

结合PCA与K-Means降维的Fast-MPC无人驾驶实时轨迹跟踪

耿战龙¹, 叶洪涛²

¹广西科技大学自动化学院, 广西 柳州

²广西汽车零部件与整车技术重点实验室(广西科技大学), 广西 柳州

收稿日期: 2026年3月16日; 录用日期: 2026年4月9日; 发布日期: 2026年4月15日

摘要

本文提出了一种基于快速模型预测控制(Fast-MPC)的方法, 旨在解决无人驾驶车辆在全速域轨迹跟踪中的二次规划(QP)问题。传统的模型预测控制(MPC)方法在高维不等式约束下的计算效率较低, 尤其在无人驾驶车辆的高速轨迹跟踪任务中, 难以满足实时性要求。尽管Fast-MPC显著加快了求解速度, 但在处理高维不等式约束时表现出敏感性, 导致计算效率下降。为了解决这一问题, 本文应用了一种基于主成分分析(PCA)和K-means聚类的不等式降维方法。在PCA的基础上, 本文进一步引入K-means聚类算法对降维后的数据进行聚类, 从而在降低问题维度的同时, 保留了不同约束类群中的关键信息。从而降低了QP问题的复杂度。该算法在Simulink和CarSim联合仿真平台上进行了双移线场景测试。实验结果表明, 结合PCA的Fast-MPC方法不仅有效提高了求解速度, 最大减少了约束维度的60%, 并在高速轨迹跟踪中有效避免了振荡现象。同时, 车辆在高动态环境中的轨迹跟踪精度得到了显著改善, 尤其是在横向控制方面的表现更加稳定可靠。该方法为无人驾驶车辆的实时轨迹跟踪提供了高效且稳定的解决方案。

关键词

无人驾驶, 轨迹跟踪, 快速模型预测控制, 主成分分析, K均值算法

Fast-MPC with PCA and K-Means Dimensionality Reduction for Real-Time Trajectory Tracking in Autonomous Vehicles

Zhanlong Geng¹, Hongtao Ye²

¹School of Automation, Guangxi University of Science and Technology, Liuzhou Guangxi

²Guangxi Key Laboratory of Automobile Components and Vehicle Technology (Guangxi University of Science and Technology), Liuzhou Guangxi

Received: March 16, 2026; accepted: April 9, 2026; published: April 15, 2026

文章引用: 耿战龙, 叶洪涛. 结合PCA与K-Means降维的Fast-MPC无人驾驶实时轨迹跟踪[J]. 建模与仿真, 2026, 15(4): 105-120. DOI: 10.12677/mos.2026.154057

Abstract

This study proposes a fast model predictive control (Fast-MPC)-based method with dimensionality reduction to address quadratic programming (QP) challenges in real-time trajectory tracking for autonomous vehicles. Traditional model predictive control (MPC) suffers from low computational efficiency under high-dimensional inequality constraints, making it difficult to meet real-time requirements, especially in high-speed trajectory tracking tasks. Although Fast-MPC significantly accelerates the optimization process, its sensitivity to high-dimensional inequality constraints may cause computational efficiency degradation and solution instability. To overcome these limitations, this paper introduces an inequality dimension-reduction method that integrates principal component analysis (PCA) and k-means clustering algorithm (K-means). By employing dynamic grouping and an optimized constraint retention strategy, the proposed approach effectively reduces QP complexity while significantly enhancing computational efficiency. Compared to conventional dimension-reduction techniques, the proposed algorithm achieves a 60% dimensionality reduction in constraint space and avoids trajectory oscillations through more accurate feature preservation. Comprehensive co-simulations using Simulink/CarSim platforms under double-lane-change scenarios validate that the algorithm notably improves trajectory tracking accuracy, particularly showcasing stable and reliable performance in high-dynamic lateral stabilization. Further analysis reveals that this method exhibits strong adaptability, making it extendable to dynamic obstacle avoidance and real-time trajectory tracking in complex urban environments. This work provides an efficient and robust solution for autonomous vehicle trajectory tracking, demonstrating strong potential for deployment in complex driving scenarios with dynamic obstacles.

Keywords

Autonomous Driving, Trajectory Tracking, Fast Model Predictive Control, Principal Component Analysis, K-Means Clustering

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着学习算法和硬件发展的最新进展,自动驾驶汽车在良好驾驶条件下的结构化环境中表现出了潜力[1]。在复杂和高不确定性环境下,自动驾驶系统仍表现出较多不足。这些系统在应对突发事件和高度动态的驾驶环境时,仍可能导致潜在危险。自动驾驶车辆通常可以模块化为环境感知、路径规划和轨迹跟踪。轨迹跟踪是自动驾驶车辆的基本组成部分,它有效而稳定地控制自动驾驶车辆,以跟踪由路径规划模块预先确定的参考轨迹[2]。轨迹跟踪的性能直接决定了自动驾驶车辆的表现,涉及驾驶安全、乘客舒适度、出行效率和能源消耗。自动驾驶车辆的轨迹跟踪控制是一个具有挑战性的研究领域,因为这些系统通常是带有非完整约束的非线性系统[3]。在自动驾驶系统中,路径跟踪层定义了跟随参考路径和速度轮廓的执行器命令。模型预测控制(MPC)因其管理多变量问题的能力,系统地考虑状态和控制动作的约束,以及预测系统未来行为的能力而广泛用于轨迹跟踪[4]。然而,随着轨迹跟踪精度和车辆速度的提升,MPC算法的计算复杂性显著增加,这在一定程度上限制了其应用场景。尤其是在高速行驶下,该方法在实时性方面仍面临挑战[5]。自动驾驶车辆的轨迹跟踪控制是一个具有挑战性的研究领域,因为这些系统通常是具有非完整约束的非线性系统[6]。尤其是,传统的MPC方法在面对高维不等式约束时,计算效率

低下, 难以满足实时控制的要求。因此, 需要针对高速动态环境下的轨迹跟踪提出更加高效的解决方案。对于模型预测控制而言, 随着车辆动力学模型和约束的维度增加其优化求解的计算量增大, 并且无人驾驶车辆的速度提升也会进一步增加优化求解的难度, 因此计算的实时性是 MPC 在运动控制上应用的主要技术瓶颈之一[7]。如何在保证计算效率的同时提升控制精度仍是未解决的挑战。

Domahidi 等人[8]提出了一种高效的内点方法, 专门用于多级凸优化问题, 并展示了其在 MPC 中的应用。然而, 尽管该方法在某些情况下提高了计算效率, 但在面对高维度、不等式约束较多的复杂问题时, 其求解时间仍然较长, 难以满足实时控制需求。Ferreau 等人[9]开发了一种在线有效集策略, 能够有效地解决显式 MPC 中的参数化问题, 并提高了求解速度。然而, 该策略的局限性在于其依赖于先前 QP 解的假设, 当系统出现显著变化时, 策略的有效性可能受到影响, 这在动态环境下尤为突出。因此, 该方法在处理高速动态轨迹跟踪时, 可能无法充分应对所有场景变化。单睿斌等人[10]提出了一种基于 ADMM 算法的 MPC 求解方法, 通过引入辅助变量, 将 QP 问题转化为可分结构的优化问题。这种方法在某些场景下提升了求解速度。然而, 随着预测时域的增加, 该算法的计算时间也会显著增加, 尤其在高速动态环境中, 求解效率和结果的鲁棒性可能不足。因此, 进一步优化 ADMM 算法以适应更复杂的轨迹跟踪任务仍是未来的研究方向。Li 等人[11]提出了 MPC 的快速在线计算方法, 只有第一时间步的约束直接影响实际的控制输入。因此, 可以在预测范围内的某些选定步骤中停用约束, 以减少约束集的维度, 从而降低计算强度。但面对高速的两个较近连续转弯情况由于只考虑第一时间步的约束可能大致转弯角度过大, 导致后面更大的跟踪误差。Marcucc 和 Tedrake [12]研究了混合整数 MPC 问题的热启动策略, 通过重用先前求解中的部分计算结果, 显著减少了计算时间。然而, 该策略在面对高速复杂轨迹跟踪任务时, 其求解速度和精度可能受到限制, 特别是在环境剧烈变化时, 热启动策略的效果会有所下降。但是这种方法仍然是一种很有效的方法, 只要合理利用可以和别的方法相结合使用。Salzmann 等人[13]提出了通过分离来自 MPC 的计算量大的数据驱动模型优化, 允许无约束模型架构选择, 嵌入式实时更大模型的能力和 GPU 加速, 无需性能下降。Li 等人[14]提出了将 MPC 算法部署到 FPGA 硬件上的加速方法, 以提高自动驾驶控制的实时性。然而, 尽管硬件加速显著提升了计算速度, FPGA 和 GPU 的资源有限性使得该方法在处理复杂、高维度的轨迹跟踪任务时仍有局限。因此, 如何在资源有限的硬件平台上有效平衡计算速度和控制精度仍是一个需要深入研究的问题。

尽管传统的 MPC 方法广泛应用于无人驾驶车辆的轨迹跟踪中, 但其在高速动态环境下暴露出计算效率和实时性能的不足。高速轨迹跟踪要求引入大量不等式约束, 以确保系统的稳定性和安全性。然而, 这些不等式约束在高维空间中的处理极为复杂, 显著增加了优化求解的计算负担, 影响了系统的实时性与控制效能。为应对这一挑战, 本文提出了一种基于 Fast-MPC 的改进方法, 并结合 PCA 与 K-means 聚类进行降维。首先, 通过 PCA 减少不等式约束的维数, 从而降低 QP 问题的复杂度。其次, 利用 K-means 聚类对 PCA 降维后的数据进行进一步分类与聚合, 从而保留数据中的关键特征并消除冗余信息。这种双重降维策略显著减少了计算负担, 同时确保了不等式约束的有效性。

2. 预测模型建立

2.1. 车辆动力学模型建模

本文主要研究目标是使车辆快速而稳定的跟踪给定的期望路径, 属于车辆操纵稳定性范畴, 因此选用车辆轮胎 - 模型来分析车辆纵向及侧向动力学特性[15]。

车辆横摆动力学模型常用于车辆路径跟踪控制。考虑道路曲率影响时, 如图 1 所示, 建立大地坐标系 (X, Y) 和车辆坐标系 (x, y) 。其中, (x_c, y_c) 为车辆质心坐标, v_x 和 v_y 分别为纵向速度和横向速度, v_f 为前轮轮速, φ 为航向角, δ_f 为前轮转角, α_f 和 α_r 为前后轮侧偏角, α_{v_f} 为前轮速度方向角, a 和 b 分

别为质心到前后轴的距离, F_{yf} 和 F_{yr} 分别为前后轮侧偏力。

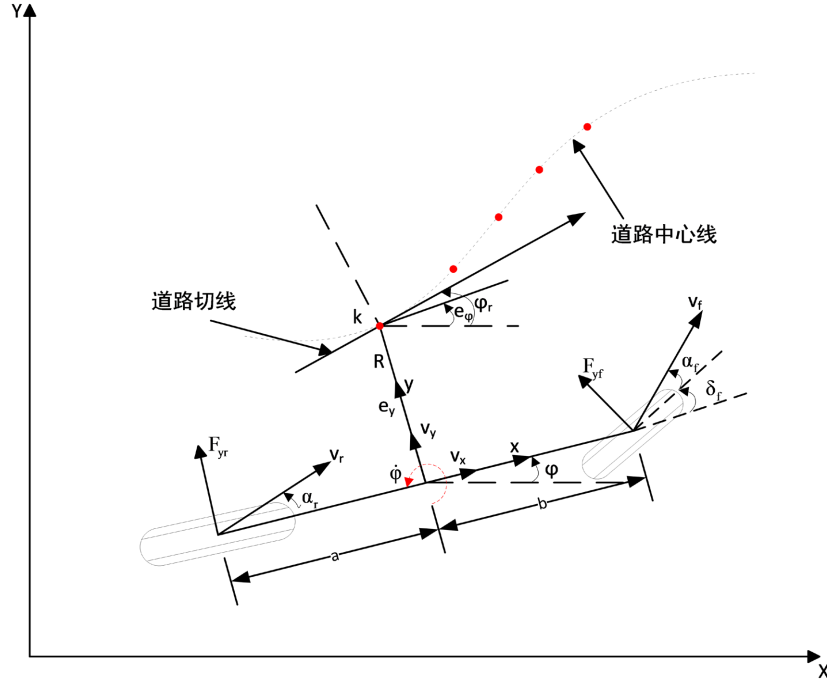


Figure 1. Vehicle linear two-degree-of-freedom dynamic model
图 1. 车辆线性二自由度动力学模型

本研究采用 Pacejka 提出的以魔术公式(Magic Formula, MF)轮胎模型来表征这些动态相互作用。在线性侧偏角区间内($|\alpha| < 5^\circ$), 侧向力与侧偏角呈线性正比关系[6]。

$$F_y = C_\alpha \alpha \tag{1}$$

本式中 C_α 为轮胎线性侧偏刚度。

根据横摆动力学模型与轮胎模型, 对前轮转角 δ_f 做小角度假设可以得到考虑道路曲率的车辆动力学模型:

$$\begin{aligned} \dot{\xi} &= A(t)\xi(t) + B(t)u(t) \\ Y &= C(t)\xi(t) \end{aligned} \tag{2}$$

$$A(t) = \frac{\partial f}{\partial \xi} \Big|_{\xi_r, u_r} \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & 0 & a_{24} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ a_{41} & a_{42} & 0 & a_{44} & 0 & 0 \\ a_{51} & a_{51} & a_{53} & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & 0 & 0 & 0 \end{bmatrix}$$

$$B(t) = \frac{\partial f}{\partial u} \Big|_{\xi_r, u_r} [b_{11} \quad b_{12} \quad 0 \quad b_{14} \quad 0 \quad 0] \tag{3}$$

$$C(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

式中, 状态量为 $\xi = [\dot{y}, \dot{x}, \varphi, \dot{\varphi}, Y, X]^T$, 控制变量取为 $\mu = \delta_f$, $a_{11} = -2(C_{cf} + C_{cr})/(m\dot{x}_t)$, $a_{12} = [2C_{cf}(\dot{y}_t + a\dot{\varphi}_t) + 2C_{cr}(\dot{y}_t - b\dot{\varphi}_t)]/(m\dot{x}_t^2 - \dot{\varphi}_t)$, $a_{14} = -\dot{x}_t + 2(bC_{cr} - aC_{cf})/(m\dot{x}_t)$, $a_{21} = \dot{\varphi} - 2C_{cf}\delta_{f,t-1}/(m\dot{x}_t)$, $a_{41} = -2(bC_{cr} - aC_{cf})/(I_z\dot{x}_t)$, $a_{63} = -\dot{x}\cos(\varphi_t) - \dot{y}_t\sin(\varphi_t)$, $a_{22} = 2[C_{cf}\delta_{f,t-1}(\dot{y}_t + a\dot{\varphi}_t)]/(m\dot{x}_t^2)$, $a_{44} = -2(a^2C_{cf} + b^2C_{cr})/(I_z\dot{x}_t)$, $a_{52} = \sin(\varphi_t)$, $a_{24} = \dot{y}_t - 2aC_{cf}\delta_{f,t-1}/(m\dot{x}_t)$, $a_{61} = -\sin(\varphi_t)$, $a_{62} = \cos(\varphi_t)$, $b_{14} = 2aC_{cf}/I_z$, $b_{11} = 2C_{cf}/m$, $b_{12} = 2C_{cf}[2\delta_{f,t-1} - (\dot{y}_t + a\dot{\varphi}_t)/x_t]/m$, 采用一阶差商的方法进行离散化处理, 得到离散的状态方程:

$$\xi(k+1) = A(k)\xi(k) + B(k)u(k) \quad (4)$$

式中, $C(k) = TC(t)$, $A(k) = I + TA(t)$, $B(k) = TB(t)$ 。

2.2. 模型预测控制

用控制增量取代控制量并且加入松弛因子将式(上一个)(6), 得到新的状态空间表达式:

$$\begin{aligned} \xi(k+1|t) &= \tilde{A}_{k,t}\xi(k|t) + \tilde{B}_{k,t}\Delta U(k|t) \\ \eta(k|t) &= \tilde{C}_{k,t}\xi(k|t) \end{aligned} \quad (5)$$

式中, $\tilde{A}_{k,t} = \begin{bmatrix} A_{k,t} & B_{k,t} \\ 0_{m \times n} & I_m \end{bmatrix}$, $\tilde{B}_{k,t} = \begin{bmatrix} B_{k,t} \\ I_m \end{bmatrix}$, $\xi(k|t) = \begin{bmatrix} \tilde{X}(k|t) \\ \tilde{u}(k-1|t) \end{bmatrix}$ 。 n 状态变量维度, m 控制变量维度。

经过推导, 可以得到系统的预测输出表达式:

$$Y(t) = \Psi_t \xi(t|t) + \Theta_t \Delta U(t) \quad (6)$$

$$Y(t) = \begin{bmatrix} \eta(t+1|t) \\ \eta(t+2|t) \\ \vdots \\ \eta(t+N_c|t) \\ \vdots \\ \eta(t+N_p|t) \end{bmatrix} \Psi_t = \begin{bmatrix} \tilde{C}_{k,t} \tilde{A}_{k,t} \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^2 \\ \vdots \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c} \\ \vdots \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_p} \end{bmatrix} \Delta U(t) = \begin{bmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \vdots \\ \Delta u(t+N_c|t) \end{bmatrix} \quad (7)$$

$$\theta_t = \begin{bmatrix} \tilde{C}_{k,t} \tilde{B}_{k,t} & 0 & 0 & 0 \\ \tilde{C}_{k,t} \tilde{A}_{k,t} \tilde{B}_{k,t} & \tilde{C}_{k,t} \tilde{B}_{k,t} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-1} \tilde{B}_{k,t} & \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-2} \tilde{B}_{k,t} & \cdots & \tilde{C}_{k,t} \tilde{A}_{k,t} \tilde{B}_{k,t} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-1} \tilde{B}_{k,t} & \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-2} \tilde{B}_{k,t} & \cdots & \tilde{C}_{k,t} \tilde{A}_{k,t} \tilde{B}_{k,t} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-1} \tilde{B}_{k,t} & \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-2} \tilde{B}_{k,t} & \cdots & \tilde{C}_{k,t} \tilde{A}_{k,t}^{N_c-1} \tilde{B}_{k,t} \end{bmatrix} \quad (8)$$

目标函数要能够保证无人驾驶车辆快速且平稳地追踪上期望轨迹, 因此需要加入对系统状态量的偏差和控制量的优化。在设计轨迹跟踪控制器时, 采用如下新形式的目标函数:

$$J(\xi(t), u(t-1), \Delta U(t)) = \sum_{i=1}^{N_p} \|\eta(t+i|t) - \eta_{ref}(t+i|t)\|_Q^2 + \sum_{i=1}^{N_c-1} \|\Delta u(t+i|t)\|_R^2 + p\varepsilon^2 \quad (9)$$

式中, Q 和 R 分别是系统输出量, 控制增量, 权重系统矩阵。

在车辆轨迹跟踪过程中, 为实现高效、安全且平稳的控制, 需要综合考虑车辆动力学特性、执行机构约束以及乘坐舒适性, 因此有必要对控制量及其变化率进行限制。尤其在高速行驶条件下, 车辆紧急转向时轮胎附着力容易接近饱和, 导致侧偏力达到极限。

变量为控制时域内的控制增量, 约束条件中也只能以控制增量或者控制增量与转换矩阵的相乘的形式表示, 设置约束条件为:

$$\begin{cases} \Delta U_{\min} \leq \Delta U_t \leq \Delta U_{\max} \\ U_{\min} \leq A\Delta U_t + U_t \leq U_{\max} \\ y_{hc,\min} \leq y_{hc} \leq y_{hc,\max} \\ y_{sc,\min} - \varepsilon \leq y_{sc} \leq y_{sc,\max} + \varepsilon \\ \varepsilon > 0 \end{cases} \quad A = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \otimes I_m \quad (10)$$

式中, y_{hc} 为硬约束输出; y_{sc} 为软约束输出; $y_{hc,\min}$ 和 $y_{hc,\max}$ 为硬约束极限值; $y_{sc,\min}$ 和 $y_{sc,\max}$ 为硬约束极限值; 综合上面的目标函数和约束条件, 将所描述的问题转换为标准的 QP 问题:

$$\begin{aligned} \min J &= \frac{1}{2} \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix}^T H_t \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix} + G_t \begin{bmatrix} \Delta U(t) \\ \varepsilon \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} \Delta U_{\min} \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U_{\max} \\ \varepsilon \end{bmatrix} \end{aligned} \quad (11)$$

$$\begin{bmatrix} \theta_{hc} & 0 \\ -\theta_{hc} & 0 \\ \theta_{sc} & -1 \\ -\theta_{sc} & 1 \\ A & 0 \\ -A & 0 \end{bmatrix} \begin{bmatrix} \Delta U \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} Y_{hc,\max}(t) - \Psi(t)\xi(t) - E(t)D(t) \\ Y_{hc,\min}(t) + \Psi(t)\xi(t) + E(t)D(t) \\ Y_{sc,\max}(t) - \Psi(t)\xi(t) - E(t)D(t) \\ Y_{sc,\min}(t) + \Psi(t)\xi(t) + E(t)D(t) \\ U(t)_{\max} - U(t) \\ U(t) - U(t)_{\min} \end{bmatrix} \quad (12)$$

式中, $Y_{ref} = [\eta_{ref}(t) \ \dots \ \eta_{ref}(t + N_p)]$, $E(t) = \psi\xi(t) + E(t)D(t) - Y_{ref}(t)$, $G_t = [2E(t)^T Q\theta_t \ 0]$, $H_t = \begin{bmatrix} \theta_t^T Q\theta_t & 0 \\ 0 & \rho \end{bmatrix}$ 。

3. Fast MPC

在本节中, 描述了求解 QP 问题的一个基本的原始屏障内法点, 利用了它的特殊结构[16]。

首先以更紧凑的形式重写 QP 问题, 定义一个整体优化变量:

$$\text{minimize } x^T Hx + g^T x \quad \text{subject to } Px \leq h \quad (13)$$

3.1. 不可行的初始牛顿法

将使用不可行的启动原始屏障方法来求解 QP 问题。用目标中的障碍项代替 QP 式(17)中的不等式约束, 得到近似问题:

$$\text{minimize } x^T Hx + g^T x + k\phi(x) \quad (14)$$

式中, $k > 0$ 为障碍参数, 并且 ϕ 为对数垒不等式约束, 表达式:

$$\phi(x) = \sum_{i=1}^{IT+k} -\log(h_i - p_i^T x) \quad (15)$$

式中, $p_1^T, \dots, p_{l^*}^T$ 为 P 的行向量, (当 $Px < h$ 取 $\phi(x) = \infty$), 问题(18)是一个目标光滑, 约束线性相等的凸优化问题, 可以用牛顿法来求解。

使用不可行初始牛顿法求解问题(18)。将对偶变量联系起来。优化条件表达式:

$$r_d = 2Hx + g + \kappa P^T d = 0 \quad (16)$$

式中, $d_i = 1/(h_i - p_i^T x)$, p_i^T 为 P 的第 i 个行向量, 式中 kP^T 记为 $K\phi(x)$ 的导数, r_d 为偶数残差, 优化(16)的条件就是 $r_d = 0$ 。

在每一步都保持近似值 $x(s)$, 如果残差 r_d 足够小就退出更新。否则, 通过线性最优条件(17)并计算线性化近似中 $x + \Delta x$ 给出零残差的原步和对偶步 Δx 来改进估计。原始和对偶搜索步骤和通过求解线性方程得到:

$$\left(2H + kP^T \text{diag}(d)^2 P\right) \Delta x = -r_d \quad (17)$$

式中, $kP^T \text{diag}(d)^2 P$ 是 $k\phi(x)$ 的 Hessian 矩阵。一旦计算出 Δx , 使用残差 r_d 范数的回溯线搜索来找到一个步长 $s \in (0, 1)$, 以确保 $(Px < h)$ 更新点保持不变。然后更新原始变量: $x = x + s\Delta x$ 。重复这个过程, 直到残差的范数低于可接受的阈值。

可以证明, 问题(16), 在迭代的过程中是可实现的。当得到 $r_d = 0$, 它将在后面的迭代中保持为零, 并且会收敛到一个最优点 x 。计算精确解所需的牛顿步数取决于初始点 x^0 。

3.2. 牛顿步的快速计算

可以利用问题中的结构求解(上一个), 将使用块消除。定义 $\Phi = 2H + kP^T \text{diag}(d)^2 P$, 这是一个块对角线矩阵[17]。

- 1) 通过求解 $\Phi \Delta x = -r_d$ 来确定 Δx 。
- 2) 通过对矩阵 Φ 进行 Cholesky 分解为 $\Phi = LL^T$, L 为下三角矩阵。
- 3) 对线性系统求解 $L\Delta \tilde{x} = -r_d$
- 4) 再对线性系统求解 $L^T \Delta x = \Delta \tilde{x}$ 得到 Δx 。

3.3. 结合 PCA 与 K-Means 不等式降维

高性能 QP 求解器虽可提升 Fast-MPC 的求解速度, 但当变量维度或不等式约束维度较大时, 计算负担仍然较重。文献[16]通过矩阵分块策略重构目标函数与约束条件, 利用调整控制时域的方式减少优化变量数量, 从而降低计算量。对于仅含不等式约束的二次规划问题, 采用对约束矩阵进行降维, 也能够保留主要约束信息的同时削减冗余约束, 进而提高求解效率。

对目标函数公式(17)的不等式约束 $(Px < h)$ 进行改写, P 为 $m \times n$ 矩阵首先, 对矩阵 P 计算行均值:

$$\mu_i = \frac{1}{n} \sum_{j=1}^n P_{i,j} \quad (18)$$

式中 $i = 1, 2, \dots, m$ 。

对 P 的行进行中心化:

$$P_{\text{centered}}(i,:) = P(i,:) - \mu_i \quad (19)$$

对中心化的数据执行 PCA, 以识别到不等式中捕捉最多方差的方向(主成分)。计算行的协方差:

$$C = \frac{1}{n-1} P_{\text{centered}} P_{\text{centered}}^T \quad (20)$$

特征值分解:

$$C = U\Sigma U^T \quad (21)$$

式中, $U \in \mathbb{R}^{m \times m}$ 包含特征向量, $\Sigma \in \mathbb{R}^{m \times m}$ 是特征值的对角矩阵。按特征值降序排列特征值和对应的特征向量。选择捕捉所需方差的前 k 个主成分, 并将中心化的不等式投影到这些主成分上, 选择前 k 个主成分:

$$U_k = U(:, 1:k) \quad (22)$$

投影矩阵:

$$P_{\text{projected}} = U_k^T P_{\text{centered}} \quad (23)$$

对投影后的约束进行聚类, 将相似的约束分组, 对 $P_{\text{projected}}^T$ 应用 K-means 聚类算法, 聚类数 c 的选取需要在约束表达能力与计算复杂度之间进行权衡。本文采用参数对比方式, 对不同 c 下的 QP 求解效率与轨迹跟踪性能进行综合评估, 并据此确定最终聚类数。

将每个聚类中的约束聚合, 形成一个简化后的约束集。对于每个聚类 $j, j = 1, 2, \dots, c$ 。识别聚类中的约束:

$$I_j = \{i \mid \text{约束 } i \text{ 属于聚类 } j\} \quad (24)$$

均值聚合:

$$P_{\text{reduced}}(j, :) = \frac{1}{|I_j|} \sum_{i \in I_j} P(i, :), \quad h_{\text{reduced}}(j) = \frac{1}{|I_j|} \sum_{i \in I_j} h(i) \quad (25)$$

通过仔细地聚合约束, 目标是确保简化问题的可行域是原始问题可行域的子集或接近的近似。原始问题的可行域:

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid Px \leq h\} \quad (26)$$

简化问题的可行域:

$$\mathcal{F}_{\text{reduced}} = \{x \in \mathbb{R}^n \mid P_{\text{reduced}} x \leq h_{\text{reduced}}\} \quad (27)$$

通过均值聚合, 确保 $\mathcal{F}_{\text{reduced}} \subseteq \mathcal{F}$, 因此简化问题的任何可行解对于原始问题也是可行的。鉴于简化问题的可行域位于原始问题的可行域内, 简化问题的最优解 x_{reduced}^* 对于原始问题是可行的。虽然 x_{reduced}^* 可能不是原始问题的精确最优解, 但它原始问题的最优目标值提供了一个上界。

目标函数被重构为:

$$\text{minimize } x^T Hx + g^T x \quad \text{subject to } P_{\text{reduced}} x \leq h_{\text{reduced}} \quad (28)$$

图 2 和图 3 为双移线工况下的仿真结果, 此时车辆速度为 90 km/h, 预测时域 $N_p = 20$, 控制时域 $N_c = 10$ 。其中, PK-Fast-MPC 采用 PCA 与 K-means 方法将不等式约束维数由 384 维降至 100 维, BB-Fast-MPC 则将控制时域由 10 缩减至 5。图 2 结果表明, PK-Fast-MPC 的求解时间明显低于 BB-Fast-MPC 和原始 Fast-MPC。结合图 3 可知, BB-Fast-MPC 在直线路段出现了一定振荡, 收敛效果较差; 而 PK-Fast-MPC 在降低计算时间的同时, 仍保持了较好的求解精度, 且在弯道处的跟踪性能优于原始 Fast-MPC。

表 1 对比了各求解器的平均求解时间、时间方差及最大最小求解时间等指标。虽然 BB-Fast-MPC 提升了计算效率, 但其控制结果存在明显震荡。相比之下, PK-Fast-MPC 的平均求解时间仅为 1.1 ms, 时间方差仅为 0.0014 ms, 表现出更高的求解效率与稳定性, 其求解速度较 Fast-MPC 提高了 10 倍以上。由

于 Fast-MPC 最大求解时间高达 54.1 ms 且波动较大, 其实时性受到限制。总体来看, PK-Fast-MPC 在效率、稳定性和控制效果之间取得了更好的平衡。

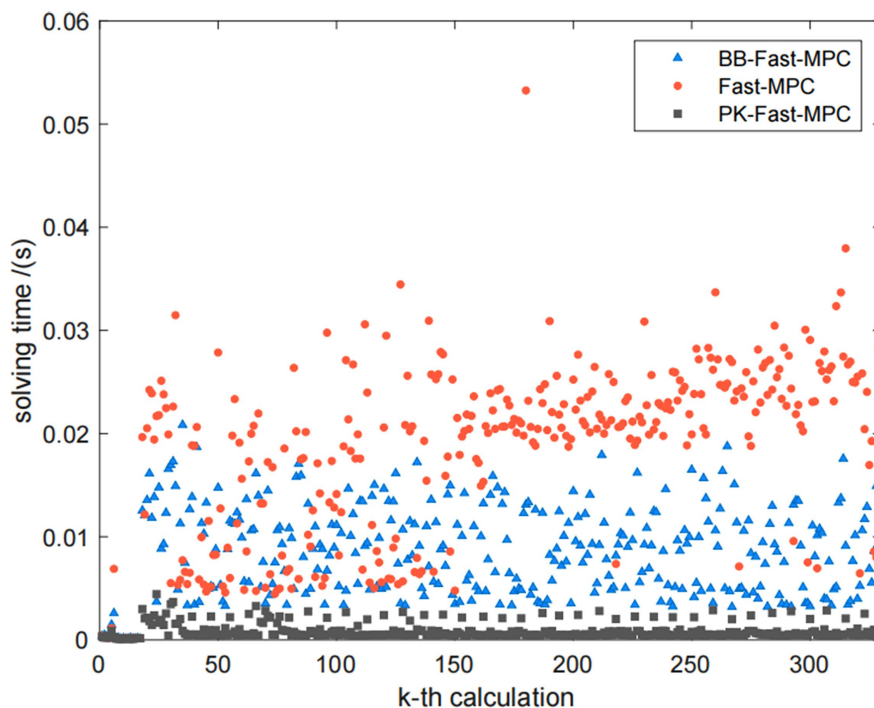


Figure 2. Scatter plot of solution time

图 2. 求解时间散点图

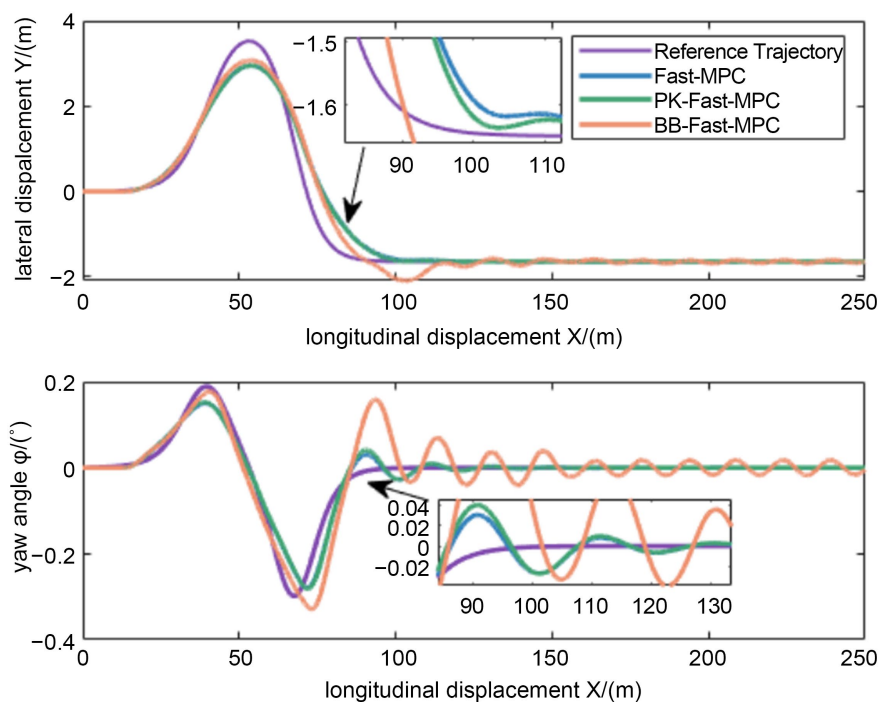


Figure 3. Diagram of tracking effect

图 3. 跟踪效果图

Table 1. Solver performance comparison
表 1. 求解器性能对比

求解器	PK-Fast-MPC	Fast-MPC	BB-Fast-MPC
平均求解时间/ms	1.1	18.7	8.6
求解时间方差/ms ²	0.0014	0.074097	0.019
最小求解时间/ms	0.098	0.1187	0.1002
最大求解时间/ms	12.3	54.1	20.8

4. 仿真结果

为验证所提出算法的有效性, 搭建了基于 Simulink 和 CarSim 的联合仿真平台, 对设计的控制器进行详细的仿真与测试。该平台能够有效模拟车辆的动力学和运动学特性, 确保测试环境的真实性和可靠性。传统的 MPC 问题常采用主动集成模型(ASM)、内点法(IPM)和交替方向乘法(ADMM)来求解。为更全面地评估所提出算法的性能, 本文将其与这些经典算法进行了对比分析, 重点考察了求解速度、控制精度和系统稳定性等关键指标。

车辆的动力学参数如表 2 所示。相关参数主要来源于 CarSim 整车模型的参数设置, 并真实车辆动力学典型取值进行统一设定。

Table 2. Vehicle dynamics parameters
表 2. 车辆参数

参数	数值	单位
车重	1723	kg
轴距	2.7	m
横向转动惯量	4331.6	kg/m ²
侧倾刚度	2328/2653	N·m/rad
侧倾阻尼	47298/37311	N·m/rad
轮胎刚度	61900	N/rad
轮胎旋转惯量	0.9	Kg·m ²
道路附着系数	0.85	-

构建仿真模型时, 针对横向 MPC 控制器的具体参数设计进行了细致规划, 其关键参数设置如表 3 所示。本设计旨在确保控制器在复杂动态环境中能够稳定、准确地实现轨迹跟踪目标, 同时优化控制性能以满足实时性要求。

Table 3. PK-Fast-MPC control parameters
表 3. PK-Fast-MPC 控制参数

参数	数值	单位
采样周期(T)	0.05	s
预测时域(N_p)	20	s
控制时域(N_c)	10	s
降维后维度 k	100	-
松弛因子 ρ	1000	-

续表

预测输出控制权重矩阵 Q	$\begin{bmatrix} 300 & 0 \\ 0 & 100 \end{bmatrix}$	-
控制增量权重 R	10	-
PCA-Fast-MPC 障碍参数	0.02	-
求解最大迭代次数	5	-

在图 4 所展示的 MPC 横向结构图控制系统主要由前馈控制和反馈控制组成。具体而言, 前馈控制部分主要根据式(23)计算前馈控制量期望轨迹的曲率, 并将计算结果输入转向控制。在反馈控制部分, 通过分析构建状态空间方程车辆动力学模型, 通过 PCA 与 K-means 技术结合相对不等式约束进行高效的降维处理, Fast-MPC 计算最优控制量, 并反馈计算结果。该算法能够迅速响应系统状态的变化, 并生成相应的控制指令, 最终通过反馈机制调整车辆行为, 确保跟踪性能的稳定与准确。

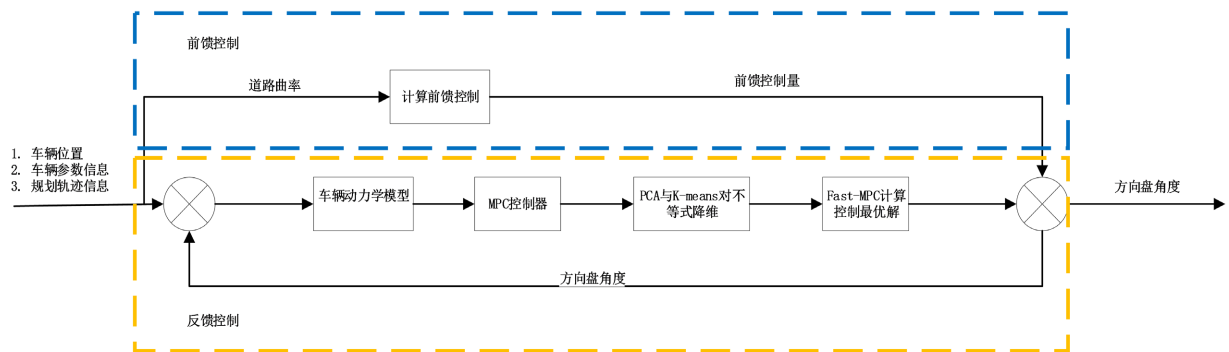


Figure 4. MPC lateral control system
图 4. MPC 横向控制系统

通常在车辆驾驶测试中, 双移线条件频繁出现在测试部分, 许多学者也使用这种条件测试自动驾驶车辆的轨迹跟踪能力。需要的轨迹由下面的公式给出:

$$\begin{cases} Y_{ref}(X) = \frac{d_{y1}}{2} [1 + \tanh(z_1)] - \frac{d_{y2}}{2} [1 + \tanh(z_2)] \\ \varphi_{ref}(X) = \arctan \left[d_{y1} \left(\frac{1}{\cosh(z_1)} \right)^2 \left(\frac{1.2}{d_{x1}} \right) - d_{y2} \left(\frac{1}{\cosh(z_2)} \right)^2 \left(\frac{1.2}{d_{x2}} \right) \right] \end{cases} \quad (29)$$

式中, $z_1 = 2.4(X - 27.19)/d_{x1} - 1.2$, $d_{x1} = 25$, $z_2 = 2.4(X - 56.46)/d_{x2} - 1.2$, $d_{x2} = 21.95$, $d_{y1} = 4.05$, $d_{y2} = 5.7$ 。

在 PK-Fast-MPC 中, PCA 降维维数 k 的选择直接影响轨迹跟踪性能与求解效率。原始不等式矩阵维度为 384×31 , 图 5 和图 6 给出了不同 k 取值下的跟踪结果, 为定量比较不同主成分数 k 对求解效率的影响, 统计了不同 k 取值下的平均计算时间, 如表 4 所示。结果表明, 当 $k = 40$ 时, 车辆在前段能够较好跟踪参考轨迹, 但在 100 m 后逐渐出现明显偏差, 说明降维过度会导致关键动态信息丢失, 进而影响后段跟踪性能。当 $k = 60$ 和 $k = 80$ 时, 跟踪效果有所改善, 但在 100~150 m 区间仍存在一定振荡; 其中 $k = 80$ 时振荡较小, 说明其保留了更多有效信息。相比之下, 当 $k = 100$ 和 $k = 120$ 时, 跟踪曲线与参考轨迹基本重合, 且整体稳定性较好, 表明此时降维后仍能较完整地保留系统动态特征。由于 k 由 100 增大至 120 后性能提升不明显, 因此 $k = 100$ 可视为较优选择。

Table 4. Average computation time for different values of k
表 4. 不同取值 k 的平均计算时间

k	40	60	80	100	120
计算平均时间/ms	0.482	0.542	0.625	0.806	1.026

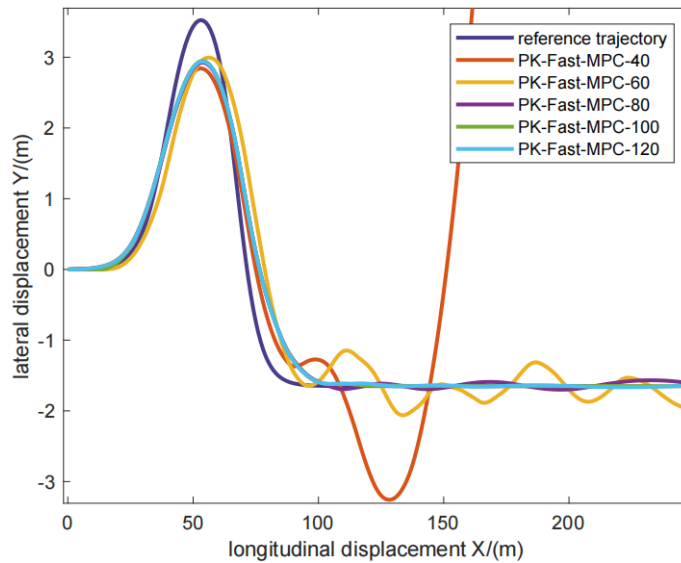


Figure 5. Trajectory tracking effect under different PCA dimensions
图 5. PCA 不同维度跟踪效果图

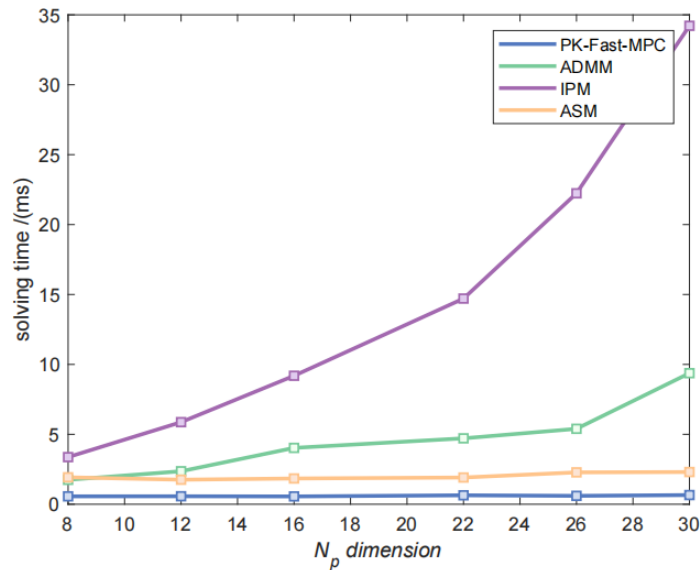


Figure 6. Diagram of solver solution time under different N_p dimensions
图 6. 不同 N_p 维度下求解器求解时间图

图 6 为不同求解器在不同 N_p 维度下求解器求解时间图,从图 6 可以看出在所有维度下,PK-Fast-MPC 求解器的求解时间始终是最低的,表现出极高的求解效率。随着 N_p 维度的增加,其求解时间变化不大,表明该求解器对问题规模的扩展具有较强的适应性和鲁棒性。ADMM 求解器的求解时间略高于 PK-Fast-MPC 求解器,但整体表现也相对稳定,随着 N_p 维度的增加,求解时间呈现平缓的线性增长。IPM 求解器

在 N_p 维度较小(低于 20)时求解时间略高于 ADMM 求解器, 但随着 N_p 维度的增加, 求解时间急剧上升, 尤其在 N_p 维度达到 30 时, 求解时间明显增大。ASM 求解器比较稳定, 而且完全不会随着 N_p 维度增加而增加, 但仍高于 PK-Fast-MPC 求解器。PK-Fast-MPC 求解器在不同维度下表现最为优越, 特别适合需要高效求解和实时性的自动驾驶应用。

在 60 km/h 中速行驶的双移线场景下, 各类求解器所求解的纵向位移和横摆角如图 7 所示。结果表明, PK-Fast-MPC 的求解结果最为贴合参考轨迹。传统 MPC 通过 ASM 和 IPM 两种方法进行求解, 虽然这两种方法的求解结果几乎一致, 但由于 IPM 的内在特性, 导致初始阶段存在一定的振荡现象。相较之下, ADMM 求解器的性能优于传统方法, 但略逊于 PK-Fast-MPC 求解器。根据图 8 的显示, PK-Fast-MPC 在纵向误差方面表现更为优异, 其最大误差控制在 0.5 米以内, 充分满足了自动驾驶对误差范围的要求。这表明 PK-Fast-MPC 求解器在控制精度上具有显著优势。

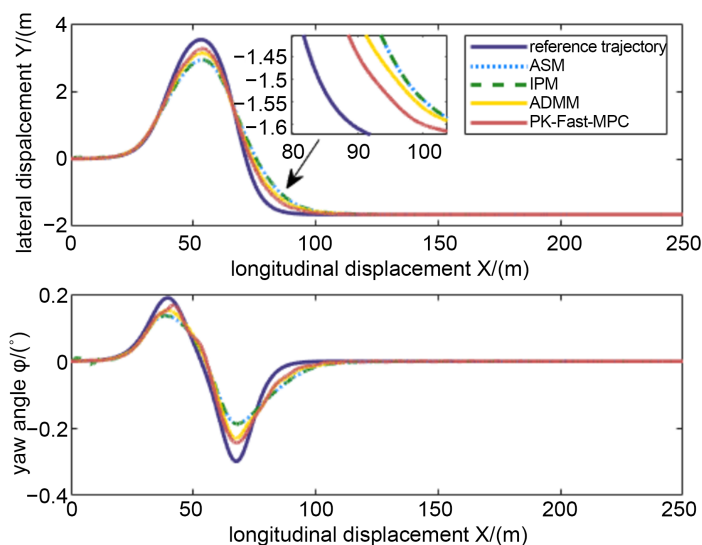


Figure 7. Diagram of trajectory tracking effect of the solver at 60 km/h
图 7. 60 km/h 求解器跟踪效果图

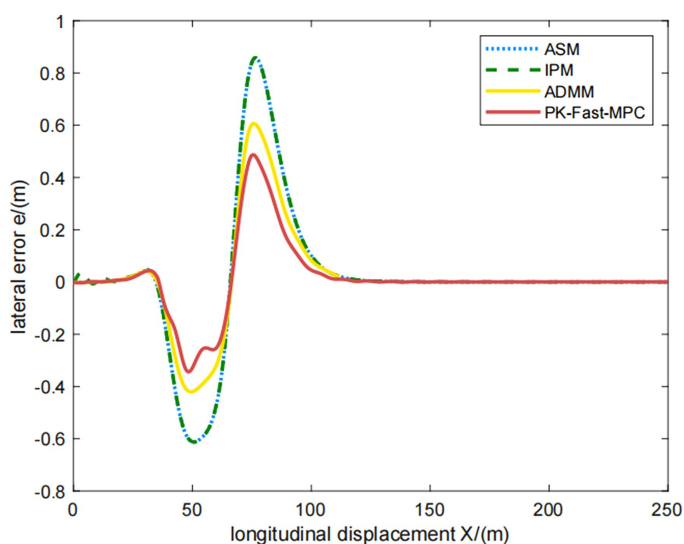


Figure 8. Error plot of the solver at 60 km/h
图 8. 60 km/h 求解器误差图

图 9 展示了在 60 km/h 双移线工况下, 各种求解器进行二次规划的求解速度。结果显示, Active Set 求解器的平均求解速度为 1.9 ms, Interior Point 求解器为 3.1 ms, ADMM 求解器为 2.6 ms, 而 PK-Fast-MPC 求解器的平均求解速度则显著优于其他三种求解器, 仅为 0.55 ms。从图 8 中还可以观察到, Interior Point 求解器、Active Set 求解器和 ADMM 求解器在部分情况下表现出明显的不稳定性, 导致求解速度显著减慢。然而, PK-Fast-MPC 求解器的求解速度始终稳定在 5 ms 以内, 展现出极高的稳定性, 这使得其非常适合对稳定性有严格要求的自动驾驶工况。

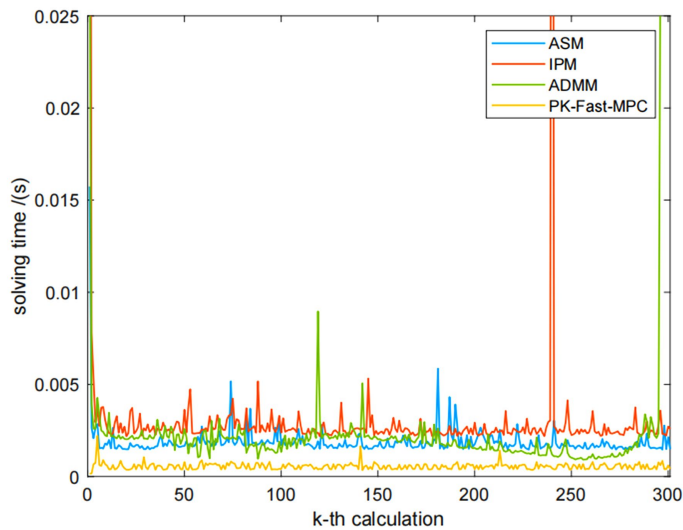


Figure 9. Diagram of solver solution time
图 9. 求解器求解时间图

图 10 展示了在 30 km/h 低速双移线工况下, 轨迹跟踪的横摆角和纵向位移表现。四种求解器均表现出良好的跟踪效果。然而, PK-Fast-MPC 求解器在 50 米和 70 米处的转弯位置相较于其他三种求解器, 更加贴近参考轨迹。这表明 PK-Fast-MPC 求解器在低速复杂工况下具备更高的轨迹跟踪精度。

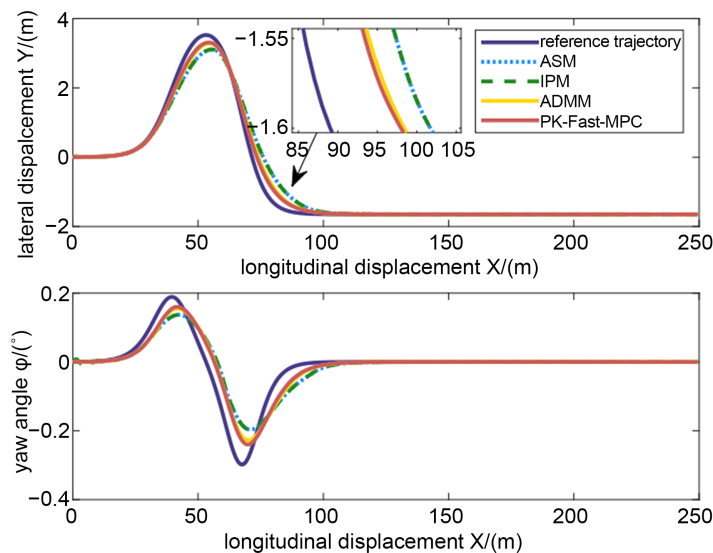


Figure 10. Tracking performance of the controller at 30 km/h
图 10. 30 km/h 控制器跟踪效果图

图 11 展示了在 90 km/h 高速双移线工况下的轨迹跟踪横摆角和纵向位移表现。从图中可以看出, ADMM 求解器在起始点 50 米处的表现略优于其他三种求解器。然而, 由于 ADMM 求解器在 50 米拐弯处出现了明显的急转, 导致出弯后无法有效跟踪参考轨迹。相比之下, PK-Fast-MPC 求解器在各个阶段的表现均优于其他求解器, 尤其在 90 米处优先收敛并重合于参考轨迹。尽管在 80 米拐弯处其横摆角出现了轻微的超调, 但对整体结果影响较小, 仍能维持较高的跟踪精度。

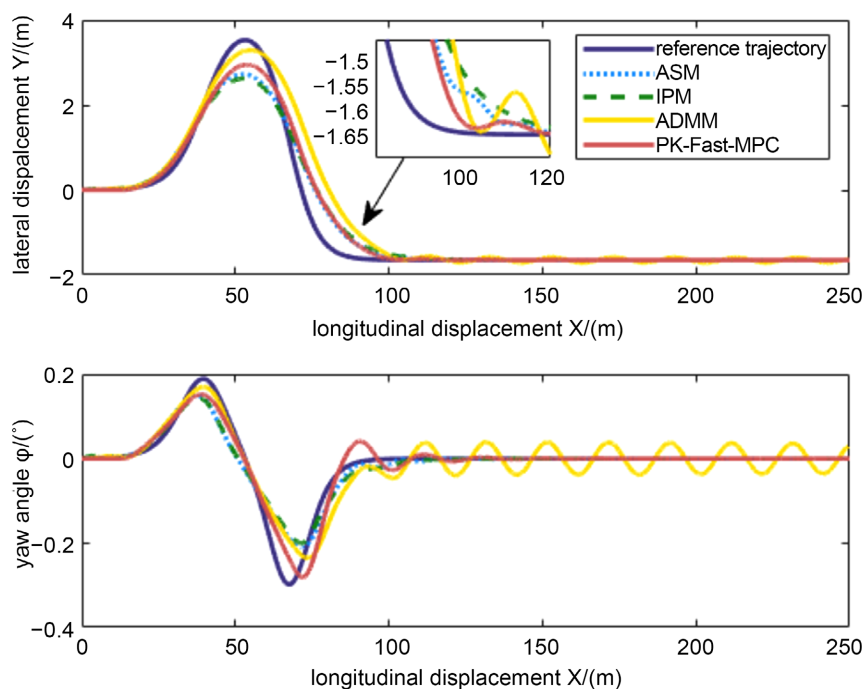


Figure 11. Tracking performance of the solver at 90 km/h
图 11. 90 km/h 求解器跟踪效果图

由于 PK-Fast-MPC 求解器和其他求解器的结果从图片中看并无法直接的看出好坏, 所以引入了另一种误差评判方法。使用了车辆行驶轨迹和期望轨迹以获得每个控制器的均方根误差(RMSE), 如下式所示:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y(i) - Y_{ref}(i))^2} \quad (30)$$

PCA-Fast-MPC 求解器的 RMSE 为 0.2746 m, active set 求解器的 RMSE 为 0.2862 m, interior point 求解器的 RMSE 为 0.3184 m, ADMM 求解器的 RMSE 为 0.4249 m。PK-Fast-MPC 求解器便显出最高的精度, 误差最小。

5. 结论

本文提出了一种基于 Fast-MPC 结合 PCA 与 K-means 聚类的无人驾驶车辆全速域轨迹跟踪优化方法。通过对约束矩阵使用 PCA 降维, 再引入 K-means 聚类算法对降维后的数据进行分类与聚合, 显著降低了二次规划(QP)问题中的不等式约束维度。实验结果表明, 结合 PCA 与 K-means 的 Fast-MPC 方法在双移线场景下求解速度比传统方法提升了 10 倍, 最大减少了约束维度的 60%, 同时在高速动态环境中的轨迹跟踪表现更加优异, 消除了求解过程中的振荡现象。这一研究为未来无人驾驶车辆在复杂环境中的实时轨迹跟踪应用提供了更高效、可靠的解决方案, 并为大规模优化问题的快速求解奠定了基础。

参考文献

- [1] Guo, J., Kurup, U. and Shah, M. (2020) Is It Safe to Drive? An Overview of Factors, Metrics, and Datasets for Driveability Assessment in Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, **21**, 3135-3151. <https://doi.org/10.1109/tits.2019.2926042>
- [2] Li, L., Li, J. and Zhang, S. (2021) Review Article: State-of-the-Art Trajectory Tracking of Autonomous Vehicles. *Mechanical Sciences*, **12**, 419-432. <https://doi.org/10.5194/ms-12-419-2021>
- [3] Huang, Z., Li, H., Li, W., Liu, J., Huang, C., Yang, Z., et al. (2021) A New Trajectory Tracking Algorithm for Autonomous Vehicles Based on Model Predictive Control. *Sensors*, **21**, Article No. 7165. <https://doi.org/10.3390/s21217165>
- [4] Stano, P., Montanaro, U., Tavernini, D., Tufo, M., Fiengo, G., Novella, L., et al. (2023) Model Predictive Path Tracking Control for Automated Road Vehicles: A Review. *Annual Reviews in Control*, **55**, 194-236. <https://doi.org/10.1016/j.arcontrol.2022.11.001>
- [5] 刘聪, 刘辉, 韩立金, 等. 基于学习型滑模预测控制的无人驾驶车辆非结构化环境轨迹跟踪及稳定性控制[J]. 机械工程学报, 2024(10): 399-412.
- [6] 杜荣华, 胡鸿飞, 高凯, 等. 基于变预测时域 MPC 的自动驾驶汽车轨迹跟踪控制研究[J]. 机械工程学报, 2023, 58(24): 275-288.
- [7] 熊璐, 杨兴, 卓桂荣, 等. 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, 56(10): 127-143.
- [8] Domahidi, A., Zraggen, A.U., Zeilinger, M.N., Morari, M. and Jones, C.N. (2012). Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control. 2012 51st *IEEE Conference on Decision and Control (CDC)*, Maui, 10-13 December 2012, 668-674. <https://doi.org/10.1109/cdc.2012.6426855>
- [9] Ferreau, H.J., Bock, H.G. and Diehl, M. (2008) An Online Active Set Strategy to Overcome the Limitations of Explicit MPC. *International Journal of Robust and Nonlinear Control*, **18**, 816-830. <https://doi.org/10.1002/rnc.1251>
- [10] 单睿斌, 李秋红, 何凤林, 等. 基于 ADMM 算法的航空发动机模型预测控制[J]. 北京航空航天大学学报, 2019, 45(6): 1240-1247.
- [11] Li, S.E., Jia, Z., Li, K. and Cheng, B. (2014) Fast Online Computation of a Model Predictive Controller and Its Application to Fuel Economy-Oriented Adaptive Cruise Control. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 1199-1209. <https://doi.org/10.1109/tits.2014.2354052>
- [12] Marcucci, T. and Tedrake, R. (2020) Warm Start of Mixed-Integer Programs for Model Predictive Control of Hybrid Systems. *IEEE Transactions on Automatic Control*, **66**, 2433-2448. <https://doi.org/10.1109/tac.2020.3007688>
- [13] Salzmann, T., Kaufmann, E., Arrizabalaga, J., Pavone, M., Scaramuzza, D. and Ryll, M. (2023) Real-Time Neural MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms. *IEEE Robotics and Automation Letters*, **8**, 2397-2404. <https://doi.org/10.1109/lra.2023.3246839>
- [14] Li, Y., Li, S.E., Jia, X., Zeng, S. and Wang, Y. (2022) FPGA Accelerated Model Predictive Control for Autonomous Driving. *Journal of Intelligent and Connected Vehicles*, **5**, 63-71. <https://doi.org/10.1108/jicv-03-2021-0002>
- [15] 龚建伟, 刘凯, 齐建永. 无人驾驶车辆模型预测控制[M]. 北京: 北京理工大学出版社, 2020.
- [16] 高少帅. 考虑实时性的主动变道运动规划与模型预测控制研究[D]: [硕士学位论文]. 长春: 吉林大学, 2023.
- [17] Wang, Y. and Boyd, S. (2009) Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology*, **18**, 267-278. <https://doi.org/10.1109/tcst.2009.2017934>