

分类号：U471.15
U D C：629

密级：
编号：

工程硕士学位论文

基于 OpenCV 的自动泊车技术研究

硕士研究生：王若羽
指导教师：王靖岳 教授
学科、专业：车辆工程

沈阳理工大学

2022 年 03 月

分类号：U471.15
U D C：629

密级：
编号：

工程硕士学位论文

基于 OpenCV 的自动泊车技术研究

硕士研究生：王若羽
指导教师：王靖岳 教授
学位级别：工程硕士
学科、专业：车辆工程
所在单位：汽车与交通学院
论文提交日期：2021 年 12 月 9 日
论文答辩日期：2022 年 3 月 11 日
学位授予单位：沈阳理工大学

Classification Index:U471.15

U.D.C: 629

A Thesis for the Master Degree of Science in Engineering

Research on automatic parking technology based on OpenCV

Candidate : Wang Ruoyu

Supervisor: Prof. Wang Jingyue

Academic Degree Applied for: Master of Science in Engineering

Speciality: Vehicle Engineering

Date of Submission: December 9th,2021

Date of Examination: March 11th,2022

University: Shenyang Ligong University

摘 要

针对目前停车难现象，本文针对自动泊车技术进行研究，分别展开讨论自动泊车系统的三大模块：环境感知、路径规划、路径跟踪，相应提出了基于边缘编组的车位线检测方法，规划三种不同车位类型的泊车路径，设计相应模糊控制器仿真验证车辆自动泊车运动过程。

首先，本文基于 OpenCV 开源软件作为自动泊车系统车位线检测的基础，对泊车位图像进行预处理，包括灰度化、直方图均衡化、二值化和形态学去噪等操作运算，减少干扰因素，提高图像质量；之后分析对比了几种常用的边缘检测算法，确定采用 Canny 算法作为车位线边缘检测方法，并改进其梯度计算模板。然后对边缘图像进行 8 邻域边缘编组，得到独立的互相连接的边缘组，消除了噪声和短线的影响，接着针对非车位线边缘组中梯度方向不一致这一特征，运用差分概念剔除边缘组中杂乱的信息边缘，再对剩余边缘组做概率霍夫变换，最终检测得到车位线。结果表明本文提出的车位线检测算法相较于传统霍夫检测，减少了车位线识别时间，提高了检测正确率。

其次，在车辆泊车过程中建立运动坐标系，并结合阿克曼转向原理建立车辆运动学模型，对车辆运动过程进行详细分析，针对平行、垂直和斜式泊车，规划设计了相应泊车路径，确定三种泊车类型的泊车位尺寸。

最后，采用模糊控制理论设计三种泊车类型相应模糊控制器，确定模糊控制器输入输出量、隶属度函数和模糊规则等，实现不同泊车路径跟踪控制，再根据不同泊车类型设定不同起始泊车条件进行仿真，结果表明车辆均可以完成自动泊车操作，验证了之前路径规划的有效性和模糊控制器的可靠性。

关键词：自动泊车；OpenCV；路径规划；路径跟踪；模糊控制

Abstract

In view of the current parking difficulty, this thesis studied the automatic parking technology, and discussed the three modules of the automatic parking system: environment perception, path planning and path tracking. Accordingly, a parking line detection method based on edge grouping was proposed, plan the parking paths of three different types of parking spaces, the corresponding fuzzy controller was designed to simulate and verify the motion process of vehicle automatic parking.

Firstly, based on OpenCV open source software as the basis of parking line detection of automatic parking system, this thesis preprocessed the parking space image, including grayscale, histogram equalization, binarization, morphological denoising and other operation operations, so as to reduce interference factors and improve image quality; After that, several common edge detection algorithms were analyzed and compared, Canny algorithm was determined as the edge detection method of parking space line, and its gradient calculation template was improved. Then, the edge image is grouped into 8 neighborhood edges to obtain independent interconnected edge groups, which eliminated the influence of noise and short lines. Then, aimed at the feature of inconsistent gradient direction in the edge group of non-parking space lines, the difference concept was used to eliminate the messy information edges in the edge group, and then the probability Hough transform was performed on the remaining edge groups to finally detect the parking space lines. The results showed that the parking space line detection algorithm proposed in this thesis reduces the parking space line recognition time and improved the detection accuracy compared with the traditional Hough detection.

Secondly, the motion coordinate system was established in the process of vehicle parking, and the vehicle kinematics model was established combined with Ackerman steering principle. The vehicle motion process was analyzed in detail. For parallel,

vertical and oblique parking, the corresponding parking paths are planned and designed, and the parking space dimensions of three types of parking were determined.

Finally, the fuzzy control theory was used to design the corresponding fuzzy controllers for three parking types, determined the input and output of the fuzzy controller, membership function and fuzzy rules, realized different parking path tracking control, and then set different initial parking conditions according to different parking types for simulation. The results showed that all vehicles can complete automatic parking operation, verified the effectiveness of previous path planning and the reliability of fuzzy controller.

Key words: Automatic parking; OpenCV; Path planning; Path tracking; Fuzzy control

目 录

第 1 章 绪论	1
1.1 论文研究背景及意义	1
1.2 自动泊车系统简介	2
1.3 自动泊车技术国内外研究与应用现状	4
1.3.1 自动泊车技术国外研究现状	4
1.3.2 自动泊车技术国内研究现状	6
1.3.3 自动泊车技术车企应用现状	8
1.4 论文主要研究内容	9
第 2 章 基于边缘编组的车位线检测	11
2.1 OpenCV 概述	11
2.1.1 计算机视觉	11
2.1.2 OpenCV 开源工具	12
2.1.3 泊车位标记法	13
2.2 图像预处理	13
2.2.1 灰度化	13
2.2.2 直方图均衡化	15
2.2.3 二值化	16
2.2.4 形态学去噪	19
2.3 边缘检测	21
2.3.1 几种边缘检测算法	21
2.3.2 改进的 Canny 边缘检测算法	25
2.4 车位线检测	27
2.4.1 边缘编组	27
2.4.2 剔除非车位线特征边缘组	29
2.4.3 霍夫变换方法选择	30

2.5 仿真验证	31
2.6 本章小结	34
第 3 章 车辆运动学模型建立及分析	35
3.1 泊车参数描述	35
3.2 车辆运动坐标系	36
3.3 阿克曼转向原理	38
3.4 车辆运动学模型	39
3.4.1 低速泊车运动学模型	39
3.4.2 泊车运动稳态分析	40
3.4.3 车速对泊车系统的影响	41
3.5 本章小结	42
第 4 章 泊车路径规划策略	43
4.1 泊车类型	43
4.2 泊车过程分析及路径规划	44
4.2.1 平行泊车路径规划	44
4.2.2 垂直泊车路径规划	46
4.2.3 斜式泊车路径规划	47
4.3 泊车位尺寸	48
4.4 本章小结	48
第 5 章 模糊控制	49
5.1 MATLAB 简介	49
5.2 模糊控制理论概述	49
5.2.1 模糊控制发展历程及背景	49
5.2.2 模糊控制特点	50
5.2.3 模糊控制主要组成	50
5.3 模糊控制器设计	52
5.3.1 平行泊车模糊控制器设计	52
5.3.2 垂直泊车模糊控制器设计	56

目 录

5.3.3 斜式泊车模糊控制器设计	58
5.4 自动泊车仿真验证	60
5.4.1 平行泊车仿真验证	60
5.4.2 垂直泊车仿真验证	62
5.4.3 斜式泊车仿真验证	65
5.5 本章小结	67
结 论	68
参考文献	70
攻读硕士学位期间发表的论文和获得的科研成果	76
致 谢	77

第 1 章 绪论

1.1 论文研究背景及意义

汽车工业经过几十年的变革期后，从单纯零部件的机械组合变成复杂的、高工艺技术的制造领域，并且逐渐演变成高科技领域，包含不同专业知识、人工智能等先进计算机科学，并且未来汽车行业的发展终将与人工智能、机器视觉、神经网络、传感器技术、计算机编程等多学科领域相结合，不仅仅在人们生活中体现出商业价值，并且会最终改变人类的生活方式。城市交通以及国民经济的快速发展，乘坐机动车出行是人们最便捷的方式，这也必然会使得我国机动车数量和机动车驾驶员人数的急剧上升，截至 2021 年 6 月，全国汽车保有量达到 3.84 亿辆，占到机动车保有量 76.1%，其中机动车驾驶人员达到 4.69 亿人，创历史新高，并且根据数据显示机动车新注册数量为 1871 万辆，其中汽车注册数量为 1414 万辆，占到总数量的 75.57%，同比前年增长 35.86%，机动车辆数目越多，公共交通资源紧缺是必然结果，尤其是停车难现象。

在狭小的空间里，对于一个有着几十年驾驶经验和良好驾驶行为的驾驶员来说，安全地停车和返回也并不是特别容易，倒车时在周围环境不完全了解的情况下，驾驶员只通过两侧后视镜进行泊车操作难度相当大，因为后视镜由于安装原则和个人习惯不同，会影响驾驶者视界，所以驾驶员需要通过扭动身体了解车辆所处环境以及外界障碍物信息，这样的做法十分的不安全。由此可见，倒车过程是一个比较复杂的过程，这取决于驾驶员的经验、驾驶技巧和敏感性，任何随机性问题都会导致驾驶员无法快速准确地完成任务，由于缺乏经验，大多数新手在进入车库或停车场时会感到紧张，虽然倒车速度不快，但发生事故的概率很大，虽然一般情况下不会发生比较严重的伤害，但如果未观察到突发状况或者有小朋友经过车身后方，后果将会相当严重。

自动泊车概念的提出就是为了解决泊车不便，降低泊车过程中出现隐患的可能性，从而将目标车辆快速、安全、可靠的泊入目标泊车区域，该系统的出现有

效减少了驾驶员在泊车过程中所承受的精神压力，让车辆准确的完成泊车操作，使得泊车过程中出现事故的概率大大降低，强大的自动泊车系统具有很高的商业价值以及市场前景，解放驾驶员，提高城市车位利用效率。

1.2 自动泊车系统简介

自动泊车系统是指控制目标车辆自动泊入库位，提高泊车效率，第一步先对外界环境进行感知，识别空闲车位或车位线，成功后，则进行第二步，路径规划，根据车辆自身参数以及周围环境、车位、障碍物的信息，规划出一条泊车路径，之后自动泊车控制器会控制车辆完成之前设计规划好的路径，或者可以总结驾驶员的泊车经验，通过模糊控制器或者其他人工智能手段控制车辆实现自动泊车^[1]。

环境感知技术是自动泊车系统的第一步，通过车载传感器获取车辆周围环境信息以及车位线信息等，这些信息是自动泊车系统进行路径规划以及路径跟踪控制的基础。泊车周围环境检测及识别通常可以通过两种方式实现，两种方法的不同之处就在于是否采用视觉传感器，其中基于非视觉传感器环境感知的类型有：激光雷达、超声波雷达等^[2]。

其中激光雷达的信号源为激光脉冲光束，是由发射器发射出去的，在行进过程中与障碍物（行人、车辆、建筑物等）发生碰撞，引起散射，后被系统接收，发射器与障碍物之间的距离通过激光运行时间和速度所决定^[3-4]。J Zhou 等使用车载激光线扫描仪来检测停车场中的空闲停车位，其原理是检测两车辆保险杠之间距离来判定是否存在空闲泊车位，并使用 AdaBoost 训练机对车辆保险杠数据训练分拣^[5]，还有其他研究人员利用两车辆之间的距离来检测空闲停车位^[6-7]，王海则利用目标泊车位之间障碍物信息检测泊车位^[8]。虽然激光雷达可以较快速的检测出空闲停车位，但是还是存在缺点，这是因为传感器本身限制所决定的，它只能检测单车辆或两车辆之间的空闲停车位，如果车辆之间存在大于一个停车位以上的泊车空间，该传感器很难将每一个泊车位精确的标注出来。超声波传感器原理大致和激光雷达传感原理相同，也是利用声波反射后，通过声波速度以及运行时间得出到目标位置之间的距离^[9-10]。WJ Park 利用多回波函数提高边缘检

测的准确性，接着采用超声波传感器采集不同分辨率情况下空闲泊车位信息^[11]。R Vishnubhotla 通过超声波传感器检测车辆间距离信息，从而提取泊车位信息^[12]。文献[13][14]采用了超声波传感器与 Zigbee 网络结合检测空闲停车位，主要应用场景为室内停车场。超声波传感器用于车位识别的缺点是：首先，它对障碍物的依赖性很大；其次，超声波运行速度与空气温度有关，温度测量的不准确将严重影响距离计算；第三，环境是否混乱，是否有较多杂乱信息^[15]，当超声波传感器接收到这些复杂信号后，会导致系统产生虚假数据。上述两种传感器类型都是利用距离和本身传感器发射速度确定泊车位信息，不能体现泊车位周围环境的具体信息。

近年来基于机器视觉的泊车位检测应运而生，该技术拥有广泛的应用前景和商业价值。机器视觉是对收集到的图像进行自动处理及运算，通过计算机输入指令实现内容识别，即输出我们想要得到的关于图像的信息。图像处理在工业领域的应用主要是依据人类视觉习惯来决定系统输入与输出，并且采用多种手段来满足我们工业工程需要实现的目标^[16]，基于机器视觉这种手段来对泊车位进行识别需要借助 CCD、CMOS 等摄像机原件以及计算机和图像识别等软件实现，该方法可以提取较多并且更加丰富的目标信息（车辆以及周围环境信息、色彩信息、边缘信息、几何信息），这些信息都是二维图像信息，在图像识别问题中可以从多角度进行分析，并且 CCD、CMOS 等元器件相较激光雷达和超声波传感器的造价也更低一点，后期维修费用也比较低，并且在工业识别领域中该方法检测精度较高，鲁棒性强，可以同时检测多个泊车位信息，是一种非常便捷的手段。当然基于视觉的环境感知方法也是会存在一些问题，如容易遭受阴影、光照、杂乱信息的干扰，造成识别效果不理想，在车位线检测问题中，如果车位线比较模糊或较细也会造成检测精度降低，还有一方面是图像处理算法的问题，已有的算法虽然具有一定实时性和准确性，但是对于复杂的图像必须对源函数改进，否则不能应对复杂多变的场景。综上所述目前存在的这些问题也是今后基于机器视觉方法的研究发展方向。

在讨论完自动泊车系统的第一步后，需要对自动泊车策略问题进行研究，主要分为以下两种思路：（1）路径规划：根据泊车位类型的不同、车辆运动学特性、

车辆本身限制以及周围障碍物环境限制等因素规划出一条合适合理的泊车路径，再设计一系列控制算法引导车辆按照之前设计的参考路径泊入车位^[17]。（2）人工智能控制：主要是通过模糊控制或神经网络等方法将具有驾驶员经验行为输入到自动泊车控制器中，不需要预先设定泊车路径，自动控制方向盘转角将目标车辆驶入目标车位。

1.3 自动泊车技术国内外研究与应用现状

1.3.1 自动泊车技术国外研究现状

国外学者最先提出自动泊车的研究，上世纪九十年代，Murray 等人对平行泊车问题展开探讨^[18]，Walsh G 等人提出了非完整系统稳定到轨迹的方法，其中就包括汽车机器人在进行自动泊车操作碰撞因素的阐述^[19]。

1994 年，Laumond 等人研究了一类基于忽略运动约束的无碰撞路径，对合成轨迹进行优化，得到同类中长度接近最小的路径^[20]，尽管这种方法可以完成平行泊车操作，但是对于泊车起始位置要求较高，并且花费时间较多，不满足实时性的要求。为了解决 Laumond 的问题，Paromtchik 等人提出了一种实用的车辆自主平行泊车运动生成与控制方法，采用正弦参考函数将初始泊车位置和目标泊车位置连接，控制车辆在泊车过程中转向角和纵向速度^[21-22]，但是对于不同泊车初始位置都需要规划无碰撞路径是十分繁琐的^[23]。1998 年，Lauquier C 等人将泊车初始位置存储并后台计算，虽然可以节省时间，但还是难免要反复计算^[24]。之后 Paromtchik 研究了自动泊车转向角度和速度的规划控制指令，利用控制命令和路径的结果形状之间的一致性，从环境模型中评估了停车机动所需的路径形状，根据之前获得的外界环境信息规划出泊车路径^[25]。

英国学者 K.Jiang 考虑机器人在矩形空间内平行泊车的一般情况，通过机器人上的超声波传感器检测停车环境，如果符合泊车条件就产生相应的泊车路径，定位阶段，机器人反转到停车位的边缘，避免潜在的碰撞，最后在机动阶段，机器人以统一的模式移动到车位上的停车位置，这需要根据车位的尺寸进行前后制

动^[26]。

新加坡南洋理工大学 J Xu 等人提出了一种基于视觉系统使用的解决方案，同样的视觉系统也可以实现障碍物的检测，之后采用基于五次多项式对路径进行局部优化，完成最终泊车过程^[27]。

佛罗里达大学 Zhao Y 开发并实验验证了一种适用于狭窄车位的鲁棒自动平行泊车算法，为机动过程的每一步设计了新的模糊控制器，采用遗传模糊系统对滑移转向和前轮转向 AGV 的模糊逻辑算法进行优化，提高泊车精度^[28]。

韩国教授 Zang Joung 等人在建立小车运动学模型的基础上，实现对车辆泊车操作中横向和竖向移动的控制，但需要对电机转速和方向转角进行控制，保证车辆沿 X 和 Y 轴方向运动的正确性，使小车的泊车路径正确并实现泊车操作^[29]。

渥太华大学教授 Miah S 利用无线射频识别技术，设计并实现一种具有自动平行泊车能力的类车移动机器人的智能控制方案，控制车辆平行于车位停放^[30]。

2006 年 Ho Gi Jung 提出了一种只利用一台摄像机就完成停车位标线识别的算法，用于自动泊车辅助系统中目标位置的自动选择^[31]。

2010 年 Jeprol 使用图像处理来检测车辆的存在，并提供可用停车位数量和停车位置等信息，该系统利用监控摄像机采集图像，并对图像进行处理以计算可用停车位，并采用改进的软件开发生命周期（SDLC）进行规划、分析、设计、开发和测试^[32]。

2013 年 Yusnita R 提出了一种基于图像分割技术的智能停车位检测系统，该系统能采集并处理停车场的棕色圆形图像，并生成空车停车位的信息^[33]。

2014 年新西兰的 Hilal Al-Kharusi 等学者并没有使用传统直线检测技术来识别源图片中的车位信息，而是采用图像识别技术实现泊车位的检测。该算法原理是通过摄像机提取停车位照片进行车位识别，之所以能够识别到具体停车位是因为车位内部有一个棕色实心圆圈，提取这个圆圈信息以得到车位占用情况^[34]。

2017 年 Nazia Bib 提出了一个基于视觉的智能停车系统框架，帮助驾驶员有效地找到合适的停车位并进行预留，先使用校准将停车场分割成块，然后对每个街区进行分类以识别车辆^[35]。

2019 年 Timo Féret 提出了一种新的基于摄像机的停车位标线检测方法，使

用安装在车辆上的全方位视野摄像机，通过引入标记特定的图像特征和新的滤波，在不使用计算密集鸟瞰变换的情况下，检测针孔摄像机和鱼镜头相机中的停车槽标记^[36]。

2020 年 Zhou 等人提出一种新的基于几何的方法解决路径跟踪算法中存在的误差，该方法建立了 24 个多段模式的路径集，由直线和圆弧组成，采用遍历策略选择路径的模式^[37]。

2021 年 Ohol S.S 等人提出了一种自动垂直泊车策略，通过车身上超声波传感器检测车位环境，在车位满足条件的情况下产生车位位置和机动路径，根据停车位的尺寸进行离合器、制动器等控制，并将该策略集成到自动泊车系统中^[38]。

1.3.2 自动泊车技术国内研究现状

近年来，中国自动泊车技术不断提高，其中包括各个大学学者以及汽车产业的贡献，例如广汽集团、长城汽车、比亚迪等车企的研发，使得自动泊车技术加速应用到日常生活当中。

1999 年吴志坚等学者并没有采用传统的二维模糊控制器，而是采用精度高的多维控制器实现车辆从泊车空间中任意一点泊车入库的问题^[39]。

2005 年杨昔阳等学者在泊车问题中采用模糊控制原理进行仿真，不过并不是采用传统的模糊控制原理，而是采用变论域自适应理论，试验结果表明与传统方法相比，该方法控制灵敏度几乎无超调^[40]。

2009 年宋金泽等人提出将分数阶微分运用到泊车控制系统中，并且设计了分数阶控制器自整定方法，提高泊车控制能力^[41]。同年郭孔辉院士计算了车辆在泊车过程中留有安全间隙下的极限车位尺寸，并总结驾驶员泊车经验，最后结合模糊控制器设计自动泊车控制器，在 Simulink 平台中仿真实现^[42]。

2011 年，郭孔辉院士及其学生分析泊车过程中可能发生的四种碰撞，规划目标车辆泊车路径，同时根据碰撞分析得到泊车起始区域范围，最终实现平行泊车转向策略^[43]。

2012 年浙江大学的周弘等学者对车位识别、路径规划、运动跟踪控制这三

个算法进行研究^[44]。

2013 年李红等学者分析了两种不同的平行泊车形式：平移式和自由双向式，之后进行碰撞分析，建立约束函数，以泊车位最小为目标，求解出最小车位，并在最小车位情况下采用两种方法进行自动泊车^[45]，试验结果表明，自由平行式更优。

2014 年朴昌浩等学者研究的是车辆在初始泊车位置并不是平行状态时的泊车情形，融合车辆动力学模型和运动学模型，最终设计三段式圆弧泊车路径实现平行泊车，结果表明在一定车辆倾斜情况下，该方法可以控制车辆驶入目标车位^[46]。

2015 年任坤等学者通过机器视觉获取位置环境情形下的车位以及车身信息，之后结合车辆动力学模型实际模糊控制器，实现自动泊车控制^[47]。

2018 年熊璐等学者提出了一种对车辆初始泊车位置无要求的自动泊车决策规划算法，针对一次泊车入库不能满足要求的情况，对车辆泊车路径进行多次规划和动态调整算法，在泊车过程中持续更新库位坐标，最终通过仿真实验验证可行性^[48]。

2019 年胡文等学者首先总结熟练驾驶员泊车经验，将泊车路径分为六段，增加路径约束，建立数学模型，设计了一种新的泊车路径规划算法，主要包括泊车、转向、速度控制三部分^[49]。

2020 年魏翠芳针对平行车位轨迹规划问题采用逆向行驶方法对路径进行约束分析，并用采用五次多项式数学模型表示泊车路径轨迹，使得泊车路径更加的平滑^[50]。

2021 年姜武华等学者并没有采用传统的单一传感器方式实现车位检测和识别，而是先使用摄像头传感器收集车位照片，并对车位坐标标定，利用逆透视变换得到车位距离信息，再通过与超声波传感器相结合建立车位线平面模型，之后利用车位坐标平面和模型进行路径规划并仿真^[51]。

1.3.3 自动泊车技术车企应用现状

2003 年，丰田汽车的自动泊车系统将获取到的车位信息在显示屏上显示，供驾驶员进行泊车操作，并且根据方向盘转角的变化实时预测泊车路径，这是最早的泊车辅助功能。

2005 年，雪铁龙利用车辆保险杠上的传感器检测车位周边障碍物信息，再通过数据处理的计算机控制方向盘转角大小，驾驶员控制油门大小以及前后方向实现泊车操作。但是计算量大，实时性差，对于泊车位较小情况不能解决。

2007 年，丰田在雷克萨斯 LS 系列开发了自己的自动泊车系统，该系统融合超声波传感器和摄像头识别车位线，获取到的车位线更加准确，速度更快，驾驶员可以在电子屏幕上选择侧方入库或者倒车入库的形式。

2008 年大众同样是将超声波传感器与摄像头结合，但是该方法不同之处在：泊车过程中不断利用超声波原理测量目标车位与泊车车辆之间的距离，实时更新泊车路径。

2011 年，大众将 APA 系统应用到了途安、夏利车型上，改进了平行泊车方案，减小了最小泊车位尺寸，并且具备垂直泊车和车辆驶出车位功能。

2012、2015 年大众将半自动泊车系统应用到 CC 和凌渡车型上，升级控制系统，使得车辆更加快速、准确泊入目标车位。2017 年，大众高尔夫应用了记忆式自动泊车功能。

2018 年浙江吉利控股集团推出的缤越配备全自动泊车系统，可以满足各种情况下的自动泊车要求，并且全过程解放驾驶员，无需任何操作，该车型已经大致达到 L2 智能技术级别^[52]。

2020 年长城汽车推出的 VV5 汽车，系统会根据泊车位的不同以及车辆相对位置，自动更新规划泊车路径，控制车辆完成泊车操作。

2021 款理想 ONE 新车的自动泊车融合了视觉与感知雷达，能够实现有线和无线车位的自动泊入，且能够对地面上的桩桶、地锁进行识别，新车型在车机页面也增加了一个自动泊车选项^[53]。

综上所述，尽管我国自动泊车技术的研究与开发相较国外仍有一定的差距，

但这更能促进我国在自动泊车技术的研究发展，甚至是之后人工智能的推广，但还是需要我国研究人员自主创新能力的提高，车企间的良性竞争。因此，自动泊车系统的研究与应用相当重要。

1.4 论文主要研究内容

本论文经阅读国内外文献，针对传统自动泊车系统中存在的问题，详细分析了车位线检测、路径规划和路径跟踪这三个主要步骤，在现有研究基础上提出一种基于边缘编组的车位线检测方法，确定三种泊车路径规划策略，同时设计了相应的模糊控制器，最终在软件中进行仿真验证。

全文主要研究内容和技术路线流程图如下：

（1）绪论。介绍本课题研究背景及意义，通过自动泊车技术的国内外研究现状及车企应用现状分析得到目前研究的不足之处，本文中对其中一些问题给出解决方案。

（2）提出基于边缘编组的车位线检测方法。针对传统非机器视觉传感器检测车位线存在的问题，采用 OpenCV 进行图像预处理，其中处理手段包括灰度化、直方图均衡化、二值化、形态学去噪等，目的是降低噪声干扰，提高图像质量。之后改进 Canny 边缘检测算法、对得到的图像边缘进行 8 邻域边缘编组、利用梯度差分概念剔除非车位线特征边缘组等，最终采取概率霍夫变换检测得到车位线。

（3）车辆运动学模型建立及分析。根据车辆几何参数以及泊车特性建立车辆二维运动学坐标系，推导车辆关键点轨迹方程，之后基于阿克曼转向原理建立车辆运动学模型，并对泊车过程中做了稳态分析，便于后续路径规划和路径跟踪控制的研究。

（4）多种泊车路径规划策略。针对平行、垂直和斜式泊车三种不同泊车工况，分别制定两圆弧相切和两步式泊车的路径规划策略，最后确定每种泊车类型的泊车位尺寸大小。

（5）设计自动泊车模糊控制器。概述模糊控制原理和设计方法等，设计了

平行、垂直和斜式泊车三种模糊控制器，并且针对每一种泊车类型，分别设定三种不同初始工况进行仿真，验证之前路径规划的可行性和所设计模糊控制器的可靠性。

(6) 结论。对本文研究内容及研究方法进行总结与展望。

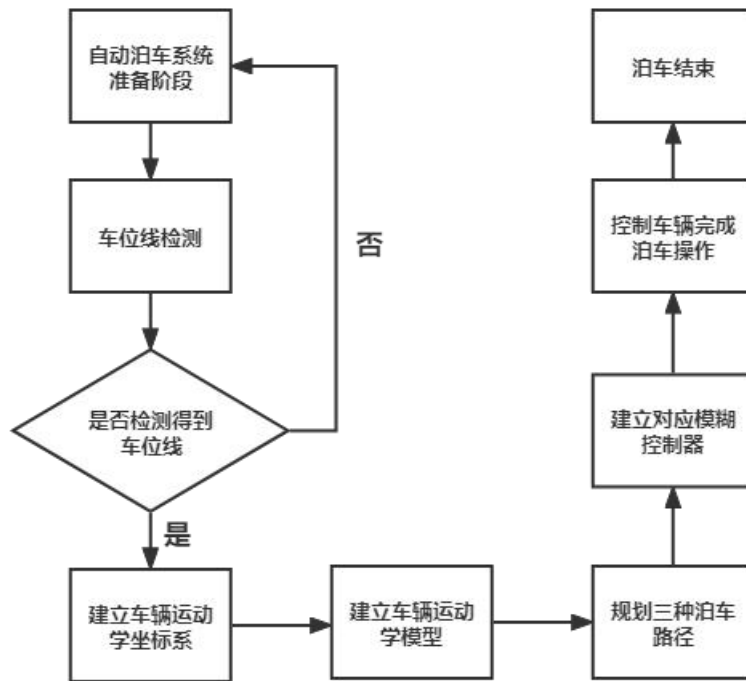


图 1.1 流程图

Fig.1.1 Flow chart

第 2 章 基于边缘编组的车位线检测

自动泊车系统最终目标是车辆自动完成泊车操作，但在这之前首先需要了解车辆周围的泊车环境，即识别出车位线，这是第一步，也是最重要的一步。基于计算机视觉的识别技术是近几年实现车位线检测的新手段，本章采用 OpenCV 和基于泊车位标记特征的检测方法对车位线进行识别，首先对泊车位图片进行图像预处理，之后对经过预处理的图像做边缘检测和 8 邻域边缘编组，然后剔除非车位线边缘特征的边缘组，最终利用概率霍夫变换检测得到车位线。

2.1 OpenCV 概述

2.1.1 计算机视觉

21 世纪社会及经济的快速发展，这相应使得我国科技水平能力大幅提升，加快进入信息化时代，计算机的应用也相对普及大众，利用计算机不仅能够完成日常生活工作等需求，还可以对接到工业领域，弥补人力所不能达到的计算等其他能力，但是这也意味着人们对计算机的要求越来越高，不再局限于过去严格按照计算机使用规则和各种设置程序严格执行命令，人们越来越希望计算机能以人们习惯的方式与人们交换信息，也就是说，让计算机能够看到人们，并与他们交流，倾听和表达你的意见。

计算机视觉是让计算机“复制”人类的视觉感知系统^[54]，换言之，使得计算机能与人眼拥有对外部世界的感知能力，人眼比作计算机的摄像头，人的大脑比作计算机 CPU，摄像头负责捕捉各种场景在图像坐标系中的空间信息，之后计算机可以像人一样对各种场景中的信息进行处理，提取识别出感兴趣区域。这一过程就相当于人类对于不同场景中的信息识别能力。半个世纪以来，计算机视觉技术因为其处理信息快、存储信息量大、易与其他机器学习模块融合等诸多优点得到了广泛的应用，主要应用领域有：

（1）日常生活：车牌识别等；竞技体育项目中启动现场警报、保龄球得分等；品牌管理中创建每个品牌的图像数据库，最终确定新品牌是否与现有图标重复或相似。

(2) 视觉导航：无人机器人自主巡逻和驾驶，降低了危险事故中的人员伤亡的概率，还能对即将到来的危险提供预警提示功能。

(3) 人机交互：核对安检线数以及人员等，主要通过指纹或面部识别等。

(4) 工业视觉：生产线的实时监控等。

2.1.2 OpenCV 开源工具

OpenCV 是基于计算机视觉的开源项目，其中 Open 意为开源，该项目总耗资十亿美元，由美国英特尔公司资助并开发的。OpenCV 的名字包含了计算机打开和查看的含义，用 C++编写的，但还具有其他编程接口（API），如 C#等，具有轻巧、高效的特点，对项目迭代更新以及其他工业领域具有重要意义，图 2.1 为利用 OpenCV 具体应用实例。



图 2.1 OpenCV 应用:(a)车牌识别; (b)人脸识别

Fig.2.1 OpenCV application:

(a)License plate recognition;(b)Face recognition

之所以 OpenCV 目前如此热门^[55]，其重要原因之一就是因为开源思想的引入，这也符合当前社会的共享思潮，开源思想目前是推动软件开发、迭代的关键方式，开源资源正在作为一种新的方法出现，这并不仅仅是针对某一项目的源代码开放性或者可用性，而是能够让广大代码开发人员更精准的针对某一项目，提高工作效率并且实现软件创新，可以预期，今后更多的互联网开发公司会加入到开源趋势当中来。其次，对于项目漏洞的修改和云端维护，开源的生态系统也做出了相应的贡献。总体来说，OpenCV 具有如下优点：

(1) 开源代码 C;

- (2) 基于英特尔处理器优化代码;
- (3) 它具有统一的功能和结构定义;
- (4) 二维数组运算和图像信息处理能力强;
- (5) 灵活的用户界面;
- (6) 可以同时 windows 和 Linux 平台上开发, 特殊情况下还可以将其集成到 DSP 系统中。

2.1.3 泊车位标记法

目前针对泊车位识别手段大致可以分为两类: 基于基础设施和基于车载传感器的解决方案。其中基于基础设施的方法虽然具有管理停车位的特点, 但由于需要在现有的停车场和车辆中安装其他设备, 所以可能无法短时间内应用, 而基于车辆自带传感器感知车位又分为两种: 基于自由空间和基于泊车位标记的方法^[56]。基于自由空间的方法是比较受市场欢迎的, 因为它可以通过各种测距传感器, 例如激光扫描仪、超声波传感器、近程雷达等, 检测两相邻车辆之间的空闲区域来定位停车位, 但是该方法有一个致命的缺点: 当没有邻车辆这一条件时, 则不能找到自由空间, 而基于泊车位标记方法是通过获得包含车位线以及其他信息的图像二维信息的, 通过图像处理等方法知识获取得到相应的目标信息, 并且可以得到多目标信息还具有成本低、可视化等优点。综上所述, 本文将 OpenCV 开源工具箱和基于泊车位标记法相结合实现车位线检测。

2.2 图像预处理

对图像进行车位线检测时, 第一步是要对其做预处理, 因为在真实工况下, 如文献[57]中图 1 (本文图 2.2 中 a) 所示, 除了能获取到车位线标记信息以外, 由于光线条件、背景环境等因素影响, 还会出现很多无用的信息或噪声, 影响车位线检测的准确度, 因此采用图像预处理来保证后续边缘检测的准确性。

2.2.1 灰度化

图像预处理首先是进行灰度化处理, 因为初始图像为彩色图像, 有 R 、 G 、

B 三个分量，若直接用来进行后续操作会加大计算量，并且占用空间较大，而灰度图像的像素只包含一个值，图像的输出数据是传感器针对可见光波长的感应，但颜色的定义是基于人类的感知，颜色空间是颜色坐标系，颜色空间中点的坐标值表示像素的值，颜色空间的选择取决于应用场景，两点之间的颜色距离可能与人类颜色感知的差异不匹配，色彩数据的存储、处理和显示需要恰当的表达，不同的个体之间对颜色的敏感程度也是不一样的，例如人类视觉和机器视觉传感器之间的差异，所以灰度化的方法也是各不相同，要根据实际需求来选择合适的方法，具体方法如下：

(1) 最大值法

选择 R 、 G 、 B 三个分量中数值最大的作为图像的灰度值，如公式 (2-1)，实现效果如图 2.2 (b) 所示。

$$f(i, j) = \max(R(i, j), G(i, j), B(i, j)) \quad (2-1)$$

(2) 平均值法

将 R 、 G 、 B 中这三个分量求取平均值，并将这个值赋予给原图像中对应坐标像素点的像素值，如公式 (2-2)，实现效果如图 2.2 (c) 所示。

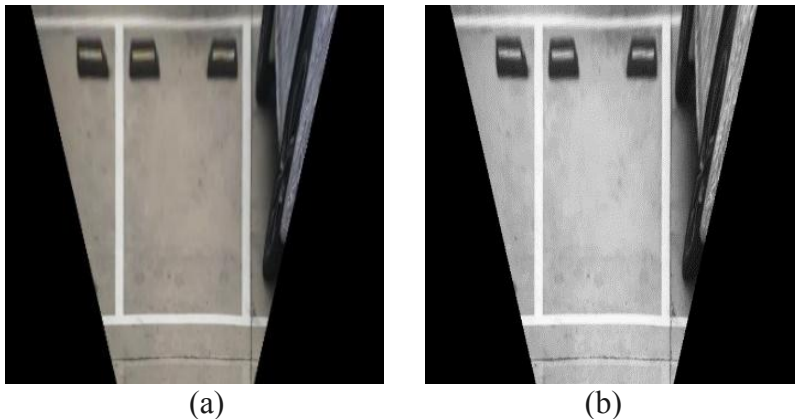
$$f(i, j) = \frac{R(i, j) + G(i, j) + B(i, j)}{3} \quad (2-2)$$

(3) 加权平均法

分别赋予 R 、 G 、 B 三个分量不同的权重，加权平均法公式如 (2-3) 所示，实现效果如图 2.2 (d) 所示。

$$f(i, j) = 0.2989 * R(i, j) + 0.5870 * G(i, j) + 0.1140 * B(i, j) \quad (2-3)$$

考虑到人眼对颜色的敏感程度，本文选择第三种方法来实现图像灰度化。



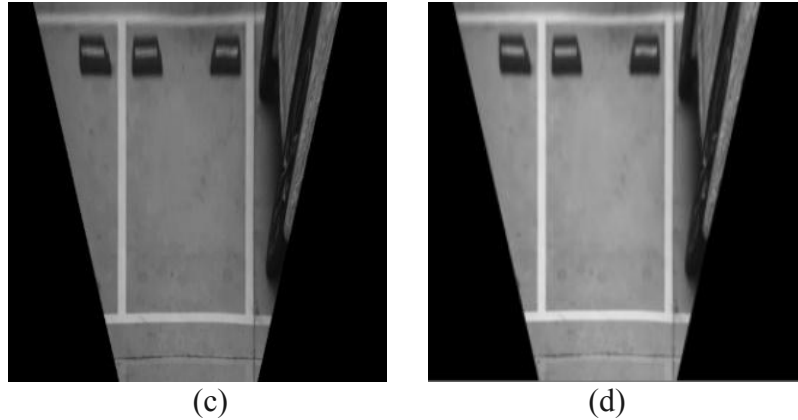


图 2.2 灰度化:(a)初始图像;(b)最大值法;
(c)平均值法;(d)加权平均法

Fig.2.2 Grayscale:
(a)Initial image;(b)The maximum value method;
(c)Average method;(d)Weighted average method

2.2.2 直方图均衡化

图像经过灰度化处理后，仍然包含诸如光线等因素导致的噪声，这些因素会影响识别车位线识别，故需要对已处理过的图像进行增强，这是工程技术中运用计算机视觉图像处理的必要操作，该操作的目的是改变原有灰度图像中各坐标点对应的像素灰度值，并且在此过程中，会根据工程应用所达到的目标对图像关键信息较少部分和噪声因素进行削弱，并增强关键信息细节。虽然图像增强算法相较于边缘检测等算法属于比较基本和简单的，但其显示出来的效果是非常立体和具有艺术性的，经过图像增强操作后的图像能够提高低分辨率图像的质量和可见度，以便计算机进行后续图像预处理操作。

图像增强算法一般分为两种形式：频率域和空间域，划分依据为按照图像处理中直接操作对象的不同，算法包括图像锐化、直方图均衡化等。其中空间域的图像增强是指图像二维坐标系本身，它的工作结果是与图像本身像素点的像素值有关的，例如直方图均衡化算法，而频率域图像增强经过傅里叶变化分析后再逆变换得到最终结果，虽然从表述而言这两种增强方式截然不同，但实际上这两种方法有很大的相似性，用哪种方法是参照图像特征来选择的。

本文选择直方图均衡化的方法来降低噪声的影响，该算法是扩展图像中灰度级个数多的像素点，减少图像中灰度级个数少的像素点^[58]，从而提高对比度，使得图像灰度均衡，增加图像中关键信息点的概率值，降低杂乱信息和噪声点出现

的概率，图 2.3 为图像锐化示意图，图 2.4 为直方图均衡化示意图，尽管这两种算法都可以消除噪声，改善低对比度图像质量，但是明显采用直方图均衡化后算法后的图像更清晰，更能凸显出目标信息。

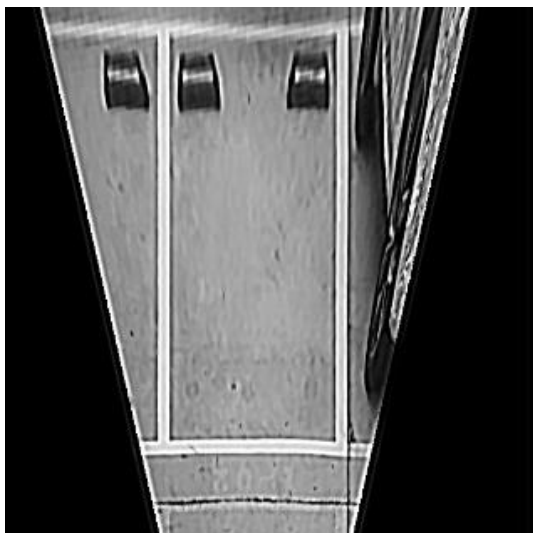


图 2.3 图像锐化
Fig.2.3 Image Sharpening



图 2.4 直方图均衡化
Fig.2.4 Histogram Equalization

2.2.3 二值化

图像二值化原理是对输入图像的每个像素点的像素值进行二分类，通过一个阈值，把图像中所有像素点划为两类，设置值为 0 或 255，那么图像显示效果就为非黑即白，从而使图像变得更简单，待处理的数据量变小，目标的边界变得明显，便于对图像进行更深层的处理，一般来说，二值化可以分为三种方法：

(1) 全局迭代阈值法

全局迭代阈值法第一步是先设定一个初始阈值，该值的确定是利用初始所有像素点中像素值最大和最小值的均值，并使用这个初始阈值将图像像素点的像素值分为两类，接着计算两类中像素点像素值的均值，再求这两个值的平均值，再重新分类，迭代运算，该算法的终止条件为相邻两次计算得到的值之间的差值处于一定范围内，方可结束，该方法一般适用于图像中目标的关键信息与噪声之间灰度值相差很大的情况，具体步骤如下：

1) 选择一个初始阈值 $T = \frac{1}{2}(f_{\max} + f_{\min})$ 。

2) 根据 T 将图像像素点分为 $E1$ 和 $E2$ 两类，这里 $E1$ 内的所有像素点灰度值大

于 T ， $E2$ 内的所有像素点灰度值小于 T 。

3) 分别计算 $E1$ 和其包含区域的灰度均值 μ_1 和 μ_2 。

4) 计算新的阈值 $T = \frac{\mu_1 + \mu_2}{2}$ 。

5) 迭代步骤 2、3、4，当 T_N 与 T_{N+1} 之间的差值小于一定范围时，算法结束。

(2) 最大类间方差法 (Otsu)

Otsu 算法是目前二值化处理中用到最多的一种图像处理方法，这归根于它可以脱离人工确定阈值这一局限性，该算法可以自动根据图像各类灰度值的均值和方差的协方差最大时，得到最佳阈值^[59]，阈值分离后最大化像素之间的差距，因为类之间就是整体图像的差异，该算法在不同背景以及光照条件下自适应选取阈值，算法具体内容如下：设 $f(i, j)$ 为 $M \times N$ 图像中 (i, j) 点的灰度值，取值范围为 $[0, M-1]$ 。设 $p(k)$ 为图像中像素点灰度值为 k 的概率，如公式 (2-4)：

$$p(k) = \frac{1}{MN} \sum_{f(i,j)=k} 1 \quad (2-4)$$

设用阈值取 t 分割出来的目标和背景分别为： $f(i, j) \leq t$ 和 $f(i, j) > t$ ，这样目标比例如公式 (2-5)：

$$\omega_0(t) = \sum_{0 \leq i \leq t} p(i) \quad (2-5)$$

目标部分的点数为

$$N_0(t) = MN \sum_{0 \leq i \leq t} p(i) \quad (2-6)$$

背景部分的比例为

$$\omega_1(t) = \sum_{t \leq i \leq m-1} p(i) \quad (2-7)$$

背景部分的点数为

$$N_1(t) = MN \sum_{t \leq i \leq m-1} p(i) \quad (2-8)$$

目标的均值为

$$\mu_0(t) = \sum_{0 \leq i \leq t} \frac{ip(i)}{\omega_0(t)} \quad (2-9)$$

背景的均值为

$$\mu_1(t) = \sum_{t \leq i \leq m-1} \frac{ip(i)}{\omega_1(t)} \quad (2-10)$$

总的均值为

$$\mu = \omega_0(t)\mu_0(t) + \omega_1(t)\mu_1(t) \quad (2-11)$$

依据 Otsu 方法，最佳阈值的公式见（2-12）：

$$g = \text{Arg} \underset{0 \leq i \leq m-1}{\text{Max}} [\omega_0(t)(\mu_0(t)-\mu)^2 + \omega_1(t)(\mu_1(t)-\mu)^2] \quad (2-12)$$

（3）最优阈值法

由于图像采集手段不同、环境因素不同以及图像本身的影响，有时图像内的关键目标信息的灰度值与某些杂乱信息或不重要信息的灰度值是相同的，若仅靠经验确定一个阈值来实现图像二值化操作的话，检测结果往往会把背景像素误认为是目标信息，故最优阈值法弥补了这个缺陷，在确保图像信息分类错误概率最小的基础上，计算最优阈值，如公式（2-13）：

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad (2-13)$$

式中： P_1 为输入图像中目标信息的概率值， P_2 为输入图像中杂乱信息的概率值。

$$E_1(t) = \int_{-\infty}^t p_2(z) dz \quad (2-14)$$

$$E_2(t) = \int_t^{\infty} p_1(z) dz \quad (2-15)$$

式中： $E_1(t)$ 为杂乱信息被认为是目标信息的概率值， $E_2(t)$ 为目标信息被认为是杂乱信息的概率值。

总的误差公式如（2-16）所示：

$$E(t) = P_2 E_1(t) + P_1 E_2(t) \quad (2-16)$$

使 $E(t)$ 最小即可得到最终的最优阈值，如公式（2-17）：

$$t = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right) \quad (2-17)$$

根据上述几种二值化方法的原理以及特定适应情形，结合车位线检测这种特殊情形，并且为了适应大多数场景下提取的车位线图片，避免某一阈值选取误差过大而导致车位线信息丢失，这里选择 Otsu 算法作为二值化处理手段，如图 2.5 所示。

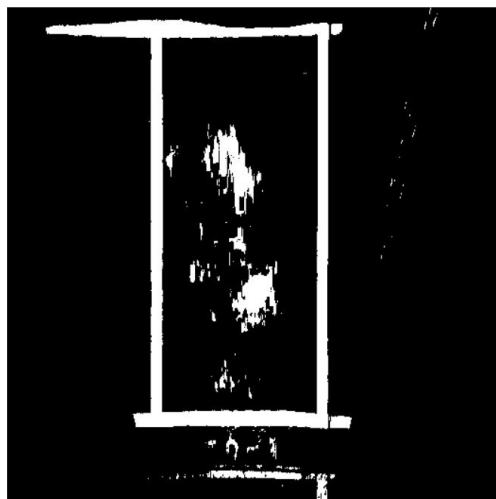


图 2.5 二值化
Fig.2.5 Binarization

2.2.4 形态学去噪

图像形态学去噪方法的基础是数学理论，它起源于代数思想，这项工作使用结构化理论的概念来准确描述物体的几何形状，并利用其分析被处理对象中包含的信息，其中结构化意为“探针”本身具有一定形状，通过“探针”在处理对象图像中移动，最终，提取并测定处理对象图像的形态结构。当然，这里选择的结构元件不仅具有形状的特征，还能添加灰度、方向、颜色等特征，一些构造结构函数元件往往包含以上一种或多种特征，这种特定的“探针”往往用于处理图像中包含复杂且多类别的信息，从而获得不错的处理效果，这就是形态学与数学理论相结合。形态学的操作在图像的预处理和后处理中使用，二值图像的数学形态是形态学理论中重要的部分。下面简单介绍腐蚀、膨胀、开、闭运算四个形态学运算。

2.2.4.1 腐蚀

腐蚀运算从数学形态学的角度来分析，为了得到被获取图像某点或者全局的腐蚀结果，就需要设定具有一定特征的“探针”，利用该“探针”从左往右，从上至下对图像进行滑动，当“探针”遍历完整幅图像后得到新的图像信息，即为腐蚀结果。实际上，该操作是对图像像素起到滤波的效果，去除一些孤立的无意义边界点，增加必要信息所占图像总数的概率，并且如果图像中的两个元素之间

没有什么相关性，且它们之间的距离小于“探针”核大小，是可以通过“探针”将它们完全分开的。腐蚀可以获得全新的图像，其结果是缩小区域，膨胀的结果是扩大区域，想要达到不错的腐蚀效果，还是需要选择合适的“探针”，这样才能最大限度保留原图像中关键信息，并且腐蚀尽可能多的噪声信息，如图 2.6 所示。

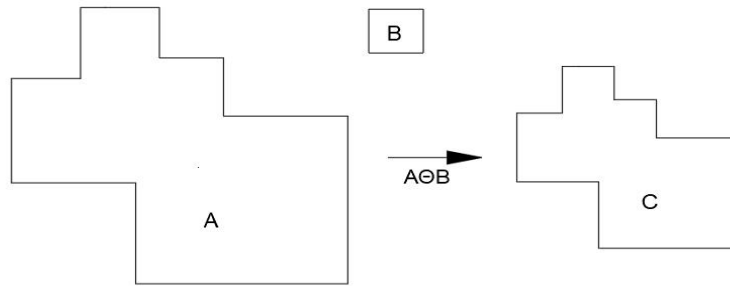


图 2.6 腐蚀原理示意图

Fig.2.6 Schematic diagram of corrosion principle

2.2.4.2 膨胀

膨胀运算与前文介绍的腐蚀运算互为对偶运算，该算法原理还是通过“探针”遍历获取的图像，只不过达到的效果与腐蚀相反，它可以连接图像中分离的像素，将图像中空洞和断裂部分填充。

2.2.4.3 开闭运算

开、闭运算是将上述两种基本形态学运算相结合的算法，其中开运算为先腐蚀后膨胀，闭运算为先膨胀后腐蚀。开运算通常用来去掉灰度值较高的大细节，而相对来说图像的整个灰度平面保持不变，亮度面积较大的像素值保持不变，有必要通过膨胀增强图像的整体，而这样做并不会将之前腐蚀运算已剔除的部分再次添加进去。闭运算与开运算相反，先利用膨胀保证关键信息不被删除，图像整体亮度级增加，然后再利用腐蚀运算，剔除噪声点、边界毛刺、微小结构元素等。

无论是开运算还是闭运算，都能使物体表面光滑，开运算可以消除小突起，与图形中心紧密连接，闭运算会封闭紧间隙，消除小孔，由于开、闭运算均用同一结构元件，故图像大小（尤其图像中主要目标背景大于结构元件时）不受影响，

非常适合在提取泊车位骨架之前进行形态学过滤，以消除缺陷和断点，本文选择开运算作为形态学去噪手段，如图 2.7 所示。

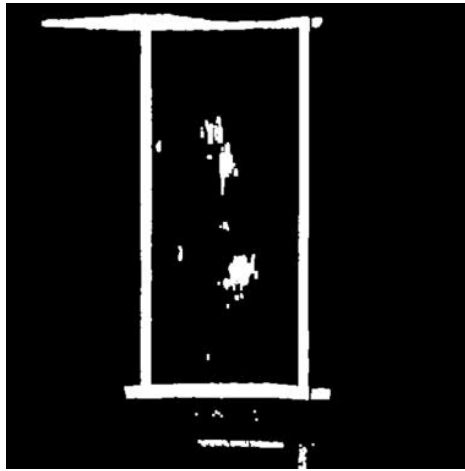


图 2.7 开运算形态学去噪
Fig.2.7 Open operation morphology denoising

2.3 边缘检测

对于车位线检测，如果能在图像中划分出它们的轮廓线，就可以很容易地得到车位的主体部分，图像的边缘包含了很多重要而有意义的信息，图像边缘的识别对于图像分割、特征识别等领域的研究是十分重要的，之所以它这么重要，是因为其包括了图像局部特征（梯度变换、颜色变化、纹理变化等），根据图像中像素点特征检测得到边缘，保留了图像关键信息与结构属性，对于车位线检测而言，泊车位边缘的提取是必要的。

2.3.1 几种边缘检测算法

目前图像处理领域中存在很多类型边缘检测算法，最常用的有四种：罗伯茨算法、Prewitt 算法、索贝尔算法和 Canny 算法，下面一一介绍各算法原理及特点：

（1）罗伯茨算法

罗伯茨算法是一种利用一阶微分来识别边缘的算法，由 Roberts 提出，本质是用局部差分实现边缘检测，在边缘检测算法中最容易实现。

梯度公式如（2-18）、（2-19）所示：

$$f'_x(i, j) = f(i, j) - f(i-1, j-1) \quad (2-18)$$

$$f'_y(i, j) = f(i-1, j) - f(i, j-1) \quad (2-19)$$

梯度幅值如公式（2-20）所示：

$$g_r = \sqrt{f_x'^2 + f_y'^2} \quad (2-20)$$

式中： g_r 表示梯度幅值； f'_x 表示X方向上的灰度变化； f'_y 表示Y方向上的灰度变化。罗伯茨算法由阈值确定最终边缘，阈值是通过计算每个像素的梯度的绝对值，罗伯茨算法不需要另外设置参数，只需要和图像本身有关的四个像素点的像素值，故其优点是计算速度快，有利也有弊，这样的算法原理导致其定位精度不高，对噪声更敏感。

（2）Prewitt 算法

Prewitt 算法是一种基于图像灰度值近似梯度的离散微分算子，主要用于图像边缘识别算法中，当目标图像使用该算法后，对于输入图像中的每个像素，Prewitt 操作符计算每个像素最可能的方向增量以及该方向上的变化率。图像中每个像素的梯度均可认为是一个二维向量，该矢量不仅可以表示像素的梯度变化率，而且可以表示每个像素在该位置处的增量方向，所以也可以认为是二维变量变换。Prewitt 算法利用两个方向的卷积产生灰度的差值，再用差值是否存在极值来判断是否为图像的边缘，两种卷积模板分别在水平方向和垂直方向检测图像，计算公式如（2-21）、（2-22）所示：

$$f_y(i, j) = f(i-1, j+1) + f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - f(i, j-1) - f(i+1, j-1) \quad (2-21)$$

$$f_x(i, j) = f(i+1, j-1) + f(i+1, j) + f(i+1, j+1) - f(i-1, j-1) - f(i-1, j) - f(i-1, j+1) \quad (2-22)$$

Prewitt 算法由于是人为设定阈值，若阈值过大则会忽略小边缘，而且当一张图片中噪声较强时，会误把噪声作为有效边缘而保留下来，所以会存在人为因素影响检测效果，这也是 Prewitt 算法最明显的缺点，故与罗伯茨算法相比，Prewitt 算法的精度要相对较低一些，这也是由于人为设定阈值导致的。

（3）索贝尔算法

上世纪七十年代，Irwin Sobel 第一次提出索贝尔算法，该算法第一步先计算图像像素点的一阶导数，各像素点的一阶导数值是与由梯度模板确定周围邻域像素点像素值有关，根据水平和垂直不同方向的区分，八个像素点之间的数学运算

也是不同的，比如某点作为中心像素点进行计算时，该点的左右上下四个相邻点所占比是最重的，占比系数为 2，这也是索贝尔算法相较于 Prewitt 算法的优点，由于索贝尔算法加入了加权因素，所以提取出的边缘模糊程度较低，效果更好，以水平方向为例，左右两边之间的像素点差值越大，说明图像中该像素点左右两边图像信息变化越剧烈，该点是边缘的可能性越大，反之，变化越缓，是边缘的可能性越小。索贝尔边缘检测算法使用的卷积核大小为 3×3 ，包括水平模板和方向模板，如图 2.8 所示，前者在水平边缘检测中起着最重要的作用，后者在垂直边缘检测中起着最重要的作用。对于一般情形下的索贝尔边缘检测是分别利用两卷积核进行计算，之后利用阈值判断是否作为图像边缘输出。

-1	0	1
-2	0	2
-1	0	1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

图 2.8 梯度模板:(a)水平梯度模板;(b)垂直梯度模板

Fig.2.8 Gradient template:(a)Horizontal gradient template;(b)Vertical gradient template

梯度计算公式如 (2-23)、(2-24) 所示：

$$G = \sqrt{f_x'^2(i, j) + f_y'^2(i, j)} \quad (2-23)$$

$$\theta = \arctan\left(\frac{f_y'}{f_x'}\right) \quad (2-24)$$

式 (2-23) 中 $f_x'(i, j)$ 和 $f_y'(i, j)$ 计算公式如 (2-25)、(2-26) 所示：

$$f_x'(i, j) = f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - 2f(i, j-1) - f(i+1, j-1) \quad (2-25)$$

$$f_y'(i, j) = f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) - f(i+1, j-1) - 2f(i+1, j) - f(i+1, j+1) \quad (2-26)$$

索贝尔算子的 3×3 卷积核可以很好的处理不均匀的图像，但在计算速度方面不如罗伯茨算子快。非光滑图像的噪声对索贝尔算子的影响很小，但是如果更

好的剔除噪声，需要对卷积核进行相应的扩展，但是会增加计算量。该算法通过确定像素的邻灰度加权差在周围方向上是否达到极值来确定像素的位置，其计算方式能有效去除图像中的噪声，但是还是存在一定缺点，其定位精度不是特别高，应用受到很大限制，例如索贝尔算子对彩色图像进行处理时，并不会向人眼视觉一样自动区分出目标图片的前景与背景，得到的边缘在定义和定位精度上都有算欠缺。

(4) Canny 算法

上世纪八十年代，John Canny 提出了 Canny 边缘检测算法，该算法原理大致分为以下四个步骤：

1) 将图像中各像素点输入到高斯滤波器中进行运算，达到平滑图像的效果，公式如 (2-27)、(2-28) 所示：

$$G(x, y) = f(x, y) * H(x, y) \quad (2-27)$$

$$H(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2-28)$$

式中： $G(x, y)$ 为输出图像； $f(x, y)$ 为经过预处理操作后的图像； $H(x, y)$ 为高斯滤波器， σ 为高斯函数的标准差。

2) 计算梯度矢量：

传统 Canny 算法在计算图像中像素点边缘梯度信息是利用相邻像素点之间的一阶偏导差分，公式如 (2-29) 至 (2-32) 所示：

$$G_x = \frac{f(i, j) - f(i+1, j) + f(i, j+1) - f(i+1, j+1)}{2} \quad (2-29)$$

$$G_y = \frac{f(i, j+1) - f(i, j) + f(i+1, j+1) - f(i+1, j)}{2} \quad (2-30)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (2-31)$$

$$\theta = \arctan \frac{G_y}{G_x} \quad (2-32)$$

式中： G_x 为水平方向梯度幅值； G_y 为垂直方向梯度幅值； G 为合成后的梯度幅值； θ 为梯度方向。

3) 非极大值抑制

目的是获取图像边缘的特征位置，它是基于之前得到的梯度矢量信息进行局

部非极大值抑制的，换言之，对于图像像素点中不属于局部最大值的点，将其置零，但是首先判断梯度大小是不是相邻点，然后沿着梯度方向找到另两个相邻点，如果这两个相邻点的振幅大于它们的起始位置，则可以判定该点不在边缘上，即该点的边缘强度为零；反之，这一点可以是边缘点的候选。通过上述操作，可以得到经过非极大值抑制后的局部图像边缘。

4) 双阈值法检测和连接边缘

一般的边缘检测算法都采用单个阈值对梯度等信息进行划分，虽然计算量会变小，但是提取边缘也有一定的缺陷，错误率较高，而且单个阈值选取的值过大或者过小都会产生不同的影响，如果选择过小，则错误地选择许多无用边缘，从而导致后续检测过程困难。如果阈值选择过高，则边缘信息漏检。这个边缘的连续性也带来大量的计算。针对上述现象，Canny 算法放弃了传统的单阈值划分思想，创新性采用双阈值，在目标像素点大于高阈值的情况下，将其认为是图像边缘，小于低阈值时，立即被视为非边缘信息，落入这两个阈值范围中间时，则需要再做判断，只有当该信息点与先前经过阈值筛选且并被却认为是边缘点的点相连，该信息点才会被认为是边缘。

根据以上四种主流边缘检测算法原理，最终选择 Canny 算法作为本文边缘检测算法，因为该算法相较其他算法具有较强的抗噪性，提取出的车位线特征更为细致。

2.3.2 改进的 Canny 边缘检测算法

虽然 Canny 算子的鲁棒性较强，但是还是会受到环境、人为因素、设备等的影响，这样 Canny 算子的问题就体现出来了，主要有以下两个方面：

(1) Canny 算法中在提取图像像素点的梯度幅值与方向中，采用的是 2×2 的模板进行计算的，虽然计算量较小，大致能够提取出图像中的边缘，但总体还是会忽略一些关键信息，并且对错误信息的包容更强，容易将它们添加进图像边缘中。

(2) Canny 算法中采取的是双阈值判定准则，相较于单阈值提高了算法鲁棒性，但是该高、低阈值还是人为设定的，针对车位线检测而言，该背景是复杂多变的、处于不同环境下的，所以需要提高边缘检测算法的自适应能力，满足各工

况需求。

根据以上两个缺陷，本文在传统 Canny 算法的基础上做了以下改进：

(1) 本文采用 3×3 的方向模板替换掉传统的 2×2 方向模板，保留 Sobel 算子中对 X ， Y 方向的计算模板^[60]，还增加 45° 和 135° 这两个方向模板，梯度模板如图 2.9 所示，这样一共四个方向计算边缘幅值和方向， G_x 、 G_y 、 G_{45} 、 G_{135} 分别为像素点在 x ， y ， 45° ， 135° 方向的一阶偏导数值，根据 G_x 、 G_y 、 G_{45} 、 G_{135} 求得 G_x 和 G_y ，再利用 G_x 和 G_y 合成像素点 (i, j) 处的梯度幅值 $G(i, j)$ 和梯度方向 $\theta(i, j)$ ，公式如 (2-33) 至 (2-40) 所示：

$$G_x(i, j) = (f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1) - f(i-1, j-1) - 2f(i-1, j) - f(i-1, j+1)) / 4 \quad (2-33)$$

$$G_y(i, j) = (f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1) - f(i-1, j+1) - 2f(i, j+1) - f(i+1, j+1)) / 4 \quad (2-34)$$

$$G_{45}(i, j) = (2f(i+1, j+1) + f(i+1, j) + f(i, j+1) - 2f(i-1, j-1) - f(i-1, j) - f(i, j-1)) / 4 \quad (2-35)$$

$$G_{135}(i, j) = (2f(i+1, j-1) + f(i+1, j) + f(i, j-1) - 2f(i-1, j+1) - f(i-1, j) - f(i, j+1)) / 4 \quad (2-36)$$

$$G_x(i, j) = G_x(i, j) + \frac{G_{45}(i, j)}{2} - \frac{G_{135}(i, j)}{2} \quad (2-37)$$

$$G_y(i, j) = G_y(i, j) + \frac{G_{45}(i, j)}{2} - \frac{G_{135}(i, j)}{2} \quad (2-38)$$

$$G(i, j) = \sqrt{G_y(i, j)^2 + G_x(i, j)^2} \quad (2-39)$$

$$\theta(i, j) = \arctan\left(\frac{G_y(i, j)}{G_x(i, j)}\right) \quad (2-40)$$

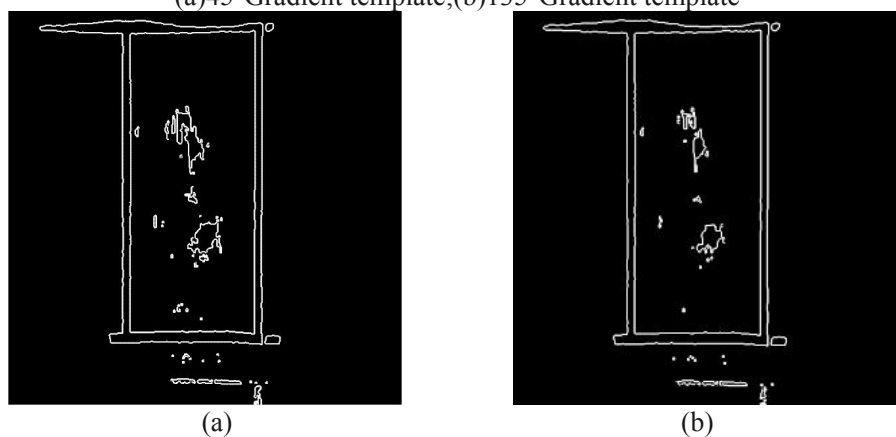
(2) 利用双阈值减少伪边缘数量，采用 Otsu 算法来选取高阈值 T_h ，令低阈值 $T_l = 0.5T_h$ 。

根据以上两点的改进，实现边缘检测，由图 2.10 中可以看出本文算法减少了传统 Canny 边缘检测算法导致的假边缘和噪声的产生，边缘的梯度和幅值更加准确，算法适应性更强。

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

(a) (b)
图 2.9 梯度模板:(a)45°方向模板;(b)135°方向模板
Fig.2.9 Gradient template:
(a)45°Gradient template;(b)135°Gradient template



(a) (b)
图 2.10 边缘检测:
(a)未改进前的边缘图像;(b)改进后的边缘图像
Fig.2.10 Edge detection:
(a)Unimproved edge image;(b)Improved edge image

2.4 车位线检测

通过 2.3 节已经成功把车位线轮廓大致提取出来,但还会存在一些细小噪声以及非车位线特征边缘,若直接对其进行直线检测,处理过程中会受到由噪声产生的短直线对真正车位线的影响,故本节提出基于边缘编组的车位线检测方法,之后利用梯度差分别除非车位线特征边缘组,最后通过概率霍夫变换检测得到车位线。

2.4.1 边缘编组

对边缘图像做 8 邻域搜索并聚类分组,这样每个边缘组都是独立的,单独计算各边缘组直线参数,可以剔除噪声和短线的影响,解决过连接问题。

(1) 遍历边缘图像，搜索顺序从左到右，从上到下，将得到第一个边缘像素点 (x_i, y_i) 作为开头点；

(2) 将开头点 (x_i, y_i) 作为 P_0 ，令 P_0 处灰度值为 0，并搜索 P_0 的 8 邻域中是否存在边缘点，若不存在，返回步骤 (1)，重新寻找开头点。若存在建立边缘组 H_m ，将 (x_i, y_i) 存入 $H_m = \{(x_i, y_i)\}$ ，其中 m 代表第 m 边缘；

(3) 将 P_0 处 8 邻域内存在的边缘点存入 $H_m = \{(x_i, y_i), (x_{i+1}, y_{i+1}), \dots, (x_{i+n}, y_{i+n})\}$ ，从而更新边缘组，并且将这些边缘点处灰度值置为 0，使这些点不被再次扫描，同时建立一个空的边缘组 $G_m = \{(x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}), \dots, (x_{i+n}, y_{i+n})\}$ ，用于记录除 P_0 外新增加边缘点；

(4) 把 G_m 中最后一点作为 P_0 继续遍历，之后将该点从 G_m 中删除，并且更新 $G_m = \{(x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}), \dots, (x_{i+n-1}, y_{i+n-1})\}$ ，若该点 8 邻域内存在边缘点，则转到步骤 (3)，若不存在，继续执行本步骤，直至 G_m 为空；

(5) 重复步骤 (1) 到 (4)，直至该图像没有再发现边缘点；

(6) 设定阈值 T_2 与每个边缘组像素数目比较，剔除小于该阈值的边缘组，从而去除短直线和独立像素点等信息；

经过边缘编组后的图像如图 2.11 所示。

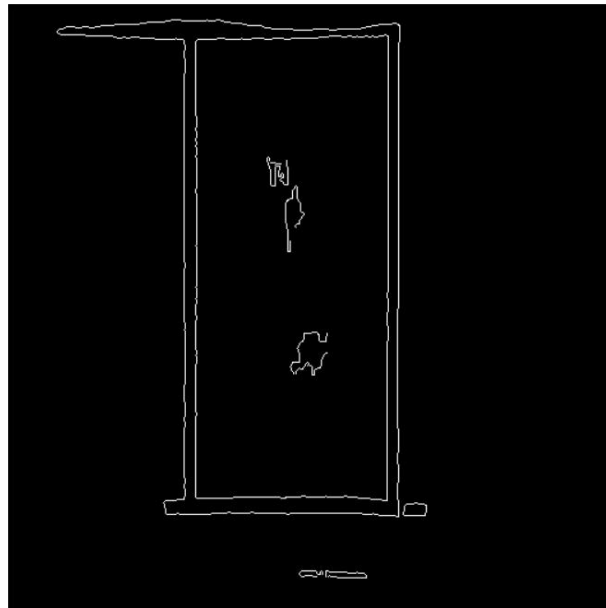


图 2.11 边缘编组
Fig.2.11 Edge grouping

2.4.2 剔除非车位线特征边缘组

经过边缘分组后，各边缘组内均为 8 邻域内互相连通的边缘像素点，但是由于环境等因素，其中还是会存在非车位线边缘组，如图 2.11 所示，这其中一部分原因是因为当获取到的图片中车位线相对较细时，之前形态学操作中的核不能很大，如果较大，虽然可以剔除噪声边缘，但是本身车位线边缘信息也会被删除，如图 2.12 所示。所以当形态学操作中选取较小的核运算时，可以最大程度保留车位线信息，但不可避免也会存在一些杂乱信息，如图 2.13 所示。本文利用图像中杂乱信息边缘组内各点梯度方向不一致和车位线边缘组内各点梯度方向相同这一特征，用各边缘组内两像素点梯度方向差分概念剔除非车位线特征边缘，最终剔除非车位线特征后的所剩边缘组效果如图 2.14 所示，算法基本步骤如下：

(1) 任选取边缘组 H_m ；

(2) 梯度方向 $\theta(i, j)$ 利用上文 Canny 算子计算得到，任意边缘组 $H_m \{(x_i, y_i)\} (i=1, 2, \dots, n)$ 中相邻像素点梯度差分 ΔU 公式定义为 $\Delta U = |\theta(x_{i+1}, y_{i+1}) - \theta(x_i, y_i)| (i=1, 2, \dots, n-1)$ ；

(3) 设定计数器 N_H 并将初始值赋为 0；设定计数器定值 N_{LH} ；设定梯度方向差分阈值 θ_H ；

(4) 若 $\Delta U < \theta_H$ ，则累加器 $N_H = N_H + 1$ ；

(5) 若 $N_H > N_{LH}$ ，则该边缘组存在车位线边缘；若不成立，则该将该边缘组像素点灰度值全部置零并返回步骤 (1)，直至所有边缘组检测完成；

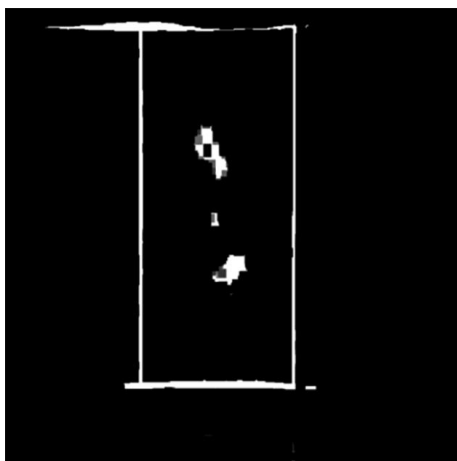


图 2.12 核为 4 时
Fig.2.12 When the core is four

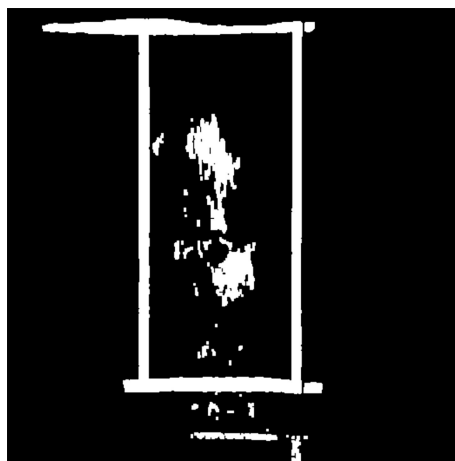


图 2.13 核为 2 时
Fig.2.13 When the core is two

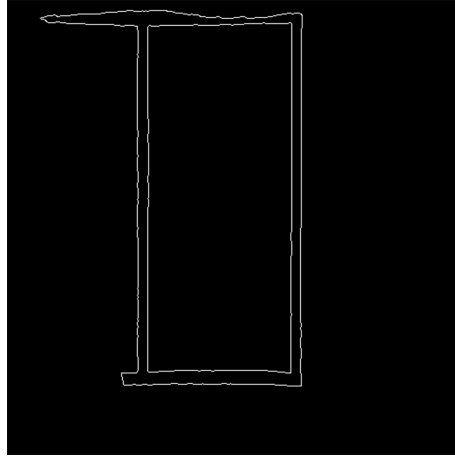


图 2.14 剔除非车位线特征后效果图

Fig.2.14 Effect picture after excluding non-parking space line features

2.4.3 霍夫变换方法选择

霍夫变换分为霍夫直线检测和霍夫圆检测，在车位线检测中常被用于直线检测，该变换的本质是两个坐标系之间的转换，并且在转换过程中还涵盖着整体特性问题转变为局部特征问题，换言之，当遇到无法便捷处理的情形时，需要转换到极坐标系下进行解决，并且在转换完毕后，在该极坐标系下的特征被采集出来，找到峰值即可得到最终结果。通过霍夫变换原理我们可以得知，如果在直角坐标系的参数逻辑中，普通直线是很容易检测出来的，但是对于某些特殊情况，例如 $y=0,1,\dots,n$ 等这类型直线，是不好进行峰值检测的。因此，对于这一局限，上世纪七十年代，Dudia 和 Hart 将直角坐标系转换为极坐标形式，公式如（2-41）所示：

$$\rho = x \cos \theta + y \sin \theta \quad (2-41)$$

由式（2-41）中可知，新的坐标形式下一组坐标对应的是原坐标形式下的一条线，车位线检测中的一幅图像中不可能只有一条直线，那么一条直线由于多方面原因会有相同斜率的多条直线，所以选择他们之中在新坐标形式下 (ρ, θ) 数目最多的点，则对应到原坐标形式下，即为最多的直线，其内部关系就是线和点一对多的映射法则。但是传统霍夫变换中有两点明显的缺点：

- （1）每个非零像素点都会在投票过程中累加；
- （2）图像上的某一像素点可能映射到参数空间中多个累加器中，造成伪峰值。

针对以上提出的缺点，本文选择概率霍夫变换方法来检测最终车位线^[61]，算法步骤具体如下：

- (1) 整理经过非车位线特征剔除后剩余边缘组 $H_K (K=1,2,\dots,n)$ ，初始化参数空间 P ，令 $k_i (i=1,2,\dots,n)=0$ ；
- (2) 从 H_K 中随机选取点 (x_i, y_i) 代入公式 (2-41) 求得 $P_i = (\rho_i, \theta_i)$ ；
- (3) 在参数空间 P 中寻找 P_i 是否存在，若存在，则 P_i 的累加器加 1；若不存在，则把 P_i 插入 P 中，且迭代步骤 $k_i = k_i + 1$ ；
- (4) 当 P 中某一直线对应的累加器 k_i 大于一定阈值，结束循环，删除在该直线上的所有点，接着对剩余点进行步骤 (2)，直至该边缘组为空，返回步骤 (1)，对下一边缘组继续检测，完成所有边缘组检测；若没有则返回步骤 (2) 继续循环。

2.5 仿真验证

为验证基于边缘编组的车位线检测的有效性和实时性，对车位线图片分别应用传统霍夫变换和本文算法检测车位线来对比结果，本文算法利用 OpenCV4.2.0 库在 Visual Studio 2015 开发环境中实现，计算机 CPU 为 Core i5 10300H，主频为 2.50GHz，内存 16G。其中 $T_2 = 40$ ， $N_{LH} = 100$ ， $\theta_{LH} = 0.08$ ，传统霍夫变换中 ρ 、 θ 间隔分别为 1、 0.01° ， σ 为高斯函数的标准差，取 0.5。检测结果如图 2.15 所示。

图 2.15 中(a)为灰度图，(b)为传统霍夫检测后的车位线图，(c)为本文算法检测后所剩边缘组图像，(d)为最终经过概率霍夫变换后标定出的车位线图像，当图像(a)中存在一定畸变时，传统霍夫变换并不能够准确检测到车位线，并且会出现漏检情况，而本文算法通过边缘分组解决过连接问题和利用梯度差分概念去除非车位线特征边缘组，最后做概率霍夫变换检测车位线。

为验证本文算法的实用性，另外添加另一组工况作为验证，如文献[62]中图 5（本文图 2.16 中 a）所示，实验组二中传统检测方法不仅检测出较多的杂乱直线，还会将噪声点和边缘点检测为同一直线，但本文提出的检测算法可以得到目标图像中的车位线。

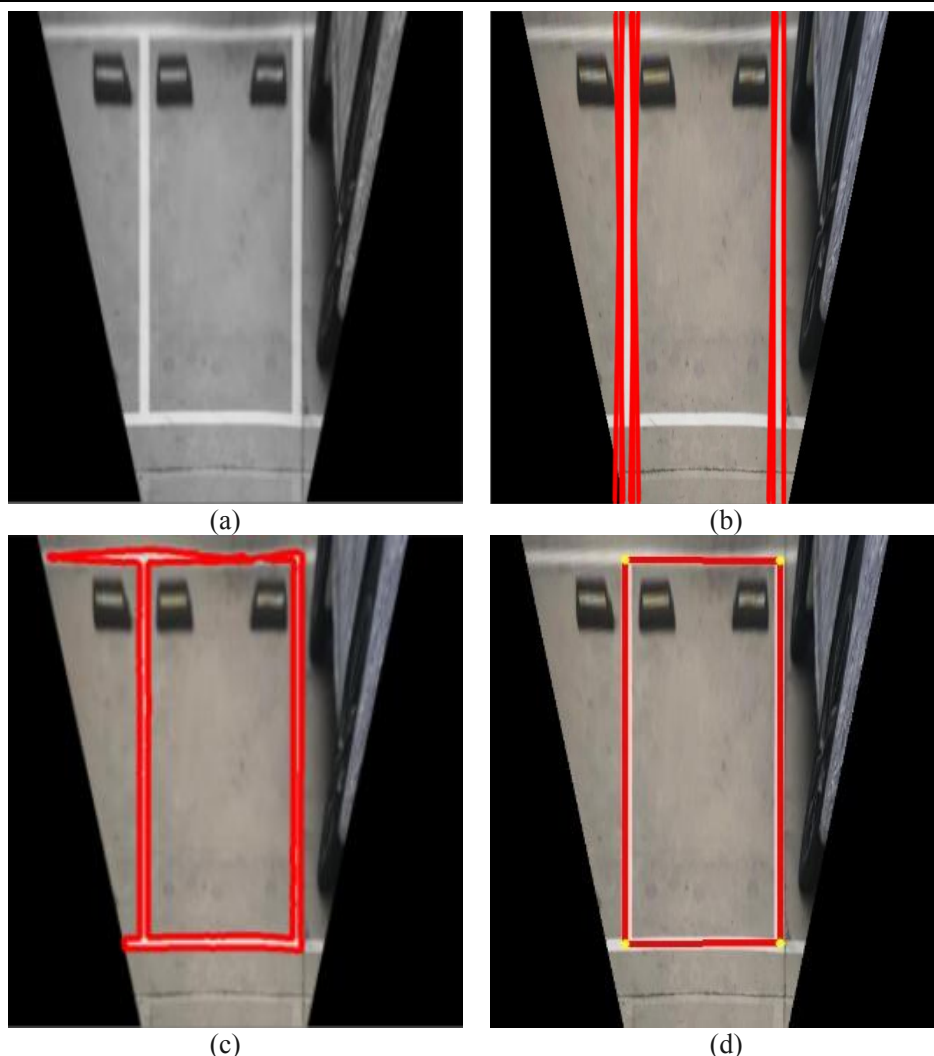
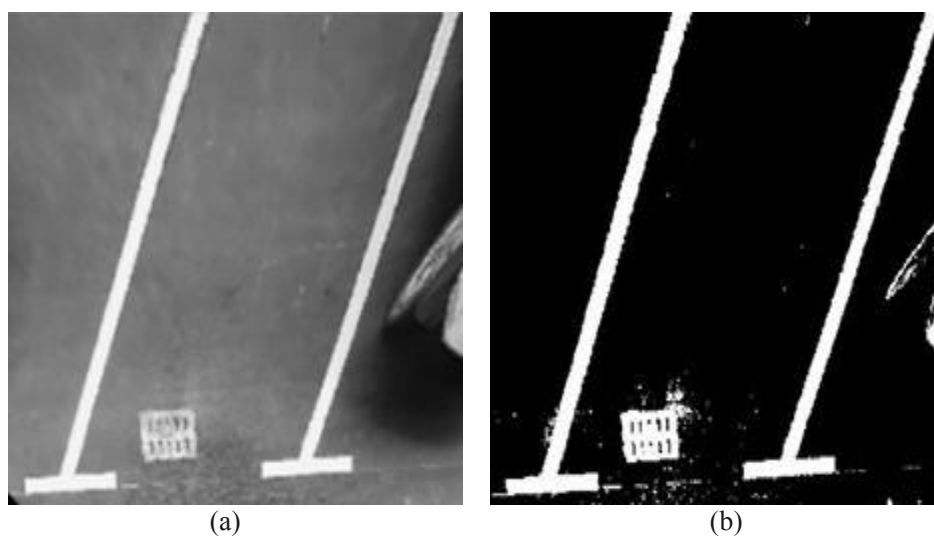


图 2.15 实验组一:(a)灰度图;(b)传统霍夫检测
(c)剔除非车位线特征; (d)最终标定车位线

Fig.2.15 Experimental group one:
(a)Grayscale image;(b)Traditional Hough detection;
(c)Exclude non-parking space line features;(d)Final calibration of the parking space line



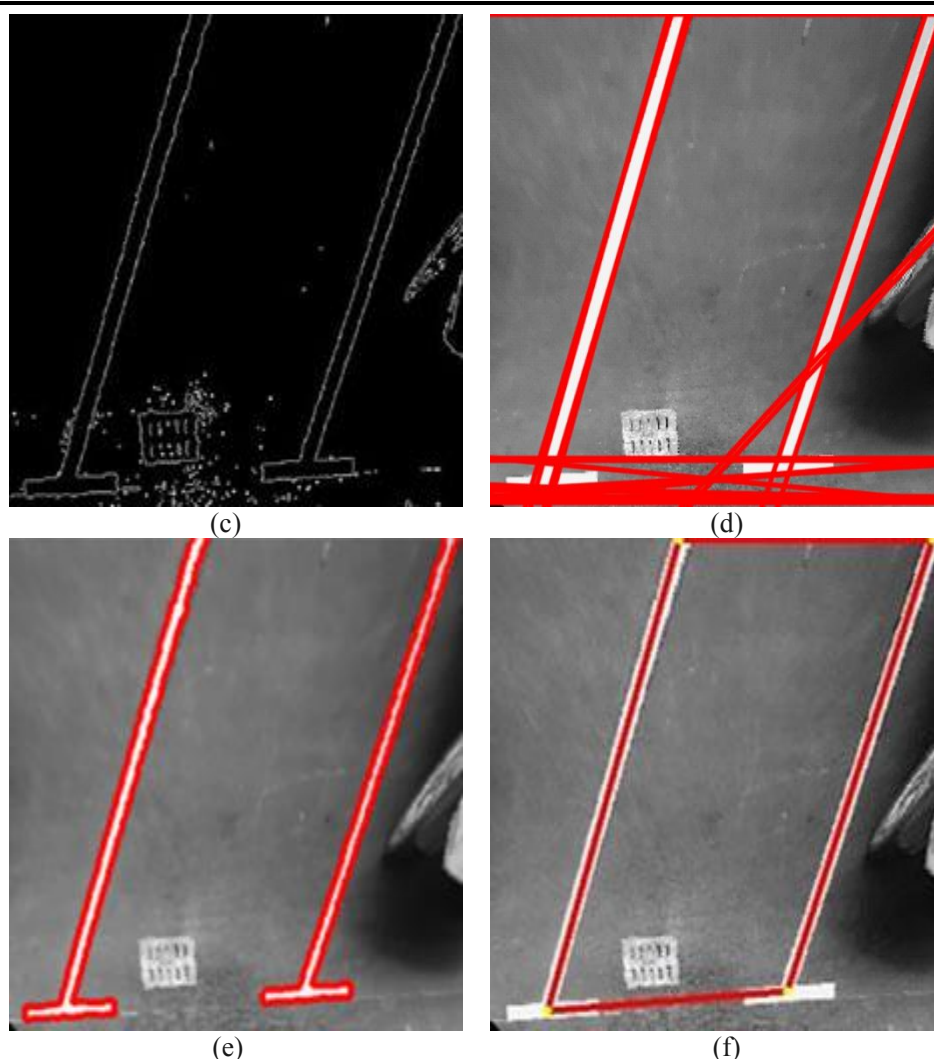


图 2.16 实验组二:(a)灰度图;(b)二值化;

(c)边缘检测;(d)传统霍夫检测;

(e)剔除非车位线特征;(f)最终标定车位线

Fig.2.16 Experimental Group Two:(a)Grayscale image;(b)Binarization;

(c)Edge detection;(d)Traditional Hough detection;

(e)Exclude non-parking space line features;(f)Final calibration of the parking space line

为验证本文算法的运行效率以及鲁棒性，实验 30 次取平均所需时间，算法时间为边缘编组后到检测车位线所需时间，标准霍夫时间为边缘检测后直接运用传统霍夫变换检测车位线所需时间，并判定 30 次实验结果是否正确，由表 2.1 中数据可知，本文算法对于车位线检测具有较好的鲁棒性和实时性。

表 2.1 两种方法运行效率实验对比

Table 2.1 Comparison of the operating efficiency of the two methods

实验组	标准霍夫时间 t1/s	本文算法时间 t2/s	传统霍夫 正确次数	本文算法 正确次数
实验组一	0.104	0.074	7	29
实验组二	0.159	0.101	22	29

2.6 本章小结

本章先介绍了计算机视觉理论和 OpenCV 开源计算机视觉项目及其优点，针对基于自由空间法的缺点，确定采用 OpenCV 开源工具箱和基于泊车位标记法检测车位线；接下来对泊车位图像进行预处理操作，包括灰度化、直方图均衡化、二值化和形态学去噪等运算；之后分析对比了几种常用边缘检测算法，确定采用 Canny 算法并对其做了改进；接着对图像边缘进行 8 邻域边缘编组，得到独立的互相连接的边缘组，削减短线噪声影响，而后对剩余边缘组内相邻像素点做梯度差分，剔除非车位线特征边缘；最后采用概率霍夫变换检测得到车位线，结果表明本文算法与传统霍夫变换检测车位线相比，克服了漏检情况，提高了车位线检测效率。

第3章 车辆运动学模型建立及分析

在完成车位线检测的基础上，自动泊车系统接下来需要对车辆内外部参数进行确定，之后建立车辆运动学坐标系对泊车过程进行描述，最后根据阿克曼转向原理建立车辆运动学模型并在 Simulink 中搭建。

3.1 泊车参数描述

自动泊车系统控制车辆进行泊车操作时，除了周围环境等参数会对此过程造成影响，车辆自身参数也会存在约束，本文为了便于后期车辆运动学模型建立等，这里描述了泊车过程中需要用到车辆自身的参数和外部参数，如表 3.1 所示。

表 3.1 车辆参数列表
Table 3.1 List of vehicle parameters

参数	定义	数值(mm)
车长	纵向最前端到最后端长度	3990
车宽	横向最左端到最右端长度	1695
轴距	前轴中心到后轴中心长度	2490
前悬	最前端与前轴中心长度	662
后悬	最后端与后轴中心长度	771
前轮距	前轮两轮与地接触点长度	1475
后轮距	后轮两轮与地接触点长度	1465
最小转弯半径	方向盘打到极限位置，车辆外侧转向轮的中心面的支承点划过的圆轨迹的半径	4700

泊车外部因素如下图 3.1 所示：

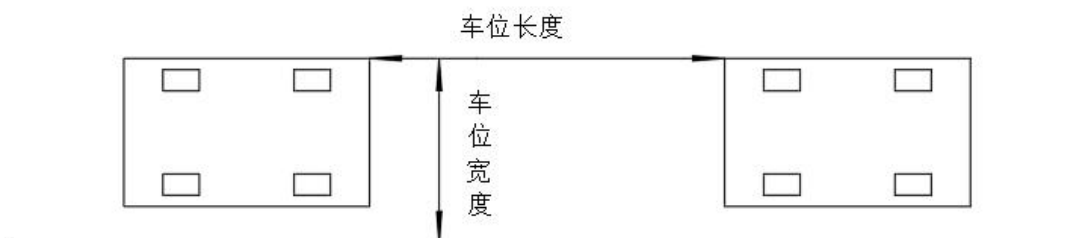


图 3.1 车辆外部因素示意图
Fig.3.1 Schematic diagram of vehicle external factors

- (1) 车位长度：前后方障碍物之间的横向距离。
- (2) 车位宽度：最外侧障碍物到内边界线之间的纵向距离。

3.2 车辆运动坐标系

要分析车辆泊车运动过程，首先需要建立车辆运动坐标系，得到车辆顶点、前后轴中心点、轮胎接地点之间的几何关系，但是在这之前需要先进行假设，一般来说车辆在泊车过程中都是低速行驶，故假设轮胎在转向过程中只做纯转动，忽略侧偏以及打滑等因素，将泊车过程简化为刚体运动，并且为了便于分析，将坐标系原点设定在目标泊车位左下角顶点处，以目标泊车位右边线作为坐标系 X 轴，垂直于目标泊车位边线为 Y 轴。如图 3.2 所示建立车辆运动坐标系，其中 a 、 b 、 c 、 d 分别为目标车辆轮胎与地面接地点， f 为目标车辆前轴中心点投影到地面的交点， g 为目标车辆中心点投影到地面的交点， r 为目标车辆后轴中心点投影到地面的交点， A 、 B 、 C 、 D 分别为目标车辆顶点投影到地面的交点， A' 、 B' 、 C' 、 D' 分别为目标泊车位顶点。

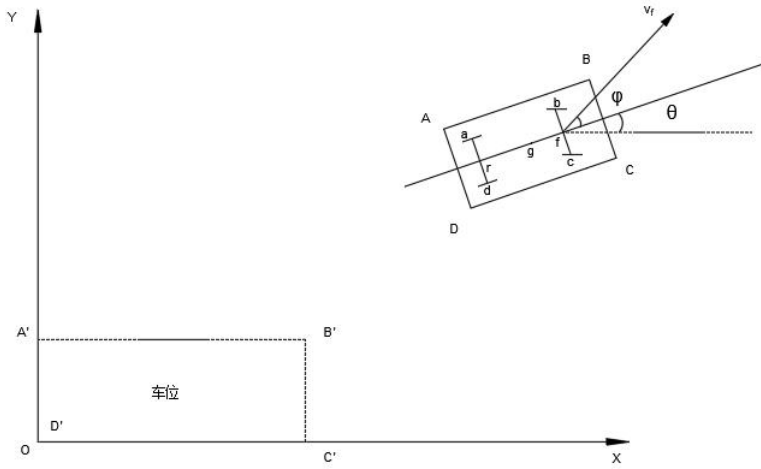


图 3.2 车辆运动学坐标系图

Fig.3.2 Vehicle kinematics coordinate system diagram

如图 3.2 中所表示的数学关系，车辆前轴中心点坐标 (x_f, y_f) 和后轴中心点坐标 (x_r, y_r) 如式 (3-1)，(3-2) 所示：

$$\begin{cases} x_f = x + \frac{l}{2} \cos \theta \\ y_f = y + \frac{l}{2} \sin \theta \end{cases} \quad (3-1)$$

$$\begin{cases} x_r = x - \frac{l}{2} \cos \theta \\ y_r = y - \frac{l}{2} \sin \theta \end{cases} \quad (3-2)$$

车辆四个轮胎接地点 $(x_a, y_a), (x_b, y_b), (x_c, y_c), (x_d, y_d)$ 坐标见式 (3-3) 至 (3-6) 所示:

$$\begin{cases} x_a = x - \frac{l}{2} \cos \theta - \frac{L_{rw}}{2} \sin \theta \\ y_a = y - \frac{l}{2} \sin \theta + \frac{L_{rw}}{2} \cos \theta \end{cases} \quad (3-3)$$

$$\begin{cases} x_b = x + \frac{l}{2} \cos \theta - \frac{L_{fw}}{2} \sin \theta \\ y_b = y + \frac{l}{2} \sin \theta + \frac{L_{fw}}{2} \cos \theta \end{cases} \quad (3-4)$$

$$\begin{cases} x_c = x + \frac{l}{2} \cos \theta + \frac{L_{fw}}{2} \sin \theta \\ y_c = y + \frac{l}{2} \sin \theta - \frac{L_{fw}}{2} \cos \theta \end{cases} \quad (3-5)$$

$$\begin{cases} x_d = x - \frac{l}{2} \cos \theta + \frac{L_{rw}}{2} \sin \theta \\ y_d = y - \frac{l}{2} \sin \theta - \frac{L_{rw}}{2} \cos \theta \end{cases} \quad (3-6)$$

同理, 车辆四个顶点 $(x_A, y_A), (x_B, y_B), (x_C, y_C), (x_D, y_D)$ 坐标见式 (3-7) 至 (3-10) 所示:

$$\begin{cases} x_A = x - \frac{L}{2} \cos \theta - \frac{W}{2} \sin \theta \\ y_A = y - \frac{L}{2} \sin \theta + \frac{W}{2} \cos \theta \end{cases} \quad (3-7)$$

$$\begin{cases} x_B = x + \frac{L}{2} \cos \theta - \frac{W}{2} \sin \theta \\ y_B = y + \frac{L}{2} \sin \theta + \frac{W}{2} \cos \theta \end{cases} \quad (3-8)$$

$$\begin{cases} x_C = x + \frac{L}{2} \cos \theta + \frac{W}{2} \sin \theta \\ y_C = y + \frac{L}{2} \sin \theta - \frac{W}{2} \cos \theta \end{cases} \quad (3-9)$$

$$\begin{cases} x_D = x - \frac{L}{2} \cos \theta + \frac{W}{2} \sin \theta \\ y_D = y - \frac{L}{2} \sin \theta - \frac{W}{2} \cos \theta \end{cases} \quad (3-10)$$

式中： L_{fw} 为车辆前轮轮距， L_{rw} 为车辆后轮轮距， φ 为 Ackerman 转向角；
 θ 为汽车车身方向角； L 为车长； W 为车宽； l 为轴距； v_f 为车速； (x, y) 为汽车中心点 g 于地面的投影坐标；

3.3 阿克曼转向原理

在建立车辆运动学模型之前，首先要求得表征车辆泊车过程中转向状态的重要变量即车辆前轴处等效转角，但这个等效转角是无法实际测量得到的，所以要结合阿克曼转向几何原理等效代替车辆前轴处等效转角，如图 3.3 所示。

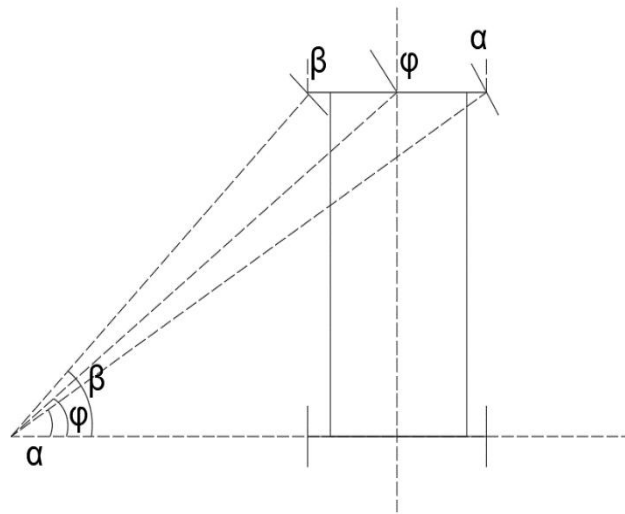


图 3.3 阿克曼转向图
 Fig.3.3 Ackerman steering diagram

由阿克曼转向原理有：

$$\cot(\alpha) = \cot(\beta) + \frac{H}{l} \quad (3-11)$$

$$\varphi = \frac{\alpha + \beta}{2} \quad (3-12)$$

式中， α 、 β 分别为车辆外、内轮转向角； H 为两主销中心距；实际上对于自动泊车系统以及车辆运动学模型的建立肯定是精度越高，泊车效果越好，用阿

阿克曼转向原理获取到的等效转角是符合要求的，并不会降低系统精度，因为泊车过程中泊车速度很小，车辆的方向盘转角 δ 与前轴中心点等效转角，即阿克曼转角之间关系是线性的，公式如（3-13）所示：

$$\delta = k\varphi \quad (3-13)$$

3.4 车辆运动学模型

3.4.1 低速泊车运动学模型

结合之前分析得到的结果建立坐标系，其中原点为 r ，即车辆后轴中心点，进而分析车辆运动学模型，如下图 3.4 所示：

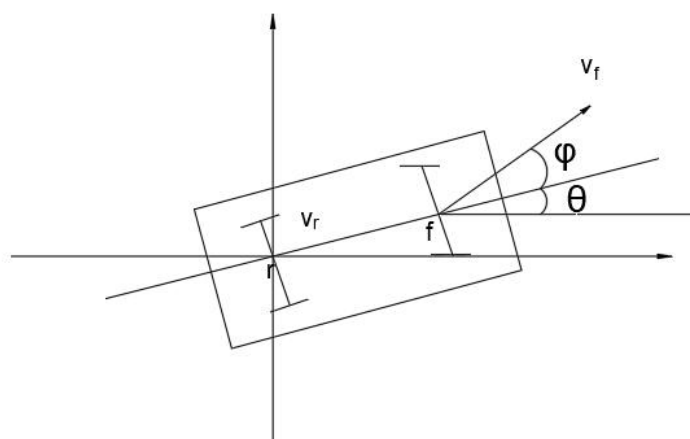


图 3.4 车辆运动学模型

Fig.3.4 Vehicle kinematics model

车辆后轴中心点 r 处方程：

$$\dot{x}_r \sin \theta - \dot{y}_r \cos \theta = 0 \quad (3-14)$$

车辆前轴中心点 f 处关系式为：

$$\begin{cases} \dot{x}_f = -v_f (\cos \theta \cos \varphi - \sin \theta \sin \varphi) = -v_f \cos(\theta + \varphi) \\ \dot{y}_f = -v_f (\sin \theta \cos \varphi + \cos \theta \sin \varphi) = -v_f \sin(\theta + \varphi) \end{cases} \quad (3-15)$$

由上式可得：

$$\begin{cases} x_f = \int -v_f \cos(\theta + \varphi) dt \\ y_f = \int -v_f \sin(\theta + \varphi) dt \end{cases} \quad (3-16)$$

根据上式可得 $f(x_f, y_f)$, $r(x_r, y_r)$ 之间的关系式为:

$$\begin{cases} x_r = x_f - l \cos \theta \\ y_r = y_f - l \sin \theta \end{cases} \quad (3-17)$$

由上式可得车辆后轴中心点处 \dot{x}_r 、 \dot{y}_r 的关系式为:

$$\begin{cases} \dot{x}_r = \dot{x}_f + l\dot{\theta} \cos \theta \\ \dot{y}_r = \dot{y}_f - l\dot{\theta} \sin \theta \end{cases} \quad (3-18)$$

将 (3-18) 代入 (3-14) 有

$$\dot{x}_f \sin \theta - \dot{y}_f \cos \theta + \dot{\theta} l = 0 \quad (3-19)$$

将 (3-15) 代入 (3-19) 得:

$$\dot{\theta} = -\frac{v_f \sin \varphi}{L} \quad (3-20)$$

3.4.2 泊车运动稳态分析

当车辆在理想状态下以稳态转弯半径泊车时, 可以确定车辆运动轨迹, 将式 (3-13), (3-17) 代入式 (3-15) 可得后轴中心点 r 速度有:

$$\begin{cases} \dot{x}_r = -v_f \cos \theta \cos \varphi \\ \dot{y}_r = -v_f \sin \theta \cos \varphi \end{cases} \quad (3-21)$$

对式 (3-21) 做积分变换可得 r 处坐标关系表达式:

$$\begin{cases} x_r(t) = \int -v_f \cos \theta \cos \varphi dt = -l \cot \varphi \sin\left(\frac{v_f \sin \varphi}{l} t\right) \\ y_r(t) = \int -v_f \sin \theta \cos \varphi dt = l \cot \varphi \cos\left(\frac{v_f \sin \varphi}{l} t\right) + l \cot \varphi \end{cases} \quad (3-22)$$

将式 (3-22) 中两个方程分别平方再相加有,

$$x_r^2 + (y_r - l \cot \varphi)^2 = (l \cot \varphi)^2 \quad (3-23)$$

同理 (x_a, y_a) , (x_d, y_d) 可表示为:

$$\begin{cases} x_a(t) = (l \cot \varphi - \frac{L_{rw}}{2}) \sin\left(\frac{v_f \sin \varphi}{l} t\right) \\ y_a(t) = -(l \cot \varphi - \frac{L_{rw}}{2}) \cos\left(\frac{v_f \sin \varphi}{l} t\right) + l \cot \varphi \end{cases} \quad (3-24)$$

$$\begin{cases} x_d(t) = (l \cot \varphi + \frac{L_{rw}}{2}) \sin(\frac{v_f \sin \varphi}{l} t) \\ y_d(t) = (l \cot \varphi + \frac{L_{rw}}{2}) \cos(\frac{v_f \sin \varphi}{l} t) + l \cot \varphi \end{cases} \quad (3-25)$$

将公式 (3-24)、(3-25) 简化得到车辆前、后轮的方程有，

$$x_a^2 + (y_a - l \cot \varphi)^2 = (l \cot \varphi - \frac{L_{rw}}{2})^2 \quad (3-26)$$

$$x_d^2 + (y_d - l \cot \varphi)^2 = (l \cot \varphi + \frac{L_{rw}}{2})^2 \quad (3-27)$$

由公式 (3-26)、(3-27) 可知，由上文车辆前后轮关系表达式可知，当自动泊车系统进行泊车规划时，固定方向盘在某一稳定转角时，车辆运动轨迹只与阿克曼转角、轴距、轮距这些因素有关，车辆泊车过程的路径为稳态圆周，泊车速度是不会影响泊车轨迹的，只是会对车辆实际操作有影响。为了后续自动泊车路径跟踪的便捷，现在 Simulink 中建立车辆运动学模型，如图 3.5 所示：

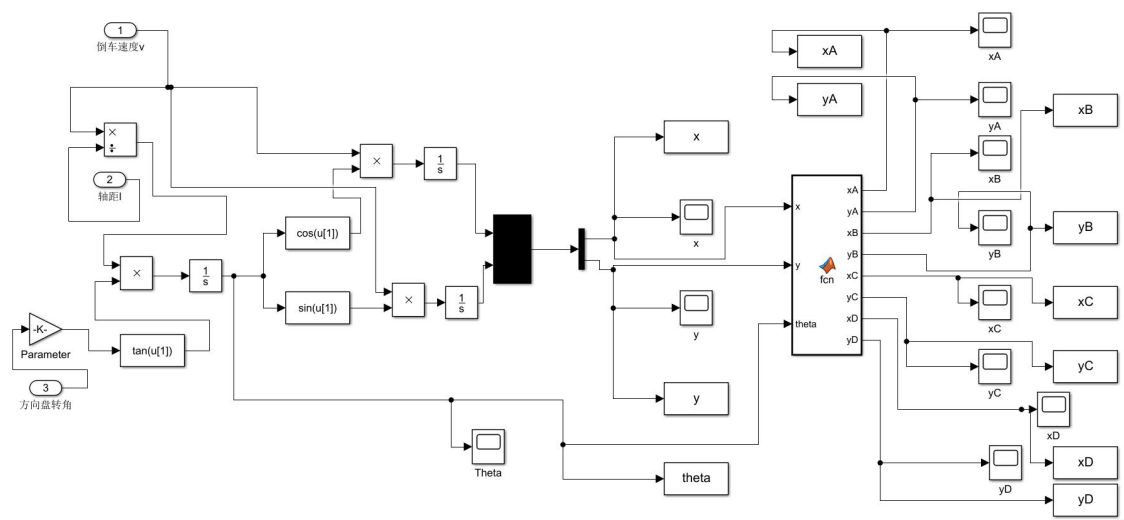


图 3.5 Simulink 仿真模型
Fig.3.5 Simulink simulation model

3.4.3 车速对泊车系统的影响

车速主要影响以下因素：

(1) 车辆泊车速度对电子设备有着重要影响，例如当泊车过程中车速较快，导致路径跟踪控制中由于信息采集不完全，造成泊车操作失败或者泊车结果不理想。

(2) 自动泊车系统中路径跟踪控制对于泊车速度的响应也有不同。

(3) 结合驾驶员经验，车辆泊车速度适中最好，速度过快会使得车辆发生控制误差不易改正，速度过慢会导致发动机容易熄火或泊车过程中发生停车现象，所以在泊车过程中控制泊车速度在适中范围内，泊车结果才会较理想，一般速度控制在 1~5km/h。

3.5 本章小结

本章内容为自动泊车系统路径规划研究内容的上层结构，主要描述了车辆泊车参数，分为内部与外部参数，接着建立车辆运动坐标系，确定车辆前后轴中心点、车轮接地点、车身顶点坐标关系表达式，基于阿克曼转向原理建立车辆运动学模型，并在 Simulink 中搭建相应模型；最后通过对泊车稳态运动分析，证明车辆在泊车过程中泊车轨迹与车速无关。

第 4 章 泊车路径规划策略

制定合适的泊车路径是自动泊车中非常关键的一步，其目的就是提供一条参考路径来指导车辆进行泊车操作，上一章建立了车辆运动学模型，并且对车辆运动规律进行了详细分析，本章在此基础上，选择自动泊车三种类型进行研究，针对平行泊车类型确定两圆弧相切路径规划策略，垂直、斜式泊车类型采取两步式策略规划路径。

4.1 泊车类型

根据车辆与空闲泊车位之间的位置关系，可以把自动泊车类型分为以下三种：

（1）平行式泊车：三种泊车类型中最复杂的一种，车辆先向右转动方向盘，之后再向左转动方向盘，最终停入泊车位，一般泊车位位于车辆右方，如图 4.1 所示。

（2）垂直式泊车：泊车位与水平方向呈 90° 夹角，如图 4.2 所示。

（3）斜式泊车：车位方向与水平方向形成一定夹角的泊车位，如图 4.3 所示。

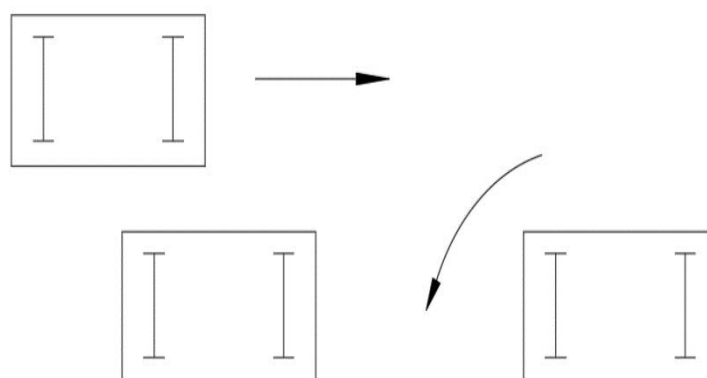


图 4.1 平行式泊车示意图
Fig.4.1 Schematic diagram of parallel parking

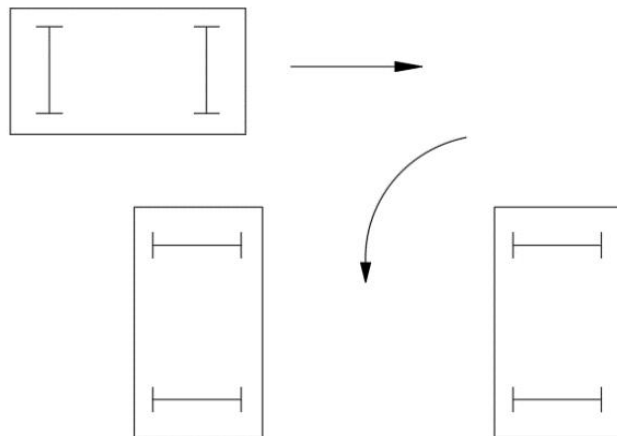


图 4.2 垂直式泊车示意图
Fig.4.2 Schematic diagram of vertical parking

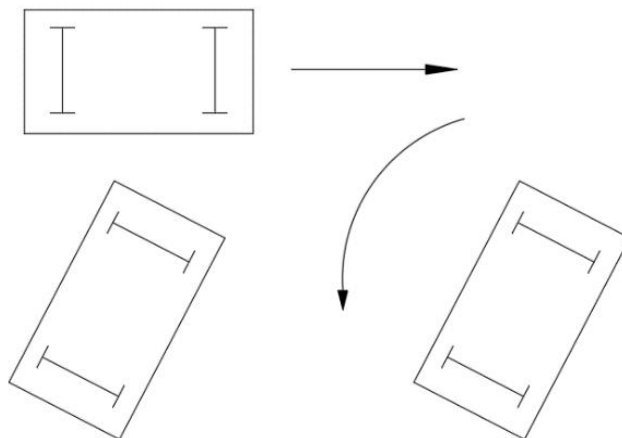


图 4.3 斜式泊车示意图
Fig.4.3 Schematic diagram of oblique parking

4.2 泊车过程分析及路径规划

4.2.1 平行泊车路径规划

现实生活中，最难的泊车类型为平行泊车，其发生交通事故的概率是这三种泊车类型中最高的，根据车辆与泊车位之间的关系，平行泊车可以分为车辆在泊车位左边和右边两种类型，本文选择车辆位于泊车位左边进行分析，当然另一种类型的理论推导也是一样的。

由日常生活可知，平行泊车过程中主要分为两个步骤：当车辆到达初始泊车区域时，先向右转动方向盘，控制车辆以一定半径的圆周运动向目标泊车区域靠

近，当车辆到达某一特定位置时，再将方向盘向相反方向转动，控制车辆回正之后再与之前轨迹运动方向相反做圆周运动，直至车辆泊入目标区域并且车身回正。综上所述，可以概括为以 O_1 、 O_2 为圆心， R_1 、 R_2 为半径相切圆弧组成的泊车路径，俗称两圆弧相切路径法，如图 4.4 所示。

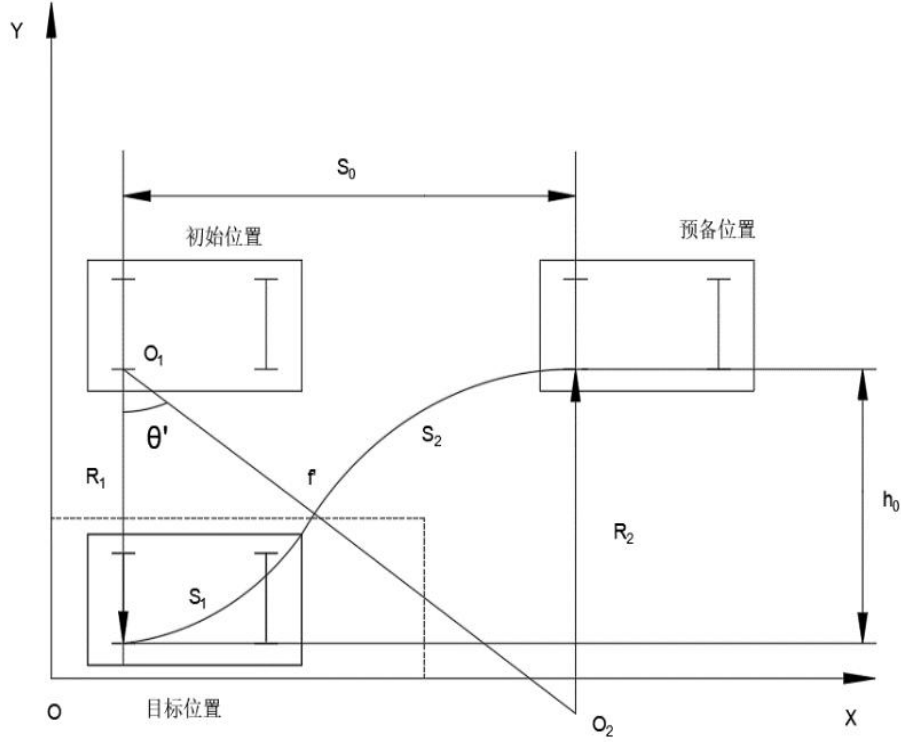


图 4.4 两圆弧相切路径示意图

Fig.4.4 Schematic diagram of the tangent path of two arcs

由图 4.4 中数学关系可以得到：

$$\Delta_T x = (R_1 + R_2) \sin \theta' - S_0 = 0 \quad (4-1)$$

$$\Delta_T y = (R_1 + R_2)(1 - \cos \theta') = h_0 \quad (4-2)$$

$$\Delta_T \theta = \Delta_1 \theta + \Delta_2 \theta = \frac{S_1}{R_1} + \frac{S_2}{R_2} = 0 \quad (4-3)$$

$$S_0 = (R_1 + R_2) \sin \theta' \quad (4-4)$$

由公式 (4-2) 有：

$$\cos \theta' = 1 - \frac{h_0}{R_1 + R_2} \quad (4-5)$$

将公式 (4-4) 与公式 (4-5) 平方相加有，

$$\left(\frac{S_0}{R_1 + R_2}\right)^2 + \left(1 - \frac{h_0}{R_1 + R_2}\right)^2 = 1 \quad (4-6)$$

根据公式（4-6）得，

$$R_1 + R_2 = \frac{S_0^2 + h_0^2}{2h_0} \quad (4-7)$$

$$S_0 = \sqrt{2h_0(R_1 + R_2) - h_0^2} \quad (4-8)$$

式中： R_1 、 R_2 分别为转向盘左、右打死时车辆外侧轮胎转弯半径； S_1 ， S_2 分别为转向盘左、右打死时车辆行驶路径； S_0 为车辆从初始泊车区域到目标泊车区域之间的横向位移； h_0 为车辆从初始泊车区域到目标泊车区域之间的纵向位移； θ' 为车辆由 f' 行驶至目标泊车区域转动的角度。

分析公式（4-7）可以看到假定 S_0 ， h_0 固定时，两圆弧半径之间总有如下关系：

$$R_1 + R_2 = \text{常值} \quad (4-9)$$

δ_x 为车辆平行泊车操作完成时，车身后与目标车位之间的安全距离，设定 $\delta_x = 0.2m$ 。

4.2.2 垂直泊车路径规划

对于垂直泊车而言，一般情况下需要车辆先行驶超过泊车位，保持车辆与泊车位水平方向平行，之后操纵车辆开始泊车，最终车辆全部驶入泊车位，并且车身与垂直泊车位竖直方向平行，目前垂直泊车一般采取单步式路径策略，虽然该方法较简单，但一步直接完成泊车操作不仅需要足够好的泊车环境，而且对车位本身也有要求，不能太小。本文选择在单步式泊车策略上加以改进，采取两步式泊车策略，如图 4.5 所示，车辆首先从初始泊车位置点 a 以泊车转弯半径行驶到点 b，此时车身转角为 90° ，接着车辆按照当前航向角直线倒车行驶至点 c 处完成泊车，当车辆初始泊车位置不同时，改变点 b 到点 c 之间的长度即可得到泊车路径。

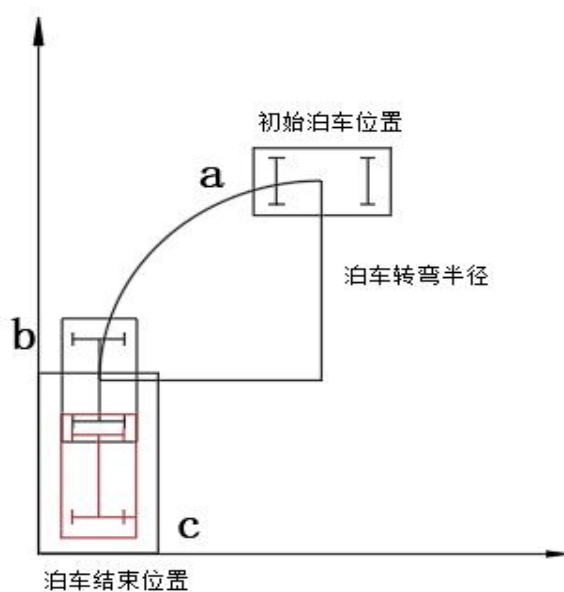


图 4.5 两步式垂直泊车路径示意图
Fig 4.5 Schematic diagram of two-step vertical parking path

4.2.3 斜式泊车路径规划

斜式泊车路径规划策略与上节分析的垂直泊车类型相似，如图 4.6 所示，分为两步：车辆先从初始位置向后倒车直至车身角度摆至与斜式车位角度一致时，完成第一步，之后车辆按照当前车身航向角沿直线行驶，直至车身完全进入泊车位，泊车完成。

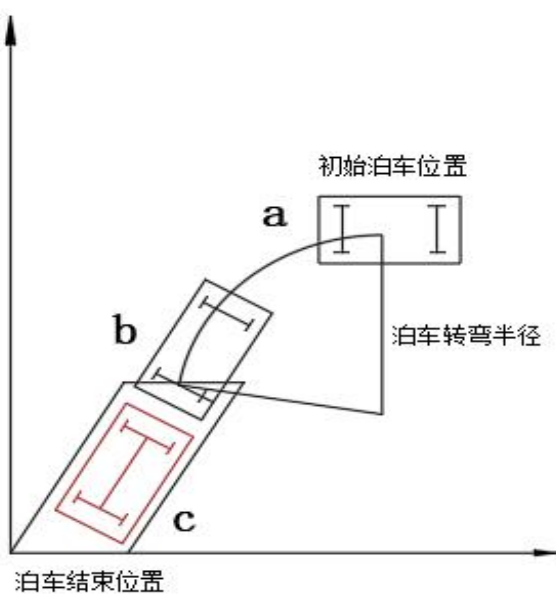


图 4.6 两步式斜式泊车路径示意图
Fig.4.6 Schematic diagram of two-step oblique parking path

4.3 停车位尺寸

根据驾校考试项目规定可知，一般侧方停车（平行泊车）的目标车位的大小为 $7\text{m} \times 2.5\text{m}$ ，即使对于新手来说，掌握一定泊车技巧后，在该车位大小的情况下也是可以完成平行泊车操作的，所以本文将平行泊车中目标车位宽设为 2.5m ，长设为 7m 作为停车位尺寸，垂直泊车中目标车位大小为 $5.3\text{m} \times 2.5\text{m}$ ，斜式泊车中目标车位大小为 $6\text{m} \times 2.8\text{m}$ 。

4.4 本章小结

本章首先确定三种泊车类型，分别为平行泊车、垂直泊车和斜式泊车，并且规划相应泊车类型的泊车路径，针对平行泊车确定两圆弧相切泊车路径，垂直和斜式泊车采用两步式泊车路径策略，最后设定三种类型目标停车位大小。

第 5 章 模糊控制

模糊控制是一种新型先进的控制系统，它所采用的是模糊数学语言描述的控制规则对给定系统工作进行控制的形式，与传统的自动控制相比，系统无需精确的模型也能得到有效的控制，而且能够应对不同的初始泊车工况。本文针对三种不同泊车类型和模糊控制特点，采用模糊逻辑控制理论建立了自动泊车模糊控制器^[63]。

5.1 MATLAB 简介

在模糊控制设计之前，首先介绍 Matlab，它是一个功能强大的数学应用软件，从创建到成为一个工业巨头，它只有大约四十年的历史。Matlab 以其强大的功能、优异的性能和很强的开放性，使用起来非常快捷，得到了大众的认可。它具有矩阵处理、图像处理、实时控制、模型仿真、符号计算等功能，在以商品的形式出现之前，Matlab 在大众的眼里还并不是那么亮眼，并没有被大众认可，但是在出现后的几年里，由于它超强的开放性与可拓展性，可以对自带内容进行拓展研究，并对其中一些功能进行完善与改进，实现自己研究方法机理，本文采用 Matlab 中 fuzzy 模糊逻辑控制箱进行建模与仿真。

5.2 模糊控制理论概述

5.2.1 模糊控制发展历程及背景

目前模糊控制理论的地位与二十世纪 60 年代、70 年代有着根本的不同，模糊逻辑具有良好的数学基础，已经逐渐渗透到许多基础学科当中，例如汽车理论、材料力学等，影响日益显著，并且模糊控制理论也开始向家电、生物工程、决策分析和识别系统等各个方向发展。模糊集理论是由 Zadeh 教授提出的，他将该理论具体划分为两个模块，分别是模糊集和模糊算法，之后关于模糊控制的应用与研究都是在该基础之上的。1974 年，Mandani 教授创造性的将模糊控制理论应用于工业产业当中，具体是将模糊理论与蒸汽机、锅炉温度等因素的控制相结合，这一创举为今后的模糊控制应用于理论创新提供了指导价值。在前人研究的基础

上，日本等国家将模糊控制应用于公共交通当中，具体是通过模糊集、模糊规则等控制地铁的横向与纵向运动，加速了模糊控制的应用场景。自此以后，模糊控制在多方面领域全面发展，并且自身也在不断与各项新技术融合。

经典的规则理论在解决线性时变系统的控制问题上是非常有效的，虽然随着人工智能以及 5G 的问世，传统控制理论和现代控制理论与之前相比也有了很大的进步，但是它在解决某些系统的控制问题还很困难，原因在于该控制原理都是需要精确描述系统中被控制变量的数学模型，再由实际问题中的情形制定相应的控制规则，由建立的数学模型输出得到需要的变量，这样对于工业领域中的实际问题应用，是非常繁琐的，往往很难建立被控对象（如本文研究情形之一的平行泊车），即便搭建好模型，其效率也是十分低的，而模糊控制是不需要对受控对象（或过程）建立精准的数学模型，它是依靠先验知识，人类实践过程中总结的知识与经验，运用人类的思维逻辑来对系统进行控制，这是它与经典控制之间的根本区别。

5.2.2 模糊控制特点

模糊控制理论是一种计算机辨识方法，其原理是非结构化语言变量和模糊逻辑中模糊推理得到的模糊规则进而控制被控变量，由上文关于模糊控制理论的描述可知，当传统控制理论面临无法解决的问题时，通常采用模糊控制理论代替解决，这是由于模糊控制具有以下特点：

（1）系统的不确定性。一是系统模型未知或对系统已知条件过少，导致系统模型难以建立；二是系统的结构和参数没有固定值并且上下浮动较大；对于以上这两种情况，模糊控制理论都可以解决。

（2）系统的非线性。当被控对象具有非线性特性，传统的线性系统控制规则理论虽然较成熟，但是控制效果不是很理想。

（3）系统对复杂任务的要求。对于许多智能控制系统来说，自动决策规划能力是必须的。

5.2.3 模糊控制主要组成

模糊控制系统主要组成部分如下^[64]：

(1) 模糊化：事实上就是将具体数值域与模糊语言域之间建立联系，例如把一个人的年龄处于零到二十五岁，都统称为青年，在数值测量分析中，将数值信号描述在语言域中，将输入变量转换为语言变量的过程，其中最重要的模糊处理方法是语义关系的生成、训练、线性与非线性划分。

(2) 模糊规则库：将系统控制的 if 规则库存储在模糊规则库中，模糊规则库是通过客观世界真实存在的经验所建立的，该经验是通过实践活动累积得到的，它是根据人类的直觉逻辑通过智能化的方法提取出来的，获得模糊规则的方法有多种^[65]：

1) 基于专家的实际操作过程。

当许多合格的技术人员在人工控制的复杂工业系统中工作时，这些专业技术人员使用模糊控制规则来工作，尽管他们基本上都不知道什么是模糊控制，但是在实际工业生产等工作中，他们凭借多年的经验都可以完成既定的工作效果，这是因为其中的经验就相当于是专家经验，在模糊控制器的设计过程中，这种经验就可以作为模糊规则对被控变量进行控制，指定输出变量，规则的提取可以通过记录实际操作步骤，例如，当工人控制设备时，可以通过输入和输出记录来总结模糊规则。

2) 模糊规则的提取与实际过程相似。

这种方法可以说是直接提取专家经验，但是可以利用每个控制系统的不同以及控制原理对模糊控制规则完善以及调整，使得对系统控制更加精准。

3) 基于模糊控制器的自主学习。

这种模糊控制器具有自学习功能，是自学习模糊控制的过程，它类似于人类的自我学习，但不完全相同。这种控制方式是一种全新的控制方式，其结构是层次化的，其规则库可分为两部分：第一部分是一般规则库；二是宏规则库，由宏规则组成，跟神经网络相似，根据实际问题以及想要达到的控制效果，自主学习以达到目的。

(3) 模糊推理：先得到之前由数值域变换为语言域的模糊量进行变换，变换过程依据是模糊规则控制表。

(4) 解模糊化：由上述几步得到的仍然是一个模糊变量，需要把它转换为一个精确量，这样才可以作为控制量对系统进行控制，这一操作就是解模糊化，

虽然模糊控制中的模糊二字代表的是之前几步的含义，但最终还是要趋向精确化的，解模糊化方法如下：

1) 最大隶属度法

由方法名称即可知要找到隶属度最大的元素，如果存在多个，则需将多个元素求取平均值，这种方法对于控制系统而言简单操作有效，实施性好，但是它可利用的信息较少，因为在实际控制过程中不会选择较小的元素。

2) 中位数法

这种方法是严格按照数学方法得到相应的论域元素，认为将隶属度函数对应曲线与 x 轴包围面积均分的点，即得到论域元素。

3) 加权平均判决法

该方法的关键在于系数的选择，通常根据实际系统的要求或者经验来选择，与系统响应特性有关，加权系数的确定也是跟以上所述两点有关，确定权系数 $k_i (i=1,2,3,\dots,m)$ ，并且可以由以下公式给出模糊量的输出。

$$u' = \frac{\sum_{i=1}^m k_i u_i}{\sum_{i=1}^m k_i} \quad (5-1)$$

4) 重心法

取推理结论模糊集合隶属函数曲线与横坐标轴所围成面积重心作为代表点，即：

$$u = \frac{\int x u_N(s) dx}{\int u_N(x) dx} \quad (5-2)$$

若输出控制量的隶属函数不是连续曲线，而是单点形式时公式为：

$$u = \frac{\sum x_i u_N(x_i)}{\sum u_N(x_i)} \quad (5-3)$$

本文选择重心法作为解模糊化方法。

5.3 模糊控制器设计

5.3.1 平行泊车模糊控制器设计

由上文可知，模糊控制器变量设计越详细，模糊控制器对受控系统就能达到

更高的目标或者误差更小，这是有利有弊的，随着输入输出量个数的增加，不仅需要多重数值域和语言域进行定义，同时还意味着模糊量与输出量之间对应的模糊规则数量上升，出现“维数灾难”现象，导致系统控制错误率提升，降低模型质量，所以在选择模糊集合和输入输出变量个数以及模糊规则时要慎重，保证控制精度的同时也要提高实时性。

根据 3.4.2 中的分析可知，车辆在低速泊车过程中，车辆运动轨迹与车速并没有关系，本文平行泊车模糊控制器选择车辆后轴中心点坐标 x_r ， y_r 和车身航向角 θ 作为输入量，语言值为 N 、 Z 、 P 、 S 、 LB 、 B 、 VB ，方向盘转角 $DELTA$ 为输出变量，语言值为 NB 、 NM 、 Z 、 PM 、 PB ，各模糊语言意义见表 5.1。

表 5.1 符号的意义
Table 5.1 Meaning of symbols

N	Z	P	S	LB	B	VB	NB	NM	PM	PB
负	零	正	小	微小	大	很大	负大	负中	正中	正大

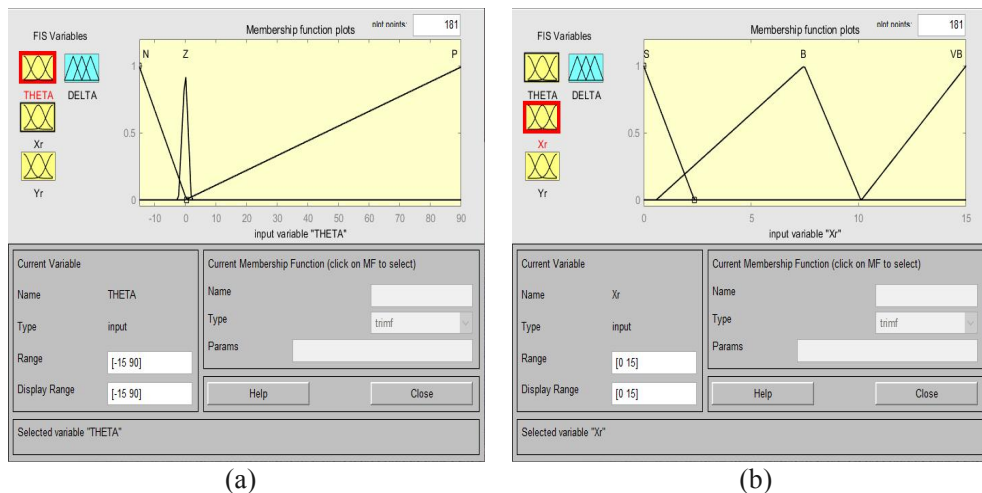
下一步是设计模糊变量隶属度函数，其确定方法分为以下几种^[66]：

(1) 主观经验法。

(2) 分析推理方法：隶属度函数的确定要根据控制系统具体问题特点，选择典型函数代替，这种方法称之为分析推理法，其中典型函数有 `trimf`、`sigmf`、`gaussmf` 等。

(3) 调查统计方法：隶属度函数由调查结果和统计数据得到曲线所确定，曲线的函数表达式即为隶属度函数。

本文针对平行泊车类型，选择三角形隶属度函数，各输入输出量对应语言域如图 5.1 所示。



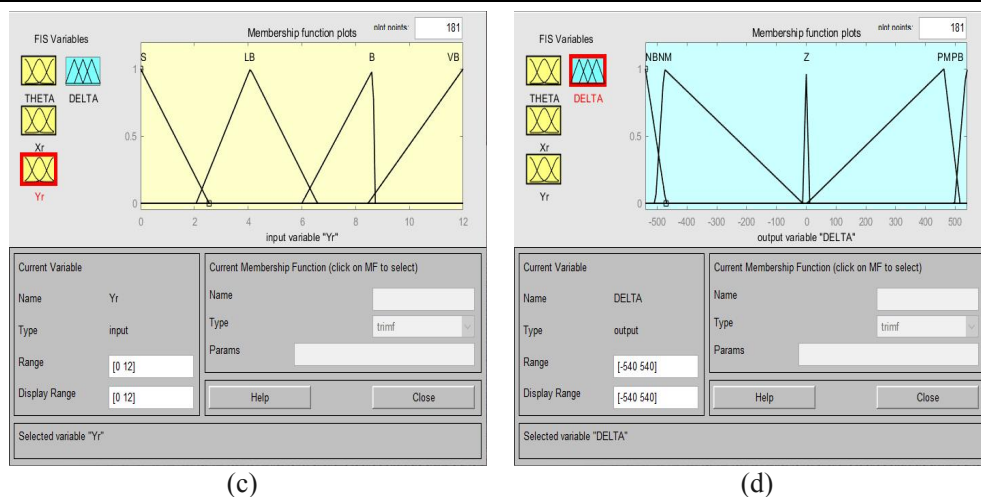


图 5.1 平行泊车隶属度函数:

(a) θ 隶属度函数;(b) x_r 隶属度函数;(c) y_r 隶属度函数;(d) $DELTA$ 隶属度函数

Fig.5.2 Parallel parking membership function:

(a) θ membership function;(b) x_r membership function;(c) y_r membership function;(d) $DELTA$ membership function;

由 4.2.1 中的泊车路径策略可知, 自动平行泊车过程分为三个阶段, 初始阶段, 方向盘向右打车辆转弯阶段, 方向盘向左打车辆转弯阶段, 并且在这三个阶段都不能与周边外界环境发生碰撞, 根据日常泊车经验, 制定平行泊车模糊控制器的模糊规则如表 5.2 所示:

表 5.2 平行泊车模糊规则控制表

Table 5.2 Parallel parking fuzzy rule control table

θ	y_r x_r	S	LB	B	VB
N	S	NM	NM	NB	NB
	B	NM	NM	NB	NB
	VB	Z	NM	NB	NB
Z	S	Z	NM	NB	NB
	B	PM	NM	NM	NB
	VB	Z	NM	NB	NB
P	S	PB	PB	NB	NB
	B	PB	PM	NB	NB
	VB	PM	NM	NM	NM

下面分析几种平行泊车模糊规则控制表中比较典型的情况:

(1) 车身航向角 θ 为 Z 、车辆后轴中心点坐标 x_r 、 y_r 均为 VB 时, 说明此时车辆位置与目标停车位之间距离较大, 并且车身水平线与坐标系水平方向之间没

有角度，这时应该增加方向盘转角，即模糊控制器输出量 $DELTA$ 为 NB 。

(2) 车身航向角 θ 为 P 、车辆后轴中心点坐标 x_r 、 y_r 均为 VB 时，说明此时不仅车辆位置与目标泊车位之间距离较大，并且车身水平线与坐标系水平方向之间存在一定角度，这时应该适当增加方向盘转角，即模糊控制器输出量 $DELTA$ 为 NM 。

(3) 车身航向角 θ 为 Z 、车辆后轴中心点坐标 x_r 、 y_r 均为 S 时，说明此时车辆位置与目标泊车位之间距离较小，并且车身水平线与坐标系水平方向之间的角度为零，这时应该保持当前方向盘转角不变，即模糊控制器输出量 $DELTA$ 为 Z 。

(4) 车身航向角 θ 为 P 、车辆后轴中心点坐标 x_r 、 y_r 均为 S 时，说明此时车辆位置与目标泊车位之间距离较小，但是车身水平线与坐标系水平方向之间存在一定角度，这时应该加大向左打方向盘转角，即模糊控制器输出量 $DELTA$ 为 PB 。

平行泊车模糊规则观察器如图 5.2 所示，Simulink 搭建仿真模型如图 5.3 所示

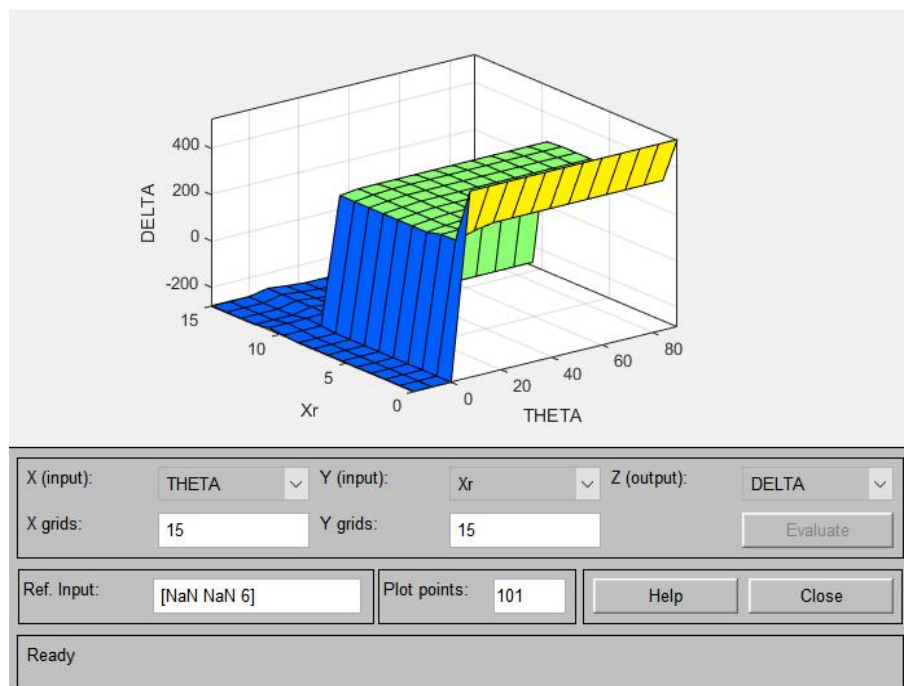


图 5.2 平行泊车模糊规则输出曲面观察器
Fig.5.3 Parallel parking fuzzy rule output surface observer

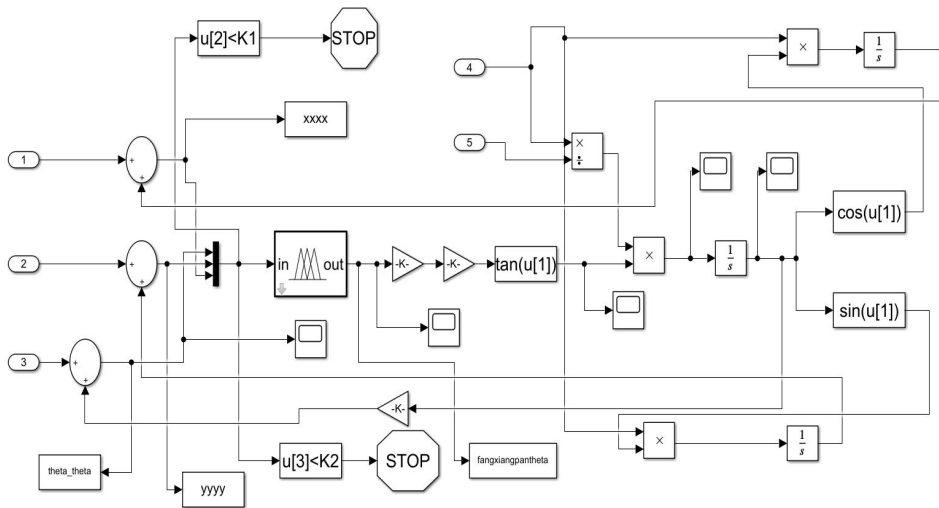


图 5.3 平行泊车仿真模型

Fig.5.3 Parallel Parking Simulation Model

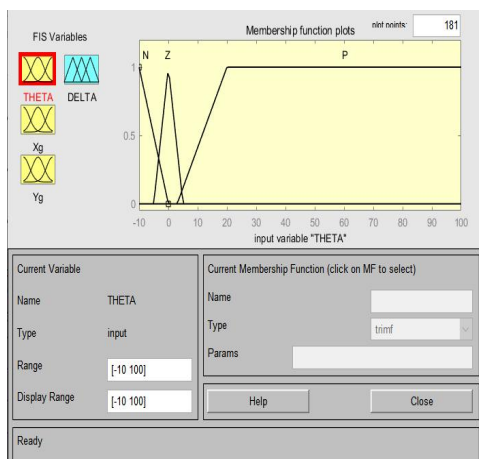
5.3.2 垂直泊车模糊控制器设计

本文垂直泊车模糊控制器选择车辆中心点坐标 x_g , y_g 和车身航向角 θ 作为模糊控制系统的控制变量, 语言值为 N 、 Z 、 P 、 S 、 B 、 VB , 方向盘转角 $DELTA$ 为输出变量, 语言值为 NB 、 NM 、 NS 、 NZ , 模糊语言意义见表 5.3, 本文选择三角形和梯形隶属度函数。

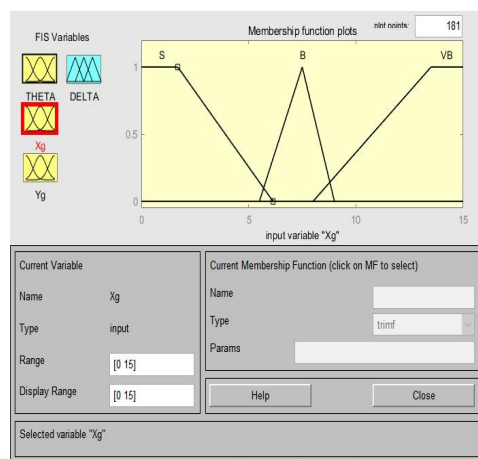
表 5.3 符号的意义
Table 5.3 Meaning of symbols

N	Z	P	S	B	VB	NB	NM	NS	NZ
负	零	正	小	大	很大	负大	负中	负小	负微

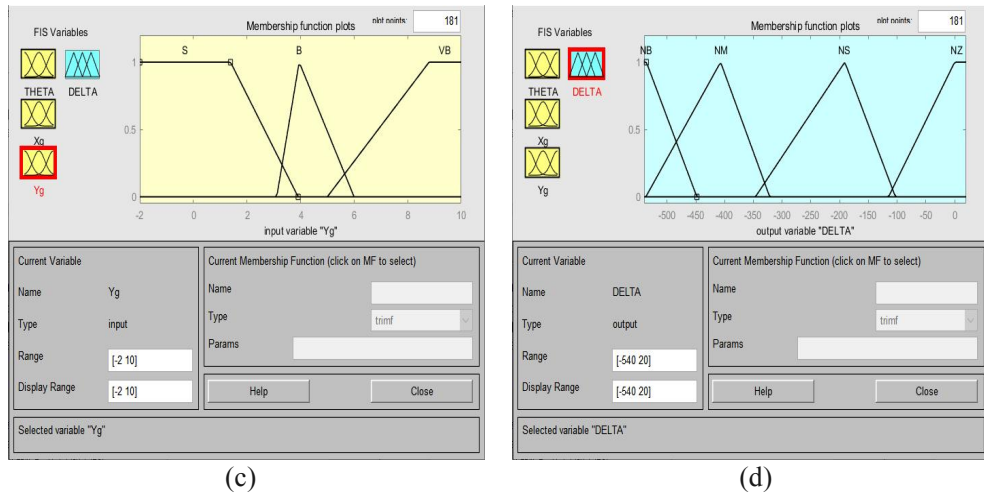
其中各输入输出模糊量的隶属度函数及对应语言值如图 5.4 所示。



(a)



(b)



(c)

(d)

图 5.4 垂直泊车隶属度函数:

(a) θ 隶属度函数;(b) x_g 隶属度函数;(c) y_g 隶属度函数;(d) $DELTA$ 隶属度函数

Fig.5.4 Vertical parking membership function:

(a) θ membership function;(b) x_g membership function;(c) y_g membership function;(d) $DELTA$ membership function;

垂直泊车模糊控制器的模糊规则如表 5.4 所示。

表 5.4 垂直泊车模糊规则控制表

Table 5.4 Vertical parking fuzzy rule control table

θ	y_g x_g	S	B	VB
N	S	NB	NM	NB
	B	NM	NS	NS
	VB	NM	NS	NS
Z	S	NB	NB	NB
	B	NM	NM	NM
	VB	NS	NM	NZ
P	S	NB	NM	NS
	B	NS	NM	NS
	VB	NZ	NS	NS

下面分析垂直泊车几种模糊规则控制表中比较典型的情况:

(1) 车身方向角 θ 为 Z 、车辆中心点坐标 x_g 、 y_g 均为 VB 时,说明此时车辆位置与目标泊车位之间距离较大,并且车身水平线与坐标系水平方向之间没有角度,这时应该转动方向盘,即模糊控制器输出量 $DELTA$ 为 NZ 。

(2) 车身方向角 θ 为 P 、车辆中心点坐标 x_g 、 y_g 均为 S 时,说明此时车辆位置与目标泊车位之间距离较小,并且车身水平线与坐标系水平方向夹角为正,

说明此时车辆即将驶入或者完成垂直泊车操作，这时应该加大方向盘转角，即模糊控制器输出量 $DELTA$ 为 NZ 。

垂直泊车中模糊规则输出曲面观察器如图 5.5 所示

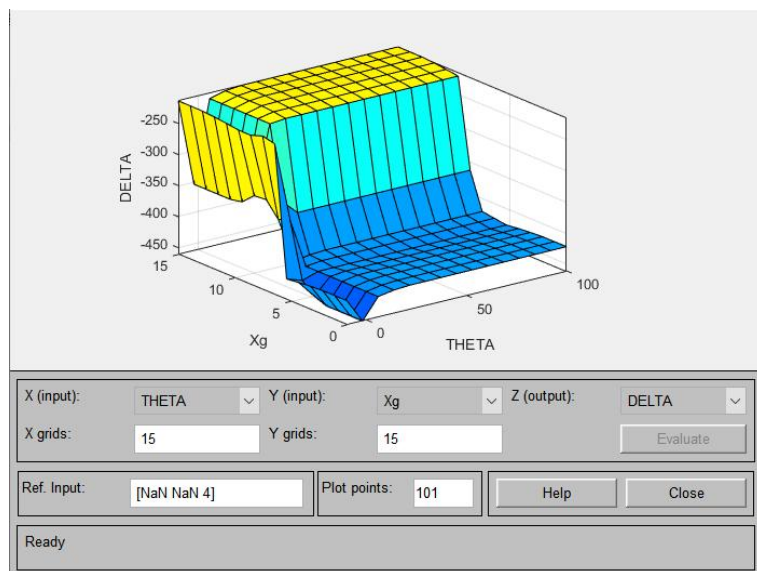


图 5.5 垂直泊车模糊规则输出曲面观察器

Fig.5.5 Vertical parking fuzzy rule output surface observer

5.3.3 斜式泊车模糊控制器设计

本文斜式泊车模糊控制器选择车辆后轴中心点坐标 x_r , y_r 和车身航向角 θ 作为模糊控制系统的控制变量，语言值为 N 、 Z 、 P 、 S 、 B 、 VB ，方向盘转角 $DELTA$ 为输出变量，语言值为 NB 、 NM 、 NZ ，模糊语言意义见表 5.5，本文选择三角形和梯形隶属度函数。

表 5.5 符号的意义
Table 5.5 Meaning of symbols

N	Z	P	S	B	VB	NB	NM	NZ
负	零	正	小	大	很大	负大	负中	负小

其中各输入输出模糊量的隶属度函数及对应语言值如图 5.6 所示。

斜式泊车模糊控制器的模糊规则如表 5.6 所示。

斜式泊车模糊规则输出曲面观察器如图 5.7 所示

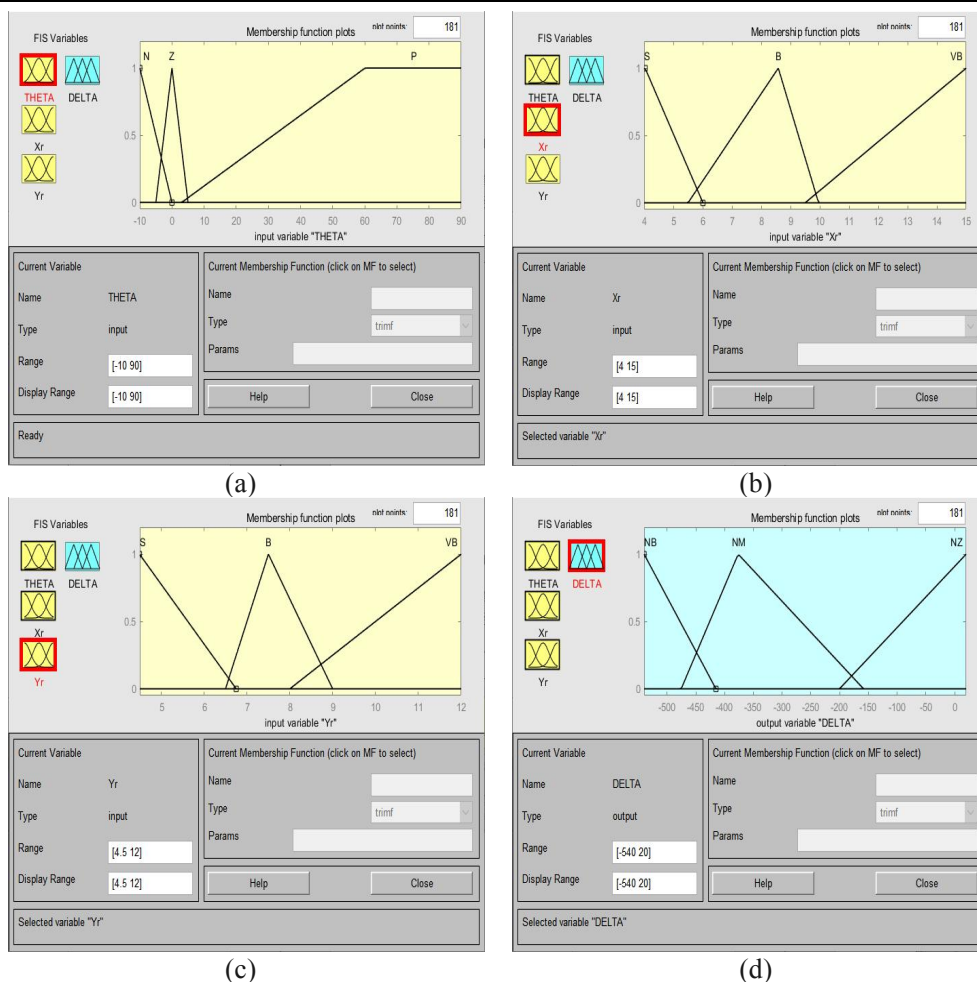


图 5.6 斜式泊车隶属度函数:

(a) θ 隶属度函数;(b) x_r 隶属度函数;

(c) y_r 隶属度函数;(d) $DELTA$ 隶属度函数

Fig.5.6 Oblique parking membership function:

(a) θ membership function;(b) x_r membership function;

(c) y_r membership function;(d) $DELTA$ membership function;

表 5.6 斜式泊车模糊规则控制表

Table 5.6 Oblique parking fuzzy rule control table

θ	y_r x_r	S	B	VB
N	S	NB	NM	NB
	B	NM	NM	NZ
	VB	NZ	NZ	NM
Z	S	NB	NB	NB
	B	NM	NM	NB
	VB	NZ	NZ	NM
P	S	NB	NB	NB
	B	NB	NB	NZ
	VB	NZ	NZ	NM

下面分析几种斜式泊车模糊规则控制表中比较典型的情况：

(1) 车身方向角 θ 为 Z 、车辆后轴中心点坐标 x_r 为 VB ， y_r 为 S 时，说明此时车辆水平位置距离车位较远，但是竖直方向距离车位较近，并且车身水平线与坐标系水平方向之间没有角度，此时应该轻微转动方向盘，控制车辆先向车位靠近，即模糊控制器输出量 $DELTA$ 为 NZ 。

(2) 车身方向角 θ 为 P 、车辆后轴中心点坐标 x_r 、 y_r 均为 S 时，说明此时车辆无论是水平还是数值方向均距离车位较近，并且车身水平线与坐标系水平方向之间角度为正，此时应该加大方向盘转角，控制车辆以更小转弯半径驶入车位，即模糊控制器输出量 $DELTA$ 为 NB 。

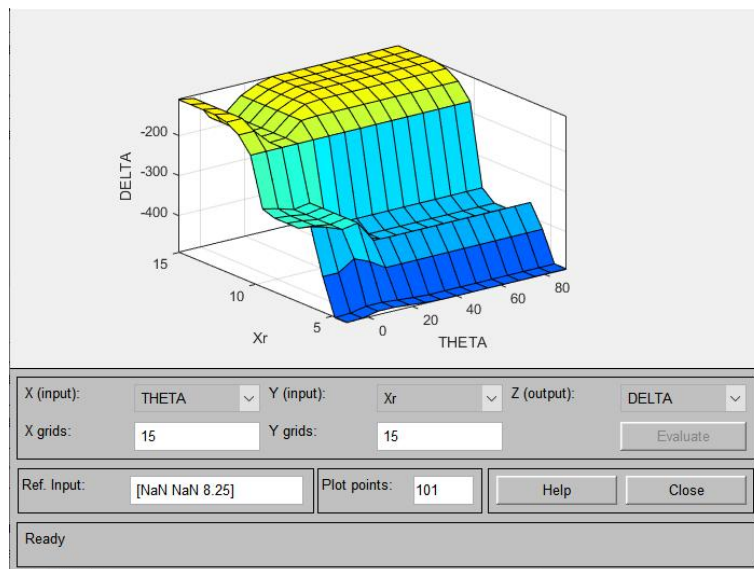


图 5.7 斜式泊车模糊规则输出曲面观察器

Fig.5.7 Oblique parking fuzzy rule output surface observer

5.4 自动泊车仿真验证

基于第三、四章中建立的车辆运动学坐标系、车辆运动学模型以及本章建立的模糊控制器，针对三种不同泊车类型，分别采用三种不同初始泊车工况对设计的模糊控制器进行验证，实现了在一定初始条件下的自动泊车操作。

5.4.1 平行泊车仿真验证

(1) 工况一：

当 $x_r = 9.86\text{m}$, $y_r = 6\text{m}$, $THETA = 0^\circ$ 时, 车辆距车位的横纵向距离均适中, 即工况一是验证 4.2.1 节中规划的平行泊车路径, 模糊控制器跟踪路径与之前规划的自动泊车路径对比如下图 5.8 所示, 图 5.9 (a) 为工况一下平行泊车仿真结果, 车身航向角结果如图 5.9 (b) 所示。由图 5.8、5.9 可知, 本文方法设计的模糊控制器基本可以控制车辆按照既定泊车路径, 实现泊车操作且没有与停车位周边发生碰撞, 安全泊车, 虽然最后终止泊车角度为 1.9785° , 并没有完全实现车身与水平方向完全平行, 但在实际泊车过程中, 角度较小可以忽略对泊车终止位姿的影响, 达到既定控制效果。

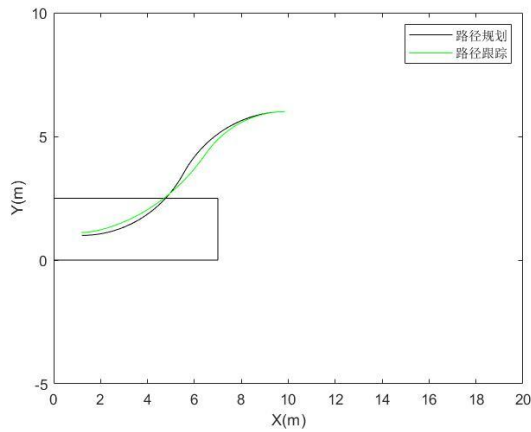


图 5.8 规划路径与跟踪路径对比图

Fig.5.8 Comparison of planned path and tracking path

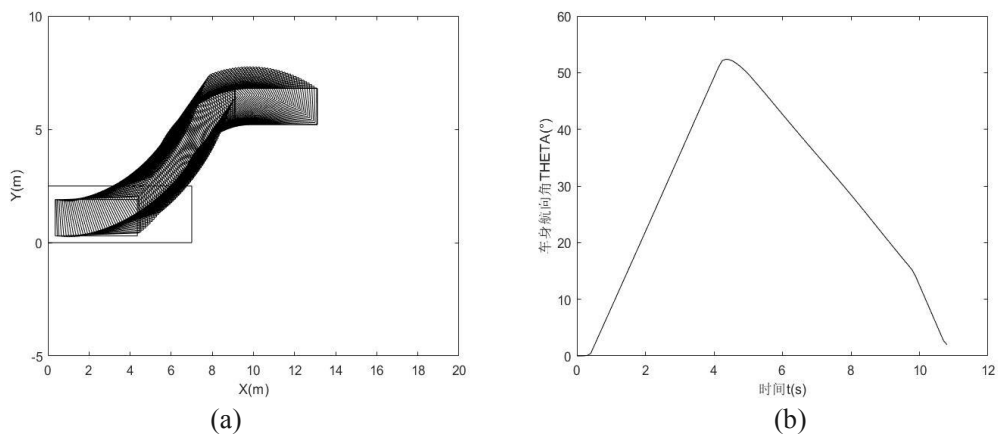


图 5.9 平行泊车工况一:

(a)平行泊车仿真图;(b)车身航向角变化图

Fig.5.9 Parallel parking condition one:

(a)Parallel parking simulation diagram;(b)Map of body heading angle change

(2) 工况二、工况三:

当 $x_r = 9.86\text{m}$, $y_r = 5.5\text{m}$, $THETA = 5^\circ$ 时, 此时车辆初始泊车车身位姿与水

平方向成正向夹角，图 5.10 (a) 为工况二下车辆泊车仿真结果，由图 5.10 (b) 可知，最终终止泊车角度为 0.4649° 。当 $x_r = 9.7\text{m}$ ， $y_r = 6\text{m}$ ， $THETA = -5^\circ$ 时，此时车辆初始泊车车身位姿与水平方向成负向夹角，图 5.11 (a) 为工况三下车辆泊车仿真结果，由图 5.11 (b) 可知，最终终止泊车角度为 -0.1965° 。由图 5.10、5.11 中可以看出，两种不同工况下，模糊控制器控制车辆泊车操作完成度较高，没有特别明显的波动迹象，并且没有发生碰撞现象，车身完全进入车位并且接近于摆正，适用于平行泊车工况。

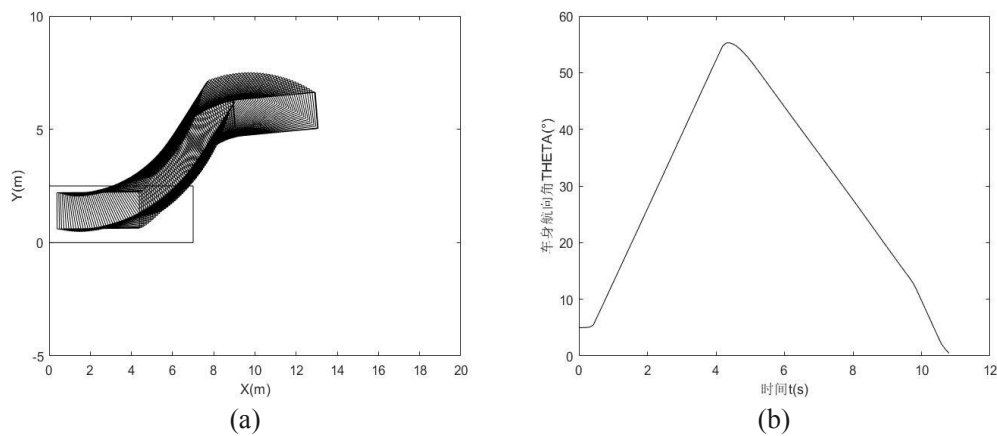


图 5.10 平行泊车工况二：
(a)平行泊车仿真图;(b)车身航向角变化图

Fig.5.10 Parallel parking condition two:

(a) Parallel parking simulation diagram;(b) Map of body heading angle change

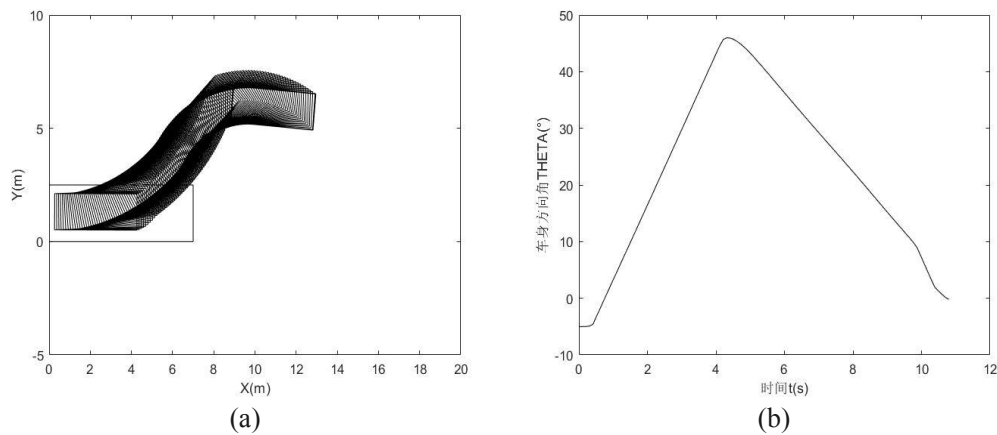


图 5.11 平行泊车工况三：
(a)平行泊车仿真图;(b)车身航向角变化图

Fig.5.11 Parallel parking condition three:

(a) Parallel parking simulation diagram;(b) Map of body heading angle change

5.4.2 垂直泊车仿真验证

(1) 工况一：

当 $x_g = 6.3\text{m}$, $y_g = 3.5\text{m}$, $THETA = 0^\circ$ 时, 验证之前 4.2.2 规划的垂直泊车路径, 图 5.12 (a) 为垂直泊车模糊控制器跟踪路径与之前规划的自动泊车路径对比, 图 5.12 (b) 为垂直泊车沿跟踪路径仿真结果, 图 5.12 (c) 为车身航向角示意图, 图 5.12 (d) 为完整泊车仿真图, 蓝色轮廓为车辆按照终止航向角继续沿直线驶入泊车位轨迹。由图 5.12 可知, 在模糊控制器跟踪路径基本与规划路径相吻合, 并且在车辆按直线驶入车位之前, 车辆的终止车身航向角为 89.8321° , 与垂直泊车位之间相差角度为 0.1679° , 可以认为车辆已摆正方向, 之后直线行驶车辆终止位姿基本与泊车位平行, 履行两步式规划策略且完成度较高, 并且在整体泊车过程中, 车身航向角变化平稳, 没有较大波动, 符合垂直泊车一般情形。

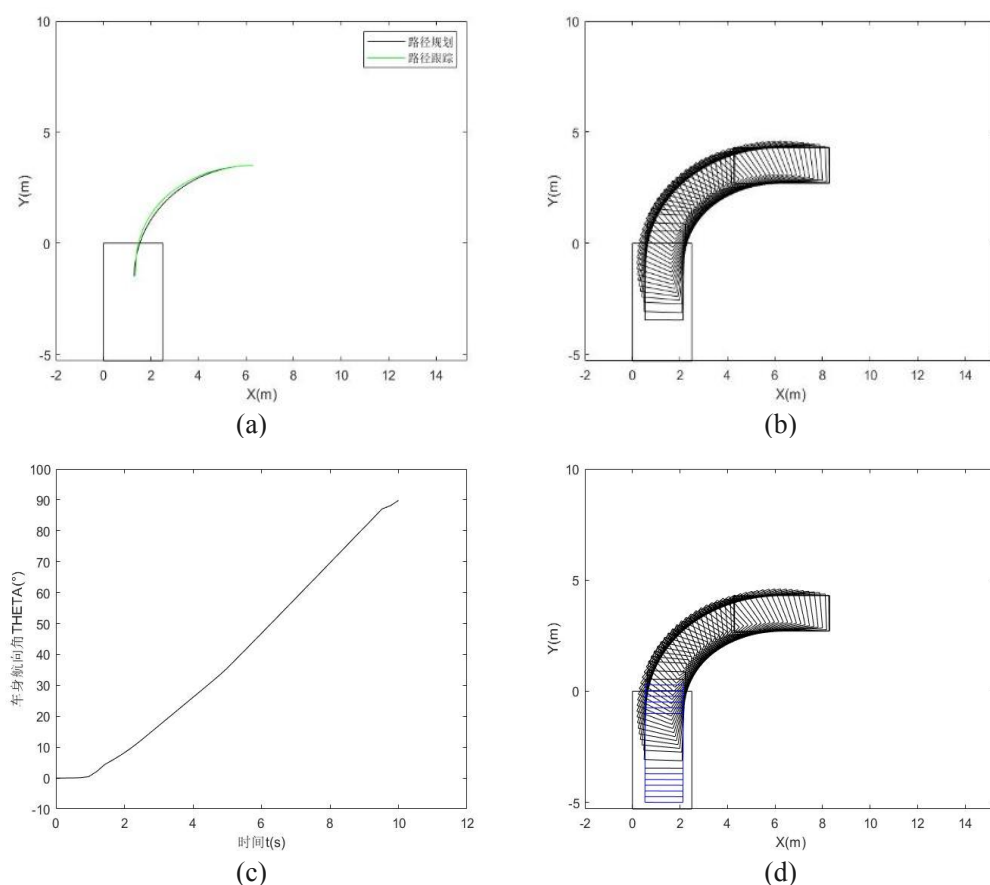


图 5.12 垂直泊车工况一:

(a)规划路径与跟踪路径对比图;(b)跟踪路径仿真图;
 (c)车身航向角变化图;(d)完整垂直泊车仿真图

Fig.5.12 Vertical parking condition one:

(a)Comparison chart of planned path and tracking path;(b)Path tracking simulation diagram;
 (c)Map of body heading angle change;(d)Complete vertical parking simulation diagram

(2) 工况二、工况三:

当 $x_g = 6\text{m}$, $y_g = 3.5\text{m}$, $THETA = 15^\circ$, 此时车辆初始泊车车身位姿与水平方向成正向夹角, 图 5.13 (a) 为工况二车辆泊车仿真结果, 由图 5.13 (b) 可知, 车辆直线驶入泊车位之前的车身航向角为 91.0546° 。当 $x_g = 7.5\text{m}$, $y_g = 5.5\text{m}$, $THETA = -10^\circ$, 此时车辆初始泊车车身位姿与水平方向成负向夹角, 图 5.14 (a) 为工况三车辆泊车仿真结果, 由图 5.14 (b) 可知, 车辆直线驶入泊车位之前车身航向角为 89.8602° 。由以上两种不同工况可以看出, 当车辆初始位姿中车身与水平方向成一定角度时, 垂直泊车模糊控制器可以控制车辆完成泊车操作, 车辆均可以泊入泊车位, 但是在工况三中车辆右侧与泊车位边界之间距离为 20.38cm , 刚刚达到安全距离, 出现这种情况是因为工况三中车辆初始角度为负造成的, 由驾驶员经验可知, 工况三一般是在实际垂直泊车情形中出现的, 所以本文设计的垂直泊车控制器适用于自动泊车垂直工况中。

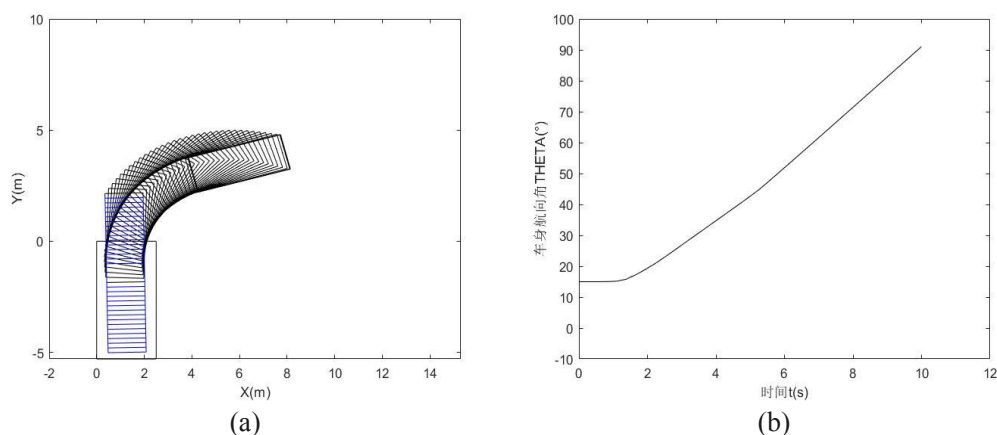


图 5.13 垂直泊车工况二:(a)垂直泊车仿真图;(b)车身航向角变化图

Fig 5.13 Vertical parking condition two:

(a)Vertical parking simulation diagram;(b)Map of body heading angle change

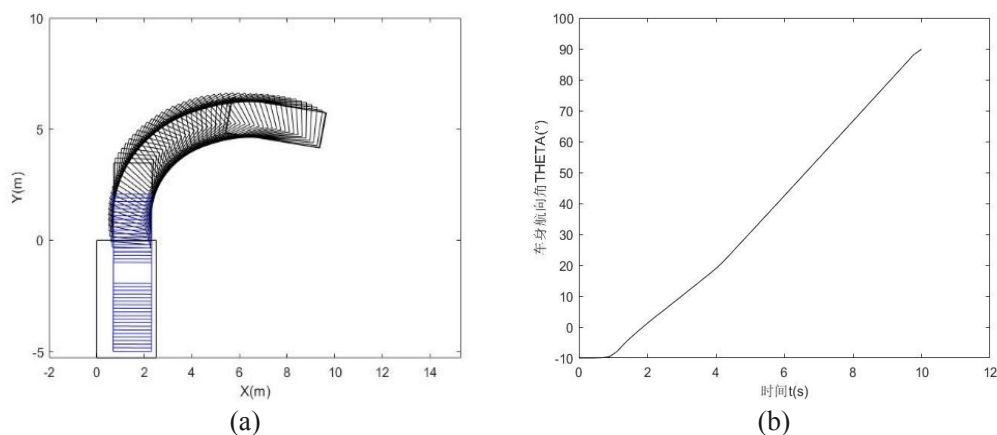


图 5.14 垂直泊车工况三:(a)垂直泊车仿真图;(b)车身航向角变化图

Fig 5.13 Vertical parking condition three:

(a)Vertical parking simulation diagram;(b)Map of body heading angle change

5.4.3 斜式泊车仿真验证

(1) 工况一:

当 $x_r = 8.7\text{m}$, $y_r = 7.7\text{m}$, $THETA = 0^\circ$ 时, 验证之前 4.2.3 规划的斜式泊车路径, 图 5.15 (a) 为斜式泊车模糊控制器跟踪路径与之前规划的自动泊车路径对比, 图 5.15 (b) 为斜式泊车沿跟踪路径仿真结果, 图 5.15 (c) 为车身航向角示意图, 图 5.15 (d) 为完整泊车仿真图, 蓝色轮廓为车辆按照终止航向角继续沿直线驶入停车位轨迹。由图 5.15 可知, 模糊控制器跟踪路径基本与规划路径相吻合, 本文选择斜式停车位角度为 60° , 车辆在工况一按直线驶入车位之前的终止车身航向角为 60.4905° , 基本可以认为车身已经与车位平行, 符合一般斜式泊车情形。

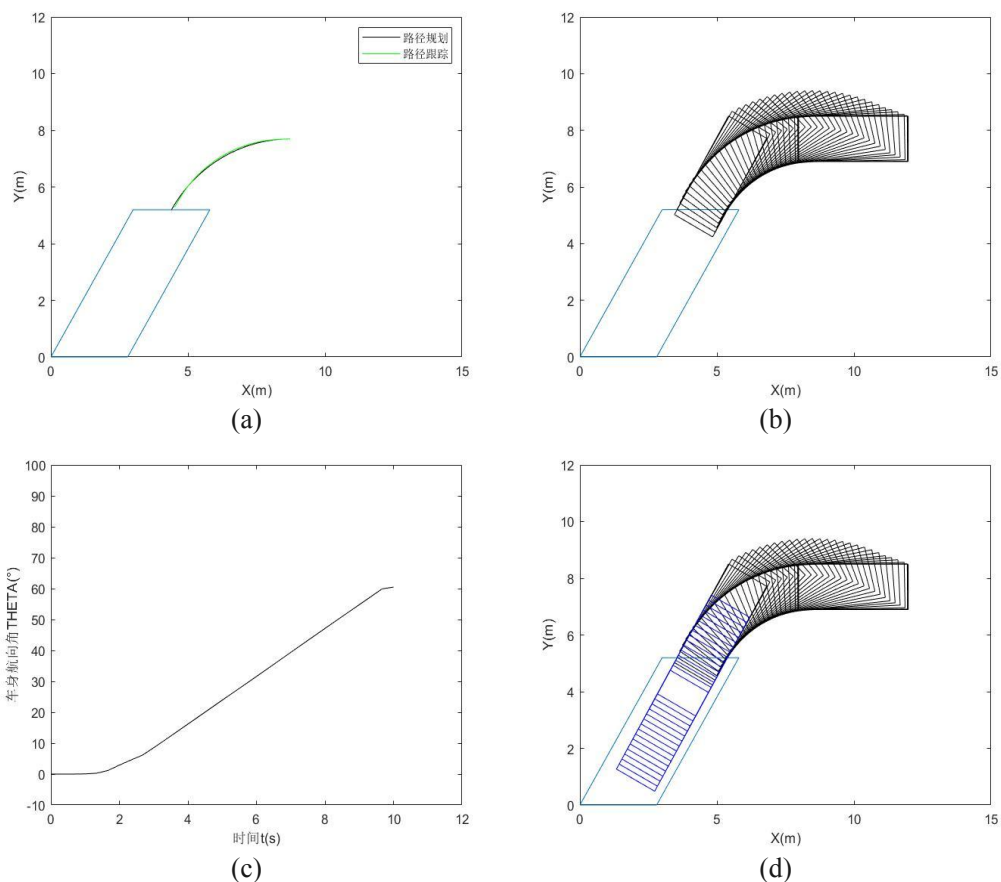


图 5.15 斜式泊车工况一:

(a)规划路径与跟踪路径对比图;(b)跟踪路径仿真图;
(c)车身航向角变化图;(d)完整斜式泊车仿真图

Fig.5.15 Vertical parking condition one:

(a)Comparison chart of planned path and tracking path;(b)Path tracking simulation diagram;
(c)Map of body heading angle change;(d)Complete inclined parking simulation diagram

(2) 工况二、工况三:

当 $x_r = 7.8\text{m}$, $y_r = 7\text{m}$, $THETA = 15^\circ$ 时, 此时车辆初始泊车车身位姿与水平方向成正向夹角, 图 5.16 (a) 为工况二车辆泊车仿真结果, 由图 5.16 (b) 可知, 车辆直线驶入泊车位之前的车身航向角为 60.7287° 。当 $x_r = 9\text{m}$, $y_r = 8\text{m}$, $THETA = -10^\circ$, 此时车辆初始泊车车身位姿与水平方向成负向夹角, 图 5.17 (a) 为工况三车辆泊车仿真结果, 由图 5.17 (b) 可知, 车辆直线驶入泊车位之前的车身航向角为 60.8576° 。由以上三种不同工况可知, 车辆通过斜式泊车模糊控制器控制可以正确驶入泊车位当中, 并且车身方向与车位方向几乎平行, 完成斜式自动泊车操作。

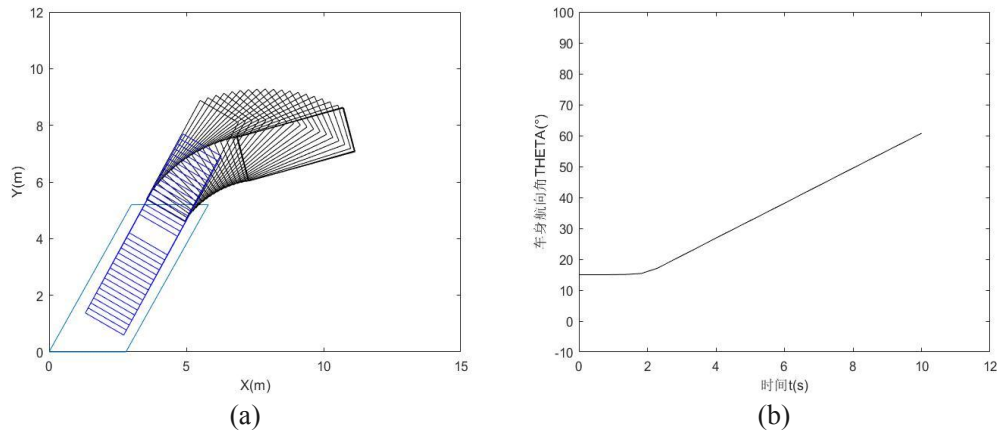


图 5.16 斜式泊车工况二:

(a)斜式泊车仿真图;(b)车身航向角变化图

Fig.5.16 Inclined parking condition two:

(a)Simulation diagram of oblique parking;(b)Map of body heading angle change

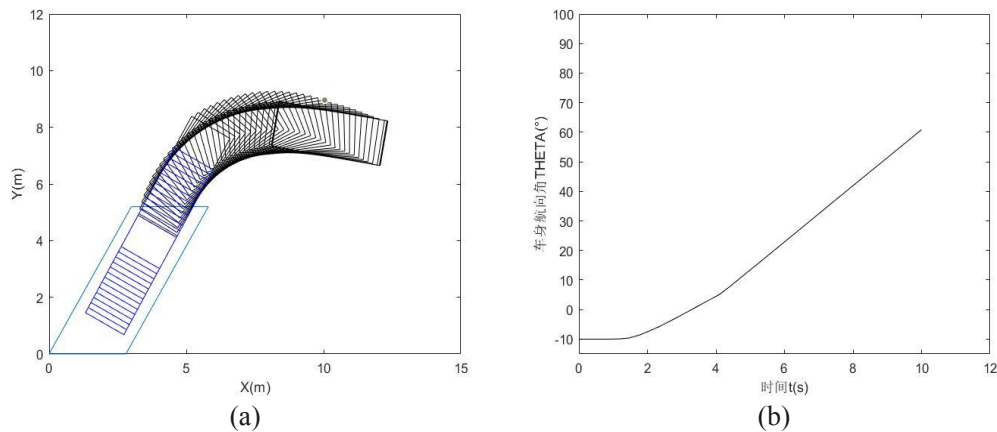


图 5.17 斜式泊车工况三:

(a)斜式泊车仿真图;(b)车身航向角变化图

Fig.5.17 Inclined parking condition three:

(a)Simulation diagram of oblique parking;(b)Map of body heading angle change

5.5 本章小结

本章主要介绍了 Matlab 中模糊控制工具箱、模糊控制发展历程、特点以及主要组成部分，突出了模糊控制的优越性；对于三种不同泊车类型：平行、垂直和斜式泊车，分别设计相应模糊控制器，确定模糊控制器输入输出量和隶属度函数，建立模糊规则控制集；针对每种泊车类型，分别设定三种不同初始条件泊车工况，验证模糊控制器控制效果，结果表明，车辆泊车操作完成度较高且终止位姿基本与相应车位类型所平行，验证本文设计自动泊车模糊控制器的有效性以及路径规划策略的可靠性。

结 论

本文针对自动泊车系统，首先提出了一种自动检测车位线方法，之后对不同泊车类型进行路径规划，最后利用模糊控制原理设计了模糊控制器对之前规划的路径进行跟踪控制，以下为本文的主要研究成果：

(1) 提出了一种基于边缘编组的车位线检测方法，利用了当下流行的图像处理工具 OpenCV，对泊车位图片做图像预处理，包括灰度化、二值化、直方图均衡化、形态学去噪等操作，结果图像质量明显提升，有效削减噪声等其他因素影响；接着对预处理后的图像采用 Canny 边缘检测算法，并通过对该算法梯度计算模板和双阈值的改进，使检测得到的图像边缘更加细致、假边缘更少；之后采用 8 邻域边缘编组，得到独立的且互相连接的边缘组，消除短线的干扰，再利用相邻像素点间梯度差分概念剔除非车位线特征边缘组，得到更加细致简洁的车位线轮廓，最后通过概率霍夫变换检测得到车位线。结果表明：本文算法相对比与传统算法检测得到的车位线正确率更高并且所耗时间相对减少，且能有效克服漏检和错检现象。

(2) 建立车辆运动坐标系和车辆运动学模型，确定车辆前后轴中心点、车轮接地点、车身顶点等关键点坐标关系表达式，通过对泊车运动过程分析，证明车辆在泊车过程中路径轨迹与车速无关。

(3) 针对三种不同泊车位类型，分别对平行、垂直和斜式车位规划泊车路径，对于平行泊车类型确定两圆弧相切路径，垂直和斜式泊车类型采用两步式泊车策略，并确定相应泊车位尺寸。

(4) 采用模糊控制理论对三种不同泊车类型的路径进行跟踪控制，结合主观经验法确定平行、垂直和斜式模糊控制器的隶属度函数以及具体模糊规则，最后通过仿真验证本文设计的模糊控制器可以跟踪之前的规划泊车路径，并且每种泊车类型在设定的不同工况下，均可完成自动泊车操作，且完成度较高，车身航向角变化平稳，终止泊车位姿与对应车位类型平行，实现了自动泊车操作。

本文存在的不足以及对后续工作的展望：

(1) 泊车环境感知趋于多传感器融合策略。

(2) 模糊控制器中的模糊规则是通过驾驶员主观经验提取的，所以模糊控制逻辑还有提升空间。

(3) 只将自动泊车系统中三大模块单独验证，未联合起来。

参考文献

- [1] 李红, 王文军, 李国强. 基于 B 样条理论的平行泊车路径规划[J]. 中国公路学报, 2016, 29(9): 143-151.
- [2] 潘姝安, 陈晨. 自动泊车用环境感知系统技术综述[J]. 中国科技信息, 2019, (13): 77-78.
- [3] C. Breck Hitz; James J. Ewing; Jeff Hecht. An Overview of Laser Technology: Introduction to Laser Technology[C]. USA: IEEE, 2012: 1-5.
- [4] 姜海娇, 来建成, 王春勇, 等. 激光雷达的测距特性及其测距精度研究[J]. 中国激光, 2011, 38(05): 234-240.
- [5] J. Zhou, L. E. Navarro-Serment and M. Hebert. Detection of parking spots using 2D range data: 15th International IEEE Conference on Intelligent Transportation Systems[C]. USA: IEEE, 2012: 1280-1287.
- [6] Jung H G , Cho Y H , Yoon P J , et al. Scanning Laser Radar-Based Target Position Designation for Parking Aid System[J]. IEEE Transactions on Intelligent Transportation Systems, 2008, 9(3): 406-424.
- [7] 史晓磊. 基于激光雷达的自动泊车系统研究[D]. 上海: 上海交通大学, 2010.
- [8] 王海. 基于激光雷达的自动泊车环境感知技术研究[D]. 大连: 大连理工大学, 2013.
- [9] Silva O D , Mann G , Gosine R G . An Ultrasonic and Vision-Based Relative Positioning Sensor for Multirobot Localization[J]. IEEE Sensors Journal, 2015, 15(3): 1716-1726.
- [10] 渠笑纳. 超声波测距在泊车辅助系统中的应用[D]. 大连: 大连理工大学, 2010.
- [11] Wan-Joo Park, Byung-Sung Kim, Dong-Eun Seo, et al. Parking space detection using ultrasonic sensor in parking assistance system,; 2008 IEEE Intelligent Vehicles Symposium[C]. USA :IEEE, 2008: 1039-1044.
- [12] R. Vishnubhotla. ZigBee based multi-level parking vacancy monitoring system: 2010 IEEE International Conference on Electro/Information Technology[C]. USA:

IEEE, 2010: 1-4.

- [13] 李坤, 陈庆胜, 仲文祥. 智能停车场车位检测系统设计与实现[J]. 长江大学学报(自科版), 2014, 11(34): 68-70+95+5.
- [14] 肖祖铭, 郭瞻. 自动泊车的超声波车位探测系统研究[J]. 计算机产品与流通, 2020, (09):146.
- [15] 孙博逊. 基于超声波测距的自动泊车辅助系统[D]. 长春: 长春理工大学, 2011.
- [16] 夏庆凯, 于大泳. 基于图像处理的工业生产手机屏幕裂纹检测[J]. 农业装备与车辆工程, 2021, 59(01): 112-115.
- [17] 黄江, 魏德奎, 秦良艳, 等. 自动泊车系统路径规划与跟踪控制方法研究[J]. 汽车技术, 2019, (08): 39-45.
- [18] R. M. Murray and S. S. Sastry. Steering nonholonomic systems using sinusoids:29th IEEE Conference on Decision and Control[C]. USA: IEEE, 1990: 2097-2101.
- [19] G. Walsh, D. Tilbury, S. Sastry, R. Murray, et al. Stabilization of trajectories for systems with nonholonomic constraints: Proceedings 1992 IEEE International Conference on Robotics and Automation[C]. USA: IEEE, 1992: 1999-2004.
- [20] J Laumond, P. E. Jacobs, M. Taix and R. M. Murray. A motion planner for nonholonomic mobile robots: IEEE Transactions on Robotics and Automation[C]. USA: IEEE, 1994: 577-593.
- [21] Paromtchik I E , Laugier C . Autonomous Parallel Parking of a Nonholonomic Vehicle[J]. Intelligent Vehicles Symposium Proceedings of the IEEE, 1997: 13-18.
- [22] I. E. Paromtchik and C. Laugier. Motion generation and control for parking an autonomous vehicle: Proceedings of IEEE International Conference on Robotics and Automation[C]. USA: IEEE, 1996: 3117-3122.
- [23] I. E. Paromtchik and C. Laugier. Automatic parallel parking and returning to traffic manoeuvres: Proceedings of the 1997 IEEE International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications[C]. USA: IEEE, 1997: V21-V23.

- [24] Laugier, C., Fraichard, T., Garnier, P. et al. Sensor-Based Control Architecture for a Car-Like Vehicle[J]. Autonomous Robots. 1999,(03): 165–185.
- [25] Paromtchik IE. Steering and velocity commands for parking assistance: Proceedings of the 10th IASTED International Conference on Robotics and Applications, Honolulu. [C]. USA. HI. 2004: 178-183.
- [26] Jiang K, Seneviratne L D. A Sensor Guided Autonomous Parking System for Nonholonomic Mobile Robots[J]. IEEE International Conference on Robotics and Automation, 1999, (1): 311-316.
- [27] Xu J, Chen G, Xie M. Vision-guided automatic parking for smart car: Intelligent Vehicles Symposium, 2000. IV 2000[C]. USA: IEEE, 2000: 725-730.
- [28] Zhao Y, Collins Jr E G. Robust automatic parallel parking in tight spaces via fuzzy logic[J]. Robotics and Autonomous Systems, 2005, 51(2): 111-127.
- [29] Z J, Ji X D, Wan K J, et al. A study of autonomous parking for a 4-wheel driven mobile robot: Control Conference, 2007[C]. USA: IEEE, 2007: 179-184.
- [30] Miah S, Gueaieb W. "Intelligent Parallel Parking of a Car-like Mobile Robot Using RFID Technology:2007 International Workshop on Robotic and Sensors Environments.[C]. USA: IEEE, 2007: 98-103.
- [31] Ho Gi Jung, Dong Suk Kim, Pal Joo Yoo, et al. Parking Slot Markings Recognition for Automatic Parking Assist System: 2006 IEEE Intelligent Vehicles Symposium, [C]. USA: IEEE, 2006: 106-113.
- [32] Jeprol, Magansal. Counting available parking space using image processing[J]. 2010, 11(2): 1-13.
- [33] Yusnita R, Norbaya F, Basharuddin N. Intelligent parking space detection system based on image processing[J]. International Journal of Innovation, Management and Technology, 2012, 3(3): 232-235.
- [34] Al-Kharusi H , Al-Bahadly I . Intelligent Parking Management System Based on Image Processing[J]. World Journal of Engineering & Technology, 2014, 02(2): 55-67.

- [35] N. Bibi, M. N. Majid, H. Dawood, et al. Automatic Parking Space Detection System: 2017 2nd International Conference on Multimedia and Image Processing[C]. USA: IEEE,2017: 11-15.
- [36] T. Féret, P. Chandrashekhariah and N. Trujillo:Real-time Parking Slot Detection for Camera-equipped Vehicles, 2019 IEEE Intelligent Transportation Systems Conference (ITSC). [C]. USA: IEEE, 2019: 4107-4114.
- [37] Zhou R F, Liu X F, Cai G P. A new geometry-based secondary path planning for automatic parking[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 1-17.
- [38] S. S. OHOL, S. KAWARE. Semi-Autonomous Parking system for Automatic Transmission vehicles[J]. IOP Conference Series: Materials Science and Engineering, 2021, 1012(1): 012-051.
- [39] 林瑞焱木, 吴志坚, 姚必正. 小车倒车问题的多维模糊控制器研究[J]. 上海大学学报(自然科学版), 1999, (04): 43-46.
- [40] 杨昔阳, 尤晴曦, 李洪兴. 基于变论域理论的自动倒车控制[J]. 北京师范大学学报:自然科学版, 2005, (04): 348-350.
- [41] 宋金泽. 自主泊车系统关键技术研究[D]. [博士学位论文]长沙: 国防科学技术大学, 2009.
- [42] 郭孔辉, 姜辉, 张建伟, 等. 基于模糊逻辑的自动平行泊车转向控制器[J]. 吉林大学学报(工学版), 2009, 39(S2): 236-240.
- [43] 姜辉, 郭孔辉, 张建伟. 基于路径规划的自动平行泊车转向控制器[J]. 吉林大学学报, 2011, 41(02): 293-297.
- [44] 周泓, 王文飞. 自动泊车系统中核心算法的研究[J]. 电路与系统学报, 2012, 17(03): 71-76.
- [45] 李红, 郭孔辉, 宋晓琳. 多约束双向自动平行泊车轨迹规划[J]. 中国机械工程, 2013, 24(16): 2135-2140.
- [46] 朴昌浩, 张乐, 张艳, 等. 非平行初始状态自动泊车轨迹生成方法[J]. 北京理工大学学报, 2014, 34(05): 506-511.

- [47] 任坤, 许艺, 丁福文, 等. 基于机器视觉和模糊控制的自动泊车[J]. 华中科技大学学报(自然科学版), 2015, 43(S1): 88-92.
- [48] 熊璐, 严森炜, 余卓平, 等. 基于库位跟踪的自动泊车决策规划系统[J]. 汽车技术, 2018, (08): 18-26.
- [49] 胡文, 谭运生, 康龙云, 等. 基于驾驶员经验的自动泊车规划算法研究[J]. 汽车工程, 2019, 41(12): 1394-1400-1415.
- [50] 魏翠芳. 基于超声波传感器的平行车位自动泊车轨迹规划[J]. 农业装备与车辆工程, 2020, 58(7): 106-109.
- [51] 姜武华, 辛鑫, 陈无畏, 等. 基于信息融合的自动泊车系统车位线车位识别和决策规划[J]. 中国机械工程, 2020, 31(10): 1190-1196.
- [52] 刘奕宏. 注重颜值 更注重品质——静态体验吉利缤越[J]. 中国汽车市场, 2018(18): 26-27.
- [53] 金娇荣. 理想 ONE 增程式电动汽车动力系统解析[J]. 汽车维护与修理, 2020, (03): 74-75.
- [54] 王华溢. 计算机视觉技术在工业领域中的应用[J]. 科技创新与应用, 2021, 11(29): 177-179.
- [55] 邹宏伟. 基于 OpenCV 的数字图像处理技术研究 with 实现[J]. 无线互联科技, 2019, 16(22): 118-119.
- [56] Suhr J K, Jung H G. Automatic parking space detection and tracking for underground and indoor environments[J]. IEEE Transactions on Industrial Electronics, 2016, 63(9): 5687-5698.
- [57] 江浩斌, 王成雨, 马世典, 等. 基于图像梯度匹配的自动泊车系统车位识别方法[J]. 江苏大学学报(自然科学版), 2020, 41(06): 621-626.
- [58] 刘阳, 彭刚, 刘诚, 等. 改进后的 Otsu 自适应背景分离的局部直方图均衡化增强算法[J]. 惠州学院学报 2021, (03): 30-35.
- [59] Otsu N . A Threshold Selection Method from Gray-Level Histograms[J]. IEEE Transactions on Systems Man & Cybernetics, 2007, 9(1): 62-66.
- [60] 范晞, 费胜巍, 储有兵. 基于 Canny 算子的改进型图像边缘提取算法[J]. 自动化

与仪表, 2019, 34(01): 41-44.

- [61] 钟志伟, 张寿明. 基于概率 Hough 变换的指针仪表去影算法[J]. 传感器与微系统, 2018, 37(06): 126-128.
- [62] 郭剑鹰, 周小兵, 管西强. 基于多视觉传感器的泊车辅助系统设计[J]. 传动技术, 2019, 33(03): 34-40.
- [63] 程志江. 遗传算法在自动泊车辅助系统模糊控制中的应用[J]. 模糊系统与数学, 2010, 24(03): 160-167. 李根. 基于 MATLAB 的模糊逻辑控制系统设计[J]. 机电信息, 2020, (06): 72-73.
- [64] 颜杰, 秦飞舟, 袁延, 等. 自提取模糊规则的小车模糊避障[J]. 工业控制计算机, 2018, 31(11): 54-56.
- [65] 万中, 梁文冬, 卢宗娟. 模糊数的隶属度区间分布函数[J]. 重庆理工大学学报(自然科学版), 2011, 25(01): 107-112.

攻读硕士学位期间发表的论文和获得的科研成果

- [1] 王靖岳, 王若羽, 王浩天. 基于遗传算法的自动泊车控制方法研究[J]. (已录用)

致 谢

时光荏苒，两年半的时间匆匆而过，非常开心可以来到沈阳理工大学继续深造学习。在沈理工，我不仅仅结识了许许多多学识广博的老师，也认识一群有着雄心壮志的同学和伙伴，使得我在短短两年多的时间内收获颇丰，受益匪浅。

首先，感谢我的老师王靖岳教授，从最开学时候就告诉我们要多看论文，在我求学过程中给与我很多帮助，培养了我坚持不懈的性格，同时老师渊博的知识，使得我更加认识到学习的重要性，相信老师身上的优良品质也将在我今后的道路上指导我一往无前。

其次，感谢和我共同生活的寝室舍友乔乐群、王濠和孙绍东，他们是我在沈理两年半时间中最重要的朋友和最亲密的伙伴，我们互帮互助互相勉励，相信彼此都会一帆风顺，同时也要感谢睢志伟同学，在我的求学道路上提供了很多帮助以及教导，我们一起学习一起讨论彼此鼓励，都使得我获得了巨大的动力。

接下来感谢我的父母以及家人，我的父亲从小教导我认真做人，我的母亲教导我认真学习，感谢它们多年的辛勤培养以及对我人生方向的指引。

最后感谢学校，感谢汽车与交通学院的各位老师提供的帮助。