arXiv:2303.00861v1 [cs.RO] 1 Mar 2023

SLAS：用于高速公路导航的速度与车道建议系统

Faizan M. Tariq[1], David Isele[2],John S. Baras[1] 和 Sangjae Bae[2]

摘要—本文提出了一种分层式自动驾驶车辆导航架构，由高层速度与车道建议系统（SLAS）以及低层轨迹生成和轨迹跟随模块组成。具体而言，我们针对自动驾驶自车在交通流中行驶的多车道高速公路驾驶场景。我们提出了一种基于滚动时域混合整数优化的 SLAS 新方法，旨在最小化行驶时间，同时兼顾乘客舒适度。进一步地，我们对所提出的方法进行了多项改进，以提高整体计算效率并实现实时性能。在 CARLA 仿真环境中，将该方法与先进的轨迹生成和轨迹跟随框架结合使用，并与现有方法进行对比，结果展示了所提出方法的有效性。

I. 引言

变道被认为是风险最高的驾驶行为之一，因为它高度依赖于对邻近车辆多模态轨迹的预测，并且需要及时决策 [1]。此外，它还受到多种不确定性因素的影响，例如道路状况、测量精度，以及道路交通参与者行为中具有长尾特征的不确定性。然而，如果执行得当，变道结合速度调整能够在确保乘客舒适度的同时，显著减少总体行驶时间 [2]。

进一步来说，可以考虑图1所示的场景。根据对邻近车辆（以橙色显示）的预测运动（以较浅色调显示），自车（以蓝色显示）可能会决定向左变更车道，以尽量缩短其行驶时间；也可能在当前车道减速，以保持安全性。然而，尽管当前车头时距更大，自车贸然向右变更车道并因此被困在一辆行驶缓慢的车辆后方，仍是不明智的做法。这一简单场景凸显了前瞻性以及较长规划时域对于自动驾驶车辆战略决策的重要性。

MOBIL [3] 等现有方法使我们能够变更车道，但其行为往往具有贪婪性（优先考虑即时奖励），这可能导致次优性能。文献 [2]表明，受 A* 启发的方法能够改善变道性能，但其公式仅适用于恒定速度。此类方法无法评估速度调整在降低整体行驶时间方面的收益。

[1]美国马里兰大学帕克分校，马里兰州，美国。电子邮箱：{mftariq,baras}@umd.edu.[2]本田研究所，圣何塞，加利福尼亚州，美国。电子邮箱：{disele,sbae}@honda-ri.com.本研究得到美国本田研究所的支持。

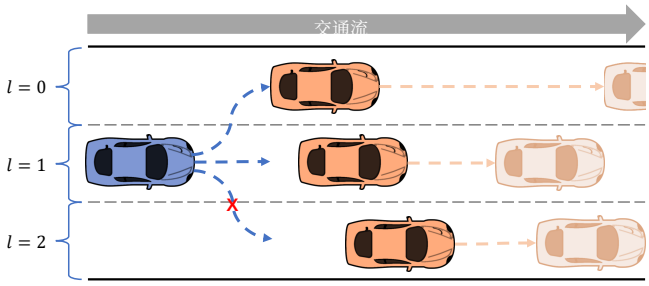

图1. **示例说明**。当前方有一辆低速行驶的车辆时，自车（蓝色）可以选择变道至快速行驶的车道（左侧）以缩短行驶时间，或在不变道的情况下调整速度以确保安全性；但转入低速行驶的车道（右侧）并不明智，因为这样既无法缩短行驶时间，也无法提高安全性。

正如第 IV 节将要说明的，有时可能需要牺牲短期收益，以获得长期性能改进。在这种情况下，采用速度调整并结合较长规划时域的方法具有前瞻性，能够显著改善结果。此外，将速度调整纳入决策过程可以降低轨迹不可行性的风险，因为环境条件可能会使自车无法以恒定参考速度行驶，而低层规划器可能无法处理这种偏差。因此，本文提出一种基于低复杂度滚动时域优化的方法，在较长规划时域内输出结合速度调整的变道机动（$> 15s$），同时保证安全性。长时域战略决策使自车能够主动预判并应对复杂的驾驶情境。

文献综述： 在文献中，速度和变道决策通常从运动规划器的角度进行考虑 [4], ，这使得规划器能够同时确定目标车道和执行机动所需的航路点。文献中的运动规划方法大致可分为基于采样的方法、基于学习的方法和基于优化的方法。

在基于采样的方法中，相较于基于路网的方法等多查询方法，尤其是 RRT 的各种变体在内的单查询方法更受青睐，因为它们执行速度更快，并且能够纳入非完整约束 [5]。尽管这些方法能够通过从可达安全集[6],中采样可行轨迹来提供安全保证，但由于单条轨迹的拼接，整体驾驶体验往往不太舒适。此外，这些方法所提供的渐近最优性保证

并不能助力其在复杂驾驶场景中的现实世界实现，因为它们往往具有较高的样本复杂度 [5]。

对于基于学习的方法，首选方案似乎是在仿真环境中应用各种强化学习技术 [7], [8], [9], [10], [11]。这些方法虽然在仿真中表现良好，但由于需要大量训练数据、训练过程中会探索不安全行为，而且通常无法处理边缘情况，因此在实际应用方面令人担忧。它们主要使用神经网络作为函数逼近器，虽然计算复杂度较低，但也缺乏可解释性和安全保证。

最后，基于优化的方法，尤其是最优控制方法的衍生方法，在文献中十分常见。与基于势场的方法[12] 相比，后者虽能实现不错的碰撞规避效果，却无法纳入车辆动力学；而最优控制方法 [13], , 尤其是模型预测控制（MPC）方法的衍生方法 [14], [15], [16], , 则能在纳入车辆动力学的同时实现出色的碰撞规避效果。然而，这种性能是以较高的计算复杂度为代价的，主要原因在于非凸碰撞规避和非线性动力学约束。进而，规划时域被限制在短短几秒内。

贡献：自动驾驶车辆算法设计的关键要求包括实时运行、安全保证、针对某些指标的最优性，以及对道路交通参与者行为变化性的考量。鉴于这些要求，我们提出了一种基于优化的行为规划框架，使自动驾驶车辆能够在多车道高速公路上进行机动。该方法在具备基于优化方法优势的同时，通过采用解耦车道指示器动力学的二进制表示来替代横向动力学，并利用算法改进辅助数值计算，实现了较低的计算复杂度。具体而言，我们的方法提供：

- 在行驶时间和舒适性方面的最优性；
- 安全保证和可行性保证；
- 适用于较长规划时域的实时性；以及
- 设计模块化，支持集成外部轨迹预测模块。

所提出的方法填补了研究空白，满足所有关键算法要求，同时具备前瞻能力，能够做出带来长期性能收益的战略决策，正如第 IV 节所验证的那样。

II. 问题表述

本节介绍算法流程，并对后续章节将使用的道路、观测和车辆动力学模型进行形式化定义。

A. 算法流程

图 2 展示了所提出导航架构的算法流程，并参考了现有的各种

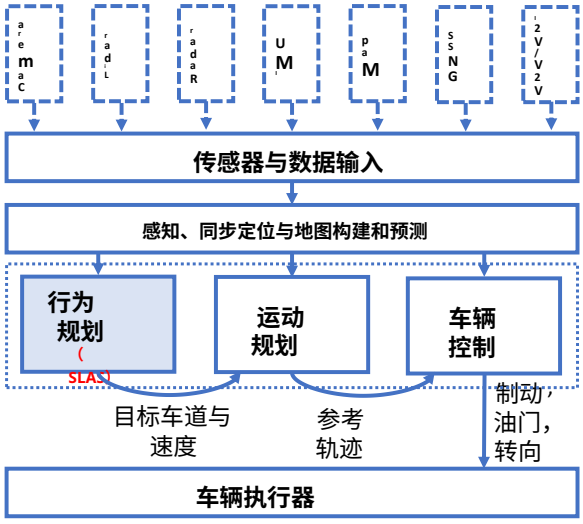

图2. 所提出的导航架构的算法流程。感知和同步定位与地图构建（SLAM）模块处理原始传感器输入数据，以在统一的参考坐标系中确定自动驾驶车辆相对于各类环境实体的位置。随后，这些信息被传递至导航栈，该导航栈由行为规划、运动规划和车辆控制模块组成。导航模块的输出进一步以执行指令（制动、油门和转向）的形式传递给执行器。

部署在自动驾驶车辆上的算法模块。导航栈各组成部分（以虚线矩形标出）的分类体系借鉴自 [17]。该流程本质上改进了 [2] 中介绍的流程，方法是增加了一个速度建议系统。

我们的重点是开发行为规划模块，该模块在图2中以 SLAS 标出。SLAS 输出目标车道和参考速度，供运动规划模块用于为自车生成参考轨迹。车辆控制器计算油门和转向指令，以相应地跟踪该轨迹。

对于运动规划模块，我们采用神经网络集成模型预测控制（NNMPC）[18] ，因为它能够在轨迹生成过程中兼顾邻近车辆的行为。为简化问题，我们假设对其他车辆的感知以及对自车的定位均已知且不存在任何不确定性；不过，模块化架构使我们能够在整体框架中集成任意感知或 SLAM 模块。

符号说明

在全文中， $\mathbb{Z}$ 表示整数集， $\mathbb{R}$ 表示实数集。对于某些 a， $c \in \mathbb{Z}$和 $a < c$，我们记作 $\mathbb{Z}_{[a,c]} = \{b \in \mathbb{Z} \mid a \leq b \leq c\}$。对于某些 e， $g \in \mathbb{R}$ 和 $e < g$，我们记作 $\mathbb{R}_{[e,g]} = \{f \in \mathbb{R} \mid e \leq f \leq g\}$。

B. 道路模型

物理道路结构被建模为一条曲率可忽略、交通流单向行驶的连续多车道高速公路。高速公路上的车道界限清晰，在任意给定时刻 k，可用

供车辆行驶的车道数量记为 $N_l(k)$，道路限速记为 V_l。因此，在给定时刻 k 可供行驶的车道集合记为 $\mathbb{L}(k) = \mathbb{Z}_{[0,N_l(k)]}$。我们采用 Frenet坐标系，其中沿道路方向的距离称为纵向位移（s），垂直于道路方向的距离定义为横向位移（d）。每条车道均分配一个车道指示变量 l。相对于交通流方向，最左侧车道的取值为 $l = 0$，其余车道的 l 值则依次递增，如图1所示。

C. 车辆模型

由于我们的目标是在较长的规划时域（$> 15s$）内进行实时计算，因此我们采用线性化、解耦的动态系统对车辆动力学建模。对于道路曲率通常较小的高速公路驾驶场景，假设横向动力学与纵向动力学相互解耦是合理的， [19],尤其适用于行为规划层。因此，我们采用线性恒加速度模型描述纵向动力学，并用车道指示变量对横向动力学进行抽象。对于换道动力学，我们将移动平均滤波器与取整函数结合起来，以建模自车完成变道所需的时间。该模型可简洁地表示为:

$$s_0(k) = s_0(k-1) + \frac{v_0(k-1) + v_0(k)}{2} \cdot T_s \tag{1}$$

$$l_0(k) = \left\lfloor \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}(k-i) + \frac{1}{2} \right\rfloor \tag{2}$$

其中， s_0（k）、 v_0（k）、 l_0（k）和 $\mathcal{L}$（k）分别表示时间点 k自车的纵向位移、速度、车道指示变量和目标车道；下标 $_i$ 用于索引道路上的车辆，其中$_0$ 专用于自车； T_s 表示离散时间步长； N表示变道所需的时间步数。系统在时间点 k 的状态（x_0（k））和控制输入（u_0（k））定义如下:

$$x_0(k) = \begin{bmatrix} s_0(k) & l_0(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R} \times \mathbb{L}(k) \tag{3}$$

$$u_0(k) = \begin{bmatrix} v_0(k) & \mathcal{L}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}_{[0,V_m]} \times \mathbb{L}(k) \tag{4}$$

其中， V_m 表示自车的最大速度。

D. 观测模型

出于实际考虑，我们将自车的可视范围限制在传感器感知极限内，记为R_v。因此，在时间点 k处自车可视范围内的车辆集合，表示为 $\mathbb{O}(k)$，定义如下:

$$\mathbb{O}(k) = \{ i \in \mathbb{Z}_{>0} \mid |s_i(k) - s_0(k)| \le R_v \} \tag{5}$$

其中， $s_i(k)$对应于被观测车辆的纵向位移。

备注*1*： 对于多车道高速公路驾驶场景，遮挡并不起显著作用，因此我们在现有公式中不予考虑这一因素。不过，正如 [20] 所示，由于滚动时域方法会基于任意时刻可用的最新信息做出决策，因此所提出的框架能够轻松适应遮挡和测量不确定性。

III. 方法论

本节介绍用于生成被观测车辆未来轨迹的预测模型，并讨论所提出的基于滚动时域优化的行为规划模块。

A. 轨迹预测

可靠地预测其他交通参与者的行为和轨迹，对于自动驾驶车辆安全机动至关重要。第 III-B 节提出的算法能够集成文献中任何通用的预测模块 [21]，只要该模块能够为给定车辆提供确定性的未来预测轨迹即可。在本文中，我们构建了一个低复杂度预测模型，以体现所提出方法的灵活性和高效性。

对于观测到的车辆 $i \in \mathbb{O}(k)$，使用分段线性函数预测其未来速度曲线，同时假设在预测时域内车道状态曲线保持不变。在给定时间步 k，通过对过去 $o_i^k > 1$ 个速度观测值进行线性回归并最小化均方误差，得到估计的加速度参数（$\bar{a}_i^k$）和估计的速度参数（$\bar{v}_i^k$）。基于这些估计参数，我们按如下方式预测未来速度和纵向位移:

$$\hat{v}_i^k(j) = \begin{cases} \bar{v}_i^k, & j = 0 \\ \hat{v}_i^k(j-1) + \bar{a}_i^k \cdot j, & 0 < j \le H_a \\ \hat{v}_i^k(j-1), & j > H_a \end{cases} \tag{6}$$

$$\hat{s}_i^k(j) = \begin{cases} s_i(k), & j = 0 \\ \hat{s}_i^k(j-1) + \frac{T_s}{2} \cdot (\hat{v}_i^k(j-1) + \hat{v}_i^k(j)), & j > 0. \end{cases} \tag{7}$$

这里， H_a 对应于加速度时域，而 $\hat{v}_i^k(j)$ 和 $\hat{s}_i^k(j)$ 分别表示从当前时刻 k起向未来 j 个时间步时车辆 i 的预测速度和纵向位移。

注 *2*： 由于所提出框架具有模块化特性，第 III-B 节详述的行为规划模块可以与基于先进机动（例如马尔可夫链 [22]）和基于交互（例如社交生成对抗网络 [23]）的轨迹预测模块协同工作，从而实现交互式机动行为。

B. 速度与车道建议系统

我们的行为规划模块——速度与车道建议系统，简称 SLAS——旨在确定一系列速度和变道指令，使自车能够尽可能提高速度，从而缩短行驶时间，同时兼顾驾驶员舒适性，并遵守车辆的动力学、执行器和安全限制。该模块的输出是一条相对平滑的速度和车道

变化曲线，随后将其传递给运动规划器。行为规划模块必须纳入动力学和执行器限制，以免向运动规划器提供无法实现的目标，从而危及整个系统的安全性。

在后续讨论中，我们将给出 SLAS 优化问题的形式化表述；重点说明为降低计算复杂度所需的修改；并进行安全性和可行性分析。

1) 带整数约束的优化问题： SLAS 被表述为一个优化问题，其目标是在最大化速度的同时，尽量减少频繁变道和速度的突然变化。在时间点k，SLAS 的输出是控制输入 $u_0(k+1)$，如式（4）所定义。优化问题表述如下：

$$\min_{\substack{v^k(1),\cdots,v^k(H);\\ \mathcal{L}^k(1),\cdots,\mathcal{L}^k(H)}} \sum_{j=1}^{H} [-\gamma_1 \cdot v^k(j) + \gamma_2 \cdot (\mathcal{L}^k(j) - \mathcal{L}^k(j-1))^2 + \gamma_3 \cdot (v^k(j) - v^k(j-1))^2] \tag{8}$$

$$\text{s.t.} \quad s^k(0) = 0 \tag{9}$$

$$v^k(0) = v_0(k) \tag{10}$$

$$\mathcal{L}^k(p) = l_0(k), \quad \forall p \in \mathbb{Z}_{[-N+1,0]} \tag{11}$$

$$\forall j \in \mathbb{Z}_{[1,H]}:$$

$$v^k(j) \in \mathbb{R}_{[0,V_l]} \tag{12}$$

$$\frac{v^k(j) - v^k(j-1)}{Ts} \in \mathbb{R}_{[A_{\min}, A_{\max}]} \tag{13}$$

$$s^k(j) = s^k(j-1) + \frac{v^k(j-1) + v^k(j)}{2} \cdot T_s \tag{14}$$

$$\mathcal{L}^k(j) \in \mathbb{L}(k) \tag{15}$$

$$\mathcal{L}^k(j) - \mathcal{L}^k(j-1) \in \mathbb{Z}_{[-1,1]} \tag{16}$$

$$l^k(j) = \left\lfloor \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}^k(j-i) + \frac{1}{2} \right\rfloor \tag{17}$$

$$\min_{i \in \mathbb{A}(k)} \{|\hat{s}_i^k(j) - s^k(j)|\} \geq L_i^s(j), \tag{18}$$

$$\mathbb{A}(k) = \{a \in \mathbb{O}(k) \mid l^k(j) = l_a(k)\}.$$

目标函数： 在上述公式中，优化变量是自车的速度（$v^k(j)$）和目标车道（$\mathcal{L}^k(j)$），即从时间点 k开始向未来规划 j 步。其中， H 表示规划时域。目标函数（8）中的标量化参数 γ_1、 γ_2 和 γ_3 分别用于权衡最大化速度、减少变道以及减少速度的突然变化。增大 γ_1 会产生更激进的驾驶行为，优先考虑最大化速度；而 γ_2 和 γ_3 则共同侧重于通过分别减少变道和速度变化来最大化乘客舒适度。

动态约束：这些约束旨在确保解的动态可行性。约束式（9）、（10）和（11）分别用于根据时刻 观测到的数值，为优化器初始化纵向位移、速度和目标车道。约束式（12）和（13）分别通过限速和车辆的加速度限制来约束自车的速度。随后，利用自车的速度计算式（14）中的预测纵向位移。

任一规划步（j）上的目标车道值均受 (15)、(16) 和 (17) 约束，只能取可达值。其中，(15) 将目标车道限制在可用车道集合（$\mathbb{L}(k)$）内，(16) 确保如有需要，只变道至相邻车道；(17) 则对变道所需的时间步（N）进行建模。通过引入辅助整数变量，取整函数可以轻易转换为一组线性约束，如附录所示。最后， $l^k(j)$ 仅表示自车在规划步 j 预计行驶的车道的内部表示。

安全约束： 安全约束 (18) 确保自车在规划时刻 j，于其预计行驶的车道（$l^k(j)$）上，与最近的车辆 i 保持最小安全距离（$L_i^s(j)$）。我们借用了 [24], 中对这一安全距离的定义；该文作者依据《维也纳道路交通公约》中的条款进行了形式化表述，该条款指出：“车辆 *[...]* 应保持足够的距离 *[...]* 以避免前车突然减速或停车时发生碰撞。”此外，如附录所示，应用大M法并引入辅助变量，可将绝对值约束分解为线性约束。

注释 *3*： 所提出的建模方法能够适用于任意时刻的任意数量车道 k。这意味着，在任何时刻，可供行驶的车道数量无论增加还是减少，所提出的建模方法仍然适用。这一点十分重要，因为高速公路上经常会因道路事故、道路施工、道路变窄等各种意外情况而导致部分车道封闭。

2) 计算复杂度降低： 本节详细介绍通过二进制变量重构优化问题、优化热启动技术以及惰性约束实现。这些方法相结合，能够降低 SLAS 模块的计算复杂度。

二进制变量： 第 III-B 节提出的建模方法具有较高的计算复杂度（最坏情况下的计算时间为 $\sim 2s$ ——缓慢移动的交通流阻塞所有车道），这是由整数决策变量导致其成为混合整数优化问题 [25]。为避免计算负担过重，我们使用二进制变量替代整数变量，对该问题进行重构，如下所示：

$$\forall i \in \mathbb{L}(k), \forall j \in \mathbb{Z}_{[1,H]} : \tilde{\mathcal{L}}^k(i,j) \in \{0,1\} \tag{19}$$

其中， $\tilde{\mathcal{L}}^k(i,\ j)$ 表示修改后的目标车道变量，其索引分别为车道（i）和规划步（j）； $\tilde{\mathcal{L}}^k(a,\ b) = 1$ 表示在规划步 $b \in \mathbb{Z}_{[1,H]}$ 选择车道 $a \in \mathbb{L}$ 作为目标车道。随后，SLAS 公式中第 III-B 节的部分约束

修改如下：

$$(11) \rightarrow \tilde{\mathcal{L}}^k(l_0(k),0)=1 \tag{20}$$

$$(15) \rightarrow \sum_{i\in\mathbb{L}(k)} \tilde{\mathcal{L}}^k(i,j)=1, \forall j\in\mathbb{Z}_{[1,H]} \tag{21}$$

$$(16) \rightarrow \tilde{\mathcal{L}}^k(a,j)=1 \implies \sum_{b\in\mathbb{B}(k)} \tilde{\mathcal{L}}^k(b,j)=1, \tag{22}$$

$$\mathbb{B}(k)=\mathbb{Z}_{[a-1,a+1]}\cap\mathbb{L}(k)$$

$$(18) \rightarrow \min_{i\in\mathbb{A}(k)} \{|\hat{s}_i^k(j)-s^k(j)|\} \geq \tilde{L}_i^s(j), \tag{23}$$

$$\mathbb{A}(k)=\{a\in\mathbb{O}(k)\mid \tilde{\mathcal{L}}^k(l_a(k),j)=1\}.$$

此处，式（20）用于初始化目标车道，式（21）将任一规划步的目标车道限制在可用车道集合内，式（22）将连续规划步之间的变道限制为相邻车道之间的切换，式（23）则表示增广安全约束。式（22）中的蕴含关系（$\implies$）可以轻松转换为线性约束（见附录）。增广最小安全距离（L_i^s（j））将式（17）中的变道操作执行时间（N）纳入以下统一安全约束：˜

$$\tilde{L}_i^s(j)=L_i^s(j)+\gamma_d(\delta^k(j))\cdot(v^k(j)\cdot N\cdot T_s) \tag{24}$$

$$\gamma_d(\delta^k(j))=\gamma_4\cdot\frac{2|\delta^k(j)|}{L_l} \tag{25}$$

$$\delta^k(j)=\min\left\{\delta(k)+\frac{L_l\cdot j}{N},\frac{L_l}{2}\right\} \tag{26}$$

其中，L_l 为车道宽度（见图1），δ（k）表示时间步 k时自车相对于**先前** 目标车道边界的有符号横向偏差，γ_d（δ^k（j））表示相对于先前目标车道（$\mathcal{L}$（$k-1$））的偏差动态代价。此外，在代价函数（8）中，我们取$\mathcal{L}^k$（0）$=\mathcal{L}$（$k-1$）。引入这些代价是为了防止车辆出现蛇形行驶（在车道切换完成前取消切换）的行为，除非出于安全考虑确有必要。

备注 *4*：由于自车一旦越过车道边界，就被视为已完成变道，因此，为了保持自车横向位移相对于 $\gamma_d(\delta^k(j))$ 的连续性，偏差 $\delta^k(j)$ 是相对于车道边界而非目标车道中心来计算的。具体而言，若自车已越过先前目标车道的边界，则 $\delta^k(j)>0$；否则为 0。这一点十分重要，因为变道完成时，$\gamma_d(\delta^k(j))$ 的不连续可能导致不可行。

备注 *5*：通过硬约束抑制转向摆动行为，但并未将其完全消除，因为有时必须通过这种行为来应对环境不可预测性。这种反应式策略是我们方法的一项显著特点，使算法能够主动“改变主意”，以应对环境中发生的意外情况，避免危及安全性。

优化热启动：为帮助优化器找到初始可行解，我们将上一个时间步的解作为参考。形式上，$\{v^{k-1}(2)\cdots,v^{k-1}(H)\ \mathcal{L}^{k-1}(2)\cdots,\mathcal{L}^k(H)\}$被作为$\{v^k(1)\cdots,v^k(H-1)\ \mathcal{L}^k(1)\cdots,\mathcal{L}^k(H-1)\}$。这并不意味着时间步 $k-1$ 的解在时间步 k会完全保持不变，这是由于存在未建模扰动；但提供该参考解有助于优化器在参考解附近找到初始可行解。这一观察基于以下前提：鉴于采样时间并不太长，且道路交通参与者的预测行为不会发生显著变化，较长规划时域内的解在不同时间步之间预计不会发生显著变化。

还值得指出的是，此处的首要目标是快速找到满足安全约束和执行器限制的可行解，并在此基础上递归改进，而不是过度迭代以求得全局最优解。在我们的实验中，我们观察到，次优解在定性上与最优解并无显著差异。因此，我们采用割平面法进行优化 [26],：该方法首先利用我们提供的参考值寻找可行解，然后递归更新该解，直到找到全局最小解或达到时间限制。

惰性约束：为进一步提升计算效率，我们对换道约束（22）采用惰性实现。在实验中我们发现，不包含换道约束（22）时，找到可行解的速度比包含这些约束时快几个数量级（$\sim 10\times$），因此我们决定对这些约束采用惰性实现。采用惰性实现时， [27], 求解器会先在不包含这些约束的情况下找到一组可行解，然后根据惰性约束，通过约简问题判定这些解是否可行。

3）可行性：根据 [20], 中提出的类似论证，可以相对直接地证明该问题具有递归可行性，也就是说，如果该优化问题初始时可行，就会始终保持可行；其平凡解是与前车保持相同速度且不变道。

IV. 结果

本节将详细介绍实验设置，展示 SLAS 的性能，并报告定性和定量的对比分析。对比分析中的基线设为：扩展 A*（EA*）[2], MOBIL [3],以及不采用换道模型（不变道）。

A. 实验设置

图 3 所示的实现设置由 CARLA 仿真器（版本 0.9.11） [28],SLAS 模块（第 III-B 节）以及规划器和控制器模块[18]组成。为求解 SLAS 的优化问题，我们使用 Gurobi 优化器（版本 9.1.1 ） [29]。仿真在一台配备 Intel Xeon(R) CPU E5-2643 v4 @ 3.40GHz × 12 和 NVIDIA Titan XP、运行 Ubuntu 20.04 LTS 的计算机上进行。平均而言，每个优化步骤所需的时间为 $\sim 0.096s$，而优化器的最大时间限制设置为 $0.2s$，表明其具有很强的实时应用潜力。

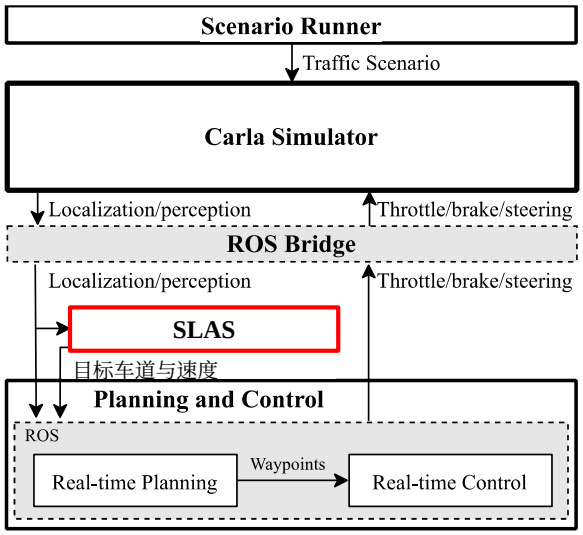

图 3. 仿真设置。场景运行器为 CARLA 仿真器设置场景，随后仿真器通过 ROSbridge 节点与 SLAS 以及规划和控制 ROS（机器人操作系统）节点进行通信。

B. 案例研究

图 4 展示了用于比较分析的测试用例场景。该场景由一段包含四条车道的高速公路路段组成，其中最右侧车道专供汇入车辆使用。自车初始化为跟随车道 1 中的一辆低速行驶车辆，其右侧车道 2 中的交通流速度更慢。因此，为了最小化行驶时间，自车唯一的选择是向左变道至车道 0，因为该车道中的交通流速度更快且车头时距更大。当自车驶入车道 0 并超过车道 1 中的低速车辆后，它有两种选择：要么继续在车道 0 中行驶，在接近前车之前不作任何变道决策；要么主动利用车道 2 中的间隙，预判车道 1 和车道 2 中的交通拥堵，并变道至车道 3。一名具有前瞻性的战略决策者会选择后者，并主动作出决策，以获得更大的整体收益。

对比分析的评估指标包括：行驶时间、横向位移、车头时距以及与最近车辆之间的距离。关于仿真参数，仿真步长设置为 $0.05s$（仿真频率为 $20Hz$）；车道0、1和2中车辆的速度分别设置为8、5和 2 m/s，而限速 V_l 设置为 $15m/s$；高速公路路段的长度设置为 $350m$，车道之间的宽度设置为 $3.5m$；传感器可视范围设置为 $R_v = 50m$。SLAS的参数设置如下：$T_s = 0.4s$、$H = 40$、$N = 3$、$A_{min} = -5m/s^2$、$A_{max} = 3.5m/s^2$、$\gamma_1 = 1$、$\gamma_2 = 0.1$ 和$\gamma_3 = 0.01$。通过调整这些参数的值，可以使该算法表现出激进或保守的驾驶行为。

1）行驶时间：图5左侧的图展示了四种算法的行驶时间随纵向位移变化的情况。从图中可以看出，与其他方法相比，我们的方法（SLAS）能够保持较短的总体行驶时间。具体而言，SLAS 的表现优于 EA*、MOBIL 和不变道方法，完成仿真场景所需时间分别缩短了 12.72%、23.52% 和 54.34%。这表明，我们的方法具有前瞻性，能够弥补其表面上的保守性；这种保守性源于需要保障乘客舒适度。

2）横向位移：为识别四种方法在变道行为上的差异，图5中央图突出显示了仿真过程中横向位移与纵向位移之间的关系。在图中，横向位移为 0 对应车道 0 的中心，而后续每条车道的中心均相距 $3.5m$ 。比较四种算法的表现可以看出，SLAS 和 EA* 的表现较为相似，这是主动决策的结果。相比之下，由于 MOBIL 只评估切换至相邻车道的优势，因此无法发现主动切换至车道 2 的益处。这解释了为何在纵向位移约为 130 [m] 时，EA* 和 SLAS 开始在行驶时间方面（左图）优于 MOBIL。

至于 SLAS 与 EA*的直接比较，从这张中央图可以明显看出采用速度建议系统的优势。得益于速度控制，SLAS 能够持续保持更大的车头时距（右图），而无需在过于接近前车时大幅制动。这使横向位移曲线更加平滑，车辆能够以最小加加速度完成变道（定量分析见 IV-C节），并取得更好的整体时间表现（左图）。

3）车头时距：图5右侧的图展示了仿真过程中自车保持的车头时距。正如前文所述，MOBIL 会跟随前车行驶，在找到相邻车道中足够的空间以执行变道操作之前，保持较短的车头时距。另一方面，EA* 和 SLAS 的车头时距轨迹相近，但 SLAS 始终保持更大的车头时距，并且在 EA*之前达到最大值。定量而言，与 EA*、MOBIL 和不变道方案相比，SLAS 平均分别保持多出 9.43%、36.57% 和 113.17% 的车头时距。SLAS 的出色表现可归因于其纳入了安全保证，同时兼顾乘客舒适度。

4）与最近车辆的距离：最后，我们比较了仿真过程中自车与最近车辆之间保持的距离。平均而言，与 EA*、MOBIL 和不变道方案相比，SLAS 保持的距离分别多出 9.28%、32.01% 和 22.84%。这些数据充分证明了我们的方法的优势，这得益于其对较长规划时域的考量以及速度控制。

C. 蒙特卡洛仿真

为展示三种方法（SLAS、EA* 和MOBIL）的长期性能，我们针对交通参与者初始位置随机化（在 $8m$范围内）且速度随机化（在

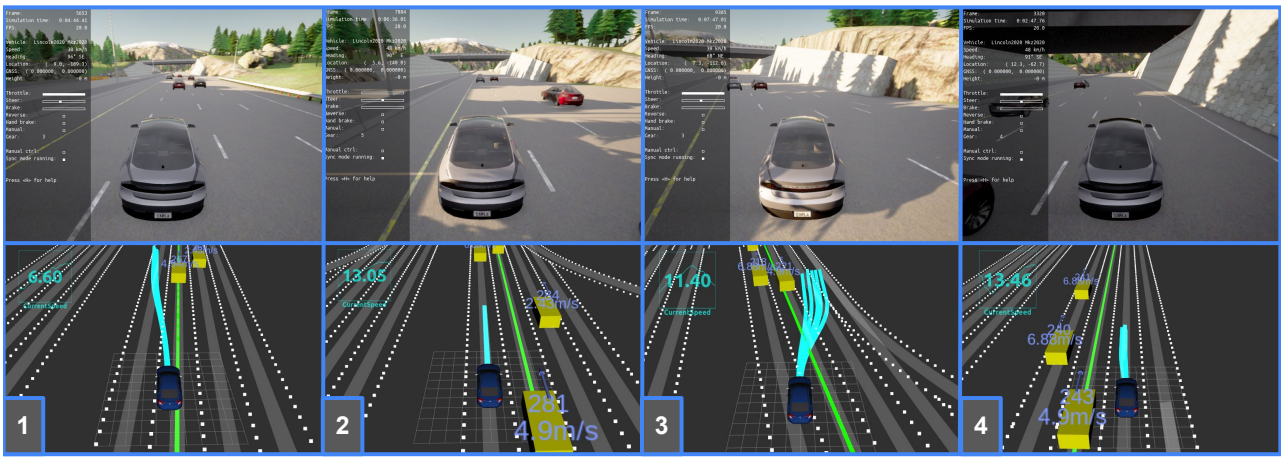

图4。三车道测试场景：车道0（左侧）、车道1（中间）和车道2（右侧）。图中以编号帧展示了仿真过程中自车的预期运动。最右侧车道（车道3）专供汇入车流使用，因此未在我们的仿真中使用。

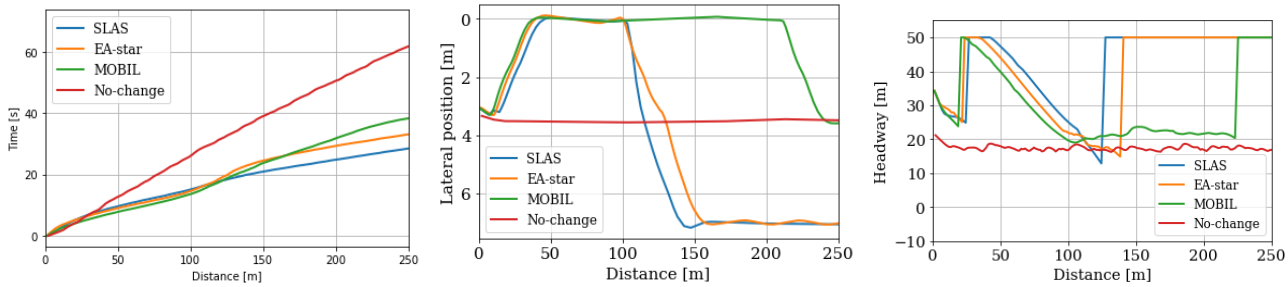

图5。左：行驶时间比较。中：车道选择（横向位置）比较。车道0（左侧）、车道1（中间）和车道2（右侧）的中心线分别具有固定的横向位移 $0m$、 $3.5m$ 和 $7m$ 。右：车头时距比较。在没有前车的情况下，车头时距受 $50m$的可视范围限制。

模型	比较 Time	制动	制动 Jerk	Thr.	Thr. Jerk	Ang. Acc.	Ang. Jerk
			平均值				
SLAS	27.84	-0.46	-0.45	0.74	0.69	2.21	5.04
EAstar	27.23	-0.56	-0.47	0.83	0.77	2.83	6.69
MOBIL	28.06	-0.62	-0.49	0.86	0.79	2.43	5.62
			标准差				
SLAS	1.77	0.25	0.09	0.08	0.15	0.73	1.89
EAstar	2.85	0.38	0.15	0.11	0.21	1.48	3.93
MOBIL	3.82	0.42	0.18	0.11	0.23	0.77	2.05

表 I 蒙特卡洛模拟结果

分别为8、5和 2 m/s 的范围内，并随机分配至三条车道）的场景进行了一系列蒙特卡洛仿真。50次仿真的结果如表I所示。

在此表中，各列代表不同的评估指标，各行对应三种算法，绿色标出的数值表示相应评估指标的最佳结果。表中的评估指标从左至右依次为完成时间（s）、制动（$\mathbb{R}_{[-1,0]}$）、制动加加速度（$\mathbb{R}_{[-1,0]}$）、油门（$\mathbb{R}_{[0,1]}$）、油门加加速度（$\mathbb{R}_{[0,1]}$）、角加速度（$^\circ/s^2$）和角加加速度（$^\circ/s^3$）。除完成时间外，其余指标均基于传递给车辆执行器的指令（图 2），用于对乘客舒适度建模。就平均性能而言，在乘客舒适度方面，SLAS 的表现远胜其他方法，因为它在设计中明确考虑了舒适性。不过，与 EA* 相比，它在行驶时间方面的表现略有下降，这是因为 SLAS 力求在缩短行驶时间与提升乘客舒适度之间取得平衡。与其他方法相比，SLAS 在每项评估指标上的标准差也都是最低的，这表明其长期性能具有一致性。

五、结论

我们提出了一种面向多车道高速公路机动场景的新型行为规划模块，该模块可输出战略层面的目标车道和参考速度指令，并将其集成到先进的运动规划与控制框架中。我们将该方法构建为一种滚动时域混合整数优化问题，旨在兼顾乘客舒适度，并在较长规划时域内最小化行驶时间。为减轻计算负担过重的问题，我们通过以二进制变量取代整数变量来重新构建该问题，并进一步进行多项改进，以利于数值计算。我们还进行了详细的对比分析，以展示该方法在 CARLA 仿真器中的性能。未来，我们计划在感知、定位和预测模块中纳入各种延迟和不确定性度量，以评估该方法的鲁棒性。

参考文献

[1] G. Liu、S. Chen、Z. Zeng、H. Cui、Y. Fang、D. Gu、Z. Yin 和 Z. Wang，“极其严重道路事故的风险因素：中国道路交通事故统计年报的结果”，PLoS One，第 13 卷，第 8 期，页码 e0201587，2018 年。[2] S. Bae、D. Isele、K. Fujimura 和 S. J. Moura，“存在动态障碍物的高速公路风险感知车道选择”，载于 2021 年 IEEE智能车辆研讨会（IV），IEEE，2021 年，页码 652–659。[3] A. Kesting、M. Treiber 和 D. Helbing，“用于跟驰模型的通用换道模型 MOBIL”，《交通研究记录》，第 1999 卷，第 1 期，页码 86–94，2007 年。 [在线]。可访问：https://doi.org/10 3141 1999 10 。/ - Y. Chen、C. Hu 和 J. Wang，“自动[4]驾驶车辆换道策略的速度预测运动规划与复合非线性反馈跟踪控制”，IEEE《智能车辆汇刊》，第 5 卷，第 1 期，页码 63–74，2019 年。[5] W. Khaksar、K. S. M. Sahari 和 T. S. Hong，“基于采样的运动规划算法在自动驾驶车辆导航中的应用”，《自动驾驶车辆》，第 735 卷，2016 年。[6] Y. Kuwata、J. Teo、G. Fiore、S. Karaman、E. Frazzoli 和 J. P. How，“实时运动规划及其在自动驾驶城市驾驶中的应用”，IEEE《控制系统技术汇刊》，第 17 卷，第 5 期，页码 1105–1118，2009 年。[7] M. Mukadam、A. Cosgun、A. Nakhaei 和 K. Fujimura，“基于深度强化学习的变道战术决策”，2017 年。[8] C.-J. Hoel、K. Wolff 和 L. Laine，“利用深度强化学习进行自动车速与换道决策”，载于 2018 年第 21 届智能交通系统国际会议（ITSC），IEEE，2018 年，页码 2148–2155。[9] M. Kaushik、V. Prasad、K. M. Krishna 和 B. Ravindran，“利用深度强化学习在模拟高速公路驾驶中执行超车操作”，载于 2018 年 IEEE智能车辆研讨会（IV），2018 年，页码 1885–1890。[10] J. Yang、A. Nakhaei、D. Isele、K. Fujimura 和 H. Zha，“CM3：协作式多目标、多阶段、多智能体强化学习”，载于国际学习表征会议，2020 年。[11] D. M. Saxena、S. Bae、A. Nakhaei、K. Fujimura 和 M. Likhachev，“利用无模型强化学习在密集交通中驾驶”，载于 2020 年 IEEE机器人与自动化国际会议（ICRA），IEEE，2020 年，页码 5385–5392。[12] S. Glaser、B. Vanholme、S. Mammar、D. Gruyer 和 L. Nouveli`ere，“考虑交通与驾驶员交互的真实道路高度自动化车辆基于机动的轨迹规划”，IEEE《智能交通系统汇刊》，第 11 卷，第 3 期，页码 589–606，2010 年。[13] G. Franze 和 W. Lucia，“动态环境下自动驾驶车辆的滚动时域控制策略”，IEEE《控制系统技术汇刊》，第 24 卷，第 2 期，页码 695–702，2015 年。[14] P. Falcone、F. Borrelli、J. Asgari、H. E. Tseng 和 D. Hrovat，“自动驾驶车辆系统的预测式主动转向控制”，IEEE《控制系统技术汇刊》，第 15 卷，第 3 期，页码 566–580，2007 年。[15] C. Liu 、 S. Lee 、 S. Varnhagen 和 H. E. Tseng，“利用模型预测控制进行自动驾驶车辆路径规划”，载于 2017 年 IEEE智能车辆研讨会（IV），IEEE，2017 年，页码 174–179。[16] G. Schildbach 和 F. Borrelli，“高速公路换道辅助的场景模型预测控制”，载于 2015 年 IEEE智能车辆研讨会（IV），IEEE，2015 年，页码 611–616。[17] B. Paden、M. ˇC´ap、S. Z. Yong、D. Yershov 和 E. Frazzoli，“城市自动驾驶车辆运动规划与控制技术综述”，IEEE《智能车辆汇刊》，第 1 卷，第 1 期，页码 33–55，2016 年。[18] S. Bae、D. Saxena、A. Nakhaei、C. Choi、K. Fujimura 和 S. Moura，“基于循环神经网络模型预测控制的密集交通协作感知换道操作”，载于 2020 年美国控制会议（ACC），IEEE，2020 年，页码 1209–1216。[19] R. Attia、R. Orjuela 和 M. Basset，“自动驾驶车辆引导的纵向与横向综合控制”，《车辆系统动力学》，第 52 卷，第 2 期，页码 261–279，2014 年。[20] F. M. Tariq、N. Suriyarachchi、C. Mavridis 和 J. S. Baras，“双向混合交通场景下的自动驾驶车辆超车”，载于 2022 年美国控制会议（ACC），2022 年。[21] S. Lef`evre、D. Vasquez 和 C. Laugier，“智能车辆运动预测与风险评估综述”，《ROBOMECH期刊》，第 1 卷，第 1 期，页码 1–14，2014 年。[22] K. Okamoto、K. Berntorp 和 S. Di Cairano，《基于相似性的车辆运动预测》，载于*2017*年美国控制会议 *(ACC)*。IEEE，2017年，第303–308页。[23] A. Gupta、J. Johnson、L. Fei-Fei、S. Savarese 和 A. Alahi，《Social GAN：利用生成对抗网络生成社会可接受轨迹》，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第2255–2264页。[24] C. Pek、P. Zahn 和 M. Althoff，《基于形式化交通规则验证自动驾驶车辆变道操作的安全性》，载于*IEEE*智能车辆研讨会论文集，2017年。[25] A. Richards 和 J. How，《用于控制的混合整数规划》，载于*2005*年美国控制会议论文集，*2005*年。，2005年，第2676–2683页，第4卷。[26] H. Marchand、A. Martin、R. Weismantel 和 L. Wolsey，《整数规划与混合整数规划中的割平面》，离散应用数学，第123卷，第1–3期，第397–446页，2002年。[27] B. Cuteri、C. Dodaro、F. Ricca 和 P. Sch¨uller，《答案集编程求解中的约束、惰性约束或传播器：一项实证分析》，逻辑编程理论与实践，第17卷，第5–6期，第780–799页，2017年。[28] A. Dosovitskiy、G. Ros、F. Codevilla、A. Lopez 和 V. Koltun，《CARLA：开放式城市驾驶模拟器》，载于机器人学习会议。PMLR，2017年，第1–16页。[29] L. Gurobi Optimization，《Gurobi优化器参考手册》，2021年。[在线]。可访问：http://www.gurobi.com

附录

A. 向下取整约束

对于 $y \in \mathbb{Z}$ 和 $x \in \mathbb{R}$，约束 $y = \lfloor x \rfloor$ 可由以下线性约束表示：

$$y \leq x, \quad y + 1 \geq x + \epsilon$$

其中 $\epsilon > 0$ 表示可行性容差。

B. 蕴含约束

对于 a，$b \in \{0, 1\}$，约束 $(a = 1) \implies (b = 1)$ 可由以下线性约束表示：

$$b + M \cdot (1 - a) \geq 1 - \epsilon$$

其中，$M \gg 0$（大M）和 $\epsilon > 0$ 用于考虑数值误差（在我们的实现中设为 0.1）。

C. 绝对值约束

对于 Δs，$L_s \in \mathbb{R}$，约束 $|\Delta s| - L_s \geq 0$ 可表示为：

$$\Delta s \geq L_s \vee \Delta s \leq -L_s.$$

这还可以进一步推广，如我们的实现中所做的那样，以具有不同的前向和后向安全裕度：

$$\Delta s \geq L_s^f \vee \Delta s \leq -L_s^r$$

其中，L_s^f 和 L_s^r 分别为前向和后向安全裕度。这可以用以下线性约束表示：

$$\begin{aligned} \Delta s + M \cdot c - L_s^f &\geq 0 \\ -\Delta s + M \cdot (1 - c) - L_s^r &\geq 0 \end{aligned}$$

其中，$M \gg 0$（大M）和 $c \in \{0, 1\}$ 负责在这两个约束之间进行选择。