

文献阅读之自主泊车APA的路径规划

The future of parking A survey on automated valet parking with an outlook on high density parking

- 综述，简要介绍了传统的路径规划方法，如Hybrid A*, Anytime D*, RRT, 类似于Lattice的等。

Sampling Based Vehicle Motion Planning for Autonomous Valet Parking with Moving Obstacles

- 文章提到的方法分为4部分：障碍物车辆轨迹预测，可行驶区域选择，局部轨迹规划，车辆控制。

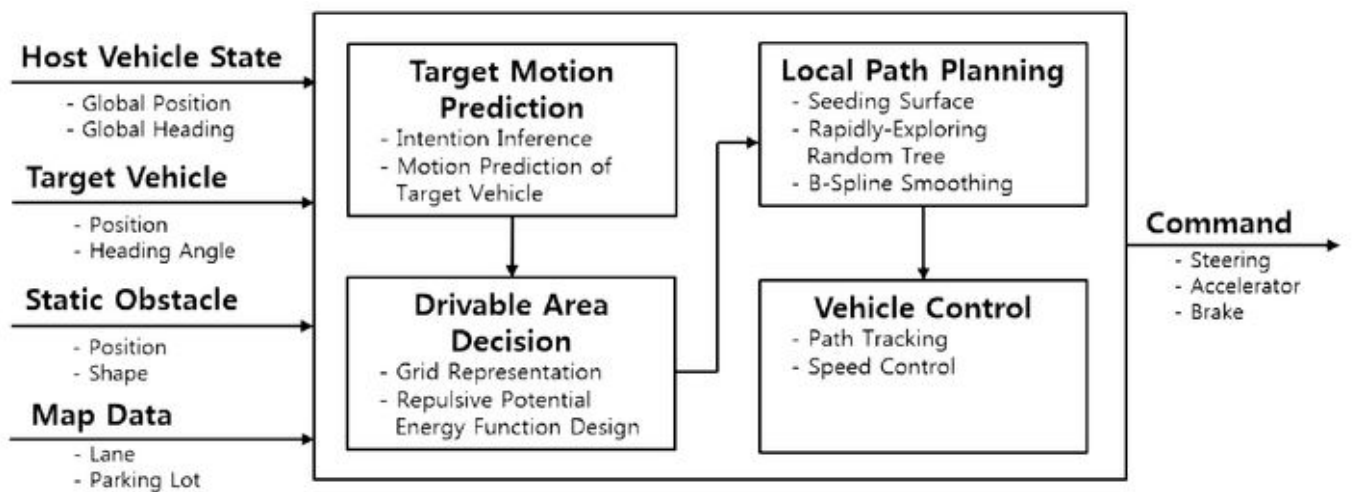
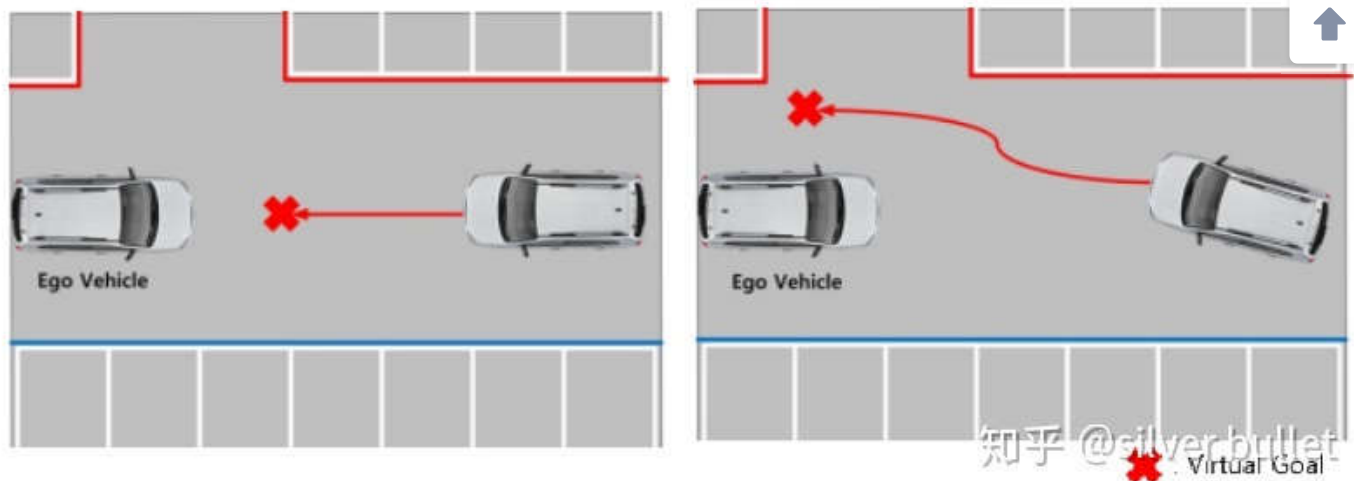


Fig. 1 Overall Architecture of Proposed Parking Algorithm.

知乎 @silver bullet

- 预测使用了Interacting multiple model (IMM) filter，输入自车位置、障碍物车的heading angle, velocity, acceleration, yaw-rate，输出的intention分2种：直行，规避对向来车（自车）。



- 可行驶区域选择采用势场法，障碍物预测轨迹、道路边沿、静态障碍物的位姿作为输入，构造势能函数，以评估危险性以及是否可行驶。

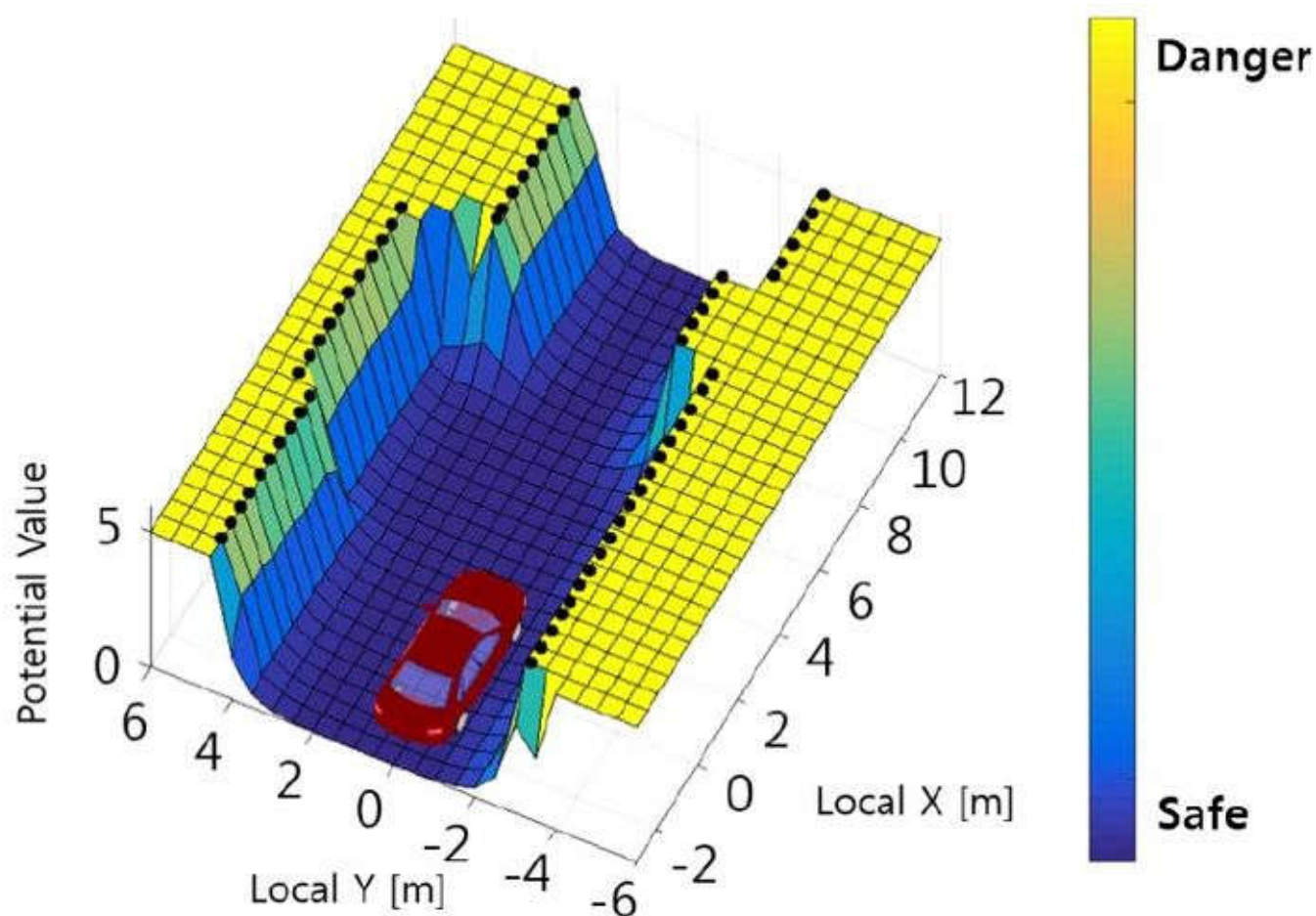


Fig. 6 Drivable Area Representation and Risk Assessment Using Potential Field Approach 知乎 @silverbullet

- 局部路径规划采用RRT算法，采用了2个小技巧来提升速度：将可行驶区域划分为seeding surfaces，在这些surfaces上随机采样；使用前述的势能函

数的倒数作为采样的概率密度函数，以减少采样点数。在采样点中寻找最小路径更像是动态规划（DP）方法。输出的轨迹用B-spline（piecewise polynomial）进行平滑。

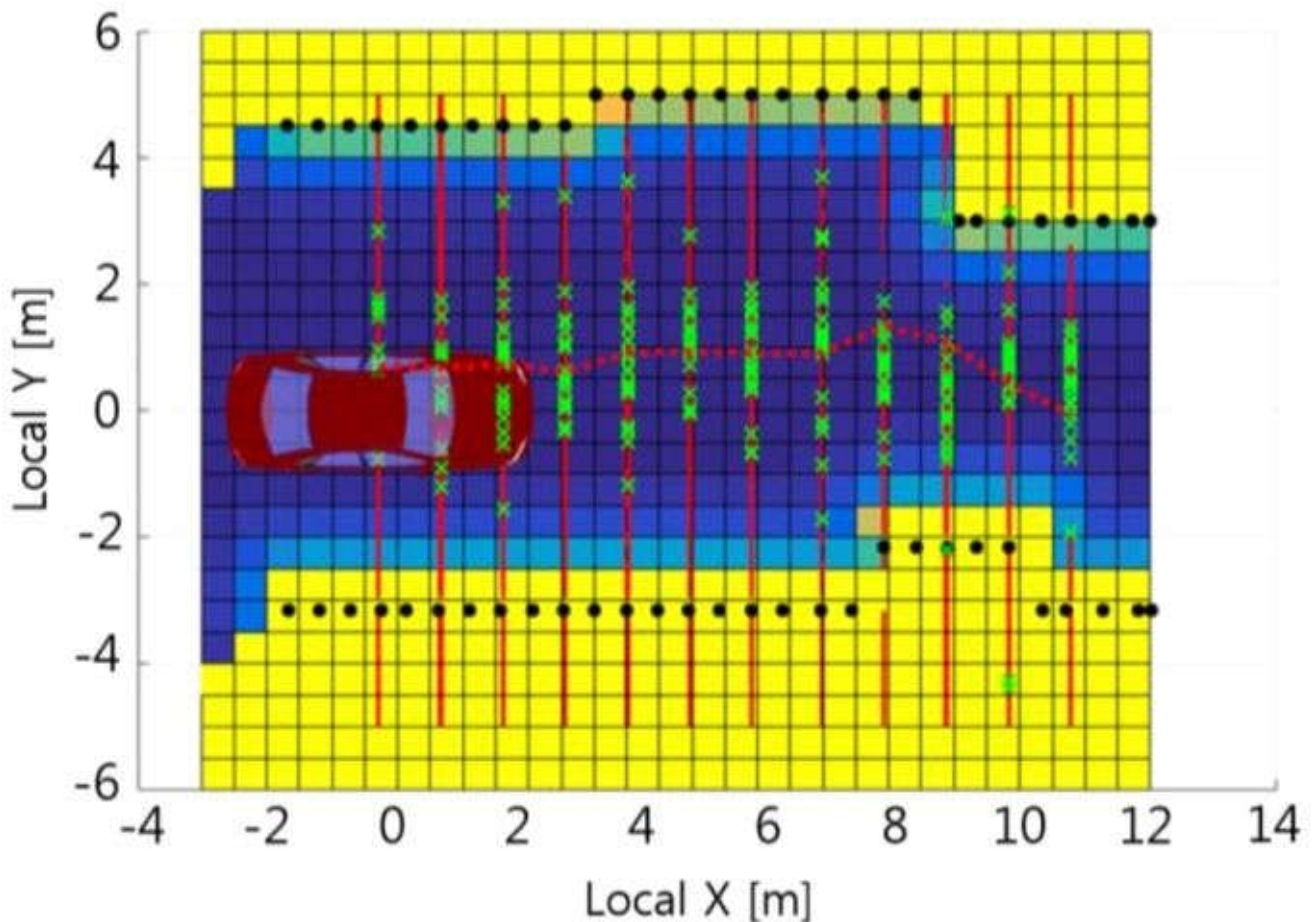


Fig. 10 Sample Generation and Local Desired Path

图中，蓝色是安全区域，黄色是危险区域，红色是seeding surfaces，绿色是采样点。

- 应用的难点：地图信息已知，已有Parking Pot目标点，预测输入是否满足？势能函数的构造与调参较难，是否适用于多种场景？RRT给出path，speed如何规划？本文只有仿真没有实车测试，仿真时速度保持在7km/h。

A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed

- 适用于智能停车场、已知全局地图，可适用于多种停车位场景，基于优化的方法。（位置，速度，朝向，steering angle）作为状态变量，可以全部求出。
- 以最小化时间消耗为目标函数，给出了运动学约束、物理约束、不碰撞约束、终点状态约束等不等式。

$$\begin{aligned} & \min t_f \\ & \text{s.t.} \left\{ \begin{array}{l} \text{kinematic principles (defined in Eq. (1)) for } t \in [0, t_f] \\ \text{mechanical constraints (Eq. (2)) for } t \in [0, t_f] \\ \text{collision-free conditions (described based on Eq. (3)) for } t \in [0, t_f] \\ \text{initial conditions when } t = 0 \\ \text{terminal conditions (defined in Eqs. (4) and (5)) when } t = t_f \\ \text{other user-specified constraints} \end{array} \right. \end{aligned}$$

知乎 @silverbullet

- 将动态优化问题转化为了非线性规划问题，使用Interior Point Method(IPM) 作为求解器。但求解器的原理与使用并没有详细介绍。
- 给出了仿真结果和仿真使用的参数推荐设置，没有实车测试。
- 应用的难点：全局地图已知，解算方法需研究，没有实车测试。

Autonomous Parking Using Optimization-Based Collision Avoidance

- 提出了一种有层次的轨迹规划方法 Hierarchical Optimization-Based Collision Avoidance (H-OBICA)，上层使用Hybrid A*（或其他的最短路径方法，如RRT*）做粗略的全局轨迹规划，输出的结果作为下层 Optimization-Based Collision Avoidance (OBICA) 方法的warm start。最终由OBICA输出一条平滑无碰的轨迹。

Algorithm 1 H-OBCA for Autonomous Parking

Input: x_S, x_F, τ^{ws} , vehicle shape \mathbb{B} , obstacles $\{\mathbb{O}^{(i)}\}_i$

Output: Optimal trajectory \mathbf{x}^* and input sequence \mathbf{u}^*

- 1: Compute $\{X_k^{HA}, Y_k^{HA}, \varphi_k^{HA}\}_k$ using Hybrid A*.
- 2: Compute initial guesses $\{v_k^{ws}, a_k^{ws}, \delta_k^{ws}, \lambda_k^{m,ws}, \mu_k^{m,ws}\}_{k,m}$.
- 3: Solve OBCA (7) with the initial guesses $\{X_k^{HA}, Y_k^{HA}, \varphi_k^{HA}, v_k^{ws}, a_k^{ws}, \delta_k^{ws}, \lambda_k^{m,ws}, \mu_k^{m,ws}, \tau^{ws}\}_{k,m}$.

- 将停车问题建模为如下形式，目标函数是最小化control effort and time, τ 是采样时间, u_k 是控制量, 即横向steering angle和纵向acceleration。采用JuMP对如下方程建模, 使用非线性求解器IPOPT求解。

$$\begin{aligned}
 \min_{\tau, \mathbf{x}, \mathbf{u}, \lambda, \mu} \quad & \kappa\tau + \sum_{k=0}^N u_k^\top Q u_k + \Delta u_k^\top Q_\Delta \Delta u_k \\
 \text{s.t.} \quad & x_0 = x_S, x_{N+1} = x_F, \\
 & x_{k+1} = x_k + \tau f(x_k + 0.5\tau f(x_k, u_k), u_k), \\
 & h(x_k, u_k) \leq 0, \\
 & -g^\top \mu_k^{(m)} + (A^{(m)} t(x_k) - b^{(m)})^\top \lambda_k^{(m)} > 0, \\
 & G^\top \mu_k^{(m)} + R(x_k)^\top A^{(m)\top} \lambda_k^{(m)} = 0, \\
 & \|A^{(m)\top} \lambda_k^{(m)}\| = 1, \lambda_k^{(m)} \geq 0, \mu_k^{(m)} \geq 0, \\
 & \forall k = 0, \dots, N, \forall m = 1, \dots, M,
 \end{aligned}$$

- 对reverse parking 和 parallel parking 进行了仿真测试, 没有实车测试。仿真的平均轨迹计算时间为1.8s, 最长时间为4.7s。虽然模型中考虑了避免碰撞障碍物, 但是仿真章节没有提到障碍物的设置情况, 全文也从未提及动态障碍物的处理方法。
- Hybrid A* 没有对steering angle rate 的限制, 可能会产生aggressive的轨迹曲线。
- 应用的难点: 全局地图已知, 目标点已知。原理与逻辑较复杂。计算量大,

耗时较长。没有考虑动态障碍物，也无法靠更新地图、周期计算新轨迹来避动态障碍物。

Path Planning for Autonomous Car Parking

- 将停车的轨迹分为2段，第1段是假设一辆虚拟车停在目标停车位上，用几何法求其离开车位到达车道target line的轨迹；第2段是求自车从当前位姿向前行驶到虚拟车所停位姿的轨迹，使用5次多项式拟合第2段轨迹。

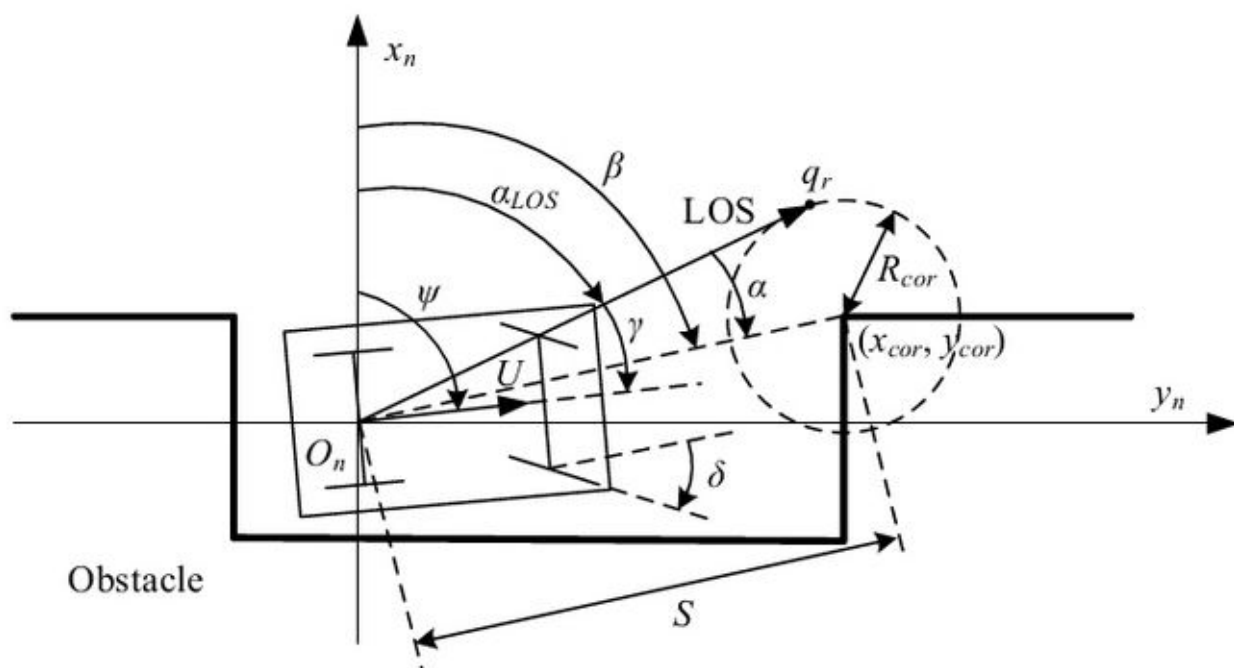


FIGURE 7. RETRIEVING A CAR FROM THE PARKING SPACE

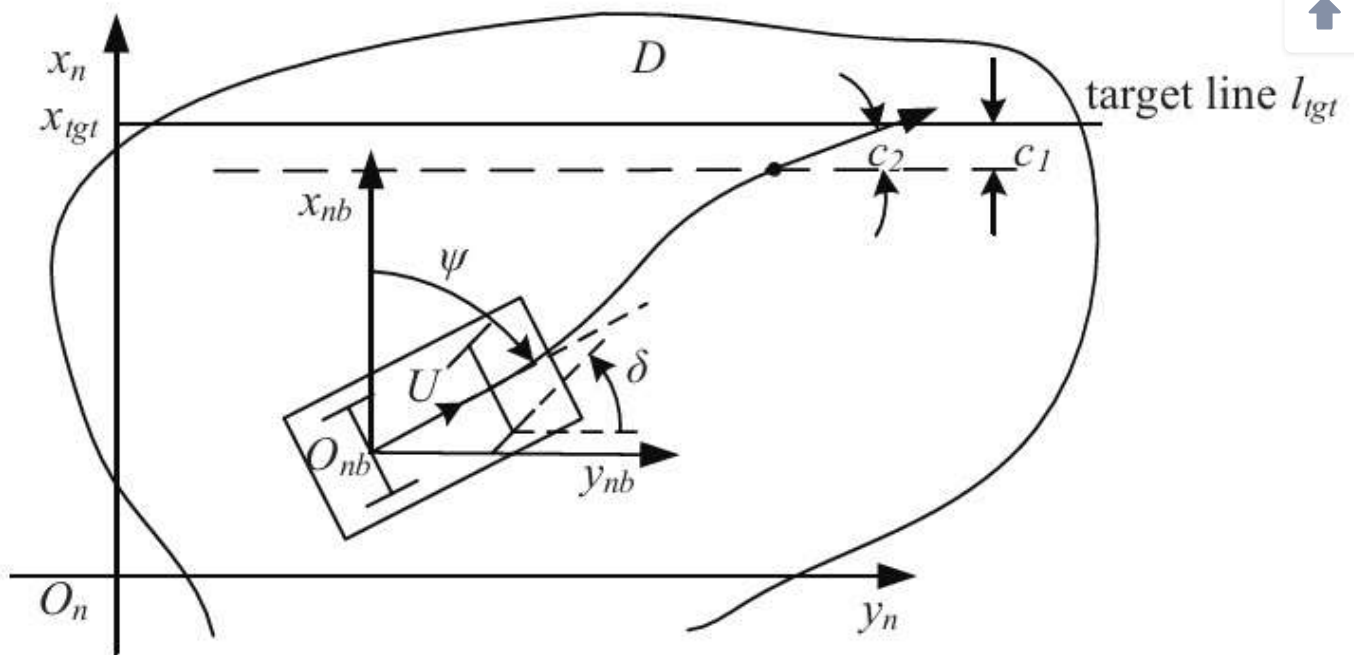
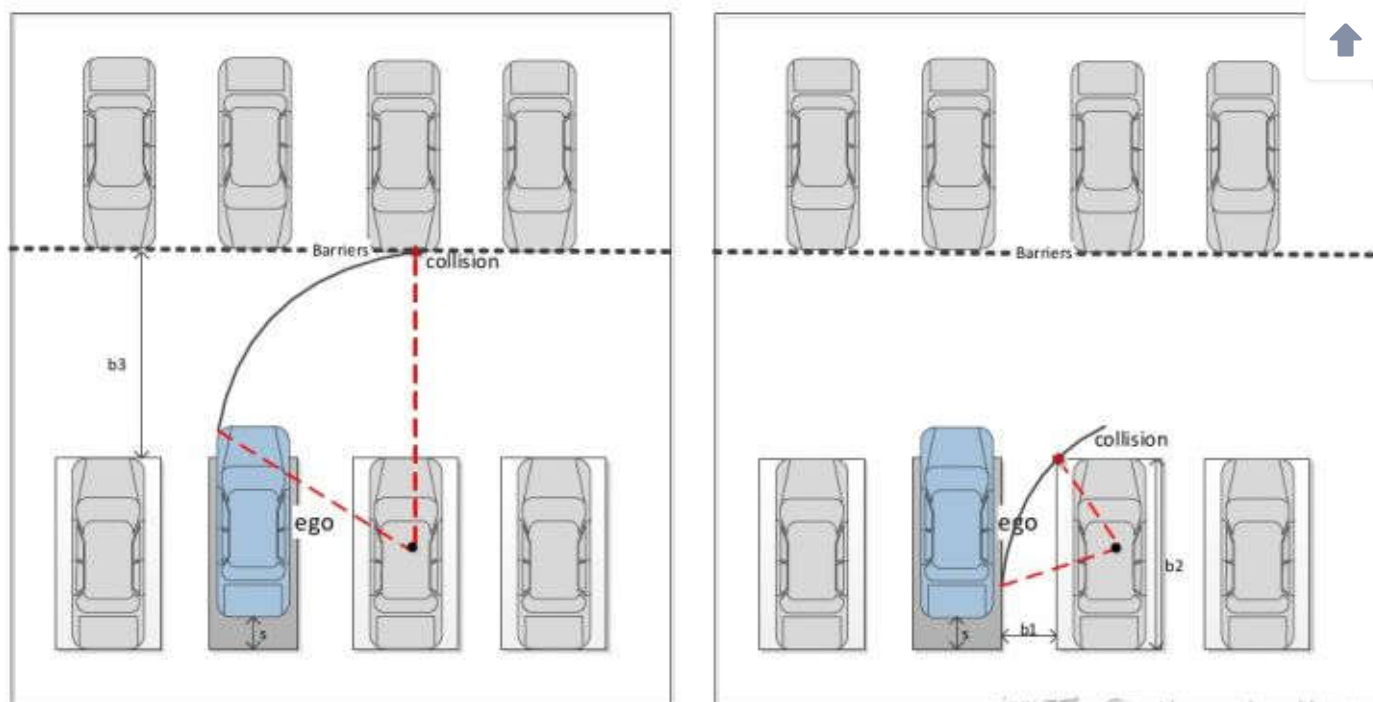


FIGURE 5. STEERING A CAR TO THE TARGET LINE.

文中没有提动态障碍物与静态障碍物的处理，但是限定了自车只能在无障碍的区域 D 中运动。对target line 与 y_n 轴的距离也做了限定。

- 对parallel (tight), parallel (wide), perpendicular (head-in), perpendicular (back-in) 4种场景进行了仿真，没有实车测试。
- 没有速度规划。
- 仿真说明计算速度很快。
- 文章的创新点——switch control law在停车轨迹规划中的应用，原理没看懂。

Trajectory Planning for Cooperative Autonomous Valet Parking



Two possible collisions. Left: Front collision. Right: Side collision

- 如上图所示，本文只考虑垂直停入的场景，采用几何法计算停入路径，基本和参考文献5相同。只有2页介绍parking in，内容较少。
- 为求取有效的转弯半径范围，分别以前方碰撞点和右侧碰撞点为临界条件，给出计算公式，但并没有给推导过程。对障碍物的考虑都是按照车辆规范停入的理想假设。

$$R_{fc} \geq \sqrt{\left(\frac{(s - b_2 + k_r)^2 + b_1^2 + b_1 \cdot W}{2b_1}\right)^2 + L_r^2}$$

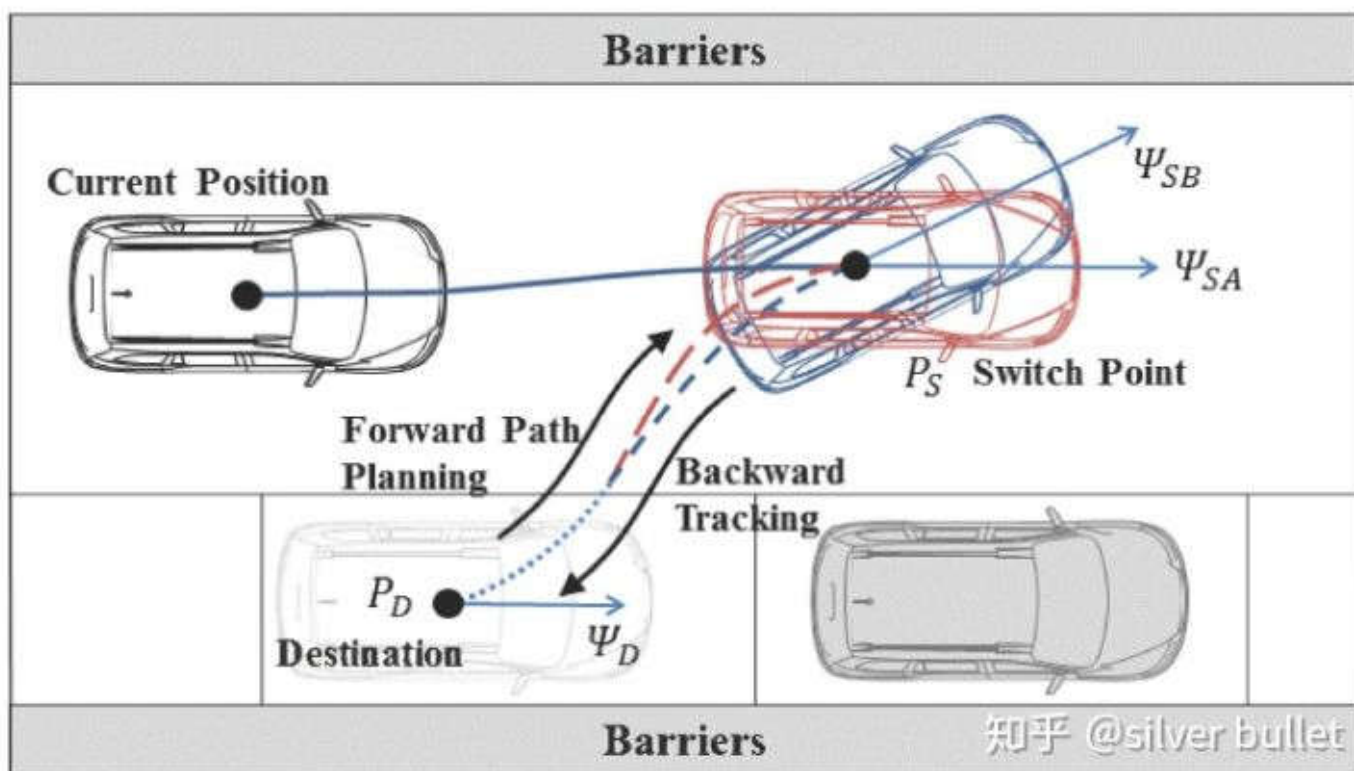
$$R_{sc} \leq \sqrt{\left(\sqrt{(s - b_3)(s - b_3 - 2K_f - 2L) - \frac{W}{2}}\right)^2 + L_r^2}$$

- 使用 Schoenberg and Reinsch smoothing spline approach 将路径转换为3阶 splines。
- 停车过程中没提如何避免与静态障碍物和动态障碍物的碰撞。

A trajectory planning method based on forward path generation and backward tracking algorithm for Automatic Parking Systems



- 本文是《Trajectory Planning for Cooperative Autonomous Valet Parking》的参考文献5，适用于平行停入和垂直停入车位2种场景，尤其是倒车停入的场景。
- 整个方法分为2个部分：1是向前的轨迹生成，即计算车头朝前从车位开出停到路边的轨迹，即下图中的 $P_D \rightarrow P_S$ ；2是向后的轨迹跟踪，即自车沿着1生成的轨迹逆向倒车进入车位， $P_S \rightarrow P_D$ 。



- 规划的轨迹也分为2部分，第1部分是locating segment，是圆弧，使车无碰的开出车位；第2部分是entering segment，是Bezier曲线，使车向前行驶到start point（也叫switch point或target point）。如下图所示。

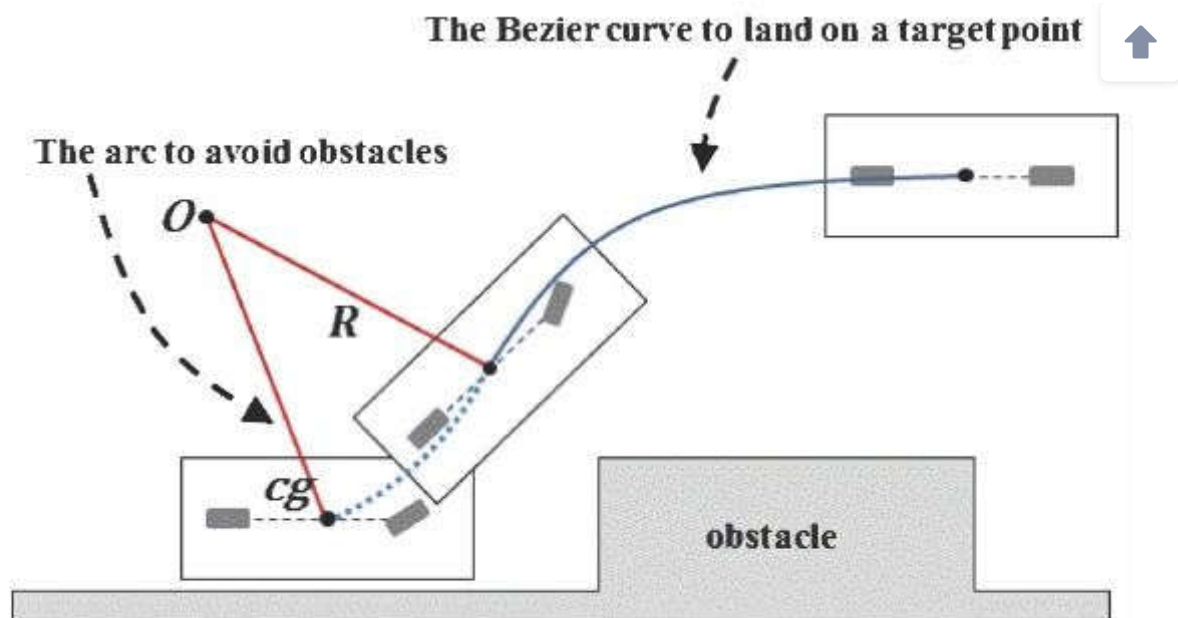


Fig. 8 An example of combined path for parallel parking.

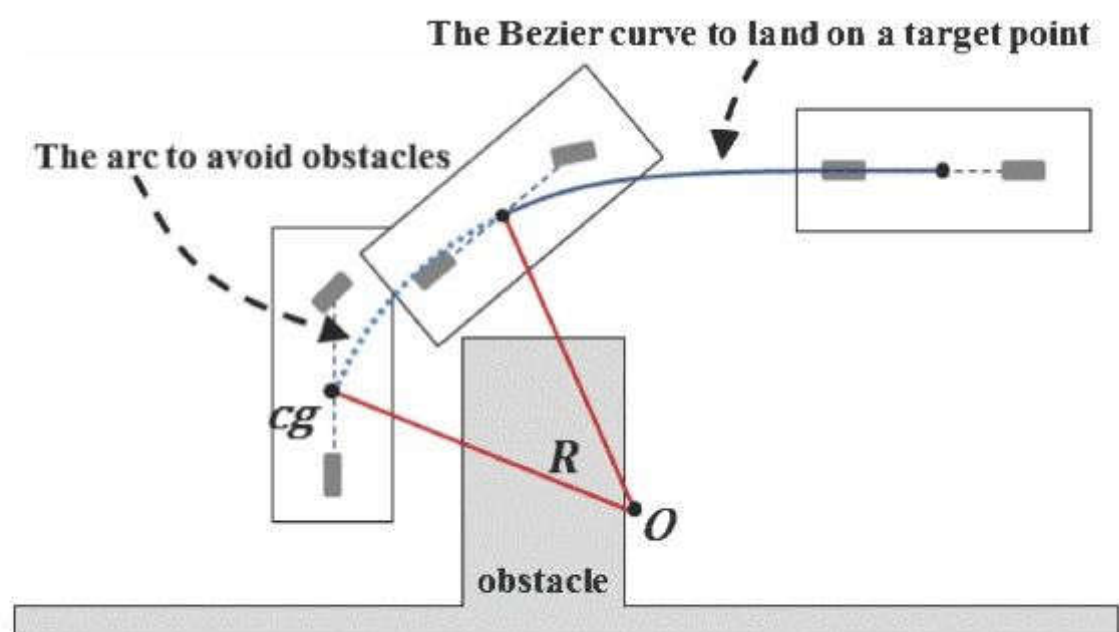
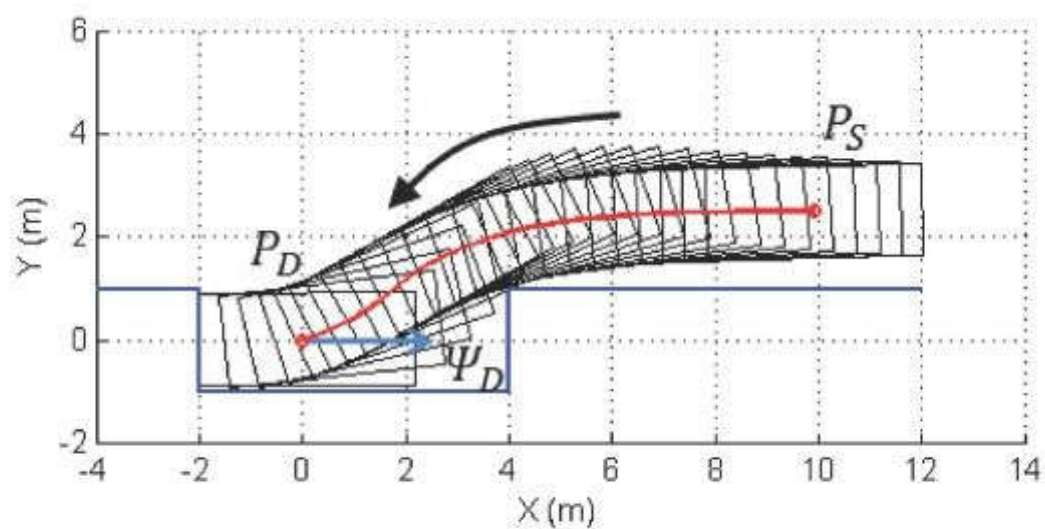
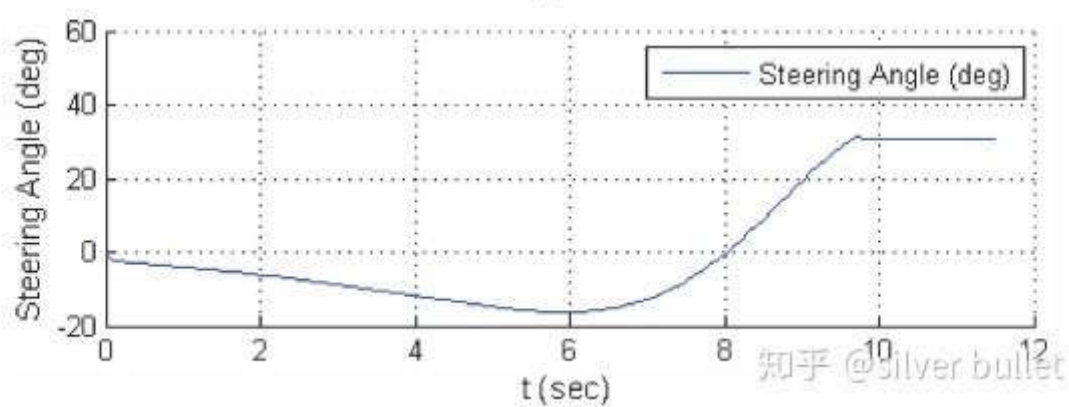


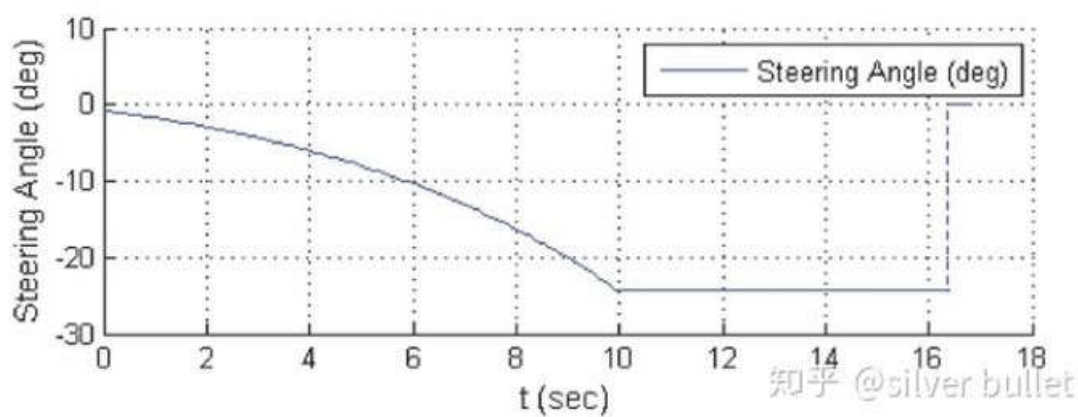
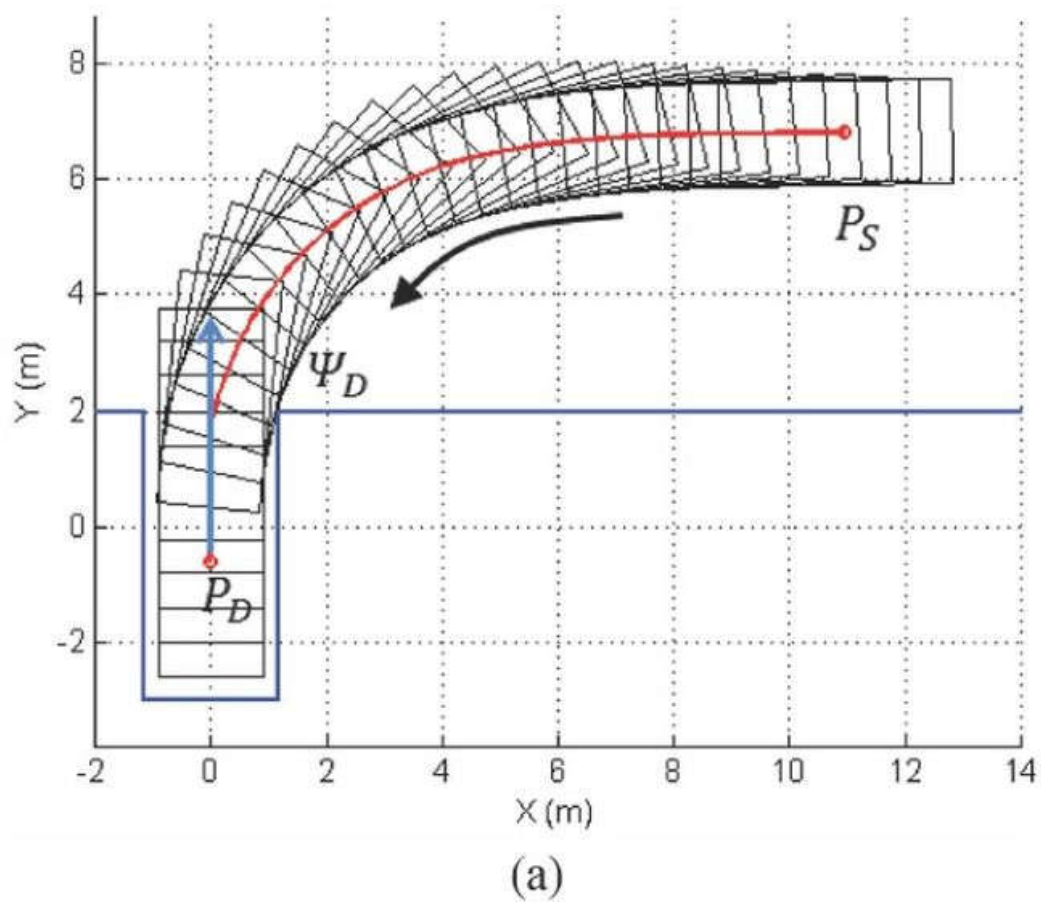
Fig. 9 An example of combined path for perpendicular parking.

- 可以输出连续的steering angle, 车一次操作, 无需停顿后打方向盘。下图是仿真的结果。



(a)





- 既有仿真也有测试。在仿真中按0.1m/s的匀速计算，测试没提设置。
- 计算第1部分圆弧轨迹时，考虑了不同停入方式可能会发生的前方碰撞和侧方碰撞作为边界条件。

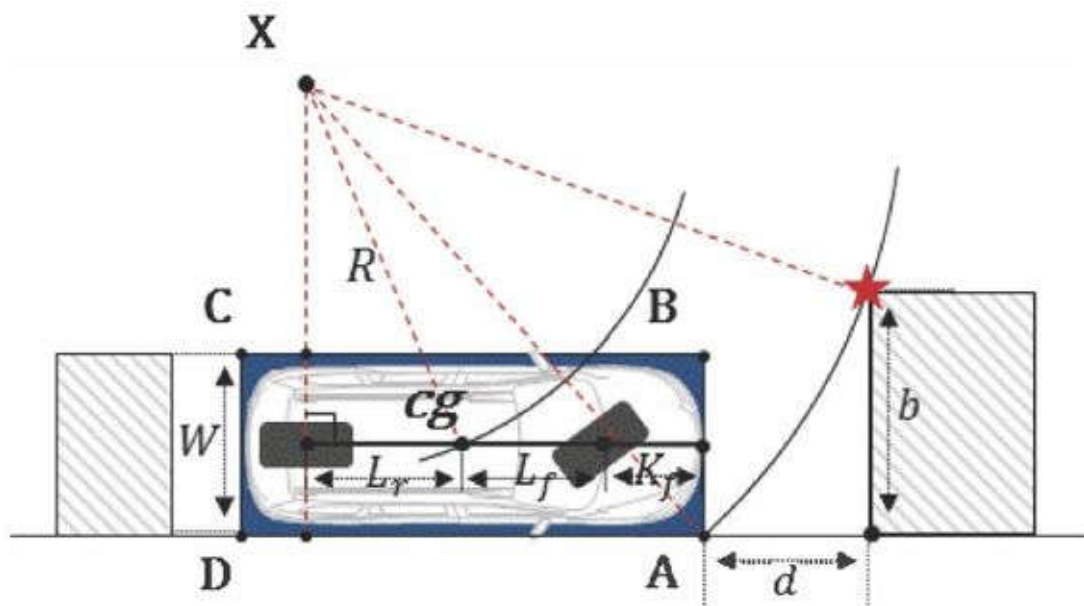


Fig. 4 An illustration of circular geometric model of cg reference for parallel parking.

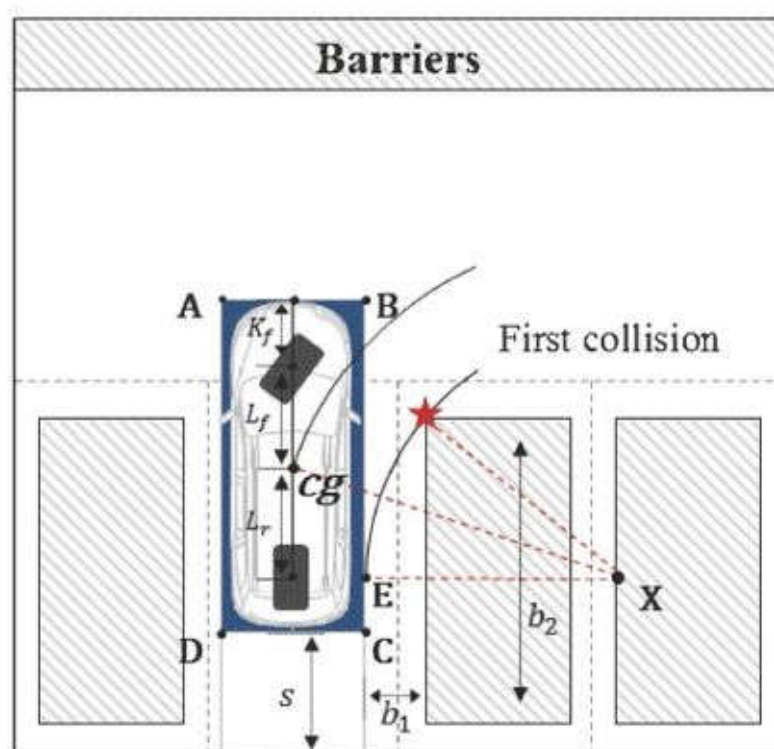


Fig. 5 First collision of circular geometric model for cg reference for vehicle during perpendicular parking.

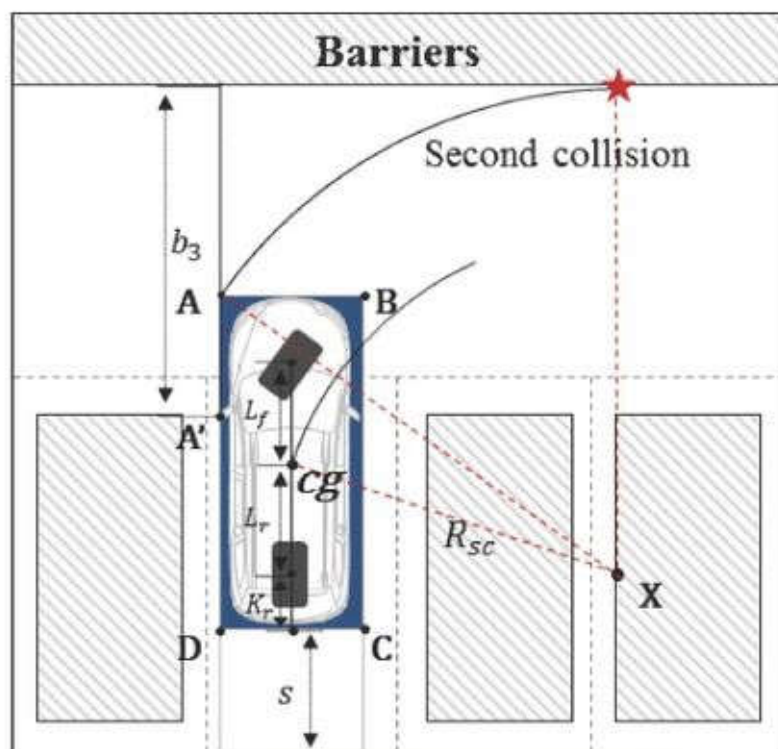


Fig. 6 Second collision of **circular geometric model** for *cg* reference for vehicle during perpendicular parking.

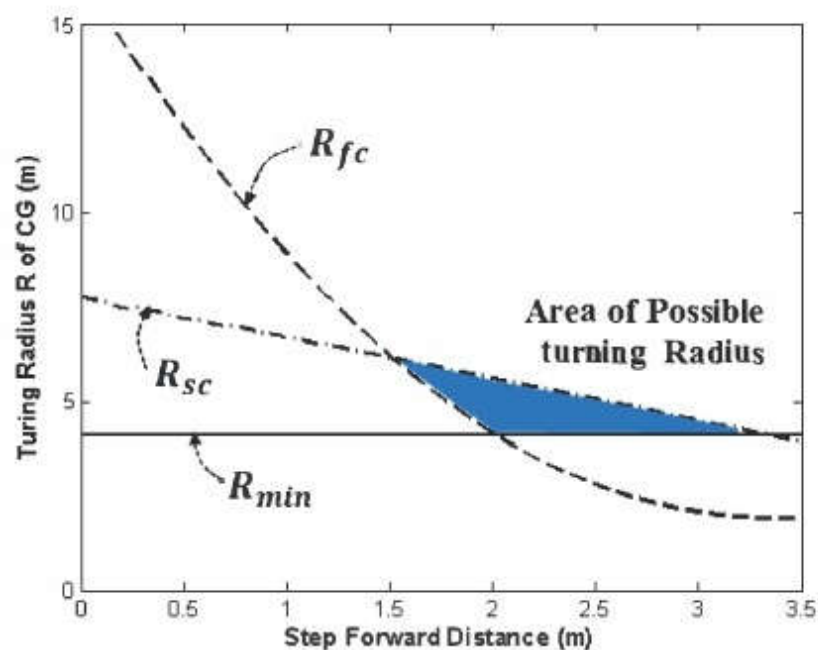


Fig. 7 Area of possible turning radius R during perpendicular parking.

- 所采用的车辆模型和自行车模型的不同在于考虑了车辆重心的slip angle。即下图中的 β 。

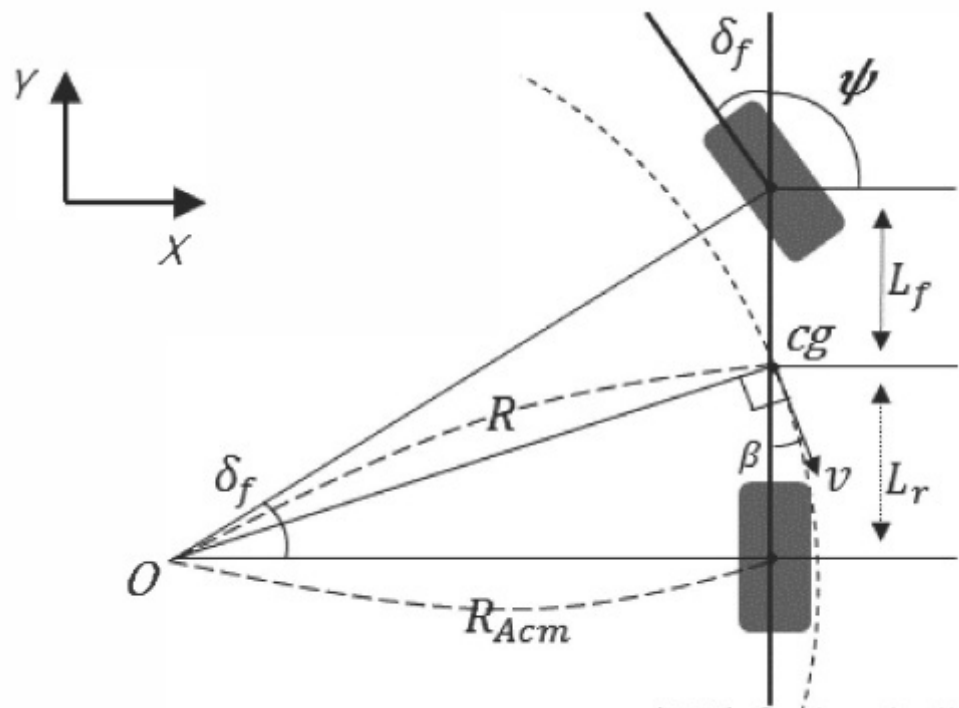


Fig. 3 Modified bicycle kinematic model for backward moving.

- 没有提动态、静态障碍物的避障。
- 常用的双圆弧法会有steering angle突变的问题。如下图：

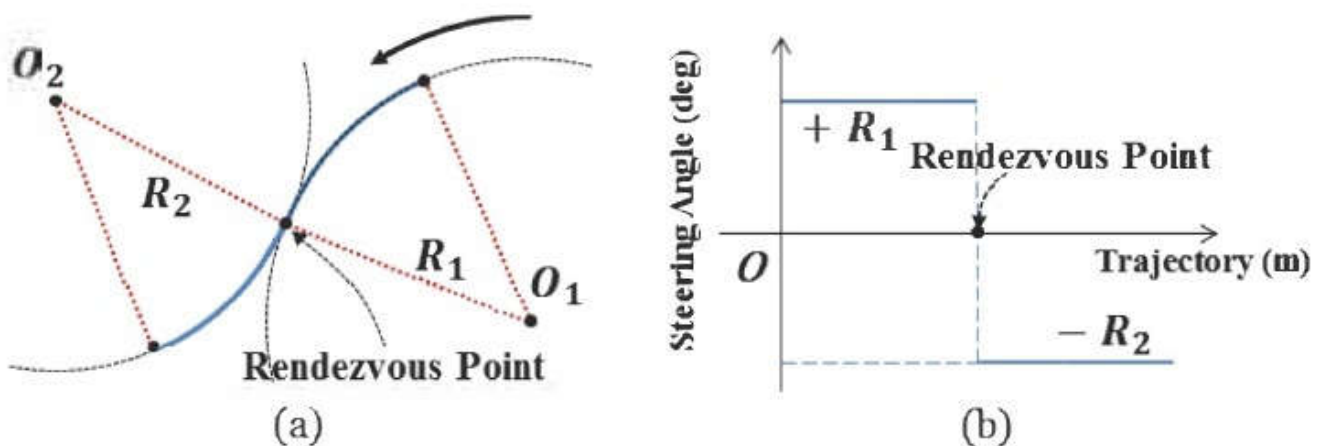


Fig. 1 An illustration of abrupt steering angle change at the rendezvous of two circles in a case of parallel parking maneuver.

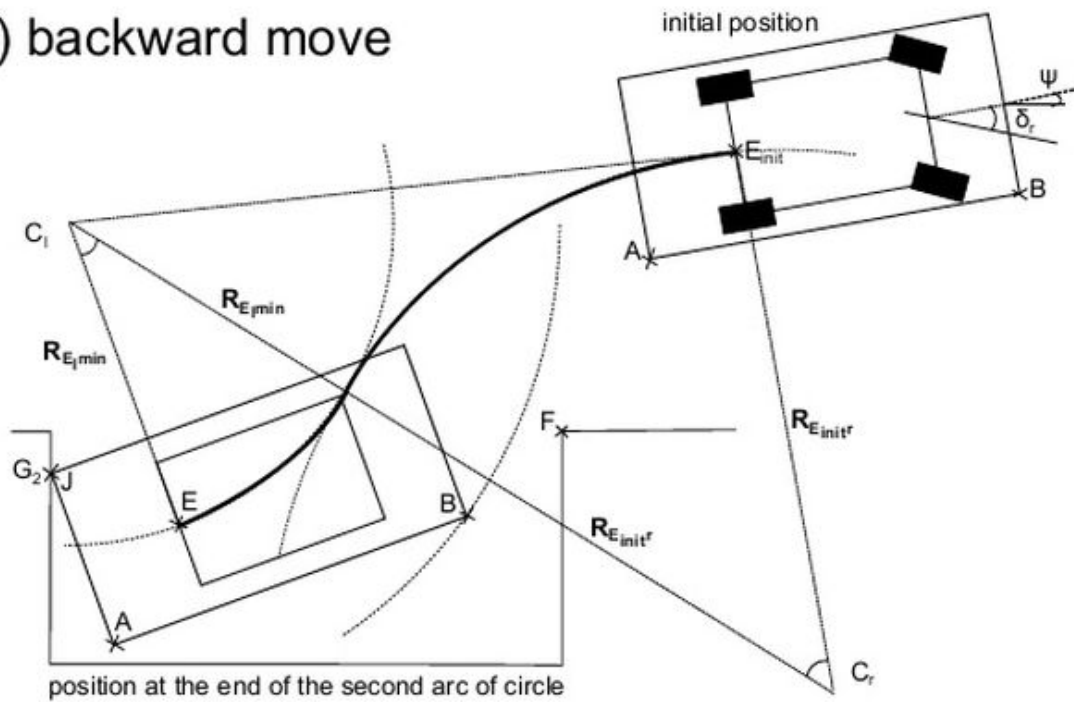
- 这是目前遇到的比较详实清晰的方法，难点在于Bezier曲线的控制点比较难调。

Geometric continuous-curvature path planning for automatic parallel parking

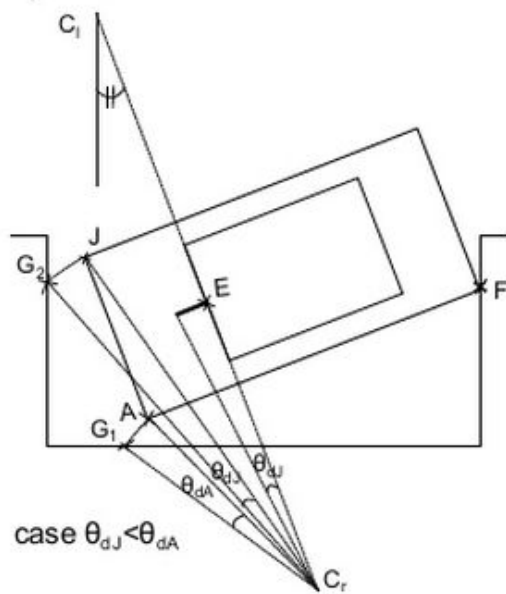
- 本文只关注平行停车场景，在几何法输出的圆弧路径的基础上，将其转换为曲率连续的clothoid曲线，避免Parking过程中多次动作，可以更快的停入。
- 给出路径的同时，还可以给出Parking过程控制量：longitudinal velocity and steering angle。
- 包含很多公式，或可参考。粗读没有看懂。
- 普通的几何法一般是求停稳在车位上的车辆parking out的路径，将路径反向，作为parking in的路径。如下图，车是按照 $c \rightarrow b \rightarrow a$ 的顺序 (parking out) 求取的，车parking in 时却需要反向的沿着这条路径行进，即 $a \rightarrow b \rightarrow c$ 。整个停入的过程被分为数个maneuver。本文就是通过将分段的arc path，转换为曲率连续的clothoid path，将parking 过程整理为1个maneuver。



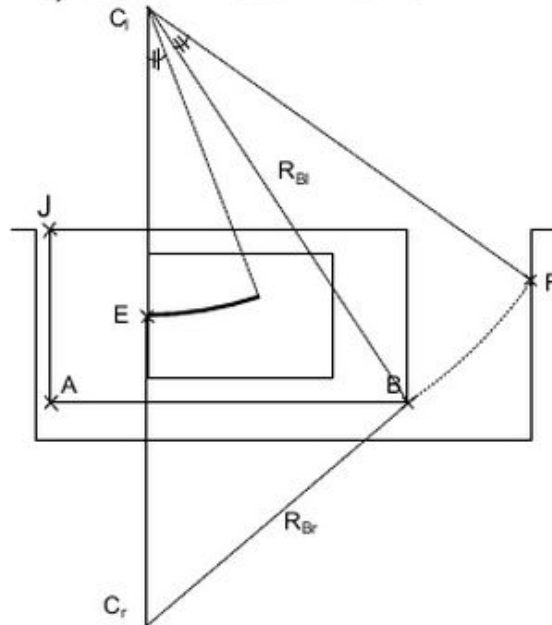
a) backward move



b) forward move

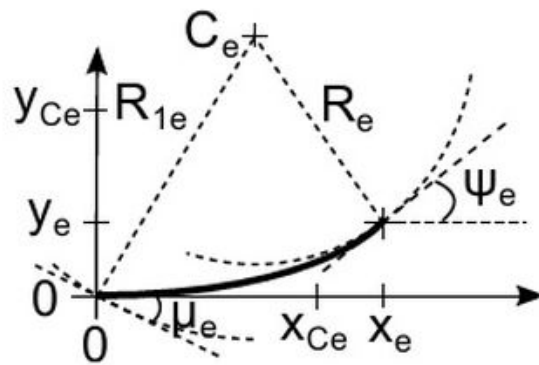


c) backward move

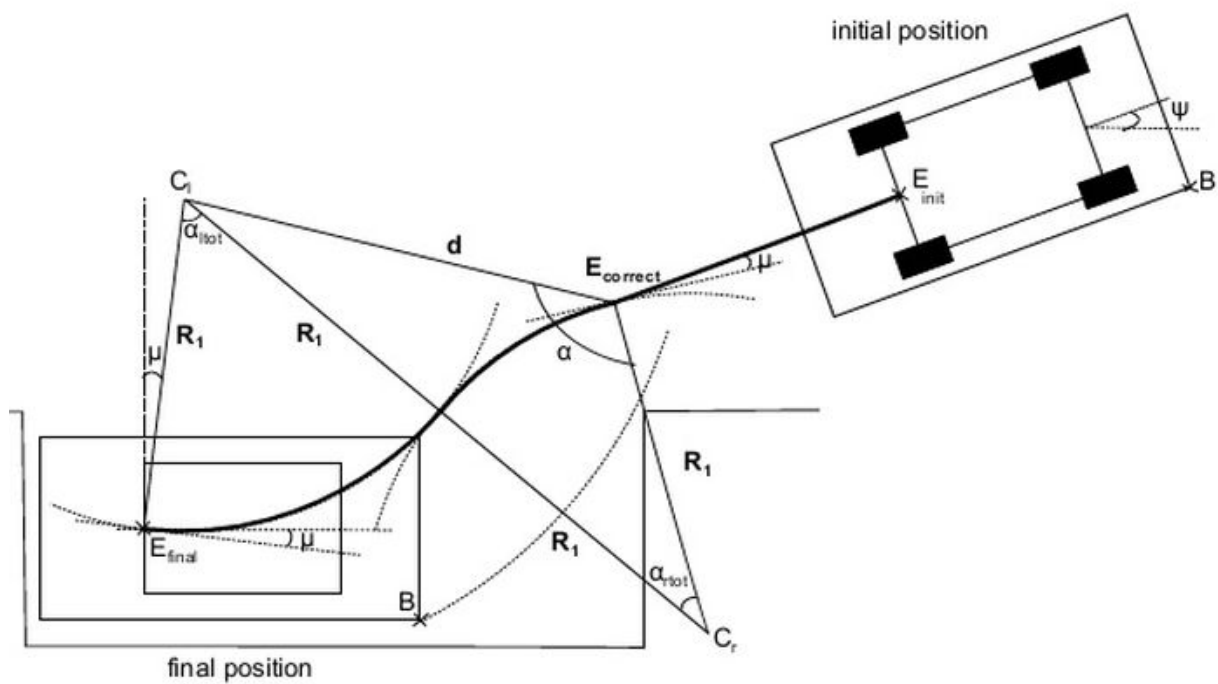


Parking in three maneuvers with the "reversed" geometric method

知乎 @silverbullet



Clothoid turns



Path for a parking in one maneuver

知乎 @silver bullet

Path Planning and Steering Control for an Automatic Perpendicular Parking Assist System

- 本文只考虑垂直倒车停入车位的场景，有仿真和实车测试。
- 以前方、侧方碰撞为临界条件，求倒入车位的圆弧路径。

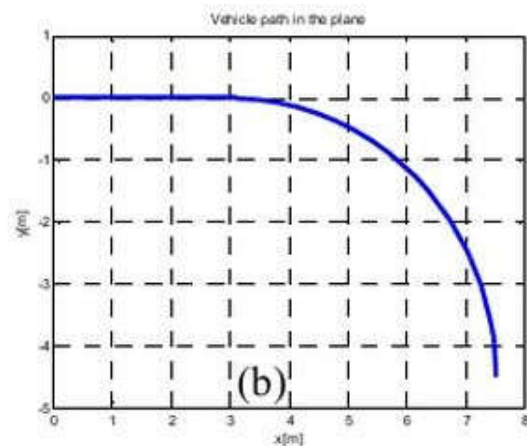
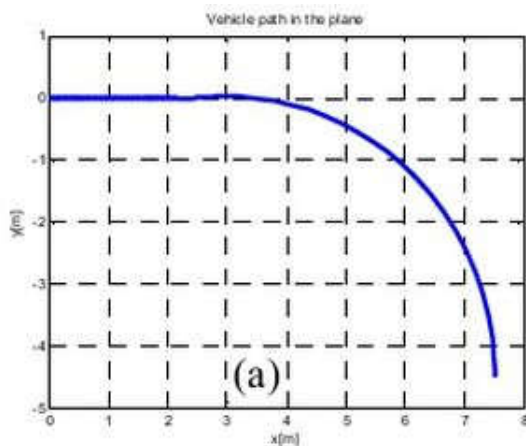


Figure 5. Perpendicular parking: Planar paths of the vehicle using bang-bang control (a) and saturated control (b).

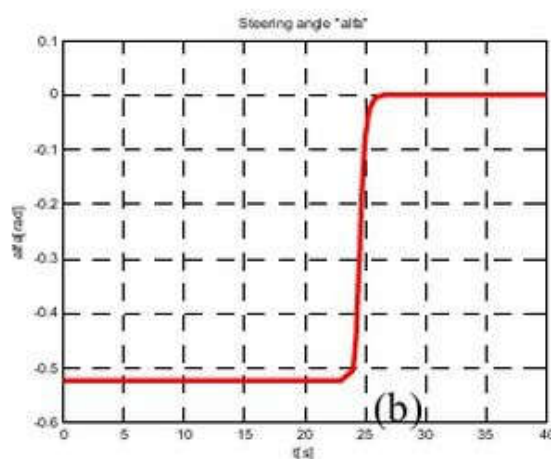
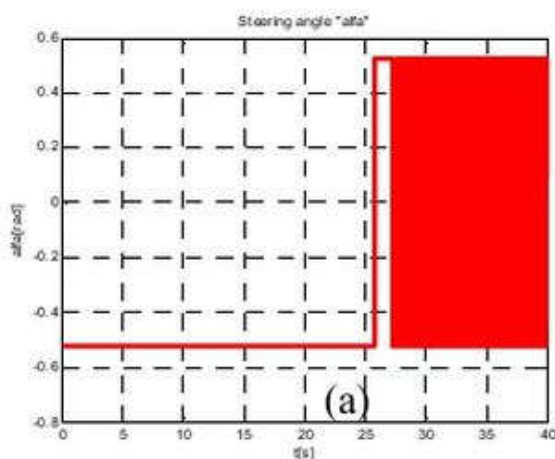


Figure 6. Perpendicular parking: Evolution in time of the front-wheel steering angle using bang-bang control (a) and saturated control (b).

A Fast Motion Planning Algorithm for Car Parking Based on Static_Optimization

- 局部优化的方法，不是一次性求出车parking in的轨迹，而是设定步长，迭代的求每个步长运动的轨迹。并且在迭代的过程中会改变车的运动方向。是局部最优解，是否是全局最优解，本文没有给出证明。
- 将动力学模型的约束条件使用Runge-Kutta方法离散化，分解停入过程为2个阶段：parking point→switch point, switch point→start point，并给出了选取switch point的具体方法和原则，但是没有看懂。

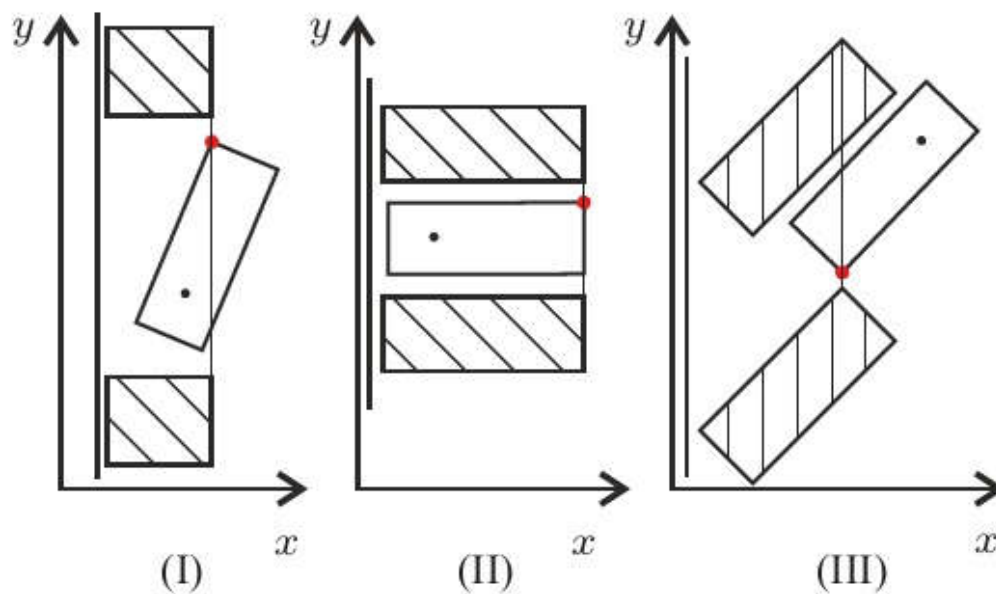


Fig. 3. Phase switching point for parallel (I), garage (II) and angle parking (III) for left parking spots.

- 对于不同的阶段，其cost function中的参数、权重要设定不同的值。
- 对parallel, garage, angle parking 3种不同的场景进行了仿真，计算速度极快，耗时ms级。

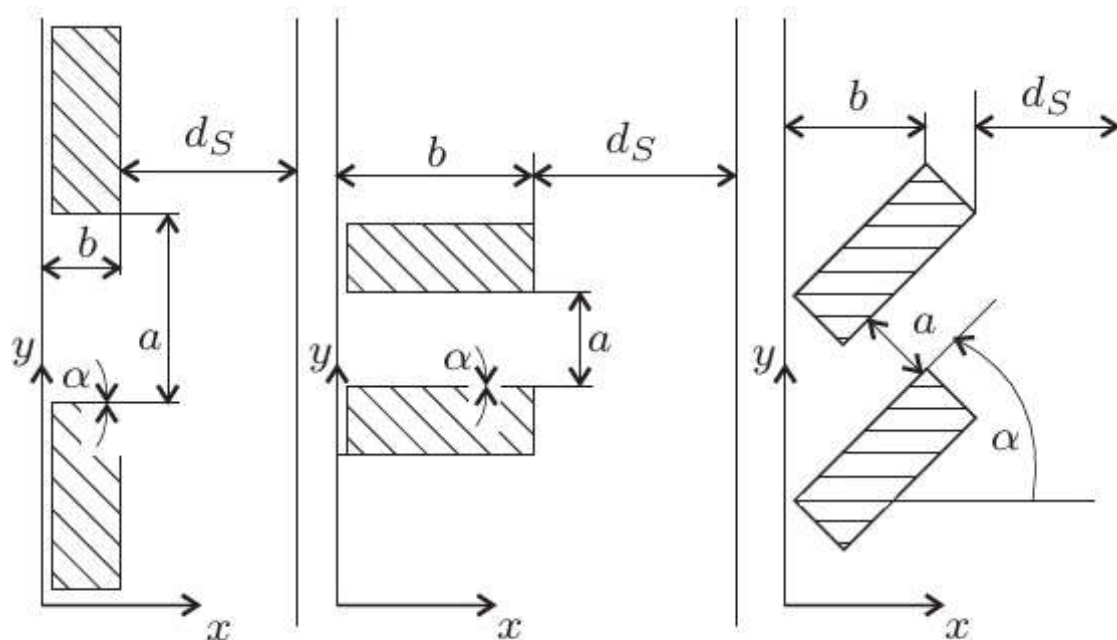


Fig. 1. Notation for parallel, garage and angle parking.

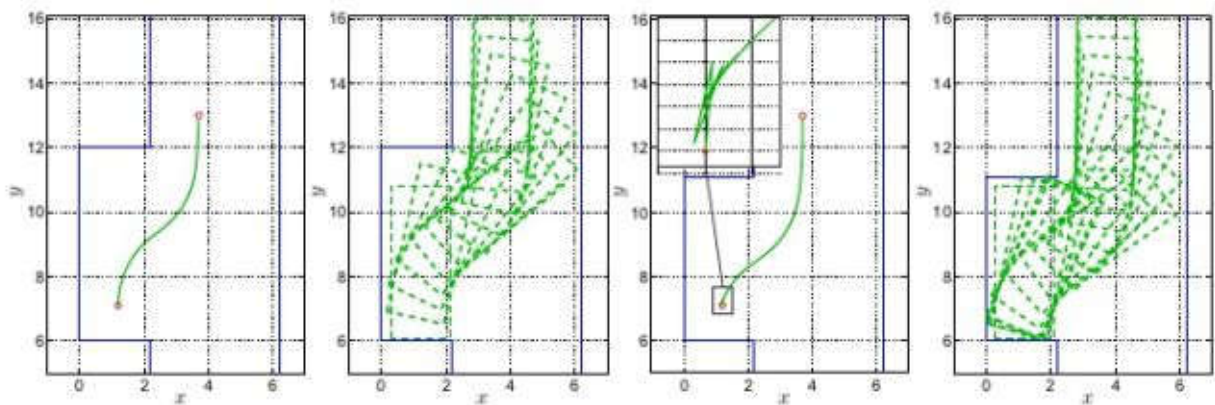


Fig. 6. Parallel parking scenarios.

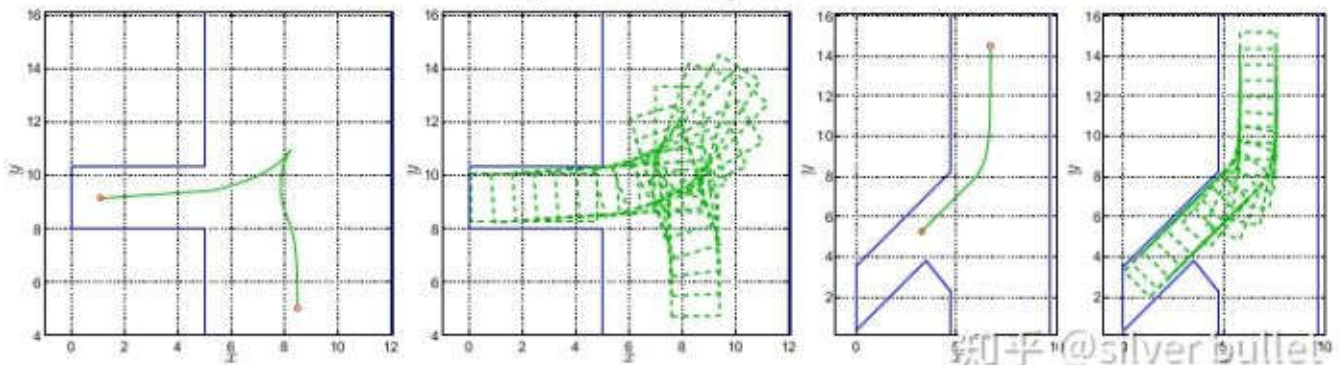


Fig. 7. Garage and angle parking scenario.

- 总的来说，不明白本文的整体思路。

Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments

- 本文的规划方法分为 free space 环境和 semi-structured 环境下的路径规划，目标是规划出一条平滑且满足车辆non-holonomic约束的最优路径。free space的方法分为2部分，第1部分是采用启发式搜索方法获得一条满足车辆运动学约束的可行驶轨迹，比如Hybrid A*方法，通常结果不是最优、但接近最优。第2部分是在此基础上，采用数值优化产生更平滑的轨迹。semi-structured 的规划是在free space 环境规划方法上的调整。

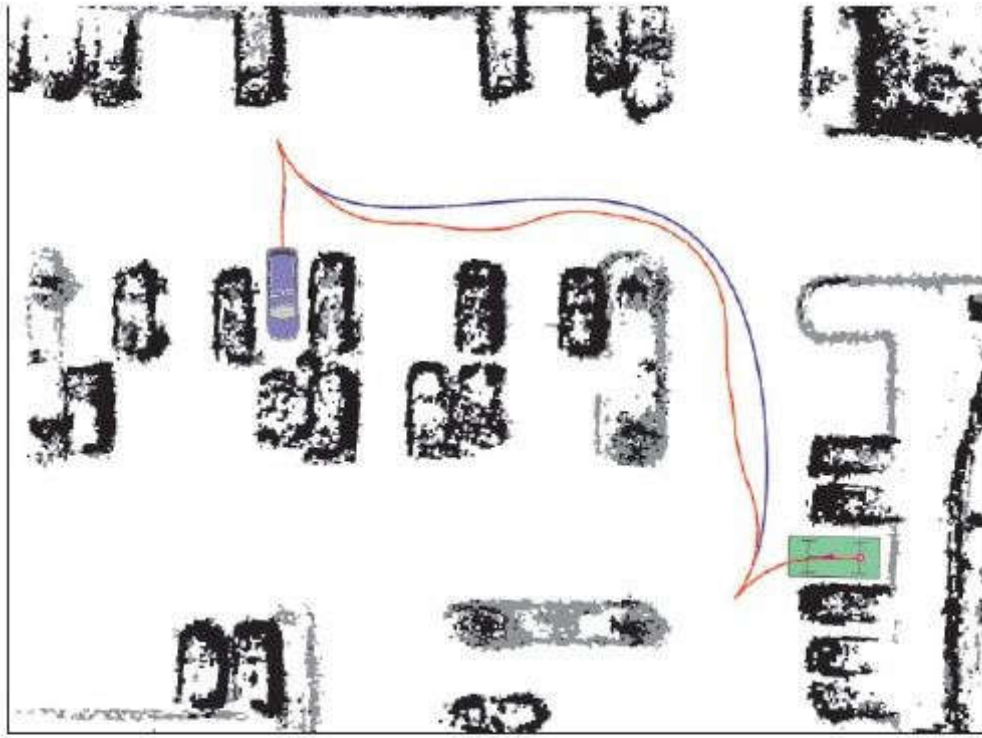


Fig. 5. Hybrid-A* and CG-smoothed paths for a complicated maneuver, involving backing out of and into parking spots. The Hybrid-state A* path is the wavy red line and the CG solution is the smooth blue line.

- 第2节介绍Hybrid A*的实施细节、问题与相应的解决技巧。
2.1提出了2条Heuristics帮助计算代价、较少需要拓展的节点数，分别是non-holonomic-without-obstacles和holonomic-with-obstacles。
2.2为了提升精度、加快搜索速度，在按Hybrid A*策略拓展节点的同时，计算了从当前节点到目标点的Reed-Shepp曲线路径。并不在每一个节点计算，而是每N个点计算一次，且越靠近目标点，计算的越频繁，计算间隔越小。
2.3讲分辨率的问题。采用了固定的grid尺寸、可变的搜索范围。 $do+dv$ 越大，从当前节点出发搜索的范围越广、本次规划的路径越长。

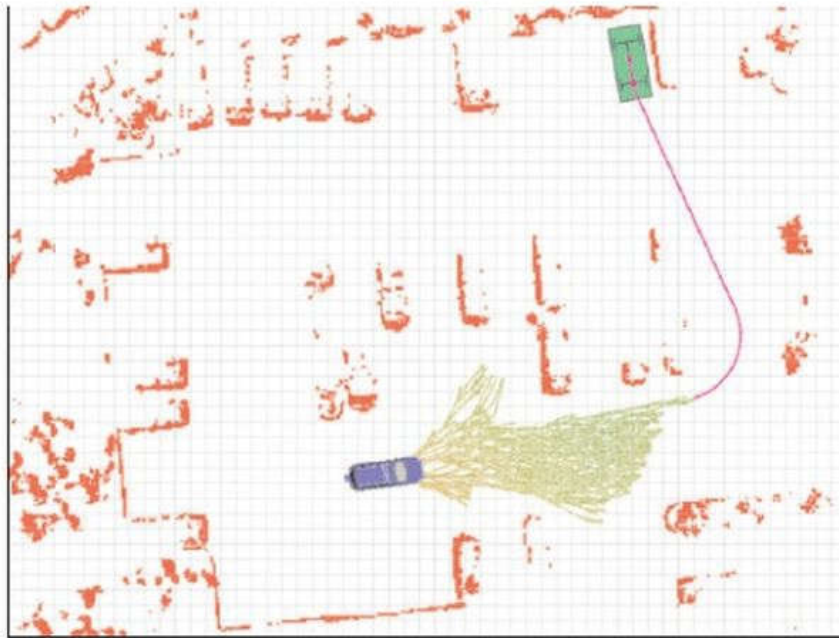


Fig. 4. Analytic Reed-Shepp expansion. The search-tree branches corresponding to short incremental expansions are shown in the yellow-green color range and the Reed-Shepp path is the purple segment leading towards the goal.

- 第3节在第2节输出的drivable但是steer可能会突变的路径基础上，使用 conjugate-gradient descent (CG) 技术进行优化。
 3.1给出了cost function以及其梯度。良好的梯度是高效率计算的保障。
 3.2讲轨迹的无碰撞安全性。因为优化仅仅把离障碍物远作为一项cost，受计算消耗限制没有把碰撞检测放入优化环节，因此，优化的输出轨迹无法保证无碰和安全。解决方法是，对优化的结果进行碰撞检测，若某节点处危险，则固定其对应的Hybrid A*节点作为Anchors，重新优化，并且在次轮优化中不移动Anchors，平滑其他的节点。利用了Hybrid A*的输出drivable且safe的特点。
 3.3探讨了最短距离与距障碍物的距离的折衷，解决最短距离往往紧紧围绕障碍物的问题。常用势场法，根据路径点与附近障碍物的距离，计算惩罚，在cost function中最小化这个惩罚。这有3个缺点：1是较窄的路被排除，2是需要计算与多个障碍物的距离，计算复杂，3是梯度性质不好，不利于优化。本文提出了利用do、dv计算势能的新方法。
 3.4探讨如何使输出轨迹点更密集以利用车辆控制。因为参数化插值方法，如cubic spline，对输入点集特别敏感，故本文对固定轨迹点并对其上采样，然后CG优化、最小化路径的曲率。和A* Anchors同样思路。

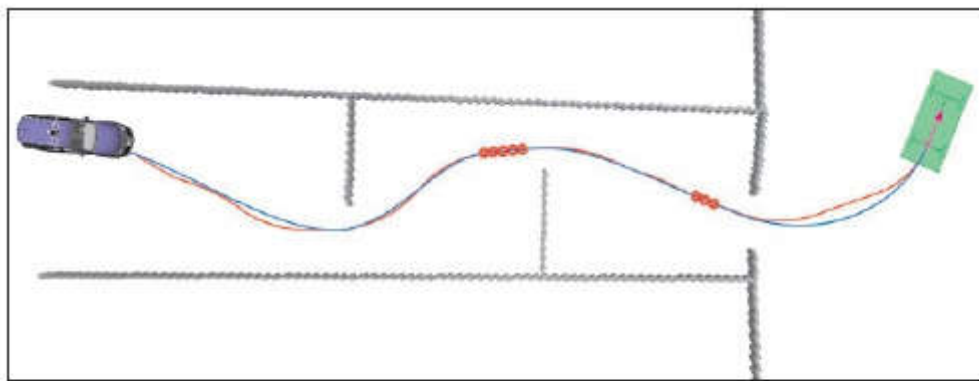


Fig. 6. Anchoring waypoints to guarantee smoother safety.

- 第4节为了识别高级驾驶行为，如not cut across the parking lot, stay in the right lane, 获取拓扑图等，将freespace的规划方法针对semi-structured环境进行了改进。拓扑环境有2个作用：惩罚偏离、提供macro-actions信息。


4.1提出对Hybrid A*的改进，主要是在cost function中加入了对节点偏离lane network的惩罚。2个Heuristics基本没变，只是第2条对距离计算的方式略有更新。利用micro-actions指导A*节点的扩展，以及Reed-Shepp扩展。

4.2在smooth优化的potential function中加入了偏离lane network的惩罚。并计算了点与图边距离对x的导数。

- 总的来说，本文方法非常全面，理论清晰，但是代码实现困难。

总结

- APA常用的方法有这么几类：图搜索、基于采样、几何法、基于优化。因为单纯的图搜索或基于采样的方法不满足车辆non-holonomic的约束以及轨迹不平滑等缺陷，常常是结合几何法和优化法一起使用。这里重点讨论几何法和优化法。
- 几何法逻辑简单，计算量小，已被较广泛的应用，但一般只针对平行或垂直Parking in车位的场景，部分参数需要结合实际场景调参，对场景的广泛适用性较差，且只输出路径（静态信息）。一般采用2段式，把路径分为“把车从车位开出”的圆弧段和“开出后停在车道上”的曲线段，并且在2段的

switch point处steering angle连续，保障可以一次操作完成，无需在车静止时转向。 

- 优化法计算量大，耗时长，难度高，好处在于轨迹平滑，对场景的限制较少。Baidu Apollo 5.0版本在其Open Space Planner中，参考了上述《Autonomous Parking Using Optimization-Based Collision Avoidance》和《Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments》，采用了先Hybrid A*规划，后数值优化的方法，解决Parking和U-Turn的问题。