D(\pi(t), \pi'(t), \pi''(t), \dots) \quad \forall t \in [0, T] \end{aligned} \tag{18}$$

复杂度: 由于动态环境中的轨迹规划问题是静态环境中路径规划问题的一个推广应用, 因此该问题仍然是 PSPACE-hard 的。此外, 动态环境中的轨迹规划比路径规划更难; 因为在动态环境中考虑类似问题时, 在静态环境中可驾驭的变量会变得难以控制。特别是, 回想一下, 静态二维多边形环境时, 点机器人 (point robot) 可在多项式时间内有效找到最短路径; 研究人员将这种方式与 Canny 和 Reif[76] 的研究结果进行了对比, 证明了在移动的多边形障碍物中寻找速度有界的 (velocity bounded) 的完整性点机器人 (holonomic point robot) 的无碰撞轨迹属于一个 NP-hard 问题。类似地, 当在三维多面体环境中为具有固定自由度的机器人规划路径可驾驭时, Reif 和 Sharir[85] 证明了存在平移和旋转的三维多面体障碍物之间为具有 2 个自由度的机器人规划轨迹是一个 PSPACE-hard 问题。

对于自动驾驶过程中出现的非平凡轨迹规划问题, 通常无法获得精确算法, 因此普遍选择数值方法解决这类任务。轨迹规划问题可以直接在时域内用变分方法的数值进行求解, 也可以将轨迹规划问题转化额外加上时间维的构型空间中的路径规划问题 [86]。

从轨迹规划问题 $(\mathcal{X}_{free}^T, x_{init}^T, X_{goal}^T, D^T, J^T, T)$ 到路径规划问题 $(\mathcal{X}_{free}^P, x_{init}^P, X_{goal}^P, D^P, J^P)$ 的转换过程通常为: 产生路径规划的构型空间定义为: $\mathcal{X}^P := \mathcal{X}^T \times [0, T]$ 对于任意的 $y \in \mathcal{X}^P$, 符号 $t(y) \in [0, T]$ 表示时间分量, $c(y) \in \mathcal{X}^T$ 表示 y 点的 “构型” 分量。当路径 $\sigma(\alpha) : [0, 1] \rightarrow \mathcal{X}^P$ 能转换到轨迹 $\pi(t) : [0, T] \rightarrow \mathcal{X}^T$ 时, 约束路径起点和终点为:

$$\begin{aligned} t(\sigma(0)) &= 0 \\ t(\sigma(1)) &= T \end{aligned} \tag{19}$$

当路径是单调递增的, 通过微分进行约束:

$$t(\sigma'(\alpha)) > 0 \quad \forall \alpha \in [0, 1] \tag{20}$$

此外, 再将自由构型空间、初始构型、目标区域和差分约束映射到该路径规划的对应项, 如下所

示：

$$\begin{aligned}
 X_{\text{free}}^P &= \{(x, t) : x \in X_{\text{free}}^T(t) \wedge t \in [0, T]\} \\
 \mathbf{x}_{\text{init}}^P &= (\mathbf{x}_{\text{init}}^T, 0) \\
 X_{\text{goal}}^P &= \{(x, T) : x \in X_{\text{goal}}^T\} \\
 D^P(\mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots) &= D^T \left(c(\mathbf{y}), \frac{c(\mathbf{y}')}{t(\mathbf{y}')}, \frac{c(\mathbf{y}'')}{t(\mathbf{y}'')}, \dots \right)
 \end{aligned} \tag{21}$$

然后，利用一种能够处理微分约束并能转换回轨迹形式的路径规划算法，找到这种路径规划问题的解。

4.3 变分方法

我们将先在非线性连续优化框架下解决轨迹规划问题。此情况下的问题通常称为轨迹优化。在本小节中，我们将介绍轨迹规划公式，并理解这样做不会影响通用性，因为路径规划问题可以视为单位时间间隔内的轨迹优化问题。为利用现有的非线性优化方法，有必要将轨迹的无穷维函数空间投影到有限维向量空间。此外，大多数非线性规划方法须将 **Problem IV.2** 中的轨迹优化问题转换为以下形式：

$$\begin{aligned}
 \arg \min_{\pi \in \Pi(X, T)} \quad & J(\pi) \\
 \text{subj. to} \quad & \pi(0) = \mathbf{x}_{\text{init}} \text{ and } \pi(T) \in X_{\text{goal}} \\
 & f(\pi(t), \pi'(t), \dots) = 0 \quad \forall t \in [0, T] \\
 & g(\pi(t), \pi'(t), \dots) \leq 0 \quad \forall t \in [0, T]
 \end{aligned} \tag{22}$$

其中完整性约束和微分约束表示为由等式约束和不等式约束所组成的系统。

在某些应用中，使用惩罚函数或障碍函数 (barrier function) 可将约束优化问题松弛为无约束优化问题。在这两种情况下，约束皆被经过扩展的代价函数所取代。在惩罚函数法中，代价函数的形式为：

$$\tilde{J}(\pi) = J(\pi) + \frac{1}{\varepsilon} \int_0^T \left[\|f(\pi, \pi', \dots)\|^2 + \|\max(0, g(\pi, \pi', \dots))\|^2 \right] dt \tag{23}$$

同样，障碍函数可以用来代替不等式约束。此种情况下，增强代价函数形式为：

$$\tilde{J}(\pi) = J(\pi) + \varepsilon \int_0^T h(\pi(t)) dt \tag{24}$$

其中障碍函数满足 $g(\pi) < 0 \Rightarrow h(\pi) < \infty$, $g(\pi) \geq 0 \Rightarrow h(\pi) = \infty$, 和 $\lim_{g(\pi) \rightarrow 0} \{h(\pi)\} = \infty$ 。两个增强代价函数背后的含义是，通过使 ε 变小，最小代价将接近原始代价函数的最小值。障碍函数的一个优势是使局部极小值变得可行，但须用一个可行解来初始化才能获得到有限的增广代价。另一方面，惩罚函数的方法可以用任意轨迹初始化并优化为局部极小值。然而，局部极小值可能会违反问题的约束条件。文献 [60] 中提出了一个使用障碍函数的变分公式，其中使用了坐标变换将车辆保持在道路上的约束转化为线性约束。对数障碍法 (logarithmic barrier) 与类牛顿法 (Newton-like method) 一起使用，其使用方式与内点法类似。该方法对具体车辆模型在一段道路上的最短时间轨迹进行了有效计算。

接下来，我们讨论两类变分法：直接法和间接法。

直接法： 直接变量法背后的一般原理为将近似解限制到 $\Pi(X, T)$ 的有限维子空间上。为此，

通常有以下假设:

$$\pi(t) \approx \tilde{\pi}(t) = \sum_{i=1}^N \pi_i \phi_i(t) \quad (25)$$

其中 π_i 是 \mathbb{R} 上的系数, 而 $\phi_i(t)$ 是所选子空间的基函数。许多数值逼近方法证明将轨迹优化问题表示为非线性优化问题是可行的。我们在这里提到两种最常见的方式: 配置型数值积分器 (Numerical Integrators with Collocation) 和伪谱方法。

1) 基于配点法的数值积分器: 配点时, 要求近似的轨迹满足从离散点 $\{t_j\}_{j=1}^M$ 取值的约束。该要求衍生出两种离散约束系统: 一个为满足系统动力学约束的非线性等式方程系统:

$$f(\tilde{\pi}(t_j), \tilde{\pi}'(t_j)) = 0 \quad \forall j = 1, \dots, M \quad (26)$$

另一个为施加到轨迹上、满足状态约束的非线性不等式方程系统:

$$g(\tilde{\pi}(t_j), \tilde{\pi}'(t_j)) \leq 0 \quad \forall j = 1, \dots, M \quad (27)$$

数值积分技术用来逼近配点 (the collocation points) 之间的轨迹。例如, 分段线性基函数:

$$\phi_i(t) = \begin{cases} (t - t_{i-1}) / (t_i - t_{i-1}) & \text{if } t \in [t_{i-1}, t_i] \\ (t_{i+1} - t) / (t_{i+1} - t_i) & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

及配点一起使用就产生了 Euler 积分方法; 高阶多项式产生了 Runge-Kutta 系列积分方法。通过配点与 Euler 方法或 Runge-Kutta 方法之一来表示非线性方程比其他方法更为直接, 这使其成为一种技术潮流。文献 [14] 中设计出成功使用 Euler 方法对轨迹进行数值逼近的实验系统。

除了 Euler 方法, 文献 [87] 对 Adams 近似方法进行了研究, 以优化具体车辆模型的轨迹, 并证明它可以改进数值精度和收敛速度。

2) 伪谱方法: 数值积分技术利用了配点之间的插值函数离散化时间间隔。伪谱近似方法在它的基础上通过基表示插值函数。在配点间进行插值的典型基函数有基于 Legendre 或 Chebyshev 多项式的有限子集。与基本置点方法相比, 这类方法通常具有更好的收敛速度, 尤其是如 [88] 那样采用自适应方法选择配点和基函数时。

间接法: Pontryagin 的最小性原则 [89] 是最优控制的著名论断, 它提供了 **Problem IV.2** 存在解的最优化条件。间接法, 顾名思义, 通过找到满足这些最优化条件的解来解决问题。这些最优化条件通过管理系统状态和一组共态的集合的常微分方程组 (ODEs) 的增强型系统进行描述。然而, 这种常微分方程组系统会产生两点边值问题, 并且很难用数值方法求解。一种方法是改变问题的无约束初始条件, 并将系统向前集成, 以搜索能进入停止状态的初始条件。这种方法被称为射击法 (shooting method), 它的一个版本已经应用到了文献 [90] 所阐述的停车场下的移动规划中。间接法的优点, 例如射击法, 是将优化问题的维数降到状态空间的维数。

关于变分法的主题非常多, 因此, 以上仅是对所选方法的简要描述。有关此主题的专门研究, 请参见文献 [91] 和 [92]。

4.4 图搜索方法

虽然变分方法在许多情况下都很有用, 但其收敛性仅限于局部极小。在本节中, 我们将讨论一类方法, 这类方法尝试在离散化的路径空间中执行全局搜索来解决问题。这类所谓的图搜索

方法将车辆的构型空间 X 离散化并组织成图 (Graph) 的形式，然后在图上搜索代价最小的路径。

在基于图优化的方法中，构型空间用图表示 $G = (V, E)$ ，其中 $V \in X$ 是所选择的构型空间所组成的离散集合，称为顶点； $E = \{(o_i, d_i, \sigma_i)\}$ 是边的集合，其中 $o_i \in V$ 表示边的起始点， d_i 表示边的终点， σ_i 表示连接 o_i 和 d_i 的路径线段。假设路径线段 σ_i 连接两个顶点： $\sigma_i(0) = o_i$ 和 $\sigma_i(1) = d_i$ 。此外，假定初始构型 x_{init} 为图中的一个顶点。边的构造方式为：与其相关联的路径线段完全位于 X_{free} 中，同时满足微分约束。因此，图上的任何路径都可以通过将与图中路径相關的边相互连接的办法来将它转换为车辆的可通行路径。

通过离散化车辆的构型空间构造图的策略有很多。在下面的小节中，我们将讨论三种常见的策略：手工绘制的车道图、从几何表示派生的图以及通过控制或构型采样构造的图。

4.4.1 车道线图

当路径规划问题的应用环境为结构化道路时，一个充分的图分解可以充分表示车辆在每个车道内应遵循路径和穿过交叉口路径所需要的边。

车道线路图通常由更高级的街道网络图算法生成，部分为人工编辑生成。图 6 是这种图表的一个例子。

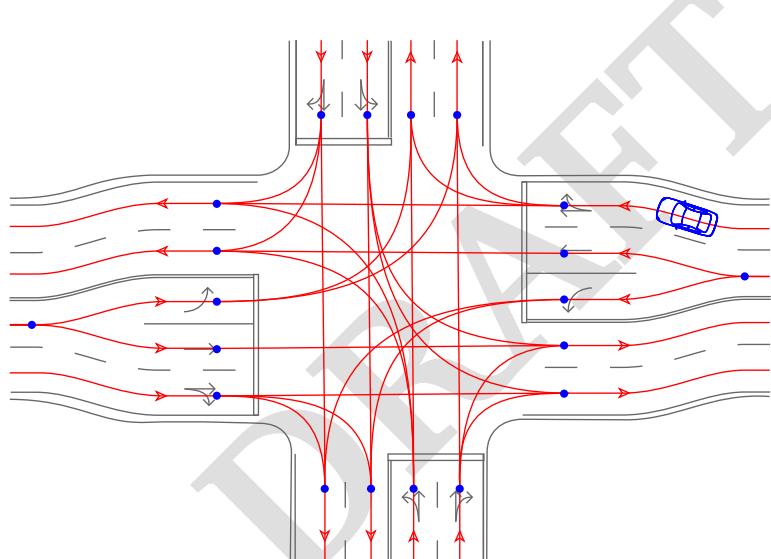


图 6：人工绘制的正常情况下所需的行驶路径。

尽管大多数情况下，自动驾驶车辆遵循车道线图的路径已经足够覆盖多数驾驶场景了，但有时它必须能够绕过在设计道路网络图时未予考虑的障碍物、或在路网图中未覆盖到的环境中进行巡航。例如，假如一辆发生故障的车辆阻塞了车辆计划要通行的车道——在这种情况下，必须使用更为通用的运动规划方法，以便在检测到的障碍物周围找到无碰撞路径。

一般的路径规划方法可以根据它们如何表示环境中的障碍物大致分为两类。所谓的几何方法或者组合方法常常用几何规则来描述障碍物，而在实际应用时通常使用多边形或多面体来描述障碍物。另一方面，所谓的基于采样的方法则对障碍物的内在表示形式进行抽象，并且设计一个函数，该函数用于检验任何给定的路径段是否会与障碍物发生碰撞。

4.4.2 几何方法

在本节中，我们将重点介绍用几何描述法对障碍物进行表示的路径规划。我们将首先介绍没有差分约束的路径规划，因为这一公式表明了有效、精确的路径规划算法的存在性。虽然没有考虑车辆的最小转弯半径的约束导致无法对传统转向车辆的路径规划方法实加差分约束，但这些方法可获得受曲率约束的路径长度的上下界，尤其是对就地转向的特殊车辆的路径规划问题。

在路径规划中，路线图 (roadmap) 一词用来描述一个离散图 X_{free} ，它很好地描述了自由构型空间的连通性情况，具有从路线图的某些顶点到达 X_{free} 中的任何点皆可通行的特性。当可以使用线性模型或半代数模型 (semi-algebraic model) 对集合 X_{free} 进行几何描述时，可以通过算法为 X_{free} 构造不同类型的路线图，并获得完整的路径规划算法。最重要的是，在构型空间中，当 $X_{free} \subseteq \mathbb{R}^2$ 并且采用多边形模型时，存在几种有效算法可用于构建这样的路线图，例如垂直单元分解法 [93]、广义 Voronoi 图法 [94]、[95] 和可视图法 (visibility graphs)[33]、[96]。对于由一个通用半代数模型描述的高维构型空间，一种称为柱面代数分解 (cylindrical algebraic decomposition) 的方法可用于在构型空间中构造路线图 [47]，[97]，从而得到一类对通用路径规划问题的完备算法。这类算法中速度最快的是由 Canny[65] 开发的算法，该算法在构型空间维度上具有（单个）指数时间的复杂度。然而，这一结论主要是理论性质的，迄今为止没有任何已知的实施例。

由于几何规划与汽车类车辆路径规划的相关性，对于最大曲率约束的路径规划问题，亦存在许多成果。Backer 和 Kirkpatrick[98] 为连接指定构型的最简单无障碍 Dubins 路径提出了一种在特征维数、输入精度和线段数量可在多项式时间内计算的曲率有界的路径构造算法。由于在障碍物中寻找曲率有界的最短路径属于一个 NP-hard 问题，因此不知道确切的多项式算法时间也不足为奇。Jacobs 和 Canny[99] 首次提出了一种可在多边形障碍物中寻找曲率有界的最短路径近似算法，Wang 和 Agarwal [100] 在该基础上进行了改进，将时间复杂度提升到 $O\left(\frac{n^2}{\epsilon^4} \log n\right)$ ，其中， n 为障碍物顶点的数目， ϵ 为近似因子。Boissonnat 和 Lazard[73] 对用曲率边界 κ 描述的光滑边界的所谓中等规模障碍物的特殊情况，提出了一种计算复杂度为多项式时间的精确算法，用于寻找曲率边界至多为 κ 的路径。

4.4.3 基于采样的方法

在自动驾驶中， X_{free} 的几何模型通常是不能直接使用的，从原始的传感器数据构建可通行路径代价过大。此外，对所产生的路径的要求往往比简单的最大曲率约束复杂得多。这可能解释了盛行使用基于采样方法的原因，这些技术不对自由构型集和动态约束施加具体的形式。基于采样的方法不对几何规则的表示进行推理，而是使用转向 (steering) 和碰撞检测 (collision checking) 来探索自由构型空间的可达性：

转向函数 `steer(x, y)` 返回从构型 x 开始向构型 y (但不一定到达 y) 的路段，以确保满足差分约束，即得到的运动对于所考虑的车型是可行的。转向函数的具体实现方式取决于使用它的上下文。文献中提到的一些典型选择为：

- 1) 随机转向：该函数返回一条路径，该路径通过在固定或可变的时间间隔里依据车辆的前向模型对状态 x 应用随机控制输入得到 [101]。

2) 启发式转向: 该函数返回一条应用启发式策略构造的路径, 以引导系统从 x 走向 y [102] ——[104]。过程涉及从预设计的机动离散集中选择机动动作。

3) 精确转向: 函数返回一条可行路径, 该路径引导系统从 x 点到 y 点。该路径与两点边值问题的解相对应。对于某些系统和代价函数, 可以获得解析解, 例如完整性系统的直线路径、独轮车前进的 Dubins 曲线路径 [57] 或双向独轮车的 Reeds-Shepp 曲线路径 [58]。对于差分平面系统 (differentially flat systems)[61] 也存在解析解; 而对于更复杂的模型, 通过求解两点边值问题可得到精确转向。

4) 最优精确转向: 该函数返回给定代价函数的最优精确转向路径。事实上, 假设代价函数等于路径弧长情况下, 前面讨论的直线、Dubins 曲线、Reeds-Shepp 曲线皆为最优解。

如果路径可在 X_{free} 中完整划分为 ρ 段, 则碰撞检测函数 $\text{col-free}(\rho)$ 返回 `true`, 以确保生成的路径不会与任何障碍物碰撞。

在获得转向并且进行碰撞检查的情况下, 现在挑战变成: 如何构造一个离散量, 使得它可以很好地近似 X_{free} 的连通性, 而无需访问其几何显式模型。我们从文献中回顾基于采样的离散化策略。

Algorithm 1: Recursive Roadmap Construction

```

 $V \leftarrow \{\mathbf{x}_{init}\}; E \leftarrow \emptyset; Q \leftarrow \text{new queue}(\mathbf{x}_{init});$ 
while  $Q \neq \emptyset$  do
     $\mathbf{x} \leftarrow \text{pop element from } Q;$ 
     $M \leftarrow \text{generate a set of path segments by applying}$ 
     $\text{motion primitives from configuration } \mathbf{x};$ 
    for  $\sigma \in M$  do
        if  $\text{col-free}(\sigma)$  then
             $E \leftarrow E \cup \{(\mathbf{x}, \sigma(1), \sigma)\};$ 
            if  $\sigma(1) \notin V$  then
                 $\text{add } \sigma(1) \text{ to } Q;$ 
                 $V \leftarrow V \cup \{\sigma(1)\};$ 

```

return (V, E)

一种直接的方法是选择一组运动原语 (固定操作), 并通过从车辆的初始构型 x_{init} 递归地应用这些运动原语来生成搜索图, 例如, 使用 Algorithm 1 中的方法。对于没有微分约束的路径规划, 运动基元可以是一组具有不同方向和长度的直线。对于汽车类车辆, 这样的运动原语可用一组具有不同数值的转向 (steering) 的弧表示汽车将要遵循的路径。可以使用多种技术来生成无人驾驶车辆的运动原语。一种简单的方法是对多个控制输入进行采样, 并使用车辆模型及时模拟前向运动以获得可行的运动。为获得连续的曲率路径, 有时也使用回旋弧线段 (clothoid segments)[105]。运动原语也可以通过记录由专家驾驶员驾驶的车辆的运动来获得 [106]。

注意, 运动原语的递归应用可能生成一个树图, 在最坏的情况下, 没有两条边会产生相同的构型。然而, 也有一些运动原语集, 称为晶格图元生成 (lattice-generating), 它们产生于类似晶格图元的规则图。见图 7。晶格生成图元的优点在于, 搜索图的顶点均匀覆盖构型空间, 而树通常在根顶点附近才可能高密度的顶点。Pivtoraiko 等人在文献 [107] 中使用术语 “状态晶格” (state lattice) 来描述这样的图形, 并指出可以通过首先在原点周围生成均匀间隔的构型, 然后通过路径将两种构型之间的问题表示为两点边值问题, 将原点连接到这样的系统, 获得针对现有系统的一组晶格生成运动原语 (lattice-generating motion primitives)。

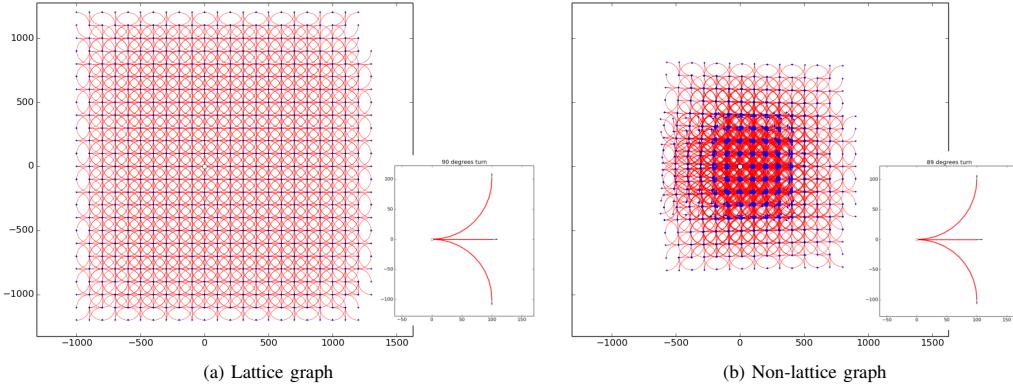


图 7: 具有 5000 条边的晶格图和非晶格图。(a) 递归应用 90° 左圆弧、 90° 右圆弧和直线得到的图形。(b) 递归应用 89° 左圆弧、 89° 右圆弧和直线得到的图形。这些原语的递归应用确实形成了一棵树，而不是在原点附近存在许多分支循环的晶格。因此，右图所覆盖的区域较小。

可以通过生成覆盖整个（自由）构型空间的离散样本集，并通过使用精确转向过程获得的可行路径段将它们连接起来，来实现类似于从初始构型中递归应用晶格生成运动晶格的效果。

多数基于采样的路线图构建方法遵循 Algorithm 2 中所示的算法方案，但在 `sample-points(\mathcal{X}, n)` 和 `neighbors(x, V)` 的实现上有所不同。函数 `sample-points(\mathcal{X}, n)` 表示从构型空间 \mathcal{X} 中选择 n 个点的策略，而函数 `neighbors(x, V)` 表示为顶点 x 选择一组相邻顶点 $N \subseteq V$ 的策略，该算法尝试使用精确的转向函数 `steerexact(x, y)` 通过线段连接点 x 。

Algorithm 2: Sampling-based Roadmap Construction

```

 $V \leftarrow \{\mathbf{x}_{\text{init}}\} \cup \text{sample-points}(\mathcal{X}, n); E \leftarrow \emptyset;$ 
for  $x \in V$  do
    for  $y \in \text{neighbors}(x, V)$  do
         $\sigma \leftarrow \text{steer}_{\text{exact}}(x, y);$ 
        if col-free( $\sigma$ ) then
             $E \leftarrow E \cup \{(x, y, \sigma)\};$ 
return ( $V, E$ )

```

函数 `sample-points(\mathcal{X}, n)` 的两种最常见的实现方式为：1) 返回排列在规则网格中的 n 个点，以及 2) 从 \mathcal{X} 返回 n 个随机采样的点。尽管随机采样具有普遍适用且易于实现的优点所谓的 Sukharev 网格已被证明可以在单位超立方体中实现 L_∞ 最优散布，即，它们最大程度地减小了内部没有采样点的最大空球的半径。在基于采样的路径规划中对随机和确定性采样的相关优点的深入讨论，请读者参考 [108]。实现 `neighbors(x, V)` 函数的两种最常用的策略为：1) 取 x 的 k 个近邻；2) 取中心为 x 、半径为 r 的球内的点集。

特别地，在 2 维时采用 4 领域或 8 领域、或者高维情况下类似做法的 d —维网格中以确定性方式排列的样本可视为自由构型空间的直接确定性离散化表示。一定程度上是因为它们来源于用位图的自由区域和占据区域表示机器人的构型空间 [109]。

Kavraki 等人 [110] 提倡在概率路线图（Probabilistic Roadmaps, PRM）框架内使用随机抽样，以便在高维构型空间中构造路线图，因为与网格（grids）不同，它们可以随时随地自然运行。PRM 的批处理版本 [79] 遵循 Algorithm 2 中的方案，具有随机采样和在固定半径为 r 的球内选择邻居点的特性。由于 PRM 的通用形式，它们已被应用于各种系统进行路径规划，包括具有微

分约束的系统。但是，该算法的理论分析主要集中在无差分约束的系统的算法性能上，即使用直线连接两个构型点时。在这样的假设下，文献 [80] 中证明 PRM 是概率完备的和渐近最优的 (probabilistically complete and asymptotically optimal)。也就是说，获得的图包含有效解（如果存在）的概率会随着图大小的增加而收敛，并且图中最短路径的代价收敛到最优代价。Karaman 和 Frazzoli [80] 提出了一种自适应的批处理 PRM 改编方案，称为 PRM*，该方法仅将球中的相邻顶点与对数收缩的半径进行连接，并随着样本数量的增加而保持渐近最优性和计算效率。

在同一篇论文中，作者提出了一种快速探索随机图 (Rapidly-exploring Random Graphs, RRG*) 的方法，它是一种增量离散化策略，在保持渐近最优性的同时，可以随时终止。最近，快速行进树 (Fast Marching Tree, FMT*) [111] 被认为是 PRM* 的一种非全局最优选择。该算法通过对一组采样顶点执行惰性动态规划递归，将离散化和搜索结合到一个过程中，随后该组采样点可用于快速确定从初始构型到目标区域的路径。

近年来，还对微分约束系统的理论分析进行了扩展。Schmerling 等人 [81] 提出了 PRM* 和 FMT* 的微分版本，并证明了无漂移控制仿射动力学系统 (driftless control-affine dynamical systems) 算法的渐近最优性，该类别包括不打滑的轮式车辆模型。

4.4.4 图搜索策略

在上一节中，我们讨论了用图的形式对自由构型空间进行离散化的技术。为了在这种离散化中获得实际的最优路径，必须采用一种图搜索算法。在本节中，我们将回顾与路径规划相关的图搜索算法。

在图中搜索最短路径的最为人知的算法可能是 Dijkstra 算法 [32]。该算法首先执行最佳搜索，以构建一棵树，表示从给定源顶点到图中所有其他顶点的最短路径。当只需要一条指向单个顶点的路径时，可以使用启发式算法来指导搜索过程。最著名的启发式搜索算法是 Hart, Nilsson 和 Raphael 开发的 A* 算法 [112]。如果提供的启发式函数是可容许的（即，它永远不会高于 cost-to-go 的代价），则 A* 被证明是最优且有效的，并且保证返回最优解。对于许多问题，使用 Weighted A*[113] 可以用较少的计算量获得有界次优解，这相当于简单地将启发式算法乘以常数因子 $\epsilon > 1$ 。可以证明，这种膨胀的启发式方法解得的路径其计算代价不会差于最优路径的 $(1 + \epsilon)$ 倍。

通常，每次使用感知的数据更新 world 模型时，都会反复寻找从车辆的当前构型到目标区域的最短路径。由于每次进行此类更新通常只影响图的一小部分，因此每次完全从头开始搜索可能会造成计算开销的浪费。实时重规划搜索系列算法，如 D*[114]，Focussed D*[115] 和 D* Lite[116]，被设计为每次底层图发生改变时就高效地重新计算最短路径，同时利用搜索的信息。

任何时候搜索算法试图快速提供第一条最佳路径，并采用更多的计算时间来不断改善解决方案。Anytime A* [117] 使用加权试探法找到第一个可行解，并通过以第一条路径的代价为上限，允许的启发值为下限继续搜索，从而实现 Anytime 行为；而 Anytime Repairing A* (ARA*) [118] 则使用权重递减的启发式算法执行一系列搜索，并重用前一次迭代的信息。另一方面，Anytime Dynamic A* (ADA*) [119] 结合了 D* Lite 和 ARA* 背后的思想，以生成用于在动态环境中进行实时重新计划的随时搜索算法。

在构型空间的离散化图上搜索路径的一个明显的限制是，在这样的图上得到的最优路径可

能比配置空间中的真正最短路径长得多。任何角度路径规划算法 [120]-[122] 都设计用于在网格上操作，或者更一般地在表示自由配置空间的单元分解的图上操作，并且通过在搜索期间考虑图上顶点之间的“快捷方式”来尝试减轻这一缺点。此外，字段 D* 将线性插值引入搜索过程以生成平滑路径 [123]。

在配置空间的离散化图上搜索路径的算法的明显局限性在于，此类图上的最佳路径可能比构型空间中真正的最短路径要长。任何角度路径规划算法(Any-angle path planning algorithms)[120]-[122] 设计为在网格上、或者更一般地在表示自由构型空间的单元进行分解后的图上操作，并且在搜索期间考虑图上顶点之间的“捷径”(shortcuts) 来尝试弥补这一缺点。另外，Field D* 在搜索过程中引入线性插值以生成平滑路径 [123]。

4.5 增量搜索方法

在固定的离散化图上搜索的技术的一个缺点是，它们仅可从图离散化后由原语构造的路径集中搜索。因此，这些方法可能无法返回可行的路径或仅返回明显次优的路径。

基于增量的运动规划器尝试解决此问题，并在给定的计算时间下为任何运动规划问题实例（如果存在的话）提供可行运动路径。通常，这些方法在尝试离散化过程中确定每个步骤中是否存在从初始构型到目标区域的路径，同时，逐步增加构型空间的精细化程度。如果实例“简单”，则可以快速提供解决方案，但通常计算时间不受限制。类似地，增量最优运动规划方法在快速找到可行路径的基础上，试图提供一系列质量不断提高、收敛到最优路径的解。

在文献中使用术语“概率完备”来描述找到一个解决方案的算法（如果存在的话），随着计算时间的增加，其概率会接近“概率完备”。注意，如果不存在解决方案，则概率完备算法可能不会终止。类似地，术语“渐近最优”用于描述以概率 1 收敛到最优解的算法。

在固定的离散化构型空间下获得极端完备性和最优性的朴素策略是求解一系列路径规划问题，每次求解时提高离散化的分辨率。这种方法的一个缺点是，各个层次的分辨率下路径规划过程是独立的，没有重用任何信息。此外，在启动新的图搜索之前应该以多快速度提高离散化的分辨率还不明确，即，添加一个新配置是否适合，将配置数量加倍，还是将每个配置空间维度上的离散值加倍，为克服这些问题，增量运动规划方法将构型空间的增量离散化与路径搜索集成并交织在一个过程中。

用于增量路径规划的一类重要方法是基于这样的想法，即，将根为车辆初始构型的树向外逐渐生长，以探索可到达的构型空间。“探索性”行为通过从树中迭代选择一个随机节点并通过应用转向函数扩展所选顶点。一旦树生长到足以到达目标区域，就可以通过跟踪从目标区域中的顶点向后回溯到初始构型的链接来恢复所生成的路径。算法 3 描述了增量树算法的一般算法方案。

Algorithm 3: Incremental Tree-based Algorithm

```
V ← {xinit} ∪ sample-points(ℳ, n); E ← ∅;
while not interrupted do
    xselected ← select(V);
    σ ← extend(xselected, V);
    if col-free(σ) then
        xnew = σ(1);
        V ← V ∪ {xnew}; E ← E ∪ {(xselected, xnew, σ)};
return (V, E)
```

最早提出的基于树的随机增量规划器之一是 Hsu 等人的扩展空间树 (expansive spaces tree, EST) 规划器 [124]。该算法从 V 中随机选择一个扩展顶点 $x_{selected}$, 其概率与邻域中的顶点数量成反比, 从而鼓励向未探索区域增长。在扩展过程中, 该算法在 $x_{selected}$ 的固定半径附近采样一个新顶点 y , 并使用相同的技术对采样过程进行偏袒以从相对较少探索的区域中选择一个顶点。然后, 它返回 $x_{selected}$ 和 y 之间的直线路径。文献 [125] 在动态环境中对具有运动学约束的规划将该思想进行推广, 在不同的非完整性机器人系统上演示了该算法的能力, 并且作者使用算法的理想化版本来确定无法找到可行路径的概率取决于状态空间的扩展特性, 并随样本数量呈指数衰减。

快速探索随机树 (Rapidly-exploring Random Trees, RRT) [101] 是 La Valle 提出的一种寻找高维非完整系统可行轨迹的有效方法。通过从自由构型空间中抽取随机样本 x_{rnd} , 并在随机样本的方向扩展树, 实现了快速搜索。在 RRT 中, 顶点选择函数 $select(V)$ 根据两个构型之间给定的距离度量方式返回随机样本 x_{rnd} 的最近邻。扩展函数 $extend()$ 在构型空间中为能最小化到 x_{rnd} 距离的固定时间步长应用一个控制以生成一条路径, 在某些简化假设条件下 (使用随机转向进行扩展), RRT 算法已被证明是概率完备的 [83]。我们指出, 关于概率完备性的结果不容易推广到许多已在实际实施的 RRT 的版本, 这些版本通常采用启发式策略进行转向控制。事实上, 文献 [126] 最近已经证明, 使用固定时间步长的启发式转向策略的 RRT 并不是概率完备的。

Algorithm 4: RRT* Algorithm. The cost-to-come to vertex \mathbf{x} is denoted as $c(\mathbf{x})$, the cost of path segment σ is denoted as $c(\sigma)$ and the parent vertex of vertex \mathbf{x} is denoted by $p(\mathbf{x})$.

```

 $V \leftarrow \{\mathbf{x}_{\text{init}}\}; E \leftarrow \emptyset;$ 
while not interrupted do
     $\mathbf{x}_{\text{selected}} \leftarrow \text{select}(V);$ 
     $\sigma \leftarrow \text{extend}(\mathbf{x}_{\text{selected}}, V);$ 
    if col-free( $\sigma$ ) then
         $\mathbf{x}_{\text{new}} = \sigma(1) ;$ 
         $V \leftarrow V \cup \{\mathbf{x}_{\text{new}}\};$ 

        // consider all vertices in ball
        // of radius  $r$  around  $\mathbf{x}_{\text{new}}$ 
         $r = \gamma \sqrt[d]{\frac{\log |V|}{|V|}};$ 
         $X_{\text{near}} \leftarrow \{\mathbf{x} \in V \setminus \{\mathbf{x}_{\text{new}}\} : d(\mathbf{x}_{\text{new}}, \mathbf{x}) < r\};$ 

        // find best parent
         $\mathbf{x}_{\text{par}} \leftarrow \arg \min_{\mathbf{x} \in X_{\text{near}}} c(\mathbf{x}) +$ 
         $c(\text{connect}(\mathbf{x}, \mathbf{x}_{\text{new}}))$  subj. to
         $\text{col-free}(\text{steer}_{\text{exact}}(\mathbf{x}, \mathbf{x}_{\text{new}}));$ 
         $\sigma' = \text{steer}_{\text{exact}}(\mathbf{x}_{\text{par}}, \mathbf{x}_{\text{new}});$ 
         $E \leftarrow E \cup \{(\mathbf{x}_{\text{par}}, \mathbf{x}_{\text{new}}, \sigma')\};$ 

        // rewire vertices in neighborhood
        for  $\mathbf{x} \in X_{\text{near}}$  do
             $\sigma' \leftarrow \text{steer}_{\text{exact}}(\mathbf{x}_{\text{new}}, \mathbf{x});$ 
            if  $c(\mathbf{x}_{\text{new}}) + c(\sigma') < c(\mathbf{x})$  and
             $\text{col-free}(\sigma')$  then
                 $E \leftarrow (E \setminus \{(p(\mathbf{x}), \mathbf{x}, \sigma'')\}) \cup$ 
                 $\{(\mathbf{x}_{\text{new}}, \mathbf{x}, \sigma')\},$  where  $\sigma''$  is the path
                from  $p(\mathbf{x})$  to  $\mathbf{x};$ 

return (V,E)

```

此外, Karaman 和 Frazzoli[127] 证明了 RRT 以概率 1 收敛到次优解, 并设计了渐近最优自适应版本的 RRT 算法, 称为 RRT*。如算法 4 所示, RRT* 的每一次迭代中都会考虑位于新添加顶点 x_{new} 附近的一组顶点, 并且 a) 将 x_{new} 连接到邻域中的顶点, 该顶点使从 x_{init} 到 x_{new} 的路径代价最小, b) 如果领域中的任意顶点能使从 x_{init} 到该顶点的代价下降, 则将领域中的任意顶点重新布线到 x_{new} 。该算法的一个重要特点是将邻域定义为以 x_{new} 为中心的球, 其半径为一个关于树的大小的函数: $r = \gamma \sqrt[d]{(\log n)/n}$, 其中 n 为树的顶点数, d 是构型空间的维数, γ 为一个与实例相关的常数。结果表明, 对于这种函数, 球中期望的顶点数为树的大小的对数, 这对于确保算法几乎能确定性地收敛到最优路径、同时保持与次优 RRT 算法拥有相同的渐近复杂性是很必要的。

在微分约束下, RRT* 渐近最优的充分条件在文献 [80] 中进行了说明, 该文献还证明 Dubins 车辆和双积分器系统 (double integrator systems) 是满足该条件的。之后的工作中, 作者进一步证明了对于短时局部系统 (small-time locally attainable systems), 该算法不仅可以保持渐近最优性, 而且还可以保持计算效率 [82]。其他相关工作着重于通过局部线性化系统动力学 [128] 或通过导出具有线性动力学的系统的闭式解来推导非完整系统的距离和转向函数 [129]。另一方面, RRTX

是一种 RRT* 的扩展算法，当障碍物区域发生变化时（例如，面对来自传感器的新数据），可以进行实时增量重规划 [130]。

基于采样的算法的发展包括在无需了解精确控制过程情况下实现渐近最优算法。特别的是，Li 等人 [84] 最近提出了渐近（近似）最优地规划路径的稳定稀疏树 (Stable Sparse Tree, SST) 方法，该方法基于通过系统动力学的前向模型传播的随机采样控制树，对树进行剪枝以保持稀疏性从而确保局部次优。

4.6 实际部署

本节讨论了三类自动驾驶车辆的路径规划方法：变化方法，图搜索方法和基于增量树的方法。实际部署在自动驾驶系统上的算法全部来源于上述类别。例如，即使在 DARPA Urban Challenge 的前四名中，用于运动规划的方法也存在显著差异。作为此次比赛的赢家，CMU 的 Boss 车辆在结构化环境中使用变分方法生成局部轨迹，并在停车场环境中通过四维构型空间（包含位置，方向和速度）中应用晶格图 (lattice graph) 以及 Anytime D* 来寻找无碰撞路径 [131]。据报道，获得亚军的斯坦福大学团队开发了一种称为混合 A*(Hybrid A*) 的搜索策略，该搜索策略在搜索过程中通过递归地应用有限集的一组动作来惰性地构造运动原语树 (a tree of motion primitives)。搜索过程由精心设计的启发式方法进行引导，树的稀疏性通过在构型空间指定区域内保留单个节点进行保证 [132]。类似地，获得第三名的来自弗吉尼亚理工大学 (Virginia Tech) 的 VictorTango 团队开发使用了包含可能操作的图离散化技术，并使用 A* 算法对图进行搜索 [133]。最后，麻省理工学院开发的车辆则使用了带有偏置采样的 RRT 算法的变体，称为闭环 RRT(closed-loop RRT)[134]。

5 车辆控制

运动规划过程可以为问题 IV.1 或 IV.2 提供解。反馈控制器的作用是在存在建模误差和其他形式的不确定性的情况下稳定参考路径或轨迹。取决于运动规划器提供的参考路径，车辆控制的目标为稳定路径 (path stabilization) 或稳定轨迹 (trajectory stabilization)。路径稳定问题可正式描述如下：

Problem V.1. (路径稳定问题) 给定一个关于控制的微分方程 $\dot{x} = f(x, u)$ ，参考路径为 $x_{ref} : \mathbb{R} \rightarrow \mathbb{R}^n$ ，速度为 $v_{ref} : \mathbb{R} \rightarrow \mathbb{R}$ ，找到一个可行法则， $u(x)$ ，使得 $\dot{x} = f(x, u(x))$ 的解满足：对 $\forall \varepsilon > 0$ 和 $t_1 < t_2$ ，存在 $\delta > 0$ 和一个可微变量 $s : \mathbb{R} \rightarrow \mathbb{R}$ ，使得：

- 1) $\|x(t_1) - x_{ref}(s(t_1))\| \leq \delta \Rightarrow \|x(t_2) - x_{ref}(s(t_2))\| \leq \varepsilon$
- 2) $\lim_{t \rightarrow \infty} \|x(t) - x_{ref}(s(t))\| = 0$
- 3) $\lim_{t \rightarrow \infty} \dot{s}(t) = v_{ref}(s(t))$

定性地，这些条件为：(1) 一个较小的初始跟踪误差，并且一直将此误差保持得很小，(2) 跟踪误差必须收敛到零，并且 (3) 沿参考路径前进的速率趋于名义速率。

许多已提出的车辆控制规则，包括本节所讨论的规则，皆使用以下形式的反馈法则：

$$u(x) = f \left(\operatorname{argmin}_{\gamma} \|x - x_{ref}(\gamma)\| \right) \quad (29)$$

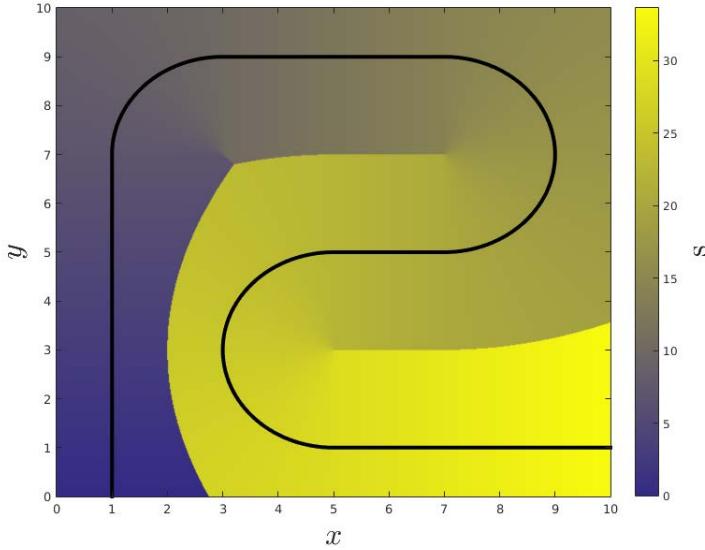


图 8：黑色表示的是公式(33)的参考路径。颜色表示平面中的每个点的 s 的值，同时表示式(33)的不连续性。

其中，反馈是一个关于参考路径上的最近点的函数。该形式的控制存在的一个重要问题是闭环矢量场 $f(x, u(x))$ 不是连续的。如果路径是自相交的或在某个点不可微，则 $f(x, u(x))$ 的不连续点将直接位于路径上。如果无人车辆运行的是不连续的轨迹，会出现无法预测的行为。图 8 说明了这种不连续性。文献 [135] 提出了一种不使用式(29)的反馈定律的反推控制设计。

轨迹稳定问题更为直观，但是这种控制器更容易遇到性能瓶颈 [136]。

Problem V.2. (轨迹稳定问题) 给定一个控制微分方程 $\dot{x} = f(x, u)$ 及一个参考轨迹 $x_{ref}(t)$ ，找到 $\pi(x)$ ，使得方程 $\dot{x} = f(x, u)$ 的解满足以下准则： $\forall \varepsilon > 0$ and $t_1 < t_2$ ，存在 $\delta > 0$ ，使得：

- 1) $\|x(t_1) - x_{ref}(t_1)\| \leq \delta \Rightarrow \|x(t_2) - x_{ref}(t_2)\| \leq \varepsilon$
- 2) $\lim_{t \rightarrow \infty} \|x(t) - x_{ref}(t)\| = 0$

许多情况下，可以减少分析轨迹的稳定性来确定时变系统中原点的稳定性。李雅普诺夫定理的基本形式仅适用于时不变系统。但是，时变系统的稳定性理论也已建立（例如 [137，定理 4.9]）。

在许多情况下，分析轨迹的稳定性可以归结为确定时变系统中原点的稳定性。李雅普诺夫定理 (Lyapunov's theorem) 的基本形式仅适用于时不变系统。然而，时变系统的稳定性理论也得到了很好的建立（如 [137，定理 4.9]）。

一些用于各种稳定性场合的有用限定词包括：时变系统的一致渐近稳定性 (Uniform Asymptotic Stability) 断言，上述问题条件 1) 中的 δ 与 t_1 无关。指数稳定性 (Exponential Stability) 断言，收敛速度受指数衰减的限制。

应该注意的一个棘手问题为，随着时间趋于无穷大，控制器的规范通常用渐近跟踪误差表示。实际上，参考轨迹是有限的，因此还应考虑系统的瞬态响应。

本节的其余部分将选择性地介绍适用于无人驾驶汽车的控制设计。表2中提供了选定的控制器的概述。

第5.1小节详细介绍了用于运动学模型的路径稳定的许多有效控制策略，而第5.1小节则讨论

Controller		Model	Stability	Time Complexity	Comments/Assumptions
Pure Pursuit	(V-A1)	Kinematic	LES* to ref. path	$O(n)^*$	No path curvature
Rear wheel based feedback	(V-A2)	Kinematic	LES* to ref. path	$O(n)^*$	$C^2(\mathbb{R}^n)$ ref. paths
Front wheel based feedback	(V-A3)	Kinematic	LES* to ref. path	$O(n)^*$	$C^1(\mathbb{R}^n)$ ref. paths; Forward driving only
Feedback linearization	(V-B2)	Steering rate controlled kinematic	LES* to ref. traj.	$O(1)$	$C^1(\mathbb{R}^n)$ ref. traj.; Forward driving only
Control Lyapunov design	(V-B1)	Kinematic	LES* to ref. traj.	$O(1)$	Stable for constant path curvature and velocity
Linear MPC	(V-C)	$C^1(\mathbb{R}^n \times \mathbb{R}^m)$ model ‡	LES* to ref. or path	$O\left(\sqrt{N} \ln\left(\frac{N}{\epsilon}\right)\right)^\dagger$	Stability depends on horizon length
Nonlinear MPC	(V-C)	$C^1(\mathbb{R}^n \times \mathbb{R}^m)$ model ‡	Not guaranteed	$O(\frac{1}{\epsilon})^\ddagger$	Works well in practice

表 2: 本节讨论的控制器概述。图例: *: 局部指数稳定性 (local exponential stability, LES); *: 假设式(29)是通过对路径或轨迹用 n 个点离散化线性搜索求出的; \dagger : 假设使用内点法求解 (V28), 时间范围为 n 且求解精度为 ϵ ; \ddagger : 使用梯度下降法基于 (V.25) 用渐近收敛速度到局部极小值。它不保证返回解或者找到全局最小值。 \sharp : 由 \mathbb{R}^m 上的输入所定义的 \mathbb{R}^n 上的矢量场, 是一个连续可微函数, 从而定义了关于参考路径的代价梯度或者线性化。

了轨迹稳定技术。在第5.3小节讨论的是预测控制策略对于复杂的车辆模型同样有效, 并且可以应用于路径稳定和轨迹稳定。

5.1 基于运动学模型的路径稳定

5.1.1 纯追踪 (Pure Pursuit)

纯追踪方法是最早提出路径跟踪策略的方法。第一次对它进行讨论出现于文献 [138] 中, 并在文献 [52] 和 [139] 得到详细阐述。由于其简单的实现方式及令人满意的计算性能, 该策略及其变种方法 (如 [140]、[141]) 已被证明是车辆控制不可或缺的技术。许多出版刊物在其报告中提及使用了纯追踪控制器, 包括参加 DARPA Grand Challenge[8] 的两辆汽车和参加 DARPA Urban Challenge [9] 的三辆汽车。

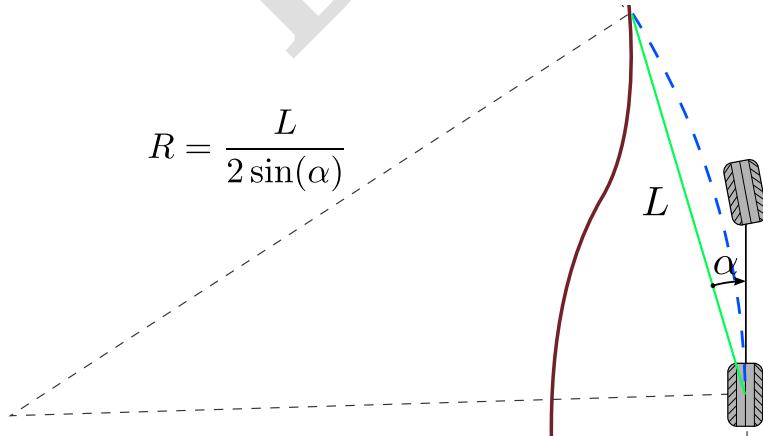


图 9: xx。

控制的原则是通过车辆的当前构型拟合一个半圆到车辆前方距离为 L 处 (称为前视距离) 的

位于参考路径的一个点。图 9 说明了它的几何外形。该圆被定义为经过汽车的当前位置、同时经过汽车前方一个前视距离处的位于参考路径上的点，并且圆与汽车的方向相切。圆的曲率计算方式由下式给出：

$$\kappa = \frac{2 \sin(\alpha)}{L} \quad (30)$$

给定车速 v_r ，指令的转向速率为：

$$\omega = \frac{2v_r \sin(\alpha)}{L} \quad (31)$$

原版的控制器 [138] 中，直接从摄像机的输出数据中计算角度 α 。然而， α 可以在惯性坐标系下定义一个状态反馈控制来表示。考虑构型 $(x_r, y_r, \theta)^\top$ 及路径上的点 $x_{ref}(s), y_{ref}(s)$ ，使得： $\|(x_{ref}(s), y_{ref}(s)) - (x_r, y_r)\| = L$ 由于在路径上存在多个类似性质的点，因而取参数 s 最大值的那个点来定义控制的唯一性。此时， α 由下式给出：

$$\alpha = \arctan\left(\frac{y_{ref} - y_r}{x_{ref} - x_r}\right) - \theta \quad (32)$$

假设路径没有曲率并且车速恒定（可能为负值），则纯跟踪控制器解决的是 **Problem V.1**。对于固定的非零曲率，纯追踪的稳态跟踪误差很小。

如果车辆到路径的距离大于 L ，则不定义控制器输出。另一个考虑因素是参考路径的曲率变化会导致汽车偏离参考轨迹。这种方式对于沿道路行驶的情况是可以接受的，但对于跟踪原地转向 (parking maneuver) 这样的操作可能会出现问题。最后，随着车速的增加，航向速度指令 ω 对反馈角 α 越来越敏感。解决此问题的常用方法是根据车速缩放 L 。

5.1.2 后轮位置反馈 (Rear wheel position based feedback)]

下一种方法是使用后轮位置作为输出来稳定后轮的名义路径 [56]。控制器分配量为：

$$s(t) = \underset{\gamma}{\operatorname{argmin}} \| (x_r(t), y_r(t)) - (x_{ref}(\gamma), y_{ref}(\gamma)) \| \quad (33)$$

关于参考路径和包含参考路径的有限域的假设在文献 [56] 中进行了详细描述，其中式(33)为一个连续函数。从位置 $s(t)$ 到参考路径的单位正切角 (unit tangent) 为：

$$\hat{t} = \frac{\left(\frac{\partial x_{ref}}{\partial s}|_{s(t)}, \frac{\partial y_{ref}}{\partial s}|_{s(t)} \right)}{\left\| \left(\frac{\partial x_{ref}(s(t))}{\partial s}, \frac{\partial y_{ref}(s(t))}{\partial s} \right) \right\|} \quad (34)$$

跟踪误差向量为：

$$d(t) := (x_r(t), y_r(t)) - (x_{ref}(s(t)), y_{ref}(s(t))) \quad (35)$$

这些值用于从路径 e 计算中计算一个横向误差坐标 (a transverse error coordinate)，其中路径 e 是两个向量之间的叉积：

$$e = d_x \hat{t}_y - d_y \hat{t}_x \quad (36)$$

其中下标表示向量的分量索引。控制使用车辆的航向向量和到路径的切向向量之间的夹角 θ_e ：

$$\theta_e(t) = \theta - \arctan_2\left(\frac{\partial y_{ref}(s(t))}{\partial s}, \frac{\partial x_{ref}(s(t))}{\partial s}\right) \quad (37)$$

几何形状如图 10 所示。

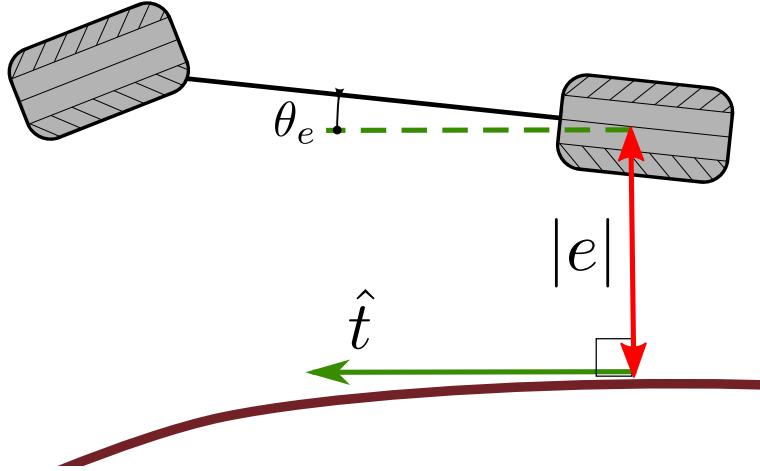


图 10: 基于后轮的反馈控制的反馈变量。 θ_e 是路径的最近点到后轮的切线方向角与汽车行驶方向角之差。标量值 e 的大小用红色显示。图示为 $e > 0$ 情况；对于汽车在路径左侧的情况， $e < 0$ 。

将坐标更改为 (s, e, θ_e) 后得到：

$$\begin{aligned}\dot{s} &= \frac{v_r \cos(\theta_e)}{1 - \kappa(s)e} \\ \dot{e} &= v_r \sin(\theta_e) \\ \dot{\theta}_e &= \omega - \frac{v_r \kappa(s) \cos(\theta_e)}{1 - \kappa(s)e}\end{aligned}\quad (38)$$

其中 $\kappa(s)$ 表示路径在 s 处的曲率。以下航速指令为二次连续可微的路径提供局部渐近收敛性：

$$\omega = \frac{v_r \kappa(s) \cos(\theta_e)}{1 - \kappa(s)e} - g_1(e, \theta_e, t) \theta_e - k_2 v_r \frac{\sin(\theta_e)}{\theta_e} \quad (39)$$

其中， $g_1(e, \theta_e, t) > 0, k_2 > 0, v_r \neq 0$ ，文献 [56] 使用式(38)定义的坐标系统通过 Lyapunov 函数 $V(e, \theta_e) = e^2 + \theta_e^2/k_2$ 进行验证。路径为二次可微的要求来源于反馈准则中出现曲率。这种控制准则的优点是稳定性不受 v_r 的符号的影响，使得它适合于倒车情况。设置 $g_1(v_r, \theta_e, t) = k_\theta |v_r|$ ；当 $k_e > 0$ 时，只需要 $v_r \neq 0$ ，即可用跟车速无车的速率实现局部指数收敛。此种情况下的控制准则为：

$$\omega = \frac{v_r \kappa(s) \cos(\theta_e)}{1 - \kappa(s)e} - (k_\theta |v_r|) \theta_e - \left(k_e v_r \frac{\sin(\theta_e)}{\theta_e} \right) e \quad (40)$$

5.1.3 前轮位置反馈 (Front wheel position based feedback)

斯坦福大学在参加 2005 年 DARPA 大挑战赛 (DARPA Grand Challenge) 时提出并使用了这种方法 [55][8]。该方法将前轮位置作为调节变量。像前面小节一样，控制使用变量 $s(t)$ 、 $e(t)$ 和 $\theta_e(t)$ ，但修改 $e(t)$ 使用前轮位置而不是后轮位置进行计算。取横向误差的时间导数：

$$\dot{e} = v_f \sin(\theta_e + \delta) \quad (41)$$

对于幅度小于 v_f 的错误率，可直接通过转向角控制式(41)的错误率。求解出转向角，使 $\dot{e} = -ke$ 可驱使 $e(t)$ 以指数速率快速降至零。

$$\begin{aligned}v_f \sin(\delta + \theta_e) &= -ke \\ \Rightarrow \delta &= \arcsin(-ke/v_f) - \theta_e\end{aligned}\quad (42)$$

在这种情况下， θ_e 项不能解释为航向误差，因为即使存在着理想化的跟踪，它也不会为零。可以恰当地将其解释为可追踪出参考路径的名义转向角的前馈项和追踪误差项的组合。

这种控制法则的缺陷是，当 $|ke/v_f| > 1$ 时，没有对它进行定义。利用反馈定律，可以将有限域上的指数收敛放宽为局部指数收敛：

$$\delta = \arctan(-ke/v_f) - \theta_e \quad (43)$$

它关于 e 的一阶形式与前面的表达式相同。图 11 对此进行了说明。

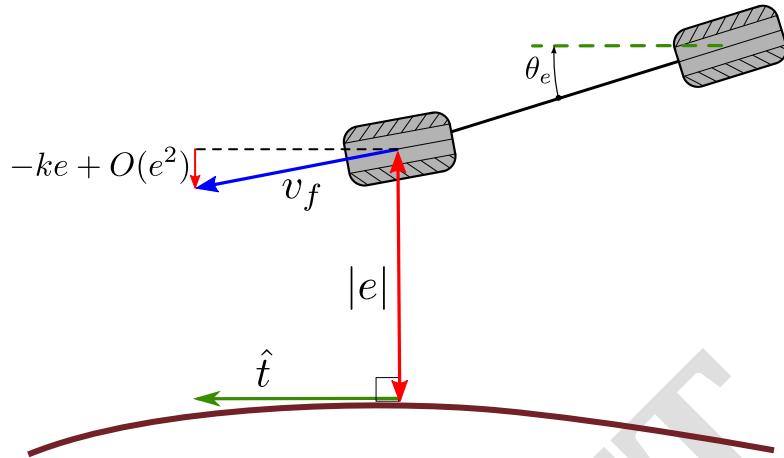


图 11：基于前轮输出的控制。控制策略是将前轮指向路径，使前轮垂直于路径的速度分量并与路径的距离成正比。图中为局部实现的方式，能以局部指数进行收敛。

与式(42)中的控制定律一样，在路径连续可微的条件下，该控制器将汽车稳定到曲率变化的局部路径上。路径的条件来自反馈策略中 θ_e 的定义。该控制器的缺点在于，它在倒车时不够稳定，因此不适合用于停车。

运动学模型路径跟踪控制器的比较： 基于运动学模型的控制器对车轮有无打滑约束，其优点是计算量低，易于实现，并且在中等速度下具有良好的性能。图 12 提供了本节讨论的路径稳定控制器基于式(3)进行车道变更模拟的定性比较。在基于前轮输出的控制器的仿真实验中，后轮参考路径用前轮参考路径进行替代，同时满足以下条件：

$$\begin{aligned} x_{ref}(s) &\mapsto x_{ref}(s) + l \cos(\theta) \\ y_{ref}(s) &\mapsto y_{ref}(s) + l \sin(\theta) \end{aligned} \quad (44)$$

这次仿真实验的参数总结于表 3 中。

参看图 12，纯跟踪控制在路径无曲率期间跟踪的参考路径。纯跟踪控制使系统偏离参考路径。与之相反，后两个控制器收敛到参考路径并通过高曲率区域特性对它进行跟踪。

使用式(33)作为反馈策略的两个控制器中，只有在参考路径的领域附近，此时式(33)是连续的情况下，局部收敛才能获得证明。直观上看，这意味着路径不能跨越自身，必须是可微的。

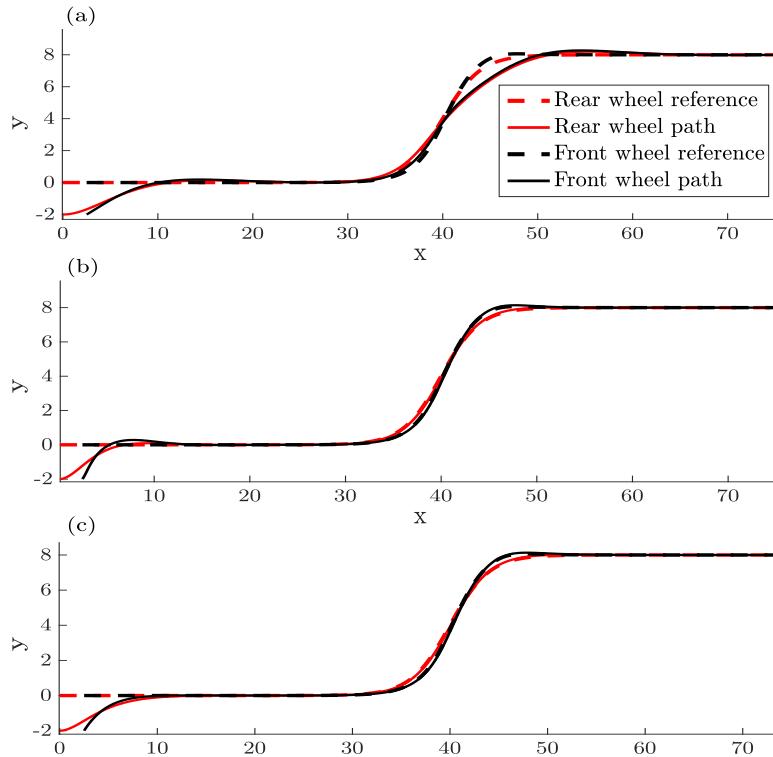


图 12: 本节讨论的三种路径稳定控制原则的跟踪性能比较。**(a)** 当曲率不为零时, 纯追踪偏离参考路径。**(b)** 基于后轮输出的控制器将后轮驱动到后轮参考路径。超调 (Overshoot) 为系统二阶响应的结果。**(c)** 基于前轮输出的控制器以一阶响应将前轮驱动到参考路径。。

Example Parameters			
Reference path	Wheel-base	Steering limit	Initial Configuration
$(x_{ref}(s), y_{ref}(s)) = (s, 4 \cdot \tanh(\frac{s-40}{4}))$	$L = 5$	$ \delta \leq \pi/4$	$(x_r(0), y_r(0), \theta(0)) = (0, -2, 0)$
Controller Parameters			
Pure pursuit	Rear wheel feedback		Front wheel feedback
$L = 5, v_r = 1$	$k_e = 0.25, k_\theta = 0.75, v_r = 1$		$k = 0.5, v_r = 1$

表 3: 用于生成图 12 的仿真和控制参数。

5.2 基于运动学模型的轨迹控制

5.2.1 设计 Control-Lyapunov 函数 (Control Lyapunov based design)

[142] 中描述了基于控制 Lyapunov 函数的控制设计。该方法是在固定在汽车上的坐标系中定义跟踪误差。配置误差可以用参考轨迹和速度从惯性坐标系的基础变化来表示，

文献 [142] 中讨论了基于 Control-Lyapunov 函数 (控制——李雅普诺夫函数) 的一种控制设计方式。该方法在汽车固定的一个坐标系中定义跟踪误差。通过使用参考轨迹、惯性坐标、及 $(x_{ref}, y_{ref}, \theta_{ref}, v_{ref}, \omega_{ref})$ 可从惯性坐标系中改变基准来表示构型误差。

$$\begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ref} - x_r \\ y_{ref} - y_r \\ \theta_{ref} - \theta \end{pmatrix} \quad (45)$$

之后，构型误差演示为：

$$\begin{aligned} \dot{x}_e &= \omega y_e - v_r + v_{ref} \cos(\theta_e) \\ \dot{y}_e &= -\omega x_e + v_{ref} \sin(\theta_e) \\ \dot{\theta}_e &= \omega_{ref} - \omega \end{aligned} \quad (46)$$

则分配的控制量为：

$$\begin{aligned} v_r &= v_{ref} \cos(\theta_e) + k_1 x_e \\ \omega &= \omega_{ref} + v_{ref} (k_2 y_e + k_3 \sin(\theta_e)) \end{aligned} \quad (47)$$

闭环误差动力学系统 (the closed loop error dynamics) 变为：

$$\begin{aligned} \dot{x}_e &= (\omega_{ref} + v_{ref} (k_2 y_e + k_3 \sin(\theta_e))) y_e - k_1 x_e \\ \dot{y}_e &= -(\omega_{ref} + v_{ref} (k_2 y_e + k_3 \sin(\theta_e))) x_e + v_{ref} \sin(\theta_e) \\ \dot{\theta}_e &= \omega_{ref} - \omega \end{aligned} \quad (48)$$

对于 $k_{1,2,3}, \omega_{ref} = 0, v_{ref} = 0$, 稳定性通过 Lyapunov 函数进行检验：

$$V = \frac{1}{2} (x_e^2 + y_e^2) + \frac{(1 - \cos(\theta_e))}{k_2} \quad (49)$$

其中，负值的半正定时间导数 (negative semi-definite time derivative) 定义为：

$$\dot{V} = -k_1 x_e^2 - \frac{v_{ref} k_3 \sin^2(\theta_e)}{k_2} \quad (50)$$

局部分析表明，该控制定律具有局部指数稳定性。但是，由于系统是时域不变的， ω_{ref} 和 v_{ref} 需要指定为常数。

文献 [143] 中提出了一种相关的控制器，该控制器利用反推设计来为带有时变作为参照的有限域实现统一的局部指数稳定性。

5.2.2 输出反馈线性化 (Output feedback linearization)

对于较高的车速，适当限制转向角使其具有如式(8)中所述的连续运动。随着状态的增加，从简单的几何角度设计控制器变得更加困难。这种情况下，一个更好的办法是将系统的输出线性化。

使用前轮或后轮位置不容易做到这一点。文献 [144] 提出了一种简化反馈线性化的输出办法，其中选择了与转向角对齐的、在车辆前方距离 $d \neq 0$ 的任意一点。

令 $x_p = x_f + d \cos(\theta + \delta)$ 、 $y_p = y_f + d \sin(\theta + \delta)$ 为系统的输出，取这些输出的导数，将式(8)的动态系统项进行替换后，得到：

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\theta + \delta) - \frac{d}{l} \sin(\theta + \delta) \sin(\delta) & -d \sin(\theta + \delta) \\ \sin(\theta + \delta) + \frac{d}{l} \cos(\theta + \delta) \sin(\delta) & d \cos(\theta + \delta) \end{pmatrix}}_{A(\theta, \delta)} \begin{pmatrix} v_f \\ v_\delta \end{pmatrix} \quad (51)$$

然后，将式(51)的右侧用辅助控制变量 u_x 和 u_y 定义，得到：

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (52)$$

这使控制变得简单明了。通过使用式(51)中的逆矩阵来恢复 u_x 和 u_y ，现提供如下：

$$[A(\theta, \delta)]^{-1} = \left(-\frac{1}{d} \sin(\theta + \delta) - \frac{1}{l} \cos(\theta + \delta) \sin(\delta) \frac{1}{d} \cos(\theta + \delta) - \frac{1}{l} \sin(\theta + \delta) \sin(\delta) \right) \quad (53)$$

从输入输出线性系统中，可以通过控制来实现局部轨迹稳定：

$$\begin{aligned} u_x &= \dot{x}_{p,ref} + k_x (x_{p,ref} - x_p) \\ u_y &= \dot{y}_{p,ref} + k_y (y_{p,ref} - y_p) \end{aligned} \quad (54)$$

为避免混淆，注意在这种情况下，输出位置 (x_p, y_p) 和控制速度 v_f 与前面讨论的控制器不同。

5.3 可预测的控制方法

上面讨论的简单控制原则适用于中等驾驶条件。但是，在湿滑的道路或紧急制动情况下可能需要更精确的模型，例如第3.2节中介绍的模型。复杂的模型所附加的控制细节使控制设计变得异常复杂，难以根据构型空间的直觉理解和几何外形构造控制器。

模型预测控制 (Model Predictive Control, MPC)[145] 是一种通用的控制设计方法，对解决这一问题非常有效。从概念上讲，为解决短时间内的运动规划问题，该方法在产生的开环控制中采取间隔很短的时间，然后将其应用到系统中。在执行时，重新解决运动规划问题，以找到适合下一时间间隔的控制。计算硬件技术和数学编程算法的进步使预测控制在无人驾驶车辆中的实时应用成为可能。MPC 本身是一个主要的研究领域，本节仅对其技术进行简要描述，并对其在无人驾驶车辆控制中的应用进行综述。

由于模型预测控制是一种非常通用的控制技术，因此该模型采用具有控制 $u(t) \in \mathbb{R}^m$ 和状态 $x(t) \in \mathbb{R}^n$ 的通用连续时间控制系统的形式，

$$\dot{x} = f(x, u, t) \quad (55)$$

给出了一个可行的参考轨迹 $x_{ref}(t)$ ，并对一些运动规划器 $u_{ref}(t)$ 给出了满意的结果 (V.21)。然后通过适当的数值逼近选择离散化系统，从而在离散时间实例中给出 (V.21)。提供了可行的参考轨迹 $x_{ref}(t)$ ，并且对于某些运动计划器 $u_{ref}(t)$ 提供了满足要求 (V.21)。然后通过适当选择数值逼近来离散化系统，以便在 (V.21) 的离散时间实例中给出对于某些运动规划器 $u_{ref}(t)$ ，须提供了一个可行的参考轨迹 $x_{ref}(t)$ ，并满足式(55)。然后通过选择合适的数值逼近来离散化系统，在离散时间实例中式(55)可通过下式给出：

$$x_{k+1} = F_k(x_k, u_k), \quad k \in \mathbb{N} \quad (56)$$

最简单的离散化方案之一是对控制变量具有零阶保持的欧拉方法:

$$F_k(x(k \cdot \Delta t), u(k \cdot \Delta t)) = x(k \cdot \Delta t) + \Delta t \cdot f(x(k \cdot \Delta t), u(k \cdot \Delta t), t_k), \quad k \in \mathbb{N} \quad (57)$$

可通过在时间 $t_k = k \cdot \Delta t$ 上的近似将状态和控制进行离散化。离散系统的解是近似解，不会与连续时间下的方程完全匹配。类似地，在离散时间 t_k 采样的参考轨迹和控制也不会满足离散时间方程。例如，式(56)和式(57)的解的不匹配度为 $O(\Delta t)$ ，在时刻 t_k 的采样轨迹将产生：

$$F_k(x_{ref}(t_k), u_{ref}(t_k)) - x_{ref}(t_{k+1}) = O(\Delta t^2) \quad (58)$$

为避免以下讨论过于复杂，我们在本节的其余部分假设离散化的精度是足够的。控制准则通常采用以下形式进行描述：

$$\begin{aligned} u_k(x_{meas.}) = \arg \min_{\substack{x_n \in \mathcal{X}_n \\ u_n \in \mathcal{U}_n}} & \left\{ h(x_N - x_{ref,N}, u_N - u_{ref,N}) + \sum_{n=k}^{k+N-1} g_n(x_n - x_{ref,n}, u_n - u_{ref,n}) \right\} \\ \text{subject to} \\ x_k &= x_{meas} \\ x_{n+1} &= F(x_n, u_n) \\ n &\in \{k, \dots, k+N-1\} \end{aligned} \quad (59)$$

函数 g_n 惩罚每个时间步长内参考轨迹和控制的偏差，而函数 h 是时间结束时的最终惩罚。集合 \mathcal{X}_n 是一组允许状态，可限制不希望的位置或速度，例如过度的轮胎打滑或障碍物。集合 \mathcal{U}_n 对输入信号的幅度极限进行编码。现在最重要的考虑因素是式(59)右侧的解是否存在；以及当存在时，闭环系统的稳定性和鲁棒性如何。这些问题在关于预测控制的文献 [146][147] 进行了研究。

要在无人驾驶汽车上实现 MPC (V.25)，必须每秒解决好几次，这是使用 MPC 的主要障碍。在 h 和 g_n 是二次的， U_n 和 X_n 是多面体， F 是线性的特殊情况下，问题变成二次规划。不同于一般的非线性规划公式，内点算法可用于在多项式时间内求解二次规划。为了利用这一点，复杂的车辆模型经常被线性化以获得近似线性模型。线性化方法通常在计算线性化的参考值方面有所不同-当前操作点 (148) - (150)、参考路径 (151) 或更一般地，关于参考轨迹，这导致以下近似线性模型：

要在无人驾驶汽车上实现 MPC，须每秒时间内将方程(59)解好几次，这是使用 MPC 的主要障碍。在特殊情况下， h 和 g_n 是二次方程， \mathcal{U}_n 和 \mathcal{X}_n 是多面体，而 F 是一个线函数，此时问题转变成了二次规划。不同于一般的非线性规划公式，内点法可在多项式时间内求解二次规划。为利用这一点，通常对复杂的车辆模型进行线性化以获得近似线性模型。通常线性化方法在关于参照物如何进行线性化计算上有所不同——用当前操作点 [148]-[150]，参考路径 [151] 或者更一般地关于参考轨迹，这导出以下近似线性模型：

$$\begin{aligned} x_{ref,k+1} + \xi_{k+1} &= F_k(x_{ref,k}, u_{ref,k}) \\ &+ \underbrace{\nabla_x F_k(x_{ref,k}, u_{ref,k})}_{A_k} \xi_k + \underbrace{\nabla_u F_k(x_{ref,k}, u_{ref,k})}_{B_k} \eta_k \\ &+ O(\|\xi_k\|^2) + O(\|\eta_k\|^2) \end{aligned} \quad (60)$$

其中 $\xi := x - x_{ref}$ 和 $\eta := u - u_{ref}$ 为状态变量和控制变量与参考轨迹的偏差。摄动动力学

(perturbation dynamics) 的一阶扩展产生线性时变系统 (linear time varying (LTV) system):

$$\xi_{k+1} = A_k \xi_k + B_k \eta_k \quad (61)$$

然后采用二次函数作为目标函数，并代数方法对多面体进行约束，我们得到：

$$\begin{aligned} u_k(x_{meas.}) &= \underset{\xi_k, \eta_k}{\operatorname{argmin}} \left\{ \xi_N^T H \xi_N + \sum_{n=k}^{k+N-1} \xi_n^T Q_k \xi_k + \eta_k^T R_k \eta_k \right\} \\ &\text{subject to} \\ \xi_k &= x_{meas.} - x_{ref} \\ C_n \xi_n &\leq 0, \quad D_n \eta_n \leq 0 \\ \xi_{k+1} &= A_k \xi_k + B_n \eta_n, n \in \{k, \dots, k+N-1\} \end{aligned} \quad (62)$$

其中 R_k 和 Q_k 为正半定的。如果状态和输入不受约束，即 $\mathcal{U}_n = \mathbb{R}^m, X_n = \mathbb{R}^n$ ，则可以通过仅需进行 N 步矩阵递归计算的动态规划即可获得半封闭形式的解 [152]。当车辆模型用无状态和输入约束的受控自回归积分移动平均值 (controlled auto-regressive integrated moving average, CARIMA) 表示时，也对类似的封闭式递归求解方案进行了探索 [153][154]。

模型预测控制方法 (MPC) 的另一种变化是用性能函数中的罚函数来代替状态约束（例如障碍物）和输入约束。文献 [155] 提出了一种基于轮胎非线性动态模型的预测控制方法。不考虑障碍物情况下，除了惩罚到达目标点的参考路径的控制和偏离外，性能函数还惩罚在有限控制范围内违反输入约束和与障碍物发生碰撞的行为。在这个意义上，它类似于基于势场 (potential field) 的运动规划，但被证明具有改进的性能。

以下是汽车控制器相关的文献中模型预测控制框架 (model predictive control framework) 的一些变体：

5.3.1 具有运动学模型的无约束 MPC(Unconstrained MPC with Kinematic Models)

文献 [153] 中最早出现的预测控制器属于此类别，其中使用 CARIMA 模型在没有输入或状态约束的情况下应用模型预测控制框架。由此得到的半封闭式解有最小的计算要求，并且在文献 [154] 中获得采纳。此外，文献 [154] 使用线性运动模型考虑了无输入或状态约束的时变线性二次规划方法。

5.3.2 路径跟踪器 (Path Tracking Controllers)

文献 [151] 中，使用基于质心的线性动力学模型（假设为恒定速度）进行路径跟踪，和同时使用近似转向模型，研究了预测控制。通过详细的自动转向模型和 27 个自由度的 CarSim 车辆模型对所生成的集成模型进行了验证。

5.3.3 轨迹跟踪控制器 (Trajectory Tracking Controllers)

文献 [148] 研究了使用类似于第三节介绍的轮胎模型的预测控制器。仿真结果表明，采用完整的非线性预测控制策略，在 20Hz 的控制频率下，可以稳定地模拟冰冷条件下的紧急机动。然而，在控制范围只有两个时间步的情况下，计算时间是控制器采样时间的三倍，因此无法进行实验验证。在文献 [148]-[150] 中还基于线性化的方法对当前时间步的车辆状态进行了研究。求

解二次规划的复杂度降低到可接受的计算时间范围，在冰冻条件下以 21m/s 的速度实验成功并进行了报道。通过为时变系统的均匀局部渐近稳定性提供条件，文献 [150] 对这种方法的模拟结果和实验结果进一步改善。

5.4 线性变参控制器 (Linear Parameter Varying Controllers)

许多控制器设计技术可用于线性系统，使线性模型成为很多人向往的系统。然而，在正常驾驶条件下，可操作点的范围很广，因此很难依赖于一个单一的对操作点进行线性化表示的模型。为说明这一点，考虑式(38)中的横向误差动力学项。假设跟踪误差保持得较小，则绕操控点 $\theta_e = 0$ 和 $e = 0$ 进行动力学系统线性化会产生：

$$\begin{pmatrix} \dot{e} \\ \dot{\theta}_e \end{pmatrix} = \begin{pmatrix} 0 & v_r \\ 0 & 0 \end{pmatrix} \begin{pmatrix} e \\ \theta_e \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \omega + \begin{pmatrix} e \\ 0 \\ -v_r \kappa(s) \end{pmatrix} \quad (63)$$

引入新的控制变量，加入一个前馈 $u = \omega + v_r \kappa(s)$ ，可简化该讨论。现在动力学系统为：

$$\begin{pmatrix} \dot{e} \\ \dot{\theta}_e \end{pmatrix} = \begin{pmatrix} 0 & v_r \\ 0 & 0 \end{pmatrix} \begin{pmatrix} e \\ \theta_e \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \quad (64)$$

可以观察到该模型确实是线性的，但是前进速度 v_r 出现在线性模型中。一个简单的比例加微分控制，其增益分别为 k_p 和 k_d ，可以稳定系统的横向动力学，但闭环系统的极点由 $(-k_d \pm \sqrt{k_d^2 - 4k_p v_r})/2$ 给出。更高速度情况下，磁极进入复杂平面，从而发生振荡响应。相比之下，较小的 k_p 增益会导致低速时的响应不及时。解决该问题的一种非常直观且广泛使用的措施是对增益进行调节。在本例中，将 k_p 参数化为 v_r 的函数可对每个速度将极点 (poles) 固定为一个定值。该技术属于线性参数变化 (linear parameter varying, LPV) 模型的控制设计范畴 [156]。增益调度是设计这类控制器的经典方法。鲁棒控制和凸优化工具很容易用于处理更复杂的模型。

侧向控制的 LPV 控制设计见 [157]-[159]。在较低的自动化水平上，LPV 控制技术被提出用于集成系统控制。在这些设计中，几个子系统被组合在一个控制器下，以获得更好的操控性能。在 [162]、[163] 中开发了用于驱动主动和半主动悬架系统的 LPV 控制策略，而在 [164]、[165] 中开发了集成悬架和制动控制系统。

文献 [157]-[159] 中介绍了用于横向控制的 LPV 的控制设计技术。在文献 [160] 和 [161] 中，LPV 模型与预测控制方法一起用于路径稳定和轨迹稳定。在较低的自动化水平上，已经提出了用于集成系统控制的 LPV 控制技术。在这些设计中，几个子系统组合在一个控制器下，以提高处理性能。文献 [162] 和 [163] 开发了用于激活主动悬架系统和半主动悬架系统 (semi-active suspension systems) 的 LPV 控制策略；而文献 [164] 和 [165] 则开发了集成的悬架系统和制动控制系统。

6 结论

过去的三十年中，无人驾驶汽车技术发展日新月异。除了计算和感知硬件方面的进步外，移动机器人运动规划和反馈控制理论在计算方面的重大理论进展也促成了这一快速进展。毫无疑问，无人驾驶汽车将提高道路网络的利用率和安全性，从而推动了整个领域研究工作。

无人驾驶车辆是个复杂的系统，它被分解成一个层次的决策问题，其中一个问题的解是下一个问题的输入。对单个决策问题进行分解促使人们能够使用各个研究领域的成熟方法和技术。这之后的任务是集成这些方法，以便它们在语义上更为有效地交互，并且组合后的系统在计算上更有效。而更有效的运动规划算法可能仅与计算密集型的反馈控制器（例如，模型预测控制 MPC）兼容。相反，简单的控制法则可能需要执行较少的计算，但也不够鲁棒，而且需要使用更为详细的模型来进行运动规划。

本文综述了无人驾驶汽车决策问题的各个方面，重点是运动规划和反馈控制。对各种运动规划和控制技术的性能和计算需求进行的调研，可在系统设计时评估不同方法的兼容能力和计算能力提供一份参考。

【参考文献】

- [1] “2014 crash data key findings.” National Highway Traffic Safety Administration, Report No. DOT HS 812 219, 2014.
- [2] S. Singh, “Critical reasons for crashes investigated in the national motor vehicle crash causation survey.” National Highway Traffic Safety Administration, Report No. DOT HS 812 115, 2014.
- [3] B. McKenzie and M. Rapino, “Commuting in the United States: 2009, American Community Survey Reports,” *U.S. Census Bureau*, 2011. ACS-15, 2011.
- [4] D. A. Hennessy and D. L. Wiesenthal, “Traffic congestion, driver stress, and driver aggression,” *Aggressive behavior*, pp. 409–423, 1999.
- [5] E. D. Dickmanns and V. Graefe, “Dynamic monocular machine vision,” *Machine vision and applications*, vol. 1, pp. 223–240, 1988.
- [6] E. D. Dickmanns *et al.*, “Vehicles capable of dynamic vision,” in *IJCAI*, pp. 1577–1592, 1997.
- [7] “No Hands Across America Journal.” <http://www.cs.cmu.edu/afs/cs/usr/tjochem/www/nhaa/Journal.html>. Accessed: 2015-12-14.
- [8] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The great robot race*, vol. 36. Springer Science & Business Media, 2007.
- [9] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous vehicles in city traffic*, vol. 56. Springer, 2009.
- [10] J. Xin, C. Wang, Z. Zhang, and N. Zheng, “China future challenge: Beyond the intelligent vehicle,” *IEEE Intell. Transp. Syst. Soc. Newslett*, vol. 16, pp. 8–10, 2014.
- [11] P. Cerri, G. Soprani, P. Zani, J. Choi, J. Lee, D. Kim, K. Yi, and A. Broggi, “Computer vision at the Hyundai autonomous challenge,” in *International Conference on Intelligent Transportation Systems*, pp. 777–783, IEEE, 2011.
- [12] A. Broggi, P. Cerri, M. Felisa, M. C. Laghi, L. Mazzei, and P. P. Porta, “The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles,” *International Journal of Vehicle Autonomous Systems*, vol. 10, pp. 147–164, 2012.
- [13] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti, “Proud-public road urban driverless-car test,” *Transactions on Intelligent Transportation Systems*, vol. 16, pp. 3508–3519, 2015.
- [14] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, “Making Bertha drive—an autonomous journey on a historic route,” *Intelligent Transportation Systems Magazine*, vol. 6, pp. 8–20, 2014.
- [15] “Google Self-Driving Car Project.” <https://www.google.com/selfdrivingcar/>. Accessed: 2015-12-14.
- [16] “Tesla Motors: Model S Press Kit.” <https://www.teslamotors.com/presskit/autopilot>. Accessed: 2016-3-15.
- [17] S. O.-R. A. V. S. Committee *et al.*, “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems,” 2014.
- [18] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

- [19] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, pp. 431–438, 2001.
- [20] M. Bränström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *Transactions on Intelligent Transportation Systems*, vol. 11, pp. 658–669, 2010.
- [21] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *Transactions on Intelligent Transportation Systems*, vol. 4, pp. 143–153, 2003.
- [22] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *Transactions on Intelligent Transportation Systems*, vol. 14, pp. 1162–1175, 2013.
- [23] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pp. 145–154, ACM, 2012.
- [24] H. Kowshik, D. Caveney, and P. Kumar, "Provable systemwide safety in intelligent intersections," *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 804–818, 2011.
- [25] A. Eskandarian, *Handbook of intelligent vehicles*. Springer London, UK, 2012.
- [26] D. Miculescu and S. Karaman, "Polling-systems-based control of high-performance provably-safe autonomous intersections," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 1417–1423, IEEE, 2014.
- [27] F. Zhou, X. Li, and J. Ma, "Parsimonious shooting heuristic for trajectory control of connected automated traffic part I: Theoretical analysis with generalized time geography," *arXiv preprint arXiv:1511.04810*, 2015.
- [28] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *Transactions on Pattern Analysis & Machine Intelligence*, pp. 1239–1258, 2009.
- [29] G. Ros, A. Sappa, D. Ponsa, and A. M. Lopez, "Visual SLAM for driverless cars: A brief survey," in *Intelligent Vehicles Symposium (IV) Workshops*, 2012.
- [30] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 963–968, IEEE, 2011.
- [31] P. Liu, A. Kurt, K. Redmill, and U. Ozguner, "Classification of highway lane change behavior to detect dangerous cut-in maneuvers," in *The Transportation Research Board (TRB) 95th Annual Meeting*, 2015.
- [32] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, pp. 269–271, 1959.
- [33] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," tech. rep., DTIC Document, 1969.
- [34] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 156–165, Society for Industrial and Applied Mathematics, 2005.
- [35] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transportation Science*, vol. 46, pp. 388–404, 2012.
- [36] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," *arXiv preprint arXiv:1504.05140*, 2015.
- [37] F. Havlak and M. Campbell, "Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments," *Transactions on Robotics*, vol. 30, pp. 461–474, 2014.
- [38] Q. Tran and J. Firle, "Modelling of traffic situations at urban intersections with probabilistic non-parametric regression," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 334–339, IEEE, 2013.
- [39] A. C. Madrigal, "The trick that makes Google's self-driving cars work," *The Atlantic*, 2015. <http://www.theatlantic.com/technology/archive/2014/05/all-the-world-a-track-the-trick-that-makes-googles-self-driving-cars-work/370871/>.
- [40] R. Verma and D. D. Vecchio, "Semiautonomous multivehicle safety," *Robotics & Automation Magazine, IEEE*, vol. 18, pp. 44–54, 2011.
- [41] S. Z. Yong, M. Zhu, and E. Frazzoli, "Generalized innovation and inference algorithms for hidden mode switched linear stochastic systems with unknown inputs," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 3388–3394, IEEE, 2014.

- [42] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic MDP-behavior planning for cars,” in *14th International Conference on Intelligent Transportation Systems*, pp. 1537–1542, IEEE, 2011.
- [43] S. Ulbrich and M. Maurer, “Probabilistic online POMDP decision making for lane changes in fully automated driving,” in *16th International Conference on Intelligent Transportation Systems*, pp. 2063–2067, IEEE, 2013.
- [44] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs,” in *17th International Conference on Intelligent Transportation Systems*, pp. 392–399, IEEE, 2014.
- [45] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, “Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction,” in *Proceedings of Robotics: Science & Systems Conference*, p. 2, 2015.
- [46] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, “Intention-aware motion planning,” in *Algorithmic Foundations of Robotics X*, pp. 475–491, Springer, 2013.
- [47] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [48] A. De Luca, G. Oriolo, and C. Samson, “Feedback control of a nonholonomic car-like robot,” in *Robot motion planning and control*, pp. 171–253, Springer, 1998.
- [49] R. M. Murray and S. S. Sastry, “Nonholonomic motion planning: Steering using sinusoids,” *Transactions on Automatic Control*, vol. 38, pp. 700–716, 1993.
- [50] T. Fraichard and R. Mermond, “Path planning with uncertainty for car-like robots,” in *International Conference on Robotics and Automation*, vol. 1, pp. 27–32, IEEE, 1998.
- [51] M. Egerstedt, X. Hu, H. Rehbinder, and A. Stotsky, “Path planning and robust tracking for a car-like robot,” in *Proceedings of the 5th symposium on intelligent robotic systems*, pp. 237–243, Citeseer, 1997.
- [52] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” tech. rep., DTIC Document, 1992.
- [53] H. Goldstein, *Classical mechanics*. Pearson Education India, 1965.
- [54] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, “Real-time motion planning with applications to autonomous urban driving,” *Transactions on Control Systems Technology*, vol. 17, pp. 1105–1118, 2009.
- [55] M. D. Ventures, “Stanley: The robot that won the DARPA Grand Challenge,” *Journal of field Robotics*, vol. 23, pp. 661–692, 2006.
- [56] C. Samson, “Path following and time-varying feedback stabilization of a wheeled mobile robot,” in *2nd Int. Conf. on Automation, Robotics and Computer Vision*, 1992.
- [57] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.
- [58] J. Reeds and L. Shepp, “Optimal paths for a car that goes both forwards and backwards,” *Pacific journal of mathematics*, vol. 145, pp. 367–393, 1990.
- [59] E. Velenis and P. Tsotras, “Minimum time vs maximum exit velocity path optimization during cornering,” in *International symposium on industrial electronics*, pp. 355–360, 2005.
- [60] A. Rucco, G. Notarstefano, and J. Hauser, “Computing minimum lap-time trajectories for a single-track car with load transfer,” in *Conference on Decision and Control*, pp. 6321–6326, IEEE, 2012.
- [61] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsotras, and K. Iagnemma, “Optimal motion planning with the half-car dynamical model for autonomous high-speed driving,” in *American Control Conference*, pp. 188–193, IEEE, 2013.
- [62] S. C. Peters, E. Frazzoli, and K. Iagnemma, “Differential flatness of a front-steered vehicle with tire force control,” in *International Conference on Intelligent Robots and Systems (IROS)*, pp. 298–304, IEEE, 2011.
- [63] E. Bakker, L. Nyborg, and H. B. Pacejka, “Tyre modelling for use in vehicle dynamics studies,” tech. rep., SAE Technical Paper, 1987.
- [64] J. H. Reif, “Complexity of the mover’s problem and generalizations,” in *20th Annual Symposium on Foundations of Computer Science, SFCS ’79*, (Washington, DC, USA), pp. 421–427, IEEE, 1979.
- [65] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.
- [66] T. Lozano-Pérez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,”

Communications of the ACM, vol. 22, pp. 560–570, 1979.

- [67] J. A. Storer and J. H. Reif, “Shortest paths in the plane with polygonal obstacles,” *Journal of the ACM*, vol. 41, pp. 982–1012, 1994.
- [68] M. H. Overmars and E. Welzl, “New methods for computing visibility graphs,” in *Proceedings of the fourth annual symposium on Computational geometry*, pp. 164–171, ACM, 1988.
- [69] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational geometry*. Springer, 2000.
- [70] S. Lazard, J. Reif, and H. Wang, “The complexity of the two dimensional curvatureconstrained shortest-path problem,” in *Proceedings of the Third International Workshop on the Algorithmic Foundations of Robotics*, pp. 49–57, 1998.
- [71] S. Fortune and G. Wilfong, “Planning constrained motion,” *Annals of Mathematics and Artificial Intelligence*, vol. 3, pp. 21–82, 1991.
- [72] P. K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides, “Curvature-constrained shortest paths in a convex polygon,” *SIAM Journal on Computing*, vol. 31, pp. 1814–1851, 2002.
- [73] J.-D. Boissonnat and S. Lazard, “A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles,” in *Proceedings of the twelfth annual symposium on Computational geometry*, pp. 242–251, ACM, 1996.
- [74] F. Lamiraux, E. Ferre, and E. Vallee, “Kinodynamic motion planning: Connecting exploration trees using trajectory optimization methods,” in *International Conference on Robotics and Automation*, vol. 4, pp. 3987–3992, IEEE, 2004.
- [75] F. Boyer and F. Lamiraux, “Trajectory deformation applied to kinodynamic motion planning for a realistic car model,” in *International Conference on Robotics and Automation*, pp. 487–492, IEEE, 2006.
- [76] J. Canny and J. Reif, “New lower bound techniques for robot motion planning problems,” in *28th Symposium on Foundations of Computer Science*, pp. 49–60, IEEE, 1987.
- [77] D. P. Bertsekas, *Nonlinear programming*. Athena scientific, 1999.
- [78] M. L. Fredman and R. E. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms,” *Journal of the ACM (JACM)*, vol. 34, pp. 596–615, 1987.
- [79] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, “Analysis of probabilistic roadmaps for path planning,” *Transactions on Robotics and Automation*, vol. 14, pp. 166–171, 1998.
- [80] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.
- [81] E. Schmerling, L. Janson, and M. Pavone, “Optimal sampling-based motion planning under differential constraints: the driftless case,” *International Conference on Robotics and Automation*, 2015.
- [82] S. Karaman and E. Frazzoli, “Sampling-based optimal motion planning for non-holonomic dynamical systems,” in *International Conference on Robotics and Automation*, pp. 5041–5047, IEEE, 2013.
- [83] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *The International Journal of Robotics Research*, vol. 20, pp. 378–400, 2001.
- [84] Y. Li, Z. Littlefield, and K. E. Bekris, “Sparse methods for efficient asymptotically optimal kinodynamic planning,” in *Algorithmic Foundations of Robotics XI*, pp. 263–282, Springer, 2015.
- [85] J. Reif and M. Sharir, “Motion planning in the presence of moving obstacles,” *Journal of the ACM*, vol. 41, pp. 764–790, 1994.
- [86] T. Fraichard, “Trajectory planning in a dynamic workspace: a’state-time space’ approach,” *Advanced Robotics*, vol. 13, pp. 75–94, 1998.
- [87] J. Kasac, J. Deur, B. Novakovic, I. Kolmanovsky, and F. Assadian, “A conjugate gradient-based BPTT-like optimal control algorithm with vehicle dynamics control application,” *Transactions on Control Systems Technology*, vol. 19, pp. 1587–1595, 2011.
- [88] C. L. Darby, W. W. Hager, and A. V. Rao, “An hp-adaptive pseudospectral method for solving optimal control problems,” *Optimal Control Applications and Methods*, vol. 32, pp. 476–502, 2011.
- [89] L. S. Pontryagin, *Mathematical theory of optimal processes*. CRC Press, 1987.
- [90] Y. Tassa, N. Mansard, and E. Todorov, “Control-limited differential dynamic programming,” in *International Conference*

on Robotics and Automation, pp. 1168–1175, IEEE, 2014.

- [91] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 193–207, 1998.
- [92] E. Polak, “An historical survey of computational methods in optimal control,” *SIAM Review*, vol. 15, pp. 553–584, 1973.
- [93] B. Chazelle, “Approximation and decomposition of shapes,” *Advances in Robotics*, vol. 1, pp. 145–185, 1987.
- [94] C. Ó'Dúnlaing and C. K. Yap, “A "retraction" method for planning the motion of a disc,” *Journal of Algorithms*, vol. 6, pp. 104–111, 1985.
- [95] O. Takahashi and R. J. Schilling, “Motion planning in a plane using generalized voronoi diagrams,” *Transactions on Robotics and Automation*, vol. 5, pp. 143–150, 1989.
- [96] J.-C. Latombe, *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.
- [97] J. T. Schwartz and M. Sharir, “On the "piano movers" problem. ii. general techniques for computing topological properties of real algebraic manifolds,” *Advances in applied Mathematics*, vol. 4, pp. 298–351, 1983.
- [98] J. Backer and D. Kirkpatrick, “Finding curvature-constrained paths that avoid polygonal obstacles,” in *Proceedings of the twenty-third annual symposium on Computational geometry*, pp. 66–73, ACM, 2007.
- [99] P. Jacobs and J. Canny, “Planning smooth paths for mobile robots,” in *Nonholonomic Motion Planning*, pp. 271–342, Springer, 1993.
- [100] H. Wang and P. K. Agarwal, “Approximation algorithms for curvature-constrained shortest paths.,” in *SODA*, vol. 96, pp. 409–418, 1996.
- [101] S. M. La Valle, “Rapidly-exploring random trees a new tool for path planning,” tech. rep., Computer Science Dept., Iowa State University, 1998.
- [102] S. Petti and T. Fraichard, “Safe motion planning in dynamic environments,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2210–2215, IEEE, 2005.
- [103] A. Bhatia and E. Frazzoli, “Incremental search methods for reachability analysis of continuous and hybrid systems,” in *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004. Proceedings* (R. Alur and G. J. Pappas, eds.), pp. 142–156, Springer Berlin Heidelberg, 2004.
- [104] E. Glassman and R. Tedrake, “A quadratic regulator-based heuristic for rapidly exploring state space,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5021–5028, IEEE, 2010.
- [105] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, “Primitives for smoothing mobile robot trajectories,” *Transactions on Robotics and Automation*, vol. 11, pp. 441–448, 1995.
- [106] E. Velenis, P. Tsotras, and J. Lu, “Aggressive maneuvers on loose surfaces: Data analysis and input parametrization,” in *Mediterranean Conference on Control Automation*, pp. 1–6, 2007.
- [107] M. Pivtoraiko, R. A. Knepper, and A. Kelly, “Differentially constrained mobile robot motion planning in state lattices,” *Journal of Field Robotics*, vol. 26, pp. 308–333, 2009.
- [108] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, “On the relationship between classical grid search and probabilistic roadmaps,” *The International Journal of Robotics Research*, vol. 23, pp. 673–692, 2004.
- [109] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg, *Real-time robot motion planning using rasterizing computer graphics hardware*, vol. 24. ACM, 1990.
- [110] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1996.
- [111] L. Janson, E. Schmerling, A. Clark, and M. Pavone, “Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions,” *The International Journal of Robotics Research*, 2015.
- [112] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968.
- [113] I. Pohl, “First results on the effect of error in heuristic search,” *Machine Intelligence*, pp. 219–236, 1970.
- [114] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *International Conference Robotics and Automation*, pp. 3310–3317, IEEE, 1994.
- [115] S. Anthony, “The focussed D* algorithm for real-time replanning.,” in *IJCAI*, vol. 95, pp. 1652–1659, 1995.

- [116] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *Transactions on Robotics*, vol. 21, pp. 354–363, 2005.
- [117] E. A. Hansen and R. Zhou, “Anytime heuristic search.,” *J. Artif. Intell. Res.(JAIR)*, vol. 28, pp. 267–297, 2007.
- [118] M. Likhachev, G. J. Gordon, and S. Thrun, “ARA*: Anytime A* with provable bounds on sub-optimality,” in *Advances in Neural Information Processing Systems*, p. None, 2003.
- [119] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, “Anytime dynamic A*: An anytime, replanning algorithm.,” in *ICAPS*, pp. 262–271, 2005.
- [120] K. Daniel, A. Nash, S. Koenig, and A. Felner, “Theta*: Any-angle path planning on grids,” *Journal of Artificial Intelligence Research*, pp. 533–579, 2010.
- [121] A. Nash, S. Koenig, and C. Tovey, “Lazy theta*: Any-angle path planning and path length analysis in 3d,” in *Third Annual Symposium on Combinatorial Search*, 2010.
- [122] P. Yap, N. Burch, R. C. Holte, and J. Schaeffer, “Block a*: Database-driven search with applications in any-angle path-planning.,” in *AAAI*, 2011.
- [123] D. Ferguson and A. Stentz, “Using interpolation to improve path planning: The field d* algorithm,” *Journal of Field Robotics*, vol. 23, pp. 79–101, 2006.
- [124] D. Hsu, J.-C. Latombe, and R. Motwani, “Path planning in expansive configuration spaces,” in *International Conference on Robotics and Automation*, vol. 3, pp. 2719–2726, IEEE, 1997.
- [125] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, “Randomized kinodynamic motion planning with moving obstacles,” *The International Journal of Robotics Research*, vol. 21, pp. 233–255, 2002.
- [126] T. Kunz and M. Stilman, “Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete,” in *Algorithmic Foundations of Robotics XI*, pp. 233–244, Springer, 2015.
- [127] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in *Conference on Decision and Control*, pp. 7681–7687, IEEE, 2010.
- [128] A. Perez, R. Platt Jr, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, “LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics,” in *International Conference on Robotics and Automation*, pp. 2537–2542, IEEE, 2012.
- [129] D. J. Webb and J. van den Berg, “Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics,” in *International Conference on Robotics and Automation*, pp. 5054–5061, IEEE, 2013.
- [130] M. Otte and E. Frazzoli, “RRT-X: Real-time motion planning/replanning for environments with unpredictable obstacles,” in *International Workshop on the Algorithmic Foundations of Robotics*, 2014.
- [131] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al., “Autonomous driving in urban environments: Boss and the Urban Challenge,” *Journal of Field Robotics*, vol. 25, pp. 425–466, 2008.
- [132] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Practical search techniques in path planning for autonomous driving,” *Ann Arbor*, vol. 1001, p. 48105, 2008.
- [133] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, et al., “Odin: Team victorTango’s entry in the DARPA Urban Challenge,” *Journal of Field Robotics*, vol. 25, pp. 467–492, 2008.
- [134] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, et al., “A perception-driven autonomous urban vehicle,” *Journal of Field Robotics*, vol. 25, pp. 727–774, 2008.
- [135] J. P. Hespanha et al., “Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty,” *Transactions on Automatic Control*, vol. 52, pp. 1362–1379, 2007.
- [136] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, “Path-following for nonminimum phase systems removes performance limitations,” *Transactions on Automatic Control*, vol. 50, pp. 234–239, 2005.
- [137] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice hall New Jersey, 1996.
- [138] R. Wallace, A. Stentz, C. E. Thorpe, H. Maravec, W. Whittaker, and T. Kanade, “First results in robot road-following.,” in *IJCAI*, pp. 1089–1095, 1985.

- [139] O. Amidi and C. E. Thorpe, “Integrated mobile robot control,” in *Fibers’ 91, Boston, MA*, pp. 504–523, International Society for Optics and Photonics, 1991.
- [140] A. L. Rankin, C. D. Crane III, D. G. Armstrong II, A. D. Nease, and H. E. Brown, “Autonomous path-planning navigation system for site characterization,” in *Aerospace/Defense Sensing and Controls*, pp. 176–186, International Society for Optics and Photonics, 1996.
- [141] J. Wit, C. D. Crane, and D. Armstrong, “Autonomous ground vehicle path tracking,” *Journal of Robotic Systems*, vol. 21, pp. 439–449, 2004.
- [142] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *International Conference on Robotics and Automation*, pp. 384–389, IEEE, 1990.
- [143] Z.-P. Jiang and H. Nijmeijer, “Tracking control of mobile robots: a case study in backstepping,” *Automatica*, vol. 33, pp. 1393–1399, 1997.
- [144] B. d’Andréa Novel, G. Campion, and G. Bastin, “Control of nonholonomic wheeled mobile robots by state feedback linearization,” *The International journal of robotics research*, vol. 14, pp. 543–559, 1995.
- [145] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice-a survey,” *Automatica*, vol. 25, pp. 335–348, 1989.
- [146] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.
- [147] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, pp. 789–814, 2000.
- [148] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *Transactions on Control Systems Technology*, vol. 15, pp. 566–580, 2007.
- [149] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, “A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems,” in *46th Conference on Decision and Control*, pp. 2980–2985, IEEE, 2007.
- [150] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation,” *International journal of robust and nonlinear control*, vol. 18, pp. 862–875, 2008.
- [151] E. Kim, J. Kim, and M. Sunwoo, “Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics,” *International Journal of Automotive Technology*, vol. 15, pp. 1155–1164, 2014.
- [152] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2012.
- [153] A. Ollero and O. Amidi, “Predictive path tracking of mobile robots. application to the CMU Navlab,” in *5th International Conference on Advanced Robotics*, vol. 91, pp. 1081–1086, 1991.
- [154] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, “A predictive controller for autonomous vehicle path tracking,” *Transactions on Intelligent Transportation Systems*, vol. 10, pp. 92–102, 2009.
- [155] Y. Yoon, J. Shin, H. J. Kim, Y. Park, and S. Sastry, “Model-predictive active steering and obstacle avoidance for autonomous ground vehicles,” *Control Engineering Practice*, vol. 17, pp. 741–750, 2009.
- [156] O. Sename, P. Gaspar, and J. Bokor, *Robust control and linear parameter varying approaches: application to vehicle dynamics*, vol. 437. Springer, 2013.
- [157] P. Gáspár, Z. Szabó, and J. Bokor, “LPV design of fault-tolerant control for road vehicles,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, pp. 173–182, 2012.
- [158] J. Huang and M. Tomizuka, “LTV controller design for vehicle lateral control under fault in rear sensors,” *Transactions on Mechatronics*, vol. 10, pp. 1–7, 2005.
- [159] S. Ç. Baslamisli, İ. E. Köse, and G. Anlaş, “Gain-scheduled integrated active steering and differential control for vehicle handling improvement,” *Vehicle System Dynamics*, vol. 47, pp. 99–119, 2009.
- [160] T. Besselmann and M. Morari, “Autonomous vehicle steering using explicit LPV-MPC,” in *Control Conference (ECC), 2009 European*, pp. 2628–2633, IEEE, 2009.
- [161] T. Besselmann, P. Rostalski, and M. Morari, “Hybrid parameter-varying model predictive control for lateral vehicle stabilization,” in *European Control Conference*, pp. 1068–1075, IEEE, 2007.

- [162] P. Gáspár, Z. Szabó, and J. Bokor, “The design of an integrated control system in heavy vehicles based on an LPV method,” in *Conference on Decision and Control*, pp. 6722–6727, IEEE, 2005.
- [163] C. Poussot-Vassal, O. Sename, L. Dugard, P. Gaspar, Z. Szabo, and J. Bokor, “A new semi-active suspension control strategy through LPV technique,” *Control Engineering Practice*, vol. 16, pp. 1519–1534, 2008.
- [164] M. Doumiati, O. Sename, L. Dugard, J.-J. Martinez-Molina, P. Gaspar, and Z. Szabo, “Integrated vehicle dynamics control via coordination of active front steering and rear braking,” *European Journal of Control*, vol. 19, pp. 121–143, 2013.
- [165] P. Gaspar, Z. Szabo, J. Bokor, C. Poussot-Vassal, O. Sename, and L. Dugard, “Toward global chassis control by integrating the brake and suspension systems,” in *5th IFAC Symposium on Advances in Automotive Control, IFAC AAC 2007*, p. 6, IFAC, 2007.

DRAFT