北京航空航天大學

Trajectory Planning for Autonomous Vehicles Using Hierarchical Reinforcement Learning

Kaleb Ben Naveed, Zhiqian Qiao and John M. Dolan

分享人：黄靖宜

日　期：2022/5/14

目录

CONTENTS

背景介绍

问题：

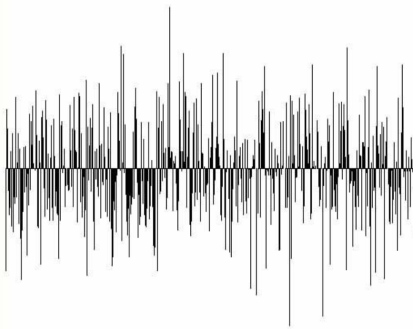
自动驾驶汽车的安全轨迹规划

特点：

操控规划复杂

行驶环境随机

感知系统嘈杂



背景介绍

轨迹规划方法：

启发式方法

Slot based method

Time-To-Collision (TTC)

机器学习方法

Imitation Learning

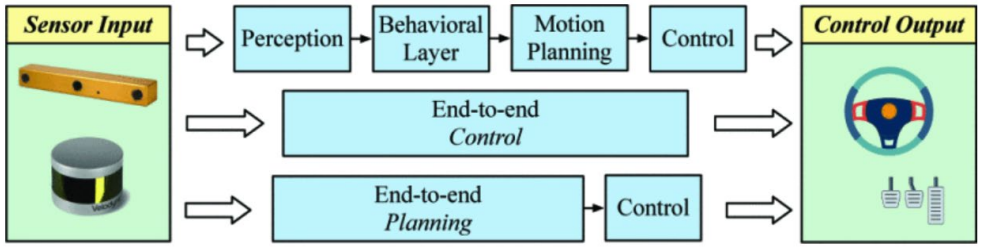
模仿学习

Reinforcement Learning

强化学习

[1] C. R. Baker and J. M. Dolan, "Traffic interaction in the urban challenge: Putting boss on its best behavior," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1752–1758.

[2] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.



背景介绍

本文贡献：

- **高层决策**：HRL框架的较高级别负责选择机动选项，可以是车道跟随/等待或车道变换；
- **规划平滑航点轨迹**：低级规划器基于高级选项，生成可变长度的航点轨迹，由PID控制器跟踪；
- **状态观测的历史**：我们使用具有状态观测历史的LSTM层来补偿观察噪声，并通过交互式驾驶条件改善学习；
- **提高样品效率**：我们使用混合奖励机制和奖励驱动探索，以提高样品效率和收敛时间。

预备知识

DDQN：

Q-Learning	通过计算Q值表，确定最优策略
Deep Q-Network	通过深度学习的方法，解决维数灾难
Double DQN	通过解耦目标Q值动作的选择和目标Q值的计算这两步，消除过度估计的问题

预备知识

分层强化学习：

HRL在多个层面上学习策略，因为元控制器Q1为下面的步骤生成子目标g，控制器Q2根据选择的子目标输出行动a，直到元控制器生成下一个子目标。

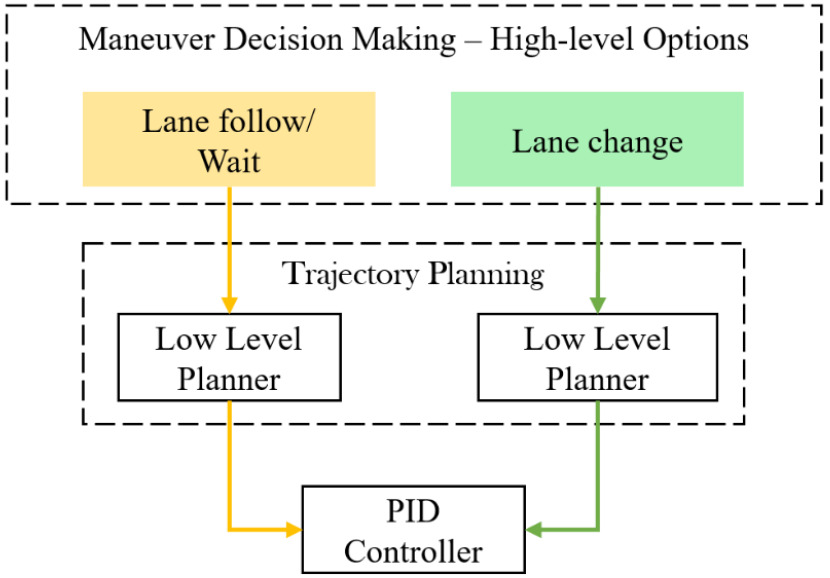
分层强化学习将复杂问题分解成若干子问题，通过分而治之的方法，逐个解决子问题从而最终解决一个复杂问题。

架构设计

等级结构和决策：

在提出的Robust-HRL框架中，文章使用三层结构进行决策和轨迹规划，具有两个全连接网络：一个用于高级决策，另一个用于低级轨迹规划。

最顶层负责从车道跟踪/等待或车道变换选项中选择高级机动选项。一旦进行了高级选择，信息就会传递给低级规划者，后者根据学习到的政策生成航点轨迹。之后，利用PID控制器进行轨迹跟踪。



架构设计

轨迹规划和航点生成：

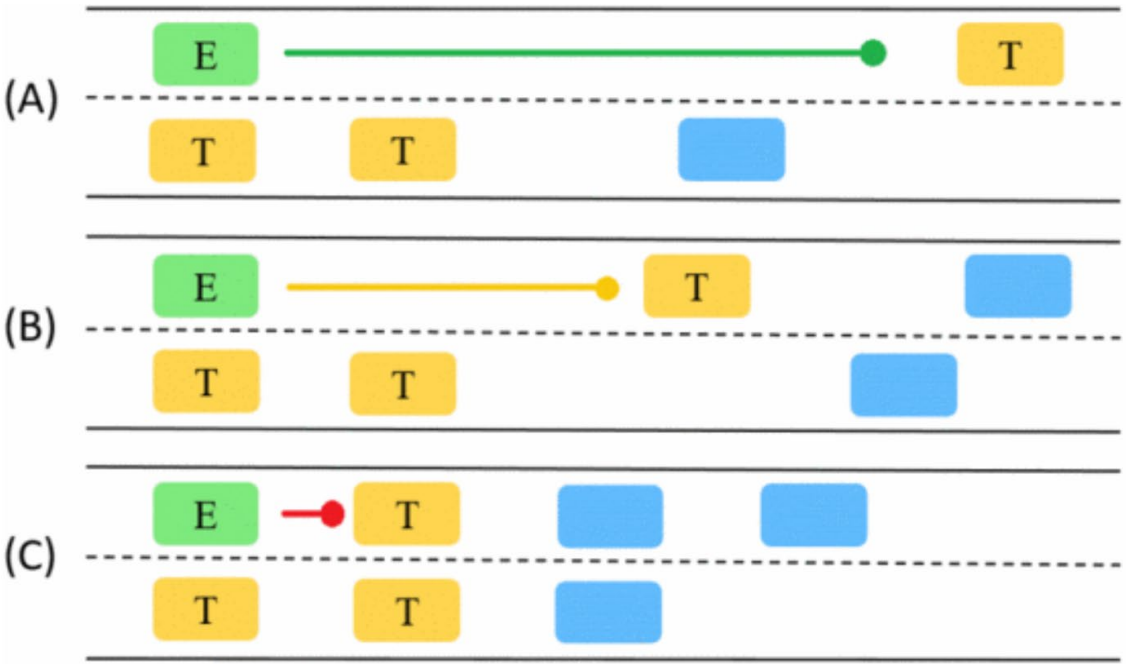
轨迹规划在分层框架的第二层实施。

- 选择高级选项后，低级轨迹规划器将从离散的航点选项中选择最终航点。
- 选择最终航点后，使用最大加速/减速约束计算ego car的目标速度，以确保平稳的子轨迹。
- 将目标速度和最终航点值提供给PID控制器，PID控制器又产生纵向和横向控制。
- 这些子轨迹完全形成了一个完整的轨迹，构成车道跟踪/等待和车道变换。

架构设计

车道跟踪/等待：

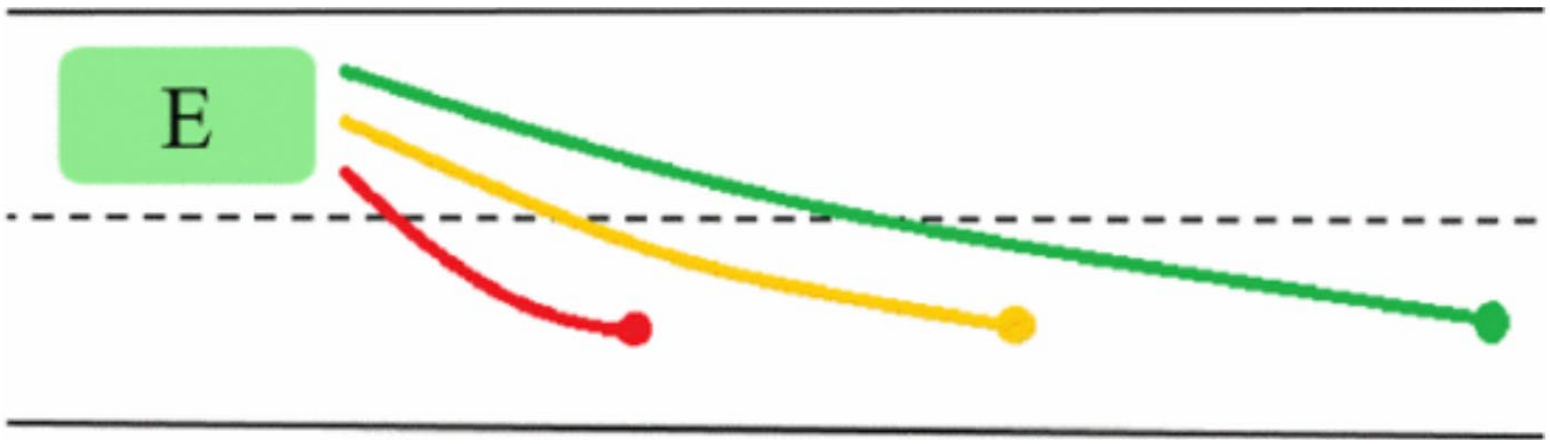
一旦ego car选择了车道跟踪/等待选项，低级轨迹规划器就被用来规划路径。低级规划器根据不同的驾驶场景生成可变长度的轨迹。



架构设计

车道变换：

一旦决定进行车道变换，就会通过epsilon贪婪策略，使用ego car的状态信息来选择目标车道中的航点。



架构设计

状态观测历史：

本文用状态观测的历史作为模型中两个LSTM层的输入，以补偿状态空间中的观测噪声，并促进在交互式和随机驾驶条件下的学习。

奖励驱动型探索：

强化学习中最常见的训练策略之一是epsilon贪婪策略，本文通过平均总奖励，而不是定期衰减的方式设定epsilon的值。

仿真实验

场景和实验设置：

本文使用CARLA开源仿真平台对算法进行验证。

仿真实验

状态空间：

状态空间由元组s表示：

$$s = [v_e, lane_{ide}, v_t, d_t, d_{tr}, lane_{idt}]$$

v_e = ego-car 的速度

lane_ide = ego-car所在的车道

v_t = 目标车辆的速度

d_t = ego-car与目标车辆的距离

d_tr =车距与安全阈值距离之比

lane_idt = 障碍车和目标车的车道

仿真实验

奖励结构：

本文使用混合奖励机制：

1）对每一步：

时间惩罚： $-\sigma 1$

用于向最终目的地前进的常规时间步长奖励： $\sigma 2$

安全车距处罚： $\exp -(dtr)$

2）对终止条件：

碰撞惩罚： $-\sigma 4$

成功奖励： $\sigma 5$

超时处罚： $-\sigma 6$

仿真实验

评价指标选取：

总平均奖励：高等级选项奖励和低级规划器选择奖励之和除以测试集总数;

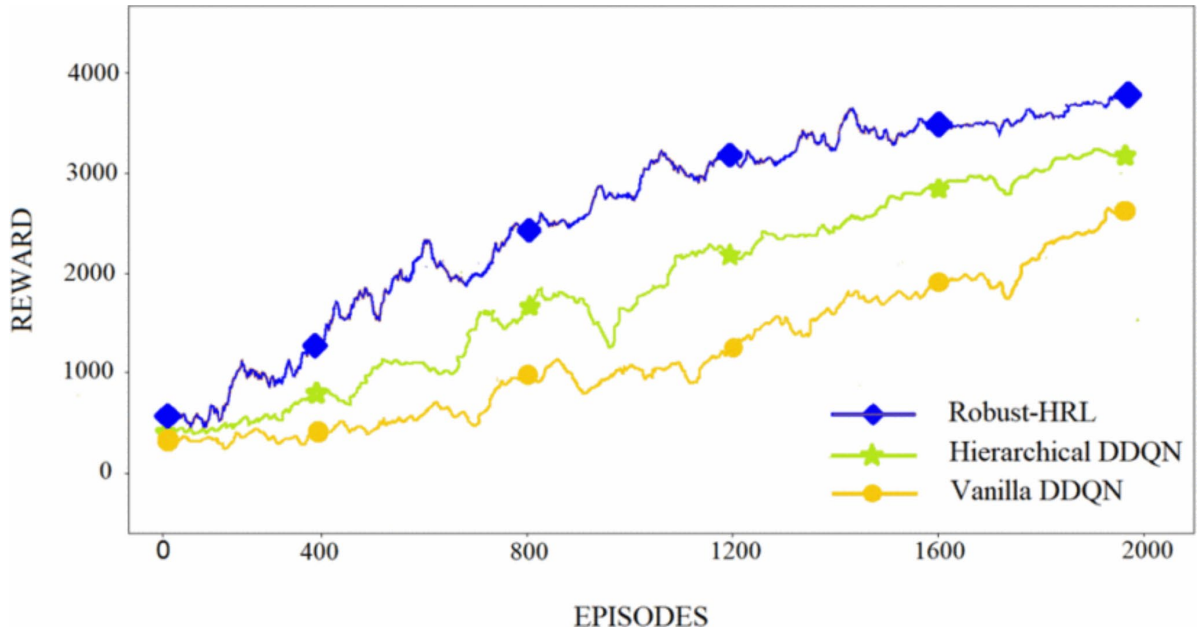
车道入侵率：测试集中记录的平均车道入侵率。当ego car在跟随车道状态下越过自己车道的边界时，就会发生车道入侵;

碰撞率：发生碰撞的测试的百分比;

成功率：ego car能够在没有碰撞的情况下完成从起点到终点的轨迹的测试集的百分比。

仿真实验结果：

为了评估作者提出的Robust-HRL算法的有效性，作者将其与基于启发式的方法和现有的最先进的RL方法进行了比较。



Method	Gaussian Noise	Total Average Reward	Lane Invasion Rate %	Collision Rate %	Success Rate %
Slot-based with PID	No	2679.40	0	17.5	82.5
Slot-based with PID	Yes	2150.25	0	27.0	73.0
Vanilla DDQN	No	2750.41	19.4	16.5	83.5
hDDQN	No	3117.82	15.9	12.0	88.0
Robust-HRL w/o LSTM	Yes	3531.24	0	9.0	91.0
Robust-HRL	**Yes**	**3728.00**	**0**	**4.5**	**95.5**
Robust-HRL	**No**	**3761.44**	**0**	**2.5**	**97.5**

问题与展望

问题：

决策选项比较简单，仅涉及变道问题

没有进行实物实验

展望：

可以加入交叉路口、匝道合并等复杂场景

可以将这套框架移植到无人机的决策、规划问题上