

文章编号: 2095-2163(2022)10-0086-06

中图分类号: U461.1

文献标志码: A

垂直泊车路径跟踪控制研究

徐远征, 吴长水

(上海工程技术大学 机械与汽车工程学院, 上海 201620)

摘要: 自动泊车控制系统的好坏对泊车成功率具有重要影响, 本文以垂直车位为研究对象, 提出了一种基于模糊 PID 算法与模型预测控制算法(MPC)的自动泊车联合控制算法。首先建立了车辆运动学模型, 设置了泊车碰撞约束, 并在此基础上进行无碰撞路径规划; 然后建立模糊 PID 算法并调节相应参数, 控制车辆纵向速度恒定, 继而建立 MPC 算法控制车辆的横向角度。最后使用 Simulink 搭建相应的控制算法模型并与 Carsim 进行联合仿真分析。仿真结果表明: 本文提出的联合控制算法具有较高的控制精度, 其中偏航角误差控制在 0.057 m 以内, 横向偏差的绝对值控制在 0.08 m 以内, 能够满足垂直泊车对车辆控制的要求。

关键词: 垂直泊车; 模糊 PID 算法; MPC 算法; 路径跟踪控制

Research on vertical parking control and path tracking

XU Yuanzheng, WU Changshui

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

[Abstract] The quality of the automatic parking control system has an important impact on the success rate of parking. This paper takes vertical parking spaces as the research object and proposes a joint automatic parking control algorithm based on fuzzy PID algorithm and model predictive control algorithm (MPC). First, the vehicle kinematics model is established, parking collision constraints are set, and collision-free path planning is performed on this basis; then the fuzzy PID algorithm is established and the corresponding parameters are adjusted to control the longitudinal speed of the vehicle to be constant, and the MPC algorithm is established to control the lateral direction of the vehicle angle. Finally, Simulink is used to build the corresponding control algorithm model and joint simulation analysis is conducted with Carsim. The simulation results show that the joint control algorithm proposed in this paper has high control accuracy, in which the yaw angle error is controlled within 0.057 m, and the absolute value of the lateral deviation is controlled within 0.08 m, which can meet the requirements of vertical parking for vehicles control.

[Key words] vertical parking; fuzzy PID calculation; MPC algorithm; path control and tracking

0 引言

城市停车位紧张、车位空间往往较小, 泊车时所造成的剐蹭事故则不可避免地会时有发生。为提高泊车过程的安全性, 自动泊车系统已成为了当前车企及研究机构的研究重点^[1]。一套完整的自动泊车系统可分为感知、规划以及运动控制三部分, 而运动控制的好坏将直接决定最终的泊车效果及成功率, 因此对其展开研究就有着较高的应用价值^[2]。通常来说, 运动控制主要分为 2 部分: 纵向控制以及横向控制。其中, 纵向控制主要为针对车速的控制, 稳定的车速对于泊车过程至关重要。横向控制则主要通过方向盘控制车身的横向运动, 其控制精度将对路径跟踪效果产生较大影响。

目前, 已有多位学者对自动泊车运动控制算法

进行了研究。张家旭等人^[3]针对水平泊车, 设计了一种基于滑模控制的泊车路径跟踪控制方法, 仿真结果证明了所提出方法相较于传统的 PID 控制方法具有更好的抗干扰性及控制精度。黄晓峰^[4]综合考虑泊车过程中的各性能指标, 分别采用前馈加反馈控制以及模糊控制方式, 对方向盘转角和车速进行跟踪控制, 在确保泊车控制稳定的基础上保证了一定程度的舒适性。Petrov 等人^[5]则就泊车过程中转角连续抖动问题, 针对不同的泊车工况, 提出了一种基于饱和控制反馈的泊车控制方法, 并通过仿真验证了所提出方法的可行性。

本文以垂直泊车为研究对象, 提出了一种基于模糊 PID 和模型预测控制(Model Predict Control, MPC)算法的车辆联合控制跟踪算法, 并通过 Simulink 与 Carsim 进行联合仿真, 验证了所提出算

作者简介: 徐远征(1997-), 男, 硕士研究生, 主要研究方向: 汽车电子及汽车自动驾驶技术; 吴长水(1978-), 男, 博士, 副教授, 主要研究方向: 汽车电子及汽车自动驾驶技术。

通讯作者: 吴长水 Email: wuchangshui@sues.edu.cn

收稿日期: 2021-12-12

法的有效性。

1 泊车运动学模型及路径规划

1.1 车辆运动学建模

自动泊车运动过程中车速较低,此时可以忽略轮胎的侧向滑移,整个车身结构可视为刚体^[6],因此本文将车辆简化为矩形,选用车辆运动学模型进行建模。车辆运动学模型如图 1 所示。

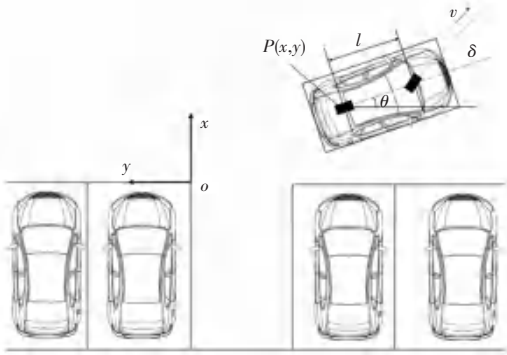


图 1 车辆运动学模型

Fig. 1 Vehicles kinematics model

由图 1 可知,本文以垂直车位左上角点为原点建立全局坐标系,此时车辆位置坐标由车辆后轴中心 P 在全局坐标系下的坐标 (x, y) 确定,车身的角度由车辆在全局坐标系下的航向角 θ 确定。本文建立的车辆运动学模型如式(1)所示:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \frac{v}{l} \tan \delta \end{cases} \quad (1)$$

其中, v 为车速; l 为轴距; δ 为车辆前轮等效转向角。研究可知只需输入车辆速度 v 及前轮的等效转向角 δ , 便可求得当前车辆在全局坐标系下的位姿 (x, y, θ) 。

1.2 垂直泊车路径规划

本文使用快速探索随机树 (Rapidly-exploring Randomized Tree, RRT) 算法^[7] 进行垂直泊车路径规划, RRT 算法是一种使用增长树的搜索算法, 其优势在于无需对泊车空间进行几何划分, 算法灵活性较高。RRT 算法伪代码详见如下。

$GENERATE_RRT(x_{init}, K, \Delta t, x_{goal})$

- 1: $T.init(x_{init}, x_{goal})$;
- 2: for $k = 1$ to K do

- 3: $x_{rand} \leftarrow RANDOM_STATE()$;
- 4: $x_{near} \leftarrow NEAREST_NEIGHBOR(x_{rand}, T)$;
- 5: $u \leftarrow SELECT_INPUT(x_{rand}, x_{near})$;
- 6: $x_{new} \leftarrow NEW_STATE(x_{near}, u, \Delta t)$;
- 7: $T.add_vertex(x_{new})$;
- 8: $T.add_edge(x_{near}, x_{new}, u)$;
- 9: RETURN T

路径规划的具体步骤为:

(1) 根据起始点 x_{init} 、目标点 x_{goal} 、循环次数 K 、拓展步长 Δt 以及环境等变量初始化各项参数。

(2) 在环境中生成随机采样点 x_{rand} , 然后遍历 RRT 树上已有的节点, 通过 $NEAREST_NEIGHBOR$ 函数选择离其最近的 RRT 树节点作为扩展节点 x_{near} 。

(3) 通过拓展函数 NEW_STATE 生成新节点 x_{new} , 并将其索引、边界添加到树中。

(4) 判断 RRT 树是否拓展到距离 x_{goal} 的某一范围内, 若是, 则结束拓展、并输出路径 T , 否则进行下一轮循环。

路径规划结束后, 往往由于路径不够平滑, 无法满足车辆运动学约束。本文使用 Reeds - Shepp (RS) 曲线进行路径优化。RS 曲线是在 Dubins 曲线上的改进、即假定车辆转向时始终按照最小转弯半径 R_{min} 进行转向, 且曲线中的各段路径均为圆弧及直线的组合。RRT 路径优化过程如图 2 所示。优化方法为: 从起始点开始, 从小到大遍历 RRT 树上点的索引, 若 2 个点间存在无碰撞的 RS 路径, 则用 RS 曲线替换原有的路径, 而后再以当前点为起始点进行新一轮的优化, 直到优化结束。

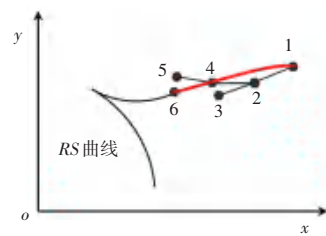


图 2 RRT 路径优化

Fig. 2 RRT path optimization

2 基于模糊 PID 算法的车辆纵向控制

2.1 模糊 PID 算法架构

模糊 PID^[8] 控制算法将传统的 PID 控制算法和模糊算法相结合, 可通过经验化的模糊规则实时调整 PID 算法的参数, 以此弥补传统 PID 算法不适用于非

线性系统的不足,本文设计的模糊PID算法框图如图3所示。由图3可知,模糊控制算法的输入为目标车速以及车辆模型返回的实际车速,输出为车辆模型所需的加速度。算法运行时,可循环比较目标车速与实际车速之间的的差值 E 以及差值的变化率 E_c ,通过模糊控制算法求得 K_p 、 K_i 、 K_d 的补偿值更新PID控制器的参数,再由PID控制算法计算出期望加速度,进而控制车辆运动。

实际车速之间的的差值 E 以及差值的变化率 E_c ,通过模糊控制算法求得 K_p 、 K_i 、 K_d 的补偿值更新PID控制器的参数,再由PID控制算法计算出期望加速度,进而控制车辆运动。

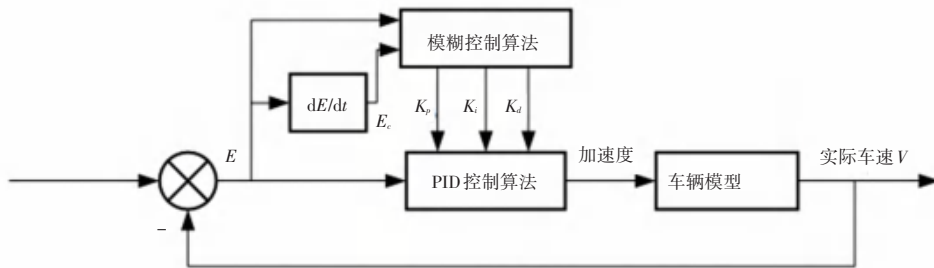


图3 模糊PID算法框图

Fig. 3 Block diagram of fuzzy PID algorithm

2.2 模糊控制算法

模糊控制器主要由模糊化、模糊推理以及清晰化三个模块组成,如图4所示。图4中,模糊化模块通过输入隶属度函数将清晰的论域转化为模糊集

合;模糊推理模块是通过设定的模糊规则推理出输出变量的隶属度;清晰化模块则根据输出变量的隶属度函数及论域进行反模糊化,计算得到输出变量的精确值。



图4 模糊控制算法框图

Fig. 4 Block diagram of fuzzy control algorithm

2.3 模糊化

本文将各参数的模糊子集定义为: {负大, 负中, 负小, 零, 正小, 正中, 正大}, 对应符号为 {NB, NM, NS, Z, PS, PM, PB}。参考实际调试过程中各参数的实际变化范围,对于输入参数,本文设定输出值与参考值之间的误差值 E 的论域为 $[-1, 1]$, 误差值变化率 E_c 的论域为 $[-1.5, 1.5]$ 。对于输出值,本文设定输出的 K_p 的论域为 $[-0.3, 0.3]$, K_i 的论域为 $[-0.06, 0.06]$, K_d 的论域为 $[-3, 3]$ 。若上述参数的实际值超过变化范围,则取边界值进行计算。

常用的隶属度函数有:三角形函数、梯形函数、高斯分布函数等。参考文献[9]的方法以及实际调试效果,本文将各变量中间的隶属度函数设定为三角形函数,两侧的隶属度函数设定为高斯分布函数,如图5所示。

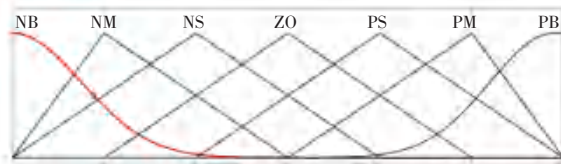


图5 各变量隶属度函数设置

Fig. 5 Membership function setting of each variable

2.4 清晰化

本文根据泊车过程的工程实践经验分别对 K_p 、 K_i 、 K_d 设置了相应的模糊规则表,根据误差及误差的变化率即可推理得到 K_p 、 K_i 、 K_d 的隶属度。

常用的清晰化方法有:最大隶属度法、重心法以及加权平均法^[9],综合考虑计算复杂度以及实际控制效果,本文采用加权平均法进行清晰化。各个输出参数的输出曲面如图6所示。

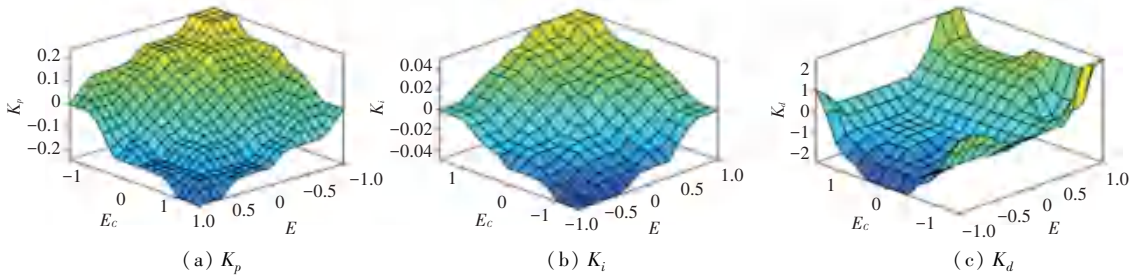


图 6 输出参数曲面图

Fig. 6 Output parameters surface diagrams

3 基于 MPC 算法的车辆横向控制

3.1 MPC 算法架构

模型预测控制算法通常可分为预测模型、滚动优化以及反馈校正三部分^[10]。本文根据垂直泊车

的横向控制需求,设计得到的 MPC 算法架构如图 7 所示。图 7 中,输入为车辆当前的位姿以及车速,输出为目标车速以及方向盘转角的控制信号,其中目标车速作为模糊 PID 算法的参考值。

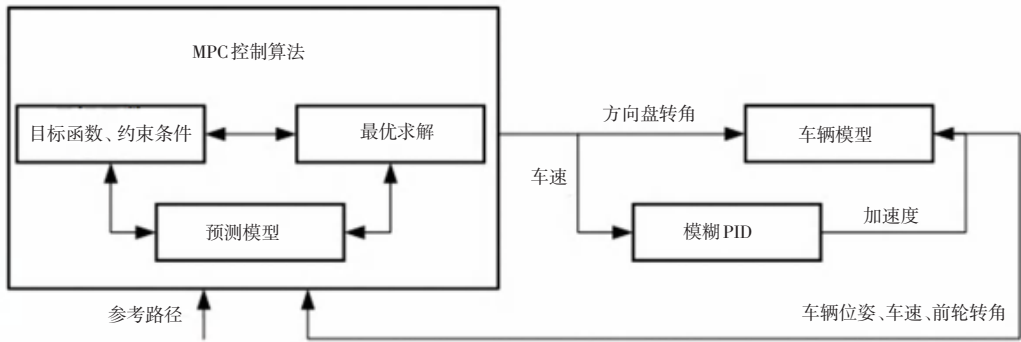


图 7 MPC 算法架构

Fig. 7 MPC algorithm architecture

3.2 预测模型

低速状态下轮胎滑移量可忽略,本文使用车辆低速运动学模型作为预测模型。由于上文建立的低速车辆运动学模型为非线性时变系统,无法确保控制系统的实时性,因此本文对其进行线性化处理。

在控制车辆横向运动时需跟踪车辆的实时位姿,本文选取系统的输入为 $u(v, \delta)$, 将系统的状态变量用 $\chi(x, y, \theta)$ 表示。根据路径规划的任意参考点的参考状态 $\chi_r(x_r, y_r, \theta_r)$ 以及对应的参考输入 $u_r(v_r, \delta_r)$, 推导可得到以 T 为采样时间的离散化误差模型,推导得到的该模型的数学表述见式(2):

$$\begin{aligned} \tilde{\chi}(k+1) &= A_{k,t} \tilde{\chi}(k) + B_{k,t} \tilde{u}(k) \\ \tilde{A}_{k,t} &= \begin{bmatrix} \hat{e}^1 & 0 & -v_r \sin \theta_r T \\ \hat{e}^0 & 1 & v_r \cos \theta_r T \\ \hat{e}^0 & 0 & 1 \end{bmatrix} \\ \tilde{B}_{k,t} &= \begin{bmatrix} \hat{e}^1 \cos \theta_r T & 0 \\ \hat{e}^1 \sin \theta_r T & 0 \\ \hat{e}^1 \tan \delta_r T & v_r T \\ \hat{e}^1 & l \end{bmatrix} \end{aligned} \quad (2)$$

量进行增量约束,取 $\xi(k|t) = \begin{bmatrix} \hat{e}^1 \tilde{\chi}(k|t) \\ \hat{e}^0 \tilde{u}(k-1|t) \end{bmatrix}$, 结

合式(2)可推得新的状态空间方程为:

$$\begin{cases} \xi(k+1|t) = \tilde{A}_{k,t} \xi(k|t) + \tilde{B}_{k,t} \Delta U(k|t) \\ \eta(k|t) = \tilde{C}_{k,t} \xi(k|t) \end{cases} \quad (3)$$

其中, $\tilde{A}_{k,t} = \begin{bmatrix} \hat{e}^1 A_{k,t} & B_{k,t} \\ \hat{e}^0 I_{m \times n} & I_m \end{bmatrix}$; $\tilde{B}_{k,t} = [B_{k,t} \quad I_m]^T$;

$\tilde{C}_{k,t} = \begin{bmatrix} \hat{e}^1 C & 0_{n \times m} \\ \hat{e}^0 I_{m \times n} & 0_{m \times m} \end{bmatrix}$; $C = \text{diag}(1 \quad 1 \quad 1)$; m 和 n 分别为控制变量和状态变量的维度; η 为状态变量。

为提高模型效率,令 $A_{k,t} = A_{t,t}$, $B_{k,t} = B_{t,t}$, 得到的预测模型可根据 t 时刻的系统状态以及 N_c 各时域的预测输入预测 N_p 各时域内的输出。此时推得的数学公式为:

$$Y(t) = \psi(t) \xi(t|t) + \Theta_t \Delta U(t) \quad (4)$$

其中,

为防止横向控制的控制量波动过大,需对控制

$$Y(t) = \begin{bmatrix} \hat{e} \eta(t+1|t) \\ \hat{e} \eta(t+2|t) \\ \dots \\ \hat{e} \eta(t+N_c|t) \\ \dots \\ \hat{e} \eta(t+N_p|t) \end{bmatrix}, \psi(t) = \begin{bmatrix} \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t} \dot{u} \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^2 \dot{u} \\ \dots \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_c} \dot{u} \\ \dots \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_p} \dot{u} \end{bmatrix}$$

$$\Delta U(t) = \begin{bmatrix} \hat{e} \Delta u(t|t) \\ \hat{e} \Delta u(t+1|t) \\ \hat{e} \dots \\ \hat{e} \Delta u(t+N_c|t) \end{bmatrix}$$

$$\Theta_t = \begin{bmatrix} \hat{e} \tilde{C}_{t,t} \tilde{B}_{t,t} & 0 & 0 & 0 \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t} \tilde{B}_{t,t} & \tilde{C}_{t,t} \tilde{B}_{t,t} & 0 & 0 \\ \hat{e} \dots & \dots & \ddots & \dots \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_c-1} \tilde{B}_{t,t} & \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_c-2} \tilde{B}_{t,t} & \dots & \tilde{C}_{t,t} \tilde{B}_{t,t} \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_c} \tilde{B}_{t,t} & \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_c-1} \tilde{B}_{t,t} & \dots & \tilde{C}_{t,t} \tilde{A}_{t,t} \tilde{B}_{t,t} \\ \hat{e} \vdots & \vdots & \ddots & \vdots \\ \hat{e} \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_p-1} \tilde{B}_{t,t} & \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_p-2} \tilde{B}_{t,t} & \dots & \tilde{C}_{t,t} \tilde{A}_{t,t}^{N_p-N_c-1} \tilde{B}_{t,t} \end{bmatrix}$$

3.3 目标函数及约束条件

为使控制过程更为平稳,本文使用目标函数为:

$$J(k) = \sum_{i=1}^{N_p} \|\eta(k+i|t) - \eta_{ref}(k+i|t)\|_Q^2 + \sum_{i=1}^{N_c-1} \|\Delta U(k+i|t)\|_R^2 + \rho \varepsilon^2 \quad (5)$$

其中, N_p 为预测步长; N_c 为控制步长; ρ 为权重系数; ε 为松弛因子; Q 和 R 分别为状态量和控制量的权重。

设定采样时间 $T = 0.02 \text{ s}$, 为了使泊车期间车辆速度及航向角变化平稳, 本文设定的速度及前轮转角的控制量及控制增量约束如式(6)所示:

$$\begin{cases} -2 \text{ km/h} \leq v \leq 2 \text{ km/h} \\ -0.18 \text{ km/h} \leq \Delta v \leq 0.18 \text{ km/h} \\ -33^\circ \leq \delta \leq 33^\circ \\ -0.48^\circ \leq \dot{\delta} \leq 0.48^\circ \end{cases} \quad (6)$$

4 仿真实验验证

4.1 仿真参数设置

本文在 Matlab/Simulink 搭建相应的算法, 并在 Carsim 中配置相应的车辆参数, 实现 Carsim 与 Simulink 的联合仿真。本文搭建的仿真环境如图 8 所示, 所用的仿真参数见表 1。

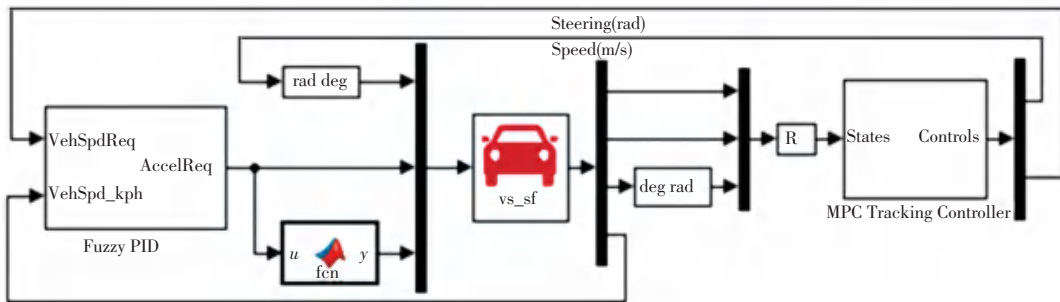


图 8 联合仿真模型

Fig. 8 Co-simulation model

表 1 仿真参数

Tab. 1 Simulation parameters

参数	车长/m	车宽/m	轴距/m	前悬/m	最大方向盘 转角/(°)	车位 长度/m	车位 宽度/m	车辆起始 位置/m	车辆起始 偏航角/rad
值	4.6	1.8	2.7	1	450	5.9	3	(3,-1)	-0.17

4.2 实验结果分析

垂直泊车路径跟踪控制的整体效果如图 9 所示。图 9 中, 实线部分为车辆实际运动轨迹。图 10 显示了车辆各执行器的相应情况。泊车开始后, 车

辆首先挂 D 档, 向右打方向盘行驶到点 2, 在此过程中车速缓慢上升。然后向左打方向盘行驶到点 3, 此时车速缓慢下降至 0 km/h。之后车辆挂 R 档, 向右打方向盘开始倒车, 行驶到点 4 后回正方向盘, 并

适时调整其角度,使车辆沿直线泊入车位至缓慢停止。最后回正方向盘,档位切换为 P 档,泊车结束。

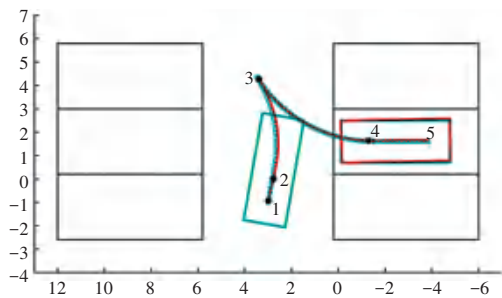


图 9 规划路径与实际路径对比

Fig. 9 Comparison of planned path and actual path

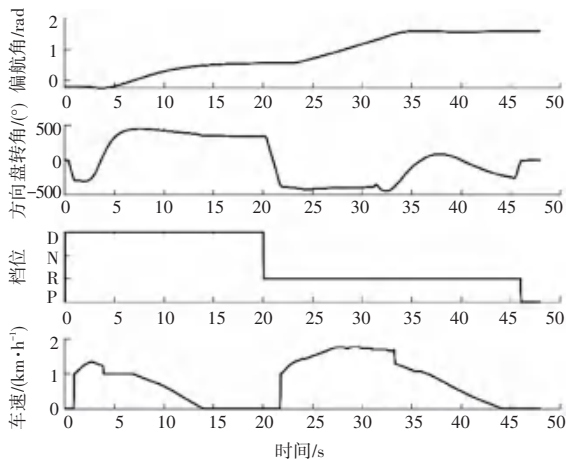


图 10 执行器响应

Fig. 10 Actuators response

综上所述,实际路径与规划路径较为吻合且变化趋势一致,车辆最终的泊车位姿与目标位姿偏差不大,且在泊车过程中没有碰撞、压线等情况的发生。整个泊车过程中方向盘转角不超过 450° ,满足车辆转向约束。车辆车速变化较为平缓,车速控制在 2 km/h 以内,验证了模糊 PID 算法的有效性。故本文所提出的垂直泊车跟踪控制算法能够对规划的路径进行较好的跟踪,并控制车辆以较小的误差按照该路径泊入目标车位。

图 11 为规划车辆位姿与实际车辆位姿的误差。由图 11 可知,车辆实际的偏航角与路径规划的偏航角误差的绝对值控制在 0.057 m 以内,规划路径与实际路径的横向偏差的绝对值控制在 0.08 m 以内。分析图 9 可知,上述误差并不会使车辆与外界环境发生碰撞,考虑到 Carsim 车辆模型模拟了真实的泊车环境,在泊车过程中存在较多的环境干扰,上述误差范围区间较为合理。因此,本文建立的垂直泊车路径跟踪控制算法具有良好的跟踪控制效果。

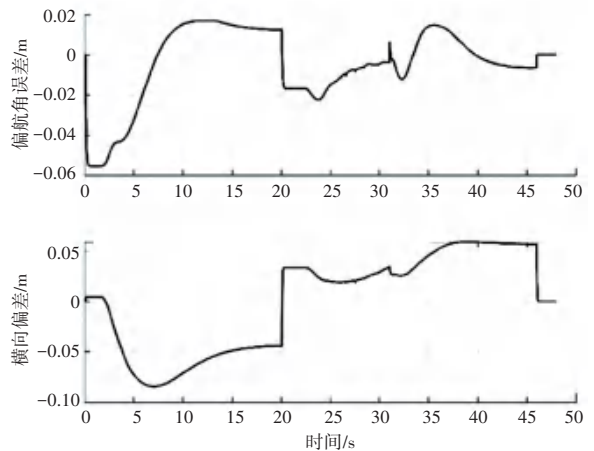


图 11 偏航角、横向角误差

Fig. 11 Error of yaw angle and lateral angle

5 结束语

垂直泊车车位狭小,易造成刚蹭碰撞等危险情况的发生,往往对车辆控制精度具有较高的要求。本文以垂直泊车为研究对象,提出了一种基于模糊 PID 算法与模型预测控制算法的自动泊车联合控制算法。联合仿真结果表明:本文提出的跟踪控制算法精度较高,可以控制车辆在允许的误差范围内沿规划的路径行驶。此外,该方法对水平泊车、斜列泊车也都具有一定的参考价值。

参考文献

- [1] 朱旺旺,黄宏成,马晋兴. 基于图像识别的泊车车位检测算法研究[J]. 汽车工程,2019,41(07):744-749,756.
- [2] MA Shidan, JIANG Haobin, HAN Mu, et al. Research on automatic parking systems based on parking scene recognition[J]. IEEE Access, 2017,5:21901-21917.
- [3] 张家旭,王晨,赵健,等. 面向狭小平行泊车位的规划与跟踪控制[J]. 吉林大学学报(工学版),2021,51(05):1879-1886.
- [4] 黄晓峰. 自动泊车系统最优轨迹决策及控制算法研究[D]. 长春:吉林大学,2018.
- [5] PETROV P, NASHASHIBI F. Saturated feedback control for an automated parallel parking assist system [C]//2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). Singapore: IEEE, 2014:577-582.
- [6] 文琼. 基于 RRT 的无人驾驶车辆路径规划方法研究[D]. 沈阳:东北大学,2019.
- [7] LAVALLE S M, KUFFNER J J. Randomized kinodynamic planning [J]. The International Journal of Robotics Research, 2001, 20(5): 378-400.
- [8] 陈政和. 基于 Fuzzy-PID 的自动泊车车速控制系统的研究[D]. 广州:华南理工大学,2018.
- [9] 杨萍萍,马亮. 基于模糊 PID 的汽车空调控制器设计[J]. 自动化仪表, 2021, 42(09): 39-44.
- [10] 王雪霖. 自动驾驶车辆垂直车位自动泊车控制算法研究[D]. 长春:吉林大学,2021.