



上海交通大学学位论文

基于四轮转向的泊车规划控制系统设计

姓 名： 赵泽华

学 号： 520021911201

导 师： 卢俊国

学 院： 电子信息与电气工程学院

专业名称： 自动化

申请学位层次： 学士

2024 年 05 月

A Dissertation Submitted to
Shanghai Jiao Tong University for Bachelor Degree

Design of Parking Planning and Control System
Based on Four-Wheel Steering

Author: Zhao Zehua
Supervisor: Lu Junguo

School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, P.R.China
May, 2024

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全知晓本声明的法律后果由本人承担。

学位论文作者签名：

日期： 年 月 日

上海交通大学

学位论文使用授权书

本人同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于：

☐ 公开论文

☐ 内部论文，保密 ☐ 1 年 / ☐ 2 年 / ☐ 3 年，过保密期后适用本授权书。

☐ 秘密论文，保密 ____ 年（不超过 10 年），过保密期后适用本授权书。

☐ 机密论文，保密 ____ 年（不超过 20 年），过保密期后适用本授权书。

（请在以上方框内选择打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

摘 要

家用车数量的增加正在导致停车环境变得复杂和拥挤，驾驶员难以处理，需要自动泊车系统进行辅助，而四轮转向车辆比前轮转向车辆拥有更短的转弯半径，更加适用于复杂的泊车环境。因此本文针对四轮转向车辆在复杂泊车环境下的泊车规划控制系统设计进行了深入探究，首先通过运动学和几何建模确立了四轮转向车辆的数学模型，并对停车场环境进行了建模，进而开发了高效的碰撞检测方法以简化车辆路径与障碍物的碰撞分析。本文提出了一种融合 Reeds-Shepp 曲线和 RRT*算法的泊车路径规划算法（RRT*-RS），并针对其高计算复杂度的问题，设计了基于解析法的快速规划方法，显著提高了泊车路径规划的效率。使用 Matlab 得到的仿真结果表明，四轮转向车辆在泊车效率和转弯半径上相较于前轮转向车辆具有明显优势，尤其在极限泊车情景下，四轮转向车辆能够以更高的适应性完成泊车任务。此外，本文还设计并验证了一套适用于低速泊车环境的四轮转向运动跟踪控制器，包括串级 PID 纵向控制器和改进的纯跟踪横向控制器，并通过 CarSim 与 Simulink 联合仿真测试了控制器的性能，仿真结果表明控制器可以很好的跟随路径。本文针对泊车问题提出了更加高效的碰撞检测分析方法以及路径规划方法，还对纯跟踪控制器进行了适用于四轮转向车辆的改进，不仅为四轮转向车辆的自动泊车提供了理论基础，而且为自动驾驶技术在泊车领域的应用与发展贡献了新的技术方案。

关键词：四轮转向车辆，自动泊车，路径规划，运动跟踪控制

ABSTRACT

The increase in the number of domestic cars is causing the parking environment to become complex and congested, which is difficult for drivers to deal with and requires the assistance of an automatic parking system, while four-wheel steering vehicles have a shorter turning radius than front-wheel steering vehicles, which is more suitable for complex parking environments. Therefore, this paper provides an in-depth investigation on the design of a parking planning control system for four-wheel steering vehicles in complex parking environments. Firstly, the mathematical model of four-wheel steering vehicles is established through kinematic and geometrical modelling, and the car park environment is modelled, and then a highly efficient collision detection method is developed to simplify the collision analysis of vehicle paths and obstacles. In this paper, a parking path planning algorithm (RT*-RS) incorporating Reeds-Shepp curves and RT* algorithm is proposed, and a fast planning method based on the analytical method is designed for its high computational complexity, which significantly improves the efficiency of parking path planning. The simulation results obtained using Matlab show that the four-wheel-steering vehicle has obvious advantages over the front-wheel-steering vehicle in terms of parking efficiency and turning radius, especially in the extreme parking scenarios, the four-wheel-steering vehicle is able to complete the parking task with higher adaptability. In addition, a set of four-wheel steering motion tracking controllers for low-speed parking environment, including serial PID longitudinal controllers and improved pure tracking transverse controllers, are designed and validated in this paper, and the performance of the controllers is tested through joint simulation of CarSim and Simulink, and the simulation results show that the controllers can follow the path well. This paper proposes a more efficient collision detection analysis method as well as a path planning method for the parking problem, and also improves the pure tracking controller for four-wheel steering vehicles, which not only provides a theoretical basis for the automatic parking of four-wheel steering vehicles, but also contributes a new technological solution for the application and development of the automatic driving technology in the field of parking.

Key words: four-wheel steering vehicles, automated parking, path planning, motion tracking control

目 录

摘 要.....I

ABSTRACT II

第一章 绪论 1

 1.1 研究背景与意义 1

 1.2 国内外研究现状 2

 1.2.1 路径规划算法国内外研究现状..... 2

 1.2.2 运动跟踪控制器国内外研究现状..... 5

 1.3 本论文主要研究内容 6

第二章 四轮转向车辆建模与泊车碰撞检测研究 8

 2.1 四轮转向车辆建模 8

 2.1.1 四轮转向车辆运动学建模..... 8

 2.1.2 四轮转向车辆几何模型建模..... 10

 2.2 停车场环境建模 13

 2.3 泊车碰撞检测分析 15

 2.3.1 线段与线段的碰撞检测分析..... 15

 2.3.2 线段与圆弧的碰撞检测分析..... 17

 2.3.3 圆弧与圆弧的碰撞检测分析..... 21

 2.4 本章小结 23

第三章 基于四轮转向的泊车路径规划算法设计 24

 3.1 满足运动学模型约束的 R-SHEPP 曲线 25

 3.2 基于随机采样与 REEDS-SHEPP 曲线的 RRT*算法 27

 3.3 基于解析法与 RRT*-RS 算法的快速规划方法 29

 3.3.1 垂直停车位倒车点的计算..... 31

3.3.2 平行停车位倒车点的计算.....	35
3.4 本章小结	39
第四章 基于四轮转向的运动跟踪控制器设计	41
4.1 纵向控制器设计	41
4.2 横向控制器设计	43
4.3 仿真模型搭建与控制器仿真实验	45
4.4 本章小结	50
第五章 全文总结	51
5.1 研究总结	51
5.2 研究展望	52
5.3 非技术性分析	52
参 考 文 献	54
致 谢	57

第一章 绪论

1.1 研究背景与意义

近年来，随着汽车行业的迅速发展以及国民生活水平的提高，越来越多的家庭选择汽车作为日常代步工具。然而，随着汽车数量的不断增加，城市停车位的供应无法满足需求，停车场的拥挤现象日益凸显，给驾驶员的停车体验带来了许多困难和问题。在这种下，自动泊车辅助系统应运而生，它的出现不仅能够有效减轻驾驶员的负担，还能提升在复杂环境中泊车的安全性，从而为驾驶员提供更加便利和舒适的停车体验。



图 1-1 美国汽车工程师学会最新发布的驾驶自动化分级图例

关于自动驾驶系统,美国汽车工程师学会(Society of Automotive Engineers, SAE)在 2021 年 4 月 30 日发布的标准中^[1],将自动驾驶系统按驾驶员参与程度由高到低分

为了六个等级，并记为 L0 至 L5。L0 为无自动驾驶（No Driving Automation），即完全由人手动驾驶，系统仅能提供一些警告或瞬时协助，如自动紧急制动，车道偏离警告等，如今市面上绝大部分车辆都属于这个等级；L1 为辅助驾驶（Driver Assistance），它可以提供转向或加速支持，如车道居中或自适应巡航控制；L2 为部分自动驾驶（Partial Driving Automation），相较于 L1，它可以同时提供转向和加速功能，也就是说驾驶操作已主要由车辆接管，驾驶员仅负责除加减速和转向之外的其他驾驶操作以及对周边环境进行监控，确保在特殊情况下可以及时接管驾驶系统，目前市面上如小鹏、蔚来、华为等许多国产的量产电动汽车都已具备这一功能；L3 为有条件的自动驾驶（Conditional Driving Automation），从这个等级开始，自动驾驶系统正式进入了车辆自动驾驶而非驾驶员驾驶的阶段，此时周边环境也由车辆进行监控，并在发生紧急情况时向驾驶员发送请求，此时需要驾驶员接管车辆，目前本田、奔驰、特斯拉等品牌的部分车型已达到 L3，2024 年上半年发售的小米 su7 也已实现 L3 级别的自动驾驶功能；L4 为高度自动驾驶（High Driving Automation），这个等级的自动驾驶不会要求驾驶员接管车辆，驾驶员无需再保持注意力，但应用场景仅限于一些限定道路与环境条件；L5 为完全自动驾驶（Full Driving Automation），它相较于 L4，可以在任意情况下自动驾驶且不会要求驾驶员接管，是自动驾驶的最高水平。

目前，大多数家用汽车采用阿克曼模型，也称为前轮转向模型。这种车型仅有前轮具备转向功能，其转弯半径较大，导致在拥挤或复杂的停车环境中泊车困难，并需要更多的空间才能顺利完成泊车。然而，四轮转向模型的四个车轮均具备转向功能，能够有效减小车辆的转弯半径，因而减少了泊车所需的空间。这种模型不仅提高了停车位的利用率，还增强了泊车过程的舒适性。此外，四轮转向模型还能够处理更加复杂和拥挤的泊车环境，表现出更强的适应性。

1.2 国内外研究现状

1.2.1 路径规划算法国内外研究现状

在运动规划方面，余卓平等人对近年来自动驾驶运动规划算法进行了综述^[2]。运动规划分为了路径规划和轨迹规划两大类，其中，路径规划是生成一条可行的，不与障碍物发生碰撞的路径，轨迹规划则是要生成一条满足车辆运动学模型的路径，其中，路径规划又进一步分为基于随机采样的算法和基于网格搜索的算法两种算法。

基于随机采样的算法通过在空间中随机采样来获得一个包含起始节点和目标节点的连通图，再利用碰撞检测来判断可行性，从而得到可行轨迹。KAVRAKI L 等人提出了概率路图法（probabilistic roadmaps, PRM）^[3]，这个算法首先在空间中随机取若干不与障碍物碰撞的点，再将每个采样点与其附近一定距离的其他采样点相连，去除掉其中与障碍物发生碰撞的路径，最后再用搜索算法找出可行解。LAVALLE S 等人提出了快速随机扩展树法（rapidly random tree, RRT）^[4]，它首先定义一个以起始点为根节点的搜索树，再在空间中随机选取一个采样点，找出树中离它最近的点，定义为最近点，若采样点与最近点之间的距离小于事先规定好的步长，则将二者相连，并将采样点作为新节点，和路径一起加入搜索树中，若采样点与最近点之间的距离大于步长，则从最近点向着采样点方向移动步长距离，将此时的位置作为新节点，和路径一起加入搜索树中。当新节点与目标点之间的距离小于步长时，就可以直接将新节点与目标点相连加入搜索树中，此时从目标点开始不断地访问父节点就可以得到一条从起始点到目标点的可行路径。PRM 和 RRT 算法是两个最为经典的基于随机采样的算法，它们可以找出可行解，但运算效率并不高，得出的路径也不一定是最优路径。KUFFNER J 等人提出了 RRT-Connect 算法^[5]，它分别以起点和目标点为根节点构建两个搜索树，当两个树相遇时便能找到可行解。URMSON C 等人提出了启发式 RRT（heuristic RRT）算法^[6]，即先赋予每个节点一个代价值，这个代价值可以选取该节点到目标点的距离，也可以选取该节点到障碍物的距离，再利用启发式函数增加代价低的节点被采样的概率。CHENG P 等人提出了决策完备 RRT（resolution complete RRT, RC-RRT）^[7]，当在扩展时，若某个节点与障碍物发生了碰撞，则增加它的惩罚值，在后续采样过程中，惩罚值高的节点被采样的概率小，这样可以有效减少障碍物附近节点的扩展。FERGUSON D 等人提出了 anytime RRTs^[8]，当 RRT 找出可行解后，算法继续进行，比较新采样的节点，选出比原路径更好的节点加入路径，从而使路径可以不断优化。KARAMAN S 提出了 RRT*算法^[9]，它在 RRT 算法找到新节点后，以新节点为圆心，以一个事先规定好的、大于步长的距离为半径，得到一个圆形区域，并找到该圆形区域内所有的节点，比较新节点分别以这些节点为父节点的情况下，得到的路径代价是否比原路径更小，找出代价最小的一个节点作为新节点新的父节点并加入搜索树中，而后再判断这些节点分别以新节点为父节点的情况下得到的路径代价是否比原先的路径更小，若更小，则将新节点作为该节点新的父节点加入到搜索树中，这样一来可以使得最后得到的路径更优。RRT*算法也是目前最常用的路径规划算法之一。GAMMELL J D 等人在 RRT*算法的基础上提出了 informed RRT*算法^[10]，在 RRT*

算法得出一个可行解后,将区域缩小为以起点和目标点为焦点,并包含这个解的椭圆区域内,再进行后续的优化,这样可以避免对整个地图进行随机采样,减少不必要的采样点的选取,从而提高效率。

基于网络搜索的算法则是将空间离散成一个图,通过搜索算法找出其中的最优解。基于网格搜索的算法可以找出最优路径。其中最为经典的算法就是 HART P E 等人提出的 A*算法^[11],它在每个节点的代价中加入了启发式函数来提高靠近目标点的节点的优先级,通过不断扩展优先级最高的节点来得到最优解,常用的启发式函数为该节点到目标点的距离,如果节点扩展方向为四个方向,则代价函数可以选取曼哈顿距离,如果扩展方向为八个方向,则可以选取欧拉距离。FURCY D A 等人提出 Weight A*算法^[12],通过提高启发式函数的权重来使得启发式函数小的节点优先被扩展,从而更快找出最优解。AINE S 等人提出了 Multi-heuristic A* (MHA) 算法^[13],即使用多个启发式函数,只需保证其中一个启发式函数可以得到最优解,就可以通过改变各个启发式函数的权重来同时保证效率和最优性。与 RRT-Connect 算法类似,ISLAM F 等人提出了 A*-Connect 算法^[14],即从起点和目标点同时开始搜索,从而提高运算速率。KOENIG S 等人提出了 Lifelong Planning A* (LPA) 算法^[15],它在环境发生改变时可以做到仅重新搜索少部分节点就找到新的最优解。

轨迹规划则是生成一条平滑、连续的,符合小车自身运动学模型约束的路径。DUBINS L E 提出了 Dubins 曲线来得到一条从起点到目标点的满足约束的最短路径^[16],该路径由直线和最大曲率的曲线拼接而成,并且满足位姿要求。REEDS J 和 SHEPP L 二人则在 Dubins 曲线的基础上提出了 Reeds-Shepp (RS) 曲线^[17],该曲线加入了倒车的环节,即小车可以通过前进后退两个运动方向来抵达目标点,在大多数情况下可以得到比 Dubins 曲线更优的路线,但这两种曲线在不同路段前后曲率不一致,需要在每个拼接处停车来改变车轮方向。WILDE D K 等人提出了回旋线^[18],回旋线的各处曲率与已经过的弧长成正比,从而保证曲率的连续性,因此可以用回旋线替换 Dubins 曲线和 RS 曲线的拼接点。KOMORIYAK 等人提出了 B 样条曲线^[19],通过选取几个额外的节点来得到一系列曲率连续的贝塞尔曲线,将其拼接起来得到连接起点和目标点的 B 样条曲线,B 样条曲线虽然曲率连续,但没有界,可能会超出车辆最大曲率,需要通过合理选取节点位置来控制曲率大小。

1.2.2 运动跟踪控制器国内外研究现状

在运动控制方面，熊璐等人对近年来自动驾驶运动控制的发展进行了综述^[20]，将运动控制分为了纵向运动控制、路径跟踪控制和轨迹跟踪控制三个部分，并介绍了各部分近年来常用的一些控制方法。

纵向运动控制主要是通过驱动和制动控制器来控制车辆加速度，从而实现对车速或位置的控制。纵向运动控制器一般分上下两层，上层控制器根据位置或速度闭环得到期望的加速度和转矩，下层控制器通过驱动和制动来跟踪上层期望的加速度和转矩。经典的控制方法有：PID 控制，即通过比例、积分、微分环节来进行控制，是最简单实用的控制方法，如斯坦福大学的 Stanley 无人车使用的就是简单的 PI 控制^[21]；线性二次型调节器（Linear Quadratic Regulator, LQR）控制，即给出系统的性能函数，并根据性能函数设计优化问题，从而设计出相应的控制器，使得系统稳定且性能函数取到极值，如 XU S 等人使用的基于最优预瞄控制的纵向车速跟踪方法^[22]，将最优控制问题变为 LQR 问题，并计算出了优化问题的解析解，提高了控制精度；模型预测控制（Model Predictive Control, MPC），常用于处理多输入多输出系统，通过反复预测并优化来进行控制，如 ZHU M 等人就是使用 MPC 设计上层控制器来得到期望加速度^[23]；内模控制（Internal Model Control, IMC），即在系统中加入一个内部模型，通过与实际输出比较来进行控制可以有效的补偿扰动，防止车辆速度发生波动，如 WANG J 等人设计了内模控制器来对加减速进行控制^[24]。

路径跟踪控制是控制车辆上的一个点跟踪一条给定的与时间无关的曲线，使得小车的侧向误差和方向角误差趋近于 0。常用的控制方法有：基于几何学模型的控制，如以后轴中心点为控制点的纯跟踪（Pure Pursuit）算法^[25]和以前轴中心点为控制点的 Stanley 算法^[21]；PID 控制，如 MARINO R 等人使用串级 PID 控制^[26]，其中使用了 PI 控制内环；LQR 控制，如 SNIDER J M 提出在 LQR 的基础上加入前馈控制消除扰动，再利用预瞄来减小超调^[27]；滑模控制（Sliding Mode Control, SMC），先建立一个滑模面，利用控制器将系统强行拉到滑模面上，使系统沿着滑模面运动，如余卓平等人设计的抗积分饱和的差动转向车辆动力学滑模控制器^[28]；MPC，如 FALCONE P 等人所设计的基于非线性模型预测控制器以及线性时变模型预测控制器^[29]；鲁棒控制（Robust Control），是基于鲁棒控制理论设计的控制器，可以实现当环境出现扰动时可以保持系统原有性能，如宋彦等人所设计的基于 μ 综合横摆稳定控制方法^[30]；自抗扰控制（Active Disturbance Rejection Control, ADRC），是将所有扰动项和不确定项通过状态观测器进行估计并补偿，如 CHU Z 等人利用 ADRC 设计的转向控制器^[31]。

轨迹跟踪控制相比于路径跟踪控制，则是让选取的控制点跟随一条时变的路线运动，因此轨迹跟踪控制要同时考虑纵向运动与横向运动，使二者误差同时收敛 0。由此自然而然地就能分出两种控制方案：一种是将纵向运动控制和横向路径跟随分开来设计控制器，即分散式控制，一种是设计一个全局控制器来统一控制纵向运动和横向运动，即集中式控制。分散式控制在控制方法上就是之前提到的纵向运动控制和路径跟踪控制的方法，重点是如何将二者解耦，对此，YU Z 等人推导出了车辆纵侧向误差动力学方程，得出了二者可以近似解耦的条件，从而可以对二者进行分别设计^[32]。而集中式控制则需要建立车辆运动学位姿误差微分方程，而后在利用前面提到过的各种控制方法来对这进行统一控制，如 ALCALAE 等人使用了 LQR 控制^[33]，但仅考虑了运动学上的耦合而忽略了动力学上的耦合，LIM E 等人使用了动态面滑模控制^[34]，但实际情况下映射关系并不准确，GAO Y 则使用了 MPC^[35]，将纵向控制和横向控制转化为同一个带约束的优化问题，使得纵向控制器与横向控制器耦合。

1.3 本论文主要研究内容

本论文主要针对四轮转向车辆在泊车环境下进行路径规划与控制器设计，并与前轮转向车辆进行对比，研究四轮转向车辆在泊车环境下的优势。各章节间逻辑关系如图 1-2 所示。

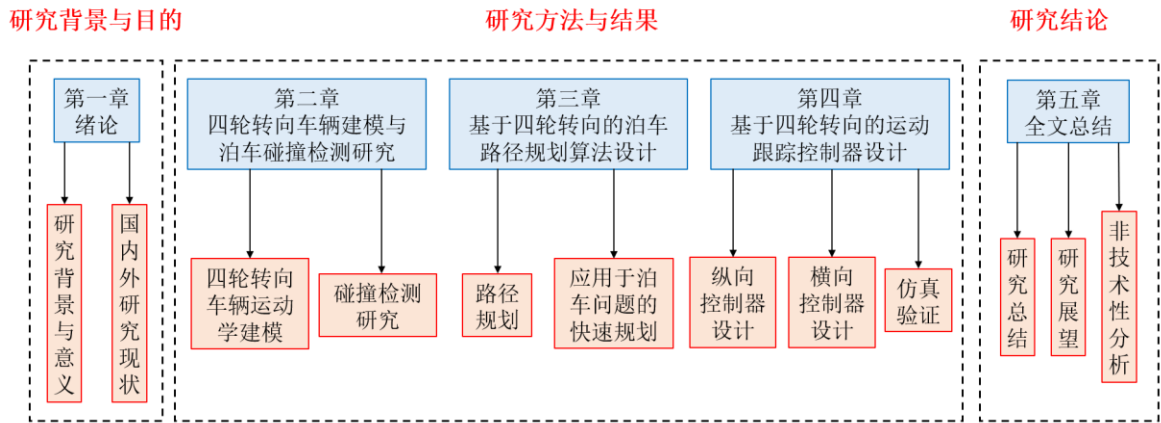


图 1-2 各章节逻辑关系示意图

具体研究内容如下：

(1) 对四轮转向车辆进行运动学建模。自动驾驶系统的路径规划部分需要满足车辆运动学模型的约束，控制器设计则需要根据运动学模型来搭建控制系统，因此本

论文首先对四轮转向车辆进行运动学建模。目前绝大部分自动驾驶研究围绕前轮转向车辆展开,因此本论文在前轮转向车辆运动学模型求解的基础上,引入后轮的转向角,从而求解出四轮转向车辆的运动学模型。

(2) 泊车碰撞检测研究。首先对车辆以及停车场障碍物建模,将其抽象成若干简单的几何模型,再进行泊车碰撞检测研究,将所有碰撞情况分为三种情况:线段与线段的碰撞检测研究、线段与圆弧的碰撞检测研究以及圆弧与圆弧的碰撞检测研究。

(3) 基于四轮转向的泊车路径规划算法设计。由于后续研究采用点云地图而非栅格地图,难以部署 A* 算法,因此路径规划部份选用了目前较为成熟的 RRT* 算法,而由于泊车会涉及倒车入库,而 Reeds-Shepp 曲线相较于 Dubins 曲线更适用于泊车环境,因此轨迹规划部分则选用了 Reeds-Shepp 曲线。而针对泊车环境本身的低复杂性,本论文提出了一种通过解析法来降低代码运算复杂度的方法,大大提高了路径规划的运算效率。

(4) 基于四轮转向的运动跟踪控制器设计。泊车环境是一种低速环境,因此纵向控制器与横向控制器可以解耦,从而对两者分别进行设计。纵向控制部分要对小车速度进行控制,其中包括前进与倒车两部分路径,需要将两段路径分开,再在每段路径上单独使用串级 PID 控制。横向控制部分则希望小车跟随路径,低速环境下可以使用几何学模型的路径跟踪控制方案,由于 Stanley 算法只适用于前轮转向,因此本论文采用纯跟踪算法,并把控制点从后轴中心移动到了车辆几何中心,使得控制策略更适用于四轮转向车辆。最后再使用 Simulink 与 CarSim 联合仿真来验证控制器设计的可行性。

第二章 四轮转向车辆建模与泊车碰撞检测研究

2.1 四轮转向车辆建模

2.1.1 四轮转向车辆运动学建模

四轮转向车辆模型机械结构由车体和前后四个转向轮和驱动轮组成,相较于前轮转向模型而言,多了后轮的转向角,此时后轮不仅是驱动轮,也是转向轮,因此建模过程可以在前轮转向车辆运动学建模的基础上加上后轮的转向角。

在笛卡尔坐标系下,笛卡尔坐标系下,四轮转向移动机器人小车运动模型如图 2-1 所示。小车重心 CG 坐标为 (x, y) ,后轮中心坐标为 (x_1, y_1) ,前轮中心坐标为 (x_2, y_2) 。

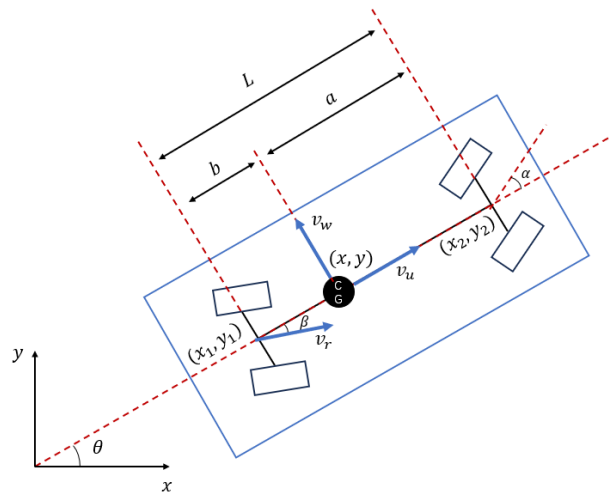


图 2-1 四轮转向车辆运动学模型

这里规定逆时针方向为正方向,顺时针方向为负方向。

定义后轮中心到前轮中心的轴线为小车中轴线,其与 x 轴夹角为方位角 θ ,小车前轮方向与中轴线夹角为 α ,后轮方向与中轴线夹角为 β 。重心 CG 到前轮中心距离为 a ,到后轮中心距离为 b ,前轮中心与后轮中心距离,即轴距为 L 。

定义小车中轴线为 u 轴,其法向为 w 轴,则小车质心法向速度为 v_w ,小车质心轴向速度为 v_u ,后轮速度为 v_r 。

则后轮与 CG 的坐标关系为：

$$\begin{aligned} x_1 &= x - b \cos \theta \\ y_1 &= y - b \sin \theta \end{aligned} \quad (2-1)$$

后轮与车辆的速度关系为：

$$\begin{aligned} \dot{x}_1 &= \dot{x} + b\dot{\theta}\sin\theta \\ \dot{y}_1 &= \dot{y} - b\dot{\theta}\cos\theta \end{aligned} \quad (2-2)$$

在泊车环境下，我们认为车辆始终处于低速状态，并且低速状态下车辆轮胎不会发生侧滑。因此由后轮不会侧滑的特性得：

$$\frac{\dot{x}_1}{\dot{y}_1} = \frac{\sin(\theta + \beta)}{\cos(\theta + \beta)} \quad (2-3)$$

由式 (2-1) (2-2) (2-3) 可得：

$$\dot{x}\sin(\theta + \beta) - \dot{y}\cos(\theta + \beta) + b\dot{\theta}\cos\beta = 0 \quad (2-4)$$

同理，前轮与 CG 的坐标关系为：

$$\begin{aligned} x_2 &= x + a \cos \theta \\ y_2 &= y + a \sin \theta \end{aligned} \quad (2-5)$$

前轮与车辆的速度关系为：

$$\begin{aligned} \dot{x}_2 &= \dot{x} - a\dot{\theta}\sin\theta \\ \dot{y}_2 &= \dot{y} + a\dot{\theta}\cos\theta \end{aligned} \quad (2-6)$$

由前轮不会侧滑的特性得：

$$\frac{\dot{x}_2}{\dot{y}_2} = \frac{\sin(\theta + \alpha)}{\cos(\theta + \alpha)} \quad (2-7)$$

由式 (2-5) (2-6) (2-7) 可得：

$$\dot{x}\sin(\theta + \alpha) - \dot{y}\cos(\theta + \alpha) - a\dot{\theta}\cos\alpha = 0 \quad (2-8)$$

此外，车辆在 x 、 y 方向的速度为：

$$\begin{aligned} \dot{x} &= v_u \cos \theta - v_w \sin \theta \\ \dot{y} &= v_u \sin \theta + v_w \cos \theta \end{aligned} \quad (2-9)$$

将式 (2-4) (2-8) (2-9) 联立，得到四轮转向车辆的质心运动学模型为：

$$\begin{aligned} \dot{x} &= v_u \left(\cos \theta - \frac{b}{L} \tan \alpha - \frac{a}{L} \tan \beta \right) \\ \dot{y} &= v_u \left(\sin \theta + \frac{b}{L} \tan \alpha + \frac{a}{L} \tan \beta \right) \\ \dot{\theta} &= \frac{v_u}{L} (\tan \alpha - \tan \beta) \end{aligned} \quad (2-10)$$

因此四轮转向车辆的转弯半径为：

$$R = \frac{v_u}{\dot{\theta}} = \frac{L}{\tan\alpha - \tan\beta} \quad (2-11)$$

四轮转向车辆的转向曲率为：

$$\kappa = \frac{1}{R} = \frac{\tan\alpha - \tan\beta}{L} \quad (2-12)$$

这样我们就得到了四轮转向车辆的运动学模型以及四轮转向车辆转弯半径与转向曲率与车辆前后轮转向角和前后轴距之间的关系。

而在泊车场景下，我们认为四轮转向车辆的重心位于它的几何中心上，即 $a = b = L/2$ ，并且前后轮转向角大小相同，方向相反，因此可以以前轮转向角为主，记为 δ ，则泊车场景下的四轮转向车辆的质心运动学方程可以改写为：

$$\begin{aligned} \dot{x} &= v_u \cos\theta \\ \dot{y} &= v_u \sin\theta \\ \dot{\theta} &= \frac{2v_u \tan\delta}{L} \end{aligned} \quad (2-13)$$

泊车场景下的四轮转向车辆的转弯半径可以重新表达为：

$$R = \frac{L}{2\tan\delta} \quad (2-14)$$

而在以往的研究当中，前轮转向车辆的转弯半径表达式为：

$$R = \frac{L}{\tan\delta} \quad (2-15)$$

因此不难看出，四轮转向车辆相较于前轮转向车辆具有更小的转弯半径，可以应对更为复杂的泊车环境。

2.1.2 四轮转向车辆几何模型建模

在对四轮转向车辆进行路径规划时，我们会将车辆看作位于控制点上的一个点，将路径轨迹看成一条线，而在实际泊车过程中，车辆存在碰撞体积，需要考虑小车外轮廓是否会与障碍物发生碰撞而导致路径无法完成。为了便于后续对车辆进行碰撞检测研究，我们需要对车辆几何模型进行建模。

首先我们将车辆外轮廓抽象成一个矩形，如图 2-2 所示。

其中车辆重心为 CG，选在车辆几何中心，在全局笛卡尔坐标系中的坐标为 (x, y) ，当前车辆方位角为 θ ，前后轮轴距重心长度均为 $L/2$ 。车辆长度为 L_c ，宽度为 W_c 。

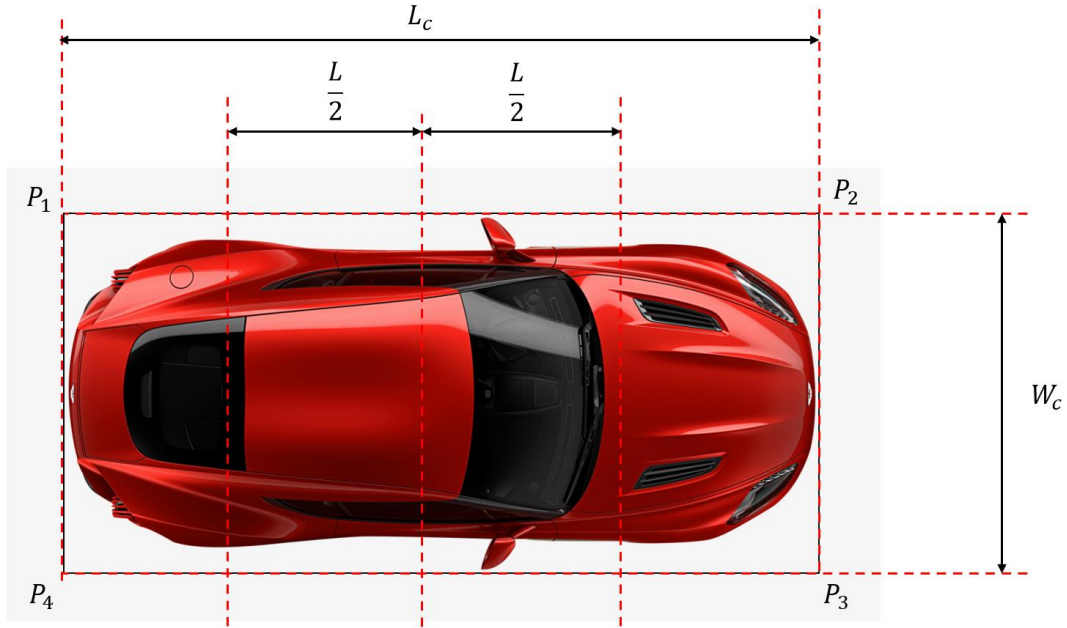


图 2-2 车辆几何模型

这样一来，车辆几何模型便被抽象成了矩形 $P_1P_2P_3P_4$ ，四个顶点坐标分别为：

$$\begin{aligned}
 (x_1, y_1) &= (x - \frac{L_c}{2} * \cos\theta - \frac{W_c}{2} * \sin\theta, y - \frac{L_c}{2} * \sin\theta + \frac{W_c}{2} * \cos\theta) \\
 (x_2, y_2) &= (x - \frac{L_c}{2} * \cos\theta + \frac{W_c}{2} * \sin\theta, y - \frac{L_c}{2} * \sin\theta - \frac{W_c}{2} * \cos\theta) \\
 (x_3, y_3) &= (x + \frac{L_c}{2} * \cos\theta + \frac{W_c}{2} * \sin\theta, y + \frac{L_c}{2} * \sin\theta - \frac{W_c}{2} * \cos\theta) \\
 (x_4, y_4) &= (x + \frac{L_c}{2} * \cos\theta - \frac{W_c}{2} * \sin\theta, y + \frac{L_c}{2} * \sin\theta + \frac{W_c}{2} * \cos\theta)
 \end{aligned} \tag{2-16}$$

下面定义车辆在转弯时的转弯圆心。以左转为例，在过小车重心 CG，垂直于小车当前方位角的方向上，车辆左侧距离 CG 长度为 R 的位置上的点定义为转弯圆心，位置如图 2-3 所示。

此时不难看出，若以目前的建模方式，车辆在左转时，右前角会比车辆右边缘额外多扫过一块区域，这种情况下，在进行后续的碰撞检测研究时，就不能简单地以转弯半径加上半个车宽作为外侧扫过区域半径，会导致碰撞检测的工作变得复杂，计算量变大，因此我们可以通过改变车辆的几何模型来使得多出的这一小部分可以忽略不计。

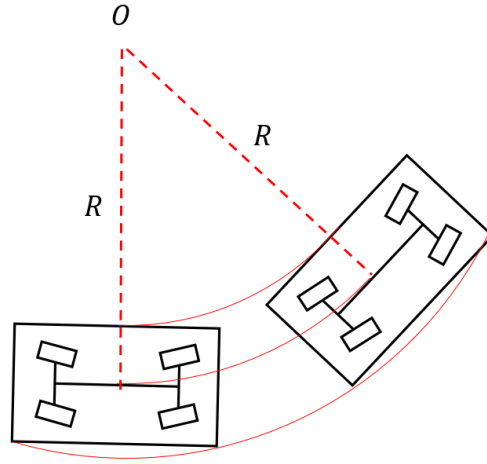


图 2-3 小车左转时的转弯圆心

首先计算出这一区域超出车辆原本几何模型边缘的最大距离 d_{max} 为：

$$d = \sqrt{\left(\frac{L_c}{2}\right)^2 + \left(\frac{L}{2\tan\delta_{max}} + \frac{W_c}{2}\right)^2} - \left(\frac{L}{2\tan\delta_{max}} + \frac{W_c}{2}\right) \quad (2-17)$$

因此我们可以在原先几何模型的基础上，将其宽度向两侧各延伸 d_{max} ，得到的新的简化模型如图 2-4 所示。

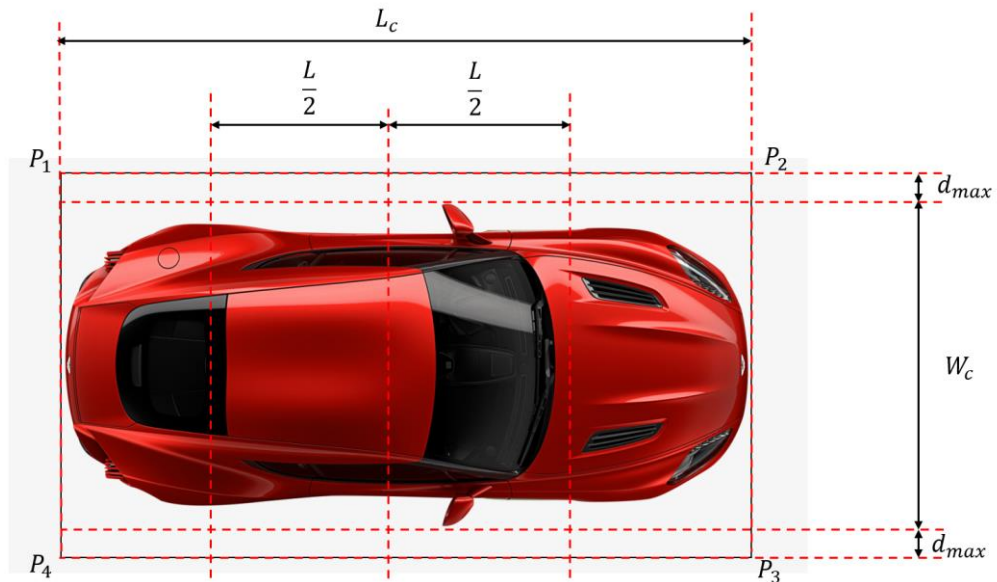


图 2-4 修正后的车辆几何模型

这样一来，四个顶点的坐标就变为：

$$\begin{aligned}
 (x_1, y_1) &= (x - \frac{L_c}{2} * \cos\theta - (\frac{W_c}{2} + d_{max}) * \sin\theta, y - \frac{L_c}{2} * \sin\theta + (\frac{W_c}{2} + d_{max}) * \cos\theta) \\
 (x_2, y_2) &= (x - \frac{L_c}{2} * \cos\theta + (\frac{W_c}{2} + d_{max}) * \sin\theta, y - \frac{L_c}{2} * \sin\theta - (\frac{W_c}{2} + d_{max}) * \cos\theta) \\
 (x_3, y_3) &= (x + \frac{L_c}{2} * \cos\theta + (\frac{W_c}{2} + d_{max}) * \sin\theta, y + \frac{L_c}{2} * \sin\theta - (\frac{W_c}{2} + d_{max}) * \cos\theta) \\
 (x_4, y_4) &= (x + \frac{L_c}{2} * \cos\theta - (\frac{W_c}{2} + d_{max}) * \sin\theta, y + \frac{L_c}{2} * \sin\theta + (\frac{W_c}{2} + d_{max}) * \cos\theta)
 \end{aligned}
 \tag{2-18}$$

后续的碰撞检测分析时，车辆路径模型可以简化为若干线段与圆弧的组合，从而简化碰撞检测分析。

2.2 停车场环境建模

由于停车环境日益复杂，乱停乱放现象增多，直接对原始停车场环境进行泊车碰撞检测较为复杂，为了简化碰撞检测分析，我们需要对停车场环境进行建模。我们需要找出其他车辆作占据的网格，并将其框选视作障碍物，用多条线段组合来表示障碍物边界。

理想的垂直停车位的环境模型如图 2-5、2-6 所示，此时所有车辆有序停放，浅黄色区域表示该环境下的停车位的模型。



图 2-5 理想的垂直停车位模型

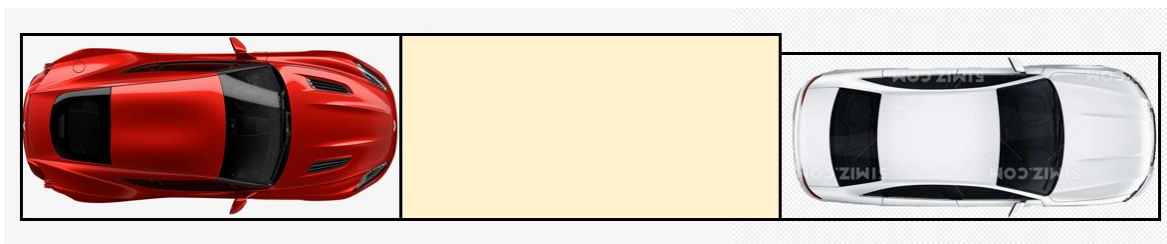


图 2-6 理想的平行停车位模型

而不理想的垂直停车位的环境模型如图 2-7、2-8 所示，浅黄色区域表示该环境下的停车位的模型，此时部分车辆乱停乱放，导致停车空间减小，泊车变得困难。

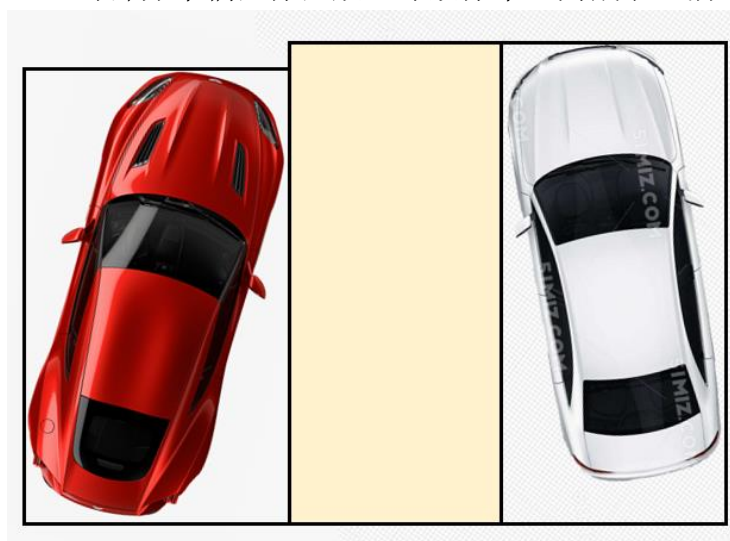


图 2-7 非理想的垂直停车位模型

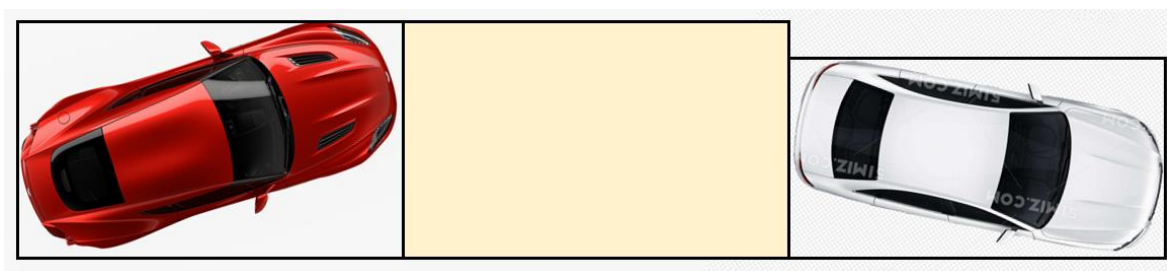


图 2-8 非理想的平行停车位模型

此时所有障碍物都简化为了矩形，后续碰撞检测时只需要判断车辆路径与这些矩形是否发生碰撞。

2.3 泊车碰撞检测分析

路径规划时，要确保得到可行路径，需要保证路径与障碍物不会发生碰撞，因此需要进行碰撞检测分析来判断车辆路径与障碍物是否发生碰撞。

前文已经将车辆路径与障碍物模型简化为线段和圆弧的组合，因此碰撞检测研究只需要检测属于车辆路径的线段和圆弧是否与属于障碍物的线段与路径发生碰撞，因此可以将碰撞检测分析分为三大类：线段与线段的碰撞检测分析、线段与圆弧的碰撞检测分析以及圆弧与圆弧的碰撞检测分析。其中，线段与线段的碰撞检测分析使用目前常用的快速排斥-跨立实验，而线段与圆弧的碰撞检测分析和圆弧与圆弧的碰撞检测分析目前没有已有的系统性的方法，因此我受快速排斥-跨立实验启发，即用简单的几何关系来快速排除部分不碰撞的情况，从而避免过多进行解析运算，从而提高代码效率

2.3.1 线段与线段的碰撞检测分析

线段间的碰撞检测可以通过已有的快速排斥-跨立实验来进行。首先将两条线段补成平行于坐标系的两个矩形，如图 2-9 所示，若两个矩形不碰撞，即可确定两条线段不碰撞。

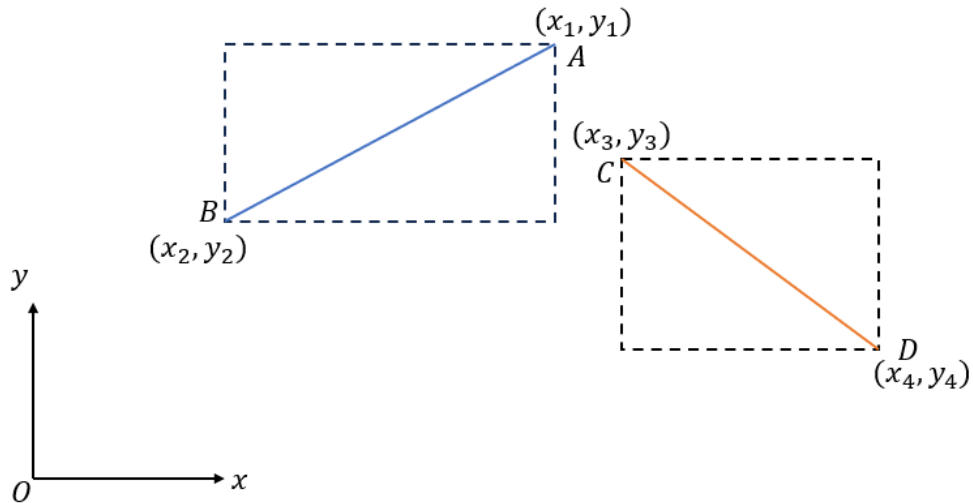


图 2-9 两条线段满足快速排斥且不碰撞

计算方法为两条线段的端点坐标只需要满足以下任意一个不等式即可：

$$\begin{aligned}
\max(x_1, x_2) &\leq \min(x_3, x_4) \\
\max(x_3, x_4) &\leq \min(x_1, x_2) \\
\max(y_1, y_2) &\leq \min(y_3, y_4) \\
\max(y_3, y_4) &\leq \min(y_1, y_2)
\end{aligned} \tag{2-19}$$

若两个矩形碰撞，即不满足式（2-19）时，则要进行下一步跨立实验来判断两条线段是否碰撞。

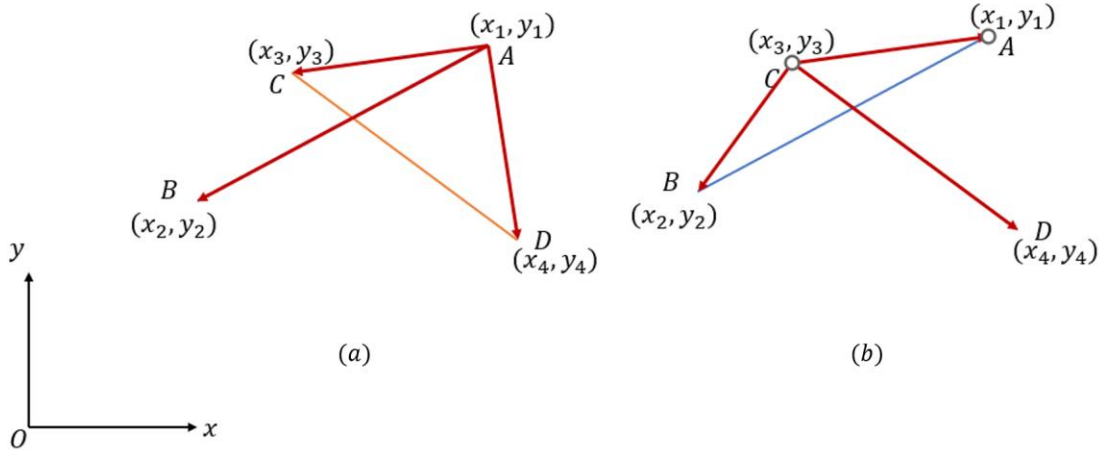


图 2-10 两条线段满足跨立实验且碰撞

从图 2-10 中，我们可以看出，若两条线段相交，那么每条线段的两个端点应居于另一条线段的两侧，即如图 2-10（a）中，向量 \overrightarrow{AC} 、 \overrightarrow{AD} 位于向量 \overrightarrow{AB} 的两侧，图 2-10（b）中，向量 \overrightarrow{CA} 、 \overrightarrow{CB} 位于向量 \overrightarrow{CD} 的两侧。用数学的语言表示，则是 \overrightarrow{AC} 与 \overrightarrow{AB} 的叉积、 \overrightarrow{AD} 与 \overrightarrow{AB} 的叉积方向相反， \overrightarrow{CA} 与 \overrightarrow{CD} 、 \overrightarrow{CB} 与 \overrightarrow{CD} 的叉积方向相反，即两条线段同时满足不等式（2-20）（2-21）：

$$(\overrightarrow{AB} \times \overrightarrow{AC}) \cdot (\overrightarrow{AB} \times \overrightarrow{AD}) \leq 0 \tag{2-20}$$

$$(\overrightarrow{CD} \times \overrightarrow{CA}) \cdot (\overrightarrow{CD} \times \overrightarrow{CB}) \leq 0 \tag{2-21}$$

计算方法为两条线段的端点坐标需要同时满足不等式（2-22）（2-23）：

$$\begin{aligned}
&[(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)] * \\
&[(x_2 - x_1)(y_4 - y_1) - (y_2 - y_1)(x_4 - x_1)] \leq 0
\end{aligned} \tag{2-22}$$

$$\begin{aligned}
&[(x_4 - x_3)(y_1 - y_3) - (y_4 - y_3)(x_1 - x_3)] * \\
&[(x_4 - x_3)(y_2 - y_3) - (y_4 - y_3)(x_2 - x_3)] \leq 0
\end{aligned} \tag{2-23}$$

利用这两步判断方法就可以快速判断两条线段是否碰撞。

快速排斥-跨立实验从方法上看是利用两条线段简单的几何关系来快速排除掉两条线段不发生碰撞的一种情况，再对剩余的情况运用解析法来进行计算，从而判断两条线段是否发生碰撞。因此受快速排斥-跨立实验的启发，我认为线段与圆弧的碰撞检测分析、圆弧与圆弧的碰撞检测分析都可以用简单的几何关系排除一些一定不会发生碰撞的情况，再利用解析法去判断剩余的情况，加快运算效率。

2.3.2 线段与圆弧的碰撞检测分析

直接判断线段与圆弧是否碰撞比较困难，因此我们可以先将圆弧补全为圆，判断圆与线段的碰撞关系，若圆与线段不发生碰撞，那么圆弧和线段一定不会发生碰撞，因此可以排除一些线段与圆弧不发生碰撞的情况，在完成圆与线段的碰撞检测分析后，在对其中圆与线段发生碰撞的情况进行圆弧与线段的碰撞检测分析，从而提高碰撞检测分析效率。

首先将圆弧补全为一个完整的圆，再来判断圆与线段两个端点之间的位置关系。

若线段的两个端点都在圆内，如图 2-11 (a) 所示，则线段与圆不发生碰撞；若线段的两个端点一个在圆内，一个在圆外，如图 2-11 (b) 所示，则线段与圆发生碰撞。

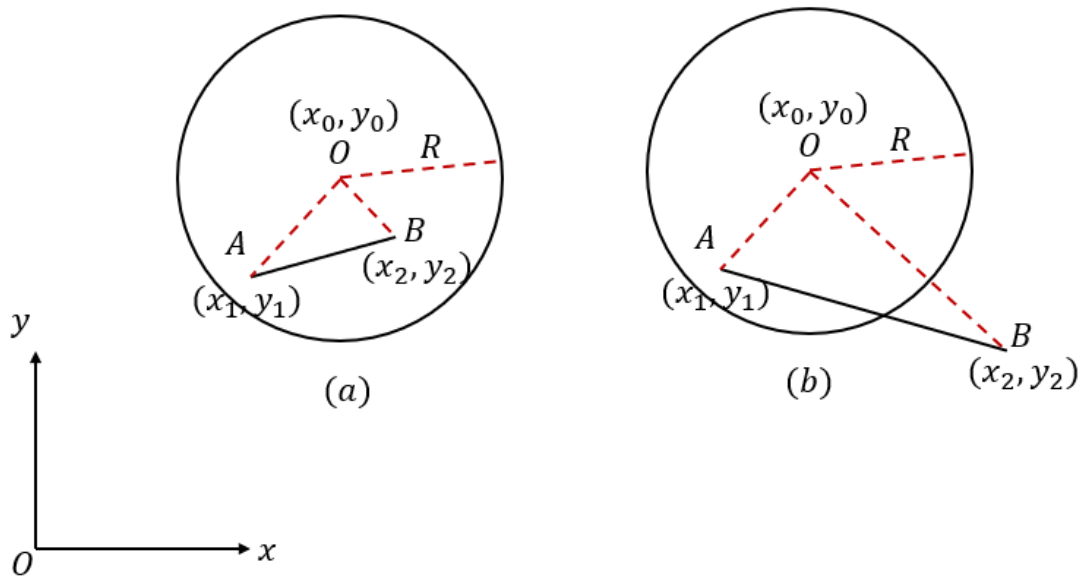


图 2-11 根据线段端点与圆的位置关系判断碰撞情况

用线段的端点坐标 (x_1, y_1) 、 (x_2, y_2) ，圆的圆心坐标 (x_0, y_0) 与半径 R 来表达上述的

碰撞情况：若满足式

$$\begin{aligned}\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} &\leq R \\ \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} &\leq R\end{aligned}\quad (2-24)$$

则圆和线段不发生碰撞，若满足式

$$\left(\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} - R\right)\left(\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} - R\right) < 0 \quad (2-25)$$

则圆和线段发生碰撞。

若线段的两个端点都在圆外，则需要将圆的圆心 O 与线段的两个端点 A 、 B 相连，并通过 $\angle OAB$ 与 $\angle OBA$ 来判断圆与线段的碰撞关系。

若 $\angle OAB$ 和 $\angle OBA$ 中有一个为钝角，如图 2-12 所示，则圆与线段不发生碰撞。

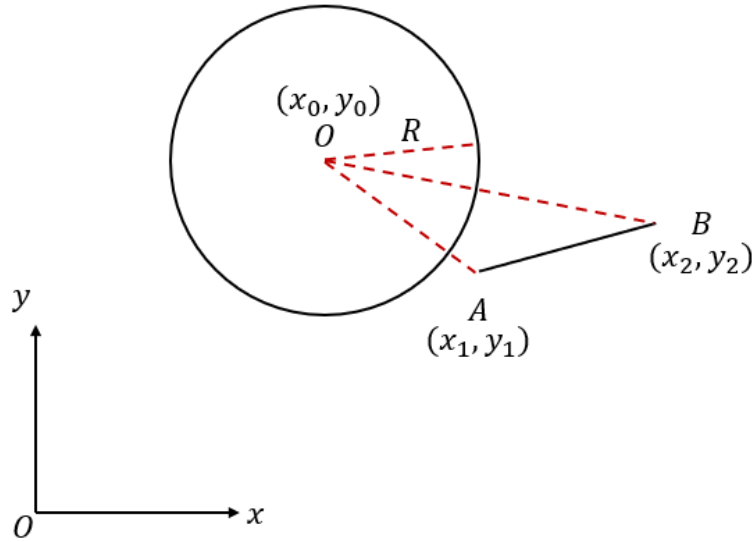


图 2-12 根据 $\angle OAB$ 和 $\angle OBA$ 判断碰撞情况

用数学语言来表示这一情况，为：

$$\frac{\overrightarrow{AO} \cdot \overrightarrow{AB}}{|\overrightarrow{AO}| |\overrightarrow{AB}|} \frac{\overrightarrow{BO} \cdot \overrightarrow{BA}}{|\overrightarrow{BO}| |\overrightarrow{BA}|} < 0 \quad (2-26)$$

由于 \overrightarrow{AO} 、 \overrightarrow{AB} 、 \overrightarrow{BO} 、 \overrightarrow{BA} 一定大于 0，则判断条件可以简化为：

$$\overrightarrow{AO} \cdot \overrightarrow{AB} * \overrightarrow{BO} \cdot \overrightarrow{BA} < 0 \quad (2-27)$$

用线段的端点坐标 (x_1, y_1) 、 (x_2, y_2) ，圆的圆心坐标 (x_0, y_0) 来表示上述的判断条件为：

$$\begin{aligned} & [(x_0 - x_1)(x_2 - x_1) + (y_0 - y_1)(y_2 - y_1)] * \\ & [(x_0 - x_2)(x_1 - x_2) + (y_0 - y_2)(y_1 - y_2)] < 0 \end{aligned} \quad (2-28)$$

若 $\angle OAB$ 与 $\angle OBA$ 均为锐角,则需要利用圆的圆心到线段的距离来判断圆与线段是否发生碰撞。

若圆的圆心到线段的距离大于圆的半径,如图 2-13 (a) 所示,则圆与线段不发生碰撞,若圆的圆心到线段的距离小于圆的半径,如图 2-13 (b) 所示,则圆与线段发生碰撞。

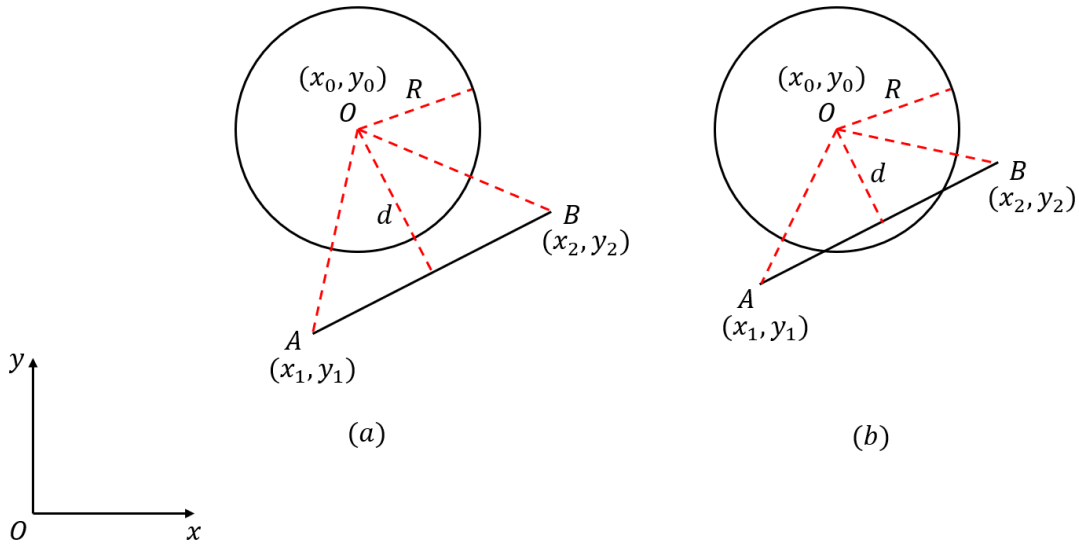


图 2-13 根据圆心到线段距离判断碰撞情况

用线段的端点坐标 (x_1, y_1) 、 (x_2, y_2) , 圆的圆心坐标 (x_0, y_0) 来表示圆心到线段的距离 d 为:

$$d = \frac{|x_0 y_1 + x_1 y_2 + x_2 y_0 - x_2 y_1 - x_1 y_0 - x_0 y_2|}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad (2-29)$$

若满足 $d < R$, 则圆与线段发生碰撞; 若满足 $d \geq R$, 则圆与线段不发生碰撞。

至此, 我们讨论了圆与线段的所有碰撞情况, 并对所有情况给出了对应的碰撞情况与判断条件, 圆与线段不发生碰撞的情况下, 对应的圆弧与线段也一定不发生碰撞, 而圆与线段发生碰撞的情况则需要进一步讨论对应的圆弧与线段的碰撞情况。

圆与线段发生碰撞时, 会存在 1 个或 2 个交点, 若其中有交点位于圆弧所对应的

角度范围上，则圆弧与线段发生碰撞，否则，即便圆与线段发生了碰撞，对应的圆弧与线段也不会发生碰撞。因此接下来我们要求解线段与圆的交点在圆上所对应的角度。

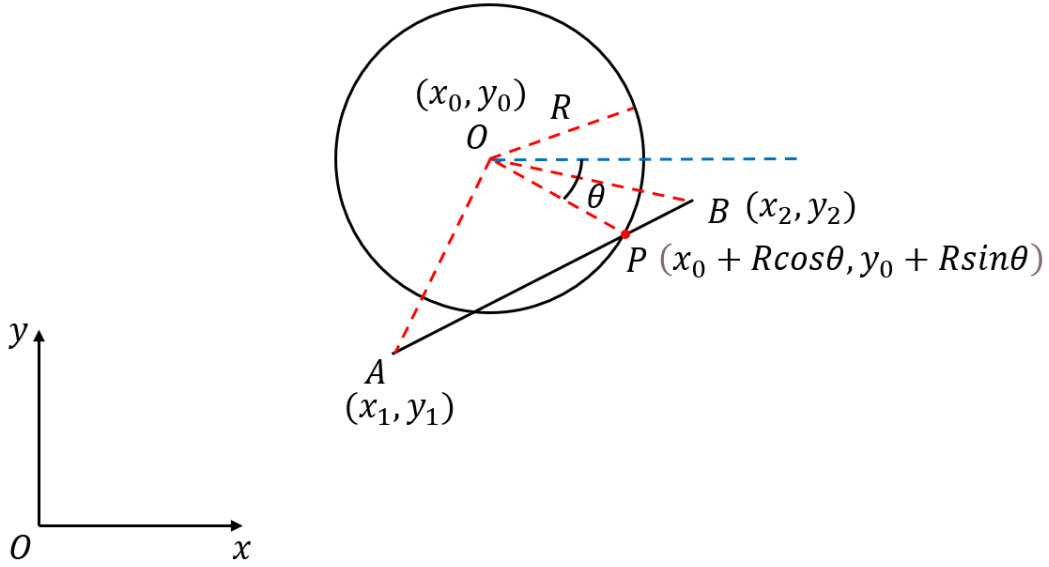


图 2-14 圆与线段碰撞时交点信息

如图 2-14, 设交点在圆上对应角度为 θ , 则交点 P 的坐标可以写为 $(x_0 + R\cos\theta, y_0 + R\sin\theta)$, 代入直线的两点式方程

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (2-30)$$

中, 令 $a = (x_1 - x_2)R$, $b = (y_2 - y_1)R$, $c = (x_2 - x_1)(y_0 - y_1) - (y_2 - y_1)(x_0 - x_1)$, 得到关于 θ 的方程:

$$a\cos\theta + b\sin\theta = c \quad (2-31)$$

由三角函数的性质, 有:

$$\sin\theta = \sqrt{1 - \cos^2\theta} \quad (2-32)$$

将式 (2-32) 代入式 (2-31) 中, 令 $p = a^2 + b^2$, $q = -2ac$, $r = c^2 - b^2$, 得:

$$p\cos^2\theta + q\cos\theta + r = 0 \quad (2-33)$$

若线段的两个端点都在圆外, 如图 2-13 (b), 那么线段与圆有两个交点, 由求根公式计算可得, 交点对应的角度 θ 满足:

$$\cos\theta = \frac{-q \pm \sqrt{q^2 - 4pr}}{2p} \quad (2-34)$$

若线段的两个端点一个在圆外, 一个在圆内, 如图 2-11 (b), 那么线段与圆只有

一个交点，由求根公式计算可得，交点对应的角度 θ 满足：

$$\cos\theta = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad (2-35)$$

最后将解出的角度 θ 与圆弧所对应的角度范围进行比较，若 θ 在圆弧所对应的角度范围内，则线段与圆弧发生碰撞，若 θ 在圆弧所对应的角度范围外，则线段与圆弧不发生碰撞。

2.3.3 圆弧与圆弧的碰撞检测分析

圆弧与圆弧的碰撞检测方法和线段与圆弧的碰撞检测方法相类似，同样需要先将圆弧补全成完整的圆，再判断两圆是否相交，若两圆不相交，则对应的圆弧一定不发生碰撞，若两圆相交，则需要再判断交点是否在圆弧对应的角度上，若交点在圆弧对应的角度上，则两条圆弧发生碰撞，若交点不在圆弧对应的角度上，则两条圆弧不发生碰撞。

首先将两条圆弧补全成两个完整的圆，再判断两圆是否相交。首先定义较大的圆的半径为 R ，较小的圆的半径为 r ，两圆圆心距离为 d ，那么根据两圆半径与两圆圆心距离的关系，两圆的碰撞关系分为以下几种情况：

1. $d \geq R + r$ 或 $d \leq R - r$ 时，两圆不发生碰撞，如图 2-15 所示。

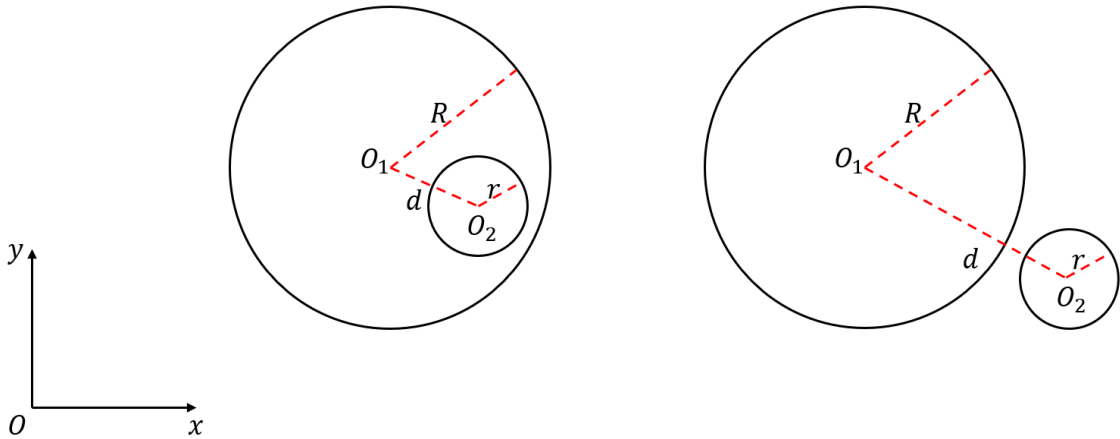


图 2-15 两圆不发生碰撞

2. $R - r < d < R + r$ 时，两圆发生碰撞，如图 2-16 所示。

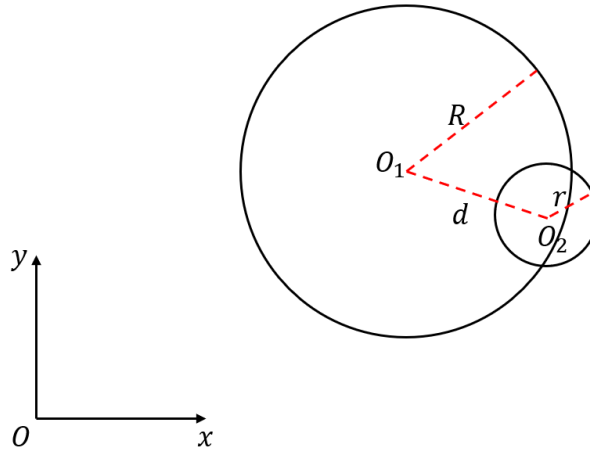


图 2-16 两圆发生碰撞

接下来判断当 $R - r < d < R + r$ 时, 对应的圆弧是否发生碰撞。与线段与圆弧的碰撞检测方法类似, 需要解出交点对应的角度再进行碰撞检测, 不同的是, 两条圆弧的碰撞检测需要再两个圆上各进行一次, 只要有一个圆弧的角度范围中不包含交点的角度, 这两条圆弧就不发生碰撞。

以大圆的判断过程为例, 首先定义交点在大圆上对应角度为 θ , 大圆圆心 O_1 坐标为 (x_1, y_1) , 小圆圆心 O_2 坐标为 (x_2, y_2) , 则交点坐标可以写为 $(x_1 + R\cos\theta, y_1 + R\sin\theta)$, 代入另一个圆的解析式 $(x - x_2)^2 + (y - y_2)^2 = r^2$ 中, 令 $a = 2(x_1 - x_2)R$, $b = 2(y_1 - y_2)R$, $c = r^2 - R^2 - (x_1 - x_2)^2 - (y_1 - y_2)^2$, 得:

$$a\cos\theta + b\sin\theta = c \quad (2-36)$$

将式 (2-32) 代入式 (2-36) 中, 令 $p = a^2 + b^2$, $q = -2ac$, $r = c^2 - b^2$, 得:

$$p\cos^2\theta + q\cos\theta + r = 0 \quad (2-37)$$

因此由求根公式计算可得, 两个交点对应的角度 θ 满足:

$$\cos\theta = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad (2-38)$$

然后将解出的角度 θ 与大圆圆弧所对应的角度范围进行比较, 同理再解出交点在小圆上对应的角度, 再与小圆圆弧所对应的角度范围进行比较, 若两次判断中有任意一次判断出交点不在圆弧对应角度范围内, 则两条圆不发生碰撞, 否则两条圆弧发生碰撞。

有了线段与线段、线段与圆弧、圆弧与圆弧三种类型的碰撞检测分析, 就可以对

车辆路径与停车场障碍物模型的所有线段与圆弧进行碰撞检测,来判断车辆路径是否与障碍物发生碰撞。

2.4 本章小结

本章首先对四轮转向车辆与停车场环境进行了建模,其中四轮转向车辆建模包括四轮转向车辆运动学建模与几何建模两部分。运动学建模用于后续基于四轮转向的路径规划和控制器设计两个部分。轨迹规划时运动学模型用于确定车辆的运动学模型约束,保证规划出的路径满足车辆运动学模型约束,从而确保车辆有能力跟随路径运动。控制器设计时运动学模型用于确定车辆速度与位姿之间的关系,用以设计对应的纵向控制器。同时将四轮转向车辆运动学模型与前轮转向车辆运动学模型进行比较,得出四轮转向车辆具有更小的转弯半径的结论。而车辆几何模型与停车场环境模型将车辆路径与障碍物简化成了由若干线段与圆弧组成的几何体,用于后续的碰撞检测研究。

碰撞检测研究时,首先将所有碰撞可能情况根据几何体种类分为了线段与线段、线段与圆弧、圆弧与圆弧三种情况,并依次对三种情况进行碰撞检测分析,利用简单的几何关系排除一些一定不会发生碰撞的情况,再利用解析法去判断剩余的情况,加快了运算效率,并成功完成了碰撞检测分析。

第三章 基于四轮转向的泊车路径规划算法设计

泊车路径规划算法设计需要得到一条从起点到终点的、满足车辆运动学约束并且曲率连续的路径，并且该路径需要满足起点与终点的位姿要求，且不与环境中障碍物发生碰撞。

在泊车环境中，起点位姿为当前车辆的位姿，终点位置为车辆停进目标停车位时，几何中心所在位置，终点方向角则选取停车位所对应方向，其中垂直停车位选择车头向外，平行停车位选择车头向右，如图 3-1、3-2 所示。障碍物则为其他车辆所占据的网格。

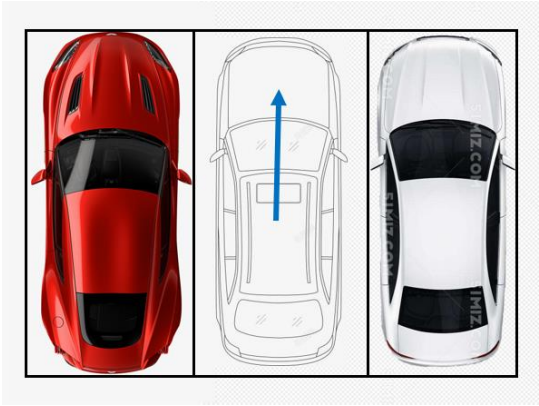


图 3-1 垂直停车位的终点位姿

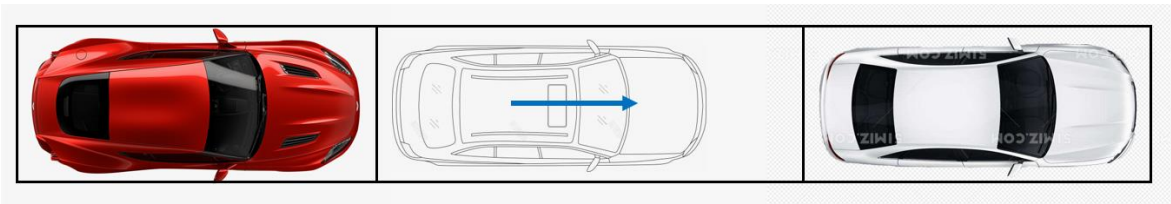


图 3-2 平行停车位的终点位姿

本章将选择 Reeds-Shepp 曲线规划满足四轮转向车辆运动学模型的轨迹，并用 RRT*算法进行避障，使得到的路径可行。这种方法可以完成所有环境下的轨迹规划，但该算法的运算复杂，运算量极大，因此本章最后，针对泊车环境，又设计了一种通过计算解析解来降低倒车部分运算复杂程度的方法，从而使得泊车环境的路径规划算法运算效率提高。

3.1 满足运动学模型约束的 Reeds-Shepp 曲线

首先需要得到一条满足的四轮转向车辆运动学模型约束的轨迹，所谓的满足四轮转向车辆运动学模型约束，就是得到的轨迹曲率不能超过车辆的最大转向曲率，即满足车辆的曲率约束。

常用的轨迹规划方法有 Dubins 曲线与 Reeds-Shepp (RS) 曲线两种方法，二者都是规划一条从起始位姿到终点位姿的满足车辆曲率约束的最短路径，区别是 Dubins 曲线不允许倒车，而 RS 曲线允许进行倒车。而在泊车环境中，倒车入库是必不可少的操作，因此本论文的轨迹规划选择 RS 曲线进行研究。

在 RS 曲线中，车辆只有六种运动方式：以最小转弯半径左转前进、直线前进、以最小转弯半径右转前进，以最小转弯半径左转倒车、直线倒车、以最小转弯半径右转倒车。我们将这六种运动方式记为 L^+ 、 S^+ 、 R^+ 、 L^- 、 S^- 、 R^- ，并将两种直线运动统一记为 S，将四种转弯运动统一记为 C。文献[17]中，Reeds J 等人将所有可能的最优路径分为了以下 9 种，即最终的路径一定属于这九种路径类型中的一种： $C|C|C$ 、 $C|CC$ 、 $CC|C$ 、 $CC_\beta|C_\beta C$ 、 $C|C_\beta C_\beta|C$ 、 CSC 、 $C|C_{\pi/2}SC$ 、 $CSC_{\pi/2}|C$ 、 $C|C_{\pi/2}SC_{\pi/2}|C$ 。其中，|表示前后两次运动方向相反，连续的 C 表示前后两次运动转向角相反， $C_{\pi/2}$ 表示对应的转弯运动转过的角度为 $\pi/2$ ， $C_\beta C_\beta$ 表示前后两次转弯运动转过的角度相等。

在进行计算前，我们需要将要处理的轨迹规划问题的位姿进行统一，即将起点位置移动到坐标系原点，并将起点方向角旋转至 0，使得起点位姿变为 $(0,0,0)^T$ 。设初始状态下起点位姿为 $(x_1, y_1, \theta_1)^T$ ，终点位姿为 $(x_2, y_2, \theta_2)^T$ ，车辆最小转弯半径为 ρ 并化归为 1，则位姿统一后，起点位姿变为 $(0,0,0)^T$ ，终点位姿变为 $((x_2 - x_1)\cos\theta_1 + (y_2 - y_1)\sin\theta_1)/\rho, (-(x_2 - x_1)\sin\theta_1 + (y_2 - y_1)\cos\theta_1)/\rho, \theta_2 - \theta_1)^T$ ，记终点位姿为 $(x, y, \theta)^T$ 。

除此之外文献[17]中还提到了运用三种简单的变换可以将同一类型的不同路径组合变换成同一种路径组合，从而降低计算路径长度的工作量。

时间翻转 (timeflip)，就是将路径中的运动方向取成其反方向，即 C^+ 与 C^- 相互变换， S^+ 与 S^- 相互变换，并将终点位姿从 $(x, y, \theta)^T$ 变为 $(-x, y, -\theta)^T$ 。例如，想要计算从起始位姿 $(0,0,0)^T$ 到终点位姿 $(x, y, \theta)^T$ 的 $L^-S^+R^+$ 路径长度，可以计算从起始位姿 $(0,0,0)^T$ 到变换后的终点位姿 $(-x, y, -\theta)^T$ 的 $L^+S^-R^-$ 路径长度。

反射 (reflect)，就是将路径中的转弯方向取成其反方向，即 L 与 R 相互变换，并将终点位姿从 $(x, y, \theta)^T$ 变为 $(x, -y, -\theta)^T$ 。例如，想要计算从起始位姿 $(0,0,0)^T$ 到终点

位姿 $(x, y, \theta)^T$ 的 $L^-S^+R^+$ 路径长度, 可以计算从起始位姿 $(0, 0, 0)^T$ 到变换后的终点位姿 $(x, -y, -\theta)^T$ 的 $R^-S^+L^+$ 路径长度。

向后变换 (backwards), 即将原本的路径组合以相反的顺序排列, 并将终点位姿从 $(x, y, \theta)^T$ 变为 $(x\cos\theta + y\sin\theta, x\sin\theta - y\cos\theta, \theta)^T$ 。例如想要计算从起始位姿 $(0, 0, 0)^T$ 到终点位姿 $(x, y, \theta)^T$ 的 $L^-S^+R^+$ 路径长度, 可以计算从起始位姿 $(0, 0, 0)^T$ 到变换后的终点位姿 $(x\cos\theta + y\sin\theta, x\sin\theta - y\cos\theta, \theta)^T$ 的 $R^+S^+L^-$ 路径长度。

这样一来, 许多路径组合可以通过改变终点位姿来化归为同一种路径组合, 减少了计算路径长度时的计算量。

此后就可以通过计算从起始位姿 $(0, 0, 0)^T$ 到终点位姿 $(x, y, \theta)^T$ 的所有路径长度并进行比较来得到最优路径了, Reeds J 等人在文献[17]中给出了各种情况的详细计算方法, 而在泊车环境下, 为了使计算方便, 我们只选取了包含三段路径所有路径组合, 即 CCC 与 CSC。

首先声明两种运算: $(r, \theta) = R(x, y)$ 表示将直角坐标 (x, y) 转化为极坐标 (r, θ) , $M(x)$ 表示将 x 限制在区间 $[-\pi, \pi]$ 内。

所有的 CCC 类型的路径都可以通过三种变换方式变成两种路径组合 $L^+R^-L^+$ 与 $L^+R^-L^-$ 中的一种, 两种路径组合的计算方式相同, 计算公式为:

$$\begin{aligned}
 \xi &= x - \sin\theta \\
 \eta &= y - 1 + \cos\theta \\
 (u_1, \theta') &= R(\xi, \eta) \\
 u_1 > 4 &\rightarrow L = \infty \\
 u_1 \leq 4 &\rightarrow \begin{cases} u = -2 \arcsin\left(\frac{u_1}{4}\right) \\ t = M\left(\theta' + \frac{u}{2} + \pi\right) \\ v = M(\theta - t + u) \end{cases} \\
 L &= |t| + |u| + |v|
 \end{aligned} \tag{3-1}$$

其中 $L = \infty$ 表示该路径组合无解, t 、 u 、 v 为三段路径的长度。

而所有的 CSC 类型的路径都可以通过三种变换方式变为两种路径组合 $L^+S^+L^+$ 与 $L^+S^+R^+$ 中的一种。路径组合 $L^+S^+L^+$ 的计算公式为:

$$\begin{aligned}
 \xi &= x - \sin\theta \\
 \eta &= y - 1 + \cos\theta \\
 (u, t) &= R(\xi, \eta) \\
 v &= M(\theta - t) \\
 L &= |t| + |u| + |v|
 \end{aligned} \tag{3-2}$$

路径组合 $L^+S^+R^+$ 的计算公式为：

$$\begin{aligned}
 \xi &= x + \sin\theta \\
 \eta &= y - 1 - \cos\theta \\
 (u_1, t_1) &= R(\xi, \eta) \\
 u_1^2 < 4 &\rightarrow L = \infty \\
 u_1^2 \geq 4 &\rightarrow \begin{cases} u = \sqrt{u_1^2 - 4} \\ (T, \theta') = R(u, 2) \\ t = M(t_1 + \theta') \\ v = M(t - \theta) \end{cases} \\
 L &= |t| + |u| + |v|
 \end{aligned} \tag{3-3}$$

由此便可以得到所有满足 CCC 与 CSC 的路径的长度，再通过比较找出长度最短的路径，即为满足四轮转向车辆运动学模型约束的最优路径。

3.2 基于随机采样与 Reeds-Shepp 曲线的 RRT*算法

在得到满足四轮转向车辆运动学模型约束的轨迹后，我们还需要把它应用在避障路径规划算法中，目前常用的较为成熟的路径规划算法有基于网络搜索的算法与基于随机采样的算法两种，其中基于网络搜索的算法以 A*算法为主，基于随机采样的算法以 RRT*算法为主，由于后续实验所用地图为点云地图，而基于网络搜索的算法需要将地图转化为栅格地图，因此本章采用 RRT*算法来进行路径规划。

RRT*算法的原理是创建一棵以起点为根节点的搜索树，通过在地图中随机采样来扩展随机数，最终扩展到终点，完成路径规划。

首先创建一棵搜索树 T，T 以起始点 x_{start} 为根节点，然后在地图上随机采样一个点 x_{rand} ，找出 T 中离 x_{rand} 最近的点 x_{near} 。如果 x_{rand} 与 x_{near} 之间的距离小于步长 step，则将 x_{rand} 与 x_{near} 相连，判断是否与环境中的障碍物发生碰撞，若不发生碰撞，则将 x_{near} 作为新节点 x_{new} 加入 T 中，若发生碰撞，则重新采样。如果 x_{rand} 与 x_{near} 之间的距离大于步长 step，则将 x_{near} 向 x_{rand} 方向延伸 step 距离到达 x_{new} ，若 x_{new} 与 x_{near} 不发生碰撞，则将 x_{new} 加入 T 中，若发生碰撞，则重新采样。

在完成这次采样后，以 x_{new} 为圆心，事先规定好的 r 为半径，从而得到一个圆形区域，并找到该圆形区域内的所有节点 x_{search} ，进行以下判断：当 x_{new} 以 x_{search} 为父节点时，代价是否减小，若减小，则将 x_{new} 的父节点替换为 x_{search} 。对所有 x_{search} 均执行完这个判断后，继续执行下一步判断：当 x_{search} 以 x_{new} 为父节点时，代价是否减

小，若减小，则将 x_{search} 的父节点替换为 x_{new} 。对所有 x_{search} 均执行完这个判断后，本次 T 扩展完成，开始执行下一次采样，直到 x_{new} 到终点 x_{goal} 的距离小于 step。

注意到 RRT*算法中需要判断新路径与障碍物是否发生碰撞，由于 RRT*的扩展只有线段，因此每一步只需要对新路径与障碍物边界进行一次线段与线段的碰撞检测分析。RRT*算法的伪代码如下表 3-1 所示。

表 3-1 RRT*算法伪代码

Algorithm 1 RRT*

Input: $\mathcal{M}, x_{start}, x_{end}, n$

Output: A path Γ from x_{start} to x_{end}

```

1:  $\mathcal{T}.init(x_{start})$ 
2: for  $i = 1 \rightarrow n$  do
3:    $x_{rand} \leftarrow Sample(\mathcal{M})$ ;
4:    $x_{near} \leftarrow NearestNeighbor(x_{rand}, \mathcal{T})$ ;
5:    $x_{new} \leftarrow Steer(x_{rand}, x_{near})$ ;
6:   if  $CollisionFree(\mathcal{M}, x_{new}, x_{near})$  then
7:      $\mathcal{T}.addNode(x_{new})$ ;
8:      $NearNodes \leftarrow NearSearch(\mathcal{T}, x_{new}, r)$ ;
9:      $x_{search} \leftarrow ChooseParent(\mathcal{T}, x_{new}, NearNodes)$ ;
10:     $\mathcal{T}.addEdge(x_{search}, x_{new})$ ;
11:     $x_{search} \leftarrow ChooseChild(\mathcal{T}, x_{new}, NearNodes)$ ;
12:     $\mathcal{T}.addEdge(x_{new}, x_{search})$ ;
13:     $\mathcal{T}.Rewire(\mathcal{T}, x_{new}, NearNodes)$ ;
14:   end if
15: end for
16: return  $\mathcal{T}$ 

```

但 RRT*本身不满足四轮转向车辆的运动学模型约束，因此需要与 RS 曲线相结合，将节点的连接方式从直接相连改为用 RS 曲线相连成一条满足四轮转向车辆运动学模型约束的路径，并把 x_{rand} 与 x_{near} 之间的距离改为 RS 曲线距离，当 RS 曲线距离小于 step 时，将 x_{near} 作为新节点 x_{new} ，并将 x_{rand} 与 x_{near} 之间的 RS 曲线加入 T 中，当 RS 曲线距离大于 step 时，令 x_{near} 沿着 RS 曲线前进 step 距离，将该点作为 x_{new} 和刚刚走过的 RS 曲线加入 T 中。

此时 RRT*-RS 算法中的路径变为线段与圆弧的组合，因此在判断新路径与障碍物是否发生碰撞时，需要对新路径与障碍物边界进行线段与线段的碰撞检测分析和线

段与圆弧的碰撞检测分析。RRT*-RS 算法的伪代码如表 3-2 所示

表 3-2 RRT*-RS 算法伪代码

Algorithm 2 RRT*-RS

Input: $\mathcal{M}, x_{start}, x_{end}, n$

Output: A path Γ from x_{start} to x_{end}

```

1:  $\mathcal{T}.init(x_{start})$ 
2: for  $i = 1 \rightarrow n$  do
3:    $x_{rand} \leftarrow Sample(\mathcal{M});$ 
4:    $x_{near} \leftarrow NearestNeighbor(x_{rand}, \mathcal{T});$ 
5:    $x_{new} \leftarrow ReedsSheppSteer(x_{rand}, x_{near});$ 
6:   if  $CollisionFree(\mathcal{M}, x_{new}, x_{near})$  then
7:      $\mathcal{T}.addNode(x_{new});$ 
8:      $NearNodes \leftarrow NearSearch(\mathcal{T}, x_{new}, r);$ 
9:      $x_{search} \leftarrow ChooseParent(\mathcal{T}, x_{new}, NearNodes);$ 
10:     $\mathcal{T}.addEdge(x_{search}, x_{new});$ 
11:     $x_{search} \leftarrow ChooseChild(\mathcal{T}, x_{new}, NearNodes);$ 
12:     $\mathcal{T}.addEdge(x_{new}, x_{search});$ 
13:     $\mathcal{T}.Rewire(\mathcal{T}, x_{new}, NearNodes);$ 
14:   end if
15: end for
16: return  $\mathcal{T}$ 

```

这样一来，RRT*-RS 算法便可以得到一条既满足四轮转向车辆运动学模型约束，又可以实现避障的路径，并且适用于几乎所有的环境。

3.3 基于解析法与 RRT*-RS 算法的快速规划方法

RRT*-RS 适用于几乎所有环境，但代价是算法运算量较大，RRT*本身就需要进行多次的采样，而 RS 曲线的加入，导致每一次采样还需要额外运算 RS 曲线所有可能路径的长度并选出最短路径，并且碰撞检测时还需要对每段路径单独做一次碰撞检测，导致运算量大大增加。而泊车环境相对简单，且泊车过程的后半段是倒车入库的部分，需要考虑的因素较少，几乎只受停车位与车辆本身信息的影响。因此，本论文将泊车过程分为了前进与倒车两个部分，并针对到处部分提出了一种基于解析法的快速规划方法，使得整个路径规划算法的效率大大提升。

这种方法即通过解析法提前规划好后半段倒车入库的部分，得益于泊车问题的特性，这一段路径规划并不复杂，路径只与车辆信息、停车位信息有关。利用解析法规划好这一部分路径后，就可以得到一个倒车过程开始的点，定义这个点为倒车点，则前半段的路径规划就变为从起始位姿到倒车点位姿的路径规划问题，并且此部分没有倒车环节，因此可以用 Dubins 曲线代替 RS 曲线，或者直接将 RS 曲线中包括倒车环节的路径组合排除，得到一个简化后的 RRT*-RS 算法。这种方法可以大大减少路径规划运算量，提高路径规划效率。

而这一方法的关键点就在于倒车点坐标的计算，该点仅取决于车辆与停车位的信息，并且可以求出解析解。倒车点的计算可以反向理解为将车辆从停车位中开出来，从车辆开始移动到车辆完全离开停车位的过程看作车辆停车时的完倒车部分，为了便于前进部分有可行路径以及运算的便利性，我们将车辆方向平行于停车位边缘时的位置选取为倒车点，如图 3-3、3-4 中 P 点所示所示。

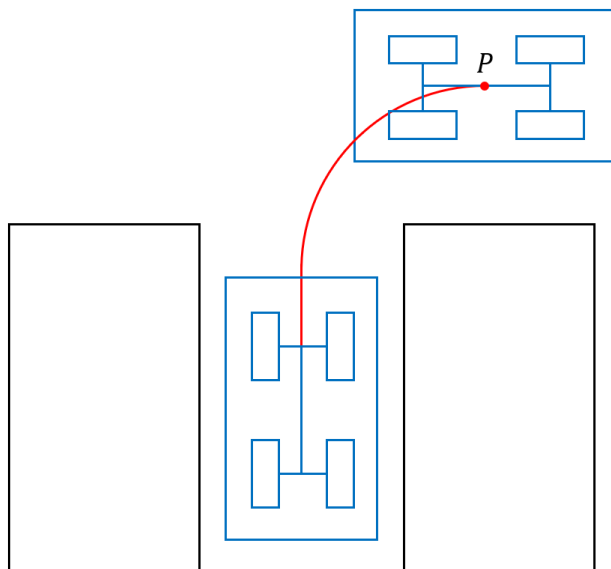


图 3-3 垂直停车位的倒车点

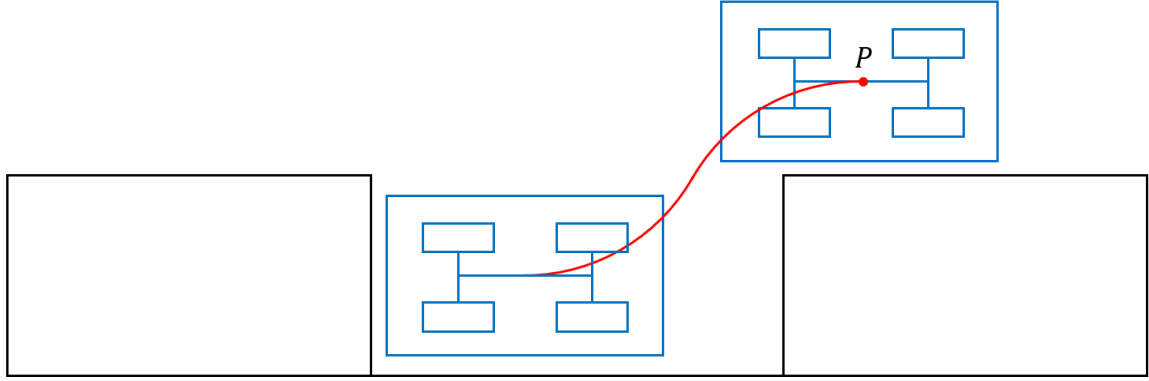


图 3-4 水平停车位的倒车点

本章将会分为垂直停车位和平行停车位两种情况进行讨论。

3.3.1 垂直停车位倒车点的计算

首先要选取停车点，即最终车辆所停位置的几何中心坐标，记为 (x, y) ，横坐标选择停车位的中心点，纵坐标则越深越好，极限情况则是令车辆最终完全紧贴停车位末端，但需要保留一定的误差距离，防止后续控制器设计时存在控制精度上的误差发生碰撞，这里误差距离选取车辆与停车位长度差值的四分之一，记为 ε 。

找出停车点后需要分情况进行讨论，并且讨论的方向为车辆从车库向外开。假设车辆从左向右前进，则倒车点方向为水平向右。记车辆长度为 l ，宽度为 w ，停车位长度为 L ，宽度为 W ，车辆最小转弯半径为 r 。

第一种情况下，车辆可以直接以最小转弯半径右转前进从而离开车库，如图 3-5 所示。

此时车辆信息与停车位信息之间满足：

$$L - \frac{l}{2} - \varepsilon < \sqrt{-wr + \frac{w^2}{4} + Wr - \frac{W^2}{4}} \quad (3-4)$$

此时不难看出倒车点坐标为：

$$(x + r, y + r) \quad (3-5)$$

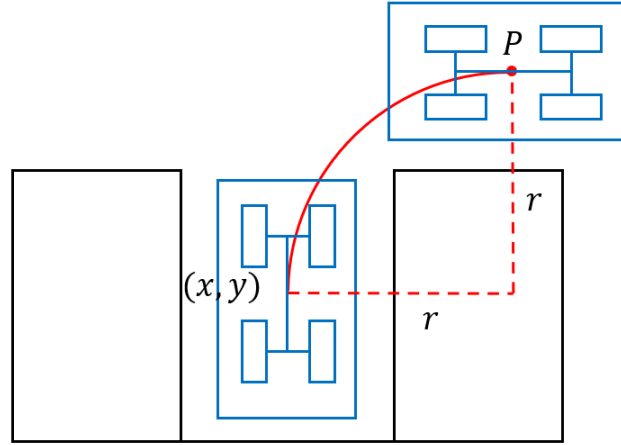


图 3-5 垂直停车位倒车点计算第一种情况示意图

第二种情况下，车辆需要前进一定距离才能以最小转弯半径向右转离开车库，但最小转弯半径小于停车位宽度的一半，如图 3-6 所示。

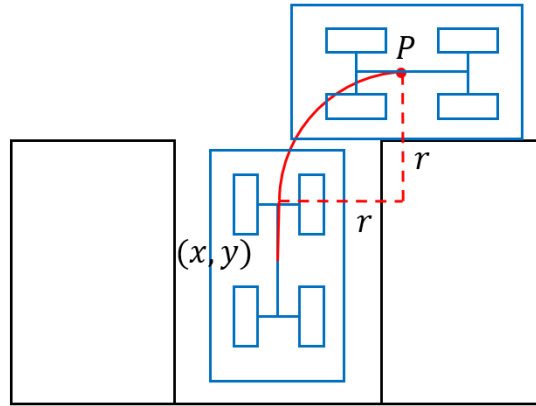


图 3-6 垂直停车位倒车点计算第二种情况示意图

此时车辆信息与停车位信息之间满足：

$$r < \frac{W}{2} \quad (3-6)$$

此时倒车点坐标为：

$$\left(x + r, y + L - \frac{l}{2} - \varepsilon + \frac{W}{2} - r \right) \quad (3-7)$$

第三种情况下,车辆的最小转弯半径大于停车位宽度的一半,此时存在一个临界条件,即车辆右侧边缘会与停车位右上角恰好接触,如图 3-7 所示,其中深蓝色对应的时刻为临界时刻。

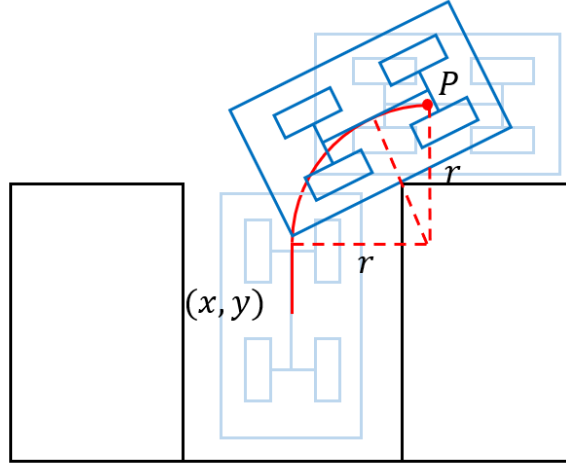


图 3-7 垂直停车位倒车点计算第三种情况示意图

此时通过计算可以得出倒车点坐标为:

$$\left(x + r, y + r + L - \frac{l}{2} - \varepsilon - \sqrt{-wr + \frac{w^2}{4} + Wr - \frac{W^2}{4}} \right) \quad (3-8)$$

这三种情况即可包含垂直停车位的所有情况,在得到倒车点坐标时,倒车部分的路径就已经得到了。第一种情况的倒车路径就是从倒车点开始以最小转弯半径右转倒车,直到车辆与垂直停车位平行,而后两种情况的倒车路径则是从倒车点开始以最小转弯半径右转倒车到车辆与垂直停车位平行后,继续沿直线倒车直到车辆到达停车点。而前进部分的路径则使用简化后的 RRT*-RS 算法得到可行路径。

将垂直停车位的路径规划算法用 Matlab 进行仿真,并对比四轮转向车辆与前轮转向车辆的路径规划效果,期间保证两个模型的车辆长度、宽度、轴距相同。仿真结果如图 3-8 所示,其中蓝色路径为四轮转向车辆规划出的垂直停车位泊车路径,粉色路径为前轮转向车辆规划出的垂直停车位泊车路径,从仿真结果不难看出,四轮转向车辆得益于更小的转弯半径,相较于前轮转向小车具有更短的泊车路径。

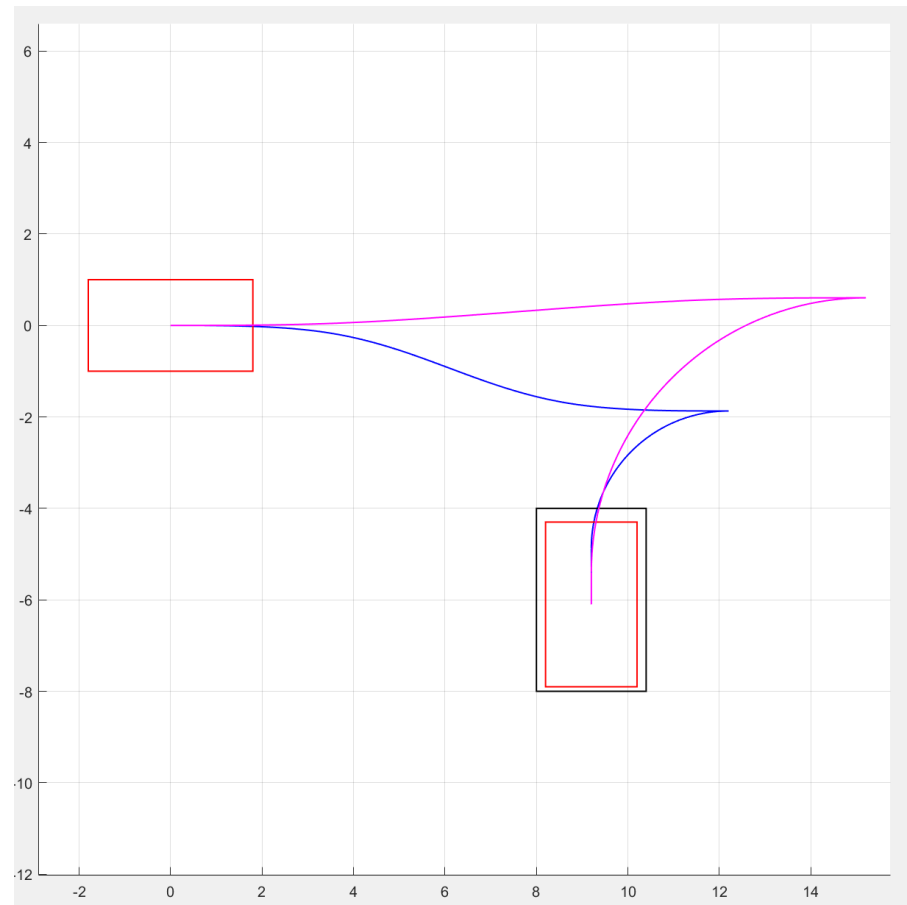


图 3-8 垂直停车位四轮转向车辆与前轮转向车辆仿真实验

同时，通过改变停车位的位置、大小以及车辆大小，让四轮转向车辆和前轮转向车辆进行多组对照实验，对比规划出的路径长度，得到的实验结果如表 3-3 所示。

表 3-3 垂直停车位四轮转向车辆与前轮转向车辆路径长度对比结果

实验轮次	四轮转向车辆路径长度（m）	前轮转向车辆路径长度（m）
1	15.6123	22.3622
2	16.5054	23.3586
3	17.4145	23.9371
4	17.4382	24.3558
5	17.5196	24.6972
6	18.3925	25.3537

由对比结果可知，四轮转向车辆得相较于前轮转向小车具有更短的泊车路径，可以更好地应对复杂拥挤的泊车环境。

3.3.2 平行停车位倒车点的计算

平行停车位的倒车过程与垂直停车位不同，当停车位过短时，车辆想要停进停车位需要进行多次变向与倒车，理论上，只要通过有限次前进-倒车循环，车辆就一定可以停进停车位中，因此本章仅讨论在停车环境较小的情况下，车辆只经过一次倒车就完成泊车的路径规划方法。

平行停车位的倒车点坐标计算与垂直停车位类似，需要先选取停车点，纵坐标选择停车位的中心点，为了可以实现只经过一次倒车就完成泊车，横坐标需要选择的越靠后越好，极限情况则是令车辆最终完全紧贴停车位后侧，同样，为了避免控制器精度上存在误差导致车辆发生碰撞，需要保留一定的误差距离，这里同样选取小车长与停车位长度差值的四分之一，记为 ε 。

类似于垂直停车位的求解过程，平行停车位也需要找出停车点后分情况进行讨论，并且讨论的方向同样为车辆从车库向外开。假设车辆从左向右前进，则倒车点方向为水平向右。记车辆长度为 l ，宽度为 w ，停车位长度为 L ，宽度为 W ，车辆最小转弯半径为 r 。

第一种情况下，车辆可以通过两次弧度相同，方向相反的转弯就完全离开停车位，并且整个过程中车辆不会与停车位边界出现接触，如图 3-9 所示。

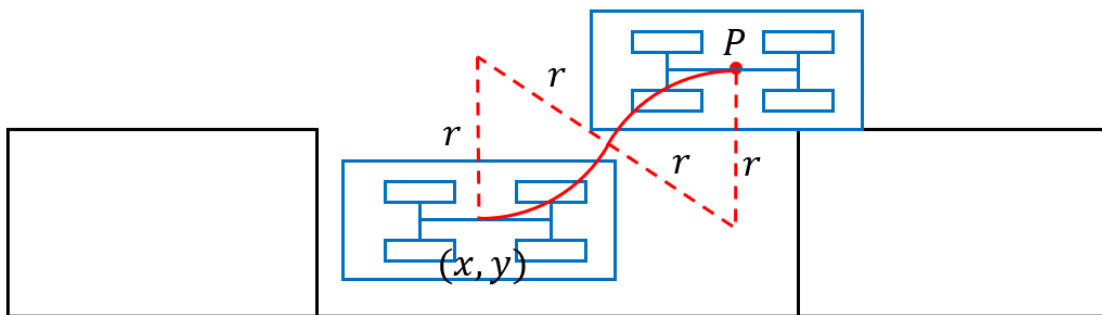


图 3-9 平行停车位倒车点计算第一种情况示意图

此时车辆信息与停车位信息之间满足：

$$\frac{l}{2} + \varepsilon + \sqrt{2r(W+w) - \frac{(W+w)^2}{4}} < L \quad (3-9)$$

此时倒车点坐标为：

$$\left(x + \sqrt{2r(W+w) - \frac{(W+w)^2}{4}}, y + \frac{W}{2} + \frac{w}{2} \right) \quad (3-10)$$

第二种情况下，车辆即便以最小转弯半径左转前进也会与停车位边缘发生碰撞，那么这种情况下，车辆无法通过一次倒车完成泊车。此时存在一个临界情况，即车辆以最小转弯半径左转前进时，车辆的右边缘恰好与停车位右上角重合，如图 3-10 所示，其中深蓝色对应的时刻为临界时刻。

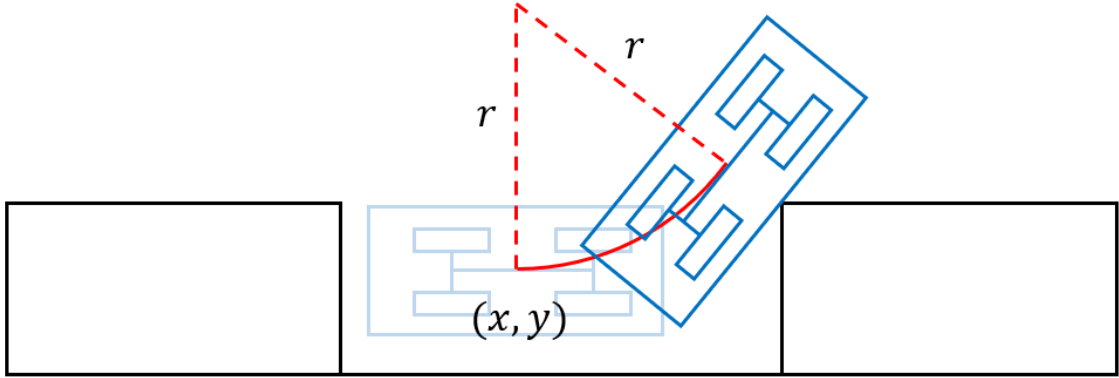


图 3-10 平行停车位倒车点计算第二种情况临界示意图

根据临界条件下，车辆信息与停车位信息之间的关系，可以得出第二种情况下，车辆信息与停车位信息之间满足：

$$\left(L - \frac{l}{2} - \frac{\varepsilon}{2} \right)^2 + \left(\frac{W}{2} - r \right)^2 - \left(\frac{W}{2} + r \right)^2 < 0 \quad (3-11)$$

此时不存在倒车点。

第三种情况下，车辆可以通过两次弧度相同，方向相反的转弯完全离开停车位，但在第二段转弯时，车辆的右边缘恰好与停车位右上角重合，如图 3-11 所示，其中深蓝色对应的时刻为临界时刻。

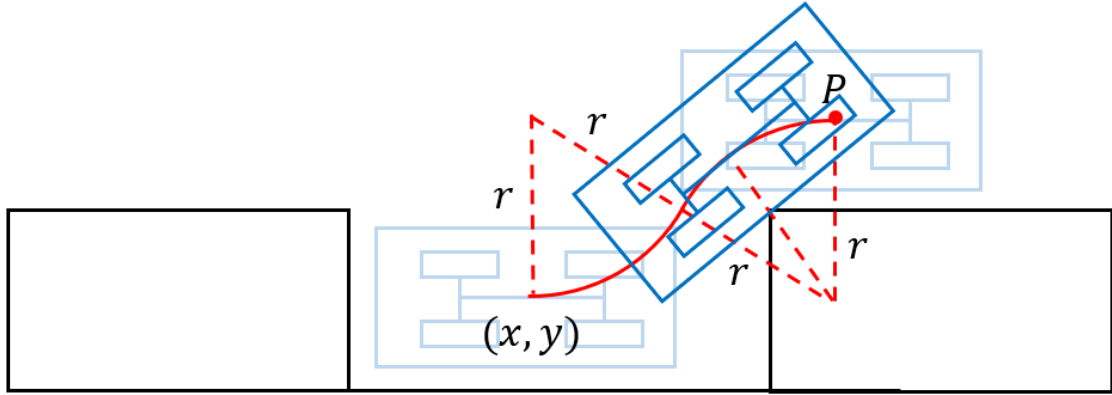


图 3-11 平行停车位倒车点计算第一种情况示意图

此时倒车点坐标为:

$$(x_0, y_0 + r) \quad (3-12)$$

其中, y_0 是方程

$$\left(\frac{4r^2 - \left(r - \frac{w}{2}\right)^2 - (y + W)^2 - \left(x + L - \frac{l}{2} - \varepsilon\right)^2}{2\left(L - \frac{l}{2} - \varepsilon\right)} - \frac{(r + W)y_0}{L - \frac{l}{2} - \varepsilon} - x \right)^2 + (y_0 - y - r)^2 = 4r^2 \quad (3-13)$$

的解中较小的一个, 而 x_0 满足表达式

$$x_0 = \frac{4r^2 - \left(r - \frac{w}{2}\right)^2 - (y + W)^2 - \left(x + L - \frac{l}{2} - \varepsilon\right)^2}{2\left(L - \frac{l}{2} - \varepsilon\right)} - \frac{(r + W)y_0}{L - \frac{l}{2} - \varepsilon} \quad (3-14)$$

方程 (3-13) 是关于 y_0 的一元二次方程, 存在解析解。

这三种情况即可包含平行停车位的所有情况, 其中第二种情况不存在倒车点。在得到倒车点后, 可以用简化后的 RRT*-RS 算法得到可行的前进路径, 而倒车路径则是每种情况下所对应的两段圆弧。

同样将水平停车位的路径规划算法用 Matlab 进行仿真, 并对比四轮转向车辆与前轮转向车辆的路径规划效果, 期间保证两个模型的车辆长度、宽度、轴距相同。仿真结果如图 3-12 所示, 其中蓝色路径为四轮转向车辆规划出的垂直停车位泊车路径, 粉色路径为前轮转向车辆规划出的垂直停车位泊车路径。

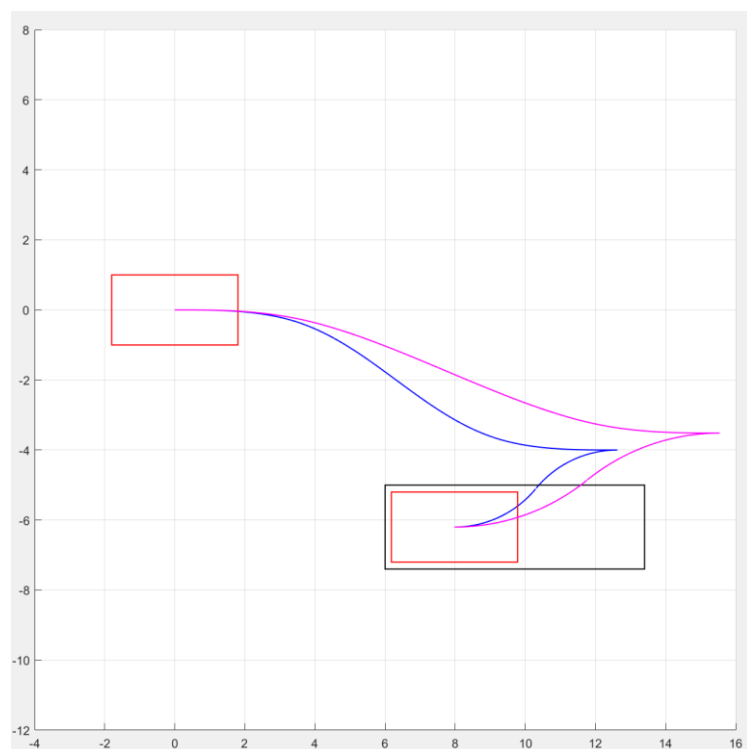


图 3-12 水平停车位四轮转向车辆与前轮转向车辆仿真实验

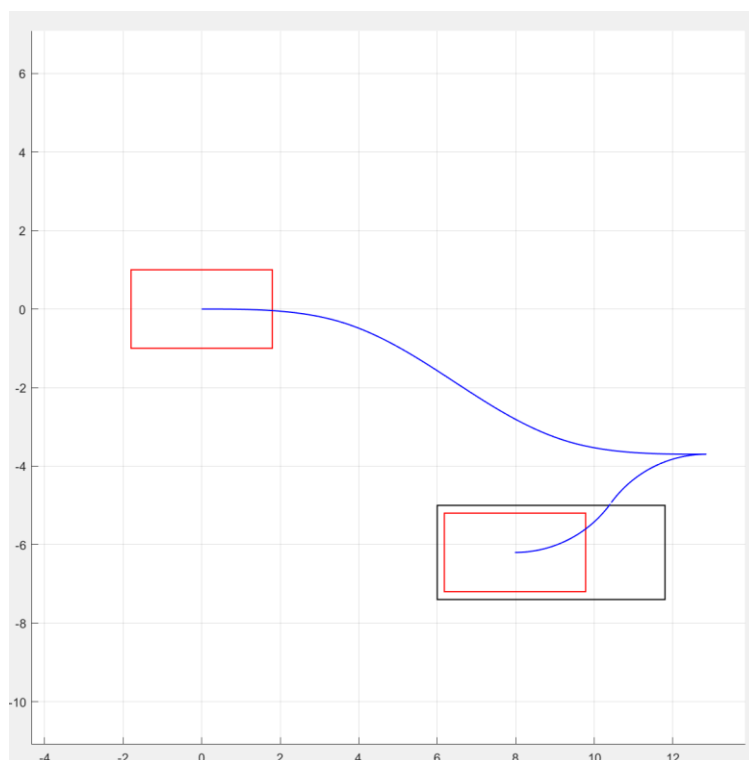


图 3-13 极限情况下水平停车位四轮转向车辆仿真实验

从仿真结果可以看出，四轮转向车辆得益于更小的转弯半径，相较于前轮转向小车具有更短的泊车路径。而在更极限的泊车环境中，停车位长度更小，此时四轮转向车辆仍然可以只进行一次倒车就完成泊车，而前轮转向小车则需要两次以上倒车才能完成泊车，如图 3-13 所示。

同时，通过改变停车位的位置、大小以及车辆大小，让四轮转向车辆和前轮转向车辆进行多组对照实验，对比规划出的路径长度，得到的实验结果如表 3-4 所示，其中部分实验情况下，前轮转向车辆无法进行一次倒车完成泊车。

表 3-4 平行停车位四轮转向车辆与前轮转向车辆路径长度对比结果

实验轮次	四轮转向车辆路径长度 (m)	前轮转向车辆路径长度 (m)
1	17.6628	22.1812
2	17.6628	23.3084
3	18.0456	23.3349
4	18.9017	24.2753
5	18.0456	-
6	18.0480	-

由对比结果可知，四轮转向车辆得相较于前轮转向小车具有更短的泊车路径，可以更好地应对复杂拥挤的泊车环境。

3.4 本章小结

本章完成了基于四轮转向的泊车路径规划算法设计。首先将泊车路径规划分为了轨迹规划和路径规划两个部分。轨迹规划需要得到一条满足四轮转向车辆运动学模型约束的轨迹，由于泊车环境下存在倒车情况，本章选择了允许倒车的 RS 曲线进行轨迹规划，并选择其中只有三段路径的几种情况进行计算。路径规划则需要得到一条满足避障的可行路径，由于后续实验中采用的是点云地图，因此本章选择了较为成熟的 RRT*算法得到可行路径，并用 RS 曲线替代了其中节点的连接方式与距离计算。

由于 RRT*-RS 算法计算复杂程度过高，针对泊车问题，本章提出了一种基于解析法和 RRT*-RS 算法的快速规划方法，这种方法将泊车过程分为了前进过程和倒车

过程两个部分，倒车过程通过解析法求出了倒车点的解析解，从而将倒车过程的轨迹规划复杂度降到最低，而前进部分则可以忽略掉 RS 曲线中倒车的部分，相较于原先的 RRT*-RS 算法效率也有所提升。

最后通过 Matlab 进行了泊车路径规划算法的仿真，成功得到满足车辆运动学约束的可行路径，并将四轮转向车辆和前轮转向车辆进行了对比，发现四轮转向车辆得益于更小的转弯半径，相较于前轮转向小车具有更短的泊车路径，并且在部分极限平行停车位泊车环境下，四轮转向车辆可以通过一次倒车就完成泊车，而前轮转向小车需要两次以上倒车才能完成泊车。由此可见四轮转向车辆在处理更加复杂和拥挤的泊车环境时，表现出了更强的适应性。

第四章 基于四轮转向的运动跟踪控制器设计

基于四轮转向的运动跟踪控制器设计需要设计出一组横向与纵向控制器，使得车辆可以跟随规划好的路径进行运动，而泊车环境是一种低速环境，因此纵向控制器与横向控制器可以解耦，从而对两者分别进行设计。

纵向控制器需要通过控制车辆的加速度来控制车辆在路径纵向上的位置，使得车辆可以在正确的位置进行加减速。横向控制器则要通过控制车辆上的一个点，使这个点可以沿着规划好的路径运动，从而实现车辆的跟随。

本章主要设计适用于泊车环境下的纵向控制器和横向控制器，并利用 CarSim 与 Simulink 联合仿真来验证控制器的可行性及其效果。

4.1 纵向控制器设计

纵向控制器需要通过控制车辆的加速度来控制车辆在路径纵向上的位置，使得车辆可以在正确的位置进行加减速。PID 控制是工业上较为成熟的一种控制器，适用于泊车环境的低速状态，并且对速度的控制精度要求较低，因此本章采用串级 PID 控制来设计纵向控制器。

PID 控制器包括比例环节（proportion, P）、积分环节（integration, I）和微分环节（differentiation, D）三个部分并联而成，其中 P 环节负责控制系统的跟随速度，系数越大跟随速度快，I 环节负责消除稳态误差，D 环节负责减少 I 环节引起的振荡现象。

连续 PID 控制算法的输出表达式为：

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (4-1)$$

其中 $u(t)$ 为控制器输出信号， $e(t)$ 为控制器输入信号， K_p 为比例系数， T_i 为积分时间常数， T_d 为微分时间常数，通过调节 K_p 、 T_i 与 T_d 就可以得到较为理想的控制效果。

而部署在车辆上的控制器是数字信号，需要对模拟信号离散化，其中 $u(t) \approx u(k)$ ， $e(t) \approx e(k)$ ，将积分环节改为后向矩形积分，即：

$$\int_0^t e(t)dt \approx T \sum_{i=1}^k e(i) \quad (4-2)$$

将微分环节改为后向差分，即：

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T} \quad (4-3)$$

此时，连续 PID 控制算法变为离散的位置式 PID 控制算法，输出表达式为：

$$u(k) = K_p e(k) + K_i \sum_{j=1}^k e(j) + K_d [e(k) - e(k-1)] \quad (4-4)$$

其中 $K_i = K_p T / T_i$ ， $K_d = K_p T_d / T$ 。此时只需要调整参数 K_p 、 K_i 、 K_d 便可以对被控对象进行控制。

而串级 PID 控制中有两个 PID 控制器，其中一个 PID 控制器的输出信号是另一个 PID 控制器的输入信号。在泊车问题的纵向控制器设计中，两个 PID 控制器分别为位置 PID 控制器和速度 PID 控制器，并且位置 PID 控制器的输出信号是速度 PID 控制器的输入信号，控制系统如图 4-1 所示。

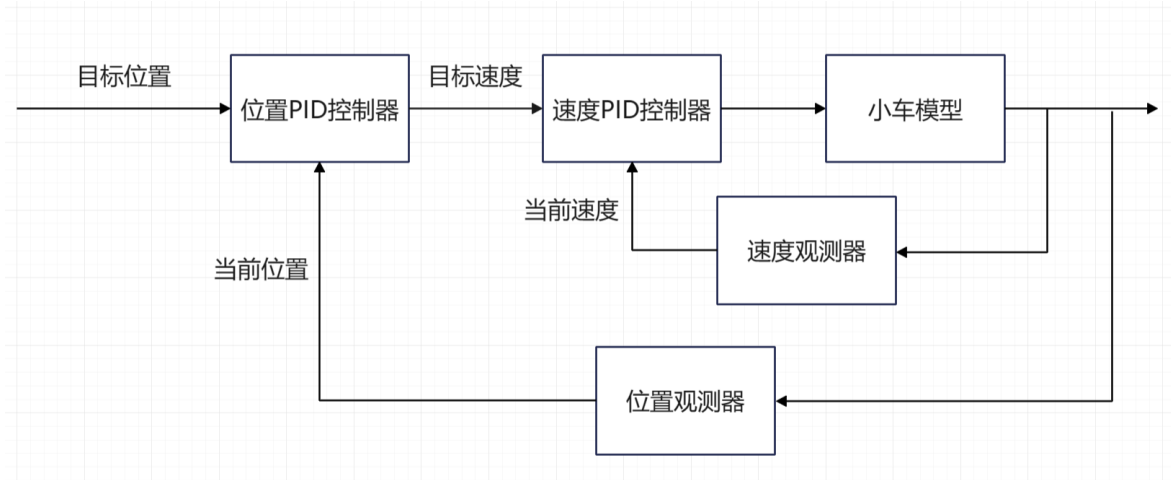


图 4-1 纵向控制器控制系统

其中目标位置为路径的终点坐标，当前位置为车辆位置观测器输出的坐标，二者的差值为目前位置的误差值，经过位置 PID 控制器运算后，输出车辆现在的目标速度，再与速度观测器输出的速度做差获得速度误差值，将速度误差值输入速度 PID 控制器后，输出一个加速度信号给车辆模型，而加速度为车辆模型的控制变量，车辆在得到加速度信号后会改变自己的加速度，从而使速度达到期望值，而速度的改变使得

车辆可以到达目标位置。

理想的串级 PID 控制器希望在位置 PID 控制器输出下一个信号前,速度环已经达到稳态,因此速度环的步频应该达到位置环的 4-10 倍。

根据上一章规划出的路径,车辆具有前进部分和倒车部分两个环节,而直接使用串级 PID 控制器难以处理两部分之间连接的部分。因此本论文将整个纵向控制过程也按照前进部分和倒车部分分成两个环节。

前进部分的目标位置为倒车点,当车辆到达目标位置附近后,结束前进部分的控制,进入倒车部分,此时目标位置变为停车点,纵向控制器再控制车辆到达停车点,从而完成整个纵向控制环节。

4.2 横向控制器设计

横向控制器需要通过控制车辆上的一个控制点,使这个点可以沿着规划好的路径运动,从而实现车辆的跟随。在泊车问题的低速环境下,适合使用基于几何学模型的控制,因为这种控制方法可以忽略车辆的动力学模型。目前常用的基于几何学模型的控制方法有纯跟踪方法和 Stanley 算法,其中,Stanley 算法是一种以前轮轴心为控制点,应用于前轮转向车辆的跟踪控制方法,难以部署到四轮转向车辆上,而纯跟踪算法是一种以后轮轴心为控制点,应用于前轮转向车辆的跟踪控制方法,而在四轮转向车辆控制器设计中,可以将控制点移动到车辆的几何中心来进行横向控制器设计,因此本章选择纯跟踪控制在设计横向控制器。

纯跟踪算法需要选取一个路径上的预瞄点,选取方法为先找到路径上距离控制点最近的路径点,取该路径点向前若干个点后的路径点为预瞄点。此时车辆与预瞄点之间的位置关系如图 4-2 所示。

预瞄点 C 的坐标为 (x_{pre}, y_{pre}) ,控制点 G 的坐标为 (x, y) ,控制点与预瞄点的距离为 L_d ,连线 GC 和车辆夹角为 α ,那么现在的目标是希望车辆运动一个圆弧后到达预瞄点 C,画出圆弧后,由几何关系可知圆弧所对应的圆心角为 2α ,转弯圆心为 O' ,在三角形 CGO' 中,根据正弦定理,有:

$$\frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} = \frac{L_d}{\sin(2\alpha)} \quad (4-5)$$

将式(2-14)代入式(4-5)中,可以得到转向角 δ 的表达式为:

$$\delta = \arctan\left(\frac{L \sin \alpha}{L_d}\right) \quad (4-6)$$

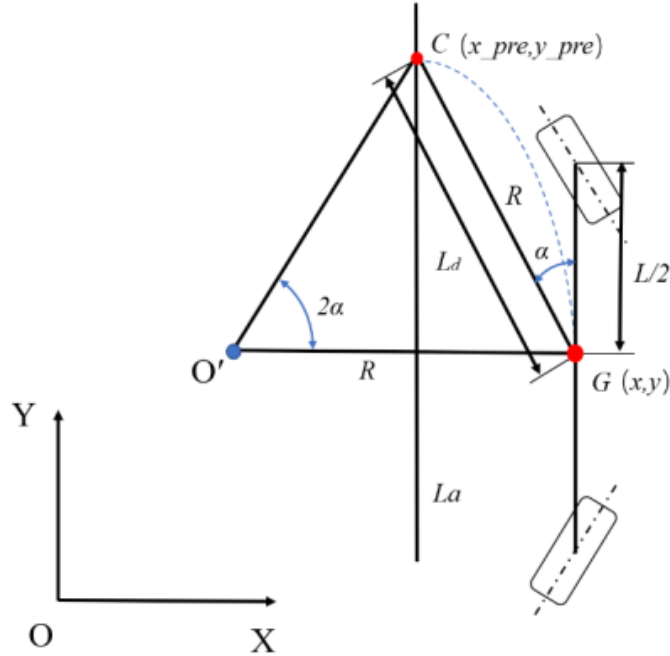


图 4-2 基于四轮转向的纯跟踪算法示意图

此时 δ 就表示在控制点位于 $G(x, y)$ 时, 车辆的前轮转向角应该为 δ , 后轮转向角应该为 $-\delta$, 这样四轮转向车辆就可以跟随路径运动了。

观察式(4-6)关于 δ 的表达式, 其中 L 是车辆本身的长度, 是一个常量, α 反应的是车辆目前偏移路径的程度, 取决于设计的横向控制器对路径的跟踪能力, 无法在设计横向控制器时直接对其进行控制, 而 L_d 代表预瞄点与控制点之间的距离, 取决于预瞄点选取的距离, 因此在设计控制器时 L_d 可以通过预瞄点选取的位置来进行控制。

从式(4-6)可知, L_d 越大, δ 越小, 车辆转弯的曲率越小, 此时车辆实际运动的路径就越平滑, 但车辆跟随路径的速度就越慢, 导致实际路径与目标路径的误差难以消除, 并且转弯时容易出现无法跟随的现象。 L_d 越小, δ 越大, 车辆转弯的曲率越大, 此时车辆跟随路径的效果就越好, 但由于较大的转向角, 车辆在沿着直线目标路径时可能会出现摆动现象。而根据日常经验, 驾驶车辆直线行驶时, 视线应当向远看, 保证车辆行驶平稳, 在设计横向控制器时对应着选取较远的预瞄点, 而在驾驶车辆转弯时, 视线应当向近看, 保证车辆准确跟踪弯道, 在设计横向控制器时对应着选取较近的预瞄点。在工业上, 我们常用路径曲率来表现当前路径是直线还是弯道, 因此在设

计横向控制器时，我们需要选取一个观测点来观测当前路径的曲率大小，实际设计中我们可以选择路径上距离控制点最近的路径点为观测点。观测点曲率小，说明目前车辆正在进行直线行驶，则预瞄点需要选取的远；观测点曲率大，说明目前车辆正在转弯，则预瞄点需要选取的近。根据以上信息设计出的横向控制系统如图 4-3 所示。

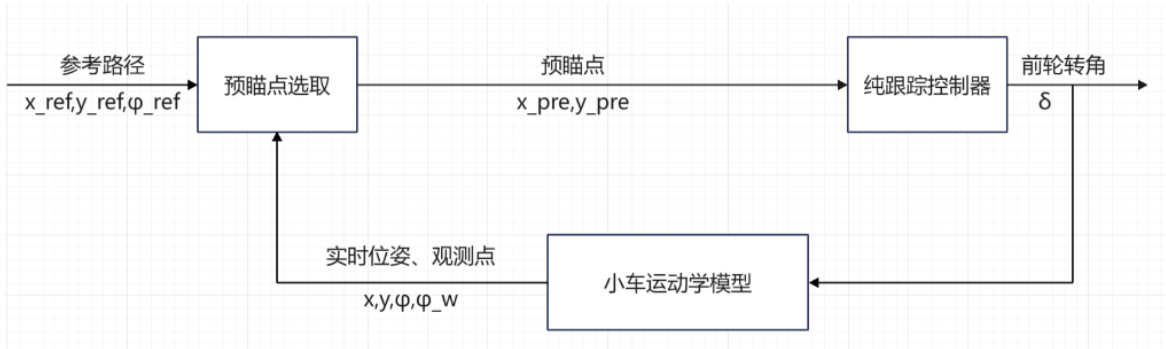


图 4-3 横向控制器控制系统

首先系统根据目前车辆实际位姿与参考路径找出观测点，在计算出观测点曲率后选取路径上的预瞄点，接下来根据车辆实时位姿与预瞄点信息，纯跟踪控制器会计算出车辆前后轮此时的转向角大小，从而控制车辆跟踪参考路径。

与纵向控制类似，横向控制也需要将控制过程分为前进部分和倒车部分。前进部分以起始位置为起点，倒车点为终点进行横向控制，若预瞄点的选取超出倒车点，则以倒车点为预瞄点，当车辆到达倒车点时，执行倒车部分，此时以倒车点为起点，停车点为终点，若预瞄点的选取超出停车点，则以停车点为预瞄点，当车辆到达停车点时，泊车完成。

由于四轮转向车辆自身的特性以及控制点从后轮中心移动到几何中心，四轮转向车辆前进和倒车的模型是完全对称的，因此在倒车时只需将前轮看作后轮，后轮看作前轮，就可以将倒车部分的控制转换成前进部分的控制。

4.3 仿真模型搭建与控制器仿真实验

完成控制器的设计后，还需要进行仿真实验来验证控制器的可行性，本章采用了 CarSim 与 Simulink 联合仿真来验证本章设计的纵向控制器与横向控制器的可行性。

CarSim 是专门针对车辆动力学的仿真软件，用户可以直接在界面上搭建车辆模型，并且模型在计算机上运行的速度比实时快 3-6 倍。除此之外，CarSim 还可以仿真

车辆对驾驶员、路面及空气动力学输入的响应。CarSim 主要用来预测和仿真汽车整车的操纵稳定性、制动性以及平顺性，并且可以方便灵活的定义试验环境和试验过程。除此之外，CarSim 还可以将用户搭建好的车辆模型上传到 Simulink 来进行控制器的设计与搭建。本章便是利用这一功能来完成控制器的仿真实验。

本章选择车辆的长度为 3.6 米，宽度为 2 米，轴距为 2.8 米，则控制点的选取应该在车辆的中心。而 CarSim 的原点设定在车辆前轮中心，因此需要坐标变换到车辆的几何中心处，如图 4-4 所示。

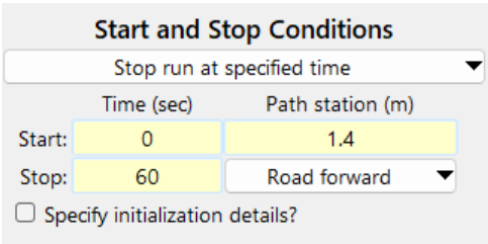


图 4-4 将 CarSim 的原点移动到车辆几何中心

同时，由于车辆为四轮转向车辆，因此需要保证 CarSim 的车辆模型的前后轮转向都可以直接控制，CarSim 的转向系统设置如图 4-5 所示。

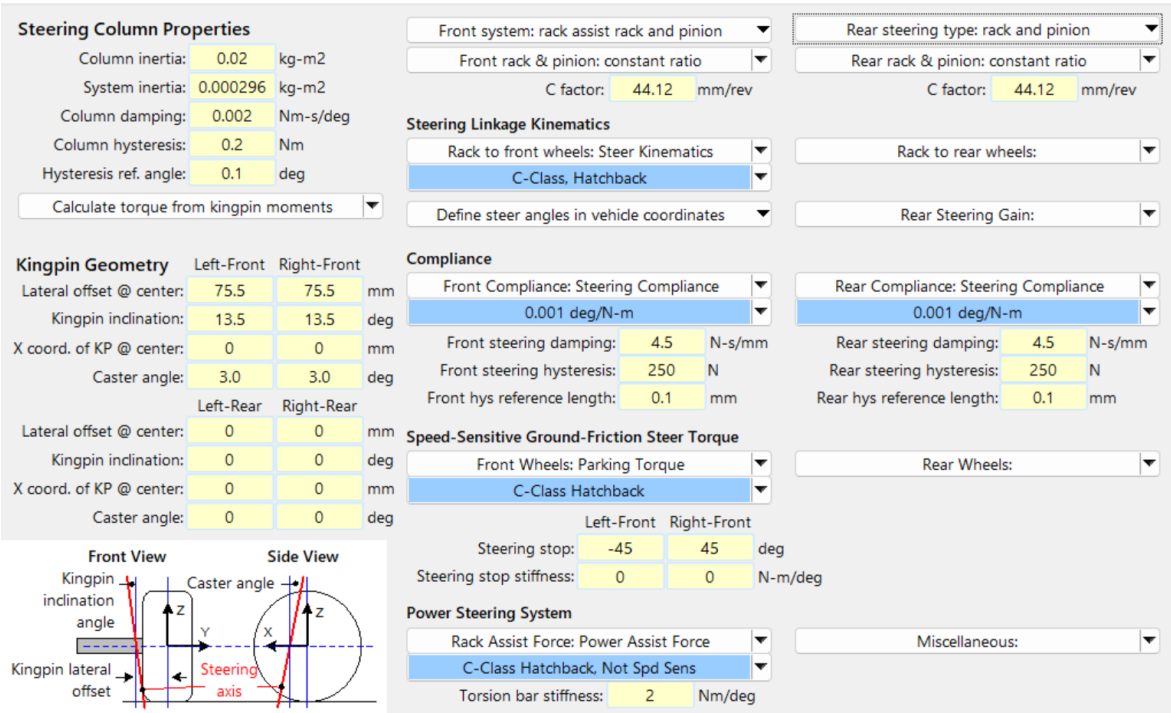


图 4-5 CarSim 转向部分设置

这样一来，就可以把 CarSim 上搭建好的模型传送到 Simulink 中，进行控制器的实现。

接下来就是在 Simulink 中搭建设计好的控制器，如图 4-6 所示。

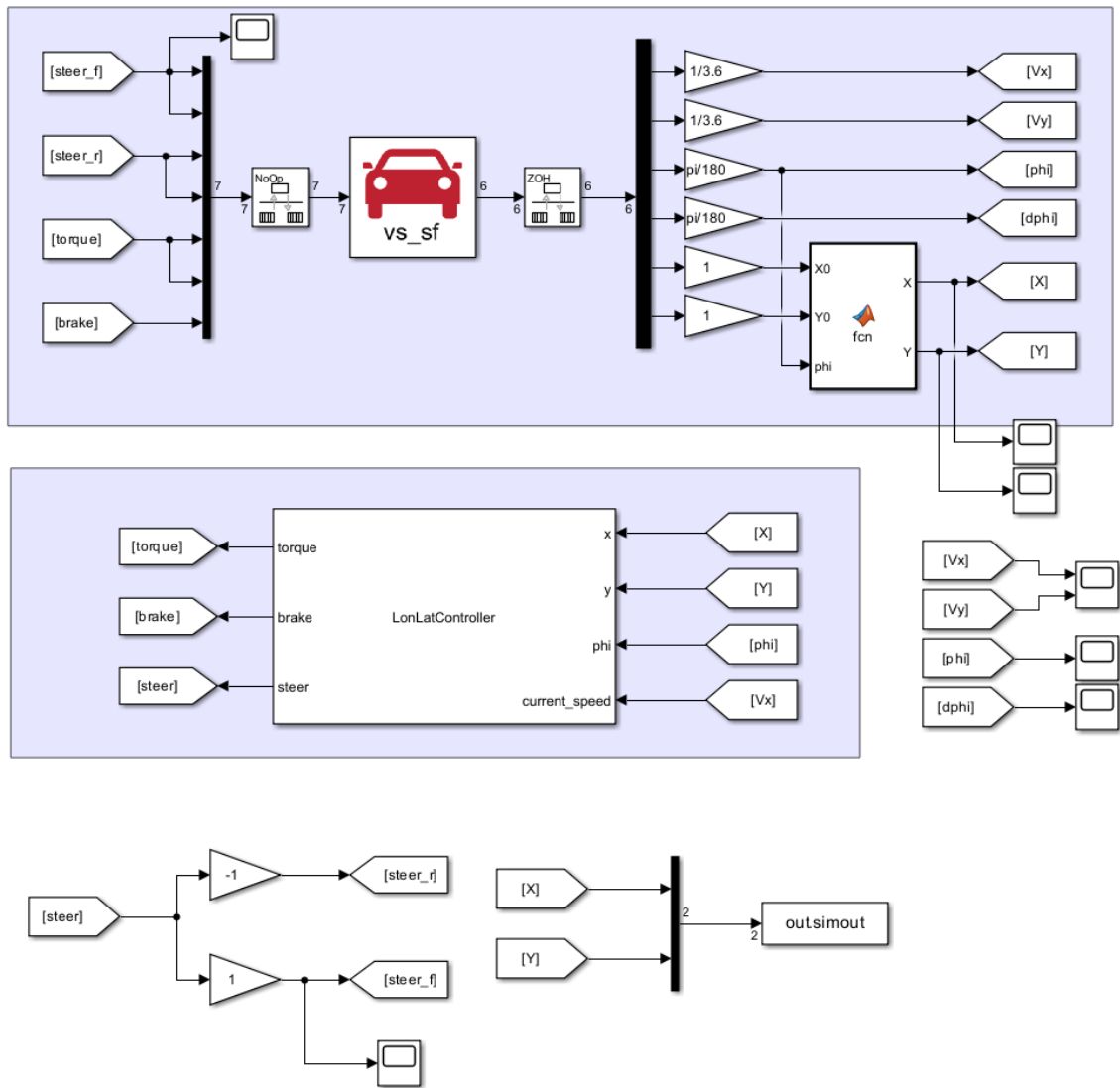


图 4-6 Simulink 搭建控制器

由于 CarSim 的坐标系是跟随车辆运动的，因此先使用 MATLAB Function 将车辆坐标转换成初始参考系下的坐标。而 LonLatController 则是根据本章前两节设计出的控制器，根据车辆当前位置、速度以及方向角来计算此刻车辆的加速度和转向角应该为多少，其中加速度由转矩和刹车来控制。仿真结果如图 4-7、4-8 所示。

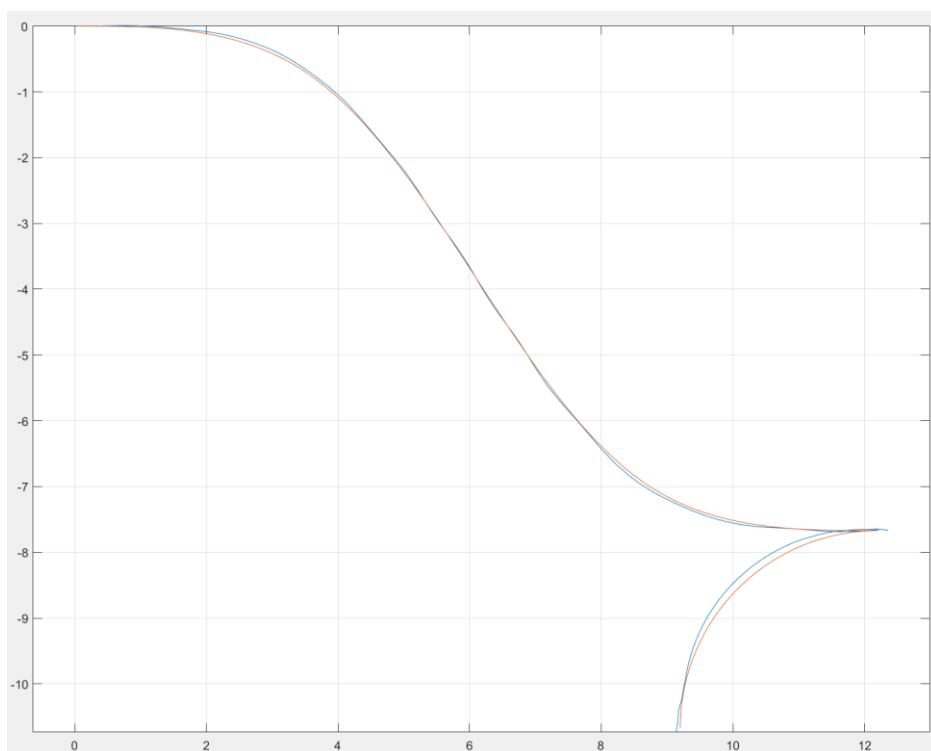


图 4-7 CarSim 与 Simulink 联合仿真垂直停车位仿真结果

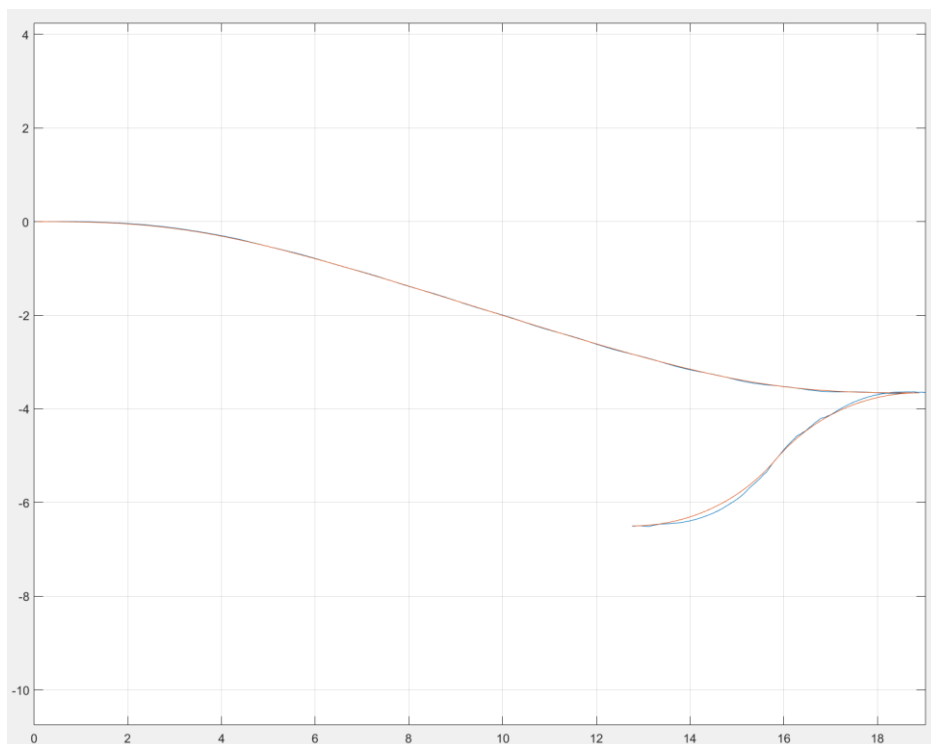


图 4-8 CarSim 与 Simulink 联合仿真平行停车位仿真结果

其中红色路径为参考路径，即第三章中规划出的路径，蓝色路径为车辆实际运行路径，可以看出车辆跟随效果较好，本章设计的控制器可行。在 CarSim 中的仿真结果如图 4-9、4-10 所示。



图 4-7 CarSim 垂直停车位仿真结果

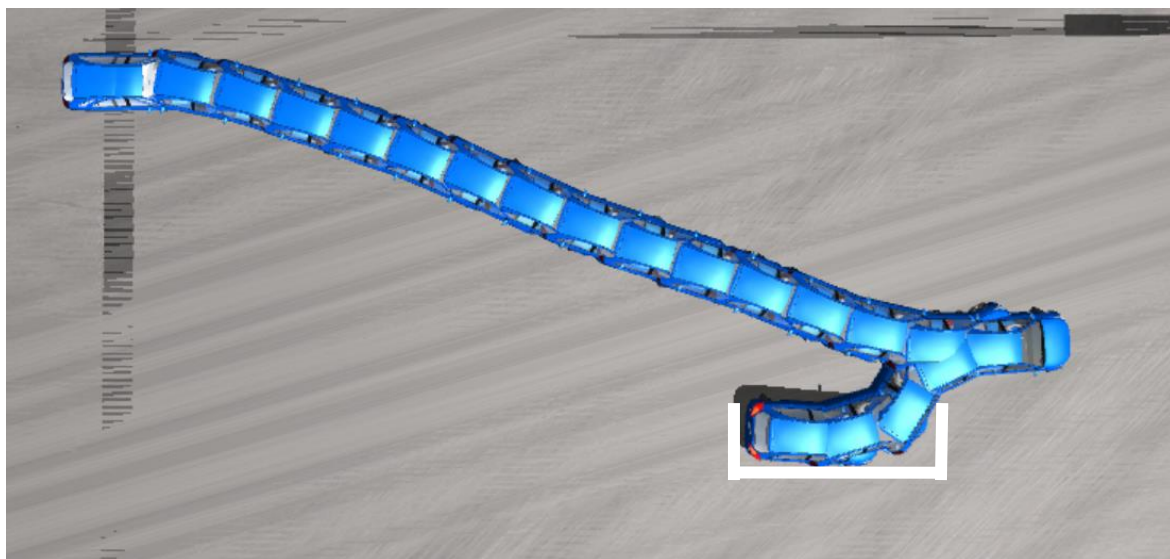


图 4-8 CarSim 平行停车位仿真结果

4.4 本章小结

本章进行了基于四轮转向的运动跟踪控制器设计，并对控制器进行了仿真实验。

由于泊车环境下，车辆处于低速状态，因此控制器可以解耦为纵向控制器和横向控制器来分别进行设计。纵向控制器用于控制车辆的纵向速度，用于实现车辆速度的控制，本章采用了串级 PID 控制器，使得车辆可以顺利到达目标点。横向控制器用于控制车辆跟随路径，本章采用了纯跟踪控制，并通过将控制点移动到车辆几何中心上来将它部署在四轮转向车辆上，并且根据观测点的曲率控制预瞄点的选取距离，使得车辆可以平稳的跟随路径。而由于上一章中设计的路径分为前进部分和倒车部分，因此本章的控制器也分为前进部分和倒车部分来分别进行控制，在执行完前进部分后，控制器才会进入下一阶段，执行倒车部分。

此外，本章还利用 CarSim 与 Simulink 联合仿真来验证控制器的可行性，最终的仿真结果良好，成功验证了本章设计的控制器是可行的。

第五章 全文总结

5.1 研究总结

本论文以四轮转向车辆为研究对象，完成了四轮转向车辆的建模、碰撞检测分析、泊车路径规划算法设计以及运动跟踪控制器设计，并通过仿真实验验证了该泊车规划控制系统的可行性。

第一章对研究背景进行了介绍，并提出了四轮转向车辆在复杂泊车环境下具有更好的适应性。对国内外在路径规划以及控制器设计的研究现状进行了介绍和比较，并从中选取了适用于四轮转向车辆的方法进行后续实验。

第二章对四轮转向车辆进行了建模，主要分为运动学建模和几何建模两个方面。运动学建模根据车辆自身运动学特性得到了四轮转向车辆的质心运动学方程，并计算出了车辆转向角和转弯半径之间的关系。几何建模则是将车辆外轮廓简化为了一个便于计算的、修正过的矩形。除此之外还对停车场环境进行了建模，将环境中的障碍物简化为多个矩形组成的障碍物模型。利用车辆几何模型和障碍物模型进行后续的碰撞检测分析。根据障碍物和车辆路径的几何类型，将碰撞检测分析分为线段与线段、线段与圆弧、圆弧与圆弧三种类型的碰撞检测分析，并对每种类型进行了研究，得出了一套运算效率较高的碰撞检测分析方法。

第三章对基于四轮转向的泊车路径规划算法进行了设计，轨迹规划选用了 RS 曲线，路径规划则采用 RRT* 算法，二者结合得出了一种用于规划满足约束且不发生碰撞的路径的 RRT*-RS 算法。RRT*-RS 算法使用场景广泛，但复杂度较高，因此本论文又设计了一种用于解决泊车问题的快速规划方法。通过解析法计算出泊车路径中前进部分与倒车部分的转换点，即倒车点，并将倒车部分提前快速规划好，对前进部分则使用简化后的 RRT*-RS 算法，即不考虑倒车情况，从而提高路径规划效率。最后进行 Matlab 仿真实验，成功得出一条满足约束且避障的路径。

第四章对基于四轮转向的运动跟踪控制器进行了设计。首先将泊车问题中的控制器进行解耦，分别设计纵向控制器和横向控制器。纵向控制器选用了串级 PID 控制器，这种方法适用于低速且对速度精度要求不高的控制系统。横向控制器选用了纯跟踪控制器，并基于四轮转向车辆进行了改进，将控制点移动到车辆几何中心，并用四

轮转向车辆运动学模型计算出了对应的转向角,并对预瞄点的选取进行了反馈控制使其根据路径曲率变化。最后利用 CarSim 与 Simulink 联合仿真对控制器进行了仿真实验,实验结果验证了控制器的可行性。

综上,本论文完成了基于四轮转向的泊车规划控制系统设计,且仿真结果可行。

5.2 研究展望

本论文在设计泊车路径规划算法时采用了 RRT*-RS 算法,但由于 RRT*算法的采样点是随机采样得到的,存在大量采样点浪费、路径收敛速度较慢的情况,可以通过改进采样点采样方式来改善 RRT*-RS 算法的运算效率,但由于研究时间有限,本论文没有针对 RRT*-RS 算法本身给出改进方法,在后续研究中可以对 RRT*-RS 算法的采样方式进行改进研究。

此外,在设计运动跟踪控制器中的横向控制器时,由于泊车问题的低俗特性,本论文并没有根据车辆动力学模型设计横向控制器,因此在实际控制过程中必然会有误差的出现,并不适用于如结冰路段等极限工况,在后续研究中可以将一些基于动力学的控制器如鲁棒控制、模糊控制等控制器部署在四轮转向车辆上用以提高控制性能。

之后可以将泊车规划控制系统部署在四轮转向模型的车型上进行应用,将泊车环节交由自动驾驶系统来完成,从而提高泊车的安全性与舒适性。

5.3 非技术性分析

近年来,自动驾驶系统在家用车上的部署越来越广泛,许多家用车已经达到了 L2 级甚至 L3 级的自动驾驶等级,然而,尽管自动驾驶技术日益成熟,在某些场景下,自动驾驶系统依然存在安全性、伦理性以及法律性的问题。

自动驾驶系统的功能是辅助乃至代替驾驶员操纵车辆,但交通事故依然无处不在,自动驾驶尽管可以降低事故发生的风险,但当事故发生时,责任的划分在目前依然争论不休,此外,当自动驾驶系统与驾驶员在处理问题上发生分歧时,自动驾驶系统究竟应该继续辅助驾驶员的操作还是根据自身判断选择更安全的行为,这一准则目前依然不明确,这些问题也导致很多人对自动驾驶功能不信任甚至抵触,也导致很多乘客不愿意让驾驶员使用自动驾驶系统。

要想提高自动驾驶系统在广大驾驶员以及乘客心中的信任度,不仅仅需要工程师

与研究员不断地提高自动驾驶系统的安全性及稳定性，也需要国家和政府尽快制定相关的法律法规，让事故发生时有法可依，此外，政府还应当尝试为自动驾驶系统提供相应的基础设施，为自动驾驶系统提供便利。只有所有人共同努力，才能让自动驾驶系统在群众中可以广泛普及。

参 考 文 献

- [1] Society of Automotive Engineers. J3016_202104 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles[EB/OL].(2021-04-30)[2023-12-26].https://www.sae.org/standards/content/j3016_202104/.
- [2] 余卓平, 李奕姗, 熊璐. 无人车运动规划算法综述[J]. 同济大学学报(自然科学版), 2017, 第 45 卷 (8): 1150-1159.
- [3] KAVRAKI L, KOLOUNTZAKIS M, LATOMBE J C. Analysis of probabilistic roadmaps for path planning[J]. IEEE Transactions on Robotics and Automation, 1998, 14(1): 166-171.
- [4] LAVALLE S, KUFFNER J. Randomized kinodynamic planning[C]//Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C): vol. 1. 1999: 473-479 vol.1.
- [5] KUFFNER J, LAVALLE S. RRT-connect: An efficient approach to single-query path planning[C]//Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065): vol. 2. 2000: 995-1001 vol.2. D.
- [6] URMSON C, SIMMONS R. Approaches for heuristically biasing RRT growth[C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453): vol. 2. 2003: 1178-1183 vol.2.
- [7] CHENG P, LAVALLE S. Reducing metric sensitivity in randomized trajectory design[C]//Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180): vol. 1. 2001:43-48 vol.1.
- [8] FERGUSON D, STENTZ A. Anytime RRTs[C]//2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006: 5369-5375.
- [9] KARAMAN S. Sampling-Based Algorithms for optimal path planning Problems[D]. AAI0829068.USA: Massachusetts Institute of Technology, 2012.
- [10] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2014: 2997-3004.
- [11] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [12] FURCY D A, KOENIG S. Speeding up the convergence of online heuristic search and scaling up offline heuristic search[D]. AAI3154931. USA: Georgia Institute of Technology, 2004.
- [13] AINE S, SWAMINATHAN S, NARAYANAN V, et al. Multi-Heuristic A*[J/OL]. Int. J. Rob. Res., 2016, 35(1-3): 224-243.<https://doi.org/10.1177/027836491559/>.
- [14] ISLAM F, NARAYANAN V, LIKHACHEV M. A*-Connect: Bounded suboptimal bidirectional heuristic search[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). 2016:

- 2752-2758.
- [15] KOENIG S, LIKHACHEV M, FURCY D. Lifelong planning A*[J/OL]. Artificial Intelligence, 2004, 155(1): 93-146. <https://www.sciencedirect.com/science/article/pii/S0004370203/>.
- [16] DUBINS L E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents[J/OL]. American Journal of Mathematics, 1957, 79(3): 497-516 [2023-12-27]. <http://www.jstor.org/stable/2372>.
- [17] REEDS J, SHEPP L. Optimal paths for a car that goes both forwards and backwards[J]. Pacific journal of mathematics, 1990, 145(2): 367-393.
- [18] WILDE D K. Computing clothoid segments for trajectory generation[C]//2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2009: 2440-2445.
- [19] KOMORIYA K, TANIE K. Trajectory design and control of a wheel-type mobile robot using B-spline curve[C]//Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications. 1989: 398-405.
- [20] 熊璐, 杨兴, 卓桂荣, 冷搏, 章仁燮. 无人驾驶车辆的运动控制发展现状综述[J]. 机械工程学报, 2020, 第 56 卷: 127-143.
- [21] THRUN S, MONTEMERLO M, DAHLKAMP H, et al. Stanley: The robot that won the DARPA Grand Challenge[M/OL]//BUEHLER M, IAGNEMMA K, SINGH S. The 2005 DARPA Grand Challenge: The Great Robot Race. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 1-43.
- [22] XU S, PENG H, SONG Z, et al. Accurate and smooth speed control for an autonomous vehicle[C]//2018 IEEE Intelligent Vehicles Symposium (IV). 2018: 1976-1982.
- [23] ZHU M, CHEN H, XIONG G. A model predictive speed tracking control approach for autonomous ground vehicles[J/OL]. Mechanical Systems and Signal Processing, 2017, 87: 138-152. <https://www.sciencedirect.com/science/article/pii/S0888327016001205>.
- [24] WANG J, SUN Z, XU X, et al. Adaptive speed tracking control for autonomous land vehicles in all - terrain navigation: An experimental study[J/OL]. Journal of Field Robotics, 2013, 30. <https://api.semanticscholar.org/CorpusID:35484373>.
- [25] COULTER C. Implementation of the pure pursuit path tracking algorithm[C/OL]//. 1992. <https://api.semanticscholar.org/CorpusID:62550799>.
- [26] MARINO R, SCALZI S, NETTO M. Nested PID steering control for lane keeping in autonomous vehicles[J/OL]. Control Engineering Practice, 2011, 19(12): 1459-1467. <https://www.sciencedirect.com/science/article/pii/S0967066111001808>.
- [27] SNIDER J M. Automatic steering methods for autonomous automobile path tracking[C/OL]//. 2009. <https://api.semanticscholar.org/CorpusID:17512121>.
- [28] 余卓平, 章仁燮, 熊璐, 黄楚骏. 基于条件积分方法的无人差动转向车辆动力学控制[J]. 机械工程学报, 2017, 第 53 卷(14): 29-38.
- [29] FALCONE P, BORRELLI F, ASGARI J, et al. Predictive active steering control for autonomous vehicle systems[J]. IEEE Transactions on Control Systems Technology, 2007, 15(3): 566-580.
- [30] 宋彦, 赵盼, 陶翔, 李碧春, 梁华为, 梅涛. 基于 μ 综合的无人驾驶车辆路径跟随串级鲁棒控制方法[J]. 机器人, 2013, 第 35 卷(4): 417-424.
- [31] CHU Z, SUN Y, WU C, et al. Active disturbance rejection control applied to automated steering for

- lane keeping in autonomous vehicles[J/OL]. Control Engineering Practice, 2018, 74: 13-21.
<https://www.sciencedirect.com/science/article/pii/S0967066118300121>.
- [32] YU Z, WANG J. Automatic vehicle trajectory tracking control with self-calibration of nonlinear tire force function[C]//2017 American Control Conference (ACC). 2017: 985-990.
- [33] ALCALA E, PUIG V, QUEVEDO J, et al. Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning[J/OL]. Control Engineering Practice, 2018, 73: 1-12.
<https://www.sciencedirect.com/science/article/pii/S0967066117302721>.
- [34] LIM E, HEDRICK J. Lateral and longitudinal vehicle control coupling for automated vehicle operation[C]//Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251): vol. 5. 1999: 3676-3680 vol.5.
- [35] GAO Y. Model predictive control for autonomous and semiautonomous vehicles[C/OL]//.2014.
<https://api.semanticscholar.org/CorpusID:110450172>.

致 谢

初稿暂不致谢。

DESIGN OF PARKING PLANNING AND CONTROL SYSTEM BASED ON FOUR-WHEEL STEERING

This thesis takes four-wheel steering vehicles as the research object, completes the modelling of four-wheel steering vehicles, collision detection analysis, parking path planning algorithm design and motion tracking controller design, and verifies the feasibility of this parking planning control system through simulation experiments.

Chapter 1 introduces the research background and research motivation. At present, with the increasing number of cars, the supply of urban parking spaces cannot meet the demand, and the congestion of car parks is becoming more and more prominent, which brings a lot of difficulties and problems to the driver's parking experience; therefore, drivers need automatic parking systems to reduce the burden of drivers, improve the safety of parking in complex environments, and provide drivers with a more convenient and comfortable parking experience. In addition, most of the current family cars use the front-wheel steering model, which has a large turning radius, resulting in parking difficulties in crowded or complex parking environments, and requires more space to successfully complete the parking, while the four-wheel steering model of the four-wheel steering model has a steering function, which can effectively reduce the turning radius of the vehicle, thus reducing the space required for parking, improving the utilisation of the parking space and enhancing the parking process. comfort, and shows stronger adaptability to more complex and crowded parking environments. Afterwards, this chapter introduces and compares the current research status of path planning and controller design at home and abroad, analyses the characteristics and application scenarios of various path planning methods and controller design methods, and then for the characteristics of the four-wheel steering model, selects the methods suitable for four-wheel steering vehicles for the discussion and research in the subsequent chapters.

Chapter 2 models the four-wheel steering vehicle and the car park environment, in which the four-wheel steering vehicle modelling consists of two parts: the kinematic modelling of the four-wheel steering vehicle and the geometrical modelling. The kinematic modelling is based on the analysis and calculation of the vehicle's own characteristics, the

kinematic equations of the vehicle's centre of mass and the relationship between the steering angle and the turning radius are derived, and the proof that the turning radius of the four-wheeled-steering vehicle is smaller than that of the front-wheeled-steering vehicle is mathematically given. The kinematic model is used for two subsequent parts of path planning and controller design based on four-wheel steering. The relationship between the steering angle and turning radius of the four-wheel steering vehicle is used to determine the kinematic model constraints of the vehicle during trajectory planning to ensure that the curvature of the planned path does not exceed the maximum steering curvature of the four-wheel steering vehicle, so that the path satisfies the vehicle kinematic model constraints, thus ensuring that the vehicle has the ability to follow the path. The vehicle centre-of-mass kinematic equations are used in the controller design to determine the relationship between the vehicle speed and attitude, so that the corresponding longitudinal controller can be designed to control the speed of the four-wheel steering vehicle. In addition, the relationship between steering angle and turning radius of the four-wheel steering vehicle is also used to calculate the target steering angle of the vehicle at the current position when designing the lateral controller. The vehicle geometry model and the car park environment model simplify the vehicle path and obstacles into a geometry consisting of a number of line segments and arcs, which are used for the subsequent collision detection study. In the collision detection study, all possible collision situations are firstly classified into three situations of line segments and line segments, line segments and circular arcs, and circular arcs and circular arcs according to the type of geometries, and the collision detection analysis is carried out for the three situations in turn, and simple geometrical relations are used to exclude some situations where collision is certain not to occur, and then analytical method is used to judge the remaining situations, which speeds up the calculation efficiency and successfully completes the collision detection Analysis.

Chapter 3 completes the design of the parking path planning algorithm based on four-wheel steering. Firstly, the parking path planning is divided into two parts: trajectory planning and path planning. Trajectory planning needs to get a trajectory that meets the constraints of the four-wheel steering vehicle kinematics model, due to the existence of reversing situations in the parking environment, this chapter selects the RS curve that allows reversing to carry out trajectory planning and selects several situations in which there are

only three paths to carry out calculations. Path planning is required to obtain a feasible path that meets obstacle avoidance, due to the point cloud map used in the subsequent experiments, this chapter chooses the more mature RT* algorithm to obtain a feasible path, and replaces the connection of the nodes with RS curves for the calculation of the distance therein. Due to the high computational complexity of the RT*-RS algorithm, for the parking problem, this chapter proposes a fast planning method based on the analytical method and the RT*-RS algorithm, which divides the parking process into two parts, the forward process and the reverse process; the reverse process is solved by the analytical method with the analytical solution of the reversing point, which reduces the complexity of the trajectory planning of the reversing process to a minimum, while the forward part of the The reversing part of the RS curve can be neglected, and the efficiency is also improved compared to the original RRT*-RS algorithm. Finally, the simulation of the parking path planning algorithm is carried out through Matlab, and the feasible paths that satisfy the vehicle kinematics constraints are successfully obtained, and the comparison between the four-wheeled transformed vehicle and the front-wheeled steering vehicle is carried out, and it is found that the four-wheeled steering vehicle benefits from the smaller turning radius, and it has shorter parking paths than the front-wheeled steering trolley, and in the environment of parking in some of the extreme parallel parking spaces, the four-wheeled steering vehicle can In some extreme parallel parking environments, the four-wheel steering vehicle can complete parking by reversing once, while the front-wheel steering trolley needs more than two reversals to complete parking. This shows that the four-wheel steering vehicle shows stronger adaptability when dealing with more complex and congested parking environments.

Chapter 4 carries out the design of motion tracking controller based on four-wheel steering and simulation experiments on the controller. Since the vehicle is at low speed in the parking environment, the controller can be decoupled into a longitudinal controller and a lateral controller for separate design. The longitudinal controller is used to control the longitudinal speed of the vehicle, which is used to realise the control of the longitudinal position and speed of the vehicle. In this chapter, a serial PID controller is used so that the vehicle can reach the target point and stop smoothly with the target speed. The lateral controller is used to control the vehicle following path, this chapter adopts the pure tracking control and deploys it on the four-wheel steering vehicle by moving the control point from

the centre of the rear wheels of the vehicle to the geometric centre of the vehicle, and controls the distance of the pre-scanning point according to the curvature of the observation point, when the curvature of the observation point is larger, it means that the vehicle is turning, and the distance of the pre-scanning point is farther away at this time, when the curvature of the observation point is smaller, it means that the vehicle is going straight, and the pre-scanning point is farther away at this time. When the curvature of the observation point is small, it means that the vehicle is going straight, and the distance of the pre-sighting point is closer. This method of feedback control on the distance of the pre-sighting point can make the vehicle follow the path smoothly. As the path designed in the previous chapter is divided into forward and reverse parts, the controller in this chapter is also divided into forward and reverse parts to control them separately, and the controller will enter the next stage to execute the reverse part only after executing the forward part. In addition, this chapter also uses CarSim and Simulink co-simulation to verify the feasibility of the controller, and the final simulation results are good, successfully verifying that the controller designed in this chapter is feasible.

Chapter 5 summarises the research content of this thesis and puts forward possible solutions to the problems that still exist in it, and in the end, it provides a non-technical analysis of the automatic driving technology and an outlook on the future development of automatic driving technology.

In summary, this paper proposes a more efficient collision detection analysis method and path planning method for the parking problem, and also improves the pure tracking controller for four-wheel steering vehicles, which not only provides a theoretical basis for the automatic parking of four-wheel steering vehicles, but also contributes a new technical solution for the application and development of the automatic driving technology in the field of parking.