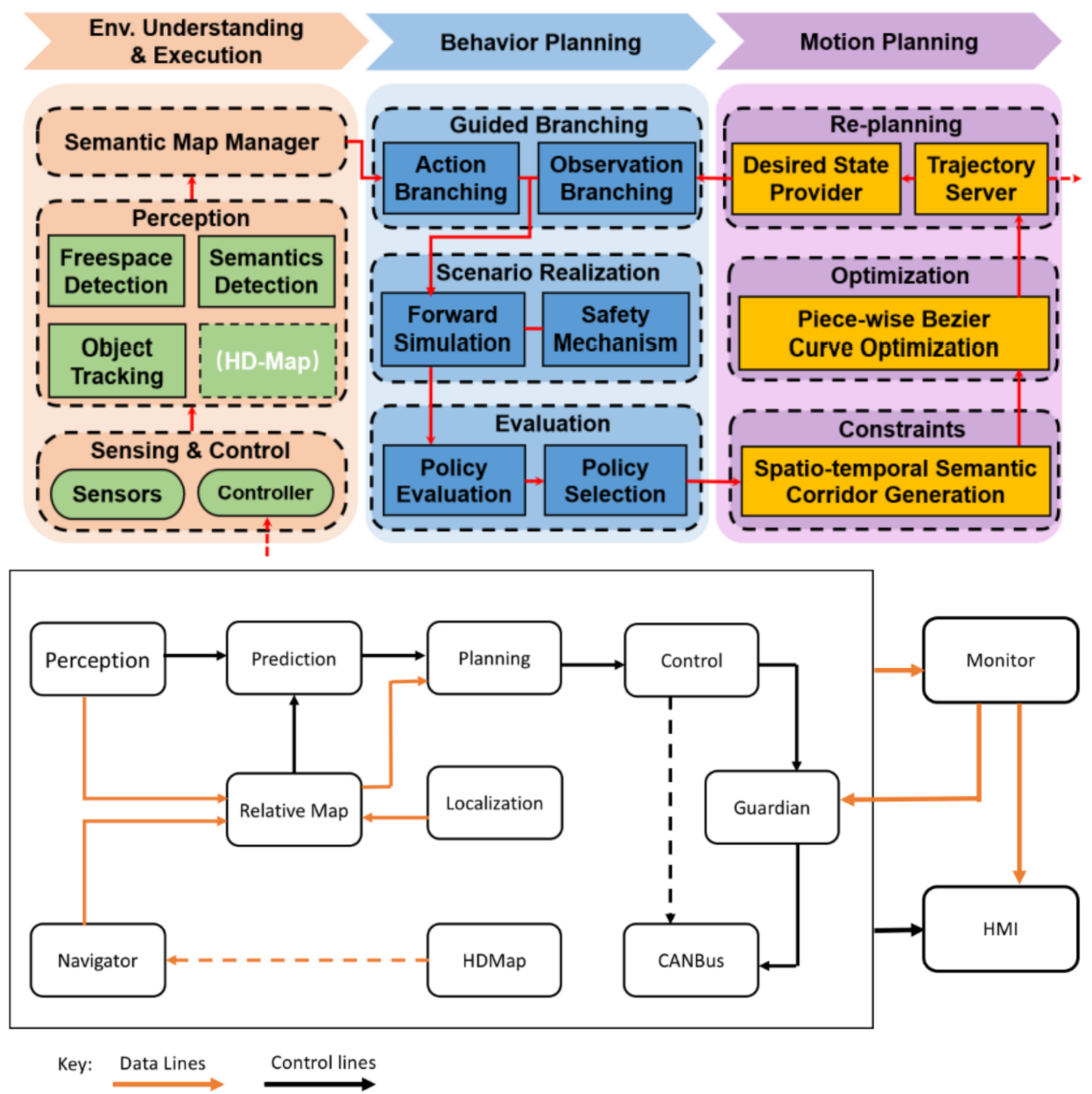


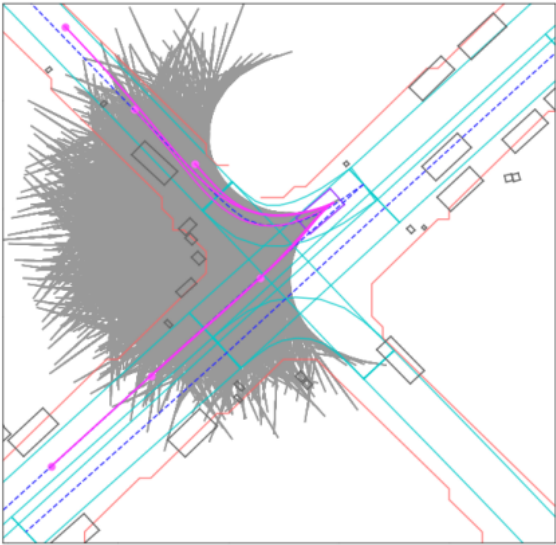
(2021)EPSILON: An Efficient Planning System for Automated Vehicles in Highly Interactive Environments
(EPSILON: 面向环境交互的高效规划系统)



(2020)Efficient Uncertainty-aware Decision-making for Automated Driving Using Guided Branching (基于导向分支的高效不确定性自动驾驶决策)

要解决的问题

- 1. 如何把环境中的其他交通参与者的潜在随机行为和感知不确定性考虑到决策系统中？
POMDP 模型框架有完整的理论基础来解决这类问题
- 2. 如何处理真实世界中维度爆炸的问题？
核心思想：利用特定领域的专家知识来引导行为空间和意图空间的分支，就是对问题进行剪枝处理



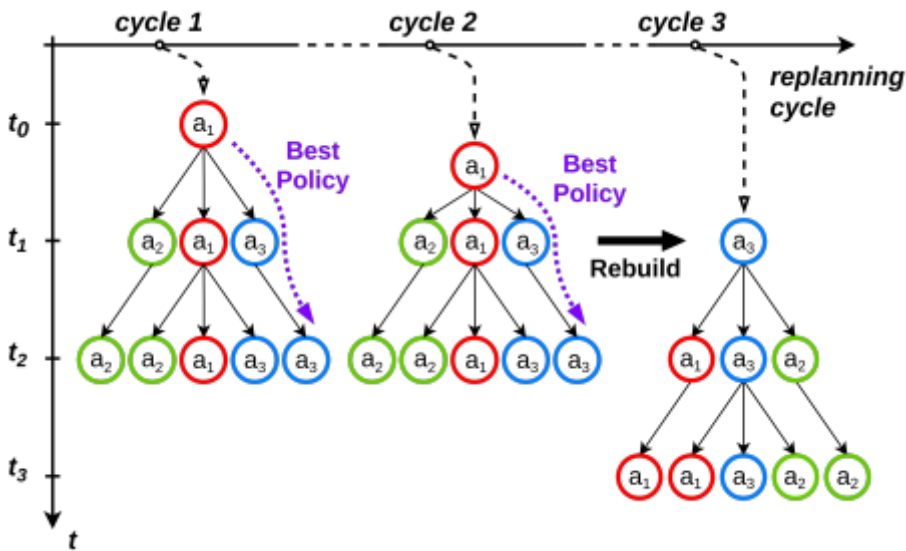
解决方案

提出了一种高效的**不确定性感知决策框架(EUDM)**，该框架可实时生成复杂驾驶环境下的长期横向和纵向行为。通过领域特定的闭环策略树(DCP-Tree)结构和条件聚焦分支(CFB)机制，将计算复杂度控制在合适的水平。

related works

不确定感知决策(EUDM)框架

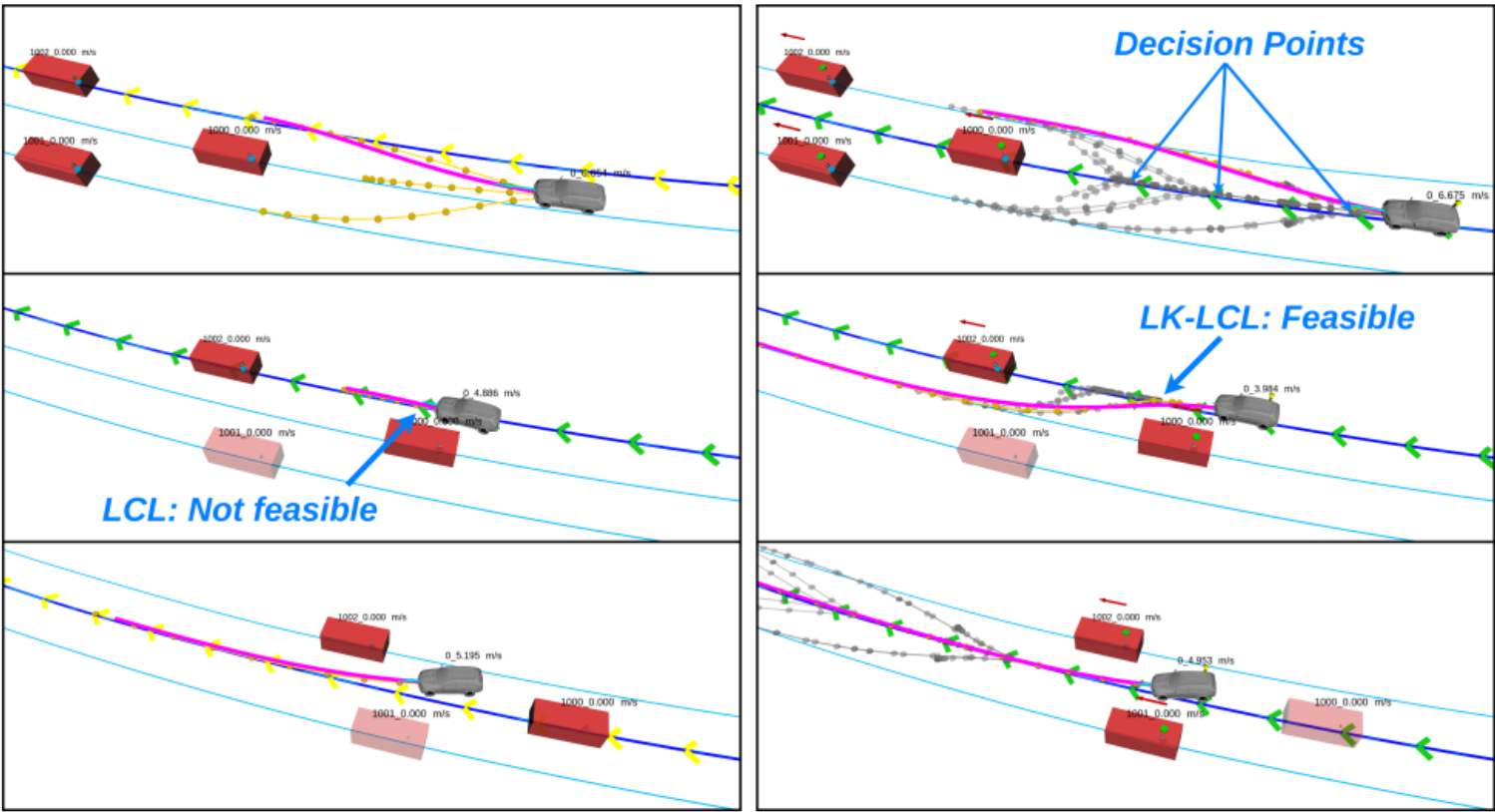
- 1. 使用特定于领域的闭环策略树(DCP-Tree)来构造语义层的操作空间
- 2. 策略树中的每个节点都是自车的有限范围语义行为
- 3. 从根节点到叶节点的每个轨迹代表了自车的语义动作序列



{a1,a2,a3}表示离散语义级操作可以理解为 (LK-LC-LC-LC. . .), (LK-LK-LC-LC. . .) and (LK-LK-LK-LC. . .)这里的LK,LC是加入了人类的先验知识做出的行为策略

横向作用定义为{LK, LCL, LCR}。纵向动作定义为{加速, 保持速度, 减速}

- 4. 每个轨迹以闭环模拟的形式进行评估，但自我行为允许在规划范围内更改
评估策略引入了RSS模型



(左图表示在规划周期内预测的行为是固定的，所以只有当自车超过左边红色车辆才进行换道;

右图表示提前对前方车辆做出判断，所以在未超过左边红色车之前就执行换道策略，考虑了未来时刻的交通参与者的行为变化)

预测和规划解耦带来的问题

1. 自车和他车的交互问题，这一个周期(8s)内自车和他车都在运动，而预测和规划分离却认为预测结果在一个周期内是固定不变的，灵活性不高
2. 考虑到车载传感器不完善的跟踪可能会导致**预测错误**，从而影响决策的安全性和通行效率
当前环境感知系统不太准确，导致预测模块(意图与轨迹)结果不准确，导致自车与目标博弈算法失败，使得决策系统更倾向于保守的规则(停车)。比如评估当前场景变化莫测，系统很难避障，系统就会选择停车，而如果与后车间距又太小，就会一直等着，直到前方可通行
3. 假设完美的感知，预测仍然会存在很多**不确定性**(例如鬼探头，行人过马路)

环境不确定性解决方案

引入**POMDP**模型框架来解决上述问题

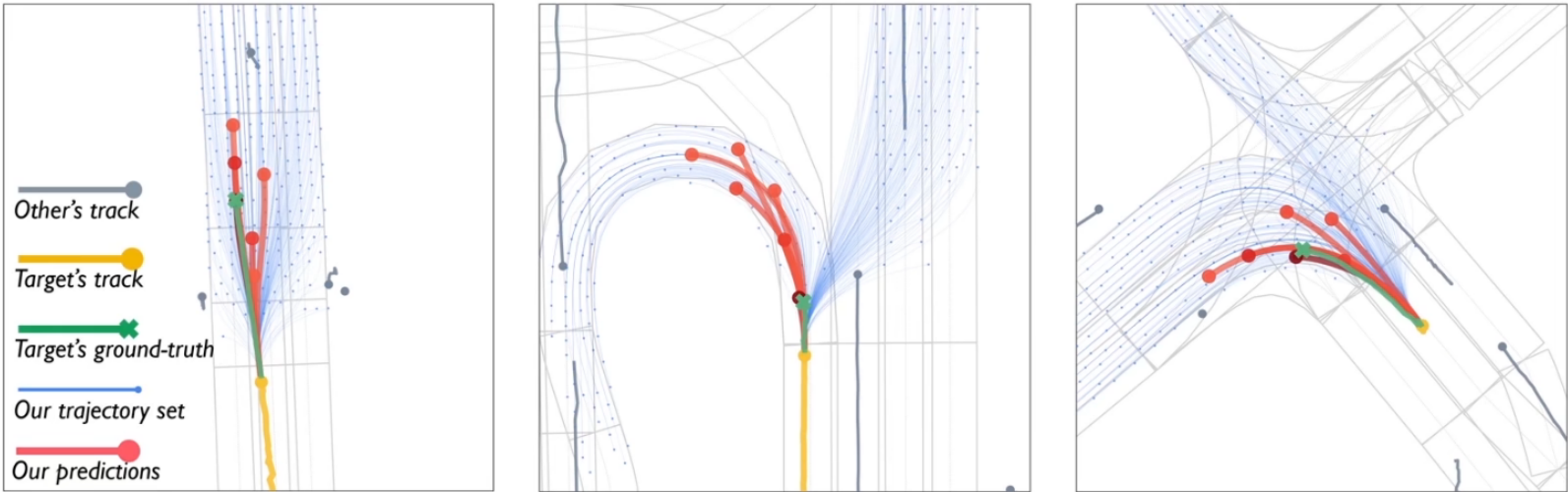
部分观测马尔科夫决策过程（Partial Observable Markov Decision Process, POMDP）能提供理论上最优的解决方案，即在考虑多车相互影响的不确定性前提下，提供最优的决策和规划。

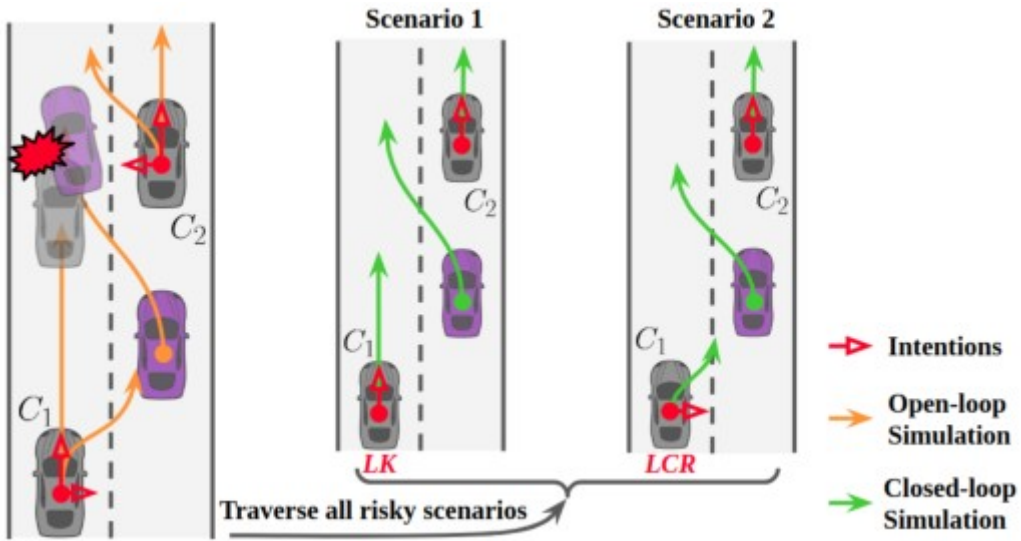
1. DCP决策树用于指导操作域中的分支，并基于之前的最佳策略更新语义级策略树
2. CFB机制用于识别附近车辆的危险隐藏意图
3. 通过闭环模拟对每个场景进行评估(考虑代理之间的交互)
4. 将所有场景都被输入到成本评估模块，并对风险分支机构应用有偏惩罚
5. 闭环正向仿真生成的一系列离散车辆状态(实验中分辨率为0.4s)

正向模拟闭环模拟是在考虑潜在交互的**同时向前推进多智能体系统的状态**。仿真模型应在仿真逼真度和推理效率之间实现良好的平衡。分别采用智能驾驶模型和纯追踪控制器作为纵向和横向仿真模型。

当前的方法是基于规则生成动作引导，也就是加入**人工和专家知识**，预制多种最优解以应对不同的状况，**POMDP**仅需要挑选对应目前状况最优的一个**policy**即可，但是但难以涵盖所有的状况，后面改用基于模仿学习的方式

6. 将状态序列传入运动规划器，以指导轨迹生成过程





决策算法发展

1. **sequential planning** 基于规则的方式， 模块界限分明,不包含学习类算法,决策主要依赖状态机或者部分融入优化cost function里.
2. **behavior-aware planning** 行为感知与规划融合,通过MDP,Game theory,POMDP, Reinforment learning等更靠近机器学习的算法.
3. **end-to-end planning**:一步到位,省略中间步骤,从传感器直接学习到自车行为

Apollo中的应用



Apollo6.0中首次引入了基于语义地图的模仿学习，通过大量的真实路测数据，模仿人类司机在一些特定场景下动态避障的能力，与已有基于优化的规划相结合，增强行车的安全性和舒适性。

(2019)Safe Trajectory Generation for Complex Urban Environments Using Spatio-Temporal Semantic Corridor

(基于时空语义廊道的复杂城市环境安全轨迹生成)

特色

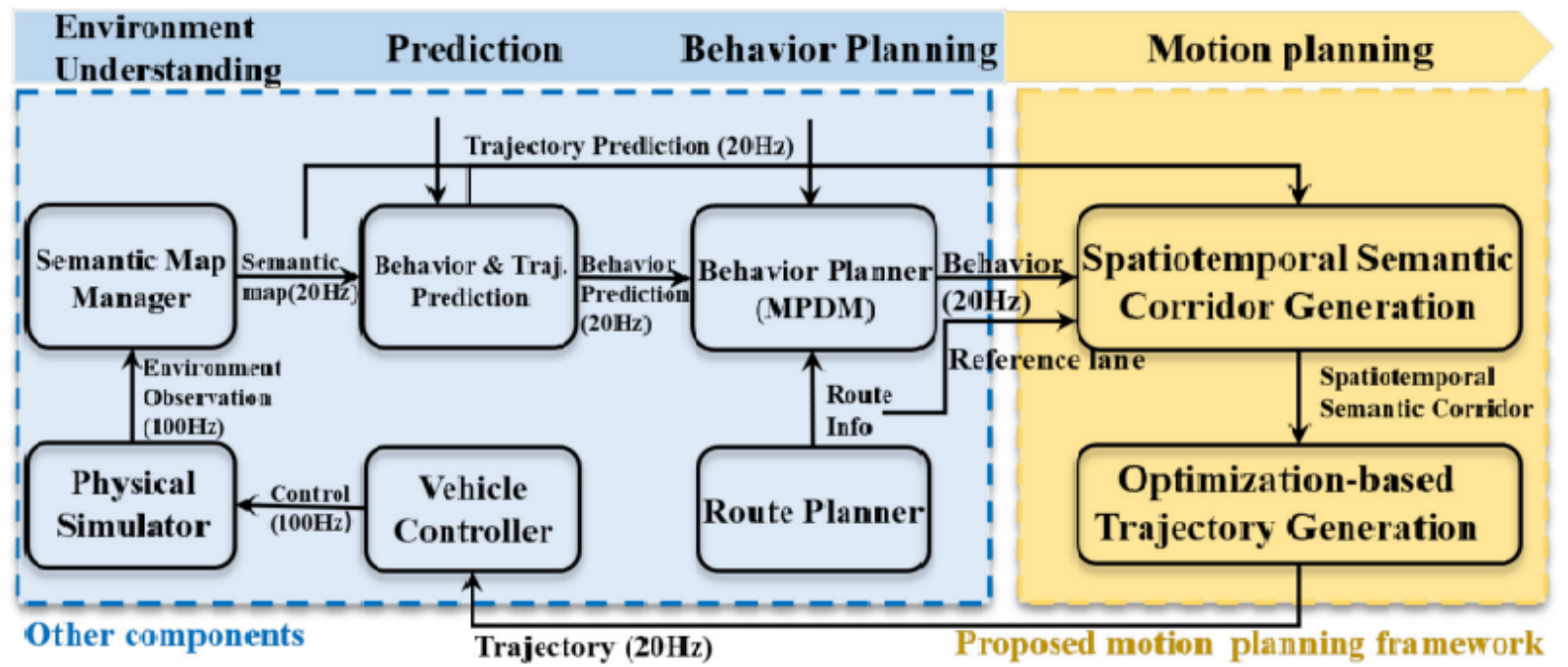
- 1. 在复杂城市环境中，针对不同的语义元素提供一种新的**统一时空语义走廊(ssc)**结构，为不同类型的语义元素提供一个抽象的层次。
- 2. SSC由一系列相互连接的无碰撞立方体组成，这些立方体由时空域的语义元素构成动态约束。
- 3. 将轨迹生成可以归结为一般的二次规划问题，在统一ssc框架约束的前提下，可以泛化任意场景的任意组合，最终转化为二次规划问题求解就OK了。
- 4. 另外轨迹采用分段贝塞尔曲线，利用的是其凸包性质

related works

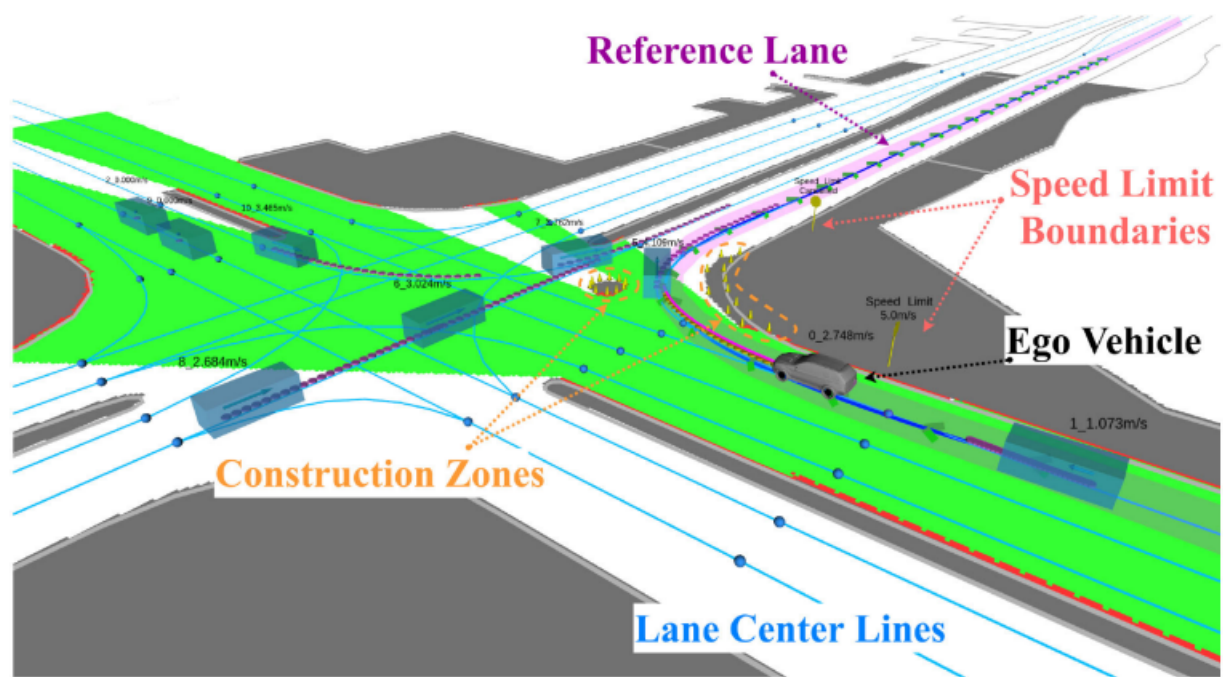
SSC(时空语义走廊)

1 构建SSC的要素

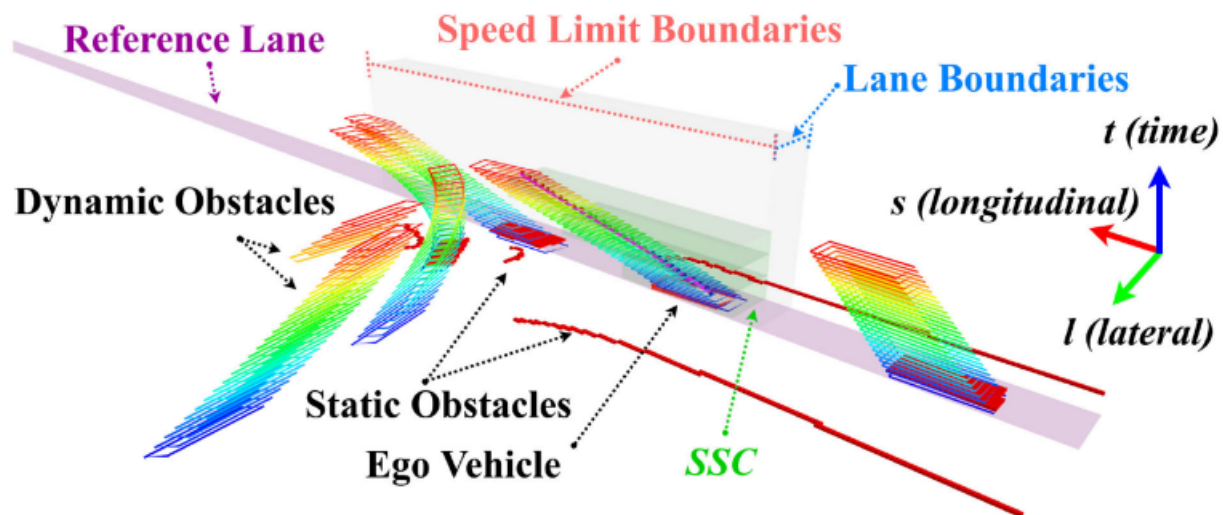
- 1. 语义元素组成的语义地图
- 2. 动态障碍物的预测轨迹(可选)
- 3. 正向模拟状态
 - 如果正向模拟状态已经包括其他障碍物车辆(如MPDM)的状态，预测模块可作为可选，这种情况下，我们可以将其他车辆的模拟状态作为预测轨迹，便于从行为规划层到运动规划层进行交互预期。当本文中为实现这一功能，仍采用轨迹预测进行泛化
- 4. 由Route Planner给出的参考车道



2 要素分类



(a) Complex urban driving environments



(b) Spatio-temporal semantic corridor (SSC)

- 由语义元素和Frenet框架构成一个slt的三维空间,将类障碍语义元素渲染到slt域后, 形成一个三维占栅格。
- 语义元素分为两类:类障碍语义元素和类约束语义元素。
 - **类障碍语义元素**:许多语义元素具有不允许进入slt域的某一部分的物理意义。
 - 静态障碍物: 跨越整个时间轴的障碍物
 - 动态障碍物: 可以看做是在时域内的一系列静止的障碍物
 - 红灯可看做是特定纵向位置的障碍物体
 - **类约束语义元素**:除类障碍语义元素外, 许多语义元素代表动态约束或时间约束。例如, 速度限制和停车标志可以被视为速度限制
- 还有一些语义元素会造成时间限制。例如, 在过车道时, 总变道时间不应过长。

2.1 语义边界

即对类似约束的语义元素提出一个统一的表示, 本质是表示某个语义元素开始和停止生效的位置

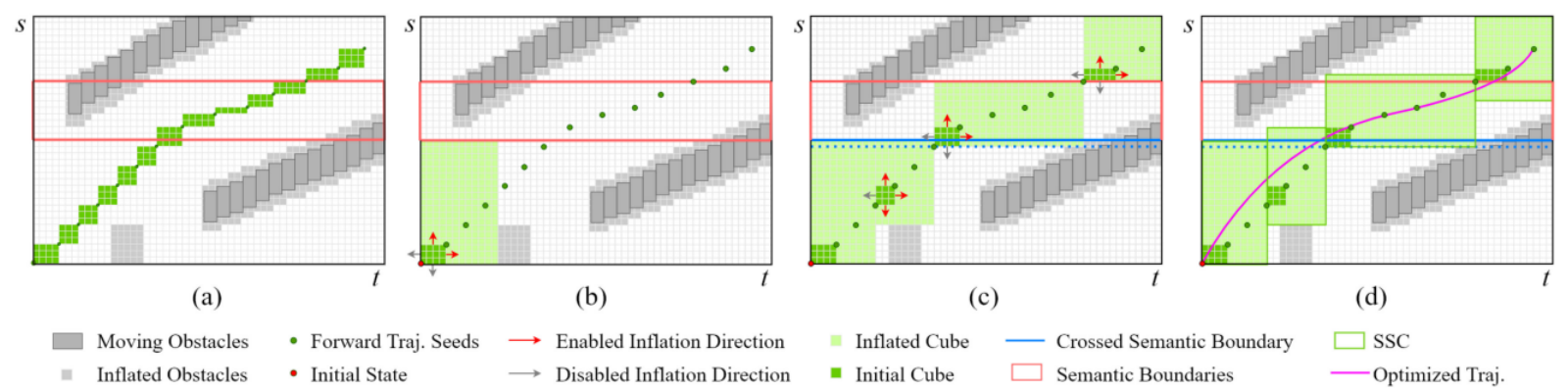
1. 速度限制可以看作是应用于纵向范围的速度约束
2. 变道时间约束作为当前车道横向范围的时间约束
3. 交通规则在约束中属于硬约束, 不可跨越, 像换到时间属于认为因素, 不可定量去描述

3 SSC生成

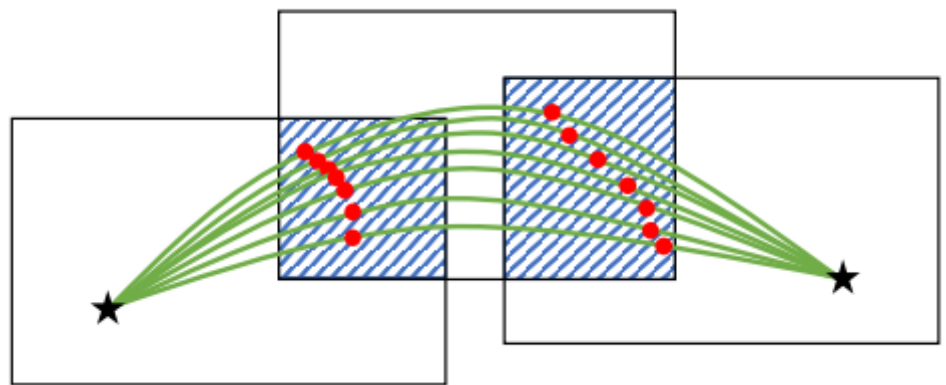
Algorithm 1: Semantic Corridor Generation.

- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
- 2: Initializes: seeds $\mathcal{S}^{seed} = \emptyset$;
- 3: $\mathcal{S}^{seed} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{des})$;
- 4: $\mathcal{C}^{infl} \leftarrow \text{CubeInflation}(\mathcal{S}^{seed}, \mathcal{B}, \mathcal{E})$;
- 5: $\mathcal{C}^{infl} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{infl}, \mathcal{B})$;
- 6: $\mathcal{C}^{final} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{infl}, \mathcal{E})$;

步骤:第3行: 生成粗略立方体; 第4行: 立方体膨胀; 第5步: 立方体膨胀限制; 第6步: 立方体松弛。

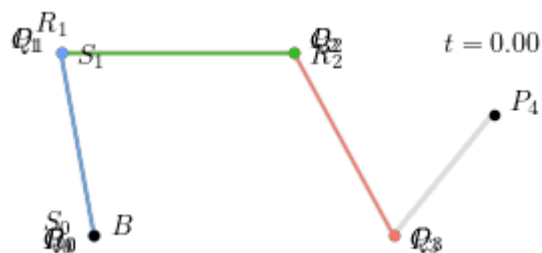


- (a) 绿色立方体为决策层给出的初始值，后面的轨迹生成以此为基础来做优化
 - 初始方块由两个初始点作为方块的顶点构成，二者间隙0.15s，假设30m/s，速度行驶，则二者相距4.5m
- (b) 对初始绿色立方体进行膨胀，膨胀到接触到障碍物(静态/动态的外扩)以及 **constraint (例如速度限制)**
- (c) 重复进行膨胀。但要注意拓展的方向部分需要省略，例如第三个立方体需要去掉向下和向左的膨胀，另外图中还有蓝色的线，是对速度约束的一个松弛变量
- (d) 在立方体膨胀之后，根据相关约束和自由空间产生立方体弛豫过程，在组合的立方体中利用分段贝塞尔曲线曲线拟合得到优化轨迹



轨迹生成的安全性和可行性保证

1 贝塞尔曲线简介



一般形式

$$C(u) = \sum_{i=0}^n B_{i,n}(u) \cdot P_i \quad u \in [0, 1]$$

即多项式

$$B_{i,n}(u) = C_n^i \cdot u^i (1-u)^{n-i} \\ = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

称为伯恩斯坦基函数

2 贝塞尔曲线 vs 多项式曲线

- 伯恩斯坦基函数与幂基函数的关系

$$C(u) = [B_{0,n}(u) \quad B_{1,n}(u) \quad \cdots \quad B_{n,n}(u)] \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{bmatrix}$$

$$\begin{aligned}
B_{i,n}(u) &= C_n^i \cdot u^i (1-u)^{n-i} \\
&= C_n^i \cdot u^i \sum_{k=0}^{n-i} (-1)^k C_{n-i}^k u^k \\
&= C_n^i \cdot \sum_{k=0}^{n-i} (-1)^k C_{n-i}^k u^{i+k} \\
&= C_n^i \cdot \sum_{k=i}^n (-1)^{k-i} C_{n-i}^{k-i} u^k \\
&= C_n^i \cdot \sum_{k=i}^n (-1)^{k-i} (-1)^{2i} C_{n-i}^{k-i} u^k \\
&= C_n^i \cdot \sum_{k=i}^n (-1)^{i+k} C_{n-i}^{k-i} u^k
\end{aligned}$$

$$\begin{aligned}
C_n^i \cdot C_{n-i}^{k-i} &= \frac{n!}{i!(n-i)!} \cdot \frac{(n-i)!}{(k-i)! \cdot ((n-i)-(k-i))!} \\
&= \frac{n!}{i!} \cdot \frac{1}{(k-i)! \cdot (n-k)!} \\
&= \frac{n!}{k!(n-k)!} \cdot \frac{k!}{i!(k-i)!} \\
&= C_n^k \cdot C_k^i
\end{aligned}$$

最终可以得到

$$\begin{aligned}
B_{i,n}(u) &= C_n^i \cdot u^i (1-u)^{n-i} = \sum_{k=i}^n (-1)^{i+k} C_n^k C_k^i \cdot u^k \\
C(u) &= \begin{bmatrix} 1 & u & \cdots & u^n \end{bmatrix} \begin{bmatrix} m_{00} & & & \\ m_{10} & m_{11} & & \\ \vdots & \vdots & \ddots & \\ m_{n0} & m_{n1} & \cdots & m_{nn} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{bmatrix}
\end{aligned}$$

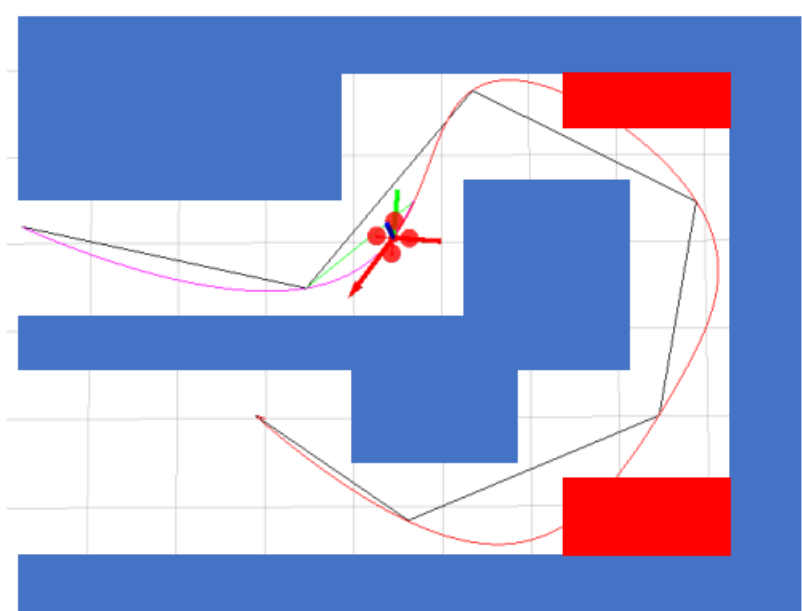
简单来说就是贝塞尔曲线能通过一个**M**矩阵转换为多项式曲线，即 $C(u) = M \cdot P(t)$

二者最大的不同

多项式曲线的系数并没有任何的物理意义，而贝塞尔曲线的系数，即公式中的 p_i 代表的是实际控制点。

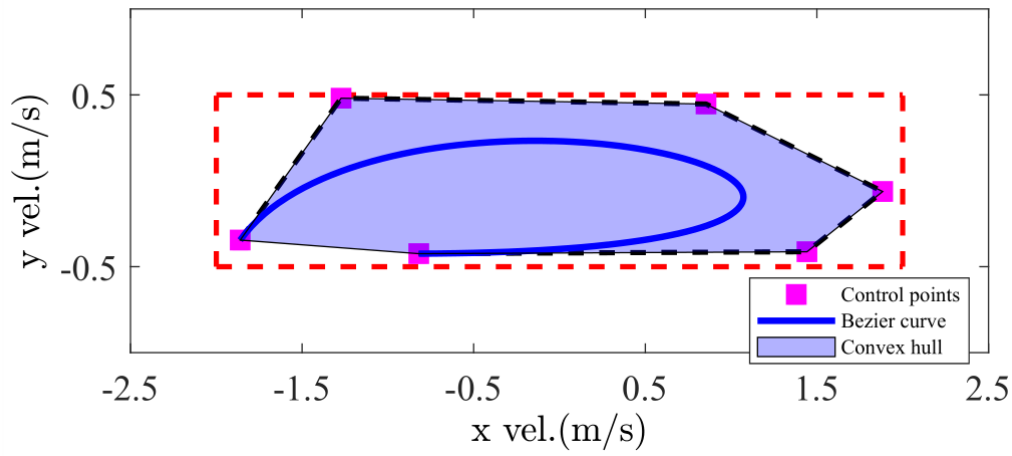
3 多项式曲线运用在轨迹规划中的不足

单条多项式曲线无法检测样本点之间的碰撞，因此无法提供对安全性和可行性的任何保证



4 贝塞尔曲线的特点

- 一定经过起点和终点
- 一定不经过控制点 ==> 凸包性质



3. m阶贝塞尔曲线 $B(t)$ 的(m-1)导数后的曲线 $B'(t)$ 仍是一条贝塞尔曲线, $B'(t)$ 的控制点系数满足 $n \cdot (C_{i+1} - C_i)$
4. 固定时间间隔[0,1]间取值,需要对变量做归一化处理

5 Piecewise Bezier Curve Representation

$$f_j^\sigma(t) = \begin{cases} \alpha_1 \cdot \sum_{i=0}^m p_i^1 \cdot b_m^i \left(\frac{t-t_0}{\alpha_1} \right), & t \in [t_0, t_1] \\ \alpha_2 \cdot \sum_{i=0}^m p_i^2 \cdot b_m^i \left(\frac{t-t_1}{\alpha_2} \right), & t \in [t_1, t_2] \\ \vdots & \vdots \\ \alpha_n \cdot \sum_{i=0}^m p_i^n \cdot b_m^i \left(\frac{t-t_{n-1}}{\alpha_n} \right), & t \in [t_{n-1}, t_n] \end{cases} \quad (6)$$

其中 $\sigma \in s, l$ 两个维度的曲线

代价函数

$$J_j = w_s \int_{t_{j-1}}^{t_j} \left(\frac{d^3 f^s(t)}{dt^3} \right)^2 dt + w_l \int_{t_{j-1}}^{t_j} \left(\frac{d^3 f^l(t)}{dt^3} \right)^2 dt \quad (7)$$

J_j 为第i段的cost function， 分别对横纵向求最小 Jerk

约束

1. 起点和终点约束

起点状态 $[\sigma_{t_0}^{(0)}, \sigma_{t_0}^{(1)}, \sigma_{t_0}^{(2)}]$; 终点状态 $[\sigma_{t_n}^{(0)}, \sigma_{t_n}^{(1)}, \sigma_{t_n}^{(2)}]$

$$\frac{d^k f_0^\sigma(t_0)}{dt^k} = \sigma_{t_0}^{(k)}, \quad \frac{d^k f_n^\sigma(t_n)}{dt^k} = \sigma_{t_n}^{(k)} \quad (8)$$

2. 连续性约束

$$\frac{d^k f_j^\sigma(t_j)}{dt^k} = \frac{d^k f_{j+1}^\sigma(t_j)}{dt^k} \quad (9)$$

3. 边界约束

$$\begin{aligned} \beta_{j,-}^{\sigma,(k)} &\leq \alpha_j^{1-k} \cdot q_{j,i}^{\sigma,(k)} \leq \beta_{j,+}^{\sigma,(k)}, \forall i \Rightarrow \beta_{j,-}^{\sigma,(k)} \leq \frac{d^k f_j^\sigma(t)}{dt^k} \leq \beta_{j,+}^{\sigma,(k)} \\ \frac{d^k f_j^\sigma(t_{j-1})}{dt^k} &= \alpha_j^{1-k} \cdot q_{j,0}^{\sigma,(k)}, \quad \frac{d^k f_j^\sigma(t_j)}{dt^k} = \alpha_j^{1-k} \cdot q_{j,m}^{\sigma,(k)} \\ q_{j,i}^{\sigma,(0)} &= p_i^j, q_{j,i}^{\sigma,(k)} = \frac{m!}{(m-k)!} \left(q_{j,i+1}^{\sigma,(k-1)} - q_{j,i}^{\sigma,(k-1)} \right) \end{aligned} \quad (10)$$

第一行是约束的最终表达式

第二三行为第一行中的参数解释, 若k=0, 表示立方体膨胀后的边界, 如果需要限制曲率或者更高阶导数的化可以对 k=2,3...的情况下进行约束

4. 限制最大加速度为 2m/s^2、最大减速度 3m/s^2