Motion Planning for Automated Driving: Functional Architecture, Con-

cepts, and Algorithms

# 摘要

**问题（情况、动机和任务）:**

CC/PJ-FA Highway Pilot 和 CR/SP11-004 Urban Automated Driving 项目的任务是在复杂的城市和高速公路环境中实现高度自动化驾驶。运动规划子系统负责生成对感知到的动态障碍做出反应并考虑的安全、可行和舒适的轨迹。运动规划方法必须具有通用性，能够处理许多不同的交通场景，并且必须足够高效，以便快速重新规划以应对实际交通中突然的动态变化。

**结果：**

我们提出了一种运动规划方法，能够可靠地为高速公路和城市交通场景中的各种情况规划安全、可行和舒适的轨迹。运动规划器使用 Frenet 坐标系简化轨迹规划问题，并实现在不同车道之间的平滑转换。该 Frenet 坐标系是根据从已知地图中存储的稀疏中心线动态生成的平滑参考线定义的。所呈现的系统使用智能的末端条件采样和启发式任务预决策来减少可能轨迹的搜索空间，并确保高效的计算。在运动规划过程中考虑来自整个自动驾驶过程（例如感知、定位、预测、执行）的不确定性，通过在碰撞风险确定中使用碰撞概率来实现。还提出了一种针对具有高斯相对位置不确定性的定向边界框的极其高效的碰撞概率边界。

**结论和影响:**

我们已经开发了一个能够在结构化环境（如车道驾驶）中规划安全、可行和舒适轨迹的系统。尽管本报告中呈现的系统为高度自动化驾驶提供了坚实的基线运动规划系统，但在将这样的系统用于市场产品之前，还有许多需要探索和解决的挑战和主题。除了描述我们的运动规划方法外，我们还试图指定进一步工作的范围和研究主题，以提高系统的性能和效率。

# 1 介绍

高度自动化和完全自动化的驾驶系统需要在一个不可预测且动态变化的环境中安全地控制车辆，几乎不需要人类驾驶员的监督或干预。图1展示了适用于道路车辆的SAE自动化等级分类。SAE级别4和5对应于我们对高度和完全自动化驾驶的理解。由于高度自动化驾驶系统（SAE级别4）不能依赖人类驾驶员及时接管控制，因此级别4和5系统的安全性和性能要求非常相似。在这两种类型的系统中，自动化系统最终负责车辆的安全操作，直到达到预先确定的系统边界为止。虽然在级别4系统中系统边界可能更为严格，但在自主操作期间的责任水平是相同的。因此，在本报告中，我们将高度自动化和完全自动化的驾驶系统简称为‘自动驾驶’系统。此外，虽然这里介绍的运动规划技术可能适用于半自动驾驶系统（SAE级别2-3），但我们在本报告中的重点是适用于自动驾驶的算法和方法。

自动驾驶系统需要一个规划子系统，它能够进行战略层面的决策制定和响应性生成轨迹（时空曲线，完全定义了自动驾驶车辆未来一段时间内的状态）。然后，这些轨迹被发送到高频跟踪控制器，该控制器用于生成控制信号（加速度和转向速率/角度命令），最终由车辆执行。在一些简单的机器人系统或有限的驾驶辅助系统中，可能可以跳过轨迹生成过程，直接为车辆规划控制命令。然而，在高度自动化和完全自动化的驾驶系统中，我们假设运动规划问题过于复杂，无法以足够的速率运行以平稳跟踪曲线或调整自动驾驶车辆的速度曲线。相反，我们期望运动规划器以安全地对环境动态变化做出反应所需的速率运行（10 Hz），同时允许一个解耦的跟踪控制器以更快的速率运行（100 Hz），以适应平稳的执行和控制。这种战略性的解耦仅仅是规划系统内的几种层次分解中的一种，它允许不同的组件模块以不同的速率运行，根据它们的关键性和计算复杂性。在复杂的机器人系统（例如自动驾驶车辆）中，这种分解是必要的，因为复杂的规划是在高度动态和不确定的环境中进行的。

运动规划与控制之间的解耦允许我们定义自动驾驶系统中运动规划模块的角色。在CC/PJ-FA和CR/SP11-004项目的背景下，运动规划是根据更高级别的战略行动或行为规划系统提供的明确定义的任务描述生成最优、可行的轨迹。在这里，最优性是在启发式参数化成本函数的背景下定义的，旨在奖励舒适性、安全性和目标进展。类似地，运动规划器最终负责评估轨迹的可行性，以确保遵循运动学和动力学约束。这些约束确保轨迹在车辆的物理驾驶限制（执行限制、摩擦限制、非完整运动约束）、法律限制（例如速度限制）以及轨迹的碰撞风险低于指定的允许阈值范围内。

除了本报告中介绍的方法外，CC/EYM部门还在为SAE级别2/3系统开发运动规划概念。该方法的目的是提供更适合当前系列系统的辅助驾驶功能。如果感兴趣，建议读者联系CC/EYM以获取更多信息。

## 运动规划的概念

近年来，针对高度和完全自动驾驶车辆在结构化车道驾驶场景中的运动规划技术已经收敛到少数适合的方法。这些方法很适合自动车道驾驶问题的独特特征，包括：

动力学约束的重要性：即使在城市场景中，自动驾驶汽车也预计会以显著的速度运行，因此车辆的动力学约束（如有限加速度、颠簸和转向率）是规划问题的关键方面。

舒适性是首要考虑因素：与大多数机器人系统不同，自动驾驶汽车必须考虑并优先考虑乘客的舒适度。这极大地强调了减小纵向和横向颠簸的重要性。

需要平稳和一致性：自动驾驶汽车预期以平稳一致的方式行驶，这不仅是为了乘客的舒适和安心，还为了与其他车辆的安全、可预测的交互。运动规划方法本身是车辆外部人机界面的关键要素，一种一致、可预测的驾驶风格对于其他人类驾驶者准确预测自动驾驶车辆的未来运动是重要的。同样，不规律的驾驶可能会给其他驾驶者带来不必要的压力，并且可能导致在常规交互（如并线和在相邻车道行驶）中出现危险的模糊性。

高度结构化环境：车道驾驶任务中的固有结构减少了自动驾驶汽车的可能配置空间。这允许一个小的、定义良好的搜索区域，并且允许使用离散采样方法，这些方法不适用于通用机器人导航，因为它们计算可行性。

动态重新规划是必要的：自动驾驶汽车必须与其他具有未知意图的车辆并行行驶并协作。对其他交通参与者的意图和未来运动的不确定性，以及车辆感知系统的固有不确定性意味着给定的计划轨迹仅在非常短的时间内有效。自动驾驶汽车必须能够快速重新规划其计划的轨迹，以考虑动态环境中频繁的变化。

为了本报告的目的，我们在口语上使用“平滑”一词来指代视觉上平滑的路径，或者表示一条路径或轨迹会导致舒适的车辆运动。在这个语境中，平滑的轨迹不一定是一个无限可微的函数（C ∞），但通常至少表示连续的二阶导数（C 2）。对于空间曲线来说，C 2 连续性意味着曲率连续变化。布洛伊德曲线，或布洛伊德曲线序列，代表了我们所考虑的“平滑”类型中最简单的曲线。

一些最流行的通用机器人导航方法因为在上述一个或多个方面的不足而失宠。快速探索随机树（RRT）及其变体曾经被简要尝试过[9]，但导致了不规律的行为和不一致的运行时。此外，它们的优势在于高维空间的高效探索，而这在结构化车道驾驶问题中并不那么有用。

类似地，基于纯图搜索的方法（如A\*、D\*、D\*-lite）通过一组运动基元也被证明是可行的[12, 16]，但在确保一致的运行时方面存在困难，并且未能充分利用车道结构。基于专用状态格的这些方法的变体已被证明是成功的，并在下文中讨论。

需要注意的是，RRT和图搜索在运动基元库上的应用都是可行的方法，用于停车相关任务。本报告中的讨论和结论涉及标准车道驾驶任务，包括高速公路和城市使用情况。

近期自动驾驶文献中出现了三种主要的运动规划技术。

空间-时间状态格方法：状态格方法生成了一个预先计算的在弗雷内框架中的网格或“格子”，允许使用经典的基于图的搜索技术，如A\* [20, 21]。这些方法创建了平滑的轨迹，并大大简化了搜索空间，但需要按照预先确定的网格行驶可能会导致次优性和平滑性问题。由于轨迹必须在每个格点处的高阶导数上平滑匹配，所以离散搜索空间很快变得难以处理，除非对节点处的高阶导数进行显著的假设，例如将高阶导数在节点处置零。这些必要的近似可能导致不舒适或意外的驾驶行为。

基于连续优化的方法：基于优化的方法依赖于标准的凸优化技术，如序贯二次规划或内点方法，直接优化初始轨迹 [11, 23]。这些方法可能是最灵活的，特别擅长于半结构化车道驾驶场景，如适应停放的汽车。然而，它们在构建凸且良好行为的成本函数方面需要极大的注意，并且在有限的规划周期内可靠地找到有效的轨迹是困难的。这种方法也被广泛应用于经典的机器人运动规划任务 [14]。

基于离散采样的方法：离散采样方法涉及直接在自动驾驶车辆的所需搜索空间中对轨迹进行采样 [6, 19]。这些方法利用了车道驾驶任务的受限结构，允许对可能轨迹的一组明确枚举，并且具有搜索空间的密集覆盖。这种方法的优点包括几乎确定性的运行时和相对简单的实现。这种方法也可以高度并行化。

由于需要保持一致的周期时间，同时最大程度地提高乘客舒适度和安全性，在我们的系统中采用了连续优化和离散采样的混合规划方法。基于[2, 3]的基于优化的方法被用于在线生成平滑的车道中心线。然后，这些平滑的中心线被用作参考线，来定义一个良好行为的弗雷内框架，供基于离散采样的规划方法使用。

## 1.2 空间曲线表示

任何运动规划系统的一个关键考虑因素是如何在空间和时间中表示平滑曲线。这些表示可以从一个简单的离散点序列，如[23]所示，到基于空间或时间样条、Bézier曲线或螺旋线的完整参数曲线定义。鼓励读者参考[10]，了解机械绘图、道路和铁路设计以及现代计算机图形学中空间曲线的历史和用法。我们分享[10]的结论，即布洛伊德的高阶概括提供了一种特别简洁、稀疏和有效的空间曲线表示，具有高阶连续性。这些布洛伊德的高阶概括是一类称为螺旋线的曲线。布洛伊德本身是螺旋线的特殊情况，称为欧拉或Cornu螺旋。螺旋线一般采用多项式形式定义方向，θ，作为路径长度s的函数。 θ(s) = a\_n \* s^n + a\_(n-1) \* s^(n-1) + a\_(n-2) \* s^(n-2) + ... + a\_0

这种表示对于定义空间路径特别简洁，因为它通过一个单一函数完全定义了路径的形状（与需要同时定义x(t)和y(t)函数相比），并且它避免了基于时间的笛卡尔曲线中方向计算中固有的奇异性（当ẋ(t) = ẏ(t) = 0时，方向信息丢失）。在这种表示中，布洛伊德是一个简单的二阶螺旋线，而布洛伊德样条提供了C 2 连续性。 利用方程1中的螺旋线定义，沿着曲线的曲率简单地是第一导数。 κ(s) = dθ(s) / ds = n \* a\_n \* s^(n-1) + (n - 1) \* a\_(n-1) \* s^(n-2) + ... + a\_1

这意味着布洛伊德螺旋线沿其长度具有线性变化的曲率（这当然是布洛伊德的经典定义）。不幸的是，布洛伊德并不灵活，很难将一系列布洛伊德螺旋线拟合到任意曲线上。 对于自动驾驶来说，更好的制定是使用三阶螺旋线。 θ(s) = as^3 + bs^2 + cs + d

可以通过按顺序组合螺旋线段来形成复杂的空间曲线，以创建螺旋样条，并在节点点匹配θ和κ。在我们的系统中，这样的样条既是本地Frenet框架的基础，也是最终轨迹表示的空间组件。需要注意的是，与布洛伊德一样，三阶螺旋线也不能匹配任意的端点条件（在s、θ、κ中）。然而，如第3节所讨论的，我们系统中使用的路径优化能够共同优化三阶螺旋线的样条，以紧密适应任何所需的形状或形式。需要曲线能够适应任意端点条件（在s、θ、κ中）的运动规划方法，例如第1.1节讨论的空间-时间状态格方法，必须使用五阶螺旋线以获得足够的灵活性。

### 1.2.1 评估螺旋样条

使用螺旋的主要挑战在于，在曲线上的某个位置评估笛卡尔坐标没有闭式解。沿着路径的给定点的笛卡尔坐标使用以下公式计算。 x(s) = ∫(0到s) cos(θ(τ ))dτ y(s) = ∫(0到s) sin(θ(τ ))dτ

幸运的是，高效的数值积分方法使得评估这些公式成为一个较小的问题。我们使用高斯-勒让德积分，如[2, 3]中建议的那样。此外，我们使用三次Hermite插值来评估沿着三阶θ(s)多项式的每个点，这样我们就可以仅使用其端点条件（s₁，θ₁，κ₁，s₂，θ₂，κ₂）作为参数来评估多项式。使用这些技术，沿着螺旋样条的任何点的笛卡尔坐标都可以有效地计算为相对于（s、θ、κ）节点点的线性项的和。在第3节讨论的参考线优化过程中，利用这个计算来确保相对于路径上的笛卡尔点的凸成本项也相对于优化变量（节点参数）是凸的。

这种表示完全定义了复杂的空间曲线，并且非常适合作为本地Frenet框架的基础。

我们轨迹规划系统的输出是一个2D轨迹，其形式与图2所示的空间曲线类似。为了定义一个完整的空间-时间轨迹，我们将独立变量s替换为时间t，并在每个节点附加纵向速度和加速度信息。这允许沿着路径的距离也被定义为一个立方样条，可以使用立方Hermite插值来评估，给定其端点条件（t₁，v₁，a₁，t₂，v₂，a₂）。使用这种轨迹定义，可以像图3所示那样进行插值，其中计算两个括号节点之间的临时∆s值，以执行空间插值。

# 2 功能架构

图4显示了我们自动驾驶系统的规划系统。规划系统接收来自感知和定位模块的动态输入。规划系统还接收两个静态输入，这里没有显示出来：环境的已知地图，包括几何和语义元素，以及通过道路网络的选择路线。该地图在我们的系统中被称为分层地图，详细信息可以在[13]中找到。计划的路线是在启动时计算的，并以一种限制性的、无环的车道图的形式提供给行为规划器模块，称为战术搜索空间（TSS）。关于TSS的其他细节也可以在[13]中找到。感知模块接收来自车辆传感器套件的输入，负责检测和暂时跟踪静态障碍物，以及提供传感器覆盖和遮挡信息的可见性网格。未来，这些传感器信息很可能会扩展，包括检测到的交通元素，如交通灯状态和路标。定位模块提供了相对于已知地图的自车当前状态的估计，包括完整的状态协方差。与感知模块类似，定位模块为车辆的特定定位方法和硬件提供了一个抽象接口。规划系统的所有输入（除了TSS）都被馈送到环境建模模块。环境建模模块负责将感知模块检测到的对象与已知地图相关联。环境建模模块还推理感知模块提供的报告的可见性信息，并在地图上生成可见或已知区域的语义表示。环境建模模块产生了一个一致的内部语义表示，称为环境模型，该模型描述了自车的局部环境，包括地图、障碍物和可见性信息，以及它们的语义关系。在环境建模模块之后，规划过程按照层次结构进行。行为规划器模块接收作为TSS定义的路线以及当前环境模型，并与预测模块迭代，探索自车可能行为动作的树。行为规划过程的结果是一组候选行为和相关行为成本。然后，在并行的运动规划实例中评估每个候选行为，为该行为生成最佳轨迹以及运动规划成本和可行性判定。然后，将每个候选行为的行为成本、运动规划成本和运动规划器的可行性信息发送到决策模块，决策模块选择最佳行动。与所选行为相关联的最佳轨迹然后发送到控制和执行系统。关于环境建模、预测和行为规划模块的功能和交互的更多细节可以在[13]中找到。运动规划模块及其相关接口是本报告的主题，将在本节和后续节中详细介绍。

## 2.1 行为规划与运动规划的耦合

规划系统的分层分解成行为规划和运动规划模块是一种旨在解决集成任务和运动规划的常见机器人学挑战的方法。在理想情况下，行为规划模块在其自我行为搜索的每个节点扩展时将简单地查询完整的运动规划器，而产生的行为序列和计划的轨迹将是一个最优解，并且可以直接由控制器使用。不幸的是，即使对于我们高度优化的运动规划系统，一次对完整运动规划器的调用也可能占据整个规划周期。为了克服这些计算限制，我们将规划问题分解为任务或行动规划阶段（行为规划器），该阶段使用高效的近似运动规划来识别良好的长期策略，以及一个负责详细评估成本和可行性的运动规划系统。 在我们的系统中，行为规划模块使用运动规划模块的简化版本来估计给定自我动作的标准自我轨迹。这些简化包括仅基于标准期望的最终条件生成单个轨迹（与完整运动规划器中探索可行驾驶包络的一束轨迹相反），并仅在1D车道空间中执行近似碰撞检查。由于行为规划模块仅使用近似运动规划估计，因此无法检查不同自我动作的真实可行性。因此，行为规划器旨在基于它们的战略效用（即它们如何有效地定位自我车辆以实现其未来目标）来识别k个最佳初始动作。然后，运动规划模块负责为这k个选项中的每一个评估最佳轨迹。最终的决策模块根据行为规划模块的战略效用成本、运动规划模块的舒适性和安全成本，以及运动规划模块还从运动规划模块中确定的可行性和碰撞风险确定选择最佳动作。 由于单个运动规划调用可能占据整个规划周期，因此最佳的k个自我动作通过并行运行k个运动规划器实例来评估。这使我们能够在运动规划器中保持固定的高频率规划周期时间，同时评估多个可能的动作和回退动作。由于运动规划模块是安全性的关键组件，这种安排还允许我们异步运行行为规划模块，在非固定的周期时间内允许行为搜索大小的非确定性。关于规划系统异步架构的更详细讨论将在第2.3节中提供。

### 2.1.1 行为规划器和运动规划器接口描述

行为规划模块为每个运动规划器的k个实例提供了足以完全定义运动规划问题的任务描述。该任务描述称为行为动作，包括行为类型（如跟随车道或变道）和行为目标。

行为目标包括目标车道序列、跟随目标、速度目标、停止目标以及（在变道的情况下）起始车道序列。跟随目标基于目标车道和起始车道中的相关前车和后车。速度目标基于目标车道的速度限制，再加上由即将到来的曲率引起的横向加速度约束。停止目标包括下一个指定的停止点（如果有的话）。目标和起始车道序列是车道描述，指定了在分层地图中自车应该行驶的车道段序列。

图5显示了跟随目标的示例，分别用于跟随车道和变道行为。对于图5a中的跟随车道行为，目标车道只是自车行驶的车道，跟随目标仅包括当前的前车。对于图5b中的变道行为，目标车道是变道的目标车道，跟随目标包括目标车道和起始车道中的前车和后车，以及起始车道中的前车。有关行为动作和各种行为类型的其他信息，请参阅[13]。

## 2.2 运动规划架构

运动规划模块负责生成执行由行为规划模块指定的行为动作的成本最优、可行且无碰撞的轨迹。图6显示了运动规划模块的结构。在执行轨迹规划之前，运动规划器将给定行为动作中的目标车道序列与参考线生成器预先计算的平滑参考线进行匹配。然后，针对由所选参考线定义的新Frenet框架，重新计算跟随、速度和停止目标。指定的自车行为动作还用于执行预测过程，以生成轨迹规划过程中用于碰撞检查的未来动态障碍物状态。匹配的参考线和预测的动态障碍物状态被用作轨迹规划模块的输入。对于运动规划模块的每个实例，其结果是如第1.2.1节所述的最优的空间-时间轨迹，以及由舒适性、安全性和目标实现成本组成的运动规划成本，以及最优轨迹的可行性和碰撞风险评估。

参考线生成器为自车可能选择在有限距离视野内行驶的每个可能的车道段序列提供即时预先计算的平滑参考线。这些参考线在未覆盖的车道序列进入参考线生成视野范围内或自车距离先前计算的参考线末端低于触发距离时重新计算。这种重新计算的速度比规划系统的其他部分慢得多，可以被认为更多地是一种即时预先计算而不是在线计算。计算出的参考线被存储在本地，并且每个运动规划器实例都查询参考线生成模块以请求与其目标车道序列匹配的参考线。有关参考线生成过程的更多详细信息，请参见第3节。

轨迹规划模块在其轨迹搜索过程中需要一组预测的动态障碍物状态来进行碰撞检查。由于行为规划模块中使用的预测假设了一系列自车行为，而运动规划模块一次只能规划单个行为，因此这些预测无法直接使用。为了生成仍然捕捉到其他交通参与者对单个自车行为的反应和互动的预测，对当前行为动作进行了一种非分支展开。这种非分支展开实际上相当于重新运行行为规划搜索，同时限制自车只传播其当前行为。这意味着在跟随车道行为中的自车将在整个预测范围内保持跟随车道行为。类似地，处于变道行为中的自车将完成变道，然后在预测范围的剩余时间内切换到目标车道中的跟随车道行为。这种非分支预测提供了一种迭代预测，其中动态障碍物对规划的自车行为和彼此的反应是正常的。

轨迹规划在由匹配的参考线定义的Frenet框架中进行。轨迹是使用由行为动作提供的目标信息生成的，并且使用来自非分支预测步骤的预测动态障碍物状态进行碰撞检查。最终得到的最优轨迹，以及其成本、可行性和碰撞风险，随后被发送到决策模块。

## 2.3 规划系统时序

规划系统被设计为以固定的10 Hz速率运行。这是为了提供足够的反应时间来应对自车环境中的动态变化，并实现与其他交通参与者的无滞后交互。为了实现这一点，规划系统的许多非关键组件是异步运行的，而安全关键的运动规划组件以所需的速率运行，并使用异步组件的最新输出。这种异步架构允许环境建模模块以比主要规划循环更快的速率运行，从而可以为每个感知输入生成一个新的环境模型，从而在环境建模周期之间实现更小的更新。这有助于改善环境建模模块内的语义数据关联性能。规划系统的异步架构还允许行为规划和参考线生成模块以比主要规划循环更慢的频率运行，为困难但不是很紧迫的计算提供了额外的时间。

异步参考线计算几乎没有副作用，因为只要覆盖了所需的车道间隔，预先计算的参考线就永远不会“过时”。另一方面，从行为规划模块异步计算的结果必须根据最新的环境模型进行仔细更新，以确保轨迹规划模块使用的行为动作与车辆周围当前的交通场景保持一致。由于嘈杂、不确定的感知系统的限制，动态障碍物在感知步骤之间可能会出现、消失或分裂和合并。这意味着先前计算的行为动作的跟随目标必须根据其当前的行为类型更新，以重新获取前方或后方的目标车辆。类似地，与非分支预测一样，行为类型在完成变道操作后可能需要从“变道”转换为“跟随车道”。

理论上，行为规划模块可能需要执行任意长的时间，这意味着在最坏的情况下，运动规划模块必须能够在无限长的时间内处理最后收到的行为动作。尽管行为规划模块计算了一系列行为动作，但由于运动规划模块无法根据当前环境的更新独立地重新评估序列中的后续动作，因此只考虑了来自行为规划模块的第一个行为动作。实际上，行为规划器被设计为以1 - 5 Hz的相当可靠的频率执行，而这种程度上的解耦行为并不是真正需要的。

## 2.4 轨迹规划与运动控制的耦合

如图4所示，一旦轨迹由运动规划模块规划并由决策模块选择，车辆运动控制器的任务就是控制执行器，以紧密地跟随规划的轨迹。

该系统的闭环反馈模型如图9所示。控制器使用规划的轨迹作为参考输入，并计算车辆控制所需的数值。从规划的轨迹到控制器的主要输入是当前时刻自车的期望状态。为了保持规划器与车辆的独立性，复杂的车辆动态模型计算留给了控制器，规划仅基于满足基本的动力学和非完整运动约束。这也有助于分别调整规划器和控制器，以适应决策和性能相关的能力。为了控制器的平滑运行，持续的输入在期望位置及其时间和空间导数方面至关重要。有关运动规划器如何确保规划轨迹之间的连续性的更多详细信息在第4.2节中介绍。

当控制器无法跟踪期望的轨迹时，基于期望位置而不是实际位置继续规划新的轨迹最终会变得不安全。尽管这些情况应该几乎不会发生在正确调整的运动规划器和控制器上，但一旦发生，我们将重新初始化运动规划器到当前车辆的位置。由于控制器中存在积分组成部分，在重新初始化时，集成误差将被设为零，以避免不舒服的瞬态行为。这种重新初始化更新在本报告中称为 Re-Init，并且随着规划的轨迹一起向控制器发送一个标志，以通知控制器发生了这样的更新。

下面提供了从运动规划器发送到控制器的标志的完整描述。

* **OK：** 该标志用作有效性检查，以确保规划器已成功完成一个循环。
* **RE-INIT：** 该标志用于通知控制器，规划器从当前状态重新初始化，以适当地处理控制器中的积分误差项。
* **STITCHED：** 该标志通知控制器，规划的轨迹与先前的轨迹平滑连接。
* **SYNTHETIC：** 当向控制器发送预定义的轨迹时，会发送此标志。此类轨迹用作调试工具，用于测试控制器的性能。
* **DROPOUT：** 该标志表明规划器无法计算可行轨迹，并指示自动驾驶系统应请求驾驶员交接。
* **DEADZONE STOP：** 该标志用于静止管理。
* **EMERGENCY TRAJECTORY：** 该标志用于通知控制器选择了紧急轨迹。目前，这不会影响控制行为，但在这种情况下可以用于将控制参数集从舒适值切换到动态值。

这些标志并不是互斥的，规划器可以使用多个标志来描述其状态。例如，一个正常成功规划的轨迹可能同时包括 OK 和 STITCHED 两个标志。

在从规划器接收到所需的轨迹以及标志信息后，控制器就有足够的信息来规划车辆的控制策略并生成相应的执行器输入。在我们的情况下，我们采用了一种单一的控制概念，它使用期望的车辆状态来计算期望的转向角和目标加速度。然后将这些值传递给相应的执行器来控制车辆。

控制本身被分解为单独的纵向和横向控制器。在图 9 中显示的模块图中，黄色模块属于横向控制器，绿色模块属于纵向控制器，而青色模块用于纵向和横向控制器。纵向控制器由一个正常的 PI 反馈控制组成，而横向控制器则由两种方法组成，分别处理高速和低速情况。在高速情况下，这两种横向控制方法都是必要的，以处理高速时的非线性，并在低速时保持精确的控制性能。高速控制器的概念基于[7]中解释的非线性车辆动力学模型。

两种横向控制情况之间的转换由软过渡范围定义。在此过渡范围中的中间速度值，控制输出将作为两种横向控制策略的组合。

在低速区域由于传感器噪声较高，静止管理一直是一个棘手的问题。如果没有静止管理，低速区域的车辆状态和前方车辆状态中的传感器噪声可能会导致间歇性的爬行行为和其他不良效果。这种静止情况由控制器专门处理。我们处理静止管理的方法在第 5.3 节中描述。

# 3 参考线

运动规划器在 Frenet 坐标系中操作，这意味着所有轨迹都是相对于一个固定的参考线生成和定义的。获得一个具有高阶连续性的平滑参考线对于使运动规划器能够产生平滑、连续的轨迹至关重要。在 Frenet 坐标系中规划允许我们大大简化运动规划问题的表述，并且能够轻松地泛化到各种道路几何形状。然而，在 Frenet 坐标系中规划时，参考线及其空间导数中的任何急转弯或其他不连续性都会被运动规划器生成的每条轨迹自动继承。

图 10 展示了车道段的几何描述，这是我们内部分层地图数据结构中的基本元素。在我们的系统中，我们假设外部提供的地图可转换为这种分层地图格式，并且提供了每个车道段中心线的粗略近似，以多边形线段（即 (x, y, z) 点的序列）的形式。这些点的间距可以任意，不包括方向、曲率或其他状态信息。这种表述允许外部地图提供者提供简单、健壮且内存高效的地图规范。每个车道段的左右边界也提供了多边形线段的描述，从而使我们能够定义覆盖车道段区域的多边形。

车道中心线的多边形线段描述不足以定义我们的运动规划器的 Frenet 坐标系。我们也无法保证提供的中心线实际上位于车道中心，或者在急转弯和交叉口的情况下采取合理的形状。为了解决这些问题，我们执行非线性优化过程来将平滑曲线拟合到车道几何形状上，从而得到平滑连续的参考线。

## 3.1 参考线生成架构

图 11 描述了我们运动规划系统中参考线是如何生成的。由于参考线拟合过程仅基于静态地图进行，因此无需与规划器具有相同的频率重新计算参考线。从理论上讲，可以为给定地图完全离线地预先计算参考线，并存储供运动规划器使用。这种方法的挑战在于，平滑曲线的局部性差，这意味着与多边形线段不同，曲线的一个部分的小变化可能会影响整个曲线的形状。这种局部性问题使得在分叉道路网络的情况下，离线优化问题变得更加困难。

图 12 显示了整个地图的离线联合优化和在线参考线预计算之间的区别。图 12a 显示了最简单的分叉场景，其中一个车道段在分叉处结束，车辆可以继续前往两个重叠的后续车道段之一。为了创建适用于两个后续车道段的平滑连续参考线，离线解决方案需要执行联合优化，以修改分支点以平滑适应两个选项。图 12c 显示了此离线联合优化案例。将这种方法推广到一个更大的地图，比如整个城市地图，将需要对整个车道图进行联合优化。这种优化可能是可行的，但与优化单个线性车道段的参考线相比，它是一个更加困难的命题。相反，如果我们仅在有限的视野内计算车辆当前位置附近的参考线，我们可以简单地计算一组重叠的线性参考线，这些线性参考线完全覆盖了自动车辆可能采取的所有可能路径，如图 12b 所示。

这种在线、有限范围的参考线生成只需要在自动驾驶车辆接近先前计算的参考线集的末尾时才需要重复进行，这意味着这种优化可以以比运动规划器本身更慢的频率进行，从而实现准离线的优化条件。每次刷新时都会从头重新计算参考线。由于通常情况下，两条重叠的参考线只有在它们共享固定的结束条件时才会匹配，因此这种完全的重新计算是必要的，这是由于局部性问题引起的。

## 3.2 任务枚举

为了确定需要生成哪些参考线，我们希望枚举所有参考线，直到指定的前瞻视距。图13显示了由分层地图提供的车道段和车道连接的图形。突出显示的区域显示了受限车道（即战术搜索空间或TSS），在该空间中自动驾驶车辆应该行驶。从当前车辆位置开始，我们考虑到指定前瞻视距的所有连续车道序列。此外，我们还考虑可以通过车道变更到达的车道序列（在图中由双向箭头表示）。只考虑连续的车道序列，这意味着枚举的车道序列从不包括车道变更连接。 总之，搜索获取TSS的局部子图，并从该子图的根节点开始枚举车道序列。 一旦构建了可能的车道序列集，就会将其与已经计算了参考线或正在计算新参考线的参考线任务集进行比较。当某个车道序列不受所有现有参考线任务充分覆盖时，就会启动新的参考线计算。

## 3.3 参考线优化

一旦触发新的参考线任务，就会启动异步的参考线优化计算，以找到最佳且平滑的参考线。这里介绍的参考线优化的形式和结构受到[2, 3]的重大影响。该优化的输入是一个折线（即一系列(x, y)点），通过将每个期望车道序列的地图中心线附加起来构造而成。然后，该折线以固定的距离分辨率（通常为1米）进行重新采样，并使用每个点的近似方向和曲率信息生成近似的初始解决方案，基于在第1.2.1节介绍的螺旋样条空间曲线表示。目前使用IPOPT优化库[17]执行优化。

图14展示了优化成本函数中的一些项和约束。优化成本函数包括以下项：

* 基于与车道中心线的标准化距离的走廊成本，如图14的横截面视图所示。
* 基于沿螺旋样条的曲率二次导数的积分的平滑度成本。

此外，优化使用以下约束项：

* 走廊约束，以确保螺旋样条的节点保持在车道边界内，如图14的横截面视图所示。
* 位置约束，以确保每个积分的螺旋段在空间上连接。
* 节点间距约束，以确保螺旋样条上的节点等距分布，如图14所示。限制间距至关重要，因为走廊约束仅应用于节点。

## 3.4 挑战

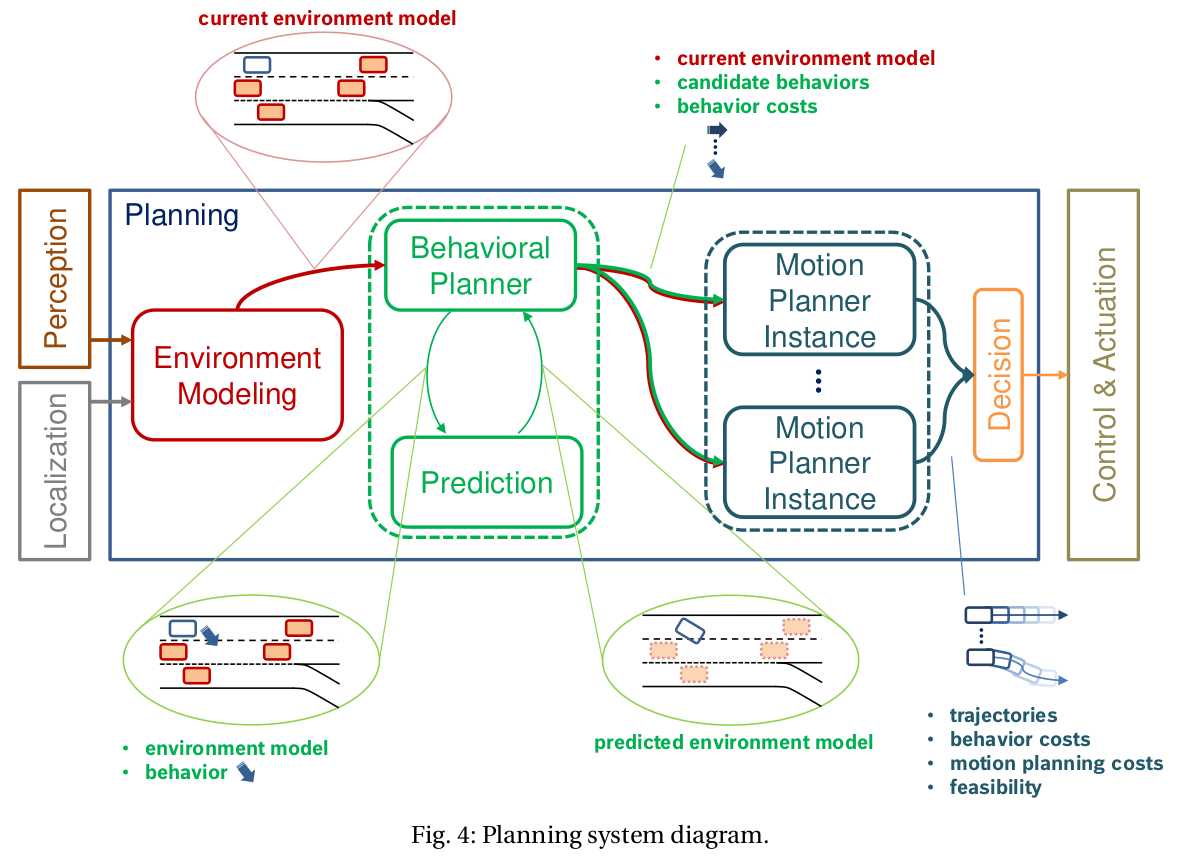
当前参考线生成过程的一个主要挑战是我们使用于参考线优化的IPOPT库支持多线程的能力较差。因此，目前我们每个进程只能运行一个参考线优化实例。可能有更适合此目的的内点或顺序二次规划库可供选择。

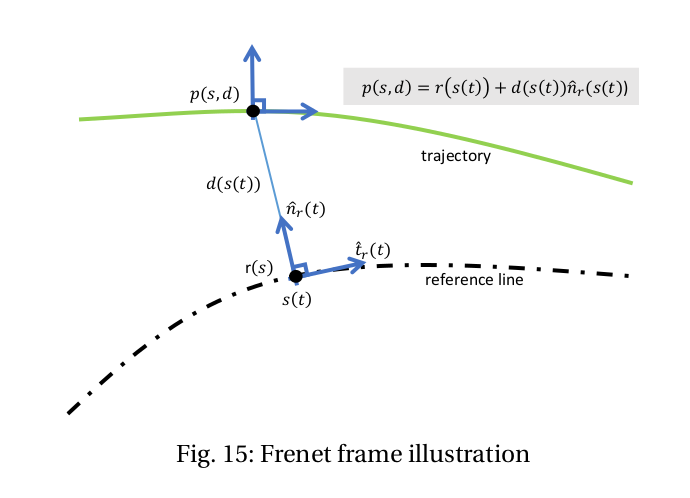
提高参考线生成过程性能的另一个可能途径是在智能选择的固定节点处将参考线拼接在一起。固定节点的明显选择是曲率可以被限制为零的长直道路段。有效的拼接将允许分段优化，从而实现对更长参考线的更高效的在线优化，或者可能更易处理的离线优化整个车道图的方法。

除了分段优化之外，还有可能在成本函数设计和参考线优化的雅可比矩阵/海森矩阵计算方面有进一步的改进空间。目前的系统在大多数情况下都表现出了相当强大的能力，但是在复杂的城市场景中偶尔仍然无法收敛。此外，当前系统使用的是拟牛顿法近似而不是解析计算的海森矩阵，因为实验证明后者在优化性能上提供了更一致的结果。可能可以调整解析海森矩阵以提供更好的收敛特性，而无需使用昂贵的近似方法。

# 4 运动规划方法

如图 4 所示，在城市和高速公路场景中，自动驾驶车辆需要一种运动规划方法，在正常条件下实现协同和平滑的驾驶体验。由于运动规划模块是安全性的关键组成部分，它必须保证一个硬性的周期时间（约 10 Hz）。我们系统中使用的运动规划算法的基本概念是在 [19] 中提出的。本节的目的是对实现的算法进行简要概述，并指出与原始概念相比的重大改进和变化。这些改变的实施导致了驾驶性能的显着提高。





如 [19] 中所讨论的并在图 15 中说明的，2D 轨迹规划平面被分为横向和纵向的 1D 空间（构成了 Frenet 坐标系的基础）。这种方法将道路的几何形状与实际轨迹生成过程分离开来，因为我们使用了前述的参考线作为运动规划对环境几何的内部参考。这关键地限制了可行轨迹的搜索空间，并且显著简化了实现的运动规划方法。此外，我们获得了对基础 1D 轨迹更大的灵活性。进一步，将生成的参考线与模拟的横向空间依赖性相结合，使得考虑自动驾驶车辆的非完整性变得更加容易。另一个优势是，人类通常参考车道标线或当前车道的中心车道来执行其与速度无关的横向移动。因此，分离的规划方式导致了自动驾驶车辆的协同和可理解的运动。

为了应对运动规划问题中的不确定性，我们在预测过程中对动态障碍物状态进行了单峰状态分布建模。这些未来状态分布的生成由预测模块在第2.2节中概述的非分支预测步骤中执行。这使我们能够有效地将关于未来动态障碍物预测状态的不确定性整合到我们的轨迹评估过程中。组合不确定性，如围绕或包含自动车辆的交通互动的不同可能排列，被假定由行为规划模块处理，因此每个运动规划实例只需要评估一个明确定义的未来交通场景的排列。

图16显示了轨迹规划模块的内部结构。轨迹规划模块在运动规划器中的作用在第2.2节中进行了讨论，并在图6中描述。"计算目标"模块重新计算了来自行为规划模块的给定行为动作的跟随目标、停车目标和速度目标，使用匹配的参考线。"任务决策"模块然后使用先进的启发式方法预先确定轨迹规划的最关键目标（跟随、停车、保持速度）。这种预决策使我们能够在与[19]中概述的方法相比显著减少搜索空间。在"生成结束条件"模块中，根据所选择的任务对结束条件进行采样，并对这些结束条件样本进行轨迹搜索以识别最佳轨迹。对于每组结束条件样本，在Frenet坐标系中计算横向和纵向轨迹，并基于相关的成本项计算相关成本。然后，在Frenet坐标系中将横向和纵向轨迹与相应的参考线相结合，以在笛卡尔空间中生成完整的2D轨迹以进行碰撞检查。

运动规划系统的组件将按以下顺序呈现。在第4.1节中介绍了Frenet坐标系计算中使用的1D轨迹类型。第4.2节讨论了初始边界条件及其对运动规划器和控制器之间相互作用的影响。第4.3节涵盖了运动规划任务，第4.4节描述了结束条件的采样。随后在第4.5节讨论了实现的碰撞概率计算。最后，在第4.6节解释了轨迹搜索和选择。

## 4.1 轨迹定义和类型

在Frenet坐标系中独立规划纵向和横向运动的事实极大地简化了规划问题，并使我们能够组合不同类型的轨迹以适应特定情况。对于大多数情景，轨迹选择的驾驶标准是整体舒适性。在文献中，多项式被广泛用于此目的。其他情景则需要考虑到车辆动态能力极限的轨迹。在我们的系统中，我们将这种极限轨迹称为bang-bang轨迹，因为它们在概念上与所谓的bang-bang控制器相似。 在横向和纵向轨迹在Frenet坐标系中生成后，它们被合并并转换为笛卡尔坐标系以检查动态约束的满足程度，并评估与其他障碍物的碰撞风险。在本节中，我们简要介绍所使用的不同类型的1D轨迹。

可行的1维轨迹 在尊重 jerk 和速度相关的加速度限制的同时。这种方法在整个运动规划器中用于评估自动驾驶车辆的动态限制（例如计算最小行驶距离或最小时间，以停止或加速到目标速度）。这种受 jerk 限制的轨迹评估对于任务决策（第4.3节）和有效地对末端条件进行采样（第4.4节）等功能至关重要。此外，如果计算得到的舒适多项式违反了车辆的动态约束，bang-bang 轨迹将作为备用解决方案。 尊重 jerk 和速度相关的加速度约束的 bang-bang 轨迹如下图所示。两列显示了相同的初始和结束条件（在 ṡ、s̈、s 中），但使用了不同的约束值进行轨迹计算。左侧的图显示了带有舒适 jerk 和加速度约束的 jerk 和加速度剖面。使用更保守的舒适约束允许我们计算保持在舒适驾驶区域的 bang-bang 轨迹。右侧的图显示了使用系统的最大 jerk 和加速度约束的 bang-bang 轨迹。在这里，生成的轨迹沿着运动包络的物理可能边界行驶，并尽可能快地达到所需的结束条件。两个加速度剖面显示了速度相关的加速度约束，即随着速度增加，最大加速度逐渐减小。这些速度相关的加速度约束在内部表示为与连续速度范围相对应的离散集合加速度约束。 对于基于时间的计算，计算 bang-bang 轨迹是直接的，因为在这种情况下加速度剖面由分段线性函数给出。通过设置最大（或最小）允许的加速度剖面，我们可以找到达到给定速度所需的最短时间，以及所产生的覆盖距离。另一方面，对于轨迹固定覆盖的情况，计算更加复杂。这是因为没有封闭的形式解来找到由行驶距离参数化的速度。在这种情况下，可以使用基于时间的计算，并使用根查找方法来实现所需的距离。由于时间和距离之间的依赖关系是单调的，具有近似多项式依赖关系，因此这些方法提供了快速和准确的结果。

二维笛卡尔轨迹

横向和纵向轨迹被合并并从 Frenet 坐标系转换到笛卡尔坐标系，结果是一个时空结合的轨迹，如第 1.2 节所述。在构建这些组合轨迹时，纵向轨迹使用非线性采样方案进行时间采样，以便在接近未来时更密集地采样，而在评估范围末端更稀疏地采样（因为未来轨迹将重新规划并可能在较长时间内有所不同）。这有助于减少车辆控制器在二维轨迹点之间引入的数值插值偏差，并使碰撞检测在不久的将来更加精确。

## 4.2 计算初始条件

控制器需要连续的输入以实现平稳和稳定的驾驶行为（如前文所述）。这意味着新规划的轨迹应始终从先前的轨迹中平稳地延续下去。为了使用多项式计算轨迹集，需要定义完全约束的边界问题的初始和结束条件。如第4.4节所述，结束条件在预定的时间范围内变化，但所有轨迹的初始点都是与前一个轨迹对齐的固定点。

为了计算初始点，预测车辆的位置直到当前规划周期的结束时间。由于最终规划的轨迹将在此时刻发送到车辆控制器，因此这个预测步骤对于确保规划周期之间的平稳过渡非常重要。如果存在先前的轨迹，预测的车辆状态将在时间上与轨迹进行匹配，并且沿着轨迹的下一个节点点被作为初始点。先前轨迹的前导部分也被保存并附加到新规划的轨迹前面，以确保控制器有足够长度的轨迹来匹配。如果控制器是根据空间位置匹配来确定来自规划轨迹的期望点，则这一点尤其重要。如果没有先前的轨迹（见下文），则预测的车辆状态本身被视为初始点。

为了在Frenet框架中规划轨迹，必须首先将此笛卡尔初始点转换为本地Frenet框架。为了执行此转换，将初始笛卡尔点与参考线上最接近的点进行匹配，并根据[19]中描述的方法进行坐标转换。计算初始条件的整个任务可以使用方程式7进行总结。

## 4.3 任务决策

（怎么减小采样空间）

在我们的系统中，有效的运动规划需要对可能的轨迹终点条件进行有效的采样。正如已经讨论的，我们通过将可能的纵向终点条件空间分解为一组任务，并对每个任务评估所有终点条件来进行采样，这是之前版本的 CC/PJ-FA 运动规划器所采用的方法，如[19]中所建议的。然后比较每个任务的最佳轨迹，以确定最关键的任务。我们通过在采样之前启发性地预先确定正确的任务来显著提高我们的终点采样的效率，使我们只探索原始搜索空间的一小部分。有关不同采样方案的详细信息请参见第 4.4 节。本节解释了用于预先确定最关键任务的启发式方法。

对于从行为规划器接收到的每个行为动作，运动规划器的任务决策模块在以下三个纵向终点条件子任务之间做出决定：

* 速度保持
* 跟随
* 停止

对于车道行驶，这三个任务选择允许我们定义处理所有必要情况的终点条件采样区域。请注意，这些任务是运动规划模块内部进行的采样分解，旨在提高运动规划效率，并不是行为规划模块中执行的行为动作搜索的一部分。

速度保持是简单的默认驾驶模式，在可达到的距离内没有对自车路径的阻碍。对于速度保持任务，会计算加速到期望速度的终点样本。通常，这个期望速度简单地是车道的限速，然而，如果需要的话，我们会调整目标速度，以确保自车在即将到来的弯道中不违反横向加速度约束。

跟随对于车道内行驶和变道动作都是重要的决定。跟随任务通过调整自车速度以跟踪一个或多个目标车辆的期望跟随范围来考虑围绕自车的动态障碍物的反应。图 5 显示了车道跟随和变道动作的跟随目标示例。速度保持和跟随任务之间的转换应该使驾驶质量受到最小影响。这种转换也应该被自动驾驶车辆的乘客充分感知到。太早决定跟随慢车会导致自车不必要地减速。类似地，决定跟随快车可能会导致自车超过速度限制，或者不自然地追赶一个人类司机不认为重要的车辆。对于领先车辆的反应太迟增加了碰撞风险，并可能导致自车乘客的不满。对于一个人类司机来说，这个任务决定是相当自然的，而且在日常驾驶中不断发生。为了在速度保持和跟随任务之间做出决定，首先计算了自车和领先车辆轨迹的简化预测。自车轨迹是通过计算到达期望速度的舒适轨迹确定的。对于领先车辆，查询了非分支预测（见第 2.2 节），并查找了相应时间的车辆预测。图 19 对比了自车和领先车辆纵向位置的说明性比较。参数 t eval 是将比较车辆位置的时间范围。如果在评估点，预测的自车速度保持轨迹比领先

停止

在以下情况下是相关的：当我们有一个停车点时（例如，在停车标志或人行横道处，融入主要街道，或到达目的地时）。与高速公路情景相比，停车情况在城市驾驶中更为相关。在所有这些情况下，都存在一个明确的停车点，在继续前进之前我们可能希望完全停下来。在在选择其他两个任务之间做出决定后，会决定选择的任务和停止任务之间的决定。首先，计算进行舒适停车所需的行驶距离。如果当前自车位置加上这个舒适停车距离超过了停车点，那么停车任务被认为是潜在关键的，然后将 t eval 时刻的停车任务轨迹距离与所选的速度保持/跟随任务的参考距离进行比较。再次选择最关键的任务。图 19 说明了执行速度保持任务和领先车辆的自车随时间行驶距离。对于自车轨迹，其中 t eval 位置小于 t crit 时，选择速度保持。而对于 t eval 位置大于等于 t crit 的自车轨迹，则选择跟随任务，因为否则自车会违反领先车辆的安全跟随距离。整个决策逻辑总结如图 20 所示。

## 4.4 采样方法

这里采取的运动规划方法需要生成大量的轨迹，以充分覆盖搜索空间。这些轨迹是通过对时间、位置和速度采样不同的终点条件来生成的。在合理的采样分辨率下，对所有可能的终点条件进行详尽采样是不可行的，因为可能产生大量的轨迹。正如在第 4.3 节中讨论的，我们将有趣的终点条件空间分解为三个任务：速度保持、跟随和停止。这极大地减少了我们的采样搜索空间，并有助于提高选择的轨迹质量，因为它允许对终点条件进行更精细的采样。我们通过将采样限制在基于自车动力学约束的可行终点条件范围内，进一步减少了采样工作量。

成本估计可以基于任务相关的启发式方法。由于轨迹是在 Frenet 坐标系中计算的，所以纵向和横向轨迹是分别计算的。由于横向和纵向的解耦，可以对所有轨迹使用单一的横向采样方案，而纵向采样则以任务为依据进行。

在每种情况下，终点条件由四个状态变量描述，其中只有两个值被采样，另外两个值要么被固定，要么未设置。纵向终点条件由向量 (t, s, ṡ, s̈) 描述，其中 s(t)，而横向终点条件由 (s, d, d 0 , d 00 ) 描述。

估计哪些终点条件应该产生可行的轨迹，以及估计哪些轨迹成本低，使我们能够大大减少终点条件范围。可行性可以使用如“bang-bang 轨迹”中的极端加速度轮廓来估计，该轮廓定义了车辆的行驶包络。

**速度保持** 对于速度保持任务，终点条件形式为 (t, ·, ṡ, 0)。目标位置未定义，目标加速度为 0。采样参数为 t 和终点速度 ṡ。速度在 0 和目标速度之间采样，而 t 在达到该速度的最短时间和最短时间加上允许的时间延迟之间采样。对于速度保持任务，我们一直采样到零，以便为可能有益于减速或停车但没有明确定义的领先车辆或停车点的情况提供选项。这种情况可能发生在动态障碍物横穿自车目标行驶车道或静态障碍物需要反应但不完全阻塞车道的情况下（完全阻塞车道的静态障碍物会在行为规划器中产生一个停车点）。图 21 显示了速度保持任务的纵向 ṡ 和 t 采样。图中的绿色区域表示可行驶区域，由最小和最大 bang-bang 轨迹确定，如第 4.1 节所讨论的。

**跟随** 对于跟随任务，终点条件形式为 (t, s, v lead , 0)。目标速度简单地是领先车辆的当前速度 v lead，目标加速度为 0。采样参数为 t 和终点位置 s。请注意，在这种情况下，终点位置 s 是 t 的函数。与速度保持任务类似，使用最短时间 bang-bang 轨迹计算跟随任务的有效时间范围。然后，在领先车辆后面的最小和最大时间间隔内对给定的终点时间采样。对于多辆领先车辆（如图 5b 所示），使用最保守的跟随范围。图 22 显示了跟随任务的纵向 s 和 t 采样。

**停止** 对于停止任务，终点条件形式为 (t, s, 0, 0)。即目标速度和目标加速度均为 0。采样参数为 t 和终点位置 s。时间在使用最大减速 bang-bang 轨迹达到停车点所需的最短时间和舒适到达停车点所需时间之间采样

## 4.5 碰撞风险

对于大多数运动规划方法来说，碰撞检查是主要的计算瓶颈。有效的轨迹规划需要非常高效的碰撞检查，因为每个候选轨迹都必须进行密集采样，并与场景中的每个动态障碍物进行比较。

当然，可以进行简化，例如修剪与自车无交互的对象，或者尝试直接在Frenet框架或甚至1D车道空间中进行碰撞检查。修剪不重要的动态障碍物是一个有用的想法，我们在系统中使用了这种修剪的变体，在每个预测的动态障碍物状态的每个时间步中执行极其高效的轴对齐边界框碰撞检查，以确定给定障碍物是否足够接近以进行碰撞检查。这种接近性测试在每个时间步骤都会执行，以确保只有相关情况会传递给更昂贵的完整碰撞检查。

直接在Frenet框架中进行规划可以允许在执行将每个轨迹从Frenet转换为笛卡尔空间的昂贵过程之前对每个组合轨迹进行碰撞检查。然而，在Frenet框架中进行几何检查并不容易，因为Frenet框架根据参考线的曲率扭曲几何形状。在1D车道空间中进行碰撞检查极大地简化了碰撞检查过程，并且是我们的行为规划模块用于速度的碰撞检查形式。然而，1D碰撞检查过于保守和不灵活，并不适合详细的碰撞风险确定，因为我们的运动规划器预期能够在车道内进行调整，以通过非阻挡的静态障碍物，如停放的汽车。

尽管进行了这些简化，但我们最终仍然需要在2D参考框架中对所有轨迹进行密集的碰撞检查（关于3D问题，如位于自车上方立交桥上的车辆，可以使用上述的3D版本接近性测试进行处理）。有许多算法可以进行快速的二进制碰撞检查，请参阅[22]以了解适用于自动驾驶车辆的方法概述。然而，我们的碰撞检查问题更加复杂，因为我们希望考虑到我们感知、定位和障碍物预测的不确定性——在已经是轨迹规划过程中计算量最大的部分中计算碰撞概率的更复杂的问题中，我们使用了从[4]中极其高效的碰撞概率计算的改编。

[4]中的碰撞概率计算是针对具有不确定的相对位置和方向的两个任意多边形的。为了简化，我们忽略了方向的不确定性（尽管如[4]所示，可以有效地界定此方向不确定性，而不会实质性地影响碰撞概率的计算时间）。我们还将障碍物表示为定向边界框（OBBs），而不是任意多边形，这样我们就可以使用更高效的内部表示。

图 23 显示了两个具有不确定的相对位置的OBB的高效碰撞概率计算的概述。首先，执行近似卷积以生成所谓的组合体OBB。这种卷积是通过将一个OBB的角投影到另一个OBB的本地旋转框架中，并适当地添加它们的尺寸来执行的。这里，本地旋转框架作为OBB定义的一部分进行存储，因此不需要昂贵的三角函数。一旦生成了组合体OBB，碰撞概率就简单地是从原始相对位置不确定性分布中抽样的点落在组合体内的概率。

为了避免昂贵的二维积分，然后应用一个线性变换 L 来对位置不确定性的高斯分布进行归一化，将任意位置不确定性的高斯分布转换为协方差变成单位矩阵的归一化高斯分布。如果将任意高斯分布旋转，使其主要分量轴与参考框架的坐标轴对齐，那么任意高斯分布可以分解为两个1D分布的乘积。

## 4.6 轨迹选择

动态约束、碰撞风险和舒适性是选择车辆最佳轨迹的重要标准。这些标准被合并并制定为一个成本函数和一组约束，这使得可以在统一的尺度上比较轨迹。基于这个成本函数，可以使用不同的搜索方法来找到成本最低的轨迹。

在轨迹选择过程中的主要优先事项是找到具有最小碰撞风险的可行轨迹。智能采样的轨迹应该提供一个合理的候选轨迹集合，这些轨迹不违反自车的动态约束（v、a\_T、a\_N、j\_T、j\_N、κ）。在任何约束违规的情况下，将轨迹标记为不可行，并在轨迹选择过程中予以排除考虑。在进行了初始动态可行性检查后，根据第4.5节中计算的碰撞风险计算碰撞的风险。超过预定义的碰撞风险阈值的轨迹也将在轨迹选择过程中排除考虑。然后，基于一个成本函数评估可行且无碰撞的轨迹，该成本函数衡量了乘客舒适度，如第4.1节所讨论的。用于计算这些舒适成本的任务相关启发式方法在第4.6.1节中解释。

轨迹选择过程的结果优先顺序是：

1. 动态约束满足
2. 碰撞风险
3. 总舒适成本

根据得到的成本函数，可以使用第4.6.2节中描述的搜索方法之一选择最佳轨迹。

### 4.6.1 成本函数

实现的总成本函数 JtotalJtotal​ 定义为 Jtotal=Jlon+Jlat+J2DJtotal​=Jlon​+Jlat​+J2D​ 其中 JlonJlon​ 和 JlatJlat​ 分别表示分配给一维纵向和横向轨迹的成本。项 J2DJ2D​ 考虑了在合并适当的纵向和横向轨迹之后在二维笛卡尔空间中运动所产生的成本。下面讨论了每个成本函数的各个组成部分。

下面的两个函数已用于定义每个成本函数中的不同项。在这里，uu 是一个可测状态变量，依赖于参数 xx（通常是时间 tt 或曲线长度 ss）

### 4.6.2 搜索方法

轨迹的选择基于在为不同结束条件生成的一组轨迹中选择成本最低的轨迹。我们考虑三种不同的方法来找到成本最低的轨迹，这些方法分别是暴力搜索、分层搜索和DIRECT搜索。所有的搜索方法都在相同的结束条件范围内运作，即在第4.4节描述的采样结束条件范围内。更准确地说，暴力搜索和分层搜索在采样结束条件本身上运作，而DIRECT搜索允许在结束点样本之间进行线性插值。这导致DIRECT搜索对采样分辨率不太敏感。另一个区别是结束条件被探索的顺序。这在超时时是相关的，当找到的最佳可行轨迹应该被使用时。

在本节中，我们使用术语“搜索空间”来表示结束条件的空间。这可以是一个离散空间，比如采样结束条件的空间，也可以是一个连续空间，它由采样结束条件的范围表示。

暴力搜索：暴力搜索是最简单的搜索方法。采样结束条件按顺序处理直到完成。对于每个采样结束条件，生成一个轨迹并计算其成本。暴力搜索中的这种顺序评估在搜索被早期中断时是有问题的。在这种情况下，只有一部分连续的结束条件被探索，返回的轨迹可能远离搜索范围中的最优轨迹。当暴力搜索失败时的一个例子是，当前一半的采样结束条件导致引起碰撞的轨迹，但后一半是无碰撞的轨迹。如果只有时间来评估前一半的轨迹，则所有评估的轨迹都聚集在不可行的区域内。另一种思考方式是，暴力搜索对搜索空间的几何形状和该空间上的成本函数的形状是无视的。一般来说，我们可以假设无碰撞轨迹的集合和可行轨迹的集合各自形成一个开集。这又意味着导致可行且无碰撞轨迹的采样结束条件在搜索空间中形成聚类。暴力搜索确保我们找到了最优解（如果它存在于搜索空间中）。但是，如果搜索被早期终止，暴力方法无法有效地探索搜索空间以找到可行的无碰撞聚类。

分层搜索：分层搜索与暴力搜索的唯一区别在于轨迹被评估的顺序。也就是说，在完整的评估周期中，这两种方法所探索的轨迹包括完全相同的结束条件集。在轨迹搜索被早期终止的情况下，返回的轨迹应该提供对实际最佳轨迹的近似。分层搜索通过将搜索空间划分为嵌套网格树来解决早期终止问题，其中搜索空间是根网格。网格使用基于树的广度优先搜索进行扩展，直到探索了所有大于阈值大小的方框为止。之后，搜索根据成本的先验下界猜测进行贪婪搜索。

DIRECT搜索：DIRECT搜索是在文献[5]中引入的。它是一个用于成本函数 f：[0, 1] n → R 的一般优化算法（即，对于定义为超立方体的搜索空间上的任何点返回一个标量成本的成本函数）。在这里，搜索空间是采样结束条件的凸包，它被映射到单位超立方体。映射通过将[0, 1]中的等距网格与每个维度上的样本索引匹配，并在这些点之间线性插值来定义。因此，如果使用了非线性采样，则结束条件的采样率仅影响DIRECT搜索的搜索空间。否则，每个维度上的两个采样点就足够了。

原则上，DIRECT搜索以与分层搜索类似的方式探索搜索空间，即通过将搜索空间划分为尺寸递减的块（在这种情况下是超矩形）。DIRECT搜索从搜索空间的中心点开始，使用整个空间[0, 1] n 作为相应的超矩形。在每一步中，Lipschitz类型的标准为迄今考虑的所有点（及其相应的矩形）提供了一个成本的下界估计，并确定了进一步划分搜索空间的周围点。这导致了在广度和深度上的同时探索。作为停止标准，我们限制了可以评估的点数，以及每个维度允许的划分数。请注意，不会对不同的维度进行区分，并且我们在所有维度上使用相同的相对分辨率。

图 26 显示了每种搜索算法在可能的纵向结束条件的搜索空间上的示例操作。DIRECT搜索方法能够快速缩小到最优解。

## 4.7 开放挑战

在本节中，介绍了一种运动规划方法，已经在高速公路和城市场景中证明成功。然而，在实施和测试过程中，我们已经确定了一些改进措施，可以进一步提高整体系统的性能和稳健性。在本节中，我们想提到这些改进措施，以及提出的建议，这些建议可以作为进一步工作的起点。

### 4.7.1 改进采样方法

在第 4.4 节中概述的采样方法相对于[19]中建议的采样方案以及CC/PJ-FA运动规划器的第一个版本中使用的采样方案是一项重大改进。然而，应该可能进行额外的参数适应，以更好地将我们对端点条件的采样适配到自车辆的可行驶范围。这种改进的空间在图 22 中清晰可见，其中在 t min 时，我们在只能实际达到范围边界时对整个期望的跟随范围进行采样。

此外，目前正在对以下端点条件进行采样，以确切匹配前车的平均速度，并计算相对于前车的平均位置的跟随范围。对于具有单峰状态不确定性的前车，希望还在速度维度进行采样，并根据不确定性的大小采样更大的位置范围。对于具有多峰状态不确定性的前车，我们目前使用最保守的模式来计算我们的跟随案例的样本。以可处理的方式正确处理多模态不确定性是一个未解之谜。

### 4.7.2 考虑车道边界

如第 4.6.1 节所述，横向成本函数仅评估相对于参考线的横向偏移的积分。这会增加具有大横向偏移的轨迹的成本，但并不保证车辆在车道内行驶，或在需要变道时保持在相邻车道内。

因此，可以添加对于2D轨迹与相应车道的车道边界的额外检查。这将确保自动驾驶车辆不会驶出其预期的车道。

### 4.7.3 成本函数权重优化

如第 4.6.1 节所述，成本函数对用户定义的权重非常敏感。为了使系统在不同速度范围内更加健壮，我们首先尝试了基于时间的成本函数。这减少了不同成本项中速度和距离依赖性所带来的困难，但并没有简化成本函数权重的调整过程。由于成本项之间的相互依赖性以及自动驾驶中的使用案例数量、交通和道路配置的显著增加，参数化轨迹选择的过程变得非常困难。尤其是在高速公路和城市场景中经历了各种情况的情况下，找到一组权重以实现所需的驾驶行为特别困难。

在这种情况下，首要挑战是找到各自 1D 或 2D 成本函数中的显著成本项。其次，这些相关项必须适当加权以在所有情况下实现类似人类的驾驶行为。一个可能的方法是开发一个优化框架，利用预定义的一组预期操作和交通情况来优化成本函数的权重。

另一个可能的方法是将手动驾驶操作记录的结果轨迹与模拟运动规划器在循环模型设置中的轨迹进行比较。通过调整成本函数的权重，以使手动轨迹与运动规划器选择的轨迹之间的偏差最小化，将导致各自成本项的适当权重。

这里的主要挑战是定义这个过程的关键操作，并识别和实现可能成本项的优化过程。特定情况下的权重也可以被视为一个潜在解决方案，如果一组权重无法泛化到所有必要的情况。

### 4.7.4 改进轨迹选择

成本函数被设计为产生一个反映运动规划器要求和优先级的轨迹排序。这在使用蛮力搜索时非常适用，因为在这种情况下，所有轨迹都被评估和考虑。

使用基于优化的搜索方法的优点是双重的。首先，评估的轨迹较少，从而减少了计算时间，因此可以扩展搜索空间或提供更精细的解决方案。其次，优化技术可以迅速导向最佳区域，即使在几个循环后被中断也能返回良好的结果。另一方面，任何优化技术都对待被优化的成本函数进行假设。一个基本的假设是连续性，尽管通常需要更强的利普希茨连续性。这使得优化方法能够估计函数变化的程度，从而减少了最优解可能出现的区域。

然而，这里使用的成本函数并没有专门设计用于基于优化的搜索方法。这是一个相当困难的问题，因为成本函数在轨迹空间上定义，即 c: {轨迹} → R，而搜索方法在端点条件上定义成本，即 f: {端点条件} → R。如果我们用一个函数 g 来表示将端点条件转换为轨迹的轨迹生成，我们可以通过复合函数 f = c ◦ g 来关联上述成本函数。这意味着，对于一个连续函数 c，我们仍然可能有 f 是不连续的，仅仅是因为轨迹生成不会连续地将端点条件映射到轨迹上。一个例子是当多项式轨迹违反约束时，使用"bang-bang"轨迹代替。为了更好地使用已建立的优化技术（例如DIRECT搜索），值得调整成本函数 c 并证明 f 的适用性。

### 4.7.5 重用先前规划的轨迹

当前，规划问题在每个规划周期都是从头开始制定的，并且生成并评估了一组新的轨迹。这是一个好的方法，因为它使系统能够快速对环境中的动态变化做出反应。如果由于时间限制或不幸的采样选择了一个糟糕的轨迹，它还允许进行即时恢复。然而，在许多情况下，动态场景在规划周期之间变化不大，新轨迹的主要部分与先前规划的轨迹相似。在下一个周期的轨迹选择方法中使用先前规划的轨迹可能会提供更一致的结果。这也可以通过重用先前规划的轨迹的端点条件或引入一个新的成本项来偏向于选择与先前规划的轨迹相似的轨迹来实现。

# 5 特殊情况

到目前为止，所介绍的运动规划方法是一种非常通用的方法，旨在处理正常车道驾驶中发生的大多数情况。在本节中，我们指出了一些需要特殊处理的特定情况，并详细介绍了它们的处理方式。这些特殊主题包括：进行车道变换，处理找不到有效轨迹的情况以及静止管理。

## 5.1 车道变换

在Frenet框架中以参考线规划的概念是基于车道内驾驶的，因为它假设与参考线的偏差相对较小。当执行到相邻车道的车道变换时，我们只需从当前驾驶车道的参考线切换到目标车道的参考线。这种参考线切换与我们的运动规划方法无缝配合，并导致新Frenet框架中具有较大横向偏移的初始点。在规划横向车道变换轨迹时可能会出现的一个问题是，可以创建明显超出目标参考线的有效轨迹。对一系列终点条件进行广泛采样可以缓解这个问题，因为我们很可能会采样一些不超出的轨迹，然而，我们在运动规划器内的目标是尽可能只采样可行的轨迹。

减少超出轨迹的一种方法是在必要时使用更激进的轨迹。如果使用允许的舒适约束生成的多项式横向轨迹超出了范围，那么将使用具有增加横向动力学的Bang-Bang轨迹。由于Bang-Bang轨迹只有两个加速脉冲，我们可以执行约束松弛搜索来确保我们得到的轨迹在不超出目标车道的情况下尽可能舒适。

## 5.2 回退轨迹

我们的上下文中的自动驾驶车辆具有完全责任，并且需要具有故障操作能力。即使在由于约束、碰撞或规划系统其他部分的问题而找不到采样轨迹的情况下，车辆也应该能够继续执行合理的动作。

这是一个关键的话题，在系统可以引入市场之前，需要具有高水平的解决方案成熟度和测试。作为该项目的一部分，我们已经探索了使用运动规划架构来生成回退轨迹，以便稍后可以轻松地将类似的回退功能添加到现有的规划系统中。

由于回退轨迹是最后的手段，并且应始终可用，因此基本思想是首先计算一小组回退轨迹，以便它们在规划周期的任何时间都可用。这对于规划器在时间用尽时未能找到有效轨迹的情况至关重要。对于我们的系统，回退轨迹限制为使车辆停下的轨迹，并且也被称为紧急轨迹。

与运动规划器中使用的多项式轨迹相比，紧急轨迹是通过组合用于纵向和横向移动的Bang-Bang轨迹来计算的。首先，通过手动指定的参数范围计算具有不同加速度限制的纵向轨迹。引入这些纵向加速度范围参数是为了确保在由于一些采样问题或轻微动态约束违反而导致故障时生成平稳制动轨迹。此外，对横向偏移进行一系列采样，以便在适当时以偏离参考线的方式停下。对于我们的项目，紧急轨迹的成本函数与正常轨迹评估的成本函数保持相同（在第 4.6.1 节中解释）。然而，显然需要一个更复杂和专门的成本函数来确保安全和健壮的决策制定。除了运动规划方面的考虑外，紧急轨迹通常在系统边界处运行，并且可能需要不同的控制器参数化。

## 5.3 静止管理

在非常低速下控制汽车在自动驾驶和辅助系统中一直以来都是一个历史性的难题。这是由于在这个速度范围内相对较高的传感器噪声和执行器滞后性。它导致了较大的控制误差，并与底层/高层概念交互非常糟糕，请参阅第 4.2 节。静止前车状态中的噪声尤其棘手，因为小量感知噪声可能会导致期望的跟随行为出现较大的相对波动。因此，应用正常的运动规划方法会导致爬行式自动驾驶车辆，它将缓慢地撞上静止的前车或越过期望的停车点。

为了避免这种情况下的爬行运动，运动规划器简单地指示车辆控制器在车辆速度低于指定阈值并且计划轨迹还不足以离开静止管理区域，也称为“死区”，时施加恒定减速。在静止管理期间，转向角保持恒定，以避免在此停止阶段进行不必要的转向运动。这种方法导致了舒适可靠的停车运动，并通过避免处理接近零速度的需求来降低了规划问题的复杂性。

## 5.4 挑战

将某些特殊情况单独处理的问题在于它可能导致过于专门化的情景解决方案，并且很难将其很好地整合到正常的运动规划系统中。在本节中，我们列出了在应用这些方法时观察到的一些令人关注的问题。

### 5.4.1 车道变换动力学

实现的运动规划方法仅操纵横向动态以避免超出目标车道。当前的客户系列项目表明，有必要有机会指定车道变换运动的横向动态。一个方法是基于两步轨迹。这类似于一些ACC系统中车辆接近的建模方式，请参阅 [15] 和第 4.1 节。采用这种方法，首先选择两点之间的目标点，然后通过将初始轨迹与该目标点结合，以及从该点到终点的第二个轨迹来获得轨迹。选择目标点允许我们操纵横向运动的动力学。例如，选择目标点在目标车道的中心线上会导致更快、更动态的车道变换，而选择两条车道中间的目标点位置和加速度可以强制在车道变换中间产生指定的加速度。类似的方法已经在 CC/EYM 内部的 Lane-Change-Assist 系统中实现。

### 5.4.2 过渡到备用轨迹

由于我们主要使用五次多项式来计划我们的一维轨迹，我们只能确保对称约束的端点的 C2 连续性。这意味着在每个规划周期中，对于第三阶导数，即纵向的 jerk 和横向规划中的曲率率，连续性不能保证。然而，我们系统中使用的控制器概念，特别是前馈控制，使用这些高阶导数输入来计算控制器输出，从而直接影响车辆行为。这意味着 jerk 和曲率率的不连续性可能会被乘客感觉到不愉快的行为。这种问题在横向移动的情况下最为突出，因为人类乘客对微妙的横向加速度和方向盘的可见运动非常敏感。

### 5.4.3 横向备用轨迹

对于备用轨迹的生成，我们目前与参考线相当接近以生成横向轨迹。备用轨迹通常受到初始车辆状态和与参考线的偏移的约束。对于车道内驾驶的情况，这非常有效，因为最佳行为通常是尽可能接近参考线。然而，如在第 5.1 节中已经解释的车道变换的情况下，参考线被切换到目标车道，并且触发一个零偏移的备用轨迹会导致自动车辆朝着相邻的车道进行激烈转向。从安全性的角度来看，这种行为是极不可接受的。另一个选择是恢复到原始车道的参考线，但是，如何选择参考线必须以稳健和防错的方式处理。

### 5.4.4 静止管理的碰撞检查

如第 5.3 节中所述，实施的静止管理方法主要导致舒适的停车运动。然而，在此恒定减速阶段期间，自动车辆的移动未受运动规划器的碰撞检查。特别是在恒定转向角和车辆控制器内部减小的减速值的情况下，可能会导致不自然的横向偏移，并可能导致在狭窄空间中发生碰撞。由于这种方法分离了计划的和最终执行的运动，因此从理论上讲，无法保证碰撞风险最小的机动。实际上，到目前为止还没有观察到这种问题，因为对于我们的系统，静止管理是在 vego < 0.5 m/s 的情况下应用的。这是因为，对于高速公路和城市驾驶来说，安全距离已经足够大，可以容纳小的误差。然而，这种方法不推荐用于停车或类似情况，这些情况在低速时需要高精度的位置和碰撞检查。

### 5.4.5 停止的位置精度

静止管理盲目地应用恒定减速度的一个缺点是失去了位置的可控性。这意味着我们无法保证严格准确地在所需距离处停车。我们的保守方法是在停止点之前停止，这可能会导致问题，例如在交通灯或停车线处，需要更严格的纵向精度。在城市交通密集的情况下，不自然的间隙和小的位置不准确性可能会让其他驾驶员感到恼火。

# 6. 结果

以下部分突出了所提出的运动规划方法在常见的高速公路和城市场景中的结果。下面显示的结果是使用我们集成的仿真环境创建的，但它们与相应的观察到的真实世界场景相当吻合。

图 28 显示了一个示例的跟随场景。自动驾驶车辆显示为灰色车模型，而前导车显示为实心的蓝色矩形框。彩色的透明框代表车辆的预测状态。位于前导车后面的黄色标记区域描述了自动车辆相对于当前时间步的前导车的理想跟随范围。这个预测到未来的跟随范围定义了用于终点条件采样的位置范围。计算得到的轨迹以绿色和蓝色显示在自动车上方，其中高度表示速度。蓝色轨迹表示违反约束的轨迹，而绿色轨迹表示无碰撞且动态可行的轨迹。与一个或多个周围车辆相撞的轨迹将被着色为红色。选择的轨迹显示为厚厚的黄线。有关运动规划信息，例如选择的行为动作、计算得到的任务、当前自动车辆的状态、前导车辆的状态、期望速度和到达下一个停止点的距离显示在附加到自动车辆的白色文本显示中。圆形的蓝色线显示了计算得到的参考线。

## 6.1 车道跟随

在高速公路上的大部分时间都是在车道内行驶，这使得车道跟随成为运动规划系统的一个重要方面。如上所述，对于来自行为规划器的特定行为，运动规划器的任务决策模块会决定进一步的三个子任务。图 29 显示了车道内跟随行为的情景。

图 29a 展示了自动驾驶车辆实现期望速度尽可能舒适地行驶的基本优先级（任务：速度保持）。这种行为是由于没有其他车辆限制自动车辆的运动。所示的轨迹束说明了对不同终点速度的采样。图 29b 显示了在即将到来的弯道中限制横向加速度而降低期望速度的情况。弯道后，自动车辆再次加速以达到最初期望的速度。图 29c 展示了几秒后的同一情景，此时自动车辆靠近了一辆较慢的前车。自动车辆被迫减速以保持期望的跟随范围，并跟踪前车的速度。这些机动过程中的速度曲线显示在图 30 中。该图显示了自动车辆由于即将到来的弯道和较慢的前车而降低其速度的过程。

## 6.2 车道变换

对于高速公路和城市环境而言，最重要的机动之一就是变道超车慢车。图 31 展示了一种在高速公路上的典型变道情景。自动驾驶车辆被车道上的一辆较慢车辆挡住，因此执行了变道超车。在变道过程中，运动规划器使用的参考线是目标（目的地）车道的参考线。为了超车，生成的轨迹会穿过前车，因为预测到前车会持续行驶。一旦变道完成，生成的轨迹再次用于正常的车道内行驶（图 31c）。这种机动过程中的速度曲线显示在图 32 中。该图显示了自动驾驶车辆由于较慢的前车而减速，然后在变道过程中略微加速。完成变道后，它加速到期望速度。

## 6.3 交叉口

在城市场景中经常能观察到的另一个使用案例是具有强制停车点的交叉口。图 33 展示了这样一个城市场景。在图 33a 中，自动驾驶车辆正在接近 T 型交叉口前的停车点。对于停车任务，轨迹终点在所需停车点周围密集采样。图 33b 显示了自动驾驶车辆在停车点停止。停下后，行为规划器发送一个继续行动，自动驾驶车辆继续前进，跟随参考线并右转。然后，自动驾驶车辆加速到期望速度，如图 33c 所示。这些机动过程中的速度曲线显示在图 34 中。该图显示了自动驾驶车辆由于停车点而减速，停下来并再次加速以达到其期望速度

# 7 摘要与讨论

本报告中描述的运动规划系统旨在在结构化环境中工作，例如车道驾驶，并为在这些情况下运行的自动驾驶系统提供了一个良好的潜在解决方案。运动规划系统表现出可预测性和安全性，并优化乘客的舒适度。该系统易于适应新的高级行为，并应为未来规划系统提供坚实的运动规划核心。与基于优化的方法相比（这很可能是这种系统的主要竞争方法），该系统的响应速度（因此在时序上的稳定性）更好，并且即使在复杂情况下也易于维护。

## 7.1 展望

当前系统可以通过各种方式进行改进，并扩展其功能。运动规划系统的模块化方法使我们能够独立地解决许多问题，正如本报告中各节中描述的挑战所示。

当前系统主要设计和测试用于高速公路驾驶和轻度城市使用情况。具有紧急转弯、机动空间少、交通密集的挑战性城市使用情况将会带来问题，并需要额外的改进或重新设计。这对于城市驾驶情况尤其如此，其中车道驾驶概念暂时中断，例如对于需要交通双方合作通过可用空间的城市小街道。

从长远来看，基于优化的方法提供了许多潜在的优势，并且可能在能力和稳健性上超越离散采样的方法。然而，这种方法在工作上本质上更难处理，并且在涉及的概念（轨迹生成、轨迹评分、车辆约束和碰撞检查）之间引入了更高程度的耦合。这种紧密耦合的运动规划系统不适合由大型团队共同开发，而且必须有证据表明它们在离散采样的方法之上提供了明显的优势，然后才能作为主要的运动规划系统进行追求。

另一种目前备受关注的运动规划方法是基于深度学习的方法，例如端到端深度学习系统，它们直接从传感器输入到轨迹输出或执行器命令学习功能映射。然而，基于深度学习的系统在理论层面仍然不够清晰，目前此类端到端自动驾驶系统属于投机领域。学术界和工业界目前尚未完全理解这些方法将面临的限制和规模化挑战。此外，这种系统在行为验证和内省方面存在重大挑战。我们认为这些方法在未来几年内对于工业应用来说是不现实的。